Smart Package Robot Helpfile

USER MANUAL

© 2023 Theo Gottwald * https://www.smart-package.com
Theo Gottwald * https://www.smart-package.com

1. 1. Starting Guides 31

1.1 1.1. About the program ... 32
1.2 1.1.1 What else is it good for? ... 35
1.3 1.1.2 What you See is what you get: as Standalon EXE-File ... 37
1.4 1.2. System Requirements ... 39
1.5 1.3. How to Start ... 41
1.6 1.3.1 The Scripting Process ... 44
1.7 1.3.2 Commands you should start with ... 50
1.8 1.3.3 Compiling a Standalone Executable ... 53
1.9 1.3.4 Compiling a package (SPRE) ... 60
1.10 1.3.5 The Script Execution Process ... 68
1.11 1.4. The Prototype Parameter System ... 73
1.12 1.5. Speed Considerations ... 80
1.13 1.5.1 FAQ - How to 86
1.14 1.5.2 Using Variables ... 92
1.15 1.5.3 More about Variables ... 96
1.16 1.5.5 Global and Local Variables ... 100
1.17 1.5.6 Using the data stacks ... 104
1.18 1.5.7 Using Text-Strings ... 110
1.19 1.5.8 Basic String Operations ... 113
1.20 1.5.9. Organizing data items in blocks ... 117
1.21 1.6.1. Program Flow Control ... 121
1.22 1.6.2. Looping around ... 126
1.23 1.6.3 Flexible Loops ... 132
1.24 1.7. Application and PID ... 137
1.25 1.7.1 Applications and Threads ... 145
1.26 1.8. Selecting the Application (Topwindow) ... 150
1.27 1.8.1 The "Client" and the "Non-Client" Parts of a window ... 156
1.28 1.9. Locating Child-Windows ... 162
1.29 1.9.1 Types of controls ... 170
1.29.1 1.9.1.1 Types of Controls ... 170
1.29.2 1.9.1.2 Working with controls ... 174
1.29.3 1.9.1.3 Button Control ... 182
1.29.4 1.9.1.4 Checkmark Control ... 184
1.29.5 1.9.1.5 Radio Button Control ... 191
1.29.6 1.9.1.6 Edit-Textbox Control ... 194
1.29.7 1.9.1.7 Dropdown- Combobox Control ... 198
1.29.8 1.9.1.8 Treeview Control ... 202

© 2023 Theo Gottwald * https://www.smart-package.com

1.29.9 1.9.1.9 Page- Tab Control ... 208
1.29.10 1.9.1.A Listbox Control ... 210
1.30 1.9.2 WEB-based controls ... 212
1.30.1 1.9.2.1 WEB-based Textbox ... 212
1.31 1.A Special Issues ... 215
1.32 1.B Error Management ... 224
1.33 1.C Direct Engine and Compatible Engine ... 231
1.34 Contact ... 236
1.34.1 Hotline ... 236
1.35 General Issues ... 238
1.35.1 C. Licence Issues ... 238
1.35.2 B. General Explanation ... 240
1.35.3 A. Components of the System ... 242

2. 2. Components 245

2.36 1. PreProcessor & EXE-Compiler ... 246
2.36.1 1.0 Pre-Processor ... 246
2.36.2 1.1 #INC: - Pre-Processor File-Include ... 248
2.36.3 1.2 #ONCE / # OEND - Multiple Include Protection ... 251
2.36.4 1.3 : - MACRO-Definitions ... 253
2.36.5 1.4 #IF ... #EIF - Decisions in Macros ... 262
2.37 2. EXE-Compiler ... 267
2.37.1 2.0.0 Smarty's EXE-Compiler ... 267
2.37.2 2.1.1 '#RAU: - Run As User ... 271
2.37.3 2.1.2 #AOS: - Choose valid OS for executable ... 273
2.37.4 2.1.3 '#EXE: - Set Executable Path and Name ... 277
2.37.5 2.1.4 '#EXV: - Execute on System Desktop ... 280
2.37.6 2.1.5 '#HID: - Hidden Desktop Scripts ... 283
2.37.7 2.1.7 '#SPI: - Special Instructions ... 287
2.37.8 2.1.8 '#INC: - Include Files and Folders into the Executable ... 290
2.37.9 2.1.9 '#EXC: - Exclude Files or Dirs from Executable ... 295
2.37.10 2.1.A #LIB: - User-Library Path ... 298
2.37.11 2.1.B #CRY: - enCrypt the Script ... 302
2.37.12 2.1.C '#AFT: - RUn AFTER Compile ... 310
2.37.13 2.2.1 '#ICO: - Choose own Icon ... 313
2.37.14 2.2.2 '#ICF: - Icon from Icon-Folder ... 316
2.37.15 2.2.3 '#ICS:- Icon from internal Icon-Set ... 318
2.37.16 2.3.0. '#MAN: - Manifest for Executable ... 321
2.37.17 2.3.1. '#MAU: - UI-Access for Manifest for Executable ... 325
2.37.18 2.4.0. '#VER: - Version-Informations ... 328
2.37.19 2.5.0. '#CIL: - Direct Compiler Interface ... 332
2.38 3. System-Variable and Specialfolders ... 334
2.38.1 3.1 Systemvariables ... 334

*© 2023 Theo Gottwald * https://www.smart-package.com*

2.38.2 3.2 Standard-Search Pattern ... 341
2.38.3 3.3 Specialfolders ... 345
2.38.4 3.4 System-Colors ... 349
2.38.5 3.5 Hexadezimal, Decimal and Binary and more ... 352
2.38.6 3.6 Using Quadrant-Coordinates ... 356
2.39 4. Special features ... 361
2.39.1 4.1 : - Definiton-Command ... 361
2.39.2 4.2 :on_Error - Label ... 364
2.39.3 4.3 Variables, global and local ... 366
2.39.4 4.4 The global and local Stack ... 369
2.39.5 API-Error-Codes ... 372
2.39.6 Regular Expressions ... 381
2.39.7 Language-ID's ... 386
2.39.8 MSI - Parameters ... 389

3. 3. Script Language 392

3.40 ! Smarty's Script language ... 393
3.41 ! User-Macros ... 399
3.41.1 ! User-Macros ... 399
3.41.2 Firefox ... 401
3.41.3 Graphical Subprograms ... 406
3.41.4 Point-Convert-Lib ... 412
3.41.5 Testing-Macros ... 415
3.42 Background Operation ... 421
3.42.1 ! Smarty's Background Operations ... 421
3.42.2 DRS. - Drop Result ... 427
3.42.3 GRS. - Get Result ... 430
3.42.4 IRS. / NRS. - If Result available ... 434
3.42.5 WRS. - Wait for Result ... 436
3.43 BIT. - Work with Bits ... 441
3.43.1 ! Smarty's BIT. - Operations ... 441
3.43.2 BIT.16to32 - concatenate 2 Words to 32 Bit DWord ... 446
3.43.3 BIT.16to8 - Split 1 Word into 2 Bytes ... 449
3.43.4 BIT.32to16 - Split 1 DWord into 2 Words ... 452
3.43.5 BIT.8to16 - concatenate 2 Bytes to 16 Bit Word ... 455
3.43.6 BIT.AND - Use logical AND with Bits ... 458
3.43.7 BIT.Clear - Set Bit to 0 ... 461
3.43.8 BIT.Get - check if Bit is set ... 464
3.43.9 BIT.LShift - Shift Bits to Left, insert 0 ... 467
3.43.10 BIT.NOT - Use logical NOT with Bits ... 470
3.43.11 BIT.OR - Use logical OR with Bits ... 473
3.43.12 BIT.RotateLeft - Rotate Bits Left ... 476
3.43.13 BIT.RotateRight - Rotate Bits Right ... 479
3.43.14 BIT.RShift - Shift Bits to Right, insert 0 ... 482

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.43.15 BIT.Set - Set Bit to 1 ... 485
3.43.16 BIT.Show - Show Bits in Binary-Numbers ... 488
3.43.17 BIT.Store - Set Bit to 1 or 0 ... 491
3.43.18 BIT.ToBin - change Parameter to Binary-Format ... 494
3.43.19 BIT.Toggle - Switch Bit ... 498
3.43.20 BIT.ToVal - change Parameter to Value-Format ... 501
3.43.21 BIT.XOR - Use logical XOR with Bits ... 505
3.44 Browser and Office (Acc.-Engine) ... 508
3.44.1 OFC. - Office-Commands ... 508
3.44.1.1 MS-Excel Scripting ... 508
3.44.1.1.1 !Excel-Scripting ... 508
3.44.1.1.2 !Sample-Scripts: Neuro-Net Cell ... 513
3.44.1.1.3 CLOSE ... 516
3.44.1.1.4 EXIST ... 518
3.44.1.1.5 FORMULA GET ... 520
3.44.1.1.6 FORMULA SET ... 522
3.44.1.1.7 GET_CELL...BY COORD ... 524
3.44.1.1.8 GET_CELL...BY RANGE ... 527
3.44.1.1.9 LASTROW / LASTCOLUMN ... 530
3.44.1.1.10 LETTER_TO_NUMBER ... 532
3.44.1.1.11 NUMBER_TO_LETTER ... 534
3.44.1.1.12 OPEN ... 536
3.44.1.1.13 PAGE_UPDOWN ... 538
3.44.1.1.14 SAVE ... 540
3.44.1.1.15 SAVEAS ... 542
3.44.1.1.16 SCROLL_LEFTRIGHT ... 545
3.44.1.1.17 SCROLL_UPDOWN ... 547
3.44.1.1.18 SENDKEY ... 549
3.44.1.1.19 WRITE ... 552
3.44.2 ! Smarty's Browser and Office Commands ... 554
3.44.3 AFT. / AFF.. - Acc.- Find Text ... 558
3.44.4 AGR. / AGF. - Accessibility Get Rect ... 564
3.44.5 AMS. - Acc.-Multi Scan ... 570
3.44.6 AVF. - Accesibility Visual Find ... 575
3.44.7 CBR. - Control Browser ... 582
3.44.8 GAI. - Get-Acc-Information ... 591
3.44.9 IBR. / NBR. - If Browser ... 595
3.44.10 LAP. - Locate at Point ... 599
3.44.11 LBR. - Launch (Mini-)Browser ... 601
3.44.12 NAV - NAVigate to Object ... 606
3.44.12.1 ! NAV. - NAVigate in Dialog ... 606
3.44.12.2 NAV - "Down" ... 612
3.44.12.3 NAV - "DownAll" ... 613
3.44.12.4 NAV - "Left" ... 614
3.44.12.5 NAV - "LeftAll" ... 615
3.44.12.6 NAV - "Right" ... 616
3.44.12.7 NAV - "RightAll" ... 617

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.44.12.8 NAV - "Up" ... 618
3.44.12.9 NAV - "UpAll" ... 619
3.44.12.10 NAV - Example 1 ... 620
3.44.12.11 NAV - Example 2 ... 621
3.44.13 Roles and States from the Acc.-System (E8) ... 622
3.44.14 SAO. - Search Accessible Object ... 625
3.44.15 WBR. - Wait Browser ... 630
3.45 Control specific operations ... 635
3.45.1 BTC. - Button-Check ... 635
3.45.2 CDD. - Combo Drop Down ... 638
3.45.3 GBC. - Get-Button-Check ... 640
3.45.4 GCI. - Get-Combo-Item ... 642
3.45.5 LBE. - ListBox Entry ... 649
3.45.6 LVI. - Listview Command ... 651
3.45.7 SCI. - Set-Combo-Item ... 657
3.45.8 SMH. - Set Menu Hook ... 660
3.45.9 TVF. - Tree View Find ... 665
3.45.10 TVI. - Tree View Item ... 668
3.46 Data-Block Commands ... 671
3.46.1 ! Smarty's Data-Block Commands ... 671
3.46.2 CNB. - Create New Block ... 672
3.46.3 GBE. - Get Block Element ... 675
3.46.4 GBN. - Get Block Number ... 677
3.46.5 GFB. - Get-From-Block ... 680
3.46.6 SBE. - Set Block Element ... 682
3.46.7 SIB. - Set-In-Block ... 685
3.47 Date and Time Calculation ... 688
3.47.1 ! Smarty's Date & Time Calculations ... 688
3.47.2 DAT. - Date and Time Information (System independent) ... 689
3.47.3 DTC. - Date and Time Calculation ... 693
3.47.4 IAT. / NAT. - If After Timestamp ... 699
3.47.5 IBT. / NBT. - If Before Timestamp ... 704
3.47.6 IDC. / NDC. - If Date Condition ... 709
3.47.7 TCA. - Time Calculation ... 715
3.47.7.1 TCA.Add ... 717
3.47.7.2 TCA.Between ... 720
3.47.7.3 TCA.Diff ... 723
3.47.7.4 TCA.Divide ... 726
3.47.7.5 TCA.GetPart ... 729
3.47.7.6 TCA.Make ... 732
3.47.7.7 TCA.Overflow ... 735
3.47.7.8 TCA.Split ... 737
3.47.8 VDF. - Verify Date Format ... 739
3.47.9 VTF. - Verify Time Format ... 742
3.47.10 Time to Value Commands ... 745
3.47.10.1 DTV. - Date to Value ... 746

 *© 2023 Theo Gottwald * https://www.smart-package.com*

3.47.10.2 TTV. - Time to Value ... 751
3.47.10.3 VTD. - Value to Date ... 756
3.47.10.4 VTT. - Value to Time ... 760
3.48 Debugging - Commands ... 763
3.48.1 ! Smarty's Debugging Commands ... 763
3.48.2 DBM. - DeBugMode ... 765
3.48.3 DBO. - Debug Raw Output ... 767
3.48.4 DBP. - Debug Print ... 769
3.48.5 DBV. - Debug Print Variable ... 771
3.48.6 DIP. - Delayed Instruction Processing ... 773
3.48.7 DMP. - Dump internat Informations ... 775
3.48.8 GRI. - Get Run Information ... 778
3.48.9 PAU. - Pause ... 783
3.48.10 USM. - Use silent Mode ... 786
3.49 E-Mail Commands ... 789
3.49.1 ! Smarty's Send-Mail ... 789
3.49.2 SMA . - Send e-Mail ... 797
3.49.3 SMB. - Send Mail Background ... 799
3.49.4 SMP. - Set e-Mail Parameter ... 803
3.50 Encryption and Hash Functions ... 812
3.50.1 ! Smarty's Encryption and Hashing Commands ... 812
3.50.2 GEC. - Get Encryption ... 825
3.50.3 GMD. - Get Message Digest ... 831
3.50.4 HIT. - Hide Text ... 835
3.50.5 SHA. - Safer Hash Algo ... 840
3.50.6 XOR. - XOR - Encryption ... 843
3.51 Environment Variables and Global Signals ... 851
3.51.1 EVE. - Environment Variable Expand ... 851
3.51.2 EVG. - Environment Variable Get ... 854
3.51.3 EVS. - Environment-Variable-Set ... 857
3.51.4 IGS. / NGS. - If-Global-Signal (Mutex) ... 859
3.51.5 SGS. - Set Global Signal (Mutex) ... 861
3.52 Execute-Commands ... 867
3.52.1 ! Smarty's Execution Commands ... 867
3.52.2 EXA. - Execute Asynchron ... 872
3.52.3 EXB. - Execute-Blocking ... 876
3.52.4 EXC. - Execute Batch Command ... 879
3.52.5 EXD. - Execute Direct ... 881
3.52.6 EXE. - Execute Programm ... 883
3.52.7 EXF. - Execute File/Document ... 885
3.52.8 EXH. - Execute-Handle ... 890
3.52.9 EXM. - Execute MSI ... 892
3.52.10 EXO. - Execute Open Document ... 895
3.52.11 EXR. - Executer-from-Registry ... 902

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.52.12 EXS. - Execute Shell ... 905
3.52.13 EXU. - Execute Uninstall ... 910
3.52.14 EXV. - Execute via System-Desktop ... 912
3.52.15 EXX. - Execute-Extended ... 916
3.53 FEX. - Enumerations ... 920
3.53.1 ! Smarty's FEX-Commands ... 920
3.53.2 EFX. - Exit - FEX ... 922
3.53.3 FEF. - For-Each-File(/Dir) ... 924
3.53.4 FEL. - For-Each-Line ... 927
3.53.5 FEM. - For Each Member ... 929
3.53.6 FEP. - For-Each-Process ... 932
3.53.7 FER. - For-Each-Registry ... 935
3.53.8 FEV. - For-Each-Variable(-Line) ... 937
3.54 Filesystem-Commands ... 940
3.54.1 ! Smarty's File-Commands ... 940
3.54.2 ATF. - Append-to-File ... 946
3.54.3 CFF. - Copy-From-File ... 948
3.54.4 CFI. - Copy-From_IniFile ... 951
3.54.5 CHD. - Change Directory and Drive ... 955
3.54.6 COP. - Copy File ... 957
3.54.7 CSC. - Create ShortCut ... 960
3.54.8 CTF. - Copy-To-File ... 964
3.54.9 CTI. - Copy-To-Inifile ... 966
3.54.10 CVF. - Convert File ... 969
3.54.11 DCP. - Directory Copy ... 971
3.54.12 DEL. - Delete File ... 974
3.54.13 FIF. - Find in File ... 977
3.54.14 FIL. - File-Commands ... 983
3.54.14.1 FileDiff ... 983
3.54.14.2 Compare ... 987
3.54.14.3 Join Delete ... 990
3.54.14.4 Join File ... 994
3.54.14.5 Length ... 998
3.54.14.6 Split File ... 1001
3.54.14.7 Generate numbered Files ... 1006
3.54.14.8 Find free Number ... 1009
3.54.14.9 Generate numbered Name ... 1012
3.54.14.10 Move numbered Files ... 1015
3.54.14.11 Swap Files ... 1018
3.54.15 FPC. - Filepath Physical Convert ... 1020
3.54.16 GFD. - Get File Date ... 1023
3.54.17 GFI. - Get File Information ... 1026
3.54.18 GFL.-Get File Language ... 1031
3.54.19 GFT. - Get-File-Token ... 1037
3.54.20 GFV. - Get File Version ... 1040
3.54.21 GLC. - Get-Line-Count ... 1043

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.54.22 GSI. - Get Shortcut Information ... 1046
3.54.23 IEF. / NEF. - If - exist - File ... 1050
3.54.24 IVA. - If Volume available ... 1053
3.54.25 LFF. - Line-From-File ... 1056
3.54.26 LFN. - Long-File-Name ... 1058
3.54.27 LFT. - Line-From-Text ... 1060
3.54.28 LOF. - Load-Text-File ... 1063
3.54.29 MDF. - Make-Direktory-File ... 1066
3.54.30 MKD. - Make-Directory ... 1068
3.54.31 MOV. - Move File/Dir ... 1070
3.54.32 REN. - Rename File/Dir ... 1074
3.54.33 RMD. - Remove-Directory ... 1077
3.54.34 SFD. - Set File Date ... 1080
3.54.35 SHF. - Shell-File-Operation ... 1082
3.54.36 WFF. - Wait-For-File ... 1085
3.55 Image/Pixel-Color Operations ... 1088
3.55.1 ! Smarty's Pixel-Operations ... 1088
3.55.2 ! Smarty's Pixel-Processing ... 1092
3.55.3 Additional Commands ... 1102
3.55.3.1 CAP. - Capture Screenshot ... 1103
3.55.3.2 CCD. - Calculate Color Distance ... 1109
3.55.3.3 CCV. - Color Conversion ... 1117
3.55.3.4 DPI. - Get Desktop DPI-Settings ... 1123
3.55.3.5 IDP. / NDP. - If desktop dpi Setting ... 1126
3.55.3.6 QFR. - Quadrant from RECT ... 1130
3.55.4 ANA. - Imageworks ... 1136
3.55.4.1 !The Human-Eye Recorder ... 1140
3.55.4.2 Blit-Operations ... 1155
3.55.4.2.1 !Blit-Modes ... 1156
3.55.4.2.2 ANA.MixTar ... 1162
3.55.4.2.3 ANA.BlendTo ... 1168
3.55.4.2.4 ANA.Blit ... 1172
3.55.4.2.5 ANA.BlitCol ... 1175
3.55.4.2.6 ANA.BlitShift ... 1179
3.55.4.3 Change Resolution ... 1185
3.55.4.3.1 ANA.CutRect ... 1186
3.55.4.3.2 ANA.EqualRes ... 1189
3.55.4.3.3 ANA.GetRes ... 1192
3.55.4.3.4 ANA.GetXYFromBytePos ... 1195
3.55.4.3.5 ANA.GetXYFromPixelPos ... 1198
3.55.4.3.6 ANA.Grab ... 1201
3.55.4.3.7 ANA.ResizeTo ... 1204
3.55.4.4 Color Table Operations ... 1209
3.55.4.4.1 ANA.AnalyzeColor ... 1225
3.55.4.4.2 ANA.GetBGColor ... 1239
3.55.4.4.3 ANA.GetColorCount ... 1242
3.55.4.4.4 ANA.Reduce_Colors ... 1246

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.4.5 ANA.RemoveIndexColors ... 1254
3.55.4.4.6 ANA.RemoveSortedColors ... 1259
3.55.4.4.7 ANA.ReplaceColorD ... 1268
3.55.4.4.8 ANA.ReplaceColorT ... 1275
3.55.4.4.9 ANA.Show...Array ... 1284
3.55.4.5 Color-Calculation ... 1297
3.55.4.5.1 ANA. Inverse_Color / Complementary_Color ... 1298
3.55.4.5.2 ANA.Add/Sub_Color / Mix_Colors ... 1307
3.55.4.5.3 ANA.Color_Diff-Commands ... 1315
3.55.4.5.4 ANA.GetGradientColor ... 1325
3.55.4.5.5 ANA.Split_RGB / Combine_RGB ... 1328
3.55.4.6 Convert Colors ... 1335
3.55.4.6.1 ANA.Brightness ... 1336
3.55.4.6.2 ANA.BW ... 1342
3.55.4.6.3 ANA.Contrast ... 1352
3.55.4.6.4 ANA.Grey ... 1358
3.55.4.6.5 ANA.Luminance / Change_Luminance ... 1366
3.55.4.6.6 ANA.TriColor ... 1377
3.55.4.7 Debug & Show IR's ... 1386
3.55.4.7.1 ANA.Show ... 1387
3.55.4.7.2 ANA.ShowHits ... 1390
3.55.4.8 Flip IR's ... 1394
3.55.4.8.1 ANA.HFlip ... 1395
3.55.4.8.2 ANA.HVFlip ... 1398
3.55.4.8.3 ANA.VFlip ... 1401
3.55.4.9 Load/Save-Import/Export IR's ... 1405
3.55.4.9.1 ANA.#FromVar ... 1406
3.55.4.9.2 ANA.FromClipboard ... 1410
3.55.4.9.3 ANA.Load ... 1412
3.55.4.9.4 ANA.Save ... 1415
3.55.4.9.5 ANA.Snap ... 1418
3.55.4.9.6 ANA.ToClipboard/Rect ... 1421
3.55.4.9.7 ANA.ToVar ... 1424
3.55.4.10 Logic-Operations ... 1426
3.55.4.10.1 ANA.ColorDiffMap ... 1429
3.55.4.10.2 ANA.Complement / ANA.Complementary_Color ... 1435
3.55.4.10.3 ANA.Invert / ANA.Inverse_Color ... 1442
3.55.4.10.4 ANA.Logic ... 1449
3.55.4.10.5 ANA.Mask ... 1455
3.55.4.11 Manage IR's ... 1473
3.55.4.11.1 ANA.Compare ... 1474
3.55.4.11.2 ANA.Copy ... 1477
3.55.4.11.3 ANA.Move ... 1479
3.55.4.11.4 ANA.New ... 1481
3.55.4.11.5 ANA.Swap ... 1484
3.55.4.12 Paint Operations ... 1486
3.55.4.12.1 ANA.DrawBox ... 1488
3.55.4.12.2 ANA.DrawLine ... 1493
3.55.4.12.3 ANA.Fill ... 1499

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.12.4 ANA.HGradient ... 1503
3.55.4.12.5 ANA.PatternFill ... 1508
3.55.4.12.6 ANA.PatternFillHwn ... 1511
3.55.4.12.7 ANA.PrintAt ... 1516
3.55.4.12.8 ANA.VGradient ... 1521
3.55.4.13 Pattern-Recognition ... 1526
3.55.4.13.1 ANA.CompareBlock ... 1527
3.55.4.13.2 ANA.DownScale ... 1530
3.55.4.13.3 ANA.FindPattern ... 1537
3.55.4.13.4 ANA.FindPattern2 ... 1546
3.55.4.13.5 ANA.FindPattern4 ... 1550
3.55.4.13.6 ANA.FindPattern5 ... 1555
3.55.4.13.7 ANA.FindPClick ... 1560
3.55.4.13.8 ANA.Get/Set_Data ... 1564
3.55.4.13.9 ANA.GetResult ... 1567
3.55.4.13.10 ANA.GetResultsPack ... 1571
3.55.4.13.11 ANA.ResetList ... 1575
3.55.4.13.12 ANA.ResetResultsPack ... 1577
3.55.4.13.13 ANA.SetMaxResult ... 1579
3.55.4.13.14 ANA.SnapFind1 ... 1581
3.55.4.13.15 ANA.SortResultsPack ... 1585
3.55.4.14 Pixel-Operation ... 1587
3.55.4.14.1 ANA.GetPixelColor ... 1619
3.55.4.14.2 ANA.GetPixelColor_4D ... 1629
3.55.4.14.3 ANA.GetPixelColor_4S ... 1634
3.55.4.14.4 ANA.GetPixelColor_5h ... 1639
3.55.4.14.5 ANA.GetPixelColor_5v ... 1645
3.55.4.14.6 ANA.GetPixelColor_8 ... 1650
3.55.4.14.7 ANA.GetPixelRGB ... 1655
3.55.4.14.8 ANA.SetPixelColor ... 1665
3.55.4.14.9 ANA.SetPixelColor_5h ... 1685
3.55.4.14.10 ANA.SetPixelRGB ... 1691
3.55.4.14.11 ICC./NCC. - If Color Condition ... 1700
3.55.4.14.12 RCC. - Replace If Color Condition ... 1706
3.55.4.14.13 RIC. - Replace Pixel If Color Condition ... 1719
3.55.4.14.14 SRG./GRG. - Set/Get Color Condition Register ... 1725
3.55.4.15 RGB- and HLS-Operations ... 1731
3.55.4.15.1 ANA.Change_HLSA / Change_HLSM ... 1734
3.55.4.15.2 ANA.HLS_RGB / RGB_HLS ... 1741
3.55.4.15.3 ANA.HLS_To_RGB / RGB_To_HLS ... 1745
3.55.4.15.4 ANA.MixNG/NR/NB ... 1750
3.55.4.15.5 ANA.MixRGB/BGR/BRG/GRB ... 1756
3.55.4.15.6 ANA.MixROR/ROL/DROR/DROL ... 1765
3.55.4.15.7 ANA.SplitRGB ... 1780
3.55.5 GRA. - Picture/Image Commands ... 1784
3.55.5.1 GRA.GetResolution ... 1784
3.55.5.2 GRA.Scale ... 1787
3.55.5.3 GRA.Convert ... 1792
3.55.5.4 GRA.vToBMP ... 1797

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6 Non-IR Operations 1800
3.55.6.1 GPC. - Get-Pixel-Colour 1801
3.55.6.2 ILC. / NLC. - If Line Color 1805
3.55.6.3 IPC. / NPC. - If-Pixel-Colour 1816
3.55.6.4 IRA. / NRA. - If Rect altered 1824
3.55.6.5 IRC. / NRC, - If Rect Color 1828
3.55.6.6 RRA. - Remember Rectangular Area 1836
3.55.6.7 SPC. - Set-Pixel-Colour 1843
3.55.6.8 WLC. - Wait for Line Color 1846
3.55.6.9 WRA. - Wait Rect-Area Altered 1851
3.55.6.10 WRC. - Wait Rect Color 1856
3.55.7 Pattern-Search 1861
3.55.7.1 ! How to record Patterns 1862
3.55.7.2 GFP. - Get Find Parameter 1869
3.55.7.3 IFP. / NFP. - If Find Pattern 1873
3.55.7.4 ITP. / NTP. - If Find any of Two Pattern 1877
3.55.7.5 JFP. / JNP. - Jump Find Pattern 1881
3.55.7.6 JTP. / JTN. - Jump Two Pattern 1885
3.55.7.7 LFP. - Locate Find Pattern 1890
3.55.7.8 LTP. - Locate Two Pattern 1894
3.55.7.9 SFP. - Set Find Parameter 1899
3.55.7.10 WFP. - Wait For Pattern 1903
3.55.7.11 WTP. - Wait For Two Pattern 1907
3.55.8 Undercover and Graphics 1911
3.55.8.1 ! Smarty's Undercover Operations 1911
3.55.8.2 ! UCV. - Undercover Graphics 1914
3.55.8.3 bgr - background 1921
3.55.8.4 box 1925
3.55.8.5 cov - cover 1929
3.55.8.6 dbw - desktop bw 1933
3.55.8.7 dc - desktop-copy 1937
3.55.8.8 eli - ellipse 1941
3.55.8.9 fgr - foreground 1945
3.55.8.10 glas - suppress mouse and keyboard 1949
3.55.8.11 grd - gradient 1953
3.55.8.12 ldi - load image 1958
3.55.8.13 lin - line / drawto 1962
3.55.8.14 mat - matte, dark transparent cover 1967
3.55.8.15 nb - no blend 1971
3.55.8.16 nbi - no blend-in 1974
3.55.8.17 nbo - no blend-out 1977
3.55.8.18 ntr - no transparency 1980
3.55.8.19 pri - print 1984
3.55.8.20 rsf - reset font 1987
3.55.8.21 sft - set font 1990
3.55.8.22 spi - plot - set pixel 1995
3.55.8.23 tra - transparency 2001
3.55.8.24 wri - write 2005
3.56 Input- and Output commands 2008
3.56.1 ! Smarty's Input and Output Commands 2008
3.56.2 CLP. - Clipboard - Operation 2009

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.56.3 DDD. - Display Directory Dialog 2012
3.56.4 DFD. - Display File Dialog 2015
3.56.5 DIT. - Dial Tone MFV 2018
3.56.6 DSD. - Display Save-File Dialog 2021
3.56.7 GUP. - Get User Password 2024
3.56.8 GUT. - Get User Text 2028
3.56.9 MBT. - MessageBox Timed 2031
3.56.10 MBX. - MessageBox 2036
3.56.11 MSB. - Message Beep 2040
3.56.12 OUT. - Send to Standard-Output 2043
3.56.13 PRT. PRinT 2045
3.56.14 SAY. - Say something 2048
3.56.15 SBT. - Show Bubble Text 2055
3.56.16 SHT. - Show Text 2061
3.57 Internet and Network 2066
3.57.1 ! Smarty's Internet and Network Commands 2066
3.57.2 CFU. - Copy-From-URL 2067
3.57.3 CNS. - Connect Net Share 2070
3.57.4 DLF. - DownLoad-File 2074
3.57.5 DLV. - Download-to-Variable 2076
3.57.6 FTP. - and more 2078
3.57.6.1 FTP. - File Transfer Protokoll 2078
3.57.6.2 fto - file transfer protocol 2080
3.57.6.3 tdl - Test Download 2085
3.57.6.4 tpw - test with password 2088
3.57.7 GCN. - Get Computer Name 2091
3.57.8 GIP. - Get IP Adress 2094
3.57.9 GNS. - Get Network Share 2098
3.57.10 GUD. - Get User and Domain 2101
3.57.11 GUN. - Get User Name 2103
3.57.12 GVL. - Get Volume List 2105
3.57.13 MND. - Map Network Drive 2108
3.57.14 RDM. - Remove Drive Mapping 2112
3.57.15 SID. - Security Identifier 2114
3.58 IPC, InterKOM and Exit Code 2116
3.58.1 ! Smarty's InterKOM Technology 2116
3.58.2 GIK. - Get InterKOM 2122
3.58.3 IIK. - If InterCOM Condition 2125
3.58.4 SEC. - Set Exit Code 2129
3.58.5 SIK. - Set InterKOM 2131
3.58.6 SRV. - Set Return Value (Executable) 2134
3.58.7 TWM. - Text-Window-Message 2136
3.58.8 WFM. - Wait-For-Message 2140
3.58.9 WIK. - Wait InterKOM 2142

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.59 IXX. - Choose one of two cases 2146
3.59.1 ! Smarty's IXX-Commands 2146
3.59.2 EIF. - END IF 2148
3.59.3 ELS. - ELSE 2150
3.59.4 IFD. / NFD. - If File Date 2152
3.59.5 IFL. / NFL. - If File Language 2155
3.59.6 IFV. / NFV. If File Version 2161
3.59.7 ILI. / NLI. - If LanguageID 2163
3.60 JXX. - Jump - Instructions 2168
3.60.1 ! Smarty's JXX-Commands 2168
3.60.2 ENR. - End Return 2169
3.60.3 JIS. - Jump If String 2171
3.60.4 JIT. - Jump if TimeOut 2174
3.60.5 JIV. - Jump-If-Variable 2177
3.60.6 JIZ. - Jump-If-Zero 2180
3.60.7 JLE. - Jump on Less or Equal 2183
3.60.8 JME. - Jump on More or Equal 2186
3.60.9 JMP./GTO. - Jump to Label 2189
3.60.10 JNF. - Jump New File 2191
3.60.11 JNJ. - JumpNewFile-Jump to Label 2194
3.60.12 JNR. - Jump if no Result 2197
3.60.13 JNS. - Jump If No String 2200
3.60.14 JNT. - Jump If-No TimeOut 2203
3.60.15 JNZ. - Jump-if-not-zero 2206
3.60.16 JOR. - Jump On Repeat-Count 2209
3.60.17 JRR. - Jump and Reset Return-Stack 2212
3.60.18 JRS. - Jump if Result 2215
3.60.19 JSR./GSB. - Jump SubRoutine (Label) 2218
3.60.20 RET. - RETurn 2224
3.60.21 RRC. - Reset Repeat-Counter 2227
3.60.22 RRS. - Reset Return Stack 2230
3.60.23 RTS. - ReTurn from Subprogram 2233
3.61 Keyboard Typing Simulation 2235
3.61.1 ! Keyboard-Keycode-List 2235
3.61.2 GEK. - Get Extended Key-State 2240
3.61.3 GKS. - Get Key-State 2243
3.61.4 IEK. / NEK. - If Extended Key-State 2246
3.61.5 IKS. / NKS. - If Key-State 2249
3.61.6 KBC. / KBI. - KeyBoard Command 2252
3.61.7 KYD. - Key Down 2256
3.61.8 KYU. - Key Up 2260
3.61.9 RST. - Replace Selected Text 2264
3.61.10 SDA. / SDU. - Send Input ANSI/UNICODE 2267
3.61.11 SDH. - Send Keys Hexcode 2271

 *© 2023 Theo Gottwald * https://www.smart-package.com*

3.61.12 SDK. - SendKeys VB 2275
3.61.13 SDT. - SenD-window-Text 2278
3.61.14 SEK. - Set Extended Key-State 2282
3.61.15 SKI. - Send Key-Input 2285
3.61.16 SKP. Send KeyPress 2289
3.61.17 SKT. - Send - Key-Text 2293
3.61.18 WNK. - Wait-No-Key 2296
3.62 Loops, and repeated action 2298
3.62.1 ! Smarty's Loops 2298
3.62.2 DOL. - OOP. - Conditional Loop 2299
3.62.3 FOR. - NEX. Counting Loop 2302
3.62.4 NEX. - Bottom of FOR.-Loop 2308
3.62.5 OOP. - Bottom of DOL.-Loop 2312
3.63 Mathematical and algorithmic operations 2314
3.63.1 ! Smarty's mathematical commands 2314
3.63.2 CAL. - mathematical CALculation 2315
3.63.3 CAX. - Calculate Extended 2320
3.63.4 CST. - Coordinate System Transform 2331
3.63.5 FZC. - Fuzzy Compare 2334
3.63.6 GRM. - Get Rect Middle-Point 2337
3.63.7 IFC. - If Fuzzy Compare 2340
3.63.8 IVN. - If Variable Numeric 2343
3.63.9 RND. - RandDom Number 2348
3.63.10 VDC. - Variable Decrement 2350
3.63.11 VIC. - Variable Increment 2352
3.63.12 VTB. - Variable to Binary 2354
3.63.13 VTH. - Variable to Hexadecimal 2356
3.63.14 VTN. - Variable Numeric 2358
3.64 Miscellaneous commands 2361
3.64.1 ! Smarty's MISC-Operations 2361
3.64.2 END. - End Script 2362
3.64.3 UNI. - Universal Command 2364
3.64.4 IEM. / NEM. - If Execution-Mode 2402
3.64.5 ITO. / NTO. - If TimeOut 2404
3.64.6 OPT. - Optional Settings 2406
3.64.7 STF. - Set TimeOut Flag 2410
3.65 Mouse-Operations 2412
3.65.1 ! Smarty's Mouse-Operations 2412
3.65.2 CPC. - Convert Pixel Coordinate 2425
3.65.3 Automated Clicks 2431
3.65.3.1 BCS. - Button Click Special 2431
3.65.3.2 BSC. - Button Special Click 2435
3.65.4 Compatible Engine 2439
3.65.4.1 GMP. - Get-Mouse-Position 2439

 *© 2023 Theo Gottwald * https://www.smart-package.com*

3.65.4.2 MDC. - Mouse Double-left Click (Invinsible) ... 2442
3.65.4.3 MDI. - Mouse Double-Click Invisible ... 2445
3.65.4.4 MLC. - Mouse-Left-Click (Invisible) ... 2448
3.65.4.5 MLD. - Mouse left down ... 2451
3.65.4.6 MLI. - Mouse Left Click quIckly (Invinsible) ... 2454
3.65.4.7 MLM. - Mouse Left Click Move (Invinsible) ... 2457
3.65.4.8 MRM. - Mouse Right Click Move (Invinsible)_2 ... 2460
3.65.4.9 MLU. - Mouse left up ... 2463
3.65.4.10 MMV. - Mouse MoVe ... 2466
3.65.4.11 MPO. - MousePosition ... 2469
3.65.4.12 MRC. - Mouse-Right-Click ... 2472
3.65.4.13 MRI. - Mouse Right Click Invisible ... 2476
3.65.4.14 MWR. - Mouse Wheel Roll ... 2480
3.65.5 Direct Engine ... 2483
3.65.5.1 MAP. - Move Active Point ... 2483
3.65.5.2 MDE. - Mouse Double Click Event ... 2487
3.65.5.3 MDM. - Mouse Down Message ... 2490
3.65.5.4 MLE. - Mouse-Left-Click Event ... 2492
3.65.5.5 MRE. - Mouse-Right-Click Event ... 2495
3.65.5.6 MUM. - Mouse Up Message ... 2498
3.65.5.7 MWE. - Mouse Wheel Event ... 2500
3.65.6 IMO. / NMO. - If Mouse Option ... 2503
3.65.7 LBO. - Left Border ... 2506
3.65.8 MOS. - Mouseless Object Select ... 2509
3.65.9 MWC. - Mouse Window Click ... 2514
3.65.10 MWT. - Mouse Wheel Tile ... 2516
3.65.11 RBO. - Right Border ... 2519
3.66 Multimedia Commands ... 2522
3.66.1 ! Smarty's Multimedia Commands ... 2522
3.66.2 MPA. - Music Player ... 2523
3.66.3 MPB. - Music Play Background ... 2526
3.66.4 MPC. - Music Player Command ... 2530
3.66.5 MPS. - Music Play Speaker ... 2536
3.66.6 SVL. - Set Volume ... 2539
3.67 NT Service commands ... 2544
3.67.1 !Service Commands ... 2544
3.67.2 FES. - For Each Service ... 2550
3.67.3 ISC. - If Servic Condition ... 2552
3.67.4 ISI. / NSI. - If Service Installed ... 2557
3.67.5 SER. - Service Command ... 2559
3.67.6 WSC. - Wait Service Command ... 2566
3.68 OS and Software ... 2570
3.68.1 ! Smarty's OS & Software Ops ... 2570
3.68.2 ADV. - Get Advanced Operator Privilege ... 2572
3.68.3 CPL. - Control-Panel ... 2574
3.68.4 GEN. - Get-EXE-Number ... 2580
3.68.5 GEP. - Get-Executable-Path ... 2582

 *© 2023 Theo Gottwald * https://www.smart-package.com*

3.68.6 GII. - Get-Installation-Information ... 2584
3.68.7 GLI. - Get Language ID ... 2587
3.68.8 GOS. - Get OS Information ... 2591
3.68.9 GSD. - Get Session Data ... 2595
3.68.10 GUS. - Get Usage State ... 2598
3.68.11 IAI. / NAI. - If-Application-Installed ... 2601
3.68.12 IOS. / NOS. - If-Operating System ... 2607
3.68.13 IUA. / NUA. - If User is Administrator ... 2612
3.68.14 LIB. - Library ... 2614
3.68.15 Systray-Icon Commands ... 2617
3.68.15.1 SYT.Add Icon To ... 2618
3.68.15.2 SYT.Add Own Icon ... 2620
3.68.15.3 SYT.Change Any Icon ... 2622
3.68.15.4 SYT.Change Own Icon ... 2624
3.68.15.5 SYT.Has Icon ... 2626
3.68.15.6 SYT.Own Tooltip ... 2628
3.68.15.7 SYT.Tooltip ... 2630
3.69 Parallel-Robot Operations ... 2632
3.69.1 ! Smarty's Parallel Robot Operations ... 2632
3.69.2 PRE. - Parallelrobot-Run-End ... 2638
3.69.3 PRR. - Parallel-Robot-Run ... 2642
3.69.4 PRS. - Parallel Robot Set Resukt ... 2647
3.69.5 PRV. - Parallel-Robot send Variable ... 2651
3.69.6 WFV. - Wait-For-VariableChange ... 2655
3.70 PDC. - Private Desktop Commands ... 2658
3.70.1 !PDC. - Private Desktop Command ... 2658
3.70.2 HDC. - Hidden Desktop Control ... 2664
3.70.3 ICD./NCD. - If/Not Current Desktop ... 2666
3.70.4 IPD./NPD. - If/Not Private Desktop ... 2669
3.70.5 PDC.Close Desktop ... 2671
3.70.6 PDC.Create Desktop no Explorer ... 2674
3.70.7 PDC.Create Desktop with Explorer ... 2677
3.70.8 PDC.Get current Desktop Name ... 2679
3.70.9 PDC.Goto Default ... 2682
3.70.10 PDC.Run on Desktop WFI/WFO ... 2684
3.70.11 PDC.Set User / Reset User ... 2688
3.70.12 PDC.Switch to Desktop ... 2691
3.71 PlugIn-Commands ... 2693
3.71.1 ! Smarty's PlugIn's ... 2693
3.71.2 PIL. - PlugIn Launch ... 2702
3.71.3 PIW. - PlugInWait ... 2706
3.71.4 PIS. - PlugIn Send ... 2710
3.71.4.1 ! Buttonbar for Blender-Sample Code ... 2710
3.71.4.2 ! Buttonbar for HM8-Sample Code ... 2733
3.71.4.3 ! PIS. - PlugIn Send ... 2749

3.71.4.4 PIS.attach/detach ... 2755
3.71.4.5 PIS.ButtonEnable / ButtonDisable ... 2763
3.71.4.6 PIS.GetHandle ... 2766
3.71.4.7 PIS.GetMouseButton ... 2777
3.71.4.8 PIS.GetMouseButtonHandle ... 2787
3.71.4.9 PIS.MouseMessageFilter ... 2797
3.71.4.10 PIS.SetAutoHide ... 2800
3.71.4.11 PIS.SetAutoRunSynchron / SetAutoRunASynchron ... 2802
3.71.4.12 PIS.SetAutoRunSynchronNoDisable / SetAutoRunASynchronNoDisable ... 2804
3.71.4.13 PIS.SetButtonForeColor / SetButtonBackColor ... 2807
3.71.4.14 PIS.SetButtonNumber ... 2810
3.71.4.15 PIS.SetButtonsHDist / SetButtonsVDist ... 2812
3.71.4.16 PIS.SetButtonsHVSize ... 2815
3.71.4.17 PIS.SetButtonText ... 2818
3.71.4.18 PIS.SetcorrectionFactor ... 2823
3.71.4.19 PIS.SetPicture ... 2830
3.71.4.20 PIS.SetSetHorizontal / SetVertical ... 2833
3.71.4.21 PIS.SetTopMost ... 2836
3.71.4.22 PIS.SetTransparent ... 2847
3.71.4.23 PIS.SetWindowAlert ... 2850
3.71.4.24 PIS.Show/Hide ... 2860
3.72 PopUp - Handling ... 2863
3.72.1 DBC. - Do-Button-Clicks ... 2863
3.73 Powershell Inline ... 2866
3.73.1 !Smarty's Powershell Inline ... 2866
3.73.2 PWS. - Powershell-Script Start ... 2874
3.73.3 PWE. - Powershell-Script End ... 2877
3.74 Process Operations ... 2881
3.74.1 ! Smarty's Process Commands ... 2881
3.74.2 CPR. - Close-PRocess ... 2882
3.74.3 CTR. - Close-ThRead ... 2884
3.74.4 FPI. - Filepath to PID ... 2886
3.74.5 GDF. - Get-Data-From ... 2889
3.74.6 GPI. - Get Process Information ... 2893
3.74.7 GPU. - Get Process User ... 2898
3.74.8 IEP. / NEP. - If-Exists-Process ... 2900
3.74.9 KPN. - Kill Process by Name ... 2904
3.74.10 MRK. - MiniRobotKill ... 2907
3.74.11 WII. - Wait-for-Input-Idle ... 2909
3.74.12 WPR. - Wait-Process-Running ... 2911
3.74.13 WPT. - Wait-Process-Termination ... 2913
3.75 Registry-commands ... 2915
3.75.1 ! Smarty's Registry Commands ... 2915
3.75.2 DID. - Decode (MS-)Product-ID ... 2916
3.75.3 IER. / NER. - If-Exist-Registry ... 2918
3.75.4 RDI. - Registry-Delete-Item ... 2921
3.75.5 RDT. - Registry-DelTree ... 2923

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.75.6 RER. - Registry Entry Read 2925
3.75.7 REW. - Registry-Entry-Write 2928
3.76 RES. - EXE-Resourcen managen 2931
3.76.1 ! Resourcen Commands 2931
3.76.2 RES. - Manage Resources 2936
3.76.3 RES.Change Icon 2938
3.76.4 RES.Change Manifest 2941
3.76.5 RES.Change Version 2944
3.76.6 RES.Refresh Explorer 2948
3.76.7 RES.Extract All 2950
3.76.8 RES.Extract Icon 2955
3.76.9 RES.Extract Manifest 2958
3.76.10 RES.Extract Version 2964
3.76.11 RES.Last Manifest 2969
3.76.12 RES.Last Version 2973
3.76.13 RES.Last Icon 2978
3.77 SCS. - Select from Mutiple Cases 2982
3.77.1 ! Select from multiple choices 2982
3.77.2 1. SCS. - Select Case 2983
3.77.3 2. CAS. - Case-String 2987
3.77.4 3. CAN. - Case-Numeric 2990
3.77.5 4. CFP. - Case-Pattern 2993
3.77.6 5. CAE. - Case-Else 2996
3.77.7 6. ESC. - End-Select-Case 2999
3.77.8 7. ESL. - End Select Loop 3002
3.78 Stack-Operations 3005
3.78.1 ! Smarty's Stack Commands 3005
3.78.2 POP. - POP Variable from Stack 3007
3.78.3 POV. - POp Variables from User-Stack 3010
3.78.4 PUS. - Push Parameter onto Stack/Que 3013
3.78.5 PVS. - Push Variables on Stack/Que 3015
3.78.6 PUV. PUsh variables on the User-Stack 3019
3.78.7 PNS. Push Numeric variables on the User-Stack 3022
3.78.8 PSS. Push String variables on the User-Stack 3026
3.78.9 STJ. - STack Job 3030
3.78.10 STS. - Set STack Settings 3038
3.78.11 STV. - Stack To/from Variable 3040
3.79 String commands 3042
3.79.1 ! Smarty's String Operations 3042
3.79.2 ACH. - After Character 3043
3.79.3 ALB. - After last Backslah 3045
3.79.4 ASC. - Get ASCII - Code 3047
3.79.5 BCH. - Before Character 3049
3.79.6 BLB. - Before last Backslash 3051

 *© 2023 Theo Gottwald * https://www.smart-package.com*

3.79.7 Change case (upper/ lowercase) ... 3053
3.79.7.1 VTL. - Variable to Lowercase ... 3053
3.79.7.2 VTM. - Variable to Mixedcase ... 3056
3.79.7.3 VTU. - Variable to Uppercase ... 3059
3.79.8 CHR. - Get CHaRacter ... 3062
3.79.9 CSP. - Clip String Part ... 3064
3.79.10 Delimited Strings ... 3067
3.79.10.1 GES. - Get-Extracted-String ... 3067
3.79.10.2 SBD. - String between Delimiter ... 3070
3.79.10.3 SBM. - String between Delimiter-Multiple ... 3073
3.79.10.4 VRD. - Variable-Remove-Doubles ... 3076
3.79.11 DIC.-Delimited Item Commands ... 3078
3.79.11.1 DIC.Eval_Column_List ... 3080
3.79.11.2 DIC.Format_Numeric ... 3082
3.79.11.3 DIC.Get_Field_Index ... 3084
3.79.11.4 DIC.Get_Fields ... 3086
3.79.11.5 DIC.Get_in_Order ... 3088
3.79.11.6 DIC.Insert ... 3090
3.79.11.7 DIC.Letter_to_Number ... 3092
3.79.11.8 DIC.Number_to_Letter ... 3094
3.79.11.9 DIC.Remove ... 3096
3.79.11.10 DIC.Set_Multiple ... 3098
3.79.11.11 DIC.Set_Value ... 3101
3.79.11.12 DIC.Swap ... 3104
3.79.12 Format Strings ... 3107
3.79.12.1 FMT. - Format Numeric Output ... 3107
3.79.12.2 VSL. - Variable Set Left ... 3113
3.79.12.3 VSM. - Variable Set Middle ... 3116
3.79.12.4 VSR. - Variable Set Right ... 3119
3.79.13 GFS. - Get-Filtered-String ... 3122
3.79.14 GRE. - Get Regular Expression ... 3128
3.79.15 GRR. - Get Regular Replace ... 3131
3.79.16 GSC. - Get String Count ... 3134
3.79.17 GSS. - GetSplitString ... 3137
3.79.18 LEN. - Length-of-String ... 3140
3.79.19 RPL. - RePLace in String ... 3143
3.79.20 SRT. - Sort Items ... 3147
3.79.21 STR.- String Command ... 3150
3.79.21.1 ! STR. - String Command ... 3150
3.79.21.2 STR.CHARS A ... 3155
3.79.21.3 STR.CHARS B ... 3159
3.79.21.4 STR.CHARS C ... 3163
3.79.21.5 STR.CHARS D ... 3167
3.79.21.6 STR.CLONE ... 3171
3.79.21.7 STR.DELETE ... 3174
3.79.21.8 STR.FIND ... 3176
3.79.21.9 STR.FIND ANY ... 3179
3.79.21.10 STR.FROM BASE64 URL ... 3182
3.79.21.11 STR.FROM HEXASCI ... 3185

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.79.21.12 STR.GENERATE ... 3188
3.79.21.13 STR.INSERT ... 3192
3.79.21.14 STR.MAKE PAD ... 3195
3.79.21.15 STR.MIME TO TEXT ... 3198
3.79.21.16 STR.NINSTR ... 3202
3.79.21.17 STR.OVERWRITE ... 3205
3.79.21.18 STR.PATCH ... 3208
3.79.21.19 STR.REVERSE ... 3212
3.79.21.20 STR.TEXT TO MIME ... 3215
3.79.21.21 STR.TO BASE64 URL ... 3219
3.79.21.22 STR.TO HEXASCI ... 3222
3.79.22 Trim String ... 3225
3.79.22.1 VBT. - Variable-Both-sides-Trim ... 3225
3.79.22.2 VLT. - Variable-Left-Trim ... 3228
3.79.22.3 VRT. - Variable-Right-Trim ... 3231
3.79.23 URL. - Split URL ... 3234
3.79.24 Unicode-Conversion ... 3237
3.79.24.1 ATU. - ANSI to Unicode ... 3237
3.79.24.2 UTA. - Unicode to ANSI ... 3239
3.80 TXT. Text-File Engine ... 3241
3.80.1 !Sample-Code: Indentation Checker ... 3247
3.80.2 !TXT. - Text File Engine ... 3252
3.80.3 Conversions ... 3257
3.80.3.1 Convert_to_ANSI ... 3258
3.80.3.2 Convert_to_Unicode ... 3261
3.80.3.3 CR_to_CRLF ... 3264
3.80.3.4 CRLF_to_CR ... 3267
3.80.3.5 CRLF_to_LF ... 3270
3.80.3.6 LF_to_CRLF ... 3273
3.80.3.7 Translate_Chars ... 3275
3.80.4 CSV-Related Commands ... 3278
3.80.4.1 Delimited_Items_Insert ... 3280
3.80.4.2 Delimited_Items_Remove ... 3283
3.80.4.3 Delimited_Items_Swap ... 3286
3.80.4.4 Filter_Lines ... 3289
3.80.4.5 Find_first_in_Column ... 3293
3.80.4.6 Find_last_in_Column ... 3297
3.80.4.7 Get_any_Delimited_Item_from_Line ... 3301
3.80.4.8 Get_any_Delimited_Item_from_Text ... 3303
3.80.4.9 Get_Column_Block ... 3305
3.80.4.10 Get_Delimited_Item_from_Line ... 3309
3.80.4.11 Get_Delimited_Item_from_Text ... 3311
3.80.4.12 Keep_Columns ... 3313
3.80.4.13 Repair_CSV ... 3316
3.80.4.14 Sort_by_Column_Float ... 3318
3.80.4.15 Sort_by_Column_Integer ... 3322
3.80.4.16 Sort_by_Column_Letter ... 3326
3.80.5 Find- and Replace Operations ... 3330
3.80.5.1 Find_Line_with ... 3331
3.80.5.2 Find_Text ... 3334

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.80.5.3 Replace ... 3338
3.80.5.4 Replace_in_Line ... 3341
3.80.5.5 Replace_in_Text ... 3344
3.80.6 Get Lines and Text ... 3347
3.80.6.1 Get_after_Line ... 3351
3.80.6.2 Get_before_Line ... 3353
3.80.6.3 Get_Line ... 3355
3.80.6.4 Get_Line_Count ... 3357
3.80.6.5 Get_Line_lefttrim ... 3359
3.80.6.6 Get_Line_Position ... 3361
3.80.6.7 Get_Line_trimmed ... 3363
3.80.6.8 Get_Text_Block ... 3365
3.80.6.9 Get_Text_Lines ... 3367
3.80.6.10 Get_To_End ... 3369
3.80.6.11 Get_To_Start ... 3371
3.80.6.12 Line_Start_End_by_Position ... 3373
3.80.6.13 Linenumber_from_Position ... 3376
3.80.7 Insert/Delete Text ... 3378
3.80.7.1 Delete_Characters ... 3379
3.80.7.2 Delete_Line ... 3381
3.80.7.3 Delete_Lines ... 3383
3.80.7.4 Insert_Line_after_Line ... 3385
3.80.7.5 Insert_Line_at_End ... 3387
3.80.7.6 Insert_Line_at_Pos ... 3390
3.80.7.7 Insert_Line_at_Start ... 3392
3.80.7.8 Insert_Line_before_Line ... 3395
3.80.7.9 Insert_Text_after_Line ... 3397
3.80.7.10 Insert_Text_at_End ... 3400
3.80.7.11 Insert_Text_at_Pos ... 3403
3.80.7.12 Insert_Text_at_Start ... 3406
3.80.7.13 Insert_Text_before_Line ... 3409
3.80.7.14 Set_Line_to ... 3411
3.80.7.15 Set_Text_Lines ... 3413
3.80.8 Load / Save Commands ... 3415
3.80.8.1 Clear ... 3416
3.80.8.2 Load Text File ... 3418
3.80.8.3 Load Unicode_File ... 3420
3.80.8.4 Save_as_Unicode ... 3422
3.80.8.5 Save_File ... 3424
3.80.9 Push / Pop ... 3427
3.80.9.1 Append_from_Internal ... 3428
3.80.9.2 Append_to_Internal ... 3431
3.80.9.3 Clear_Intern ... 3433
3.80.9.4 Pop_Intern ... 3435
3.80.9.5 Pop_Keep ... 3437
3.80.9.6 Prepend_from_Internal ... 3439
3.80.9.7 Prepend_to_Internal ... 3442
3.80.9.8 Push_Intern ... 3444
3.80.9.9 Push_Keep ... 3446
3.80.10 To LOF / To Var ... 3448
3.80.10.1 From_Var ... 3449

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.80.10.2 Get_from_LOF ... 3452
3.80.10.3 Get_pure_Text ... 3455
3.80.10.4 Load_pure_Text ... 3457
3.80.10.5 To_LOF ... 3459
3.80.10.6 To_Var ... 3462
3.80.10.7 To_Var_and_Clear ... 3464
3.80.11 Other Commands ... 3466
3.80.11.1 DebugPrint ... 3467
3.80.11.2 Get_Filename ... 3470
3.80.11.3 Get_Lenght ... 3472
3.81 Variable Definitions and Operations ... 3474
3.81.1 ! Smarty's Variable Commands ... 3474
3.81.2 GVT. - Get Variable Type ... 3477
3.81.3 IEV. / NEV. - If exists Variable ... 3481
3.81.4 Inline Data Commands ... 3483
3.81.4.1 Inline-Data Sample ... 3488
3.81.4.2 VDA. / VDE. - Inline Data ... 3516
3.81.5 IVB. / NVB. - If Variable Between ... 3521
3.81.6 IVC. / NVC. - If-Variable-Contains ... 3524
3.81.7 IVP./ NVP. - If-Variable-Pattern ... 3529
3.81.8 IVS. / NVS. - If-Variable-String ... 3531
3.81.9 IVV. / NVV. - If Variable Value ... 3534
3.81.10 IWC. / NWC. - If Wildcard Pattern ... 3537
3.81.11 SAV. - Save/Restore Variable ... 3539
3.81.12 VAB. - Variable Binary ... 3544
3.81.13 VAC. - Variable reCt ... 3546
3.81.14 VAF. - Variable Filename ... 3548
3.81.15 VAI. - Variable Indirekt ... 3550
3.81.16 VAN. - Variable Numeric ... 3553
3.81.17 VAO. - Variable Once / Variable Clear ... 3556
3.81.18 VAP. - Variable Point ... 3560
3.81.19 VAR. - Variable Set Value/Clear ... 3562
3.81.20 VAS. - Variable with String ... 3564
3.81.21 VAV. - Variables with Variables ... 3566
3.81.22 VBA. - Variable Binary Append ... 3569
3.82 VB-Script ... 3572
3.82.1 !Smarty's VB-Script Support ... 3572
3.82.2 VBE. - VB-Script End ... 3578
3.82.3 VBS. - VB-Script Start ... 3581
3.83 Window Operations ... 3584
3.83.1 ! Smarty's Window Operations ... 3584
3.83.2 CAW. - Close-All-Windows ... 3586
3.83.3 CFW. - Copy-From-Window ... 3589
3.83.4 CLW. - Close-Window ... 3592
3.83.5 DTI.-Desktop & Mouse Commands ... 3594

3.83.5.1 DTI.Count Desktops ... 3595
3.83.5.2 DTI.DCoord ... 3597
3.83.5.3 DTI.Depth ... 3599
3.83.5.4 DTI.Frequency ... 3601
3.83.5.5 DTI.Height ... 3603
3.83.5.6 DTI.MousePos ... 3605
3.83.5.7 DTI.Name ... 3607
3.83.5.8 DTI.ScaleFactor ... 3609
3.83.5.9 DTI.ScalePos ... 3611
3.83.5.10 DTI.Size ... 3613
3.83.5.11 DTI.UnScalePos ... 3615
3.83.5.12 DTI.Width ... 3617
3.83.6 GAR. - Get actual Rectangle ... 3619
3.83.7 GCT. - Get-Childs-TopWindows ... 3622
3.83.8 GFW. - Get Foreground Window ... 3624
3.83.9 GRW. - Get Related Window ... 3626
3.83.10 GSW. - Get Special Window ... 3630
3.83.11 GWP. - Get Window Position/Size ... 3633
3.83.12 HTV. - Handle-to-Variable ... 3636
3.83.13 HWI. - Hide Window ... 3638
3.83.14 ICW. - If-Child-Window ... 3640
3.83.14.1 ! ICW. - If-Child-Window ... 3640
3.83.15 IEW. / NEW. - If Exist Window ... 3643
3.83.16 ITW. - If Top Window ... 3646
3.83.16.1 ! ITW. / NTW. -If Top-Window ... 3646
3.83.16.2 "A" - Flag - ITW. ... 3651
3.83.16.3 "B" - Flag - ITW. ... 3653
3.83.16.4 "C" - Flag - ITW. ... 3655
3.83.16.5 "D" - Flag - ITW. ... 3656
3.83.16.6 "E" - Flag - ITW. ... 3657
3.83.16.7 "F" - Flag - ITW. ... 3659
3.83.16.8 "G" - Flag - ITW. ... 3661
3.83.16.9 "H" - Flag - ITW. ... 3663
3.83.16.10 "I" - Flag - ITW. ... 3665
3.83.16.11 "J" - Flag - ITW. ... 3667
3.83.16.12 "K" - Flag - ITW. ... 3669
3.83.16.13 "M" - Flag - ITW. ... 3671
3.83.16.14 "N" - Flag - ITW. ... 3673
3.83.16.15 "Q" - Flag - ITW. ... 3675
3.83.16.16 "V" - Flag - ITW. ... 3677
3.83.16.17 "W" - Flag - ITW. ... 3679
3.83.16.18 "X" - Flag - ITW. ... 3681
3.83.16.19 "a" - Flag - ITW. ... 3683
3.83.16.20 "b" - Flag - ITW. ... 3685
3.83.16.21 "c" - Flag - ITW. ... 3687
3.83.16.22 "e" - Flag - ITW. ... 3689
3.83.16.23 "f" - Flag - ITW. ... 3691
3.83.16.24 "g" - Flag - ITW. ... 3693
3.83.16.25 "h" - Flag - ITW. ... 3695
3.83.16.26 "j" - Flag - ITW. ... 3697

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.27 "k" - Flag - ITW. 3699
3.83.16.28 "o" - Flag - ITW. 3701
3.83.16.29 "p" - Flag - ITW. 3703
3.83.16.30 "q" - Flag - ITW. 3705
3.83.16.31 "r" - Flag - ITW. 3707
3.83.16.32 "t" - Flag - ITW. 3709
3.83.16.33 "u" - Flag - ITW. 3712
3.83.16.34 "v" - Flag - ITW. 3714
3.83.16.35 "w" - Flag - ITW. 3716
3.83.17 IWE. / NWE. - If Window Enabled 3718
3.83.18 IWI. / NWI. - If Window Iconic 3721
3.83.19 IWS. / NWS. - If Window State 3724
3.83.20 IWV. / NWV. - If Window Visible 3727
3.83.21 IWZ. / NWZ. - If Window Zoomed 3730
3.83.22 MAW. - Missing-Active-Window 3733
3.83.23 SAW. - Set Active Window 3736
3.83.24 SCW. - Search-Child-Window 3738
3.83.24.1 ! SCW. - Search-Child-Window 3738
3.83.24.2 "E" - Flag - SCW. 3741
3.83.24.3 "I" - Flag - SCW. 3743
3.83.24.4 "K" - Flag - SCW. 3745
3.83.24.5 "Q" - Flag - SCW. 3748
3.83.24.6 "S" - Flag - SCW. 3750
3.83.24.7 "T" - Flag - SCW. 3752
3.83.24.8 "V" - Flag - SCW. 3754
3.83.24.9 "W" - Flag - SCW. 3756
3.83.24.10 "X" - Flag - SCW. 3758
3.83.24.11 "a" - Flag - SCW. 3761
3.83.24.12 "b" - Flag - SCW. 3764
3.83.24.13 "c" - Flag - SCW. 3767
3.83.24.14 "d" - Flag - SCW. 3769
3.83.24.15 "e" - Flag - SCW. 3771
3.83.24.16 "f" - Flag - SCW. 3774
3.83.24.17 "g" - Flag - SCW. 3777
3.83.24.18 "h" - Flag - SCW. 3779
3.83.24.19 "i" - Flag - SCW. 3782
3.83.24.20 "j" - Flag - SCW. 3785
3.83.24.21 "k" - Flag - SCW. 3787
3.83.24.22 "n" - Flag - SCW. 3789
3.83.24.23 "o" - Flag - SCW. 3791
3.83.24.24 "p" - Flag - SCW. 3793
3.83.24.25 "q" - Flag - SCW. 3795
3.83.24.26 "r" - Flag - SCW. 3797
3.83.24.27 "s" - Flag - SCW. 3799
3.83.24.28 "t" - Flag - SCW. 3801
3.83.24.29 "u" - Flag - SCW. 3804
3.83.24.30 "v" - Flag - SCW. 3806
3.83.24.31 "w" - Flag - SCW. 3809
3.83.25 SFW. - Set Foreground WIndow 3812
3.83.26 STW. - Search Top Window 3814

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.26.1 ! STW. -Search Top-Window ... 3814
3.83.26.2 "A" - Flag - STW. ... 3819
3.83.26.3 "B" - Flag - STW. ... 3821
3.83.26.4 "C" - Flag - STW. ... 3823
3.83.26.5 "D" - Flag - STW. ... 3824
3.83.26.6 "E" - Flag - STW. ... 3825
3.83.26.7 "F" - Flag - STW. ... 3827
3.83.26.8 "G" - Flag - STW. ... 3829
3.83.26.9 "H" - Flag - STW. ... 3830
3.83.26.10 "I" - Flag - STW. ... 3832
3.83.26.11 "J" - Flag - STW. ... 3834
3.83.26.12 "K" - Flag - STW. ... 3836
3.83.26.13 "M" - Flag - STW. ... 3838
3.83.26.14 "N" - Flag - STW. ... 3840
3.83.26.15 "Q" - Flag - STW. ... 3842
3.83.26.16 "V" - Flag - STW. ... 3844
3.83.26.17 "W" - Flag - STW. ... 3846
3.83.26.18 "X" - Flag - STW. ... 3848
3.83.26.19 "a" - Flag - STW. ... 3850
3.83.26.20 "b" - Flag - STW. ... 3852
3.83.26.21 "c" - Flag - STW. ... 3854
3.83.26.22 "e" - Flag - STW. ... 3855
3.83.26.23 "f" - Flag - STW. ... 3857
3.83.26.24 "g" - Flag - STW. ... 3859
3.83.26.25 "h" - Flag - STW. ... 3861
3.83.26.26 "j" - Flag - STW. ... 3863
3.83.26.27 "k" - Flag - STW. ... 3865
3.83.26.28 "o" - Flag - STW. ... 3867
3.83.26.29 "p" - Flag - STW. ... 3869
3.83.26.30 "q" - Flag - STW. ... 3871
3.83.26.31 "r" - Flag - STW. ... 3873
3.83.26.32 "t" - Flag - STW. ... 3875
3.83.26.33 "u" - Flag - STW. ... 3878
3.83.26.34 "v" - Flag - STW. ... 3880
3.83.26.35 "w" - Flag - STW. ... 3882
3.83.27 SWP. - Set Window Position and Size ... 3884
3.83.28 SWS. - Set Window State ... 3889
3.83.29 WCW. - Wait for new Child-Window ... 3893
3.83.29.1 ! WCW. - Wait for new Child-Window ... 3893
3.83.30 WTW. - Wait new Top Window ... 3896
3.83.30.1 ! WTW. -Wait New Top-Window ... 3896
3.83.30.2 "a" - Flag - WTW. ... 3901
3.83.30.3 "A" - Flag - WTW. ... 3903
3.83.30.4 "b" - Flag - WTW. ... 3905
3.83.30.5 "B" - Flag - WTW. ... 3907
3.83.30.6 "C" - Flag - WTW. ... 3909
3.83.30.7 "c" - Flag - WTW. ... 3910
3.83.30.8 "D" - Flag - WTW. ... 3911
3.83.30.9 "e" - Flag - WTW. ... 3912
3.83.30.10 "E" - Flag - WTW. ... 3914
3.83.30.11 "F" - Flag - WTW. ... 3916

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.12 "f" - Flag - WTW. ... 3918
3.83.30.13 "g" - Flag - WTW. ... 3920
3.83.30.14 "G" - Flag - WTW. ... 3922
3.83.30.15 "H" - Flag - WTW. ... 3923
3.83.30.16 "h" - Flag - WTW. ... 3925
3.83.30.17 "I" - Flag - WTW. ... 3927
3.83.30.18 "J" - Flag - WTW. ... 3929
3.83.30.19 "j" - Flag - WTW. ... 3931
3.83.30.20 "k" - Flag - WTW. ... 3933
3.83.30.21 "K" - Flag - WTW. ... 3935
3.83.30.22 "M" - Flag - WTW. ... 3937
3.83.30.23 "N" - Flag - WTW. ... 3939
3.83.30.24 "o" - Flag - WTW. ... 3941
3.83.30.25 "p" - Flag - WTW. ... 3943
3.83.30.26 "q" - Flag - WTW. ... 3945
3.83.30.27 "Q" - Flag - WTW. ... 3947
3.83.30.28 "r" - Flag - WTW. ... 3949
3.83.30.29 "t" - Flag - WTW. ... 3951
3.83.30.30 "u" - Flag - WTW. ... 3954
3.83.30.31 "V" - Flag - WTW. ... 3956
3.83.30.32 "v" - Flag - WTW. ... 3958
3.83.30.33 "W" - Flag - WTW. ... 3960
3.83.30.34 "w" - Flag - WTW. ... 3962
3.83.30.35 "X" - Flag - WTW. ... 3964

3.84 XML-Parser ... 3966

3.84.1 ! Sample-XML Data ... 3966
3.84.2 ! Smarty's XML-Features ... 3973
3.84.3 ! SPR - XML-Parser Flags ... 3981
3.84.4 ! XML. - eXtended Markup Language ... 3990
3.84.5 IML./ NML. - no parse ... 3995
3.84.5.1 nsc - Name Char ... 3995
3.84.5.2 nsc - Name Start ... 3998
3.84.5.3 wsa - Whitespace Asc ... 4001
3.84.5.4 wsc - Whitespace Character ... 4003
3.84.6 IML./NML. - after parse ... 4006
3.84.6.1 aeq - After EQ ... 4006
3.84.6.2 afs - Any Flags Set ... 4010
3.84.6.3 afz - Any Flags Zero ... 4016
3.84.6.4 bdq - Between DQ ... 4022
3.84.6.5 beq - Before EQ ... 4026
3.84.6.6 bsq - Between SQ ... 4030
3.84.6.7 btw - Between ... 4034
3.84.6.8 cbt - Close Bracket ... 4038
3.84.6.9 cmt - Comment ... 4042
3.84.6.10 eat - End Attribute ... 4047
3.84.6.11 edn - End Name ... 4051
3.84.6.12 etg - End Tag ... 4055
3.84.6.13 idq - Is DQ ... 4059
3.84.6.14 isq - Is SQ ... 4063
3.84.6.15 obt - Open Bracket ... 4067

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.6.16 osd - Outside SDQ ... 4071
3.84.6.17 sta - Start Attribute ... 4075
3.84.6.18 stg - Start Tag ... 4079
3.84.6.19 stm - Self Terminate ... 4083
3.84.6.20 stn - Start Name ... 4087
3.84.6.21 stt - System Tag ... 4091
3.84.7 XML - no parse ... 4095
3.84.7.1 Clr - Clear ... 4095
3.84.7.2 UnClean - UnClean Text ... 4098
3.84.7.3 Wrap ... 4101
3.84.7.4 cle - clear error ... 4106
3.84.7.5 clean - Clean Text ... 4108
3.84.7.6 geterr - get error ... 4111
3.84.7.7 lf - load file ... 4113
3.84.7.8 par - parse from var ... 4117
3.84.7.9 spl - Split Attrib ... 4121
3.84.7.10 unw - UnWrap ... 4123
3.84.8 XML. - after parse ... 4128
3.84.8.1 abm - Attrib by Number ... 4128
3.84.8.2 abn - Attrib by Name ... 4131
3.84.8.3 abv - Get Attrib by Value ... 4135
3.84.8.4 acnt - Get Attrib Count ... 4139
3.84.8.5 avbn - Attrib Value by Name ... 4142
3.84.8.6 dec pos - Decrement Position ... 4146
3.84.8.7 dmp - Dump ... 4149
3.84.8.8 dncb - Down Close Bracket ... 4152
3.84.8.9 dnob - Down Open Bracket ... 4155
3.84.8.10 ftxt - Flags to Text ... 4158
3.84.8.11 gbl - Get Block ... 4161
3.84.8.12 gend - Goto End Tag ... 4165
3.84.8.13 gfl - Get Flags ... 4168
3.84.8.14 gfn - Get First Node ... 4171
3.84.8.15 gftxt - Get Flags as Text ... 4174
3.84.8.16 gle - Get Len ... 4177
3.84.8.17 glev - Get Level ... 4179
3.84.8.18 gpo - Get Position ... 4182
3.84.8.19 gpr - Get Position for Previous ... 4185
3.84.8.20 gro - Get Root ... 4188
3.84.8.21 gsc - Get Asc ... 4191
3.84.8.22 gse - Get Start End Tags ... 4194
3.84.8.23 gst - Goto Start Tag ... 4197
3.84.8.24 gtfn - Goto First Node ... 4200
3.84.8.25 gtn - Get this Name ... 4203
3.84.8.26 gtr - Goto Root ... 4206
3.84.8.27 gxp - Get XML Part ... 4209
3.84.8.28 inc pos - Increment Position ... 4213
3.84.8.29 ixp - Insert XML-Part ... 4216
3.84.8.30 met - My End Tag ... 4220
3.84.8.31 mst - My Start Tag ... 4223
3.84.8.32 prt - Print ... 4226
3.84.8.33 ptse - Position to Start or End Tag ... 4229

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.34 rpa - Reparse ... 4232
3.84.8.35 sad - Search All Down ... 4236
3.84.8.36 sau - Search All Up ... 4239
3.84.8.37 scr - Set Chr ... 4242
3.84.8.38 sdn - Search Down Tag Name ... 4247
3.84.8.39 sid - Search Inside Down ... 4250
3.84.8.40 siu - Search Inside Up ... 4254
3.84.8.41 sna - Search Name and Attribute ... 4258
3.84.8.42 snaup - Search Name and Attribute Up ... 4262
3.84.8.43 spa - Split Attrib at ... 4267
3.84.8.44 spdn - Search Pattern Down ... 4270
3.84.8.45 spo - Set Position ... 4273
3.84.8.46 spup - Search Pattern Up ... 4276
3.84.8.47 ssc - Set Asc ... 4280
3.84.8.48 std - Start Tag Down ... 4285
3.84.8.49 stu - Start Tag Up ... 4287
3.84.8.50 sun - Search Up Tag Name ... 4290
3.84.8.51 sxp - Set XML Part ... 4293
3.84.8.52 upcb - Up Close Bracket ... 4298
3.84.8.53 upob - Up Open Bracket ... 4301

3.85 ZIP. - Handle Archives ... 4304

3.85.1 ! Smarty's ZIP-Commands ... 4304
3.85.2 IFZ. / NFZ. - If File in ZIP ... 4306
3.85.3 ZIP. - Handle Archives ... 4309
3.85.4 cf - compress file ... 4313
3.85.5 cv - compress var ... 4318
3.85.6 gtfn - get zip filename ... 4321
3.85.7 sdk - set default compression ... 4323
3.85.8 uf - unzip file/folder ... 4326
3.85.9 uv - uncompress var ... 4329
3.85.10 zf - Zip Folder ... 4332

4. 4. AddOn Components 4335

4.86 MR Connect for use with SAP® ... 4336

4.86.1 ! Smarty's Addon for use with SAP® (Connect to SAP®) ... 4336
4.86.2 GSP. - Get SAP® Data ... 4342
4.86.3 ISP. / NSP. - If SAP® State ... 4349
4.86.4 SSP. - Set SAP® Data ... 4352
4.86.5 WSP. - Wait for SAP® State ... 4354

4.87 InstaRec - the Recorder ... 4357

4.87.1 4.9.1 InstaRec - User-Templates ... 4357
4.87.2 4.9.2 InstaRec - AutoPatchSkript.exe ... 4360

4.88 GPX-Korrektor ... 4362

4.88.1 Eingabe des BING bzw. Goggle-API Keys ... 4368
4.88.2 FIle Load/Save-Buttons ... 4370
4.88.3 H-/V+- Buttons ... 4371
4.88.4 Line Points - Button ... 4374

4.88.5 Make Inbetween / Delete Button 4379
4.88.6 UNDO Selected - Button 4384
4.88.7 Restore/Clear Selection Buttons 4385
4.88.8 Check Timestamp Button 4386
4.88.9 Check and Repair Button 4387
4.88.10 ADD / SUB / GS / CLR / FND Buttons 4389
4.88.11 Choose between BING Maps and Google Maps 4392
4.88.12 Display from / No. of Points 4393
4.88.13 Additional Comments 4394

5. Last Changes & Updates 4395

Index 4415

*© 2023 Theo Gottwald * https://www.smart-package.com*

1. Starting Guides

1 1. Starting Guides

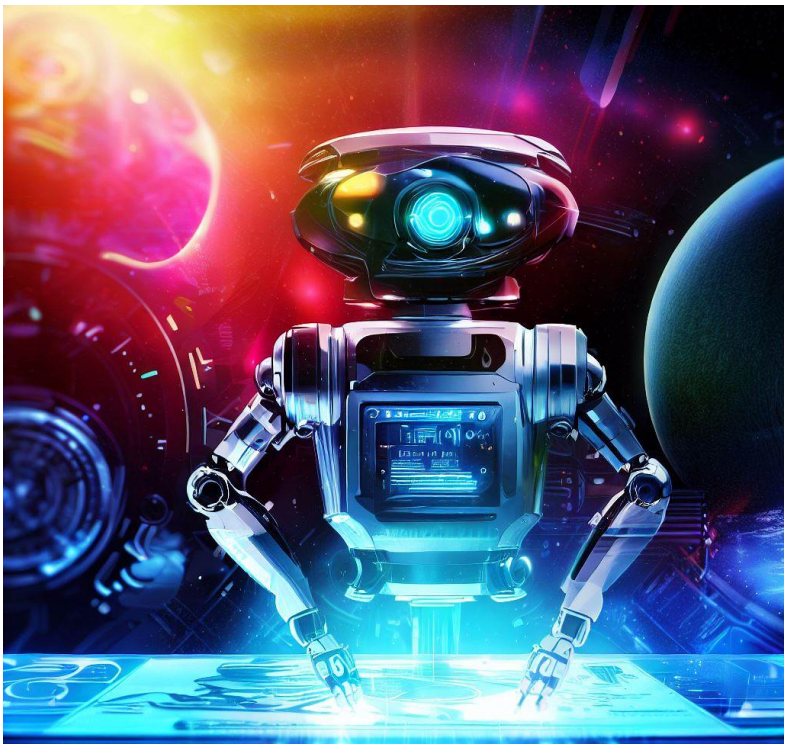
1.1 1.1. About the program

1.1. About the program Top 41 Next 39

Starting Guide

Introducing Smart Package Robot 2.0

What it is and what it is good for.



The Smart Package Robot (SPR) is a fascinating tool that allows you to automate tasks, almost like a magical scriptwriting system. It's built on Theo Gottwald's MiniRobots technology, so you might see the term "MiniRobot" sprinkled throughout our explanations like little breadcrumbs on a trail.

While we officially call it the "Smart Package Robot" (SPR), you might also stumble upon the name "IT-Robot". Why "IT-Robot", you ask?

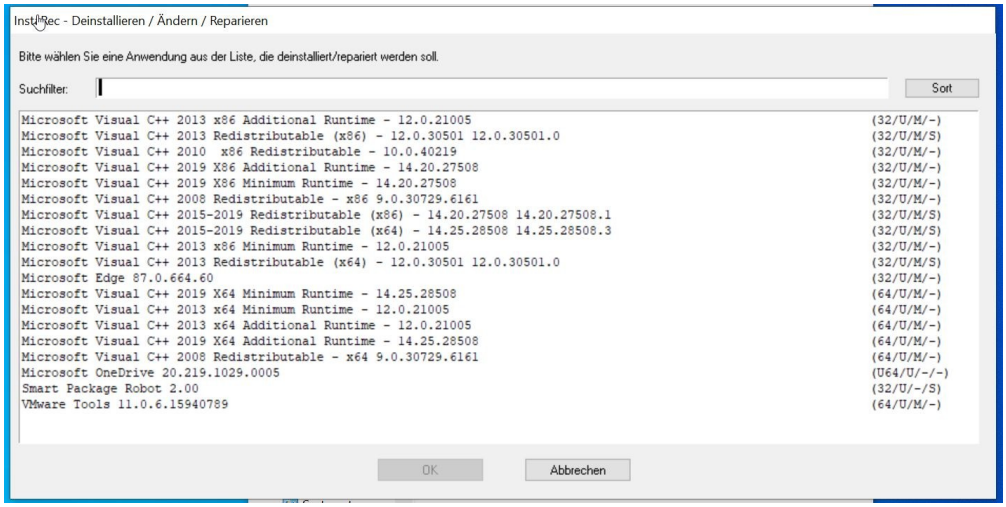
Well, it's because this incredible tool can capture or script almost any IT-related task you can think of. It's as if you had a little robot assistant sitting at your computer, typing away!

We've also given it the name "Smart Package Robot" (SPR) because we think it's really clever to let a robot handle the monotonous, repetitive work for you. Imagine a friendly robot taking care of all the dull stuff so you don't have to!

© 2023 Theo Gottwald * https://www.smart-package.com

One of the coolest features of the Smart Package Robot is that it can record the installation and uninstallation of software. With the built-in "InstaRec" and "UnInstaRec" programs, it's like having a video camera for your software, recording every step so you can play it back whenever you want

Pictures: InstaRec (for Installations)

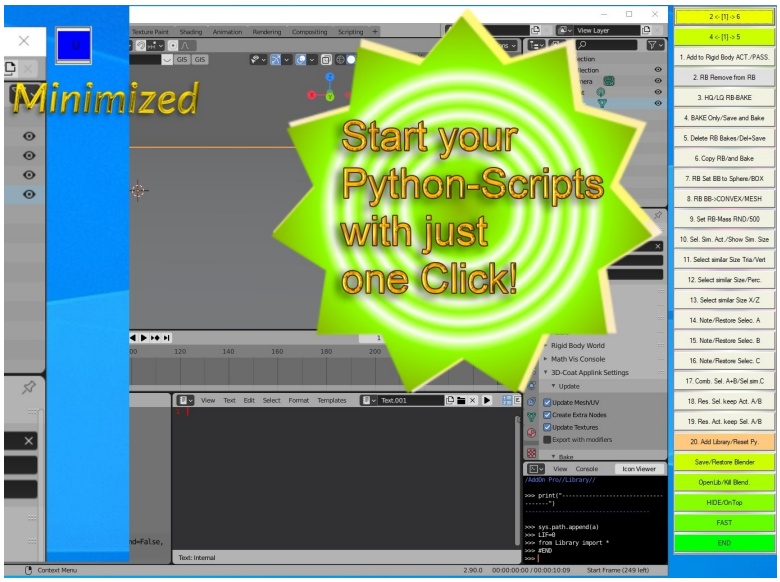


Picture: Un-InstaRec (for Uninstallations).

You can add a Buttonbar to any Programm and put your own Macros on these Buttons.
This way you can use the SPR as Software-Amplifier.

With SPR's nifty Buttonbar feature, you can customize any program by adding your own macros to buttons. Think of it as a way to supercharge your software, making SPR your personal software amplifier. This allows you to enhance the functionality of your software, adding new features and tools at the click of a button

© 2023 Theo Gottwald * https://www.smart-package.com



Picture: Quickbuttons AddOn for Blender (from www.blender.org)

But that is just one thing you can do with your new Smart Package Robot. With time, you'll discover many more things.

SPR can be used on any Windows OS, from Vista to Windows 10/x64.
If you still use Windows XP or lower, please contact us, we can provide you with a older version that may supports these windows versions.

Using SPR you will be able to script and automate many tasks of your daily work.
And you can compile your automation scripts into Standalone EXE-Files.

These files are very compact and can be distributed with any Software Distribution System that allows the distribution of Executable files.With SPR, you'll be able to script and automate many of your daily tasks, making your work more efficient and less monotonous. Plus, you can compile your automation scripts into standalone EXE files, perfect for distribution via any software system that supports executable files. These files are incredibly compact, making them easy to distribute and use across different systems

*© 2023 Theo Gottwald * https://www.smart-package.com*

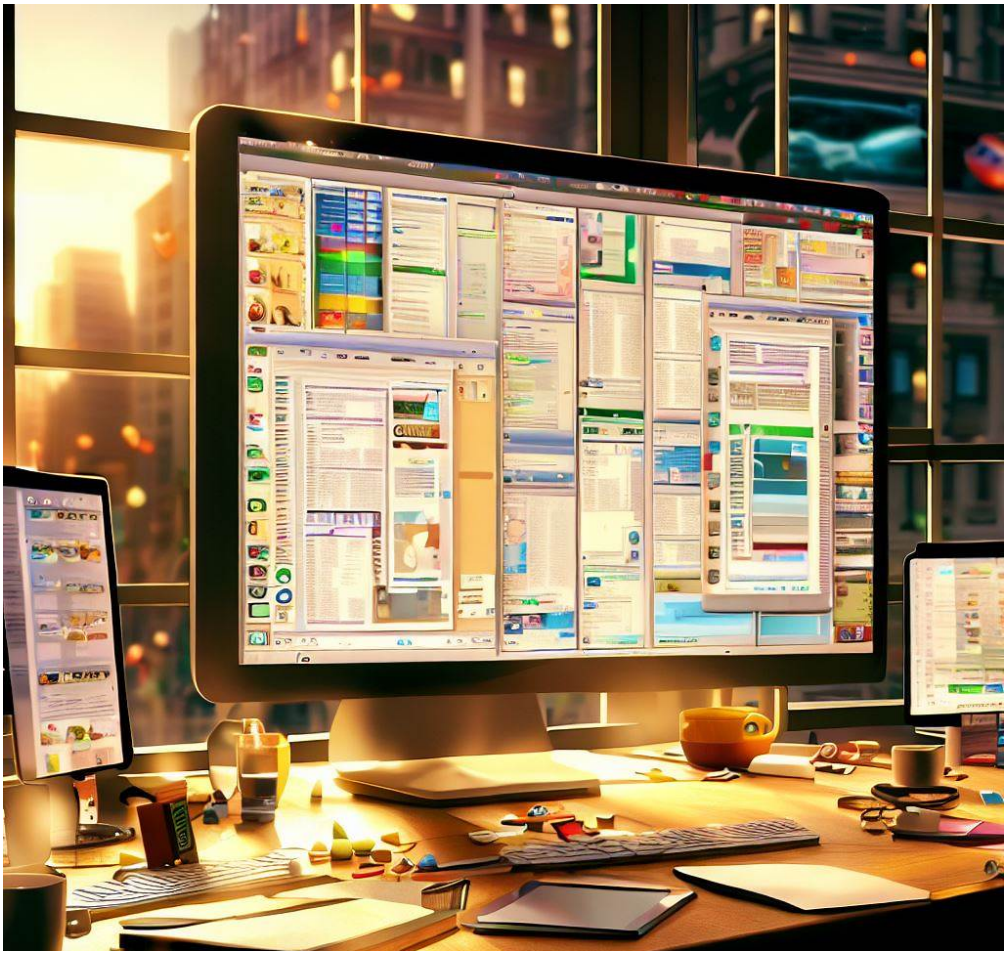
1.2 1.1.1 What else is it good for?

1.1. About the program Top 41 Next 39

Starting Guide

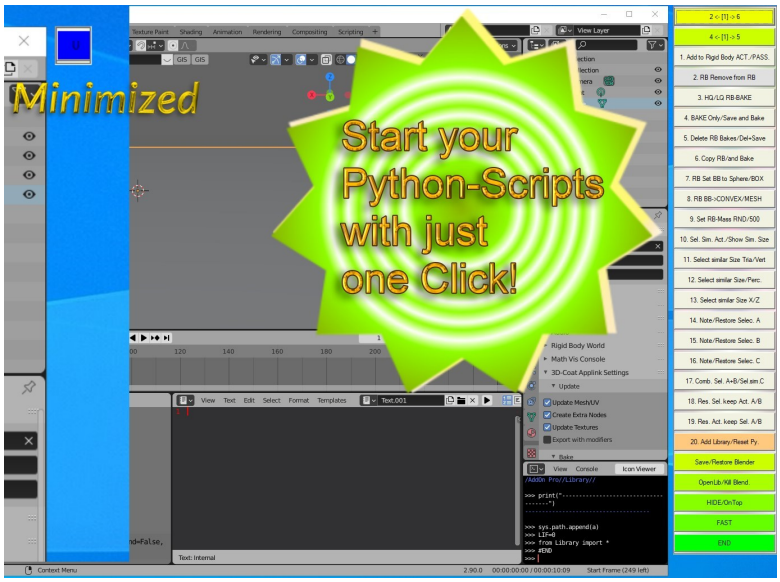
Smart Package Robot 2.0

What else it is good for?



Did you know that you can customize your Smart Package Robot (SPR) to have its very own button bar for any program?
You can put your personalized macros on these buttons, like sticking your favorite stickers on a notebook. In this way, the SPR becomes not just a tool, but a software amplifier, boosting your programs with extra features and capabilities.
It's like giving your software a supercharge, making it even more powerful and efficient.

© 2023 Theo Gottwald * https://www.smart-package.com



Picture: Quickbuttons AddOn for Blender (from www.blender.org)

Your journey with the Smart Package Robot (SPR) is just beginning, and there's so much more to discover. Over time, you'll unearth a treasure trove of features and abilities!
The SPR is like a chameleon, adapting to any Windows Operating System from Vista to Windows 10/x64. If you're still using Windows XP or an older version, don't worry! Just give us a call and we'll provide you with an older version that's compatible with your system.
With the SPR at your side, you'll have the power to script and automate a multitude of tasks, making your daily work as smooth as a magic carpet ride. Even better, you can transform your automation scripts into standalone EXE files.
These compact files can be shared far and wide through any software distribution system that allows executable files.

It's like having your own little robots ready to lend a hand wherever they're needed!

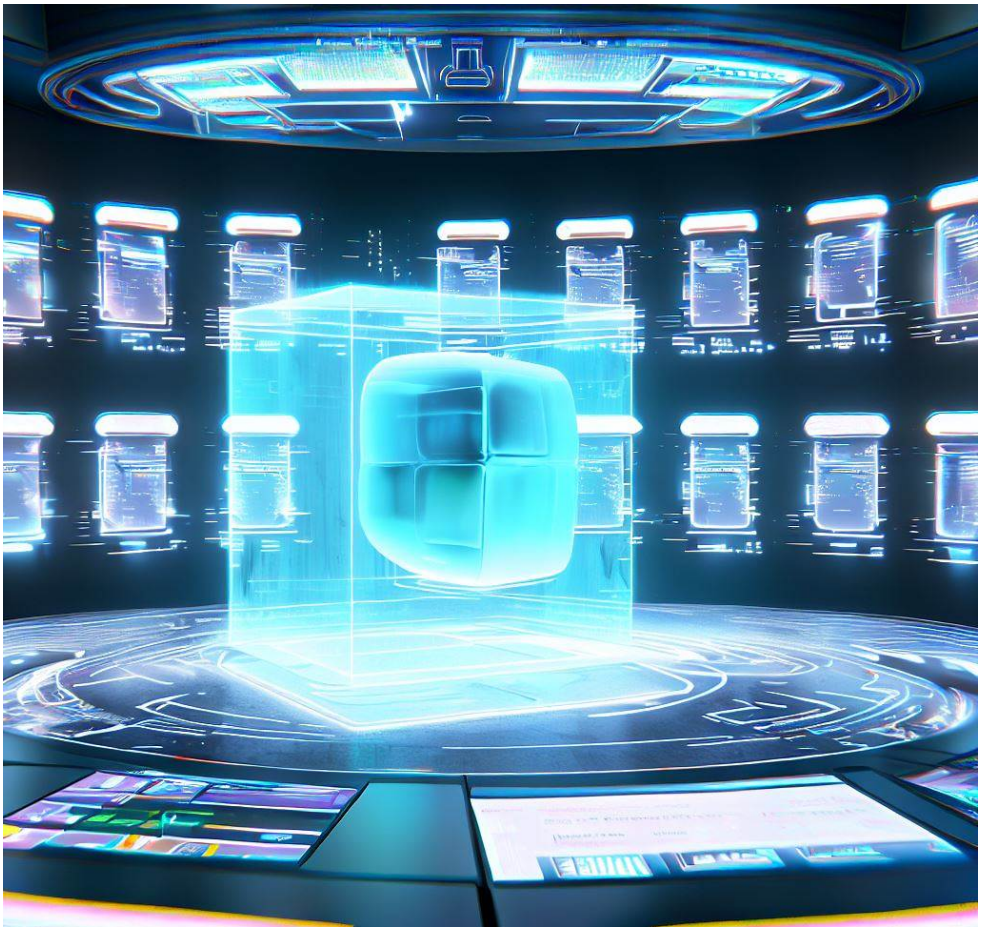
*© 2023 Theo Gottwald * https://www.smart-package.com*

1.3 1.1.2 What you See is what you get: as Standalon EXE-File

Starting Guide

Smart Package Robot 2.0

Why do we call them "Standalone-EXE" files?



Why do we use the term 'Standalone-EXE' files?

Well, it's because these files are self-sufficient and independent.
They don't require any other files or programs to run on the target system, which means you don't need to install SPR or anything else for that matter.
They are truly standalone!

What's more, these Standalone-EXE files are incredibly compact.
Believe it or not, they could even fit on an old-school floppy disk!
When you launch one of these files, it replays all of your recorded actions on the target machine, just like a perfectly rehearsed performance.

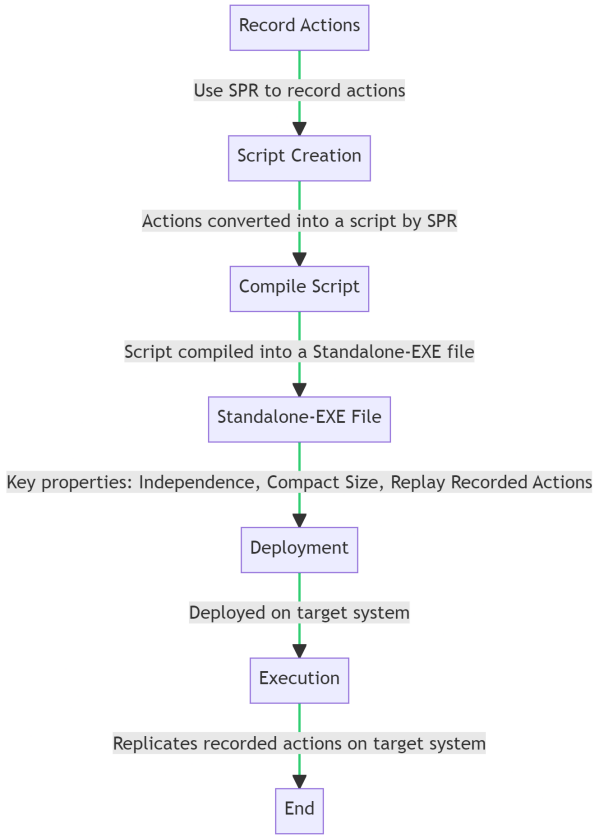
It's crucial to highlight that the underlying MiniRobots Technology does not rely on screen coordinates or fixed time delays, unlike some other automation systems. Instead, your scripts are synchronized with the actions of the setup program, mimicking the way a human would interact with the system.

© 2023 Theo Gottwald * https://www.smart-package.com

This makes SPR's automation more reliable and adaptable.

Moreover, the current version of MiniRobots does not depend on VB 6, Visual C, or any other runtimes being installed on the target system.
This makes SPR even more flexible and easy to use across various systems.

If you use the SPR for automated Setups, you may often follow this Path:



The flowchart follows these steps:

1. **Record Actions**: Use the SPR to record the desired actions on your own system.
2. **Script Creation**: The actions recorded are then converted into a script by SPR.
3. **Compile Script**: The script is then compiled into a Standalone-EXE file.
4. **Standalone-EXE File**: Highlight the key properties of this file - independence (no dependencies on the target system), compact size (small enough to fit on a floppy disk), and the capability to replay recorded actions.
5. **Deployment**: The Standalone-EXE file is then deployed on the target system where it can run without the need for SPR or any other program.
6. **Execution**: The Standalone-EXE file runs on the target system, replicating the recorded actions as if a human were performing them.

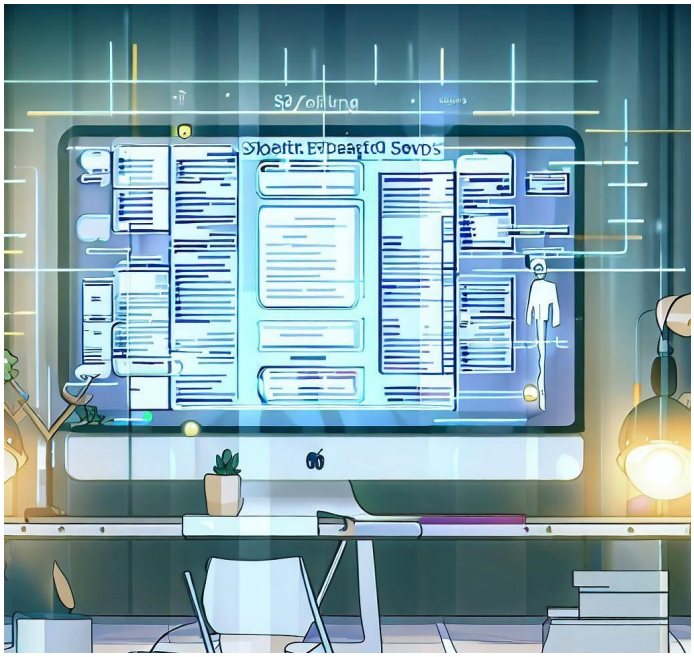
© 2023 Theo Gottwald * https://www.smart-package.com

1.4 1.2. System Requirements

Starting Guide

System Requirements

What are the hardware and software requirements for this program?



The Smart Package Robot (SPR) for Windows is a versatile automation and scripting system.
It's compatible with most modern Windows operating systems (x86 or x64), making it a great fit for a wide range of computers.

The "standalone packages" created with SPR can be used on any newer Windows OS. This includes **Windows Vista™, Windows 7™, Windows 8™, Windows 10™, and Windows 11™.**

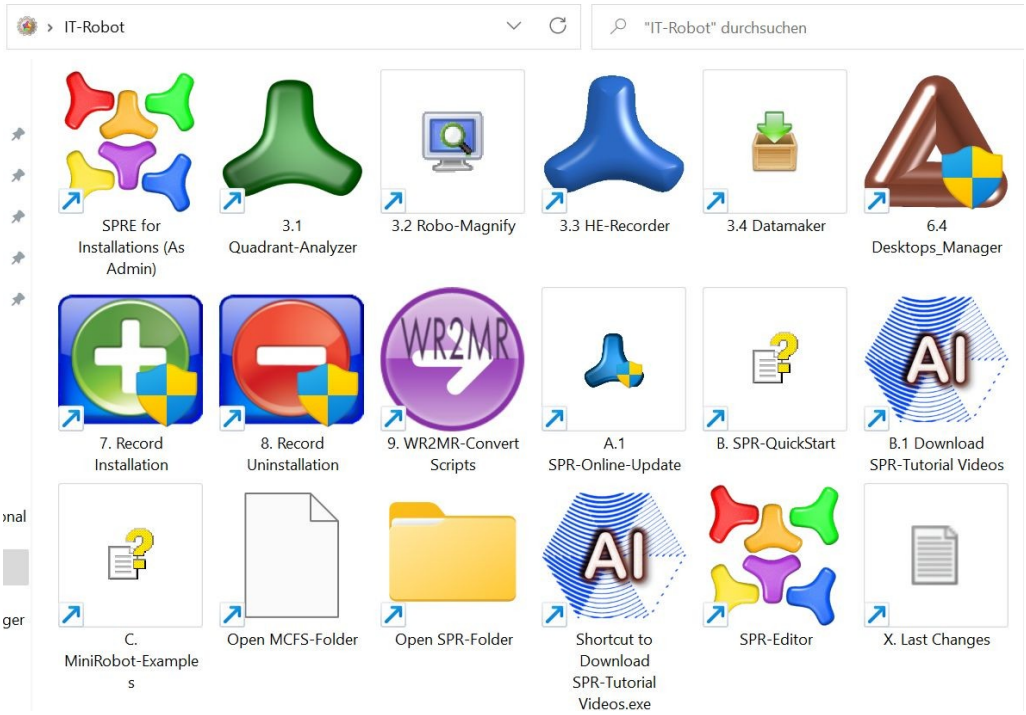
Just make sure your system is up-to-date and you're good to go!
When it comes to recording installations, there's one thing to note if you're using Windows Vista or a later version:
you'll need to turn off the User Account Control (UAC) and restart your computer.
But don't worry - this only applies to the computer where you're recording the installation.

As for the target machines, you won't have any issues regardless of whether UAC is on or off.
The standalone packages run with system rights from the software distribution agent, so they won't be affected by UAC settings.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Remember, all Windows and other trademarks mentioned belong to their respective owners.

Enjoy using the Smart Package Robot to automate your tasks and enhance your productivity



***If you need to use the SPR for Windows XP, please contact us, we can provide you with an older XP-compatible version.**

*© 2023 Theo Gottwald * https://www.smart-package.com*

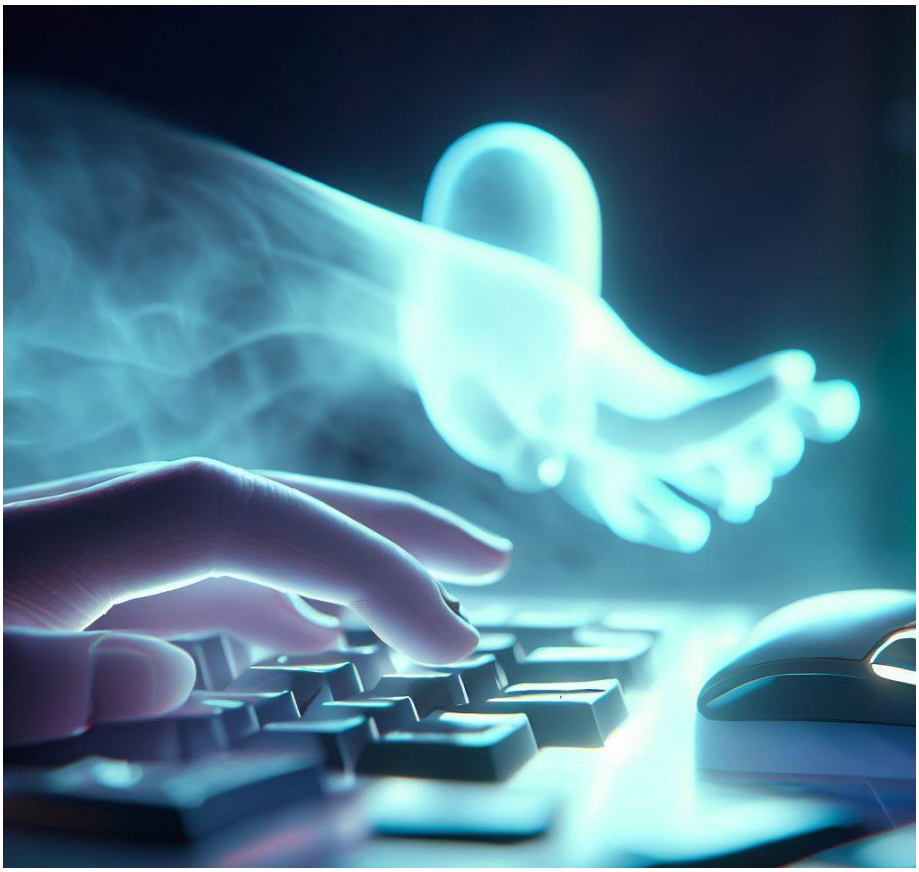
1.5 1.3. How to Start

Starting Guide

How to Start

Things you should be aware of before you begin using the program.

Picture a friendly ghost that moves your mouse, types on your keyboard, and carries out your tasks.
This isn't just any ghost - meet Smarty, your new virtual helper.



Smarty can perform a series of commands like moving your mouse and clicking your keyboa
But to unlock Smarty's potential, you'll need to learn its unique language - a scripting langua

Think of this scripting language as the secret handshake between you and Smarty.
It's how you tell Smarty exactly what you want it to do.

It may seem like a mystery at first, but with practice, it will start to make sense.
As an added bonus, Smarty is also fluent in **PowerShell** and **VBS-Code**!

*© 2023 Theo Gottwald * https://www.smart-package.com*

There are numerous videos on YouTube™ about the Smart Package Robot, both in German and English.
Watching these videos is a great way to learn how to navigate the SPR more comfortably.

To kick things off, let's create a simple "Hello World" program. Open the SPR-Editor and type:

```
MBX.Hello World!
```

Don't forget to save your script before running it. If you need help, refer to the floating bar buttons on the SPR-Editor (picture below) for guidance.
To run the script, click on the Play Button on the side of the SPR-Editor. Save the script in a location of your choice if needed.
The image below explains what each button on the SPR-Editor's floating bar does. Happy scripting with your new friend Smarty.

© 2023 Theo Gottwald * https://www.smart-package.com

For a deeper dive into Smarty's capabilities and to enhance your proficiency, here are some steps you can take:

1. Begin with the "Getting Started Guides":

This chapter is designed to introduce you to the basic principles and functions of the system.

It's a great starting point to get a handle on the foundations of the Smart Package Robot and its scripting language.

2. Utilize Online Learning Resources:

YouTube hosts a range of educational content on the Smart Package Robot, with both English and German language videos available.

These tutorials cover everything from basic scripting to advanced automation techniques.

They're a fantastic resource for visual learners and can be paused or replayed as often as needed.

3. Ask for Help When Needed:

If you're ever unsure or run into a challenge, don't hesitate to reach out to us.

We're here to assist you and ensure you have the best possible experience using the Smart Package Robot.

Our team can answer questions, provide clarification, and offer guidance to help you navigate any hurdles.

Remember, mastery comes with time and practice.

Enjoy the journey of learning and exploring the power of automation with Smarty.

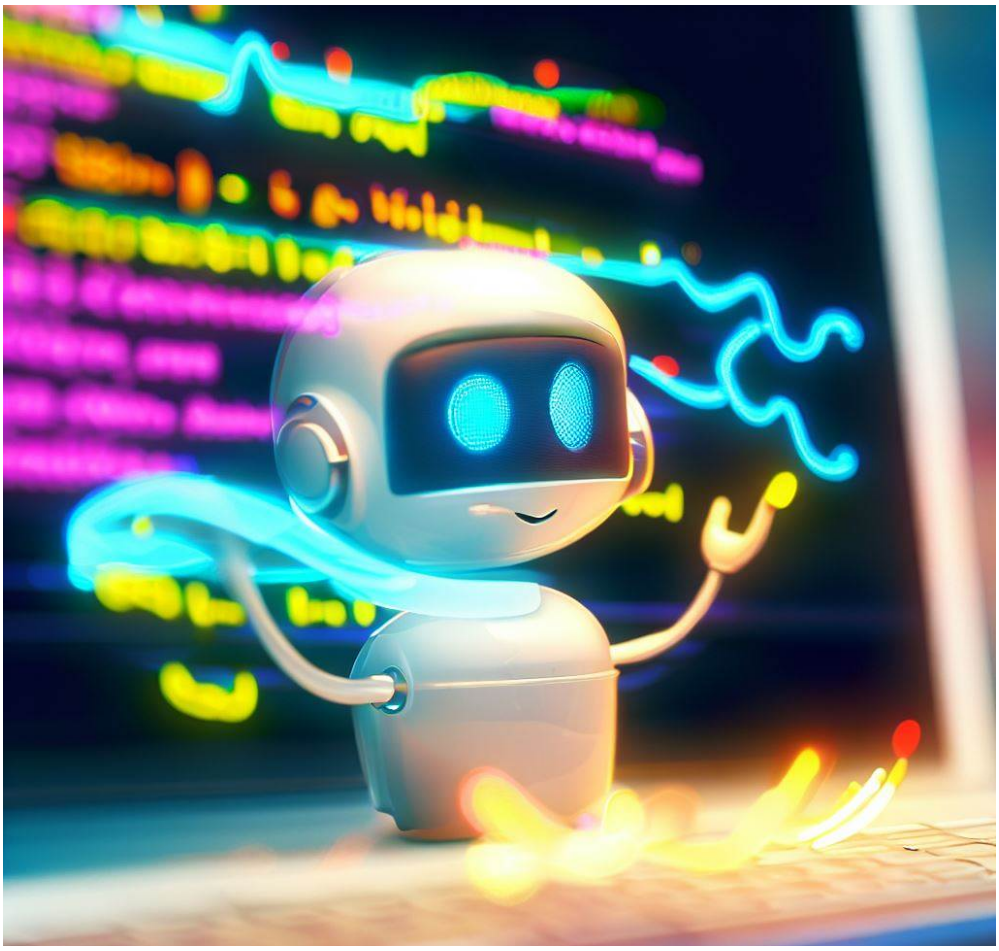
*© 2023 Theo Gottwald * https://www.smart-package.com*

1.6 1.3.1 The Scripting Process

Starting Guide

The Scripting Process

How Can You Create Your Scripts?



1.Using InstaRec to Record Scripts:

The InstaRec Recorder is like a reliable secretary, dutifully noting down your actions during an installation process.
You simply perform the installation steps as you usually would, and InstaRec does the rest, creating a script that can be replayed with ease.
This is particularly useful when you need to deploy the same installation on multiple systems.
You just need to perform the installation once, and then let the Smart Package Robot replicate it everywhere else.

2. Writing Scripts by Hand:

© 2023 Theo Gottwald * https://www.smart-package.com

For tasks that can't be recorded—perhaps because they involve complex or non-standard interactions—you can directly write your own scripts using the Smart Package Robot Editor.
This approach gives you the freedom to input the necessary commands manually, offering you total control over every detail of the automation process.
Think of it as a direct conversation between you and Smarty, where you tell Smarty exactly what to do.

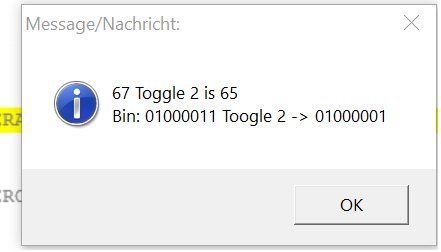
3. Recording with the Smart Package Robot Editor:

If your task isn't too complex and you don't want to launch InstaRec, the Smart Package Robot Editor has a built-in recording feature that can come in handy.
With a simple press of the 'N' (Normal) or 'P' (Popup) button, you can start recording your actions and creating a script on the fly.

4. Utilizing Sample Scripts as Templates:

In the installation directory of the Smart Package Robot, you'll find a treasure trove of resources in the "Sample Scripts Folder".
This is not just a collection of examples, but rather a launching pad for your own scripting journey.
Many of the sample scripts can serve as a template for your own tasks.
You can modify these templates as needed, add your own touch, or even combine different elements from various sample scripts to create a completely new automation process.
These templates are like the scaffolding of a building, providing a solid structure upon which you can construct your own unique scripts.
It's a great starting point for those new to scripting and a time-saver for more experienced users.
Consider these sample scripts as your personal scripting toolkit, helping you to master the art of automation with the Smart Package Robot.

```
VAN.$$LEA=1
VAN.$$P01=67
VAN.$$P02=2
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Toggle|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 Toggle $$P02 is $$P03$crlf$Bin: $$ERA
BIT.Toggle|$$P03|$$P02|$$P04|$$LEA
BIT.Show|v$$P04|$$ERD|$$LEA
MBX.$$P03 Toggle $$P02 is $$P04$crlf$Bin: $$ERC
ENR.
```

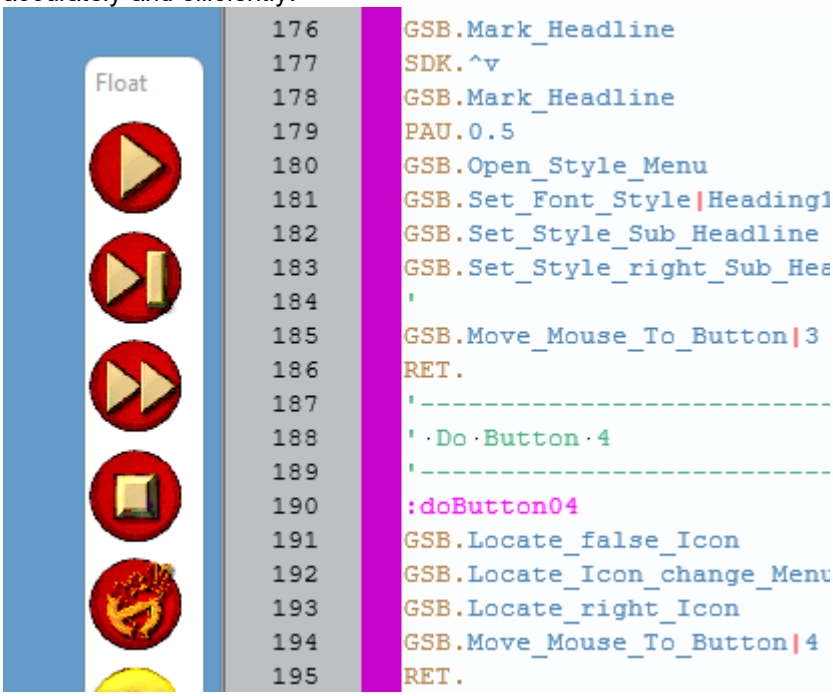


5. Understanding the Script Execution Process:

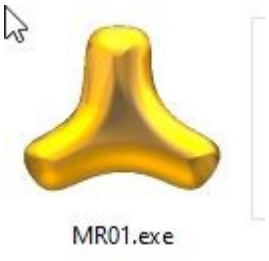
You've meticulously written or recorded your script, but what happens when you hit that "PLAY" button?
Let's break it down. As soon as you press "PLAY", the Smart Package Robot springs into action.
It starts by interpreting your script, understanding each command and preparing the necessary actions.
It's like a director reading through a script before the curtain rises on a play. This process ensures that everything is in order and ready to go.

© 2023 Theo Gottwald * https://www.smart-package.com

Following this, the Smart Package Robot performs each action sequentially, just as you have instructed.
It moves the cursor, clicks on buttons, types in information, and so on, all while maintaining the flow and timing you've set.
It's like watching a well-rehearsed performance, with each action smoothly transitioning into the next.
With the Smart Package Robot, you can be sure that your tasks will be performed accurately and efficiently.



Diving Into the Execution Process:
When you press "PLAY" in the SPR-Editor, it sets off a chain of events.
The first thing it does is to summon our dependable friend, **MR01.exe**.
This file is the "Robot Executable", the brains of the operation.
Think of **MR01.exe** as the marionette master who pulls all the strings.
It takes your script commands, interprets them, and brings your Smart Package Robot to life, guiding it to perform the tasks you've outlined.
In this way, your script becomes a set of precise, actionable instructions that **MR01.exe** uses to orchestrate the Smart Package Robot's actions.



Delving Deeper into MR01.exe
This is where the magic happens!
The **MR01.exe** is the heart and soul of your Smart Package Robot. Once you've hit "PLAY" in the SPR-Editor, the Robot Executable, **MR01.exe**, takes over.
It's responsible for translating and executing the majority, if not all, of your script commands.
It's as if **MR01.exe** takes your script and breathes life into it, making it dance to your tunes.

© 2023 Theo Gottwald * https://www.smart-package.com

Your Smart Package Robot's actions are orchestrated entirely within the confines of **MR01.exe**.
This is why we often say that the magic of automation lives within **MR01.exe .**

6. 6. The Art of Preprocessing:

The first pivotal step in executing your script lies in the hands of the "Pre-Processor." This diligent component meticulously prepares your script, making it ready for the ensuing "Speedcode Compiler."
The Pre-Processor's role is multifaceted, focusing on refining your script to be as precise and streamlined as possible, thereby facilitating a smooth execution.

Unraveling MACROs:
One of the key tasks of the Pre-Processor involves deciphering MACROs. MACROs are concise code substitutes that represent a more extensive piece of code.
They help keep your script compact and more readable. However, improper definition or usage of MACROs can introduce errors. The Pre-Processor, acting as a vigilant guard, checks and resolves these potential discrepancies, ensuring a seamless run of your script.

Integrating #INC Files:
The Pre-Processor also takes on the role of a detective when it comes to #INC files. These files are external entities containing supplementary code or resources, such as libraries or images, which are essential for your script to function.
The Pre-Processor hunts for any missing #INC files and incorporates them into your script, guaranteeing that all necessary elements are in place.

Bug Squashing:
Next, the Pre-Processor assumes the persona of a bug exterminator. It scans your script for coding bugs - pesky issues that may arise due to typos, incorrect syntax, or missing punctuation, like semicolons.
The Pre-Processor rectifies these bugs, ensuring your script can perform its dance without stumbling.

Syntax Error Checking:
Finally, the Pre-Processor morphs into a proofreader, looking out for syntax errors that can't be automatically corrected. Syntax errors occur when the computer cannot comprehend the script due to issues like missing brackets or incorrect variable names. If such errors are detected, the Pre-Processor not only identifies them but also suggests possible solutions for you to rectify them.

Once the Pre-Processor has diligently performed all of these tasks, it passes the refined script onto the Speedcode Compiler. The Compiler then transforms the preprocessed script into machine code, a language that your computer can understand and execute.
In this manner, the Pre-Processor plays a critical role in the script creation process, ensuring your script is as optimized and error-free as possible before it faces the Compiler.

7. The Speedcode-Compiler

In script creation, a significant hurdle lies in the time it takes to fetch a command.
The Speedcode Compiler, a pivotal component in the script execution process, addresses this by transforming various elements of the script into a format that facilitates quicker execution.
This transformative process is commonly known as "converting to Speedcode."

Efficient Label Formatting:

© 2023 Theo Gottwald * https://www.smart-package.com

The Speedcode Compiler begins its work by streamlining labels. Labels serve as milestones, marking specific segments in the script for easier navigation and comprehension.
By reformatting these labels into a more efficient format, the Speedcode Compiler significantly reduces the command fetch time.

Optimizing Conditional Statements:
Next, the Speedcode Compiler optimizes conditional statements.
These statements play a crucial role in steering the script's flow, determining which commands are executed and which are bypassed.
By reshaping these statements into a more conducive format, the Speedcode Compiler accelerates the command fetch process.

Loop Formatting:
Loops, which are employed to repeat a section of the script multiple times, are also a focal point for the Speedcode Compiler.
Improperly formatted loops can cause substantial delays. By remodeling loops into a more efficient format, the Speedcode Compiler curtails the command fetch time.

General Command Formatting:
Finally, the Speedcode Compiler reformats all commands into a format that expedites execution.
This includes any other commands not explicitly mentioned above, such as function calls, assignments, and more.

After the entire code has been converted to Speedcode, it is ushered to the next phase of the process.
This could involve further compilation, optimization, or direct execution of the script.
The overarching goal of this intricate process is to enhance the script's efficiency, thereby minimizing command fetch time and amplifying the script's overall performance.

8. The Script Execution

The Script execution is the grand finale in the script creation process.
It breathes life into your script.
This is the moment where the meticulously prepared script is let loose on a computer, with each command sequentially coming to life.

Just like an orchestra performing a symphony, the computer meticulously reads and performs each line of the script, usually from top to bottom.
The maestro, which is the computer in this case, reads the first line of the script, interpreting and executing the command it contains.
Next, it smoothly transitions to the second line, performing the same process.
This sequential performance continues unabated, line by line, until the final note - or in our case, the final line of the script.

To ensure a flawless performance, the execution of the script is closely monitored.
If all elements align harmoniously, the script executes without a hitch, each command sequentially springing into action, leading the script to its intended conclusion.
Once the final command has been executed, the task at hand is complete, marking the end of the process.

To sum up, script execution is the culminating point in the script creation process, turning lines of code into a symphony of actions.
Like a well-rehearsed ensemble, the script, if well-written, will perform seamlessly from start to finish, bringing your digital masterpiece to life.

*© 2023 Theo Gottwald * https://www.smart-package.com*

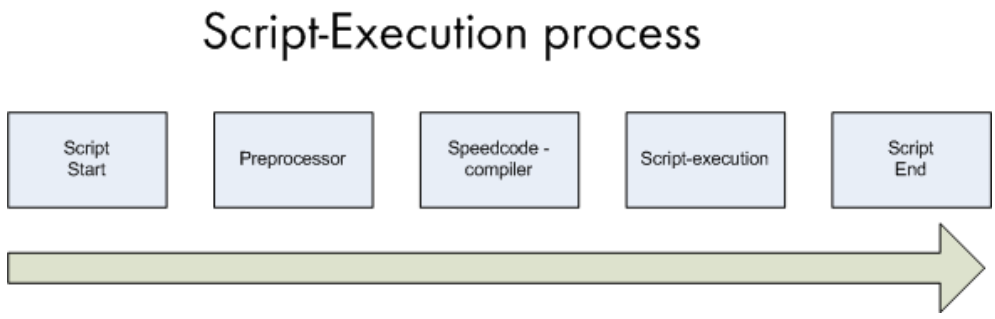
9. The Script End

In the grand theatre of script execution, the final act is often overlooked, but it's an essential part of the process: the cleanup phase.
This phase ensures a graceful and efficient exit, tying up loose ends by releasing the resources that were employed during the script's grand performance, such as memory, files, and other resources.

Picture this phase as the crew that enters the stage after the play has ended, tidying up the props and scenery.
Without them, the theatre would soon fall into disarray.
In the same way, the cleanup phase prevents memory leaks and other potential issues that could turn into major headaches down the line if resources aren't properly managed.

Now, if the script is a compiled executable, it doesn't just exit stage left without a word.
Instead, it leaves behind a parting gift: an error code.
This error code acts as a messenger, communicating the final status of the script's performance to the initiating process.
A standing ovation is represented by a value of zero, indicating a flawless execution, while any other number signals that there was a hiccup during the performance.
This error code can be used by the backstage crew (the initiating process) to figure out what went wrong and how to fix it in future performances.

In conclusion, the script end, or the cleanup phase, plays a vital but often underappreciated role in the execution process of a script. It ensures a tidy and efficient conclusion, diligently releasing resources, and, if it's a compiled executable, providing valuable feedback through an error code.



© 2023 Theo Gottwald * https://www.smart-package.com

1.7 1.3.2 Commands you should start with

1.3. How to Start Previous 39 Top 41 Next 73

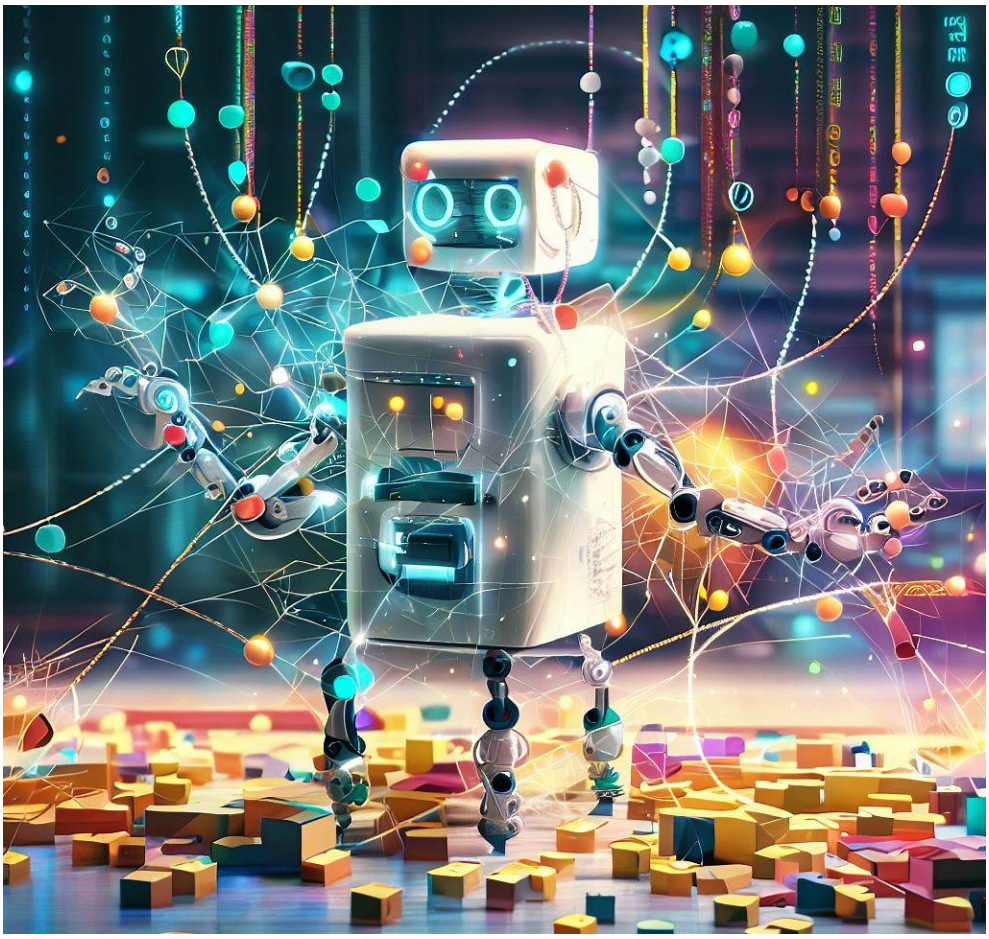
Starting Guide

Commands you should know

These are the commands that you should start with

1. Understand the Prototype Parameter-System

You need to understand some general concepts how the Robot-Script works.
"How do I tell the Robot what to do". For this the "Prototype Parameter System" is important.
So take a look:
The Prototype Parameter System 73



2. Know all Commands that are used by InstaRec

Here are some Commands that you should Know because they are used by InstaRec:

STW. 3814 (Find Application WIndow)
SCW. 3738 (Find Button etc. inside the Application Window)

© 2023 Theo Gottwald * https://www.smart-package.com

MLC. 2448 (Click with the Mouse using the Mousepointer "Compatible Engine")
MLE. 2492 (Click with the Mouse not using the Mousepointer "Direct Engine")
BSC. 2435 (Click as often as needed to close a Window)
BTC. 635 (Check or Uncheck Checkmarks)
WPT. 2913 (Wait until a Program has ended)
VAR. 3562 (Define a Variable)
DBC. 2863 (Set up a Parallel running Button-Clicker Robot)
END 2362 (End the Script immediately)

If you know what these few commands do, then you can do already a LOT of things.
Next possibly look at:
IAI. 2601 (Check if an Application is installed or not)
GEP. 2582 (Get Path of an registered Executable)
MBX. 2036 (Display a Messagebox)
IVV. 3534 (Check if a Varibale has some value or not)
CAL. 2315 (So mathematical and logical caluclations)

These are the most used commands.

3. Take a look at the more advanced commands

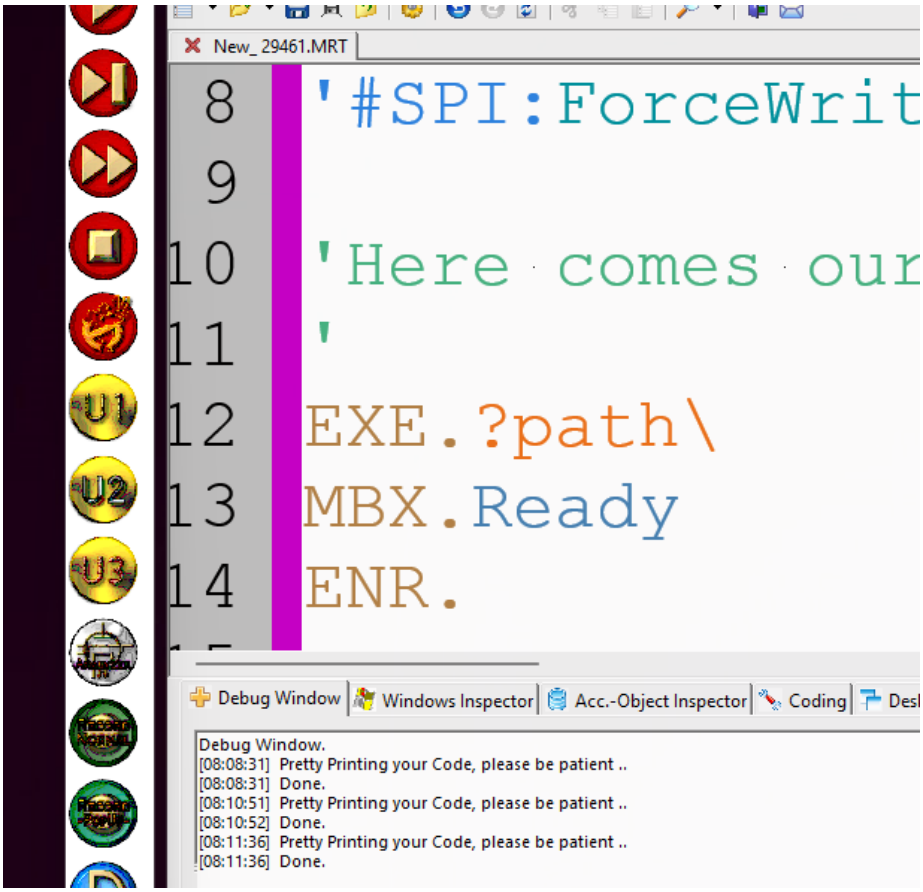
Here we have commands that use the Accessibility Engine and the graphical Commands.
Also we have the Commands to use the Buttonbar (Plugin01.exe).

AGR. 564 - (Find Element in Browser or Office Application)
PIL. 2702 - (Launch the Buttonbar)

4. Invest some time. Save more time.

You will also find a folder with "Sample Scripts". These scripts are carefully designed and commented.
Use them to see how things are done. Try them, change them. Play with the system. Make experiences.

© 2023 Theo Gottwald * https://www.smart-package.com



4. Later you can take a look on the advanced features. Try the MACRO-Option 253 and **create your own commands**. Include them into your scripts using the '#INC:-Preprocessor-command. 248

5. Its highly recommended that you make a training with a SPR*-Professional. Ask your distributor about the conditions.

6. Don't think the **SPR** is **only** usable for automated Software Installations. While this is the case, it is also **usable** for multiple **other tasks**.

7. Use it for Logging in to your favorite web-site, use it for automated testing. Let other people in your company also use it! You have the license to automate things! Use it.

* Smart-Package Robot

*© 2023 Theo Gottwald * https://www.smart-package.com*

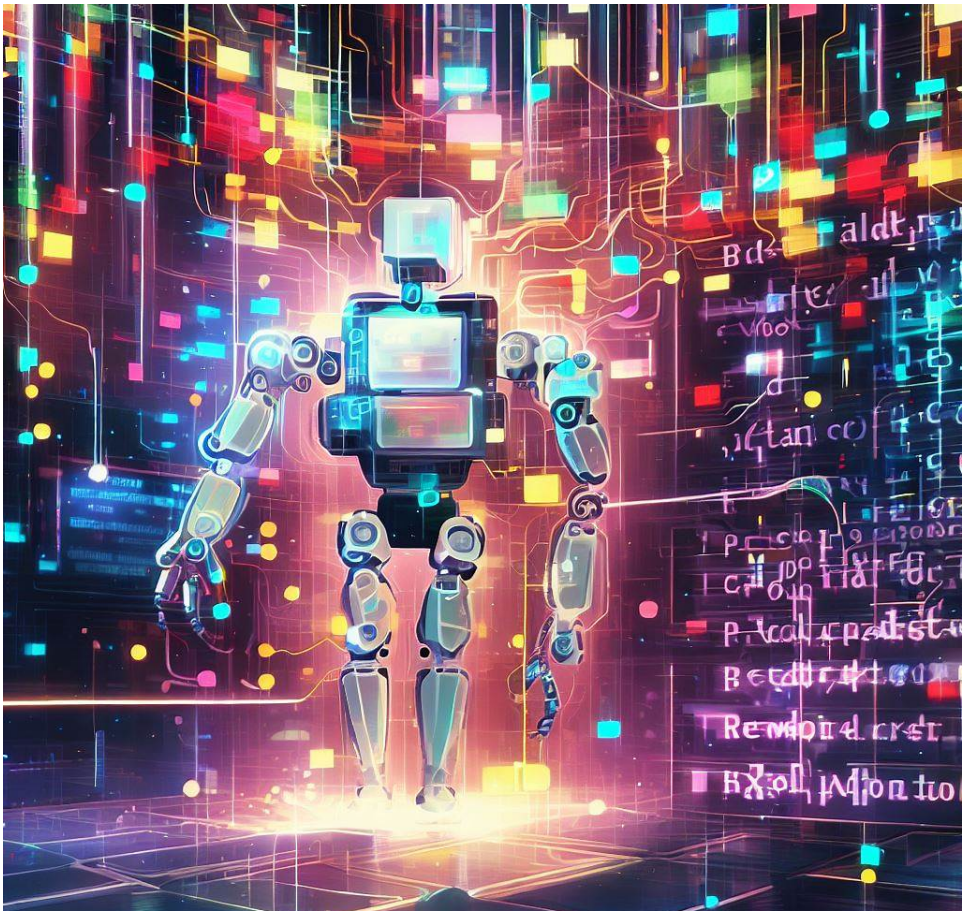
1.8 1.3.3 Compiling a Standalone Executable

1.3.3 Compiling a package (SPRE) Previous 86 Top 41 Next 92

Starting Guide

How to compile an Stand-Alone executable ...

Here is how you get your first executable running,

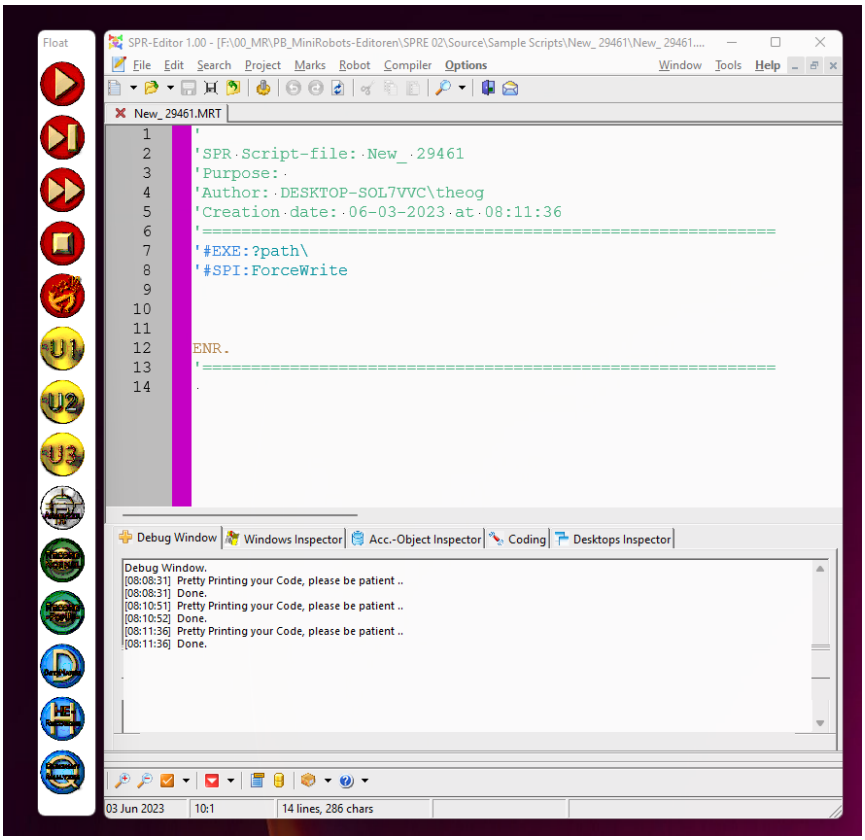


The SPRE is the most actual "Smart Package Robot Editor".

Step by Step:

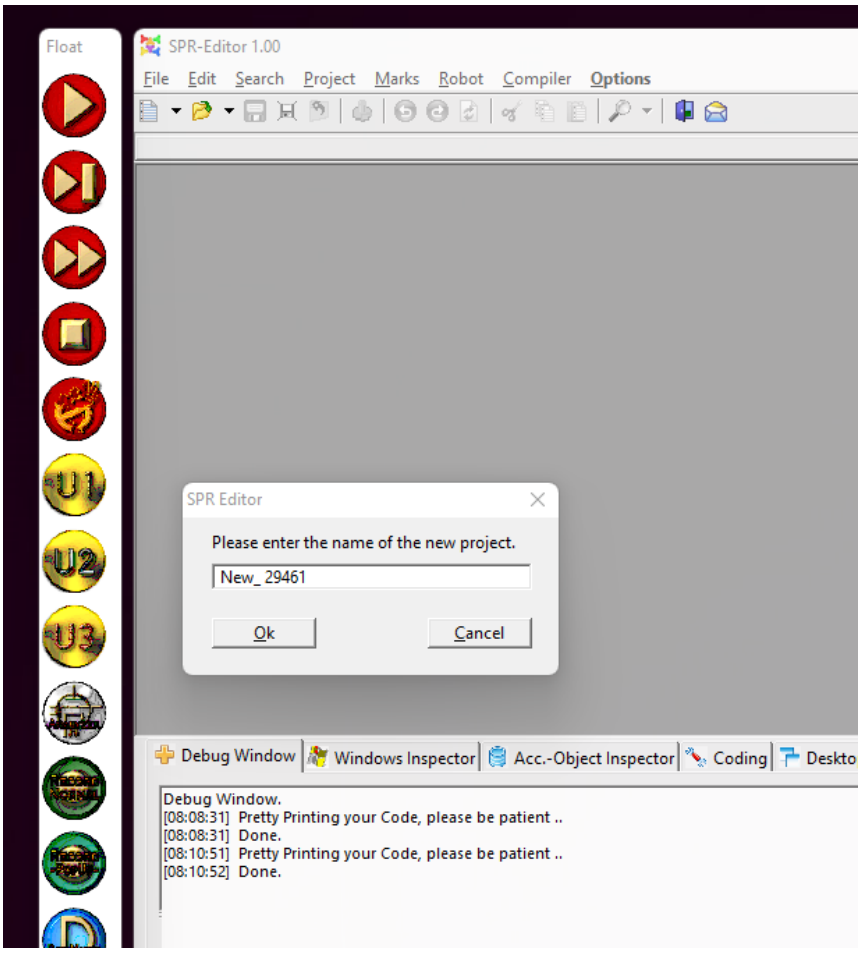
1. Start the SPR-Editor (SPRE) and select "New Project"

*© 2023 Theo Gottwald * https://www.smart-package.com*



2. In the window that appears, give your project a name, for example:

*© 2023 Theo Gottwald * https://www.smart-package.com*



3. Then click on "Ok".

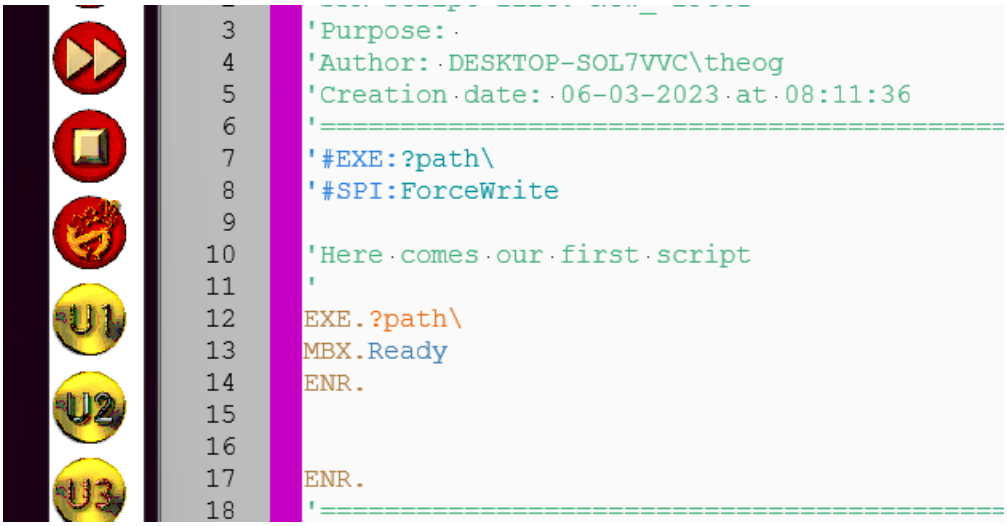
4. The Editor will automatically make an project folder for you, and open it. Looks somehow like this:

5. Now start typing your fist script

```
'Here comes our first script
'
EXE.?path\
MBX.Ready
ENR.
```

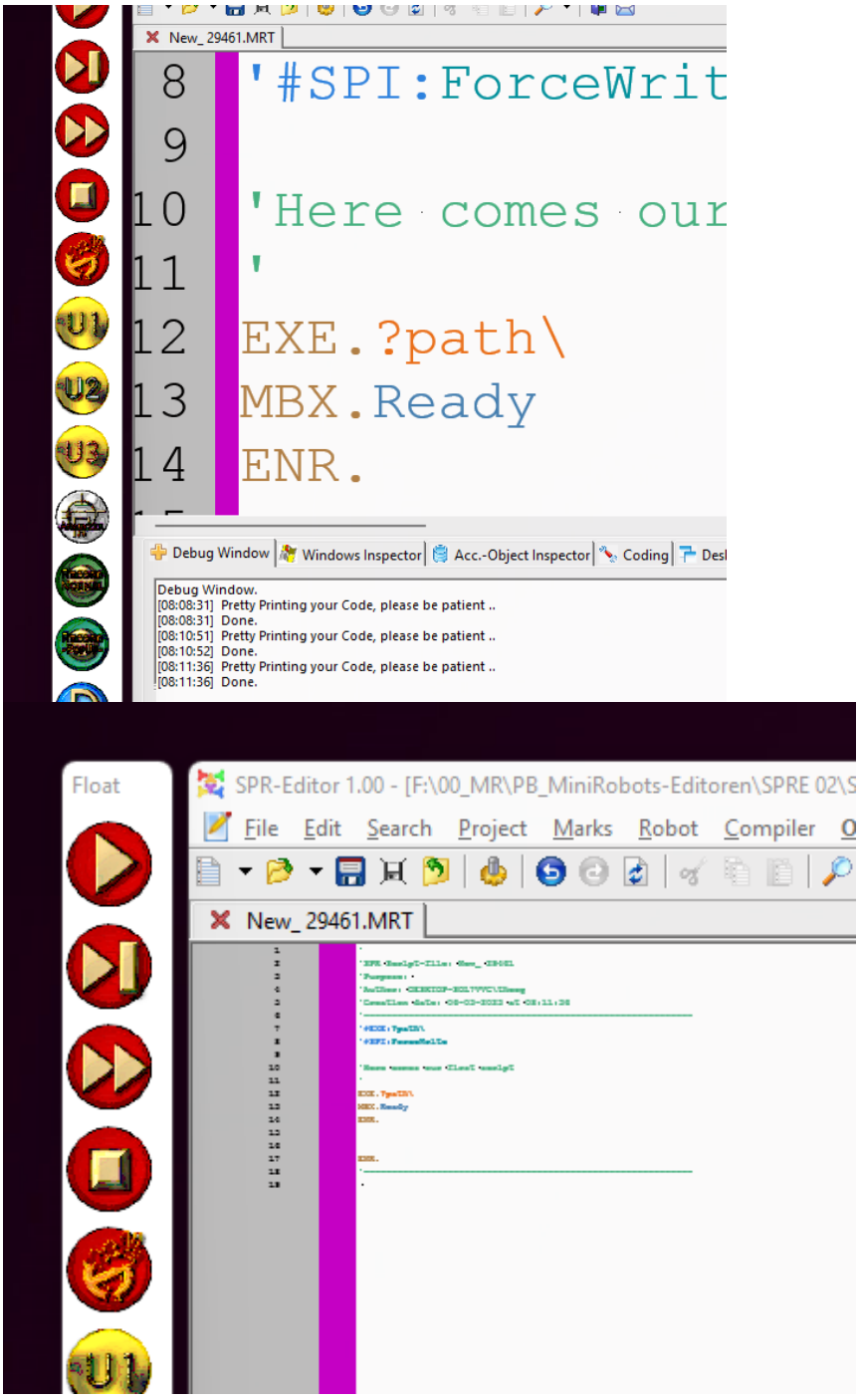
Which will look like this:

© 2023 Theo Gottwald * https://www.smart-package.com



6. If you press the left Ctrl Key on the keyboard (LCTRL) and rotate the Mouse wheel up or down,
you can change the size of the used Font.

© 2023 Theo Gottwald * https://www.smart-package.com

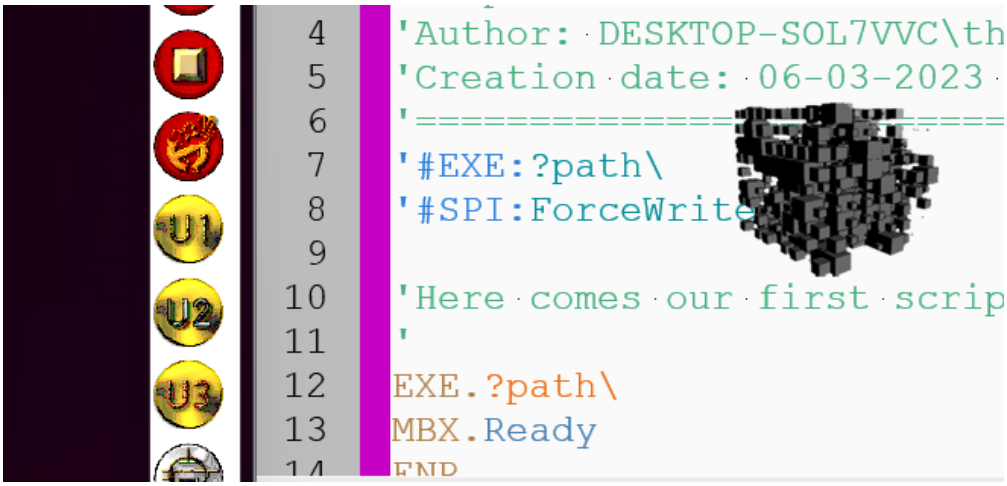


7. You may not exactly know at this time what each thing means, no problem. We'll take a look on this below at point 9.

At this time we just want to compile something into an executable.

8. Press the "Compile" Button in the Editor Toolbar.
You'll see a notice in the "Debug Window" of the Editor. Also there may be a compiler-animation in the Editor.
And you'll get an executable in your project folder. See pictures below.

© 2023 Theo Gottwald * https://www.smart-package.com



Note that the icon and the size of your executable depends on the version of this program you are using.

9. Let's first take a look on the script and see what it will do:

```
' This Line will open the path of the running script-file in
' an explorer window.
EXE.?path\

' This Line will show an Messagebox.
' Important: When you click or close the Messagebox,
' the temporary Executable Folder will also be closed/deleted.
MBX.Ready

' This command will just end the script.
ENR.
```

If you want more detailed Information about an command, just click on the command and then press the F1 Button.

10. Now we are ready to start our first executable. As you can see, there is a clear difference between the icons of the Scriptfile (ending with ".mrt" like "MiniRobots-Text") and the Executable/Package-Icon.

11. If you click on the Executable Icon twice, it will run and do what we expect.

12. Here is another way how you can compile an executable.

© 2023 Theo Gottwald * https://www.smart-package.com

Click on a Scriptfile (.mrt) with the right Mouse Button. You will see a menu somehow like this one. Now choose "Compile".

See also

- 1.0 '#RAU: - Run As User 271
- 1.3 '#SPI: - Special Instructions 287
- 1.2 '#EXE: - Set Executable Path and Name 277
- 1.5 '#INC: - Include Files and Folders into the Executable 290
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

1.9 1.3.4 Compiling a package (SPRE)

1.3.4 Compiling a package (SPRE) Previous 86 Top 41 Next 92

Starting Guide

How to compile an Stand-Alone executable Package

Here is how you get additional Files packed into your Executable

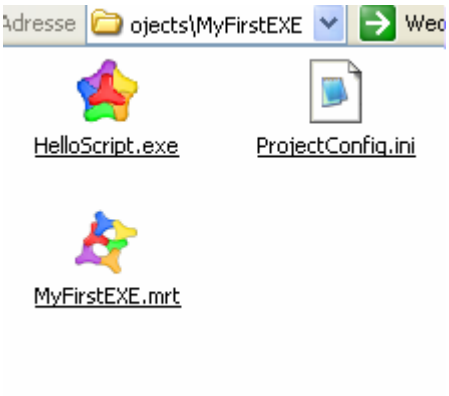
How do i include a folder into my package/executable?

You can easily include Setup-files into your executable.
Here is how to do this:

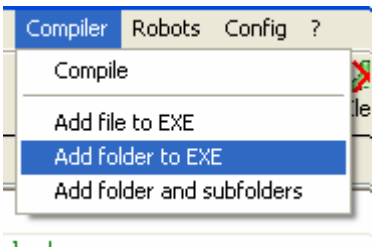
1. Create your project and Script as seen in the example above.

2. Copy the folder with the files that you want to include into the Project folder.
This look somehow like this:

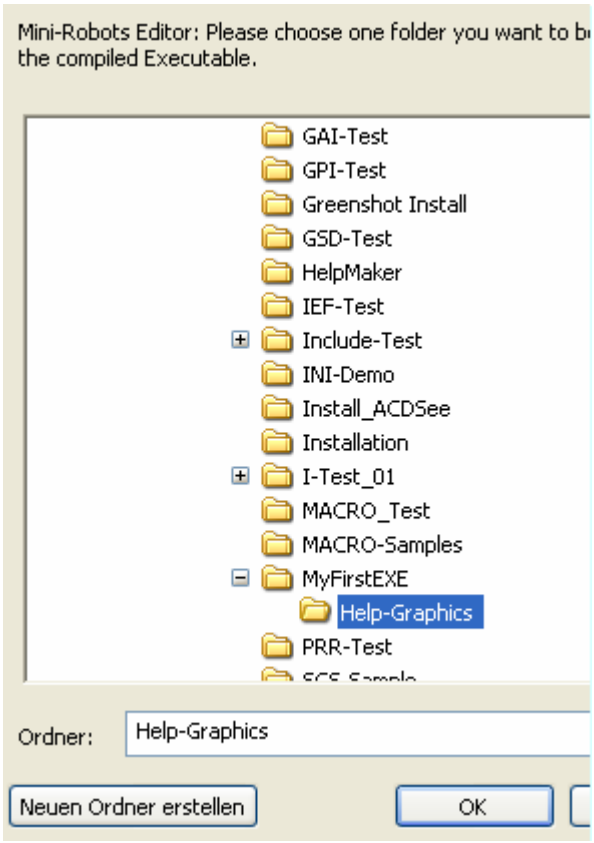
*© 2023 Theo Gottwald * https://www.smart-package.com*



3. Now click in the Editor Menu on "Compiler->Add folder to EXE".



4. In the appearing dialog choose the folder that we want to add to the package, and click the OK Button.



5. You will now get something like this:

© 2023 Theo Gottwald * https://www.smart-package.com

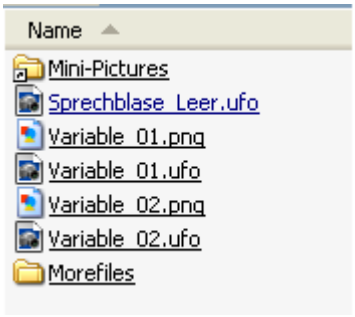
```
'-----------------------------
'Here comes our first script
'#INC:?pfad\Help-Graphics\
'#SPI:ForceWrite
'#EXE:?path\HelloScript.exe
EXO.?path\
MBX.Ready
```

The Editor has inserted this line:

```
'#INC:?pfad\Help-Graphics\
```

into your script. In fact this line makes the folder "Help-Graphics" that is in your project folder (=?path\) to be inserted into the executable.

Let me give you just a quick hint at this place. In case there would be sub folders in that folder, for example:



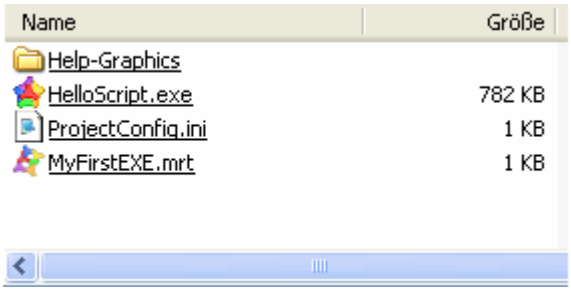
then these Sub-Folders will not be included, unless you add a "*" (asterisk) to the Line.
Like this:

```
' This will include folders AND all subfolders
'#INC:?pfad\Help-Graphics\*
```

First we'll leave the asterisk away and see what happens.

6. Press "Compile".

7. What wee see is, that the executable is a bit larger then before.

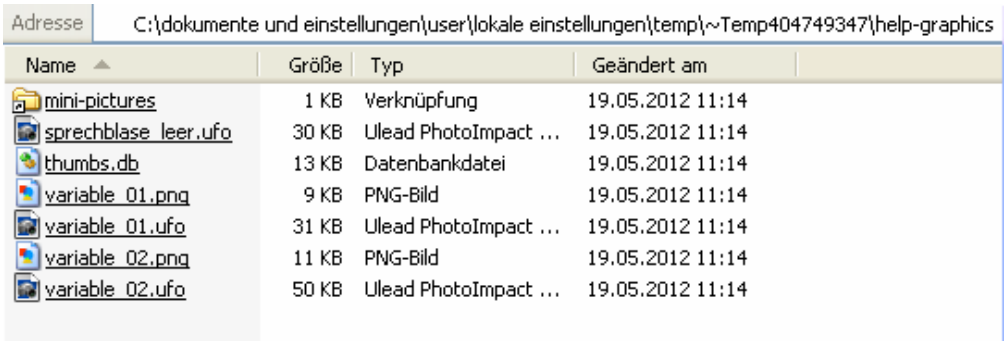


© 2023 Theo Gottwald * https://www.smart-package.com

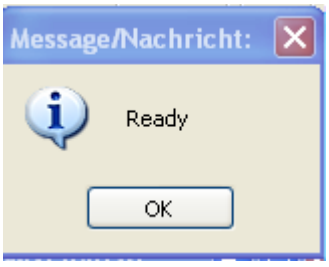
8. The package is larger because of the size of the additional files, we have packed into the package. Let's now start the package and see if the files are really in there!

9. Click the the "HelloScript.exe" twice to start it. A folder opens, somehow like this:

10. Enter the folder and take a look inside if it's all there.



11. It's all there - just the subfolder is missing, because we did not specify the additional "*" asterisk. We'll do that now. But first click on



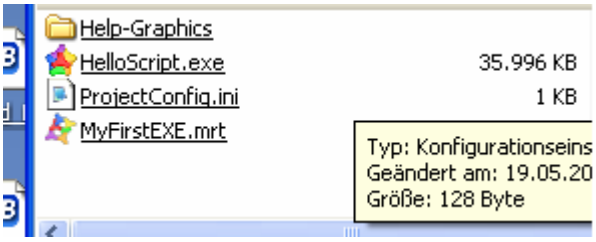
to make that folder disappear.

12. Now let's add that asterisk and re-compile our script:

© 2023 Theo Gottwald * https://www.smart-package.com

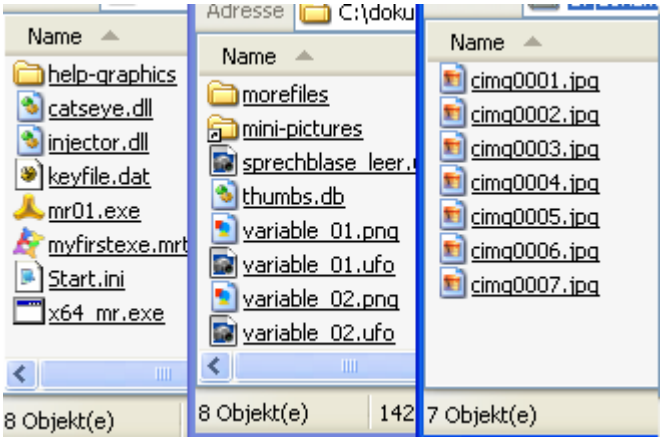
```
'Here comes our first script
'#INC:?pfad\Help-Graphics\*
'#SPI:ForceWrite
'#EXE:?path\HelloScript.exe
EXO.?path\
MBX.Ready
```

13. Press "Compile" again. Now the package got larger:



This time it got 35 MB. Don't worry the size limit for a single package is somewhere below 2 GB!

14. Now we'll start our package and see if anything gets unpacked properly.
Start the Package with a double click on "`MyFirstEXE.exe`".



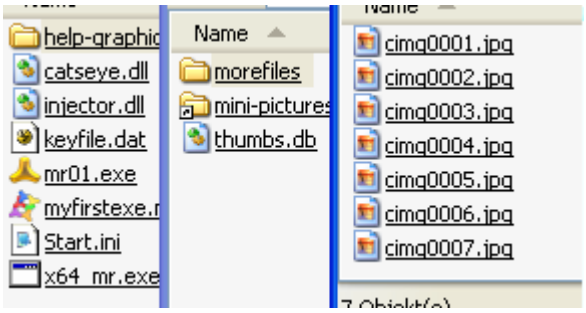
15. Everything is there as expected. Now let's get get a bit more deep into packaging.
We do not want to include those ".ufo" and those ".png" files into our package.

16. For this we can use the '`#EXC:` - Directive. Like this:

© 2023 Theo Gottwald * https://www.smart-package.com

```
'-----------------------------
'Here comes our first script
'#EXC:*.png|*.ufo
'#INC:?pfad\Help-Graphics\*
'#SPI:ForceWrite
'#EXE:?path\HelloScript.exe
EXO.?path\
```

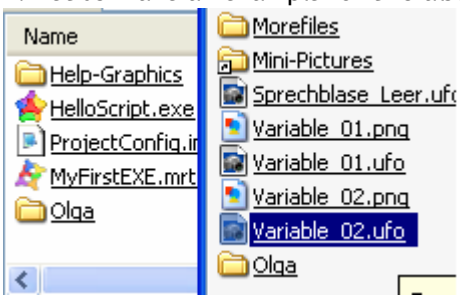
It works!



Hint: The `'#EXC:` - Directive must always before ALL Include Directives to work as expected. It will not influence inclusions before above itself.

Can I use that feature to exclude folders from the package also?

1. Let us make an example for this also. Here is our project-folder and a sub-folder:



2. Inside the folder "Help-Graphics" there is an folder named "Olga". We do **not** want to include this folder into our final package.

Take note that there is another folder named "Olga" in the Main Project folder. This folder shall be included with our package.

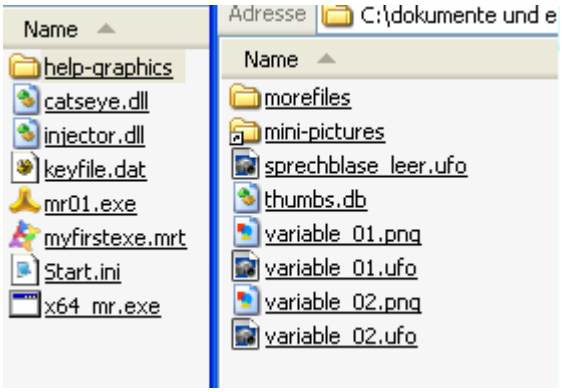
```
'#INC:?pfad\Help-Graphics\*
'#INC:?pfad\Olga\*
'#SPI:ForceWrite
```

Now comes the Exclusion. To exclude a folder, we can choose the name of the folder followed by a "\". We could simply write:

```
'Here comes our first script
'#EXC:olga\
'#INC:?pfad\Help-Graphics\*
```


© 2023 Theo Gottwald * https://www.smart-package.com

and here is what we get:



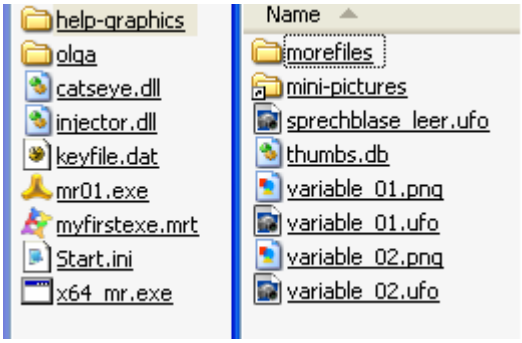
Good, the folder was not included in the package.

But look, the folder "Olga" from the Main-Project folder has also disappeared!
This way we told the compiler to exclude ALL Folders that have the name "Olga".

Now this is a special case, we'll solve it like this:

```
'Here comes our first script
'#EXC:Help-Graphics\Olga
'#INC:?pfad\Help-Graphics\*
```

And this is the result as expected.



Hint: You can use WildCards like "*","?" and "#" within the Exclusion Patterns.

Therefore even wildcard-patterns like this:

```
'#EXC:Help-*\Olga\
'#EXC:*elp-*\Olg?\
```

will do what you expect. The important point is that you do not forget the "\" at the end of the pattern, to signal that you are excluding a folder and not a file.

© 2023 Theo Gottwald * https://www.smart-package.com

See also

- 1.0 '#RAU: - Run As User 271
- 1.3 '#SPI: - Special Instructions 287
- 1.2 '#EXE: - Set Executable Path and Name 277
- 1.5 '#INC: - Include Files and Folders into the Executable 290
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

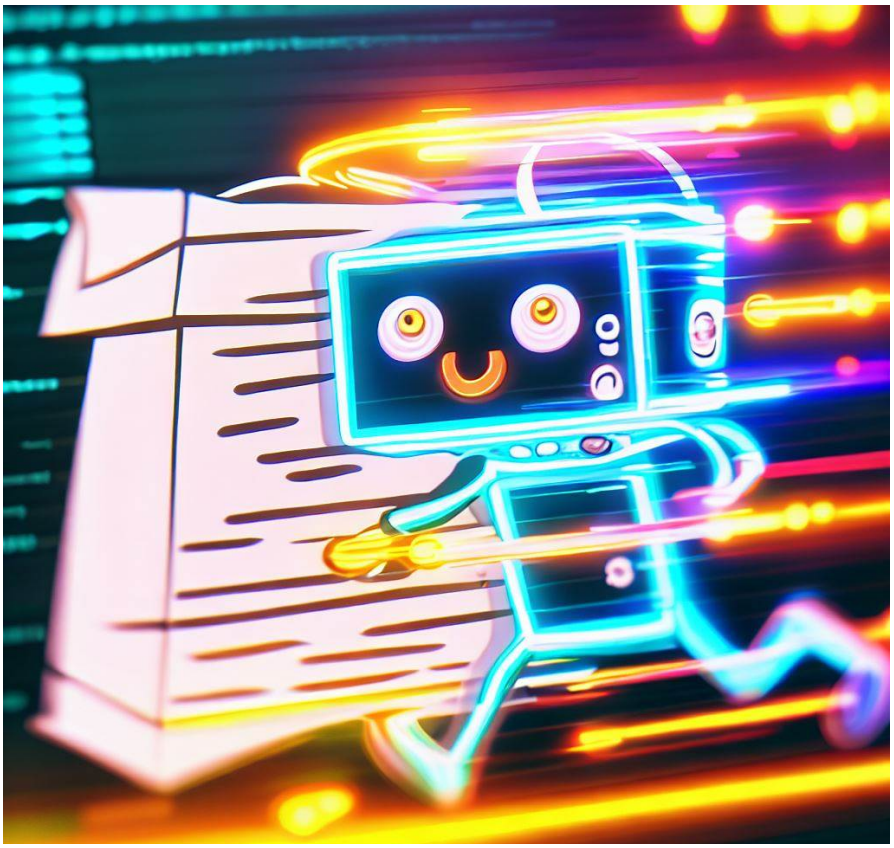
1.10 1.3.5 The Script Execution Process

1.3.5 Script execution Previous 39 Top 41 Next 73

Starting Guide

The Scripting Execution Process

How does the Robot make your Scripts run?



Understanding the Script Execution Process with the Smart Package Robot:

Creating a script, whether written or recorded, is only the first step in automating your tasks. The real magic happens when you initiate the execution process. Here's a simplified breakdown of what happens when you press the "PLAY" button in the Smart Package Robot (SPR):

Script Interpretation: As soon as you press "PLAY", the SPR springs into action. Its first task is to interpret your script. This involves understanding each command and preparing the necessary actions. Think of it as a director meticulously reading through a script before the curtain rises on a play. This careful preparation ensures that everything is in order and ready for execution.

Sequential Execution: Once the SPR has fully understood your script, it begins to perform each action in the order you've specified. This could involve moving the cursor,

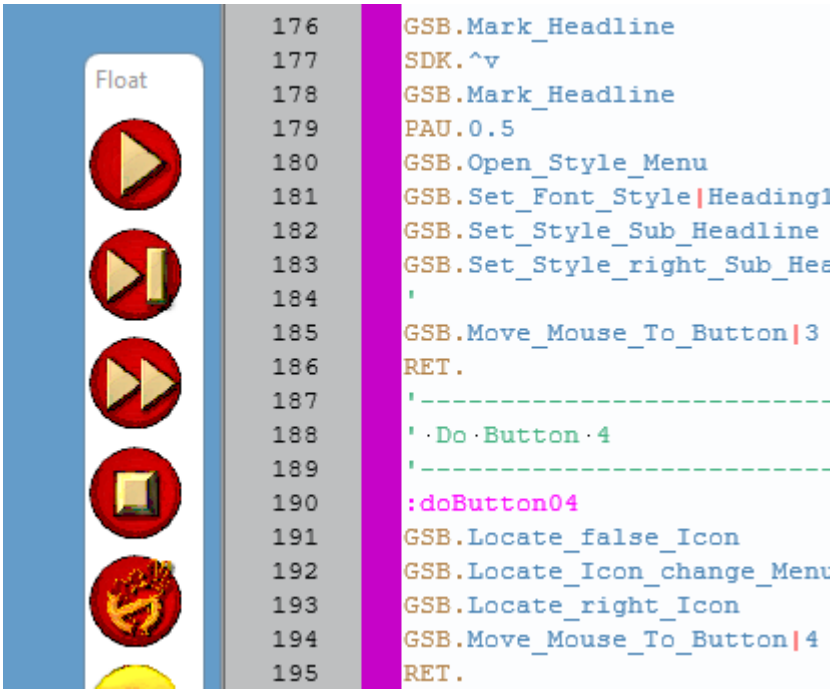
*© 2023 Theo Gottwald * https://www.smart-package.com*

clicking on buttons, typing in information, and more. The SPR maintains the flow and timing you've set, ensuring a smooth and efficient execution process.

Smooth Transitions: The execution process is akin to watching a well-rehearsed performance. Each action transitions smoothly into the next, creating a seamless automation process. This is where the SPR truly shines, as it replicates your actions with precision and accuracy.

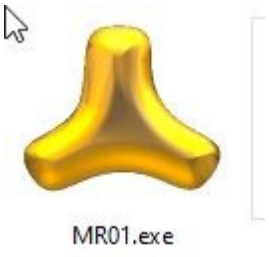
Accurate and Efficient Performance: With the SPR, you can be confident that your tasks will be performed accurately and efficiently. It's a reliable tool that takes the manual labor out of repetitive tasks, freeing up your time for more important matters.

Remember, the SPR is a powerful tool, but like any tool, its effectiveness depends on how well you use it. Take the time to familiarize yourself with its features and capabilities, and you'll soon be reaping the benefits of automation.



Diving Into the Execution Process:
When you press "PLAY" in the SPR-Editor, it sets off a chain of events.
The first thing it does is to summon our dependable friend, **MR01.exe**.
This file is the "Robot Executable", the brains of the operation.
Think of **MR01.exe** as the marionette master who pulls all the strings.
It takes your script commands, interprets them, and brings your Smart Package Robot to life, guiding it to perform the tasks you've outlined.
In this way, your script becomes a set of precise, actionable instructions that **MR01.exe** uses to orchestrate the Smart Package Robot's actions.

© 2023 Theo Gottwald * https://www.smart-package.com



Delving Deeper into MR01.exe
This is where the magic happens!
The **MR01.exe** is the heart and soul of your Smart Package Robot. Once you've hit "PLAY" in the SPR-Editor, the Robot Executable, **MR01.exe**, takes over.
It's responsible for translating and executing the majority, if not all, of your script commands.
It's as if **MR01.exe** takes your script and breathes life into it, making it dance to your tunes.
Your Smart Package Robot's actions are orchestrated entirely within the confines of **MR01.exe**.
This is why we often say that the magic of automation lives within **MR01.exe .**

6. 6. The Art of Preprocessing:

The first pivotal step in executing your script lies in the hands of the "Pre-Processor." This diligent component meticulously prepares your script, making it ready for the ensuing "Speedcode Compiler."
The Pre-Processor's role is multifaceted, focusing on refining your script to be as precise and streamlined as possible, thereby facilitating a smooth execution.

Unraveling MACROs:
One of the key tasks of the Pre-Processor involves deciphering MACROs. MACROs are concise code substitutes that represent a more extensive piece of code.
They help keep your script compact and more readable. However, improper definition or usage of MACROs can introduce errors. The Pre-Processor, acting as a vigilant guard, checks and resolves these potential discrepancies, ensuring a seamless run of your script.

Integrating #INC Files:
The Pre-Processor also takes on the role of a detective when it comes to #INC files. These files are external entities containing supplementary code or resources, such as libraries or images, which are essential for your script to function.
The Pre-Processor hunts for any missing #INC files and incorporates them into your script, guaranteeing that all necessary elements are in place.

Bug Squashing:
Next, the Pre-Processor assumes the persona of a bug exterminator. It scans your script for coding bugs - pesky issues that may arise due to typos, incorrect syntax, or missing punctuation, like semicolons.
The Pre-Processor rectifies these bugs, ensuring your script can perform its dance without stumbling.

Syntax Error Checking:
Finally, the Pre-Processor morphs into a proofreader, looking out for syntax errors that can't be automatically corrected. Syntax errors occur when the computer cannot comprehend the script due to issues like missing brackets or incorrect variable names.
If such errors are detected, the Pre-Processor not only identifies them but also suggests possible solutions for you to rectify them.

© 2023 Theo Gottwald * https://www.smart-package.com

Once the Pre-Processor has diligently performed all of these tasks, it passes the refined script onto the Speedcode Compiler. The Compiler then transforms the preprocessed script into machine code, a language that your computer can understand and execute. In this manner, the Pre-Processor plays a critical role in the script creation process, ensuring your script is as optimized and error-free as possible before it faces the Compiler.

7. The Speedcode-Compiler

In script creation, a significant hurdle lies in the time it takes to fetch a command. The Speedcode Compiler, a pivotal component in the script execution process, addresses this by transforming various elements of the script into a format that facilitates quicker execution.
This transformative process is commonly known as "converting to Speedcode."

Efficient Label Formatting:
The Speedcode Compiler begins its work by streamlining labels. Labels serve as milestones, marking specific segments in the script for easier navigation and comprehension.
By reformatting these labels into a more efficient format, the Speedcode Compiler significantly reduces the command fetch time.

Optimizing Conditional Statements:
Next, the Speedcode Compiler optimizes conditional statements.
These statements play a crucial role in steering the script's flow, determining which commands are executed and which are bypassed.
By reshaping these statements into a more conducive format, the Speedcode Compiler accelerates the command fetch process.

Loop Formatting:
Loops, which are employed to repeat a section of the script multiple times, are also a focal point for the Speedcode Compiler.
Improperly formatted loops can cause substantial delays. By remodeling loops into a more efficient format, the Speedcode Compiler curtails the command fetch time.

General Command Formatting:
Finally, the Speedcode Compiler reformats all commands into a format that expedites execution.
This includes any other commands not explicitly mentioned above, such as function calls, assignments, and more.

After the entire code has been converted to Speedcode, it is ushered to the next phase of the process.
This could involve further compilation, optimization, or direct execution of the script.
The overarching goal of this intricate process is to enhance the script's efficiency, thereby minimizing command fetch time and amplifying the script's overall performance.

8. The Script Execution

The Script execution is the grand finale in the script creation process.
It breathes life into your script.
This is the moment where the meticulously prepared script is let loose on a computer, with each command sequentially coming to life.

Just like an orchestra performing a symphony, the computer meticulously reads and performs each line of the script, usually from top to bottom.

© 2023 Theo Gottwald * https://www.smart-package.com

The maestro, which is the computer in this case, reads the first line of the script, interpreting and executing the command it contains.
Next, it smoothly transitions to the second line, performing the same process.
This sequential performance continues unabated, line by line, until the final note - or in our case, the final line of the script.

To ensure a flawless performance, the execution of the script is closely monitored.
If all elements align harmoniously, the script executes without a hitch, each command sequentially springing into action, leading the script to its intended conclusion.
Once the final command has been executed, the task at hand is complete, marking the end of the process.

To sum up, script execution is the culminating point in the script creation process, turning lines of code into a symphony of actions.
Like a well-rehearsed ensemble, the script, if well-written, will perform seamlessly from start to finish, bringing your digital masterpiece to life.

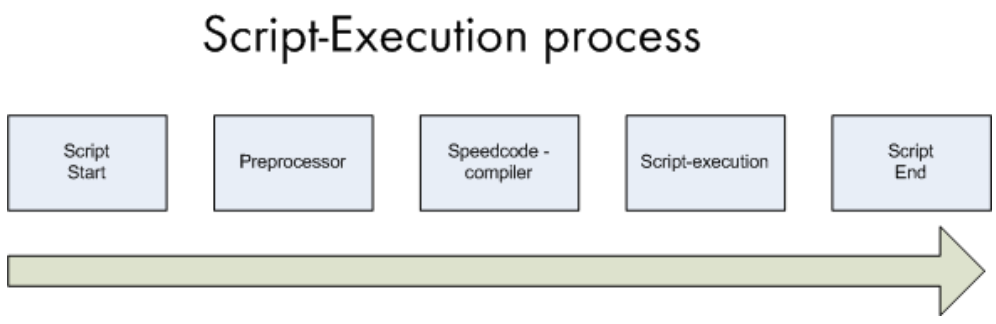
9. The Script End

In the grand theatre of script execution, the final act is often overlooked, but it's an essential part of the process: the cleanup phase.
This phase ensures a graceful and efficient exit, tying up loose ends by releasing the resources that were employed during the script's grand performance, such as memory, files, and other resources.

Picture this phase as the crew that enters the stage after the play has ended, tidying up the props and scenery.
Without them, the theatre would soon fall into disarray.
In the same way, the cleanup phase prevents memory leaks and other potential issues that could turn into major headaches down the line if resources aren't properly managed.

Now, if the script is a compiled executable, it doesn't just exit stage left without a word.
Instead, it leaves behind a parting gift: an error code.
This error code acts as a messenger, communicating the final status of the script's performance to the initiating process.
A standing ovation is represented by a value of zero, indicating a flawless execution, while any other number signals that there was a hiccup during the performance.
This error code can be used by the backstage crew (the initiating process) to figure out what went wrong and how to fix it in future performances.

In conclusion, the script end, or the cleanup phase, plays a vital but often underappreciated role in the execution process of a script. It ensures a tidy and efficient conclusion, diligently releasing resources, and, if it's a compiled executable, providing valuable feedback through an error code.



© 2023 Theo Gottwald * https://www.smart-package.com

1.11 1.4. The Prototype Parameter System

Starting Guide

The Prototype Parameter System

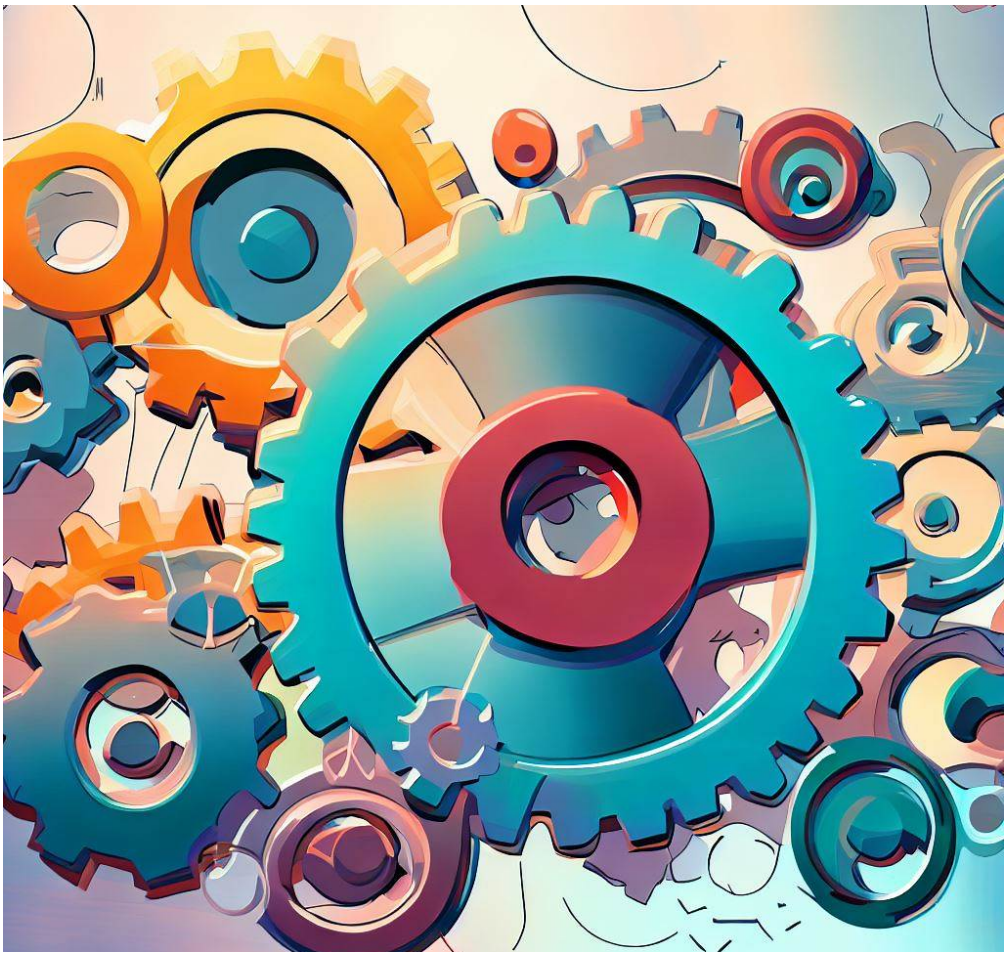
What are all these letters and parameters good for?

Intention

Let's imagine you're a puppeteer and you have a robot puppet.
To make your puppet do things, you need to tell it what to do using a special language called Robot-Script.
It's a bit like playing Simon says, but with a robot!

What is a Parameter?
First, let's think about how you might tell the robot to look for a button that says `"Next"`.
You could tell the robot using a command like this: `%Look_For_Button "Next"`.
In this command, `%Look_For_Button` is the action you want the robot to do, and `"Next"` is the detail about the action, which is called a **parameter**.
A parameter is like a piece of information that you add to the command to make it more specific.
So instead of just saying "Look for a button", you're saying "Look for a button that says `'Next'`.

© 2023 Theo Gottwald * https://www.smart-package.com



How can we handle more then one Parameter?
Sometimes, you might want to give the robot more than one piece of information at the same time.
For example, you might want to tell it to look for a button that says "Next" and is also a certain color.
You can do this using the pipe-symbol "|", like this: `%Look_For_Button "MyClass"|"Next"`.
In this command, "`MyClass`" and "`Next`" are both parameters.

What is the "Normal Parameter System"?
Just separating Parameters with a pipe-symbol "|", like this: `%Look_For_Button "MyClass"|"Next"`.
This is called the "**normal parameter system**". Each Parameter has a specified position that also tells the robot "what it is good for".

What is the "Prototype Parameter System"?
But sometimes, you might want to give the robot lots of pieces of information at the same time.
If you try to do this using the normal parameter system, it can get confusing because you have to count the pipe-symbols.
That's why we have something called the "**Prototype Parameter System**".
It's like a smarter way of giving the robot lots of pieces of information at the same time.
Let's see how it works.

*© 2023 Theo Gottwald * https://www.smart-package.com*

First, you need to know exactly what information you need to give to the robot.
For example, let's say you want to tell the robot to look for a button that says "Next" and is a certain color and is in a certain place on the screen.
You can do this using the prototype parameter system, like this:

```
SCW.t|Next
```

In this command, SCW. is the action you want the robot to do, t is the prototype parameter that tells the robot you're going to give it some text, and "Next" is the text you're giving it.
If you want to give the robot more than one piece of information at the same time, you can just add more prototype parameters and pieces of information, like this:

```
SCW.ct|button|Next
```

In this command, ct is the prototype parameter that tells the robot you're going to give it a class and some text, "button" is the class you're giving it, and "Next" is the text you're giving it.
The cool thing about the prototype parameter system is that you can give the robot any combination of information in any order, as long as the order of the prototype parameters matches the order of the pieces of information.

Many commands use the Prototype Parameter system. Let's go step-by-step into this.

What is an parameter, anyway?
Commands need parameters. For example, if we look for a button with the text "next" written on it, we could possbly write:

```
%Look_For_Button "Next"
```

In this case the command would be "%Look_For_Button" and "Next" would be the parameter. This is not a real command of our system, we just take it as an example.

What is a parameter? It's a part of a complete command.
The command is fixed and the parameter is the changing part of it.

The "Normal" parameter system:
Some commands use the normal parameter system. If we have more parameters, we could do like this:

```
%Look_For_Button "MyClass"|"Next"
```

In this case we have separated the Parameters using the "Pipe-Symbol" | .
This shows the Computer where one Paramter ends, and a new Parameter starts.
Also, if we do not need a Parameter we can just omit the Parameter. Like this:

```
%Look_For_Button |"Next"
```

This "normal Parameter System" will work up to a number of lets say 6 parameters.
Then it will start getting annoying. Because you have to count the delimiters.

© 2023 Theo Gottwald * https://www.smart-package.com

```
%Look_For_Button |"Next"|||
```

If you need 20 or more possible parameters, you may get totally lost with the normal parameter system.

```
%Look_For_Button |"Next"|||1||||||5
```

To solve this problem, we have developed the "Prototype Parameter System".
TO use it, we need to first answer the question:

Which parameters do i really need?

Not all Parameters are needed at any use of the specified command.
If we know which Parameters we do really need in our case, we can specify only these parameters.
The Prototype parameter system solves this issue with the prototype-field. Its always field **[P1],** directly after the command.

CMD.[Prototype-Field]|[Data defined by Prototypes, separated by "|"]

The prototype field contains some letters which are the "prototype" (model) of the following parameters.
The prototype field tells the robot which parameter has which meaning. Before it gets too technical lets see a simple example.

The solution: a prototyped parameter system.

Here is an Example of the prototype parameter system:

```
SCW.t|Next
```

In this case the prototype-field contains just the letter "t". If we look in the description of the command, we see that "t" is the "control text"-Parameter.
If this Parameter is used then the Script will search for a button (or anything) with the text "Next" on it.

This way the robot knows what he should look for. But what we need is not "anything" we want to search for a control with the class "button". Like this:

```
SCW.ct|button|Next
```

In this example we have two letters in the prototype field.
First a "c" - which is the "control class" in our case "button".
Second a "t" which is the control text. In our case "Next".

We could write the technically exactly same command like this:

```
SCW.tc|Next|button
```


© 2023 Theo Gottwald * https://www.smart-package.com

This is absolutely the same command.
The order of the prototypes does not play a role, other then the order of the Parameters must match the Order of the Prototypes.
Some prototypes need an extra information on the right side, like "c" and "t". Because you need to tell the robot which "text" or "class" you are looking for. But then there are prototypes that just tell the robot "move the window to the desktop center".
They do not need an extra parameter. It's just enough if they are there.
Here is an example which uses several prototypes. Doing so, the final Skript can be very short.

```
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' U ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
' l ... center window on desktop
STW.wtUXl|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
```

This Script, which has effectively just two lines of code and a Label, will test if a Program is already been started, and if not, start that Program.
The whole functionality in just two lines of code!

Here we have two prototypes that do not need an extra parameter on the right side. In this case we have the "X" and the "l". If given, they just flag something to the command.

```
"X" - continue without errormessage if the window was not
found in time
"l" - center window on desktop
```

Please note that **the meaning of the prototype letters are command dependent**. For example, in `GDF`. the prototype "f" means something else then the prototype "f" in `SCW`.
Also **Prototypes are case-dependent. So always use the right case for a Prototype!**

Prototypes help you to control the behavior of the command. Some need additional parameters and some do not.
The sequence of the prototypes must fit tot the sequence of parameters. Then the robot knows which parameter belongs to which prototype.

Why do we use the prototype system?

© 2023 Theo Gottwald * https://www.smart-package.com

Some commands have a lot of possible parameters. If we would use a fixed sequence of parameters, you would end up with making commands like this:

```
SCW.|||MyClass|Myytext|||||||1|||5
```

We think this is not acceptable for complex commands. Using the prototype system you can avoid such obstacles and write only those parameters that are needed.

The important sentence on the prototype parameter system
While the sequence of the prototypes themselves is mostly unimportant, the sequence of the following parameters must be matching the sequence of their prototypes perfectly. However there are exceptions.

The parameters may have a priority, which is from left (lowest) to right (highest)
In few cases, where prototypes have actions that can not be applied together, there are special rules.
For example, with the `STW.` command, the prototype "A" switches of the prototype "C" internally. If both are given, the last one beats the first one. See the examples below.

In this case the "A" Flag has priority over the "C" Flag, because its right of it.

```
STW.tCA|explorer
MMV.
```

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
STW.tAC|explorer
MMV.
```

Remarks

Most often variables are allowed in the prototype field, but this may not yet be the case for all commands. The command:

```
%Look_For_Button "Next"
```

Is not a command in our system. You can however define it for yourself, using the MACRO-Feature- For example like this:

```
: %Look_For_Button 1
SCW.ct|Button|§§§01
END%
```

Note that you have to select a Top-Window (Application-Window) first using `STW.`

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.5.1.2. Using Variables 92
- ! STW. -Search Top-Window 3814
- ! SCW. - Search-Child-Window 3738

© 2023 Theo Gottwald * https://www.smart-package.com

1.12 1.5. Speed Considerations

Starting Guide

Speed Considerations

Some general tips for using the Script language,

Get a Speed Dump

Mostly speed is not an issue for our tasks. But sometimes it is, and then you will be quite happy with your robot.
Because **he's really fast**.

If you want to see, how fast, this is your chapter.

How fast is the Robot?
On my Computer the Robot can process **~16.000.000 Cycles in one second.**
Each command needs a certain amount of such Cycles. For example an Empty Line or a Commented-Line needs 1 Cycle.
A Line with a `FOR`.-Loop may need 51 Cycles (see below). A `NEX.` may need 26 Cycles.

© 2023 Theo Gottwald * https://www.smart-package.com

The Speed Dump shows you how many Cycles each Command has needed.

We have just seen that we can easily measure, how long time a script needs from start to end.
But how much time does each line in the script need?

You can get the timing of each line of your code, using a speed-dump.
To get it, we will just add a single line to our code from above.

```
' Start our stop-watch
VAR.§§TIM=#dtime#
' Init counter variable
FOR.§§CNT|1|50
' Print the actual number
  PRT.§§CNT
NEX.
' Get the time how long we needed.
VAR.§§TIM=#dsince#
' This is the Speed Dump.
DMP.
' We need this kine, or else the Speed Dump will immediate
MBX.§§TIM
ENR.
```

In the picture below is the resulting speed dump i get here on my system. Let us take a close look on what all these numbers mean.

On the Left side you see the **Line-Number**. This Line number is not always identical with the Line number in the Editor. While in most cases the line numbers are identical, however if you have expanded Macros, they will not be identical.

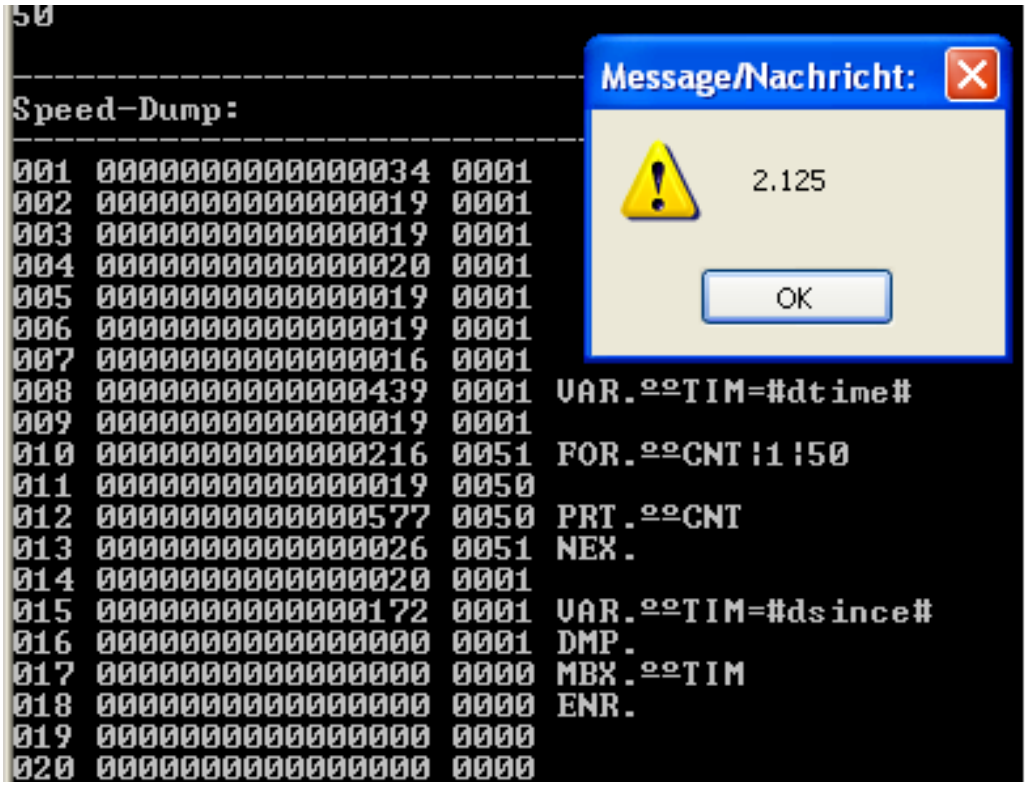
Next is a large number with a lot of zero's. That is the CPU "**cycle counter**".
This is not in seconds or milliseconds, its the amount of work the CPU had to do with your statement.

How much is it in seconds?
On my Computer the Robot can process **~16.000.000 Cycles in one second.**
But anyway, the higher the number - the longer it takes!

Next comes the "**Line execution counter**" a field that shows you, how often a line was executed.
As our loop was from 1 to 50, all lines inside the loop were executed 50 times. Thats why there is a 50 in that field.

Lastly you can see **the code of that line**.

© 2023 Theo Gottwald * https://www.smart-package.com



Does it only count to 9999 ?
No, it doesn't.
As you can see, the code execution counter shows only 4 zeroes. What will it do **above 9999**?
The answer is: It will just continue counting.

How fast is this?
But, how long does it take to count to ... for example **100200**?
The answer is, it need just **a bit more time then a second.**

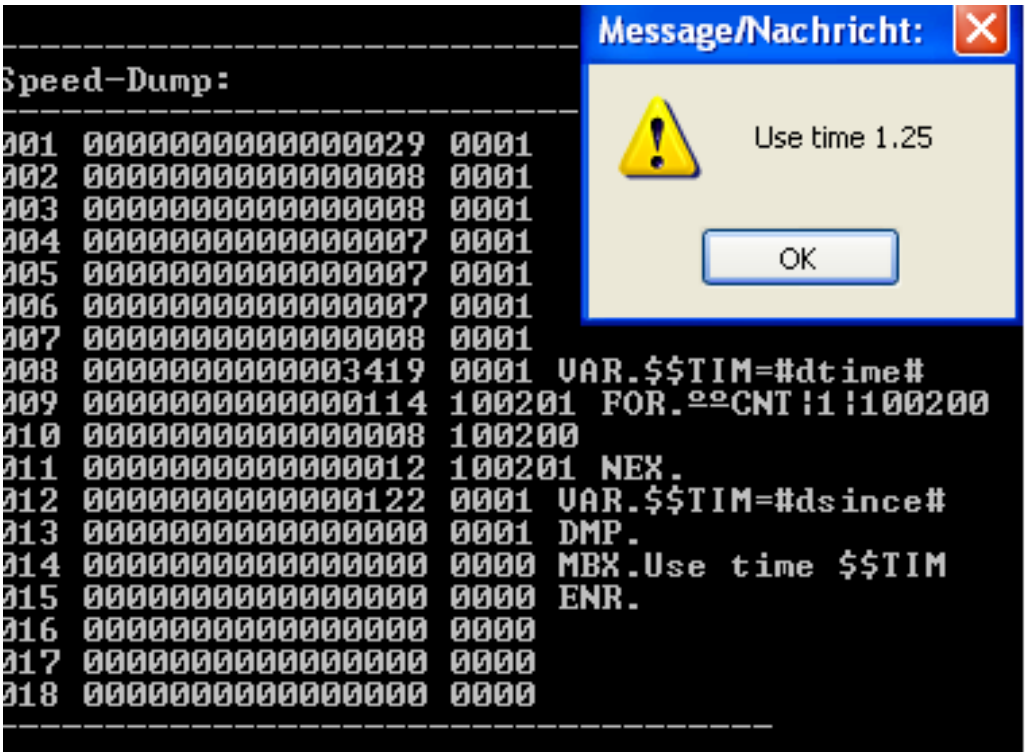
However you need to **run the script as a compiled executable.**
If it would be running in the editor, it would take too long time.

This is the script:

```
VAR.$$TIM=#dtime#
FOR.$$CNT|1|100200
  'PRT. $$CNT
NEX.
VAR.$$TIM=#dsince#
DMP.
MBX.Use time $$TIM
ENR.
```

And here is the result:

© 2023 Theo Gottwald * https://www.smart-package.com



Maybe we should have called it "Speed Robot" because it is really fast. This speed will help you to use it even for tasks you have never thought about before.

Sort those 100.000 Names ... check Files for errors. Test applications and create Log-Files.

Whatever it is, your robot is ready to do it with you.

© 2023 Theo Gottwald * https://www.smart-package.com

One more word about speed

One of the key features of the "Speed Robot" is its optimized use of CPU caches, which significantly increases its speed and performance. CPU caches are small, fast memory banks that store frequently used data, allowing the CPU to access it quickly. By utilizing these caches, the Speed Robot can access the data it needs more efficiently, resulting in faster execution times.

However, it is important to note that the initial execution of a command may be slower compared to successive calls of the same command. This is because the code needs to be loaded into the CPU cache for the first time, and potentially out of Windows virtual memory. Once the code is loaded into the cache, subsequent calls to the same command can be executed more quickly.

This optimization is especially beneficial for tasks that need to be performed repeatedly, such as processing large amounts of data. The more times the task is repeated, the more the robot's speed accelerates, making it an ideal tool for these types of tasks.

When testing the Speed Robot's performance, it's important to keep in mind that the best results will be achieved when the commands are already in the cache. This can be achieved by executing the commands once within the same script. This is because once the code is loaded into the cache, the CPU can access it more quickly, resulting in faster execution times.

In summary, the Speed Robot is optimized to use CPU caches, which increases its speed and performance. The initial execution of a command may be slower, but the

© 2023 Theo Gottwald * https://www.smart-package.com

robot's speed accelerates with each repetition, making it an ideal tool for tasks that need to be performed multiple times. To get the best results, make sure the commands are already in the cache by executing them once within the same script.

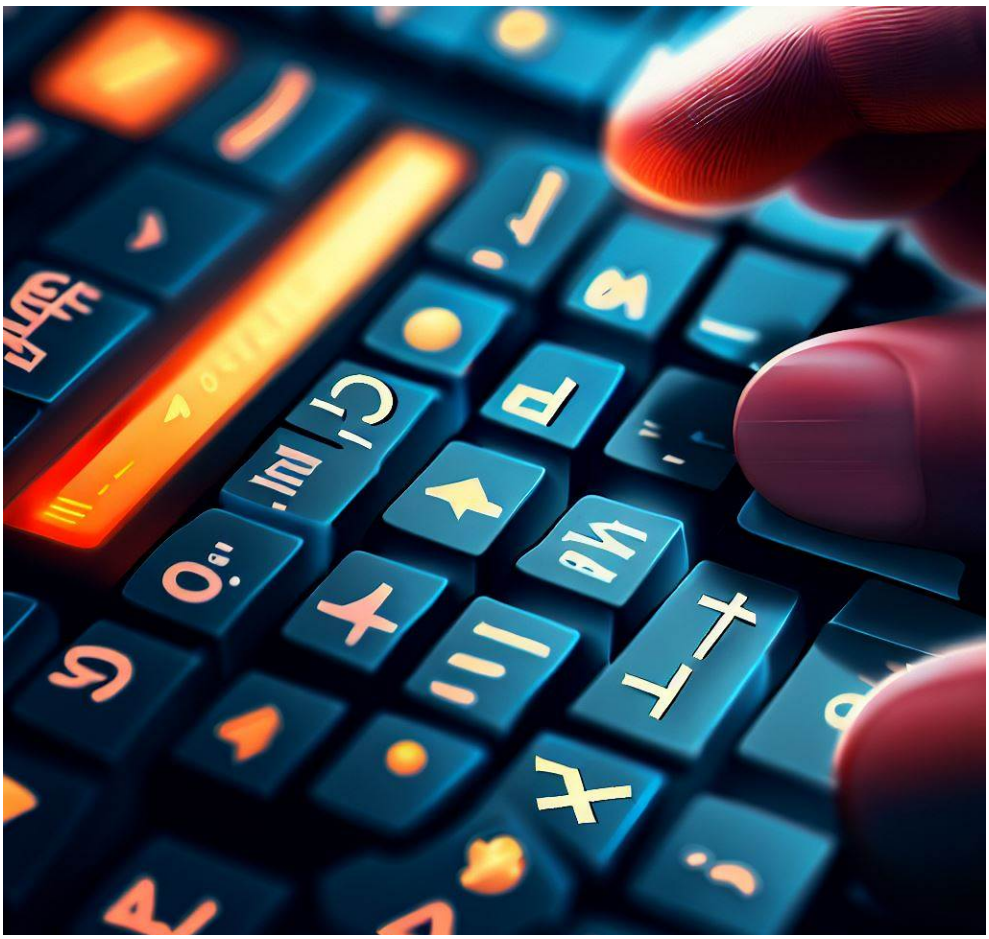
© 2023 Theo Gottwald * https://www.smart-package.com

1.13 1.5.1 FAQ - How to ...

Starting Guide

FAQ - How to do several things ...

More specific tips for using the Script language,



How do you get the § and the |-Character on non-german Keyboards?

Press "**ctrl-B**". The Editor will give you the **§** character.

Pressing "**ctrl-H**" will give you the **|** (Pipe) Character.

© 2023 Theo Gottwald * https://www.smart-package.com

How to read text from controls

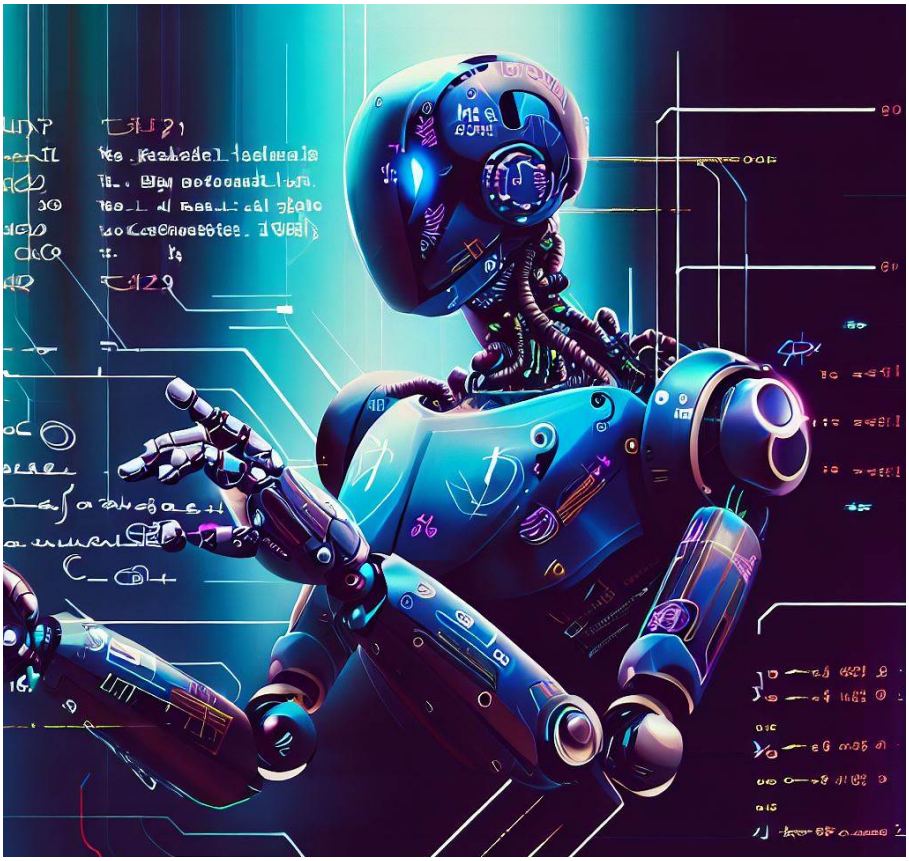
To read text from windows, use `CFW`.

```
STW.ct|PBWindowClass:0|TestApp
SCW.nc|1|SysIPAddress32
CFW.$$TXT|u
```

If `CFW.` does not work, or if you want to read from within "Browser" or Office-Programms, try `GAI`.

---

*© 2023 Theo Gottwald * https://www.smart-package.com*



Numeric parameters and formulas

Wherever you have numeric parameters, you can also use a formula. For example:

The command `PAU.` want a numeric parameter first. We could write:

```
' The script will Pause for 5 seconds
PAU.5
```

We can also use a variable here:

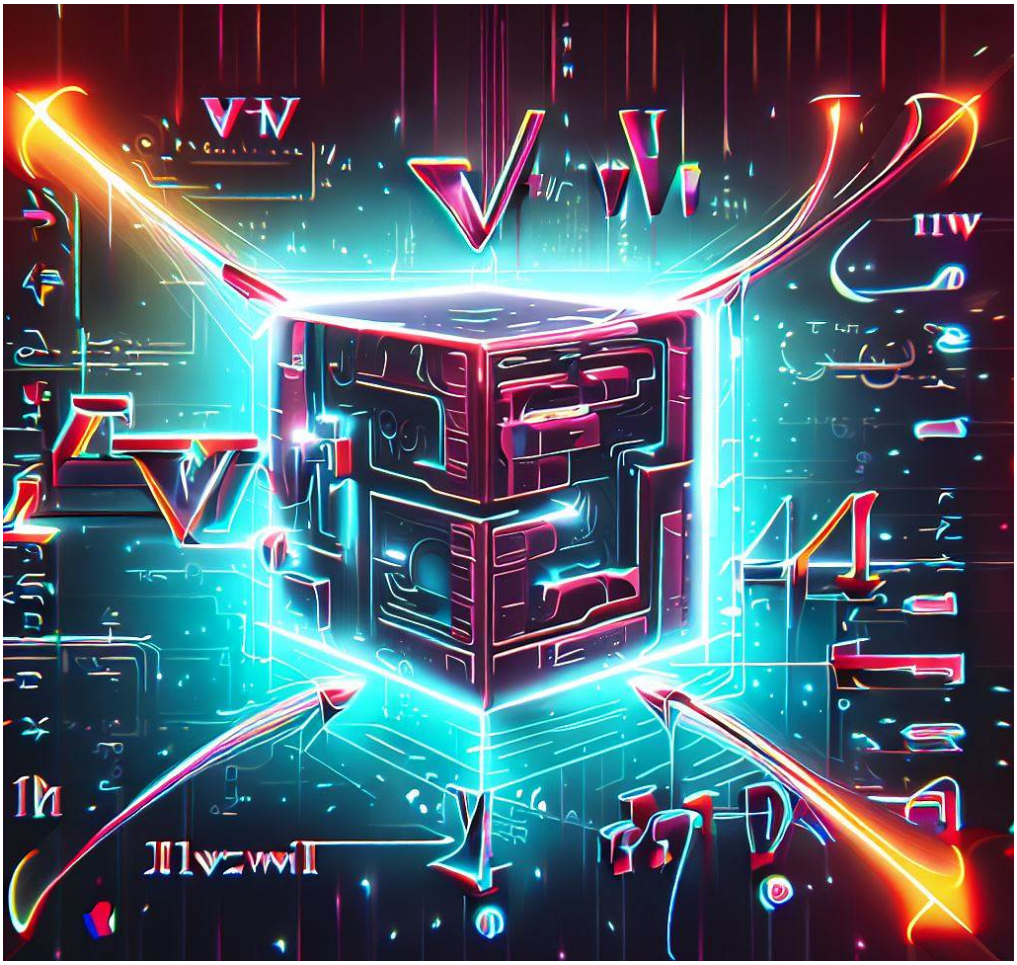
```
' The script will Pause for 5 seconds
: §§TIM=5
PAU.§§TIM
```

Now we can also use a formula here. The trick is, that we must put the formula into brackets (...).

```
' The script will Pause for 5 seconds
: §§TIM=1
PAU.(§§TIM*5)
```

You can use formulas at most places where numeric parameters are valid. The formula can use all expressions that are valid for the CAL.-command also.

© 2023 Theo Gottwald * https://www.smart-package.com



How to use formulas in decisions

There is the mathematical `IVV.` It is formed like this:

`IVV.` (Any Formula like in `CAL.`)

The brackets (...) are needed so I`VV.` knows that you want the mathematical `IVV.`

The result is tested against zero to decide whether the condition is true (if it is non-zero) or not true (when it is zero).
You can use formulas with all operators from `CAL.`, if you include them into (...).

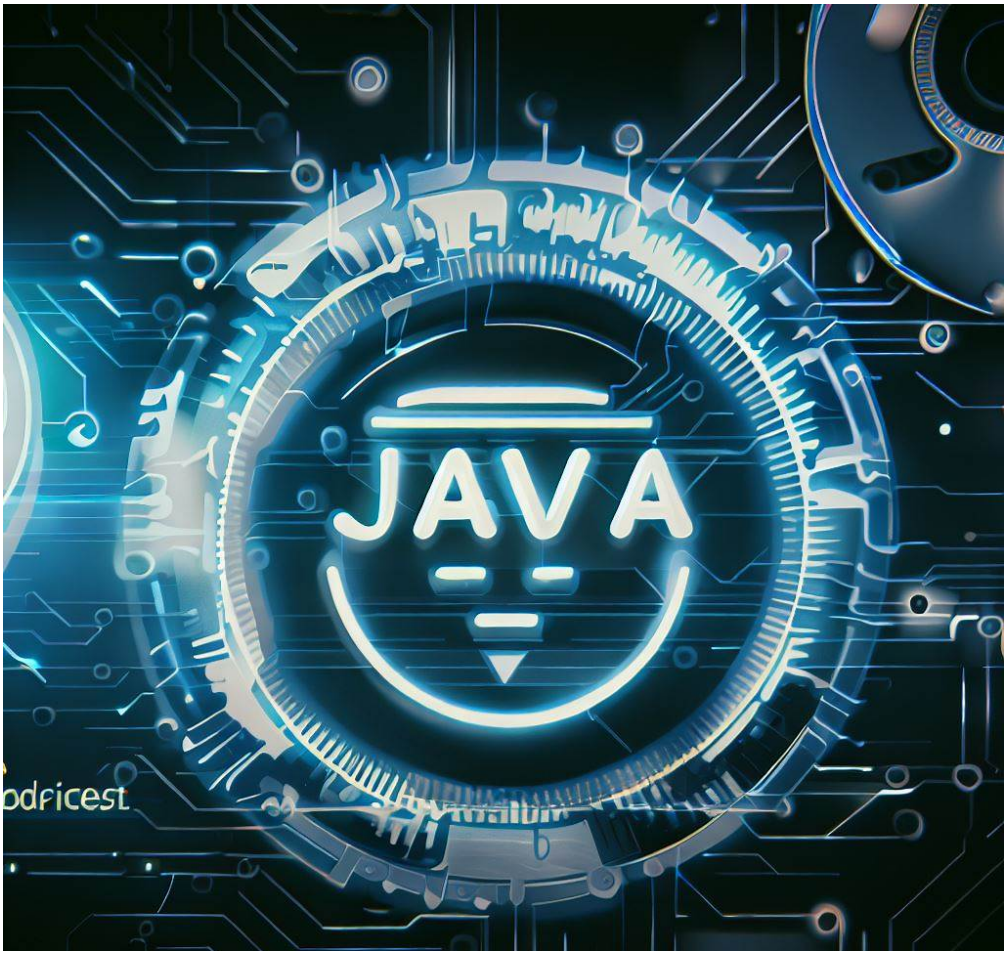
In that case the condition is true, when the result is NOT zero.

```
' This will make an OR'ed decision
IVV.($$MAA<9)+($$MAB>0)
' Do something if result is NOT zero
ELS.
' Do this if result is zero.
EIF.
```

You can also use a Syntax with delimiters "|". In this case you can use formulas on both sides as you wish.

```
'
IVV.($$MAA*5)|>|(8*1+0.5+$$XYZ)
```


© 2023 Theo Gottwald * https://www.smart-package.com



How to use the SPR with "Java-type" Setup programs

If you see that the `MEL.` command does not click anything and the window-class is "`Sunawt ...`" you are going to automate an "Java-type" Application.
Now the first step is, use only Compatible-Engine Commands like `MLC.` or `MLI.` for clicking on Buttons.
You may need to use the "Pattern Technology" (`ANA.-Command` etc.) or simple Mouse-Positions to locate Buttons.

Second, use Window-relative coordinates to locate the buttons.

If you are a more experienced user, use the Quadrant coordinate System and the `QFR.`-command.
This will scale your clicks when the User has a larger font size.

Third, there will be moments when the Setup is working, and your Script has to wait.
Of course you can simply use `PAU.` Command. But this is not the perfect solution.
With the SPR, use `RRA.` and `WRA.` instead.
These commands will react on "changes" in the Setup-Programs windows.
This way you can see when the Setup has finished his Job.

© 2023 Theo Gottwald * https://www.smart-package.com

Also there are quite a number of Pixel-Color Operations. You can use these to find out, if a Checkmark is set or to make more complex operations.
Also use Quadrant-Coordinates here to make things "Scale" when the User has another font-size.
To check which "DPI" is set on the local system, use `IDP.` and the `DPI.` command.

© 2023 Theo Gottwald * https://www.smart-package.com

1.14 1.5.2 Using Variables

1.5.1.2.1 Using Variables **Previous** 60 **Top** 41 **Next** 104

Starting Guide

Using Variables

Update: Due to the new `SAV.Save` and `SAV.Restore` Command the use of `$$XXX`-Variables (Script-local Variables) is not more recommended.
Just use the normal `$$XXX` (Global Variables) and make them "local" using the `SAV.`- and `SAV.Restore` Command.

What are Variables?

Variables are like containers that hold important information in a program or script. They help us keep track of different kinds of data, such as numbers or words. Just imagine having different boxes to store different things.

Variables are good because they allow us to:

1. Remember things: We can give a name to a variable, like "myNumber," and put a number inside it. Then we can use that number later in our program whenever we need it.

2. Change things: We can update the value inside a variable. For example, if we have a variable called "score" and we win a game, we can increase the score by adding more points to it.

3. Solve problems: Variables help us solve puzzles or perform calculations. We can use them to add numbers together, find the average of a group of numbers, or even create stories with different characters and their actions.

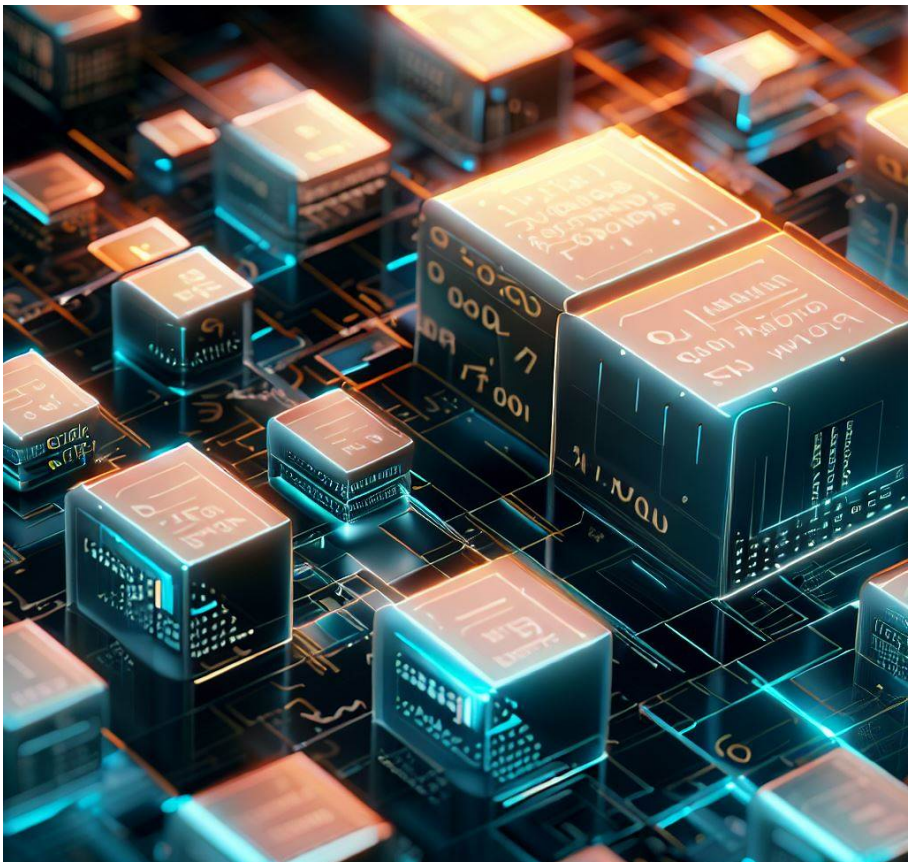
4. Make things easier to understand: By giving variables meaningful names, we can make our code easier to read and understand. Instead of using numbers directly in our

*© 2023 Theo Gottwald * https://www.smart-package.com*

program, we can use variable names like "numberOfApples" or "playerName" to make it clear what the values represent.

5. Play and be creative: Variables allow us to be creative and make our programs more interactive. We can create games where the player's score changes, or stories where the characters' names can be different each time we play.

In simple words, variables help us remember, change, and solve problems with different kinds of information in our programs. They make programming more fun and allow us to create amazing things with our imagination.



Intention

In the SPR (Smart Package Robot Script), variables are symbols that store numerical, text, or string data. They are used in scripts to store and manipulate information. In the SPR, variables are identified by their names, which always start with two dollar signs ($$). Here are some examples of valid variable names in the SPR:

```
$$A00
$$HAL
```

Please do NOT use Variables like:
`$$000` (3*Numeric) as these may have another usage and reference the Execution Stack.

These names can be chosen freely by the script writer, as long as they start with two dollar signs.
Variables allow scripts to be more dynamic and flexible, as they can hold different

© 2023 Theo Gottwald * https://www.smart-package.com

values at different points in the script. By using variables, you can store and retrieve data, perform calculations, and make your scripts more powerful and adaptable to different situations.

Variables are Symbols which contain numerical, text or string-data. When used in a script, **the variable name is replaced with the variable content.**
With the SPR, any Variable starts with two **$$** Signs. The following are valid names for variables.

```
$$AAA
$$FRA
```

You can even use Special-Characters and the Variable-Name will still be valid, like:

```
$$---  or $$###
```

Yet, we do not recommend this, unless for special Purposes.

Variables are Case-Sensitive!
Therefore this is not going to work:

```
VAR.$$FRA=10
MBX.$$fra
END.
```

This is going to work:

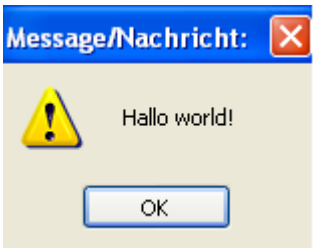
```
VAR.$$FRA=10
    MBX.$$FRA
    END.
```

Try the example below.

```
VAR.$$FIL=C:\MyDir\Mydir2
MBX.The dir was:$crlf$ $$FIL
ENR.
```

```
' You can take that further ...
VAR.$$TXT=Hallo world
VAR.$$TXA=$$TXT!
MBX.$$TXA
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



When working with variables in the SPR, it is important to note that:

1. Variables do not need to be explicitly declared, and memory allocation is handled automatically.
2. Variables can be defined, changed and cleared at runtime.
3. Variables can only be accessed at runtime, the preprocessor does not have access to the variable's content.
4. Each variable has an internal variable type, which can be determined and manipulated with specific commands.

The SPR's handling of variables allows for a flexible and efficient programming experience, while also providing the necessary tools to work with variable types and memory allocation.

Important!

How do you get the §-Character on non-german Keyboards?

Press "**ctrl-B**".

The SPRE-Editor will give you the § character when you press left "Ctrl" and "**B**" on your keyboard.

Pressing "**ctrl-H**" will give you the | (Pipe) Character.

Here are some of the different Methods to assign values to variables.

- The : - Definiton-Command 361
- VAR. - Variable Set Value/Clear 3562
- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP. - Variable Point 3560
- VAS. - Variable with String 3564
- VBA. - Variable Binary Append 3569
- OPT. - Optional Settings 2406

© 2023 Theo Gottwald * https://www.smart-package.com

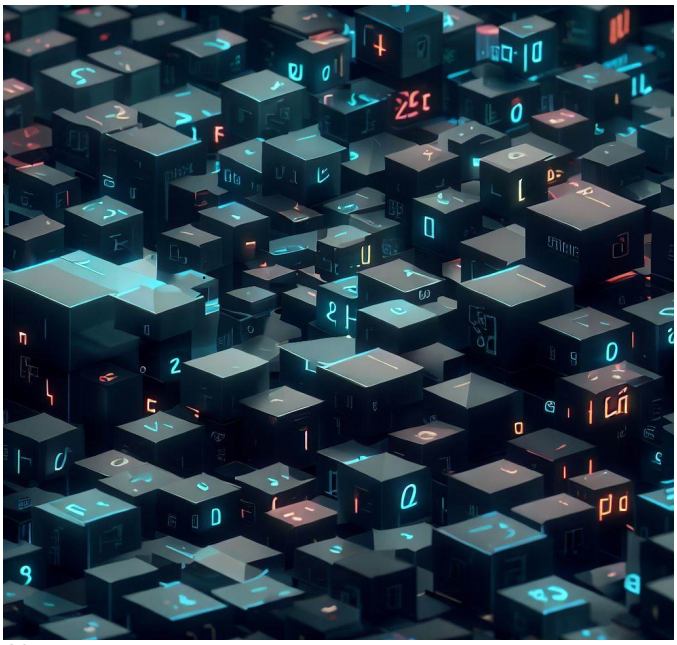
1.15 1.5.3 More about Variables

1.5.1.2.1 Using Variables **Previous** 60 **Top** 41 **Next** 104

Starting Guide

Variable Resolution

At some point the Robot will resolve the variables to their content.
During this topic, the content of the variables can contain more variables that may be also resolved or not depending on which commands you use.



Often variables are visualized as boxes that can contain something.

What is "variable resolution"?

It is the process, when a *variable-name* in the script is replaced with the *variable-content*.
In fact this is why variables are used.

Variable-resolution is: *the robot takes the **variable-name** and replaces it with the **variable content**.*

Generally the process is simple, but it can be more difficult in few cases. Let us see an example:

```
VAR.$$TXT=Computer
VAR.$$LIN=$$TXT
' Write the content of $$LIN in the Editor-Debug window
DBV.$$LIN
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

In this example, we first tell the robot that the variable $$TXT means "Computer".
In the second line, we tell it that $$LIN is the same as $$TXT, which means that $$LIN is also "Computer".
Looks easy. Is easy.

Resolving multiple variables
Lets make it a bit more tricky for the robot. What will the SPR do, if there are many variables in a variable?

```
' First we define three variables
VAR.$$TXA=Computer
VAR.$$TXB=Software
VAR.$$TXC=Center

' Now we assign them to $$LIN
VAR.$$LIN=$$TXA$$TXB$$TXC

' Write the content of $$LIN in the Editor-Debug window
DBV.$$LIN
```

The output is as expected:

```
ComputerSoftwareCenter
```

VAR. resolved all the variables.
Now lets go one step ahead. What will happen, if there will be variables inside these variables? Will they also be resolved?l
To do this we use a special Command, that is VAB. This Command will just take the right side and not go through a variable-resolution.

```
' First we define three variables
' Note that i have used VAB. instead of VAR.
VAB.$$TXA=Computer
VAB.$$TXB=$$TXA
DBV.Variable TXB contains >$$TXB<
' Here we use VAR. again
VAR.$$LIN=$$TXB
' Write the content of $$LIN in the Editor-Debug window
DBV.Variable LIN contains: <$$LIN>
ENR.
```

What do we expect to get here?
$$TXB contains $$TXA
and that is the result that we get.

```
Variable TXB contains >$$TXA<
Variable LIN contains: <$$TXA>
```

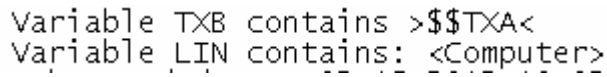
VAR. will resolve all variables on the first level, to be as binary compatible as possible, it will not make a "deep resolution".
Because this way, binary content inside a variable is safe. If you want a "deep resolution" choose VAV. instead.

© 2023 Theo Gottwald * https://www.smart-package.com

Let us give it a try:

```
' First we define three variables
' Note that i have used VAB. instead of VAR.
VAB.$$TXA=Computer
VAB.$$TXB=$$TXA
DBV.Variable TXB contains >$$TXB<
' Here we use VAV. to get a deep resolution
VAV.$$LIN=$$TXB
' Write the content of $$LIN in the Editor-Debug window
DBV.Variable LIN contains: <$$LIN>
ENR.
```

This is the result:



Now when we deal with text this is mostly an less important issue. Variable resolution becomes important when we deal with files that contain elements that could possibly be resolved.

For example, you want to have the phrase "`$$TXT is not 99`" in a Variable `$$LIN`.
If you try this:

```
VAR.$$TXT=Computer
VAR.$$LIN=$$TXT is not 99
```

then `$$TXT` will immediately be resolved to "Computer" and you do not get what you want.
Therefore we have a couple of different `VAR.`-commands with different behavior about variable-types and variable-resolution.
If you use `VAB.` instead of `VAR.` it will work.

```
VAR.$$TXT=Computer
VAB.$$LIN=$$TXT is not 99
' Output Variable to Editor
DBV.$$LIN
```

Important!

How do you get the §-Character on non-german Keyboards?

Press "**ctrl-B**".

The SPRE-Editor will give you the **§** character when you press left "Ctrl" and "**B**" on your keyboard.

Pressing "**ctrl-H**" will give you the | (Pipe) Character.

Here are some of the different Methods to assign values to variables.

*© 2023 Theo Gottwald * https://www.smart-package.com*

- The : - Definiton-Command 361
- VAR. - Variable Set Value/Clear 3562
- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP. - Variable Point 3560
- VAS. - Variable with String 3564
- VBA. - Variable Binary Append 3569
- OPT. - Optional Settings 2406

© 2023 Theo Gottwald * https://www.smart-package.com

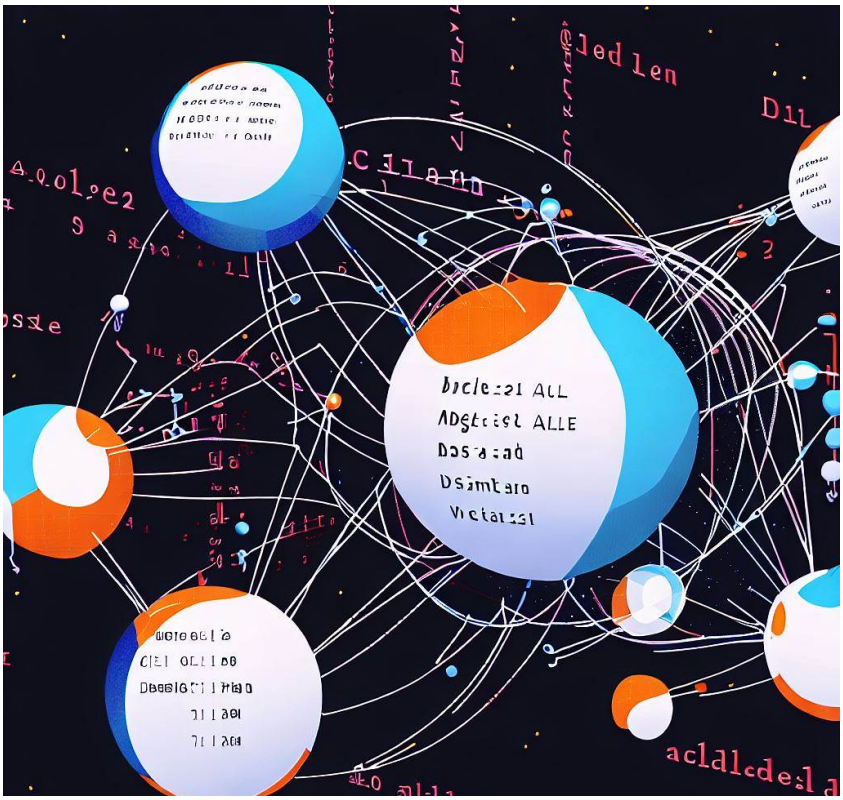
1.16 1.5.5 Global and Local Variables

1.5.1.2. Using Variables Previous 60 Top 41 Next 104

Starting Guide

Using Variables

There are two sorts of variables: global and local



Please prefer using the `SAV.` and `RES.` command before using "local variables".
In fact they are not more needed and more or less obsolete.
Yet they will be supported for compatibility.

VAR.$$TXT = VALUE

```
command|Target-side|    |value-side
```

Update: Due to the new `SAV.Save` and `SAV.Restore` Command the use of `$$XXX`-Variables (Script-local Variables) is not more recommended.
Just use the normal `$$XXX` (Global Variables) and make them "local" using the `SAV.`-Command.

Global Variables are valid everywhere in a Script.

© 2023 Theo Gottwald * https://www.smart-package.com

Local Variables are used inside "Sub-Scripts" which are called with the JNF.-Command.
Local variables contain their value until the script is left to the calling script. Then their content is lost.

If a script calls another script using JNF., the content of its local variables is preserved and automatically restored on return from the subscript.

$$xxx for global variables

Global variables keep their value anywhere in a script or subscript.

§§xxx for script-local variables

Local variables are cleared when the Script returns to the Main-Script, using RTS.
Local Variables and JNF. can be nested to unlimited depth.The Format for Variables is always:

Variable-names:
For xxx all letters or numbers as well as the "-" can be used.

Variable-names have a fixed length. Therefore we do not need any concatenation characters like in other scripting systems.
The variable-content can be any size, depending on the free memory in the windows system (memory fragmentation may play a role).
As a rule of thumb: several 100 MB should never be a problem.

There are few Variables which are reserved for Special Purposes which shall not be used for regular operations. Many Commands do return results directly into variables.

You can DUMP all currently used variables using the DMP.-Command.

```
' Dump Global Variables
DMP.2
MBX.Ready?
ENR.

' Dump Local Variables
DMP.3
MBX.Ready?
ENR.
```

For more Details see the DMP. Command.

When using the VAR.- type Commands,variable, Special-Folders 345 and Systemvariables 334 will be resolved!. The VAO. command for example does not resolve special folders or system-variables. If you choose VAB. the nothing will be resolved on the value-side.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

$$xxx global Variable
§§xxx **Script** local Variables
(local to Scripts until "ENR.")

Parameter Explanation

If the Parameter is omitted, the Variable is set to empty, which represents string with a of length 0, or a value of 0.

Example

```
' define global variable without Special-folders
: $$TXT=Hallo World

' define global variable with Special-folders and System-
variable Resolution
VAR.$$FIL=?progs\FS Office\Myprog.exe
PRT.$$FIL
' Display a list with all global Variables
DMP.4

' define local variable with Special-folders and System-
variable Resolution
VAR.§§FIL=?progs\FS Office\Myprog.exe
PRT.§§FIL
' Following clears variable §§NUM
: §§NUM=
' Display a list with all local Variables
DMP.5
```

Remarks

Global Variables are valid everywhere in a Script. Local Variables are used inside "Sub-Scripts" which are called with the `JNF`.-Command.

Such variables are cleared when the Script returns to the Main-Script, using `RTS`.
Local Variables and `JNF`. can be nested to unlimited depth.

Limitations:

How long can variables get? What is the maximum size? 100 MB ? 250, 500 MB ?
It depends on the amount of unfragmented Memory that the robot can get from the OS.
Under normal circumstances variables can always exceed a size of 50 or 100 MB (MegaBytes) without problems. Once you clear the variable with
`VAR.$$XXX=`

© 2023 Theo Gottwald * https://www.smart-package.com

the memory is safely returned to the OS. Of course this happens automatically when the script ends.

Please prefer using the `SAV.` and `RES.` command before using "local variables".
In fact they are not more needed and more or less obsolete.
Yet they will be supported for compatibility.

See also:

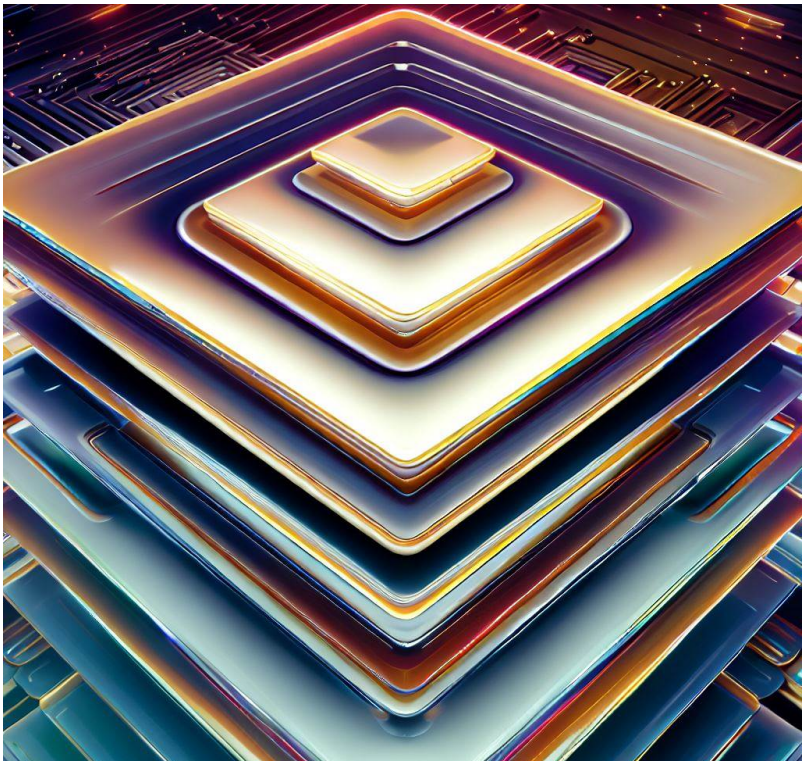
- The : - Definiton-Command 361
- VAR. - Variable Set Value/Clear 3562
- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP. - Variable Point 3560
- VAS. - Variable with String 3564
- VBA. - Variable Binary Append 3569
- OPT. - Optional Settings 2406

© 2023 Theo Gottwald * https://www.smart-package.com

1.17 1.5.6 Using the data stacks

Starting Guide

The Stack and Top of Stack (TOS)



Intention

What can you do with a stack?

The stack is a space where you can store thousands of items. The stack can hold:

1. text- or
2. binary strings or
3. numeric items.

A stack is something that is similar to a stack of paper. The last thing you put on top of it, is the first thing you get back from it.
This is called a LIFO-Stack (LIFO -> Last in First Out). Now lets answer some basic questions about the stack.

© 2023 Theo Gottwald * https://www.smart-package.com

How many stacks do i have?
You have two stacks. The global, and the local stack. More to this later.
Both are nearly equal in practical use. We'll just use the default stack for the following samples.

How can i see what data actually on the stack is?
Just type this sequence:

```
' Show stack content
STS.DUMP
'  Keep concole window open until we press OK
MBX.Ready
```

you can also use:

```
DMP.6
MBX.!
ENR.
```

to see the content of the actual selected stack.

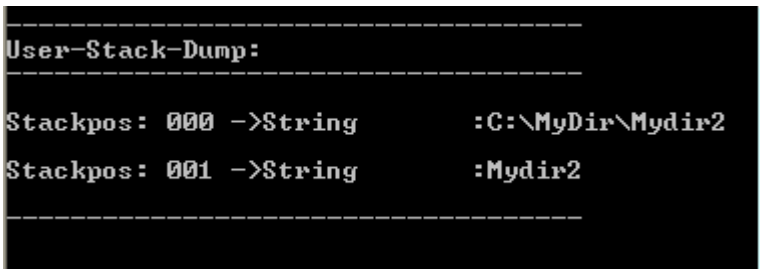
How can i put things on the stack?
You can use the commands:

- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019

But most often, you will just use just another command that will leave its result on the stack. In the following example, we will place one thing on the stack, and one will be placed on the stack by the ALB-command.

```
VAR.§§FIL=C:\MyDir\Mydir2
' In this case the result will be placed on the TOS
ALB.§§FIL
PUV.§§FIL
DMP.6
MBX.!
ENR.
```

And here is the output we get. It shows the two used Stack-Positions.



How do i get things back from the stack?

First, you can use these commands:

- POP. - POP Variable from Stack 3007

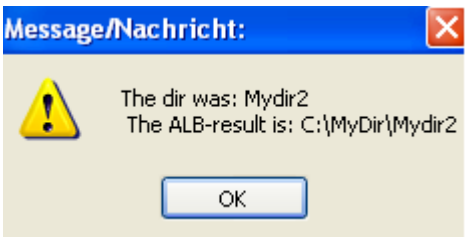
© 2023 Theo Gottwald * https://www.smart-package.com

- POV. - POp Variables from User-Stack 8010

but second there are a lot of other commands that may take parameters from the stack.

```
VAR.§§FIL=C:\MyDir\Mydir2
' In this case the result will be placed on the TOS
ALB.§§FIL
PUV.§§FIL
POV.$$TXA
POV.$$TXB
DMP.6
MBX.The dir was: $$TXA $crlf$ The ALB-result is: $$TXB
ENR.
```

Try this example. What you also see is, that once you access a stack element, its automatically removed. You will get a Messagebox like the one in the picture below.



Commands use it, too
Many commands leave their results on the stack. The stack organization can be customized to user needs.

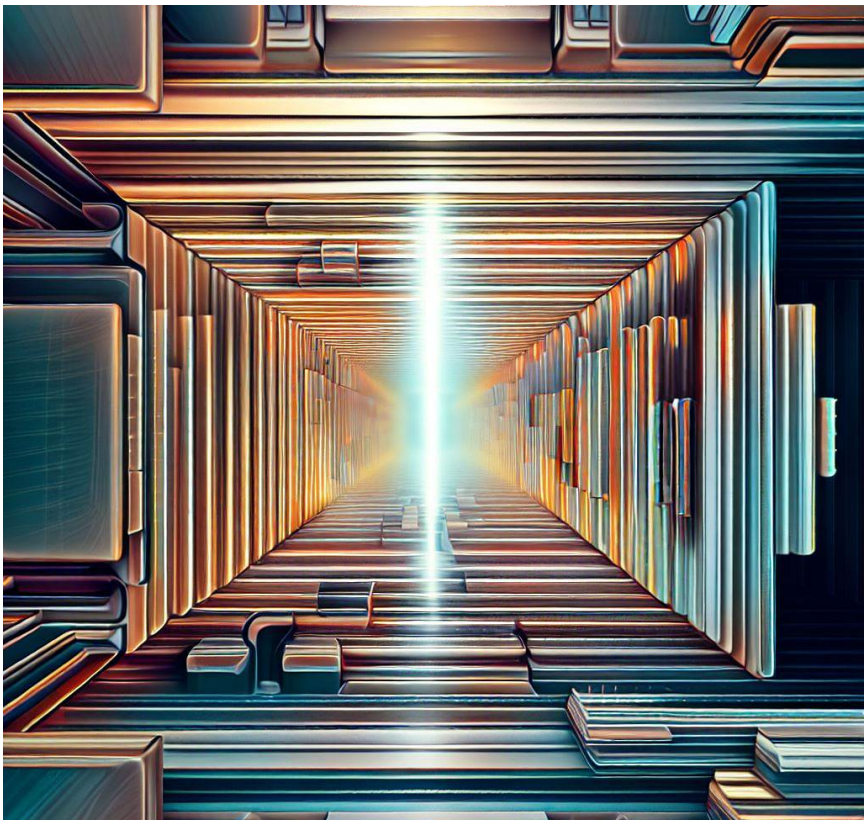
Customize your stacks
First you can change the number of elements the stack can hold. Use the STV. - command for this.

Any stack can be organized as a **stack** (LIFO - Last-In-First-Out) or as a **que** (First-In-First-Out). The default Stack setting is LIFO. Use the STS. - command to switch the stack between LIFO and FIFO. Note that if a stack gets to many elements, some overflowing elements are automatically deleted. These overflowing elements are generally the oldest elements. In a LIFO stack the oldest element is the first element added, while in FIFO (Que), the last added elements added ist the "oldest".

Another setting is, if t he stack element is remove if you touch it. Normally a stack element it deleted immediately when you touch it. This way you get all stack elements one-by-one, just by using `$tos$` which is always the TOS (Top of Stack) in the actual selected stack. If you want to get the TOS without having it removed, use `$dtos$`.

What if i jump into a subprogram that also uses the stack?
You can anytime take the whole stack-content and put it into a variable, using the STV.-command. When you return from your subprogram, you can "restore" the complete stack from that variable using the STV.-command.

© 2023 Theo Gottwald * https://www.smart-package.com



Use the script local stack
If you use Subscripts (scripts in separate files) you can take advantage of another feature.
Local and Global stacks. In fact, we have two stacks, a script-local stack and a global stack. The global stack is valid through the whole script execution. The local stack holds the same rules like the script local variables.

If a script calls another Subscript in another file, using the `JNF. command.` At this time the content of its local stack is preserved and automatically restored on return from the subscript. Thats the use of a local stack. To hold your script local data.

However, the Local stack is cleared when the Script returns upwards to a Main-Script, from which it was called, using `JNF./RTS`. Its the same sytem like when using script local variables.

How do I select the actual stack/que?
To select which stack you want to work with, use the `STS.GLOB` or `STS.LOCA` command.
These will make the selected stack to the actual stack.

If you want to know which stack is currently the actual stack, use the **`#stack#`** systemvariable. It will return a number that shows you if the Local or the Global stack is currently selected as the actual stack.
#stack# -> 1 GLOBAL Stack is selected
#stack# -> 2 LOCAL Stack is selected

*© 2023 Theo Gottwald * https://www.smart-package.com*

There are shortcuts to accessing the stack values inside your script.

§§000 to §§999
for directly accessing the local stack

$$000 to $$999
for directly accessing the global stack

§§000 - Top of Stack (TOS) local stack
$$000 - Top of Stack (TOS) global stack

The global stack keeps his content anywhere in a script or subscript.

There are different Methods to access the stack. First you can use the Stack-variables above to directly access the first 999 stack elements. Secondly you can use multiple stack-commands.

- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010
- STJ. - STack Job 3030
- STS. - Set STack Settings 3038
- STV. - Stack To/from Variable 3040

Most Commands may return results directly into stack-variables if you give these for the result.

Example

```
' Choose LOCAL Stack as Actual Stack
STS.LOCA
' Organize Stack as First-in-First-Out
STS.FIFO

' Put 10 Items on actual Stack
FOR.$$A01|1|10
  VAR.$$B00=Stackplace $$A01
  PUV.$$B00
NEX.

' Show all Items on Local Stack
DMP.5
MBX.Ready
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The $$000 and §§000 refer to the TOS (Top of stack). See several example scripts in the examples folder.

Limitations:
Per default the maximum stack size limit is set to 25 Elements. This can be changed using the STS.SIZE command.
How long can stack-entry get? Can you strore a file in it?
What is the maximum size? 100 MB ? 250, 500 MB ?
It depends on the amount of unfragmented Memory that the robot can get from the OS.
Under normal circumstances any stack entry can always exceed a size of 50 or 100 MB (Megabyte) without problems. Once you clear the stack entry, in any way, the memory is safely returned to the OS. Of course this happens automatically when the script ends.

See also:

- STJ. - STack Job 3030
- STS. - Set STack Settings 3038
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

1.18 1.5.7 Using Text-Strings

1.5.1.4. Using Text-Strings **Previous** 104 **Top** 41 **Next** 117

Starting Guide

Working with text strings

How do you analyze and synthesize text-strings,

What is a text string?

We have two kinds of strings in our daily IT-work:

1. Text-strings - these are mostly letters, linefeeds. Text-Strings are, waht you see in textfiles.
2. Binary strings - this strings can contain any of the ASC-Characters 0 to 255

When dealing with **Binary strings**, you need to be careful, as they may accidentally contain sequences 334 that could be interpreted by the robot-system. This can be variable names as well as special folders or system-variables.

Therefore you should only use commands that do a "one time resolution". These commands will just replace the variable-name of the binary string with the string and they will not touch the inside.

This way you can deal with both types of strings if you know that, and use only commands that are suitable for the specific case.

What can i do with binary strings?

Binary strings can be

- read from disk using CFF. - Copy-From-File 948
- find about its length using LEN. - Length-of-String 3140
- write them back to disk using CTF - Copy-To-File 964
- patch them using GSS. - GetSplitString 3137 and VAB. - Variable Binary 3544
- Replace part of it using RPL. - RePLace in String 3143
- They can be encrypted 825 and decrypted using GEC. - Get Encryption 825
- you can verify them using a checksum with GMD. - Get Message Digest 831
- and you can merge them in a package using SIB. - Set-In-Block 685
- or extract them from a package using GFB. - Get-From-Block 680

and there is more but these are the most important uses for binary string data.

What can you do with text string?

- read from disk using CFF. - Copy-From-File 948
- write them back to disk using CTF - Copy-To-File 964
- read text strings from INI-Files using CFI. - Copy-From_IniFile 951
- write them back to an INI-File with CTI. - Copy-To-Inifile 966
- find about its length using LEN. - Length-of-String 3140

*© 2023 Theo Gottwald * https://www.smart-package.com*

- Find out if it contains a substring using IVC. / NVC. - If-Variable-Contains 3524
- Check it against a Standard-Search Pattern 341 with IVP./ NVP. - If-Variable-Pattern 3529
- break it into delimited parts using SBD. - String between Delimiter 3070
- Sort delimited parts of a string using SRT. - Sort Items 3147
- split them using GSS. - GetSplitString 3137 and VAR. - Variable Set Value/Clear 3562
- Replace part of it using RPL. - RePLace in String 3143
- They can be encrypted 825 and decrypted using GEC. - Get Encryption 825
- you can verify them using a checksum with GMD. - Get Message Digest 831
- and you can merge them in a package using SIB. - Set-In-Block 685
- or extract them from a package using GFB. - Get-From-Block 680
- work with a file path using GFT. - Get-File-Token 1037
- or use regular expression with GRE. - Get Regular Expressio 3128 and GRR. - Get Regular Replace 3131

and much more!

Let us look at some examples on how to deal with strings.

How do we define a Textstring?
Its as easy as this:

```
: $$TXT=Hello world!
```

this sequence defines a Test-String.
In the same way, you can use the VAR. - Variable Set Value/Clear 3562 command to define a text string.

```
VAR.$$TXT=Hello world!
```

There is no difference between these two.While you can also just leave the : away and write:
`$$TXT=Hello world! $crlf$It's $date$ today.`
which will do all the same.

THere are other commands to define variables.
`VAB.` for example will not resolve 3.1 Systemvariables 334 or 3.3 Specialfolders 345 but just leave the variable content as it is. The second form will do that. Let's see the difference in a practical example:

```
' Will NOT resolve specialfolders,systemavariables etc.
VAR.$$TXT=Hello world! $crlf$It's $date$ today.
DBV.$$TXT
' Will resolve specialfolders,systemavariables etc.
VAB.$$TXT=Hello world! $crlf$It's $date$ today.
DBV.$$TXT
MBX.READY
```

And this is the output in the Editor Debug-window:

© 2023 Theo Gottwald * https://www.smart-package.com

```
[11:38:24] MRE: Robot start: VAR-Sample.mrt (On line #1)
[11:38:24] [9] Hello world!
It's 14-01-2012 today.
[11:38:24] [13] Hello world! $crlf$It's $date$ today.
[11:38:27] MRE: Robot ended: VAR-Sample.mrt
```

As expected, `VAR.` has resolved the system-variables and the `VAB.` did not.

Note that we use `DBV.` here, and not `DBP.` Because `DBP.` itself will resolve all specialfolder and system-variables. Then we would not been able to see the difference.

Why do we need this difference?
It's mostly to protect your code from unwanted code-changes. There is no real speed penalty when using `VAR..`

Note that the ":"-Definition Operator 361 still resolves variable names. It does not make a binary compatible resolution (suitable for binary strings) like VAB. - Variable Binary 3544.

```
' Will NOT resolve specialfolders,systemavariables but
' will resolve variable names!
: $$TXT=Hello world! $crlf$It's $date$ today.
: $$TXT=$$TXT $$TXT
DBV.$$TXT
```


© 2023 Theo Gottwald * https://www.smart-package.com

1.19 1.5.8 Basic String Operations

1.5.1.4. Using Text-Strings **Previous** 104 **Top** 41 **Next** 117

Starting Guide

BASIC String Operations

How do you work with string-data?

Adding string

That is a very basic issue. We have two strings and we want to add them to one string
We have already done that in the example above. Adding strings is as easy as writing their variable names one after the other.

```
: $$TXA=Hello
: $$TXB= world!
: $$TXT=$$TXA$$TXB
DBV.$$TXT
```

The output is as expected:

```
[11:52:58] MRE: File saved as: 'F:\00_RPGM\C
[11:52:58] MRE: Robot start: VAR-Sample.mrt
[11:52:58] [14] Hello  world!
```

Are there any size limit for strings?

Generally there is a 32-bit size limit that limits string sizes to 2 GB. In real life the limits depend on your system, on the fragmentation of your memory. The limit will be less because string memory must be sequential. 500 MB should be no problem, if your RAM allows it even more. Note that the larger strings become, the slower the string operations may get.

```
: $$TXT=This text is going to be very large.
FOR.$$MUC|1|22
  VAR.$$TXT=$crlf$ $$TXT $crlf$ $$TXT
NEX.
LEN.$$TXT|$$SIZ
DBP.$$SIZ
DMP.1
MBX.READY
ENR.
```

This script is going to concatenate a string of Size ~180 MB. Once we are adding up 40 MB and more, it will start to get a bit slower.

---

*© 2023 Theo Gottwald * https://www.smart-package.com*

Subtracting / Splitting strings

We have a string, but we do only need a part of it. There are many such cases in real life.
For example you get a phone number from a list or file. Or from a text-box.

```
: $$TXT=07247/9851112
GES.f|$$TXT|/|$$RES
DBP.Predial is: $$RES
GES.h|$$TXT|/|$$RES
DBP.Number is: $$RES
ENR.
```

The output will be:

```
[10] Predial is: 07247
[12] Number is: 9851112
```

What, if we do not know how the number is written?
It may be like this:

```
: $$TXT=(07247)9851112
' Will split thestring at ANY of the given characters
GES.f|$$TXT|)/|$$RES
' removes left and right of $$TXT all of the given characters
VBT.$$RES|()/
DBP.Predial is: $$RES
```

The output will be as expected:

```
[10] Predial is: 07247
```

Let us try something else. We have now the Pre-Dial and we just want the **first two characters** of it into a variable.

```
' We define our string
: $$TXT=07247
' We want character 1 to 2 from $$TXT into $$RES
GSS.1|2|$$TXT|$$RES
DBP.Predial is: $$RES
```

The output will be:

```
[9] Predial is: 07
```

Now we want **the last three characters** from our string.

```
' We define our string
: $$TXT=07247
' We want the last three characters from $$TXT into $$RES
GSS.-3|-1|$$TXT|$$RES
DBP.Predial is: $$RES
```

The output will be:

```
[11] Predial is: 247
```


© 2023 Theo Gottwald * https://www.smart-package.com

Get and set ASCII-values

Next we want to get an ASC-Values[3047] of one of the characters.

```
: $$TXT=07247
ASC.$$TXT|$$RES|1
DBP.The ASC-Value of Character 1 is: $$RES
```

The output will be:
[11] The ASC-Value of Character 1 is: 48

Next we want to put an ASC-[3047]character into a variable. This way you can add any international character to an variable, even if you may not easily find it on your keyboard.

```
: $$ASC=99
CHR.$$ASC|$$RES
DBP.The Character of ASC-Value $$ASC is: $$RES
```

Now lets take a look on string based decisions.

String-based decisions

We have a string for example from a command-line-parameter[334] and if it contains the sequence **/u** we want to do something different in our script.

We do it like this:

```
' To really get the command-line of a compiled Script-exe
' we would use this:
' VAR.$$CMD=$cmdexe$
' As we run it from the Editor, we use:
VAR.$$CMD=parameter is /q etc.
VAR.$$STR=/q
IVC.$$CMD=$$STR
  DBP. We do it different
ELS.
  DBP.We do it normal
EIF.
STS.DUMP
MBX.Ready
```

As you can see, `IVC.` will also place the position where it has found the string on to the TOS[369].

Besides checking if a string is inside another, we can do basic string compare using IVS. / NVS. - If-Variable-String[3531] .

```
VAR.$$CMD=/q
VAR.$$STR=s/q
IVS.$$CMD=$$STR
  DBP. We do it different
ELS.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  DBP.We do it normal
EIF.
```

IVS. and IVC. contain many more options then i have shown in these examples. Take some time and try these commands yourself. Experimenting is the best way to get experience.

Debugging/dumping all used variables

You can DUMP all currently used variables using the DMP.-Command.

```
' Dump Global Variables
DMP.2
MBX.Ready?
ENR.

' Dump Local Variables
DMP.3
MBX.Ready?
ENR.
```

For more Details see the DMP. Command.

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

1.20 1.5.9. Organizing data items in blocks

1.5.1.5. Organizing data items in blocks **Previous** 110 **Top** 41 **Next** 121

Starting Guide

Organizing data items in blocks

How to use Blocks (sort of data arrays)

Normally a variable can hold just one item. For example:

```
VAR.$$TXA=Rainer
```

That is definitely ok for most uses. There are however situations when you want to store multiple items in the same variable, to handle them as one item.

For example, this is a set of three variables:

```
VAR.$$PRN=Rainer
VAR.$$LAS=Gottwald
VAR.$$JOB=Diver
```

Now we want to put all three items into one variable. We want to handle them like they are one thing, one block of data. What we do is, we put them into a block (data-block).
Later we will see how we get them out there again.

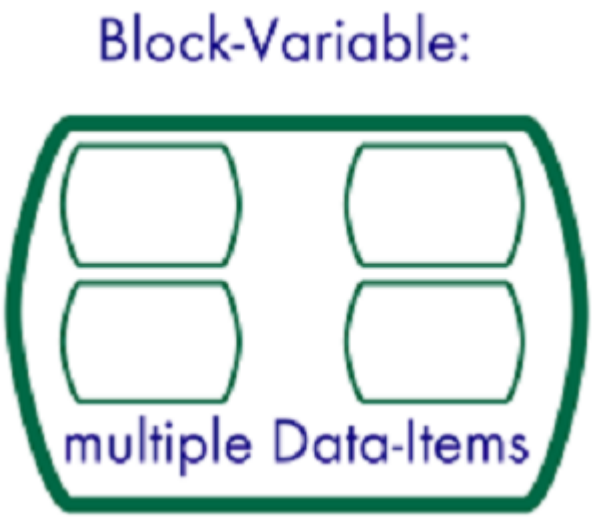
First we tell the robot that we want to have data block. And that we have three items. And that we want to have the data-block in the variable `$$BLOC`. You can choose any variable name for a block variable.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Tell the computer we want have a data-block for three items
CNB.$$BLO|3
```

Now we would go on and set the Block items, to values. We can use the `SBE.` or the `SIB.` command for that purpose. For example:

```
' Using SIB. we can set many block items in one strike
SIB.Element 01|Element 02|Element 03
```




```
' Using SBE. we are more flexible.
' The following will just set Element Nr.3
SBE.$$BLO|3|Element 03
```

Imagine you have Data that belongs together. For example, two names and a number.

Rainer Gottwald 3456
Anthon Robinson 3498
Oskar Schellbach 2467

Now how could we store them in variables and keep them together?
Of course you could store them in a delimited variable:

```
VAR.$$TXA=Rainer Gottwald 3456*Anthon Robinson 3498*Oskar
Schellbach 2467
```

This way we would have them delimited with the asterisk-sign * and we could get them easily out using the `SBD.` or the `SBM.` command. This is safe so long we are working with text.

If we have Binary data, that contains any of the 256 ASC-Characters this is not going to work. We can not use delimiters with Binary data. Thats where blocks enter the game.
Using blocks things are easy!

© 2023 Theo Gottwald * https://www.smart-package.com

Lets first create a Block for each Name and number:

```
CNB.$$BLO|3

Then we set the values into this block

SIB.$$PRE|$$NAM|$$NUM
```

Now the complete name and address data is in one block, `$$BLO.` We can add `$$BLO` to another block, just as we can add any variable to a block.

Blocks can be large, 10000 Elements per block is no problem possible.If blocks get larger, you may need too much time to change or retrieve single elements. After all it depends on your hardware, and on the size of the single elements.

```
' We create our two blockvariables
CNB.$$ADR|3
CNB.$$MAI|1
VAR.$$ITE=1
VAR.$$PRE=Rainer
VAR.$$NAM=Gott
VAR.$$NUM=3456
GSB.block
VAR.$$PRE=Anthon
VAR.$$NAM=Rob
VAR.$$NUM=3498
GSB.block

MBX.Ready
END.

:block
' Tell which block we want to fill
SBE.$$ADR
' Use the easy way to fill all elements in
SIB.$$ADR|$$PRE|$$NAM|$$NUM
' Now we fill the whole BLock into another one
SBE.$$MAI|$$ITE|$$ADR
VIC.$$ITE
RET.
```

What you can also see is, that we have "dimensioned" the Main Block `$$MAI` with only one element at start.

How do we extend a data-block?

Later we added a second element using `SBE.` Because we have added more items then the data block was dimensioned at this time, we have "extended" the data-block. Internally the data-block is going to be resized to the new size.

You can extend the element number of a block at any time by just adding the element using `SBE.` Extending a data-block will also work with `SIB.` Here is an example where `SIB.` extends a data-block to 3 items.:

```
CNB.$$BLO|1
SIB.Hallo|Thomas|Lange
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GBN.$$BLO|$$RES
'GBE.$$BLO|3|$$E01
DBP.Blo hat $$RES Elemente
GFB.$$E01|$$E02|$$E03
DBP.$$E01- $$E02 - $$E03
MBX.Blo hat $$RES Elemente
ENR.
```

Blocks are internally checked for authenticity. This is done to prevent hard to track errors.
If you change bytes or the length of a block variable externally, the block may not be unpacked anymore.

Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using `SBD.` `FEM.` and `SBM.` may be significantly faster.

There is no Limit, on Data-Block nesting. You can put a data-block into another data-block and so on. The only limit is the overall variable size. And this depends on the amount of Memory in the computer, as well as the actual memory fragmentation.
On a strong system, data-blocks can easily reach a size of several hundreds of Megabyte.

See also:

- 1.5.1.3. Using Variables 92
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685
- GFB. - Get-From-Block 680

© 2023 Theo Gottwald * https://www.smart-package.com

1.21 1.6.1. Program Flow Control

Starting Guide

Program Flow Control

How to script decisions

Program decisions (IF ... ELSE ... ENDIF (Ixx) and Negative decisions (Nxx)

The script language supports a lot of "two way decision" commands. These are called "Ixx" because they always start with an "I" followed by two more letters. Two way decision commands let the script choose between two alternative ways.

In natural language we would say for example:

IF "File exists" THEN
... do something
ELSE (if the file does not exist)
.... do something else
ENDIF ' end of the statement

Translated to our script language this same logic would look like:

```
' This is a real filename we are looking for
VAR.$$FIL=?desktop\MRE.lnk
'IEF. = If Exists File (or folder if it ends with a "\")
IEF.$$FIL
      ' This will happen if the file exists
      MBX.The shortcut is available.
ELS.
      ' This will happen if the file DOES NOT exist
      MBX.The shortcut is not available.
EIF.
```

Often enough we only need a subset of the two-way decision. Especially if we only need to do something in one of the two cases. For example, if we need to delete a file only if it exists,

```
VAR.$$FIL=?desktop\MRE.lnk
IEF.$$FIL
      ' This will happen only if the file exists
      DEL.$$FIL
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Now what, if we need to do something if the file does not exist only?
Of course we could write this:

```
VAR.$$FIL=?desktop\MRE.lnk
IEF.$$FIL
ELS.
      ' This will happen only if the file does not exist
      DEL.$$FIL
EIF.
```

But that is not the proper way to do it.
Instead we can just change the "I" in "IEF." into an "N", making it to "NEF.". Like this:

```
VAR.$$FIL=?desktop\MRE.lnk
NEF.$$FIL
      ' This will happen only if the file does not exist
      DEL.$$FIL
EIF.
```

This will work for all "Ixx" commands, not just for IEF. There are many such Ixx-commands like IEF., take a look into the help and try some of them. And also try the negative version "Nxx".

And what do i use, if i want to select between multiple choices?

You can use the Multiple-choice commands `SCS. ... ESC.` These commands are technically similar to SELECT CASE ... END SELECT statements in many other scripting languages. You can choose between string and numeric value comparison.

You always start with SCS.(variable)
In case of numeric comparison, you use CAN. (CAse Numeric). In case you prefer string comparison, using the standard-search patterns 341, CAS. (CAse String) is the instruction of your choice. See this example:

```
: $$TXT=Karl Kovse
SCS.$$TXT
' Use a part of the string or standard search patterns
CAS.Karl
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready Sample 1
ENR.
```

Numeric comparison is done using the CAN. - instruction.

© 2023 Theo Gottwald * https://www.smart-package.com

```
: $$VAL=1
: $$TXT=5
SCS.$$TXT
' As you can see, you can use formulas here
CAN.(2*2+$$VAL)
  PRT.This will come
CAN.3
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready
ENR.
```

Any formula that works with CAL. should also work with CAN. About CAS. you can use any search pattern, including variables. See this example:

```
: $$TXT=Karl Kovse
: $$STR=Karl&ANDTHEN:se
SCS.$$TXT
CAS.$$STR
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready Sample 1
ENR.
```

Can i define my own control flow instructions with Macros?

Yes, Macros enable you to define your own instructions 253. Even Control-Flow Instructions.
Lets make an example.

```
: %SELECT_CASE 1
SCS.§§§01
END%

: %CASE 1
CAN.§§§01
END%

: %CASE_ELSE 0
CAE.
END%

: %END_SELECT 0
ESC.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END%

: $$TXT=Karl Kovse
: $$STR=Karl&ANDTHEN:se

' Now look how the final script looks.
' And it works exactly like before.
%SELECT_CASE $$TXT
  %CASE $$STR
   PRT.This will come
  %CASE Peter
   PRT.This will not come
  %CASE_ELSE
   PRT. This will not come
%END_SELECT
MBX.Ready Sample 1
ENR.
```

Lets see a more complex Macro Example. We want a command that checks if a variable is in a specified range, and if not corrects their values.

```
: %Check_Bounds 2
IVV.§§§01<§§§02
  : §§§01=§§§02
EIF.

#IF PARAMS>2
IVV.§§§01>§§§03
  : §§§01=§§§03
EIF.
#EIF
END%

: §§VAL=5

' Here we check only the lower bound
%Check_Bounds §§VAL|6
PRT. First Result is §§VAL

' Here we check the lower and the upper bound
%Check_Bounds §§VAL|1|4
PRT. Second Result is §§VAL

%Check_Bounds §§VAL|5|12
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT. Third Result is §§VAL

MBX. Ready
ENR.

MBX.Ready Sample 1
ENR.
```

These Macros enable you to define your own instructions. You can define these Macros and put them into an separate file that you include to your script using `JNF`. This way you have them always at hand. Please note that MACROS are only expanded inside the robot.

*© 2023 Theo Gottwald * https://www.smart-package.com*

1.22 1.6.2. Looping around

Starting Guide

Looping around

Understanding Loops: The Power of Repetition

A conveyor belt is a good example of a repetitive activity

Looping around - what is "a Loop"?

Computers excel at performing repetitive tasks, tasks that might be monotonous if done manually. Let's delve into the concept of 'loops', a fundamental principle in programming that enables this powerful repetition. We'll begin with a straightforward example to ease your understanding.

Assume you want to count from 1 to 50. In a script you could do it like this:

```
'  This the most complicated way to do it.
' Init counter variable
: §§CNT=1
' This is just a Label (jump mark)
:Loop
' Print the actual number
PRT.§§CNT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'Increment the number (do a +1)
VIC.§§CNT
' Leave the Loop, if we are above 50
IVV.§§CNT>50
    GTO.exit
EIF.
' Jump back to the start of our Loop
GTO.Loop
' Another Label
:exit
```

This is really a long script for a loop!
Lets try to a shorter version. We can change the "IVV." to a JIV.-command.

```
' Init counter variable
: §§CNT=1
' This is just a Label (jump mark)
:Loop
' Print the actual number
PRT.§§CNT
'Increment the number (do a +1)
VIC.§§CNT
' Leave the Loop, if we are above 50
JIV.§§CNT>50|exit
' Jump back to the start of our Loop
GTO.Loop
' Another Label
:exit
```

This will work but is not really a lot shorter. If we choose a JME. instead we will save one more line, because JME. make an Auto-Increment if we use the ">" operator. I'll also add some statements that do a time-measurement.:

```
' Remember actual time
VAR.§§TIM=#time#
' Init counter variable
: §§CNT=1
' This is just a Label (jump mark)
:Loop
' Print the actual number
PRT.§§CNT
' Leave the Loop, if we are above 50 and Auto-Increment
JME.§§CNT>50|exit
' Jump back to the start of our Loop
GTO.Loop
' Another Label
:exit
' Get time since last call to #dtime#
VAR.§§TIM=#since#
' Tell me how long time the execution needed.
MBX.§§TIM
```

If you run the script in the editor, you can see that we needed for example about 4 seconds. If you would run it as a compiled executable, it will easily be 100 or more times faster.

---

*© 2023 Theo Gottwald * https://www.smart-package.com*

The FOR .. NEX.-Loop
At this point, we could get a bit simpler when using `JOR`. I'll leave that for yourself to try. Instead we will now take the proper way of a "counting loop". It is the "FOR. ... NEX."-Loop.

Imagine you want to count from 1 to 10. How would you do that?
You would start counting with 1 ... 2 ... until you reach 10.

That is what the `FOR. ... NEX.` command is doing. If we count like In our example, from 1 to 10, it would look like this:

```
' This will count from 1 to 10
FOR.$$CNT|1|10
' ... we can do something with the number in $$CNT
NEX.
```

It will start counting from a given value (in our example it is 1), until another given value (in the example 10) is reached. All the way, we have the actual value in the variable `$$CNT`.

`FOR. .. NEX.` has **several Modes of Operation** to match all your needs.

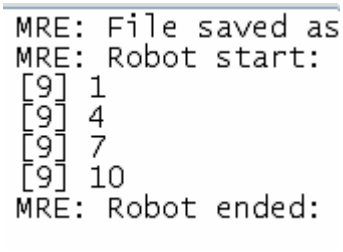
First, you can specify a STEP-value. It will look like this:

```
FOR.$$CNT|1|10|3
  DBP.$$CNT
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

You will see an Output in the Editor-Debug-Area like this:



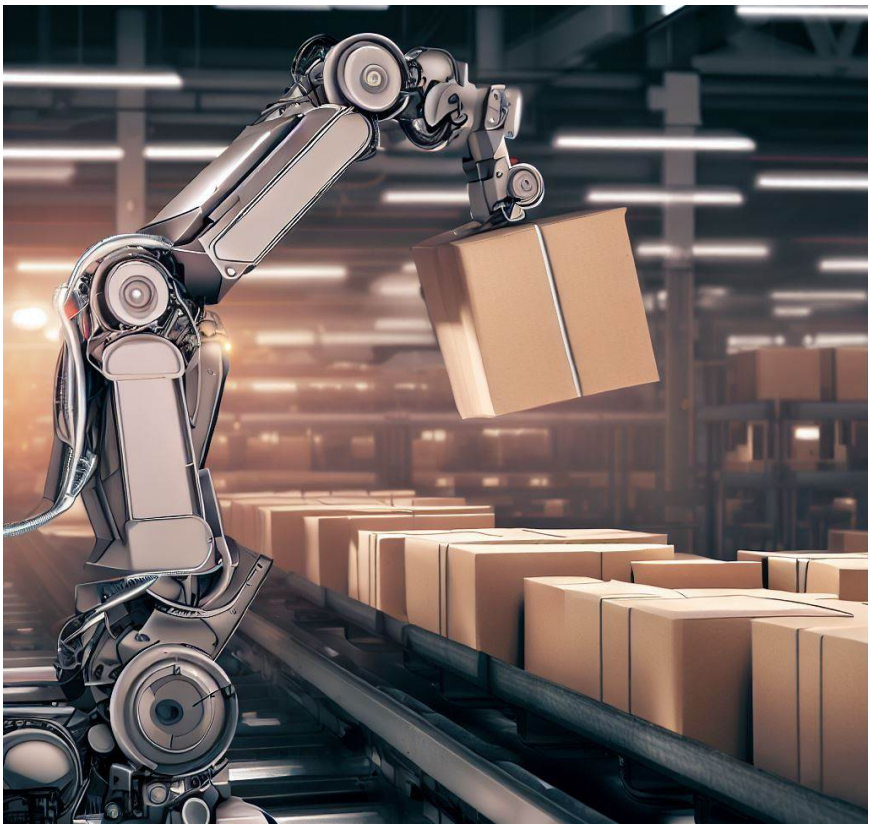
The `[9]` tells you that the `DBP.$$CNT` command was in line 9 of our script. More important are the other numbers. They start with 1 and then the Counter always adds **3.**
That's exactly the **step-value** we have given!

Here is another example, like the one we made before.

```
' Start our stop-watch
VAR.§§TIM=#dtime#
' Init counter variable
FOR.§§CNT|1|50
' Print the actual number
  PRT.§§CNT
NEX.
' Get the time how long we needed.
VAR.§§TIM=#dsince#
MBX.§§TIM

ENR.
```

How many seconds does it need on your computer? Using the proper counting loop, we saved a lot of lines and made all the same in for example 2 seconds!

© 2023 Theo Gottwald * https://www.smart-package.com



Do we need loops to wait for something?
Lets think of a situation, where we are waiting for files to appear or disappear.

Normally we do not need to use a loops to wait for one event to happen. The reason is that such loops are already built in. For example, to wait for a file, we can simply use the WFF. - "Wait For File" Instruction. We would write:

```
WFF.c:\Logfile.txt
```

to wait for the specified file to appear. In the opposite case, that we need to wait for a file to disappear, we would add a "<>" like that:

```
WFF.<>c:\Logfile.txt
```

This will wait for the file to "Not be there". You can also specify a timeout and test if the timeout time was reached, using the ITO. "If Time Out" - Instruction.

The same applies for many other tasks, where we can use the "Wait-For" Instruction Group.

But what do we do, if we want to wait for "one of several files" to appear?

I am using files in this example, but the same applies to any other system-object. Like windows, processes, or whatever.

To do this we must construct our own loop. This time we do not need to count anything, We need just to "Loop in a circle" until things happen.Like an endless loop.

© 2023 Theo Gottwald * https://www.smart-package.com

Until something happens. When this thing we are waiting for happens, then we leave the loop and do something.

© 2023 Theo Gottwald * https://www.smart-package.com

1.23 1.6.3 Flexible Loops

Starting Guide

Flexible Loops

Understanding Loops: The Power of Repetition

A conveyor belt is a good example of a repetitive activity

The conditional DOL. - OOP. Loop

For such situations you have the DOL. OOP. construct. Lets make a endless loop.

```
DOL.1
  PRT. I am in here forever!
OOP.
ENR.
```

Try this in the Editor, it will run in an endless loop, until you stop it. This was just for you to see how it works. Now we'll use it for a real world example.

© 2023 Theo Gottwald * https://www.smart-package.com

In reality you will have conditions when you jump out of this loop. Lets take a situation where we wait for two files. If one of them appears, we will leave the waiting loop.

```
VAR.§§FIA=?pfad\myfile1.txt
VAR.§§FIB=?pfad\myfile2.txt
' Here is our endless loop
DOL.1
  PRT. I am here!
' We test for File1
  IEF.§§FIA
    GTO.exit
  EIF.
' We test for File2
  IEF.§§FIB
    GTO.exit
  EIF.
  ' We'll give the CPU a tenth of a second sleep time
  ' always do this in endless loops to save CPU time.
  ' for other running programs
  PAU.0.1
OOP.
:exit
MBX. A file appeared!
ENR.
```

You can run this and when the file appears, the script will exit its endless loop.
While this script is completely correct, we will change it a bit. So we do not need an label "exit", and no GTO. "GOTO". We use the power of DOL. ... OOP.
These two statements allow you to construct whatever Loop you want.

```
VAR.§§FIA=?pfad\myfile1.txt
VAR.§§FIB=?pfad\myfile2.txt
' This variable will be set to 1 if the file is found
: §§XIT=0
' Loop will run WHILE §§XIT=0
DOL.§§XIT=0
  PRT. I am here!
' We test for File1
  IEF.§§FIA
    : §§XIT=1
  EIF.
' We test for File2
  IEF.§§FIB
    : §§XIT=1
  EIF.
  ' We'll give the CPU a tenth of a second sleep time
  ' always do this in endless loops to save CPU time.
  ' for other running programs
  PAU.0.1
OOP.
:exit
MBX. A file appeared!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

We would construct the same loop better as a "REPEAT ... UNTIL" loop. Because that is what we are doing. We REPEAT ... UNTIL a file is found. Therefore we change it again.

```
VAR.§§FIA=?pfad\myfile1.txt
VAR.§§FIB=?pfad\myfile2.txt
' This variable will be set to 1 if the file is found
: §§XIT=0
' Here is our loop
DOL.
  PRT. I am here!
' We test for File1
  IEF.§§FIA
    : §§XIT=1
  EIF.
' We test for File2
  IEF.§§FIB
    : §§XIT=1
  EIF.
  ' We'll give the CPU a tenth of a second sleep time
  ' always do this in endless loops to save CPU time.
  ' for other running programs
  PAU.0.1
' This loop will repeat UNTIL §§XIT=1
OOP.(§§XIT=1)
:exit
MBX. A file appeared!
ENR.
```

CPU-time considerations

Now let us see an endless loop:

```
 :Loop
... do something
GTO.Loop
```

This is just an example for an endless loop. You have many more options to build loop.
See here: 1.6.1. Program Flow Control

Now if you run this as a compiled executable it will run very fast!
Here is an example:

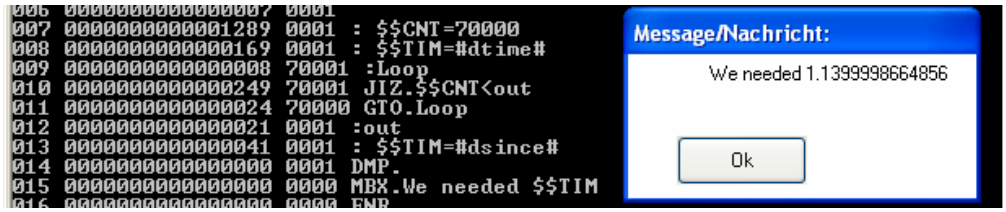
```
: $$CNT=70000
: $$TIM=#dtime#
:Loop
JIZ.$$CNT<out
GTO.Loop
:out
: $$TIM=#dsince#
DMP.
MBX.We needed $$TIM
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

This Loop will be processed about 70000 times in about a second, see picture below. This needs a good part of possible CPU-time. If you have a multicore CPU, it may take a complete core just to run in a circle.

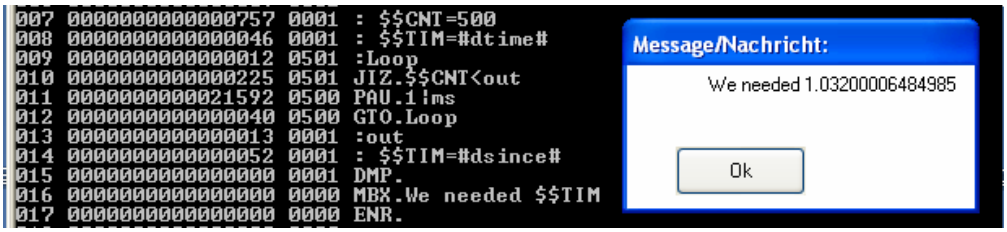
In most cases you wait for something to appear or something to disappear. You don't need 70000 checks per second, 100 would be more then enough.
Using the `PAU.` - Instruction, you can slow the process down and give CPU-time to other running processes.



We can change the script like this:

```
: $$CNT=500
: $$TIM=#dtime#
:Loop
JIZ.$$CNT<out
PAU.1|ms
GTO.Loop
:out
: $$TIM=#dsince#
DMP.
MBX.We needed $$TIM
ENR.
```

and get this result:



This time we just make 500 rounds (still a lot!) and in each round a rest of 1 ms.
1 ms = 1/1000 th part of a second.

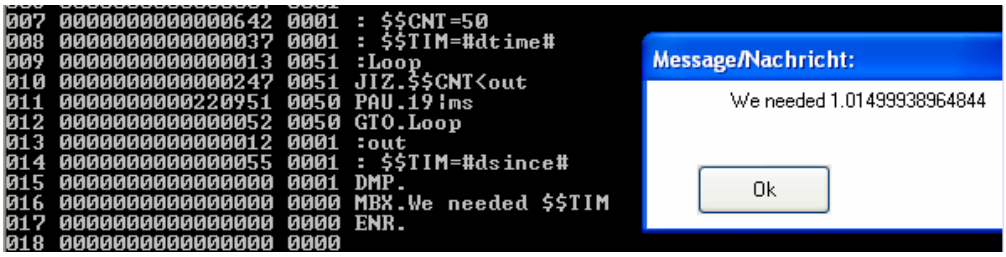
Lets get it down to about 50 rounds per second. This way we can finetune the CPU Usage of our script.

```
: $$CNT=50
: $$TIM=#dtime#
:Loop
JIZ.$$CNT<out
PAU.19|ms
GTO.Loop
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:out
: $$TIM=#dsince#
DMP.
MBX.We needed $$TIM
ENR.
```

this will get the expected result, 50 rounds per second with a waiting time of about **19 ms** per round. Of course these time differ with the used system, CPU etc. You may get other values on your system.



However, this example should get you started how to "fine tune" Loops, for example in case you run server-based scripts.

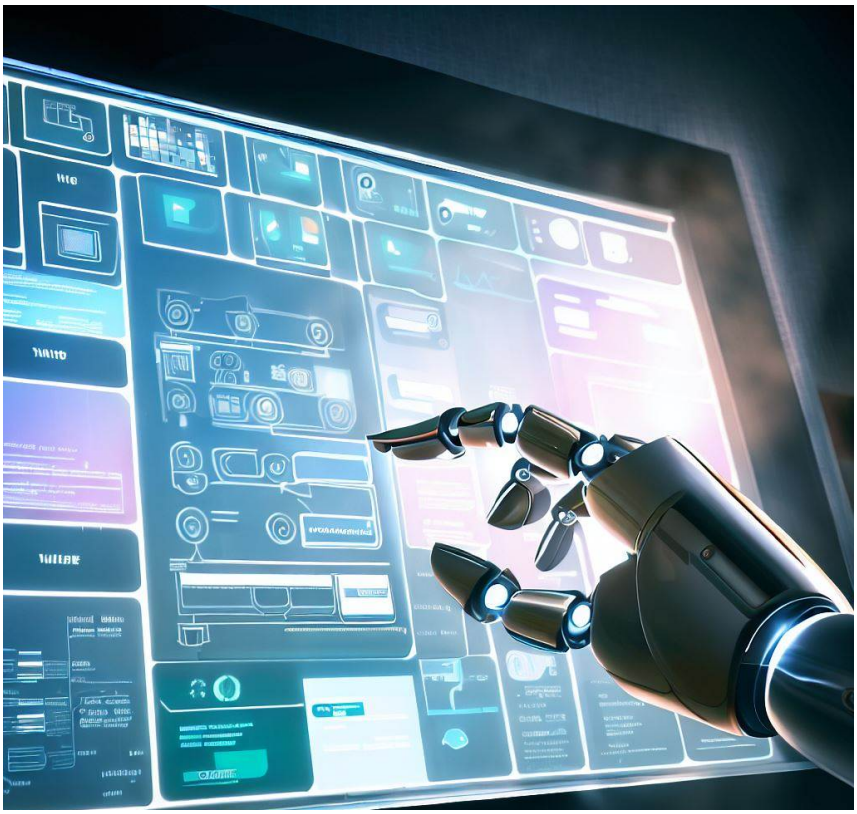
© 2023 Theo Gottwald * https://www.smart-package.com

1.24 1.7. Application and PID

Starting Guide

Application and PID

Understanding Applications and Processes



What is an application?

An application is a program that you interact with on your computer. When you start an application, typically a window appears which allows you to input commands and control the application via clicking and typing.

What is an application window?

An application window is an interface that allows you to interact with the application. It lets you control the application and view its output.

What happens when you close an application window?

Usually, when you close an application window, the corresponding application process also ends and is removed from the task manager. However, there are instances where the window is closed, yet the process remains in the task manager.

What does 'killing a process' mean?

© 2023 Theo Gottwald * https://www.smart-package.com

Killing a process refers to manually terminating it, typically through a tool like the task manager. When you kill a process, its associated application window will close and disappear from the task manager.

Can an application have multiple windows?

Yes, an application can operate with one, several, or even no windows.

Can a single window be linked to multiple processes?

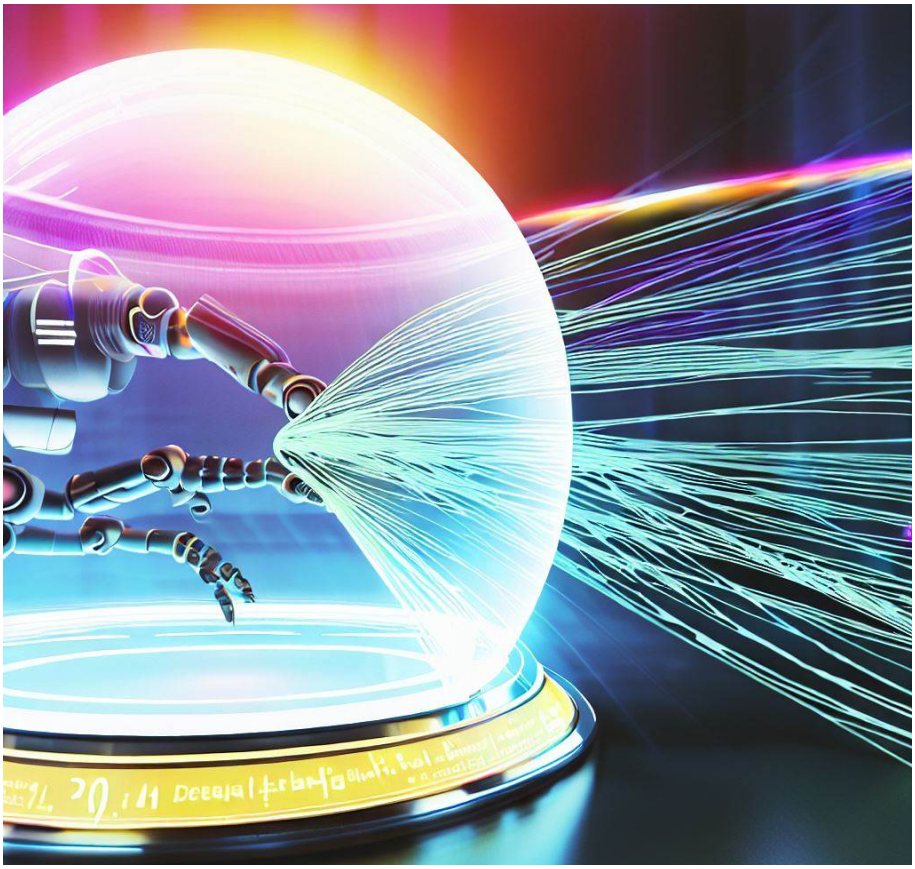
No, each window is directly associated with a single application process.

What is the difference between an application and a process?

A process refers to any task that's visible in the task manager, whether it's tied to a window or not. An application, on the other hand, requires a window because you need an interface to interact with and control it.

What is a PID or Process-ID?

PID stands for Process Identifier. In the context of computing, a PID is a unique numerical identifier assigned by the operating system to each running process. This allows the system and users to track and manage processes. It's similar to how each student in a school is assigned a unique roll number.



**A multitasking OS enables you to run hundreds or Processes at the same time.
Each of these processes can contain several parallel running Threads.
All of these are defined by a unique Process ID and Thread ID.**

What is a Thread?

© 2023 Theo Gottwald * https://www.smart-package.com

A Thread is "Multitasking inside of a Single process". If an Application has several Threads running, this means it has several small processes running at the same time.
A thread is the smallest unit of execution within a process. While a process contains the program code and its own memory space, a thread contains a program counter, a register set, and a stack space. Multiple threads within a process share the same data space with the main thread and can therefore communicate with each other more easily than if they were separate processes. Threads are sometimes called lightweight processes.

What is a Thread-ID (TID)?

Similar to a PID, a Thread-ID (TID) is a unique identifier assigned by the operating system to each thread within a process.
In computing, a thread is the smallest sequence of programmed instructions that can be managed independently by an operating system's scheduler. Essentially, a thread is a unit of execution within a process.

Each thread within a process has a unique identifier known as a Thread-ID or TID. This identifier allows the operating system to manage and keep track of individual threads within a process. It's important to note that a single process can have multiple threads, each with its own unique TID.

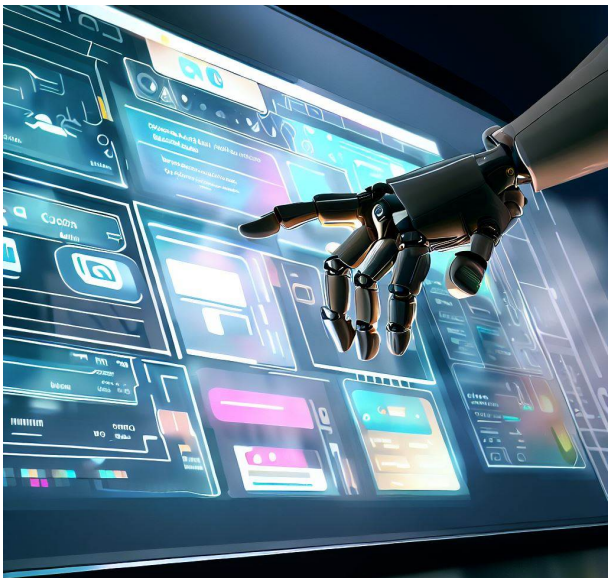
In terms of how threads are used, they allow for concurrent operations within a process, making it possible for multiple tasks to be performed at the same time within the same program. For example, in a web browser, one thread may display the user interface, another might handle network communication, and yet another might run JavaScript code on a webpage.

This allows for the tracking and management of individual threads. It's important to note that a single process can have multiple threads, each with its own unique TID.

How can robot-scripts interact with applications?

Robot scripts can be written to perform a variety of tasks within an application, such as clicking buttons, entering data, and more.

© 2023 Theo Gottwald * https://www.smart-package.com



What can we do with applications using the robot-scripts?
We can start an application in multiple ways. For example using:
EXA. - Execute Asynchron 872
EXB. - Execute-Blocking 876

We can kill an application using the commands:
KPN. - Kill Process by Name 2904

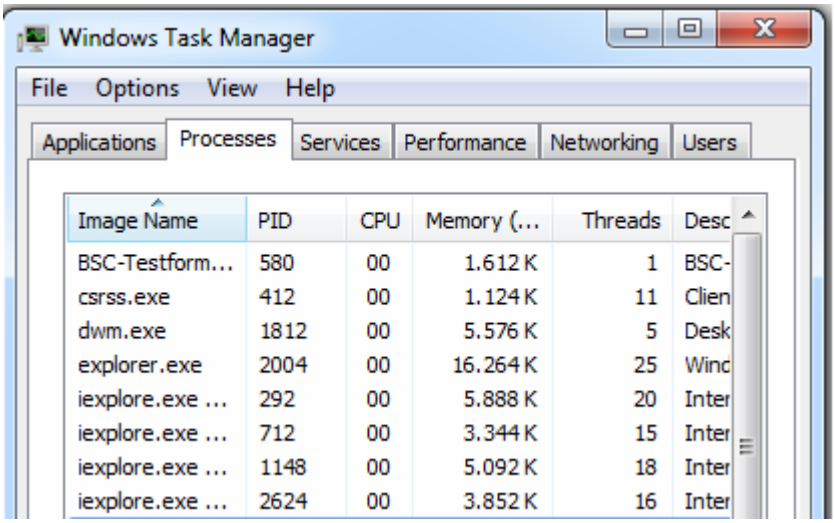
We can check if an specified application is actually running.
IEP. / NEP. - If-Exists-Process 2900

And finally we can enumerate/count all running processes using
FEP. - For-Each-Process 932

Which is less often used.

Lets take a look into the task-manager, which applications/processes are currently running:

© 2023 Theo Gottwald * https://www.smart-package.com



What we can see is that there are several processes with the same name "iexplore.exe".
But how do we find the one we need?

How can we select a specific process/application?

Rather then selecting a process "by name", we can use the process-ID (PID).
The PID is a number that is unique to each process. Having the PID, we can work with the process.

Using the PID we can do things like:
CPR. - Close-PRocess 2882

Find out which user has started the process:
GPU. - Get Process User 2898

we can wait until a process ends. Using
WPT. - Wait-Process-Termination 2913

If we have just started a process - for example with -
EXE. - Execute Programm 883

then we can use
WPR. - Wait-Process-Running 2911

to be sure the process we need is running. A PID is a magic number when dealing with applications and processes. If you have the PID, you can do anything you want with that process. As long as you have the sufficient rights.

How do we get the PID from an process or application?

1. We may get the PID directly from the starting command. Only starting commands that wait until the process is running, can support filling #pid#. Therefore see the description of the starting command if it supports #pid#.

If we use:

© 2023 Theo Gottwald * https://www.smart-package.com

```
EXE.?pfad\UltraVNC_1.0.9.6.1_Setup.exe
VAR.$$PID=#pid#
MBX.MY PID is: $$PID
```

we get the PID in the system variable `#pid#` automatically.

2. We get the PID from any window. It doesn't matter if its a Top-Level window or a control (child-window). If we have the window, we can get the PID like this:

```
STW.t|Testapp
' We'll put the PID of the process from that window
' into the variable $$PID using the GDF.-command
GDF.ap|$$PID
```

It does not need to be a Top-Window, it can be any actual localized window, try this

```
STW.t|Testapp
SCW.c|butto
GDF.ap|$$PID
' Will print the PID in the Debug-Area of the Editor
DBP.$$PID
ENR.
```

3. We can get the PID from the name of the EXE-File

```
VAR.$$FIL=?exepath\MR01.exe
GDF.fp|$$FIL|$$PID
' Will print the PID in the Debug-Area of the Editor
DBP.$$PID
ENR.
```

or we can use `IEF.` for that.

```
' Get the PID using IEP.
IEP.Testapp_01.exe
     MBX.#pid#
ELS.
     MBX. That Application is not running!
EIF.
ENR.
```

`GDF.` works with any process in the Task-Manager, no matter if that process has a window or not.

What makes IEP. (If exist process) so powerful?

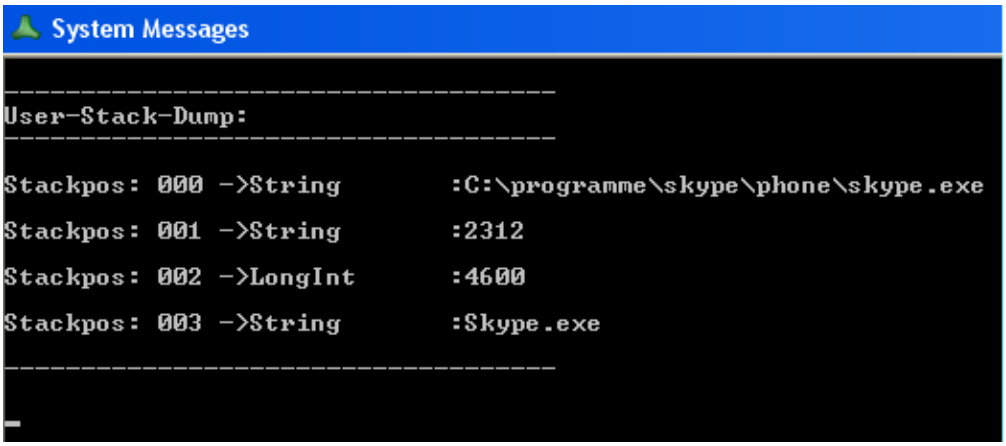
IEP. primarly helps you to find out, if a process is running. For this you can specify the process in several ways, including the use of 3.2 Standard-Search Pattern 341.

Assume you use IEP. to check if there is a process like this:

 © 2023 Theo Gottwald * https://www.smart-package.com

```
IEP.sky
  DMP.6
EIF.
MBX.Ready

ENR.
```

Now IEP. finds such a process. But what is the real name of the process?
sky ... what?
We have just specified "sky". As IEP. uses Standards Search Patterns, any part of thereal name is valid.
The name of the process must only contain that phrase. We get this information if we take a look on the actual stack.



`IEP.` provides us with a lot of information about the found process. In position zero

```
IEP.x64.exe
   MBX. Process ID is: #pid#
EIF.
```

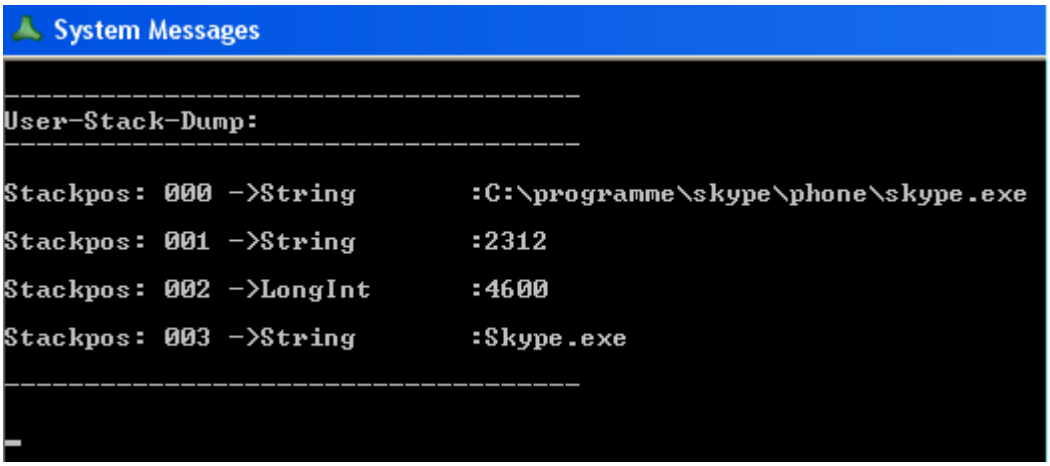
If you have multiple EXE running with the same name, you can specify a number. If the EXE is found, the #pid# is set to the EXE's Process-ID. Please note that #tid# is always set to zero.

You can specify a full path, or only the filename or a part of it. If you search for a dll, then the extension ".dll" must be included in the pattern.

You can use `NEP.`, the negative Form of this command (`NEP.` - Not Exists Process).

If a process is found, valuable information is placed on the actual stack. These include the full name and path of the found executable.

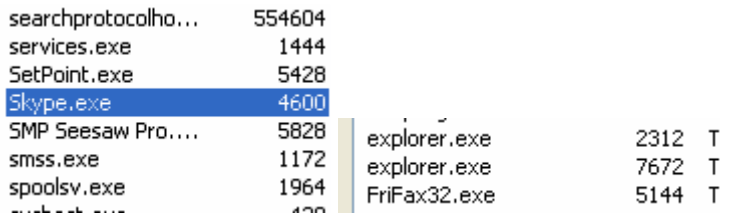
© 2023 Theo Gottwald * https://www.smart-package.com



Below are two snapshots of the windows-taskmanager. Skype.exe has the PID (process-id) 4600. Its on the stack on position 02 in the picture above.
The 2312 which is on position 01 (picture above) is the parent-process, that is the process which has started skype.exe.
In the task manager, we can see that this is the PID of the windows-explorer. This way you can find out which process has started another process.

If you give IEP. a variable as third parameter, then the stack will not get the full path of the found file. You will find it in the given variable instead.

Please note that the 2.4 The global and local Stack 369 as Que or as Stack will influence the sequence in which the items appear on the stack.



You can also use IEP. to verify, if a process with a specified **PID is alive** or not.
To do this, the Syntax is:

```
STS.GLOBAL
IEP.>1234
   MBX.PID 1234 (Filename=$$000) is alive!
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

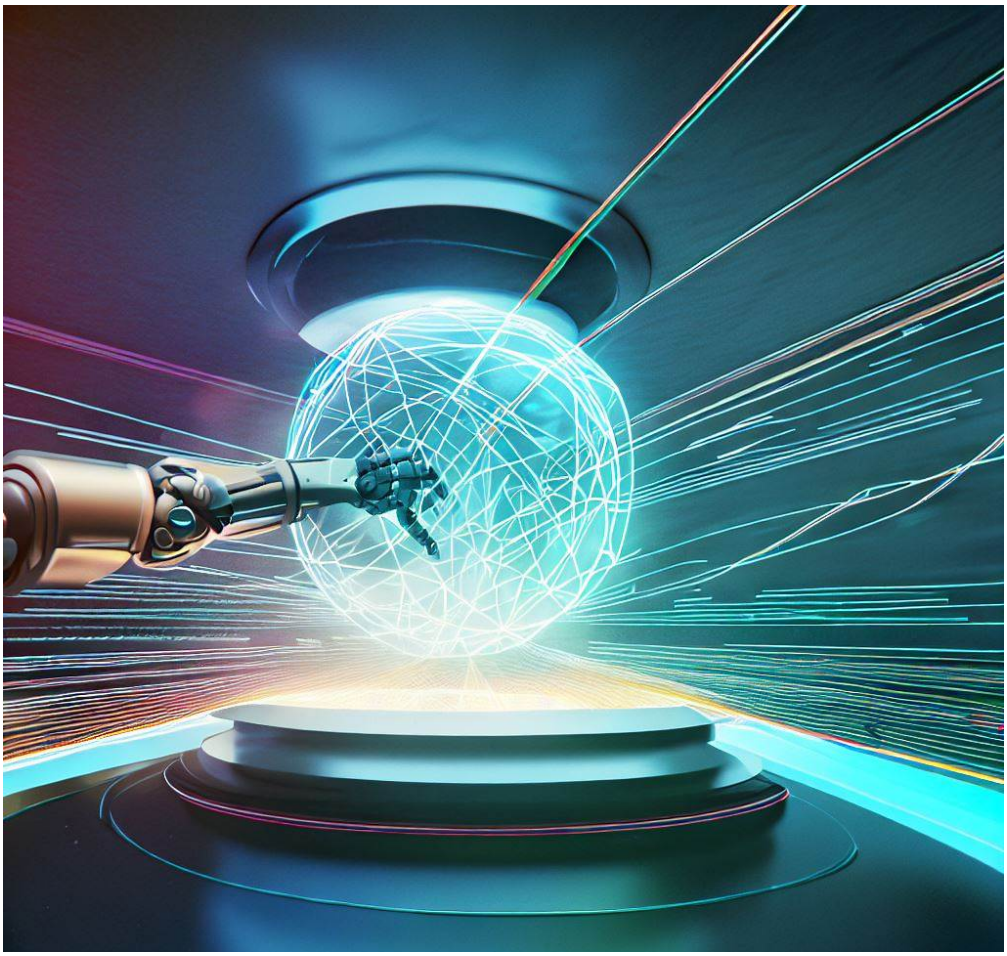
1.25 1.7.1 Applications and Threads

1.7. Application, PID and TID Previous 126 Top 41 Next 150

Starting Guide

Applications, PID and TID

Understanding Applications, Processes and Threads



The running processes and Threads make all things that computers do.

What can we do with applications using the robot-scripts?

We can start an application in multiple ways. For example using:
EXA. - Execute Asynchron 872
EXB. - Execute-Blocking 876

We can kill an application using the commands:
KPN. - Kill Process by Name 2904

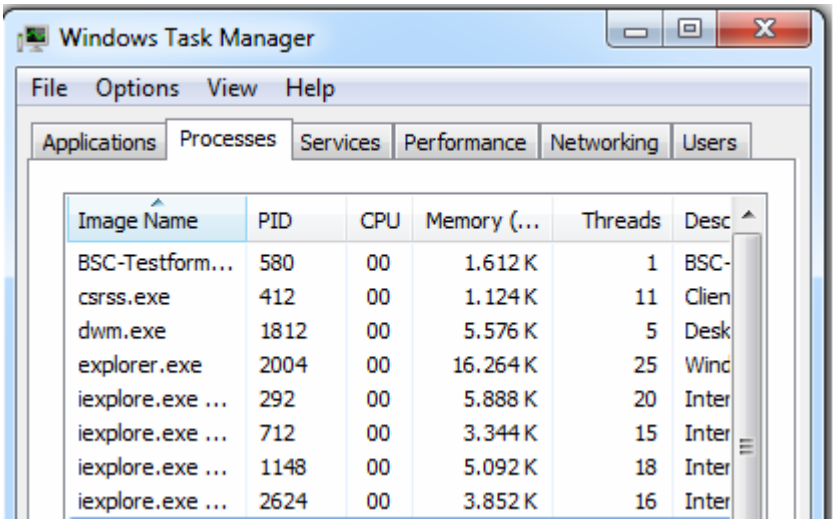
We can check if an specified application is actually running.
IEP. / NEP. - If-Exists-Process 2900

© 2023 Theo Gottwald * https://www.smart-package.com

And finally we can enumerate/count all running processes using
FEP. - For-Each-Process 932

Which is less often used.

Lets take a look into the task-manager, which applications/processes are currently running:



What we can see is that there are several processes with the same name "iexplore.exe".
But how do we find the one we need?

How can we select a specific process/application?

Rather then selecting a process "by name", we can use the process-ID (PID).
The PID is a number that is unique to each process. Having the PID, we can work with the process.

Using the PID we can do things like:
CPR. - Close-PRocess 2882

Find out which user has started the process:
GPU. - Get Process User 2898

we can wait until a process ends. Using
WPT. - Wait-Process-Termination 2913

If we have just started a process - for example with -
EXE. - Execute Programm 883

then we can use
WPR. - Wait-Process-Running 2911

© 2023 Theo Gottwald * https://www.smart-package.com

to be sure the process we need is running. A PID is a magic number when dealing with applications and processes. If you have the PID, you can do anything you want with that process. As long as you have the sufficient rights.

How do we get the PID from an process or application?

1. We may get the PID directly from the starting command. Only starting commands that wait until the process is running, can support filling #pid#. Therefore see the description of the starting command if it supports #pid#.

If we use:

```
EXE.?pfad\UltraVNC_1.0.9.6.1_Setup.exe
VAR.$$PID=#pid#
MBX.MY PID is: $$PID
```

we get the PID in the system variable `#pid#` automatically.

2. We get the PID from any window. It doesn't matter if its a Top-Level window or a control (child-window). If we have the window, we can get the PID like this:

```
STW.t|Testapp
' We'll put the PID of the process from that window
' into the variable $$PID using the GDF.-command
GDF.ap|$$PID
```

It does not need to be a Top-Window, it can be any actual localized window, try this

```
STW.t|Testapp
SCW.c|butto
GDF.ap|$$PID
' Will print the PID in the Debug-Area of the Editor
DBP.$$PID
ENR.
```

3. We can get the PID from the name of the EXE-File

```
VAR.$$FIL=?exepath\MR01.exe
GDF.fp|$$FIL|$$PID
' Will print the PID in the Debug-Area of the Editor
DBP.$$PID
ENR.
```

or we can use `IEF.` for that.

```
' Get the PID using IEP.
IEP.Testapp_01.exe
     MBX.#pid#
ELS.
     MBX. That Application is not running!
EIF.
ENR.
```

`GDF.` works with any process in the Task-Manager, no matter if that process has a window or not.

© 2023 Theo Gottwald * https://www.smart-package.com

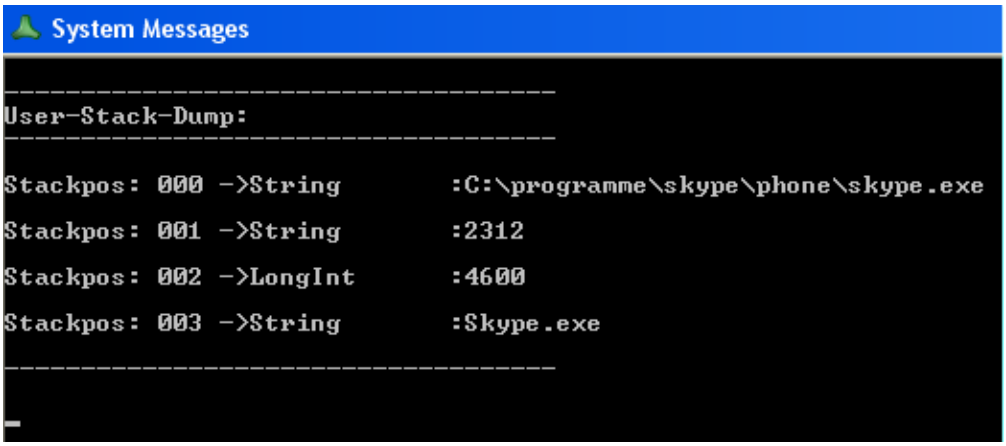
What makes IEP. (If exist process) so powerful?

IEP. primarly helps you to find out, if a process is running. For this you can specify the process in several ways, including the use of 3.2 Standard-Search Pattern 341.

Assume you use IEP. to check if there is a process like this:

```
IEP.sky
  DMP.6
EIF.
MBX.Ready

ENR.
```

Now IEP. finds such a process. But what is the real name of the process?
sky ... what?
We have just specified "sky". As IEP. uses Standards Search Patterns, any part of thereal name is valid.
The name of the process must only contain that phrase. We get this information if we take a look on the actual stack.



`IEP.` provides us with a lot of information about the found process. In position zero

```
IEP.x64.exe
   MBX. Process ID is: #pid#
EIF.
```

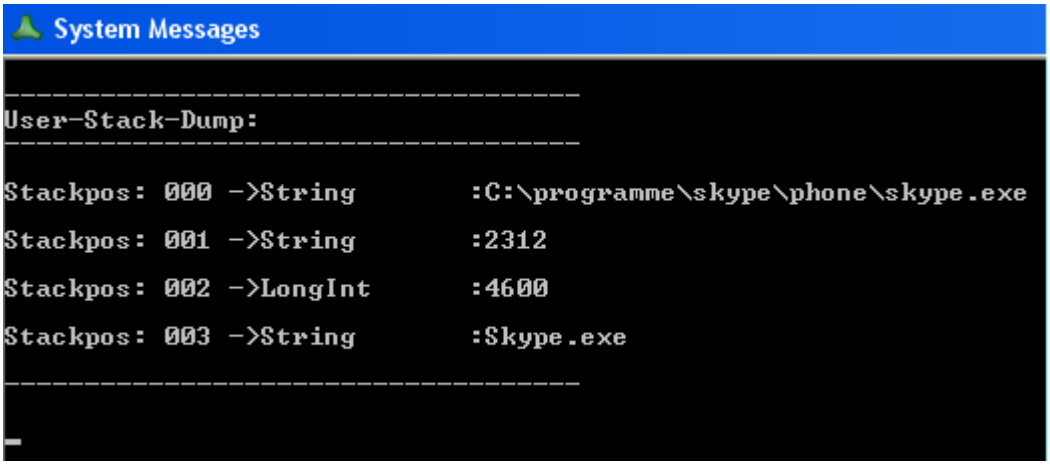
If you have multiple EXE running with the same name, you can specify a number. If the EXE is found, the #pid# is set to the EXE's Process-ID. Please note that #tid# is always set to zero.

You can specify a full path, or only the filename or a part of it. If you search for a dll, then the extension ".dll" must be included in the pattern.

You can use `NEP.`, the negative Form of this command (`NEP.` - Not Exists Process).

© 2023 Theo Gottwald * https://www.smart-package.com

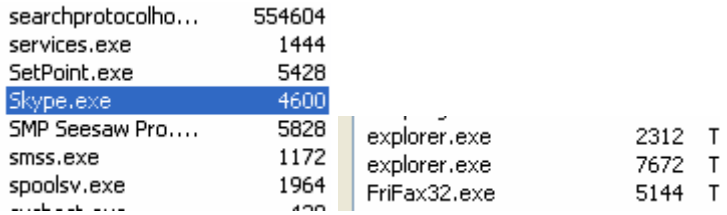
If a process is found, valuable information is placed on the actual stack. These include the full name and path of the found executable.



Below are two snapshots of the windows-taskmanager. Skype.exe has the PID (process-id) 4600. Its on the stack on position 02 in the picture above.
The 2312 which is on position 01 (picture above) is the parent-process, that is the process which has started skype.exe.
In the task manager, we can see that this is the PID of the windows-explorer. This way you can find out which process has started another process.

If you give IEP. a variable as third parameter, then the stack will not get the full path of the found file. You will find it in the given variable instead.

Please note that the 2.4 The global and local Stack 369 as Que or as Stack will influence the sequence in which the items appear on the stack.



You can also use IEP. to verify, if a process with a specified **PID is alive** or not.
To do this, the Syntax is:

```
STS.GLOBAL
IEP.>1234
   MBX.PID 1234 (Filename=$$000) is alive!
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

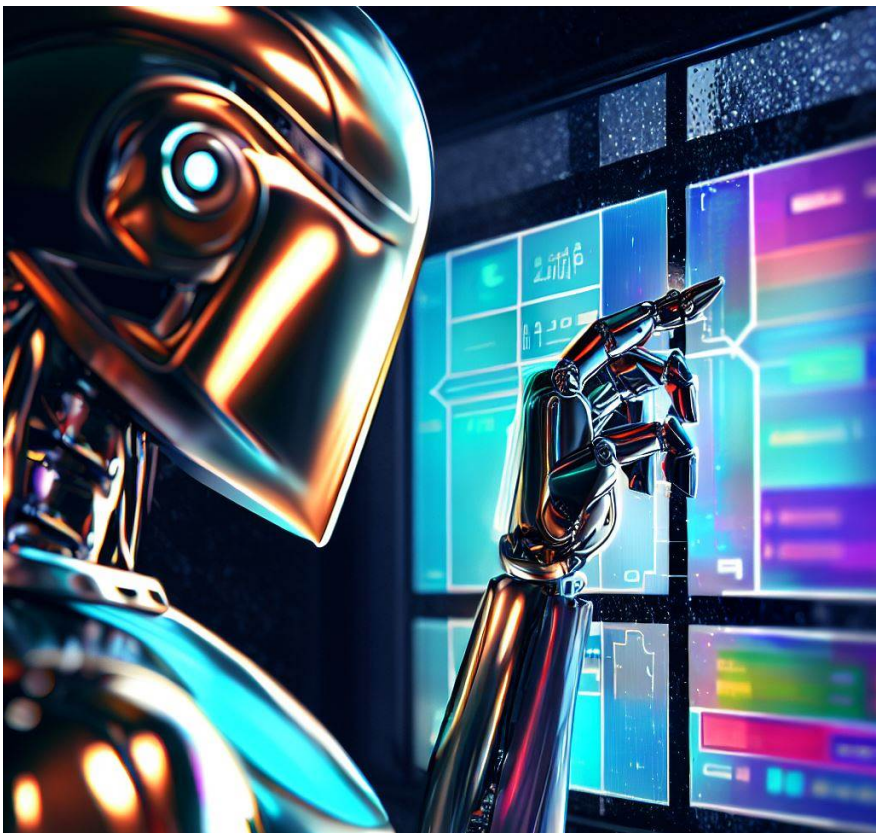
1.26 1.8. Selecting the Application (Topwindow)

1.8. Selecting the Application (Topwindow) Previous 137 Top 41 Next 156

Starting Guide

Selecting the Application Locating the Topwindow

Choosing the Application: Understanding and Interacting with Windows



On your command will the robot interact with Applications and windows.

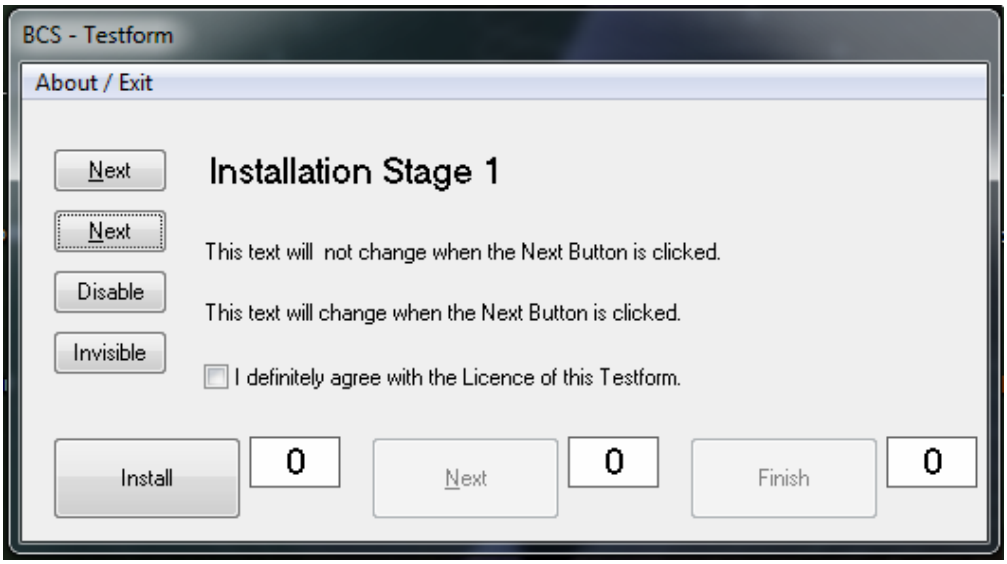
How do we instruct a robot to select a specific window for interaction?

At first glance, the concept of a 'window' might seem rudimentary, given our daily interactions with them. We routinely open, close, resize, minimize, and maximize windows in conjunction with applications. However, there's more to windows than meets the eye. Let's delve deeper into understanding what windows are and their significance.

Exploring Application Windows: The Gateway to Software Interaction

Before we proceed with our discussion on windows, let's take a moment to visually explore a window.

© 2023 Theo Gottwald * https://www.smart-package.com



The "Sample Apps" subfolder in your installation directory contains a variety of test applications. However, what you're interacting with is not the application itself, but its window. Since we control the application via the window's buttons and controls, it's easy to conflate the two. To delve deeper into the distinction between processes and applications, refer to Chapter 5: 5. Application and PID 32.

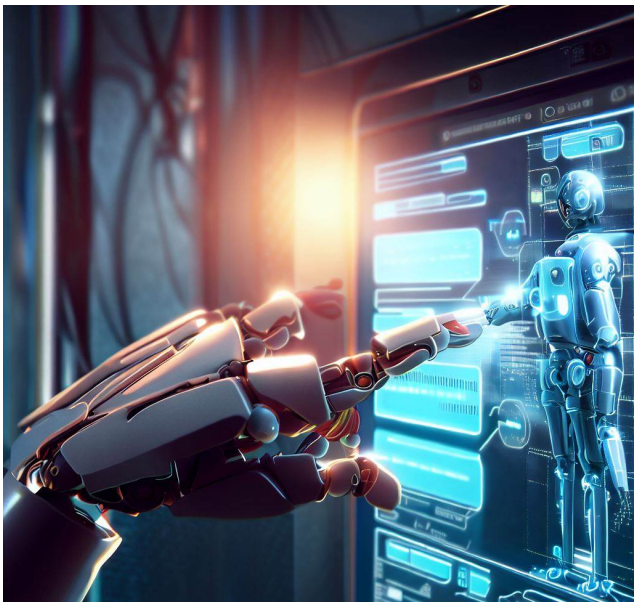
So, what can we do with these application windows?

Here's a rundown of the various actions you can perform:

1. Launch it by starting its associated application.
2. Move it around to adjust the window position on your screen.
3. Resize the window to fit your viewing preference.
4. Minimize it to temporarily remove it from view while keeping it running.
5. Maximize it to occupy the entire screen for a more immersive experience.
6. If it's minimized or maximized, you can "normalize" it back to its original size and position.
7. Close it to terminate the application.
8. Adjust its Z-Position, allowing you to bring it to the forefront or send it behind other windows.
9. Interact with its elements according to their roles, such as pressing a button or adjusting a slider.

Most of these actions can be performed effortlessly with your mouse.
And also the robot can do all of this.

© 2023 Theo Gottwald * https://www.smart-package.com

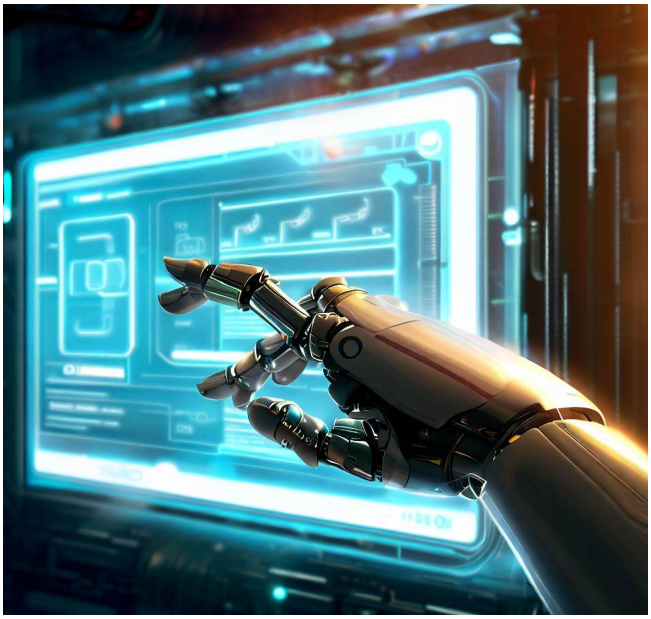


Harnessing Advanced Capabilities with the Robot

Utilizing the robot provides us with an expanded set of options for interacting with windows. With the robot, we can:

1. Disable any window, effectively preventing any interaction with it.
2. If a window is disabled, we can enable it, restoring its functionality.
3. Make the window invisible, hiding it from view without closing it.
4. If a window is invisible, we can make it visible again, bringing it back into view.
5. Change the text displayed in the window.
6. Perform a variety of other actions, depending on the specific type of window we're dealing with.

These advanced capabilities allow us to manipulate and control windows in ways that go beyond standard user interactions.



© 2023 Theo Gottwald * https://www.smart-package.com

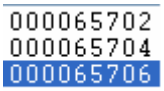
Unveiling the Secret of Windows: The Window Handle

Now, it's time to share a little-known fact about windows. Consider the analogy of a newborn child. Upon birth, the child is given a unique name that they carry throughout their life. If you wish to call upon the child, you use their name. Similarly, windows receive a unique identifier at their creation, but in their case, it's a number, not a name. Why a number? Because computers process numbers much faster than names.

Each window is assigned a unique number, known as the "window handle." This number is guaranteed to be unique, much like a person's name, but it's a numerical value, making it easier for the computer to process. The window handle allows you to interact with the window, much like using a person's name to get their attention.

So, if you want to instruct the robot to interact with a specific window, all you need is the window handle. It's unique and concise, making it an efficient way to identify windows.

Let's take a look at some examples of window handles before we proceed.



The Dynamic Nature of Window Handles

Indeed, using the window handle, such as

```
STW.h|65702
```

makes it straightforward to select a specific window for interaction.
It guarantees that you will access the exact window you desire.

However, there is a crucial aspect to consider: the lifespan of windows. Each window is created when you start your program and ceases to exist when you close your program or power off your computer. Here lies the challenge: Every time a window is born, it is assigned a new and unique window handle.

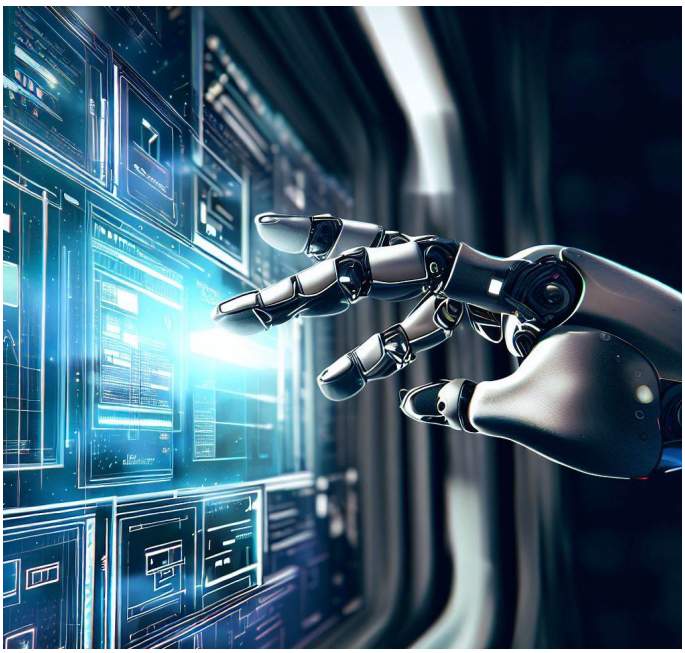
Knowing this, we must acknowledge that our script,

```
STW.h|65702
```

will not work if we run it again after closing the application and restarting it.
The window handle associated with the desired window will have changed due to the window's new instantiation.

Therefore, it is essential to consider the dynamic nature of window handles and adapt our approach accordingly.
But then ...

© 2023 Theo Gottwald * https://www.smart-package.com



How do we really select a window to deal with?

To effectively select a window for manipulation, we must obtain its unique handle. Once we have the window handle, we gain the ability to interact with the window as desired. The process of obtaining the handle involves "locating" the window and determining its handle and position.

Fortunately, we do not have to perform this task manually. The robot can handle it internally by retrieving a list of all available windows and selecting the specific window we desire.

To achieve this, we provide the robot with a set of window parameters that uniquely identify the desired window. These parameters should remain consistent, regardless of when or where the script is executed.

In many cases, we use the window's caption text as a parameter. For example:

```
' Locate the window
STW.t|BCS - Testform
' Store the window handle in the variable $$MYH
HTV.$$MYH
```

When the window becomes available, the robot will locate it and internally store its handle. We can then store the handle in a variable, such as `$$MYH`. From that point on, whenever we write:

```
STW.h|$$MYH
```

we can be confident that the script will locate the exact window we desire—the window with the handle stored in `$$MYH`.
If you wish to see the window handle, you can add the following line to the script:

```
' Locate the window
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.t|BCS - Testform
' Store the window handle in the variable $$MYH
HTV.$$MYH
MBX. The handle is: $$MYH
```

In practice, we often specify multiple parameters for increased reliability. This ensures that we locate the correct window, even if multiple windows share the same caption on the desktop.

Some commonly used window parameters include:

1. Window class: This parameter specifies the window class, which may look like:
   `STW.ct|PBWindowClass|BCS - Testform`

2. Window ID: Please note that the window ID should not be confused with the window handle. The window ID is often a unique and permanent number, but its usability varies depending on the window class. It is advisable to avoid using the window ID, especially if the window class begins with a "t," as in
   `"tnewcheckbox"`

3. Process ID: You can instruct the robot to select windows only from a known process ID (PID). If your application has only one window associated with its PID, you can write:

```
STW.p|2320
```

to locate the desired top window, regardless of whether it is a 32-bit or 64-bit application. If multiple windows share the same PID, there is no guarantee which window will be selected, although it is likely to be the topmost window.

Therefore, locate a Top-Window best with a combination of these parameters. There are a lot of parameters available, see the

The SPR-Editor's recording option and the Installation Recorder InstaRec are valuable tools that simplify the process of locating windows. With their automated functionality, they can intelligently choose the most suitable variant to locate the window, streamlining the window identification process. This feature saves time and effort by automatically determining the appropriate parameters based on the specific window being accessed. Whether you're using the SPR-Editor or the Installation Recorder InstaRec, these tools provide a convenient and efficient way to handle window location in your automation tasks.

1. STW. - Decription 3814

The next step is, that you learn about Child-Windows and how to locate then in the chapter:

7. Locating Child-Windows 32

* the Installation folder is the folder, where you have installed the robot program.

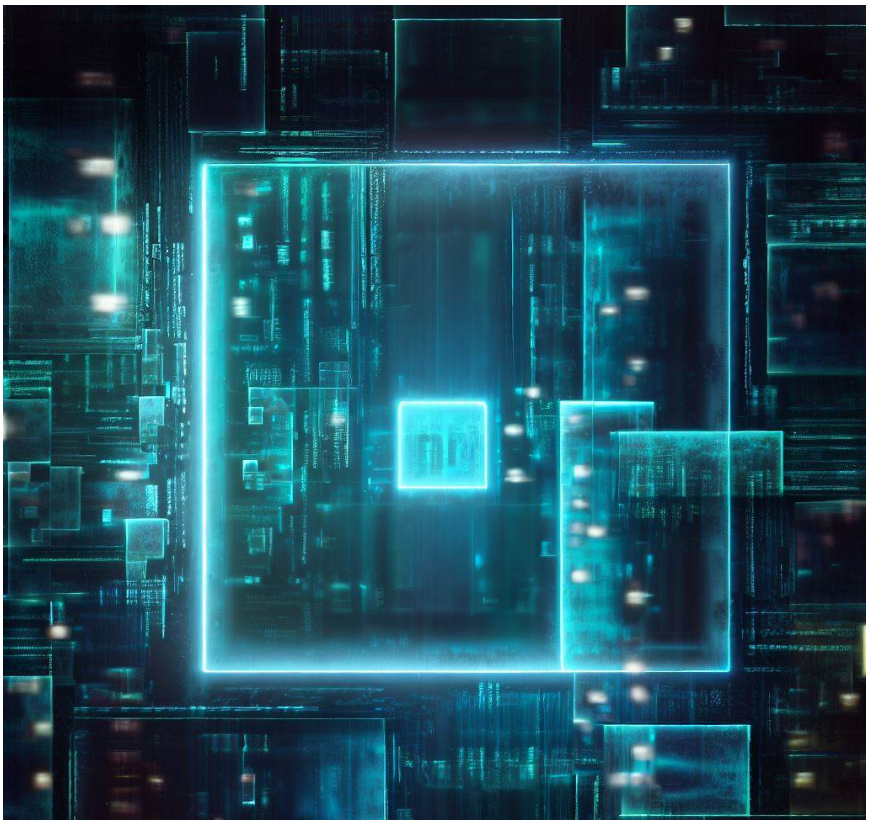
© 2023 Theo Gottwald * https://www.smart-package.com

1.27 1.8.1 The "Client" and the "Non-Client" Parts of a window

Starting Guide

"Client" and "Non-Client" Area

Whats it, and what's it good for?



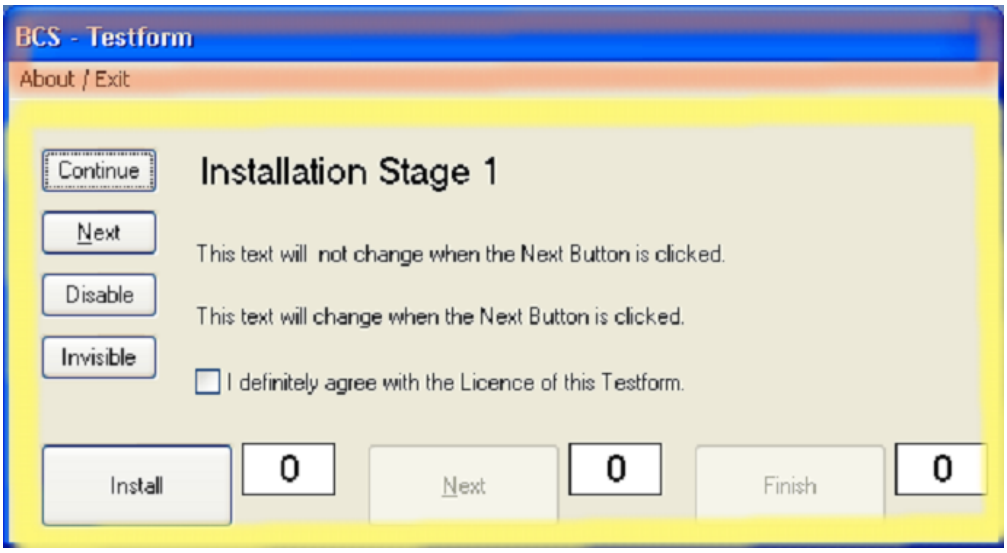
What is the "Client Area" of a window and why do we need it?

To answer this question best, think a moment like a developer of a windows-application.

Building a windows-application a developer can place controls, statics and pictures inside a window. But not really anywhere in the window, but only in the so called "Client Area".
That is the area, where a Software developer can place its controls.
The "Non Client Area" is the common part, that is equal to all windows. The parts that are drawn from the OS, to some degree independent from the Software in the "Client Area".

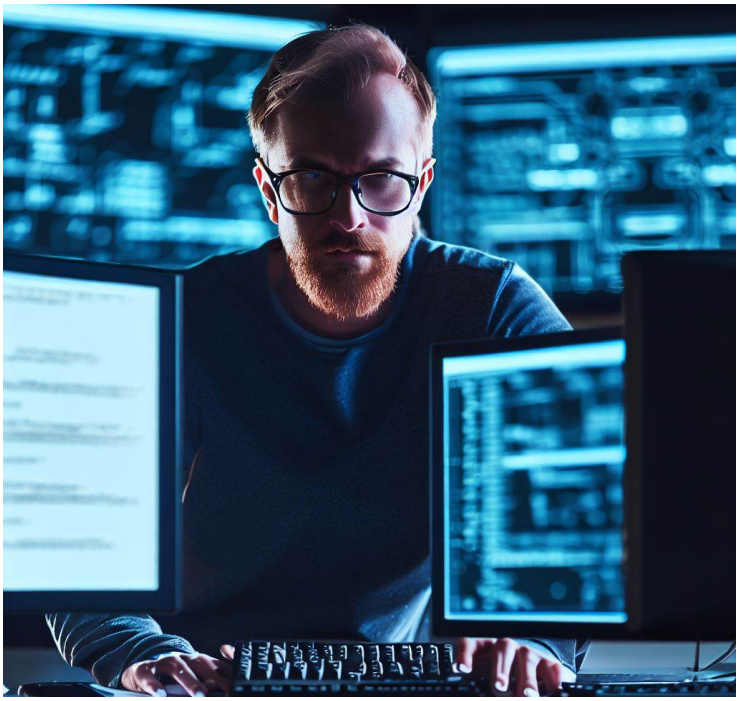
Lets take a look on the window below and see the difference between the "Client-Area" (yellow) and the "Non-Client Area" (orange color).

© 2023 Theo Gottwald * https://www.smart-package.com



As you can see, the following parts of a window belong to the "Non-Client Area":
- Complete Window Border
- Sizing, Foreground, Help, Close and Minimize Gadgets
- System Menu
- Real" Top-Menus, means Menus which belong directly to the Top-Window

The rest of the window is under the control of the software developer and is as such called "Client Area".



Here comes the "input focus" ...

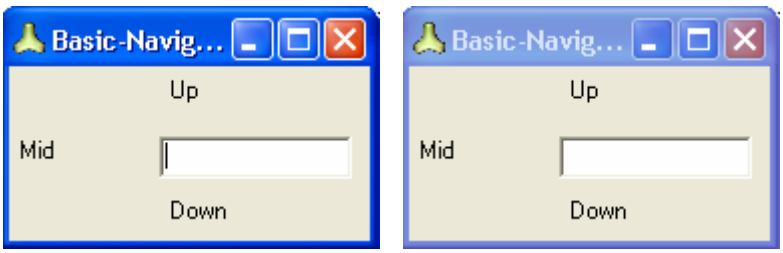
- When taking a Screenshot from a window, we must know if want the complete window or (as most often) just the "Client Area" of the window.

*© 2023 Theo Gottwald * https://www.smart-package.com*

When using IRA. or WRA. we should work with the Client Area only.

Otherwise simply giving the windows the "Input focus" would change the color of the "Non-client area. You can see this below.

The window below on the left side has the "Input focus". Please note the blinking cursor inside the textbox. The picture on the right side shows the same window without the input focus. As you can see the "non client area" has a completely different Color.



Note that also a blinking cursor like the one in the left picture may cause IRA. /WRA. and such, to notify "a change" in the client area.

Coordinate Systems

Coordinate Systems in our sense are 2-dimensional. They describe a location on the screen using an
X - value (Horizontal position)
Y - value (Vertical position)

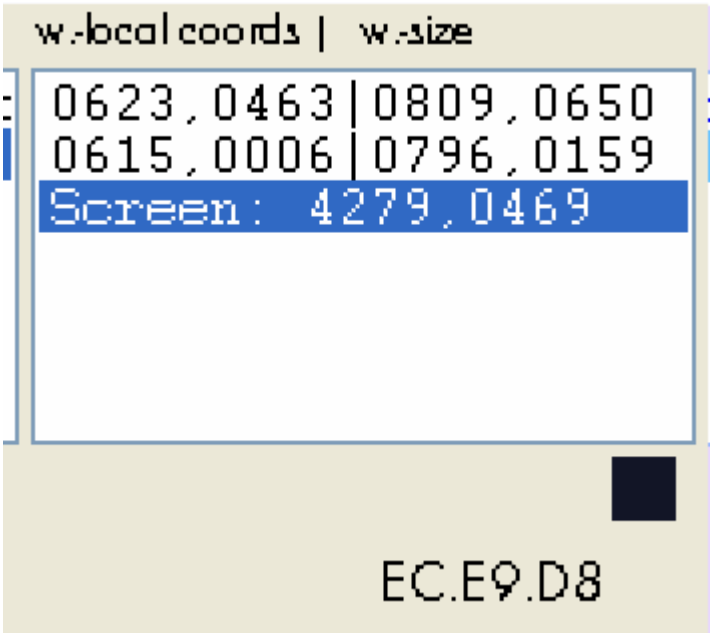
Our Coordinate systems generally have their zero (0,0)-Point in the upper, left corner.

We have three coordinate systems in use with windows.

© 2023 Theo Gottwald * https://www.smart-package.com

1. Screen-Coordinates

Screen-Coordinates have their zero (0,0)-Point in the upper, left corner of the desktop.
The total and maximum resolution is the Screen-Resolution.
If you have multiple Screens (Dual Screen), then the coordinate system goes over all available Screens.
The Screen-Coordinate where the mouse currently is, is displayed in the Windows-Inspector behind the "`Screen:`", see here:



© 2023 Theo Gottwald * https://www.smart-package.com

Screen coordinates have some disadvantages.

For example, if you have the coordinates of a button, and you move the window around, the screen coordinates are not more accurate.

Therefore we have a second coordinate system:

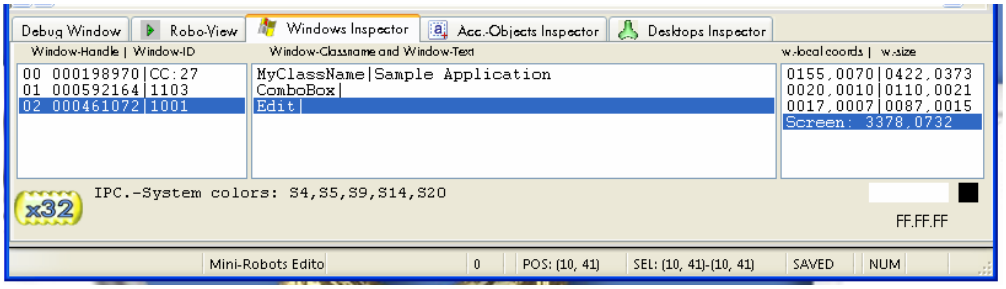
---

2. Window-relative coordinates:

These coordinates have their origin (0,0-Point) in the upper left corner of the "non-Client-Area" of the window. Therefore if you move the window, they will move with the window.

Window coordinates have the advantage that you can reach any part of a window, whether in the client- or in the Non-Client-Area. That's the reason why we use them most often. And their main-advantage is that they move with the window.

You can see the window coordinates right of each child- or Top-window in the windows-inspector. See the numbers on the right side. They show you first the current Mouse-Position in window-coordinates. And right of that you see the dimensions of the child-window.

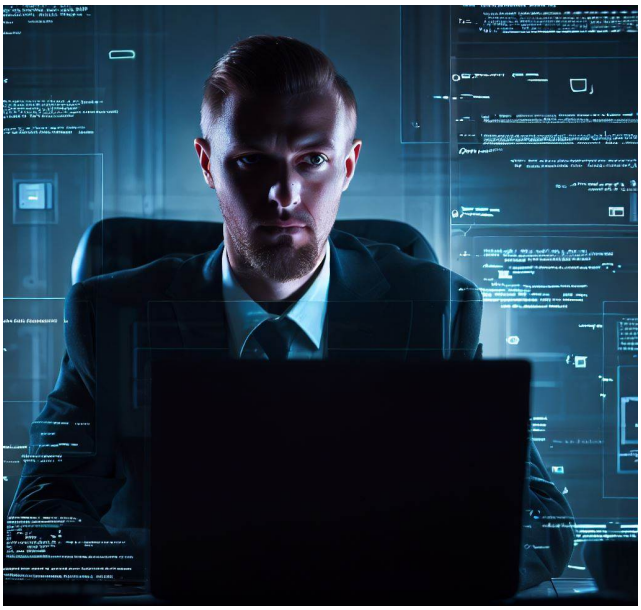


© 2023 Theo Gottwald * https://www.smart-package.com

But as anything, windows-coordinates have a small disadvantage when it comes to different Window-OS'es.

Imagine that the Title-bar would have another size on another window-OS.
In this case the Window coordinates will not be accurate anymore. This way, they may not work on another windows-system.

We have a third coordinate system that overcomes this problem:



3. Client-Coordinates

These coordinates have their origin (0,0-Point) in the upper left corner of the "Client-Area" of the window.
Therefore if you move the window, they will move with the window like window-coordinates. And they do not depend on the styles of the windows-outline. Because they only contain the "Client-Area".

The disadvantage of Client Coordinates is, that you can not easily get to any point outside the client Area. In fact you could do that by using negative coordinates.

Currently we do not use Client Coordinates, they are however implemented in the `CST.`-command so you can use them your self if you want.

*© 2023 Theo Gottwald * https://www.smart-package.com*

1.28 1.9. Locating Child-Windows

Starting Guide

Childwindows and how to locate them

What are these child-windows and why do we need them?

What is a child-window and why do we need it?

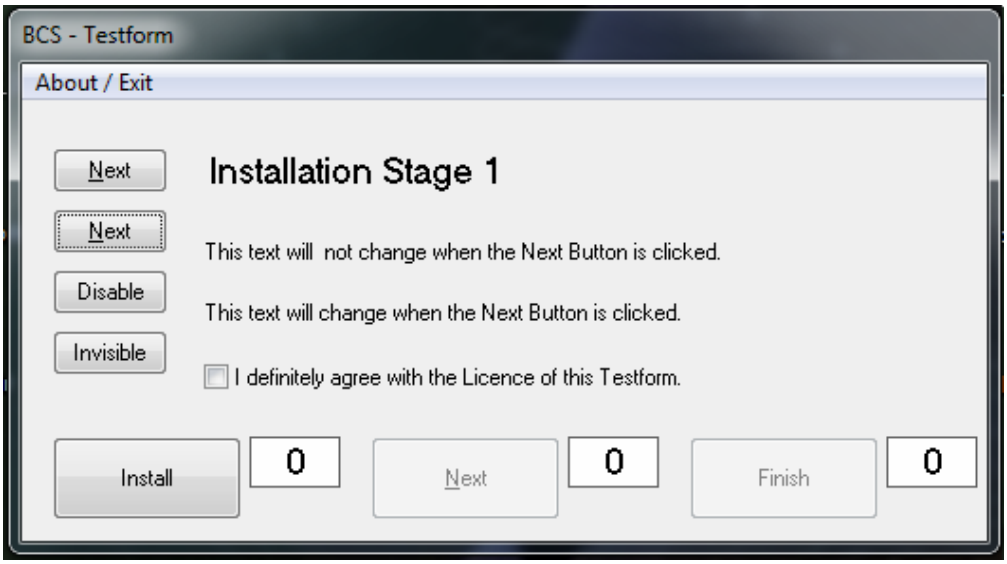
Before you start reading this chapter, make sure you are familiar with
6. Selecting the Application (Topwindow) 32

At this point i assume that you know the terms:
- Top-Window
- Window-Handle

and you should also be familiar with the chapter
5. Application and PID 32
means you should know what a process is and what a PID is.

Again we take a look on a window here.

© 2023 Theo Gottwald * https://www.smart-package.com



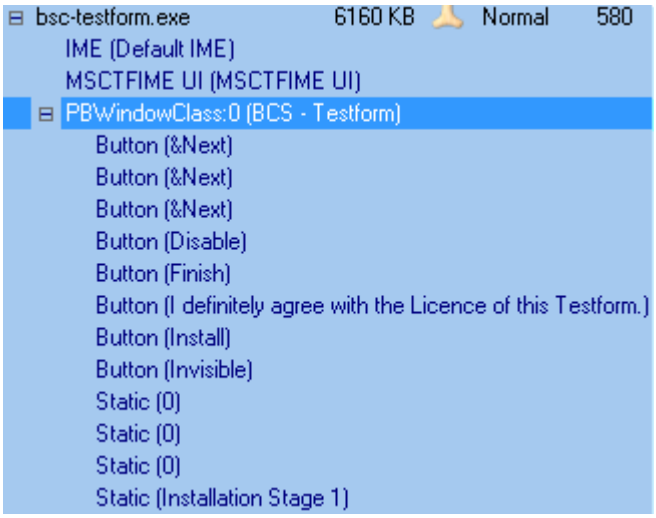
This is one of the Test-Applications in the "Sample Apps" Subfolder of your Installation folder*. In fact its not really the Application its just the Application window. Because we control the application via the window buttons and controls, we often think of the application as the window.

In this chapter we are not going to talk more about the whole window (the Top-Window). Instead we will take a look on the buttons and text-boxes on this window. These are the controls with which you control the application. And in fact, that is why they are called "controls". You can read their text. See if they are enabled or disabled.
You can press a button or type text into a text box (while not in this application).
And now let me lift the secret about Child-Windows.

What are "Child-Windows"?

So far it belongs to us, we can safely assume that Childwindows and "Controls" are just the same. A button is a child window. A text-box is a child-window. There are 50 child-windows in that Top-Window we see above. Technically for windows the Hierarchy of "Process, Top-Window and Child-Windows" look like this:

© 2023 Theo Gottwald * https://www.smart-package.com



To understand it a bit better, we need to know that the Windows OS protects the Memory in which an Application resides from any access through other applications. If an application tries to read or write behind that Protection Border, you will see that application crash with one of those GPF's (General Protection Fault) Error Messages you know.
And now imagine that in this protected Application Memory there is everything inside that belongs to that application, that includes the application windows.

Now back to the picture above. Below the Top-Window you can see a lot of Buttons and other Child-Windows. They are childs from that Top-Window. If the Top-Window is been destroyed (closed) all of these Child-Windows are also history.

What can we do with child-windows?

- we open them together with their Top-Window, by starting their application
- we can handle them according to their role (press a button, pull a slider etc.)

Using the robot you get advanced powers with child-windows:

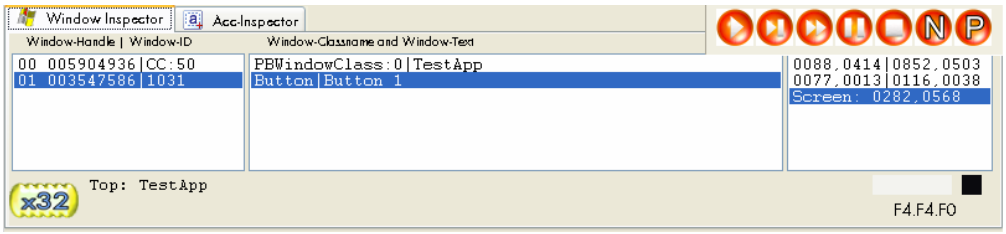
Using the robot we have more options, about what we can do with child-windows.
We can:
- change their window position and size
- minimize it (yes you can really minimize a child-window using the `SWS.m` command)
- maximize it
- if it was minimized or maximized, we can "normalize" it, to its normal size and position
- and finally we can close it
- we can disable any window
- if it was disabled, we can enable the window
- we can make the window invisible or
- if its invisible, we can make it visible again

we can also

© 2023 Theo Gottwald * https://www.smart-package.com

- change its displayed text using the `SDT.`-command
and do a lot more such things, depending on what sort of window we have to deal with.

Now its time to tell you a secret about child-windows.
This time its maybe not really a secret, if you have read the chapter about Top-Windows. Yes, its true. Child-Windows do also each have its unique window handle. A button has a window handle. As well as a checkmark or a textbox or whatever. They just look exactly like window-handles from top-windows. You can not see any difference.
Only if you look into the windows-inspector you will see that ...



The Top-Window is at the Top of the List and has a Level (the number on the very left) of "00". The child windows are below the Top-Window. They start with a Level of "01".

The Inspector will show you all windows below the mouse-pointer. He will show you the Level, the window handle and in the first line you see a "CC:50". This means that this Top-Window has 50 Child-Windows. Next in the Middle part you will see the "window class" and then a Pipe-Symbol (|) followed by the window-text.

On the right side in the Inspector you will see the actual Mouse-position relative to the window in which the mouse is in. And then again a pipe followed by the size of that window/child-window.

Below you can see the position of the mouse in Screen-coordinates (relative to the desktop window). "Screen: 0282,0568".

Now, how do we locate Child-Windows (buttons etc.)?

We do it using the `SCW.` command and in the same way like with Top Windows. While we have some more options with Child-Windows, then we had with Top-Windows.
Top-Windows do mostly not have a usable Window-ID.
Top-Windows do not have a "depth" because they are always on Level "00".
Child-Windows may have a valid ID and they always have a (depth=) Level of "01" or higher. Because they re below a Top-Window or below other Child windows in the Window-Hierarchy.
These are additional parameters we can use when addressing child-windows.

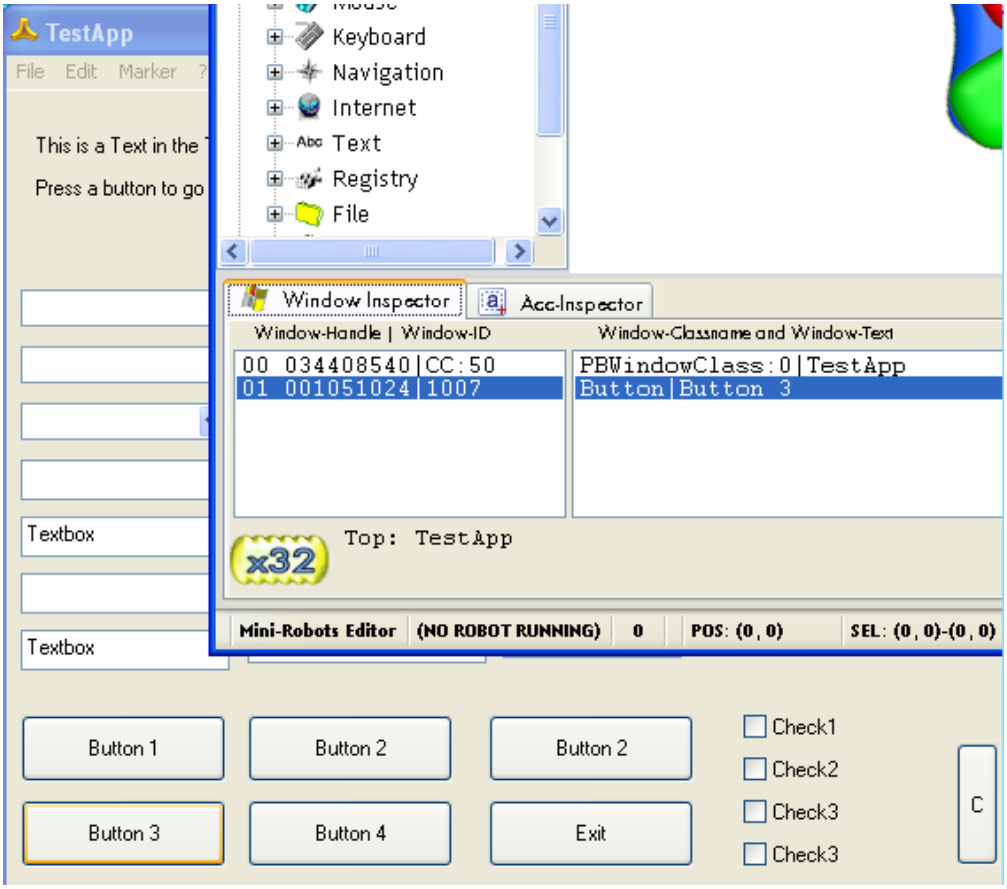
© 2023 Theo Gottwald * https://www.smart-package.com

Knowing all this, we can locate a Top-Window and a button inside this Top-Window like this:

```
STW.ct|PBWindowClass:0|BCS - Testform
' Locate the Child-Window
' c - the class must contain the word "Button"
' t- the text on the control must contain the phrase "&Next"
' w - wait up to 5 seconds for the window
SCW.ctw|Button|&Next|5
' Mouse Move to the center of the located window rectangle
MMV.
```

If you have problems to understand the parameters in the SCW., see the chapter
4. The Prototype Parameter System 240

The final test
Now lets if you can really do it.Lets make a test.You know there is a built in recording feature, but this time we will make it "from hand".
Take a look on the picture below.



It shows the windows-Inspector while the mouse is over "button 3" in our TestApp. This is the button we want to click. Lets make a complete script to locate a child-window "button 3" and click on it. As said before, you find the TestApp in the "Sample Apps" folder in the Installation folder* of the program. I recommend that you start it now, then we can make a practical exercise together.

How do we start?

© 2023 Theo Gottwald * https://www.smart-package.com

We always need to locate the Application-Window (Top-WIndow) first. This is done using the `STW.` command.

What parameters will we use?

We will use the window caption "TestApp" and we'll use a piece of the window class. As we can see in the Windows-Inspector, the complete window-class is "`PBWindowClass:0`". We do not want to write so much. We just need a unique part out of it. We take "`BWindowC`". This is completely valid as we can always just specify a part of a text.

We write:

```
' "bwindowc" is a part of the full classname as its shown
' in the Window-Inspector
STW.tc|TestApp|BWindowC
```

Now lets try if this works. We add one line:

```
STW.tc|TestApp|BWindowC
MMV.
```

and start our script. What happens?
The Mouse-Pointer will drive into the center of the located window.

To be 100% sure we have located the right window, we can use this:

```
STW.tc|TestApp|BWindowC
HTV.$$MYH
```

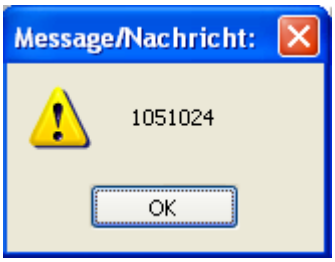
and start our script. We get this result. Watch number 34408540. Its exactly the same number, thar we see in the windows-inspector for Level "00". This proofs that we have located the right Top-Window. No doubt!

We are now inside the Application window. Now we must locate the button. We want to locate "button 3". In this case we will use the window-class and a text-part for the SCW.-command.
We write this:

```
STW.tc|TestApp|BWindowC
SCW.ct|butt|n 3
MBX.#hwn#
```

and start our script. We get the result below.

© 2023 Theo Gottwald * https://www.smart-package.com



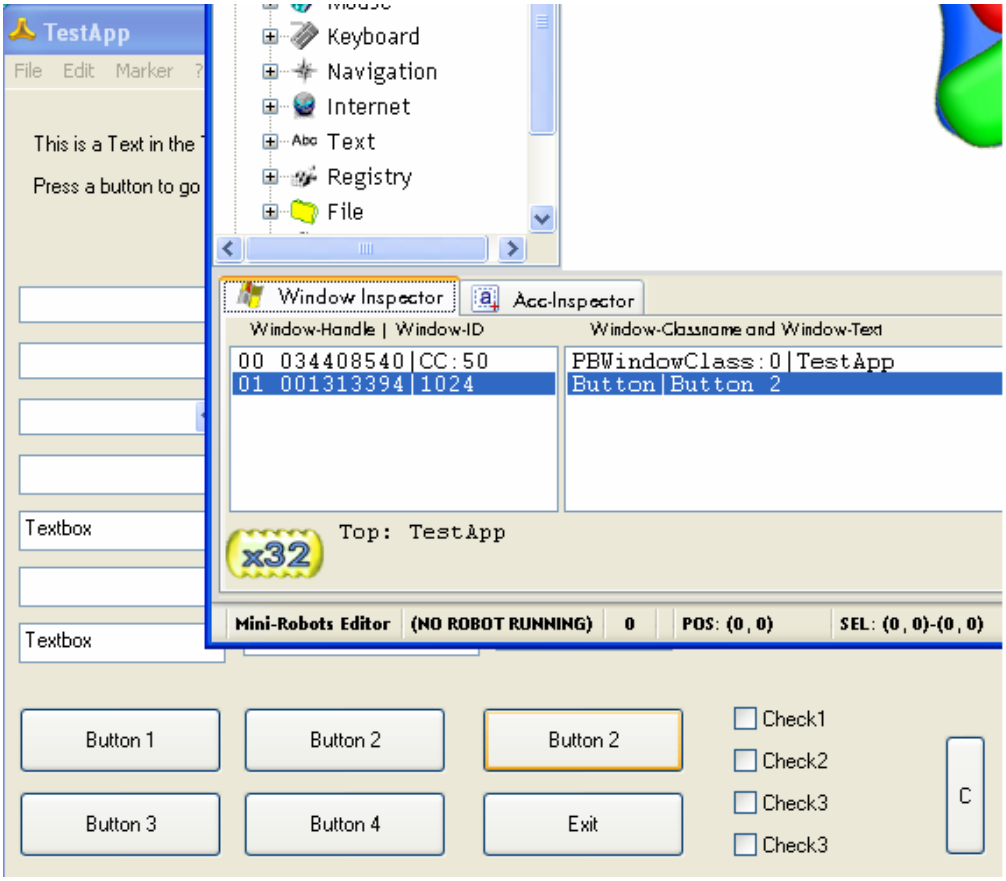
again this is exactly the same number that we see in the window inspector. This proofs that we have located the right window. Now lets make a click on it. To click on a button or whatever, we have many choices, how to click. In this case we will make a very normal Click with the left mouse-button. Just as if we would have done it manually. We write.

```
STW.tc|TestApp|BWindowC
SCW.ct|butt|n 3
MLC.
```

Start the script and watch what happens.

Ok, we will take one more example from this Area. Take a look on the picture below. As you can see, we have two Buttons "Button 2". How can we handle this situation? This time, we will use the "n" flag. It gives us the option to count the results and use result Nr.?, in our case we want to click on the second "fitting button". Let's try this first without click, just by moving the mouse there.

```
STW.tc|TestApp|BWindowC
SCW.ctn|butt|n 2|2
MMV.
```



© 2023 Theo Gottwald * https://www.smart-package.com

But ... the mouse moves to the first button! What is wrong here?
The answer is: nothing is wrong. We have no warranty, that the optical placement of a button anywhere in a Window is consistent with it's "count number". Technically the question is rather in which sequence the buttons have been created by the application. But we do not know the sequence, because this is up to the programmer. What we do is, we will just change the number to "1".

```
STW.tc|TestApp|BWindowC
SCW.sctn|butt|n 2|1
MMV.
```

That was the whole secret. Try and find the right number. No we just need to change the Move (`MMV.`) to a Click. In this case we have chosen another Mouse-Click, that will not move the mouse around, but just click. (`MLE..`).

```
STW.tc|TestApp|BWindowC
SCW.sctn|butt|n 2|1
MLE.
```

And thats all for now!
This was the chapter about locating child-windows. Take some time and experiment with it.
I am sure you will like it!

---

* the Installation folder is the folder, where you have installed the robot program.

© 2023 Theo Gottwald * https://www.smart-package.com

1.29 1.9.1 Types of controls

1.29.1 1.9.1.1 Types of Controls

1.9.1 Types of controls Previous 174 Top 41 Next 182

Starting Guide

Types of Controls in Windows GUI

Controls in Windows GUI

What is a "control"?

In the context of Windows Graphical User Interface (GUI), controls are fundamental building blocks that are used to interact with the application. They are also known as "child windows" as they are typically contained within a parent window, which is the main application window.

Controls are used to perform various functions and to control the behavior of the application. They can accept user input, display text or images, or perform actions in response to user interactions.

Here are some of the common types of controls available in Windows GUI:

Buttons: Buttons are perhaps the most basic type of control. They are used to initiate actions when clicked.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Text Boxes: Text boxes are used to accept input from the user. They can be used to enter a single line of text, like a search box, or multiple lines of text, like a text editor.

Labels: Labels are used to display static text on the screen. They are often used to describe other controls or provide instructions to the user.

Check Boxes and Radio Buttons: Check boxes allow the user to select multiple options from a set, while radio buttons allow the user to select a single option from a set.

Drop-Down Lists and Combo Boxes: These controls allow the user to select an option from a predefined list. A combo box also allows the user to enter a custom value.

Sliders and Progress Bars: Sliders allow the user to select a value from a range by moving a slider thumb along a track. Progress bars are used to display the progress of a long-running operation.

List Boxes and List Views: These controls display a list of items to the user. The user can select one or more items from the list.

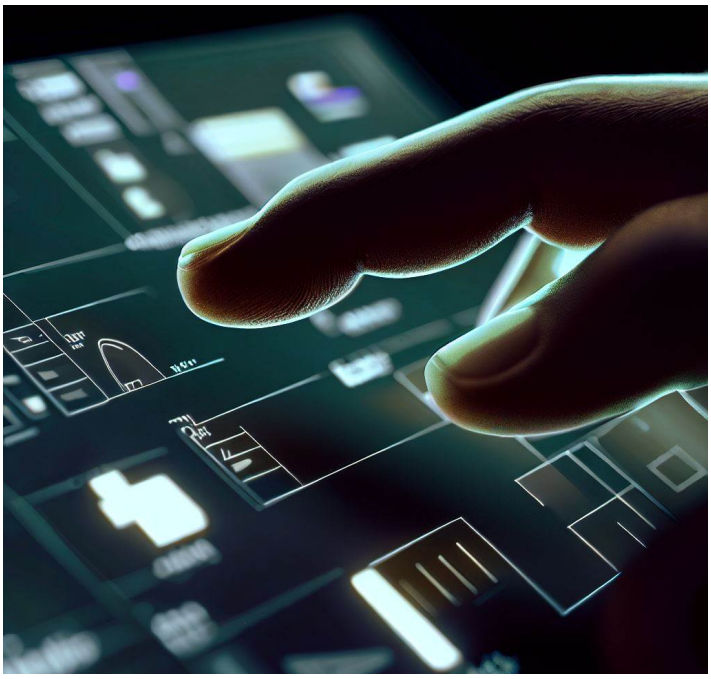
Tree Views and Grid Views: These controls are used to display hierarchical or tabular data.

Menus and Toolbars: Menus provide a way to organize commands in an application. Toolbars provide quick access to commonly used commands.

Each of these controls has its own properties, methods, and events that allow you to customize its appearance and behavior. The choice of control depends on the specific requirements of your application.

---

*© 2023 Theo Gottwald * https://www.smart-package.com*



Is Automation Limited to Controls?

Certainly not. In addition to automating controls, we can also identify "GUI elements" using Web/Browser Accessibility Technology.
Moreover, with the advent of new pattern recognition capabilities, we can automate interactions with all types of GUI elements.

Given the trend in modern software development to move away from traditional controls, it's increasingly likely that you'll need to familiarize yourself with these other technologies.
They offer a broader scope for automation and can significantly enhance the functionality and user experience of your software.

The most commonly used types of controls are:

1. Button Control
2. Checkmark Control
3. Combo Box/Dropdown List Control
4. Dialog Box Control
5. Edit/Textbox Control
6. Grid View Control
7. Label Control
8. Listbox Control
9. Menu Control
10. Page-Tab Control
11. Progress Bar Control
12. Radio Button (Option) Control
13. Scroll Bar Control
14. Slider Control

© 2023 Theo Gottwald * https://www.smart-package.com

15.Spinner Control
16.Status Bar Control
17.Toolbar Control
18.Tooltip Control
19.Treeview Control
20.Window Frame Control

In the following chapters we will take a closer look on some of these.

© 2023 Theo Gottwald * https://www.smart-package.com

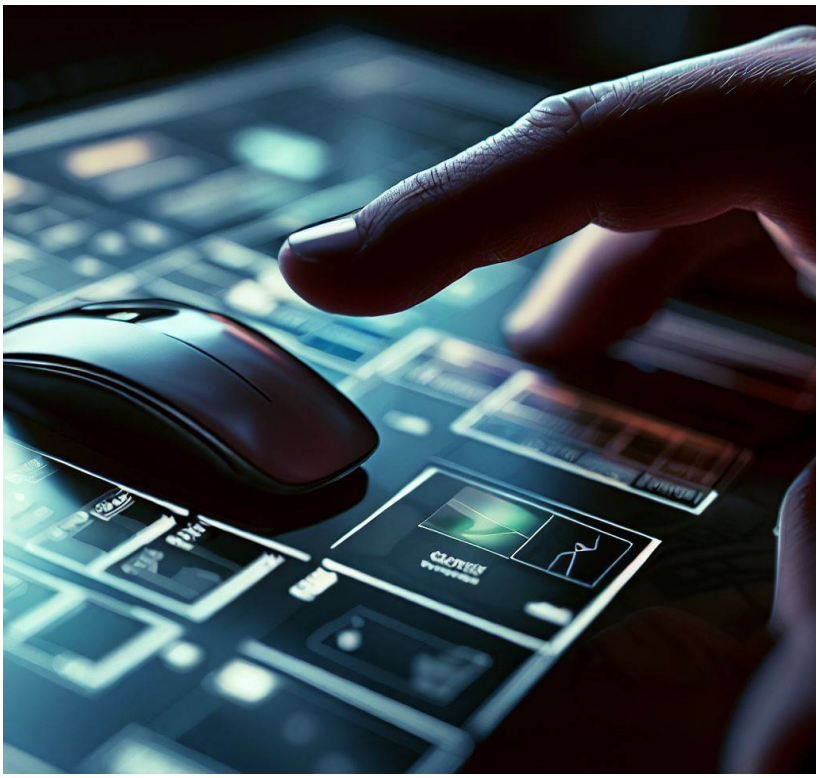
1.29.2 1.9.1.2 Working with controls

1.9.1 Types of controls Previous 162 Top 41 Next 170

Starting Guide

Working with controls

What can we do with controls?



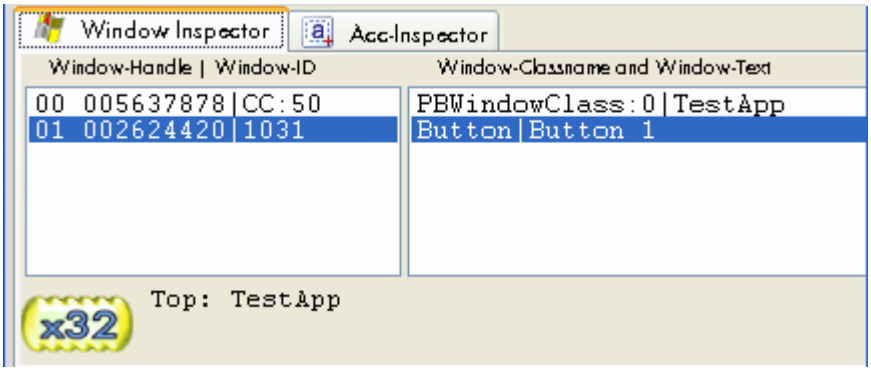
How do we classify controls?

We classify them by their "control class". You can see the class of a control in the "windows inspector" that is built into the editor. For this, just move the mouse over the control and take a look into the Inspector. In our example the class name is "button".

Move the mouse over this control.

Then you will see this in the inspector, and know "its a button control". BUt i believe you have had that idea "this is a button" before.

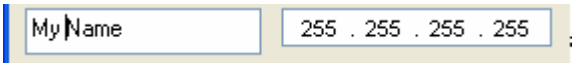
© 2023 Theo Gottwald * https://www.smart-package.com



What do we have to do with controls?

We need to work with controls in several ways. Some controls like buttons are simply clicked with the mouse.

Other controls like the text-box below, need to be read. Or we may need to type text into them.



Here is an example, that reads a text out of a control. The control is a "SysIPAddress-Control" in our TestApp, which you can find in the Sample Application folder.

```
STW.ct|PBWindowClass:0|TestApp
SCW.nc|1|SysIPAddress32
CFW.$$TXT|u
```

What can we do with controls?

An Automation System like this one needs to work with controls in several ways. Note that all things described here can be done with mostly any control, not only with the button control, that we take as example here. Now, what can we do with it?

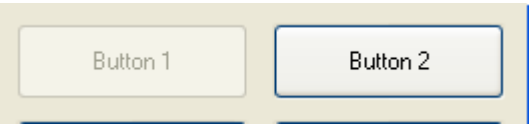
1. Controls can be **enabled or disabled** (use SWS3889.).
2. Controls can be made **visible or invisible** (use SWS3889.).
3. Controls can be **closed** (means they are destroyed, there is no way to get them back unless you restart the application)
4. Controls can be re-sized (change the size of a control)
5. Controls can be repositioned (move them around)
6. The Controls Text can be changed.
7. Then there are specific properties of controls that can be influenced. For example, a List box can change the actual selected item. Or a scroll bar can be moved up or down.

© 2023 Theo Gottwald * https://www.smart-package.com

You can try these with our TestApp anytime without danger. All changes you do to the Application will be lost when you close the Application. Nothing of these is permanent in any way. Here are some examples:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Button|Button 1
' This will disable the button
SWS.d
```

will disable the button 1, making him look like this:



Now its important to know that the system will not detect disabled buttons in its standard settings. To reverse this last operation, we have to change something:

```
STW.ct|PBWindowClass:0|TestApp
' We have added the "a" prototype
SCW.ncta|1|Button|Button 1
' This will enable the button again
SWS.e
```

Will bring the button back to life. SWS. gives you the power to make all sorts of control invisible. Or disabled. Or get them back. Here are some of the SWS.-Options. Note that you can use multiple of them in one line.

```
STW.ct|PBWindowClass:0|TestApp
' We have added the "a" prototype
SCW.ncta|1|Button|Button 1
' This will enable the button and make it invisible
SWS.ei


' These are some prototypes from the SWS.-command.
'
' a - activates the window after setting the state.
' e - enabled
' d - disabled
' v - visible
' i - invisble
' m - minimzed
' n - normal (not minimized)
' z - zoomed (maximized)
```

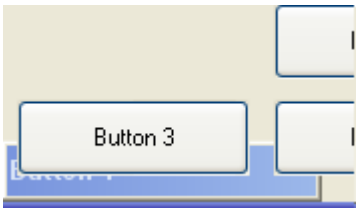
Can I maximize and minimize a control?

© 2023 Theo Gottwald * https://www.smart-package.com

It does not make much sense, but **Yes - you can!**

```
STW.ct|PBWindowClass:0|TestApp
SCW.ncta|1|Button|Button 1
' This will minimize the button
SWS.m
```

For example, a minimized Button looks like this:



If you double click it with the mouse he will get normal again. Be careful with **maximizing** of controls, as there is no easy way back. You may have to completely "close" the control like this:

```
STW.ct|PBWindowClass:0|TestApp
SCW.ncta|1|Button|Button 1
' This will completely remove the button
CLW.
```

to get rid of it.

And how about the text in controls, can we read it, and change it?

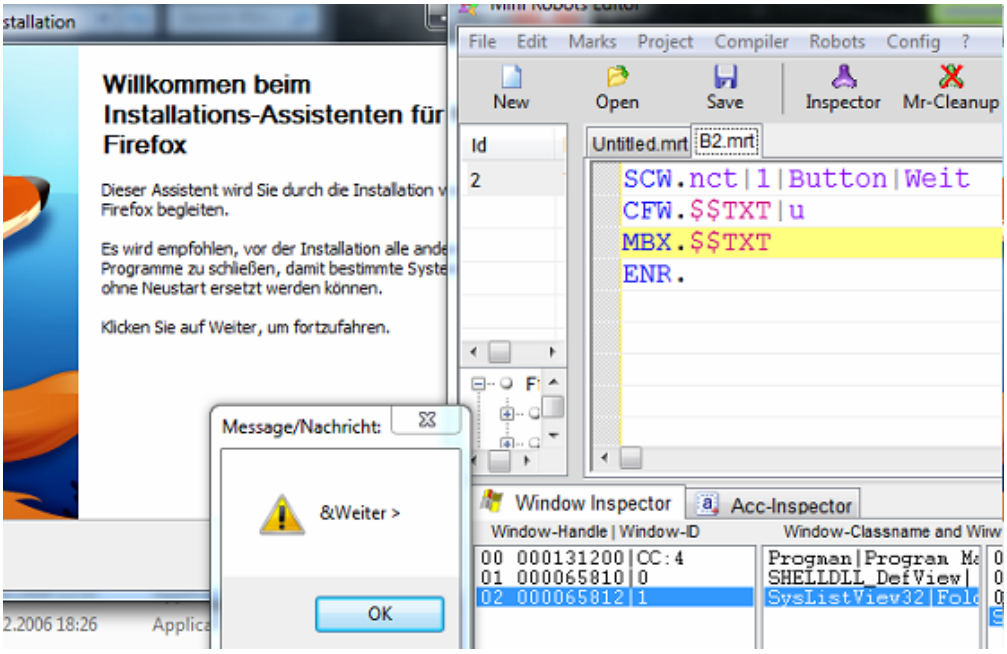
We are not really going to be complete with our list here. We will just look at the most important things, that we can do with controls. And one of these is, that you can read text from controls.
And you can change this text. Not from all controls, but you can try!
Let's try this with a button.

This time we'll not use our TestApp, but a real setup-program. This one:

© 2023 Theo Gottwald * https://www.smart-package.com

First we are going to read the text on that button:

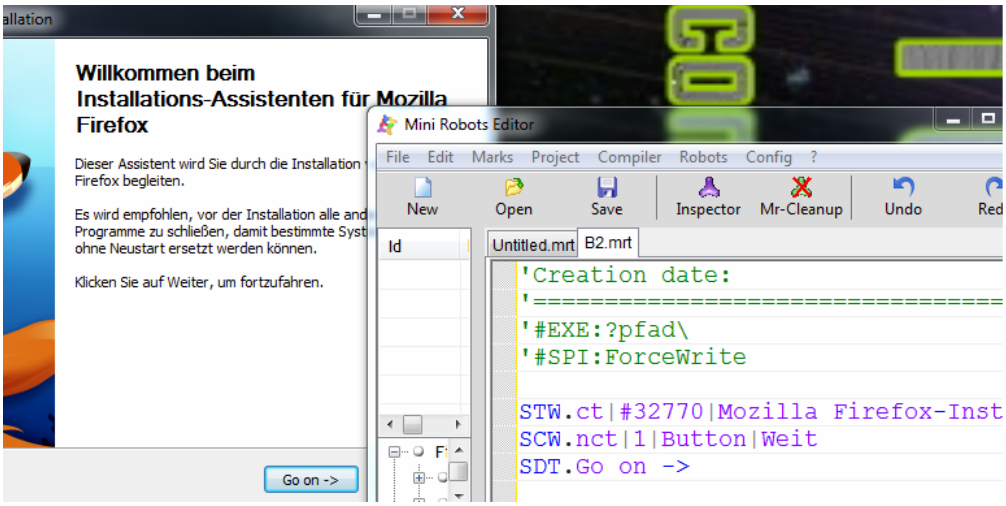
```
STW.ct|#32770|Mozilla Firefox-Installation
SCW.nct|1|Button|Weit
CFW.$$TXT|u
MBX.$$TXT
```



Which works as expected, see above picture.

Next we are going to change the text in these buttons.

© 2023 Theo Gottwald * https://www.smart-package.com



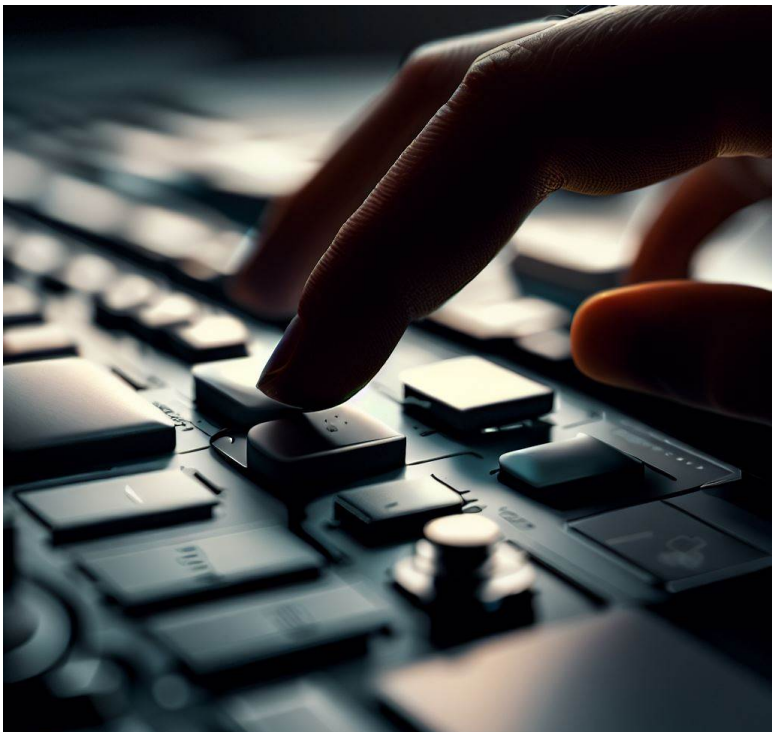
Now one final thing. Let us disable the "Cancel" button, so that nobody can press it.

This is our script:

```
STW.ct|#32770|Mozilla Firefox-Installation
' We use a pattern so that it will work in german and english
Setups
SCW.nct|1|Button|Abbrechen&OR:Cancel
SWS.d
ENR.
```

Which of course also worked as expected.

© 2023 Theo Gottwald * https://www.smart-package.com

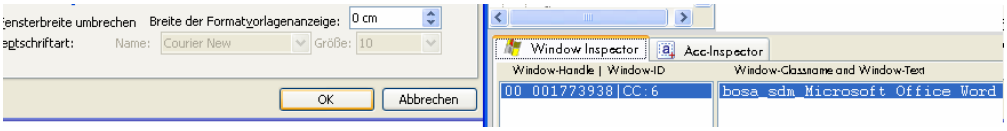


Are There Controls That Don't Support All These Features?

Indeed, there are. Certain dialog boxes contain controls that require a different approach because they aren't traditional windows controls. These include:

1. Dialogs in Office programs
2. Browser dialogs
3. Dialogs generated by Java

When working with these types of controls, the window inspector may not display the expected "button" class. For instance, consider this dialog from Word.



© 2023 Theo Gottwald * https://www.smart-package.com

Navigating Web-Based Controls

The SPR toolkit is adept at handling web-based controls. For instance, to pinpoint a button, you would employ the SAO, AGR, or NAV commands.

The effectiveness of automation scripts can vary depending on the browser used for dialogues. While Internet Explorer is no longer available, Firefox is recommended and highly compatible with our automation tools. Other browsers might pose challenges when trying to locate elements using our primary tools.

Nevertheless, we have a robust solution: a system of graphical recognition commands. These commands can be utilized in less compatible browsers and even in Java-based applications, which form a third category of applications.

It's crucial to understand that Java-based applications often resist scripting using traditional windows-control-commands. Even browser-control-commands, which typically work well in office applications, might not be effective here.

However, there's no cause for concern. Our fallback strategy always involves the use of graphical recognition commands. These commands are highly adaptable and can be employed in nearly any environment.

© 2023 Theo Gottwald * https://www.smart-package.com

1.29.3 1.9.1.3 Button Control

1.9.1 Types of controls Previous 170 Top 41 Next 184

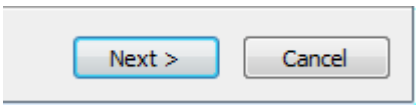
Starting Guide

Button Control

The simplest Control: Button Control

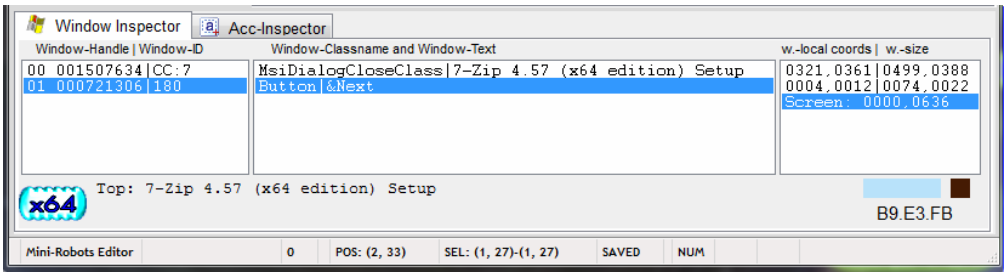
Buttons are mostly not round but somehow cornered.

Understanding the Button Control



A button control is a fundamental element in a GUI. It typically carries the window class "button" and is most commonly interacted with through a click action.
When inspected using the windows-inspector tool, a button control with the window class "button" will be displayed as follows:

© 2023 Theo Gottwald * https://www.smart-package.com



It's important to note that window classes can be named at the discretion of the application's programmer.
Therefore, the window class of a button may not always adhere to the standard "button" classification.

To interact with a button, any of the Mouse Click commands can be utilized:

- MLE. - Mouse-Left-Click Event 2492
- MLC - Mouse-Left-Click (Invisible) 2448
- MLM - Mouse Left click Move (Invinsible) 2457
- BSC. - Button Special Click 2435

or use the combination of

```
STW.ct|#32770|Mozilla Firefox-Installation
AGR.43|cancel&OR:Abbrech
MOS.
```

which will also click the button.

For a detailed comparison and understanding of the variations between these mouse click commands, please refer to the following section:
1.C Direct Engine and Compatible Engine 231

© 2023 Theo Gottwald * https://www.smart-package.com

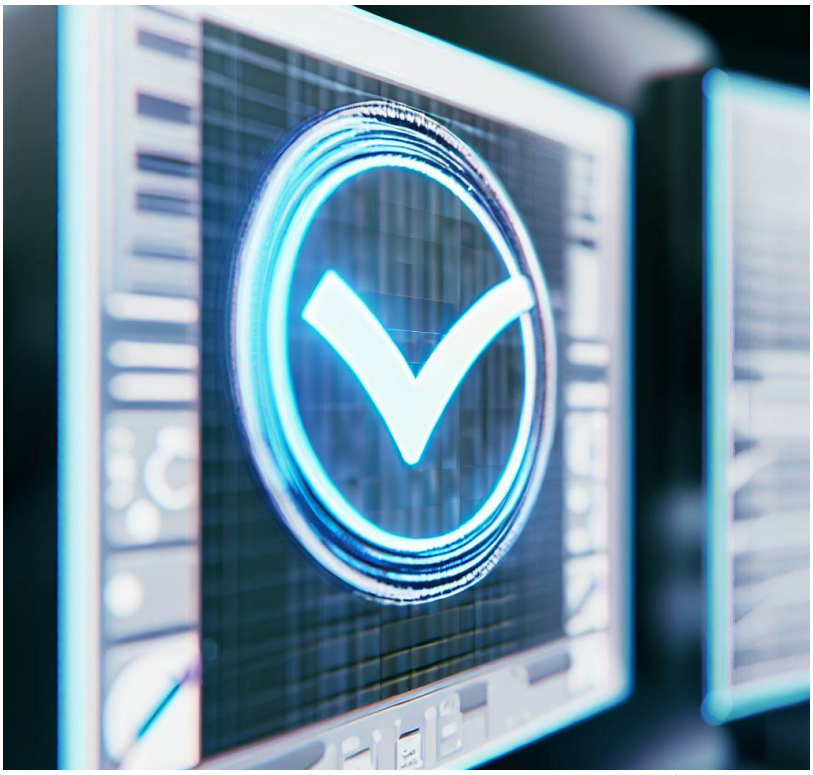
1.29.4 1.9.1.4 Checkmark Control

1.9.1 Types of controls Previous 182 Top 41 Next 191

Starting Guide

Checkmark Control

Selecting Options.



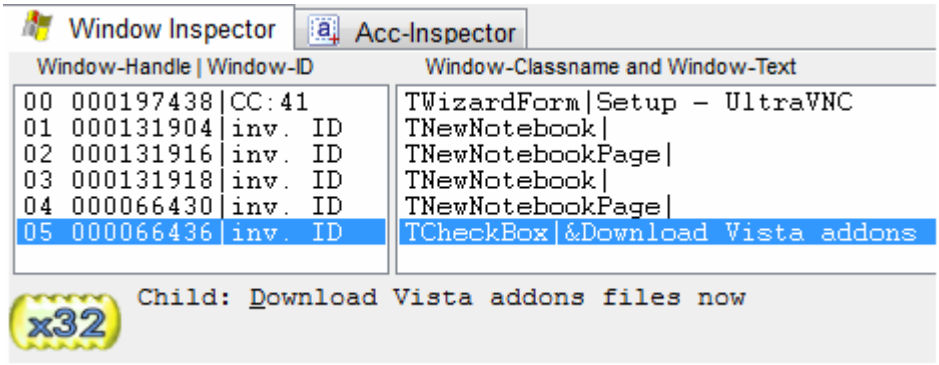
The checkmark control is often used to have the user to confirm something.

Understanding the Checkmark Control

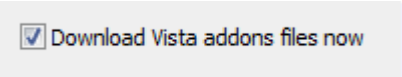
Firstly, let's take a look at a "Checkmark Control". Here's an example:

The windows-inspector for this control might appear as follows:

© 2023 Theo Gottwald * https://www.smart-package.com



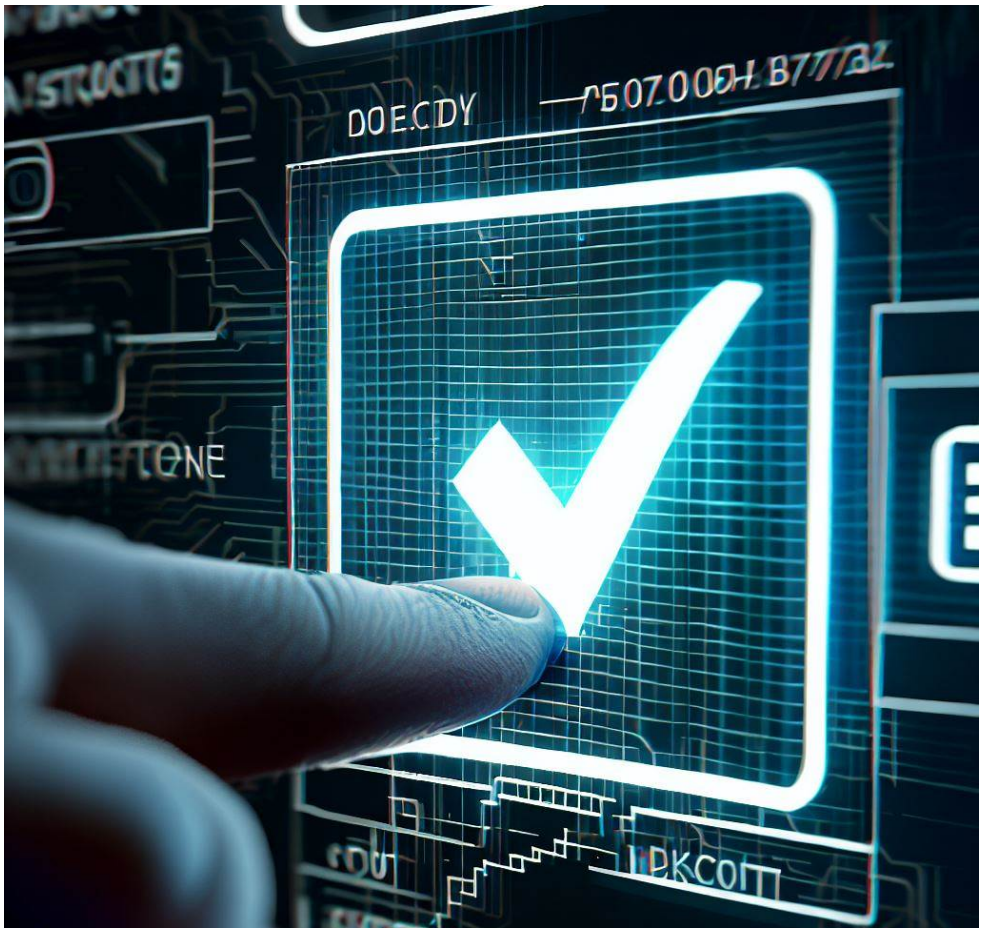
The presence of "check" in the class-name "TCheckbox" is a strong indicator that we're dealing with a genuine checkmark.
Typically, checkmarks have two states: checked and unchecked.



It's important to note that Checkmark Controls differ from Radio Button Controls in that they operate independently of each other. Checking one will not alter the state of another checkmark within the same dialog.
To uncheck the checkmark above, you would use the following script:

```
' Get the application Setup-window
STW.ct|TWizardForm|Setup - UltraVNC

' Locate the checkmark-child window
SCW.ct|TCheckBox|Download Vista

' Set the checkmark to unchecked
BTC.0

' End the script.
ENR.
```

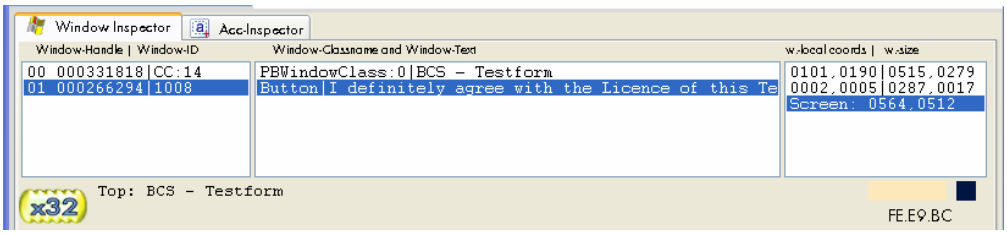

© 2023 Theo Gottwald * https://www.smart-package.com



Interacting with the Checkmark Control

Ideally, with the SPR, we don't merely click on a checkmark with the mouse.
Instead, we use the dedicated BTC command, accompanied by a parameter (0 or 1) that indicates the desired state of the checkmark.
This approach offers an advantage: it makes us independent of the checkmark's previous state.
Regardless of whether the checkmark was checked or unchecked, the BTC command will attempt to achieve the state we want.

The windows inspector displays a checkmark as follows:



Window Class Variations for Checkmark Controls

The window class for a checkmark might be labeled as "button", or it could contain "check" in the name. However, this isn't always the case. The window class could bear different names as well. Given that window classes can be named at the programmer's discretion, the window class for a checkmark control can vary.

© 2023 Theo Gottwald * https://www.smart-package.com

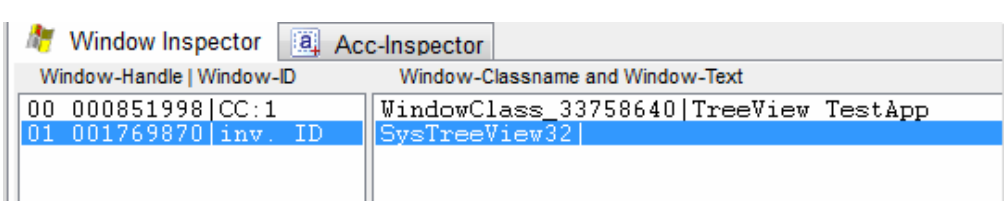
Normally you can change the value of an checkmark using the BTC.-command.

```
' BTC.-Sample
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|I definitely agree with the Licence of this
Testform.
BTC.1
```

There may be checkmark controls out there, that do not react on the BTC. - Button-Check-command as we would expect it.
In that case use the normal mouse clicks to change their state.

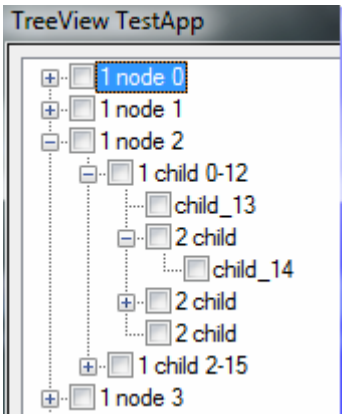
Understanding Treeview Controls with Built-In Checkmarks

Consider the treeview control shown below. It appears to have built-in checkmarks. The windows-inspector identifies it as "systreeview32". However, it's important to note that these are not actual checkmarks, and as such, the BTC command cannot be used with them!

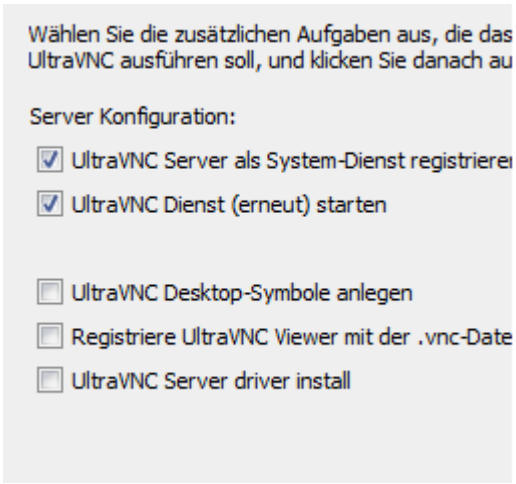


As the window-class shows, this is a treeview. And we need to use the treeview commands for it.

© 2023 Theo Gottwald * https://www.smart-package.com



This is not a checkmark control. It is a "Treeview-Checkmark". The windows-inspector will show it as a treeview "`systreeview32`".
In this case you need to use the `TVI.` 668 - or the TVF. - Tree View Find 665 - command with its options to check and uncheck the boxes.
There are other checkmarks out there which are also not true checkmarks. For example this one:



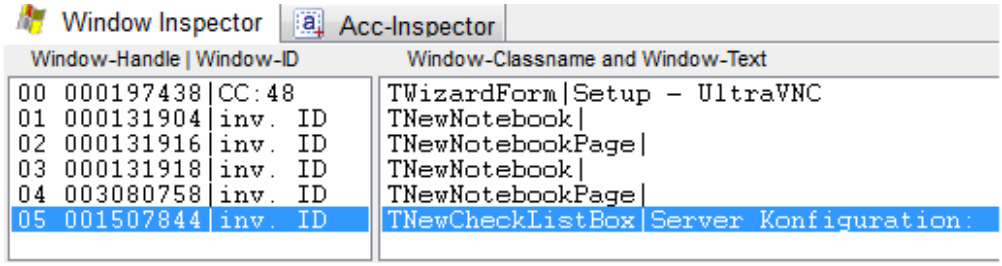
Understanding Listbox Controls with Checkmarks

© 2023 Theo Gottwald * https://www.smart-package.com

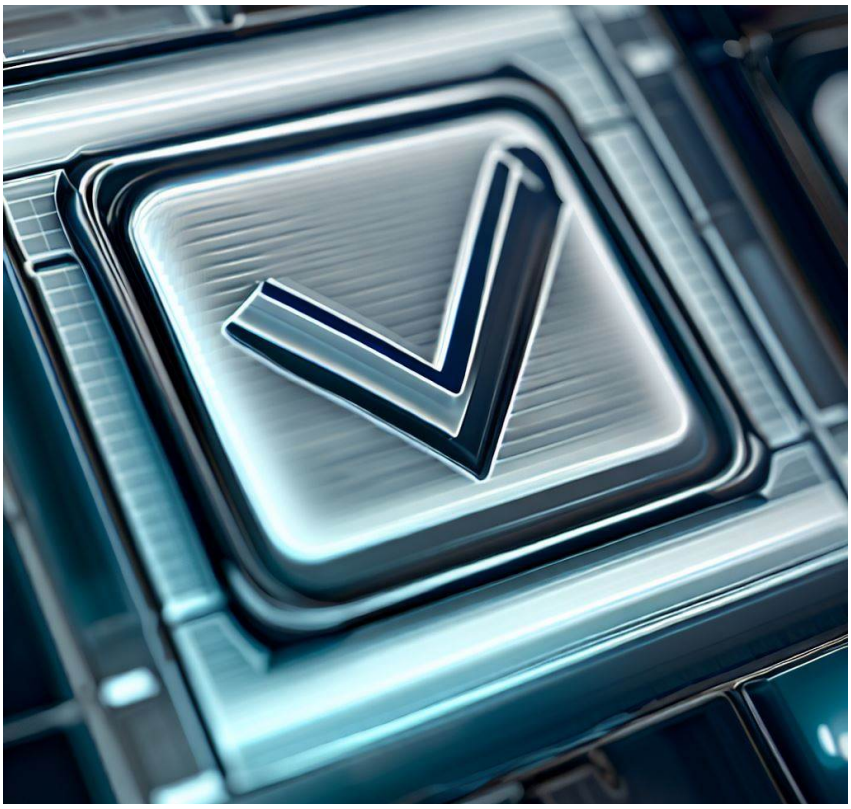
The elements you see are not actual checkmark controls. Instead, they are part of a Listbox that includes checkmarks. To check or uncheck these Listbox Checkmarks, you'll need to use the LBE command along with its options.
The windows-inspector, as shown below, identifies it as a Listbox, specifically "TNewCheckListBox".

To check the first checkmark in this case, you would use the following code:

```
STW.HBJwct|45|TWizardForm|Setup - UltraVNC
SCW.wctd|45|TNewCheckListBox|Server Konfiguration:|5
' This will check the Listbox-Checkmark
LBE.UltraVNC Server als System-Dienst &registrieren||1
```



In some cases, when the Listbox-commands do not work, you can often still locate the Listbox using the SAO 625. command. And then just click it with the mouse using MLI - Mouse Left Click qulckly (Invinsible) 2454, or use the MOS. - Mouseless Object Select 2509-command.



See also:

© 2023 Theo Gottwald * https://www.smart-package.com

These are the most used commands for checkmarks:

- BTC. - Button-Check 635
- GBC. - Get-Button-Check 640
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MLM - Mouse Left click Move (Invinsible) 2457
- SAO. - Search Accessible Object 625
- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- TVI. - Tree View Invisible 668
- LBE. - Listbox-Entry 649

© 2023 Theo Gottwald * https://www.smart-package.com

1.29.5 1.9.1.5 Radio Button Control

1.9.1 Types of controls Previous 184 Top 41 Next 194

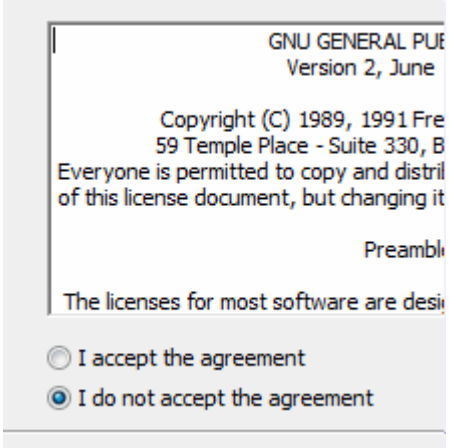
Starting Guide

Radio Button Control

What controls do we have to do with?

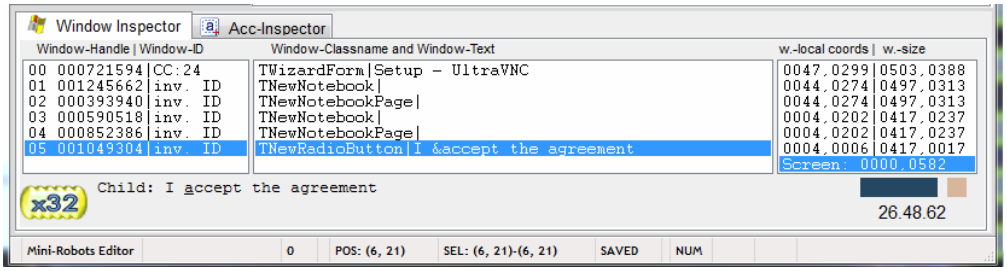
Exploring the Radio Button Control

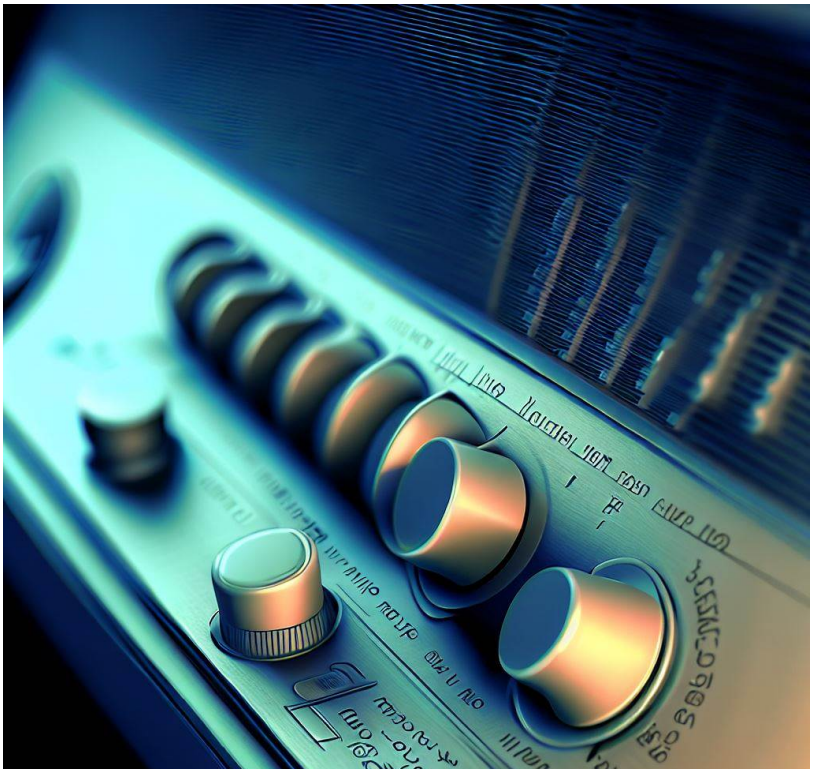
The control we're focusing on now is the Radio Button Control, also known as an "Option Control".
This is because it allows you to select one option from several available choices.
Let's take a look at a typical use case for this control.
For instance, you might encounter a Radio Button Control when presented with an agreement where you can choose to accept or decline.



© 2023 Theo Gottwald * https://www.smart-package.com

In this case the window class of the Option control is "TNewRadioButton". As window classes can be named freely by the programmer of an application, the window class can differ from this.





Interacting with the Radio Button Control

Here's a script that checks the Option Control:

```
STW.ct|TWizardForm|Setup - UltraVNC
SCW.ct|TNewRadioButton|I &accept the agreement
BTC.1
```

Simply changing the number to

```
BTC.0
```

would have the opposite effect. While some Radio Button Controls allow you to uncheck the buttons using the `BTC.0` command, others might not.
In such cases, you're limited to simulating a standard mouse click to select one of the radio buttons.

© 2023 Theo Gottwald * https://www.smart-package.com

You might wonder why it's called a "Radio Button Control". The term "Radio Button" harks back to old radios where pressing one button would automatically disengage any other pressed button. The same principle applies here. Once you click on any of these buttons, the others will switch off.

See also

Here is a list of commands that are often used with Radio Button controls:

- BTC. - Button-Check 635
- GBC. - Get-Button-Check 640
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MLM - Mouse Left click Move (Invinsible) 2457
- SAO. - Search Accessible Object 625
- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509

© 2023 Theo Gottwald * https://www.smart-package.com

1.29.6 1.9.1.6 Edit-Textbox Control

1.9.1 Types of controls Previous 191 Top 41 Next 198

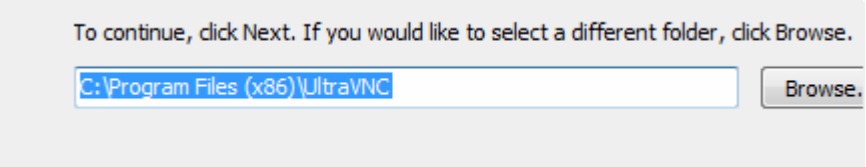
Starting Guide

Edit- or Textbox Control

What controls do we have to do with?

Interacting with the Edit or Textbox Control

The control we're focusing on now is the Textbox Control, also known as an "Edit Control".



There are two primary methods for inputting text into the Edit Control:
• Simulating Keyboard Typing (compatible engine)
• Directly Setting the Text of the Control (direct engine)

Typically, we favor the direct engine 231 for its numerous advantages in automated installations.
It's faster, more reliable, and functions even if the user isn't logged into the machine. The commands for the direct engine setting a textbox are:

- SDT. - SenD-window-Text 2278 (direct engine)

© 2023 Theo Gottwald * https://www.smart-package.com

- RST. - Replace Selected Text 2264 (direct engine)

Here's a script that sets a textbox using the direct engine:

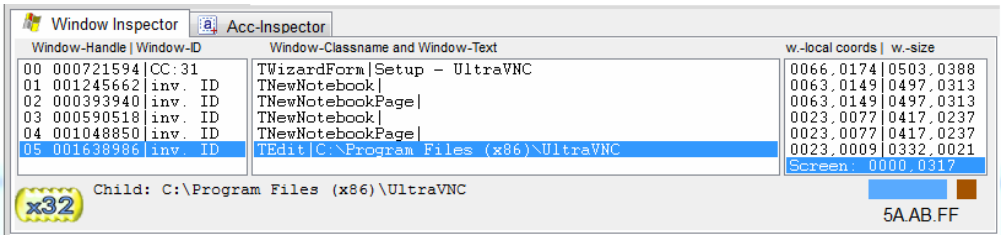
```
VAR.$$S00=?progs\UltraVNC
STW.wct|45|TWizardForm|Setup - UltraVNC
SCW.wcd|45|TEdit|5
RST.$$S00|0,-1
```

In some cases, depending on the target control, the direct engine might be ignored by the control. If this occurs, you'll need to use the compatible engine instead. The compatible engine simulates key presses like a human user and therefore can't be easily ignored by an application. For example, if you want to type text into a web browser, you'll need to use the compatible engine.

The most used compatible engine commands for setting a textbox are:

- SKP. Send-Key-Press 2289 (compatible engine)
- SKT. - Send-Key-Text 2293 (compatible engine)

Note that there are much more then these two.
Most Edit Controls have "edit" or "text" in their class name. However, as window classes can be named at the programmer's discretion, the window class for an Edit Control can vary.
Here's how a windows inspector would display a Edit-Control:



Exploring the Diversity of Edit Controls

There's a wide variety of Edit Controls beyond the two we've discussed. Most Edit Controls have "edit" or "text" in their class name. However, as window classes can be named at the programmer's discretion, the window class for an Edit Control can vary. Here's how a windows inspector would display an Edit Control:

It's important to note that text-boxes in a browser window, like the one shown below, are not traditional windows controls. They aren't genuine text-boxes and can only be accessed using compatible engine commands. This also applies to buttons and other web-based controls.

In such cases, you should use the Acc-Inspector instead. Utilize the SAO 625., MOS 2509., and compatible engine commands. Here's a script to get you started with web-based controls:

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Script to type into a WEB-based text-box using the
' compatible engine

' Check if the Internet-Explorer is already running
STW.wXgct|0|over|IEFrame|Theo Gottwald&ANDTHEN:Internet
Explorer

' If its not running start it with the right starting page
EXO.iexplore|www.fa2.de

' If it was just started, locate the application window
STW.ct|IEFrame|Theo Gottwald&ANDTHEN:- Windows Internet
Explorer

' This label is where we land, if the window is already o
:over

' We set the position of the IE-Window to the upper, left
corner (0,0)
' And set the size of the window to 827,700
SWP.0,0|827,700

' We locate the web-control that contains the web-page
SCW.ct|Internet Explorer_Server|{&NOTEXT:}

' we remember the handle of the web-control in a variable
HTV.$$MYW

' Locate the web-based text-box
SAO.rn|42|{&NOTEXT}
' Klick inside
MLI.
' Type the text
SKT.info@it-berater.org
' Wait because typing is asynchrone
PAU.0.5

' Locate the web-control that contains the web-page
' we use the handle that we have remembered
SCW.h|$$MYW

' Locate the web-based button
SAO.rn|43|Google Search

' Click the Search button
MLE.
ENR.
```

See also:

- SKP. Send-Key-Press 2289 (compatible engine)

© 2023 Theo Gottwald * https://www.smart-package.com

- SKT. - Send-Key-Text 2293 (compatible engine)
- SDT. - SenD-window-Text 2278 (direct engine)
- RST. - Replace Selected Text 2264 (direct engine)
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

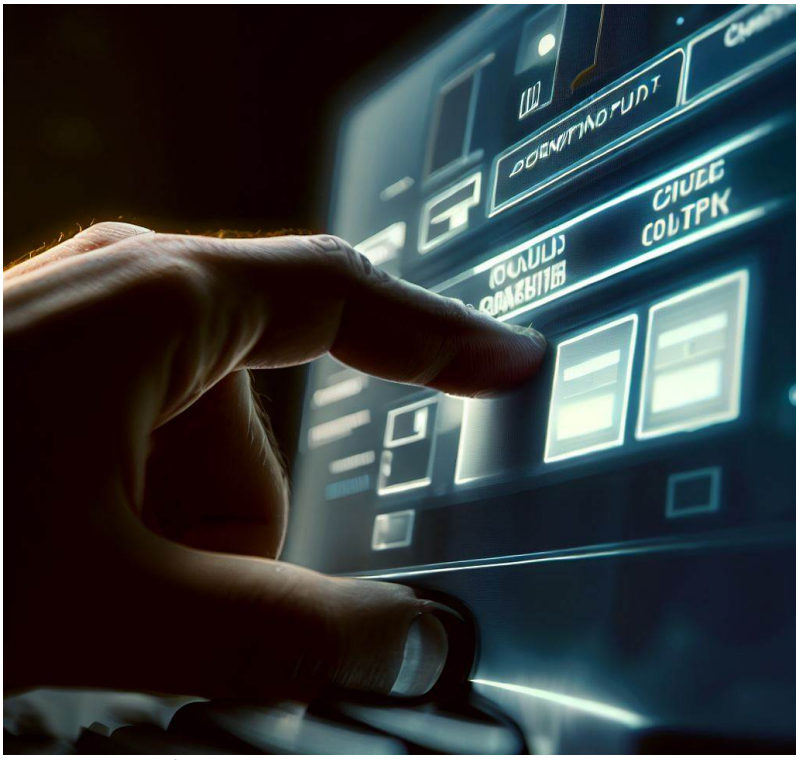
1.29.7 1.9.1.7 Dropdown- Combobox Control

1.9.1 Types of controls Previous 194 Top 41 Next 202

Starting Guide

Dropdown and Combobox Control

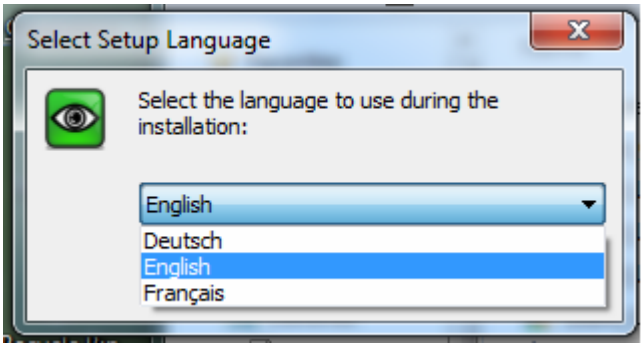
What controls do we have to do with?



A Dropdown Control is like a Listbox combined with a menu.

Understanding Dropdown and Combobox Controls

Below is an example of a "Dropdown Control".



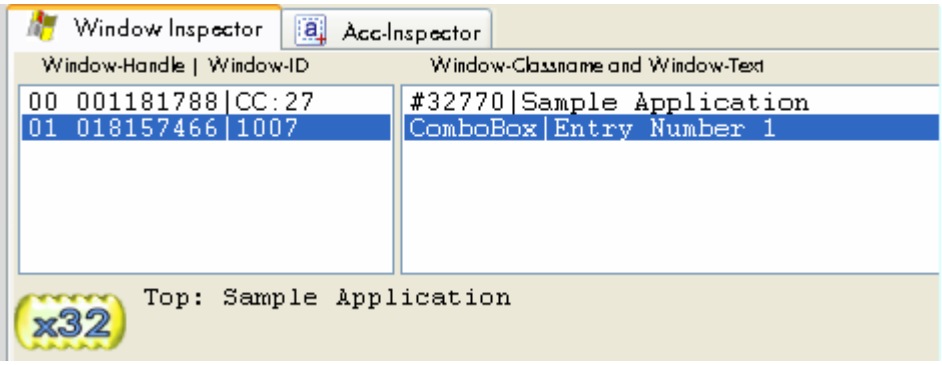
There are two distinct types of dropdown controls:

© 2023 Theo Gottwald * https://www.smart-package.com

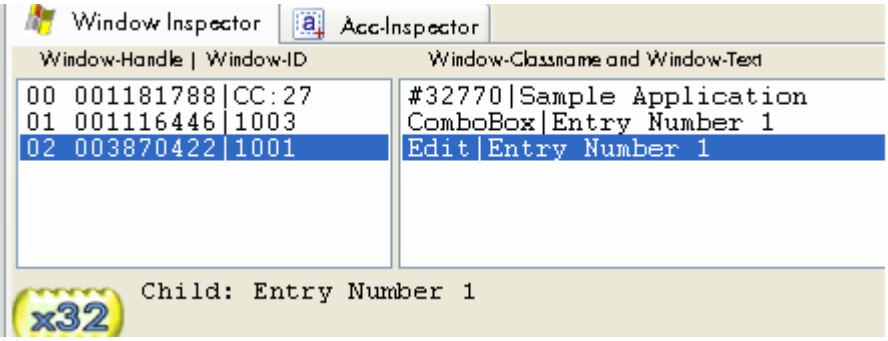
1. The standard Dropdown Control, which only allows selection.
2. The Combobox Control, which combines a dropdown and an edit-box control.

For convenience, we often refer to both as a Combobox Control. Regardless, both types of controls are addressed. Sometimes, the windows-inspector will clearly indicate the type of control you're dealing with.

Here's an example of a Dropdown Control:



In the image below, you'll notice an Edit Control beneath the Combobox Control. This is a true Combobox Control because it also includes an Edit Control, as you can see in the image below:



When you want to input text into the built-in text-box, you can locate either the Combo-Box or the Edit-Control. Normally, both will function as expected. These two scripts will both perform the same action and write some text into the control:

```
' Locate the Combobox-Control
STW.ct|#32770|Sample Application
SCW.nc|1|ComboBox
SCW.nc|1|Edit
SDT.Hallo

' Locate the Edit-Control below the Combobox
STW.ct|#32770|Sample Application
SCW.nc|1|ComboBox
SCW.nc|1|Edit
SDT.Hallo
```

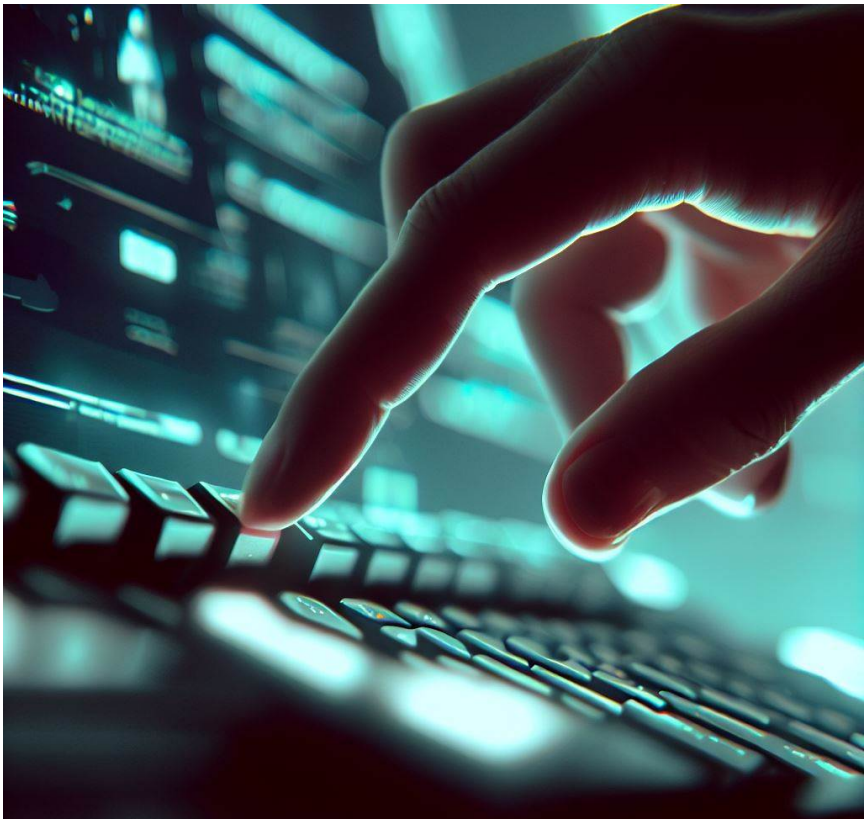
If you want to select an item from the Combo-Box, then you need to locate the Combo-Box.
You have two ways of doing so:

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|#32770|Sample Application
SCW.c|ComboBox
SCI.Entry Number 1
```

or using the `SAO.`-command:

```
STW.ct|#32770|Sample Application
SAO.rn|46|{&NOTEXT}
SCI.Entry Number 1
```



Troubleshooting Dropdown-Controls

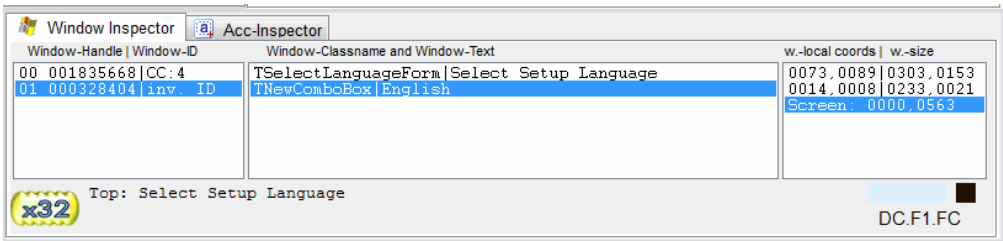
In a few applications, you might encounter a problem where a Dropdown-Control displays the value that you've sent using the SCI. (direct engine command). However, the application does not react to that value, or it reacts as if the value wasn't set. In these cases, the application reads the value once the Dropdown control is opened and closed. This is done using the CDD. - command.
A single `CDD.`-command without any parameters will open and close the currently located Dropdown control.

Here's a complete code example:

```
' Here is an complete code example.
STW.ct|TSelectLanguageForm|Select Setup Language
SCW.nc|1|TNewComboBox
SCI.english
' the following line opens and closes the dropdown control
```


© 2023 Theo Gottwald * https://www.smart-package.com

CDD.



As window classes can be named freely by the programmer of an application, the window class of dropdown controls can differ from those in the examples here.

Important Note:

Dropdown Controls in WEB-Browsers are not windows controls and do not react to neither SCI. or CDD. In this case, you need to use the command SAO. and MOS. or MLI. to open, select, and close the WEB-Browser based Control.

See also:

- 1.9. Locating Child-Windows 162
- SCI. - Set-Combo-Item 657
- GCT. - Get-Childs-TopWindows 3622
- 1.9.2.1 WEB-based Textbox 194
- SAO. - Search Accessible Object 625
- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- MLI - Mouse Left Click quickly (Invinsible) 2454

© 2023 Theo Gottwald * https://www.smart-package.com

1.29.8 1.9.1.8 Treeview Control

1.9.1 Types of controls Previous 198 Top 41 Next 208

Starting Guide

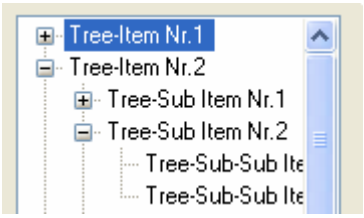
Treeview Control

What controls do we have to do with?

Treeview" as the AI imagines it.

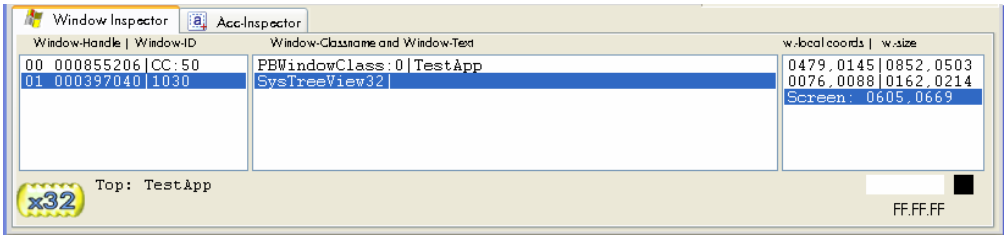
Navigating the Treeview Control

A Treeview Control might look something like this:

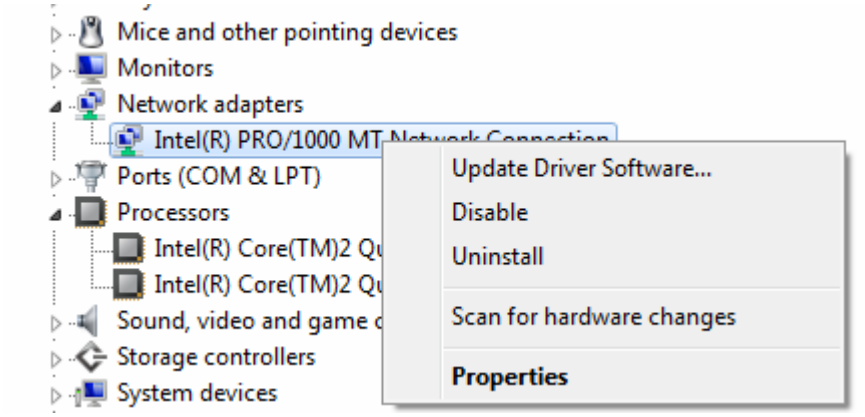


In the Windows Inspector, the Treeview displays a window class of `systreeview32`, as shown below:

© 2023 Theo Gottwald * https://www.smart-package.com



Treeviews have the capability to display icons, checkmarks, and even small menus. Here are a few examples:



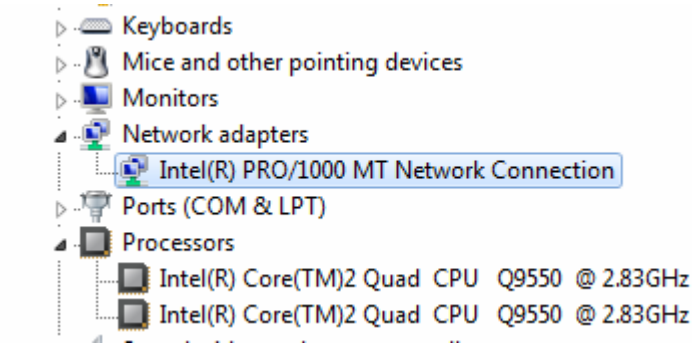
To select an item in a Treeview Control, we have two options. One way is to describe a path, similar to a file-system path, but leading to a Treeview item instead of a file.
This is done using the TVI. - Tree View Invisible 668 command. You can use wildcards like ? and * in the item description. Here's an example:

```
TVI.VM_WIN7_x64\Network adapters\Intel(R) PRO/10?? MT*
```

Alternatively, you can use the TVF. - Tree View Find 665 command to directly locate the item you want. In this case, you can use all 3.2 Standard-Search Pattern 341 to locate the item.
Here's an example:

```
TVF.Intel(R) PRO/1000

' Now here is a complete example, that will open a PopUp Menu
on
' the item we have located using TVI.
STW.wct|60|MMCMainFrame|Device Manager
SCW.wcd|60|SysTreeView32|7
TVI.VM_WIN7_x64\Network adapters\Intel(R) PRO/1000 MT*
' Open the Menu, click on Properties
SMH.Properti
MRC.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Here's another example that would locate the first of the two Intel CPUs in the device manager:

```
TVI.VM_WIN7_x64\Processors\Intel(R) Core(TM)2 Quad  CPU*
```

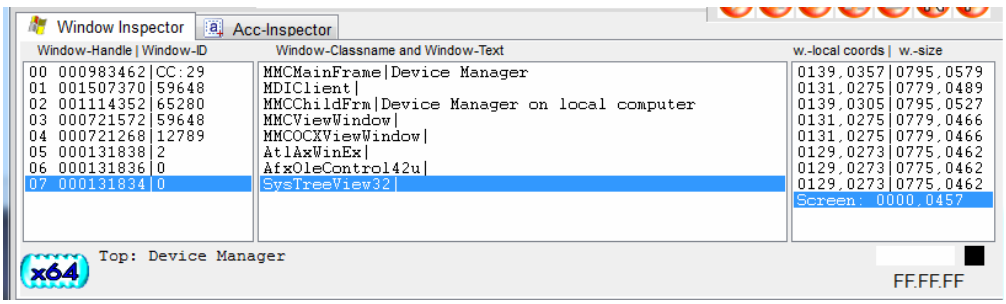
What if we want to get the second of the two items? In that case, you would write:

```
TVI.VM_WIN7_x64\Processors\Intel(R) Core(TM)2 Quad  CPU*|2|
'Don't leave the second | away, its needed!
```

`TVI.` and `TVF.` have a special syntax. If they have just two parameters, the second parameter is the flags.
If they have three parameters, the second parameter is the count while the third parameter is the flags.

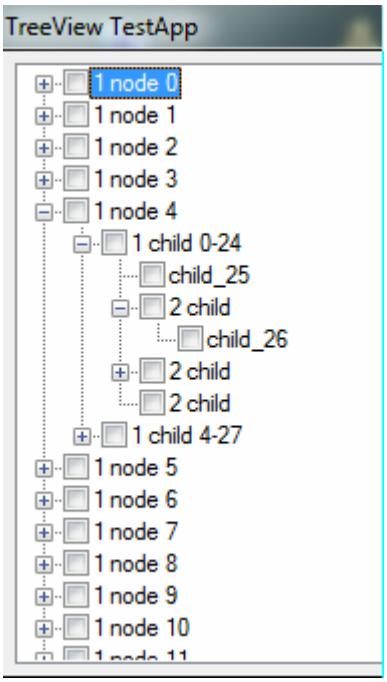
```
' You can do this
TVF.Intel(R)

' or do this using a flag
TVF.Intel(R)|i

' or do this using a flag and a number to choose the second
' fitting element
TVF.Intel(R)|i|2
```

Now, here's what the Windows Inspector shows you when you analyze the Windows 7/x64 Device Manager:



The picture below shows another type of Treeview. This one contains checkmarks. You can check them using the

© 2023 Theo Gottwald * https://www.smart-package.com

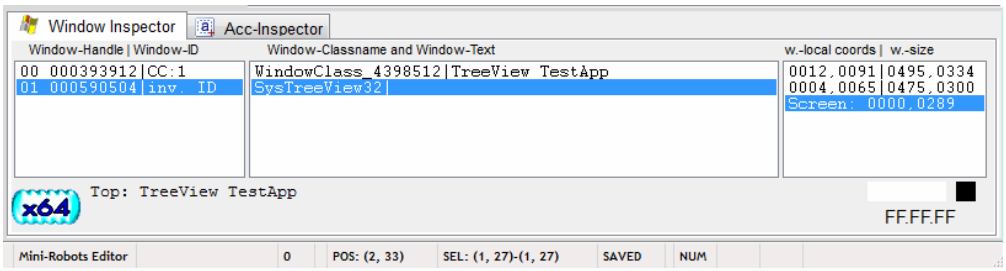



```
TVI. ..|b
```

and uncheck them using the

```
TVI...|d
```

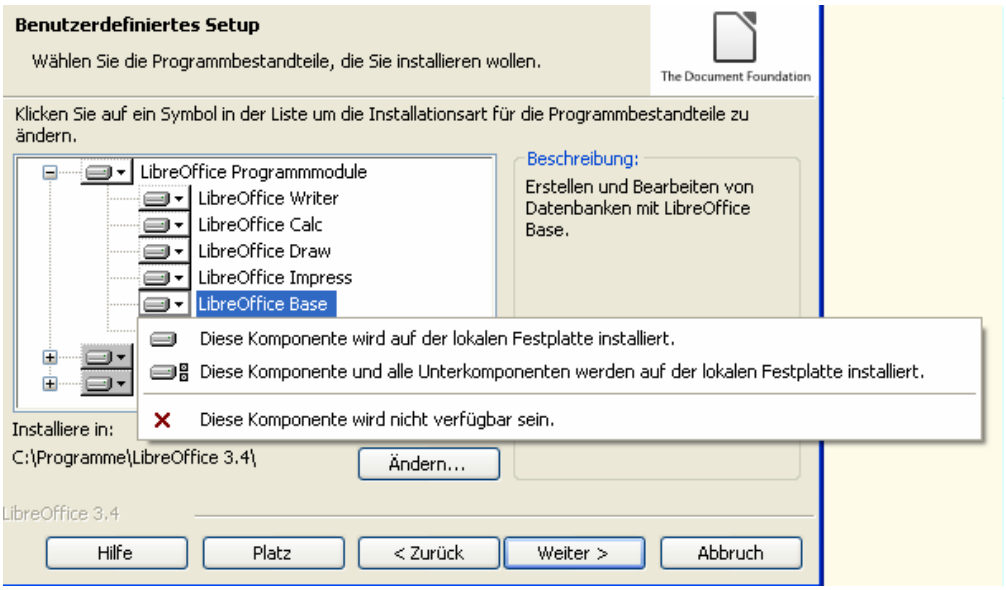
flags. Do not try `BTC.` here, it will not work.Below you can see the windows inspector showing the TestApp_x64 Treeview.



© 2023 Theo Gottwald * https://www.smart-package.com

Treeview" as the AI imagines it.

There's another type of Treeview that you should be aware of. It's an MSI-customizing Treeview:



This is a customizing Treeview Control. We have a separate chapter about it:

1.A Special Issues 215

See also:

- TVI. - Tree View Item 668
- TVF. - Tree View Find 665
- SAO. - Search Accessible Object 625

© 2023 Theo Gottwald * https://www.smart-package.com

- !. NAV - NAVigate in Dialog 606

© 2023 Theo Gottwald * https://www.smart-package.com

1.29.9 1.9.1.9 Page- Tab Control

1.9.1 Types of controls Previous 202 Top 41 Next 210

Starting Guide

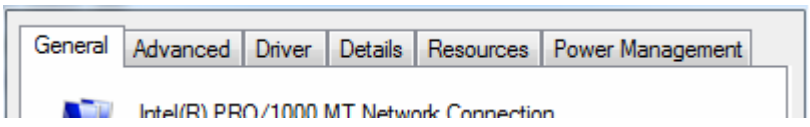
Page Tab Control

What controls do we have to do with?

A Page Tab Control is the digital Representation of a File-Tab.

Exploring the Page Tab Control

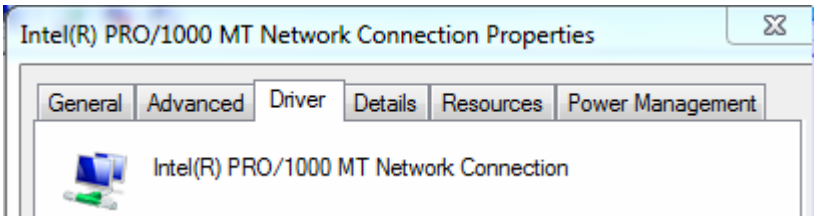
Let's first take a look at a "Page Tab Control":



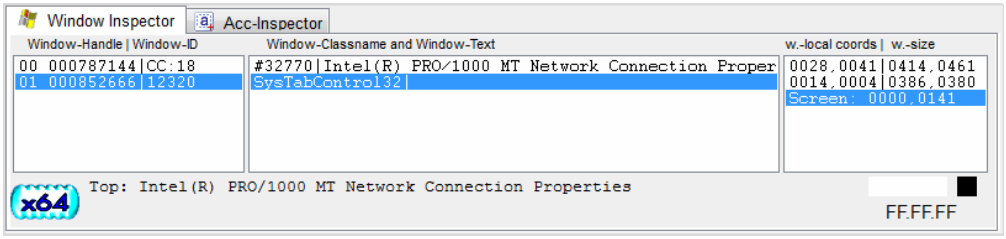
The control we're focusing on now is the Page Tab Control, the digital equivalent of a "File Tab".
It facilitates navigation between different forms.
The typical code to manage a TAB-Control looks like this:

```
STW.ct|#32770|Intel(R) PRO/1000 MT Network Connection Properties
SCW.nct|1|SysTabControl32|{&NOTEXT:}
SAO.rn|37|Driver
MLE.
```


© 2023 Theo Gottwald * https://www.smart-package.com



And here's how the Windows Inspector displays a Tab-Control:



To interact with it, use the SAO. - Search Accessible Object 625 .- and MLE. - Mouse-Left-Click Event 2492 commands.
As window classes can be named freely by the programmer of an application, the window class can differ from this.

© 2023 Theo Gottwald * https://www.smart-package.com

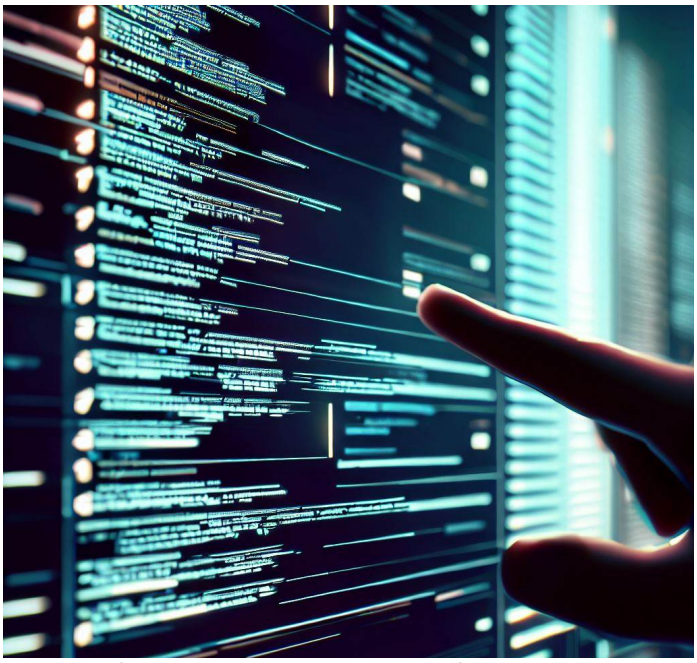
1.29.10 1.9.1.A Listbox Control

1.9.1 Types of controls Previous 208 Top 41 Next 212

Starting Guide

Listbox Control

What controls do we have to do with?



A Listbox Control is the representation of a digital list.

The Listbox Control.

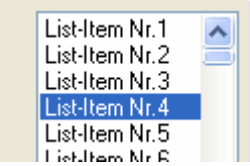
This is a Listbox-Control. A listbox shows you a number of items. You can choose one or many items, by clicking at them with the mouse.

Here is a script to do just that. The LBE.-command will scroll the needed icon and is even able to check a Checkmark of a Listbox-Checkmark.
If LBE. doesn't work, try SAO. SAO. will however not scroll a Listbox, you have to bring the item into visible area yourself, using mouse or keyboard..

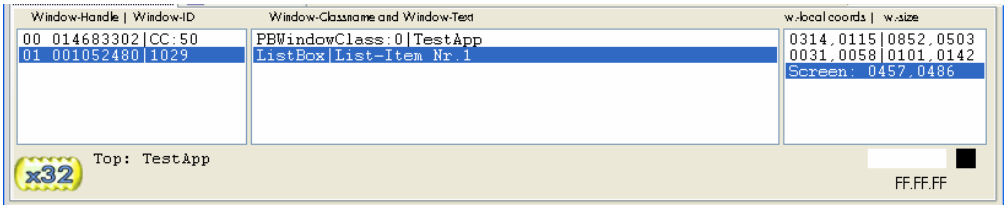
```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|ListBox|List-Item Nr.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SAO.rn|34|List-Item Nr.4
MLE.

' This will check a "Listbox checkmark"
STW.wct|45|TWizardForm|Setup - UltraVNC
SCW.wctd|45|TNewCheckListBox|Server Konfiguration:|5
SFW.
LBE.UltraVNC &Desktop-Symbole anlegen||1
```



As window classes can be named freely by the programmer of an application, the window class can differ from this.



© 2023 Theo Gottwald * https://www.smart-package.com

1.30 1.9.2 WEB-based controls

1.30.1 1.9.2.1 WEB-based Textbox

WEB-based Textbox **Previous** 210 **Top** 41 **Next** 215

Starting Guide

WEB-based Textbox

WEB-Browser based controls

The WEB-Browser based Textbox

To automate WEB-Sites or Intranet Sites, prefer using the Internet Explorer. That is first choice because you can control it best using the built in accessible engine robot commands.
The only alternative at this time is Firefox, who also can be controlled to a good degree.
Actually do not use Google Chrome or Opera, because with these it will not work.

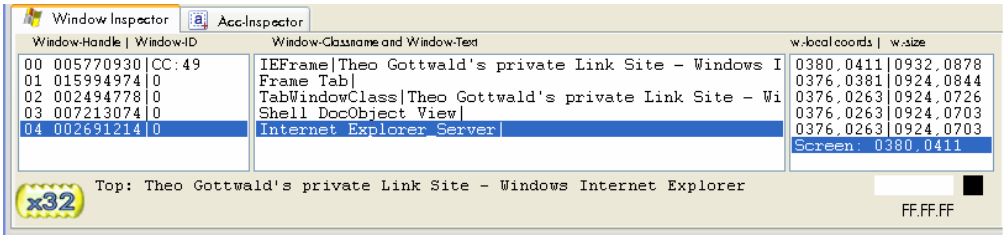
The first thing you need to know is, that the direct engine 231 commands for the keyboard are ignored by web-browser based controls.
Therefore, you will need to use the compatible engine 231 commands instead. The compatible engine simulates key.presses like a human user and therefore can not easily be ignored by an application. For example if you want to type text into a web-browser, you will need to use the compatible engine.

The compatible engine commands for setting a text-box are:

- SKP. Send-Key-Press 2289 (compatible engine)
- SKT. - Send-Key-Text 2293 (compatible engine)

This is **not** the case for mouse-clicks. You can use `MLE.` or `MOS.` (direct engine) to simulate mouse-clicks inside the browser.

Below you see the windows inspector showing an Internet Explorer window.



Note that text-boxes in a browser-window like the one below, are not windows controls. They are not even real text-boxes. Thats why they can only be accessed using compatible engine commands. The same is true for many other web-based controls.

*© 2023 Theo Gottwald * https://www.smart-package.com*

In such cases use the Acc.-Inspector instead. Use SAO625., MOS2509. and the compatible engine commands. Below is a script that wil get you started using web-based controls.

```
' Script to type into a WEB-based text-box using the
' compatible engine

' Check if the Internet-Explorer is already running
STW.wXgct|0|over|IEFrame|Theo Gottwald&ANDTHEN:Internet
Explorer

' If its not running start it with the right starting page
EXO.iexplore|www.fa2.de

' If it was just started, locate the application window
STW.ct|IEFrame|Theo Gottwald&ANDTHEN:- Windows Internet
Explorer

' This label is where we land, if the window is already o
:over

' We set the position of the IE-Window to the upper, left
corner (0,0)
' And set the size of the window to 827,700
SWP.0,0|827,700

' We locate the web-control that contains the web-page
SCW.ct|Internet Explorer_Server|{&NOTEXT:}

' we remember the handle of the web-control in a variable
HTV.$$MYW

' Locate the web-based text-box
SAO.rn|42|{&NOTEXT}
' Klick inside
MLI.
' Type the text
SKT.info@it-berater.org
' Wait because typing is asynchrone
PAU.0.5

' Locate the web-control that contains the web-page
' we use the handle that we have remembered
SCW.h|$$MYW

' Locate the web-based button
SAO.rn|43|Google Search

' Click the Search button
MLE.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- SKP. Send-Key-Press 2289 (compatible engine)
- SKT. - Send-Key-Text 2293 (compatible engine)
- MOS. - Mouseless Object Select 2509
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- SAO. - Search Accessible Object 625
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

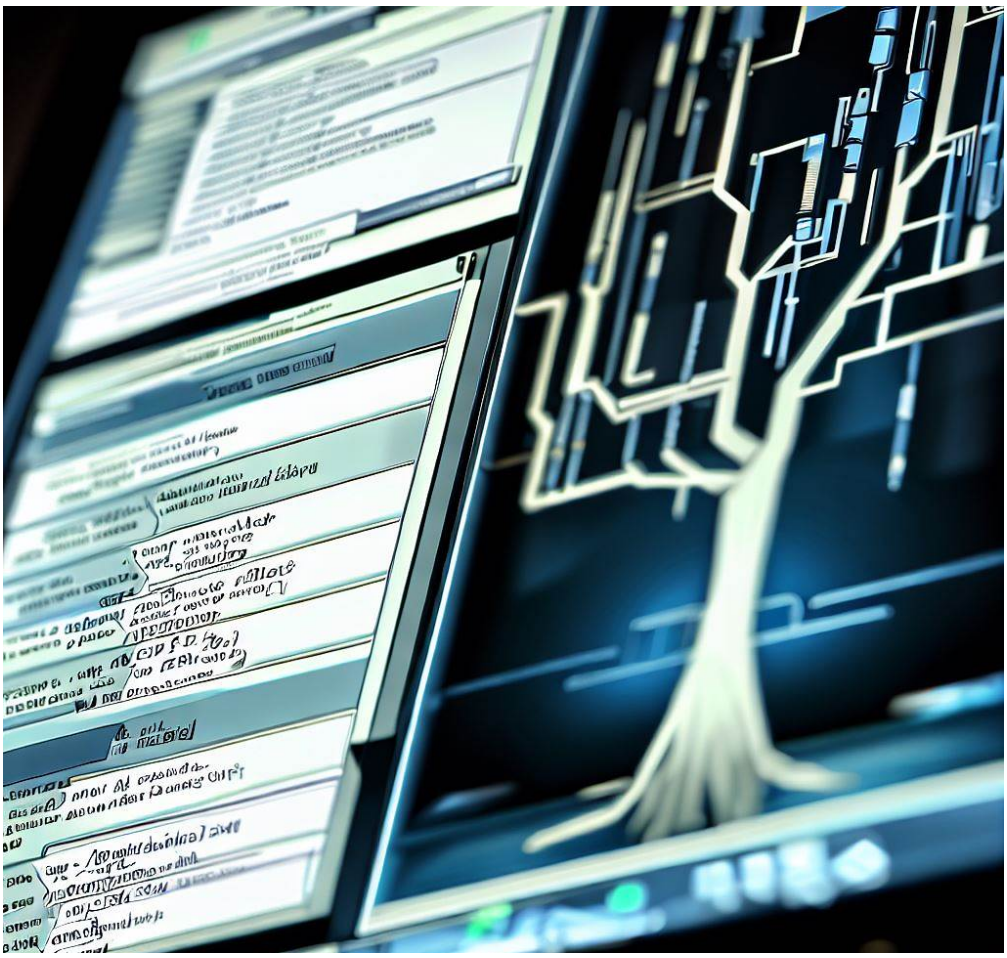
1.31 1.A Special Issues

Special Issues with visual setups Previous 212 Top 41 Next 224

Starting Guide

Special issues with visual Setup Automation

The MSI-Installer Treeview and other Topics

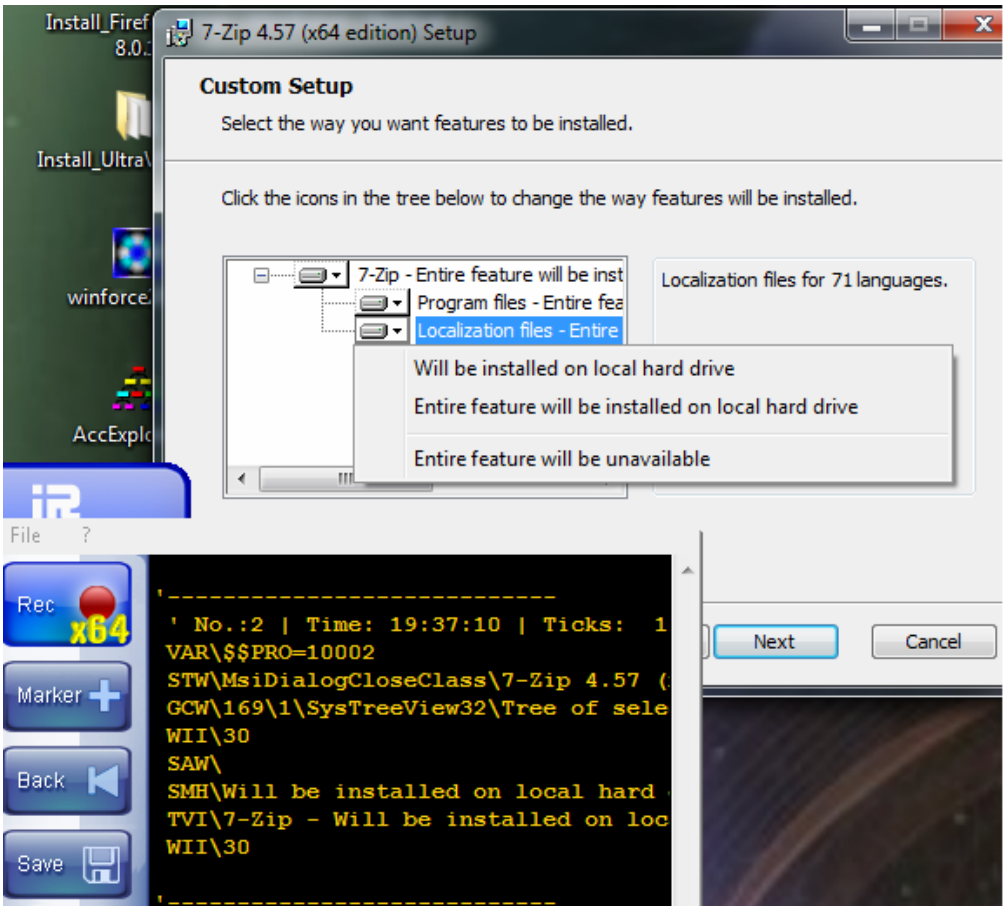


This is how AI thinks an Installer Treeview could look like.

Navigating the MSI Installation Treeview

Sooner or later, you'll encounter the MSI Setup Treeview.
The image below showcases an MSI Treeview with an open menu.

*© 2023 Theo Gottwald * https://www.smart-package.com*



As you can see, InstaRec is the simplest method for generating code for this type of interface. In most cases, this is all you need to create a functional script.
Let's examine the final recording to understand how it works. Later, we'll explore advanced features that can help troubleshoot more complex situations.
Here's an example of the code that InstaRec generates for a menu selection in an MSI Treeview:

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
Setup
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
WII.30
SFW.
SMH.Will be installed on local hard drive
TVI.7-Zip - Will be installed on local hard drive\Localization
files - Entire feature will be unavailable|i
WII.30
```

Let's go together through these lines to understand what each of them is good for.

First Line:

```
VAR.$$PRO=10002
```


© 2023 Theo Gottwald * https://www.smart-package.com

This Line will set the Variable $$PRO to the value of 10002. This is only needed in case of an error. For details see the chapter 1.B Error handling 224.

Second line:

```
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
Setup
```

This line localizes the Application window. For details see 1.8. Selecting the Application (Topwindow) 150

Third line:

```
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
```

This line selects the control (child-window). In our case its the tree-control. For details on selecting a child-window read this: 1.9. Locating Child-Windows 162 .

Fourth line:

```
WII.30
```

This is a "Wait for Input Idle" command. It makes the robot wait until the tree-control can accept our input. For details see WII. - Wait-for-Input-Idle 2909 .

Fifth line:

```
SFW.
```

This is a "Set Foreground window" command. It's maybe not needed here, but as it does not cost a thing, InstaRec just places it here. It will bring the window on top of other windows that may possibly be at the same place.

Line No. six

```
SMH.Will be installed on local hard drive
```

Now it gets interesting. This is the SMH-command (Set Menu Hook). It doesn't really set a hook, the name is more for traditional reasons. However the SMH tells the robot that soon a menu may appear. And it tells the robot what to click in that menu. Like in most cases there is no need to always write the full string, as SMH. supports the 3.2 Standard-Search Pattern 341 .

```
SMH.installed&ANDTHEN:drive
```

would also have been ok in our current case. Now we have told the robot that a menu will come up. The next line will bring the menu up:

Line No. seven

```
TVI.7-Zip - Will be installed on local hard drive\Localization
files - Entire feature will be unavailable|i
```


© 2023 Theo Gottwald * https://www.smart-package.com

This line is the key part of the whole code-block. Therefore we will take a closer look at it. First we have a TVI. - command. This command does NOT support 3.2 Standard-Search Pattern 341
for compatibility reasons with WinRobots compatible Scripts.
Instead it supports the old wildcards the * and the ?. There is a difference however. While you could use these wildcards only at the start of the tree in WinRobots-Script, the more advanced MR-Engine allows you to use them anywhere in the command, even at multiple places.

And in case you prefer to use the 3.2 Standard-Search Pattern 341 you have as an alternative to TVI. the TVF.-command, that supports these more advanced Patterns.

Lets take a look at the TVI.-parameters. First we have the description of the Treeview Item.
Its like a path, even like a file-path. It also uses the \ as a divider of the tree branches. So we have these tree branches:

Root: `7-Zip - Will be installed on local hard drive`
then: `Localization files - Entire feature will be unavailable`

at last we see that there is a

```
|i
```

at the end of the TVI.-line. Taking a look in the help file here: TVI. - Tree View Invisible 668
we can see that this should open the MSI-Menu on the selected tree-item.

If you record MSI-Tree's using InstaRec, the result should always look somehow like this.
It should never contain screen or window coordinates. This can happen, if you move out of the menu window at recording time. If thats the case, you best do the whole thing again.

Now lets take a case where anything was properly recorded, but still doesn't work.

Troubleshooting: What to Do When Things Go Wrong?

Our example above presents a challenging situation for three reasons:
a) The item's position isn't perfect after localization.
b) The treeview is too small.
c) The "|i"-flag doesn't open the menu as expected.

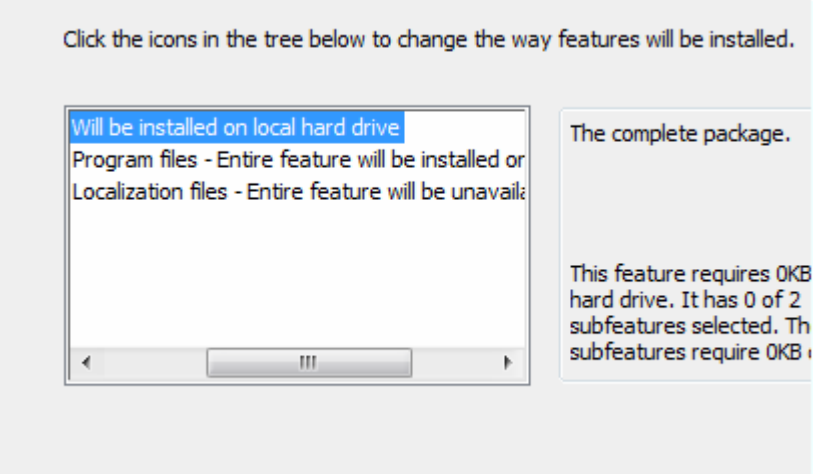
To address these issues, we need to break down the problem and analyze each step. Let's start by checking if the tree-item's localization is functioning correctly.

```
STW.HBJwct|45|MsiDialogCloseClass|7-Zip
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
' I have removed the |i here
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TVI.7-Zip - Will be installed on local hard drive\Localization
files - Entire feature will be unavailable
' The Mouse should move to the tree-item
MMV.
```

This is what we get.



Yes, the Mouse moves into the middle of the tree-item. Thats normally just what we want, and its completely Ok. But now - how do we select the Menu from here?

First we will use the capability of the TVI.-command and shorten the recorded line a bit. This makes it easier for us to work with it.

```
TVI.7-Zip*\Localization files*
```

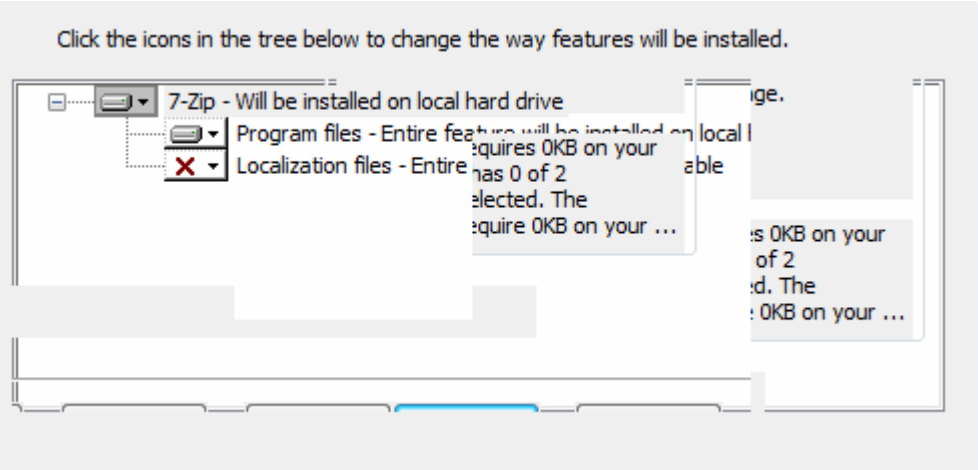
Will do the same as the large line from above. We have used the ability of using the * wildcard to shorten the line. Now back to our problem. We can't shorten the Tree-Item text, but we can resize the tree-control. And this is what we do.

```
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
Setup
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
' resize the tree control
SWP.10,100|466,166|C

SCW.ct|static|The complete
CLW.

SCW.ct|static|this feature requires
CLW.

ENR.
```

The tree-control is too small. So we just make it larger.It looks like this (step-by-step):
After the:

```
SWP.10,100|466,166|C
```


© 2023 Theo Gottwald * https://www.smart-package.com

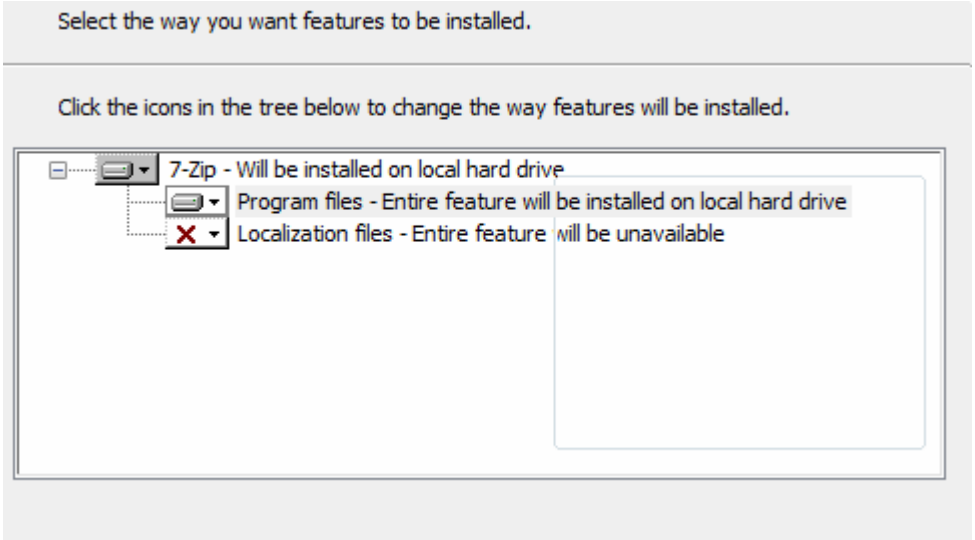


it looks a bit chaotic, but we are not yet ready. There are two statics in our way. As statics are child-windows, we can make them invisible, or just close them using the CLW.-command.

After these two lines:

```
SCW.ct|static|The complete
' Close the text-child-window
CLW.

SCW.ct|static|this feature requires
' Close the text-child-window
CLW.

' End the Testscript here
ENR.
```

These statics are history. We have a clear view now:



This means we have enough space also!

Now we try to solve the next problem: the menu doesn't open!
We use this code from the Editor.

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
WII.30
TVI.7-Zip*\Localization files*|1
MMV.
PAU.1
ENR.
```

What we see is, that the mouse moves really to the right point in the tree. But the click seems not to work! Next we try it ourselves.

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 ed
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
WII.30
TVI.7-Zip*\Localization files*|1
MLE.
PAU.1
ENR.
```

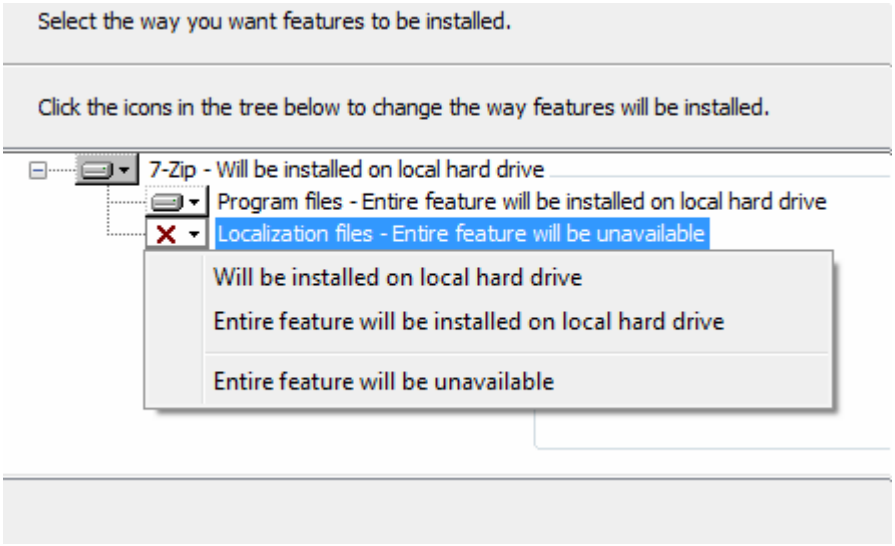
We have changed the MMV. (Mouse Move) to a MLE. (Direct Engine Mouse Click).
Starting the script, all we see is that the selection of the tree-item changes.
But no menu opens.

In such cases, we need to switch over to the "real Mouse-Click". We need the "compatible engine". In short, we need to use MLI. or MLC.

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edi
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
WII.30
TVI.7-Zip*\Localization files*|1
MLI.
PAU.1
ENR.
```

And, voila - problem solved, here is our menu. That will do the Job! See the picture below.

© 2023 Theo Gottwald * https://www.smart-package.com



Now we just need to put the whole thing together.
First we prepare the tree-control.

```
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
SCW.wctid|45|SysTreeView32|Tree of selections|169|1

SWP.10,100|466,166|C

SCW.ct|static|The complete
CLW.

SCW.ct|static|this feature requires
CLW.
```

Then we make our selection:

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
SMH.unavailable
TVI.7-Zip*\Localization files*|1
MLI.
WII.30
PAU.1
ENR.
```

Our Final, Functional Code

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
SMH.unavailable
TVI.7-Zip*\Localization files*|1
MLI.
WII.30
PAU.1
ENR.
```

Here's our final, working code.
If you're using this for software distribution, please note that it employs the "compatible engine."
As such, it must be distributed following the rules applicable to the compatible engine which needs a real Mouse-Pointer.

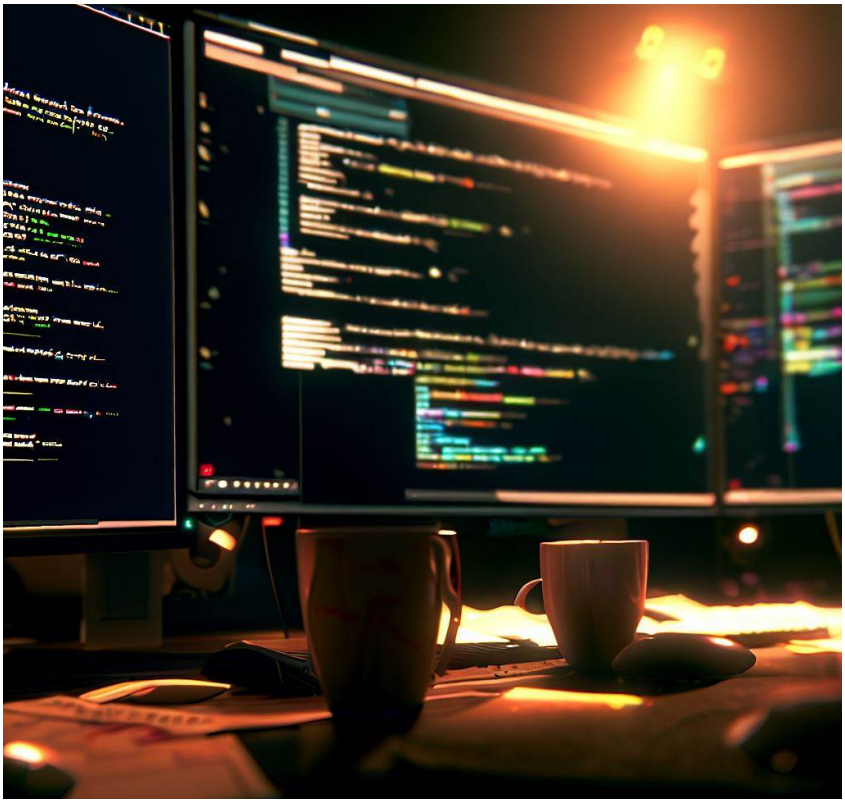
© 2023 Theo Gottwald * https://www.smart-package.com

1.32 1.B Error Management

Starting Guide

Error Management

How can we prevent error messages from being displayed to clients?



An Error is not something bad. It is a hint that shows where the right direction is.

What occurs when an error arises?

There's a high probability that you're using this program for software distribution or in other production environments.
In such scenarios, you wouldn't want error messages to be displayed.
The first tool you need for error handling is the command:

USM. [number]

In our subsequent tests, we'll utilize the following script and simply alter the parameters of the `USM.` command to observe the outcomes.
Here's the test script for reference:

```
'**********************************
' USM.-Demo
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

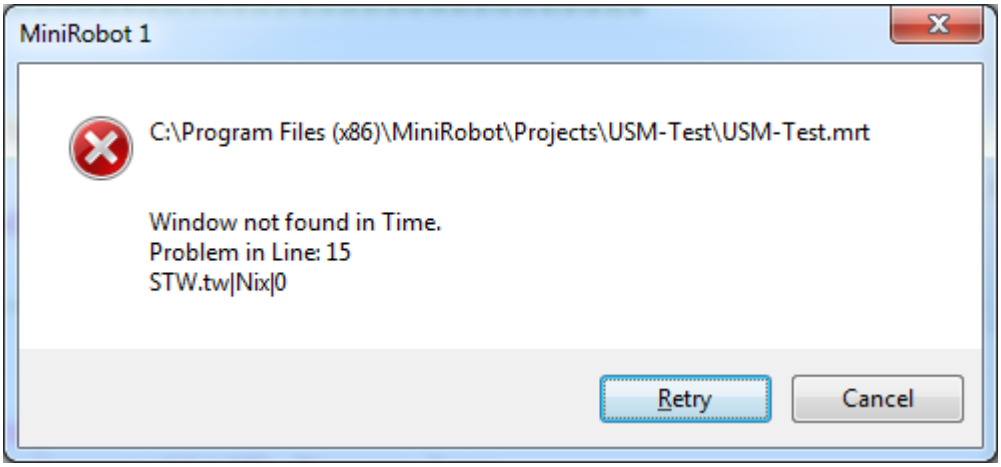
```
'**********************************
'
USM.32
STW.tw|Nix|0
MBX.Ready
ENR.
'------------------------------------------------------------
:on_Error
: $$ERF=?path\MR_Error.log

IEF.$$ERF
  COP.$$ERF|?cdesktop\MR_Error.log
EIF.
MBX. AN error happened!
ENR.
'------------------------------------------------------------
```

Manual Error handling starts with

```
USM.32
```

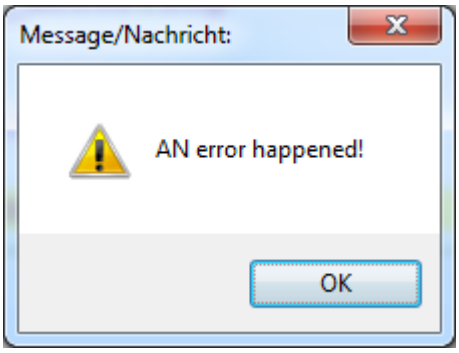
This is the default setting if you don't include the `USM.` command in your script. When you use `USM.32`, you'll get a response similar to the one shown in the image below in the event of an error.

Depending on whether you select "Retry" or "Cancel", the script will perform different actions each time.



If you select "RETRY", the script will internally treat the line where the error occurred as a comment, and it will continue executing from the next line as if no error had occurred.
As a result, you'll see the Messagebox from the normal script termination, as shown in the image below.

© 2023 Theo Gottwald * https://www.smart-package.com



If you pressed "CANCEL", which is the recommended selection, the script will go into "Error-Mode" and do one of two things.

1. If there is no Label `:on_Error` present in your actual script, the script will end immediately.
2. If there is a Label `:on_Error` present in your actual script, the script will just continue running at that label.

If the error happens in the section of the Label `:on_Error` then the script will not jump to this Label again, of course.

Lets take a look on this special Label. You can see how it really looks in the picture below.
One important thing is that you have a regular script end before the Label :on_Error.

You can use the `END.` or the `ENR.` statement for this. If you fail ending your script before that Label, then the script will anyway run into that code that comes after the :On_Error Label. Even if there was no error. And this is in most cases not useful.

```
' Code-Ende
END.

:on_Error
' If this command is executed an error has happened
SRV.$$PRO
' and the return code from the script will be the block number o

' Code-Ende
END.
```

`:on_Error` is a special Label. Its not a Label like any other!

If **`:on_Error`** is there, the script will jump there after an error and continue script execution there. `:on_Error` does NOT prevent error messages, For this you have to use the USM.-command.

InstaRec makes your scripts in a way that you can find the source of Errors.

Assume an error happened. The script runs into the `:on_Error` subprogram.

Now the next thing that happens is the `SRV.` - command. `SRV -> Set Return Value.`

© 2023 Theo Gottwald * https://www.smart-package.com

SRV. will set the Exit code for your compiled script. The Exit- or Return-code is likely the code that is shown in the Management console of your software distribution program.

Now the Full line is:

```
SRV.$$PRO
```

Hint: While the actual Robot program is the "**MR01.exe**", the executable that you have compiled is another program. If you just call the "MR01.exe" manually, use SEC. to set the "Return Error-Level". In all other cases SRV. will work fine.

$PRO is a variable. This means that it is replaced with a value at runtime. For more info on variables see here:

2.3 Variables, global and local 366

Then, what value has $$PRO at this time?
To understand this, let's take a look on a real recoding and see what $$PRO is.

This is an shortened example code, that InstaRec provides.

```
VAR.$$PRO=10002
STW.HBJwct|45|MsiDialogCloseClass|7-Zip 4.57 (x64 edition)
Setup
SCW.wctid|45|SysTreeView32|Tree of selections|169|1
WII.30
SFW.
SMH.Will be installed on local hard drive
TVI.*will be unavailable|i
WII.30
```

In the First Line of each block is the secret:

```
VAR.$$PRO=10002
```

This Line will set the Variable $$PRO to the value of 10002. We call this value the "blocknumber". Because each block of recorded code will have such a number.
If an error happens in this block, then $$PRO will give you a return code of 10002.

Why do we use 10002 and not just 2 ?

The reason is that the normal DOS Error Messages are in that range. Using Block number plus 10000, we are in a range that is free. Therefore if your software distribution tool shows you a return code of 10000+, then you know there has been a problem with a robot-script.

What do we get if there was no error?

In that case, the script will just run into the SRV.0 at the end of the script.

© 2023 Theo Gottwald * https://www.smart-package.com

```
' No.:12 | Time: 16:19:37 | Ticks:  1295
VAR.$$PRO=10012
' This command will set the return code.
SRV.0

' Code-Ende
END.

:on_Error
' If this command is executed an error has happened
SRV.$$PRO
' and the return code from the script will be the block number

' Code-Ende
END.
```

And we get a return code of zero. Which says that no error happend and all was Ok.

The combination of

```
SRV.0
END.
```

can be shortened to

```
END.0
```

from this version. End will internally just do the same, if it gets an Parameter.

Now how can we get rid of the messagebox with the error?

In case you distribute automated installations to clients, you may not want that error-messages appear on the clients desktop. Instead you want to have them in a log-file.

You can do this using the `USM.10` command.

The USM.10 is an simpe addition of an USM.8 and a USM.2.
It will do two things:

1. USM.8 will redirect all Output from Messageboxes into the console-window.
2. USM. 2 will redirect all console-window output to a file

And how do we get the real error-message?

This way we will get all Error-Messages into a log-file. This log file is by default at two possible places.

1. If you run the script from the editor, the log-file is in your "Project folder".
   That is, where the script itself is located.
2. If the robot runs without an Editor, the log-file is on the ?cdesktop\345.

You can however specify any other place for this error-log file and override the default settings, if you add a second parameter to the USM.-command. Like this:

```
USM.10|?pfad\My_Error.txt
```


© 2023 Theo Gottwald * https://www.smart-package.com

There are more less often used options for your error-handling

While having the script run without visible errors, and getting the error into an log-file at the place of your choice is all you will need in most cases. There are more choices.

Most of them just for special cases. The `USM.`-command with the "**0**" as parameter, will prevent the robot from showing any error-messages. The script will just immediately end.

However this will not help us in most cases. We will therefore look at better combinations.

`USM.4`

1. will ignore any error,
2. comment that line
3. and continue running the script as if nothing had happened.

Be careful with this option. If you add more options, the "4" will anyway prevent the `:on_Error` Label from being called, and also prevent the script from being ended.

`USM.16`

is another Option. `USM.16` will show up a **Messagebox** that will stay for less than a minute an then **automatically be closed**. After this, the script will continue running at the `:on_Error` Label, or just end if no such Label was found in the actual script.

`USM.8`

will redirect all Messages to the console-output.

`USM.2`

will redirect all Concole-Output to the a file. Note that this will redirect all console Output to the Error-Log-File.

`USM.1`

will just make a Message-Beep if an error occurs. It will not do anything else.It makes only sense, if you add the 1 to other options. For example, you can use `USM.11` which is 10+1.

`USM.64`

If an error happens, after the script already jumped to the `:on_Error` Label, this Error will not jump to the `:on_Error` Label again. Therefore after a Jump to the `:on_Error` Label, a internal flag is set, that prevents the robot from jumping there again. This is to prevent a endless-loop.
The `USM.64` will clear that internal flag. After it was called, the script will jump to the `:on_Error` Label, even if the error appeared after `:on_Error` Label was already been used.

*© 2023 Theo Gottwald * https://www.smart-package.com*

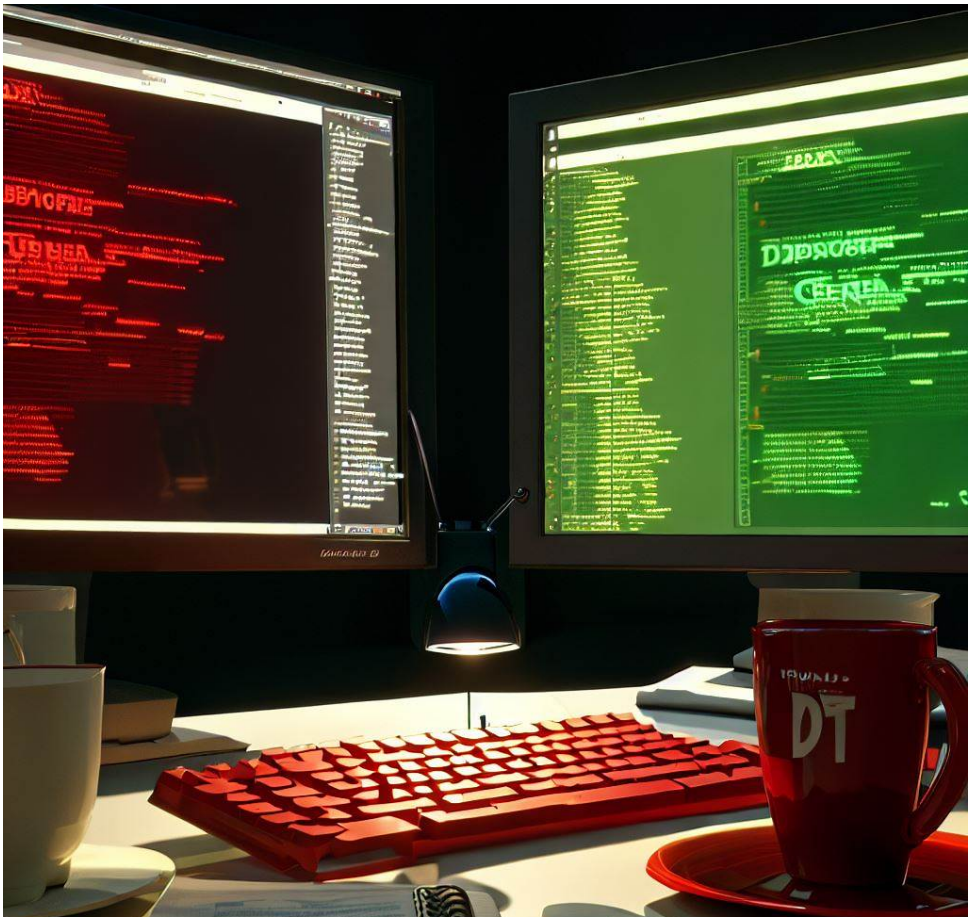
© 2023 Theo Gottwald * https://www.smart-package.com

1.33 1.C Direct Engine and Compatible Engine

Starting Guide

Direct Engine and Compatible Engine

What it is and what it is good for.



The two engines: "direct engine" and "compatible engine".

When simulating Mouse and keyboard there are generally two ways to do so. Most often we go the direct way and talk directly to the "Target-Application" or Setup-Program. We sent them a Message telling them "Hey App, you got a Mouse-Click.". Or "Hey App, there is a keyboard event for you.".

Direct and compatible Engine only have to do with simulating Mouse and Keyboard action. They have nothing to do with window or Element-Localisation.

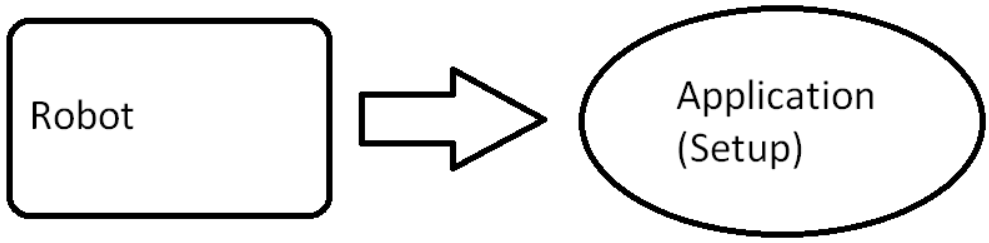
*© 2023 Theo Gottwald * https://www.smart-package.com*

This way we can work very efficient in simulating mouse and keyboard. Because the limit is just the speed in which the target application understands our messages. Here are some advantages of using the direct engine.

- Fast execution, capable of sending multiple commands per second.
- Can interact with invisible windows and disabled windows.
- Can interact with windows that are behind other windows.
- Does not require a physical mouse, keyboard, or desktop.
- Enables the execution of scripts without a logged-on user.
- Typing text is particularly fast with the Direct Engine.

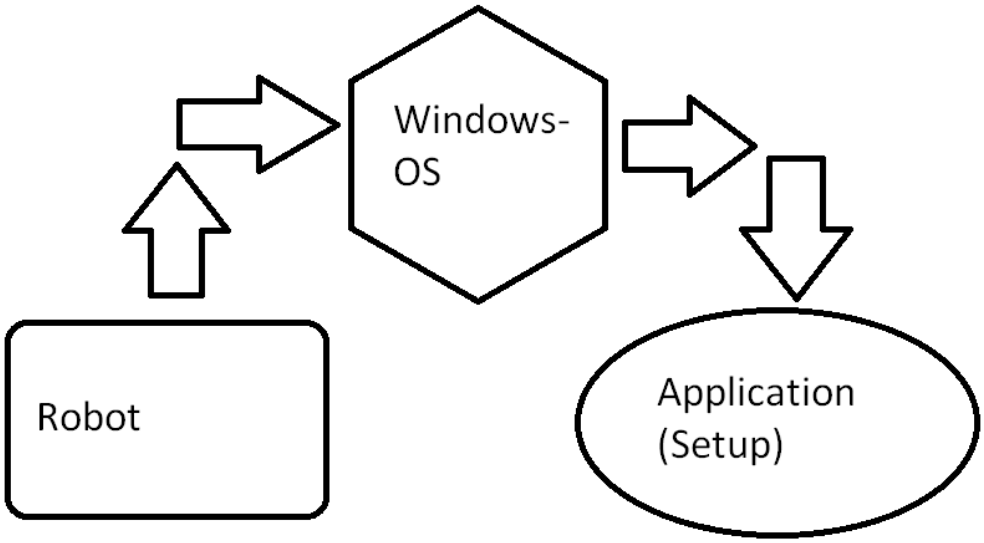
About the speed, there are differences between the commands. Typing text is much faster then using other options. You may be able to fill ten or more text fields in a second using the direct engine.

The Direct Engine



But unfortunately, not all applications and Setup-programs are willing to talk with the robot directly. Few of them, will just not react on direct engine commands. In this case we need to do what a manual user would do. We need to use the mouse and the keyboard. As we are very compatible this way, this is called the

The Compatible Engine.



The Compatible Engine sends the mouse and keyboard simulations to windows. Just like you, when you use the mouse and the keyboard. Therefore it will work in all

© 2023 Theo Gottwald * https://www.smart-package.com

applications for sure*. However being compatible has disadvantages in terms of speed.
Here is a list with the characteristics of the compatible Engine commands:

- Sends mouse and keyboard simulations to windows, mimicking user actions.
- Ensures compatibility with all applications.
- Slower compared to the Direct Engine but faster than manual interaction.
- Works on windows where the Direct Engine may not be effective.
- Cannot interact with invisible or deactivated windows.
- Requires a physical mouse, keyboard, and desktop.
- Suitable for scripts that run when the user is logged on.

Please note that the choice between the Direct Engine and the Compatible Engine depends on the target application's responsiveness and compatibility.
In most cases, the Direct Engine provides faster and more efficient automation, but the Compatible Engine ensures broader compatibility across various applications.

And how do you select between the two engines?

That's easy. You just choose the right command. You will find on the help-page of each command a hint, which engine is used.

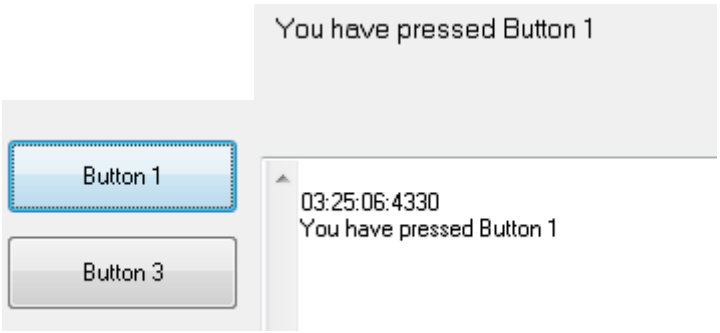
Some commands support only one of both engines. In this case there is another command that supports the other engine. Some commands support all of the two engines and you choose by giving a parameter which engine to use..

Let's test both engines so you get a feeling for the difference. First you should start the Test-Application in the "Sample Applications" Folder.

Then just run this script:

```
'Move pointer to Window-Area.
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Button|Button 1
MLE.

ENR.
```

Here is what you see in the Test-Application:



The Button gets selected and the Application tell you that "Button 1" was pressed.
However you did not see the mouse do anything, right?

© 2023 Theo Gottwald * https://www.smart-package.com

That is because the mouse is not involved when we use the direct engine. Now we change the program - just a bit. In fact we use the compatible Engine now. Look at the script.

The only change was that we have replaced the "`MLE. (Mouse Left Event)`" with a "`MLC. (Mouse left Click)`" command.

```
'Move pointer to Window-Area.
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Button|Button 1
MLC.

ENR.
```

That is all. Now it uses the compatible engine. Start the script again. Now you see the Mouse-Pointer move to the "button 1". In the same way like if you had clicked it manually. Thats why its called "compatible". It's just like "If you do it yourself". Anyway, moving the Mouse around is a bit slow. There must be a faster way, which is still compatible. And that is called "`MLI. (Mouse Left Click Invisible)`".

Try it yourself and replace the `MLC.` with an `MLI.` and run the script again. The effect on the Test-App is the same. However its a compatible click, that will work very fast in most of all cases. In those very few cases, where `MLI.` will not work, you can use `MLC.` or you can try `MLM (Mouse left Click Move).` And now we get to the next question:

How do i know which engine to choose?

The answer is simple. You have to try. There is no way to see or know it without trying. Record your setup using the InstaRec. Test your script from the Editor.

In most cases, its not the whole Setup, that does not react to the direct engine. That's very seldom. In most cases, its a control with settings, that needs the direct engine click.

- First thing to do is, record your script, using the direct engine.
- Test is slowly from the editor
- If you see that some clicks are not pressed, choose compatible clicks at these place

This is the way to do it. Here a short list with the most often used commands and a hint for the Engine:

```
MLE. - Direct Engine
MLI. - compatible Engine (faster then MLM.)
MLM. - compatible Engine (faster then MLC., more compatible
then MLI.)
MLC. - compatible Engine (Most compatible)

SDT. - direct Engine
RST. - direct Engine
SKP. - compatible Engine

SCI. - direct engine
CDD. - compatible Engine
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
BTC. - 0/1 - direct Engine, 3,4 compatible Engine (Details see
command)

TVI. / TVF. - direct Engine (depends on Options)
```

* There may be applications that defend actively against simulated mouse and keyboard input. In these cases. even the compatible engine may not work.

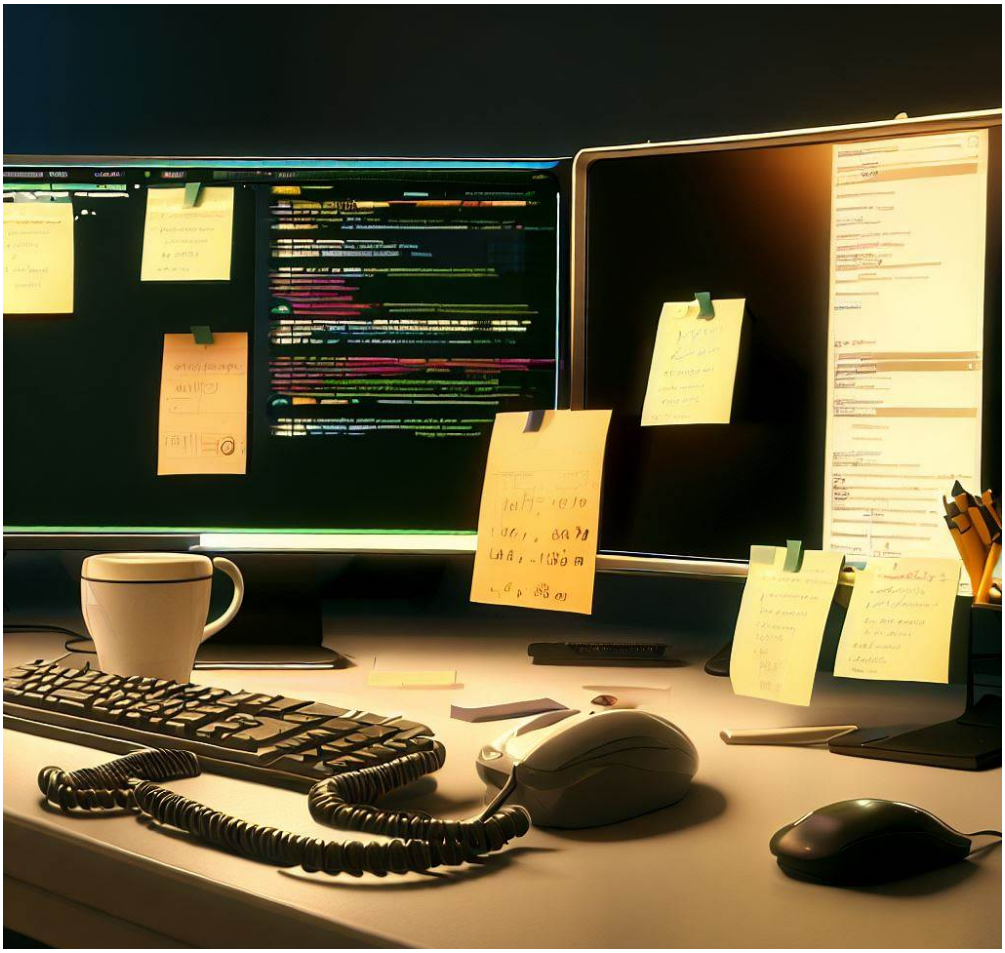
© 2023 Theo Gottwald * https://www.smart-package.com

1.34 Contact

1.34.1 Hotline

Company Information

HOTLINE



How AI thinks an helpdesk could look like.

Get help from our professionals

H O T L I N E

MiniRobots Headquarters

Theo Gottwald
Herrenstr. 11
76706 Dettenheim

*© 2023 Theo Gottwald * https://www.smart-package.com*

Telephone: +49 (0)7247 98 51 112
Telefax: +49 (0)7247 98 51 113
Mobile: +49 (0)160 66 88 222

We are also available in SKYPE:

https://join.skype.com/invite/ljhyWgLkTXdm

Office opening time

Monday to Friday from 07:00 to 21:00
Saturday from 08:00 to 12:00
German Time Zone (GMT + 1)

© 2023 Theo Gottwald * https://www.smart-package.com

1.35 General Issues

1.35.1 C. Licence Issues

Company Information

Licence Issues

This Software is copyrighted.
(c) 2012-2023 Theo Gottwald
All rights reserved.

Theo Gottwald
Herrenstr.11
76706 Dettenheim
Germany

http://www.smart-package.com/
e-mail: info@smart-package.com

This Software may contain Software (MCFS) from Stefan Schnell.
e-mail: Schnell@smart-package.com

This Software may use Scintilla in one or more of its files.

The names "Microsoft", "MS Excel" "MS Word" and such are properties of their owners.
In this Informational Text all such Brand-Names or Trademarks are just use to explain technical issues.

License for Scintilla and SciTE
Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org>
All Rights Reserved
Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Pictures:
We used some pictures that have been downloaded from external WEB-Sites like Dreamstime.com.
We do not claim any copyright on these pictures, we just use them for educational purpose to show image processing strategies.

© 2023 Theo Gottwald * https://www.smart-package.com

1.35.2 B. General Explanation

Starting Guide

General Explanation

Intention

This area explains what the command is good for.

Syntax

Command. [P1][|P2]

Parameter Explanation

Here you'll find an Explanation of the parameters, which are allowed or required for the explained command.

Some commands support two or more different syntaxes, so, those commands may have more than one parameter explanation after each syntax example. See below.

Some commands may take more than one kind of parameter, in such cases, there may be not a parameter explanation, but included examples, in the **Example** section.

```
P1 ... Px - Parameters 1 .. x

STR - String-Parameter (Any Binary is valid)
TXT - Text-Parameter (should be Text-Characters, numbers)
FILE - Filename with complete path
FOLD - Foldername with complete path
SCFN - Scriptfilename: with complete path if elsewhere,
filename only if its in ?path
VAR - Any valid Variablename can be given as parameter
LNG - 32 bit signed Integer
QINT- 64 bit signed Integer
INT - 64 bit signed Integer or 32 bit signed Integer
EXT- signed high precision Floating Point value
NUM - any numeric parameter (INT or EXT)
LAB - Label in current script.
ELAB - Label in other script.
TOS  - Top-of-Stack (see SCM.)
[optional] - Parameters in [] are optional
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'******************************************************
' In this area you will find "ready-to-run" Examples
'******************************************************
```

Remarks

Some commands have special remarks or extra explanations, in this area you will find comments which are general to the command, if the command explanation requires it.

Limitations:

If the command has Limitations, you should find them here.

Unlimited means that the total Memory usage may not exceed 2 GB (Gigabyte), including all Variables and Skriptcode.

The general Number of Parameters in one Line shall not exceed 60 (actual Limit).

See also:

Here you can find other commands which could be intresting for you. Related to the command or topic explained in the page.

© 2023 Theo Gottwald * https://www.smart-package.com

1.35.3 A. Components of the System

Starting Guide

Components of the System

What's more in the box?

So, what is in the box?

InstaRec - the Recorder

SPRE.exe - the Editor

*© 2023 Theo Gottwald * https://www.smart-package.com*

MRE.exe - an old Editor for up to Win 7

WR2MR.exe - the Converter

Diff-Tool.exe - Robo-Diff-Tool

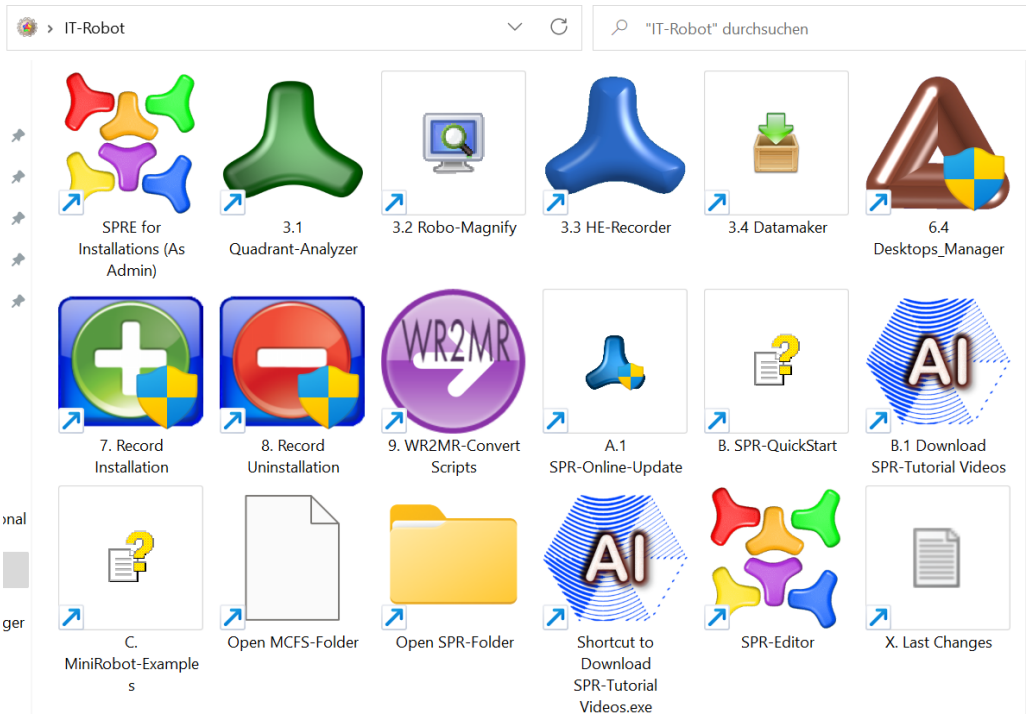
Datamaker.exe - Make Data to "Inline Data"

Pattern-Recorder - (Record graphical Patterns)

and a lot more ....

at least 5 Sample-Applications

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

2. Components

2 2. Components

2.36 1. PreProcessor & EXE-Compiler

2.36.1 1.0 Pre-Processor

1.0 Smarty's Preprocessor **Previous** 290 **Top** 41 **Next** 248

Preprocessor-Directives

Smarty's PreProcessor

The Preprocessor is the first Step in Script Execution. During this step, the Code will be changed by certain rules, before any other execution is done.

Smarty's unique Preprocessor offers features like:

- define your own commands (`MACRO`'s)
- The Preprocessor will also read the INLINE-Data between `VDA. - VDE.` and bring them into a Form where the Script can be executed with maximum Speed.
- use parameters with your own commands
- return results from your own "Macros" inside variables
- Overload Macros using `GVT.` (get Variable Type)
- react on the number of Parameters given.
- use Macros inside Macros
- use `SAV.` and `SAV.Restore` to make all used Variables "Local".
- Use VBS and Powershell inside your Macro's. (See `VBS.` and `PWS.`)
- use unique-labels and variables in your `MACRO`'s
- include your personal subprograms with just one line of code, before the script runs using `#INC:`
- Influence the script-code before its being executed

*© 2023 Theo Gottwald * https://www.smart-package.com*

To enable these features, the preprocessor analyzes and changes the script immediately before script execution.
Therefore the Preprocessor is not really a part of script-execution, its rather a script preparation.

Below is a schematic diagram of the Script execution process.
As you can see, the Preprocessor takes its part as step number two.

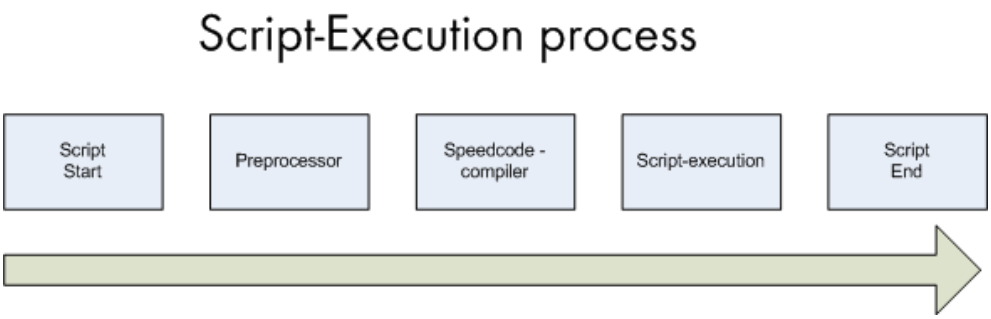
The preprocessor touches the script before execution. Therefore the execution stage will get the script that has already gone through the preprocessor.

PreProcessor commands generally start with a "#", followed by three letters. Please do not intermix PreProcessor-commands with Compiler-Commands. Compiler Commands are executed during compilation, and when the script is unpacked on the target machine.

Compiler-Commands generally start with a ' (Comment-Sign) to prevent them from being executed.

Preprocessor commands start with a "#".

Compiler-Directives start with " '#".



Sample-Macros see here: ! User-Macros 399

© 2023 Theo Gottwald * https://www.smart-package.com

2.36.2 1.1 #INC: - Pre-Processor File-Include

Preprocessor-Directives

#INC

Preprocessor File-Include

Intention

This Preprocessor-Directive will insert Code from a file at the Place where it occurs.
It can occur many times in the Skript and can even be used recursively.

#INC: is a preprocessor command, to include code-lines or complete files into your Scripts.
These can be your personal MACROS or variable definitions into your script. The inclusion of the file will be done immediately and on the lowest level.
The rest of the script will not see any difference between, if the file was included or if it had been pasted. It will be like the file plus the inclusions is one large file.

Normally, if you want to call a "SubScript" you would call `JNF.` at runtime. In that case you can use system local variables and the systen local stack. `#INC` is very different from that. It just mixes the the files into one file before they really get processed.

Assume that you have several MACRO packages that you want to load into your script, using `#INC:`. You can even nest as many `#INC:`'s to unlimited depth on your own risk.

What if each of the packages calls a main-package with your own, general definition?
How can you prevent, that a file with Macro-Definition (or any other) is included several times?

This can be done unsing the #ONCE ... #OEND 251 Option. This way you can be sure that any package with definition is just loaded once.

If you want to run your files as compiled Stand-Alone executable, make sure that all needed scriptfiles are being included into the executable package. For this you will need the compile-time command `'#INC:`. It looks similar but is something else!

Important note:
Do not intermix the Preprocessor-File-Include `#INC:` with the Compile time command `'#INC:`. The later will include files int othe executable package. While looking similar, these are two very different animal. The difference is the small " ' " comment operator. If its missing, we have the preprocessor-file include. If the comment-sign is there, its a compile.time command!

*© 2023 Theo Gottwald * https://www.smart-package.com*

To learn more about the compile time command '#INC: see
2.1.8 '#INC: - Include Files and Folders into the Executable 290

Hint: Use this command together with the 2.1.A #LIB: - User-Library Path 298 command. You can #INC: directly from the User-Library.

Also note that all ".mrt"-files that are included with the #INC: directive can be encrypted using the '#CRY: - directive.

Syntax:

#INC:P1

P1 - Filename including path to the file. If you want to run the script as Standalone "EXE" File, you should copy the file to the project folder, that is the folder where the script itself is.

Example:

For this example we need two files.The Main file, we want to execute/compile. And the Macro file that we want to include.

```
' Main-File

DBM.1
#INC:Macros.mrt
: §§VAL=5
' Here we check only the lower bound
%Check_Bounds §§VAL|6
PRT. First Result is §§VAL
MBX. Ready
ENR.


' This is the additional file,
' which is included.
'
' The Macro checks parameter 1 against
' parameter 2 as Lower Bound.
' if parameter 3 is available, parameter 1 will be
' checked against parameter 3 as upper bound
'
: %Check_Bounds 2
IVV.§§§01<§§§02
  : §§§01=§§§02
EIF.
#IF PARAMS>2
IVV.§§§01>§§§03
  : §§§01=§§§03
EIF.
#ELS
PRT.Nur ein Parameter
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
#EIF
PRT.Hallo
END%
```

Remarks:

All Commands starting with a "#" are Preprocessing Commands. These commands are used by the built-in preprocessor to change the code before execution. These commands will not be available in the code at runtime. These commands have no access to Variables and Runtime Resources. These commands will influence the code before it is actually starting.
When compiling an executable, the editor tries to include the filename after `#INC:` automatically into the package. If no path is given, "`?path\`" is assumed to be the location of the script file. Therefore its best practice to have the include-files in the same folder as the Main script file.

Limitations:
-

See also:

- : - MACRO-Definitions 253
- #ONCE / # OEND 251
- 2.1.A #LIB: - User-Library Path 298
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

2.36.3 1.2 #ONCE / # OEND - Multiple Include Protection

Preprocessor-Directives

#ONCE / #OEND.

Include code-parts only once

Intention

All Commands starting with a "#" are Preprocessing Commands. These commands will influence the code before it is actually starting.These commands have no access to Variables and Runtime Resources, as they work in the Preprocessor, which is before Runtime.
The #ONCE and #OEND is a preprocessor command. The block between #ONCE and #OEND will not be read two times. There are two versions of the #ONCE command.

First version:
The #ONCE **without any parameters**. In this case the internal preprocessor uses the name of the actual loaded file as parameter. This will prevent the whole file from being loaded a second time. even if the block is somewhere inside the file.

Second version:
To just block a part of a code, use **#ONCE [parameter-string].** In this case the parameter string will be stored in an internal memory and compared to any further call to #ONCE.
If this string is found a second time, the whole enclosed Block is been removed before any further processing is done.
This way, you can prevent files which are included with #INC: from being loaded and processed multiple times.
For example if the files contain MACRO Definitions. You do not want them to be loaded and parsed several times. Even if they are called from multiple incclude files.

There is no internal space after #ONCE, **anything after the "#"** is taken as part of the string. You can use the **:** or many other signs for formatting, these will however be part of the string. Variables are not resolved after #ONCE because this is done at Preprocessing-time, not at runtime.

Syntax:

#ONCE [P1]

P1 - string which identifies the block as unique.

Example:

© 2023 Theo Gottwald * https://www.smart-package.com

```
'----------------------------------------
' Macro/#ONCE-Example 1
'----------------------------------------
' Minimum Number of parameters is set to zero
: %MyMacroA 0
#ONCE:ME
PRT.Number of Parameters:§§§00-Parameter=§§§01
#OEND
' Replace the MEX with ME to prevent the block from
' being processed.
#ONCE:MEX
PRT.Number of Parameters:§§§00-Parameter=§§§01
#OEND
END%

' Minimum Number of parameters is set to 1
: %MyMacroB 1
%MyMacroA Mytext
PRT.Number of Parameters:§§§00
END%


%MyMacroA Mytext

%MyMacroB Mytext


DMP.
END.
```

Parameter Explanation:
-

Remarks:
-

Limitations:
-

See also:

- #IF ... #EIF - Decisions in Macros 262
- : - MACRO-Definitions 253
- 1.6 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.36.4 1.3 : - MACRO-Definitions

Preprocessor-Directives

MACRO. Definition

Macro definition

Intention

Define your own Commands!
Bundle them to build your personalized robot-language.
Use available commands and put them together to MACROs (to new commands).
Customize the scrip-language to your needs.

While not starting with a '#, this is an preprocessor command. Imagine that you can define your own commands. Save them in include files and call them in your scripts.

Macronames always start with a "%" (percentage sign, ASC 37). Before the script is executed, all MACROS are expanded to their content.

Macros can be used to make your own Instruction.
They can also be used to make your own "condition commands".
In this case, you may end up in the Editor with %IF_Something"
with a incorrect Formatting. Because Number of `IXX. / EXX.`
do not match.
To handle this, the SPRE-Editor now supports these two Pseudo-Instructions:

```
'EIF+
'EIF-
```

These will not influence the Script in any way, just correct the Formatting of the Script in the Editor.

See Example below.

```
VAN.$$NUA=1
%IF_OR $$NUA|$$NUA|$$NUA
'EIF+
  MBX.Was here
EIF.

DMP.1
MBX.1
ENR.
'=========================================================

: %IF_OR 1
SAV.$$M00
#IF PARAMS=1
CAL.$$M00=($$$01>0)
#EIF
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
#IF PARAMS=2
CAL.$$M00=(§§§01>0) OR (§§§02>0)
#EIF
#IF PARAMS=3
CAL.$$M00=(§§§01>0) OR (§§§02>0) OR (§§§03>0)
#EIF
#IF PARAMS=4
CAL.$$M00=(§§§01>0) OR (§§§02>0) OR (§§§03>0) OR (§§§04>0)
#EIF
IVV.$$M00>0
  #IF RECURS<0
EIF.
#EIF.
SAV.Restore
END%
```

Important:
We recommend that the Macro-Definitions are being placed at the End of the Script. As Macros are collected in a separate Pass, they can still be used inside the script. This way stepping through the script and testing will be easier, as you do not have always to go through the Macro-Definitions first.

Syntax

```
: %Macroname P1
[Macro Content]
...
END%
```

Let's see an very easy MACRO example:

```
: %PrintIt 1
  DBP.Hallo my dear §§§01
END%
```

Lets look at this in detail. The Line:

```
: %PrintIt 1
```

starts the MACRO definition and defines that there will be at least one parameter at runtime. Calling the MACRO with less parameters will lead you to an error:

"Macrodefinition contains less parameters as Macrocall"

Now if we call the Macro later in our script we must give the MACRO one parameter, for example a number or a text or a variable. Like this:

© 2023 Theo Gottwald * https://www.smart-package.com

```
%PrintIt Tomas
```

If we start the script, the Macro will be executed.
The DBP. command will write something like this in the Editor-Text-Area.

```
[10:22:14] [12] Hallo my dear Thomas
```

In the first bracket you can see the time when the Message was written. In the second bracket you will see the linenumber of the DBP.-command[769]. In our case its the linenumber of our MACRO.

Executing MACROs in Single step mode will make the Editor "stay on the MACRO line" until the MACRO is completely executed. You can use the DBM.-command[765].

```
DBM.2
```

will make you see every line that was executed inside the MACRO.

Macros make sense, if you need to call subprograms several times in a script.

Like this:

```
DBM.2

: %PrintIt 1
DBP.Hallo my dear §§§01
:MyDupLabel
END%

%PrintIt Thomas
%PrintIt Ralf

ENR.
```

Now imagine you call your Macro several times and it contains a label (Jump mark for GTO./ JMP. (=GOTO).

What would happen?
The Macro would expand the Label each time and we would get a Error message: *"Duplicate Label ignored."*

Similar problems could arise, if we used fixed variable names inside Macros, and these Macros would call themselves. Or would call other Macros that call these Macros.

To prevent these type of "duplication problems", the §§§95 to §§§99 preprocessor variables are provided. They will be expanded to a numeric value at preprocessing time.
This value has a defined format and will be increased by 1, each time a new MACRO Definition is been expanded. This way you can easily create unique labels and variable names.

© 2023 Theo Gottwald * https://www.smart-package.com

Let's see an example:

```
: %MoreClicks 1
' Get the parameter into a variable
VAR.$$V§§§97=§§§01
IVV.$$V§§§97=0
  JMP.Label§§§98
EIF.
FOR.$$X§§§97|1|$$V§§§97
 MLC.
NEX.
:Label§§§98
END%
```

If we call our Macro like this:

```
%MoreClicks 3
%MoreClicks 4
```

The source code line

```
VAR.$$V§§§97=§§§01
```

will expand to something like that for the first Macro call:

```
VAR.$$V01=3
```

and like this for the second Macro call:

```
VAR.$$V02=4
```

As we can see, the §§§01 contains the first given parameter. You can call a MACRO with more parameters then defined, but not with less. We could give more parameters to our MACROS like that:

```
%MoreClicks 3|hallo|99
```

They are separated with the pipe symbol. In this case the additional parameters will be in

```
§§§01 -> 3
§§§02 -> hallo
§§§03 -> 99
```

We can also check how many parameters have been given to our MACRO using the #IF ... #EIF - Decisions in Macros 262 option. The number of given parameters is available inside `the §§§00` also.

You can use the special-preprocessor variables §§§95 to §§§99 and the §§§00 to make Macro-unique Labels and temporary variables inside the Macros.

Macros can call themselves. They can call Macros that contain calls to themselves. They can be nested to any depth. This way you can use recursion to solve your problems. However, **the resolver does not check for unlimited nesting.**

© 2023 Theo Gottwald * https://www.smart-package.com

You can use the #IF ... #EIF - Decisions in Macros 262 #IF RECURS option to prevent unlimited Macro nesting, which will possibly lead to an memory error or stack overflow. Inside Macro-Definitions you can use these special variables:

```
- §§§95 - unique variable-counter variable, Format "000"
- §§§96 - unique variable-counter variable, Format "00"
- §§§97 - unique variable-counter variable, kein Format
- §§§98 - unique-Label counter-variable, kein Format
- §§§99 - actual recursion-depth at expansion time
- §§§00 - number of Parameters given to the Macro at call-
time
```

Macro-Parameter Substitution:

`§§§01` through `§§§90` will be substituted with the Macro parameters, which are given to the Macro at call-time. Macro-Parameter must be separated with | (Pipe-Symbol/ASC 124).

Important Note:
Macro Names are case-sensitive!
Using false written Macro-Names will not give an error-message!
These lines will just be ignored.

Parameter Explanation

Macroname - must start with %. Can be of any length. Can not contain spaces.

P1 - Number, defines the minimum number of parameters required for this definition

You can call a MACRO with more parameters but not with less.

Example

```
'-------------------------------------------------------------
' Macro-Example 1
'-------------------------------------------------------------

: %Press 1
PRT.Forall
#IF PARAMS=1
SCW.ct|button|§§§01
MLC.
PRT.Nur 1 Parameter
#EIF
#IF PARAMS=2
SCW.ct|button|§§§01
MLC.
PRT.Zweiter: §§§02
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
#EIF
END%

STW.ct|PBWindowClass:0|TestApp
FOR.§§LOP|1|4
  %Press Button 1
  %Press Button 2
  %Press Button 3|we did it!
NEX.

DMP.
MBX.Thats all.
END.
ENR.
'-------------------------------------------------------------

'---------------------------------------
' Macro-Example 2
'---------------------------------------
' Minimum Number of parameters is set to zero
: %MyMacroA 0
PRT.Number of Parameters:§§§00-Parameter=§§§01
END%

' Minimum Number of parameters is set to 1
: %MyMacroB 1
%MyMacroA Mytext
PRT.Number of Parameters:§§§00
END%


%MyMacroA Mytext

%MyMacroB Mytext


DMP.
END.

'---------------------------------------
' Macro-Example 3
'---------------------------------------
PRT.Running
: %Parameter 1
 ' Example for Macro-unique variable using §§§97
 VAR.$$V§§§97=unique Variable
 JMP.Label§§§98
 PRT.Forall
 #IF PARAMS=1
  PRT.Only 1 Parameter: §§§01
 #ELS
  PRT.Not only 1 Parameter: §§§01,§§§02
 #EIF
 #IF PARAMS>2
  PRT.More then 1 Parameter: §§§01,§§§02
 #ELS
  PRT.3 or more then 3 Parameter
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 #EIF
 ' Example for a Macro-unique Label, using §§§98
 :Label§§§98
 PRT.$$V§§§97
 PRT.Macro-Recursion-Depth=§§§99
 PRT.Macro- got §§§00 Parameters.
 #IF RECURS<5
  %Parameter §§§01
 #EIF
 END%


%Parameter 1
%Parameter 1|2
%Parameter 1|2|3

DMP.
MBX.1
END.

'----------------------------------------
' Macro-Example 3
'----------------------------------------
' Macro prüft und korrigiert parameter 1 gegen
' parameter 2 als Lower Bound und
' falls vorhanden gegen parameter 3 als upper bound
: %Check_Bounds 2
'Here we are!
IVV.§§§01<§§§02
  : §§§01=§§§02
EIF.
#IF PARAMS>2
IVV.§§§01>§§§03
  : §§§01=§§§03
EIF.
#ELS
PRT.Nur ein Parameter
#EIF
PRT.Hallo
END%


'----------------------------------------
' Macro-Example 4
'----------------------------------------
 PRT.1-----
 %Parameter 1
 PRT.2-----
 ' Mit DMP.1 kann man sich das expandierte Skript anzeigen lassen.
 DMP.1
 MBX.Warten
 ENR.
 '=======================================================

 : %Parameter 1
 #IF RECURS<3
 %Parameter §§§01
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 PRT.Hallo Depth is: §§§99
 #EIF
 END%


'----------------------------------------
' Macro-Example 5 using VBS.
'----------------------------------------
  'Sample:
  STW.ct|ConsoleWindowClass|Eingabeaufforderung
  MLI.
  VAR.$$NAM=DIR{ENTER}
  %VB_Sendkeys $$NAM
  ENR.
  '-----------------------------------
  : %VB_Sendkeys
  ' Use TOS
  #IF PARAMS=0
  GSB.VB_Sendkeys|$tos$
  #EIF
  #IF PARAMS=1
  GSB.VB_Sendkeys|§§§01
  #EIF
  END%
  '-----------------------------------
  ' P1 - Zu sendender Text
  :VB_Sendkeys
  SAV.Save|$$NAM
  IVV.§§_00|=|1
    $$NAM=§§_01
  ELS.
    $$NAM=Hallo Welt!{ENTER}
  EIF.
  ' Auflösung da VBS. nur vorsichtig, binärkompatibel auflöst.
  VAR.$$NAM=$$NAM

  VBS.VB|Test|$$NAM
  Option Explicit
    Function Test(x)
    Dim a
    Set a = CreateObject("WScript.Shell")
    a.SendKeys x(0)
    End Function
  VBE.
  SAV.Restore
  RET.
```

Remarks

Macro Names are case-sensitive!
Using false written Macro-Names will not give an error-message!
These lines will just be ignored.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- : - Definiton-Command 361
- #IF ... #EIF - Decisions in Macros 262
- #ONCE / # OEND - Multiple Include Protection 251
- DBM. - DeBugMode 765
- VBS. - VB-Script Start 3581
- VBE. - VB-Script End 3578

© 2023 Theo Gottwald * https://www.smart-package.com

2.36.5 1.4 #IF ... #EIF - Decisions in Macros

Preprocessor-Directives

#IF / #EIF

Two way decisions in Macros

Intention

All Commands starting with a "#" are Preprocessing Commands. The #IF commands can only be used inside Macros. See chapter MACRO-Definition 253.

These commands are used by the built-in preprocessor to change the code before execution. They will not be available in the code at runtime. These commands have no access to Variables and Runtime Resources. These commands will influence the code before it is actually starting.

These #IF Commands can be nested to unlimited Depth.

#IF is a special "**expansion-time evaluation**"

The following two **#IF ... #EIF** are supported:

```
#IF PARAMS[condition][number or formula in ()]
[code]
#ELS
[code]
#EIF
```

and

```
#IF RECURS[condition][number or formula in ()]
[code]
#ELS
[code]
#EIF
```

To influence Macro-Expansion at resolving-time. Use >,<,= or ! as condition. Please note that number must be any real number. Variables can not be taken at this place (except §§§00 and §§§99. While §§§98 can be used at this place, its expanded to a decimal number at this place. §§§97 can not be used at this place).

Inside Macro-Definitions you can use these special variables:

```
- §§§95 - unique variable-counter variable, Format "000"
- §§§96 - unique variable-counter variable, Format "00"
- §§§97 - unique variable-counter variable, kein Format
- §§§98 - unique-Label counter-variable, kein Format
- §§§99 - actual recursion-depth at expansion time
- §§§00 - number of Parameters given to the Macro at call-
time
```


© 2023 Theo Gottwald * https://www.smart-package.com

Macro-Parameter Substitution:

§§§01 through §§§90 will be substituted with the parameters, which are given to the Macro at call-time. Macro-Parameter must be separated with | (Pipe-Symbol/ASC 124).

Syntax:

```
#IF PARAMS[operator][value]
...
#EIF
```

Parameter Explanation:

Operator - can be any of: >,<,=, !
value - Number, will be compared to the number of parameters actually given to this Macro call. Based on result of the compare operation, the block is been added or removed from processing code.

```
#IF RECURS[operator][value]
...
#EIF
```

Parameter Explanation:

Operator - can be any of: >,<,=, !
value - Number, will be compared to the actual recursion depth inside macro expansion. Based on result of the compare operation, the block is been removed from processing code or not. Can be used to break recursive Macro calls at a specified depth.

Example:

```
'-------------------------------------------------------
' Macro-Example 1
' Please start "TestApp_01.exe" from the TestApp Folder
'-------------------------------------------------------

: %Press 1
PRT.Forall
#IF PARAMS=1
SCW.ct|button|§§§01
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MLC.
PRT.Nur 1 Parameter
#EIF
#IF PARAMS=2
SCW.ct|button|§§§01
MLC.
PRT.Zweiter: §§§02
#EIF
END%

'Move pointer to NonClient-Area.
STW.ct|PBWindowClass:0|TestApp

FOR.§§LOP|1|4
  %Press Button 1
  %Press Button 2
  %Press Button 3|we did it!
NEX.

DMP.
MBX.Thats all.
END.
ENR.
'----------------------------------------

'----------------------------------------
' Macro-Example 2
'----------------------------------------
' Minimum Number of parameters is set to zero
: %MyMacroA 0
PRT.Number of Parameters:§§§00-Parameter=§§§01
END%

' Minimum Number of parameters is set to 1
: %MyMacroB 1
%MyMacroA Mytext
PRT.Number of Parameters:§§§00
END%


%MyMacroA Mytext

%MyMacroB Mytext


DMP.
END.


'----------------------------------------
' Macro-Example 3
'----------------------------------------
PRT.Running
: %Parameter 1
 ' Example for Macro-unique variable using §§§97
 VAR.$$V§§§97=unique Variable
 JMP.Label§§§98
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 PRT.Forall
 #IF PARAMS=1
  PRT.Only 1 Parameter: §§§01
 #ELS
  PRT.Not only 1 Parameter: §§§01,§§§02
 #EIF
 #IF PARAMS>2
  PRT.More then 1 Parameter: §§§01,§§§02
 #ELS
  PRT.3 or more then 3 Parameter
 #EIF
 ' Example for a Macro-unique Label, using §§§98
 :Label§§§98
 PRT.$$V§§§97
 PRT.Macro-Recursion-Depth=§§§99
 PRT.Macro- got §§§00 Parameters.
 #IF RECURS<5
  %Parameter §§§01
 #EIF
 END%


%Parameter 1
%Parameter 1|2
%Parameter 1|2|3

DMP.
MBX.1
END.
```

Remarks:

-

Limitations:

`#IF RECURSE` uses a Simple Algo. Internally it will check if the last resolved MACRO-NAME is equal to the currently resolved MACRO-NAME and if the MACRO-Definition contains the same Name in Code.
In this case it will increase the RECURSE-Counter.
This may fail under complex Conditions where multiple Macros are being called from within the same Macro. It will also fail, if a **Macro A** calls a **Macro B** that will call **Macro A** again.
We recommend to carefully examine, if `#IF RECURSE` does work in your case.
If not use an non-recursive algo.

Calling Macros from within Macros is supported. "Direct nesting" of `#IF PARAMS` is supported.
Below are two examples. Both will work as expected.

```
' Manually formatted code
' Nesting of #IF ...
: %Loc_Edit
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
#IF PARAMS=0
  GSB.Loc_Edit
#ELS
  #IF PARAMS=1
    GSB.Loc_Edit|§§§01
  #ELS
    #IF PARAMS>1
      MBX.More Params are not supported (§§§00)(§§§01)!
      END.
    #EIF
  #EIF
#EIF
END%
```

```
' Example 2, nested #IF
: %Loc_Edit
#IF PARAMS=0
  GSB.Loc_Edit
#ELS
  ' Nesting is currently NOT supported
  #IF PARAMS=1
    GSB.Loc_Edit|§§§01
  #ELS
    MBX.More Params are not supported!
    END.
  #EIF
  '--- The #ELS case will be included
#EIF
END%
```

Below is the same code in a serial way that.

```
: %Loc_Edit
#IF PARAMS=0
GSB.Loc_Edit
#EIF
#IF PARAMS=1
GSB.Loc_Edit|§§§01
#EIF
#IF PARAMS>1
MBX.More Params are not supported!
END.
#EIF
END%
```

See also:

- : - MACRO-Definitions 253
- #ONCE / # OEND 251
- 2.1.A #LIB: - User-Library Path 298

© 2023 Theo Gottwald * https://www.smart-package.com

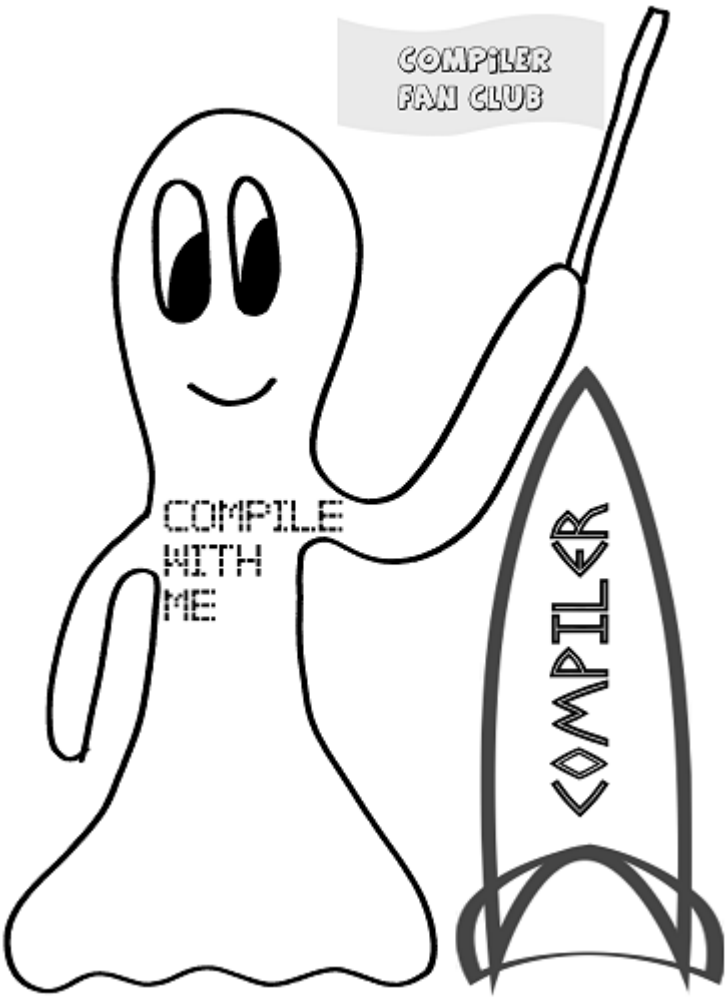
2.37 2. EXE-Compiler

2.37.1 2.0.0 Smarty's EXE-Compiler

Smarty's "Script to Executable" Compiler **Previous** 236 **Top** 41 **Next** 271

Compiler-Directives

Smarty's Script-Compiler



Smarty's script compiler allows you to **compile your script** into a small,
fast, standalone, executable.file.
To do this, just press the button "**compile**" in the Editor.

Smarty's script compiler **offers features** like:

- Compile your Script to Standalone Executable(-package)
- Add your Files to the package
- Add your Folders and Subfolders to the package using `'#INC:`
- Large Set of Special Instructions (see `'#SPI:`)
- utilizes Compression to remove data-redundancy
- utilizes Hashing, to verify the authenticity of your packages

*© 2023 Theo Gottwald * https://www.smart-package.com*

- has multiple Special Options, like:
- "'#SPI:SelfDel" (Executable can delete itself after running)
- and many more.

The generated executable are **completely Standalone**, they do **not** need any Runtime Files to be on the Target PC.

We call the process of the Package-Creation "compiling", as we compile the script-code into an Standalone Executable.

The Compilation process can be influences using a special sort of commands, called "Compiler Directives". These commands are not real commands, as they do not influence the script-behavior at runtime. They are used by the Editor, the Compiler and the "Runtime Environment" which is used to unpack your Data, when the Executable is been started.

Where is my compiled executable?

A short example for a "Compiler Directive" would be:

```
'#EXE:?desktop\MyExe.exe
```

Placing this line at the top of your script will change the name and path where the executable is been created. In that special case it would be created on your "user-desktop" and would use the filename "MyExe.exe".

If you omit any line with the Compiler Directive "'#EXE:" then the file is been created on the desktop. Note that in the script there may per default be a

```
'#EXE:?path
```

or something like that. In that case the executable is been created in the same folder, where the actual script-file was saved (that is "?path\").

How do i recognize a Compiler Directive?

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution)*.

The comment sign is important. Omitting or forgetting the comment sign, you will not get '#INC: but #INC: (no comment sign). Both are two different instructions.
#INC (no comment sign) is the Preprocessor-File-Include.

Compile-Time, that is, when the script is being compiled to an Standalone EXE-File. At ***Compile-Time*** these commands can change several behaviors.

There is another term, you should know and that is ***Unpack-Time***.
Unpack-Time is the time after the executable was started on the Target PC.

© 2023 Theo Gottwald * https://www.smart-package.com

It will then unpack all included data, together with some Runtime Files into the Temporary
Folder of the Target-PC..This happens at "**Unpack-Time**".

(User presses "Compile") (Executable is starting) (Script-Execution starts)
Compile-Time -> Unpack-Time -> Run-Time

All Options that influence the creation of compiled executable / packages are specified using
such Compiler Directives. Therefore they are saved with your script.

What is "?path\" ?

Be sure that you understand the concept of Sprecialfolders and "?path\" before you start to
include files or folders into your package. Normally you copy all files and/or folders that you
want to include with your package into the same folder, where your script file is been located.
This place is called "?path\".

During the Unpacking of your executable on the target system, Smarty will re-create the situation in "?path\" on the target system, in a temporary folder.
Therefore you address all files that are included in your script with "?path\". Because only then, they will be found on the target system.

"?path\" will always be evaluated at runtime to the path where the actual running script file is located. This way we have a relative path that can be used to find any inluded file or directory, as long as it is below "?path\".

What is "?exeloc\"?

Using the code sequence:

```
EXO.?path\
MBX.!
```

with an executable, will show you the content of the package. In case you add a "Setup.exe" File to your package, make sure that you also change the EXE. or EXX. calls to that file, in the script.

For example:

```
EXE.?exeloc\Setup.exe
```

must be changed to

```
EXE.?path\Setup.exe
```

to use the file that is inside the package, instead of a Setup that is in the same folder like the executable.

?exeloc\ will evaluate at runtime to the location where the executable itself resides.

© 2023 Theo Gottwald * https://www.smart-package.com

For example, if you have compiled a script to the file "MyScript.exe". And the location of "MyScript.exe" is on your desktop. Then `?exeloc\` will evaluate to the desktop-path at runtime.

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.2 2.1.1 '#RAU: - Run As User

Compiler-Directives

#RAU.

Run Executable as User

Intention

You can start a package in the context of another user. You have to provide the Login, password and domain (Login Information) for that user.
This command is processed by the editor and the compiler at compiletime. It is completely ignored while script execution. The Login informations are stored in an encrypted way in the package. They will be removed from the script that is finally compiled into the package. This way there is a maximum (while not perfect) level of security for your user-login information.

There are two versions of this command:

'#RAU:ask
A dialog will appear. In this dialog you can provide the user login information.

'#RAU:(Login)|(Password)|(Domain)
In this case you need to provide the user login information in the script file. Note that these will be removed from the script that is finally compiled into the package.

Syntax:

```
'#RAU:Login|Password|Domain
'#RAU:ask
```

Example:

```
'----------------------------------------
' '#RAU:ask
'----------------------------------------
' The following line will show a dialog to set the user Login
and password for the final package
'#RAU:ask
'
'MBX.!
END.

'----------------------------------------
' '#RAU:(Login)|(Password)|(Domain)
'----------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' The following line will make the package use this user('s
rights) to start the package
' You need to replace these with real account informations
'
'#RAU:MyLogin|MyPassword|MyDomain
'
'MBX.!
END.
```

Parameter Explanation:

-

Remarks:

-

Limitations:

'#RAU: does NOT cooperate with '#EXV: at this time. You can only use one of these in one compiled ex
In case you would need both, try using '#RAU: and then start another compiled executable using EXV.-C

See also:

- '#EXV: - Compiler-Directive 280
- Execution Commands and the Hidden Desktop 867
- EXV: -Execute via System 912

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.3 2.1.2 #AOS: - Choose valid OS for executable

Compiler-Directives

'#AOS: - Allowed OS

Used to prevent unpacking of a package under the wrong OS

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution)*. The '#AOS. takes place immediately after the Executable is been started on the Target PC.
It will check the OS on the Target PC against a list of valid OS'es that you can specify.

In case the OS-check fails and the actual OS does not match the List of allowed OS, per default an File "`SPR-Unpack-Error.txt`" is been created at the place where the executable was started.

The Executable-Unpack behavior that will occur, when an error happens or when the OS-check fail, can be set using the
`'#SPI:silent=0/1/2`
Compiler-Directive. For example:
`'#SPI:silent=0`
will make the Executable to show a Error-Message instead of the default behavior, that is to create an Error-Textfile.
Using
`'#SPI:silent=2`
will switch into complete silent mode. In case of an error, nothing will happen, the executable will just end itself.

'#AOS= Specify the allowed Operating System for the Package. You can use any of the following OS-Shortcuts. You can also use multiple of these shortcuts (left side), separated by comma.

```
Win95 - Windows 95
Win98 - Windows 98
WinMe - Windows Me
Win2k - Windows 2000
WinXP - Windows XP
WinVi - Windows Vista
Win7  - Windows 7
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Win8  - Windows 8
Win9  - Windows 9  (expected a this time)

' and for Servers
WinS3  - Windows Server 2003
WinS8  - Windows Server 2008
WinS82 - Windows Server 2008 R2
WinS13 - Windows Server for Win8 (expected)
WinS14 - Windows Server for Win9 (expected)
```

You can add multiple "allowed OS" separated by comma, like this:

```
'#AOS:WinXP,Win2K,Win7
```

or you can use the ">" or the "<" (optional together with the "=") Operator to specify that any OS below or above a specified OS is valid. Like this:

```
'#AOS:=>Win2K
```

If you specify multiple OS, all of these are OR'ed, means ANY of the given OS is valid. For example:

```
'#AOS:>WinVi,<Win7
```

will include ALL possible OS'es, as if you would not have used the `'#AOS`:-directive.

You can use the following Operators with `#AOS:`

```
>  - OS must be newer then specified OS
=> - OS must be newer or equal then the specified
OS
>  - OS must be older then specified OS
=< - OS must be older or equal then the specified
OS
A-B - specify a valid Range of OS, including A
and B
```

You can only specify one `#AOS`: directive per Script.
You can add as many comma separated conditions to that instruction as you want.
This way you can exactly specify under which OS the packet should run or not,

To specify a Range of OS's, use the "-" Operator, like this:

```
'#AOS:WinXP-Win8,Win2k
```

In this case all OS between WinXP and Windows 8 are valid.

Syntax:

'#AOS:[>,<,=>,<=,OS-]OS[,OS]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation:

OS - a keyword out of the following. This will specify the allowed OS.

```
Win95 - Windows 95
Win98 - Windows 98
WinMe - Windows Me
Win2k - Windows 2000
WinXP - Windows XP
WinVi - Windows Vista
Win7  - Windows 7
Win8  - Windows 8
Win9  - Windows 9  (expected a this time)

' and for Servers
WinS3  - Windows Server 2003
WinS8  - Windows Server 2008
WinS82 - Windows Server 2008 R2
WinS13 - Windows Server for Win8 (expected)
WinS14 - Windows Server for Win9 (expected)
```

You can specify as man OS'es as you want, separated by a comma.
You can add one of these Operators below before the OS to select all newer or all older OS.

```
>  - OS must be newer then specified OS
=> - OS must be newer or equal then the specified
OS
>  - OS must be older then specified OS
=< - OS must be older or equal then the specified
OS
A-B - specify a valid Range of OS, including A
and B
```

You can only specify one `#AOS:` directive per Script.
You can add as many comma separated conditions to that instruction as you want.
This way you can exactly specify under which OS the packet should run or not,

To specify a Range of OS's, use the "-" Operator, like this:

```
'#AOS:WinXP-Win8,Win2k
```

In this case all OS between Windows XP and Windows 8 are valid.

Example:

```
'---------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Compile-Time Commands
'---------------------------------------
' Special Instructions
'#SPI:silent|0
'#AOS:>Win2k
EXO.?path\
MBX.!
END.
```

Remarks:

-

Limitations:

-

See also:

- 1.3 '#SPI: - Special Instructions 287
- #ONCE / # OEND - Multiple Include Protection 251
- 1.6 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.4 2.1.3 '#EXE: - Set Executable Path and Name

#EXE: - Specify Name and Path of the Executable Previous 273 Top 41 Next 321

Compiler-Directives

#EXE: - Executable Name

Used to specify the name and path of the executable that is to be created

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

'#EXE:

The '#EXE: Compiler Directive is used to change File-Name or File-Path of the final Standalone EXE-File.

If there is no such Compiler Directive, the default path for any created executable file is on the actual user's Desktop. The filename for the generated executable is by default the filename of the Main-Script-File, with a ".exe" instead of the ".mrt".

For example, if you compile the script "InstallMe.mrt". Then the final executable will be on your desktop with the Name "InstallMe.exe". Unless you specify something else using "'#EXE:".

You can specify Specialfolders or just a File-Name. Here are some examples:

```
' If you specify this:
'#EXE:
' The compiler assumes that you mean:
'#EXE:?path\
' and will treat it like that.

Therefore, if you write:
'#EXE:?path\
or
'#EXE:?path
```

then your executable will have the default filename and will be created in the same folder, where you have saved your scriptfile.

```
' You can also write this:
' '#EXE:Peter.exe
```

In this case, your scriptfile will be created in "?path\" with the name "Peter.exe". If you prefer to have it on the desktop, write:

© 2023 Theo Gottwald * https://www.smart-package.com

```
'#EXE:?desktop\Peter.exe
```

In that case your executable will appear on the desktop.
If you just write:

```
'#EXE:Peter
```

your scriptfile will be created in "?path\" with the name "Peter.exe", as the ".exe" will automatically be added, if its missing.

Syntax:

'#EXE:[P1]

P1 - (***optional***) Name and or Path of the Executable-Package that should be created.

Example:

```
'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'
' The following line will change the target path for
' the EXE-Compiler.
'#EXE:?path\
'
EXO.?path\
MBX.!
END.
```

Parameter Explanation:

-

Remarks:

I f no `'#EXE:` is given, the Targetfile is been created on the current User-Desktop. The Filename of the Executable will be the name of the Main-Script File, with ".exe" suffix.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- Specialfolders 345
- #ONCE / # OEND - Multiple Include Protection 251
- 1.6 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.5 2.1.4 '#EXV: - Execute on System Desktop

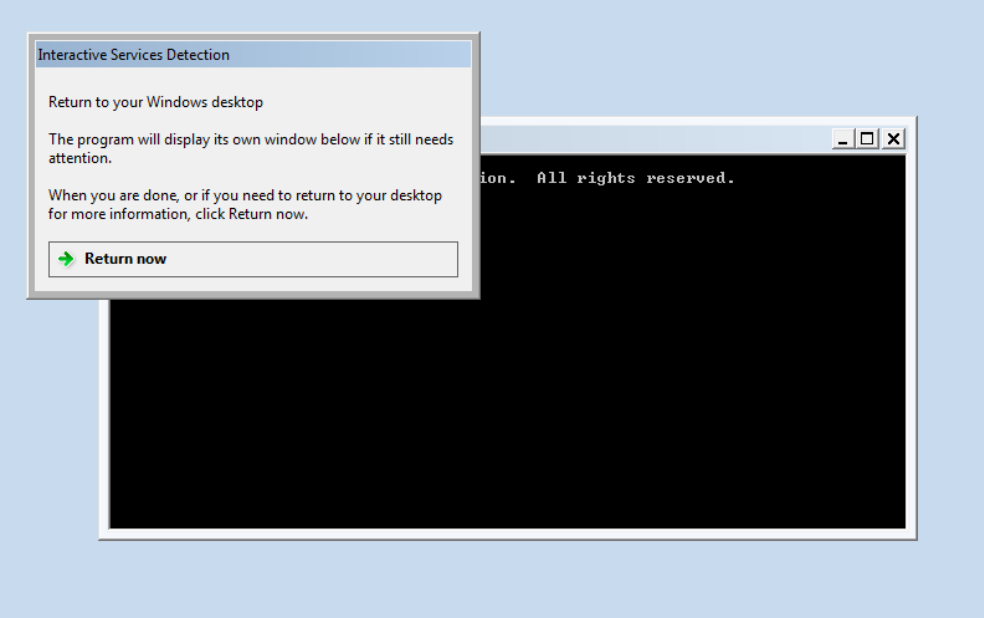
Compiler-Directives

#EXV: - Execute via System

Used to start the Automation on the Hidden System Desktop

Intention

This Compiler-Directive enables you, to start your compiled SPR-Skript on the "Hidden System Desktop".



Generally Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

'#EXV:

The '#EXV: Compiler Directive is used to have the compiled executable start up on the "Hidden System Desktop 867".
For this it must be started with full Admin-Rights!

If there is no such Compiler Directive, the compiled Executable will start on the actual User Desktop.

To take a look on the Hidden System Desktop. follow the steps here 867.

© 2023 Theo Gottwald * https://www.smart-package.com

You do not need to specify an additional parameter to '#EXV.

Use the HDC.0 command to prevent your Clients from entering the Hidden Desktop.

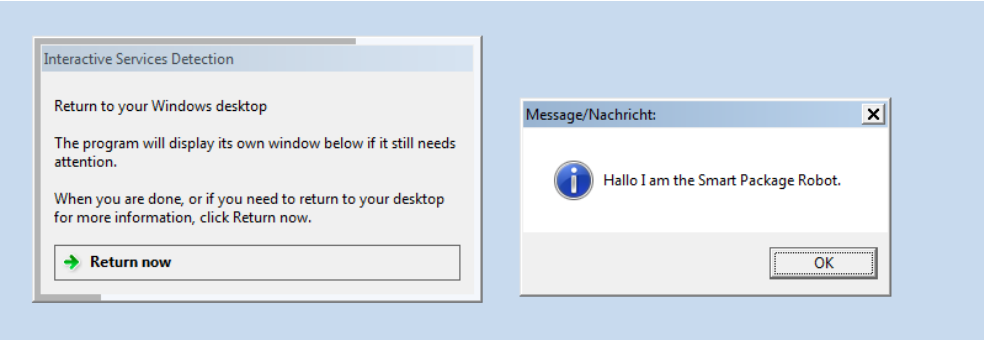
Syntax:

'#EXV:[P1]

P1 - (***optional***) Name that will be used internally. If you use several Skripts at the same time with '#EXV.
then this name should not be equal. Otherwise you can omit **P1**.

Example:

```
'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'
' The following line will make the compiled
' executable start up on the Hidden System Desktop.
' It will only work if the Script has full admin-rights.
' To see the Message box, you need to switch to the System
Desktop.
'
'#EXV:Youdontseeme
'
MBX.I am the Smart Package Robot!
END.
```



Parameter Explanation:

-

Remarks:

Limitations:

'#EXV: does NOT cooperate with '#RAU: at this time. You can only use one of these in one compiled ex

© 2023 Theo Gottwald * https://www.smart-package.com

In case you would need both, try using `'#RAU:` and then start another compiled executable using `EXV.-C`

This Feature requires Windows Vista or higher.

See also:

- Specialfolders 345
- EXV. - Execute via System 912
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 1.6 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.6 2.1.5 '#HID: - Hidden Desktop Scripts

Compiler-Directives

#HID: - Execute compiled Script on private Desktop

Used to start the Automation on a private Desktop (Hidden Desktop)

Intention

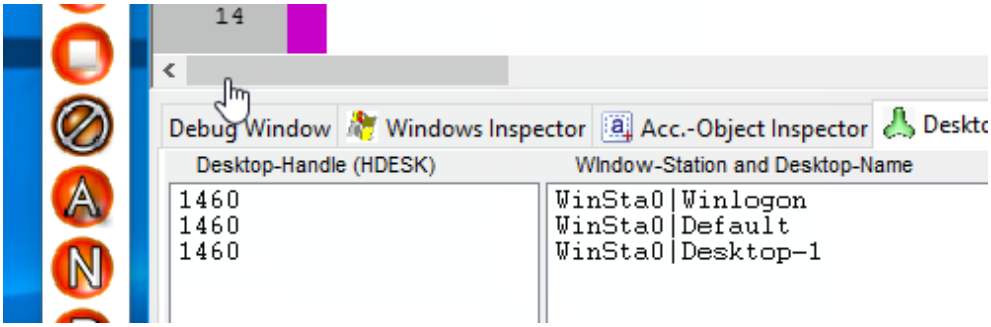
This Compiler-Directive enables you, to start your compiled SPR-Skript on a "Hidden Private Desktop".

To start SPR Scripts on a private Desktop, the Executable needs Admin-Rights.

The **'#HID:** Compiler Directive is used to have the compiled executable start up on the private Desktop.
For this it must be started with full Admin-Rights!

If you have created private Desktops, these will be displayed in the SPR-Editor, in the "Desktops-Inspector".
In this Example, you can see a Desktop that was created using:

```
'#HID:1
```



**If you specify a number after '"#HID:", the SPR will prefix ist with "Desktop-" and generate a new private Desktop
with the name "Desktop-1".**

If there is no such Compiler Directive, the compiled Executable will start on the actual User Desktop.

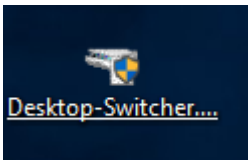
The following Sample-Script will generate a Messagebox on an private Desktop, and switch to this desktop.
Once you have confirmed the Messagebox, the SPR will close the private Desktop.

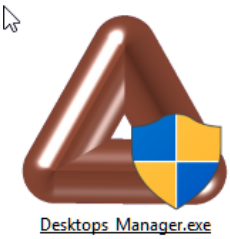
```
'#EXE:?path\
'#SPI:ForceWrite
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'#MAN:ra
'#HID:1|scn
MBX.!!
ENR.
```

Important:
To take a look on an private Desktop - for example to check for error-Messages - as well, as to
Switch back from an "private Desktop" to the "Default Desktop",
you may need to use the "Switch Desktop"-Tool that is provided together with this SPR-Release.





Also you will find the Tool "Desktops Manager.exe" in the Startmenu.

For more Information on "private Desktops" see here: PDC. - Private Desktop Commands 2658

Syntax:

'#HID:P1[|P2]

Parameter Explanation:

P1 - Number from 1-9 that will be used internally to create the private Desktop. The name of the new created private Desktop will be "Desktop-**P1**"
If you specify "`'#HID:1`", then the name of the new created private Desktop will be "Desktop-1".

You can use Numbers from 1 to 9 with `'#HID:`.
Using higher Numbers may work with Scripts, but these numbers may not work with the "Desktop-Switcher
If you really ever need higher numbers, you can switch to these Desktops using a SPR-Script that is compi

P2 - can be any or more of these letters (flags):
S - **s**witch to Desktop, using this flag, will make the Script to switch to the specified private Desktop.
N - **n**ew Desktop. Using this Flag will force the creation of a new private Desktop, rather then using an existing one.
C - **c**lose the Desktop when the Script ends.

© 2023 Theo Gottwald * https://www.smart-package.com

DANGER: Using the "c"-Flag will forcefully terminate all processes that are bound to the new created Desktop! Also note that this flag is currently experimental.

Before the new created Desktop can be closed, the Script switches to the new Desktop.
When the new Desktop has been closed, the Script switches to the Default desktop.

Example:

```
'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'
' The following line will make the compiled
' executable start up on the private Desktop "Desktop-1".
' It will only work if the Script has full admin-rights.
' To see the Message box, you need to switch to the private
Desktop.
'
'#HID:1|c
'
MBX.I am the Smart Package Robot!
END.


'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'
' The following line will make the compiled
' executable start up on the private Desktop "Desktop-2".
' It will only work if the Script has full admin-rights.
' The command will switch to the private Desktop.
' After you click the Messagebox, the private Desktop will be
removed.
'
'#HID:2|scn
'
MBX.I am the Smart Package Robot on Desk 2!
END.
```

Remarks:

'#HID: **does** work with '#RAU:.

Limitations:

This Feature may require Windows Vista or higher.
This feature needs Admin-Rights!
This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops.

© 2023 Theo Gottwald * https://www.smart-package.com

Using the "c"-Option may have unwanted effects on programs running on the Computer, also on otehr desktops.
For Limitations in the Windows-Explorer itself, not all Actions will be possible on "private Desktops".
Sometimes System-Windows - like the Start-Menu - will not open on the "private Desktop" but on the Default Desktop.

See also:

- PDC. - Private Desktop Commands 2658
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- 2.3.0. '#MAN: - Manifest for Executable 321

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.7 2.1.7 '#SPI: - Special Instructions

Compiler-Directives

'#SPI: - Special Instructions

Used to modify several compile- and unpack behavior

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

Autoinclude
Another feature, "Auto-Include" will automatically include used script files into the compiled "Stand-Alone" package. Files are known if they are used in an JNF.-Command. Also the complete filename must be resolved at compiletime.

Syntax:

```
'#SPI:[spi-code][|Parameter]
```

Parameter Explanation:

`[spi-code]` - a keyword out of the following. This will specify the special instruction.

SPI-Instructions for Compile-Time:

- **forcewrite** - no parameter, will make the compiler to overwrite already existing executables with that name. If omitted, executables will not be overwritten, but numbered.
- **NoExe**- no parameter, will make the compiler to skip creation of the executable.

SPI-Instructions that influence Unpack-Time-Action:

- **SelfDel** - no parameter, will make the Final executable to kill itself after it was started.

© 2023 Theo Gottwald * https://www.smart-package.com

- **NoWait**- no parameter, will make the Final Executable to NOT wait for the script has been ended , and to NOT delete all temporary files.
- **Startfile=** - you can specify another scriptfile as starting point. Needs a starting file as a parameter. Normally this file resides in "?path\".
- **ProgressBox=** - numeric parameter, will make a Progressbox appear at Unpack-Time.
- **ProgressText1=** - Textual parameter. Use this Text as headline for the Progressbox.
- **ProgressText2=** - Textual parameter. Use this Text as textine for the Progressbox.
- **Mutex=** - (0/1/2) numerical parameter. Specifies the type of Mutex to use.
- **NoMutex** - no parameter, will make it possible to start the same executable multiple times. This is normally prevented using "Mutexes".
- **ShortTmp=** -(1/2) Numerical parameter. specifies the path to a temporary folder that is used instead of the default temp-folder. It makes sense, to also use the "**'#SPI:UTF**"-Option also with this.
  1 -> Drive from `?exeloc\`
  2 -> Drive from Windows-Folder
- **UTF** - no parameter. Specifies that a unique Temporary folder is been created inside the given Temporary folder, for the Runtime Environment.
- **TargetFolder=** - Textual parameter. specifies the path to a temporary folder that is used instead of the default temp-folder. Can use special folders. It makes sense, to also use the "**'#SPI:UTF**"-Option also with this.
- **silent**- numeric parameter, can be: 0,1 or 2.
  0 - display an error message in case of an error before RunTime.
  1 - (default mode) write a Textfile at the same location where the executable was started.
  2 - complete silent. Does not do anything in case of an error, just ends the process.
- **forceupdate** - no parameter, will make the compiler to overwrite already existing files that are in the way of files that are being unpacked. This flag is mostly needed if files are being unpacked outside "`?path\`".
- **DiskFree** - Numerical parameter that specifies the amount of free Diskspace in MB, that must be free on the target drive, to unpack the Executable.

Some Special Instructions are only for debugging purposes.
These are:

- **NoKill** - no parameter, will make the compiler to NOT delete the internal compilation files after the compilation.
- **TextOn** - do not compile an executable, but an textual file that can easily be sent via e-mail. Needs to be decoded before it can be used. This option is not yet available.

Example:

```
'----------------------------------------
' Special Instructions
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'----------------------------------------
' Special Instructions
'#SPI:ForceWrite
'#SPI:ForceUpdate
'#SPI:silent=0
'#SPI:NoExe
EXO.?path\
MBX.!
END.
```

Remarks:

-

Limitations:

-

See also:

- Specialfolders 345
- #ONCE / # OEND - Multiple Include Protection 251
- 1.6 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.8 2.1.8 '#INC: - Include Files and Folders into the Executable

Compiler-Directives

'#INC: - Include Files & Folders

Used to add files and folders to your executable package

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution)*.

Compile-Time, that is, when the script is being compiled to an Standalone EXE-File. At ***Compile-Time*** these commands can change several behaviors.

'#INC:
The "#INC:" statement can be used to include Files or folders into the final EXE-Package

How do I include files or folders into the package?

Using the "\" Option with '#INC: will include the **folder but no subfolders**.

'#INC:myfolder\

Using the "*" with '#INC: will include the **folder and all subfolders.**

'#INC:myfolder*

This will include only the given file, which should be placed in the same folder as the script file (shortname for the script-file folder is **?path**).

'#INC:?path\myfile.txt

All of the above examples will only make sense if the specified files are within the project folder.
Now you can also include files from other places, outside the project folder.
For this the:

->

Operator can be used. Usage is like this:

© 2023 Theo Gottwald * https://www.smart-package.com

'#INC:?desktop\Infos.pdf->?path\

Anything on the right sider of the -> Operator is treated as a folder.
Therefore, you can not use the -> Operator to rename files.

For example:

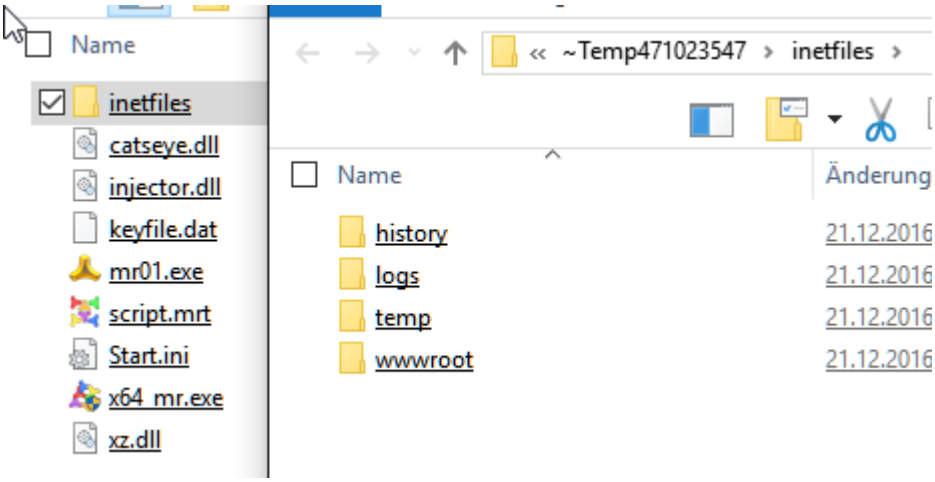
'#INC:?desktop\Infos.pdf->?path\XY.pdf

Will result in a Folder "XY.pdf" to be created. Insider the folder you will find then "Infos.pdf"-file.

You can use the -> Operator for files, and also for folders and sub-folders.

'#INC:c:\inetpub*->?path\inetfiles\

Will result in something like this, when the executable is unpacked.

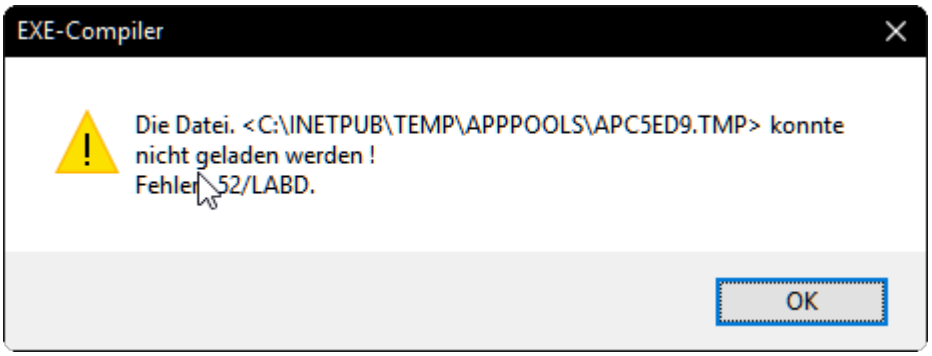


What happens if there are files that can not be included, because they are "Locked" by windows?

In this case the Compiler will bring up an error-message like this.

Detailed Description of Error-codes about files can be found online, here and on other places:
API-File Error-Codes

© 2023 Theo Gottwald * https://www.smart-package.com



Autoinclude:
Script-Files that are used with `JNF. JNJ.` or #INC: (Preprocessor File-Include) are automatically added to a executable package, if the path does not contain any variables.
Thats is, because the value of variables is undefined at compiletime. Specialfolders like `?path\` are ok, and will not block Autoinclude of Scriptfiles.

Will Include-Directives in Library files be found?

Yes, any "**.mrt**" file that is referenced with a "`JNF.`"-like command or included in the EXE, will be read completely and all include statements that are found will be evaluated.

This way, you can include selective Library files, using:

'#INC:?path\..\MyLib\Thisfile.mrt->?path\LibXXSPR\

Inside the Library file.Therefore in the calling Script you do not need to do anything else then to call the Script-file using `JNF.` or `#INC:`, if this is specified inside the Library-file.

Syntax:

'#INC:

Example:

```
'----------------------------------------
' Compile-Time Commands
'----------------------------------------
' The following line will include the specified folder and
all its subfolders into compiled EXE.
'#INC:C:\Files\*->?remote\
'
' More Examples follow
'#INC:C:\Files02\*
'#INC:?path\Install-Pack\*->?remote\install-Pack
'#INC:TabMdi.h.bak
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' The following line will change the target path for the EXE-
Compiler.
'#EXE:?desktop\
'
EXO.?path\
MBX.!
END.
```

Parameter Explanation:

-

Remarks:

Please note that the '#INC: and the #INC: (no comment sign) are two different instructions. #INC is the Preprocessor-File-Include.

Using the code sequence:

```
EXO.?path\
MBX.!
```

with an executable, will show you the content of the package. In case you add a "Setup.exe" File to your package, make sure that you also change the EXE. or EXX. calls to that file, in the script.

For example:

```
EXE.?exeloc\Setup.exe
```

must be changed to

```
EXE.?path\Setup.exe
```

to use the file that is inside the package, instead of a Setup that is in the same folder like the executable.
.

Limitations:

-

See also:

- 1.4 '#EXC: - Exclude Files or Dirs from Executable 295
- 1.3 '#SPI: - Special Instructions 287
- 1.7 #ONCE / # OEND - Multiple Include Protection 251
- Specialfolders 345
- 1.6 #INC: - Pre-Processor File-Include 248
- 2.1.A #LIB: - User-Library Path 298

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.9 2.1.9 '#EXC: - Exclude Files or Dirs from Executable

Compiler-Directives

'#EXC: - Exclude from Package

Used to prevent files and folders from being packed into the executable.

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

At *Compile-Time* these commands can change several behaviors. Compile-Time, that is, when the script is been compiled to an Standalone EXE-File

'#EXC:

The "'#EXC:" will exclude files or folders from the Package.

Important:
Use the `'#EXC:` - Directive **before any other directive** that may include files or folders. The `'#EXC:` will starting from the time it was read. Lines above the directive are not influenced.

The `'#EXC:` understands the File-Wildcards, these are:
* - match anything that follows
? - match to a single character, letter or number
- match to a numeric character (0-9)

Examples:
' The following lines will exclude files from the package.
```
'#EXC:Folder*.exe
'#EXC:Folder.e??
'#EXC:Folder*.*
'#EXC:*.bmp
```

' The following lines will exclude Dirs from the package.
```
'#EXC:mydir\
'#EXC:my*\
'#EXC:Sub Fol*\
'#EXC:Subfolder\Subfolder_??\
```

How do I exclude files or folders into the package?

Using the "\" at the end or inside the specified parameter, will exclude a **folder**.

© 2023 Theo Gottwald * https://www.smart-package.com

```
'#EXC:myfolder\
```

Just giving a name without a \ will exclude a File..

```
'#EXC:myfile.bmp
```

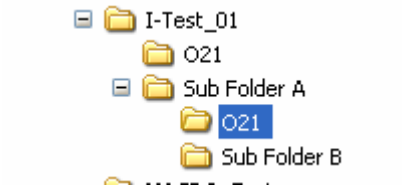
... or a group of files. You can use wildcards for this.

```
'#EXC:*.bmp
```

You can use special folders, and you can use wildcards. Excluding a folder will always exclude all subfolders and sub-files from this folder.

```
'#EXC:?path\reserve\
```

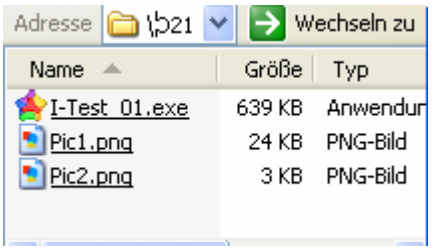
You can also give a path between folders, to exclude a specified folder, even if there are multiple folders with the same name on different locations.




```
'#EXC:*A\O21\
```

The given example above will exclude the folder O21 below the "Sub Folder A", but will not exclude the other O21 Folder in `?path\`.

Here is another example. Assume we have the folder "O21" in `?path\`.



Now we want to include the file "I-Test_01.exe" but wedo not want to include the PNG-Files. We can do it like that:

```
'#EXC:*.png
'#INC:?pfad\O21\*
```

This way, we include the whole folder excluding all *.PNG-Files.
Always remember that

Syntax:

`'#EXC:P1[*][?][#][\]`

© 2023 Theo Gottwald * https://www.smart-package.com

Example:

```
'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'#SPI:ForceWrite
'#EXC:*.png
'#INC:?pfad\O21\*
'
' The following line will change the target path for the EXE-
Compiler.
'#EXE:?desktop\
'
EXO.?path\
MBX.!
END.
```

Parameter Explanation:

-

Remarks:

-

Limitations:

-

See also:

- 1.5 '#INC: - Include Files and Folders into the Executable 290
- 1.3 '#SPI: - Special Instructions 287
- Specialfolders 345
- #ONCE / # OEND - Multiple Include Protection 251
- 1.6 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.10 2.1.A #LIB: - User-Library Path

Compiler-Directives

#LIB: - User-Library Path

Used to automatically include User-Lib files and folders into the executable.

Intention

The `#LIB:`-Directive will enable you to include your personal Library-Script-Files into any of your Scripts easily.
To make this possible, the commands does a lot of things "under the hood".
What you need to provide is the "Library Path". That is the Path where your Library-Skript files are.
Then all you write is this:

```
#LIB:(Library-Path)\
```

For example:

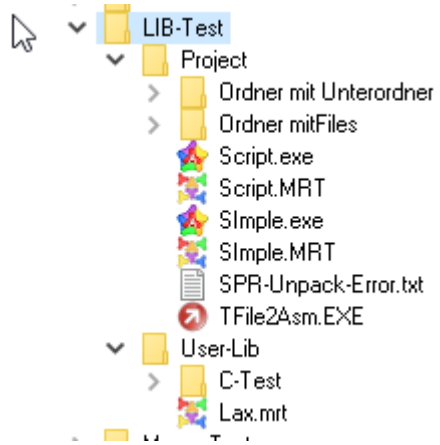
```
#LIB:X:\SPR-Library\
```

the folder can also be "relative to your Skript's Folder", using:

```
#LIB:?path\..\My-Library\
```

Here is a Sample that shows how to use the Directive:

#LIB:?path\..\User-Lib\



The "User-Lib" Folder can be anywhere, even on another drive. In this case its just at the same Level as `?path\` which is the Project Folder where the Skript itself is located. To make this possible, `'#LIB:` is a mixed Preprocessor- and Compiler Directive.
The `#LIB:` Command will therefore do a couple of things, during Compilation and during Preprocessing. These are:

*© 2023 Theo Gottwald * https://www.smart-package.com*

1. When starting a Script inside the Editor, it will also search Skripts inside the User-Lib Path.
   Example: `JNF.MyLibscript.mrt`
   will be executed if in ?path or in `?lib\`
2. The Special-Folder `?lib\` will be set to the Path that was specified behind the `#LIB:`-Directive.
   `JNF.`**`?lib\`**`MyLibscript.mrt`
   will be executed, if the file is in the Library-Path.
3. When compiling an EXE-File, the whole Folder "User-Lib" **and all Sub-Folders and files inside** will be compiled into the Folder:

?path\LibXXSPR\

This is the Path that `?lib\` will have when the Script runs in EXE-Mode (when the Script runs from a compiled Executable).
This way, the Library will be included and also deleted after Script execution.

How to Avoid to include always the whole library:
You can prevent the Compiler to generally include the whole library, by adding the `|N`
Doing so, you will have to manually add

```
'#INC:?path\..\User-Lib\WasMe.mrt->?path\LibXXSPR\
```

to your Script. You can simply add this Line into the Library file. It will look like this:

Main-Script:

```
#LIB:?path\..\User-Lib\|N
JNF.WasMe
```

Library file:

```
'#INC:?path\..\User-Lib\WasMe.mrt->?path\LibXXSPR\
MBX. Was me!
RET.
```

This way, you can just call your "Library Skript" using `JNF.Wasme`
and the Preprocessor will know that this shall be included into the Library during compilation.

You can use the `#LIB:` - Directive as well together with the
#INC: - Preprocessor-Directive. 290

Using this, you can add all used "Subroutines" to the bottom of your Script using a Single (or multiple)
`#INC:`(path and name of lib-file)

Example:

```
#INC:Subroutines.mrt
```

or

```
#INC:?lib\Subroutines.mrt
```


© 2023 Theo Gottwald * https://www.smart-package.com

Hint: To protect your Scripts and Library Scripts from being seen by unauthorized persons, you can use the '#CRY: - Directive.

Syntax:

#LIB:P1[|N]

Example:

```
'----------------------------------------
' Using the User-Library Command with #INC:
'----------------------------------------
'#SPI:ForceWrite
'#LIB:?pfad\..\User-Library\
'
#INC:MyLine.mrt
' This Subprogram will be included below with the #INC:
GSB.DoX

EXO.?path\
MBX.!
ENR.

' This will include your Subprograms
#INC:MySubs.mrt
END.


' MyLine.mrt must be in the Library-Folder and could look
like this:
'
MBX.I was here A!
'


' MySubs.mrt must be in the Library-Folder and could look
like this:
'
:DoX
MBX.I was here B!
RET.
'



   '----------------------------------------
' Using the User-Library Command with JNF.
'----------------------------------------
'#SPI:ForceWrite
'#LIB:?pfad\..\User-Library\
'
JNF.MySubs.mrt
EXO.?path\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.!
END.


' MySubs.mrt must be in the Library-Folder and could look
like this:
'
MBX.I was here!
' In this case we need the "ENR." Command to return to the
Main-Script.
ENR.
```

Parameter Explanation:

-

Remarks:

Preprocessor-Directives start with the Sequence: **#**
Because this is only important for the Preprocessor, the running script ignores these commands at ***Runtime*** *(at time of script execution)*.

Limitations:

-

See also:

- 2.1.8 '#INC: - Include Files and Folders into the Executable 290
- 1.3 '#SPI: - Special Instructions 287
- Specialfolders 345
- #ONCE / # OEND - Multiple Include Protection 251
- 1.1 #INC: - Pre-Processor File-Include 248
- 2.1.B #CRY: - enCrypt the Script 302

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.11 2.1.B #CRY: - enCrypt the Script

'#CRY: - User-Library Path Previous 295 Top 41 Next 310

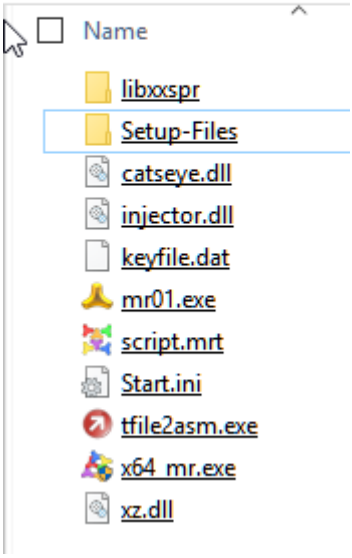
Compiler-Directives

#CRY: - Encrypt Scripts on Disc

Used to encrypt the Scripts that are stored on Disc to make them unreadable.

Intention

When you have a Script you will often compile it to an EXE-File. This EXE-File may be given to Clients or Customers.
Now when the EXE-File is been started, it will make a "Quick Install" in the Temp-Folder of the Client-PC.
It will unpack all the Scripts as well as included Setup-Files. Generally you can see this Project in the Temp-Folder as an
`~Temp(unique number)` somewhere. Here is an example:



Now somebody could "steal" your code-algorithm or included Passwords, by just opening the file inside the SPR-Editor, or even Notepad.
To make this very heavy (yet nothing is impossible in the world of hacking today) we have included the

`'#CRY:` - Directive
If you specify this anywhere in your code (can also be in a Library-File), ALL "`.mrt`"-Script-files will be encrypted.

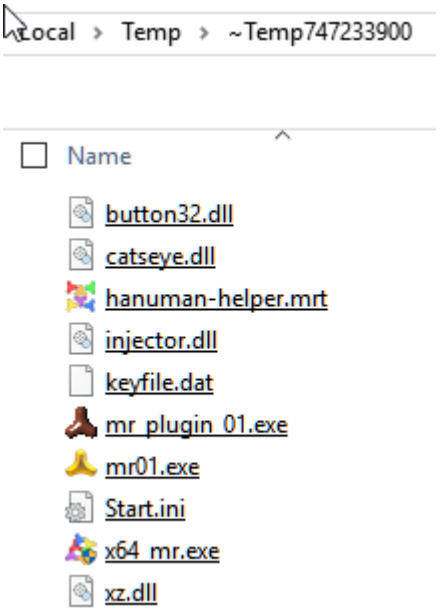
How can i check if my Scripts are really encrypted?

© 2023 Theo Gottwald * https://www.smart-package.com

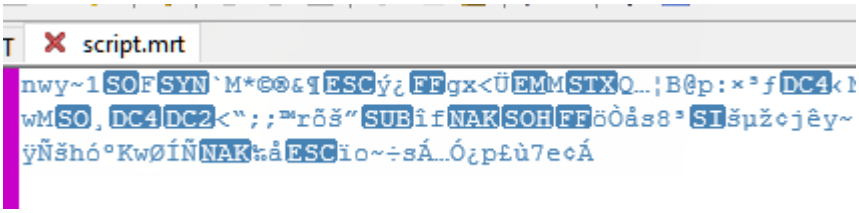
You can check if your Scripts are encrypted, by adding these statements at the Top of your code:

```
EXO.?path\
MBX.!
```

Now you compile the Script and Run the compiled EXE-File.
Once these commands are executed, the Script will open its folder, which may look similar like this:



Now just double click on the Script-File and load it into the Editor. You may see something like this:



If its completely unreadable then your Script is crypted. You will NOT be able to recognize ANY WORD that was in the Script before.

Now we are going to take a look on some encrypted Scripts to give you an impression of the Encryption.
We are going to use the following Script. This Script will "include" the small Script "X.mrt" via

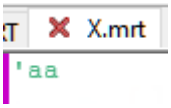
```
#INC:X.mrt
```

For Details on the Preprocessor-Include see here: 1.1 #INC: - Pre-Processor File-Include 248
Now this is the Main-Script:

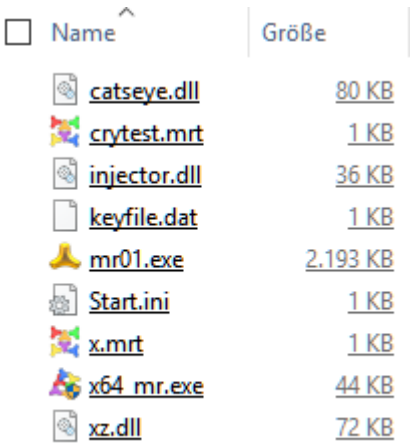
© 2023 Theo Gottwald * https://www.smart-package.com

```
'Creation·date:·12-2:
'====================
'#CRY:
'#EXE:?path\
'#SPI:ForceWrite
EXO.?path\
MBX.Ende
ENR.
'====================
#INC:X.mrt
```

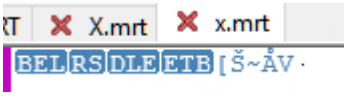
Here is the Original "X.mrt" in the Editor. As you can see its just 3 Bytes long.



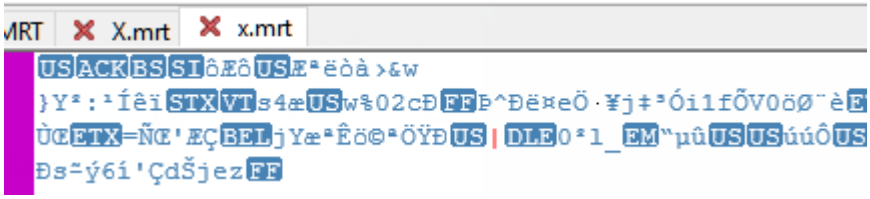
Now we compile the Script and start it. The Script will then open its temporary Folder and show us the unpacked Script-Files.



Now I am going to open the "`X.mrt`"-Script-File. It looks like this:



Ok, I will do the same again. NO CHANGE in the Code. Just take the same Script and compile it again.
Will it look equal? No, this time it looks quite different.



Ok, lets do it a third time. Compile ... and run. Open "X.mrt" in the Editor.

© 2023 Theo Gottwald * https://www.smart-package.com

Even the third time it looks totally different. Here are some Keypoints about the used Encryption-Scheme.
Now we will check something else.

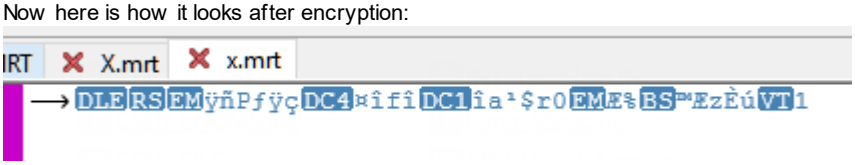
We will now use this file as "`X.mrt`". How will the encrypted version look?
The first thing we need to know is, that the "Original file" has a size of 4826 Byte.
See below.

© 2023 Theo Gottwald * https://www.smart-package.com

Crytest.MRT X.mrt

```
1  'aaaa
```

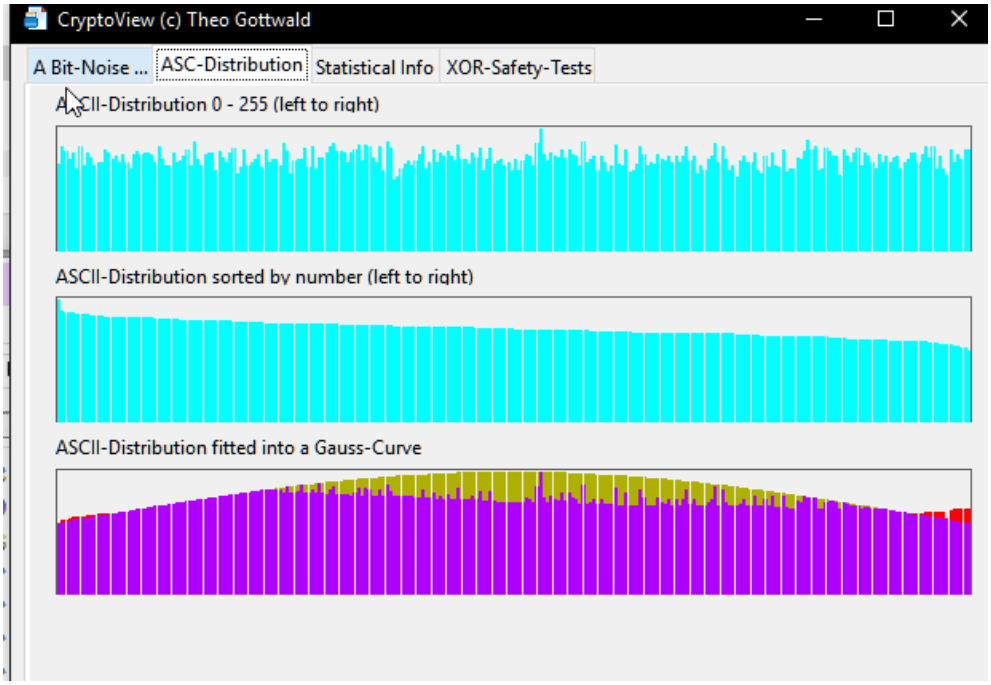
After Encryption the size of the file has reduced to 788 Byte.
From this example, you can see that the internal Algo will also remove redundancies before doing an Encryption.
Now here is how it looks after encryption:



You will not have any idea how it looked before, or even how long the original file was before.

Here are some Points about the internal Encryption Algo.
1.. Code is Compressed to reduce Redundancies
2. Code is Padded to fixed sizes.
Therefore, what you can see is that there are no valid conclusions about the original Code-Size.
3. Code is encrypted in an way that can mathematically not be reversed other then using the code that is inside the SPR it self..

YOU CAN TRUST the Encryption of the `'#CRY:` - Directive that it will definitely hide your Passwords very well.
Nothing that was readable in the Original Code will be readable in the Encrypted Code.



We added a improved Version of "CryptoView" to this SPR, and it will show you that any statistical Method of cracking the Script-Files is likely going to fail.

Having said this, there is no 100% Warranty that nobody will be able to find and extract the loaded Script in Memory or
use the built-in routines via Code-Hacking. However only Professionals with a lot of time and Know-How will be able to succeed in this.
And they need to have the SPR also.

Hint: You can use the `'#CRY:` - Directive together with the JNF.-Command as well as with the `#INC:` - Preprocessor-File-include.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax:

'#CRY:[P1]

P1 - Can only be a 0 or a 1. if omitted a "1" is used and this says that the code shall be encrypted.

Actually there is no way to encrypt only a few scripts. Using `'#CRY:` or `'#CRY:1` will always encrypt ALL SCRIPTS in a compiled package.

In the same way, if there is a `'#CRY:0` in the code, this will switch OFF Decryption for ALL Scripts.

If there are multiple such statements found, the LAST statement will be the one that counts.

Example:

```
'----------------------------------------
' Test if your Script is crypted
' compile this and run the EXE-File.
' then
'----------------------------------------
'#SPI:ForceWrite
'#EXE:
'#CRY:
'
EXO.?path\
MBX.!
END.
```

Parameter Explanation:

-

Remarks:

Preprocessor-Directives start with the Sequence: **#**
Because this is only important for the Preprocessor, the running script ignores these commands at ***Runtime*** *(at time of script execution)*.

Limitations:

How SAFE is the crypted Script-File and contained Passwords etc.?

About giving the file to somebody and say "please decrypt it" is mathematically impossible as we see it.
We have used protected ASM code therefore this is the maximum Level of safety.

However we see the following possible Safety problems:

1. Decryption is possible only when using the built in decryption subprograms that are contained somewhere in the Robot's internal code.
The only ways a potential Hacker could get the content of the Script-code is:

© 2023 Theo Gottwald * https://www.smart-package.com

a) To hack a program to get to its internal data is theoretically possible with any program out there,
And the SPR- even though there is some protection - can not make an exception to this.

b) If somebody can get hands on the SPR-Source code (buying the Company :-))
he may be able to decrypt Scripts.

2. If you need to make a compiled Script readable AND if you can proof that YOU are the valid Owner of the Script,
AND if you can pay some days of work-time, WE here at the SPR-Development can also make encrypted Scripts readable again.
For this we would need to compile a special version of the SPR which is actually nowhere available.

These are the the only ways we see, to get hands on the Script-Code. None of these will be easy or cheap.
We believe that, any statistical attack on the encrypted code will fail.

See also:

- 2.1.8 '#INC: - Include Files and Folders into the Executable 290
- 1.3 '#SPI: - Special Instructions 287
- Specialfolders 345
- #ONCE / # OEND - Multiple Include Protection 251
- 1.1 #INC: - Pre-Processor File-Include 248

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.12 2.1.C '#AFT: - RUn AFTER Compile

Compiler-Directives

'#AFT: - Run Executable after Compile

Used to specify the name and path of an executable that is started immediately after Compile. It will become the name of the Executable as Commandline-Parameter.

Intention

```
'#AFT:?path\CopyStuff.exe
```

The `'#AFT:` Compiler Directive is used to start an Executable immediately after the Compilation of an executable is done.

There are two usage-modes:

1. You can start this way the created executable immediately after Compilation, Just write:

```
'#AFT:
```

2. You can start another Executable, in this case the complete path of the just created executable will be added as a commandline-parameter.

You can specify Specialfolders or just a File-Name.

```
'#AFT:Tester.exe
```

This is just the same as:

```
'#AFT:?path\Tester.exe
```

In both cases, the executable will be started like this:

```
(path)\Tester.exe (path and filename of created executable)
```

Below is an working Sample-Script.

Syntax:

```
'#AFT:[P1]
```


© 2023 Theo Gottwald * https://www.smart-package.com

P1 - **N**ame and or Path of the Executable that should be started immediately after Compiling the program.
The program will be started with the Path and filename of the created executable as commandline-parameter.

'#AFT:Me
'#AFT:

In both cases, the executable that was just compiled will be immediately started without any commandline-Parameter.

Example:

```
'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'
' The following line will start an Executable
' "Tester.exe" that must be located in ?path\
' that is the same folder where the Script is.
'
'#EXE:?path\
'#AFT:?path\Tester.exe
'
MBX.Thats me.
END.

'#EXE:?path\
'#SPI:ForceWrite


'----------------------------------------
' TESTER.REM
' Compile this Script to "Tester.exe"
' Then use it with "#AFT:
' and see what happens.
'----------------------------------------
'
' Commandline holen
$$FIL=$cmdexe$
IEF.$$FIL
  MBX. Die Datei $$FIL wurde kompiliert.
ELS.
  MBX. Kann $$FIL nicht finden.
EIF.
ENR.
```

Parameter Explanation:

-

Remarks:

© 2023 Theo Gottwald * https://www.smart-package.com

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

Limitations:

-

See also:

- Specialfolders 345
- #ONCE / # OEND - Multiple Include Protection 251
- 1.6 #INC: - Pre-Processor File-Include 248

*© 2023 Theo Gottwald * https://www.smart-package.com*

2.37.13 2.2.1 '#ICO: - Choose own Icon

Compiler-Directives

'#ICO:

Changing the Icon for the final Executable

Intention

All Commands starting with a "'#" Sequence are Compiler Directives. They are executed at the time of compilation.
That is when you create an executable out of your script-file.

1. You can specify directly an Icon (.ico-File).
'**#ICO:** enables you to specify an ".ico" File for your executable.

Example:

```
'#ICO:?path\smiley.ico
' The following line is functionally exactly the same
'#ICO:smiley.ico
```

2. You can specify that NO ICON at all is used. This will decrease the size of your final executable.

'**#ICO:** enables you to specify that NO ICON shall be used.

Example:

```
'#ICO:-
or
'#ICO:No
```

3. You can specify an Folder with a Set of Icons..

'**#ICF:** enables you to specify an Folder with a set of Icons.
One of these Icons is automatically chosen, if you compile an Executable.
The type of specified "Manifest" decides also which Icon is been chosen.

Thje Icons in this folder must be named like this:

```
  "ST.ico" - This is the Standard Icon, its used if
no Manifest is specified
  "AI.ICO" - "As Invoker" Icon
  "HA.ICO" - "Highest available" Icon
  "RA.ico" - "require Administrator" Icon
  "UM.ico" - "User Manifest" Icon
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

4. You can specify an ".exe" or ".dll" File instead a ".ico" file. In this case the Icons from this file will be extracted to "?path\XIcons"
during compilation. The first available Icon will be used for the executable.
Generally this folder will not be deleted so you can see these Icons after Compilation is done.

```
 '#ICO:?path\Setup.exe
or
 '#ICO:?path\Icons.dll
```

Please note that Icons may be Protected Artwork and may under some circumstances not legally be used for own files.

Syntax:

```
'#ICO:[?path\]filename.ico
'#ICO:[?path\]filename.exe
'#ICO:[?path\]filename.dll
'#ICO:-

'#ICF:Iconfolder-Path\
```

Parameter Explanation:

-

Parameter Explanation:

-

Example:

-

Remarks:
If the Icons is not available, then the default Icon is used instead.
Please note that Icons may be Protected Artwork and may under some circumstances not legally be used for own files.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.14 2.2.2 '#ICF: - Icon from Icon-Folder

Compiler-Directives

'#ICF:

Choosing the Icon for the final Executable from own "Icon Folder"

Intention

All Commands starting with a "'#" Sequence are Compiler Directives. They are executed at the time of compilation.
That is when you create an executable out of your script-file.

You can specify an Folder with a Set of Icons. If any of these Icons is not specified the default Icon is used instead.

'#ICF: enables you to specify an Folder with a set of Icons.
One of these Icons is automatically chosen, if you compile an Executable.
The type of specified "Manifest" decides also which of these Icon is been chosen.

The Icons in this folder must be named like this:

```
"ST.ico" - This is the Standard Icon,
           its used if no Manifest is specified
"AI.ICO" - "As Invoker" Icon
"HA.ICO" - "Highest available" Icon
"RA.ico" - "require Administrator" Icon
"UM.ico" - "User-Manifest" Icon
```

Syntax:

'#ICF:Iconfolder-Path\

Parameter Explanation:
-

Example:
-

Remarks:

*© 2023 Theo Gottwald * https://www.smart-package.com*

If any of these Icons is not specified the default Icon is used instead.
The Icon-Folder can only contain ".ico"-Files.
To extract Icons from ".exe" or ".dll"-files, use the " '#ICO:-Directive and then rename the Icons and use them here.

Limitations:

-

See also:

- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.15 2.2.3 '#ICS:- Icon from internal Icon-Set

Use Icon from internal "Icon-Set" Previous 253 Top 41 Next 361

Compiler-Directives

'#ICS:

There is a bunch of Icons already available inside the SPR. You can choose one of these Icons using the '#ICS:-Direktive.

Intention

All Commands starting with a "'#" Sequence are Compiler Directives. They are executed at the time of compilation. That is when you create an executable out of your script-file.

You can specify an Folder with a Set of Icons. If any of these Icons is not specified the default Icon is used instead.

'#ICS: enables you to specify an Folder with a set of Icons.
One of these Icons is automatically chosen, if you compile an Executable.
The type of specified "Manifest" decides also which of these Icon is been chosen.

The Icons in this folder must be named like this:

```
"ST.ico" - This is the Standard Icon,
           its used if no Manifest is specified
"AI.ICO" - "As Invoker" Icon
"HA.ICO" - "Highest available" Icon
"RA.ico" - "require Administrator" Icon
"UM.ico" - "User-Manifest" Icon
```

Syntax:

'#ICS:[Number][|N]

[Number] - a number in the range of 10 to the maximum number of available Icon-Sets. Currently this maximum Number is 66.
Numbers from 1 to 10 are reserved and repeat later Icons (doubles).
[|N] - "No Text". If this is specified, then always the "ST.ico" Icon is used, there will
be no letters on the Icon like "HA","UM" etc. No Matter what Manifest is used!

Parameter Explanation:

*© 2023 Theo Gottwald * https://www.smart-package.com*

The number of supplied icons may vary.
If you specify an invalid number, the standard Icon will be used.

Example:

```
    '#ICS:10|N
'#ICS:34
```

Remarks:
If any of these Icons is not specified the default Icon is used instead.
The Icon-Folder can only contain ".ico"-Files.
To extract Icons from ".exe" or ".dll"-files, use the " `'#ICO:`-Directive and then rename the Icons and use them here.

The following Icons can be accesse via the `'#ICS:`

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.16 2.3.0. '#MAN: - Manifest for Executable

'#MAN: - Specify a Manifest for the Executable Previous 277 Top 41 Next 287

Compiler-Directives

'#MAN:

Used to specify a manifest type for the executable that is to be created

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

'#MAN:

The '#MAN: Compiler Directive is used to change the Manifest-Type of the final Standalone EXE-File. You can choose between:

- *asInvoker*
- *requireAdministrator*
- *highestavailable*
- *none*

alternative you can specify an own ".xml"-File with Path:

```
'#MAN:?path\User-Manifest.xml
```

For example:

```
'
' choose Manifest that shows that admin rights are required:
'#MAN:ra
or
'#MAN:requireAdmin
or
'#MAN:User-Manifest.xml
```

If you omit the `'#MAN:` - directive, per default "*asInvoker*" is used for the compiled Package. The internally used `MR01.exe` (Robot-Executable) does not have an own Manifest therefore.

Syntax:

'#MAN:[P1]

P1 - (***optional***) Manifest Type can be one of these:

© 2023 Theo Gottwald * https://www.smart-package.com

```
- ai,asInvoker
- ra,requireAdministrator
- ha,highestavailable
- -,none
```

or you can specify an ".xml"-Manifest File with complete path.
If you just specify the filename, ?path\ is taken as path.

```
'#MAN:User-Manifest.xml
```

Example:

```
'----------------------------------------
' Compiler Manifest
'----------------------------------------
'
'
' "requireAdmin" Manifest is used
'#MAN:ra
' or
'#MAN:requireAdmin

' No Manifest will be used
'#MAN:-
' or
'#MAN:None

' "highestavailable" Manifest is used
'#MAN:ha
' or
'#MAN:ra
'
'#MAN:User-Manifest.xml


END.
```

Manifest Explanation:

This is the Manifest that is basically used for the SPR.
The brown colored lines are internally changed to the value that is needed for "required execution level" and "UIAccess".

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1" xmlns:a

<assemblyIdentity type="win32" name="{decriptioA}" version="1.0.0.0" processorAr

<description>{decriptioB}</description>

<dependency> <dependentAssembly> <assemblyIdentity
type="win32" name="Microsoft.Windows.Common-Controls" version="6.0.0.0"
processorArchitecture="x86" publicKeyToken="6595b64144ccf1df" language="*"  />
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
</dependentAssembly> </dependency> <trustInfo xmlns="urn:schemas-microsoft-com:a

<security>
<requestedPrivileges> <requestedExecutionLevel

' The following two lines differ due to the Manifest settings
level="{levelio}"
uiAccess="{uiacio}"
  />

</requestedPrivileges> </security> </trustInfo>
<compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
<application>

<!-- Windows 10 -->
<supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}"/>
<!-- Windows 8.1 -->
<supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/>
<!-- Windows Vista -->
<supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}"/>
<!-- Windows 7 -->
<supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/>
<!-- Windows 8 -->
<supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/>
</application>
</compatibility>

<asmv3:application>
<asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSetti
<dpiAware>true</dpiAware>
</asmv3:windowsSettings>
</asmv3:application>
</assembly>
```

Remarks:

The Icon of the Executable will show the Letter of the used Manifest, A, H or R.
In case you specify a "User-Manifest.xml", the Icon will show "U".

Internally this statement can be emulated using the '#CIL:-Directive:

```
'#CIL:Rempara|Manifest|(Number 0 to 3)
```

and in case a User-Manifest is used:

```
'#CIL:Rempara|Manipfad|(path)
```

Limitations:

-

See also:

- 2.3.1. '#MAU: - UI-Access for Manifest for Executable 325
- 2.4.0. '#VER: - Version-Informations 328

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.17 2.3.1. '#MAU: - UI-Access for Manifest for Executable

Compiler-Directives

'#MAU:

Used to specify a manifest type for the executable that is to be created

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

'#MAU:

The '#MAU: Compiler Directive is used to change the Manifest-UIAccess of the final Standalone EXE-File.

It's needed for "assistive technology application" (for example helpdesk-applications).

You can choose between:

```
- 0 (default) - UI-Access is switched Off, switched to
"false"
- 1 - UI-Access is switched On, set to "true"
```

If you omit the Directive, the default value is : 0

For example:

```
'
' choose Manifest that shows that admin rights are required:
'#MAU:0
or
'#MAU:1
```

In the picture below you can see, how it looks in the final Manifest.

```
23  <security>
24  <requestedPrivileges> <requestedExecutionLevel
25  level="asInvoker"
26  uiAccess="false" ... />
27  </requestedPrivileges>
28  </security>
```

Important Note:
If you specify a User-Manifest, this directive will not change the manifest.
It will only work with the default- built in Manifest's.

© 2023 Theo Gottwald * https://www.smart-package.com

If you omit the '#MAU: - directive, per default "0" is used for the compiled Package.
For more Infos on the uiAccess-Parameter see here:

https://msdn.microsoft.com/en-us/library/windows/desktop/ee671610(v=vs.85).aspx

Syntax:

'#MAU:[P1]

P1 - Can only be 0 or 1, anything else will evaluate to zero.

Example:

```
'----------------------------------------
' Compiler Manifest
'----------------------------------------
'
'
' Switch UIAccess off
'#MAU:0

' Switch UIAccess on
'#MAU:1
'

END.
```

Manifest Explanation:

This is the Manifest that is basically used for the SPR.
The brown colored lines are internally changed to the value that is needed for "required execution level" and "UIAccess".

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1" xmlns:a

<assemblyIdentity type="win32" name="{decriptioA}" version="1.0.0.0" processorAr

<description>{decriptioB}</description>

<dependency> <dependentAssembly> <assemblyIdentity
type="win32" name="Microsoft.Windows.Common-Controls" version="6.0.0.0"
processorArchitecture="x86" publicKeyToken="6595b64144ccf1df" language="*"  />

</dependentAssembly> </dependency> <trustInfo xmlns="urn:schemas-microsoft-com:a

<security>
<requestedPrivileges> <requestedExecutionLevel

' The following two lines differ due to the Manifest settings
level="{levelio}"
uiAccess="{uiacio}" />
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
</requestedPrivileges> </security> </trustInfo>
<compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
<application>

<!-- Windows 10 -->
<supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}"/>
<!-- Windows 8.1 -->
<supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/>
<!-- Windows Vista -->
<supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}"/>
<!-- Windows 7 -->
<supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/>
<!-- Windows 8 -->
<supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/>
</application>
</compatibility>

<asmv3:application>
<asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSetti
<dpiAware>true</dpiAware>
</asmv3:windowsSettings>
</asmv3:application>
</assembly>
```

Remarks:

1. If needed, SPR-compiled EXE-Files can now be digitally signed with a certificate.

2. This command will internally evaluate to

```
'#CIL:Rempara|Manifest_UIAccess|0
'#CIL:Rempara|Manifest_UIAccess|1
```

Limitations:

-

See also:

- 2.3.0. '#MAN: - Manifest for Executable

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.18 2.4.0. '#VER: - Version-Informations

'#VER: - Specify custom Version-Informations Previous 277 Top 41 Next 287

Compiler-Directives

'#VER:

Used to specify custom version Information for the executable that is to be created.

Intention

Compiler-Directives start with the Sequence:

'#

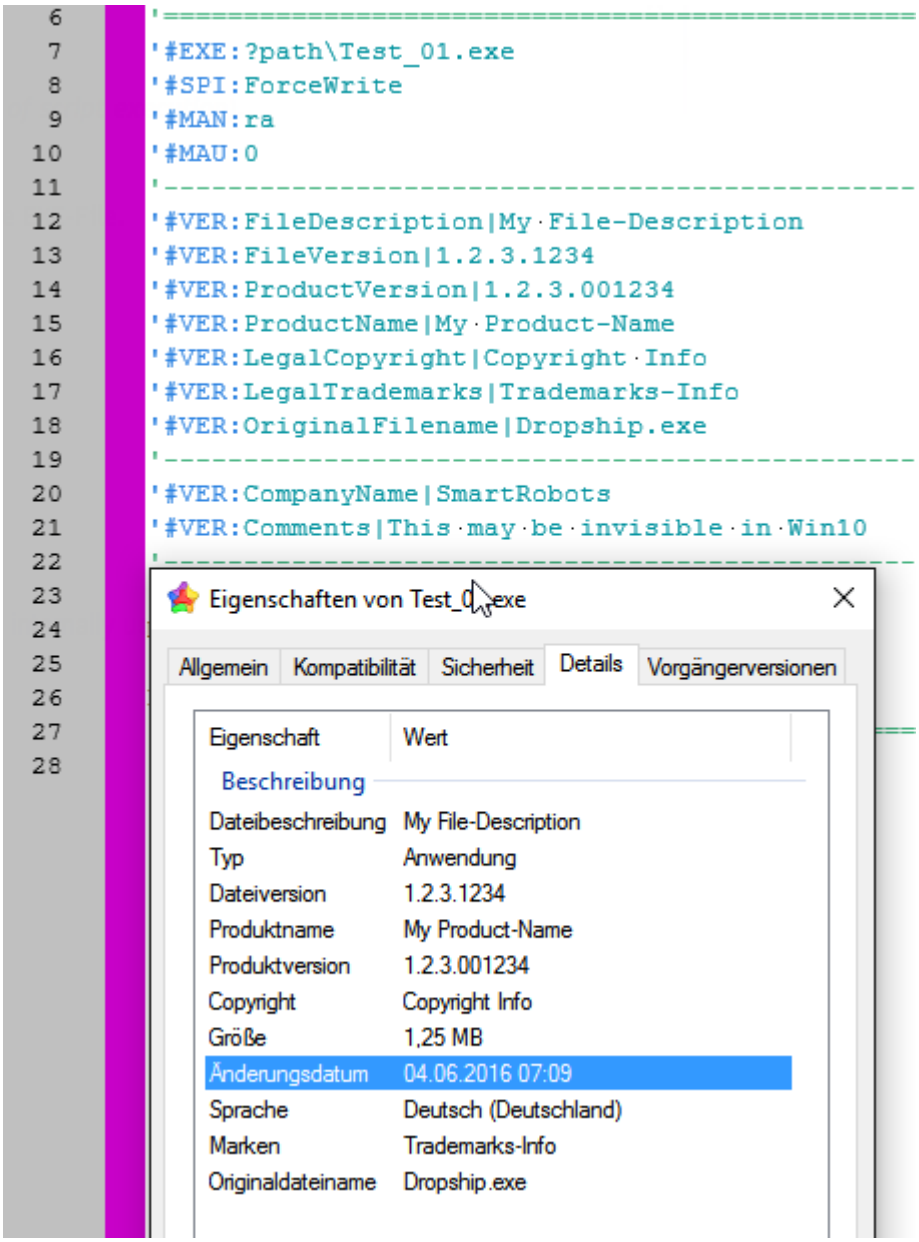
Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution)*.

'#VER:

The '#VER: Compiler Directive is used to change the Version Information of the final Standalone EXE-File.
Below is an example.

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax:

'#VER:P1|P2

P1 - One of the following Version-Information:

- *FileDescription*
- *FileVersion*
- *ProductName*
- *ProductVersion*
- *LegalCopyright*
- *LegalTrademarks*
- *OriginalFilename*

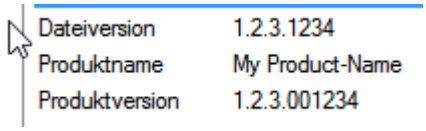
© 2023 Theo Gottwald * https://www.smart-package.com

- *CompanyName*
- *Comments*

P2 - Value for the Version-Information.
For *FileVersion and ProductVersion* there is a special rule: use only numbers and use three ".". For example:

```
'#VER:FileVersion|1.0.0.12
'#VER:ProductVersion|1.0.0.120
```

Also note, that the internal formatting of the last number is not equal between these two parameters.



Example:

```
'----------------------------------------
' Changed Version Informations
'----------------------------------------
'#VER:FileDescription|My File-Description
'#VER:FileVersion|1.2.3.1234
'#VER:ProductVersion|1.2.3.001234
'#VER:ProductName|My Product-Name
'#VER:LegalCopyright|Copyright Info
'#VER:LegalTrademarks|Trademarks-Info
'#VER:OriginalFilename|Dropship.exe
'-----------------------------------------------
'#VER:CompanyName|SmartRobots
'#VER:Comments|This may be invisible in Win10

END.
```

Parameter Explanation:

-

Remarks:

Internally uses:
`'#CIL:VersioninfoA|(P1)=(P2)`
There is also:
`'#CIL:VersioninfoB|(P1)=(P2)`
which is for an english set of File-Descriptions.

Limitations:

These two may not be visible under Windows 10.

```
'#VER:CompanyName|SmartRobots
'#VER:Comments|This may be invisible in Win10
```


© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 2.3.0. '#MAN: - Manifest for Executable 321

© 2023 Theo Gottwald * https://www.smart-package.com

2.37.19 2.5.0. '#CIL: - Direct Compiler Interface

'#CIL: - Specify custom Options for the Compiler Previous 277 Top 41 Next 287

Compiler-Directives

'#CIL:

Used to specify custom options for the executable that is to be created.

Intention

Compiler-Directives start with the Sequence:

'#

Which is a **comment-sign** on the left and then the **#.**

Because of the comment sign, the running script ignores these commands at ***Runtime*** *(at time of script execution).*

'#CIL:

The '#CIL: Compiler Directive is used to directly write Options to the Compiler. It's not needed for the "normal user". It enables advanced access to the Compiler-Internals.

For example:

'

If you omit the `'#MAN:` - directive, per default "*asInvoker*" is used for the compiled Package. The internally used `MR01.exe` (Robot-Executable) does not have an own Manifest therefore.

Syntax:

'#CIL:[P1]|[P2]|[P3]

P1 - Section: These can be one of the following values:

- `Rempara`
- `RemRun`
- `Remcust`
- `Resourcen`
- `VersioninfoA`
- `VersioninfoB`

P2 - Key: These must be the key-value. These values represent internal variables in the Compiler-/Compilation Process.

P3 - Value: This is the value that is been assigned to the Key **(P2).**

© 2023 Theo Gottwald * https://www.smart-package.com

Example:

```
'----------------------------------------
' Compiler Manifest
'

' This will set Version-info similar to "'#VER:" directive
'#CIL:VersioninfoA|ProductVersion|1.2.3.4

' This is equivalent to the '#ICO:
'#CIL:Rempara|Icon-Path=?path\Face1.ico
'#CIL:Rempara|Icon-Path=?path\Restuner.exe

END.
```

Parameter Explanation:

-

Remarks:
Please use this command only, if you know what you are doing.
Inputs from this command are in many cases not checked for validity and can prevent the script from compiling as it should..

Limitations:
-

See also:

- 2.0.0 Smarty's EXE-Compiler 267

 © 2023 Theo Gottwald * https://www.smart-package.com

2.38 3. System-Variable and Specialfolders

2.38.1 3.1 Systemvariables

Systemvariables Previous 389 Top 41 Next 341

MiniRobotLanguage (MRL)

Systemvariables

Intention

Systemvariables are replaced with the appropriate Content at runtime. They need to be written in lowercase.

Parameter Explanation

SYSVAR		Usage		ASCII
$crlf$	-	CR+LF	-	(13,10)
cr	-	Carriage Return	-	(13)
lf	-	Line Feed	-	(10)
sp	-	Space	-	(32)
tab	-	Horizontal Tab	-	(9)
nul	-	Null	-	(0)
sq	-	Singel Quote	-	(39)
dq	-	Double Quote	-	(34)
esc	-	Escape	-	(27)
eof	-	End-of-File	-	(26)
ff	-	Form Feed	-	(255)
vt	-	Vertical Tab	-	(11)
bs	-	Back Space	-	(8)
bel	-	Bell	-	(7)

Robot Command-Line

$cmdl$ - custom commandline (in EXE-Mode use $cmdexe$)

(Text-)Clipboard Content

cpb - Clipboard-Content (Text)

Current running Script

#linenum# - aktuelle Zeilennumber im Script
$script$ - actual Filename, currently running script.

© 2023 Theo Gottwald * https://www.smart-package.com

```
                Without Path.
#errnum#  - Code Number for Type of Error:
            0 - Unspecified Error. Details see Error-Message.
            1 - Syntax-Error,
            2 - Parameter-Nzmber,
            4 - Size-Error (to much or to few of something)

#errlin#   -    Linenumber where the last error has occurred.
$errtxt$   -    Error-Message, identical to what is shown in
the Messagebox.
                This one is available in Silent-Mode, where no
Messagebox appears.
```

Variables for use with the Stack's

```
$tos$   -       insert Top-of-stack
$dtos$  -       insert Top-of-stack (nodelete)
#tos#   -       Number of items on current stack
#stack# -       returns  1 if  GLOBAL Stack is selected
                         2 -if  LOCAL Stack is selected
```

Variables for time-measurement

```
$now$   -       (Local time) Exact Time,
                Format: 2006:01:30:16:29:17:0250
$utime$ -       (UTC/Universal Coordinated Time),
                Format: 2006:01:30:16:29:17:0250
$date$  -       Actual Date, Example:    01-01-2006
$time$  -       Actual Time, Example:    12:01:40
#date#  -       Date, Format: 01302006 for 30/01/2006
#time#  -       Time, Format: 173226 for 17:32:26
#timer# -       Timer, Format: 63220.625
#datetime#      Date and Time, Format: 01302006_172430
#dtime# -       Time, Format: 3348146124.41
#dsince#-       Time, since last call to #dtime# or #dsince#
#since# -       Time since last call to #since# or #time#
```

Process- and ThreadID and Window Handles

```
#pid#   -       Process-ID of the last program or window. Not
all execute-commands provide this.
                See help for EXS and other EXE-commands. Also
set by STW. and some other such Window-Commands.

#tid#   -       Thread-ID of the last program or window. Not
all execute-commands provide this.
                See help for EXS and other EXE-commands. Also
set by STW. and some other such Window-Commands.
                Prefer using the PID, as the TID is often not
set.

#cbw#   -       Handle of own Window
#cwh#   -       Debug Console Window-Handle
#hwn#   -       Handle, actual located window
```


© 2023 Theo Gottwald * https://www.smart-package.com

InterKOM

```
Note: the term "CUN" (Communication Unique Number) is the same
as
      the "InterKOM-Handle".

#cun#    -     CUN of the actual Robot, own InterKOK-Handle.
#ecun#   -     CUN of the actual Editor, if any
#fcun#   -     CUN of the Father-Process ("Father")
#bcun#   -     CUN of the last started Mini-Browser,
               it must have been started using the LBR.
Command.
#pcun#   -     CUN of the last started Parallel-Robot,
               using the PRR. Command.
```

Mini-Browser only

Note: these Systemvariables are only resolved if an Mini-Browser (MB**)** has been started from this robot, using the LBR.-command.
If you have multiple TAB's open in the same MB, some of these System-Variables react only on the actual selected TAB, while others are been set by any TAB that is loading at the same time.
You can use Parallel-Robots and multiple Instances of the MB to work with several Sites at the same time to overcome possible confusion when dealing with multiple sites at the same time.

```
#mb_prog#     -  0-100%, Loading Progress (Site-dependend)

#mb_state#    - 1/0 if actual selected TAB Browser Control
                is busy. 0 - ready, 1 - busy loading
                Result 99 if no state is available.
#mb_busy#     - 1/0 if MB overall is actually busy working
                1 - busy, 0 - ready, waiting for command
```

Generally the Browser-States occur one after the other when you load a WEB-Site. It may look like this for example:

```
[13:00:33] [12]  TAB: 0*269*SetSecureLockIcon*0*
[13:00:33] [12]  TAB: 0*102*StatusTextChange*Fertig*
[13:00:33] [12]  TAB: 0*108*ProgressChange*-1,10000*
[13:00:34] [12]  TAB: 0*104*DownloadComplete**
```

Using these System-Variables, you can either get them "One after the other" or you can only get the state that is "actual", at the time when you use the System-Variable.
Therefore you have two complete sets of system Variables.
One set wit "s" like "Serial" to get all states in order.
One set with "a" like "actual", to get only the state that is NOW.

© 2023 Theo Gottwald * https://www.smart-package.com

Note: the following SysVars contain all Browser Messages in serial order. To have them switch to the next message, use the #mb_nex# at any place.

#mb_time# - Time of last call to LBR., CBR.goto etc..
This is the time when you commanded to load a site in the MB.

#mb_since# - DTime since last call to LBR., CBR.goto.
Use to measure how long a site takes to load.

mb_sli - all Informations about the serial Event, in one System-Variable, separated by "*". Example:
TAB: 0*108*ProgressChange*-1,10000*
(Tab-Number)*(Event-Code)*(Event Name)*Additional Info)

#mb_stnr# - Number of the selected TAB for the serial message number starting from 0.

#mb_sevn# - Numeric Event Number for the serial Browser Event.

mb_sai - Additional Information for the serial Browser Event.
mb_stxt - State-Text Information for the serial Browser Event.

#mb_nex# - Use this to switch all "Serial MB-System-Variables" to the next serial Event. It will not evaluate to any value.

Note: the following SysVars contain all Browser Messages in actual order. This is the state of the Browser right when you use the SysVar. Using these Variables, states can be lost, if they are not more actual. You do not need to use #mb_nex# with these, because you get the actual state.

#mb_atnr# - Number of the selected TAB for the actual message number starting from 0.

#mb_aevn# - Numeric Event Number for the actual Browser Event.
mb_aai - Additional Information for the actual Browser Event.
mb_atxt - State-Text Information for the actual Browser Event.

mb_ali - all Informations about the actual Event, in one System-Variable, separated by "*". Example:
TAB: 0*108*ProgressChange*-1,10000*
(Tab-Number)*(Event-Code)*(Event Name)*Additional Info)

Note: All Errors will stay in these System-Variables until they are confirmed using the #mb_cerr# System-Variable. If there occur multiple errors, only the last error is stored. The last error
overwrites a previous error, whether it was confirmed with #mb_cerr# or not.

© 2023 Theo Gottwald * https://www.smart-package.com

```
#mb_err# - 0 or last MB-Error number from Mini-Browser.
           Could be for example: 2148270085
$mb_err$ - A short text that identifies the Error.
           For example: "INET_E_RESOURCE_NOT_FOUND"
$mb_ert$ - A textual description of the error.
           For example: "The server or proxy was not found."

#mb_cerr# - Will confirm the last Error and Reset the mb_er*
Systemvariables.

        A list with possible Errors can be found here:
        Internet Explorer Errors and here: HTTP-Status Codes
```

Mouse-Position

```
#mpx#    -       actual Mouse position X
#mpy#    -       actual Mouse position Y
#opx#    -       actual Mouse position X - Client-Coords.
#opy#    -       actual Mouse position Y - Client-Coords.
#ixp#,#iyp# -    Actual Point x,y (screen Coords)

#il#     -       actual Element, left side X (Sreenccords)
#it#     -       actual Element, upper side Y (Sreenccords)
#ir#     -       actual Element, right side X (Sreenccords)
#ib#     -       actual Element, bottom side y (Sreenccords)
#iw#     -       actual Element, width X
#ih#     -       actual Element, height Y
#icx#    -       actual Element, Centerpoint X
#icx#    -       actual Element, Centerpoint Y
```

Timeout-Flag

```
#tio#    -       Timeout-Flag
#til#    -       Timeout-Line-Number
```

Screen-/ Desktop Sizes

```
#ssx#,#ssy# -    Screen size X,Y
#wsx#,#wsy# -    Work-area-Size X,Y
#sbp#    -       Screen-bits per pixel (Colordepth)
```

FEX-Enumeration

```
#fex#    -       FEX - Elements left on the FEX-Stack
#fxd#    -       FEX - Nesting-Depth
```

There are some more for **compiled EXE's**

```
$exename$     - Name of the compiled EXE (Filename)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$starttime$    - time when EXE was started
$startdate$    - date when exe was started
$cmdexe$       - commandline given to the compiled EXE
$startscript$ - Name of script-file to start first
#exemode#      - if Script runs inside an EXE (1) or not (0)
#mutex1#       - was Script started from the same EXE before?
#mutex2#       - was a script of this name started from
                 anywhere before?

Hint: Timing functions like #time# and others internally use
                 the
      windows timer. This timer will be set to zero normally
                 around
      midnight. If your measurement will fall just into the
                 moment
      of the timer reset, the results may be wrong.
```

Example

```
'****************************
' Example 1
'****************************
' Print items on Stack
PRT.Items on Stack: #tos#

' Time Measurement
VAR.$$TIA=#dtime#
CAS.1
VAR.$$TIB=#dsince#
PRT. Time elapsed: $$TIB

'****************************
' Example 2
'****************************
EXH..html|§§HWN|http://www.fa2.de
STW.h|§§HWN
SWP.0,0|400,400
SAO.nvw|google|google|30
MPO.
PRT. We start at: #dtime#
PRT. Process-ID of our actual window is: #pid#.
PRT. The Window-handle is #hwn#.
PRT. The actual Mouse-Position (Screen) is: #mpx#,#mpy#
PRT. The actual Mouse-Position (Window) is: #opx#,#opy#
PRT. The actual Screensize is: #ssx#,#ssy#
PRT. The actual Workarea-Size is: #wsx#,#wsy#
PRT. The actual point is also at: #ixp#,#iyp#
PRT. Our Screen-colourdepth is: #sbp#
PRT. These Printouts took: #dsince# ms.
DMP.
END.

'****************************
' Example 3
' Mini-Browser
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'****************************
LBR.svf=0|Pos=Mid|site=www.fa2.de
DOL.1
  DBP. Progress: #mb_prog#
  DBP.State: #mb_busy# - #mb_state#
OOP.(#mb_busy#=0)
MBX.Ready
ENR.

'****************************
' Example 4
' Mini-Browser
'****************************
LBR.svf=0|Pos=Mid|site=www.fa2.de|site=www.fa33.de
FOR.$$CNT|1|300
  DBP. $mb_sli$ #mb_nex#
NEX.
MBX.Ready
ENR.

'****************************
' Example 5
' Mini-Browser - Site loading.
'****************************
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
WBR.
DBP. Site loaded in #mb_since# seconds (Start-Time:
#mb_time#).
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.3 Specialfolders 345
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

2.38.2 3.2 Standard-Search Pattern

MiniRobotLanguage (MRL)

Standard Search Pattern

Intention

Generally spoken, its mostly never needed to use the whole string for a text search. In most cases and with nearly all commands, a unique substring is all you need to describe an element.

Additionally you can use search patterns to describe elements. These patterns can be used in most commands, where you search for Text. Using the patterns, you can make your scripts recognize controls from multiple languages.

For example, when you open a WEB-Site and you want to check if the page is really available, you could use the following vommand:

```
' Wait 15 seconds until whether the term "Description" or
' the term "Media not visible" appears.
SAO.rnw|42|Description&OR:Media not visible|15
```

Parameter Explanation

Search Pattern	Shortform	Example
{&DOCAPS:}	{&00:}	{&DOCAPS:ExPE}
{&OR:}	{&01:}	{Text1&OR:Text2}
{&XOR:}	{&02:}	{Text1&XOR:Text2}
{&AND:}	{&03:}	{Text1&AND:Text2}
{&ANDTHEN:}	{&04:}	{Text1&ANDTHEN:Text2}
{&FUZZY:}	{&05:}	{{Text1}&FUZZY:[value]}
{&TOLERANT:}	{&06:}	{{Text1}&TOLERANT:[value]}
{&NOT:}	{&07:}	{&NOT:{Text1}}
{&EXACT:}	{&08:}	{&EXACT:Text1}
{&NOTEXT:}	{&09:}	{&NOTEXT:}
{&WILDC:}	{&0c:}	{&WILDC:Hallo??.*}

*© 2023 Theo Gottwald * https://www.smart-package.com*

{&t:}		
{&f:}		

{&DOCAPS:} = Activate Case-Sensitive

Normally all search patterns are not case sensitive. If you need case sensitivity, you can activate, the {**&DOCAPS:**}.

Text is: „Andreas is a human".

Searching {**&DOCAPS:**andreas} will not find the text.
Searching {**&DOCAPS:**Andreas} will find the text.

{&OR:} = OR

To perform a logical 'OR' of two options.
If either are true, then the result is true, otherwise it is false.
This is most often used to accept buttons in different languages.

Example:

AFT.**{yes&OR:ja&OR:Oui}**

will find any such element no matter if its english, german or french. There is no limit on how many values are used.

{&XOR:} = XOR

To perform a logical 'OR' of two values.
Only one can be true to get a true result. When both are true, the "XOR" result is false.

{&AND:} = AND

All the Words (conditions) must be true.

{&ANDTHEN:}

The Words must be in sequence one after the other with any number of chars between.

{&FUZZY:}

The Value must be given in numbers. 0 ... 100. It enables the found string to be slightly different from the original source. {&FUZZY:} calculates the "Edit-Distance" which tells how many wrong key-press would lead to this result.

For example:

If your window has the Title:
"Walkenhorsts Letter.doc - Microsoft Word"

© 2023 Theo Gottwald * https://www.smart-package.com

and you are using Matchstring: "{Microsaft&FUZZY:1}" then the result is true, because "Microsaft" is matching by 1 wrong character to "Microsoft".

{&TOLERANT:}

The Value must be given in 0% to 100%. It enables the found string to be slightly different from the original source. It uses a percentage Scale (0-100%). It usese percentage in different to Fuzzy.

{&NOT:}

The negotiation is true.
For example: The white light is not blue = TRUE

{&EXACT:}

When an expression should be exact the same you are looking for.
It is by default not case-sensitive.
For that you need to apply {&DOCAPS:} as well if needed.

{&EXACT:} will match the text only to true, if the text is equal.
This is important (for example) if you want to find a window "C:\Programme" while there is another one "C:\Programme\something".
Without {&EXACT:} you could match the wrong one.

{&NOTEXT:}

When you are looking for a window it has normally a name.
Some windows dont have names, so u can test it in your search by using the “&NOTEXT” pattern.

{&DOCAPS:&EXACT:Hallo&OR:{&notext:}}

{&WILDC:}

This arconym stands for “wildcard” and there are some symbols for different comparings between a source and a target string.

These are the expression_pattern:

? Use this if you don`t know any single character (all but no digit)
* Use this if you don`t know any characters, it can be zero or more.
# Use this if you don`t know any single digit (0-9)
[charlist] For complex search algorythem. Any single character in 'charlist'
[!charlist] Any single character not in 'charlist'

{&WILDC:Andrea*M[üä]ller*}" match "Andreas Müller

© 2023 Theo Gottwald * https://www.smart-package.com

{&t:}

Use this to tell „true" to your command, because its always true.
{&f:}

Use this to tell „false" to your command, because its always false.

Important: Standard-Search Pattern-terms must be written in UPPERCASE letters. Like:

&WILDC:

This i wrong:

&wildc:

This will also not work:

&WildC:

These patterns will only be recognized if they are written in big letters (uppercase).

Example:

```
{&DOCAPS:TheoGottwald&ANDTHEN:Internet}&OR:{&EXACT:Explorer}}

{&WILDC:Andrea*}" match "Andreas Müller

SCI.{&DOCAPS:ExPE}

STW.ct|#32770|Microsoft
&ANDTHEN:Word&ANDTHEN:2.29&ANDTHEN:.doc

STW.ct|afx:|Microsoft Word V.2.29 vom 23-01-97
&ANDTHEN:liste.doc

STW.ct|#32770|{Mikrosx&TOLERANT:4}&ANDTHEN{&NOT:word}
&ANDTHEN{&EXCEL&OR:Frontpage}

STW.ct|#32770|{Microsaft&TOLERANT:60}
```


© 2023 Theo Gottwald * https://www.smart-package.com

2.38.3 3.3 Specialfolders

MiniRobotLanguage (MRL)

Specialfolders

Intention

Specialfolders are also systemvariables.

- They are replaced on runtime with a real, physical folder-name.
- This folder name is evaluated at runtime on the target machine.

Specialfolders need to be written in small letters. Specialfolders need the "\" at the end of the name to be complete.

Parameter Explanation

```
Currently we support these OS-Specialfolders:

?wi\              - current windows-folder
?ws\              - windows\system32 - folder
?desktop\         - current user-desktop
?cdesktop\        - common desktop
?auto\            - current user-startmenu/autostart
?cauto\           - common startmenu/autostart
?fonts\           - (system) fonts folder
?startmen\        - current user startmenu
?cstarmen\        - common startmenu
?sendto\          - current user SendTo Folder
?progs\           - c:\programm files (x86) etc.
?temp\            - current user temporary folder
?startprog\       - current user\Startmenu\Programme\
?cstartprog\      - common \Startmenu\Programme\
?sysfolder(xx)    - get Specialfolder by code (see code below)
?appdata\         - C:\Documents and Settings\<username>\Application
Data\

And these Robot-specific folders:

?curdir\        - The current directory path
?path\          - The path to the Script itself
?pfad\          - The path to the Script itself (german)
?pattern\       - "?path\Pattern\" this is the default Path for
Search-Patterns in the Scriptfile.
```


© 2023 Theo Gottwald * https://www.smart-package.com

use '#INC:?path\Pattern* for including the Pattern into an Executable.
?exepath - The path to the Main Robot-Executable
?exepfad - The path to the Main Robot-Executable (german)
?mainpath - The path to the main-script file
?lib - **Path to the Library, see #LIB:**
This path is different depending
if in Editor or in EXE-Mode.
In EXE-Mode this Path will always be:
?path\LibXXSPR
From the Editor the Path will be what was specified with the #LIB: - Directive.

and only for compiled EXE's

?exeloc - Path of compiled EXE-Package

Under x64 OS (Vista, Win 7 and higher) you have:

?progs64 - c:\programm files\
?ws64 - c:\windows\System32\
?ws - c:\windows\SysWOW64\

And here are the Codes for ?sysfolder(xx):
Prefix with &H (hexadezimal).
Example: ?sysfolder(&H0A)\ -> Recycle Bin

00 - <desktop>
01 - Internet Explorer (icon on desktop)
02 - Start Menu\Programs
03 - My Computer\Control Panel
04 - My Computer\Printers
05 - My Documents
06 - <user name>\Favorites
07 - Start Menu\Programs\Startup
08 - <user name>\Recent
09 - <user name>\SendTo
0a - <desktop>\Recycle Bin
0b - <user name>\Start Menu
0d - "My Music" folder
0e - "My Videos" folder
10 - <user name>\Desktop
11 - My Computer
12 - Network Neighborhood
13 - <user name>\nethood
14 - windows\fonts
15 - Templates
16 - All Users\Start Menu
17 - All Users\Programs
18 - All Users\Startup
19 - All Users\Desktop
1a - <user name>\Application Data
1b - <user name>\PrintHood
1c - <user name>\Local Settings\Applicaiton Data (non roaming)
1d - non localized startup

© 2023 Theo Gottwald * https://www.smart-package.com

```
1e - non localized common startup
1f - COMMON FAVORITES
20 - INTERNET CACHE
21 - COOKIES
22 - HISTORY
23 - All Users\Application Data
24 - Windows-Directory
25 - System-Directory
26 - C:\Program Files
27 - C:\Program Files\My Pictures
28 - USERPROFILE
29 - x86 system directory on RISC
2a - x86 C:\Program Files on RISC
2b - C:\Program Files\Common
2c - x86 Program Files\Common on RISC
2d - All Users\Templates
2e - All Users\Documents
2f - All Users\Start Menu\Programs\Administrative Tools
30 - <user name>\Start Menu\Programs\Administrative Tools
31 - Network and Dial-up Connections
35 - All Users\My Music
36 - All Users\My Pictures
37 - All Users\My Video
38 - Resource Direcotry
39 - Localized Resource Direcotry
3a - All Users OEM specific apps
3b - USERPROFILE\Local Settings\Application Data\Microsoft\CD Burning
3c - unused
3d - Computers Near Me (computered from Workgroup membership)
8000 - combine with other numbers to force folder creation
```

Example

```
' See the Printout in the Text-Area of the Editor.
' The path to the Script itself

VAR.$$PFA=?path\
DBP.$$PFA

' User-Profile
VAR.$$PFA=?sysfolder(&H28)\
DBP.$$PFA

VAR.$$PFA=?mainpath\
DBP.$$PFA

VAR.$$PFA=?cstartmen\
DBP.$$PFA

VAR.$$PFA=?progs\
DBP.$$PFA

VAR.$$PFA=?cdesktop\
DBP.$$PFA

VAR.$$PFA=?ws\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.$$PFA

VAR.$$PFA=?wi\
DBP.$$PFA

ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

2.38.4 3.4 System-Colors

MiniRobotLanguage (MRL)

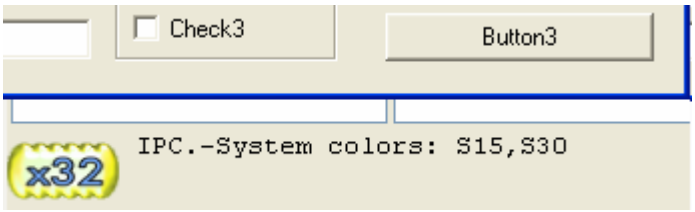
System-Colors

Intention

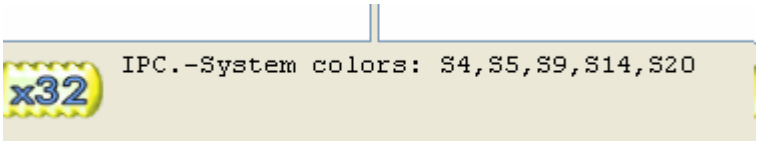
Normal colors values may differ on different target systems. This is because of changes in themes and graphics drivers and hardware.

To be able to use color values, you can use the "Windows System Colors" in many commands where Colors can be used.

System Colors are displayed in the Windows-Inspector, for the point which is currently below the mouse-cursor.



System-Colors may have the same values as other System-Colors on different systems. This is however a question of the design/theme that is used.
However, it is a fact that text is often black, as well as shadows.



You can use the CCV.-command to directly get the values of any System-Color.
Here is an example:

```
' Show value od System color S2
CCV.H|S2|$$HEX
DBP.$$HEX
ENR.
```

Systemcolor Definition (Color Type)

```
S0          - SCROLLBAR COLOR
S1          - BACKGROUND/DESKTOP COLOR
S2          - ACTIVE-CAPTION COLOR
S3          - INACTIVE-CAPTION COLOR
```


© 2023 Theo Gottwald * https://www.smart-package.com

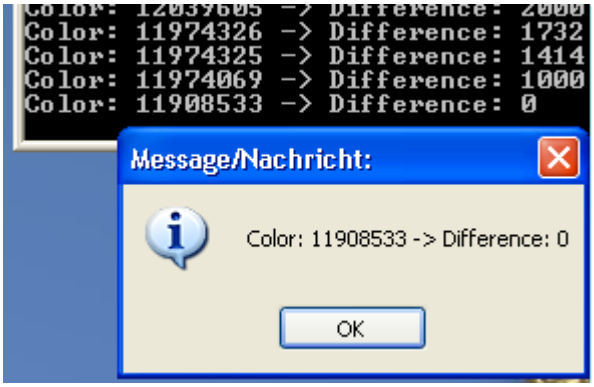
```
S4          - MENU COLOR, MSGBOX COLOR
S5          - WINDOW COLOR
S6          - WINDOW-FRAME COLOR
S7          - MENU-TEXT COLOR, MSGBOXTEXT COLOR
S8          - WINDOW-TEXT COLOR
S9          - CAPTION-TEXT COLOR
S10         - ACTIVEBORDER COLOR
S11         - INACTIVEBORDER COLOR
S12         - APPWORKSPACE COLOR
S13         - HIGHLIGHT COLOR
S14         - HIGHLIGHT-TEXT COLOR
S15         - BUTTON-FACE COLOR
S16         - BUTTON-SHADOW/3DSHADOW COLOR
S17         - GRAY-TEXT COLOR
S18         - BUTTON-TEXT COLOR
S19         - INACTIVE-CAPTION-TEXT COLOR
S20         - BUTTON-HIGHLIGHT COLOR

The following colors will not be available below Windows NT 4:

S21         - 3DDKSHADOW COLOR
S22         - 3DLIGHT COLOR
S23         - INFO-TEXT COLOR
S24         - INFO-BACKGROUND COLOR
S25         - not defined (&HB5B5B5 on XP here)

The following colors will not be available below Windows 2000:

S26         - HOTLIGHT COLOR
S27         - GRADIENT-ACTIVE-CAPTION COLOR
S28         - GRADIENT-INACTIVE-CAPTION COLOR
S29         - MENU-HILIGHT COLOR
S30         - MENUBAR
```

Example

```
'****************************
' Example 1
'****************************
' Brute Force Algo to guess undefined Systemcolor "S25"
' Needs quite a while to run, even as EXE.
'
$$COA=S25
$$COM=&HFFFFFF
FOR.$$COB|&HFFFFFF|0|-1
  CCD.$$COA|$$COB|$$COD
  IVV.$$COM>$$COD
    $$COM=$$COD
    $$COS=$$COB
    PRT.Color: $$COS -> Difference: $$COM
  EIF.
  JIZ.$$COD|out
NEX.
:out
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.Color: $$COS -> Difference: $$COM
ENR.

Here is the output of the program on WinXP:
```



Remarks

When specifying Colors as hexadecimal values, prefer using this format:
`&H0FFFFFF.`
`   RRGGBB`
If you omit the "`&H0`" or just use the "`&H`" this may lead to strange results when the values is converted to decimal. The reason for this is::
`&HFFFF is -1 (conversion to signed number using "&H")`
while
`&H0FFFF is 65535 (conversion to unsigned number using "&H0")`
as expected.

Limitations:

-

See also:

- 3.3 Specialfolders 345
- CCD. - Calculate Color Distance 1109
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

2.38.5 3.5 Hexadezimal, Decimal and Binary and more

Hexadecimal and other numeric formats Previous 349 Top 41 Next 356

MiniRobotLanguage (MRL)

Hexadecimal & numeric formats

Intention

In normal life, we most often use only "decimal" numbers. In some situation it is easier to specify a number in "Hexadecimal" notation.For example when specifying a color-value.

Also as you may know, computers only work with binary data. BInary means it has only the two values 0 and 1 - it is **base 2**. Hexadecimal has all numbers you can see below, it is **base 16**.

It is easier to convert Binary to Hex and vice-verse compared to standard numbers (base 10). By grouping Binary numbers by 4, you can actually convert it directly to Hex, and vice versa. Here are all HEX-numbers and their binary conversion.

```
0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
8 = 1000
9 = 1001
A = 1010
B = 1011
C = 1100
D = 1101
E = 1110
F = 1111

For example:
In short B0D3 (using the above table) converts to binary as
1011 0000 1101 0011

So if you write something like 1101001010010011, and group it
by four 1101 0010 1001 0011. Then use the table and you get
"D293".

The robot will understand Hexadezimal numbers if they are
prefixed with either "&H0" (unsigned numbers). Or "&H" (signed
numbers).

With base 10, you will need to compute it manually (for A0D3
see below)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
A = 10 * 16^3 (or 10 * 16 * 16 * 16) = 40960
0 = 0 * 16^2 (or 0 * 16 * 16) = 0
D = 13 * 16^1 (or 13 * 16) = 208
3 = 3 * 16^0 (or 3 * 1) = 3
```

then just add the results and get the decimal number, in this case `41171`.

You can use the `CCV`.-command to directly get the values of any System-Color.
Here is an example:

```
' Show value od System color S2
CCV.H|S2|$$HEX
DBP.$$HEX
ENR.


The CCV.-Command can also convert numbers into Hexadecimal
Color values:

' Show value od System color S2
$$NUM=12345368
CCV.H|$$NUM|$$HEX
DBP.$$HEX
ENR.
```

Details

You can use these numeric formats at many places in your scripts:

1. Decimal
You can just write a number, if the number has a fractional part, use the "." for example:

```
CAL.$$RES=4.5*9
```

Decimal numbers can be prefixed with a "-" sign to make them have a negative value.

2. Octal
You can use Octal numbers, if you prefix the number with "`&O`". Octal numbers will be converted to signed decimal values. For example:

```
CAL.$$RES=&10*9
' Result is 72 (dec)
```

Octal numbers are signed numbers and may not contain an fractional part.

3. Hexa-Decimal
You can use signed or unsigned Hexadecimal numbers. They may not have an fractional part. Be sure to choose the right Prefix! If you specify

```
&H(number) -> signed number conversion
&H0(number) -> unsigned number conversion
```

For example:

© 2023 Theo Gottwald * https://www.smart-package.com

```
CAL.$$RES=&HFFFF*9
DBP.$$RES
ENR.
' Result is -9

CAL.$$RES=&H0FFFF*9
DBP.$$RES
ENR.
' Result is 589815
```

To convert a decimal number to a Hexadecimal number, use
VTH. - Variable to Hexadecimal 2356

4. Binary
You can specify binary values as well. Prefix them with "&B". No fractional part can be used.
For example:

```
VAN.$NUM=&B01010011
DBP.$$NUM
' Result is 83

CAL.$$RES=&B10*9
DBP.$$RES
' Result is 18
```

To convert a decimal number to a BInary number, use
VTB. - Variable to Binary 2354

Numeric parameters and formulas

Wherever you have numeric (decimal-) parameters, you can also use a formula.
For example:

The command PAU. want a numeric parameter first. We could write:

```
' The script will Pause for 5 seconds
PAU.5
```

We can also use a variable here:

```
' The script will Pause for 5 seconds
: §§TIM=5
PAU.§§TIM
```

Now we can also use a formula here. The trick is, that we must put the formula into brackets (...).

```
' The script will Pause for 5 seconds
: §§TIM=1
PAU.(§§TIM*5)
```


© 2023 Theo Gottwald * https://www.smart-package.com

You can use formulas at most places where numeric parameters are valid. The formula can use all expressions that are valid for the CAL.-command also.

Remarks

Prefer using decimal numbers in formulas to prevent confusion. If you place the numbers inside variables, using the `VAN.` Command they are converted immediately to decimal numbers, which is the best way to go.
If you use the `VAR.` command instead, the numbers will be stored as text inside the variable.

Limitations:

-

See also:

- VTH. - Variable to Hexadecimal 2356
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320
- CCD. - Calculate Color Distance 1109
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

2.38.6 3.6 Using Quadrant-Coordinates

MiniRobotLanguage (MRL)

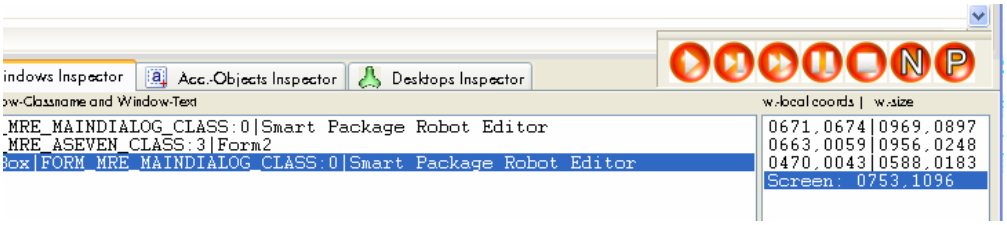
Quadrant-Coordinates

Intention

Normal Line or Rectangle (Rect-) Coordinates look like this:

```
x1,x2,y1,y2
```

Using these, you specify a (often) resolution dependent position on the screen.
Therefore they are generally screen-coordinates.



You can find out the screen coordinates where the mouse is, using the "windows Inspector". In this case it shows the values: `753,1096`
Example using Screen-Coordinates.

```
ILC.753,1096,900,1500|!&HFFFFFF|0|3
```

You can also specify Rect-Coordinates relative to the upper, left corner of the last window that you had located. These coordinates are called "Window coordinates".

You see them on the right side of each child-window in the "windows Inspector".
Example using "window coordinates":

```
ILC.w1,1,10,10|!&HFFFFFF|0|3
```

As an easier variant, you can use "Quadrants".

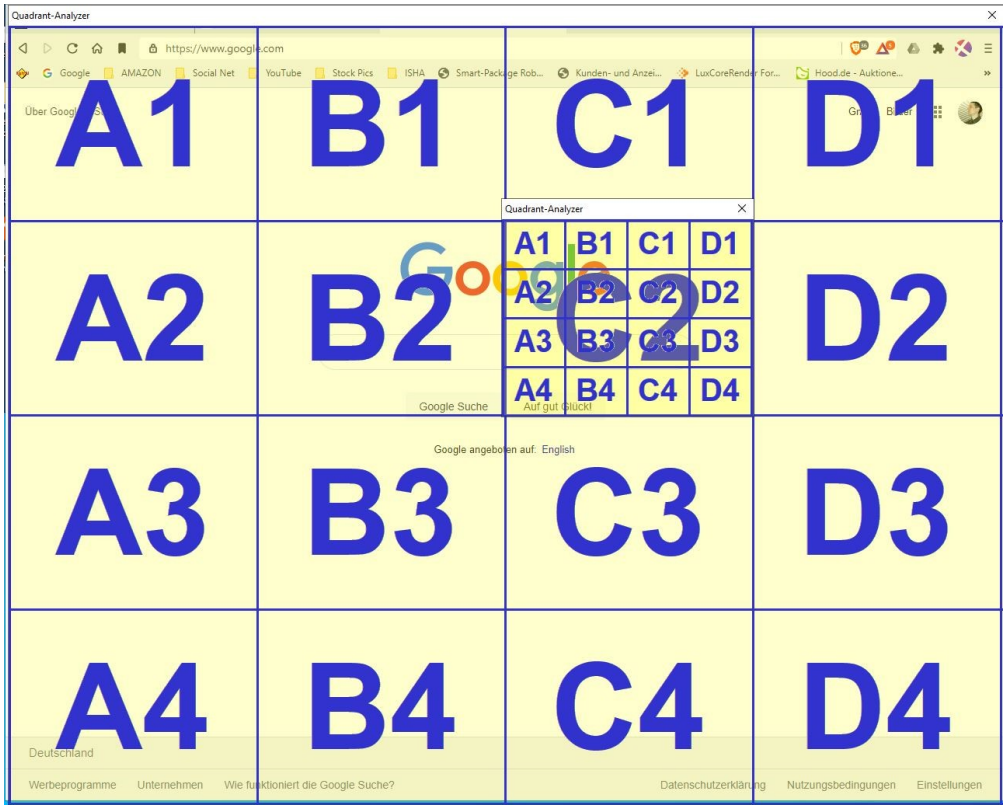
In short: **Quadrant-Coordinates are screen- and window-size independent.**
That is the advantage of using them. They are somehow similar numbered like the coordinates in a Spreadsheet.
Means that you may know the system already from your favorite Spreadsheet program.
It's the same system.

Using Quadrant- Coordinates, you specify the so called "Quadrant" that is

"`A1`" to "`D4`"

*© 2023 Theo Gottwald * https://www.smart-package.com*

Here is a Window showing its 16 Quadrants:



Notice that the Quadrant for Top-Windows may or may not contains the so called "non-Client Area", that is the Title-Bar and the complete windows-border.

In case you will wait for a change in this window, in Quadrant `{D4}` you would write:

```
STW.ct|#32770|Basic-NAV Matrix 01
RRA.{D4}
WRA.
DBP.Change happened!
ENR.
```

Be aware that even the window becoming "Active" or losing the Input-Focus will change the color of the window border and thus influence/change all outer Quadrants that are:
`A1-A4,C1,C4, D1-D4.`
Only `B2,B3,C2` and `C3` will not be changed if the window gets/loses Input Focus.

That is why you would normally locate a child-window or avoid using a Quadrant over the full window. More Details on this topic, see here:

1.8.1 The "Client" and the "Non-Client" Parts of a window 156

Here is an example:

```
  STW.ct|Chrome_WidgetWin_1|- Brave
  SWP.805,531|1076,1033|T

  IRC.t{B2}{A4,D4}@|&H0|45%|3
    DMP.6
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    $$XPO=$tos$
    $$YPO=$tos$
    MMV.s|$$XPO,$$YPO

  EIF.
  MBX.!

  QFR.|{D1}{D1}|$$REC
  GRM.$$REC|$$MID
  MMV.$$MID

  MBX.!
  ENR.
```

As you can see the Quadrant is been calculated from the last located object, in this case from another Quadrant.

Quadrant coordinates are transformed to screen coordinates before they are used. Therefore moving a window that is inspected by `WRC.`, `WRA.` or such an command will make it be "changed".

Details

The regular Pixel-commands like `RRA./WLC./WRC./WRA.` etc. interpret Quadrants different from the "Analyze"-Command, that will be available in future versions.

If you give the `RRA.`-command this Parameter:

```
RRA.{A1,B2}
```

this will internally be calculated to a larger rectangle (-union) that will include all given rectangles in one large rectangle. In this case it will be an rectangle that includes `A1,A2,B1` and `B2`.
This is done, as these commands can only work with a single rectangle at one time, but not with multiple rectangles.

This treatment of Quadrants is different from using the (future) Analyze Command.
In the Analyze command you do include the Quadrants into "()" brackets, not into "{}" brackets, which is the first difference.

More important, When using the `ANA.`.-command, you can specify which types of controls should be included in the Quadrants-Mask for other commands.

When using a "Quadrant", all Controls or Buttons etc. can be:

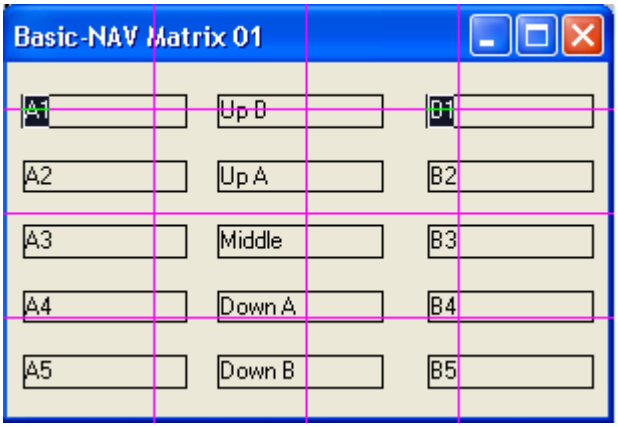
- completely inside a Quadrant (`f - full`)
- mostly inside a quadrant (middle-point is inside) (`m - middle point inside`)
- partly inside a Quadrant means that the control is inside but with less then half of the area. (`p - partly inside`)

Therefore acting on all Controls in a Quadrant, you will need to specify which controls to include or exclude. For example:

© 2023 Theo Gottwald * https://www.smart-package.com

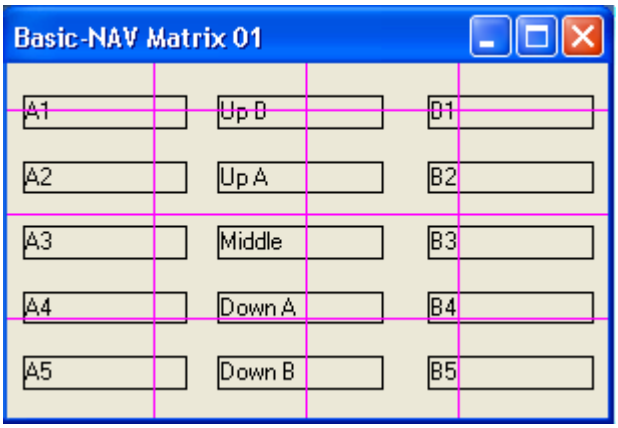
```
STW.ct|#32770|Basic-NAV Matrix 01
ANA.PD:8,TL:40,FL:48,MS:(A1,A3,p)
ENR.
```

Will act on all controls which touch the A1 or the A3 Quadrant:




```
STW.ct|#32770|Basic-NAV Matrix 01
ANA.PD:8,TL:40,FL:48,MS:(A1,A3,f)
ENR.
```

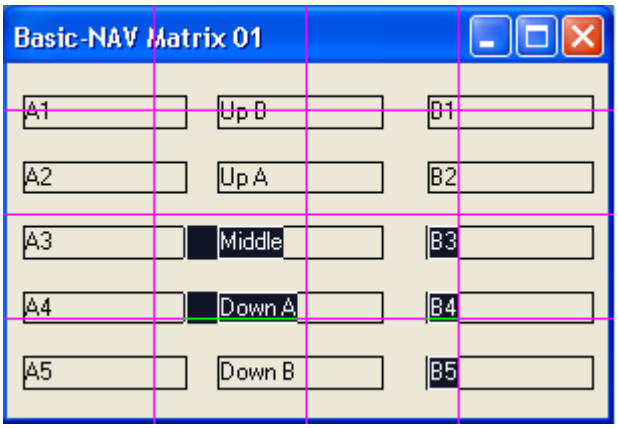
Will act only on all controls which are completely inside the A1 or the A3 Quadrant:
As there are no such controls, the result is this:



Here is a more complicated Example:

```
STW.ct|#32770|Basic-NAV Matrix 01
ANA.PD:8,TL:40,FL:48,MS:(C2,C3,D3,m)
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



It will mark all controls that are more then half inside the given Quadrants.

Remarks

-

Limitations:

-

See also:

- CCD. - Calculate Color Distance 1109
- CCV. - Color Conversion 1117
- WLC. - Wait for Line Color 1846
- ILC. / NLC. - If Line Color 1805
- IRC. / NRC, - If Rect Color 1828
- WRC. - Wait Rect Color 1856
- WRA. - Wait Rect-Area Altered 1851
- IRA. / NRA. - If Rect altered 1824
-

© 2023 Theo Gottwald * https://www.smart-package.com

2.39 4. Special features

2.39.1 4.1 : - Definiton-Command

MiniRobotLanguage (MRL)

Define Variable, MACRO or Label

Purpose:
This command can be used to define a **Variable**, **Macro** or a **Label**.

Define a variable

A **variable** is a placeholder for a numeric or string value.
A variable is replaced at runtime with the value behind.

Variables can be used to make your program more readable.
Also variables can save you a lot of typing work.
There are different ways to define a variable. Each of these have their own advantages and disadvantages.

If a variable name is used and there is no such variable defined, nothing will happen. It will be processed like normal text. The definition command makes a complete numeric and string resolution of the given values.

Using the definition command you can define and clear variables.

Example:

```
'This will clear the variable (set variable value to "empty"):
: $$ABC=

' This will assign a Value with complete Specialfolder 345-Processing
: $$ABC=Hallo
```

' This will assign the content of the Variable including the Variable-Type
' at this time there will be no Processing for Specialfolders etc. its just a COPY.

```
: $$ABC=$$CDE
```

Define a Label or Macro

A label is a jump-mark in the code. Program execution can jump to this mark, using the JMP or GSB, and other commands.

© 2023 Theo Gottwald * https://www.smart-package.com

A Macro is a user-defined command. It may contain several lines of code and parameters.
For Details on the MACRO option see the : - MACRO-Definitions 253.
For Details on variable-types see the VAR. 3562 - Command.

How does the robot know what you want?

The definiton of a Variable or a MACRO starts with a ": " (colon-blank).
A Label-Definiton may not contain a Blank! A label is defined like this:

```
:Here
```

Note that there can't be a blank after the colon - or it would be taken as the start of a Macro-Definition.

```
' This is a Label
:Label

' This is a MACRO:
: %Press 0
 MLE.
END%
```

' This will define a script-local **Variable**.

```
: $$BBB=99
```

' This will define a global **Variable**.

```
: $$BBB=99
```

For Information on Variables please see VAR. - Variable Set Value/Clear 3562.
For Information on Macros please look at : - MACRO-Definitions 253.

Syntax:

```
: [Variable]=[P1]
: [Macroname]
:LABEL
```

Example:

```
'This will clear the variable (set variable value to "empty"):
 : $$ABC=

' This will assign a Value without Specialfolder-Processing to
    a global variable
 : $$ABC=Hallo

' This will assign a Value with Specialfolder-Processing to a
    local variable
 : $$ABC=?path\hallo
```


© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation:

-

Remarks:

Its important to understand that the VAR. 3562-**Command will resolve Special-Folders** and Special Variables, while the **":" Definition-Command** will do that and also try a numeric resolution if there is a formula on the value-side.

Limitations:

-

See also:

- : - MACRO-Definitions 253
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218
- VAR. - Variable Set Value/Clear 3562

© 2023 Theo Gottwald * https://www.smart-package.com

2.39.2 4.2 :on_Error - Label

MiniRobotLanguage (MRL)

:on_Error Label

on Error goto

Intention

`:on_Error` is a special Label. Its not a Label like any other!

If **`:on_Error`** is there, the script will jump there after an error and continue script execution there.

`:on_Error` does NOT prevent error messages, For this you have to use the USM.-command.

If `:on_Error` is really called, will also depend on the settings of an `USM.` -command.

Assume an error occurs and there is such a Label in a script. Then the robot will continue script execution in that line that is below that label.

The ":`on_Error`" Label does not prevent the Error-Message to be displayed, use the `USM.10` command for that.

See more details and examples on error-handling here: 1.B Error handling 224.

Syntax

```
:on_Error
```

Parameter Explanation

-

Example

```
'*********************************************
' USM. - Samples
'*********************************************

'Switch on silent Mode
USM.10
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Switch on Silent Mode with Error-Log file
USM.10|?pfad\MyLog.txt

' Clear ":on_Error" Retry Flag
USM.64

' Switch off Silent Mode
USM.0

END.

:on_Error
MBX. There was an Error
END.
'*********************************************
```

Remarks

Limitations:

-

See also:

- 1.B Error handling 224
- **USM - Use silent Mode** 786

© 2023 Theo Gottwald * https://www.smart-package.com

2.39.3 4.3 Variables, global and local

MiniRobotLanguage (MRL)

Variables, GLOBAL and LOCAL

Intention

Variables are Symbols which contain numerical or string-data. When used in a script, the variable name is replaced with the variable content at runtime. The variable content can be defined, changed and cleared at runtime.

Global Variables are valid everywhere in a Script. Local Variables are used inside "Sub-Scripts" which are called with the JNF.-Command. Local variables contain their value until the script is left to the calling script. Then their content is lost.

If a script calls another script using JNF., the content of its local variables is preserved and automatically restored on return from the subscript.

Local variables are cleared when the Script returns to the Main-Script, using RTS. Local Variables and JNF. can be nested to unlimited depth.The Format for Variables is always:

§§xxx for script-local variables

$$xxx for global variables

Global variables keep their value anywhere in a script or subscript. Variables have a fixed length. Therefore we do not need any concatenation characters like in other scripting systems.

For xxx all letters or numbers as well as the "-" can be used.

There are few Variables which are reserved for Special Purposes which shall not be used for regular operations.

There are different Methods to assign values to variables.

- The : - Definiton-Command 361
- The VAR. - Variable Set Value/Clear 3562
- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAS. - Variable with String 3564

*© 2023 Theo Gottwald * https://www.smart-package.com*

Only when using the VAR.- type Commands, Speical-Folders and System-Variables will be resolved! The Definition command does not resolve special folders.

Also Commands do return results directly into variables.

Syntax

$$xxx global Variable
§§xxx local Variable

Parameter Explanation

If the Parameter is omitted, the Variable is set to empty, which represents string with a of length 0, or a value of 0.

Example

```
' define global variable without Special-folders
: $$TXT=Hallo World

' define global variable with Special-folders and System-variable Resolution
VAR.$$FIL=?progs\FS Office\Myprog.exe
PRT.$$FIL
' Display a list with all global Variables
DMP.4

' define local variable with Special-folders and System-variable Resolution
VAR.§§FIL=?progs\FS Office\Myprog.exe
PRT.§§FIL
' Following clears variable §§NUM
: §§NUM=
' Display a list with all local Variables
DMP.5
```

Remarks

Global Variables are valid everywhere in a Script. Local Variables are used inside "Sub-Scripts" which are called with the JNF.-Command.
Such variables are cleared when the Script returns to the Main-Script, using RTS.
Local Variables and JNF. can be nested to unlimited depth.

Limitations:

How long can variables get? What is the maximum size? 100 MB ? 250, 500 MB ?

© 2023 Theo Gottwald * https://www.smart-package.com

It depends on the amount of unfragmented Memory that the robot can get from the OS.
Under normal circumstances variables can always exceed a size of 50 or 100 MB (MegaBytes) without problems. Once you clear the variable with
`VAR.$$XXX=`
the memory is safely returned to the OS. Of course this happens automatically when the script ends.

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

2.39.4 4.4 The global and local Stack

MiniRobotLanguage (MRL)

The Stack - global and Local

Intention

Why can you do with a stack?

The stack is a space where you can store hundreds of items. These items can be strings (text) or numeric items. Note that

Commands use it, too
Many commands leave their results on the stack. The stack organization can be customized to user needs.

Customize your stacks
First you can change the number of elements the stack can hold. Use the STV. - command for this.

Any stack can be organized as a **stack** (LIFO - Last-In-First-Out) or as a **que** (First-In-First-Out). The default Stack setting is LIFO. Use the STS. - command to switch the stack between LIFO and FIFO. Note that if a stack gets to many elements, some overflowing elements are automatically deleted. These overflowing elements are generally the oldest elements. In a LIFO stack the oldest element is the first element added, while in FIFO (Que), the last added elements added ist the "oldest".

Another setting is, if t he stack element is remove if you touch it. Normally a stack element it deleted immediately when you touch it. This way you get all stack elements one-by-one, just by using `$tos$` which is always the TOS (Top of Stack) in the actual selected stack. If you want to get the TOS without having it removed, use `$dtos$`.

What if i jump into a subprogram that also uses the stack?
You can anytime take the whole stack-content and put it into a variable, using the STV.-command. When you return from your subprogram, you can "restore" the complete stack from that variable using the STV.-command.

Use the script local stack
If you use Subscripts (scripts in separate files) you can take advantage of another feature.
Local and Global stacks. In fact, we have two stacks, a script-local stack and a global stack. The global stack is valid through the whole script execution. The local stack holds the same rules like the script local variables.

If a script calls another Subscript in another file, using the `JNF. command`. At this time the content of its local stack is preserved and automatically restored on return from the subscript. Thats the use of a local stack. To hold your script local data.

*© 2023 Theo Gottwald * https://www.smart-package.com*

However, the Local stack is cleared when the Script returns upwards to a Main-Script, from which it was called, using `JNF./RTS`. Its the same sytem like when using script local variables.

How do I select the actual stack/que?
To select which stack you want to work with, use the `STS.GLOB` or `STS.LOCA` command.
These will make the selected stack to the actual stack.

If you want to know which stack is currently the actual stack, use the **`#stack#`** systemvariable. It will return a number that shows you if the Local or the Global stack is currently selected as the actual stack.
#stack# -> 1 GLOBAL Stack is selected
#stack# -> 2 LOCAL Stack is selected

There are shortcuts to accessing the stack values inside your script.

§§000 to §§999
for directly accessing the local stack

$$000 to $$999
for directly accessing the global stack

§§000 - Top of Stack (TOS) local stack
$$000 - Top of Stack (TOS) global stack

The global stack keeps his content anywhere in a script or subscript.

There are different Methods to access the stack. First you can use the Stack-variables above to directly access the first 999 stack elements. Secondly you can use multiple stack-commands.

- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010
- STJ. - STack Job 3030
- STS. - Set STack Settings 3038
- STV. - Stack To/from Variable 3040

Most Commands may return results directly into stack-variables if you give these for the result.

Example

```
' Choose LOCAL Stack as Actual Stack
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STS.LOCA
' Organize Stack as First-in-First-Out
STS.FIFO

' Put 10 Items on actual Stack
FOR.$$A01|1|10
  VAR.$$B00=Stackplace $$A01
  PUV.$$B00
NEX.

' Show all Items on Local Stack
DMP.5
MBX.Ready
ENR.
```

Remarks

The $$000 and §§000 refer to the TOS (Top of stack). See several example scripts in the examples folder.

Limitations:

Per default the maximum stack size limit is set to 25 Elements. This can be changed using the STS.SIZE command.
How long can stack-entry get? Can you strore a file in it?
What is the maximum size? 100 MB ? 250, 500 MB ?
It depends on the amount of unfragmented Memory that the robot can get from the OS.
Under normal circumstances any stack entry can always exceed a size of 50 or 100 MB (Megabyte) without problems. Once you clear the stack entry, in any way, the memory is safely returned to the OS. Of course this happens automatically when the script ends.

See also:

STJ. - STack Job 3030
STS. - Set STack Settings 3038
STV. - Stack To/from Variable 3040
PUS. - Push Parameter onto Stack/Que 3013
PUV. PUsh variables on the User-Stack 3019
POP. - POP Variable from Stack 3007
POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

2.39.5 API-Error-Codes

MiniRobotLanguage (MRL)

API_ERROR_CODES

Error Codes

Intention

Normally you should not get this codes unless explicitly stated in this helpfile.

DOS-Error Code (1 - 26):

```
1 - Incorrect function call.
2 - The system cannot find the file specified.
3 - The system cannot find the path specified.
4 - The system cannot open the file. (too many open files)
5 - Access to that file was denied.
6 - The handle is invalid.
7 - The storage control blocks were destroyed.
8 - Not enough storage is available to process this command.
9 - The storage control block address is invalid.
10 - The environment is incorrect.
11 - The .EXE file is invalid (non-Win32 .EXE).
12 - The access code is invalid.
13 - The data is invalid.
14 - Not enough storage is available to complete this operation.
15 - The system cannot find the drive specified.
16 - The directory cannot be removed (current dir).
17 - The system cannot move the file to a different disk drive.
18 - There are no more files.
19 - The media is write protected.
20 - The system cannot find the device specified.
21 - The device is not ready.
22 - The device does not recognize the command.
23 - Data error (cyclic redundancy check).
24 - The program issued a command but the command length is
incorrect.
25 - The drive cannot locate a specific area or track on the disk.
26 - The specified disk or diskette cannot be accessed.
```

Error Codes

```
ERROR_SUCCESS              = 0&
INVALID_FUNCTION           = 1&      ' dderror
FILE_NOT_FOUND             = 2&
PATH_NOT_FOUND             = 3&
TOO_MANY_OPEN_FILES        = 4&
ACCESS_DENIED              = 5&
INVALID_HANDLE             = 6&
ARENA_TRASHED              = 7&
NOT_ENOUGH_MEMORY          = 8&      ' dderror
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
INVALID_BLOCK               = 9&
BAD_ENVIRONMENT             = 10&
BAD_FORMAT                  = 11&
INVALID_ACCESS              = 12&
INVALID_DATA                = 13&
OUTOFMEMORY                 = 14&
INVALID_DRIVE               = 15&
CURRENT_DIRECTORY           = 16&
NOT_SAME_DEVICE             = 17&
NO_MORE_FILES               = 18&
WRITE_PROTECT               = 19&
BAD_UNIT                    = 20&
NOT_READY                   = 21&
BAD_COMMAND                 = 22&
CRC                         = 23&
BAD_LENGTH                  = 24&
SEEK                        = 25&
NOT_DOS_DISK                = 26&
SECTOR_NOT_FOUND            = 27&
OUT_OF_PAPER                = 28&
WRITE_FAULT                 = 29&
READ_FAULT                  = 30&
GEN_FAILURE                 = 31&
SHARING_VIOLATION           = 32&
LOCK_VIOLATION              = 33&
WRONG_DISK                  = 34&
SHARING_BUFFER_EXCEEDED     = 36&
HANDLE_EOF                  = 38&
HANDLE_DISK_FULL            = 39&
NOT_SUPPORTED               = 50&
REM_NOT_LIST                = 51&
DUP_NAME                    = 52&
BAD_NETPATH                 = 53&
NETWORK_BUSY                = 54&
DEV_NOT_EXIST               = 55&     ' dderror
TOO_MANY_CMDS               = 56&
ADAP_HDW_ERR                = 57&
BAD_NET_RESP                = 58&
UNEXP_NET_ERR               = 59&
BAD_REM_ADAP                = 60&
PRINTQ_FULL                 = 61&
NO_SPOOL_SPACE              = 62&
PRINT_CANCELLED             = 63&
NETNAME_DELETED             = 64&
NETWORK_ACCESS_DENIED       = 65&
BAD_DEV_TYPE                = 66&
BAD_NET_NAME                = 67&
TOO_MANY_NAMES              = 68&
TOO_MANY_SESS               = 69&
SHARING_PAUSED              = 70&
REQ_NOT_ACCEP               = 71&
REDIR_PAUSED                = 72&
FILE_EXISTS                 = 80&
CANNOT_MAKE                 = 82&
FAIL_I24                    = 83&
OUT_OF_STRUCTURES           = 84&
ALREADY_ASSIGNED            = 85&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
INVALID_PASSWORD           = 86&
INVALID_PARAMETER          = 87&     ' dderror
NET_WRITE_FAULT            = 88&
NO_PROC_SLOTS              = 89&
TOO_MANY_SEMAPHORES        = 100&
EXCL_SEM_ALREADY_OWNED     = 101&
SEM_IS_SET                 = 102&
TOO_MANY_SEM_REQUESTS      = 103&
INVALID_AT_INTERRUPT_TIME  = 104&
SEM_OWNER_DIED             = 105&
SEM_USER_LIMIT             = 106&
DISK_CHANGE                = 107&
DRIVE_LOCKED               = 108&
BROKEN_PIPE                = 109&
OPEN_FAILED                = 110&
BUFFER_OVERFLOW            = 111&
DISK_FULL                  = 112&
NO_MORE_SEARCH_HANDLES     = 113&
INVALID_TARGET_HANDLE      = 114&
INVALID_CATEGORY           = 117&
INVALID_VERIFY_SWITCH      = 118&
BAD_DRIVER_LEVEL           = 119&
CALL_NOT_IMPLEMENTED       = 120&
SEM_TIMEOUT                = 121&
INSUFFICIENT_BUFFER        = 122&     ' dderror
INVALID_NAME               = 123&     ' dderror
INVALID_LEVEL              = 124&
NO_VOLUME_LABEL            = 125&
MOD_NOT_FOUND              = 126&
PROC_NOT_FOUND             = 127&
WAIT_NO_CHILDREN           = 128&
CHILD_NOT_COMPLETE         = 129&
DIRECT_ACCESS_HANDLE       = 130&
NEGATIVE_SEEK              = 131&
SEEK_ON_DEVICE             = 132&
IS_JOIN_TARGET             = 133&
IS_JOINED                  = 134&
IS_SUBSTED                 = 135&
NOT_JOINED                 = 136&
NOT_SUBSTED                = 137&
JOIN_TO_JOIN               = 138&
SUBST_TO_SUBST             = 139&
JOIN_TO_SUBST              = 140&
SUBST_TO_JOIN              = 141&
BUSY_DRIVE                 = 142&
SAME_DRIVE                 = 143&
DIR_NOT_ROOT               = 144&
DIR_NOT_EMPTY              = 145&
IS_SUBST_PATH              = 146&
IS_JOIN_PATH               = 147&
PATH_BUSY                  = 148&
IS_SUBST_TARGET            = 149&
SYSTEM_TRACE               = 150&
INVALID_EVENT_COUNT        = 151&
TOO_MANY_MUXWAITERS        = 152&
INVALID_LIST_FORMAT        = 153&
LABEL_TOO_LONG             = 154&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TOO_MANY_TCBS              = 155&
SIGNAL_REFUSED             = 156&
DISCARDED                  = 157&
NOT_LOCKED                 = 158&
BAD_THREADID_ADDR          = 159&
BAD_ARGUMENTS              = 160&
BAD_PATHNAME               = 161&
SIGNAL_PENDING             = 162&
MAX_THRDS_REACHED          = 164&
LOCK_FAILED                = 167&
BUSY                       = 170&
CANCEL_VIOLATION           = 173&
ATOMIC_LOCKS_NOT_SUPPORTED = 174&
INVALID_SEGMENT_NUMBER     = 180&
INVALID_ORDINAL            = 182&
ALREADY_EXISTS             = 183&
INVALID_FLAG_NUMBER        = 186&
SEM_NOT_FOUND              = 187&
INVALID_STARTING_CODESEG   = 188&
INVALID_STACKSEG           = 189&
INVALID_MODULETYPE         = 190&
INVALID_EXE_SIGNATURE      = 191&
EXE_MARKED_INVALID         = 192&
BAD_EXE_FORMAT             = 193&
ITERATED_DATA_EXCEEDS_64k  = 194&
INVALID_MINALLOCSIZE       = 195&
DYNLINK_FROM_INVALID_RING  = 196&
IOPL_NOT_ENABLED           = 197&
INVALID_SEGDPL             = 198&
AUTODATASEG_EXCEEDS_64k    = 199&
RING2SEG_MUST_BE_MOVABLE   = 200&
RELOC_CHAIN_XEEDS_SEGLIM   = 201&
INFLOOP_IN_RELOC_CHAIN     = 202&
ENVVAR_NOT_FOUND           = 203&
NO_SIGNAL_SENT             = 205&
FILENAME_EXCED_RANGE       = 206&
RING2_STACK_IN_USE         = 207&
META_EXPANSION_TOO_LONG    = 208&
INVALID_SIGNAL_NUMBER      = 209&
THREAD_1_INACTIVE          = 210&
LOCKED                     = 212&
TOO_MANY_MODULES           = 214&
NESTING_NOT_ALLOWED        = 215&
EXE_MACHINE_TYPE_MISMATCH  = 216&
BAD_PIPE                   = 230&
PIPE_BUSY                  = 231&
NO_DATA                    = 232&
PIPE_NOT_CONNECTED         = 233&
MORE_DATA                  = 234&     ' dderror
VC_DISCONNECTED            = 240&
INVALID_EA_NAME            = 254&
EA_LIST_INCONSISTENT       = 255&
%WAIT_TIMEOUT              = 258&
NO_MORE_ITEMS              = 259&
CANNOT_COPY                = 266&
DIRECTORY                  = 267&
EAS_DIDNT_FIT              = 275&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EA_FILE_CORRUPT            = 276&
EA_TABLE_FULL              = 277&
INVALID_EA_HANDLE          = 278&
EAS_NOT_SUPPORTED          = 282&
NOT_OWNER                  = 288&
TOO_MANY_POSTS             = 298&
PARTIAL_COPY               = 299&
OPLOCK_NOT_GRANTED         = 300&
INVALID_OPLOCK_PROTOCOL    = 301&
MR_MID_NOT_FOUND           = 317&
INVALID_ADDRESS            = 487&
ARITHMETIC_OVERFLOW        = 534&
PIPE_CONNECTED             = 535&
PIPE_LISTENING             = 536&
EA_ACCESS_DENIED           = 994&
OPERATION_ABORTED          = 995&
IO_INCOMPLETE              = 996&
IO_PENDING                 = 997&     ' dderror
NOACCESS                   = 998&
SWAPERROR                  = 999&
STACK_OVERFLOW             = 1001&
INVALID_MESSAGE            = 1002&
CAN_NOT_COMPLETE           = 1003&
INVALID_FLAGS              = 1004&
UNRECOGNIZED_VOLUME        = 1005&
FILE_INVALID               = 1006&
FULLSCREEN_MODE            = 1007&
NO_TOKEN                   = 1008&
BADDB                      = 1009&
BADKEY                     = 1010&
CANTOPEN                   = 1011&
CANTREAD                   = 1012&
CANTWRITE                  = 1013&
REGISTRY_RECOVERED         = 1014&
REGISTRY_CORRUPT           = 1015&
REGISTRY_IO_FAILED         = 1016&
NOT_REGISTRY_FILE          = 1017&
KEY_DELETED                = 1018&
NO_LOG_SPACE               = 1019&
KEY_HAS_CHILDREN           = 1020&
CHILD_MUST_BE_VOLATILE     = 1021&
NOTIFY_ENUM_DIR            = 1022&
DEPENDENT_SERVICES_RUNNING = 1051&
INVALID_SERVICE_CONTROL    = 1052&
SERVICE_REQUEST_TIMEOUT    = 1053&
SERVICE_NO_THREAD          = 1054&
SERVICE_DATABASE_LOCKED    = 1055&
SERVICE_ALREADY_RUNNING    = 1056&
INVALID_SERVICE_ACCOUNT    = 1057&
SERVICE_DISABLED           = 1058&
CIRCULAR_DEPENDENCY        = 1059&
SERVICE_DOES_NOT_EXIST     = 1060&
SERVICE_CANNOT_ACCEPT_CTRL = 1061&
SERVICE_NOT_ACTIVE         = 1062&
FAILED_SERVICE_CONTROLLER_CONNECT = 1063&
EXCEPTION_IN_SERVICE       = 1064&
DATABASE_DOES_NOT_EXIST    = 1065&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SERVICE_SPECIFIC_ERROR     = 1066&
PROCESS_ABORTED            = 1067&
SERVICE_DEPENDENCY_FAIL    = 1068&
SERVICE_LOGON_FAILED       = 1069&
SERVICE_START_HANG         = 1070&
INVALID_SERVICE_LOCK       = 1071&
SERVICE_MARKED_FOR_DELETE  = 1072&
SERVICE_EXISTS             = 1073&
ALREADY_RUNNING_LKG        = 1074&
SERVICE_DEPENDENCY_DELETED = 1075&
BOOT_ALREADY_ACCEPTED      = 1076&
SERVICE_NEVER_STARTED      = 1077&
DUPLICATE_SERVICE_NAME     = 1078&
DIFFERENT_SERVICE_ACCOUNT  = 1079&
CANNOT_DETECT_DRIVER_FAILURE = 1080&
CANNOT_DETECT_PROCESS_ABORT = 1081&
NO_RECOVERY_PROGRAM        = 1082&
SERVICE_NOT_IN_EXE         = 1083&
END_OF_MEDIA               = 1100&
FILEMARK_DETECTED          = 1101&
BEGINNING_OF_MEDIA         = 1102&
SETMARK_DETECTED           = 1103&
NO_DATA_DETECTED           = 1104&
PARTITION_FAILURE          = 1105&
INVALID_BLOCK_LENGTH       = 1106&
DEVICE_NOT_PARTITIONED     = 1107&
UNABLE_TO_LOCK_MEDIA       = 1108&
UNABLE_TO_UNLOAD_MEDIA     = 1109&
MEDIA_CHANGED              = 1110&
BUS_RESET                  = 1111&
NO_MEDIA_IN_DRIVE          = 1112&
NO_UNICODE_TRANSLATION     = 1113&
DLL_INIT_FAILED            = 1114&
SHUTDOWN_IN_PROGRESS       = 1115&
NO_SHUTDOWN_IN_PROGRESS    = 1116&
IO_DEVICE                  = 1117&
SERIAL_NO_DEVICE           = 1118&
IRQ_BUSY                   = 1119&
MORE_WRITES                = 1120&
COUNTER_TIMEOUT            = 1121&
FLOPPY_ID_MARK_NOT_FOUND   = 1122&
FLOPPY_WRONG_CYLINDER      = 1123&
FLOPPY_UNKNOWN_ERROR       = 1124&
FLOPPY_BAD_REGISTERS       = 1125&
DISK_RECALIBRATE_FAILED    = 1126&
DISK_OPERATION_FAILED      = 1127&
DISK_RESET_FAILED          = 1128&
EOM_OVERFLOW               = 1129&
NOT_ENOUGH_SERVER_MEMORY   = 1130&
POSSIBLE_DEADLOCK          = 1131&
MAPPED_ALIGNMENT           = 1132&
SET_POWER_STATE_VETOED     = 1140&
SET_POWER_STATE_FAILED     = 1141&
TOO_MANY_LINKS             = 1142&
OLD_WIN_VERSION            = 1150&
APP_WRONG_OS               = 1151&
SINGLE_INSTANCE_APP        = 1152&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RMODE_APP                   = 1153&
INVALID_DLL                 = 1154&
NO_ASSOCIATION              = 1155&
DDE_FAIL                    = 1156&
DLL_NOT_FOUND               = 1157&
NO_MORE_USER_HANDLES        = 1158&
MESSAGE_SYNC_ONLY           = 1159&
SOURCE_ELEMENT_EMPTY        = 1160&
DESTINATION_ELEMENT_FULL    = 1161&
ILLEGAL_ELEMENT_ADDRESS     = 1162&
MAGAZINE_NOT_PRESENT        = 1163&
DEVICE_REINITIALIZATION_NEEDED = 1164&     ' dderror
DEVICE_REQUIRES_CLEANING    = 1165&
DEVICE_DOOR_OPEN            = 1166&
DEVICE_NOT_CONNECTED        = 1167&
NOT_FOUND                   = 1168&
NO_MATCH                    = 1169&
SET_NOT_FOUND               = 1170&
POINT_NOT_FOUND             = 1171&
NO_TRACKING_SERVICE         = 1172&
NO_VOLUME_ID                = 1173&
INVALID_VLM_OPERATION       = 1174&
UNABLE_TO_REMOVE_REPLACED   = 1175&
UNABLE_TO_MOVE_REPLACEMENT  = 1176&
UNABLE_TO_MOVE_REPLACEMENT_2 = 1177&
JOURNAL_DELETE_IN_PROGRESS  = 1178&
JOURNAL_NOT_ACTIVE          = 1179&
POTENTIAL_FILE_FOUND        = 1180&
JOURNAL_ENTRY_DELETED       = 1181&
CANCELLED                   = 1223&
USER_MAPPED_FILE            = 1224&
CONNECTION_REFUSED          = 1225&
GRACEFUL_DISCONNECT         = 1226&
ADDRESS_ALREADY_ASSOCIATED  = 1227&
ADDRESS_NOT_ASSOCIATED      = 1228&
CONNECTION_INVALID          = 1229&
CONNECTION_ACTIVE           = 1230&
NETWORK_UNREACHABLE         = 1231&
HOST_UNREACHABLE            = 1232&
PROTOCOL_UNREACHABLE        = 1233&
PORT_UNREACHABLE            = 1234&
REQUEST_ABORTED             = 1235&
CONNECTION_ABORTED          = 1236&
RETRY                       = 1237&
CONNECTION_COUNT_LIMIT      = 1238&
LOGIN_TIME_RESTRICTION      = 1239&
LOGIN_WKSTA_RESTRICTION     = 1240&
INCORRECT_ADDRESS           = 1241&
ALREADY_REGISTERED          = 1242&
SERVICE_NOT_FOUND           = 1243&
NOT_AUTHENTICATED           = 1244&
NOT_LOGGED_ON               = 1245&
CONTINUE                    = 1246&     ' dderror
ALREADY_INITIALIZED         = 1247&
NO_MORE_DEVICES             = 1248&     ' dderror
NO_SUCH_SITE                = 1249&
DOMAIN_CONTROLLER_EXISTS    = 1250&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ONLY_IF_CONNECTED          = 1251&
OVERRIDE_NOCHANGES         = 1252&
BAD_USER_PROFILE           = 1253&
NOT_SUPPORTED_ON_SBS       = 1254&
LICENSE_QUOTA_EXCEEDED     = 1395&
WRONG_TARGET_NAME          = 1396&
MUTUAL_AUTH_FAILED         = 1397&
TIME_SKEW                  = 1398&
GLOBAL_ONLY_HOOK           = 1429&
NO_SYSTEM_RESOURCES        = 1450&
NONPAGED_SYSTEM_RESOURCES  = 1451&
PAGED_SYSTEM_RESOURCES     = 1452&
WORKING_SET_QUOTA          = 1453&
PAGEFILE_QUOTA             = 1454&
COMMITMENT_LIMIT           = 1455&
MENU_ITEM_NOT_FOUND        = 1456&
INVALID_KEYBOARD_HANDLE    = 1457&
HOOK_TYPE_NOT_ALLOWED      = 1458&
REQUIRES_INTERACTIVE_WINDOWSTATION = 1459&
TIMEOUT                    = 1460&
INVALID_MONITOR_HANDLE     = 1461&
INSTALL_SERVICE_FAILURE    = 1601&
INSTALL_USEREXIT           = 1602&
INSTALL_FAILURE            = 1603&
INSTALL_SUSPEND            = 1604&
UNKNOWN_PRODUCT            = 1605&
UNKNOWN_FEATURE            = 1606&
UNKNOWN_COMPONENT          = 1607&
UNKNOWN_PROPERTY           = 1608&
INVALID_HANDLE_STATE       = 1609&
BAD_CONFIGURATION          = 1610&
INDEX_ABSENT               = 1611&
INSTALL_SOURCE_ABSENT      = 1612&
INSTALL_PACKAGE_VERSION    = 1613&
PRODUCT_UNINSTALLED        = 1614&
BAD_QUERY_SYNTAX           = 1615&
INVALID_FIELD              = 1616&
DEVICE_REMOVED             = 1617&
INSTALL_ALREADY_RUNNING    = 1618&
INSTALL_PACKAGE_OPEN_FAILED = 1619&
INSTALL_PACKAGE_INVALID    = 1620&
INSTALL_UI_FAILURE         = 1621&
INSTALL_LOG_FAILURE        = 1622&
INSTALL_LANGUAGE_UNSUPPORTED = 1623&
INSTALL_TRANSFORM_FAILURE  = 1624&
INSTALL_PACKAGE_REJECTED   = 1625&
FUNCTION_NOT_CALLED        = 1626&
FUNCTION_FAILED            = 1627&
INVALID_TABLE              = 1628&
DATATYPE_MISMATCH          = 1629&
UNSUPPORTED_TYPE           = 1630&
CREATE_FAILED              = 1631&
INSTALL_TEMP_UNWRITABLE    = 1632&
INSTALL_PLATFORM_UNSUPPORTED = 1633&
INSTALL_NOTUSED            = 1634&
PATCH_PACKAGE_OPEN_FAILED  = 1635&
PATCH_PACKAGE_INVALID      = 1636&
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PATCH_PACKAGE_UNSUPPORTED  = 1637&
PRODUCT_VERSION            = 1638&
INVALID_COMMAND_LINE       = 1639&
INSTALL_REMOTE_DISALLOWED  = 1640&
SUCCESS_REBOOT_INITIATED   = 1641&
PATCH_TARGET_NOT_FOUND     = 1642&
```

Example

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

2.39.6 Regular Expressions

MiniRobotLanguage (MRL)

Regular_Expression

Regular Expressions in detail

Intention

Regular Expressions can be used with the GRE. and the GRR. Command.

Syntax

Details of "regular expressions" for the GRE. and the GRR. command

Parameter Explanation

. (period)
Matches any character, except the end-of-line.

^ (caret)
Matches the actual beginning-of-line position or the preceding line-delimiter character pair (ACII(13,10)), as taken from the start& character position. The line-delimiter characters themselves are not replaced. Also see [^] below for usage within a character class definition.

$ (dollar)
Matches the end-of-line position, which may be the either the first line-delimiter character pair (ACII(13,10)) that is encountered in the search to the right of the start position, or the actual end of the Mainstring, whichever occurs first. The line-delimiter characters themselves are not replaced.
| (stile)
Specifies alternation (the OR operator), so that an expression on either side can match. Precedence is from left-to-right, as encountered in the expression.

? (question mark)

*© 2023 Theo Gottwald * https://www.smart-package.com*

Specifies that zero or one match of the preceding sub-pattern is allowed. Cannot be used with a Tag.

+ (plus)
Specifies that one or more matches of the preceding sub-pattern are allowed. Cannot be used with a Tag.

*** (asterisk)**
Specifies that zero or more matches of the preceding sub-pattern are allowed. Cannot be used with a Tag.

Character classes

[ ] (square brackets)
Identifies a user-defined class of characters, any of which will match: [abc] will match a, b, or c. Only three special metacharacters are recognized within a class definition, the caret (^) for complemented characters, the hyphen (-) for a range of characters, or one of the following \ backslash escape sequences:
\\ \- \] \e \f \n \q \r \t \v \x##

Any other use of a backslash within a class definition yields an undefined operation that should be avoided.

[-] (hyphen)
The hyphen identifies a range of characters to match. For example, [a-f] will match a, b, c, d, e, or f.
Characters in an individual range must occur in the natural order as they appear in the character set. For example, [f-a] will match nothing.
Lists of characters, and one or more ranges of characters, may be intermixed in a single class definition. The start and end of a range may be specified by a literal character, or one of the \ backslash escape sequences:

\\ \- \] \e \f \n \q \r \t \v \x##
Any other use of a backslash within a class definition yields an undefined operation.
Multiple ranges in a class are valid. For example, [a-d2-5] matches a, b, c, d, 2, 3, 4, or 5.

When the hyphen is escaped, it is treated as a literal.
For example, [a\-c] is a list, not a range, and matches a, -, or c due to the \ backslash escape sequence.

[^] (caret)
When the caret appears as the first item in a class definition, it identifies a complemented class of characters, which will not match. For example, [^abc] matches any character except a, b, or c.
A range can also be specified for the complemented class.
For example, [^a-z] matches any character except a through z.

© 2023 Theo Gottwald * https://www.smart-package.com

A caret located in any position other than the first is treated as a literal character.

Tags/sub-patterns

( )(parentheses)
Parentheses are used to match a Tag, or sub-pattern, within the full search pattern, and remember the match. The matched sub-pattern can be retrieved later in the mask, or in a replace operation, with \01 through \99, based upon the left-to-right position of the opening parentheses.
Parentheses may also be used to force precedence of evaluation with the alternation operator. For example, "(Begin)|(End) File" would match either "BeginFile" or "EndFile", but without the Tag designations, "Begin|EndFile" would only match either "BeginndFile" or "BegiEndFile".

Note: Parentheses may not be used with ? + * as any match repetition could cause the tag value to be ambiguous. To match repeated expressions, use parentheses followed by \01*.

Escaped characters

\ (backslash).
The escape operator (single-character quote). The following character will be treated as a literal value rather than being interpreted as a special character. Note that the character following the backslash must actually be a special character, as follows:
\b
A word boundary. The start or end of a word, where a word is defined as one or more characters that include an alphabetic character (A-Z or a-z), a numeric character (0-9), and an underscore. For example, "abc_123" is considered a single word and "abc-123" is considered two words.

\c Case-sensitive search.
Without the \c operator, the default is to ignore case when matching. Unlike some other implementations of regular expressions, case-insensitivity is recognized in all operations, even a range of characters such as "[6-Z]". The
\c operator may appear at any position in the mask.
\e Escape character: ACII(27).
\f Formfeed character: ACII(12).
\n Linefeed (or newline) character: ACII(10).
\q Double-quote mark ("): ACII(34).
\q is included for ease of inclusion within a literal string. For example: "\qHello\q".
\r Carriage-return character: ACII(13).
\s Shortest match character:
The \s flag causes the shortest matching string to be returned, rather than the longest (the default). For example, when searching for the mask "abc.*abc" in "abcdabcabc", the default setting would return position 1 and length 10. With the \s switch set, it returns position 1 and length 7. This option may cause a slight increase in processing time. The \S flag must appear at the beginning of the mask string.

© 2023 Theo Gottwald * https://www.smart-package.com

\t Horizontal tab character: ACII(9).

\v Vertical tab character: ACII(11).

\x## Hex character code: Indicates that an ASCII code follows, given by two hexadecimal digits. For example, \xFF = ACII(255) (which is equivalent to ACII(255)). XX must be in the range 0 through 255.

\## Tag number: Evaluated as the characters matched by tag number ## (where ## is in the range 01 through 99, in decimal). Tags are implicitly numbered from 01 through 99, based upon the left-to-right position of the left parenthesis. "(…)w\01" would match "abcwabc" or "456w456". Tags cannot be forward-referenced - that is, if a reference is made to any Tag that is not yet defined, a non-match is presumed.

Example

```
DBM.2
PRT.Running!
PRT.#dtime#
: $$NUM=2
: $$MAN=please send an email to info@it-berater.org
: $$MAC=please send an email to info@it-berater.org
: $$MSK=([a-z0-9._/+-]+)(@[a-z0-9.-]+)
: $$REP=[E-Mail]
GRE.$$NUM,$$MAN,$$MSK
PRT.#dsince#
STS.DUMP
GRR.$$NUM,$$MAC,$$MSK,$$REP
PRT.$$MAC
STS.DUMP
PRT.$$000-$$001-$$002
STS.DUMP
STS.DUMP
DMP.
END.
```

Remarks

Limitations:

To maximize performance, avoid overuse of the *, + and ? metacharacters.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

2.39.7 Language-ID's

Language-ID's

Language-ID's

Intention

1078 Afrikaans	1037 Hebrew
1052 Albanian	1081 Hindi 5121 Arabic (Algeria)
1038 Hungarian	
15361 Arabic (Bahrain)	1039 Icelandic
3073 Arabic (Egypt)	1057 Indonesian
2049 Arabic (Iraq)	1040 Italian (Italy)
11265 Arabic (Jordan)	2064 Italian (Switzerland)
13313 Arabic (Kuwait)	1041 Japanese
12289 Arabic (Lebanon)	1099 Kannada
4097 Arabic (Libya)	1087 Kazakh
6145 Arabic (Morocco)	1111 Konkani
8193 Arabic (Oman)	1042 Korean
16385 Arabic (Qatar)	1088 Kyrgyz (Cyrillic)
1025 Arabic (Saudi Arabia)	1062 Latvian
10241 Arabic (Syria)	1063 Lithuanian
7169 Arabic (Tunisia)	1071 Macedonian (FYROM)
14337 Arabic (U.A.E.)	2110 Malay (Brunei Darussalam)
9217 Arabic (Yemen)	1086 Malay (Malaysia)
1067 Armenian	1102 Marathi
2092 Azeri (Cyrillic)	1104 Mongolian (Cyrillic)
1068 Azeri (Latin)	1044 Norwegian (Bokmal)
1069 Basque	2068 Norwegian (Nynorsk)
	1065 Persian
1059 Belarusian	1045 Polish
1026 Bulgarian	1046 Portuguese (Brazil)
1027 Catalan	2070 Portuguese (Portugal)
3076 Chinese (Hong Kong SAR)	1094 Punjabi
5124 Chinese (Macau SAR)	1048 Romanian
2052 Chinese (PRC)	1049 Russian
4100 Chinese (Singapore)	1103 Sanskrit
1028 Chinese (Taiwan)	3098 Serbian (Cyrillic)
1050 Croatian	2074 Serbian (Latin)
1029 Czech	1051 Slovak
1030 Danish	1060 Slovenian
1125 Divehi	11274 Spanish (Argentina)
2067 Dutch (Belgium)	16394 Spanish (Bolivia)
1043 Dutch (Netherlands)	13322 Spanish (Chile)
3081 English (Australia)	9226 Spanish (Colombia)
10249 English (Belize)	5130 Spanish (Costa Rica)
4105 English (Canada)	7178 Spanish (Dominican Republic)
9225 English (Caribbean)	12298 Spanish (Ecuador)
6153 English (Ireland)	17418 Spanish (El Salvador)
8201 English (Jamaica)	4106 Spanish (Guatemala)
5129 English (New Zealand)	18442 Spanish (Honduras)
13321 English (Philippines)	2058 Spanish (Mexico)

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
7177 English (South Africa) 19466 Spanish (Nicaragua)
11273 English (Trinidad)    6154 Spanish (Panama)
2057 English (United Kingdom) 15370 Spanish (Paraguay)
1033 English (United States) 10250 Spanish (Peru)
12297 English (Zimbabwe)   20490 Spanish (Puerto Rico)
1061 Estonian              3082 Spanish (Spain)
1080 Faeroese              1034 Spanish (Traditional Sort) -
Spain
                           14346 Spanish (Uruguay)
1035 Finnish               8202 Spanish (Venezuela)
2060 French (Belgium)      1089 Swahili
3084 French (Canada)       1053 Swedish
1036 French (France)       2077 Swedish (Finland)
5132 French (Luxembourg)   1114 Syriac
6156 French (Monaco)       1097 Tamil
4108 French (Switzerland)  1092 Tatar
1110 Galician              1098 Telugu
1079 Georgian              1054 Thai

3079 German (Austria)      1055 Turkish
1031 German (Germany)      1058 Ukrainian
5127 German (Liechtenstein) 1056 Urdu
4103 German (Luxembourg)   2115 Uzbek (Cyrillic)
2055 German (Switzerland)  1091 Uzbek (Latin)

1032 Greek                 1066 Vietnamese
1095 Gujarati
```

Example

-

Remarks

Copyright notice:
Note that any copyrighted or otherwise protected name or text is hereby used just for learning purposes and is property of the respective owner.

More Infos on MSI-Command-Line-Parameters can be found here (WEB-Link):

MSDN-Technet Language ID's

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

2.39.8 MSI - Parameters

MSI - Parameters

MSI - Parameters

Intention

```
Microsoft Windows Installer*.

Syntax
   Install
      MSIEXEC /i package options

   Uninstall
      MSIEXEC /x package options

   Advertise to current user
      MSIEXEC /ju package options [/t Transform_List | /g
LanguageID]

   Advertise to all users
      MSIEXEC /jm package options [/t Transform_List | /g
LanguageID]

   Administrative install - install on the network.
      MSIEXEC /a package

   Apply a patch to an installed Admin image
      MSIEXEC /p patchPKG /a package

Options:
   /fp          fix - replace missing files
   /fo          fix - replace Older files
   /fe          fix - replace older or Equal date files
   /fd          fix - replace Different version files
   /fc          fix - replace files based on Checksum
differences
   /fa          fix - replace All files
   /fu          fix - rewrite HKCU registry
   /fm          fix - rewrite HKLM registry
   /fs          fix - recreate shortcuts
   /fv          fix - rewrite local cache from source
   /l* Logfile  Log Everything (not Verbose)
   /l*v Logfile Log Everything Verbose
   /lv Logfile  Log Verbose
   /le Logfile  Log All error messages
   /lw Logfile  Log Non-fatal warnings
   /li Logfile  Log Status messages
   /la Logfile  Log Startup actions
   /lr Logfile  Log Actions
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
   /lu Logfile  Log User requests
   /lc Logfile  Log User Interface (UI) parameters
   /lm Logfile  Log memory use
   /lp Logfile  Log Terminal properties
   /l+ Logfile  Append to an existing log file.
   /l! Logfile  Clear an existing log file.
   /q , /qn  No UI.
   /qb       Basic UI.
   /qb!      Basic UI with no cancel button.
   /qr       Reduced UI. A modal dialog box is displayed at
the end of the install.
   /qf       Full UI. A modal dialog box is displayed at the
end of the install.
   /qn+      No UI. However, a modal dialog box is displayed
at the end of the installation.
   /qb+      Basic UI. A modal dialog box is displayed at the
end of the installation. If you cancel the installation, a
modal dialog box is not displayed.
   /qb-      Basic UI with no modal dialog boxes.
   /y module Register a DLL - only use for registry
information that cannot be added using the registry tables of
the .msi file.
   /z module UnRegister a DLL - only use for registry
information that cannot be removed using the registry tables
of the .msi file.
```

Example

Remarks

*Copyright notice:
Note that any copyrighted or otherwise protected name or text is hereby used just for learning purposes and is property of the respective owner.

More Infos on MSI-Command-Line-Parameters can be found here (WEB-Link):

MSDN-MSI Command-Line Options

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

-

© 2023 Theo Gottwald * https://www.smart-package.com

3. Script Language

3 3. Script Language

3.40 ! Smarty's Script language

MiniRobotLanguage (MRL)

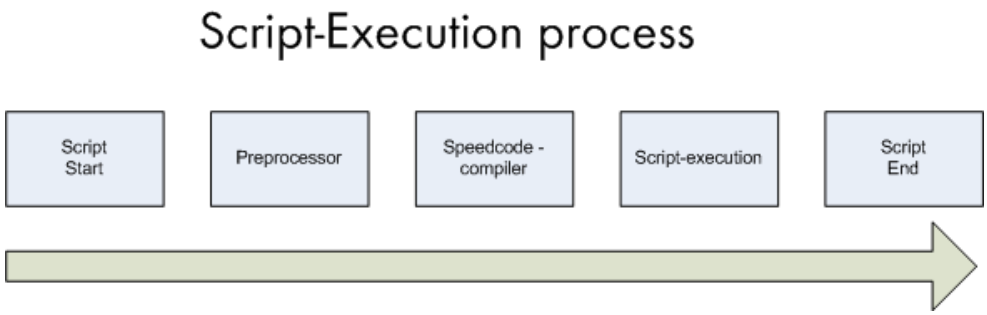
Smarty's Script language

The Script language is executed in stage 4 of the "Script-Execution process" (see picture below).

The Script language tells Smarty what to do. It will tell her to move the mouse around, to click on buttons. It will make her do key-presses for you and many more other things.

The Script language is the most important part for you to know.

If you know how to use the Script language, you will be able to get Smarty wake up and running!



Actually you can use 3 Sorts of Scripts:

1. SPR-Script
2. VBS. / JS - Script (See VBS. / VBE.)
3. Powershell-Script (See PWS. / PWE.)

Besides using the Commands, that are already defined for you, you can define own, new commands as ! User-Macros 399.

Sample-Script:

```
'#####################################################################
' Sicherheitspatch 4012212 installieren auf W7
' Version: 15.05.2017
' von: Andreas Walkenhorst
```

 © 2023 Theo Gottwald * https://www.smart-package.com

```
'#############################################################################
'#EXE:?path\
'#SPI:ForceWrite
'#FDC:Compiled by Andreas Walkenhorst
' programm direkt in das Paket einbinden
''#INC:?pfad\<PROGRAMM.EXE>

'EXE\?pfad
'MBX\!
'DWP\3
' Silent mode (keine fehlermeldungen vom Robot.
USM.3

'DIP.150
'#############################################################################
' Globale Variablen setzen
'#############################################################################
VAR.$$TXA=scripted by
VAR.$$TXB=Andreas Walkenhorst
VAR.$$TXP=Sicherheitspatch 4012212

' Aender  ?exeloc' zu '?pfad'
' wenn das SETUPFILE includiert ist
VAR.$$PFA=?exeloc

' Process ID
VAR.$$PID
' Programm
VAR.$$PRG=
' ProgrammLevel
VAR.$$PRO=10001
' Wartezeit
VAR.$$TIM=90
' Loggen 1=an 0=aus
VAR.$$LOA=1
' LogText
VAR.$$LOT=Programmstart
' LogZeile
VAR.$$LOZ $now$ $$LOT $crlf$
' LOGPFAD
VAR.$$LOP=c:\temp
' LOGFILE
VAR.$$LOF=$$LOP\spr_log_$$TXP.txt
' TESTMODE 1=an 0=aus
VAR.$$TES=0


' alle anderen Bots killen
MRK.2


' laufen wir im EXE modus? -> loggen
IEM.
' Set log directory
' EVG.LogDirectory>$$LOG
' Use silent mode with redirected error messages to error files
' USM.10|$$LOG\SetupErrorLog\SPR_Error.Log
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' 150 ms delay between each command
' DIP.150
EIF.

IVV.$$LOA=1
  $$LOT=$now$ Programmstart $$TXP $crlf$
  MKD.$$LOP
  CTF.$$LOF|$$LOT
EIF.

GSB.TXT_AN
'#############################################################################
'
GSB.MAIN
'GSB.MAILSEND
'
'#############################################################################
GSB.TXT_AUS


VAR.$$PRO=0
VAR.$$LOT=    sauber durchgelaufen
GSB.ausgang

PAU.1
GSB.MAILSEND
' Return code = 0 bei erfolgreich
SRV.0

ENR.


'#############################################################################
'
' Unterroutinen
'
'#############################################################################

:on_Error
' Wird nur im Fehlerfall angefahren
VAR.$$LOT=fehlerhaft durchgelaufen
GSB.LOGGEN
PAU.1
'GSB.MAILSEND
CPR.$$PID
' Fehlercode <> 0 zurückgeben
SRV.$$PRO

END.

:TXT_AN
'#############################################################################
' Text anzeigen
'#############################################################################
'VAR.$$TXA=scripted by
'VAR.$$TXB=Andreas Walkenhorst
'VAR.$$TXP=<ProgrammName>
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IVV.$$TES=1
  RET.
EIF.
VAR.$$PRO=
VAR.$$LOT=     TextCover an
GSB.LOGGEN

'fading ausschalten
UCV.nbi
'set farbverlauf hintergrund
UCV.gra|&HFF00FF|&HFFFF00
UCV.font set||64|1||
' Set black foreground color
UCV.fgr|&H040404
' text schreiben
UCV.write|$$TXP|50,100
PAU.250|ms
UCV.wri|$$TXA|150,300
PAU.250|ms
UCV.wri|$$TXB|200,400
RET.

:TXT_AUS
'#########################################################################
' Text ausschalten
'#########################################################################
IVV.$$TES=1
  RET.
EIF.
VAR.$$PRO=
VAR.$$LOT=     TextCover aus
GSB.LOGGEN
' This will remove the Cover
UCV.
RET.

:LOGGEN
§§TXT=$now$ $$PRO   $$LOT$crlf$
ATF.$$LOF|§§TXT
RET.

:MAIN
VAR.$$PRO=10000
VAR.$$LOT=Haben wir es mit WINDOWS 7 zu tun?
GSB.LOGGEN
NOS.Win7
  VAR.$$LOT=NICHT WINDOWS 7, somit sauber verlassen
  GSB.LOGGEN
  GTO.ausgang
EIF.

'
' Fesstellen ob der Patch schon vorhanden ist
GSB.patchchecker
VAR.$$PRO=10001
IVV.$$RES!0
  VAR.$$LOT=Patch bereits installiert, somit sauber verlassen
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GSB.LOGGEN
  GTO.ausgang
ELS.
  VAR.$$LOT=Patch muss noch installiert werden
  GSB.LOGGEN
EIF.

'#########################################################################
' No.:1 | Time: 07:58:39 | Ticks:  0
VAR.$$PRO=10002
VAR.$$LOT=Installtion nun starten
GSB.LOGGEN
' Das Skript ermittelt automatisch ob das Setup-Executable in das Paket eingebu
VAR.$$SEA=Sicherheitsqualitätsupdate für Windows 7 für x64-basierte Systeme (KB
GSB.LocateSetup
' Der komplette Pfad zum Setup ist nun in der Variablen $$SEB
EXO.$$SEB


'VAR.$$PRO=10003
'VAR.$$LOT=OK Button druecken
'GSB.LOGGEN
'STW.ct|#32770|Eigenständiges Windows Update-Installationsprogramm
'SCW.nictwo|1|2|Button|OK|5|weiter
' Possible Mouse-Click Commands are: MLE. | MLC. | MLM. | MLI. | BCS. | BCS.
'MLE.
'GTO.ausgang

PAU.5

No.:2 | Time: 07:59:14 | Ticks:  1076
VAR.$$PRO=10003
VAR.$$LOT=JA Button druecken
GSB.LOGGEN
STW.Hwct|172|#32770|Eigenständiges Windows Update-Installationsprogramm
SCW.wctid|172|Button|&Ja|6|1
SFW.
BCS.plts|240

PAU.5

' No.:3 | Time: 08:00:25 | Ticks:  3344
VAR.$$PRO=10004
VAR.$$LOT=Kein Neustart, nur schliessen
GSB.LOGGEN
STW.Hwct|362|#32770|Updates herunterladen und installieren
SCW.wctid|362|Button|Schließen|1|2
SFW.
BCS.plts|240

:ausgang
RET.

:patchchecker
'wmic qfe get hotfixid | find "KB4012212" > c:\temp\KB4012212.txt
$$HOT=KB4012212
$$TXT=C:\temp\$$HOT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$VAR=wmic qfe get hotfixid | find "$$HOT" > $$TXT
$$ZIE=c:\temp\$$HOT.bat
CTF.$$ZIE|$$VAR
EXC.$$ZIE
$$PID=#pid#
WPT.$$PID

FIL.Length|$$TXT|$$RES

IVV.$$RES!0
  VAR.$$EXI=bereits vorhanden
ELS.
  VAR.$$EXI=wird installiert
EIF.
RET.


:MAILSEND
GCN.$$PCN
SMP.smtp-server=192.168.44.12
SMP.Server-TimeOut=10
SMP.port=25
SMP.sender=$$PCN@test.de
SMP.receiver=helpdesk@test.de
SMP.subject=SMB Sicherheitspatch 4012212 $$EXI
SMP.text=$$PCN hat das Script gestartet
SMP.auth=Off
SMP.attachment=$$LOF
SMA.
RET.

'================================================
' Unterprogramme default vom Recorder
'================================================
' Stelle fest, wo das Setup-Executable sich befindet.
:LocateSetup
' Finde heraus, ob die Setup.exe im gleichen Ordner wie das Paket liegt.
VAR.$$SEB=?exeloc\$$SEA
IEF.$$SEB
  GTO.LeaveLocateSetup
EIF.

' Finde den Pfad zur Setup.exe Datei, egal ob diese in das Paket eingebunden is
VAR.$$SEB=?path\$$SEA
IEF.$$SEB
  GTO.LeaveLocateSetup
ELS.
  VAR.$$ERO=Setup.exe-Datei kann nicht gefunden werden. Bitte prüfen Sie die Sc
  GTO.on_Error
EIF.
:LeaveLocateSetup
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.41 ! User-Macros

3.41.1 ! User-Macros

MiniRobotLanguage (MRL)

Defining your own Commands: MACROS

Smarty's Script language can be extended using own Macros.

Besides using the Commands, that are already defined for you, you can define own, new commands.
You'll find this file also in the file "`Point_Convert-Lib.mrt`" in the User-Library Folder.

To use this Library in your Script you can copy the file into your Project folder and add the following Line at the end of your Script-Code (example):

```
#INC:Point_Convert-Lib.mrt
```

To make Using Macros more easy, use
2.1.A #LIB: - User-Library Path 298

and

1.1 #INC: - Pre-Processor File-Include 248

In this Chapter you will find examples of useful User-Macros that you can use to enhance your Scripts.

```
' Macro-Example:
' These Macros will implement a Time-FOR-Loop using Macros
' Please note the use of 'EIF- and 'EIF+ in the Macro for the Editor Formatting.
' This Loop will Loop from 08:09:59.000 to 08:10:01.000

$$TIA=08:09:59.000
$$TIB=08:10:01.000
'$$LOP=$$TIA
$$INC=00:00:00.010

%FOT $$TIA|$$TIB|$$INC
PRT.Looping: $$LOP
%NXT
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
MBX.$$LOP
ENR.

'----------------------------------
'  THe two Macros that implement a Time-Loop
'----------------------------------
: %FOT 3
$$FOA=§§§01
$$FOB=§§§02
$$FOC=§§§03
$$LOP=$$FOA
DOL.
'EIF-
END%
'----------------------------------
: %NXT
TCA.Add|$$LOP|$$INC|$$LOP
IVS.$$LOP=$$TIB
  $$LOV=1
ELS.
  $$LOV=0
EIF.
OOP.$$LOV
'EIF+
END%
'----------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.41.2 Firefox

Sample Code

Firefox-Library

These Subprogrammes are just Samples that i use to Click inside Firefox*-Browser
.
*For Details and Download see here: Firefox-Browser.

To use these Macros in your Script you may need to change some german Texts to fit to your your favourite language and Browser Version.

```
'++++++
```

```
' This will locate FF and run it
' If the window is too large (1840,1644) change size in Library
VAR.$$PAX="$$LNK" $$PAR
%Firefox $$PAX
WII.10

' This will open a second TAB-Page with that link
%New_TAB $$LNK

%Firefox_Set_Site $$LNK

' Now we have 2 TAB's from the same site in Firefox
' This will Locate TAB #1 in Firefox with the given Text
' Changing the number will locate the TAB #2 (with the same text)
' To locate another Tab you only need to change the text and use $$NUM=1
MBX.Now we locate TAB #1 (which has the same Text)
%Locate_TAB $$TAB|1

MBX.Now we locate TAB #2 (which has the same Text)

%Locate_TAB $$TAB|2
MBX.Now we locate TAB #1 (which has the same Text)

%Locate_TAB $$TAB|1

' Here you can do your CLicks on the Site

' Sample Click
GSB.Loc_TW
AGR.43!WII.MOS.|Nicht mehr abonnieren||1|3|60

' Here comes your Script
%Firefox_Close


ENR.
'============================================================
' Sub-Programmes
'============================================================
'
: %Locate_TW 0
GSB.Loc_TW
END%

:Loc_TW
NEW.$$HWN
  STW.ct|MozillaWindowClass|Mozilla Firefox
  SWP.0,0|1840,1644|T
  HTV.$$HWN
ELS.
  STW.h|$$HWN
EIF.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Kann ohne Parameter aufgerufen werden,
' oder mit einem Parameter mit dem Link für die neue Site
'
: %New_TAB 0
#IF PARAMS=1
SAV.Save|$$LNA
VAR.$$LNA=§§§01
GSB.New_TAB
PAU.1
%Firefox_Set_Site $$LNA
SAV.Restore
#EIF
#IF PARAMS=0
GSB.New_TAB
#EIF
END%
'-------------------------------------------------------------
:New_TAB
GSB.Loc_TW
AGR.43!WII.MLI.|Einen neuen Tab öffnen (Strg+T)||1|3|60
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
'
: %Firefox_Set_Site 1
SAV.Save|$$TXT
GSB.Set_Site|§§§01
SAV.Restore
END%
:Set_Site
VAR.$$TXT=§§_01
GSB.Loc_TW
AGR.43!WII.MLI.|Startseite||1|3|60
NAV.Rr|R|42
MLI.
KBC.Paste Over|$$TXT
SKP.{Enter}
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
'
' P1 - Text that is on the TAB we want
' P2 - optional the number of the TAB if it has the same Text like on other TABs
'  normally Nr. is 1
' The minimum Number of Parameters is 1 thats why we have a "1" here after The M
'
: %Locate_TAB 1
SAV.Save|$$NUM|$$TXT
VAR.$$TXT=§§§01
#IF PARAMS=1
VAR.$$NUM=1
GSB.Loc_TAB
#EIF
#IF PARAMS=2
VAR.$$NUM=§§§02
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Loc_TAB|§§§01
#EIF
SAV.Restore
END%
'-------------------------------------------------------------
:Loc_TAB
GSB.Loc_TW
AGR.42!WII.MOS.|$$TXT||$$NUM|3|60
GSB.Loc_TW
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
'
: %Firefox 1
GSB.Lab_Run_Firefox|§§§01
END%
'-------------------------------------------------------------
:Lab_Run_Firefox
SAV.Save|$$PAT|$$NAM|$$RES|$$EXE|$$LNK
'-------------------------------------------------------------
IAI.Mozilla Firefox
  VAR.$$LNK=§§_01
  GSB.Lab_FF_starten
ELS.
  MBX.Firefox ist nicht installiert.
  END.
EIF.
RET.

:Lab_FF_starten
VAR.$$NAM=$tos$
GII.$$NAM|$$RES|u
BLB.$$RES|$$RES
BLB.$$RES|$$RES
  ' $$PAT enthällt Programmpfad ohne "\"
VAR.$$PAT=$$RES
VBT.$$PAT|"
VAR.$$EXE=$$PAT\Firefox.exe
IEF.$$EXE
  EXE.$$EXE|$$LNK
ELS.
  MBX.Kann FF nicht starten.
  END.
EIF.
SAV.Restore
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
: %Firefox_Close 0
GSB.Lab_Close_FF
END%

:Lab_Close_FF
GSB.Loc_TW
CLW.$$HWN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|MozillaDialogClass|schließen?&OR:close
' AGR. uses a waiting time of 60 seconds, you can change it below.
AGR.43!WII.MOS.|Tabs schließen&OR:close||1|3|60
MLI.
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------

'-------------------------------------------------------------
'=============================================================
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.41.3 Graphical Subprograms

Sample Code

Graphics-Library

These Macro-Commands can be used to do Pixel-Processing.

This Command will Scan a Line in an Image-Register until a specified Color is found.
The specification is the Color as RGB and the Minimum and/or Maximum Color-Difference.
The Command will return wether -1 (="no such color found") or the X-Position of the found color.

```
VAN.$$COA=&HFF00FF
VAN.$$COB=&H00FF00
VAN.$$STO=100
VAN.$$CST=25
%Calculate_Color_between $$COA|$$COB|$$STO|$$CST
POP.$$RES
VTH.$$RES|$$REA|6
PRT.Erg. -- $$REA
MBX.!
ENR.
```

```
Result:
193.8 -- 61.2 -- 193.8
C23DC2
```

```
'-------------------------------------------------------------
'
'-------------------------------------------------------------
'
' Parameter:
' 1 - Number of the IR
' 2 - Start-Pos X
' 3 - Pos Y
' 4 - End-Pos X
' 5 - Color to Scan for
' 6 - Colordif - Max (or -1)
' 7 - Colordif - Min (or -1)
' If 6/7 do fit, result is returned on TOS
' 000 - X-Pos
' or -1 if not found
'
: %Find_Block_of_Color_X 6
SAV.Save|$$LOP|$$E01|$$E02|$$E03|$$E04|$$E05|$$E06|$$E07|$$ERM|$$ERC|$$ERD|$$ERA
#IF PARAMS<6
MBX.Not enough Parameter.
#EIF.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' IR - Number
VAN.$$E01=§§§01
' Startpos X
VAN.$$E02=§§§02
' Y-Pos
VAN.$$E03=§§§03
' X-Pos End
VAN.$$E04=§§§04
' Color to scan for
VAN.$$E05=§§§05
VAN.$$E06=§§§06
#IF PARAMS=6
VAN.$$E07=-1
#ELS
VAN.$$E07=§§§07
#EIF
'-------------------------------------------------------------
' The Loop-Initialization is necessary because we jump out of the Loop many time
VAN.$$LOP=$$E02
FOR.$$LOP|$$E02|$$E04
  ANA.GetPixelColor|$$E01|$$LOP|$$E03|$$C01
'-------------------------------------------------------------
  IVV.$$E06!-1
' This Command will return only the largest Color-Distance of the 3 values: Red,
    ANA.Color_Diff_Max|$$C01|$$E05|$$ERA
    IVV.$$ERA<$$E06
      VAN.$$ERC=1
    ELS.
      VAN.$$ERC=0
    EIF.
  ELS.
    VAN.$$ERC=1
  EIF.
'-------------------------------------------------------------
  IVV.$$E07!-1
' This Command will return only the largest Color-Distance of the 3 values: Red,
    ANA.Color_Diff_MIN|$$C01|$$E05|$$ERA
    IVV.$$ERA<$$E07
      VAN.$$ERD=1
    ELS.
      VAN.$$ERD=0
    EIF.
  ELS.
    VAN.$$ERD=1
  EIF.
'-------------------------------------------------------------
  CAL.$$ERM=$$ERC+$$ERD
  IVV.$$ERM=2
    VAN.$$XPA=$$LOP
    GTO.Leave_Macro_§§§98
  EIF.
NEX.
'-------------------------------------------------------------
VAN.$$XPA=-1
:Leave_Macro_§§§98
PUV.$$XPA
'-------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SAV.Restore
END%
'------------------------------------------------------------
'
'------------------------------------------------------------
'
' Parameter:
' 1 - Number of the IR
' 2 - Start-Pos X
' 3 - Pos Y
' 4 - End-Pos X
' 5 - Colordif - Min (or 0)
'
' If 5 matches, result is returned on TOS
' 000 - X-Pos End of Color-Block
' or -1 if not found
'
: %Find_End_of_Color_Block_X 4
SAV.Save|$$LOP|$$E01|$$E02|$$E03|$$E04|$$E05|$$E06|$$ERM|$$ERC|$$ERD|$$ERA|$$XPA
#IF PARAMS<4
MBX.Not enough Parameter.
#EIF.
' IR - Number
VAN.$$E01=§§§01
' Startpos X
VAN.$$E02=§§§02
' Y-Pos
VAN.$$E03=§§§03
' X-Pos End
VAN.$$E04=§§§04
' Maximum Color Diff to end Block
VAN.$$E05=§§§05
#IF PARAMS=4
VAN.$$E05=0
#ELS
VAN.$$E05=§§§05
#EIF
'------------------------------------------------------------
' Get Block Color
ANA.GetPixelColor|$$E01|$$E02|$$E03|$$E06
'------------------------------------------------------------
' The Loop-Initialization is necessary because we jump out of the Loop many time
VAN.$$LOP=$$E02
FOR.$$LOP|$$E02|$$E04
  ANA.GetPixelColor|$$E01|$$LOP|$$E03|$$C01

' This Command will return only the largest Color-Distance of the 3 values: Red,
  ANA.Color_Diff_Max|$$C01|$$E06|$$ERA
  IVV.$$ERA>$$E05
    VAN.$$XPA=$$LOP
    GTO.Leave_Macro_§§§98
  EIF.
NEX.
'------------------------------------------------------------
VAN.$$XPA=-1
:Leave_Macro_§§§98
PUV.$$XPA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
SAV.Restore
END%
'-------------------------------------------------------------
'
'-------------------------------------------------------------
' Calculate a Color between two Colors in steps
'
' 1 - Color A
' 2 - Color B
' 3 - Steps total
' 4 - Aktueller Step von 0 ... P3
' Resulting Color is on TOS 000
'
: %Calculate_Color_between 4
SAV.Save|$$E01|$$E02|$$E03|$$E04|$$E05|$$RST|$$GST|$$BST|$$RAA|$$GAA|$$BAA
#IF PARAMS<4
MBX.Not enough Parameter.
#EIF.
' Color A
VAN.$$E01=§§§01
' Color B
VAN.$$E02=§§§02
' Steps total
VAN.$$E03=§§§03
' Current Step, First step is "0"
VAN.$$E04=(§§§04-1)
IVV.$$E01=$$E02
  VAR.$$E05=$$E01
  GTO.Leave_Macro_§§§98
EIF.
'-------------------------------------------------------------
ANA.Split_RGB|$$E01|$$R01|$$G01|$$B01
ANA.Split_RGB|$$E02|$$R02|$$G02|$$B02

CAL.$$RST=($$R01-$$R02)/$$E03
CAL.$$GST=($$G01-$$G02)/$$E03
CAL.$$BST=($$B01-$$B02)/$$E03

CAL.$$RAA=$$RST*$$E04|a
IVV.$$R01>$$R02
  CAL.$$RAA=&HFF-$$RAA
EIF.

CAL.$$GAA=$$GST*$$E04|a
IVV.$$G01>$$G02
  CAL.$$GAA=&HFF-$$GAA
EIF.

CAL.$$BAA=$$BST*$$E04|a
IVV.$$B01>$$B02
  CAL.$$BAA=&HFF-$$BAA
EIF.

ANA.Combine_RGB|$$RAA|$$GAA|$$BAA|$$E05
'-------------------------------------------------------------
:Leave_Macro_§§§98
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PUV.$$E05
'-------------------------------------------------------------
SAV.Restore
END%
'-------------------------------------------------------------
'
'-------------------------------------------------------------
' Calculate a Color between two Pixels in Steps
' 1 - IR
' 2 - XPos - Pixel A
' 3 - XPos - Pixel B
' 4 - YPos
' 5 - Actual X-Pos
' Resulting Color is on TOS 000
'
: %Calculate_Color_between_X 4
SAV.Save|$$E01|$$E02|$$E03|$$E04|$$E05|$$C01|$$C02|$$STO|$$CST
#IF PARAMS<5
MBX.Not enough Parameter.
#EIF.
' IR
VAN.$$E01=§§§01
' XPos A
VAN.$$E02=§§§02
' XPos B
VAN.$$E03=§§§03
' YPos
VAN.$$E04=§§§04
' Current Pos
VAN.$$E05=§§§05
'-------------------------------------------------------------
ANA.GetPixelColor|$$E01|$$E02|$$E04|$$C01
ANA.GetPixelColor|$$E01|$$E03|$$E04|$$C02
' Error-Conditions
IVV.$$E01=$$E02
  GTO.Leave_MacroA_§§§98
EIF.
IVV.$$C01=$$C02
  GTO.Leave_MacroA_§§§98
EIF.
' Swap variables if needed (P1 needs to be > P2)
IVV.$$E01>$$E02
  VAR.$$E04=$$E01
  VAR.$$E01=$$E02
  VAR.$$E02=$$E04
EIF.
' Calc total Steps
CAL.$$STO=$$E03-$$E02
' Current Step
CAL.$$CST=$$E05-$$E02
%Calculate_Color_between $$C01|$$C02|$$STO|$$CST
POP.$$E05
'-------------------------------------------------------------
GTO.Leave_MacroB_§§§98
:Leave_MacroA_§§§98
VAR.$$E05=$$C01
:Leave_MacroB_§§§98
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'PRT.Leaving with $$E05
PUV.$$E05
'-------------------------------------------------------------
SAV.Restore
END%
'-------------------------------------------------------------
'
'-------------------------------------------------------------


'-------------------------------------------------------------
'
'-------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.41.4 Point-Convert-Lib

Sample Code

Point-Convert-Lib.mrt

These Macro-Commands can be used to convert Point and RECT Variables into each other.

You'll find this file also in the file "`Point_Convert-Lib.mrt`" in the User-Library Folder.

To use this Library in your Script you can copy the file into your Project folder and add the following Line at the end of your Script-Code:

```
#INC:Point_Convert-Lib.mrt
```

```
' Sample Call:
VAP.$$MYP=10,20
%Point_to_Vars $$MYP|$$XPA|$$YPA
MBX.$$XPA+$$YPA

'Sample Call:
VAC.$$REC=10,20,30,40
%Rect_to_Point $$REC|$$REA|$$REB
MBX.$$REA+$$REB

'Sample Call:
VAC.$$REC=10,20,30,40
%Rect_to_Vars $$REC|$$XPA|$$YPA|$$XPB|$$YPB
DBP.$$XPA+$$YPA+$$XPB+$$YPB


'++++++
```

```
GES.d|$$P01|,|$$P03
VAI.§§_02=$$P02
VAI.§§_03=$$P03
SAV.Restore
RET.

'
'-------------------------------------------------------------
' Convert RECT into 2 Point-Variables
'
'-------------------------------------------------------------

'VAC.$$REC=10,20,30,40
'%Rect_to_Point $$REC|$$REA|$$REB
'MBX.$$REA+$$REB
: %Rect_to_Point 3
VAO.§§§02
VAO.§§§03
GSB.Lab_RTP|§§§01|§§§02|§§§03
END%

:Lab_RTP
SAV.Save|$$P01|$$P02|$$P03
VAC.$$P01=§§_01
SBM.$$P01|,|1|2|$$P02
SBM.$$P01|,|3|4|$$P03
VAR.$$P01=,$sp$
VBT.$$P02|$$P01
VBT.$$P03|$$P01
VAI.§§_02=$$P02
VAI.§§_03=$$P03
SAV.Restore
RET.

'-------------------------------------------------------------
' Convert RECT into 4 Numeric-Variables
' 10,20,30,40 -> $$VA1,$$VA2,$$VA3,$$VA4
'-------------------------------------------------------------
'
: %Rect_to_Vars 5
VAO.§§§02
VAO.§§§03
VAO.§§§04
VAO.§§§05
GSB.Lab_RTV|§§§01|§§§02|§§§03|§§§04|§§§05
END%

:Lab_RTV
SAV.Save|$$P01|$$P02|$$P03|$$P04|$$P05|$$P06|$$P07|$$P08|$$P09|$$NUA
VAC.$$P09=§§_01
VAO.$$P01=§§_02
VAO.$$P02=§§_03
VAO.$$P03=§§_04
VAO.$$P04=§§_05
VAR.$$P07=,$sp$
VAN.$$NUA=1
GSB.Lab_RTVA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAI.$$P01=$$P05
VAI.$$P02=$$P06
VAN.$$NUA=3
GSB.Lab_RTVA
VAI.$$P03=$$P05
VAI.$$P04=$$P06
SAV.Restore
RET.

:Lab_RTVA
SBM.$$P09|,|$$NUA|($$NUA+1)|$$P08
VBT.$$P08|$$P07
%Point_to_Vars $$P08|$$P05|$$P06
RET.

'------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.41.5 Testing-Macros

Testing Library Previous 356 Top 41 Next 421

Sample Code

Testing-Lib.mrt

These Macro-Commands are just Samples that i use to test new features of the SPR automatically..

To use this Library in your Script you can copy the file into your Project folder and add the following Line at the end of your Script-Code.

You will also need the Label: `Lab_failed` somewhere in your Script.

```
#INC:Testing-Lib.mrt
```

```
'++++++
```

```
#EIF
END%
'-------------------------------------------------------------
' Goto mit Reset Return Stack
: %GRS 1
#IF PARAMS=1
RRS.
GTO.§§§01
#ELS
MBX.Benötige hier einen Parameter!
END.
#EIF
END%
'-------------------------------------------------------------
' JNZ mit Reset Return Stack
: %JNZ_RRS 2
#IF PARAMS=2
IVV.§§§01!0
  RRS.
  GTO.§§§02
EIF.
#ELS
MBX.Benötige hier zwei Parameter!
END.
#EIF
END%
'-------------------------------------------------------------
' JIV mit Reset Return Stack
: %JIV_RRS 3
#IF PARAMS=3
IVV.§§§01!§§§02
  RRS.
  GTO.§§§03
EIF.
#ELS
MBX.Benötige hier drei Parameter!
END.
#EIF
END%
'-------------------------------------------------------------
: %T_RunTestApp
GSB.T_RunTestApp
END%

:T_RunTestApp
SAV.Save|$$PAT
$$PAT=?exepath\TestApp_01.exe
NEF.$$PAT
  $$MSG=Kann $$PAT nicht starten.
  END.
EIF.
STW.Xctwg|PBWindowClass:0|TestApp|0|Lab_ova
EXE.$$PAT
:Lab_ova
STW.ct|PBWindowClass:0|TestApp
SWP.0,0
SAV.Restore
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.

'-------------------------------------------------------------
: %Loc_Top
GSB.Loc_Top
END%
:Loc_Top
IEW.$$MYT
  STW.h|$$MYT
ELS.
  STW.ct|PBWindowClass:0|TestApp
  HTV.$$MYT
EIF.
'SWP.0,0||T
SWP.1448,777||T
RET.
'-------------------------------------------------------------
: %T_CloseTestApp
GSB.T_CloseTestApp
END%

:T_CloseTestApp
STW.Xctwo|PBWindowClass:0|TestApp|0|Lab_ovb
CLW.
:Lab_ovb
STW.Xctwg|PBWindowClass:0|TestApp|0|T_CloseTestApp
SAV.Restore
RET.
'-------------------------------------------------------------
' Testet ob lokalisiertes Textfeld leer ist
: %T_Test_Empty
GSB.T_Test_Empty
END%

:T_Test_Empty
SAV.Save|$$RES
CFW.$$RES
JIS.$$RES|Lab_failed
SAV.Restore
RET.
'-------------------------------------------------------------
: %T_Test_Text 1
#IF PARAMS=1
GSB.T_Test_Text|§§§01
#ELS
MBX. Need 1 Parameter here!
END.
#EIF
END%

:T_Test_Text
SAV.Save|$$RES
CFW.$$RES
JIS.$$RES!§§_01|Lab_failed
SAV.Restore
RET.
'-------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
: %GRI_to_Time 1
#IF PARAMS=0
MBX.Need 1 Parameter here!
END.
#EIF
#IF PARAMS=1
GSB.GRI_to_Time|§§§01
#EIF
#IF PARAMS=2
GSB.GRI_to_Time|§§§01|§§§02
#EIF
#IF PARAMS>2
MBX.Need 1 Parameter here!
END.
#EIF
END%
'-------------------------------------------------------------
' Here is the Needed Subroutine
'-------------------------------------------------------------
' Wandelt die Rückgabe von GRI in eine Zeit um
:GRI_to_Time
SAV.Save|$$P01|$$P02|$$P03
VAR.$$P01=§§_01
SBD.$$P01|$sp$|2|$$P02
CAL.$$P03=$$P02/1656527/10/3
IVV.§§_00>1
  VAI.§§_02=$$P03
ELS.
  PUV.$$P03
EIF.
SAV.Restore
RET.

'-------------------------------------------------------------
' P1 - Label - Zeile darunter wird gemessen
' P2 - Variable for Result
'
: %T_GetLinetime 2
#IF PARAMS=2
GSB.T_GetLinetime|§§§01|§§§02
#ELSE
MBX.Need 2 Params here!
#EIF
END%

:T_GetLinetime
SAV.Save|$$P04|$$P05
VAO.$$P05=
GRI.§§_01|1|$$P04
%GRI_to_Time $$P04|§§_02
SAV.Restore
RET.
'-------------------------------------------------------------
' Ohne Parameter wird Editfeld 1023 lokalisiert, sonst das Feld mit der angegebe
: %Loc_Edit
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
#IF PARAMS=0
GSB.Loc_Edit
#EIF
#IF PARAMS=1
GSB.Loc_Edit|§§§01
#EIF
#IF PARAMS>1
MBX.More Params are not supported (§§§00)(§§§01)!
END.
#EIF
END%

:Loc_Edit
SAV.Save|$$PAA|$$PAB
VAN.$$PAA=§§_00
IVV.$$PAA=1
  $$PAB=§§_01
EIF.
%Loc_Top
IVV.$$PAA=1
  SCW.nic|1|$$PAB|Edit
ELS.
  IEW.$$MYC
    SCW.h|$$MYC
  ELS.
    SCW.nic|1|1023|Edit
    HTV.$$MYC
  EIF.
EIF.
SAV.Restore
RET.
'-------------------------------------------------------------
: %ShowText 1
GSB.ShowText|§§§01
END%
'-------------------------------------------------------------
:ShowText
MPO.0,0
SBT.$$_01|MP: T:5 FS:24 AX:100 AY:100
RET.
'-------------------------------------------------------------
' Clear Diagnose-Text in Testapp
: %Dia_Clear
GSB.Lab_Dia_Clear
END%

:Lab_Dia_Clear
SAV.Save|$$OLD|$$TXT
HTV.$$OLD
STW.ct|PBWindowClass:0|TestApp
SCW.nict|1|1000|Button|C
MLE.
GCT.
SCW.nic|1|1008|Edit
%T_Test_Empty
SCW.h|$$OLD
SAV.Restore
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'-------------------------------------------------------------
' Testet ob im Diagnosefenster der Text enthalten ist
: %Contains 1
GSB.Lab_Contains|§§§01
END%

:Lab_Contains
SAV.Save|$$OLD|$$TXT
HTV.$$OLD
STW.ct|PBWindowClass:0|TestApp
SCW.nic|1|1008|Edit
CFW.$$TXT
NVC.$$TXT=§§_01
  GTO.Lab_failed
EIF.
SCW.h|$$OLD
SAV.Restore
RET.
'-------------------------------------------------------------
: %Contains_Clear 1
%Contains §§§01
%Dia_Clear
END%
'-------------------------------------------------------------
: %Test_Coords 1
SAV.Save|$$REC|$$XAA|$$XAB|$$MAP
GES.b|§§§01|,|$$XAA
GES.d|§§§01|,|$$XAB
GSB.Test_Coords|$$XAA|$$XAB
SAV.Restore
END%

:Test_Coords
GMP.at|$$REC
DBP.$$REC
VAP.$$MAP=§§_01,§§_02
JIS.$$REC!$$MAP|Lab_failed
RET.
'-------------------------------------------------------------
ENR.
'=============================================================
#OEND
```

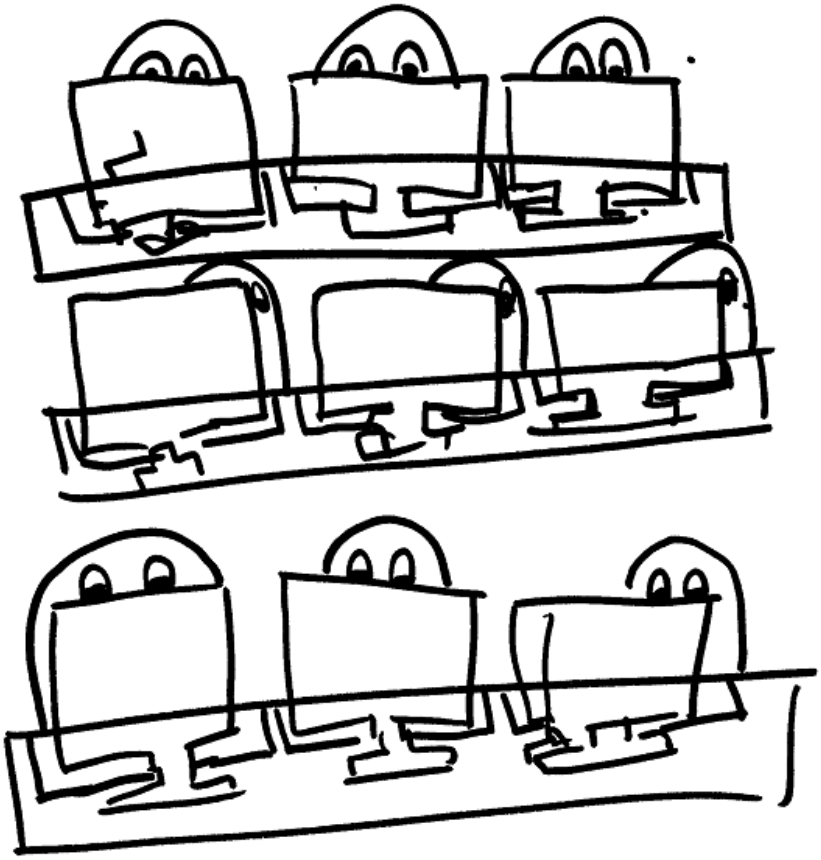

© 2023 Theo Gottwald * https://www.smart-package.com

3.42 Background Operation

3.42.1 ! Smarty's Background Operations

MiniRobotLanguage (MRL)

Smarty's Parallel Robot Operations



Intention

What are Parallel-Robot Operations and Background Processing good for?

Take a look at the picture above. As you can see, many "Smarty's" work at the same time.
Having several "Smarty's" work at the same time can save you processing time.

Especially if you have a Multi-Core-CPU, then each of the Tasks uses his own CPU-Core and therefore makes perfect use of your hardware.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Using the Parallel-Robot feature, you can run a lot of things in the background. Some examples are:

- start large applications in the background
- check and handle complicated pop-up-windows in the background (prefer `DBC.`)
- split large scripts in smaller portions that run parallel
- use all cores of your computer (Multi-Core/CPU usage)
- have things done in shorter time

Is there a difference between a Parallel-Robot and a Background Operation?

Yes, there is a technical difference. A Parallel Robot will run as a new process with an own PID. It will be visible in the Windows Task Manager as a separate process.
A Background Operation starts as a so called "thread" and therefore has the same `PID`.
Background Operations are more efficient in terms of resources.
There is no problem to have 100, 200 or 400 and maybe more Background Operations running at the same time.
Parallel running robots need more system resources therefore you will not use so many under normal circumstances. The main difference is this:
A Background Operation is already defined and somehow "hard-wired". Therefore it will give you an Operation Result - or not.
A Parallel Robot is not hard-wired, it will do just what you want it to do. It will only give you an Operation Result, if you use the `PRS.`-command. We'll come back on this topic later.

How can i start a Parallel-Robot?

It's easy. Technically it is just a pair of `PRR.` and `PRE.` A short example could look like this:

```
PRR.
  ' Script that the parallel-robot will execute
  ' This is the necessary END for the Parallel robot
  END.
PRE.$$RET
' Here the Main-Robot continues with his script
' $$RET contains the window-handle of the Parallel-Robot
```

There are different sorts of Background-Jobs and Parallel Robots. For most of them, you can use the commands:

- `WRS.` - Wait until the Process has ended / is ready.
- `GRS.` - Get the Result of that Bavkground process, if there is any.
- `DRS.` - Drop the result of a Background-Operation.
- `IRS. / NRS.` - Check if an Result is waiting for you
- `JRS. / JNS.` - Jump if an Result is/not available
- `PRS.` - Set the Result of an Parallel-Robot Operation.

Can i receive an result from calculations inside parallel running Robots?

Yes - you can!

© 2023 Theo Gottwald * https://www.smart-package.com

Second answer: In case of an Parallel-Robot you have to sent it your self - of course. In case of other Background-Operations, (like `SMB.`) the command will sent the result of the operation by itself automatically.

You can receive a calculation Result from a Parallel-Robot, using the Ticket-Number and the commands `WRS. / GRS.` etc. See this example:

```
' We start the Parallel Robot
PRR.
  ' This is the Result the Parallel-Robot sends back to us.
  ' Note that we did not specify an ticket number in P1,
  ' because it will be sent to his "father-robot" in this
case.
  PRS.|Hallo!
  ENR.
PRE.
' This is the Stack after the Parobot was started
STS.DUMP
' We wait for the Parobot to end.
WRS.
' And get the calculation result
GRS.
' This is the Stack after the Parobot sent his result
STS.DUMP
MBX.!
ENR.
```

To see, how you get an Result from a Background-Operation, take a look at

SMB. - Send Mail Background 799

or

MPB. - Music Play Background

Can i receive an result from multiple parallel running Robots?

Yes, here is an example:

```
PRR.
  GUT.Please give Result for Robot 1
  PRS.|Result from Robot 1 is: $$000
  ENR.
PRE.
$$ROA=$$001
PRR.
  GUT.Please give Result for Robot 2
  PRS.|Result from Robot 2 is: $$000
  ENR.
PRE.
$$ROB=$$001
STS.DUMP
WRS.$$ROA,$$ROB
GRS.$$ROA
GRS.$$ROB
STS.DUMP
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

You will need to put some text in the two input-boxes, when you run that example.

Important Note:

The parallel-robot will inherit all variables that the starting robot knows at the starting time.

In other words:
All variables and Stack-content will be available on the Parallel-Robot immediately. But from then each robot continues his work in his own address-space and with its own variables. Technically it looks like this:

```
$$MYA=1001
PRR.
  ' Script that the parallel-robot will execute
  ' This is the END for the Parallel robot
  MBX.$$MYA = 1001
  $$MYA=99
  END.
PRE.$$RET
' Here the Main-Robot continues with his script
' Result will be 1001
MBX.MYA
```

The Main Robot will not execute anything that is between the `PRR.` and the `PRE.`

And how do i know that the operation was finished and there is an Result available?

There are several ways to find that out. The easiest is to simply wait for the background Operation to finish, using the `WRS.`-command (Wait for Result). You can also use the `IRS. / NRS.` - Conditional statement.
And theoretically you could call `GRS.` and check the timeout-flag after that. If it was set, there was no such result available.

Important Note:
Any result behind an "ticket-number" is stored as long, until you drop it using `DRS.` You do not need to wait until the Operation is finished, to drop a result. If you do not need the result of an Operation, you can drop it immediately after you started the background-task. In this case, the robot will internally note that the result is been dropped and drop it immediately when it becomes available.

There is an exception:
Of course you only need to drop the Result of an Parallel-Robot (started with `PRR. / PRE.`) , if you use the `PRS,`-command to return a Result. If no Result is returned, it does not need to be dropped.

Use these commands together with any Background Operation or Parallel-Robot Operation:

- `GRS.` - Get Result of Operation
- `WRS.` - Wait until (one or more) Operation(s) is/are finished

© 2023 Theo Gottwald * https://www.smart-package.com

- DRS. - Drop Result of Operation if you do not need it.
- IRS. / NRS. Check if an Result is available
- JRS. / JNS. - Jump if an Result is/not available

Use this command:

- PRS. - Set the Result of an Parallel-Robot Operation.

only for Parallel Robots, to set the Operation Result..

Background Operation Example, using SMB.:

```
' Assume that a file "?path\MailConfig.dat"
' is available
SMP.
' It makes mostly sense with attachements
' SMP.attachment=?path\Send-Mail.zip
' Send the Mail, get the ticket number
SMB.
' If we do not specify a ticket number,
' WRS. will automatically remember the last ticket number
' and use it.
WRS.
' If we do not specify a ticket number,
' GRS. will automatically remember the last ticket number
' and use it.
GRS.
DBP.$$000
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
' and use it.
' DRS. free's the Memory location that was used to hold the
' Result of the background operation.
DRS.
ENR.
```

Limitations

The Parallel-Robot does NOT inherit the E-Mail Configuration for SMA./SMB. Remember that you can send E-Mails in the background using SMB. without explicit starting a Parallel Robots.

There are no limitations about Parallel-Robots starting other Parallel Robots, or Background-Tasks.

See also:

- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- GRS. - Get Result 799

© 2023 Theo Gottwald * https://www.smart-package.com

- SMB. Send Mail Background 799

© 2023 Theo Gottwald * https://www.smart-package.com

3.42.2 DRS. - Drop Result

MiniRobotLanguage (MRL)

DRS. - Drop Result

Drop Result from Background Operation

Intention

Using "Background Operations" means, that your Script execution can continue while the Robot is doing something in the background.

Some hardwired Background Operations store their Operation Results automatically when they are ready. If you do not need the Result, you should free that memory. This is, what the `DRS.` command is good for.

Any result behind an "ticket-number" is stored as long, until you drop it using `DRS.` You can apply the `DRS.` command immediately when you know the "Ticket Number" of the result.

You do not need to wait until the Operation is finished, to drop a result. If you do not need the result of an Operation, you can drop it immediately after you started the background-task. In this case, the robot will internally note that the result is been dropped and drop it immediately when it becomes available.

There is an exception:
Of course you only need to drop the Result of an Parallel-Robot (started with `PRR. / PRE.`) , if you use the `PRS,`-command to return a Result. If no Result is returned, it does not need to be dropped.

Use these commands together with Background Operations:

- `GRS.` - Get Result of Operation
- `WRS.` - Wait until (one or more) Operation(s) is/are finished
- `DRS.` - Drop Result of Operation if you do not need it.
- `IRS. / NRS.` Check if an Result is available
- `JRS. / JNS.` - Jump if an Result is/not available

to handle the result of the background mail-operation(s).

Example:

```
' Assume that a file "?path\MailConfig.dat"
' is available
SMP.
' It makes mostly sense with attachements
' SMP.attachment=?path\Send-Mail.zip
' Send the Mail, get the ticket number
SMB.
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' and use it.
DRS.
WRS.
ENR.
```

Syntax

DRS.[P1]

Parameter Explanation

P1 - ***(optional)*** Variable that contains the "Ticket-Number" which leads to the Return code of the background job. If omitted, the last Background Operation will be taken.

Example

```
'**********************************
' SMB.-Sample
'**********************************
' Send away 25 Mails
' then wait until they are sent
' and then show the results of all 25 Mails.
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Background Mailer-Demo
' Please provide a complete MailConfig.dat
SMP.?path\MailConfig.dat
' Try with attachement also!
' SMP.attachment=?path\Send-Mail.zip
$$HND=
' Be careful or you may end up with in
' an error 421 - too many connections
' from your provider
FOR.$$CNT|1|25
  GSB.Lab
NEX.
DBP.$$HND
WRS.$$HND
' Display Results
FEM.Lab_dis|$$TXT|$$HND|,
MBX.!
ENR.
'-------------------------------------------------------------
:Lab_dis
GRS.$$TXT|$$RES
' Always drop Results you do not need anymore
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DRS.$$TXT
DBP. Result from $$TXT = $$RES
RET.
'------------------------------------------------------------
:Lab
SMB.
$$HND=$$HND,$$000
RET.
'------------------------------------------------------------
```

Remarks

If you use Background Operations, always drop the result of the operation using `DRS..` Otherwise it will be kept in Memory until the Script ends. This only applies to background operations that do post an result to the main script.

Limitations:

-

See also:

- ! Smarty's Parallel Robot Operations 2632
- GRS. - Get Result 430
- DRS. - Drop Result 427
- WRS. - Wait for Result 436
- JRS. - Jump if Result 2215
- JNR. - Jump if no Result 2197
- IRS. / NRS. - If Result available 434
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- ! Smarty's Send-Mail 789
- SMB. Send Mail Background 799

© 2023 Theo Gottwald * https://www.smart-package.com

3.42.3 GRS. - Get Result

MiniRobotLanguage (MRL)

GRS. - Get Result

Get Result from Background Operation

Intention

Using "Background Operations" means, that your Script execution can continue while the Robot is doing something in the background.

How can I get the result of a background operation, if its running in the background?

The answer to this question is, that you get a "Ticket-Number", when you start the background operation. `SMB.` for example, will drop an "**ticket number**" on the TOS. Or, if **P2** is given, the ticket-number will be in the given variable.

Now you can get the result any time using the "**ticket number**" and the `GRS.`-command.
Just give `GRS.` the ticket-number, and in exchange you will get the result of the operation - when its already available.

And how do i know that the operation was finished and there is an Result available?

There are several ways to find that out. The easiest is to simply wait for the background Operation to finish, using the `WRS.`-command (Wait for Result). You can also use the `IRS. / NRS.` - Conditional statement.
And theoretically you could call `GRS.` and check the timeout-flag after that. If it was set, there was no such result available.

To wait for a background operation to be finished, the `WRS.`-command was made. It can also collect the Results of multiple background operations.

The system works like this:

1. Start a background operation and get a "ticket number"
2. keep the ticket number in a variable
3. do whatever you want ...
4. and call `WRS.` with that "ticket number" as parameter
5. `WRS.` will wait until the background Operation has finished
6. Call `GRS.` to get the result of the background operation

Important Note:
Any result behind an "ticket-number" is stored as long, until you drop it using `DRS.`
You do not need to wait until the Operation is finished, to drop a result. If you do not need the result of an Operation, you can drop it immediately after you started the

*© 2023 Theo Gottwald * https://www.smart-package.com*

background-task. In this case, the robot will internally note that the result is been dropped and drop it immediately when it becomes available.

There is an exception:
Of course you only need to drop the Result of an Parallel-Robot (started with `PRR. / PRE.`) , if you use the `PRS,`-command to return a Result. If no Result is returned, it does not need to be dropped.

Use these commands together with Background Operations:

- `GRS.` - Get Result of Operation
- `WRS.` - Wait until (one or more) Operation(s) is/are finished
- `DRS.` - Drop Result of Operation if you do not need it.
- `IRS. / NRS.` Check if an Result is available
- `JRS. / JNS.` - Jump if an Result is/not available

to handle the result of the background mail-operation(s).

Example:

```
' Assume that a file "?path\MailConfig.dat"
' is available
SMP.
' It makes mostly sense with attachements
' SMP.attachment=?path\Send-Mail.zip
' Send the Mail, get the ticket number
SMB.
' If we do not specify a ticket number,
' WRS. will automatically remember the last ticket number
' and use it.
WRS.
' If we do not specify a ticket number,
' GRS. will automatically remember the last ticket number
' and use it.
GRS.
DBP.$$000
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
' and use it.
' DRS. free's the Memory location that was used to hold the
' Result of the background operation.
DRS.
ENR.
```

Syntax

GRS.[P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - ***(optional)*** Variable that contains the "Ticket-Number" which leads to the Return code of the background job. If omitted, the last Background Operation will be taken.

P2 - ***(optional)*** Variable for the the Result of the background job. If omitted, TOS is used.

Example

```
'**********************************
' SMB.-Sample
'**********************************
' Send away 25 Mails
' then wait until they are sent
' and then show the results of all 25 Mails.
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Background Mailer-Demo
' Please provide a complete MailConfig.dat
SMP.?path\MailConfig.dat
' Try with attachement also!
' SMP.attachment=?path\Send-Mail.zip
$$HND=
' Be careful or you may end up with in
' an error 421 - too many connections
' from your provider
FOR.$$CNT|1|25
  GSB.Lab
NEX.
DBP.$$HND
WRS.$$HND
' Display Results
FEM.Lab_dis|$$TXT|$$HND|,
MBX.!
ENR.
'------------------------------------------------------------
:Lab_dis
GRS.$$TXT|$$RES
' Always drop Results you do not need anymore
DRS.$$TXT
DBP. Result from $$TXT = $$RES
RET.
'------------------------------------------------------------
:Lab
SMB.
$$HND=$$HND,$$000
RET.
'------------------------------------------------------------
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

If you use Background Operations, always drop the result of the operation using `DRS..` Otherwise it will be kept in Memory until the Script ends. This only applies to background operations that do post an result to the main script.

Limitations:

-

See also:

- ! Smarty's Parallel Robot Operations 2632
- GRS. - Get Result 430
- DRS. - Drop Result 427
- WRS. - Wait for Result 436
- JRS. - Jump if Result 2215
- JNR. - Jump if no Result 2197
- IRS. / NRS. - If Result available 434
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- ! Smarty's Send-Mail 789
- SMB. Send Mail Background 799
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.42.4 IRS. / NRS. - If Result available

MiniRobotLanguage (MRL)

IRS. Conditional Statement

If Result available

Intention

Conditional Statement.

Test if a specified Parallel-Robot or Background Operation has released a Result. `IRS.` does not really check if the Operation is ongoing. It will just check if there is a Result from that "Ticket Number" available.

`IRS.` will accept only one "Ticket Number" to test for the result. If you want to wait for multiple running Background- or Parallel Robots to finish, use `WRS.`

`IRS.` will drop the Result of the Ticket on the Stack, if there is one. Therefore you do not need to call `GRS.`, just take the result from the Stack.

You should anyway call `DRS.` to Drop the Result, when you do not need it anymore.

`NRS.` is the negative Form of this statement.

Syntax

IRS.[P1] ... ELS. ... EIF.

Parameter Explanation

P1 - (optional) Is the "Ticket Number" of a running Parallel Robot or background Operation. It can be omitted then the last "Ticket Number" that was emitted is been taken.

`IRS.` can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or `FEX.` (Enumerations) to unlimited Depth.

Example

```
'**********************************
' IRS.-Sample
'**********************************
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
SMP.
SMB.

:retry
IRS.
  DBV.Yes, we have the result! its $$000
  DRS.
ELS.
  PAU.1
  GTO.retry
EIF.
ENR.

'**********************************
' NRS.-Sample
'**********************************
SMB.
:Loop
'
NRS.
  PAU.0.1
  GTO.Loop
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- ! Smarty's Parallel Robot Operations 2632
- 1.5. Features and Hints 80
- GRS. - Get Result 430
- DRS. - Drop Result 427
- WRS. - Wait for Result 436
- JRS. - Jump if Result 2215
- JNR. - Jump if no Result 2197

© 2023 Theo Gottwald * https://www.smart-package.com

3.42.5 WRS. - Wait for Result

MiniRobotLanguage (MRL)

WRS. - Wait for Result

Wait for Result from Background Operation

Intention

This command will wait until:
- a Background Operation / Parallel-Robot has ended
- there is an incoming Result from an Operation / Parallelrobot
  - using for example `PRS.`

When waiting for an "Result", the command will still react if the given process ends.

Using "Background Operations" means, that your Script execution goes on, while the Robot is doing something in the background.

To wait for a background operation to be finished, the `WRS.`-command was made.
It can also collect the Results of multiple background operations.

The system works like this:
1. Start a background operation and get a "ticket number"
2. keep the ticket number in a variable
3. do whatever you want ...
4. and call `WRS.` with that "ticket number" as parameter
5. `WRS.` will wait until the background Operation has finished
6. Call `GRS.` to get the result of the background operation

Hint: If you have just one background operation running, you do not need to specify an ticket number. The system always remembers the and uses the ticket number from the last background operation.

Example:

```
' Assume that a file "?path\MailConfig.dat"
' is available
SMP.
' It makes mostly sense with attachements
' SMP.attachment=?path\Send-Mail.zip
' Send the Mail, get the ticket number
SMB.
' If we do not specify a ticket number,
' WRS. will automatically remember the last ticket number
' and use it.
WRS.
' If we do not specify a ticket number,
' GRS. will automatically remember the last ticket number
' and use it.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
GRS.
DBP.$$000
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
' and use it.
' DRS. free's the Memory location that was used to hold the
' Result of the background operation.
DRS.
ENR.
```

How do i know that the operation was finished and there is an Result available?

As said, you can use `WRS.` And that is maybe the best version. Besides `WRS.` , there are several more ways to find that out.
The easiest is to simply wait for the background Operation to finish, using the `WRS.`-command (Wait for Result).
You can also use the `IRS. / NRS.` - Conditional statement.
And theoretically you could call `GRS.` and check the timeout-flag after that.
If the timeout-flag was set after a call to `GRS.` it was set, there was no result available.

How do i collect results from multiple parallel running robots/operations?

Here is an example:

```
PRR.
  GUT.Please give Result for Robot 1
  PRS.|Result from Robot 1 is: $$000
  ENR.
PRE.
$$ROA=$$001
PRR.
  GUT.Please give Result for Robot 2
  PRS.|Result from Robot 2 is: $$000
  ENR.
PRE.
$$ROB=$$001
STS.DUMP
WRS.$$ROA,$$ROB
GRS.$$ROA
GRS.$$ROB
STS.DUMP
MBX.!
ENR.
```

You will need to put some text in the two input-boxes, this text is the result from each of the two parallel robot operations.
As you can see, the `WRS.` will wait until both Results are available. And then present them to you.

Here is an Example that shows how you can wait for 5 Parallel Processes to end.
Click on the 5 Messageboxes and see what happens.

```
 VAO.$$TIB=
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$LOP|1|5
' Run 5 Robots
  PRR.0
    MBX.!
    END.
  PRE.$$TIC
  ' Collect the Tickets separated with ","
  $$TIB=$$TIB,$$TIC
NEX.
' Wait until they end
WRS.!$$TIB
ENR.
```

Syntax

WRS.[!][P1[,..Px]][|P2]

Parameter Explanation

! - **(optional)** if specified, the command will NOT react on an Result, but wait until the specified process has ended.

P1 – ***(optional)*** Variable that contains the "Ticket-Number" which leads to the Return code of the background job. If omitted, the last Background Operation will be taken.

P2 – ***(optional)*** Timeout-Value. If omitted no timeout is used.

Example

```
'**********************************
' SMB.-Sample
'**********************************
' Send away 25 Mails
' then wait until they are sent
' and then show the results of all 25 Mails.
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Background Mailer-Demo
' Please provide a complete MailConfig.dat
SMP.?path\MailConfig.dat
' Try with attachement also!
' SMP.attachment=?path\Send-Mail.zip
$$HND=
' Be careful or you may end up with in
' an error 421 - too many connections
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' from your provider
FOR.$$CNT|1|25
  GSB.Lab
NEX.
DBP.$$HND
WRS.$$HND
' Display Results
FEM.Lab_dis|$$TXT|$$HND|,
MBX.!
ENR.
'-------------------------------------------------------------
:Lab_dis
GRS.$$TXT|$$RES
' Always drop Results you do not need anymore
DRS.$$TXT
DBP. Result from $$TXT = $$RES
RET.
'-------------------------------------------------------------
:Lab
SMB.
$$HND=$$HND,$$000
RET.
'-------------------------------------------------------------


'**********************************
' PRR.-Sample
'**********************************
PRR.0|Desktop-1
   PRR.0|Desktop-1
     PRS.Robot 2 Result
     END.
   PRE.$$RET
   WRS.$$RET
   GRS.$$RET|$$RES
   PRS.$$RES
   DRS.$$RET
   END.
 PRE.$$RET
 WRS.$$RET
 GRS.$$RET|$$RES
 MBX.Result: $$RES
 DRS.$$RET
 ENR.
```

Remarks

If you use Background Operations, always drop the result of the operation using `DRS..` Otherwise it will be kept in Memory until the Script ends. This only applies to background operations that do post an result to the main script.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

There is no technical limit in the number of background operations, that you can wait for with a single `WRS.` - Instruction.

See also:

- ! Smarty's Parallel Robot Operations 2632
- GRS. - Get Result 430
- DRS. - Drop Result 427
- JRS. - Jump if Result 2215
- JNR. - Jump if no Result 2197
- IRS. / NRS. - If Result available 434
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- ! Smarty's Send-Mail 789
- SMB. Send Mail Background 799

© 2023 Theo Gottwald * https://www.smart-package.com

3.43 BIT. - Work with Bits

3.43.1 ! Smarty's BIT. - Operations

General comments on BIT. Commands Previous 2356 Top 41 Next 2361

MiniRobotLanguage (MRL)

BIT. Command

Get, Set Clear or Toggle Bits and make Logic Operations with Bits

Intention

Using Operations with Single Bits, you can do a lot of stuff.

You can find out if specified Bits are set or not. And you can also manipulate Bits to have then set or not.
This can be used to analyze results from other operations, or you can use it - for example - to make your own Encryption algo.
Before we take a look on the BIT-Options, lets take a look on the required Parameters.

Whatever you want to do with BITs, generally there are two ways to specify Parameters for the Bit-Operations.
You can specify a number "16" as "Text" then it has two Bytes length. Byte 1 is "1" and the second Byte is "6" - we call this "by value".
You can also use the ASC-Code "16" and then only use 1 Byte to store the number. We call this "as binary".

1. Specify Parameter as VALUE

```
' To specify the Parameter "as VALUE" we can just write it as a number, using the
' 9 is 1001 in Binary
VAN.$$P01=9
' We want extract Bit "1" which is "1" in Result
VAN.$$P02=1
BIT.Get|v$$P01|$$P02|$$RES
' We change the Result to a readable Value using the ASC.-Command. This will only
' The result will be "1" as CHR-value (Binary). To make it readable we use the AS
ASC.$$RES|$$ERG
MBX.Result is $$ERG
```

This will give the result: "1", because the first bit in "1001" (dez. 9) is a "1".
In both cases, we did not specify P4 (Size-Parameter) so the default size of 8-bit was used.

2. Specify Parameter as BINARY

```
' To specify the Parameter "as Binary" we use the CHR. Command if the Parameter
' 9 is 1001 in Binary
CHR.9|$$P01
```

 © 2023 Theo Gottwald * https://www.smart-package.com

```
  ' We want extract Bit "1" which is "1" in Result
  VAN.$$P02=1
  BIT.Get|b$$P01|$$P02|$$RES
  ' We change the Result to a readable Value using the ASC.-Command. This will onl
  ' The result will be "1" as CHR-value (Binary). To make it readable we use the A
  ASC.$$RES|$$ERG
  MBX.Result is $$ERG
```

3. Specify multi-Byte Parameters "as Binary" or "as Value"

```
' Use the "ToBin" or the "ToVal" Option to convert the Parameters
VAN.$$P01=9
' $$P02 will contain the result
BIT.ToBin|$$P01|$$P02|2
' $$P02 will contain the result
BIT.ToVal|$$P01|$$P02|2
```

4. Specify multi-Byte Parameters using the "Mask" Option.

IMPORTANT:

5. Using the 64-bit Options / Why are the 64-bit operations unusable?

While there is a 64-Bit Mask Option implemented, and also other 64-bit Operations, we found during testing that the
SPR is currently unable to show exact results in many cases.
The problem happens, if a 64-bit number will evaluate to a number in Exponential writing during variable resolution.
For example "6.144567 E+18".
In such case the result number will be enough precise for all "normal" calculations, but the real "Bits" inside can differ from the expected result.
As you do not know in which cases this rounding will happen, most 64-bit operations will be useless at this time.
The problem appears when the SPR looks what is inside of a numeric variable and replaces it with the number that is inside.
Currently this is an decimal number and decimal numbers are not 100% precise at all times. And this is when the problem appears.
As an alternative, we have evaluated to change the variables of the SPR to NOT output decimal numbers but Hexadecimal numbers like **&HAAAAAAAAAAAAAAAAFFFFFFFFFFFFFFFF**, in such case the output would be unreadable for most people.
So we have not done this, instead 64-bit operations with bits are there but you should currently not use them because the results may be wrong.

For details on how to prefix Hexadecimal, Octal or Binary numbers, see
3.5 Hexadezimal, Decimal and Binary and more 352

Syntax

BIT.P1[|P2][|P3][|P4]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - Subcommand

P2 - Input Parameter

P3 -- Px Parameter depending on the Subcommand

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=67
VAN.$$P02=2
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Toggle|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 Toggle $$P02 is $$P03$crlf$Bin: $$ERA Toogle $$P02 -
> $$ERC
BIT.Toggle|$$P03|$$P02|$$P04|$$LEA
BIT.Show|v$$P04|$$ERD|$$LEA
MBX.$$P03 Toggle $$P02 is $$P04$crlf$Bin: $$ERC Toogle $$P02 -
> $$ERD
ENR.
```

Remarks

Please always use the proper prefix for non-decimal numbers.
More details on this, see here:
3.5 Hexadezimal, Decimal and Binary and more 352

Limitations:

Why are the 64-bit operations unusable and give wrong results?

While there is a 64-Bit Mask (8-Byte Option) implemented, and also other 64-bit Operations, we found during testing that the
SPR is currently unable to show exact results in many cases.
The problem happens, if a 64-bit number will evaluate to a number in Exponential writing during variable resolution.
For example "6.144567 E+18".
In such case the result number will be enough precise for all "normal" calculations, but the real "Bits" inside can differ from the expected result.
As you do not know in which cases this rounding will happen, most 64-bit operations will be useless at this time.

© 2023 Theo Gottwald * https://www.smart-package.com

The problem appears when the SPR looks what is inside of a numeric variable and replaces it with the number that is inside.
Currently this is an decimal number and decimal numbers are not 100% precise at all times. And this is when the problem appears.
As an alternative, we have evaluated to change the variables of the SPR to NOT output decimal numbers but Hexadecimal numbers like &HAAAAAAAAAAAAAAAAFFFFFFFFFFFFFFFF, in such case the output would be unreadable for most people.
So we have not done this, instead 64-bit operations with bits are there but
you should currently not use them because the results may be wrong.

```
' Maximum 60 bit Precision
VAN.$$P01=9223372036854775807
VAN.$$LEA=8
BIT.Show|$$P01|$$ERG|$$LEA
LEN.$$ERG|$$LEN
NVV.$$LEN=64
  GTO.ero
EIF.
NVS.$$ERG=0111111111111111111111111111111111111111111111111111111111111000
  GTO.ero
EIF.

' The last 4 Bits (right side) are not precide
VAN.$$P01=-9223372036854775807
VAN.$$LEA=8
BIT.Show|$$P01|$$ERG|$$LEA
LEN.$$ERG|$$LEN
NVV.$$LEN=64
  GTO.ero
EIF.
NVS.$$ERG=1000000000000000000000000000000000000000000000000000000000001000
  GTO.ero
EIF.
```

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.2 BIT.16to32 - concatenate 2 Words to 32 Bit DWord

MiniRobotLanguage (MRL)

BIT.16to32

concatenate 2 Words to 32 Bit DWord

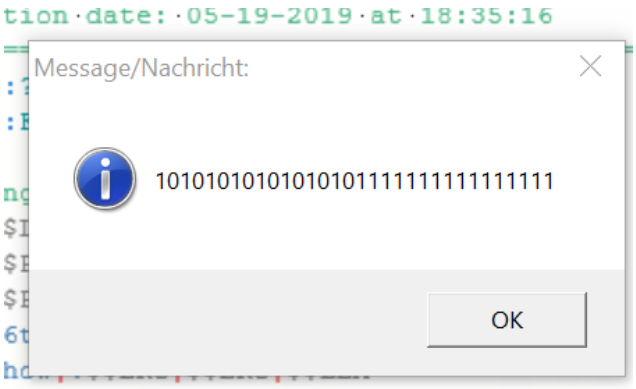
Intention

Concatenate two 16-Bit variables into One 32-Bit Variable.

The parameters **P1** and **P2** can be given as value or as "binary".
The result is a 32-bit variable.

You can define for all 3 Parameters if they should be in "Value" or "Binary Format.
For Details on these Parameter-Formats see
! Smarty's BIT. - Operations 441

```
' Using a value Number as Input
VAN.$$LEA=4
VAN.$$P01=&B1111111111111111
VAN.$$P02=&B1010101010101010
BIT.16to32|v$$P01|v$$P02|v$$ERG
BIT.Show|v$$ERG|$$ERG|$$LEA
NVS.$$ERG=10101010101010101111111111111111
  GTO.ero
EIF.
:ero
```



This picture shows the result of the Script.

```
' Having a Binary result, using "b"
VAN.$$LEA=4
VAN.$$P01=&B1111111111111111
VAN.$$P02=&B1010101010101010
BIT.16to32|v$$P01|v$$P02|b$$ERG
BIT.Show|b$$ERG|$$ERG|$$LEA
MBX.$$ERG
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

BIT.16to32|[b/v]P1|[b/v]P2[|P3]

Parameter Explanation

P1 **-** Variable containing a 16 bit numeric value.
You can define with a prefix of "v" for value and "b" for "binary" which format the value is.
If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a 16 bit numeric value.
You can define with a prefix of "v" for value and "b" for "binary" which format the value is.
If the prefix is omitted, "v" is the default setting for P1.

P3 - optional,Variable for result. If you use "-" the result is placed on the TOS. Also here you can specify [b/v] for the format of the result.
If you do not specify anything the result is "value"-Format.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
' Having a Binary result, using "b"
VAN.$$LEA=4
VAN.$$P01=&B1111111111111111
VAN.$$P02=&B1010101010101010
BIT.16to32|v$$P01|v$$P02|b$$ERG
BIT.Show|b$$ERG|$$ERG|$$LEA
MBX.$$ERG
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.3 BIT.16to8 - Split 1 Word into 2 Bytes

MiniRobotLanguage (MRL)

BIT.16to8

Split 1 Word into 2 Bytes

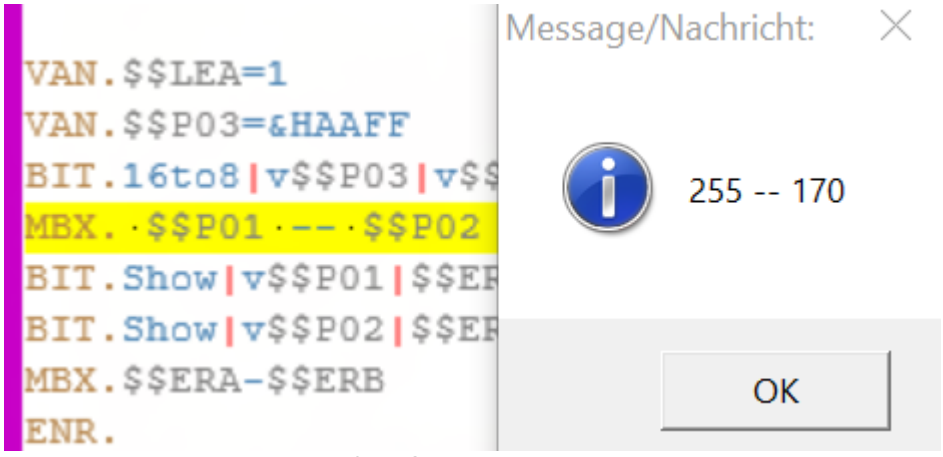
Intention

Split a 16-bit Variablöe in two 8-Bit Variables (High Byte and Low-Byte).

The parameters **P1** and **P2** can be given as value or as "binary".

You can define for all 3 Parameters if they should be in "Value" or "Binary Format.
For Details on these Parameter-Formats see
! Smarty's BIT. - Operations 441

```
' Here is a Sample Script and the result.
VAN.$$LEA=1
VAN.$$P03=&HAAFF
BIT.16to8|v$$P03|v$$P01|v$$P02
MBX. $$P01 -- $$P02
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
MBX.$$ERA-$$ERB
ENR.
```



This picture shows the result of the Script.

Syntax

BIT.16to8|[b/v]P1|[b/v]P2|[b/v]P3

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - Variable containing a value or binary number that is to be splitted into 2 parts.
You can define this with a prefix of "v" for value and "b" for "binary" the format of the Parameter. If the prefix is omitted, "v" is the default setting for ***P1***.

P2 - Variable that will contain the first Byte of the result. Unless "b" is prefixed the default type for ***P2*** is "value".

P3 - Variable that will contain the second Byte of the result. Unless "b" is prefixed the default type for ***P2*** is "value".

Example

```
'******************************************************
' EXAMPLE 1:
'******************************************************
VAN.$$LEA=1
VAN.$$P01=
VAN.$$P02=
VAN.$$P03=&HAAFF
BIT.16to8|v$$P03|v$$P01|v$$P02
MBX. $$P01 -- $$P02
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
MBX.$$ERA-$$ERB
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.4 BIT.32to16 - Split 1 DWord into 2 Words

MiniRobotLanguage (MRL)

BIT.32to16

Split 1 DWord into 2 Words

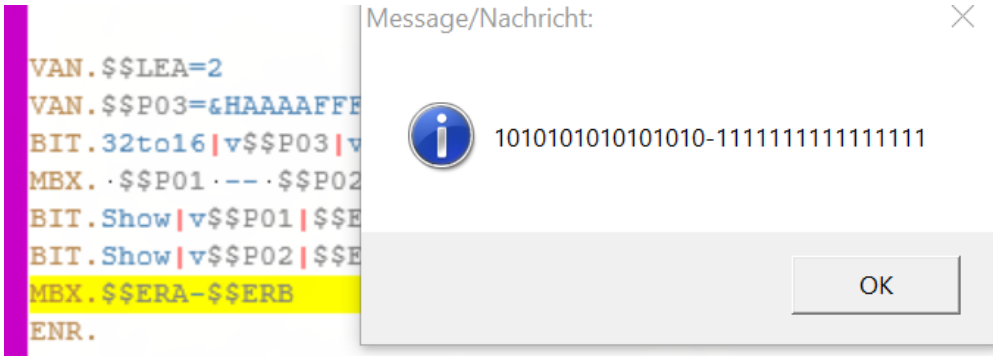
Intention

Split a 32-bit Variablöe in two 16-Bit Variables (High word and Low-Word).

The parameters P1 and P2 can be given as value or as "binary".

You can define for all 3 Parameters if they should be in "Value" or "Binary Format.
For Details on these Parameter-Formats see
! Smarty's BIT. - Operations 441

```
' Hee is a Sample-Script with Result:
VAN.$$LEA=2
VAN.$$P03=&HAAAAFFFF
BIT.32to16|v$$P03|v$$P01|v$$P02
MBX. $$P01 -- $$P02
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
MBX.$$ERA-$$ERB
ENR.
```



Syntax

BIT.32to16|[b/v]P1|[b/v]P2|[b/v]P3

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 - Variable containing a value or binary number that is to be splitted into 2 parts.
You can define this with a prefix of "v" for value and "b" for "binary" the format of the Parameter. If the prefix is omitted, "v" is the default setting for P1.

P2 - Variable that will contain the first Word of the result. Unless "b" is prefixed the default type for P2 is "value".

P3 - Variable that will contain the second Word of the result. Unless "b" is prefixed the default type for P2 is "value".

Example

```
'*****************************************************
' EXAMPLE 1
'*****************************************************
VAN.$$LEA=2
VAN.$$P03=&HAAAAFFFF
BIT.32to16|v$$P03|v$$P01|v$$P02
MBX. $$P01 -- $$P02
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
MBX.$$ERA-$$ERB
ENR.
```

Remarks
-

Limitations:
-

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.43.5 BIT.8to16 - concatenate 2 Bytes to 16 Bit Word

MiniRobotLanguage (MRL)

BIT.8to16

concatenate 2 Bytes to 16 Bit Word

Intention

Concatenate two 8-Bit variables into one 16-Bit Variable.

The parameters **P1** and **P2** can be given as value or as "binary".
The result is a 32-bit variable. You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Please see ! Smarty's BIT. - Operations 441

You can define for all 3 Parameters if they should be in "Value" or "Binary Format.

```
VAN.$$LEA=2
VAN.$$P01=&HAA
VAN.$$P02=&HFF
BIT.8to16|$$P01|$$P02|$$P03
MBX. $$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$ERA
ENR.
```

Syntax

BIT.8to16|[b/v]P1|[b/v]P2[|[b/v]P3]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - Variable containing a 8 bit numeric value.
You can define with a prefix of "v" for value and "b" for "binary" which format the value is.
If the prefix is omitted, "v" is the default setting for P1.

P2 - Variable containing a 8 bit numeric value.
You can define with a prefix of "v" for value and "b" for "binary" which format the value is.
If the prefix is omitted, "v" is the default setting for P1.

P3 - optional,Variable for 16 bit result. If you use "-" the result is placed on the TOS. Also here you can specify [b/v] for the format of the result.
You can choose whether P3 will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.

Example

```
'******************************************************
' EXAMPLE 1:
'******************************************************
VAN.$$LEA=2
VAN.$$P01=&HAA
VAN.$$P02=&HFF
BIT.8to16|$$P01|$$P02|$$P03
MBX. $$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$ERA
ENR.END.
```

Remarks

-

Limitations:

-

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.6 BIT.AND - Use logical AND with Bits

MiniRobotLanguage (MRL)

BIT.AND

Use logical AND with Bits

Intention

AND returns TRUE (non-zero) if (and only if) the Bits in P1 AND P2 are "1".
Therefore AND masks clear selected bits of P1 without affecting the other bits.
For example, to clear the most-significant (leftmost) 2 bits in the integer value &H9700, AND it with &H3FFF.
That is, the mask contains all 1s, except for the bit positions you want to force to 0:

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit.

Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=&H1
VAN.$$P02=129
BIT.AND|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 AND $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```

Syntax

```
BIT.AND|[b/v]P1|[b/v]P2[|P3][|P4]
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**.
If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=&H1
VAN.$$P02=129
BIT.AND|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 AND $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.7 BIT.Clear - Set Bit to 0

MiniRobotLanguage (MRL)

BIT.Clear

Set Bit to 0

Intention

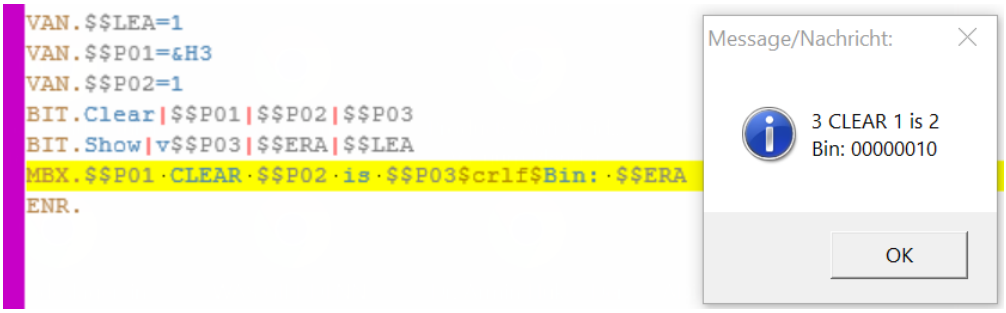
CLEAR will set a defined Bit in **P1** to "0".

For example, if you have the number "3" and you set Bit 1 to "0" the result will be "2".

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit.

Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=&H3
VAN.$$P02=1
BIT.Clear|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 CLEAR $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```



Syntax

BIT.CLEAR|[b/v]P1|[b/v]P2[|[b/v]P3][|P4]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number. **P2** is the number of the Bit to Clear.Counting is 1-based and bit 1 has the value 1, Bit 8 would be 128 for example.
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**.
If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=&H3
VAN.$$P02=1
BIT.Clear|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 CLEAR $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.8 BIT.Get - check if Bit is set

MiniRobotLanguage (MRL)

BIT.Get - check if Bit is set

Check if bit is set

Intention

Check if a specified Bit is set or cleared.

The parameters **P1** and **P2** can be given as value or as "binary".
The result is always "0" or "1" as a value. Please see ! Smarty's BIT. - Operations 441

These two example below will check if Bit Nr.1 is set in the binary number "9" which is "1001".
Therefore the result will be a "1" in both Samples.

```
' Using a binary Number as Input
CHR.9|$$P01
VAN.$$P02=1
BIT.Get|b$$P01|$$P02|$$RES
MBX.$$RES

' Using a vale/numeric Number as Input
VAN.$$P01=9
VAN.$$P02=1
BIT.Get|$$P01|$$P02|$$RES
MBX.$$RES
```

Syntax

BIT.Get|[b/v]P1|[b/v]P2|P3[|P4]

Parameter Explanation

P1 - Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 - Variable containing a value or binary number for the bit to get. Depending on **P4** this number can be 1 to 8, 1 to 16 or

*© 2023 Theo Gottwald * https://www.smart-package.com*

1 to 32. Unless "b" is prefixed the default type for **P2** is "value".

P3 - Variable for result or "-". The result will always be numeric/value and can only be 1 or 0.
If omitted the result is placed on the TOS. Also here you can specify [b/v] for the format of the result. If you do not specify anything the result is "value"-Format.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$P01=9
VAN.$$P02=4
VAN.$$SOL=1
BIT.Get|v$$P01|$$P02|$$ERG
MBX.$$ERG
END.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.9 BIT.LShift - Shift Bits to Left, insert 0

MiniRobotLanguage (MRL)

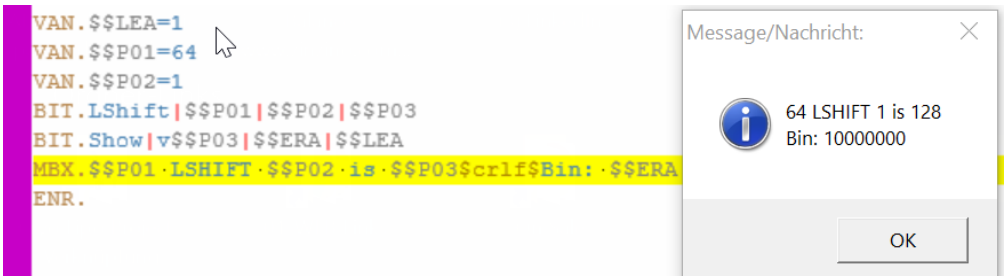
BIT.LShift - Shift Bits to Left, insert 0

Shift Bits to the Left side, insert 0 at the right side

Intention

LSHIFT shifts all the bits in **P1** inserting a "0" at the right side. In general a LShift equals an Integer multiplication by 2.
The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit and also how many bits are shifted.
Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=64
VAN.$$P02=1
BIT.LShift|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 LSHIFT $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```



Syntax

BIT.LShift|[b/v]P1|[b/v]P2[|[b/v]P3][|P4]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
This number contains the Bits "to be shifted".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number. This number is the amount of bits to "Shift Left".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**. If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=64
VAN.$$P02=1
BIT.LShift|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 LSHIFT $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.10 BIT.NOT - Use logical NOT with Bits

MiniRobotLanguage (MRL)

BIT.NOT - Use logical NOT with Bits

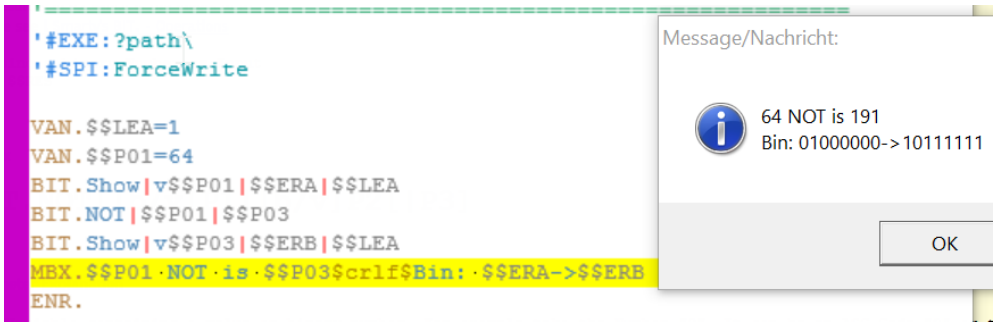
Invert Bits

Intention

NOT generally inverts bit, it makes a 0 to a 1 and makes a 1 to a 0.

The parameters **P1** can be given as value or as "binary".
The result is in **P2** which is the invers of P1.
Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=64
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.NOT|$$P01|$$P03
BIT.Show|v$$P03|$$ERB|$$LEA
MBX.$$P01 NOT is $$P03$crlf$Bin: $$ERA->$$ERB
ENR.
```



Syntax

BIT.NOT|[b/v]P1|[b/v]P2[|P3]

Parameter Explanation

P1 - Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - P2 - Variable for result or"-". If P2 is not a variable, the result is placed on the TOS.
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.
Also here you can specify [b/v] for the format of the result. If you do not specify anything the result is "value"-Format.

P3 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.

1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=64
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.NOT|$$P01|$$P03
BIT.Show|v$$P03|$$ERB|$$LEA
MBX.$$P01 NOT is $$P03$crlf$Bin: $$ERA->$$ERB
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.11 BIT.OR - Use logical OR with Bits

MiniRobotLanguage (MRL)

BIT.OR - Use logical OR with Bits

Use logical OR with Bits

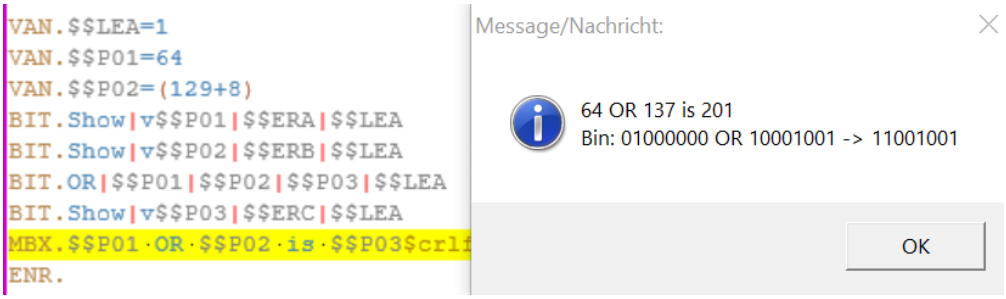
Intention

OR returns TRUE (non-zero) if any of the Bits in P1 AND P2 are "1".
Therefore OR will make sure that the given bits are set to 1.

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit.

Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=64
VAN.$$P02=(129+8)
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.OR|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 OR $$P02 is $$P03$crlf$Bin: $$ERA OR $$ERB -> $$ERC
ENR.
```



Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

BIT.OR|[b/v]P1|[b/v]P2[|[b/v]P3][|P4]

Parameter Explanation

P1 - Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 - Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**.
If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=64
VAN.$$P02=(129+8)
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.OR|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 OR $$P02 is $$P03$crlf$Bin: $$ERA OR $$ERB -> $$ERC
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.

Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.12 BIT.RotateLeft - Rotate Bits Left

BIT.RotateLeft - Rotate Bits to Left Previous 2350 Top 41 Next 2354

MiniRobotLanguage (MRL)

BIT.RotateLeft - Rotate Bits to Left

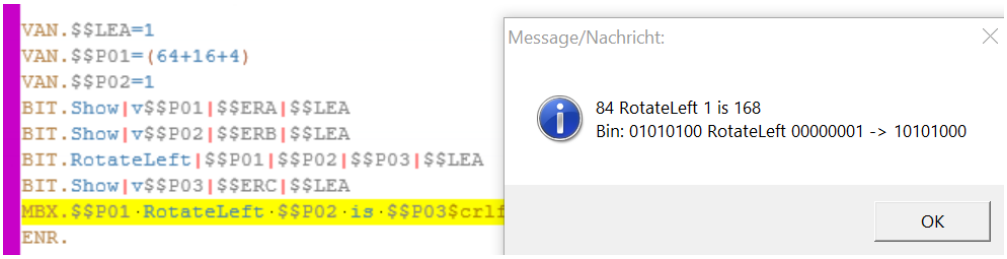
Rotate Bits to the Left side, insert what falls out at the right side

Intention

RotateLeft shifts all the bits in **P1** inserting all bits that fall out at the right side, are move in ton the left side.
In general a RotateLeft somehow equals an Integer multiplication by 2.

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit and also how many bits are shifted.
Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=1
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.RotateLeft|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 RotateLeft $$P02 is $$P03$crlf$Bin: $$ERA RotateLeft $$ERB -> $$ERC
ENR.
```



Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

BIT.RotateLeft|[b/v]P1|[b/v]P2[|[b/v]P3][|P4]

Parameter Explanation

P1 - Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
This number contains the Bits "to be shifted".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 - Variable containing a value or binary number. This number is the amount of bits to "RotateLeft".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**. If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=1
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.RotateLeft|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 RotateLeft $$P02 is $$P03$crlf$Bin: $$ERA RotateLeft $$ERB -> $$ERC
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.13 BIT.RotateRight - Rotate Bits Right

BIT.RotateLeft - Rotate Bits to Left **Previous** 2350 **Top** 41 **Next** 2354

MiniRobotLanguage (MRL)

BIT.RotateRight - Rotate Bits to Left

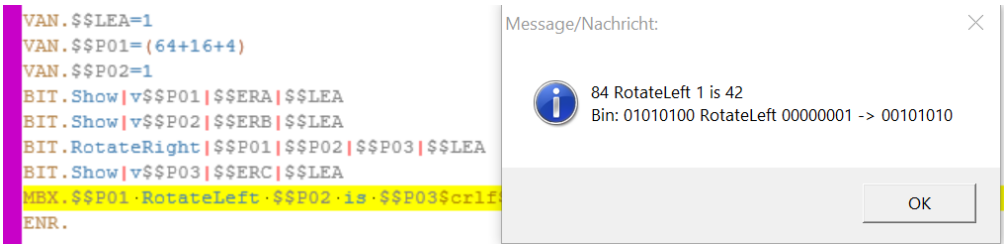
Rotate Bits to the Left side, insert what falls out at the right side

Intention

RotateRight shifts all the bits in **P1** inserting all bits that fall out on teh right side, back in at the left side.
In general a RotateRight somehow equals an Integer multiplication by 2.

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit and also how many bits are shifted.
Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=1
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.RotateRight|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 RotateRight $$P02 is $$P03$crlf$Bin: $$ERA RotateRight $$ERB -> $$ERC
ENR.
```



Syntax

BIT.RotateRight|[b/v]P1|[b/v]P2[|[b/v]P3][|P4]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
This number contains the Bits "to be shifted".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number. This number is the amount of bits to "RotateRight".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**. If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=1
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.RotateRight|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 RotateRight $$P02 is $$P03$crlf$Bin: $$ERA RotateRight $$ERB -> $$ERC
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.14 BIT.RShift - Shift Bits to Right, insert 0

MiniRobotLanguage (MRL)

BIT.RShift - Shift Bits to Right, insert 0

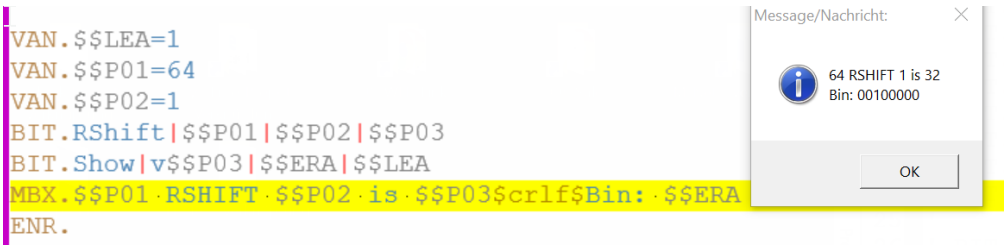
Shift Bits to the Right side, insert 0 at the right side

Intention

RSHIFT shifts all the bits in **P1** inserting a "0" at the left side. In general a RShift equals an Integer division by 2.

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit and also how many bits are shifted.
Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=64
VAN.$$P02=1
BIT.RShift|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 RSHIFT $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```



Syntax

BIT.RShift|[b/v]P1|[b/v]P2[|[b/v]P3][|P4]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
This number contains the Bits "to be shifted".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 - Variable containing a value or binary number. This number is the amount of bits to "Shift Right".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result.
If the prefix is omitted the result will have the format of **P1**. If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=64
VAN.$$P02=1
BIT.RShift|$$P01|$$P02|$$P03
BIT.Show|v$$P03|$$ERA|$$LEA
MBX.$$P01 RSHIFT $$P02 is $$P03$crlf$Bin: $$ERA
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.15 BIT.Set - Set Bit to 1

MiniRobotLanguage (MRL)

BIT.Set - Set Bit to 1

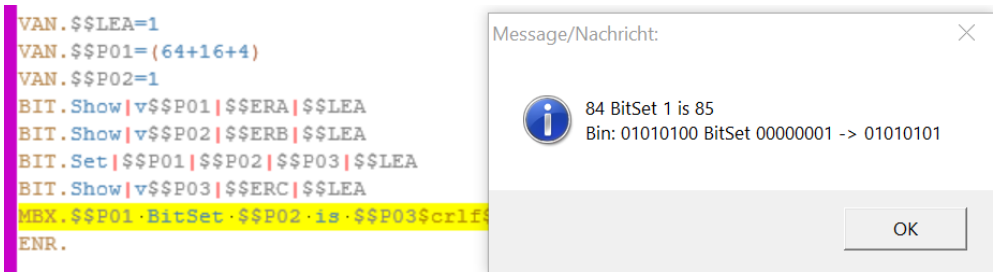
Check if bit is set

Intention

Check if a specified Bit is set or cleared. Technically Bit.Set is somehow equal to an OR.
But more easy to use with Single Bits.

The parameters **P1** and **P2** can be given as value or as "binary".
Please see details on v/b here ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=1
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.Set|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 BitSet $$P02 is $$P03$crlf$Bin: $$ERA BitSet $$ERB -> $$ERC
ENR.
```



Syntax

BIT.Set|[b/v]P1|[b/v]P2|[b/v]P3[|P4]

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number for the bit to set. Depending on **P4** this number can be 1 to 8, 1 to 16 or 1 to 32. Unless "b" is prefixed the default type for **P2** is "value".

P3 **-** Variable for result or "-". The result will always be numeric/value and can only be 1 or 0.
If omitted the result is placed on the TOS. Also here you can specify [b/v] for the format of the result. If you do not specify anything the result is "value"-Format.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=1
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.Set|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 BitSet $$P02 is $$P03$crlf$Bin: $$ERA BitSet $$ERB -
> $$ERC
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.16 BIT.Show - Show Bits in Binary-Numbers

MiniRobotLanguage (MRL)

BIT.SHOW - Show binary Bits

Show Bits

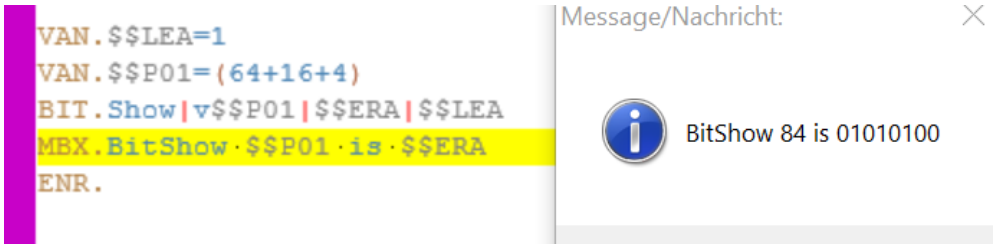
Intention

SHOW will convert a Number into Binary format.

The parameters **P1** can be given as value or as "binary".
The result is in **P2** which is the binary version of P1 "as Text".

Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
BIT.Show|v$$P01|$$ERA|$$LEA
MBX.BitShow $$P01 is $$ERA
ENR.
```



Syntax

BIT.SHOW|[b/v]P1|P2[|P3]

Parameter Explanation

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".

You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - Variable for result or "-". You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.
    If omitted the result is placed on the TOS. Also here you can specify [b/v] for the format of the result. If you do not specify anything the result is "value"-Format.

P3 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
     1 - 8 Bits
     2 - 16 Bits
     4 - 32 Bits

     Do not use the "8" Option. See Limitations below.

Example

```
'******************************************************
' EXAMPLE 1:
'******************************************************
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
BIT.Show|v$$P01|$$ERA|$$LEA
MBX.BitShow $$P01 is $$ERA
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.
Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.17 BIT.Store - Set Bit to 1 or 0

BIT.Store - Set Bit to 1 or 0 **Previous** 2350 **Top** 41 **Next** 2354

MiniRobotLanguage (MRL)

BIT.Store - Set Bit to 1 or 0

Set Bit to 1 or to 0

Intention

Technically `BIT.Store` will set or clear a specified Bit in **P1**, depending on **P3** is 0 or 1.

The parameters **P1** and **P2** can be given as value or as "binary".
The result is in **P1**. Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=2
VAN.$$P03=1
VAN.$$P04=$$P01
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Store|v$$P01|v$$P02|$$P03|$$LEA
BIT.Show|v$$P01|$$ERC|$$LEA
MBX.$$P04 Store $$P02/$$P03 is $$P01$crlf$Bin: $$ERA Store $$P02/$$P03 -> $$ERC
VAN.$$P04=$$P01
VAN.$$P03=0
BIT.Store|v$$P01|v$$P02|$$P03|$$LEA
BIT.Show|v$$P04|$$ERB|$$LEA
BIT.Show|v$$P01|$$ERC|$$LEA
MBX.$$P04 Store $$P02/$$P03 is $$P01$crlf$Bin: $$ERB Store $$P02/$$P03 -> $$ERC
ENR.
```

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

BIT.Store|[b/v]P1|[b/v]P2|P3[|P4]

Parameter Explanation

P1 **-** Variable containing a value or binary number, will also contain the result. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number for the bit to get. Depending on **P4** this number can be 1 to 8, 1 to 16 or 1 to 32. Unless "b" is prefixed the default type for **P2** is "value".

P3 **-** Variable or "0" or "1". P3 will decide whether the Command will "set" or "clear" the specified Bit.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.

1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=(64+16+4)
VAN.$$P02=2
VAN.$$P03=1
VAN.$$P04=$$P01
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Store|v$$P01|v$$P02|$$P03|$$LEA
BIT.Show|v$$P01|$$ERC|$$LEA
MBX.$$P04 Store $$P02/$$P03 is $$P01$crlf$Bin: $$ERA Store
$$P02/$$P03 -> $$ERC
VAN.$$P04=$$P01
VAN.$$P03=0
BIT.Store|v$$P01|v$$P02|$$P03|$$LEA
BIT.Show|v$$P04|$$ERB|$$LEA
BIT.Show|v$$P01|$$ERC|$$LEA
MBX.$$P04 Store $$P02/$$P03 is $$P01$crlf$Bin: $$ERB Store
$$P02/$$P03 -> $$ERC
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.

Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.18 BIT.ToBin - change Parameter to Binary-Format

MiniRobotLanguage (MRL)

BIT.ToBin - Convert Parameter from "Value" to "Binary"

Convert Parameter to "Binary"

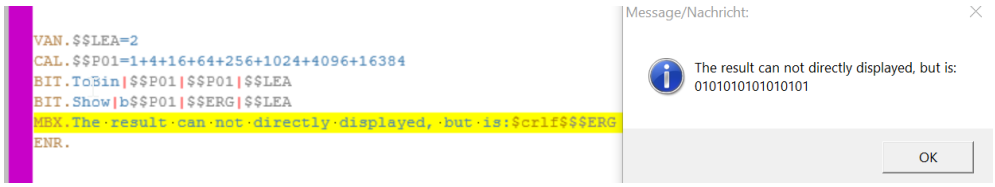
Intention

Generally there are two types od Parameters, these are "as value" and "as binary".

Whatever you want to do with BITs, generally there are two ways to specify Parameters for the Bit-Operations.
You can specify a number "16" as "Text" then it has two Bytes length. Byte 1 is "1" and the second Byte is "6" - we call this "by value".
You can also use the ASC-Code "16" and then only use 1 Byte to store the number. We call this "as binary".
More explanation is below under Remarks and see ! Smarty's BIT. - Operations 441

The parameters **P1** is the "Input" and it is "as Value".
The result is in **P2** which is the "as binary" of **P1**.

```
VAN.$$LEA=2
CAL.$$P01=1+4+16+64+256+1024+4096+16384
BIT.ToBin|$$P01|$$P01|$$LEA
BIT.Show|b$$P01|$$ERG|$$LEA
MBX.The result can not directly displayed, but is:$crlf$$$ERG
ENR.
```



Syntax

BIT.ToBin|[b/v]P1|P2[|P3]

Parameter Explanation

 *© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable for result or"-". If **P2** is not a variable, the result is placed on the TOS.

P3 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=2
CAL.$$P01=1+4+16+64+256+1024+4096+16384
BIT.ToBin|$$P01|$$P01|$$LEA
BIT.Show|b$$P01|$$ERG|$$LEA
MBX.The result can not directly displayed, but is:$crlf$$$ERG
ENR.
```

Remarks

Whatever you want to do with BITs, generally there are two ways to specify Parameters for the Bit-Operations.
You can specify a number "16" as "Text" then it has two Bytes length. Byte 1 is "1" and the second Byte is "6" - we call this "by value".
You can also use the ASC-Code "16" and then only use 1 Byte to store the number. We call this "as binary".

1. Specify Parameter as VALUE

```
' To specify the Parameter "as VALUE" we can just write it as a number, using the
' 9 is 1001 in Binary
VAN.$$P01=9
' We want extract Bit "1" which is "1" in Result
VAN.$$P02=1
BIT.Get|v$$P01|$$P02|$$RES
' We change the Result to a readable Value using the ASC.-Command. This will only
' The result will be "1" as CHR-value (Binary). To make it readable we use the AS
ASC.$$RES|$$ERG
MBX.Result is $$ERG
```

This will give the result: "1", because the first bit in "1001" (dez. 9) is a "1".

© 2023 Theo Gottwald * https://www.smart-package.com

In both cases, we did not specify P4 (Size-Parameter) so the default size of 8-bit was used.

2. Specify Parameter as BINARY

```
' To specify the Parameter "as Binary" we use the CHR. Command if the Parameter i
' 9 is 1001 in Binary
CHR.9|$$P01
' We want extract Bit "1" which is "1" in Result
VAN.$$P02=1
BIT.Get|b$$P01|$$P02|$$RES
' We change the Result to a readable Value using the ASC.-Command. This will only
' The result will be "1" as CHR-value (Binary). To make it readable we use the AS
ASC.$$RES|$$ERG
MBX.Result is $$ERG
```

3. Specify multi-Byte Parameters "as Binary" or "as Value"

```
' Use the "ToBin" or the "ToVal" Option to convert the Parameters
VAN.$$P01=9
' $$P02 will contain the result
BIT.ToBin|$$P01|$$P02|2
' $$P02 will contain the result
BIT.ToVal|$$P01|$$P02|2
```

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.

Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.19 BIT.Toggle - Switch Bit

MiniRobotLanguage (MRL)

BIT.Toggle - Switch Bit

Change state of a bit from "0" to "1" or from "1" to "0"

Intention

`BIT.Toggle` will check if a bit is currently "1" or "0" first, and then switch it over.
Its somehow like a light-switch. If you click it, it will make the light "ON" or "OFF" depending on how it w

The parameters **P1** and **P2** can be given as value or as "binary".
Please see ! Smarty's BIT. - Operations 441 for details on this topic.

```
VAN.$$LEA=1
VAN.$$P01=67
VAN.$$P02=2
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Toggle|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 Toggle $$P02 is $$P03$crlf$Bin: $$ERA Toogle $$P02 -> $$ERC
BIT.Toggle|$$P03|$$P02|$$P04|$$LEA
BIT.Show|v$$P04|$$ERD|$$LEA
MBX.$$P03 Toggle $$P02 is $$P04$crlf$Bin: $$ERC Toogle $$P02 -> $$ERD
ENR.
```

Syntax

BIT.Toggle|[b/v]P1|[b/v]P2|P3[|P4]

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number for the bit to Toggle. Depending on **P4** this number can be 1 to 8, 1 to 16 or 1 to 32. Unless "b" is prefixed the default type for **P2** is "value".

P3 - optional,Variable for result or "-". The result will always be numeric/value and can only be 1 or 0.
If omitted the result is placed on the TOS. Also here you can specify [b/v] for the format of the result. If you do not specify anything the result is "value"-Format.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=67
VAN.$$P02=2
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Toggle|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 Toggle $$P02 is $$P03$crlf$Bin: $$ERA Toogle $$P02 -
> $$ERC
BIT.Toggle|$$P03|$$P02|$$P04|$$LEA
BIT.Show|v$$P04|$$ERD|$$LEA
MBX.$$P03 Toggle $$P02 is $$P04$crlf$Bin: $$ERC Toogle $$P02 -
> $$ERD
ENR.
```

Remarks

-

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.

© 2023 Theo Gottwald * https://www.smart-package.com

Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.20 BIT.ToVal - change Parameter to Value-Format

BIT.ToVal - change Parameter to Value-Format Previous 2350 Top 41 Next 2354

MiniRobotLanguage (MRL)

BIT.ToVal - Convert Parameter from "Binary" to "Value"

Convert Parameter to "Value"

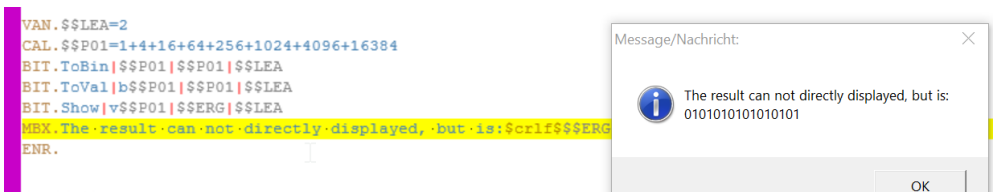
Intention

Generally there are two types od Parameters, these are "as value" and "as binary".

Whatever you want to do with BITs, generally there are two ways to specify Parameters for the Bit-Operations.
You can specify a number "16" as "Text" then it has two Bytes length. Byte 1 is "1" and the second Byte is "6" - we call this "by value".
You can also use the ASC-Code "16" and then only use 1 Byte to store the number. We call this "as binary".
More explanation is below under Remarks and see ! Smarty's BIT. - Operations 441

The parameters **P1** is the "Input" and it is "as Value".
The result is in **P2** which is the "as binary" of **P1**.

```
VAN.$$LEA=2
CAL.$$P01=1+4+16+64+256+1024+4096+16384
BIT.ToBin|$$P01|$$P01|$$LEA
BIT.ToVal|b$$P01|$$P01|$$LEA
BIT.Show|v$$P01|$$ERG|$$LEA
MBX.The result can not directly displayed, but is:$crlf$$$ERG
ENR.
```



Syntax

BIT.ToVal|[b/v]P1|P2[|P3]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable for result or"-". If **P2** is not a variable, the result is placed on the TOS.

P3 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=2
CAL.$$P01=1+4+16+64+256+1024+4096+16384
BIT.ToBin|$$P01|$$P01|$$LEA
BIT.ToVal|b$$P01|$$P01|$$LEA
BIT.Show|v$$P01|$$ERG|$$LEA
MBX.The result can not directly displayed, but is:$crlf$$$ERG
ENR.
```

Remarks

Whatever you want to do with BITs, generally there are two ways to specify Parameters for the Bit-Operations.
You can specify a number "16" as "Text" then it has two Bytes length. Byte 1 is "1" and the second Byte is "6" - we call this "by value".
You can also use the ASC-Code "16" and then only use 1 Byte to store the number. We call this "as binary".

1. Specify Parameter as VALUE

```
' To specify the Parameter "as VALUE" we can just write it as a number, using the
' 9 is 1001 in Binary
VAN.$$P01=9
' We want extract Bit "1" which is "1" in Result
VAN.$$P02=1
BIT.Get|v$$P01|$$P02|$$RES
' We change the Result to a readable Value using the ASC.-Command. This will only
' The result will be "1" as CHR-value (Binary). To make it readable we use the AS
ASC.$$RES|$$ERG
MBX.Result is $$ERG
```


© 2023 Theo Gottwald * https://www.smart-package.com

This will give the result: "1", because the first bit in "1001" (dez. 9) is a "1".
In both cases, we did not specify P4 (Size-Parameter) so the default size of 8-bit was used.

2. Specify Parameter as BINARY

```
' To specify the Parameter "as Binary" we use the CHR. Command if the Parameter i
' 9 is 1001 in Binary
CHR.9|$$P01
' We want extract Bit "1" which is "1" in Result
VAN.$$P02=1
BIT.Get|b$$P01|$$P02|$$RES
' We change the Result to a readable Value using the ASC.-Command. This will only
' The result will be "1" as CHR-value (Binary). To make it readable we use the AS
ASC.$$RES|$$ERG
MBX.Result is $$ERG
```

3. Specify multi-Byte Parameters "as Binary" or "as Value"

```
' Use the "ToBin" or the "ToVal" Option to convert the Parameters
VAN.$$P01=9
' $$P02 will contain the result
BIT.ToBin|$$P01|$$P02|2
' $$P02 will contain the result
BIT.ToVal|$$P01|$$P02|2
```

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.

Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482

© 2023 Theo Gottwald * https://www.smart-package.com

- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.43.21 BIT.XOR - Use logical XOR with Bits

MiniRobotLanguage (MRL)

BIT.XOR

Use logical XOR with Bits

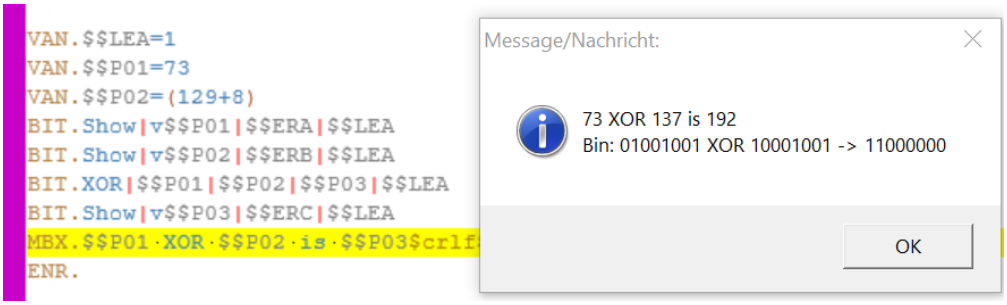
Intention

XOR returns TRUE (non-zero) if (and only if) the Bits in P1 AND P2 are uneqal. Its a reversable operation and therefore the number one operation in data-encryption and decryption.

The parameters **P1** and **P2** can be given as value or as "binary".
You can choose whether **P3** will be "as value" or "as binary". If you do not specify whether "b" or "v" the result will take the format of **P1**.
Using **P4** you can decide how many Bytes the result should have, 8, 16 or 32 bit.

Please see ! Smarty's BIT. - Operations 441

```
VAN.$$LEA=1
VAN.$$P01=73
VAN.$$P02=(129+8)
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.XOR|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 XOR $$P02 is $$P03$crlf$Bin: $$ERA XOR $$ERB -> $$ERC
ENR.
```



Syntax

BIT.XOR|[b/v]P1|[b/v]P2[|P3][|P4]

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for **P1**.

P2 **-** Variable containing a value or binary number. For example take the Number "9". It can be an ASC-Code "9" or the numeric number "9".
You can define this with a prefix of "v" for value and "b" for "binary". If the prefix is omitted, "v" is the default setting for P2.

P3 - optional,Variable for result. You can define with a prefix of "v" for value and "b" for "binary" the format of the result. If the prefix is omitted the result will have the format of **P1**.
If the P3 is "-" then result is placed on the TOS.

P4 - optional, can be 1,2 or 4. Number of Bytes that are used internally. The number of Bits is the number of Bytes times 8.
1 - 8 Bits
2 - 16 Bits
4 - 32 Bits

Do not use the "8" Option. See Limitations below.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAN.$$LEA=1
VAN.$$P01=73
VAN.$$P02=(129+8)
BIT.Show|v$$P01|$$ERA|$$LEA
BIT.Show|v$$P02|$$ERB|$$LEA
BIT.XOR|$$P01|$$P02|$$P03|$$LEA
BIT.Show|v$$P03|$$ERC|$$LEA
MBX.$$P01 XOR $$P02 is $$P03$crlf$Bin: $$ERA XOR $$ERB ->
$$ERC
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

When using the "|8" Parameter, the lower 4 bits are possibly wrong. This is not a bug its due to the way variables are stored internally.

Therefore we recommend to not use this option.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553
- ! Smarty's BIT. - Operations 441
- BIT.16to8 - Split 1 Word into 2 Bytes 449
- BIT.16to32 - concatenate 2 Words to 32 Bit DWord 446
- BIT.32to16 - Split 1 DWord into 2 Words 452
- BIT.8to16 - concatenate 2 Bytes to 16 Bit Word 455
- BIT.AND - Use logical AND with Bits 458
- BIT.Clear - Set Bit to 0 461
- BIT.LShift - Shift Bits to Left, insert 0 467
- BIT.NOT - Use logical NOT with Bits 470
- BIT.OR - Use logical OR with Bits 473
- BIT.RotateLeft - Rotate Bits Left 476
- BIT.RotateRight - Rotate Bits Right 479
- BIT.RShift - Shift Bits to Right, insert 0 482
- BIT.Set - Set Bit to 1 485
- BIT.Show - Show Bits in Binary-Numbers 488
- BIT.Store - Set Bit to 1 or 0 491
- BIT.ToBin - change Parameter to Binary-Format 494
- BIT.Toggle - Switch Bit 498
- BIT.ToVal - change Parameter to Value-Format 501
- BIT.XOR - Use logical XOR with Bits 505
- BIT.Get - check if Bit is set 464
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.44 Browser and Office (Acc.-Engine)

3.44.1 OFC. - Office-Commands

3.44.1.1 MS-Excel Scripting

3.44.1.1.1 !Excel-Scripting

MiniRobotLanguage (MRL)

OFC.Excel|Commands

Control MS-Excel worksheet via the SPR

Intention

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
There is one exception that is the "EXIST" Command. It will always execute and return whether a "1" or a "0" (Excel is installed -> 1, Excel is not installe-> 0) on the TOS.

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.

Hint:

1. Due to the nature of the underlying Excel-Scripting System it may not work to repeat tasks of opening and closing Excel hundreds of times. If such issues appear a reboot may be needed.
We have not tested this until now.

2. You may need to replace in german systems the "." with a "," in numbers that have been read. Because excel may read the numbers with ".".
For example "3.141"<>"3,131".
To replace it use:
`RPL.$$RES|.|,`

Syntax:

OFC.Excel|Subcommand and Data

Parameter Explanation:

Subcommands:

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
EXIST - Test if Excel is properly installed
OPEN - Open a existing Excel-File
CLOSE - Close an Excel-File that was opened using OPEN
SAVE - Save changes
SAVEAS - Save changes under another filename
SHEET_ACTIVATE - Activate another Sheep by Name/Number
SHEET_SETNAME - Set a new Name for a Sheep
SHEET_GETNAME - Get the name of a Sheet
FULLSCREEN - Make Excel Fullscreen
VISIBLE - Switch between visible/Invisible Mode
PAGE_UPDOWN - Scroll Page Up or down
SCROLL_UPDOWN - Scroll Up or down
SCROLL_LEFTRIGHT - Scroll Left or right
NUMBER_TO_LETTER - Convert a number to a letter
GET_CELL_???_BY_COORD - Get contents of a Cell by X and Y
Coordinates
GET_CELL_???_BY_RANGE - Get contents of a Cell by Range
"A1:A2".
FORMULA - Set a Cell Content
GET FORMULA - Get Formula from a Cell
LASTROW - Get the last used Row
LASTCOLUMN - Get the last used Column
EXCEL_GETTOPWIN - Get the Excel TopWindow-Handle
CALCULATE - Make Excel Calculate the sheet
SET_SIZE - Set Size of the Excel-Window
SET_POS - Set Position of the Excel-Window

' These SET Commands tell Excel how to load and save a Sheet
' for example using a Password or not.
SET UpdateLinks ON","SET UL ON"
SET UpdateLinks OFF","SET UL OFF"
SET FORMAT","SET FMT"
SET READONLY ON","SET RO ON"
SET READONLY OFF","SET RO OFF"
SET PASSWORD","SET PWD"
SET WRITE_PASSWORD","SET WPWD"
SET IGNORE_ROM ON","SET IROM ON"
SET IGNORE_ROM OFF","SET IROM OFF"
SET ORIGIN WINDOWS","SET ORI WIN"
SET ORIGIN MAC","SET ORI MAC"
SET ORIGIN DOS","SET ORI DOS"
SET DELIMITER","SET DELM"
SET EDITABLE ON","SET ED ON"
SET EDITABLE OFF","SET ED OFF"
SET NOTIFY ON","SET NF ON"
SET NOPTIFY OFF","SET NF OFF"
SET ADDTOMRU ON","SET MRU ON"
SET ADDTOMRU OFF","SET MRU OFF"
SET LOCAL ON","SET LOC ON"
SET LOCAL OFF","SET LOC OFF"
SET CORRUPTLOAD NORMAL","SET CL NORMAL"
SET CORRUPTLOAD REPAIR","SET CL REPAIR"
SET CORRUPTLOAD EXTRACT","SET CL EXTRACT"

Usage Examples:
OFC.Excel|SET CORRUPTLOAD REPAIR
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
OFC.Excel|SET WRITE_PASSWORD|4Z7ds74
OFC.Excel|SET PASSWORD|$$PWD
```

Example:

```
' EXCEL-Example
VAR.$$PFA=?path\Test3.xls
OFC.excel|OPEN|$$PFA|0|1

' Activate by name
OFC.excel|SHEET_Activate|2

'OFC.excel|SHEET_GetName|$$TXT|2

' Write a Block of cells
OFC.excel|FORMULA|A1:B10|2

' Write more Cells
FOR.$$LOP|1|12|6
  CAL.$$NUM=$$LOP+4
  OFC.excel|WRITE|B$$LOP:C$$NUM|=+A$$LOP*2,+A3*2,+A$$LOP*2,
+A2*2|1
  OFC.excel|FORMULA|E$$LOP:F$$NUM|=+A$$LOP*2|8
  OFC.excel|Get Formula|E$$LOP|0|$$TRE
  DBP.$$TRE
NEX.

' Max used Row/Column
OFC.EXCEL|LR|$$RES
OFC.EXCEL|LC|$$REC
MBX.Max used: $$RES-$$REC

' Max used Row/Column in Column 2
OFC.EXCEL|LR|$$RES|2
OFC.EXCEL|LC|$$REC|2
MBX.Max used: $$RES-$$REC

MBX.Click to Close
OFC.excel|CLOSE|1
END.

'-------------------------------------------------------

'
'SPR Script-file: Excel Benchmark
'Purpose:
'Author: TEOT\Theo
'Creation date: 05-11-2018 at 20:14:50
'=============================================================
'#EXE:?path\
'#SPI:ForceWrite
VAR.$$STA=#dtime#
VAR.$$SIZ=40
'VAR.$$PFA=?path\Test.xls
OFC.excel|OPEN||0|1
' Activate by name
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
OFC.Excel|EXCEL_GETTOPWIN|$$HWN
OFC.Excel|SET_POS|0|0
OFC.Excel|SET_SIZE|1024|740
OFC.excel|SHEET_Activate|1
OFC.excel|SET_SPEED|0
OFC.excel|SET ITERATION|1
VAR.$$STB=#dsince#
' Part 1 - Insert some numbers in A
FOR.$$LOP|1|$$SIZ
  RND.1|100000|$$RND
  CAL.$$RND=$$RND/100
  RPL.$$RND|.|,
  OFC.excel|WRITE|A$$LOP:A$$LOP|$$RND
NEX.
' Part 2
FOR.$$OUL|1|$$SIZ
  OFC.Excel|NUMBER_TO_LETTER|$$OUL|$$XLE
  CAL.$$XLA=$$OUL+1
  OFC.Excel|NUMBER_TO_LETTER|$$XLA|$$XLD
  FOR.$$LOP|1|$$SIZ
    VAR.$$FOR==1/SUMME(A1:$$XLE$$SIZ)
    IVV.$$LOP>1
      CAL.$$LOA=$$LOP+$$OUL-1
      VAR.$$FOR=$$FOR*$$XLA$$LOA+SUMME($$XLD1:$$XLD$$LOP)
    EIF.
    OFC.excel|FORMULA SET|$$XLD$$LOP|$$FOR|0
  NEX.
NEX.
VAR.$$STC=#dsince#
' Part III

FOR.$$LOP|1|($$SIZ/2)
  FOR.$$LOC|1|$$SIZ
    OFC.Excel|SCROLL_UPDOWN|$$LOP
    PAU.0.01
    OFC.Excel|SCROLL_LEFTRIGHT|$$LOC
    PAU.0.01
    OFC.Excel|SCROLL_UPDOWN|-$$LOP
    PAU.0.01
    OFC.Excel|SCROLL_LEFTRIGHT|-$$LOC
    PAU.0.01
  NEX.
NEX.
VAR.$$STD=#dsince#
VAV.$$ERG=Settings:$$STB$crlf$Recalculation:$$STC$crlf$Scrolli
ng: $$STD
MBX.$$ERG
OFC.Excel|CLOSE|1
ENR.
```

Remarks:

Microsoft, Excel and other used Trademarks are copyrighted by their respective owners. None of these Programs is part of the SPR-Distribution.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

This command can only be used if MS Excel is properly installed on the System. We have tested with MS Excel 2016.

See also:

- See Sample Script 513
-
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.2 !Sample-Scripts: Neuro-Net Cell

```
'
'SPR Script-file: NN_Excel
'Purpose: Neuronet-Cell in Excel
'Author: DESKTOP
'Creation date: 07-15-2019 at 17:26:54
'============================================================
'#EXE:?path\
'#SPI:ForceWrite

' EXCEL-Example
OFC.excel|OPEN||0|1

' Activate by name
OFC.excel|SHEET_ACTIVATE|Tabelle

GSB.Beschriftung
GSB.Input_orig
GSB.Set_Weight_Start

' Forward Pass: Summe der Inputs
OFC.Excel|FORMULA|F4|=+E4+E5+E6+E7|0

' Output= SumInput / NumInputs
OFC.Excel|FORMULA|H4|=+F4/4|0

' Setze Sollwert
OFC.Excel|FORMULA|H7|32640|1

' Berechne 1/Lernfaktor Epsilon
OFC.Excel|FORMULA|J10:J10|0,01|1
OFC.Excel|FORMULA|F7|=+1/J10|0

' Berechner Fehler
OFC.Excel|FORMULA|H10:H10|=+H4-H7|0
OFC.Excel|FORMULA|J4|=+(H10/F7)*-1|0
OFC.Excel|FORMULA|J7|=+(J4/4)|0

' Backward-Pass
OFC.Excel|FORMULA|F10|=+J7/F7|0

OFC.Excel|FORMULA|L4|=+E4*F10/F7|0
OFC.Excel|FORMULA|L5|=+E5*F10/F7|0
OFC.Excel|FORMULA|L6|=+E6*F10/F7|0
OFC.Excel|FORMULA|L7|=+E7*F10/F7|0

OFC.Excel|FORMULA|M4|=+D4+L4|0
OFC.Excel|FORMULA|M5|=+D5+L5|0
OFC.Excel|FORMULA|M6|=+D6+L6|0
OFC.Excel|FORMULA|M7|=+D7+L7|0

MBX. Start Backward-Pass

' With each Backward-Pass the Error of the Cell will reduce
DOL.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GSB.Backward
  MBX. Go On
OOP.

PAU.1
ENR.
'============================================================
' Sub-Programms
' Copy Result to Weights
:Backward
FOR.$$LOP|4|7
  GSB.CopyLine|$$LOP
NEX.
RET.


:CopyLine
VAN.$$COR=§§_01
OFC.Excel|GET_CELL_ANY_BY_RANGE|M$$COR|$$RES
' We replace in german systems the "." with a ","
RPL.$$RES|.|,
OFC.Excel|FORMULA|D$$COR|$$RES|1
RET.

:Beschriftung
OFC.Excel|WRITE|B3|Input
OFC.Excel|WRITE|D3|Gewichte
OFC.Excel|WRITE|E3|SumInput
OFC.Excel|WRITE|F6|1/Epsilon
OFC.Excel|WRITE|F3|AddInputs
OFC.Excel|WRITE|H3|Output
OFC.Excel|WRITE|H6|Soll
OFC.Excel|WRITE|H9|Error
OFC.Excel|WRITE|F9|Korr.-Faktor
OFC.Excel|WRITE|J3|Error gew.
OFC.Excel|WRITE|J6|Error/NumInp
OFC.Excel|WRITE|L3|Backward Pass
OFC.Excel|WRITE|J9|Lernfaktor Epsilon
RET.

' Setze Inputs for Start
:Input_orig
OFC.Excel|FORMULA|A4:B7|255|1
OFC.Excel|FORMULA|A5|12|1
OFC.Excel|FORMULA|A7|32|1

'VAN.$$NEG=127
VAN.$$NEG=0
OFC.Excel|FORMULA|B4|=+A4-$$NEG|0
OFC.Excel|FORMULA|B5|=+A5-$$NEG|0
OFC.Excel|FORMULA|B6|=+A6-$$NEG|0
OFC.Excel|FORMULA|B7|=+A7-$$NEG|0

OFC.Excel|FORMULA|E4|=+B4*D4|0
OFC.Excel|FORMULA|E5|=+B5*D5|0
OFC.Excel|FORMULA|E6|=+B6*D6|0
OFC.Excel|FORMULA|E7|=+B7*D7|0
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.

' Setze Weights for Start
:Set_Weight_Start
OFC.Excel|FORMULA|D4|16|1
OFC.Excel|FORMULA|D5|32|1
OFC.Excel|FORMULA|D6|128|1
OFC.Excel|FORMULA|D7|255|1
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.3 CLOSE

MiniRobotLanguage (MRL)

OFC.Excel|CLOSE

CLOSE MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
To Close your Excel-Instance, use the CLOSE Subcommand.
There is one Parameter, that will force Excel to close whether there are Error-Messages like "File not saved" etc. or not.
Set the Force Flag to "1" if you want to close Excel anyway.

The CLOSE Subcommand will only Close Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|CLOSE[|P1]

Parameter Explanation:

P1 - 0/1 - FORCE Flag suppress Error-Messages and force Excel to close

Example:

```
' Excel CLOSE Example
VAR.$$PFA=?path\Test.xls
OFC.excel|OPEN||0|1
DMP.6
OFC.excel|CLOSE|1
DMP.6
MBX.!
END.
```

Remarks:

To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The CLOSE Subcommand will only Close Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.4 EXIST

MiniRobotLanguage (MRL)

OFC.Excel|EXIST

Verify if MS-Excel is installed and functional

Intention

Normally all SPR commands for use with "MS Excel" need an valid Installation of MS-Office Excel.
There is one exception that is the "EXIST" Command. This command will always execute and return whether a "1" or a "0" on the TOS.

Excel is installed -> 1
Excel is not installed -> 0

This command will internally check the Registry if the Entry for Excel is set properly.
Secondly this command will check if the SPR can access the Excel internal Interfaces.
Only if all of this works, then the command will return "1" on TOS.
Else the command wil lreturn a "0" on TOS.

Syntax:

OFC.Excel|EXIST

Parameter Explanation:

-

Example:

```
' Check if MS-Excel is installed and functional
OFC.excel|Exist
DBP.$$000
ENR.
```

Remarks:

This command is automatically executed one time when the SPR starts and the first `OFC.EXCEL` command is executed.
For this reason, the first call to a Excel command may take a second longer then succeeding commands.
The result is internally remembered for Speed reasons.
So all further calls to `OFC.EXCEL` may fail if Excel was not working on first attempt.

© 2023 Theo Gottwald * https://www.smart-package.com

Unless you call this command, which will do the check again and then set the internal values new.

Limitations:

-

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Close 516

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.5 FORMULA GET

MiniRobotLanguage (MRL)

OFC.Excel|FORMULA GET

Read a Formula or any value from a Cell in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.Also you need to activate the sheet first using:

```
' Activate by Number
OFC.excel|SHEET_Activate|1
```

The Formula Command will read any sort of Text or Formula or numbers from the given Range of Cells.
For this you can use **P3** to specify the way how the Cell-Content is interpreted.

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

```
OFC.Excel|FORMULA|P1|P2[|P3]
OFC.Excel|FORMULA GET|P1|P2[|P3]
```

Parameter Explanation:

```
OFC.excel|Formula|$$RAN|$$RESUL-VAR[|numeric Paste Method]
```

P1 - RANGE, A1:A1 oder A1:B400 specifies the coordinates of the cells to be written to.
P2 - (opt.) Variable for result. If missing, invalid or it's "-" then TOS is used.
P3 - (opt./numeric) 0-8 Read Method, technically the way how Excel should interpret the Cell value/text

```
     0 - use "Cells.Formula" (default)
     1 - use "Cells.Value"
     2 - use "Cells.Value2"
     3 - use "Cells.FormualArray"
     4 - use "Cells.FormulaLabel"
     5 - use "Cells.FormulaHidden" (Get Cell Property)
     6 - use "Cells.FormulaLocal"
     7 - use "Cells.FormulaR1C1"
     8 - use "Cells.FormulaR1C1Local"
     9 - use "Cells.Numberformat" (Get Format of the specified Range)
```

Example:

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Excel FORMULA Example
OFC.excel|OPEN||0|1
DBP.---------------
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|Formula Set|A1:B3|25|1
OFC.excel|Formula Set|A2:C2|=+A1*2
OFC.excel|Formula Set|A3:C3|=+A1*2|6
OFC.excel|Formula Set|A4:C4|=+A1*2|8

OFC.excel|Formula Get|A2:A2|0|$$RES
OFC.excel|Formula Get|A2:A2|1|$$VAL
DBP.Formula at A2: $$RES
DBP.Value at A2: $$VAL

MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:

Some of the **P3** Parameters may not make sense (Hidden?) there may be changes in the future.

We replace in german systems the "." with a "," in numbers that have been read.

```
RPL.$$RES|.|,
```

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The FORMULA GET Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518
- Excel_Formula_Set 522

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.6 FORMULA SET

MiniRobotLanguage (MRL)

OFC.Excel|FORMULA SET

Write a Formula to a Cell in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Also you need to activate the sheet first using:

```
' Activate by Number
OFC.excel|SHEET_Activate|1
```

The Formula Command will write any sort of Text or Formula or numbers into the given Range of Cells.
For this you can use **P3** to specify the way how the Text is written into the Sheet.

The Command with the Parameter "9" can be used to set the Format (Numeric, General, Text etc.) of a Cell or a Range of Cells.

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|FORMULA|P1|P2[|P3]
OFC.Excel|FORMULA SET|P1|P2[|P3]

Parameter Explanation:

OFC.excel|Formula|$$RAN|$$FOR[| Paste Method]

P1 - RANGE, A1:A1 oder A1:B400 specifies the coordinates of the cells to be written to.
P2 - Text, this text is written in the cells. Delimiter can be used (P3).
P3 - (opt./numeric) 0-8 Paste Method, technically the way how Excel should interpret the written Text

```
     0 - use "Cells.Formula" (default)
     1 - use "Cells.Value"
     2 - use "Cells.Value2"
     3 - use "Cells.FormualArray"
     4 - use "Cells.FormulaLabel"
     5 - use "Cells.FormulaHidden" (Set Cell Property)
     6 - use "Cells.FormulaLocal"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
     7 - use "Cells.FormulaR1C1"
     8 - use "Cells.FormulaR1C1Local"
     9 - use "Cells.Numberformat" -> This can be use to change
the Cell-Format to Text or numeric etc.
```

Example:

```
' Excel FORMULA Example
OFC.excel|OPEN||0|1
OFC.excel|SHEET_Activate|2
FOR.$$LOP|0|8
  OFC.excel|FORMULA|B$$LOP:B$$LOP|=+A1*2|$$LOP
NEX.

DMP.6
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:

Some of the **P3** Parameters may not make sense (Hidden?) there may be changes in the future..

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The FORMULA Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518
- Excel_Formula_get 520

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.7 GET_CELL...BY COORD

MiniRobotLanguage (MRL)

OFC.Excel|GET_CELL_???_BY_COORD

Read a Cell in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Also you need to activate the sheet first using:

```
' Activate by Number
OFC.excel|SHEET_Activate|1
```

There are 3 versions og the Subcommand:

1. To read Simple numbers:
OFC.Excel|GET_CELL_NUM_BY_COORD|P1|P2[|P3]

2. To read Text
OFC.Excel|GET_CELL_TEXT_BY_COORD|P1|P2[|P3]

3. This will read the Cell as both Numbers and Text. And return the (longer) result that looks more valid.
OFC.Excel|GET_CELL_ANY_BY_COORD|P1|P2[|P3]

The GET_CELL ... BY_COORD Commands will read Text and Numbers from the Sheet. It may not read Formulas there.
Use the GET FORMULA-Command for that.
To specify which Cell to read, this Subcommand uses X- and Y-Coordinates, starting in the upper right Corner of the Sheet with (1,1).
If you prefer to use a RANGE like: "`A1:A1`" then use the `GET_CELL_???_BY_RANGE` Group of Commands.

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|GET_CELL_TEXT_BY_COORD|P1|P2[|P3]
OFC.Excel|GET_CELL_NUM_BY_COORD|P1|P2[|P3]

© 2023 Theo Gottwald * https://www.smart-package.com

OFC.Excel|GET_CELL_ANY_BY_COORD|P1|P2[|P3]

Parameter Explanation:

P1 - X-Coordinate, starting with "1" in the Upper,left corner of the sheet, specifies the X-coordinates of the cells to be written to.
P2 - Y-Coordinate, starting with "1" in the Upper,left corner of the sheet, specifies the Y-coordinates of the cells to be written to.
P3 - (optional, Variable or "-") contains the variable that will receive the Cell-Content.
If a Cell has String Content, "GET_NUM" will return "0". If P3 is missing then the result will be placed on TOS.
If a Cell has numeric Content then "GET_TEXT" will not return a result. Use "GET_ANY" to have an internal check which one to use.

Example:

```
' Excel GET_CELL Example
OFC.excel|OPEN||0|1
DBP.---------------
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|write|A1:A1|54
OFC.excel|write|A2:A2|Hallo

DBP. Text
OFC.excel|GET_CELL_TEXT_BY_RANGE|A1:A1|$$REA
OFC.excel|GET_CELL_TEXT_BY_RANGE|A2:A2|$$REB
DBP.$$REA-$$REB

DBP. Num
OFC.excel|GET_CELL_NUM_BY_RANGE|A1:A1|$$REA
OFC.excel|GET_CELL_NUM_BY_RANGE|A2:A2|$$REB
DBP.$$REA-$$REB

DBP. ANY
OFC.excel|GET_CELL_ANY_BY_RANGE|A1:A1|$$REA
OFC.excel|GET_CELL_ANY_BY_RANGE|A2:A2|$$REB
DBP.$$REA-$$REB

DBP. Text
OFC.excel|GET_CELL_TEXT_BY_COORD|1|1|$$REA
OFC.excel|GET_CELL_TEXT_BY_COORD|1|2|$$REB
DBP.$$REA-$$REB

DBP.Num
OFC.excel|GET_CELL_NUM_BY_COORD|1|1|$$REA
OFC.excel|GET_CELL_NUM_BY_COORD|1|2|$$REB
DBP.$$REA-$$REB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP. ANY
OFC.excel|GET_CELL_ANY_BY_COORD|1|1|$$REA
OFC.excel|GET_CELL_ANY_BY_COORD|1|2|$$REB
DBP.$$REA-$$REB
MBX.$$REA-$$REB

DMP.6
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:
To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

Limitations:
The GET_CELL ... BY_COORD Commands will read Text and Numbers from the Sheet. It may not read Formulas there.
Use the GET FORMULA-Command for that.

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The GET_CELL ... BY_COORD Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.8 GET_CELL...BY RANGE

MiniRobotLanguage (MRL)

OFC.Excel|GET_CELL_???_BY_RANGE

Read a Cell in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Also you need to activate the sheet first using:

```
' Activate by Number
OFC.excel|SHEET_Activate|1
```

There are 3 versions og the Subcommand:

1. To read Simple numbers:
OFC.Excel|GET_CELL_NUM_BY_RANGE|P1[|P2]

2. To read Text
OFC.Excel|GET_CELL_TEXT_BY_RANGE|P1[|P2]

3. This will read the Cell as both Numbers and Text. And return the (longer) result that looks more valid.
OFC.Excel|GET_CELL_ANY_BY_RANGE|P1[|P2]

The GET_CELL ... BY_RANGE Commands will read Text and Numbers from the Sheet. It may not read Formulas there.
Use the GET FORMULA-Command for that.

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|GET_CELL_TEXT_BY_RANGE|P1[|P2]
OFC.Excel|GET_CELL_NUM_BY_RANGE|P1[|P2]
OFC.Excel|GET_CELL_ANY_BY_RANGE|P1[|P2]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation:

P1 - RANGE, A1:A1 oder A1:B400 specifies the coordinates of the cells to be written to.
P2 - (optional, Variable or "-") contains the variable that will receive the Cell-Content.
If a Cell has String Content, "GET_NUM" will return "0". If P3 is missing then the result will be placed on TOS.
If a Cell has numeric Content then "GET_TEXT" will not return a result. Use "GET_ANY" to have an internal check which one to use.

Example:

```
' Excel GET_CELL Example
OFC.excel|OPEN||0|1
DBP.---------------
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|write|A1:A1|54
OFC.excel|write|A2:A2|Hallo

DBP. Text
OFC.excel|GET_CELL_TEXT_BY_RANGE|A1:A1|$$REA
OFC.excel|GET_CELL_TEXT_BY_RANGE|A2:A2|$$REB
DBP.$$REA-$$REB

DBP. Num
OFC.excel|GET_CELL_NUM_BY_RANGE|A1:A1|$$REA
OFC.excel|GET_CELL_NUM_BY_RANGE|A2:A2|$$REB
DBP.$$REA-$$REB

DBP. ANY
OFC.excel|GET_CELL_ANY_BY_RANGE|A1:A1|$$REA
OFC.excel|GET_CELL_ANY_BY_RANGE|A2:A2|$$REB
DBP.$$REA-$$REB

DBP. Text
OFC.excel|GET_CELL_TEXT_BY_COORD|1|1|$$REA
OFC.excel|GET_CELL_TEXT_BY_COORD|1|2|$$REB
DBP.$$REA-$$REB

DBP.Num
OFC.excel|GET_CELL_NUM_BY_COORD|1|1|$$REA
OFC.excel|GET_CELL_NUM_BY_COORD|1|2|$$REB
DBP.$$REA-$$REB

DBP. ANY
OFC.excel|GET_CELL_ANY_BY_COORD|1|1|$$REA
OFC.excel|GET_CELL_ANY_BY_COORD|1|2|$$REB
DBP.$$REA-$$REB
MBX.$$REA-$$REB

DMP.6
MBX.Click to Close
OFC.excel|CLOSE|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END.
```

Remarks:
To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

Limitations:
The GET_CELL ... BY_RANGE Commands will read Text and Numbers from the Sheet. It may not read Formulas there.
Use the GET FORMULA-Command for that.

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The GET_CELL ... BY_RANGE Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.9 LASTROW / LASTCOLUMN

MiniRobotLanguage (MRL)

OFC.Excel|LastRow/LastColumn

Find out how much Rows/Columns are used in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Also you need to activate the sheet first using:

```
' Activate by Number
OFC.excel|SHEET_Activate|1
```

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

```
OFC.Excel|LastRow[|P1][|P2]
OFC.Excel|LastColumn[|P1][|P2]
```

Parameter Explanation:

P1 - (optional, Variable or "-") contains the variable that will receive the Result. The first Row/Column is "1" (1-based).
P2 - (optional, numeric) - Specify the Row(Column) number which is used to count the number of used Cells.
If sheets are not rectangular, means the headline is possibly larger then the rest of the sheet, you can specify where (in which row/column) to count.

Example:

```
' Excel LASTROW Example
OFC.excel|OPEN||0|1
DBP.---------------
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|write|A1:D1|54
OFC.excel|write|A2:C8|Hallo
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
OFC.Excel|LastRow|$$REA
OFC.Excel|LastColumn|$$REB

MBX.Last used Column $$REA $crlf$ Last used Row: $$REB

OFC.Excel|LastRow|$$REA|1
OFC.Excel|LastColumn|$$REB|1

MBX.Measured in Row 1 Last used Column $$REA $crlf$ Last used
Row: $$REB

DMP.6
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:

-

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The `LASTROW/LASTCOLUMN` Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.10 LETTER_TO_NUMBER

MiniRobotLanguage (MRL)

OFC.Excel|LETTER_TO_NUMBER

Convert a numeric value int othe Alphabetic value of the Excel X-Axis

Intention

This command does not need Excel to be opened. It is a pure mathematical operation.
It will convert a pure Excel-X-Axis "Letter-Number" to a numeric representation.
It is the logical opposite from the "NUMBER_TO_LETTER" command.

```
   ' Convert "DD" to "99"
OFC.Excel|LETTER_TO_NUMBER|DD|$$RES
DBP.$$RES
```

Syntax:

OFC.Excel|LETTER_TO_NUMBER|P1[|P2]

Parameter Explanation:

P1 - (numeric) value that specifies the excel representation of a Cell on the X-Axis starting with "A".
P2 - (optional/Variable) Variable that will receive the numeric result of the operation. If omitted, TOS is used.

Example:

```
' Excel NUMBER_TO_LETTER Example
OFC.excel|OPEN||0|1

OFC.Excel|LETTER_TO_NUMBER|ZZZ|$$RES
DBP.$$RES
OFC.Excel|NUMBER_TO_LETTER|$$RES|$$LET
DBP.$$LET
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks:
-

Limitations:
These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
Will work on a large range of numbers at least from "A" to "ZZZZZZZ" (equals 236852010). Which is more then any Excel will possibly provide in the near future.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518
- Excel_NUMBER_TO_LETTER 534

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.11 NUMBER_TO_LETTER

MiniRobotLanguage (MRL)

OFC.Excel|NUMBER_TO_LETTER

Convert a numeric value int othe Alphabetic value of the Excel X-Axis

Intention

This command does not need Excel to be opened. It is a pure mathematical operation.
It will convert a pure numberical count to the Excel-X-Axis "Letter-Number".
This way you can more easily iterate through all fields and still use the "RANGE" Parameter in other commands.

```
   ' Convert "99" to "DU"
OFC.excel|NUMBER_TO_LETTER|$$NUM|$$RES
   MBX.$$RES
```

Syntax:

OFC.Excel|NUMBER_TO_LETTER|P1[|P2]

Parameter Explanation:

P1 - (numeric) value that specifies the numeric representation of a Cell on teh X-Axis starting with "1".
P2 - (optional/Variable) Variable that will receive the result of the operation. If omitted, TOS is used.

Example:

```
' Excel NUMBER_TO_LETTER Example
OFC.excel|OPEN||0|1
DBP.---------------
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|write|A9:DU9|54
OFC.excel|write|A6:CU9|Hallo

OFC.Excel|LastRow|$$REA
OFC.Excel|LastColumn|$$REB

OFC.Excel|NUMBER_TO_LETTER|$$REB|$$REB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.Last used Column $$REA $crlf$ Last used Row: $$REB

DMP.6
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:
-

Limitations:
These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
Will work from "A" to "ZZZZZZ" (equals 321272406).

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518
- Excel_LETTER_TO_NUMBER 532

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.12 OPEN

MiniRobotLanguage (MRL)

OFC.Excel|OPEN

OPEN or CREATE NEW MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
There are two parameters which can be changed to have Excel open in FULL-SCREEN Mode, or to open an invisible Instance.
If the command succeeds, it will drop an "1" on the TOS.

To Close your Excel-Instance, use the CLOSE Subcommand.

To Open an Instance of Excel, which does not load any worksheet, use:

```
OFC.excel|OPEN||0|1
```

Syntax:

OFC.Excel|OPEN|P1[|P2][|P3]

Parameter Explanation:

P1 - (optional) Filename of the XLS-File to open, can be any file that Excel can open.
P2 - (optional) 0/1 Fullscreen or Normal Window
P3 - (optional) 0/1 Visibility - if this is "0" then Excel will open invisible.

Example:

```
' Excel OPEN Example
VAR.$$PFA=?path\Test.xls
OFC.excel|OPEN|$$PFA|0|1
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|WRITE|A3:A4|44
OFC.excel|CLOSE|1
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks:
To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

Limitations:
These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.

See also:

- !Excel-Scripting 508
- Excel_Close 516
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.13 PAGE_UPDOWN

MiniRobotLanguage (MRL)

OFC.Excel|PAGE_UPDOWN

SCROLL MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Use the PAGE_UPDOWN Subcommand to scroll your Sheet by one or more full page(s) up or down.

Parameter **P1** specifies how many pages to scroll up (negative parameter) or down (positive parameter).

```
' Scroll down 2 Pages
OFC.excel|PAGE_UPDOWN|2

' Scroll up 2 Pages
OFC.excel|PAGE_UPDOWN|-2
```

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|PAGE_UPDOWN|P1

Parameter Explanation:

P1 - (numeric) - Negative or positive number of pages to scroll up (negative) or down (positive)

Example:

```
' Excel PAGE-SCROLL UP/DOWN Example
OFC.excel|OPEN||0|1
' Activate by name
OFC.excel|SHEET_Activate|2
FOR.$$LOP|1|199
  CAL.$$LOA=$$LOP+1
  OFC.excel|WRITE|A$$LOP:A$$LOP|$$LOP
  OFC.excel|FORMULA SET|B$$LOP:XX$$LOP|=A$$LOP*78+$$LOP|0
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$LOP|1|90
  ' Scroll down
  OFC.excel|PAGE_UPDOWN|2
  PAU.1
  ' Scroll up
  OFC.excel|PAGE_UPDOWN|-2
  PAU.1
NEX.

MBX.!
'PAU.4
'OFC.excel|SAVE|1
OFC.excel|CLOSE|1
END.
```

Remarks:

You can not scroll much more down then there are lines in your Spreadsheed.

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The `PAGE_UPDOWN` Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518
- Excel_SCROLL_UPDOWN 547

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.14 SAVE

MiniRobotLanguage (MRL)

OFC.Excel|SAVE

SAVE MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
To SAVE your Excel-Worksheet, use the SAVE Subcommand.
There is one Parameter, that will force Excel to suppress any Error-Messages like "File exists" etc.
Set the Force Flag to "1" if you want Excel to Save your Files anyway (if possible with the given name and path).

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|SAVE[|P1]

Parameter Explanation:

P1 - 0/1 - FORCE Flag suppress Error-Messages and force Excel to SAVE even if the file already exists

Example:

```
' Excel SAVE Example
VAR.$$PFA=?path\Test5.xls

' Open New empty Sheet
OFC.excel|OPEN||0|1

' Activate by name
OFC.excel|SHEET_Activate|2

' Save as Filename
OFC.excel|SAVEAS|$$PFA
MBX.!

' Write some numbers into Sheet
OFC.excel|FORMULA|A1:B10|2

' SAVE
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
OFC.excel|SAVE

' Show Stack
DMP.6

MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:
To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

Limitations:
These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The SAVE Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.15 SAVEAS

MiniRobotLanguage (MRL)

OFC.Excel|SAVEAS

SAVE MS-Excel worksheet via the SPR using a Filename

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
To SAVE your Excel-Worksheet, use the SAVEAS Subcommand.
There is one Parameter, that will force Excel to suppress any Error-Messages like "File exists" etc.
Set the Force Flag to "1" if you want Excel to Save your Files anyway (if possible with the given name and path).

You can use the "SET" Commands to save with a Password and for other Options.

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|SAVEAS[|P1][|P2][|P3][|P4]

Parameter Explanation:

Filename$, Filetype&,Quite& (1-keine Alerts/0-Alerts),Create_Backup& (1/0)

P1 - (Text) Filenpath&Filename
P2 - (opt,numeric) - Fileformat - leave zero to use Standard "xlTextWindows" alternative numbers see below in "Remarks" Section. (default 0)
P3 - (opt,0/1) - FORCE Flag suppress Error-Messages and force Excel to SAVEAS even if the file already exists
P4 - (opt,0/1) - Generate Backup on Save (default 0)

Example:

```
' Excel SAVEAS Example
VAR.$$PFA=?path\Test5.xls

' Open New empty Sheet
OFC.excel|OPEN||0|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Activate by name
OFC.excel|SHEET_Activate|2

' Save as Filename
OFC.excel|SAVEAS|$$PFA
MBX.!

' Write some numbers into Sheet
OFC.excel|FORMULA|A1:B10|2

' SAVE
OFC.excel|SAVE

' Show Stack
DMP.6

MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:

You can use the "SET" Commands to save with a Password and for other Options.

```
    XlFileFormat - please use the numbers on the right side:
     xlAddIn                                                = 18          ' &H00000
     xlCSV                                                  = 6           ' &H00000
     xlCSVMac                                               = 22          ' &H00000
     xlCSVMSDOS                                             = 24          ' &H00000
     xlCSVWindows                                           = 23          ' &H00000
     xlDBF2                                                 = 7           ' &H00000
     xlDBF3                                                 = 8           ' &H00000
     xlDBF4                                                 = 11          ' &H00000
     xlDIF                                                  = 9           ' &H00000
     xlExcel2                                               = 16          ' &H00000
     xlExcel2FarEast                                        = 27          ' &H00000
     xlExcel3                                               = 29          ' &H00000
     xlExcel4                                               = 33          ' &H00000
     xlExcel5                                               = 39          ' &H00000
     xlExcel7                                               = 39          ' &H00000
     xlExcel9795                                            = 43          ' &H00000
     xlExcel4Workbook                                       = 35          ' &H00000
     xlIntlAddIn                                            = 26          ' &H00000
     xlIntlMacro                                            = 25          ' &H00000
     xlWorkbookNormal                                       = -4143       ' &HFFFFF
     xlSYLK                                                 = 2           ' &H00000
     xlTemplate                                             = 17          ' &H00000
     xlCurrentPlatformText                                  = -4158       ' &HFFFFF
     xlTextMac                                              = 19          ' &H00000
     xlTextMSDOS                                            = 21          ' &H00000
     xlTextPrinter                                          = 36          ' &H00000
     xlTextWindows                                          = 20          ' &H00000
     xlWJ2WD1                                               = 14          ' &H00000
     xlWK1                                                  = 5           ' &H00000
     xlWK1ALL                                               = 31          ' &H00000
     xlWK1FMT                                               = 30          ' &H00000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
xlWK3                                  = 15          ' &H00000
xlWK4                                  = 38          ' &H00000
xlWK3FM3                               = 32          ' &H00000
xlWKS                                  = 4           ' &H00000
xlWorks2FarEast                        = 28          ' &H00000
xlWQ1                                  = 34          ' &H00000
xlWJ3                                  = 40          ' &H00000
xlWJ3FJ3                               = 41          ' &H00000
xlUnicodeText                          = 42          ' &H00000
xlHtml                                 = 44          ' &H00000
xlWebArchive                           = 45          ' &H00000
xlXMLSpreadsheet                       = 46          ' &H00000
xlExcel12                              = 50          ' &H00000
xlOpenXMLWorkbook                      = 51          ' &H00000
xlOpenXMLWorkbookMacroEnabled          = 52          ' &H00000
xlOpenXMLTemplateMacroEnabled          = 53          ' &H00000
xlTemplate8                            = 17          ' &H00000
xlOpenXMLTemplate                      = 54          ' &H00000
xlAddIn8                               = 18          ' &H00000
xlOpenXMLAddIn                         = 55          ' &H00000
xlExcel8                               = 56          ' &H00000
xlOpenDocumentSpreadsheet              = 60          ' &H00000
xlOpenXMLStrictWorkbook                = 61          ' &H00000
xlCSVUTF8                              = 62          ' &H00000
xlWorkbookDefault                      = 51          ' &H00000

For technical details see:
https://msdn.microsoft.com/de-de/library/microsoft.office.interop.excel.xlfile
```

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The SAVE Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.16 SCROLL_LEFTRIGHT

MiniRobotLanguage (MRL)

OFC.Excel|SCROLL_LEFTRIGHT

SCROLL MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Also you need to activate the sheet first using:

```
' Activate by Number
OFC.excel|SHEET_Activate|1
```

Use the SCROLL_LEFTRIGHT Subcommand to scroll your Sheet by one or more cell(s) left or right.

Parameter **P1** specifies how many cells to scroll left (negative parameter) or right (positive parameter).

```
'Scroll 2 Cells to the right
OFC.excel|SCROLL_LEFTRIGHT|2

'Scroll 2 Cells to the left
OFC.excel|SCROLL_LEFTRIGHT|-2
```

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|SCROLL_LEFTRIGHT|P1

Parameter Explanation:

P1 - 0/1 - FORCE Flag suppress Error-Messages and force Excel to close

Example:

```
' Excel SCROLL UP/DOWN Example
OFC.excel|OPEN||0|1
' Activate by name
OFC.excel|SHEET_Activate|2
FOR.$$LOP|1|199
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  CAL.$$LOA=$$LOP+1
  OFC.excel|WRITE|A$$LOP:A$$LOP|$$LOP
  OFC.excel|FORMULA SET|B$$LOP:XX$$LOP|=A$$LOP*78+$$LOP|0
NEX.

FOR.$$LOP|1|90
  'Scroll 2 Cells to the right
  OFC.excel|SCROLL_LEFTRIGHT|2
  PAU.1
  'Scroll 2 Cells to the left
  OFC.excel|SCROLL_LEFTRIGHT|-2
  PAU.1
NEX.

MBX.!
'PAU.4
'OFC.excel|SAVE|1
OFC.excel|CLOSE|1
END.
```

Remarks:

You may not be able to scroll more to the right then there are used Cells in your sheet!

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The `SCROLL_LEFTRIGHT` Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_SCROLL_UPDOWN 547
- Excel_Page_UD 538

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.17 SCROLL_UPDOWN

MiniRobotLanguage (MRL)

OFC.Excel|SCROLL_UPDOWN

SCROLL MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
Use the SCROLL_UPDOWN Subcommand to scroll your Sheet by one or more line(s) up or down.

Parameter **P1** specifies how many lines to scroll up (negative parameter) or down (positive parameter).

```
' Scroll 2 Lines down
OFC.excel|SCROLL_UPDOWN|2

'Scroll 2 Lines up
 OFC.excel|SCROLL_UPDOWN|-2
```

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|SCROLL_UPDOWN|P1

Parameter Explanation:

P1 - 0/1 - FORCE Flag suppress Error-Messages and force Excel to close

Example:

```
' Excel SCROLL UP/DOWN Example
OFC.excel|OPEN||0|1
' Activate by name
OFC.excel|SHEET_Activate|2
FOR.$$LOP|1|199
  CAL.$$LOA=$$LOP+1
  OFC.excel|WRITE|A$$LOP:A$$LOP|$$LOP
  OFC.excel|FORMULA SET|B$$LOP:XX$$LOP|=A$$LOP*78+$$LOP|0
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$LOP|1|90
' Scroll 2 Lines down
  OFC.excel|SCROLL_UPDOWN|2
  PAU.1
  'Scroll 2 Lines up
  OFC.excel|SCROLL_UPDOWN|-2
  PAU.1
NEX.

MBX.!
'PAU.4
'OFC.excel|SAVE|1
OFC.excel|CLOSE|1
END.
```

Remarks:

To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The `PAGE_UPDOWN` Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_SCROLL_LEFTRIGHT 545
- Excel_Page_UD 538

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.18 SENDKEY

MiniRobotLanguage (MRL)

OFC.Excel|SENDKEY

Write to a Cell in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.

The Sendkey Subcommand will be useful to navigate inside Excel. To find the right keys to send, go into Excel manually and pres the ALT-key.

```
' Open the File-Menu
OFC.Excel|SENDKEY|"%d"
' Choose new Page
OFC.Excel|SENDKEY|"%n"

' Write "Hallo"
OFC.Excel|SENDKEY|Hallo
```

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|SENDKEY|P1

Parameter Explanation:

P1 - Characters to send. Besides normal Letters there are several special characters.

© 2023 Theo Gottwald * https://www.smart-package.com

Key	Code
BACKSPACE	{BACKSPACE} or {BS}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
CLEAR	{CLEAR}
DELETE or DEL	{DELETE} or {DEL}
DOWN ARROW	{DOWN}
END	{END}
ENTER (numeric keypad)	{ENTER}
ENTER	~ (tilde)
ESC	{ESCAPE} or {ESC}
HELP	{HELP}
HOME	{HOME}
INS	{INSERT}
LEFT ARROW	{LEFT}
NUM LOCK	{NUMLOCK}
PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
RETURN	{RETURN}
RIGHT ARROW	{RIGHT}
SCROLL LOCK	{SCROLLLOCK}
TAB	{TAB}
UP ARROW	{UP}
F1 through F15	{F1} through {F15}

© 2023 Theo Gottwald * https://www.smart-package.com

To combine a key with	Precede the key code with
SHIFT	+ (plus sign)
CTRL	^ (caret)
ALT	% (percent sign)

For more Details see MSDN.

Example:

```
' Excel WRITE Example
OFC.excel|OPEN||0|1
' Activate by name
OFC.excel|SHEET_Activate|2
' Write "Hallo"
OFC.Excel|SENDKEY|Hallo
' Open the File-Menu
OFC.Excel|SENDKEY|"%d"
' Choose new Page
OFC.Excel|SENDKEY|"%n"
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:

Not all Character-Kombinations make sense using SENDKEY. You can use the Standard SPR Keyboard Simulation Features also, but the Chance that Keys get lost are minimized if you use this Sendkey-Version, as it internally checks that no further Key is send until the Key before has been processed.

Limitations:

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The SENDKEY Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.1.1.19 WRITE

MiniRobotLanguage (MRL)

OFC.Excel|WRITE

Write to a Cell in an MS-Excel worksheet via the SPR

Intention

Generally you will need to OPEN a Excel-Instance first, this is done using the "OPEN" Subcommand.
To SAVE your Excel-Worksheet, use the SAVE Subcommand.
There is one Parameter, that will force Excel to suppress any Error-Messages like "File exists" etc.
Set the Force Flag to "1" if you want Excel to Save your Files anyway (if possible with the given name and path).

The WRITE Command wiill just put Text and Numbers into the Sheet. It will not place Formulas there.
Use the FORMULA-Command for that.

The Command will drop an "1" on the TOS if it was successful, otherwise a "0".

Syntax:

OFC.Excel|WRITE|P1|P2[|P3]

Parameter Explanation:

P1 - RANGE, A1:A1 oder A1:B400 specifies the coordinates of the cells to be written to.
P2 - Text, this text is written in the cells. Delimiter can be used (P3).
P3 - (opt./numeric) 0/1/2/3 - Which delimiters are evaluated.
0 - no delimiters are evaluated, just write one Cell
1 - use Row-Delimiter (default = ",")
2 - use Column-Delimiter (default = ";")

Example:

```
' Excel WRITE Example
OFC.excel|OPEN||0|1
' Activate by name
OFC.excel|SHEET_Activate|2
OFC.excel|WRITE|B1:F8|23,44;+A1*2,=+A2*2|1
OFC.excel|WRITE|B9:F9|55,-6;8,+A2*2|2
OFC.excel|WRITE|B12:e19|55,-6;8,+A2*2|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.6
MBX.Click to Close
OFC.excel|CLOSE|1
END.
```

Remarks:
To Close your Excel-Instance, use the CLOSE Subcommand.
However, its not necessary to close an Excel File, if you want to leave it open, even after the SPR exits.

Limitations:
The WRITE Command will just put Text and Numbers into the Sheet. It will not place Formulas there.
Use the FORMULA-Command for that.

These commands need an valid Installation of MS-Office Excel. It will check if MS Excel is properly installed, and if not just do nothing.
The WRITE Subcommand will only work with Excel-Instances that have previously been opened with the OPEN Subcommand of the same Robot-Instance.
If the command succeeds, it will drop an "1" on the TOS.

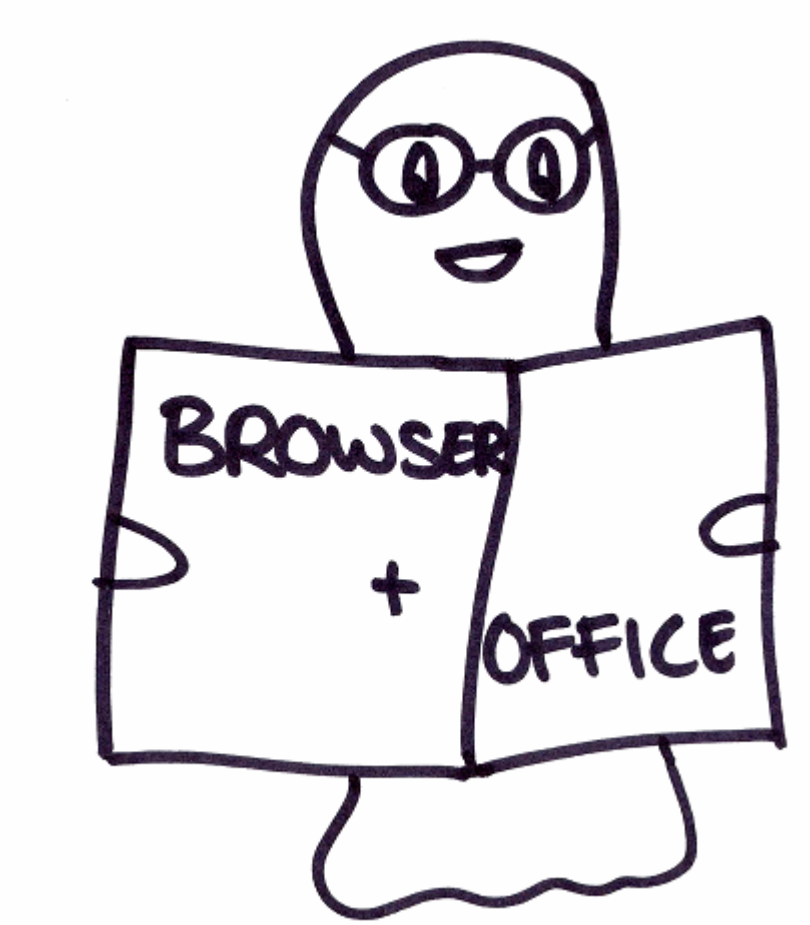
See also:

- !Excel-Scripting 508
- Excel_Open 536
- Excel_Exist 518

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.2 ! Smarty's Browser and Office Commands

Smarty's Browser and Office Commands



The Mini-Browser

The Mini-Browser is a Component of the SPR System. Internally it uses a Internet Explorer Control to display all the WEB-Sites in the same way as a Standard Internet Explorer would do. Technically the Internet Explorer Control is on any Windows Computer, no matter if the Internet Explorer itself is installed or not.

Using this Standard Internet Explorer Technology, the Mini-Browser (MB) is perfectly compatible with all the Commands `SAO.`, `AGR.` `etc.`
The `AGR.` Command will even Scroll the MB in the same way as if it would be an Standard Internet Explorer.

Why an own Browser?

*© 2023 Theo Gottwald * https://www.smart-package.com*

One important advantage is, that you are not so much dependent on the Browser that is installed on the target system. When you compile an executable, then **the Mini-Browser is been included in the package** and will be available on the target system.
This makes the target situation equal to the recording situation.

Secondly the MB opens its heartbeats to you. Using the supplied Systemvariables 334 and commands that are specifically for the MB, you can "hear" any event that occurs while loading a WEB- or Intranet Site. And of course you will know when its completely loaded and ready to take action.
And you can check if there have been any errors. In that later case, you will also know, exactly which error has happened.

What is it good for?

This gives you a new level of control over WEB-based automations.
No matter if you want to register a specific software package, or if you want to test the availability of your company WEB-Site.
The new MB will help you to get the Job done in a fraction of the time, that you would have needed without it.

How do I start the Mini-Browser (MB) ?

The Mini-Browser should be started using the `LBR.`-command to take full advantage of all its features.

Using the `LBR.`-Command you can immediately give all these needed Information like

- Size and Position
- Starting Page
- Survive when Robot/Script ends? Or close in this case?

to the MB directly on Startup.

As you can see in the picture below, the MB is optimized for cisual automation.
It does not only have the normal "Status Line", but also has another Events-Static, at the Top of the window. Using these, you can perfectly control its actions within a script.
Even without using any of the special features that are available.

Having said this, you can start the MB just like any other browser, using a Commandline:

```
EXE.SPR-Browser.exe www.fa2.de
MBX.Ready
ENR.
```

This way you can only give one hyperlink and no other parameters.

Can I "connect" my Script to a MB that was already running?

To use all the features, the MB must be connected with the Robot-Script.
This is automatically done, when you use the `LBR.`-Command to start the MB.

© 2023 Theo Gottwald * https://www.smart-package.com

If the MB is already running, you can "connect" your Script to that MB.
This is useful, when the MB is been started from other tools or programs.

```
' Connect to a running MB usign
' LBR.connect

EXE.SPR-Browser.exe www.fa2.de
' First locate the MB you want to connect
STW.t|SPR Mini
' Or get its CUN
GIK.cfh.||$$BCU
' THen use LBR. to connect to the desired MB
LBR.connect|$$BCU
' Here is check if it worked
DOL.
  DBP.$mb_ert$
  DBP.#mb_busy#-#mb_prog#
OOP.(0)
MBX.Ready
ENR.
```

How do i register a browser as the default browser?

You can register the MB temporarily or permanent as the default browser.
This may also be needed, when you have a program that automatically starts a browser.
If you register the MB as Default Browser, you can more easy use it for browsing the WEB.

These are the registry keys you need to change to set the default browser:

```
HKEY_CLASSES_ROOT\http\shell\open\command
HKEY_CLASSES_ROOT\https\shell\open\command
```

You can just write the path of the desired browser into this Reg-Key.

© 2023 Theo Gottwald * https://www.smart-package.com

A Screenshot of the Mini-Browser.

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.3 AFT. / AFF.. - Acc.- Find Text

AFT. / AFF. - Acc.- Find Text (Sear Previous 554 Top 41 Next 564

MiniRobotLanguage (MRL)

AFT. Command

Accessibility Find Text

AFF. Command

Accessibility Find Fast

Intention

These commands will search for a specified text in the given window. It was designed for most easy usage. You search a text?
`AFT./ AFF.` will search that text in that window.
`AFT./ AFF.` was made for easy usage. Just give the text to search for. Like in this example.
The only difference between `AFT.` and `AFF.` is, that `AFF.` uses multiple CPU-Cores. This may lead to a SpeedUp under circumstances when AFT. is too slow.
If AFT. is fast enough, use AFT. because in this case AFF. may be slower then. If AFT, is too slow, try AFF. instead.

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
AFT.|google&ANDTHEN:nwshp
MMV.
ENR.
```

You can specify a role-parameter, if you want. Like below.

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
AFT.40,43|google&ANDTHEN:nwshp
MMV.
ENR.
```

But maybe you don't want in most cases, if your text is unique. If you want to click immediately on the found object, `AFT.` understands all "Inline Role-Parameters" (Subcommands) like `AGR`.

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
' This will immediately click the link.
AFT.!MOS.|google&ANDTHEN:nwshp
ENR.
```

If the target application is an Internet explorer, `AFT.` will scroll to bring the located object into the visible area.

© 2023 Theo Gottwald * https://www.smart-package.com

Technically AFT. localizes a specified Acc.-Object in a Child-Window or Top-Window and sets the actual point into the object center. It's similar to AGR. but it does not separate the value and the name parameter. They are both treated as "Text".

Additionally you can specify a number, if there are multiple objects that fit your textual description, and you prefer to not take the first one. If you omit the number the default is 1. Means that you will get the first object that fits your description.

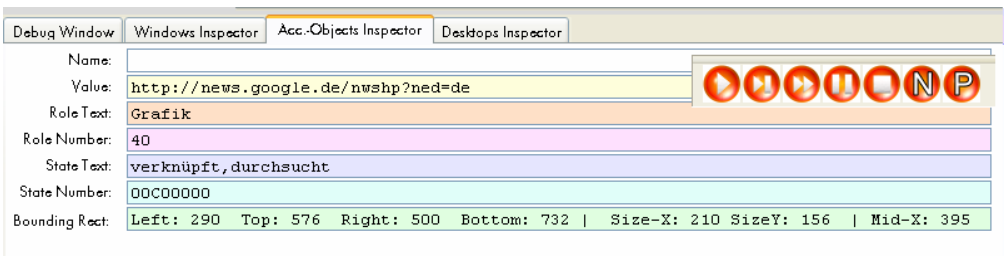
AFT. supports the use of so called "Role-Parameter". These are Inline-Subcommands which get executed only if the object could be found. Here is an example:

```
' The "Inline MOS." will select the Link in the IE
AFT.!MOS.|google&ANDTHEN:nwshp
```

For a complete list of all "Role-Parameters" see below.

You can see Acc-Objects in the Acc.-Inspector.Let us see an example from a WEB-Site in the Internet Explorer:

In the picture below you will see the resulting Display in the Acc.-Inspector.



The most important parameters for AFT. are:
"**text**" and "**role**". In most cases you can even ignore the role parameter and just use "text".

Now we'll take the above picture from the Acc.-Analyzer, and make the code for it.

Using AFT. we do best locate a Top- and a child-window. Like this:

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
AFT.40|google&ANDTHEN:nwshp
MMV.
ENR.
```

In some cases this may not be possible, or will not work. Then we can use `AFT.` just on the Top-Window:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
AFT.40|google&ANDTHEN:nwshp
MMV.
ENR.
```

Which may be a bit slower in execution times, depending on the overall complexity of the WEB-/Office program.

Hints:
It is important, that you localize a window before using `AFT.`

`Unlike SAO., AFT.` does not use the 1.4. The Prototype Parameter System 73.

`AFT.` supports 3.2 Standard-Search Pattern 341 and they can help you to better define the things you look for. See example above "`google&ANDTHEN:nwshp`".

Please note that `AFT.` can be quite slow when being used on complex applications and slow Computers.

Use the Acc.-Inspector in the Editor to get the necessary parameter data.

`AFF.`

We have implemented an (undocumented) experimental) version of `AFT.`, called `AFF.` Unlike `AFT.`, `AFF.` uses multilpe CPU-cores to speed things up. It will not scroll the `IE`, and the "Nr." parameter may not work as expected (which is due to the fact that we can not predict, which core finds what element first). It may however be faster in complex cases then `AFT.` `AFF.` is experimental, Use it with caution.

Syntax

AFT.[P1][|P2...|Px]
AFF.[P1][|P2...|Px]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - (optional)* Role, and Role-Parameter,** a numeric value as you it in the Acc.-Inspector, when you are over the element with the mouse. ***You can specify additional "Subcommands" after the Role-Parameter* (See below), separated by a "!". Example:

```
AFT.40!WII.MLE.|google&ANDTHEN:nwshp
```

P2 - (optional)* Text**, a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse. ***"Text" can be the "Value" as well as the "Name".

***P3 - (optional)* Number** of the element. There may be multiple elements that fit your description. In that case you can choose which one you want using this parameter. You need to try which number you need to locate the element of your choice.

***P4 - (optional)* Flags.** A numeric value that is a combination of the below numbers.

```
1 - only Objects,
2 - only Simple Elements
4 - display debug info while searching.
```

```
while all parameters are marked as "optional", you need to
specify some of them. An empty AGR. does not make any sense.
```

*Role-Parameter:

You can specify multiple Roles, they must be separated by ",".
Any of these will be valid.

If you specify a "#" before the Role-Parameter, it will mean "Role-Group" and specifies any of the following:
0: unknown
1: User-bedienbare Controls
2: Panels, Client und Background
4: Borders etc.
8: Window-Controls (Titlebar, Minimize etc.)
16: Scrollbars
32: Spezial, Cursor, Sound etc.
These inline-commands can be specified together with the role-parameter, separated by a "!". You can specify as man of them as you wish. They will only get executed, in case of an element is found:

```
"TAF" - "a" ' Take Focus
"TAS" - "b" ' Take Selection
"ESE" - "c" ' Ext. Select
"ASE" - "d" ' Add to Selection
"RSE" - "e" ' Remove Selection
"SSE" - "f" ' Strong Select
"SHC" - "g" ' Shift-Click
"CCS" - "h" ' ctrl-click - select
"CSE" - "i" ' ctrl-click cancel selection
"MER" - "j" ' MER at Pos.
"MSE" - "k" ' do Multiselect
"MOS" - "m" ' Mouseless Operation Select
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"DMO" - "n" ' Double Mouseless Operation Select
"MLI" - "o" ' MLI at place
"DMR" - "p" ' Double Mouseless Operation Select with rest
"MPO" - "q" ' Mousepointer Position Set
"MSO" - "r" ' Mouse Operation
"MDO" - "s" ' Mouse Double Operation
"WIN" - "t" ' Set Wait-Input Idle- Time to infinte
"MDC" - "u" ' MDC at place
"MRC" - "v" ' MRC at place
"WII" - "w" ' Wait-for-Input-Idle (max. 5 Seconds)
"MLE" - "x" ' MLE.
"SMP" - "y" ' Set Mousepos Mode
"BTP" - "z" ' Set Left-ControlPosition like for Checkmark
"BTC" - "0" ' check State before click and click
              if actually state=0 (check it)
"BTS" - "1" ' check State before click and click if actually
              state=1 (uncheck it)
```

Example

```
' This Example will find an item in the Internet Explorer
' on a WEB-Site
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
AGR.40||google&ANDTHEN:nwshp|1|64
MMV.
ENR.
```

Remarks

-

Limitations:

This command can be slow on complex applications.

See also:

- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591
- SAO. - Search Accessible Object 625
- AGR. - Accessibility Get Rect 564

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.4 AGR. / AGF. - Accessibility Get Rect

AGR. / AGF. - Acc.-Get Rect (Search Acc.-Object) Previous 558 Top 41 Next 570

MiniRobotLanguage (MRL)

AGR. Command

Search Accessible Object

AGF. Command

Search Accessible Object Fast

Intention

This command localizes a specified Acc.-Object in a Child-Window or Top-Window and sets the actual point into the object center.
It's similar to `SAO.` but it does not use the 1.4. The Prototype Parameter System 73 for its parameters. Therefore it has only the basic options, but may be easier to use.

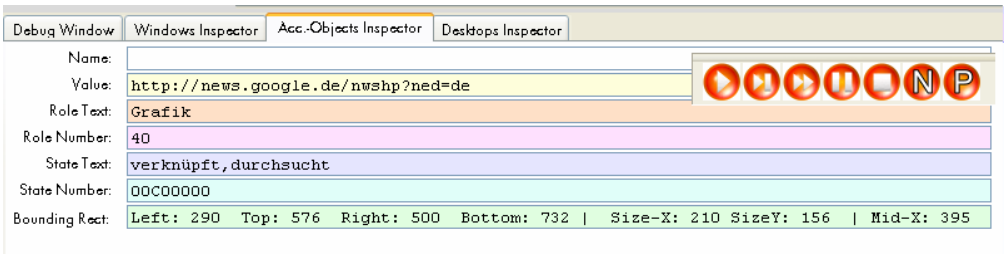
AGR. is using just one CPU-Core and therefore the order in which you find Objects is clearly defined.
AGF. uses multiple CPU-Cores, this makes it faster on large search operations, but does not allow you to specify the order in which objects are returned.
As we can not know which CPU finds which Object first.

```
012 0000000000332592 0001 STW.ct|IEFrame|Theo Gottwald
013 0000000009320706 0001 AGR.40||findip
014 0000000029714108 0001 MMV.
015 0000000047683591 0001 MBX.!
016 0000000007006184 0001 AGF.40||findip
017 0000000029856331 0001 MMV.
018 0000000000000000 0001 DMP.1
019 0000000000000000 0000 MBX.1
020 0000000000000000 0000 ENR.
```

You can see Acc-Objects in the Acc.-Inspector.Let us see an example from a WEB-Site in the Internet Explorer:

© 2023 Theo Gottwald * https://www.smart-package.com

In the picture below you will see the resulting Display in the Acc.-Inspector.



The most important parameters for AGR. are:
"**name**", "**role**" and "**value**". In most cases you can ignore the other parameters.

Now we'll take the above picture from the Acc.-Analyzer, and make the code for it.

Using AGR. we do best locate a Top- and a child-window. Like this:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
AGR.40||google&ANDTHEN:nwshp
MMV.
ENR.
```

In some cases this may not be possible, or will not work. Then we can use AGR just on the Top-Window:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
AGR.40||google&ANDTHEN:nwshp
MMV.
ENR.
```

Which may be a bit slower in execution times, depending on the overall complexity of the WEB-/Office program.

© 2023 Theo Gottwald * https://www.smart-package.com

If the target application is an Internet explorer, AGR. will scroll to bring the located object into the visible area.

AGR. supports the use of so called "Role-Parameter". These are Inline-Subcommands which get executed only if the object could be found. Here is an example:

```
' The "Inline MOS." will select the Link in the IE
AGR.40!MOS.||google&ANDTHEN:nwshp
```

For a complete list of all "Role-Parameters" see below.

Hints:
It is important, that you localize a window before using AGR.

Unlike SAO., AGR. does not use the 1.4. The Prototype Parameter System 73.

AGR. supports 3.2 Standard-Search Pattern 341 and they can help you to better define the things you look for. See example above "google&ANDTHEN:nwshp".

Please note that AGR. can be quite slow when being used on complex applications and slow Computers.

Use the Acc.-Inspector in the Editor to get the necessary parameter data.

If you use the AGR.-Command for a**utomated Installations**, like for Office 2007 or Office 2010, its absolutely necessary that you specify the **Flag 19.**
Otherwise the AGR. Command will localize invisible elements and the installation will not work as expected.

Syntax

```
AGR.[P1][|P2...|Px]
AGF.[P1][|P2...|Px]
```

Parameter Explanation

P1 - (optional)* Role, and Role-Parameter,** a numeric value as you it in the Acc.-Inspector, when you are over the element with the mouse. ***You can specify additional "Subcommands" after the Role-Parameter* (See below), separated by a "!". Example:

```
AGR.40!WII.MLE.||google&ANDTHEN:nwshp
```

***P2 - (optional)* Name**, a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse.

***P3 - (optional)* Value,** a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse.

© 2023 Theo Gottwald * https://www.smart-package.com

***P4 - (optional)* Number** of the element. There may be multiple elements that fit your description. In that case you can choose which one you want using this parameter. You need to try which number you need to locate the element of your choice.

***P5 - (optional)* Flags.** A numeric value that is a combination of the below numbers.

1 - only Objects,
2 - only Simple Elements
4 - display debug info while searching.
8 - if set, then elements with the %STATE_SYSTEM_INVISIBLE will also be found.
16 - if set, then elements with the %STATE_SYSTEM_INVISIBLE or the %STATE_SYSTEM_UNAVAILABLE will also be found. Needed for Setup automation, like MS-Office 2007 or 2010.
32 - NoScroll-Flag. Using this Flag, you can permit AGR to scroll in the Internet Explorer.

while all parameters are marked as "optional", you need to specify some of them. An empty AGR. does not make any sense.

***P6 - (optional)* Waiting Time.** if given, AGR. will wait the specified amount of time if that specified Object will appear. Needed to wait for WEB-Sites or in Setup Scripts.

***P7 - (optional)* On NOT Found Jump to.** if given, AGR. will jump to the specified Label, if that specified Object is not found in the specified time.

***Role-Parameter:**

You can specify multiple Roles, they must be separated by ",".
Any of these will be valid.

If you specify a "#" before the Role-Parameter, it will mean "Role-Group" and specifies any of the following:
0: unknown
1: User-bedienbare Controls
2: Panels, Client und Background
4: Borders etc.
8: Window-Controls (Titlebar, Minimize etc.)
16: Scrollbars
32: Spezial, Cursor, Sound etc.

These inline-commands can be specified together with the role-parameter, separated by a "!". You can specify as man of them as you wish. They will only get executed, in case of an element is found:

```
"TAF" - "a" ' Take Focus
"TAS" - "b" ' Take Selection
"ESE" - "c" ' Ext. Select
"ASE" - "d" ' Add to Selection
"RSE" - "e" ' Remove Selection
"SSE" - "f" ' Strong Select
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 "SHC" - "g" ' Shift-Click
 "CCS" - "h" ' ctrl-click - select
 "CSE" - "i" ' ctrl-click cancel selection
 "MER" - "j" ' MER at Pos.
 "MSE" - "k" ' do Multiselect
 "MOS" - "m" ' Mouseless Operation Select
 "DMO" - "n" ' Double Mouseless Operation Select
 "MLI" - "o" ' MLI at place
 "DMR" - "p" ' Double Mouseless Operation Select with rest
 "MPO" - "q" ' Mousepointer Position Set
 "MSO" - "r" ' Mouse Operation
 "MDO" - "s" ' Mouse Double Operation
 "WIN" - "t" ' Set Wait-Input Idle- Time to infinte
 "MDC" - "u" ' MDC at place
 "MRC" - "v" ' MRC at place
 "WII" - "w" ' Wait-for-Input-Idle (max. 5 Seconds)
 "MLE" - "x" ' MLE.
 "SMP" - "y" ' Set Mousepos Mode
 "BTP" - "z" ' Set Left-ControlPosition like for Checkmark
 "BTC" - "0" ' check State before click and click
               if actually state=0 (check it)
 "BTS" - "1" ' check State before click and click if actually
               state=1 (uncheck it)
```

Example

```
' This Example will find an item in the Internet Explorer
' on a WEB-Site
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
AGR.40||google&ANDTHEN:nwshp|1|64
MMV.
ENR.
```

Remarks

-

Limitations:

This command can be slow on complex applications. In such case use AGF. insteead of AGR.
You can not use this command to wait for a WEB-Site to be opened as it tries to analyze internal structures that are going to be rebuild in the WEB-Site.
Using this commands on WEB-Sites that do just load/rebuild internal structures may lead to all sorts of crashes.
Using the command on a WEB-Site that is going to be opened, may therefore result in a unrecoverable crash of the robot program.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591
- SAO. - Search Accessible Object 625
- AFT. - Acc.- Find Text 558
- KBC. - KeyBoard Command 2252

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.5 AMS. - Acc.-Multi Scan

AMS. - Acc.- Multi Scan (Visual Acc.-Object Scan) Previous 564 Top 41 Next 582

MiniRobotLanguage (MRL)

AMS. Command

Scan for Accesible Object

Intention

This command localizes a specified Acc.-Object in a Child-Window or Top-Window and sets the actual point into the object center.

Unlike `SAO.,AFT.,AGR.` etc. `AMS.` uses a scanning technique instead of a search technique.
Therefore `AMS.` can find Acc.-Objects that can not be found using any other command. Technically the `AMS.` does the same, like when you use the Acc.-Inspector from the Editor, and drive through the window, scanning for a object.
Therefore only Objects that are currently visible and on Top can be scanned/searched using `AMS.`.

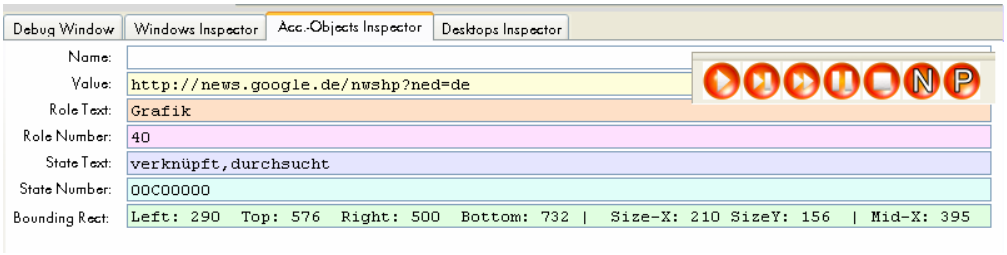
You can see Acc-Objects in the Acc.-Inspector. If you see Acc.-Objects in teh Acc.-Inspector, but you can not localize them with any of the search commands, then `AMS.` can get them for you. This is where `AMS.` can be used.

`AMS.` uses multiple CPU-Cores at the same time to speed things up. Therefore the time, `AMS.` needs to locate an object depends heavily on the size of the window, and as well on the number and speed of the CPU cores.

Let us see an example from a WEB-Site in the Internet Explorer:

In the picture below you will see the resulting Display in the Acc.-Inspector.

© 2023 Theo Gottwald * https://www.smart-package.com



The most important parameters for AMS. are:
"**name**", "**role**" and "**value**" and we also have "**Step**". In most cases you can ignore the other parameters.

Now we'll take the above picture from the Acc.-Analyzer, and make the code for it.

Using AMS. we do best locate a Top- and a child-window. Like this:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
AMS.40||google&ANDTHEN:nwshp
MMV.
ENR.
```

In some cases this may not be possible, or will not work. Then we can use AGR just on the Top-Window:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet Explorer
AMS.40||google&ANDTHEN:nwshp|1|25
MMV.
ENR.
```

Which may be a bit slower in execution times, depending on the overall complexity of the WEB-/Office program.

If the target application is an Internet explorer, AMS. will not scroll to bring the located object into the visible area. It will not find any element that is not completely visible for scanning. Therefore use the SWP. command, to make sure that the window has a sufficient size.

AMS. will never work behind a cover, like the cover that can be created using the UCV. command.

AMS. supports the use of so called "Role-Parameter". These are Inline-Subcommands which get executed only if the object could be found. Here is an example:

```
' The "Inline MOS." will select the Link in the IE
AMS.40!MOS.||google&ANDTHEN:nwshp
```

For a complete list of all "Role-Parameters" see below.

Hints:
It is important, that you localize a window before using AMS.

© 2023 Theo Gottwald * https://www.smart-package.com

Unlike SAO., the AMS. command does not use the 1.4. The Prototype Parameter System 73.

AMS. supports 3.2 Standard-Search Pattern 341 and they can help you to better define the things you look for. See example above "google&ANDTHEN:nwshp".

Please note that AMS. can be quite slow when being used on large windows and with slow Computers. AMS. will speed up with multiple CPU cores, the more the better.

Use the Acc.-Inspector in the Editor to get the necessary parameter data.

Syntax

AMS.[P1][|P2...|Px]

Parameter Explanation

P1 - (optional)* Role, and Role-Parameter,** a numeric value as you it in the Acc.-Inspector, when you are over the element with the mouse. ***You can specify additional "Subcommands" after the Role-Parameter* (See below), separated by a "!". Example:

```
AMS.40!WII.MLE.||google&ANDTHEN:nwshp
```

***P2 - (optional)* Name**, a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse.

***P3 - (optional)* Value,** a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse.

***P4 - (optional)* Number** of the element. There may be multiple elements that fit your description. In that case you can choose which one you want using this parameter. You need to try which number you need to locate the element of your choice.

***P5 - (optional)* Step.** A numeric value that is the raster-size, of the scanning-net. The higher this parameter, the faster AMS. will work. Imagine the Step parameter like the size of the holes in a fisher net. The larger the holes, the bigger the fishes have to be, to be caught. The default value is 5. A good number would be, a bit less then the half size of the element you are looking for, in pixels.

While all parameters are marked as "optional", you need to specify some of them. An empty AGR. does not make any sense.

***Role-Parameter:**
These inline-commands can be specified together with the role-parameter, separated by a "!". You can specify as man of them as you wish. They will only get executed, in case of an element is found:

```
"TAF" - "a" ' Take Focus
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"TAS" - "b" ' Take Selection
"ESE" - "c" ' Ext. Select
"ASE" - "d" ' Add to Selection
"RSE" - "e" ' Remove Selection
"SSE" - "f" ' Strong Select
"SHC" - "g" ' Shift-Click
"CCS" - "h" ' ctrl-click - select
"CSE" - "i" ' ctrl-click cancel selection
"MER" - "j" ' MER at Pos.
"MSE" - "k" ' do Multiselect
"MOS" - "m" ' Mouseless Operation Select
"DMO" - "n" ' Double Mouseless Operation Select
"MLI" - "o" ' MLI at place
"DMR" - "p" ' Double Mouseless Operation Select with rest
"MPO" - "q" ' Mousepointer Position Set
"MSO" - "r" ' Mouse Operation
"MDO" - "s" ' Mouse Double Operation
"WIN" - "t" ' Set Wait-Input Idle- Time to infinte
"MDC" - "u" ' MDC at place
"MRC" - "v" ' MRC at place
"WII" - "w" ' Wait-for-Input-Idle (max. 5 Seconds)
"MLE" - "x" ' MLE.
"SMP" - "y" ' Set Mousepos Mode
"BTP" - "z" ' Set Left-ControlPosition like for Checkmark
"BTC" - "0" ' check State before click and click
              if actually state=0 (check it)
"BTS" - "1" ' check State before click and click if actually
              state=1 (uncheck it)
```

P6 **-** ***(optional)*** **Waiting Time.** if given, AGR. will wait the specified amount of time if that specified Object will appear. Needed to wait for WEB-Sites or in Setup Scripts.

P7 **-** ***(optional)*** **On NOT Found Jump to.** if given, AGR. will jump to the specified Label, if that specified Object is not found in the specified time.

Example

```
' Example comparing AVF. with AMS
'
STW.ct|IEFrame|Theo Gottwald
AVF.40||ing-diba|1|70
MMV.
AMS.40||ing-diba|1|70
MMV.
DMP.1
MBX.1
ENR.

' Choosing a raster size of 40 is possible for long elements
' and will speed up search significantly
'
STW.ct|IEFrame|theo&ANDTHEN: - Windows Internet Explorer
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
AMS.40||focus|1|40
MMV.
ENR.
```

Remarks

This command will speed up with a growing number of CPU-Cores.

Limitations:

This command can be slow on large windows.

See also:

- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591
- SAO. - Search Accessible Object 625
- AFT. - Acc.- Find Text 558
- AGR. - Accessibility Get Rect 564
- Acc.-Visual Find 575

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.6 AVF. - Accesibility Visual Find

AVF. - Acc.- Visual Find (Visual Acc.-Object Scan) Previous 564 Top 41 Next 582

MiniRobotLanguage (MRL)

AVF. Command

Scan for Accesible Object

Intention

This command localizes a specified Acc.-Object in a Child-Window or Top-Window and sets the actual point into the object center.

Unlike `SAO.,AFT.,AGR.` etc. `AVS.` uses a scanning technique instead of a search technique.
Therefore `AVF.` can find Acc.-Objects that can not be found using any other command. Technically the `AVF.`
does the same, like when you use the Acc.-Inspector from the Editor, and drive through the window, scanning for a object.

Therefore only Objects that are currently visible and on Top can be scanned/searched using `AVF.`
You can see Acc-Objects in the Acc.-Inspector. If you see Acc.-Objects in the Acc.-Inspector,
but you can not localize them with any of the search commands, then `AMS.` or `AVF.` can get them for you. This is where `AMS.` and `AVF.` can be used.

`AVF.` uses multiple CPU-Cores at the same time to speed things up.
Therefore the time, `AVF.` needs to locate an object depends heavily on the size of the window, and as well on the number and speed of the CPU cores.

AMS. and AVF. use different internal Implementation. Also AVF. leaves other information on the TOS.
In terms of Speed both are similar fast.

```
010 0000000018315666 0001 STW.ct|#32770|PB/Win IDE 10.04 beta build
011 0000000009056948 0001 SCW.nict|1|302|ToolbarWindow32|{&NOTEXT:}
012 0000000024558055 0001 AVF.43|Print
013 0000000026442359 0001 AMS.43|Print
014 0000000029818745 0001 MMV.
015 0000000000000000 0001 DMP.1
016 0000000000000000 0000 MBX.1
```

In this Sample AVS. is a bit faster

© 2023 Theo Gottwald * https://www.smart-package.com

```
012 0000000000225522 0001 STW.ct|IEFrame|Theo Gottwald
013 0000000515178748 0001 AVF.40||ing-diba
014 0000000029706078 0001 MMV.
015 0000000479818740 0001 AMS.40||ing-diba
016 0000000012501280 0001 MMV.
017 0000000000000000 0001 DMP.1
018 0000000000000000 0000 MBX.1
019 0000000000000000 0000 ENR.
```

In this Sample AMS. is faster

Here are some Speed Considerations:
First we use a step of 7 (Default)

```
012 0000000000700386 0001 STW.ct|IEFrame|Theo Gottwald
013 0000000554899556 0001 AVF.40||ing-diba|1|7
014 0000000029917839 0001 MMV.
015 0000000887212953 0001 AMS.40||ing-diba|1|7
016 0000000029778553 0001 MMV.
017 0000000000000000 0001 DMP.1
018 0000000000000000 0000 MBX.1
019 0000000000000000 0000 ENR.
```

Here we use a Step-Value of 7

Second we use a step of 70: Note that AVF. will internally scale down if it does not find anything until the Step-Value is smaller then 4.

```
012 0000000000208610 0001 STW.ct|IEFrame|Theo Gottwald
013 0000000027980512 0001 AVF.40||ing-diba|1|70
014 0000000012524153 0001 MMV.
015 0000000128987777 0001 AMS.40||ing-diba|1|70
016 0000000029821614 0001 MMV.
017 0000000000000000 0001 DMP.1
018 0000000000000000 0000 MBX.1
019 0000000000000000 0000 ENR.
```

Here we use a Step-Value of 70 which is significantly faster.

Let us see an example from a WEB-Site in the Internet Explorer:

*© 2023 Theo Gottwald * https://www.smart-package.com*

In the second picture above you see the resulting Display in the Acc.-Inspector.

The most important parameters for AVF. are:
"**name**", "**role**" and "**value**" and we also have "**Step**".
Now we'll take the above picture from the Acc.-Analyzer, and make the code for it.

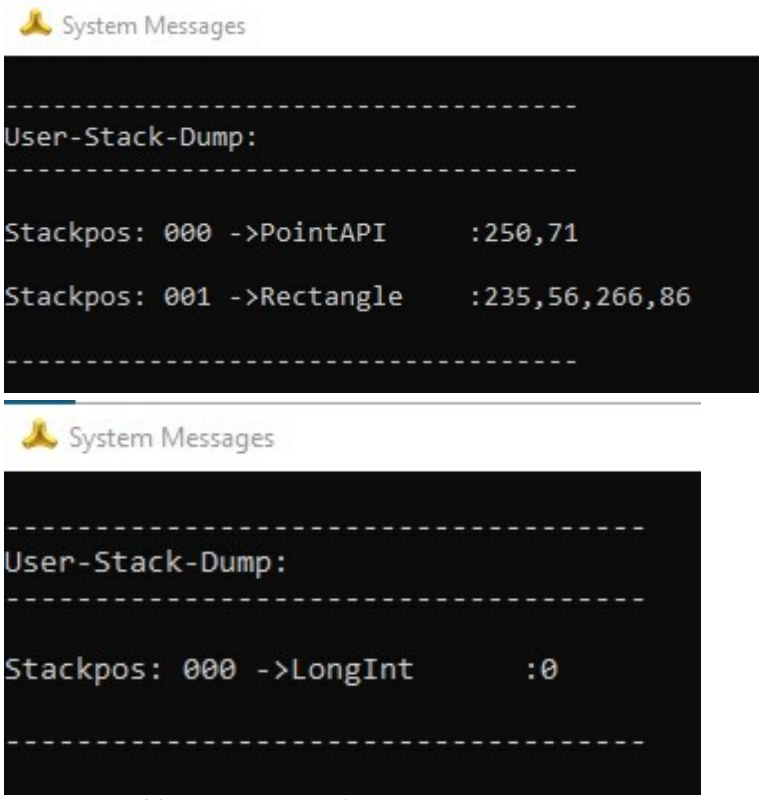
Using AVF. we do best locate a Top- and a child-window. Like this:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
AVF.40||google&ANDTHEN:nwshp
MMV.
ENR.
```

In some cases this may not be possible, or will not work. Then we can use AGR just on the Top-Window:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
AMS.40||google&ANDTHEN:nwshp|1|25
MMV.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



This is the TOS when we have found something.
This is the TOS when we did not find anything.

Which may be a bit slower in execution times, depending on the overall complexity of the WEB-/Office program.

If the target application is an Internet explorer, `AMS.` will not scroll to bring the located object into the visible area. It will not find any element that is not completely visible for scanning. Therefore use the `SWP.` command, to make sure that the window has a sufficient size.

Use `AVF.` with a larger Step value, for example the Size of the object. It will automatically scale down internally if it does not find anything.
This is the major difference to `AMS.`

`AVF.` will never work behind a cover, like the cover that can be created using the `UCV.` command.

`AVF.` supports the use of so called "Role-Parameter". These are Inline-Subcommands which get executed only if the object could be found. Here is an example:

```
' The "Inline MOS." will select the Link in the IE
AVF.40!MOS.||google&ANDTHEN:nwshp
```

For a complete list of all "Role-Parameters" see below.

Hints:
It is important, that you localize a window before using `AVF.`
Unlike `SAO.`, the `AVF.` command does not use the 1.4. The Prototype Parameter System 73.

 © 2023 Theo Gottwald * https://www.smart-package.com

AVF. supports 3.2 Standard-Search Pattern 341 and they can help you to better define the things you look for. See example above "google&ANDTHEN:nwshp". Please note that AVF. can be quite slow when used on large windows and with slow Computers. AVF. will speed up with multiple CPU cores, the more the better.

Keep your Target-Window as small as possible when using AVF. or AMS.

Use the Acc.-Inspector in the Editor to get the necessary parameter data.

Syntax

AVF.[P1][|P2...|Px]

Parameter Explanation

P1 - (optional)* Role, and Role-Parameter,** a numeric value as you it in the Acc.-Inspector, when you are over the element with the mouse. ***You can specify additional "Subcommands" after the Role-Parameter* (See below), separated by a "!". Example:

```
AMS.40!WII.MLE.||google&ANDTHEN:nwshp
```

***P2 - (optional)* Name,** a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse.

***P3 - (optional)* Value,** a string / text as you it in the Acc.-Inspector, when you are over the element with the mouse.

***P4 - (optional)* Number** of the element. There may be multiple elements that fit your description. In that case you can choose which one you want using this parameter. You need to try which number you need to locate the element of your choice.

***P5 - (optional) Pixel*-Step.** A numeric value that is the raster-size, of the scanning-net. The higher this parameter, the faster AVF. will work.

Imagine the Step parameter like the size of the holes in a fisher net. The larger the holes, the bigger the fishes have to be, to be caught.

The default value is 5. A good number would be, a bit less then the half size of the element you are looking for, in pixels.

Special when using AVF:
Use AVF. with a larger Step value, for example the Size of the object. It will automatically scale down internally if it does not find anything.
This is the major difference to AMS. Technically AVF: will run with the given large Step-Value and if it did not find anything it will run again with the step value of half the size until the step values is below 3.

If the Object can not be found this may take a while, even a Minute. If you do not want that use AMS. instead.

***Role-Parameter:**

© 2023 Theo Gottwald * https://www.smart-package.com

These inline-commands can be specified together with the role-parameter, separated by a "!". You can specify as man of them as you wish. They will only get executed, in case of an element is found:

```
"TAF" - "a" ' Take Focus
"TAS" - "b" ' Take Selection
"ESE" - "c" ' Ext. Select
"ASE" - "d" ' Add to Selection
"RSE" - "e" ' Remove Selection
"SSE" - "f" ' Strong Select
"SHC" - "g" ' Shift-Click
"CCS" - "h" ' ctrl-click - select
"CSE" - "i" ' ctrl-click cancel selection
"MER" - "j" ' MER at Pos.
"MSE" - "k" ' do Multiselect
"MOS" - "m" ' Mouseless Operation Select
"DMO" - "n" ' Double Mouseless Operation Select
"MLI" - "o" ' MLI at place
"DMR" - "p" ' Double Mouseless Operation Select with rest
"MPO" - "q" ' Mousepointer Position Set
"MSO" - "r" ' Mouse Operation
"MDO" - "s" ' Mouse Double Operation
"WIN" - "t" ' Set Wait-Input Idle- Time to infinte
"MDC" - "u" ' MDC at place
"MRC" - "v" ' MRC at place
"WII" - "w" ' Wait-for-Input-Idle (max. 5 Seconds)
"MLE" - "x" ' MLE.
"SMP" - "y" ' Set Mousepos Mode
"BTP" - "z" ' Set Left-ControlPosition like for Checkmark
"BTC" - "0" ' check State before click and click
              if actually state=0 (check it)
"BTS" - "1" ' check State before click and click if actually
              state=1 (uncheck it)
```

***P6 - (optional)* Waiting Time.** if given, AGR. will wait the specified amount of time if that specified Object will appear. Needed to wait for WEB-Sites or in Setup Scripts.

***P7 - (optional)* On NOT Found Jump to.** if given, AGR. will jump to the specified Label, if that specified Object is not found in the specified time.

Example

```
' This Example will find an item in the Internet Explorer
' on a WEB-Site and select that item mouse-less.
STW.ct|IEFrame|theo&ANDTHEN: - Windows Internet Explorer
AMS.40!MOS.||focus
ENR.


' Choosing a raster size of 40 is possible for long elements
' and will speed up search significantly
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|IEFrame|theo&ANDTHEN: - Windows Internet Explorer
AMS.40||focus|1|40
MMV.
ENR.
```

Remarks

This is one of the Slowest commands, therefore keep the size of the target-window as small as possible.
Increase P4 to half the size of the Object that you want to search, this will speed up the command significantly.

This command will also speed up with a growing number of CPU-Cores.
If possible use the new Pattern-Search Commands or AGR. instead of these Commands.
In a Target Window 576x1090 i received scanning times of up to 17 Seconds.

Limitations:

This command can be slow on large windows.

See also:

- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591
- SAO. - Search Accessible Object 625
- AFT. - Acc.- Find Text 558
- AGR. - Accessibility Get Rect 564
- AMS. - Acc-Multi-Scan 570

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.7 CBR. - Control Browser

MiniRobotLanguage (MRL)

CBR. Command

Send or receive Data from/to the Mini-Browser

Intention

Using `CBR.` you can control the Mini-Browser (=*MB*). For this to work, the *MB* must have been started using the `LBR.` command. This ensures that there is a technical connection between the Script and the MB.

`CBR.` can switch or close sites in the MB. It can even give you the HTML or Text that is currently on the WEB-Site.

For more details on the Mini-Browser (MB) 554 see there.

One special usage for `CBR.` is
`CBR.errclear`

Any error tha occurs while loading a site will stay until it is "confirmed" from the script. This is to make sure that you have seen that error.
`CBR.errclear`
Will clear the internal Error variables. Note that in case of successive Errors, the last will overwrite any previous Error, no matter if that one was confirmed or not.

Please see the 3.1 Systemvariables 334 for more Information on Errorhandling with the MB.

`CBR.` is a multi-purpose command that includes multiple possible subcommands. Usage is simple:

```
' Start the Mini-Browser
LBR.Size=1024,980|Pos=0,0|svf=0|Site=www.it-berater.org
' Wait until the Starting Site is loaded,
' please look at the Logo on the site, left upper corner.
' The loading will not be "done" until it has been
' completely aninmated!
WBR.
' Use CBR to open a new site also
CBR.new|www.fa2.de
' Wait until that site is loaded
WBR.
' Select the first TAB
CBR.select|1
' Switch the selected TAB to this site
CBR.Goto|www.smart-package.com
WBR.
' Wait 5 seconds
PAU.5
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Select the second TAB
CBR.select|2
' Cose it
CBR.Close
WBR.
MBX.Ready
' then the SPR-Browser will close
' because we started it with "svf=0"
ENR.
```

Syntax

CBR.P1[|P2]

Parameter Explanation

P1 – `Subcommand`, the following Subcommands are currently supported. Note that for all of the subcommands there is a long form that is better readable, as well as a short form, that can be used to write faster.

- `"New"` - open a new TAB in the MB
  ```
  CBR.New[|page to load]
  CBR.New
  CBR.New|about:blank
  CBR.New|www.fa2.de$crlf$www.c10.de
  CBR.New|$$TXT
  ```

- `"Close","Cl."` - Close the active TAB
  or close a TAB using a numeric Index (1..X)
  ```
  CBR.Close
  CBR.Close|(number)
  CBR.Close|2
  CBR.Close|$$NUM
  ```

- `"TClose","tc."` - Close a TAB using a pattern
  ```
  CBR.TClose|$$PAT
  CBR.TClose|Theo$OR:Site
  ```

- `"Select","s."` - Select a TAB using a number (1..X)
  ```
  CBR.Select|$$NUM
  CBR.Select|3
  ```

- `"TSelect","ts."` - Select a TAB using a pattern
  ```
  CBR.TSelect|$$PAT
  CBR.ts.|Theo$OR:Site
  ```

- `"Exit","ex."` - Exit MiniBrowser
  ```
  CBR.Exit
  CBR.ex.
  ```

- `"Goto","gt."` - Goto this site ...

© 2023 Theo Gottwald * https://www.smart-package.com

```
CBR.Goto|$$SIT
CBR.gt.|www.fa2.de
```

- "ErrClear","er." - Confirm / Clear Error
```
CBR.ErrClear
CBR.ErrClear
```

- "gback","gb." - Press Back Button (1 or more times)
```
' Press "Back" 2 times
CBR.gback|2
' Press "Back" 1 time
CBR.gback
```

- "gforward","gf." - Press Forward Button (1 or more times)
```
' Press "Forward" 2 times
CBR.gforward|2
' Press "Back" 1 time
CBR.gf.
```

- "GHome","gh." - Load Default page
```
CBR.GHome
CBR.gh.
```

- "gsearch","gs." - Load Search Site
```
CBR.gsearch
CBR.gs.
```

- "refresh","rf." - Load actual Site again/refresh Site
```
CBR.refresh
' refresh 2 times
CBR.rf.|2
```

- "silent","si." - Set/Reset IE-Silent Mode
```
CBR.silent|1
CBR.si.|0
```

- "SetFocus","sf." - SetFocus on current Tab or numeric TAB
```
' SetFocus on TAB 1
CBR.SetFocus|1
' SetFocus on current TAB
CBR.sf.
```

- "iHTML","ih." - Get Site HTML
```
' Get Site HTML into Variable
CBR.iHTML|$$HTM
' Get HTML on TOS
CBR.ih.
```

- "iText","it." - Get Site Text
```
' Get Site Text into Variable
CBR.iText|$$TXT
' Get Text on TOS
CBR.it.
```

- "GetZoom","gz." - Get Site Zoom-Factor (10-1000)

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Get Site Zoom into Variable
CBR.GetZoom|$$TXT
```

- "SetZoom","sz." - Set Site Zoom-Factor (10-1000)
```
' Set Site Zoom
CBR.sz.|200
' Formulas are allowed
CBR.sz.|(10*$$TXT)
```

- "getwindow","gw." - Locate selected TAB Browser-Control
This will locate the Browser Control like if you would have used SCW.
```
' Locate Control and put Control Handle into $$TXT
CBR.getwindow|$$TXT
' Locate Control and place Control Handle on TOS
CBR.it.
```

- "ScrollTo","st." - Scroll to defined Position
```
CBR.ScrollTo|45,344
' Formulas in () are allowed
CBR.ScrollTo|(45+$$NUM),(344-$$IXX)
```

- "XScroll","xp." - Get actual X-Scrolling Position
```
' Get X-Pos. into Variable
CBR.xp.|$$TXT
```

- "YScroll","yp." - Get actual Y-Scrolling Position
```
' Get Y-Pos. into Variable
CBR.yp.|$$TXT
```

- "XSize","xs." - Get actual X-Scrolling Size
```
' Get X-Size. into Variable
CBR.xs.|$$TXT
```

- "YSize","ys." - Get actual Y-Scrolling Size
```
' Get Y-Size. into Variable
CBR.ys.|$$TXT
```

- "Scroll","sc." - Scroll Browser
```
' Scroll PageDown 4 times
CBR.Scroll|pageDown,4
' Scroll 1 time PageUp
CBR.Scroll|scrollbarPageUp
```

The "Scroll" Subcommand has additional options. With some of these, you can give an additional numeric parameter that specifies how often you want to to the specified action.

- "scrollbarDown","Down" – scroll down, like a click on the Scrolldown Button
```
' Press "Scrolldown" 1 time
CBR.Scroll|scrollbarDown,1
```

- "scrollbarUp","Up" – scroll Up, like a click on the Scroll-Left Button
```
' Press "scrollbarUp" 2 times
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  CBR.Scroll|scrollbarUp,2
```

- "scrollbarLeft","left" - scroll left, like a click on the Scroll-Left Button

```
' Press "scrollbarLeft" 2 times
  CBR.Scroll|scrollbarLeft,2
```

- "scrollbarRight","Right" - scroll right, like a click on the Scroll-Right Button

```
' Press "scrollbarRight" 2 times
  CBR.Scroll|scrollbarRight,2
```

- "scrollbarPageDown" - scroll full page down

```
' Press "scrollbarPageDown" 1 times
  CBR.Scroll|scrollbarPageDown,1
```

- "scrollbarPageUp" - scroll full page Up

```
' Press "scrollbarPageUp" 2 times
  CBR.Scroll|scrollbarPageUp,2
```

- "scrollbarPageRight" - scroll full page to right

```
' Press "scrollbarPageRight" 2 times
  CBR.Scroll|scrollbarPageRight,2
```

- "scrollbarPageLeft" - scroll full page to left

```
' Press "scrollbarPageLeft" 2 times
  CBR.Scroll|scrollbarPageLeft,2
```

Depending on the chosen Subcommand the further Parameters **P2** to **P4** differ and may be needed or not.

P2 - (optional) Depending on the chosen Subcommand this Parameter may be needed or not.

Example

```
'************************************
' CBR. Sample Locate WEB-Elements
' and compute Link
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
WBR.
CBR.gw.
FOR.$$NUM|1|40
  AGR.40|||$$NUM
  GAI.v|$$LNK
  DBP.Link ($$CNT):$crlf$$$LNK
  MMV.
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.Browser exit
CBR.exit
ENR.


'************************************
' CBR. Sample 1 (New and Close)
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0
PAU.2
' Open 2 new default sites
CBR.New
CBR.New
' Open a new blank page
CBR.New|about:blank
PAU.5
CBR.Close
CBR.Close
CBR.Close
PAU.5
$$TXT=www.fa2.de$crlf$www.facebook.com
CBR.New|$$TXT
WBR.
PAU.5
ENR.

'************************************
' CBR. Sample "Close, TClose and Exit"
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0
PAU.2
CBR.New|www.fa2.de
CBR.New|www.it-berater.org
' open a blank page
CBR.New|about:blank
WBR.
MBX.Closing TAB by textual pattern
CBR.TClose|Theo&AND:Site
MBX.Closing Tab by Number (1...x)
CBR.Close|1
MBX.Closing active TAB
CBR.Close
MBX.Browser exit
cbr.exit
MBX.Browser is gone!
ENR.


'************************************
' CBR. Sample "Select, TSelect and Exit"
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0
PAU.2
CBR.New|www.fa2.de
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CBR.New|www.it-berater.org
' open a blank page
CBR.New|about:blank
WBR.
MBX.Selecting TAB by textual pattern
CBR.TSelect|Theo&AND:Site
MBX.Selecting Tab by Number (1...x)
CBR.Select|1
MBX.Closing active TAB
CBR.Close
MBX.Browser exit
CBR.exit
MBX.Browser is gone!
ENR.

'************************************
' CBR. Sample "Text, iHTML and Exit"
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0
' open a blank page
CBR.New|about:blank
WBR.
' Switch active TAB to this page
CBR.Goto|www.fa2.de
WBR.
CBR.iText|$$TXT
MBX.Text from this Site: $crlf$$$TXT
' Open another TAB
CBR.New|www.it-berater.org
WBR.
CBR.ihtml|$$TXT
MBX.HTML-Source of this Site: $crlf$$$TXT
MBX.Closing active TAB
CBR.Close
MBX.Browser exit
CBR.exit
MBX.Browser is gone!
ENR.


'************************************
' CBR. Sample "Get Offset"
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
WBR.
CBR.ho.|$$HOF
CBR.vo.|$$VOF
MBX.Hor. Offset=$$HOF$crlf$Vert. Offest=$$VOF
MBX.Browser exit
CBR.exit
ENR.

'************************************
' CBR. Sample "ScrollTo"
'************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
WBR.

FOR.$$CNT|1|2000|20
  GSB.check
  CBR.Scrollto|$$CNT,1
  GSB.check
  CBR.Scrollto|1,$$CNT
  GSB.check
NEX.

MBX.Browser exit
CBR.exit
ENR.

' Subprogram
:check
' Find out Scroll-Positions
CBR.xp.|$$TXT
DBP.Xpos=$$TXT
CBR.yp.|$$TXT
DBP.Ypos=$$TXT
RET.

'************************************
' CBR. Sample "Get-/Set Zoom"
'************************************
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
WBR.

FOR.$$CNT|10|1000|10
  CBR.SetZoom|$$CNT
  CBR.GetZoom|$$TXT
  DBP. Zoom is: $$TXT
NEX.

MBX.Browser exit
CBR.exit
ENR.
```

Remarks

At nearly all places, where you need numerical parameters, you can also use Formulas in (). Even if it is not explicitly stated in the description.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

The two Scroll-commands (ScrollTo/Scroll) depend to some degree on the used WEB-Environment/IE-Control. However one of them should work in any environment. In most circumstances both should work.

See also:

- ! Smarty's Browser and Office Commands 554
- LBR. - Launch (Mini-)Browser 601
- CBR. - Control Browser 582
- WBR. - Wait Browser 630
- IBR. / NBR. - If Browser 595
- GAI. - Get-Acc-Information 591

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.8 GAI. - Get-Acc-Information

MiniRobotLanguage (MRL)

GAI. Command

Get accessible information

Intention

This command read information from the actual selected point/object. This can be for example a text in a window or browser or office-program.

Technically it reads accessible information from the Screen Objects at the **Actual Point**.
It can be the state of a element, or its size (location).
It will also return the position of the actual point if the prototype "**p**" is specified.

`GAI.` is slower then `CFW` but it may return more Details and can read also where `CFW.` might not.

Internally `GAI.` calls MAP 2483. which makes an separate call to MMV - Mouse MoVe 2466., MPO. - Mouse-POsition move. or MAP - Move Active Point 2483. obsolete.

In the easiest form, `GAI.` will read texts from a screen-position onto `TOS`.
In this example we read text from the Internet Explorer.

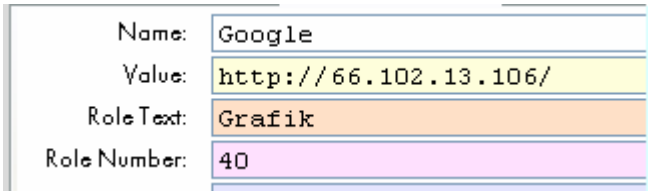
```
' In this case we had the page www.fa2.de open
' in the internet explorer. Change the window-title:
STW.t|internet
' and the SAO.-parameter to your needs
' SAO. locates a point in the IE window
SAO.nvw|google|http|40
' GAI. reads that point's texts
GAI.
' Now we gave the results on the stack
STS.DUMP
```

Now we find the results from `GAI.` on the TOS. Here:

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->String        :Google
Stackpos: 001 ->String        :http://66.102.13.106/

-----------------------------------
```

These are the same values, that you can see in the windows-inspector in the editor, when the mouse is exactly at that place.

© 2023 Theo Gottwald * https://www.smart-package.com



Next, you can use GAI. with parameters to get the values directly into variables.

Syntax

GAI.P1|P2[|P3]...[|PX]

Parameter Explanation

```
Px - (optional) l,n,p,r,s,v -  output-prototypes
      Output variables according to prototype(s)

 l - Get Item Location, this will result in a RECT with the
     item-Coordinates
 n - Get the Item Name. Text result.
 p - Position, this will return the (x,y) from the Actual
Point-
     Position which was used for the Operation. You get a
     point-variable-result.
 r - Get Element "Role".  Numeric result.
 s - Get Element State. Numeric result.
 v - Get Element Value. Text (or numeric) result.
```

Example

```
'******************************************************
' Example 1 GAI.
'******************************************************
LBR.Size=1024,980|Pos=0,0|svf=0|Site=www.fa2.de
WBR.
CBR.gw.
AGR.40||powerbasi
MMV.
GAI.v|$$LNK
DBP.$$LNK

'******************************************************
' Example 2 GAI.
'******************************************************
EXS.iexplore|www.fa2.de
PAU.3
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.t|theo gott
SWP.|600,600
SAO.nvw|google|http|40
GAI.
' Now we take it from the stack and display it each
' Note that we use $$001 twice because once its touched,
' its taken from the stack!
DBP.Name: $$001-- Value:$$001
MBX.Ready
' Here we close the IE
GCT.
CLW.
ENR.

'******************************************************
' Example 3 GAI.
'******************************************************
' We open a Browser-Window and search for the Google-Logo
EXS.iexplore|www.fa2.de
PAU.3
STW.t|theo gott
SWP.|600,600
SAO.nvw|google|http|40
' GAI. - Example 1
' This will put Value and Name on the TOS
GAI.
' Now we take it from the stack and display it each
DBP.Name: $$001-- Value:$$001

' GAI. - Example 2
' This will get the state into the given Variable
GAI.$$STA
' Now we take it from the stack and diplay it each
PRT. The state of the Element at this place is: $$STA

' GAI. - Example 3
' Normal use, with prototypes
' This time we will put Value and Location into variable
GAI.vl|§§VAL|§§LOC
' Now we display it each
PRT.Value: §§VAL. Location:§§LOC

' GAI. - Example 3
' Normal use, with prototypes
' This time we will get Position (actual Point), Location and
State
GAI.pls|§§POS|§§LOC|§§STA
' Now we display it each
PRT.----------------------
PRT. Actual Point: §§POS
PRT. Location:§§LOC.
PRT. State: §§STA
PRT.----------------------
MBX.Ready
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- CFW. - Copy-From-Window 3589
- SAO. - Search Accessible Object 625
- AFT. - Acc.- Find Text 558
- AGR. - Accessibility Get Rect 564
- ! NAV. - NAVigate in Dialog 606

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.9 IBR. / NBR. - If Browser

IBR./NBR. - If Browser, Not If Browser Previous 591 Top 41 Next 599

MiniRobotLanguage (MRL)

IBR. Conditional Statement

If Browser Event/ Site loaded

Intention

Using `IBR.` you can check until a Site had been completely loaded in the Mini-Browser (=MB). `IBR.` can also be used to check for several Browser Events or Errors.

For more details on the Mini-Browser (MB) 554 see there.

`IBR.` is a multi-purpose command that includes multiple possible subcommands. Like with all conditional statements, `IBR.` is used like this:

```
IBR.(parameter)
   ...(this is executed if the condition evaluates to True)
ELS.
   ...(this is executed if the condition evaluates to False
EIF.)
```

Here is an example:

```
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa33.de
WBR.
IBR.Err.=2148270085
  MBX.Browser reported Error
  DBP. Err-Code: #mb_err#
  CBR.errclear
  CBR.exit
ELS.
  MBX.Browser ended regularly
EIF.
ENR.
```

Below you will see more examples on how to check for Errors and loading Progress.

`NBR.` is the negative form of `IBR.`

Syntax

IBR.[P1]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

[!] – If you prefix the Subcommand **P1** with a "**!**" then the Logic is negated and the given
condition is true, if its originally false.

P1 – `Subcommand`, the following Subcommands are currently supported:

- `IBR. without any options is the same as IBR.all`
  `IBR.`
- `"All"` - wait until MB is ready with all Jobs
  `IBR.all`
- "Page" - Wait until the actual Page has been loaded
  `IBR.Page`
- "Job" - Wait until the actual Job is done
  `IBR.Job`

The following Subcommands support Comparisons **!,=,>,<**
If you prefix these Subcommands with a "**q**" they will "confirm" any Error like if you had been using the command:

```
CBR.errclear
```

If you omit the `"q"` you will need to manually call

```
CBR.errclear
```

to clear internal Error Variables, and make them ready for the next Event.

- `"Progress","Pro."` - Compare actual loading Progress (0-100%)
  `IBR.Progress=`*`progress value`*
  `IBR.Progress>0`
  `IBR.Progress<19`
  `IBR.Progress<19`

- `"Error","Err."` - Compare actual loading Error State
  `IBR.Err.=0`
  `IBR.Error>0`
  `IBR.Error=0`
  `IBR.Error!200`

- `"State","Sta."` - Compare actual loading Error State as textual pattern 341
  For this textual compare the sign is been ignored.
  `IBR.Sta.=HTTP&AND:Error`
  `IBR.State=HTTPS&OR:Error`
  `IBR.qState=HTTPS&OR:Error`

© 2023 Theo Gottwald * https://www.smart-package.com

- "Description","Des." - Compare loading State as textual pattern[341]
  For this textual compare the sign is been ignored.
  IBR.Des.=HTTP&AND:Error
  IBR.description=HTTPS&OR:Error
  IBR.qDescription=HTTPS&OR:Error

cond - IBR. supports the following conditions:

! - NOT

= - Is Equal to

> - Larger then

< - Smaller then

Hint: Combined Operators like "**>=**" are not supported. In this case, just use ! and "<"
and switch the operators. Because **A>=B** is the same as **NOT B<A**.

Example

```
'##############################
' Sample 2
'##############################
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa33.de
WBR.
IBR.Des.=was not found
  DBP. Descr.: $mb_ert$
  CBR.exit
ELS.
  DBP.No error loading the site
EIF.
ENR.


'##############################
' Sample 2
'##############################
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
:Retry
PAU..25
NBR.
  GTO.Retry
EIF.
CBR.gw.
FOR.$$NUM|1|400
  AGR.40|||$$NUM|||out
  GAI.v|$$LNK
  DBP.Der hinterlegte Link:$crlf$$$LNK
  MMV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEX.
:out
MBX.Browser exit
CBR.exit
ENR.
```

Remarks

There are multiple examples on using the Mini-Browser within the supplied Sample Scripts. A list with possible Errors can be found here:
Internet Explorer Errors and here: HTTP-Status Codes

Limitations:

-

See also:

- ! Smarty's Browser and Office Commands 554
- LBR. - Launch (Mini-)Browser 601
- CBR. - Control Browser 582
- WBR. - Wait Browser 630
- IBR. / NBR. - If Browser 595
- GAI. - Get-Acc-Information 591

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.10 LAP. - Locate at Point

MiniRobotLanguage (MRL)

LAP. Command

Locate any Object/Window at a specified coordinate

Intention

This command will locate a Top- or Child-Window and possibly an "Accessible Object" (~ Browser/Office Object that is not a window) that is at the given Coordinate. Depending on the specified parameters, you do not need to locate any Top-Window prior using this command.

Most often, this command may be used together with `GMP. (Get Mouse Position)` to interact with the object, where the user has placed his mouse-pointer.

The LAP.-command can be used to locate a Top- or Child-Window as well as to locate another non-window (Acc.-)Object.

Syntax

LAP.[P1][|P2,P3]

Parameter Explanation

P1 - (optional) Prototypes, can be one of these:

- ***s*** **-** screen coordinates are specified
- ***o*** **-** ***(Acc-) Object*** coordinates are specified
- ***r*** **-** Coordinates are specified relativ to active point
- ***w*** **-** ***Window*** coordinates are specified
- ***m*** **-** *Use Mouse Coordinates and add specified coordinates in this case Coordinates can be omitted.*

you can add these two prototypes as second prototype:

- **l** **-** Use given coordinates and make left-shift
- **n** **-** Do NOT locate an Acc.-Object - just locate the window.

Example:

```
LAP.sn|44,56
LAP.ml|109,12
```

P2, P3 **-** ***(optional)*** - X- and Y-Coordinates in the Coordinate System that is specified using the above prototypes.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
' Run this script in the Editor, drive with the Mouse on
' an object within the first PAU. command.
' then wait until the LAP. command was executed.
' Now drive the mouse away.
' After 3 seconds the mouse will return to the center-point of
' the Object that was under the mouse while LAP. was called.
' Note that this worked without any need for previous STW.
'
PAU.2
GMP.ms|$$XPO|$$YPO
DBP.$$XPO-$$YPO
LAP.s|$$XPO,$$YPO
PAU.3
MMV.
:Lab_nocon
ENR.
```

Remarks

-

Limitations:

Using invalid coordinates with LAP. may result in strange errors.

See also:

- GMP. - Get-Mouse-Position 2439
- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591
- SAO. - Search Accessible Object 625
- AGR. - Accessibility Get Rect 564

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.11 LBR. - Launch (Mini-)Browser

MiniRobotLanguage (MRL)

LBR. Command

Launch (Mini-)Browser

Intention

This command is used to start the Mini-Browser (=*MB*). It can also connect to a MB that is already running.

If you use `LBR.` to start a new MB then you can specify startup-parameters when using `LBR.` to start the *MB*.

These include:
- the window position
- the window size
- the WEB-/Intranet-Sites to load at startup
- if the MB should automatically close at script-end or not (svf="survive father")

These features are only available when starting the MB using the `LBR.`-command.

To start the MB, a simple

```
LBR.
```

is all you need. If you want to specify that a site should be loaded immediately after startup, use:

```
LBR.site=http://www.smart-package.com
```

You can specify multiple sites in the same way:

```
LBR.site=http://www.smart-package.com|site=www.fa2.de
```

This will do the same:

```
LBR.site=http://www.smart-package.com$crlf$www.fa2.de
```

Then all these sites will be opened in the MB, one after the other.

Hint: To wait until the sites have been opened, use the command `WBR.`

You can specify a window size and position and other startup parameters, for the starting Browser-window.Like this:

```
' We put the Sites into a variable to have a shorter line
length
' for this example.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$SIT=Site=www.fa2.de|site=www.it-
berater.org$crlf$www.facebook.com
LBR.Size=1024,980|Pos=0,0|svf=0|$$SIT
```

If `LBR.` is used in your scripts, in any way, then the EXE-Compiler will automatically include the needed Executable "SPR-Browser.exe" into the final EXE-Package. This way it can be used on any Client-PC in a consistent way.

Alternatively you can start the MB using any "`EXE.`"-command, or have the MB been started by other applications as the registered browser.
In this case you need to take yourself care that the MB is available on the target system, and if needed, is registered as the default browser.

In this case, you can use `LBR.` to connect to an already running MB, no matter how it was started. For more Information about the term *CUN*, see ! Smarty's InterKOM Technology 2116.

Example:

```
' $$BCU holds the CUN of the MB
' That we want to connect to
LBR.connect|$$BCU
```

Hint:

For yet unknown reasons, the Mini-Browser may in seldom cases crash immediately after starting. To make sure the MB is running properly, you can use this script:

```
' Button-Text must be changed for other then german language.
DBC.a|§§A00|SPR-Browser|&Programm schließen|||2|#32770
DBC.a|§§A01|SPR-Browser|Abbrechen|||2|#32770

$$URL=www.bitcoin.fa2.de
:retry
LBR.site=$$URL|Size=1024,980|Pos=2820,0|svf=0
' Here comes the trick, we first get the Window handle
' that is zero in case of a crash
CBR.gw.
HTV.$$WND
IVV.$$WND=0
  ' Now we shoot the MB
  KPN.SPR-Browser.exe
  PAU.200|ms
  GTO.retry
EIF.
```

For more details on the Mini-Browser (MB) 554 see there.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

LBR.[P1][|P2...|Px]

Parameter Explanation

P1...Px **- *(optional)*** If no Parameter is given, MB will launch with its default site. that is: "about:blank". Otherwise you can specify the following startup-parameters:

"connect=" - this Subcommand is special, because it can not be combined with any other Subcommand.
It does not start a new MB, but tries to connect the Script to an existing MB,
which CUN is in the given Variable.
If CUN is zero, it tries to get the CUN of the last localized Window.
Example:

```
LBR.connect=$$BCU
```

"size=" - this Subcommand can be used to give the new starting MB a defined Window-Size. Example:

```
LBR.Size=1024,800
```

You can just omit coordinates, then they will be replaced with default coordinates.

"Pos=" - this Subcommand can be used to give the new starting MB a defined Window-Position. While you can give screen coordinates as parameters, you can also write this:

```
' Center the MB
LBR.Pos=mid

' Place the MB at the given Coords
LBR.Pos=1200,900
```

You can just omit coordinates, then they will be replaced with default coordinates.

"Svf=" - this Subcommand means "Survive Father". It can be used to define a behavior for the new starting MB. If set to "0" then the MB will automatically close, when the robot / script ends.
If set to"1", then the MB will survive the "end of script". Example:

```
LBR.Svf=1
```

"Site=" - this Subcommand defines the sites that will be opened, immediately when the MB starts up.
You can give multiple "SIte=..." Subcommands. Also you can give a variable,

© 2023 Theo Gottwald * https://www.smart-package.com

in which the names of the sites to be opened are delimited by `$crlf$`.
Example:
`LBR.Site=www.fa2.de`
`LBR.Site=www.fa2.de|site=www.it-berater.org`

Example:

```
LBR.Size=1024,800|Pos=40,0|svf=0|Site=www.fa2.de
```

Example

```
'*********************************************
' LBR. - Sample
'*********************************************
$$SIT=Site=www.fa2.de|site=www.it-
berater.org$crlf$www.facebook.com
LBR.Size=1024,980|Pos=0,0|svf=0|$$SIT
WBR.
CBR.select|2
CBR.goto|www.theogott.de
WBR.
CBR.CloseT|face
PAU.20
ENR.

'*********************************************
' LBR.connect - Sample
'*********************************************
EXE.SPR-Browser.exe www.fa2.de
PAU.2
STW.t|SPR Mini
GIK.cfh.||$$BCU
LBR.connect=$$BCU
DOL.
  DBP.$mb_ert$
  DBP.#mb_busy#-#mb_prog#
OOP.(0)
MBX.Ready
ENR.
```

Remarks

There are multiple examples on using the Mini-Browser within the supplied Sample Scripts. A list with possible Errors can be found here:
Internet Explorer Errors and here: HTTP-Status Codes

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's Browser and Office Commands 554
- LBR. - Launch (Mini-)Browser 601
- CBR. - Control Browser 582
- WBR. - Wait Browser 630
- IBR. / NBR. - If Browser 595
- GAI. - Get-Acc-Information 591
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12 NAV - NAVigate to Object

3.44.12.1 ! NAV. - NAVigate in Dialog

NAV - Navigate in Dialog

Previous 601 Top 41 Next 612

MiniRobotLanguage (MRL)

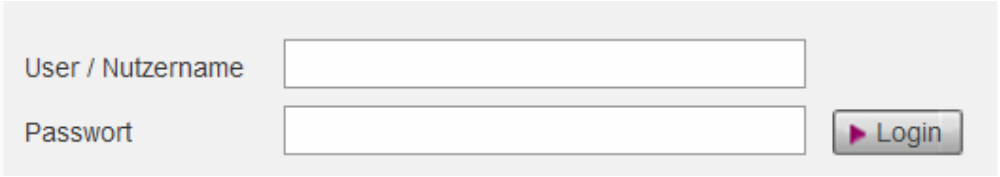
NAV. Command

Navigate in Dialog

Intention

This command locates screen-elements relative to another element. It is used most often in WEB-based GUI's (GUI = Graphical User Interface). But it can also be used in typical windows-based GUIs.

It is used to locate elements that can hardly be located in conventional, direct way. Here is an example. Take a look on the two textboxes inside the Internet Explorer (see picture below).



Why are these textboxes hard to locate?
Normally we locate an element by describing it to the robot. We say "Look for an element with the Name XY". As you see, these textboxes are empty. They may not have a valid name or a value parameter.
Then the only chance we have to grab such nameless boxes, would be, by using the role-parameter. Still there is a risk that if there are many elements with the same role, on a more or less dynamic WEB-Site (changes may happen), that we would locate not the right element.

Here is when the `NAV.` command takes is place. NAV. takes a special property to describe an element. NAV. looks "Where is it? How far is it from another element? "

Using `NAV.` we do like this:

1. Locate the window
2. locate an object we can safely locate, near the object we want to locate finally
3. use `NAV.` to navigate from the Object we have to the Object we want.

In this case we would like to locate the Textbox.We do like this:

1. Locate the window

```
STW.ct|IEFrame|Login - Windows Internet Explorer
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SCW.ct|Internet Explorer_Server|{&NOTEXT:}
```

2. Locate an object that we can describe
```
SAO.rn|42|User / Nutzername
```

3. NAV.igate to the Object that we really want
```
NAV.R|r
```

Here is the complete sample script:

```
STW.ct|IEFrame|Login - Windows Internet Explorer
SCW.ct|Internet Explorer_Server|{&NOTEXT:}
SAO.rn|42|User / Nutzername
NAV.R|r
MMV.
```

How does it work?
Let me tell you that NAV. is technically the most complex command we have built into the robot. Therefore you may need some time to understand its usage and all of its features.

This command localizes a specified Object in a Child-Window or Top-Window and sets the actual point into its center.
Please note that you need to localize a Element before, using SAO 625. Please note that NAV 606 can be quite slow when being used on complex applications and slow Computers.

For example, an Login Page in the Internet Explorer:
After you localized an unique Element (for Example the Text "Login") using SAO 625.

```
STW.c|ieframe
SAO.n|Login
```

You want to locate the empty Textbox at the Left side from that, to type the Login-Name in there. While it may be at the left side of the Text, the Center-Point of the Element is likely to be below of the one which is actually selected. Therefore we use a NAV 606.-Direction of "LD" (Left or Down).
If you want - for example in a Toolbar - to localize an equal element which is exactly right or down to a unique, already located element, you can use only one of these: "R" or "D".

The **8 possible Directions** are:
- (**A**)ll (All 4 directions)
- (**L**)eft
- (**R**)ight
- (**U**)p
- (**D**)own

and all **4 possible Combinations** of them.
Additionally there are:
- **LA** (Left-All) as a short replacement for: LUD
- **RA** (Right-All) as a short replacement for: RUD
- **UA** (Up-All) as a short replacement for: ULR
- **DA** (Down-All) as a short replacement for: DLR

© 2023 Theo Gottwald * https://www.smart-package.com

You can also use the "N" to jump to the next Element. Example:

```
STW.c|ieframe
SAO.n|Login
NAV.RN|DR|4
```

This one would jump to the 4'th next Element on the Down/Right Side. The 4'th next is the Element which Center-Point (!) is the 4'th next on that side.

Using NAV606., you can easily NAVIGATE to any Element in a Browser Window, by first using SAO625. to get an unique Element. Then secondly use NAV606. to get where you want.

To see which Element has which Number/Distance use A- with Parameter 8:

```
STW.c|ieframe
SAO.n|Login
NAV.RNA|DR|4|11
```

To get the nearest Element just use:

```
STW.c|ieframe
SAO.n|Login
NAV.q|edit
```

or to get the 3rd nearest Element which is a "edit-control" use

```
 STW.t|Basic
 SAO.n|Up
 MPO.
 NAV.Nq|3|edit
 MMV.
 END.
```

Most important with using NAV606. is:

- the Concept of the **8- Directions**,
- the Element **borderline**
- the distance between two element borders and
- the Element-**Center-Point** and
- the **distance** between two Elements Center-Points

Hint:

1. When you note that an Object that is there, is not found, you possibly need to use the "**u**" Options.

2. In regular Mode, NAV. will calculate the distance between elements from borderline to borderline.

3. When you reach the wrong Object, try using the "**g**" Prototoype and the Number "**&H8000**". This will exclude invisible targets.

```
SAO.grnvo|&H8000|42|×|adpubcomb&ANDTHEN:#close|nowin
```


© 2023 Theo Gottwald * https://www.smart-package.com

4. You can `NAV.`igate from any Object to another Object. This includes Objects that have been found using the `NAV.` command. You can use as many `NAV.` commands as you want.

```
NAV.uRKrn|LA|$$NAM|42|Date Qualified
NAV.uRKrn|d|$$NAM|42|:
```

Syntax:

NAV.[prototypes|][Data|...]

Parameter Explanation:

Prototype-Field:

```
A - (n) search only Elements of Group: X (0.1.2.3)
     0 = default (is internally change to 3)
     these values can be added:
     1 - only Objects,
     2 - only Simple Elements add
     4 - display Objects while searching.
     8 - paint Elements,
     16 - selective debug info (only found matches).

B - (var) Variable will contain the actual
     Acc-Object-Handle (LPAcc)
C - (var) Variable will contain the Object-Rectangle
D - (n) maximum Search-Depth
E - (-) close Acc-Object-Handle (LPAcc) after Operation
F - (var) Variable will contain the window handle of the
    localized object
G - (-) Start Search at the Top-Object.
H - (n) Handle (Input Window-Handle) instead of last
    Acc-Object-Handle (LPAcc). Counterpart to "L"
K - (var) Variable, set to 0/1 to indicate if the Object
    could be localized.
L - Input-Values Acc-Object-Handle (LPAcc), counterpart to "H"
M - Variable with Object Middlepoint.
N - (v) Number of the Object to search for.
    1st fitting, 2nd, 3rd ... etc.)
P - (-) switch from Border- (default) to Middlepoint-Distance
R - (s) DiRection in which to search (up, down etc.). Give
     direction in big letters, U,D,L,R, A and Combinations of
     two letters.
T - (-) Start Search from the Top-Window.
U - (n) minimale Suchtiefe
    If H and L are not explicitly specified, the actual window
    and option H is choosen.
V - (n) Role-group (1 - Control, 2 - Background)
a - (s) defaultAction (string)
b - (n) helptopic-id
c - (n) child-id
d - (s) Description (s)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
e - (n) state AND (these states must be set, there can be
    others set as well)
f - (s) helpfile (string)
g - (n) state (these states must NOT be set, others may be
set)
h - (n) window-handle from Acc-Object-Handle (LPAcc)
i - (n) "iac"-tree-depth
j - (s) help-string
k - (s) keybrd shortcut

m - (s) helptopic
n - (s) name   (s)
o - (l) On not found goto Label.
p - (n) PID
q - (s) window-class
r - (n) role (v)
s - (n) state (v) muss exact stimmen
t - (v) TID
u - (-) invisible auch enumerieren (ist unabhängig vom state)
v - (s) value (s)
w - (n.fp) Waiting-Time to wait for appearance of the Element
x - (v) Number in Tree-Level
y - (-) must be linked
z - (-) search only depth of last found SAO.-Object
```

Example:

```
' Simple NAV-Example
 STW.t|Basic
 SAO.n|Up
 MPO.
 NAV.R|D
 MMV.
 NAV.R|D
 MMV.
 NAV.R|RD
 MMV.
 NAV.R|U
 MMV.
MBX.!
END.

' This Example will find a text in the Browser
' and move the Mouse to the control below.
'
DBM.2
JNF.Move-Console.txt

: §§TXT=Google-Suchen

STW.t|Google&AND:Explorer
MPO.

SAO.n|bilder
MMV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU.1
' Navigate to the first Element in the "down-right" direction
' Display (A/1) Elements inverse and with distance-numbers.
' Use only Group 1 Elements ( V/1 - recommended).
NAV.RNAV|DR|1|11|1
MMV.

MBX.!
END.
```

Remarks:

Please always use E if you do not need the Object for further `NAV./SAO`-Commands. Otherwise small amounts of Memory may not be free'd by the system.

When you note that an Object that is there, is not found, you possibly need to use the "**u**" Options.

You can specify multiple Roles, they must be separated by ",".
Any of these will be valid.

If you specify a "#" before the Role-Parameter, it will mean "Role-Group" and specifies any of the following:
0: unknown
1: User-bedienbare Controls
2: Panels, Client und Background
4: Borders etc.
8: Window-Controls (Titlebar, Minimize etc.)
16: Scrollbars
32: Spezial, Cursor, Sound etc.

Limitations:

This command can be slow on complex applications.
Wrong usage may lead to steadily decreasing memory.
You can currently not navigate more far then 20 Elements away from an element.

Using this commands on WEB-Sites that do just load/rebuild internal structures may lead to all sorts of crashes.

See also:

- SAO. - Search Accessible Object 625
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591

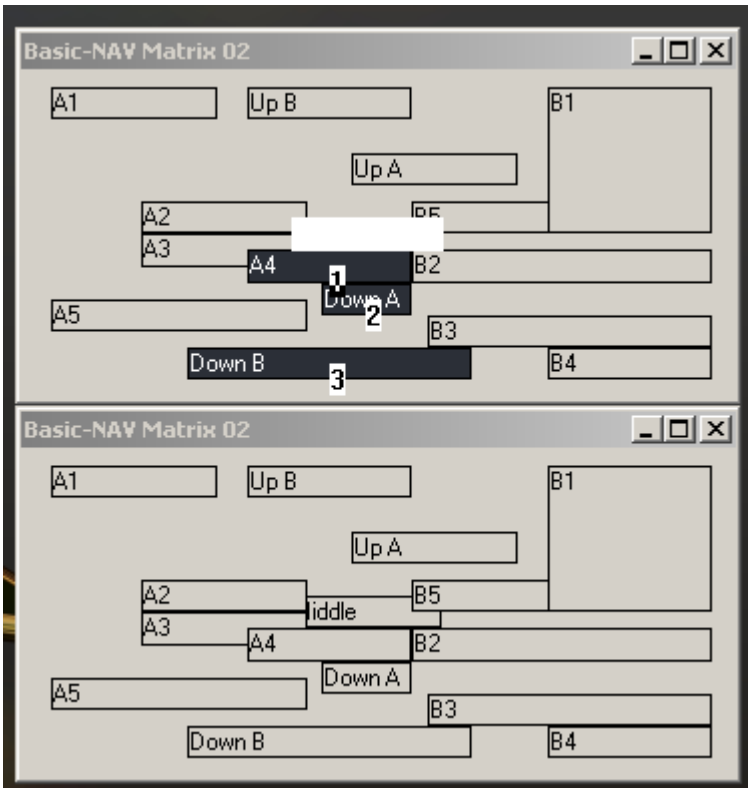
© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.2 NAV - "Down"

NAV - "Down"

The following two Picture shows a NAV.R|D (Down) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

You can see, that only those Elements are treated as "Down", where the Center-Point is between the left and right border of the Reference-Object.



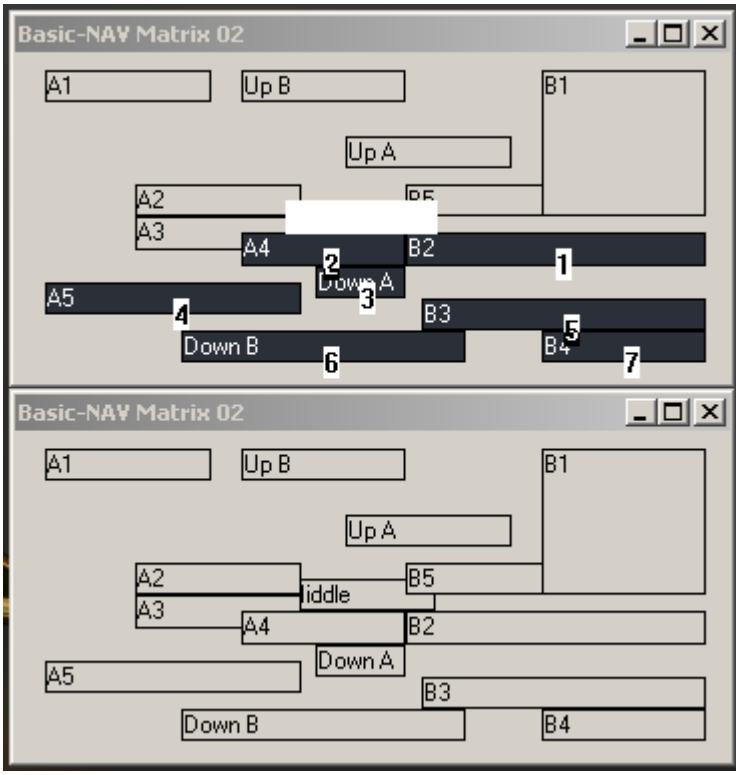
© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.3 NAV - "DownAll"

NAV - DownAll

The following two Picture shows a NAV.R|DA (DownAll) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

Elements which overlap the reference Object are sorted by an independent scheme, which may lead to effects like A3, which is been treated as "Left", not "down" because its center-point is higher then lower border from Reference-Object.



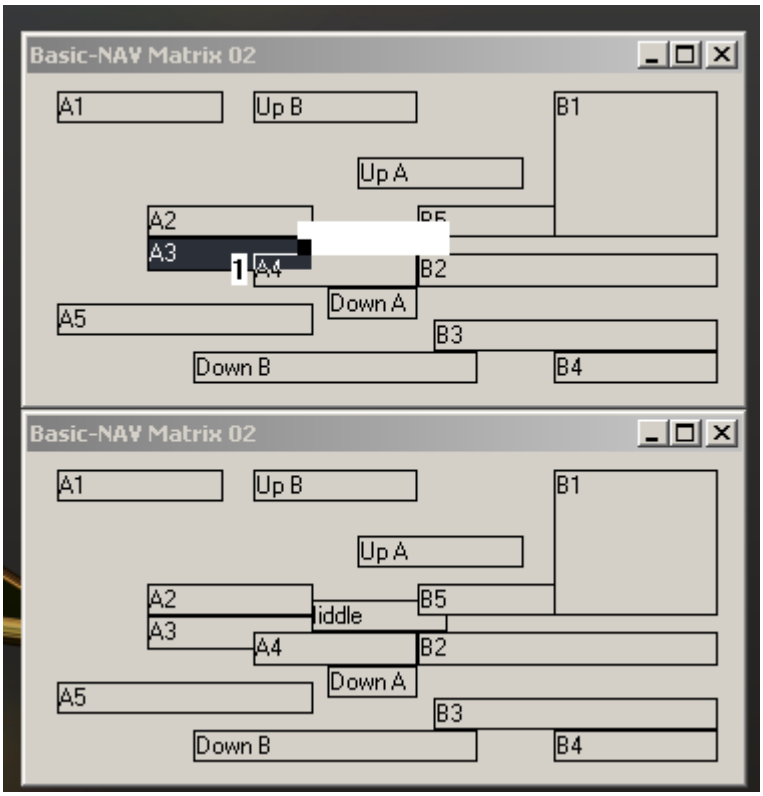
© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.4 NAV - "Left"

NAV - "Left"

The following two Picture shows a NAV.R|L (Left) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

In this example, A3 is the only "pure Left" object.
All other objects are "Left-Up" or "Left-Down".



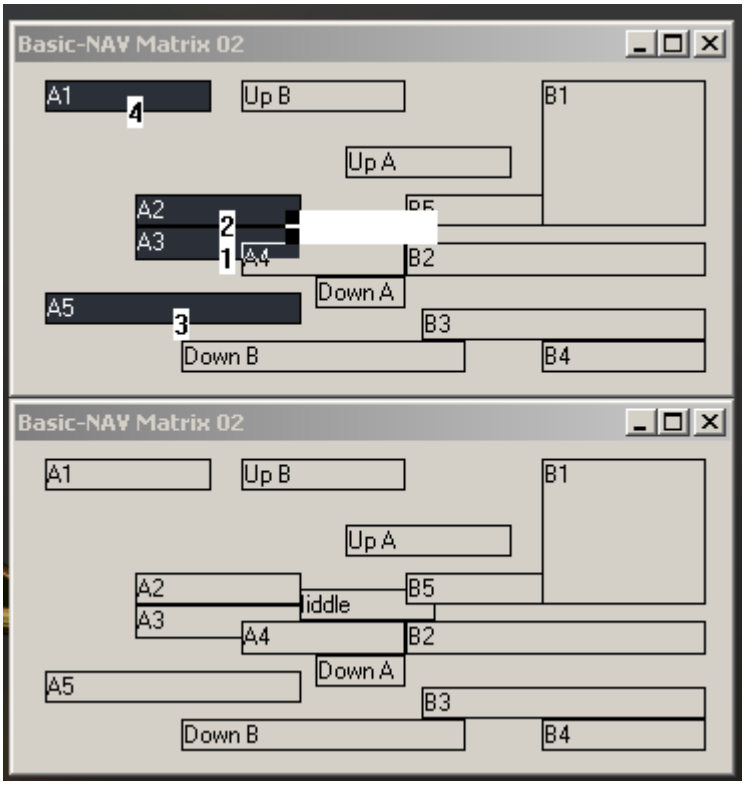
© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.5 NAV - "LeftAll"

NAV - "LeftAll"

The following two Picture shows a NAV.R|LA (LeftAll) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

Please note that the field "Down B" is not taken as "Left from" because its Center-Point is right from the Left Border of the Reference-Object (white Rectangle).



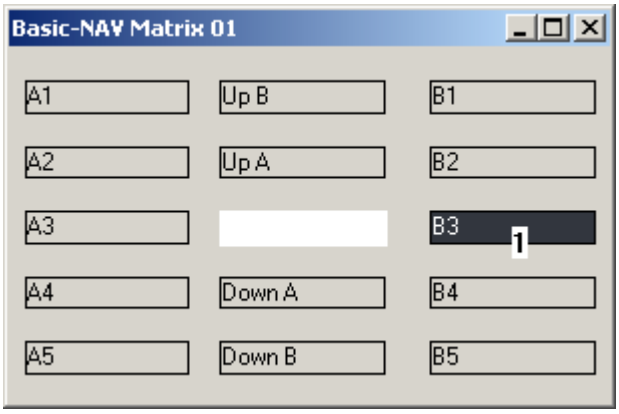
© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.6 NAV - "Right"

NAV - "Right"

The following two Picture shows a NAV.R|UA (UpAll) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

In this special case, there is only one Element recognized as "right".
All other elements are "Right Up" or "Right Down".

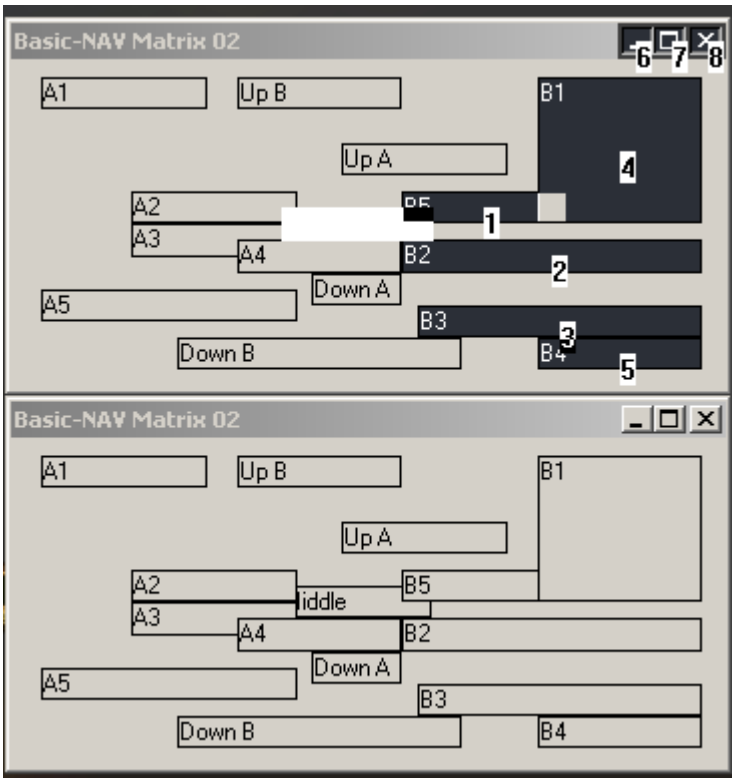


© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.7 NAV - "RightAll"

NAV - "RightAll" Previous 616 Top 41 Next 618

The following two Picture shows a NAV.R|RA (RightAll) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.



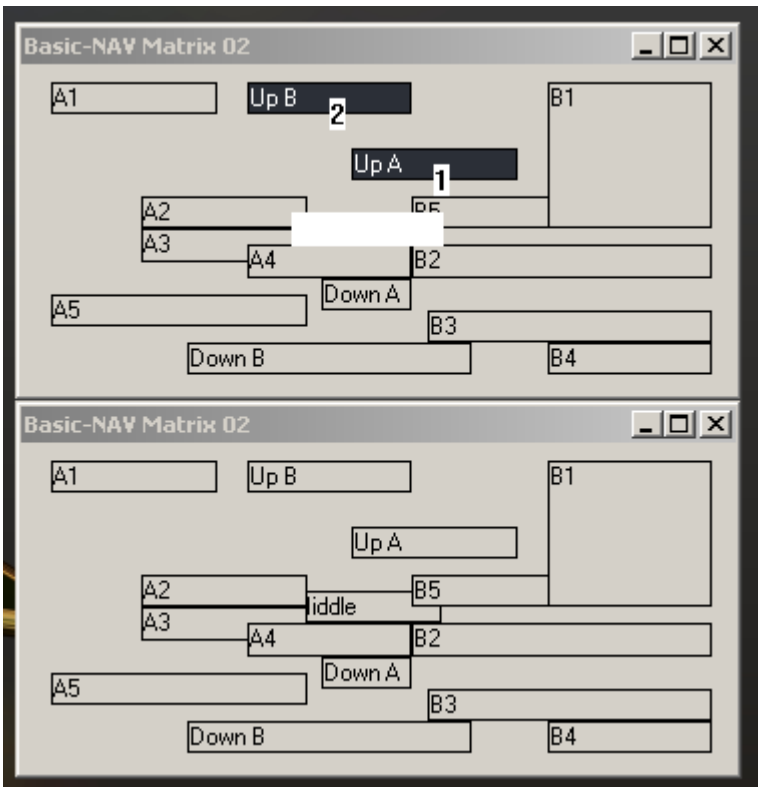
© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.8 NAV - "Up"

NAV - "Up"

The following two Picture shows a NAV.R|U (Up) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

Only the two shown Elements are treated as "Up", all other Elements are "Up Left" or "Up Right".

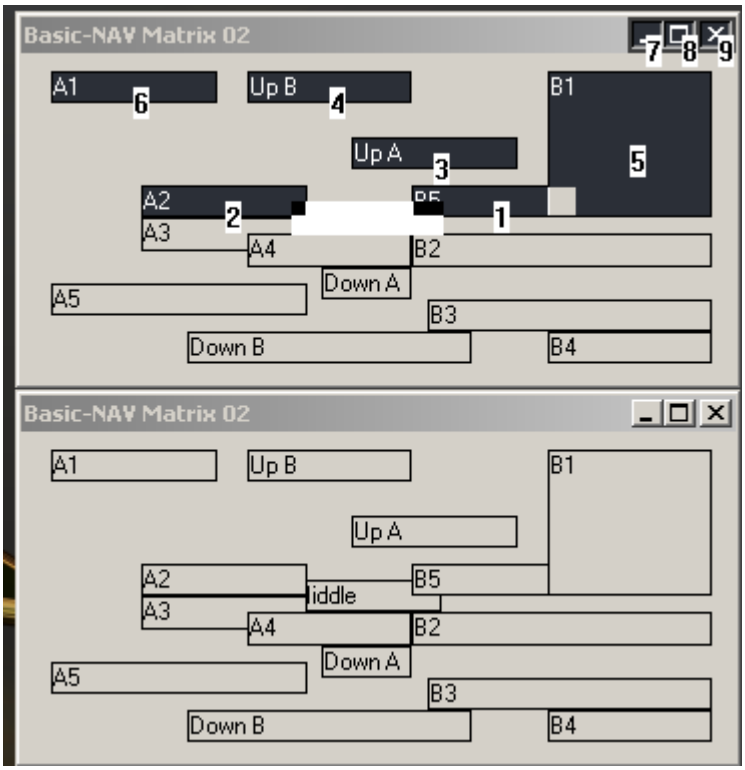


© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.9 NAV - "UpAll"

NAV - "UpAll"

The following two Picture shows a NAV.R|UA (UpAll) from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.



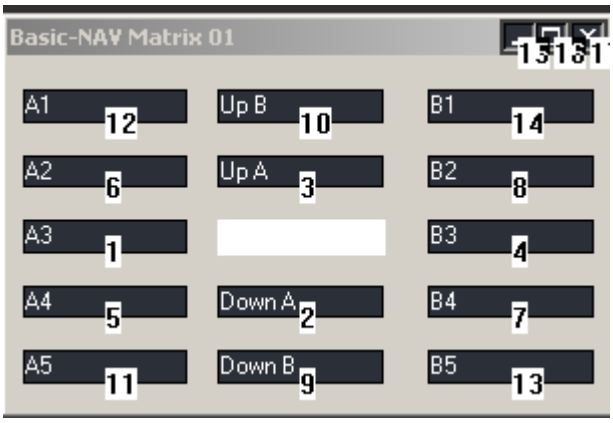
*© 2023 Theo Gottwald * https://www.smart-package.com*

3.44.12.1 NAV - Example 1

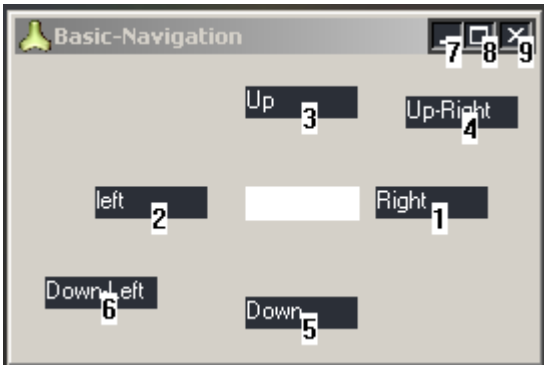
NAV - Example 1

The following two Picture shows a NAV.R|A from the Element in the Center (white Rectangle), the numbers demonstrate the Order of the Elements if the Prototype "N" would be used.

Picture 1:



Picture 2:

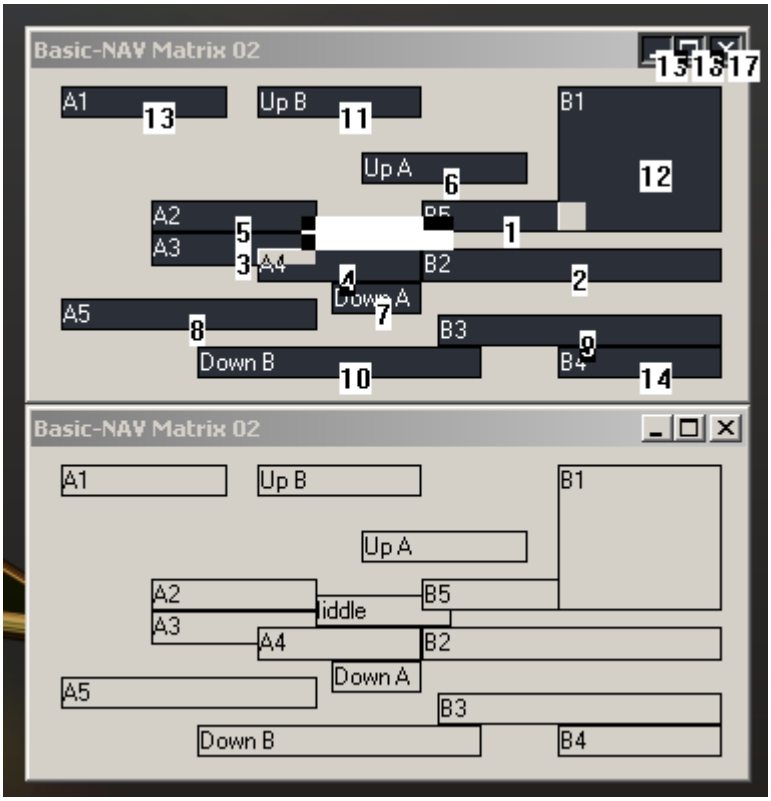


© 2023 Theo Gottwald * https://www.smart-package.com

3.44.12.1 NAV - Example 2

NAV - Example 2

The following Picture shows a more complex situation using a NAV.R|A from the Object in the center (white rectangle).



© 2023 Theo Gottwald * https://www.smart-package.com

3.44.13 Roles and States from the Acc.-System (E8)

Roles and States from the Acc.-System (E8) Previous 621 Top 41 Next 625

Role-Parameter for AGR. etc.

You can specify multiple Roles, they must be separated by ",".
Any of these will be valid.

If you specify a "#" before the Role-Parameter, it will mean "Role-Group" and specifies any of the following:

0: unknown

1: User-bedienbare Controls

2: Panels, Client und Background

4: Borders etc.

8: Window-Controls (Titlebar, Minimize etc.)

16: Scrollbars

32: Spezial, Cursor, Sound etc.

State-Values:

```
STATE_SYSTEM_NORMAL          = 0
STATE_SYSTEM_UNAVAILABLE     = &H1
STATE_SYSTEM_SELECTED        = &H2
STATE_SYSTEM_FOCUSED         = &H4
STATE_SYSTEM_PRESSED         = &H8
STATE_SYSTEM_CHECKED         = &H10
STATE_SYSTEM_MIXED           = &H20
STATE_SYSTEM_INDETERMINATE   = STATE_SYSTEM_MIXED
STATE_SYSTEM_READONLY        = &H40
STATE_SYSTEM_HOTTRACKED      = &H80
STATE_SYSTEM_DEFAULT         = &H100
STATE_SYSTEM_EXPANDED        = &H200
STATE_SYSTEM_COLLAPSED       = &H400
STATE_SYSTEM_BUSY            = &H800
STATE_SYSTEM_FLOATING        = &H1000
STATE_SYSTEM_MARQUEED        = &H2000
STATE_SYSTEM_ANIMATED        = &H4000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'************************************
STATE_SYSTEM_INVISIBLE        = &H8000
STATE_SYSTEM_OFFSCREEN        = &H10000
'************************************
STATE_SYSTEM_SIZEABLE         = &H20000
STATE_SYSTEM_MOVEABLE         = &H40000
STATE_SYSTEM_SELFVOICING      = &H80000
STATE_SYSTEM_FOCUSABLE        = &H100000
STATE_SYSTEM_SELECTABLE       = &H200000
STATE_SYSTEM_LINKED           = &H400000
STATE_SYSTEM_TRAVERSED        = &H800000
STATE_SYSTEM_MULTISELECTABLE  = &H1000000
STATE_SYSTEM_EXTSELECTABLE    = &H2000000
STATE_SYSTEM_ALERT_LOW        = &H4000000
STATE_SYSTEM_ALERT_MEDIUM     = &H8000000
STATE_SYSTEM_ALERT_HIGH       = &H10000000
STATE_SYSTEM_PROTECTED        = &H20000000
'STATE_SYSTEM_VALID           = &H7fffffff
STATE_SYSTEM_HASPOPUP         = &H40000000
```

Role-Values:

```
ROLE_SYSTEM_TITLEBAR          = &H1
ROLE_SYSTEM_MENUBAR           = &H2
ROLE_SYSTEM_SCROLLBAR         = &H3
ROLE_SYSTEM_GRIP              = &H4
ROLE_SYSTEM_SOUND             = &H5
ROLE_SYSTEM_CURSOR            = &H6
ROLE_SYSTEM_CARET             = &H7
ROLE_SYSTEM_ALERT             = &H8
ROLE_SYSTEM_WINDOW            = &H9
ROLE_SYSTEM_CLIENT            = &Ha
ROLE_SYSTEM_MENUPOPUP         = &Hb
ROLE_SYSTEM_MENUITEM          = &Hc
ROLE_SYSTEM_TOOLTIP           = &Hd
ROLE_SYSTEM_APPLICATION       = &He
ROLE_SYSTEM_DOCUMENT          = &Hf
ROLE_SYSTEM_PANE              = &H10
ROLE_SYSTEM_CHART             = &H11
ROLE_SYSTEM_DIALOG            = &H12
ROLE_SYSTEM_BORDER            = &H13
ROLE_SYSTEM_GROUPING          = &H14
ROLE_SYSTEM_SEPARATOR         = &H15
ROLE_SYSTEM_TOOLBAR           = &H16
ROLE_SYSTEM_STATUSBAR         = &H17
ROLE_SYSTEM_TABLE             = &H18
ROLE_SYSTEM_COLUMNHEADER      = &H19
ROLE_SYSTEM_ROWHEADER         = &H1a
ROLE_SYSTEM_COLUMN            = &H1b
ROLE_SYSTEM_ROW               = &H1c
ROLE_SYSTEM_CELL              = &H1d
ROLE_SYSTEM_LINK              = &H1e
ROLE_SYSTEM_HELPBALLOON       = &H1f
ROLE_SYSTEM_CHARACTER         = &H20
ROLE_SYSTEM_LIST              = &H21
ROLE_SYSTEM_LISTITEM          = &H22
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ROLE_SYSTEM_OUTLINE          = &H23
ROLE_SYSTEM_OUTLINEITEM      = &H24
ROLE_SYSTEM_PAGETAB          = &H25
ROLE_SYSTEM_PROPERTYPAGE     = &H26
ROLE_SYSTEM_INDICATOR        = &H27
ROLE_SYSTEM_GRAPHIC          = &H28
ROLE_SYSTEM_STATICTEXT       = &H29
ROLE_SYSTEM_TEXT             = &H2a
ROLE_SYSTEM_PUSHBUTTON       = &H2b
ROLE_SYSTEM_CHECKBUTTON      = &H2c
ROLE_SYSTEM_RADIOBUTTON      = &H2d
ROLE_SYSTEM_COMBOBOX         = &H2e
ROLE_SYSTEM_DROPLIST         = &H2f
ROLE_SYSTEM_PROGRESSBAR      = &H30
ROLE_SYSTEM_DIAL             = &H31
ROLE_SYSTEM_HOTKEYFIELD      = &H32
ROLE_SYSTEM_SLIDER           = &H33
ROLE_SYSTEM_SPINBUTTON       = &H34
ROLE_SYSTEM_DIAGRAM          = &H35
ROLE_SYSTEM_ANIMATION        = &H36
ROLE_SYSTEM_EQUATION         = &H37
ROLE_SYSTEM_BUTTONDROPDOWN   = &H38
ROLE_SYSTEM_BUTTONMENU       = &H39
ROLE_SYSTEM_BUTTONDROPDOWNGRID     = &H3a
ROLE_SYSTEM_WHITESPACE       = &H3b
ROLE_SYSTEM_PAGETABLIST      = &H3c
ROLE_SYSTEM_CLOCK            = &H3d
ROLE_SYSTEM_SPLITBUTTON      = &H3e
ROLE_SYSTEM_IPADDRESS        = &H3f
ROLE_SYSTEM_OUTLINEBUTTON    = &H40
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.44.14 SAO. - Search Accessible Object

MiniRobotLanguage (MRL)

SAO. Command

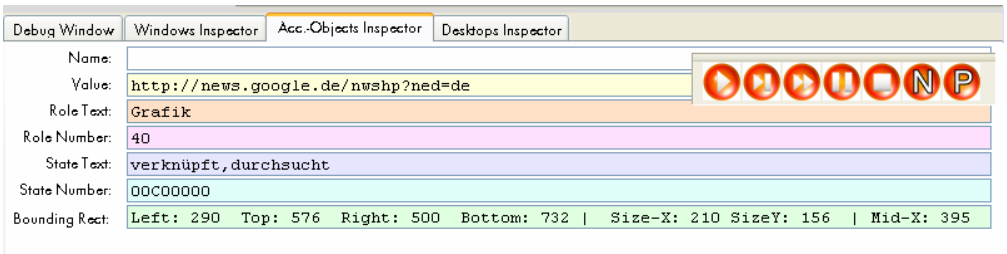
Search Accesible Object

Intention

This command localizes a specified Acc.-Object in a Child-Window or Top-Window and sets the actual point into the object center.

You can see Acc-Objects in the Acc.-Inspector.Let us see an example from a WEB-Site in the Internet Explorer:

In the picture below you will see the resulting Display in the Acc.-Inspector.



The most important parameters for SAO. are:
"**name**", "**role**" and "**value**". In most cases you can ignore the other parameters.

Now we'll take the above picture from the Acc.-Analyzer, and make the code for it.

© 2023 Theo Gottwald * https://www.smart-package.com

Using SAO. we do best locate a Top- and a child-window. Like this:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SCW.ct|Shell DocObject View|{&NOTEXT:}
SAO.rv|40|google&ANDTHEN:nwshp
MMV.
ENR.
```

In some cases this may not be possible, or will not work. Then we can use SAO. just on the Top-Window:

```
STW.ct|IEFrame|Theo Gottwald&ANDTHEN: - Windows Internet
Explorer
SAO.rv|40|google&ANDTHEN:nwshp
MMV.
ENR.
```

Which may be a bit slower in execution times, depending on the overall complexity of the WEB-/Office program.

Hints:
It is important, that you localize a window before using SAO.

Unlike AGR., SAO. uses the 1.4. The Prototype Parameter System 73. Before using SAO. please read that chapter in the help.

SAO. supports 3.2 Standard-Search Pattern 341 and they can help you to better define the things you look for. See example above "google&ANDTHEN:nwshp".

Please note that SAO can be quite slow when being used on complex applications and slow Computers.

Use the Acc.-Inspector in the Editor to get the necessary parameter data.

Hint:

1. When you note that an Object that is there, is not found, you possibly need to use the "**u**" Options.

2. When you reach the wrong Object, try using the "**g**" Prototoype and the Number "**&H8000**". This will exclude invisible targets.

```
SAO.grnvo|&H8000|42|×|adpubcomb&ANDTHEN:#close|nowin
```

3. If you can not safely locate a Object directly, use the NAV. Command to navigate to the Object.

Syntax

SAO.P1[|P2...|Px]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** Prototypes

```
A - (n) search only Elements of Group: X (0.1.2.3)
     0 = default (is internally change to 3)
     these values can be added:
     1 - only Objects,
     2 - only Simple Elements add
     4 - display Objects while searching.
     8 - paint Elements,
     16 - selective debug info (only found matches).

B - (var) Variable will contain the actual
     Acc-Object-Handle (LPAcc)
C - (var) Variable will contain the Object-Rectangle
D - (n) maximum Search-Depth
E - (-) close Acc-Object-Handle (LPAcc) after Operation
F - (var) Variable will contain the window handle of the
    localized object
G - (-) Start Search at the Top-Object.
H - (n) Handle (Input Window-Handle) instead of last
    Acc-Object-Handle (LPAcc). Counterpart to "L"
K - (var) Variable, set to 0/1 to indicate if the Object
    could be localized.
L - Input-Values Acc-Object-Handle (LPAcc), counterpart to "H"
M - Variable with Object Middlepoint.
N - (v) Number of the Object to search for.
    1st fitting, 2nd, 3rd ... etc.)
T - (-) Start Search from the Top-Window.
U - (n) minimale Suchtiefe
    If H and L are not explicitly specified, the actual window
    and option H is choosen.
V - (n) Role-group (1 - Control, 2 - Background)
a - (s) defaultAction (string)
b - (n) helptopic-id
c - (n) child-id
d - (s) Description (s)
e - (n) state AND (these states must be set, there can be
    others set as well)
f - (s) helpfile (string)
g - (n) state (these states must NOT be set, others may be
set)
h - (n) window-handle from Acc-Object-Handle (LPAcc)
i - (n) "iac"-tree-depth
j - (s) help-string
k - (s) keybrd shortcut

m - (s) helptopic
n - (s) name   (s)
o - (l) On not found goto Label.
p - (n) PID
q - (s) window-class
r - (n) role (v)
s - (n) states (v) must match exactly
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
t - (v) TID
u - (-) include invisible objects (is state independent)
v - (s) value (s)
w - (n.fp) Waiting-Time to wait for appearance of the Element
x - (v) Number in Tree-Level
y - (-) must be linked
```

P2-Px

```
- Data(s)
```

Example

```
' This Example will find an item in the XP-Quickstart
' and move the Mouse there
VAR.$$NAM=Internet
STW.c|shell_traywnd
SAO.nKAN|$$NAM|$$ERG|3|2
PRT.Item was found (0-NO, 1 -YES)$crlf$ Answer =$$ERG
MMV.
DMP.
```

Remarks

Please use the prototype E if you do not need the Object for further `SAO./NAV.-` Commands. Otherwise small amounts of Memory may not be free'd by the system.

You can not use this command to wait for a WEB-Site to be opened.
Using the command on a WEB-Site that is going to be opened, may result in a unrecoverable crash of the robot program.

You can specify multiple Roles, they must be separated by ",".
Any of these will be valid.

If you specify a "#" before the Role-Parameter, it will mean "Role-Group" and specifies any of the following:
0: unknown
1: User-bedienbare Controls
2: Panels, Client und Background
4: Borders etc.
8: Window-Controls (Titlebar, Minimize etc.)
16: Scrollbars
32: Spezial, Cursor, Sound etc.

Limitations:

This command can be slow on complex applications.
Wrong usage may lead to steadily decreasing memory.
Using this commands on WEB-Sites that do just load/rebuild internal structures may lead to all sorts of crashes.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- !. NAV - NAVigate in Dialog 606
- MOS. - Mouseless Object Select 2509
- GAI. - Get-Acc-Information 591

© 2023 Theo Gottwald * https://www.smart-package.com

3.44.15 WBR. - Wait Browser

MiniRobotLanguage (MRL)

WBR. Command

Wait for Browser Event/ Site loaded

Intention

Using `WBR.` you can wait until a Site had been completely loaded in the Mini-Browser (=MB). `WBR.` can also be used to wait for Browser Events.

For more details on the Mini-Browser (MB) 554 see there.

`WBR.` is a multi-purpose command that includes multiple possible subcommands.
`WBR.` supports a set of typical "Wait" Subcommands that do not have further parameters,
as well of a second set that is used to check Error states.

Usage is simple. To just wait until a WEB-Site is loaded, write:

```
WBR.
```

Below you will see more examples on how to check for Errors and loading Progress.

Syntax

WBR.[P1][|[!]P2]

Parameter Explanation

[!] – If you prefix the time-value **P2** with a "**!**" then the Logic is negated and the given
condition is true, if its false.

P1 – `Subcommand`, the following Subcommands are currently supported:

- `WBR.`
  `WBR. without any options is the same as WBR.all`

- `"All"` - wait until MB is ready with all Jobs
  `WBR.all`
  `WBR.all`
  `WBR.all[|[!]max time]`

*© 2023 Theo Gottwald * https://www.smart-package.com*

- "Page" - Wait until the actual Page has been loaded
  ```
  WBR.Page[|[!]max time]
  WBR.Page
  ```

- "Job" - Wait until the actual Job is done
  ```
  WBR.Job[|[!]max time]
  WBR.Job
  ```

The following Subcommands support Comparisons **!,=,>,<**
If you prefix these Subcommands with a "**q**" they will "confirm" any Error like if you had been using the command:

```
CBR.errclear
```

If you omit the "q" you will need to manually call

```
CBR.errclear
```

to clear internal Error Variables, and make them ready for the next Event.

- "Progress","Pro." - Compare actual loading Progress (0-100%)
  ```
  WBR.Progress=progress value[|[!]max time]
  WBR.Progress>0|120
  WBR.Progress<19|60
  WBR.Progress<19|!60
  ```

- "Error","Err." - Compare actual loading Error State
  ```
  WBR.Err.=0[|[!]max time]
  WBR.Error>0|120
  WBR.Error=0|!120
  WBR.Error!200|66
  ```

- "State","Sta." - Compare actual loading Error State as textual pattern 341
  For this textual compare the sign is been ignored.
  ```
  WBR.Sta.=HTTP&AND:Error|55
  WBR.State=HTTPS&OR:Error|!251
  WBR.qState=HTTPS&OR:Error|!251
  ```

- "Description","Des." - Compare loading State as textual pattern 341
  For this textual compare the sign is been ignored.
  ```
  WBR.Des.=HTTP&AND:Error|55
  WBR.description=HTTPS&OR:Error|!251
  WBR.qDescription=HTTPS&OR:Error|!203
  ```

Depending on the chosen Subcommand the further Parameters **P3** to **P4** differ and may be
needed or not.
If you prefix the Subcommand or the max-time value with a "**!**" this will negate the logic of the command. If you prefix both, the logic will be unchanged.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - **`[optional] Time-Value`** If given, this is the maximum time to wait for the condition to be true. If missing, the command will wait "endless". If the condition does not become true, then the command will set the timeout-flag and just continue with the script. You can check the timeout-flag to immediately after this command, to see if an timeout has occurred.

cond - WBR. supports the following conditions:

! - NOT

= - Is Equal to

> - Larger then

< - Smaller then

Hint: Combined Operators like "**>=**" are not supported. In this case, just use ! and "<" and switch the operators. Because **A>=B** is the same as **NOT B<A**.

Example

```
' This example uses WBR. to wait until a Site is been loaded.
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa2.de
WBR.
CBR.gw.
FOR.$$NUM|1|40
  AGR.40|||$$NUM
  GAI.v|$$LNK
  DBP.Der hinterlegte Link:$crlf$$$LNK
  MMV.
NEX.
MBX.Browser exit
CBR.exit
ENR.


' Example on how to track Site-loading Errors
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa33.de
WBR.
DBP. Descr.: $mb_ert$
DBP. Err-Code: #mb_err#
DBP. Err.-Name: $mb_err$
MBX.Browser exit
CBR.exit
ENR.

Output:
[10:15:00] [12]  Descr.: The server or proxy was not found.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
[10:15:00] [13]  Err-Code: 2148270085
[10:15:00] [14]  Err.-Name: INET_E_RESOURCE_NOT_FOUND

' Example, if we wait for an specific Error to happen
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa33.de
' This will wait forever until the site is "Not found".
'
WBR.qDes.=was not found
' Because WBR.Des. will confirm the Error,
' you will get a state 0 (no error) here:
DBP. Descr.: $mb_ert$
'DBP. Err-Code: #mb_err#
'DBP. Err.-Name: $mb_err$
'MBX.Browser exit
CBR.exit
ENR.

' Example, if we wait for an specific Error to happen
' In this case the Error-Number is 2148270085
'
LBR.Size=1024,980|Pos=0,0|svf=0|site=www.fa33.de
WBR.Err.=2148270085
DBP. Err-Code: #mb_err#
CBR.exit
ENR.
```

Remarks

There are multiple examples on using the Mini-Browser within the supplied Sample Scripts. A list with possible Errors can be found here:
Internet Explorer Errors and here: HTTP-Status Codes

Limitations:

-

See also:

- ! Smarty's Browser and Office Commands 554
- LBR. - Launch (Mini-)Browser 601
- CBR. - Control Browser 582
- WBR. - Wait Browser 630
- IBR. / NBR. - If Browser 595
- GAI. - Get-Acc-Information 591

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.45 Control specific operations

3.45.1 BTC. - Button-Check

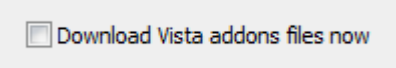
MiniRobotLanguage (MRL)

BTC. Command

Button Check

Intention

Check Radio-Buttons or Checkmarks, which have been previously located by SCW. or other commands like SAO.



This could be a 1.9.1.4 Checkmark Control 184.

```
' To check a checkmark, use
BTC.1

' To uncheck a checkmark, use
BTC.0
```

These commands use the MLE.-command (direct-engine). If the just do not work, you can make the command to use the MLI. (compatible engine) by adding a 3 to that number.

```
' To check a checkmark, (compatible engine) use
BTC.4

' To uncheck a checkmark, (compatible engine) use
BTC.3
```

You can make the command to use the MLC.-command (compatible engine) by adding a 6 to that number.

```
' To check a checkmark, (compatible engine) with Mouse use
BTC.6

' To uncheck a checkmark, (compatible engine) with Mouse use
BTC.7
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

BTC.P1

Parameter Explanation

P1 - Numeric parameter, represents the target-status of the option button.

```
0,3 or 6 - Uncheck
1,4 or 7 - Check
2,5 or 8 - Set to "indeterminate" (3-State checkmarks Only)
```

Example

```
' This Example will check a checkmark and uncheck it.
' Please note that you need to have such a window.
DBM.2
STW.t|sample
SCW.ct|button|check1
' check the checkmark
BTC.1
' uncheck the checkmark
BTC.0

DMP.
END.
```

Remarks

For Treeview-Checkmarks use the **TVI.** or **TVF.**-command with the "**d**" or "**b**"-Flag.

Limitations:

Works only in standard Controls. Does not work in "Ownerdrawn" or Java Checkmarks. May work or not in in Browser and Office. Use Mouse-Clicks (MLC. MOS.etc.) in use-cases.
"Intermediate state" may not in for all programs work as expected. Use normal Mouse-Clicks (MLE., MLI. etc.) in such cases.

See also:

- GBC. - Get-Button-Check 640
- TVF. - Tree View Find 665
- TVI. - Tree View Invisible 668

© 2023 Theo Gottwald * https://www.smart-package.com

- AGR. - Accessibility Get Rect 564

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.2 CDD. - Combo Drop Down

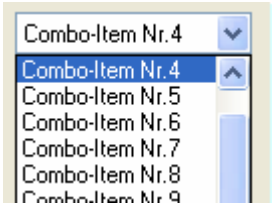
MiniRobotLanguage (MRL)

CDD. Command

Combobox open/close

Intention

A Dropdown-List Control that includes Here is a Picture of a Combobox/Dropdown with item Nr.4 selected.



A Dropdown-control, often also contains a builtin

This command is an add-on to the SCI.-command to make it a compatible engine command (see: 1.C Direct Engine and Compatible Engine 231) command.

In very few cases, SCI. 657 will visually set the Combobox to the right value, but the program will not have accepted it. In these cases, just add a CDD. after the SCI. This will make the Combobox accept the value.

Syntax

```
CDD.[P1]
```

Parameter Explanation

P1 - (optional) 0 or 1, if omitted, the Combobox will be opened and closed after 150 milliseconds.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
 ' Selects the 1-based Index-Item Nr.34
  SCI.34|N
' open and close the Combobox
  CDD.

 ' Selects the Entry which contains "Hallo"
  SCI.Hallo -> Textangabe
CDD.0
PAU.1
CDD.1
```

Remarks

-

Limitations:

-

See also:

- 1.9. Locating Child-Windows 162
- 3.2 Standard-Search Pattern 341
- **SCI. - Set-Combo-Item** 657
- GCI. - Get-Combo-Item 642
- SKP. Send-Key-Press 2289
- SDT. - SenD-window-Text 2278
- SKT. - Send-Key-Text 2293

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.3 GBC. - Get-Button-Check

MiniRobotLanguage (MRL)

GBC. Command

Get Button Check

Intention

Get the status of system's check-mark or radio-button controls.

Syntax

GBC.[P1]

Parameter Explanation

P1 - (optional) Variable to store the checkmark-state to. If omitted, TOS is used.

Example

```
'************************************************
' GBC- Sample #1
'************************************************
' This Example will read the state of a checkmark
' that has been located as a window.
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
SCW.nct|1|Button|Option122
GBC.
MBX.The state of the checkmark is $$000
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'************************************************
' GBC- Sample #2
'************************************************
' This Example will read the state of a checkmark
' that has been located using SAO. (accessibility).
'
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
SAO.rn|45|122
GBC.$$RES
MBX.The state of the checkmark is $$RES
ENR.
```

Remarks

-

Limitations:

-

See also:

- TVF. - Tree View Find 665
- TVI. - Tree View Invisible 668
- **BTC. - Button-Check** 635

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.4 GCI. - Get-Combo-Item

MiniRobotLanguage (MRL)

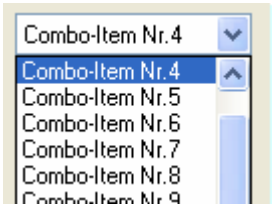
GCI. Command

Get Combo Item

Intention

The `GCI.`-Command can be used to return the Combo-Box-Items.

Here is a Picture of a Combobox with item Nr.4 selected.



It is important, that you first locate a Combobox 162. To do this, You can use SCW. 3738 as well as SAO 625..

In our Test-Application this would look like:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nc|3|ComboBox
```

Once this is done, the computer knows which Combobox you want to analyze. So we can start using `GCI.` The simplest way of operation is, if you just write:

```
GCI.
```

In this case you will find the actual selected Combo-Item (see below) on the TOS:

© 2023 Theo Gottwald * https://www.smart-package.com

The `TOS` contains the 1-based Index number of the actual selected item. Below the TOS is the text of the selected Combo-Item.

The `GCI.`-command has multiple Modes of operation. To switch between them, the command knows a number of prototypes, see below.

Lets start with the c - prototype. **C** like **C**ount. Using this prototype you can simply find out how many items the Combobox has.

There are two modes of operation using this prototype:

```
'This will just place the number of Combo-Items on the TOS.
GCI.c
```

```
' This will store the result in the given variable instead on
TOS.
GCI.c|$$NUM
```

As you can see the only difference is, that the result is once on the Stack, and once in a given variable. This will be the same with most of the following operation modes.

Using the **s** - like **s**elected prototype, you can find out which Combo-Item is currently selected. As the returned index is 1 - based, a return value of zero means that currently there is no Item selected. The modes of operation for the "**s**"-prototype are:

```
GCI.
GCI.s
GCI.s|$$TXT
GCI.s||$$NUM
GCI.s|$$TXT|$$NUM
```

 © 2023 Theo Gottwald * https://www.smart-package.com

The only difference is, which result is been placed on the TOS and which is placed inside variables. $$TXT[366] hereby is the Text of the selected Combo-Item, while `$$NUM` is the 1-based Index of it.

Using the **t** - like **t**ext prototype, you can get the Text of an Combo-Item from an 1-based Item-Index. The details from where the numeric index is taken, and where the result is been place, are straightforward.

```
' Take number from TOS, place result on TOS
GCI.t

' Take number from $$NUM, place result on TOS
GCI.t|$$NUM

' Take number from $$NUM, place result in $$TXT
GCI.t|$$NUM|$$TXT
```

Using the **n** - like **n**umeric prototype, you can find out the numeric, 1-based Index of an Combo-Item. The string you specify can contain 3.2 Standard-Search Pattern[341].

```
' Will get the search string from TOS, and place result on TOS
GCI.n

' Search string from $$TXT, result on TOS
GCI.n|$$TXT

' Search string from $$TXT, choose result number $$NUM
' place result on TOS
GCI.n|$$TXT|$$NUM ' resultat number in $$NUM

' Search string from $$TXT, choose result number $$NUM
' place result in variable $$RES
GCI.n|$$TXT|$$NUM|$$RES
```

Using the **f** - like **f**ind prototype, you can find a Combo-Item using 3.2 Standard-Search Pattern[341].

```
' Will get the search string from TOS, and place result on TOS
GCI.f

' Search string from $$TXT, result on TOS
GCN.f|$$TXT

' Search string from $$TXT, choose result number $$NUM
' place result on TOS
GCN.f|$$TXT|$$NUM

' Search string from $$TXT, choose result number $$NUM
' place result in variable $$RES
GCN.f|$$TXT|$$NUM|$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

Using the **a** - like **a**ll prototype, you can

```
' Concatenates all Combo-Items using $crlf$ as delimiter.
' Result is been placed on TOS.
GCI.a

' Concatenates all Combo-Items using $$DEM as delimiter.
' Result is been placed on TOS.
GCI.a|$$DEM

' Concatenates all Combo-Items using $$DEM as delimiter.
' Result is been placed in $$RES
GCI.a|$$SDEM|$$RES
```

If the delimiter expression is a null (zero-length) string, no separators are inserted between the string sections.

If the delimiter expression is the following 3-byte value of "," a leading and trailing double-quote is added to each string section.

Syntax

GCI.[P1][|P2][|P3]

Parameter Explanation

The details of the parameter depend on the used prototype.

P1 - (optional) prototype, selects the kind of operation, if omitted "s" is used. The following prototypes are supported:

```
c - count Combo-items
s - selected Combo-Item (get selected)
t - text of Combo Item
n - numeric, get 1-based Index of Combo-Item
f - find, find a Combo-Item based on a Text using Patterns
a - all, fill a variable with all Combo-Items, possibly with a
    delimiter of your choice.
```

P2 - (optional) VAR to store results, STRING or number.
P3 - (optional) VAR to store results.
P4 - (optional) VAR to store results.

Example

```
'************************************************
' GCI- Sample #1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Get actual selected Item
'************************************************
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
' Select Combo-Box
SCW.i|1027
' Will return the actual selected Item on TOS
GCI.
' Lets see the TOS
STS.DUMP
MBX.Ready
ENR.


'************************************************
' GCI- Sample #2
' Get all Combo-Items in a List
'************************************************
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
' Locate the Combobox by ID
SCW.i|1027
' Get all Combo-Item delimited by newlines
GCI.a
STS.DUMP
MBX.Ready $$TXT
ENR.

'************************************************
' GCI- Sample #3
' Get all Combo-Items in a delimited List
'************************************************
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
' Locate the Combobox by ID
SCW.i|1027
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Get all Combo-Item delimited by newlines
GCI.a|"."|$$TXT
' Complete the result with leading and trailing "
VAR.$$FIN="$$TXT"
MBX.$$FIN
ENR.


'************************************************
' GCI- Sample #4
' Get Combo-Item No.12
'************************************************
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
' Locate the Combobox by ID
SCW.i|1027
' Get Combo-Item No.12
GCI.t|12|$$TXT
MBX.$$TXT
ENR.


'************************************************
' GCI- Sample #5
' Ermittelt 1 - basierten Index des aktuell
' selektierten Elementes
'************************************************
'
' This first line will just test if the TestApp is already
' running and must not be started.
STW.Xwtg|0|TestApp|over
' We start the TestApp
EXE.?exepath\Sample Apps\TestApp_01.exe
' We can locate it directly with the PID
STW.p|#pid#
:over
' Locate the Combobox by ID
SCW.i|1027
' Get Combo-Item No.12
GCI.s||$$TXT
MBX.$$TXT
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

There may be non-Standard Comboboxes out there, where this command may not work as expected. If that happens, its not a bug, its just that the developer of that Application made his Combobox too far away from Common-Control-Standards. Please note that this command does generally not support Comboboxes in Browser- or Java-Applications.

Also this command may currently not work with all x64 applications.

See also:

- 1.9. Locating Child-Windows 162
- 3.2 Standard-Search Pattern 341
- **SCI. - Set-Combo-Item** 657
- CDD. - Combo Drop Down 638
- SKP. Send-Key-Press 2289
- SDT. - SenD-window-Text 2278
- RST. - Replace Selected Text 2264
- SKT. - Send-Key-Text 2293

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.5 LBE. - ListBox Entry

MiniRobotLanguage (MRL)

LBE. Command

Listbox Entry

Intention

This command is used to select an item in a Listbox Control. It is also used to check checkmarks in Listbox controls "Listbox Checkmark".

There are several modes of operation. In the simplest mode, `LBE.` gets just one parameter. And that is the entry to locate. You can specify 3.2 Standard-Search Pattern 341.
And you can use variables for any of the parameters.

```
LBE.Entry-Name
```

If there are multiple equal named Entries, you can additionally specify a number.

```
LBE.Entry-Name|number or formula in ()
' Looks like
LBE.Entry Nr.|5
```

If there is a checkmark that needs to be checked, you can specify an third parameter:

```
' without the number parameter (default is 1)
LBE.Entry Nr.||1
' or if you need the number parameter P2 also:
LBE.Entry Nr.|2|1
```

Syntax

LBE.P1|P2[|P3]

Parameter Explanation

P1 – Entry name. Can use 3.2 Standard-Search Pattern 341

P2 – ***(optional)*** number of fitting entry (1- first, 2 - second etc.).
if omitted, default value is 1

*© 2023 Theo Gottwald * https://www.smart-package.com*

P3 **-** ***(optional)*** value for internal "Buttoncheck" (Listbox-Checkmarks).

These are:
0 - uncheck checkmark (direct engine 231)
1 - check checkmark (direct engine)
2 - uncheck checkmark (compatible engine 231)
3 - check checkmark (compatible engine)

Example

```
'*****************************************
' Demo for LBE.
'*****************************************
LBE.Entry Nr.
LBE.Entry Nr.||1
LBE.Entry Nr.||(1-$$VAL)
LBE.Entry Nr.|($$NUM-1)|1

END.
```

Remarks

Works with x32 and also with x64 Listbox-Controls.

Limitations:

This command is generally for use with **Listbox Controls**. A good hint is, when the Acc.-Inspector shows a **role** of: **34,36 or 44**. In these cases you can try using the `LBE.` command. There may be controls out there that do not react on this command. This is not a bug, but a result of the "Inner living" of that specific control.

See also:

- SAO. - Search Accessible Object 625
- !. NAV - NAVigate in Dialog 606
- BTC. - Button-Check 635
- MLI - Mouse Left Click qulckly (Invinsible) 2454

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.6 LVI. - Listview Command

MiniRobotLanguage (MRL)

LVI. Command

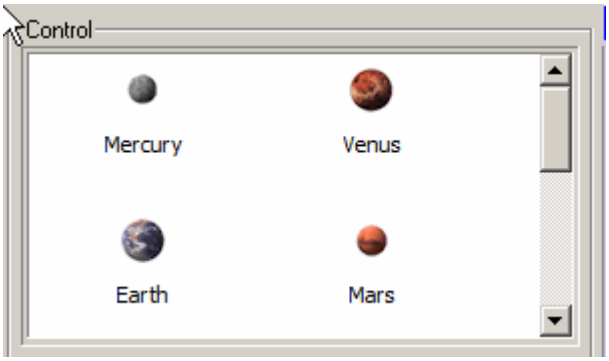
ListView32 Command

Intention

These command will select a Listview item. A Listview is a special type of control and may look like this (Listview in Report-Mode):

Column #1	Column #2	Column #3
Row #001 Item #01	Row #001 Item #02	Row #001 Item #03
Row #002 Item #01	Row #002 Item #02	Row #002 Item #03
Row #003 Item #01	Row #003 Item #02	Row #003 Item #03
Row #004 Item #01	Row #004 Item #02	Row #004 Item #03
Row #005 Item #01	Row #005 Item #02	Row #005 Item #03
Row #006 Item #01	Row #006 Item #02	Row #006 Item #03
Row #007 Item #01	Row #007 Item #02	Row #007 Item #03
Row #008 Item #01	Row #008 Item #02	Row #008 Item #03

Listviews that are not in Report-Mode may show Icons for the elements they contain:



The desktop is also a Listview Control, and it shows Icons.
Therefore the LVI.-Command can also be applied on the Windows Desktop.

Using LVI. you can locate Items in a Listview by their numeric position, or using a textual description. If you use a textual description, 3.2 Standard-Search Pattern 341 can be used.

Depending on the SubCommand, `LVI.` has multiple purposes.

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

LVI.P1[.OPT][...|Px]

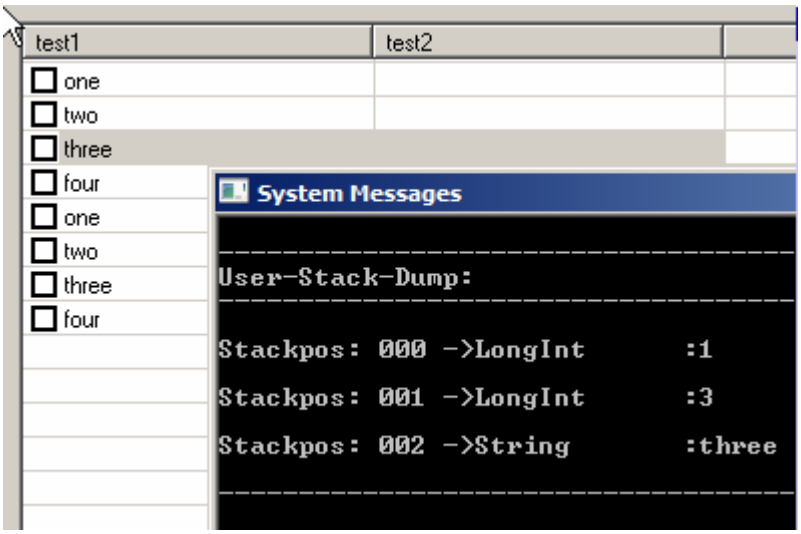
The Syntax depends on the used Subcommands:

Parameter Explanation

P1 – Subcommand, is one of the following Subcommands:

Find a Listview Item by name, locate center of text area (and navigate)
"ftbn","find text by name"
This Subcommand finds a Listview Item using a Substring of the Text that it contains.
3.2 Standard-Search Pattern can be used.
The item will be located, you will find Information about the item on the TOS.

```
' Using this example, the TOS looksl ike this:
LVI.ftbn.unc|thre
```




```
LVI.ftbn|P2|P3|P4|P5

P2 - Text textual Pattern you want to search for
P3 - (optional) Number of occurrence
P4 - (optional) Navigate X by Steps
P5 - (optional) Navigate Y by Steps
```

using P4 and P5, you can find a specified Item and then navigate from there in X- or Y-direction to another item.

Examples:

```
LVI.ftbn|Earth
LVI.ftbn|Ear&OR:Mar
LVI.ftbn.mlm|Earth
LVI.ftbn|Earth||1
```


© 2023 Theo Gottwald * https://www.smart-package.com

Find a Listview Item by name, locate Icon area (and navigate)

"fibn","find icon by name"

This Subcommand finds a Listview Item using a Substring of the Text that it contains.

It will locate the Icon Area to the left instead of the Text-Area.

3.2 Standard-Search Pattern 341 can be used.

The item will be located, you will find Information about the item on the TOS.

```
LVI.fibn|P2|P3|P4|P5

 P2 - Text textual Pattern you want to search for
 P3 - (optional) Number of occurrence
 P4 - (optional) Navigate X by Steps
 P5 - (optional) Navigate Y by Steps
```

using P4 and P5, you can find a specified Item and then navigate from there in X- or Y-direction to another item.

Examples:

```
LVI.fibn|Earth
LVI.fibn|Ear&OR:Mar
LVI.fibn.mlm|Earth
LVI.fibn.chk|Earth
LVI.fibn|Earth||1
```

Find a Listview Item by number

"fbn","find by number"

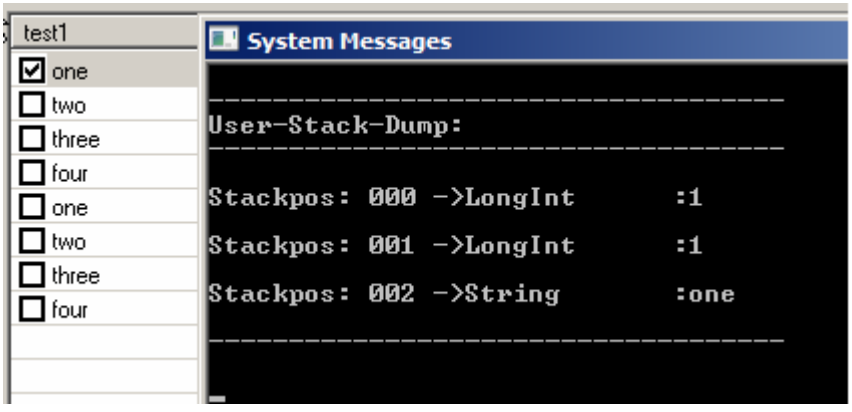
This Subcommand finds a Listview Item using numerical coordinates.

You can specify an X- (Row) and an Y- (Column).

Both numbers are 1-based.

The item will be located, you will find Information about the item on the TOS.

```
000  Column-Number or zero
001  Row-Number, 1- based
002 Name if item that was found
```



Examples:

```
' Locate 4'th item in a single column LV
LVI.fbn|4
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
      ' Locate 4'th item in the second row of a multi-column
LV
      LVI.fbn|4|2
      ' Locate the first item and check the checkmark
      LVI.fbn.chk|1
```

Get number of Items, number of columns, Listview-Style number and Unicode Information

```
   "gin","get info"
```

This `Subcommand` will retrieve needed informations about the Listview. You can omit
parameters that you do not need. In that case these values are placed on the TOS.

```
 LVI.get info|P2|P3|$$STL|$$UNI
```

The first Variable **P2** receives the number of Lines thats in the Listview (Line-Count).
Variable **P3** receives the number of Columns if the Listview has more then one.
Variable **P4** receives the Listview Style value. This is for advanced use only.
Variable **P5** receives 0 or 1 if the Listview is a Unicode Control (1) or an ANSI Control (0).
There is a Unicode conversion built in. Some Listviews are marked as Unicode and are
not used as such, in this case you may get "??????" as textual result.
You can then use the "Force Unicode" Subcommand to set this Flag to 0 and make the
Control readable.

```
   LVI.get info|$$TAY|$$TAX|$$STL|$$UNI
```

Set the text of the selected item to another text / change text

```
"sit","set item text"
```

You can change the text of an item using this command.

```
   LVI.sit|TY|TX|Text
```

Unselect all Items in the Listview

```
   "usa","unselect all items"
```

This will unselect all selected items in the Listview.
Depending on the Listview, some Listviews may not support this command.

Select all Items in the Listview

```
   "sai","select all items"
```

This will select all items in the Listview.
Depending on the Listview, some Listviews may not support this command.

© 2023 Theo Gottwald * https://www.smart-package.com

Cancel Edit Mode

```
"nedi","cancel edit label"
```

Set the Unicode Flag of the Listview

```
"funi","force unicode"
```

You can set the unicode flag of the Listview to 0 or to 1.
This will influence the way that the text is internally interpreted, at least under x32.
Sometimes Controls are marked as Unicode but are not used as that.
In that case you may receive textual results like ?????? ??? and nothing is found.
If so, set the unicode flag to zero using this command before using other LVI. commands.

Fake Unicode processing

```
"fauni","fake unicode"
```

This will influence the way that the text is internally interpreted, at least under x32.
It will not physically change the Unicode flag inside the Control.
It will only change the way, all further LVI.-Commands process the Text.
This currently has only an effect under x32 Applications.

`.OPT` - You can add one or more of these Options to any of the Subcommands.
Then the selected Options will be used on the Target-Listview-Item.
Possible Options are:

```
.lbo - Move Actual Point to Left Border for all later commands
.rbo - Move Actual Point to Right Border for all later
commands
.foc - focus item
.sel - select item
.chk - check a checkamrk on that item
.unc - uncheck a checkamrk on that item
.vse - Enshure visible + select Item
.usa - unselect all items
.ndi - cancel Item-Text editing
.edi - Start Item-Text Editing
.upd - Update/Redraw Listview Control
.uns - unselect this item
.del - delete Item from Listview
.wii - Wait for Input Idle
```

The following Options are not available as **`.OPT`** under x64.
Just use the normal SPR Commands (`MLE.`,`MLM.` etc. instead).
Also note that these clicks always hit the item-Center, even if you have specified `.lbo` or `.rbo`.

```
.mle - Direct Engine Click
.mli - do a compatible Click
.mdc - do a Double Click
.mlm - do a MLM-Click
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
.clr - do not do any selection, this will clear all other .OPT
Flags
```

Example

```
'*********************************************
' LVI.-Sample
'*********************************************
' If no such item as specified is found,
' the timeout-flag is neen set.
' You can check that using the ITO-Query.
'
STW.ct|#32770|item
SCW.nc|1|SysListView32
LVI.ftbn.unc|threx
ITO.
  MBX.No such item was found
EIF.
MMV.
STS.DUMP
MBX.!
ENR.

'*********************************************
```

Remarks

On the stack, you can find the name of the selected element.
This useful, if it was selected using a Pattern.

The command has internally a different path for x32 and x64 applications.
Some features may not do exactly the same under x64 and x32.
Also some features may not work under both types of OS.

Limitations:

Some features may not work under x64, in th3e same way as they do under x32.

See also:

- TVI. - Tree View Invisible 668
- GBC. - Get-Button-Check 640
- BTC. - Button-Check 635

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.7 SCI. - Set-Combo-Item

MiniRobotLanguage (MRL)

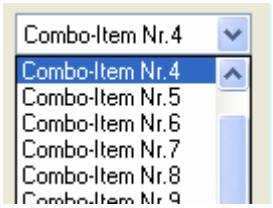
SCI. Command

Select Combo Item

Intention

With this command you directly select a specified combo box entry.
You can specify the entry to select, by its name, 1 based index or position.

Here is a Picture of a Combobox with item Nr.4 selected.



It is important, that you first locate a Combobox 162. To do this, You can use SCW. 3738 as well as SAO 625..

In our Test-Application this would look like:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nc|3|ComboBox
```

Once this is done, the computer knows in which Combobox you want to select an item.
When choosing an item "by text", you can use 3.2 Standard-Search Pattern 341 or just give a unique part of the text you want to select.

```
SCI.Item-Nr.7
```

As alternative, you can select an item by its 1-based item-index. If you count the items from top to bottom, the most upper item has Nr.1, the next is Nr.2 ... until you reach the bottom item. For this you would write:

```
SCI.7|N
```

Which would select the same item as before. In case you want to make sure that **P1** is a text item (part of), you can use the **T**(ext) instead of the **N**(umeric).

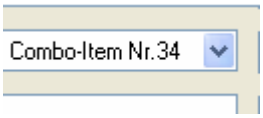
```
SCI.Item-Nr.7|T
```


© 2023 Theo Gottwald * https://www.smart-package.com

And in this case you also have the chance to add a item-number. For example in our Testapplication the script::

```
STW.ct|PBWindowClass:0|TestApp
SCW.nc|1|ComboBox
SCI.4|T|3
```

will make this item to be selected, as it is the third item that contains a "4".



Hint:
In case the `SCI.` will change the Combo-Item only visually, but not really, you can use a script like this below.

```
STW.ct|FORM_DIFF-TOOL_MAINDLG_CLASS:0|Robo-Diff
SCW.nct|1|FORM_DIFF-TOOL_REGOPTION_CLASS:3|Registry-Options
SCW.nic|1|1001|ComboBox
' Instead of SCI. we click on the Combobox to open it
MLE.
' Now we get the Combolbox
STW.Ac|combolbox
' Here we do the selection
AFT.34!MLI|HKEY_CURRENT_USER\Test
```

Here is another example:

Instead of
```
'SCI.MSK_ws.png
```

you write something like:

```
' Klick on the Combobox to open it
STW.ct|#32770|Berechnung
SCW.nict|1|83|ComboBox
MLI.

' Treat it like a Menu and click on the Item
STW.Ac|Combolbox
' Possible Commands are: SCI. | CDD.
AFT.34!MLI.|MSK_ws.png
```

Syntax

SCI.[P1][|P2][|P3]

Parameter Explanation

P1 - A value like a number a text or variable

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - (optional) N tells the command that P1 is a numeric value for index access
T tells the command that P1 is a text value. The default when P2 is missing.

P3 - (optional) Numeric value, fitting item number, in case there are multiple equal named items.

Example

```
' Selects the 1-based Index-Item Nr.34
SCI.34|N

' Selects the Entry which contains "Hallo"
SCI.Hallo -> Textangabe

' Selects the 3.rd Entry which contains "Hallo"
SCI.Hallo|T|3
CDD.
```

Remarks

SCI. is a direct engine command 231. In few cases where it might select the right item, but the application does not react on it, use CDD. additionally to open and close the Dropdown.This will often solve the problem.

Limitations:

See also:

- GCI. - Get-Combo-Item 642
- CDD. - Combo Drop Down 638
- SKP. Send-Key-Press 2289
- SDT. - SenD-window-Text 2278
- RST. - Replace Selected Text 2264
- SKT. - Send-Key-Text 2293

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.8 SMH. - Set Menu Hook

MiniRobotLanguage (MRL)

SMH. Command

Set Menu Hook

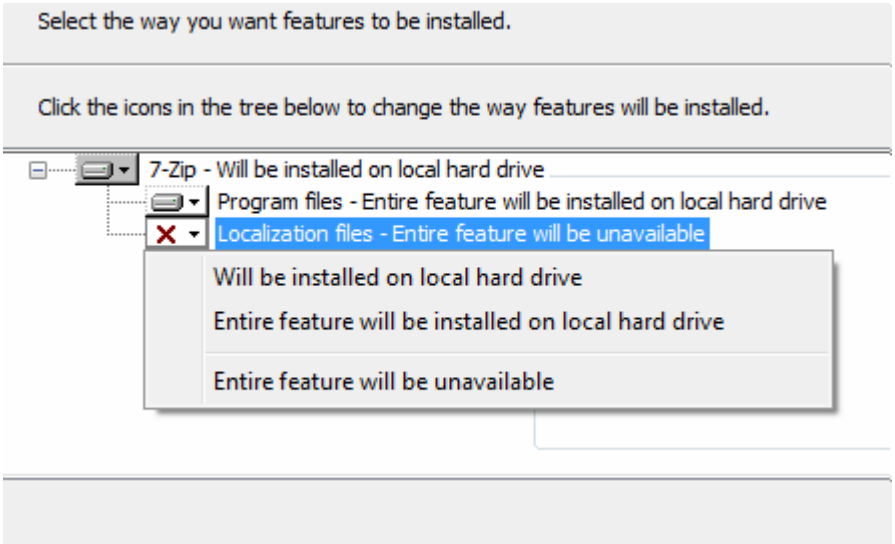
Intention

These command will click a menu item in a popup-menu.

You can specify the menu item by text (also use 3.2 Standard-Search Pattern 341 = or you can xpecify the menu item by number.

Additionally you can choose which action is been made with the located menu-item.

You can also use SMH. with customizing-Treeviews in MSI-Setup's. 215 It will click in the upcoming Menü and select the item of your choice. Here is an example:



Here is an example that will paste something into a textbox in our Testapplication.

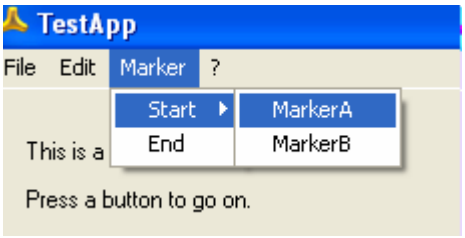
```
' Select the TestApp
STW.ct|PBWindowClass:0|TestApp

' Select the Textbox
SCW.nc|3|edit

' Tell Hook that soon a Menu will pop up, he should click
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Einfügen (engl. "Paste")
SMH.Einfügen

' Now we move the Mouse to the Textbox and click the right
Mousebutton
MRC.

' This Pause is needed, because if the script ends, the SMH.
' will also end, before the Menu is been clicked!
PAU.2

ENR.
```

In case you have a deep-menu, you may combine `SMH.` with `SAO.`

```
STW.ct|PBWindowClass:0|TestApp
SAO.rn|12|Marker
SMH.Start
MOS.

STW.ActE|#32768|{&NOTEXT:}
SAO.rn|12|MarkerA
MOS.
PAU.1
```



Syntax

SMH.P1|P2

Parameter Explanation

```
P1 - action you want


    c = use mouse click by using COMPATIBLE-ENGINE
    d = use mouse click (MLE) by using DIRECTENGINE
    e = use mouseless Mouseaction (the default) by using
ENGINE8
    p = no further action, set only the point on the actual
entry

    a = search menu item by by number (count separators)
    m = search menu item by number (do not count seperators)
    t = search menu item by text
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    i = mouse jump to point
    v = mouse move to point
```

P2 - Argument to P1. It can be text or number.

Example

```
STW.t|sample ap
SMH.i|Submen&ANDTHEN:8
SAO.rn|12|edit
MLI.
SMH.Submen&ANDTHEN:12
PAU.3


SMH.i|Neu
MRC.
STW.c|#32768
SMH.Verknüp

SMH.c|Eigensch&AND:{&NOT:nvi}

'*****************************************
' SMH._Sample with deep Menu
' use TestApp
'*****************************************
STW.ct|PBWindowClass:0|TestApp
SAO.rn|12|Marker
SMH.Start
MOS.

STW.ActE|#32768|{&NOTEXT:}
SAO.rn|12|MarkerA
MOS.
PAU.1
MBX.Ready
ENR.


'*****************************************
' SMH._Sample
'*****************************************
' Select the TestApp
STW.ct|PBWindowClass:0|TestApp

' Select the Textbox
SCW.nc|3|edit

' Tell Hook that soon a Menu will pop up, he should click
' Einfügen (engl. "Paste")
SMH.Einfügen

' Now we move the Mouse to the Textbox and click the right
Mousebutton
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MRC.

' This Pause is needed, because if the script ends, the SMH.
' will also end, before the Menu is been clicked!
PAU.2
ENR.


#########################################################

Subroutines to Click inside the Windows Explorer Menu (Win 10)
Note that the :MenuClick must be changed to the local
Situation
to bring up the Explorer Menu.

Call liket this:
GSB.SetSizeBig

'============================================================
:SetSizeBig
GSB.DoEMenu|Ansicht|&EXACT:Große Symbole
RET.
'============================================================
:SetSizeMed
GSB.DoEMenu|Ansicht|Mittelgr
RET.
'============================================================
:Sort_Typ
GSB.DoEMenu|Sortieren|Typ
RET.
'============================================================
:DoEMenu
§§TMA=§§_01
§§TMB=§§_02
GSB.MenuClick

STW.AcE|#32768
GSB.DoCLickM|§§TMA

STW.Ac|#32768
HTV.§§MEN
GSB.DoCLickM|§§TMB
MAW.hp|§§MEN
RET.
'============================================================
:DoCLickM
§§TMC=§§_01
SAO.rn|12|§§TMC
MPO.
MLI.
RET.
============================================================
:MenuClick
STW.h|$$WIN
' THIS COORDINATE must be somewhere where a Explorer Menu
comes up!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' This will find a place where the System-Menu will come.
AGR.33!WII.MRC.|||1|3|60
RET.
'=========================================================
```

Remarks

To find place where to click in the Windows-Explorer, you can use:

```
AGR.33!WII.|||1|3|60
MRC.
```

Limitations:

If you run into problems with Menus that are more then 2 Levels deep, use STW. with the "I"-Flag (Exclude-List) and **SAO.** and **MLI.** to select further Menu entries.

See also:

- SAO. - Search Accessible Object 625
- MRC. / MRI. - Mouse-Right-Click 2472

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.9 TVF. - Tree View Find

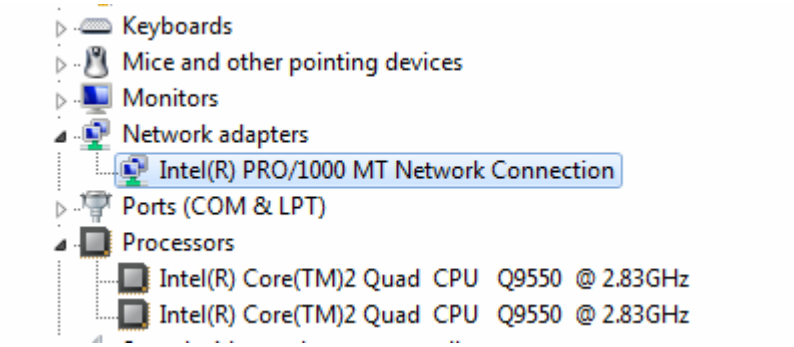
MiniRobotLanguage (MRL)

TVF. Command

Tree View Found

Intention

These command will select a treeview item. A treeview is a special type of control and may look like this:



Please see here for more details and examples about TVF.and treeviews. 202

Here are some more hints:

```
' Locate the first fitting element
TVF.Intel(R) Core||1

' Locate the second fitting element
TVF.Intel(R) Core||2
```

For P1 a 3.2 Standard-Search Pattern 341 can be used.

```
TVF. uses the direct engine for most flags.

' Here is an example that will uncheck an treeview-checkmark
STW.ct|#32770|Main
SCW.nct|1|SysTreeView32|SysTreeView32_1
TVF.Item8|d
```

There are two commands to work with treeviews.

```
TVI. -  uses a path to the item and uses wildcard-patterns
TVF. - uses 3.2 Standard-Search Pattern
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

TVF.P1[|P2][|P3]

Parameter Explanation

Searchargument

P1 - item name can use 3.2 Standard-Search Pattern 341

P2 - ***(optional)*** flags

Flags

n - avoid first regular "Enshure visible" for item
c - collapse the tree before going on
e - expand the tree
v - do a second "Enshure visible"
w - do a "Wait for Input Idle" with max time 5 seconds.
l - locate checkbox/Install-Box on the left, no click
a - Algo 2 to find a checkbox/Install-Box on the left side of the tree-item
i - "install" - opens the the selected MSI-Tree-menu

b - Check checkmark Item in a Treeview (direct engine)
d - UnCheck Checkmark Item in a Treeview (direct engine)

s - Select, select the treeview Item (direct engine)
m - Klick with MLE. on tree-item (direct engine)
o - click using compatible engine (MLI.)

You can combine all Flags, but not all combinations are useful. Also note that the order of the flags may influence the result. Generally the flags are processed from left to right.

P3 - ***(optional)*** number, if omitted, default number is 1.

Example

```
'***********************************************
' TVF.-Sample
'***********************************************

STW.ct|MMCMainFrame|Geräte
SCW.c|systreeview32

TVF.Bluetooth

TVF.&DOCAPS:&EXACT:HP Integrated Bluetooth module
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TVF.:&EXACT:HP Integrated Bluetooth module

TVF.Bluetooth||3
MMV.

'*********************************************
```

Remarks

On the stack, you can find the name of the selected element .
It's only the name of the element, not the full path to the element, like in `TVI.`.
The command has internally a different path for x32 and x64 applications.

Limitations:

-

See also:

- TVI. - Tree View Invisible 668
- GBC. - Get-Button-Check 640
- BTC. - Button-Check 635

© 2023 Theo Gottwald * https://www.smart-package.com

3.45.10 TVI. - Tree View Item

MiniRobotLanguage (MRL)

TVI. Command

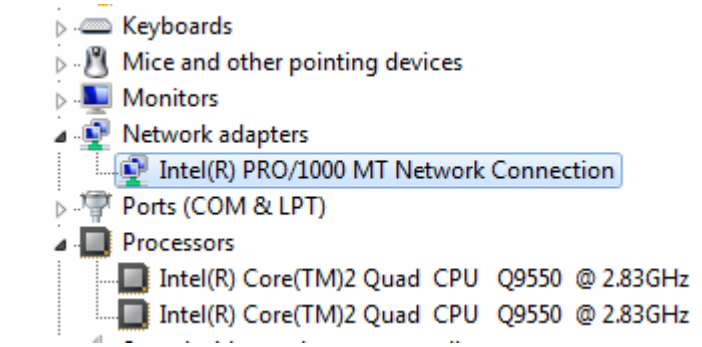
Tree View Item

Intention

These command selects an treeview Item. If necessary you can expand or collapse branches in treeviews. You can check checkmarks in treeviews. And you can open Menus in treevies, for example in MSI-Installations.

This command uses a special technology to also support x64 applications. However the results may be slower and slightly differ from the behavior in normal x32 applications.

A treeview is a special type of control and may look like this:



Please see here for more details and examples about TVI.and treeviews. 202

Example TVI.:

```
TVI.*\Tast*\*102*|e

' This uses a number to get the 3rd fitting element
TVI.*\Tast*\*102*|e|3
```

TVI. uses the direct engine for most flags.

There are two commands to work with treeviews.

```
TVI. - uses a path to the item and uses wildcard-patterns
TVF. - uses 3.2 Standard-Search Pattern 341
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

TVI.P1[|P2][|P3]

Parameter Explanation

P1 - item path and name can contain wildcards ? and *

P2 - ***(optional)*** flags

Flags

n - avoid first regular "Enshure visible" for item
c - collapse the tree before going on
e - expand the tree
v - do a second "Enshure visible"
w - do a "Wait for Input Idle" with max time 5 seconds.
l - locate checkbox/Install-Box on the left, no click
a - Algo 2 to find a checkbox/Install-Box on the left side of the tree-item
i - "install" - opens the the selected MSI-Tree-menu

b - Check checkmark Item in a Treeview (direct engine)
d - UnCheck Checkmark Item in a Treeview (direct engine)

s - Select, select the treeview Item (direct engine)
m - Klick with MLE. on tree-item (direct engine)
o - click using compatible engine (MLI.)

You can combine all Flags, but not all combinations are useful. Also note that the order of the flags may influence the result. Generally the flags are processed from left to right.

P3 - ***(optional)*** number, if omitted, default number is 1.

You have to specify the path to the destination item. You can use wildcards like '?' or '*' anywhere in the path.

```
' Example
TVI.*\Tast*\*102*
```

```
' Example, choose second fitting item
TVI.*\Core*||2
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************************
' TVI.-Sample
'************************************************

STW.ct|MMCMainFrame|Geräte
SCW.c|systreeview32
TVI.*\Tast*\*102*
MMV.
POP.$$TXT
CLP.t|$$TXT
PRT.$$TXT
MBX.!

'************************************************
```

Remarks

On the stack, you can find the name of the element (the full path).
The command has internally a different path for x32 and x64 applications.

Limitations:

-

See also:

- TVF. - Tree View Find 665
- GBC. - Get-Button-Check 640
- BTC. - Button-Check 635

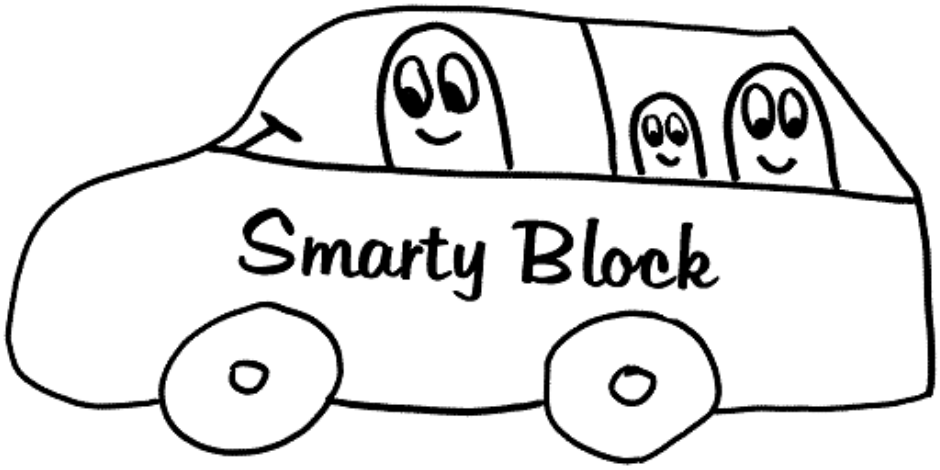
© 2023 Theo Gottwald * https://www.smart-package.com

3.46 Data-Block Commands

3.46.1 ! Smarty's Data-Block Commands

Smarty's Data-Block Commands

Data Blocks are sort of Array-Command.
While a normal Array can only hold Numbers, a Smarty Data-Block can hold any Item, even another Data Block.



More Info and example to Data blocks, see here:

1.5.1.5. Organizing data items in blocks 117

© 2023 Theo Gottwald * https://www.smart-package.com

3.46.2 CNB. - Create New Block

MiniRobotLanguage (MRL)

CNB. Command

Create New Block

Intention

What is a "data-block" ?
A data-block is a compound variable. A variable that contains multiple independent variables inside.

How do i create a data-block?
Use CNB. to create a new data-block container.

More on the CNB.-command:
This command is used to create a new data-block. Using data-blocks, any variable can be used to store multiple data items.

Using `CNB.` you can create a data item out of up to 20 data items **in one step**.

```
' Create a block with 3 elements
CNB.$$BLO|3|Element 1|Element 2|Element 3

' retrieve element Nr.1 into $$RES
GBE.$$BLO|1|$$RES

' Debug-Print the element in the editor
DBP.$$RES
ENR.
```

If you want to **haven more then 20 data-items** in the block, you need to create the block first, and add the items in a second step. Like this:

```
CNB.$$BLO|30
FOR.$$CNT|1|30
  VAR.$$ELE=Data-Item $$CNT
  SBE.$$BLO|$$CNT|$$ELE
NEX.
' Now see if they are really in there
FOR.$$CNT|1|30
  GBE.$$BLO|$$CNT|$$ELE
  DBP. ELement $$CNT=>$$ELE
NEX.
ENR.
```

There is no need to know perfectly how many block you will have at the end.
You can at any time extend (increase) the number of items in a data-block, without any items in that data-block being lost.

© 2023 Theo Gottwald * https://www.smart-package.com

Extending a data-block is done using `SIB.` or `SBE.`

For more details on 1.5.1.5. Organizing data items in blocks 117 see there.

Syntax

CNB.P1|P2[|P3]...[|PX]

Parameter Explanation

CNB.P1|P2

In this case **P1** is taken as Block-variable, **P2** is the number of the blocks to create. After creation, all blocks are there, while empty.

CNB.P1|P2[|P3 ... Px]

In this case **P1** is taken as Block-variable, **P2** is the number of the blocks to create. All further variables fill the blocks starting at block Nr. 1. This way you can create and define small blocks in one strike.

Example

```
'*********************************************
' GBE./CNB. - Sample
' Small block created in one strike
' with only 3 items
'*********************************************
CNB.$$BLO|3|Element 1|Element 2|Element 3
FOR.$$CNT|0|2
  GBE.$$BLO|($$CNT+1)|$$RES
  DBP.Element $$CNT is $$RES
NEX.
ENR.


'*********************************************
' SBE./CNB. - Sample 2
' Larger Block with >20 Items
'*********************************************
CNB.$$BLO|30
FOR.$$CNT|1|30
  VAR.$$ELE=Data-Item $$CNT
  SBE.$$BLO|$$CNT|$$ELE
NEX.
' Now see if they are really in there
FOR.$$CNT|1|30
  GBE.$$BLO|$$CNT|$$ELE
  DBP. ELement $$CNT=>$$ELE
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEX.
ENR.
```

Remarks

Blocks are internally checked for authenticity. This is done to prevent hard to track error.
If you change bytes or the length of a block variable externally, the block will not be unpacked anymore.
Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

Limitations:

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using `SBD.` `FEM.` and `SBM.` may be significantly faster.

See also:

- 1.5.1.3. Using Variables 92
- 1.5.1.5. Organizing data items in blocks 117
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685
- GFB. - Get-From-Block 680
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.46.3 GBE. - Get Block Element

MiniRobotLanguage (MRL)

GBE. Command

Get Block Element

Intention

This command can be used to retrieve a specified element from an block (another word for a data-array). A block can be created using the `CNB.` - command.

There are two Modes of operation for `GBE.`:
First, you can write:

```
CNB.$$BLO|3|Element 1|Element 2|Element 3
' Result is been place on the TOS
GBN.$$BLO|1
DBP.$$000
ENR.
```

Second you can do this:

```
CNB.$$BLO|3|Element 1|Element 2|Element 3
' Result is in $$RES
GBN.$$BLO|$$NUM|$$RES
DBP.$$RES
ENR.
```

For more details on 1.5.1.5. Organizing data items in blocks 117 see there.

Syntax

GBE.P1|P2[|P3]

Parameter Explanation

GBE.P1|P2

In this case **P1** is taken as Block-variable. **P2** is the number of the element to retrieve. The result is placed on TOS.

GBE.P1|P2|P3

In this case **P1** is taken as Block-variable, **P2** is the number of the element to retrieve.
P3 is the variable that will hold the result.

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*******************************************
' GBE./CNB. - Sample
'*******************************************
CNB.$$BLO|3|Element 1|Element 2|Element 3
GBN.$$BLO|$$RES
FOR.$$CNT|1|$$BLO
  GBE.$$BLO|$$CNT|$$RES
  DBP.Element $$CNT is $$RES
NEX.
ENR.
```

Remarks

Blocks are internally checked for authenticity. This is done to prevent hard to track error.
If you change bytes or the length of a block variable externally, the block will not be unpacked anymore.

Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

Limitations:

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using `SBD.` `FEM.` and `SBM.` may be significantly faster.

See also:

- 1.5.1.3. Using Variables 92
- 1.5.1.5. Organizing data items in blocks 117
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685
- GFB. - Get-From-Block 680

© 2023 Theo Gottwald * https://www.smart-package.com

3.46.4 GBN. - Get Block Number

MiniRobotLanguage (MRL)

GBN. Command

Get Block Elements Count

Intention

How many independent data-items are in a given data-block?
`GBN.` will tell you the answer to this question..

`GBN.` can be used to retrieve the number of elements in an block (another word for a data-array). A block can be created using the `CNB.` - command.

There are two Modes of operation for `GBE.`:
First, you can write:

```
CNB.$$BLO|3|Element 1|Element 2|Element 3
GBN.$$BLO
DBP.$$000
ENR.
```

Second you can do this:

```
CNB.$$BLO|3|Element 1|Element 2|Element 3
GBN.$$BLO|$$RES
DBP.$$RES
ENR.
```

In this case the result will be in a variable, instead of on `TOS`.

For more details on 1.5.1.5. Organizing data items in blocks 117 see there.

Syntax

GBN.P1[|P2]

Parameter Explanation

GBN.P1

In this case **P1** is taken as Block-variable. The result (number of blocks in that variable) is placed on TOS.

GBN.P1|P2

© 2023 Theo Gottwald * https://www.smart-package.com

In this case **P1** is taken as Block-variable, **P2** is the variable that will hold the result.

Example

```
'*******************************************
' GBE./CNB. - Sample
'*******************************************
CNB.$$BLO|3|Element 1|Element 2|Element 3
GBN.$$BLO|$$RES
FOR.$$CNT|1|$$RES
  GBE.$$BLO|$$CNT|$$ELE
  DBP.Element $$CNT is $$ELE
NEX.
ENR.
```

Remarks

Blocks are internally checked for authenticity. This is done to prevent hard to track error.
If you change bytes or the length of a block variable externally, the block will not be unpacked anymore.

Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

Limitations:

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using `SBD.` `FEM.` and `SBM.` may be significantly faster.

See also:

- 1.5.1.3. Using Variables 92
- 1.5.1.5. Organizing data items in blocks 117
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685
- GFB. - Get-From-Block 680

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.46.5 GFB. - Get-From-Block

MiniRobotLanguage (MRL)

GFB. Command

Get from Block

Intention

This command can be used to get several elements of an data-block in one strike. It saves you a bit typing compared to `GBE.`

When using `GFB.` there is no need to specify the Block-Variable. GFB. will automatically remember and use the last used block variable. Also the index (the first element number) is fix - the first parameter will retrieve element 1.

A short example, using `GFB.` and `SIB.` is:

```
CNB.$$BLO|6
SIB.Element 01|Element 02|Element 03
GFB.$$EL1|$$EL2|$$EL3
DBP.$$EL1$crlf$$$EL2$crlf$$$EL3
ENR.
```

Syntax

GFB.P1[|P2][|P3][|P4]..[|PX]

Parameter Explanation

P1- variable, `will hold first block element`

P2 ... Px (optional) `more variables, they will get all successive Block-Elements.`

Example

```
'*********************************************
' GFB./ SIB. - Sample
'*********************************************
CNB.$$BLO|6
SIB.Element 01|Element 02|Element 03
GFB.$$EL1|$$EL2|$$EL3
DBP.$$EL1$crlf$$$EL2$crlf$$$EL3
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Remarks

Blocks are internally checked for authenticity. This is done to prevent hard to track error.
If you change bytes or the length of a block variable externally, the block will not be unpacked anymore.

Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

Limitations:

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using SBD. FEM. and SBM. may be significantly faster.

See also:

- 1.5.1.3. Using Variables 92
- 1.5.1.5. Organizing data items in blocks 117
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685
- GFB. - Get-From-Block 680
- FER. - For-Each-Registry 935

© 2023 Theo Gottwald * https://www.smart-package.com

3.46.6 SBE. - Set Block Element

SBE. - Set Block Element **Previous** 680 **Top** 41 **Next** 685

MiniRobotLanguage (MRL)

SBE. Command

Set Block Element

Intention

This command can be used to Set individual elements of an data-block.

Using `SBE.` you can selectively Set individual data-items to an value you like.

Usage of `GBE.` is simple, see this example:

```
' Create a block with 3 elements
CNB.$$BLO|3|Element 1|Element 2|Element 3

' retrieve element Nr.1 into $$RES
GBE.$$BLO|1|$$RES

' Debug-Print the element in the editor
DBP.$$RES
ENR.
```

For more details on 1.5.1.5. Organizing data items in blocks 117 see there.

Syntax

SBE.P1|P2[|P3]

Parameter Explanation

SBE.P1|P2

In this case **P1** is taken as Block-variable, **P2** is the number of the block-element (1-based) to get whatever is on **TOS**. If TOS evaluates empty, the element will be set to empty.

P2 - If **P2** is higher then the actual number of blocks in data-block **P1**, then the data-block is automatically extended to that number of blocks.

SBE.P1|P2|P3

In this case **P1** is taken as Block-variable. **P2** is the number of the block-element (1-based) which is going to be filled with the content from variable **P3**.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*******************************************
' SBE./CNB. - Sample
'*******************************************
CNB.$$BLO|6000
FOR.$$CNT|1|6000
  VAR.$$ELE=Data-Item $$CNT
  SBE.$$BLO|$$CNT|$$ELE
NEX.
' Now see if they are really in there
FOR.$$CNT|1|6000
  GBE.$$BLO|$$CNT|$$ELE
  PRT. ELement $$CNT=>$$ELE
NEX.
ENR.
```

Remarks

Blocks are internally checked for authenticity. This is done to prevent hard to track error.
If you change bytes or the length of a block variable externally, the block will not be unpacked anymore.

Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

Limitations:

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using `SBD.` `FEM.` and `SBM.` may be significantly faster.

See also:

- 1.5.1.3. Using Variables 92
- 1.5.1.5. Organizing data items in blocks 117
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685
- GFB. - Get-From-Block 680

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.46.7 SIB. - Set-In-Block

MiniRobotLanguage (MRL)

SIB. Command

Set In Block

Intention

This command can be used to set successive elements of an data-block to values in one strike. It saves you a bit typing compared to `SBE.`

When using `SIB.` there is no need to specify the Block-Variable. `SIB.` will automatically remember and use the last used block variable. Also the index (the first element number) is fix - the first parameter will set element 1.

A short example, using `GFB.` and `SIB.` is:

```
CNB.$$BLO|6
SIB.Element 01|Element 02|Element 03
GFB.$$EL1|$$EL2|$$EL3
DBP.$$EL1$crlf$$$EL2$crlf$$$EL3
ENR.
```

`SIB.` will not create a new block, use `CNB.` for this purpose.

Syntax

SIB.P1[|P2][|P3]...[|PX]

Parameter Explanation

P1- content, `the first block element will be set to this value`

P2 ... Px (optional) `more variables, they will set all successive Block-Elements to these values.`

If the number of parameters is higher then the actual number of blocks in the data-block, then the data-block is automatically extended (blocks are added) to match that number of blocks.

Example

```
'*********************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' GFB./ SIB. - Sample
'*******************************************
CNB.$$BLO|6
SIB.Element 01|Element 02|Element 03
GFB.$$EL1|$$EL2|$$EL3
DBP.$$EL1$crlf$$$EL2$crlf$$$EL3
ENR.


'*******************************************
' GFB./ SIB. - Sample
'*******************************************
CNB.$$BLO|1
SIB.Hallo|Thomas|Lange
GBN.$$BLO|$$RES
'GBE.$$BLO|3|$$E01
DBP.Blo hat $$RES Elemente
GFB.$$E01|$$E02|$$E03
DBP.$$E01- $$E02 - $$E03
MBX.Blo hat $$RES Elemente
ENR.
```

Remarks

Blocks are internally checked for authenticity. This is done to prevent hard to track error.
If you change bytes or the length of a block variable externally, the block will not be unpacked anymore.

Data blocks can contain all 256 ASC Characters (even binary files) and are binary-compatible. Therefore whether special folders nor system variables are resolved inside blocks. Variables are resolved one time.

Limitations:

There is no explicit limit on data-items per block. However timing considerations may set you a limit at around 10000 Items, depending on your available hardware. For a larger number of non-binary items, Member-Operations, using `SBD.` `FEM.` and `SBM.` may be significantly faster.

See also:

- 1.5.1.3. Using Variables 92
- 1.5.1.5. Organizing data items in blocks 117
- CNB. - Create New Block 672
- GBN. - Get Block Number 677
- GBE. - Get Block Element 675
- SBE. - Set Block Element 682
- SIB. - Set-In-Block 685

© 2023 Theo Gottwald * https://www.smart-package.com

- GFB. - Get-From-Block 680

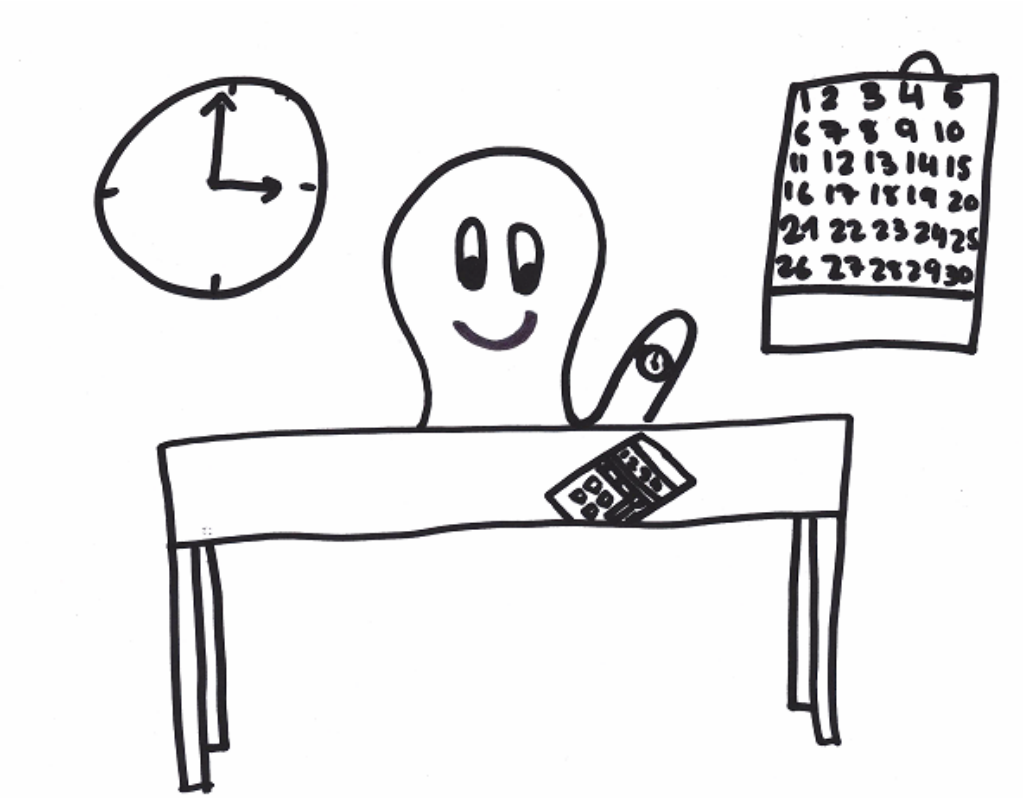
© 2023 Theo Gottwald * https://www.smart-package.com

3.47 Date and Time Calculation

3.47.1 ! Smarty's Date & Time Calculations

Smarty's Date and Time Calculations **Previous** 685 **Top** 41 **Next** 689

Smarty's Date and Time Calculations



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.47.2 DAT. - Date and Time Information (System independent)

DAT. - Date to Value Previous 688 Top 41 Next 693

MiniRobotLanguage (MRL)

DAT. Command

Date and Time Informations

Intention

This command returns date and time information.

Most of them are system-independent. No matter in which country the script will run, you will get exactly the same formatted time/date information.

There are several different Information you can retrieve using `DAT.`

For this you call the `DAT.`-command and specify an number from 0 to 10.
Here is a list of the numbers, and what they return.

```
DAT.0 ->Samstag<-
DAT.1 ->Februar<-
DAT.2 ->25<-
DAT.3 ->2012<-
DAT.4 ->11:28:16<-
DAT.5 ->11:28:16:406<-
DAT.6 ->6<- (DAY-Nr. of the WEEK)
DAT.7 ->25.02.2012<-
DAT.8 ->Saturday<- (DAYNAME)
DAT.9 ->SA<-     (DAYNAME-2 Letter)
DAT.10 -> 11:28<- (hh::MM)
DAT.11 ->2<-    (Month number)
DAT.12 ->02<-    (Month number with "0"-Prefix Format)
DAT.13 ->2020_12_01<-    (YYYY_MM_DD)
DAT.14 ->2020.12.01<-    (YYYY.MM.DD)
DAT.15 ->2020-12-01<-    (YYYY-MM-DD)
DAT.16 ->01_12_2020<-    (DD_MM_YYYY)
DAT.17 ->01122020<-    (DDMMYYYY)
DAT.18 ->20201201<-    (YYYYMMDD)
DAT.19 ->01<-    (DD)
```

For example:

```
DAT.10|$$RES
DBV.$$RES
ENR.
```

You can also use the same variable for source and target:

```
$$RES=10
DAT.$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

DBV.$$RES
ENR.

Syntax

DAT.[P1][|P2]

Parameter Explanation

P1 - ***(optional)*** If omitted or empty, the actual time is returned. The result will be placed on the TOS. If you specify an variable in **P1**, and **P2** is omitted, then the result will be placed in that variable **P1**.

These are the numbers you can specify as **P1, and what they return:**

```
 ' Returns the Dayname in computer local language
 DAT.0 ->Samstag<-

' Returns the  Monthname in computer local language
 DAT.1 ->Februar<-

 ' Returns the Daynumber in the month
 DAT.2 ->25<-

 ' Returns the actual year
 DAT.3 ->2012<-

 ' Returns the actual time in fixed format.
 DAT.4 ->11:28:16<-

 ' Returns the actual time in fixed format incl. ms.
 DAT.5 ->11:28:16:406<-

 ' Returns the day of the week, numeric
 DAT.6 ->6<-

 ' Returns the actual date in fixed format
 DAT.7 ->25.02.2012<-

 ' Returns todays day in english
 DAT.8 ->Saturday<-

 ' Returns first two letters of todays day (engl.)
 DAT.9 ->SA<-

 ' Returns how many days this actual month has
 DAT.10->11:28<-
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Returns the Month-Number
 DAT.11 ->2<-     (Month number)

' Returns the Month number in 2-Prefix "00"
 DAT.12 ->02<-     (Month number with "0"-Prefix Format)

' Returns the Date (YYYY_MM_DD)
 DAT.13 ->2020_12_01<-

' Returns the Date  (YYYY.MM.DD)
 DAT.14 ->2020.12.01<-

' Returns the Date   (YYYY-MM-DD)
 DAT.15 ->2020-12-01<-

' Returns the Date  (DD_MM_YYYY)
 DAT.16 ->01_12_2020<-

' Returns the Date (DDMMYYYY)
 DAT.17 ->01122020<-

' Returns the Date  (YYYYMMDD)
 DAT.18 ->20201201<-

' Returns the Day-number in Month in "00"-Format
 DAT.19 ->01<-     (DD)
```

P2 - ***(optional)*** If specified, this variable will be used for the resulting value. If omitted, and P1 is a variable, P1 will be used. Else the result will be placed on TOS.

Example

```
'*******************************************
' DAT. - Sample
'*******************************************
'
FOR.$$CNT|0|10
  DAT.$$CNT|$$RES
  DBP.DAT.$$CNT->$$RES<-
NEX.
ENR.
DAT.10
DBV.$$000
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

All Date-calculations should be accurate up to a Millisecond at least back until 1601.

See also:

- 1.5.1.3. Using Variables 92
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.3 DTC. - Date and Time Calculation

MiniRobotLanguage (MRL)

DTC. Command

Date- and Time Calculations

Intention

What is the difference between date and time?
A date is this: `"23.05.2014"`.
A time is this: `"15:12:45"`.

What does DTC. do?

- delivers you Date in any format you want.
- delivers you Time in any format you want.
- allows you to add or
- to subtract from date or time

`DTC.` can give you the current date, or current time.
It can also reformat the current date or time for you into any format you want.

It can do any date-/time calculation for you. And also mix date and time calculations. For example:

- What do i get, if i add 5000 seconds and 403 days to the 12.12.2013, 9:12:53 ?
- What date was 12316 seconds before the 01.01.2014?
- What time will be in 5 hours and 15 Minutes from now?

These are just examples, to show the flexibility of `DTC.` Any of these and many more questions can quickly be solved using `DTC.`

In short: Its a complete all-in-one solution for date/time calculations and re-formatting of the result into the format that you like.
When you need to deal with dates or with time, this is the command that may help you.

And after all, we have tried hard to make all this as easy as possible, for you.

Generally the `DTC.` command has two modes of operation:

- formatting date and/or time (2 Parameters)
- formatting *and* calculation (3 Parameters)

To give you a quick overview, look at the general use of the 2 or 3 parameters:

P1 - is the variable for the source date and it will also receive the result after all
P2 - is the mask for the format of the result
P3 - is the parameter for adding and subtracting to the date/time

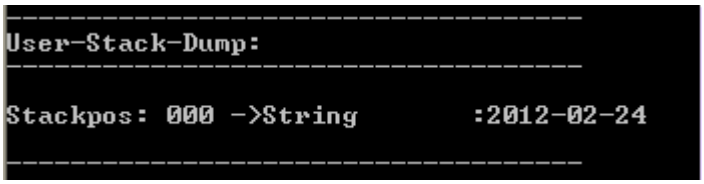
*© 2023 Theo Gottwald * https://www.smart-package.com*

Note that there are exceptions from this general rules, see the details below in the section "Parameter".

Let us first start with an simple example. We have given the term "now" in **P1**, this shows that we want to get the current date/time. Then this script:

```
DTC.now|yyyy-mm-dd
STS.DUMP
```

Will put the date, formatted like we want it, on the TOS 104,
The TOS is used, if we do not specify a variable in **P1**.



Next, let us use a variable for **P1**, then we will get the return value in that variable. Anyway we want to use the term "now" also, to get the actual date.

```
: $$TIM=now
DTC.$$TIM|dd
MBX.$$TIM
```

In the same way you can get any date or time component into the variable.
What date is in 90 days from today?

```
: $$TIM=now
DTC.$$TIM|dd|AD:90
MBX.$$TIM
```

Next lets take a look at time calculations. We can get the actual time in the same way:

```
: $$TIM=now
DTC.$$TIM|hh:nn
MBX.$$TIM
```

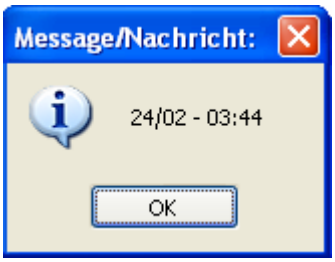
And of course we can add and subtract from that also.
What time was 29000 seconds before now?

```
: $$TIM=now
DTC.$$TIM|hh:nn|SS:29000
MBX.$$TIM
```

Sure, this was on another day, lets see which day. We will just need to tweak the format-parameter a bit. Like this:

```
: $$TIM=now
DTC.$$TIM|dd/mm - hh:nn|SS:29000
MBX.$$TIM
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

DTC.P1|P2[|P3]...[|PX]

Parameter Explanation

P1 - (can be empty) Variable with base-time and or base-date, or can contain the "now", or can be empty.
If the term "now" is specified, or **P1** evaluates to empty, current date and time is taken as base-time for further formatting and calculations.
If variable is given in **P1**, the result is placed in the variable else the result is placed on TOS.

P2 - Output-Mask. Mask that contains the letters below. These will be replaced with the values of the date/time-result. You can specify the following letters in the output-mask:

y - Year
m - Month
d - Day
h - Hours
n - Minutes
s - Seconds
t - Milliseconds

Example: the mask "dd-mm" will be replaced with "12-03" if the date was the 12.03.XXXX. Time works in the same way. You can specify a mask that contains time and date:

DTC.now|yyyy-mm-dd,hh:nn

If you specify less characters then the original value needs, for example: "yy" instead of "yyyy", then you will get the numbers from the right side, in this case "15", if the result year is 2015.
A mask of "mm-yy" could be for example "03-15" in the march 2015.

© 2023 Theo Gottwald * https://www.smart-package.com

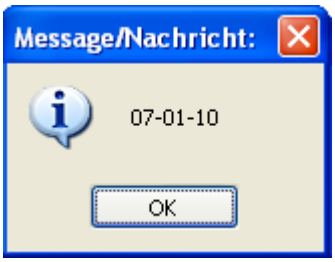
P3 - (optional) - Add / Subtract Parameter. If you want to add or subtract time to/from the base-time, then you can do that using **P3**. Here you specify the amount of time that should be added or subtracted. Generally all Options are made of two uppercase letters. The first letter specifies what to do:

```
A - Add something to the base-time/date
S - Subtract something from the base-time/date
N - Set the value of the base-time/date to this number before
    addition/subtraction.

The second letter specifies the time/date parameter that is
going to be changed. The used letters are the same like in the
list above: Y- Year, until ... T - Milliseconds.

You can use any combination of the following options:

NY:(number) ' Set Years
AY:(number) ' Add Years
SY:(number) ' Subtract Years

NM:(number) ' Set Month
AM:(number) ' Add Month
SM:(number) ' Subtract Month

ND:(number) ' Set days
AD:(number) ' Add days
SD:(number) ' Subtract days

NH:(number) ' Set hours
AH:(number) ' Add hours
SH:(number) ' Subtract hours

NN:(number) ' Set Minutes
AN:(number) ' Add Minutes
SN:(number) ' Subtract Minutes

NS:(number) ' Set Seconds
AS:(number) ' Add Seconds
SS:(number) ' Subtract Seconds

NT:(number) ' Set Milliseconds
AT:(number) ' Add Milliseconds
ST:(number) ' Subtract Milliseconds
```

For example, your base-time is the 01.01.2010 and you add 1222 Minutes (thats 20 hrs.) and 5 days, you will write:

```
: $$TIM=01.01.2010
' Additions and Subtractions get separated by a comma
DTC.$$TIM|dd-mm-yy|AN:1222,AD:5
MBX.$$TIM
```

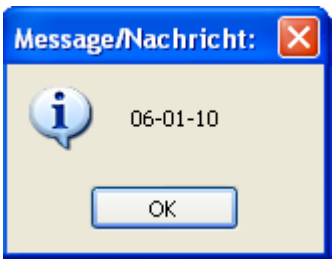

© 2023 Theo Gottwald * https://www.smart-package.com



All parameters that are not specified, are replaced with current time! In this case, the base-time is the current time when the script runs. To be more exact we would therefore need to also specify a base-time:

```
: $$TIM=01.01.2010,01:00
DTC.$$TIM|dd-mm-yy|AN:1222,AD:5
MBX.$$TIM
```

and the result will be:



You can specify as many Additions and subtractions as you want. Separate them with a comma.

Example

```
'*********************************************
' DTC. - Sample
' Add 1222 Minutes and 5 Days
' to the time in $$TIM
'
'*********************************************
: $$TIM=01.01.2010,01:00
DTC.$$TIM|dd-mm-yy|AN:1222,AD:5
MBX.$$TIM
ENR.


'*********************************************
' DTC. - Sample
' What date was 149385 days earlier?
'*********************************************
: $$TIM=now
DTC.$$TIM|dd-mm-yyyy|SD:149385
MBX.$$TIM
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

All Date-calculations should be accurate up to a Millisecond at least back until 1601.

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- DTV. - Date to Value 746
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.4 IAT. / NAT. - If After Timestamp

MiniRobotLanguage (MRL)

IAT. / NAT. Conditional Statement

If After Timestamp / If NOT After Timestamp

Intention

What is a TimestamP
A Timestamp is a Time-Value in the Format: `HH:MM:SS` or
`HH:MM:SS.mmm`

Using the TCA. and these other Time-Stamp Commands you can work with Timestamps.

IAT. will need two timestamps as Parameter. It will then compare both Timestamps and decide
if Timestamp **P1** is AFTER Timestamp **P2**.
If Timestamp **P1** is equal to Timestamp **P2** then the time **P1** is **NOT** After **P2**.
If you specify an `@` Sign in front of **P1** then the statement will evaluate if **P1 = P2** means If Timestamp 1 is equal to Timestamp 2 instead.
In case of an error, the `IAT.` will by default evaluate to true, but will raise an error on TOS.

Hint: `$time$` will return a timestamp like: `12:01:40`
If one or both Parameters include `".000"` Microseconds, then these will also be evaluated.
If Microseconds are missing, they will be ignored.

The negative Form of this statement is `NAT`.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$TIA=10:46:24
$$TIB=10:46:29

IAT.$$TIB|$$TIA
DBP. $$TIB is AFTER $$TIA
ELS.
DBP. $$TIB is NOT AFTER $$TIA
EIF.
```

```
[11:08:26] Done.
[11:08:27] 10:46:29 is AFTER 10:46:24
```

```
$$TIA=10:47:24
$$TIB=10:48:29

IAT.$$TIA|$$TIB
DBP. $$TIA is AFTER $$TIB
ELS.
DBP. $$TIA is NOT AFTER $$TIB
EIF.
```

```
[11:22:00] 10:47:24 is NOT AFTER 10:48:29
```

Here is the special Mode that uses the @ Operator.

© 2023 Theo Gottwald * https://www.smart-package.com

```
$$TIA=@11:45:59
$$TIB=11:45:59

IAT.$$TIA|$$TIB
DBP. $$TIA is $$TIB
ELS.
DBP. $$TIA is NOT $$TIB
EIF.
```

```
[11:53:30] @11:45:59 is 11:45:59
```

And the other case:

```
$$TIA=@11:45:59
$$TIB=11:45:49

IAT.$$TIA|$$TIB
DBP. $$TIA is $$TIB
ELS.
DBP. $$TIA is NOT $$TIB
EIF.
```

```
[11:55:44] @11:45:59 is NOT 11:45:49
```

Syntax

```
IAT.[@]P1|P2 ... ELS. ... EIF.
NAT.[@]P1|P2 ... ELS. ... EIF.
```

Parameter Explanation

P1 – Timestamp 1 that needs to be AFTER **P2** for this to evaluate to TRUE.

© 2023 Theo Gottwald * https://www.smart-package.com

If you specify an @ Sign in front of **P1** then the statement will evaluate if **P1 = P2** means If Timestamp 1
is equal to Timestamp 2 instead.

P2 – Timestamp 2

Example

```
'*********************************************
' IAT. - Sample
' Timeloop will produce Timestamps with an Intervall
'*********************************************
'
$$TIA=08:09:59.000
$$TIB=08:10:01.000
$$LOP=$$TIA
$$INC=00:00:00.010
$$LOV=0
DOL.
  TCA.Add|$$LOP|$$INC|$$LOP
  PRT.Looping: $$LOP
  IAT.$$LOP|$$TIB
    $$LOV=1
   EIF.
OOP.$$LOV
MBX.$$LOP
ENR.

'*********************************************
' IAT. - Sample Macros
'*********************************************
':Insert_TPS
$$EIN=00:00:01
$$ZWO=00:00:03
VAN.$$CNT=0
TCA.Add|$$TIA|$$EIN|$$STA
TCA.Dif|$$TIB|$$ZWO|$$ENA
%FOT $$STA|$$ENA|$$EIN
 VIC.$$CNT
 $$TIM=$$LOP
%NXT
RET.

'-------------------------------------------------------
' These two Macros define a Timestamp-FOR Loop
' So you can Loop from P1 to P2 using Step P3
'
: %FOT 3
$$FOA=§§§01
$$FOB=§§§02
$$FOC=§§§03
$$LOP=$$FOA
DOL.
'EIF-
END%
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'------------------------------------------------------------
'
'------------------------------------------------------------
'
: %NXT
TCA.Add|$$LOP|$$FOC|$$LOP
BCH.$$LOP|.|1|$$TOP
BCH.$$FOB|.|1|$$BOP
IAT.$$TOP|$$BOP
$$LOV=1
ELS.
$$LOV=0
EIF.
PRT.$$TOP / $$BOP  ($$LOV)
OOP.$$LOV
'EIF+
END%
```

Remarks

-

Limitations:

-

See also:

- GFD. - Get File Date 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.5 IBT. / NBT. - If Before Timestamp

MiniRobotLanguage (MRL)

IBT. / NBT. Conditional Statement

If Before Timestamp / If NOT Before Timestamp

Intention

What is a TimestamP
A Timestamp is a Time-Value in the Format: `HH:MM:SS` or `HH:MM:SS.mmm`

Using the TCA. and these other Time-Stamp Commands you can work with Timestamps.

IBT. will need two timestamps as Parameter. It will then compare both Timestamps and decide
if Timestamp **P1** is BEFORE Timestamp **P2**.
If Timestamp **P1** is equal to Timestamp **P2** then the time **P1** is **NOT** BEFORE **P2**.
If you specify an `@` Sign in front of **P1** then the statement will evaluate if **P1 = P2** means If Timestamp 1 is equal to Timestamp 2 instead.

In case of an error, the `IBT.` will by default evaluate to true, but will raise an error on TOS.

Hint: `$time$` will return a timestamp like: `12:01:40`
If one or both Parameters include `".000"` Microseconds, then these will also be evaluated.
If Microseconds are missing, they will be ignored.

The negative Form of this statement is NBT.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$TIA=10:47:24
$$TIB=10:48:29

IBT.$$TIA|$$TIB
DBP. $$TIA is BEFORE $$TIB
ELS.
DBP. $$TIA is NOT BEFORE $$TIB
EIF.
```

```
[11:24:11] 10:47:24 is BEFORE 10:48:29
```

```
$$TIA=11:45:59
$$TIB=10:46:29

IBT.$$TIA|$$TIB
DBP. $$TIA is BEFORE $$TIB
ELS.
DBP. $$TIA is NOT BEFORE $$TIB
EIF.
```

```
11:48:37] Done.
11:48:39] 11:45:59 is NOT BEFORE 10:46:29
```

Now here is the special Mode using the @ Operator.

© 2023 Theo Gottwald * https://www.smart-package.com

```
$$TIA=@11:45:59
$$TIB=11:45:59

IBT.$$TIA|$$TIB
DBP. $$TIA is $$TIB
ELS.
DBP. $$TIA is NOT $$TIB
EIF.
```

```
[11:50:38] Done.
[11:50:38] @11:45:59 is 11:45:59
```

And finally.

```
$$TIA=@11:45:59
$$TIB=11:46:59

IBT.$$TIA|$$TIB
DBP. $$TIA is $$TIB
ELS.
DBP. $$TIA is NOT $$TIB
EIF.
```

```
[11:52:19] Done.
[11:52:20] @11:45:59 is NOT 11:46:59
```

Syntax

```
IBT.[@]P1|P2 ... ELS. ... EIF.
NBT.[@]P1|P2 ... ELS. ... EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 – Timestamp 1 that needs to be AFTER P2 for this to evaluate to TRUE.
If you specify an @ Sign in front of **P1** then the statement will evaluate if **P1 = P2** means If Timestamp 1 is equal to Timestamp 2 instead.

P2 – Timestamp 2

Example

```
'*********************************************
' IAT. - Sample Macros
'*********************************************
':Insert_TPS
$$EIN=00:00:01
$$ZWO=00:00:03
VAN.$$CNT=0
TCA.Add|$$TIA|$$EIN|$$STA
TCA.Dif|$$TIB|$$ZWO|$$ENA
%FOT $$STA|$$ENA|$$EIN
 VIC.$$CNT
 $$TIM=$$LOP
%NXT
RET.

'------------------------------------------------------------
' These two Macros define a Timestamp-FOR Loop
' So you can Loop from P1 to P2 using Step P3
'
: %FOT 3
$$FOA=§§§01
$$FOB=§§§02
$$FOC=§§§03
$$LOP=$$FOA
DOL.
'EIF-
END%
'------------------------------------------------------------
'
'------------------------------------------------------------
'
: %NXT
TCA.Add|$$LOP|$$FOC|$$LOP
BCH.$$LOP|.|1|$$TOP
BCH.$$FOB|.|1|$$BOP
IAT.$$TOP|$$BOP
$$LOV=1
ELS.
$$LOV=0
EIF.
PRT.$$TOP / $$BOP  ($$LOV)
OOP.$$LOV
'EIF+
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END%
```

Remarks

-

Limitations:

-

See also:

- **GFD. - Get File Date** 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.6 IDC. / NDC. - If Date Condition

MiniRobotLanguage (MRL)

IDC. Conditional Statement

If Date Condition

Intention

Conditional Statement. Test multiple conditions about the current date or time.
You can test:

- if the current date is before or after a specified date.
- if the current time is before or after a specified time.
- if we have a specified date or time.
- if a specified year is a leap year

Let us start with this:

```
' Get Leap Years between 1997 and 2012
FOR.$$CNT|1997|2012
' This will do the magic
  IDC.is_lj|$$CNT
    DBP.$$CNT is a leap year!
  EIF.
NEX.
ENR.
```

You can test if we have a given date:

```
: $$HRR=23.2.2012
IDC.date_equal|$$HRR
  DBV.Yes, we have the $$HRR
ELS.
  DBV.No, its not the $$HRR
EIF.
ENR.
```

You can test if a given number is the current year:

```
: $$CNT=2012
IDC.is_year|$$CNT
    DBP.$$CNT is the current year!
EIF.
```

In the same way you can check the month, day, hour, minute, second.

If you need to check if the current date is before or after a given date, its as easy as before:

© 2023 Theo Gottwald * https://www.smart-package.com

```
' This is the date, we check
: $$CNT=26.02.2012
' and this will do the magic
IDC.date_after|$$CNT
  DBP.$$CNT is after current time!
EIF.
IDC.date_before|$$CNT
  DBP.$$CNT is before current time!
EIF.
```

Note that you need to give a complete date with all parts, in the order `DD-MM-YYYY` for this to work. This is not the case when we do the same with times. With time we have more options. You can find out, if a given time is before or after the actual time (now).

When you specify a time, you can choose, what time you want to specify:

- the hour
- the hour and the Minute
- the hour and the minute and seconds
- the hour and the minute, seconds and milliseconds

This way you can check if its before or after 12 o'clock or you can check if its before or after 12:37:29. Just specify what you want to have. Generally the command looks like this:

```
: $$CNT=08:36
' check if time is yet to come
IDC.time_after|$$CNT

' Check for past time
IDC.time_before|$$CNT

' Check if time is now
IDC.time_equal|$$CNT
```

Here is an complete running example .

```
' Change the time (next line) to any time you want
: $$CNT=17:36:59

IDC.time_after|$$CNT
  DBP.$$CNT is after current time!
EIF.

IDC.time_equal|$$CNT
  DBP.$$CNT is current time!
EIF.

IDC.time_before|$$CNT
  DBP.$$CNT is before current time!
EIF.
ENR.
```

`IDC.` can also be used to directly test components of date and time.
Let us start with the date. We want to know if its currently 2012.

© 2023 Theo Gottwald * https://www.smart-package.com

```
: $$YEA=2012
IDC.is_year|$$YEA
  DBV.Yes, we have $$YEA
ELS.
  DBV.No, its not $$YEA
EIF.
ENR.
```

You can test in the same way the actual month or day, see examples below.

```
'  Test the month
: $$MON=3
IDC.is_month|$$MON
  DBV.Yes, we have $$MON
ELS.
  DBV.No, its not $$MON
EIF.
ENR.

'  Test the actual day, if its the 24 th
: $$DAY=24
IDC.is_day|$$DAY
  DBV.Yes, we have $$DAY
ELS.
  DBV.No, its not $$DAY
EIF.
ENR.
```

Also the same will work with time. You can test the hour or minutes or even the seconds.
In just the same way. Here is an example:

```
' Test if we have 19:xx o'clock
: $$HRR=19
IDC.is_hour|$$HRR
  DBV.Yes, we have $$HRR
ELS.
  DBV.No, its not $$HRR
EIF.
ENR.
```

Hint:
`IDC.` will internally verify the date- and the time format when you use the `date_after,date_before,time_after,time_after` options. However the order of the parameters will not be touched. You need to specify all dates like this:`DD-MM-YYYY` and all times like this: `HH:MM:SS or HH:MM:SS:MS`.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

IDC.P1[|P2][,P3] … ELS. … EIF.

Parameter Explanation

P1 **-** prototype

Prototype can be omitted or any of:
- "is_leap_year" - test if the actual/specified year is a leap year or not.
- "is_year" - test if the given year matches the actual date (now).
- "is_month" - test if the given year matches the actual date (today).
- "is_day" - test if the given year matches the actual date (today).
- "is_hour" - test if the given number matches the actual time (now).
- "is_minute" - test if the number matches the actual time (now).
- "is_second" - test if the given number matches the actual time (now).
- "time_after" - test if specified time is after actual time.
- "time_before" - check if specified time is in the past
- "time_equal" - check if specified time is right now
- "date_after" - test if specified date is after actual date
- "date_before" - check if specified date is in the past
- "date_equal" - check if specified date is today

P2 **-** (optional) - The usage of **P2** and **P3** depends on the prototype **P1**

Important:: If you specify a Year below 100, for example 23.02.12, the time-logic will add 2000 to it. Therefore the 30.03.12 will be internally treated like the 30.03.2012. We have made this addition, because we believe that the chances that somebody wants to make calculations with the Year 0 are lower, then the chance that somebody will forget the first two number for the actual date. Here is an example:

```
'  this date will be treated as 24.02.2012
: $$HRR=24.2.12
IDC.date_after|$$HRR
  DBV.Yes, the $$HRR is in the future
ELS.
  DBV.No, the $$HRR is not after today
EIF.
ENR.
```

Example

```
'**********************************
' IDC.-Sample (Leap Year)
'**********************************
'
' Get Leap Years between 1997 and 2012
FOR.$$CNT|1997|2012
  IDC.is_lj|$$CNT
    DBP.$$CNT is a leap year!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  EIF.
NEX.
ENR.


'**********************************
' IDC.-Sample (time)
'**********************************
' Use a time from now in 30 seconds
: $$CNT=08:36:00

FOR.$$LOP|1|1000
  IDC.time_after|$$CNT
    DBP.$$CNT is after current time!
  EIF.
  IDC.time_before|$$CNT
    DBP.$$CNT is before current time!
  EIF.
  PAU.0.5
NEX.
ENR.

'**********************************
' IDC.-Sample (Year)
'**********************************
: $$YEA=2012
IDC.is_year|$$YEA
  DBV.Yes, we have $$YEA
ELS.
  DBV.No, its not $$YEA
EIF.
ENR.

'**********************************
' IDC.-Sample (Month)
'**********************************
'  Test the month
: $$MON=3
IDC.is_month|$$MON
  DBV.Yes, we have $$MON
ELS.
  DBV.No, its not $$MON
EIF.
ENR.

'**********************************
' IDC.-Sample (date is)
'**********************************
'
: $$HRR=23.2.2012
IDC.date_equal|$$HRR
  DBV.Yes, we have the $$HRR
ELS.
  DBV.No, its not the $$HRR
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- **VTT. - Value to Time** 760
- **TTV. - Time to Value** 751
- **DTV. - Date to Value** 746
- **VTD. - Value to Date** 756
- **DTC. - Date and Time Calculation** 693
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7 TCA. - Time Calculation

MiniRobotLanguage (MRL)

TCA. Command

Time Calculation

Intention

Using TCA. you can easily work with Times in this Standard-Format:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - sometimes not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.023` this will be treated as `00:00:00.023`

If the addition of any of the values goes beyond their limit, the Carry is added to the next higher Parameter.

Example

-

Remarks

-

Limitations:

-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756

*© 2023 Theo Gottwald * https://www.smart-package.com*

- **DTC. - Date and Time Calculation** 693
- **VTF. - Verify Time Format** 742
- **VDF. - Verify Date Format** 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.1 TCA.Add

MiniRobotLanguage (MRL)

TCA.Add - Command

Add 2 Timestamps

Intention

Using TCA. you can easily add Timestamps in this Standard-Format:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.023` this will be treated as `00:00:00.023`

If the addition of any of the values goes beyond their limit, the Carry is added to the next higher Parameter.

Sample: Add (Add Timestamps)

```
' Technically the robot works with 08:09:59:098
$$LAA=08:14:59.98
' Technically the robot works with 12:00:00:002
$$LAB=12:59:5.2
TCA.Add|$$LAA|$$LAB|$$LAC
' Result is  21:14:04:100
MBX.$$LAC
ENR.
```

Syntax

TCA.Add|P1|P2[|P3]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 – Timestamp A
P2 – Timestamp B
P3 – Variable for Result Timestamp. If omitted the result is placed on **TOS**.

Example

```
'*********************************************
' TCA. - Sample
'*********************************************
'
$$LAA=08:11:12.124
$$LAB=12:11:12.123
$$ERG=20:22:24.247
$$MSG=Teste TCA. 1
TCA.Add|$$LAA|$$LAB|$$LAC
NAT.@$$LAC|$$ERG
  MBX.Result is wrong.
ELS.
  MBX.Result is $$ERG
EIF.
ENR.

'*********************************************
' TCA. / IAT. - Sample
' Timeloop will produce Timestamps with an Interval
'*********************************************
'
$$TIA=08:09:59.000
$$TIB=08:10:01.000
$$LOP=$$TIA
$$INC=00:00:00.010
$$LOV=0
DOL.
  TCA.Add|$$LOP|$$INC|$$LOP
  PRT.Looping: $$LOP
  IAT.$$LOP|$$TIB
    $$LOV=1
  EIF.
OOP.$$LOV
MBX.$$LOP
ENR.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.2 TCA.Between

MiniRobotLanguage (MRL)

TCA.Betweeen - Command

Get Time between 2 Timestamps

Intention

With TCA.Between you can get the exact time between two timestamps in this standard format:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.23` this will be treated as `00:00:00.023`

The result will be the same if the second parameter P2 is larger (later) than P1 or earlier.

Sample: Between (Get Timestamp between 2 Timestamps)

```
' Simple Example
$$LAA=00:00:00.000
' Technically the robot works with 12:00:00:002
$$LAB=12:00:00.2
TCA.Bet|$$LAA|$$LAB|$$LAC
' Result is  06:00:00:001
MBX.$$LAC
ENR.

' Difficult Example
$$LAA=23:59:36.740
$$LAB=12:00:00.212
TCA.Bet|$$LAA|$$LAB|$$LAC
' Result is  17:59:58:476
MBX.$$LAC
ENR.

' Simpler Example
$$LAA=10:00:00.000
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$LAB=20:20:10.500
TCA.Bet|$$LAA|$$LAB|$$LAC
' Result is  15:10:05:250
MBX.$$LAC
ENR.
```

Syntax

TCA.Bet|P1|P2[|P3]
TCA.Between|P1|P2[|P3]

Parameter Explanation

P1 – Timestamp A
P2 – Timestamp B
P3 – Variable for Result Timestamp. If omitted the result is placed on **TOS**.

Example

```
'*********************************************
' TCA. - Sample
'*********************************************
'
$$LAA=03:11:11.111
$$LAB=05:55:55.555
$$ERG=04:33:33.333
$$MSG=Teste TCA.Between
TCA.Bet|$$LAA|$$LAB|$$LAC
NAT.@$$LAC|$$ERG
  MBX.Result is wrong.
ELS.
  MBX.Result is $$ERG
EIF.
ENR.


'*********************************************
' TCA. / IAT. - Sample
' Timeloop will produce Timestamps with an Interval
'*********************************************
'
$$TIA=08:09:59.000
$$TIB=08:10:01.000
$$LOP=$$TIA
$$INC=00:00:00.010
$$LOV=0
DOL.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  TCA.Add|$$LOP|$$INC|$$LOP
  PRT.Looping: $$LOP
  IAT.$$LOP|$$TIB
    $$LOV=1
  EIF.
OOP.$$LOV
MBX.$$LOP
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.3 TCA.Diff

MiniRobotLanguage (MRL)

TCA.Diff - Command

Subtracts 2 Timestamps

Intention

Using TCA.Diff you can easily Subtracts Timestamps in this Standard-Format:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.023` this will be treated as `00:00:00.023`

If the result of any of the values goes beyond their limit, the Carry is added to the next higher Parameter.
If the result is smaller then P1, the result will roll over to the previous time.

Sample: Diff (Add Timestamps)

```
' Here P1 is smaller then P2, therefore the result will roll over
$$LAA=01:24:12.001
$$LAB=02:50:13.002
$$ERG=22:33:58:999
$$MSG=Teste TCA. 4
TCA.Dif|$$LAA|$$LAB|$$LAC
MBX.$$LAC
NAT.@$$LAC|$$ERG
  GTO.Lab_failed
EIF.
ENR.

' Here P1 is larger then P2 threrfore this is a normal Subtraction
$$LAA=02:50:13.002
$$LAB=01:24:12.001
$$ERG=01:26:01:001
$$MSG=Teste TCA. 4
TCA.Dif|$$LAA|$$LAB|$$LAC
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  MBX.$$LAC
  NAT.@$$LAC|$$ERG
    GTO.Lab_failed
  EIF.
  ENR.
```

Syntax

TCA.Diff|P1|P2[|P3]
TCA.Dif|P1|P2[|P3]

Parameter Explanation

P1 – Timestamp A
P2 – Timestamp B
P3 – Variable for Result Timestamp. If omitted the result is placed on **TOS**.

Example

```
'*********************************************
' TCA.Dif - Sample
'*********************************************
'
$$LAA=08:11:12.124
$$LAB=12:11:12.123
$$ERG=20:00:00.001
$$MSG=Teste TCA. 1
TCA.Dif|$$LAA|$$LAB|$$LAC
NAT.@$$LAC|$$ERG
  MBX.Result is wrong.
ELS.
  MBX.Result is $$ERG
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.4 TCA.Divide

MiniRobotLanguage (MRL)

TCA.Divide - Command

Divide a Timestamp

Intention

Using TCA.Div you can easily divide Timestamps in this Standard-Format:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.023` this will be treated as `00:00:00.023`

If any of the values goes beyond their limit, the Carry is added to the next higher Parameter.
The Division is always an Integer Division, means if **P2** is "2.5" then the Robot will use a "2".

The Hours are hereby not limited to 23 because this way you can also calculate working times.

Sample: Div (Divide Timestamps)

```
$$LAA=30:14:12.900
$$LAB=2
TCA.Div|$$LAA|$$LAB|$$LAC
' Result is  15:07:06.450
MBX.$$LAC
ENR.

' Lets say the total amount of time to do the Job would be 45 Hours
' And you can use 5 Computers for it.
$$LAA=45:01:4.100
$$LAB=5
TCA.Div|$$LAA|$$LAB|$$LAC
' Result is  09:00:12.820
DBP.$$LAC
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

TCA.Divide|P1|P2[|P3]
TCA.Div|P1|P2[|P3]

Parameter Explanation

P1 – Timestamp A
P2 – Timestamp B
P3 – Variable for Result Timestamp. If omitted the result is placed on **TOS**.

Example

```
'*********************************************
' TCA.Div - Sample
'*********************************************
'
$$LAA=08:11:12.124
$$LAB=12
$$ERG=20:22:24.247
$$MSG=Teste TCA. 1
TCA.Div|$$LAA|$$LAB|$$LAC
' Result is 00:40:56.010
NAT.@$$LAC|$$ERG
  MBX.Result is wrong.
ELS.
  MBX.Result is $$ERG
EIF.
ENR.
```

Remarks

-

Limitations:

Its a Integer Division. You can not divide using FP Numbers like "3.5".

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689

 © 2023 Theo Gottwald * https://www.smart-package.com

- **VTT. - Value to Time** 760
- **TTV. - Time to Value** 751
- **VTD. - Value to Date** 756
- **DTC. - Date and Time Calculation** 693
- **VTF. - Verify Time Format** 742
- **VDF. - Verify Date Format** 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.5 TCA.GetPart

MiniRobotLanguage (MRL)

TCA.GetPart

Get a part of a Timestamp

Intention

Using `TCA.GetPart` you can easily just get a defined Part of the Timestamp.
This is similar to `TCA.Split`, but here you can define the Parts you want and also the Output Format.

For this you need to use the following Placeholder which will be replaced by the values during the process.

H:M:S.x

Hereby the
- H is Hours (0 to 23)
- M is the Minutes limited 59
- S is the Seconds limited 59
- x is the Milliseconds limited to 9999

The Format of the Milliseconds is internally rounded to 4 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.0023`
If you specify `00:00:00.023` this will be treated as `00:00:00.0230`

Sample: GetPart (Add Timestamps)

```
' Get Hour only
$$LAA=25:09:03.001
$$LAB=H
TCA.Get|$$LAA|$$LAB|$$LAC
DBP.$$LAC
' Result is here "01" due to the fact that the System will automatically detect
' Over 23:00 and then calculate the difference which is "01"

' Get Minutes only
$$LAA=03:19:23.01
$$LAB=M
TCA.Get|$$LAA|$$LAB|$$LAC
DBP.$$LAC
' Result is "19"

' Get Microseconds only
$$LAA=03:19:23.0199
$$LAB=X
TCA.Get|$$LAA|$$LAB|$$LAC
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.$$LAC
' Result is "019"

' Get Hours:Minutes.Microseconds with Overflow
$$LAA=25:79:03.001
$$LAB=H:M.x
TCA.Get|$$LAA|$$LAB|$$LAC
DBP.$$LAC
' Result is "01:19.001" due to Overflow calculations for the "25"->"01" and the

' Get Hours:Minutes.Microseconds, notice the Microseconds are filled to the Left
$$LAA=23:59:03.98
$$LAB=H:M.x
TCA.Get|$$LAA|$$LAB|$$LAC
DBP.$$LAC
' Result is "23:59.098"

' Get Hours:Minutes.Microseconds
$$LAA=08:09:59.098
$$LAB=H:M.X
TCA.Get|$$LAA|$$LAB|$$LAC
DBP.$$LAC
ENR.
' Result is "08:09.098"
```

Syntax

TCA.GetPart|P1|P2[|P3]
TCA.Get|P1|P2[|P3]

Parameter Explanation

P1 – Timestamp `like 08:39:40.123`
If Microseconds are not 3 digits, they will be filled with "0" from left. `side.`

P2 – Format of Output, `"H:M:S.x" -> Complete Timestamp`
`"H" -> Return only Hours`
`"S" -> Return only Seconds etc.`

P3 – ***(Optional)*** Variable for result. If missing, Result is stored in Variable P1.

Example

```
'*********************************************
' TCA.Get - Sample
'*********************************************
  ' Get Hours:Minutes.Microseconds
  $$LAA=08:09:59.098
  $$LAB=H:M.X
  TCA.Get|$$LAA|$$LAB|$$LAC
  DBP.$$LAC
  ' Result is "23:59:098"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.6 TCA.Make

MiniRobotLanguage (MRL)

TCA.Make - Command

Make 2 Timestamps

Intention

Using TCA.Make you can easily create a Timestamp in this Standard-Format:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.023` this will be treated as `00:00:00.023`

If the Makeition of any of the values goes beyond their limit, the Carry is Makeed to the next higher Parameter.

Sample: Make (Make Timestamps)

```
$$LAF=
$$LAB=14
$$LAC=40
$$LAD=21
$$LAE=999
$$MSG=Teste TCA.MAKE
TCA.Make|$$LAF|$$LAB|$$LAC|$$LAD|$$LAE
' Result is "14:40:21.999"
ENR.
```

Syntax

TCA.Make|P1|P2[|P3]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 – Variable for Result Timestamp.
P2 – Hours-Value
P3 – Minutes Value
P4 – Seconds-Value
P6 – ***(optional)*** Miliseconds Value

Example

```
'*******************************************
' TCA.MAKE - Sample with Rollover
' If the given values are too high, the command will
autoamtically
' Roll over to the next allowed Timestamp.
'*******************************************
'
$$LAF=
$$LAB=24
$$LAC=69
$$LAD=81
$$LAE=999
$$MSG=Teste TCA.MAKE
TCA.Make|$$LAF|$$LAB|$$LAC|$$LAD|$$LAE
$$ERG=01:10:21.999
JIS.$$LAF!$$ERG|Lab_failed
```

Remarks

-

Limitations:
-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.7 TCA.Overflow

MiniRobotLanguage (MRL)

TCA.Overflow - Command

Calculate Numbers Overflow

Intention

Using TCA.Overflow you can calculate an numeric Overflow.

Question:
What time is it in 81 Hours, if its 12:00 now?

```
' This is the Time (21:00) and the hours we want go into the future (81)
CAL.$$LAB=81+21
' This is the Overflow Number, its 23 for hours
$$LAC=23
$$MSG=Teste TCA.Ovl 2
TCA.Ovl|$$LAB|$$LAC|$$LAD
' Result is "10"
DBP.$$LAD
ENR.
```

Sample: Overflow (Overflow Timestamps)

```
' This is the current Minutes (:22) and the Minutes we
' want go into the future (281)
CAL.$$LAB=281+22
' This is the Overflow Number, its 59 for Minutes
$$LAC=59
TCA.Ovl|$$LAB|$$LAC|$$LAD
' Result is "08"
DBP.$$LAD
ENR.
```

Syntax

TCA.Overflow|P1|P2[|P3]

Parameter Explanation

P1 - Numeric value (Integer)
P2 - Overflow Value (Integer)

*© 2023 Theo Gottwald * https://www.smart-package.com*

P3 – Variable for Result. I omitted, the Result is placed on **TOS**.

Example

```
'*******************************************
' TCA.Overflow - Sample with Rollover
' If the given values are too high, the command will
autoamtically
' Roll over to the next allowed Timestamp.
'*******************************************
'
$$LAF=
$$LAB=24
$$LAC=69
$$LAD=81
$$LAE=999
$$MSG=Teste TCA.Overflow
TCA.Overflow|$$LAF|$$LAB|$$LAC|$$LAD|$$LAE
$$ERG=01:10:21.999
JIS.$$LAF!$$ERG|Lab_failed
```

Remarks

-

Limitations:

-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.7.8 TCA.Split

MiniRobotLanguage (MRL)

TCA.Split - Command

Split Timestamp into its Components

Intention

Using TCA.Split you can easily split a Timestamp in all Components:

HH:MM:SS.ddd

Hereby the
- **HH** is Hours (0 to 99 - not limited to 12 or 24)
- **MM** is the Minutes limited 59
- **SS** is the Seconds limited 59
- **ddd** is the Milliseconds limited to 999

The Format of the Milliseconds is internally rounded to 3 places by prefixing zeroes.
If you specify `00:00:23` this will be treated as `00:00:00.023`
If you specify `00:00:00.023` this will be treated as `00:00:00.023`

Sample: Split (Split Timestamps)

```
$$LAA=01:24:12.14
TCA.Split|$$LAA|$$LAB|$$LAC|$$LAD|$$LAE
DBP.$$LAB#$$LAC#$$LAD#$$LAE
' Result is "01#24#12#014"
ENR.
```

Syntax

TCA.Split|P1|P2|P3|P4|P5|P6
TCA.Spl|P1|P2|P3|P4|P5|P6

Parameter Explanation

P1 – Variable for original Timestamp
P2 – Variable for Hours-Value
P3 – Variable for Minutes Value
P4 – Variable for Seconds-Value

*© 2023 Theo Gottwald * https://www.smart-package.com*

P6 – Variable for Miliseconds Value

Any Parameter **P2** to **P6** can be omitted or replaced with a "-", in which case this value is been placed on **TOS**.

Example

```
'*********************************************
' TCA.Split - Sample
'*********************************************
'
$$LAA=01:24:12.14
TCA.Split|$$LAA|$$LAB|$$LAC|$$LAD|$$LAE
DBP.$$LAB#$$LAC#$$LAD#$$LAE
TCA.Make|$$LAF|$$LAB|$$LAC|$$LAD|$$LAE
DBP. Result is: $$LAF
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.8 VDF. - Verify Date Format

MiniRobotLanguage (MRL)

VDF. Command

Verify Date Format

Intention

What is a date?

A date is this: "01.01.2010". It can also be "1.1.10".
A US-formatted Date like "2022/12/31" will be converted to the European Date that can be used with these functions here.

What does VDF do?
VDF. will check that you have to do with a valid date. For example:
"39.13.2013" looks like it would be would be the right date format.
But its not a valid date.
There are no 39 days in a month. And you can't have a 13 th month in a year.
Year-numbers can go from 100 to 9999. Years below 100 get a 2000 added.
If you specify "01.01.13" it will become "01.01.2013".

Let us see what VDF. will do in such a cases.

```
$$TIM=23/45/12
VDF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
EIF.
' Timeout Flag: 23-12-2012
```

As you can see here, in case that it did any changes in the supplied date, it will also set the "Timeout Flag". This way you know that VDF. has changed something in the date-format.

This does not necessarily mean that your date was wrong. It just means "it was changed". See this example::

```
$$TIM=01/01/2012
VDF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

VDF. does not really check, if your actual month has exactly a 31th day. It will generally check the date-range. It will check that any month will not have more the 31 days, and that any year will have 1 to 12 month.

If you just write VDF. you will get this output:on TOS:

```
00-00-0000
```

Which is the mask for the date. You can use this date immediately in DTC.

Syntax

VDF.P1[|P2]

Parameter Explanation

P1 - Variable or number that should be verified.
If you specify an variable in P1, and P2 is omitted, then the result will be placed in that variable P1.

P2 - (optional) If specified, this variable will be used for the resulting value.

Example

```
'*********************************************
' VDF. - Sample
'*********************************************
'
$$TIM=23/45/12
VDF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
EIF.
' Timeout Flag: 23-12-2012
ENR.
```

Remarks

Year-numbers can go from 100 to 9999. Years below 100 get a 2000 added.
If you specify "01.01.13" it will become "01.01.2013".

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTF. - Verify Time Format 742
- TTV. - Time to Value 751
- DTV. - Date to Value 746
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- IDC. / NDC. - If Date Command 709
- FMT. - Format Numeric Output 3107
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.9 VTF. - Verify Time Format

MiniRobotLanguage (MRL)

VTF. Command

Verify Time Format

Intention

What is a time?
A time is this: "08:37". A time is also this ""08:37:15", which includes seconds. While this "08:37:15:2544" is also a valid time and include Milliseconds.

What does VTF do?
VTF. will check that you have to do with valid times. For example:
"29:65:99" looks like it would be would be the right time format.
But its not. There are no 29 hours on each day. And you can't have 99 seconds in a minute.

Let us see what VTF. will do in such a case.

```
$$TIM=25:65:-2
VTF.$$TIM
DBV.$$TIM
ENR.
' Output is 23:29:02
```

It will do a little bit more then that. In case that it did any changes in the supplied date, it will also set the "Timeout Flag". This way you know that VTF. changed something in the time-format. This does not necessarily mean that your time was wrong. It just means "it was changed". See here:

```
$$TIM=25:65:-2
VTF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
EIF.
ENR.
' Timeout-Flag was set
' Result is 23:59:02
```

Here the timeout flag will not be set:

```
$$TIM=23:59:00
VTF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
```

And the following code will set the timeout flag, because the time was changed.

```
$$TIM=23:59
VTF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
EIF.
```

Here is another example, what its good for:

```
$$TIM=23/45/00
VTF.$$TIM
' Result is 23:45:00
```

Syntax

VTF.P1[|P2]

Parameter Explanation

P1 - Variable or number that should be verified.
If you specify an variable in P1, and P2 is omitted, then the result will be placed in that variable P1.

P2 - ***(optional)*** If specified, this variable will be used for the resulting value.

Example

```
'*********************************************
' VTF. - Sample
'*********************************************
'
$$TIM=23:59
VTF.$$TIM
ITO.
  DBV.Timeout Flag: $$TIM
ELS.
  DBV.$$TIM
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

These time-calculations depend on the current system time. If your system time is wrongly set, the command may also deliver the wrong time.

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VDF. - Verify Date Format 739
- TTV. - Time to Value 751
- DTV. - Date to Value 746
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- IDC. / NDC. - If Date Command 709
- FMT. - Format Numeric Output 3107
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.10 Time to Value Commands

Using these Commands, you can convert Date and Time to numbers and do simple Mathematics with these numbers.

Using `DTV.` you can convert a date to the equivalent number of days since January 1601.
Now you can do any sort of math using this number. When you're done, just use VTD. to convert the result back into a valid date! This way calculations with days get very easy.

`TTV.` delivers you the "**number of seconds since midnight**" for a specified time. It will tell you "How many seconds old your day is. You can also do Mathematics using this number and convert it back.

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.47.10.1 DTV. - Date to Value

MiniRobotLanguage (MRL)

DTV. Command

Date to Value

Intention

Using DTV. you can convert a date to the equivalent number of days since January 1601.
Now you can do any sort of math using this number. When you're done, just use VTD. to convert the result back into a valid date! This way calculations with days get very easy.

Example:

```
VTD.1|$$RES
MBX.$$RES
' Will return "01-91-1601"

DTV.01.01.1601|$$RES
MBX.$$RES
' Will return "1"

' Using "Now" the current Date is used.
DTV.now|$$RES
VTD.$$RES|$$RES
MBX.$$RES
```

What is the difference between date and value?
A date is this: "01.01.2010".
A value is this: "149386". This the number of days that have passed, since January 1601.

What does DTV. do?
DTV. delivers you the value for a date. That is the number of the given day, counted from January 1601, one for each day.
For example, you will get a "149385" as value for the "01.01.2010" and a "149386" as value for the "02.01.2010", which is one day later.

What do i do with these values?
Values that represent a date, are more easy to handle. Store them in files, do calculations with them. A date is just a number with DTV. Of course you can also add or subtract days from this value.

Can i convert that number back to a date?
Yes, finally you can convert that value back to a valid date using VTD. (Value to Date).

© 2023 Theo Gottwald * https://www.smart-package.com

One thing to know is, that if you specify the term "now" instead of a date, then the current date will be used. The same will happen, if you omit **P1** or if you leave **P1** empty.

Usage is simple:

```
DTV.
STS.DUMP
MBX.!
```

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt      :150169

------------------------------------
```

which is today's value on TOS.

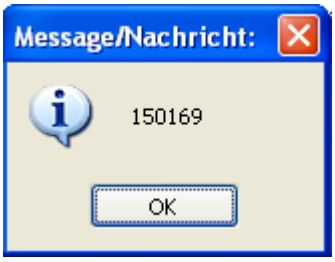
If we provide **P1**, the variable will be used as Source and Target, if its a variable.
Like this:

```
$$TIM=now
DTV.$$TIM
MBX.$$TIM
```

will deliver this result:below (left picture). If you leave away the first line and just write:

```
DTV.$$TIM
MBX.$$TIM
```

you will get a zero, like in the second picture. Why this?



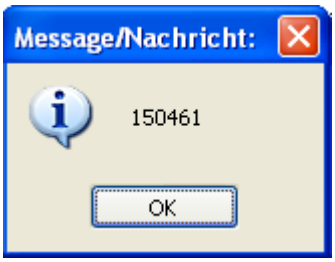
The reason is, that undefined variables will not be resolved. In that case the script tries to get the "date-value" for `$$TIM` which is zero, like it is zero for any invalid (non-date) string..

```
' This will work as expected and return the value for "Now"
$$TIM=
DTV.|$$TIM
MBX.$$TIM

' This will do just the same (P1 = empty)
DTV.|$$TIM
MBX.$$TIM
```

Now we will get a value for another date, let us take the `12.12.2012`.
Like this:

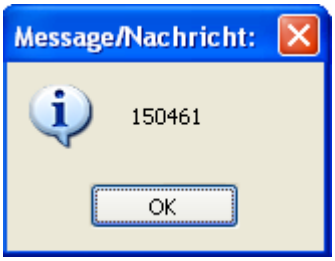
© 2023 Theo Gottwald * https://www.smart-package.com

```
 DTV.12.12.2012|$$TIM
MBX.$$TIM
```



Of course we can use variables. Here is an example:

```
$$DAT=12.12.2012
DTV.$$DAT|$$TIM
MBX.$$TIM
```



It works. The result is in the second variable `$$TIM`.

Important:
You need to specify the date in the format "`dd.mm.yyyy`". Giving a useless input you will get:
a) If there are no numbers contained, Result is "0"
b)If there are numbers contained, you will get a date based on the 01.01.2001 as result. For example:

```
$$DAT=Anton
DTV.$$DAT|$$TIM
MBX.$$TIM
'Result is "0"

$$DAT=Anton6
DTV.$$DAT|$$TIM
MBX.$$TIM
'Result is "06.01.2001"
```

Syntax

DTV.[P1][|P2]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - ***(can be omitted or empty)*** If omitted or empty, the actual time (Now) is used. The result will be placed on the TOS. The same will happen, if the term now is given, no matter if in an variable or direct. If you specify an variable in P1, and P2 is omitted, then the result will be placed in that variable.

P2 - ***(optional)*** If specified, this variable will be used for the resulting value.

Example

```
'*********************************************
' DTV. - Sample
'*********************************************
'
$$TIA=01.01.2010
$$TIB=02.01.2010
DTV.$$TIA
DTV.$$TIB
CAL.$$RES=($$TIB-$$TIA)
DBP.$$RES, $$TIB,$$TIA
ENR.

'*********************************************
' DTV. - Sample
'*********************************************
'
$$DAT=12.12.2012
DTV.$$DAT|$$TIM
MBX.$$TIM

'*********************************************
' DTV. - Sample
' Convert Date to Number, Add 1 day and convert
' Number back to Date
'*********************************************
'
$$DAT=12.12.2012
DTV.$$DAT|$$TIM
VIC.$$TIM
VTD.$$TIM
MBX.$$TIM
' Result is 13.12.2012
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

All Date-calculations should be accurate up to a Millisecond at least back until 1601.
You can not specify Dates before the 01-01-1601. Doing so will always return "0".

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.10.2 TTV. - Time to Value

MiniRobotLanguage (MRL)

TTV. Command

Time to Value

Intention

What is the difference between time and value?
A time is this: "08:37". A time is also this ""08:37:15", which includes seconds. While this "08:37:15:2544" is also a valid time, we don't need Milliseconds here because TTV. does only work down to seconds. A value is this: "31006" which is about "07:36".

What does TTV. do?
TTV. delivers you the "**number of seconds since midnight**" for a specified time. It will tell you "How many seconds old your day is.

For example, you will get a "31151" as value for the "07:39:" and a "149386" as value for the "02.01.2010", which is one day later.

What do i do with these values?
Values that represent a time, are more easy to handle. Store them in files, do calculations with them. A time is just a number with DTV. Of course you can also add or subtract seconds to / from this value.

Can i convert that number back to a time?
Yes, finally you can convert that value back to a valid date using VTT. (Value to Time).

One thing to know is, that if you specify the term "**now**" instead of a date, then the current time will be used. The same will happen, if you omit **P1** or if you leave **P1** empty.

Usage is simple:

```
TTV.
STS.DUMP
MBX.!
```

```
-------------------------------------
User-Stack-Dump:
-------------------------------------

Stackpos: 000 ->LongInt      :30922

-------------------------------------
```

which is actual time (in seconds after midnight) as a value on TOS.

If we provide **P1**, the variable will be used as Source and Target, if its a variable.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Like this:

```
$$TIM=now
TTV.$$TIM
MBX.$$TIM
```

will deliver this result:below (left picture). If you leave away the first line and just write:

```
TTV.$$TIM
MBX.$$TIM
```

you will get a zero, like in the second picture. Why this?

The reason is, that undefined variables will not be resolved. In that case the script tries to get the "date-value" for `$$TIM` which is zero, like it is zero for any invalid (non-time) string..

```
' This will work as expected and return the value for "Now"
$$TIM=
TTV.|$$TIM
MBX.$$TIM

' This will do just the same (P1 = empty)
TTV.|$$TIM
MBX.$$TIM
```

Now we will get a value for another time. But what if we have times without the seconds?
Let us try:

```
' We specify only the hours
TTV.10|$$TIM
DBV.$$TIM
' Result is 36000

' We specify Hours and Minutes
TTV.10:00|$$TIM
DBV.$$TIM
' Result is 36000

' We specify the complete time
TTV.10:00:00|$$TIM
DBV.$$TIM
' Result is 36000

' We specify even Milliseconds
TTV.10:00:00:0000|$$TIM
DBV.$$TIM
```


© 2023 Theo Gottwald * https://www.smart-package.com

As you can see in this example, the result is always the same. Missing time-informations are being treated as zero. If you omit minutes or seconds, they are zero'ed.
The Milliseconds do not play a role here and are simply ignored.

Of course we can use variables. Here is an example:

```
$$TIA=10
TTV.$$TIA|$$TIM
DBV.$$TIM
```

It works. The result is 36000 as expected.

```
' Get the time-value into variable $$TIA
$$TIA=10:00:30
TTV.$$TIA|$$TIM

' Subtract 156 Seconds
CAL.$$TIM=$$TIM-156
VTT.$$TIM|$$TIA

' Print it in the editor
DBV.$$TIA
ENR.
```

Important:
You do need to specify the time in the format "`hh:mm:ss`", but you can leave parts away from the right side. You can not leave away the **hh**-part.

Syntax

TTV.[P1][|P2]

Parameter Explanation

P1 - ***(can be omitted or empty)*** If omitted or empty, the actual time (Now) is used. The result will be placed on the TOS. The same will happen, if the term now is given, no matter if in an variable or direct. If you specify an variable in P1, and P2 is omitted, then the result will be placed in that variable.

P2 - ***(optional)*** If specified, this variable will be used for the resulting value.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*********************************************
' TTV. - Sample
'*********************************************
'
TTV.
DBV.$$000
PAU.1
TTV.
VTT.$$000
DBV.$$000
ENR.


'*********************************************
' TTV. - Sample
'*********************************************
'
TTV.now|$$TIM
DBV.$$TIM
VTT.$$TIM
DBV.$$TIM
ENR.


'*********************************************
' TTV. - Sample
'*********************************************
'
TTV.now
VTT.$$000
DBV.$$000
ENR.

'*********************************************
' TTV. - Sample
'*********************************************
'
TTV.10:00:00|$$TIM
DBV.$$TIM
' Result is 36000
```

Remarks

-

Limitations:

These time-calculations depend on th current system time. If your system time is wrongly set, the command may also deliver the wrong time.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- TTV. - Time to Value 751
- DTV. - Date to Value 746
- VTD. - Value to Date 756
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.10.3 VTD. - Value to Date

MiniRobotLanguage (MRL)

VTD. Command

Value to Date

Intention

Using `DTV.` you can convert a date to the equivalent number of days since January 1601.
Now you can do any sort of math using this number. When you're done, just use `VTD.` to convert the result back into a valid date!
This way calculations with days get as easy as adding or subtracting numbers.

Example:

```
VTD.1|$$RES
MBX.$$RES
' Will return "01-91-1601"

DTV.01.01.1601|$$RES
MBX.$$RES
' Will return "1"
```

What is the difference between a date and a value?
A date is this: `"01.01.2010"`.
A value is this: `"149386"`.

What does VTD. do?
`VTD.` is the counterpart to `DTV.` It will convert a value back to a date.
`DTV.` delivers you the value for a date. That is the number of the given day, counted from January 1601, one for each day.

For example, you will get a `"149385"` as value for the `"01.01.2010"` and a `"149386"` as value for the `"02.01.2010"`, which is one day later.

What do i do with these values?
Values that represent a date, are more easy to handle. Store them in files, do calculations with them. A date is just a number with `DTV.` Of course you can also add or subtract days from this value.

Can i convert that number back to a date?
Yes, finally you can convert that value back to a valid date using `VTD.` (Value to Date).

See DTV. - Date to Value 746 on how to convert a date into a corresponding number.

Once you have such a number, `VTD.` will convert it back to a date. If you want to change the format of the given date, use `DTC.`

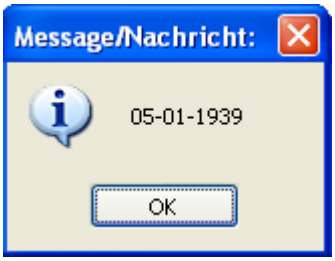
*© 2023 Theo Gottwald * https://www.smart-package.com*

Here is an example, how
- a date is been converted into a number
- one day is added
- and then the number is converted back to the date..

```
$$DAT=12.12.2012
DTV.$$DAT|$$TIM
VIC.$$TIM
VTD.$$TIM
MBX.$$TIM
ENR.

' This example uses a given date-number "123456!
$$TIM=123456
VTD.$$TIM
MBX.$$TIM
```

Which will evaluate to the below result (picture below).
Of course you can also use 2 variables, in that case you get the same result.

```
$$TIM=123456
VTD.$$TIM|$$RES
MBX.$$RES
ENR.
```



Important:
You can use this date immediately for DTC. to convert it to any format you like.
Like this:

```
$$TIM=123456
VTD.$$TIM|$$RES
' change the date-format to anything you want.
DTC.$$RES|mm/dd/yy
MBX.$$RES
ENR.
```

Syntax

VTD.P1[|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - Variable or number that should be converted to a date.
If you specify an variable in P1, and P2 is omitted, then the result will be placed in that variable P1.

P2 - (optional) If specified, this variable will be used for the resulting value.

Example

```
'*********************************************
' VTD. - Sample
'*********************************************
'
$$TIM=123456
VTD.$$TIM|$$RES
DTC.$$RES|dd+mm+yy
MBX.$$RES
ENR.
```

Remarks

-

Limitations:

All Date-calculations should be accurate up to a Millisecond back until 1601.
You can not specify negative values, therefore earlier dates then 1601 can not be calculated using `VTD./DTV.`.

See also:

- 1.5.1.3. Using Variables 92
- DAT. - Date and Time Information (System independent) 689
- VTT. - Value to Time 760
- TTV. - Time to Value 751
- DTV. - Date to Value 746
- DTC. - Date and Time Calculation 693
- VTF. - Verify Time Format 742
- VDF. - Verify Date Format 739
- IDC. / NDC. - If Date Command 709
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.47.10.4 VTT. - Value to Time

MiniRobotLanguage (MRL)

VTT. Command

Value to Time

Intention

What is the difference between time and value?
A time is this: "08:37". A time is also this ""08:37:15", which includes seconds. While this "08:37:15:2544" is also a valid time, we don't need Milliseconds here because TTV. does only work down to seconds. A value is this: "31006" which is about "07:36".

What does VTT. do?
VTT. is the counterpart to TTV. It will convert a value back to a Time.
TTV. delivers you the "**number of seconds since midnight**" for a specified time. It will tell you "How many seconds old your day is. And VTT. takes that number and converts it back.
For example, you will get a "36000" as value for the 10:00:00" and a "36001" as value for the "10:00:01", which is one second later.

What do i do with these values?
Values that represent a time, are more easy to handle. Store them in files, do calculations with them. A time is just a number of seconds with TTV. Of course you can also add or subtract seconds from this value. Here is an example:

```
' Get the time-value into variable $$TIA
$$TIA=10:00:30
TTV.$$TIA|$$TIM

' Subtract 30 Seconds
CAL.$$TIM=$$TIM-30
VTT.$$TIM|$$TIA

' Print it in the editor
DBV.$$TIA
ENR.
```

See TTV. - Time to Value 751 for additional samples and explanations on this topic.
See DTC. - Date and Time Calculation 693 for formatting and calculation with times and dates.
See also FMT. - Format Numeric Output 3107 for formatted numeric output.

Important:
You can use this time immediately for DTC. to convert it to any format you like.
Like this:

```
$$TIM=36000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VTT.$$TIM|$$RES
' change the date-format to anything you want.
DTC.$$RES|hh/nn
DBV.$$RES
ENR.
' Output is 10/00
```

Syntax

VTT.P1[|P2]

Parameter Explanation

P1 - Variable or number that should be converted to a time.
If you specify an variable in P1, and P2 is omitted, then the result will be placed in that variable P1.

P2 - (optional) If specified, this variable will be used for the resulting value.

Example

```
'*********************************************
' VTT. - Sample
'*********************************************
'
$$TIA=10:00:30
TTV.$$TIA|$$TIM
' Subtract 30 Seconds
CAL.$$TIM=$$TIM-30
VTT.$$TIM|$$TIA
DBV.$$TIA
ENR.
```

Remarks

-

Limitations:

These time-calculations depend on th current system time. If your system time is wrongly set, the command may also deliver the wrong time.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- **1.5.1.3. Using Variables** 92
- **DAT. - Date and Time Information (System independent)** 689
- **TTV. - Time to Value** 751
- **DTV. - Date to Value** 746
- **VTD. - Value to Date** 756
- **DTC. - Date and Time Calculation** 693
- **VTF. - Verify Time Format** 742
- **VDF. - Verify Date Format** 739
- IDC. / NDC. - If Date Command 709
- FMT. - Format Numeric Output 3107
- CAL. - mathematical CALculation 2315
- CAX. - Calculate Extended 2320

© 2023 Theo Gottwald * https://www.smart-package.com

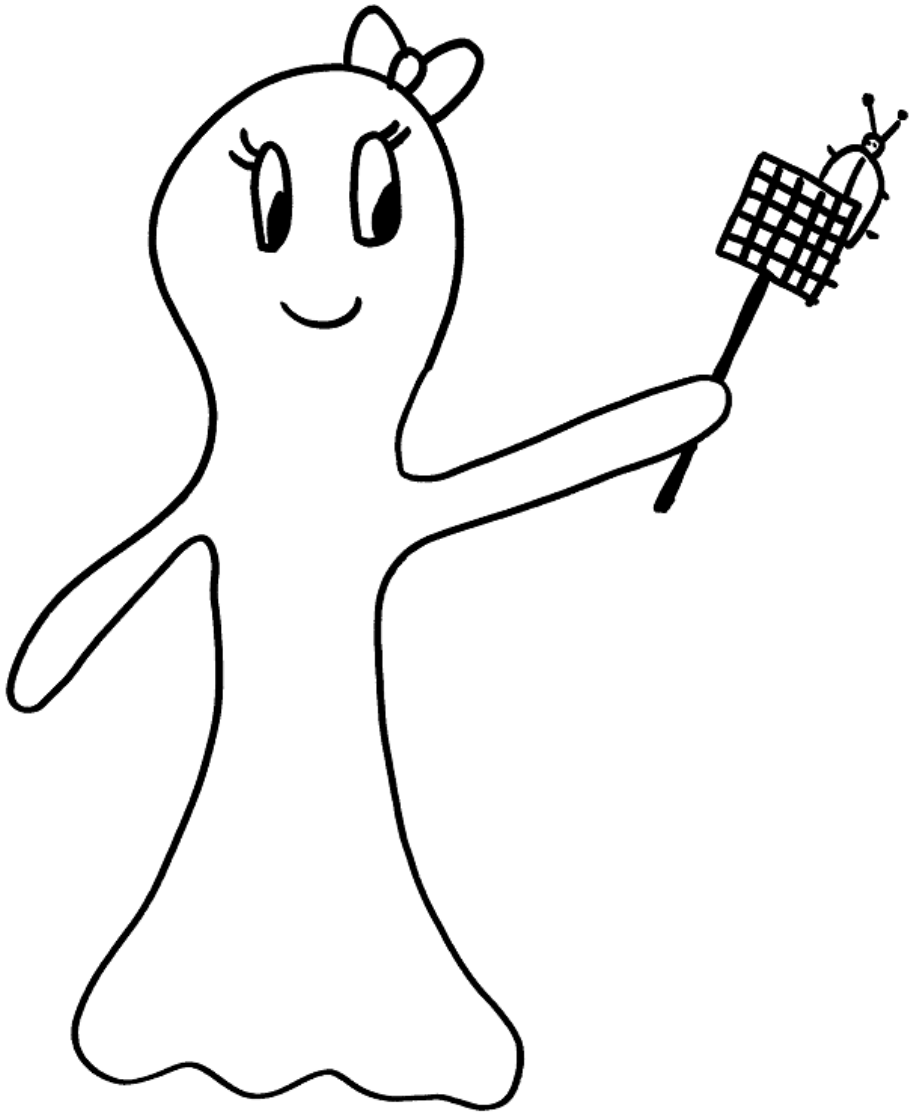
3.48 Debugging - Commands

3.48.1 ! Smarty's Debugging Commands

Smarty's Debugging Commands **Previous** 760 **Top** 41 **Next** 765

Smarty's Debugging Commands

These Commands are helpful not only in Debugging, they can also be used to slow down Script Execution whenever the target GUI can not handle the Robots Speed. And thats often the case.



Your script does not do what you want?

*© 2023 Theo Gottwald * https://www.smart-package.com*

It does not work as expected?

In this case "debugging" is your next step.

In this chapter you find all the commands that you need to trace your script and find out what's wrong.

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.2 DBM. - DeBugMode

MiniRobotLanguage (MRL)

DBM. Command

Debug Mode

Intention

The `DBM.`-Command enables different options for Debugging.

`DBM.1` opens a separate Window with Single-Step option.

`DBM.2` prints the actual Command into a Debug-Console-Window.

`DBM.4` prints all Browser-Events that come from the Mini-Browser, if it was launched using `LBR.`

`DBM.32` enables additional Infos.

Syntax

DBM.P1

Parameter Explanation

```
P1 -  0 = Mode off
      1 = Mode on, messages in debug-window of the editor
      2 = Mode on, messages in Concole-window
      4 = Show Browser Events, while loading a WEB-Site
           MiniBrowser only, and must be started using LBR.
     32 = Mode on, Additional infos mostly for internal use.

Debugmode 32 prints every executed Line in the Debug-Window
together with following informations:

001   000000782162   0001 DBM.1

1. Linenumber
2. used CPU-Time
3. Times the Line was executed
4. Command
```

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' DBM.-Demo
'**********************************
DBM.2
PRT.This is a text to display something
:over
MBX.Halt to read output
'**********************************
```

Remarks

With the `USM.` Command you can redirect the ouput of `DBM.`-Mode2 into an separate Log-file.

Limitations:

-

See also:

- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- DBO. - Debug Raw Output 767
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.3 DBO. - Debug Raw Output

MiniRobotLanguage (MRL)

DBO. Command

Debug Raw Output

Intention

This command is used to print debug variables in the debug-window of the editor.
Unlike DBP - Debug Print 769, specialfolders 345 and systemvariables 334 will not be resolved.
Unlike DBV. - Debug Print Variable 771 variables will also not be resolved.
The output will be completely and exactly what you write.

`DBO.` will be ignored, if the script is not running from within the editor. This way, Debug-Messages will not be see from an compiled executable.

```
VAB.$$NAM=Rainer
VAB.$$TXT=$$NAM
DBO.$$TXT
ENR.
' Output is "$$TXT"
```

Syntax

DBO.P1

Parameter Explanation

P1 - Any text or variables you want to print in the Editor-Debug-Window.

Example

```
'*****************************************************
' EXAMPLE : DBO.
' Run from the Editor!
'*****************************************************
: $$TXT=?path\Logfile.log $$TXT
DBO. My variable is: $$TXT
ENR.
MBX.Ready
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'******************************************************
```

Remarks

-

Limitations:

-

See also:

- **DBP - Debug Print** 769
- **DBO. - Debug Raw Output** 767
- DBM. - DeBugMode 765
- SAY. - Say something 2048
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.4 DBP. - Debug Print

MiniRobotLanguage (MRL)

DBP. Command

Debug Print

Intention

This command is used to print debug information in the debug-window of the editor. `DBP.` will be ignored, if the script is not running from within the editor. This way, Debug-Messages will not be see from an compiled executable.

```
DBP. Hallo World!
```

`DBP.` will resolve systemvariables and specialfolders, which could be an unwanted feature for debugging. If you do not want that, use DBV. - Debug Print Variable 771.

Syntax

DBP.P1

Parameter Explanation

P1 - `Any text or variables you want to print in the Editor-Debug-Window.`

Example

```
'*****************************************************
' EXAMPLE : DBP.
' Run from the Editor!
'*****************************************************
DBP.Hallo
DBP.ENDE
:over
MBX.Halt to read output
'**********************************
```

Remarks

-

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Limitations:

-

See also:

- **DBV. - Debug Print Variable** 771
- **DBO. - Debug Raw Output** 767
- DBM. - DeBugMode 765
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.5 DBV. - Debug Print Variable

MiniRobotLanguage (MRL)

DBV. Command

Debug Print Variable

Intention

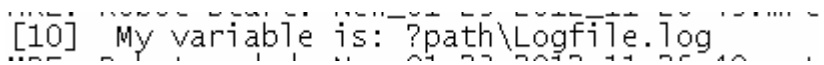
This command is used to print debug variables in the debug-window of the editor. Unlike DBP - Debug Print 769, specialfolders 345 and systemvariables 334 will not be resolved.

`DBV.` will be ignored, if the script is not running from within the editor. This way, Debug-Messages will not be see from an compiled executable.

Here is a good example what `DBV.` is useful for. You want to know what is the content of this variable. But the content should not be resolved, just be displayed. This is where `DBV.` takes place.

```
VAB.$$NAM=Rainer
VAB.$$TXT=$$NAM
DBV.$$TXT
ENR.
' Output is "$$NAM"

' Special folders are not resolved
: $$TXT=?path\Logfile.log
DBV. My variable is: $$TXT
```

You will get this output:



Syntax

DBV.P1

Parameter Explanation

P1 - Any text or variables you want to print in the Editor-Debug-Window.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*****************************************************
' EXAMPLE : DBV.
' Run from the Editor!
'*****************************************************
: $$TXT=?path\Logfile.log $$TXT
DBV. My variable is: $$TXT
ENR.
MBX.Ready
'*****************************************************
```

Remarks

-

Limitations:

-

See also:

- **DBP - Debug Print** 769
- DBM. - DeBugMode 765
- SAY. - Say something 2048
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.6 DIP. - Delayed Instruction Processing

MiniRobotLanguage (MRL)

DIP. Command

Delayed Instruction Processing

Intention

These command is used to slow down script execution. The robot will pause after each line for the given amount of time in milliseconds (ms).

`DIP.200` is a common used value.

You can switch off the slow-down with:

```
DIP.0
```

or simply

```
DIP.
```

These will make the script run with full speed. DIP. is used to slow down the whole script or parts of a script. If you experience problems running a script as executable (compiled script), while the script has worked fine from the Editor, try a DIP.250.

Generally we recommend to design scripts in a way that they run at full speed. However for debugging use, DIP. is a good choice.

Syntax

DIP.[number]

Parameter Explanation

P1 - (optional) `NUM/VAR`

```
   the given number is the amount of time in msec
(Milliseconds =
   1/1000 th of a second), that the Script waits between the
commands.
   This way the script execution is slowed down.
```

Example

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'************************************************
' DIP. - Sample
'************************************************

'Switch on Slow Mode
DIP.450

' Switch off Slow Mode
DIP.


'************************************************
```

Remarks

-

Limitations:

-

See also:

- DBM. - DeBugMode 765
- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- DMP. - Dump internat Informations 775
- USM - Use silent Mode 786
- **OPT. - Optional Settings** 2406
- PAU. - Pause 783

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.7 DMP. - Dump internat Informations

MiniRobotLanguage (MRL)

DMP. Command

Dump internal informations

Intention

This command makes internal structures like the stack or the speed-table visible to the user.
You can see all variables 366, the stack 369. Or you can get the information on how fast single lines have been processed (speed table 80).

1. DMP.1 - Show Speed-Table

Using `DMP.1` to get a "Speed Table", you can get a lot of **information for debugging**.

In the very left column, you can see the line-number.
In the second column, you can see the Number of "Ticks" that this line needed to execute. My current i7 CPU can make 16.000.000 Ticks in one seconds. On your computer this may differ.
The third row shows you how often this line has been executed.
You will also see all Macros expanded in this DMP.1-View.
Also you see here that the SPR will jump over a lot of Comment-Lins, thats why they did not use any time. Therefore the (automatic) Jump-Instruction in Line 001 needs 130 Ticks.

```
-----------------------------------
Speed-Dump:
-----------------------------------
001 0000000000000130 0001
002 0000000000000000 0000
003 0000000000000000 0000
004 0000000000000000 0000
005 0000000000000000 0000
006 0000000000000000 0000
007 0000000000000000 0000
008 0000000000000000 0000
009 0000000000000000 0000
010 0000000000104023 0001 STW.ct|Chrome_WidgetWin_1|- Brave
011 0000000000549567 0001 ANA.Snap|0
012 0000000001272838 0001 ANA.Save|0|?path\Before.png
013 0000000050642061 0001 ANA.bw|0|460
014 0000000000482603 0001 ANA.Save|0|?path\After.png
015 0000000000000000 0001 DMP.1
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

DMP.[P1][|P2|P3]

Parameter Explanation

P1 - Select which table or stack you want to see.
  1 or "Speed"   - Speed-Table
  2 or "Global Var"   - Global Variables Table
  3 or "Local Var"   - Local Variables Table
  4 or "Global Stack" - Global User-Stack
  5 or "Local Stack"   - Local User-Stack
  6 or "Current Stack" - dump actual selected stack

P2 - (optional) Lower value of the displayed range

P3 - (optional) Upper value of the displayed range

If no argument is given the whole table is shown.

Important: Currently the Stack-Dump will NOT Show Floating-Point Variables
correctly due to the used Formatting. To check these, use:

```
POP.$$TST
MBX.$$TST
```

Example

```
'***********************************
' it will show Speedtable entry 4 to 6
DMP.1|4|6
' or
DMP.Speed|4|6


:over
MBX.Halt to read output
'***********************************

'***********************************
' This example shows the Speed-Dump of the script
' Put this command at the end of your script.
DMP.
:over
MBX.Halt to read output
'***********************************
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

-

See also:

- GRI. - Get Run Information 778
- 1.5. Features and Hints 80
- 2.4 The global and local Stack 369
- 2.3 Variables, global and local 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.8 GRI. - Get Run Information

MiniRobotLanguage (MRL)

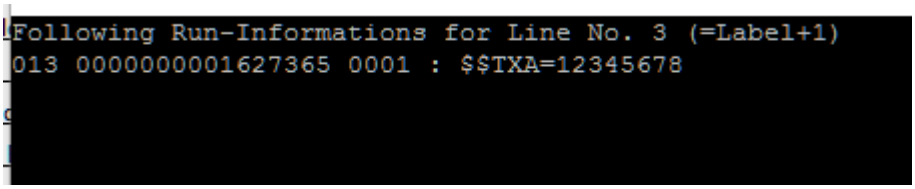
GRI. Command

Get Run Information

Intention

Get internal Information on how long a single line needed to be processed by the CPU.

```
' GRI - Example 1 using Label-Reference
:Label
: $$TXA=12345678
GRI.:Label|1
PRT.Following Run-Informations for Line No. 3 (=Label+1)
PRT.$$000
MBX.!
END.

' This Example uses a Variable for the GRI.-Result
:Label
: $$TXA=12345678
GRI.:Label|1|$$RES
PRT.Following Run-Informations for Line No. 3 (=Label+1)
PRT.$$RES
MBX.!
END.
```



Syntax

GRI.[P1][|P2][|P3]

Parameter Explanation

P1 - `Line-Number or Label` (can be inside a variable, too).
P2 - ***(optional)*** `NUM` - Increment, is added to the Line-Number/Label **P1**.
P3 - ***(optional)*** ***VAR*** - Variable for Result, if omitted the Result goes to the TOS.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 or **P2** can be inside a variable.

Example

```
'-----------------------------------------------------------
' GRI - Example
'-----------------------------------------------------------
: $$TXA=12345678
: $$TXB=2
VAR.$$TXA=12345678
VAR.$$TXA=12345678
VAR.$$TXB=2
GRI.4
PRT.The following Line shows Run-Informations for Line No. 4
PRT.$$000
GRI.5
PRT.The following Line shows Run-Informations for Line No. 5
PRT.$$000
GRI.6
PRT.The following Line shows Run-Informations for Line No. 6
PRT.$$000
END.

'-----------------------------------------------------------
' GRI - Example 2 using Label-Reference
'-----------------------------------------------------------
:Label
PAU.0.5
: $$TXA=12345678
GRI.:Label|1|$$RES
PRT.Following Run-Informations for Line No. 3 (=Label+1)
SBD.$$RES|$sp$|2|$$TIA
CAL.$$TIM=$$TIA/1656527/10/3
PRT.$$TIM
MBX.!
END.

#####################################################

' Macro - Converts GRI-Result into Seconds
:Label
PAU.0.5
: $$TXA=12345678
GRI.:Label|1|$$RES
PRT.Following Run-Informations for Line No. 3 (=Label+1)
%GRI_to_Time $$RES|$$TIX
PRT.$$TIX
MBX.!
ENR.
 '-----------------------------------------------------------
' Here is the Needed Macro-Definition
 '-----------------------------------------------------------
 : %GRI_to_Time 1
 #IF PARAMS=0
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.Need 1 Parameter here!
END.
#EIF
#IF PARAMS=1
GSB.GRI_to_Time|§§§01
#EIF
#IF PARAMS=2
GSB.GRI_to_Time|§§§01|§§§02
#EIF
#IF PARAMS>2
MBX.Need 1 Parameter here!
END.
#EIF
END%
'-------------------------------------------------------------
' Here is the Needed Subroutine
'-------------------------------------------------------------
' Wandelt die Rückgabe von GRI in eine Zeit um
:GRI_to_Time
SAV.Save|$$P01|$$P02|$$P03
VAR.$$P01=§§_01
SBD.$$P01|$sp$|2|$$P02
CAL.$$P03=$$P02/1656527/10
IVV.§§_00>1
  VAI.§§_02=$$P03
ELS.
  PUV.$$P03
EIF.
SAV.Restore
RET.

#########################################################

' GRI - Example 4 Macro Nesting Sample

:Label
PAU.0.5
: $$TXA=12345678
%T_GetLinetime Label|$$RES
PRT.Following Run-Informations for Line No. 3 (=Label+1)
PRT.$$RES
MBX.!
END.

ENR.
'=============================================================
' P1 - Label - Zeile darunter wird gemessen
' P2 - Variable for Result
'
: %T_GetLinetime 2
#IF PARAMS=2
GSB.T_GetLinetime|§§§01|§§§02
#ELSE
MBX.Need 2 Params here!
#EIF
END%
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:T_GetLinetime
SAV.Save|$$P04|$$P05
VAO.$$P05=
GRI.§§_01|1|$$P04
%GRI_to_Time $$P04|§§_02
SAV.Restore
RET.

'-------------------------------------------------------------
' Here is the Needed Macro-Definition
'-------------------------------------------------------------
: %GRI_to_Time 1
#IF PARAMS=0
MBX.Need 1 Parameter here!
END.
#EIF
#IF PARAMS=1
GSB.GRI_to_Time|§§§01
#EIF
#IF PARAMS=2
GSB.GRI_to_Time|§§§01|§§§02
#EIF
#IF PARAMS>2
MBX.Need 1 Parameter here!
END.
#EIF
END%
'-------------------------------------------------------------
' Here is the Needed Subroutine
'-------------------------------------------------------------
' Wandelt die Rückgabe von GRI in eine Zeit um
:GRI_to_Time
SAV.Save|$$P01|$$P02|$$P03
VAR.$$P01=§§_01
SBD.$$P01|$sp$|2|$$P02
CAL.$$P03=$$P02/1656527/10
IVV.§§_00>1
  VAI.§§_02=$$P03
ELS.
  PUV.$$P03
EIF.
SAV.Restore
RET.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- DMP. - Dump internat Informations 775
- DBV. - Debug Print Variable 771

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.9 PAU. - Pause

MiniRobotLanguage (MRL)

PAU. Command

Pause

Intention

Pauses Script execution for a given amount of time. Give CPU-time back to the windows system. Possible use cases are:

- At some places the script may execute too fast, then you may want to slow it down.
- if you make a so called "endless loop", it makes sense to give some computing power back to the system.

In both cases you can use the `PAU.` Instruction. Usage is as easy as possible. Just write:

```
' pause for 1 second
PAU.1
```

When the script comes to that place it will just wait for 1 second. Behind the curtains, it will leave the CPU to other processes for the given amount of time.

1 second can be a lot of time, but its up to you, how long you want the script to **PAU.**
You can make it wait for 5 hr like this:

```
PAU.5|hr
```

or even let it wait for a Year (in seconds):

```
'  This is 6 hrs in milliseconds
PAU.21600000|ms
' this is also 6 hrs., in minutes
PAU.360|min
' and this is 6 hrs. in secons
PAU.21600|s
```

but normally you would write it like this:

```
PAU. 6|hr
```

You can use smaller times, starting up with 1 ms.

```
' This will wait for a tenth of a second
PAU.0.01
' this is the same
PAU.100|ms
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

You can choose values down to 1 ms, there are no half or zero milliseconds.

Syntax

PAU.[P1][|P2]

Parameter Explanation

P1 - Time to pause

P2 - ***(optional)*** Time unit (default is seconds)
Possible values are:

- hr = hour
- min = minute
- sec = seconds (default when P2 is missing)
- ms = milliseconds

Example

```
'****************************************
' Sample: PAU
' AWT at 2011-07-30
'****************************************

' pause in length of 1 second
PAU.1

' pause in lenght of half second
PAU. 0.5

'pause in length of 1 hour
PAU. 1|hr

'pause in length of 2 minutes
PAU. 2|MIN

' Example using a variable
$$PAU=0.25
PAU.$$PAU
```

Remarks

PAU. will only PAUSE the script at one place. If you want the whole script to be slowed down, use DIP. - Delayed Instruction Processing 773. The robot is able to receive messages while in PAU. Mode.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

The internal time calculations are done with 32 bits of Precision in ms. You can choose values down to 1 ms, there are no half or zero milliseconds. There is no upper limit you should be able to run against.

See also:

- DIP. - Delayed Instruction Processing 773
- 1.B Error handling 224

© 2023 Theo Gottwald * https://www.smart-package.com

3.48.10 USM. - Use silent Mode

MiniRobotLanguage (MRL)

USM. - prevent error messages

Use Silent Mode * prevent and redirect error messages

Intention

These command is used to tell the robot what to do in case of errors. Its mostly used to prevent errors popping up on client computers. Use it together with the Label 2.2 :on_Error - Label 364.

For more details on Error handling see: 1.B Error handling 224.

Syntax

USM.[P1][|P2]

Parameter Explanation

P1 -(optional) numeric parameter, if omitted, 32 is the default value

```
  0 - immediately end the script, no error handling.
      Can not be combined with other flags.
      :on_Error Label will be ignored!


  1 - just Beep on error, no error message,
       add +4 to make script continue running

  2 - redirect all console Messages to a file,
      use this together with Flag 8 to log all errors in a
file

  4 - ignore error-line and continue script execution,
        if possible. :on_Error Label will be ignored! Has
priority
       over additional options, like 32 (pressing CANCEL).

   8 - Redirect Error-Message to Console window,
       use this together with Flag 2 to log all errors in a
file

   16 -  Show error-message for ~ 50 Seconds, then kill
Message
         and End Script.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
   32 - Normal / default-Mode:
        Show Messagebox with "Retry/Cancel". If user presses
        "Retry", Error line will be skipped, like in Option 4.

  64 - Clear ":on_Error" Stop-Flag. See remark below for this
       special option.

  P2 - Path of the Error-Log file. If omitted:
       In case the script runs from Editor, ?path\MR_Error.log
       else, ?cdesktop\MR_Error.log is used.
```

Example

```
'*********************************************
' USM. - Samples
'*********************************************

'Switch on silent Mode
USM.10

' Switch on Silent Mode with Error-Log file
USM.10|?pfad\MyLog.txt

' Clear ":on_Error" Retry Flag
USM.64

' Switch off Silent Mode
USM.0


END.
:on_Error
MBX. There was an Error


'**********************************
' USM.-Demo
'**********************************
'
USM.10|?desktop\MR_Error.log
STW.tw|Nix|0

MBX.Ready

'-------------------------------------------------------------
ENR.

:on_Error
: $$ERF=?exepath\MR_Error.log
MBX.$$ERF
IEF.$$ERF
  COP.$$ERF|?cdesktop\MR_Error.log
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX. AN error happened!
ENR.
```

Remarks

Remark to Option "64":
In case there is a Label ":on_Error" somewhere in the script,
the robot will jump there in case of an error.
To prevent a circular reference, if the error is in the part after ":on_Error", there is an internal safety-flag that will prevent the "on_Error" to be called a second time.
This flag can be cleared to (retry aomething) using USM.64

`USM.0` does not do any defined action in case of error. The script will just end. However it will not even run into the `:on_Error Label`.

`USM.2` will redirect all console output to the error-log file. This includes all output from any command that writes to the console.

ParallelRobots (`PRR. - PRE.`) do actually not run in he Editor and may use the `?cdesktop\MR_Error.log Log-Path`
even, if the Main-Script runs in the Editor.

Limitations:

-

See also:

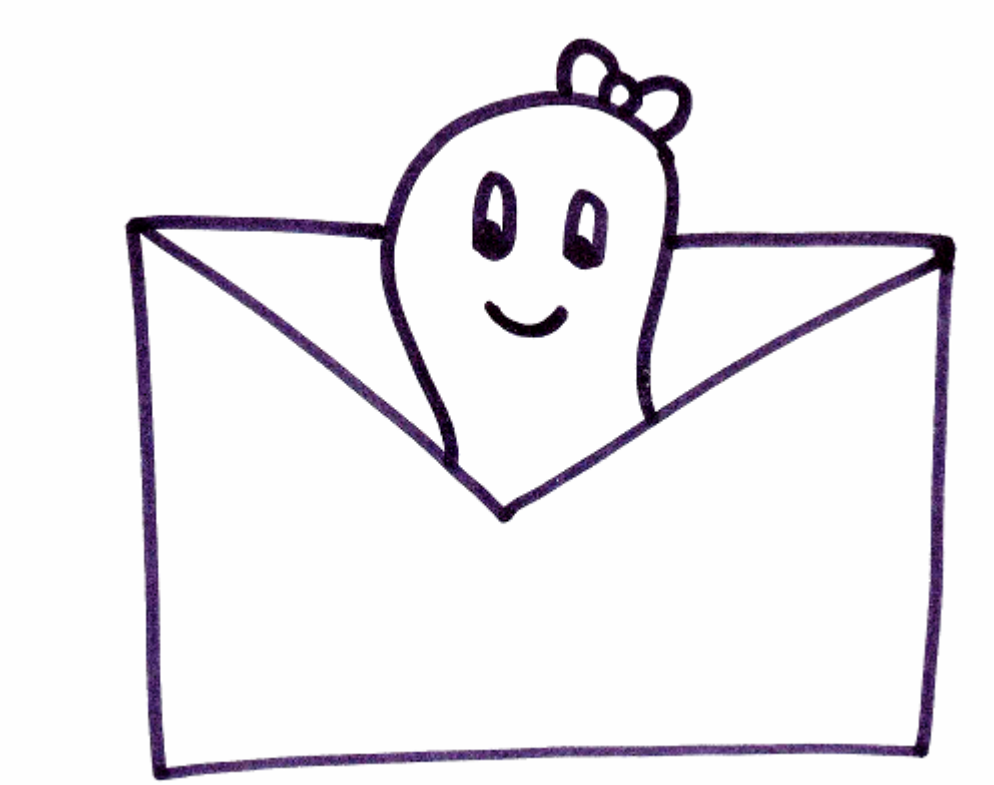
- 1.B Error handling 224
- 2.2 :on_Error - Label 364
- **OPT. - Optional Settings** 2406
- DBP - Debug Print 769
- DBV. - Debug Print Variable 771

© 2023 Theo Gottwald * https://www.smart-package.com

3.49 E-Mail Commands

3.49.1 ! Smarty's Send-Mail

Smarty's Send-Mail



Smarty's Send-Mail

You can use Smarty's Mail-System to send e-mails. For example, to signal States in a End2End-Testing Script.

If you combine the String-capabilities with the `FEX.` (Enum-Commands), Smarty will also be a great Bulk-Mail program.

Smarty's Mail-System supports the following features:

- configurable SMTP Send-Mail
- **easy to use** (GUI for Parameters)
- choose Port 25 or any alternative Port you need
- supports Login Authentication if needed
- **encrypted Password** if you use "Mailconfig.dat"-feature
- returns success or failure
- GUI-Mailconfig.exe to set reusable Send-Mail Configurations

*© 2023 Theo Gottwald * https://www.smart-package.com*

- can send Plain-Text, HTML or combined Mails
- can send unlimited Number of mail-attachments
- supports **CC**: and **BCC**: fields
- supports variable Fields for "**Localhost**" and "**Reply To**"
- *Fast, fast, fast* ... use `SMB.` to send Mails with maximum speed.
- Send **large Mails** that need long time to be **sent, in the background**
- **Sent multiple Mails at the same time**
- the only limit is the limit your provider sets you.

In short, you can sent any thinkable E-Mail using Smarty's Mail-Command.
I have sent up to 100 E-Mails at the same time using the `SMB.`-command.

Trying to send **more then 100 E-Mails (at the same time !)** i got this Error from my Providers Mail-Server:

```
Result from 5972 = ERROR: 421 too many connections
Result from 5968 = ERROR: 421 too many connections
Result from 5980 = ERROR: 421 too many connections
Result from 5996 = ERROR: 421 too many connections
```

If you overrun the capacity that the Mail-Provider gives you, you will get this error.
It proofs, that using Smarty's Send-Mail, you can sent mails to the maximum.

Configuring Smarty's Send-Mail Settings

All Mails are sent using the **POP / SMTP**-Protocol. For this to work, there are some settings that need to be configured.

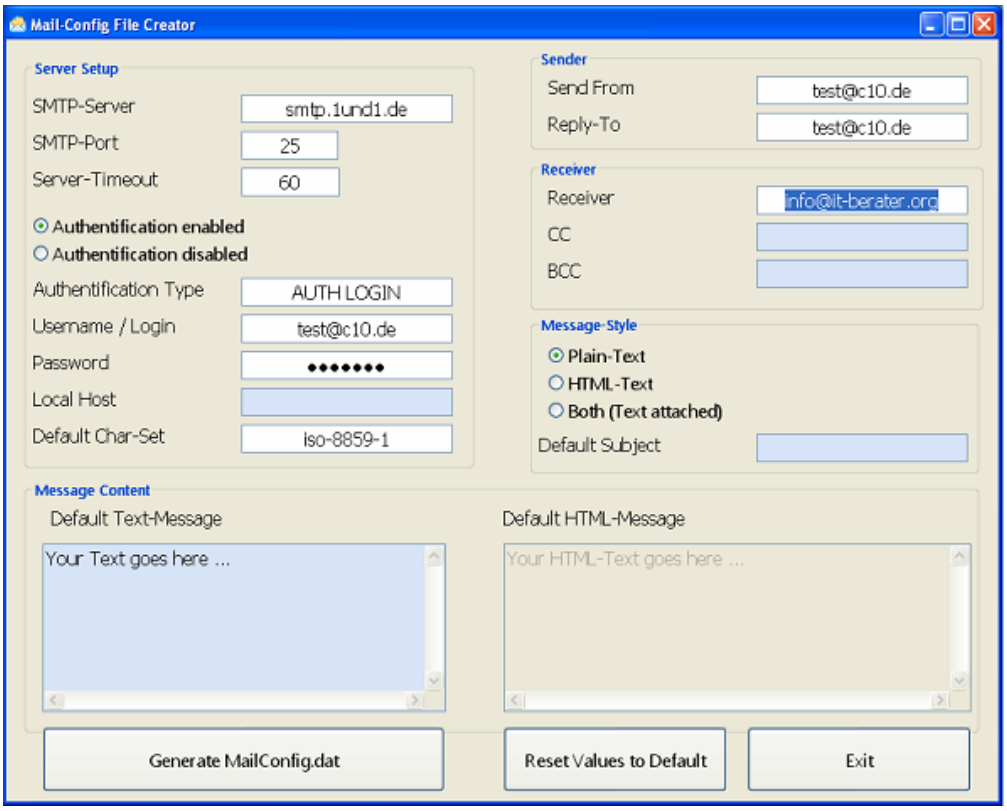
There are two ways to configure Smarty's Send-Mail-System:

1. the `SMP.` - command
2. the "MailConfig.exe" Utility and the resulting "MailConfig.dat"-File

If you use the "Mailconfig.dat"-Option, you can override any of the loaded settings using the `SMP.` commands. This way, you can combine both methods.

Let us now take a quick look on Smarty's "**MailConfig.exe**".

*© 2023 Theo Gottwald * https://www.smart-package.com*



What is a "MailConfig.dat"-File?

A "MailConfig.dat"-File is a file that contains all settings needed to sent an E-Mail out.
A "MailConfig.dat"-File may contain only basic settings, in that case you can define all other settings using the `SMP.`-command.

A "MailConfig.dat"-File **is encrypted,** There is no supported way to read - for example - an **E-Mail Password** out of it. Therefore it serves the purpose that nobody can get your e-mail Password "the easy way".
Note that while all Data inside the "Mailconfig.dat"-file is heavily encrypted, there is no technical way, to guarantee that a dedicated hacker can not get the password out of it.
Therefore you should think of this encryption as a sort of "hiding the password".

How do i generate a "MailConfig.dat"-File?

You can generate a "MailConfig.dat"-File using the supplied "MailConfig.exe"-Utility.

1. To do this, you simply fill out all required fields (they are marked white).
2. And you can fill all optional fields (they are marked blue).
3. Then you press the button "Generate MailConfig.dat".

That's all. Finally you can grab your "MailConfig.dat" on your desktop.

Why do i need the "MailConfig.exe"-Utility?

*© 2023 Theo Gottwald * https://www.smart-package.com*

The "MailConfig.exe"-Utility is the only way to generate a encrypted File called "MailConfig.dat". This file contains all settings that are needed to send e-mails using the `SMA.`-Command.

Do i need to use '#INC: to include the MailConfig.dat into my executable?

No, generally this is done automatically, for you.
If the path of the "`MailConfig.dat`" file is "`?path\`" and if it does not contain variables, it will automatically be added to an executable during compile.
Generally said, the path must be available at compile-time, variables are only available at runtime. In all other cases, you have to add this file to the executable using `'#INC:.`

What does the SMP-Command do, if the file is not available?

You will get a error-message that tells you exactly what was wrong.

How do i add attachments to my Mail?

Use the `SMP.`-command like in the example below.

```
SMP.addfile=?path\Text1.txt;?desktop\results.dat
```

What do i specify as "Receiver"?

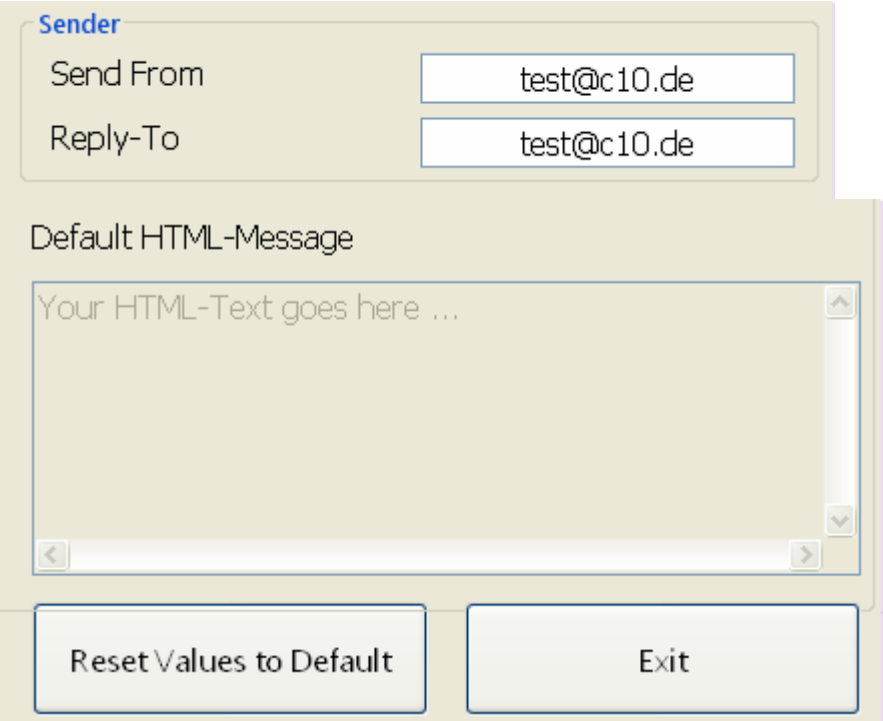
In the fields "Receiver","CC" and "BCC" must be only valid e-mail addresses.
From these fields, the "Receiver" filed is necessary, because that is, where you specify the receiver of the mail. The other two fields van be left blank if you do not need them.
If you sent the same mail to multiple persons, you may think of putting all the receiver mail addresses into BCC, then they will not be visible to each other.

Receiver - e-mail address of the person that shall become the e-mail.

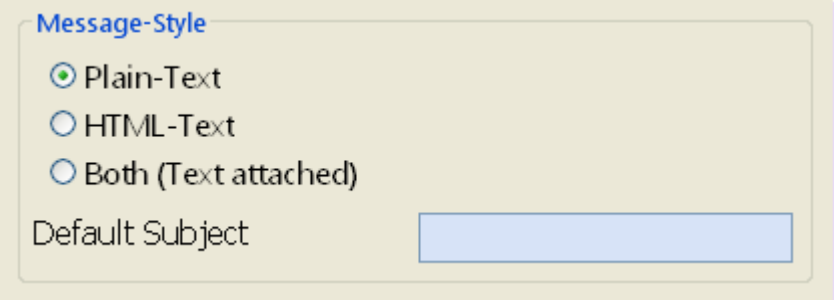
CC (Carbon Copy) - (optional) - e-mail address of person(s) that shall be informed on the subject but are not the primary receiver of the mail.

BCC (Blind Carbon Copy) - e-mail address of person(s) that shall be informed on the subject but are not the primary receiver of the mail and shall not be visible with their mail address to any of the receivers of the mail.

© 2023 Theo Gottwald * https://www.smart-package.com

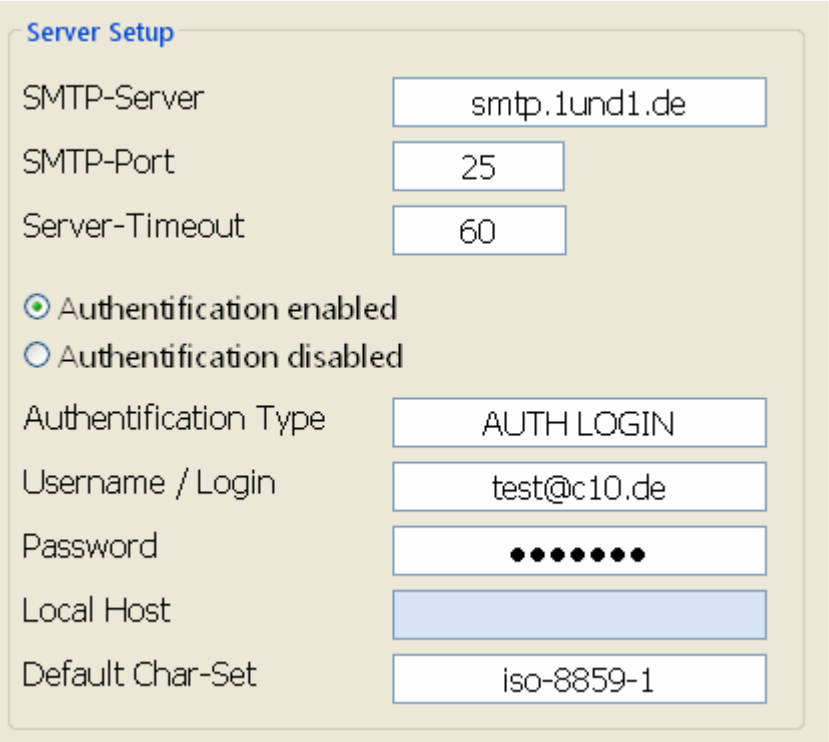


Sent From - This field shows the Sender that is displayed in the receivers e-mail program.
Reply-To: this is the e-mail address which will be used by an e-mail program in case that the receiver of your e-mail will open your mail and press the "Reply" Button in his e-mail program.



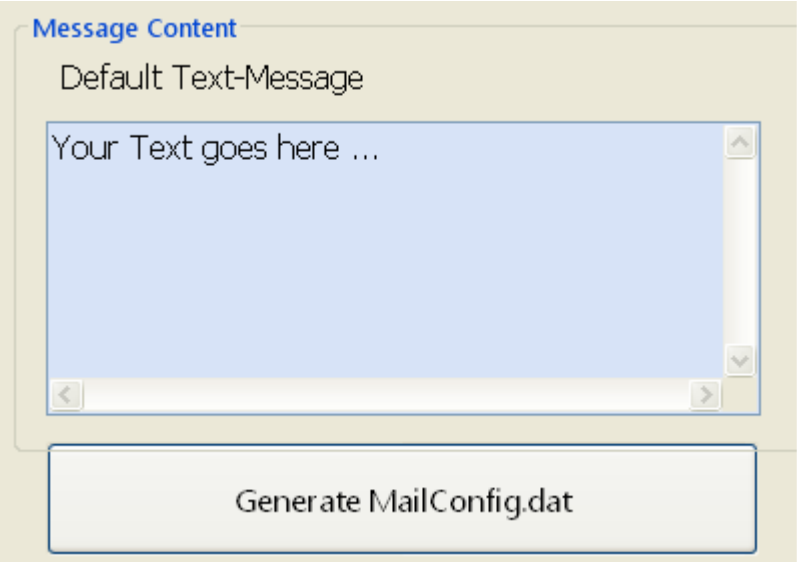
Message-Style: Plain Text / HTML / Both - Choose the format of the e-mail that shall be sent. Depending on your choice some fields in the GUI may become activated or deactivated.

© 2023 Theo Gottwald * https://www.smart-package.com



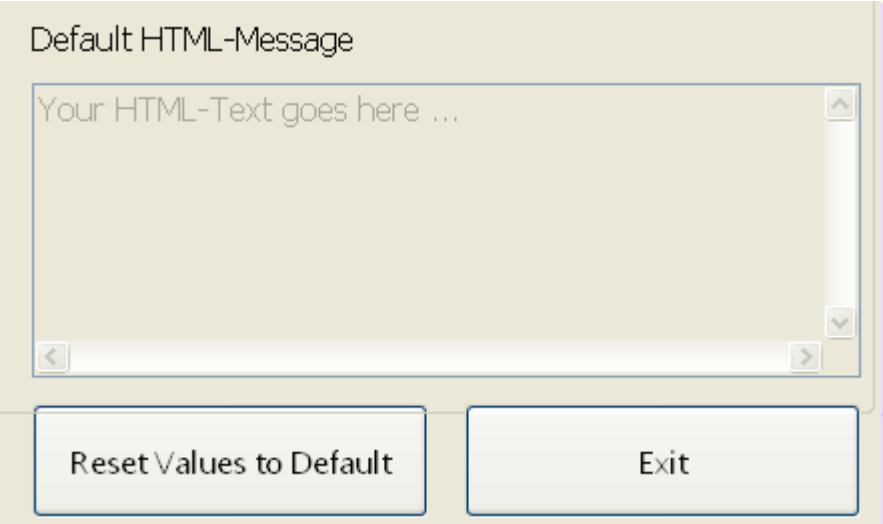
SMTP-Server - these and all following parameters are the same parameters that you may have used in your actual e-mail program. If you do not know them, ask your e-mail administrator or your e-mail provider.

Local Host - E-Mail Programs do include the name of the sending computer with any e-mail. In case you do not want that, you can override this using the "Local Host" Fields. In most cases, its best that you just leave that field empty. The Mail-Command will then internally provide the needed information.



Default Text Message.- you can specify a default text message, that is sent if the mail-message-style is set to "Plain" or to "both". You can override it anytime using the `SMP`.-command.

© 2023 Theo Gottwald * https://www.smart-package.com



Default HTML Message.- you can specify a default HTML-message, that is sent if the mail-message-style is set to "HTML" or to "both". Preferably, you will just paste the HTML code into this field. You can override it anytime using the `SMP.`-command.

What happens in case of errors during the send?

Generally the `SMA.` command does not generate a Error-Message with a dialog. Because even if everything is all right, a mail may not be sent, because the provider has trouble on his mail or DNS-Server.

What you get is a Return-Value.
If you use a "MailConfig.dat", and the file is in `?path\`, then your mail-script can be as simple as:

```
SMP.
SMA.
' Print Returncode or OK Message
DBP.$$000
```

If everything worked fine, this is what you get:

```
[15] OK
```

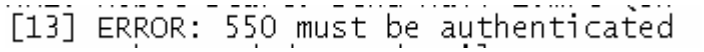
In case your parameters are not valid, or in case the server returns an error, this is what you get:

```
[13] ERROR: 501 Address syntax error: <sendto>
```
or
```
[13] ERROR: 501 Address syntax error: <Theo's Testaccount (@.)>
```

In these cases, check the field that is shown in the error-message.

If you get this error, you need to provide valid Login-Credentials for your **SMTP-Server.**

© 2023 Theo Gottwald * https://www.smart-package.com



See also:

- **SMA . Send e-Mail** 797
- **SMB. Send Mail Background** 799
- **SMP. - Set e-Mail Parameter** 803
- DLF. - DownLoad-File 2074
- DLV. - Download-to-Variable 2076
- URL. - Split URL 3234

© 2023 Theo Gottwald * https://www.smart-package.com

3.49.2 SMA . - Send e-Mail

MiniRobotLanguage (MRL)

SMA. - Send e-Mail

Send e-Mail

Intention

Send a e-Mail that was previously defined using the SMP.-command.
Wait until the e-mail was sent, or until he server returned an error.

Please see the SMP.-command 803 for further information on how to define and send an e-mail. Example:

```
' Set E-Mail Configuration
SMP.?path\MailConfig.dat
' Send E-Mail and wait ...
SMA.
' Return the Answer of the Mail-Server from Stack
DBP.$$000
' End the script
ENR.
```

In case you need to send multiple Mails with attachments, this can take a while. To prevent your Script from being blocked, you can use SMB. (Send Mail Background) in that case.

Syntax

SMA.[P1][|P2]

Parameter Explanation

P1 - (optional) If given, this value will overwrite the content of the "Send-To" Field.

P2 - *(optional)* Variable that will contain the answer of the Mail server.

Example

```
'**********************************
' SMA.-Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
SMP.smtp-server=smtp.1und1.de
SMP.port=25
SMP.sender=you@yourdomain.com
SMP.receiver=yourfriend@hisdomain.com
SMP.subject=Invitation
SMP.text=Hallo friends,$crlf$this is going to be interesting!
SMP.auth=1
SMP.password=dsa8sdkj-*
' And let it go ...
SMA.
' Print if it was OK
DBP.$$000
ENR.

'**********************************
' SMA.-Sample
'**********************************
SMP.?path\MailConfig.dat
SMA.
DBP.$$000
ENR.
```

Remarks

-

Limitations:

When using `PRR.`, the Parallelrobot will not inherit the Mailconfiguration from the Starting Robot. To send Mails in the Background, use `SMB`.

See also:

- ! Smarty's Send-Mail 789
- SMB. Send Mail Background 799
- SMP. - Set e-Mail Parameter 803
- DLF. - DownLoad-File 2074
- DLV. - Download-to-Variable 2076
- URL. - Split URL 3234
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.49.3 SMB. - Send Mail Background

MiniRobotLanguage (MRL)

SMB. - Send e-Mail Background

Send an e-Mail in the Background

Intention

Send a e-Mail that was previously defined using the `SMP.`-command.
Sent the E-Mail "Non Blocking", script execution will go on, while the e-mail is been send.

Unlike with `SMA.`, the script execution will **not** wait until the e-mail is sent.
Instead, the script can go on with execution. The e-mail is sent in the background.
`SMB.` will leave a "ticket number" that will give you anytime access to the Result of the background operation. Please use `DRS.` (Drop Result) is you do not want to Get that result of the background operation.

Please see the SMP.-command 803 for further information on how to define and send an e-mail. Example:

```
' Set E-Mail Configuration
SMP.?path\MailConfig.dat
' Send E-Mail and wait ...
SMB.
' Do anything else you want in this time
' here while the mail is sent
' ...
' Then wait until the background operation has finished.
WRS.
' Get the result of the last background operation on TOS
GRS.
' Always Drop the result, so it won't waste Memory
DRS.
' Return the Answer of the Mail-Server from Stack
DBP.$$000
' End the script
ENR.
```

How can I get the result of exactly this mail, if its running in the background?

The `SMB.` - command will drop an "**ticket number**" on the TOS.
Or, if **P2** is given, the ticket-number will be in the given variable.

Now you can get the result any time using the "**ticket number**" and these commands.
Any result behind an "ticket-number" ist stored as long, until you get it using `GRS.` or until you Drop it, using `DRS.` You do not need to wait until the Operation is

*© 2023 Theo Gottwald * https://www.smart-package.com*

finished, to drop the result. You can drop the result at any time, even if the Operation is just been started.

Use these commands:

- GRS. - Get Result of Operation
- WRS. - Wait until (one or more) Operation(s) is/are finished
- DRS. - Drop Result of Operation if you do not need it.

to handle the result of the background mail-operation(s).

Example:

```
' Assume that a file "?path\MailConfig.dat"
' is available
SMP.
' It makes mostly sense with attachements
' SMP.attachment=?path\Send-Mail.zip
' Send the Mail, get the ticket number
SMB.
' If we do not specify a ticket number,
' WRS. will automatically remember the last ticket number
' and use it.
WRS.
' If we do not specify a ticket number,
' GRS. will automatically remember the last ticket number
' and use it.
GRS.
DBP.$$000
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
' and use it.
' DRS. free's the Memory location that was used to hold the
' Result of the background operation.
DRS.
ENR.
```

In case you need to send multiple Mails with attachments, this can take a while. To prevent your Script from being blocked, you can use SMB. (Send Mail Background) in that case.

Syntax

SMB.[P1][|P2]

Parameter Explanation

P1 - (optional) If given, this value will overwrite the content of the "Send-To" Field.

P2 - *(optional)* Variable that will contain the "Ticket-Number" which can be used to get the answer of the Mail server an any time, using the GRS.-command..

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' SMB.-Sample
'**********************************
' Send away 25 Mails
' then wait until they are sent
' and then show the results of all 25 Mails.
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Background Mailer-Demo
' Please provide a complete MailConfig.dat
SMP.?path\MailConfig.dat
' Try with attachement also!
' SMP.attachment=?path\Send-Mail.zip
$$HND=
' Be careful or you may end up with in
' an error 421 - too many connections
' from your provider
FOR.$$CNT|1|25
  GSB.Lab
NEX.
DBP.$$HND
WRS.$$HND
' Display Results
FEM.Lab_dis|$$TXT|$$HND|,
MBX.!
ENR.
'-------------------------------------------------------------
:Lab_dis
GRS.$$TXT|$$RES
' Always drop Results you do not need anymore
DRS.$$TXT
DBP. Result from $$TXT = $$RES
RET.
'-------------------------------------------------------------
:Lab
SMB.
$$HND=$$HND,$$000
RET.
'-------------------------------------------------------------
```

Remarks

If you use Background Operations, always drop the result of the operation using `DRS..` Otherwise it will be kept in Memory until the Script ends.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

When using PRR., the Parallelrobot will not inherit the Mailconfiguration from the Starting Robot. To send Mails in the Background, use SMB.

See also:

- ! Smarty's Send-Mail 789
- SMA . Send e-Mail 797
- SMP. - Set e-Mail Parameter 803
- DLF. - DownLoad-File 2074
- DLV. - Download-to-Variable 2076
- URL. - Split URL 3234
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.49.4 SMP. - Set e-Mail Parameter

MiniRobotLanguage (MRL)

SMP. Command

Set e-Mail Parameter

Intention

The `SMP.` command will define all parameters, that are needed to send an e-mail, using the `SMA.` - command.

You have two ways to use the SMP. - command:

1. Use it to **load an "`MailConfig.dat`" file** with an previously saved, encrypted Mail-Configuration.
2. Use it to **define some or all parameters directly**. In this mode, parameters that have already been defined using a "MailConfig.dat" can be overwritten with other values.

Let's start using Mode 1 and the "MailConfig.dat". How do you generate this file? There is GUI-Utility that helps you to generate it. You can read all this here:

! Smarty's Send-Mail 789

Be sure to read that before going on.

whether you use a "`MailConfig.dat`" as Base-Configuration, or not, in any way you can set or change Mail-Parameters using `SMP.`

Actually you can not retrieve Mail-Parameters that have been set. If you are not sure about these values, just set them again with the proper values. This way, any parameters that have been set using the "MailConfig.dat" are kept secret.

Let us now take a look on how you are really sing to sent an e-mail, in both modes of operation.

If you have already defined an "`MailConfig.dat`" file, and placed it in ?path\ 345, then there is not much to do. Your script could look as simple as:

```
' SMP without any parameter will load a file called
' "MailConfig.dat" from ?path\
SMP.

' SMA. will just send away the previously defined e-mail.
SMA.

' Thats just the END of the Script
END.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Now in most cases, you may have defined a base-Configuration containing your provider-Settings etc. in a "MailConfig.dat" file. And you add all variable settings later using SMP. Like in this example:

```
' First we load our predefined Mailconfiguration
SMP.
' Now we add two file-attachments
SMP.addfile=?path\Text.txt;?desktop\File.zip
' This is the new text for our mail
SMP.text=Hallo Peter$crlf$Do you have time this weekend for a
football match?
' We make sure that the mail is sent as an plain-text-mail
SMP.style=plain
' We set the receivers e-mailaddress
SMP.to=Peter@c10.de
' Now we let our bird fly.
SMA.
' and end the script
ENR.
```

But there is no need to use the "Mailconfig.dat" file. You can do the whole Mail-configuration solely using the SMP.command. You will however need to write a bit more, like in the following example:

```
SMP.smtp-server=smtp.1und1.de
SMP.port=25
SMP.sender=you@yourdomain.com
SMP.receiver=yourfriend@hisdomain.com
SMP.subject=Invitation
SMP.text=Hallo friends,$crlf$this is going to be interesting!
SMP.auth=1
SMP.password=dsa8sdkj-*
' And let it go ...
SMA.
' Print if it was OK
DBP.$$000
ENR.
```

As you can see, we did not specify the fields:

- reply-to
- login

because these two are set automatically to the same value as "send from", in case you leave them "empty".

The field "Localhost" is been set to the name of the actual computer, once when the robot starts up. If you clear the field, it will be automatically set to the term "localhost".

The following fields are set up with default values, once the robot starts up:

```
Timeout     =   60 seconds
CharSet     =   "iso-8859-1"
Style       =   "plain text"
SMTPPort    =   25
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EnableAuth   =    %FALSE
AuthType     =    "AUTH LOGIN"
LocalHost    = actual Computer's name
```

Syntax

SMP.[P1]

Parameter Explanation

P1 - (optional) can be one of two formats:

1. **empty**, in this case a file "MailConfig.dat" is searched for in "?path\". If the file is found, it is loaded and all values are set according to the content of the file.

2. A **value-equation**, like: smtpport=25
   in this case, the equation is been evaluated and the value is set. You can not make multiple equations in a single line.
   Use a new line with a new SMP.-command for each necessary equation. This was necessary, to have all sorts of characters available for the "text" and the "password" fields.
   Also you will find it very readable.

You can specify the following equations. Note that for all equations, the term on the left side is **not** case sensitive.

"SMTP-Server","smtp"
Will choose the desired SMTP-Server.

Examples:
```
SMP.smtp=pop.gmail.com
SMP.smtp=pop3.live.com
SMP.mail.comcast.net
SMP.popmail.t-online.de
```

"SMTP-Port","SmtpPort","port"
Will choose the desired SMTP-Server Port.
Often used values are 25 or 587. As the value of 25 is pre-set, in many cases you do not need to touch this setting.

Example:
```
SMP.port=25
SMP.port=587
```


© 2023 Theo Gottwald * https://www.smart-package.com

"SendFrom","send from","send-from","from","Sender"
Will set the e-mails "sender" address. This is the mailaddress that is shown to receiver of the mail, when he finds the mail in his Inbox. You have to specify something that looks like a e-mail address in this field, to have the mail sent. Note that this is not necessarily the mail addresss, a "reply" will be sent to. Use the "Reply-to" field for this.

Example:
```
' All three lines below do the same.
SMP.Sender=info@it-berater.org
SMP.from=info@it-berater.org
SMP.sendfrom=info@it-berater.org
```

"receiver","sendto","send-to","to"
You need to specify this field. It must contain a valid e-mail address.
This field contains the e-mail address of the primary receiver(s) of the e-mail.
You can specify multiple receivers, if you separate them using a semicolon ";".
There is no limitation on number of receivers from the `SMP.`-command.
Note that any receiver can see all other receivers that are in the receiver field.
Example:
```
SMP.receiver=mailbox1@fa2.de;yourmail@c10.de
SMP.to=mailbox1@fa2.de;yourmail@c10.de
SMP.send-to=mailbox1@fa2.de;yourmail@c10.de
```

"CC","Carbon Copy","carbon-copy"
You do not need to specify this field. If you do so, it must contain a valid e-mail address.
This field contains the e-mail address of secondary receiver(s) of the e-mail.
You can specify multiple secondary receivers, if you separate them using a semicolon ";".
There is no limitation on number of CC-receivers from the `SMP.`-command.
Note that any receiver can see all other receivers that are in the CC-receiver field.

Example:
```
SMP.CC=mailbox1@fa2.de;yourmail@c10.de
SMP.carbon copy=mailbox1@fa2.de;yourmail@c10.de
SMP.carbon-copy=mailbox1@fa2.de;yourmail@c10.de
```

"bcc","blind copy","blind-copy"
You do not need to specify this field. If you do so, it must contain a valid e-mail address.
This field contains the e-mail address of secondary receiver(s) of the e-mail.
You can specify multiple secondary receivers, if you separate them using a semicolon ";".
There is no limitation on number of CC-receivers from the `SMP.`-command.
Note that any receiver that was specified in the BCC-field is hidden and can not be seen by all other mail-recipients Example:

```
SMP.BCC=mailbox1@fa2.de;yourmail@c10.de
SMP.blind copy=mailbox1@fa2.de;yourmail@c10.de
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SMP.blind-copy=mailbox1@fa2.de;yourmail@c10.de
```

"Reply","reply to","reply-to","ReplyTo"
You do not need to specify this field. If you do so, it must contain a valid e-mail address.
This field contains the e-mail address that a mail will be sent to, if the recipient of the mail click on the "Reply" button.

Example:

```
SMP.reply=mailbox1@fa2.de
SMP.reply to=mailbox1@fa2.de;yourmail@c10.de
SMP.replyto=mailbox1@fa2.de;yourmail@c10.de
```

"subject","titel"
This is the Subject of your e-mail. You do not need to specify a subject.

```
SMP.Titel=Birthday Invitation
SMP.Subject=Testing Result from X5/8452
```

"plain-msg","plain-message","text","text-msg"
This field contains a plain-text message, if you send one. There is no limit in text-size from the `SMP.`-commands. There may however very likely be a limit from your SMTP-Server for the total size of your mail.
You would do something like this:

```
' Read Textmessage from a file
CFF.$$FIL|$$TXT
' You can do Text-Replace here to customize Mails.
' using the String-Commands.
' This Line will set the text for the Mail
SMP.Text=$$TXT
```

"html","html-text","html-msg","Html-Message"
This field contains a html-text message, if you send one. There is no limit in html-text-size from the `SMP.`-commands. There may however very likely be a limit from your SMTP-Server for the total size of your mail. You can do something like this:

```
' Read Textmessage from a file
CFF.$$FIL|$$TXT
' You can do Text-Replace here to customize Mails.
' using the String-Commands.
' This Line will set the text for the Mail
SMP.HTML=$$TXT
```

"auth","anmeldung","authentication"
This field contains a "1" if the SMTP-Server needs a User-Authentication and a "0" if not.
You can also use the terms "On" and "Off" to make it more readable.

Examples:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
SMP.Auth=On
SMP.Authentication=Off
```

"Auth-type"
This field contains the only currently supported Authentication type. You can change it, however that may lead to unforeseeable results. Most likely things will not work anymore.
If you set this field to empty, it will automatically be reset to its default value, that is "`AUTH LOGIN`".

"Login","user","Username","Auth-Username"
If User-Authentication is set to "**On**", then this field contains the Login-credential. This is most often the users e-mail-address.
If you leave this field empty, then it is automatically filled with the content of the "**sender**" field. Example:

```
SMP.Login=info@it-berater.org
```

"pwd","Password","pass","AuthPassword"
If User-Authentication is set to "**On**", then this field contains the needed Password. The password can contain any character, especially if you deliver it inside a variable. Variables behind the password-field are only resolved once (binary compatible). Therefore your password could contain a sequence like "`$$TXT`" without having it resolved as a variable, if the password itself is fully inside a variable. You can use the `VAB.`-command to specify a password safely. If you do not want to specify the password openly visible, use the "`MailConfig.exe`" and encrypt it inside the "`Mailconfig.dat`" file.

Example:

```
' This will really work.
VAB.$$PWD=a3D$$TXA34?pa
SMP.pwd=$$PWD

' This is the normal case
SMP.pwd=das8!%sasad34_2
```

"pc","localhost","host","SenderHost"
The localhost field specifies the name of the sending computer.
If you set the field to empty, because you do not want your computers name to show up,
it will internally be set to "localhost"-

"style","msg-style","MessageStyle","message-style"
Using this field, you can specify the e-mail format.
That is, if you want to send the e-mail as "Plain Text" or as "HTML-Text" or as both.
The default format is "Plain" and therefore you may not need to change that in most cases. You can choose between:

*© 2023 Theo Gottwald * https://www.smart-package.com*

- **"1","Plain"**
- **"2","html"**
- **"3","Both"**

Example:

```
' Send message as Text-Message
SMP.style=plain
' Send Message as HTML Message
SMP.style=2
' Send Message as combined Text/HTML Message
SMP.style=Both
```

"attachment","addfile","sendfile"
You can add attachments to your e-mail. Unlike all other properties, attachments can only be added using the `SMP.`-command. They can not be predefined in the `"Mailconfig.dat"` file. There is no technical limit on the number of attachments until your Mail-Server will not accept any more.
To add one or more attachments, just add their filenames (including pathes) separated by semicolons behind the `SMP.` command. Like this:

```
SMP.addfile=?desktop\myfile.zip;?path\mytext.txt
' You can also specify a variable, if the line
' would be getting too long otherwise.
SMP.sendfile=$$FIL
SMP.attachment=$$FIL;$$FIA;$$FIB
```

"char","Char-Set","chars","charset"
You can specify the used char-set to have the receiver see the mail in the proper way.
Most often, you will use just:

```
' Use "us-ascii" charset
SMP.char=us

or
' Use "iso-8859-1" charset
SMP.char=iso
```

You can specify any other char-set you want:

```
' PL, CZ etc. may use this charset
SMP.char=iso-8859-2
```

"TimeOut","Server-TimeOut","ServerTimeout"
This field contains the Server-Timeout in seconds.
If the server does not respond within this time, an error is been internally generated.

More Hints:

© 2023 Theo Gottwald * https://www.smart-package.com

The fields:

- **reply-to**
- **login**

are set automatically to the same value as ifrom "**send from**", in case you leave them "empty".

The field "**Localhost**" is been set to the name of the actual computer, once when the robot starts up. If you clear the field, it will be automatically set to the term "**localhost**".
Because it can't be empty.

The following fields are set up with default values, once the robot starts up:

```
Timeout      =   60 seconds
CharSet      =   "iso-8859-1"
Style        =   "plain text"
SMTPPort     =   25
EnableAuth   =   %FALSE
AuthType     =   "AUTH LOGIN"
LocalHost    = actual Computer's name
```

Example

```
'**********************************
' SMP./SMA.-Sample
'**********************************
' First we load our predefined Mailconfiguration
SMP.
' Now we add two file-attachments
SMP.addfile=?path\Text.txt;?desktop\File.zip
' This is the new text for our mail
SMP.text=Hallo Peter$crlfDo you have time this weekend for a
football match?
' We make sure that the mail is sent as an plain-text-mail
SMP.style=plain
' We set the receivers e-mail-address
' please put your own mail-address here
' so you can receive your own Test-mail.
SMP.to=Peter@c10.de
' Now we let our bird (with the message) fly.
SMA.
' and end the script
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

When using `PRR.`, the Parallelrobot will not inherit the Mailconfiguration from the Starting Robot. To send Mails in the Background, use `SMB`.

See also:

- ! Smarty's Send-Mail 789
- SMA . Send e-Mail 797
- DLF. - DownLoad-File 2074
- DLV. - Download-to-Variable 2076
- URL. - Split URL 3234

© 2023 Theo Gottwald * https://www.smart-package.com

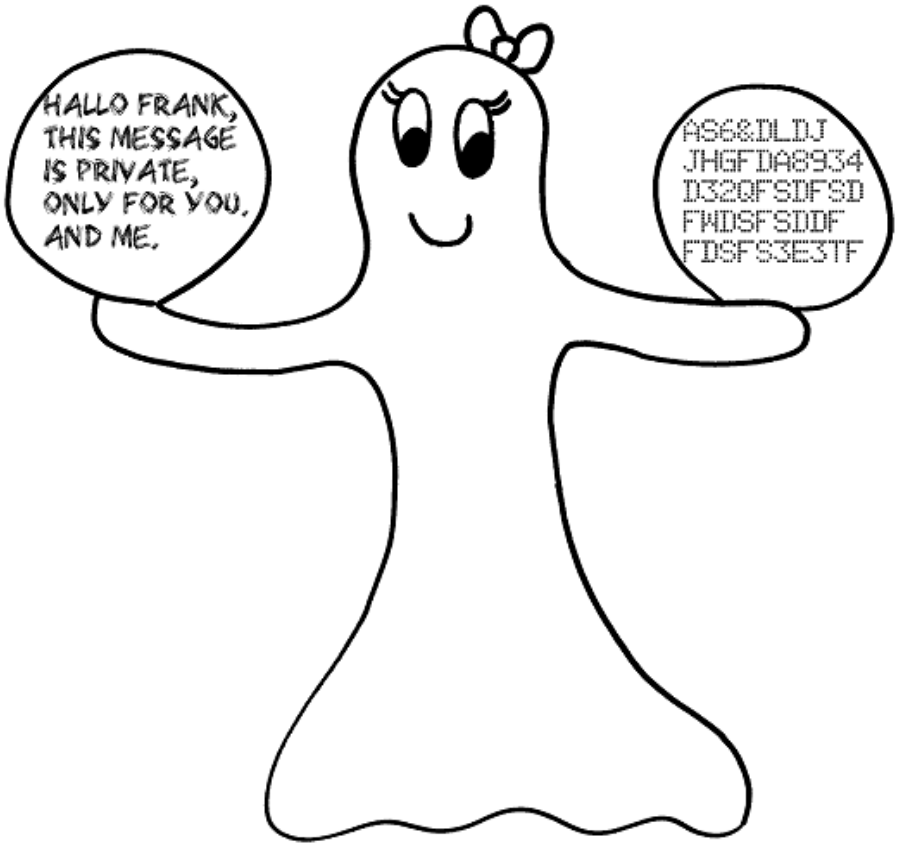
3.50 Encryption and Hash Functions

3.50.1 ! Smarty's Encryption and Hashing Commands

Smarty's Encryption- and Hash-Functions **Previous** 803 **Top** 41 **Next** 825

Smarty's Encryption- and Hash-Function

Using GEC, you can make Smarty encrypt Messages for you. To make a long story short, **Smarty supports unbreakable Cryptography, if you keep some rules.**



Using GEC, you can make Smarty encrypt Messages for you. To make a long story short, **Smarty supports unbreakable Cryptography, if you keep some rules.**

There are many cryptographical algorhytmns out there with nice sounding names.
But there is **only one of all these, that is scientifically unbreakable**.
It is the so called "**One Time Pad**" (OTP).
Its mathematically proven, that a OTP is safe against cracking, as long as its done properly. Read more about a "One Time Pad".
In short, a OTP is a combination of your (prepared) message with a matrix of totaly random characters. We will however use the Term OTP also for the "Matrix of random characters" itself.

*© 2023 Theo Gottwald * https://www.smart-package.com*

To use the OTP properly, these rules must be followed:
1.The OTP must only be know by you and be the receiver of the message
2. the OTP must have at least the size of the Message that is to be encrypted
3. the OTP must only be used once and then deleted on both sides
4. the OTP must be cryptographically random, this means that a "true random number generator" with a very long periode - or a hardware random number generator - must be used.
5. Your computer, as well as the computer from your contact, must not be compromised (Trojan etc.).
6. To delete the OTP, you need to use a safe method, not just the windows "Delete File".
For this purpose, we have included a "safe-erase file" in to the "DEL.-Command".

If any of these points above is not followed, the OTP may not be perfectly safe, see here: Possible OTP-Exploits.

Comment about deleting used "Pads":
As said, OTP's should only be used "once". Why?
Because once somebody can guess part of the content of a message, he can calculate the OTP and would then be able to decode all conversations.
Imagine an OTP as

There are mostly two standards to safely erase files,
1. The US DoD 5220.22-M standard (3-pass and 7-pass) -which is to weak for today.*
2. the Gutmann method, which overwrites the file with random data 35 times.
The SPR will beat both standards and add some passes to make sure your file is safely overwritten.
File-Buffering should be taken into account when using the SPR or any other of such tools.
Also if you use SSD's, they must never get into wrong hands, as a safe data-erase on SSD's may be a problem, due to the fact that the SSD internally tries to limit the write access to single blocks by exchanging them transparently against "blocks with less writes".. Which could lead to the fact that the blocks with the data may be left somewhere on the SSD, until the SSD is completely cleared.

Comments about the PRNG's in the SPR:
To make a unbreakable OTP-Encryption, we need a "Pad" of the highest possible quality.

To make sure, that the SPR meets all of these requirements, we have:
1. Included several Pseudo-Random Number Generator's (PRNG) into the SPR.
Any of these should be safe for cryptographical use. These PRNGs are integrated in the
! STR. - String Command 3150 and this way you have a choice which of these you use.
And nobody knows.
2. Of course you can and should test the Quality of your SPR generated Random Numbers here:
BATTERIES OF STATISTICAL TESTS FOR RANDOM NUMBER GENERATORS
but after submitting them to the Internet, don't use them any more - generate new OTP's
instead.
3. By just combining a OTP of a limited Bit-Range - like a Text - with a Full-ASCII-Range OTP, the result may be statistically unsafe. Therefore the GEC-Command will first encrypt the
Message with the TEA-Encryption, and then use the unbreakable OTP.

© 2023 Theo Gottwald * https://www.smart-package.com

This way, the Message has already a nearly perfect bit-distribution, before the OTP is used.
4. You can combine generated OTP's using the XOR. - XOR - Encryption 843. In fact this is done
internally already with STR.CHARS D and therefore of no need from as we see it.
If you decide to do so, use only Pads of highest cryptographicaly safe randomness.
Otherwise this may decrease the level of safety, rather then improving it. For details on
the used PRNG's see the ! STR. - String Command 3150.
5. While we believe that the Random number generators that we use in the SPR are safe for
cryptographical use, we can supply a hardware-Random Number generator that uses electronical noise to generate random numbers. If you need such a device you can contact us.

Other possible encryption methods

The SPR can apply normal Password-Protected Encryption using the TEA-Encryption scheme.

TEA is safe and generally fine when:

1. keys are random, this is the case if you use ! STR. - String Command 3150. to generate the Keys.
2. the encrypted data is less then 1 GB
3. Safety is even better, if you use the optional "Bit-Flood" in the GEC.-Command.
Anyway for a maximum security against professionals, we can only recommend OTP's.

Here is an example on how to go:
1. Use the ! STR. - String Command 3150 to generate as many "One Time Pads", as you need for your communication. The SPR can easily create 1000'th of them in a Minute.
2. Share these "One-Time Pads" with your communication partner.
3. Use each OTP only one time (one day for example), next day - next Pad.
Think of these OTP's like your TAN-Numbers from the home-banking.
4. Have the OTP's numbered for each day in an month and use the OTP only on that day.

There is more ...
Using the SPR you can split large files of virtually any size into smaller files, using the `FIL.Split File` - command.
And later you can join the small parts again to the original file, using `FIL.Join delete` - command.
Also the above command will make an basic encryption of your splitted file with (default) 30 Rounds of TEA.

Then there is the "Byte-Flood"-Algo. Its implemented in the `HIT.-command.`
This command can be used to make a fast XOR.-encryption without having a password. Using "Byte-Flood", the file itself is been used as Encryption Matrix.

There are no secrets ...
All cryptographical technologies used in teh SPR are based on public available, scientifically approved methods.Mostly based on direct machine-code of very short size. Therefore the result of all algorhytms (TEA., XOR.,SHA., GMD. etc.) can be tested and compared to public sources.

Random Number generation...

© 2023 Theo Gottwald * https://www.smart-package.com

Its like the generation of TAN-Numbers for your home-banking. Once an attacker would get an idea how you generate your numbers, he could theoretically calculate the whole periode of random numbers and try to guess them.
The bad news is that the time to calculate the whole periode of random numbers, we use in the SPR is theoretically about 1700 years on a Standard computer.
But now what we do is, we combine the resulting numbers (random times) with other generated numbers. The result is that even if somebody could "crack" our PRNG, he would need to also check a lot of combinations and the possible combinations of such a large sequence is quadratic to such a sequence. Now note that we do not only combine 2 such random "Pads" but 10 or 20 when using for example
`STR.CHARS D`
to generate a One-TIme-Pad.

The only alternative to this is, to use a **hardware-Random Number Generator** on a Computer that is not connected to any network.For companies who want the maximum safety, this is the system that we recommend for the OTP-(Random Number) Generation.
The "Random Numbers" (OTP's) can then be given to company members via USB-Sticks or on a CDR. We can supply such a Hardware RNG, please contact us if you are interested.

How can you make your own perfect OTP's?
The perfect OTP-Generator is a Computer that is not connected to a network.
It will just be dedicated to the Generation of OTP's and store them on USB-Sticks.
If possible, it will use a hardware random device. However ... our Software PRNG's are also very good. And at the time of this writing i have no reason to assume that anybody in the world could guess their random number sequence with actual technology.

How fast does the SPR create "OTP"'s / random numbers?

We have used the script below to generate random numbers using all four *PRNG's*

```
'
'SPR Script-file: PAD-Sample
'Purpose:
'Author: Theo
'Creation date: 11-19-2014 at 13:24:40
'=============================================================
' PAD-Size 1 Mrd. Byte
$$NUM=5E8

VAR.$$LET=A
GSB.create_pad

VAR.$$LET=B
GSB.create_pad

VAR.$$LET=C
GSB.create_pad

VAR.$$LET=D
GSB.create_pad

MBX.Ready
ENR.
:create_pad
VBT.$$LET
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$CNT|1|3
  GSB.make_pad
NEX.
RET.
ENR.
:make_pad
VAR.$$TIA=#dtime#
VAF.$$FIL=?pfad\PAD_$$LET$$CNT.txt
$$PAD=
STR.CHARS $$LET|$$NUM|$$PAD
VAR.$$TIB=#dsince#
CTF.$$FIL|$$PAD
VAR.$$TIC=#dsince#
VAR.$$TXT=STR.CHARS $$LET needed $$TIB seconds to$crlf$ create
the Pad of $$NUM Bytes. $crlf$ Additionally $$TIC sec. for
writing  the Pad to HD).$crlf$
PRT.$$TXT
RET.
ENR.
'============================================================
```

The picture below shows the time in seconds, for the generation of 500 Million Random Numbers using all four algos. As you can see, Algo "`STR.CHARS B`" is the slowest. That is because he will internally generate 20 "Random Pads" and combine them.

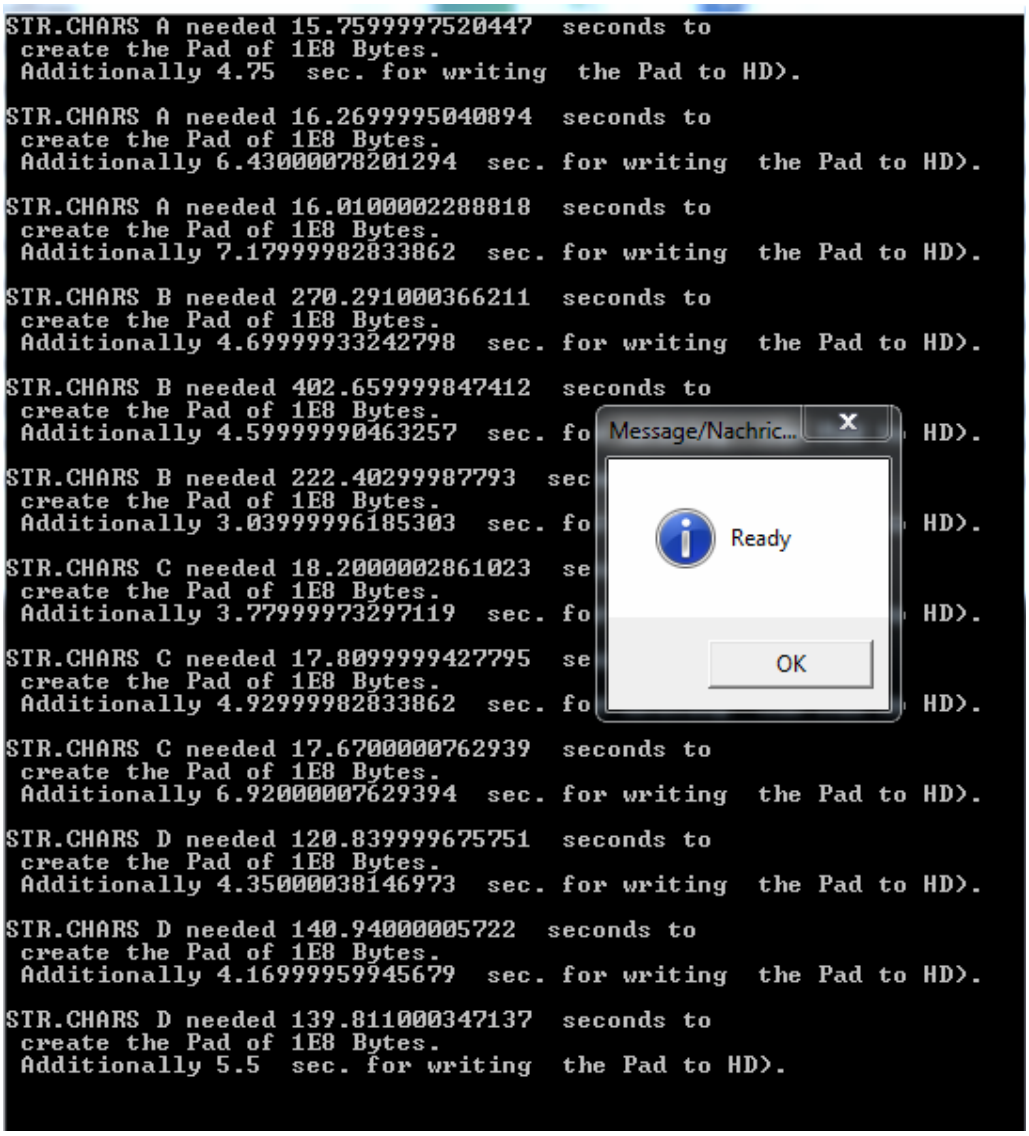
The next slowest Algo is "`STR.CHARS D`", he will combine about ~5 Random Pads. You can however get a OTP of 500 MB in 1 and a half Minute, using the "`STR.CHARS A`" Algo.

© 2023 Theo Gottwald * https://www.smart-package.com

How good are our random numbers?

To Test Random Numbers there are several Methods out there,
One of them is the so called "**ENT-Test Suite**".
We could not use this suite on our 500 MB Pads because the ENT Programm was unable to handle data of this size. SO we created smaller Pads of 100 MB.
Note that these numbers below may not be perfectly accurate because there were other processes running on the test computer in the background.

© 2023 Theo Gottwald * https://www.smart-package.com



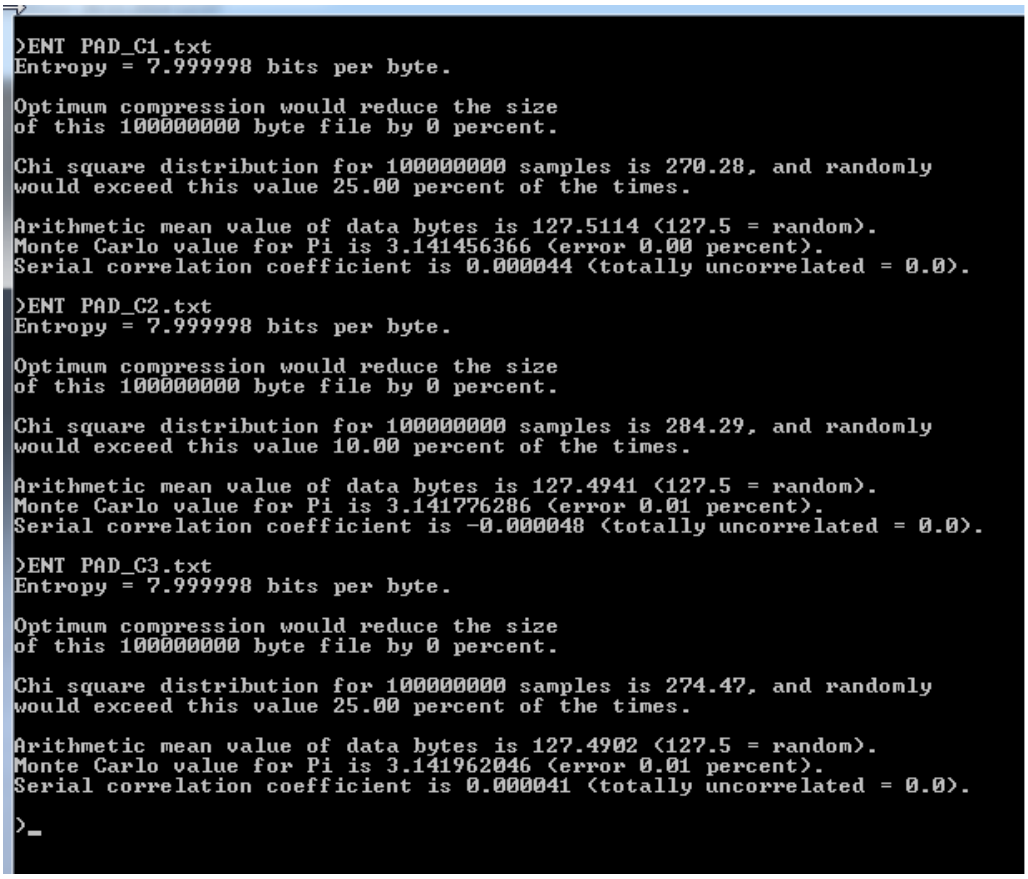
Next we tried the "ENT-Suite" on our 100 MB OTB's.
Here is the result.

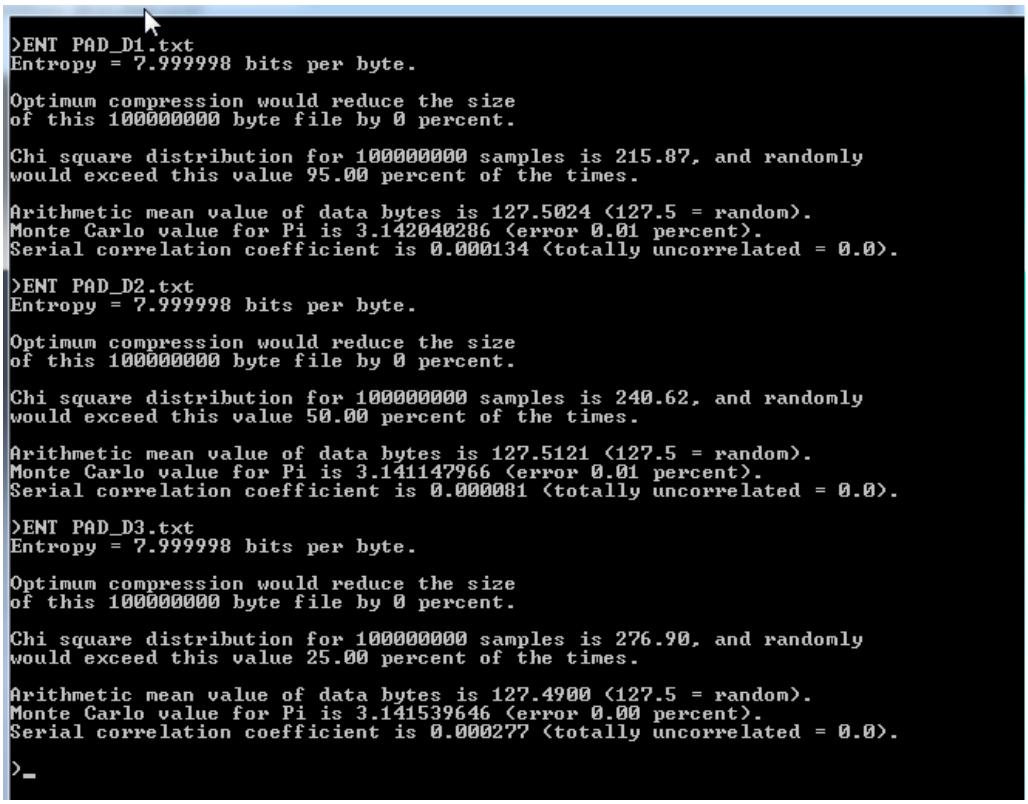
© 2023 Theo Gottwald * https://www.smart-package.com

```
>ENT PAD_A1.txt
Entropy = 7.999998 bits per byte.

Optimum compression would reduce the size
of this 100000000 byte file by 0 percent.

Chi square distribution for 100000000 samples is 262.98, and randomly
would exceed this value 50.00 percent of the times.

Arithmetic mean value of data bytes is 127.4912 (127.5 = random).
Monte Carlo value for Pi is 3.141722046 (error 0.00 percent).
Serial correlation coefficient is -0.000151 (totally uncorrelated = 0.0).

>ENT PAD_A2.txt
Entropy = 7.999998 bits per byte.

Optimum compression would reduce the size
of this 100000000 byte file by 0 percent.

Chi square distribution for 100000000 samples is 275.60, and randomly
would exceed this value 25.00 percent of the times.

Arithmetic mean value of data bytes is 127.5022 (127.5 = random).
Monte Carlo value for Pi is 3.141564846 (error 0.00 percent).
Serial correlation coefficient is 0.000046 (totally uncorrelated = 0.0).

>ENT PAD_A3.txt
Entropy = 7.999998 bits per byte.

Optimum compression would reduce the size
of this 100000000 byte file by 0 percent.

Chi square distribution for 100000000 samples is 277.06, and randomly
would exceed this value 25.00 percent of the times.

Arithmetic mean value of data bytes is 127.5102 (127.5 = random).
Monte Carlo value for Pi is 3.141658446 (error 0.00 percent).
Serial correlation coefficient is 0.000027 (totally uncorrelated = 0.0).

>
```

```
>ENT PAD_B1.txt
Entropy = 7.999998 bits per byte.

Optimum compression would reduce the size
of this 100000000 byte file by 0 percent.

Chi square distribution for 100000000 samples is 269.32, and randomly
would exceed this value 50.00 percent of the times.

Arithmetic mean value of data bytes is 127.5083 (127.5 = random).
Monte Carlo value for Pi is 3.141868686 (error 0.01 percent).
Serial correlation coefficient is 0.000063 (totally uncorrelated = 0.0).

>ENT PAD_B2.txt
Entropy = 7.999998 bits per byte.

Optimum compression would reduce the size
of this 100000000 byte file by 0 percent.

Chi square distribution for 100000000 samples is 223.41, and randomly
would exceed this value 90.00 percent of the times.

Arithmetic mean value of data bytes is 127.5009 (127.5 = random).
Monte Carlo value for Pi is 3.141755166 (error 0.01 percent).
Serial correlation coefficient is -0.000148 (totally uncorrelated = 0.0).

>ENT PAD_B3.txt
Entropy = 7.999998 bits per byte.

Optimum compression would reduce the size
of this 100000000 byte file by 0 percent.

Chi square distribution for 100000000 samples is 254.71, and randomly
would exceed this value 50.00 percent of the times.

Arithmetic mean value of data bytes is 127.5110 (127.5 = random).
Monte Carlo value for Pi is 3.141103086 (error 0.02 percent).
Serial correlation coefficient is 0.000017 (totally uncorrelated = 0.0).

>_
```


© 2023 Theo Gottwald * https://www.smart-package.com

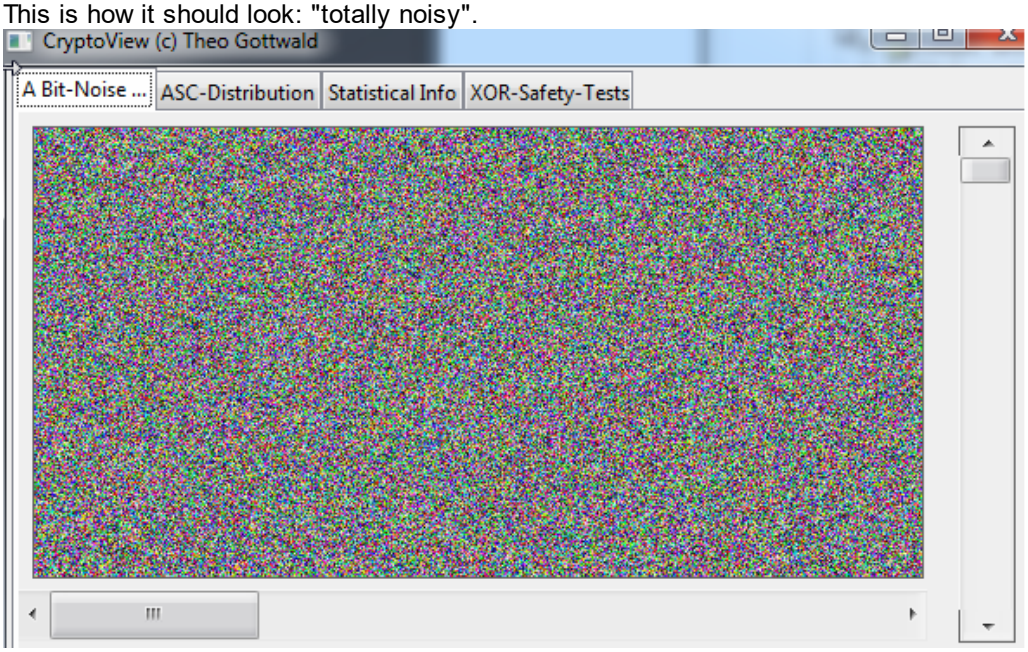




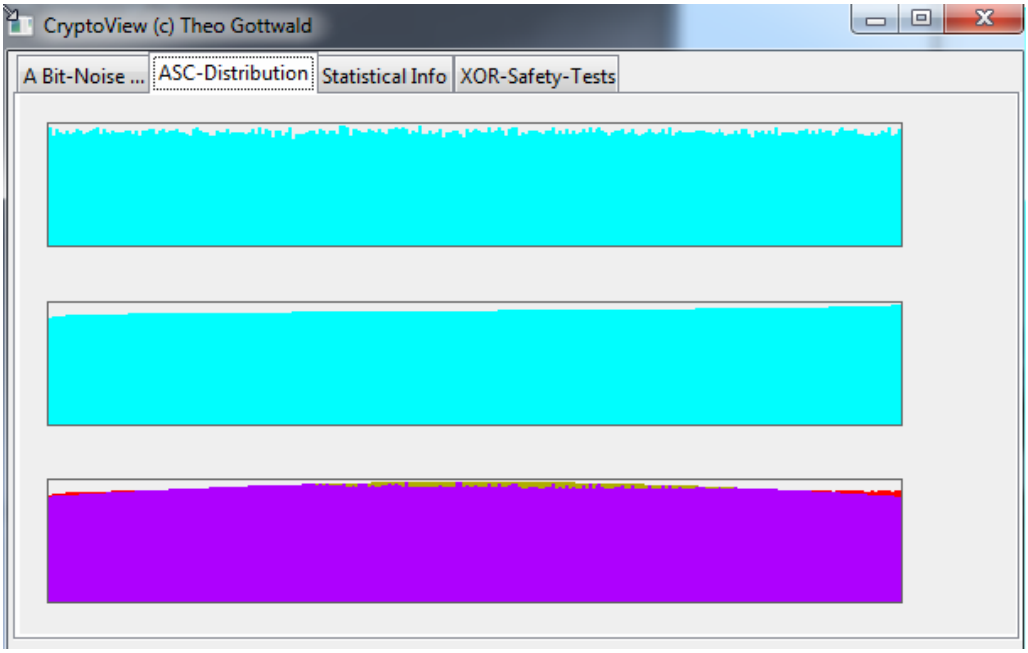
We will supply also a new tool together with this version of the SPR: "Crypto-View".
Using Crypto-View you can identify non perfect random distributions more easily.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Lets first take a look on our Pads in "Crypto-View".
This is how it should look: "totally noisy".

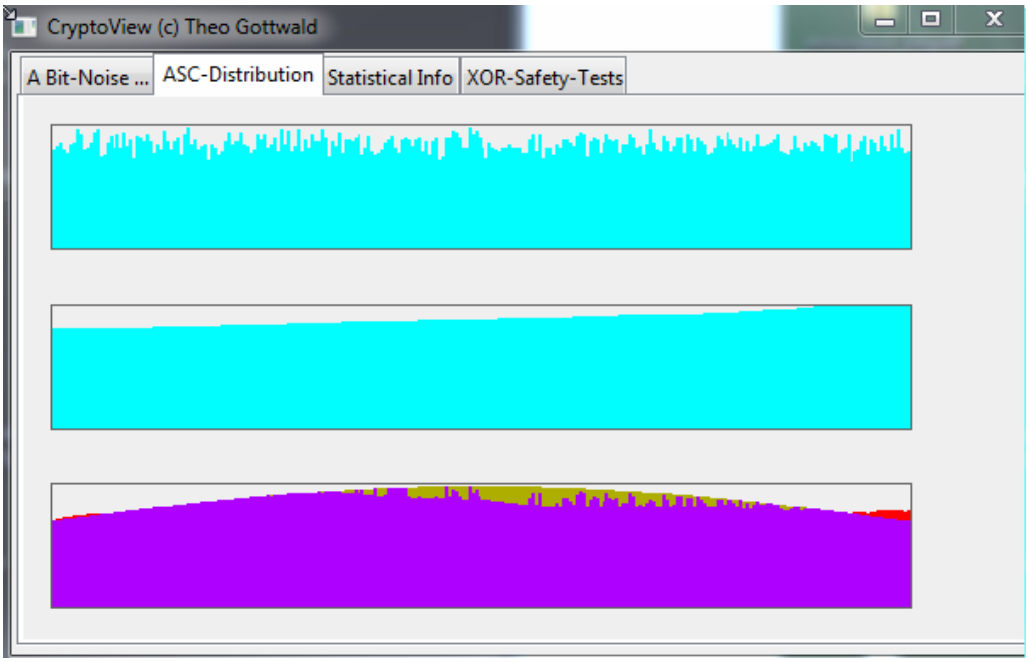


On the second TAB "ASC-Distribution", you should see a GAUSS-Curve in the third picture,
The larger the OTP-file, the less "round". Here are two examples of SPR generated OTP's.



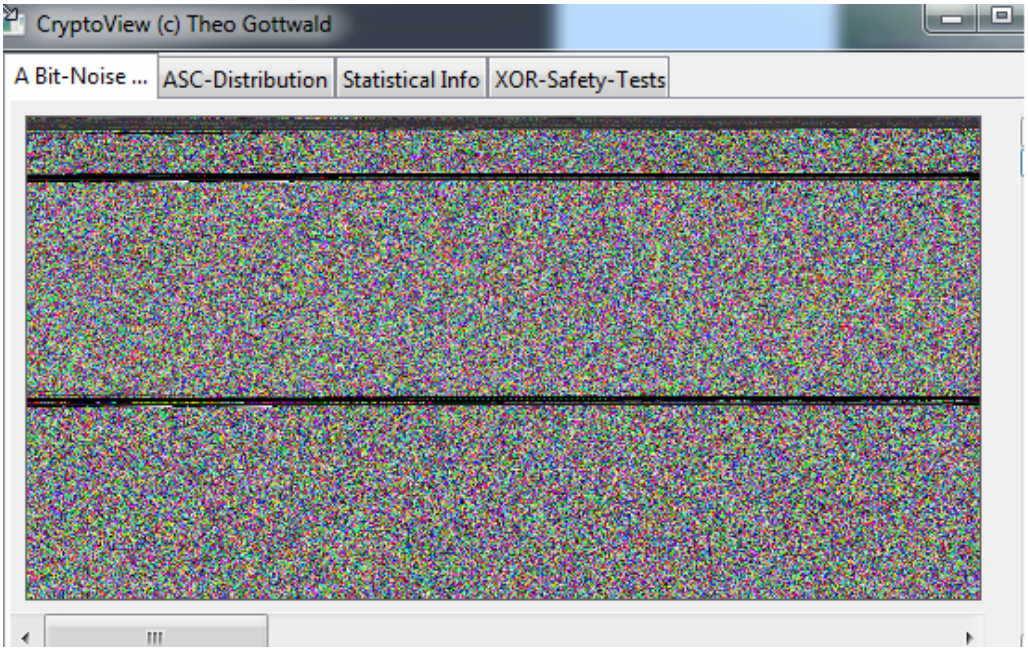
This is a "Crypto-View" from an OTP of about 500.000 Bytes.

© 2023 Theo Gottwald * https://www.smart-package.com



This is a "Crypto-View" from an OTP of about 50.000 Bytes. Of course with such smaller files, the distortion is more clear visible.

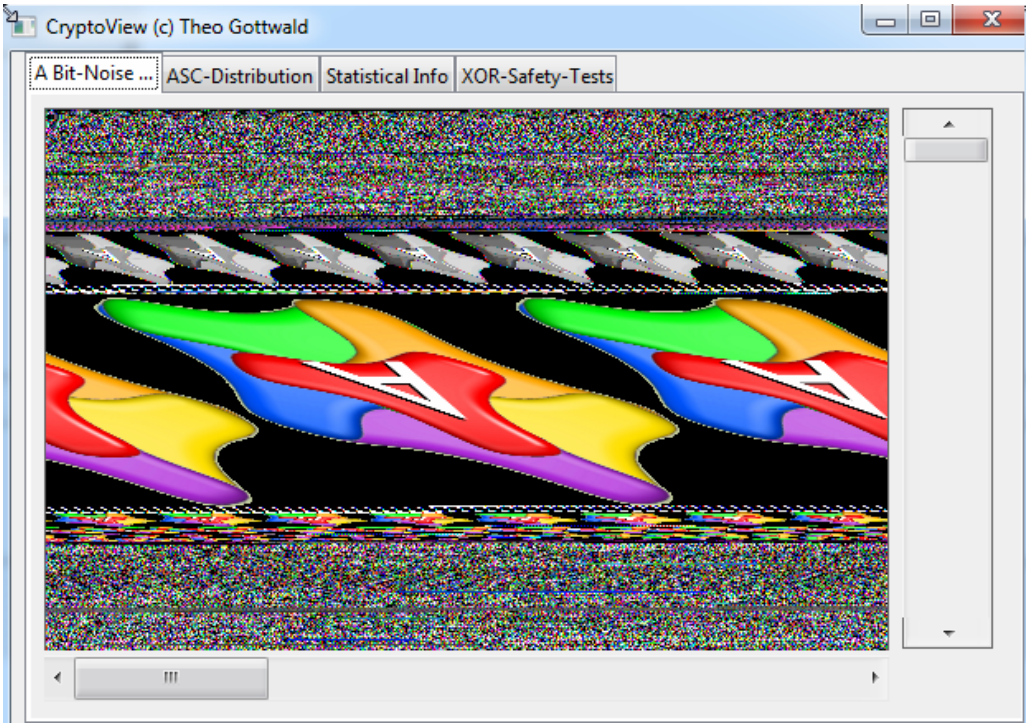
Lets take a look on a file that is not perfectly distributed. A ZIP-Archive for example. You can drag and Drop it on Crypto-View. Watch out for these horizontal lines. They tell you ... its something else.



This is a PDF-File.

© 2023 Theo Gottwald * https://www.smart-package.com

This is an executable. You see the "lines" sometimes more clear - sometimes less clear (with larger files). If you move the slider on the right side downwards, the colors may change and get a red, green or blue "touch". If that happens - its NOT RANDOM.



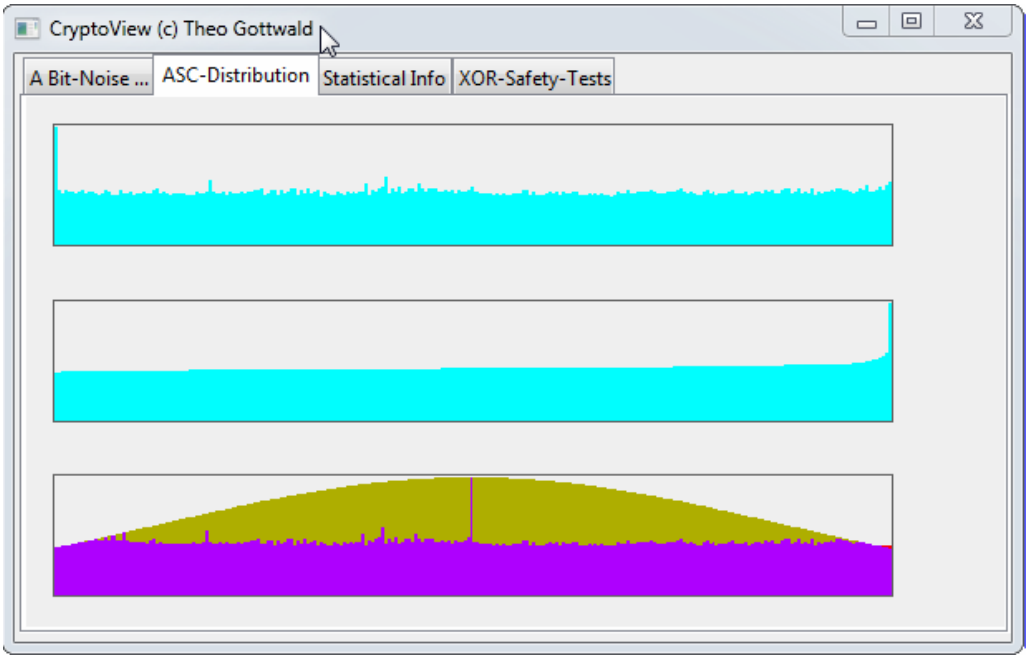
This is the Crypto-View of an SPR-Executable-File. As you can see the most space is used by the new modern Icon-Look. And the rest contains some horizontal lines, which identify it as "Non-random".

The easiest way to sort bad Random numbers out, is to watch for the Gauss-Curve.

What is a Gauss-Curve?

© 2023 Theo Gottwald * https://www.smart-package.com

You know that story.If you have a shop, long time no customer comes ... and then they come all at once. Just if they had an appointment at that time. Thats a so called "Gauss-bell".
Its just "random distribution". The same happens in true Random numbers.
With very large files you get more and more straight lines and the curve is not so good to see.
But you will always know when its a Gauss - or not.
For example ... this below is not a Gauss (red lines).
If its not a ... more or less ... Gauss ... its not a true Random File.



The SPR is a Cryptography Construction Set
You can generate your own Encryption-System using the SPR.
You can compile it and use it.

* NIST Guidelines for Media Sanitization

© 2023 Theo Gottwald * https://www.smart-package.com

3.50.2 GEC. - Get Encryption

MiniRobotLanguage (MRL)

GEC. Command

Get Encryption

Intention

Encrypts a string or a file, to a string or a file. And decrypts such a string or file. The usage on string is simple:

```
' Encrypt Variable into Variable
GEC.e1|$$TXT|$$KEY|$$TXT

'This way you can save the result as file immediately.
' Decrypt Variable into file
GEC.do1|$$TXT|$$KEY|$$FIL

' Here we encrypt a file
GEC.efo1|$$FIL|$$KEY|$$FIL

' Here we decrypt a file
GEC.dfo1|$$FIL|$$KEY|$$FIL

' This example encrypts a file into a variable
GEC.ef1|$$FIL|$$KEY|$$TXT
```

For details see example 2 below.

File-size of string or file is been changed to a multiple of 8. Optimal size for the password is 16 bytes. Longer password size will internally be shortened to 16 Bytes.

GEC. has been enhanced in the update 1/2015.
In the previouse implemtation of `GEC.`, Passwords wit ha length of less then 16 Bytes (128 Bit) have been autofilled with ASCII(32) (Space).
In the actual GEC.-Implementation this is been changed to a more difficult Bit-Pattern that is computed from the given Password.
Therefore, if you recompile "old Scripts" that have used short passwords, they may not be able to decrypt old data-files, until you fill the old Password with ASC 32.

Also there are many new Options with `GEC.` that make it to a simple to use - yet unbreakable Crypto-Construction Set.

Speed considerations:

© 2023 Theo Gottwald * https://www.smart-package.com

On a typical test-system, a file with **17.8 MB** is encrypted with lowest strenght in **0.3 seconds**. The decryption of the same file takes about 0.2 seconds. Note that for most use-cases, this level is all you will ever need.

The same file is been encrypted with standard strength (140 Rounds) in about 1.9 seconds. The decryption for that file takes 1.7 seconds.

The same file is been encrypted with highest strength (level 9) in about 12.8 seconds. The decryption for that file takes 12.3 seconds.

Syntax

GEC.P1|P2|P3[|P4]

Parameter Explanation

P1 **-** VAR/Prototypes
P2 **-** VAR with the String or the Filename to process
P3 **-** VAR with Key to en-/decrypt
P4 **-** ***(optional)*** VAR with Filename or Variable for the result, if omitted, result goes to TOS

Prototypes must be lowercase letters like following:
f -input from file - if f is not specified, the Variable is been resolved to a string.
e - "Encrypt". If **e** is not specified the input is been decrypted.
p - Use "one Time Pad" encryption (XOR) after using the TEA-Encryption.
In this case **P3** contains the "One Time Pad".
You can create "One Time Pads" using the STR.CHARS D etc. command.
When using the "**p**"- Option, the Passwort for the TEA-Encryption is taken out of the "One Time Pad".
q - Additionally use a "Byte-Flood"-Algo. This will additionally confuse the resulting Bit-Pattern.
o - output result to a file. if o is not specified, output is directed into the variable P3, or if none to the TOS.
k - kill-file. If specified, source-file is been deleted after encryption. If specified with variable, the variable is been cleared.
m - **max** Encryption. Multiply number of Rounds times 1000.
Specify this to use maximum strong encryption. If given, the normally used number of
rounds is multiplied by 1000.
For example:
m1 - normally 1 Round will then be 1000 Rounds.
m9 - 1024000 Rounds (don't worry its still very fast!
1-9 strength of encryption (if omitted, default = 140 Rounds).

Here are the details on the strength value:

© 2023 Theo Gottwald * https://www.smart-package.com

0 - Encrypt with just 1 Round. This is for testing and learning purpose, do NOT use it for encryption. Unless you use other algos after this.
1 - (10 Rounds) is already above so called "weak encryption"
2 to **8:** Multiply the number times 25 to get the number of rounds.
Example: 2 = 50 Rounds, 8 = 200 Rounds
9 - Use 1024 Rounds (maximum strength)

As we have **complete diffusion** after only 6 rounds, even the lowest Level of "**1**" is already above "weak encryption" and strong enough for non-professional use. The strongest level has 1024 Rounds.

Example

```
'*************************************************************
*****
' GEC-Sample 1
'*************************************************************
*****
' The following line contains the password
' for encryption and de-cryption
: $$KEY=My_Password
: $$FIL=?desktop\Mysecret.txt
: $$TXT=This text is going to be hidden!$crlf$Even this line
is a secret!

FOR.$$MUC|1|18
  VAR.$$TXT=$crlf$ $$TXT $crlf$ $$TXT
NEX.
LEN.$$TXT|$$SIZ
MBX.We are now going to encrypt a string of size: $$SIZ to a
file on your desktop.$crlf$The filename is:$$FIL
VAR.$$TIM=#dtime#
' Encrypt Variable into file with highest strength
GEC.eo|$$TXT|$$KEY|$$FIL
MBX.The file (Size: $$SIZ) was successfully encrypted with
high strength.$crlf$Take a look on it before we de-crypt it
again.$crlf$ Time needed for encryption: #dsince# seconds.
' Now we decrypt the file from desktop to itself
VAR.$$TIM=#dtime#
GEC.dfo|$$FIL|$$KEY|$$FIL
MBX.The file was successfully decrypted.$crlf$Take a look on
it before we encrypt it again.$crlf$Time needed for
decryption: #dsince# seconds.
VAR.$$TIM=#dtime#
GEC.efo|$$FIL|$$KEY|$$FIL
MBX.The file (Size: $$SIZ) was successfully encrypted with
high strength.$crlf$Take a look on it before we de-crypt it
again.$crlf$ Time needed for encryption: #dsince# seconds.
VAR.$$TIM=#dtime#
GEC.dfo|$$FIL|$$KEY|$$FIL
MBX.The file was successfully decrypted.$crlf$Take a look on
it before we encrypt it again.$crlf$Time needed for
decryption: #dsince# seconds.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*************************************************************
*****
' GEC-Sample 2
'*************************************************************
*****
' The following line contains the password for
' encryption and de-cryption
: $$KEY=aaaaa
: $$FIL=?desktop\Mysecret.txt
: $$TXT=aaaaaaaaaaaaaaaaaa

' Encrypt Variable into Variable
GEC.e1|$$TXT|$$KEY|$$TXT
MBX.The text was successfully encrypted.$crlf$+$$TXT
' Encrypt Variable into file with highest strength
GEC.do1|$$TXT|$$KEY|$$FIL
MBX.The text was successfully decrypted to file $crlf$ $$FIL.
GEC.efo1|$$FIL|$$KEY|$$FIL
MBX.The file was successfully encrypted.
GEC.dfo1|$$FIL|$$KEY|$$FIL
MBX.The file was successfully decrypted.
GEC.efo1|$$FIL|$$KEY|$$FIL
MBX.The file was successfully encrypted to File.
GEC.df1|$$FIL|$$KEY|$$TXT
MBX.The File was successfully decrypted to:$crlf$$$TXT.
ENR.
```

Remarks

GEC. is provided to enable you, to hide data, like Passwords and such, with a cryptographic algo, that is tested to make your data unreadable.
From a scientific standpoint, the use of GEC. together with the OTP (One-Time-Pad) Option "**p**", the result should be unbreakable.
For more Details see ! Smarty's Encryption and Hashing Commands 812.

While giving the best we can provide, we can not take any responsibility for safety of your encrypted data. Computers get faster, science makes progress. There is no warranty that data being encrypted today can be seen as safe tomorrow.
Therefore use GEC. on your own risk.
Note that even if you specify the "**k**"-Option, which will delete a file after overwriting it 40 times with Byte-Patterns, the original file may under unknown circumstances still be undeleted by professional undelete experts, especially when it was on a SSD-Drive.

To use the encrypted characters in Chat-Programs or e-mail, do not forget to use the STR.TEXT TO MIME 3215 before,
Of course the encoding must be reversed on the other side before decryption.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

1. File-Size
The actual Implementation tries to load the complete file into Memory before applying the Encryption. Therefore Files with a size that exceeds the Systems RAM Capabilities can not be encrypted or decrypted.
In case there are File-Errors, an empty string is returned.
Please note that the smallest possible Encryption-Result is always 8 bytes or a multiple of 8 bytes. After Decryption the File/string will have its original size, that it the size it had before being encrypted.

2. Encryption Limitations:
TEA* - Tiny Encryption Algorhytmus
Stats of the TEA*-Encryption:
· Block Symmetric Feistel cipher.
· 128 bit key
· Encrypts 64 bits at a time
· Highly resistant to differential cyptanalysis
· Achieves complete diffusion after 6 rounds
- one bit difference in plaintext will cause approximately 32 bit differences in ciphertext

The used TEA*-Encryption is very fast and can be seen as safe for normal "house use". From seven "Encryption-Rounds" up it is already above so called "weak Encryption". As each "Prototype" value counts for 25 Encryption rounds, this is already the case for our lowest Encryption level (level "1"). Because even in level "1" we make 10 internal "Encryption Rounds".
While being already safe for "house use", it may be vulnerable to professionals via "related-key attacks". To prevent this, it should be used with keys that are generated by a strong Random Number Generator, or by a source of bits that are sufficiently uncorrelated (such as the output of a hash function).
Even with a non-perfect key, the prototype "4" (internally= 100 Encryption rounds) should cure most sins, while still being very fast.

For more in Detail mathematical information see "Block Ciphers and Cryptanalysis".

3. CPU-Limitations
For best possible Speed, MMX was used. Therefore the results on "non-MMX CPU's are unpredictable.

*** Developed by David Wheeler and Roger Needham at Cambridge University**

See also:

- STR.CHARS A 3155
- STR.CHARS B 3159
- STR.CHARS C 3163
- STR.CHARS D 3167
- HIT. - Hide Text 835
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840

© 2023 Theo Gottwald * https://www.smart-package.com

- STR.GENERATE - Mode 3188
- STR.FROM HEXASCI 3185
- STR.TO HEXASCI 3222
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- FIL.Split File 1001
- FIL.Join File 994
- FIL.Join Delete 990

© 2023 Theo Gottwald * https://www.smart-package.com

3.50.3 GMD. - Get Message Digest

MiniRobotLanguage (MRL)

GMD. Command

Get Message Digest

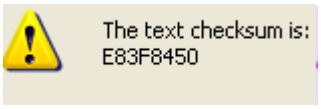
Intention

This command returns the checksum of a string or a file. If you are not sure what a checksum is, or what it is good for, read this WEB-Link: Checksum.

Lets start with the checksum of a string:

```
: $$TXT=Smart
GMD.|$$TXT|$$TXT
MBX.The text checksum is:$crlf$$$TXT
```

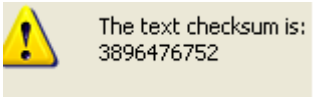
For this simple operation, no prototypes are needed.
The result is a hexadecimal checksum (see picture below) that used an FNV32 algo.
This is the default algorithm, if no other algorithm is selected.



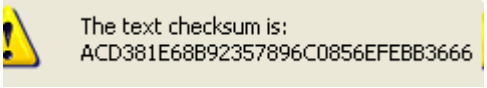
We can give some prototypes to change the checksum algo or the output format.
Like this:

```
GMD.d|$$TXT|$$TXT
```

The result is the same number as decimal number. See following picture.



Now lets choose another checksum algo. Lets try MD5.



MD5 will always return such a long hexadezimal string. It will ignore other options.

Next, lets take a checksum from a file. It's as simple as before. We just give the prototype "f" and the first variable contains the filename.

```
: $$FIL=?desktop\New_Update.zip
GMD.vf|$$FIL|$$TXT
```


© 2023 Theo Gottwald * https://www.smart-package.com

This way you can easily make a file integrity check of any file!

Now let's do some ...

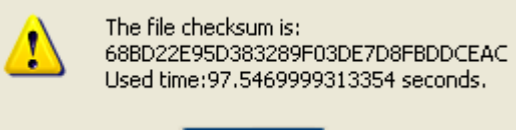
Speed considerations:

For a file with a size of >100 MB i get a checksum-time of about 0.3 seconds.

```
' Checksum-time FNV32: 0.3 seconds
GMD.vf|$$FIL|$$TXT

' Checksum-time Elf32: 0.5 seconds
GMD.ef|$$FIL|$$TXT

' Checksum-time CRC-32: 0.5 seconds
GMD.cf|$$FIL|$$TXT
```

As we can see, this checksums does not cost you a lot of time. But how about MD5?

```
' Checksum-time MD5: 97.5 seconds
GMD.mf|$$FIL|$$TXT
```



MD5 may eat a full second per Megabyte, depending on the speed of your system. But even this is a very good time!

Now this speed can only be reached, because we hold the complete data in computer memory in one piece.

For very large files (>500 MB? or >1 GB? this depends on your memory situation) this is not possible. If you need checksums of large files, choose the option "b" or for special uses "s".

```
' Get Checksum from Textfile
VAR.$$TIM=#dtime#
GMD.s|$$FIL|$$TXT
MBX.The file checksum is:$crlf$$$TXT$crlf$Used time:#dsince#
seconds.
ENR.
```

Syntax

GMD.P1|P2[|P3]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - VAR/Prototypes

```
Prototypes must consist of one or two letters.

The first letter shall be:
f - input from file, if omitted input is taken from variable
b - file is bigger then System RAM, choose slower "block-by
block"
    algorithm. This option (unlike f or s) does not support
    a second letter.
s - calculate a string checkum from a long file. This Option
is
    similar to "b" but the result is much more exact. Result
is
    a string with a checksum for each 8192 characters of the
    source-file.  This option (unlike f or s) does
    not support a second letter. It may return a large
    checksum-string.

All of the following prototypes will be ignored when options
"b" or "s" are used:

v - use FNV32 Hash (default if omitted).
c - use CRC32 Hash.
e - use Elf 32 Hash.
m - use the MD5 checksum (slower, more safe). Result is a
string.


All of the following prototypes will be ignored when option
"s" or "m" is used:

h  - Hexadecimal Output as string (default if omitted)
o  - Octal Output as string
d  - Decimal Output
i  - Binary Output as string
```

P2 - VAR with the String or the Filename to process
P3 - optional VAR for Result, if omitted TOS is used

Example

-

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

File-Size
File-Size is limited by System RAM, and generally to a maximum of 2 GB (32 bit), unless the option **b** or **s** is specified. These Options should be able to handle any possible file-size (64 bit).

See also:

- GEC. - Get Encryption 825
- SHA. - Safer Hash Algo 840
- STR.GENERATE - Mode 3188
- STR.FROM HEXASCI 3185
- STR.TO HEXASCI 3222
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198

© 2023 Theo Gottwald * https://www.smart-package.com

3.50.4 HIT. - Hide Text

MiniRobotLanguage (MRL)

HIT. Command

Hide Text using the "Byte-Flood" Algo

Intention

This command offers a very basic encryption, by XOR'ing a Text with an internal 250 Byte hardcoded Pad. Also it will do an XOR with the neighbour Byte ("Byte Flood") and therefore make things definitely unreadable.
The advantage is that it does not need a Password, because it uses the internal Pad and the Data itself as "Key".
Note that this algo alone is not safe encryption, because anybody who owns the SPR can decrypt it!
It shall only be used in combination with other commands like `XOR` and `GEC`..
It can also be used if you just want to hide something and make it unreadable.
I recommend this algo as a part of a private encryption scheme, not as a standalone encryption.
You can influence the number of rounds by choosing an appropriate selection of prototypes. Unlike with XOR. - XOR - Encryption 843, the "Byte Flood" encryption and decryption use separate algos, therefore you need to choose "**d**" or "**e**" depending on what you are doing.

Here is an example, before and after using :

```
Danke für Ihren Auftrag.  Ihre Transaktionsnummer (MTCN) ist: 1559298090
Wir sind gerade dabei, die Transaktion mit Ihrer Bank abzuschließen.
Sobald sie durchgeführt wurde, senden wir eine Bestätigung an Sie.
```

We use a Script that produces an encrypted file:

```
$$FIL=?path\Western.txt
$$OUT=?path\WesternC.txt
HIT.efo0|$$FIL|$$OUT
```

```
i...nw2¥=pJ"Ä·ð¾÷¶¬[ÿF/ß›ñÍ;...MîWD¹-]ó¢Ø¯öØ°‹,Á1  ›KóÎÔÿwŽsmnÈ†I6×þ
```

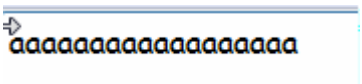
This is just a part of the result. And its definitely unreadable.
This way you can send the Western Union code safely.

While do not use it for the new important new developments of your company.
Because anybody who owns an SPR can decrypt it. This is the decryption script:

```
$$NEW=?path\Receiver.txt
HIT.dfo0|$$OUT|$$NEW
```

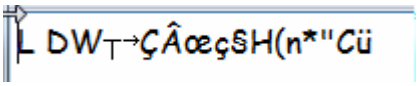

*© 2023 Theo Gottwald * https://www.smart-package.com*

HIT. will not fail fail even on diffcult Plain-Texts, due to the used hardcoded Pad.
Therefore in the actual implementation, 0 rounds is completely safe.
A higher number of Rounds produce other results.
If you use more then 0 rounds, check the result. For example, take this Plain-Text:



Crypted with a Single Round HIT., it will look like this:

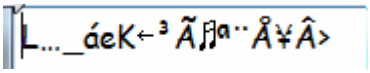
```
$$FIL=?path\Difficult.txt
$$OUT=?path\Crypted.txt
HIT.0ef|$$FIL|$$OUT
```



Does not leave any wishes open so far. Lets see what happens if we increase number of rounds. Now the result from this:

```
HIT.efo2|$$FIL|$$OUT
```

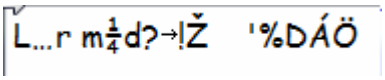
looks like this:



A simple

```
HIT.efo2h|$$FIL|$$OUT
```

would have this result:



Generally HIT. is an additional tool to quickly make stuff unreadable.
Its not an unbreakable encryption.

Back to decryption. Note, that we have used the "2h" and that if we use it for encryption,
we must use the same parameters for decryption!

```
' We decrypt it like this
$$NEW=?path\Receiver.txt
HIT.2hdfo|$$OUT|$$NEW
ENR.
```

Looks like HIT. can get a good result with difficult Plain-Texts.

To get a much better result, we can add a fix Pattern in front of the message.
The longer, the better.

© 2023 Theo Gottwald * https://www.smart-package.com

Doing this, the result can only easily be decrypted from somebody who knows that Prefix.
Is it safe?
Possibly not. But its good for hiding Text. Thats why the name os "HIT."
Here is an example:

```
' We encrypt "file to Variable"
$$FIL=?path\Difficult.txt
$$OUT=?path\Crypted.txt
$$PRE=1-+rD
CFF.$$FIL|$$PLA
' Add a Prefix for better encryption
VAR.$$PLB=$$PRE$$PLA
HIT.1eo|$$PLB|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
' We encrypt "file to Variable"
HIT.1df|$$OUT|$$PLC
' Remove $$PRE from Result
GSS.6|0|$$PLC|$$PLD
' Save it to a file
CTF.$$NEW|$$PLD
ENR.
```

On the other side ... who needs a Message like "aaaaaaaaaaaaaaaa"?

If you have difficult Plain-Texts, i recommend to combine multiple systems, for example GEC. and XOR. (one time Pad).
HIT. is for hiding unimportant stuff, where you do not need to remember a Password.

Syntax

HIT.proto|P1[|P2]

Parameter Explanation

P1 - VAR, Filename or String to encrypt.

P2 - *(optional)* - VAR for Return value. If omitted, TOS is used.

The following prototypes can be used:

£ - if given, then **P1** is a filename, not a Text-String. If omitted, the content of **P1** is taken.

o - if given, then **P2** is a filename, not a Variable for ther result.
If omitted, the content of **P2** is treated as a variable where the result shall be stored.
If **P2** is omitted, then the result is been placed on TOS.
If "**o**" is specified, then a filename for the result-file must be given!

© 2023 Theo Gottwald * https://www.smart-package.com

z - multiply number of rounds times 10.
m - multiply number of rounds times 100.
h - multiply number of rounds times 500.
e - signals that the encryption-algo shall be used.
d - signals that the decryption-algo shall be used.

Numbers from 0 to 9

0-9 You can specify numbers from 0 to 9 to specify the number of rounds that should be done.

Example

```
'******************************************************
' HIT. - Sample
'******************************************************
' We encrypt "file to file"
$$FIL=?path\Western.txt
$$OUT=?path\WesternC.txt
HIT.efo|$$FIL|$$OUT

' We decrypt it like this
$$NEW=?path\WesternD.txt
HIT.dfo|$$OUT|$$NEW
ENR.

'******************************************************
' HIT. - Sample #2
'******************************************************
' We encrypt "file to Variable"
$$FIL=?path\Difficult.txt
$$OUT=?path\Crypted.txt
$$PRE=1-+rD
CFF.$$FIL|$$PLA
' Add a Prefix for better encryption
VAR.$$PLB=$$PRE$$PLA
HIT.1eo|$$PLB|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
' We encrypt "file to Variable"
HIT.1df|$$OUT|$$PLC
' Remove $$PRE from Result
GSS.6|0|$$PLC|$$PLD
' Save it to a file
CTF.$$NEW|$$PLD
ENR.
```

Remarks

To use the encrypted characters in Chat-Programs or e-mail, do not forget to use the STR.TEXT TO MIME[3215] before,
Of course the encoding must be reversed on the other side before decryption.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

Possible maximum file-Size is limited by System RAM, and generally to a maximum of 1-2 GB (32 bit).

See also:

- XOR. - XOR - Encryption 843
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- STR.GENERATE - Mode 3188
- STR.FROM HEXASCI 3185
- STR.TO HEXASCI 3222
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- FIL.Split File 1001
- FIL.Join File 994
- FIL.Join Delete 990

© 2023 Theo Gottwald * https://www.smart-package.com

3.50.5 SHA. - Safer Hash Algo

MiniRobotLanguage (MRL)

SHA. Command

Get Message Digest SHA I from String or File

Intention

This command returns the checksum of a string or a file. If you are not sure what a checksum is, or what it is good for, read this WEB-Link: Checksum.

Using `SHA.` you can currently choose between:

- **SHA** Details SHA
- **MD5** Details MD5

Usage is very simple:

```
' Name and Path of the file
$$FIL=?desktop\MyFile.exe
SHA.$$FIL
' Display result
DBP.$$000
```

You can use a variable instead of the TOS:

```
' Name and Path of the file
$$FIL=?desktop\MyFile.exe
' $$RES will contain the SHA checksum
SHA.$$FIL|$$RES
' Display result
DBP.$$RES
```

If you prefer to get the MD4 checksum:

```
' Name and Path of the file
$$FIL=?desktop\MyFile.exe
' $$RES will contain the SHA checksum
SHA.$$FIL|$$RES|m
' Display result
DBP.$$RES
```

There are some more built in Options that influence the outlook of the result. You can:

- make the result string uppercase (use U in **P3**)
- make the result string lowercase (use L in **P3**)
- choose to have spaces for better readability when using `SHA` I, like this:
  `EC91328073 A651B13403 BA5B2D9730 861F312F88`
- choose the MD5 Algo instead, specify "`M`" as **P3**

*© 2023 Theo Gottwald * https://www.smart-package.com*

- Prefix **P1** with the **<** Sign. In this case the String will be taken as Binary Data, instead of as Filename

Syntax

SHA.P1[|P2][|P3]

Parameter Explanation

P1 - VAR, Filename or String to calculate the checksum for. If the first letter is "<" then
input is taken from following variable using binary, one time resolution.
Otherwise P1 is taken as a filename/path for file that should be loaded
and processed.

P2 - *(optional)* - VAR for Return value. If omitted, TOS is used.

P3 - *(optional)* - any combination of these letters:
M - use the MD5 checksum. Result is a string of length 32.
L - Lowercase Result
U - Uppercase Result
S - Insert Spaces for better readability into SHA I Output.
This flag has no influence on the MD5 output.

Example

```
' Load the file and calculate SHA I checksum
$$FIL=?pfad\Sha1.exe
SHA.$$FIL|$$RES|s
DBP.$$RES
ENR.
```

Remarks

-

Limitations:

Possible maximum file-Size is limited by System RAM, and generally to a maximum of 2 GB (32 bit).

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- **GEC. - Get Encryption** 825
- **GMD. - Get Message Digest** 831
- **STR.GENERATE - Mode** 3188
- **STR.FROM HEXASCI** 3185
- STR.TO HEXASCI 3222
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.50.6 XOR. - XOR - Encryption

MiniRobotLanguage (MRL)

XOR. Command

XOR Encryption

Intention

Logical operations on Bit-Level. Combines/Encrypts or decrypts Bit-Patterns.
Works from a string or a file, to a string or a file.

While the name of the Command is just "**XOR.**" this command knows all possible encryptions-on-bit level.

You can choose from the following logical operations to encrypt or decrypt any Plain-Text on Bit-Level:

- XOR (default option or "**x**")
- ADD/SUB (Option "**a**")
- ROL/ROR (Option "**s**")
- Byte-Flood

And you can even combine all of them just as you like in a single command.
This command uses a "Pipelined architecture" therefore you can apply all of the above options with a single command. Then all Options will be applied with the same KEY on to the same Clear-Text. One after the other, In the right Order - which is important for decryption.

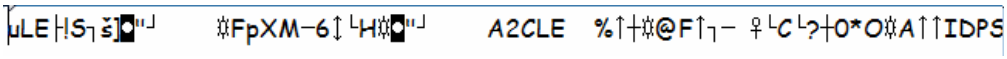
We will take a look on what happens with a Message when being decrypted.

> Danke für Ihren Auftrag. Ihre Transaktionsnummer (MTCN) ist: 1559298090
> Wir sind gerade dabei, die Transaktion mit Ihrer Bank abzuschließen.
> Sobald sie durchgeführt wurde, senden wir eine Bestätigung an Sie.

```
$$FIL=?path\Western.txt
$$OUT=?path\Crypted.txt
$$KEY=1-+rDsdf/(klsdaf
' We encrypt "file to file"
XOR.efo|$$FIL|$$KEY|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
XOR.dfo|$$OUT|$$KEY|$$NEW
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```



To make sure that we do not have a pure "character substitution", the command offers additional features (Byte-Flood).

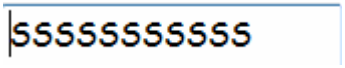
This will help especially with short keys, if there is a chance that somebody will try brute force to crack your Cryptography.

`XOR.` with a weak and short Password is far from being a safe method.
It becomes safe, in combination with a "One Time Pad" that has the size of the code that is to be encrypted.

Let me show this to you with an example:

```
$$OUT=?path\Crypted.txt
$$KEY=1
$$TEX=bbbbbbbbbbb
' We encrypt "file to file"
XOR.eo|$$TEX|$$KEY|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
XOR.dfo|$$OUT|$$KEY|$$NEW
ENR.
```

Will produce this crypted text:

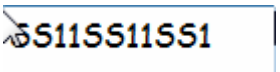


you can still make assumptions on the source-code.
Of course this will get better if we use a more complex `$$KEY`.

But first lets try this addition to XOR:

```
$$OUT=?path\Crypted.txt
$$KEY=1
$$TEX=bbbbbbbbbbb
' We encrypt "file to file"
XOR.3eo|$$TEX|$$KEY|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
XOR.3dfo|$$OUT|$$KEY|$$NEW
ENR.
```

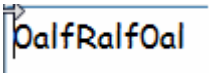
The result we get is this, and this is at least a few bits better.



© 2023 Theo Gottwald * https://www.smart-package.com

Note that there is a chance that Passwords can become visible when using XOR. and Bit-Flood in an unlucky way and with difficult Plain-text. Like in this case:

```
$$OUT=?path\Crypted.txt
$$KEY=Ralf
$$TEX=bbbbbbbbbbb
' We encrypt "file to file"
XOR.2eo|$$TEX|$$KEY|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
XOR.2dfo|$$OUT|$$KEY|$$NEW
ENR.
```



Therefore check the result of your operations in a HEX-Editor.
Or better use only cryptographical safe Passwords that make no sense when using `XOR.`
Of course the best way is to first use `GEC.` and then add up something else.
Best use a real "One Time Pad" 812.

The usage on strings is simple:

```
' Combine $$TXT and $$KEY
XOR.e|$$TXT|$$KEY|$$TXT

' This way you can save the result as file immediately.
' Combine Variables into a file
XOR.do|$$TXT|$$KEY|$$FIL
XOR.efo|$$FIL|$$KEY|$$FIL

' This example loads a file and places
' the result into a variable
XOR.ef|$$FIL|$$KEY|$$TXT
```

What happens, if the $$KEY is shorter or longer then the $$SRC
In either case, the XOR COmmand will handle the situation automatically.
To long `$$KEY's` will be shortened internally before beeing used. And if $$KEY is shorter then t$$TXT, the `$$KEY` will be repeated as often as needed to fill the gap.
For example:

```
$$TXT=Hallo Leute
$$KEY=12
' Then XOR will repeat $$KEY and use internally:
$$TXT=Hallo Leute
$$KEY=12121212121
```

While this will definitely make the text unreadable, it is far from bullet-proof encryption.

XOR-Encryption alone may under several circumstances not lead to professional security, therefore we recommend to use the **GEC.**-Command for encryption.

© 2023 Theo Gottwald * https://www.smart-package.com

The XOR. can be used to combine Random Numbers of different sources to a single result.
If you just want to hide something, you can use the HIT.-Command.

Do multiple XOR's make the result more secure?
No, as mathematically all XOR's can be shortened to a single XOR Operation.
Its just like an addition ... many additions can be shortened to a single addition.
However if you additionally use the ADD/SUB or ROL/ROR and the "Byte-Flood", the result will be hard to beat.

How can i make a definitely scientifically proven safe encryption that is unbreakable?
The only way is to use a so called "One-Time Pad". That is a "Key-File" with Random Data Bytes, that is only available to you, and tot the receiver of the message.
Then I recommend to use first GEC. on any important Plain-Text Message - just to make a statistical Binary from your Plain-Text.
And after that apply XOR. with some of its Options and a "Random Pad Key" that you have generated using the STR.CHARS D command. Doing so the result is mathematically unbreakable as long as the sender doesn't get hands on the decryption key.
Then make 500 Pads on a USB-Stick and give the stick to your communication partner.
So he can use every day a new "Pad" and delete the old Pad.
This way the message can never in the future be encrypted any more.
To help you about this we have now a included a "Safe-Erase" for "Pads" in the DEL.-Command. It will overwrite the Pad 40 times with Random Data before deleting it.
Making as sure as possible that your messages will stay private to you and the receiver.
What is most important when encrypting with XOR?
The most important point is the quality of the used PRNG for the Encryption Key.
For Details about it see ! Smarty's Encryption and Hashing Commands 812

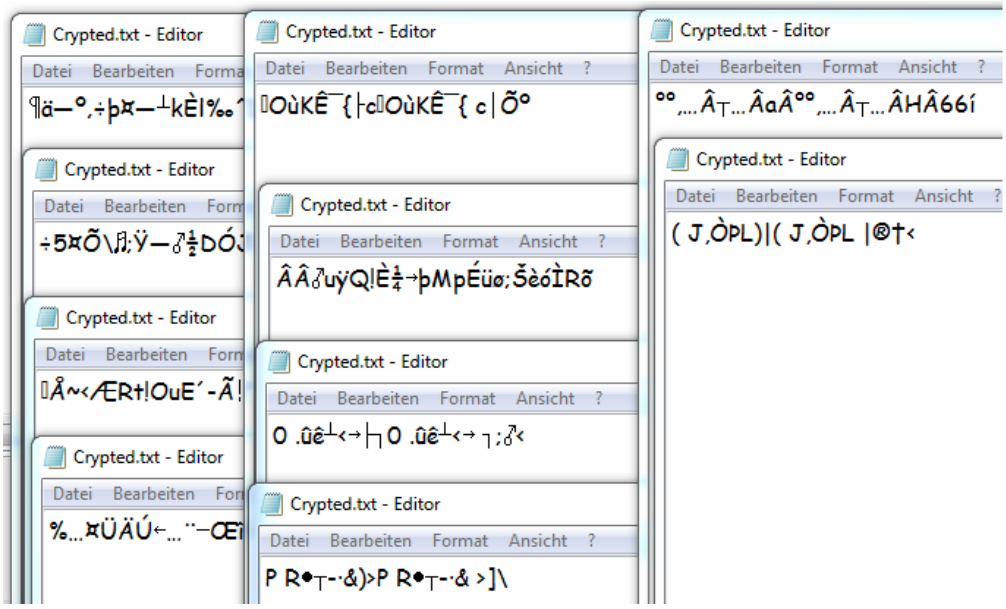
Making your own Encryption System
Using XOR. and GEC. the SPR is a "Crypro-Construction Set".
Making things as easy for you as possible. However there is something for you, to care about. In case you want to construct your own Encryption/Decryption using multiple
XOR.-Commands please note the following rule:

```
Plain-Text -> XOR.KEY1 -> XOR.KEY2 -> XOR.KEY3 -> Ciphertext
```
must be encrypted in reverse Order like this:
```
Ciphertext -> XOR.KEY3 -> XOR.KEY2 -> XOR.KEY1 ->Plain-Text
```

You only need to worry about this if you use multiple XOR./GEC. Encryption Commands in a chain. If you use just one command with multiple Options, they will internally be sorted in a way that it will work as expected.
By varying the prototypes between the given options, you always get different Ciphertexts. Below some Ciphertexts all from the same Plain-Text. All with just one XOR.-Command.
Technically the "**d**" - decrypt - Option is always the reverse of the "**e**" - encrypt option.

© 2023 Theo Gottwald * https://www.smart-package.com

Therefore you can also use "**d**" for encryption and in this case "**e**" for decryption.
The ciphertext will in this case look different from the normal results.



This picture shows some possible Ciphertexts, all from the same Plain-Text. All with just one `XOR.`-Command and changed prototypes.

What is most important when encrypting with XOR?
The most important point is the quality of the used PRNG (Noise-Overlay/ the One-Time-Pad) for the Encryption Key.
For Details about it see ! Smarty's Encryption and Hashing Commands 812

Speed considerations:
On a typical test-system, a file with **17.8 MB** is encrypted with lowest strenght in **0.3 seconds**. The decryption of the same file takes about 0.2 seconds. Note that for most use-cases, this level is all you will ever need.

The same file is been encrypted with standard strength (140 Rounds) in about 1.9 seconds. The decryption for that file takes 1.7 seconds.

The same file is been encrypted with highest strength (level 9) in about 12.8 seconds. The decryption for that file takes 12.3 seconds.

Syntax

XOR.P1|P2|P3[|P4]

Parameter Explanation

P1 – VAR/Prototypes
P2 – VAR with the String or the Filename to process

© 2023 Theo Gottwald * https://www.smart-package.com

P3 – VAR with Key to en-/decrypt
P4 – (optional) VAR with Filename or Variable for the result, if omitted, result goes to TOS

Prototypes must be lowercase letters like following:
f -input from file - if f is not specified, the Variable is been resolved to a string.
e - "Encrypt". If **e** is not specified the input is been decrypted.
o - output result to a file. if o is not specified, output is directed into the variable P3, or if none to the TOS.
s - (Shift Operation) "ROR"/"ROL" - used together with "**e**" it will use the "ROR" (Rotate Right) algo. Used together with "**d**", this will use the "ROL" (Rotate Left) algo. "ROR" and "ROL" are complementary operations. Each can be used to decrypt the other, or to encrypt for the other operation.
a - "ADD"/"SUB" - used together with "**e**" it will use the "ADD" algo. Used together with "**d**", this will use the "SUB" algo. "ADD" and "SUB" are complementary operations. Each can be used to decrypt the other, or to encrypt for the other operation.
x - (default if no "**s**" or "**a**" is specified. Will use XOR-Encryption/decryption. If "**s**" or "**a**" was specified, then "**x**" must be given to additionally use XOR.
Numbers from 1 to 4 add the "Byte-Flood Algo".
This will additionally shift Bytes around to confuse the result.
While for normal XOR-Operation, the Use of "e" is optional, in case of using "Byte Flood", the use of "d" or "e" are necessary to get the wanted result.

0 - Do a single "Byte-Flood-Round".
1 - Do as Many Rounds as the Target has Bytes.
2 - Do 5 times the Rounds as the Target has Bytes.
3 - Do 6 times the Rounds as the Target has Bytes.
4 - Do 8 times the Rounds as the Target has Bytes.

Example

```
'*************************************************************
*****
' XOR-Sample
'*************************************************************
*****
'
' We encrypt "file to file"
$$FIL=?path\Western.txt
$$OUT=?path\Crypted.txt
$$KEY=1-+rDsdf/(klsdaf
CFF.$$FIL|$$PLA
XOR.eo|$$PLA|$$KEY|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
XOR.df|$$OUT|$$KEY|$$PLC
' Save it to a file
CTF.$$NEW|$$PLC
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*************************************************************
*****
' XOR-Sample 2
'*************************************************************
*****
$$FIL=?path\Western.txt
$$OUT=?path\Crypted.txt
$$KEY=1-+rDsdf/(klsdaf
' We encrypt "file to file"
XOR.efo|$$FIL|$$KEY|$$OUT

' We decrypt it like this
$$NEW=?path\Receiver.txt
XOR.dfo|$$OUT|$$KEY|$$NEW
ENR.


'*************************************************************
*****
' XOR-Sample 3
' Single Character XOR
'*************************************************************
*****
' A - 65 - 01000001
' 1 - 49 - 00110001
' XOR================
' p   112- 01110000
' 1 - 49 - 00110001
' XOR================
' A - 65 - 01000001
'
VAS.$$KLA=A
VAS.$$KEY=1
PRT.Klartext=$$KLA
XOR.e$$CNT|$$KLA|$$KEY
PRT.Verschlüsselt: $$KLA
XOR.d$$CNT|$$KLA|$$KEY
PRT.Klartext=$$KLA
MBX.!
ENR.
```

Remarks

`XOR.` is provided to enable you, to apply mathematical XOR on any amount of Data.
To generate an encrypted "One Time Pad", we recommend that you rather use the `GEC.`-Command.

To use the encrypted characters in Chat-Programs or e-mail, do not forget to use the STR.TEXT TO MIME 3215 before,
Of course the encoding must be reversed on the other side before decryption.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

1. File-Size
The actual Implementation tries to load the complete file into Memory before applying the Encryption. Therefore Files with a size that exceeds the Systems RAM Capabilities can not be encrypted or decrypted. In case there are File-Errors, an empty string is returned.

See also:

- HIT. - Hide Text 835
- STR.CHARS A 3155
- STR.CHARS B 3159
- STR.CHARS C 3163
- STR.CHARS D 3167
- STR.GENERATE - Mode 3188
- STR.FROM HEXASCI 3185
- STR.TO HEXASCI 3222
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- FIL.Split File 1001
- FIL.Join File 994
- FIL.Join Delete 990

© 2023 Theo Gottwald * https://www.smart-package.com

3.51 Environment Variables and Global Signals

3.51.1 EVE. - Environment Variable Expand

EVE. - Environment Variable Expand Previous 843 Top 41 Next 854

MiniRobotLanguage (MRL)

EVE. Command

Environment Variable Expand

Intention

Expands Environment Variables. It will expand one or more environment-variables in the given P1 parameter. The result will be on TOS (Top of Stack) or in P2 (Parmeter 2) if given.

Usage is as simple as you can see in the picture below.

While in normal use, you should always use variables which would look like this:

```
: $$WIN=%windir%
EVE.$$WIN
MBX.The result is: $tos$
ENR.
```

With exactly the same result. You can give a second parameter, in that case the result will not be on TOS (Top of Stack(, but in that variable.

```
: $$WIN=%windir%\system\
EVE.$$WIN|$$RES
MBX.The result is: $$RES
ENR.
```

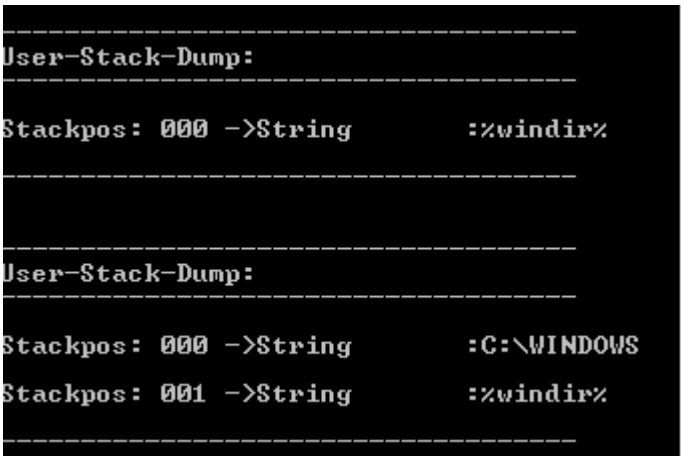
EVE. allows you to work "from stack to stack 369". If you do this:

```
VAR.$$WER=%windir%
PUV.$$WER
STS.DUMP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EVE.
STS.DUMP
```

This is how the stack will look like:



If you just want to get the value of a Environment variable, you should use `EVG.`
Use `EVS.` to set/change the value of environment variables.

Syntax

EVE.P1[|P2]

Parameter Explanation

P1 - (in) Environment variable
P1 - (optional) (out) Variable if omitted, TOS is used

Example

```
'*****************************************
' EVE. - Sample
'*****************************************
'DBM.2
VAR.$$WER=%windir%
PRT.Unexpanded: $$WER
EVE.$$WER
PRT.from stack ->$$000-
PRT.=======================================
VAR.$$WER=%windir%
PRT.Unexpanded: $$WER
EVE.$$WER|$$RES
PRT.from variable->$$RES-
PRT.=======================================
DMP.4
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:over
MBX.Halt to read output
'*********************************************
```

Remarks

-

Limitations:

-

See also:

- EVG. - Environment Variable Get 854
- EVS. - Environment-Variable-Set 857

© 2023 Theo Gottwald * https://www.smart-package.com

3.51.2 EVG. - Environment Variable Get

MiniRobotLanguage (MRL)

EVG. Command

Environment Variable Get

Intention

Reads Environment Variables and return their values. Depending on parameters these can be from the local or from the global Environment variables table.

If you specify a numbered parameter P1, then you can get the Environment variable with that number. Example:

```
EVG.#3
STS.DUMP
```

the result on theTOS (Top of stack) 369 will look like this:

```
------------------------------------
User-Stack-Dump:
------------------------------------
Stackpos: 000 ->String       :CLASSPATH=.;C:\Programme\Java\jre6\lib\ext\QTJava.zip
------------------------------------
```

of course it can look different on your computer, as you may have other environment variables.

The global Environment table is been read from (32-bit) registry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager\Environment
```

Syntax

EVG.[P1][|P2][|P3]

Parameter Explanation

P1 - (in) is taken as name for the environment-variable to get. Result is placed on TOS (top on stack). You can specify a "#(num)Parameter to get the Entry with that number. "#(num)" will only work with the local table. Trying "#(num)" with the global table will result in an error.

 *© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - `(out) Result is placed in variable, if given.`

P3 - `Options`
```
  e - (expand) will expand environment variable.
  g - (gobal) reads String from global Environment-table
      instead from local Application Table.
  n - (noexpand) will not expand the result.
  t - (table)  reads String from local Application Table
      instead of from global Environment-table
```

* Expand means %path(something)% is replaced with the real physical path.

In case there is no Parameter, TOS is taken as Name for the Environment-Variable to get.
Result is placed on TOS.

Example

```
'*******************************************
' EVG./EVS. - Sample
'*******************************************
'DBM.2
' Get %SystemRoot%
EVG.SYSTEMROOT|$$VAR
PRT.Systemroot=>$$VAR
PRT.=========================================
' EVS. without opt writes a global Environment variable
EVS.MyVar|MyValue
EVG.MyVar|$$RES|g
PRT.$$RES
PRT.=========================================
' EVS. with opt t writes a local Environment variable
EVS.MyVar|MyValue|t
EVG.MyVar|$$RES
PRT.$$RES
PRT.=========================================
DMP.4


:over
MBX.Halt to read output
'*******************************************
```

Remarks

-

Limitations:

The option to get Environment entries by their number `"#(num)"` will only work with the local table. Trying "#(num)" with the global table will result in an error.

© 2023 Theo Gottwald * https://www.smart-package.com

The global environment table is read from 32-bit registry.

See also:

- **EVE. - Environment Variable Expand** 851
- EVS. - Environment-Variable-Set 857

© 2023 Theo Gottwald * https://www.smart-package.com

3.51.3 EVS. - Environment-Variable-Set

MiniRobotLanguage (MRL)

EVS. Command

Environment Variable Set

Intention

Sets or clears an Environment Variable entry. Depending on parameters these can be from the local or from the global Environment variables table.

The global Environment table is been read directly out of 32-bit registry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager\Environment
```

Syntax

EVS.P1|P2[|P3]

Parameter Explanation

P1 **-** (in) is taken as name for the environment-variable to use.
P2 **-** (in) this is the value that is stored in P1

P3 **-** Options
  g - (gobal) write String from global Environment-table
instead
  to local Application Table.
  t - (table) writes String to local Application Table
instead
  of to global Environment-table

Example

```
'*******************************************
' EVG./EVS. - Sample
'*******************************************
'DBM.2
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Get %SystemRoot%
EVG.SYSTEMROOT|§§VAR
PRT.Systemroot=>§§VAR
PRT.========================================
' EVS. without opt writes a global Environment variable
EVS.MyVar|MyValue
EVG.MyVar|§§RES|g
PRT.§§RES
PRT.========================================
' EVS. with opt t writes a local Environment variable
EVS.MyVar|MyValue|t
EVG.MyVar|§§RES
PRT.§§RES
PRT.========================================
DMP.4



:over
MBX.Halt to read output
'*****************************************
```

Remarks

-

Limitations:

The global environment table is read from 32-bit registry.

See also:

- EVE. - Environment Variable Expand 851
- EVG. - Environment Variable Get 854

© 2023 Theo Gottwald * https://www.smart-package.com

3.51.4 IGS. / NGS. - If-Global-Signal (Mutex)

IGS. / NGS. - If-Global-Signal (Mutex,Atom) ***Previous*** 857 ***Top*** 41 ***Next*** 861

MiniRobotLanguage (MRL)

IGS. Conditional Statement

If Global Signal

Intention

Conditional Statement.

Test if a specified Global Signal (Atom) or Global Alive-Signal (Mutex) exists.
Most often they will be set from another Robot.
Such a Signal has a name, and it is is set or is not set.
Such a Signal can be set or tested from any windows application to signal something to other programs. Normally the `SGS.` - command is used to set or delete these signals.

A Mutex can be set from another instance of this robot using SGS - Set Global Signal (Mutex) 861 or any other application, using the windows-API CreateMutex.
While a "Alive Signal" (=Mutex) can only be deleted by the application that created the Signal, the second Sort of Signals, the so called "Atoms" can be deleted by anybody who knows their name.

The negative Form of this statement is `NGS.`.

* *Mutex:* systemwide, global Signal.

Syntax

```
IGS.P1[|A] … ELS. … EIF.
```

Parameter Explanation

P1 - `Variable = Variable (must be a Variable or Stackvar, Len=5)`
`do NOT use Search Patterns with this command.`

A - `If P2 evaluates to "A" then IGS. does check for so called "Atoms". This is the second sort of global Signals that can be set using SGS. Command.`

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'***********************************
' IGS.-Test
' Let this script run on instance 1
'***********************************
SGS.Peter
DOL.1
IGS.Theo
 PRT. -Theo-
 'SGS.Theo|R
ELS.
 PRT. . . .
EIF.
PAU.0.1
OOP.
END.


'***********************************
' IGS.-Test
' Let this script run on instance 2
'***********************************
SGS.Theo
PAU.5
SGS.Theo|R
PAU.5
SGS.Theo

PAU.3

END.
```

Remarks

-

Limitations:

-

See also:

- SGS - Set Global Signal (Mutex) 861

© 2023 Theo Gottwald * https://www.smart-package.com

3.51.5 SGS. - Set Global Signal (Mutex)

SGS - Set Global Signal (Mutex,Atom) Previous 859 Top 41 Next 867

MiniRobotLanguage (MRL)

SGS. Command

Set Global Alive-Signal (Mutex)

Intention

The `SGS.`-Command can be used to **set/remove or wait for a PC-Global Signal**, which can be checked for its existence in another instance of this robot using `IGS.` or `SGS.`

You can imagine these Signals like a Table with names in it.
Any program in the Computer can
- add a name to that table
- ask "Is the name XXX already in the table?"
- delete self created names (in case of Mutex)
- or delete any names (in case of Atoms)

There are two types of Global Signals that are currently available: one is called "Mutex" and the second is called "Atom". Both are different in some points.

1. Mutex disappear when the program closes that has set these mutex.
2. Atom do not necessarily disappear when a program closes
3. Mutex can only be deleted by the program which created them
4. Atoms can be deleted by anybody who knows the name of the "Atom"

A Mutex can be set from another instance of this robot using SGS - Set Global Signal (Mutex) 861 or any other application, using the windows-API CreateMutex.
While a "Alive Signal" (=Mutex) can only be deleted by the application that created the Signal, the second Sort of Signals, the so called "Atoms" can be deleted by anybody who knows their name.

You can set a signal, then this signal will stay until the Robot is closed, or until the

```
SGS.[Signal]|R
```

Command is executed. If you prefer to use an Atom rather then a Mutex, just add an "A".

```
SGS.[Signal]|RA
```

Please note that in case of Mutex, only the Process that created the Signal can remove the Signal.

`SGS.` can be used to **synchronize systemwide resources.**

© 2023 Theo Gottwald * https://www.smart-package.com

Therefore it can wait for a Signal to be removed, and then it can automatically set the Signal for itself. This way resources can be shared over many instances or programs.

You could even run several installations at the same time, while leaving the mouse to the instance which actually owns the Signal.

The Signal is an "Alive-Signal", it will only be there as long as the robot that has set the signal is "alive". Therefore the Signal **can also be used to trap, if a robot has been killed** or has ended its task. Because then the signal is automatically destroyed.

Technically the robot should be able to receive Variable-Content (`PRV.`,`WFV.`) while waiting for a signal with `SGS.`.

"Global Alive Signals" (Mutex) can also be used to synchronize your script with self written applications*.

*use MSDN-API: OpenMutex()

Syntax

SGS.P1[|P2]

Parameter Explanation

P1 - Name of the Mutex or Atom that should be handled.
P2 - ***(optional)***one of the letters below. If omitted, the "M" (create Mutex) is the default.

```
For Mutex:
 M - Create Mutex (Raise Signal).
 R - Remove Signal - the Signal will be removed.
     Only Signals which have been raised by this instance
     can be removed from this instance.
 W - Wait (unlimited time!) for Signal to be removed by
     another instance or program.
 C - Wait-and-set signal. Wait till the Signal is removed,
     then set the signal himself.
 F - Wait for Signal to be set.

For Atoms:
 MA - Create Mutex (Raise Atom).
 RA - Remove Signal - the Signal will be removed.
 WA - Wait (unlimited time!) for Signal to be removed by
     another instance or program.
 CA - Wait-and-set signal. Wait till the Signal is removed,
      then set the signal himself.
 FA - Wait for Signal to be set.
 FA - Wait for Signal to be set. When Signal was set,
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
      delete Signal and go on ...
```

Example

```
'************************************************
' SGS. - Exampel 1 using Mutex
'************************************************

$$MYS=Mya_
PRR.0
  DOL.1
    IGS.$$MYSSignal
      PRT.Got Signal !!
      SGS.$$MYSSignal_Rec
      PAU.0.1
      SGS.$$MYSSignal_Rec|R
    EIF.
    PRT.Looping
    PAU.0.2
  OOP.
  END.
PRE.

$$MYS=Myb_
PRR.0
  DOL.1
    IGS.$$MYSSignal
      PRT.Got Signal !!
      SGS.$$MYSSignal_Rec
      PAU.0.1
      SGS.$$MYSSignal_Rec|R
    EIF.
    PRT.Looping
    PAU.0.1
  OOP.
  END.
PRE.


FOR.$$CNT|1|9
  PAU.1
  $$MYS=Mya_
  GSB.do_send

  $$MYS=Myb_
  GSB.do_send
NEX.

MBX.Waiting
ENR.

:do_send
' Sende globales Signal
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SGS.$$MYSSignal
' Warte auf Received Signal
SGS.$$MYSSignal_Rec|F
' Lösche Signal
SGS.$$MYSSignal|R
RET.
ENR.

'************************************************
' SGS. - Exampel 1 using Atoms
'************************************************

$$MYS=Mya_
PRR.0
  DOL.1
    IGS.$$MYSSignal|A
      PRT.Got Signal !!
      SGS.$$MYSSignal|RA
    EIF.
    PRT.Looping
    PAU.0.2
  OOP.
  END.
PRE.

$$MYS=Myb_
PRR.0
  DOL.1
    IGS.$$MYSSignal|A
      PRT.Got Signal !!
      SGS.$$MYSSignal|RA
    EIF.
    PRT.Looping
    PAU.0.1
  OOP.
  END.
PRE.


FOR.$$CNT|1|9
  PAU.1
  $$MYS=Mya_
  GSB.do_send

  $$MYS=Myb_
  GSB.do_send

NEX.

MBX.Waiting
ENR.

:do_send
' Sende globales Signal
SGS.$$MYSSignal|MA
RET.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'************************************************
' SGS. - Exampel 2 using Atoms
'************************************************

' We start 2 Parallel Robots
$$MYS=Mya_
$$MYT=0.2
GSB.Paro

$$MYS=Myb_
$$MYT=0.1
GSB.Paro


' This is the Main Robot
FOR.$$CNT|1|9
  PAU.1
  $$MYS=Mya_
  GSB.do_send

  $$MYS=Myb_
  GSB.do_send

NEX.

MBX.Waiting
ENR.

:do_send
' Sende globales Signal
SGS.$$MYSSignal|MA
RET.
ENR.

:Paro
PRR.0
  DOL.1
    SGS.$$MYSSignal|KA
    PRT.Got Signal !!
    PAU.$$MYT
  OOP.
  END.
PRE.
RET.
```

Remarks

-

Limitations:

Currently there is no way to change the waiting time to anything else then unlimited if using `SGS.`

© 2023 Theo Gottwald * https://www.smart-package.com

To use a timeout, use `IGS.` and `#dsince#` (See Systemvariables 334).

See also:

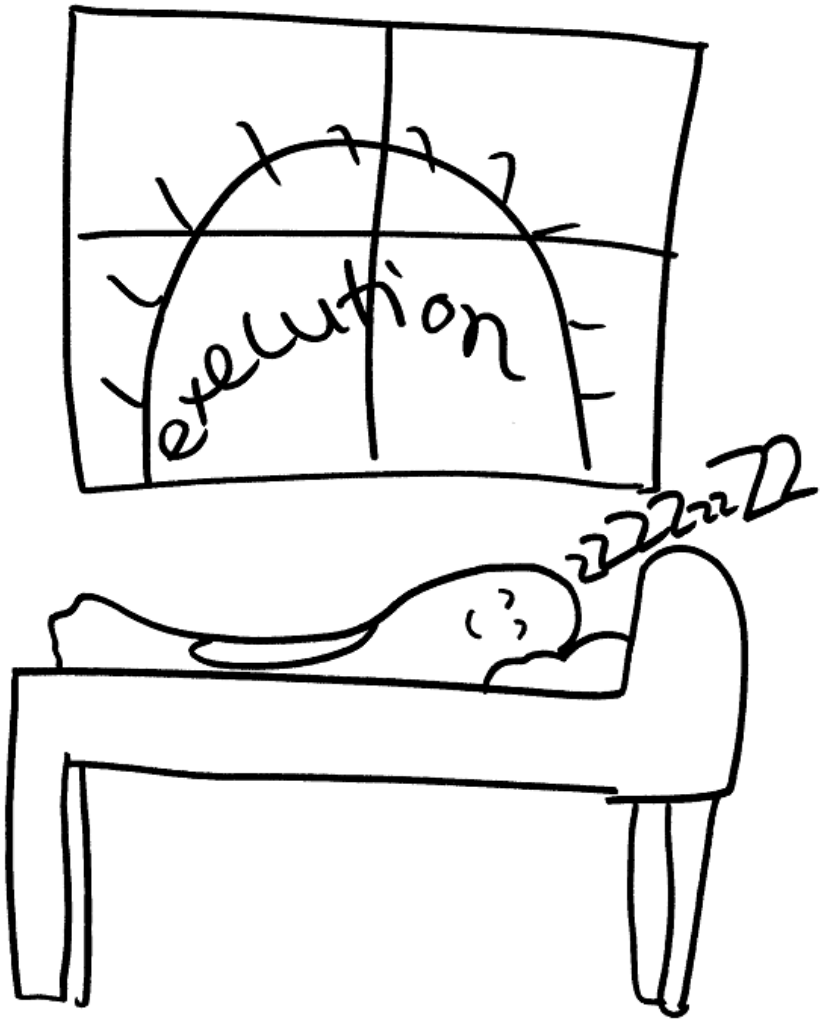
- IGS. / NGS. - If-Global-Signal (Mutex) 859
- 1.6.1. Program Flow Control 121
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- PRV. - Parallel-Robot send Variable 2651
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861
- WFM. - Wait-For-Message 2140
- TWM. - Text-Window-Message 2136

© 2023 Theo Gottwald * https://www.smart-package.com

3.52 Execute-Commands

3.52.1 ! Smarty's Execution Commands

Smarty's Execution Commands



In many cases, you can just drag and drop an executable into the Editor to get the right execution command inserted.

Execution commands in general

These commands will "start" another programm, a Uninstallation or a document.
They will of course also start an compiled SPR-Script or nearly an other file that can be started.

Sometimes you will need to try which of the commands works fine in your case.

*© 2023 Theo Gottwald * https://www.smart-package.com*

They use different technologies internally to give you a broad range of options.

`EXE.` is maybe the most often used execution command.
In case that does not work, try `EXO.`, or `EXS.`

Now if all of these do not do what you want, take a closer look.

Is it an MSI.-file?
Use `EXM.`

Is it an registered document file?
Use `EXR.`

Do you want to uninstall something?
Then `EXU.` is possibly what you need.

Do you want to run a batch file with DOS-Commands?
Use `EXC.` for that.

Do you want to start a lot of Applications as fast as possible?
Use `EXF.` then.

Generally there are two ways of starting an executable:
- blocking -> script will wait until the executable has finished)
- non blocking -> script will start the executable and continue running, without waiting for the executable to end.

This is the first choice you have to make, when choosing the right execution command.

Execution on the "Hidden System Desktop"

Hint: This Feature requires Windows Vista or higher.

Imagine, you can execute a Installation or Automation on a hidden Desktop that the user can not see.
To be able to do this, the SPR must have full Admin-Rights.

Then you can simply use the Command:

```
EXV.?path\myfile.exe|c
ENR.
```

To start your executable with System-Rights on the Hidden Desktop.
The "`|c`" means "Cleanup" and should be used after any call to start something on the Hidden System-Desktop.
In case you have started your executable like this:

```
EXV.?path\myfile.exe
PAU.10
' then you can later do the cleanup like this
EXV.|c
ENR.
```

How can I see the the "Hidden System Desktop"?

© 2023 Theo Gottwald * https://www.smart-package.com

There are some steps necessary to be able to see the Hidden Desktop.

1. Preparation:

We have implemented all necessary steps into the `EXV.` - Command to make it easy for you.
Just type:

```
EXV.|pc
ENR.
```

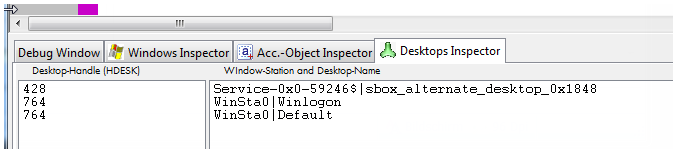
And execute this script. This will make all preparation to enable you to see the hidden Desktop.

2. Open a Window on the Hidden Desktop

This can most easily be done like this:

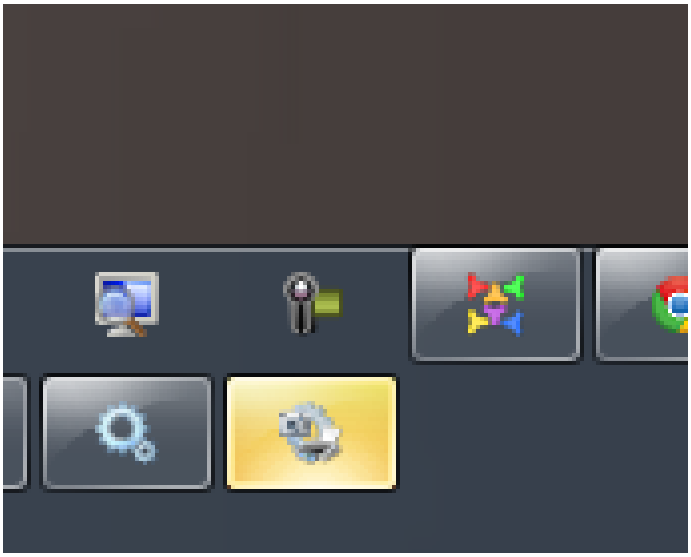
```
EXV.-|c
ENR.
```

The SPR-Editor has a "Desktops-Analyzer" that will show the "Hidden Desktop" once it has been opened.



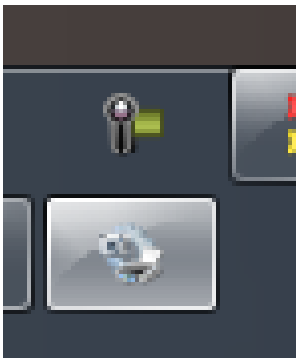
3. Now you should see something like this in your Windows-Taskbar

Click on it.



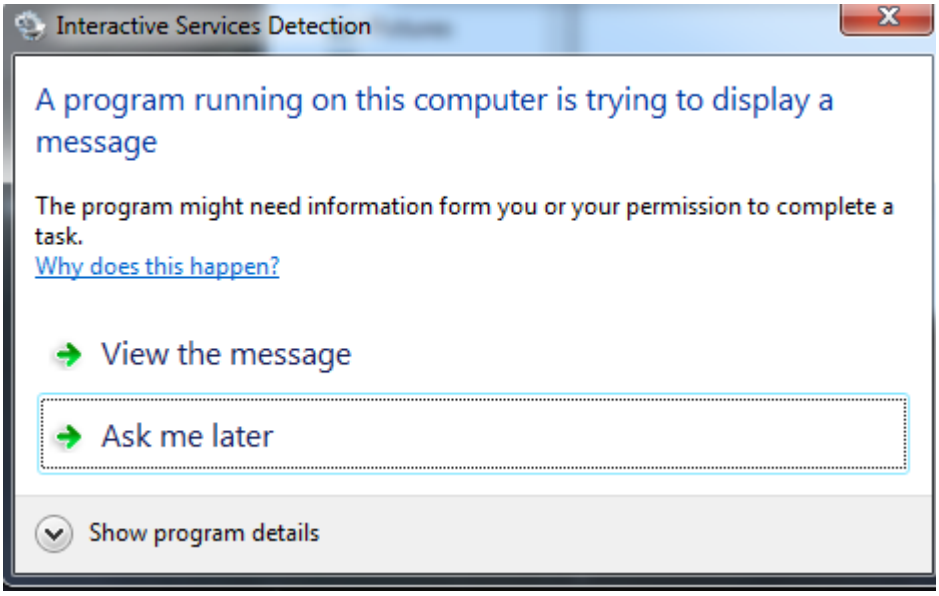
Once you have touched it with the Mouse, it will stop blinking and look like this:

© 2023 Theo Gottwald * https://www.smart-package.com



4. Now click the upper choice.

Anyway, just click on it with the Mouse. Next you will see this window:

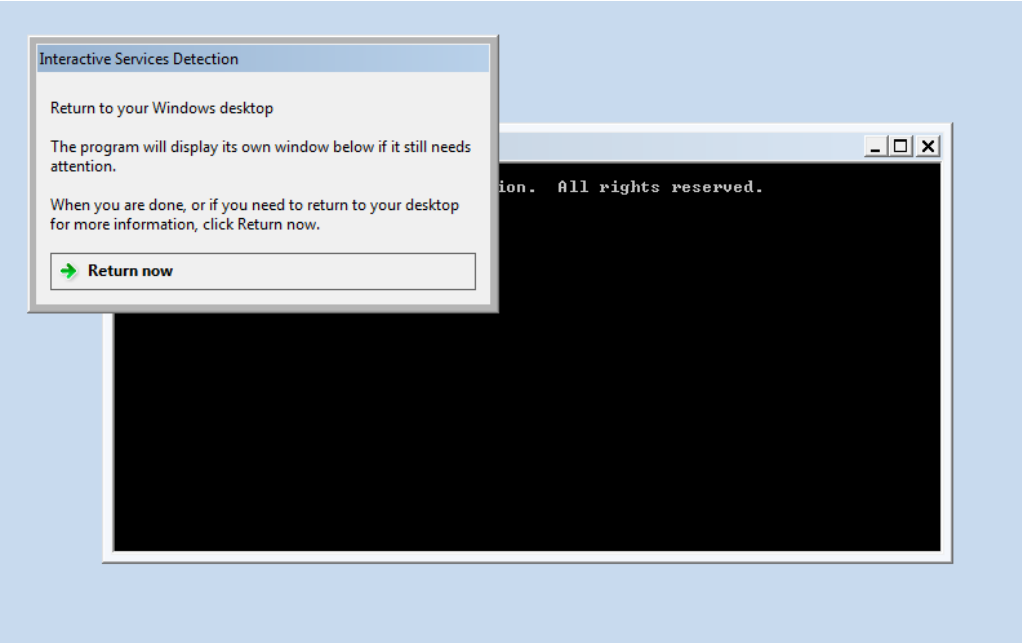


In this case you would click on:

"View the message"

5. Voila ... the Hidden Desktop will show up to you!

© 2023 Theo Gottwald * https://www.smart-package.com



It will look somehow like this:

Note: On the "Hidden Desktop" you have complete System-Rights!
Type "`WhoAmI`" in to the Console window and hit enter to see your current account.
To return to your normal Desktop, close the console window and click the Button in the grey window.
If you leave the console window open, you will always be able to easily return to the "Hidden Desktop".

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.52.2 EXA. - Execute Asynchron

MiniRobotLanguage (MRL)

EXA. Command

Execute Asyncron

Intention

`EXA.` can be used to start executable files, non-blocking*. The command will not wait until the process is started. Therefore `EXA.` is very fast.
Usage is as simple as:

```
EXA.?exepath\Sample Apps\TestApp_01.exe
```

If you use `EXA.`, internally a background process will be started to do the job. This **background process** will:

1. Set the given variable - if any - to zero
2. start the executable
3. wait for it to run
4. assign the PID 137 to the given variable if there is any. This can happen at any time later.

Therefore the PID may not be available immediately after the command. As this depends on the target process, there is no warranty when/if the PID will be available.

As the given PID-variable (**P2**) will be set to zero at the start of the `EXA.`-command, you can wait for the PID to become available in case you need that.

`EXA.` itself can not be blocked and that is its strong side. Internally its as fast as `EXF.`
It may deliver the PID if you give a variable.

Special Mode: Stack-Variable as P2
If you give a stack variable (`$$000` for example) as **P2** then the script assumes that a real variable name is been placed on the stack before and will try to resolve the stack for that variable. If that is successful, this real variable will get the PID when its available. If that fails because there is no variable-name on the stack, the result is been placed in the given stack variable.

*The term "non-blocking" means that the command will never block the script from further execution. The script will continue with execution no matter what sort of executable has been started.

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

EXA.P1[|P2]

Parameter Explanation

P1 - Document-Filename (can be a hyperlink)

P2 - (optional) Variable that will be filled with PID of the started process. Important: This happens at ANY TIME later.
If omitted, PID is been dropped (NOT placed on TOS, NOT put into #pid#).

P3 - (optional) Window Size Parameter
You can use the following window-size parameters for P2

0 - HIDE - Hides the window and activates another window.
1 - SHOW NORMAL Activates and displays a window.
2 - SHOW MINIMIZED - Activates the window and displays it as a minimized window.
3 - MAXIMIZE - Maximizes the specified window.
4 - SHOW-NO-ACTIVATE - Displays a window in its most recent Size and position. The active window remains active.
5 - SHOW - Activates the window and displays it in its current size and position.
6 - MINIMIZE - Minimizes the specified window and activates the next top-level window in the Z order.
7 - SHOW-MIN-NO-ACTIVE - Displays the window as a minimized window. The active window remains active.
8 - SHOW-NO-ACTIVE - Displays the window in its current state. The active window remains active.
9 - RESTORE - Activates and displays the window.
10 - SHOW-DEFAULT - Sets the SHOW STATE to default.

Example

```
'***********************************************
' Example 1 - without Windows-Parameter
'***********************************************
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
FOR.$$CNT|1|5
  EXA.$$LNK|$$PID
  DBP.$$PID
NEX.
PAU.10
KPN.TestApp|a
MBX.Ready!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'***********************************************
' Example 2 - using Windows-Parameter
'***********************************************
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
FOR.$$CNT|1|5
' Start them minimized
  EXA.$$LNK|$$PID|6
  DBP.$$PID
NEX.
PAU.10
KPN.TestApp|a
MBX.Ready!
ENR.
'***********************************************

'***********************************************
' Example 3 - Wait for PID to appear
'***********************************************
KPN.TestApp|a
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
FOR.$$CNT|1|15
  EXA.$$LNK|$$PID|1
  :Loop
  PAU..1
  JIZ.$$PID|Loop
  PRT.$$PID
NEX.
PAU.20
KPN.TestApp|a
MBX.Ready!
ENR.
```

Remarks

Technical remark: `EXA.` uses multithreaded asynchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

Limitations:

-

See also:

- 1.7. Application and PID 137
- 3.3 Specialfolders 345
- EXX. - Execute-Extended 916
- EXE. - Execute Programm 883
- EXH. - Execute-Handle 890
- EXB. - Execute-Blocking 876

© 2023 Theo Gottwald * https://www.smart-package.com

- EXO. - Execute Open Document 895
- **IEF. / NEF. - If - exist - File** 1050
- **IEP. / NEP. - If-Exists-Process** 2900

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.52.3 EXB. - Execute-Blocking

MiniRobotLanguage (MRL)

EXB. Command

Execute Blocking. Halt script execution until the started program has finished.

Intention

Run an executable program synchronously and deliver the exit-code. The current script is suspended until the called program ends.

The **Exitcode** (Error-level, Returncode) of the started application/command-line will be **placed on the stack** (TOS 369) .

Usage of `EXB.` is as simple as it can be:

```
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXB.$$LNK
```

You can add a window-size parameter **P2** for example if you want the application to start maximized or minimized.

```
' Start the application maximized
EXB.$$LNK|3
```

As an alternative to `EXB.` you can use EXE 883. and use the PID 137 together with WPT 2913.

```
' Wait until the started application ends
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXE.$$LNK
WPT.#pid#
ENR.
```

You can work with the started application from a parallel running robot.
See example below.

Syntax

EXB.P1[|P2]

Parameter Explanation

P1 - `Document-Filename (can be a hyperlink)`
P2 - `Window Size Parameter`
`You can use the following window-size parameters`

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
    0 - HIDE    -   Hides the window and activates another
    window.
    1 - SHOW NORMAL   Activates and displays a window.
    2 - SHOW MINIMIZED - Activates the window and displays it
    as a
        minimized window.
    3 - MAXIMIZE -   Maximizes the specified window.
    4 - SHOW-NO-ACTIVATE - Displays a window in its
        most recent Size and position.
        The active window remains active.
    5 - SHOW - Activates the window and displays it
        in its current size and position.
    6 - MINIMIZE - Minimizes the specified window and
        activates the next top-level window in the Z order.
    7 - SHOW-MIN-NO-ACTIVE - Displays the window as
        a minimized window. The active window remains active.
    8  - SHOW-NO-ACTIVE - Displays the window in its current
    state.
          The active window remains active.
    9 - RESTORE - Activates and displays the window.
    10 - SHOW-DEFAULT - Sets the SHOW STATE to default.
```

The Exitcode (Errorlevel, Returncode) of the started Application/comandline will be placed on the stack (TOS 369).

Example

```
'***********************************************
' Example 1 - without Windows-Parameter
'***********************************************
' First we start a Parallelrobot that will
' end the started Application after 5 seconds.
PRR.
  PAU.5
  KPN.TestApp_01.exe|a
  ENR.
PRE.
' EXB. - Beispiel
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
' Here we will wait until the Application ends
EXB.$$LNK
DBP.Exit-Code: $tos$
ENR.'***********************************************
```

Remarks

Technical remark: `EXB.` uses synchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

© 2023 Theo Gottwald * https://www.smart-package.com

EXB. will suspend the script process until the started process has ended. The script process is not waiting, its suspended. The suspended script process does not need any CPU resources, but also does not react on any message or whatever.

Limitations:

-

See also:

- 1.7. Application and PID 137
- 3.3 Specialfolders 345
- EXX. - Execute-Extended 916
- EXE. - Execute Programm 883
- EXH. - Execute-Handle 890
- EXA. - Execute Asynchron 872
- EXO. - Execute Open Document 895
- **IEF. / NEF. - If - exist - File** 1050
- **IEP. / NEP. - If-Exists-Process** 2900

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.52.4 EXC. - Execute Batch Command

MiniRobotLanguage (MRL)

EXC. Command

Execute Batch Command

Intention

Starts a Batch-File. `EXC.` will try to deliver #pid# 137 and `#tid#`. `EXC.` is generally a non-blocking command. This means that normally script execution goes on, while the started application is still running.

`EXC.` will only Batch-files. Usage is as simple as this:

```
EXC.?desktop\test.bat
```

You can omit the ".bat" and write also:

```
EXC.?desktop\test
```

Which will do just the same. Also you can specify a second parameter, as the current directory for the batch process:

```
EXC.?desktop\test.bat|?desktop\
```

While `EXC.` is generally a "Non blocking" command, you can make it wait until the batch file is processed, like this:

```
EXC.?desktop\test.bat|?desktop\
WPT.#pid#
```

If `EXC.` can not start the file, the timeout-flag #tio# 334 will be set to 1, else it will be reset to 0.

Syntax

EXC.P1[|P2]

Parameter Explanation

P1 - batch file path and name

P2 - (optional) current directory for the batch process

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'Example-Batch File: "Test.bat"
@ECHO off
ECHO Hello World!
PAUSE

'***********************************************
' Example 1
'***********************************************
EXC.?desktop\test.bat
'***********************************************


'***********************************************
' Example 2 - using Windows-Parameter
'***********************************************'
'
EXC.?desktop\test.bat|?desktop\
'***********************************************
```

Remarks

-

Limitations:

There may be cases where `EXC.` seems not to deliver the proper PID[137]/ TID. This depends on the target application, and may be influenced by the fact, if the target application is already running or not. In a closer look, the PID and TID from `EXC.` was accurate, but the target application gave control to another (already running or newly started) instance of herself.

See also:

- GEP. - Get-Executable-Path[2582]
- **EXS. - Execute Shell**[905]
- EXF. - Execute Fast[885]
- EXH. - Execute-Handle[890]
- EXO. - Execute Open Document[895]
- **WPT. - Wait-Process-Termination**[2913]

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.5 EXD. - Execute Direct

MiniRobotLanguage (MRL)

EXD. Command

Execute Direct Command

Intention

Starts a File/executable. `EXD.` will try to deliver #pid# 137 and `#tid#`. `EXD.` is generally a non-blocking command. This means that normally script execution goes on, while the started application is still running.

But there is a chance, that `EXD.` can possibly be blocked by some applications. Use another command in that case, for example EXA. - Execute Asynchron 872.

`EXD.` will only start executable files. Usage is as simple as this:

```
EXD.?pfad\TestApp.exe
```

You can specify a command-line parameter for the started application.

```
EXD.?pfad\TestApp.exe|your command-line
```

If `EXD.` can not start the file, the timeout-flag #tio# 334 will be set to 1, else it will be reset to 0.

Syntax

EXD.P1[|P2]

Parameter Explanation

P1 - executable with complete path

P2 - (optional) `command-line Parameter for the started application`

Example

```
'*********************************************
' Example 1
'*********************************************
' Start the batch file
EXD.?ws\cmd.exe|/c "?desktop\Test.bat"
' Wait until the batch file has ended
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
WPT.#pid#
'*******************************************
```

Remarks

Technical remark: EXD. uses Create-Process--Technology and is compatible with other commands that use this technology.
EXD. is non-blocking, but can be blocked by a process that it has started. If this turns to be a problem, choose EXA. or EXF. instead. For starting Batch-Files, prefer EXC. - Execute Command 879.

Limitations:

There may be cases where EXC. seems not to deliver the proper PID 137/ TID. This depends on the target application, and may be influenced by the fact, if the target application is already running or not. In a closer look, the PID and TID from EXC. was accurate, but the target application gave control to another (already running or newly started) instance of herself.

See also:

- **1.7. Application and PID** 137
- GEP. - Get-Executable-Path 2582
- **EXC. - Execute Command** 879
- **EXS. - Execute Shell** 905
- EXF. - Execute Fast 885
- EXH. - Execute-Handle 890
- EXO. - Execute Open Document 895

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.6 EXE. - Execute Programm

MiniRobotLanguage (MRL)

EXE. Command

Execute Programm

Intention

Will execute the given program or command-line and return the `PID` of the started program in `#pid#`. The system variable `#tid#` is set to zero.

`EXE.` is optimized for command-lines that need parameters. Place the parameters in P1.

If you want to wait until the started process ends, use WPT. - Wait-Process-Termination 2913. with the PID 137 that has been delivered by `EXE.`.

```
' Wait until the started application ends
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXE.$$LNK
WPT.#pid#
ENR.
```

Syntax

EXE.P1[|P2]

Parameter Explanation

```
P1 - Filename of program to start

P2 - (optional) command-line-parameter to command in P1
```

Example

```
'*********************************************
' Example 1
'*********************************************

EXE.?ws\mspaint.exe|yourpic.bmp
PRT.#pid#--#tid#
MBX.!
END.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'************************************************

'************************************************
' Example 2
'************************************************
' Wait until the started application ends
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXE.$$LNK|1
WPT.#pid#
ENR.
```

Remark

Technical remark: `EXE.` uses asynchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

If you want to have the Exit-Code of the started application, use EXB. - Execute-Blocking 876 instead.

Limitations:

-

See also:

- **3.3 Specialfolders** 345
- EXX. - Execute-Extended 916
- EXH. - Execute-Handle 890
- EXA. - Execute Asynchron 872
- EXB. - Execute-Blocking 876
- EXO. - Execute Open Document 895
- **IEF. / NEF. - If - exist - File** 1050
- **IEP. / NEP. - If-Exists-Process** 2900

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.7 EXF. - Execute File/Document

EXF - Execute File/Document Previous 883 Top 41 Next 890

MiniRobotLanguage (MRL)

EXF. Command

Execute File/Document

Intention

Starts a file or document like when you click on it with the mouse. `EXF.` will not only start executable files. It will also **start any registered file**, like a office-document or text-file. `EXF.` will also open folders.

`EXF.` will open hyperlinks and even download-links. This will both work as expected:

```
EXF.http://www.heise.de/
EXF.www.heise.de/
```

The document will be started somehow like if you have clicked on it with the mouse.

```
' Open a document
EXF.?desktop 345\MyActions.txt

' Open a system folder, like you click on a folder-icon
EXF.?ws\ 345
```

Disadvantage from that system is, that it does NOT deliver `#pid#` or `#tid#`.

`EXF.` is technically similar to `EXS.` (Execute Shell). It will however be more fast. In result, `EXF.` is not able to return an error code if there was a problem, because it just doesn't wait for that.

If you plan to start a lot of executable files very fast, or if the executable that you start blocks the script when you use `EXS.`, then `EXF.` is just what you need.

If you start a lot of executable files, `EXS.` is safe and slower, because it waits(/blocks the script-execution) until all internal system processes that start the file are running.

`EXF.` will just be very fast and start whatever you want as fast as your system can do.
`EXF.` unlike `EXS.` breaks through all API-limits (and can even crash a process if used careless). Try this example using `EXS.` or `EXF.` and see the difference.

```
KPN.iexplore.exe|a
FOR.$$CNT|1|20
  EXF.iexplore|www.fa2.de
NEX.
STS.DUMP
MBX.Click to Close all IE's
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
KPN.iexplore.exe|a
ENR.
```

however if you want to open many TAB's in your browser, `EXF.` is just faster: Try this:

```
' Open your daily browser
EXF.http://dict.leo.org
EXF.http://www.facebook.com
EXF.http://www.heise.de/
EXF.www.fa2.de
ENR.
```

Syntax

EXF.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - Executable or Document-Filename (can be a hyperlink)
P2 - ***(optional)*** Parameter for command-line
P3 - ***(optional)*** Path/Folder for command, if omitted current folder is used
P4 - ***(optional)*** Window Size Parameter, if omitted NORMAL is the default size

```
Parameter - (optional) Parameter/timeout
It will wait the given timeout that the window will be
realized.
When you add +100 to the windowsize, then you tell them NOT to
wait.
```

You can use the following window-size parameters

```
    0 - HIDE    -   Hides the window and activates another
    window.
    1 - SHOW NORMAL   Activates and displays a window.
    2 - SHOW MINIMIZED - Activates the window and displays it
    as a
        minimized window.
    3 - MAXIMIZE -   Maximizes the specified window.
    4 - SHOW-NO-ACTIVATE - Displays a window in its
        most recent Size and position.
        The active window remains active.
    5 - SHOW - Activates the window and displays it
        in its current size and position.
    6 - MINIMIZE - Minimizes the specified window and
        activates the next top-level window in the Z order.
    7 - SHOW-MIN-NO-ACTIVE - Displays the window as
        a minimized window. The active window remains active.
    8  - SHOW-NO-ACTIVE - Displays the window in its current
    state.
          The active window remains active.
    9 - RESTORE - Activates and displays the window.
    10 - SHOW-DEFAULT - Sets the SHOW STATE to default.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*********************************************
' Example 1 - without Windows-Parameter
'*********************************************

' Example1
DBM.2
EXS.?desktop\MR.txt|103
DMP.
END.

'*********************************************
' Example 2 - using Windows-Parameter
'*********************************************
' Example 2
DBM.2
' Variablen und Spezialordner können verwendet werden.
: §§FIL=?desktop\MyWord.doc
EXS.§§FIL|§§WIN
DMP.
END.
```

Remarks

Technical remark: `EXF.` uses multithreaded ShellExecute-Technology and is compatible with other commands that use this technology. `EXF.` can never be blocked by a process that it has started.

`EXF.` may work with some of these (depending on the OS):

```
EXF.system.ini
EXF.control
EXF.write
EXF.pbrush
EXF.winword
EXF.excel

' Start NetMeeting (XP)
EXF.conf
```

etc.

```
EXF. uses the default search path and will understand
following shortcuts to these executables in the window-folder.

'  Will open system control panel (XP, Win 7)
EXF.control

' Will open "Control Panel\All Control Panel
Items\Administrative Tools"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' XP, Win7
EXF.control admintools

' WIll open the  Clipboard Viewer
' XP only
EXF.clipbrd

' Systemconsole Prompt
' XP, Win7
EXF.CMD


' NotePad
' XP, Win7
EXF.notepad

' Windows Explorer
' XP, Win7
EXF.explorer

' XP, Win7
EXF.mspaint

' Open RegEdit
' XP, WIn7
EXF.regedit


' Windows Soundrecorder
' XP only
EXF.sndrec32

' Windows System Editor
' XP only
EXF.sysedit

You can use EXF. to open folders:

EXF.?ws\
EXF.?wi\
```

Limitations:

EXF. does deliver #pid# 137 or #tid#. It also does not deliver an error-code in case something did not work.
EXF. may not work well as the last command in your script, because the script may end before the background processes were processed. Also some programs may get an overrun from the speed of EXF.

See also:

- EXS. - Execute Shell 905
- EXH. - Execute-Handle 890

© 2023 Theo Gottwald * https://www.smart-package.com

- **EXR. - Executer-from-Registry** 902
- **GEP. - Get-Executable-Path** 2582
- EXO. - Execute Open Document 895

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.8 EXH. - Execute-Handle

MiniRobotLanguage (MRL)

EXH. Command

Execute Handle

Intention

Starts a program and returns directly the Window-Handle and the PID of the stared program.
The system Variable #pid# is set to the PID of the started process. #tid# is set to 0.
The Top-window handle from that process is been placed on the stack (TOS).

`EXH.` is using the Settings of OPT. 2406 for `WII` (Wait for Input Idle) and for WS (Windows-Size).

Syntax

EXH.P1

Parameter Explanation

P1 – ***complete Path and*** Executable (incl. Commandline)

This command will wait until the process is started and has a valid PID. It will then try to catch a window from that process within 30 seconds.

uses several general settings and sets the Systemvariable `#pid#`.

The general settings include:
- Sleep time in waiting-cycle

Example

```
'*********************************************
' Example EXH.
'*********************************************
EXH.?exepath\Sample Apps\TestApp_01.exe
VAR.§§HWN=$$000
MBX.PID=#pid# / Handle=§§HWN
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Technical remark: `EXH.` uses asynchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

The general default waiting time, that the program actually waits for a window to appear can not be changed at this time. This will be possible in a future version.

Limitations:

If the process you start, does not create a window by itself, this command will wait up to 30 seconds for such a window and set the timeout flag if no window appeared in this timne. Take for example the "Google Chrome" Browser. If you start it like this:

```
EXH.C:..\chrome.exe
```

and there are already running instances of it, the starting process is not the process that finally creates the window that you get to see. Therefore the command would end up waiting for a window that will never come.

Here is another example that will solve this issue using `CAW`.

```
CAW.RegEdit_RegEdit|Reg&AND:Edit
EXX.regedit
STS.DUMP
MBX.Ready
```

See also:

- 3.3 Specialfolders 345
- EXX. - Execute-Extended 916
- EXE. - Execute Programm 883
- OPT. - Optional Settings 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.9 EXM. - Execute MSI

MiniRobotLanguage (MRL)

EXM. Command

Execute MSI-Installer

Intention

`EXM.` is specially designed to easily start ".msi"-setup programs. Usage is as easy as possible. In most cases you can simply drag the MSI-File into the MRE-Editor and you will get a line like this:

```
EXM./i "?desktop\LibreOffice 3.4\libreoffice34.msi"
```

which is often all you need. If not, you can add MSI-Parameters 389.

`EXM.` will set `#pid#` and place the PID 137 of the started MSI-Setup on the Stack. Note that often the process you start will start another process. In that case the PID that you got from `EXM.` is not the PID of the process that generates a new window.

Important Hint:

`EXM.` will possibly **NOT work** on LAN Drives, like `\\drive\something`.

This compromise has been made to deliver #pid#.

If you need to start the Package on LAN, use `EXS.` instead.
Example:

```
EXS. ?ws\msiexec.exe|/i "$$PFA\tightvnc-2.7.10-setup-
64bit.msi"
```

Syntax

EXM.P1[|P2]

Parameter Explanation

```
P1 - Filename of MSI to start
     or in case of 2 Parameters the MSI-Call

P2 - Filename of MSI to start
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*********************************************
' Example 1
'*********************************************

EXM./i "?path\myfile.msi"
PRT.#pid#--$$000
MBX.!
END.

'*********************************************
' Example 2 (2 Parameter)
'*********************************************

EXM.?ws\msiexec.exe|/i "?path\myfile.msi"
PRT.#pid#--$$000
MBX.!
END.

'*********************************************
' Example 3 (1 Parameter wie EXE.)
'*********************************************

EXM.?ws\msiexec.exe /i "?path\myfile.msi"
PRT.#pid#--$$000
MBX.!
END.

'*********************************************
```

Remarks

Technical remark: `EXM.` uses asynchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

Limitations:

`EXM.` will possibly **NOT work** on LAN Drives, like

```
\\drive\something
```

This compromise has been made to deliver `#pid#.`
If you need to start the Package on LAN, use `EXS.` instead.
Example:

```
EXS. ?ws\msiexec.exe|/i "$$PFA\tightvnc-2.7.10-setup-
64bit.msi"
```


© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- MSI - Parameters 389
- EXS. - Execute Shell 905
- EXE. - Execute Programm 883
- EXO. - Execute Open Document 895
- **IAI. / NAI. - If-Application-Installed** 2601
- **GII. - Get-Installation-Information** 2584

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.52.10 EXO. - Execute Open Document

MiniRobotLanguage (MRL)

EXO. Command

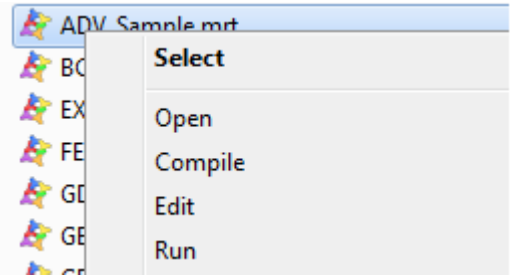
Execute Open Document

Intention

`EXO.` is an enhanced version of EXS. 905 It will simulate an "Explorer command".

`EXO.` will start a File or Document just like if you click on it with the right Mouse Button and choose something in the Explorer-Menu.

There are several options and Modes of operation. In the picture below you can see some examples. "**open**" is the default option, which is used if you omit parameter **P4**.



Here are some examples:

```
' Print the file
EXO.?desktop\readme.txt|||print

EXO.?desktop\readme.txt|||edit

' Open the "Find Dialog"
EXO.?desktop\|||find
```

EXO. is a versatile command that can start nearly anything you can imagine. Also it will provide you with `#pid#` 137 if possible. `#tid#` is not changed. Here is a sample script.

```
' Close all IE's
KPN.iexplore|a
' Open an IE
EXO.iexplore|"www.fa2.de"||open|(1+16+64)
MBX.PID of the new started process is: #pid#
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

EXO.P1[|P2][|P3][|P4][|P5]

Parameter Explanation

P1 **- executable** or Document-Filename or link, hyperlink

P2 **-** ***(optional)*** Parameter for executable that is been started
Example: EXO.iexplore|www.fa2.de

P3 **-** ***(optional)*** Folder for executable. This is for special purposes only. Current directory is set to this folder.

P4 **- (optional) Action parameter**
Some commonly used actions for P4 are:

edit
Launches an editor and opens the document for editing. If **P1** is not a valid executable/document file, the function may fail.

explore
Explores the folder specified by **P1**. May **need** the specification of **+32** for **P5 to work**.

find
Initiates a search starting from the specified directory. Opens the Find-dialog.

open
This is the default action, if none is specified. It opens the file specified by the **P1** parameter. The file can be an executable file, a document file, a folder or an hyperlink.

print
Prints the document file specified by P1. If P1 is not a valid document file, or can not be printed, the function may fail.

properties
Displays the file or folder's properties.

runas
There is also a new action called "**runas**". Under XP it will show the "RunAs"-Dialog. Under Vista and higher, it will launch the process as Administrator causing UAC elevation if required. Under the hood, it is the same as the explorer menu command "Run as administrator".

```
EXO.$$LNK|||runas
```


© 2023 Theo Gottwald * https://www.smart-package.com

***P5* - (optional) window-size and options parameter**

```
0 - HIDE    -   Hides the window and activates another
window.
1 - SHOW NORMAL   Activates and displays a window.
2 - SHOW MINIMIZED - Activates the window and displays it
as a
    minimized window.
3 - MAXIMIZE -   Maximizes the specified window.
4 - SHOW-NO-ACTIVATE - Displays a window in its
    most recent Size and position.
    The active window remains active.
5 - SHOW - Activates the window and displays it
    in its current size and position.
6 - MINIMIZE - Minimizes the specified window and
    activates the next top-level window in the Z order.
7 - SHOW-MIN-NO-ACTIVE - Displays the window as
    a minimized window. The active window remains active.
8  - SHOW-NO-ACTIVE - Displays the window in its current
state.
      The active window remains active.
9 - RESTORE - Activates and displays the window.
10 - SHOW-DEFAULT - Sets the SHOW STATE to default.
```

P5 - Special Flags
Add the following numbers to **P5** to get the desired effect:

+16 - enable internal asynchronous execution (faster!)

+32 - Display Error-Messages and User Interfaces
Example:

```
' Leads to an error-message
EXO.iexplore|www.google.de||explore|33
```

+64 - Deactivates the built in "Wait-for-input-Idle".
If this flag is set, the script will not wait until the started executable is able to receive input. It will immediately return. This default-active builtin feature has an unchangeable time-limit of 60 seconds. Example with deactivated WII.:

```
EXO.iexplore|www.google.de||open|(33+16+64)
MBX.PID of the new started process is: #pid#
```

+128 - special quotation processing. If you use command-line-scripts and they do not work as expected, you can try this Option.

Hint: For better readability you can specify a formula as **P5** like here:

```
EXO.excel|mysheet.xls||edit|(1+32+16+64)
MBX.PID of the new started process is: #pid#
```

Error-codes from EXO. on TOS 369.

© 2023 Theo Gottwald * https://www.smart-package.com

```
   0       - (no error) the command executed as expected
   1       - OS reports out out memory
   2       - The system cannot find the file specified.
   3       - The system cannot find the path specified.
   5       - Access to that file was denied.
   8       - Not enough memory is available to process this command.
   11      - The .EXE file is invalid (non-Win32 .EXE).
   15      - The system cannot find the drive specified.
   27      - The filename association is incomplete or invalid.
   28 - 30   DDE-Errors
   31      - This file-ending is not registered
   32      - There is a DLL is missing.
```

Example

```
'***********************************************
' Example 1 - start new IE with Link
'***********************************************
VAR.$$LNK=http://www.fa2.de
' Will run internet explorer
EXO.iexplore|$$LNK
PAU.1
' Will run the default browser
EXO.$$LNK
:over
MBX.Halt to read output
'***********************************************


'***********************************************
' Example 2 - without Windows-Parameter
'***********************************************'
DBM.2
' Variables und Spezialfolder can be used.
EXO.?desktop\MR.txt
DMP.
:over
MBX.Halt to read output
'***********************************************



'***********************************************
' Example 3 - using Windows-Parameter
'***********************************************
' EXO. - Beispiel
DBM.2
' Variables und Spezialfolder can be used.
: §§FIL=http://www.fa2.de
EXO.§§FIL|§§WIN
DMP.
:over
MBX.Halt to read output
'***********************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:over
MBX.Halt to read output
'*******************************************
```

Remarks

Technical remark: EXO. uses extended ShellExecute-Technology and is compatible with other commands that use this technology.

EXO. can be blocked by a process that it has started. In this case specify the +64 and the +16 (= +80) parameter for P5. If that does not work, you can use EXF. in many cases.

Note that the returned PID[137] in #pid# may not always be accurate, which is not a problem of the command but depends on target application design.
Some applications do behave that they do not keep the started process, but just open a TAB in an already open process. Examples are IE, or Chrome WEB-Browser. In such cases you may need to first close all running instances using CAW. or KPN. and then start a new instance. In that case the #pid# will be the true PID of that instance.

For example when doing things like this, you will successfully open the browser(-Tab), but get only a zero as #pid#:

```
EXO.www.fa2.de||||(16+1)
MBX.#pid#
```

Error-messages:
It may be a good idea to enable error-messages, If you add **+32** to the **P5** flags, the command will display interactive User Interfaces and also **Error-Messages**, no matter if the command USM.[786] is specified or not.

Zone-Check for downloaded files:
This command bypasses the "Zone Check" for files that have been downloaded from the internet.

Shortcut commands:
EXO. may work with some of these (depending on the OS):

```
EXO.system.ini
EXO.control
EXO.write
EXO.pbrush
EXO.winword
EXO.excel

' Start NetMeeting (XP)
EXO.conf
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EXO. uses the default search path and will understand
following shortcuts to these executables in the window-folder.

'  Will open system control panel (XP, Win 7)
EXO.control

' Will open "Control Panel\All Control Panel
Items\Administrative Tools"
' XP, Win7
EXO.control admintools

' WIll open the  Clipboard Viewer
' XP only
EXO.clipbrd

' Systemconsole Prompt
' XP, Win7
EXO.CMD

' NotePad
' XP, Win7
EXO.notepad

' Windows Explorer
' XP, Win7
EXO.explorer

' XP, Win7
EXO.mspaint

' Open RegEdit
' XP, WIn7
EXO.regedit


' Windows Soundrecorder
' XP only
EXO.sndrec32

' Windows System Editor
' XP only
EXO.sysedit

You can use EXS. to open folders:

EXO.?ws\
EXO.?wi\
```

Limitations:

This command requires Windows XP, SP1 or higher.

See also:

- EXF. - Execute Fast 885

© 2023 Theo Gottwald * https://www.smart-package.com

- EXS. - Execute Shell 905
- EXF. - Execute Fast 885
- EXH. - Execute-Handle 890
- GEP. - Get-Executable-Path 2582

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.11 EXR. - Executer-from-Registry

MiniRobotLanguage (MRL)

EXR. Command

Execute from Registry

Intention

Starts a File or Document.

`EXR.` will try to deliver #pid# 137 and `#tid#`.
`EXR.` will only start files with a registered, valid File-Extension.

You can start a document, like this:
`EXR.?desktop\MR.txt`

or an executable:

```
EXR.?pfad\XYZ.msi
EXR.?pfad\TestApp.exe
```

You can specify a window-size-flag, however this will only work, if the application supports this.

If `EXR.` can not start the file, the timeout-flag #tio# 334 will be set to 1, else it will be reset to 0.

Syntax

EXR.P1[|P2]

Parameter Explanation

P1 - executable or Document-Filename or a hyperlink, ending with a valid registered file-ending.

P2 - (optional) Window Size Parameter

```
    0 - HIDE    -   Hides the window and activates another
    window.
    1 - SHOW NORMAL   Activates and displays a window.
    2 - SHOW MINIMIZED - Activates the window and displays it
    as a
        minimized window.
    3 - MAXIMIZE -   Maximizes the specified window.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
4 - SHOW-NO-ACTIVATE - Displays a window in its
    most recent Size and position.
    The active window remains active.
5 - SHOW - Activates the window and displays it
    in its current size and position.
6 - MINIMIZE - Minimizes the specified window and
    activates the next top-level window in the Z order.
7 - SHOW-MIN-NO-ACTIVE - Displays the window as
    a minimized window. The active window remains active.
8  - SHOW-NO-ACTIVE - Displays the window in its current
state.
      The active window remains active.
9 - RESTORE - Activates and displays the window.
10 - SHOW-DEFAULT - Sets the SHOW STATE to default.
```

Example

```
'*********************************************
' Example 1 - without Windows-Parameter
'*********************************************
VAR.$$LNK=http://www.fa2.de
EXR.?desktop\MR.txt
:over
MBX.Halt to read output
'*********************************************


'*********************************************
' Example 2 - using Windows-Parameter
'*********************************************'
' EXR. - Beispiel
DBM.2
' Start document minimized
: §§FIL=?desktop\MyWord.doc
EXR.§§FIL|6
DMP

:over
MBX.Halt to read output
'*********************************************
```

Remarks

Technical remark: `EXR.` uses Create-Process-FindExecutable-Technology and is compatible with other commands that use this technology.
`EXR.` can be blocked by a process that it has started. If this turns to be a problem, choose `EXA.` or `EXF.` as alternative.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

There may be cases where EXR. can not deliver the proper PID 137/ TID. This depends on the target application, and may be influenced by the fact, if the target application is already running.

See also:

- GEP. - Get-Executable-Path 2582
- **EXS. - Execute Shell** 905
- EXF. - Execute Fast 885
- EXH. - Execute-Handle 890
- EXO. - Execute Open Document 895

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.12 EXS. - Execute Shell

MiniRobotLanguage (MRL)

EXS. Command

Execute Shell

Intention

Starts a file or document like when you click on it with the mouse. `EXS.` will not only start executable files. It will also **start any registered file**, like a office-document or text-file.

`EXS.` will open hyperlinks and even download-links. This will both work as expected:

```
EXS.http://www.heise.de/
EXS.www.heise.de/
```

The document will be started somehow like if you have clicked on it with the mouse.

```
' Open a document
EXS.?desktop345\MyActions.txt

' Open a system folder, like you click on a folder-icon
EXS.?ws\345
```

Disadvantage from that system is, that it does NOT deliver `#pid#` or `#tid#`.

`EXS.` is technically similar to `EXF.` (Execute Fast). It will however be less fast.
In result, `EXS.` is able to return an error code if there was a problem, while `EXF.` will not.

`EXS.` will return an error code on the TOS 369, if there was a problem. If there was no problem, `EXS.` will return a zero on the TOS.

If you plan to start a lot of executable files very fast, or if the executable that you start blocks the script when you use `EXS.`, then prefer using `EXF.`

If you start a lot of executable files, `EXS.` is safe and slower, because it waits(/blocks the script-execution) until all internal system processes that start the file are running.

`EXF.` will just be very fast and start whatever you want as fast as your system can do.
`EXF.` unlike `EXS.` breaks through all API-limits (and can even crash a process if used careless). Try this example using `EXS.` or `EXF.` and see the difference.

```
KPN.iexplore.exe|a
FOR.$$CNT|1|20
  EXS.iexplore|www.fa2.de
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
NEX.
STS.DUMP
MBX.Click to Close all IE's
KPN.iexplore.exe|a
ENR.
```

however if you want to open many TAB's in your browser, `EXF.` is just faster: Try this:

```
' Open your daily browser
EXF.http://dict.leo.org
EXF.http://www.facebook.com
EXF.http://www.heise.de/
EXF.www.fa2.de
ENR.
```

Syntax

EXS.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - Executable or Document-Filename (can be a hyperlink)
P2 - ***(optional)*** Parameter for command-line
P3 - ***(optional)*** Path/Folder for command, if omitted current folder is used
P4 - ***(optional)*** Window Size Parameter, if omitted NORMAL is the default size

You can use the following window-size parameters for P4

```
0 - HIDE    -   Hides the window and activates another
window.
1 - SHOW NORMAL   Activates and displays a window.
2 - SHOW MINIMIZED - Activates the window and displays it
as a
    minimized window.
3 - MAXIMIZE -   Maximizes the specified window.
4 - SHOW-NO-ACTIVATE - Displays a window in its
    most recent Size and position.
    The active window remains active.
5 - SHOW - Activates the window and displays it
    in its current size and position.
6 - MINIMIZE - Minimizes the specified window and
    activates the next top-level window in the Z order.
7 - SHOW-MIN-NO-ACTIVE - Displays the window as
    a minimized window. The active window remains active.
8  - SHOW-NO-ACTIVE - Displays the window in its current
state.
    The active window remains active.
9 - RESTORE - Activates and displays the window.
10 - SHOW-DEFAULT - Sets the SHOW STATE to default.
```

Error-codes from EXS.

© 2023 Theo Gottwald * https://www.smart-package.com

```
   2       - The system cannot find the file specified.
   3       - The system cannot find the path specified.
   5       - Access to that file was denied.
   8       - Not enough memory is available to process this command.
   11      - The .EXE file is invalid (non-Win32 .EXE).
   15      - The system cannot find the drive specified.
   27      - The filename association is incomplete or invalid.
   28 - 30   DDE-Errors
   31      - This file-ending is not registered
   32      - DLL is missing.
```

Example

```
'***********************************************
' Example 1 - without Windows-Parameter
'***********************************************

' Example1
DBM.2
' Vars and special folders can be used.
' Don`t wait = 100 + 3 for maximize window = 103
EXS.?desktop\MR.txt|103
DMP.
END.

'***********************************************
' Example 2 - using Windows-Parameter
'***********************************************
' Example 2
DBM.2
' Variablen und Spezialordner können verwendet werden.
: §§FIL=?desktop\MyWord.doc
EXS.§§FIL|§§WIN
DMP.
END.
```

Remarks

Technical remark: `EXS.` uses ShellExecute-Technology and is compatible with other commands that use this technology. `EXS.` can be blocked by a process that it has started.

`EXS.` may work with some of these (depending on the OS):

```
EXS.system.ini
EXS.control
EXS.write
EXS.pbrush
EXS.winword
EXS.excel

' Start NetMeeting (XP)
EXS.conf
```


© 2023 Theo Gottwald * https://www.smart-package.com

etc.

```
EXS. uses the default search path and will understand
following shortcuts to these executables in the window-folder.

'  Will open system control panel (XP, Win 7)
EXS.control

' Will open "Control Panel\All Control Panel
Items\Administrative Tools"
' XP, Win7
EXS.control admintools

' WIll open the  Clipboard Viewer
' XP only
EXS.clipbrd

' Systemconsole Prompt
' XP, Win7
EXS.CMD


' NotePad
' XP, Win7
EXS.notepad

' Windows Explorer
' XP, Win7
EXS.explorer

' XP, Win7
EXS.mspaint

' Open RegEdit
' XP, WIn7
EXS.regedit


' Windows Soundrecorder
' XP only
EXS.sndrec32

' Windows System Editor
' XP only
EXS.sysedit

You can use EXS. to open folders:

EXS.?ws\
EXS.?wi\
```

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

`EXF`. does deliver #pid# 137 or `#tid#`. It does not deliver an error-code in case something did not work.

See also:

- EXF. - Execute Fast 885
- EXR. - Executer-from-Registry 902
- EXH. - Execute-Handle 890
- **GEP. - Get-Executable-Path** 2582
- EXO. - Execute Open Document 895
- **EXM. - Execute MSI** 892

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.52.13 EXU. - Execute Uninstall

MiniRobotLanguage (MRL)

EXU. Command

Execute Uninstall

Intention

This command is be used to start an Uninstallation, just like if you call "Startmenu/Controlpannel/Programs/Uninstall".

It will retrieve the complete Application name and start the Uninstallation. Standard-Search Pattern can be used.

Please be careful using Patterns, as the first fitting Uninstall-Entry is started.

If you want to wait until the started process ends, use WPT. - Wait-Process-Termination 2913. with the PID 137 that has been delivered by `EXU. in #pid#`.

```
' Wait until the started application ends
VAR.$$LNK=Paint 1.54
EXU.$$LNK|s
WPT.#pid#
ENR.
```

Syntax

EXU.P1[|P2]

Parameter Explanation

P1 **-** Application to start uninstall/modify
P2 **-** letter specifies what to look for. Details see below.

```
s - return command-line for silent Uninstallation
u - return command-line for standard Uninstallation.
m - return command-line for standard Modify-Path
   (Add-/Remove Setup)
a - return Standard Application Name
```

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'*********************************************
' EXU. - Sample
'*********************************************
DBM.2
'PRT.=========================================
: §§MYA=quote|u
GII.§§MYA|§§NAM|a
PRT. Application-name is: §§NAM
EXU.§§NAM
PRT.=========================================

:over
MBX.Halt to read output

END.
```

Remarks

The command works transparently with both x32 and x64 applications.

Technical remark: `EXU.` uses asynchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

Limitations:

-

See also:

- IAI. / NAI. - If-Application-Installed 2601
- GII. - Get-Installation-Information 2584
- EXH. - Execute-Handle 890
- EXS. - Execute Shell 905
- EXF. - Execute File/Document 885

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.14 EXV. - Execute via System-Desktop

MiniRobotLanguage (MRL)

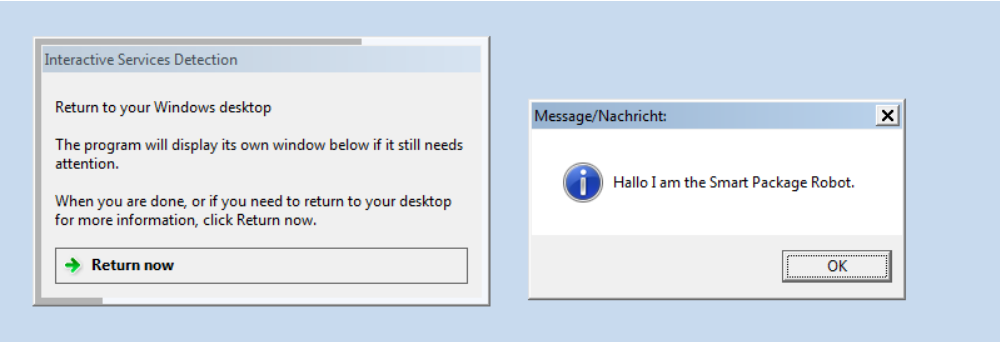
EXV. - Execute via System-Desktop

Execute Blocking via System-Desktop.

Intention

Run an executable on the hidden System-Desktop synchronously and deliver the exit-code.
The current script is suspended until the called program ends.

What is the System-Desktop?
Imagine there is a hidden place in your computer, where you can execute all your automation scripts, and the user can not see it,
because its on another desktop. You can even click on Buttons, using the Mouse!



Notice that this picture was made using the following Script using the Compiler-directive '"#EXV:", NOT using the command `EXV.`

```
'#EXV:itsme
 MBX.Hallo, I am the Smart Package Robot.
 ENR.
```

Generally for most tasks, the Compiler-directive `'#EXV:` is more useful then using the `EXV.`-command.
`EXV.` as a command will only start another task on the "Hidden desktop". The robot itself will stay on the normal user-desktop.
Using the compiler-directive `'#EXV:`, the robot will start on the "Hidden Desktop".

How can I see what is going on, on the "Hidden Desktop"?

You can switch to the "Hidden Desktop" using the `EXV.`-Command.

*© 2023 Theo Gottwald * https://www.smart-package.com*

For more Details see here "Switch to the Hidden Desktop 867".

Will I get an Exit-Code from the Executable that i have started?

The command EXV. will return an **Exitcode** (Error-level, Returncode) of the started application/command-line will be **placed on the stack** (TOS 369) .
General Usage of EXV. is as simple as it can be:

```
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXV.$$LNK|c
```

You should always add the "|c" at the end of the command, or you can specify the "Cleanup" in a separate command like this:

```
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXV.$$LNK
PAU.10
EXV.|c
ENR.
```

The EXV. -command is blocking the script-execution as long, as the started program is alive.
If you need to do something while the application was started, use PRR. / PRE.
These commands will make the EXV. run in a parallel process and will therefore not block your script.
You can still access the return code using WRS. and GRS.
Like in this example:

```
PRR.
 VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
 EXV.$$LNK
 PRS.$$000
 ENR.
PRE.
$$ROA=$$001
' Wait for Parallel-Process to end
WRS.$$ROA
' Get Result from EXV.
GRS.$$ROA
ENR.
```

Syntax

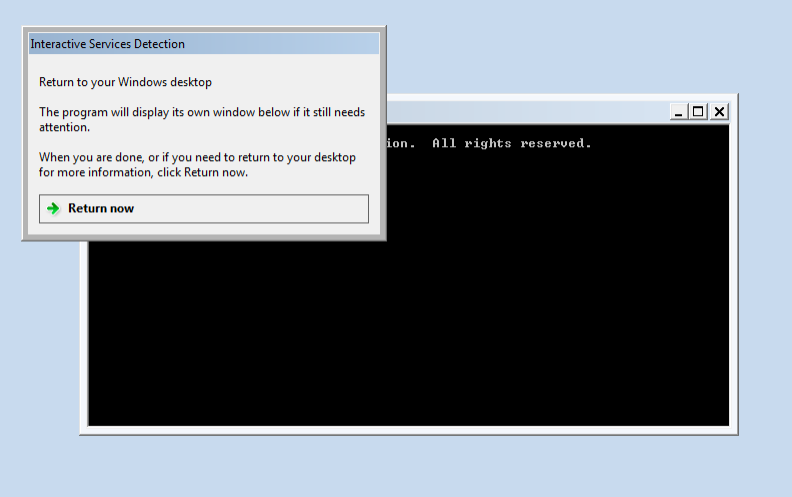
EXV.P1[|P2]

Parameter Explanation

***P1* - *[optional] Filename & Path* of the executable that shall be started on the Hidden Desktop.**

© 2023 Theo Gottwald * https://www.smart-package.com

If P1 is "-", just a commandline is started on the hidden desktop.(See picture below).
This will enable you to take a look on the Hidden Desktop 867.



P2 - [optional] can be one of two Parameters "c" or "p":
c - cleanup
p - prepare

c - will cleanup some internal structures that have been needed to start the executable.

d - disable switching to "Hidden Desktop". This is the opposite to "p - prepare". THis will change your system settings in a way that you can not switch to the Hidden Desktop.

p - prepares your system in a way that you can switch to the Hidden Desktop.
Espeacially under Windows 8 it may not possible to switch to the "Hidden Desktop" without first using
EXV.|p

All three options (c,d and p) can be combined or used alone.

EXV. returns:
The Exitcode (Errorlevel, Returncode) of the started Application/comandline will be placed on the stack (TOS 369).

Example

```
'************************************************
' Example 1
'************************************************
PRR.
 VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
 EXV.$$LNK
 PRS.$$000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 ENR.
PRE.
$$ROA=$$001
' Wait for Parallel-Process to end
WRS.$$ROA
' Get Result from EXV.
GRS.$$ROA
ENR.
ENR.'*******************************************
```

Remarks

Technical remark: `EXV.` uses synchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.
`EXV.` will suspend the script process until the started process has ended. The script process is not waiting, its suspended.
The suspended script process does not need any CPU resources, but also does not react on any message or whatever.

Generally for most tasks, the Compiler-directive `'#EXV:` is more useful then using the `EXV.`-command.
`EXV.` as a command will only start another task on the "Hidden desktop". The robot itself will stay on the normal user-desktop.
Using the compiler-directive `'#EXV:`, the robot will start on the "Hidden Desktop".

Limitations:

Hint: This Feature requires Windows Vista or higher.

See also:

- Execute-Commands 867
- '#EXV: - Compiler-Directive 280
- HDC. - Hidden Desktop Control 2664
- · Parallel Robot Operations 2632
- · Background-Operations 421
- · 1.7. Application and PID 137
- · 3.3 Specialfolders 345
- · EXX. - Execute-Extended 916
- · EXE. - Execute Programm 883
- · EXH. - Execute-Handle 890
- · EXA. - Execute Asynchron 872
- · EXO. - Execute Open Document 895
- · **IEF. / NEF. - If - exist - File** 1050
- · **IEP. / NEP. - If-Exists-Process** 2900

© 2023 Theo Gottwald * https://www.smart-package.com

3.52.15 EXX. - Execute-Extended

MiniRobotLanguage (MRL)

EXX. Command

Execute Extended

Intention

Start a program.

`EXX.` will start a program asynchronous. It will not wait until the program has ended.

`EXX.` uses an internal "Wait for Input Idle" (`WII.`) to wait for a window to appear. For this, it uses the default WII.-timeout (taken from the last WII.-call).

Therefore EXX. may deliver a window handle on TOS for some programs, or may just deliver a zero. This depends on the started software as well as other factors, like the WII.-timeout. If you need a handle to be delivered on stack, prefer `EXH`.

Example for this:

```
CAW.RegEdit_RegEdit|Reg&AND:Edit
EXX.regedit
STS.DUMP
MBX.Ready
```

This example will deliver the handle of the RegEdit-Window on the stack. As long, as RegEdit has not been started before, and the window was there before. You can use CAW. 3586 to prevent that case. See example.

`EXX.` will deliver #pid# 137 but will set `#tid#` to zero.

You can supply a second, numerical parameter P2 that may influence the size of the started application window, if the started application supports that. This way you can start an application minimized. Like this:

```
' Will start the Furmark minimized.
EXX.?progs\Geeks3D\Benchmarks\FurMark_1.9.1\FurMark.exe|2
```

If you add +100 to that second parameter, then `EXX.` will NOT even try to wait for the window of the started application to appear and therefore will also not deliver a window handle on TOS 369.

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

EXX.P1[|P2]

Parameter Explanation

P1 - Document-Filename (can be a hyperlink)
P2 - (optional) Window Size Parameter
When you add +100 to the window size, the command will not wait for the window to appear and yet not be able to deliver the window handle.

You can use the following window-size parameters

```
0 - HIDE    -   Hides the window and activates another
window.
1 - SHOW NORMAL   Activates and displays a window.
2 - SHOW MINIMIZED - Activates the window and displays it
as a
    minimized window.
3 - MAXIMIZE -   Maximizes the specified window.
4 - SHOW-NO-ACTIVATE - Displays a window in its
    most recent Size and position.
    The active window remains active.
5 - SHOW - Activates the window and displays it
    in its current size and position.
6 - MINIMIZE - Minimizes the specified window and
    activates the next top-level window in the Z order.
7 - SHOW-MIN-NO-ACTIVE - Displays the window as
    a minimized window. The active window remains active.
8  - SHOW-NO-ACTIVE - Displays the window in its current
state.
      The active window remains active.
9 - RESTORE - Activates and displays the window.
10 - SHOW-DEFAULT - Sets the SHOW STATE to default.
```

Example

```
'*********************************************
' Example 1 - EXX.
'*********************************************

' Example1
DBM.2
' Vars and special folders can be used.

' Will start the Furmark minimized.
EXX.?progs\Geeks3D\Benchmarks\FurMark_1.9.1\FurMark.exe|2

' Will start the FUrMark minimized and without waiting
EXX.?progs\Geeks3D\Benchmarks\FurMark_1.9.1\FurMark.exe|
(2+100)
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*********************************************
' Example 2 - EXX.
'*********************************************
'
CAW.RegEdit_RegEdit|Reg&AND:Edit
EXX.regedit
STS.DUMP
MBX.Ready
```

Remarks

Technical remark: `EXX.` uses asynchronous Shell- / Create-Process-Technology and is compatible with other commands that use this technology.

```
EXX. uses the default search path and will understand
following shortcuts to these executables in the window-folder.

'  Will open system control panel (XP, Win 7)
EXX.control

' Will open "Control Panel\All Control Panel
Items\Administrative Tools"
' XP, Win7
EXX.control admintools

' WIll open the  Clipboard Viewer
' XP only
EXX.clipbrd

' Systemconsole Prompt
' XP, Win7
EXX.CMD


' NotePad
' XP, Win7
EXX.notepad

' Windows Explorer
' XP, Win7
EXX.explorer

' XP, Win7
EXX.mspaint

' Open RegEdit
' XP, WIn7
EXX.regedit


' Windows Soundrecorder
' XP only
EXX.sndrec32
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Windows System Editor
' XP only
EXX.sysedit
```

You can not use EXX. to open a folder. Use EXS. or EXF. for that.
If you do this:

```
EXS.?ws\
EXS.?wi\
```

you can see the executable files in your system folders. They can be started using EXX. like see above.

Limitations:

EXX. will start a program asynchronous. It will not wait until the program has ended. In some cases it can still happen that it will wait, or that it will wait too long for #pid# to be delivered. Use EXA. in these cases, as EXA. can not be blocked, yet may also not deliver #pid#.

See also:

- EXH. - Execute-Handle 890
- EXS. - Execute Shell 905
- EXF. - Execute Fast 885

© 2023 Theo Gottwald * https://www.smart-package.com

3.53 FEX. - Enumerations

3.53.1 ! Smarty's FEX-Commands

Smarty's FEX-Commands

Using `FEX`, you can make Smarty count for you. There are many things she will count for you:

- Files or Folders

*© 2023 Theo Gottwald * https://www.smart-package.com*

- Lines in a text file
- Lines in a variable that contains a text-file
- processes
- parts of a delimited string (`FEM.`)
- and much more.

And Smarty will do more.

She will execute tasks that you define, FOR EACH of the elements that she is counting.
This way she can save you a lot of work!

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.53.2 EFX. - Exit - FEX

MiniRobotLanguage (MRL)

EFX. Statement

Exit FEX Command

Intention

Stops the current Enumeration of any of the following "FEX" commands:

`FEF. FEL. FEM. FEP. FER. FEV. FES.`

In nested FEX-commands, only the actual enumeration is been stopped.

`EFX.` does not need or accept any parameter.

Syntax

EFX.

Parameter Explanation

There is no Parameter

Example

```
'*********************************************
' EFX-Sample
'*********************************************
DBM.2
: $$PAT=Ralf Rolf Peter Franz Jacob
: $$TAR=Monika Sabine Pauline June Sabine

FEM.:Lev1|$$VAR|$$TAR
DMP.
MBX.Ready
END.

'------------------------------------------------------------
:Lev1
'DBP.-Level 1 ($$VAR)

FEM.:Lev2|$$BAR|$$PAT
RET.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'-------------------------------------------------------------
:Lev2
'DBP.--Level 2 ($$BAR)

DBP.Now is $$VAR + $$BAR
IVS.$$VAR=June
  IVS.$$BAR=Peter
    EFX.
  EIF.
EIF.
RET.
'-
```

Remarks

`EFX.` is a member of the FEX-family "For Each X". They are used for easy enumeration.

Limitations:

-

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- FEF. - For-Each-File(/Dir) 924
- FEL - For-Each-Line 927
- FEM. - For Each Member 929
- FEP. - For-Each-Process 932
- FER. - For-Each-Registry 935
- FEV. - For-Each-Variable(-Line) 937

© 2023 Theo Gottwald * https://www.smart-package.com

3.53.3 FEF. - For-Each-File(/Dir)

MiniRobotLanguage (MRL)

FEF. Enumeration Command

For each File (directory)

Intention

You want to make a list with all files in a folder? Print those hundreds of files? Or join them in one file? Or run a scan over all dirs in a drive or folder?

All these and many more jobs can be done using `FEF`.

Generally `FEF.` will enumerate and count Files and/or Directories from a given source path. After all this function leaves up to four information on the Stack (see below).

These are:

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->QuadInt       :1091
Stackpos: 001 ->QuadInt       :16827
Stackpos: 002 ->QuadInt       :171670813215
Stackpos: 003 ->QuadInt       :15736

-----------------------------------
```

In detail the four values inform you about this:

1. Number of directories
2. Number of files and directories together
3. Bytes in all files, total
4. Number of files in all folders

Syntax

FEF.P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 - `Label where the for each will jump "for each element"`
P2 - `Temp. Variable, updated by FEX each round.`
P3 - `Root-Directory`

*© 2023 Theo Gottwald * https://www.smart-package.com*

P4 - (optional) Filepattern. If omitted, ****.**** is used.
P5 - (optional) Attribute, if omitted 69 is used. (Recurse, Files, Cooperative)

```
    Attributes-Parameter

     1 - Recurse into Subfolders
     2 - Hidden files and folder, too.
     4 - Files
     8 - Dirs
    16 - SizeDepth
    32 - CountAll
    64 - CoOperative Mode (Slower)
   128 - System Files, too.
   256 - Sort Downwards
```

Example

```
'*********************************************
' FEF-Sample
'*********************************************
' DBM.2
' Use an existing path here
: §§ROT=F:\00_Image
: §§PAT=*.*

FEF.:Tar|§§FIL|§§ROT|§§PAT|(1+4+8+64)
DMP.4


:over
MBX.Halt to read output

END.

:Tar
PRT.§§FIL
RET.
'*********************************************
```

Remarks

FEF. is a Member of the FEX-Family "For Each X". They are used for easy enumeration

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

The internal data storage can hold up to several hundred Megabytes of Directory data. Yet there is no warranty that there is not a file-system out there, that is too large in size to be fully enumerated.

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- IEF. / NEF. - If - exist - File 1050
- EFX. - Exit - FEX 922
- FEL - For-Each-Line 927
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.53.4 FEL. - For-Each-Line

MiniRobotLanguage (MRL)

FEL. Enumeration Command

For each Line (in a file)

Intention

Enumerate all lines in a file.

Syntax

FEL.P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 - Label where the for each will jump "for each element"
P2 - Temp. Variable, updated by FEX each round.
P3 - ***Filename***
P4 - ***(optional)*** Pattern for the Lines to enumerate. If omitted, ****.**** is used and all lines are enumerated.
P5 - ***(optional)*** maximum line-number to enumerate (stop-line)

Example

```
'*********************************************
' FEL-Sample
' Copies all line containing a C-Comment into
' a new Textfile.
'*********************************************
DBM.2
' Source-File (must exist)
VAR.$$FIL=?pfad\Orig.txt
' Targetfile
VAR.$$TAR=?pfad\Tar.txt
DEL.$$TAR
FEL.doit|$$LIN|$$FIL|{/*&ANDTHEN:*/}

:over
MBX.Halt to read output

END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:doit
VAR.$$LIN=$crlf$$$LIN
ATF.$$TAR|$$LIN
RET.
'*********************************************
```

Remarks

`FEL.` is a Member of the FEX-Family "For Each X". They are used for easy enumeration

Limitations:

-

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- IEF. / NEF. - If - exist - File 1050
- EFX. - Exit - FEX 922
- LOF. - Load-Text-File 1063
- LFT. - Line-From-Text 1060
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.53.5 FEM. - For Each Member

MiniRobotLanguage (MRL)

FEM. Enumeration Command

For each Member (string between delimiter)

Intention

Enumerate parts of a string. They must be delimited with a blank or other specified character.

Syntax

FEM.P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 - Label where the for each will jump "for each element"
P2 - Temp. Variable, updated by FEX each round.
P3 - ***(variable)*** source string that will be enumerated
P4 - ***(optional)*** delimiter between the parts of P3,
if omitted Space (ASCI 32) is used.
P5 - ***(optional)*** optional Standard Pattern

Example

```
'*********************************************
' Simple FEM.
'*********************************************
: $$TAR=Hallo Peter das ist ein Satz.
FEM.:Tar|$$VAR|$$TAR|$sp$

DMP.

:over
MBX.Halt to read output
END.

:Tar
PRT.$$VAR
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*********************************************
' Nested FEM.
' FEM./FEX can be nested to unlimited
' Depth.
'*********************************************
'DBM.2
: $$TAR=H!a!l!l!o P!e!t!er d!a!s i!s!t e!i!n S!a!t!z.
' We did not write the delimiter then $sp$ is used.
FEM.:Tar|$$VAR|$$TAR
DMP.
:over
MBX.Halt to read output
END.

:Tar
PRT.$$VAR
FEM.:Tab|$$BAR|$$VAR|!
RET.

:Tab
PRT.$$BAR
RET.


'*********************************************
' EFX-Sample
'*********************************************
DBM.2
: $$PAT=Ralf Rolf Peter Franz Jacob
: $$TAR=Monika Sabine Pauline June Sabine

FEM.:Lev1|$$VAR|$$TAR
DMP.
MBX.Ready
END.

'-------------------------------------------------------------
:Lev1
'DBP.-Level 1 ($$VAR)

FEM.:Lev2|$$BAR|$$PAT
RET.
'-------------------------------------------------------------
:Lev2
'DBP.--Level 2 ($$BAR)

DBP.Now is $$VAR + $$BAR
IVS.$$VAR=June
  IVS.$$BAR=Peter
    EFX.
  EIF.
EIF.
RET.
'-
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

`FEM`. is a Member of the FEX-Family "For Each X". They are used for easy enumeration

Limitations:

-

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- FEF. - For-Each-File(/Dir) 924
- FEL - For-Each-Line 927
- FEP. - For-Each-Process 932
- FEV. - For-Each-Variable(-Line) 937

© 2023 Theo Gottwald * https://www.smart-package.com

3.53.6 FEP. - For-Each-Process

MiniRobotLanguage (MRL)

FEP. Enumeration Command

For each process

Intention

Enumerate (count ...) all running EXE's (x32 and x64).

Process a script or subprogram for each Exe. You can also specify a 3.2 Standard-Search Pattern 341, this is applied to the filename of the loaded executable. If given only those executables are enumerated, which fit to the specified pattern.

You get in the Variable P2 the name of the EXE with or without a path. Additionally #pid# is set to the PID (Process ID) of the enumerated EXE. Using the PID you will be able to get any further needed Information. For example using GDF. or GPI.

Syntax

FEP.P1|P2|P3[|P4]

Parameter Explanation

P1 - Label where the for each will jump "for each element"
P2 - Temp. Variable, updated by FEX each round.
P3 - Flag can be:

```
' 1,3,5 - Enum loaded EXE's
' 2,4,6 - Enum loaded DLL's (32 bit only)
' +0 give only exe/dll-name
' +2 give full path instead
' +4 give PID of parent process instead
' #pid# is always set
```

P4 - optional Standard-Pattern, is always applied to ONLY the name of the exe (without path). You can omit P4, then ALL Exe's are enumerated.

Example

```
'*******************************************
' FEP-Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Shows all running processes with PID
'*********************************************

FEP.doit|$$FIL|1
MBX.Ready
ENR.
'---------------
:doit
'GDF.pf|
DBP.Filename is:$$FIL    PID is: #pid#
RET.
'---------------
'*********************************************

'*********************************************
' FEP-Sample
' Shows different Flags for enum of executables
'*********************************************
FOR.$$FLA|1|6|2
  MBX. Enumerating with Flag=$$FLA
  : $$CNT=0
  FEP.doit|$$FIL|$$FLA
NEX.

MBX.Ready
ENR.

:doit
VIC.$$CNT
DBP.$$CNT. [PID=#pid#]  $$FIL
RET.
```

Remarks

`FEP.` is a Member of the FEX-Family "For Each X". They are used for easy enumeration of system objects like for example windows, processes or files.

Limitations:

The enumeration of executables will work for all executables, including x64 executables. The enumeration of DLL's is experimental and only for x32.

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- GPI. - Get Process Information 2893
- IEP. / NEP. - If-Exists-Process 2900
- GDF. - Get-Data-From 2889
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913

© 2023 Theo Gottwald * https://www.smart-package.com

- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.53.7 FER. - For-Each-Registry

MiniRobotLanguage (MRL)

FER. Enumeration Command

For each Registry

Intention

Enumerate registry entries. These can be Registry-keys or values.

The result is delivered as a Array which is placed in the FEX-Variable.

Use GFB. - Get-From-Block 680 to extract the single Blocki-Elements (1-6) out of the FEX-Variable. (see example)

Syntax

FER.P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 - Label where the for each will jump "for each element"
P2 - Temp. Variable, updated by FEX each round.

P3 - Start Registry Path
P4 - optional 1,2,4,8 or 16, default is 3

```
   1 - only get keys
   2 - only get values
   4 - use Pattern with actual key
   8 - use Pattern with actual value
  16 - use Pattern for Entry-Name

Values of  3 and 0 - are equal, therefore 3 is default.
```

P5 - optional Standard-Pattern

Example

```
'*********************************************
' FER. - Sample
'*********************************************
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'DBM.2
VAR.$$REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curren
tVersion\Uninstall\
FER.doit|$$BLO|$$REG|2
:over
MBX.Halt to read output
END.

:doit

GFB.§§REA|§§REB|§§REC|§§RED|§§REE|§§REF
DBP.§§REA$crlf$§§REB
RET.
'*********************************************
```

Remarks

`FER.` is a Member of the FEX-Family "For Each X". They are used for easy enumeration.

Limitations:

This command is currently limited to the x32-Registry.

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- IER. / NER. - If-Exist-Registry 2918
- FER. - For-Each-Registry 935
- RER. - Registry Entry Read 2925
- REW. - Registry-Entry-Write 2928
- RDI. - Registry-Delete-Item 2921
- RDT. - Registry-DelTree 2923

© 2023 Theo Gottwald * https://www.smart-package.com

3.53.8 FEV. - For-Each-Variable(-Line)

MiniRobotLanguage (MRL)

FEV. Enumeration Command

For each Variable

Intention

Enumerate all lines in a given variable.

Checks first if the Lines are delimited with CR,LF or CRLF and chooses the right Line-Separator.

If you load a complete Text-file into a variable using `CFF.` (Copy from file), you can go through each line of the text very fast, using `FEV.`

As `FEV.` work in Memory, its much faster then `FEL.` (For each line) that works on the file-system.

Syntax

FEV.P1|P2|P3[|P4]

Parameter Explanation

P1 - Label where the for each will jump "for each element"
P2 - Temp. Variable, updated by FEX each round.
P3 - Variable with Text
P4 - optional Standard-Pattern

Example

```
'*********************************************
' FEV-Sample
'*********************************************
VAR.$$FIL=?desktop\2Line.txt
CFF.$$FIL|$$FIA
PRT.$$FIA
PRT.**************************************
FEV.:Tar|$$VAR|$$FIA|Line
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
:over
MBX.Halt to read output
END.


:Tar
PRT.$$VAR
RET.


'*****************************************************
' EXAMPLE 2: FEV./CTI./CFI. Enumerate INI-File
'*****************************************************
VAR.§§FIL=?desktop\one.ini
VAR.§§SEC=Section
VAR.§§KEY=Key
VAR.$$BLO=ABCDEFGHIJ
VAR.§§VAL=MyValue

' First we write to a ini-file
FOR.§§CNT|1|3
  : §§SEA=§§SEC_§§CNT
  FOR.§§CNX|1|5
    DBP.§§SEA
    CTI.§§FIL|§§SEA|§§KEY_§§CNX|§§VAL#dtime#|§§RES
    IVV.§§RES=1
      DBP. Ini was written with success!
    ELS.
      DBP. Ini was NOT written with success!
    EIF.
  NEX.
NEX.
' Now we read all Section-Names
CFI.§§FIL|§§MYS
PRT.Section-Names:$crlf$§§MYS

FEV.doit|§§SKA|§§MYS

MBX.Ready
ENR.

:doit
' Get all Keyf from that section
CFI.§§FIL|§§SEA|§§MYK
PRT.Key-Names from §§SKA:$crlf$§§MYK

' Enumerate all Keys also
PRT. Key-Enumeration:
FEV.doitb|§§SIK|§§MYK
RET.

:doitb
CFI.§§FIL|§§SKA|§§SIK|§§VAL
PRT. Key: §§SIK   Value=§§VAL
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

`FEV.` is a Member of the FEX-Family "For Each X". They are used for easy enumeration.

Limitations:

-

See also:

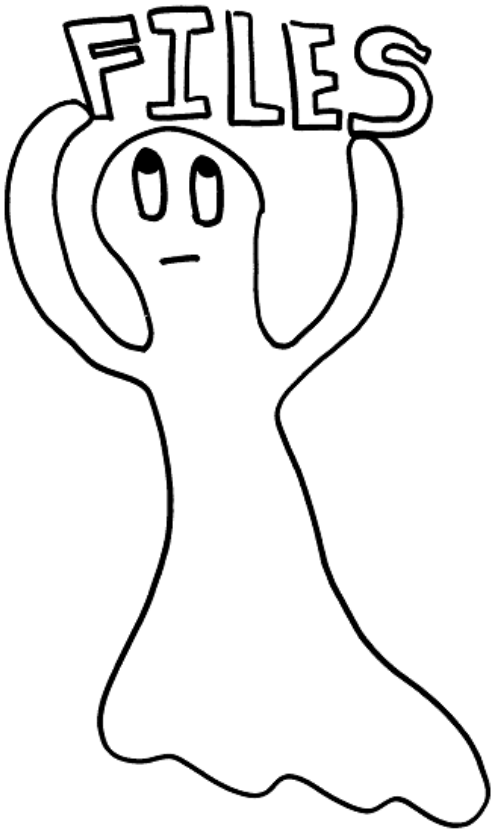
- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- FEF. - For-Each-File(/Dir) 924
- FEL - For-Each-Line 927
- FEM. - For Each Member 929
- FEP. - For-Each-Process 932
- FER. - For-Each-Registry 935

© 2023 Theo Gottwald * https://www.smart-package.com

3.54 Filesystem-Commands

3.54.1 ! Smarty's File-Commands

Smarty's File Commands



Smarty can work with files in any way you want.
You can:

- Write or append to Log-Files (`ATF.`)
- Write text- or binary files (`CTF.`)
- read text- or binary files (`CFF.`)
- change the actual directory and drive (`CHD.`)
- copy files (`COP.`)
- rename files (`REN.`)
- delete files (`DEL.`)
- Make directories (`MKD.`)
- delete directories (`RMD.`)
- check if a file exists (`IEF.`) or
- Wait until a file appears or disappears (`WFF.`)
- Find any Text or Wildcard Pattern in files using the sophisticates `FIF.` - command
- Split the path of a file into its parts

*© 2023 Theo Gottwald * https://www.smart-package.com*

- Split the Filename into its parts (GFT.)
- Get file-date and version
- work with INI-Files (CTI./CFI.)
- Split large files of any size to smaller files and (FIL.SPLIT FILE)
- Concatenate these files to the original file again (FIL.JOIN FILE)
- copy files using the Windows-Shell (SHF.)
- move files using the Windows-Shell (SHF.)
- copy, rename and move directories and / or files using the SHF.-command
- and much more ...

You can also apply these methods to groups of files, using FEF.

Some commands call internally the windows shell, these commands are limited to the normal path length of ~260 characters.
Most other commands directly call the NTFS-API (Advanced Programming Interface) and therefore support one or more of these features:

The NTFS-Direct Feature enables you to use the full power of the NTFS-Filesystem.
You can use Pathes up to 32760 Bytes!
Note that most window-programs may not be able to access files deeper then ~250 Bytes-Path-length.

If you see this Logo, then you can use "Wildcards" in the Filesystem-Commands.

These are the well-known "*" and "?" - Wildcards.

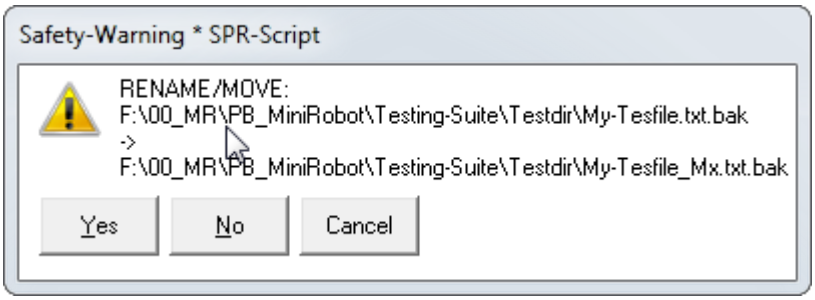
© 2023 Theo Gottwald * https://www.smart-package.com

Using Wildcards in the Pathname is not recommended, use them only in the Filename or extension.
Note that the first file or folder that fits to the wildcard is been taken.
Use Wildcards with extreme caution, as this may lead to unexpected results if there are multiple fitting files.

If you want to debug your Script safely, you can use the

```
OPT.SAFEFILE|1
```

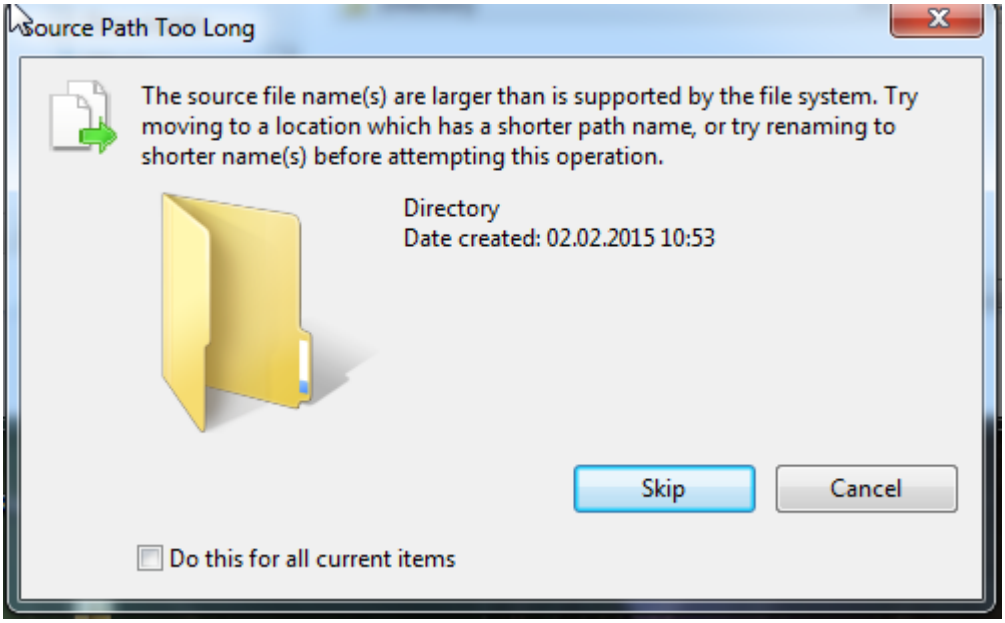
this will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`



You can choose between

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

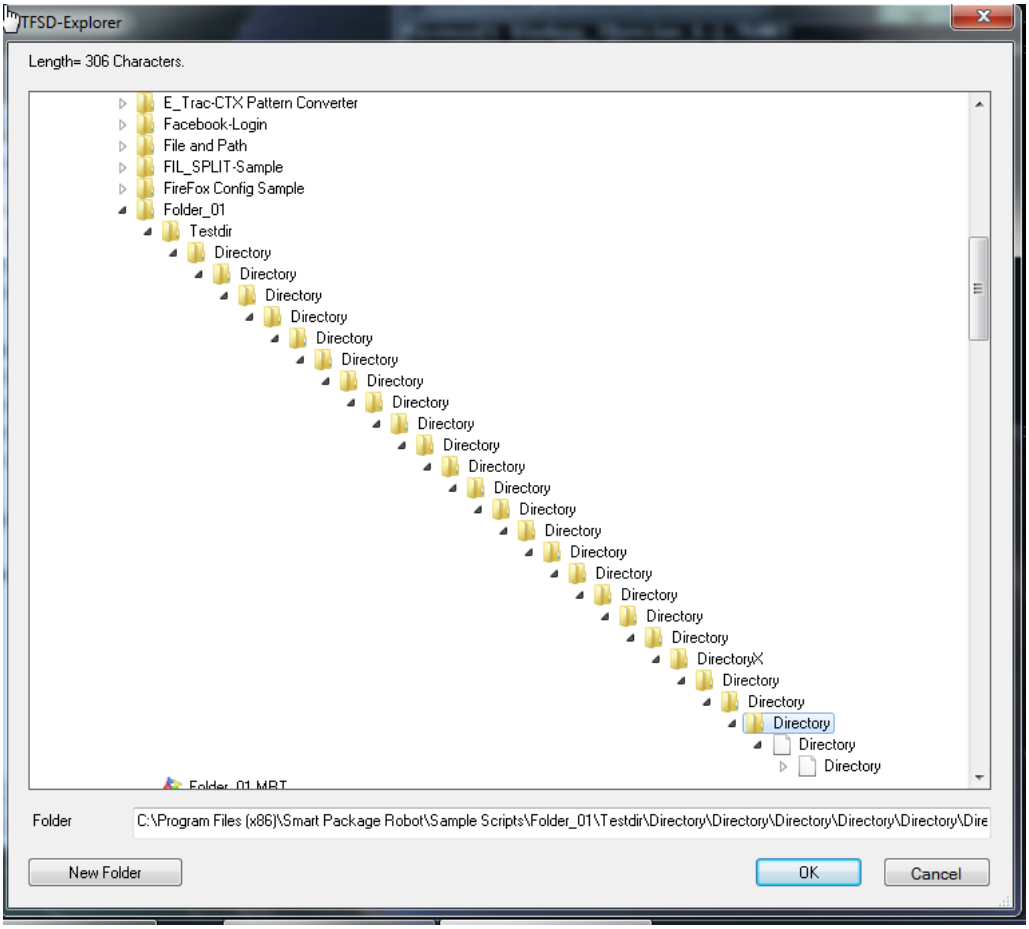
If you try to delete Pathes >260 characters with the Windows Explorer or Shell,
you may run into an error like the one below.
These Pathes can only be deleted, using the SPR-Command "`RMD.`".



Error-Message "Path longer then 260 character".

© 2023 Theo Gottwald * https://www.smart-package.com

Please note that the Explorer and Anti-Virus Tools may lock files and folders and prevent
the `RMD.`-Command from working as expected. In this case the Command will not delete the folder completely and the Timeout-Flag will be set.



You can use the supplied NTFSD-Explorer for simple File-Operations with very long pathes.
Note that Files may not have the proper Icons if you go deeper then 260 Characters.
This is not as bug its because the windows shell does not support this currently.

Here is a more complex Sample that will do a lot of file operations.

```
'
'SPR Script-file: FIF-Replacer
'Purpose: Find a Date in a Text-File and
'replace it with another Date
'Author: Theo Gottwald
'Creation date: 10-07-2014 at 14:28:09
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
$$FIL=?path\Test.txt
$$TAR=?path\Target.txt
' Delete it if it existed before
DEL.$$TAR
' We search for the line with the date
$$SRC=11.?7.*
' New date - we replace the old one with this
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   $$NEW=12.08.2015
   FIF.tww|$$FIL|$$SRC
   ' Check if we found something
   JIZ.$$000|Lab_nf
   ' Note that once we take something from TOS, it is
automatically removed
   $$LIN=$$000
   $$STA=$$000
   $$END=$$000
   $$STR=$$000
   $$TRE=$$000
   PRT.String: $$STR Found in Line: $$LIN
   PRT.From Character $$STA to $$END
   GLC.$$FIL|$$REA
   ' Line where String is minus 1
   CAL.$$TOL=$$LIN-1
   ' Copy all lines until the wanted Line
   FOR.$$COP|1|$$TOL
     LFF.$$FIL|$$COP|$$RES
      ' We use the delimiter that was found in the source file
     $$TXT=$$RES$$TRE
     ATF.$$TAR|$$TXT
   NEX.
      ' Here is the Line with the date
   LFF.$$FIL|$$LIN|$$TXA
   RPL.$$TXA|$$STR|$$NEW
   $$TXB=$$TXA$$TRE
   ATF.$$TAR|$$TXB
   ' Now copy Rest of file
   ' Line where String is plus 1
   CAL.$$TOL=$$LIN+1
   FOR.$$COP|$$TOL|$$REA
     LFF.$$FIL|$$COP|$$RES
      ' We use the delimiter that was found in the source file
     $$TXT=$$RES$$TRE
     ATF.$$TAR|$$TXT
   NEX.

   :Lab_nf
   MBX.!
   ENR.
   '============================================================



   ' BACKUP.Skript
   ' This Script will Backup 4 Folder including ALL Subfolders
   ' From Drive F: to Drive G:
   ' Compiled as EXE, it will run AS SYSTEM ('#EXV:) and
therefore
   ' survive an User-Logoff.
   ' It will also not be killed with the "KillRobots" or
CleanUp-Button because of the SYSTEM Rights.
   ' All Folders will be Backuped "in Parallel", the Script
will wait until all Backus are ended.
   '
   '#EXE:?path\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'#SPI:ForceWrite
'#MAN:ra
'#EXV:Robi

$$DAT=$date$
VAO.$$TIB=

$$NAM=Folder1
GSB.Lab_Copy

$$NAM=Folder2
GSB.Lab_Copy

$$NAM=Folder3
GSB.Lab_Copy

$$NAM=Folder4
GSB.Lab_Copy

WRS.!$$TIB
MBX.Ready
ENR.
'============================================================
:Lab_Copy
PRR.
  $$SRC=F:\$$NAM\
  $$TAR=g:\$$DAT\
  PRT.Dircopy:$$SRC->$$TAR
  MKD.$$TAR
  DCP.$$SRC|$$TAR
  PAU.15
  END.
PRE.$$TIC
$$TIB=$$TIB,$$TIC
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.54.2 ATF. - Append-to-File

MiniRobotLanguage (MRL)

ATF. Command

Append to File

Intention

This command is useful to write Log-Files. It will append any content (Text or binary) to an already existing file. Or, if the file does not already exist, it will create a new file.
`ATF.` does not append newline-characters `$crlf$` automatically. This has to be done manually if needed.

Syntax

ATF.[P1] [|P2] [|P3]

Parameter Explanation

P1 - File Name

P2 - Text or Data

P3 - Opt. Variable which receives 1/0 if operation was successful.

Example

```
'******************************************************
' EXAMPLE 2: ATF. Append with Newlines
' create a Logfile on yourdesktop
'******************************************************
'DBM.2
' This is the filename and path on the desktop
: $$FIL=?desktop\RND.txt

FOR.$$AAA|1|20
' Generate a Random Number between 124 and 123125
  RND.124,123125,$$REB
' Here comes the Newline
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
  VAR.$$REB=$date$ ** $time$> The Number is: $$REB$crlf$
' Append the Numbers to the File without Newline
  ATF.$$FIL|$$REB|$$ERG
  PRT.[$$AAA]->$$ERG
NEX.
MBX.Please see the File $$FIL on your desktop.
END.
'***********************************************
```

Remarks

Limitations:

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.3 CFF. - Copy-From-File

MiniRobotLanguage (MRL)

CFF. Command

Copy from File

Intention

This command is used to read any Text- or Binary Data from any file on into a variable.
It just like if you say "Put the content of this file into a variable". The syntax is very easy:

```
' put the filename into a variable
: $$FIL=C:\MyFilename.txt
' GEt the content of this file into $$CON
CFF.$$FIL|$$CON
```

Now the question is, did you get the file or not? To answer this question, there is a third parameter, which must be a variable. If you got the file, the content of this variable will be 1. If you did not get the file, the content of the variable will be 0. Also the timeout-flag will be set according to the success of the file-operation.

```
' put the filename into a variable
: $$FIL=C:\MyFilename.txt
' Get the content of this file into $$CON
CFF.$$FIL|$$CON|$$RES
IVV.$$RES=0
  MBX.I did not get the file!
EIF.
```

If you just specify a filename and you omit the path, it is assumed that you load the file from `?path\` - that is your project folder.

```
' put the filename into a variable
: $$FIL=ProjectConfig.ini
' Get the content of this file into $$CON
' the filename will internally be completed to
' ?path\MyFilename.txt
CFF.$$FIL|$$CON
```

And the limits?
The size of the file that is read, can be up to several hundred Megabytes. While the theoretical size limit of a variable is 2 GB, the real-life limit is set by windows, depending on how fragmented your memory is.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

CFF.[P1]|[P2]|[P3]

Parameter Explanation

P1 - Filename

P2 - Variable to insert data from file

P3 - Opt. Variable which receives a TRUE value when the operation was successful and a FALSE value when the operation failed.

Example

```
'*****************************************************
' EXAMPLE 1: CFF. Read from file
'*****************************************************
VAR.$$FIL=ProjectConfig.ini
IEF.?path\$$FIL
  CFF.$$FIL|$$VAR|$$RES
  MBX.$$RES
EIF.
MBX.Success: $$VAR
ENR.

'*****************************************************
' EXAMPLE 1: CFF. Read from file
'*****************************************************
VAR.§§FIL=?desktop\one.txt
CFF.§§FIL|§§DAT|§§RES
PRT. Success: §§RES - Data: §§DAT
' Read file, put on TOS
CFF.§§FIL
' Get TOS
PRT.$$000
CFF.§§FIL|§§DAT
PRT.§§DAT
END.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

While the theoretical limit of variables is 2 GB, in real life windows may not have so much string-memory available. This depends mostly on how fragmented your computer memory is. Under normal circumstances a file that is below 400 MB should be read into a variable with no problems.

See also:

- ATF - Append-to-File 946
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- 3.3 Specialfolders 345
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.4 CFI. - Copy-From_IniFile

MiniRobotLanguage (MRL)

CFI. Command

Copy from Inifile

Intention

This command is used to Read Entries from an INI-File. You can also read a full section from an INI-file. In case you want to read a whole INI-File, use the `CFF.`-command.
In short these are the four Modes of operation:

```
' Mode 1: Copies all Section-Names into Output-Var
CFI.Inifilename|Output-Var              '

' Mode 2: Copies all Keys to Output-Var
CFI.Inifilename|Section|Output-Var

' Mode 3: Copies the specified Key to the Output-Var
CFI.Inifilename|Section|Key|Output-Var

' Mode 4: Copies the given Key to Output-Var, succes-flag to
0/1
CFI.Inifilename|Section|Key|Output-Var|Success-Flag
```

Mode 3 is the one that is most often used. You can read a value or string from an INI-File into a given variable.

Mode 4 is an enhancement of Mode 3. It gives you a flag that shows you if the operation was successful or not. This makes sense, because the result of the operation would be the same, whether the operation failed or the value was empty. See the example in the example -section below. The syntax is like that:

```
VAR.$$FIL=?desktop\one.ini
VAR.$$SEC=Section
VAR.$$KEY=Key
CFI.$$FIL|$$SEC|$$KEY|$$VAL|$$RES
MBX. We got this value from the INI-File: $$VAL
```

In Mode 1, all returned Section Names are separated by `$crlf$`. Therefore you can easily enumerate them using `FEV`.

Note that the Timeout-flag is also set according to the success of the operation. See the example below for a complete example that writes and reads an INI-File.

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

CFI.P1|P2[|P3][|P4][|P5]

Parameter Explanation

P1 **-** Filename of Ini to write to
P2 **-** Section-Name
P3 **-** Key-Name*
P4 **-** ***(out)*** Value*
P5 **-** (out) Variable for Success-flag 1/0*

*Position of parameters differs according to Mode of operation, see explanation above. The last Parameter is always the return variable. See examples.

- CFI.Inifilename|Output-Var ' Copies all Section-Names
- CFI.Inifilename|Section|Output-Var ' Copies all Keys
- CFI.Inifilename|Section|Key|Output-Var
  ' Copies the given Key to Output-Var
- CFI.Inifilename|Section|Key|Output-Var|Success-Flag
  ' Copies the given Key to Output-Var, succes-flag to 0/1

Speed in Ticks:

This command uses typically around 4500 Ticks heavily depending on many factors like Filesystem-Speed and Parameters.

Example

```
'*****************************************************
' EXAMPLE 1: Enumerate INI-File
'*****************************************************
VAR.§§FIL=?desktop\one.ini
VAR.§§SEC=Section
VAR.§§KEY=Key
VAR.$$BLO=ABCDEFGHIJ
VAR.§§VAL=MyValue

' First we write to a ini-file
FOR.§§CNT|1|3
  : §§SEA=§§SEC_§§CNT
  FOR.§§CNX|1|5
    DBP.§§SEA
    CTI.§§FIL|§§SEA|§§KEY_§§CNX|§§VAL#dtime#|§§RES
    IVV.§§RES=1
      DBP. Ini was written with success!
    ELS.
      DBP. Ini was NOT written with success!
    EIF.
  NEX.
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Now we read all Section-Names
CFI.§§FIL|§§MYS
PRT.Section-Names:$crlf§§§MYS

FEV.doit|§§SKA|§§MYS

MBX.Ready
ENR.

:doit
' Get all Keys from that section
CFI.§§FIL|§§SEA|§§MYK
PRT.Key-Names from §§SKA:$crlf§§§MYK

' Enumerate all Keys also
PRT. Key-Enumeration:
FEV.doitb|§§SIK|§§MYK
RET.

:doitb
CFI.§§FIL|§§SKA|§§SIK|§§VAL
PRT. Key: §§SIK   Value=§§VAL
RET.

'*****************************************************
' EXAMPLE 2: CTI./CFI. Write/Read INI-File
'*****************************************************
VAR.§§FIL=?desktop\one.ini
VAR.§§SEC=Section
VAR.§§KEY=Key
VAR.$$BLO=ABCDEFGHIJ
' Generating a string of length ~80000 ...
§§VAL=$$BLO
FOR.$$CNT|1|12
  VAR.§§VAL=$$BLO§§VAL§§VAL
NEX.
GSS.1|65530|§§VAL|§§VAL
LEN.§§VAL|$$RES
DBP.Trying to write a string of Len=$$RES to INI.
' First we write to a ini-file
FOR.§§CNT|1|3
  : §§SEA=§§SEC_§§CNT
  FOR.§§CNX|1|5
    DBP.§§SEA
    CTI.§§FIL|§§SEA|§§KEY_§§CNX|§§VAL|§§RES
    IVV.§§RES=1
      DBP. Ini was written with success!
    ELS.
      DBP. Ini was NOT written with success!
    EIF.
  NEX.
NEX.
' Now we read all Section-Names
CFI.§§FIL|§§MYS
PRT.Section-Names:$crlf§§§MYS

: §§SEA=Section_1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CFI.§§FIL|§§SEA|§§MYT
PRT.Key-Names:$crlf$§§MYT

MBX.Ready

ENR.
```

Remarks

In Mode 1+2, all returned Section and Key Names are separated by `$crlf$`.
Therefore you can easily enumerate them using `FEV`.

Limitations:

The maximum length of values to be read from an INI-File is is 65535 Bytes.

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- 3.3 Specialfolders 345
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.5 CHD. - Change Directory and Drive

MiniRobotLanguage (MRL)

CHD. Command

Change current Directory and Drive

Intention

Change the "current directory". This command is somehow like CHDIR in old DOS times, but it will also change the current Drive. Usage is simple:

```
' Please use folders that are valid on your system
CHD.F:\00_S-Files\00_Desktop
DBP.?curdir\
CHD.?ws\
DBP.?curdir\
ENR.
```

Will result in something like this.

```
[9] F:\00_S-Files\00_Desktop\
[11] C:\WINDOWS\system32\
```

As you can see, we can switch the drive and the current directory in one strike.
If you only want to switch the current drive you can do it like that:

```
CHD.|C
```

The second parameter can be as long as it is, only the first letter on the left side will be taken, it should be a valid drive letter. For example:

```
VAR.$$FIL=F:\00_S-Files\00_Desktop
CHD.|$$FIL
```

is completely valid.

Note: `CHD.` will execute the command internally and then place the current directory on TOS 104. If `CHD.` will run into an error, you will find the unchanged current directory on the TOS.

Syntax

CHD.[P1][|P2]

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 - (optional) File-path to be set as current directory.

P2 - (optional) valid drive-Letter to change the current drive

Example

```
'*****************************************
' Demo for CHD.
'*****************************************
CHD.F:\00_S-Files\00_Desktop
DBP.?curdir\
CHD.?ws\
DBP.?curdir\
ENR.
END.
```

Remarks

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- FEL - For-Each-Line 927
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.6 COP. - Copy File

MiniRobotLanguage (MRL)

COP. Command

Copy File

Intention

This command is used to copy files.
The command puts a zero (0) = "Operation failed or was aborted" or a one (1) = "operation successful" on the stack to flag if the operation was successful or not.

Usage is as easy as:

```
COP.source|dest
```

The Command has a optional network timeout.

```
COP.source|dest|timeout in sec.
```

In that case a timeout is applied. This can be helpful in Network environments like VPN's or WAN-Networks with high Latency times.

If only the folder is specified, the command will use the filename from the source. Like in this sample:

```
COP.C:\Myfolder\Myfile.txt|D:\Myfolder\
```

The command in its default-settings does not ask for any confirmation when overwriting files. It can do serious damages to your files and data. Therefore use it with care and on your own risk!

```
' Copy file with COP.
VAR.$$TIM=#dtime#
: $$SRC=F:\05-26-05.pst
: $$TAR=?desktop\Testfile.pst
COP.$$SRC|$$TAR|1
STS.DUMP
MBX.Ready: #dsince#
ENR.
```

On an test-system a file with 11.1 GB was copied in ~167 Seconds. The same operation used ~169 seconds using `SHF`. (Shell-File Operations). These times depend on the specific test system.

```
' Copy file with SHF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$TIM=#dtime#
: $$SRC=F:\05-26-05.pst
: $$TAR=?desktop\Testfile.pst
SHF.c|pcm|$$SRC|$$TAR
STS.DUMP
MBX.Ready: #dsince#
ENR.
```

Syntax

COP.P1|P2[|P3]

Parameter Explanation

P1 - Variable with source-file

P2 - Variable with destination-file name. If only the folder is specified, the command will use the filename from the source.

Example:
COP.C:\Myfolder\Myfile.txt|D:\Myfolder\

P3 - (optional) Timeout in seconds, default is 1 sec.

Example

```
'*****************************************************
' EXAMPLE: REN. and COP.
'*****************************************************
: §§PAT=F:\Testfiles
VAR.§§FIL=§§PAT\Audiograbber.zip
VAR.§§TAC=§§PAT\recent\Target.zip
: §§TIM=2
VAR.§§TAR=§§PAT\Target.zip
VAR.§§TAB=§§PAT\Targez.zip

COP.§§FIL|§§TAB
COP.§§FIL|§§TAB|(§§TIM+3)
' If Operation was successfull then Stack contains a 1 else a
0
PRT.$$000
IVV.$$000=1
PRT.Copy was successful
ELS.
PRT. There was an Error.
EVV.

' Rename-Example
REN.§§TAR|§§TAB

' Rename-Example with File-Timeout of 5 Seconds
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
REN.§§TAR|§§TAB|5
END.
```

Remarks

Wildcards are supported. The first file that fits the description is been copied.
The command will not copy multiple files that fit a description.

The Timeout will not be applied if directories shall be renamed.
For file-operations using wildcards and more options, try `SHF.`

Limitations:

-

See also:

- 3.3 Specialfolders 345
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.7 CSC. - Create ShortCut

MiniRobotLanguage (MRL)

CSC. Command

Create Shortcut

Intention

This command is used to create a shortcut in the startmenu or an the desktop.
There are two paths we have to deal with.

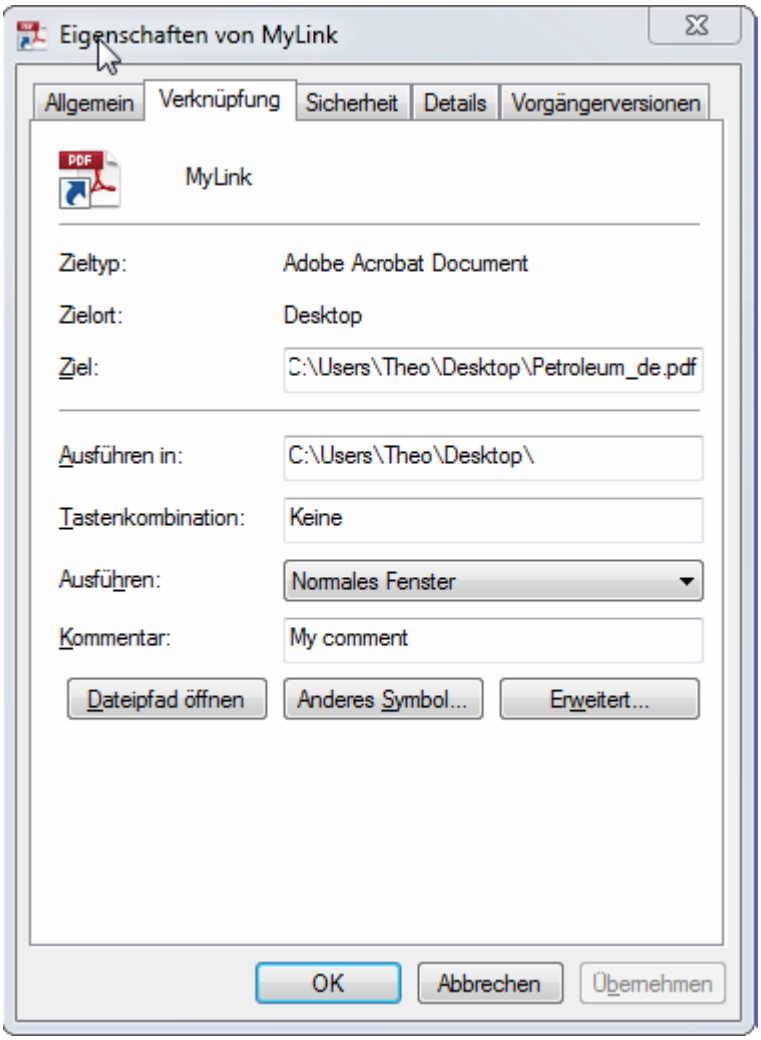
1. The Path where the shortcut is been created. We call it the "Link-Path".
2. The Path of the file that the shortcut links to. We call it the "Target-Path".

The "Link-Path" is given by a keyword. For a complete List of the available keywords see below.

```
' Create a shortcut with CSC.

' This is the Target-Path
$$FIL=?desktop\Petroleum_de.pdf

' The Link-Path is specified by a keyword "desktop", and
' the Link-Name is "MyLink"
CSC.cdesktop|MyLink|$$FIL|||6|My comment
ENR.
```

Here is the Result:

© 2023 Theo Gottwald * https://www.smart-package.com



These are the Parameters that can be specified.

Syntax

COP.P1|P2|P3[|P4][|P5][|P6][|P7]

Parameter Explanation

P1 - Specify "Link-Path". That is the Place where the Link is been created. One of the following keywords can be used:

```
"desktop","cdesktop"    ->  <desktop>
"udesktop","userdesk"   ->  <user name>\Desktop
"cdesktop","cserdesk"   -> All Users\Desktop
"usmprog","ustartmenu programs" -> Start Menu\Programs
"csmprog","cstartmenu programs" -> All Users\Programs
"ustartmenu","usm"      -> <user name>\Start Menu
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 "cstarmenu","csm"      -> All Users\Start Menu
 "mydocs"               -> My Documents
 "fav","favorites"       -> <user name>\Favorites
 "usmstartup","ustartup" -> Start Menu\Programs\Startup
(Autorun)
 "csmstartup","cstartup" -> All Users\Startup
 "usendto"               -> <user name>\SendTo
 "recent"                -> <user name>\Recent
 "bin","recycle bin"     -> <desktop>\Recycle Bin
 "drives"                -> My Computer
```

P2 - Link-Name, without the ".lnk"

P3 - Target Path. Path of the file that is linked.

P4 - (optional) - Commandline-Arguments for Target-File

P5 - (optional) Working-Directory.
If omitted, the Directory of the Target-Path is used.

P6 - (optional) numerical value that has the following meaning:

- 0 - HIDE Window
- 1 - SHOW NORMAL
- 2 - SHOWMINIMIZED
- 3 - SHOWMAXIMIZED

Please note that these values are interpreted (and often ignored) by the Target-program, not by windows.

P7 - (optional) Comment for the Comments field.

Example

```
'*****************************************************
' EXAMPLE: CSC. - Create Shortcut
'*****************************************************
$$FIL=?desktop\Petroleum_de.pdf
CSC.cdesktop|MyLink|$$FIL|||6|My comment
END.
```

Remarks

Wildcards are not supported.

The Timeout will be cleared. currently there is no internal check, if the operation was successful. You can use `IEF. (If exist file)` for this, if needed.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- **GSI. - Get Shortcut Information** 1046
- IEF. / NEF. - If - exist - File 1050
- 3.3 Specialfolders 345
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- GLC. - Get-Line-Count 1043
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.8 CTF. - Copy-To-File

MiniRobotLanguage (MRL)

CTF. Command

Copy to File

Intention

This command is used to write any content (Text or binary) to a file.
If the file does not exist it will be created.
If the file exists it will be overwritten (the old file is deleted and the new file will take its place).

Syntax

CTF.P1|P2[|P3]

Parameter Explanation

P1 – Filename
P2 – Text or Data
P3 – Opt. Variable which receives a 1 value if operation was successful and a 0 value if the operation failed.

Example

```
'*****************************************************
' EXAMPLE 1: CTF. Write anything to a file
'*****************************************************
'DBM.2
' This is the filename and path on the desktop
: $$FIL=?desktop\RND.txt

FOR.$$AAA|1|200
' Generate a Random Number between 124 and 123125
RND.124,123125,$$REB
' and append it to $$TXT
VAR.$$TXT=$$TXT$$REB
NEX.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Now we write it all to the file
CTF.$$FIL|$$TXT|$$ERG
PRT.Succesful if its a One: $$ERG

MBX.!
END.
```

Remarks

Existing files with the given file-name will be overwritten without any confirmation-request.

Limitations:

-

See also:

- 3.3 Specialfolders 345
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.9 CTI. - Copy-To-Inifile

MiniRobotLanguage (MRL)

CTI. Command

Copy to Inifile

Intention

This command is used to
- write entries into INI-Files,
- create INI files,
- delete keys, or
- delete sections from an INI-files.

When writing a value to an INI-File, the command
- checks if the length of the string to write is smaller then 65536 Bytes (MAXLEN for INI-Files is 65535 Bytes)
- if not the string is shortened to that length
- then writes that value into the INI-File
- then reads that value back and verifies that it has been written properly.

Only in that case you get an success (1) in Parameter P5. In case the written and read values differ, you get an failed (0) in P5.

Here are the 4 Modes of operation:

```
1.  CTI.Inifilename|Section -> Entire Section is been deleted
2.  CTI.Inifilename|Section|Key -> Key is deleted
3.  CTI.Inifilename|Section|Key|Value
    ' Copies the given Value into Key
4.  CTI.Inifilename|Section|Key|Value|Success-Flag
    ' Copies the given Value into Key, succes-flag to 0/1
```

Mode 2: You can delete the key:

```
CTI.§§FIL|§§SEC|§§KEY
```

There will not be any warning or such. The key will just be deleted, therefore use this with care!

Using Mode 1, you can delete the whole section. Its not much to type, but the result can be fatal if you don't know what you do. Use with care also.

```
CTI.§§FIL|§§SEC
```

The maximum length of values to be read from INI-Files is 65535 Bytes (tested under Win XP). Therefore the value to be written will internally be shortened to this length, in case its longer then this. Even then you will get a writing success in P5.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

CTI.P1|P2[|P3][|P4][|P5]

Parameter Explanation

P1 - Filename of Ini to write to
P2 - Section-Name
P3 - Key-Name
P4 - Value
P5 - (out) Variable for Success-flag 1/0.

Example

```
'******************************************************
' EXAMPLE 1: CTI. Write INI-File
'******************************************************
VAR.$$FIL=?desktop\one.ini
VAR.$$SEC=Section
VAR.$$KEY=Key
VAR.$$VAL=Value
CTI.$$FIL|$$SEC|$$KEY|$$VAL|$$RES

IVV.$$RES=1
     PRT. Ini was written with success!
ELS.
     PRT. Ini was NOT written with success!
EIF.

' Now we write without check
CTI.$$FIL|$$SEC|$$KEY|$$VAL

' Now we read the file
EXO.$$FIL

MBX. Now we delete this key!
CTI.$$FIL|$$SEC|$$KEY
MBX. Now we delete the whole section!
CTI.$$FIL|$$SEC

END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

The maximum length of values to be read from INI-Files is 65535 Bytes (XP). Therefore the value to be written will internally be shortened to this length, in case its longer then this. Even then you will get a writing success.

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- 3.3 Specialfolders 345
- CTF - Copy-To-File 964
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.10 CVF. - Convert File

MiniRobotLanguage (MRL)

CVF. Command

Coonvert File / Replace in File

Intention

This command is used to replace characters in a file and then write the file to an output file. Technically `CVF.` will load the whole file into memory and then do the replace.

You can convert files with different line-endings:

```
$$FIL=?path\MyFIle.xml
$$OUT=?path\MyFIle.txt
$$MAC=$lf$
$$PCE=$crlf$
CVF.$$FIL|$$MAC|$$PCE|$$OUT
```

or you can convert a XML-File into a textual readable file:

```
$$FIL=?path\MyFIle.xml
$$OUT=?path\MyFIle.txt
$$XML=><
$$PCE=>$crlf$<
CVF.$$FIL|$$XML|$$PCE|$$OUT
```

If you omit P4 then P1 is overwritten without further asking!

Syntax

CVF.P1|P2|P3[|P4]

Parameter Explanation

P1 – Filename
P2 – ***(Replace)*** Text or Data that is going to be replaced
P3 – ***(with)*** Text or Data that is going to replace the text **P2**
P4 – Opt. Variable which contains the Output filename, that the processed *file should be written to. If omitted **P1** is overwritten!*

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'******************************************************
' EXAMPLE 1: CVF.
'******************************************************
$$FIL=?path\MyFIle.xml
$$OUT=?path\MyFIle.txt
$$MAC=$lf$
$$PCE=$crlf$
CVF.$$FIL|$$MAC|$$PCE|$$OUT
MBX.!
END.
```

Remarks

Existing files with the given file-name **P4** will be overwritten without any confirmation-request.

Limitations:

-

See also:

- 3.3 Specialfolders 345
- **CTF. - Copy-To-File** 964
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.11 DCP. - Directory Copy

MiniRobotLanguage (MRL)

DCP. Command

Copy Directory

Intention

This command is used to copy complete folders with all files and sub-folders.
The command puts a zero (0) = "Operation failed or was aborted" or a one (1) = "operation successful" on the stack to flag if the operation was successful or not.
It will also set the TimeOut-Flag in case of an error.

Usage is as easy as:

```
DCP.source|dest
```

The Command needs a "\" at the end of each Path to make sure that its a Pathname, not a filename..

```
DCP.c:\folder\|d:\folderX\
```

will result in a new folder:

```
d:\folderX\folder\
```

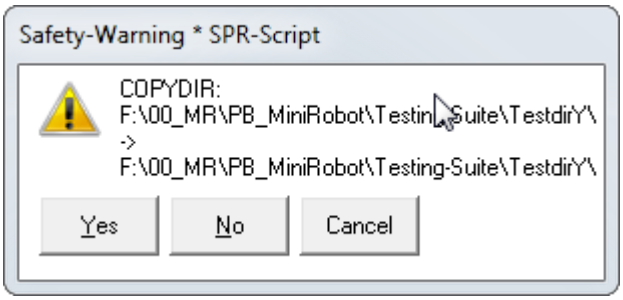
WARNING:
The command in its default-settings does not ask for any confirmation when overwriting files, even protected files.
Wrongly used on existing files, It can do serious damages to your files and data.
Therefore use it with care and on your own risk!

To debug your scripts safely, you can use:

```
OPT.SAFEFILE|1
```

this will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`

*© 2023 Theo Gottwald * https://www.smart-package.com*



You can choose between

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

Syntax

DCP.P1|P2[|P3]

Parameter Explanation

P1 – Variable with source-folder, ending with a "`\`"
Example: `F:\folder\`

P2 – Variable with destination-folder, ending with a "`\`"
Example: `X:\folder\`

P3 – *(optional)* can only be "`a`". In case **P3** is given, then
DirCopy will continue to copy even in case of copy-errors.
It will place the path of files that can not be copied on TOS.
In case "**a**" is given, the TIO-Flag may not be accurate (will not reflect an error).
Look on the TOS instead.

The result of the Operation (0/1) can be found on the TOS.
Also the Timeout-Flag is been set if the operation has failed.

Example

```
'******************************************************
' EXAMPLE: DCP.
 '******************************************************
 $$BAS=F:\Testing-Suite\
 $$TPA=C:\
 $$TPB=$$TPATestdirX\
 $$STA=190
 DCP.$$TPB|$$BAS
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
JIT.ero
§§STA=192
NEF.§§PAE
  JIT.ero
EIF.
ENR.
:ero
MBX.Error §§STA happened!
END.
```

Remarks

Wildcards are not supported.
The Timeout will not be applied if directories shall be renamed.
To show that something is a folder, suffixe it with a "\".

```
' This is a file
§§FIA=C:\myname

' This is a folder
§§FIA=C:\myname\
```

Limitations:

-

See also:

- 3.3 Specialfolders 345
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.12 DEL. - Delete File

MiniRobotLanguage (MRL)

DEL. Command

Delete File

Intention

This command is used to delete a file or more files. You can specify wildcards (e.g. ? and *). DEL. will not delete directories. Use `RMD.` for such tasks.

If you specify a "<>" in front of the filename, the file will be deleted to the User-trash of the current user.

```
' Delete to bin
DEL.<>?path\MyFile.txt
```

In case you use wildcards, the real, full filename and path of the file is been returned below TOS on position 2. On TOS is the success-flag.

`DEL.` has also a "**>>**" Option that will "**Safe-Erase**" Files. For this there are several Standards, one the most used Standards overwrites a file 35 times before deleting it.
We overwrite it 40 times, therefore we assume we are at the safe end with this option.

```
' This may take a while !
DEL.>>?path\MyFile.txt
```

Of course a safe-erase can never be combined with the recycle option.

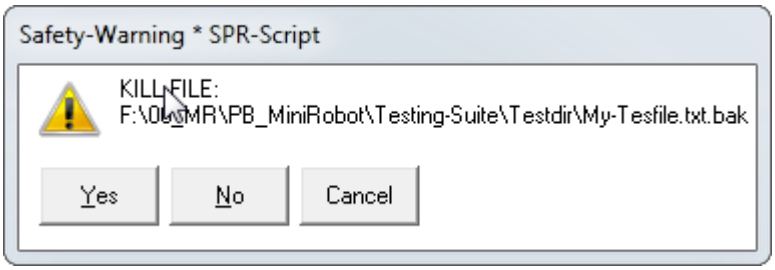
DANGER: If you do not use the "<>" then this command will permanently delete files from your hard disk. Use with care and on your own risk.

If you want to debug your Script safely, you can use the

```
OPT.SAFEFILE|1
```

this will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`

*© 2023 Theo Gottwald * https://www.smart-package.com*



You can choose between

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

Syntax

DEL.P1[|P2]

Parameter Explanation

P1 **-** Filename to delete.
    <> - If you prefix the filename wit "<>" then the file will
          be deleted to the recycle bin.
    >> - Safe-Erase File. The file will be overwritten
          about 40 times, with changing Bit-Patterns and Random
          Characters before the file will be finally deleted.
          Should be at least as Strong as needed.
          Standard for other such programs is 35 times.
          This algo is for files below 1.5 GB.

    << - Large File Safe-Erase. This is a slower implementation,
          that uses 64 Bit and can therefore handle files of any
          size.

P2 **- (optional)** Variable for Success-flag 1/0.

Example

```
'*****************************************************
' EXAMPLE 1: DEL - File
'*****************************************************
'DBM.2
VAR.$$FIL=?desktop\one.ini
DEL.$$FIL|$$RES
PRT.$$RES -> success?
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END.

'******************************************************
' EXAMPLE 1: DEL - File
'******************************************************
'
DEL.<>?path\Kopie*.in?
ENR.
```

Remarks

In case you use wildcards, the real full filename and path of the file is been returned below TOS on position 2. On TOS is the success-flag.

Limitations:

-

See also:

- **RMD. - Remove-Directory** 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.13 FIF. - Find in File

MiniRobotLanguage (MRL)

FIF. Command

Find in File

Intention

This command can be used to find Text or Patterns in files.
You can search in binary files or in text-files.
Search in Text-Files is line-based.
In all files, Wildcard-Search tries to deliver the optimum result.
Wildcards use:

```
?               Any single character
*               Zero or more characters
#               Any single digit (0-9)
[charlist]      Any single character in 'charlist'
[!charlist]     Any single character not in 'charlist'
```

Example:

```
$$TXT=Peter*Müller
FIF.tw|$$FIL|$$TXT|3
```

This will search in a Text File the 3rd Person with the Name
`"Peter (anything) Müller".`
It will find
`"Peter Ralf Müller"`
as well as
`"Peter Klaus Müller"`
The command will deliver that
- found name and the
- start and Ending position in the file where it was found.

It will try to find the smallest fitting pattern.

You can specify the term "`word`" to have an additional tracer, complete found words until their word-end.

You can specify "`nocase`" in all text searches to search case independent.

Usage is simple:

```
' Specify a filename
$$FIL=?path\Test.txt

' Set a Pattern to search for
$$SRC=11.?7.*
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' start the desired search
FIF.tsn|$$FIL|$$SRC

' See the results on TOS
STS.DUMP
' This is just to show we are ready
MBX.!
ENR.
```

Syntax

FIF.P1|P2|P3[|P4]

Parameter Explanation

P1 - Prototype can be one of these:

```
bw,binary wildcard
bs,binary string
tw,text wildcard
ts,text string
```

by adding "word" you have 2 more options

```
tww,text wildcard word
tsw,text string word
```

by adding "nocase" you have 4 more options

```
twwn,text wildcard nocase
tswn,text string nocase
twwn,text wildcard word nocase
tswn,text string word nocase
```

Generally all "binary" searches are "byte based" and not "Line based".
They will search any file "Byte by Byte".

This is slower then searching in text files. Therefore you can search in files that are not "Text-Files".
The result will show the start and end-byte-position in the binary file.
There is no "word"-option for binary files.

All searches with "text" are line-based. For this, the file must have an line-delimiter, best is $crlf$. But cr or lf should also work as line-delimiter.

In text based searches, the result on TOS will show the 1-based Linenumber, and the start- and end-byte-positions in the line.

Also the last stack position will contain the found line-delimiter as a string.
This is done so you can easily reconstruct the file with the same type of delimiter, in case you want to make changed copy.

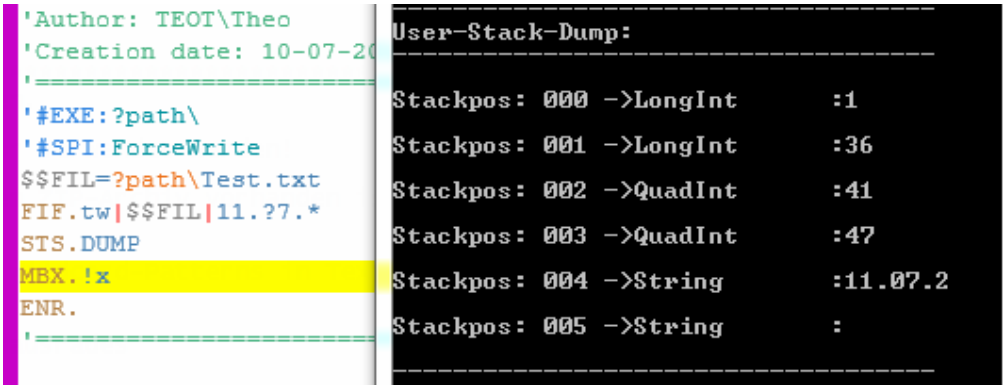
Adding the term "word" to the prototype will call an additional task, that will trace the found end-position to a word-boundary.
Currently word boundaries are:
ASCII(10,13,160,32) and TAB and ".,;"

© 2023 Theo Gottwald * https://www.smart-package.com

The "string" search modes do not compare pattern. They will just do an "Instring"-compare.

All compares are case-sensitive unless the two "nocase" searches.

Here is an example how the TOS looks after a text based Wildcard-search.

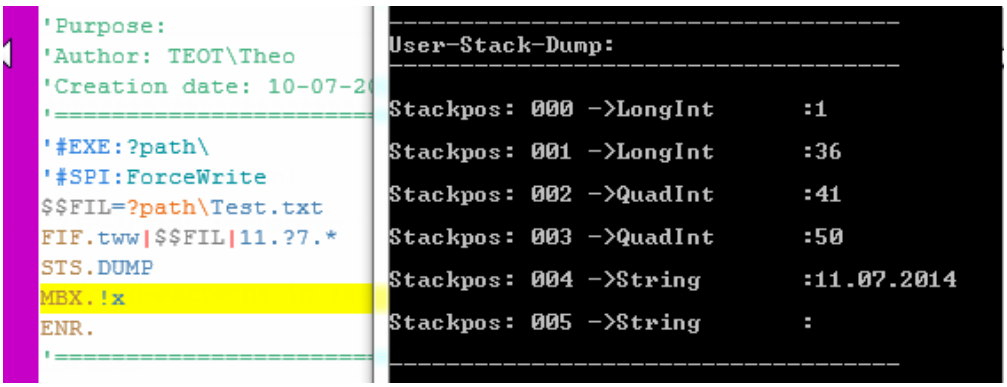


TOS

000 - Number 1 - signals that something was found.
In case the search fails, the TOS just contains a "0".

001 - Linenumber where the search found the text
002 - Starting Byte-position of the found text (in the line)
003 - Ending Byte-position of the found text (in the line)
004 - Found text as string
005 - Line delimiter that was found in the textfile as string

Using the "word" addition we can easily get the complete date!



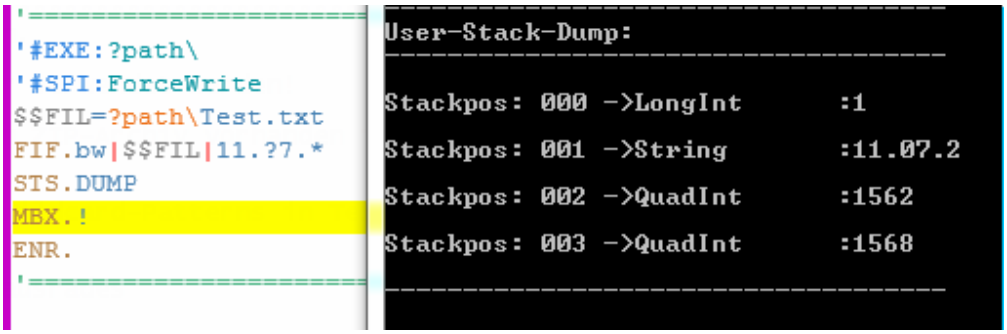
TOS - same as before. Just the string is been traced to the word-end.

000 - Number 1 - signals that something was found.
In case the search fails, the TOS just contains a "0".

001 - Linenumber where the search found the text
002 - Starting Byte-position of the found text (in the line)
003 - Ending Byte-position of the found text (in the line)
004 - Found text as string
005 - Line delimiter that was found in the textfile as string

© 2023 Theo Gottwald * https://www.smart-package.com

In binary searches the TOS looks a bit different:




```
TOS

000 - Number 1 - signals that something was found.
      In case the search fails, the TOS just contains a "0".

001 - Found text as string
002 - Starting Byte-position of the found text (in the line)
003 - Ending Byte-position of the found text (in the line)
```

P2 - Filename of the File to search.

P3 - Text or Wildcard-Pattern to search for

P4 - (*optional*) number of found result (1st .. 2nd .. 3rd etc.)

```
Wildcard-Pattern (in Wildcard-Mode only):
   ?              Any single character
   *              Zero or more characters
   #              Any single digit (0-9)
   [charlist]     Any single character in 'charlist'
   [!charlist]    Any single character not in 'charlist'
```

Example

```
'*****************************************
' FIF. - Sample
'*****************************************
$$FIL=?path\Test.txt
$$SRC=11.?7.*
FIF.tww|$$FIL|$$SRC
JIZ.$$000|Lab_nf
$$LIN=$$000
$$STA=$$000
$$END=$$000
$$STR=$$000
PRT.String: $$STR Found in Line: $$LIN
PRT.From Character $$STA to $$END
:Lab_nf
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.!
ENR.
END.

'**********************************
' FIF.-Sample
' Replace in File
' Here we use RPL to replace Text in
' a line-based Textfile
' We search for a Date like 11.?7.*
' using Wildcards ...
' and replace it with 12.08.2015
'**********************************
$$FIL=?path\Test.txt
$$TAR=?path\Target.txt
' Delete it if it existed before
DEL.$$TAR
' We search for the line with the date
$$SRC=11.?7.*
' New date - we replace the old one with this
$$NEW=12.08.2015
FIF.tww|$$FIL|$$SRC
JIZ.$$000|Lab_nf
$$LIN=$$000
$$STA=$$000
$$END=$$000
$$STR=$$000
$$TRE=$$000
PRT.String: $$STR Found in Line: $$LIN
PRT.From Character $$STA to $$END
GLC.$$FIL|$$REA
' Line where String is minus 1
CAL.$$TOL=$$LIN-1
' Copy all lines until the wanted Line
FOR.$$COP|1|$$TOL
  LFF.$$FIL|$$COP|$$RES
   ' We use the delimiter that was found in the source file
  $$TXT=$$RES$$TRE
  ATF.$$TAR|$$TXT
NEX.
   ' Here is the Line with the date
LFF.$$FIL|$$LIN|$$TXA
RPL.$$TXA|$$STR|$$NEW
$$TXB=$$TXA$$TRE
ATF.$$TAR|$$TXB
' Now copy Rest of file
' Line where String is plus 1
CAL.$$TOL=$$LIN+1
FOR.$$COP|$$TOL|$$REA
  LFF.$$FIL|$$COP|$$RES
   ' We use the delimiter that was found in the source file
  $$TXT=$$RES$$TRE
  ATF.$$TAR|$$TXT
NEX.

:Lab_nf
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Remarks

Internally a brute force algo is used for the wildcard-search, therefore Wildcard-Search in large binary files can be quite slow!
This has to do with the Maximum Pattern size that is checked against the Pattern if you use a * - Wildcard.

Because a * Wildcard can have theoretically any length - up to the full length of the file, there is an internal limit for the maximum found String.
Currently that limit is 256 per default.
You can change that maximum Pattern size limit using the OPT.FIF|(number) - command. The larger the maximum Pattern size, the longer the search will possibly take in wildcard search mode. The larger results you can find using one or more * - Wildcards.

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- FEL - For-Each-Line 927
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14 FIL. - File-Commands

3.54.14.1 FileDiff

MiniRobotLanguage (MRL)

FIL.FileDiff

Compare two files including their content and return the Position of the first difference

Intention

This command is used to compare two files "Byte by Byte" for equality.
If both files are equal, the result is 1, otherwise the result is zero. On the TOS is the Position of the fist different Byte.

The command checks these files see error-codes below.

Usage of the command is simple:

```
  ' $$FIA - File 1
  ' $$FIA - File 2
  ' $$SPO - Starting Position in Byte starting at 1 .. File-Len
  ' $$RES - Variable mit Ergebnis
FIL.FileDiff|$$FIA|$$FIB|$$SPO|$$RES
```

if P3 is missing, default Starting Position is always 1
If P4 is missing, the result will be placed on the TOS.

Another Sample:

```
  $$FOL=H:\Videotemp
  $$ORI=$$FOL\Text a.txt
  $$FIA=$$FOL\Text b.txt
  FIL.FileDiff|$$ORI|$$FIA|-1|$$RES
  VAN.$$DIF=$$000
  FIL.Length|$$FIA|$$LEN
  CAL.$$PLA=$$LEN-$$DIF+1
  IVV.$$RES=0
    DBP. Files are different at Reverse Position:$$DIF
    DBP. Files are different at Position $$PLA
  ELS.
    DBP.Files are equal
  EIF.
  FIL.FileDiff|$$ORI|$$FIA||$$RES
  IVV.$$RES=0
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  DBP. Files are different at: $$000
ELS.
  DBP.Files are equal
EIF.
MBX.!
ENR.


File A is:  AAAAAB
File B is:  AAAACB

Output:
[16:35:51]  Files are different at Reverse Position:2
[16:35:51]  Files are different at Position 5
[16:35:51]  Files are different at: 5
```

Possible Results:

```
1 - Files are Equal
```

If Files are not Equal, the Result is negative

```
-1 - can't compare Files or unequal size
-2 - File does not exist?, INVALID_FILE_ATTRIBUTES of Fi
-3 - File does not exist?, INVALID_FILE_ATTRIBUTES of Fi
-4 - File A File is a Folder
-5 - File B is a Folder
-6 - Can't Open File A
-7 - Can't Open File B
-8 - reserved
-9 - Starting Position > Filesize
-10 - Error during Read
```

Syntax

FIL.FileDiff|P1|P2[|P3][|P4]

Parameter Explanation

P1 - Filename for file 1

P2 - Filename for file 2

© 2023 Theo Gottwald * https://www.smart-package.com

P3 - (optional) Starting Byte Position for Compare (1 ... File-Length). if P3 is
If P3 = -1 then the Search will start at the End of the File. The result wi
See Sample below.

P4 - (optional) Variable for the result. The result will be 0 or 1 or one of the
In case the files compute as identical, the result is 1.
If the files are computed as "non identical", the result is zero. If P4 is

The Number of the first differing Byte is placed on TOS.

Example

```
'******************************************************
' EXAMPLE 1: FIL.FileDiff
'******************************************************
$$FOL=H:\Videotemp
$$ORI=$$FOL\TestA.mp4
$$FIA=$$FOL\TestA_injected A.mp4
FIL.FileDiff|$$ORI|$$FIA||$$RES
IVV.$$RES=0
  DBP. Files are different at: $$000
ELS.
  DBP.Files are equal
EIF.
ENR.
'------------------------------------------------------------

'******************************************************
' EXAMPLE 1: FIL.FileDiff
'******************************************************
$$FOL=H:\Videotemp
$$ORI=$$FOL\Text a.txt
$$FIA=$$FOL\Text b.txt
FIL.FileDiff|$$ORI|$$FIA|-1|$$RES
VAN.$$DIF=$$000
FIL.Length|$$FIA|$$LEN
CAL.$$PLA=$$LEN-$$DIF+1
IVV.$$RES=0
  DBP. Files are different at Reverse Position:$$DIF
  DBP. Files are different at Position $$PLA
ELS.
  DBP.Files are equal
EIF.
FIL.FileDiff|$$ORI|$$FIA||$$RES
IVV.$$RES=0
  DBP. Files are different at: $$000
ELS.
  DBP.Files are equal
EIF.
MBX.!
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

If both of the given Parameters evaluate to a folder that exists, the result is 1.
In this case there is no binary compare of the content.
This command does not compare folders. It should only be used on files.

See also:

- **Split File** 1001
- **Join File** 994
- **OPT. - Optional Settings** 2406
- **GEC. - Get Encryption** 825
- **RMD. - Remove-Directory** 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.2 Compare

MiniRobotLanguage (MRL)

FIL.Compare

Compare two files including their content

Intention

This command is used to compare two files "Byte by Byte" for equality.
If both files are equal, the result is 1, otherwise the result is zero.

The command checks these files for existence, if any of the files does not exist, the result is 0.
If one of the given Parameters evaluates to a folder the result is zero.

Usage of the command is simple:

```
  ' $$FIA - File 1
  ' $$FIA - File 2
  ' $$RES - Variable mit Ergebnis
FIL.Compare|$$FIA|$$FIB|$$RES
```

If P3 is missing, the result will be placed on the TOS.

```
  ' $$FIA - File 1
  ' $$FIA - File 2
  '
FIL.Compare|$$FIA|$$FIB
MBX.$tos$
```

Syntax

FIL.Compare|P1|P2[|P3]

Parameter Explanation

P1 - Filename for file 1

P2 - Filename for file 2

© 2023 Theo Gottwald * https://www.smart-package.com

P3 **- (optional)** Variable for the result. The result will be 0 or 1.
In case the files compute as identical, the result is 1.
If the files are computed as "non identical", do not exist, or if there are errors,
the result is zero.

Example

```
'*****************************************************
' EXAMPLE 1: FIL.Compare
' Save as "Tests.mrt"
'*****************************************************
$$FIA=?path\Tests.mrt
$$FIB=?path\Identic.txt
$$FIC=?path\False.txt
CFF.$$FIA|$$TXT
' Make an equal File thats identical
CTF.$$FIB|$$TXT
' Make a file that differs at Byte Position 10
STR.PATCH|$$TXT|10|1|Ä
CTF.$$FIC|$$TXT
FIL.Compare|$$FIA|$$FIB|$$RES
GSB.resul
FIL.Compare|$$FIA|$$FIC|$$RES
GSB.resul
ENR.
'-------------------------------------------------------------
:resul
IVV.$$RES=1
  MBX. The files are identical.
ELS.
  MBX. The files are NOT identical!
EIF.
RET.
'-------------------------------------------------------------
```

Remarks

-

Limitations:

If both of the given Parameters evaluate to a folder that exists, the result is 1.
In this case there is no binary compare of the content.
This command does not compare folders. It should only be used on files.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- **Split File** 1001
- **Join File** 994
- **OPT. - Optional Settings** 2406
- **GEC. - Get Encryption** 825
- **RMD. - Remove-Directory** 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.54.14.3 Join Delete

MiniRobotLanguage (MRL)

FIL.Join Delete

Join previously splitted files into a larger file and delete Splits

Intention

This command is used to join the files that have been created with Split File 1001 into the original large file. To join other files, prefer the ATF. - Append-to-File 946 command instead.
This command together with the Split Fil 1001 command help to automatically split larger files into smaller pieces. The command
`FIL.Join File`
does generally the same like
`FIL.Join delete`
command to restore the original file. The only difference between these two is, that the first command does not automatically delete the split-files after restoring the original file.

Usage of the command is simple:

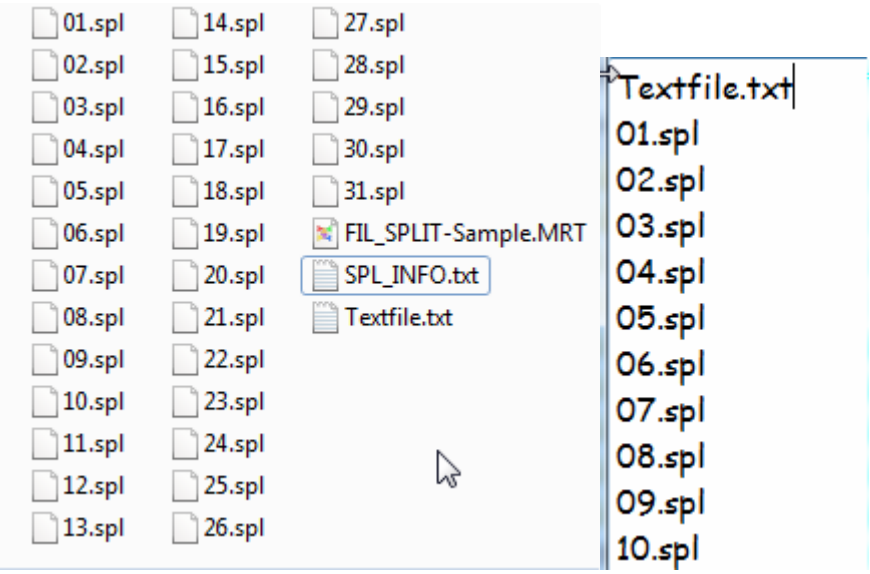
```
' To split the file
VAF.$$FIA=?path\File.exe
FIL.SPLIT FILE|$$FIA|16000

' To Join the file
```

would split the File into pieces of each 16000 Bytes, Of course the last piece may have a different size, as it contains just the remaining Bytes.

Also note that the `FIL.Split File` command has two dependencies on global settings.

Here is an example, how the File-Splitting looks in an explorer folder:

*© 2023 Theo Gottwald * https://www.smart-package.com*



On the right side is the content of the "SPL_INFO.txt"-file.
All lines are separated with $crlf$.

All the split-files are numbered and have currently the extension "spl".

The file "SPL_INFO.txt" contains all informations that are needed for the the `FIL-Join File` command to put the files together.
You can see an example, in the picture on the right side.
First it contains the original Filename of the file before it was splitted.
Secondly it contains all the filenames of the splitter-files in the right order.
Any changes you make in the "SPL_INFO.txt" will lead to changes in the final outcome.
There are no additional information on the splitting process in the splitter-files.
Therefore an "SPL_INFO.txt" is needed to concatenate these files to the original file.

Syntax

FIL.Join Delete|P1[|P2]

Parameter Explanation

P1 - `Filename and Path to "SPL_INFO.txt" file. Only a path is Ok.`

P2 - **`(optional)`** `Variable for 16 Byte-Key for TEA Encryption of splitted files. If given this password must of course be the same like it was used for decryption.*`

There are 2 dependencies on global settings:
*If "compression and encryption" is used, the size of the resulting files may

© 2023 Theo Gottwald * https://www.smart-package.com

be smaller then the given number in **P2**. To decide whether "compression and encryption" should be used, Set
`OPT.SPLITCOMP|1`
to 0 or 1. The default number of rounds (encryption strength) is 30.
You can change that using
`ENCRYPT|100`

`OPT.`this will set the default encryption to 100 rounds.

Example

```
'*****************************************************
' EXAMPLE 1: FIL.JoinFile
'*****************************************************
' no compression/encryption (default)
OPT.SPLITCOMP|0
VAF.$$FIA=?path\Textfile.txt
VAF.$$FIB=?path\TextfileX.txt
FIL.SPLIT FILE|$$FIA|160
REN.$$FIA|$$FIB|1000
' automatically delete splitted files after
' restoring the original file
FIL.join delete
ENR.

'*****************************************************
' EXAMPLE 2: FIL.JoinFile / FIL.Join Delete
'*****************************************************
' enable compression/Encryption
OPT.SPLITCOMP|1
' Set encryption to 100 TEA Rounds
OPT.ENCRYPT|100
' Time-Measurement
VAR.$$TIA=#dtime#
VAF.$$FIA=?path\Textfile.txt
VAF.$$FIB=?path\TextfileX.txt
FIL.SPLIT FILE|$$FIA|160
ITO.
  goto ero
EIF.
VAR.$$TIB=#dsince#
PRT.Split took:$$TIB
MBX.!
REN.$$FIA|$$FIB|1000
OPT.SPLITCOMP|1
VAR.$$TIA=#dtime#
FIL.join delete
ITO.
  GTO. ero
EIF.
VAR.$$TIB=#dsince#
PRT.Join took:$$TIB
:enx
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:ero
MBX.Got TIO
GTO. enx
```

Remarks

Time considerations:
In my testing environment, I tested to split a file of **6.85 GB** into **49 Pieces**.
The needed time with compression and encryption was: **10.5 Minutes**
The needed time without compression and encryption was: **6.5 Minutes**
The time needed **to join the file** was **8.3** Minutes with compression and encryption and **6** Minutes without compression and encryption.

Limitations:

The Splitter works with 64 bit internally. Therefore files of up to 18446744073709551615 Bytes can be splitted to smaller split-files. But there is a limit on the size of these splitted files.
Due to speed-considerations, the split-files can not exceed ~1-2 GB (depending on the pool of free 32-bit Memory in WOW64).
There is no limit on the number of Split-Files that you can generate (other then 32-bit).

See also:

- **Split File** 1001
- **Join File** 994
- **OPT. - Optional Settings** 2406
- **GEC. - Get Encryption** 825
- **RMD. - Remove-Directory** 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.4 Join File

MiniRobotLanguage (MRL)

FIL.Join File

Join previously splitted files into a larger file

Intention

This command is used to join the files that have been created with Split File 1001 into the original large file. To join other files, prefer the ATF. - Append-to-File 946 command instead.
This command together with the Split Fil 1001 command help to automatically split larger files into smaller pieces. The command
`FIL.Join File`
does generally the same like
`FIL.Join delete`
command to restore the original file. The only difference between these two is, that the first command does not automatically delete the split-files after restoring the original file.

Usage of the command is simple:

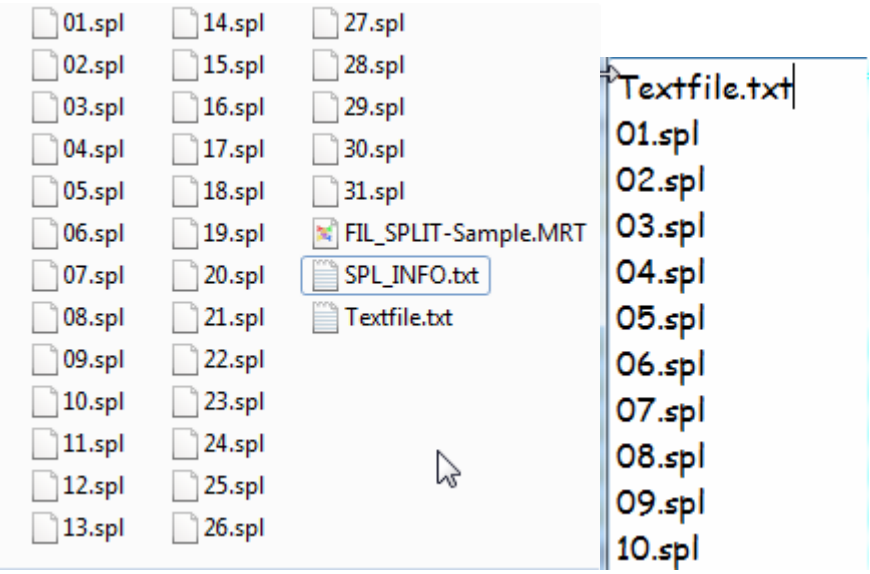
```
' To split the file
VAF.$$FIA=?path\File.exe
FIL.SPLIT FILE|$$FIA|16000

' To Join the file
```

would split the File into pieces of each 16000 Bytes, Of course the last piece may have a different size, as it contains just the remaining Bytes.

Also note that the `FIL.Split File` command has two dependencies on global settings.

Here is an example, how the File-Splitting looks in an explorer folder:

*© 2023 Theo Gottwald * https://www.smart-package.com*



On the right side is the content of the "SPL_INFO.txt"-file.
All lines are separated with $crlf$.

All the split-files are numbered and have currently the extension "spl".

The file "SPL_INFO.txt" contains all informations that are needed for the the `FIL-Join File` command to put the files together.
You can see an example, in the picture on the right side.
First it contains the original Filename of the file before it was splitted.
Secondly it contains all the filenames of the splitter-files in the right order.
Any changes you make in the "SPL_INFO.txt" will lead to changes in the final outcome.
There are no additional information on the splitting process in the splitter-files.
Therefore an "SPL_INFO.txt" is needed to concatenate these files to the original file.

Syntax

FIL.Join File|P1[|P2]

Parameter Explanation

P1 - Filename and Path to "SPL_INFO.txt" file. Only a path is Ok.

P2 - **(optional)** Variable for 16 Byte-Key for TEA Encryption of splitted files. If given this password must of course be the same like it was used for decryption.*

There are 2 dependencies on global settings:
*If "compression and encryption" is used, the size of the resulting files may

© 2023 Theo Gottwald * https://www.smart-package.com

be smaller then the given number in **P2**. To decide whether "compression and encryption" should be used, Set
`OPT.SPLITCOMP|1`
to 0 or 1. The default number of rounds (encryption strength) is 30.
You can change that using
`OPT.ENCRYPT|100`
this will set the default encryption to 100 rounds.

Example

```
'*****************************************************
' EXAMPLE 1: FIL.JoinFile
'*****************************************************
' no compression/encryption (default)
OPT.SPLITCOMP|0
VAF.$$FIA=?path\Textfile.txt
VAF.$$FIB=?path\TextfileX.txt
FIL.SPLIT FILE|$$FIA|160
REN.$$FIA|$$FIB|1000
' automatically delete splitted files after
' restoring the original file
FIL.join delete
ENR.

'*****************************************************
' EXAMPLE 2: FIL.JoinFile / FIL.Join Delete
'*****************************************************
' enable compression/Encryption
OPT.SPLITCOMP|1
' Set encryption to 100 TEA Rounds
OPT.ENCRYPT|100
' Time-Measurement
VAR.$$TIA=#dtime#
VAF.$$FIA=?path\Textfile.txt
VAF.$$FIB=?path\TextfileX.txt
FIL.SPLIT FILE|$$FIA|160
ITO.
  goto ero
EIF.
VAR.$$TIB=#dsince#
PRT.Split took:$$TIB
MBX.!
REN.$$FIA|$$FIB|1000
OPT.SPLITCOMP|1
VAR.$$TIA=#dtime#
FIL.join delete
ITO.
  GTO. ero
EIF.
VAR.$$TIB=#dsince#
PRT.Join took:$$TIB
:enx
MBX.!
ENR.
:ero
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.Got TIO
GTO. enx
```

Remarks

Time considerations:
In my testing environment, I tested to split a file of **6.85 GB** into **49 Pieces**.
The needed time with compression and encryption was: **10.5 Minutes**
The needed time without compression and encryption was: **6.5 Minutes**
The time needed **to join the file** was **8.3** Minutes with compression and encryption and **6** Minutes without compression and encryption.

Limitations:

The Splitter works with 64 bit internally. Therefore files of up to 18446744073709551615 Bytes can be splitted to smaller split-files. But there is a limit on the size of these splitted files.
Due to speed-considerations, the split-files can not exceed ~1-2 GB (depending on the pool of free 32-bit Memory in WOW64).
There is no limit on the number of Split-Files that you can generate (other then 32-bit).

See also:

- Split File 1001
- Join Delete 990
- OPT. - Optional Settings 2406
- GEC. - Get Encryption 825
- RMD. - Remove-Directory 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.5 Length

MiniRobotLanguage (MRL)

FIL.Length

Returns the Length of a given File

Intention

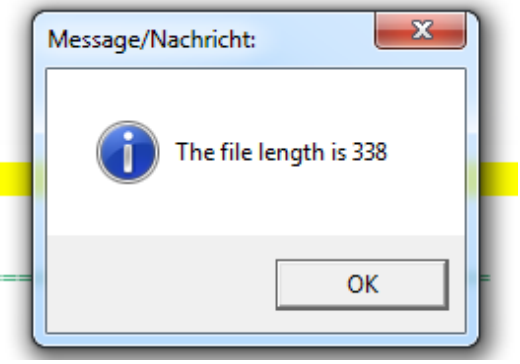
This command is used to return the File-Length . The command

```
$$FIL=C:\Tesffile.txt
FIL.Length|$$FIL|$$RES
MBX.The file length is $$RES
```

If you omit P3 then the result will be placed on the TOS.

```
$$FIL=C:\Tesffile.txt
FIL.Length|$$FIL
MBX.The file length is $tos$
```

```
'Creation date: 02-09-2015 at 19:36:11
'================================================================
'#EXE:?path\
'#SPI:ForceWrite

$$FIL=?path\Tests.mrt
FIL.Length|$$FIL
MBX.The file length is $tos$

ENR.
'================================
.
```



Syntax

FIL.Length|P1[|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** Filename and Path to the file

P2 **-** **(optional)** Variable for the result. If omitted the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: FIL.Length
'*****************************************************
$$FIL=?path\Tests.mrt
FIL.Length|$$FIL|$$RES
MBX.The file length is $$RES
ENR.

'*****************************************************
' EXAMPLE 2: FIL.Length
'*****************************************************
$$FIL=?path\Tests.mrt
FIL.Length|$$FIL
MBX.The file length is $tos$
ENR.
```

Remarks

Uses 64 bit numbers, can be used for files of any size.

Limitations:

-

See also:

- Split File 1001
- Join File 994
- OPT. - Optional Settings 2406
- GEC. - Get Encryption 825
- RMD. - Remove-Directory 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.6 Split File

MiniRobotLanguage (MRL)

FIL.Split File

Split large File into smaller files

Intention

This command is used to split large files up to any size (64-bit) can be splitted into smaller files - for example to send them away via Mail.

The receiver can use the
`FIL.Join File`
or the
`FIL.Join delete`
command to restore the original file. The only difference between these two is, that the first command does not automatically delete the split-files after restoring the original file.

Usage of the command is simple:

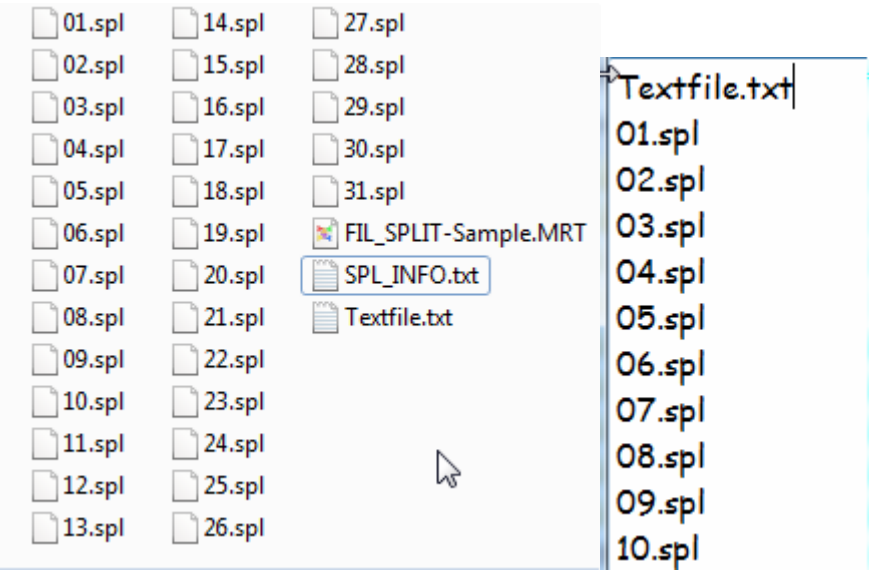
```
VAF.$$FIA=?path\File.exe
FIL.SPLIT FILE|$$FIA|16000
```

would split the File into pieces of each 16000 Bytes, Of course the last piece may have a different size, as it contains just the remaining Bytes.

Also note that the `FIL.Split File` command has two dependencies on global settings.

Here is an example, how the File-Splitting looks in an explorer folder:

© 2023 Theo Gottwald * https://www.smart-package.com



On the right side is the content of the "SPL_INFO.txt"-file.
All lines are separated with $crlf$.

All the split-files are numbered and have currently the extension "spl".

The file "SPL_INFO.txt" contains all informations that are needed for the the `FIL-Join File` command to put the files together.
You can see an example, in the picture on the right side.
First it contains the original Filename of the file before it was splitted.
Secondly it contains all the filenames of the splitter-files in the right order.
Any changes you make in the "SPL_INFO.txt" will lead to changes in the final outcome.
There are no additional information on the splitting process in the splitter-files.
Therefore an "SPL_INFO.txt" is needed to concatenate these files to the original file.

Syntax

FIL.Split File|P1|P2[|P3][|P4]

Parameter Explanation

P1 - Filename and Path.
P2 - ***numerical, size of the Split-Files in*** Bytes.
If you plan to send the files via E-Mail to somebody, you could for example choose 12000000 (12 MB).*

P3 - **(optional)** Variable for Target Folder. If omitted, ?path\ is taken.
P4 - **(optional)** Variable for 16 Byte-Key for TEA Encryption

© 2023 Theo Gottwald * https://www.smart-package.com

```
     of splitted files. If omitted a hardcoded password is
used.
     Of course anybody who owns a SPR can then decrypt the
files.*
```

There are 2 dependencies on global settings:
*If "compression and encryption" is used, the size of the resulting files may be smaller then the given number in **P2**. To decide whether "compression and encryption" should be used, Set

```
 OPT.SPLITCOMP|1
```

to 0 or 1. The default number of rounds (encryption strength) is 30.
You can change that using

```
OPT.ENCRYPT|100
```

this will set the default encryption to 100 rounds.

Example

```
'*****************************************************
' EXAMPLE 1: FIL.Split File
'*****************************************************
' no compression/encryption (default)
OPT.SPLITCOMP|0
VAF.$$FIA=?path\Textfile.txt
VAF.$$FIB=?path\TextfileX.txt
FIL.SPLIT FILE|$$FIA|160
REN.$$FIA|$$FIB|1000
' automatically delete splitted files after
' restoring the original file
FIL.join delete
ENR.

'*****************************************************
' EXAMPLE 2: FIL.Split File / FIL.Join Delete
'*****************************************************
' enable compression/Encryption
OPT.SPLITCOMP|1
' Set encryption to 100 TEA Rounds
OPT.ENCRYPT|100
' Time-Measurement
VAR.$$TIA=#dtime#
VAF.$$FIA=?path\Textfile.txt
VAF.$$FIB=?path\TextfileX.txt
FIL.SPLIT FILE|$$FIA|160
ITO.
  goto ero
EIF.
VAR.$$TIB=#dsince#
PRT.Split took:$$TIB
MBX.!
REN.$$FIA|$$FIB|1000
OPT.SPLITCOMP|1
VAR.$$TIA=#dtime#
FIL.join delete
ITO.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GTO. ero
EIF.
VAR.$$TIB=#dsince#
PRT.Join took:$$TIB
:enx
MBX.!
ENR.
:ero
MBX.Got TIO
GTO. enx
```

Remarks

Time considerations:
In my testing environment, I tested to split a file of **6.85 GB** into **49 Pieces**.
The needed time with compression and encryption was: **10.5 Minutes**
The needed time without compression and encryption was: **6.5 Minutes**
The time needed **to join the file** was **8.3** Minutes with compression and encryption and **6** Minutes without compression and encryption.

Limitations:

The Splitter works with 64 bit internally. Therefore files of up to 18446744073709551615 Bytes can be splitted to smaller split-files. But there is a limit on the size of these splitted files.
Due to speed-considerations, the split-files can not exceed ~1-2 GB (depending on the pool of free 32-bit Memory in WOW64).
There is no limit on the number of Split-Files that you can generate (other then 32-bit).

See also:

- Join File 994
- OPT. - Optional Settings 2406
- RMD. - Remove-Directory 1077
- GEC. - Get Encryption 825
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056

© 2023 Theo Gottwald * https://www.smart-package.com

- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.7 Generate numbered Files

MiniRobotLanguage (MRL)

FIL.gnf
FIL.Generate numbered Files

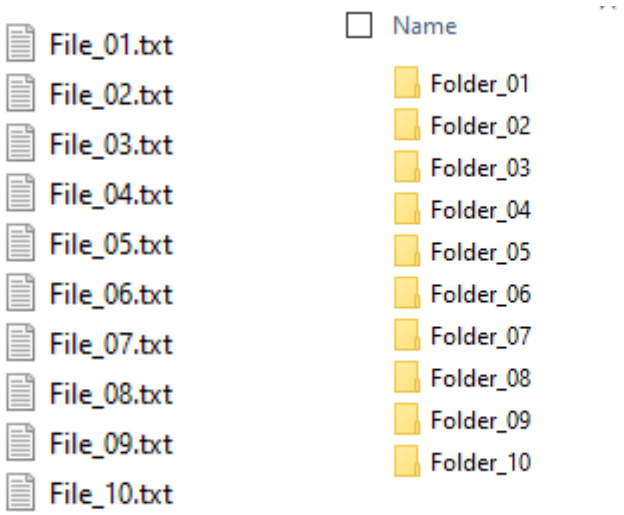
Generates a series of numbered files or folders

Intention

This command is used generate a series of Files or Folders with a builtin Number.
Generally the "?" in the Filename is replaced with that number.
If there is a supplied Text, with a "%n" in it, this will be replaced with the formatted number.

```
$$MSG=Teste FIL "GNF mit File".
$$LAB=$$LAAFile_?.txt
$$LAC=This Text has number %n.
FIL.gnf|$$LAB|$$LAC|1|10|2|1
VAR.$$LAE=$tos$
```

This is how such "Numbered files" or folders do look.



Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

FIL.gnf|P1|P2|P3[|P4][|P5]
FIL.generate numbered files|P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 **-** Filename and Path to the file, it must contain a "?".
The "?" will be replaced with the formatted Number during the process.
If **P1** ends with a \ then its taken as a folder-name.

P2 **- (numeric)** Variable or value with the first number (starting number).

P3 **- (numeric)** Variable or value with the last number (ending number).
The process will generate the numbered files between the number in **P1** and **P2**.

P4 **- (numeric)** Variable or value with number of digits to format the number.
-1 - do not format the numbers, e.g. File_2.txt
0 - format with number of digits in **P3**, e.g. if **P3**=10, File_02.txt
1 - format with number of (digits in **P3**)+1, e.g. if **P3**=10, File_002.txt
For example: If the number is 3, then the values will be formatted like this: "005".
If this is 0 or 1 there will be no visible change.
Default: Numbers will be formatted depending on the (amount of digits in **P3**).

P5 **- (numeric)** Variable or value can be 0 or 1 and describes if you want to overwrite existing Files or not.
Default is "1" (Overwrite existing files).

Speed in Ticks:

This command uses typically around 1020 Ticks.

Example

```
'*****************************************************
' EXAMPLE 2: FIL.gnf
'*****************************************************
$$MSG=Teste FIL "GNF mit Ordner".
$$LAB=$$LAAFolder_?\
$$LAC=Dies ist ein Text mit der number %n.
FIL.gnf|$$LAB|$$LAC|1|10|2|1
' TOS contains the number of created files
VAR.$$LAE=$tos$
MBX.Files created: $$LAE
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- **Split File** 1001
- **Join File** 994
- **OPT. - Optional Settings** 2406
- **GEC. - Get Encryption** 825
- **RMD. - Remove-Directory** 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.54.14.8 Find free Number

MiniRobotLanguage (MRL)

FIL.ffn
FIL.Find free Number

Find the next free number in a series of numbered files or folders

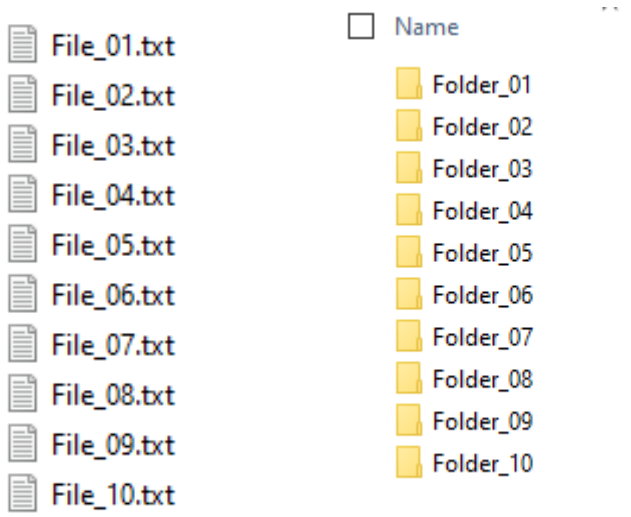
Intention

This command is used to find the next free number in a series of Files or Folders with a builtin Number.
Generally the "?" in the Filename is replaced with the number during the process.

```
$$MSG=Teste FIL "FFN Part 2".
$$LAB=$$LAAFile_?.txt
$$LAC=
' Create series of 10 files
FIL.gnf|$$LAB|$$LAC|1|5|3|1
VAR.$$LAE=$tos$
MBX. Created $$LAE files
$$LAC=1
$$LAD=3
$$LAE=0
FIL.ffn|$$LAB|$$LAC|$$LAD|$$LAE
'JIV.$$LAE!11|Lab_failed
MBX.Next Free Filenumber is: $$LAE
```

This is how such "Numbered files" or folders do look.

*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

FIL.ffn|P1|P2|P3[|P4][|P5]
FIL.find free number|P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 - Filename and Path to the first numbered file, it must contain a "?".
The "?" will be replaced with the formatted Number during the process.
If P1 ends with a \ then its taken as a folder-name.

P2 - **(numeric)** Variable or value with the first number (starting number).

P3 - **(numeric)** Variable or value with number of digits to format the number.
0/1 - do not format the numbers, e.g. File_2.txt
2 and higher: - format with number of digits in P3,
Default: Numbers will not be formatted.

P4 - **(numeric)** Variable with the result of the operation, that is a number between P1 and more.
If the last file that was found was File_010.txt, then this result will just be "11".

Speed in Ticks:

This command uses typically around 2150 Ticks.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 2: FIL.ffn
'*****************************************************
$$MSG=Teste FIL "FFN".
$$LAB=$$LAAFile_?.txt
$$LAC=
' Create series of 10 files
FIL.gnf|$$LAB|$$LAC|1|5|3|1
VAR.$$LAE=$tos$
MBX. Created $$LAE files
$$LAC=1
$$LAD=3
$$LAE=0
FIL.ffn|$$LAB|$$LAC|$$LAD|$$LAE
'JIV.$$LAE!11|Lab_failed
MBX.Next Free Filenumber is: $$LAE

ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- Split File 1001
- Join File 994
- OPT. - Optional Settings 2406
- GEC. - Get Encryption 825
- RMD. - Remove-Directory 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.9 Generate numbered Name

MiniRobotLanguage (MRL)

FIL.gnn
FIL.Generate numbered Name

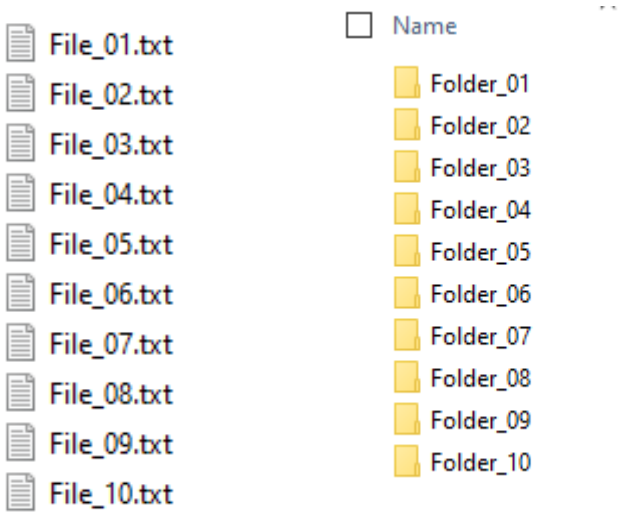
Just generate a Filename with a formatted Number in it

Intention

This command is used to generate a name for Files or Folders with a builtin Number, compatible with the otehr FIL.-commands of that sort.
Generally the "?" in the Filename is replaced with the number during the process.

```
'Teste FIL "GNN mit Folder".
VAR.$$LAA=?path\Testfolder\
$$LAB=$$LAAFile?\
$$LAC=5
$$LAD=3
FIL.gnn|$$LAB|$$LAC|$$LAD|$$RES
$$LAE=$$LAAFile005\
'JIS.$$RES!$$LAE|Lab_failed
MBX.Filename is: $$RES
```

This is how such "Numbered files" or folders do look.



 *© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

FIL.gnn|P1|P2|P3[|P4]
FIL.Generate numbered Filename|P1|P2|P3[|P4]

Parameter Explanation

P1 - Filename and Path, it must contain a "?".
The "?" will be replaced with the formatted Number during the process.
If P1 ends with a \ then its taken as a folder-name.

P2 - **(numeric)** Variable or value with the number to insert.

P3 - **(numeric)** Variable or value with number of digits to format the number.
0/1 - do not format the numbers, e.g. File_2.txt
2 and higher: - format with number of digits in P3,
Default: Numbers will not be formatted.

P4 - **(Output)** Variable with the result of the operation.
If omitted the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 2: FIL.gnn
'*****************************************************
'Teste FIL "GNN mit Folder".
VAR.$$LAA=?path\Testfolder\
$$LAB=$$LAAFile?\
$$LAC=5
$$LAD=3
FIL.gnn|$$LAB|$$LAC|$$LAD|$$RES
$$LAE=$$LAAFile005\
'JIS.$$RES!$$LAE|Lab_failed
MBX.Filename is: $$RES
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- **Split File** 1001
- **Join File** 994
- **OPT. - Optional Settings** 2406
- **GEC. - Get Encryption** 825
- **RMD. - Remove-Directory** 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.54.14.1 Move numbered Files

MiniRobotLanguage (MRL)

FIL.mov
FIL.Move numbered File

Move/Insert a file within a List of numbered Files up or down by swapping all intermediate files

Intention

This command is used to move files with a builtin Number between more such files.
For example, you want to move the File "File_02.txt" to Position "04" as "File_04.txt".
You could just Swap the files but then "File_02.txt" would be at Position "4".
So this command will now start swapping files in the right order, to insert the file at the right position.
All other files are just moved one place downward.
You could also think of "Insert numbered file" in this command.

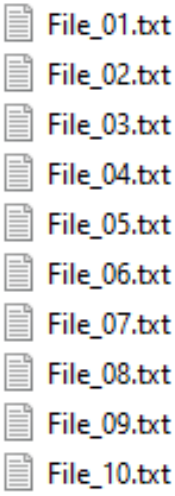
Move UP or DOWN?
If the file is moved "UP" or "DoWN", depends only on the Start- and End-Number Parameter.
If Start is higher, then the file is moved upwards until the END-Number.
If Start is lower number, then the file is moved downwards to the higher END-Number.

Generally the "?" in the Filename is replaced with the number during the process.

```
'Teste FIL "GNN mit Folder".
VAR.$$LAA=?path\Testfolder\
$$LAB=$$LAAFile_?.txt
' Move File_002 to the Place of File_009 shiftig the inbetween
files one place up
FIL.mov|$$LAB|2|8|3|$$RES
' Move file 009 to place of file 001 shiftig ann inbetween
files one place down
FIL.mov|$$LAB|9|1|3|$$RES
```

This is how such "Numbered files" look.

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

FIL.mov|P1|P2|P3[|P4][|P5]
FIL.Move numbered File|P1|P2|P3[|P4][|P5]

Parameter Explanation

P1 **-** Filename and Path, it must contain a "?".
The "?" will be replaced with the formatted Number during the process.
If P1 ends with a \ then its taken as a folder-name.

P2 **-** **(numeric)** Variable or value with the number of the File to Move.
P3 **-** **(numeric)** Variable or value with the number of the target-place.

P4 **-** **(numeric)** Variable or value with number of digits to format the number.
0/1 - do not format the numbers, e.g. File_2.txt
2 and higher: - format with number of digits in P3,
Default: Numbers will not be formatted.

P5 **-** **(Output)** Variable with the result of the operation.
If omitted the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 2: FIL.gnn
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'Teste FIL "MOV".
VAR.$$LAA=?path\Testfolder\
$$LAB=$$LAAFile_?.txt
$$LAC=$$LAAFile_002.txt
GMD.cf|$$LAC|$$CSA
FIL.mov|$$LAB|2|8|3|$$RES
JIV.$$RES!1|Lab_failed
$$LAC=$$LAAFile_008.txt
GMD.cf|$$LAC|$$CSD
JIS.$$CSA!$$CSD|Lab_failed
$$LAC=$$LAAFile_009.txt
GMD.cf|$$LAC|$$CSA
FIL.mov|$$LAB|9|1|3|$$RES
JIV.$$RES!1|Lab_failed
$$LAC=$$LAAFile_001.txt
GMD.cf|$$LAC|$$CSB
JIS.$$CSA!$$CSB|Lab_failed
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- Split File 1001
- Join File 994
- OPT. - Optional Settings 2406
- GEC. - Get Encryption 825
- RMD. - Remove-Directory 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.14.1 Swap Files

MiniRobotLanguage (MRL)

FIL.swp
FIL.Swap Files

Exchange the name of two files

Intention

This command is used to "Swap two Files". This means that File A becomes File B and File B becomes File A.

```
'Teste FIL "Swap Files".
VAR.$$LAA=?path\Testfolder\
$$LAB=$$LAAFile_005.txt
$$LAC=$$LAAFile_009.txt
FIL.swp|$$LAB|$$LAC|$$RES
' In case of success result will be "1"
MBX.$$RES
ENR.
```

Syntax

FIL.swp|P1|P2[|P3]
FIL.Swap Files|P1|P2[|P3]

Parameter Explanation

P1 - Filename and Path, of File A

P2 - Filename and Path, of File B

P3 - **(Output)** Variable with the result of the operation.
If omitted the result is placed on TOS.
1 - Success 0 - Error
The timeout Flag is also set in case of Error.

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'*****************************************************
' EXAMPLE 2: FIL.Swp
'*****************************************************
'Teste FIL "Swap Files".
VAR.$$LAA=?path\Testfolder\
$$LAB=$$LAAFile_005.txt
$$LAC=$$LAAFile_009.txt
GMD.cf|$$LAB|$$CSA
GMD.cf|$$LAC|$$CSB
FIL.swp|$$LAB|$$LAC|$$RES
GMD.cf|$$LAB|$$CSC
GMD.cf|$$LAC|$$CSD
JIS.$$CSB!$$CSC|Lab_failed
JIS.$$CSA!$$CSD|Lab_failed
FIL.swp|$$LAB|$$LAC|$$RES
GMD.cf|$$LAB|$$CSC
GMD.cf|$$LAC|$$CSD
JIS.$$CSB!$$CSD|Lab_failed
JIS.$$CSA!$$CSC|Lab_failed
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- Split File 1001
- Join File 994
- OPT. - Optional Settings 2406
- GEC. - Get Encryption 825
- RMD. - Remove-Directory 1077
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.15 FPC. - Filepath Physical Convert

MiniRobotLanguage (MRL)

FPC. Command

Filepath Physical Convert

Intention

Convert a **physical File-path** like:

\Device\HarddiskVolume4\projects\explorer-guard.exe

into a **Letter-File-Path** like this:

F:\projects\explorer-guard.exe

You can also go all the way back and convert a Letter-File-path into a physical File-path.

`FPC.` has two Modes of Operations and in each you can choose, in which direction you want to convert.

If you simply write:

```
VAR.$$FIL=\Device\HarddiskVolume4\projects\explorer-guard.exe
FPC.$$FIL
DBP.$$FIL
```

You have converted the physical File-path to a Letter-File-path. This is what you get:

```
[15] F:\projects\explorer-guard.exe
```

To go the opposite direction, all you need to do is, to prefix the filename with a ">" Sign. Like this:

```
VAR.$$TXA=F:\projects\explorer-guard.exe
FPC.>$$TXA
DBP.$$TXA
ENR.
```

This way you have converted the Letter-File-path to a physical File-path. This is what you get:

```
\Device\HarddiskVolume4\projects\explorer-guard.exe
```

You can give two Parameters. In that case **P1** is the source-file-path and **P2** is the variable that will get the result after the conversion.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Or you can just write this:

```
FPC.\Device\HarddiskVolume4\projects\explorer-guard.exe|$$RES
```

In this case, the result will be placed in `$$RES`. Of course you can use a variable for the source filename:

```
VAR.$$SRC=\Device\HarddiskVolume4\projects\explorer-guard.exe
FPC.$$SRC|$$RES
```
is equal to:

```
VAR.$$SRC=\Device\HarddiskVolume4\projects\explorer-guard.exe
FPC.<$$SRC|$$RES
```

and is the same like:

```
VAR.$$SRC=<\Device\HarddiskVolume4\projects\explorer-guard.exe
FPC.$$SRC|$$RES
```

All these will do the same. We have just omitted the"<" because its the default operation, if the ">" is not given.

Syntax

FPC.[</>]P1[|P2]

Parameter Explanation

[</>] - Direction of conversion:
> - Convert Letter-File-Path to physical File-Path
< - **(defaul, if omitted)** Convert physical File-Path to Letter File-Path

P1 – VAR with the Source-Filename. If **P2** is omitted, **P1** is also the destination.

P2 – ***(optional)*** VAR for the conversion result. If omitted, the result will be placed in **P1**.

Example

```
'**********************************
' FPC.-DEMO 1
'**********************************
VAR.$$TXA=>explorer-guard.exe
FPC.$$TXA
DBP.$$TXA
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' FPC.-DEMO 2
'**********************************
VAR.$$TXA=\Device\HarddiskVolume4\projects\explorer-guard.exe
FPC.$$TXA
DBP.$$TXA
ENR.


'**********************************
' FPC.-DEMO 3
'**********************************
FPC.\Device\HarddiskVolume4\projects\explorer-guard.exe|$$TXP
DBP.$$TXP
ENR.
```

Remarks

The file/folders must not physically exist, but the drive must exist and have a valid drive letter.

Limitations:

-

See also:

- LFN. - Long-File-Name 1058
- 3.3 Specialfolders 345
- CFF. - Copy-From-File 948
- ATF - Append-to-File 946
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.16 GFD. - Get File Date

MiniRobotLanguage (MRL)

GFD. Command

Get File Date

Intention

Returns a date information from a file. Depending on parameters this can be the:

1. creation date
2. last write
3. last access

date of the specified file.

If you just use

```
GFD.?desktop\SLI-Bridge.txt
```

This is what you get on the TOS:

It's the "Last Write" Date, delimited by *****. The reason, why it is delimited by ***** is, that this way you can more easy analyze the date using commands like SBD. 3070 and SBM. 3073 GES. 3067 Or FEM. 929

For most uses, these are the following three operation modes of GFD.

1. `GFD.$$FIL|$$RET|`**c**

This will put the **c**reation date of file $$FIL into $$RET.

2. `GFD.$$FIL|$$RET|`**a**

This will put the last **a**ccess date of file $$FIL into $$RET.

3. `GFD.$$FIL|$$RET|`**w**

This will put the last **w**rite date of file $$FIL into $$RET.

© 2023 Theo Gottwald * https://www.smart-package.com

To get more information on what to do with the results from this command, see also: 1.5.2 Working with Text-Strings 110.

Syntax

GFD.[P1][|P2[|P3]

Parameter Explanation

P1 - Filename
P2 - Variable or "-" to return the file-date (alternatively TOS is used)
P3 - optional c/a/w depending what file-date shall be returned

Example

```
''*******************************************
' GFD. - Sample
' Please use a path from files which are available
' on your system.
'*******************************************
'DBM.2
' Path to an existing file
VAR.$$FIL=?path\TestApp.exe
PRT.=========================================
GFD.$$FIL
PRT.Creation Date: $$000
GFD.$$FIL|-|a
PRT.Last access Date: $$000
GFD.$$FIL|$$RES|w
PRT.Last write Date: $$RES
PRT.=========================================
DMP.4
END.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- GFD. - Get File Date 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.54.17 GFI. - Get File Information

MiniRobotLanguage (MRL)

GFI. Command

Get File Information

Intention

Returns version and other information from a file. Depending on parameters this can be the:

1. CompanyName
2. FileDescription
3. FileVersion
4. InternalName
5. LegalCopyright
6. OriginalFilename
7. ProductName
8. ProductVersion

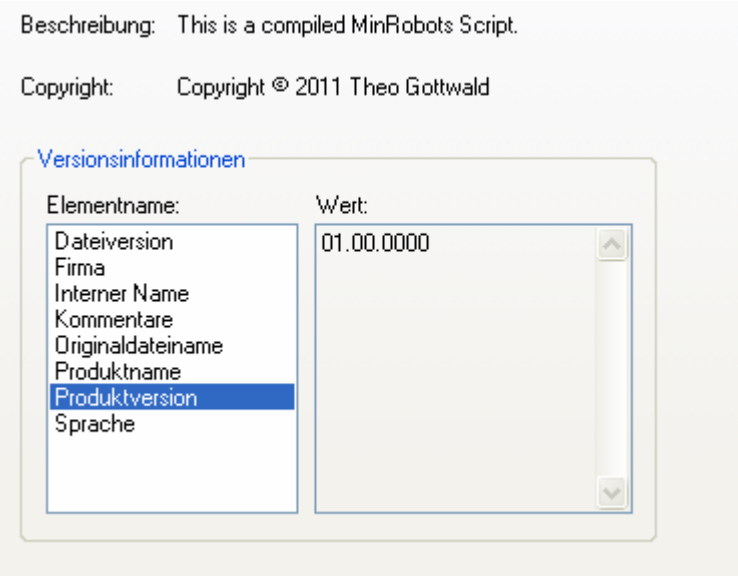
or any other specified Information which is accessible in the Version Tab of the File-Properties..

Using the script below, we will get something like this on the Stack 369:

```
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->String        :CompanyName:*Theo Gottwa
FileDescription:*This is a compiled MinRobots Script.*
FileVersion:*01.00.0000*
InternalName:*MR Dropship.exe (Vista/Win 7)*
LegalCopyright:*Copyright © 2011 Theo Gottwald*
OriginalFilename:*Dropship.exe*
ProductName:*MiniRobots Automation System*
ProductVersion:*01.00.0000*

------------------------------------
```

which is another representation of what you get to see if you look in the Properties dialog for the file version (see picture below).

© 2023 Theo Gottwald * https://www.smart-package.com



And here is the example script that produces this output:

```
GFI.?desktop\IC_Stromwandler.exe
STS.DUMP
MBX.Ready
ENR.
```

`GFI.` has several modes of operation. Let us see them in more detail. If we just write:

```
GFI.
```

then the command will try to get the needed filename from the Stack[369]. Filename is taken from TOS. Result is been put on TOS. Result contains all above listed file-Infos in one Element. Here is an example script:

```
VAR.$$FIL=?desktop\IC_Stromwandler.exe
PUV.$$FIL
GFI.
STS.DUMP
MBX.Ready
ENR.
```

Secondly, you can provide the filename in a variable.

```
VAR.$$FIL=?desktop\IC_Stromwandler.exe
GFI.$$FIL
```

The result is just the same as above. Result is been put on TOS. Result contains all above listed file-Information.

Now, if this is too much unneeded Information for you, you can be more selective what Information you want to have returned. Use a second parameter with the Info about what you want.

```
VAR.$$FIL=?desktop\IC_Stromwandler.exe
VAR.$$PAR=ProductVersion
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GFI.$$FIL|$$PAR|$$RES
PRT.Version: $$RES
```

The possible values for $$PAR are:

1. CompanyName
2. FileDescription
3. FileVersion
4. InternalName
5. LegalCopyright
6. OriginalFilename
7. ProductName
8. ProductVersion

and finally you can provide a country-code as **P4**, to get exactly the information you are looking for.

Syntax

GFI.[P1][|P2[|P3][|P4]

Parameter Explanation

P1 - Filename
P2 - Which Information to get
P3 - Variable or "-" to return the information,
if omitted TOS is used.
P4 - Country-code/Key-code to use.

GFI.P1

P1 - Filename

GFI.P1|P2

P1 - Filename P2 - Which Information to get
Result is been put on TOS. Result contains all above listed file-Infos.

GFI.P1|P2|P3

P1 - Filename
P2 - Which Information to get
P3 - Variable or "-" to return the information

GFI.[P1][|P2][|P3][|P4]

P1 - Filename
P2 - Which Information to get
P3 - Variable or "-" to return the information (alternatively TOS is used)

© 2023 Theo Gottwald * https://www.smart-package.com

P4 - Country-code/Key-code to use.

```
Country-Keycode can be a combination like:
  040904B0
 or can be omitted, in that case the code is automatically
choosen.
Alternatively a "*" can be specified in that case:
  040904E4
is used.
```

Example

```
'*********************************************
' GFI. - Sample
'*********************************************
'DBM.2
' Use Path to a existiing files
VAR.§§FIA=?path\PR01.Exe
VAR.§§FIB=?path\u9.exe
PRT.=========================================
' Get the Company-name on TOS
GFI.§§FIA|CompanyName

' Get Infos on TOS, separated with *
GFI.§§FIB

: §§VER=ProductVersion
GFI.§§FIB|§§VER|§§RES
PRT.Version: §§RES

: §§VER=ProductVersion
GFI.§§FIB|§§VER|§§RES
PRT.Version: §§RES

GFL.§§FIB|§§CKC|b
PRT.Country-Keyboard-Code: §§CKC

' uses §§CKC as StringFileInfo
GFI.§§FIA|CompanyName|§§RES|§§CKC
PRT.CompanyName: §§RES

' Uses 040904E4 as StringFileInfo
GFI.§§FIA|CompanyName|§§RES|*
PRT.CompanyName: §§RES

PRT.=========================================
DMP.4
END.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:
-

See also:

- 1.5.2 Working with Text-Strings 110
- **GFD. - Get File Date** 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137
- IEF. / NEF. - If - exist - File 1050

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.54.18 GFL.-Get File Language

MiniRobotLanguage (MRL)

GFL. Command

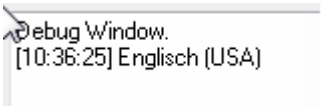
Get File Language

Intention

Returns the language from a file (exe or DLL).

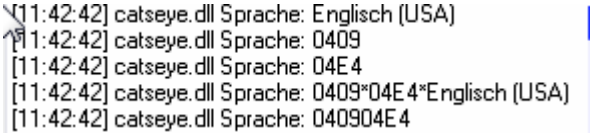
```
VAR.$$FIL=?exepath\MR01.Exe
GFL.$$FIL|$$RET
DBP.$$RET
ENR.
```

will return the information in a variable.



You can choose the Format of the Result using **P3**:

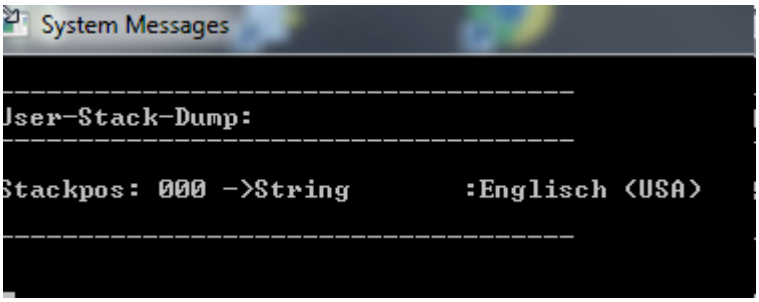
```
$$FIL=?exepath\catseye.dll
GFL.$$FIL|$$RES|t
DBP.catseye.dll Sprache: $$RES
GFL.$$FIL|$$RES|c
DBP.catseye.dll Sprache: $$RES
GFL.$$FIL|$$RES|k
DBP.catseye.dll Sprache: $$RES
GFL.$$FIL|$$RES|a
DBP.catseye.dll Sprache: $$RES
GFL.$$FIL|$$RES|b
DBP.catseye.dll Sprache: $$RES
```




```
VAR.$$FIL=?exepath\MR01.Exe
GFL.$$FIL
' Show the content of the stack
STS.DUMP
MBX.!
ENR.
```

will return the information on the Stack (TOS) 369.
You can see that in the picture below.

*© 2023 Theo Gottwald * https://www.smart-package.com*



You can also specify the format of the result:

```
GFV.?exepath\catseye.dll|§§RES
DBP.catseye.dll Version: §§RES
```

will result in something like:

```
catseye.dll Version: 3.0.2.24
```

If you need to compare file-versions, to find out if a file is never or older, use `IFL`.

Syntax

GFL.[P1][|P2][|P3]

Parameter Explanation

P1 **-** Filename (if omitted its taken from TOS)
P2 **- *(optional)*** Variable or "-" to return the file-date to TOS.
P3 **- *(optional)*** t/c/k/a/b - specify the format of the result as follows:

```
t - Text-Resut
c - codepage
k - keyboard code
a - all -> returns multiple formats delimited by a "*"
b - both
```

Here is the Format for **b** and **c**

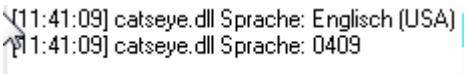
```
[11:42:42] catseye.dll Sprache: 0409*04E4*Englisch (USA)
[11:42:42] catseye.dll Sprache: 040904E4
```

Example

```
'*******************************************
' GFL./GFV. - Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*********************************************
$$FIL=?exepath\catseye.dll
GFV.$$FIL|$$RES|t
DBP.catseye.dll Sprache: $$RES
GFL.$$FIL|$$RES|c
DBP.catseye.dll Sprache: $$RES
```




```
'*********************************************
' GFL.- Sample
'*********************************************
DBP.=========================================
VAR.$$FIL=?exepath\MR01.Exe
GFL.$$FIL
DBP.MR01.exe Sprache: $$000
DBP.=========================================
GFL.?exepath\catseye.dll|$$RES
DBP.catseye.dll Version: $$RES
DBP.=========================================
'You can use a return variable instead of the stack
GFL.$$FIL|$$RES
DBP.Sprache: $$RES
DBP.=========================================
GFL.?exepath\catseye.dll|$$RES
DBP.catseye.dll Version: $$RES
DBP.=========================================
GFL.?wi\Notepad.exe|$$RES
DBP.Notepad Sprache: $$RES
DBP.=========================================
END.
```

Remarks

Here are the numeric values for the country code:

```
0000 = Language Neutral
0400 = Process Default Language
0436 = Afrikaans
041C = Albanian
0401 = Arabic (Saudi Arabia)
0801 = Arabic (Iraq)
0C01 = Arabic (Egypt)
1001 = Arabic (Libya)
1401 = Arabic (Algeria)
1801 = Arabic (Morocco)
1C01 = Arabic (Tunisia)
2001 = Arabic (Oman)
2401 = Arabic (Yemen)
2801 = Arabic (Syria)
2C01 = Arabic (Jordan)
3001 = Arabic (Lebanon)
3401 = Arabic (Kuwait)
3801 = Arabic (U.A.E.)
3C01 = Arabic (Bahrain)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
4001 = Arabic (Qatar)
042B = Armenian. Unicode only (Windows 2000 and later).
044D = Assamese. Unicode only (Windows 2000 and later).
042C = Azeri (Latin)
082C = Azeri (Cyrillic)
042D = Basque
0423 = Belarussian
0445 = Bengali. Unicode only (Windows 2000 and later).
0402 = Bulgarian
0455 = Burmese
0403 = Catalan
0404 = Chinese (Taiwan Region)
0804 = Chinese (PRC)
0C04 = Chinese (Hong Kong SAR, PRC)
1004 = Chinese (Singapore)
1404 = Chinese (Macau)
041A = Croatian
0405 = Czech
0406 = Danish
0413 = Dutch (Netherlands)
0813 = Dutch (Belgium)
0409 = English (United States)
0809 = English (United Kingdom)
0C09 = English (Australian)
1009 = English (Canadian)
1409 = English (New Zealand)
1809 = English (Ireland)
1C09 = English (South Africa)
2009 = English (Jamaica)
2409 = English (Caribbean)
2809 = English (Belize)
2C09 = English (Trinidad)
3009 = English (Zimbabwe)
3409 = English (Philippines)
0425 = Estonian
0438 = Faeroese
0429 = Farsi
040B = Finnish
040C = French (Standard)
080C = French (Belgian)
0C0C = French (Canadian)
100C = French (Switzerland)
140C = French (Luxembourg)
180C = French (Monaco)
0437 = Georgian. Unicode only (Windows 2000 and later).
0407 = German (Standard)
0807 = German (Switzerland)
0C07 = German (Austria)
1007 = German (Luxembourg)
1407 = German (Liechtenstein)
0408 = Greek
0447 = Gujarati. Unicode only (Windows 2000 and later).
040D = Hebrew
0439 = Hindi. Unicode only (Windows 2000 and later).
040E = Hungarian
040F = Icelandic
0421 = Indonesian
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
0410 = Italian (Standard)
0810 = Italian (Switzerland)
0411 = Japanese
044B = Kannada. This is Unicode only (Windows 2000 and later).
0860 = Kashmiri (India)
043F = Kazakh
0457 = Konkani. This is Unicode only (Windows 2000 and later).
0412 = Korean
0812 = Korean (Johab)
0426 = Latvian
0427 = Lithuanian
0827 = Lithuanian (Classic)
042F = Macedonian
043E = Malay (Malaysian)
083E = Malay (Brunei Darussalam)
044C = Malayalam. Unicode only (Windows 2000 and later).
0458 = Manipuri
044E = Marathi. Unicode only (Windows 2000 and later).
0861 = Nepali (India). Unicode only (Windows 2000 and later).
0414 = Norwegian (Bokmal)
0814 = Norwegian (Nynorsk)
0448 = Oriya. Unicode only (Windows 2000 and later).
0415 = Polish
0416 = Portuguese (Brazil)
0816 = Portuguese (Standard)
0446 = Punjabi. Unicode only (Windows 2000 and later).
0418 = Romanian
0419 = Russian
044F = Sanskrit. Unicode only (Windows 2000 and later).
0C1A = Serbian (Cyrillic)
081A = Serbian (Latin)
0459 = Sindhi
041B = Slovak
0424 = Slovenian
040A = Spanish (Traditional Sort)
080A = Spanish (Mexican)
0C0A = Spanish (Modern Sort)
100A = Spanish (Guatemala)
140A = Spanish (Costa Rica)
180A = Spanish (Panama)
1C0A = Spanish (Dominican Republic)
200A = Spanish (Venezuela)
240A = Spanish (Colombia)
280A = Spanish (Peru)
2C0A = Spanish (Argentina)
300A = Spanish (Ecuador)
340A = Spanish (Chile)
380A = Spanish (Uruguay)
3C0A = Spanish (Paraguay)
400A = Spanish (Bolivia)
440A = Spanish (El Salvador)
480A = Spanish (Honduras)
4C0A = Spanish (Nicaragua)
500A = Spanish (Puerto Rico)
0430 = Sutu
0441 = Swahili (Kenya)
041D = Swedish
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
081D = Swedish (Finland)
0449 = Tamil. Unicode only (Windows 2000 and later).
0444 = Tatar (Tatarstan)
044A = Telugu. Unicode only (Windows 2000 and later).
041E = Thai
041F = Turkish
0422 = Ukrainian
0420 = Urdu (Pakistan)
0820 = Urdu (India)
0443 = Uzbek (Latin)
0843 = Uzbek (Cyrillic)
042A = Vietnamese
```

Limitations:

-

See also:

- 3.3 Specialfolders 345
- IFL. / NFL. - If File Language 2155
- GFD. - Get File Date 1023
- GFI. - Get File Information 1026
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.19 GFT. - Get-File-Token

MiniRobotLanguage (MRL)

GFT. Command

Get File Token

Intention

GFT. evaluates a file-path. It returns a requested part of the file-name/path. The functionality is strictly one of parsing alone. No attempt is made to find the file on disk.

Here is an example:

```
$$FIL=c:\Folder\Filename.pdf
GFT.+hi|$$FIL|$$RET
' $$RET will return: "c:\Folder\Filename"
' Result will be placed in Clipboard
CLP.$$RET
END.
```

Syntax

GFT.P1|P2[|P3]...[|PX]

Parameter Explanation

P1: - Prototypes

We use $$FIL=c:\Folder\Filename.pdf as example for every option.

```
 a - Root-Drive -> "c:\"
 b - ALB 3045 -> "Filename.pdf"
 c - BLB 3051 -> "c:\Folder"
 d - File-Extension without "." -> "pdf"
 e - Filename without Extension -> "Filename"
 f - complete Path/filename without Root-Drive ->
"Folder\Filename.pdf"
 g - returns all without extension. -> "c:\Folder\Filename"
 h - Returns the path portion of the path/file name. -> "c:
\Folder\"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
     That is the text up to and including the last backslash
(\) or colon (:).
    i - Return the name portion of the path/file name. That is the text to the rig
        ending just before the last period (.). -> "Filename"
    j - Returns the extension portion of the path/file name.
        That is the last period (.) in the string plus the text to the right of it
 k - filename.extension -> "Filename.pdf"
 l - If specified, the first letter of the Parts of the
filename will be   Uppercased (MCase-Processing).
 m - remove "\" from right side of P2 before processing
     e.g. C:\file\ -> c:\file
 + - If the "+" Sign is specified in prototypes, the result
will be concatenated and all be placed in P3.
     Example:
     $$FIL=c:\Folder\Filename.pdf
     GFT.+hi|$$FIL|$$RET
     ' $$RET will return: "c:\Folder\Filename"
- - If the "-" Sign is specified as a prototype, any "\\?\"
that prefixes the filename will be removed.
    Otherwise the "\\?\" will be treated as a part of the
root-drive.
```

P2 **-** Filepath: Path which shall be split into parts. Must be given.

P3 **..** ***Px*** **-** ***(optional)*** Variables which will be filled with the Parts according to the prototypes

Example

```
'**********************************
' GFT.-Demo
'**********************************
VAR.§§MYP=?pfad\Myexe.dll
GFT.abcdef|§§MYP|§§AAA|§§BBB|§§CCC|§§DDD|§§EEE|§§FFF
PRT.-------------------------------------------
PRT.Full-Path is: §§MYP$crlf$GFT. makes:
PRT.Parameter a: §§AAA
PRT.Parameter b: §§BBB
PRT.Parameter c: §§CCC
PRT.Parameter d: §§DDD
PRT.Parameter e: §§EEE
PRT.Parameter f: §§FFF
PRT.-------------------------------------------
:enx
MBX.!
END.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **URL. - Split URL** 3234
- ALB - After last Backslah 3045
- BLB - Before last Backslash 3051
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.20 GFV. - Get File Version

MiniRobotLanguage (MRL)

GFV. Command

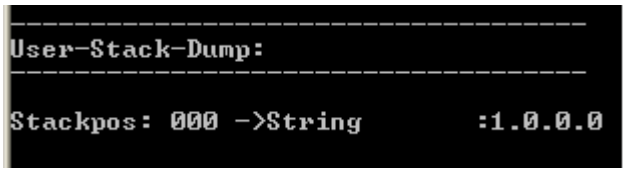
Get File Version

Intention

Returns the version information from a file (exe or DLL).

```
VAR.§§FIL=?exepath\MR01.Exe
GFV.§§FIL
STS.DUMP
```

will return the information on the Stack (TOS) 369.
You can see that in the picture below.



You can also specify a variable for the result:

```
GFV.?exepath\catseye.dll|§§RES
DBP.catseye.dll Version: §§RES
```

will result in something like:

```
catseye.dll Version: 3.0.2.24
```

If you need to compare file-versions, to find out if a file is never or older, use `IFV`.

The `GFV.`-Command has a search path. This path is used, if you omit the file-path.
The Search-Path is:

1. `?path\ - the path of the script-file itself.`
2. `?ws\ - the windows\system32 path`
3. `?wi\ - the windows-folder`

See Specialfolders 345.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

GFV.[P1][|P2]

Parameter Explanation

P1 **-** Filename (if omitted its taken from TOS)
P2 **-** ***(optional)*** Variable or "-" to return the file-date to TOS.

Example

```
'*******************************************
' GFV. - Sample
'
'*******************************************
DBP.=========================================
VAR.§§FIL=?exepath\MR01.Exe
GFV.§§FIL
DBP.MR01.exe Version: $$000
DBP.=========================================
GFV.?exepath\catseye.dll|§§RES
DBP.catseye.dll Version: §§RES
DBP.=========================================
'You can use a return variable instead of the stack
GFV.§§FIL|§§RES
DBP.Version: §§RES
DBP.=========================================
GFV.?exepath\catseye.dll|§§RES
DBP.catseye.dll Version: §§RES
DBP.=========================================
' You can omit path information if the DLL is in
' windows- or system32 or in the ?path\.
GFV.shlwapi.dll|§§RES
DBP.shlwapi.dll Version: §§RES
DBP.=========================================
END.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- **GFD. - Get File Date** 1023
- GFI. - Get File Information 1026
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.21 GLC. - Get-Line-Count

MiniRobotLanguage (MRL)

GLC. Command

Get Line Count

Intention

This command can be used to get the number of Lines of a given Text-File as well as the biggest line-width. Using with with binary files may lead to unexpected results.

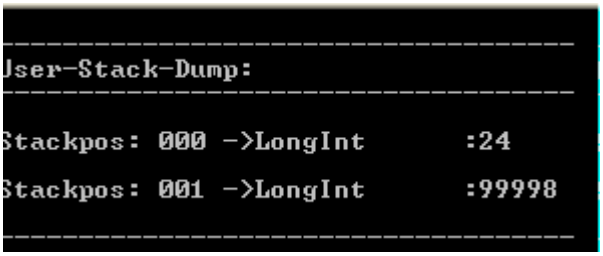
Usage is simple:

```
: $$FIL=C:My_Textfile.txt
GLC.$$FIL|$$TXT|$$WIT
MBX.The Linecount is: $$TXT
MBX. The largest line-width is: $$WIT$

You can omit all parameters except the filename.

GLC.$$FIL
```

In that case you find the number of lines on the stack. The second information, the maximum line with is dropped in that case. It will only be delivered when using a third variable parameter.

You can however do this:

```
GLC.$$FIL||$$000
STS.DUMP
MBX.Whats on the stack?
ENR.
```



And also mix variable output and stack output.

```
GLC.$$FIL||$$WIT
STS.DUMP
MBX.Width is $$WIT
ENR.
```

This will get the Line-count onto the stack and the maximum line-width into a variable.

© 2023 Theo Gottwald * https://www.smart-package.com

Speed Considerations:

Our tests show that a Textfile with 99998 Lines and **2.5 MB** in Size is scanned in about **0.03 seconds**.

Syntax

GLC.P1[|P2][|P3]

Parameter Explanation

P1 **-** Filename to count Lines in.

P2 **-** optional variable to return the number of lines. If omitted TOS is used.

P3 **-** optional variable to return the width of the longest line. If omitted Value is been placed on TOS.

Example

```
'*****************************************
' Demo for GLC.
'*****************************************
VAR.$$TIM=#dtime#
GLC.$$FIL|$$TXT|$$WIT
MBX.The Linecount is: $$TXT-$$WIT$crlf$#dsince#
ENR.

'*****************************************
' Demo for LFF and GLC.
'*****************************************
'DBM.2
' This should be a valid name of a Text-File
: §$FIL=start.ini
GLC.§$FIL|$$RET
PRT.File: $$FIL contains: $$RET Lines.
FOR.$$AAA|1|20
' Read Lin Nr. $$AAA from File ?path\$$FIL into $$RET
LFF.§$FIL|$$AAA|$$REB
PRT.[$$AAA]->$$REB
NEX.
MBX.!
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.22 GSI. - Get Shortcut Information

MiniRobotLanguage (MRL)

GSI. Command

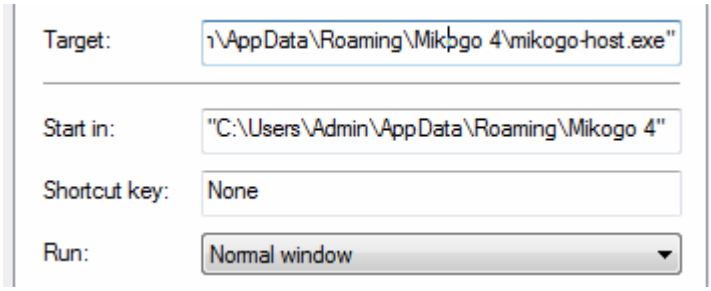
Get Shortcut Information

Intention

Returns Targetpath, workdirectory and other information from a specified shortcut. Depending on parameters this can be the:

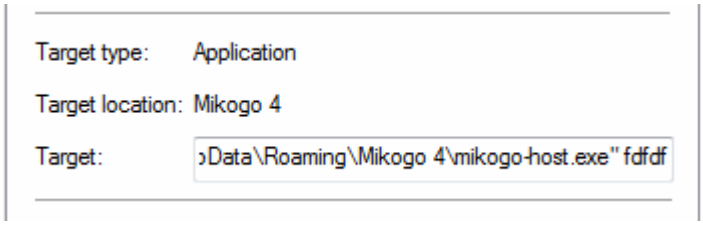
1. **LinkTarget**
2. **WorkDir**
3. **HotKey**
4. **ShowState**

these are equivalent to the contents of the fields in the dialog (see picture below).



Further you can get information about:

5. **Arguments**




```
$$LNK=?desktop\BSC-Testform
GSI.arguments|$$LNK
DBP.$$000
ENR.
```

In the case above, the result would be:

```
[13] fdfdf
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

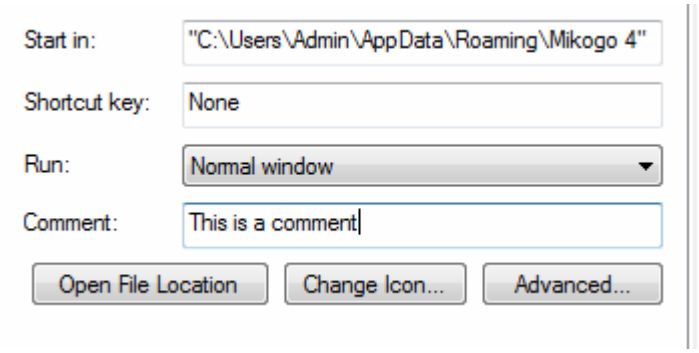
as these are the arguments of the commandline.

You can also return the following values:

6. **Hotkey**
7. **ShowState**
8. **Comment**

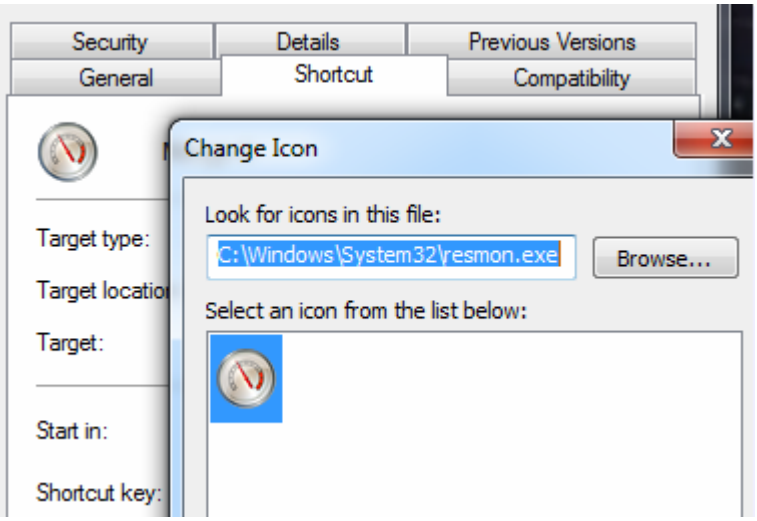
The following script will return the text in the "comment" field. Note that the number that is returned from the hotkey field is a "keycode" and you may only want to know if its a zero (no hotkey) or its a number.

```
$$LNK=?desktop\BSC-Testform
GSI.comment|$$LNK
DBP.$$000
ENR.
```



9. Iconlocation

returns the path of the location of the icon, if it is different from the link-target file.



10. show properties

You can make the command to open the file-properties dialog for any file (must not be a link).

© 2023 Theo Gottwald * https://www.smart-package.com

```
$$LNK=?desktop\BSC-Testform
GSI.sfp|$$LNK
DBP.$$000
PAU.5
ENR.
```

Syntax

GSI.P1|P2[|P3]

Parameter Explanation

P1 - Subcommand

"comment","com" - returns the text from the "Comment field"
"arguments","arg" - returns the arguments from the "Command line"
"hotkey","hot" - returns a zero if no hotkey is defined, or a keycode number.
"workdir","wd" - returns the text from the "Start in" field
"linktarget","tar" - returns the text from the "Target" field
"iconlocation","ico" - returns the path of the location of the icon, if it is different from the link-target file.
"show state","sho" - returns a numerical representation of the "Run"-Setting as follows:
1 - normal
3 - start maximized
7 - start minimized

"show properties","sfp" - Will open the file-properties dialog for the given file, folder or shortcut.

P2 - Filename
P3 - ***(optional)*** Variable for the result, if omitted TOS is used.

Example

```
'*******************************************
' GSI. - Sample
'*******************************************
$$LNK=?desktop\BSC-Testform
GSI.sho|$$LNK
DBP.$$000
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$LNK=?desktop\BSC-Testform.lnk
GSI.sfp|$$LNK
DBP.$$000
PAU.5
ENR.
```

Remarks

-

Limitations:

-

See also:

- CSC. - Create ShortCut 960
- 1.5.2 Working with Text-Strings 110
- GFI. - Get File Information 1026
- GFD. - Get File Date 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137
- IEF. / NEF. - If - exist - File 1050

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.23 IEF. / NEF. - If - exist - File

MiniRobotLanguage (MRL)

IEF. Conditional Statement

If Exist File / Folder

Intention

Conditional Statement. Test if a specified file or directory exists or not.
You can specify additional Parameters to test if something is a file or a dir.

When you specify a directory path, the robot will check if the last sign of the name/path is a \ (backslash).
If the last sign is a backlash, then the command will check for a directory (NOT for a file!).
If the last sign is NOT a backslash, the command will check for a file NOT for a directory.

```
' We can use wildcards.
' The real name of the file or dir is on the TOS
'
IEF.?desktop\t*.*
 PRT.Is a file and exists: $tos$
ELS.
  PRT. does not exist or is not a file.
EIF.
```

You can use `NEF.`(Not Exists File) as the negative Form of the command.

Syntax

IEF.P1[|P2] … ELS. … EIF.

Parameter Explanation

P1 **-** Variable/filename/directory-path ending with a \ (backslash)
can contain wildcards like **?** and *****
P2 **-** Options: these can only be an **e** - e like "exclusive"

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  e - "exclusive" will only go on true if its a file and the
file
       is not locked (can be accessed).
```

Speed in Ticks:

This command uses typically around 1900 Ticks depending on general files-system Speed.

Example

```
'*******************************************
' IEF-Sample
'*******************************************
' DBM.2
: §§FIL=?desktop\2Line.txt

IEF.§§FIL
 PRT.File exists: §§FIL
ELS.
  PRT.File doesn't exist: §§FIL
EIF.

' Works with Dirs too ...
IEF.?ws\
 PRT.DIr exists: ?ws\
ELS.
  PRT.Dir doesn't exist: ?ws\
EIF.

' We can use wildcards.
' The real name of the fiel or dir is on the TOS
'
IEF.?desktop\t*.*
 PRT.Is a file and exists: $tos$
ELS.
  PRT. doesn't exist or is not a file.
EIF.

END.
```

Remarks

`IEF.` and other IXX. can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX.` (Enumerations) to unlimited Depth.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

ATF - Append-to-File 946
CFF. - Copy-From-File 948
CTF - Copy-To-File 964
CTI. - Copy-To-Inifile 966
CFI. - Copy-From_IniFile 951
COP. - Copy File 957
GLC. - Get-Line-Count 1043
IEF. / NEF. - If - exist - File 1050
LFF - Line-From-File 1056
SHF. - Shell-File-Operation 1082
WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.24 IVA. - If Volume available

IVA. / NVA. - If-Volume Available Previous 1050 Top 41 Next 1056

MiniRobotLanguage (MRL)

IVA. Conditional Statement

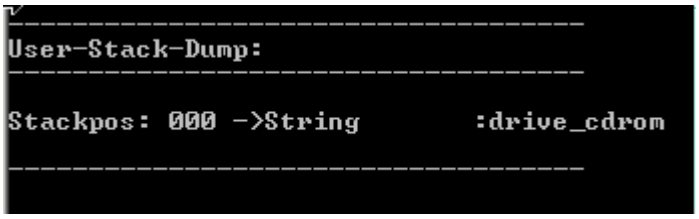
If Volume available

Intention

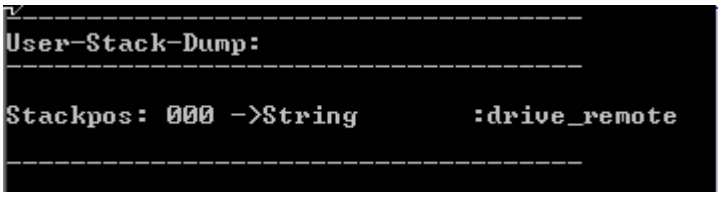
This command can be used to find out, if a specified Volume (Drive-letter) is available.

```
' Test if Volume e: is available
IVA.e
  DMP.6
  MBX.!
EIF.
```

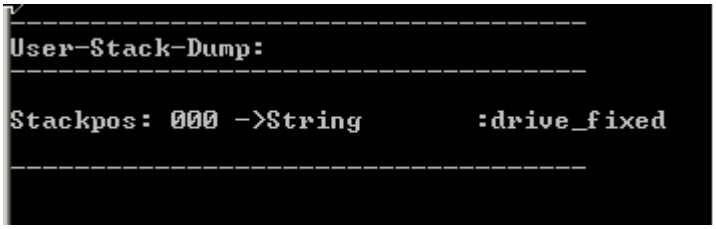
You will find the current usage on the TOS. In this case, E: is a CD-Rom or optical drive.



For a Network Drive you will get this.



and for a local Drive this:



You can specify the driveletter with- or without a ":" and in lowewr or in upper case.

```
IVA.Z:
  ' Show TOS
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  DMP.6
  MBX.!
EIF.
```

As a Bonus, you can also use the command like this:

```
$$PAT=\\PCNAME\Folder$
IVA.$$PAT|$$ERG
   PRT.$$ERG
   MBX.!
EIF.
ENR.
' Result could be "z:"
```

`NVA.` is the negative version of `IVA.`

Syntax

IVA.P1[|P2]

Parameter Explanation

P1 - driveletter, with ":" or without ":".
Alternative you can specify a path of a mapped drive to return the drive-letter.

P2 - (optional) - Variable for the result. If omitted TOS is used.

Example

```
'*******************************************
' IVA. - Sample
'*******************************************
IVA.e|$$ERG
   PRT.$$ERG
   MBX.!
EIF.
ENR.

'*******************************************
' IVA. - Sample 2
'*******************************************
$$PAT=\\PCNAME\Folder$
IVA.$$PAT|$$ERG
   PRT.$$ERG
   MBX.!
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
' Result could be "z:"
END.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- GVL. - Get Volume List 2105
- **MND. - Map Network Drive** 2108
- **CNS. - Connect Net Share** 2070
- **RDM. - Remove Drive Mapping** 2112
- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- GNS. - Get Network Share 2098
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.25 LFF. - Line-From-File

MiniRobotLanguage (MRL)

LFF. Command

Line From File

Intention

This command can be used to read any line from a file. Usage is simple:

```
' This is the Filename
VAR.$$FIL=?path\ 345 start.ini

' This is the Linen-umber of the Line we want read
VAR.$$LIN=5

' The Line will be in the variable $$RES
LFF.$$FIL|$$LIN|$$RES

' We print it in the Editor
DPB.The Line is: $$RES
```

Syntax

LFF.P1|P2[|P3]

Parameter Explanation

P1 - Filename to count Lines in.

P2 - value of the line which shall be read

P3 - optional variable to return the line from the file.
If omitted the Line is been placed on TOS.

Example

```
'*****************************************
' Demo for LFF and GLC.
'*****************************************
'DBM.2
' This should be a valid name of a Text-File
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
: §§FIL=start.ini
GLC.§§FIL|$$RET
PRT.File: $$FIL contains: $$RET Lines.
FOR.$$AAA|1|20
' Read Lin Nr. $$AAA from File ?path\$$FIL into $$RET
LFF.§§FIL|$$AAA|$$REB
PRT.[$$AAA]->$$REB
NEX.
MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- FEL - For-Each-Line 927
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.26 LFN. - Long-File-Name

MiniRobotLanguage (MRL)

LFN. Command

Long FileName

Intention

Convert a 8+1 (Short-Filename) into a Long Filename (LFN) .
If you simply write:

```
LFN.
```

then the command will try to get the source-filename (the one that should be converted) from the stack 369. If you add a variable like this:

```
LFN.§§TXT
```

then the command will take the the source-filename (the one that should be converted) from the given variable `§§TXT` and put the result on the stack 369.

The next step is, that you give two variables as parameters, like this:

```
LFN.§§TXT|§§LFN
```

in that case, the robot will take the source-filename (the one that should be converted) from the given variable `§§TXT` and put the result into `§§LFN`.

Syntax

LFN.[P1][|P2]

Parameter Explanation

P1 - ***(optional)*** VAR with the Short-Filename. If omitted, source is taken from TOS.

P2 - ***(optional)*** VAR for the result. If omitted, the result will be placed on TOS.

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'**********************************
' LFN.-DEMO
'**********************************
'DBM.2
DBP.-------------------------
' We work from Stack to Stack if no Parameters are specified.
: $$TXT=c:\progra~1
PUV.$$TXT
LFN.
DBP.Result:>$$000<
DBP.-------------------------
' We work from Variable to Stack if one Parameters is
specified.
: $$TXT=c:\progra~1
LFN.$$TXT
DBP.Result:>$$000<
DBP.-------------------------
' We work from Variable to Stack if one Parameters is
specified.
: $$TXT=c:\progra~1
LFN.$$TXT|$$LFN
DBP.Result:>$$LFN<
DBP.-------------------------
END.
```

Remarks

-

Limitations:

-

See also:

- 3.3 Specialfolders 345
- CFF. - Copy-From-File 948
- ATF - Append-to-File 946
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.27 LFT. - Line-From-Text

MiniRobotLanguage (MRL)

LFT. - Line-From-Text

Line from Text File that was loaded into internal Buffer for faster processing

Intention

When doing "Line-based" Operations on a Textfile, the use of LFF. (Line from File) can become slow.
Therefore the Combination of LOF. and LFT. was made. These Command-Pair works "in Memory".
Therefore you can access any Line of the Textfile immediately without any harddrive access.
This can speed up any Text-Processing significantly.

Please also take a look at the TXT.-Commands 3252. These Commands can be used, to save or change the Textfile.

LFT. only works together with LOF. So you need to "Load the Textfile" first using LOF. before you can use LFT.
Generally LFT. has two Working-Modes:

1. Get Line from FIle
LFT. with a number from 1 to (Number of Lines), will return the Line with that number in $$RET:

```
LFT.$$IND|$$NUM|$RET
```

2. Get number of Lines in FIle
LFT. with a number -1, will return the number of stored Lines from that file with that Index-Number.

```
LFT.$$IND|-1|$$RET
```

As you can not change the loaded file using LFT., there are two options in case you need to cahnge the file:
1. You can transfer the file to the TXT.-Command and work there. TXT. was designed specially for these tasks.
2. You can do this:
If you need to process the file, then the workflow is taht <ou read the Source-File using LOF. and LFT.
and write a changed version using ATF 946.- or CTF 964. - Command.
FInally you can "Switch the filenames" using the new FIL.Swap 1018.-Command.

If you call LFT. with a Number larger then the number of Lines in that File, the returned Variable will be empty.

Currently you can load up to 128 Textfiles into the Buffer at the same time, each having its own Index-Number from 1 to 128. This is very good if you need to work on multiple Text-Files at the same time, but you do not want to access the harddrive so

 *© 2023 Theo Gottwald * https://www.smart-package.com*

often. Using LOF. the file is buffered and all further Operations using LFT. are done in memory.

Usage is simple:

```
VAR.$$FIL=?path\Test.txt
' Read the File into Memory
LOF.$$IND|$$FIL
'The Index Number for this file is now in $$IND
LFT.$$IND|-1|$$RET
'Using LFT. with " -1" as Linenumber, we get the Number of
'Lines in return (here in $$RET).
' Now we enumerate through the File without accessing
' the harddrive, at maximum Speed
FOR.$$LOP|1|$$RET
  LFT.$$IND|$$LOP|$$LIN
  PRT.$$LIN
NEX.
MBX.Ready
ENR.
```

Syntax

LFT.P1|P2[|P3]

Parameter Explanation

P1 - ***(Variable or number)*** In this Variable the Index of the File must be specified. You receive the Index when you call the `LOF.`-Command.

P2 - ***Linenumber*** of the Line to read. If this is "-1" then the Number of Lines in the referenced file is returned.

P3 - ***(Variable)*** In this Variable the result of the Operation is returned. This is whether the Line from that file, or it is the number of lines in that file. If omitted the result will be placed on TOS.

Example

```
'*****************************************
' Demo for LOF and LFT.
'*****************************************
VAR.$$FIL=?path\Test.txt
LOF.$$IND|$$FIL
PRT.The Index Number for this file is >$$IND<
LFT.$$IND|-1|$$RET
PRT.Using LFT. with " -1" we get the Number of Lines.
PRT.Which is $$RET.
' Now we enumerate through the File without accessing
' the harddrive, at maximum Speed
FOR.$$LOP|1|$$RET
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  LFT.$$IND|$$LOP|$$LIN
  PRT.$$LIN
NEX.
MBX.Ready
ENR.
```

Remarks

The current Limitation is that you can not load more then 128 Textfiles at the same time into Memory.
Also the delimiters must be $crlf$. If that is not the case, the Text-Files may be automatically converted when they are loaded.
For example, if there are only $CR in a File that is loaded, the Loader will add a $LF to each $CR so that in the Buffer you have a $crlf$ delimited FIle.

Limitations:
Currently you can load up to 128 Textfiles into the Buffer at the same time, each having its own Index-Number from 1 to 128. This is very good if you need to work on multiple Text-Files at the same time, but you do not want to access the harddrive so often. Using LOF. the file is buffered and all further Operations using LFT. are done in memory.
You can not change the loaded file, this is the price for the speed that you get using this command-Pair.
If you need to make changes to the file, switch over to the TXT.-Command 3252.

See also:

- TXT.Get_from_LOF 3452
- TXT.To_LOF 3459
- LOF. - Load-Text-File 1063
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- FEL - For-Each-Line 927
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.28 LOF. - Load-Text-File

MiniRobotLanguage (MRL)

LOF. - Load-Text-File

Load Text File into internal Buffer for faster processing

Intention

When doing "Line-based" Operations on a Textfile, the use of LFF. (Line from File) can become slow.
Therefore the Combination of LOF. and LFT. was made. These Command-Pair works "in Memory".
Therefore you can access any Line of the Textfile immediately without any harddrive access.
This can speed up any Text-Processing significantly.

This command can be used together with the LFT. - command to process Text files faster.
Please also take a look at the TXT.-Commands 3252. These Commands can be used, if there is more to do then just loading a file "Line by line".
It has two Working-Modes:

1.Load Text FIle into Buffer and return Index (`$$IND`):

```
LOF.$$IND|$$FIL
```

2. Remove Text File (`$$IND` is the Index) from Buffer

```
LOF.!$$IND
```

Currently you can load **up to 128 Textfiles** into the Buffer at the same time, each having its own Index-Number from 1 to 128.
This is very good if you need to work on multiple Text-Files at the same time, but you do not want to access the harddrive so often.
Using LOF. the file is buffered and all further Operations using LFT. are done **in memory**.

If you want to **change or save the loaded Text-File**, you need to transfer it to the TXT.-Command 3252 using:

```
VAR.$$FIL=?path\Test.txt
' Read the File into Memory
LOF.$$IND|$$FIL
' Transfer LOF.-Buffer to TXT.-Buffer clear LOF.-Buffer
TXT.gfl|$$IND
' Make all changes and save Text using the TXT.-Commands
TXT.Save
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Transfer Text back to LOF.-Buffer, clear TXT.-Buffer
TXT.tol|$$IND
```

Usage is simple:

```
VAR.$$FIL=?path\Test.txt
' Read the File into Memory
LOF.$$IND|$$FIL
'The Index Number for this file is now in $$IND
LFT.$$IND|-1|$$RET
'Using LFT. with " -1" as Linenumber, we get the Number of
'Lines in return (here in $$RET).
' Now we enumerate through the File without accessing
' the harddrive, at maximum Speed
FOR.$$LOP|1|$$RET
  LFT.$$IND|$$LOP|$$LIN
  PRT.$$LIN
NEX.
MBX.Ready
ENR.
```

Syntax

LOF.P1|P2

Parameter Explanation

P1 - ***(Variable)*** In this Variable the Index of the File is returned.
If P1 starts with a "!" then the FIle which is indexed by P1 will be released from Memory. You do not need to release Files from memory, this is done automatically when the Script ends.

P2 - `Filename` of the File to read.

Example

```
'*****************************************
' Demo for LOF and LFT.
'*****************************************
VAR.$$FIL=?path\Test.txt
LOF.$$IND|$$FIL
PRT.The Index Number for this file is >$$IND<
LFT.$$IND|-1|$$RET
PRT.Using LFT. with " -1" we get the Number of Lines.
PRT.Which is $$RET.
' Now we enumerate through the File without accessing
' the harddrive, at maximum Speed
FOR.$$LOP|1|$$RET
  LFT.$$IND|$$LOP|$$LIN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  PRT.$$LIN
NEX.
MBX.Ready
ENR.
```

Remarks

The current Limitation is that you can not load more then 128 Textfiles at the same time into Memory.
Also the delimiters must be $crlf$. If that is not the case, the Text-Files may be automatically converted when they are loaded.
For example, if there are only $CR in a File that is loaded, the Loader will add a $LF to each $CR so that in the Buffer you have a $crlf$ delimited FIle.

Limitations:

Currently you can load up to 128 Textfiles into the Buffer at the same time, each having its own Index-Number from 1 to 128. This is very good if you need to work on multiple Text-Files at the same time, but you do not want to access the harddrive so often. Using LOF. the file is buffered and all further Operations using LFT. are done in memory.

See also:

- TXT.Get_from_LOF 3452
- TXT.To_LOF 3459
- LFT. - Line-From-Text 1060
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- FEL - For-Each-Line 927
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.29 MDF. - Make-Direktory-File

MiniRobotLanguage (MRL)

MDF. Command

Make Directory File

Intention

This command can be used to create a file and a directory in one strike.
It will create an directory with all folders and then create the file.

If the file/folder already exists or if the folder can't be created you will get an error in a return variable, if given.

You can synchronize programs using a file. This command will create a file instantly and tell you if it was successful. Usage is simple:

```
VAR.$$DIR=?desktop\MydirA\MyDirB\MyDirC\Myfile.txt
' Will create the directory and the file.
' If the directory exists, only the file is created.
MDF.$$DIB
```

Syntax

MDF.P1[|P2][|P3]

Parameter Explanation

P1 - Filepath

P2 - ***(optional)*** Filedata, Data to store in the file.
If Parameter 2 is omitted, an empty file is created.

P3 - ***(optional) Variable,*** Result of Operation (Error-Code)

Example

```
'************************************
' MFD - DEMO
'************************************
'DBM.2
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
VAR.§§DIR=?desktop\MydirA\MyDirB\MyDirC\Myfile.txt
VAR.§§DIB=?desktop\MydirA\MyDirB\MyDirC\Empty.txt
VAR.$$TXT=This text will be in that file$crlf$Line 2.
MDF.§§DIR|$$TXT|$$ERG
MDF.§§DIB
MBX.Success: $$ERG
END.
```

Remarks

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- DEL. - Delete File 974
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.30 MKD. - Make-Directory

MiniRobotLanguage (MRL)

MKD. Command

MaKe Directory

Intention

This command can be used to create unlimited deep folders in one strike.
If the folder already exists or if the folder can't be created you won't get an error message.

Syntax

MKD.P1

Parameter Explanation

P1 - Name of the Directory you want to create

P1 is necessary. For safety reason it can't be omitted.

Example

```
''************************************
' MKD - DEMO
' This will create some Folders on
' your desktop.
''************************************
'DBM.2
'
VAR.$$DIR=?desktop\MydirA\MyDirB\MyDirC
MKD.$$DIR
END.
```

Remarks

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- DEL. - Delete File 974
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.31 MOV. - Move File/Dir

MiniRobotLanguage (MRL)

MOV. Command

Move file or directory

Intention

This command is used to Move/rename files.
The command puts a 0 (Operation failed or was aborted) or a 1 (operation successful) on the stack to show if the operation was successful or not.
The Command has a network timeout when used with files that are possibly openend.
The command in its default-settings does not ask for any confirmation and can do serious damages to your files and data.
Therefore be careful and use on your own risk!
Unlike the REN. - Command, it will also MOVE Folders across devices. For example it will move a Folder from drive D: to drive F:.
In this case, you may write:

```
MOV.d:\folder-to-move\|f:\
```

This will move the folder to
`f:\folder-to-move\`

This can not be done using `REN.`
This is the only difference to the REN.-Command.

```
$$FIA=<Filename A>
$$FIB=<Filename B>
MOV.$$FIA|$$FIB
ITO.
  MBX.An Error occured!
EIF.
ENR.
```

To show that something is a folder, suffixe it with a "\".

```
' Is a file
$$FIA=C:\myname

' Is a folder
$$FIA=C:\myname\
```


© 2023 Theo Gottwald * https://www.smart-package.com

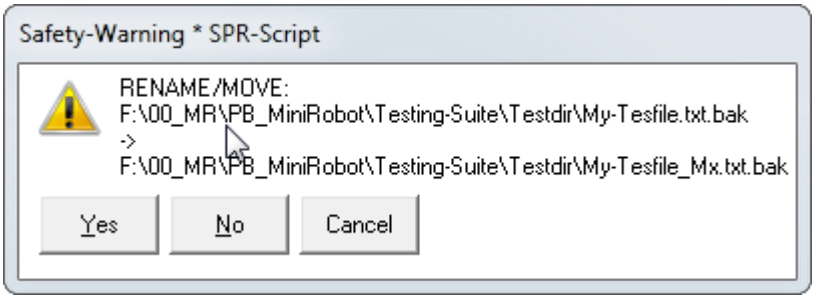
If you want to have a Warning-Messagebox before the Command does any RENAME, use the OPT.-Command and the SAFEFILE-Setting.

```
§§FIA=<Filename A>
§§FIB=<Filename B>
' Using this setting will produce a Safety Message-Box before dangerous File-Ope
OPT.SAFEFILE|1
MOV.§§FIA|§§FIB
ITO.
  MBX.An Error occured!
EIF.
ENR.
```

If you want to debug your Script safely, you can use the

```
OPT.SAFEFILE|1
```

this will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`



You can choose between

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

Syntax

MOV.P1|P2[|P3]

Parameter Explanation

P1 – Variable with source-file
P2 – Variable with Destination file
P3 – (optional) Timeout, only for renaming of files

Wildcards are not supported.

The Timeout will not be applied if directories shall be renamed.

© 2023 Theo Gottwald * https://www.smart-package.com

For file-operations using wildcards and more options, use SHF.

Example

```
'*****************************************************
' EXAMPLE: MOV. and COP.
'*****************************************************
: §§PAT=F:\Testfiles
VAR.§§FIL=§§PAT\Audiograbber.zip
VAR.§§TAC=§§PAT\recent\Target.zip
: §§TIM=2
VAR.§§TAR=§§PAT\Target.zip
VAR.§§TAB=§§PAT\Targez.zip

COP.§§FIL|§§TAB
COP.§§FIL|§§TAB|(§§TIM+3)
' If Operation was successfull then Stack contains a 1 else a
0
PRT.$$000
IVV.$$000=1
 PRT.Copy was successful
ELS.
 PRT. There was an Error.
EVV.

' Rename-Example
MOV.§§TAR|§§TAB

' Rename-Example with File-Timeout of 5 Seconds
MOV.§§TAR|§§TAB|5
END.
```

Remarks

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- REN. - Rename Files/Dirs 1074
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068

© 2023 Theo Gottwald * https://www.smart-package.com

- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- DEL. - Delete File 974

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.32 REN. - Rename File/Dir

MiniRobotLanguage (MRL)

REN. Command

REName file or directory

Intention

This command is used to rename files or folders.
The command puts a 0 (Operation failed or was aborted) or a 1 (operation successful) on the stack to show if the operation was successful or not.
The Command has a network timeout when used with files that are possibly openend.
The command in its default-settings does not ask for any confirmation and can do serious damages to your files and data.
Therefore be careful and use on your own risk!
The REN. - Command, it will NOT MOVE Folders across devices. For example it will NOT move a Folder from drive D: to drive F:.
For this use the MOV.-Command.

```
§§FIA=<Filename A>
§§FIB=<Filename B>
MOV.§§FIA|§§FIB
ITO.
  MBX.An Error occured!
EIF.
ENR.
```

To show that something is a folder, suffixe it with a "\".

```
' Is a file
§§FIA=C:\myname

' Is a folder
§§FIA=C:\myname\
```

If you want to have a Warning-Messagebox before the Command does any RENAME, use the OPT.-Command and the SAFEFILE-Setting.

```
§§FIA=<Filename A>
§§FIB=<Filename B>
' Using this setting will produce a Safety Message-Box before dangerous File-Ope
OPT.SAFEFILE|1
REN.§§FIA|§§FIB
ITO.
```

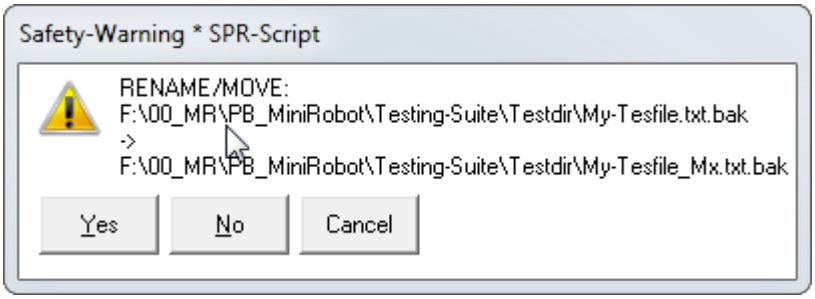

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
    MBX.An Error occured!
  EIF.
  ENR.
```

If you want to debug your Script safely, you can use the

```
OPT.SAFEFILE|1
```

this will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`



You can choose between

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

Syntax

REN.P1|P2[|P3]

Parameter Explanation

P1 - Variable with source-file
P2 - Variable with Destination file
P3 - (optional) Timeout, only for renaming of files

Wildcards are not supported.

The Timeout will not be applied if directories shall be renamed.
For file-operations using wildcards and more options, use SHF.

Example

```
'*****************************************************
' EXAMPLE: REN. and COP.
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: §§PAT=F:\Testfiles
VAR.§§FIL=§§PAT\Audiograbber.zip
VAR.§§TAC=§§PAT\recent\Target.zip
: §§TIM=2
VAR.§§TAR=§§PAT\Target.zip
VAR.§§TAB=§§PAT\Targez.zip

COP.§§FIL|§§TAB
COP.§§FIL|§§TAB|(§§TIM+3)
' If Operation was successfull then Stack contains a 1 else a
0
PRT.§§000
IVV.§§000=1
 PRT.Copy was successful
ELS.
 PRT. There was an Error.
EVV.

' Rename-Example
REN.§§TAR|§§TAB

' Rename-Example with File-Timeout of 5 Seconds
REN.§§TAR|§§TAB|5
END.
```

Remarks

-

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- DEL. - Delete File 974

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.33 RMD. - Remove-Directory

MiniRobotLanguage (MRL)

RMD. Command

ReMove Directory

Intention

This command is used to delete a folder and all of its contents.
This command will override software write protection.
Use with extreme caution.
You have to specify the clear path without wildcards.
This was done intentionally to prevent the deletion of unwanted content.

Usage is:

```
RMD.(dir to remove)
```

In case there are write-protected files and dirs in there and you want to kill'em all, choose the "force flag":

```
RMD.(dir to remove)|1
```

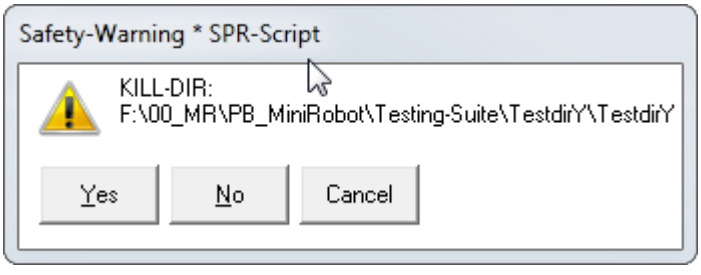
Finally you can specify a variable that will receive a 0 or a 1 if the folder could be removed or not.

```
RMD.(dir to remove)|$$RES
MBX.$$RES
```

If you want to debug your Script safely, you can use the

```
OPT.SAFEFILE|1
```

this will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`



You can choose between

© 2023 Theo Gottwald * https://www.smart-package.com

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

Please note that the Explorer and Anti-Virus Tools may lock files and folders and prevent the `RMD.`-Command from working as expected. In this case the Command will not delete the folder completely and the Timeout-Flag will be set.

Syntax

RMD.P1[|P2]

Parameter Explanation

P1 **-** `Filename to delete`
P2 **-** `opt. Variable for result (1/0)`

Example

```
'*****************************************************
' EXAMPLE 1: RMD.
'*****************************************************
'DBM.2
VAR.§§FIL=?desktop\Kopie von Joomla_1.7-Stable-Full_Package
RMD.§§FIL|§§RES
PRT.§§RES -> success?
END.
```

Remarks

This command is somehow a "DELTREE" command, it can potentially delete your whole harddisk, be careful and use on your own risk!

Please note that the Explorer and Anti-Virus Tools may lock files and folders and prevent the `RMD.`-Command from working as expected. In this case the Command will not delete the folder completely and the Timeout-Flag will be set.

Limitations:

Has not been tested with asian- or unicode file-systems.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- DEL. - Delete File 974
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074
- OPT. - Optional Settings 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.34 SFD. - Set File Date

MiniRobotLanguage (MRL)

SFD. Command

Set File Date

Intention

Set a date information from a file. Depending on parameters this can be the:

1. creation date
2. last write
3. last access

date of the specified file.

For most uses, these are the following three operation modes of `SFD`.

1. `SFD.$$FIL|$$RET|`**`c`**

This will change the **c**reation date of file $$FIL.

2. `SFD.$$FIL|$$RET|`**`a`**

This will change the last **a**ccess date of file $$FIL.

3. `GFD.$$FIL|$$RET|`**`w`**

This will change the last **w**rite date of file $$FIL.

Syntax

SFD.P1|P2[|P3]

Parameter Explanation

P1 - `Filename`
P2 - `Timestamp with Time to use. Format "YYYY:MM:DD:HH:MM:SS"`
P3 - `optional c/a/w depending what file-date shall be changed.`

Example

```
''*****************************************
' GFD. - Sample
' Please use a path from files which are available
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' on your system.
'*****************************************
'DBM.2
' Path to an existing file
VAR.§§FIL=?path\TestApp.exe
PRT.=========================================
GFD.§§FIL
PRT.Creation Date: $$000
GFD.§§FIL|-|a
PRT.Last access Date: $$000
GFD.§§FIL|§§RES|w
PRT.Last write Date: §§RES
PRT.=========================================
DMP.4
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
-
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.35 SHF. - Shell-File-Operation

MiniRobotLanguage (MRL)

SHF. Command

SHell File operation

Intention

This command is used to copy, rename, move or delete files or folders.
You can specify wildcards "*" and "?".

The command puts a 0 (Operation failed or was aborted) or a 1 (operation successfull) on the stack to show if the operation was successful or not.

`SHF`. Supports the use of wildcards (* and ?). Also a progress-dialog can be displayed using the "p"-flag.

The command in its default-settings does not ask for any confirmation and can do serious damages to your files and data. Therefore be careful and use it at your own risk!

On an test-system a file with 11.1 GB was copied in ~169 seconds using `SHF`. (Shell-File Operations). These times depend on the specific test system.

```
' Copy file with SHF.
VAR.$$TIM=#dtime#
: $$SRC=F:\05-26-05.pst
: $$TAR=?desktop\Testfile.pst
SHF.c|pcm|$$SRC|$$TAR
STS.DUMP
MBX.Ready: #dsince#
ENR.

' Copy all content of the
' folder "Test" into the Folder "tar"
'
VAR.$$SRC=?pfad\Test\*
VAR.$$TAR=?pfad\tar\
SHF.c|p|$$SRC|$$TAR
ENR.
```

This command internally calls the "Windows-Shell" and therefore does currently not support the "NTFSD-Feature".

Syntax

SHF.P1|P2|P3[|P4]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - Operations-Code

```
  c - copy,
  m - move,
  r - rename or
  d - delete
```

P2 - Flags

```
  p - progress-dialog ( if not set, no progress is
displayed. )
  c - confirmation. If set you may need to responds to dialog
boxes
      that is displayed on file-conflicts.
      Default is that the conflict is solved with "Yes to
All",
      that is, everything is forced.
  r - rename-on-collision. Gives the file being operated on a
new
      name (such as "Copy #1 of...")in a move, copy, or rename
       operation if a file of the target name already exists.
  f - Files-Only. Performs the operation only on files if a
      wildcard filename (*.*) is specified.
  m - Confirm-MakeDir. If specified, does need the user to
confirm
       the creation of a new directory if the operation
requires
       one to be created.
  u - Allow-Undo. Preserves undo information, if possible.
      By adding the Allow-undo flag you can move a file to the
      Recycle Bin instead of deleting it.
```

P3 - Variable with source-file
P4 - must be there for any operation other then delete.

Example

```
'*****************************************************
' EXAMPLE 1: SHF.
'*****************************************************
: §§PAT=F:\Testfiles
VAR.§§FIL=§§PAT\Audiograbber.zip
VAR.§§TAR=§§PAT\Target.zip
VAR.§§TAC=§§PAT\recent\Target.zip
VAR.§§TAB=§§PAT\Targez.zip

' Simple copy-Operation
SHF.c||§§FIL|§§TAR

' Rename File Operation
SHF.r||§§TAR|§§TAB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Move File Operation, where rename doesnt work
SHF.m||§§TAB|§§TAC

PAU.2
' Delete File Operation
SHF.d||§§TAB

END.
```

Remarks

This command makes permanent changes to the file-system. Use with caution.

Limitations:

-

See also:

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- REN. - Rename File/Dir 1074
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.54.36 WFF. - Wait-For-File

MiniRobotLanguage (MRL)

WFF. Command

Wait For File

Intention

This command is useful to make your batch scripts error-tolerant. Using WFF. you can really wait until a file is available and accessible. Or you can use it to wait until a file is really deleted.
This way you can synchronize batch scripts via files.

`WFF.` can be used in two ways:

1. Wait until a file is available
2. Wait until a file has disappeared

In both cases you can set a "timeout" value, or wait endless. In case you use a Timeout-Value, the timeout-Flag will be set only if that timeout was reached.
You can check the TImeout-Flag using the `JIT.` or `ITO.` and such commands.

If you wait for a file, `WFF.` will also check if the file is accessible. If it can be "opened for read".
If the file is still locked because its not yet "ready", WFF. will not accept the file in case 1 (Wait for a file).

In case 2 (Wait for a file to be deleted), the file must disappear, a locked file does not count as "deleted".

This system makes `WFF.` a perfect companion for synchronizing Batch-Jobs with compilers and such.
Because it will wait until one task is ready and then start the next.

```
' Sample Script
 VAF.$$FJB=$$FIA\Openfile_lock*.txt
 WFF.$$FJB|1
 JIT.ero
```

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

WFF.[P1] [|P2]

```
' Wait for file
WFF.(Filename)
WFF.(Filename)|Timeout in sec.
' Wait for file being deleted
WFF.<>(Filename)
WFF.<>(Filename)|Timeout in sec.
```

Parameter Explanation

P1 - File Name

P2 - Timeout in seconds

If you add the "**<>**" before the filename, WFF. will wait until the file is been deleted.

Example

```
'*****************************************************
' EXAMPLE 1: WFF/ATF. Append without Newline
' Create a file containing large numbers
' Synchronize two robots using a file.
'*****************************************************
: $$FIL=?desktop\RND.txt
' Run a second robot.
' Note that the second robot knows the value of $$FIL
PRR.
  DOL.
    PRT.Parallelrobot started, waiting for File ...
    WFF.$$FIL
    PRT.File existst ... now waiting for the file to be
deleted.
    WFF.<>$$FIL
    PRT.File was removed. Ending ...
    PAU.2
    GTO.out
  OOP.
  :out
  END.
PRE.
' This is the filename and path on the desktop
: $$FIL=?desktop\RND.txt

FOR.$$AAA|1|20
' Generate a Random Number between 124 and 123125
  RND.124,123125,$$REB
' Append the Numbers to the File without Newline
  ATF.$$FIL|$$REB|$$ERG
NEX.
DEL.$$FIL
MBX.Ready
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

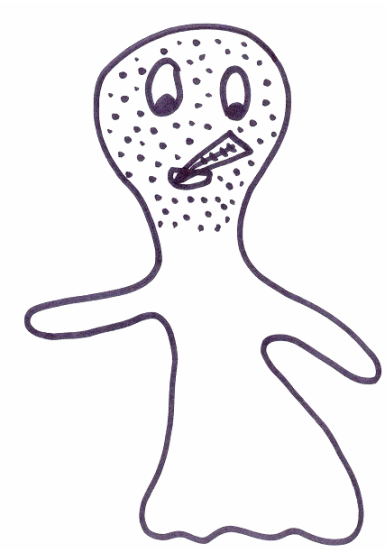
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- FEL - For-Each-Line 927
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082

© 2023 Theo Gottwald * https://www.smart-package.com

3.55 Image/Pixel-Color Operations

3.55.1 ! Smarty's Pixel-Operations

Smarty's Pixel Operations



Pixel-Operations can be used as last choice, if no other command can give you needed information on these topics:

- Did something in the dialog change? (`WRA.`)
- Is the button i need already available?
- Is the form ready to accept input?
- Or you can simply use it to take a Snapshot, for example under Error-conditions, to see what was going on (`CAP.`).

Pixel-Operations are most often the last choice, if all other commands fail to read the window content. The reasons are simple:

- Colors may differ from system to system
- coordinates may differ depending on screen dpi-settings and resolution

To make Pixel Operations as compatible as possible, you can use two special tools, provided here:

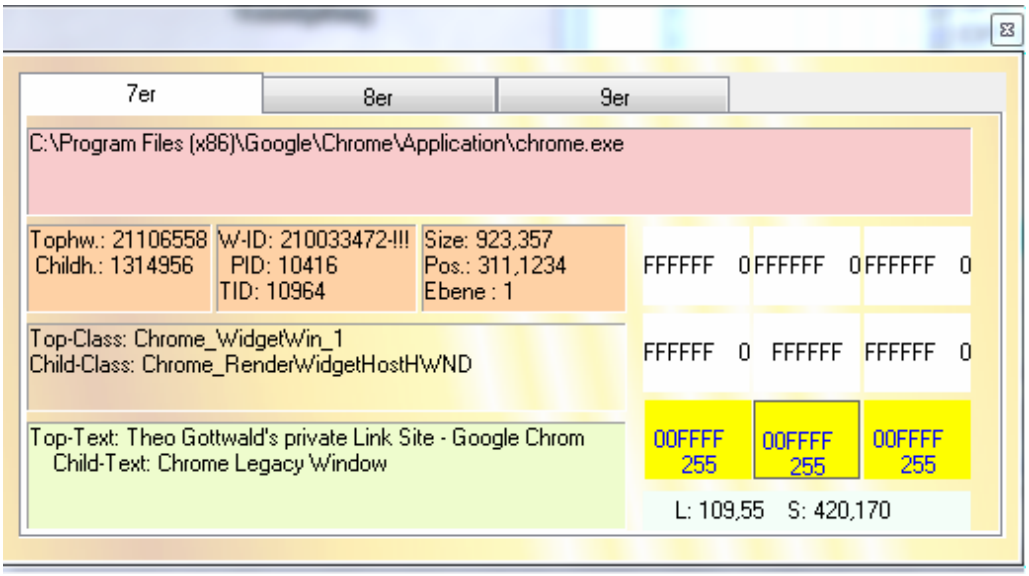
- Instead of "real colors" you can use the "System Colors" S1-S30. They are calculated on each system to their color-value. This way your color values will be independent from themes and settings. See here 3.4 System-Colors 349
- Instead of coordinates prefer using RECTs `{A1}` to `{D4}`.
  For Details see 3.6 Using Quadrant-Coordinates 356

Important:
All Pixel-Operations are "Compatible Engine" Operations.

*© 2023 Theo Gottwald * https://www.smart-package.com*

For Details see here:
1.C Direct Engine and Compatible Engine 231

Often with Pixel-Operations, you need to see single Pixels.
You can use the "Catseye-Analyzer" to find coordinates and color values.
The Catseye-Analyzer is located in the Installation Folder of the SPR.



Important note on Color-Depth.
In the actual Computers, we use 24 bit colors and this is what the Robot uses for its Pixel-commands. For Speed Reasons, no other Color-Depth is supported.
Using Pixel-Commands on other Color-Depth then 24 bit will lead to undefined Results.

Color-Values:
Color Values are decimal, integer numbers in the Range from 0 to 16777215.
In Hexadecimal Notation this is 0 to `&H FF FF FF`.

If you write the Color-Number in Hexadecimal Notation, you can see the Red-, Green-, and Blue-Part of the specified Color side-by-side.
The decimal notation does not have this feature, that is why we recommend that you use the hexadecimal numbers.
Technically there is no difference which notation you use, because the system will anyway convert it to Binary before execution.

Example using Google Chrome-Browser. In this Example, we have a Search WEB-Site with a background color of `&HF6F6F6`
Using this simple Script, we can locate the Input Box and move the Mouse there..

```
' https://duckduckgo.com/
STW.ct|Chrome_WidgetWin_1|DuckDuckGo Search Engine - Google Chrome
IRC.c|!&HF6F6F6
  VAR.$$TAR=$$000,$$000
  MMV.$$TAR
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

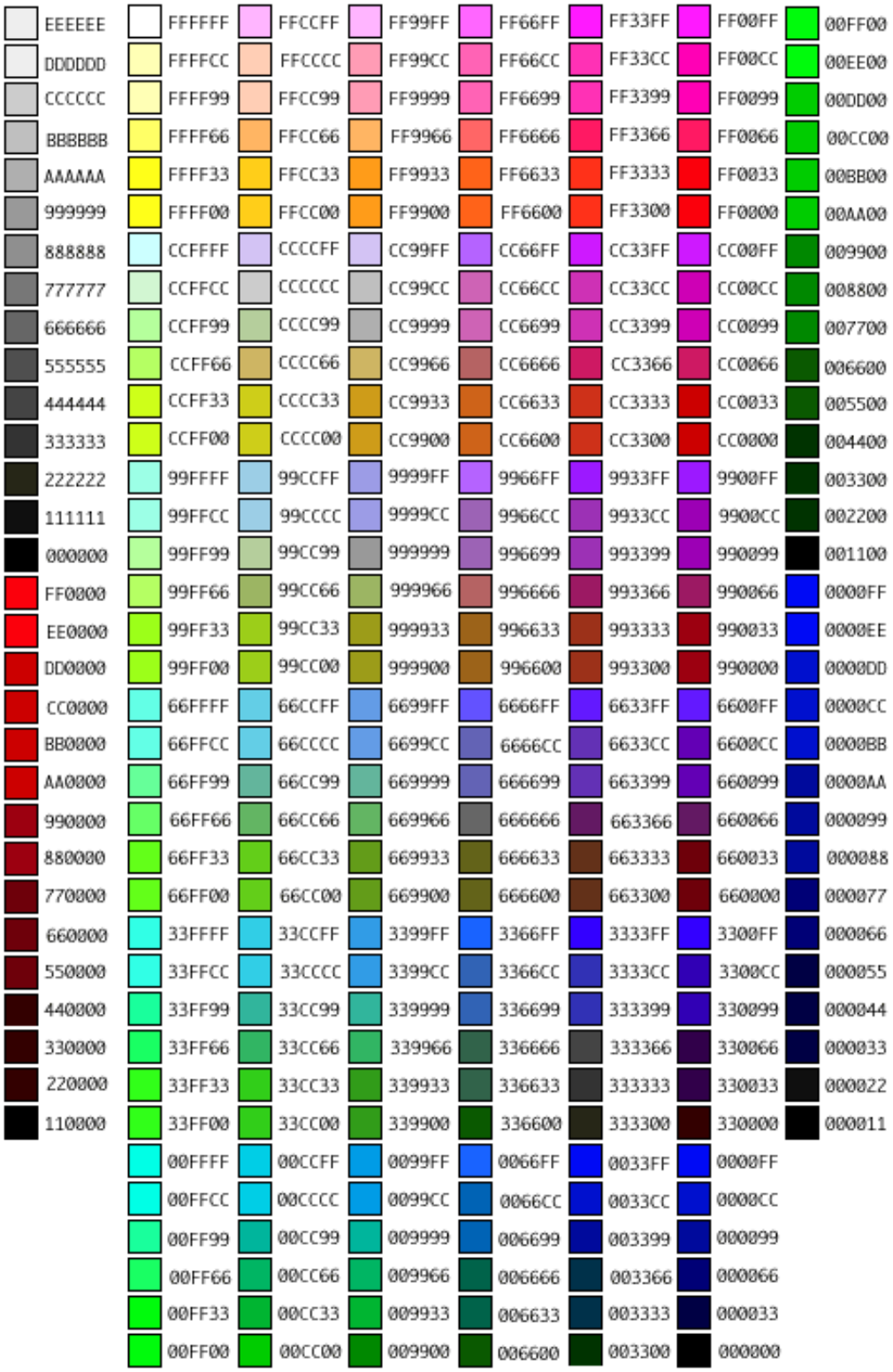
```
ENR.
```

Another Example using Google Chrome-Browser. In this Example, we have a Search WEB-Site with a background color of `&HF6F6F6`
Using this simple Script, we can locate the Input Box and move the Mouse there..Note the DMP.6 that will show the result of the IRC.-Command.

```
STW.ct|Chrome_WidgetWin_1|DuckDuckGo Search Engine - Google Chrome
IRC.c|&HFFFFFF
  DMP.6
  MBX.!
  VAR.$$TAR=$$000,$$000
  MMV.$$TAR
EIF.
ENR.
```

Here are some Color-Values in Hexadecimal notation.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

3.55.2 ! Smarty's Pixel-Processing

Smarty's Pixel Pixel Processing Operations

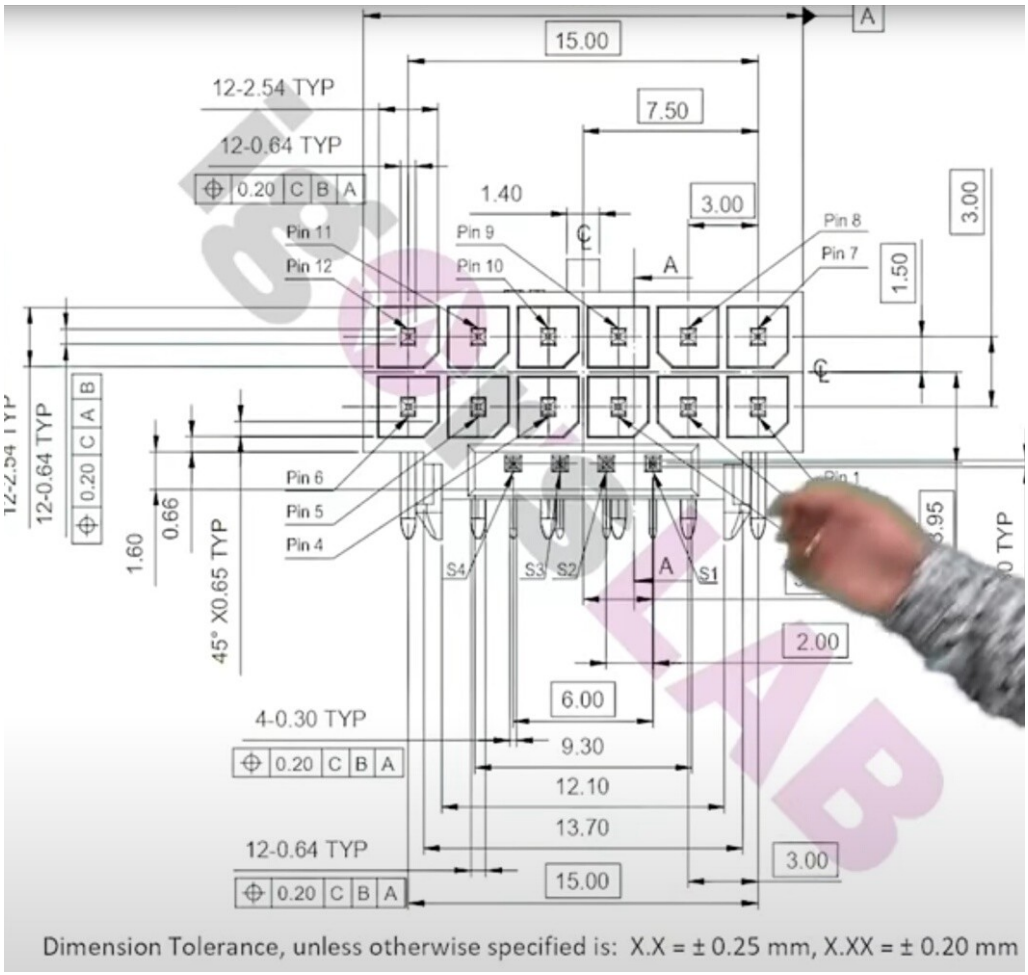
Pixel-Processing Operations are often not really fast but can save you a lot of work.
Using the Getpixel and SetPixel, you can do amazing Pixel-Operations.

Important:
Due to Speed Optimizations, the SPR is unable to STOP immediately while he is in optimzied Pixel-Processing Operations.
Therefore the SPR will not stop unless he is done with the Command. In such case you can just use the Cleanup-Button to "Kill all Robots".

Use the SPR to remove unwanted Artefacts

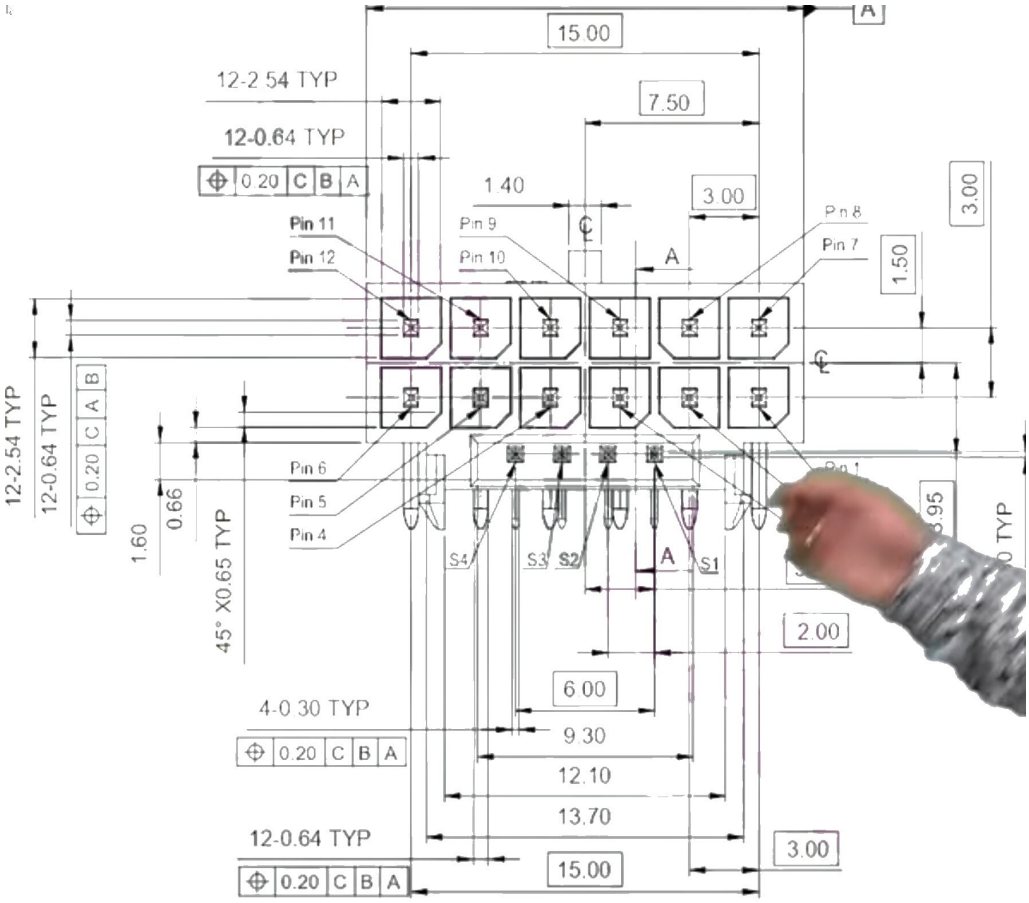
Here is a simple Example that can be adjusted for multiple purposes.
It will remove unwanted Artefacts - for example here a watermark - from a picture.

*© 2023 Theo Gottwald * https://www.smart-package.com*



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

And here is the Script:

Version 1: "Pixel by Pixel".
The structure of this Script is easy to understand and it can be modified for any case.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$RED>&D0)+($$BLU>&HD0)+($$GRN>&HD0)
    IVV.$$ERG>2
      'ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      $$PIX=&HFFFFFF
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
    CAL.$$ERG=($$RED>&HC0)+($$RED<&HF0)+($$GRN>&HA9)+($$GRN<&HE0)+($$BLU>&HB9)+
    IVV.$$ERG>5
      $$PIX=&HFFFFFF
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEX.
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

Depending on the number of Pixels, the Script will take quite some time (20 Minutes for a picture of size 1900x1700) due to the use of not optimized commands.
Therefore this method is perfectly flexible for any automated picture processing need.

Version using ICC.

You can do the same process using the specialized command ICC.:this way its twice as fast. The Script below uses only 8 Minutes compared to the Script above that uses 20 Minutes for a 1900x1700 (3 MP) Picture Process.
Note that using the "0" in the RMax-Field will set R/G/BMax to &HFF.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ICC.0|$$XPO|$$YPO|&HD0|0|&HD0|0|&HD0|0
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
  EIF.
  ICC.0|$$XPO|$$YPO|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9
  ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
EIF.
NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

There is another command that can be used to speed up this process that is RIC..
Using this command, will shorten the code more and make it even faster.
While the result is the same, this Script will only take 6 Minutes to process the same picture.

Version using RIC.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    RIC.0|$$XPO|$$YPO|&HD0|0|&HD0|0|&HD0|0|$$PIX
    RIC.0|$$XPO|$$YPO|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

Optimized Version using RCC.

There is another command that can be used to speed up this process that is RCC..
As RCC. will internally process the complete Picture, it will do the Job in 1-2 Seconds.

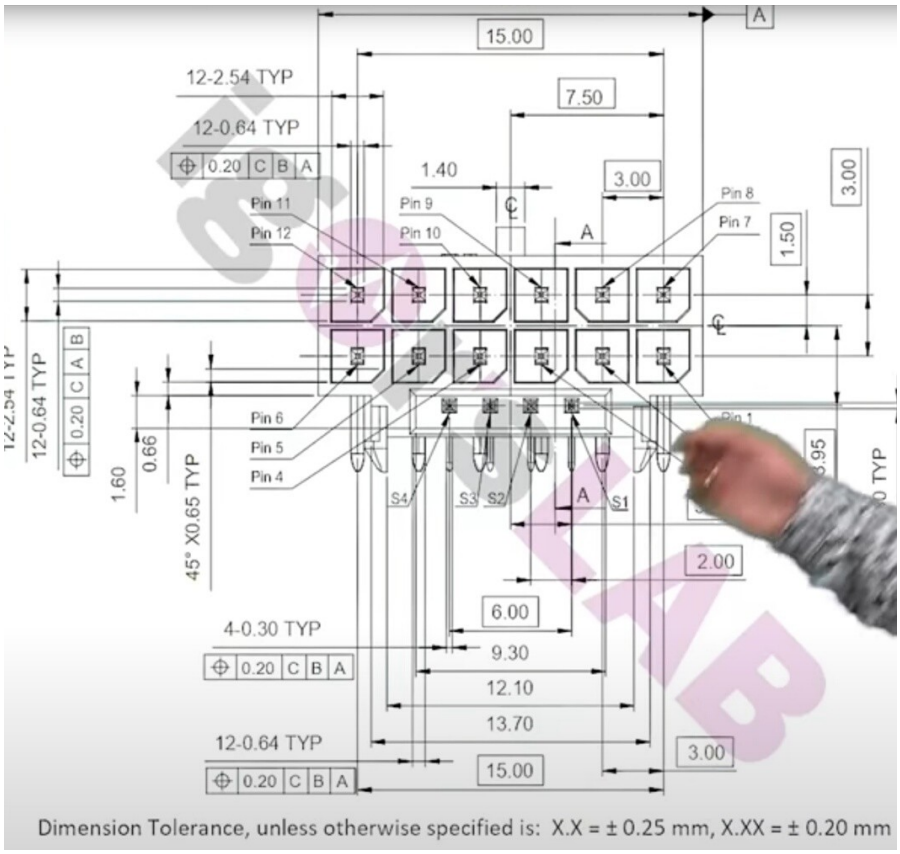
```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
RCC.0|&HD0|0|&HD0|0|&HD0|0|$$PIX
RCC.0|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
CAL.$$TIM=#time#-$$TIM
DMP.1
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

Alternatively you can choose this Setup.

Quickly remove a watermark from a b/w picture.

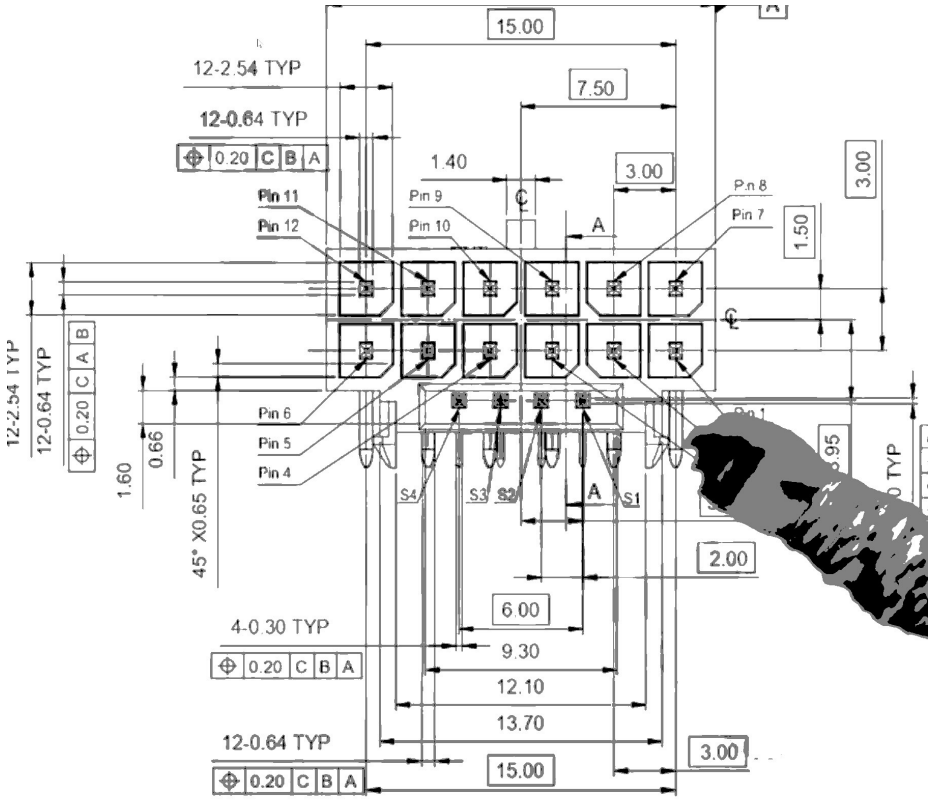
```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.TriColor|0|&H90FFFF
ANA.bw|0|760
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



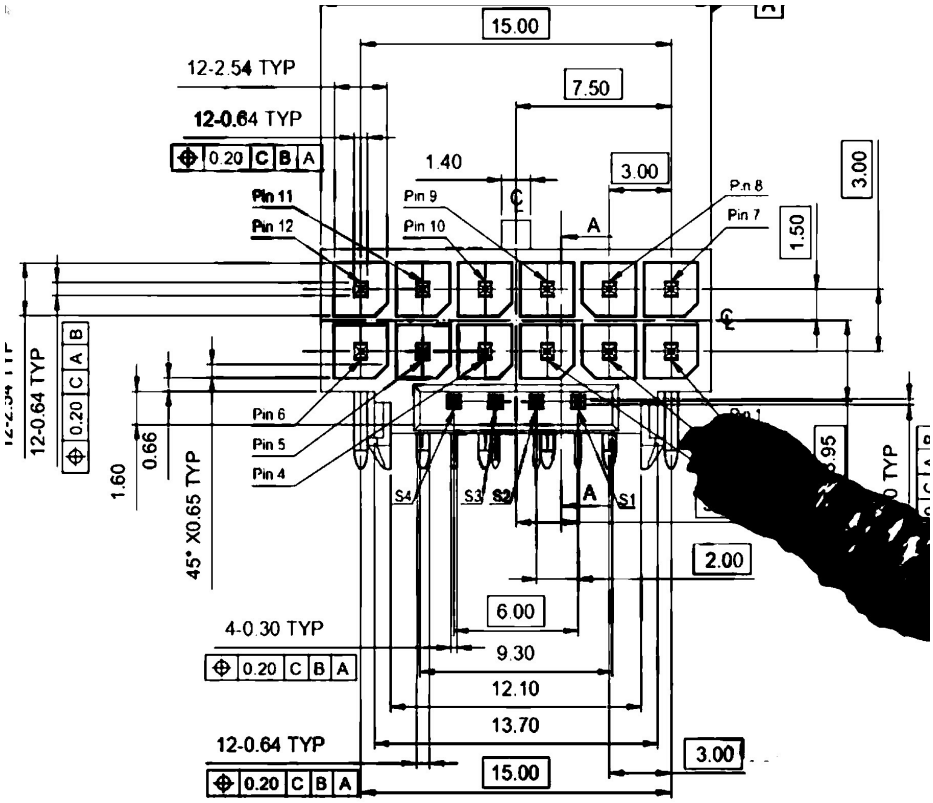
->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

Turning a b/w Picture into a 3D-Logo

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Turning a b/w picture into a 3D-Logo
$$PAB=?exeloc\Pics\P3.jpg

$$BOR=4
CAL.$$BOB=$$BOR/2

IEF.$$PAB
  ANA.Load|1|$$PAB
  'ANA.Show|1!
ELS.
  MBX.File does not exist:$crlf$$$PAT
EIF.

ANA.GetRes|1|$$XPO|$$YPO
CAL.$$XPA=$$XPO+$$BOR
CAL.$$YPA=$$YPO+$$BOR

' Make a Resized Version of P3
ANA.New|2|$$XPA|$$YPA|1
ANA.New|3|$$XPA|$$YPA

' Now mix Shiftzed and Resized Version
ANA.bsh|2|1|$$BOB|$$BOB|1
ANA.Invert|1
ANA.bw|2|600
ANA.Show|1!
ANA.Show|2!
ANA.Logic|3|1|2|A XOR B|RGB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Show|3!
ANA.Invert|3
ANA.Show|3!
ENR.

' Changing the first two lines like this, will produce the second picture:
$$BOR=2
CAL.$$BOB=$$BOR
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3 Additional Commands

Here you can find some more useful commands, that mostly do not use the newer technologies.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3.1 CAP. - Capture Screenshot

MiniRobotLanguage (MRL)

CAP. Command

Capture Screenshot

Intention

This command will capture a Screenshot from a specified part of a window or the complete Computer-Screen.

1. Capture Screenshot of complete Desktop into Clipboard

This will include all Screens on a Dual-Screen System.

```
CAP.Source=desktop;Target=clipboard
ITO.
  DBP.Desktop Capture failed!
EIF.
ENR.
```

2. Capture Screenshot of complete Desktop into a file

This will include all Screens on a Dual-Screen System. They will be saved as a single ".bmp" (BItmap-) File. You can currently not choose any other picture-format then "-bmp".

```
VAF.$$FIL=?desktop\Screenshot.bmp
CAP.Source=desktop;Target=file|$$FIL
ITO.
  DBP.Desktop Capture failed!
EIF.
ENR.
```

3. Capture Screenshot of complete Desktop into a variable

This will include all Screens on a Dual-Screen System. They will be saved as a single Screenshot into the given Variable.

```
CAP.Source=desktop;Target=var|$$VAR
ITO.
  DBP.Desktop Capture failed!
EIF.
LEN.$$VAR|$$RES
DBP.Size of Picture is $$RES Bytes.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

4. Capture Screenshot of actual window into Clipboard/file/variable

If you just locate any window, or child window, you can take a screenshot into the System-Clipboard.

```
STW.ct|#32770|Basic-NAV Matrix 02
CAP.
ENR.
```

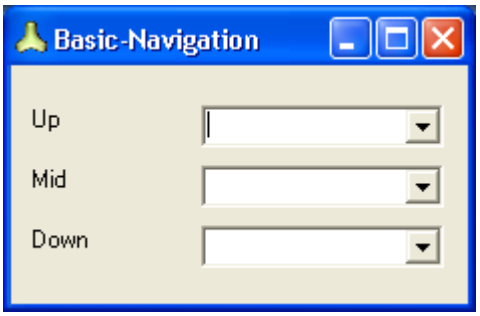
By using the Prototype "`Target=file`" or "`Target=var`" you can direct the output wherever you want it to go.

Capturing a window, you will see that only the so called "Client Area" is captured. That is the Area where the program draws its controls. The Title-bar and the window border (that is the so called "Non Client Area") is been clipped.

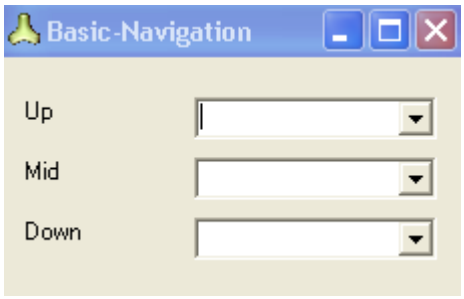
5. Capture Screenshot of actual window with Frame/Non Client Part

To capture the complete window, that includes the "Non-Client" Part of the window, use the "Frame=" Prototype. You can choose between

`Frame=Full`



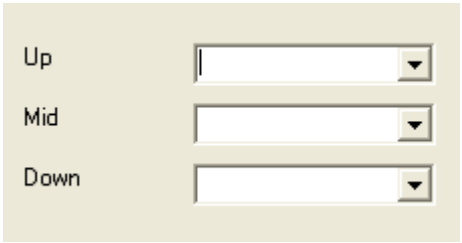
`Frame=Title`



Like this:

```
STW.ct|#32770|Basic-NAV Matrix 02
CAP.frame=full
ENR.
```

If you omit the "`Frame=`" or if you specify "`Frame=no border`", then you get the Snapshoot of just the Client-Area, like this:



`Frame=no Border` *(default)*

6. Capture Screenshot using Screen-Coordinates

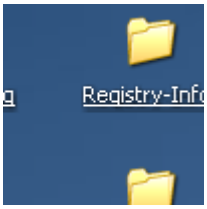
*© 2023 Theo Gottwald * https://www.smart-package.com*

To capture a screenshot using Screen-Coordinates, you can specify this:

```
CAP.Source=Screen-Coords|100,150,200,250
```

or something like this:

```
$$FIL=?desktop\Screenshot.bmp
CAP.Source=Screen-Coords;Target=File|100,150,200,250|$$FIL
```



Then you will get a Screenshot from these screen-coordinates. Screen-Coordinates should be avoided in most places. Prefer using Window-Coordinates instead.

7. Capture Screenshot using Window-Coordinates

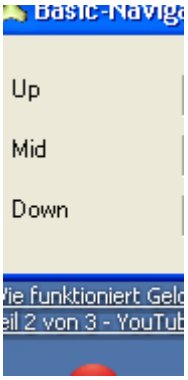
To capture a screenshot using Window-Coordinates, you can specify this:

```
STW.ct|Form1_Class|Basic-Navigation
CAP.Source=Window-Coords|100,150,200,250
```

or if you want to use a file:

```
$$FIL=?desktop\Screenshot.bmp
STW.ct|Form1_Class|Basic-Navigation
CAP.Source=Window-Coords;Target=File|10,15,100,200|$$FIL
```

Note that internally the Window-Coordinates are been converted to Screen-Coordinates, and the screenshot is taken from the desktop. Therefore, if the given Area is larger then the window, you will see what was behind in the screenshot.
Here is the picture from the example above:

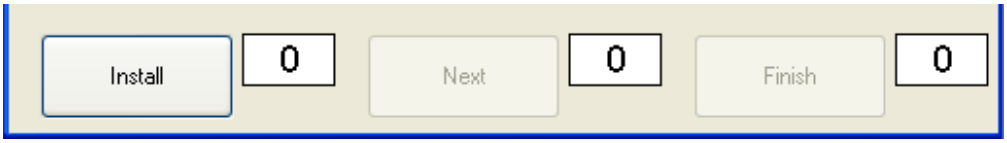


8. Capture Screenshot specifying a window-handle

To capture a screenshot using a resolution independent, yet defined part of a window, you can use Quadrant coordinates (`A1-D4`) like this:

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Will place the scrrenshot in the Clipboard
STW.ct|PBWindowClass:0|BCS - Testform
CAP.Source=wc|{D1,D4}
ENR.
```



You can combine these Prototypes, to snapshot to a file or to a variable.However any "Frame=" Prototype will be ignored in this case. Instead the Full window is taken as base-area.

Syntax

CAP.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Prototype-String in readable format can contain this values:

Source=

- handle - use a window handle in **P2.** Use **P3** for Target.
- window - will use actual child- or topwindow or selected object. **P2** is for target,
- desktop - will make a snapshoot of the complete desktop. **P2** is for target,
- screen-coords - You can give screen-coordinates in **P2**. Use **P3** for Target.
- window-coords - You can specify window-coordinates or Quadrants in **P2**. Use **P3** for Target.

Frame=

- no Frame
- Full
- Title

Target=

- Clipboard - You can paste the content of the Clipboard directly into your favorite graphics or office program.
- File - you need to specify a complete file-path and name in the last (Target-) Parameter.
- Variable - Use this, if you need the binary data from the screenshot in a variable.
  For example to do yourself a color-analysis.

Putting a Screenshot inside a Variable will produce a Variable with this format insider the Variable:

- 8 Byte (4 Byte X- and 4 Byte Y-Size Information)
- XSize * YSize * 4 Byte Color-Information (4 Byte per Pixel)

© 2023 Theo Gottwald * https://www.smart-package.com

This format is not compatible with the RRA.-Bitmaps, as these are prefixed with 8 Bytes of Origin-Positional Information.

```
Depending on the given Text-Prototypes, the other parameters
may or may not be needed.
```

P2 – (optional) - In case that a handle, or any sort of coordinates need to be specified, these will be in **P2**. Any additional Target-Parameter like Filename or Variable will then be in **P3**. If P2 is not needed for the Source, the P2 will hold the Destination Parameter and P3 is not needed. In case you use "**Source=window**" then you can specify resolution independent Quadrant-Coordinates 356 at this place.

```
CAP.Source=wc|{D2,D4}
```

Doing so will however make the robot ignore any given "Frame=" Prototype. In this case the "Frame=Full" will automatically be used.

P3 - (optional) – Holds the Target variable, if **P2** is needed to specify the Source (Coordinates, handle). Then **P3** will hold the Destination value (Variable-Name, File-Name). In case the Source does not need any specifications, then **P2** will hold the Target-Information and **P3** must be omitted.

Example

```
'**********************************
' CAP.-Sample
'**********************************
CAP.Source=desktop;Target=var|$$VAR
ITO.
  DBP.Desktop Capture failed!
EIF.
LEN.$$VAR|$$RES
DBP.Size of Picture is $$RES Bytes.
ENR.
```

Remarks

Putting a Screenshot inside a Variable will produce a Variable with this format insider the Variable:
- 8 Byte (4 Byte X- and 4 Byte Y-Size Information)
- XSize * YSize * 4 Byte Color-Information (4 Byte per Pixel)

This format is not compatible with the RRA.-Bitmaps, as these are prefixed with 8 Bytes of Origin-Positional Information.

You can display the saved bitmaps, using the UCV.-command.

```
$$FIL=?desktop\Screenshot.bmp
STW.ct|Form1_Class|Basic-Navigation
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CAP.Source=Window-Coords;Target=File|10,15,100,200|$$FIL
UCV.ldi|$$FIL
PAU.5
```

Limitations:

If Windows is unable to reserve the Memory for the Screenshot-Operation, the operation will fail and the timeout-flag will be set. In this case you can try again with a freshly booted system. It may also help, if you just screenshot a smaller area instead of the whole desktop.
In case the Screenshot command fails (for low system resources) you will have the timeout-flag set. Otherwise it will be cleared.

```
CAP.
ITO.
  DBP.Desktop Capture failed!
ELS.
  DBP. Capture was successful.
EIF.
```

See also:

- 3.6 Using Quadrant-Coordinates 356
- ! UCV. - Undercover Graphics 1914
- ldi - load image 1958
- CAL. - mathematical CALculation 2315
- IDP. / NDP. - If desktop dpi Setting 1126
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3.2 CCD. - Calculate Color Distance

MiniRobotLanguage (MRL)

CCD. Command

Calculate Color Distance

Intention

This command will tell you, how similar two colors are. For this the command calculates the "Color-Distance" between these two given colors.
There are several Modes of operation for the `CCD.`-Command, all these may give different results. You may use 3.4 System-Colors 349 as parameter with this command.

1. Color Distance on TOS

```
' White Color
$$COA=&HFFFFFF
' Black Color
$$COB=&H000000
CCD.$$COA|$$COB
DBP.$$000
ENR.
```

`Result is:` **`441673`** `which is the` **`maximum Color Distance`** `in this mode.`

2. Color Distance to Variable.

This is exactly the same, just that we put the result into a variable instead of using the stack.

```
$$COA=&HFFFFFF
$$COB=&H000000
CCD.$$COA|$$COB|$$COD
DBP.$$COD
ENR.
```

3. Color Distance in IEE-FP-Format

This is exactly the same, just that we put the result has a maximum precision.

```
$$COA=&HFFFFFF
$$COB=&H000000
CCD.$$COA|$$COB|$$COD|F
DBP.$$COD
ENR.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
Result is: 441.67295593006371

Here are some other results:

&HFFFFFF/&HFFFFFE = 1
&HFFFFFF/&HFFFEFE = 1.41421356237309505
```

This way you can really get the most accurate color distance between two colors.

4. Color Distance as percentage from 0% to 100%.

This is exactly the same, just that we get the result as a percentage from 0% to 100%.
Fractions will not be calculated, as they will also not be used in places where such a color tolerance can be specified.

```
$$COA=&HFFFFFF
$$COA=&HFFFFFF
$$COB=&H000000
CCD.$$COA|$$COB|$$COD|P
DBP.$$COD
ENR.

Result is: 100
```

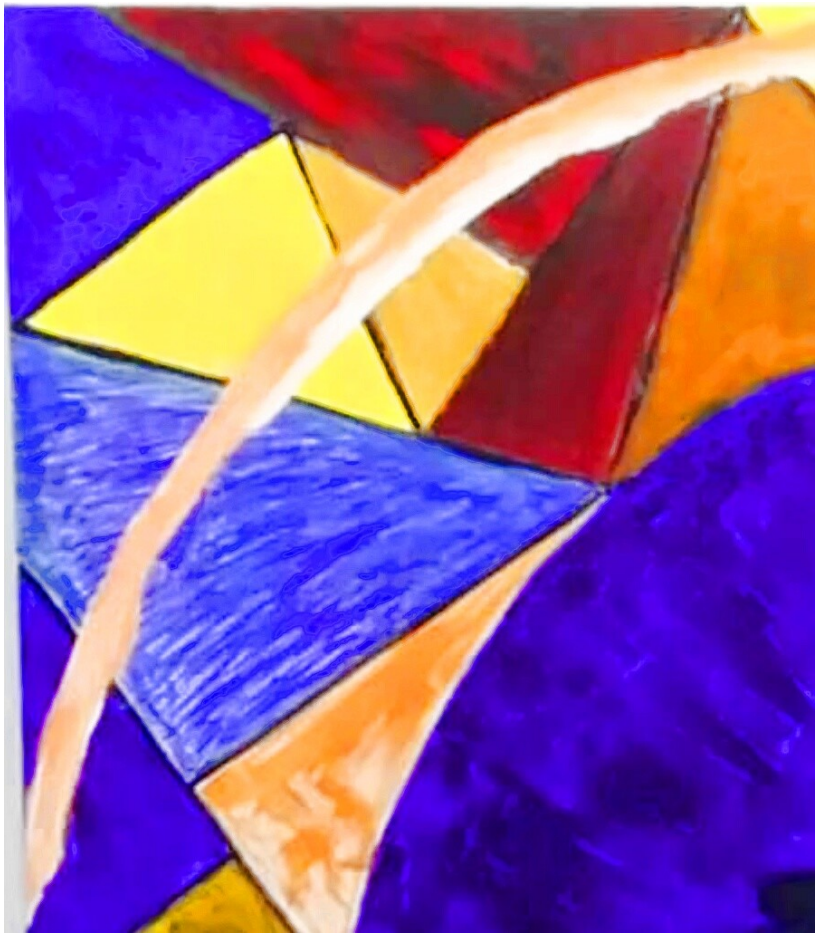
1. Extract Object from known Background using CCD. as a Tolerance-Factor
Using compressed Formats you will need to apply a Tolerance.
We can do this, using the CCD.-Command.
Here we use a Color-Distance of 5%, you can see that sharp borders make trouble.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
VAF.$$PIC=?pattern\SarahR_640.jpg
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
    ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
    CCD.$$PCA|$$PCB|$$PCR|P
    NVV.$$PCR<5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
    ELS.
    ' Paint it Yellow
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
    EIF.
  NEX.
NEX.
ANA.Save|0|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

Here we use a known Background to extract Sarah from the second Picture into the third Picture with yellow background.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Because we used a ".jpg"-File we still have some artefacts when using a Color-Distance Value of only 5%. In the second Picture we use a value of 10%. This will remove some problems and produce some other.
Thats why you should not use compressed Images for this sort of operations.

```
013 0000000000000251 0001 VAN.$$TIM=#time#
014 0000000000000101 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000062 0001 VAF.$$PIB=?pattern\Mahakala2.png
016 0000000000000061 0001 VAF.$$PIC=?pattern\SarahR_640.jpg
017 0000000000456466 0001 ANA.Load|0|$$PIA
018 0000000001862915 0001 ANA.Load|1|$$PIB
019 0000000000000784 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000000137 0963 FOR.$$YPO|1|$$YSZ
021 0000000000030412 0962 DBP.$$YPO/$$YSZ
022 0000000000000164 616642 FOR.$$XPO|1|$$XSZ/2
023 0000000000000341 615680 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
024 0000000000000350 615680 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
025 0000000000000212 615680 CCD.$$PCA|$$PCB|$$PCR|P
026 0000000000000096 615680 NVV.$$PCR<5
027 0000000000000528 204271 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
028 0000000000000018 615680 ELS.
029 0000000000000003 411409
030 0000000000000290 411409 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
031 0000000000000004 615680 EIF.
032 0000000000000006 616642 NEX.
033 0000000000000006 0963 NEX.
034 0000000000375542 0001 ANA.Save|0|$$PIC
035 0000000000000475 0001 CAL.$$TIM=#time#-$$TIM
036 0000000000137155 0001 DBP.Used Time in Seconds: $$TIM
037 0000000000000000 0001 DMP.1
```

Syntax

CCD.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Color A to calculate the distance from.
You can use 3.4 System-Colors 349 values like S1,S2 etc.

P2 - Color B to calculate the distance to
You can use 3.4 System-Colors 349 values like S1,S2 etc.

P3 - (optional) - Variable for the result. If omitted, TOS is used.

P4 - (optional) - Can be omitted then a integer result is delivered,
which is in the range from 0 to 441673.
P - Calculates a percentage result between 0 and 100
F - Calculates a Extended precision result. This is between 0 and 441.67295593006371

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' CCD.-Sample
'**********************************
'
$$COA=&HFFFFFF
$$COB=&H0
CCD.$$COA|$$COB|$$COD|P
DBP.$$COD
ENR.
```

Remarks

CCD. is a speed-optimized Operation that will only use 180 to 200 Ticks.

Limitations:

-

See also:

- CAL. - mathematical CALculation 2315
- IDP. / NDP. - If desktop dpi Setting 1126
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801
- ! UCV. - Undercover Graphics 1914

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3.3 CCV. - Color Conversion

MiniRobotLanguage (MRL)

CCV. Command

Convert and split RGB/BGR Color

Intention

This command will help you to handle the Red,Green and Blue Values of 24/32 bit Colors.
There are several Modes of operation for the `CCV.`-Command.

Using the "H" and "K"-Modes, you can use the command to convert a decimal number to Hexadezimal (HEX) Color values.

Using other Modes (X,Y,U and V) you can use the command to generate random numbers and colors.

1. Convert RGB-Color to BGR Color
```
CCV.BGR|$$RGB|$$BGR
```

2. Convert BGR-Color to RGB Color
```
CCV.RGB|$$BGB|$$RGR
```

```
3.Convert Color value into Hexadezimal
CCV.H|$$COL|$$HEX

4.Swap R/B Color value and convert into Hexadezimal
CCV.K|$$COL|$$HEX
```

Here is a Sample script, using both features:

```
$$RGB=&H00FF10
CCV.BGR|$$RGB|$$BGR
CCV.H|$$BGR|$$RES
DBP.$$RES
ENR.
' Result is: &H10FF00
```

Also you can use the CCV. command to mix the RGB-parts into one color value.

5. Convert RGB-Color parts to BGR Color
```
CCV.BGR|$$RED,$$GRN,$$BLU|$$BGR
```

6. Convert BGR-Color parts to RGB Color
```
CCV.RGB|$$RED,$$GRN,$$BLU|$$RGR
```

Here is a Sample script, using this features:

```
$$RED=&H00
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$GRN=&HFF
$$BLU=&H10
CCV.BGR|$$RED,$$GRN,$$BLU|$$BGR
CCV.K|$$BGR|$$RES
DBP.$$RES
ENR.
' Result is: &H10FF00
```

Third you can use the `CCV.` command to split a color value into its R,G and B part.

```
7.Split Color value into RGB values
CCV.Split|$$COL|$$RED,$$GRN,$$BLU

8.Split Color value into BGR values
CCV.T|$$COL|$$RED,$$GRN,$$BLU
```

Here is a Sample script, using this features:

```
$$RGB=&H00FF10
CCV.T|$$RGB|$$RED,$$GRN,$$BLU
$$RES=$$RED-$$GRN-$$BLU
DBP.$$RES
ENR.
' Result is: 0-255-16
```

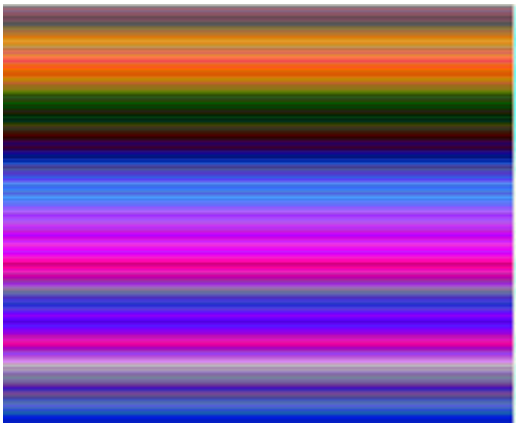
Fourth `CCV.` can be used to generate Random colors. There are are two random-number algorithms. The results are best described with these two pictures.

Algo 1 will use total random colors that are in no relation to the previous color.

Here is thesample program for algo 1:

```
UCV.nbi
UCV.cov
FOR.$$Y00|1|500
  CCV.x|$$RED,$$GRN,$$BLU
  CCV.r|$$RED,$$GRN,$$BLU|$$RGB
  UCV.fgr|$$RGB
  UCV.Line|0,$$Y00|500,$$Y00
NEX.
PAU.3
ENR.
```

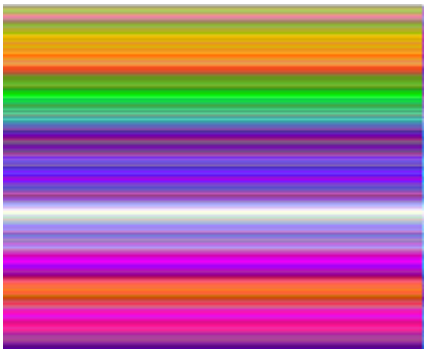

© 2023 Theo Gottwald * https://www.smart-package.com



Algo 2 will generate random numbers that are near the previous random number by the given additional value. Here a value of 25 was used.

Here is the sample program for algo 2, change the value of `$$VAL` to see the differences:

```
'#SPI:forcewrite
$$VAL=25
UCV.nbi
UCV.cov
FOR.$$Y00|1|500
 ' CCV.x|$$RED,$$GRN,$$BLU
  CCV.y|$$VAL|$$RED,$$GRN,$$BLU
  CCV.r|$$RED,$$GRN,$$BLU|$$RGB
  UCV.fgr|$$RGB
  UCV.Line|0,$$Y00|500,$$Y00
NEX.
PAU.3
ENR.
```



Here a value of 35 was used.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

CCV.[P1][|P2][,P3]

Parameter Explanation

P1 - prototypes

```
 RGB - convert input to RGB
     CCV.RGB|$$RED,$$GRN,$$BLU|$$COL
     CCV.RGB|$$INP|$$COL
     You can use 3.4 System-Colors 349 values like S1,S2 etc.

 BGR - convert input to BGR
     CCV.BGR|$$RED,$$GRN,$$BLU|$$COL
     CCV.BGR|$$INP|$$COL
     You can use 3.4 System-Colors 349 values like S1,S2 etc.

 Split - Split Input into RGB-values
     CCV.Split|$$COL|$$RED,$$GRN,$$BLU

 T - Split into RGB and Swap R and B value
     CCV.Split|$$COL|$$RED,$$GRN,$$BLU

 H - Convert number to Hexadezimal
     CCV.H|$$NUM|$$RES
     You can use this to see the values of 3.4 System-Colors 349
     like S1,S2 etc. Example:
     ' Show value od System color S2
     CCV.H|S2|$$HEX
     DBP.$$HEX

 K - Swap R and B value and Convert number to Hexadezimal
     CCV.K|$$NUM|$$RES

 X - Generate random color (algo 1)
     CCV.X|$$RED,$$GRN,$$BLU

 Y - Generate random color (algo 2 using $$VAL)
     CCV.Y|$$VAL|$$RED,$$GRN,$$BLU

 U - Generate random color (algo 1)
     CCV.U|$$COL

 V - Generate random color (algo 2 using $$VAL)
     CCV.U|$$VAL|$$COL
```

The parameters P2 and P3 depend on the mode of operation.

Example

```
'***********************************
' CCV.-Sample
'***********************************
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CCV.RGB|&HFF,&H00,&HFF|$$RES
DBP.$$RES
CCV.S|16711935|$$RED,$$GRN,$$BLU
DBP.$$RED,$$GRN,$$BLU
ENR.

'**********************************
' CCV.-Sample
'**********************************
'
$$RGB=16711939
CCV.RGB|$$RGB|$$BGR
DBP.$$BGR
ENR.

'**********************************
' CCV.-Sample
'**********************************
'
'#SPI:forcewrite
$$VAL=35
UCV.nbi
UCV.cov
FOR.$$Y00|1|500
 ' CCV.x|$$RED,$$GRN,$$BLU
  CCV.y|$$VAL|$$RED,$$GRN,$$BLU
  CCV.r|$$RED,$$GRN,$$BLU|$$RGB
  UCV.fgr|$$RGB
  UCV.Line|0,$$Y00|500,$$Y00
NEX.
PAU.3
ENR.
```

Remarks

When specifying Colors as hexadecimal values, prefer using this format:
`&H0FFFFFF.`
`RRGGBB`
If you omit the "`&H0`" or just use the "`&H`" this may lead to strange results when the values is converted to decimal. The reason for this is::
`&HFFFF is -1 (conversion to signed number using "&H")`
while
`&H0FFFF is 65535 (conversion to unsigned number using "&H0")`
as expected.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Pixel-Operations 1088
- ! UCV. - Undercover Graphics 1914
- IDP. / NDP. - If desktop dpi Setting 1126
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801
- ! UCV. - Undercover Graphics 1914

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3.4 DPI. - Get Desktop DPI-Settings

MiniRobotLanguage (MRL)

DPI. Command

Get Desktop Dots per Inch

Intention

`DPI.` will give you the

- `DPI.` X and Y values of the desktop
- `DPI.` X and Y-Ratio

Usage is simple. Using DPI. without any parameters will place the `dpi-X` and the `dpi-Y` values on Top of Stack.

```
DPI.
' Show TOS
DBP.$$000
ENR.
```

To get the X- and Y- Scaling Ratios, just add an **S**-prototype.

```
DPI.s
DBP.$tos$
ENR.
```

Now you can add prototypesand variables to get the results into variables instead into the TOS. Examples:

```
DPI.$$XDP,$$YDP
DBP.$$XDP,$$YDP
ENR.
```

and

```
DPI.s|$$XDP,$$YDP
DBP.$$XDP,$$YDP
ENR.
```

If you specify only one variable, you will get both values, separated with a comma into that variable. Examples:

```
DPI.d|$$XDP
DBP.$$XDP
ENR.

DPI.s|$$XDP
DBP.$$XDP
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

There is a special mode for `DPI.`:

```
DPI.$$XDP
DBP.$$XDP
ENR.
```

This way you can get just the x-dpi-value into a variable.As both values are often equal, it may be enough for most scripts.

Hint:
If you just want to check if the Target PC has a specific DPI (dots per Inch) -Setting, prefer the `IDP./NDP.` command.

Syntax

DPI.[P1][|P2][,P3]

Parameter Explanation

P1 **-** (optional) prototype

Prototype can be omitted or any of:

```
 d - get Screen dpi (default, can be omitted)
 s - get screen dpi-ratios (scaling)
 x - get x-dpi value
 y - get y-dpi value
```

If the **s**-prototype is omitted, **dpi**-values are returned, otherwise scaling-ratio values. The sequential order of the prototypes does not play a role in this command. The first variable will always hold the x- and the second variable may only hold the y--results.

P2 **-** (optional) - variable to store the result, depending on the prototypes

P3 **-** (optional) - variable to store the result, depending on the prototypes

Speed in Ticks:

This command uses typically around 1900 Ticks.

Example

```
'**********************************
' DPI.-Sample
'**********************************
DPI.$$XDP
DBP.$$XDP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.8. Selecting the Application (Topwindow) 150
- IDP. / NDP. - If desktop dpi Setting 1126
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3.5 IDP. / NDP. - If desktop dpi Setting

MiniRobotLanguage (MRL)

IDP. Conditional Statement

If Desktop Dots per Inch

Intention

Conditional Statement. Test if the desktop dpi-Settings are equal to your specified values. You can test the

- `DPI.` X and Y values of the desktop
- `DPI.` X and Y-scaling Ratio of the desktop

against values that you specify. The condition will evaluate TRUE if the Desktop DPI-values match those values that you have specified.

Usage is simple. Using `IDP.` without any parameters will test the `dpi-X` and the `dpi-Y` values against the default value of 96 dpi (WinXP). Example:

```
IDP.
  DBP.Yes we have 96 dpi
ELS.
  DBP.No we haven't 96 dpi
EIF.
ENR.
```

The next step would be to specify the dpi values we want to compare with. If you specify just one value, then both the X- and the Y-desktop dpi will be compared to that value.
If any of these does not match the 96 dpi, the script will decide FALSE and continue with `ELS./EIF.`

```
IDP.96
  DBP.Yes we have 96 dpi
ELS.
  DBP.No we haven't 96 dpi
EIF.
ENR.
```

Then you can specify both values, the x- and the y-dpi-value. And test for both. Like this:

```
IDP.96,96
  DBP.Yes we have 96 dpi
ELS.
  DBP.No we haven't 96 dpi
EIF.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Now you can ass additional functionality to your script by adding prototypes. This will only compare the x-dpi settings against the specifies value.

```
IDP.x|96
  DBP.Yes we have 96 dpi
ELS.
  DBP.No we haven't 96 dpi
EIF.
ENR.
```

Also you can add an y-prototype and test both values.

```
IDP.xy|96,96
  DBP.Yes we have 96 dpi
ELS.
  DBP.No we haven't 96 dpi
EIF.
ENR.
```

Then you can specify the "**s**"-prototype to make the script test the ratio of the scaling instead of the dpi-values.

```
IDP.s|1
  DBV.Yes we have a ratio of 1
ELS.
  DBV.No we haven't a ratio of 1
EIF.
```

adding the prototype "**e**" will make the robot compare both the x- and the y- values and only evaluate to TRUE, if both are equal. This way we know that both ratios are 1:

```
IDP.es|1
  DBV.Yes we have a ratio of 1
ELS.
  DBV.No we haven't a ratio of 1
EIF.
ENR.
```

The same is possible with dpi-values also:

```
IDP.e|96
  DBP.Yes we have 96 dpi
ELS.
  DBP.No we haven't 96 dpi
EIF.
ENR.
```

Hint:
If you want to get the X-and the Y-value of the Target PC for calculations, prefer the `DPI.` command.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

IDP.P1[|P2][,P3] … ELS. … EIF.

Parameter Explanation

P1 **-** (optional) prototype

Prototype can be omitted or any of:

```
 d - get Screen dpi (default, can be omitted)
 s - get screen dpi-ratios (scaling)
 x - get x-dpi value
 y - get y-dpi value
```

If the **s**-prototype is omitted, **dpi**-values are used, otherwise scaling-ratio values. The sequential order of the prototypes does not play a role in this command. The first variable will always be compared against the x- and the second variable may be compared against the the y--values.

The usage of **P2** and **P3** depends on the prototypes

P2 **-** (optional) - variable holding the value to compare the actual
dpi-settings of the desktop with.

P3 **-** (optional) - variable holding the value to compare the actual
dpi-settings of the desktop with.

Example

```
'***********************************
' DPI.-Sample
'***********************************
DPI.$$XDP
DBP.$$XDP
ENR.
END.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.6.1. Program Flow Control 121
- DPI. - Get Desktop DPI-Settings 1123
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.3.6 QFR. - Quadrant from RECT

MiniRobotLanguage (MRL)

QFR. Command

Get Quadrant Coordinates from Rect

Intention

The QFR-Command will take any Rectangular Area, and calculate the Coordinates of a Quadrant-Rect inside of this Area.
This can be used to locate graphical Elements in a form, control or any part of a form.

`QFR.` understands Quadrant Coordinates, see 3.6 Using Quadrant-Coordinates 356.
It will take the last localized Object as base for the Quadrants. This way, you can split any type of objects or control in an resolution independent way into 16 Quadrants.

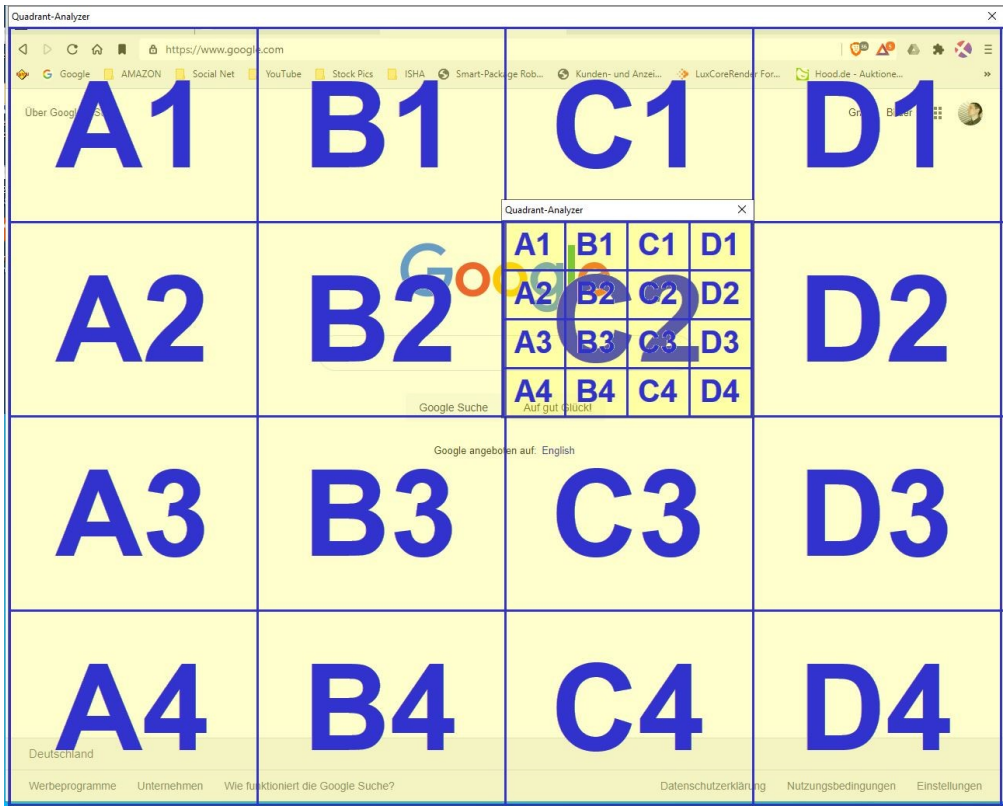
Depending on the Parameters you will normally use a 4x4 Quadrant-System.
Using the optional Parameter **P5**, `QFR.` can handle any Quadrant-System from 2x2 to 9x9.

The command is in a way designed that you can then use the result-from `QFR.` as Input for another `QFR` (nested Quadrants).

What is a Quadrant?
In the following picture you can see the Quadrants of a Topwindow.

What is a Quadrant good for?
Quadrants are resolution-independent. If you can click Buttons using Quadrants, the result will be resolution independent, as the original rectangle (for example a Top-Window) will always be divided into the same amount of Quadrants..
Using the "`o{..}`"-Option, you can split Elements that have been located, for example in a Treeview, or custom Control.
This way you can click parts of controls that can not be subdivided in other ways.

*© 2023 Theo Gottwald * https://www.smart-package.com*



This are the 4x4 Quadrants of a Window. They range from A1 to D4.

How do i Click into a Quadrant?
You can use the `GRM.`-Command to calculate the Middle-Point inside the Quadrant. Here is an example.

```
QFR.|{A1}|$$REC
GRM.$$REC|$$MID
MMV.$$MID
```

Can i combine multiple Quadrants into one?
You can combine multiple Quadrants, note that the resulting Quadrant will always be rectangular.
For example, if you combine `{A1}` and `{B2}` then the resulting Quadrant will automatically also include `{A2}` and `{B1}`.
Because the result needs to be rectangular. Technically you would write:

```
QFR.|{A1,B2}|$$REC
```

To combine these two Quadrants. Note that there is no limit on how many Quadrants you wish to combine.

```
QFR.|{A1,B2,C3,D3}|$$REC
```

Will also work as expected.

Can i combine multiple QFR-Calls into one?
You can combine multiple Calls into one. For this you can just enter the Quadrants in a sequential way.

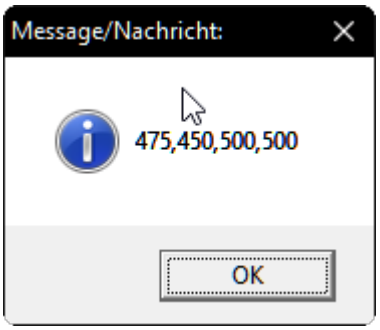
*© 2023 Theo Gottwald * https://www.smart-package.com*

```
QFR.|{A1}{B1}|$$REC
```

Will calculate Quadrant `A1`, and then inside `A1` calculate the Quadrant `B1`.

```
' This is a nested QFR. it will calculate a Quadrant D4 inside
a Quadrant D3,D4.
QFR.100,100,500,500|{D4}{D3,D4}|$$RES
MBX.$$RES

' This is the same like the statement above in two QFR.-Calls
QFR.100,100,500,500|{D4}|$$RES
QFR.$$RES|{D3,D4}|$$RES
MBX.$$RES
```

In both cases this is the result.



How do I specify the Source-Rectangle, from which the Quadrants are calculated?
There are multiple ways to specify them.

1. Variation: Just specify nothing. Doing so, **`QFR.`** will take the last localized Rectangle (window or whatever).
Note that often Windows are moved after they have been localized. In such case this version of `QFR.` may use the "old" coordinates before the window was moved!

```
QFR.|{A1,A2}|$$REC|4
```

2. Variation: Specify "**c**","**w**","**t**" or Combinations of these.
Doing so, **`QFR.`** will take the last localized window "**w**", "**t**"-Top-Window and/or the "**c**"-Client-Coordinates of these windows.
Specifying just "c" will use the Client Coordinate of whatever window that was last localized.
Unlike Variation 1, this Variation will calculate the most actual window coordinates at the time of the Call to `QFR.`.

```
QFR.tc|{A1,A2}|$$REC|4
QFR.wc|{A1,A2}|$$REC|4
QFR.t|{A1,A2}|$$REC|4
QFR.w|{A1,A2}|$$REC|4
QFR.c|{A1,A2}|$$REC|4
```

3. Variation: Specify just the Screen-Coordinates of the Rectangle.

```
QFR.0,0,400,400|{A1,A2}|$$REC|4
' Result 0,0,100,200
```

4. Variation: Specify "**w**","**t**","**c**" and the Screen-Coordinates of the Rectangle.

© 2023 Theo Gottwald * https://www.smart-package.com

```
   ' Now the Coordinates are relative to the last located window.
QFR.w0,0,400,400|{A1,B1}|$$REC|4
```

5. Variation: Specify "**w**","**t**","**c**" and the Quadrant of the last localized window/Top-Window/Client-Area or Object.

```
' Now the Coordinates are relative to the last located
' Top-Window or Window or Client-Area or Object of these.
QFR.o{A1,A2}|{A1}|$$REC|4
QFR.tc{A1}|{A1,B2}|$$REC|4
QFR.w{A1}{A2-A4}|{A1,B2}|$$REC|4
```

More Examples:

```
QFR.0,0,400,400|{A1,A2,B2}|$$REC|4
' Result 0,0,200,200

' Return complete Windows-Rect (like GAR.)
STW.tc|PB|Testap
QFR.|{A1,D4}|$$REC
MBX.$$REC
' Get the Midpoint of that RECT
GRM.$$REC|$$MID
MBX.$$MID
```

`QFR.` can be used to get the Coordinates for other Pixel-Operations like:

- ILC. / NLC. - If Line Color 1805
- IRA. / NRA. - If Rect altered 1824
- RRA. - Remember Rectangular Area 1836
- and all such commands.

Syntax

QFR.[P1][|P2][|P3][|P4]

Parameter Explanation

P1 - (optional) *"`t`","`w`","`c`" or Combinations of these.*

And/or an Input-Rectangle. Format is: `x1,y1,x2,y2` if omitted, it will use the actual window or object rectangle.

Or you can specify Quadrant coordinates like `{A1}`.In such case these Coordinates are Screen-Coordinates unless

you prefix them with "`t`","`w`" or "`c`" or Combiantions of these.

If Parameter is omitted, it will take the last localized Object-rectangle as base for the Quadrants.

Generally the given coordinates stay in the Source-Coordinate-System.

If you input window-coordinates, the command will output window-relative coordinates.

P2 - (optional) Variable that contains the Quadrant(s) to compute the coordinates for.

© 2023 Theo Gottwald * https://www.smart-package.com

`QFR.` uses the 3.6 Using Quadrant-Coordinates 356.
Can be something like: `{A1} or {A2,A4} or {D4}` or multiple of these.
If you specify multiple Quadrants like `{A1}{A2,A4}{D4}` then the command will use the result of the first calculation
as Input for the next calculation.
This way you can split Quadrants in smaller pieces and calculate "`B4` inside of `D4`". There is no depth limit other
then a mathematical. If you omit **P2**, the command will use `{A1,D4}`.

P3 - (optional) Variable for the result. If omitted, TOS is used.

P4 - (optional) Using the optional Parameter **P4**, `QFR.` can handle any Quadrant-System from 2x2 to 9x9
To use Quadrants that are divided by 8, you would specify an "8".
`QFR.w0,0,400,400|{A1}|$$REC|8`

Example

```
' Use the Testapplication
STW.ct|PBWindowClass:0|TestApp
QFR.|{A1,D4}|$$LAA
GAR.$$LAB
MBX.$$LAA-$$LAB
ENR.

'-----------------------
' Listbox-Element will be divided in Quadrants
'
STW.ct|PBWindowClass:0|TestApp
SCW.nict|1|1029|ListBox|List-Item Nr.1
' Possible Commands are: LBE. (see help) | AGR.
LBE.List-Item Nr.6
' Holt Rect von Listbox
QFR.o{A1}|{A1-A4}|$$REC|4
' and move Mouse there
 GRM.$$REC|$$MID
 MMV.$$MID
```

Remarks

-

Limitations:

-

See also:

- 3.6 Using Quadrant-Coordinates 356
- VAC. - Variable reCt 3546

© 2023 Theo Gottwald * https://www.smart-package.com

- **1.8. Selecting the Application (Topwindow)** 150
- ILC. / NLC. - If Line Color 1805
- IRA. / NRA. - If Rect altered 1824
- RRA. - Remember Rectangular Area 1836
- CST. - Coordinate System Transform 2331
- GAR. - Get actual Rectangle 3619
- MAP. - Move Active Point 2483

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4 ANA. - Imageworks

If you can not localize Controls in any way, you can now use the ANA.-Command to work on a graphical Patterns. This way you can localize things any way.

Also ANA. contains a lot of Graphic-commands that you can use for Batch-Image-Processing of Pictures.

Here are some general Informations that apply to all ANA.-Commands:

1. ANA gives you a Set of 20 Image-Registers.
In each command you can specify one or more Source-Registers, and sometimes also a destination register.

2. In most cases where you do not specify a destination register, the source register will be overweritten with the result of the operation.

3. You can copy or move content from one Image-Register to other Image Registers.

4. You can load an Image into these Registers or you can Snapshot a localized Window into these Registers
5. Then you can do several types of processing with these Images.
6. And finally you can save the pictures.

And:

7. ANA. contains commands for Patterns-Search. These commands are not simple Pattern-Searching Commands, but highly optimized commands that include a Fuzzy-Logic. This way you can also find things that do not look exactly like the pattern but "somehow".

Generally to use ANA. you will first need to fill the Image-Register(s) with some content. That will be processed in further steps.

We will call these "Image-Registers" in Short "IR".

You can use the first 20 Image Registers freely. There are some more Image-Registers available, but they are used by Operations internally (21-32).

The Blit-Command uses the GPU for fast Graphics-Operations. On some Operations there can be artefacts.
If you do not want these, you can use the "ANA.Logic" Command that uses the CPU.

Warning on Variables with Binary Content:
If you have Variables with Binary Content, or Picture-Content (like you get them from `ANA.ToVar`)
then please clear them before you use them as Return Variables for other ANA.-Commands.
Reason:
Generally all Variables that are given to the ANA.-Command are automatically resolved. Unless the Command is prefixed with a "#" (like `ANA.#FromVar`).
Now its possible that Binary Data that might be in a Variable, contains "`$$XXX`" Patterns,
and if these match with a real existing Variable, this could lead to endless Variable resolution.

© 2023 Theo Gottwald * https://www.smart-package.com

Therefore it makes sense that you clear all Variables that contain binary or picture-data before you use them with ANA.-Commands
other then those with a "#" in front of the Command-Name. The same can also happen with other commands then `ANA.`
If your Script hangs unexpectedly, this might be a reason.

Sample Image-Processing Script:

->

```
' Turning a b/w picture into a 3D-Logo
$$PAB=?exeloc\Pics\P3.jpg

$$BOR=4
CAL.$$BOB=$$BOR/2

IEF.$$PAB
  ANA.Load|1|$$PAB
  'ANA.Show|1!
ELS.
  MBX.File does not exist:$crlf$$$PAT
EIF.

ANA.GetRes|1|$$XPO|$$YPO
CAL.$$XPA=$$XPO+$$BOR
CAL.$$YPA=$$YPO+$$BOR

' Make a Resized Version of P3
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.New|2|$$XPA|$$YPA|1
ANA.New|3|$$XPA|$$YPA

' Now mix Shiftzed and Resized Version
ANA.bsh|2|1|$$BOB|$$BOB|1
ANA.Invert|1
'ANA.bw|1|100
ANA.bw|2|600
ANA.Show|1!
ANA.Show|2!
ANA.Logic|3|1|2|A XOR B|RGB
ANA.Show|3!
'ANA.MixTar|1|2|2
ANA.Invert|3
ANA.Show|3!
ENR.

' Changing the first two lines like this, will produce the second picture:
$$BOR=2
CAL.$$BOB=$$BOR
```

IMPORTANT NOTICE about missing Administration Rights Effects

When you start the Script from an SPR-Editor that has NO Admin-Rights, you may not be able to load pictures
from the Standard Robot path because its below C\Programme (x86)\ and you need Admin-Access to load files there.
As a result you will get only BLACK PICTURES using ANA.Load.
So if you only get Black Pictures, try to use ADMIN-RIGHTS for the Script.

More Commands that are not yet documented:

```
ANA.SetMaxResults (no. of Results)

ANA.CompareBlock Target Index|Pattern Index|X-Pos Pattern|Y-Pos Pattern|Color To
Will put 2 Values on the TOS, TOS -> Number of Pixelhits, TOS-1 -> 0/1 if Error

ANA.ShowHits|Index IR with Snap|Size of Marker X|Size of Marker Y

ANA.ResetList

ANA.ResetResultsPack

ANA.SortResultsPack P1|P2
P1 - 0/1  (0 - Descending, 1 - Ascending Sort)
P2 - 0/1 (0 - Sort Y / 1 - Sort X)

ANA.GetResultItem P1|P2
P1 - Number of Item to get on TOS
P2 - 0/1 delete Item after get
```

Will place Items on the TOS:

```
0/1 - Error/Success
X,Y - Middlepoint of FoundRect
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
X1,Y1,X2,Y2 - Found Rectangle
Number of Source--Pattern (Multi-Find) or 1
Pattern-Size as String "8x8"

ANA.SnapFind1 P1|P2|P3|P4|P5|P6
P1 - Filepath Patternfile
P2 - Farbtoleranz
P3 - Pixel-Hits min.
P4 - 0/1 Debugmode
P5 - "0,0" Positionscorrection
P6 - Maximum Number of Finds
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.1 !The Human-Eye Recorder

Currently the following Pattern-Sizes are supported:

The following Pattern-Sizes are supported:

```
4x16
4x32
4x64
8x8
8x16
8x32
8x64
9x9
10x10
11x11
12x12
16x8
24x8
32x4
32x8
32x16
40x8
48x8
56x8
64x6
64x8
96x4
96x8
98x6
128x4
128x8
```

The "Debug-Flag"

The Debug Flag has been constantly enhanced during development. Currently the following Options are supported:

1 - Switch Debug Console ON

2 - Use "Snap again" during Search. This will update the Snapshot using the Parameters from the last taken Snapshot.

4 - Use "X-Ray Search". This will start an enhanced search that is more color-independent and takes much longer time.
Especially together with Option "64 -Extended Search" it may find a lot more results.
Using this Option, the most upper Line and the most left Line are ignored.
Therefore this Option will reduce the effective Pattern-Size by -1 in X- and Y-Direction.
For example a Pattern that was size "8x8" will be a Pattern size "7x7", this is important when you specify needed "Pixelhits".

8 - If added, this will disable one Layer of "Fuzzy-Logic". This Flag will make the Search more sensitive to small color-differences

© 2023 Theo Gottwald * https://www.smart-package.com

16 - If used, this Flag will internally invert the Images. Use it preferably with Option "4 - X-Ray Search".

32 - If used, this Flag will internally complement the Images. Use it preferably with Option "4 - X-Ray Search".

64 - "Extended Search". This search uses longer time, but returns more results. Use it preferably with Option "4 - X-Ray Search"

Using the Human-Eye Recorder and Pattern-Search Technology

Traditionally the SPR tries to find a Control or GUI-Element using the Windows- or Accessibilty-Properties.

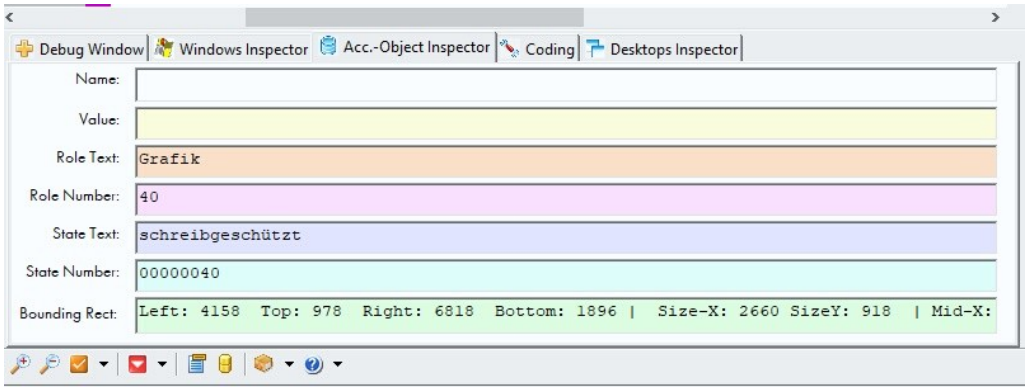
There are cases where GUI-Forms (for Example if they are Proprietary, Java-based, or in unsupported Browsers)
can not be analyzed, and therefore could traditionally not easily be accessed.

For all cases where you can not find anything using the traditional Technologies, the SPR now supports the "Human Eye Technology".

It does not analyze any "Windows internal structures". It does not need any PlugIns.
It will simply do what you also do: It will "Look at the Screen".

When do we use it?

The first thing you should do is, find out if you can use conventional Technology.
For this, the Acc.-Inspector which is built into the SPR-Editor gives you all needed Information.
If it does not reflect the Text in the Form, you are going to analyze, then the way to go is to use the new HE-Technology.



Using the Acc.-Inspoector you can easily find out if the conventional Acc.-Technology will work or not.

How does the HE-Technology work?

It will take Snapshot of the currently localized Window.

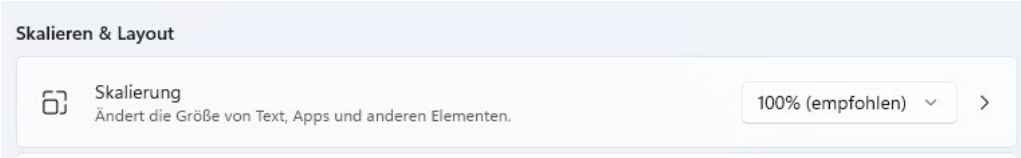
© 2023 Theo Gottwald * https://www.smart-package.com

Then it will simply find a Pattern that you provide.
It will work **very reliable** also on different Computers as long as the Application Sizing (dpi) is the same as on the recorded Computer.
This is because the internal "Fuzzy Logic" will recognize Patterns that are different from the original Pattern to a large degree.

When will it NOT work?
If the target System is set to 120% and the recording was made with 100% the current implementation will NOT work.
The reason is very Simple, in such case the Patterns do not match.

1. Best set the Scaling of your Screen to 100%, otherwise the HE-Recorder may not record the proper places.

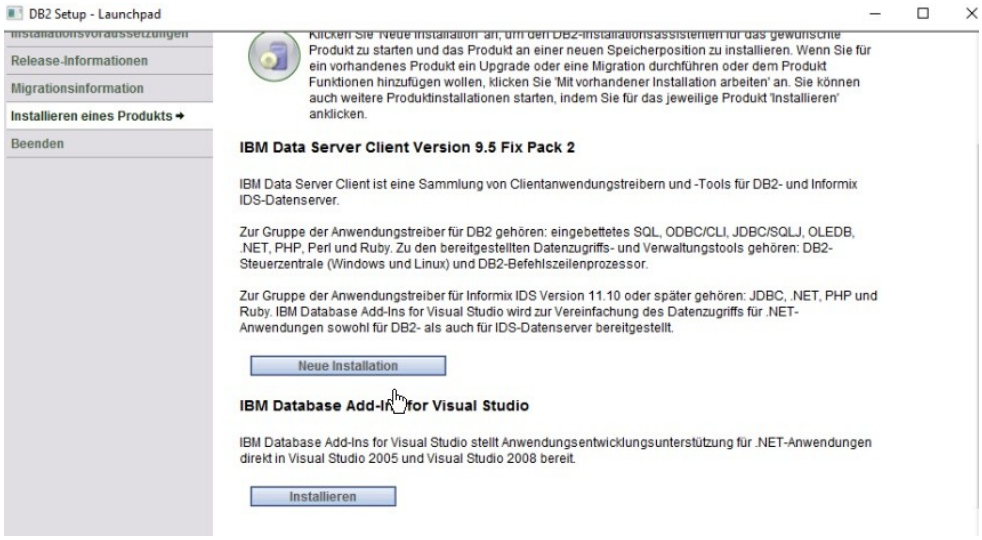
2. Under Windows 11 - generally - and under Windows 10 depending on what you are doing, the HE-Recorder NEEDS ADMIN_RIGHTS.
Otherwise you will see that only unicolored Boxes are recorded. This is a typical Sign that the HE-Recorder does not have sufficient Rights.

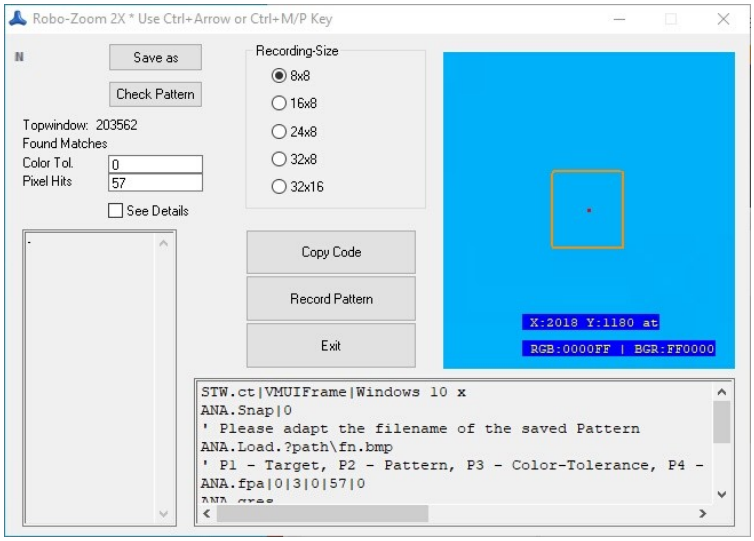


The whole Process is only a few steps.

1. Start The HE-Recorder, choose a Patternsize, for Example 16x8
2. Click on "Record Pattern"
3. Move the Mouse to the Pattern. Try to have the Pattern touch the left upper Border
4. Click the Left Mousebutton to record the Target Pattern
5. Click on "Check Pattern". A new Window opens and you can verify if the Parameters are perfect.
6. Now you can change the Parameters for "Color-Tolerance" and "Pixel-Hits"
7. When you are done, click on "Save as" and Save your recorded Pattern directly into the Project Folder.
8. Click the "Copy Code" Button to get the Code into the Clipboard.
9. Paste the Code into sour Script.

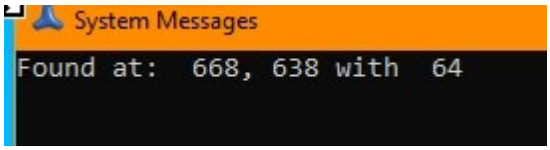
© 2023 Theo Gottwald * https://www.smart-package.com





This is the Form where we recorded a Buttonclick.

This is the HE-Recorder how he looks with the Record.



This is the Result when we click on "Check Pattern".

This is the Window that comes up when the "See Details" Button is checked.

*© 2023 Theo Gottwald * https://www.smart-package.com*

What does "Check Pattern" do?

"Check Pattern" will show you, if you have the right Parameters and get all needed Finds on a page.
You can change the "Color Tolerance" and the "Pixel Hits" Parameters to get different results.

What is the "Color Tolerance"?
The Color-Tolerance value is a number between 0 and 765 that will tell the Robot how "fuzzy" he should treat Pixels.
Even if the number is zero, there is still an internal tolerance that is generally applied. So in most cases zero will be fine.
In special cases (see example below) it may help to use a value up to 200. Higher values are not recommended.

What is the "Pixel Hits" Value?
Each Pattern has a number of Pixels, for example a 8x8 Pattern has a number of 64 Pixels.
A Pattern of size 24x8 has a Total of 192 Pixels. The Pixel Hits Values tells the Robot, how many hits are needed to accept the result.
So for a 8x8 Pattern, the value can be between 1 an 64 (while values below 32 make no sense).
And for a 24x8 Pattern the value can be 1 to 192. Again values lower then 50% of the total Pixel number do not make sense.

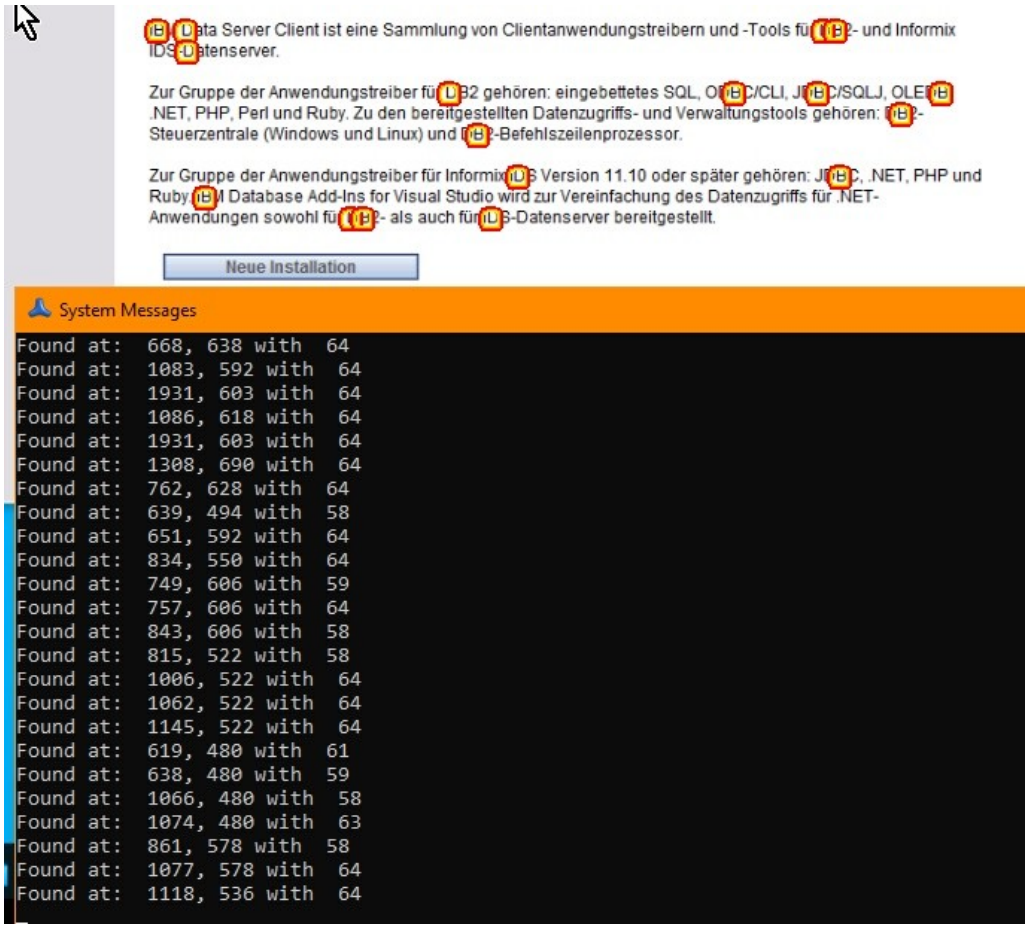
What is the "See Details" Checkmark good for?

If you use this Checkmark, a new Window will appear that show for each hit, how many Pixel-Hits it has.
In the example below (recorded with a very high Color Tolerance) we got several Hits below 64 Pixel-Hits.
So you can decide if you want these or not and adjust the Pixel-Hits - Value to lowest necessary number.

Always try to keep the Pixel Hits as high as possible to just get the results you want to get.

Also try to keep the Color Tolerance on zero if possible.

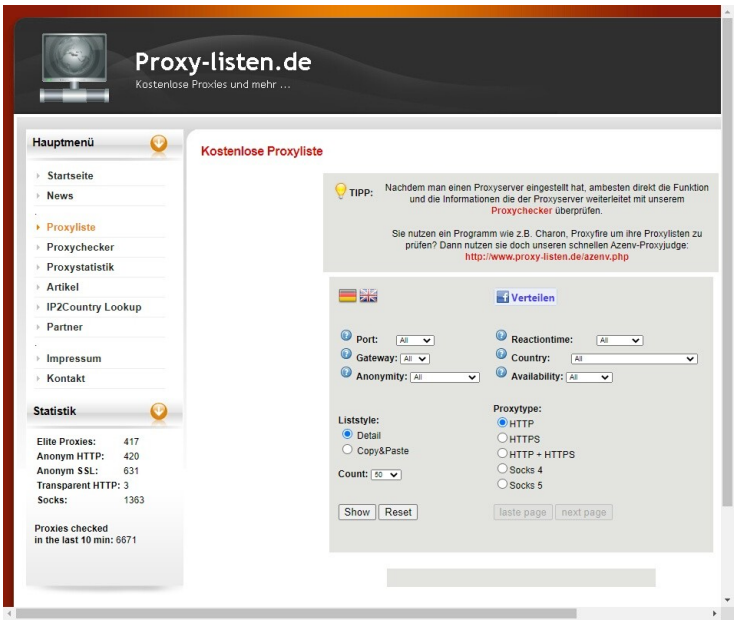
*© 2023 Theo Gottwald * https://www.smart-package.com*



Problems that can happen and the solution

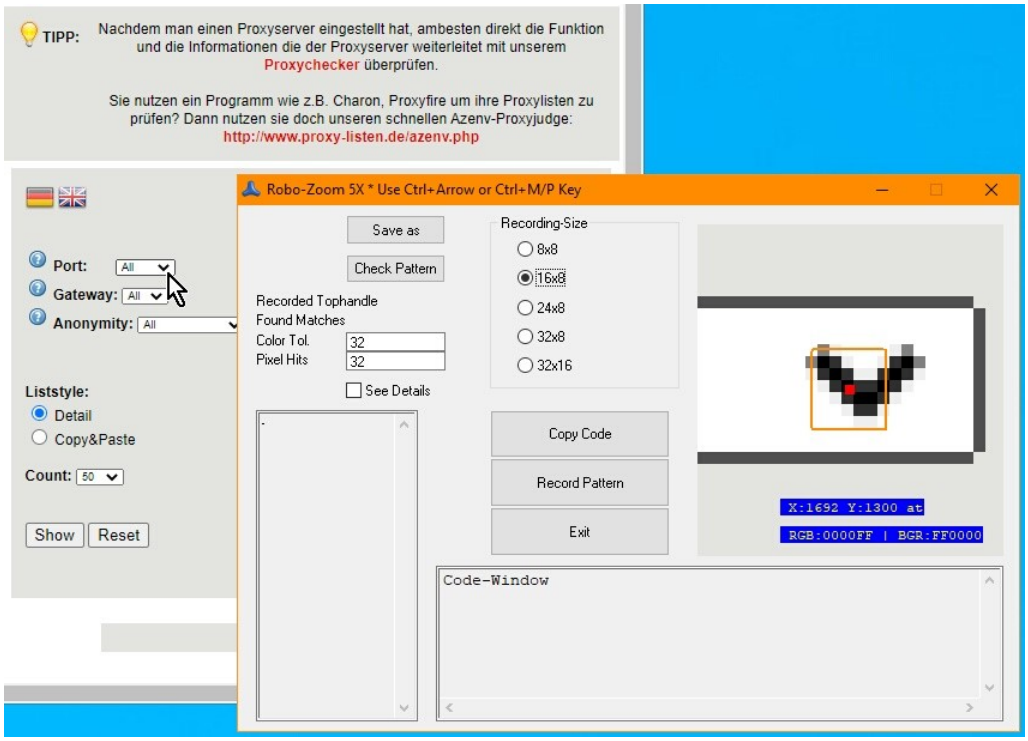
1. Lets say you have the Situation that you want to Click on some Control on a WEB-Site inside a Chrome-Type Browser.
These Browsers have not been supported so until now. Lets take this as an example.

*© 2023 Theo Gottwald * https://www.smart-package.com*



1. First we are starting the "HE-Recorder". It looks like this (Design and Features may change with next Updates).
The First thing we need to set is the "Recording Size". Currently you can choose between: 8x8, to 32x16 Pixel Objects.
As a rule of Thumb, the smaller the Object the better. Lets choose 16x8 this time.

2. While you move the Mouse over the Target Form, you can see how the Display in the "Robo-Lens" changes.
You can change the Magnification using the "ctrl-M" and "ctrl-P" Combination.

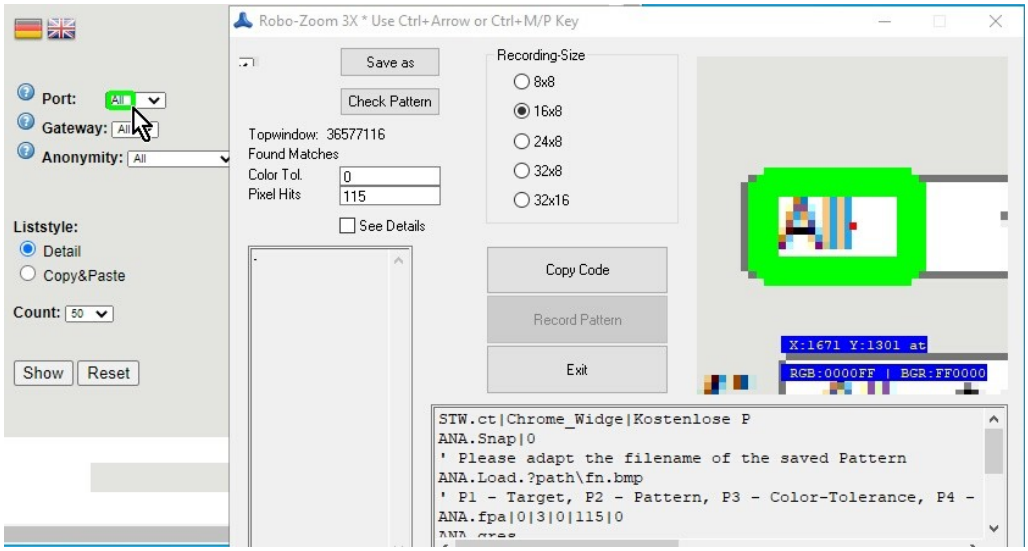


3. So next click "Record Pattern". The Mousepointer will change t and show a green Box that has inside exactly the size of 16x8.

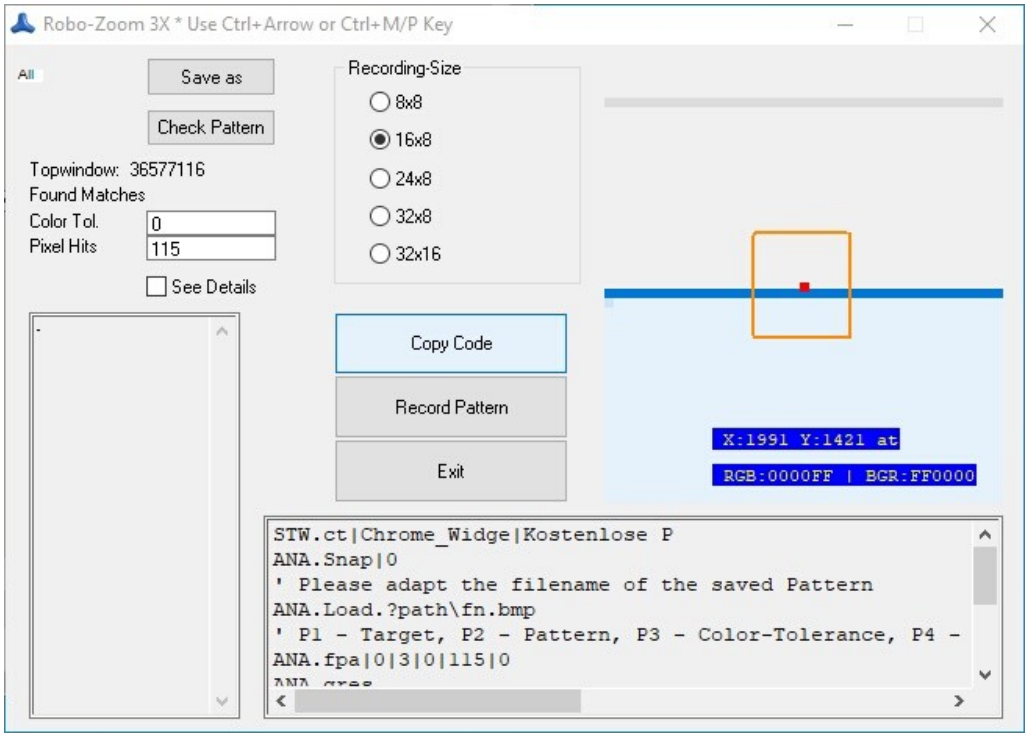
© 2023 Theo Gottwald * https://www.smart-package.com

Now drive to that Pattern and click the left Mouse-Button.

Hint: If the green Box will stay and not move, just click on the green Box and then Press "Record Pattern" again.

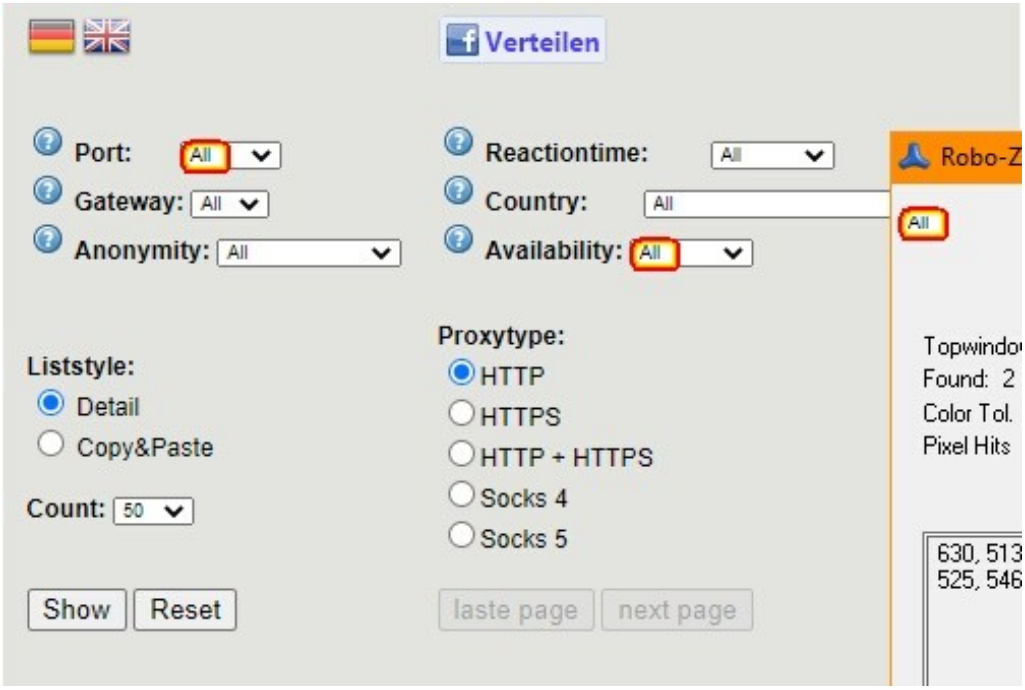


4. When the Recording is done, it is displayed in the upper left Corner. The Textbox down there shows the current Code.
But wait we are not yet done.



5. Click the "Check Pattern" Button. A new Window will appear and look like this:

© 2023 Theo Gottwald * https://www.smart-package.com



6. The Robot has found 3 Places with that Recorded Pattern. This is a very Simple Situation, you could just:

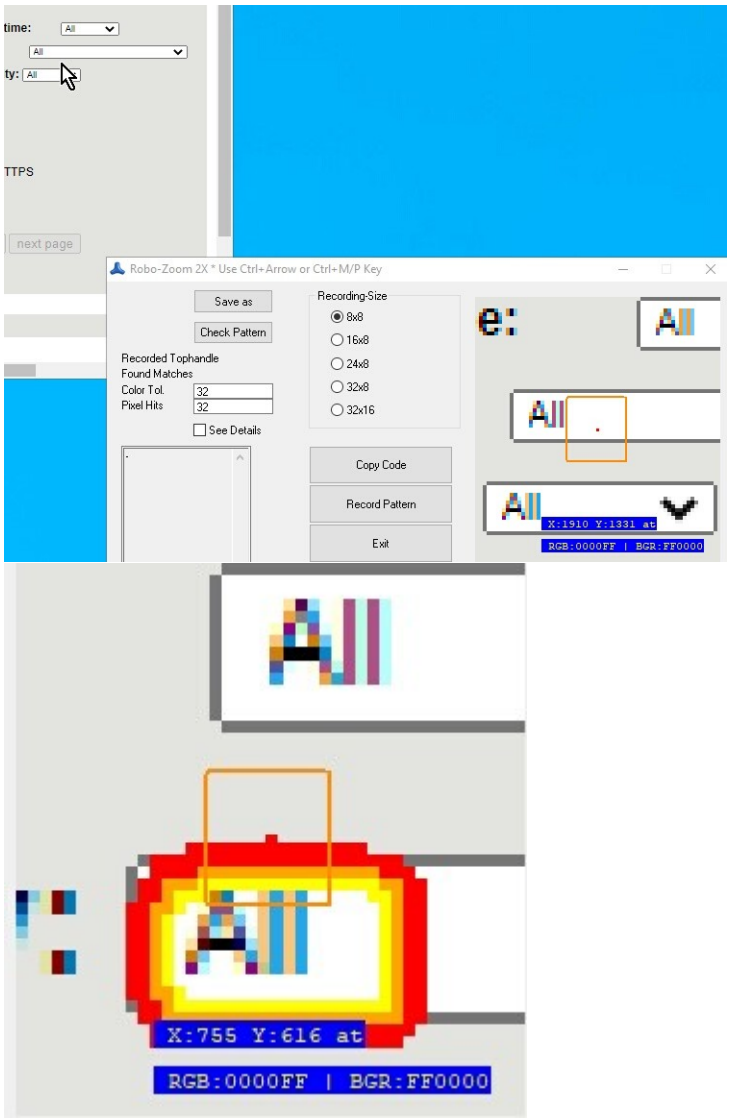
- Save the recorded Pattern into your Project Folder
- copy the code from the Textbox into your Script and correct the path for the Loading of the Pattern.
and you would be ready to go.

However, as you can see there are 4 Places with "All" that have not been selected. Click the Checkmark "See Details" and change the Value for "Pixel-Hits" to 99.

And change the Color-Tolerance Value to 200. Now click "Check Pattern" again. This time we will get nearly all "All"'s. Just one does not want to get selected.
The Lens will show you why. As you can see the rendering style for this All is too different. Watch out for this problem. If that happens, just record it separately.

Choosing the Color-Tolerance much too high may lead to thousands or even hundred thousands of "hits" that have to be tested.
Doing so may render the HE-Recorder "busy" for several Minutes. So better do not try Color Tolerances larger then 200. If that happens, just wait until the Recorder is available or in the worst case, kill it using the Task-Manager.

© 2023 Theo Gottwald * https://www.smart-package.com



If the Rendering style of the Pattern is too different, you may need to use separate Recordings.

The Final Code:

```
' Locate the Target Window
STW.ct|Chrome_Widge|Kostenlose P

' Snapshot it into an IR
ANA.Snap|0


' Please adapt the filename of the saved Pattern
ANA.Load.?path\fn.bmp

' P1 - Target, P2 - Pattern (Path and Filename)
' P3 - Color-Tolerance 0-765
' P4 - min. Hits (Pixel) this Value depends on the Patternsize. A 8x8 Pattern ha
' Generally the Recorder uses max. Hits-10%,
' P5 - Debugmode 0/1 - this is the same like the Details-Switch in the Recorder.
```


© 2023 Theo Gottwald * https://www.smart-package.com

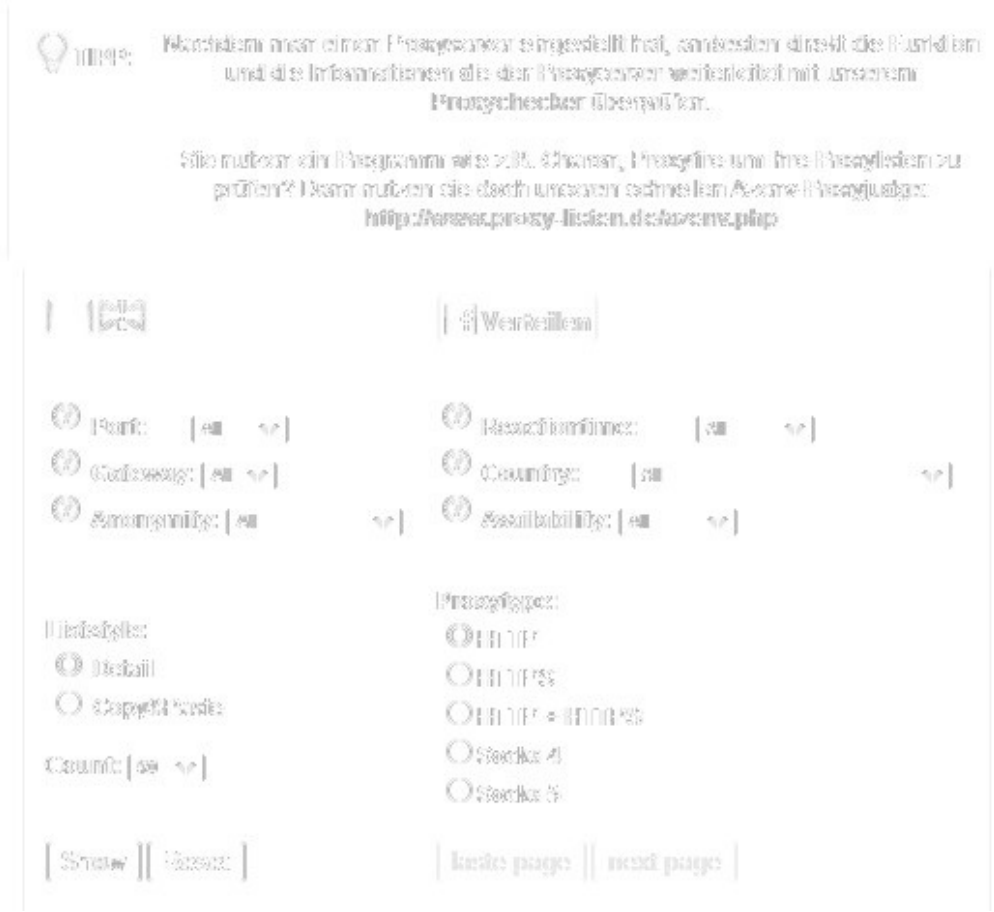
```
ANA.fpa|0|3|0|57|0

' Get The results
ANA.gres

' We take away the number of Hits from TOS
POP.$$NOH

' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits

:Lab_NoHits
```

More Options?

Due to the Architecture of the ANA.-Commands, you can also use the ANA.fpa - Command on
Objects that have been processed.
You can use the ANA.Mask - Command, or use the ANA.Cdfx or other Commands to process the picture before applying the "FindPattern".
Of course you will need to also Record the Pattern then from an image that was processed before or just also apply the process to the Pattern.
In case of ANA.cdfx and ANA.cdfy it may be better to record from an processed picture.

```
ANA.Load|0|$$PAC\Plist.png
ANA.Load|1|$$PAC\All8.bmp
ANA.cdfy|0|2
ANA.Invert|2
ANA.Show|2!
```


© 2023 Theo Gottwald * https://www.smart-package.com



This picture shows a ANA.cdfy processed version of the Browser-Window.

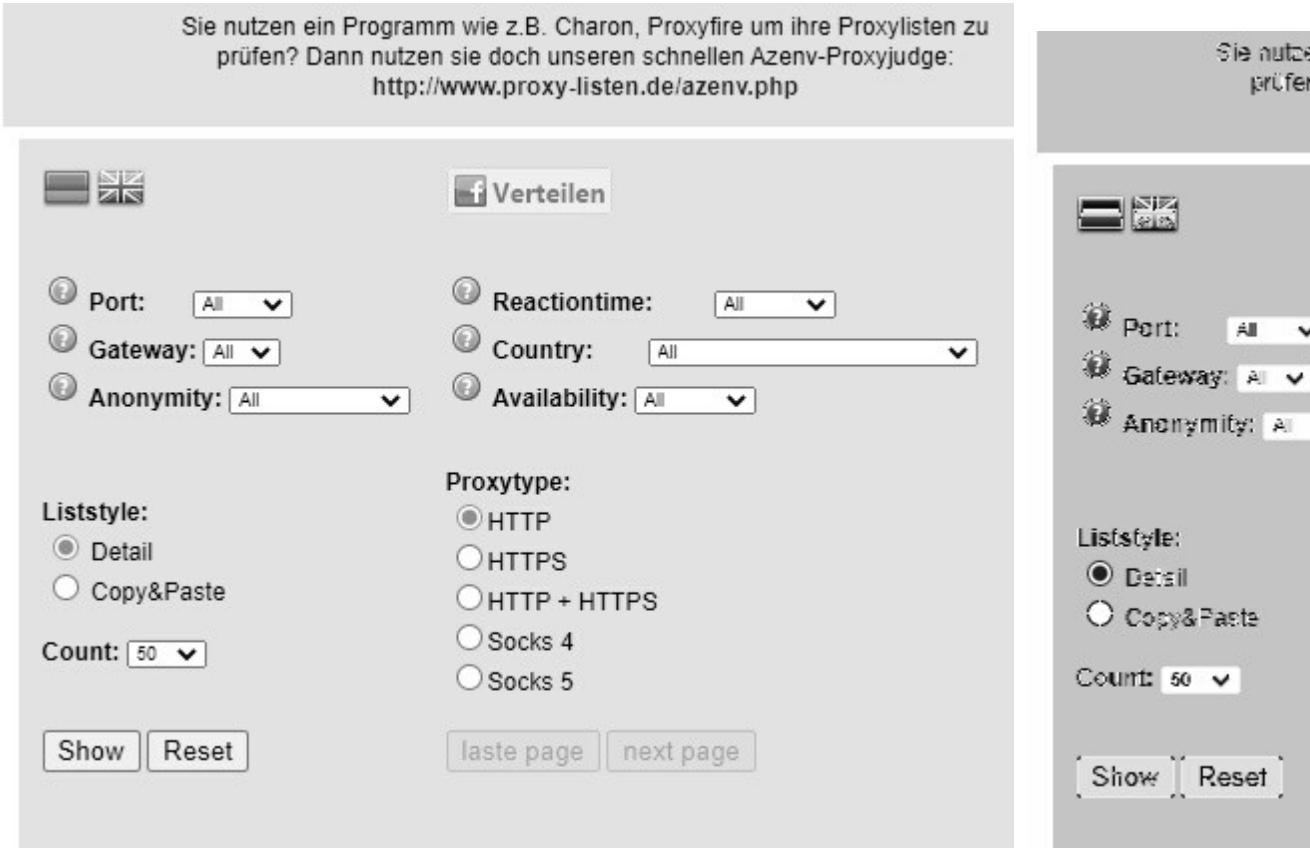
```
' Using the ANA.TriColor Option you can use TriColor pictures as Source
ANA.TriColor|0
ANA.Show|0!
```


© 2023 Theo Gottwald * https://www.smart-package.com

The ANA.Grey Command, here I used the "1" and the "2" as Parameter. Note that in the right picture the Radiobuttons an

```
ANA.Grey|0|&H2
ANA.Show|0!
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



IMPORTANT NOTICE about Mouseover Effects:

When recording with the HE-Recorder, you may not see that the Button that you are over is changing its color. Due to "Mouseover Effects".

It is however visible for the FindPattern Command.

If you do NOT get AT LEAST the Pattern that you have recorded with a Color-Tolerance of "0" then you have had "Mouseover" Effects.

To prevent these, make a Screenshot of the Form (but not as .jpg due to Artefacts, but as .png or .bmp) and record from the Screenshot.

Recording from a Screenshot will prevent "Mouseover Effects".

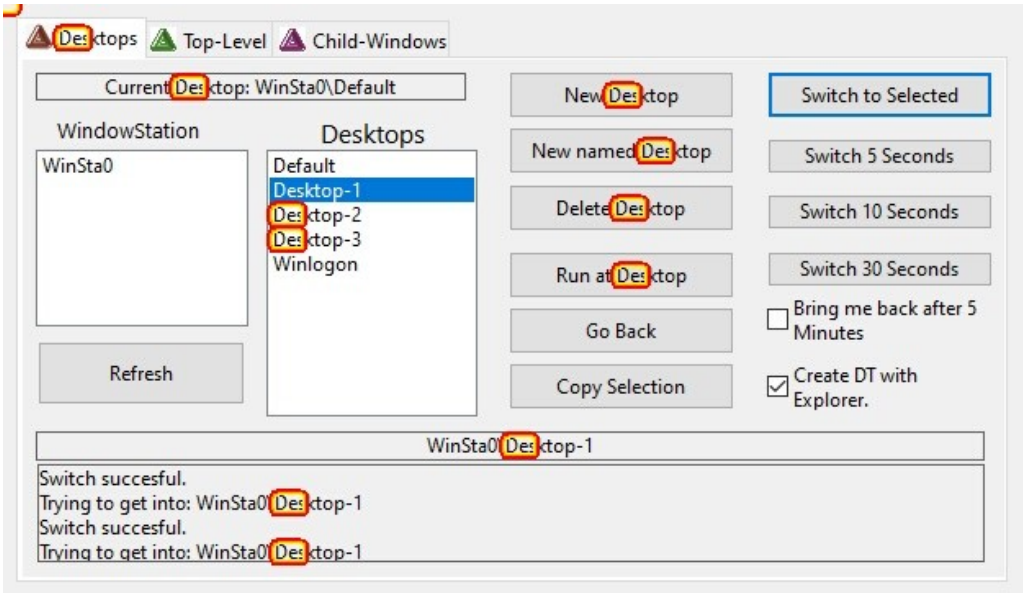
You can test this yourself with our "Desktops-Manager" Tool. If your record a Button, there is an invisible "Mouseover Effect".

Therefore you will not get even one result with a Color-Tolerance of "0".

If you record from a Screenshot of the program, you will immediately get the hits even with a Color-Tolerance of "0".

In the picture below we used a Color Tolerance of 50 to get the desired text on grey as well as on white background.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.2 Blit-Operations

These Operations move IR-Content using the graphics Hardware.

© 2023 Theo Gottwald * https://www.smart-package.com

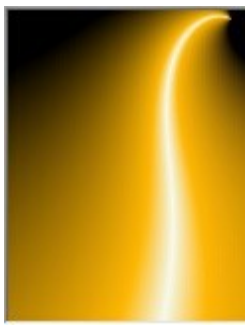
3.55.4.2.1 !Blit-Modes

In all Commands that have "Blit-Mode" Parameter you can specify any of these numbers:

Blit-Modes:

1 - "Source" The Pixel is always the Source. Sa

ANY Destination =



2 - "NotMergeSrc" The Pixel is the inverse of the Me

=

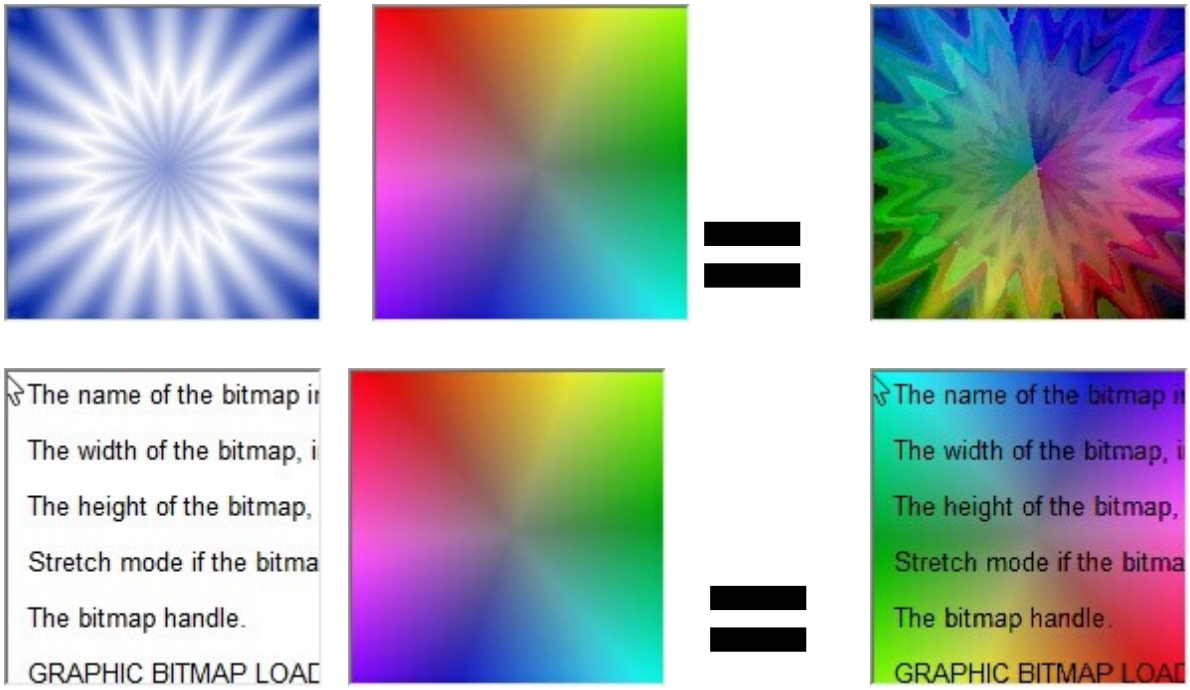
=

© 2023 Theo Gottwald * https://www.smart-package.com

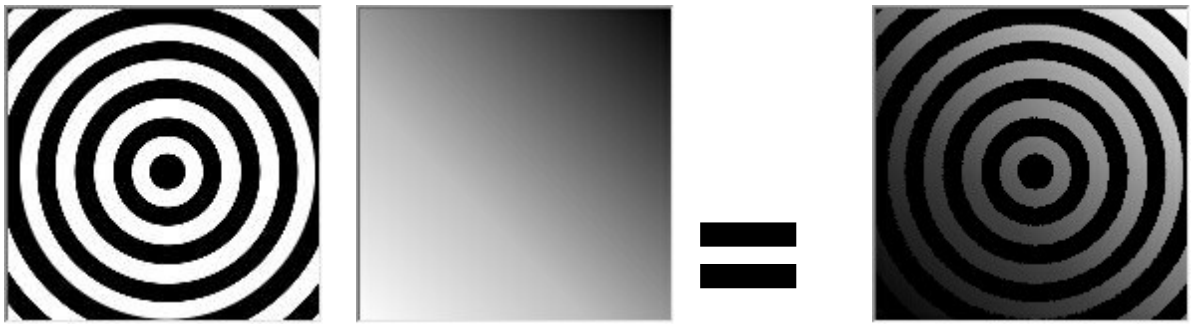
3 - not implemented

4 - "NotCopySrc" Pixel is the inverse of the Source

5 - "MaskSrcNot" Pixel is a combination of the colo

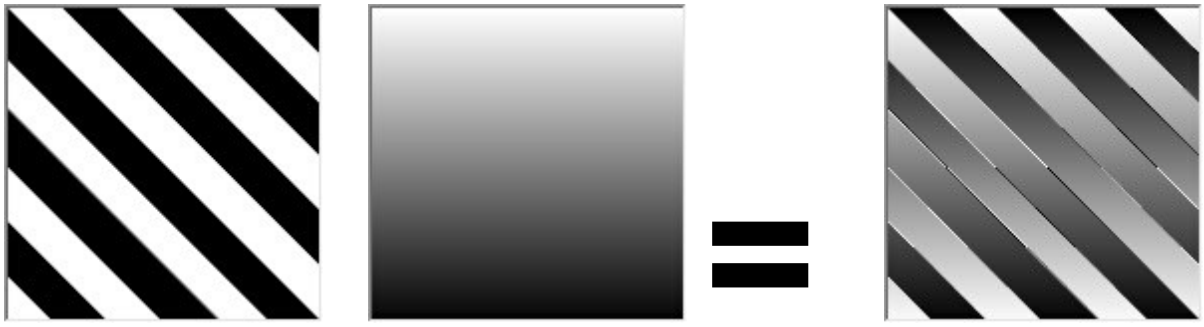


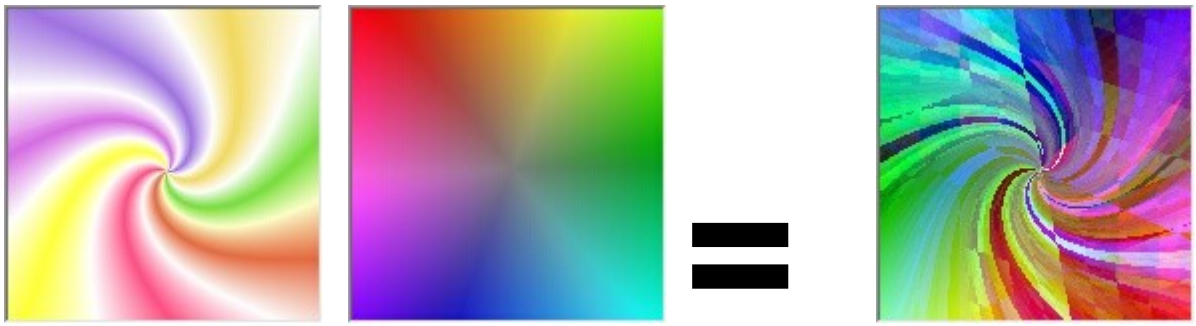
© 2023 Theo Gottwald * https://www.smart-package.com



6 - not implemented

7 - "XorSrc" Pixel is a combination of the colors i

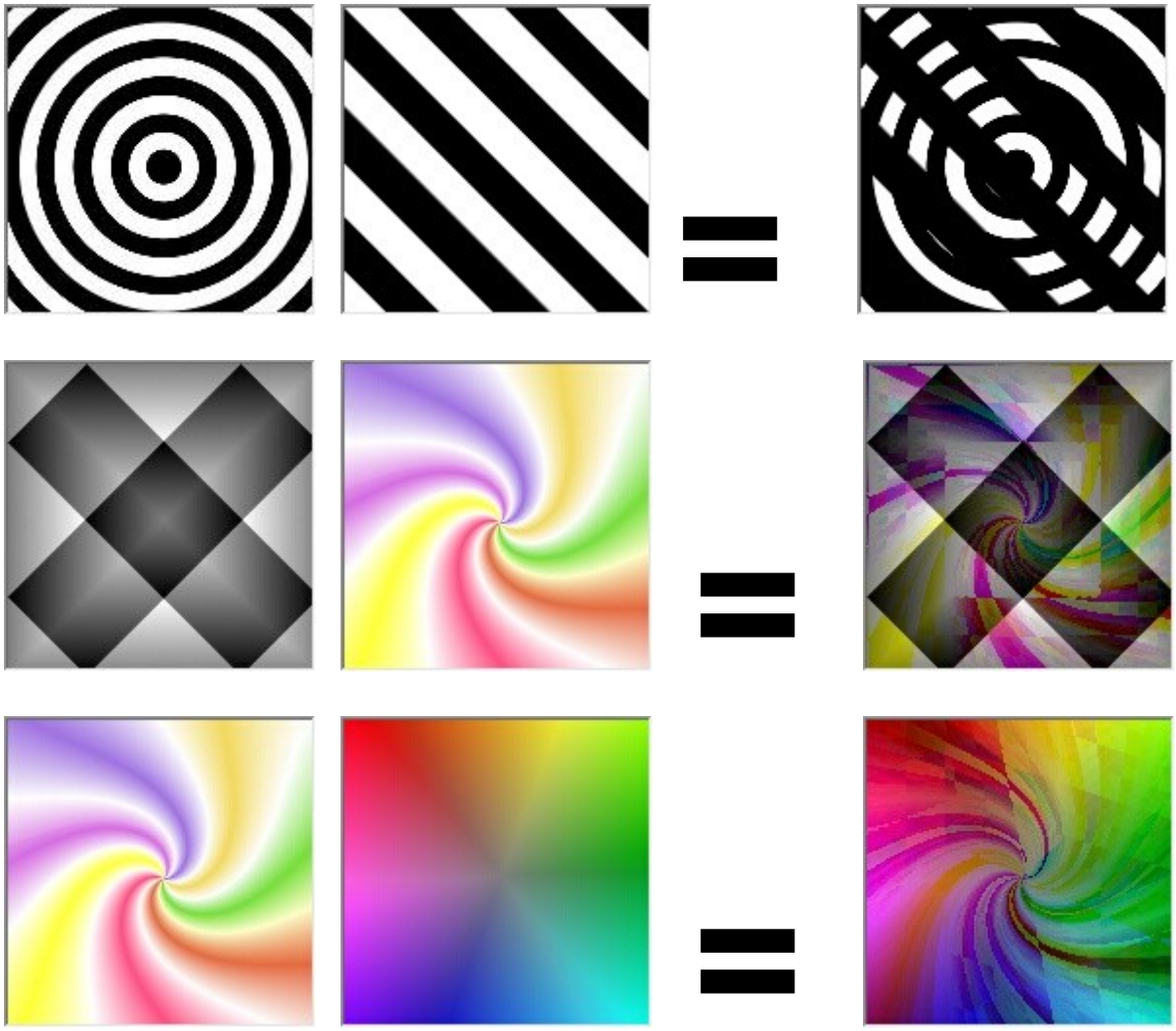




© 2023 Theo Gottwald * https://www.smart-package.com

8 - not implemented

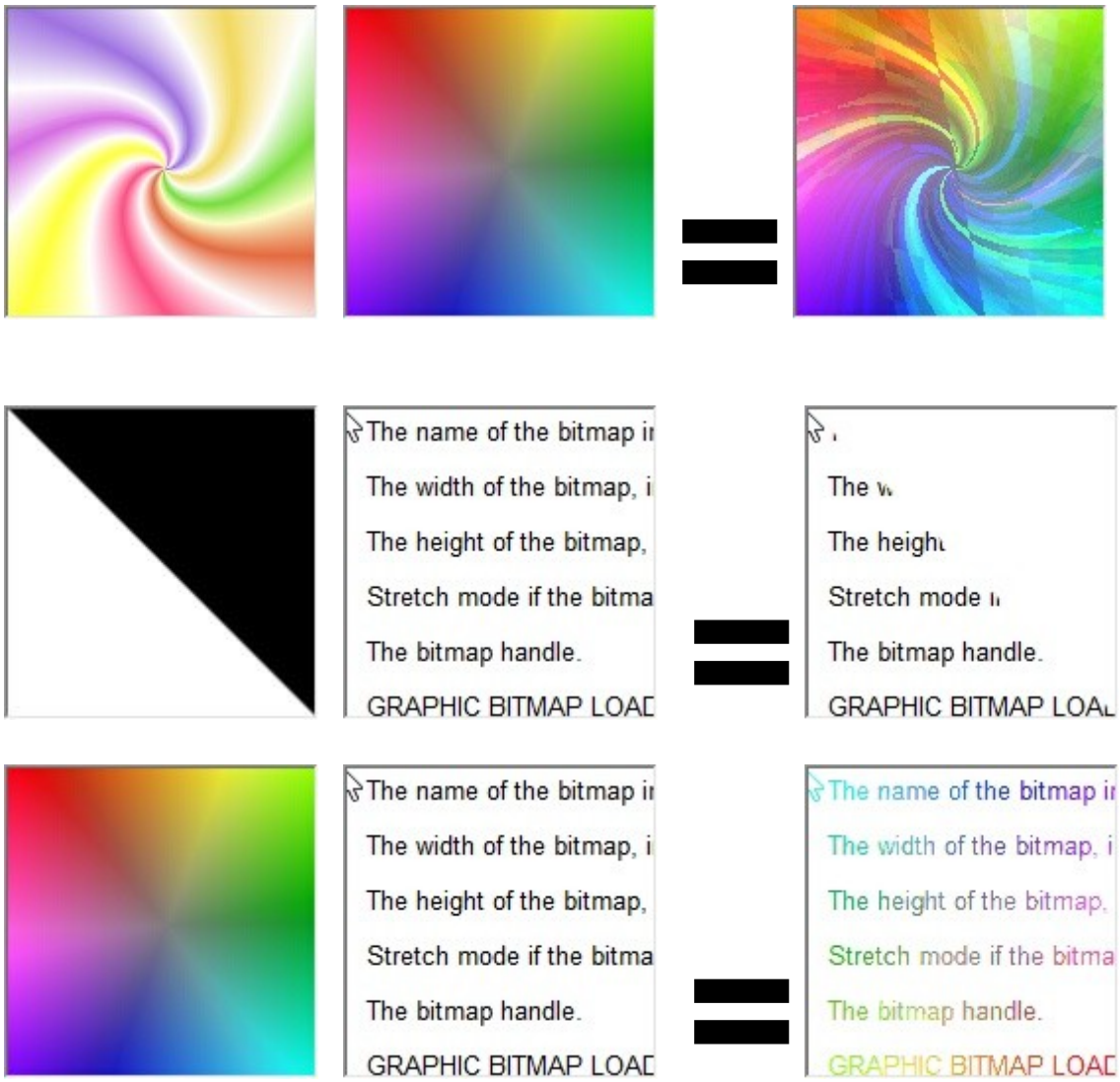
9 - "MaskSrc" Pixel is a combination of the colors c



10 - not implemented

12 - "MergeNotSrc" Pixel is a combination of the source c

© 2023 Theo Gottwald * https://www.smart-package.com

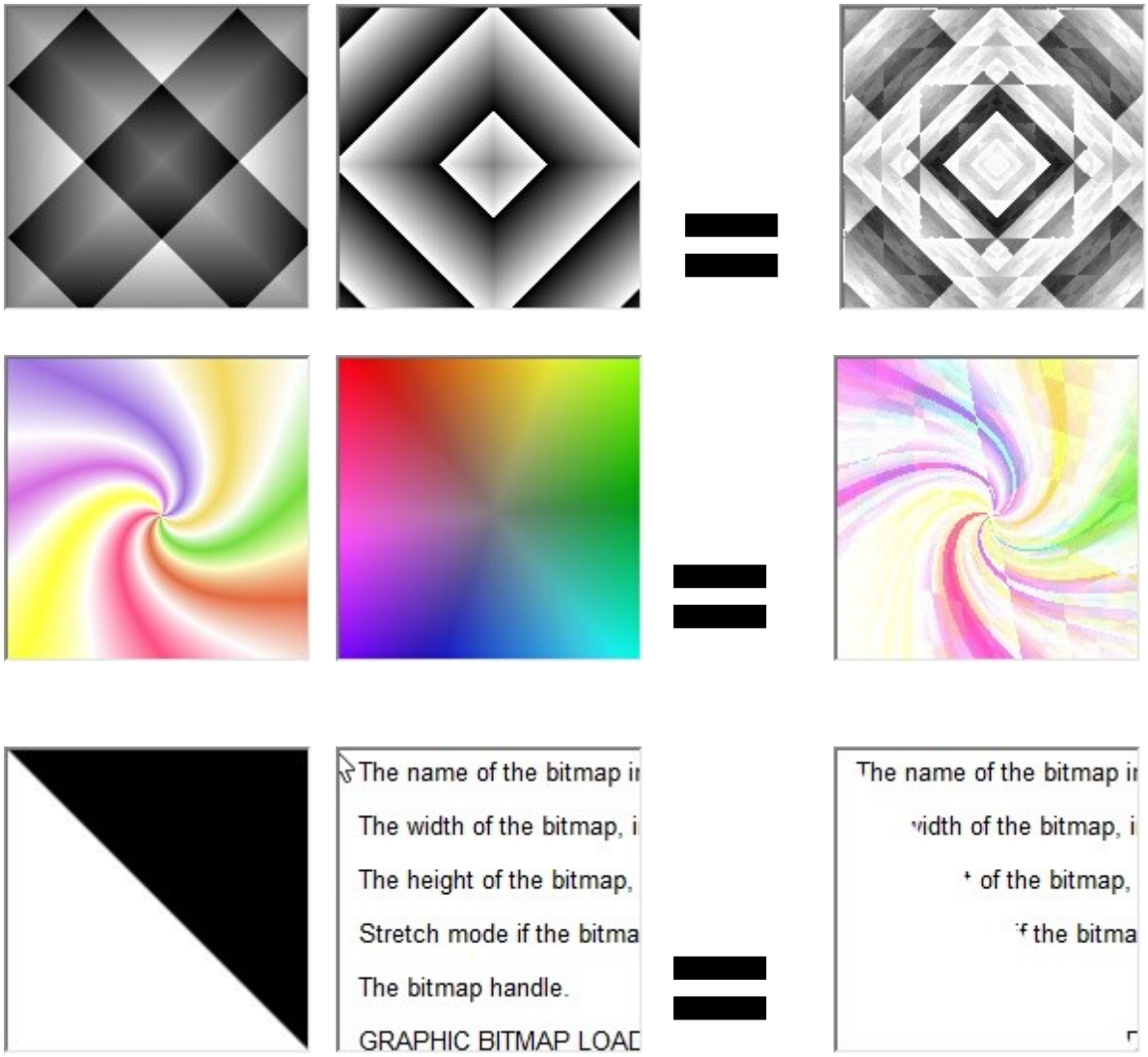


13 - "CopySrc" Pixel is the source color (default).

14 - "MergeSrcNot" - not implemented

15 - "MergeSrc" Pixel is a combination of the source c

© 2023 Theo Gottwald * https://www.smart-package.com



16 - "mix_Whiteness" Pixel is always 1 (white).

n.i.

IMPORTANT NOTICE about missing Administration Rights Effects

When you start the Script from an SPR-Editor that has NO Admin-Rights, you may not be able to load pictures
from the Standard Robot path because its below C\Programme (x86)\ and you need Admin-Access to load files there.
As a result you will get only BLACK PICTURES using ANA.Load.
So if you only get Black Pictures, try to use ADMIN-RIGHTS for the Script.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.2.2 ANA.MixTar

MiniRobotLanguage (MRL)

ANA.MixTar

Combine multiple Image-Register's (IR) in different ways

Intention

This command will combine/mix two IR's in multiple ways depending on the Blitmode.
For a Simple Color-Mix, use the "`ANA.BlendTo`" Command.
For special Mixing, look at "`ANA.MixRGB`" and there is lots more.

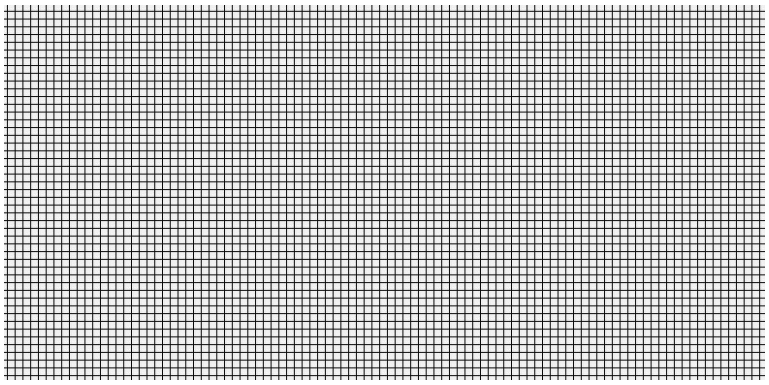
1. Mix a horizontal and a vertical Gradient and the n again mix these with another Pattern from the Fill-Command.
This will generate the Graphic below.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
ANA.Show|2!
ANA.New|1|800|400
ANA.Fill|1|&H000000|5|&HFFFFFF
ANA.Show|1!
ANA.mix|1|2
ANA.Show|2!
PAU.1
ENR.
```

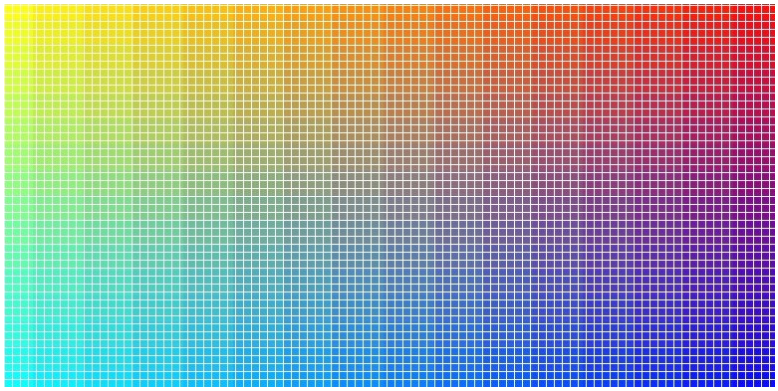
Using Blitmode 7:

© 2023 Theo Gottwald * https://www.smart-package.com

+



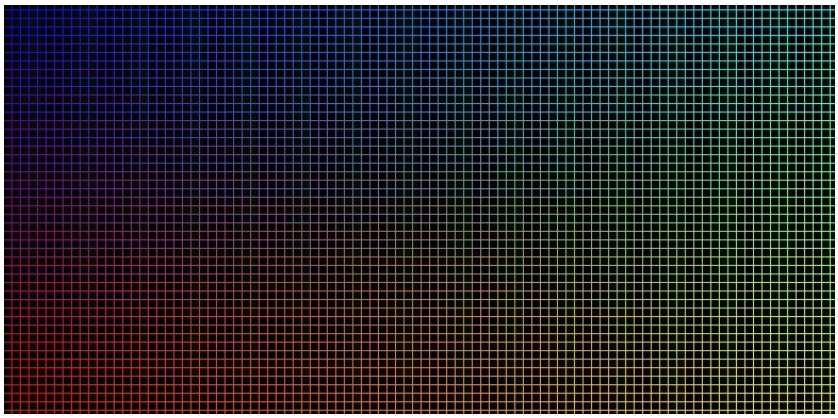
=

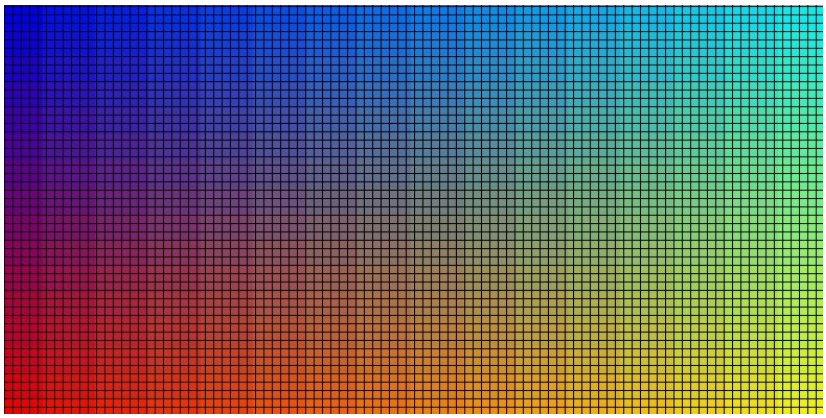


Using Blitmode 5 in the second Mix, you get this:

Using Blitmode 2 in the second Mix, you get this:

© 2023 Theo Gottwald * https://www.smart-package.com





2. Mixing IR's with themselves doe not make much sense as you will mostly get whether a white, black or inverted result.
However you an do that.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.Show|0!
ANA.Mix|0|0|3
ANA.Show|0!
ENR.
```

3. Mixing IR's using a Blimtmode to let picture "shine through".
In this script we will make the second picture to shine through the first picture using Blimtode 2.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Dependinfd on the position of the

```
You can use:
ANA.mix|0|1|2
ANA.invert|1
or
ANA.invert|1
ANA.mix|0|1|2

to get any of these results below.
```

Command, the result may be any of these:

```
$$PAA=?exeloc\Pics\P3.jpg
$$PAB=?exeloc\Pics\P4.jpg
IEF.$$PAB
  ANA.Load|1|$$PAB
  'ANA.Show|1!#
ELS.
  MBX.File does not exist:$crlf$$$PAT
EIF.
IEF.$$PAA
  ANA.Load|0|$$PAA|1
  'ANA.Show|0!#
ELS.
  MBX.File does not exist:$crlf$$$PAT
EIF.
ANA.mix|0|1|2
ANA.invert|1
ANA.Show|1!#
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.MixTar[|P1][|P2][|P3]

Parameter Explanation

P1 - opt. Source Image-Register Number, if omitted, IR 0 will be used.
P2 - opt. Destination Image-Register Number, if omitted, IR 0 will be used.
P3 - opt. Blitmode 1156 **for Mixing.**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.MixTar-Sample
'**********************************
ANA.New|0|800|400
ANA.vgradient|0|&H00FF00|&HFF0000
ANA.New|1|800|400
ANA.hgradient|1|&H00FF00|&H00FFFF
ANA.Show|0!
ANA.Show|1!
$$BLI=7
ANA.mix|0|1|$$BLI
ANA.Show|1!
ENR.

' Here you can see a part of the result of the Operation.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.2.3 ANA.BlendTo

MiniRobotLanguage (MRL)

ANA.BlendTo

Blend multiple Image-Register's (IR) in different ways.

Intention

This command will combine/mix two IR's in specified strength for each IR.
This way you can "Blend In" an IR into another.

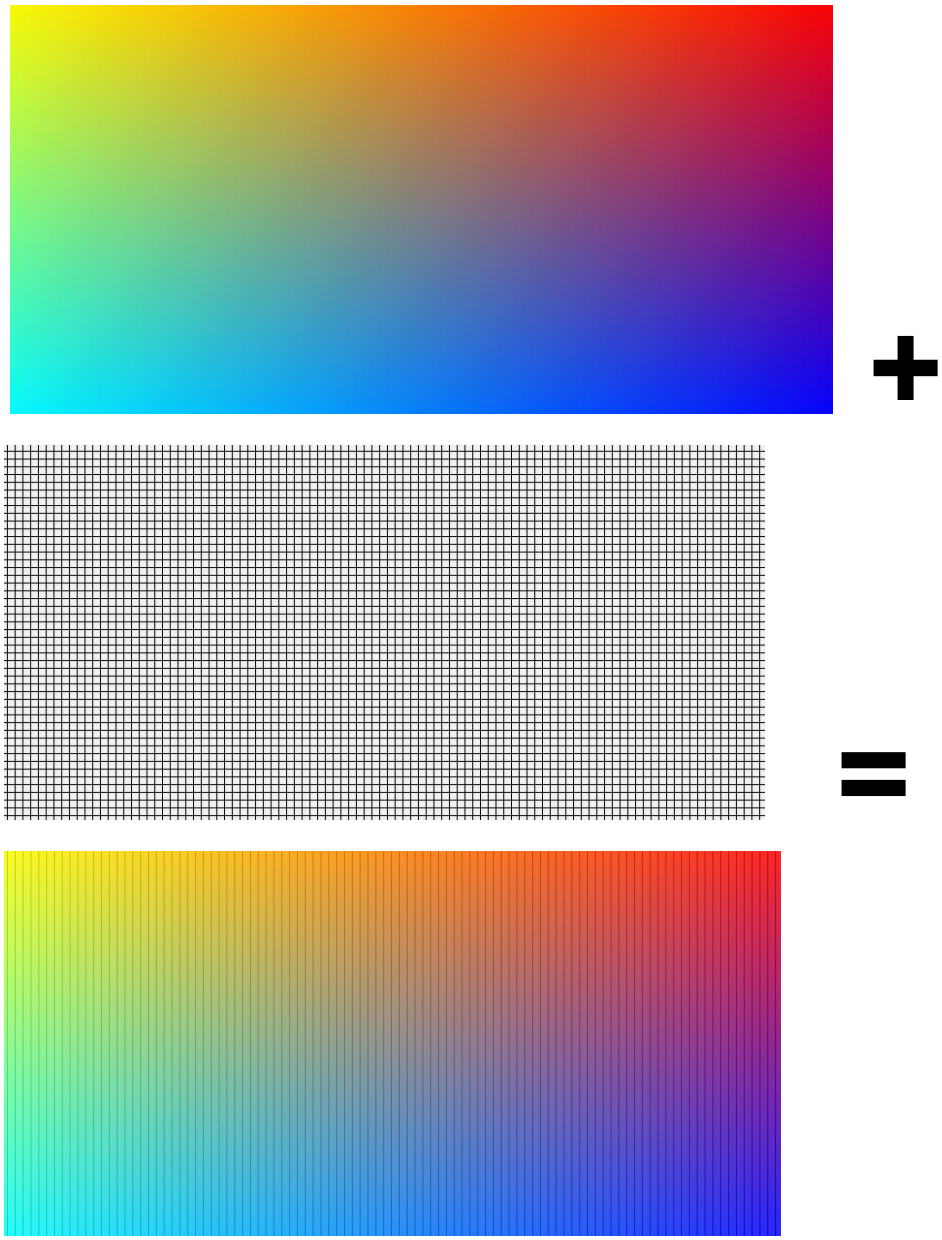
For a Simple Color-Mix, use the "`ANA.BlendTo`" Command.
For special Mixing, look at "`ANA.MixRGB`" and there is lots more.

1. Mix
This will generate the Graphic below.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
ANA.Show|2!

ANA.New|5|800|400
ANA.Fill|5|&H000000|4|&HFFFFFF
ANA.New|6|800|400
ANA.Fill|6|&H000000|3|&HFFFFFF
ANA.BlendTo|5|6|7|80
ANA.Show|7!

ANA.Show|1!
ANA.BlendTo|7|2|3|40
ANA.Show|3!
ENR.
```

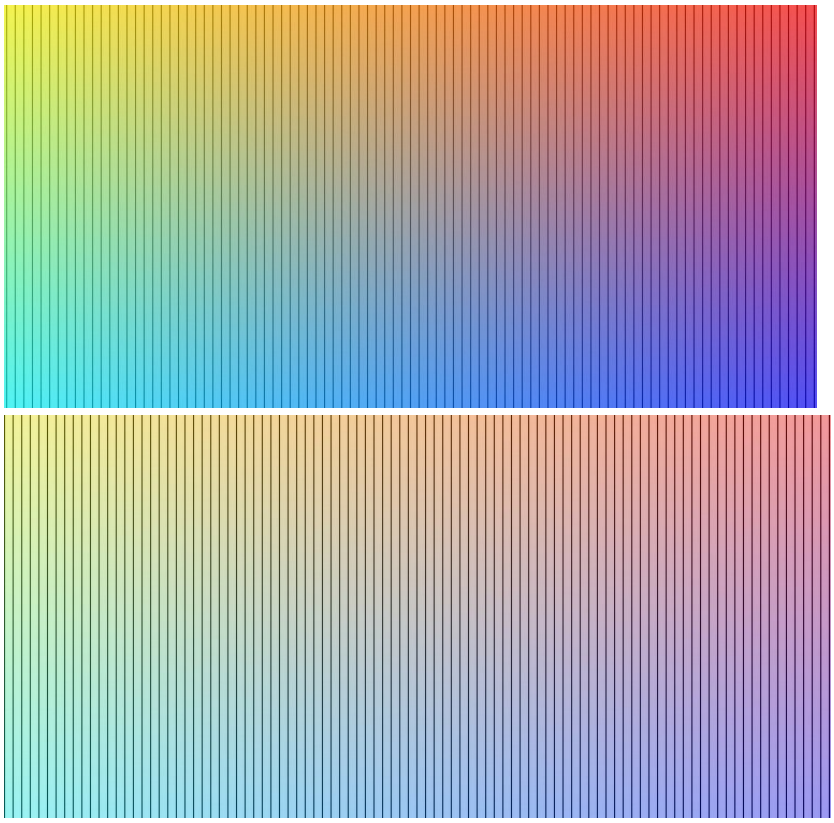
Using BlendTo with a strenght of 40

*© 2023 Theo Gottwald * https://www.smart-package.com*



Usiong different settings for strength, you can decide how strong each IR will be present in the result.

*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

ANA.BlendTo[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 - Source A Image-Register Number, if omitted, IR 0 will be used.
P2 - Source B Image-Register Number, if omitted, IR 0 will be used.
P3 - opt. Destination Image-Register Number, if omitted, IR 2 will be used.
P4 - opt. Strength of Blending, 0-255 (0-&HFF) 0 -> Source A is maximum strong, 255 - Source B is maximum strong, if omitted &H80 (128 dez) will be used.

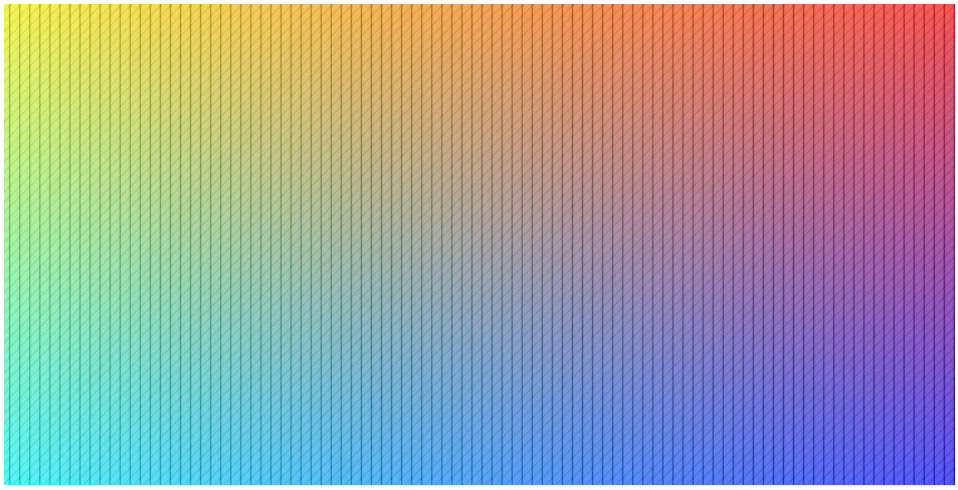
The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.BlendTo-Sample
'***********************************
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Mix|0|2|7
ANA.Show|2!

ANA.New|5|800|400
ANA.Fill|5|&H000000|4|&HFFFFFF
ANA.Show|5!
ANA.New|6|800|400
ANA.Fill|6|&H000000|2|&HFFFFFF
ANA.Show|6!
ANA.BlendTo|5|6|7|80
ANA.Show|7!

ANA.Show|1!
ANA.BlendTo|7|2|3|90
ANA.Show|3!
ENR.

' Here you can see the result of the Operation.
```



Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.2.4 ANA.Blit

MiniRobotLanguage (MRL)

ANA.Blit

Use GPU to copy or mix Image-Register(s) (IR)

Intention

This command has a lot of Options to copy or mix Image Registers.
It uses the GPU and offers the Option to use several different Logic ways to get the final result.
Using this Command, Source IR and Target IR must have the same Resolution.
If that is not the case, use "`ANA.EqualRes`" to enforce the same resolution on the target-Side.

You can simply copy a IR over another or you can make logical combinations using Blit-Modes 1156

Here are two examples:

You will find much more examples on the Blit-Modes 1156 Page.

Sometimes the result depends on whether you "Blit A on B" or you "Blit B on A"
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Generate two Bitmaps with Colours
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000F1|&H1FFF0F

' Show you the Pictures
ANA.Show|0!
ANA.Show|1!

' Set Blitmode to 5
$$BLI=5

' Make a Copy of the Original IR for later use
ANA.Copy|1|2

' Blit 0 over 2 using Style "5"
ANA.blit|0|2|$$BLI

' Show you IR 2 (Result)
ANA.Show|2!

' Copy new Target to IR 2
ANA.Copy|0|2

' Blit 1 over 2 using same Blitmode
ANA.blit|1|2|$$BLI

' Show you IR 2
ANA.Show|2!
ENR.
```

Syntax

ANA.Blit[|P1]

Parameter Explanation

P1 - opt. Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Image-Register number, if omitted, IR 0 will be used.
P3 - opt. Blit-Mode (default is 4)

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'*********************************
' ANA.Blit-Sample
'*********************************
ANA.New|0|800|400
ANA.vgradient|0|&H00FF00|&HFFFF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Show|0!
$$BLI=2
ANA.blit|0|0|$$BLI
ANA.Show|0!
ENR.
```

The Command can blit over itself, see Sample.

Remarks

-

Limitations:

-

See also:

- Blit-Modes 1156

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.2.5 ANA.BlitCol

MiniRobotLanguage (MRL)

ANA.BlitCol

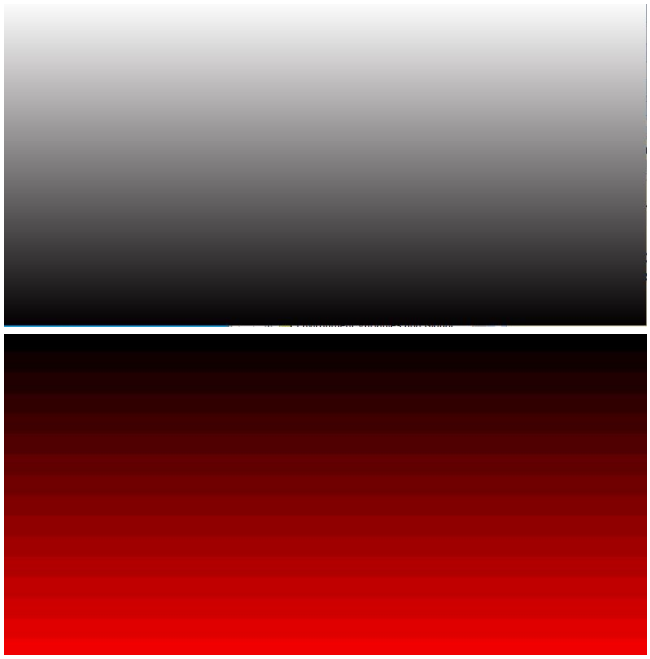
Mix an Image-Register (IR) with a solid color using Blitmodes

Intention

This command will generate an internal IR that is completely filled with a specified color.
It will then Blit this IR over the Source IR and store the result in destination.

1. Blit a defined Color over a IR
See the result below.

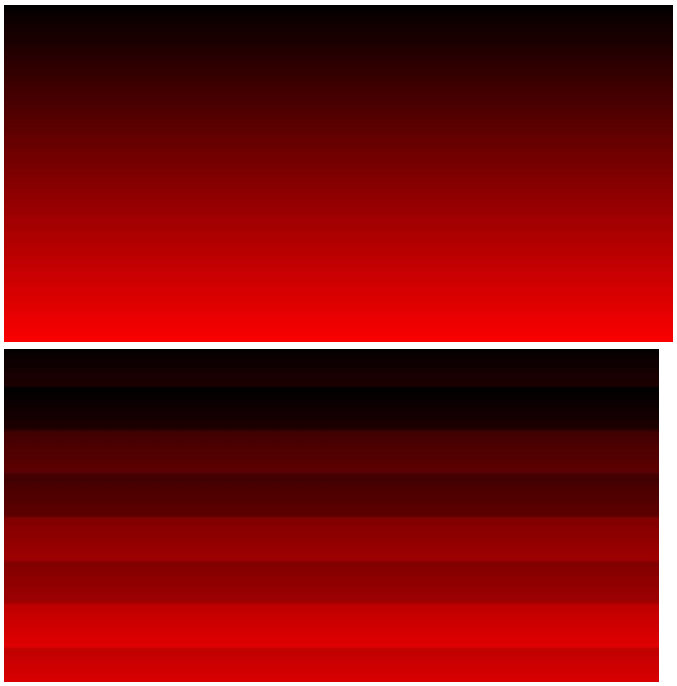
```
ANA.New|0|800|400
ANA.vgradient|0|&HFFFFFF|&H000000
ANA.Show|0!
'  Source|Target|Color|Blitmode
ANA.BlitCol|0|1|&HFFFF0F|2
ANA.Show|1!
ENR.
```



If we change the Color here to:

```
ANA.BlitCol|0|1|&HFFFF00|2
' We get less stripes
```


© 2023 Theo Gottwald * https://www.smart-package.com

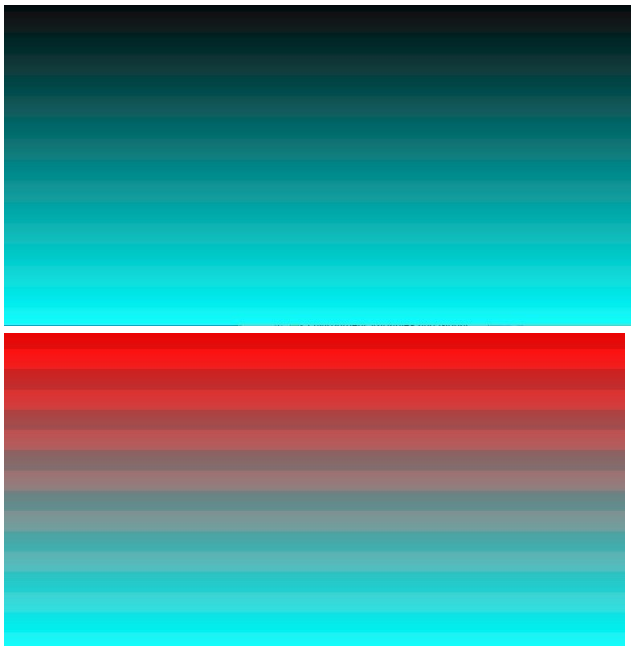


```
ANA.BlitCol|0|1|&HFFFF00|2
' We get less stripes
```

```
ANA.BlitCol|0|1|&HFFFF20|2
'We get more stripes
```

2. Use different Blitmodes to get different results.

Note that we used Blitmode 2. Depending on the Blitmode you get different result.
Here a result using Blitmode 5 and Blitmode 7.



```
ANA.New|0|800|400
ANA.vgradient|0|&HFFFFFF|&H000000
ANA.Show|0!
'  Source|Target|Color|Mixmode
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.BlitCol|0|1|&HFFFF10|5
ANA.Show|1!
ENR.
```

Syntax

ANA.BlitCol[|P1][|P2][|P3][|P4]

Parameter Explanation

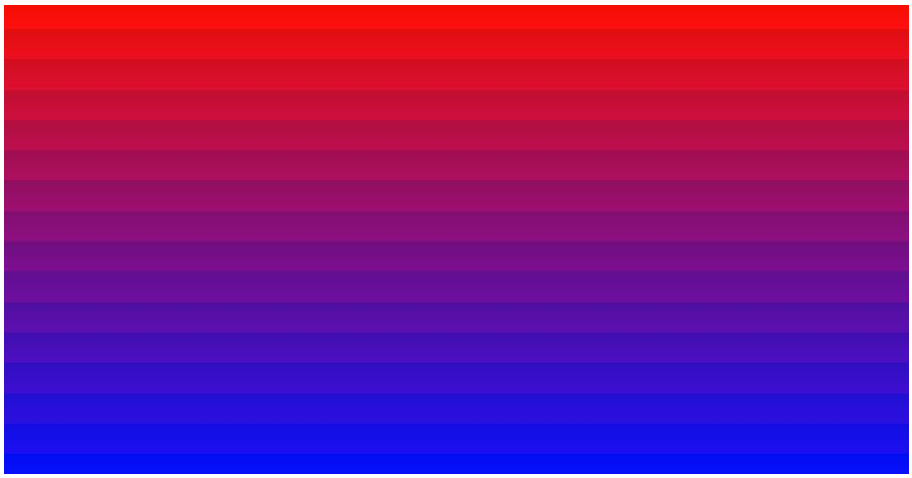
P1 - opt. Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Image-Register number, if omitted, **P1** will be used.

P3 - opt. Color, if omitted &H808080 (Grey Color) is used.
P4 - opt. **Blitmode, if omitted Blitmode 12 "MergeNotSource" is used.**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.BlitCol-Sample
'**********************************
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.Show|0!
'  Source|Target|Color|Mixmode
ANA.BlitCol|0|1|&HFF0010|7
ANA.Show|1!
ENR.

' Here you can see a part of the result of the Operation.
```



© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.2.6 ANA.BlitShift

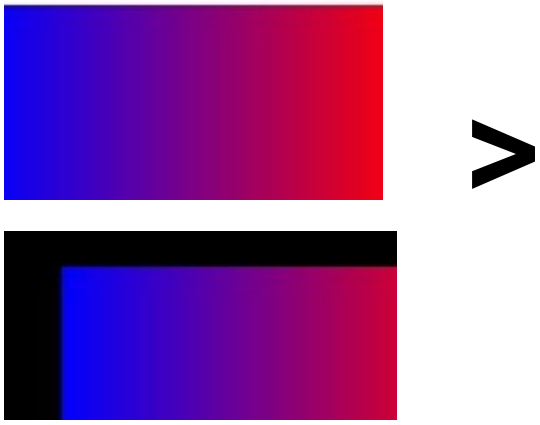
MiniRobotLanguage (MRL)

ANA.BlitShift

Move Image-Register (IR) Content left, right, up or down

Intention

This command will move the content of an IR up, down, left or right by a specified number of pixels.
The underlying IR can be automatically increased in Size - or not.

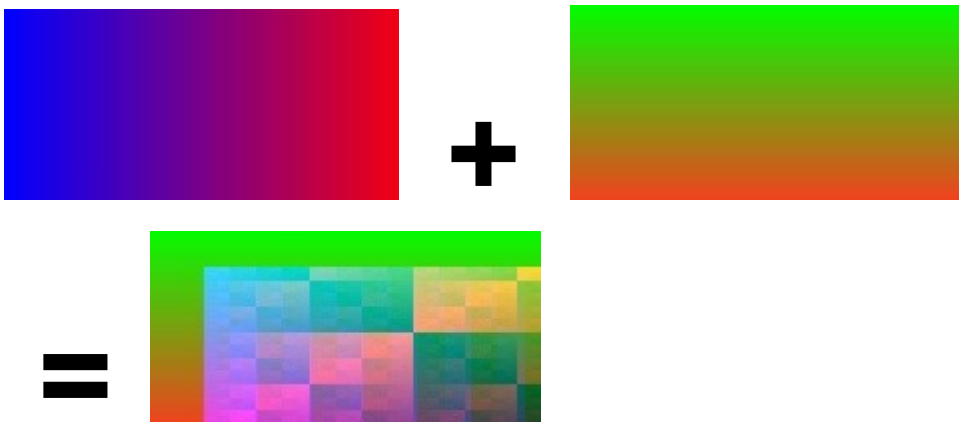


1. BlitShift with Source IR -> another IR
This will BlitShift the specified IR into another IR. Depending on the used Blitmode, the Content of the other IR will be part of the result.

```
ANA.New|0|200|100
ANA.hgradient|0|&HFF0000|&H1000F4
ANA.New|1|200|100
ANA.vgradient|1|&H00FF00|&H1F40F4
VAN.$$LOY=20
VAN.$$LOX=30
ANA.show|0!
ANA.show|1!
ANA.bsh|0|1|$$LOX|$$LOY|7
ANA.show|1!
ENR.
```

This is the above example, using Blitmode 7

© 2023 Theo Gottwald * https://www.smart-package.com



2. BlitShift over own IR
The result will vary depending on the used Blitmode.

```
ANA.New|0|200|100
ANA.hgradient|0|&HFF0000|&H1000F4
VAN.$$LOY=20
VAN.$$LOX=30
ANA.show|0!
ANA.bsh|0|0|$$LOX|$$LOY|7
ANA.show|0!
ENR.
```

2. BlitShift a 2D-Logi into a 3D-Logo
The result will vary depending on the Source Logo.

```
$$PAB=?exeloc\Pics\P3.jpg

$$BOR=10
CAL.$$BOB=$$BOR/2

IEF.$$PAB
  ANA.Load|1|$$PAB
  ANA.Show|1!
ELS.
  MBX.File does not exist:$crlf$$$PAT
EIF.

ANA.GetRes|1|$$XPO|$$YPO
CAL.$$XPA=$$XPO+$$BOR
CAL.$$YPA=$$YPO+$$BOR
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Make a Resized Version of P3
ANA.New|2|$$XPA|$$YPA|1
ANA.New|3|$$XPA|$$YPA

' Now mix Shiftzed and Resized Version
ANA.bsh|3|1|$$BOB|$$BOB|2
ANA.Invert|1
ANA.MixTar|1|2|2
ANA.Invert|2
ANA.Show|2!
ENR.
```

Syntax

ANA.BlitShift[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - opt. Source Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Destination Image-Register number, if omitted, IR 0 will be used.
P3 - opt. X-Shift in Pixel, if omitted 0 is used
P4 - opt. Y-Shift in Pixel, if omitted 0 is used
P5 - opt. Blitmode

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

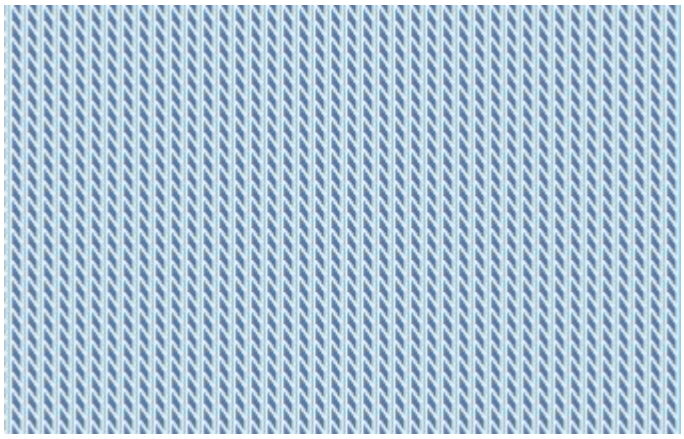
```
'***********************************
' ANA.BlitShift-Sample
' In this Sample we use a recorded 8x8 Pattern, that is filled into an IR
' and the blitted over a Picture. With the right "Shift" it will turn any a fitt
'***********************************
' Load Picture
ANA.Load|1|TP8b.bmp
ANA.Mask|1
```

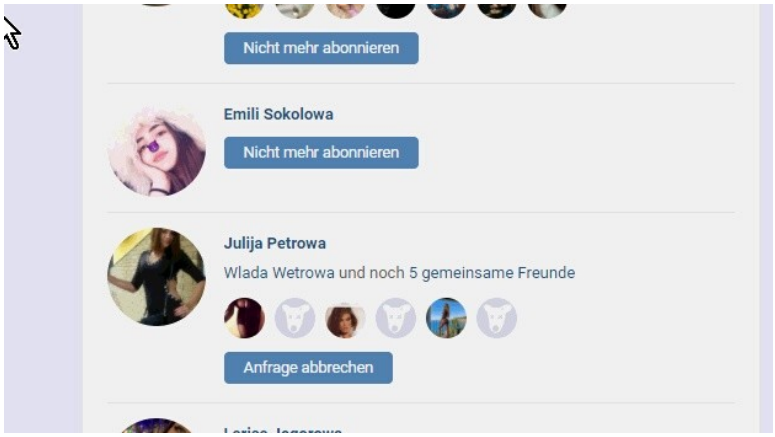

© 2023 Theo Gottwald * https://www.smart-package.com

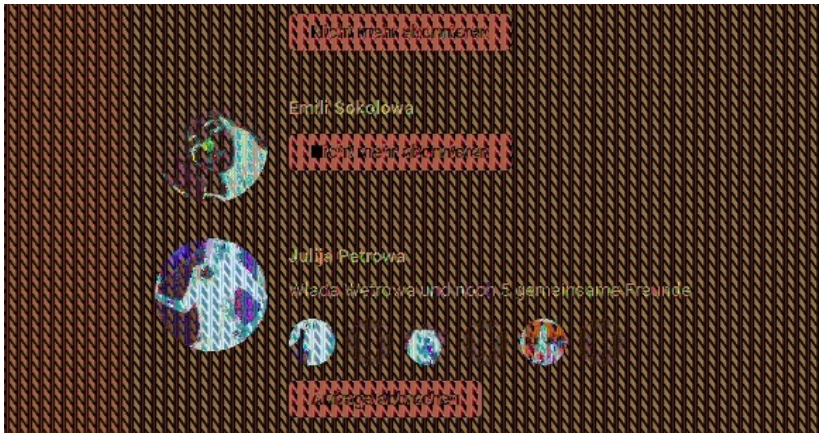
```
ANA.rcd|1|&H0|&H000101

' Load Fullpic to 0
ANA.Load|0|Soz_100.png
ANA.Mask|0
ANA.rcd|0|&H0|&H000101

ANA.copy|0|2
ANA.Show|0

STW.ct|Powergraph|debug
HTV.$$MYW
ANA.patternfillHwn|1|0|&HF0F0F0|$$MYW
' 0 ist nun das Multipattern
ANA.copy|2|1

VAN.$$LOY=0
VAN.$$LOX=2
ANA.show|0!
ANA.show|1!
ANA.bsh|0|1|$$LOX|$$LOY|7

ANA.show|1
' Find black pattern
ANA.scc|1|&H0|8|0
MMV.
MBX.!
ENR.

' Here you can see typical operation.
```

Picture 1: This is the recorded Pattern **Picture 2 This is the Fake-Site where we recorded the pattern** **As a result of the Blit-Operation, the place where the pattern fits turns into a black Square.**

© 2023 Theo Gottwald * https://www.smart-package.com







As a result of the Blit-Operation with Blit-Mode 7, the place where the pattern fits turns black, see the black square in the Middle.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3 Change Resolution

These Operations are used to get and set the IR graphical resolution.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.1 ANA.CutRect

MiniRobotLanguage (MRL)

ANA.CutRect

Crop Borders from IR

Intention

This command cut a specified amount of Lines away from all 4 sides of an IR. I can be used to cut the Borders.

Use ANA.CutRect together with ANA.FromClipboard to store Clipboard-Content in IR 0 and display it
This will use IR 0 and also show the Contents.

```
$$PAF=?exeloc\Pics
$$FIL?$$PAF\P4.jpg
ANA.Load|0|$$FIL
' The CutRect Command is optional, it can be used to "Cut the borders" of an IR
ANA.CutRect|0|150|160|400|200
ANA.ToClipboard|0
' Picture is now in the Windows Clipboard
ENR.
```

ANA.CutRect is a bit special. Assume you have a picture of 640x400 and you specify:

```
ANA.CurRect|0|50|50|640|400
```

Then you may get a black Border of size 50 on the right side. This is because the limitation is the original Picture-Size.
If you specify instead a Target-Rectangle-Size that is larger then the original size, you will get a perfect match without a black Border.

```
ANA.CutRect|0|50|50|6400|4000
```

Of course you can specify exactly what you want, in such case please note that the X- and Y-Size is reduced by the values of how much you cut out at the upper and left side.
So this line would also do it:

```
ANA.CutRect|0|50|50|590|350
```

Syntax

ANA.CutRect[|P1][|P2|P3|P4|P5][|P6]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - opt. **Image-Register number** for Source and Target IR, if omitted, IR 0 will be used.

' If given, you must specify all 4 Rect-Parameters
P2 - Left-Border of Image to copy
P3 - Upper/Top-Border of Image to copy
P4 - Right-Border of Image to copy. If too large, will be replaced by Image maximum size.
P5 - Bottom-Border of Image to copy. If too large, will be replaced by Image maximum size.

P6 - Blit-Constant Parameter, if omitted Standard will be used. Details see Blit-Modes 1156

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.ToClipboard-Sample
'**********************************
$$PAF=?exeloc\Pics
$$FIL=$$PAF\P4.jpg
ANA.Load|0|$$FIL
ANA.CutRect|0|150|160|4000|2000
ANA.ToClipboard|0
ANA.Show|0!
ENR.

'**********************************
' ANA.ToClipboardRect-Sample
'**********************************
$$PAF=?exeloc\Pics
$$FIL=$$PAF\P4.jpg
ANA.Load|0|$$FIL
ANA.ToClipboardRect|0|150|88|290|189|2
ANA.FromClipboard|1
ANA.Show|1!
ENR.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- Blit-Modes 1156
- ANA.FromClipboard 1410
- CLP. - Clipboard - Operation 2009

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.2 ANA.EqualRes

ANA.EqualRes - Force equal Resolution for IR's ***Previous*** 1088 ***Top*** 41 ***Next*** 1109

MiniRobotLanguage (MRL)

ANA.EqualRes

Force equal Resolution between Image-Register's (IR)

Intention

This command will enforce that IR **P2** will have the same resolution as IR **P1**.
If **P2** does not exist, it will be created. If it exists, it will be stretched/resized to the Size of **P1**.

Generate two IR's with different resolution and then make the resolution equal including stretch
The following Script shows the use of `ANA.EqualRes`.

```
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4

ANA.New|1|400|800
ANA.vgradient|1|&HFF0000|&H10FFF4

ANA.Show|0!
ANA.Show|1!
' Make Resolution from P2 to the resolution from P1.
ANA.EqualRes|0|1
DMP.6
ANA.Show|1!
ENR.
```

Generate new IR with the Resolution of the original IR
This will create a new, empty (black) IR with the resolution of the IR P1.

```
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4
ANA.Show|0!
ANA.EqualRes|0|1
DMP.6
ANA.Show|1!
ENR.
```

Leaving away P2

If **P2** is omitted, it will be assumed to be **P1**+1

```
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4
ANA.Show|0!
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
ANA.EqualRes|0
DMP.6
ANA.Show|1!
ENR.
```

Syntax

ANA.EqualRes|P1[|P2]

Parameter Explanation

P1 **-** ***opt.*** Image-Register number, if omitted, IR 0 will be used.
P2 **-** ***opt.*** Image-Register number, this IR will be stretched to the resolution of **P1**. If it does not exist it will be created. If omitted, **P1**+1 will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.EqualRes-Sample
'***********************************
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4

ANA.New|1|400|800
ANA.vgradient|1|&HFF0000|&H10FFF4

ANA.Show|0!
ANA.Show|1!
' Make Resolution from P2 to the resolution from P1.
ANA.EqualRes|0|1
DMP.6
ANA.Show|1!
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.3 ANA.GetRes

ANA.GetRes - Get Resolution of specified IR Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.GetRes

Get X- and Y- Resolution and the total number of Bytes of the specified Image-Register (IR)

Intention

This command will get the X- and the Y- Resolution of the specified IR.
It will also get the total number of Bytes of this IR.

1. Use ANA.GetRes with the Stack

This will use `ANA.GetRes` with the Stack.

```
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|0
ANA.GetRes|0
DMP.6
MBX.!
ENR.
```

The Stack will look like this:

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->LongInt      :624

Stackpos: 001 ->LongInt      :361

Stackpos: 002 ->LongInt      :901056

Stackpos: 003 ->LongInt      :1

-----------------------------------
```

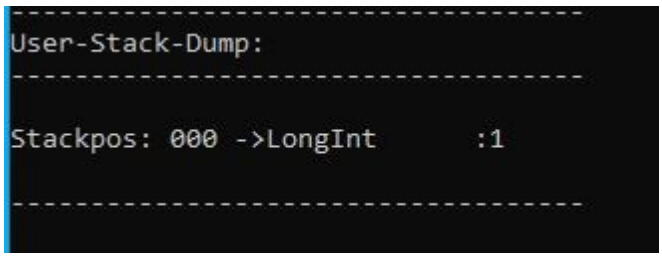
On TOS is the X-Resolution, followed by the Y-Resolution. Then the total Number of Bytes in the specified IR. This generally X+Y*4.

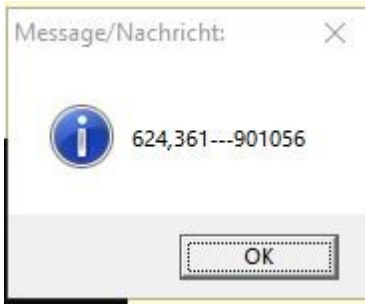
2. Use ANA.GetRes with Variables

© 2023 Theo Gottwald * https://www.smart-package.com

This will directly place the results in Variables. On the TOS there will be only a "1", or a "0" in case of error.

```
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|0
ANA.GetRes|0|$$XPO|$$YPO|$$SIZ
DMP.6
MBX.$$XPO,$$YPO---$$SIZ
MBX.!
ENR.
```





Syntax

ANA.GetRes|[|P1][|P2][|P3][|P4]

Parameter Explanation

***P1* - *opt.* Image-Register number, if omitted, IR 0 will be used.**
P2* - *opt. *Variable that will receive the X-Resolution*, if omitted, TOS will be used.
P3* - *opt. *Variable that will receive the Y-Resolution*, if omitted, TOS will be used.
P4* - *opt. *Variable that will receive the Size of the IR in Bytes*, if omitted TOS will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.GetRes-Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' Will get Desktop-Resolution
STW.ct|Progman|Program Manager
ANA.Snap|0
ANA.GetRes|0|$$XPO|$$YPO|$$SIZ
DMP.6
MBX.$$XPO,$$YPO---$$SIZ
MBX.!
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.4 ANA.GetXYFromBytePos

MiniRobotLanguage (MRL)

ANA.GetXYFromBytePos

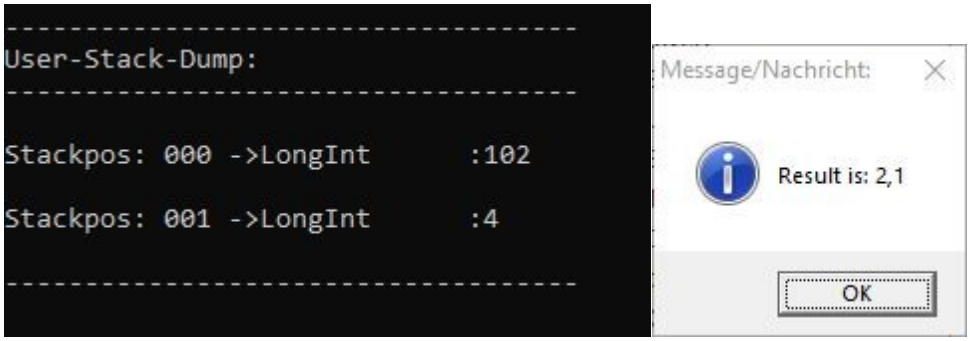
Get X- and Y- Position from the sequential Byte-Position-Number

Intention

This command will get the X- and the Y- Position from a Byte-Position of a specified IR.
The result is 0-based, means that the upper left coordinate is 0,0.
You also need to know that each Pixel has 4 Bytes (24-Bit Color+1 Byte Alpha).

For example, if IR-0 has a resolution of 100x400

```
ANA.New|0|100|400
ANA.GetXYFromBytePos|0|409|$$XPO|$$YPO
DMP.6
MBX.Result is: $$XPO,$$YPO
ENR.
```



This is the result from the above Script. On the Stack you can also see the "Pixelposition" that is 102.

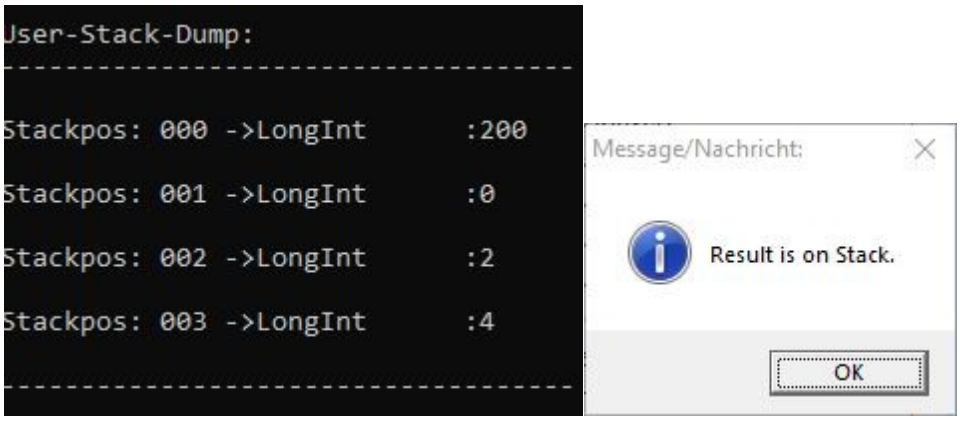
1. Use ANA.GetXYFromBytePos with the Stack

This will use `ANA.GetXYFromBytePos` with the Stack.

```
ANA.New|0|100|400
ANA.GetXYFromBytePos|0|801
DMP.6
MBX.Result is on Stack.
ENR.
```

The Stack will look like this:

© 2023 Theo Gottwald * https://www.smart-package.com



On TOS is the the "Pixelposition", followed by the X-Position, followed by the Y-Position.

Syntax

ANA.GetXYFromBytePos|[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 – Image-Register number, if omitted, IR 0 will be used.
P2 – Byte-Position to convert.
P3 – ***opt.*** Variable that will receive the X-Position, if omitted, TOS will be used.
P4 – ***opt.*** Variable that will receive the Y-Position, if omitted TOS will be used.

The Command will leave the Pixel-Position on the TOS.

Example

```
'**********************************
' ANA.GetXYFromBytePos-Sample
'**********************************
ANA.New|0|100|400
ANA.GetXYFromBytePos|0|801
DMP.6
MBX.Result is on Stack.
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.5 ANA.GetXYFromPixelPos

MiniRobotLanguage (MRL)

ANA.GetXYFromPixelPos

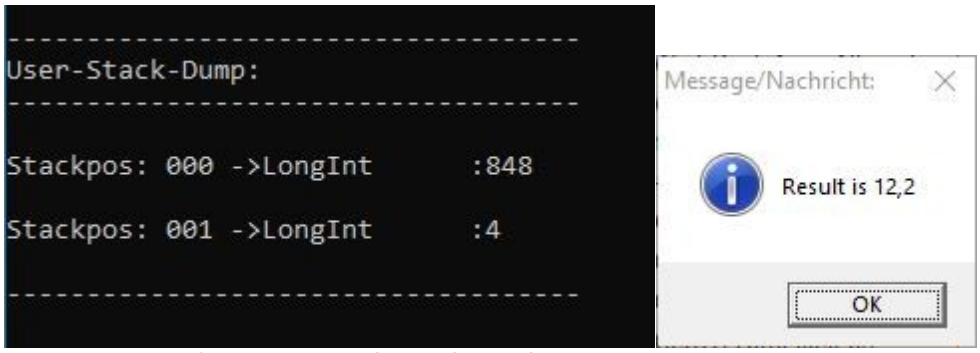
Get X- and Y- Position from the sequential Pixel-Position-Number

Intention

This command will get the X- and the Y- Position from a sequential-Pixel-Position Number of a specified IR.
The result is 0-based, means that the upper left coordinate is 0,0.
You also need to know that each Pixel has 4 Bytes (24-Bit Color+1 Byte Alpha).

For example, if IR-0 has a resolution of 100x400

```
ANA.New|0|100|400
ANA.GetXYFromPixelPos|0|212|$$XPO|$$YPO
DMP.6
MBX.Result is $$XPO,$$YPO
ENR.
```



This is the result from the above Script. On the Stack you can also see the "Byteposition" that is 848.
The last number, 4 is from the ANA.New-Command.

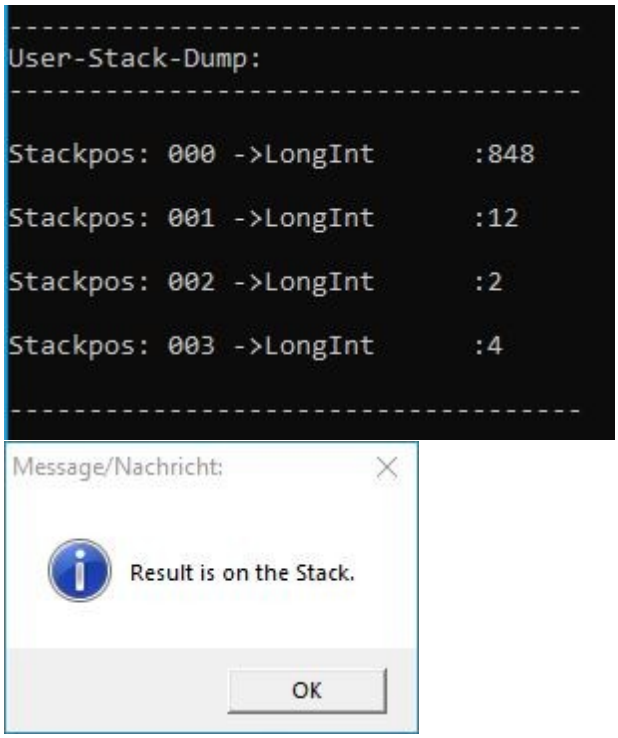
1. Use ANA.GetXYFromPixelPos with the Stack

This will use `ANA.GetXYFromPixelPos` with the Stack.

```
ANA.New|0|100|400
ANA.GetXYFromPixelPos|0|212
DMP.6
MBX.Result is on the Stack.
ENR.
```

The Stack will look like this:

© 2023 Theo Gottwald * https://www.smart-package.com



On TOS is the the "Byteposition", followed by the X-Position, followed by the Y-Position. The last number, 4 is from the ANA.New-Command.

Syntax

ANA.GetXYFromPixelPos|[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 - Image-Register number, if omitted, IR 0 will be used.
P2 - ***Pixel***-Position to convert.
P3 - ***opt.*** Variable that will receive the X-Position, if omitted, TOS will be used.
P4 - ***opt.*** Variable that will receive the Y-Position, if omitted TOS will be used.

The Command will leave the Byteposition on the TOS.

Example

```
'**********************************
' ANA.GetXYFromPixelPos-Sample
'**********************************
ANA.New|0|100|400
ANA.GetXYFromPixelPos|0|212
DMP.6
MBX.Result is on the Stack.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.6 ANA.Grab

MiniRobotLanguage (MRL)

ANA.Grab

Copy a smaller Portion from an IR starting at 0,0 to another IR removing empty Lines (empty -> just Background)

Intention

This command removes empty Lines at top and left side and cuts out a specified amount of Lines away from Right and Bottom sides of IR P1. The result will be moved to **P2.**
`ANA.Grab` will also automatically move theSource Rectangle up direction 0,0, and remove all empty lines.

Use ANA.Grab together with ANA.FromClipboard to store Clipboard-Content in IR 0 and display it
This will use IR 0 and also show the Contents.

```
$$PAF=?exeloc\Pics
$$FIL?$$PAF\P4.jpg
ANA.Load|0|$$FIL
' The Grab Command is optional, it can be used to "Cut the borders" of an IR bef
ANA.Grab|0|1|160|400|3
ANA.ToClipboard|1
' Picture is now in the Windows Clipboard
ENR.
```

Syntax

ANA.Grab[|P1][|P2|P3|P4|P5][|P6]

Parameter Explanation

***P1* - Image-Register number for Source IR**
***P2* - Image-Register number for Target IR**

P3* - *Right-Border of Image to copy
P4* - *Bottom-Border of Image to copy

***P5* - *Blit-Constant Parameter*, if omitted Standard will be used. Details see Blit-Modes 1156**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' ANA.Grab-Sample
'**********************************
$$PAF=?exeloc\Pics
$$FIL?$$PAF\P4.jpg
ANA.Load|0|$$FIL
ANA.Grab|0|1|160|400|3
ANA.Show|1!
' Picture is now in the Windows Clipboard
ENR.

'**********************************
' ANA.ToClipboardRect-Sample
'**********************************
$$PAF=?exeloc\Pics
$$FIL=$$PAF\P4.jpg
ANA.Load|0|$$FIL
ANA.ToClipboardRect|0|150|88|290|189|2
ANA.FromClipboard|1
ANA.Show|1!
ENR.


'**********************************
' ANA.Grab-Sample, used for Text recognition
'**********************************
ANA.New|0|800|400
ANA.Fill|0|&Hffffff
' Backgroundcolor
$$LAA=-2
' Fontname
$$LAC=Arial
$$TXT=add
VAN.$$COL=0
LEN.$$TXT|$$LET
ANA.PrintAt|0|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|9|$$LAC|1
ANA.Grab|0|1|(9+($$LET*8))|9
ANA.GetRes|1|$$XRE|$$YRE
DBP.Res.$$XRE,$$YRE
'VAR.$$PAT=?pattern\P1.bmp
'ANA.Load|1|$$PAT
'ANA.Show|1!
' Flag "A" can be set to 1,2 or 3
SFP.fa|1
SFP.Snap_Border|0
STW.ct|Chrome_WidgetWin| - Brave
SWP.0,0|1414,966|T
ANA.Snap|0
ANA.fpa|0|1|40|150|1
' This command will put the results of the search on to the Stack.
' Number of found Patterns is on TOS
POP.$$POA
DBP.We got $$POA Results
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
JIZ.$$POA|enx
FOR.$$LN1|1|$$POA
  POP.$$POB
  MMV.w|$$POB
  PAU.4
NEX.
:enx
ENR.
```

Remarks

-

Limitations:

-

See also:

- Blit-Modes 1156
- ANA.CutRect 1186
- ANA.FromClipboard 1410
- CLP. - Clipboard - Operation 2009

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.3.7 ANA.ResizeTo

MiniRobotLanguage (MRL)

ANA.ResizeTo

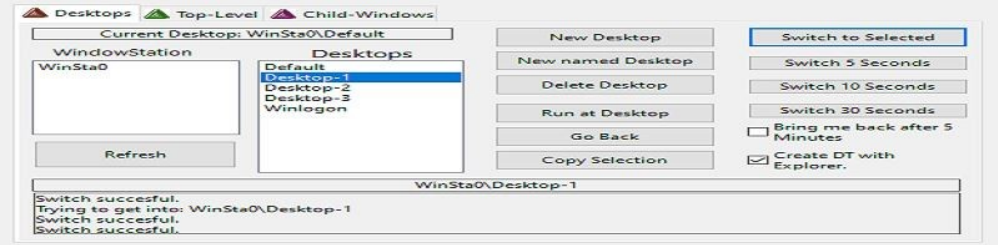
Resize existing IT to a new IR

Intention

This command can be used to resize an IR into a new IR with a specified resolution.
It will create a new IR with a specified resolution from a already existing IR.
It will "Stretch" the existing IR into the new IR's resolution.

1. Create new IR and copy the resized Content of another IR
As you can see below, the Resizing will not take any Aspect-Ratio into account, but use the full X and Y Size of the new IR.

```
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|1
ANA.ResizeTo|1|0|800|200
ANA.show|0!
ENR.
```

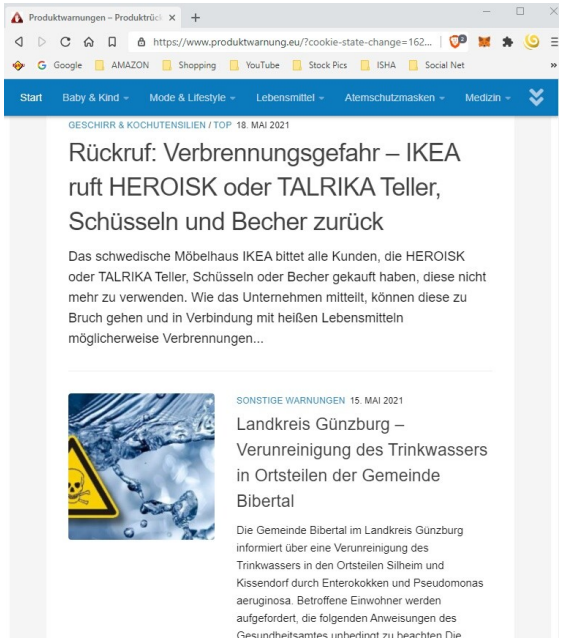


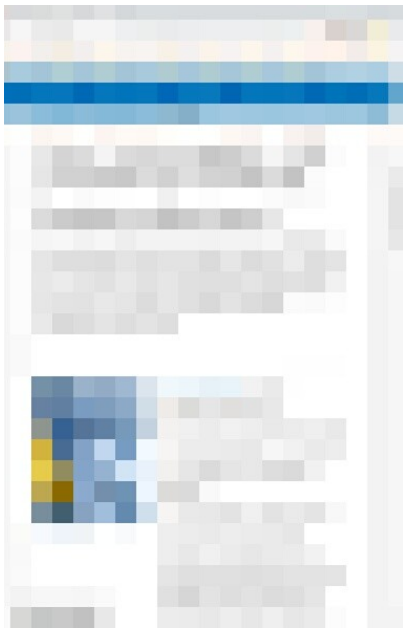
2. Resize IR to iself
You can also specify the Target IR with the same IR as the Source, then the IR itself will be resized.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.ResizeTo|0|0|20|30
ANA.ResizeTo|0|0|400|600
ANA.Show|0!
ENR.
```

Source **Result**

 *© 2023 Theo Gottwald * https://www.smart-package.com*





Syntax

ANA.ResizeTo|P1|P2|P3|P4

Parameter Explanation

P1 **- *Source* Image-Register Number for the IR that is to be resized.**
P2 **- *Target* Image-Register Number can be identical to the Source Register Number**
P3 **- *opt. X-Resolution* for the new IR Content (default is Target x-resolution)**

*© 2023 Theo Gottwald * https://www.smart-package.com*

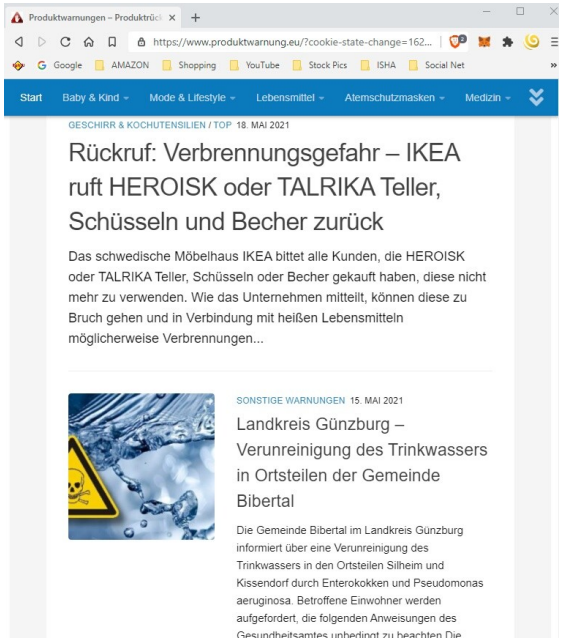
***P4 - opt. Y-Resolution* for the new IR Content (default is Target y-resolution)**

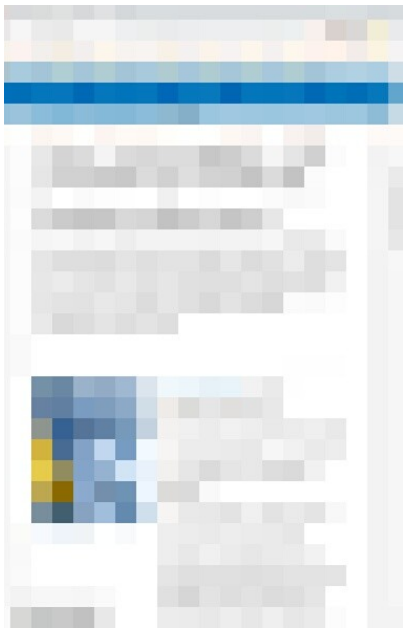
The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.ResizeTo-Sample
'**********************************
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.ResizeTo|0|1|20|30
ANA.ResizeTo|1|0|400|600
ANA.Show|0!
ENR.


' Here you can see the Original Window and the result of the
Operation.
```


© 2023 Theo Gottwald * https://www.smart-package.com





Remarks

This Functionality is also included in the "`ANA.New`"-Command. As it would be hard to find, we have sorted the Parameters here in another way and made this extra-command.

Limitations:

-

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4 Color Table Operations

These Operations work with the "Color Table", this is a sorted list with all colors in the IR.
This Color-Table is generated using

```
ANA.AnalyzeColor|0|1
```

In fact there are 2 Color-Tables:
0 - Color-Sort (Col_Srt) - This table contains the Colors sorted by number of Pixel per Color.
The Top-Entry is the color with the most Pixels. The bottom Entry is the color with the least number of Pixels.
1 - Color-Background (Col_Bgl) - This table is sorted by similarity to the Color that has most Pixel, which is assumed to be the Background Color.
Here the Top-Entry contains the Background Color and then any further Entry is sorted by Similarity to this Color.

The Col_Bgl Table can be sorted to another Color using:

```
ANA.Sort_Similar_To|&H26592A|$$MSK
```

Using the

```
ANA.AnalyzeColor|0|1

' These are some further commands that can be used after t
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
```

you can work with the list entries. For example you can remove colors from the list and the IR.

Remove Green Background

To do this, we first Analyze the Picture to generate the Color-Table.
Secondly we sort the Color Table in a way that its sorted by Similarity with the Background Green Color.
Third we remove the Entries that are similar to the Green Color.
The numbers that are used in detail are a bit of trial and error depending on the picture.

© 2023 Theo Gottwald * https://www.smart-package.com

->

```
' This Script will remove a green backgroiund
' using special sorting functions.
' The Script is fast and will only take few seconds
VAF.$$PIA=?exeloc\Sara_Green.jpg
ANA.Load|0|$$PIA
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
' Sort the Color Table with Similarity to the Background Color
ANA.Sort_Similar_To|$$COL|$$MSK
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.RemoveIndexColors|0|39300|$$CNT|&HFFFFFF
ANA.Show|0!
ENR.
```

```
010 0000000000000323 0001 VAN.$$TIM=#time#
011 0000000000000109 0001 VAF.$$PIA=?exeloc\Sara_Green.jpg
012 0000000000334747 0001 ANA.Load|0|$$PIA
013 0000000001419195 0001 ANA.AnalyzeColor|0|1
014 0000000000000008 0001
015 0000000000000187 0001 VAR.$$COL=&H30FF30
016 0000000000000042 0001 VAR.$$MSK=&HFFFFFF
017 0000000000000004 0001
018 0000000000000440 0001 ANA.GetColorCount|0|$$CNT
019 0000000000001425 0001 DBP.$$CNT
020 0000000000666471 0001 ANA.ssto|$$COL|$$MSK
021 0000000835176274 0001 ANA.Show|0!
022 0000000133032400 0001 ANA.removeindexcolors|0|39300|$$CNT|&HFFFFFF
023 0000000000000008 0001
024 0000000000000000 0001 DMP.1
025 0000000000000000 0000 ANA.Show|0!
```

The whole process is 1-2 seconds.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
'VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
GSB.Reanalyze
'ANA.GetSortedColor|0|$$LOP|$$COL
ANA.removeindexcolors|0|300|$$CNV|0
ANA.removeindexcolors|0|$$CNV|$$CNT|-1
GSB.Reanalyze
ANA.removeindexcolors|0|$$CNV|$$CNT|-1
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.

:Reanalyze
ANA.AnalyzeColor|0|1
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
CAL.$$CNV=$$CNT-400
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

While the artefacts on the white are completely removed, the artefacts in the dark are somehow visible, because we replaced the "dark grey" with black color.
Anyway using this commands you can remove all sorts of dirt and JPG-Artefacts out of pictures.

Speed-Dump of the Script.

```
010 0000000000000312 0001 VAN.$$TIM=#time#
011 0000000000000105 0001 VAF.$$PIA=?exeloc\BNB.jpg
012 0000000000000004 0001
013 0000000000000057 0001 VAF.$$PIB=?exeloc\V1.png
014 0000000001181295 0001 ANA.Load|0|$$PIA
015 0000000000000092 0001 GSB.Reanalyze
016 0000000000000005 0001
017 0000000027183868 0001 ANA.removeindexcolors|0|300|$$CNV|0
018 0000000000903861 0001 ANA.removeindexcolors|0|$$CNV|$$CNT|-1
019 0000000000000092 0001 GSB.Reanalyze
020 0000000009890444 0001 ANA.removeindexcolors|0|$$CNV|$$CNT|-1
021 0000000000000379 0001 CAL.$$TIM=#time#-$$TIM
022 0000000000001155 0001 DBP.Used Time in Seconds: $$TIM
023 0000000000000000 0001 DMP.1
024 0000000000000000 0000 ANA.Save|0|$$PIB
025 0000000000000000 0000 ANA.Show|0!
026 0000000000000000 0000 ENR.
027 0000000000000000 0000
028 0000000000000006 0002 :Reanalyze
029 0000000000931817 0002 ANA.AnalyzeColor|0|1
030 0000000000000724 0002 ANA.getcolorcount|0|$$CNT
031 0000000000000259 0002 ANA.getbgcolor|0|$$BGC
032 0000000000000238 0002 CAL.$$CNV=$$CNT-400
033 0000000000000051 0002 RET.
```

The whole process is 1-2 seconds.

*© 2023 Theo Gottwald * https://www.smart-package.com*

With a few changes, you can see in detail how the Script works:

```
' Here we have changed the script a bit to show
' how the commands paint the colors.
' Green -> would be painted in white
' red -> would be painted in black
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
'VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
GSB.Reanalyze
ANA.removeindexcolors|0|300|$$CNV|&HFF0000
GSB.Reanalyze
ANA.removeindexcolors|0|$$CNV|($$CNT-20)|&H00FF00
ANA.removeindexcolors|0|($$CNT-20)|$$CNT|&HFFFFFF
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Reanalyze
ANA.AnalyzeColor|0|1
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
CAL.$$CNV=$$CNT-400
RET.
```

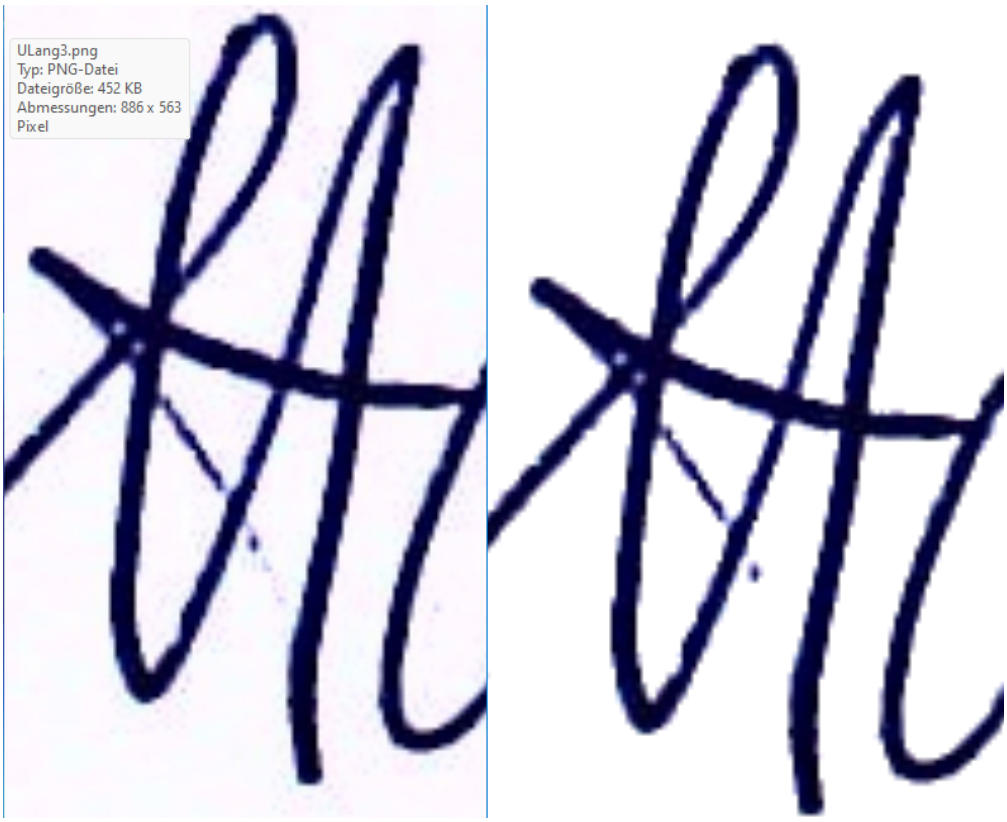
Another Sample. Your Signature.

You have digitized/scanned some Writing - typically your signature - and want to remove the unused Pixels around, just having White around.

Before: Digitzed Writing
Cleanup using the SPR

After:

© 2023 Theo Gottwald * https://www.smart-package.com



Using this Script the Writing will keep its original Color.
Single, unconnected Dots are removed in Stage 2.
Needs in my example 369 Seconds. Below is a faster Version.

```
'SPR Script-file: Cleanup Writing
'Purpose:
'Author: Theo Gottwald
'Creation date: 12-28-2021 at 20:11:00
'============================================================
'#EXE:?path\
'#SPI:ForceWrite

' This Script will clean up the background of a digitized Writing
VAF.$$PIB=?exeloc\ULang5B.png
VAF.$$PIA=?exeloc\ULang3.png
ANA.Load|0|$$PIA
ANA.Show|0!
'ANA.AnalyzeColor|0|1
$$SRC=&HFFF5FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HEADAFE
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HE4F5FF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HC8C0E9
ANA.ReplaceColorT|0|$$SRC|10|$$COL
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Rem_SngPix

ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
'============================================================
' Remove Single unconnceted Dots.
:Rem_SngPix
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|2|($$YSZ-1)
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|2|($$XSZ-1)
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
    JIV.$$PIX=&HFFFFFF|EndLoop

    ANA.getpixelcolor|0|($$XPO+1)|$$YPO|$$PAA
    ANA.getpixelcolor|0|$$XPO|($$YPO+1)|$$PAB
    ANA.getpixelcolor|0|($$XPO-1)|$$YPO|$$PAC
    ANA.getpixelcolor|0|$$XPO|($$YPO-1)|$$PAD
    ANA.getpixelcolor|0|($$XPO+1)|($$YPO+1)|$$PAE
    ANA.getpixelcolor|0|($$XPO-1)|($$YPO+1)|$$PAF
    ANA.getpixelcolor|0|($$XPO+1)|($$YPO-1)|$$PAG
    ANA.getpixelcolor|0|($$XPO-1)|($$YPO-1)|$$PAH

    CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF
    IVV.$$ERG>5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
    :EndLoop
  NEX.
NEX.
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000177 0001 VAF.$$PIA=?exeloc\ULangK.png
014 0000000000000071 0001 VAF.$$PIB=?exeloc\ULangK1.png
015 0000000000284536 0001 ANA.Load|0|$$PIA
016 0000000138827141 0001 ANA.Show|0!
017 0000000000000011 0001
018 0000000000000144 0001 VAR.$$SRC=&HFFF5FF
019 0000000000000037 0001 VAR.$$COL=&HFFFFFF
020 0000000000978706 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
021 0000000000000008 0001
022 0000000000000079 0001 VAR.$$SRC=&HEADAFE
023 0000000000962373 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
024 0000000000000009 0001
025 0000000000000076 0001 VAR.$$SRC=&HE4F5FF
026 0000000018157687 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
027 0000000000000009 0001
028 0000000000000092 0001 VAR.$$SRC=&HC8C0E9
029 0000000000967221 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
030 0000000000000009 0001
031 0000000000000153 0001 VAR.$$TIM=#dtime#
032 0000000000000071 0001 GSB.Rem_SngPix
033 0000000000000000 0000 CAL.$$TMA=#dsince#|i
034 0000000000000000 0000 DBP.Needed: $$TMA Seconds.
035 0000000000000000 0000 DMP.1
036 0000000000000000 0000 ANA.Show|0!
037 0000000000000000 0000 ANA.Save|0|$$PIB
038 0000000000000000 0000 ENR.
039 0000000000000000 0000
040 0000000000000000 0000
041 0000000000000005 0001 :Rem_SngPix
042 0000000000000519 0001 ANA.GetRes|0|$$XSZ|$$YSZ
043 0000000000000324 0001 FOR.$$YPO|2|($$YSZ-1)
044 0000000001267068 0001 PRT.$$YPO/$$YSZ
045 0000000000000446 0001 FOR.$$XPO|2|($$XSZ-1)
046 0000000000000595 0001 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
047 0000000000000234 0001 JIV.$$PIX=&HFFFFFF|EndLoop
048 0000000000000005 0001
049 0000000000000632 0001 ANA.getpixelcolor|0|($$XPO+1)|$$YPO|$$PAA
050 0000000000000505 0001 ANA.getpixelcolor|0|$$XPO|($$YPO+1)|$$PAB
051 0000000000000481 0001 ANA.getpixelcolor|0|($$XPO-1)|$$YPO|$$PAC
052 0000000000000477 0001 ANA.getpixelcolor|0|$$XPO|($$YPO-1)|$$PAD
053 0000000000000555 0001 ANA.getpixelcolor|0|($$XPO+1)|($$YPO+1)|$$PAE
054 0000000000000549 0001 ANA.getpixelcolor|0|($$XPO-1)|($$YPO+1)|$$PAF
055 0000000000000565 0001 ANA.getpixelcolor|0|($$XPO+1)|($$YPO-1)|$$PAG
056 0000000000000552 0001 ANA.getpixelcolor|0|($$XPO-1)|($$YPO-1)|$$PAH
057 0000000000000004 0001
058 0000000000001348 0001 CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+(
+($$PAF=&HFFFFFF)+($$PAG=&HFFFFFF)+($$PAH=&HFFFFFF)
059 0000000000000126 0001 IVV.$$ERG>5
060 0000000000000364 0001 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
061 0000000000000007 0001 EIF.
062 0000000000000000 0001 DMP.1
063 0000000000000000 0000 MBX.!
064 0000000000000000 0000 :EndLoop
065 0000000000000000 0000 NEX.
066 0000000000000000 0000 NEX.
067 0000000000000000 0000 RET.
068 0000000000000000 0000
```

The Speed-Dump was taken after the first round so the Line-execution count may not be accurate.

FASTER VERSION:

Using this Script the Writing will keep its original Color.
Single, unconnected Dots are removed in Stage 2.
This is technically the same Script as above, but twice as fast, as its using new Pixel-Commands
Needs 193 Seconds.

```
'
'SPR Script-file: RemoveColors
'Purpose:
'Author: Theo Gottwald
'Creation date: 12-28-2021 at 20:11:00
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' This Script will clean up the background of a digitized Writing
VAF.$$PIA=?exeloc\ULangK.png
VAF.$$PIB=?exeloc\ULangK1.png
ANA.Load|0|$$PIA
ANA.Show|0!
'ANA.AnalyzeColor|0|1
$$SRC=&HFFF5FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HEADAFE
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HE4F5FF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HC8C0E9
ANA.ReplaceColorT|0|$$SRC|10|$$COL

GSB.Rem_SngPix

ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
'============================================================

:Rem_SngPix
ANA.GetRes|0|$$XSZ|$$YSZ
PRT.Size=$$XSZ,$$YSZ
FOR.$$YPO|2|($$YSZ-1)
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|2|($$XSZ-1)
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
    JIV.$$PIX=&HFFFFFF|EndLoop

    ANA.gpc4s|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD
    ANA.gpc4d|0|$$XPO|$$YPO|$$PAE|$$PAF|$$PAG|$$PAH
    CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF
    IVV.$$ERG>5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.

    :EndLoop
  NEX.
NEX.
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000171 0001 VAF.$$PIA=?exeloc\ULangK.png
014 0000000000000067 0001 VAF.$$PIB=?exeloc\ULangK1.png
015 0000000017455743 0001 ANA.Load|0|$$PIA
016 0000000021536193 0001 ANA.Show|0!
017 0000000000000009 0001
018 0000000000000136 0001 VAR.$$SRC=&HFFF5FF
019 0000000000000041 0001 VAR.$$COL=&HFFFFFF
020 0000000000917850 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
021 0000000000000008 0001
022 0000000000000088 0001 VAR.$$SRC=&HEADAFE
023 0000000000955042 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
024 0000000000000009 0001
025 0000000000000090 0001 VAR.$$SRC=&HE4F5FF
026 0000000000957880 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
027 0000000000000008 0001
028 0000000000000086 0001 VAR.$$SRC=&HC8C0E9
029 0000000000964246 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
030 0000000000000148 0001 VAR.$$TIM=#dtime#
031 0000000000000062 0001 GSB.Rem_SngPix
032 0000000000000000 0000 CAL.$$TMA=#dsince#|i
033 0000000000000000 0000 DBP.Needed: $$TMA Seconds.
034 0000000000000000 0000 ANA.Show|0!
035 0000000000000000 0000 ANA.Save|0|$$PIB
036 0000000000000000 0000 ENR.
037 0000000000000000 0000
038 0000000000000000 0000
039 0000000000000009 0001 :Rem_SngPix
040 0000000000000687 0001 ANA.GetRes|0|$$XSZ|$$YSZ
041 0000000001259358 0001 PRT.Size=$$XSZ,$$YSZ
042 0000000000000500 0001 FOR.$$YPO|2|($$YSZ-1)
043 0000000000000574 0001 PRT.$$YPO/$$YSZ
044 0000000000000312 0001 FOR.$$XPO|2|($$XSZ-1)
045 0000000000000592 0001 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
046 0000000000000220 0001 JIV.$$PIX=&HFFFFFF|EndLoop
047 0000000000000005 0001
048 0000000000000720 0001 ANA.gpc4s|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD
049 0000000000000713 0001 ANA.gpc4d|0|$$XPO|$$YPO|$$PAE|$$PAF|$$PAG|$$PAH
050 0000000000001241 0001 CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFF
+($$PAF=&HFFFFFF)+($$PAG=&HFFFFFF)+($$PAH=&HFFFFFF)
051 0000000000000153 0001 IVV.$$ERG>5
052 0000000000000000 0000 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
053 0000000000000007 0001 EIF.
054 0000000000000000 0001 DMP.1
055 0000000000000000 0000 MBX.!
056 0000000000000000 0000 :EndLoop
057 0000000000000000 0000 NEX.
058 0000000000000000 0000 NEX.
059 0000000000000000 0000 RET.
```

The Speed-Dump was taken after the first round so the Line-execution count may not be accurate.

FASTER VERSION:

Using this Script the Writing will keep its original Color.
Single, unconnected Dots are removed in Stage 2.
This is technically the same Script as the two Scripts above, but twice as fast, as its using a new Pixel-Command that will read all 8 surrounding Pixels in one Strike.
Needs 101 Seconds.

```
'
'SPR Script-file: RemoveColors
'Purpose:
'Author: Theo Gottwald
'Creation date: 12-28-2021 at 20:11:00
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' This Script will clean up the background of a digitized Writing
VAF.$$PIA=?exeloc\ULangK.png
VAF.$$PIB=?exeloc\ULangK1.png
ANA.Load|0|$$PIA
ANA.Show|0!
'ANA.AnalyzeColor|0|1
$$SRC=&HFFF5FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HEADAFE
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HE4F5FF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HC8C0E9
ANA.ReplaceColorT|0|$$SRC|10|$$COL
$$TIM=#dtime#
GSB.Rem_SngPix
CAL.$$TMA=#dsince#|i
DBP.Needed: $$TMA Seconds.
DMP.1
MBX.!
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
'============================================================

:Rem_SngPix
ANA.GetRes|0|$$XSZ|$$YSZ
PRT.Size=$$XSZ,$$YSZ
FOR.$$YPO|2|($$YSZ-1)
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|2|($$XSZ-1)
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
    JIV.$$PIX=&HFFFFFF|EndLoop
    ANA.gpc8|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD|$$PAE|$$PAF|$$PAG|$$PAH
    CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF
    IVV.$$ERG>5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
    :EndLoop
  NEX.
NEX.
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000432 0001 VAF.$$PIA=?exeloc\ULangK.png
014 0000000000000053 0001 VAF.$$PIB=?exeloc\ULangK1.png
015 0000000000324768 0001 ANA.Load|0|$$PIA
016 0000000017534720 0001 ANA.Show|0!
017 0000000000000010 0001
018 0000000000000101 0001 VAR.$$SRC=&HFFF5FF
019 0000000000000041 0001 VAR.$$COL=&HFFFFFF
020 0000000000426034 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
021 0000000000000007 0001
022 0000000000000072 0001 VAR.$$SRC=&HEADAFE
023 0000000000408751 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
024 0000000000000009 0001
025 0000000000000074 0001 VAR.$$SRC=&HE4F5FF
026 0000000000403840 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
027 0000000000000009 0001
028 0000000000000077 0001 VAR.$$SRC=&HC8C0E9
029 0000000000395781 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
030 0000000000000125 0001 VAR.$$TIM=#dtime#
031 0000000000000053 0001 GSB.Rem_SngPix
032 0000000000000166 0001 CAL.$$TMA=#dsince#|i
033 0000000000000003 0001
034 0000000000000000 0001 DMP.1
035 0000000000000000 0000 MBX.!
036 0000000000000000 0000 ANA.Show|0!
037 0000000000000000 0000 ANA.Save|0|$$PIB
038 0000000000000000 0000 ENR.
039 0000000000000000 0000
040 0000000000000000 0000
041 0000000000000006 0001 :Rem_SngPix
042 0000000000000358 0001 ANA.GetRes|0|$$XSZ|$$YSZ
043 0000000001155265 0001 PRT.Size=$$XSZ,$$YSZ
044 0000000000000150 0562 FOR.$$YPO|2|($$YSZ-1)
045 0000000000001702 0561 PRT.$$YPO/$$YSZ
046 0000000000000162 496485 FOR.$$XPO|2|($$XSZ-1)
047 0000000000000204 495924 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
048 0000000000000079 495924 JIV.$$PIX=&HFFFFFF|EndLoop
049 0000000000000605 495924 ANA.gpc8|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD|$$PAE|$$PAF|$$PAG|$$PAH
050 0000000000000760 495924 CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF)
F)+($$PAF=&HFFFFFF)+($$PAG=&HFFFFFF)+($$PAH=&HFFFFFF)
051 0000000000000046 495924 IVV.$$ERG>5
052 0000000000000000 0000 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
053 0000000000000003 495924 EIF.
054 0000000000000003 495924 :EndLoop
055 0000000000000004 496485 NEX.
056 0000000000000004 0562 NEX.
057 0000000000000000 0001 RET.
```

Color-Separation mit dem SPR.

This Script will split a Logo into its Color-Components.

```
'#EXE:?path\
'#SPI:ForceWrite

VAF.$$PIA=?exeloc\KAIF_Logo.png
NEF.$$PIA
  MBX.File not found.
EIF.

VAF.$$PIB=?exeloc\KAIF_Separate_

' Goldene Farbton
VAR.$$C01=&HB99534

' Rosa
VAR.$$C02=&HE5006D

' Weiss
VAR.$$C03=&HFFFFFF
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Schwarz
VAR.$$C04=&H0

' Default Background GRÜN
VAR.$$BGC=&H00FF00

' Load Source
VAN.$$BMA=1
ANA.Load|$$BMA|$$PIA
ANA.GetRes|$$BMA|$$XSZ|$$YSZ
'ANA.Show|$$BMA!

' Generate P2 für Gold
VAN.$$BMB=2
ANA.Copy|$$BMA|$$BMB

' Generate P3 für Rosa
VAN.$$BMC=3
ANA.Copy|$$BMA|$$BMC

' Generate P4 für Weiss
VAN.$$BMD=4
ANA.Copy|$$BMA|$$BMD

' Generate P5 für Schwarz
VAN.$$BME=5
ANA.Copy|$$BMA|$$BME

VAN.$$TOL=4
' Alles ausser Gold ersetzen durch Grün
ANA.ReplaceColorT|$$BMB|!$$C01|$$TOL|$$BGC
ANA.Show|$$BMB!

' Alles ausser Rosa ersetzen durch Grün
ANA.ReplaceColorT|$$BMC|!$$C02|$$TOL|$$BGC

' Alles ausser Weiss ersetzen durch Grün
ANA.ReplaceColorT|$$BMD|!$$C03|$$TOL|$$BGC

' Alles ausser Schwarz ersetzen durch Grün
ANA.ReplaceColorT|$$BME|!$$C04|$$TOL|$$BGC

VAN.$$TMB=#dsince#
DBP.$$TMB used
ANA.Show|$$BMB!
ANA.Show|$$BMC!
ANA.Show|$$BMD!
ANA.Show|$$BME!

FOR.$$SAV|2|5
  VAR.$$NAM=$$PIBX$$SAV.png
  ANA.Save|$$SAV|$$NAM
NEX.
MBX.Done!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.1 ANA.AnalyzeColor

MiniRobotLanguage (MRL)

ANA.AnalyzeColor

Analyze Color in an Image-Register (IR) and generate an Color-Table

Intention

These Operations work with the "Color Table", this is a sorted list with all colors in the IR.
In fact there are 2 Color-Tables:
0 - **Color-Sort** (Col_Srt) - This table contains the Colors sorted by number of Pixel per Color.
The Top-Entry is the color with the most Pixels. The bottom Entry is the color with the least number of Pixels.
1 - **Color-Background** (Col_Bgl) - This table is sorted by similarity to the Color that has most Pixel, which is assumed to be the Background Color.
Here the Top-Entry contains the Background Color and then any further Entry is sorted by Similarity to this Color.

This list can be made using the

```
ANA.AnalyzeColor|0|1
```

The Col_Bgl Table can be sorted to another Color using:

```
ANA.Sort_Similar_To|&H26592A|$$MSK
```

Thee are several other commands that will only work if the Color-Table has been updated before.
These are:

```
ANA.GetBGColor
ANA.GetSortedColor
ANA.GetColorCount
ANA.SimilarBGColor
ANA.Sort_Similar
ANA.ShowSimArray
ANA.ShowSortArray
ANA.RemoveIndexColors
ANA.RemoveSortedColors
ANA.Reduce_Colors
ANA.ReplaceColorT
```

Using this Script:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BTC2.jpg
ANA.Load|0|$$PIA
ANA.AnalyzeColor|0|1
ANA.showsortarray|0
MBX.!
ENR.
```

You will get a listing of the color-table, this may lokk like this:

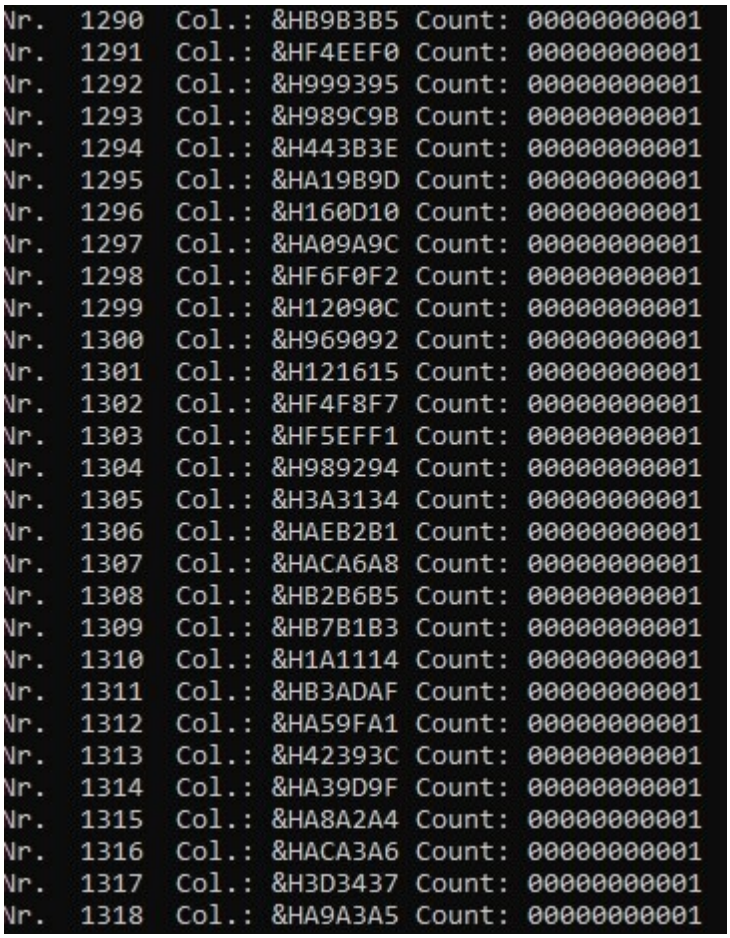
© 2023 Theo Gottwald * https://www.smart-package.com

```
Color-Array:
Nr.  0  Col.: &HFFFFFF Count: 00000368847
Nr.  1  Col.: &HFEFEFE Count: 00000125732
Nr.  2  Col.: &H231F20 Count: 00000089334
Nr.  3  Col.: &HFDFDFD Count: 00000034944
Nr.  4  Col.: &HFCFCFC Count: 00000027332
Nr.  5  Col.: &HFBFBFB Count: 00000022810
Nr.  6  Col.: &HFAFAFA Count: 00000019557
Nr.  7  Col.: &HF9F9F9 Count: 00000016872
Nr.  8  Col.: &HF8F8F8 Count: 00000014905
Nr.  9  Col.: &HF7F7F7 Count: 00000013093
Nr.  10  Col.: &HFFFEFF Count: 00000012328
Nr.  11  Col.: &HF6F6F6 Count: 00000011193
Nr.  12  Col.: &HF5F5F5 Count: 00000010183
Nr.  13  Col.: &H221E1F Count: 00000010177
Nr.  14  Col.: &H242021 Count: 00000010165
Nr.  15  Col.: &HF4F4F4 Count: 00000008620
Nr.  16  Col.: &HF3F3F3 Count: 00000007677
Nr.  17  Col.: &HF2F2F2 Count: 00000006652
Nr.  18  Col.: &H211F20 Count: 00000006330
Nr.  19  Col.: &H252122 Count: 00000006230
Nr.  20  Col.: &HF1F1F1 Count: 00000005879
Nr.  21  Col.: &H211D1E Count: 00000005854
Nr.  22  Col.: &HF0F0F0 Count: 00000005121
Nr.  23  Col.: &HEFEFEF Count: 00000004787
Nr.  24  Col.: &HFFFDFE Count: 00000004596
Nr.  25  Col.: &HEEEEEE Count: 00000004273
Nr.  26  Col.: &H262223 Count: 00000004142
Nr.  27  Col.: &H201C1D Count: 00000003965
Nr.  28  Col.: &HEDEDED Count: 00000003949
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
Nr.  35  Col.: &H1F1B1C Count: 00000003042
Nr.  36  Col.: &H201E1F Count: 00000002931
Nr.  37  Col.: &HEAEAEA Count: 00000002906
Nr.  38  Col.: &HE9E9E9 Count: 00000002720
Nr.  39  Col.: &HFCFAFB Count: 00000002698
Nr.  40  Col.: &H000000 Count: 00000002558
Nr.  41  Col.: &HE8E8E8 Count: 00000002524
Nr.  42  Col.: &HE7E7E7 Count: 00000002469
Nr.  43  Col.: &H282425 Count: 00000002434
Nr.  44  Col.: &H232122 Count: 00000002349
Nr.  45  Col.: &HFBF9FA Count: 00000002336
Nr.  46  Col.: &H1F1D1E Count: 00000002286
Nr.  47  Col.: &HE5E5E5 Count: 00000002168
Nr.  48  Col.: &H1E1A1B Count: 00000002166
Nr.  49  Col.: &HE6E6E6 Count: 00000002152
Nr.  50  Col.: &HFEFFFF Count: 00000002149
Nr.  51  Col.: &H241E20 Count: 00000002123
Nr.  52  Col.: &HFAF8F9 Count: 00000002059
Nr.  53  Col.: &H292526 Count: 00000001916
Nr.  54  Col.: &H242223 Count: 00000001910
Nr.  55  Col.: &H1E1C1D Count: 00000001903
Nr.  56  Col.: &H1D191A Count: 00000001856
Nr.  57  Col.: &HE4E4E4 Count: 00000001776
Nr.  58  Col.: &HF9F7F8 Count: 00000001755
Nr.  59  Col.: &HE3E3E3 Count: 00000001664
Nr.  60  Col.: &H252324 Count: 00000001604
Nr.  61  Col.: &H1D1B1C Count: 00000001596
Nr.  62  Col.: &H2A2627 Count: 00000001574
Nr.  63  Col.: &HF8F6F7 Count: 00000001541
Nr.  64  Col.: &HE1E1E1 Count: 00000001509
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Start of the Color-Tabel, Entries 0 to 41. the Entry for "Black" is on position 40.

Second part. As you can see, End of the Color-Table. Entries up to 1318.

Using this commands you can work with the list entries in the color table and do significant improvements in pictures.
Now you can improve the image. All colors are already sorted by Quantity (How many Pixels).

You can also get a sort on "Color difference to background". For this you can use:

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BTC2.jpg
ANA.Load|0|$$PIA
ANA.AnalyzeColor|0|1
ANA.ShowSimArray|0
MBX.!
ENR.
```

Then you will get something like this:

© 2023 Theo Gottwald * https://www.smart-package.com

System Messages

```
Similarity-Array:
Nr.  0  Col.: &H000000 Count: 00000002558 Color-Diff.:  428
Nr.  1  Col.: &H000100 Count: 00000000021 Color-Diff.:  162
Nr.  2  Col.: &H010000 Count: 00000000405 Color-Diff.:  333
Nr.  3  Col.: &H010101 Count: 00000000644 Color-Diff.:  368
Nr.  4  Col.: &H000201 Count: 00000000014 Color-Diff.:  139
Nr.  5  Col.: &H030000 Count: 00000000202 Color-Diff.:  305
Nr.  6  Col.: &H020001 Count: 00000000127 Color-Diff.:  266
Nr.  7  Col.: &H040001 Count: 00000000064 Color-Diff.:  218
Nr.  8  Col.: &H040000 Count: 00000000005 Color-Diff.:  66
Nr.  9  Col.: &H050001 Count: 00000000001 Color-Diff.:  28
Nr.  10  Col.: &H010302 Count: 00000000005 Color-Diff.:  72
Nr.  11  Col.: &H020202 Count: 00000000647 Color-Diff.:  371
Nr.  12  Col.: &H030102 Count: 00000000148 Color-Diff.:  282
Nr.  13  Col.: &H060002 Count: 00000000002 Color-Diff.:  46
Nr.  14  Col.: &H050102 Count: 00000000081 Color-Diff.:  227
Nr.  15  Col.: &H020403 Count: 00000000006 Color-Diff.:  85
Nr.  16  Col.: &H030303 Count: 00000000634 Color-Diff.:  366
Nr.  17  Col.: &H040203 Count: 00000000156 Color-Diff.:  286
Nr.  18  Col.: &H050304 Count: 00000000156 Color-Diff.:  286
Nr.  19  Col.: &H040404 Count: 00000000678 Color-Diff.:  378
Nr.  20  Col.: &H070103 Count: 00000000002 Color-Diff.:  45
Nr.  21  Col.: &H030504 Count: 00000000005 Color-Diff.:  66
Nr.  22  Col.: &H060203 Count: 00000000080 Color-Diff.:  226
Nr.  23  Col.: &H080204 Count: 00000000001 Color-Diff.:  20
Nr.  24  Col.: &H070304 Count: 00000000087 Color-Diff.:  229
Nr.  25  Col.: &H050505 Count: 00000000683 Color-Diff.:  379
Nr.  26  Col.: &H060405 Count: 00000000180 Color-Diff.:  298
Nr.  27  Col.: &H040605 Count: 00000000007 Color-Diff.:  87
Nr.  28  Col.: &H080405 Count: 00000000103 Color-Diff.:  240
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
Nr.  1290  Col.: &HF9FBFA Count: 00000000376 Color-Diff.:  330
Nr.  1291  Col.: &HFEF8FA Count: 00000000005 Color-Diff.:  68
Nr.  1292  Col.: &HFDF9FA Count: 00000000606 Color-Diff.:  357
Nr.  1293  Col.: &HFCFAFB Count: 00000002698 Color-Diff.:  428
Nr.  1294  Col.: &HFBFBFB Count: 00000022810 Color-Diff.:  440
Nr.  1295  Col.: &HFFF9FB Count: 00000000007 Color-Diff.:  94
Nr.  1296  Col.: &HFAFCFB Count: 00000000371 Color-Diff.:  330
Nr.  1297  Col.: &HFEFAFB Count: 00000000633 Color-Diff.:  365
Nr.  1298  Col.: &HFFFAFC Count: 00000000005 Color-Diff.:  69
Nr.  1299  Col.: &HFDFBFC Count: 00000003121 Color-Diff.:  430
Nr.  1300  Col.: &HFAFEFD Count: 00000000001 Color-Diff.:  23
Nr.  1301  Col.: &HFCFCFC Count: 00000027332 Color-Diff.:  440
Nr.  1302  Col.: &HFBFDFC Count: 00000000458 Color-Diff.:  335
Nr.  1303  Col.: &HFFFBFC Count: 00000000685 Color-Diff.:  380
Nr.  1304  Col.: &HFCFEFD Count: 00000000591 Color-Diff.:  353
Nr.  1305  Col.: &HFFFBFD Count: 00000000005 Color-Diff.:  66
Nr.  1306  Col.: &HFFFCFD Count: 00000000750 Color-Diff.:  394
Nr.  1307  Col.: &HFEFCFD Count: 00000003515 Color-Diff.:  431
Nr.  1308  Col.: &HFBFFFE Count: 00000000001 Color-Diff.:  21
Nr.  1309  Col.: &HFFFCFE Count: 00000000003 Color-Diff.:  54
Nr.  1310  Col.: &HFCFFFF Count: 00000000008 Color-Diff.:  104
Nr.  1311  Col.: &HFDFDFD Count: 00000034944 Color-Diff.:  440
Nr.  1312  Col.: &HFDFFFE Count: 00000000757 Color-Diff.:  395
Nr.  1313  Col.: &HFEFEFE Count: 00000125732 Color-Diff.:  441
Nr.  1314  Col.: &HFFFDFF Count: 00000000011 Color-Diff.:  126
Nr.  1315  Col.: &HFFFDFE Count: 00000004596 Color-Diff.:  434
Nr.  1316  Col.: &HFFFEFF Count: 00000012328 Color-Diff.:  438
Nr.  1317  Col.: &HFEFFFF Count: 00000002149 Color-Diff.:  424
Nr.  1318  Col.: &HFFFFFF Count: 00000368847 Color-Diff.:  442
```

In this table the black colors will be "on Top" while the White colors will be "bottom". For this sort the robot assumes that the color that is mostly in the picture "is the background color".
The he will sort all colors "by difference to the background". This is the result.

Using these tables you can significantly improve pictures with your Script.

For example you can remove colors from the list and the IR using:

```
' this command will remove the given color and replace it with another color.
ANA.removeindexcolors|0|300|$$CNV|0

VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
'VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
GSB.Reanalyze
ANA.removeindexcolors|0|300|$$CNV|0
ANA.removeindexcolors|0|$$CNV|$$CNT|-1
GSB.Reanalyze
ANA.removeindexcolors|0|$$CNV|$$CNT|-1
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Reanalyze
ANA.AnalyzeColor|0|1
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
CAL.$$CNV=$$CNT-400
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

Original Picture

Picture-Result

With a few changes, you can see in detail how the Script works:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Here we have changed the script a bit to show
' how the commands paint the colors.
' Green -> would be painted in white
' red -> would be painted in black
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
'VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
GSB.Reanalyze
ANA.removeindexcolors|0|300|$$CNV|&HFF0000
GSB.Reanalyze
ANA.removeindexcolors|0|$$CNV|($$CNT-20)|&H00FF00
ANA.removeindexcolors|0|($$CNT-20)|$$CNT|&HFFFFFF
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.

:Reanalyze
ANA.AnalyzeColor|0|1
ANA.getcolorcount|0|$$CNT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.getbgcolor|0|$$BGC
CAL.$$CNV=$$CNT-400
RET.
```

```
010 0000000000000694 0001 VAN.$$TIM=#time#
011 0000000000000525 0001 VAF.$$PIA=?exeloc\BNB.jpg
012 0000000000000016 0001
013 0000000000000383 0001 VAF.$$PIB=?exeloc\V1.png
014 0000000001215801 0001 ANA.Load|0|$$PIA
015 0000000000000118 0001 GSB.Reanalyze
016 0000000010022301 0001 ANA.removeindexcolors|0|300|$$CNV|&HFF0000
017 0000000000000118 0001 GSB.Reanalyze
018 0000000026517894 0001 ANA.removeindexcolors|0|$$CNV|($$CNT-20)|&H00FF00
019 0000000000912936 0001 ANA.removeindexcolors|0|($$CNT-20)|$$CNT|&HFFFFFF
020 0000000000001814 0001 CAL.$$TIM=#time#-$$TIM
021 0000000000006267 0001 DBP.Used Time in Seconds: $$TIM
022 0000000000000000 0001 DMP.1
023 0000000000000000 0000 ANA.Save|0|$$PIB
024 0000000000000000 0000 ANA.Show|0!
025 0000000000000000 0000 ENR.
026 0000000000000000 0000
027 0000000000000018 0002 :Reanalyze
028 0000000000901451 0002 ANA.AnalyzeColor|0|1
029 0000000000001675 0002 ANA.getcolorcount|0|$$CNT
030 0000000000000730 0002 ANA.getbgcolor|0|$$BGC
031 0000000000000968 0002 CAL.$$CNV=$$CNT-400
032 0000000000000208 0002 RET.
033 0000000000000000 0000
```

This is the Speed-Dump of the Script. It uses just few Seconds for the complete run.

Syntax

ANA.AnalyzeColors|P1[|P2]

Parameter Explanation

P1 **- *Source* Image-Register Number**
P2 **- *0/1 - Force-Reanalyze-Flag. Default is 0.***
Generally the Analyze command will do nothing if this IR has already been analyzed.
In case you have made changes to the IR and need a new AnalyzeColors then must set the "Force-Flag" to 1.

Example

Remove Green Background

To do this, we first Analyze the Picture to generate the Color-Table.
Secondly we sort the Color Table in a way that its sorted by Similarity with the Background Green Color.
Third we remove the Entries that are similar to the Green Color.
The numbers that are used in detail are a bit of trial and error depending on the picture.

© 2023 Theo Gottwald * https://www.smart-package.com

->

```
' This Script will remove a green background
' using special sorting functions.
' The Script is fast and will only take few seconds
VAF.$$PIA=?exeloc\Sara_Green.jpg
ANA.Load|0|$$PIA
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
' Sort the Color Table with Similarity to the Background Color
ANA.Sort_Similar_To|$$COL|$$MSK
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
ANA.RemoveIndexColors|0|39300|$$CNT|&HFFFFFF
ANA.Show|0!
ENR.
```

```
010 0000000000000323 0001 VAN.$$TIM=#time#
011 0000000000000109 0001 VAF.$$PIA=?exeloc\Sara_Green.jpg
012 0000000000334747 0001 ANA.Load|0|$$PIA
013 0000000001419195 0001 ANA.AnalyzeColor|0|1
014 0000000000000008 0001
015 0000000000000187 0001 VAR.$$COL=&H30FF30
016 0000000000000042 0001 VAR.$$MSK=&HFFFFFF
017 0000000000000004 0001
018 0000000000000440 0001 ANA.GetColorCount|0|$$CNT
019 0000000000001425 0001 DBP.$$CNT
020 0000000000666471 0001 ANA.ssto|$$COL|$$MSK
021 0000000835176274 0001 ANA.Show|0!
022 0000000133032400 0001 ANA.removeindexcolors|0|39300|$$CNT|&HFFFFFF
023 0000000000000008 0001
024 0000000000000000 0001 DMP.1
025 0000000000000000 0000 ANA.Show|0!
```

Remarks

–

Limitations:

-

See also:

•

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.4.2 ANA.GetBGColor

MiniRobotLanguage (MRL)

ANA.GetBGColor

Returns the Color of the Background of an IR

Intention

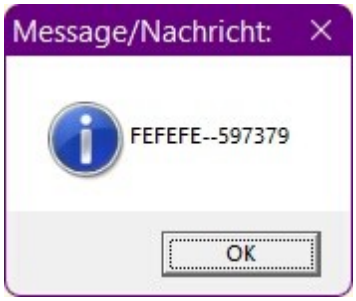
This Command needs the ANA.AnalyzeColor Command to count colors in an IR before it will work. It may also need the "Force-Flag" (P2) to be set.

```
ANA.AnalyzeColor|0|1

' These are some further commands that can be used after t
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
```

you can then work with the list entries.

This ccommand will return the color that is mostly used in the IR. It is assumed that the "Color of most of the Pixels" is the Background color.

```
$$PAT=?exeloc\pics\P5.jpg
IEF.$$PAT
  ANA.Load|0|$$PAT
  ANA.Show|0!
ELS.
  MBX: FIle does not exist.
EIF.
ANA.AnalyzeColor|0|1
ANA.GetBgcolor|0|$$RES|$$ANZ
VTH.$$RES|$$HEX
MBX.$$HEX--$$ANZ
ENR.
```



Will bring up this Messagebox: that shows that the Color &HFEFEFE is most likely the Background Color. And this color is found 597379 times in the original picture.

© 2023 Theo Gottwald * https://www.smart-package.com

With this picture:

Syntax

ANA.GetBGColor[|P1][|P2][|P3]

Parameter Explanation

P1 **- *Source* Image-Register Number, if omitted 0 is used.**
P2 **- *(Result) Color of Background*, if omitted the result is placed on TOS.**
P3 **- *(Result) Number of Pixels using the Color of Background (in P2)*, if omitted the result is placed on TOS.**

The Command will not leave a 0/1 on the TOS. If the Operation fails, **P2** may be "-1".

Example

```
'**********************************
' ANA.Fill-Sample
'**********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  ANA.New|0|320|200
  ANA.Fill|0|&HFFFFFF|1|&HFF0000
  ANA.Show|0!
  ANA.Save|0|?path\Lines01.png
  ENR.

' Below you can see the result of the Operation.
```

Remarks

-

Limitations:

-

See also:

- ANA.AnalyeColor 1225
- ANA.RemoveIndexColors 1254

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.3 ANA.GetColorCount

MiniRobotLanguage (MRL)

ANA.GetColorCount

Returns the Color of the Background of an IR

Intention

This Command needs the ANA.AnalyzeColor Command to count colors in an IR before it will work. It may also need the "Force-Flag" (P2) to be set.

```
ANA.AnalyzeColor|0|1

' These are some further commands that can be used after t
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
```

you can then work with the list entries.

This ccommand will return the color that is mostly used in the IR. It is assumed that the "Color of most of the Pixels" is the Background color.

```
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
MBX.Number of Color in IR 0: $$CNT
ENR.
```

Will bring up this Messagebox.

With this picture:

© 2023 Theo Gottwald * https://www.smart-package.com

Here is the Speed-Dump:

```
013 0000000000000200 0001 VAF.$$PIA=?exeloc\Sarah_BG.png
014 0000000000488917 0001 ANA.Load|0|$$PIA
015 0000000153351096 0001 ANA.Show|0!
016 0000000001426101 0001 ANA.AnalyzeColor|0|1
017 0000000000000008 0001
018 0000000000000116 0001 VAR.$$COL=&H30FF30
019 0000000000000044 0001 VAR.$$MSK=&HFFFFFF
020 0000000000000453 0001 ANA.GetColorCount|0|$$CNT
021 0000000000000000 0001 DMP.Speed
022 0000000000000000 0000 MBX.$$CNT
023 0000000000000000 0000 CAL.$$LES=$$CNT-5
024 0000000000000000 0000 ANA.Sort_Similar_To|&H26592A|$$MSK
025 0000000000000000 0000 ANA.Reduce_Colors|0|$$LES|$$CNT|90|-1|2
026 0000000000000000 0000 ANA.GetColorCount|0|$$CNT
027 0000000000000000 0000 DBP.$$CNT
028 0000000000000000 0000 DMP.Speed
029 0000000000000000 0000 ANA.Show|0!
030 0000000000000000 0000 ENR.
```

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.GetColorCount[|P1][|P2]

Parameter Explanation

***P1 - Source* Image-Register Number, if omitted 0 is used.**
***P2 - (Result) Color of Background*, if omitted the result is placed on TOS.**
***P3 - (Result) Number of Pixels using the Color of Background (in P2)*, if omitted the result is placed on TOS.**

The Command will not leave a 0/1 on the TOS. If the Operation fails, **P2** may be "0".

Example

```
'**********************************
' ANA.GetColorCounts-Sample
'**********************************
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
CAL.$$LES=$$CNT-5
ANA.Sort_Similar_To|&H26592A|$$MSK
ANA.Reduce_Colors|0|$$LES|$$CNT|90|-1|2
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
DMP.Speed
ANA.Show|0!
ENR.

' Below you can see the result of the Operation.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ANA.AnalyeColor 1225
- ANA.RemoveIndexColors 1254

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.4 ANA.Reduce_Colors

MiniRobotLanguage (MRL)

ANA.Reduce_Colors

Returns the number of Colors of an IR

Intention

This ccommand will choose 2 similar colors from the color table, calculate the color-mix between these two colors
and then replace both colors with the Mix-Color. This way reducing the number of colors.
This Command needs the ANA.AnalyzeColor Command to count colors in an IR before it will work. It may also need the "Force-Flag" (P2) to be set.
It will also use the Color-Similarity Table, therefore you will need to use the
`ANA.Sort_Similar_To|$$COL|$$MSK`
Command prior using this command.
Using the Tolerance Parameter with another value then 0 will slow down the process, but will also catch a lot more similar colors during the replace.
Somehow using the Tolerance, can be seen as if it is another command, as it will influence the result generally.
Therefore the number of Entries to reduce can be dramatically reduced when using Tolerance.

Using this command is in 5 steps:

```
1. Use
ANA.AnalyzeColor|0|1
to generate a Color-Table which contains all Colors and number of Pixels.

2. Sort the Color Table with Similarity to the Background Color
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
' Colors that do not apply to $$MSK will not be sorted
$$MSK=&HFFFFFF

3. (optional) Use
ANA.GetColorCount|0|$$CNT
to get the total number of colors

4. Use
ANA.Sort_Similar_To|$$COL|$$MSK
to sort the Color-Table by Similarity to the given Color

5. Now call the command to mix/reduce those colors that are most similar to the
' Remove Entries and Paint these Pixels with the Mix-color.
ANA.Reduce_Colors|0|10000|$$CNT|0|-1|2
```


© 2023 Theo Gottwald * https://www.smart-package.com

Here is a Sample-Script.

```
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
' Sort the Color Table with Similarity to the Background Color
ANA.Sort_Similar_To|$$COL|$$MSK
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
ANA.Reduce_Colors|0|30000|$$CNT|0|-1|2
DMP.6
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Reduce Number ofColors in a Picture may result in a Comic-like effect.

Sample Script using Tolerance is very fast due to the low number of Entries to process, yet the colors are reduced by over 13000 Colors in one run:

```
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
CAL.$$LES=$$CNT-5
ANA.Sort_Similar_To|&H26592A|$$MSK
ANA.Reduce_Colors|0|$$LES|$$CNT|90|-1|2
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
DMP.6
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Here we are using a 2 step color-reduction. In the first step we reduces the number of colors from 53870 to 39939.

```
013 0000000000000185 0001 VAF.$$PIA=?exeloc\Sarah_BG.png
014 0000000000425099 0001 ANA.Load|0|$$PIA
015 0000000028339564 0001 ANA.Show|0!
016 0000000001358294 0001 ANA.AnalyzeColor|0|1
017 0000000000000009 0001
018 0000000000000139 0001 VAR.$$COL=&H30FF30
019 0000000000000038 0001 VAR.$$MSK=&HFFFFFF
020 0000000000000353 0001 ANA.GetColorCount|0|$$CNT
021 0000000000001923 0001 DBP.$$CNT
022 0000000000000253 0001 CAL.$$LES=$$CNT-5
023 0000000000680886 0001 ANA.Sort_Similar_To|&H26592A|$$MSK
024 0000000006448249 0001 ANA.Reduce_Colors|0|$$LES|$$CNT|90|-1|2
025 0000000000000528 0001 ANA.GetColorCount|0|$$CNT
026 0000000000009450 0001 DBP.$$CNT
027 0000000000000000 0001 DMP.Speed
028 0000000000000000 0000 ANA.Show|0!
029 0000000000000000 0000 ENR.
```

Speed-Table of this Script.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.Reduce_Colors|P1|P2|P3[|P4][|P5][|P6]

Parameter Explanation

***P1** - Source and Destination* **Image-Register Number, if omitted 0 is used.**
***P2** - Color-Table-Start Index - must be lower then the total number of colors in the Color-Table.*
***P3** - Color-Table-End Index - In most cases this is the number of Entries in the Color-Table as its given by* ANA.GetColorCount|0|$$CNT.
If this number is too high it will automatically be cut down to the maximum. If P3<P2 then these two values will be exchanged.
***P4** - 0 or Color-Tolerance-Value. This value is used in the color-replacing part. If >0 the color-replace will also replace*
all colors within the Tolerance range of target-color + Tolerance value. This may slow down the process and also lead to unwanted results if wrongly used.
P5 - if -1 this Parameter is ignored, if 0-8, the command will use a Color-Condition Register (CCR) from 0 to 8 to replace only colors
within this range of the Color-Condition Register.
P6 - 0/1 - 0 - Use Col_Bgl (Backgound-Color Table), 1 - Use Col_Srt (Sorted Colors Table) default is 0
+ 2 Use Color-Ratio Algo for Color-Compare
+ 4 Change color-Mix Mode: 0 - Mix with evaluation by Number Count of Pixels,4 - just take larger color with more pixels
+ 8 Use Inverse CCR (Protect Colors given by the CCR)
+ 16 Use Pixelcount in Mix-Algo, means that colors with more pixels have larger influence on Mixing-Color.

The Command will leave a 0/1 on the TOS. If the Operation fails, TOS will show a "0".

Example

-

Remarks

-

Limitations:

While this command is not a slow command it may still take some seconds to some Minutes to execute, depending on how many colors to remove, the use of Color-Tolerance and the picture-Size.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ANA.AnalyeColor 1225
- ANA.RemoveIndexColors 1254

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.5 ANA.RemoveIndexColors

MiniRobotLanguage (MRL)

ANA.RemoveIndexColors

Replace colors that are similar to a given color with a given color in an IR

Intention

This ccommand will replace colors that are equal to a given color with another color.
This Command needs the ANA.AnalyzeColor Command to count colors in an IR before it will work. It may also need the "Force-Flag" (P2) to be set.
It will also use the Color-Similarity Table, therefore you will need to use the
`ANA.Sort_Similar_To|$$COL|$$MSK`
Command prior using this command.
This Command will work on the Col_Bgl-Color Table that contains all Colors sorted by the Similarity to the Background-Color.

Using this command is in 5 steps:

```
1. Use
ANA.AnalyzeColor|0|1
to generate a Color-Table which contains all Colors and number of Pixels.

2. Sort the Color Table with Similarity to the Background Color
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF

3. (optional) Use
ANA.GetColorCount|0|$$CNT
to get the total number of colors

4. Use
ANA.Sort_Similar_To|$$COL|$$MSK
to sort the Color-Table by Similarity to the given Color

5. Now call the command to remove those colors that are most similar to the give
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
ANA.RemoveIndexColors|0|40000|$$CNT|&HFFFFFF
```

Here is a Sample-Script.

```
' This Script will remove a green background
' using special sorting functions.
' The Script is fast and will only take few seconds
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
DBP.$$CNT
' Sort the Color Table with Similarity to the Background Color
ANA.Sort_Similar_To|$$COL|$$MSK
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
ANA.RemoveIndexColors|0|40000|$$CNT|&HFFFFFF
ANA.Show|0!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

© 2023 Theo Gottwald * https://www.smart-package.com

Will bring up this Messagebox: that shows that the Color &HFEFEFE is most likely the Background Color. And this color is found 597379 times in the original picture.

With this picture:

Syntax

ANA.RemoveIndexColors[|P1][|P2][|P3]

Parameter Explanation

P1 - Source and Destination **Image-Register Number, if omitted 0 is used.**
P2 - Color-Table-Start Index - must be lower then the total number of colors in the Color-Table.
P3 - Color-Table-End Index - In most cases this is the number of Entries in the Color-Table as its given by ANA.GetColorCount|0|$$CNT.
If this number is too high it will automatically be cut

© 2023 Theo Gottwald * https://www.smart-package.com

down to the maximum. If P3<P2 then these two values will be exchanged.
P4 - Color to paint over or -1 to choose Background-Color, choose -2 then you will get an automatic color-Mix between the color
that is replaced and the next color in the color-Table.

The Command will leave a 0/1 on the TOS. If the Operation fails, TOS will show a "0".

Example

-

Remarks

-

Limitations:

While this command is not a slow command it may still take some seconds to execute, depending on how many colors to remove and the picture-Size.

See also:

- ANA.AnalyeColor 1225
- ANA.Reduce_Colors 1246

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.6 ANA.RemoveSortedColors

MiniRobotLanguage (MRL)

ANA.RemoveSortedColors

Replace colors that are similar to a given color with a given color in an IR

Intention

This ccommand will replace colors that are sorted in the Col_Srt Colortable by "Number of Pixels".
This Command needs the ANA.AnalyzeColor Command to count colors in an IR before it will work. It may also need the "Force-Flag" (P2) to be set.
It will also use the Color-Similarity Table, therefore you will need to use the
`ANA.Sort_Similar_To|$$COL|$$MSK`
Command prior using this command.
This Command will work on the Col_Srt-Color Table that contains all Colors sorted by the Number of Pixels in the Picture.

Using this command is in 3 steps:

```
1. Use
ANA.AnalyzeColor|0|1
to generate a Color-Table which contains all Colors and number of Pixels.

2. (optional) Use
ANA.GetColorCount|0|$$CNT
to get the total number of colors

3. Now call the command to remove those colors that contain the most or least nu
ANA.RemoveSortedColors|0|40000|$$CNT|&HFFFFFF
```

Here is a Sample-Script. This Script will mark the Areas of the Picture which contain most of the Information.

```
VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$REP=&HFFFF00
ANA.GetColorCount|0|$$CNT
CAL.$$LES=$$CNT-10000
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
ANA.RemoveSortedColors|0|$$LES|$$CNT|$$REP
ANA.GetColorCount|0|$$CNT
CAL.$$LES=$$CNT-20000
$$REP=&HFF0000
ANA.RemoveSortedColors|0|$$LES|$$CNT|$$REP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.Speed
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Using this Command you can take a look, which Colors are not used very often in the picture. Or which parts of the picture contain most of the Information.

Here is teh Speed-Dump:

```
013 0000000000000273 0001 VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
014 0000000000428661 0001 ANA.Load|0|$$PIA
015 0000000030227133 0001 ANA.Show|0!
016 0000000001387960 0001 ANA.AnalyzeColor|0|1
017 0000000000000008 0001
018 0000000000000138 0001 VAR.$$REP=&HFFFF00
019 0000000000000390 0001 ANA.GetColorCount|0|$$CNT
020 0000000000000256 0001 CAL.$$LES=$$CNT-10000
021 0000000000000003 0001
022 0000000089921992 0001 ANA.RemoveSortedColors|0|$$LES|$$CNT|$$REP
023 0000000000000495 0001 ANA.GetColorCount|0|$$CNT
024 0000000000000267 0001 CAL.$$LES=$$CNT-20000
025 0000000000000041 0001 VAR.$$REP=&HFF0000
026 0000000195446468 0001 ANA.RemoveSortedColors|0|$$LES|$$CNT|$$REP
027 0000000000000000 0001 DMP.Speed
```

```
' This Script will mark locations that contain much information (red), and locat
VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$REP=&HFFFF00
ANA.GetColorCount|0|$$CNT
CAL.$$LES=$$CNT-10000
' Remove most often used colors and Paint these Pixels Yellow
ANA.RemoveSortedColors|0|0|30|$$REP
ANA.AnalyzeColor|0|1
ANA.GetColorCount|0|$$CNT
ANA.Show|0!
CAL.$$LES=$$CNT-30000
$$REP=&HFF0000
' Remove seldom colors and Paint these Pixels Red
ANA.RemoveSortedColors|0|$$LES|$$CNT|$$REP
DMP.Speed
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Here is the Speed-Dump for the Script.
It shows large differences depending on the Amount of Colors to replace, while the amount of Pixels does not play a big role.

```
013 0000000000000193 0001 VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
014 0000000000402286 0001 ANA.Load|0|$$PIA
015 0000000028437694 0001 ANA.Show|0!
016 0000000001381067 0001 ANA.AnalyzeColor|0|1
017 0000000000000009 0001
018 0000000000000138 0001 VAR.$$REP=&HFFFF00
019 0000000000000367 0001 ANA.GetColorCount|0|$$CNT
020 0000000000000259 0001 CAL.$$LES=$$CNT-10000
021 0000000000000004 0001
022 0000000001743344 0001 ANA.RemoveSortedColors|0|0|30|$$REP
023 0000000001410994 0001 ANA.AnalyzeColor|0|1
024 0000000000000416 0001 ANA.GetColorCount|0|$$CNT
025 0000000018841822 0001 ANA.Show|0!
026 0000000000000559 0001 CAL.$$LES=$$CNT-30000
027 0000000000000043 0001 VAR.$$REP=&HFF0000
028 0000000286746249 0001 ANA.RemoveSortedColors|0|$$LES|$$CNT|$$REP
029 0000000000000000 0001 DMP.Speed
```

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.RemoveSortedColors[|P1][|P2][|P3]

Parameter Explanation

P1 **- *Source and Destination* Image-Register Number, if omitted 0 is used.**
P2 **- *Color-Table-Start Index - must be lower then the total number of colors in the Color-Table.***
P3 **- *Color-Table-End Index - In most cases this is the number of Entries in the Color-Table as its given by*** ANA.GetColorCount|0|$$CNT.
If this number is too high it will automatically be cut down to the maximum. If P3<P2 then these two values will be exchanged.
P4 **- *Color to paint over or -1 to choose Background-Color, choose -2 then you will get an automatic color-Mix between the color***
that is replaced and the next color in the color-Table.

The Command will leave a 0/1 on the TOS. If the Operation fails, TOS will show a "0".

Example

-

Remarks

-

Limitations:

While this command is not a slow command it may still take some seconds to execute, depending on how many colors to remove and the picture-Size.

See also:

- ANA.AnalyeColor 1225
- ANA.Reduce_Colors 1246

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.7 ANA.ReplaceColorD

MiniRobotLanguage (MRL)

ANA.ReplaceColorD

Replace Color Direct - Replace a given Color with another Color directly in the IR

Intention

This Command will directly replace colors in an IR. It will replace Color **P2** with Color **P3**. In Image-Register **P1**.
This Command is NOT really an Color-Table Operation, as it does not need the Color-Table.
Instead it will directly Replace Color **P2** with Color **P3** in Image-Register **P1**.
If Color **P2** is prefixed with an **!** then the Command will work in a **negative mode**, means it will replace the Color if it does NOT Fit to the Color **P2**.

```
' ANA.ReplaceColorD
VAF.$$PIA=?exeloc\Pics\Fux.jpg
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$SRC=&H000000
$$COL=&H30FF30
ANA.ReplaceColorD|0|$$SRC|$$COL
DMP.Speed
ANA.Show|0!
ENR.
```

In this Script we will replace the Color that has the most Pixels, with white.

```
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
ANA.GetBGColor|0|$$BGC
ANA.ReplaceColorD|0|$$BGC|&HFFFFFF
DMP.Speed
ANA.Show|0!
ENR.
```

Will bring up this Messagebox.

With this picture:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Here we enhance the Script a bit to get the Background removed immediately.

```
VAF.$$PIA=?exeloc\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$COL=&H30FF30
$$MSK=&HFFFFFF
ANA.GetColorCount|0|$$CNT
CAL.$$LES=$$CNT-5
' Sort the Color Table with Similarity to the Background Color
ANA.Sort_Similar_To|$$COL|$$MSK
' Remove Entries and Paint these Pixels white that are Similar to the Backgroun
ANA.Reduce_Colors|0|$$LES|$$CNT|90|-1|2
ANA.AnalyzeColor|0|1
ANA.GetBgcolor|0|$$BGC
ANA.ReplaceColorD|0|$$BGC|&HFFFFFF
DMP.Speed
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.ReplaceColorD|P1|[!]P2[|P3]

Parameter Explanation

P1 **- *Source & Destination* Image-Register Number, if omitted 0 is used.**
P2 **- *Color to be replaced with Color in P3.***
If Color **P2** is prefixed with an **!** then the Command will work in a **negative mode**, means it will replace the Color if it does NOT Fit to the Color **P2**.
P3 **- *(opt.) Color that will replace the Color in P2,* if omitted &HFFFFFF (White) is used.**

The Command will leave a 0/1 on the TOS. If the Operation fails, **P2** may be "0".

Example

```
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' ANA.ReplaceColorD-Sample
'**********************************
VAF.$$PIA=?exeloc\Pics\Fux.jpg
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.AnalyzeColor|0|1
' Greenboard-Color, Mask and the color-range that shall be removed must be adjus
$$SRC=&H000000
$$COL=&H30FF30
ANA.ReplaceColorD|0|$$SRC|$$COL
DMP.Speed
ANA.Show|0!
ENR.

' Below you can see the result of the Operation.
```

© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000499 0001 VAF.$$PIA=?exeloc\Pics\Fux.jpg
014 0000000000259343 0001 ANA.Load|0|$$PIA
015 0000000029965062 0001 ANA.Show|0!
016 0000000001155371 0001 ANA.AnalyzeColor|0|1
017 0000000000000013 0001
018 0000000000000265 0001 VAR.$$SRC=&H000000
019 0000000000000226 0001 VAR.$$COL=&H30FF30
020 0000000000007357 0001 ANA.ReplaceColorD|0|$$SRC|$$COL
021 0000000000000000 0001 DMP.Speed
022 0000000000000000 0000 ANA.Show|0!
023 0000000000000000 0000 ENR.
```

Here is the Speed-Dump for the Operation

Remarks

This Command is really fast and will do its Job in less then a Second.

Limitations:

-

See also:

- ANA.AnalyeColor 1225
- ANA.RemoveIndexColors 1254

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.8 ANA.ReplaceColorT

MiniRobotLanguage (MRL)

ANA.ReplaceColorT

Replace Color with Tolerance - Replace given Colors with another Color directly in the IR, using a Tolerance-Value.

Intention

This Command will directly replace colors in an IR. It will replace Color P2 (and all COlors taht fit to P2 with given Tolerance P3) with Color P4. In Image-Register P1. This Command is NOT really an Color-Table Operation, as it does not need the Color-Table.
Instead it will directly Replace Color P2 with Color P3 in Image-Register P1 using the Tolerance, which is given in 0-100%.

If Color **P2** is prefixed with an **!** then the Command will work in a **negative mode**, means it will replace the Color if it does NOT Fit to the Color **P2**.

In this Script we will replace the Black Color using a Tolerance of 10%, with Green.

```
VAF.$$PIA=?exeloc\Pics\Fux.jpg
ANA.Load|0|$$PIA
ANA.Show|0!
$$SRC=&H000000
$$COL=&H30FF30
ANA.ReplaceColorT|0|$$SRC|10|$$COL
DMP.Speed
ANA.Show|0!
ENR.
```

Will bring up this Messagebox.

With this picture:

© 2023 Theo Gottwald * https://www.smart-package.com

Here is the Speed-Dump:

```
013 0000000000000429 0001 VAF.$$PIA=?exeloc\Pics\Fux.jpg
014 0000000000293838 0001 ANA.Load|0|$$PIA
015 0000000021768853 0001 ANA.Show|0!
016 0000000000000162 0001 VAR.$$SRC=&H000000
017 0000000000000343 0001 VAR.$$COL=&H30FF30
018 0000000000754420 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
019 0000000000000000 0001 DMP.Speed
020 0000000000000000 0000 ANA.Show|0!
021 0000000000000000 0000 ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.ReplaceColorT|P1|[!]P2[|P3][|P4]

Parameter Explanation

***P1 - Source & Destination* Image-Register Number, if omitted 0 is used.**
P2 - Color to be replaced with Color in P3.
If Color **P2** is prefixed with an **!** then the Command will work in a **negative mode**, means it will replace the Color if it does NOT Fit to the Color **P2**.
P3 - Color-Tolerance Value between 0-100 which is in Percent. 0 - do not use Tolerance, 100 - Replace any color.
P4 - (opt.) Color that will replace the Color in P2, if omitted &HFFFFFF (White) is used.

The Command will leave a 0/1 on the TOS. If the Operation fails, **P2** may be "0".

Example

This Example will show ANA.ReplaceColorT to make a Cleanup of a picture from a social network.
You got that Picture and it has a lot of JPG-Artefacts and the colors are not pure.
The SPR can quickly clean it up.

```
'**********************************
' ANA.ReplaceColorT-Sample
' Used to clean up JPG-Artefacts
' Used Sample picture are used for technical reasons and "may or may not be" the
'**********************************
VAF.$$PIA=?exeloc\Pics\Sam1.jpg
NEF.$$PIA
  MBX.File not found.
EIF.

ANA.Load|1|$$PIA
ANA.Load|0|$$PIA
ANA.Show|0!

'Restore red
$$SRC=&HFF0000
$$COL=&HFF0000
$$TOL=95
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL

' Restore Blue
$$SRC=&H0000FF
$$COL=$$SRC
$$TOL=90
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
```

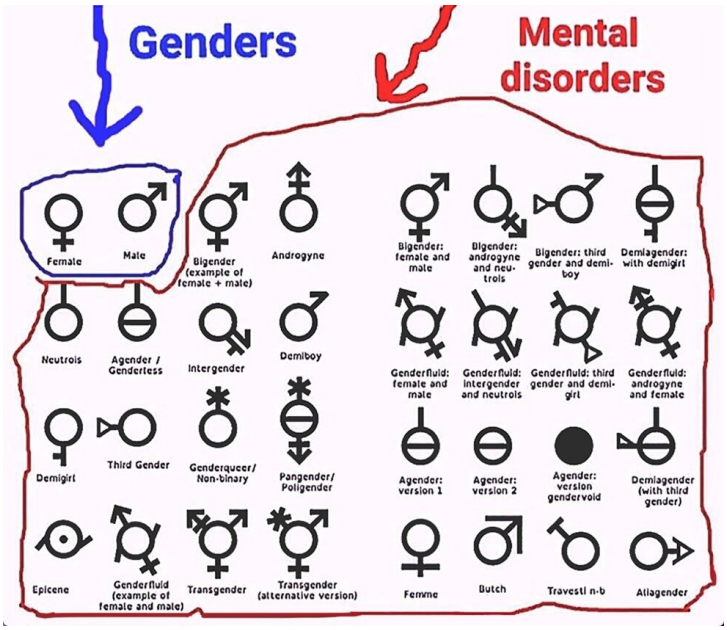

© 2023 Theo Gottwald * https://www.smart-package.com

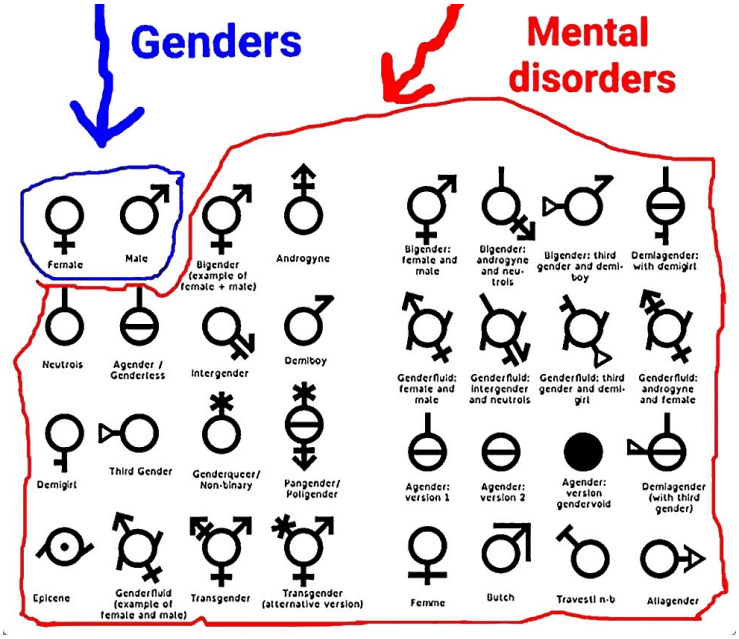
```
'Restore Black
$$SRC=&H0
$$COL=$$SRC
$$TOL=90
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL

'Restore White
$$SRC=&HFFFFFF
$$COL=$$SRC
$$TOL=90
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL

ANA.Show|0!

' Generate Outline
ANA.cdfxy|0|1|1
ANA.Contrast|1|15000

ANA.Invert|1
ANA.Show|1!

' IR 2 must not be empty for the MixTar Command
ANA.Copy|0|2

' Combine both
ANA.MixTar|1|0|2
DMP.1
ANA.Show|2!
ENR.


' Below you can see the result of the Operation.
```

Original Picture with Artefacts - pure Colors

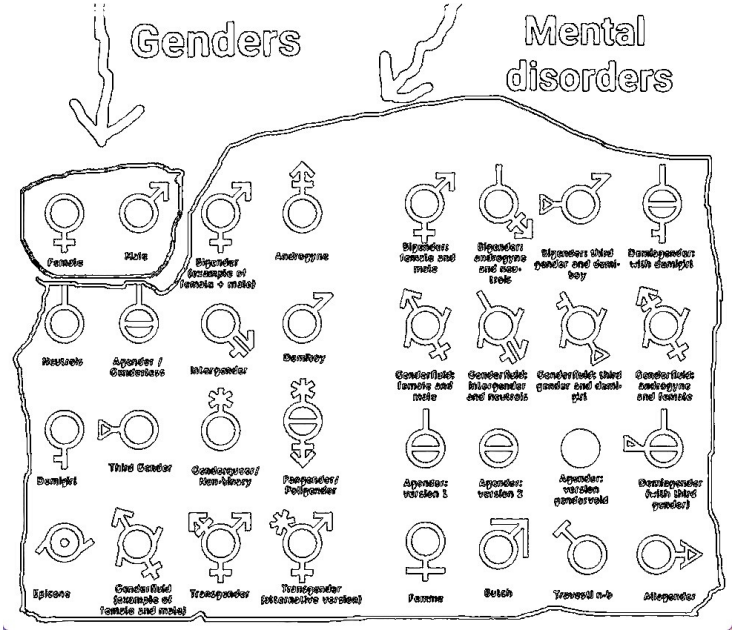
First Cleanup Generate Outline to clean up Borders

*© 2023 Theo Gottwald * https://www.smart-package.com*

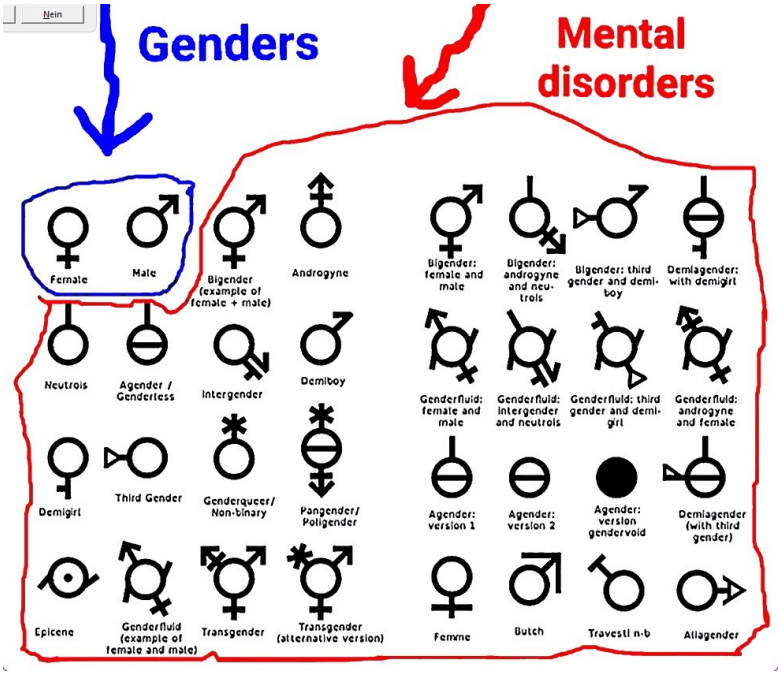




© 2023 Theo Gottwald * https://www.smart-package.com



Final Result



Here is the Speed-Dump of the Operation.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
010 0000000000000189 0001 VAF.$$PIA=?exeloc\Pics\Sam1.jpg
011 0000000000001934 0001 NEF.$$PIA
012 0000000000000000 0000 MBX.File not found.
013 0000000000000030 0001 EIF.
014 0000000000000005 0001
015 0000000000431193 0001 ANA.Load|1|$$PIA
016 0000000000389432 0001 ANA.Load|0|$$PIA
017 0000000029137881 0001 ANA.Show|0!
018 0000000000000009 0001
019 0000000000000003 0001
020 0000000000000150 0001 VAR.$$SRC=&HFF0000
021 0000000000000045 0001 VAR.$$COL=&HFF0000
022 0000000000000035 0001 VAR.$$TOL=95
023 0000000001883181 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
024 0000000000000010 0001
025 0000000000000003 0001
026 0000000000000084 0001 VAR.$$SRC=&H0000FF
027 0000000000000082 0001 VAR.$$COL=$$SRC
028 0000000000000039 0001 VAR.$$TOL=90
029 0000000001896552 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
030 0000000000000009 0001
031 0000000000000003 0001
032 0000000000000083 0001 VAR.$$SRC=&H0
033 0000000000000076 0001 VAR.$$COL=$$SRC
034 0000000000000036 0001 VAR.$$TOL=90
035 0000000001786782 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
036 0000000000000012 0001
037 0000000000000003 0001
038 0000000000000092 0001 VAR.$$SRC=&HFFFFFF
039 0000000000000079 0001 VAR.$$COL=$$SRC
040 0000000000000045 0001 VAR.$$TOL=90
041 0000000001765377 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
042 0000000000000008 0001
043 0000000025520293 0001 ANA.Show|0!
044 0000000000000009 0001
045 0000000021380221 0001 ANA.cdfxy|0|1|1
046 0000000008048198 0001 ANA.Contrast|1|15000
047 0000000000069639 0001 ANA.Invert|1
048 0000000017254000 0001 ANA.Show|1!
049 0000000000019495 0001 ANA.Copy|0|2
050 0000000000000008 0001
051 0000000000136976 0001 ANA.MixTar|1|0|2
052 0000000000000011 0001
053 0000000000000000 0001 DMP.1
054 0000000000000000 0000 ANA.Show|2!
055 0000000000000000 0000 ENR.
```

Here is the Speed-Dump for the Operation

This Script will split a Logo into its Color-Components.

```
'#EXE:?path\
'#SPI:ForceWrite

VAF.$$PIA=?exeloc\KAIF_Logo.png
NEF.$$PIA
  MBX.File not found.
EIF.

VAF.$$PIB=?exeloc\KAIF_Separate_
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Goldene Farbton
VAR.$$C01=&HB99534

' Rosa
VAR.$$C02=&HE5006D

' Weiss
VAR.$$C03=&HFFFFFF

' Schwarz
VAR.$$C04=&H0

' Default Background GRÜN
VAR.$$BGC=&H00FF00

' Load Source
VAN.$$BMA=1
ANA.Load|$$BMA|$$PIA
ANA.GetRes|$$BMA|$$XSZ|$$YSZ
'ANA.Show|$$BMA!

' Generate P2 für Gold
VAN.$$BMB=2
ANA.Copy|$$BMA|$$BMB

' Generate P3 für Rosa
VAN.$$BMC=3
ANA.Copy|$$BMA|$$BMC

' Generate P4 für Weiss
VAN.$$BMD=4
ANA.Copy|$$BMA|$$BMD

' Generate P5 für Schwarz
VAN.$$BME=5
ANA.Copy|$$BMA|$$BME

VAN.$$TOL=4
' Alles ausser Gold ersetzen durch Grün
ANA.ReplaceColorT|$$BMB|!$$C01|$$TOL|$$BGC
ANA.Show|$$BMB!

' Alles ausser Rosa ersetzen durch Grün
ANA.ReplaceColorT|$$BMC|!$$C02|$$TOL|$$BGC

' Alles ausser Weiss ersetzen durch Grün
ANA.ReplaceColorT|$$BMD|!$$C03|$$TOL|$$BGC

' Alles ausser Schwarz ersetzen durch Grün
ANA.ReplaceColorT|$$BME|!$$C04|$$TOL|$$BGC

VAN.$$TMB=#dsince#
DBP.$$TMB used
ANA.Show|$$BMB!
ANA.Show|$$BMC!
ANA.Show|$$BMD!
ANA.Show|$$BME!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$SAV|2|5
  VAR.$$NAM=$$PIBX$$SAV.png
  ANA.Save|$$SAV|$$NAM
NEX.
MBX.Done!
ENR.
```

Remarks

This Command is really fast and will do its Job in a Second. Of course it will be slower then ANA.ReplaceColorD. This is not really Color-Table Operation because it does not use the Color-Table.

Limitations:

-

See also:

- ANA.AnalyeColor 1225
- ANA.RemoveIndexColors 1254

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.4.9 ANA.Show...Array

MiniRobotLanguage (MRL)

ANA.ShowSimArray

Show the Content of the Col_BGL - Color-Table

ANA.ShowSortArray

Show the Content of the Cl_Srt - Color-Table

Intention

These Operations work with the "Color Table", this is a sorted list with all colors in the IR.
In fact there are 2 Color-Tables:
0 - **Color-Sort** (Col_Srt) - This table contains the Colors sorted by number of Pixel per Color.
The Top-Entry is the color with the most Pixels. The bottom Entry is the color with the least number of Pixels.
1 - **Color-Background** (Col_Bgl) - This table is sorted by similarity to the Color that has most Pixel, which is assumed to be the Background Color.
Here the Top-Entry contains the Background Color and then any further Entry is sorted by Similarity to this Color.

Before using these Commands the Color Tables must be generated. This is done, using:

```
ANA.AnalyzeColor|0|1
```

The Col_Bgl Table can be sorted to Similarity from another Color using:

```
ANA.Sort_Similar_To|&H26592A|$$MSK
```

Thee are several other commands that will only work if the Color-Table has been updated before.
If you want to see the Contents of the Color-Tables - for Debugging - you can use these two Commands.

```
ANA.ShowSimArray
ANA.ShowSortArray
```

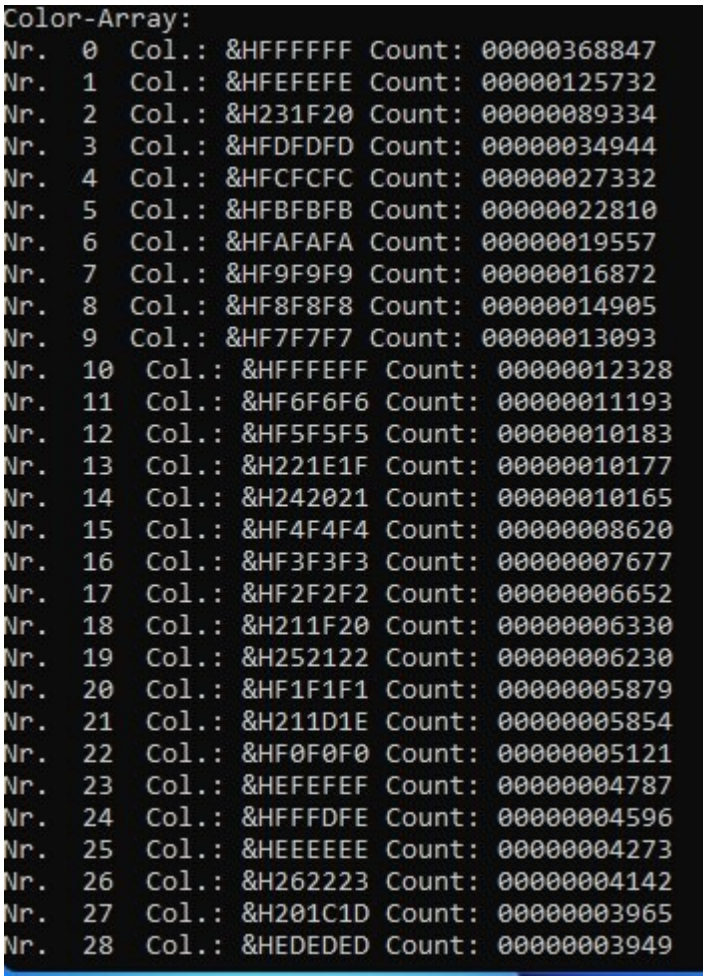
Using this Script:

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BTC2.jpg
ANA.Load|0|$$PIA
ANA.AnalyzeColor|0|1
ANA.showsortarray|0
MBX.!
ENR.
```

You will get a listing of the Col_Srt-Table, this may look like this:

© 2023 Theo Gottwald * https://www.smart-package.com

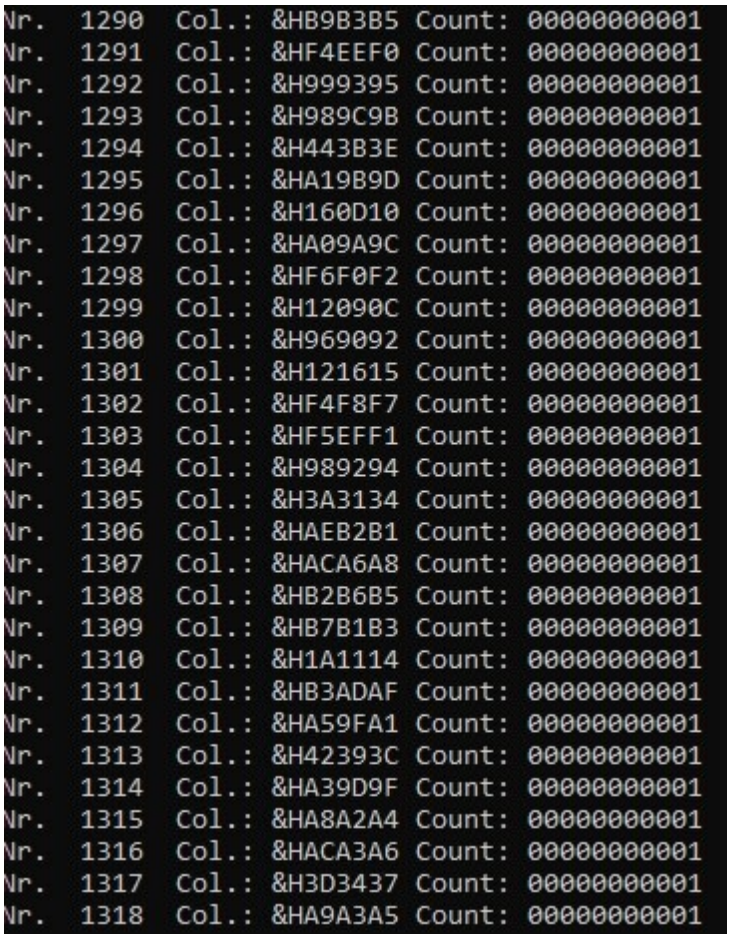
© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

```
Nr.  35  Col.: &H1F1B1C Count: 00000003042
Nr.  36  Col.: &H201E1F Count: 00000002931
Nr.  37  Col.: &HEAEAEA Count: 00000002906
Nr.  38  Col.: &HE9E9E9 Count: 00000002720
Nr.  39  Col.: &HFCFAFB Count: 00000002698
Nr.  40  Col.: &H000000 Count: 00000002558
Nr.  41  Col.: &HE8E8E8 Count: 00000002524
Nr.  42  Col.: &HE7E7E7 Count: 00000002469
Nr.  43  Col.: &H282425 Count: 00000002434
Nr.  44  Col.: &H232122 Count: 00000002349
Nr.  45  Col.: &HFBF9FA Count: 00000002336
Nr.  46  Col.: &H1F1D1E Count: 00000002286
Nr.  47  Col.: &HE5E5E5 Count: 00000002168
Nr.  48  Col.: &H1E1A1B Count: 00000002166
Nr.  49  Col.: &HE6E6E6 Count: 00000002152
Nr.  50  Col.: &HFEFFFF Count: 00000002149
Nr.  51  Col.: &H241E20 Count: 00000002123
Nr.  52  Col.: &HFAF8F9 Count: 00000002059
Nr.  53  Col.: &H292526 Count: 00000001916
Nr.  54  Col.: &H242223 Count: 00000001910
Nr.  55  Col.: &H1E1C1D Count: 00000001903
Nr.  56  Col.: &H1D191A Count: 00000001856
Nr.  57  Col.: &HE4E4E4 Count: 00000001776
Nr.  58  Col.: &HF9F7F8 Count: 00000001755
Nr.  59  Col.: &HE3E3E3 Count: 00000001664
Nr.  60  Col.: &H252324 Count: 00000001604
Nr.  61  Col.: &H1D1B1C Count: 00000001596
Nr.  62  Col.: &H2A2627 Count: 00000001574
Nr.  63  Col.: &HF8F6F7 Count: 00000001541
Nr.  64  Col.: &HE1E1E1 Count: 00000001509
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Start of the Color-Tabel, Entries 0 to 41. the Entry for "Black" is on position 40.

Second part. As you can see, End of the Color-Table. Entries up to 1318.

Using this commands you can work with the list entries in the color table and do significant improvements in pictures.
Now you can improve the image. All colors are already sorted by Quantity (How many Pixels).

You can also get a sort on "Color difference to background". For this you can use:

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BTC2.jpg
ANA.Load|0|$$PIA
ANA.AnalyzeColor|0|1
ANA.ShowSimArray|0
MBX.!
ENR.
```

Then you will get the Content of The Col_BGL-Table which looks somehow like this:

© 2023 Theo Gottwald * https://www.smart-package.com

System Messages

```
similarity-Array:
Nr.  0  Col.: &H000000 Count: 00000002558 Color-Diff.:  428
Nr.  1  Col.: &H000100 Count: 00000000021 Color-Diff.:  162
Nr.  2  Col.: &H010000 Count: 00000000405 Color-Diff.:  333
Nr.  3  Col.: &H010101 Count: 00000000644 Color-Diff.:  368
Nr.  4  Col.: &H000201 Count: 00000000014 Color-Diff.:  139
Nr.  5  Col.: &H030000 Count: 00000000202 Color-Diff.:  305
Nr.  6  Col.: &H020001 Count: 00000000127 Color-Diff.:  266
Nr.  7  Col.: &H040001 Count: 00000000064 Color-Diff.:  218
Nr.  8  Col.: &H040000 Count: 00000000005 Color-Diff.:  66
Nr.  9  Col.: &H050001 Count: 00000000001 Color-Diff.:  28
Nr.  10  Col.: &H010302 Count: 00000000005 Color-Diff.:  72
Nr.  11  Col.: &H020202 Count: 00000000647 Color-Diff.:  371
Nr.  12  Col.: &H030102 Count: 00000000148 Color-Diff.:  282
Nr.  13  Col.: &H060002 Count: 00000000002 Color-Diff.:  46
Nr.  14  Col.: &H050102 Count: 00000000081 Color-Diff.:  227
Nr.  15  Col.: &H020403 Count: 00000000006 Color-Diff.:  85
Nr.  16  Col.: &H030303 Count: 00000000634 Color-Diff.:  366
Nr.  17  Col.: &H040203 Count: 00000000156 Color-Diff.:  286
Nr.  18  Col.: &H050304 Count: 00000000156 Color-Diff.:  286
Nr.  19  Col.: &H040404 Count: 00000000678 Color-Diff.:  378
Nr.  20  Col.: &H070103 Count: 00000000002 Color-Diff.:  45
Nr.  21  Col.: &H030504 Count: 00000000005 Color-Diff.:  66
Nr.  22  Col.: &H060203 Count: 00000000080 Color-Diff.:  226
Nr.  23  Col.: &H080204 Count: 00000000001 Color-Diff.:  20
Nr.  24  Col.: &H070304 Count: 00000000087 Color-Diff.:  229
Nr.  25  Col.: &H050505 Count: 00000000683 Color-Diff.:  379
Nr.  26  Col.: &H060405 Count: 00000000180 Color-Diff.:  298
Nr.  27  Col.: &H040605 Count: 00000000007 Color-Diff.:  87
Nr.  28  Col.: &H080405 Count: 00000000103 Color-Diff.:  240
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
Nr.  1290  Col.: &HF9FBFA Count: 00000000376 Color-Diff.:  330
Nr.  1291  Col.: &HFEF8FA Count: 00000000005 Color-Diff.:  68
Nr.  1292  Col.: &HFDF9FA Count: 00000000606 Color-Diff.:  357
Nr.  1293  Col.: &HFCFAFB Count: 00000002698 Color-Diff.:  428
Nr.  1294  Col.: &HFBFBFB Count: 00000022810 Color-Diff.:  440
Nr.  1295  Col.: &HFFF9FB Count: 00000000007 Color-Diff.:  94
Nr.  1296  Col.: &HFAFCFB Count: 00000000371 Color-Diff.:  330
Nr.  1297  Col.: &HFEFAFB Count: 00000000633 Color-Diff.:  365
Nr.  1298  Col.: &HFFFAFC Count: 00000000005 Color-Diff.:  69
Nr.  1299  Col.: &HFDFBFC Count: 00000003121 Color-Diff.:  430
Nr.  1300  Col.: &HFAFEFD Count: 00000000001 Color-Diff.:  23
Nr.  1301  Col.: &HFCFCFC Count: 00000027332 Color-Diff.:  440
Nr.  1302  Col.: &HFBFDFC Count: 00000000458 Color-Diff.:  335
Nr.  1303  Col.: &HFFFBFC Count: 00000000685 Color-Diff.:  380
Nr.  1304  Col.: &HFCFEFD Count: 00000000591 Color-Diff.:  353
Nr.  1305  Col.: &HFFFBFD Count: 00000000005 Color-Diff.:  66
Nr.  1306  Col.: &HFFFCFD Count: 00000000750 Color-Diff.:  394
Nr.  1307  Col.: &HFEFCFD Count: 00000003515 Color-Diff.:  431
Nr.  1308  Col.: &HFBFFFE Count: 00000000001 Color-Diff.:  21
Nr.  1309  Col.: &HFFFCFE Count: 00000000003 Color-Diff.:  54
Nr.  1310  Col.: &HFCFFFF Count: 00000000008 Color-Diff.:  104
Nr.  1311  Col.: &HFDFDFD Count: 00000034944 Color-Diff.:  440
Nr.  1312  Col.: &HFDFFFE Count: 00000000757 Color-Diff.:  395
Nr.  1313  Col.: &HFEFEFE Count: 00000125732 Color-Diff.:  441
Nr.  1314  Col.: &HFFFDFF Count: 00000000011 Color-Diff.:  126
Nr.  1315  Col.: &HFFFDFE Count: 00000004596 Color-Diff.:  434
Nr.  1316  Col.: &HFFFEFF Count: 00000012328 Color-Diff.:  438
Nr.  1317  Col.: &HFEFFFF Count: 00000002149 Color-Diff.:  424
Nr.  1318  Col.: &HFFFFFF Count: 00000368847 Color-Diff.:  442
```

In this table the black colors will be "on Top" while the White colors will be "bottom". For this sort the robot assumes that the color that is mostly in the picture "is the background color".
The he will sort all colors "by difference to the background". This is the result.

Using these tables you can significantly improve pictures with your Script.

For example you can remove colors from the list and the IR using:

```
' this command will remove the given color and replace it with another color.
ANA.removeindexcolors|0|300|$$CNV|0

VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
'VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
GSB.Reanalyze
ANA.removeindexcolors|0|300|$$CNV|0
ANA.removeindexcolors|0|$$CNV|$$CNT|-1
GSB.Reanalyze
ANA.removeindexcolors|0|$$CNV|$$CNT|-1
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Reanalyze
ANA.AnalyzeColor|0|1
ANA.getcolorcount|0|$$CNT
ANA.getbgcolor|0|$$BGC
CAL.$$CNV=$$CNT-400
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

Original Picture

Picture-Result

With a few changes, you can see in detail how the Script works:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Here we have changed the script a bit to show
' how the commands paint the colors.
' Green -> would be painted in white
' red -> would be painted in black
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
'VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
GSB.Reanalyze
ANA.removeindexcolors|0|300|$$CNV|&HFF0000
GSB.Reanalyze
ANA.removeindexcolors|0|$$CNV|($$CNT-20)|&H00FF00
ANA.removeindexcolors|0|($$CNT-20)|$$CNT|&HFFFFFF
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.

:Reanalyze
ANA.AnalyzeColor|0|1
ANA.getcolorcount|0|$$CNT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.getbgcolor|0|$$BGC
CAL.$$CNV=$$CNT-400
RET.
```

```
010 0000000000000694 0001 VAN.$$TIM=#time#
011 0000000000000525 0001 VAF.$$PIA=?exeloc\BNB.jpg
012 0000000000000016 0001
013 0000000000000383 0001 VAF.$$PIB=?exeloc\V1.png
014 0000000001215801 0001 ANA.Load|0|$$PIA
015 0000000000000118 0001 GSB.Reanalyze
016 0000000010022301 0001 ANA.removeindexcolors|0|300|$$CNV|&HFF0000
017 0000000000000118 0001 GSB.Reanalyze
018 0000000026517894 0001 ANA.removeindexcolors|0|$$CNV|($$CNT-20)|&H00FF00
019 0000000000912936 0001 ANA.removeindexcolors|0|($$CNT-20)|$$CNT|&HFFFFFF
020 0000000000001814 0001 CAL.$$TIM=#time#-$$TIM
021 0000000000006267 0001 DBP.Used Time in Seconds: $$TIM
022 0000000000000000 0001 DMP.1
023 0000000000000000 0000 ANA.Save|0|$$PIB
024 0000000000000000 0000 ANA.Show|0!
025 0000000000000000 0000 ENR.
026 0000000000000000 0000
027 0000000000000018 0002 :Reanalyze
028 0000000000901451 0002 ANA.AnalyzeColor|0|1
029 0000000000001675 0002 ANA.getcolorcount|0|$$CNT
030 0000000000000730 0002 ANA.getbgcolor|0|$$BGC
031 0000000000000968 0002 CAL.$$CNV=$$CNT-400
032 0000000000000208 0002 RET.
033 0000000000000000 0000
```

This is the Speed-Dump of the Script. It uses just few Seconds for the complete run.

Syntax

ANA.ShowSimArray[|P1]
ANA.ShowSortArray[|P1]

Parameter Explanation

P1 **- *Source* Image-Register Number**

Example

-

Remarks

-

Limitations:

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.5 Color-Calculation

Color-Calculations take one or more 24-Bit Color Values as Input and calculate an result or result color from that.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.5.1 ANA. Inverse_Color / Complementary_Color

ANA.Complementary_Color / ANA. Inverse_Color Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.Complementary_Color

Calculate the Complementary Color Value for a given 24-bit Color-Value

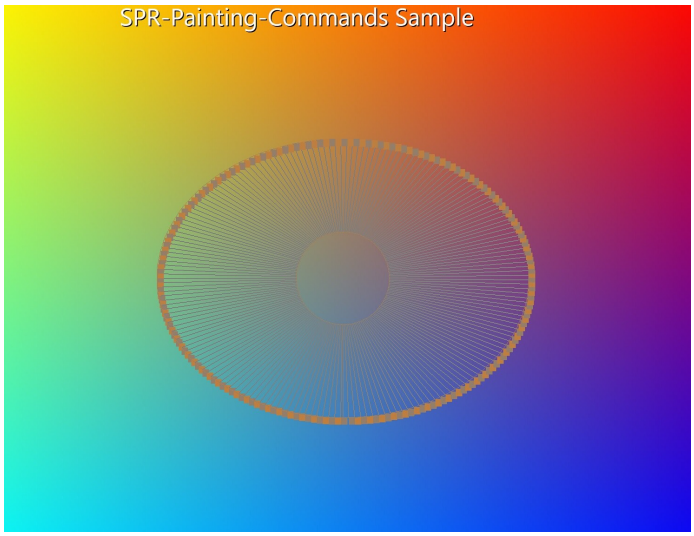
ANA.Inverse_Color

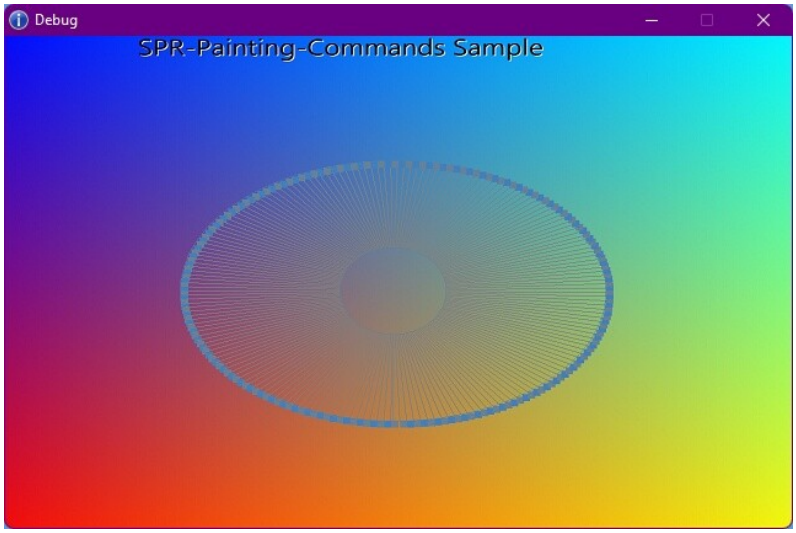
Calculate the Inverse Color Value for a given 24-bit Color-Value

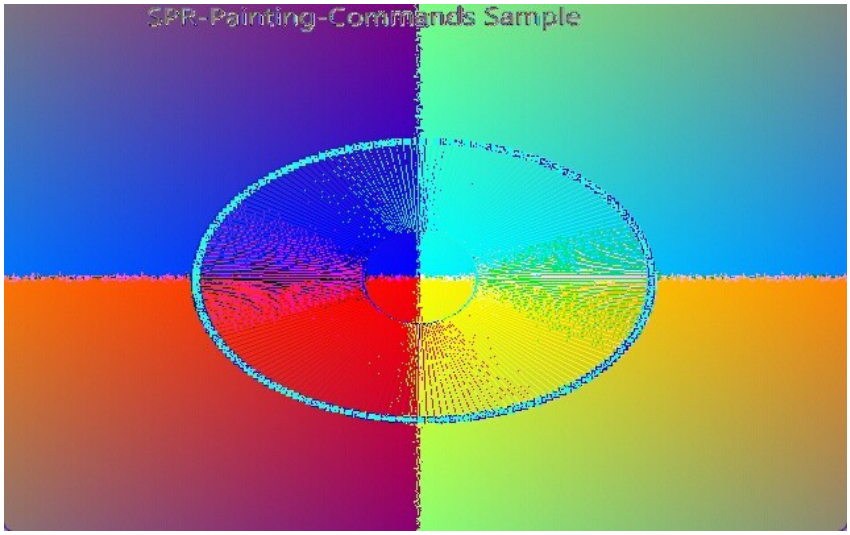
Intention

This command will return the Inverse / or complementary color value of a given 24-bit Color-Value.
To show the difference of both here is an example:

© 2023 Theo Gottwald * https://www.smart-package.com







Original Picture **Inverse Color**

Complementary Color

The Complementary Color is very good to use as Text on Pixels. The Inverse Color is exactly the Inverse Color in the Color-Wheel.

```
VAF.$$PIA=?exeloc\Colp.jpg
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.ResizeTo|0|1|640|400
ANA.Copy|1|0
ANA.Copy|1|2
ANA.GetRes|1|$$XRE|$$YRE

FOR.$$LOX|0|$$XRE
  PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|$$YRE
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB

    ANA.Inverse_Color|$$PIB|$$PIC
    ANA.Complementary_Color|$$PIB|$$PID

    ANA.Setpixelcolor|1|$$LOX|$$LOY|$$PIC
    ANA.Setpixelcolor|2|$$LOX|$$LOY|$$PID
  NEX.
NEX.
ANA.Compare|1|2|0
POP.$$RES
PRT.Result of Compare: $$RES

GSB.Save|1
GSB.Save|2
ENR.
```

Here is another Version of the Script using a Picture with a Person.

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Original Picture **ANA.Inverse_Color** **ANA.Complementary_Color** **Complement of Inverted Color**

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.Copy|0|1
ANA.Copy|0|2
ANA.Copy|0|3
ANA.GetRes|0|$$XRE|$$YRE

FOR.$$LOX|0|$$XRE
  PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|$$YRE
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB

    ANA.Inverse_Color|$$PIB|$$PIC
    ANA.Complementary_Color|$$PIB|$$PID

    ANA.Setpixelcolor|1|$$LOX|$$LOY|$$PIC
    ANA.Setpixelcolor|2|$$LOX|$$LOY|$$PID
  NEX.
NEX.
' Show Inverted Picture
GSB.Save|1

' Complement Inverted Picture
FOR.$$LOX|0|$$XRE
  PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|$$YRE
    ANA.getpixelcolor|1|$$LOX|$$LOY|$$PIB
    ANA.Complementary_Color|$$PIB|$$PID
    ANA.Setpixelcolor|1|$$LOX|$$LOY|$$PID
    'Complement Complemented Color
    ANA.Complementary_Color|$$PID|$$PIC
    ANA.Setpixelcolor|3|$$LOX|$$LOY|$$PIC
  NEX.
NEX.

' Invert Result from Complementry Color in one strike
GSB.Save|2
ANA.Invert|2

DMP.1
GSB.Save|1
GSB.Save|2
GSB.Save|3

ENR.

'============================================================
:Save
$$NUM=§§_01
RND.0|999999|$$RND
FMT.$$RND|000000
$$PAT=?exeloc\Save_$$RND.png
IEF.$$PAT
  GTO.Save
EIF.
ANA.Save|$$NUM|$$PAT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Show|$$NUM!
RET.
'============================================================
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
010 0000000000000467 0001 VAF.$$PIA=?exeloc\Sara2.jpg
011 0000000000001682 0001 NEF.$$PIA
012 0000000000000000 0000 MBX.No such file.
013 0000000000000000 0000 ENR.
014 0000000000000063 0001 EIF.
015 0000000000394610 0001 ANA.Load|0|$$PIA
016 0000000045770273 0001 ANA.Show|0!
017 0000000000015819 0001 ANA.Copy|0|1
018 0000000000015020 0001 ANA.Copy|0|2
019 0000000000014757 0001 ANA.Copy|0|3
020 0000000000000334 0001 ANA.GetRes|0|$$XRE|$$YRE
021 0000000000000005 0001
022 0000000000000080 0641 FOR.$$LOX|0|$$XRE
023 0000000000001215 0640 PRT.Column: $$LOX/$$XRE
024 0000000000000083 615040 FOR.$$LOY|0|$$YRE
025 0000000000000237 614400 ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB
026 0000000000000003 614400
027 0000000000000167 614400 ANA.Inverse_Color|$$PIB|$$PIC
028 0000000000000180 614400 ANA.Complementary_Color|$$PIB|$$PID
029 0000000000000003 614400
030 0000000000000241 614400 ANA.Setpixelcolor|1|$$LOX|$$LOY|$$PIC
031 0000000000000238 614400 ANA.Setpixelcolor|2|$$LOX|$$LOY|$$PID
032 0000000000000005 615040 NEX.
033 0000000000000005 0641 NEX.
034 0000000000000215 0001 GSB.Save|1
035 0000000000000004 0001
036 0000000000000002 0001
037 0000000000000073 0641 FOR.$$LOX|0|$$XRE
038 0000000000001846 0640 PRT.Column: $$LOX/$$XRE
039 0000000000000075 615040 FOR.$$LOY|0|$$YRE
040 0000000000000229 614400 ANA.getpixelcolor|1|$$LOX|$$LOY|$$PIB
041 0000000000000166 614400 ANA.Complementary_Color|$$PIB|$$PID
042 0000000000000220 614400 ANA.Setpixelcolor|1|$$LOX|$$LOY|$$PID
043 0000000000000002 614400
044 0000000000000168 614400 ANA.Complementary_Color|$$PID|$$PIC
045 0000000000000218 614400 ANA.Setpixelcolor|3|$$LOX|$$LOY|$$PIC
046 0000000000000005 615040 NEX.
047 0000000000000004 0641 NEX.
048 0000000000000002 0001
049 0000000000000002 0001
050 0000000000000140 0001 GSB.Save|2
051 0000000000055153 0001 ANA.Invert|2
052 0000000000000026 0001
053 0000000000000000 0000
054 0000000000000000 0000
055 0000000000000000 0000
056 0000000000000000 0001 DMP.1
057 0000000000000000 0000 GSB.Save|1
058 0000000000000000 0000 GSB.Save|2
059 0000000000000000 0000 GSB.Save|3
060 0000000000000000 0000
061 0000000000000000 0000 ENR.
062 0000000000000000 0000
063 0000000000000000 0000
064 0000000000000005 0002 :Save
065 0000000000000048 0002 VAR.$$NUM=ºº_01
066 0000000000000078 0002 RND.0|999999|$$RND
067 0000000000000135 0002 FMT.$$RND|000000
068 0000000000000083 0002 VAR.$$PAT=?exeloc\Save_$$RND.png
069 0000000000001190 0002 IEF.$$PAT
070 0000000000000000 0000 GTO.Save
071 0000000000000006 0002 EIF.
072 0000000000588361 0002 ANA.Save|$$NUM|$$PAT
073 0000000077697213 0002 ANA.Show|$$NUM!
074 0000000000000042 0002 RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.Complementary_Color[|P1][|P2][|P3][|P4]
ANA.Inverse_Color[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 - Source Color Value (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Variable for Color-Value for ***Red***
P3 - Variable for Color-Value for ***Green***
P4 - Variable for Color-Value for ***Blue***

If any of the Variables is missing, the value is been placed on the TOS.

Example

```
See above.
```

Remarks

The `ANA.Split_Color_RGB` is optimized for Speed. It will use only about 444 Ticks. Therefore the Robot could split more then 36000 Values per Second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.5.2 ANA.Add/Sub_Color / Mix_Colors

MiniRobotLanguage (MRL)

ANA.Sub_Color
Subtract Color P2 from Color P1
ANA.Add_2_Colors
Add 2 Colors additive
ANA.Add_3_Colors
Add 3 Colors additive
ANA.Add_4_Colors
Add 4 Colors additive
ANA.Mix_2_Colors
Average 2 Colors additive
ANA.Mix_3_Colors
Average 3 Colors additive
ANA.Mix_4_Colors
Average 4 Colors additive

Intention
Use the SPR like a box with watercolors. Add colors and use them for Painting. Also you can subtract colors -
that is somehow like watching through a colored glass.
These command do color calculations to Mix or Subtract colors.
For this it will internally split the colors into its three 8-Bit Color-Components for R,G and B.
And use these for the final calculation.

What is the difference between ADD and MIX?
Using **Add**, the color-values will just be added to each other therefore the result will be an real addition.
Using **Mix**, the color-values will be added and then divided by the number of Sources therefore the result will be an average of the sources.

While that does not Sound much, it enables you to do unlimited complex color Mixing and processing.
You can use Colors like physical colors that you can mix together and paint with these.

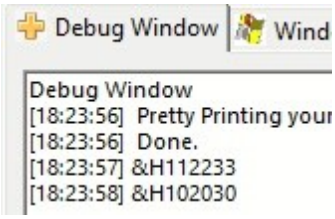
1. Color-Calculations

In this Sample we will just add and subtract some colors.

```
$$COA=&H102030
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$COB=&H010203
ANA.Mix2|$$COA|$$COB|$$RES
VTH.$$RES
DBP.&H$$RES
ANA.Sub|&H$$RES|$$COB|$$REX
VTH.$$REX
DBP.&H$$REX
ENR.
```



Here you can see the result.

2. Filter Color Bandwith

Here we can use this to clean up a picture and remove artefacts.

.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BNB.jpg
VAF.$$PIB=?exeloc\V2.jpg
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
CAL.$$FAC=&H909090
FOR.$$YPO|1|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PAX
    ANA.Sub|$$PAX|$$FAC|$$PAX
    ANA.Add2|$$PAX|&HF0F0F0|$$PBX
    ANA.Sub|$$PBX|$$FAC|$$PAX
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PAX
  NEX.
NEX.
RCC.0|&H0|&H61|0|&H61|0|&H61|0
RCC.0|&HC6e|&HFF|&HC6e|&HFF|&HC6e|&HFF|&HFFFFFF
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

The Script will remove all watermarks and JPG-Artefacts and also darken the picture.

The Speed-Dump looks like this:

```
012 0000000000000816 0001 VAN.$$TIM=#time#
013 0000000000000243 0001 VAF.$$PIA=?exeloc\BNB.jpg
014 0000000000000052 0001 VAF.$$PIB=?exeloc\V2.png
015 0000000001280879 0001 ANA.Load|0|$$PIA
016 0000000000000561 0001 ANA.GetRes|0|$$XSZ|$$YSZ
017 0000000000000221 0001 CAL.$$FAC=&H909090
018 0000000000000086 1341 FOR.$$YPO|1|$$YSZ
019 0000000000001911 1340 PRT.$$YPO/$$YSZ
020 0000000000000089 1796940 FOR.$$XPO|1|$$XSZ
021 0000000000000252 1795600 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PAX
022 0000000000000219 1795600 ANA.Sub|$$PAX|$$FAC|$$PAX
023 0000000000000194 1795600 ANA.Add2|$$PAX|&HF0F0F0|$$PBX
024 0000000000000217 1795600 ANA.Sub|$$PBX|$$FAC|$$PAX
025 0000000000000237 1795600 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PAX
026 0000000000000005 1796940 NEX.
027 0000000000000005 1341 NEX.
028 0000000000304501 0001 RCC.0|&H0|&H61|0|&H61|0|&H61|0
029 0000000000342562 0001 RCC.0|&HC6e|&HFF|&HC6e|&HFF|&HC6e|&HFF|&HFFFFFF
030 0000000000000433 0001 CAL.$$TIM=#time#-$$TIM
031 0000000000000007 0001
```

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.Sub_Color|P1|P2[|P3]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Variable the result.

ANA.Add_2_Colors|P1|P2[|P3]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Variable the result.

ANA.Add_3_Colors|P1|P2|P3[|P4]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Color Value C (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P4 - Variable the result.

ANA.Add_4_Colors|P1|P2|P3|P4[|P5]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Color Value C (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P4 - Color Value D (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P5 - Variable the result.

*© 2023 Theo Gottwald * https://www.smart-package.com*

ANA.Mix_2_Colors|P1|P2[|P3]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Variable the result.

ANA.Mix_3_Colors|P1|P2|P3[|P4]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Color Value C (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P4 - Variable the result.

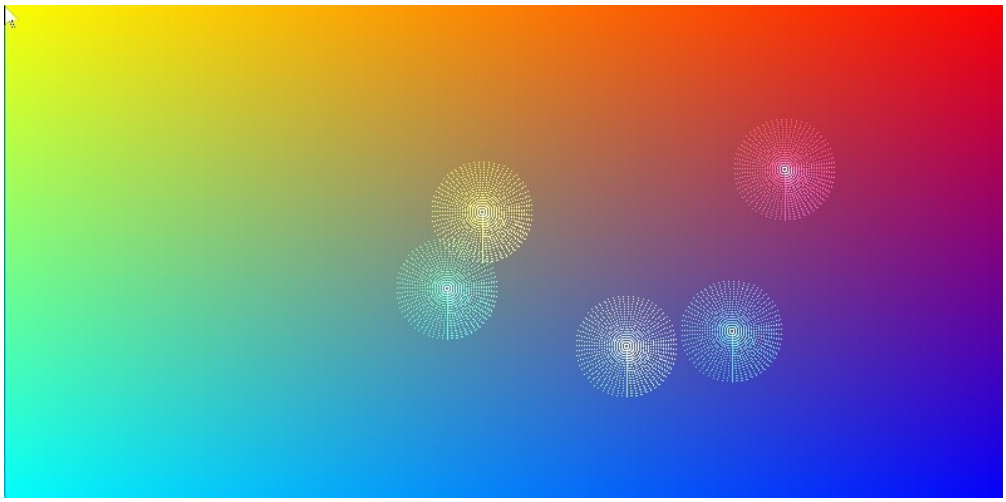
ANA.Mix_4_Colors|P1|P2|P3|P4[|P5]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P3 - Color Value C (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P4 - Color Value D (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P5 - Variable the result.

Example

© 2023 Theo Gottwald * https://www.smart-package.com



```
' Paint 5 "Fingerprints"
VAN.$$TIM=#time#
ANA.New|1|800|400
ANA.vgradient|1|&HFF00FF|&H000000
ANA.New|0|800|400
ANA.hgradient|0|&H0000FF|&H00FFFF
ANA.Mix|1|0|7
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$LOA|1|5
   PRT.$$LOA
  RND.100|($$XSZ-100)|$$XPA
  RND.100|($$YSZ-100)|$$YPA
  RND.0|&H808080|$$COL
  GSB.PaintCirc
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Save|0|$$PIB
ANA.Show|0!
ENR.
'-------------------------------------------------------------
:PaintCirc
FOR.$$LIB|2|40|2
  FOR.$$LOP|0|8|0.1
    CAX.$$XPO=SIN($$LOP)*$$LIB+$$XPA|i
    CAX.$$YPO=COS($$LOP)*$$LIB+$$YPA|i
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PAX
    ANA.Add2|$$PAX|$$COL|$$PBX
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PBX
  NEX.
NEX.
RET.
```

Here is the Speed-Dump for the Script.

© 2023 Theo Gottwald * https://www.smart-package.com

```
011 0000000000000758 0001 VAN.$$TIM=#time#
012 0000000000016010 0001 ANA.New|1|800|400
013 0000000000032791 0001 ANA.vgradient|1|&HFF00FF|&H000000
014 0000000000010320 0001 ANA.New|0|800|400
015 0000000000074774 0001 ANA.hgradient|0|&H0000FF|&H00FFFF
016 0000000000050216 0001 ANA.Mix|1|0|7
017 0000000000000492 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000065 0006 FOR.$$LOA|1|5
019 0000000000001068 0005 PRT.$$LOA
020 0000000000000268 0005 RND.100|($$XSZ-100)|$$XPA
021 0000000000000194 0005 RND.100|($$YSZ-100)|$$YPA
022 0000000000000073 0005 RND.0|&H808080|$$COL
023 0000000000000036 0005 GSB.PaintCirc
024 0000000000000004 0006 NEX.
025 0000000000000289 0001 CAL.$$TIM=#time#-$$TIM
026 0000000000000003 0001
027 0000000000000000 0001 DMP.1
028 0000000000000000 0000 ANA.Save|0|$$PIB
029 0000000000000000 0000 ANA.Show|0!
030 0000000000000000 0000 ENR.
031 0000000000000000 0000
032 0000000000000005 0005 :PaintCirc
033 0000000000000075 0105 FOR.$$LIB|2|40|2
034 0000000000000076 8200 FOR.$$LOP|0|8|0.1
035 0000000000046443 8100 CAX.$$XPO=SIN($$LOP)*$$LIB+$$XPA|i
036 0000000000043196 8100 CAX.$$YPO=COS($$LOP)*$$LIB+$$YPA|i
037 0000000000000301 8100 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PAX
038 0000000000000233 8100 ANA.Add2|$$PAX|$$COL|$$PBX
039 0000000000000239 8100 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PBX
040 0000000000000003 8100
041 0000000000000005 8200 NEX.
042 0000000000000004 0105 NEX.
043 0000000000000018 0005 RET.
```

Remarks

The `ANA.Sub/Add/Mix Commands` are optimized for Speed. They will only use about 210 Ticks. Therefore the Robot could work on more then 76000 Values per Second.

Limitations:

-

See also:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.5.3 ANA.Color_Diff-Commands

MiniRobotLanguage (MRL)

ANA.Color_Diff_RGB and other ...

Split the a specified Color-Value into R,G and B

Intention

These commands will take 2 Color-Values and calculate the Color-Difference between these values. Internally the Color-Difference is the Sum of the Differences between the R,G and B Parts of both values.
The smaller the Color Difference, the the more equal the colors should be.
Generally most Information you can get using:

```
ANA.Color_Diff_RGB|&HF0F1F2|&HFFAADD|$$RED|$$GRN|$$BLUE
```

Here you will get all 3 Color-Differences between the 2 given colors, in 3 Variables.
Then you can calculate yourself whatever you are interested in.

The other Commands try to do some Speed-Up for special purpose Operations.

```
' This Command will return only the largest Color-Distance
ANA.Color_Diff_Max|&HF0F1F2|&HFFAADD|$$MAX

' This Command will return only the smallest Color-Distanc
ANA.Color_Diff_Min|&HF0F1F2|&HFFAADD|$$MIN

' This Command will return only the RED Color-Distance betw
ANA.Color_Diff_R|&HF0F1F2|&HFFAADD|$$RED

' This Command will return only the GREEN Color-Distance be
ANA.Color_Diff_G|&HF0F1F2|&HFFAADD|$$GRN

' This Command will return only the BLUE Color-Distance bet
ANA.Color_Diff_B|&HF0F1F2|&HFFAADD|$$BLU
```

1. Sample for ANA.Color_Diff_Max

Normally this Command can be used to find structures in GUI's, for example Letters.
Here we can use it for some Artwork.
We will find Borders and paint then red.

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIC=?pattern\SarahC_640.jpg
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
ANA.New|2|$$XSZ|$$YSZ
$$COL=&H808080
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelColor|0|$$XPO|$$YPO|$$COL
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    ANA.Color_Diff_Max|$$LAS|$$COL|$$RES
    CAL.$$RES=$$RES*6
    ANA.SetPixelRGB|2|$$XPO|$$YPO|$$RES|$$GRN|$$BLU
    $$LAS=$$COL
  NEX.
NEX.
ANA.Save|2|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|2!
MBX.Used Time in Seconds: $$TIM
ENR.
```

And then Paint the Picture.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

It Needs about 250 Seconds to find the borders and paint them red.

```
013 0000000000000434 0001 VAN.$$TIM=#time#
014 0000000000000150 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000070 0001 VAF.$$PIC=?pattern\SarahC_640.jpg
016 0000000000468321 0001 ANA.Load|0|$$PIA
017 0000000000000732 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000022881 0001 ANA.New|2|$$XSZ|$$YSZ
019 0000000000000350 0001 VAR.$$COL=&H808080
020 0000000000000137 0963 FOR.$$YPO|1|$$YSZ
021 0000000000072434 0962 DBP.$$YPO/$$YSZ
022 0000000000000139 616642 FOR.$$XPO|1|$$XSZ
023 0000000000000351 615680 ANA.GetPixelColor|0|$$XPO|$$YPO|$$COL
024 0000000000000536 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
025 0000000000000308 615680 ANA.Color_Diff_Max|$$LAS|$$COL|$$RES
026 0000000000000181 615680 CAL.$$RES=$$RES*6
027 0000000000000462 615680 ANA.SetPixelRGB|2|$$XPO|$$YPO|$$RES|$$GRN|$$BLU
028 0000000000000065 615680 VAR.$$LAS=$$COL
029 0000000000000006 616642 NEX.
030 0000000000000006 0963 NEX.
031 0000000000414614 0001 ANA.Save|2|$$PIC
032 0000000000000392 0001 CAL.$$TIM=#time#-$$TIM
033 0000000000256763 0001 DBP.Used Time in Seconds: $$TIM
034 0000000000000000 0001 DMP.1
```

This is the Speed-Dump of the Operation.

2. Multi-Step Picture Processing

*© 2023 Theo Gottwald * https://www.smart-package.com*

Here we have a 2-Pass Picture Processing. As the SPR can use multiple IR's, any intelligent sort of Picture Processing can be done in as man steps as needed.
In the first step we find the borders like in the picture above. In the second step we paint the borders larger.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIC=?pattern\SarahC_640.jpg
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
ANA.New|1|$$XSZ|$$YSZ
ANA.New|2|$$XSZ|$$YSZ
$$COL=&H808080
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelColor|0|$$XPO|$$YPO|$$COL
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    ANA.Color_Diff_Max|$$LAS|$$COL|$$RES
    CAL.$$RES=$$RES*8
    ANA.SetPixelRGB|2|$$XPO|$$YPO|$$RES|$$GRN|$$BLU
    $$LAS=$$COL
  NEX.
NEX.
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelColor|0|$$XPO|$$YPO|$$COL
    ANA.GetPixelRGB|2|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$RED>$$BLU)+($$RED>$$GRN)+($$RED>&H80)+($$GRN<&H40)+($$BLU<&H40)
    IVV.$$ERG>3
      'ANA.SetPixelColor|1|$$XPO|$$YPO|&HFFFF00
      ANA.DrawBox|1|$$XPO|$$YPO|($$XPO+2)|($$YPO+2)|&HFF0000
    ELS.
      ANA.SetPixelColor|1|$$XPO|$$YPO|$$COL
    EIF.
  NEX.
NEX.
ANA.Save|1|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|1!
MBX.Used Time in Seconds: $$TIM
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

In EXE-Mode this Script needed 247 Seconds thats roughly 4 minutes.

```
013 0000000000000182 0001 VAN.$$TIM=#time#
014 0000000000000081 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000058 0001 VAF.$$PIC=?pattern\SarahC_640.jpg
016 0000000000448326 0001 ANA.Load|0|$$PIA
017 0000000000000418 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000020588 0001 ANA.New|1|$$XSZ|$$YSZ
019 0000000000019367 0001 ANA.New|2|$$XSZ|$$YSZ
020 0000000000000088 0001 VAR.$$COL=&H808080
021 0000000000000079 0963 FOR.$$YPO|1|$$YSZ
022 0000000000000003 0962
023 0000000000000084 616642 FOR.$$XPO|1|$$XSZ
024 0000000000000231 615680 ANA.GetPixelColor|0|$$XPO|$$YPO|$$COL
025 0000000000000355 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
026 0000000000000205 615680 ANA.Color_Diff_Max|$$LAS|$$COL|$$RES
027 0000000000000120 615680 CAL.$$RES=$$RES*8
028 0000000000000312 615680 ANA.SetPixelRGB|2|$$XPO|$$YPO|$$RES|$$GRN|$$BLU
029 0000000000000047 615680 VAR.$$LAS=$$COL
030 0000000000000004 616642 NEX.
031 0000000000000004 0963 NEX.
032 0000000000000081 0963 FOR.$$YPO|1|$$YSZ
033 0000000000000003 0962
034 0000000000000086 616642 FOR.$$XPO|1|$$XSZ
035 0000000000000229 615680 ANA.GetPixelColor|0|$$XPO|$$YPO|$$COL
036 0000000000000553 615680 ANA.GetPixelRGB|2|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
037 0000000000000563 615680 CAL.$$ERG=($$RED>$$BLU)+($$RED>$$GRN)+($$RED>&H80)+($$GRN<&H40)+($$BLU<&H40)
038 0000000000000055 615680 IVV.$$ERG>3
039 0000000000000003 74723
040 0000000000000522 74723 ANA.DrawBox|1|$$XPO|$$YPO|($$XPO+2)|($$YPO+2)|&HFF0000
041 0000000000000003 615680 ELS.
042 0000000000000233 540957 ANA.SetPixelColor|1|$$XPO|$$YPO|$$COL
043 0000000000000003 615680 EIF.
044 0000000000000004 616642 NEX.
045 0000000000000004 0963 NEX.
046 0000000000297477 0001 ANA.Save|1|$$PIC
047 0000000000000368 0001 CAL.$$TIM=#time#-$$TIM
048 0000000000000005 0001
049 0000000000000000 0001 DMP.1
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

The Speed-Dump of the Script shows that the ANA.Color_Diff_Max uses only about 200 Ticks execution Time.

Syntax

ANA.Color_Diff_RGB|P1|P2[|P3][|P4][|P5]

Parameter Explanation

P2 - Color A - Value, Color-Value 24-Bit. Example: &HFF00FF
P2 - Color B - Value, Color-Value 24-Bit. Example: &HFF00FF
P3 - Variable for Color-Value for ***Red, for resulting Color Value (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB). If missing the value is been placed on TOS.***
P4 - Variable for Color-Value for ***Green, for resulting Color Value (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB). If missing the value is been placed on TOS.***
P5 - Variable for Color-Value for ***Blue, for resulting Color Value (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB). If missing the value is been placed on TOS.***

ANA.Color_Diff_Max|P1|P2[|P3]
ANA.Color_Diff_Min|P1|P2[|P3]
ANA.Color_Diff_R|P1|P2[|P3]
ANA.Color_Diff_G|P1|P2[|P3]
ANA.Color_Diff_B|P1|P2[|P3]

Parameter Explanation

P2 - Color A - Value, Color-Value 24-Bit. Example: &HFF00FF
P2 - Color B - Value, Color-Value 24-Bit. Example: &HFF0000
P3 - Variable for result Color-Value ***(8-Bit, Format in Hexadezimal: 00-&HFF). If missing the value is been placed on TOS.***

Example

```
See above.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The `ANA.Color_Diff-Commands are` optimized for Speed. It will use only about 300 Ticks. Therefore the Robot could execute it more then 53000 times per Second.

Limitations:

-

See also:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.5.4 ANA.GetGradientColor

MiniRobotLanguage (MRL)

ANA.GetGradientColor

Calculate the Color of a SIngle Pixel in a Gradient between two Colors

Intention

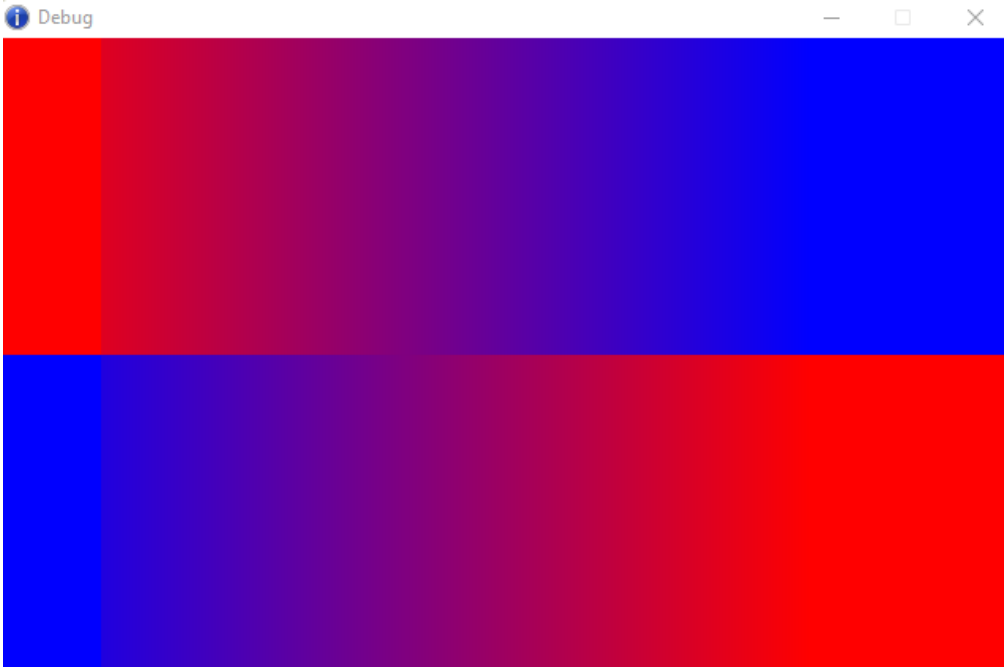
Use this command to calculate a Gradient Color between two colors.

```
$$COA=&HFF0000
$$COB=&H0000FF

VAN.$$XPS=640
VAN.$$YPS=400
ANA.New|1|$$XPS|$$YPS

VAN.$$AXA=0
VAN.$$AYA=0
VAN.$$AXB=64
VAN.$$AYB=200

VAN.$$BXA=($$XPS-$$AXB)
VAN.$$BYA=0
VAN.$$BXB=639
VAN.$$BYB=200
GSB.DrawGradient

$$COB=&HFF0000
$$COA=&H0000FF
VAN.$$AYA=200
VAN.$$AYB=400
VAN.$$BYA=200
VAN.$$BYB=400

GSB.DrawGradient

ANA.show|1!

ENR.
'=========================================================
:DrawGradient
ANA.DrawBox|1|$$AXA|$$AYA|$$AXB|$$AYB|$$COA|1
ANA.DrawBox|1|$$BXA|$$BYA|$$BXB|$$BYB|$$COB|1

CAL.$$DIF=$$BXA-$$AXB

FOR.$$XPO|$$AXB|$$BXA
  CAL.$$XPA=$$XPO-$$AXB+1
  ANA.GetGradientColor|$$COA|$$COB|$$DIF|$$XPO|$$RES
  ANA.DrawLine|1|$$XPO|$$AYA|$$XPO|$$BYB|$$RES
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
```

Output:



Syntax

ANA.GetGradientColor|P1|P2|P3[|P4]

Parameter Explanation

P1 - Color Value A (24-Bit, Format in Hexadezimal: &HFFFFFF=RRGGBB)
P2 - Color Value B (24-Bit, Format in Hexadezimal: &HFFFFFF=RRGGBB)
P3 - Number of Steps for the Gradient. 1 ... P3.
P4 - Number of the Step for which the Gradient shall be calculated.
P5 - Variable for the result. If omitted the result is placed on TOS.

Remarks

The ANA.Sub/Add/Mix Commands are optimized for Speed. They will only use about 210 Ticks. Therefore the Robot could work on more then 76000 Values per Second.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- Paint-Operations 1486

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.5.5 ANA.Split_RGB / Combine_RGB

MiniRobotLanguage (MRL)

ANA.Split_RGB

Split the a specified Color-Value into R,G and B

ANA.Combine_RGB

Combine the R,G and B Values into a Color-Value

Intention

This command will split an 24-bit Color-Value into the three 8-Bit Color-Components for R,G and B.
While that does not Sound much, it enables you to do unlimited complex Image processing,
Together with other Commands like `ANA.Combine_Color_RGB`.

1. Color-Processing

In this Sample we will additionally change the Colors.For this we use the two Commands:

```
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
  ' This is just for example normally you would use ANA.GetPixelColor_RGB which co
        ANA.Split_RGB|$$PIX|$$COR|$$COG|$$COB
  ' We Combine the Colors in away that Blue and Green is exchanged.
        ANA.Combine_RGB|$$COG|$$COR|$$COB|$$PIX
  ' And Set the Pixel
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
```

And then Paint the Picture.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

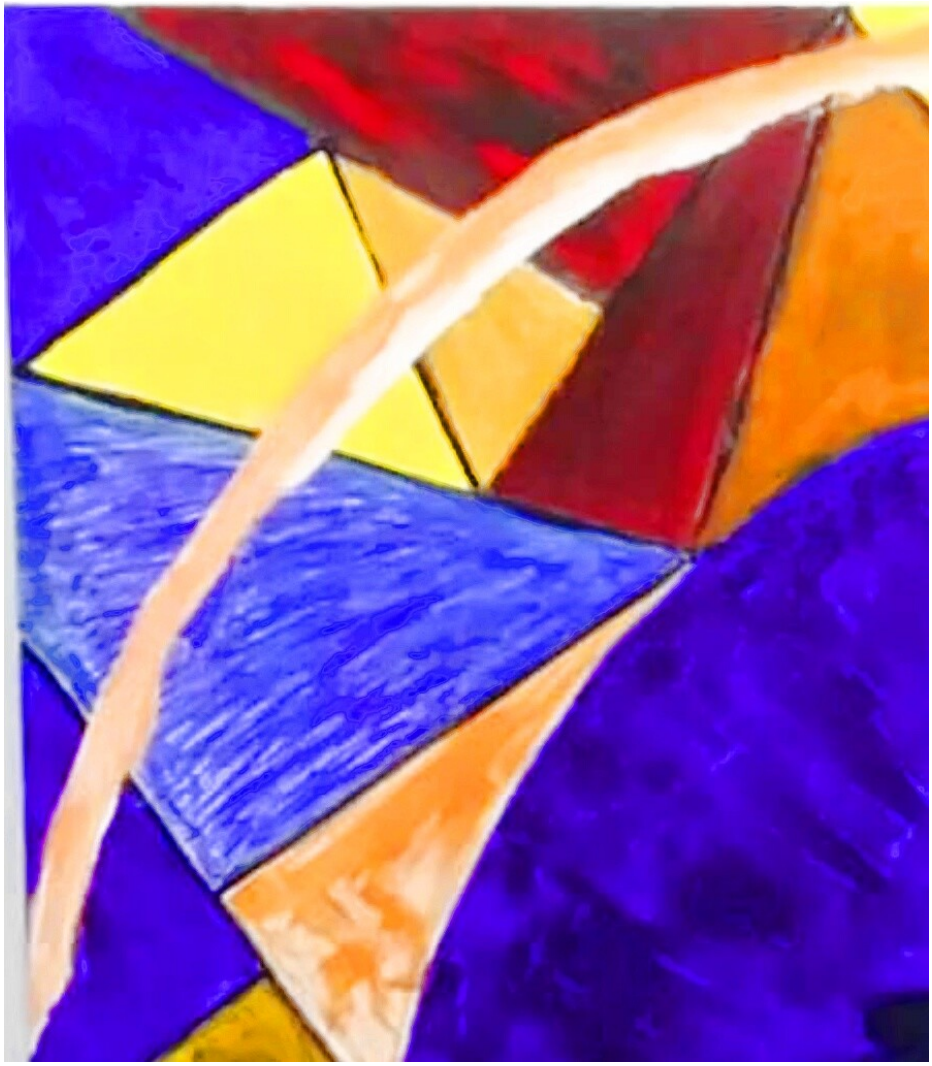
```
  ' Here is the Code.
  ' On my Computer it uses 5 Minutes Processing time for a Picture of Medium size
  VAN.$$TIM=#time#
  VAF.$$PIA=?pattern\Sarah_640.jpg
  VAF.$$PIB=?pattern\Mahakala2.png
  ANA.Load|0|$$PIA
  ANA.Load|1|$$PIB
  ANA.GetRes|0|$$XSZ|$$YSZ
  FOR.$$YPO|1|$$YSZ
    DBP.$$YPO/$$YSZ
    FOR.$$XPO|1|$$XSZ/2
      ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
      CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
      IVV.$$ERG>2
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
        ANA.Split_RGB|$$PIX|$$COR|$$COG|$$COB
        ANA.Combine_RGB|$$COG|$$COR|$$COB|$$PIX
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

The Script used ~300 Seconds to complete process the picture.

Here is the Speed-Dump:

```
012 0000000000000274 0001 VAN.$$TIM=#time#
013 0000000000000100 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000058 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000392529 0001 ANA.Load|0|$$PIA
016 0000000001986990 0001 ANA.Load|1|$$PIB
017 0000000000000568 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000126 0963 FOR.$$YPO|1|$$YSZ
019 0000000000060423 0962 DBP.$$YPO/$$YSZ
020 0000000000000146 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000504 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000493 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000082 615680 IVV.$$ERG>2
024 0000000000000383 422988 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000444 422988 ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
026 0000000000000370 422988 ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
027 0000000000000336 422988 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
028 0000000000000004 615680 EIF.
029 0000000000000006 616642 NEX.
030 0000000000000006 0963 NEX.
031 0000000000000310 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000230528 0001 DBP.Used Time in Seconds: $$TIM
033 0000000000000000 0001 DMP.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.Split_Color_RGB[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 - Color Value (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB)
P2 - Variable for Color-Value for ***Red***
P3 - Variable for Color-Value for ***Green***
P4 - Variable for Color-Value for ***Blue***

ANA.Combine_Color_RGB[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 - Variable for Color-Value for ***Red***
P2 - Variable for Color-Value for ***Green***
P3 - Variable for Color-Value for ***Blue***
P4 - opt. Variable for resulting Color Value (24-Bit, Format in Hexadezimal: &HFFFFFF=RR GG BB). If missing the value is been placed on TOS.

Example

```
See above.
```

Remarks

The `ANA.Combine_Color_RGB` is optimized for Speed. It will use only about 444 Ticks. Therefore the Robot could split more then 36000 Values per Second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6 Convert Colors

These Operations can be used to reduce color numbers in an IR.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6.1 ANA.Brightness

MiniRobotLanguage (MRL)

ANA.Brightness

Change Brightness in an Image-Register (IR)

Intention

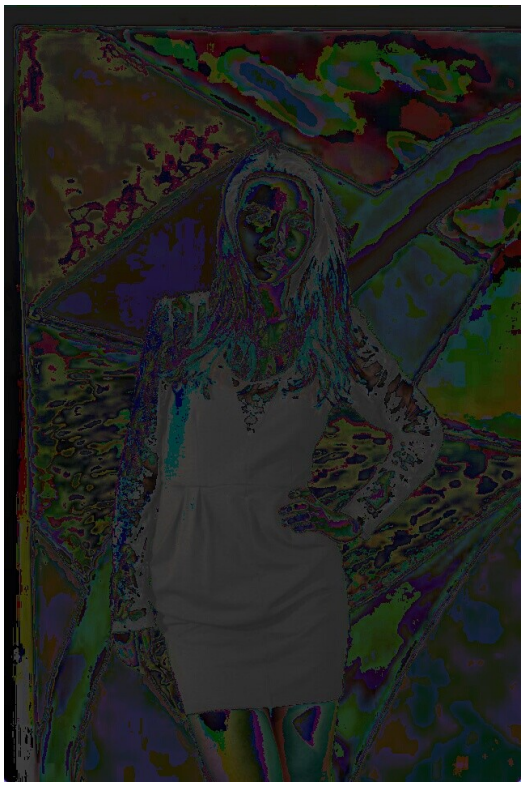
This command will change the Brightness value of an IR. The given **P2** value can be positive or negative.
Negative values for **P2**, will decrease the contrast. If **P3** is given, then the result will be stored in IR **P3**.

1. Change color Brightness in IR
In this case we do not specify **P3** therefore we will directly change the Original IR that contains the picture.

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.cow|0|64|1
ANA.Show|1!
ANA.Brightness|1|100
GSB.Save|1
ANA.Brightness|1|-50
ANA.Show|1!
ANA.Brightness|1|100
DMP.1
ANA.Show|1!
ENR.
```

Here is another Sample Picture.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

Original Picture **ANA.Complement**

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
014 0000000000000000 0000
015 0000000000000179 0001 VAF.$$PIA=?exeloc\Sara2.jpg
016 0000000000001877 0001 NEF.$$PIA
017 0000000000000000 0000 MBX.No such file.
018 0000000000000000 0000 ENR.
019 0000000000000028 0001 EIF.
020 0000000000364840 0001 ANA.Load|0|$$PIA
021 0000000028718138 0001 ANA.cow|0|64|1
022 0000000025375990 0001 ANA.Show|1!
023 0000000011331580 0001 ANA.Brightness|1|100
024 0000000000000474 0001 GSB.Save|1
025 0000000011541062 0001 ANA.Brightness|1|-50
026 0000000344593109 0001 ANA.Show|1!
027 0000000011393448 0001 ANA.Brightness|1|100
028 0000000000000000 0001 DMP.1
```

Brightness Calculation are fast and will be done mostly in less then 1 Second.

Syntax

ANA.Brightness|P1|P2[|P3]

Parameter Explanation

P1 - ***Source*** Image-Register Number, Source-IR, if omitted, IR 0 will be used.
P2 - ***(number, -255 to +255) Brightness Value,*** positive values will increase the contrast, negative values will decrease the contrast.
P3 - ***Destination*** Image-Register Number, if omitted, IR **P1** will be used as destination IR.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Brightness-Sample
'**********************************
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.Brightness|0|-100
ANA.Show|0!
ANA.Brightness|0|180
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ANA.Contrast 1352
- ANA.Luminance / Change_Luminance 1366

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6.2 ANA.BW

MiniRobotLanguage (MRL)

ANA . BW

Convert an Image-Register (IR) to Black and White using an specified threshold

Intention

This command will convert the Content of an IR directly into an B/W Image using only 2 Colours: Black and White.
The difference between Black and White is made along a user specified threshold that can vary from 1 - all White - to 768 (all Black).
Using a value of 0 will internally use the optimum value of 383.

1. Capture Screenshot of open Browser and convert it to BW
You can do this with your own preferred window - just change the STW.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
' If you use the value of "0", it will internally be used 383
ANA.bw|0|383
ANA.Save|0|?path\After.png
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

*© 2023 Theo Gottwald * https://www.smart-package.com*

Here you can see the picture before and after the operation, using the Threshhold of 383 and a threshold of 128.

2. Use BW with no Parameters
In this case the default values are taken, means "0" as the Source and Target IR, and &H17f (383 dec.) as threshold.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.bw
ANA.Save|0|?path\After.png
ENR.
```

3. Making things darker
Here we use a threshold of 460.

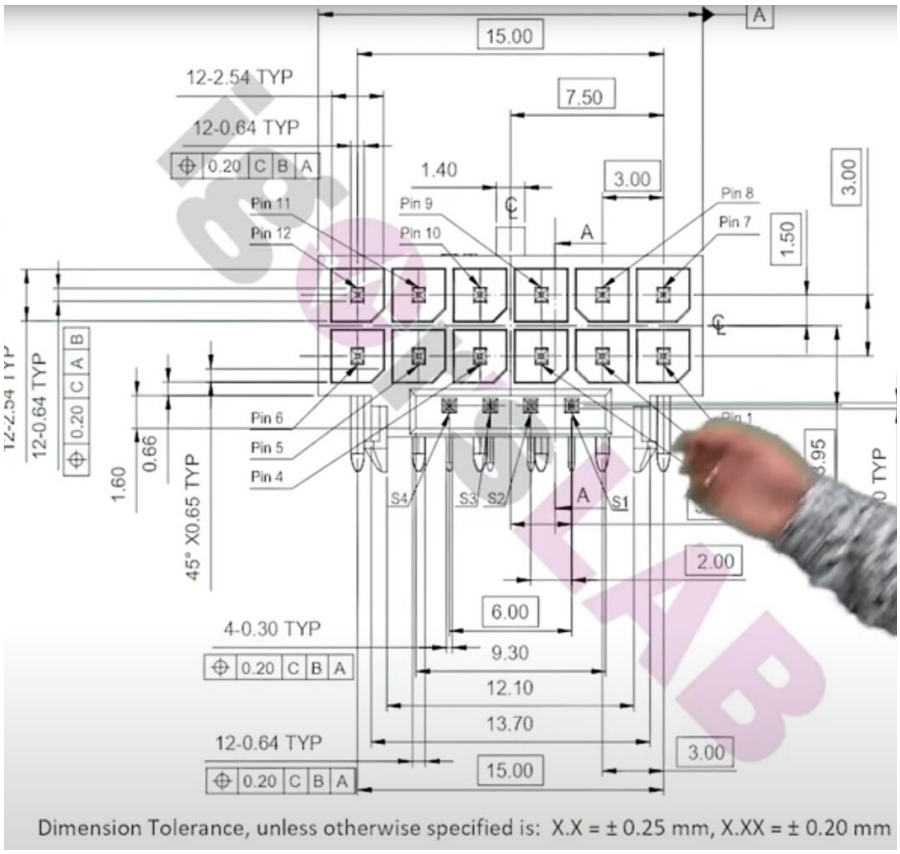
```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.bw|0|460
ANA.Save|0|?path\After.png
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

4. Quickly remove a watermark from a b/w picture.

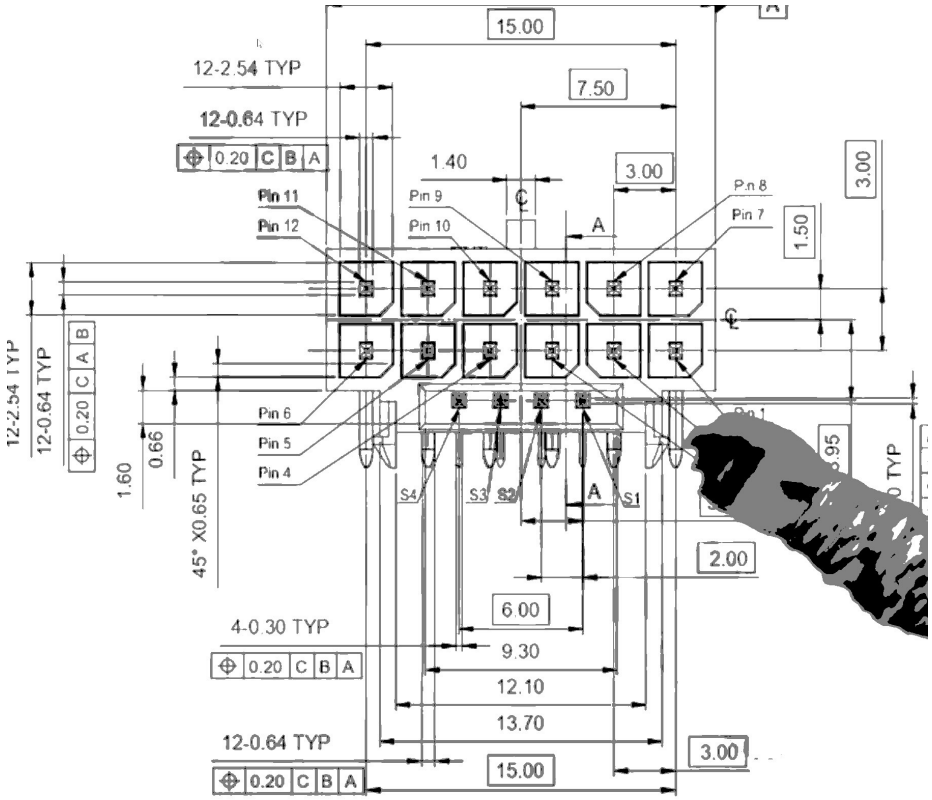
```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.TriColor|0|&H90FFFF
ANA.bw|0|760
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



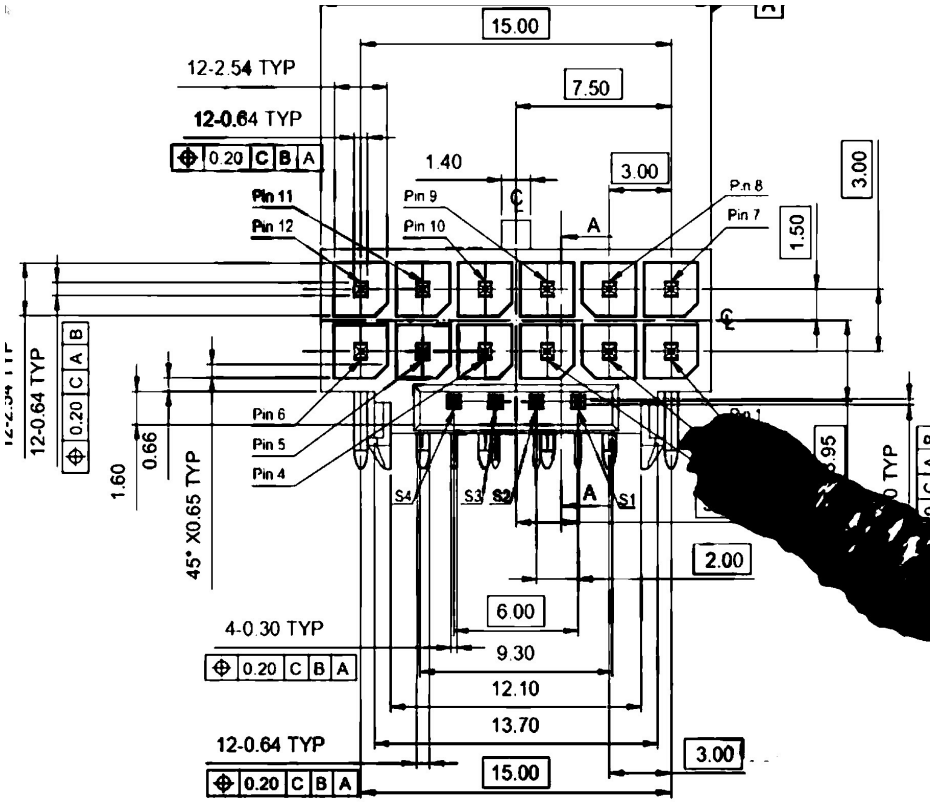
->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

Syntax

ANA.BW[|P1][|P2]

Parameter Explanation

P1 - opt. Image-Register number, if omitted, IR 0 will be used.

P2 - opt. Threshold-Level, the higher, the more black is in the picture, 765 - 1, 382.5 = Medium/default..

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'************************************
' ANA.BW-Sample
```

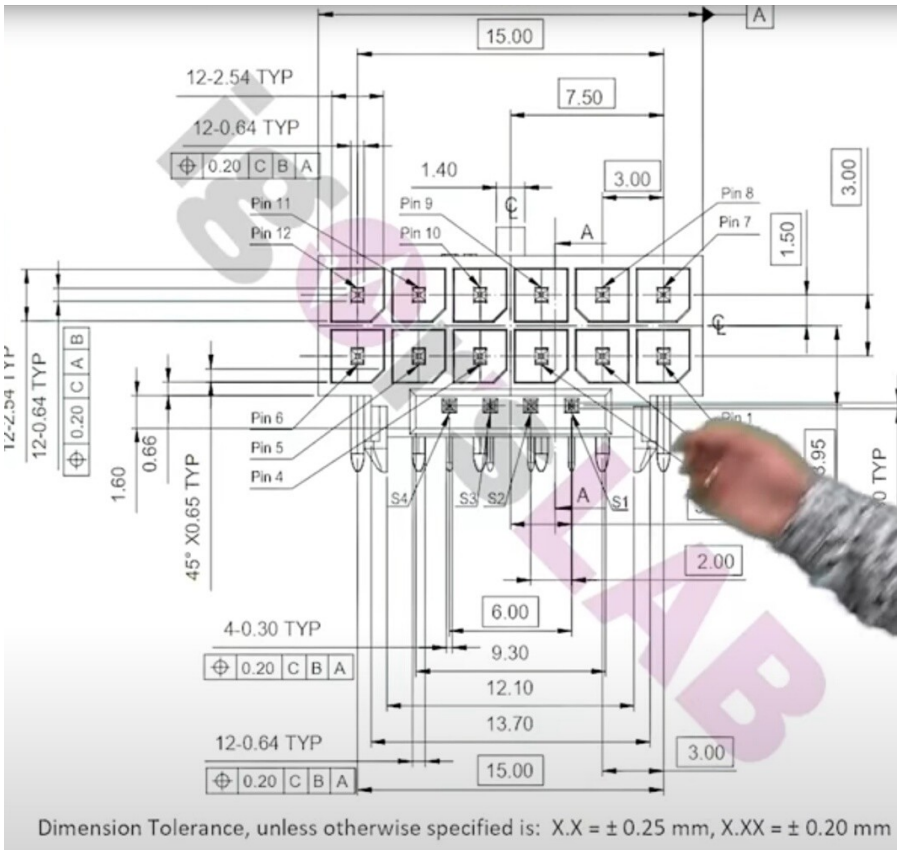

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.bw|0|60
ANA.Save|0|?path\After.png
ENR.
```

Quickly remove a watermark from a b/w picture.

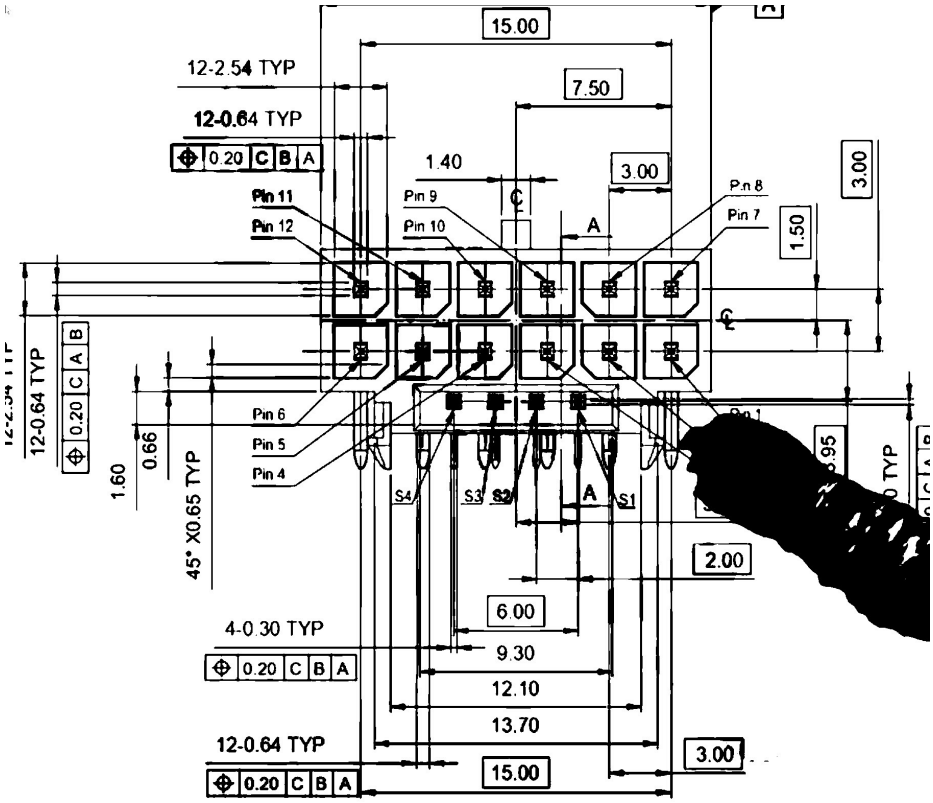
```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.TriColor|0|&H90FFFF
ANA.bw|0|760
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

Remarks

This command is not optimized, it may take up to several seconds for large pictures. For a Window with size `1140x1410` i needed `36867812 Ticks` (Speedtable using DMP.1), this is `2.3 Seconds` on my computer, as my system can process 16000000 Ticks in one second.

Limitations:

-

See also:

•

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6.3 ANA.Contrast

MiniRobotLanguage (MRL)

ANA.Contrast

Change Contrast in an Image-Register (IR)

Intention

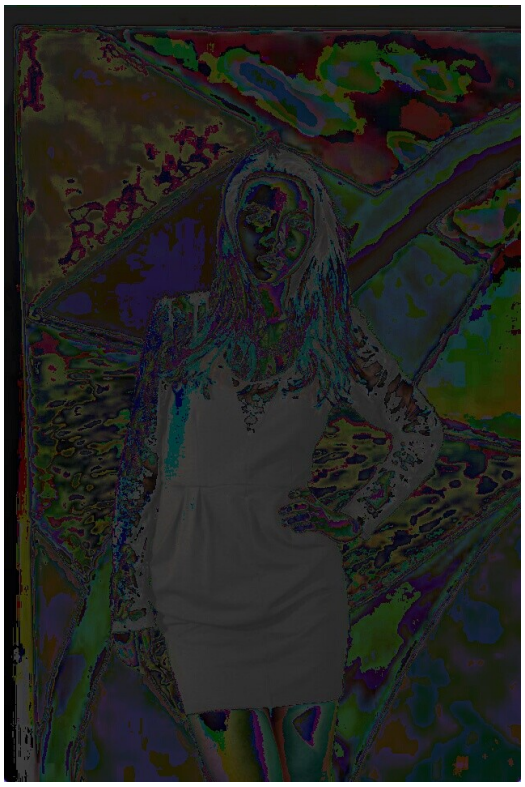
This command will change the contrast value of an IR. The given **P2** value can be positive or negative.
Negative values for **P2**, will decrease the contrast. If **P3** is given, then the result will be stored in IR **P3**.

1. Change color Contrast in IR
In this case we do not specify **P3** therefore we will directly change the Original IR that contains the picture.

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.cow|0|64|1
ANA.Show|1!
ANA.Contrast|1|2000
GSB.Save|1
ANA.Contrast|1|1000
ANA.Show|1!
ANA.Contrast|1|-500
ANA.Show|1!
ENR.
```

Here is another Sample Picture.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

Original Picture **ANA.Complement**

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
014 0000000000000000 0000
015 0000000000000175 0001 VAF.$$PIA=?exeloc\Sara2.jpg
016 0000000000001751 0001 NEF.$$PIA
017 0000000000000000 0000 MBX.No such file.
018 0000000000000000 0000 ENR.
019 0000000000000028 0001 EIF.
020 0000000000373686 0001 ANA.Load|0|$$PIA
021 0000000011523566 0001 ANA.cow|0|64|1
022 0000000031256706 0001 ANA.Show|1!
023 0000000028483133 0001 ANA.Contrast|1|2000
024 0000000000000354 0001 GSB.Save|1
025 0000000028874972 0001 ANA.Contrast|1|1000
026 0000000025937886 0001 ANA.Show|1!
027 0000000011499568 0001 ANA.Contrast|1|-500
028 0000000000000000 0001 DMP.1
029 0000000000000000 0000 ANA.Show|1!
030 0000000000000000 0000 ENR.
```

Contrast Calculation are fast and will be done mostly in less then 1 Second.

Syntax

ANA.Contrast|P1|P2[|P3]

Parameter Explanation

P1 - ***Source*** Image-Register Number, Source-IR, if omitted, IR 0 will be used.
P2 - ***(number, -16000 to +16000) Contrast Value, positive values will increase the contrast, negative values will decrease the contrast.***
P3 - ***Destination*** Image-Register Number, if omitted, IR **P1** will be used as destination IR.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.Contrast-Sample
'***********************************
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.Contrast|0|-500
ANA.Show|0!
ANA.Contrast|0|2000
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ANA.Brightness 1336

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6.4 ANA.Grey

MiniRobotLanguage (MRL)

ANA.Grey

Convert an Image-Register (IR) to Grey using a threshold

Intention

This command will convert the Content of an IR directly into an Grey Image.
Changing the Second parameter will give you a lot of options.
Using a value of 0 for **P2** will internally use the optimum value of 1.
This command can be used to prepare an IR for Patternsearch or other operations in special cases where we do not want to depend on colors.

1. Capture Screenshot of open Browser and convert it to Grey
You can do this with your own preferred window - just change the STW.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.Grey|0|0
ANA.Save|0|?path\After.png
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Here you can see the picture before and after the operation.

2. Use Grey with higher Threshold

In this case the effect is hardly foreseable. Due to the internal logic you will get an artificial result.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png

' Left Picture
ANA.Grey|0|2

' Second Picture
ANA.Grey|0|10
```

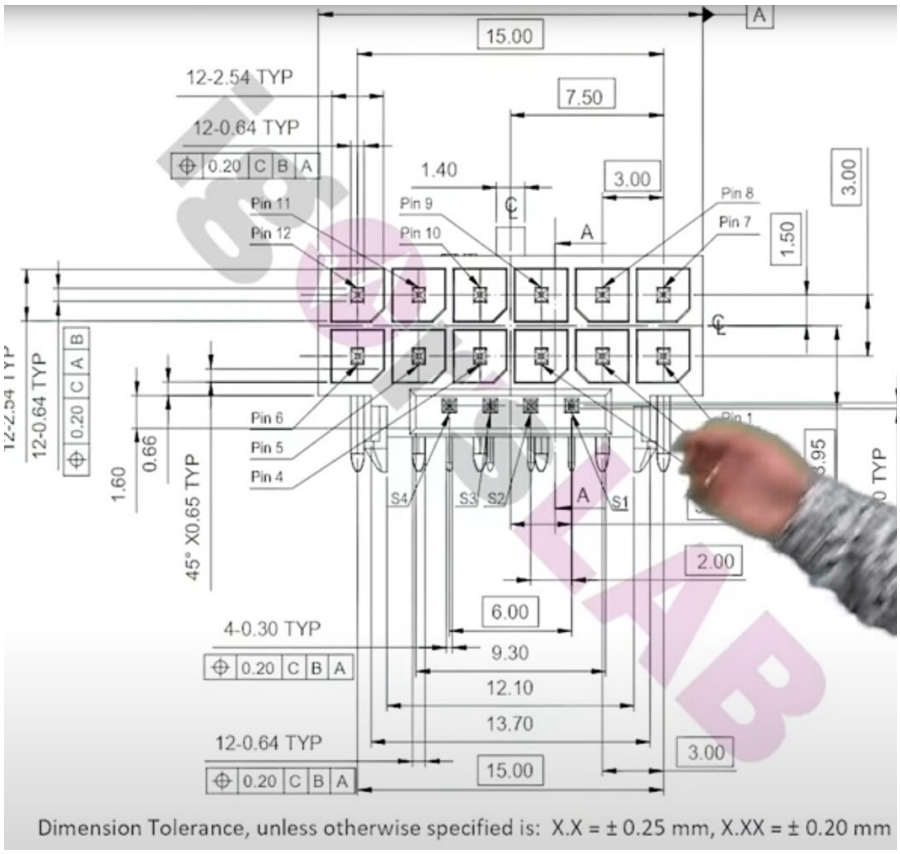

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Third Picture
ANA.Grey|0|400

' Right Picture
ANA.Grey|0|1650

ANA.Save|0|?path\After.png
ENR.
```

3. Quickly remove a watermark from a b/w picture.

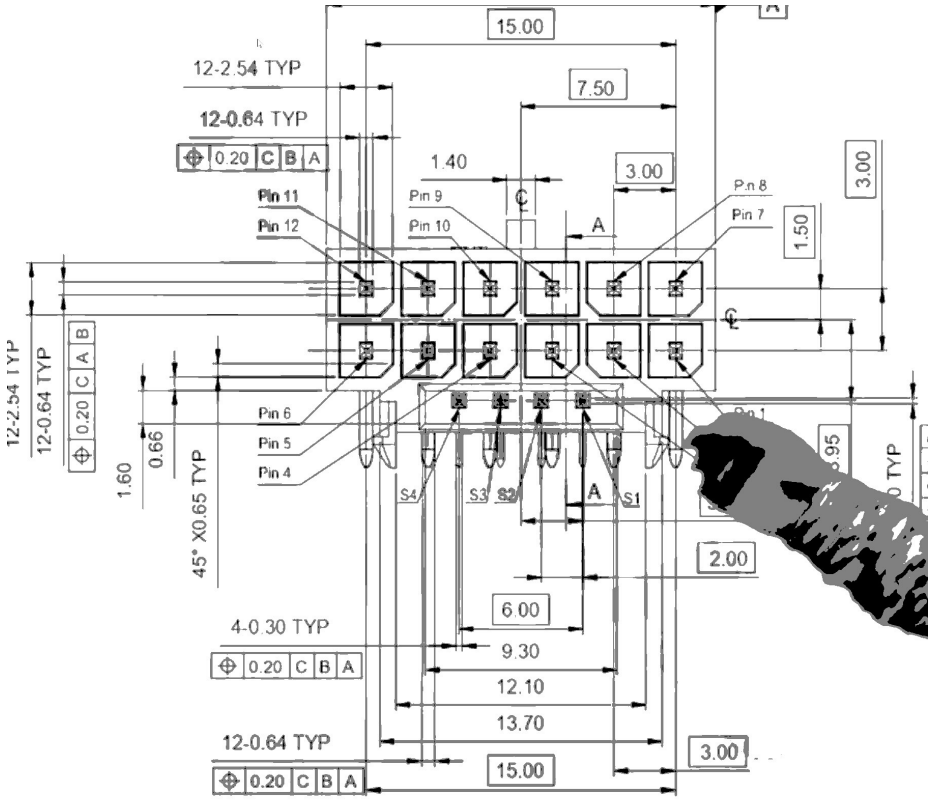
```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.TriColor|0|&H90FFFF
ANA.bw|0|760
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



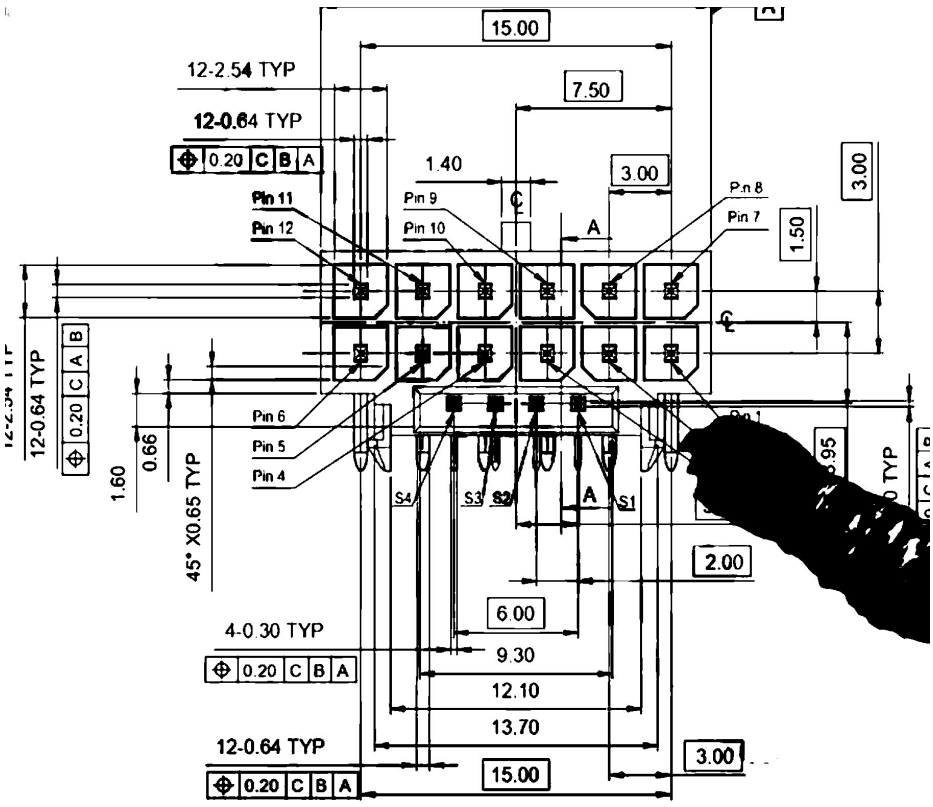
->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

Syntax

ANA.Grey[|P1][|P2]

Parameter Explanation

P1 **-** ***opt.*** Image-Register Number, if omitted, IR 0 will be used.
P2 **-** ***opt.*** ***Threshold-Number***, if omitted,1 will be used. The higher, the finer structures will be created.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' ANA.Grey-Sample
'*********************************
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.Grey|0|1
ANA.Save|0|?path\After.png
ENR.
```

Remarks

This command is optimized, it will process even large pictures in less then a second. For a Window with size 1140x1410 i needed 336402 Ticks (Speedtable using DMP.1), this is 0.02 Seconds on my computer, as my system can process 16000000 Ticks in one second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6.5 ANA.Luminance / Change_Luminance

ANA.Brightness - Change Brightness in IR Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.Luminance

Change HSL-Luminance in an full Image-Register (IR)

ANA.Change_Luminance

Calculate new HSL-Luminance for a given Color-Value

Intention

This command will change the Luminance-Value (HLS-System) for a full Image Register.
Luminosity or Luminance is somehow the brightness or brilliance of a picture.

There are 2 Modes:
A) Set Luminance to a fixed value between 0 and 1000
B) Change Luminance by a given value between -1000 and 1000
Which mode is used, depends on P3, if **P3** is 0 (absolute value to be set) or 1 (add/remove Luminance).
This command will work on the full given IR.

To just calculate the new Luminance from a single Pixel, use the equivalent command
`ANA.Change_Luminance|RGB-Color|Color-Value|0/1`

1. Change Luminance in IR relativ to current value
In this case we do set **P3** to 1. Therefore the command will change the Original luminance Value by adding or subtracting the given value **P2**..

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.Lum|0|500|1
DMP.1
ANA.Show|0!
ENR.
```

Here is another Sample Picture.

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Original Picture **ANA.Complement**

```
013 0000000000000333 0001 VAF.$$PIA=?exeloc\Sara2.jpg
014 0000000000002123 0001 NEF.$$PIA
015 0000000000000000 0000 MBX.No such file.
016 0000000000000000 0000 ENR.
017 0000000000000080 0001 EIF.
018 0000000000376635 0001 ANA.Load|0|$$PIA
019 0000000051434788 0001 ANA.Show|0!
020 0000000000447156 0001 ANA.Lum|0|500|1
021 0000000000000000 0001 DMP.1
022 0000000000000000 0000 ANA.Show|0!
023 0000000000000000 0000 ENR.
```

Luminance Calculations are fast and will be done mostly in less then 1 Second.

Here we are using the absolute Luiminance and therefore P3 is 0.

```
' Here we use P3
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.Lum|0|250|0
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Sample for
' ANA.Change_Luminance
'
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.GetRes|0|$$XRE|$$YRE|$$SIZ
FOR.$$LOX|0|($$XRE/3*2)|1.6
  PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|($$YRE/3*2)|1.6
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB
    ANA.clu|$$PIB|650|0|$$PIC
    ANA.Setpixelcolor|0|$$LOX|$$LOY|$$PIC
  NEX.
NEX.
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.Luminance|P1|P2[|P3][|P4]
ANA.Lum|P1|P2[|P3][|P4]

Parameter Explanation

P1 - ***Source*** Image-Register Number, Source-IR, if omitted, IR 0 will be used.

P2 - ***(number, -1000 to +1000) Luminance Value,*** positive values will increase the contrast, negative values will decrease the contrast.

P3 - ***0/1 Value.*** Setting **P3** to 0 will limit **P2** to 0-1000. In this case **P2** will be added to the value of the Pixels in the IR. Setting **P3** to 1 will set the limit P2 to 0-1000.

P4 - ***Destination*** Image-Register Number, if omitted, IR **P1** will be used as destination IR.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.Change_Luminance|P1|P2[|P3][|P4]
ANA.Clu|P1|P2[|P3][|P4]

Parameter Explanation

P1 - ***RGB-In Value, Example: &HF0AC45***
P2 - ***(number, -255 to +255) Brightness Value,*** positive values will increase the contrast, negative values will decrease the contrast.
P3 - ***0/1 Value.*** Setting **P3** to 0 will limit **P2** to 0-1000. In this case **P2** will be added to the value of the Pixels in the IR. Setting **P3** to 1 will set the limit P2 to 0-1000.
P4 - ***Destination Variable,*** if omitted TOS is used.

Example

```
'**********************************
' ANA.Luminance-Sample
'**********************************
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.GetRes|0|$$XRE|$$YRE|$$SIZ
FOR.$$LOX|0|($$XRE/3*2)
  PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|($$YRE/3*2)|3
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB
    ANA.clu|$$PIB|650|1|$$PIC
    ANA.clu|$$PIB|-650|1|$$PID
    ANA.Setpixelcolor|0|$$LOX|$$LOY|$$PID
    ANA.Setpixelcolor|0|$$LOX|($$LOY+1)|$$PIC
  NEX.
NEX.
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ANA.Contrast 1352
- ANA.Brightness 1336

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.6.6 ANA.TriColor

MiniRobotLanguage (MRL)

ANA.TriColor

Convert an Image-Register (IR) to Black and White using an specified threshold

Intention

This command will convert the Content of an IR directly into an B/W Image using only 2 Colours: Black and White.
The difference between Black and White is made along a user specified threshold that can vary from 1 - all White - to 768 (all Black).
Using a value of 0 will internally use the optimum value of 383.

1. Capture Screenshot of open Browser and convert it to Three Colors
Here i have used a Color-Threshold of "0" that will evaluate to the default value of `&H808080`
You can do this with your own preferred window - just change the STW.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.TriColor|0|0
ANA.Save|0|?path\After.png
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

*© 2023 Theo Gottwald * https://www.smart-package.com*

17 GESUNDHEITSMYTHEN IM CHECK
Brauche ich 5 Portionen Obst und Gemüse am Tag?
Acht Stunden Schlaf, 10 000 Schritte am Tag – was ist richtig und was überholt?
17 GESUNDHEITSMYTHEN IM CHECK
Brauche ich 5 Portionen Obst und Gemüse am Tag?
Acht Stunden Schlaf, 10 000 Schritte am Tag – was ist richtig und was überholt?
17 GESUNDHEITSMYTHEN IM CHECK
Brauche ich 5 Portionen Obst und Gemüse am Tag?
Acht Stunden Schlaf, 10 000 Schritte am Tag – was ist richtig und was überholt?

Here you can see the picture before and after the operation, using the Threshhold of &H808080 and a threshold of &HF0F0F0 the 4th picture has a Threshold of &HF010F0 and the last picture has a Threshold of &H1010F0.

2. Due to the used technology, there is no difference whether you use &H1010F0 or &HF01010

The Values are not sorted to RGB but play a role all together.

```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png

' This will result in a B/W picture (left picture below)
ANA.TriColor|0|&HFFFFFF

' This will result in a grey picture (right picture below) because "0" will use
ANA.TriColor|0|&H0

' This will result in the right picture below.
ANA.TriColor|0|&HFF40FF

ANA.Save|0|?path\After.png
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3. Making things lighter
Here we use a threshold of &H101010.

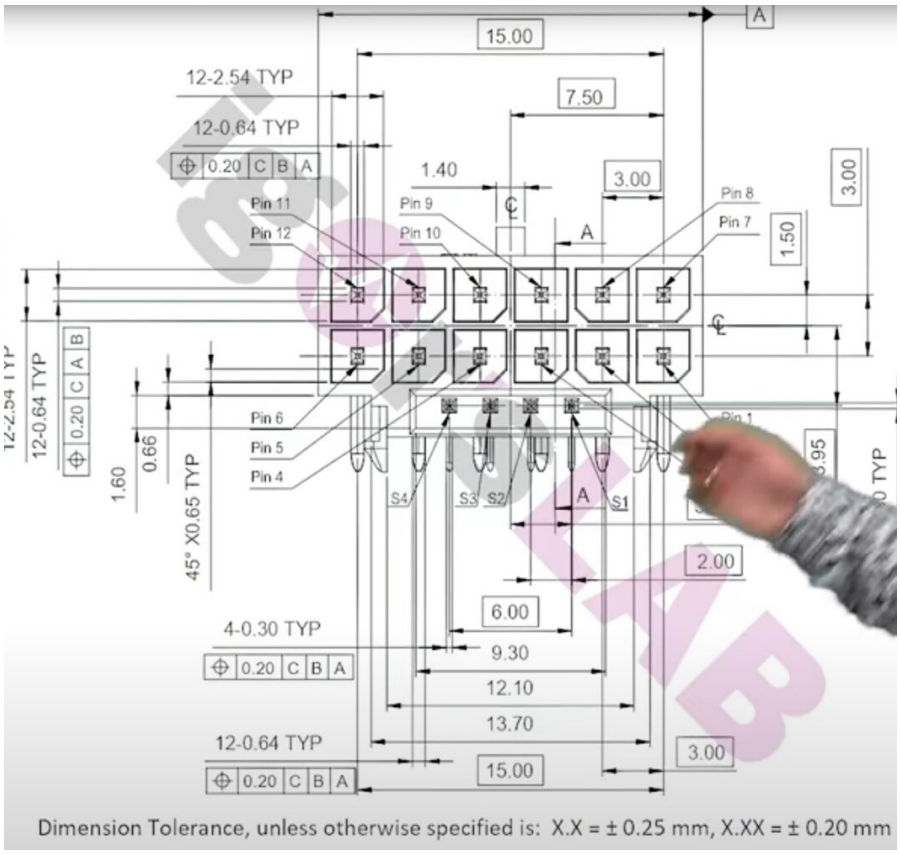
```
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
' This will result in a lighter picture (see picture below)
ANA.TriColor|0|&H010101
ANA.Save|0|?path\After.png
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Quickly remove a watermark from a b/w picture.

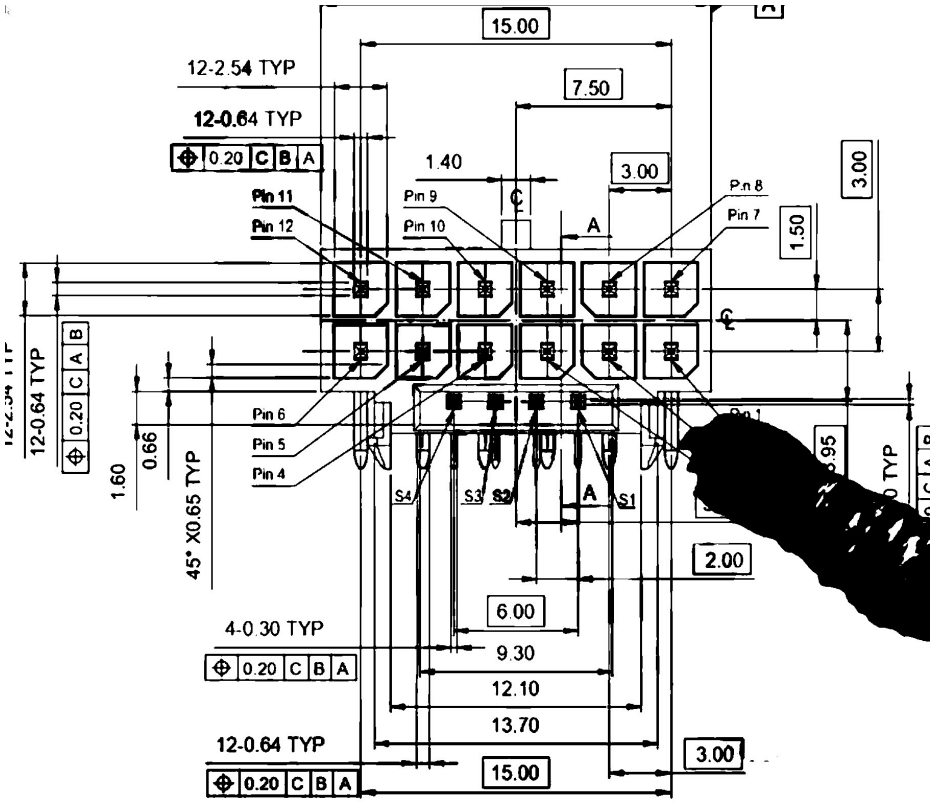
```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.TriColor|0|&H90FFFF
ANA.bw|0|760
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

Syntax

ANA.TriColor[|P1][|P2]

Parameter Explanation

P1 - opt. Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Threshold value for Grey. If omitted or zero, `&H808080` will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.TriColor-Sample
'***********************************
STW.ct|Chrome_WidgetWin_1|- Brave
ANA.Snap|0
ANA.Save|0|?path\Before.png
ANA.TriColor|0|0
ANA.Save|0|?path\After.png
' See the speedtable
DMP.1
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Remarks

This command is not optimized, it may take up to several seconds for large pictures. For a Window with size 1140x1410 i needed 40572261 Ticks (Speedtable using DMP.1), this is 2.5 Seconds on my computer, as my system can process 16000000 Ticks in one second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.7 Debug & Show IR's

These Commands are used to improve your Scripts.

Using Multi-Pattern Find Operations with scaled Patterns and high Tolerance Settings can lead to the same Object
being found from several engines. Is such cases you see this effects.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.7.1 ANA.Show

MiniRobotLanguage (MRL)

ANA . Show

Display Content of an Image-Register (IR)

Intention

This command will display the content of an Image-Register. It will stay open as long as the Script runs.
Optionally it will wait for a "Confirmation" to close the Display-Window.

You can currently display only 1 IR at a time.

1. Generate a Gradient and then display it using Show

```
' Here we have a Sample-Script
ANA.New|0|800|400
ANA.vgradient|0|&HFF0000|&H00FFFF
ANA.Show|0!
ANA.New|1|800|400
ANA.hgradient|1|&H00FFFF|&HFF0000
ANA.Show|1!
ANA.blendTo|0|1|2|160
ANA.Show|2!
ANA.VFlip|2
DMP.6
ANA.Show|2!
ENR.
```

If you add a "!" to the Show Command, this window will appear.
If you press "Yes" then the Script will continue running, and the Graphics-Window will be closed.
If you press "No", the Script will end.

2. Capture Window and scale/Stretch it using "New" then display it using show
`ANA.Show` will not do any scaling from itself. It will display the content of the IR 1:1.
If you want to do a scaling, you can use other commands like "New" or "EqualRes" first.

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS|Desktops-Manager
ANA.Snap|0
' Create New IR with 800x600 Resolution, then "stretch" IR 0 into this one
ANA.New|1|800|600|0
' Display the stretched picture
ANA.Show|1!
ENR.
```

Syntax

ANA.Show[|P1][!][@][#]

Parameter Explanation

P1 - opt. **Image-Register number, if omitted, IR 0 will be used.**

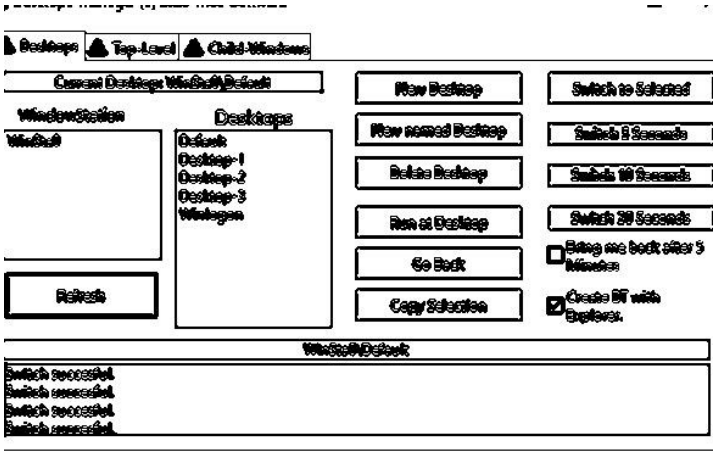
If the Snapshot worked fine, a "1" is returned on the TOS.
If Windows is unable to reserve the Memory for the Screenshot-Operation, the operation will fail and the command will return "0" on TOS.
If you add a "!" immediately after **P1**, then a Message-Box will appear and you can stop the Script and take closer Look.
If you add a "@" immediately after **P1**, then the result will be already filtered with a internal Fuzzy Logic. The effects may not be visible without a Lens.
If you add a "#" immediately after **P1**, then the Picture will be scaled to 512 Pixels using a internal logic that helps you preview the picture.

Example

```
'**********************************
' ANA.Show-Sample
'**********************************
STW.ct|Progman|Program Manager
ANA.snap|0
ANA.Copy|0|1
ANA.BlockTo|0|0|2
ANA.rct|0|&H0|260|&H0
ANA.rct|0|&HFFFFFF|200|&HFFFFFF
ANA.logic|0|1|0|A XOR B NOT T|RGB
ANA.rct|0|&H0|260|&H0
ANA.rct|0|&HFFFFFF|200|&HFFFFFF
ANA.show|0!
ENR.
```

Here you can see the result of the Operation.

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

Putting a Screenshot inside a Variable using "ToVar" will produce a Variable with this format insider the Variable:
- 8 Byte (4 Byte X- and 4 Byte Y-Size Information)
- (X-size * y-size)*4 Byte per Pixel-graphical Data in BGR-Order.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.7.2 ANA.ShowHits

MiniRobotLanguage (MRL)

ANA.ShowHits

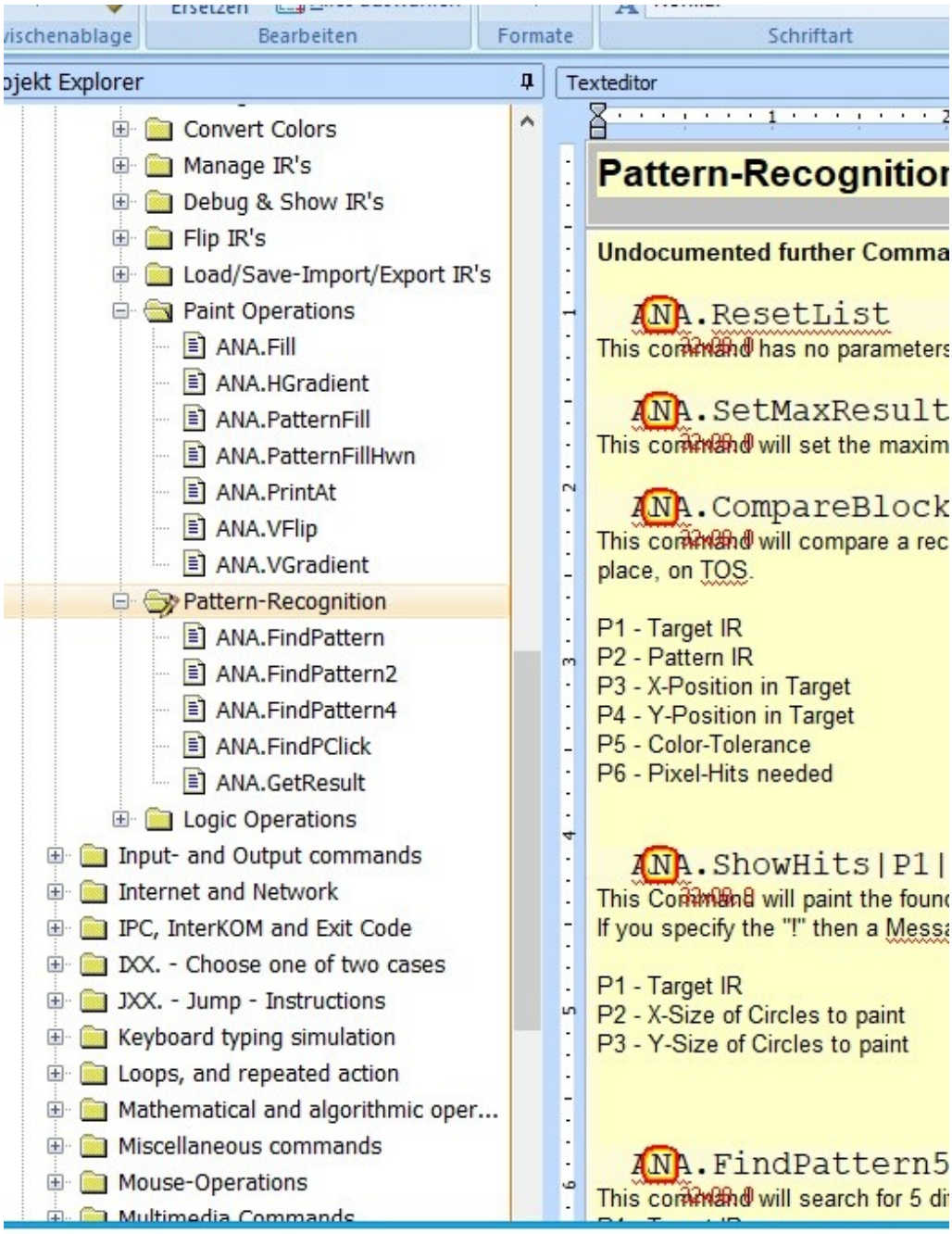
Display Result from Find-Operation

Intention

This Command will paint the found Hits in the given IR. Just like the "Show Hits" Button in the HE-Recorder.

```
'*********************************
' ANA.ShowHits-Sample
'*********************************
ANA.Load|0|?pattern\HelpM.png
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min. Hits (Pixel) maximu
ANA.Load|1|?pattern\Big_Ana.bmp
ANA.fpa|0|1|0|46|0
DMP.6
ANA.ShowHits|0!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



© 2023 Theo Gottwald * https://www.smart-package.com

Lower picture: This is the Stack after the call to ANA.fpa Command.
Upper Picture: This is the IR with the found Results painted.

Syntax

ANA.ShowHits[|P1][|P2][|P3][!][!]

Parameter Explanation

P1 - Target IR
P2 - X-Size of Circles to paint
P3 - Y-Size of Circles to paint

If you specify the "!" then a Messagebox will appear and prevent the Script from Continue running.
If you specify the "!!" then a Messagebox will appear and prevent the Script from Continue running also the picture will be prefiltered for noise like its done with the internal search

Example

```
'**********************************
' ANA.ShowHits-Sample
'**********************************
ANA.Load|0|?pattern\HelpM.png
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min. Hits (Pixel) maximu
ANA.Load|1|?pattern\Big_Ana.bmp
ANA.fpa|0|1|0|46|0
DMP.6
ANA.ShowHits|0!
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.8 Flip IR's

This command will flip the Content of IR 1 horizontally or vertically.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.8.1 ANA.HFlip

MiniRobotLanguage (MRL)

ANA.HFlip

Flip an Image-Register (IR) horizontally

Intention

This command will flip the Content of IR 1 horizontally..

1. Just Flip IR P1 horizontally
This simple Script shows the Use of `ANA.HFlip`.

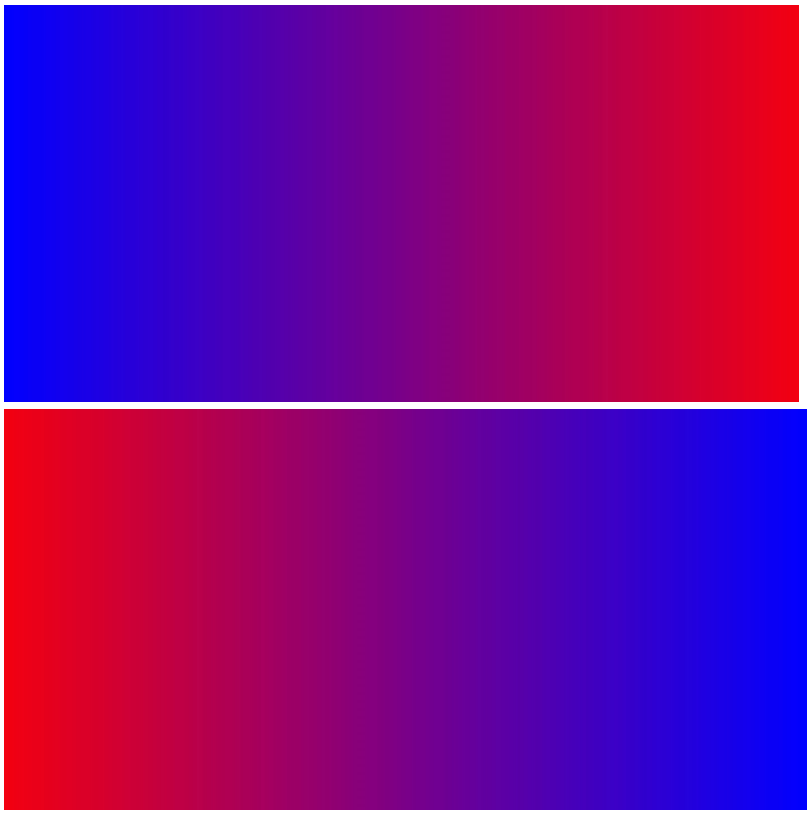
```
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4
ANA.Show|0!
' Make Resolution from P2 to the resolution from P1.
ANA.HFlip|0
DMP.6
ANA.Show|0!
ENR.
```

2. Flip P1 to P2
In this case, **P1** will be H-Flipped to **P2**.

```
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4
ANA.Show|0!
' Make Resolution from P2 to the resolution from P1.
ANA.HFlip|0|1
DMP.6
ANA.Show|1!
ENR.
```

Effect of HFlip:

*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

ANA.HFlip|P1][|P2]

Parameter Explanation

P1 – Image-Register number, this IR contains the Image that is to be H-Flipped.
P2 – ***opt.*** Destination Image-Register number, if omitted, P1 will be used as Destination.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.HFlip-Sample
'***********************************
ANA.New|0|800|400
ANA.hgradient|0|&HFF0000|&H1000F4
ANA.Show|0!
' Make Resolution from P2 to the resolution from P1.
ANA.HFlip|0
DMP.6
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.8.2 ANA.HVFlip

MiniRobotLanguage (MRL)

ANA.HVFlip

Flip an Image-Register (IR) vertically

Intention

This command will flip the Content of IR 1 vertically.

1. Just Flip IR P1 vertically and horizontally

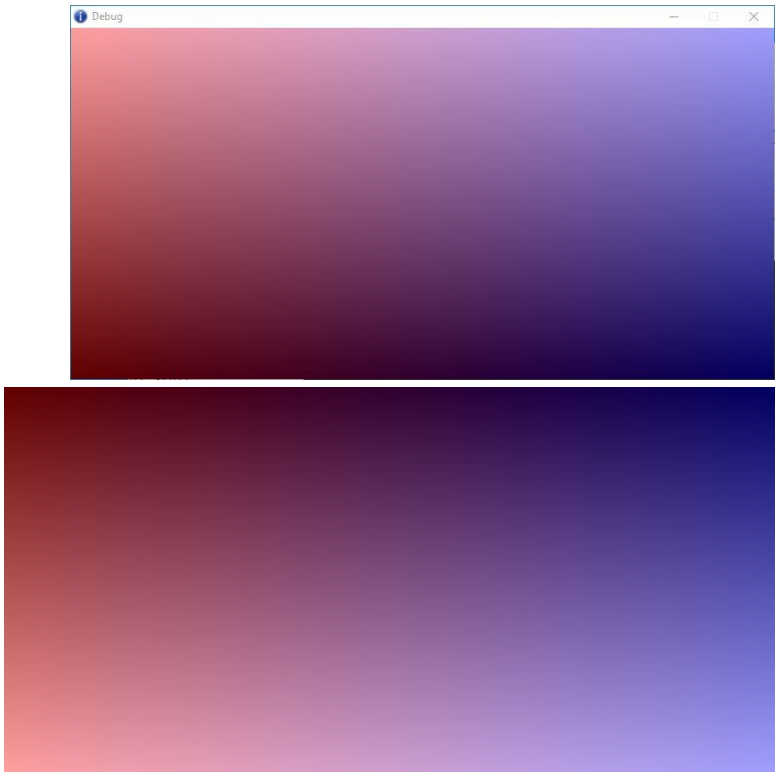
```
'This simple Script shows the Use of ANA.HVFlip.
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000FF|&HFF0000
ANA.blendTo|0|1|2|160
ANA.Show|2!
' Make Resolution from P2 to the resolution from P1.
ANA.HVFlip|2
DMP.6
ANA.Show|2!
ENR.
```

2. Flip P1 to P2
In this case, **P1** will be HV-Flipped to **P2**.

```
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000FF|&HFF0000
ANA.blendTo|0|1|2|160
ANA.Show|2!
ANA.HVFlip|2|1
DMP.6
ANA.Show|1!
ENR.
```

Effect of VFlip:

*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

ANA.HVFlip|P1][|P2]

Parameter Explanation

P1 – Image-Register number, this IR contains the Image that is to be H-Flipped.
P2 – ***opt.*** Destination Image-Register number, if omitted, P1 will be used as Destination.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.HVFlip-Sample
'***********************************
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000FF|&HFF0000
ANA.blendTo|0|1|2|160
ANA.Show|2!
' Make Resolution from P2 to the resolution from P1.
ANA.HVFlip|2
DMP.6
ANA.Show|2!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.8.3 ANA.VFlip

MiniRobotLanguage (MRL)

ANA.VFlip

Flip an Image-Register (IR) vertically

Intention

This command will flip the Content of IR 1 vertically.

1. Just Flip IR P1 vertically

```
'This simple Script shows the Use of ANA.VFlip.
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000FF|&HFF0000
ANA.blendTo|0|1|2|160
ANA.Show|2!
' Make Resolution from P2 to the resolution from P1.
ANA.VFlip|2
DMP.6
ANA.Show|2!
ENR.
```

2. Flip P1 to P2

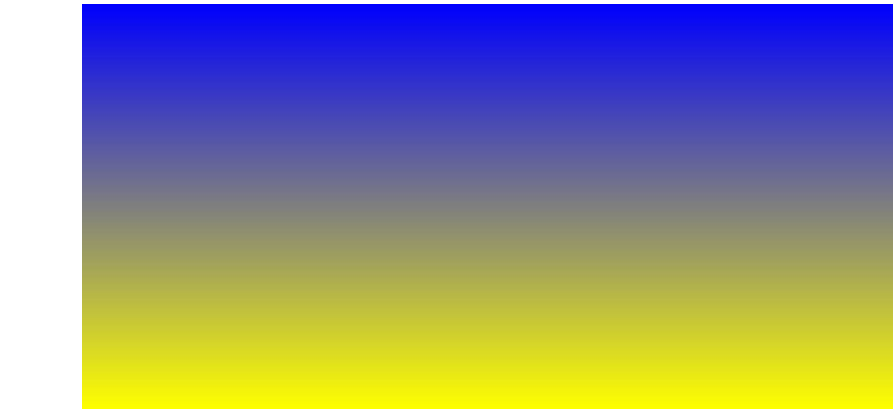
In this case, **P1** will be H-Flipped to **P2**.

```
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000FF|&HFF0000
ANA.blendTo|0|1|2|160
ANA.Show|2!
ANA.VFlip|2|1
DMP.6
ANA.Show|1!
ENR.


' One more Sample with Output pictures see below
ANA.New|0|800|400
ANA.vgradient|0|&HFF0000|&H00FFFF
ANA.Show|0!
ANA.New|1|800|400
ANA.hgradient|1|&H00FFFF|&HFF0000
ANA.Show|1!
ANA.blendTo|0|1|2|160
ANA.Show|2!
ANA.VFlip|2
```

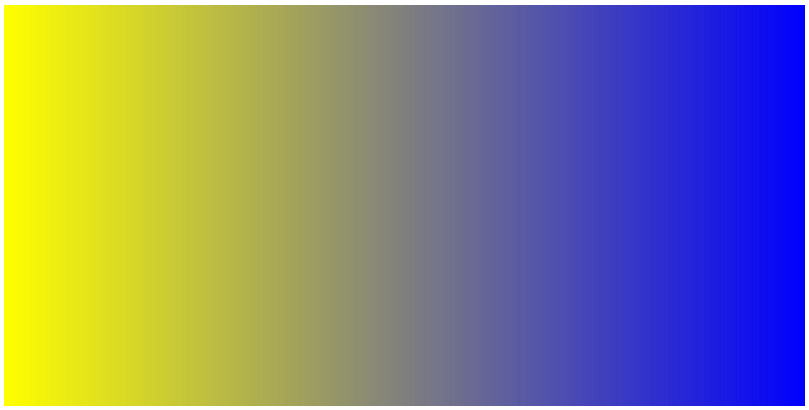

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
DMP.6
ANA.Show|2!
ENR.
```

Effect of VFlip:



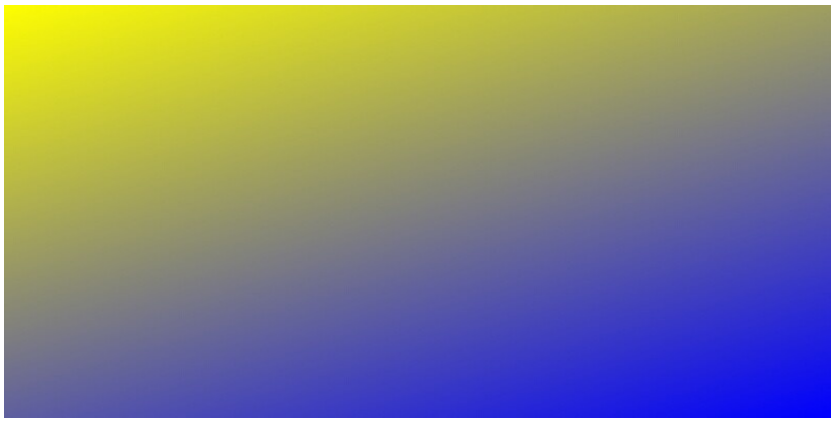
+



=

Now we do the VFLIP:

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

ANA.VFlip|P1][|P2]

Parameter Explanation

P1 – Image-Register number, this IR contains the Image that is to be H-Flipped.
P2 – ***opt.*** Destination Image-Register number, if omitted, P1 will be used as Destination.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'*********************************
' ANA.VFlip-Sample
'*********************************
ANA.New|0|800|400
ANA.vgradient|0|&H000000|&HFFFFFF
ANA.New|1|800|400
ANA.hgradient|1|&H0000FF|&HFF0000
ANA.blendTo|0|1|2|160
ANA.Show|2!
' Make Resolution from P2 to the resolution from P1.
ANA.VFlip|2
DMP.6
ANA.Show|2!
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

•

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9 Load/Save-Import/Export IR's

This command will Load and Save specified IR from a diskfile or variables.

This way you can use an nearly unlimited number of Images for your processing Operations.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.1 ANA.#FromVar

ANA.#FromVar - Restore IR-Content from Variable Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.#FromVar

Restore Image-Register (IR) Content from SPR-Variable

Intention

This command will restore the IR Content from a SPR Variable.
It is the counterpart to `ANA.ToVar`. Using "`#FromVar`" you can later transfer it back into any IR.

Alternative you can also use the new Datamaker.exe-Utility to store Images in Data-Lines directly in the Script.
Then call the Data and use the Variable to restore the IR.

The Data that ANA.#FromVar expects is in a special Format that you get using `ANA.ToVar`.

Putting a IR into a Variable using "ToVar" will produce a Variable with this format insider the Variable:
- 8 Byte (4 Byte X- and 4 Byte Y-Size Information)
- (X-size * y-size)*4 Byte per Pixel-graphical Data in BGR-Order.

Therefore in many cases when the Data is from external Files, you may need to convert the Image-Data first using
GRA.vToBMP 1797 - Command.

```
' Load some Data that was imported from Datamaker
GSB.Inline_3838
' Convert external Format to Internal BMP-Format
GRA.vbm|$$DTA|$$DTA|.bmp
' Send it to the Image-Register
ANA.#FromVar|2|$$DTA
ANA.show|2!
MBX.Data Loaded.
ENR.
```

1. Use ToVar with IR 0 and store Content on TOS
This will use IR 0 and store the Content on TOS.

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS|Desktops-Manager (c) 2020 Theo Gottwa
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This will use IR 0 and TOS because no parameter were given.
ANA.ToVar
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Here we get the TOS into the Variable
POP.$$CON
' We put the Content into IR 1
ANA.#FromVar|1|$$CON
' We display the IR 1
ANA.show|1!
ENR.
```

2. Use ToVar with Parameters

This will include all Screens on a Dual-Screen System. They will be saved as a single ".bmp" (BItmap-) File. You can choose any widely used picture format, for example ".png" or "Jpg".

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS|Desktops-Manager (c) 2020 Theo Gottwa
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This will use IR 0 and TOS because no parameter were given.
ANA.ToVar|0|$$CON
' We put the Content into IR 1
ANA.#FromVar|1|$$CON
' We display the IR 1
ANA.show|1!
ENR.
```

3. Use Datamaker and VDA. as Source for the IR

This will load an Images "Inline" from Datas that have been imported into the Script.
Use the Datamaker-Tool to generate the Data-Part.

Sample-Script using VDA. and VDE.

```
GSB.Laa
ANA.#FromVar|1|$$DTA
ANA.Show|1#!
ENR.
'============================================================
:Laa
'------------------------------------------------------------
' BITMAP from Clipboard(.bmp).
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  1392 Bytes in  22 Lines.
VDA.$$DTA|$$FIN=
?MAAGuBQAASlG0yTbYYlYMQvkOznPrYa3D1MGg62CLOJf5wNvQQ99VP4L6gQQ5AQXC
OpZoDz0mKBeeLz0riH9lGqZQA79lN4b4AtrGQBuz43T5ifxZuUGlkhKiUgJlmuYw
k3eBRZ9a0WwN+wjqxthPIlWS4rJSTa7stW/t3kQBBcK6Vgi/hQq+XAXCutbI35Vy
cnr2e4dGgPuje8jH5vNDpNH9JmQJeUncSSENg3+Fx97EhJVfEiLVUgKi6nkAOwA7
Q6tQi8CGFKHNY+9du0AbM6NUiUHgy+AzrJivUop+P6VXFqAW7M3beMK6Vgg/Rapu
dHEj5VTggYgu0pyzMxTHsdsVxDLgZZmrHywVUnJyvaZYhz4U7MHlb7IdttBLoBZs
DXvIFVaAekJ5xx+vZ5VydaSQNX0onmz9e0hVNoD7d3Y2RgQKWaHdmbz4CQArkK5g
+wgnFmufWo78ANv4v2o5oOvoz9XxAKuGe4DygnB5IQX3DmfZabFFJ9Hszdv43VVm
gFOfF99Jkd2W7D2ryF9VkuKy0g3O3E2jjw0D/LXUnIcBFGQxZfqCag4kGRm0PSsI
P8VqjuRJZwmBzcvjMop+P+VntliHAcTZ0f2mpGkpkXifWg48BZt2E9JshcfetNEY
b7LaxgNLoFM84EsgExxwA4RhtYqW4LPs8I/ortxFYspOnG2zB8lhLROSscWS3odl
J4Z4//o2WxeebI3+2EcepJSt09RBYAuA4/SHmVkoq1MvjZv209TBoOvgSxinkQw9
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
g8R1Zca9td60QSUPcvJ9Brjf+gHE2dF95kQZ8Q0+ePfmyJKWYDladdJBeP96djtn
JqhXvp91T5rmmNOaAdtG+LdGiP8lOhYorw3OXA2Df4VHHqQUbbNEmYAdm0Bzr7HY
134/5ef2uFdZECOMfWAKJygTY3fda7BbqG/t3sRBJVJy8joW61+6XgTkxWcurJ0v
EnAxD3HNHiifWg68oKtQCwCgosnR/am0VCQRRZfeDw3zjGU3yWgPevYCRyZW8SeE
40IJyHk5qeBvTSOX3bOKGfH9Czvn5shHphMzLBWffgK6GfSvi92+HGja2gTL4/QH
GYnBmMngSijd5ZyzM5R5MAXVubRZKYFFYspOnO3zZ2m01wn5iXzZaTEFh4EEOYFF
4koerO064EsI7ums2X/ooAYwOM5/slWxtwtI7QJtGqbQs2y4PyoWoBZsjbvo+l4E
oVVajvy9a2ufWg484Mvgs2y4P6pWAOZEmbFt7sDb0MOf9Q/6NtvXfnz1KubIx+bz
w+SxbW6Ae6Drq39KhvgCWoZgC7vn9jiX+UAb8LMHyWEtk/fB5W+yHbbQy2DCPBVG
OwE=
VDE.
RET.
```

Syntax

ANA.#FromVar|P1|P2

Parameter Explanation

***P1* - Image-Register number, can not be omitted.**
P2* - *Variable with Bitmap-Data.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.#FromVar/ToVar-Sample
'**********************************
ANA.New|0|640|400
ANA.vgradient|0|&HFF1010|&H10FFFF
ANA.show|0
' This will use IR 0 and TOS because no parameter were given.
ANA.ToVar|0|$$CON
'LEN.$$CON|$$LEN
' We put the COntent into IR 1
ANA.#FromVar|1|$$CON
' We display the IR 1
ANA.invert|1
ANA.show|1!
ENR.
```

```
'**********************************
' ANA.#FromVar-Sample
' using the VDA./VDE.-Feature
'**********************************
GSB.Laa
ANA.#FromVar|1|$$DTA
ANA.Show|1#!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'============================================================
:Laa
'------------------------------------------------------------
' BITMAP from Clipboard(.bmp).
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  1392 Bytes in  22 Lines.
VDA.$$DTA|$$FIN=
?MAAGuBQAASlG0yTbYYlYMQvkOznPrYa3D1MGg62CLOJf5wNvQQ99VP4L6gQQ5AQXC
OpZoDz0mKBeeLz0riH9lGqZQA79lN4b4AtrGQBuz43T5ifxZuUGlkhKiUgJlmuYw
k3eBRZ9a0WwN+wjqxthPIlWS4rJSTa7stW/t3kQBBcK6Vgi/hQq+XAXCutbI35Vy
cnr2e4dGgPuje8jH5vNDpNH9JmQJeUncSSENg3+Fx97EhJVfEiLVUgKi6nkAOwA7
Q6tQi8CGFKHNY+9du0AbM6NUiUHgy+AzrJivUop+P6VXFqAW7M3beMK6Vgg/Rapu
dHEj5VTggYgu0pyzMxTHsdsVxDLgZZmrHywVUnJyvaZYhz4U7MHlb7IdttBLoBZs
DXvIFVaAekJ5xx+vZ5VydaSQNX0onmz9e0hVNoD7d3Y2RgQKWaHdmbz4CQArkK5g
+wgnFmufWo78ANv4v2o5oOvoz9XxAKuGe4DygnB5IQX3DmfZabFFJ9Hszdv43VVm
gFOfF99Jkd2W7D2ryF9VkuKy0g3O3E2jjw0D/LXUnIcBFGQxZfqCag4kGRm0PSsI
P8VqjuRJZwmBzcvjMop+P+VntliHAcTZ0f2mpGkpkXifWg48BZt2E9JshcfetNEY
b7LaxgNLoFM84EsgExxwA4RhtYqW4LPs8I/ortxFYspOnG2zB8lhLROSscWS3odl
J4Z4//o2WxeebI3+2EcepJSt09RBYAuA4/SHmVkoq1MvjZv209TBoOvgSxinkQw9
g8R1Zca9td60QSUPcvJ9Brjf+gHE2dF95kQZ8Q0+ePfmyJKWYDladdJBeP96djtn
JqhXvp91T5rmmNOaAdtG+LdGiP8lOhYorw3OXA2Df4VHHqQUbbNEmYAdm0Bzr7HY
134/5ef2uFdZECOMfWAKJygTY3fda7BbqG/t3sRBJVJy8joW61+6XgTkxWcurJ0v
EnAxD3HNHiifWg68oKtQCwCgosnR/am0VCQRRZfeDw3zjGU3yWgPevYCRyZW8SeE
40IJyHk5qeBvTSOX3bOKGfH9Czvn5shHphMzLBWffgK6GfSvi92+HGja2gTL4/QH
GYnBmMngSijd5ZyzM5R5MAXVubRZKYFFYspOnO3zZ2m01wn5iXzZaTEFh4EEOYFF
4koerO064EsI7ums2X/ooAYwOM5/slWxtwtI7QJtGqbQs2y4PyoWoBZsjbvo+l4E
oVVajvy9a2ufWg484Mvgs2y4P6pWAOZEmbFt7sDb0MOf9Q/6NtvXfnz1KubIx+bz
w+SxbW6Ae6Drq39KhvgCWoZgC7vn9jiX+UAb8LMHyWEtk/fB5W+yHbbQy2DCPBVG
OwE=
VDE.
RET.
```

Remarks

This Command starts with a "#" due to internal Speed reasons ("#FromVar"). The reason why we did that is, that general variable resolution can not be used on Binary picture-Content, and the # signals this to the robot-engine.

Limitations:

-

See also:

- ANA.ToVar 1424
- GRA.vToBMP 1797
- Variables with Inline Data 3516
- Inline-Data and Inline-Pictures 3483

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.2 ANA.FromClipboard

MiniRobotLanguage (MRL)

ANA.FromClipboard

Transfer Image from Clipboard into IR

Intention

This command will copy any picture that is in the Windows-Clipboard into the Image-Register that is specified using **P1**.
It is the Opposite Command to "Ana.ToClipboard".

Use ANA.FromClipboard to store Clipboard-Content in IR 0 and display it
This will use IR 0 and also show the Contents.

```
ANA.FromClipboard|0
ANA.Show|0!
ENR.
```

Syntax

ANA.FromClipboard[|P1]

Parameter Explanation

***P1* - *opt.* Image-Register number, if omitted, IR 0 will be used.**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'*********************************
' ANA.FromClipboard-Sample
'*********************************
ANA.FromClipboard|0
ANA.Show|0!
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ANA.#FromVar 1406
- CLP. - Clipboard - Operation 2009

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.3 ANA.Load

MiniRobotLanguage (MRL)

ANA.Load

Load an Image-Register (IR) from diskfile, using original size or "Load-Stretch"

Intention

This command will Load an specified IR from a diskfile.
It is the counterpart to "ANA.Save".

This will Save IR 0 and Load it back into IR 1.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF1010|&H10FFFF
ANA.show|0
VAF.$$FIL=?path\Myfilename.png
' This will Save IR 0 to $$FIL
ANA.Load|0|$$FIL
'LEN.$$CON|$$LEN
' We will Load it back into IR 1
ANA.Load|1|$$FIL
' We display the IR 1
ANA.invert|1
ANA.show|1!
ENR.
```

You can omit the Parameters then Filename will be "`?pattern\BM_0.bmp`".
If you specify the IR-Number `ANA.Load|4`, then the Filename will be "`?pattern\BM_4.bmp`"

If you just write
`ANA.Load|1|<Filename>`
then the filename will be expandet to
`ANA.Load|1|?pattern\<Filename>.bmp`
which is `?path\Pattern\<Filename>.bmp`

If you specify **P3**, the Image will be stretched to the size of the specified IR.
In the Sample below we specify IR 1 in P3 so we get the size of the new created IR.

```
ANA.New|0|640|400
ANA.vgradient|0|&H10FF10|&H10FFFF
ANA.show|0
' This will Save IR 0 to $$FIL
VAF.$$FIL=?exeloc\BM_0.bmp
ANA.Save|0|$$FIL
ANA.new|1|1024|600
' We will Load it back into IR 1 using Stretch-Mode
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Load|1|$$FIL|1
' We display the IR 1
'ANA.invert|1
ANA.show|1!
ENR.
```

The ANA.Load supports these File-Extensions:

- BMP
- EMF
- JPG or JPEG
- GIF
- ICO
- PNG
- TIF or TIFF
- WMF

IMPORTANT NOTICE about missing Administration Rights Effects

When you start the Script from an SPR-Editor that has NO Admin-Rights, you may not be able to load pictures
from the Standard Robot path because its below C\Programme (x86)\ and you need Admin-Access to load files there.
As a result you will get only BLACK PICTURES using ANA.Load.
So if you only get Black Pictures, try to use ADMIN-RIGHTS for the Script.

Syntax

ANA.Load[|P1][|P2][|P3]

Parameter Explanation

P1 - opt. Image-Register number, if omitted, IR 0 will be used.

P2 - opt. *File-Path and Filename*, if omitted "?exeloc\BM_(IR-Number).bmp" will be used.

P3 - opt. Image-Register number, if given, the Image will be streched to the size of the Content of that IR.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

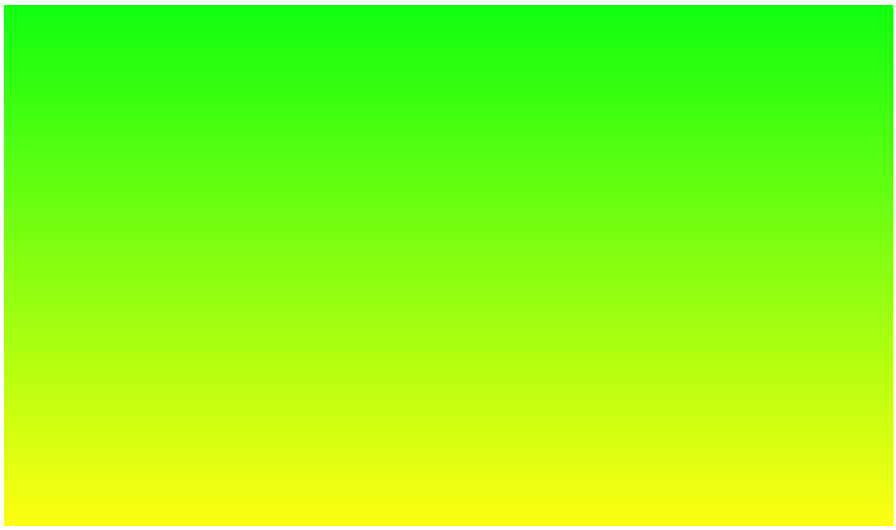
Example

```
'**********************************
' ANA.Load/Load-Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
ANA.New|0|640|400
ANA.vgradient|0|&H10FF10|&H10FFFF
ANA.show|0
' This will Save IR 0 to $$FIL
VAF.$$FIL=?exeloc\BM_0.bmp
ANA.Save|0|$$FIL
ANA.new|1|1024|600
' We will Load it back into IR 1 using Stretch-Mode
ANA.Load|1|$$FIL|1
' We display the IR 1
ANA.invert|1
ANA.show|1!
ENR.
```

Here you can see the result of the Operation.



Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.4 ANA.Save

MiniRobotLanguage (MRL)

ANA.Save

Save an Image-Register (IR) to diskfile

Intention

This command will save an specified IR to a diskfile.
It is the counterpart to "ANA.Load".

This will Save IR 0 and Load it back into IR 1.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF1010|&H10FFFF
ANA.show|0
VAF.$$FIL=?path\Myfilename.png
' This will Save IR 0 to $$FIL
ANA.Save|0|$$FIL
'LEN.$$CON|$$LEN
' We wil lLoad it back into IR 1
ANA.Load|1|$$FIL
' We display the IR 1
ANA.invert|1
ANA.show|1!
ENR.
```

You can omit the Parameters then Filename will be "`?exeloc\BM_0.bmp`**".**
If you specify the IR-Number `ANA.Save|4`**, then the Filename will be "**`?exeloc\BM_4.bmp`**"**

The `ANA.Save` supports these File-Extensions:

- BMP
- EMF
- JPG or JPEG
- GIF
- ICO
- PNG
- TIF or TIFF
- WMF

Syntax

ANA.Save[|P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - opt. **Image-Register number, if omitted, IR 0 will be used.**

P2 - opt. *File-Path and Filename*, **if omitted** "?exeloc\BM_(IR-Number).bmp" **will be used.**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Save/Load-Sample
'**********************************
ANA.New|0|640|400
ANA.vgradient|0|&HFF1010|&H10FFFF
ANA.show|0
' This will Save IR 0 to $$FIL
ANA.Save
VAF.$$FIL=?exeloc\BM_0.bmp
' We wil lLoad it back into IR 1
ANA.Load|1|$$FIL
' We display the IR 1
ANA.invert|1
ANA.show|1!
ENR.
```

Here you can see a part of the result of the Operation.

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.5 ANA.Snap

MiniRobotLanguage (MRL)

ANA.Snap

Capture Screenshot from localized WIndow and place it in an Image-Register (IR)

Intention

This command will capture a Screenshot from a specified window for further processing.

1. Capture Screenshot of complete Desktop and use it for Pattern-Search
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|Progman|Program Manager
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This is the Pattern we search for
$$PAA=?path\Help16.bmp
' We load the Pattern into IR 1
ANA.Load|1|$$PAA

' This is the Pattern Searching Command,
' Due to the value "1" as last Parameter, we will get a Debugprint
' of all found Hits. The displayed Point is in the Center of the Pattern.
ANA.fpa|0|1|0|125|1
' This command will put the results of the search on to the Stack.
ANA.gres
' Number of found Patterns is on TOS
POP.$$POA
MBX. We got $$POA Results
' We do not need first 3 Found Pattern so we remove them from Stack
STJ.Remo|0,1,2
' This is the Pattern we drive to
POP.$$POA
MMV.t|$$POA
MBX.!
ENR.
```

2. Capture Screenshot of complete Desktop and save into a file
This will include all Screens on a Dual-Screen System. They will be saved as a single ".bmp" (BItmap-) File. You can choose any widely used picture format, for example ".png" or "Jpg".

```
STW.ct|Progman|Program Manager
ANA.Snap|0
$$FIL=?path\Desktop.png
ANA.Save|0|$$FIL
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

3. Capture Screenshot of complete Desktop into a variable

This will include all Screens on a Dual-Screen System. They will be saved as a single Screenshot into the given Variable.

```
STW.ct|Progman|Program Manager
ANA.Snap|0
ANA.ToVar|0|$$FIL
LEN.$$FIL|$$RES
MBX.$$RES
ENR.
```

4. Capture Screenshot of actual window into image register
If you just locate any window, or child window, you can take a screenshot of these windows.

```
STW.ct|#32770|Basic-NAV Matrix 02
' We snap into the IR 2 here
ANA.Snap|2
ENR.
```

Syntax

ANA.Snap[|P1][|P2]

Parameter Explanation

P1 - opt. Image-Register number, if omitted, IR 0 will be used.
P2 - opt. In case there are Problems that the Mouse moves to much up or down by the height of the Titlebar, you can choose here another value.
0 - Snap only Client-Area from Window (default)
1 - Snap whole Window, remove Borders
2 - Snap whole Window incl. Borders
4 - Use internal default value that can be changed with the ANA.SetData - Command

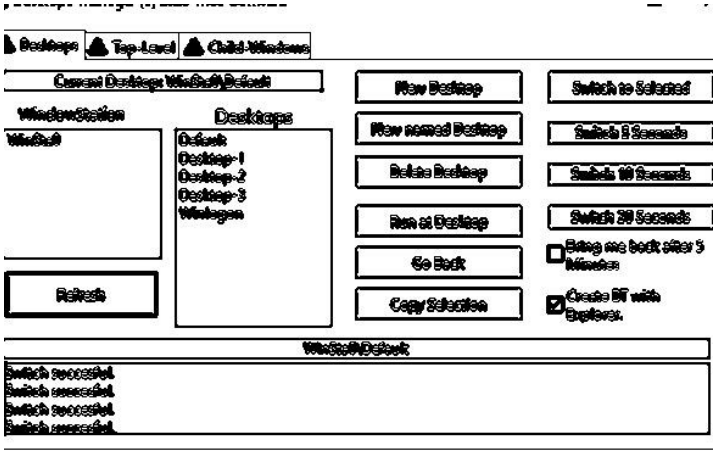
The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Snap-Sample
'**********************************
STW.ct|Progman|Program Manager
ANA.snap|0
ANA.Copy|0|1
ANA.BlockTo|0|0|2
ANA.rct|0|&H0|260|&H0
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  ANA.rct|0|&HFFFFFF|200|&HFFFFFF
  ANA.logic|0|1|0|A XOR B NOT T|RGB
  ANA.rct|0|&H0|260|&H0
  ANA.rct|0|&HFFFFFF|200|&HFFFFFF
  ANA.show|0!
  ENR.

' Here you can see a part of the result of the Operation.
```



Remarks

-

Limitations:

-

See also:

- Inline-Data and Inline-Pictures 3483

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.6 ANA.ToClipboard/Rect

MiniRobotLanguage (MRL)

ANA.ToClipboard

Copy Image from IR to Clipboard

Intention

This command will copy any picture that is in an specified IR, into the Windows-Clipboard.
It is the Opposite Command to "Ana.FromClipboard".

Use ANA.FromClipboard to store Clipboard-Content in IR 0 and display it
This will use IR 0 and also show the Contents.

```
$$PAF=?exeloc\Pics
$$FIL?$$PAF\P4.jpg
ANA.Load|0|$$FIL
' The CutRect Command is optional, it can be used to "Cut the borders" of an IR
ANA.CutRect|0|150|160|400|200
ANA.ToClipboard|0
' Picture is now in the Windows Clipboard
ENR.
```

ANA.ToClipboardRect is a bit special. Assume you have a picture of 640x400 and you specify:

```
ANA.ToClipboardRect|0|50|50|640|400
```

Then you may get a black Border of size 50 on the right side. This is because the limitation is the original Picture-Size.
If you specify instead a Target-Rectangle-Size that is larger then the original size, you will get a perfect match without a black Border.

```
ANA.ToClipboardRect|0|50|50|6400|4000
```

Of course you can specify exactly what you want, in such case please note that the X- and Y-Size is reduced by the values of how much you cut out at the upper and left side.
So this line would also do it:

```
ANA.ToClipboardRect|0|50|50|590|350
```

Syntax

ANA.ToClipboard[|P1]

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.ToClipboardRect[|P1][|P2|P3|P4|P5][|P6]

Parameter Explanation

P1 **- *opt.* Image-Register number, if omitted, IR 0 will be used.**

' If given, you must specify all 4 Rect-Parameters
P2 **- *Left-Border of Image to copy***
P3 **- *Upper/Top-Border of Image to copy***
P4 **- *Right-Border of Image to copy. If too large, will be replaced by Image maximum size.***
P5 **- *Bottom-Border of Image to copy. If too large, will be replaced by Image maximum size.***

P6 **- *Blit-Constant Parameter*, if omitted Standard will be used. Details see** **Blit-Modes** 1156

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.ToClipboard-Sample
'***********************************
$$PAF=?exeloc\Pics
$$FIL=$$PAF\P4.jpg
ANA.Load|0|$$FIL
ANA.CutRect|0|150|160|4000|2000
ANA.ToClipboard|0
ANA.Show|0!
ENR.

'***********************************
' ANA.ToClipboardRect-Sample
'***********************************
$$PAF=?exeloc\Pics
$$FIL=$$PAF\P4.jpg
ANA.Load|0|$$FIL
ANA.ToClipboardRect|0|150|88|290|189|2
ANA.FromClipboard|1
ANA.Show|1!
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- Blit-Modes 1156
- ANA.FromClipboard 1410
- CLP. - Clipboard - Operation 2009

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.9.7 ANA.ToVar

MiniRobotLanguage (MRL)

ANA.ToVar

Transfer Image-Register (IR) Content into SPR-Variable

Intention

This command will transfer the IR Content into a SPR Variable. From there you can save it to disc, manipulate it yourself, or just keep it.
Using "FromVar" you can later transfer it back into an IR.

Putting a IR into a Variable using "ToVar" will produce a Variable with this format insider the Variable:
- 8 Byte (4 Byte X- and 4 Byte Y-Size Information)
- (X-size * y-size)*4 Byte per Pixel-graphical Data in BGR-Order.

1. Use ToVar with IR 0 and store Content on TOS
This will use IR 0 and store the Content on TOS.

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS|Desktops-Manager (c) 2020 Theo Gottwa
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This will use IR 0 and TOS because no parameter were given.
ANA.ToVar
' Here we get the TOS into the Variable
POP.$$CON
' We put the COntent into IR 1
ANA.#FromVar|1|$$CON
' We display the IR 1
ANA.show|1!
ENR.
```

2. Use ToVar with Parameters
This will include all Screens on a Dual-Screen System. They will be saved as a single ".bmp" (BItmap-) File. You can choose any widely used picture format, for example ".png" or "Jpg".

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS|Desktops-Manager (c) 2020 Theo Gottwa
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This will use IR 0 and TOS because no parameter were given.
ANA.ToVar|0|$$CON
' We put the Content into IR 1
ANA.#FromVar|1|$$CON
' We display the IR 1
ANA.show|1!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Syntax

ANA.ToVar|[|P1]

Parameter Explanation

P1 - opt. **Image-Register number, if omitted, IR 0 will be used.**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'*********************************
' ANA.ToVar/FromVar-Sample
'*********************************
ANA.New|0|640|400
ANA.vgradient|0|&HFF1010|&H10FFFF
ANA.show|0
' This will use IR 0 and TOS because no parameter were given.
ANA.ToVar|0|$$CON
'LEN.$$CON|$$LEN
' We put the COntent into IR 1
ANA.#FromVar|1|$$CON
' We display the IR 1
ANA.invert|1
ANA.show|1!
ENR.
```

Remarks

-

Limitations:

-

See also:

- ANA.#FromVar 1406
- Variables with Inline Data 3516

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.10 Logic-Operations

Logic Operations will combine the IR's using Logic like AND/OR etc.
These will work on Bit-Level.

Here is a Sample:

1. **Making an Outline using Logic-Operations:**

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAF.$$PIA=?exeloc\Pics\P3.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Copy|0|1
ANA.bw|0|500
ANA.bw|1|100
ANA.Invert|0
ANA.Invert|1
GSB.Save|0
GSB.Save|1
ANA.Logic|2|0|1|A XOR NOT B
'ANA.Invert|2
GSB.Save|2
ENR.

'============================================================
:Save
$$NUM=§§_01
RND.0|999999|$$RND
FMT.$$RND|000000
$$PAT=?exeloc\Pics\Save_$$RND.png
IEF.$$PAT
  GTO.Save
EIF.
ANA.Save|$$NUM|$$PAT
ANA.Show|$$NUM!
RET.
'============================================================
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.10.1 ANA.ColorDiffMap

MiniRobotLanguage (MRL)

ANA.ColorDiffMapX

Generate a Outline based on the Color-Difference in X-Direction

ANA.ColorDiffMapY

Generate a Outline based on the Color-Difference in Y-Direction

ANA.ColorDiffMapXY

Generate a Outline based on the Color-Difference in X and Y-Direction

Intention

This Command will generate an Outline of the objects in the IR, it will search and find the borders of Objects.
Use it together with the ANA.Contrast Operation to get the results you may want.

This Example will show ANA.ReplaceColorT to make a Cleanup of a picture from a social network.
You got that Picture and it has a lot of JPG-Artefacts and the colors are not pure.
The SPR can quickly clean it up.

```
'**********************************
' ANA.ReplaceColorT-Sample
' Used to clean up JPG-Artefacts
' IMPORTANT: Used Sample picture are used for technical reasons and "may or may
'**********************************
VAF.$$PIA=?exeloc\Pics\Sam1.jpg
NEF.$$PIA
  MBX.File not found.
EIF.

ANA.Load|1|$$PIA
ANA.Load|0|$$PIA
ANA.Show|0!

'Restore red
$$SRC=&HFF0000
$$COL=&HFF0000
$$TOL=95
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL

' Restore Blue
$$SRC=&H0000FF
$$COL=$$SRC
$$TOL=90
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
```

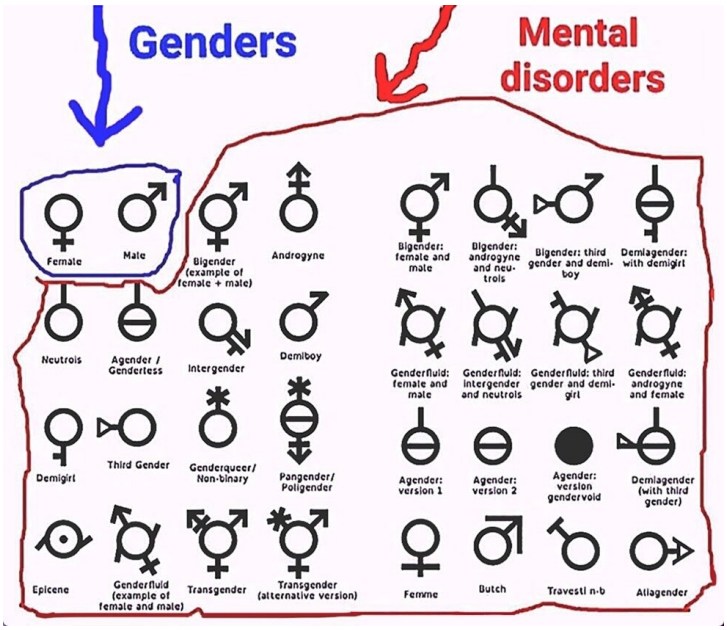

© 2023 Theo Gottwald * https://www.smart-package.com

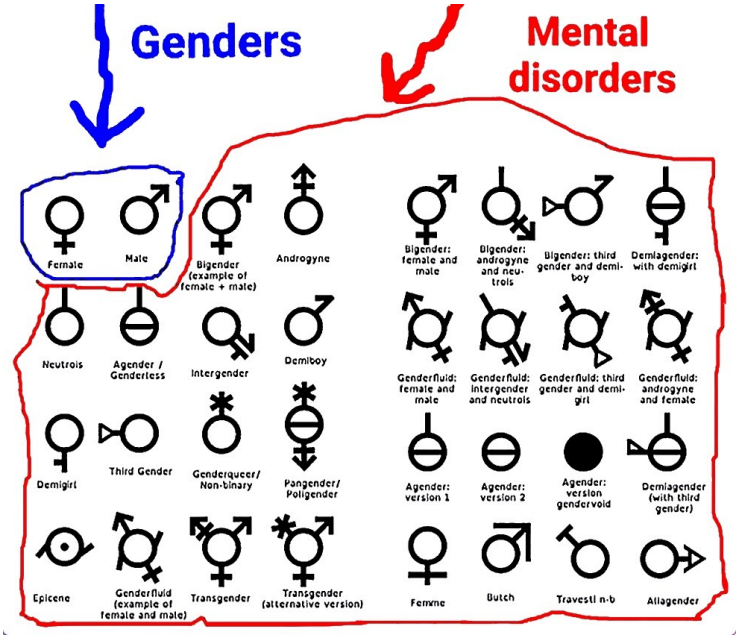
```
'Restore Black
$$SRC=&H0
$$COL=$$SRC
$$TOL=90
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL

'Restore White
$$SRC=&HFFFFFF
$$COL=$$SRC
$$TOL=90
ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL

ANA.Show|0!

' Generate Outline
ANA.cdfxy|0|1|1
ANA.Contrast|1|15000

ANA.Invert|1
ANA.Show|1!

' IR 2 must not be empty for the MixTar Command
ANA.Copy|0|2

' Combine both
ANA.MixTar|1|0|2
DMP.1
ANA.Show|2!
ENR.


' Below you can see the result of the Operation.
```

Original Picture with Artefacts - pure Colors

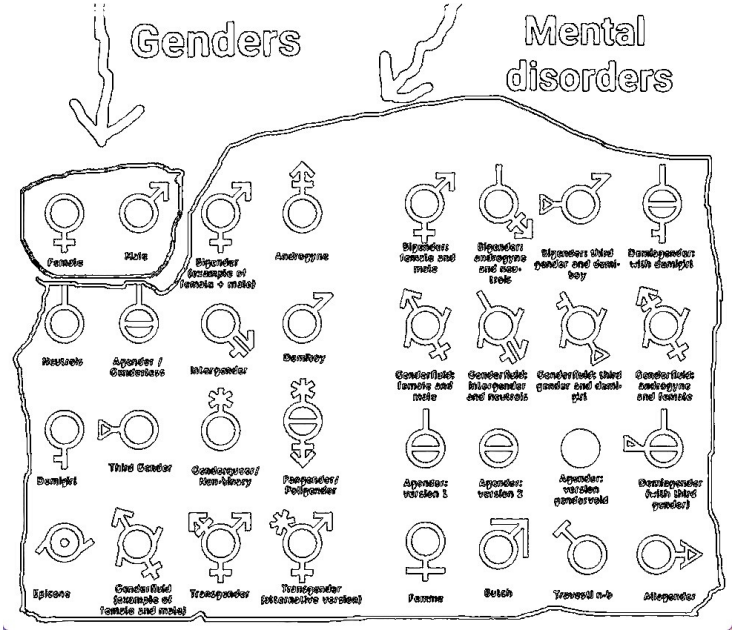
First Cleanup Generate Outline to clean up Borders

*© 2023 Theo Gottwald * https://www.smart-package.com*

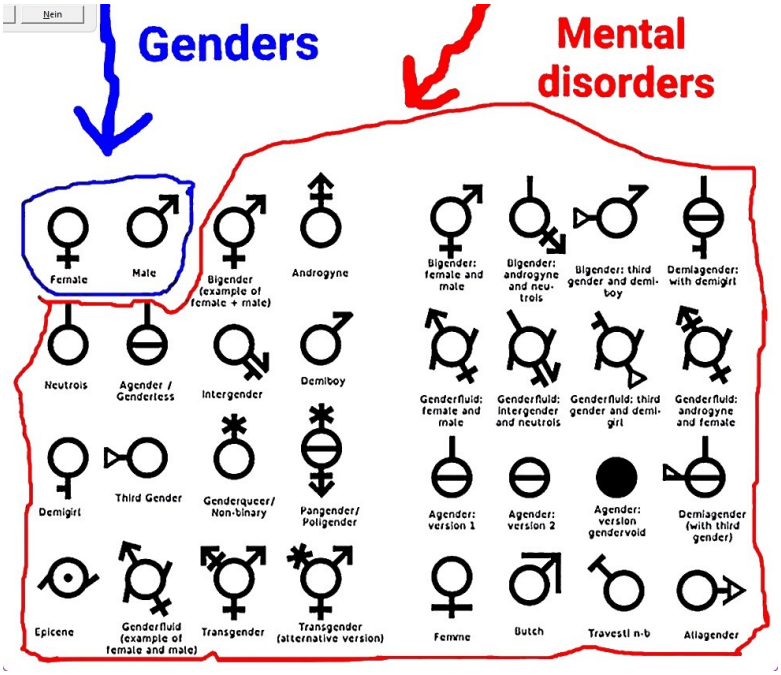




© 2023 Theo Gottwald * https://www.smart-package.com



Final Result



© 2023 Theo Gottwald * https://www.smart-package.com

```
010 0000000000000189 0001 VAF.$$PIA=?exeloc\Pics\Sam1.jpg
011 0000000000001934 0001 NEF.$$PIA
012 0000000000000000 0000 MBX.File not found.
013 0000000000000030 0001 EIF.
014 0000000000000005 0001
015 0000000000431193 0001 ANA.Load|1|$$PIA
016 0000000000389432 0001 ANA.Load|0|$$PIA
017 0000000029137881 0001 ANA.Show|0!
018 0000000000000009 0001
019 0000000000000003 0001
020 0000000000000150 0001 VAR.$$SRC=&HFF0000
021 0000000000000045 0001 VAR.$$COL=&HFF0000
022 0000000000000035 0001 VAR.$$TOL=95
023 0000000001883181 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
024 0000000000000010 0001
025 0000000000000003 0001
026 0000000000000084 0001 VAR.$$SRC=&H0000FF
027 0000000000000082 0001 VAR.$$COL=$$SRC
028 0000000000000039 0001 VAR.$$TOL=90
029 0000000001896552 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
030 0000000000000009 0001
031 0000000000000003 0001
032 0000000000000083 0001 VAR.$$SRC=&H0
033 0000000000000076 0001 VAR.$$COL=$$SRC
034 0000000000000036 0001 VAR.$$TOL=90
035 0000000001786782 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
036 0000000000000012 0001
037 0000000000000003 0001
038 0000000000000092 0001 VAR.$$SRC=&HFFFFFF
039 0000000000000079 0001 VAR.$$COL=$$SRC
040 0000000000000045 0001 VAR.$$TOL=90
041 0000000001765377 0001 ANA.ReplaceColorT|0|$$SRC|$$TOL|$$COL
042 0000000000000008 0001
043 0000000025520293 0001 ANA.Show|0!
044 0000000000000009 0001
045 0000000021380221 0001 ANA.cdfxy|0|1|1
046 0000000008048198 0001 ANA.Contrast|1|15000
047 0000000000069639 0001 ANA.Invert|1
048 0000000017254000 0001 ANA.Show|1!
049 0000000000019495 0001 ANA.Copy|0|2
050 0000000000000008 0001
051 0000000000136976 0001 ANA.MixTar|1|0|2
052 0000000000000011 0001
053 0000000000000000 0001 DMP.1
054 0000000000000000 0000 ANA.Show|2!
055 0000000000000000 0000 ENR.
```

Here is the Speed-Dump:

Syntax

ANA.ColorDiffMapX|P1|P2[|P3]
ANA.ColorDiffMapY|P1|P2[|P3]
ANA.ColorDiffMapXY|P1|P2[|P3]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **- *Source* Image-Register Number, if omitted 0 is used.**
P2 **- *Destination* Image-Register Number, if omitted 0 is used.**
P3 **- *(opt.) Value that is used for the internal "ToGrey" Conversion. This value will effect the result. Default is 1.***

The Command will not leave a 0/1 on the TOS.

Example

-

Remarks

This Command is really fast and will do its Job in a 1-2 Seconds.

Limitations:

-

See also:

- ANA.AnalyeColor 1225
- ANA.RemoveIndexColors 1254

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.10.2 ANA.Complement / ANA.Complementary_Color

MiniRobotLanguage (MRL)

ANA.Complement

Complement Colors in an Image-Register (IR)

ANA.Complementary_Color

Complement a specified color-value

Intention

This command will complement all Colours in an IR. Complementary Colors are very good for Overlays as they are good readable.
If you specify **P2**, then the result will be stored in another IR, **P2**.

1. Just complement IR 0
This is what happens if you do not specify any parameters.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.show|0!
ANA.Complement
ANA.show|0!
ENR.
```

2. Complement any specified IR
If you just specify the Source IR, it will also be the destination of the Operation.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.show|0!
ANA.Complement|0
ANA.show|0!
ENR.
```

3. Complement any specified IR to another IR
If you specify the Source IR and Destination IR, the Result will be in another IR. Anything that was in the Destination IR before will be discarded.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.copy|0|1
ANA.show|0!
ANA.show|1!
ANA.Complement|0|1
ANA.show|0!
ANA.show|1!
```

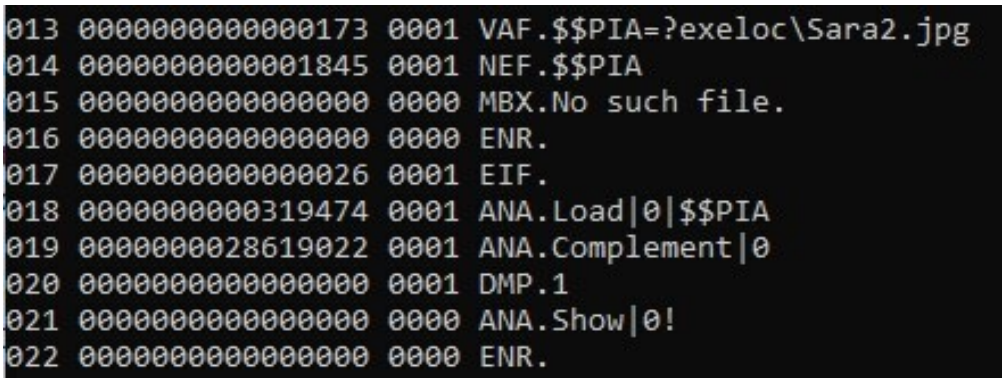
 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
ENR.
```

Here is another Sample Picture.

Original Picture **ANA.Complement**

*© 2023 Theo Gottwald * https://www.smart-package.com*



Sample for `ANA.complementary_color.`

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.GetRes|0|$$XRE|$$YRE|$$SIZ
FOR.$$LOX|0|($$XRE/3*2)|2
  PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|($$YRE/3*2)|1.6
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB
    ANA.complementary_color|$$PIB|$$PIC
    ANA.Setpixelcolor|0|$$LOX|$$LOY|$$PIC
  NEX.
NEX.
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.Complement[|P1][|P2]

Parameter Explanation

P1 - ***opt. Source*** Image-Register Number, Source-IR, if omitted, IR 0 will be used.
P2 - ***opt. Destination*** Image-Register Number, if omitted, IR **P1** will be used as destination IR.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

ANA.Complementary_Color|P1[|P2]

Parameter Explanation

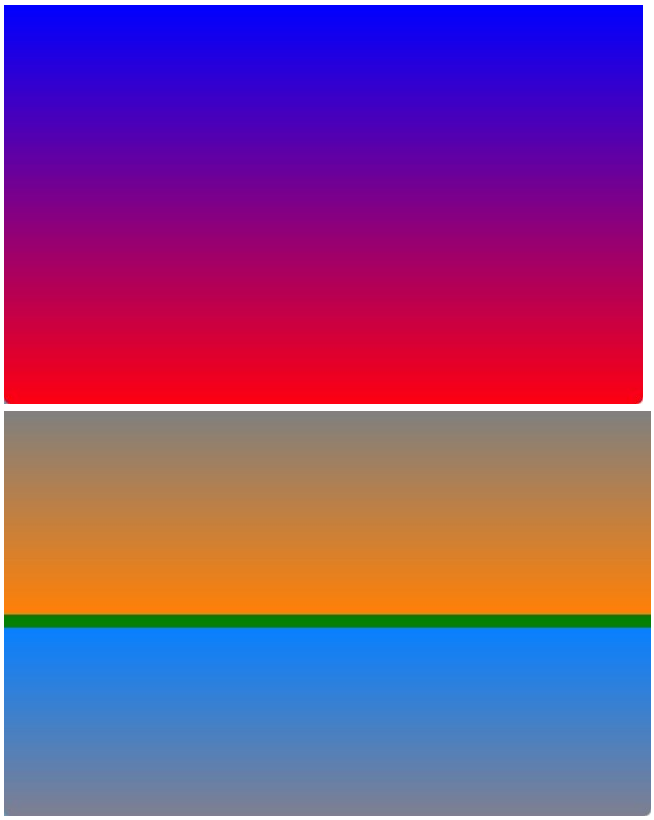
P1 - ***RGB-In Value, Example: &HF305A7***
P2 - ***opt. Variable*** for result. If omitted TOS is used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' ANA.Complement-Sample
'**********************************
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.copy|0|1
ANA.show|0!
ANA.show|1!
ANA.Complement|0|1
ANA.show|0!
ANA.show|1!
ENR.
```



Remarks

This command is not very optimized, it may take a few seconds for a pictures to get complemented.
If you need Speed, you may want to use ANA.Invert instead.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ANA.Invert 1442

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.10.3 ANA.Invert / ANA.Inverse_Color

MiniRobotLanguage (MRL)

ANA.Invert

Invert Colors in an Image-Register (IR)

ANA.Inverse_Color

Invert specified RGB-Color

Intention

This command will Invert a specified RGB COlor or a compelte Image-Regster.
It will make yellow from blue, and black will become white.
If you specify **P2**, then the result will be stored in another IR, **P2**.

1. Just Invert IR 0
This is what happens if you do not specify any parameters.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.show|0!
ANA.invert
ANA.show|0!
ENR.
```

2. Invert any specified IR
If you just specify the Source IR, it will also be the destination of the Operation.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.show|0!
ANA.invert|0
ANA.show|0!
ENR.
```

3. Invert any specified IR to another IR
If you specify the Source IR and Destination IR, the Result will be in another IR.
Anything that was in the Destination IR before will be discarded.

```
ANA.New|0|640|400
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.copy|0|1
ANA.show|0!
ANA.show|1!
ANA.invert|0|1
ANA.show|0!
ANA.show|1!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Here is another Sample Picture.

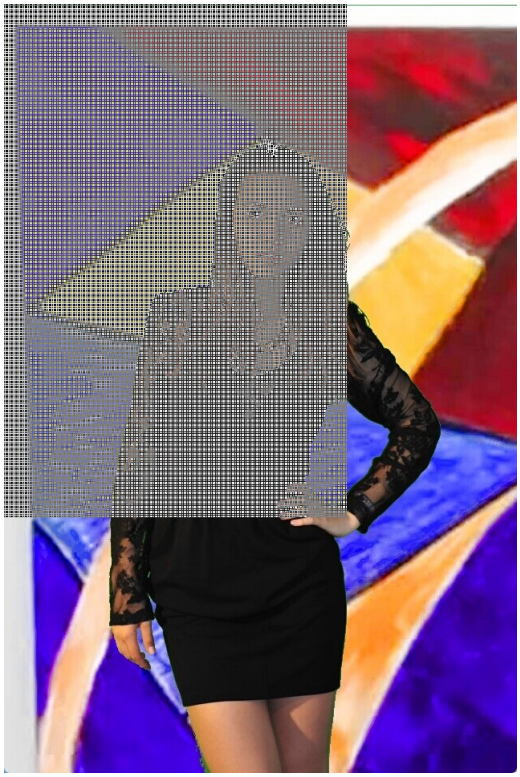
Original Picture **ANA.Inverse_Color**

4. Using `ANA.Inverse_color|$$RGB|$$OUT`
to work on partial Image-Registers.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.GetRes|0|$$XRE|$$YRE|$$SIZ
FOR.$$LOX|0|($$XRE/3*2)|1.5
  'PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|($$YRE/3*2)|1.5
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB
    ANA.inverse_color|$$PIB|$$PIC
    ANA.Setpixelcolor|0|$$LOX|$$LOY|$$PIC
  NEX.
NEX.
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

ANA.Invert[|P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - ***opt. Source*** Image-Register Number, Source-IR, if omitted, IR 0 will be used.
P2 - ***opt. Destination*** Image-Register Number, if omitted, IR **P1** will be used as destination IR.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

ANA.Inverse_Color|P1[|P2]

Parameter Explanation

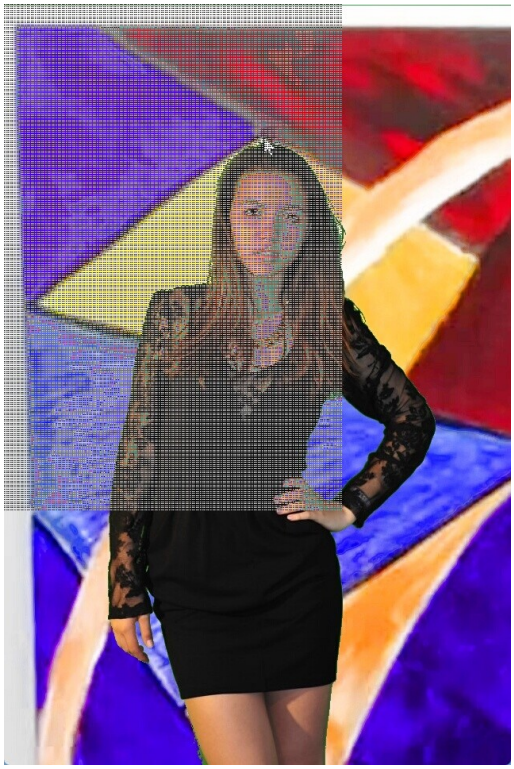
P1 - ***Source RGBIn-Value. Example: &H3AF80D***
P2 - ***opt. Destination Variable for result***, if omitted,TOS is used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Inverse_Color-Sample
'**********************************
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
ANA.GetRes|0|$$XRE|$$YRE|$$SIZ
FOR.$$LOX|0|($$XRE/3*2)|2.6
  'PRT.Column: $$LOX/$$XRE
  FOR.$$LOY|0|($$YRE/3*2)|2.6
    ANA.getpixelcolor|0|$$LOX|$$LOY|$$PIB
    ANA.inverse_color|$$PIB|$$PIC
    ANA.complementary_color|$$PIB|$$PID
    ANA.Setpixelcolor|0|$$LOX|$$LOY|$$PIC
    ANA.Setpixelcolor|0|($$LOX+1)|$$LOY|$$PID
  NEX.
NEX.
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ANA.Complement 1435

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.10.4 ANA.Logic

ANA.Logic - Combine IR with Logic Operation Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.Logic

Combine Image-Register (IR) Content with Logic Operations and with or without a Color-Mask

Intention

This command will combine the IR's using Logic like AND/OR etc. it will work on Bit-Level.

1. Logic with Source IR -A and B -> another IR
This Logic will distribute and then remove watermark artefacts using the ANA.Logic and the ANA.rct-Command.

```
' This Script is not intended for removing Copyright Artefacts
' yet i have nothing else at hand so i take it as an example
$$PAT=?path\Resized
$$FIA=$$PAT\BTC_1060.png
' We make a IR that is completely filled with the artefacts color
ANA.New|0|1060|1060
ANA.Fill|0|&H6f6d6e
ANA.Show|0!
' We load the Picture that contains the unwanted artefacts
ANA.Load|1|$$FIA
ANA.Show|1!
' We use the Logic Command to distribute the artefacts
ANA.Logic|2|1|0|A OR B
ANA.Show|2!
' Now we remove the artefacts using the "remove-color -with-Tolerance" Command.
ANA.rct|2|&H6f6d6f|10|0
ANA.rct|2|&H7f7d7f|10|0
ANA.Show|2!
ENR.
```

Picture 1 with grey artefacts, with distribution.

Picture 2 removed.

Picture 3 - Artefacts mostly

© 2023 Theo Gottwald * https://www.smart-package.com

->

=

© 2023 Theo Gottwald * https://www.smart-package.com

Making an Outline:

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAF.$$PIA=?exeloc\Pics\P3.jpg
NEF.$$PIA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Copy|0|1
ANA.bw|0|500
ANA.bw|1|100
ANA.Invert|0
ANA.Invert|1
GSB.Save|0
GSB.Save|1
ANA.Logic|2|0|1|A XOR NOT B
'ANA.Invert|2
GSB.Save|2
ENR.

'============================================================
:Save
$$NUM=§§_01
RND.0|999999|$$RND
FMT.$$RND|000000
$$PAT=?exeloc\Pics\Save_$$RND.png
IEF.$$PAT
  GTO.Save
EIF.
ANA.Save|$$NUM|$$PAT
ANA.Show|$$NUM!
RET.
'============================================================
```

Syntax

ANA.Logic[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - opt. Destination Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Source A Image-Register number, if omitted, IR **P1+1** will be used.
P3 - opt Source B Image-Register number, if omitted, IR **P2+1** will be used.
P4 - opt. Logical Operator: AND, OR, XOR, EQV, IMP, SIMP can be used.
Like this:

```
ANA.Logic|0|1|2|A AND B|RGB
ANA.Logic|0|1|2|NOT A AND B|RG
ANA.Logic|0|1|2|A OR B NOT T|B
```

where A and B are the two Source-IR's and T is the Target IR

© 2023 Theo Gottwald * https://www.smart-package.com

P5 - opt. Color Mask, can be any Combination of R G B, for example RG will allow Red and Green in the Result and Filter Blue
If omitted the default Mask is `RGB`.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Logic-Sample
' In this Sample we remove grey artefacts
' and save the result
'**********************************
$$PAT=?path\Resized
$$FIA=$$PAT\BTC_1060.png
ANA.New|0|1060|1060
ANA.Fill|0|&H6f6d6e
'ANA.Show|0!
ANA.Load|1|$$FIA
'ANA.Show|1!
ANA.Logic|2|1|0|A OR B
'ANA.Show|2!
ANA.rct|2|&H6f6d6f|10|0
ANA.rct|2|&H7f7d7f|10|0
ANA.Show|2!
ANA.Save|2|$$PAT\BTC_X_1060.png
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.10.5 ANA.Mask

MiniRobotLanguage (MRL)

ANA.Mask

Mask bits out of an entire Image-Register (IR)

Intention

This command will Mask Bits from an IR using AND/OR/XOR/EQV/IMP/SIMP.
It will work on Bit-Level.

1. Mask Source IR with a Bit-Combination
This will mask always the same picture with a defined Bit-Combination.

```
VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
'           Red    |   G   |   B
VAN.$$MAS=&B11100000100000011111111
GSB.DoIt|AND
GSB.DoIt|EQV
GSB.DoIt|SIMP
GSB.DoIt|OR
GSB.DoIt|IMP
GSB.DoIt|XOR
DMP.Speed
ANA.Show|1!
ENR.


:DoIt
VAR.$$OPE=§§_01
ANA.Mask|0|$$MAS|$$OPE|1
GSB.Print|1|$$OPE
RET.

:Print
VAN.$$NUM=§§_01
VAR.$$TXB=§§_02
DBP.$$MAS
VAN.$$COL=$H000000
VAN.$$LAA=&HFFFFFF
VAR.$$LAC=Segoe UI
VAN.$$XP1=100
$$XP1=16
$$YP1=60
VAN.$$SIZ=16
VAR.$$TXA=Mask: $$MAS -> &B11100000100000011111111
ANA.PrintAt|$$NUM|$$TXA|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
$$YP1=96
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$SIZ=64
ANA.PrintAt|$$NUM|$$TXB|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
ANA.Show|1!
RET.
```

Picture 1: Original Picture

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Speed-Dump. The ANA.Mask takes about a Second per Picture.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
013 0000000000000183 0001 VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
014 0000000000393061 0001 ANA.Load|0|$$PIA
015 0000000028255037 0001 ANA.Show|0!
016 0000000000000233 0001 VAN.$$MAS=&B11100000100000011111111
017 0000000000000196 0001 GSB.DoIt|AND
018 0000000000000335 0001 GSB.DoIt|OR
019 0000000000000199 0001 GSB.DoIt|XOR
020 0000000000000201 0001 GSB.DoIt|EQV
021 0000000000000201 0001 GSB.DoIt|IMP
022 0000000000000188 0001 GSB.DoIt|SIMP
023 0000000000000000 0001 DMP.Speed
024 0000000000000000 0000 ANA.Show|1!
025 0000000000000000 0000 ENR.
026 0000000000000000 0000
027 0000000000000006 0006 :DoIt
028 0000000000000101 0006 VAR.$$OPE=§§_01
029 0000000017350070 0006 ANA.Mask|0|$$MAS|$$OPE|1
030 0000000000000309 0006 GSB.Print|1|$$OPE
031 0000000000000018 0006 RET.
032 0000000000000000 0000
033 0000000000000006 0006 :Print
034 0000000000000140 0006 VAN.$$NUM=§§_01
035 0000000000000069 0006 VAR.$$TXB=§§_02
036 0000000000013116 0006 DBP.$$MAS
037 0000000000000179 0006 VAN.$$COL=$H000000
038 0000000000000083 0006 VAN.$$LAA=&HFFFFFF
039 0000000000000047 0006 VAR.$$LAC=Segoe UI
040 0000000000000129 0006 VAN.$$XP1=100
041 0000000000000040 0006 VAR.$$XP1=16
042 0000000000000040 0006 VAR.$$YP1=60
043 0000000000000076 0006 VAN.$$SIZ=16
044 0000000000000084 0006 VAR.$$TXA=Mask: $$MAS -> &B11100000100000011111111
045 0000000000032975 0006 ANA.PrintAt|$$NUM|$$TXA|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
046 0000000000000103 0006 VAR.$$YP1=96
047 0000000000000044 0006 VAR.$$SIZ=64
048 0000000000039187 0006 ANA.PrintAt|$$NUM|$$TXB|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
049 0000000275942336 0006 ANA.Show|1!
050 0000000000000062 0006 RET.
```

```
' This is the same Script with just another Bit-Combination
'
VAF.$$PIA=?exeloc\Pics\Sarah_BG.png
ANA.Load|0|$$PIA
ANA.Show|0!
'            Red    |   G   |   B
VAN.$$MAS=&B010101011010101001010101
GSB.DoIt|AND
GSB.DoIt|EQV
GSB.DoIt|SIMP
GSB.DoIt|OR
GSB.DoIt|IMP
GSB.DoIt|XOR
DMP.Speed
ANA.Show|1!
ENR.

:DoIt
VAR.$$OPE=§§_01
ANA.Mask|0|$$MAS|$$OPE|1
GSB.Print|1|$$OPE
RET.

:Print
VAN.$$NUM=§§_01
VAR.$$TXB=§§_02
DBP.$$MAS
VAN.$$COL=$H000000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAN.$$LAA=&HFFFFFF
VAR.$$LAC=Segoe UI
VAN.$$XP1=100
$$XP1=16
$$YP1=60
VAN.$$SIZ=16
VAR.$$TXA=Mask: $$MAS -> &B01010101101010100101010101
ANA.PrintAt|$$NUM|$$TXA|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
$$YP1=96
$$SIZ=64
ANA.PrintAt|$$NUM|$$TXB|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
ANA.Show|1!
RET.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

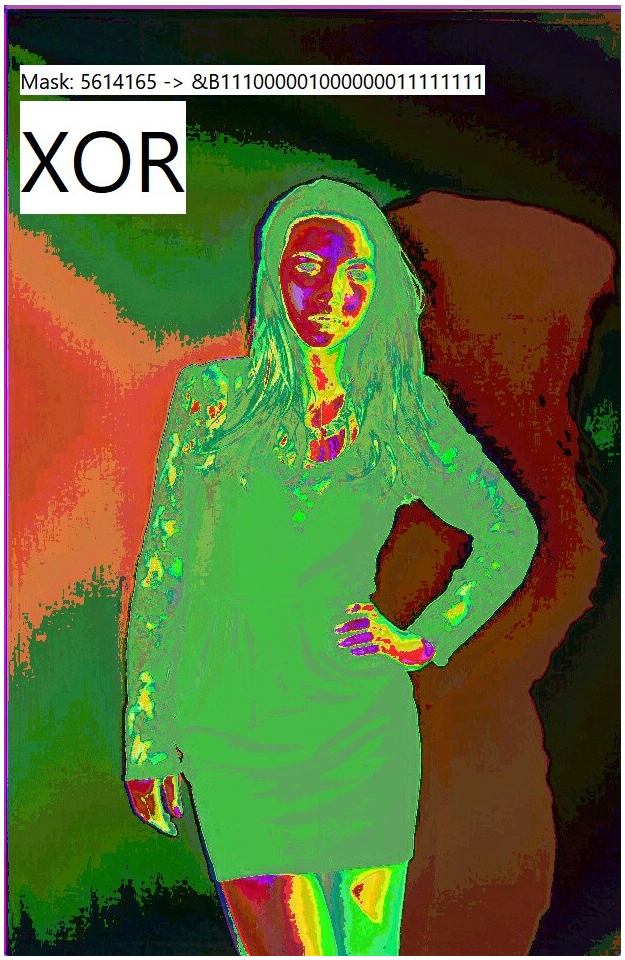
© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

ANA.Mask[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - opt. Source Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Mask - a 32 bit Integer can be given directly as Binary example: `&B01010101101010101001010101` if omitted, `&B11110000111110000011110000` is used.
P3 - opt. `Logical Operator: AND, OR, XOR, EQV, IMP, SIMP can be used. Like this:`
P4 - opt. Destination Image-Register number, if omitted, **P1** will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

-

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ANA.Logic 1449

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.11 Manage IR's

These Operations can be used to create an IR, or copy an IR.
IR's will automatically be deleted when the Script ends or when the same IR Number is used for something else.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.11.1 ANA.Compare

ANA.Compare - Compare IR vs. IR with Tolerance Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.Compare

Compare an Image-Register (IR) vs. another IR with Tolerance

Intention

This command will compare the Content of an IR with another IR, the result will be just 0 or 1 to show if the two are identical or not.
It can use a Fuzzy Logic-2 Step-Tolerance.
You can specify a Color-Tolerance between 0 and 442 and you can specify how many Pixels are allowed to be out of this Tolerance before the Compare fails.
The result will always be the Position of the Pixel that exceeds these two given Parameters.
For example: If you say "Tolerance 10, and 15 Pixel Count", the given Pixelwill be the 15th Pixel that does not match the Source Colour for more then the Pixel-Tolerance.
This Command can be combined with the "`ANA.BlockTo`" Command for Interesting effects.
If the two IR's that you want to Compare do not have the same resolution, you can use the `ANA.equalres` Command to enforce the same resolution.

Important: The Compare Operation currently is done Left->Right and Top->Bottom.
Both IR that will be compared, must have the same size!

1. Compare two IR with different results
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
ANA.New|0|160|40
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.copy|0|1
ANA.Compare|0|1
DMP.6
MBX.Result is zero, both are equal
ANA.vgradient|1|&HFF0000|&H1000F4
ANA.Compare|0|1
DMP.6
MBX.Result is Pixel Number 5442 at Position 2,34 exceeds the given Parameters.
ANA.Compare|0|1|4|100
DMP.6
MBX.Result is Pixel Number 2181 at Position 101,13 exceeds the given Parameters.
ENR.
```

Here is the result of the Stack for the first `DMP.6`

© 2023 Theo Gottwald * https://www.smart-package.com

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->LongInt      :0

Stackpos: 001 ->LongInt      :1

Stackpos: 002 ->LongInt      :4

-----------------------------------
```

Here is the result of the Stack for the second `DMP.6`

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->LongInt      :2181

Stackpos: 001 ->LongInt      :101

Stackpos: 002 ->LongInt      :13
```

Syntax

ANA.Compare|P1|P2[|P3][|P4]

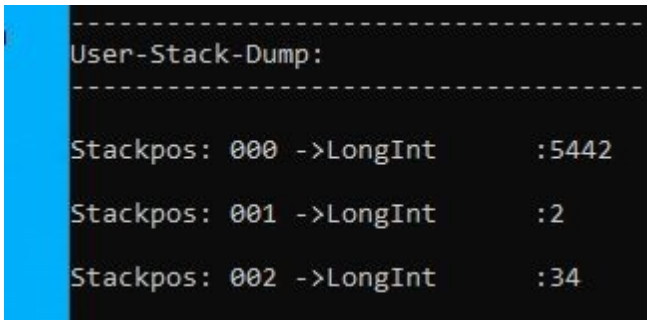
Parameter Explanation

P1 – Image-Register number
P2 – Image-Register number to compare with **P1**
P3 – ***opt.*** Colour-Tolerance between 0 and 442 (442 is the Colour-difference between black and white)
P4 – ***opt.*** Number of allowed "Hits", Pixels that do not match and do exceed the Color-Tolerance.

The Command will leave a 0 on the TOS if the two IR's do match.

If they do NOT match to each other, you will get a result like this:

© 2023 Theo Gottwald * https://www.smart-package.com



On TOS is the "Pixel-Number" of the last Pixel that exceeds the given conditions, below is the "X-Position" and below the "Y-Position" of that Pixel.

Example

```
'**********************************
' ANA.Compare-Sample
'**********************************
ANA.New|0|160|40
ANA.vgradient|0|&HFF0000|&H1000FF
ANA.copy|0|1
ANA.Compare|0|1
DMP.6
MBX.Result is zero, both are equal
ANA.vgradient|1|&HFF0000|&H1000F4
ANA.Compare|0|1
DMP.6
MBX.Result is Pixel Number 5442 at Position 2,34 exceeds the given Parameters.
ANA.Compare|0|1|4|100
DMP.6
MBX.Result is Pixel Number 2181 at Position 101,13 exceeds the given Parameters.
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.11.2 ANA.Copy

MiniRobotLanguage (MRL)

ANA.Copy

Copy Image Register (IR) to another Image-Register (IR)

Intention

This command will copy the Content of an Image Register to another Image Register.
The destination Image Register can be empty or not.
In case there is another picture in the destination IR, this will be overwritten.

```
STW.ct|#32770|Basic-NAV Matrix 02
' We snap into the IR 2 here
ANA.Snap|2
' Copy IR 2 into IR 1
ANA.Copy|2|1
' Display IR 1
ANA.Show|1!
ENR.
```

Syntax

ANA.Copy|P1|P2

Parameter Explanation

***P1* - *Source* Image-Register Number**
***P2* - *Dest* Image-Register Number**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Copy-Sample
'**********************************
STW.ct|Progman|Program Manager
ANA.Snap|0
ANA.Copy|0|1
ANA.BlockTo|1|1|4
ANA.show|1!
ENR.

' Below you can see a part of the result of the Operation.
' We used it here to have a content that can be copied.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

–

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.11.3 ANA.Move

MiniRobotLanguage (MRL)

ANA.Move

Move Image Register (IR) to another Image-Register (IR), deleting Source IR

Intention

This command will move the Content of an Image Register to another Image Register and then delete the original IR Content.
The destination Image Register can be empty or not.
In case there is another picture in the destination IR, this will be overwritten. The Source IR will be cleared.

```
STW.ct|#32770|Basic-NAV Matrix 02
' We snap into the IR 2 here
ANA.Snap|2
' Move IR 2 into IR 1
ANA.Move|2|1
' Display IR 1
ANA.Show|1!
ENR.
```

Syntax

ANA.Move|P1|P2

Parameter Explanation

***P1* - *Source* Image-Register Number**
***P2* - *Dest* Image-Register Number**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'*********************************
' ANA.Move-Sample
'*********************************
STW.ct|Progman|Program Manager
ANA.Snap|0
ANA.Move|0|1
ANA.BlockTo|1|1|4
ANA.show|1!
ENR.

' Below you can see a part of the result of the Operation.
' We used it here to have a content that can be moved.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

–

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.11.4 ANA.New

MiniRobotLanguage (MRL)

ANA.New

Create new, empty Image-Register (IR) or create new IR with resized Content from other IR

Intention

This command will create a new IR with a specified resolution.
Generally the new IR will be empty (Black). If **P4** is specified, then the content of IR P4 will be copied and resized to the resolution into the new IR.

1. Create new, empty IR
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|0
ANA.show|0!
ANA.New|0|800|200
' As you can see, IR 0 is deleted and has the specified size
ANA.show|0!
ENR.
```

2. Create new IR and copy the resized Content of another IR
As you can see below, the Resizing will not take any Aspect-Ratio into account, but use the full X and Y Size of the new IR.

```
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|2
ANA.New|0|800|200|2
ANA.show|0!
ENR.
```

 © 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.New|P1|P2|P3[|P4]

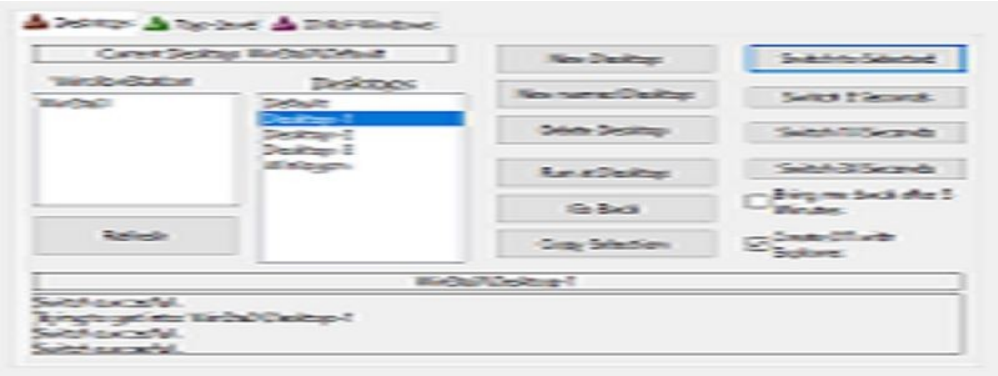
Parameter Explanation

P1 **- Image-Register Number for the IR** that will be created. If there is Content in this IR, it will be deleted.
P2 **- *X-Resolution* for the new IR Content**
P3 **- *Y-Resolution* for the new IR Content**
P4 **- *opt.* Image-Register Number**, if specified, the Content of this IR will be resized and copied into the new IR. If omitted, an empty IR will be created.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.New-Sample
'***********************************
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|0
ANA.BlockTo|0|1|5|2
ANA.New|0|800|300|1
ANA.show|0!
ENR.


' Here you can see the result of the Operation.
```



Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

•

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.11.5 ANA.Swap

MiniRobotLanguage (MRL)

ANA.Swap

Swap Image Register (IR) with another Image-Register (IR)

Intention

This command will exchange the Content of an Image Register with another Image Register.
The destination Image Register can not be empty.
In case there is another picture in the destination IR, this will be overwritten.

```
STW.ct|#32770|Basic-NAV Matrix 02
' We snap into the IR 2 here
ANA.Swap|2
' Swap IR 2 into IR 1 and IR 1 into IR 2
ANA.Swap|2|1
' Display IR 1
ANA.Show|1!
ENR.
```

Syntax

ANA.Swap|P1|P2

Parameter Explanation

P1 **- *Source* Image-Register Number**
P2 **- *Dest* Image-Register Number**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.Swap-Sample
'**********************************
STW.Act|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
ANA.Snap|0
ANA.BlockTo|0|1|40
' See the Content before the Swap
ANA.show|0!
ANA.show|1!
ANA.Swap|0|1
' See the Content after the Swap
ANA.show|0!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  ANA.show|1!
  ENR.


' Below you can see a part of the result of the
"BlockTo"Operation.
' We used it here to have a content that can be swapped.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

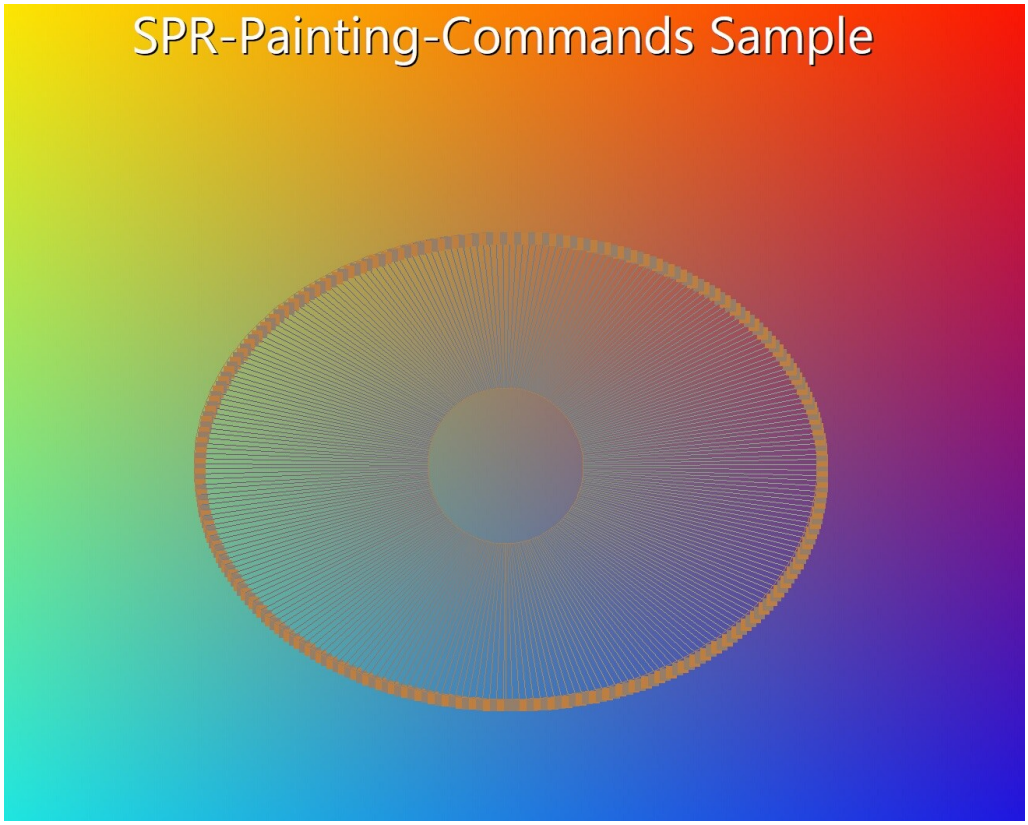
3.55.4.12 Paint Operations

Here you find some Basic Paint Operations which can be used to do complete image manipulation.

```
$$XPA=1600
$$YPA=1200

CAL.$$XPO=$$XPA/2|i
CAL.$$YPO=$$YPA/2|i

$$XSI=100
$$YSI=100

$$XSO=400
$$YSO=300

ANA.New|0|$$XPA|$$YPA
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|1|$$XPA|$$YPA
ANA.hgradient|1|&H0000FF|&H00FFFF
ANA.Mix|0|1|7
VAN.$$CNT=0
FOR.$$LRA|0|6.28|.0225
  CAX.$$SIN=SIN($$LRA)
  CAX.$$COS=COS($$LRA)

  CAL.$$XPI=$$SIN*$$XSI+$$XPO|i
  CAL.$$YPI=$$COS*$$YSI+$$YPO|i
  CAL.$$XPJ=$$SIN*$$XSO+$$XPO|i
  CAL.$$YPJ=$$COS*$$YSO+$$YPO|i

  'ANA.SetPixelColor|1|$$XPI|$$YPI|&HFF0000
  ANA.DrawLine|1|$$XPI|$$YPI|$$XPJ|$$YPJ|-1
  JIZ.$$XPL|elo
  ANA.GetPixelColor|1|$$XPM|$$YPM|$$COA
  ANA.GetPixelColor|1|$$XPL|$$YPL|$$COB
  ANA.DrawLine|1|$$XPI|$$YPI|$$XPL|$$YPL|$$COA
  ANA.DrawLine|1|$$XPJ|$$YPJ|$$XPM|$$YPM|$$COB
  ANA.DrawBox|1|$$XPJ|$$YPJ|($$XPJ+15)|($$YPJ+15)|$$COB|1
  :elo
  VAN.$$XPL=$$XPI
  VAN.$$YPL=$$YPI
  VAN.$$XPM=$$XPJ
  VAN.$$YPM=$$YPJ
  IVV.$$CNT>40
    ANA.Show|1
    $$CNT=0
  ELS.
    VIC.$$CNT
  EIF.
  '
NEX.
$$TXT=SPR-Painting-Commands Sample
$$COL=&HFFFFFF
  ' Backgroundcolor
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$LAA=-2
' Fontname
$$LAC=Segoe UI
$$XP1=320
$$YPI=40
ANA.PrintAt|1|$$TXT|0|$$LAA|($$XP1+2)|($$YP1+2)|39|$$LAC|1
ANA.PrintAt|1|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|39|$$LAC|1
ANA.Show|1!
ENR.
```



© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.1 ANA.DrawBox

MiniRobotLanguage (MRL)

ANA.DrawBox

Draw a Box into an Image-Register (IR)

Intention

The ANA.DrawBox command allows you to create a box within an Image Register (IR). This box can be either an outline or filled, and can be drawn in a solid color or with an inverted background color.

Creating a Box with Inverted Colors:

In this example, we'll create a background and then draw a filled box over it. We'll use the color "-1", which signifies "Invert Background Color".

Please note, due to the loop running 255 iterations, it's recommended to initiate the script in the Editor using the "Freerun Mode". The robot's drawing will not be visible until the Show command is executed.

Remember, the ANA commands provide you with a set of 20 Image Registers.
You can specify one or more source registers and sometimes a destination register in each command.
If a destination register is not specified, the source register will be overwritten with the result of the operation.
You can copy or move content from one Image Register to another, load an image into these registers, or snapshot a localized window into these registers. After filling the Image Registers with content, you can perform several types of processing with these images, and finally save the pictures.

To use ANA, you'll first need to fill the Image Registers (IRs) with some content, which will be processed in subsequent steps.
The first 20 Image Registers can be used freely, while registers 21-32 are reserved for internal operations.

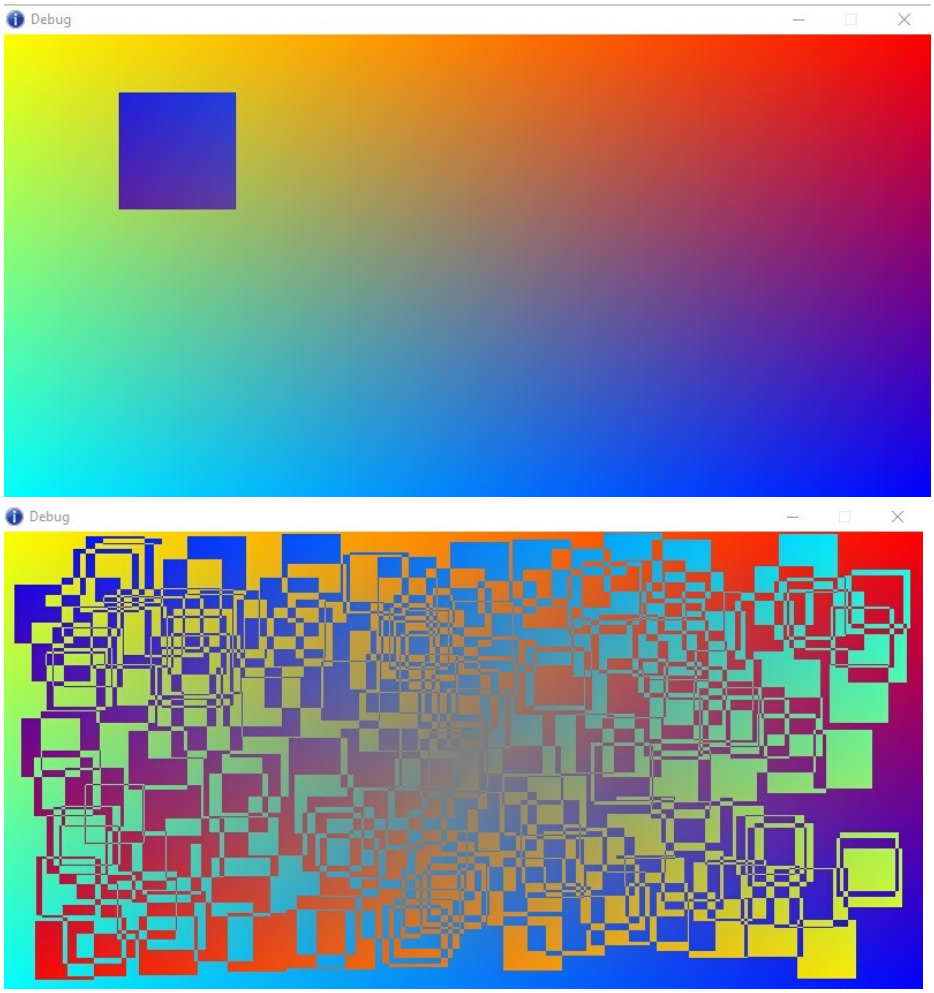
The Blit command uses the GPU for fast graphics operations, which may result in artifacts in some operations.
If you prefer to avoid these, you can use the "ANA.Logic" command, which uses the CPU.

You will not see the Robot Painting unless the Show command is been called.

```
ANA.New|0|800|400
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
ANA.Show|2!
ANA.DrawBox|2|100|50|200|150|-1|1
ANA.Show|2!
FOR.$$LJ7|1|255
  RND.0|740|$$XP1
  CAL.$$XP2=$$XP1+50
  RND.0|340|$$YP1
  CAL.$$YP2=$$YP1+50
  ANA.DrawBox|2|$$XP1|$$YP1|$$XP2|$$YP2|-1|1
NEX.
ANA.Show|2!
ENR.
```



This box is filled and it is drawn with inverted background color.
The second Part of the Script will draw 255 random filled boxes.

In the below example you will not see "how the robot paints". To refresh the Debug Display you generally need to call the "ANA.Show" Command like this:

```
' In this Sample the Debug Display is updated after every DrawBox-Command.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
FOR.$$LJ7|1|255
  RND.0|740|$$XP1
  CAL.$$XP2=$$XP1+50
  RND.0|340|$$YP1
  CAL.$$YP2=$$YP1+50
  ANA.DrawBox|2|$$XP1|$$YP1|$$XP2|$$YP2|-1|1
' Thiswill make the Drawing visible
  ANA.Show|2
  PAU..1
NEX.
ANA.Show|2!
ENR.
```

2. Draw only the Outline of the Box

In the following Script we will draw only the Outline of the box, not filling the box.
However we will draw 2500 Boxes. In "Freerun-Mode" this will take less than a second.
Infact the Robot can - in my test - draw up to 25000 unfilled Boxes per Second.

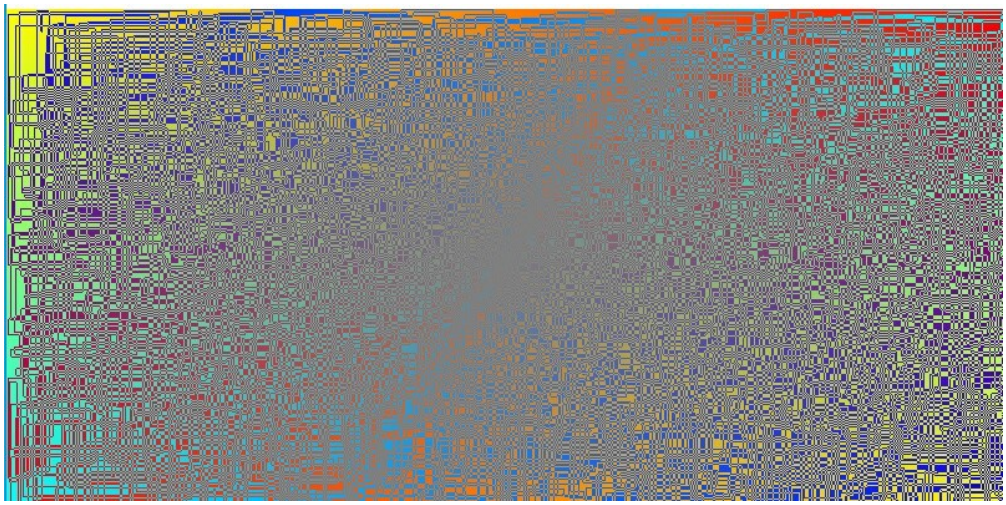
```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
FOR.$$LJ7|1|2500
  RND.0|740|$$XP1
  CAL.$$XP2=$$XP1+120
  RND.0|340|$$YP1
  CAL.$$YP2=$$YP1+80
  ANA.DrawBox|2|$$XP1|$$YP1|$$XP2|$$YP2|-1|0
NEX.
ANA.Show|2!
ENR.
```

Here is the Speed-Dump:

```
012 0000000000013883 0001 ANA.New|0|800|400
013 0000000000019916 0001 ANA.vgradient|0|&HFF00FF|&H000000
014 0000000000009288 0001 ANA.New|2|800|400
015 0000000000036653 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
016 0000000000043762 0001 ANA.Mix|0|2|7
017 0000000000000101 2501 FOR.$$LJ7|1|2500
018 0000000000000088 2500 RND.0|740|$$XP1
019 0000000000000172 2500 CAL.$$XP2=$$XP1+120
020 0000000000000090 2500 RND.0|340|$$YP1
021 0000000000000189 2500 CAL.$$YP2=$$YP1+80
022 0000000000000643 2500 ANA.DrawBox|2|$$XP1|$$YP1|$$XP2|$$YP2|-1|0
023 0000000000000006 2501 NEX.
024 0000000000000000 0001 DMP.1
```

Here is the result:

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

ANA.DrawBox[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - X1 - Lower Right Coordinat X-Value
P5 - Y1 - Lower Right Coordinat Y-Value
P6 - opt. Color-Value (like &HFF00FF), if omitted 0 is used. If -1 is given, the Pixel-Color will be inverted.
P7 - opt. Fill-Flag (0/1), if 1 the Box is filled. If its 0 the just the outline is been painted. If omitted 0 is used.

The Command will leave a 0 on the TOS in case of error, otherwise you get the number of copied patterns.

Example

```
'**********************************
' ANA.DrawBox-Sample
'**********************************
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
FOR.$$LJ7|1|25
  RND.0|740|$$XP1
  CAL.$$XP2=$$XP1+120
  RND.0|340|$$YP1
  CAL.$$YP2=$$YP1+80
  ANA.DrawBox|2|$$XP1|$$YP1|$$XP2|$$YP2|-1|1
  ANA.Show|2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  PAU..1
NEX.
ANA.Show|2!
ENR.
```

Remarks

A rather complex DrawBox-Command without Fill but with several Variables uses on my computer up to 640 Ticks. This **is 1/25000 th of a second on my Computer**.
A Drawbox invert or with color, with a filled Box can use more then 3000 Ticks on my Computer. Therefore the Robot could draw more then 5000 such filled Boxes per Second.

Limitations:

The Coordinates in the lower, right corner may not be smaller then the Coordinates in the upper, right ccorner.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.2 ANA.DrawLine

MiniRobotLanguage (MRL)

ANA.DrawLine

Draw a Line into an Image-Register (IR)

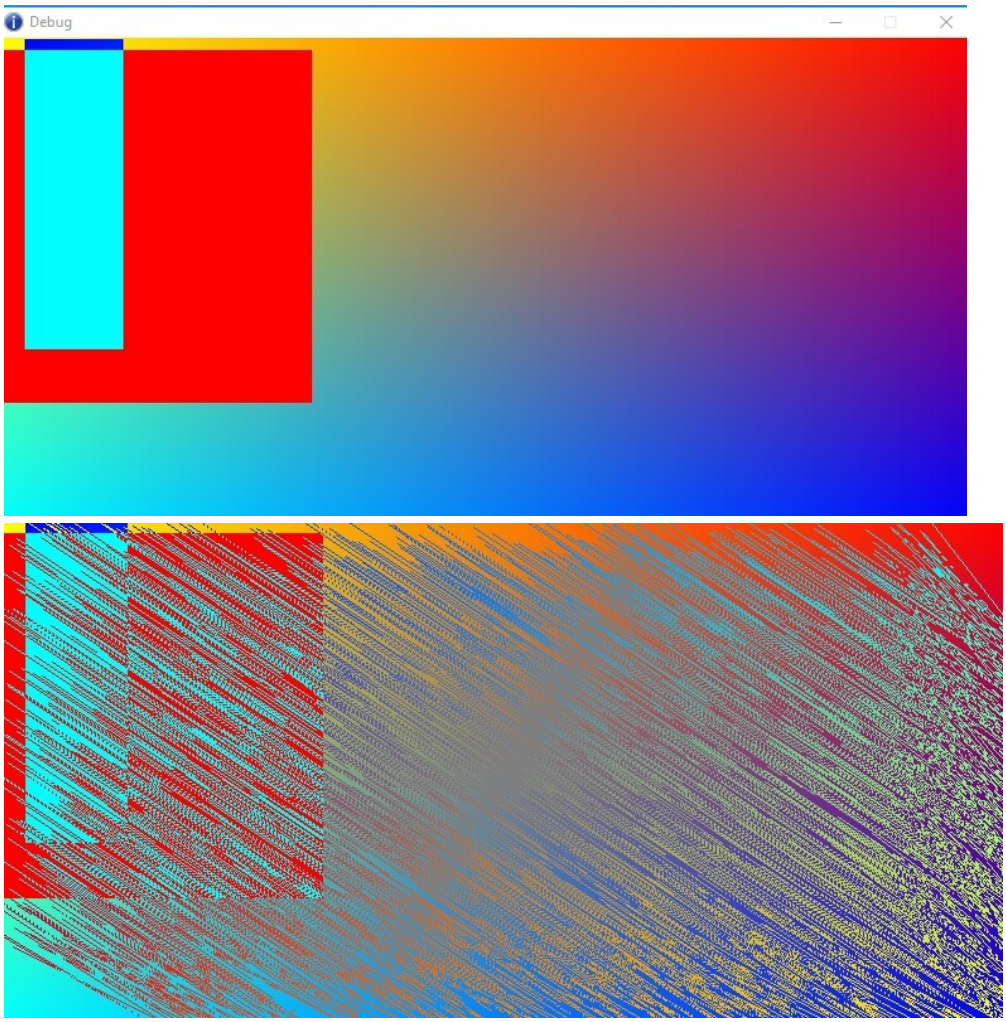
Intention

This command will draw a line into an IR.
The line can whether be drawn with a solid color or use inverted background-color.

1. Draw a Line that will invert the underlying Colors
In this Sample we will generate a Backgound and then Paint Lines over it. As Color we use "-1" that is "Invert Background Color".
Due to the use of a Loop with 255 Iterations, it is recommended to start the script in the Editor using "Freerun Mode".
You will not see the Robot Painting unless the Show command is been called.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
' Draw red line
FOR.$$LU3|1|255|1
  ANA.DrawLine|2|$$LU3|10|$$LU3|299|&HFF0000
NEX.
' Draw inverted Line
FOR.$$LU3|1|255|1
  ANA.DrawLine|2|100|$$LU3|20|$$LU3|-1
NEX.
ANA.Show|2!
FOR.$$LJ7|1|2500
  RND.0|740|$$XP1
  CAL.$$XP2=$$XP1+120
  RND.0|340|$$YP1
  CAL.$$YP2=$$YP1+80
  ANA.DrawLine|2|$$XP1|$$YP1|$$XP2|$$YP2|-1
NEX.
DMP.1
ANA.Show|2!
ENR.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*



This box is filled and it is drawn with inverted background color.
The second Part of the Script will draw 2500 random Lines.

```
012 0000000000014507 0001 ANA.New|0|800|400
013 0000000000020090 0001 ANA.vgradient|0|&HFF00FF|&H000000
014 0000000000009283 0001 ANA.New|2|800|400
015 0000000000033079 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
016 0000000000044849 0001 ANA.Mix|0|2|7
017 0000000000000013 0001
018 0000000000000129 0256 FOR.$$LU3|1|255|1
019 0000000000000455 0255 ANA.DrawLine|2|$$LU3|10|$$LU3|299|&HFF0000
020 0000000000000006 0256 NEX.
021 0000000000000004 0001
022 0000000000000100 0256 FOR.$$LU3|1|255|1
023 0000000000000402 0255 ANA.DrawLine|2|100|$$LU3|20|$$LU3|-1
024 0000000000000006 0256 NEX.
025 0000000000000004 0001
026 0000000000000000 0001 DMP.1
027 0000000000000000 0000 ANA.Show|2!
028 0000000000000000 0000
029 0000000000000000 0000 FOR.$$LJ7|1|2500
030 0000000000000000 0000 RND.0|740|$$XP1
031 0000000000000000 0000 CAL.$$XP2=$$XP1+120
032 0000000000000000 0000 RND.0|340|$$YP1
033 0000000000000000 0000 CAL.$$YP2=$$YP1+80
034 0000000000000000 0000 ANA.DrawLine|2|$$XP1|$$YP1|$$XP2|$$YP2|-1
035 0000000000000000 0000 NEX.
```

The Speed-dump shows that the Robot can draw several thousands Lines per second.

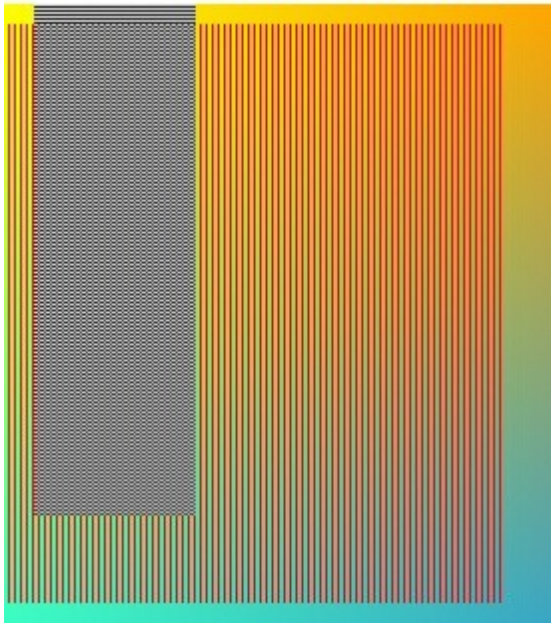
*© 2023 Theo Gottwald * https://www.smart-package.com*

2. Line-Patterns

Drawing Lines with a step of 2 or 3 can bring your display to the Limit of its abilities in showing clear colors.
If you draw Lines in a specified color and it looks like they have another color, then its a "Display issue".

.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
' Draw red line
FOR.$$LU3|1|255|3
  ANA.DrawLine|2|$$LU3|10|$$LU3|299|&HFF0000
NEX.
' Draw inverted Line
FOR.$$LU3|1|255|2
  ANA.DrawLine|2|100|$$LU3|20|$$LU3|-1
NEX.
ANA.Show|2!
ENR.
```



3. Draw inverted Lines

In the following Script we will draw sort of random lines.
We will draw 2500 Lines. In "Freerun-Mode" this will take less than a second.
Infact the Robot can - in my test - draw up to 10400 inverted Lines per Second.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
```

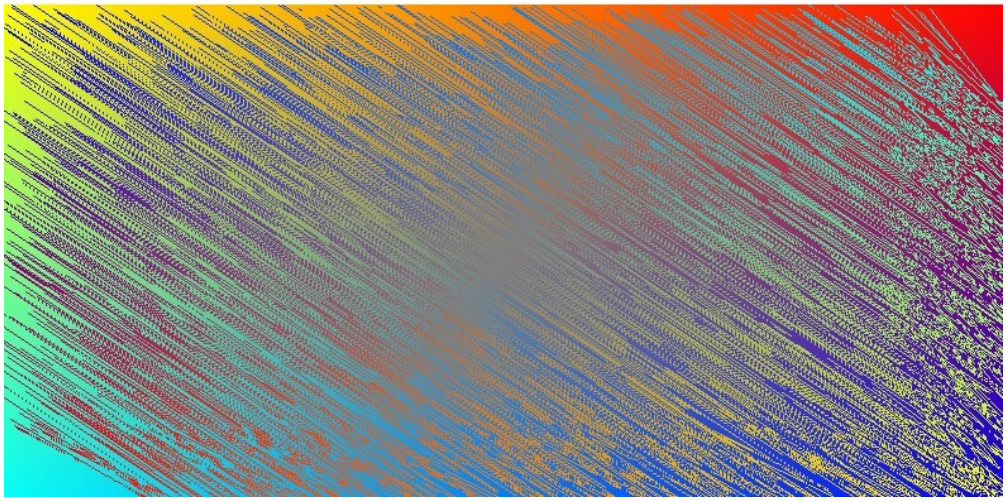

© 2023 Theo Gottwald * https://www.smart-package.com

```
FOR.$$LJ7|1|2500
  RND.0|740|$$XP1
  CAL.$$XP2=$$XP1+120
  RND.0|340|$$YP1
  CAL.$$YP2=$$YP1+80
  ANA.DrawLine|2|$$XP1|$$YP1|$$XP2|$$YP2|-1
NEX.
DMP.1
ANA.Show|2!
ENR.
```

Here is the Speed-Dump:

```
012 0000000000014341 0001 ANA.New|0|800|400
013 0000000000019643 0001 ANA.vgradient|0|&HFF00FF|&H000000
014 0000000000009416 0001 ANA.New|2|800|400
015 0000000000033432 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
016 0000000000043680 0001 ANA.Mix|0|2|7
017 0000000000000096 2501 FOR.$$LJ7|1|2500
018 0000000000000082 2500 RND.0|740|$$XP1
019 0000000000000158 2500 CAL.$$XP2=$$XP1+120
020 0000000000000086 2500 RND.0|340|$$YP1
021 0000000000000174 2500 CAL.$$YP2=$$YP1+80
022 0000000000001543 2500 ANA.DrawLine|2|$$XP1|$$YP1|$$XP2|$$YP2|-1
023 0000000000000006 2501 NEX.
```

Here is the result:



Syntax

ANA.DrawLine[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - X1 - Lower Right Coordinat X-Value
P5 - Y1 - Lower Right Coordinat Y-Value

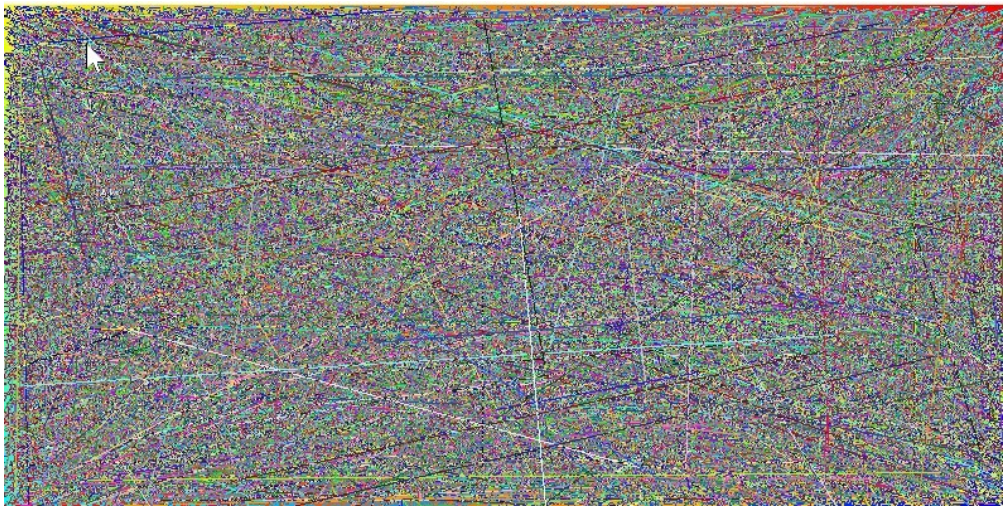
© 2023 Theo Gottwald * https://www.smart-package.com

P6 - opt. Color-Value (like &HFF00FF), if omitted -1 is used. If -1 is given, the Pixel-Color will be inverted.

The Command will leave a 0 on the TOS in case of error, otherwise you get the number of copied patterns.

Example

```
'**********************************
' ANA.DrawLine-Sample
'**********************************
' This Sample will Draw about 25000 Lines per Second
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
' Draw red line
FOR.$$LU3|1|25500
  RND.1|799|$$XP1
  RND.1|799|$$XP2
  RND.1|399|$$YP1
  RND.1|399|$$YP2
  RND.-1|1|$$COA
  IVV.$$COA<1
    VAN.$$COL=-1
  ELS.
    RND.0|&HFFFFFF|$$COL
  EIF.
  ANA.DrawLine|2|$$XP1|$$YP1|$$XP2|$$YP2|$$COL
NEX.
ANA.Show|2!
ENR.
```



Remarks

A rather complex DrawLine-Command with several Variables uses on my computer between 400 to 1540 Ticks. This **is at most 1/10000 th of a second on my Computer.**

© 2023 Theo Gottwald * https://www.smart-package.com

Therefore the Robot could draw more then 10000 Lines per Second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.3 ANA.Fill

ANA.Fill - Fill IR with Colour or Line-Pattern **Previous** 1088 **Top** 41 **Next** 1109

MiniRobotLanguage (MRL)

ANA.Fill

Fill Register (IR) with Color or Line-Pattern

Intention

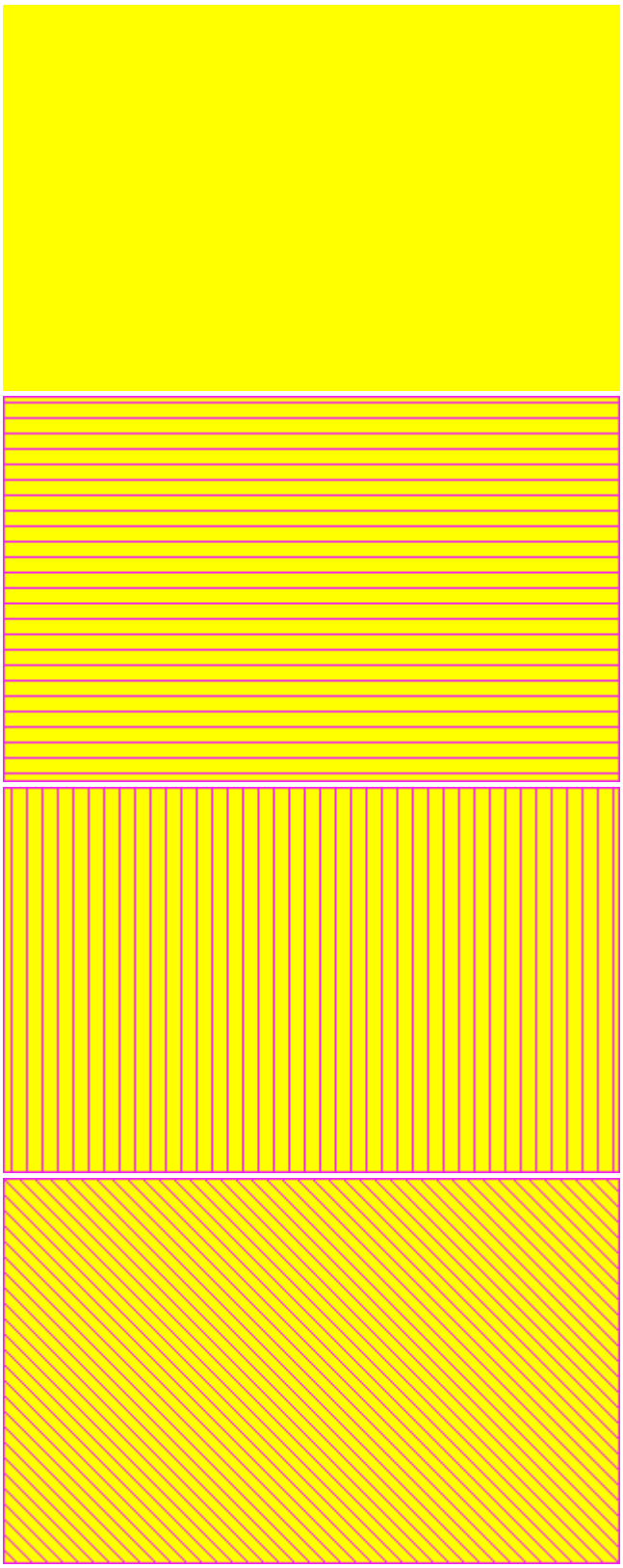
This command will fill an IR with a solid Background-Color. Optionally it will then draw a Pattern on this.
In case there is another picture in the IR, this will be overwritten.

```
STW.ct|#32770|Basic-NAV Matrix 02
' We snap into the IR 2 here
ANA.Snap|2
' Copy IR 2 into IR 1
ANA.Fill|2|1
' Display IR 1
ANA.Show|1!
ENR.
```

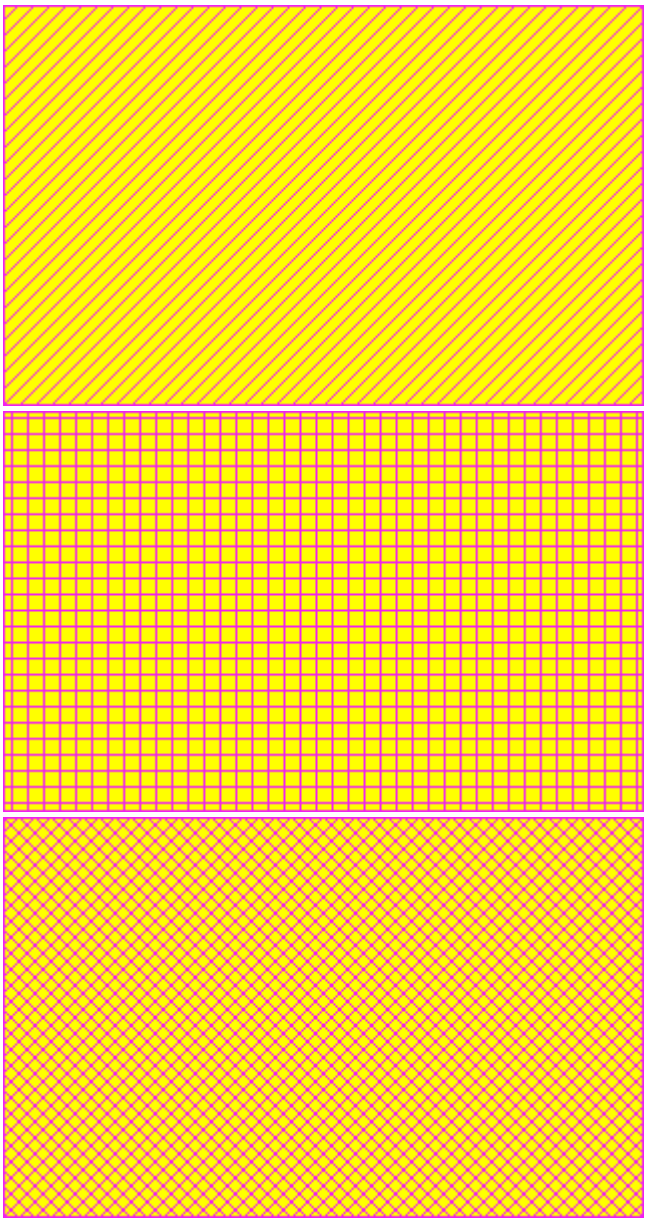
This will create a series of Patterns. Note that you can not just fill an empty IR, you need to create an IR first, using the `ANA.New`-Command.

```
ANA.New|0|320|200
FOR.$$LOP|0|6
  ANA.Fill|0|&H00FFFF|$$LOP|&HF020F0
  ANA.Save|0|?path\Pat_$$LOP.png
NEX.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

ANA.Fill[|P1][|P2][|P3]

Parameter Explanation

P1 - *opt. Source* Image-Register Number, if omitted 0 is used.
P2 - *opt. Color for Background,* if omitted White (&HFFFFFF) is used.
P3 - *opt.* Number for Pattern, if omitted 0 is used.
 0 - Solid (default)
 1 - Horizontal Lines
 2 - Vertical Lines
 3 - Upward Diagonal Lines
 4 - Downward Diagonal Lines
 5 - Crossed Lines

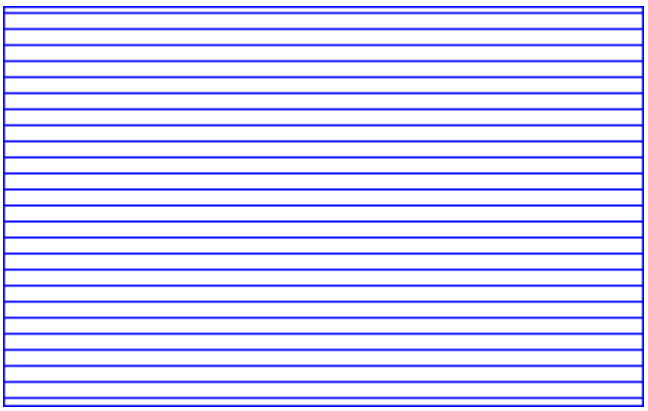
© 2023 Theo Gottwald * https://www.smart-package.com

6 - Diagonal Crossed Lines
***P4 - opt. Color for Pattern,* if omitted Black (&H0) is used.**

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.Fill-Sample
'***********************************
ANA.New|0|320|200
ANA.Fill|0|&HFFFFFF|1|&HFF0000
ANA.Show|0!
ANA.Save|0|?path\Lines01.png
ENR.

' Below you can see the result of the Operation.
```



Remarks

Currently there is no way to change the distance between the lines or the thickness of the lines.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.4 ANA.HGradient

MiniRobotLanguage (MRL)

ANA.HGradient

Generate a horizontal Color Gradient between two Colors in an Image-Register (IR)

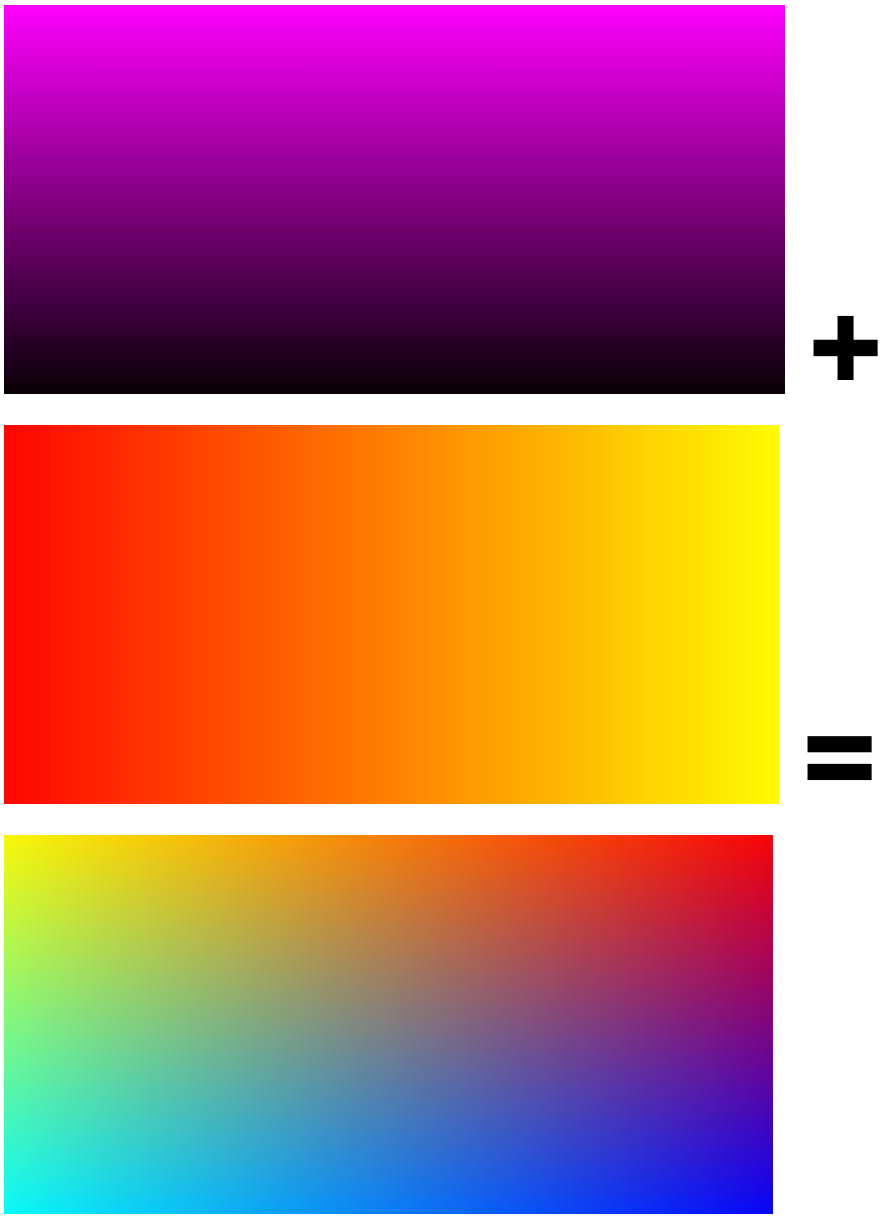
Intention

This command will generate an Gradient in horizontal direction, between two Colors.

1. Generate a H-Gradient and Mix it with a V-Gradient to get a Multi-Gradient
This will generate a V- and a H-Grradient and Mix them together using Blitmode 7 to get a Multi-Gradient.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.Show|0!
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Show|2!
ANA.Mix|0|2|7
ANA.Show|2!
ENR.
```

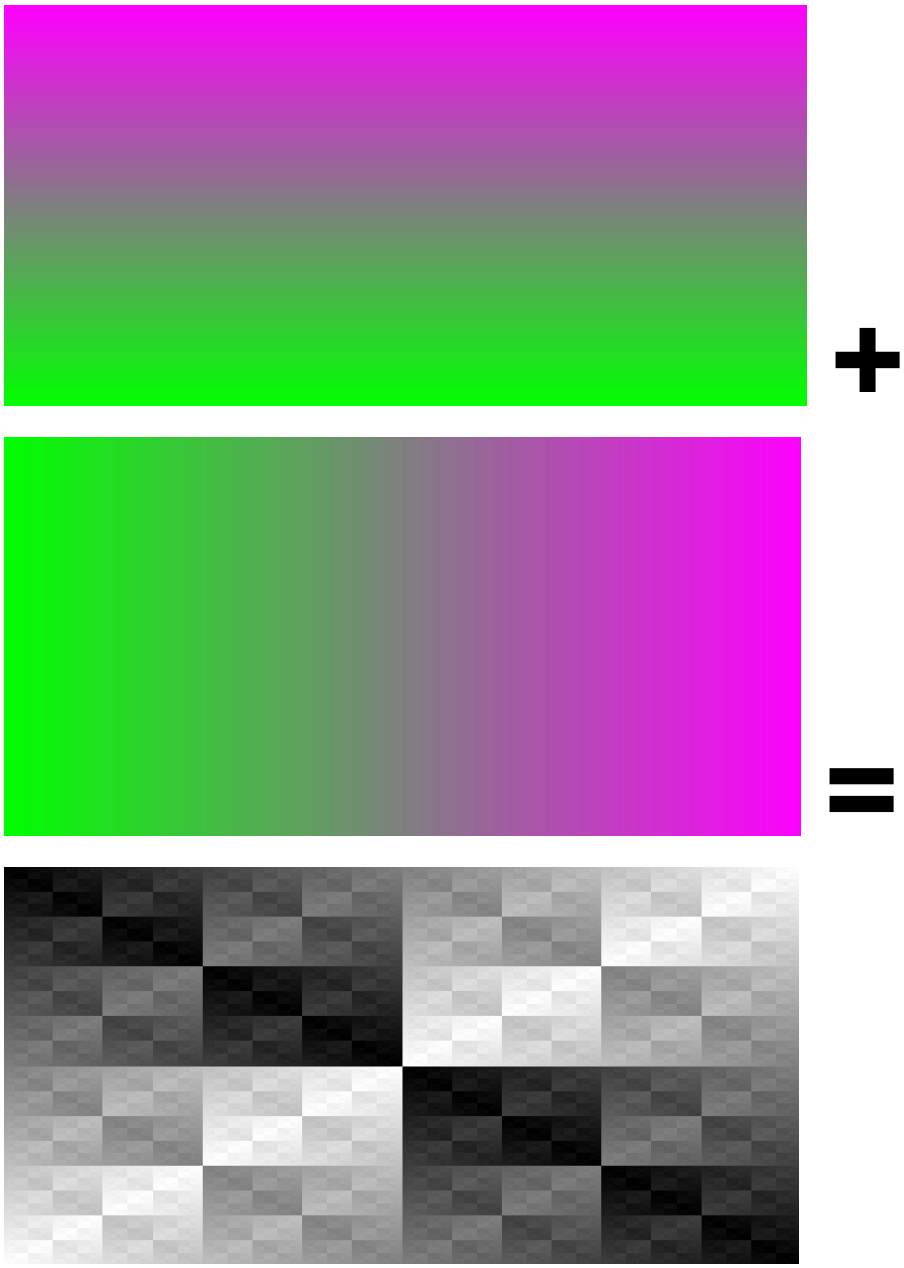

© 2023 Theo Gottwald * https://www.smart-package.com



2. If you omit P3, you will get an automatic selected color
If you om,it P3, the second Color will be automatically calculated using the Formula: &HFFFFFF-**P2**

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF
ANA.Show|0!
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF
ANA.Show|2!
ANA.Mix|0|2|7
ANA.Show|2!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

ANA.HGradient[|P1][|P2][|P3]

Parameter Explanation

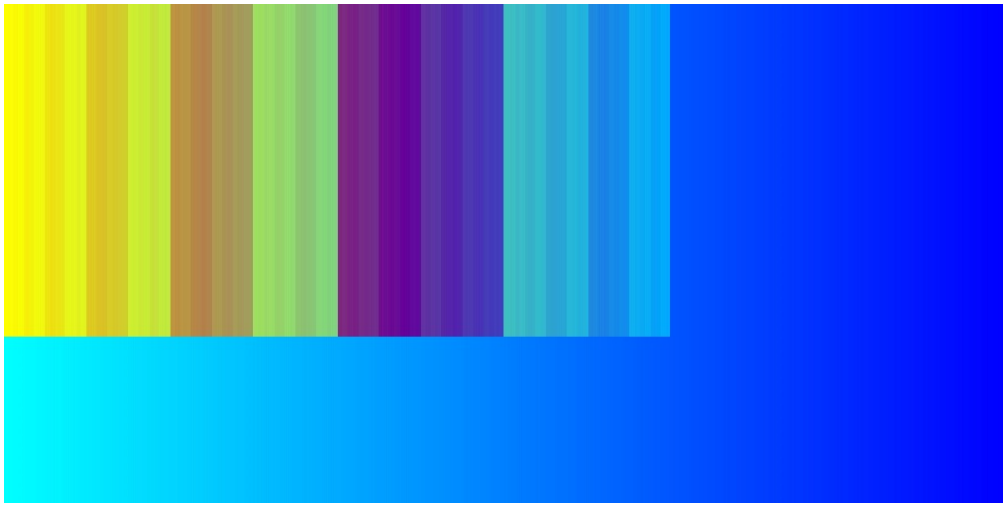
P1 - opt. Target Image-Register number, if omitted, IR 0 will be used. There must be Content in that IR, it must exist.
P2 - opt. Color A (Top Color)
P3 - opt. Color B (Bottom Color)

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' ANA.H/VGradient-Sample
'**********************************
ANA.New|0|1200|600
ANA.hgradient|0|&H0000ff|&H00FFFF
ANA.Show|0!

ANA.New|2|800|400
ANA.hgradient|2|&HFF00FF|&H00FF00
ANA.Show|2!

ANA.Mix|0|2|7
ANA.Show|2!
ENR.

' Here we mix two Gradients of different resolution.
```



Remarks

A H-Gradient on a 6000x4000 large Picture will need 3159658 Ticks, that is on my Computer 0.19 Seconds.

Using the SPR for commercial Pictures?
Yes, is possible.

I have set this Picture for Sale on Dreamstime, see: "Color Whee Background".

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.5 ANA.PatternFill

MiniRobotLanguage (MRL)

ANA.PatternFill

Copy multiple equal Patterns into an Image-Register (IR)

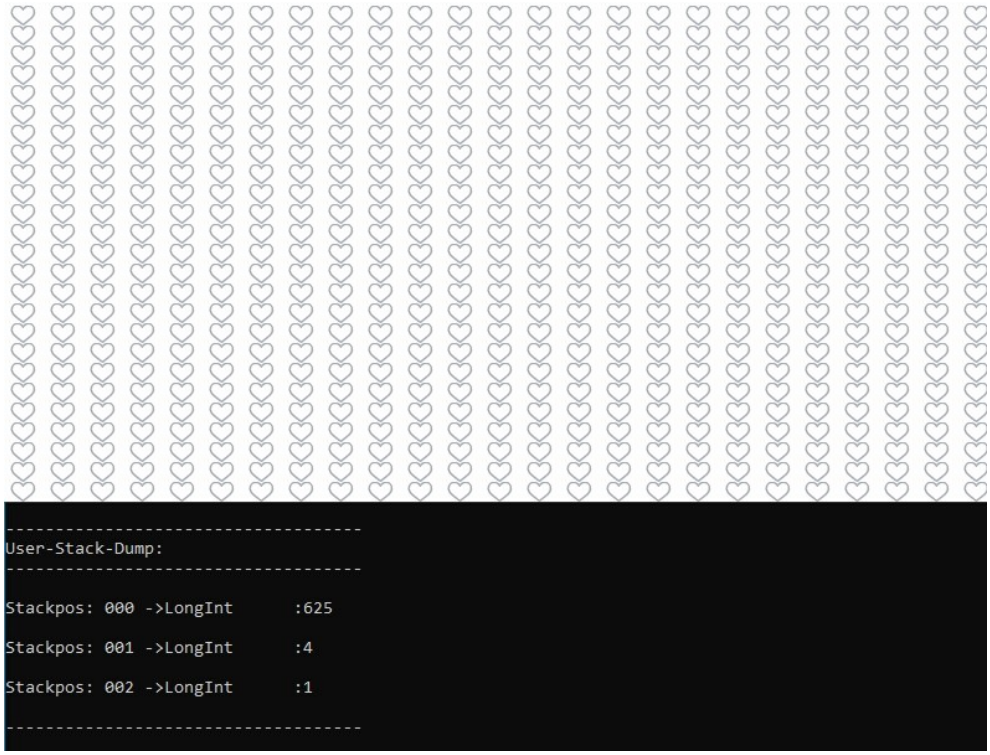
Intention

This command will fill an IR with the Pattern from a smaller IR.

1. Fill an IR with a Pattern
You need to have an IR already, otherwise create one using the ANA.New Command.
Then Load the Pattern and start the Filling.

```
ANA.Load|0|Heart.bmp
ANA.New|1|600|400
ANA.paf|0|1|&HFFFFFF
ANA.Show|1!
ENR.
```

Using this Pattern, we get the result below.
On the TOS you will find the number of copied Patterns, in this case its 625.



© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.PatternFill[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

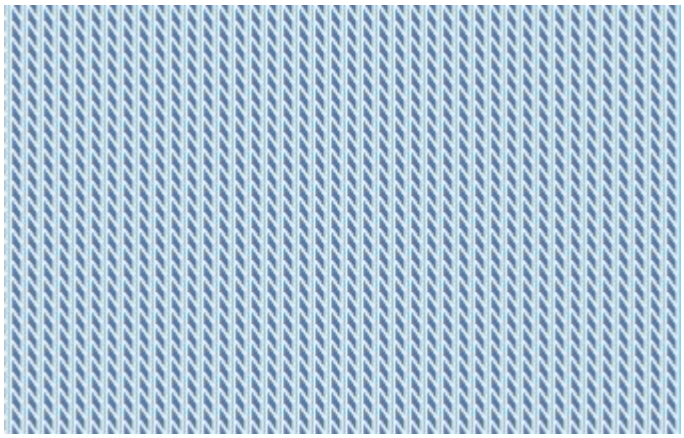
P1 - opt. Source Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Destination Image-Register number, if omitted, IR 0 will be used.
P3 - opt. Color for Background
P4 - opt. X-Shift in Pixel, if omitted 0 is used
P5 - opt. Y-Shift in Pixel, if omitted 0 is used

The Command will leave a 0 on the TOS in case of error, otherwise you get the number of copied patterns.

Example

```
'**********************************
' ANA.PatternFillHwn-Sample
'**********************************
ANA.Load|0|Heart.bmp
ANA.New|1|6000|4000
ANA.paf|0|1|&HFFFFFF
ANA.Show|1!
DMP.1
DMP.6
MBX.!
ENR.


' Here you can see a typical operation.
```

Picture 1: This is the recorded Pattern



Remarks

Plotting 93750 Copies of a Pattern on a 8000x6000 IR will take on my Computer 2498738 Ticks. That around 0.1 Seconds.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.6 ANA.PatternFillHwn

MiniRobotLanguage (MRL)

ANA.PatternFillHwn

Copy multiple equal Patterns into an Image-Register (IR) using a Windowhandle as Referenz for the Size of the Target.

Intention

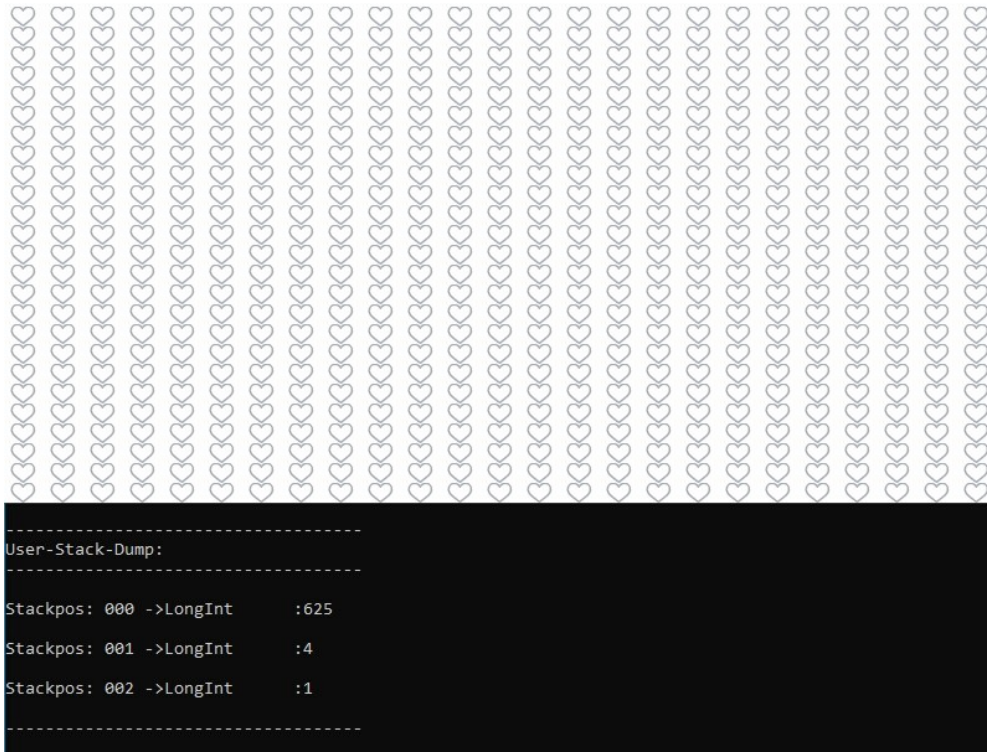
This command will fill an IR with the Pattern from a smaller IR. It will use a Windowhandle in **P4** as a Reference for the Target-Site.
The Target IR will be new created.

1. Fill an IR with a Pattern
You need to have an IR already, otherwise create one using the ANA.New Command.
Then Load the Pattern and start the Filling.

```
STW.ct|Chrome_WidgetWin_1|- Brave
HTV.$$HWN
ANA.Load|0|Heart.bmp
ANA.pafh|0|1|&HFFFFFF|$$HWN
ANA.Show|1!
ENR.
```

Using this Pattern, we get the result below.
On the TOS you will find the number of copied Patterns, in this case its 625.

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

ANA.PatternFillHwn[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - opt. Source Image-Register number, if omitted, IR 0 will be used.
P2 - opt. Destination Image-Register number, if omitted, IR 0 will be used.
P3 - opt. Color for Background
P4 - Window-Handle

The Command will leave a 0 on the TOS in case of error, otherwise you get the number of copied patterns.

Example

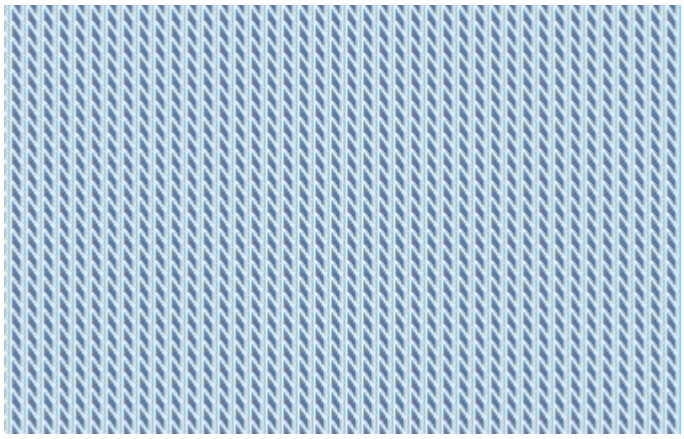
```
'***********************************
' ANA.PatternFillHwn-Sample
' In this Sample we use a recorded 8x8 Pattern, that is filled into an IR
' and the blitted over a Picture. With the right "Shift" it will turn any a fitt
'***********************************
' Load Picture
ANA.Load|1|TP8b.bmp
ANA.Mask|1
ANA.rcd|1|&H0|&H000101

' Load Fullpic to 0
```

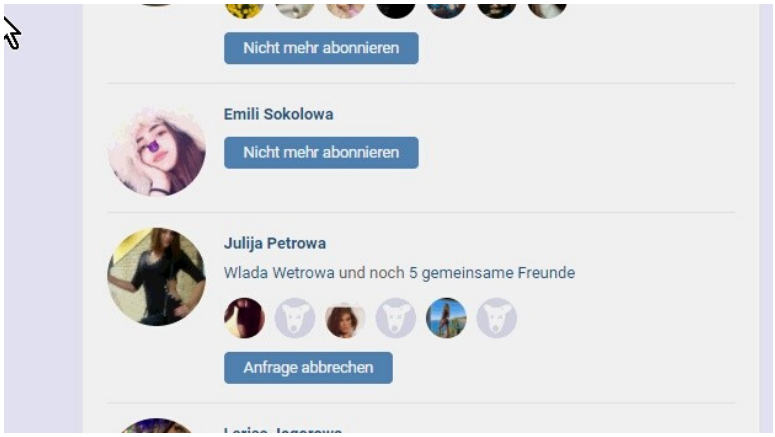

© 2023 Theo Gottwald * https://www.smart-package.com

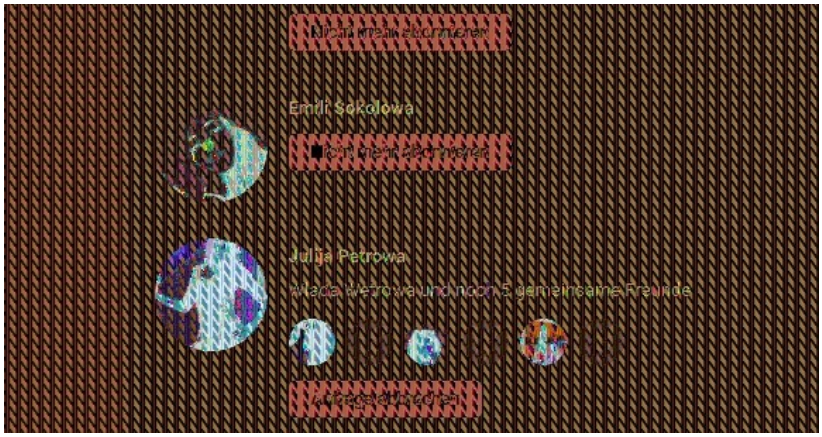
```
ANA.Load|0|Soz_100.png
ANA.Mask|0
ANA.rcd|0|&H0|&H000101

ANA.copy|0|2
ANA.Show|0

STW.ct|Powergraph|debug
HTV.$$MYW
ANA.patternfillHwn|1|0|&HF0F0F0|$$MYW
' 0 ist nun das Multipattern
ANA.copy|2|1

VAN.$$LOY=0
VAN.$$LOX=2
ANA.show|0!
ANA.show|1!
ANA.bsh|0|1|$$LOX|$$LOY|7

ANA.show|1
' Find black pattern
ANA.scc|1|&H0|8|0
MMV.
MBX.!
ENR.

' Here you can see a typical operation.
```

Picture 1: This is the recorded Pattern **Picture 2 This is the Fake-Site where we recorded the pattern** **As a result of the Blit-Operation, the place where the pattern fits turns into a black Square.**

© 2023 Theo Gottwald * https://www.smart-package.com







As a result of the Blit-Operation with Blit-Mode 7, the place where the pattern fits turns black, see the black square in the Middle.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

Plotting 93750 Copies of a Pattern on a 8000x6000 IR will take on my Computer 2498738 Ticks. That around 0.1 Seconds.

Limitations:

While the Aspect-Ration of he window will be equal with the Result, the total size may differ depending on the local settings.
This Command was used during the development, maybe its not of much use in the final version.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.7 ANA.PrintAt

MiniRobotLanguage (MRL)

ANA.PrintAt

Print Text inside of an Image-Register (IR)

Intention

This command will Print a Text inside of an IR.

1. Print a Text into an IR
This will will just Print a Text into an IR.
You can Print Text in all Colors and Sizes. You can also Print Text with an own Background-Color or with transparent Background (use -2 as Background-Color).
Here is a Sample for that.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
' Backgroundcolor
$$LAA=-2
' Fontname
$$LAC=Segoe UI
$$TXT=Relations
FOR.$$LU3|1|2550
  RND.1|799|$$XP1
  RND.1|799|$$XP2
  RND.1|399|$$YP1
  RND.1|399|$$YP2
  RND.-1|1|$$COA
' Textfarbe
  RND.0|&HFFFFFF|$$COL
  ANA.PrintAt|2|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|9|$$LAC|1
NEX.
DMP.1
ANA.Show|2!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

This is the result of the Script above. As we used "-2" as Background, the Text has no own Background-Color.

```
012 0000000000013962 0001 ANA.New|0|800|400
013 0000000000029003 0001 ANA.vgradient|0|&HFF00FF|&H000000
014 0000000000009886 0001 ANA.New|2|800|400
015 0000000000035146 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
016 0000000000043050 0001 ANA.Mix|0|2|7
017 0000000000000011 0001
018 0000000000000003 0001
019 0000000000000135 0001 VAR.$$LAA=-2
020 0000000000000003 0001
021 0000000000000046 0001 VAR.$$LAC=Segoe UI
022 0000000000000003 0001
023 0000000000000039 0001 VAR.$$TXT=Relations
024 0000000000000003 0001
025 0000000000000133 2551 FOR.$$LU3|1|2550
026 0000000000000104 2550 RND.1|799|$$XP1
027 0000000000000089 2550 RND.1|799|$$XP2
028 0000000000000088 2550 RND.1|399|$$YP1
029 0000000000000088 2550 RND.1|399|$$YP2
030 0000000000000086 2550 RND.-1|1|$$COA
031 0000000000000004 2550
032 0000000000000092 2550 RND.0|&HFFFFFF|$$COL
033 0000000000020004 2550 ANA.PrintAt|2|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|9|$$LAC|1
034 0000000000000006 2551 NEX.
```

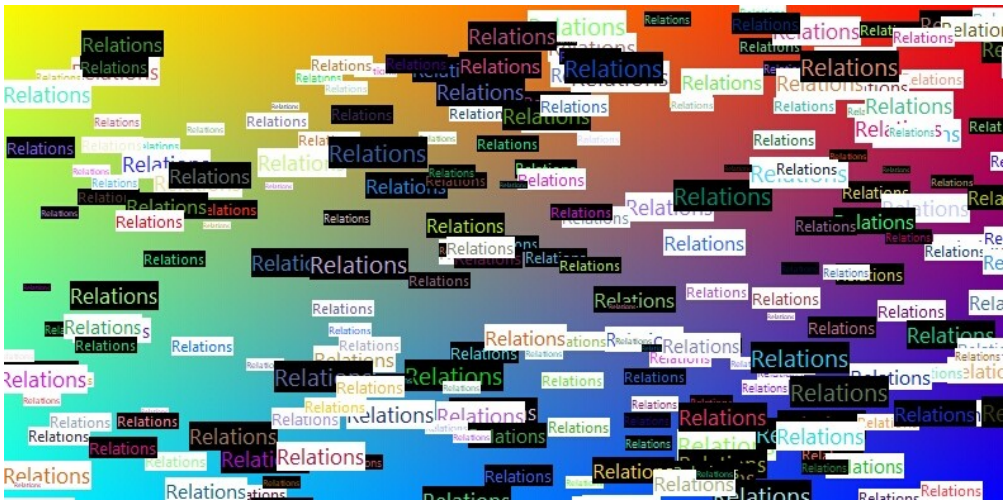
Here you can see the Speed-Dump of the Script in "FreeRun"-Mode from the Editor.

We can also dynamically change the size of the used Font:

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
' Fontname
$$LAC=Segoe UI
$$TXT=Relations
FOR.$$LU3|1|255
  RND.1|799|$$XP1
  RND.1|799|$$XP2
  RND.1|399|$$YP1
  RND.1|399|$$YP2
  RND.-1|1|$$COA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Textfarbe
  RND.0|&HFFFFFF|$$COL
' Hintergrundfarbe
  RND.0|1|$$SWI
  IVV.$$SWI=0
    $$LAA=0
  ELS.
    $$LAA=&HFFFFFF
  EIF.
  RND.4|14|$$SIZ
  ANA.PrintAt|2|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|$$SIZ|$$LAC|1
NEX.
ANA.Show|2!
ENR.
```



Syntax

ANA.PrintAt|P1|P2[|P3][|P4][|P5][|P6][|P7][|P8][|P9][|P10]

Parameter Explanation

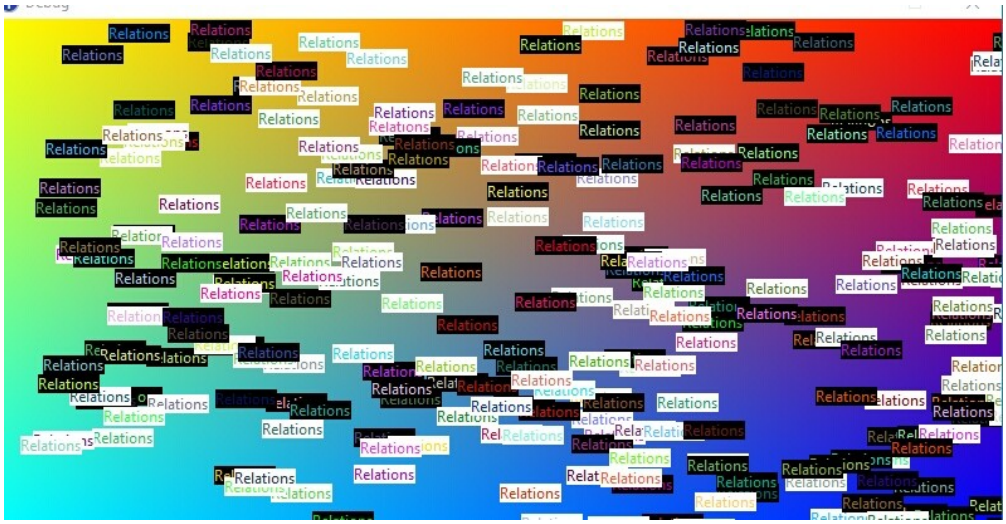
P1 - Image-Register number
P2 - Text to Print
P3 - opt. Text-Color
P4 - opt. Background-Color
P5 - opt. X-Pos for Text
P6 - opt. Y-Pos. for Text
P7 - opt. FontSize for Text
P8 - opt. FontName for Text
P9 - opt. FontSmoothing 0/1
P10 - opt. Use Cleartype 0/1

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
' ANA.PrintAt
'**********************************
ANA.New|0|100|64
' Hintergrundfarbe
$$LAA=&H585858
ANA.Fill|0|$$LAA
' Fontname
$$LAC=Segoe UI
' Textfarbe
$$LAB=&HFFFFFF
$$TXT=Relations
ANA.PrintAt|0|$$TXT|$$LAB|$$LAA|0|0|9|$$LAC|1
ANA.Show|0#!
ENR.

'**********************************
' ANA.PrintAt - Sample 2
'**********************************
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
' Fontname
$$LAC=Segoe UI

$$TXT=Relations

FOR.$$LU3|1|255
  RND.1|799|$$XP1
  RND.1|799|$$XP2
  RND.1|399|$$YP1
  RND.1|399|$$YP2
  RND.-1|1|$$COA
' Textfarbe
  RND.0|&HFFFFFF|$$COL
' Hintergrundfarbe
  RND.0|1|$$SWI
  IVV.$$SWI=0
    $$LAA=0
  ELS.
    $$LAA=&HFFFFFF
  EIF.
  ANA.PrintAt|2|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|9|$$LAC|1
NEX.
ANA.Show|2!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

This Command may use up to 20000 Ticks per call. This means it can be called about 800 times in a second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.12.8 ANA.VGradient

MiniRobotLanguage (MRL)

ANA.VGradient

Generate a Vertical Gradient between two Colors in an Image-Register (IR)

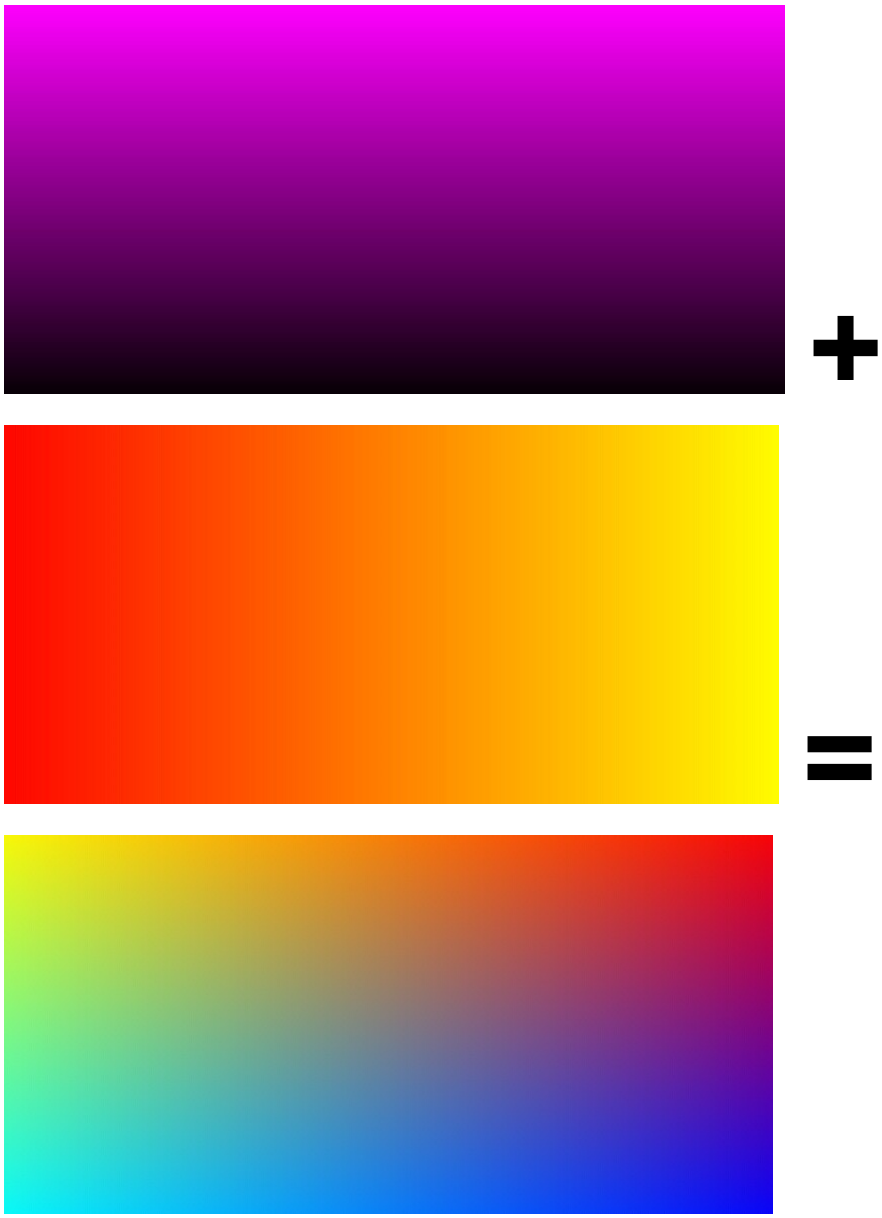
Intention

This command will generate an Gradient in vertical direction, between two Colors.

1. Generate a V-Gradient and Mix it with a H-Gradient to get a Multi-Gradient
This will generate a V- and a H- Grradient and Mix them together using Blitmode 7 to get a Multi-Gradient.

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.Show|0!
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Show|2!
ANA.Mix|0|2|7
ANA.Show|2!
ENR.
```

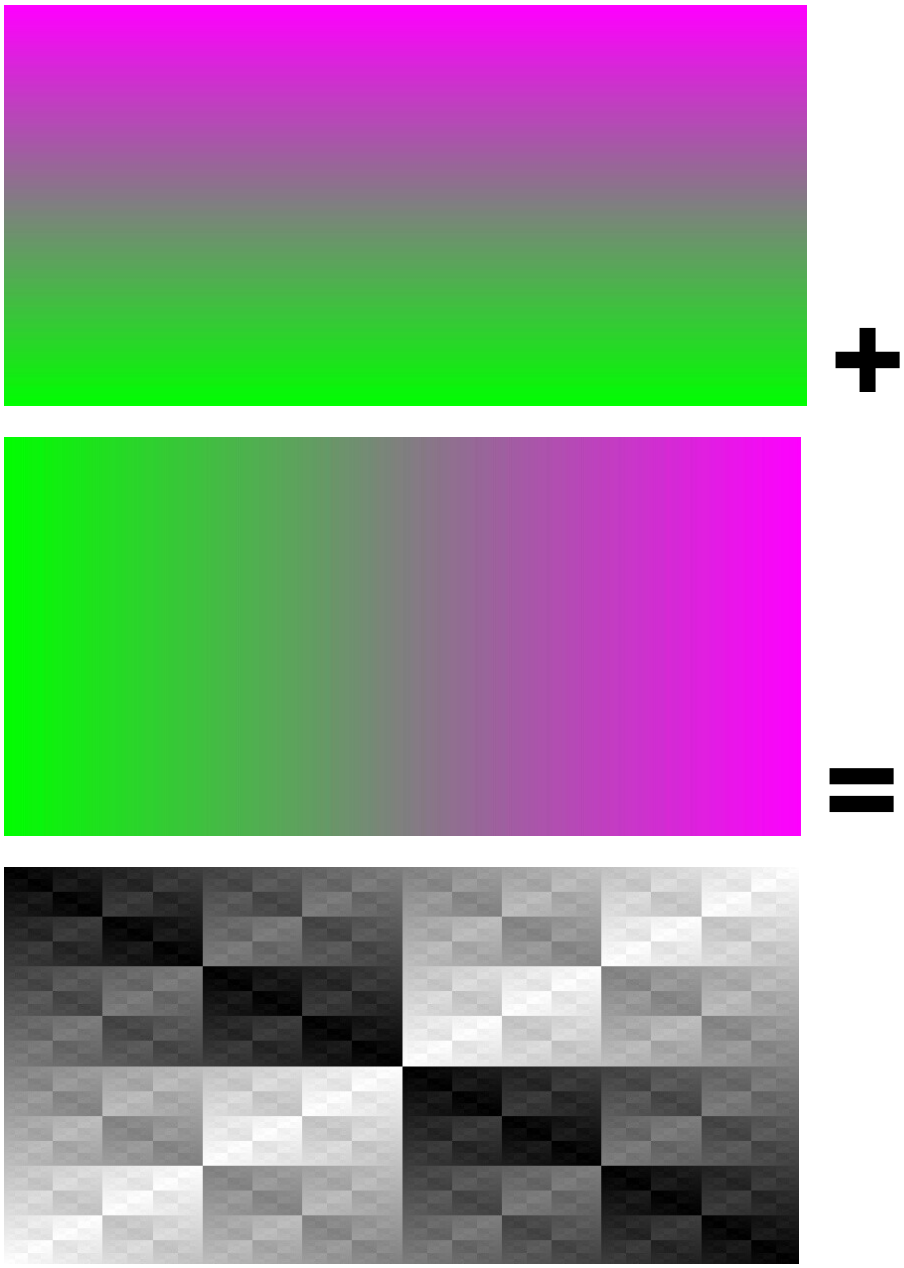

© 2023 Theo Gottwald * https://www.smart-package.com



2. If you omit P3, you will get an automatic selected color
If you om,it P3, the second Color will be automatically calculated using the Formula: &HFFFFFF-**P2**

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF
ANA.Show|0!
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF
ANA.Show|2!
ANA.Mix|0|2|7
ANA.Show|2!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

ANA.VGradient[|P1][|P2][|P3]

Parameter Explanation

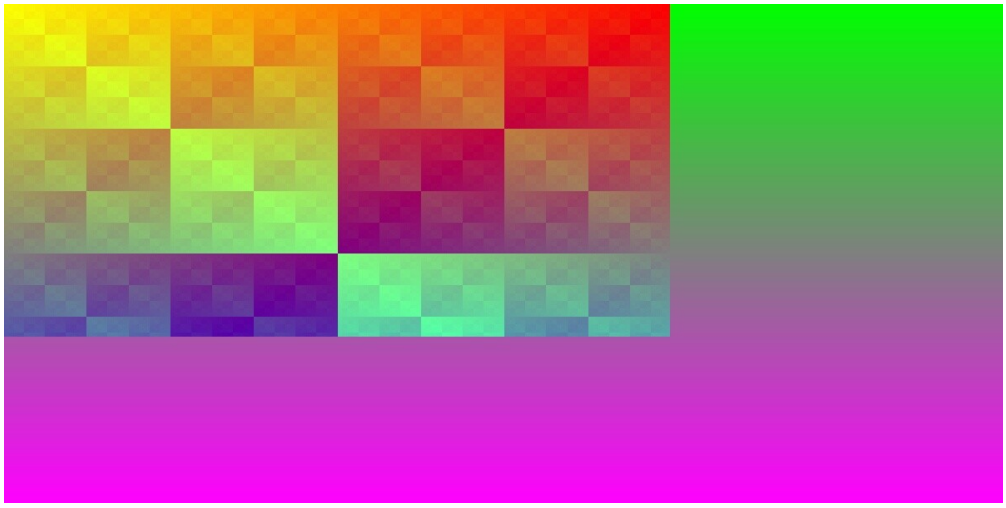
P1 - opt. Target Image-Register number, if omitted, IR 0 will be used. There must be Content in that IR, it must exist.
P2 - opt. Color A (Top Color)
P3 - opt. Color B (Bottom Color)

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' ANA.H/VGradient-Sample
'**********************************
ANA.New|0|1200|600
ANA.vgradient|0|&HFF00FF|&H00FF00
ANA.Show|0!

ANA.New|2|800|400
ANA.hgradient|2|&H0000ff|&H00FFFF
ANA.Show|2!

ANA.Mix|0|2|7
ANA.Show|2!
ENR.

' Here we mix two Gradients of different resolution.
```



Remarks

A V-Gradient on a 6000x4000 large Picture will need 1429447 Ticks, that is on my Computer 0.89 Seconds.

Using the SPR for commercial Pictures?
Yes, is possible.

I have set this Picture for Sale on Dreamstime, see: "Color Whee Background".

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

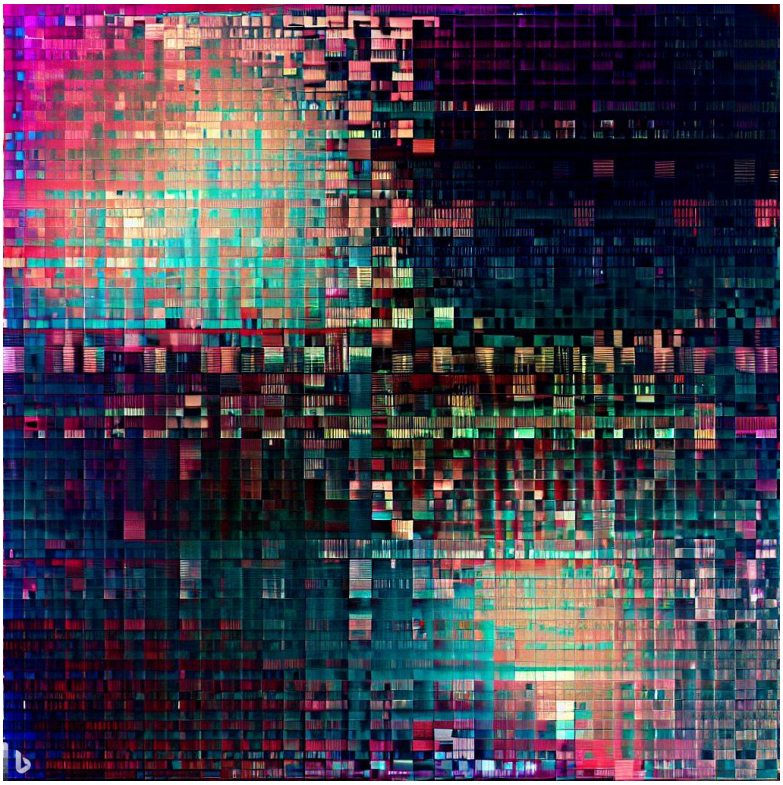
See also:

•

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13 Pattern-Recognition

This Chapter contains the new Pattern-Recognition Engine Commands.



Get more detailed Information about the usable Pattern-Sizes and the Debug-Flag, here:

Pattern-Search Details 1862

Currently "Undocumented Commands":

```
ANA.FindPattern5|P1|P2|P3|P4|P5|P6|P7|P8|P9|P10|P11
```

This command will search for 5 different Patterns in one Run. These could be different Sizes of something.

Target IR is always IR 0
First Pattern-IR is always IR 1 alle further Patterns must be in the IR's 2 to 5

P1 + P2 - Color-Tolerance and PixelHits Pattern 1
P3 + P4 - Color-Tolerance and PixelHits Pattern 2
P5 + P6 - Color-Tolerance and PixelHits Pattern 3
P7+ P8 - Color-Tolerance and PixelHits Pattern 4
P9 + P10 - Color-Tolerance and PixelHits Pattern 5
P11 - Debug Flag

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.1 ANA.CompareBlock

MiniRobotLanguage (MRL)

ANA.CompareBlock

Compares a rectangular area in an Image-Register (IR) with a given Pattern. The result includes the number of pixel hits and a 0/1 value indicating whether the pattern matches at that position.

Intention

The ANA.CompareBlock command allows you to compare a rectangular area within an Image-Register (IR) with a specified Pattern.
It provides information on the number of pixel hits and whether the pattern matches at the given position.
This command is useful when you need to assess the similarity or presence of a pattern within a specific area of an image.
This command will compare a rectangular Area in an IR with the given Pattern.
The result will be the number of Pixel-Hits and 0/1 if the Pattern will match at that place, on TOS.

For more Details please see the
ANA.FindPattern Command 1537

Syntax

ANA.CompareBlock|P1|P2[|P3][|P4][|P5][|P6]

Parameter Explanation

P1 – Target Image-Register (IR): The Image-Register containing the target area for comparison.
P2 – Pattern Image-Register (IR): The Image-Register containing the pattern to be compared.

P3 – X-Position in Target: The X-coordinate of the upper-left corner of the target area.
P4 – Y-Position in Target: The Y-coordinate of the upper-left corner of the target area.
P5 – Color Tolerance: The allowed difference in color values between the target and pattern pixels.
P6 – Pixel-Hits Needed: The minimum number of pixel hits required for the pattern to be considered a match.

About the P6 - Parameter ("Pixel-Hits"):

© 2023 Theo Gottwald * https://www.smart-package.com

The maximum number of matching pixels in your pattern is determined by its size. For example, an 8x8 pattern can have up to 64 matching pixels, whereas a 24x8 pattern can have up to 192.
P6 is based on the total number of pixels in the pattern.
If you set it to 63, for instance, it means that at least 63 pixels must match the pattern, including the tolerance.
If you input a negative number for **P6**, such as "-15", it's treated as a percentage of the total pixel count, translating to "15% of the maximum pixels of that pattern".
If you leave it empty or set it to 0, the default value will be 85% of the pixel count in the pattern.
P4 denotes the minimum number of matches needed to accept a result.
Be careful not to set **P6** too low, as it could potentially overwhelm the system with a massive number of results.

This command has some special features regarding the "**Pixel-Hit Tolerance Parameters**".

1. If you specify "0", the command will calculate the maximum possible number of hits for the loaded pattern and take 85% of that value.
2. If you specify a negative value like "-69", the command will also calculate the maximum possible number of hits for the loaded pattern and take that as 100%. In this example, it would be 69%, so if this is a 12x12 pattern and therefore has a maximum of 144 pixel-hits, 69% of that would be 99 pixels.
3. You can directly specify a positive value that is the exact number of needed pattern-pixels for the pattern to be considered as found.
4. The maximum number of found results is limited by the setting of "MaxResults" using the ANA.SetData command.
5. Due to the use of multiple CPU cores at different places at the same time, there is no guarantee in which order matches are returned.
6. The command will leave a 0/1 on the TOS. As always, a 0 means error, while a 1 means "all fine". For example, if one of the IR's does not exist, then you will get a result "0" on TOS. The program will not generate an error message.

Get the result on TOS

This command will compare a rectangular Area in an IR with the given Pattern.
The result will be the number of Pixel-Hits and 0/1 if the Pattern will match at that place, on TOS.

Example

```
' To compare a rectangular area in IR 0 with the pattern in IR 1 at position (10
' with a color tolerance of 5 and requiring at least 50 pixel hits, use the foll
ANA.CompareBlock|0|1|10|20|5|50|1
```

Remarks

Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.
5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again.

© 2023 Theo Gottwald * https://www.smart-package.com

7. A pattern that is completely empty will not work as there is nothing to recognize in it.

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

- ANA_FindPattern 1537

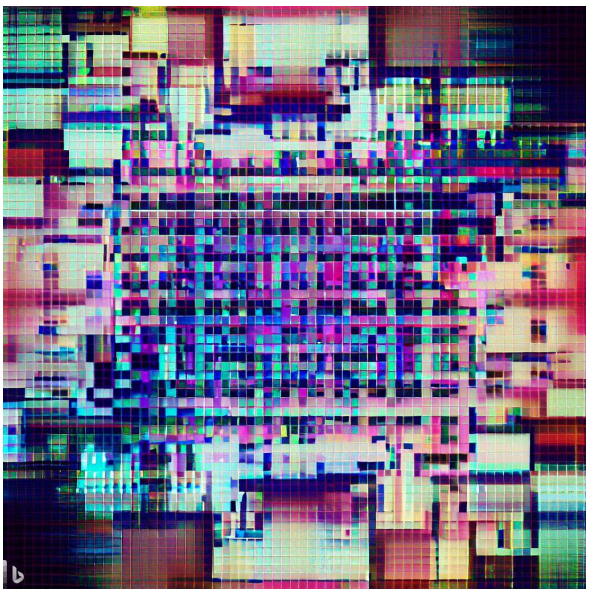
© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.2 ANA.DownScale

MiniRobotLanguage (MRL)

ANA.DownScale

DownScale a 12x12 Pattern (inside an IR) to smaller Sizes



Intention

If you have a 12x12 pattern and you want it to be recognized at different zoom settings, the best approach is to record the pattern in its native resolution.

As an alternative, you can use the ANA.DownScale command to automatically generate smaller patterns from the original 12x12 pattern. This method works best with simple symbols and patterns that do not use Cleartype.

Assume your browser setting is at 100%. We will record a red cross as a pattern and then generate a downscaled version.

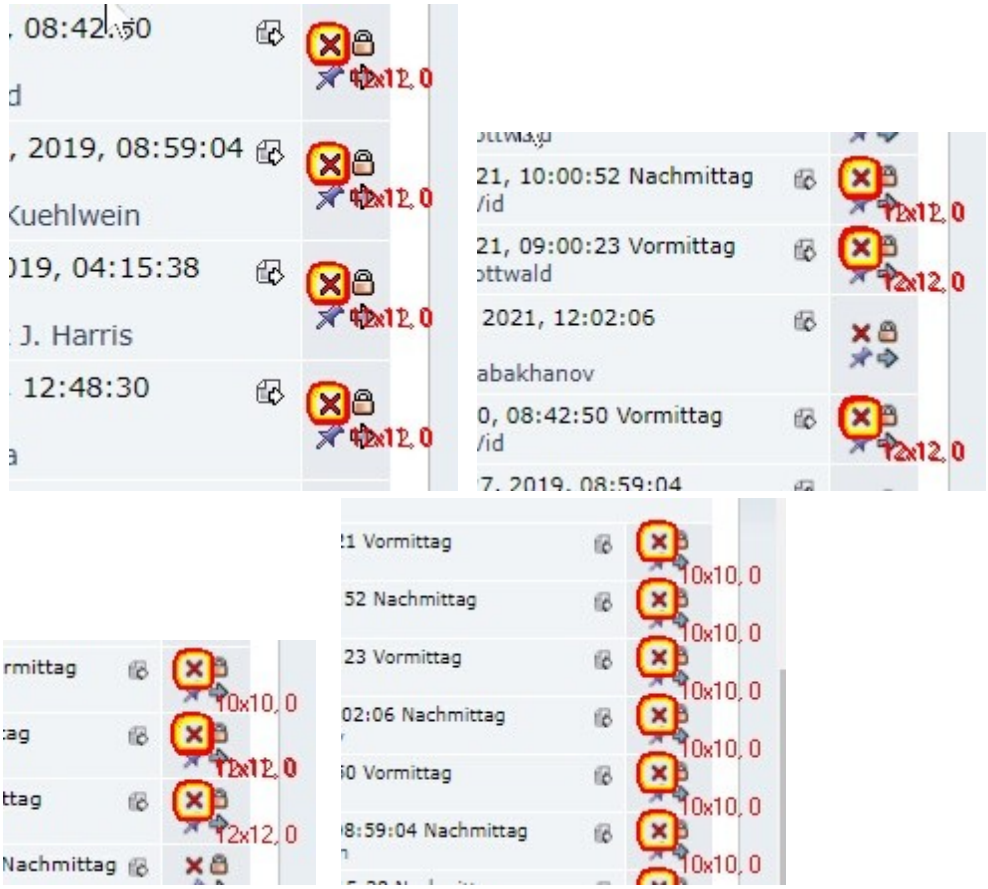
You can use the "`ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH`" command to search for multiple patterns at once. The ANA.ShowHits command can be used to identify which patterns have been detected.
Due to the way fonts are rendered in Windows (using "Cleartype"), downscaled versions may not work well with text.
They are more effective when used with symbols.

If the downscaled versions are ineffective, simply record the pattern at the desired zoom level using the HE-Recorder. If you record using the 12x12 pattern size, the HE-Recorder will automatically call the "Downscale" command and save multiple sizes. However, it's generally more effective to record patterns at their native resolution. Even with a high tolerance setting, the success rate with downscaled patterns may not be perfect.

© 2023 Theo Gottwald * https://www.smart-package.com

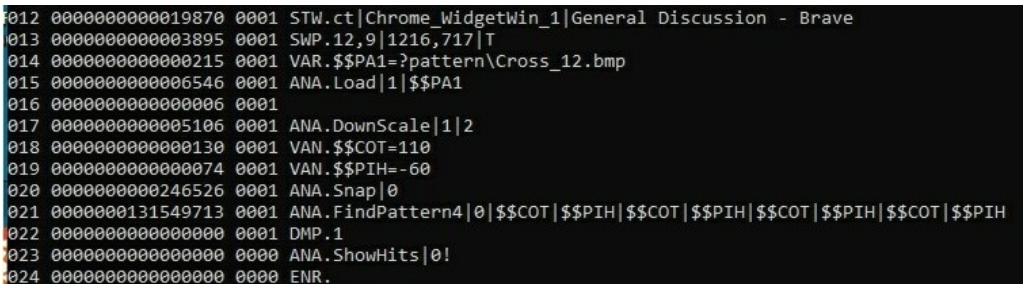
The images below show browser symbols at 100% and 75% zoom, detected using a downscaled pattern with high tolerances.

```
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
VAR.$$PA1=?pattern\Cross_12.bmp
ANA.Load|1|$$PA1
' This will generate the 4 Downscaled Versions
ANA.DownScale|1|2
VAN.$$COT=110
VAN.$$PIH=-60
ANA.Snap|0
ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
DMP.1
ANA.ShowHits|0!
ENR.
```



The first Picture shows the Finds at a Browser Zoom of 100%. The following Pictures show the findings at a Browser Zoom of 80%, 75%, and 67% using the downscaled Patterns.

© 2023 Theo Gottwald * https://www.smart-package.com



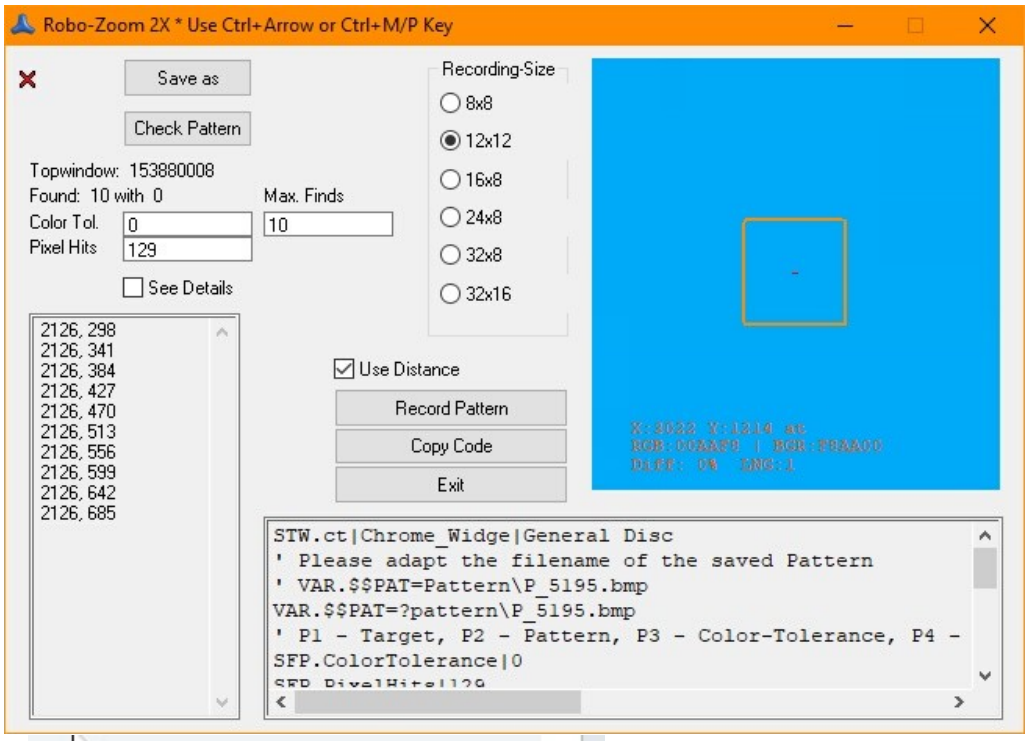
The first Picture shows the Speed-Dump.

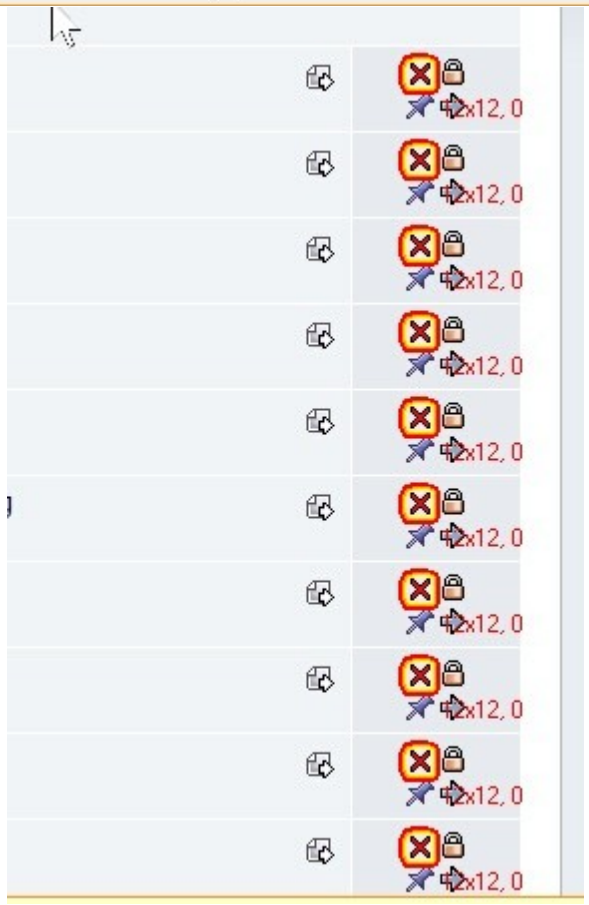
Here is another Sample-Script, in case you have the Patterns already in the ?Pattern\ Folder..

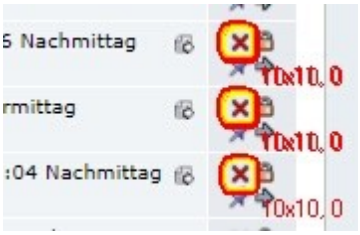
```
' If you alread have the downscaled Variants
' maybe from the HE-Recorder
' Then you can just load them like this:
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
VAR.$$PA1=?pattern\Cross_12.bmp
FOR.$$LR3|1|4
  CAL.$$NUM=$$LR3+8
  VAR.$$PA1=?pattern\Cross_$$NUM.bmp
  ANA.Load|$$LR3|$$PA1
  DBP.Loading: $$PA1
NEX.
VAN.$$COT=110
VAN.$$PIH=-60
ANA.Snap|0
ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
DMP.6
ANA.ShowHits|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com







If you save the Pattern with the HE-Recorder, you will automatically get more versions:

*© 2023 Theo Gottwald * https://www.smart-package.com*

This is internally the result of "DownScale" using the HE-Recorder with a Pattern-Size of 12x12.

Syntax

ANA.DownScale[|P1][|P2][|P3][!][!]

Parameter Explanation

P1 - Source-Pattern IR (Pattern to Scale)
P2 - First Target IR, other Targets are successive IR's -> P2+X...
P3 - Number of downscaled Variants to produce

Example

```
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
VAR.$$PA1=?pattern\Cross_12.bmp
ANA.Load|1|$$PA1
' This will generate the 4 Downscaled Versions
ANA.DownScale|1|2
VAN.$$COT=110
VAN.$$PIH=-60
ANA.Snap|0
ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
DMP.1
ANA.ShowHits|0!
ENR.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.3 ANA.FindPattern

MiniRobotLanguage (MRL)

ANA.FindPattern

Get Locations of a specified Pattern-Object in an Image-Register (IR)

Intention

The "FindPattern" command is designed to identify the positions of a specific object within an image. This command utilizes a four-level Fuzzy-Logic algorithm. Users can modify the color tolerance and specify the minimum number of "pixel hits" necessary for a match.
To streamline the process of pattern recording, we offer a tool called the
Human Eye-Recorder 1140.
This tool simplifies and accelerates the recording of patterns. However, please consider the following when using the pattern recording feature:

1. Mouseover Effects: If the recorded pattern includes objects that change color when the mouse hovers over them, these color changes might not be visually noticeable. Despite this, the pattern matcher will detect these changes and might avoid matching the pattern if the color tolerance is set to zero. To manage this, we suggest capturing a screenshot of the window and recording the pattern from the screenshot to ensure a consistent color representation.

2. Image Format: It is important to use image formats without compression artifacts when working with the pattern matcher commands. Avoid using JPG or other compressed formats, as they may introduce unknown quality issues. It is best to use uncompressed formats such as BMP or PNG.

© 2023 Theo Gottwald * https://www.smart-package.com

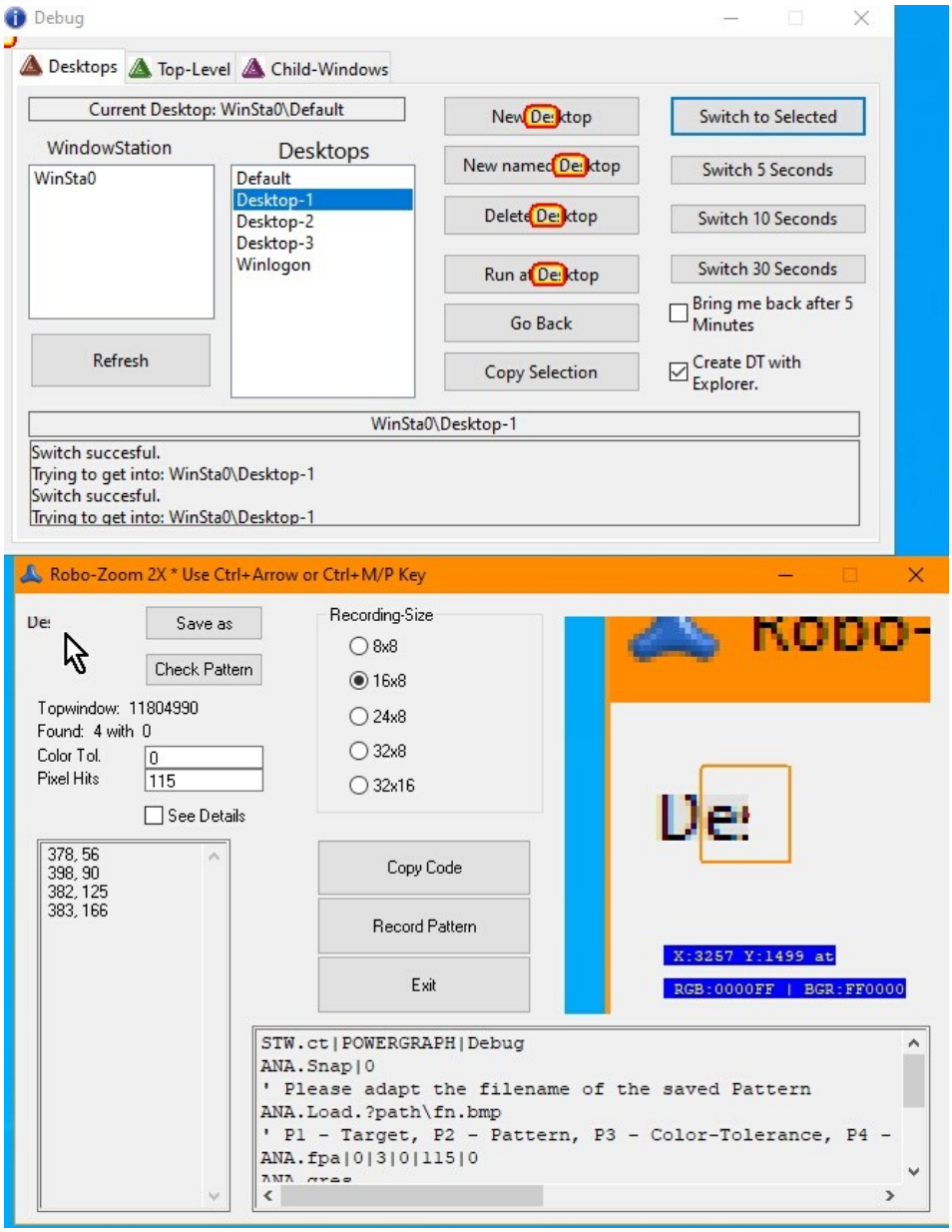
If you face difficulties with pattern matching due to mouseover effects or other factors, consider adjusting the color tolerance or capturing a screenshot for pattern recording. While increasing the color tolerance is not ideal, it might still provide satisfactory results in many cases.

1. Capture Screenshot of Window and use it for Pattern-Search
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|Progman|Program Manager
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This is the Pattern we search for
$$PAA=?path\Help16.bmp
' We load the Pattern into IR 1
ANA.Load|1|$$PAA

' This is the Pattern Searching Command,
' Due to the value "1" as last Parameter, we will get a Debugprint
' of all found Hits. The displayed Point is in the Center of the Pattern.
ANA.fpa|0|1|0|125|1
' This command will put the results of the search on to the Stack.
' Number of found Patterns is on TOS
POP.$$POA
MBX. We got $$POA Results
' We do not need first 3 Found Pattern so we remove them from Stack
STJ.Remo|0|1|2
' This is the Pattern we drive to
POP.$$POA
MMV.t|$$POA
MBX.!
ENR.
```

2. Capture Screenshot of a Window and use it for Patternsearch
This is the normal case. You will only search the Pattern in a Target Window. Using the HE-Recorder-Tool and the "Check Pattern" Button,
you may get a result like this:

© 2023 Theo Gottwald * https://www.smart-package.com



Using the HE-Recorder Program you may get a result like this.

The Code is generated using the HE-Recorder. You may need to change the Path for the Pattern.

```
    The "found hits" will be on the TOS.

STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
'SWP.4090,2|640,400|T
ANA.Snap|0
' Please adapt the filename of the saved Pattern
ANA.Load|1|?path\De16.bmp
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min. Hits (Pixel) maximu
ANA.fpa|0|1|0|115|0
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
MBX.$$NOH
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

IMPORTANT NOTICE about Mouseover Effects:

It is important to note that when recording patterns using the HE-Recorder, you may not visually perceive the "Mouseover Effects" on buttons or objects. However, these effects are visible to the FindPattern command during pattern matching.

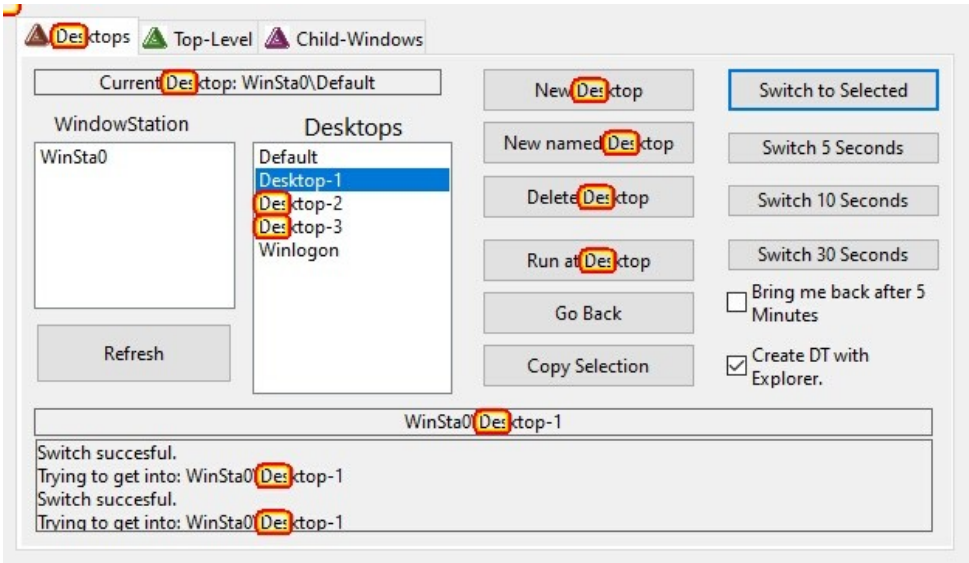
If you are searching for a recorded pattern and do not obtain any matches when using a color tolerance of "0," it indicates that "Mouseover Effects" were present during the recording. To avoid this issue, it is recommended to capture a screenshot of the form or window (preferably in .png or .bmp format to avoid compression artifacts) and record the pattern from the screenshot.

Recording from a Screenshot will prevent "Mouseover Effects".

Recording patterns from a screenshot eliminates the influence of "Mouseover Effects" and ensures accurate pattern matching. You can verify this behavior by using our "Desktops-Manager" tool and recording a button with invisible "Mouseover Effects." Without a screenshot, you will not obtain any matches with a color tolerance of "0." However, by recording from a screenshot, you will immediately obtain the desired matches even with a color tolerance of "0."

Alternatively, you can overcome "Mouseover Effects" by using a higher color tolerance. However, this approach may result in an increased number of unintended matches, so caution should be exercised. For example, using a color tolerance of 50 may yield more hits, including false positives.

© 2023 Theo Gottwald * https://www.smart-package.com



Here i have used a Color-Tolerance of 50. So i got results on grey and on white background.

IMPORTANT NOTICE about missing Administration Rights Effects:

If you're initiating the script from an SPR-Editor without administrative rights, you might face challenges in loading images from the standard robot path.
This path is located under C:\Program Files (x86), a directory that generally requires administrative access to load files. This could lead to black images when using the ANA.Load command.
To circumvent this issue, try executing the script with administrative privileges. This should provide the necessary permissions to correctly load the images.

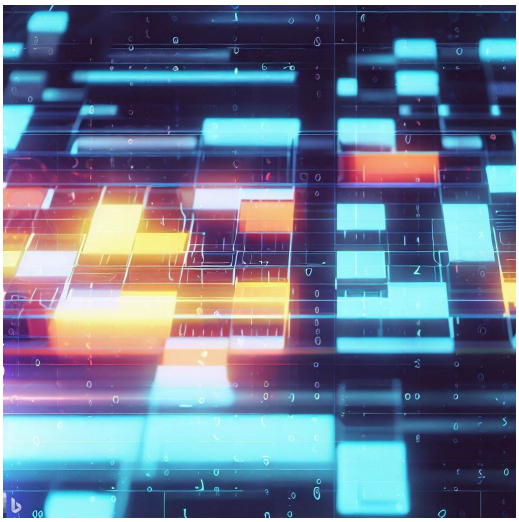
Syntax

ANA.FindPattern[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - opt. Target IR Image-Register number for Target, if omitted, IR 0 will be used.
P2 - opt. Pattern IR Image-Register number for Pattern, if omitted, IR 1 will be used.
P3 - opt. Color Tolerance, if omitted 0 will be used.
P4 - opt. Pixel-Hit Tolerance, if omitted a default value will be used. This value depends on the Pattern-Size that you feed into the Command.
Please see below for a comprehensive explanation of **P4**.
P5 - opt. 0/1 - Debug-Flag see below

© 2023 Theo Gottwald * https://www.smart-package.com



About the P4 - Parameter ("Pixel-Hits"):

The maximum number of matching pixels in your pattern is determined by its size. For example, an 8x8 pattern can have up to 64 matching pixels, whereas a 24x8 pattern can have up to 192.
P4 is based on the total number of pixels in the pattern.
If you set it to 63, for instance, it means that at least 63 pixels must match the pattern, including the tolerance.
If you input a negative number for **P4**, such as "-15", it's treated as a percentage of the total pixel count, translating to "15% of the maximum pixels of that pattern".
If you leave it empty or set it to 0, the default value will be 85% of the pixel count in the pattern.
P4 denotes the minimum number of matches needed to accept a result.
Be careful not to set **P4** too low, as it could potentially overwhelm the system with a massive number of results.

This command has some special features regarding the "**Pixel-Hit Tolerance Parameters**".

1. If you specify "0", the command will calculate the maximum possible number of hits for the loaded pattern and take 85% of that value.
2. If you specify a negative value like "-69", the command will also calculate the maximum possible number of hits for the loaded pattern and take that as 100%. In this example, it would be 69%, so if this is a 12x12 pattern and therefore has a maximum of 144 pixel-hits, 69% of that would be 99 pixels.
3. You can directly specify a positive value that is the exact number of needed pattern-pixels for the pattern to be considered as found.
4. The maximum number of found results is limited by the setting of "MaxResults" using the ANA.SetData command.
5. Due to the use of multiple CPU cores at different places at the same time, there is no guarantee in which order matches are returned.
6. The command will leave a 0/1 on the TOS. As always, a 0 means error, while a 1 means "all fine". For example, if one of the IR's does not exist, then you will get a result "0" on TOS. The program will not generate an error message.

The command leaves a 0/1 on the TOS. Consistently, 0 signifies an error, while 1 indicates that everything is fine.
For instance, if one of the IR's doesn't exist, you'll receive a "0" on the TOS.
The program won't generate an error message in this case.

About the P5 - Parameter ("Debug Flag"):

The Flag-Parameter meanwhile can do a lot more then just help Debugging.

© 2023 Theo Gottwald * https://www.smart-package.com

1 - Enable Debug-Output in Console Window
2 - Snap last used Window again if not found
4 - Use "Difference Map Search"
8 - Disable Filtering (if set)
16 - Use "Invert Pattern"
32 - Use "Complement Pattern"
64 - Use "Extended Search" (takes longer, makes more evaluations)

The results of the Operation can be accessed with the "`ANA.gres`" Command.

Example

Capture Screenshot of complete Desktop and use it for Pattern-Search
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
'**********************************
' ANA.FindPattern-Sample
' This example shows multiple technics
' That are used together with "FindPattern"
' In this case we search over the whole desktop
'**********************************
STW.ct|Progman|Program Manager
' We Snapshot the Desktop into IR 0
ANA.Snap|0
' This is the Pattern we search for
$$PAA=?path\Help16.bmp
' We load the Pattern into IR 1
ANA.Load|1|$$PAA
' This is the Pattern Searching Command,
' Due to the value "1" as last Parameter, we will get a Debugprint
' of all found Hits. The displayed Point is in the Center of the Pattern.
ANA.fpa|0|1|0|125|1
' This command will put the results of the search on to the Stack.
' Number of found Patterns is on TOS
POP.$$POA
MBX. We got $$POA Results
' We do not need first 3 Found Pattern so we remove them from Stack
STJ.Remo|0|1|2
' This is the Pattern we drive to
POP.$$POA
MMV.t|$$POA
MBX.!
ENR.


'**********************************
' ANA.FindPattern-Sample
' This is a real-Life Example-Skript
' It can scroll through a Social-Network and click all
' Found-Buttons that match the Pattern in Picture NoAbo8.bmp
' Then after all are clicked it will Scroll down to find more.
' You need to change the Top-Window STW. to your Browser
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' and you need to record the "Like-Button" ($$PAC)
' or whatever you want to Click.
'*********************************
VAF.$$PAC=?pattern\NoAbo8.bmp
ANA.Load|1|$$PAC
:Lb_Start
STW.ct|Chrome_WidgetWin_1|- Brave
HTV.$$HWN
ANA.Snap|0
GSB.Check
ANA.rsl
ANA.fpa|0|1|0|64|0
POP.$$POB
JIZ.$$POB|Do_ScrollB

FOR.$$LOP|1|$$POB
  POP.$$POA
  DBP.$$POA
  SCW.h|$$HWN
  MLI.t|$$POA
  'PAU.4
NEX.
:Do_ScrollB
GSB.ScrollX
GTO.Lb_Start
'------------------------------------------------------------
:Check
IKS.{SHIFT}
  END.
EIF.
RET.
'------------------------------------------------------------
:ScrollX
STW.ct|Chrome_WidgetWin_1|- Brave
MPO.t|1120,998
MLC.t|1120,998
MWR.-650
RET.
ENR.
'============================================================
```

Remarks

Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.
5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again.
7. A pattern that is completely empty will not work as there is nothing to recognize in it.

© 2023 Theo Gottwald * https://www.smart-package.com

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:
Choosing wrong values for **P3** and **P4** may overrun the robot with too many "fitting results" and may keep the system busy for minutes.
We did not yet implement an "Maximum number of results" at this time.

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.4 ANA.FindPattern2

MiniRobotLanguage (MRL)

ANA.FindPattern2

Locate either one or two specified Pattern-Objects in an Target Image-Register (IR)

Intention

I could have named this Command: "Find Alternative Pattern". It will take two Patterns as Parameter and find either of these.
They can, but they must not have the same size. Each Pattern also has its own values for Color-Tolerance and Minimum Pixel-Hits.

This command will find the Locations where either of these two specified Patterns are is in an Image.
You must load the Target-Picture into IR 0 and load the two Patterns into IR 1 and IR 2.

For more Details please see the
ANA.FindPattern Command 1537

Syntax

ANA.FindPattern2|P1|P2[|P3][|P4][|P5]

Parameter Explanation

Target must be loaded into Image-Register IR 0.
Patterns must be loaded into IR 1 and IR 2
P1 - opt. Color Tolerance, if omitted 0 will be used.
P2 - opt. Pixel-Hit Tolerance, This value depends on the Pattern that you feed into the Command.
If your Pattern is 8x8 then the maximum number of Pixels that could match is 64. If its 24x8 the maximum Number is 192.
If you specify a negative number like "-15" then this is treated as a Percentage of the total Pixel-Number, means "15% of the maximum Pixels of that Pattern"
If omitted or 0 the default value will be 85% of Pixel-Number in Pattern.

P3 - opt. Color Tolerance, if omitted 0 will be used.
P4 - opt. Pixel-Hit Tolerance, This value depends on the Pattern that you feed into the Command.
P5 - opt. 0/1 - Debug-Flag

*© 2023 Theo Gottwald * https://www.smart-package.com*

About the P4 - Parameter ("Pixel-Hits"):

The maximum number of matching pixels in your pattern is determined by its size. For example, an 8x8 pattern can have up to 64 matching pixels, whereas a 24x8 pattern can have up to 192.
P4 is based on the total number of pixels in the pattern.
If you set it to 63, for instance, it means that at least 63 pixels must match the pattern, including the tolerance.
If you input a negative number for **P4**, such as "-15", it's treated as a percentage of the total pixel count, translating to "15% of the maximum pixels of that pattern".
If you leave it empty or set it to 0, the default value will be 85% of the pixel count in the pattern.
P4 denotes the minimum number of matches needed to accept a result.
Be careful not to set **P4** too low, as it could potentially overwhelm the system with a massive number of results.

This command has some special features regarding the "**Pixel-Hit Tolerance Parameters**".

1. If you specify "0", the command will calculate the maximum possible number of hits for the loaded pattern and take 85% of that value.
2. If you specify a negative value like "-69", the command will also calculate the maximum possible number of hits for the loaded pattern and take that as 100%. In this example, it would be 69%, so if this is a 12x12 pattern and therefore has a maximum of 144 pixel-hits, 69% of that would be 99 pixels.
3. You can directly specify a positive value that is the exact number of needed pattern-pixels for the pattern to be considered as found.
4. The maximum number of found results is limited by the setting of "MaxResults" using the ANA.SetData command.
5. Due to the use of multiple CPU cores at different places at the same time, there is no guarantee in which order matches are returned.
6. The command will leave a 0/1 on the TOS. As always, a 0 means error, while a 1 means "all fine". For example, if one of the IR's does not exist, then you will get a result "0" on TOS. The program will not generate an error message.

The command leaves a 0/1 on the TOS. Consistently, 0 signifies an error, while 1 indicates that everything is fine.
For instance, if one of the IR's doesn't exist, you'll receive a "0" on the TOS.
The program won't generate an error message in this case.

About the P5 - Parameter ("Debug Flag"):

The Flag-Parameter meanwhile can do a lot more then just help Debugging.

1 - Enable Debug-Output in Console Window
2 - Snap last used Window again if not found
4 - Use "Difference Map Search"
8 - Disable Filtering (if set)
16 - Use "Invert Pattern"
32 - Use "Complement Pattern"
64 - Use "Extended Search" (takes longer, makes more evaluations)

The results of the Operation can be accessed with the "`ANA.gres`" Command.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

Capture Screenshot of complete Desktop and use it for Pattern-Search
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
'**********************************
' ANA.FindPattern-Sample
'
'**********************************
STW.ct|Progman|Program Manager
' We Snapshot the Desktop into IR 0
ANA.Snap|0

' This are the Pattern we search for
$$PAA=?path\Help12.bmp
$$PAB=?path\Help9.bmp
' We load the Pattern into IR 1+2
ANA.Load|1|$$PAA
ANA.Load|2|$$PAB

' This is the Pattern Searching Command,
' Due to the value "1" as last Parameter, we will get a Debugprint
' of all found Hits. The displayed Point is in the Center of the Pattern.
' Note that we use a different Numbers for the Pixel-Hits because the second Pat
ANA.fpa2|0|125|0|74|1
' This command will put the results of the search on to the Stack.
' Number of found Patterns is on TOS
POP.$$POA
MBX. We got $$POA Results
' We do not need first 3 Found Pattern so we remove them from Stack
STJ.Remo|0|1|2
' This is the Pattern we drive to
POP.$$POA
MMV.t|$$POA
MBX.!
ENR.
```

Remarks
Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.
5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again.
7. A pattern that is completely empty will not work as there is nothing to recognize in it.

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

Choosing wrong values for **P3,P4** and **P5,P6** may overrun the robot with too many "fitting results" and may keep the system busy for minutes.
We did not yet implement an "Maximum number of results" at this time.

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.5 ANA.FindPattern4

MiniRobotLanguage (MRL)

ANA.FindPattern4

Locate any of4 specified Pattern-Objects in an Target Image-Register (IR)

Intention

I could have named this Command: "Find Alternative Pattern". It will take four Patterns as Parameter and find either of these.
They can, but they must not have the same size. Each Pattern also has its own values for Color-Tolerance and Minimum Pixel-Hits.

This command will find the Locations where either of these four specified Patterns are in an Image.
You must load the Target-Picture into IR 0 and load the four Patterns into IR 1, IR 2, IR 3, and IR 4.

For more Details please see the
ANA.FindPattern Command 1537

1. Capture Screenshot of Brave-Browser and use it for Pattern-Search
We will search for 4 Patterns at the same time.

```
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
FOR.$$LR3|1|4
  CAL.$$NUM=$$LR3+8
  VAR.$$PA1=?pattern\Cross_$$NUM.bmp
  ANA.Load|$$LR3|$$PA1
  DBP.Loading: $$PA1
NEX.
VAN.$$COT=110
' Prefixing the Pixelhits with a "-" Sign makes it use the percentage
' Here we use 50% of the total possible Pixel-Hits
VAN.$$PIH=-60
ANA.Snap|0
ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
DMP.6
ANA.ShowHits|0!
ENR.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Important Notice:

Please check the **ANA.FindPattern-Command**[1537] to see details on how to use this command.
There you will find important Information about "Mouseover-Effects" and "Admin-Rights".

Syntax

ANA.FindPattern4[|P1][|P2][|P3][|P4][|P5][|P6][|P7]

Parameter Explanation

P1 - opt. Target IT Image-Register number for Target, if omitted, IR 0 will be used. The Patterns to search for will always be in IR 1 and all following.
P2 - opt. Color Tolerance A, if omitted 0 will be used.
P3 - opt. Pixel-Hit Tolerance A, This value depends on the Pattern that you feed into the Command.

If your Pattern is 8x8 then the maximum number of Pixels that could match is 64. If its 24x8 the maximum Number is 192.

If you specify a negative number like "-85" then this is treated as a Percentage of the total Pixel-Number, means "85% of the maximum Pixels of that Pattern"

© 2023 Theo Gottwald * https://www.smart-package.com

If omitted or 0 the default value will be 85% of Pixel-Number in Pattern.

P4 - opt. Color Tolerance B, see P2.
P5 - opt. Pixel-Hit Tolerance B,see P3.
P6 - opt. Color Tolerance C, see P2.
P7 - opt. Pixel-Hit Tolerance C,see P3.
P8 - opt. Color Tolerance D, see P2.
P9 - opt. Pixel-Hit Tolerance D,see P3.
P10 - opt. 0/1 - Debug-Flag

About the P9 - Parameter ("Pixel-Hits"):

The maximum number of matching pixels in your pattern is determined by its size. For example, an 8x8 pattern can have up to 64 matching pixels, whereas a 24x8 pattern can have up to 192.
P9 is based on the total number of pixels in the pattern.
If you set it to 63, for instance, it means that at least 63 pixels must match the pattern, including the tolerance.
If you input a negative number for **P9**, such as "-15", it's treated as a percentage of the total pixel count, translating to "15% of the maximum pixels of that pattern".
If you leave it empty or set it to 0, the default value will be 85% of the pixel count in the pattern.
P9 denotes the minimum number of matches needed to accept a result.
Be careful not to set **P9** too low, as it could potentially overwhelm the system with a massive number of results.

This command has some special features regarding the "**Pixel-Hit Tolerance Parameters**".

1. If you specify "0", the command will calculate the maximum possible number of hits for the loaded pattern and take 85% of that value.
2. If you specify a negative value like "-69", the command will also calculate the maximum possible number of hits for the loaded pattern and take that as 100%. In this example, it would be 69%, so if this is a 12x12 pattern and therefore has a maximum of 144 pixel-hits, 69% of that would be 99 pixels.
3. You can directly specify a positive value that is the exact number of needed pattern-pixels for the pattern to be considered as found.
4. The maximum number of found results is limited by the setting of "MaxResults" using the ANA.SetData command.
5. Due to the use of multiple CPU cores at different places at the same time, there is no guarantee in which order matches are returned.
6. The command will leave a 0/1 on the TOS. As always, a 0 means error, while a 1 means "all fine". For example, if one of the IR's does not exist, then you will get a result "0" on TOS. The program will not generate an error message.

The command leaves a 0/1 on the TOS. Consistently, 0 signifies an error, while 1 indicates that everything is fine.
For instance, if one of the IR's doesn't exist, you'll receive a "0" on the TOS.
The program won't generate an error message in this case.

About the P10 - Parameter ("Debug Flag"):

The Flag-Parameter meanwhile can do a lot more then just help Debugging.

1 - Enable Debug-Output in Console Window
2 - Snap last used Window again if not found
4 - Use "Difference Map Search"
8 - Disable Filtering (if set)
16 - Use "Invert Pattern"

© 2023 Theo Gottwald * https://www.smart-package.com

32 - Use "Complement Pattern"
64 - Use "Extended Search" (takes longer, makes more evaluations)

The results of the Operation can be accessed with the "`ANA.gres`" Command.

Example

Capture Screenshot of complete Desktop and use it for Pattern-Search
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
'**********************************
' ANA.FindPattern-Sample
'
'**********************************
STW.ct|Progman|Program Manager
' We Snapshot the Desktop into IR 0
ANA.Snap|0

' This are the Pattern we search for
$$PAA=?path\Help12.bmp
$$PAB=?path\Help9.bmp
' We load the Pattern into IR 1+2
ANA.Load|1|$$PAA
ANA.Load|2|$$PAB

' This is the Pattern Searching Command,
' Due to the value "1" as last Parameter, we will get a Debugprint
' of all found Hits. The displayed Point is in the Center of the Pattern.
' Note that we use a different Numbers for the Pixel-Hits because the second Pat
ANA.fpa2|0|1|0|125|0|74|1
' This command will put the results of the search on to the Stack.
' Number of found Patterns is on TOS
POP.$$POA
MBX. We got $$POA Results
' We do not need first 3 Found Pattern so we remove them from Stack
STJ.Remo|0|1|2
' This is the Pattern we drive to
POP.$$POA
MMV.t|$$POA
MBX.!
ENR.
```

Remarks

Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.

© 2023 Theo Gottwald * https://www.smart-package.com

5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again.
7. A pattern that is completely empty will not work as there is nothing to recognize in it.

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:
Choosing wrong values for **P3,P4** and **P5,P6** may overrun the robot with too many "fitting results" and may keep the system busy for minutes.
We did not yet implement an "Maximum number of results" at this time.

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

- ANA_FindPattern 1537

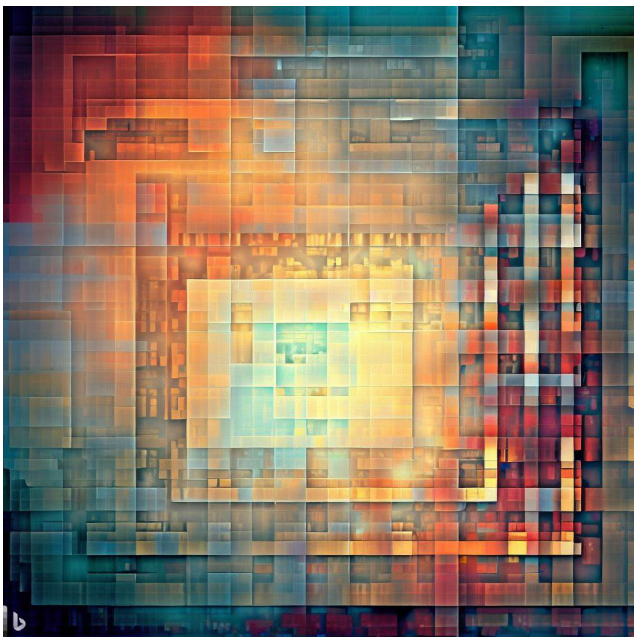
© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.6 ANA.FindPattern5

MiniRobotLanguage (MRL)

ANA.FindPattern5

Locate up to five specified Pattern-Objects in a Target Image-Register (IR)



Using FIndpattern5, you can find 5 sizes of an object to make the script independent of Scaling.

Intention

The ANA.FindPattern5 command is designed to find multiple patterns within a single image. It takes up to five Patterns as parameters and searches for their presence. Each pattern can have different sizes, color tolerances, and minimum pixel hits.

This command will identify the locations where any of the specified patterns are present in the Image.
Load the Target Image into IR 0 and load the patterns into IR 1, IR 2, IR 3, IR 4, and IR 5.

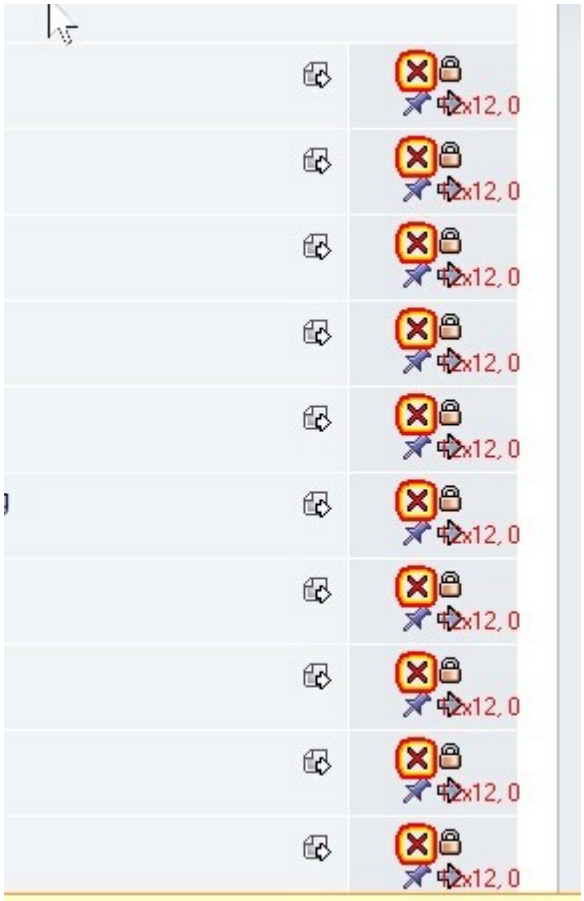
For more Details please see the
ANA.FindPattern Command 1537

1. Capture Screenshot of Brave-Browser and use it for Pattern-Search
We will search for 4 Patterns at the same time.

```
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
FOR.$$LR3|1|5
  CAL.$$NUM=$$LR3+8
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  VAR.$$PA1=?pattern\Cross_$$NUM.bmp
  ANA.Load|$$LR3|$$PA1
  DBP.Loading: $$PA1
NEX.
VAN.$$COT=110
' Prefixing the Pixelhits with a "-" Sign makes it use the percentage
' Here we use 50% of the total possible Pixel-Hits
VAN.$$PIH=-60
ANA.Snap|0
ANA.FindPattern5|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
DMP.6
ANA.ShowHits|0!
ENR.
```



Important Notice:

Please check the **ANA.FindPattern-Command**[1537] to see details on how to use this command.
There you will find important Information about "Mouseover-Effects" and "Admin-Rights".

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.FindPattern5[|P1][|P2][|P3][|P4][|P5][|P6][|P7]

Parameter Explanation

Target IR Image-Register number for Target is always 0.

First Pattern-IR is always IR 1 alle further Patterns must be in the IR's 2 to 5

P1 - opt. Color Tolerance A, if omitted 0 will be used.
P2 - opt. Pixel-Hit Tolerance A, This value depends on the Pattern that you feed into the Command.
If your Pattern is 8x8 then the maximum number of Pixels that could match is 64. If its 24x8 the maximum Number is 192.
If you specify a negative number like "-85" then this is treated as a Percentage of the total Pixel-Number, means "85% of the maximum Pixels of that Pattern"
If omitted or 0 the default value will be 85% of Pixel-Number in Pattern.
P3 - opt. Color Tolerance B, see P2.
P4 - opt. Pixel-Hit Tolerance B,see P3.
P5 - opt. Color Tolerance C, see P2.
P6 - opt. Pixel-Hit Tolerance C,see P3.
P7 - opt. Color Tolerance D, see P2.
P8 - opt. Pixel-Hit Tolerance D,see P3.
P9 - opt. Color Tolerance D, see P2.
P10 - opt. Pixel-Hit Tolerance D,see P3.
P11 - opt. 0/1 - Debug-Flag

About the P2 - Parameter ("Pixel-Hits"):

The maximum number of matching pixels in your pattern is determined by its size. For example, an 8x8 pattern can have up to 64 matching pixels, whereas a 24x8 pattern can have up to 192.
P2 is based on the total number of pixels in the pattern.
If you set it to 63, for instance, it means that at least 63 pixels must match the pattern, including the tolerance.
If you input a negative number for **P2**, such as "-15", it's treated as a percentage of the total pixel count, translating to "15% of the maximum pixels of that pattern".
If you leave it empty or set it to 0, the default value will be 85% of the pixel count in the pattern.
P2 denotes the minimum number of matches needed to accept a result.
Be careful not to set **P2** too low, as it could potentially overwhelm the system with a massive number of results.

This command has some special features regarding the "**Pixel-Hit Tolerance Parameters**".

1. If you specify "0", the command will calculate the maximum possible number of hits for the loaded pattern and take 85% of that value.
2. If you specify a negative value like "-69", the command will also calculate the maximum possible number of hits for the loaded pattern and take that as 100%. In this example, it would be 69%, so if this is a 12x12 pattern and therefore has a maximum of 144 pixel-hits, 69% of that would be 99 pixels.
3. You can directly specify a positive value that is the exact number of needed pattern-pixels for the pattern to be considered as found.

© 2023 Theo Gottwald * https://www.smart-package.com

4. The maximum number of found results is limited by the setting of "MaxResults" using the ANA.SetData command.
5. Due to the use of multiple CPU cores at different places at the same time, there is no guarantee in which order matches are returned.
6. The command will leave a 0/1 on the TOS. As always, a 0 means error, while a 1 means "all fine". For example, if one of the IR's does not exist, then you will get a result "0" on TOS. The program will not generate an error message.

The command leaves a 0/1 on the TOS. Consistently, 0 signifies an error, while 1 indicates that everything is fine.
For instance, if one of the IR's doesn't exist, you'll receive a "0" on the TOS.
The program won't generate an error message in this case.

About the P11 - Parameter ("Debug Flag"):

The Flag-Parameter meanwhile can do a lot more then just help Debugging.

1 - Enable Debug-Output in Console Window
2 - Snap last used Window again if not found
4 - Use "Difference Map Search"
8 - Disable Filtering (if set)
16 - Use "Invert Pattern"
32 - Use "Complement Pattern"
64 - Use "Extended Search" (takes longer, makes more evaluations)

The results of the Operation can be accessed with the "`ANA.gres`" Command.

Example

-

Remarks

Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.
5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again.
7. A pattern that is completely empty will not work as there is nothing to recognize in it.

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

Choosing wrong values for **P1, P2 etc.** may overrun the robot with too many "fitting results" and may keep the system busy for minutes.
We did not yet implement an "Maximum number of results" at this time.

© 2023 Theo Gottwald * https://www.smart-package.com

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.7 ANA.FindPClick

MiniRobotLanguage (MRL)

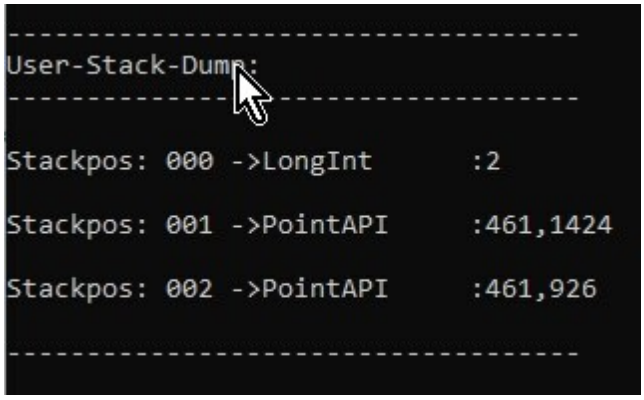
ANA.FindPClick

Get Locations of a specified Pattern-Object and Click one or multiple times

Intention

Welcome to the advanced `FindPClick`-Command.
It makes finding and clicking a Pattern a breeze.
Here's the magic it performs:

- It captures the scene of the active window (Stored in IR 24)
- Your chosen Pattern is loaded, keeping its original size (Secured in IR 23)
- It seeks out the specified Pattern, once or multiple times
- The locations of the Pattern are carefully noted on the TOS
- Clicks are performed at the identified locations, once or multiple times
- After the task is done, all the information you need is neatly arranged on the TOS



The TOS will show: 0 - If NO Pattern were found, otherwise the number of found Patterns. And below that the Coordinates of the found Patterns.

Imagine you're browsing a social network and want to click all the 'LIKE' buttons currently on display.
Just one call to this command, and it's done!

```
' Locate the Top-Window
STW.ct|Chrome_WidgetWin_1| - Brave
' Define the Location for the Pattern
$$FIA=?path\Like.bmp
' Call the Command
ANA.fpc|$$FIA|0|60|0|8|4,16|4|2
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Crucial Reminder:

To fully understand how to use this command, please look into the **ANA.FindPattern-Command**[1537].

We've intentionally omitted details on 'Mouseover-Effects' and 'Admin-Rights' here, so do take the time to read about them there.

Syntax

ANA.FindPClick[|P1][|P2][|P3][|P4][|P5][|P6][|P7][|P8]

Parameter Explanation

P1 - Filepath and Filename of the Pattern-Image-File.

P2 - opt. Color Tolerance, if omitted 0 will be used.

P3 - opt. Pixel-Hit Tolerance, if omitted 33 will be used. This valuie depends on the Pattern that you feed into the Command.

If your Pattern is 8x8 then the maximum number of Pixels that could match is 64. If its 24x8 the maximum Number is 192.

So **P4** is the minimum number of Matches to accept a result. Do not make P4 too low or else you may overrun the system with hundred-thousands of results.

P4 - opt. 0/1 - Debug-Flag

P5 - opt. Click-Type 1 - MLE. 2 - MLI. 4 - MLM. 8 - MLC. 16 - MRM. 32 - MDC.

For **P5** you can add the numbers to make multiple Clicks. For example 8+1=9 means you make an `MLE.` and an `MLC.` Click at each Location.

Please take care that all Clicks here are multiplied with the number in **P7**

P6 - opt. Move Mouse relativ before Click, defaul is "0,0".

P7 - opt. Number of Clicks to do at each found place. (default is 1, maximum is 255)

P8 - opt. How many Findings should be clicked at (default is "0" means that all Findings will be clicked at)

Assume, that you want only to click at the first element to be found, then you will just use a "1" here.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".
For example, if one of the IR's does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

Click Pattern in Browser and Scroll

This will Click a specified Pattern in a Browser-Window and then Scroll forward.

```
'**********************************
' ANA.FindPClick-Sample
' This is a real-Life Example-Skript
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' It can scroll through a Social-Network and click all
' Like-Buttons.
' You need to change the Top-Window STW. to your Browser
' and you need to record the "Like-Button" ($$PAC)
' or whatever you want to Click.
'*********************************
VAF.$$PAC=?path\Nma_16x8.bmp
ANA.Load|1|$$PAC
:Lb_Start
STW.ct|Chrome_WidgetWin_1|- Brave
HTV.$$HWN
ANA.Snap|0
IKS.{SHIFT}
  END.
EIF.
ANA.rsl
$$FIA=?path\Like.bmp
ANA.fpc|$$FIA|0|60|0|8|41,16|1|2
:Do_ScrollB
GSB.ScrollX
GTO.Lb_Start
'-------------------------------------------------------------
:ScrollX
STW.ct|Chrome_WidgetWin_1|- Brave
' You may need to adjust these values, the ROBOT clicks at
' that Point to get the Input Focus into this Window.
MPO.t|1120,998
' To use the Mouse-Wheel later
MLC.t|1120,998
MWR.-650
RET.
```

Remarks

Speed Guidelines:

1. The Command, being 'Multithreaded', loves to work with a Multicore CPUs.
   It has a hearty appetite for CPU power and may even call upon the GPU's resources.
2. It prides itself on speed, even with large windows, given a modern CPU is in action.
3. Remember, larger Patterns mean a slower operation.
4. Current speed secrets?
   Have the pattern touch the upper left corner. But stay flexible, this may not always be the case.
5. If a Pattern is being sluggish, recording a new Pattern and trying again might do the trick.
6. Patterns with nothing in them won't work. They need something to latch onto.
7. Steer clear of '.jpg' images or ones with artefact compression. Stick with '.bmp' or '.png' images for results you can count on.

Limitations:

Choosing wrong values for **P2** and **P3** may overrun the robot with too many "fitting results" and may keep the system busy for minutes.
We did not yet implement an "Maximum number of results" at this time.

© 2023 Theo Gottwald * https://www.smart-package.com

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.8 ANA.Get/Set_Data

MiniRobotLanguage (MRL)

ANA.Get_Data
ANA.Set_Data

Get or Set Internal Values mostly needed for Find-Operation

Intention

Depending on **P1** this command is used to get or set Internal Data-Registers.
These are:

0 - Number of IR where the last ANA.Snap-Command has been taken a Snapshot
1 - Windowhandle of the last ANA.Snap
2 - Snapshot- X-Size
3 - Snapshot- Y-Size
4 - CPU-Cores used for Pattern-Find Operations, this will normally return the number of CPU-Cores in the System.
Setting this number to a value, much higher then the really available Cores, may lead to unforeseable
results including can make Windows too hang due to too many open threads.
5 - Last Operation Result (currently only used by some Operations)
6 - Max-Result Number Setting, this number will limit the number of Finds of all FindPattern-Operations.
7 - Single-Result Flag. Set this Flag to "1" to only get a Single Result with the next Find Operation,
this way you can leave MaxResults untouched.
8 - Default Bordersize for Window Snapshot for the ANA.Snap-Command and FinPattern-Commands that have an internal Snapshot-Command.
The values can be:

```
0 - Clientarea only
1 - Whole Window
2 - Whole Window, No Border
4 - Use Value from Object
```

9 - Default Value for "PixelHits"
10 - Default value for "Color Tolerance"
11 Default Value for Use "Debugmode" (0/1)

Left picture: This is the Stack before and after the call to ANA.GetResult.
Right Picture: This is the Stack after the first Result - that is the number of Found Elements - has been "Pop'ed".

Here is the Code that shows the usage of ANA.GetResult.

```
' Reads the Value of MaxResults.
ANA.Get_Data|6|$$RES
MBX.$$RES
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Change and Read the Value for MaxResults
ANA.Set_Data|6|39
ANA.Get_Data|6|$$RES
MBX.$$RES
ENR.
```

Syntax

ANA.Get_Data[|P1][|P2]

Parameter Explanation

P1 **-** ***opt.*** this Parameter is the number (currently 0 - 11) that specifies the value that should be read.
The complete List is above.

P2 **-** ***opt.*** Variable that will get the result value.If omitted the value is placed on TOS.

ANA.Set_Data[|P1][|P2]

Parameter Explanation

P1 **-** ***opt.*** this Parameter is the number (currently 0 - 11) that specifies the value that should be get/set.
The complete List is above.

P2 **-** ***opt.*** Value to be assign to the internal Variable that is specified by **P1**.

Example

```
'*********************************
' ANA.Get_Data / Set_Data Sample
'*********************************
ANA.Set_Data|6|39
ANA.Get_Data|6|$$RES
MBX.$$RES
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.9 ANA.GetResult

MiniRobotLanguage (MRL)

ANA.GetResult

Get Result from Find-Operation on TOS

Intention

This command is called internally automatically after FindPattern Operations like "ANA.FIndPattern".
It is possibly not needed anymore as from now find-Operations call it internally.

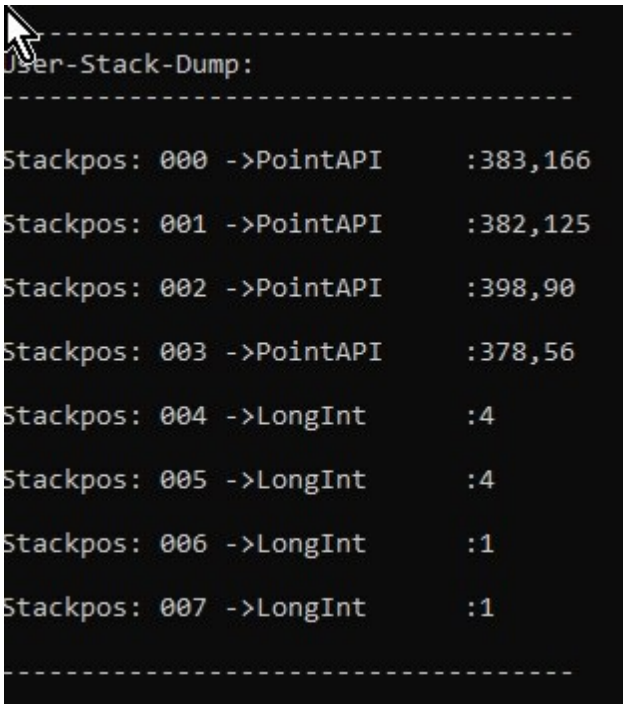
It will place the result of these Operation on the TOS.
It will also clear the internal Find-Operation Memory, therefore you can call it only one time after each Find-Operation.

If you do not need the results from the last Find-Operation, you can use
"ANA.ResetList"
to clear the Find-Memory without dropping the results on the Stack.

© 2023 Theo Gottwald * https://www.smart-package.com

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt      :4

Stackpos: 001 ->LongInt      :1

Stackpos: 002 ->LongInt      :1

------------------------------------


------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt      :4

Stackpos: 001 ->PointAPI     :383,166

Stackpos: 002 ->PointAPI     :382,125

Stackpos: 003 ->PointAPI     :398,90

Stackpos: 004 ->PointAPI     :378,56

Stackpos: 005 ->LongInt      :4

Stackpos: 006 ->LongInt      :4

Stackpos: 007 ->LongInt      :1

Stackpos: 008 ->LongInt      :1

------------------------------------
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Left picture: This is the Stack before and after the call to ANA.GetResult.
Right Picture: This is the Stack after the first Result - that is the number of Found Elements - has been "Pop'ed".

Here is the Code that shows the usage of `ANA.GetResult`.

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
'SWP.4090,2|640,400|T
ANA.Snap|0
' Please adapt the filename of the saved Pattern
ANA.Load|1|?path\De16.bmp
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min. Hits (Pixel) maximu
ANA.fpa|0|1|0|115|0
DMP.6
ANA.gres
DMP.6
' We take away the number of Hits from TOS
POP.$$NOH
DMP.6
' If no Hits, jump away to Lab_NoHits.
MBX.$$NOH
ENR.
```

Syntax

ANA.GetResult[|P1]

Parameter Explanation

P1 - opt. this Parameter is currently not used,

© 2023 Theo Gottwald * https://www.smart-package.com

The Command will leave a 0 on the TOS in case of error.
Otherwise it will put the results of the last Find-Operation on the Stack in sorted Order.

The first number on the TOS is the "Number of results", After that the results come one after the other.
Generally the results are sorted from Bottom to Top.

Example

```
'**********************************
' ANA.GetResult-Sample
'**********************************
' You can not run this example
' its a Subprogram-code that will search for alternative patterns
' if the first pattern is not found.

:GetPat16OR
GSB.GetHWN
ANA.Snap|0
ANA.Load|1|$$_01
ANA.fpa|0|1|0|$$_02|0
ANA.gres
POP.$$POA
IVV.$$POA=0
  ANA.Load|1|$$_03
  ANA.fpa|0|1|0|$$_04|0
  ANA.gres
  POP.$$POA
  IVV.$$POA=0
    MBX.Pattern not found: $$_03
  EIF.
EIF.
RET.
```

Remarks

-

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.10 ANA.GetResultsPack

MiniRobotLanguage (MRL)

ANA.GetResultsPack

Get detailled Result from Find-Operation on TOS

Intention

This command plays a crucial role in obtaining comprehensive results following FindPattern operations, such as '`ANA.FindPattern`'.
It yields more in-depth results than the standard output on TOS, and notably, these results aren't automatically deleted.

The command serves two primary functions:

1. It places a single finding on the stack.
2. When called with **P1**=0, it simply puts the number of finds on TOS.

If the results from the previous Find-Operation are not required,
'`ANA.ResetResultsPack`' can be used to clear the Find-Memory.
The subsequent code demonstrates the usage of this command:
`ANA.GetResultsPack`.

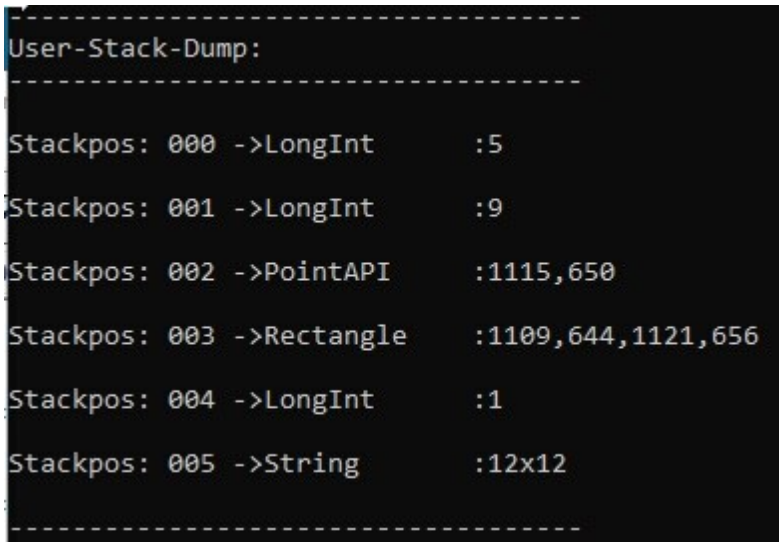
```
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
VAR.$$PA1=?pattern\Cross_12.bmp
ANA.Load|1|$$PA1
' This will generate the 4 Downscaled Versions
ANA.DownScale|1|2
VAN.$$COT=110
VAN.$$PIH=-60
' Setting Maximum stored results to 40 - is 10 by default.
ANA.Set_Data|6|40
ANA.Snap|0
ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
PRT.----------------------------
' This command will clear the stack so we only get the Results-Pack-Item
STS.CLEAR
' Here we ask for the 5th Found Item.
ANA.GetResultsPack|5
DMP.6
ANA.ShowHits|0!
ENR.
```

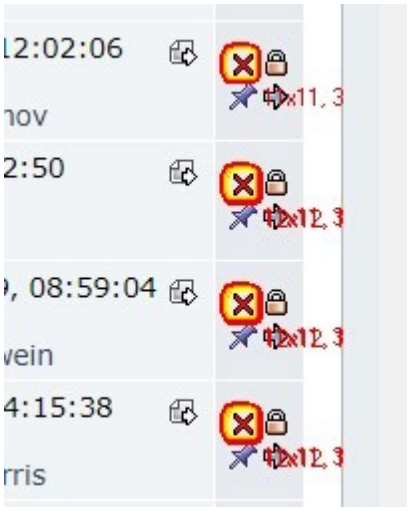
Note that this command will just place the one specified "Find" on the Stack. You will get a lot of additional Information for the find:

`$$000/TOS:` Number of Find that is on Stack.
`$$001` Number of remaining Finds in Memory.

© 2023 Theo Gottwald * https://www.smart-package.com

$$002 Coordinates of Middlepoint of this Find. These are in Top-Window-Coordinates.
$$003 Coordinates of the Find-Rectangle of this Find. These are in Top-Window-Coordinates.
$$004 Engine Number that has found the Pattern - this references the Pattern-Number in "FindPattern4"
$$005 Pattern-Size of the found Pattern.





Left picture: This is the Stack after the call to ANA.GetResultspack|5.

Right Picture: Using Multi-Pattern Find Operations with scaled Patterns and high Tolerance Settings can lead to the same Object being found from several engines.

Syntax

ANA.GetResultsPack[|P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - opt. Number of the Find to get. This number should be in a range from 1 to "Number of Finds".
If **P1** is zero, the command will just return the number of stored Finds.

P2 - opt. 0/1 If **P2** is 1 then the Result that was on TOS is been deleted from Results-Memory.
As the results are organized like an internal Stack, another result will take its number
and the number of remaining results will decrease.

The Command will leave a 0 on the TOS in case of error.
Generally the results are sorted from Bottom to Top.

Example

```
'**********************************
' ANA.GetResult-Sample
'**********************************
STW.ct|Chrome_WidgetWin_1|General Discussion - Brave
SWP.12,9|1216,717|T
VAR.$$PA1=?pattern\Cross_12.bmp
ANA.Load|1|$$PA1
' This will generate the 4 Downscaled Versions
ANA.DownScale|1|2
VAN.$$COT=110
VAN.$$PIH=-60
' Setting Maximum stored results to 40 - is 10 by default.
ANA.Set_Data|6|40
ANA.Snap|0
ANA.FindPattern4|0|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH|$$COT|$$PIH
PRT.----------------------------
STS.CLEAR
ANA.GetResultsPack|0
DMP.6
PAU.5
ANA.GetResultsPack|1|1
DMP.6
MBX.Ready
ENR.
```

Remarks

-

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

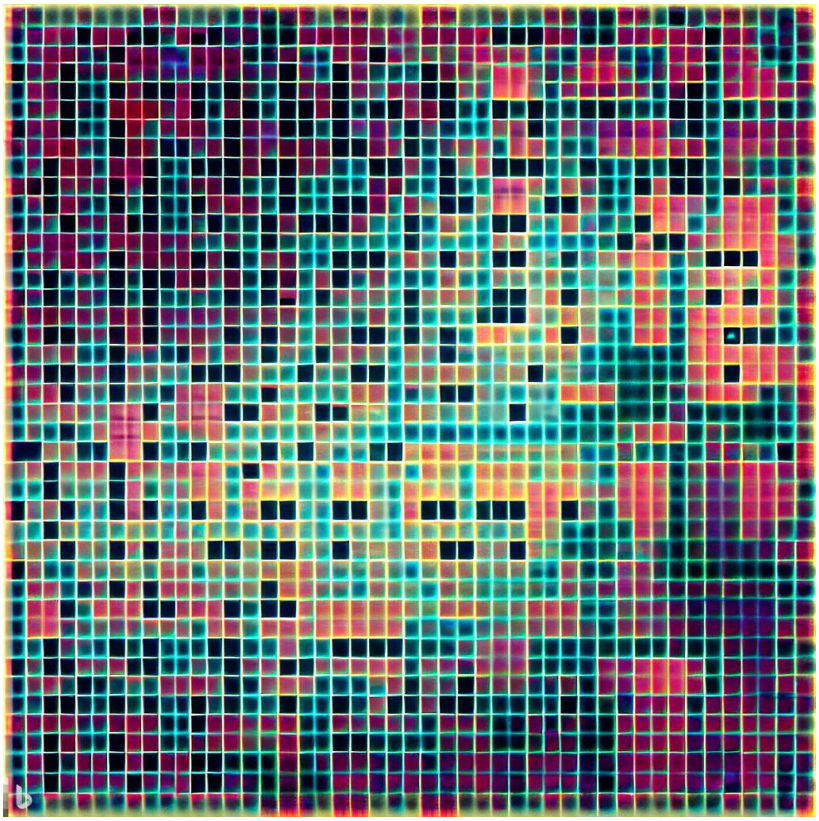
© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.11 ANA.ResetList

MiniRobotLanguage (MRL)

ANA.ResetList

Clear Results from Find-Operation



Intention

This command has no parameters.
It will reset the FindPattern-Result List which internal Name is C01.

Here is the Code that shows the usage of ANA.ResetList

```
STW.ct|FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
'SWP.4090,2|640,400|T
ANA.Snap|0
' Please adapt the filename of the saved Pattern
ANA.Load|1|?path\De16.bmp
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min. Hits (Pixel) maximu
ANA.fpa|0|1|0|115|0
DMP.6
ANA.gres
DMP.6
ANA.ResetList
ANA.gres
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.6
MBX.All cleared
ENR.
```

Syntax

ANA.ResetList

Parameter Explanation

This Command has no Parameters.

Example

–

Remarks

-

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.12 ANA.ResetResultsPack

MiniRobotLanguage (MRL)

ANA.ResetResultsPack

Sorts the Resultspack based on the specified parameter.

Intention

The ANA.ResetResultsPack command allows you to clear the Results-Pack completely.
It removes all stored results, providing a clean slate for new operations.

Syntax

ANA.SortResultsPack[|P1]

Parameter Explanation

This command does not require any parameters.

Example

```
'To clear the Results-Pack, use the following command:
ANA.ResetResultsPack
```

Remarks

- The ANA.ResetResultsPack command is useful when you want to start fresh and remove all previously stored results.
- By clearing the Results-Pack, you ensure that no old or irrelevant results are retained, avoiding any confusion or mix-up in subsequent operations.
- It's important to note that once the Results-Pack is cleared, all previous results will be permanently deleted and cannot be recovered.
- Use this command with caution, especially if you still need to refer to or analyze the existing results.

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.13 ANA.SetMaxResult

MiniRobotLanguage (MRL)

ANA.SetMaxResult

Sets the maximum number of results to be stored.

Intention

The `ANA.SetMaxResult` command allows you to define the maximum number of results to be stored in the Results-Pack.
By setting this limit, you can control the amount of memory and time used and manage the quantity of results obtained.
This command is useful when you want to restrict the number of results to a specific threshold or just to save time.

Syntax

ANA.SetMaxResult[|P1]

Parameter Explanation

P1 - Maximum number of results to acquire.

Example

```
' To set the maximum number of results to 100, use the following command:
ANA.SetMaxResult|100
```

Remarks

-

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.13.14 ANA.SnapFind1

MiniRobotLanguage (MRL)

ANA.Snapfind1

Snapshot current Window and start a Pattern-Find-Operation

Intention

The '`Snapfind`' command is an integral tool for locating and clicking a pattern within a window. It seamlessly merges the '`ANA.Snap`' and '`ANA.FindPattern`' commands, providing a robust solution for pattern recognition and interaction.
Moreover, it includes a feature for adjusting the target point by a specific pixel distance, offering you precision control.

The Snapfind command operates as follows:

1. It captures a snapshot of the current window, storing it in internal IR No. 24.
2. It loads the desired pattern into internal IR No. 23.
3. It executes the FindPattern command with IR 23 and 24.
4. If a match is identified, it utilizes the `MAP`.-Command to locate the first found object (as E9 with the size 8x8).
5. Subsequent to this, if a '`MMV`.' or '`MLC`.' etc. is written, the mouse will swiftly move to the point and execute the click.

The Snapfind command also provides a Target Adjustment feature, enabling the user to shift the target from the found position by a predetermined amount. This feature is beneficial in scenarios where the user needs to locate a point and click in a different location, such as identifying a checkmark using its text, but clicking on the checkmark box. An example of this would be locating checkmarks in an 'Unsupported Browser'."

Instead of searching the Circles to click on, like in the following picture ....

You can search the unique Text and then specify an Coordinate Adjustment - where to click. In this case you can not accidentally click the wrong checkmarks.

© 2023 Theo Gottwald * https://www.smart-package.com

In this Example,
`ANA.Snapfind1|$$LAA|0|0|1|55,-10|1`
will directly set the Active Point (where the Mouse will move to) to the Target.
This is possible because we use the "55,-10" as adjustments.

You can use the long version of Snapfind and use the "Findings on the TOS" this way:

```
'**********************************
' ANA.Snapfind-Sample
'**********************************
STW.ct|Chrome_WidgetWin_1|Amazon - Brave
'SWP.70,89|1195,977|T
$$LAA=?pattern\Checbkm.bmp
ANA.Snapfind1|$$LAA|0|0|1|55,-10|1
' Remove Number of Finds from TOS
POP.$$COO
' Get Coords from TOS
POP.$$COO
' Move Mouse to target
MMV.t|$$COO
ENR.
```

With '`Snapfind1`', all findings are conveniently left on the TOS.
At position 00, you'll first encounter the number of findings, capped by **P6**.
Moving on to position 01, you'll discover the coordinate of the first finding, with additional findings listed in the subsequent positions.

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->LongInt      :1

Stackpos: 001 ->PointAPI     :498,742

-----------------------------------
```

```
The short Form of Snapfind1 is to just use the first Finding
which will be automatically localized using an internal MAP. Command.
This way you can also write:

'**********************************
' ANA.Snapfind-Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
STW.ct|Chrome_WidgetWin_1|Amazon - Brave
'SWP.70,89|1195,977|T
$$LAA=?pattern\Checbkm.bmp
ANA.Snapfind1|$$LAA|0|0|1|55,-10|1
' Move Mouse to target and Click
MLC.
ENR.
```

Syntax

ANA.Snapfind1|P1[|P2][|P3][|P4][|P5][|P6]

Parameter Explanation

P1 - [File] Pattern-Image File
P2 - ***opt.*** Color Tolerance for Search
P3 - ***opt.*** Pixel-Hits, this number depends on the total number of Pixel in the Pattern. For example, 63 would be a minimum Hit number of 63 Pixels that must match the Pattern including the Tolerance. If you specify a negative number like "-15" then this is treated as a Percentage of the total Pixel-Number, means "-75" would be 75%. If omitted or 0 the default value will be 85% of Pixel-Number in Pattern.
P4 - ***opt. Debug-Flag 0/1***
P5 - ***opt. [Coords]*** this is a coordinate like "0,0" or "`-200,-10`". This will move the Target Point by the specified amount of Pixels.
P6 - ***opt.*** Maximum Number of Finds that should be stored.

More Details:

For more detailed Information about **P4** and **P5** please see ANA.FindPattern 1537

Example

```
'**********************************
' ANA.Snapfind-Sample 1
'**********************************
STW.ct|Chrome_WidgetWin_1|Amazon - Brave
'SWP.70,89|1195,977|T
$$LAA=?pattern\Checbkm.bmp
ANA.Snapfind1|$$LAA|0|0|1|55,-10|1
' Remove Number of Finds from TOS
POP.$$COO
' Get Coords from TOS
POP.$$COO
' Move Mouse to target
MMV.t|$$COO
ENR.


'**********************************
' ANA.Snapfind-Sample 2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
STW.ct|Chrome_WidgetWin_1|Amazon - Brave
'SWP.70,89|1195,977|T
$$LAA=?pattern\Checbkm.bmp
ANA.Snapfind1|$$LAA|0|0|1|55,-10|1
' Move Mouse to target and Click
MLC.
ENR.
```

Remarks

For more detailed Information about **P4** and **P5** please see ANA.FindPattern 1537

Limitations:

-

See also:

- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

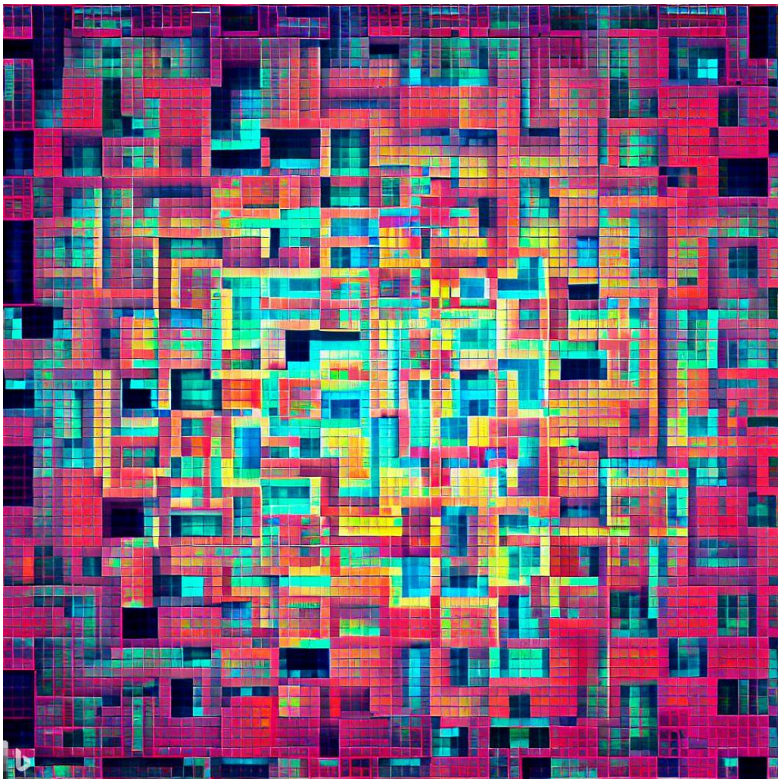
3.55.4.13.15 ANA.SortResultsPack

ANA.GetResult - Get Result on TOS Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.SortResultsPack

Sorts the Resultspack based on the specified parameter.



Intention

The `ANA.SortResultsPack` command allows you to sort the Resultspack based on a specific parameter.
It takes one parameter, P1, which determines the sorting order.
By sorting the Resultspack, you can organize the results in either descending or ascending order, depending on your needs.

Syntax

ANA.SortResultsPack[|P1]

Parameter Explanation

P1 - Sort Order: 0 for descending sort, 1 for ascending sort.

About the P1 Parameter

© 2023 Theo Gottwald * https://www.smart-package.com

The **P1** parameter specifies the sorting order for the Resultspack. It controls whether the results will be sorted in descending or ascending order.

If **P1** is set to 0, the Resultspack will be sorted in descending order.
This means that the highest values or the most relevant results will appear first in the sorted list.

If **P1** is set to 1, the Resultspack will be sorted in ascending order.
This means that the lowest values or the least relevant results will appear first in the sorted list.

Example

```
'To sort the Resultspack in descending order, use the following command:
ANA.SortResultsPack|0

'To sort the Resultspack in ascending order, use the following command:
ANA.SortResultsPack|1
```

Remarks

-

Limitations:

-

See also:

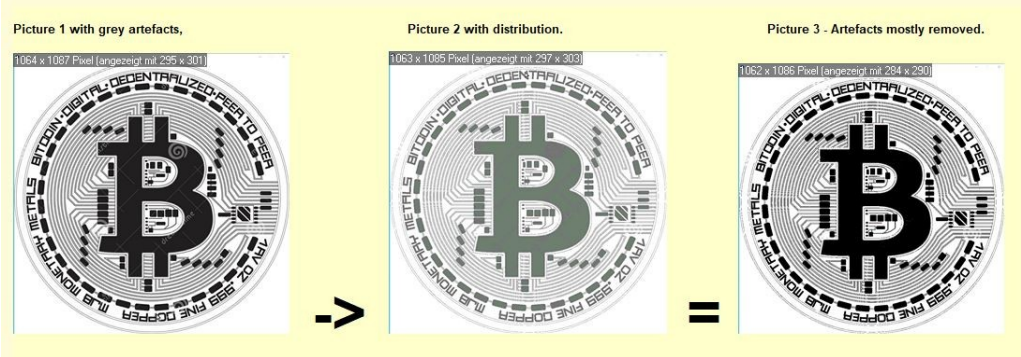
- ANA_FindPattern 1537

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14 Pixel-Operation

ANA.Pixel-Operations

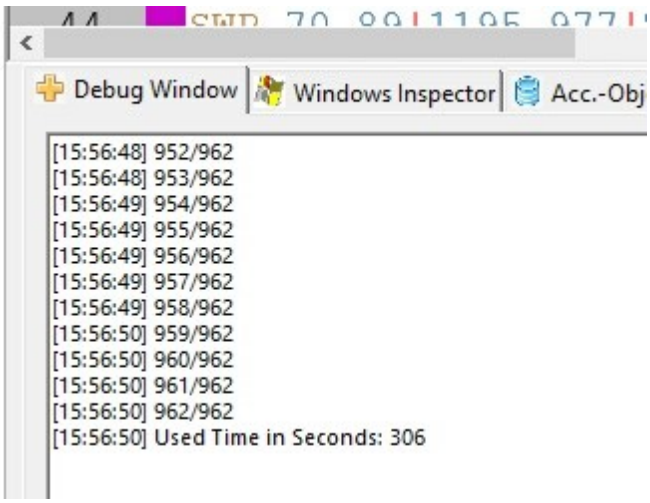
These Commands work on IR's only.
They can be used to manipulate pictures that you have loaded.
While these Operations can be slow on large pictures, you have a lot of options that you may not have available elsewhere.



Here you can see some Operations that are just Samples.

As the SPR is fast enough even for Pixel-based image-processing, you can do a lot of things. Much easier then with other systems.
And faster then most other Scripting languages.

1. Remove Green Background on Image using an SPR-Script
In this Sample we will remove the Background on a Image using the SPR.
For this we will read the Pixel, check if its mostly Green, and in that case replace it with the colors taken from another Picture.
Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.



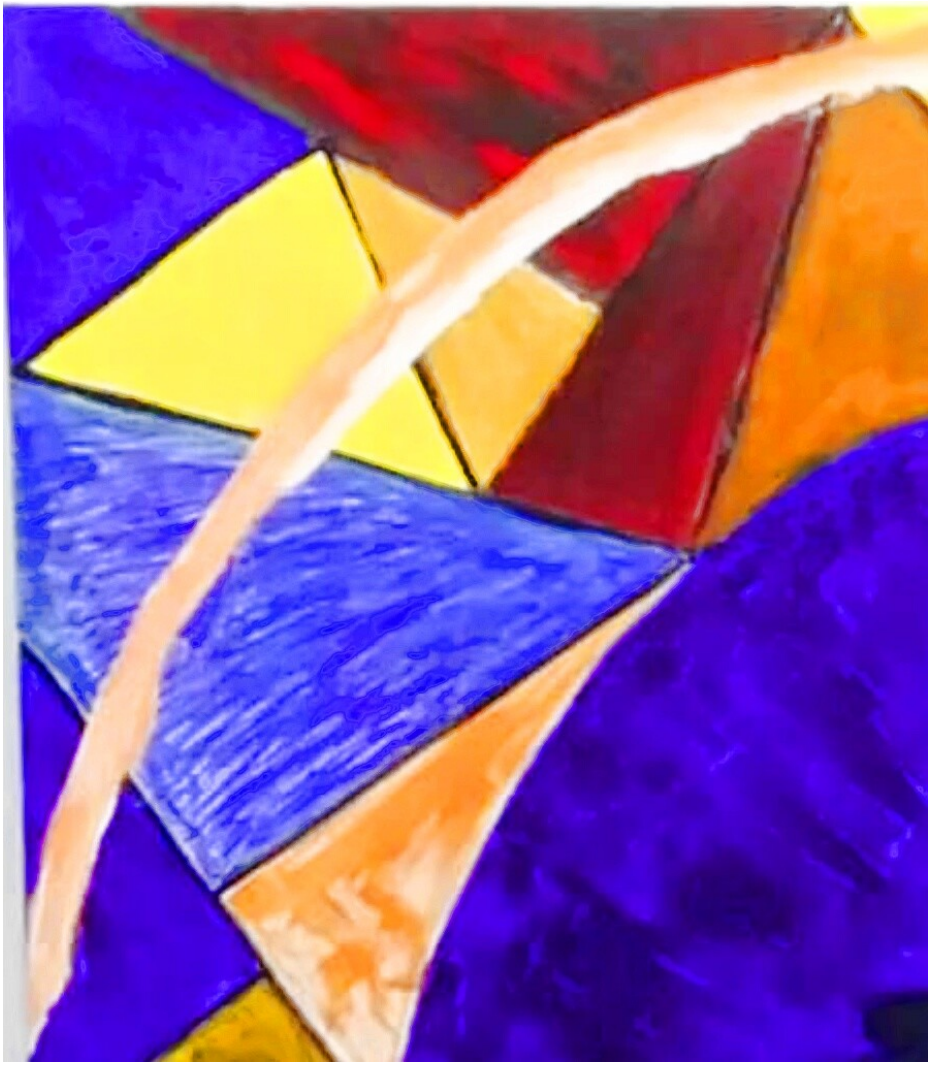
© 2023 Theo Gottwald * https://www.smart-package.com

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
VAF.$$PIC=?pattern\SarahB_640.png
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.Show|0!
ANA.Show|1!
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
ANA.Save|0|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

This box is filled and it is drawn with inverted background color.
The second Part of the Script will draw 2500 random Lines.

```
012 0000000000000269 0001 VAN.$$TIM=#time#
013 0000000000000094 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000051 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000000055 0001 VAF.$$PIC=?pattern\SarahB_640.jpg
016 0000000000861741 0001 ANA.Load|0|$$PIA
017 0000000002200014 0001 ANA.Load|1|$$PIB
018 0000000000000524 0001 ANA.GetRes|0|$$XSZ|$$YSZ
019 0000000000000084 0963 FOR.$$YPO|1|$$YSZ
020 0000000000000003 0962
021 0000000000000099 616642 FOR.$$XPO|1|$$XSZ/2
022 0000000000000367 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
023 0000000000000392 615680 CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
024 0000000000000054 615680 IVV.$$ERG>2
025 0000000000000264 423959 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
026 0000000000000241 423959 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
027 0000000000000003 615680 EIF.
028 0000000000000004 616642 NEX.
029 0000000000000004 0963 NEX.
030 0000000000279013 0001 ANA.Save|0|$$PIC
031 0000000000000364 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000000005 0001
033 0000000000000000 0001 DMP.1
```

The Speed-dump shows that the Robot can read about 40.000 Pixels per second.

2. Color-Processing

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this Sample we will additionally change the Colors.For this we use the two Commands:

```
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
  ' This is just for example normally you would use ANA.GetPixelColor_RGB which co
        ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
  ' We Combine the Colors in away that Blue and Green is exchanged.
        ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
  ' And Set the Pixel
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
```

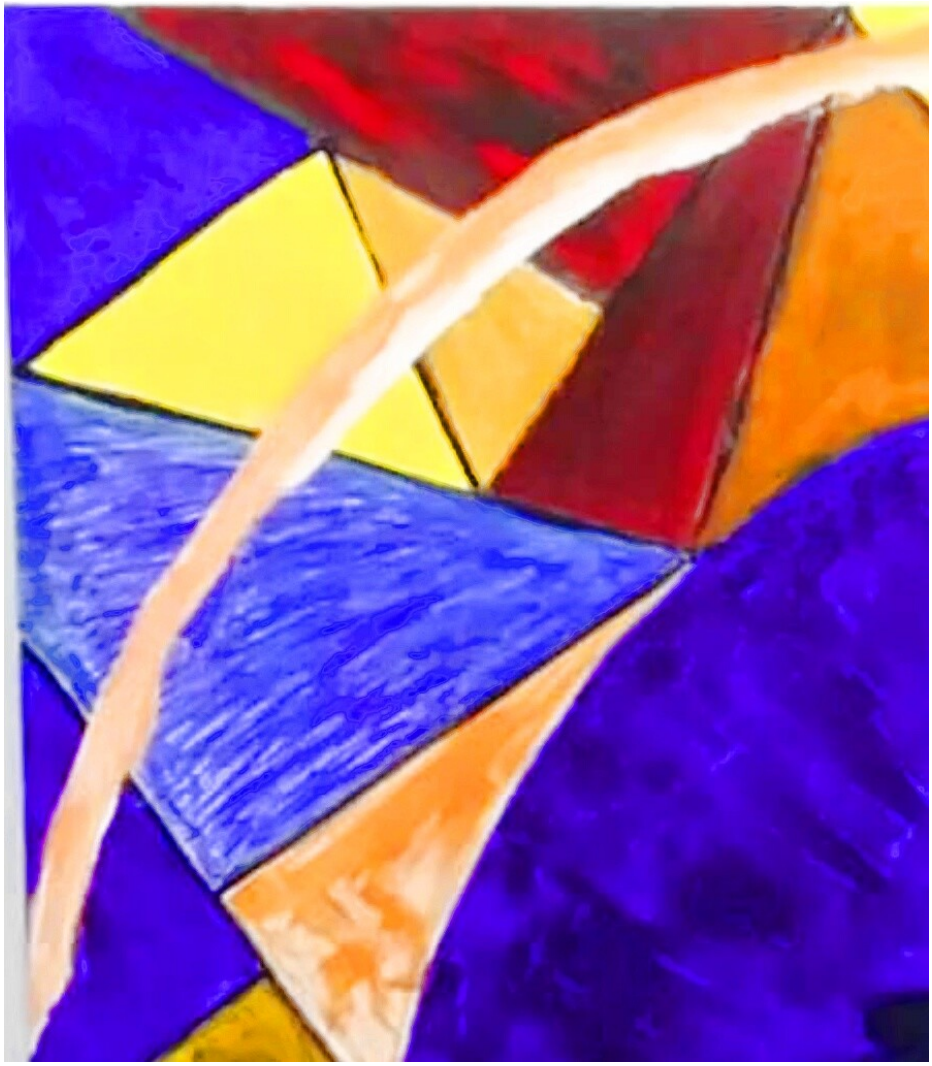
And then Paint the Picture.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

```
  ' Here is the Code.
  ' On my Computer it uses 5 MinutesProcessing time for a Picture of Medium size i
  VAN.$$TIM=#time#
  VAF.$$PIA=?pattern\Sarah_640.jpg
  VAF.$$PIB=?pattern\Mahakala2.png
  ANA.Load|0|$$PIA
  ANA.Load|1|$$PIB
  ANA.GetRes|0|$$XSZ|$$YSZ
  FOR.$$YPO|1|$$YSZ
    DBP.$$YPO/$$YSZ
    FOR.$$XPO|1|$$XSZ/2
      ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
      CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
      IVV.$$ERG>2
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
        ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
        ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
      EIF.
    NEX.
  NEX.
  CAL.$$TIM=#time#-$$TIM
  DBP.Used Time in Seconds: $$TIM
  DMP.1
  ANA.Show|0!
  ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

The Script used ~300 Seconds to complete process the picture.

Here is the Speed-Dump:

```
012 0000000000000274 0001 VAN.$$TIM=#time#
013 0000000000000100 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000058 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000392529 0001 ANA.Load|0|$$PIA
016 0000000001986990 0001 ANA.Load|1|$$PIB
017 0000000000000568 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000126 0963 FOR.$$YPO|1|$$YSZ
019 0000000000060423 0962 DBP.$$YPO/$$YSZ
020 0000000000000146 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000504 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000493 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000082 615680 IVV.$$ERG>2
024 0000000000000383 422988 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000444 422988 ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
026 0000000000000370 422988 ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
027 0000000000000336 422988 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
028 0000000000000004 615680 EIF.
029 0000000000000006 616642 NEX.
030 0000000000000006 0963 NEX.
031 0000000000000310 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000230528 0001 DBP.Used Time in Seconds: $$TIM
033 0000000000000000 0001 DMP.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

2. Background or Object-Processing

In this Sample we will just fill the background with white color.
The Script takes 207 Second on my computer.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
EXO.?path\
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

This is the result of the operation

```
012 0000000000000294 0001 VAN.$$TIM=#time#
013 0000000000000113 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000370472 0001 ANA.Load|0|$$PIA
015 0000000000000566 0001 ANA.GetRes|0|$$XSZ|$$YSZ
016 0000000000000131 0963 FOR.$$YPO|1|$$YSZ
017 0000000000184554 0962 DBP.$$YPO/$$YSZ
018 0000000000000154 616642 FOR.$$XPO|1|$$XSZ/2
019 0000000000000527 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
020 0000000000000513 615680 CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
021 0000000000000089 615680 IVV.$$ERG>2
022 0000000000000289 423959 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
023 0000000000000004 615680 EIF.
024 0000000000000006 616642 NEX.
025 0000000000000006 0963 NEX.
026 0000000000000330 0001 CAL.$$TIM=#time#-$$TIM
027 0000000000118643 0001 DBP.Used Time in Seconds: $$TIM
028 0000000000000000 0001 DMP.1
```

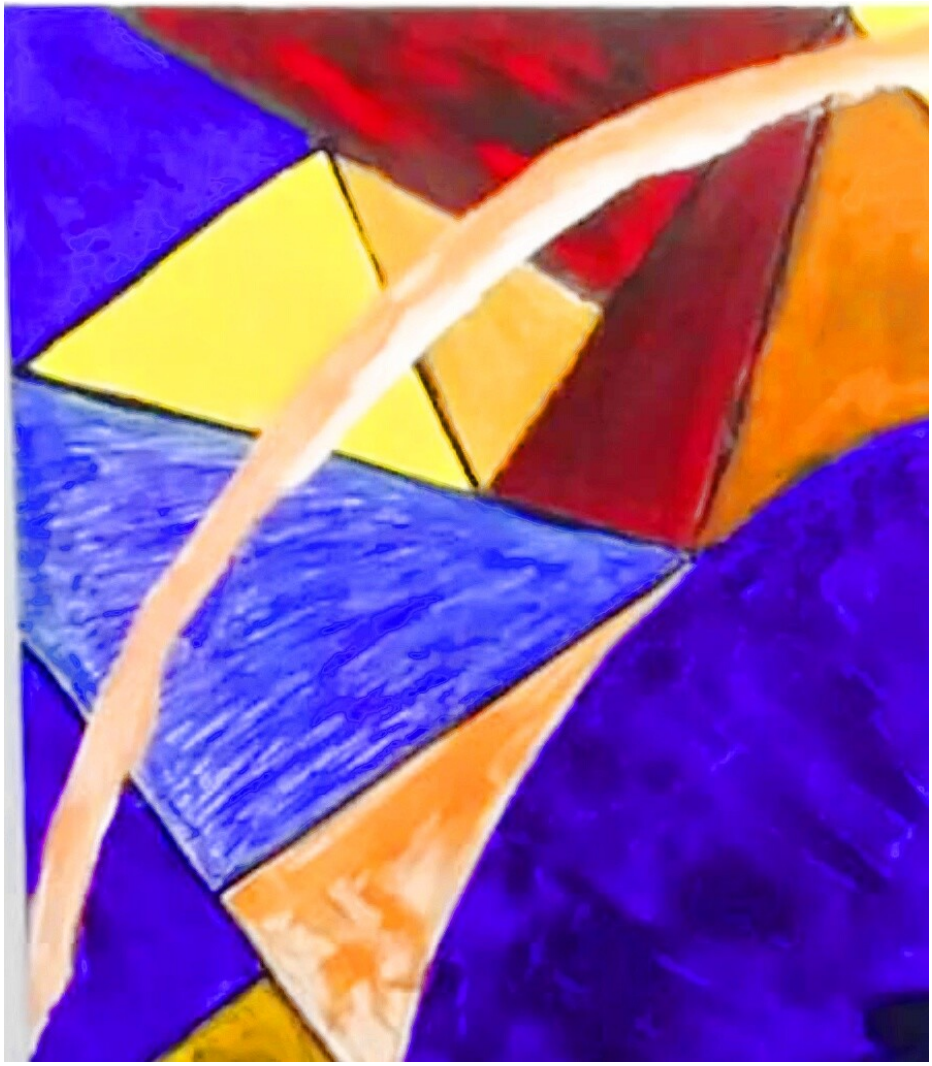
Here you can see the Speed-Dump of the Script.

3. Remove Object from Background

In this Sample we will have the Background, and a Person on the Background.
This Script will then Paint the Object yellow.

*© 2023 Theo Gottwald * https://www.smart-package.com*

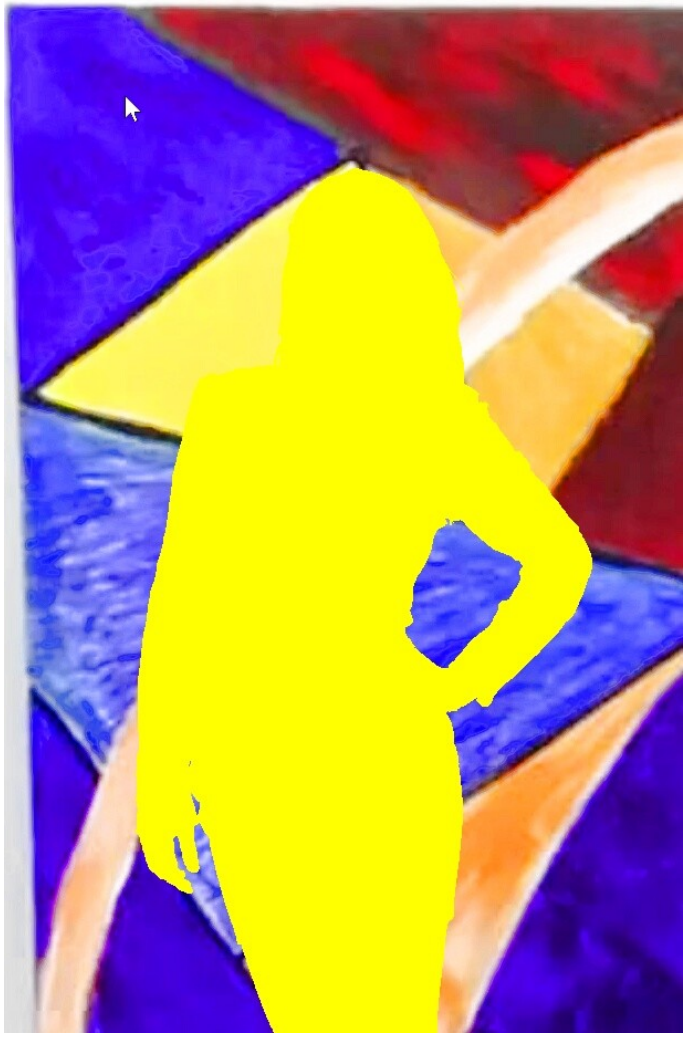
© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com



```
013 0000000000000190 0001 VAN.$$TIM=#time#
014 0000000000000088 0001 VAF.$$PIA=?pattern\SarahB_640.png
015 0000000000000062 0001 VAF.$$PIB=?pattern\Mahakala2.png
016 0000000000000055 0001 VAF.$$PIC=?pattern\SarahR_640.jpg
017 0000000000584825 0001 ANA.Load|0|$$PIA
018 0000000001989721 0001 ANA.Load|1|$$PIB
019 0000000000000409 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000000078 0963 FOR.$$YPO|1|$$YSZ
021 0000000000000003 0962
022 0000000000000091 616642 FOR.$$XPO|1|$$XSZ/2
023 0000000000000224 615680 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
024 0000000000000235 615680 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
025 0000000000000074 615680 IVV.$$PCA=$$PCB
026 0000000000000217 424450 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
027 0000000000000003 615680 ELS.
028 0000000000000002 191230
029 0000000000000199 191230 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
030 0000000000000003 615680 EIF.
031 0000000000000004 616642 NEX.
032 0000000000000004 0963 NEX.
033 0000000000231127 0001 ANA.Save|0|$$PIC
034 0000000000000392 0001 CAL.$$TIM=#time#-$$TIM
035 0000000000000005 0001
036 0000000000000000 0001 DMP.1
```

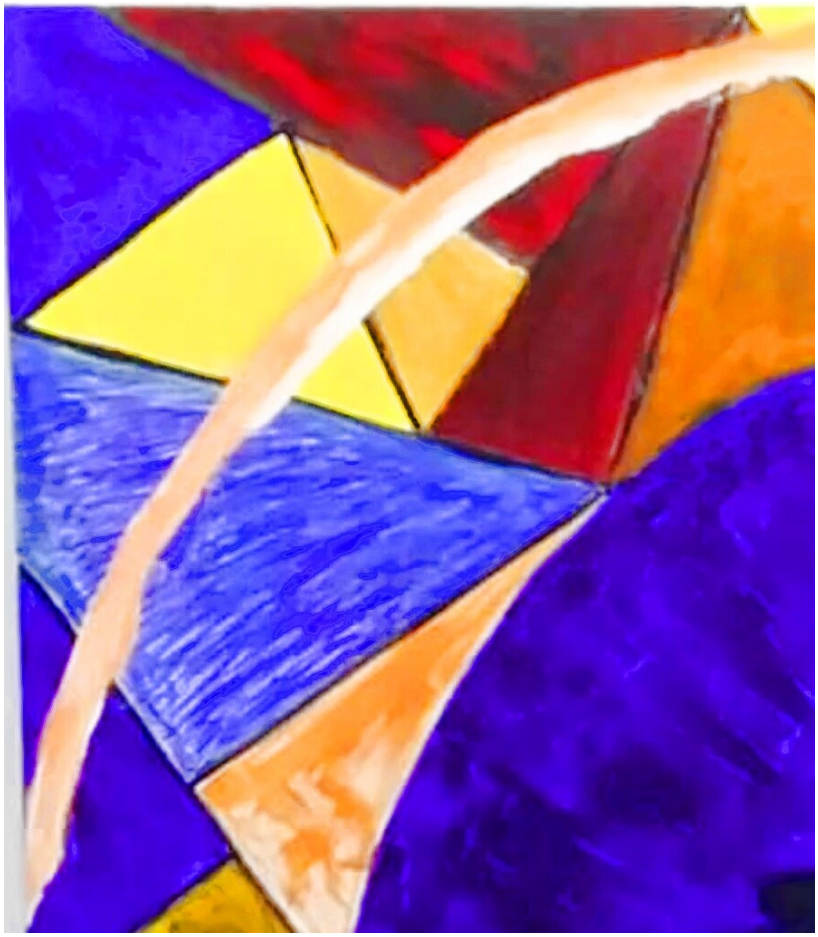
If we just

*© 2023 Theo Gottwald * https://www.smart-package.com*

4. Extract Object from known Background
If we do just change 1 Letter in the Script, we get the inverted result.
The Script will then extract the Person from the Baclground.
Hint: This is a very Simple Script it will only work with Pictures that are uncompressed like "-bmp", or ".png" etc.
Using other Formats you will need to apply a Tolerance.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.png
VAF.$$PIB=?pattern\Mahakala2.png
VAF.$$PIC=?pattern\SarahR_640.jpg
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
    ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
    NVV.$$PCA=$$PCB
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
    ELS.
    ' Paint it Yellow
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
    EIF.
  NEX.
NEX.
ANA.Save|0|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000179 0001 VAN.$$TIM=#time#
014 0000000000000084 0001 VAF.$$PIA=?pattern\SarahB_640.png
015 0000000000000060 0001 VAF.$$PIB=?pattern\Mahakala2.png
016 0000000000000057 0001 VAF.$$PIC=?pattern\SarahR_640.jpg
017 0000000000583742 0001 ANA.Load|0|$$PIA
018 0000000002013220 0001 ANA.Load|1|$$PIB
019 0000000000000617 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000000083 0963 FOR.$$YPO|1|$$YSZ
021 0000000000000003 0962
022 0000000000000097 616642 FOR.$$XPO|1|$$XSZ/2
023 0000000000000242 615680 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
024 0000000000000249 615680 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
025 0000000000000080 615680 NVV.$$PCA=$$PCB
026 0000000000000236 191721 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
027 0000000000000004 615680 ELS.
028 0000000000000003 423959
029 0000000000000217 423959 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
030 0000000000000003 615680 EIF.
031 0000000000000004 616642 NEX.
032 0000000000000004 0963 NEX.
033 0000000000229073 0001 ANA.Save|0|$$PIC
034 0000000000000379 0001 CAL.$$TIM=#time#-$$TIM
035 0000000000000005 0001
036 0000000000000000 0001 DMP.1
```

Here is the Speed-Dump for this Sample.

5. Extract Object from known Background

Using compressed Formats you will need to apply a Tolerance.
We can do this, using the CCD.-Command.

*© 2023 Theo Gottwald * https://www.smart-package.com*

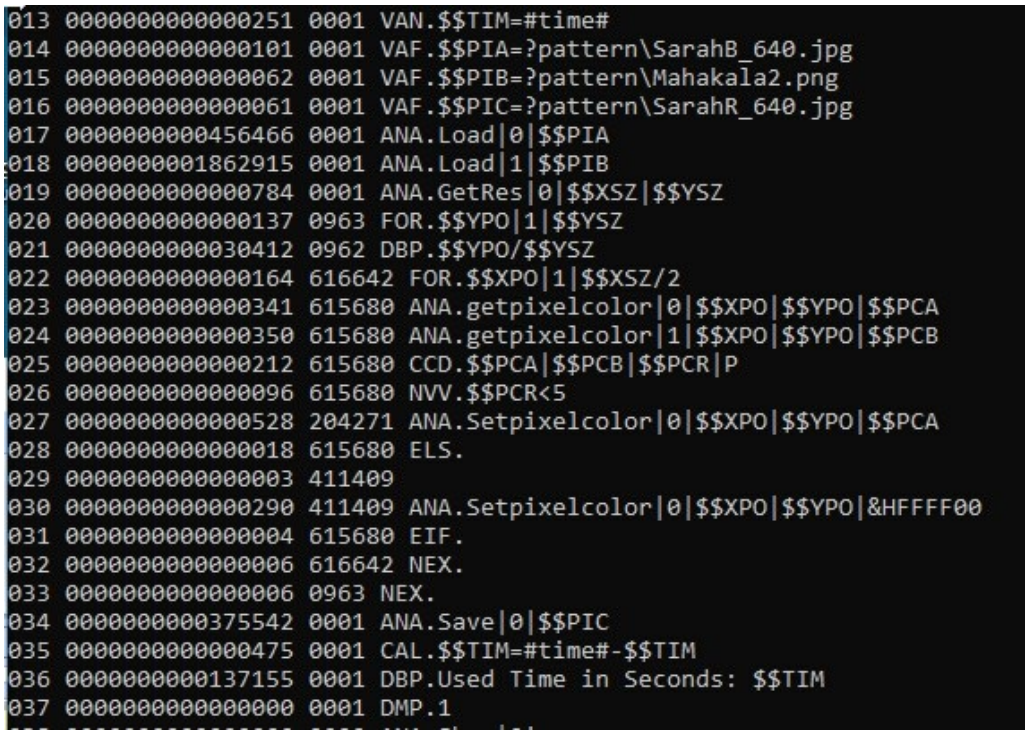
Here we use a Color-Distance of 5%, you can see that sharp borders make trouble.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
VAF.$$PIC=?pattern\SarahR_640.jpg
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
    ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
    CCD.$$PCA|$$PCB|$$PCR|P
    NVV.$$PCR<5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
    ELS.
    ' Paint it Yellow
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
    EIF.
  NEX.
NEX.
ANA.Save|0|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Here we use a known Background to extract Sarah from the second Picture into the third Picture with yellow background.
Because we used a ".jpg"-File we still have some artefacts when using a Color-Distance Value of only 5%. In the second Picture we use a value of 10%. This will remove some problems and produce some other.
Thats why you should not use compressed Images for this sort of operations.



```
013 0000000000000251 0001 VAN.$$TIM=#time#
014 0000000000000101 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000062 0001 VAF.$$PIB=?pattern\Mahakala2.png
016 0000000000000061 0001 VAF.$$PIC=?pattern\SarahR_640.jpg
017 0000000000456466 0001 ANA.Load|0|$$PIA
018 0000000001862915 0001 ANA.Load|1|$$PIB
019 0000000000000784 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000000137 0963 FOR.$$YPO|1|$$YSZ
021 0000000000030412 0962 DBP.$$YPO/$$YSZ
022 0000000000000164 616642 FOR.$$XPO|1|$$XSZ/2
023 0000000000000341 615680 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
024 0000000000000350 615680 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
025 0000000000000212 615680 CCD.$$PCA|$$PCB|$$PCR|P
026 0000000000000096 615680 NVV.$$PCR<5
027 0000000000000528 204271 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
028 0000000000000018 615680 ELS.
029 0000000000000003 411409
030 0000000000000290 411409 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
031 0000000000000004 615680 EIF.
032 0000000000000006 616642 NEX.
033 0000000000000006 0963 NEX.
034 0000000000375542 0001 ANA.Save|0|$$PIC
035 0000000000000475 0001 CAL.$$TIM=#time#-$$TIM
036 0000000000137155 0001 DBP.Used Time in Seconds: $$TIM
037 0000000000000000 0001 DMP.1
```

Doing Color-Changes/Corrections
In this Sample we will change the Colors in the Picture.
For this we will read the Pixel, check if its mostly Green, and in that case replace it with the colors taken from another Picture.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIC=?pattern\SarahC_640.jpg
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
ANA.New|1|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$RED=$$RED*1.1-70
    CAL.$$GRN=$$GRN*1.1-70
    CAL.$$BLUE=$$BLU*1.1-70
    ANA.SetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
  NEX.
NEX.
ANA.Save|1|$$PIC
CAL.$$TIM=#time#-$$TIM
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|1!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

This is the Original picture. . The is the picture with the colors changed.

```
013 0000000000000203 0001 VAN.$$TIM=#time#
014 0000000000000089 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000057 0001 VAF.$$PIC=?pattern\SarahC_640.jpg
016 0000000000461693 0001 ANA.Load|0|$$PIA
017 0000000000000406 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000024014 0001 ANA.New|1|$$XSZ|$$YSZ
019 0000000000000084 0963 FOR.$$YPO|1|$$YSZ
020 0000000000000004 0962
021 0000000000000088 616642 FOR.$$XPO|1|$$XSZ
022 0000000000000379 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
023 0000000000000171 615680 CAL.$$RED=$$RED*1.1-70
024 0000000000000161 615680 CAL.$$GRN=$$GRN*1.1-70
025 0000000000000024 615680 CAL.$$BLUE=$$BLU*1.1-70
026 0000000000000347 615680 ANA.SetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
027 0000000000000004 616642 NEX.
028 0000000000000004 0963 NEX.
029 0000000000249490 0001 ANA.Save|1|$$PIC
030 0000000000000385 0001 CAL.$$TIM=#time#-$$TIM
031 0000000000000005 0001
032 0000000000000000 0001 DMP.1
```

The Speed-dump shows that the Robot can do such correction very fast, yet simple changes on the whole picture can be done faster using other commands.

Picture Mixing

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this Sample we will mix 2 Pictures depending on the relation between Blue and Red.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIC=?pattern\SarahC_640.jpg
ANA.Load|0|$$PIA
ANA.Copy|0|1
ANA.Grey|1
ANA.GetRes|0|$$XSZ|$$YSZ
ANA.New|1|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    IVV.$$BLU>$$RES
      ANA.GetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    EIF.
    ANA.SetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
  NEX.
NEX.
ANA.Save|1|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|1!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

The Script used ~99 Seconds to complete process the picture. If you change the IVV. to NVV. Command, the result will be the third picture or when using $$BLU and $$GRN the fourth picture.

Here is the Speed-Dump:

```
013 0000000000000261 0001 VAN.$$TIM=#time#
014 0000000000000124 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000094 0001 VAF.$$PIC=?pattern\SarahC_640.jpg
016 0000000000467509 0001 ANA.Load|0|$$PIA
017 0000000000017641 0001 ANA.Copy|0|1
018 0000000000129814 0001 ANA.Grey|1
019 0000000000000360 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000114887 0001 ANA.New|2|$$XSZ|$$YSZ
021 0000000000000081 0963 FOR.$$YPO|1|$$YSZ
022 0000000000000004 0962
023 0000000000000083 616642 FOR.$$XPO|1|$$XSZ
024 0000000000000359 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
025 0000000000000076 615680 IVV.$$BLU>$$RED
026 0000000000000363 225727 ANA.GetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
027 0000000000000003 615680 EIF.
028 0000000000000326 615680 ANA.SetPixelRGB|2|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
029 0000000000000005 616642 NEX.
030 0000000000000004 0963 NEX.
031 0000000000395490 0001 ANA.Save|2|$$PIC
032 0000000000000417 0001 CAL.$$TIM=#time#-$$TIM
033 0000000000000005 0001
034 0000000000000000 0001 DMP.1
```

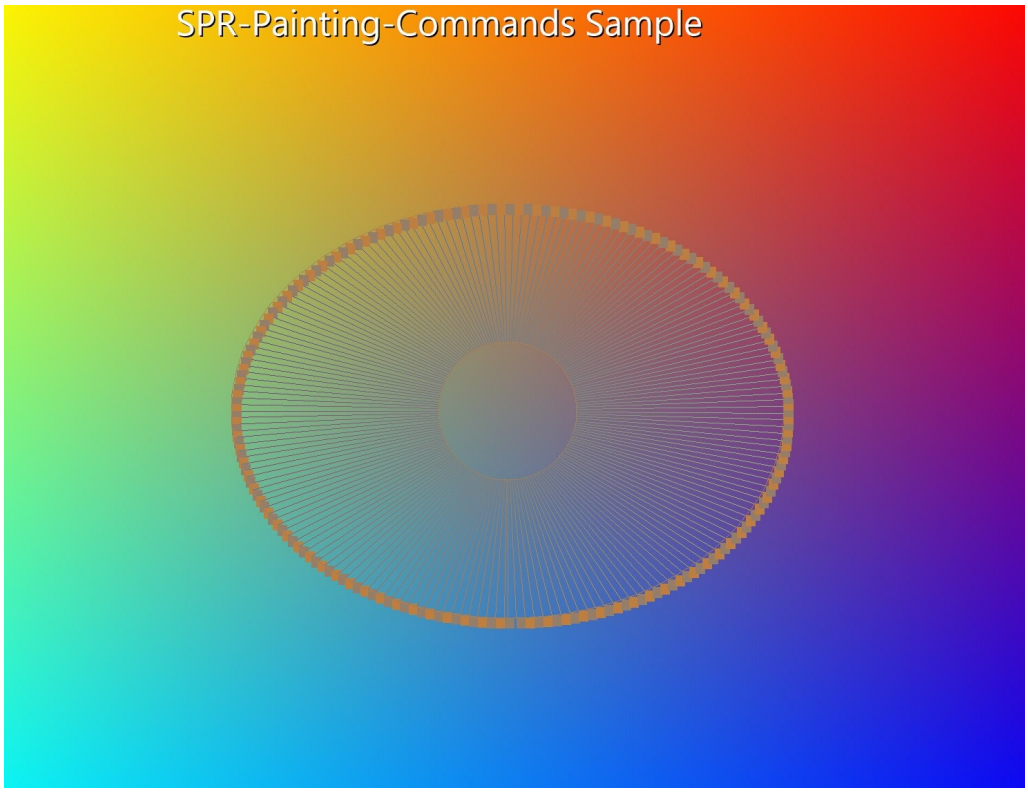

© 2023 Theo Gottwald * https://www.smart-package.com

Here is another Painting Example:

```
$$XPA=1600
$$YPA=1200

CAL.$$XPO=$$XPA/2|i
CAL.$$YPO=$$YPA/2|i

$$XSI=100
$$YSI=100

$$XSO=400
$$YSO=300

ANA.New|0|$$XPA|$$YPA
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|1|$$XPA|$$YPA
ANA.hgradient|1|&H0000FF|&H00FFFF
ANA.Mix|0|1|7
VAN.$$CNT=0
FOR.$$LRA|0|6.28|.0325
  CAX.$$SIN=SIN($$LRA)
  CAX.$$COS=COS($$LRA)

  CAL.$$XPI=$$SIN*$$XSI+$$XPO|i
  CAL.$$YPI=$$COS*$$YSI+$$YPO|i
  CAL.$$XPJ=$$SIN*$$XSO+$$XPO|i
  CAL.$$YPJ=$$COS*$$YSO+$$YPO|i

  'ANA.SetPixelColor|1|$$XPI|$$YPI|&HFF0000
  ANA.DrawLine|1|$$XPI|$$YPI|$$XPJ|$$YPJ|-1
  JIZ.$$XPL|elo
  ANA.GetPixelColor|1|$$XPM|$$YPM|$$COA
  ANA.GetPixelColor|1|$$XPL|$$YPL|$$COB
  ANA.DrawLine|1|$$XPI|$$YPI|$$XPL|$$YPL|$$COA
  ANA.DrawLine|1|$$XPJ|$$YPJ|$$XPM|$$YPM|$$COB
  ANA.DrawBox|1|$$XPJ|$$YPJ|($$XPJ+15)|($$YPJ+15)|$$COB|1
  :elo
  VAN.$$XPL=$$XPI
  VAN.$$YPL=$$YPI
  VAN.$$XPM=$$XPJ
  VAN.$$YPM=$$YPJ
  IVV.$$CNT>30
    ANA.Show|1
    $$CNT=0
  ELS.
    VIC.$$CNT
  EIF.
  '
NEX.
$$TXT=SPR-Painting-Commands Sample
$$COL=&HFFFFFF
  ' Backgroundcolor
$$LAA=-2
' Fontname
$$LAC=Segoe UI
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$XP1=320
$$YPI=40
ANA.PrintAt|1|$$TXT|0|$$LAA|($$XP1+2)|($$YP1+2)|39|$$LAC|1
ANA.PrintAt|1|$$TXT|$$COL|$$LAA|$$XP1|$$YP1|39|$$LAC|1
ANA.Show|1!
ENR.
```



Result of the Painting Script.

Template Script for Pixel-Operations.

```
VAF.$$PIA=?exeloc\Picture.png
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Copy|0|1
ANA.GetRes|0|$$XRE|$$YRE
$$SRC=&HD6E8FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|15|$$COL

$$SRC=&HB4E0FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|15|$$COL

$$SRC=&H59362C
$$COL=&H29369B
ANA.ReplaceColorT|0|$$SRC|15|$$COL
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$SRC=&H78514B
$$COL=&H00007E
ANA.ReplaceColorT|0|$$SRC|65|$$COL

FOR.$$LOX|1|$$XRE
  FOR.$$LOY|1|$$YRE
    ANA.getpixelcolor|1|$$LOX|$$LOY|$$PIB
    ANA.getpixelcolor|1|($$LOX+1)|$$LOY|$$PIA
    ANA.getpixelcolor|1|($$LOX-1)|$$LOY|$$PIC
    ANA.getpixelcolor|1|$$LOX|($$LOY-1)|$$PYA
    ANA.getpixelcolor|1|$$LOX|($$LOY+1)|$$PYC

    IVV.$$PIA=&HFFFFFF
      IVV.$$PIB!$HFFFFFF
        IVV.$$PIC=&HFFFFFF
          ANA.Setpixelcolor|1|$$LOX|$$LOY|&HFFFFFF

        EIF.
      EIF.
    EIF.
    IVV.$$PYA=&HFFFFFF
      IVV.$$PIB!$HFFFFFF
        IVV.$$PYC=&HFFFFFF
          ANA.Setpixelcolor|1|$$LOX|$$LOY|&HFFFFFF
        EIF.
      EIF.
    EIF.
  NEX.
NEX.
GSB.Save|0
ENR.

'============================================================
:Save
$$NUM=$$_01
RND.0|999999|$$RND
FMT.$$RND|000000
$$PAT=?exeloc\Save_$$RND.png
IEF.$$PAT
  GTO.Save
EIF.
ANA.Save|$$NUM|$$PAT
ANA.Show|$$NUM!
RET.
'============================================================
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.1 ANA.GetPixelColor

MiniRobotLanguage (MRL)

ANA.GetPixelColor

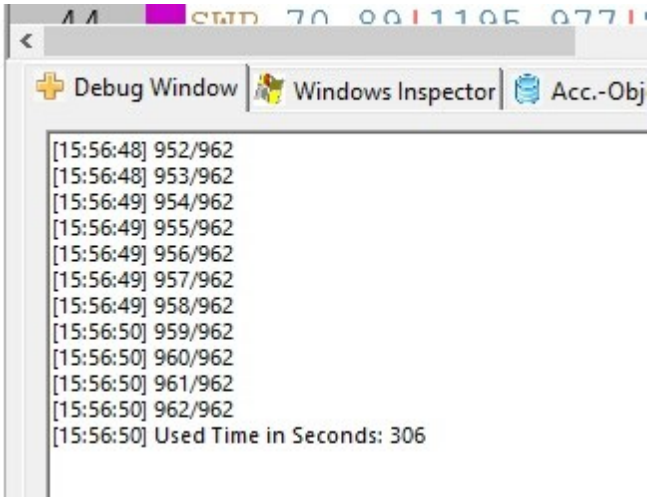
Get the Color of a Single Pixel from an Image-Register (IR)

Intention

This command will read the color of a Single Pixel in an IR. While that does not Sound much, it enables you to do unlimited complex Image processing. Use it together with other Commands like `ANA.SetPixelColor`.

With this command, the returned value is the 24-bit color of the Pixel.

1. Remove Green Background on a picture using an SPR-Script
In this Sample we will remove the Background on a picture using the SPR.
For this we will read the Pixel, check if its mostly Green, and in that case replace
Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.



You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

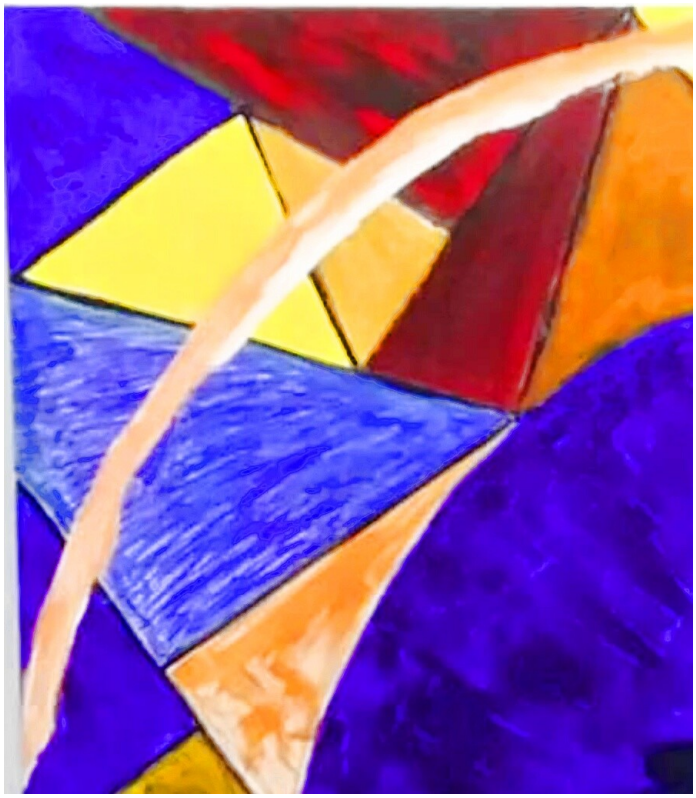
```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Show|0!
ANA.Show|1!
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
ANA.Show|0!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

This box is filled and it is drawn with inverted background color.
The second Part of the Script will draw 2500 random Lines.

```
012 0000000000000319 0001 VAN.$$TIM=#time#
013 0000000000000117 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000062 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000941118 0001 ANA.Load|0|$$PIA
016 0000000002422340 0001 ANA.Load|1|$$PIB
017 0000000000000588 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000183 0963 FOR.$$YPO|1|$$YSZ
019 0000000001338823 0962 DBP.$$YPO/$$YSZ
020 0000000000000224 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000765 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000657 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000110 615680 IVV.$$ERG>2
024 0000000000000365 422388 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000326 422388 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
026 0000000000000005 615680 EIF.
027 0000000000000009 616642 NEX.
028 0000000000000008 0963 NEX.
029 0000000000000430 0001 CAL.$$TIM=#time#-$$TIM
030 0000000000314415 0001 DBP.Used Time in Seconds: $$TIM
031 0000000000000000 0001 DMP.1
```

The Speed-dump shows that the Robot can read about 40.000 Pixels per second.

2. Color-Processing

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this Sample we will additionally change the Colors.For this we use the two Commands:

```
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
' This is just for example normally you would use ANA.GetPixelColor_RGB which co
        ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
' We Combine the Colors in away that Blue and Green is exchanged.
        ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
' And Set the Pixel
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
```

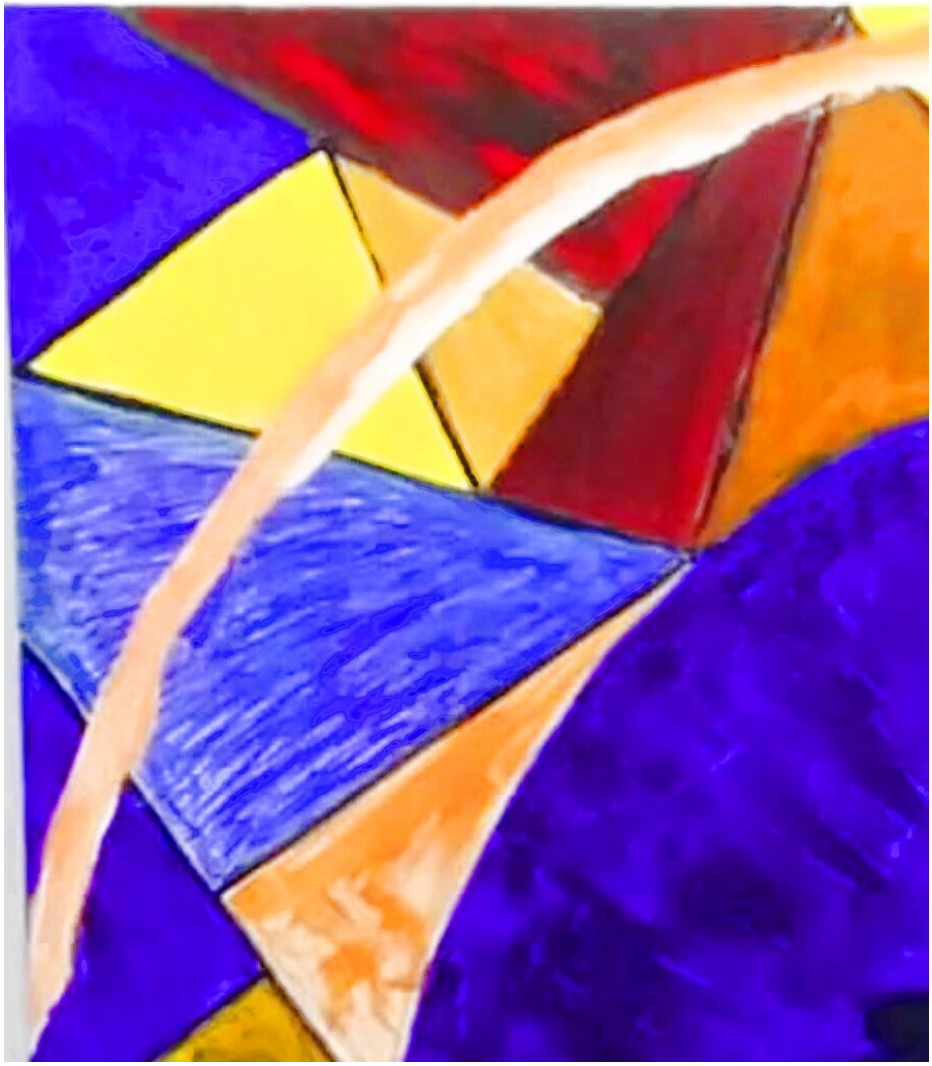
And then Paint the Picture.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

```
' Here is the Code.
' On my Computer it uses 5 MinutesProcessing time for a Picture of Medium size i
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
      ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

The Script used 306 Seconds to complete process the picture.

Here is the Speed-Dump:

```
012 0000000000000274 0001 VAN.$$TIM=#time#
013 0000000000000100 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000058 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000392529 0001 ANA.Load|0|$$PIA
016 0000000001986990 0001 ANA.Load|1|$$PIB
017 0000000000000568 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000126 0963 FOR.$$YPO|1|$$YSZ
019 0000000000060423 0962 DBP.$$YPO/$$YSZ
020 0000000000000146 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000504 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000493 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000082 615680 IVV.$$ERG>2
024 0000000000000383 422988 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000444 422988 ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
026 0000000000000370 422988 ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
027 0000000000000336 422988 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
028 0000000000000004 615680 EIF.
029 0000000000000006 616642 NEX.
030 0000000000000006 0963 NEX.
031 0000000000000310 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000230528 0001 DBP.Used Time in Seconds: $$TIM
033 0000000000000000 0001 DMP.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.GetPixelColor[|P1][|P2][|P3]

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - opt. Variable for returned Color-Value. If omitted TOS is used.

Example

```
See above.
```

Remarks

The `ANA.GetPixelColor` is optimized for Speed. It will use only about 365 Ticks. Therefore the Robot could read more then 43000 Pixels per Second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.2 ANA.GetPixelColor_4D

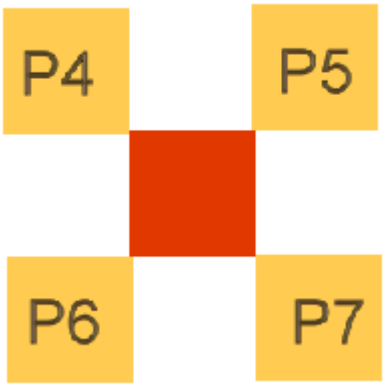
MiniRobotLanguage (MRL)

ANA.GetPixelColor_4D

Get the Color of 4 diagonal neighbor Pixels from an Image-Register (IR)

Intention

This command will read the colors of 4 Pixels that are diagonal neighbor to a given Pixel in an IR.
Together with other Commands like `ANA.SetPixelColor`, It can be used to clean up Images and remove noise.
This command replaces 4 separate GetPixelColor-Commands (plus needed calculations) and is therefore an Option to speed
up Scripts. If a Pixel has no neighbor (Border Pixels) the result of that Color is -1.



The Colors of these Pixels are returned in Variables in P4 to P7.

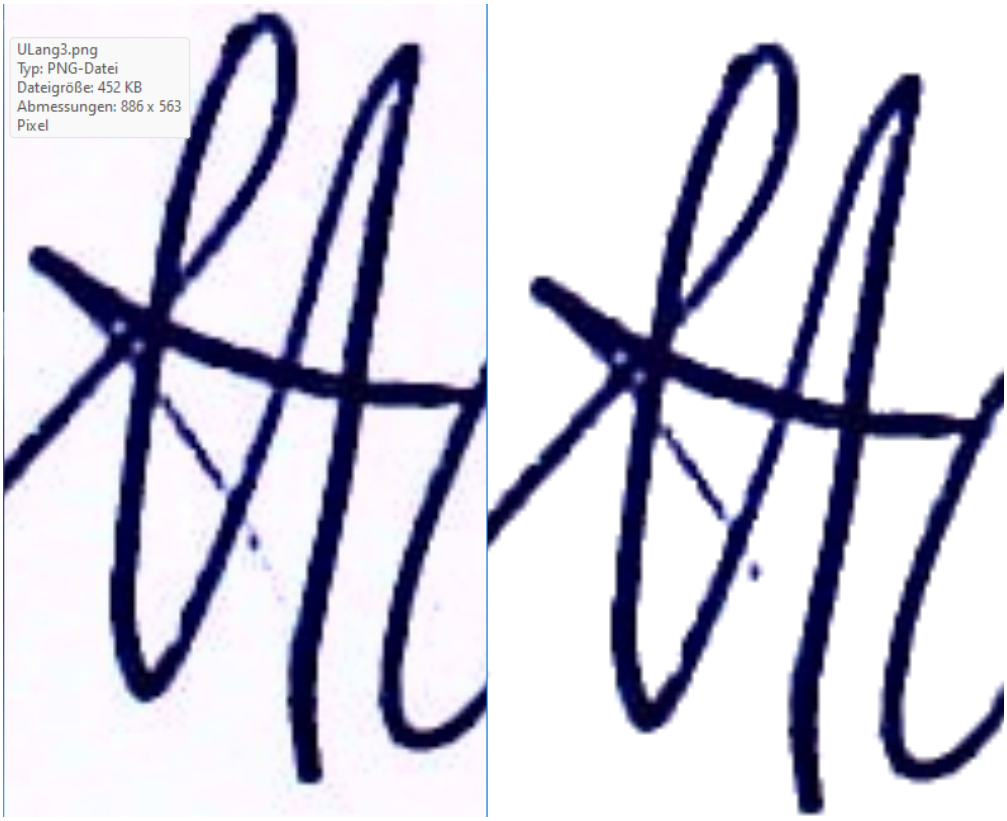
With this command, the returned value is always the 24-bit color of the Pixel unless there is no such Pixel.
If you specify a Pixel that is at a border and has no Pixel to the left/right etc. then the returned value will be -1.

1. Clean up a digitized Picture, remove noise using an SPR-Script
In this Sample we will remove the noise from a picture using the SPR.
For this we will read the Pixel, check if it has only white neighbors, and in that case replace it with a white dot.

Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.

© 2023 Theo Gottwald * https://www.smart-package.com

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.




```
' This Script will clean up the background of a digitized Writing
VAF.$$PIA=?exeloc\ULangK.png
VAF.$$PIB=?exeloc\ULangK1.png
ANA.Load|0|$$PIA
ANA.Show|0!
'ANA.AnalyzeColor|0|1
$$SRC=&HFFF5FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HEADAFE
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HE4F5FF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HC8C0E9
ANA.ReplaceColorT|0|$$SRC|10|$$COL
$$TIM=#dtime#
GSB.Rem_SngPix
CAL.$$TMA=#dsince#|i
DBP.Needed: $$TMA Seconds.
DMP.1
MBX.!
ANA.Show|0!
ANA.Save|0|$$PIB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
'============================================================

:Rem_SngPix
ANA.GetRes|0|$$XSZ|$$YSZ
PRT.Size=$$XSZ,$$YSZ
FOR.$$YPO|2|($$YSZ-1)
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|2|($$XSZ-1)
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
    JIV.$$PIX=&HFFFFFF|EndLoop

    ANA.gpc4s|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD
    ANA.gpc4d|0|$$XPO|$$YPO|$$PAE|$$PAF|$$PAG|$$PAH
    CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF
    IVV.$$ERG>5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
    :EndLoop
  NEX.
NEX.
RET.
```

```
013 0000000000000853 0001 VAF.$$PIA=?exeloc\ULangK.png
014 0000000000000117 0001 VAF.$$PIB=?exeloc\ULangK1.png
015 0000000001035912 0001 ANA.Load|0|$$PIA
016 0000000020560973 0001 ANA.Show|0!
017 0000000000000061 0001
018 0000000000000894 0001 VAR.$$SRC=&HFFF5FF
019 0000000000000092 0001 VAR.$$COL=&HFFFFFF
020 0000000001940416 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
021 0000000000000034 0001
022 0000000000000417 0001 VAR.$$SRC=&HEADAFE
023 0000000002296598 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
024 0000000000000063 0001
025 0000000000000398 0001 VAR.$$SRC=&HE4F5FF
026 0000000001890570 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
027 0000000000000058 0001
028 0000000000000394 0001 VAR.$$SRC=&HC8C0E9
029 0000000002117531 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
030 0000000000000792 0001 VAR.$$TIM=#dtime#
031 0000000000000222 0001 GSB.Rem_SngPix
032 0000000000000214 0001 CAL.$$TMA=#dsince#|i
033 0000000000019735 0001 DBP.Needed: $$TMA Seconds.
034 0000000000000000 0001 DMP.1
035 0000000000000000 0000 MBX.!
036 0000000000000000 0000 ANA.Show|0!
037 0000000000000000 0000 ANA.Save|0|$$PIB
038 0000000000000000 0000 ENR.
039 0000000000000000 0000
040 0000000000000000 0000
041 0000000000000033 0001 :Rem_SngPix
042 0000000000003052 0001 ANA.GetRes|0|$$XSZ|$$YSZ
043 0000000022353444 0001 PRT.Size=$$XSZ,$$YSZ
044 0000000000000230 0562 FOR.$$YPO|2|($$YSZ-1)
045 0000000000003158 0561 PRT.$$YPO/$$YSZ
046 0000000000000244 496485 FOR.$$XPO|2|($$XSZ-1)
047 0000000000000321 495924 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
048 0000000000000106 495924 JIV.$$PIX=&HFFFFFF|EndLoop
049 0000000000000003 495924
050 0000000000000577 495924 ANA.gpc4s|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD
051 0000000000000601 495924 ANA.gpc4d|0|$$XPO|$$YPO|$$PAE|$$PAF|$$PAG|$$PAH
052 0000000000000975 495924 CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+(
F)+($$PAF=&HFFFFFF)+($$PAG=&HFFFFFF)+($$PAH=&HFFFFFF)
053 0000000000000082 495924 IVV.$$ERG>5
054 0000000000000000 0000 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
055 0000000000000004 495924 EIF.
056 0000000000000004 495924 :EndLoop
057 0000000000000006 496485 NEX.
058 0000000000000005 0562 NEX.
059 0000000000000000 0001 RET.
060 0000000000000000 0000
```

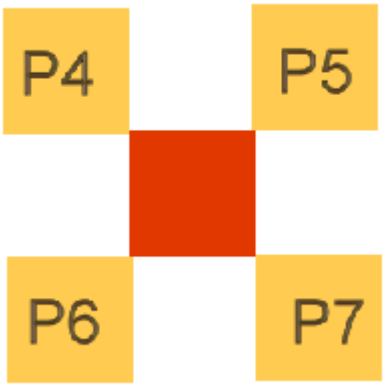

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.GetPixelColor_4D|P1|P2|P3[|P4][|P5][|P6][|P7]
ANA.GPC4D|P1|P2|P3[|P4][|P5][|P6][|P7]

Parameter Explanation

P1 - Source Image-Register number
P2 - X - Center Pixel Coordinate X-Value
P3 - Y - Center Pixel Coordinate Y-Value
P4 - opt. Variable for returned Color-Value for the Upper-Left Pixel.
P5 - opt. Variable for returned Color-Value for the Upper-Right Pixel.
P6 - opt. Variable for returned Color-Value for the Lower-Left Pixel.
P7 - opt. Variable for returned Color-Value for the Lower-RIght Pixel.
If you omit **P4** to **P7**, TOS is used.



Example

```
See above.
```

Remarks

The `ANA.GetPixelColor_4D` is optimized for Speed. It will use only about 600 Ticks. Therefore the Robot can execute the command typically more then 26600 times per Second.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ANA.GetPixelColor_4S 1634

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.3 ANA.GetPixelColor_4S

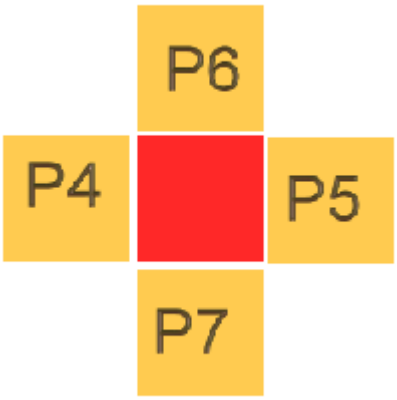
MiniRobotLanguage (MRL)

ANA.GetPixelColor_4S

Get the Color of the 4 direct neighbors of a Single Pixel from an Image-Register (IR)

Intention

This command will read the colors of 4 Pixels that are direct neighbor to a given Pixel in an IR.
Together with other Commands like `ANA.SetPixelColor`, It can be used to clean up Images and remove noise.
This command replaces 4 separate GetPixelColor-Commands (plus needed calculations) and is therefore an Option to speed
up Scripts. If a Pixel has no neighbor (Border Pixels) the result of that Color is -1.



The Colors of these Pixels are returned in Variables in P4 to P7.

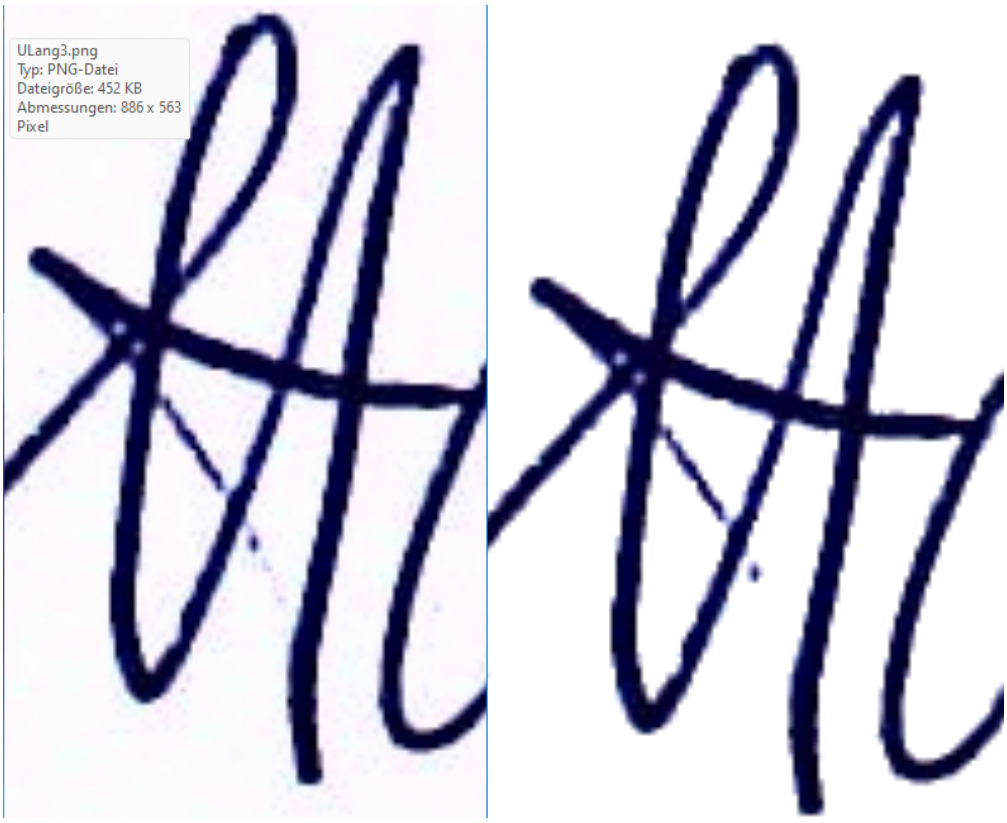
With this command, the returned value is always the 24-bit color of the Pixel unless there is no such Pixel.
If you specify a Pixel that is at a border and has no Pixel to the left/right etc. then the returned value will be -1.

1. Clean up a digitized Picture, remove noise using an SPR-Script
In this Sample we will remove the noise from a picture using the SPR.
For this we will read the Pixel, check if it has only white neighbors, and in that case replace it with a white dot.

© 2023 Theo Gottwald * https://www.smart-package.com

Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode". Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.




```
' This Script will clean up the background of a digitized Writing
VAF.$$PIA=?exeloc\ULangK.png
VAF.$$PIB=?exeloc\ULangK1.png
ANA.Load|0|$$PIA
ANA.Show|0!
'ANA.AnalyzeColor|0|1
$$SRC=&HFFF5FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HEADAFE
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HE4F5FF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HC8C0E9
ANA.ReplaceColorT|0|$$SRC|10|$$COL
$$TIM=#dtime#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Rem_SngPix
CAL.$$TMA=#dsince#|i
DBP.Needed: $$TMA Seconds.
DMP.1
MBX.!
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
'============================================================

:Rem_SngPix
ANA.GetRes|0|$$XSZ|$$YSZ
PRT.Size=$$XSZ,$$YSZ
FOR.$$YPO|2|($$YSZ-1)
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|2|($$XSZ-1)
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
    JIV.$$PIX=&HFFFFFF|EndLoop

    ANA.gpc4s|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD
    ANA.gpc4d|0|$$XPO|$$YPO|$$PAE|$$PAF|$$PAG|$$PAH
    CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF
    IVV.$$ERG>5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
    :EndLoop
  NEX.
NEX.
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000853 0001 VAF.$$PIA=?exeloc\ULangK.png
014 0000000000000117 0001 VAF.$$PIB=?exeloc\ULangK1.png
015 0000000001035912 0001 ANA.Load|0|$$PIA
016 0000000020560973 0001 ANA.Show|0!
017 0000000000000061 0001
018 0000000000000894 0001 VAR.$$SRC=&HFFF5FF
019 0000000000000092 0001 VAR.$$COL=&HFFFFFF
020 0000000001940416 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
021 0000000000000034 0001
022 0000000000000417 0001 VAR.$$SRC=&HEADAFE
023 0000000002296598 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
024 0000000000000063 0001
025 0000000000000398 0001 VAR.$$SRC=&HE4F5FF
026 0000000001890570 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
027 0000000000000058 0001
028 0000000000000394 0001 VAR.$$SRC=&HC8C0E9
029 0000000002117531 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
030 0000000000000792 0001 VAR.$$TIM=#dtime#
031 0000000000000222 0001 GSB.Rem_SngPix
032 0000000000000214 0001 CAL.$$TMA=#dsince#|i
033 0000000000019735 0001 DBP.Needed: $$TMA Seconds.
034 0000000000000000 0001 DMP.1
035 0000000000000000 0000 MBX.!
036 0000000000000000 0000 ANA.Show|0!
037 0000000000000000 0000 ANA.Save|0|$$PIB
038 0000000000000000 0000 ENR.
039 0000000000000000 0000
040 0000000000000000 0000
041 0000000000000033 0001 :Rem_SngPix
042 0000000000003052 0001 ANA.GetRes|0|$$XSZ|$$YSZ
043 0000000022353444 0001 PRT.Size=$$XSZ,$$YSZ
044 0000000000000230 0562 FOR.$$YPO|2|($$YSZ-1)
045 0000000000003158 0561 PRT.$$YPO/$$YSZ
046 0000000000000244 496485 FOR.$$XPO|2|($$XSZ-1)
047 0000000000000321 495924 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
048 0000000000000106 495924 JIV.$$PIX=&HFFFFFF|EndLoop
049 0000000000000003 495924
050 0000000000000577 495924 ANA.gpc4s|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD
051 0000000000000601 495924 ANA.gpc4d|0|$$XPO|$$YPO|$$PAE|$$PAF|$$PAG|$$PAH
052 0000000000000975 495924 CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+(
F)+($$PAF=&HFFFFFF)+($$PAG=&HFFFFFF)+($$PAH=&HFFFFFF)
053 0000000000000082 495924 IVV.$$ERG>5
054 0000000000000000 0000 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
055 0000000000000004 495924 EIF.
056 0000000000000004 495924 :EndLoop
057 0000000000000006 496485 NEX.
058 0000000000000005 0562 NEX.
059 0000000000000000 0001 RET.
060 0000000000000000 0000
```

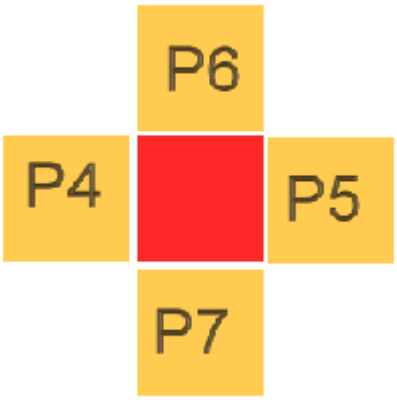
Syntax

ANA.GetPixelColor_4S|P1|P2|P3[|P4][|P5][|P6][|P7]
ANA.GPC4S|P1|P2|P3[|P4][|P5][|P6][|P7]

Parameter Explanation

P1 **-** ***Source*** Image-Register number
P2 **-** ***X - Center Pixel Coordinate X-Value***
P3 **-** ***Y - Center Pixel Coordinate Y-Value***
P4 **-** ***opt. Variable*** for returned Color-Value for the Left Pixel.
P5 **-** ***opt. Variable*** for returned Color-Value for the Right Pixel.
P6 **-** ***opt. Variable*** for returned Color-Value for the Upper Pixel.
P7 **-** ***opt. Variable*** for returned Color-Value for the Lower Pixel.
If you omit **P4** to **P7**, TOS is used.

*© 2023 Theo Gottwald * https://www.smart-package.com*



Example

```
See above.
```

Remarks

The `ANA.GetPixelColor_4S` is optimized for Speed. It will use only about 580 Ticks. Therefore the Robot can execute the command typically more then 27500 times per Second.

Limitations:

-

See also:

- ANA.GetPixelColor_4D 1629

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.4 ANA.GetPixelColor_5h

MiniRobotLanguage (MRL)

ANA.GetPixelColor_5H

Get the Color of 5 successive horizontal Pixels from an Image-Register (IR)

Intention

This command will read the color of 5 successive Pixels in an IR. Its an Option to Speed up Image-Processing.
With this command, the returned value is the 24-bit color of the Pixels.
You do not need to worry, if the resolution is not a multiple of 5, illegal Coordinates will just be ignored.

Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Free-Run Button

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

```
' This code will invert the left half of a Picture.
' We will use two technologies and you can compare the needed time.
'
VAF.$$PIA=?exeloc\Pics\Sarah_HS.png
NEF.$$PIA
  MBX.File not found.
EIF.
VAN.$$BMA=0
ANA.Load|$$BMA|$$PIA
ANA.GetRes|$$BMA|$$XRA|$$YRA
ANA.Show|$$BMA!
FOR.$$YPO|0|$$YRA
  'PRT.Line: $$YPO/$$YRA
  FOR.$$XPO|0|($$XRA/2)|5
    ' We get 5 Pixels and process them
    ANA.GetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
    CAL.$$COA=&HFFFFFF-$$COA
    CAL.$$COB=&HFFFFFF-$$COB
    CAL.$$COC=&HFFFFFF-$$COC
    CAL.$$COD=&HFFFFFF-$$COD
    CAL.$$COE=&HFFFFFF-$$COE
    ' We write the 5 Pixels back
    ANA.SetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
  NEX.
NEX.
VAN.$$TMC=#dsince#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.$$TMC used
' This Script will do the same, but process "Pixel by Pixel"
' This way it takes nearly twice the time.
ANA.Show|$$BMA!
ANA.Load|$$BMA|$$PIA
VAN.$$TMA=#dtime#
FOR.$$YPO|0|$$YRA
  PRT.Line: $$YPO/$$YRA
  FOR.$$XPO|0|($$XRA/2)
    ' Bild mit Kanten
    ANA.GetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
    CAL.$$COL=&HFFFFFF-$$COL
    ANA.SetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
  NEX.
NEX.
VAN.$$TMB=#dsince#
DBP.$$TMB used
DMP.6
ANA.Show|$$BMA!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

The Script used 6 Seconds to complete process the picture.

Here is the Speed-Dump:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
010 0000000000000284 0001 VAF.$$PIA=?exeloc\Pics\Sarah_HS.png
011 0000000000002092 0001 NEF.$$PIA
012 0000000000000000 0000 MBX.File not found.
013 0000000000000025 0001 EIF.
014 0000000000000004 0001
015 0000000000000123 0001 VAN.$$BMA=0
016 0000000000000003 0001
017 0000000000187995 0001 ANA.Load|$$BMA|$$PIA
018 0000000000000550 0001 ANA.GetRes|$$BMA|$$XRA|$$YRA
019 0000000008646981 0001 ANA.Show|$$BMA!
020 0000000000000104 0477 FOR.$$YPO|0|$$YRA
021 0000000000000003 0476
022 0000000000000201 15232 FOR.$$XPO|0|($$XRA/2)|5
023 0000000000000003 14756
024 0000000000000579 14756 ANA.GetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
025 0000000000000157 14756 CAL.$$COA=&HFFFFFF-$$COA
026 0000000000000156 14756 CAL.$$COB=&HFFFFFF-$$COB
027 0000000000000158 14756 CAL.$$COC=&HFFFFFF-$$COC
028 0000000000000158 14756 CAL.$$COD=&HFFFFFF-$$COD
029 0000000000000160 14756 CAL.$$COE=&HFFFFFF-$$COE
030 0000000000000511 14756 ANA.SetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
031 0000000000000005 15232 NEX.
032 0000000000000005 0477 NEX.
033 0000000000000127 0001 VAN.$$TMC=#dsince#
034 0000000000032934 0001 DBP.$$TMC used
035 0000000030328930 0001 ANA.Load|$$BMA!
036 0000000000239028 0001 ANA.Load|$$BMA|$$PIA
037 0000000000000215 0001 VAN.$$TMA=#dtime#
038 0000000000000131 0477 FOR.$$YPO|0|$$YRA
039 0000000000000800 0476 PRT.Line: $$YPO/$$YRA
040 0000000000000251 74256 FOR.$$XPO|0|($$XRA/2)
041 0000000000000003 73780
042 0000000000000378 73780 ANA.GetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
043 0000000000000215 73780 CAL.$$COL=&HFFFFFF-$$COL
044 0000000000000347 73780 ANA.SetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
045 0000000000000006 74256 NEX.
046 0000000000000006 0477 NEX.
047 0000000000000156 0001 VAN.$$TMB=#dsince#
048 0000000000000826 0001 DBP.$$TMB used
049 0000000000000000 0001 DMP.1
050 0000000000000000 0000 MBX.!
```

Syntax

ANA.GetPixelColor_5h|P1|P2|P3|P4|P5|P6|P7|P8

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - Variable for returned Color-Value of Pixel 1 at (X1,Y1)
P5 - Variable for returned Color-Value of Pixel 2 at (X1+1,Y1)
P6 - Variable for returned Color-Value of Pixel 3 at (X1+2,Y1)
P7 - Variable for returned Color-Value of Pixel 4 at (X1+3,Y1)
P8 - Variable for returned Color-Value of Pixel 5 at (X1+4,Y1)

Example

```
See above.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

The `ANA.GetPixelColor_5h` is optimized for Speed. It will use only about 612 Ticks. Therefore the Robot could read more then 31250*5 Pixels per Second.

Limitations:

-

See also:

- ANA.SetPixelColor_5h 1685

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.5 ANA.GetPixelColor_5v

MiniRobotLanguage (MRL)

ANA.GetPixelColor_5V

Get the Color of 5 successive vertical Pixels from an Image-Register (IR)

Intention

This command will read the color of 5 successive vertical Pixels in an IR. Its an Option to Speed up Image-Processing.
With this command, the returned value is the 24-bit color of the Pixels.
You do not need to worry, if the resolution is not a multiple of 5, illegal Coordinates will just be ignored.

Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Free-Run Button

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

```
' This code will invert the left half of a Picture.
'
VAF.$$PIA=?exeloc\Pics\Sarah_HS.png
NEF.$$PIA
  MBX.File not found.
EIF.
VAN.$$BMA=0
ANA.Load|$$BMA|$$PIA
ANA.GetRes|$$BMA|$$XRA|$$YRA
ANA.Show|$$BMA!
FOR.$$YPO|0|$$YRA|5
  PRT.Line: $$YPO/$$YRA
  FOR.$$XPO|0|($$XRA/2)
    ' Bild mit Kanten
    ANA.GetPixelColor_5V|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
    CAL.$$COA=&HFFFFFF-$$COA
    CAL.$$COB=&HFFFFFF-$$COB
    CAL.$$COC=&HFFFFFF-$$COC
    CAL.$$COD=&HFFFFFF-$$COD
    CAL.$$COE=&HFFFFFF-$$COE
    ANA.SetPixelColor|$$BMA|$$XPO|$$YPO|$$COA
    ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+1)|$$COB
    ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+2)|$$COC
    ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+3)|$$COD
    ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+4)|$$COE
  NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEX.
VAN.$$TMC=#dsince#
DBP.$$TMC used
ANA.Show|$$BMA!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

The Script used 6 Seconds to complete process the picture.

Here is the Speed-Dump:

```
010 0000000000000185 0001 VAF.$$PIA=?exeloc\Pics\Sarah_HS.png
011 0000000000001977 0001 NEF.$$PIA
012 0000000000000000 0000 MBX.File not found.
013 0000000000000024 0001 EIF.
014 0000000000000005 0001
015 0000000000000121 0001 VAN.$$BMA=0
016 0000000000000004 0001
017 0000000000173827 0001 ANA.Load|$$BMA|$$PIA
018 0000000000000469 0001 ANA.GetRes|$$BMA|$$XRA|$$YRA
019 0000000018274918 0001 ANA.Show|$$BMA!
020 0000000000000110 0097 FOR.$$YPO|0|$$YRA|5
021 0000000000000003 0096
022 0000000000000194 14976 FOR.$$XPO|0|($$XRA/2)
023 0000000000000002 14880
024 0000000000000580 14880 ANA.GetPixelColor_5V|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
025 0000000000000160 14880 CAL.$$COA=&HFFFFFF-$$COA
026 0000000000000151 14880 CAL.$$COB=&HFFFFFF-$$COB
027 0000000000000149 14880 CAL.$$COC=&HFFFFFF-$$COC
028 0000000000000151 14880 CAL.$$COD=&HFFFFFF-$$COD
029 0000000000000153 14880 CAL.$$COE=&HFFFFFF-$$COE
030 0000000000001204 14880 ANA.SetPixelColor|$$BMA|$$XPO|$$YPO|$$COA
031 0000000000000337 14880 ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+1)|$$COB
032 0000000000000336 14880 ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+2)|$$COC
033 0000000000000335 14880 ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+3)|$$COD
034 0000000000000339 14880 ANA.SetPixelColor|$$BMA|$$XPO|($$YPO+4)|$$COE
035 0000000000000005 14976 NEX.
036 0000000000000005 0097 NEX.
037 0000000000000133 0001 VAN.$$TMC=#dsince#
038 0000000000002533 0001 DBP.$$TMC used
039 0000000000000000 0001 DMP.1
040 0000000000000000 0000 MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.GetPixelColor_5V|P1|P2|P3|P4|P5|P6|P7|P8

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - Variable for returned Color-Value of Pixel 1 at (X1,Y1)
P5 - Variable for returned Color-Value of Pixel 2 at (X1,Y1+1)
P6 - Variable for returned Color-Value of Pixel 3 at (X1,Y1+2)
P7 - Variable for returned Color-Value of Pixel 4 at (X1,Y1+3)
P8 - Variable for returned Color-Value of Pixel 5 at (X1,Y1+4)

Example

```
See above.
```

Remarks

The `ANA.GetPixelColor_5V` is optimized for Speed. It will use only about 580 Ticks. Therefore the Robot could read more then 27500*5 Pixels per Second.

Limitations:

-

See also:

- ANA.SetPixelColor_5h 1685

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.6 ANA.GetPixelColor_8

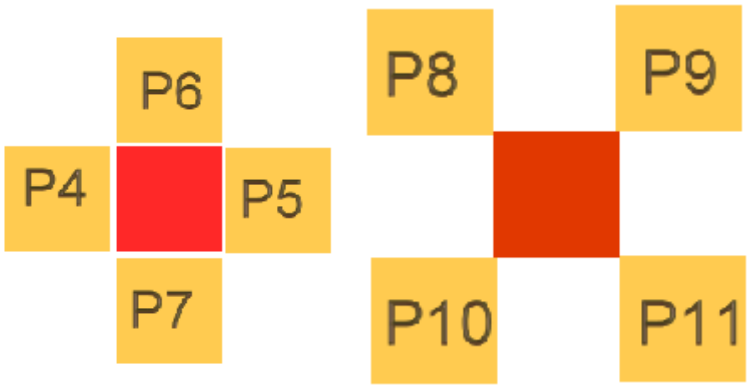
MiniRobotLanguage (MRL)

ANA.GetPixelColor_8

Get the Colors of all 8 Neighbor Pixels of a Single Pixel from an Image-Register (IR)

Intention

This command will read the colors of all 8 Pixels that are neighbor to a given Pixel in an IR.
Together with other Commands like `ANA.SetPixelColor`, It can be used to clean up Images and remove noise.
This command replaces 8 separate GetPixelColor-Commands (plus needed calculations) and is therefore an Option to speed
up Scripts. If a Pixel has no neighbor (Border Pixels) the result of that Color is -1.



The Colors of these Pixels are returned in Variables in P4 to P11.

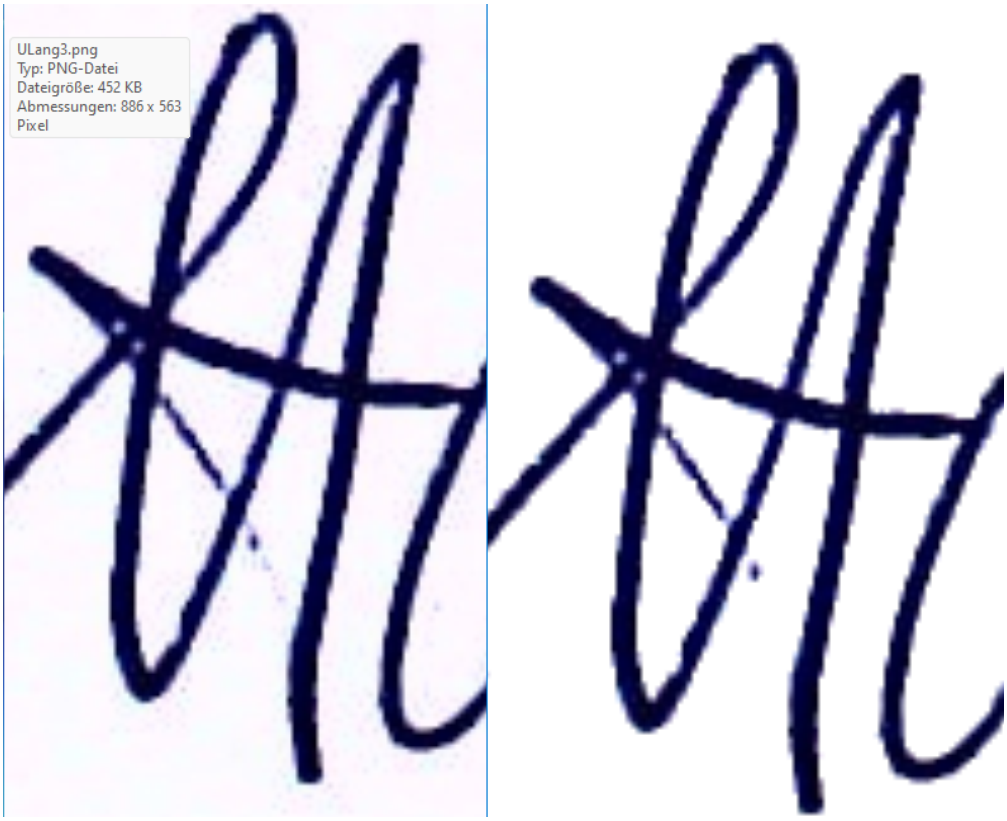
With this command, the returned value is always the 24-bit color of the Pixel unless there is no such Pixel.
If you specify a Pixel that is at a border and has no Pixel to the left/right etc. then the returned value will be -1.

1. Clean up a digitized Picture, remove noise using an SPR-Script
In this Sample we will remove the noise from a picture using the SPR.
For this we will read the Pixel, check if it has only white neighbors, and in that case replace it with a white dot.

Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.

© 2023 Theo Gottwald * https://www.smart-package.com

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.




```
'
'SPR Script-file: RemoveColors
'Purpose:
'Author: Theo Gottwald
'Creation date: 12-28-2021 at 20:11:00
'============================================================
'#EXE:?path\
'#SPI:ForceWrite

' This Script will remove a green background
' using special sorting functions.
' The Script is fast and will only take few seconds
VAF.$$PIA=?exeloc\ULangK.png
VAF.$$PIB=?exeloc\ULangK1.png
ANA.Load|0|$$PIA
ANA.Show|0!
'ANA.AnalyzeColor|0|1
$$SRC=&HFFF5FF
$$COL=&HFFFFFF
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HEADAFE
ANA.ReplaceColorT|0|$$SRC|10|$$COL

$$SRC=&HE4F5FF
ANA.ReplaceColorT|0|$$SRC|10|$$COL
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$SRC=&HC8C0E9
ANA.ReplaceColorT|0|$$SRC|10|$$COL
$$TIM=#dtime#
GSB.Rem_SngPix
CAL.$$TMA=#dsince#|i
DMP.1
MBX.Needed: $$TMA Seconds.
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
'============================================================

:Rem_SngPix
ANA.GetRes|0|$$XSZ|$$YSZ
PRT.Size=$$XSZ,$$YSZ
FOR.$$YPO|2|($$YSZ-1)
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|2|($$XSZ-1)
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
    JIV.$$PIX=&HFFFFFF|EndLoop
    ANA.gpc8|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD|$$PAE|$$PAF|$$PAG|$$PAH
    CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF
    IVV.$$ERG>5
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
    :EndLoop
  NEX.
NEX.
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000432 0001 VAF.$$PIA=?exeloc\ULangK.png
014 0000000000000053 0001 VAF.$$PIB=?exeloc\ULangK1.png
015 0000000000324768 0001 ANA.Load|0|$$PIA
016 0000000017534720 0001 ANA.Show|0!
017 0000000000000010 0001
018 0000000000000101 0001 VAR.$$SRC=&HFFF5FF
019 0000000000000041 0001 VAR.$$COL=&HFFFFFF
020 0000000000426034 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
021 0000000000000007 0001
022 0000000000000072 0001 VAR.$$SRC=&HEADAFE
023 0000000000408751 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
024 0000000000000009 0001
025 0000000000000074 0001 VAR.$$SRC=&HE4F5FF
026 0000000000403840 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
027 0000000000000009 0001
028 0000000000000077 0001 VAR.$$SRC=&HC8C0E9
029 0000000000395781 0001 ANA.ReplaceColorT|0|$$SRC|10|$$COL
030 0000000000000125 0001 VAR.$$TIM=#dtime#
031 0000000000000053 0001 GSB.Rem_SngPix
032 0000000000000166 0001 CAL.$$TMA=#dsince#|i
033 0000000000000003 0001
034 0000000000000000 0001 DMP.1
035 0000000000000000 0000 MBX.!
036 0000000000000000 0000 ANA.Show|0!
037 0000000000000000 0000 ANA.Save|0|$$PIB
038 0000000000000000 0000 ENR.
039 0000000000000000 0000
040 0000000000000000 0000
041 0000000000000006 0001 :Rem_SngPix
042 0000000000000358 0001 ANA.GetRes|0|$$XSZ|$$YSZ
043 0000000001155265 0001 PRT.Size=$$XSZ,$$YSZ
044 0000000000000150 0562 FOR.$$YPO|2|($$YSZ-1)
045 0000000000001702 0561 PRT.$$YPO/$$YSZ
046 0000000000000162 496485 FOR.$$XPO|2|($$XSZ-1)
047 0000000000000204 495924 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PIX
048 0000000000000079 495924 JIV.$$PIX=&HFFFFFF|EndLoop
049 0000000000000605 495924 ANA.gpc8|0|$$XPO|$$YPO|$$PAA|$$PAB|$$PAC|$$PAD|$$PAE|$$PAF|$$PAG|$$PAH
050 0000000000000760 495924 CAL.$$ERG=($$PAA=&HFFFFFF)+($$PAB=&HFFFFFF)+($$PAC=&HFFFFFF)+($$PAD=&HFFFFFF)
F)+($$PAF=&HFFFFFF)+($$PAG=&HFFFFFF)+($$PAH=&HFFFFFF)
051 0000000000000046 495924 IVV.$$ERG>5
052 0000000000000000 0000 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
053 0000000000000003 495924 EIF.
054 0000000000000003 495924 :EndLoop
055 0000000000000004 496485 NEX.
056 0000000000000004 0562 NEX.
057 0000000000000000 0001 RET.
```

Syntax

ANA.GetPixelColor_8|P1|P2|P3[|P4][|P5][|P6][|P7]
ANA.GPC8|P1|P2|P3[|P4][|P5][|P6][|P7]

Parameter Explanation

P1 - Source Image-Register number
P2 - X - Center Pixel Coordinate X-Value
P3 - Y - Center Pixel Coordinate Y-Value
P4 - opt. Variable for returned Color-Value for the Left Pixel.
P5 - opt. Variable for returned Color-Value for the Right Pixel.
P6 - opt. Variable for returned Color-Value for the Upper Pixel.
P7 - opt. Variable for returned Color-Value for the Lower Pixel.
P8 - opt. Variable for returned Color-Value for the Upper-Left Pixel.
P9 - opt. Variable for returned Color-Value for the Upper-Right Pixel.
P10 - opt. Variable for returned Color-Value for the Lower-Left Pixel.
P11 - opt. Variable for returned Color-Value for the Lower-RIght Pixel.
If you omit **P4** to **P7**, TOS is used.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
See above.
```

Remarks

The `ANA.GetPixelColor_8` is optimized for Speed. It will use only about 600 Ticks. Therefore the Robot can execute the command typically more then 27500 times per Second.

Limitations:

-

See also:

- ANA.GetPixelColor_4D
- ANA.GetPixelColor_4D

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.7 ANA.GetPixelRGB

MiniRobotLanguage (MRL)

ANA.GetPixelRGB

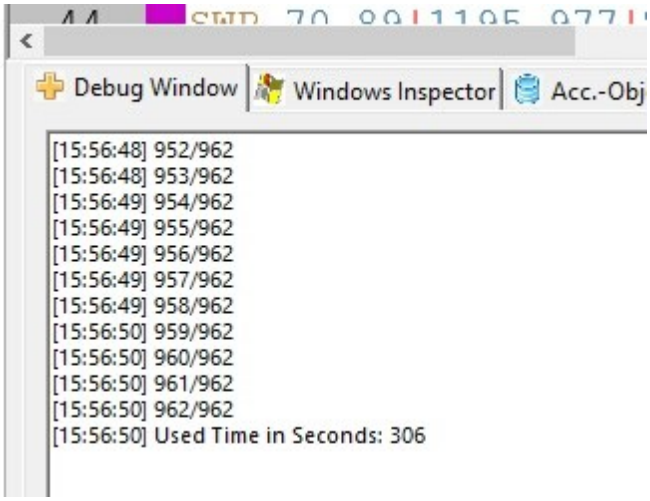
Get the Color of a Single Pixel from an Image-Register (IR) in three values for Red, Green and Blue

Intention

This command will read the color of a Single Pixel in an IR. It will return the Color in its Red, Green and Blue Component.
As we use 24-bit Colors, the returned color will be 0.255 (0-&HFF) for each Component Red, Green and Blue.

You can use this command together with other Commands like `ANA.SetPixelRGB`.

1. Remove Green Background on a picture using an SPR-Script
In this Sample we will remove the Background on a picture using the SPR.
For this we will read the Pixel, check if its mostly Green, and in that case replace
Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.



You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

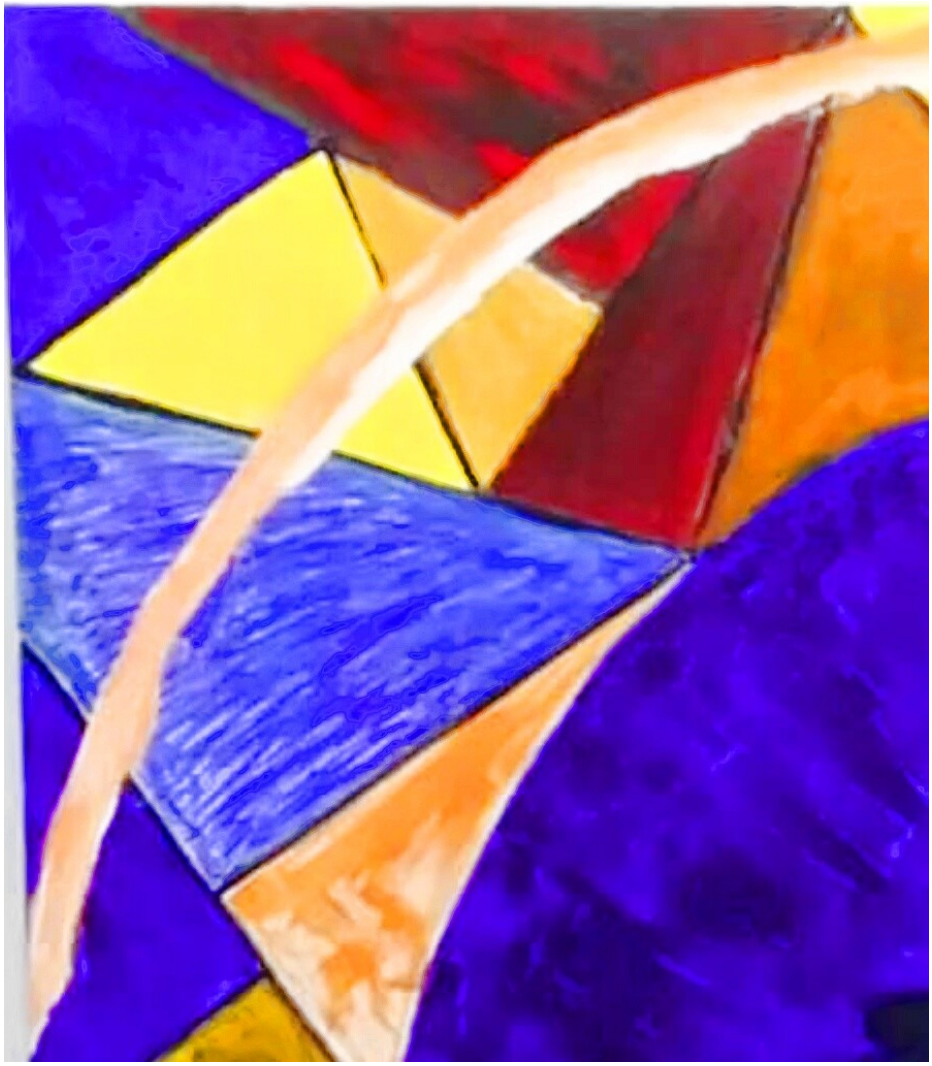
```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.Show|0!
ANA.Show|1!
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

This box is filled and it is drawn with inverted background color.
The second Part of the Script will draw 2500 random Lines.

```
012 0000000000000319 0001 VAN.$$TIM=#time#
013 0000000000000117 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000062 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000941118 0001 ANA.Load|0|$$PIA
016 0000000002422340 0001 ANA.Load|1|$$PIB
017 0000000000000588 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000183 0963 FOR.$$YPO|1|$$YSZ
019 0000000001338823 0962 DBP.$$YPO/$$YSZ
020 0000000000000224 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000765 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000657 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000110 615680 IVV.$$ERG>2
024 0000000000000365 422388 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000326 422388 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
026 0000000000000005 615680 EIF.
027 0000000000000009 616642 NEX.
028 0000000000000008 0963 NEX.
029 0000000000000430 0001 CAL.$$TIM=#time#-$$TIM
030 0000000000314415 0001 DBP.Used Time in Seconds: $$TIM
031 0000000000000000 0001 DMP.1
```

The Speed-dump shows that the Robot can read about 40.000 Pixels per second.

2. Color-Processing

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this Sample we will additionally change the Colors.For this we use the two Commands:

```
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
' This is just for example normally you would use ANA.GetPixelColor_RGB which co
        ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
' We Combine the Colors in away that Blue and Green is exchanged.
        ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
' And Set the Pixel
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
```

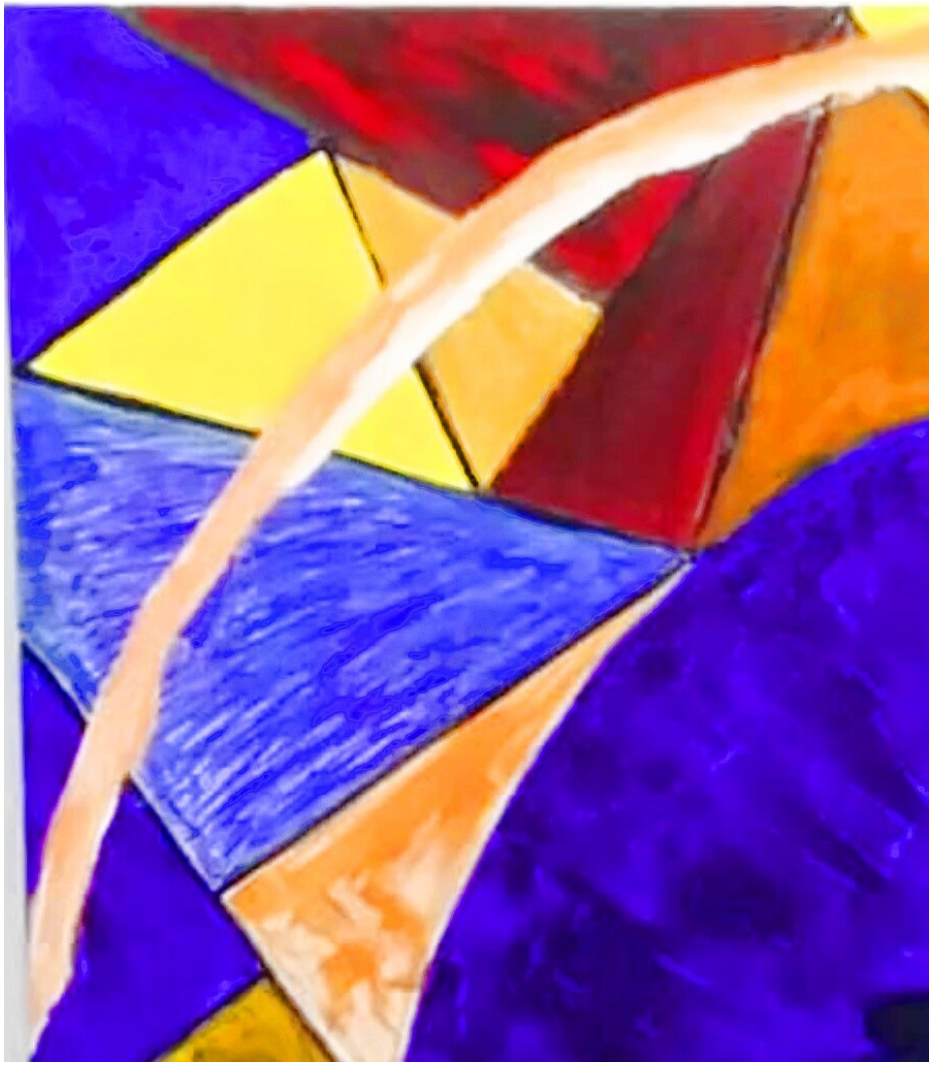
And then Paint the Picture.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

```
' Here is the Code.
' On my Computer it uses 5 MinutesProcessing time for a Picture of Medium size i
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
      ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

The Script used 306 Seconds to complete process the picture.

Here is the Speed-Dump:

```
012 0000000000000274 0001 VAN.$$TIM=#time#
013 0000000000000100 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000058 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000392529 0001 ANA.Load|0|$$PIA
016 0000000001986990 0001 ANA.Load|1|$$PIB
017 0000000000000568 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000126 0963 FOR.$$YPO|1|$$YSZ
019 0000000000060423 0962 DBP.$$YPO/$$YSZ
020 0000000000000146 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000504 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000493 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000082 615680 IVV.$$ERG>2
024 0000000000000383 422988 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000444 422988 ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
026 0000000000000370 422988 ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
027 0000000000000336 422988 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
028 0000000000000004 615680 EIF.
029 0000000000000006 616642 NEX.
030 0000000000000006 0963 NEX.
031 0000000000000310 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000230528 0001 DBP.Used Time in Seconds: $$TIM
033 0000000000000000 0001 DMP.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.GetPixelRGB[|P1][|P2][|P3][|P4][|P5][|P6]

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - opt. Variable for returned Color-Value RED. If omitted TOS is used.
P5 - opt. Variable for returned Color-Value GREEN. If omitted TOS is used.
P6 - opt. Variable for returned Color-Value BLUE. If omitted TOS is used.

The color values will always be in the range between 0-255. This is Hexadacimal: 0 - &HFF.

Example

```
See above.
```

Remarks

The `ANA.GetPixelRGB` is optimized for Speed. It will use only about 400 to 600 Ticks. Therefore the Robot could read more then 30000 Pixels per Second.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.8 ANA.SetPixelColor

MiniRobotLanguage (MRL)

ANA.SetPixelColor

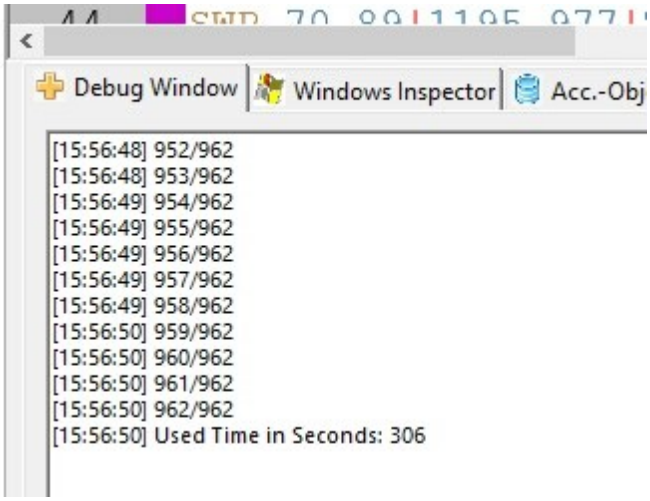
Set the Color of a Single Pixel from an Image-Register (IR)

Intention

This command will change the color of a Single Pixel in an IR. While that does not Sound much, it enables you to do unlimited complex Image processing,
Together with other Commands like `ANA.GetPixelColor`.

For Speed-Reasons, this command will never return anything on the TOS.

1. Remove Green Background on Image using an SPR-Script
In this Sample we will remove the Background on a Image using the SPR.
For this we will read the Pixel, check if its mostly Green, and in that case replace it with the colors taken from another Picture.
Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.



You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

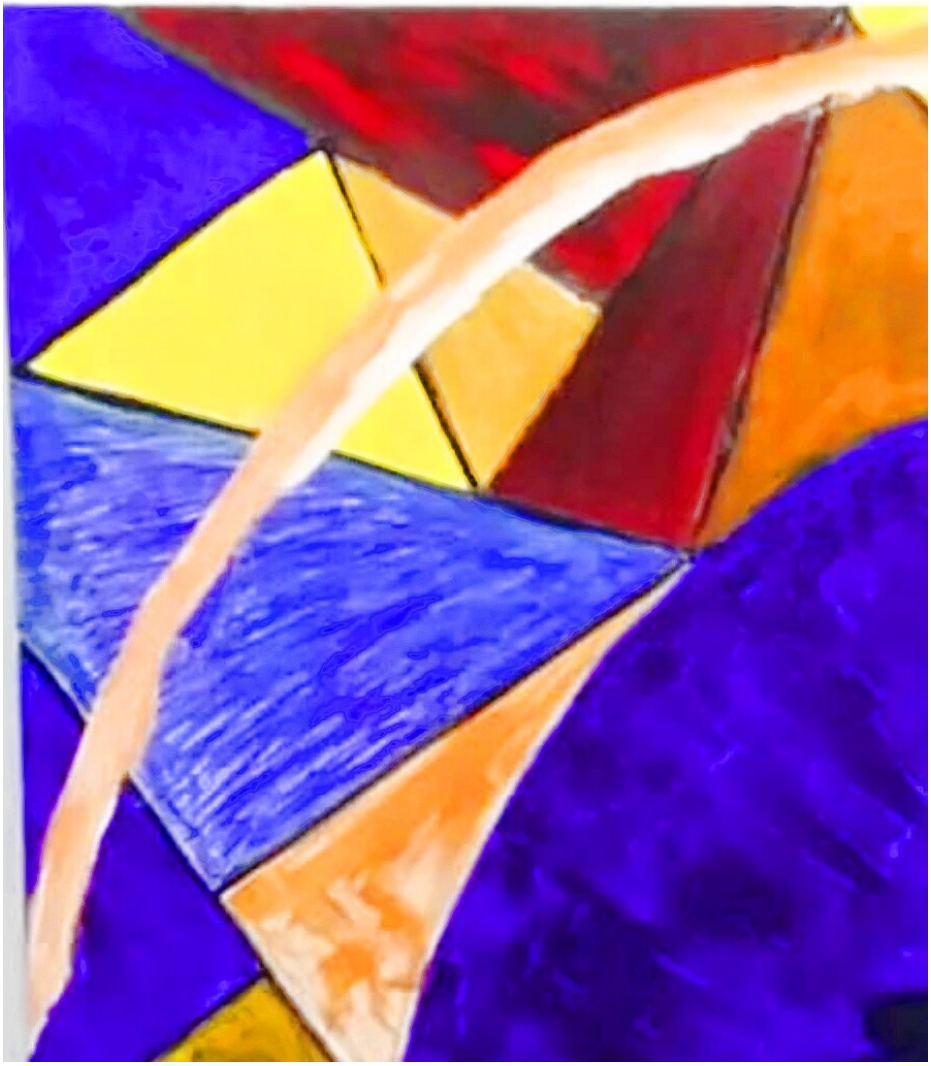
```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
VAF.$$PIB=?pattern\Mahakala2.png
VAF.$$PIC=?pattern\SarahB_640.png
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.Show|0!
ANA.Show|1!
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
    EIF.
  NEX.
NEX.
ANA.Save|0|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

This box is filled and it is drawn with inverted background color.
The second Part of the Script will draw 2500 random Lines.

```
012 0000000000000269 0001 VAN.$$TIM=#time#
013 0000000000000094 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000051 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000000055 0001 VAF.$$PIC=?pattern\SarahB_640.jpg
016 0000000000861741 0001 ANA.Load|0|$$PIA
017 0000000002200014 0001 ANA.Load|1|$$PIB
018 0000000000000524 0001 ANA.GetRes|0|$$XSZ|$$YSZ
019 0000000000000084 0963 FOR.$$YPO|1|$$YSZ
020 0000000000000003 0962
021 0000000000000099 616642 FOR.$$XPO|1|$$XSZ/2
022 0000000000000367 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
023 0000000000000392 615680 CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
024 0000000000000054 615680 IVV.$$ERG>2
025 0000000000000264 423959 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
026 0000000000000241 423959 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
027 0000000000000003 615680 EIF.
028 0000000000000004 616642 NEX.
029 0000000000000004 0963 NEX.
030 0000000000279013 0001 ANA.Save|0|$$PIC
031 0000000000000364 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000000005 0001
033 0000000000000000 0001 DMP.1
```

The Speed-dump shows that the Robot can read about 40.000 Pixels per second.

2. Color-Processing

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this Sample we will additionally change the Colors.For this we use the two Commands:

```
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
  ' This is just for example normally you would use ANA.GetPixelColor_RGB which co
        ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
  ' We Combine the Colors in away that Blue and Green is exchanged.
        ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
  ' And Set the Pixel
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
```

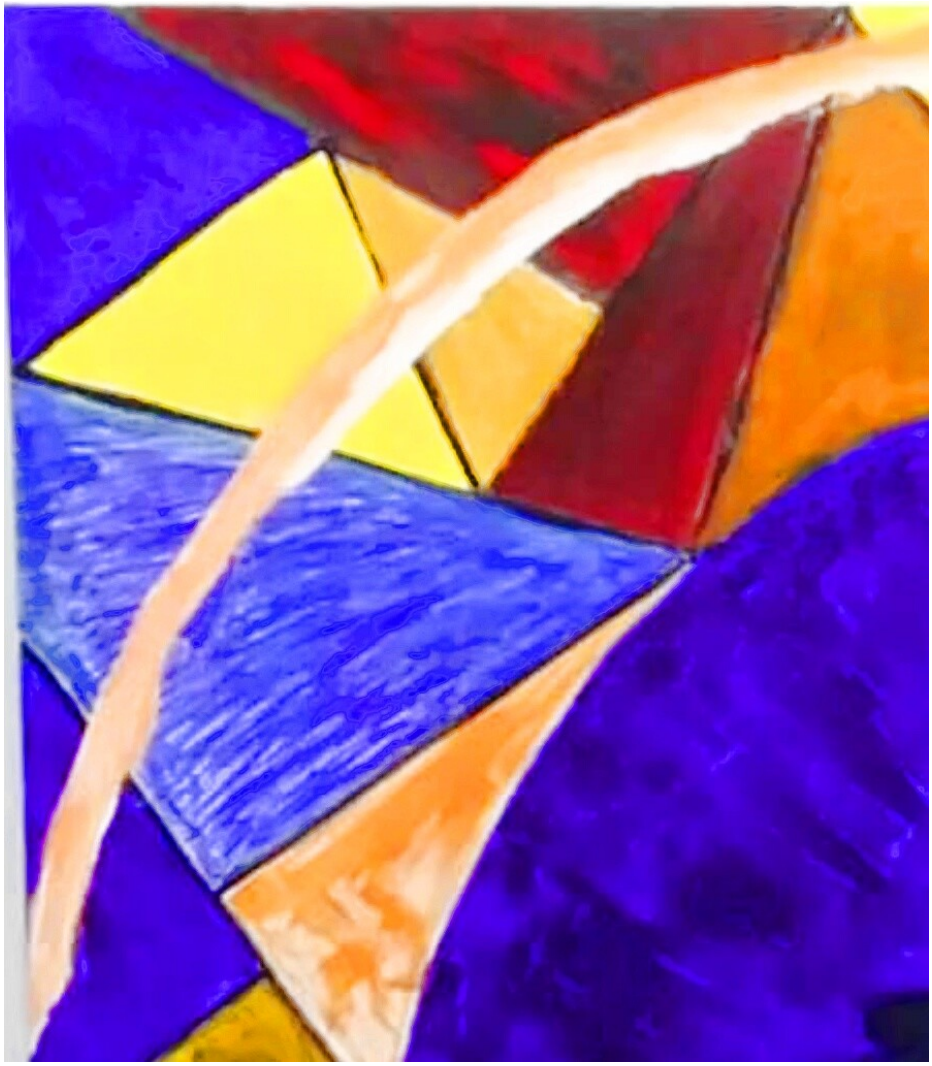
And then Paint the Picture.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

```
  ' Here is the Code.
  ' On my Computer it uses 5 MinutesProcessing time for a Picture of Medium size i
  VAN.$$TIM=#time#
  VAF.$$PIA=?pattern\Sarah_640.jpg
  VAF.$$PIB=?pattern\Mahakala2.png
  ANA.Load|0|$$PIA
  ANA.Load|1|$$PIB
  ANA.GetRes|0|$$XSZ|$$YSZ
  FOR.$$YPO|1|$$YSZ
    DBP.$$YPO/$$YSZ
    FOR.$$XPO|1|$$XSZ/2
      ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
      CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
      IVV.$$ERG>2
        ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
        ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
        ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
        ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
      EIF.
    NEX.
  NEX.
  CAL.$$TIM=#time#-$$TIM
  DBP.Used Time in Seconds: $$TIM
  DMP.1
  ANA.Show|0!
  ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

The Script used ~300 Seconds to complete process the picture.

Here is the Speed-Dump:

```
012 0000000000000274 0001 VAN.$$TIM=#time#
013 0000000000000100 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000000058 0001 VAF.$$PIB=?pattern\Mahakala2.png
015 0000000000392529 0001 ANA.Load|0|$$PIA
016 0000000001986990 0001 ANA.Load|1|$$PIB
017 0000000000000568 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000000126 0963 FOR.$$YPO|1|$$YSZ
019 0000000000060423 0962 DBP.$$YPO/$$YSZ
020 0000000000000146 616642 FOR.$$XPO|1|$$XSZ/2
021 0000000000000504 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
022 0000000000000493 615680 CAL.$$ERG=($$GRN>&H80)+($$GRN>$$BLU)+($$GRN>$$RED))
023 0000000000000082 615680 IVV.$$ERG>2
024 0000000000000383 422988 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PIX
025 0000000000000444 422988 ANA.Split_Color_RGB|$$PIX|$$COR|$$COG|$$COB
026 0000000000000370 422988 ANA.Combine_Color_RGB|$$COG|$$COR|$$COB|$$PIX
027 0000000000000336 422988 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
028 0000000000000004 615680 EIF.
029 0000000000000006 616642 NEX.
030 0000000000000006 0963 NEX.
031 0000000000000310 0001 CAL.$$TIM=#time#-$$TIM
032 0000000000230528 0001 DBP.Used Time in Seconds: $$TIM
033 0000000000000000 0001 DMP.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

2. Background or Object-Processing

In this Sample we will just fill the background with white color.
The Script takes 207 Second on my computer.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\Sarah_640.jpg
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
    IVV.$$ERG>2
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
    EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
EXO.?path\
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

This is the result of the operation

```
012 0000000000000294 0001 VAN.$$TIM=#time#
013 0000000000000113 0001 VAF.$$PIA=?pattern\Sarah_640.jpg
014 0000000000370472 0001 ANA.Load|0|$$PIA
015 0000000000000566 0001 ANA.GetRes|0|$$XSZ|$$YSZ
016 0000000000000131 0963 FOR.$$YPO|1|$$YSZ
017 0000000000184554 0962 DBP.$$YPO/$$YSZ
018 0000000000000154 616642 FOR.$$XPO|1|$$XSZ/2
019 0000000000000527 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
020 0000000000000513 615680 CAL.$$ERG=($$GRN>&H60)+($$GRN>$$BLU)+($$GRN>$$RED))
021 0000000000000089 615680 IVV.$$ERG>2
022 0000000000000289 423959 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFFFF
023 0000000000000004 615680 EIF.
024 0000000000000006 616642 NEX.
025 0000000000000006 0963 NEX.
026 0000000000000330 0001 CAL.$$TIM=#time#-$$TIM
027 0000000000118643 0001 DBP.Used Time in Seconds: $$TIM
028 0000000000000000 0001 DMP.1
```

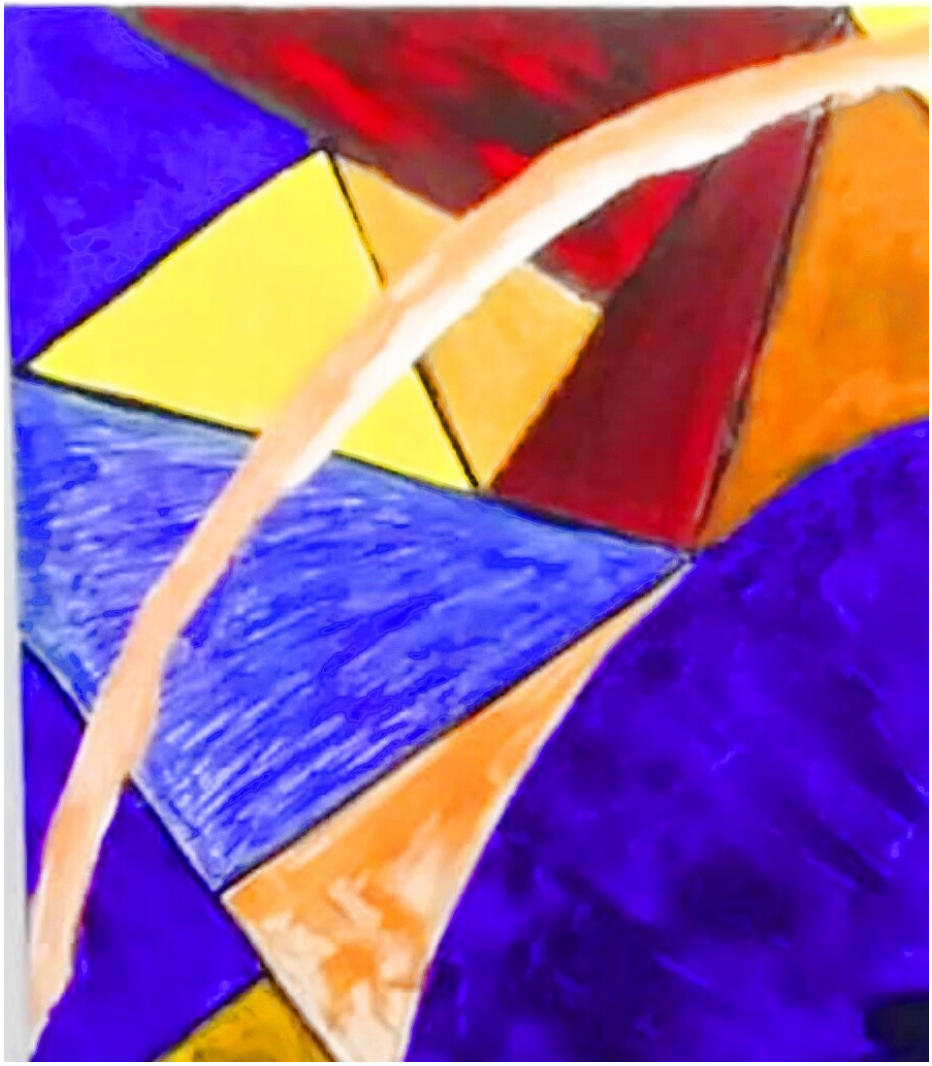
Here you can see the Speed-Dump of the Script.

3. Remove Object from Background

In this Sample we will have the Background, and a Person on the Background.
This Script will then Paint the Object yellow.

*© 2023 Theo Gottwald * https://www.smart-package.com*

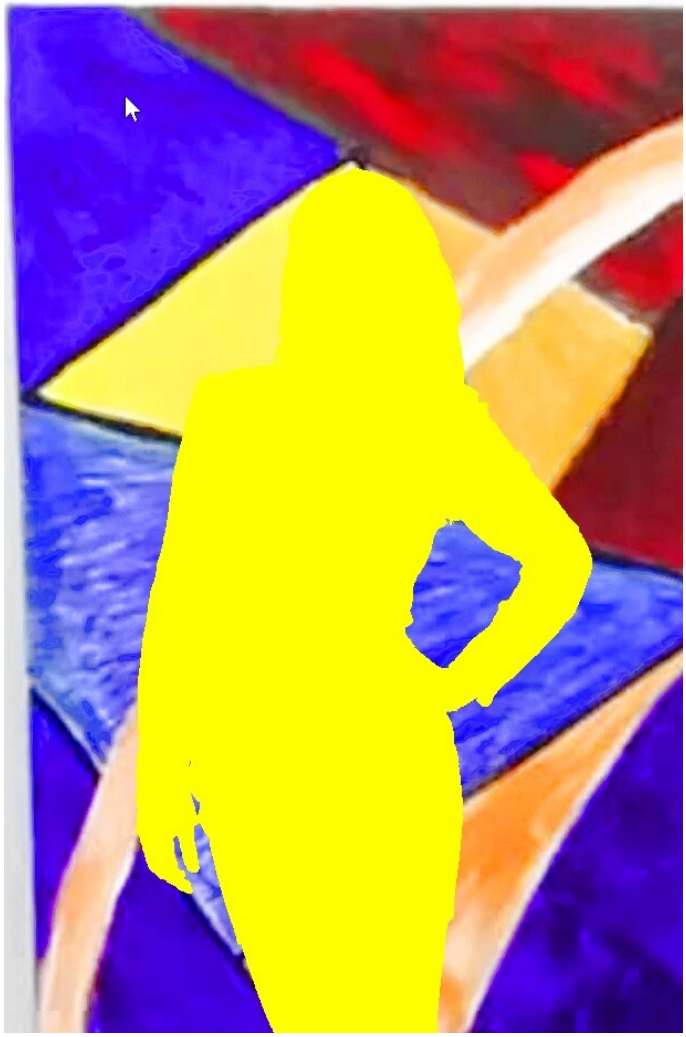
© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com



```
013 0000000000000190 0001 VAN.$$TIM=#time#
014 0000000000000088 0001 VAF.$$PIA=?pattern\SarahB_640.png
015 0000000000000062 0001 VAF.$$PIB=?pattern\Mahakala2.png
016 0000000000000055 0001 VAF.$$PIC=?pattern\SarahR_640.jpg
017 0000000000584825 0001 ANA.Load|0|$$PIA
018 0000000001989721 0001 ANA.Load|1|$$PIB
019 0000000000000409 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000000078 0963 FOR.$$YPO|1|$$YSZ
021 0000000000000003 0962
022 0000000000000091 616642 FOR.$$XPO|1|$$XSZ/2
023 0000000000000224 615680 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
024 0000000000000235 615680 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
025 0000000000000074 615680 IVV.$$PCA=$$PCB
026 0000000000000217 424450 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
027 0000000000000003 615680 ELS.
028 0000000000000002 191230
029 0000000000000199 191230 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
030 0000000000000003 615680 EIF.
031 0000000000000004 616642 NEX.
032 0000000000000004 0963 NEX.
033 0000000000231127 0001 ANA.Save|0|$$PIC
034 0000000000000392 0001 CAL.$$TIM=#time#-$$TIM
035 0000000000000005 0001
036 0000000000000000 0001 DMP.1
```

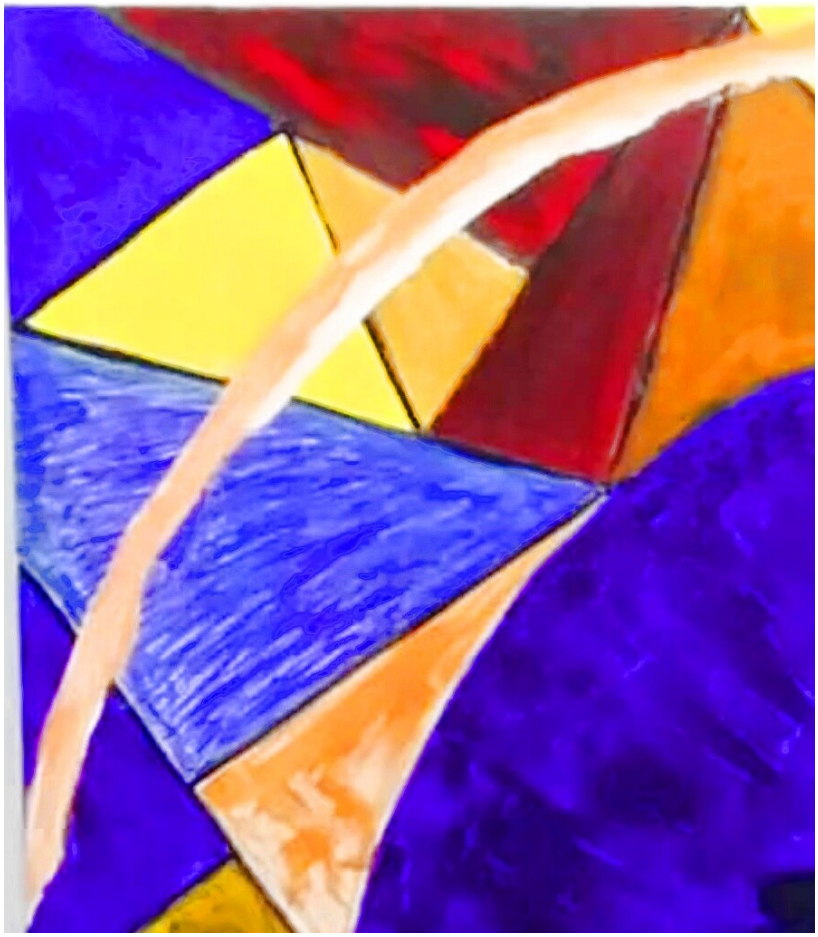
If we just

*© 2023 Theo Gottwald * https://www.smart-package.com*

4. Extract Object from known Background
If we do just change 1 Letter in the Script, we get the inverted result.
The Script will then extract the Person from the Baclground.
Hint: This is a very Simple Script it will only work with Pictures that are uncompressed like "-bmp", or ".png" etc.
Using other Formats you will need to apply a Tolerance.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.png
VAF.$$PIB=?pattern\Mahakala2.png
VAF.$$PIC=?pattern\SarahR_640.jpg
ANA.Load|0|$$PIA
ANA.Load|1|$$PIB
ANA.GetRes|0|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ/2
    ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
    ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
    NVV.$$PCA=$$PCB
      ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
    ELS.
    ' Paint it Yellow
      ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
    EIF.
  NEX.
NEX.
ANA.Save|0|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

Here we use a known Background to extract Sarah from the second Picture into the third Picture with yellow background.

```
013 0000000000000179 0001 VAN.$$TIM=#time#
014 0000000000000084 0001 VAF.$$PIA=?pattern\SarahB_640.png
015 0000000000000060 0001 VAF.$$PIB=?pattern\Mahakala2.png
016 0000000000000057 0001 VAF.$$PIC=?pattern\SarahR_640.jpg
017 0000000000583742 0001 ANA.Load|0|$$PIA
018 0000000002013220 0001 ANA.Load|1|$$PIB
019 0000000000000617 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000000083 0963 FOR.$$YPO|1|$$YSZ
021 0000000000000003 0962
022 0000000000000097 616642 FOR.$$XPO|1|$$XSZ/2
023 0000000000000242 615680 ANA.getpixelcolor|0|$$XPO|$$YPO|$$PCA
024 0000000000000249 615680 ANA.getpixelcolor|1|$$XPO|$$YPO|$$PCB
025 0000000000000080 615680 NVV.$$PCA=$$PCB
026 0000000000000236 191721 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PCA
027 0000000000000004 615680 ELS.
028 0000000000000003 423959
029 0000000000000217 423959 ANA.Setpixelcolor|0|$$XPO|$$YPO|&HFFFF00
030 0000000000000003 615680 EIF.
031 0000000000000004 616642 NEX.
032 0000000000000004 0963 NEX.
033 0000000000229073 0001 ANA.Save|0|$$PIC
034 0000000000000379 0001 CAL.$$TIM=#time#-$$TIM
035 0000000000000005 0001
036 0000000000000000 0001 DMP.1
```

Here is the Speed-Dump for this Sample.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

ANA.SetPixelColor[|P1][|P2][|P3][|P4]

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - Variable for Color-Value.

Example

```
See above.
```

Remarks

The `ANA.SetPixelColor` is optimized for Speed. It will use only about 200 to 336 Ticks. Therefore the Robot could write more then 47000 Pixels per Second. In EXE Mode its even faster then in Free-Run Mode (from Editor).

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.9 ANA.SetPixelColor_5h

MiniRobotLanguage (MRL)

ANA.SetPixelColor_5H

Set the Color of 5 successive horizontal Pixels from an Image-Register (IR)

Intention

This command will set/change the color of 5 successive Pixels in an IR. Its an Option to Speed up Image-Processing.
With this command, the returned value is the 24-bit color of the Pixels.
You do not need to worry, if the resolution is not a multiple of 5, illegal Coordinates will just be ignored.

Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Free-Run Button

You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

```
' This code will invert the left half of a Picture.
' We will use two technologies and you can compare the needed time.
'
VAF.$$PIA=?exeloc\Pics\Sarah_HS.png
NEF.$$PIA
  MBX.File not found.
EIF.
VAN.$$BMA=0
ANA.Load|$$BMA|$$PIA
ANA.GetRes|$$BMA|$$XRA|$$YRA
ANA.Show|$$BMA!
FOR.$$YPO|0|$$YRA
  'PRT.Line: $$YPO/$$YRA
  FOR.$$XPO|0|($$XRA/2)|5
    ' We get 5 Pixels and process them
    ANA.GetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
    CAL.$$COA=&HFFFFFF-$$COA
    CAL.$$COB=&HFFFFFF-$$COB
    CAL.$$COC=&HFFFFFF-$$COC
    CAL.$$COD=&HFFFFFF-$$COD
    CAL.$$COE=&HFFFFFF-$$COE
    ' We write the 5 Pixels back
    ANA.SetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
  NEX.
NEX.
VAN.$$TMC=#dsince#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.$$TMC used
' This Script will do the same, but process "Pixel by Pixel"
' This way it takes nearly twice the time.
ANA.Show|$$BMA!
ANA.Load|$$BMA|$$PIA
VAN.$$TMA=#dtime#
FOR.$$YPO|0|$$YRA
  PRT.Line: $$YPO/$$YRA
  FOR.$$XPO|0|($$XRA/2)
    ' Bild mit Kanten
    ANA.GetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
    CAL.$$COL=&HFFFFFF-$$COL
    ANA.SetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
  NEX.
NEX.
VAN.$$TMB=#dsince#
DBP.$$TMB used
DMP.6
ANA.Show|$$BMA!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

The Script used 6 Seconds to complete process the picture.

Here is the Speed-Dump:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
010 0000000000000284 0001 VAF.$$PIA=?exeloc\Pics\Sarah_HS.png
011 0000000000002092 0001 NEF.$$PIA
012 0000000000000000 0000 MBX.File not found.
013 0000000000000025 0001 EIF.
014 0000000000000004 0001
015 0000000000000123 0001 VAN.$$BMA=0
016 0000000000000003 0001
017 0000000000187995 0001 ANA.Load|$$BMA|$$PIA
018 0000000000000550 0001 ANA.GetRes|$$BMA|$$XRA|$$YRA
019 0000000008646981 0001 ANA.Show|$$BMA!
020 0000000000000104 0477 FOR.$$YPO|0|$$YRA
021 0000000000000003 0476
022 0000000000000201 15232 FOR.$$XPO|0|($$XRA/2)|5
023 0000000000000003 14756
024 0000000000000579 14756 ANA.GetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
025 0000000000000157 14756 CAL.$$COA=&HFFFFFF-$$COA
026 0000000000000156 14756 CAL.$$COB=&HFFFFFF-$$COB
027 0000000000000158 14756 CAL.$$COC=&HFFFFFF-$$COC
028 0000000000000158 14756 CAL.$$COD=&HFFFFFF-$$COD
029 0000000000000160 14756 CAL.$$COE=&HFFFFFF-$$COE
030 0000000000000511 14756 ANA.SetPixelColor_5h|$$BMA|$$XPO|$$YPO|$$COA|$$COB|$$COC|$$COD|$$COE
031 0000000000000005 15232 NEX.
032 0000000000000005 0477 NEX.
033 0000000000000127 0001 VAN.$$TMC=#dsince#
034 0000000000032934 0001 DBP.$$TMC used
035 0000000030328930 0001 ANA.Show|$$BMA!
036 0000000000239028 0001 ANA.Load|$$BMA|$$PIA
037 0000000000000215 0001 VAN.$$TMA=#dtime#
038 0000000000000131 0477 FOR.$$YPO|0|$$YRA
039 0000000000000800 0476 PRT.Line: $$YPO/$$YRA
040 0000000000000251 74256 FOR.$$XPO|0|($$XRA/2)
041 0000000000000003 73780
042 0000000000000378 73780 ANA.GetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
043 0000000000000215 73780 CAL.$$COL=&HFFFFFF-$$COL
044 0000000000000347 73780 ANA.SetPixelColor|$$BMA|$$XPO|$$YPO|$$COL
045 0000000000000006 74256 NEX.
046 0000000000000006 0477 NEX.
047 0000000000000156 0001 VAN.$$TMB=#dsince#
048 0000000000000826 0001 DBP.$$TMB used
049 0000000000000000 0001 DMP.1
050 0000000000000000 0000 MBX.!
```

Syntax

ANA.SetPixelColor_5h|P1|P2|P3|P4|P5|P6|P7|P8

Parameter Explanation

P1 - Source Image-Register number
P2 - X1 - Upper Left Coordinate X-Value
P3 - Y1 - Upper Left Coordinate Y-Value
P4 - Variable for Color-Value of Pixel 1 at (X1,Y1)
P5 - Variable for Color-Value of Pixel 2 at (X1+1,Y1)
P6 - Variable for Color-Value of Pixel 3 at (X1+2,Y1)
P7 - Variable for Color-Value of Pixel 4 at (X1+3,Y1)
P8 - Variable for Color-Value of Pixel 5 at (X1+4,Y1)

Example

See above.

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The `ANA.SetPixelColor_5h` is optimized for Speed. It will use only about 350 Ticks. Therefore the Robot could set more then 300.000 Pixels per Second.

Limitations:

-

See also:

- ANA.GetPixelColor_5h 1639
-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.14.10 ANA.SetPixelRGB

MiniRobotLanguage (MRL)

ANA.SetPixelRGB

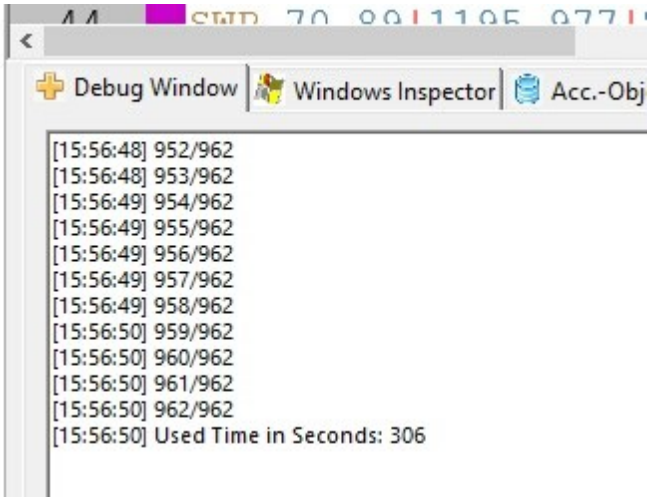
Set the Color of a Single Pixel from an Image-Register (IR) using the RGB-Values

Intention

This command will change the color of a Single Pixel in an IR. Instead of specifying the Color-Value as one Number, you can specify the R,G and B-Part here separately. Use this command together with other Commands like `ANA.GetPixelRGB`.

For Speed-Reasons, this command will never return anything on the TOS.

1. Doing Color-Changes/Corrections
In this Sample we will change the Colors in the Picture.
For this we will read the Pixel, check if its mostly Green, and in that case replace it with the colors taken from another Picture.
Due to the use of a nested Loop going "Line by Line" through all Pixels, it is recommended to start the script in the Editor using "Freerun Mode".
Even then the Script still may use several Minutes to finish. You will see in the Editor-Debug-Window the current Line and the last line.



You will not see the Robot Painting unless the Show command is been called which leads to an Update in the Show-Window.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIC=?pattern\SarahC_640.jpg
ANA.Load|0|$$PIA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.GetRes|0|$$XSZ|$$YSZ
ANA.New|1|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    CAL.$$RED=$$RED*1.1-70
    CAL.$$GRN=$$GRN*1.1-70
    CAL.$$BLUE=$$BLU*1.1-70
    ANA.SetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
  NEX.
NEX.
ANA.Save|1|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|1!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

->

© 2023 Theo Gottwald * https://www.smart-package.com

This is the Original picture. . The is the picture with the colors changed.

```
013 0000000000000203 0001 VAN.$$TIM=#time#
014 0000000000000089 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000057 0001 VAF.$$PIC=?pattern\SarahC_640.jpg
016 0000000000461693 0001 ANA.Load|0|$$PIA
017 0000000000000406 0001 ANA.GetRes|0|$$XSZ|$$YSZ
018 0000000000024014 0001 ANA.New|1|$$XSZ|$$YSZ
019 0000000000000084 0963 FOR.$$YPO|1|$$YSZ
020 0000000000000004 0962
021 0000000000000088 616642 FOR.$$XPO|1|$$XSZ
022 0000000000000379 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
023 0000000000000171 615680 CAL.$$RED=$$RED*1.1-70
024 0000000000000161 615680 CAL.$$GRN=$$GRN*1.1-70
025 0000000000000024 615680 CAL.$$BLUE=$$BLU*1.1-70
026 0000000000000347 615680 ANA.SetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
027 0000000000000004 616642 NEX.
028 0000000000000004 0963 NEX.
029 0000000000249490 0001 ANA.Save|1|$$PIC
030 0000000000000385 0001 CAL.$$TIM=#time#-$$TIM
031 0000000000000005 0001
032 0000000000000000 0001 DMP.1
```

The Speed-dump shows that the Robot can do such correction very fast, yet simple changes on the whole picture can be done faster using other commands.

2. Picture Mixing

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this Sample we will mix 2 Pictures depending on the relation between Blue and Red.
In the SPR you can use virtually unlimited Graphics Sources to be combined.

.

```
VAN.$$TIM=#time#
VAF.$$PIA=?pattern\SarahB_640.jpg
VAF.$$PIC=?pattern\SarahC_640.jpg
ANA.Load|0|$$PIA
ANA.Copy|0|1
ANA.Grey|1
ANA.GetRes|0|$$XSZ|$$YSZ
ANA.New|1|$$XSZ|$$YSZ
FOR.$$YPO|1|$$YSZ
  DBP.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    IVV.$$BLU>$$RES
      ANA.GetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
    EIF.
    ANA.SetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
  NEX.
NEX.
ANA.Save|1|$$PIC
CAL.$$TIM=#time#-$$TIM
DBP.Used Time in Seconds: $$TIM
DMP.1
ANA.Show|1!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

The Script used ~99 Seconds to complete process the picture. If you change the IVV. to NVV. Command, the result will be the third picture or when using $$BLU and $$GRN the fourth picture.

Here is the Speed-Dump:

```
013 0000000000000261 0001 VAN.$$TIM=#time#
014 0000000000000124 0001 VAF.$$PIA=?pattern\SarahB_640.jpg
015 0000000000000094 0001 VAF.$$PIC=?pattern\SarahC_640.jpg
016 0000000000467509 0001 ANA.Load|0|$$PIA
017 0000000000017641 0001 ANA.Copy|0|1
018 0000000000129814 0001 ANA.Grey|1
019 0000000000000360 0001 ANA.GetRes|0|$$XSZ|$$YSZ
020 0000000000114887 0001 ANA.New|2|$$XSZ|$$YSZ
021 0000000000000081 0963 FOR.$$YPO|1|$$YSZ
022 0000000000000004 0962
023 0000000000000083 616642 FOR.$$XPO|1|$$XSZ
024 0000000000000359 615680 ANA.GetPixelRGB|0|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
025 0000000000000076 615680 IVV.$$BLU>$$RED
026 0000000000000363 225727 ANA.GetPixelRGB|1|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
027 0000000000000003 615680 EIF.
028 0000000000000326 615680 ANA.SetPixelRGB|2|$$XPO|$$YPO|$$RED|$$GRN|$$BLU
029 0000000000000005 616642 NEX.
030 0000000000000004 0963 NEX.
031 0000000000395490 0001 ANA.Save|2|$$PIC
032 0000000000000417 0001 CAL.$$TIM=#time#-$$TIM
033 0000000000000005 0001
034 0000000000000000 0001 DMP.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.SetPixelRGB[|P1][|P2][|P3][|P4][|P5][|P6]

Parameter Explanation

P1 - Target Image-Register number
P2 - X1 - Upper Left Coordinat X-Value
P3 - Y1 - Upper Left Coordinat Y-Value
P4 - Variable for Color-Value RED.
P5 - Variable for Color-Value GREEN.
P6 - Variable for Color-Value BLUE.

The color values will always be in the range between 0-255.
This is Hexadacimal: 0 - &HFF.

Example

```
See above.
```

Remarks

The `ANA.SetPixelColor` is optimized for Speed. It will use only about 200 to 336 Ticks. Therefore the Robot could write more then 47000 Pixels per Second. In EXE Mode its even faster then in Free-Run Mode (from Editor).

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.11 ICC./NCC. - If Color Condition

MiniRobotLanguage (MRL)

ICC.If Color Condition
NCC.If NOT Color Condition

Get a Pixel from an IR and test the Pixel-Color-Value about R,G and B values

Intention

These commands will take 2 Color-Values for R,G and B value. The first in the "Minimum" and the second is the "Maximum".
So the Colour of the specified Pixel must be inside this RGB Range then the Condition will evaluate to "true".

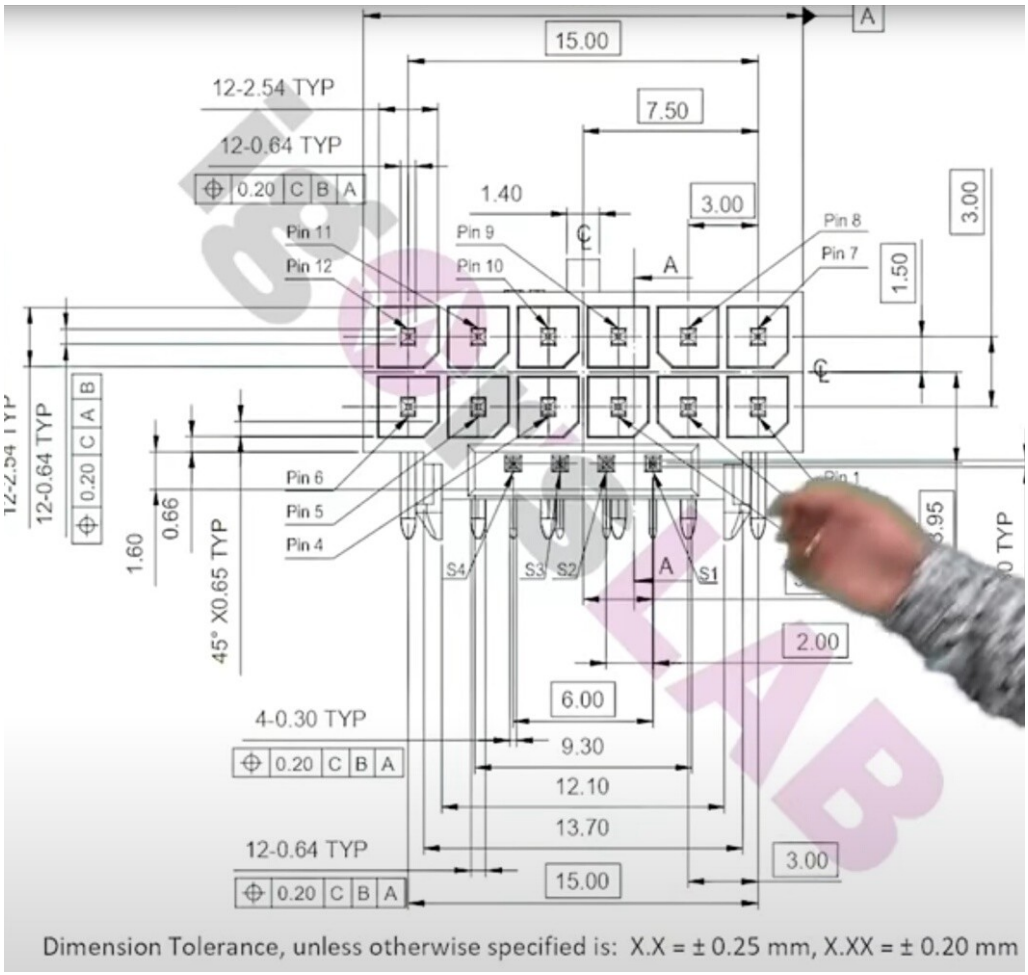
So the Command does this:

1. Read the specified Pixel from the IR.
2. Split the Pixelcolor into R,G and B values
3. Check if all 3 values (R,G and B) are inside the given range

If that is the case then the ICC. will evaluate "true" and NCC: will evaluate "false".

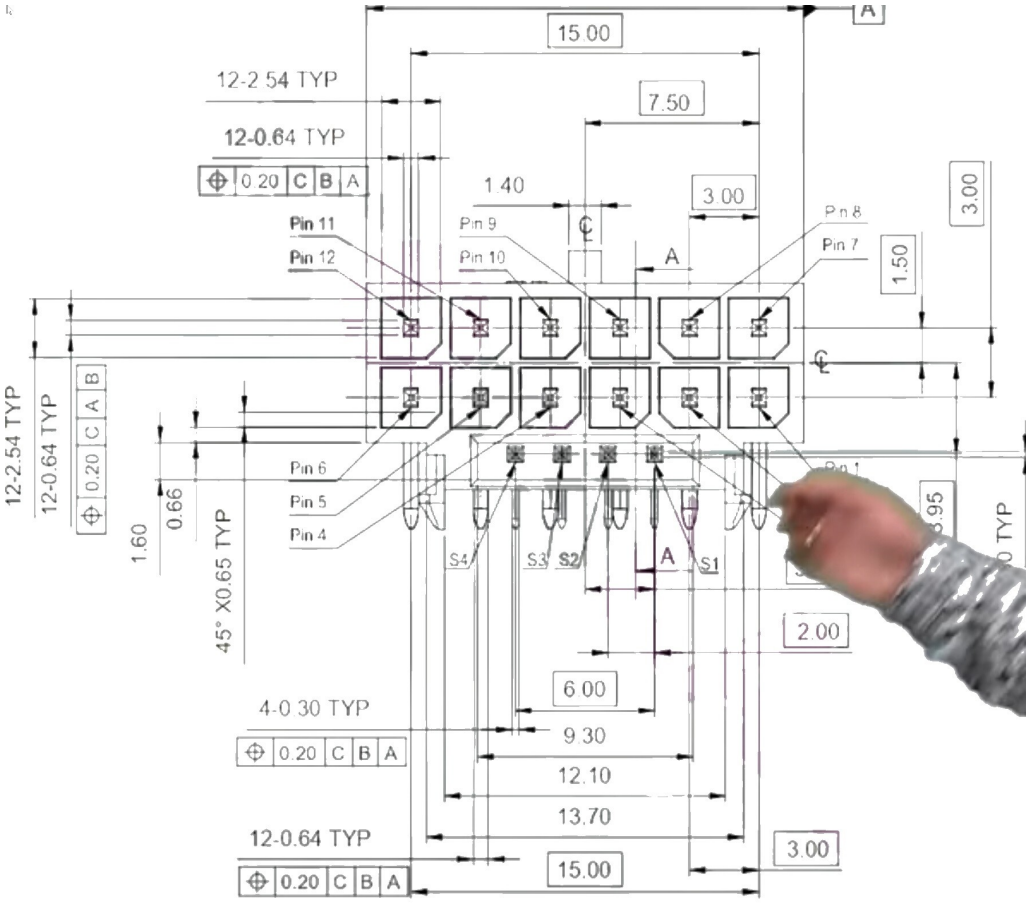
Here is a Sample Script that will remove unwanted Artefacts - for example here a watermark - from a picture.

© 2023 Theo Gottwald * https://www.smart-package.com



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

And here is the Script:

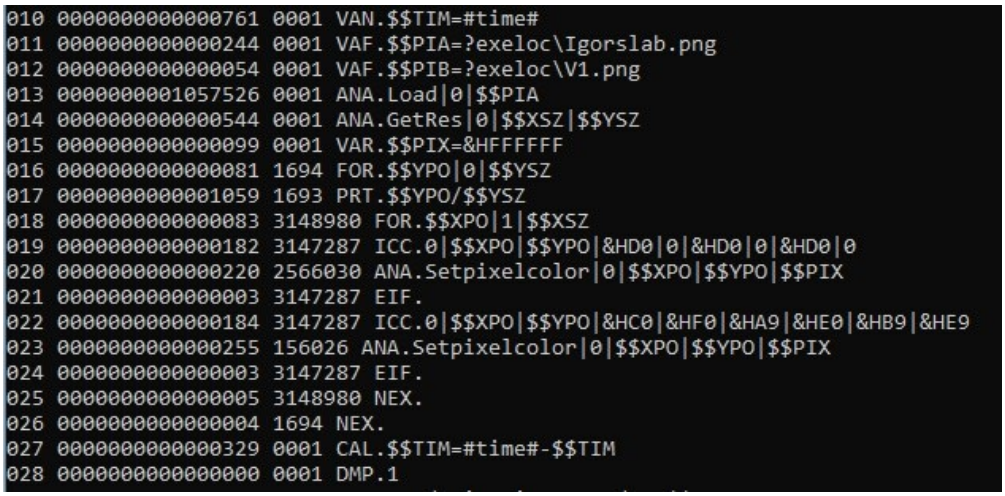
Pixelprocessing: "Pixel by Pixel" using the ICC.-Command.
The structure of this Script is easy to understand and it can be modified for any case.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ICC.0|$$XPO|$$YPO|&HD0|0|&HD0|0|&HD0|0
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
  EIF.
  ICC.0|$$XPO|$$YPO|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
  EIF.
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Speed-Dump, total Time used: 309 Seconds:



```
010 0000000000000761 0001 VAN.$$TIM=#time#
011 0000000000000244 0001 VAF.$$PIA=?exeloc\Igorslab.png
012 0000000000000054 0001 VAF.$$PIB=?exeloc\V1.png
013 0000000001057526 0001 ANA.Load|0|$$PIA
014 0000000000000544 0001 ANA.GetRes|0|$$XSZ|$$YSZ
015 0000000000000099 0001 VAR.$$PIX=&HFFFFFF
016 0000000000000081 1694 FOR.$$YPO|0|$$YSZ
017 0000000000001059 1693 PRT.$$YPO/$$YSZ
018 0000000000000083 3148980 FOR.$$XPO|1|$$XSZ
019 0000000000000182 3147287 ICC.0|$$XPO|$$YPO|&HD0|0|&HD0|0|&HD0|0
020 0000000000000220 2566030 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
021 0000000000000003 3147287 EIF.
022 0000000000000184 3147287 ICC.0|$$XPO|$$YPO|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9
023 0000000000000255 156026 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
024 0000000000000003 3147287 EIF.
025 0000000000000005 3148980 NEX.
026 0000000000000004 1694 NEX.
027 0000000000000329 0001 CAL.$$TIM=#time#-$$TIM
028 0000000000000000 0001 DMP.1
```

```
' You can use the Color Condition Registers to work with less Parameters inside
' this may speed up the process.
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
SRG.all|&HD0|0|&HD0|0|&HD0|0|0
SRG.all|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|1
FOR.$$YPO|0|$$YSZ
  PRT.! $$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ICC.0|$$XPO|$$YPO|0
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
  EIF.
  ICC.0|$$XPO|$$YPO|1
  ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
EIF.
NEX.
NEX.
DMP.1
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

Speed-Dump, total Time used: 216 Seconds:

© 2023 Theo Gottwald * https://www.smart-package.com

```
014 0000000000000807 0001 VAN.$$TIM=#time#
015 0000000000000637 0001 VAF.$$PIA=?exeloc\Igorslab.png
016 0000000000000056 0001 VAF.$$PIB=?exeloc\V1.png
017 0000000001042243 0001 ANA.Load|0|$$PIA
018 0000000000000548 0001 ANA.GetRes|0|$$XSZ|$$YSZ
019 0000000000000101 0001 VAR.$$PIX=&HFFFFFF
020 0000000000000423 0001 SRG.all|&HD0|0|&HD0|0|&HD0|0|0
021 0000000000000175 0001 SRG.all|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|1
022 0000000000000076 1694 FOR.$$YPO|0|$$YSZ
023 0000000000001829 1693 PRT.! $$YPO/$$YSZ
024 0000000000000079 3148980 FOR.$$XPO|1|$$XSZ
025 0000000000000137 3147287 ICC.0|$$XPO|$$YPO|0
026 0000000000000203 2566030 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
027 0000000000000003 3147287 EIF.
028 0000000000000142 3147287 ICC.0|$$XPO|$$YPO|1
029 0000000000000234 156026 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
030 0000000000000003 3147287 EIF.
031 0000000000000004 3148980 NEX.
032 0000000000000004 1694 NEX.
033 0000000000000000 0001 DMP.1
034 0000000000000000 0000 CAL.$$TIM=#time#-$$TIM
035 0000000000000000 0000 MBX.Used Time in Seconds: $$TIM
036 0000000000000000 0000 ANA.Show|0!
037 0000000000000000 0000 ANA.Save|0|$$PIB
038 0000000000000000 0000 ENR.
```

Syntax

```
ICC.P1|P2|P3|P4|P5|P6|P7|P8|P9
  ... ELS.   ..EIF
NCC.P1|P2|P3|P4|P5|P6|P7|P8|P9
  ... ELS.   ..EIF
or you can use CCR (Color
Condition Register)
ICC.P1|P2|P3|P4   ...
ELS.   ..EIF
NCC.P1|P2|P3|P4   ...
ELS.   ..EIF
```

Parameter Explanation

P1 *- Image Register Number*
P2 *- X-Coordinate for the Pixel to read in the Image-Register*
P3 *- Y-Coordinate for the Pixel to read in the Image-Register*

Syntax I
P4 /P5 *- (numeric) Min./Max. Values for Red, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).*

© 2023 Theo Gottwald * https://www.smart-package.com

P6 /P7 *- (numeric) Min./Max. Values for Green, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).*
P8 /P9 *- (numeric) Min./Max. Values for Blue, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).*

Syntax II
P4 *- opt. Color Condition Register* Number (default 0). The Color Condition Register must be set using the SRG. - Command

Example

```
See above.
```

Remarks

The ICC.-Commands is fast. It will use only about 185 Ticks per Pixel. Therefore the Robot could execute it more then >85000 times per Second.

Limitations:

-

See also:

- RIC. Replace If Color Condition

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.12 RCC. - Replace If Color Condition

RCC. - Replace If Color Condition for IR **Previous** 1088 **Top** 41 **Next** 1109

MiniRobotLanguage (MRL)

RCC.If Color Condition for IR

Process Block of Pixels or complete IR
Get a Pixel from an IR and test the Pixel-Color-Value about R,G and B values, if it matches the Condition, set the Pixel with a given color.

Intention

This Command is the optimized Version of RIC. It will not only process 1 Pixel but a Block of Pixels or the Complete IR in one Strike.
These command will take 2 Color-Values for R,G and B value. The first in the "Minimum" and the second is the "Maximum".
So the Colour of the specified Pixel must be inside this RGB Range then the Condition will evaluate to "true" and the Command will change the color of that Pixel to another color.

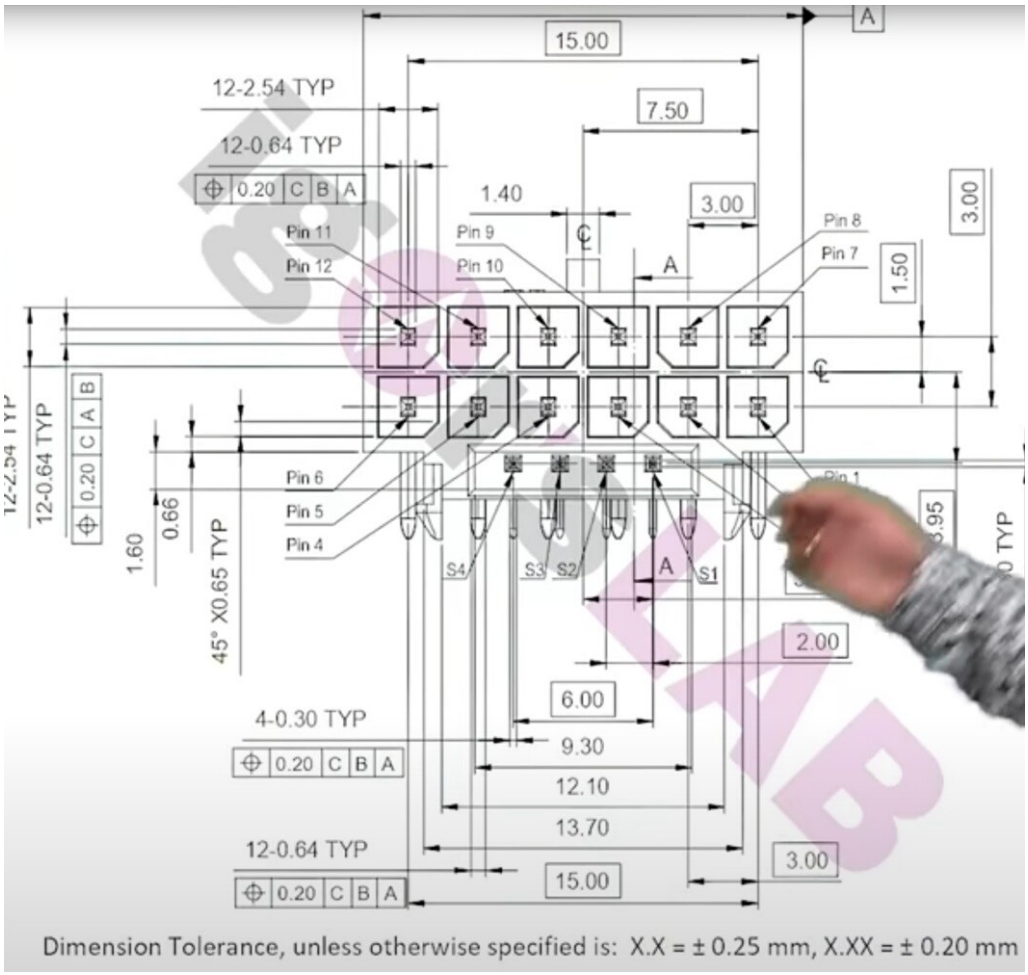
So the Command does this for the specified Block or the complete IR:

1. Read the specified Pixel from the IR.
2. Split the Pixelcolor into R,G and B values
3. Check if all 3 values (R,G and B) are inside the given range

If that is the case then the RIC. will evaluate "true" and directly change the color of that pixel to the specified color..

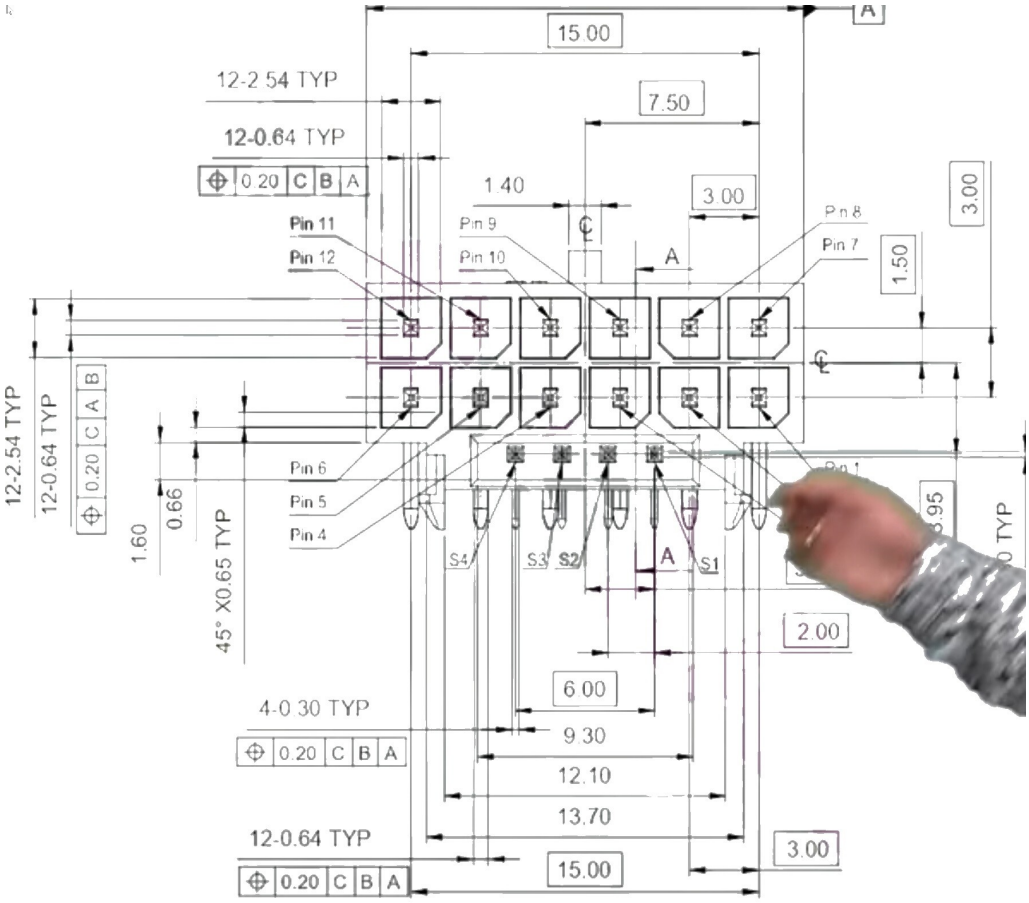
Here is a Sample Script that will remove unwanted Artefacts - for example here a watermark - from a picture.

*© 2023 Theo Gottwald * https://www.smart-package.com*



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

And here is the Script:

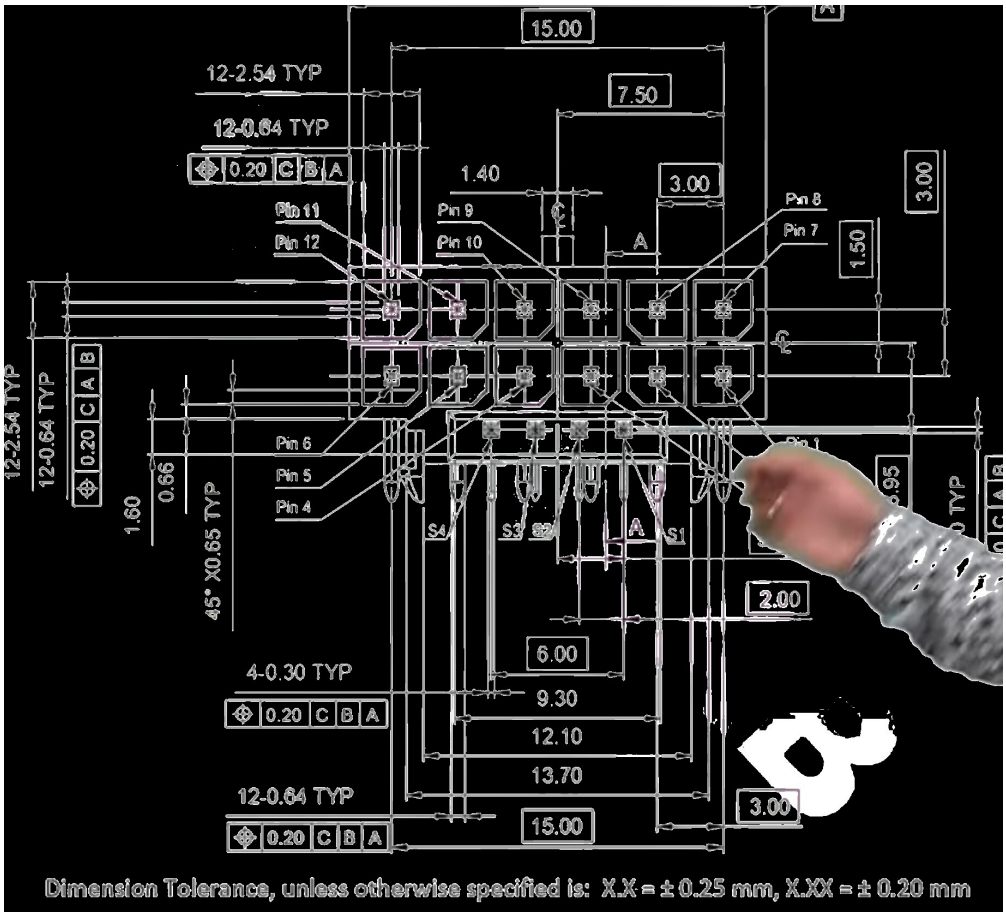
Pixelprocessing: "Complete IR in one Strike".
The structure of this Script is easy to understand and it can be modified for any case.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
RCC.0|&HD0|0|&HD0|0|&HD0|0|$$PIX
RCC.0|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
CAL.$$TIM=#time#-$$TIM
DMP.1
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

Speed-Dump:

© 2023 Theo Gottwald * https://www.smart-package.com

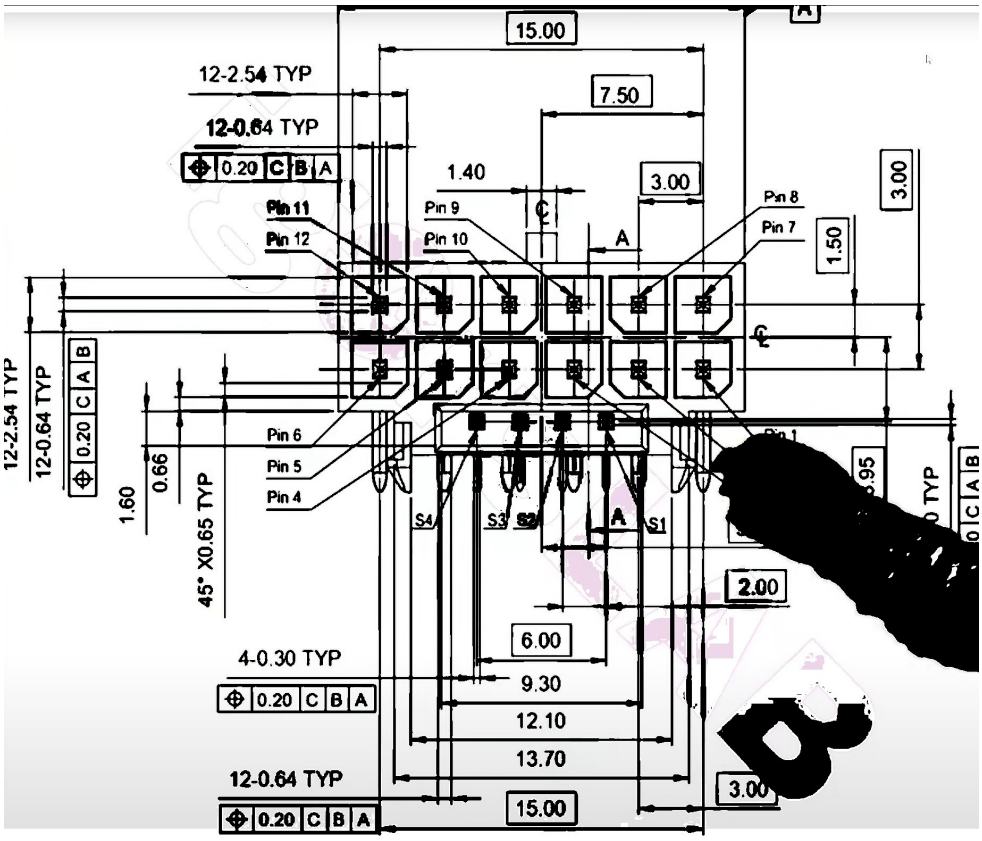
```
012 0000000000000627 0001 VAN.$$TIM=#time#
013 0000000000000175 0001 VAF.$$PIA=?exeloc\Igorslab.png
014 0000000000000063 0001 VAF.$$PIB=?exeloc\V1.png
015 0000000000939955 0001 ANA.Load|0|$$PIA
016 0000000000000694 0001 ANA.GetRes|0|$$XSZ|$$YSZ
017 0000000000000119 0001 VAR.$$PIX=&HFFFFFF
018 0000000000644295 0001 RCC.0|&HD0|0|&HD0|0|&HD0|0|$$PIX
019 0000000000501838 0001 RCC.0|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
020 0000000000000677 0001 CAL.$$TIM=#time#-$$TIM
021 0000000000000000 0001 DMP.1
```




```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
RCC.0|&HD0|0|&HD0|0|&HD0|0|0
RCC.0|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
CAL.$$TIM=#time#-$$TIM
DMP.1
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

*© 2023 Theo Gottwald * https://www.smart-package.com*

Using the ! - (Negate Result) Operator, you get such Results.
If this Operator is set then RCC. will only set the Pixel-Color if the condition is NOT True.

```
' Using the ! - Operator with RCC.
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
RCC.!0|&HD0|0|&HD0|0|&HD0|0|0
RCC.0|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

More Complex Sample

Here on the Left you can see the Original Picture with sort of watermartks
and on the right the picture after the processing with the SPR.

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Here on the Left you can see the Original Picture with sort of watermartks and on the right the picture after the processing with the SPR.

In this example we remove watermarks on the white and on the black Parts.
The same Script has been used on both pictures.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\BTC2.jpg
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
RCC.0|&HD0|0|&HD0|0|&HD0|0|$$PIX
RCC.0|&H0|&H6d|&H0|&H69|&H0|&H6a|0
RCC.0|&H5c|&H80|&H58|&H80|&H59|&H80|0
' Stronger black:
'RCC.0|&H16|&HAF|&H16|&HAF|&H16|&HAF|0
CAL.$$TIM=#time#-$$TIM
DMP.1
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```

Syntax

RCC.[!]P1|P2|P3|P4|P5|P6|P7|P8[|P9][|P10][|P11][|P12]

Parameter Explanation

P1 - Image Register Number - if prefixed with an "!" then the Result of the Operation will be negated
P2 /P3 - (numeric) Min./Max. Values for Red, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).
P4/P5 - (numeric) Min./Max. Values for Green, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).
P6 /P7 - (numeric) Min./Max. Values for Blue, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).
P8 - Color that will be used if condition is true
P9 - opt. X-Start-Position
P10 - opt. X-End-Position, if 0 then XMax is used
P11 - opt. Y-Start-Position
P12 - opt. Y-End-Position, if 0 then YMax is used

Using the ! - (Negate Result) Operator in P1, you get other Results.
If this Operator is set then RCC. will only set the Pixel-Color if the condition is NOT True.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
See above.
```

Remarks

The RCC.-Commands is optimized for Speed. It will use only about ~5 Ticks per Pixel. Therefore the Robot can process a full IR in just 1-2 seconds.

Limitations:

-

See also:

- ICC. - If Color Condition 1700
- RCC. - Replace If Color Condition 1706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.13 RIC. - Replace Pixel If Color Condition

MiniRobotLanguage (MRL)

RIC.If Color Condition

Get a Pixel from an IR and test the Pixel-Color-Value about R,G and B values, if it matches the Condition, set the Pixel with a given color.

Intention

These command will take 2 Color-Values for R,G and B value. The first in the "Minimum" and the second is the "Maximum".
So the Colour of the specified Pixel must be inside this RGB Range then the Condition will evaluate to "true" and the Command will change the color of that Pixel to another color.

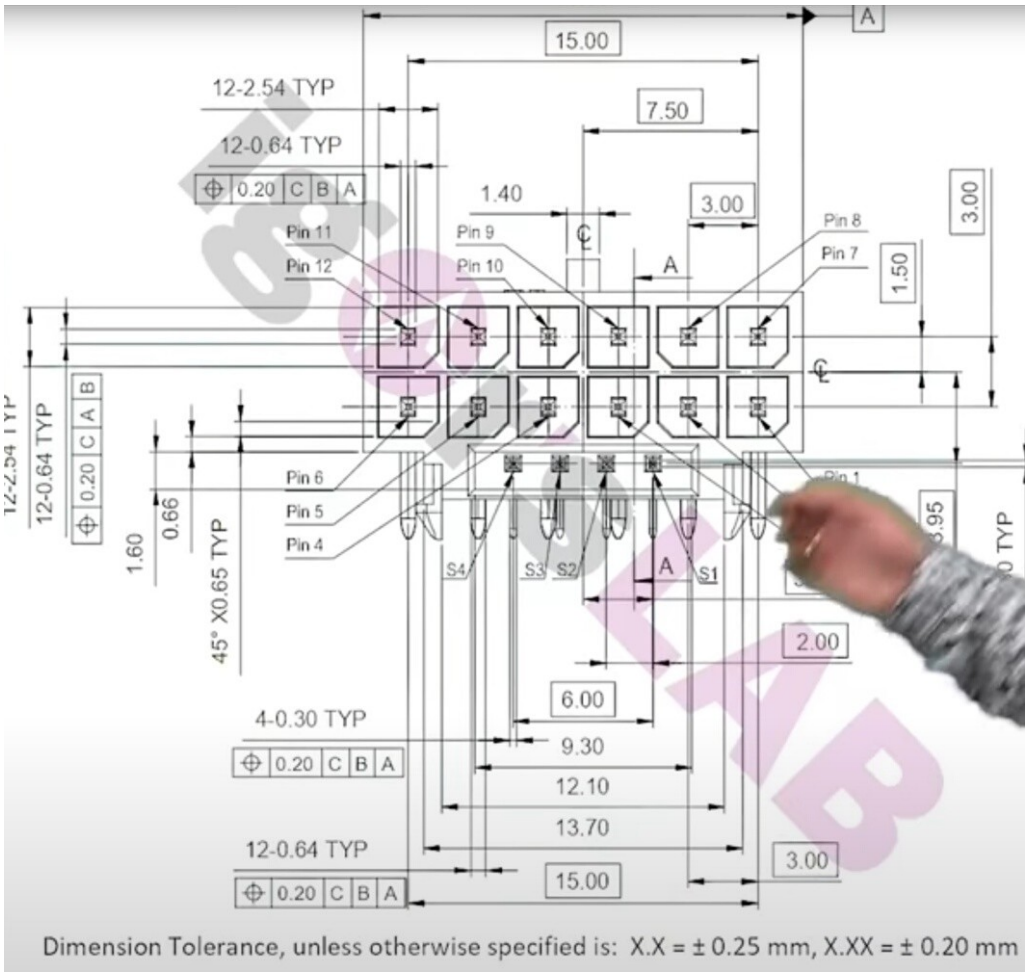
So the Command does this:

1. Read the specified Pixel from the IR.
2. Split the Pixelcolor into R,G and B values
3. Check if all 3 values (R,G and B) are inside the given range

If that is the case then the RIC. will evaluate "true" and directly cahnge the color of that pixel to the specified color..

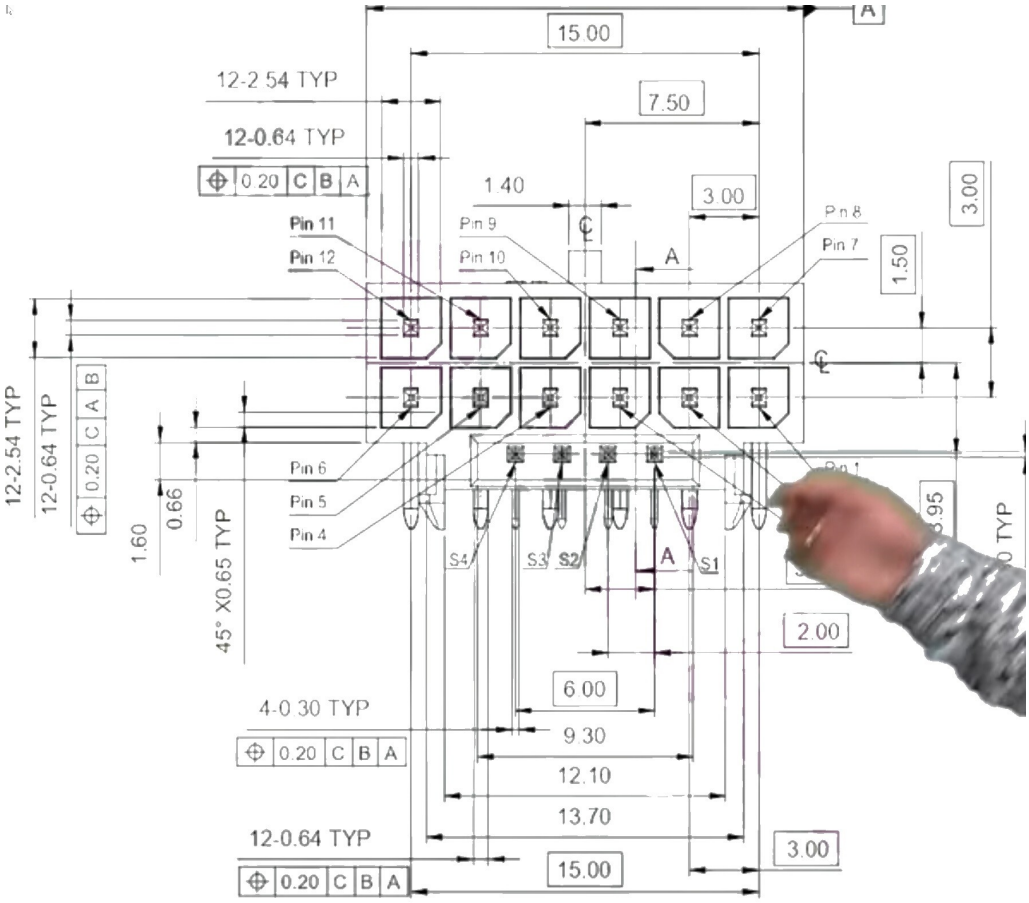
Here is a Sample Script that will remove unwanted Artefacts - for example here a watermark - from a picture.

*© 2023 Theo Gottwald * https://www.smart-package.com*



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

And here is the Script:

Pixelprocessing: "Pixel by Pixel". Used time: 235 Seconds
The structure of this Script is easy to understand and it can be modified for any case.

```
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    RIC.0|$$XPO|$$YPO|&HD0|0|&HD0|0|&HD0|0|$$PIX
    RIC.0|$$XPO|$$YPO|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DMP.1
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

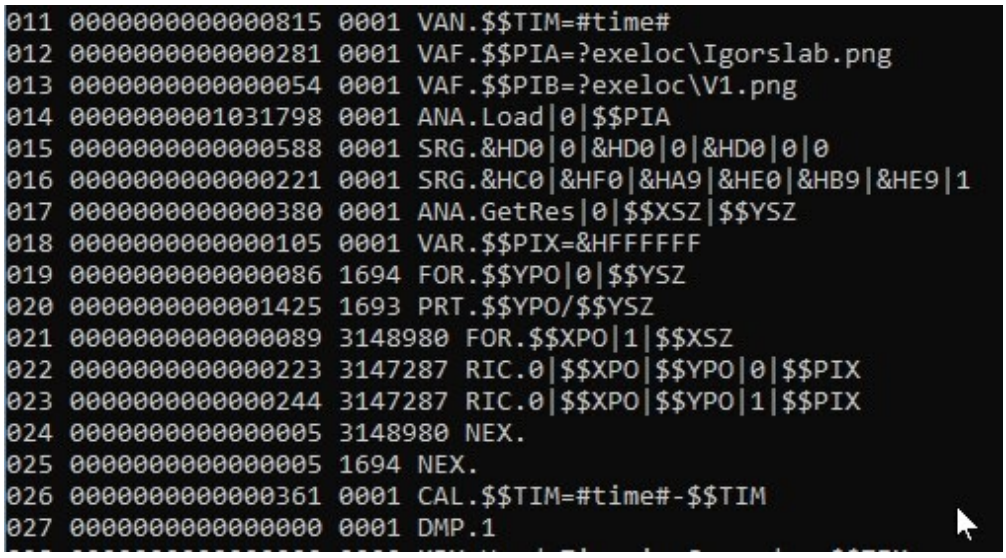
Speed-Dump:

```
011 0000000000000755 0001 VAN.$$TIM=#time#
012 0000000000000233 0001 VAF.$$PIA=?exeloc\Igorslab.png
013 0000000000000055 0001 VAF.$$PIB=?exeloc\V1.png
014 0000000001069025 0001 ANA.Load|0|$$PIA
015 0000000000000694 0001 ANA.GetRes|0|$$XSZ|$$YSZ
016 0000000000000179 0001 VAR.$$PIX=&HFFFFFF
017 0000000000000078 1694 FOR.$$YPO|0|$$YSZ
018 0000000000001112 1693 PRT.$$YPO/$$YSZ
019 0000000000000081 3148980 FOR.$$XPO|1|$$XSZ
020 0000000000000236 3147287 RIC.0|$$XPO|$$YPO|&HD0|0|&HD0|0|&HD0|0|$$PIX
021 0000000000000206 3147287 RIC.0|$$XPO|$$YPO|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|$$PIX
022 0000000000000005 3148980 NEX.
023 0000000000000004 1694 NEX.
024 0000000000000344 0001 CAL.$$TIM=#time#-$$TIM
025 0000000000000000 0001 DMP.1
```

Pixelprocessing: "Pixel by Pixel using CCR". Used time: 233 Seconds

```
' You can use the Color Condition Registers to work with less Parameters inside
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
SRG.&HD0|0|&HD0|0|&HD0|0|0
SRG.&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|1
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    RIC.0|$$XPO|$$YPO|0|$$PIX
    RIC.0|$$XPO|$$YPO|1|$$PIX
  NEX.
NEX.
CAL.$$TIM=#time#-$$TIM
DMP.1
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
ANA.Save|0|$$PIB
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

RIC.P1|P2|P3|P4|P5|P6|P7|P8|P9 or you can use CCR (Color Condition Register) RIC.P1|P2|P3[|P4]

Parameter Explanation

P1 - Image Register Number
P2 - X-Coordinate for the Pixel to read in the Image-Register
P3 - Y-Coordinate for the Pixel to read in the Image-Register

Syntax I
P4 /P5 - (numeric) Min./Max. Values for Red, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).
P6 /P7 - (numeric) Min./Max. Values for Green, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).
P8 /P9 - (numeric) Min./Max. Values for Blue, can be from 0 to 255 or in Hexadezimal from 0 to &HFF. If you write "0" in the Max-Field, it will be replaced with 255 (&HFF).

Syntax II
P4 -

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
See above.
```

Remarks

The RIC.-`Commands is fast`. It will use only about ~225 Ticks per Pixel.
Therefore the Robot could execute it more then >70000 times per Second.

Limitations:

-

See also:

- ICC. - If Color Condition 1700

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.14.14 SRG./GRG. - Set/Get Color Condition Register

SRG./GRG. - Set/Get Color Condition Register **Previous** 1088 **Top** 41 **Next** 1109

MiniRobotLanguage (MRL)

SRG. - Set Color Condition Register
GRG. - Get Color Condition Register

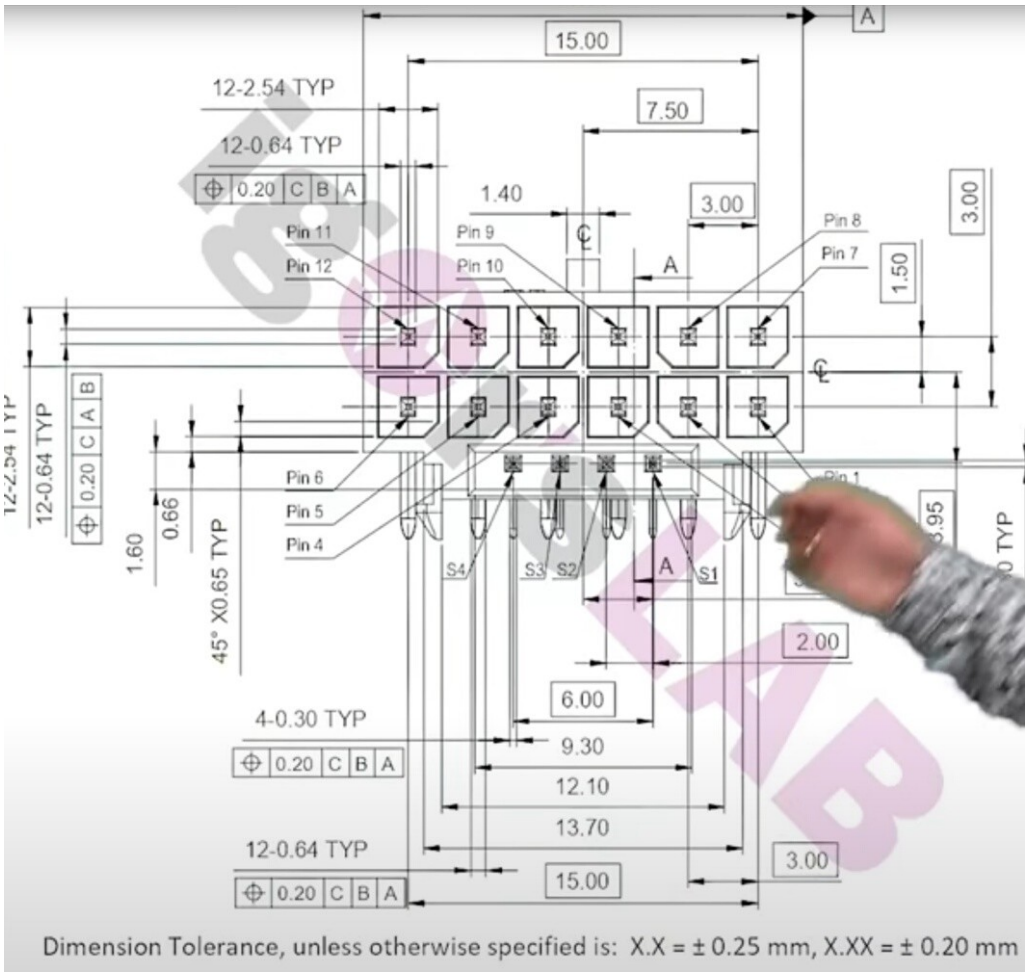
Get or Set the CCR (Color Condition Registers) for use with commands as ICC. and RIC. etc.

Intention

There are currently 8 Color Condition Registers that can be set and read using these commands.
They are numbered 0 to 7.
These commands will take 2 Color-Values for each CCR (Color Condition Register).
These two color values are the "Minimum" and the the "Maximum" value for each of "R,G and B".
So these 2 values define a "Condition". If you take any color and compare it to this condition, there is result whether the condition evaluates true or false.
So the Colour of the specified Pixel must be inside this RGB Range then the CCR Min and Max-Values then it will evaluate to "true".
Currently the SRG.-Command is the Command that is used.
The GRG. - Command is implemented for Completeness, but does not have any important use.

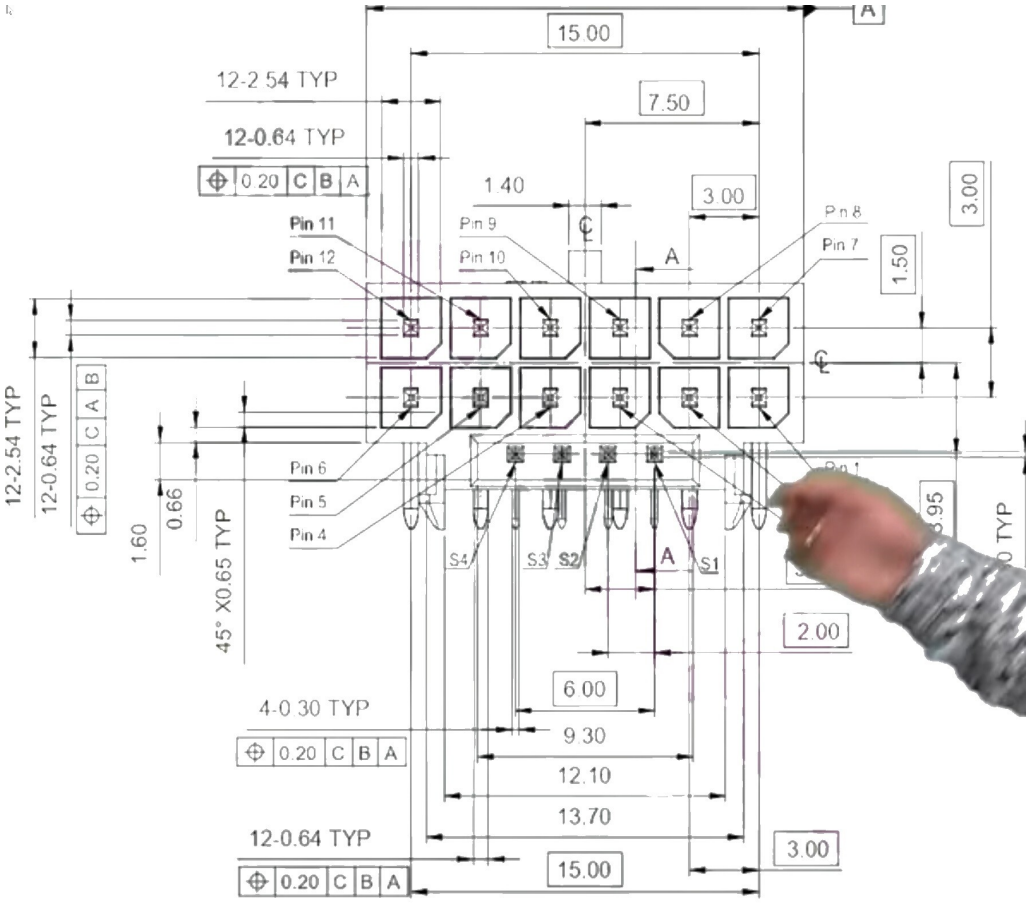
Here is a Sample Script that will remove unwanted Artefacts - for example here a watermark - from a picture.

*© 2023 Theo Gottwald * https://www.smart-package.com*



->

© 2023 Theo Gottwald * https://www.smart-package.com



Dimension Tolerance, unless otherwise specified is: X.X = ± 0.25 mm, X.XX = ± 0.20 mm

And here is a Sample Script:

```
' You can use the Color Condition Registers to work with less Parameters inside
' this may speed up the process.
VAN.$$TIM=#time#
VAF.$$PIA=?exeloc\Igorslab.png
VAF.$$PIB=?exeloc\V1.png
ANA.Load|0|$$PIA
ANA.GetRes|0|$$XSZ|$$YSZ
$$PIX=&HFFFFFF
SRG.all|&HD0|0|&HD0|0|&HD0|0|0
SRG.all|&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|1
FOR.$$YPO|0|$$YSZ
  PRT.$$YPO/$$YSZ
  FOR.$$XPO|1|$$XSZ
    ICC.0|$$XPO|$$YPO|0
    ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
  EIF.
  ICC.0|$$XPO|$$YPO|1
  ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
EIF.
NEX.
NEX.
DMP.1
CAL.$$TIM=#time#-$$TIM
MBX.Used Time in Seconds: $$TIM
ANA.Show|0!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Save|0|$$PIB
ENR.
```

Speed-Dump, total Time used: 216 Seconds:

```
011 0000000000000759 0001 VAN.$$TIM=#time#
012 0000000000000234 0001 VAF.$$PIA=?exeloc\Igorslab.png
013 0000000000000054 0001 VAF.$$PIB=?exeloc\V1.png
014 0000000001006925 0001 ANA.Load|0|$$PIA
015 0000000000000599 0001 ANA.GetRes|0|$$XSZ|$$YSZ
016 0000000000000106 0001 VAR.$$PIX=&HFFFFFF
017 0000000000000402 0001 SRG.&HD0|0|&HD0|0|&HD0|0|0
018 0000000000000209 0001 SRG.&HC0|&HF0|&HA9|&HE0|&HB9|&HE9|1
019 0000000000000078 1694 FOR.$$YPO|0|$$YSZ
020 0000000000001860 1693 PRT.$$YPO/$$YSZ
021 0000000000000081 3148980 FOR.$$XPO|1|$$XSZ
022 0000000000000164 3147287 ICC.0|$$XPO|$$YPO|0
023 0000000000000227 1784 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
024 0000000000000003 3147287 EIF.
025 0000000000000379 3147287 ICC.0|$$XPO|$$YPO|1
026 0000000000000000 0000 ANA.Setpixelcolor|0|$$XPO|$$YPO|$$PIX
027 0000000000000003 3147287 EIF.
028 0000000000000005 3148980 NEX.
029 0000000000000004 1694 NEX.
030 0000000000000000 0001 DMP.1
031 0000000000000000 0000 CAL $$TIM=#time# $$TIM
```

Syntax

SRG.P1|P2|P3|P4|P5|P6|P7
GRG.P1|P2|P3|P4|P5|P6|P7

Parameter Explanation

P1 - one of the following Terms, that will specify the used Parameters:

All - in this case you need to specify all values (See example below) - the last value is the CCR-Number (0 to 7). If omitted 0 is the default.
Details see in the Example below.

LA - you specify 2 Color Values - these are Color-Values like &HA0FF1C that represents the three values which are Red-Min,Green-Min and Blue-Min and Red-Max,Green-Max and Blue-Max.
- the **P4** value is the CCR-Number (0 to 7). If omitted 0 is the default.

Min - you only specify the 3 values which are Red-Min,Green-Min and Blue-Min.
The last value is the CCR-Number (0 to 7). If omitted 0 is the default.

© 2023 Theo Gottwald * https://www.smart-package.com

If no further values for Max will be specified, R/G/B-Max is assumed to be 255.

Max - you only specify the 3 values which are Red-Max,Green-Max and Blue-Max. The value of 0 is not allowed for a Max-Value and will be replaced by &HFF.
- the last value is the CCR-Number (0 to 7). If omitted 0 is the default.
If no further values for Min will be specified, R/G/B-Min is assumed to be 0.

Minl - you only specify the 1 Value - that is a Color-Value like &HA0FF1C that represents the three values which are Red-Min,Green-Min and Blue-Min
- the **P3** value is the CCR-Number (0 to 7). If omitted 0 is the default.
If no further values for Max will be specified, R/G/B-Max is assumed to be &HFFFFFF.

Maxl - you only specify the 1 Value - that is a Color-Value like &HA0FF1C that represents the three values which are Red-Max,Green-Max and Blue-Max
- the **P3** value is the CCR-Number (0 to 7). If omitted 0 is the default.
If no further values for Min will be specified, R/G/B-Min is assumed to be &H00000.

Using SRG. you can use numerical Variables or directly numbers.
Using GRG. the Color Values must be given as Variables, then the values will be filled into these variables.

Example

```
SRG.All|$$RMin|$$RMax|$$GMin|$$GMax|$$BMin|$$BMax|$$Num
'     0    1     2       3       4     5       6   | 7
SRG.Min|$$RMin|$$GMin|$$BMin|$$Num
'     0    1     2       3       4
SRG.Max|$$RMax|$$GMax|$$BMax|$$Num
'     0    1     2       3       4

SRG.Minl|$$Col|$$Num
SRG.Maxl|$$Col|$$Num
SRG.La|$$CMin|$$CMax[|$$Num]


GRG.All|$$RMin|$$RMax|$$GMin|$$GMax|$$BMin|$$BMax|$$Num
'     0    1     2       3       4     5       6   | 7
GRG.Min|$$RMin|$$GMin|$$BMin|$$Num
'     0    1     2       3       4
GRG.Max|$$RMax|$$GMax|$$BMax|$$Num
'     0    1     2       3       4

GRG.Minl|$$Col|$$Num
GRG.Maxl|$$Col|$$Num
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GRG.La|$$CMin|$$CMax[|$$Num]
```

Remarks

The SRG.`-Commands is fast.` It will use only about 200 to 400 Ticks. Therefore the Robot could execute it more then >50000 times per Second.

Limitations:

-

See also:

- RIC. Replace If Color Condition 1719
- ICC. If Color Condition 1700

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15 RGB- and HLS-Operations

What is HLS ?

The hexadecimal colours and the RGB colours use amounts of Red Green and **Blue** to describe colours.
This is a nice analogue to how the pixels in the screen are lighting up.
On the other side it makes picking a colour not very human friendly.

A human way to think of colour asks questions like:
Is it light or dark?
What colour is it?
Is it vibrantly coloured?

E.g. Dark Vibrant Red or Light Pale Pink

There is a way of defining colours based on these propertiess.
This is the hLS - System. It helps you, adjust a colour lighter or darker or more colourful.
Or more red or more blue.

To convert your colors into our HLS-Format you can also use:
Color Converter
The values at "WinHSL 240" are our values from the SPR.

You can find a nice HLS-Colorpicker here:
https://hslpicker.com/
or here
http://www.workwithcolor.com/hsl-color-picker-01.htm
but you will need to convert the values to WinHLS240 using the Converter here.

What are the numbers from HLS ?

HLS is another Description for Color-Information.
It contains the 3 Components: H,L and S.

0-239 **H**ue is wavelength color.
0-240 **L**uminosity (brightness, brilliance) is black 0(0%) to white240(100%).
0-240 **S**aturation is relative bandwidth amount of grey in a color. 0(0%) to 240(100%)

as you can see, the Range from HLS is not exactly the same like it was RGB.

Due to the limited Resolution of WinHLS (Goes up to 239/240 while RGB goes up to 255)
a conversion "back and forth" will look like in the picture below and has some "stairs".

```
' Depending on your Computer this Script may run for several seconds.
'
CAL.$$XSZ=10200/8
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CAL.$$YSZ=6800/8
ANA.New|0|$$XSZ|$$YSZ
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|1|$$XSZ|$$YSZ
ANA.hgradient|1|&H0000FF|&H00FFFF
ANA.Mix|0|1|7
ANA.Show|1!
ANA.RGB_HLS_IR|1|2
ANA.Show|2!
ANA.HLS_RGB_IR|2|3
DMP.1
ANA.Show|3!
ENR.
```

RGB -> HLS

```
' Test RGB->HLS Conversion
' The Output is in the SPRE-Debug-Window
'
DBP.==============================
GUT.Input R,G,B||113,100,227|$$ERG
SBD.$$ERG|,|1|$$RED
SBD.$$ERG|,|2|$$GRN
SBD.$$ERG|,|3|$$BLU
DBP.RGB=$$RED,$$GRN,$$BLU
VTH.$$RED|$$RED|2
VTH.$$GRN|$$GRN|2
VTH.$$BLU|$$BLU|2
$$RGB=&H$$RED$$GRN$$BLU
DBP.RGB=$$RGB
ANA.RGB_To_HLS|$$RGB|$$HLS
VTH.$$HLS||6
DBP.HLS=$$HLS
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSS.1|2|$$HLS|$$HUE
GSS.3|4|$$HLS|$$LUM
GSS.5|6|$$HLS|$$SAT
VBT.$$HUE
' We use VAN. to make Decimal out of Hex-Values
VAN.$$HUA=&H$$HUE
VAN.$$LUA=&H$$LUM
VAN.$$SAA=&H$$SAT
VAR.$$OUT=H=$$HUA # L=$$LUA  #  S=$$SAA
DBP.HLS=$$OUT
ENR.


' HLS -> RGB Converter
' The Output is in the SPRE-Debug-Window
'
DBP.==============================
GUT.Input H,L,S||113,100,227|$$ERG
SBD.$$ERG|,|1|$$HUE
SBD.$$ERG|,|2|$$LUM
SBD.$$ERG|,|3|$$SAT
DBP.HLS=$$HUE,$$LUM,$$SAT
VTH.$$HUE|$$HUE|2
VTH.$$LUM|$$LUM|2
VTH.$$SAT|$$SAT|2
$$HLS=&H$$HUE$$LUM$$SAT
DBP.HLS=$$HLS
ANA.HLS_To_RGB|$$HLS|$$RGB
VTH.$$RGB||6
DBP.RGB=$$RGB
GSS.1|2|$$RGB|$$RED
GSS.3|4|$$RGB|$$GRN
GSS.5|6|$$RGB|$$BLU
VBT.$$RED
VAN.$$REA=&H$$RED
VAN.$$GRA=&H$$GRN
VAN.$$BLA=&H$$BLU
VAR.$$OUT=R=$$REA # G=$$GRA  #  B=$$BLA
DBP.RGB=$$OUT
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15.1 ANA.Change_HLSA / Change_HLSM

MiniRobotLanguage (MRL)

ANA.Change_HLSA

Change full Image-Register (IR) Content HLS Additive

ANA.Change_HLSM

Change full Image-Register (IR) Content HLS using Multiplication

Intention

This command will change the HLS-Values of a complete IR.
For this it will Add or Subtract Values for H,L and S.
These values can be directly the given values using ANA.Change_HLSA
or can be values where the given values will be multiplied with the current Pixel-Values using ANA.Change_HLSM.

What is HLS? 1731

0-239 **H**ue is wavelength color.
0-240 **L**uminosity (brightness, brilliance) is black 0(0%) to white240(100%).
0-240 **S**aturation is relative bandwidth amount of grey in a color. 0(0%) to 240(100%)
as you can see, the Range from HLS is not exactly the same like it was RGB.
For further use.

1. Using the Multiplikation HLS-Modifier
Here is the Source-Code:

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
ANA.Show|0!
$$HUE=0
$$LUM=150
$$SAT=-100
ANA.Change_HLSM|0|$$HUE|$$LUM|$$SAT
ANA.Show|0!
ENR.
```

Picture 1: Original in IR 2
Picture 2 in IR 1
Picture 3 in IR 3

© 2023 Theo Gottwald * https://www.smart-package.com

->

© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000171 0001 VAF.$$PIA=?exeloc\Sara2.jpg
014 0000000000001797 0001 NEF.$$PIA
015 0000000000000000 0000 MBX.No such file.
016 0000000000000000 0000 ENR.
017 0000000000000023 0001 EIF.
018 0000000000368913 0001 ANA.Load|0|$$PIA
019 0000000026258485 0001 ANA.Show|0!
020 0000000000000140 0001 VAR.$$HUE=0
021 0000000000000037 0001 VAR.$$LUM=150
022 0000000000000033 0001 VAR.$$SAT=-100
023 0000000000458359 0001 ANA.Change_HLSM|0|$$HUE|$$LUM|$$SAT
024 0000000000000000 0001 DMP.1
```

2. Using the Additive HLS-Modifier
Here is the Source-Code:

```
VAF.$$PIA=?exeloc\Sara2.jpg
NEF.$$PIA
  MBX.No such file.
  ENR.
EIF.
ANA.Load|0|$$PIA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ANA.Show|0!
$$HUE=0
$$LUM=75
$$SAT=-50
ANA.Change_HLSA|0|$$HUE|$$LUM|$$SAT
ANA.Show|0!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.Change_HLSM|P1[|P2][|P3][|P4][|P5]
ANA.Change_HLSA|P1[|P2][|P3][|P4][|P5]

Parameter Explanation

P1 **-** ***opt. Source*** Image-Register number, if omitted, IR 0 will be used.
P2 **-** ***opt Hue-Value*** This Value will be added or Subtracted from the Pixel-Color HLS-Value. If omitted 0 is used.
P3 **-** ***opt Luminosity-Value*** This Value will be added or Subtracted from the Pixel-Color HLS-Value. If omitted 0 is used.
P4 **-** ***opt Saturation-Value*** This Value will be added or Subtracted from the Pixel-Color HLS-Value. If omitted 0 is used.
P5 **-** ***opt.Destination IR*** Image-Register number, if omitted, IR **P1** will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

–

Remarks

–

Limitations:

-

See also:

- ANA.Luminance / Change_Luminance 1366

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15.2 ANA.HLS_RGB / RGB_HLS

ANA.RGB_HLS / ANA.HLS_RGB Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.RGB_HLS
ANA.HLS_RGB

Convert full Image-Register (IR) Content from R,G B to H,L,S and opposite direction

Intention

This command will convert the full Content of an given IR from R,G, B into H,L,S.
To just convert a SingleColor-Valur between HLS and RGB use other commands.

What is HLS ?

0-239 Hue is wavelength color.
0-240 Luminosity (brightness, brilliance) is black 0(0%) to white240(100%).
0-240 Saturation is relative bandwidth amount of grey in a color. 0(0%) to 240(100%)
as you can see, the Range from HLS is not exactly the same like it was RGB.
For further use.

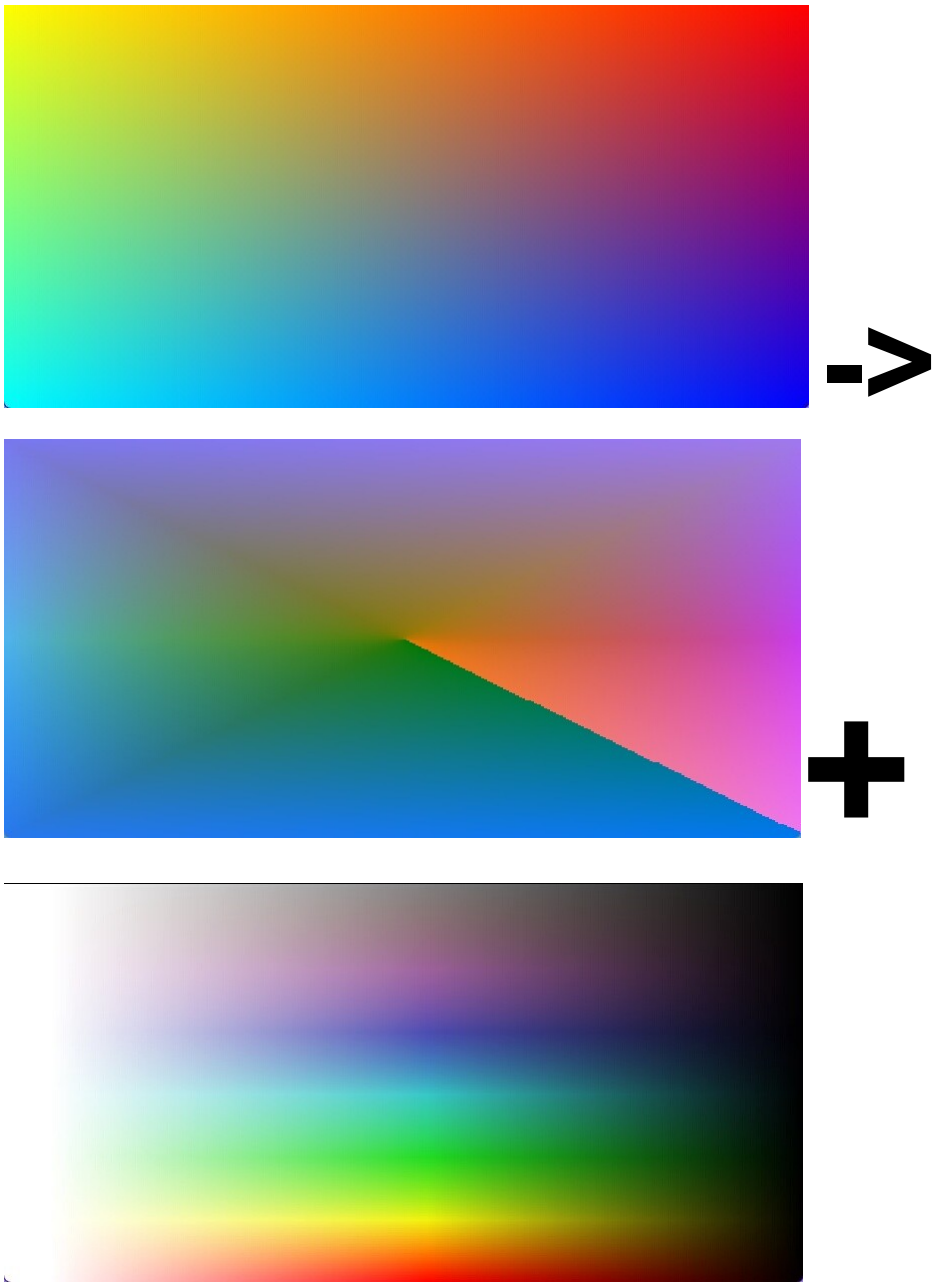
RGB -> HLS

1. Generate Gradfients, mix these and convert the result to HLS
Here is the Source-Code:

```
ANA.New|0|800|400
ANA.vgradient|0|&HFF00FF|&H000000
ANA.New|2|800|400
ANA.hgradient|2|&H0000FF|&H00FFFF
ANA.Mix|0|2|7
ANA.Show|2!
ANA.RGB_HLS_IR|2|1
ANA.Show|1!
ANA.HLS_RGB_IR|2|3
DMP.1
ANA.Show|3!
ENR.
```

Picture 1: Original in IR 2
Picture 2 in IR 1
Picture 3 in IR 3

© 2023 Theo Gottwald * https://www.smart-package.com



```
013 0000000000013033 0001 ANA.New|0|800|400
014 0000000000018679 0001 ANA.vgradient|0|&HFF00FF|&H000000
015 0000000000008864 0001 ANA.New|2|800|400
016 0000000000030889 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
017 0000000000039256 0001 ANA.Mix|0|2|7
018 0000001874789587 0001 ANA.Show|2!
019 0000000000124367 0001 ANA.RGB_HLS_IR|2|1
020 0000000299647569 0001 ANA.Show|1!
021 0000000006019279 0001 ANA.HLS_RGB_IR|2|3
022 0000000000000000 0001 DMP.1
023 0000000000000000 0000 ANA.Show|3!
024 0000000000000000 0000 ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.RGB_HLS_IR|P1[|P2]
ANA.HLS_RGB_IR|P1[|P2]

Parameter Explanation

P1 - opt. Source Image-Register number, if omitted, IR 0 will be used.
P2 - opt.Destination IR Image-Register number, if omitted, IR **P1** will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

–

Remarks

`ANA.HLS_RGB_IR` is much slower then the `ANA.RGB_HLS_IR`.
Anyway, both of these should do their work in at most some seconds.
In this example the slowest time is 21 Seconds.

At a resolution of 5100x3400 this is the Speed-Dump:

```
013 0000000000000315 0001 CAL.$$XSZ=5100
014 0000000000000164 0001 CAL.$$YSZ=3400
015 0000000000702926 0001 ANA.New|0|$$XSZ|$$YSZ
016 0000000000855235 0001 ANA.vgradient|0|&HFF00FF|&H000000
017 0000000000684891 0001 ANA.New|2|$$XSZ|$$YSZ
018 0000000002103994 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
019 0000000004004146 0001 ANA.Mix|0|2|7
020 0000000044712912 0001 ANA.Show|2!#
021 0000000006471893 0001 ANA.RGB_HLS_IR|2|1
022 0000000000000533 0001 GSB.Save|1
023 0000000341730901 0001 ANA.HLS_RGB_IR|2|3
024 0000000000000000 0001 DMP.1
025 0000000000000000 0000 GSB.Save|3
026 0000000000000000 0000 ENR.
```

Limitations:

I have tested the Script up to a resolution of 10200x6800 and it will work, however the Debug-Window can not display such a large resolution (because your Screeen may not have such a size) therefore you need to use the Parameter "#" for the ANA.Show Command. Anything else works fine, and you can also save the resulting pictures.

© 2023 Theo Gottwald * https://www.smart-package.com

```
013 0000000000000274 0001 CAL.$$XSZ=10200
014 0000000000000124 0001 CAL.$$YSZ=6800
015 0000000019925766 0001 ANA.New|0|$$XSZ|$$YSZ
016 0000000003535295 0001 ANA.vgradient|0|&HFF00FF|&H000000
017 0000000003298271 0001 ANA.New|2|$$XSZ|$$YSZ
018 0000000026375614 0001 ANA.hgradient|2|&H0000FF|&H00FFFF
019 0000000015645189 0001 ANA.Mix|0|2|7
020 0000000000000412 0001 GSB.Save|2
021 0000000027206773 0001 ANA.Show|2!#
022 0000000042930524 0001 ANA.RGB_HLS_IR|2|1
023 0000000000000215 0001 GSB.Save|1
024 0000001300047891 0001 ANA.HLS_RGB_IR|2|3
025 0000000000000000 0001 DMP.1
026 0000000000000000 0000 GSB.Save|3
027 0000000000000000 0000 ENR.
```

See also:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.4.15.3 ANA.HLS_To_RGB / RGB_To_HLS

MiniRobotLanguage (MRL)

ANA.HLS_To_RGB
ANA.RGB_To_HLS

Convert values (Single or splitted vaues) from R,G B to H,L,S and opposite direction

Intention

This command will convert single values like &HF1F0A0 or Splitted values where R/G/B or H/L/S are splitted in 3 Parameters from R,G, B into H,L,S and the opposite direction.
To convert a Full-IR between HLS and RGB use other commands.

To convert your colors into our "WinHSL240"-Format, - here in short - HLS-Format, you can also use:
Color Converter
The values at "WinHSL 240" are our values from the SPR.
Due to technical rounding, our values may differ +/-3 with the values on the WEB-Site.

These commands can be called with a variety of Parameters.

```
' R=113 G=100 B=229 (&H7164E5)
' H=164 L=155 S=171 (&HA49BAB)
'**********************************
' 1-1 Conversion
'**********************************
$$RGB=&H7164E5
ANA.RGB_To_HLS|$$RGB|$$HLS
VTH.$$HLS||6
DBP.$$HLS
' Result is &HA49BAB
ENR.

$$HLS=&HA49BAB
ANA.HLS_To_RGB|$$HLS|$$RGB
VTH.$$RGB||6
DBP.$$RGB
' Result is &H7164E6
ENR.

'**********************************
' 1-3 Conversion
'**********************************
$$RGB=&H7164E5
ANA.RGB_To_HLS|$$RGB|$$HUE|$$LUM|$$SAT
DBP.HLS=$$HUE,$$LUM,$$SAT
VTH.$$HUE|$$HUE|2
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
VTH.$$LUM|$$LUM|2
VTH.$$SAT|$$SAT|2
$$HLS=&H$$HUE$$LUM$$SAT
DBP.HLS=$$HLS
' Result is &HA49BAB
ENR.

$$HLS=&HA49BAB
ANA.HLS_To_RGB|$$HLS|$$RED|$$GRN|$$BLU
DBP.RGB=$$RED,$$GRN,$$BLU
VTH.$$RED|$$RED|2
VTH.$$GRN|$$GRN|2
VTH.$$BLU|$$BLU|2
$$RGB=&H$$RED$$GRN$$BLU
DBP.RGB=$$RGB
' Result is &H7164E6
ENR.

'**********************************
' 3-1 Conversion
' Using the 3-1 Syntax, the result is always on TOS,
' You always use 3 Parameters
' to not intermix it with the 13-Conversions that uses 4 Parameters
'**********************************

$$HUE=&HA4
$$LUM=&H9B
$$SAT=&HAB
ANA.HLS_To_RGB|$$HUE|$$LUM|$$SAT
POP.$$RES
VTH.$$RES||6
DBP.RGB=$$RES
' Result is &H7164E6
ENR.

$$RED=&H71
$$GRN=&H64
$$BLU=&HE5
ANA.RGB_To_HLS|$$RED|$$GRN|$$BLU
POP.$$RES
VTH.$$RES||6
DBP.HLS=$$RES
' Result is &HA49BAB
ENR.

'**********************************
' 3-3 Conversion
'**********************************

$$RED=&H71
$$GRN=&H64
$$BLU=&HE5
ANA.RGB_To_HLS|$$RED|$$GRN|$$BLU|$$HUE|$$LUM|$$SAT
DBP.HLS=$$HUE-$$LUM-$$SAT
' Result is HLS=164-155-171
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$HUE=&HA4
$$LUM=&H9B
$$SAT=&HAB
ANA.HLS_To_RGB|$$HUE|$$LUM|$$SAT|$$RED|$$GRN|$$BLU
DBP.RGB=$$RED-$$GRN-$$BLU
' Result is  RGB=113-100-230
ENR.
```

Here are 2 Programs where you can enter an RGB / HSL Value and then get the result of the conversion.

```
' Test RGB->HLS Conversion
' The Output is in the SPRE-Debug-Window
'
DBP.==============================
GUT.Input R,G,B||113,100,227|$$ERG
SBD.$$ERG|,|1|$$RED
SBD.$$ERG|,|2|$$GRN
SBD.$$ERG|,|3|$$BLU
DBP.RGB=$$RED,$$GRN,$$BLU
VTH.$$RED|$$RED|2
VTH.$$GRN|$$GRN|2
VTH.$$BLU|$$BLU|2
$$RGB=&H$$RED$$GRN$$BLU
DBP.RGB=$$RGB
ANA.RGB_To_HLS|$$RGB|$$HLS
VTH.$$HLS||6
DBP.HLS=$$HLS
GSS.1|2|$$HLS|$$HUE
GSS.3|4|$$HLS|$$LUM
GSS.5|6|$$HLS|$$SAT
VBT.$$HUE
' We use VAN. to make Decimal out of Hex-Values
VAN.$$HUA=&H$$HUE
VAN.$$LUA=&H$$LUM
VAN.$$SAA=&H$$SAT
VAR.$$OUT=H=$$HUA # L=$$LUA  #  S=$$SAA
DBP.HLS=$$OUT
ENR.
```

```
' HLS -> RGB Converter
' The Output is in the SPRE-Debug-Window
'
DBP.==============================
GUT.Input H,L,S||113,100,227|$$ERG
SBD.$$ERG|,|1|$$HUE
SBD.$$ERG|,|2|$$LUM
SBD.$$ERG|,|3|$$SAT
DBP.HLS=$$HUE,$$LUM,$$SAT
VTH.$$HUE|$$HUE|2
VTH.$$LUM|$$LUM|2
VTH.$$SAT|$$SAT|2
$$HLS=&H$$HUE$$LUM$$SAT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.HLS=$$HLS
ANA.HLS_To_RGB|$$HLS|$$RGB
VTH.$$RGB||6
DBP.RGB=$$RGB
GSS.1|2|$$RGB|$$RED
GSS.3|4|$$RGB|$$GRN
GSS.5|6|$$RGB|$$BLU
VBT.$$RED
VAN.$$REA=&H$$RED
VAN.$$GRA=&H$$GRN
VAN.$$BLA=&H$$BLU
VAR.$$OUT=R=$$REA # G=$$GRA  #  B=$$BLA
DBP.RGB=$$OUT
ENR.
```

Syntax

These Functions can be called with a different number of Paramenters. Depending on these number the Parameter have different meaning.

ANA.HLS_To_RGB|P1
ANA.RGB_To_HLS|P1

Parameter Explanation

P1 - Source Value for RGB / HLS. Example: &HF1A9B4 This value is interpreted as RGB or as HLS value.

The result of the Operation will be placed on TOS.

ANA.HLS_To_RGB|P1|P2
ANA.RGB_To_HLS|P1|P2

Parameter Explanation

P1 - Source Value for RGB / HLS. Example: &HF1A9B4 This value is interpreted as RGB or as HLS value.

P2 - Variable for Result.

ANA.HLS_To_RGB|P1|P2]|P3
ANA.RGB_To_HLS|P1|P2]|P3

Parameter Explanation

P1 - Source Value/Variable for Red / Hue.

P2 - Source Value/Variable for Green / Luminosity.

P3 - Source Value/Variable for Blue / Saturation.

The result of the Operation will be placed on TOS.

ANA.HLS_To_RGB|P1|P2|P3|P4

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.RGB_To_HLS|P1|P2|P3|P4

Parameter Explanation

P1 - Source Value for RGB / HLS. Example: &HF1A9B4 This value is interpreted as RGB or as HLS value.
P2 - Destination Variable for Red / Hue.
P3 - Destination Variable Green / Luminosity.
P4 - Destination Variable Blue / Saturation.

ANA.HLS_To_RGB|P1|P2|P3|P4|P5|P6
ANA.RGB_To_HLS|P1|P2|P3|P4|P5|P6

Parameter Explanation

P1 - Source Value/Variable for Red / Hue.
P2 - Source Value/Variable for Green / Luminosity.
P3 - Source Value/Variable for Blue / Saturation.
P4 - Destination Variable for Red / Hue.
P5 - Destination Variable Green / Luminosity.
P6 - Destination Variable Blue / Saturation.

Example

-

Remarks

-

Limitations:

-

See also:

- .ANA.RGB_HLS / HLS_RGB 1741

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15.4 ANA.MixNG/NR/NB

MiniRobotLanguage (MRL)

ANA.MixNG

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, will not mix the Green part

ANA.MixNR

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, will not mix the Red part

ANA.MixNB

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, will not mix the Blue part

Intention

This command will mix the Content of R,G and B-Part each in separate Image-Registers, into a Single IR-Picture.
It will drop the specified part and only mix the other two parts.

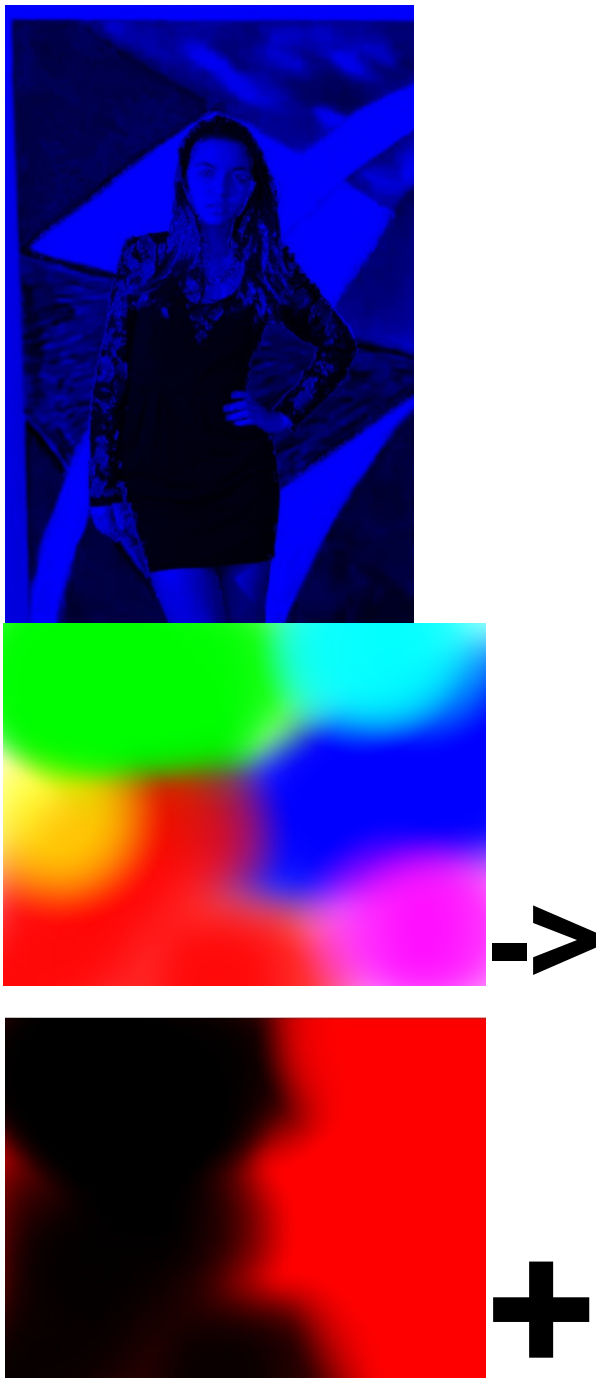
Split Picture in R,G and B-Parts and Mix it together into another Picture
SplitRGB is technically the Opposite then the ANA.MixRGB - Commands.

```
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
ANA.Show|3!
ANA.MixRGB|6|1
ANA.Show|6!
ENR.
```

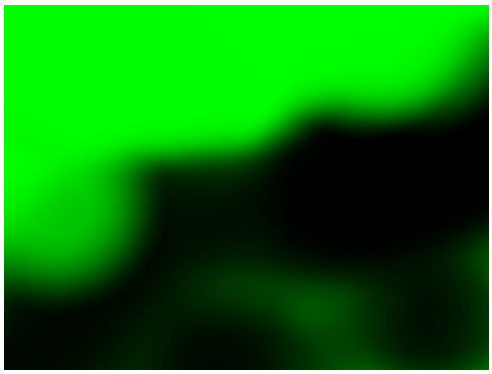
Picture 1: Original in IR 0 Picture 2 in IR 1 Picture 3 in IR 2
Picture 4 in IR 3

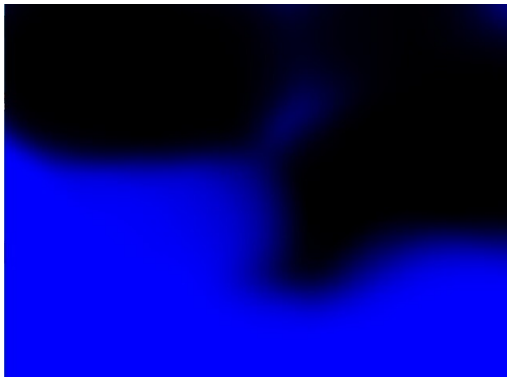
© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

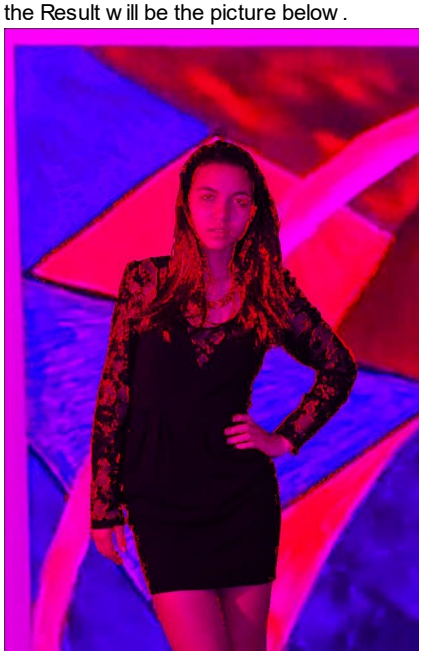




Below we will show just the result of the Mixing-Commands:

ANA.MixNG

the Result w ill be the picture below .



ANA.MixNR

the Result w ill be the picture below .

© 2023 Theo Gottwald * https://www.smart-package.com

ANA.MixNB

the Result w ill be the picture below .

Syntax

```
ANA.MixRGB[|P1][|P2]
ANA.MixBGR[|P1][|P2]
ANA.MixBRG[|P1][|P2]
ANA.MixGRB[|P1][|P2]
```

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - opt. Destination Image-Register number, if omitted, IR 0 will be used.
P2 - opt.Source R Image-Register number, if omitted, IR **P1+1** will be used.
Important:
The Image Registers for G+B are P2+1 and P2+2

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.SplitRGB and MixRGB-Sample
'
'**********************************
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
ANA.Show|3!
' Now mix the 3 Parts together in IR 6
ANA.MixRGB|6|1
ANA.Show|6!
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15.5 ANA.MixRGB/BGR/BRG/GRB

MiniRobotLanguage (MRL)

ANA.MixRGB

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR

ANA.MixBGR

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, flipping IR's into BGR

ANA.MixBRG

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, flipping IR's into BRG

ANA.MixGRB

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, flipping IR's into GRB

Intention

This command will mix the Content of R,G and B-Part each in separate Image-Registers, into a Single IR-Picture.

1. Split Picture in R,G and B-Parts and Mix it together into original Picture

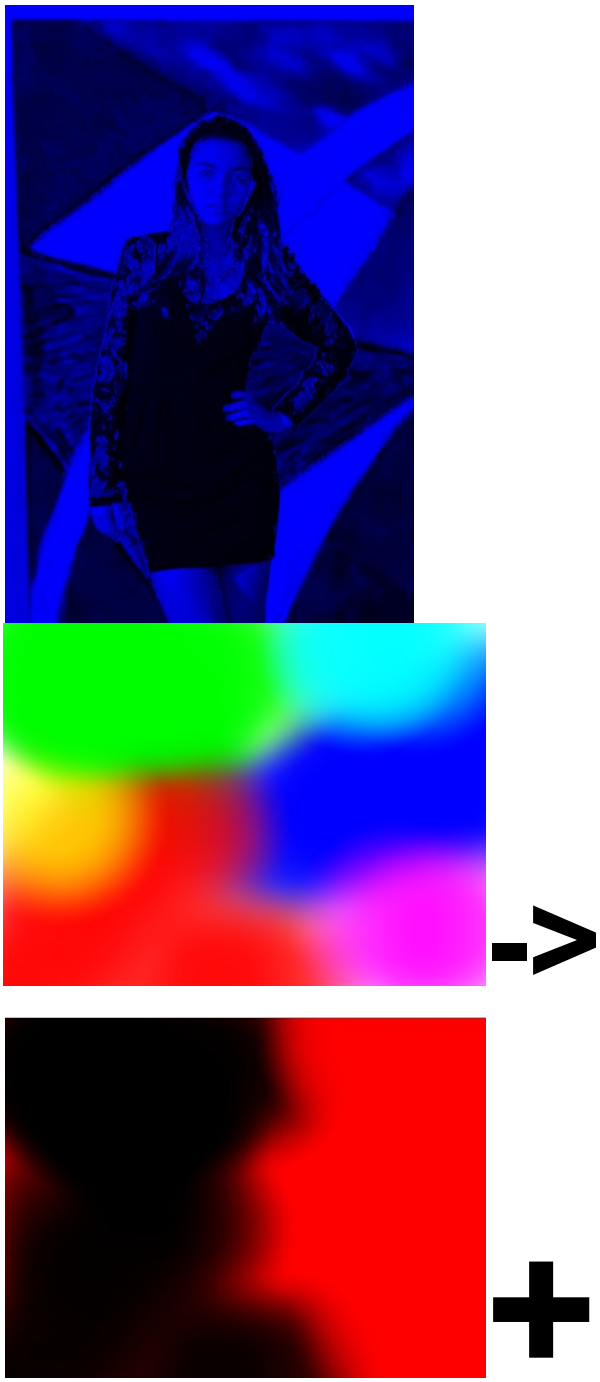
SplitRGB is technically the Opposite then the ANA.MixRGB - Command.

```
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
ANA.Show|3!
ANA.MixRGB|6|1
ANA.Show|6!
ENR.
```

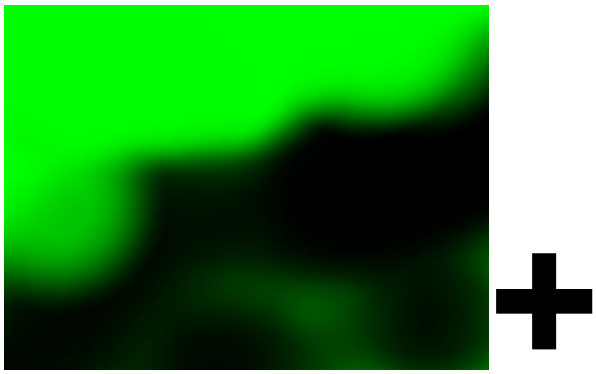
Picture 1: Original in IR 0 Picture 2 in IR 1 Picture 3 in IR 2 Picture 4 in IR 3

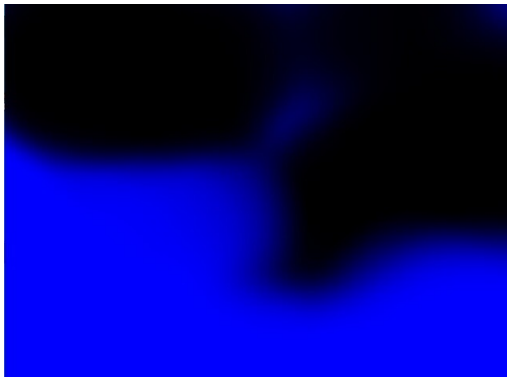
*© 2023 Theo Gottwald * https://www.smart-package.com*

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com



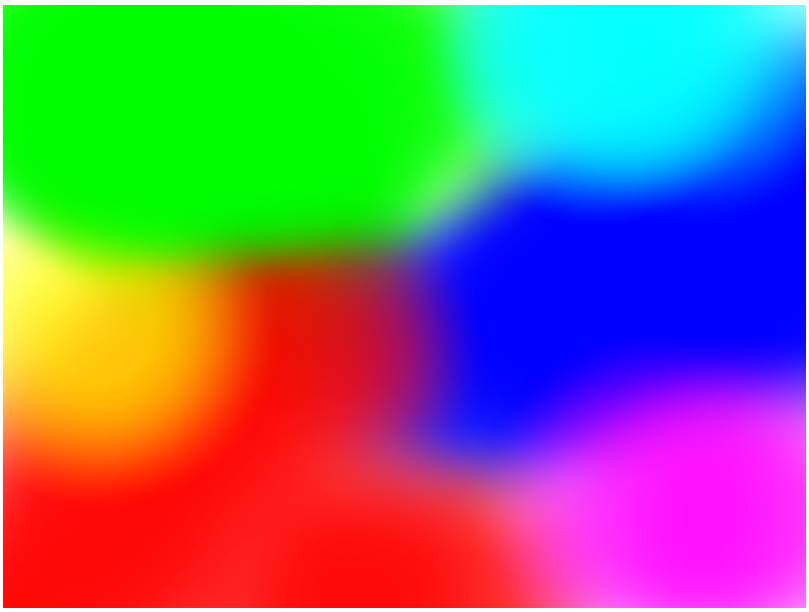


Below we will show just the result of the Mixing-Commands:

```
ANA.MixRGB
```

the Result w ill just be the Original, unless you have changed the Parts.

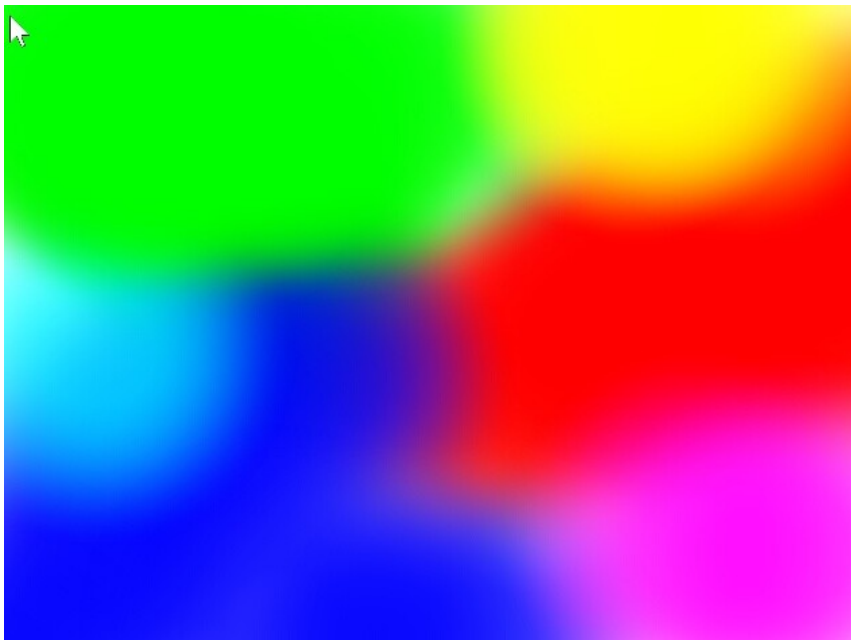
© 2023 Theo Gottwald * https://www.smart-package.com



ANA.MixBGR

the Result will just be the Original, unless you have changed the Parts.

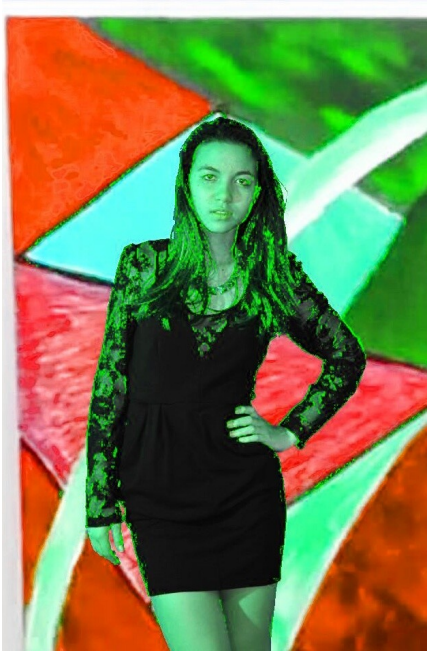
© 2023 Theo Gottwald * https://www.smart-package.com

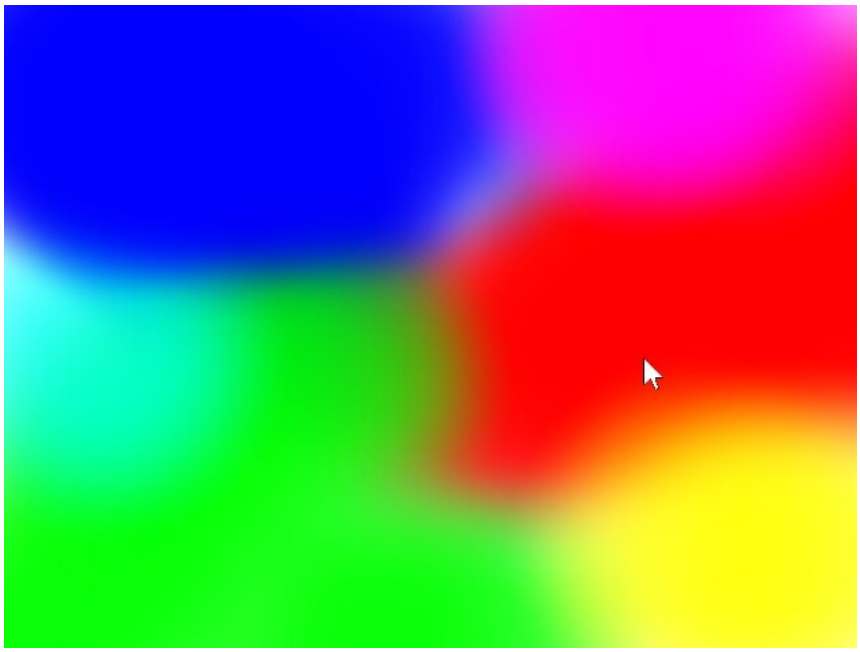


```
ANA.MixBRG
```

the Result w ill just be the Original, unless you have changed the Parts.

© 2023 Theo Gottwald * https://www.smart-package.com

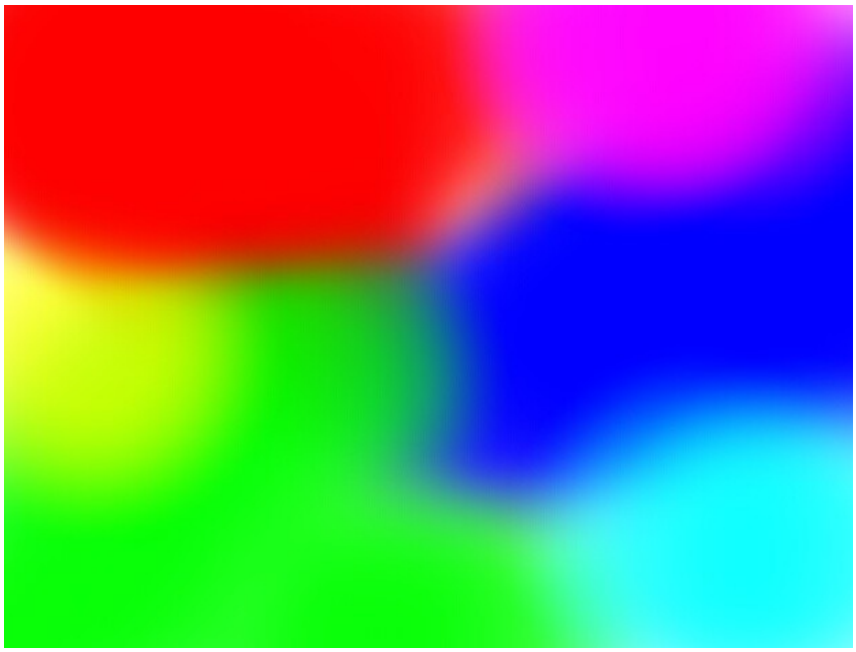




ANA.MixGRB

the Result w ill just be the Original, unless you have changed the Parts.

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

```
ANA.MixRGB[|P1][|P2]
ANA.MixBGR[|P1][|P2]
ANA.MixBRG[|P1][|P2]
ANA.MixGRB[|P1][|P2]
```

Parameter Explanation

P1 **-** ***opt. Destination*** Image-Register number, if omitted, IR 0 will be used.
P2 **-** ***opt.Source R*** Image-Register number, if omitted, IR **P1+1** will be used.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Important:
The Image Registers for G+B are P2+1 and P2+2

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.SplitRGB and MixRGB-Sample
'
'**********************************
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
ANA.Show|3!
' Now mix the 3 Parts together in IR 6
ANA.MixRGB|6|1
ANA.Show|6!
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15.6 ANA.MixROR/ROL/DROR/DROL

ANA.MixROR/ROL - Mix R,G and B Part into IR Previous 1088 Top 41 Next 1109

MiniRobotLanguage (MRL)

ANA.MixROR

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR using a Bit Right-Shift (by P3 Bits)

ANA.MixROL

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, using a Bit Left-Shift (by P3 Bits)

ANA.MixDROR

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, using a Bit Right-Shift (by P3 Bits) and doing a Bit-Reversal LSB<>MSB

ANA.MixDROL

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, using a Bit Left-Shift (by P3 Bits) and doing a Bit-Reversal LSB<>MSB

ANA.MixRev

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, using a Bit Left-Shift (by P3 Bits) and doing a Bit-Reversal LSB<>MSB

ANA.MixRevN

Mix Contents from 3 Image-Register (IR) that contain R,G and B-Part into a Single IR, using a Bit Left-Shift (by P3 Bits) and doing a Bit-Reversal LSB<>MSB and a Negation

Intention

This command will mix the Content of R,G and B-Part each in separate Image-Registers, into a Single IR-Picture.
Its intended to analyze the pictures or apply Special effects.

Split Picture in R,G and B-Parts and Mix it together using the ROL/ROR_Commands

Technically these commands eleminate some of the color-parts.

```
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
```

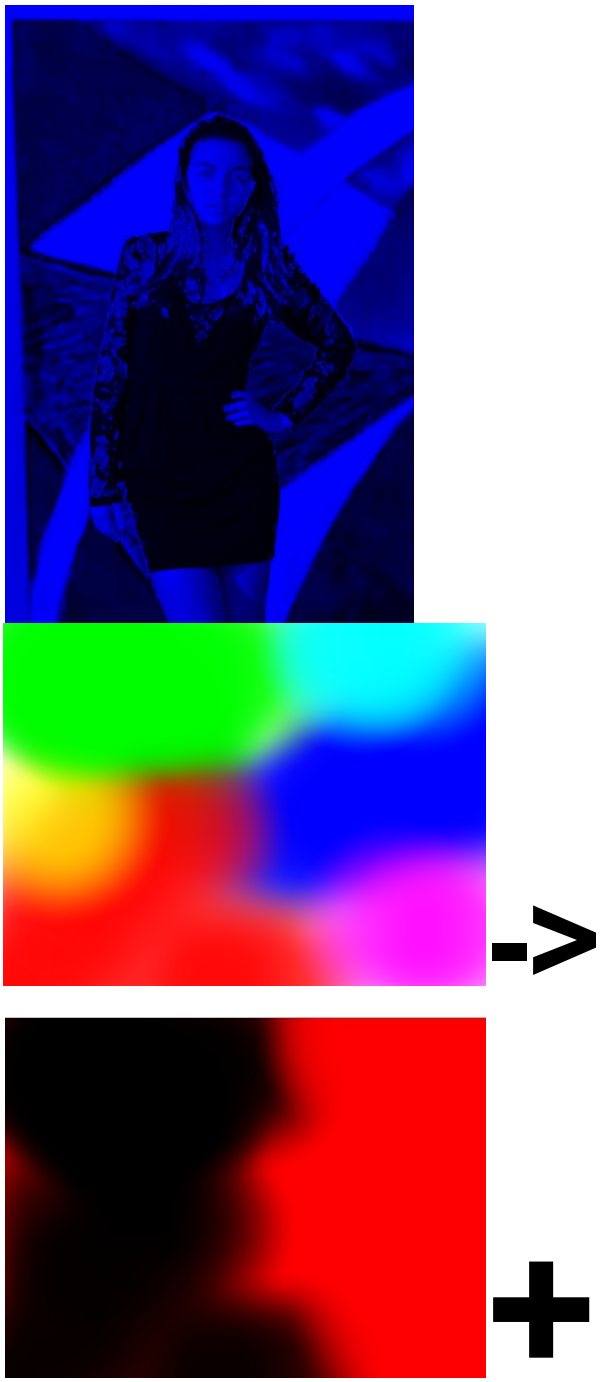

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
ANA.Show|3!
ANA.MixROL|6|1|1
ANA.Show|6!
ENR.
```

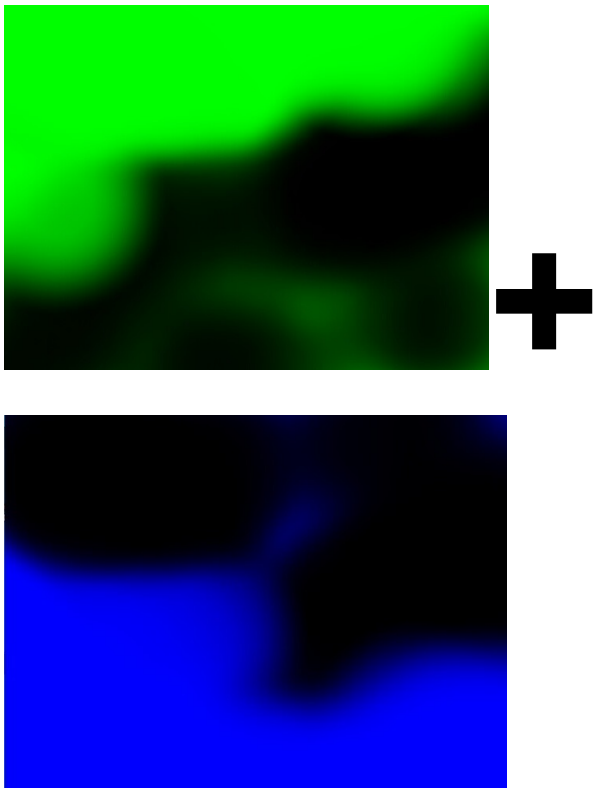
Picture 1: Original in IR 0 **Picture 2 in IR 1** **Picture 3 in IR 2**
Picture 4 in IR 3

*© 2023 Theo Gottwald * https://www.smart-package.com*

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

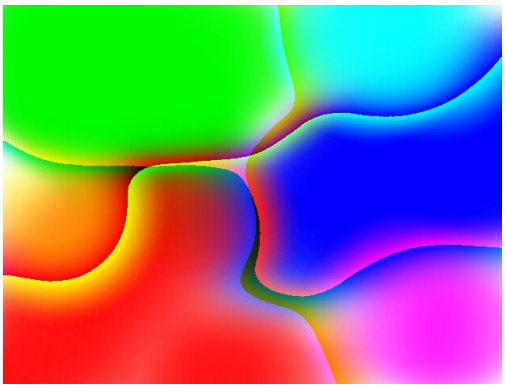


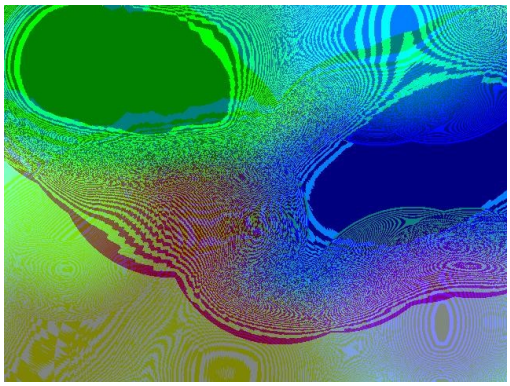
Below we will show just the result of the Mixing-Commands:

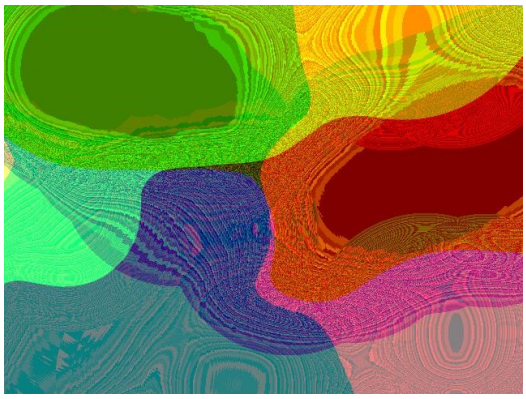
`ANA.MixROL|1`
The result will be the blue part removed.
be the red part removed
The result w ill be the red part.
just be the Blue-Part.

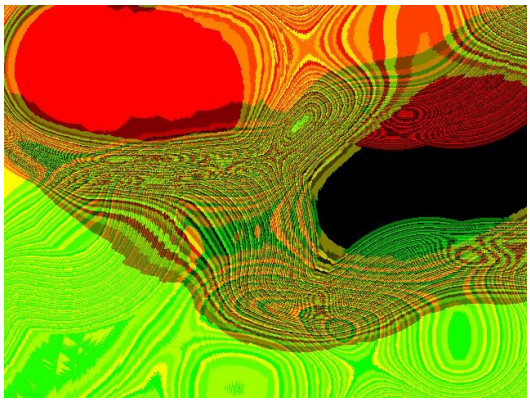
`ANA.MixROR|1`
The result will
The Result w ill just be the Blue-Part.
The Result w ill
The result w ill be the red part.

© 2023 Theo Gottwald * https://www.smart-package.com

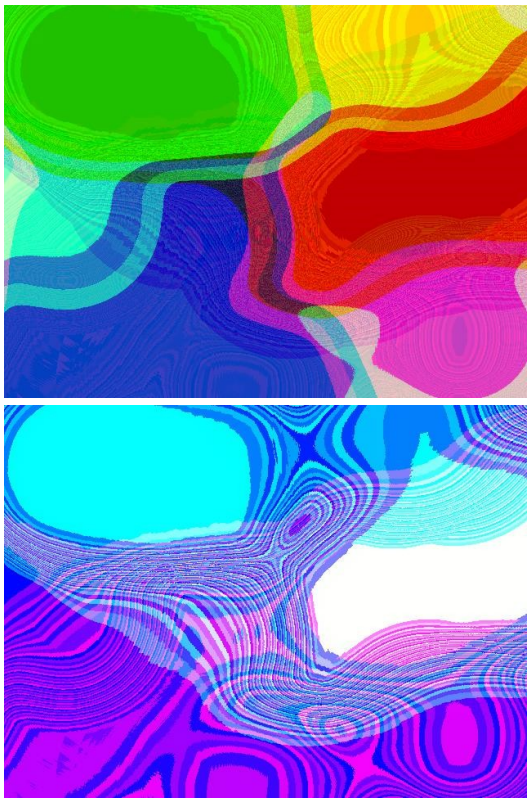








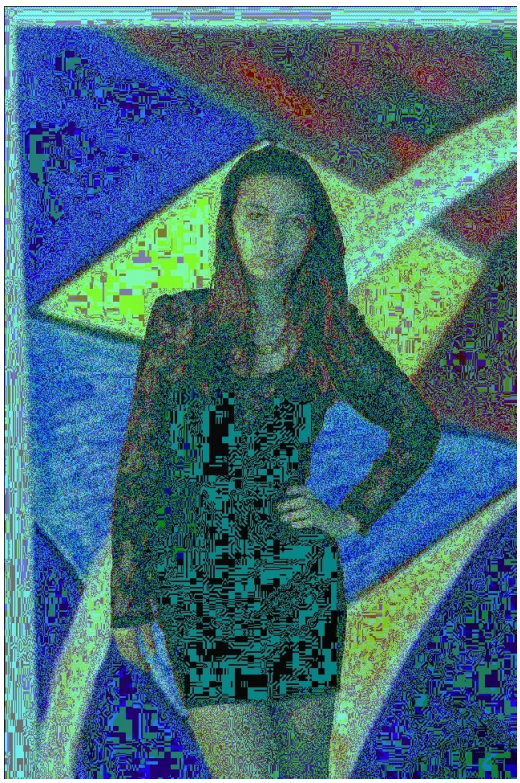
© 2023 Theo Gottwald * https://www.smart-package.com



ANA.MixROL|1 ANA.MixROR|1 ANA.MixDROR|1 ANA.MixDROL|1 ANA.MixRev|2 ANA.MixRevN|2

The result will be the blue part removed. The result will be the red part removed The Result w ill just be the Blue-Part. The result w ill be the red part. The Result w ill just be the Blue-Part. The result w ill be the red part.

© 2023 Theo Gottwald * https://www.smart-package.com



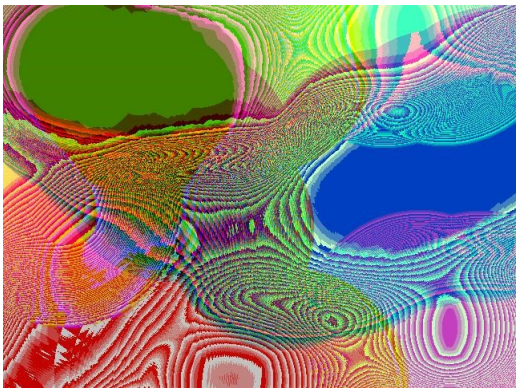
© 2023 Theo Gottwald * https://www.smart-package.com

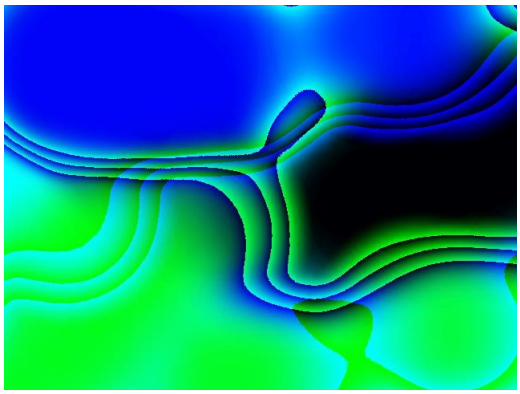
© 2023 Theo Gottwald * https://www.smart-package.com

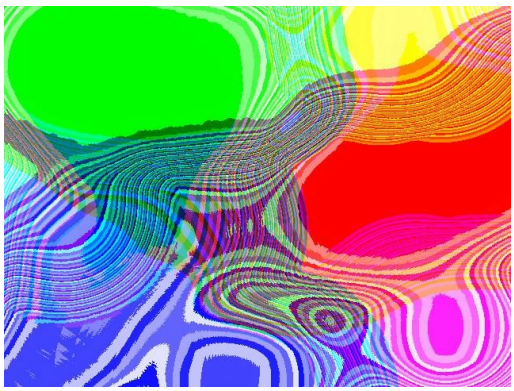
ANA.MixROL|6 ANA.MixROR|6 ANA.MixDROR|6 ANA.MixDROL|6

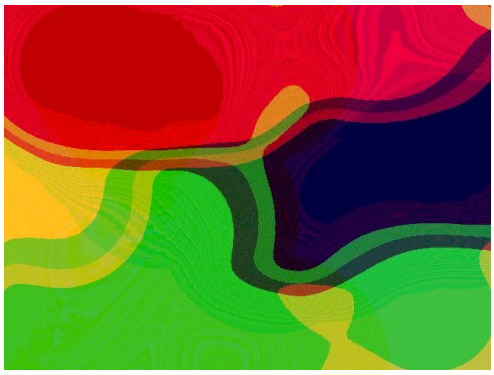
The result will be the blue part removed. The result will be the red part removed The Result w ill just be the Blue-Part. The result w ill be the red part.

© 2023 Theo Gottwald * https://www.smart-package.com







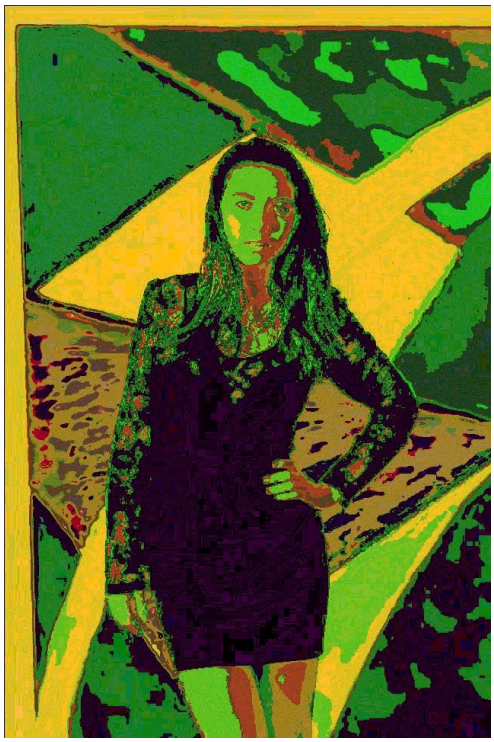


ANA.MixROL|6 ANA.MixROR|6 ANA.MixDROR|6 ANA.MixDROL|6

© 2023 Theo Gottwald * https://www.smart-package.com

The result will be the blue part removed. The result will be the red part removed The Result w ill just be the Blue-Part.
The result w ill be the red part.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.MixROR[|P1][|P2][|P3]
ANA.MixROL[|P1][|P2][|P3]
ANA.MixDROR[|P1][|P2][|P3]
ANA.MixDROL[|P1][|P2][|P3]
ANA.MixRev[|P1][|P2][|P3]
ANA.MixRevN[|P1][|P2][|P3]

Parameter Explanation

P1 - opt Destination Image-Register number, if omitted, IR 0 will be used.
P2 - opt Source R Image-Register number, if omitted, IR **P1+1** will be used.
P3 - opt Number of Bit-Shifts This numberwill specify how often the Bits will be shifted to LEFT (ROL/DROL) or RIGHT (ROR/DROR).
You can use any number from 0 (no change) to 31.

Important:
The Image Registers for G+B are P2+1 and P2+2

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'***********************************
' ANA.SplitRGB and MixROR-Sample
' Will save all resulting Pictures as ".jpg"
'***********************************
FOR.$$LOP|0|32
  $$FIA=?pattern\SarahC_640.jpg
  $$FIB=?pattern\Pic_Sa_ROR_$$LOP.jpg
  ANA.Load|0|$$FIA
  ANA.SplitRGB|0|1
  ANA.MixROR|6|1|$$LOP
  ANA.Show|6
  ANA.Save|6|$$FIB
NEX.
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.4.15.7 ANA.SplitRGB

ANA.SplitRGB - Split IR into R,G and B Part **Previous** 1088 **Top** 41 **Next** 1109

MiniRobotLanguage (MRL)

ANA.SplitRGB

Split Image-Register (IR) Content into R,G and B-Part

Intention

This command will Split the Content of an given IR into R,G and B-Part each in separate Image-Registers.
For further use.

1. Split Picture in R,G and B-Parts

SplitRGB is technically the Opposite then the ANA.MixRGB - Command.
You can NOT just swap the R,G and B - IR's and re-combine them. If you want to Swap R,G and B-Parts use the ANA. Mix_???-Commands for that.

```
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
ANA.Show|3!
ENR.
```

Picture 1: Original in IR 0 **Picture 2 in IR 1**
Picture 3 in IR 2 **Picture 4 in IR 3**

*© 2023 Theo Gottwald * https://www.smart-package.com*

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

ANA.SplitRGB[|P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - opt. Source Image-Register number, if omitted, IR 0 will be used.
P2 - opt.Destination R Image-Register number, if omitted, IR **P1+1** will be used.
P3 - opt Destination G Image-Register number, if omitted, IR **P2+1** will be used.
P4 - opt Destination B Image-Register number, if omitted, IR **P3+1** will be used.

The Command will leave a 0/1 on the TOS. As always, a 0 means Error, while a 1 means "all fine".

Example

```
'**********************************
' ANA.SplitRGB and MixRGB-Sample
'
'**********************************
$$FIA=?pattern\SarahC_640.jpg
ANA.Load|0|$$FIA
ANA.SplitRGB|0|1
ANA.Show|1!
ANA.Show|2!
ANA.Show|3!
' Now mix the 3 Parts together in IR 6
ANA.MixRGB|6|1
ANA.Show|6!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.5 GRA. - Picture/Image Commands

3.55.5.1 GRA.GetResolution

GRA. - Graphic Command: GetResolution Previous 1836 Top 41 Next 1846

MiniRobotLanguage (MRL)

GRA.GetResolution Command

Get Image X- and Y-Resolution

Intention

`GRA.get` will compute the X- and Y-Resolution of an Image in Pixels.

The following file-Types can be specified as Source file:

- BMP
- EMF
- JPG or JPEG
- GIF
- ICO
- PNG
- TIF or TIFF
- WMF

Usage is as simple as:

```
$$FIL=?desktop\MyPic.BMP
' $$XRE will get the X-Resolution, $$YRE will get the Y-Resolution
GRA.get|$$FIL|$$XRE|$$YRE
MBX.Resolutiuon is: $$XRE,$$YRE
```

Stack Usage / Return values:

Any `GRA.`-Operation will leave one or more values on the stack, depending on the success of the Operation.

Generally if there is a "0" on TOS then the Operation was successful. If you do not specify the Variables for the X- and Y-Resolution,

these values will be placed on the Stack also.

```
VAR.$$SRC=?path\DSC01987.JPG
STV.TOVCLEAR|$$DUM
GRA.get|$$SRC
DMP.6
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt       :0

Stackpos: 001 ->LongInt       :3672

Stackpos: 002 ->LongInt       :4896

------------------------------------
```

Here is an explanation of the stack values:

Position: Explanation:

```
    0 (TOS)  Is 0 if the GRA.-Operation was successful. Any other value may indica
    1       Y-Resolution
    2       X-Resolution
```

In case you specify only one of the two variables, the other value will be placed on the Stack.

Syntax

GRA.get|P1[|P2][|P3]
GRA.GetResolution|P1[|P2][|P3]

Parameter Explanation

P1 - Source file.

P2 - Variable for X-Resolution (Out). If this variable is omitted or -, the value is placed on the Stack.

P3 - Variable for Y-Resolution (Out). If this variable is omitted or -, the value is placed on the Stack.

Example:

```
    GRA.get|?desktop\MyFile.BMP|$$XRE|$$YRE
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
      ' have the Y-Resolution on the Stack
    GRA.get|$$FIL|$$XRE

      ' have the X-Resolution on the Stack
    GRA.get|$$FIL|-|$$YRE
```

See above which values are placed on the Stack during the Operation.

Example

```
VAR.$$SRC=?path\DSC01987.JPG
' Folgende Zeile löscht den Stack.
STV.TOVCLEAR|$$DUM
' Hier gehts los
GRA.get|$$SRC|$$YRE|$$XRE
' Diese Zeile zeigt den Stack an
DMP.6
MBX.$$XRE-$$YRE
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.8. Selecting the Application (Topwindow) 150
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.5.2 GRA.Scale

MiniRobotLanguage (MRL)

GRA.scale Command

Scale Image Size

Intention

`GRA.scale` will resize an image, keeping the Aspect Ratio. For this the Original image will be copied to a new file with the same or with another file extension.

The following file-Types can be specified as Source or destination file:

- `BMP`
- `EMF`
- `JPG or JPEG`
- `GIF`
- `ICO`
- `PNG`
- `TIF or TIFF`
- `WMF`

The resolution in Y-direction will automatically be calculated in a way that the original Apect-Ratio is been preserved.

To convert a .BMP-Picture to .JPG you could write:

```
' This will create a File MyPic.JPG on the Desktop
' Because this is where the Source file also comes
' from.
$$FIL=?desktop\MyPic.BMP
GRA.sca|$$FIL|JPG|1024#
```

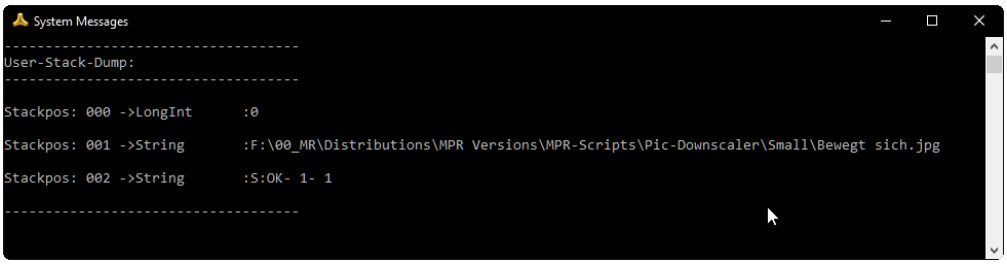
Instead of just writing the new Filetype as Destination, you can give a complete Filename and/or path.

```
GRA.sca|$$FIL|?desktop\MyFile.BMP
' The following two commands let you see the STACK content.
DMP.6
MBX.!
```

Stack Usage / Return values:

Any `GRA.`-Operation will leave several values on the stack, depending on the success of the Operation.
Here is an Example:

© 2023 Theo Gottwald * https://www.smart-package.com

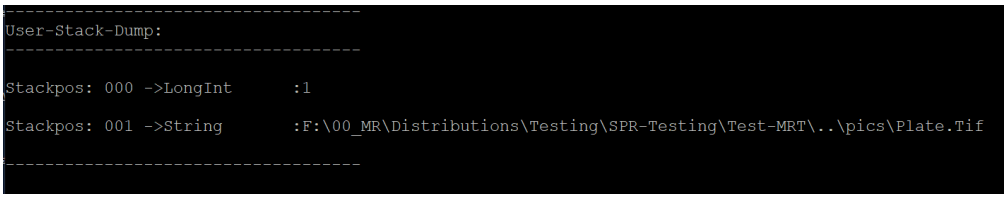


Here is an explanation of the stack values:

Position: Explanation:

```
    0 (TOS)  Is 0 if the GRA.-Operation was successful. Any other value may indica
    1        Path and Filename of the converted destination file.
    2        S:OK if all was fine or can be E:Error-Message if there was an Proble
```

In case of an parameter error - or an unusable destination path - the stack will just look like this.



Syntax

GRA.sca|P1|P2[|P3][|P4]
GRA.scale|P1|P2[|P3][|P4]

Parameter Explanation

P1 - Source file.

***P2* - Destination File Path this can be:**

a) complete path with Filename and extension

```
GRA.sca|$$FIL|?desktop\MyFile.BMP|1280#
```

b) just the extension of the new file like "JPG"

```
GRA.scale|$$FIL|BMP
```

c) Only a new Filename with Extension. In this case the Source-File Path is used for the destination file.

```
GRA.sca|$$FIL|MyFile.BMP|1280#
```

d) Omit the filetype. In this case ".jpg" is used as default.

```
GRA.sca|$$FIL|?desktop\MyFile
GRA.sca|$$LAC|MyFile
```


© 2023 Theo Gottwald * https://www.smart-package.com

P3 **-** X-Resolution of new Image. Y-Resolution is automatically choosen
using the Aspect-ratio of the Original Image.
If parameter is omitted, Original Image Resolution is taken.
Here you can specify the following 2 additional Parameters:
- This will allow to upscale pictures to a higher resolution. If omitted, the maximum resolution is the Original resolution.
@ - This will force the Process to keep the original resolution of the Source file

P4 **-** (optional) JPG - Compression Quality from 1 to 100.
If parameter is omitted, "75" is taken.

Example:
```
' Activate Up-Scaler
GRA.sca|$$FIL|?desktop\MyFile.BMP|1280#

' Force Original resolution
GRA.sca|$$FIL|?desktop\MyFile.BMP|@
```

See above which values are placed on the Stack during the Operation.

Example

```
' Dieses Beispielscript muß kompiliert werden.
' zieht man dann eine Grafikdatei auf die Verknüpfung der kompilierten EXE-Datei,
' dann wird dort wo die Grafikdatrei liegt ein Ordner "SMall" erstellt und in
' diesen Ordner wird eine verkleinerte Kopie der Grafikdatei gelegt.
'
'**********************************
 ' +++ SF_Drag Pic_Scaler.mrt +++
 ' Zweck:
 ' erstellt am: 16.04.2018
 ' von: Theo Gottwald
 '========================
 '#EXE:?path\
 '#SPI:ForceWrite
 '#ICS:5
 '#MAN:ai
 '
 MRK.2
 ' Ziel X- Auflösung,
 ' Y-Auflösung wird intern anhand Aspekt-Ratio berechnet
 VAR.$$RES=503
 ' Qualitätseinstellung (1-100)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$QUA=85

VAR.$$SRP=$cmdexe$
'VAR\$$SRP=?path\Kepler.jpg

VBT.$$SRP|"
'MBX\Bilder-Ordner: $crlf$$$SRP

NEF.$$SRP
  GTO.enx
EIF.

GFT.c|$$SRP|$$PTH
VAR.$$DEP=$$PTH\Small
'MBX\$$DEP
MKD.$$DEP
'FEF\$$SRP|*.jpg|Doit|$$SRC
VAR.$$SRC=$$SRP

GSB.Doit
:enx
MBX.!
ENR.

:Doit
SBT.$$SRC|T:2 FS:14 MP: NC:
IVC.$$SRC=$$DEP|e
  GTO.over
EIF.
GFT.je|$$SRC|$$EXT|$$FIN
' Dateiformat beibehalten
VAR.$$DES=$$DEP\$$FIN.$$EXT

' Immer nach "Jpg" Konvertieren
VAR.$$DES=$$DEP\$$FIN.jpg
'MBX.$$SRC$crlf$$$DES$crlf$$$QUA$crlf$$$RES
STV.TOVCLEAR|$$DUM
GRA.sca|$$SRC|$$DES|$$RES|$$QUA
:over
RET.
```

Remarks

-

Limitations:

A Quadratic Image should be upscaled up to about 26000 Pixels X/Y-Size that is 676 MegPixel.
Larger sizes may work or fail, depending on the Memory Situation in Windows.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- **1.8. Selecting the Application (Topwindow)** 150
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.5.3 GRA.Convert

MiniRobotLanguage (MRL)

GRA.convert Command

Convert Image

Intention

`GRA.con` will convert the format of an image into an other format. For this the Original image will be copied to a new file with another file extension.

The following file-Types can be specified as Source or destination file:

- `BMP`
- `EMF`
- `JPG or JPEG`
- `GIF`
- `ICO`
- `PNG`
- `TIF or TIFF`
- `WMF`

If you specify P$ then you can also specify a new file-resolution in X-direction. The resolution in Y-direction will automatically be calculated in a way that the original Apect-Ratio is been preserved.

To convert a .BMP-Picture to .JPG you could write:

```
' This will create a File MyPic.JPG on the Desktop
' Because this is where the Source file also comes
' from.
$$FIL=?desktop\MyPic.BMP
GRA.con|$$FIL|JPG
```

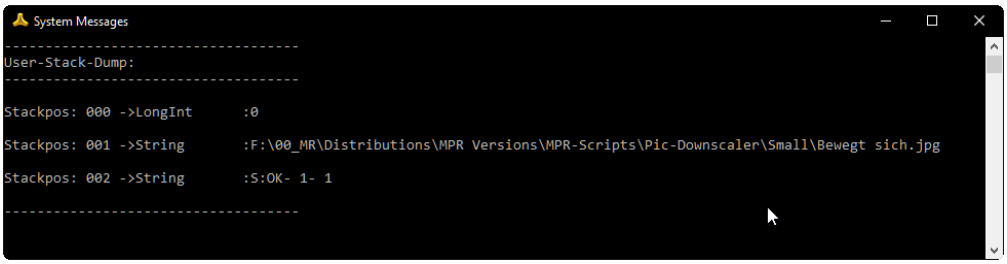
Instead of just writing the new Filetype as Destination, you can give a complete Filename and/or path.

```
GRA.con|$$FIL|?desktop\MyFile.BMP
' The following two commands let you see the STACK content.
DMP.6
MBX.!
```

Stack Usage / Return values:

Any `GRA.`-Operation will leave several values on the stack, depending on the success of the Operation.
Here is an Example:

© 2023 Theo Gottwald * https://www.smart-package.com

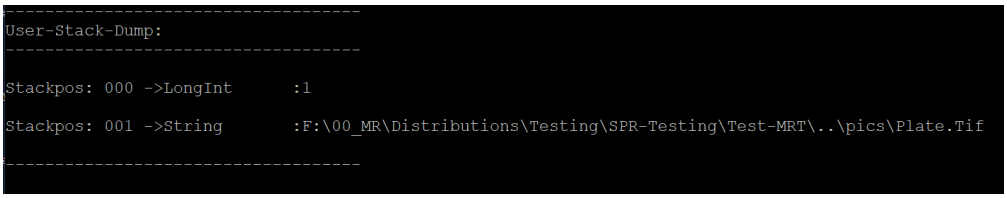


Here is an explanation of the stack values:

Position: Explanation:

```
    0 (TOS)  Is 0 if the GRA.-Operation was successful. Any other value may indica
    1        Path and Filename of the converted destination file.
    2        S:OK if all was fine or can be E:Error-Message if there was an Proble
```

In case of an parameter error - or an unusable destination path - the stack will just look like this.



Syntax

GRA.con|P1|P2[|P3][|P4]
GRA.convert|P1|P2[|P3][|P4]

Parameter Explanation

P1 - Source file.

***P2* - Destination File Path this can be:**

a) complete path with Filename and extension

```
GRA.con|$$FIL|?desktop\MyFile.BMP|78|1280#
```

b) just the extension of the new file like "JPG"

```
GRA.con|$$FIL|BMP
```

c) Only a new Filename with Extension. In this case the Source-File Path is used for the destination file.

```
GRA.con|$$FIL|MyFile.BMP||1280#
```

d) Omit the filetype. In this case ".jpg" is used as default.

```
GRA.con|$$FIL|?desktop\MyFile
GRA.con|$$LAC|MyFile
```


© 2023 Theo Gottwald * https://www.smart-package.com

P3 **-** (optional) JPG - Compression Quality from 1 to 100.
If parameter is omitted, "75" is taken.

P4 **-** X-Resolution of new Image. Y-Resolution is automatically choosen
using the Aspect-ratio of the Original Image.
If parameter is omitted, Original Image Resolution is taken.
Here you can specify the following 2 additional Parameters:
- This will allow to upscale pictures to a higher resolution. If omitted, the maximum resolution is the Original resolution.
@ - This will force the Process to keep the original resolution of the Source file

Example:
```
' Activate Up-Scaler
GRA.con|$$FIL|?desktop\MyFile.BMP||1280#

' Force Original resolution
GRA.con|$$FIL|?desktop\MyFile.BMP|75|@
```

See above which values are placed on the Stack during the Operation.

Example

```
' Dieses Beispielscript muß kompiliert werden.
' zieht man dann eine Grafikdatei auf die Verknüpfung der kompilierten EXE-Datei,
' dann wird dort wo die Grafikdatrei liegt ein Ordner "SMall" erstellt und in
' diesen Ordner wird eine verkleinerte Kopie der Grafikdatei gelegt.
'
'**********************************
 ' +++ SF_Drag Pic_Scaler.mrt +++
 ' Zweck:
 ' erstellt am: 16.04.2018
 ' von: Theo Gottwald
 '========================
 '#EXE:?path\
 '#SPI:ForceWrite
 '#ICS:5
 '#MAN:ai
 '
 MRK.2
 ' Ziel X- Auflösung,
 ' Y-Auflösung wird intern anhand Aspekt-Ratio berechnet
 VAR.$$RES=503
 ' Qualitätseinstellung (1-100)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$QUA=85

VAR.$$SRP=$cmdexe$
'VAR\$$SRP=?path\Kepler.jpg

VBT.$$SRP|"
'MBX\Bilder-Ordner: $crlf$$$SRP

NEF.$$SRP
  GTO.enx
EIF.

GFT.c|$$SRP|$$PTH
VAR.$$DEP=$$PTH\Small
'MBX\$$DEP
MKD.$$DEP
'FEF\$$SRP|*.jpg|Doit|$$SRC
VAR.$$SRC=$$SRP

GSB.Doit
:enx
MBX.!
ENR.

:Doit
SBT.$$SRC|T:2 FS:14 MP: NC:
IVC.$$SRC=$$DEP|e
  GTO.over
EIF.
GFT.je|$$SRC|$$EXT|$$FIN
' Dateiformat beibehalten
VAR.$$DES=$$DEP\$$FIN.$$EXT

' Immer nach "Jpg" Konvertieren
VAR.$$DES=$$DEP\$$FIN.jpg
'MBX.$$SRC$crlf$$$DES$crlf$$$QUA$crlf$$$RES
STV.TOVCLEAR|$$DUM
GRA.con|$$SRC|$$DES|$$QUA|$$RES
:over
RET.
```

Remarks

-

Limitations:

A Quadratic Image should be upscaled up to about 26000 Pixels X/Y-Size that is 676 MegPixel.
Larger sizes may work or fail, depending on the Memory Situation in Windows.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- **1.8. Selecting the Application (Topwindow)** 150
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.5.4 GRA.vToBMP

MiniRobotLanguage (MRL)

GRA.vToBMP Command

Convert Variable that contains an Image to the internal Bitmap-Format for use with ANA.-Commands

Intention

This Command is helpful, especially together with "ANA.#FromVar". GRA.vbm will convert the format of an image into the internal Bitmap-Format. The result will also be stored in a Variable.
This way you can use all Sorts of Images (see below) together with the ANA.-Commands, especially using "ANA.#FromVar".

The following file-Types can be specified as Source Formats:

- BMP
- EMF
- JPG or JPEG
- GIF
- ICO
- PNG
- TIF or TIFF
- WMF

If you specify P$ then you can also specify a new file-resolution in X-direction. The resolution in Y-direction will automatically be calculated in a way that the original Apect-Ratio is been preserved.

To convert a loaded .jpg you could write:

```
' Load a JPG-File, here from "Inline-Data", see Inline Data Commands 3483
GSB.GetPic
' Convert jpg to internal Bitmap Format
GRA.vbm|$$DTA|$$BMP|.jpg
' Use ANA.Command to directly load the Imagine int an Image Register
ANA.#FromVar|1|$$BMP
' Show the Image
ANA.Show|1!
ENR.
```

This may also be necessary for several types of ".bmp"-Files that do not match the internal specifications of the "ANA.#FromVar" Command.
In this case we just convert from ".bmp" to ".bmp" to be able to use the Picture.

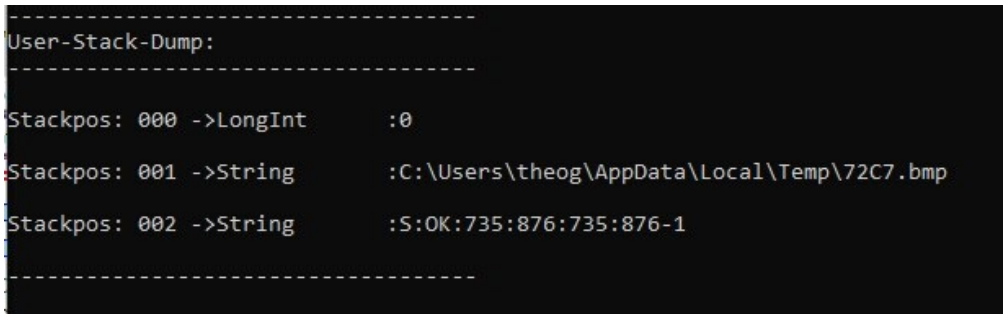
© 2023 Theo Gottwald * https://www.smart-package.com

```
' Load a JPG-File, here from "Inline-Data", see Inline Data Commands 3483
GSB.GetPic
' Convert .bmp to internal Bitmap Format
GRA.vbm|$$DTA|$$BMP|.bmp
' Use ANA.Command to directly load the Imagine int an Image Register
ANA.#FromVar|1|$$BMP
' Show the Image
ANA.Show|1!
ENR.
```

Stack Usage / Return values:

This GRA.-Operation will leave several values on the stack, depending on the success of the Operation.

Most Important is the most upper value that is a "0" if the Operation was successful.



Internally used temporary files will automatically be deleted.

Syntax

GRA.vbm|P1|P2[|P3][|P4]
GRA.vToBMP|P1|P2[|P3][|P4]

Parameter Explanation

P1 - Source file data inside a variable. Can be from Inline-Data, or you can load a picture using CFF.-Command.

P2 - Destination Variable. This Variable will receive the Data that can be used together with ANA.#FromVar

P3 - Filetype of P1-Data. Like ".bmp",".png",".jpg"

P4 - X-Resolution of new Image. Y-Resolution is automatically choosen using the Aspect-ratio of the Original Image.
If parameter is omitted, Original Image Resolution is taken.

© 2023 Theo Gottwald * https://www.smart-package.com

```
     Here you can specify the following 2 additional
Parameters:
     # - This will allow to upscale pictures to a higher
resolution. If omitted, the maximum resolution is the Original
resolution.
     @ - This will force the Process to keep the original
resolution of the Source file
```

Example

```
  '################################
  ' GRA.vbm-Sample
  '################################
  CFF.Mypicture.jpg|$$DTA
  GRA.vbm|$$DTA|$$BMP|.bmp
  ANA.#FromVar|1|$$BMP
  ANA.Show|1!
  ENR.
```

Remarks

-

Limitations:

A Quadratic Image should be upscaled up to about 26000 Pixels X/Y-Size that is 676 MegPixel.
Larger sizes may work or fail, depending on the Memory Situation in Windows.

See also:

- 1.8. Selecting the Application (Topwindow) 150
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6 Non-IR Operations

These are mostly older Commands taht are not using the newer Image-Registers (IR).

Possibly Commands like WRC. and WRA. should not be used, now that we have the new Pattern-Commands.
The contained GPC. and SPC.-Commands may also not make sense anymore and do only work under few circumstances.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.1 GPC. - Get-Pixel-Colour

MiniRobotLanguage (MRL)

GPC. Command

Get Pixel Color

Intention

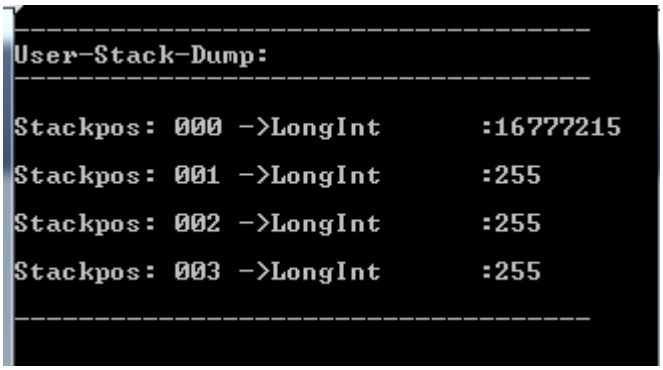
This command reads the screen color of a pixel in a window. You can specify coordinates in Screen- (desktop), or in window-coordinates.

You will also get the red, green and blue parts of the color (in this order) on the Stack 369.

In the simplest way you can check the Pixel-color at some point like this:

```
GPC.624,794
STS.DUMP
```

And this is what you may find on TOS.



First on Top, the 24-bit color value. Then next, the Blue-(`$$001`),Green-(`$$002`), and the Red (`$$003`)-Part of the Pixel-color.
If you just need the color value, you can get it using the `$$000` Stackvariable.

```
GPC.624,794
PRT.The 24-bit color is: $$000
STS.DUMP
MBX.!
```

Or you can just add one Parameter with a variable. The following Sample shows it. And it also shows that you can use a Point-Variable instead of two coordinated here.

```
VAP.$$COR=624,794
GPC.$$COR|$$COL
PRT.The 24-bit color is: $$COL
STS.DUMP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.!
ENR.
```

Syntax

GPC.[P1][|P2]
GPC.[proto]|P1,P2

Parameter Explanation

P1 - (optional) prototype coordinates
P2 - (optional) coordinates separated with ","
P3 - (optional) takes the pixel-colour, if omitted stack is used.

There are three prototypes:
- **c** - **C**hildwindow Coordinates (you can use "o" for this one also.)
- **a** - **a**ctual window coordinates
- **t** - **T**op-Window Coordinates
If prototype is omitted, Screen-Coordinates are taken.

Example

```
'***********************************
' GPC.-Demo
'***********************************
GPC.1028,1370
STS.DUMP
$$COL=$$000
VTH.$$COL
PRT.The 24-bit color is: $$COL
MBX.!
END.

'***********************************
' GPC.-Demo
'***********************************
JSR.closeNotepads
EXO.notepad
STW.p|#pid#
SCW.c|edit
MMV.o|50,100
STS.CLEAR
' Get colour from Child-Window-Coord's
VAP.$$AAA=50,100
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GPC.c|§§AAA
JSR.showcol

' Get colour from Top-Window-Coord's
VAP.§§AAA=50,100
GPC.t|§§AAA
JSR.showcol

' Get Screen-Color
VAP.§§AAA=50,100
GPC.s|§§AAA
JSR.showcol
MBX.!

' Get Screen-Color and show Stack-Order
VAP.§§AAA=50,100
GPC.s|§§AAA
STS.DUMP
MBX.!

' Show Speed-Dump
DMP.

MBX.!
END.

:showcol
PRT.----------------------
PRT. Red is: $$003
PRT. Green is: $$002
PRT. Blue is: $$001
$$COL=$$000
VTH.$$COL
PRT. Pixel-Color is: $$COL
PRT.----------------------
RET.

:closeNotepads
STW.ctwo|notepad|Editor|0|overcn
CLW.
JMP.closeNotepads
:overcn
RET.
```

Remarks

Instead of using separate variables for X and Y, you can also use a Point-Variable, where x and y values are separated by a "," as follows:

```
VAR.§§PAP=10,10
GPC.§§PRE|§§PAP|§§COL
```

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

The RGB-Splitter works default with 24- or 32 bit colors. It will not work with 8 or 16 bit color depth.

See also:

- **1.8. Selecting the Application (Topwindow)** 150
- IPC. / NPC. - If-Pixel-Colour 1816
- SPC. -Set-Pixel-Colour 1843

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.2 ILC. / NLC. - If Line Color

MiniRobotLanguage (MRL)

ILC. / NLC. Conditional Statement

If Line Color

Intention

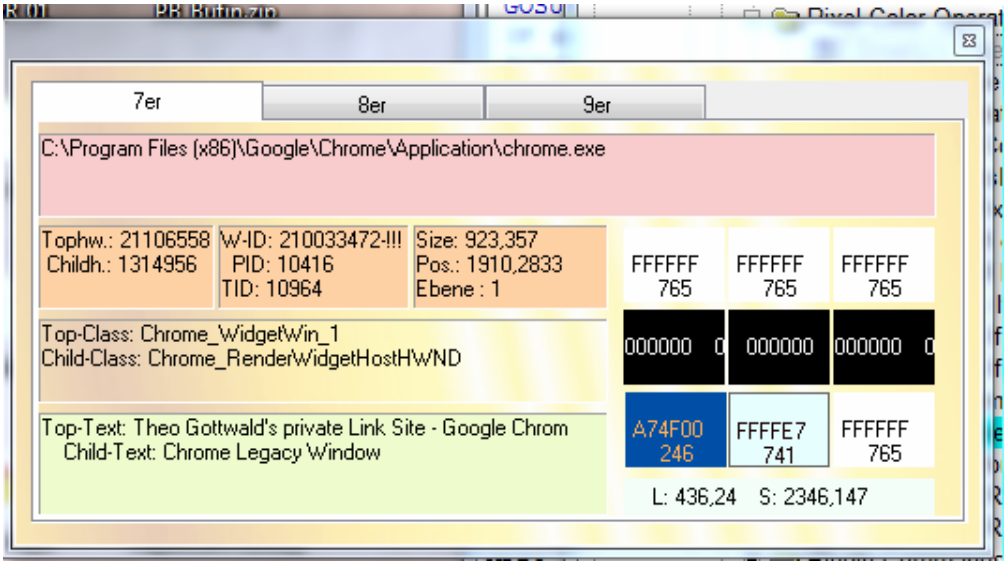
Conditional Statement.

Test if an defined Color is been found along a defined Line.
Or, test how often a color does not match a defined color along a given line.
This is done using the "**!**" Operator.

This command can be used to locate buttons in graphical environments, where we can not use any other commands to locate them.

Important: Color-Operations are often troublesome because colors may not match exactly. Therefore this command provides you with a "Color Tolerance" Option.
Also you have the choice of using a "negative version" which will just check that colors do NOT match a given color/tolerance. And you can combine all this, with the resolution independent SPR 3.6 Using Quadrant-Coordinates 356 to make your Scripts run as compatible as possible.

Hint:
When Designing Pixel-Operations, use the "Catseye-Analyzer".
It will show you Color-Values and other needed Information.
The "Catseye-Analyzer" is located in the Installation Folder.



*© 2023 Theo Gottwald * https://www.smart-package.com*

The four parameters are line-coordinates, Color, Color-Tolerance and Hit-Counter.

Parameter 1 - Line-Coordinates:
The Line is defined by a Start- and and Endpoint. Using the following Format:

```
ILC.x1,y1,x2,y2
```

As always, if you omit `y2` then `y2` is assumed to be equal to `y1`.
If you omit the coordinates, the last located window or other object-rectangle is been taken, and the line goes from the upper-left to the lower-right corner.

Instead of using Pixel-Coordinates, you can use 3.6 Using Quadrant-Coordinates 356 to make your Scripts run as compatible as possible.
Example:

```
ILC.{A1,A2}
```

Now writing this will use the Screen-Quadrant A1, which is most often not what we want. So we add an letter in front:

```
ILC.w{A1,A2}
```

This will tell the command that we want to use the Quadrant in the last located window (child- or Topwindow).
You can use:
"**w**" - Any window (Top- or Child-Window) that was last located
"**t**" - last located Top-Window
"**c**" - use Client-Coordinates of tzhe specified window. If no window is specified "w" is used as default.
"**o**" - use the last defined Object-Coordinates (window, Accessible Object or graphical Object).
You can combine the letters as far as it makes sense:
"**wc**" - use last windows Client-Coordinates
"t**c**" - use last Top-windows Client-Coordinates

at this place you can add 2 Special Characters:
"**@**" - Enable Debug Option. Using this special Character will open a Window that shows the part of the Screen that is used for the Command. Be careful, when using `WRC.` with this Option, as it will of course run in a Loop.
"#" - this Option will reduce the Border of the Target Area by 2 Pixels in all directions.
"**##**" - using multiple "#" will reduce the Border of the Target Area by 2 Pixels times the number of "#" in all directions. If you use "##" it will decrease the target size by 4 Pixel.
"**#8**" - using "#" followed by a numeric value, the border of the Target Rectangle will be decreased by the given Amount of Pixels. In this case the border would be decreased by 8 Pixels, from any direction.

You can omit the Quadrant coordinates and just write (using any of the letters from above):

```
ILC.w
```

Then internally the command will complete the target with

```
ILC.w{A1,D4}
```


© 2023 Theo Gottwald * https://www.smart-package.com

General Usage is as simple as:

```
' Using ILC. with a Edit-Box from TestApp
'
STW.ct|PBWindowClass:0|TestApp
SCW.nict|1|1023|Edit|{&NOTEXT:}
' Clear Edit-Box
SDT.
ILC.
  VAR.$$LAA=1
ELS.
  VAR.$$LAA=0
EIF.
MBX.Result is $$LAA
' Now we write into the Edit-Box
SDT.Hallo
ILC.
  VAR.$$LAA=1
ELS.
  VAR.$$LAA=0
EIF.
MBX.Result is $$LAA
ENR.

STW.ct|MyClassName|Sample Application
'SCW.nct|1|Edit|{&NOTEXT:}
:again
PAU..1
ILC.|&HFF0000|0|4
  DMP.6
  MBX.!
ELS.
  GTO.again
EIF.
:over
ENR.
```

Using local (window-) coordinates:

If you prefix the coordinates with a "w" then they will be interpreted as local window coordinates. Like in this example:

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
:again
PAU..1
ILC.w1,1,10,10|!&HFFFFFF|0|3
  GTO.over
ELS.
  DMP.6
  GTO.again
EIF.
:over
DMP.6
MBX.Ready
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Using SPR 3.6 Quadrant-Coordinates

```
' Looks like this
ILC.{D3}|$$COL|$$TOL
  DBP. 1
ELS.
  DBP. 0
EIF.
```

Parameter 2 - Color:

If this color is found along the defined Line then the decision will switch to "true". Colors can most easily be described as a HEX-Number, like this example:

```
&HFFEE22
&HRRGGBB
```

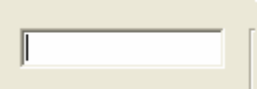
If you prefix the color-parameter with a "**!**" (NOT Operator) then the command will count all Pixels that do NOT have that defined color and tolerance.
Please note that this is different from the `NLC.`-command, which gives you additional options.

Parameter 3 - Color Tolerance:

If there are different graphic cards used, the pixels may not become rendered always in the exactly same way. Therefore the color tolerance gives you the chance to tell the robot "How sensitive" he should be when deciding **if a pixel is different or not.**
To make this decision, the robot will take the color of both pixels and calculate their "Color distance", that is "How similar the colors are". As a result he will get a number that can vary from 0 to 441673. Next Smarty will get your tolerance value and check if the calculated color distance is inside the chosen tolerance or not. Alternatively you can use a percentage value, if you add a %-sign after the number.

Parameter 4 - Hits-Counter:

Imagine a Textbox like this:



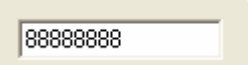
You want to wait until there is a text inside, but the blinking cursor should **not** make your Script react. Now what you tell Smarty is a "Hit Count" of lets say 12 Pixels.

This Script will switch, after you really typed "`88888888`" into the first, left textbox.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:again
PAU..1
ILC.|0|0|12
  GTO.over
ELS.
  GTO.again
EIF.
:over
ENR.
```



This makes Smarty count the changed pixels, and only if she gets more then "Hit Count" she will switch. Otherwise the condition will not become true, even if there are changed pixels (but less).

The "negative Mode" using the "!" NOT Operator

Instead of counting the Pixels that do have a defined color, you can also do the opposite.
You can count the Pixel that do NOT have that defined color.
This is done, by prefixing the "Color-Parameter" with a **!**.
Like in this example.

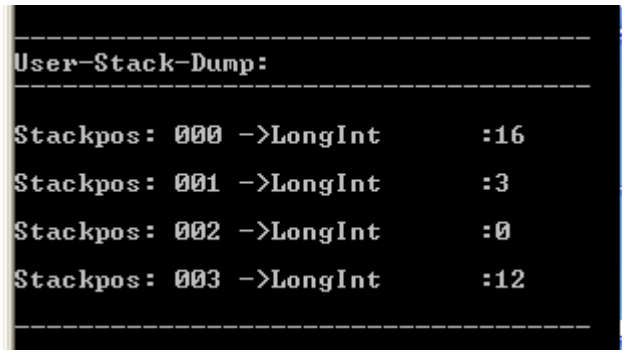
```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
:again
PAU..1
ILC.|!&HFFFFFF|0|12
  GTO.over
ELS.
  DMP.6
  GTO.again
EIF.
:over
ENR.
```

The Stack when using ILC.

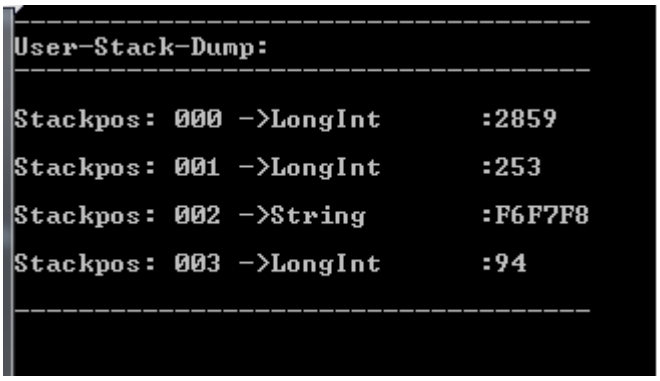
In case of success, `ILC.` will leave **four values on the stack**. As you can see below, these are:

```
TOS: X-Position of first found Hit
001: Y-Position of first found Hit
002: Color of first found Hit
003: Number of found Pixel meeting the condition
```


© 2023 Theo Gottwald * https://www.smart-package.com



Here is another example.
The color is in hexadezimal in RRGGBB.



And there is `NLC.`

`NLC.` is the negative version "If Not Line Color".

Syntax

```
ILC.[w/t/c/o][P1][|P2][|P3]…
ELS. … EIF.
```

Parameter Explanation

P1 **-** ***(optional)*** Start and End-Points of the Line. There are different possible Formats for these Coordinates. For details see description above. You can use:
"**w**" - Any window (Top- or Child-Window) that was last located
"**t**" - last located Top-Window
"**c**" - use Client-Coordinates of tzhe specified window. If no window is specified "w" is used as default.
"**o**" - use the last defined Object-Coordinates (window, Accessible Object or graphical Object).
You can combine the letters as far as it makes sense:
"**wc**" - use last windows Client-Coordinates
"**tc**" - use last Top-windows Client-Coordinates

© 2023 Theo Gottwald * https://www.smart-package.com

at this place you can add 2 Special Characters:
"**@**" - Enable Debug Option. Using this special Character will open a Window that shows the part of the Screen that is used for the Command. Be careful, when using `WRC.` with this Option, as it will of course run in a Loop.
"**#**" - this Option will reduce the Border of the Target Area by 2 Pixels in all directions.
"**##**" - using multiple "#" will reduce the Border of the Target Area by 2 Pixels times the number of "#" in all directions. If you use "##" it will decrease the target size by 4 Pixel.
"**#8**" - using "#" followed by a numeric value, the border of the Target Rectangle will be decreased by the given Amount of Pixels. In this case the border would be decreased by 8 Pixels, from any direction.

With out the leading "`w`", any coordinates are Screen-Coordinates. Its best practice, to prefer Child-Window-relative coordinates:
`ILC.w18,56,153,57|$$COL|$$TOL`

You can simply specify the start and end-point. In this format:
`ILC.x1,y1,x2,y2`
a real example would be:
`ILC.10,15,100,200`
Or you can specify a Rectangle, using only 3 Numbers like:
`ILC.10,15,100`
then `y2` is assumed to be the same as `y1.`

You can also use 3.6 Using Quadrant-Coordinates 356.
In this case the Line will be drawn from the upper-left corner straight to the lower right corner.

```
ILC.{D1,D2}|$$COL|$$TOL
ILC.{B1}|$$COL|$$TOL
```

If **P1** is omitted the Rectangle from the last localized Object or window is taken, same as "w".

P2 - (optional) numeric/variable contains the color value to scan for.

P3 - (optional) numeric/variable contains the Tolerance to be used when checking if a color is different. You can specify the color tolerance as a numeric value from 0 - `441673`, or you can specify a percentage from `0%` to `100%`. If you specify a percentage, just add the **%** Sign after the number. The color tolerance is the maximum allowed Color-Distance between two colors. The maximum color distance is **441673** that is the **distance between black and white**. If you omit this parameter, the system will use a zero Tolerance. In this case the colors are just "equal or not equal".

P4 - (optional) numeric/variable contains the minimum number of "Hits" that are pixels that have to be different to make the decision to become "`true`".

`ILC.`'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX.` (Enumerations) to unlimited Depth.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

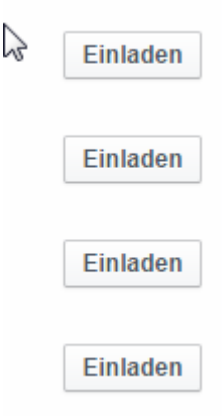
```
'**********************************
' ILC.-Sample, using Quadrant Coords
'**********************************
$$COL=!&HFFFFFF
$$TOL=100

STW.ct|Chrome_WidgetWin_1|- Google Chrome
SCW.c|renderwi

ILC.w{A1,B1}|$$COL|$$TOL
  STS.DUMP
  $$XPA=$$000
  $$YPA=$$000
  MMV.w|$$XPA,$$YPA
ELS.
  DBP.$time$
EIF.
:outA
MBX.!

'**********************************
' ILC.-Sample 2
' using window-local Coordinates
'**********************************

$$COL=!&HFFFFFF
$$TOL=100

STW.ct|Chrome_WidgetWin_1|- Google Chrome
SCW.c|renderwi

ILC.w18,56,153,57|$$COL|$$TOL
'ILC.{D1}|$$COL|$$TOL
  STS.DUMP
  $$XPA=$$000
  $$YPA=$$000
  MMV.$$XPA,$$YPA
ELS.
  DBP.$time$
EIF.
:outA
MBX.!
ENR.

'**********************************
' ILC.-Sample
' using a Diagonal Line from Top-left to lower right
' of the actual window(child-window
'**********************************

$$COL=!&HFFFFFF
$$TOL=100

STW.ct|Chrome_WidgetWin_1|- Google Chrome
SCW.c|renderwi
ILC.t|$$COL|$$TOL
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  STS.DUMP
  $$XPA=$$000
  $$YPA=$$000
  MMV.$$XPA,$$YPA
ELS.
  DBP.$time$
EIF.
:outA
MBX.!
ENR.

'**********************************
' ILC.-Sample
'**********************************
' Use "Sample Application"
' Click into the left Edit-Field to end the Loop
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
:again
PAU..1
' Coordinates omitted - he'll take the last window-Rect
ILC.|&H0|0|4
  DMP.6
  MBX.!
ELS.
  GTO.again
EIF.
:over
ENR.

'**********************************
' ILC.-Sample 2
'**********************************
' Search for any "Non-white" color
$$COL=!&HFFFFFF
' with a color Tolerance of 100
$$TOL=100

' left X-Position in window-Coordinates
$$XPO=303
' upper Y-Position in window-Coordinates
$$YPO=203
' lower Y-Position
$$YMX=365
' vertical Size is calculated
CAL.$$YSZ=$$YMX-$$YPO
' Locate Target window
STW.ct|Chrome_WidgetWin_1|- Google Chrome

' Do it again ...
:again
ILC.w$$XPO,$$YPO,($$XPO+20),($$YPO+$$YSZ)|$$COL|$$TOL
  ' This will show the results from ILC. in a Debug-COnsole
  STS.DUMP
  ' When we take the first Coordinate from stack
  ' its automatically removed, so the next coordinate
  $$XPA=$$000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  ' is also on Top of stack
  $$YPA=$$000
  ' Here we click the button
  ' note the () for the inline-calculation
  MLI.$$XPA,($$YPA+1)
  ' 70 is expected y-size of a button
  ' we increment and look for the next button
  CAL.$$YPO=$$YPO+70
  ' if we exceed the lower maximum coordinate, we jump out
  JIV.$$YPO>$$YMX|out
  ' else we do it again
  GTO.again
ELS.
  DBP.$time$
EIF.
:out
MBX.!

'-------------------------------------------------------------
ENR.

' This the situation i have used for testing.
```



`ILC.`-Example 2 was used to click several Buttons that could not be located (graphical buttons) in a Browser-Window.

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Pixel-Operations 1088
- 3.6 Using Quadrant-Coordinates 356
- WLC. - Wait for Line Color 1846
- RRA. - Remember Rectangular Area 1836
- IRA. / NRA. - If Rect altered 1824
- WRA. - Wait Rect-Area Altered 1851
- 1.8.1 The "Client" and the "Non-Client" Parts of a window 156
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.3 IPC. / NPC. - If-Pixel-Colour

MiniRobotLanguage (MRL)

IPC. / NPC. Conditional Statement

If Pixel Color

Intention

Conditional Statement. Test if a specified screen-pixel has a specified color or not. Useful in cases, where the window controls can not be accessed in any other way.

This command provides a lot of operation modes, to be as compatible as possible with changing desktop-themes, different video-cards etc. After all, this command should only be used as the last choice if nothing else will help you.
The reason is, that any dependency on pixel-colors may not work in cases of changing resolution, other **dpi**-settings or other video-cards or drivers.

Therefore it is recommended that you locate the pixel always window-relative instead of using screen coordinates. This is done if you specify the **|o** before the comma separated coordinates. In that case, the coordinates will be relative to the last located child- or top-window. Let's start with the very simple case of:

```
' Check if the colors will match exactly!
IPC.o|23,34|=$$COL
   DBP.I am in here!
EIF.
```

You can use the = (equal) or the ! (unequal) operator to check an exact match of colors.

```
 IPC.[o]|x,y|Formula
 IPC.o|23,34|=COL ' equal
 IPC.o|23,34|!COL ' unequal
```

In real life, white is not always white (`&HFFFFFF`). And black is not always really black (`&H0`). There need to be tolerances. They are implemented using the **w** (RGB<5) and the **b** (RGB>252) operators.

```
IPC.o|23,34|=w ' Is white
IPC.o|23,34|=b ' Is Black
IPC.o|23,34|=!w ' Is not white
IPC.o|23,34|=!b ' not black
```

Sometimes, you just need to know if a color is lighter or darker then another color. This is done by adding the RGB values and comparing the results.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
IPC.o|23,34|<COL ' dunkler als COLOR
IPC.o|23,34|>COL ' heller als COLOR
```

Many times, we are only interested if the red, green or blue part of a color is lighter or darker then a reference. Often that is a better indicator then the overall color. For these cases you can just add a letter:

```
IPC.o|23,34|=[rgb]COL
IPC.o|23,34|<[rgb]COL ' dunkler als COLOR
IPC.o|23,34|>[rgb]COL ' heller als COLOR

which would look like these example:

' weniger rot als COLOR
IPC.o|23,34|<r$$COL
   DBP.less red in here!
EIF.

' mehr grün als COLOR
IPC.o|23,34|>g$$COL
   DBP.More green in here!
EIF.
```

But now its gets even more compatible and interesting. To eliminate the problems i have told you above, there are more options.

```
' max. difference in % for r,g and b
' here we choose 20% max. difference
 IPC.o|23,34|~$$COL,20

' max. difference in % for r,g and b
' here we choose 15% max. difference
 IPC.o|23,34|~rgb$$COL,15

' max. difference in % for r and g
' here we choose 20% max. difference
 IPC.o|23,34|~rg$$COL,20

' max. difference in % for r only
' here we choose 20% max. difference
 IPC.o|23,34|~r$$COL,20

' max. difference in % for g only
' here we choose 20% max. difference
 IPC.o|23,34|~g$$COL,20

' max. difference in % for g and b
' here we choose 7% max. difference
 IPC.o|23,34|~gb$$COL,7

' max. difference in % for r and b
' here we choose 10% max. difference
 IPC.o|23,34|~rb$$COL,10
```


© 2023 Theo Gottwald * https://www.smart-package.com

Now we come to a very unique feature of IPC. It knows the system-colors of the actual system, its running on! You can find a table with the numbers of the system colors below "Parameters". As you can see, Nr.15 is:

```
15 = Face color for three-dimensional display elements and
     for dialog box backgrounds.
```

Now we can use these system colors directly with IPC, and NPC. using the "**S**" prefix. Here is an example, that checks if the button is there. It checks if the point has the system color it should have at this place. This is a large step ahead to make IPC. more usable even in Software distribution. Because using the "**S**" prefix, it can adapt to individual system colors!

```
STW.t|testap
SCW.ct|button|1
' Prüfen ob Farbwert mehr als 10% abweicht
IPC.o|116,38|s15
  MBX.$$CNT-$$000
EIF.
END.
```

For usage in browser-windows, take a look at the possibility of

```
' choose if the pixel is white or not
IPC.o|x,y,|!w
```

The color of the specified pixel is always placed on the actual stack. The **~** Operator can be used to check if a color is near - but not exact a given value.

This helps to make automations more compatible using different graphics-cards.

Negative form of this command is NPC.

Syntax

IPC.[P1]|P2|P3|P4 ELS. … EIF.

Parameter Explanation

P1 - (optional) can be:

```
r - do relative Movement from actual point
c - use child-coordinates
o - use coordinates from actual window whatever
t - use actual top-window coordinates
s - screen coordinates (default if omitted)
```

P2 **-** Variable[Operator]Variable

© 2023 Theo Gottwald * https://www.smart-package.com

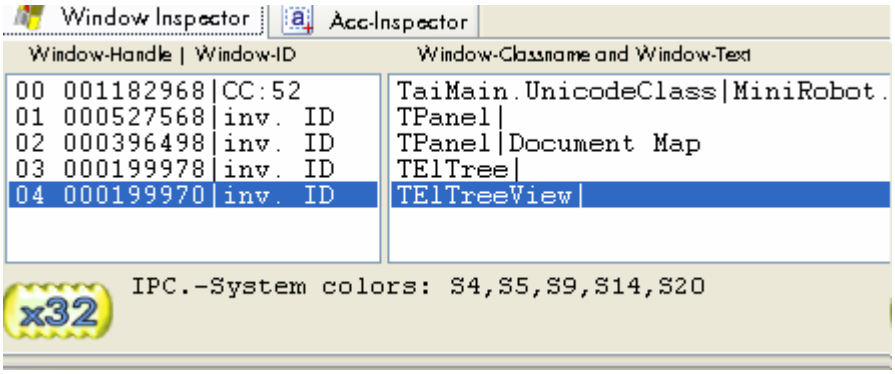
P3 **-** Options:

```
If Parameter "sxx" is used, there are 30 possible system-
colors which can be used to compare with the found pixel-
color. Value B for Black or W for white is also possible.

Please use the numeric value instead of the "xx" in "sxx".

 0 = Scroll bar gray area.
 1 = Desktop-color
 2 = Active window title bar. Also specifies the left side
     color in the color gradient of an active window's title
     bar if the gradient effect is enabled.
 3 = Inactive window caption. Also specifies the left side
color
     in the color gradient of an inactive window's title bar
if
     the gradient effect is enabled.
 4 = Menu/Messagebox  background.
 5 = Window and Button background.
 6 = Window frame.
 7 = Text in menus and Messageboxes.
 8 = Text in windows.
 9 = Text in caption, size box, and scroll bar arrow box.
 10 = Active window border.
 11 = Inactive window border.
 12 = Background color of multiple document
      interface (MDI) applications.
 13 = Item(s) selected in a control.
 14 = Text of item(s) selected in a control.
 15 = Face color for three-dimensional display elements and
      for dialog box backgrounds.
 16 = Shadow color for three-dimensional display elements
      (for edges facing away from the light source).
 17 = Grayed (disabled) text. This color is set to 0
      if the current display driver does not support
      a solid gray color.
 18 = Text on push buttons.
 19 = Color of text in an inactive caption.
 20 = Highlight color for three-dimensional display elements
     (for edges facing the light source.)
 21 = Dark shadow for three-dimensional display elements.
 22 = Light color for three-dimensional display elements
     (for edges facing the light source.)
 23 = Text color for tooltip controls.
 24 = Background color for tooltip controls.
 26 = Color for a hyperlink or hot-tracked item.
 27 = Right side color in the color gradient of an active
      window's title bar. Nr.2  specifies the left side color.
 28 = Right side color in the color gradient of an
      inactive window's title bar. Nr. 3 specifies the
      left side color.
 29 = The color used to highlight menu items when the menu
      appears as a flat menu. The highlighted menu item is
      outlined with Nr.13.
 30 = The background color for the menu bar when menus appear
```


© 2023 Theo Gottwald * https://www.smart-package.com

as flat menus. Anyway Nr.4 continues to specify the background color of menu popups.

You can see the system color of the pixel, below the mouse cursor in the windows-analyzer.



P4 - Value*

IPC's can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

Formats:

```
' IPC.[o]|x,y|Formula
' IPC.o|23,34|=COL ' EUQAL
' IPC.o|23,34|!COL ' NOT EQUAL
' IPC.o|23,34|=w ' ' Is white (R,G and B each > 252)
' IPC.o|23,34|=b ' ' Is black (R,G and B each < 5)
' IPC.o|23,34|!w ' ' Is not white (R,G and B each < 253)
' IPC.o|23,34|!b ' ' Is not black (R,G and B each > 4)
' IPC.o|23,34|<COL ' darker then COLOR
' IPC.o|23,34|>COL ' lighter then COLOR

' This version compares color with actual system colors
' IPC.o|23,34|sxx ' xx stands for a number from 0 to 30

' You can check the r or g or b part only
' IPC.o|23,34|=[rgb]COL
' IPC.o|23,34|<[rgb]COL ' r/g/b darker then COLOR
' IPC.o|23,34|>[rgb]COL ' R/g/b lighter then COLOR
' You can check for a maximum difference in %
' where 100% is 255
'
' IPC.o|23,34|~COL,Zahl 'max. difference in % for r,g and b
' IPC.o|23,34|~rgbCOL,Zahl 'max. difference in % for r,g and b
' IPC.o|23,34|~rgCOL,Zahl 'max. difference in % for r and g
' IPC.o|23,34|~rCOL,Zahl 'max. difference in % for r only
' IPC.o|23,34|~gCOL,Zahl 'max. difference in % for g only
' IPC.o|23,34|~gbCOL,Zahl 'max. difference in % for g and b
' IPC.o|23,34|~rbCOL,Zahl 'max. difference in % for r and b
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*********************************************
' IPC-Sample
'*********************************************
' DBM.2
PRT.START
JNF.Move-Console.txt

STW.ct|ieframe|internet
VAR.§§COR=274,334
MMV.§§COR
'-------------------------
GPC.
PRT.Colour is: $$000
'-------------------------
IPC.#ixp#,#iyp#|=&H08000
 PRT.IS-1
ELS.
 PRT. IS NOT
EIF.
'-------------------------
IPC.274,334|=&H08000
 PRT.IS-2
ELS.
 PRT. IS NOT
EIF.
'-------------------------
IPC.§§COR|=&H08000
 PRT.IS-3
ELS.
 PRT. IS NOT
EIF.
'-------------------------
' Darker then
IPC.§§COR|<&H09000
 PRT.IS-4
ELS.
 PRT. IS NOT
EIF.
'-------------------------
' lighter then
IPC.§§COR|>&H07000
 PRT.IS-5
ELS.
 PRT. IS NOT
EIF.
'-------------------------
' Red-part lighter then
IPC.§§COR|=r&H07000
 PRT.IS-6
ELS.
 PRT. IS NOT
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------
' Green-Part lighter then
IPC.$$COR|>g&H07000
 PRT.IS-7
ELS.
 PRT. IS NOT
EIF.
'-------------------------
' Compare a button-color with the system-colors
' This is an example for the "sxx"
STW.t|sample ap
SCW.ct|button|1
FOR.$$CNT|0|30
IPC.o|10,10|s$$CNT
 PRT.$$CNT
EIF.
NEX.
'-------------------------
' Compare a button-color with the system-colors for background
' This is an example for the "sxx"
STW.t|sample ap
SCW.ct|button|1
IPC.o|10,10|s5
 PRT.Button hat bei 10,10 Hintergrundfarbe
EIF.
NEX.
'-------------------------
' Example for the ~ (Tilde)Operator
' Check if Pixelcolor is near to the given Reference
STW.t|sample ap
SCW.ct|button|1
' 15790320 = F0F0F0
VAR.$$CNT=&HF0F0AA
' We allow 10% difference
IPC.o|10,10|~$$CNT,10
 MBX.Reference color: $$CNT $crlf$ Found color: $$000
EIF.
END.
'-------------------------
END.
```

Remarks

Limitations:

Color-Values Nr. 29 and 30 are not supported below Windows XP.-

See also:

- GPC. - Get-Pixel-Colour 1801
- 1.6.1. Program Flow Control 121

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.4 IRA. / NRA. - If Rect altered

IRA. / NRA. - If Rectangular area Altered Previous 1816 Top 41 Next 1828

MiniRobotLanguage (MRL)

IRA. / NRA. Conditional Statement

If RECT Altered

Intention

Conditional Statement.

Test if an rectangular Area (RECT) that was previously defined using `RRA.` has changed.

General Usage is as simple as:

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
' Remember the Area
RRA.
' Now check if it has changed
IRA.
  DBP.Areaa ha changed!
ELS.
  DBP.Area has not changed!
EIF.
ENR.
```

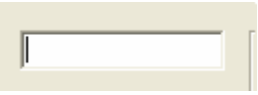
The two possible parameters are Color-Tolerance and Hit-Counter.

Color Tolerance:

If there are different graphic cards used, the pixels may not become rendered always in the exactly same way. Therefore the color tolerance gives you the chance to tell the robot "How sensitive" he should be when deciding **if a pixel is different or not.**
To make this decision, the robot will take the color of both pixels and calculate their "Color distance", that is "How similar the colors are". As a result he will get a number that can vary from 0 to 441673. Next Smarty will get your tolerance value and check if the calculated color distance is inside the chosen tolerance or not.

Hits-Counter:

Imagine a Textbox like this:



© 2023 Theo Gottwald * https://www.smart-package.com

You want to wait until there is a text inside, but the blinking cursor should **not** make your Script react. Now what you tell Smarty is a "Hit Count" of lets say 100 Pixels.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
RRA.
:again
PAU..1
IRA.0|100
  GTO.over
ELS.
  GTO.again
EIF.
:over
ENR.
```

This makes Smarty count the changed pixels, and only if she gets more then "Hit Count" she will switch. Otherwise the condition will not become true, even if there are changed pixels (but less).

`NRA.` is the negative version "If Not Rect altered".

Syntax

IRA.[P1][|P2]... ELS. ... EIF.

Parameter Explanation

P1 **-** ***(optional)*** numeric/variable contains the Tolerance to be used when checking if a color is different. You can specify the color tolerance as a numeric value from 0 - 441673, or you can specify a percentage from `0%` to `100%`. If you specify a percentage, just add the **%** Sign after the number. The color tolerance is the maximum allowed Color-Distance between two colors. The maximum color distance is **441673** that is the **distance between black and white**. If you omit this parameter, the system will use a zero Tolerance. In this case the colors are just "equal or not equal".

P2 **-** ***(optional)*** numeric/variable contains the minimum number of "Hits" that are pixels that have to be different to make the decision to become "`true`".

`IRA.`'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX.` (Enumerations) to unlimited Depth.

Example

```
'**********************************
' IRA.-Sample
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
STW.t|Sample
RRA.
' In this 2 seconds time you can change something in the
window
' or just move the window around.
PAU.2
IRA.
 MBX.Rect was altered!
ELS.
 MBX.Rect was NOT altered!
EIF.
END.


'**********************************
' IRA.-Sample 2
'**********************************
' Use "Sample Application"
' Click into the left Edit-Field to end the Loop
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
RRA.
:again
PAU..1
IRA.
  GTO.over
ELS.
  GTO.again
EIF.
:over
ENR.


'**********************************
' IRA.-Sample 2
'**********************************
' Use "Sample Application"
' Now we use a Hit-Count of 100
' The blinking Cursor will NOT switch
' But if you type something
'
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
RRA.
:again
PAU..1
IRA.0|100
  GTO.over
ELS.
  GTO.again
EIF.
:over
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's Pixel-Operations 1088
- RRA. - Remember Rectangular Area 1836
- WRA. - Wait Rect-Area Altered 1851
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.5 IRC. / NRC, - If Rect Color

MiniRobotLanguage (MRL)

IRC. / NRC. Conditional Statement

If Rectangular Area Color

Intention

Conditional Statement.

Test if an defined Color is been found inside a defined rectangular area. The Scanning is done from left to right, and top to bottom.
The first point that fits the criteria is been returned. Then you can move the mouse to that point - for example.

Or, test colors inside a defined rectangular area, if they do not match a defined color. This is done using the "**!**" Operator.

`IRC.` can also be used to locate an colored object in a window. For example if you have a graphics with red dots somewhere in a window, you can use `IRC.` to locate such a red dot. This can be done, because `IRC.` leaves you the Screen-Coordinates of the first found pixel on the Stack.

Important: Color-Operations are often troublesome because colors may not match exactly. Therefore this command provides you with a "Color Tolerance" Option.
Also you have the choice of using a "negative version" which will just check that colors do NOT match a given color/tolerance. And you can combine all this, with the resolution independent SPR 3.6 Using Quadrant-Coordinates 356 to make your Scripts run as compatible as possible.

The **four parameters** are line-coordinates, Color, Color-Tolerance and Hit-Counter.

Parameter 1 - Line-Coordinates:
The Line is defined by a Start- and and Endpoint. Using the following Format:

```
IRC.x1,y1,x2,y2
```

Instead of using Screen-Coordinates, you should prefer to use Coordinates of the last located window/object.
This is done by prefixing the coordinates with a letter. You can use:

"**w**" - Any window (Top- or Child-Window) that was last located
"**t**" - last located Top-Window
"**c**" - use Client-Coordinates of tzhe specified window. If no window is specified "w" is used as default.

© 2023 Theo Gottwald * https://www.smart-package.com

"**o**" - use the last defined Object-Coordinates (window, Accessible Object or graphical Object).
You can combine the letters as far as it makes sense:
"**wc**" - use last windows Client-Coordinates
"**tc**" - use last Top-windows Client-Coordinates

at this place you can add 2 Special Characters:
"**@**" - Enable Debug Option. Using this special Character will open a Window that shows the part of the Screen that is used for the Command. Be careful, when using `WRC.` with this Option, as it will of course run in a Loop.
"#" - this Option will reduce the Border of the Target Area by 2 Pixels in all directions.
"**##**" - using multiple "#" will reduce the Border of the Target Area by 2 Pixels times the number of "#" in all directions. If you use "##" it will decrease the target size by 4 Pixel.
"**#8**" - using "#" followed by a numeric value, the border of the Target Rectangle will be decreased by the given Amount of Pixels. In this case the border would be decreased by 8 Pixels, from any direction.

You can omit the Quadrant coordinates and just write (using any of the letters from above):

```
ILC.w
```

Then internally the command will complete the target with

```
ILC.w{A1,D4}
```

As always, if you omit `y2` then `y2` is assumed to be equal to `y1`.

```
IRC.x1,y1,x2[,y2]
```

If you omit the coordinates, the last located window or other object-rectangle is been taken, and the line goes from the upper-left to the lower-right corner.

General Usage is as simple as:

```
STW.ct|MyClassName|Sample Application
'SCW.nct|1|Edit|{&NOTEXT:}
:again
PAU..1
IRC.|&H00|0|41
  DMP.6
  MBX.!
ELS.
  GTO.again
EIF.
:over
ENR.
```

Using local (window-) coordinates:

If you prefix the coordinates with a "w" then they will be interpreted as local window coordinates. Like in this example:

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MMV.
:again
PAU..1
IRC.w1,1,10,10|!&HFFFFFF|0|32
  GTO.over
ELS.
  DMP.6
  GTO.again
EIF.
:over
DMP.6
MBX.Ready
ENR.


' Scan the complete window to find this color
STW.ct|Chrome_WidgetWin| - Google Chrome
IRC.|&H723132
  $$XPO=$$000
  $$YPO=$$000
  MMV.$$XPO,$$YPO
ELS.
  MBX.No
EIF.


' Scan only a small box inside a window to find this color
STW.ct|Chrome_WidgetWin| - Google Chrome
IRC.w459,490,461,493|&HB3B3B3
  $$XPO=$$000
  $$YPO=$$000
  MMV.$$XPO,$$YPO
ELS.
  MBX.No
EIF.
```

Parameter 2 - Color:

If this color is found along the defined Line then the decision will switch to "true".
Colors can most easily be described as a HEX-Number, like this example:

```
&HFFEE22
&HRRGGBB
```

If you prefix the color-parameter with a "**!**" (NOT Operator) then the command will count all Pixels that do NOT have that defined color and tolerance.
Please note that this is different from the `NLC.`-command, which gives you additional options.

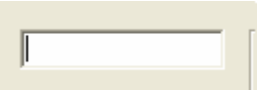
Parameter 3 - Color Tolerance:

*© 2023 Theo Gottwald * https://www.smart-package.com*

If there are different graphic cards used, the pixels may not become rendered always in the exactly same way. Therefore the color tolerance gives you the chance to tell the robot "How sensitive" he should be when deciding **if a pixel is different or not**.
To make this decision, the robot will take the color of both pixels and calculate their "Color distance", that is "How similar the colors are". As a result he will get a number that can vary from 0 to 441673. Next Smarty will get your tolerance value and check if the calculated color distance is inside the chosen tolerance or not.

Parameter 4 - Hits-Counter:

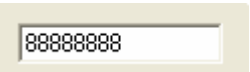
Imagine a Textbox like this:



You want to wait until there is a text inside, but the blinking cursor should **not** make your Script react. Now what you tell Smarty is a "Hit Count" of lets say 12 Pixels.

This Script will switch, after you really typed "88888888" into the first, left textbox.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
:again
PAU..1
IRC.|0|0|12
  GTO.over
ELS.
  GTO.again
EIF.
:over
ENR.
```



This makes Smarty count the changed pixels, and only if she gets more then "Hit Count" she will switch. Otherwise the condition will not become true, even if there are changed pixels (but less).

The "negative Mode" using the "!" NOT Operator

Instead of counting the Pixels that do have a defined color, you can also do the opposite.
You can count the Pixel that do NOT have that defined color.
This is done, by prefixing the "Color-Parameter" with a **!** .
Like in this example.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
:again
PAU..1
IRC.|!&HFFFFFF|0|12
  GTO.over
```

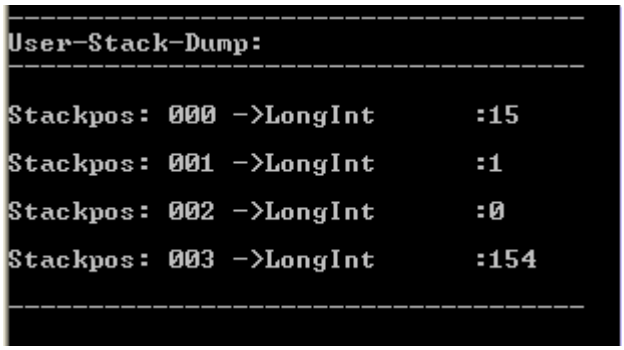

© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  DMP.6
  GTO.again
EIF.
:over
ENR.
```

The Stack when using IRC.

In case of success, `IRC.` will leave **four values on the stack**. As you can see below, these are:

```
TOS: X-Position of first found Hit
001: Y-Position of first found Hit
002: Color of first found Hit
003: Number of found Pixel meeting the condition
```



And there is NRC.

`NRC.` is the negative version "If Not Rect Color".

Syntax

IRC.[w][P1][|P2][|P3][|P4]…
ELS. … EIF.

Parameter Explanation

P1 **-** ***(optional)*** Upper-Left and lower right Coordinate Point of the rectangle. There are different possible Formats for these Coordinates. With out the leading "`w`", these are Screen-Coordinates.
Its best practice, to prefer Child-Window-relative coordinates:
`IRC.w18,56,153,57|$$COL|$$TOL`

You can simply specify the coordinates In this format:
`IRC.x1,y1,x2,y2`
a real example would be:

© 2023 Theo Gottwald * https://www.smart-package.com

```
IRC.10,15,100,200
```

Or you can specify a Rectangle, using Numbers like:

```
IRC.10,15,100
```

then `y2` is assumed to be the same as `y1`.

This will tell the command that we want to use the Quadrant in the last located window (child- or Topwindow).

You can use:

"**w**" - Any window (Top- or Child-Window) that was last located

"**t**" - last located Top-Window

"**c**" - use Client-Coordinates of tzhe specified window. If no window is specified "w" is used as default.

"**o**" - use the last defined Object-Coordinates (window, Accessible Object or graphical Object).

You can combine the letters as far as it makes sense:

"**wc**" - use last windows Client-Coordinates

"**tc**" - use last Top-windows Client-Coordinates

at this place you can add 2 Special Characters:

"**@**" - Enable Debug Option. Using this special Character will open a Window that shows the part of the Screen that is used for the Command. Be careful, when using `WRC.` with this Option, as it will of course run in a Loop.

"**#**" - this Option will reduce the Border of the Target Area by 2 Pixels in all directions.

"**##**" - using multiple "#" will reduce the Border of the Target Area by 2 Pixels times the number of "#" in all directions. If you use "##" it will decrease the target size by 4 Pixel.

"**#8**" - using "#" followed by a numeric value, the border of the Target Rectangle will be decreased by the given Amount of Pixels. In this case the border would be decreased by 8 Pixels, from any direction.

You can omit the Quadrant coordinates and just write (using any of the letters from above):

```
ILC.w
```

Then internally the command will complete the target with

```
ILC.w{A1,D4}
```

If omitted the Rectangle from the last localized Object or window is taken, same as "`W`".

P2 - (optional) numeric/variable contains the color value to scan for.

P3 - (optional) numeric/variable contains the Tolerance to be used when checking if a color is different. You can specify the color tolerance as a numeric value from 0 - `441673`, or you can specify a percentage from `0%` to `100%`. If you specify a percentage, just add the **%** Sign after the number. The color tolerance is the maximum allowed Color-Distance between two colors. The maximum color distance is **441673** that is the **distance between black and white**. If you omit this parameter, the system will use a zero Tolerance. In this case the colors are just "equal or not equal".

© 2023 Theo Gottwald * https://www.smart-package.com

P4 - (optional) numeric/variable contains the minimum number of "Hits" that are pixels that have to be different to make the decision to become "`true`".

`IRC.`'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX.` (Enumerations) to unlimited Depth.

Example

```
'**********************************
' IRC.-Sample
'**********************************
' Use "Sample Application"
' Click into the left Edit-Field
' and enter a large number
' to end the Loop
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
:again
PAU..1
IRC.|!&HFFFFFF|0|120
  GTO.over
ELS.
  DMP.6
  GTO.again
EIF.
:over
DMP.6
MBX.Ready
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's Pixel-Operations 1088
- ILC. / NLC. - If Line Color 1805
- WRC. - Wait Rect Color 1856
- WLC. - Wait for Line Color 1846
- RRA. - Remember Rectangular Area 1836

© 2023 Theo Gottwald * https://www.smart-package.com

- IRA. / NRA. - If Rect altered 1824
- WRA. - Wait Rect-Area Altered 1851
- 1.8.1 The "Client" and the "Non-Client" Parts of a window 156
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.6 RRA. - Remember Rectangular Area

MiniRobotLanguage (MRL)

RRA. Command

Remember / store / administrate Rectangular Areas

Intention

`RRA.` remembers an defined Area. Later you can check if the Area has changed in any way using `IRA.` or `WRA.`

`RRA.` can remember several Areas and check if any of these is currently available, use the advanced `RRA.` -Options together with `IRA. or WRA.` for this.

`RRA.` can save and load Rectangular Regions into files. This way your Script can check for predefined graphical Patterns. As `IRA.` allows to specify a Color-Tolerance, there is no need to have a "Perfect Match" on the target side. You will just need to use the right values.

`RRA.` can store 9 Rectangular Pattern into 9 boxes. For this you can specify the number of the box, that you wish to use. In case you do not specify a number, **Box Nr.1 is the Standard Box that is been used in all operations of** `IRA.` **or** `WRA.` **etc..**

`RRA.` has features to manipulate those boxes and their content. Using these features, your target should be to move the Rect-Pattern that you need, into Box Nr.1 because this is where the `WRA.` and the `IRA.` can use it.

What is it good for?

Take for example a Form (Class "`Sunawt`" for example), where the robot is unable to detect if a button has appeared or not.
The Installation-Program is running an undefined time, when it ends, that button appears. You can see it, but the robot may under seldom circumstances be unable to see that button.

Now comes `RRA.` and `WRA.`
These commands are just for these cases where all other commands do fail.

`RRA.` will store a copy of the window or a part of it.
`WRA.` will hold your Script as long as "it did not change". Until something happens.

Built-in Safety Belt
Sometimes windows redraws content that is inside a window, that you might be watching for changes using `WRA.` To avoid such a Redraw/Refresh to fool your Script that "something has changed", the `WRA.` command has a built-in safety mechanism.

*© 2023 Theo Gottwald * https://www.smart-package.com*

It will check 3 times within 75 ms, and all three checks need to bring results that do differ from the original Pattern.

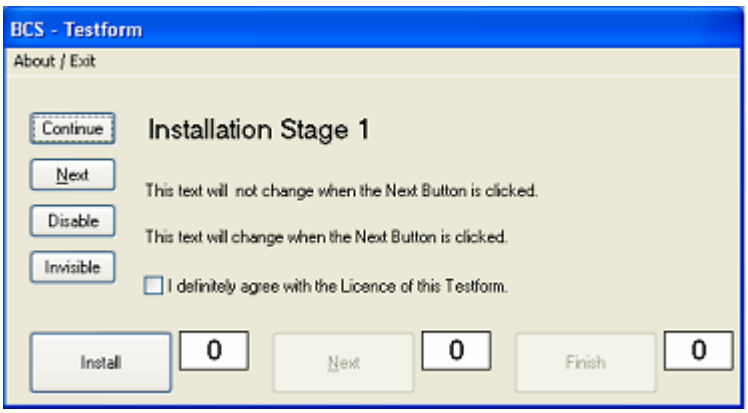
Hint: When using `RRA.`

Here are the modes of operation for `RRA.`:

```
' Using RRA. without any Parameter
RRA.
```

In this case `RRA.` will Remember the "Client-Area" of the actual Top/Childwindow.

Take this window for example:



In the picture below i have marked the "Client-Area" in yellow color, and the "Non-Client-Area" in Orange Color.

For more infos regarding the "Client" and the "Non-Client Area" see here:
1.8.1 The "Client" and the "Non-Client" Parts of a window 156

`RRA.` can be used with graphical or accessible Objects also.

You can also use `RRA.` with graphical or Accessible Objects, that are no windows. Like in this example.

Therefore it can also be used in Browser and Office-Programs.

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|PBWindowClass:0|BCS - Testform
AGR.|&EXACT:Install
MMV.
RRA.
RRA.getpos|$$POS
DBP.$$POS
ENR.
```

Always specify a window or Rectangle directly before the RRA. command

```
' Right.
STW.ct|Chrome_WidgetWin| - Google Chrome
RRA.w20,552,22,552
WRA.

' This will locate an Actual Point somewhere
WLC.w270,821,971,821|&H0
RRA.clear

' Wrong.
' Thats why this has no more window
' Therefore it would use the Desktop-Window at these
coordinates
' Thats why you should always use STW. / SCW. directly before
the RRA.
RRA.w20,552,22,552
WRA.


' To watch an Edit-Control for getting the input Focus you
could use:
%T_RunTestApp 246
%Loc_Edit
RRA.Client
WRA.

' This would include the Edges of the window in the case of an
edit-box
%T_RunTestApp 246
%Loc_Edit
RRA.Window
WRA.


' This will show the coordinates of the window that is watched
(Screen coords)
%T_RunTestApp 246
%Loc_Edit
RRA.Client
RRA.XSize|$$XSA
RRA.YSize|$$YSA
RRA.XPos|$$XPO
RRA.YPos|$$YPO
DBP.$$XSA,$$YSA-$$XPO,$$YPO
WRA.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

RRA.[P1][|P2][|P3]

Parameter Explanation

P1 – (optional) can be an prototype or screen coordinates of a rectangle.

If you use `RRA.` without any parameter, the actual window is been used. It can be a Top- or a child-window. If a Top-Window is used, `RRA.` tries to only remember the Client-Area of the window. This excludes the window borders.

Using a **Rectangle in screen coordinates.** Example(s):

```
' If you just write coordinates, these will be interpreted
' as screen coordinates:
 RRA.100,150,200,200
' If you specify 3 coordinates only, y2 will be assumed to
' be eqal to x2
 RRA.100,150,200
```

You can specify in which "box" you want the area to be stored. Just use **P2** for this:

```
RRA.(rect)|$$NUM
```

You can specify a "w" infront of the coordinates to have window coordinates used.

```
RRA.w10,20,30,40|$$NUM
```

Prototypes:

The following Prototypes are currently implemented:

Clear - clear all Boxes from 1 - 9.

```
RRA.Clear
' Will just clear the specified Box:
RRA.Clear|$$NUM
```

Client - Use last localized Top or Childwindow-Borders

```
RRA.Client
```

Window - use Borders from last localized Element to fill `Box(1)`

```
RRA.Window
```

ToVar - this will store Box(**P2**) into a variable.
P2 is the number of the box to use. **P3** is the Variable-Name where things will be stored.

```
RRA.toVar|$$NUM|$$VAR
```

FromVar - this will restore the Data from a variable into the Box(**P2**), It is for use, together with "`ToVar`". **P2** is the number of the target box to use.

© 2023 Theo Gottwald * https://www.smart-package.com

P3 is the Variable with the Area-Data, that was created using "`ToVar`"..

```
RRA.toVar|$$NUM|$$VAR
RRA.fromVar|$$NUM|$$VAR
```

Swap - this will exchange the contents of Box(1) with the content of Box(`$$NUM`).

```
RRA.Swap|$$NUM
```

if you specify a second parameter, the content of these two boxes will be exchanged.

```
RRA.swap|$$NUM|$$NUA
```

Get - copies Box Nr. `$$NUM` into `Box(1)`. `Box(1)` is is been overwritten. `Box($$NUM)` is still available.

```
RRA.get|$$NUM
```

Store - Backup-Copy `Box(1)->Box($$NUM)`

```
RRA.store|$$NUM
```

RemPos - Use Rempos, the locate another Element. The stored Rect will be applied to the coordinates of this new element.
This will remove the positional information from the rectangle in box Nr. `$$NUM`. In case the window was moved, the the position information is not more accurate, because its stored in screen coordinates. This way you can mark the positional Informations as "invalid". The rectangular data can still be used, if a new position is given. If you omit `$$NUM` then box Nr.1 is used.

```
RRA.rempos|$$NUM
```

NewPos - Set a new position to an existing Box: Use Screen-coordinates, You can only change the position but never the size of a stored Area.

```
RRA.NewPos|$$NAM|$$XPO|$$YPO
```

GetPos - get the Position-Data (in screen coordinates) of the stored Area in Box(1) into Variable `$$VAR`. Format is X-Pos,Y-Pos.

```
RRA.GetPos|$$RES[|$$NUM]
```

XPos – get the X-Position of the (upper left corner of the) Area in Box Nr.1 into Variable `$$VAR`.

```
RRA.XPos|$$RES
```

YPos – get the Y-Position of the (upper left corner of the) Area in Box Nr.1 into Variable `$$VAR`.

```
RRA.YPos|$$RES
```

Size – get the size of the stored Area in Box(1) into Variable `$$VAR`. Format is X-Size,Y-Size.

```
RRA.size|$$RES
```

XSize – get the X-Size of the stored Area in Box(1) into Variable `$$VAR`.

```
RRA.xsize|$$RES
```

YSize – get the Y-Size of the stored Area in Box(1) into Variable `$$VAR`.

```
RRA.ysize|$$RES
```

P2, ***P3***, ***P4*** – (optional) - content of these parameters depend on the prototype in **P1**.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' RRA./WRA.-Demo
'**********************************
' You need to start the BCS:-Testform (Sample Apps)
' for this Sample.
' Move the mouse over the checkmark to have the
' Mouseover effect. This will make WRA. to switch.
'
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|I definitely agree with the Licence
' Watch the Checkmark
RRA.
' Wait until something changes in that rectangular area
WRA.
ENR.


'**********************************
' RRA.-Demo
'**********************************
' You need to start the BCS:-Testform (Sample Apps)
' for this Sample.
'
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|I definitely agree with
RRA.
RRA.getpos|$$POS
DBP.$$POS
ENR.
```

Remarks

To make sure that "Redrawing Effects" can not fool `WRA.`, `WRA.` has a built-in 75 ms security mechanism. That means, that there will be 3 checks done within 75 ms.
In these checks, all 25 ms the Area is been compared to the Original Area, and each time the Area must be different from the original Pattern that was captured using `RRA.`
This is done, to prevent a normal Refresh of a window to be taken as a "Ready Signal". Only if `WRA.` is sure, `WRA.` will switch and let the Script continue running.

While `RRA.` can actually hold up to 9 Rectangular Areas, there is also a "hidden" tenth box, **Box. Nr.0**, that you should not use in regular cases. This is the box where the last snaphot from the last call to `IRA.` / `WRA.` is internally stored.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 3.6 Using Quadrant-Coordinates 356
- 1.8.1 The "Client" and the "Non-Client" Parts of a window 156
- CST. - Coordinate System Transform 2331
- GAR. - Get actual Rectangle 3619
- MAP. - Move Active Point 2483
- GRM. - Get Rect Middle-Point 2337
- QFR. - Quadrant from RECT 1130
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.7 SPC. - Set-Pixel-Colour

MiniRobotLanguage (MRL)

SPC. Command

Set Pixel Colour

Intention

SPC will set the color of a Screen-Pixel non-permanent. You can specify coordinates in Screen-, or as window-coordinates.

For drawing on any background, you can just omit the color value then the actual color of that point will be inverted.

Please prefer using the Drawing Commands from the ANA.-Command.
This Command is mostly for use with WLC. or IPC. etc.

Syntax

SPC.[P1][|P2,P3][|P4]

Parameter Explanation

P1 - (optional) prototype coordinates

```
Prototype can be omitted or any of:
 r - do relative Movement from actual point
 c - use child-coordinates
 o - use coordinates from actual window whatever
 t - use actual top-window coordinates
 l - when using "l", x and y values are ignored, and
internally
     calculated as the left side of the rectangle, where a
     checkmark could be. "l" - is for checkmarks.
 s - (default) use screen coordinates

If prototype is omitted, Screen-Coordinates are taken.
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

P4 - the color to set the pixel to, if omitted actual pixel-color is inverted.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Instead of using separate variables for X and Y, you also use a Point-Variable, where x and y values are separated by a ",".
VAR.§§PAP=10,10
SPC.§§PRE|§§PAP|§§COL

Example

```
'**********************************
' SPC.-Demo
' Run as EXE (Compiled)
' or it may be really slow!
'**********************************
JSR.closeNotepads
EXO.notepad
STW.tc|editor|notepad
SCW.c|edit

' Paint on Screen
: §§RNA=o
: §§PRE=s
JSR.paint
MBX. Point on TopWindow $crlf$ (you may not see something)
' Paint on TOP-Window-Area
: §§RNA=1
: §§PRE=t
JSR.paint
PAU.2
MBX.Paint into Control
' Paint on Child-Window-Area
: §§RNA=o
: §§PRE=c
JSR.paint
MBX.!
DMP.
PRT. Pixel-Color is: $$000
PRT. Red is: $$000
PRT. Green is: $$000
PRT. Blue is: $$000
STS.DUMP
MBX.!
END.

:paint
FOR.§§LOY|1|50
  FOR.§§LOX|1|50
    IVV.§§RNA=0
      RND.0|22000000|§§RND
    ELS.
      : §§RND=2142345
    EIF.
    SPC.§§PRE|§§LOX,§§LOY|§§RND
  ' change this line to the following and see the difference
  '  SPC.§§PRE|§§LOX,§§LOY
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  NEX.
NEX.
RET.

:closeNotepads
STW.ctwo|notepad|Editor|0|overcn
CLW.
JMP.closeNotepads
:overcn
RET.
```

Remarks

-

Limitations:

All your drawings will disappear, when the window is going to be refreshed.

See also:

- 1.8. Selecting the Application (Topwindow) 150
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.8 WLC. - Wait for Line Color

MiniRobotLanguage (MRL)

WLC. Wait Statement

Wait for Line Color

Intention

Wait for Condition Statement.

Wait until an defined Color is been found along a defined Line.

Or, Wait until colors along a given line do not match a defined color.
This is done using the "**!**" Operator.

Important: Color-Operations are often troublesome because colors may not match exactly. Therefore this command provides you with a "Color Tolerance" Option.
Also you have the choice of using a "negative version" which will just check that colors do NOT match a given color/tolerance. And you can combine all this, with the resolution independent SPR 3.6 Using Quadrant-Coordinates 356 to make your Scripts run as compatible as possible.

Parameter 1 - Line-Coordinates:
The Line is defined by a Start- and and Endpoint. Using the following Format:

```
WLC.x1,y1,x2,y2
```

Its best practice to always use window local coordinates. For this specify a small "w" just in front of the coordinates.

```
WLC.wx1,y1,x2,y2
```

As always, if you omit `y2` then `y2` is assumed to be equal to `y1`.
Also, you can simplify many tasks, by just using 3.6 Using Quadrant-Coordinates 356.
In this case do not specify the "`w`" as the Quadrants are always window local.

If you omit the coordinates, the last located window or other object-rectangle is been taken, and the line goes from the upper-left to the lower-right corner.

General Usage is as simple as:

```
STW.ct|MyClassName|Sample Application
'SCW.nct|1|Edit|{&NOTEXT:}
' Wait for Color "Black" (&H00) to appear inside the textbox
WLC.|&H00|0|4|60
DMP.6
MBX.!
ENR.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

The five parameters are line-coordinates, Color, Color-Tolerance and Hit-Counter and Timeout in Seconds.

Parameter 2 - Color:

If this color is found along the defined Line then the decision will switch to "true". Colors can most easily be described as a HEX-Number, like this example:

```
&HFFEE22
&HRRGGBB
```

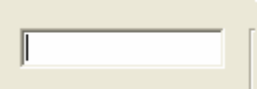
If you prefix the color-parameter with a "**!**" (NOT Operator) then the command will count all Pixels that do NOT have that defined color and tolerance.

Parameter 3 - Color Tolerance:

If there are different graphic cards used, the pixels may not become rendered always in the exactly same way. Therefore the color tolerance gives you the chance to tell the robot "How sensitive" he should be when deciding **if a pixel is different or not**.
To make this decision, the robot will take the color of both pixels and calculate their "Color distance", that is "How similar the colors are". As a result he will get a number that can vary from 0 to 441673. Next Smarty will get your tolerance value and check if the calculated color distance is inside the chosen tolerance or not.

Parameter 4 - Hits-Counter:

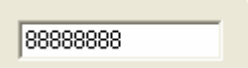
Imagine a Textbox like this:



You want to wait until there is a text inside, but the blinking cursor should **not** make your Script react. Now what you tell Smarty is a "Hit Count" of lets say 12 Pixels.

This Script will switch, after you really typed "88888888" into the first, left textbox.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WLC.|&H000|1|15|60
DMP.6
MBX.!
ENR.
```

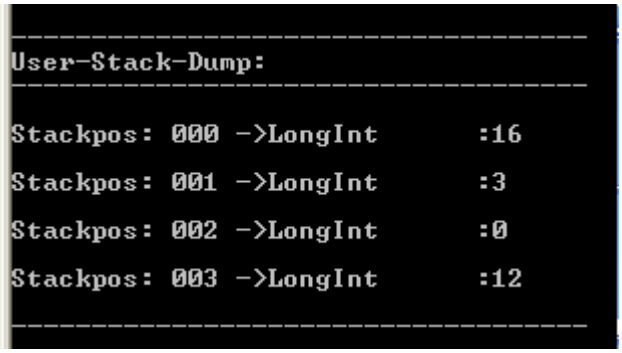


This makes Smarty count the changed pixels, and only if she gets more then "Hit Count" she will switch. Otherwise the condition will not become true, even if there are changed pixels (but less).

© 2023 Theo Gottwald * https://www.smart-package.com

In case of success, `WLC.` **will leave four values on the stack.** As you can see below, these are:

```
TOS: X-Position of first found Hit
001: Y-Position of first found Hit
002: Color of first found Hit
003: Number of found Pixel meeting the condition
```



The "negative Mode" using the "!" NOT Operator

Instead of counting the Pixels that do have a defined color, you can also do the opposite.
You can count the Pixel that do NOT have that defined color.
This is done, by prefixing the "Color-Parameter" with a **!**.
Like in this example.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WLC.|!&HFFFFFF|1|15|60
DMP.6
MBX.!
ENR.
```

Syntax

WLC.[w][P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 **-** ***(optional)*** Start and End-Points of the Line.
There are different possible Formats for these Coordinates.
With out the leading "w", these are Screen-Coordinates.
Its best practice, to prefer Child-Window-relative coordinates:
`WLC.w18,56,153,57|$$COL|$$TOL`

You can simply specify the start and end-point. In this format:

© 2023 Theo Gottwald * https://www.smart-package.com

`WLC.x1,y1,x2,y2`
a real example would be:
`WLC.10,15,100,200`
Or you can specify a Rectangle, using Numbers like:
`WLC.10,15,100`
then `y2` is assumed to be the same as `y1`.

You can also use 3.6 Using Quadrant-Coordinates 356.
In this case the Line will be drawn from the upper-left corner straight to the lower right corner.

```
WLC.{D1,D2}|$$COL|$$TOL|$$TIM
WLC.{B1}|$$COL|$$TOL
```

In the same way you can unse the large letter "`W`" or "`C`" and write:

```
ILC.W|$$COL|$$TOL|$$TIM
ILC.C|$$COL|$$TOL
```

If omitted the Rectangle from the last localized Object or window is taken, same as "`W`".

P2 - (optional) numeric/variable contains the color value to scan for.

P3 - (optional) numeric/variable contains the Tolerance to be used when checking if a color is different. You can specify the color tolerance as a numeric value from 0 - `441673`, or you can specify a percentage from `0%` to `100%`. If you specify a percentage, just add the **%** Sign after the number. The color tolerance is the maximum allowed Color-Distance between two colors. The maximum color distance is **441673** that is the **distance between black and white**. If you omit this parameter, the system will use a zero Tolerance. In this case the colors are just "equal or not equal".

P4 - (optional) numeric/variable contains the minimum number of "Hits" that are pixels that have to be different to make the decision to become "`true`".

P5 - (optional) numeric/variable contains the timeout time in seconds. After this time, the command will set the timeout-flag and just let the script go on running. You can ask for the state of the timeout-flag to find out, if a timeout happened or not.

Example

```
'**********************************
' WLC.-Sample
'**********************************
' Use "Sample Application"
'
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WLC.|&H000|1|15|60
DMP.6
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' WLC.-Sample
'**********************************
' Use "Sample Application"
'
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WLC.|!&H0FFFFFF|1|15|60
DMP.6
MBX.!
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's Pixel-Operations 1088
- 1.8.1 The "Client" and the "Non-Client" Parts of a window 156
- 3.6 Using Quadrant-Coordinates 356
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801
- ILC. / NLC. - If Line Color 1805
- RRA. - Remember Rectangular Area 1836
- IRA. / NRA. - If Rect altered 1824
- WRA. - Wait Rect-Area Altered 1851
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.9 WRA. - Wait Rect-Area Altered

WRA. - Wait until Rectangular Area Altered **Previous** 1846 **Top** 41 **Next** 1856

MiniRobotLanguage (MRL)

WRA. Wait-Command

Wait until something in a rectangular area has changed

Intention

These command is just for these cases where all other commands do fail.

`RRA.` will store a copy (a Screenshot) of the window or a part of it.
`WRA.` will pause your Script as long as the stored Area "did not change".

This way you can wait until a program GUI reacts, even if you can not read any of the GUI Output in conventional way using other commands.

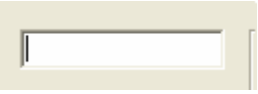
The first three possible parameters are Color-Tolerance and Hit-Counter.
The fourth Parameter is an optional Timeout in seconds.

Parameter 1 - Color Tolerance:

If there are different graphic cards used, the pixels may not become rendered always in the exactly same way. Therefore the color tolerance gives you the chance to tell the robot "How sensitive" he should be when deciding **if a pixel is different or not.**
To make this decision, the robot will take the color of both pixels and calculate their "Color distance", that is "How similar the colors are". As a result he will get a number that can vary from 0 to 441673. Next Smarty will get your tolerance value and check if the calculated color distance is inside the chosen tolerance or not.

Parameter 2 - Hits-Counter:

Imagine a Textbox like this:



You want to wait until there is a text inside, but the blinking cursor should **not** make your Script react. Now what you tell Smarty is a "Hit Count" of lets say 100 Pixels.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
RRA.
PAU..1
WRA.0|100
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

This makes Smarty count the changed pixels, and only if she gets more then "Hit Count" she will switch. Otherwise the condition will not become true, even if there are changed pixels (but less).

Parameter 3 - Wait-Timeout:

Imagine there is an Error somewhere in the target application. The reaction you are waiting for will never happen. This could make your script hang forever!
To avoid this, you have an "Timeout Option".
If you write an number here, it is a Timeout-Time in Seconds.
After this time has elapsed, the command will let the robot continue running, even if there has not been the change we have been waiting for.
In this case, the "Timeout Flag" will be set. In the normal case, the timeout flag will be cleared.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
RRA.
WRA.0|100
ITO
  DBP. Timeout, something went wrong!
ELS.
  DBP. All fine Script goes on.
EIF.
ENR.
```

Built-in Safety Belt
Sometimes windows redraws content that is inside a window, that you might be watching for changes using `WRA.` To avoid such a Redraw/Refresh to fool your Script that "something has changed", the `WRA.` command has a built-in safety mechanism. It will check 3 times within 75 ms, and all three checks need to bring results that do differ from the original Pattern.

Hint: When using `RRA.`

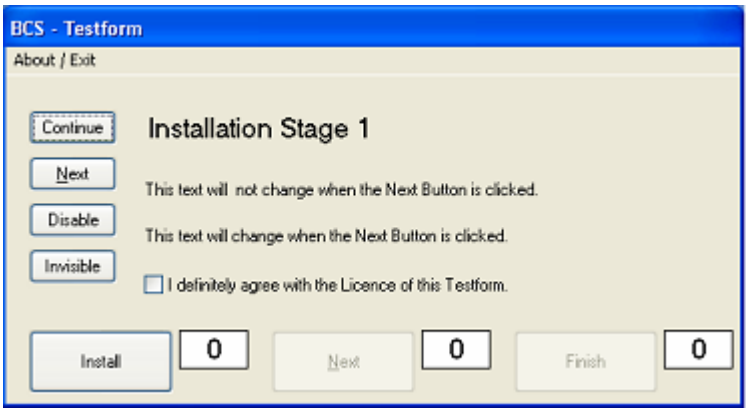
Here are the modes of operation for `RRA.`:

```
' Using RRA. without any Parameter
RRA.
```

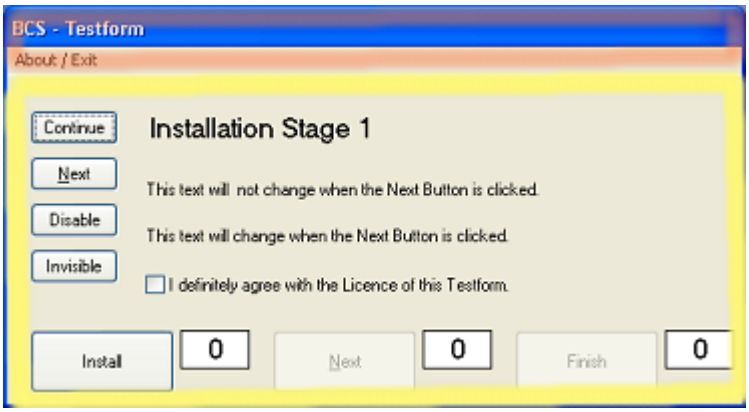
In this case `RRA.` will Remember the "Client-Area" of the actual Top/Childwindow.

Take this window for example:

© 2023 Theo Gottwald * https://www.smart-package.com



In the picture below i have marked the "Client-Area" in yellow color, and the "Non-Client-Area" in Orange Color.



For more infos regarding the "Client" and the "Non-Client Area" see here:
1.8.1 The "Client" and the "Non-Client" Parts of a window 156

WRA. can be used with graphical or accessible Objects also.

You can also use `WRA.` with graphical or Accessible Objects, that are no windows. Like in this example.

Therefore it can also be used in Browser and Office-Programs.

```
STW.ct|PBWindowClass:0|BCS - Testform
AGR.|&EXACT:Install
MMV.
RRA.
WRA.
RRA.getpos|$$POS
DBP.$$POS
ENR.
```

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

WRA.[P1][|P2][|P3]

Parameter Explanation

P1 **-** ***(optional)*** numeric/variable contains the Tolerance to be used when checking if a color is different. You can specify the color tolerance as a numeric value from 0 - 441673, or you can specify a percentage from `0%` to `100%`. If you specify a percentage, just add the **%** Sign after the number. The color tolerance is the maximum allowed Color-Distance between two colors. The maximum color distance is **441673** that is the **distance between black and white**. If you omit this parameter, the system will use a zero Tolerance. In this case the colors are equal or not equal.

P2 **-** ***(optional)*** numeric/variable contains the minimum number of "Hits" that are pixels that have to be different to make the decision to become "`true`".

P3 **-** ***(optional)*** numeric/variable contains the timeout time in seconds. After this time, the command will set the timeout-flag and just let the script go on running. You can ask for the state of the timeout-flag to find out, if a timeout happened or not.

Example

```
'**********************************
' WRA.-Demo
'**********************************
' You need to start the BCS:-Testform (Sample Apps)
' for this Sample.
' Move the mouse over the checkmark to have the
' Mouseover effect. This will make WRA. to switch.
'
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|I definitely agree with the Licence
' Watch the Checkmark
RRA.
' Wait until something changes in that rectangular area
WRA.
ITO.
  DBP.Timeout!
EIF.
ENR.


'**********************************
' WRA.-Demo
'**********************************
' You need to start the BCS:-Testform (Sample Apps)
' for this Sample.
'
STW.ct|PBWindowClass:0|BCS - Testform
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SCW.nct|1|Button|I definitely agree with
RRA.
WRA.
RRA.getpos|$$POS
DBP.$$POS
ENR.
```

Remarks

To make sure that "Redrawing Effects" can not fool `WRA.`, `WRA.` has a built-in 75 ms security mechanism. That means, that there will be 3 checks done within 75 ms.
In these checks, all 25 ms the Area is been compared to the Original Area, and each time the Area must be different from the original Pattern that was captured using `RRA.`
This is done, to prevent a normal Refresh of a window to be taken as a "Ready Signal".
Only if `WRA.` is sure, `WRA.` will switch and let the Script continue running.

While `RRA.` can actually hold up to 9 Rectangular Areas, there is also a "hidden" tenth box, **Box. Nr.0**, that you should not use in regular cases. This is the box where the last snaphot from the last call to `IRA.` / `WRA.` is internally stored.

Limitations:

-

See also:

- RRA. - Remember Rectangular Area 1836
- IRA. / NRA. - If Rect altered 1824
- 3.6 Using Quadrant-Coordinates 356
- 1.8.1 The "Client" and the "Non-Client" Parts of a window 156
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.6.10 WRC. - Wait Rect Color

MiniRobotLanguage (MRL)

WRC. Wait Statement

Wait for Color in Rectangular Areea

Intention

Wait for Condition Statement.
Wait until an defined Color is been found inside a defined rectangular area.
Or, Wait until colors inside a defined rectangular area do not match a defined color.
This is done using the "**!**" (not) Operator.

Important: Color-Operations are often troublesome because colors may not match exactly. Therefore this command provides you with a "Color Tolerance" Option.
Also you have the choice of using a "negative version" which will just check that colors do NOT match a given color/tolerance. And you can combine all this, with the resolution independent SPR 3.6 Using Quadrant-Coordinates 356 to make your Scripts run as compatible as possible.

This command is helpful to automate situations where no other command can locate a window or button.

Parameter 1 - Area-Coordinates:
The rectangular area is defined by a
- *the upper-left corner* and
- *the lower, right corner*
Using the following Format:

```
WRC.x1,y1,x2,y2
```

As always, if you omit `y2` then `y2` is assumed to be equal to `y1`.
Also, you can simplify many tasks, by just using 3.6 Using Quadrant-Coordinates 356.

If you omit the coordinates, the last located window or other object-rectangle is been taken, and the line goes from the upper-left to the lower-right corner.

General Usage is as simple as:

```
STW.ct|MyClassName|Sample Application
'SCW.nct|1|Edit|{&NOTEXT:}
:again
' Wait for Color "Black" (&H00) to appear inside the textbox
WRC.|&H00|0|4|60
DMP.6
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

The five parameters are RECT-coordinates, Color, Color-Tolerance and Hit-Counter and Timeout in Seconds.

Parameter 2 - Color:

If this color is found inside the defined rectangular area then the decision will switch to "true". Colors can most easily be described as a HEX-Number, like this example:

```
&HFFEE22
&HRRGGBB
```

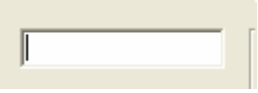
If you prefix the color-parameter with a "**!**" (NOT Operator) then the command will count all Pixels that do NOT have that defined color and tolerance.

Parameter 3 - Color Tolerance:

If there are different graphic cards used, the pixels may not become rendered always in the exactly same way. Therefore the color tolerance gives you the chance to tell the robot "How sensitive" he should be when deciding **if a pixel is different or not.**
To make this decision, the robot will take the color of both pixels and calculate their "Color distance", that is "How similar the colors are". As a result he will get a number that can vary from 0 to 441673. Next Smarty will get your tolerance value and check if the calculated color distance is inside the chosen tolerance or not.

Parameter 4 - Hits-Counter:

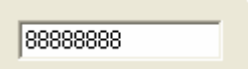
Imagine a Textbox like this:



You want to wait until there is a text inside, but the blinking cursor should **not** make your Script react. Now what you tell Smarty is a "Hit Count" of lets say 12 Pixels.

This Script will switch, after you really typed "`88888888`" into the first, left textbox.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WRC.|&H000|1|15|160
DMP.6
MBX.!
ENR.
```

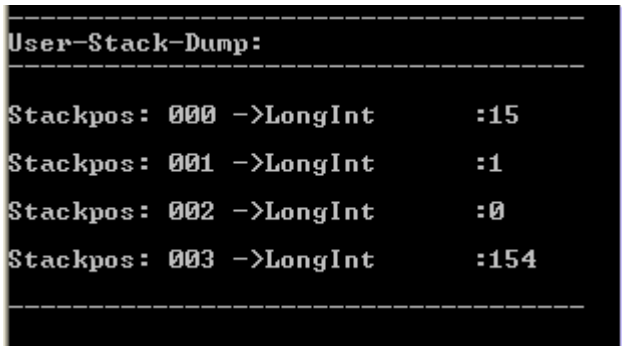


This makes Smarty count the changed pixels, and only if she gets more then "Hit Count" she will switch. Otherwise the condition will not become true, even if there are changed pixels (but less).

© 2023 Theo Gottwald * https://www.smart-package.com

In case of success, `WRC.` **will leave four values on the stack.** As you can see below, these are:

```
TOS: X-Position of first found Hit
001: Y-Position of first found Hit
002: Color of first found Hit
003: Number of found Pixel meeting the condition
```



The "negative Mode" using the "!" NOT Operator

Instead of counting the Pixels that do have a defined color, you can also do the opposite.
You can count the Pixel that do NOT have that defined color.
This is done, by prefixing the "Color-Parameter" with a **!**.
Like in this example.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WRC.|!&HFFFFFF|1|15|150
DMP.6
MBX.!
ENR.
```

Syntax

WRC.[w][P1][|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - (optional) Upper-Left and lower right Coordinate Point of the rectangle.
There are different possible Formats for these Coordinates.
With out the leading "w", these are Screen-Coordinates.
Its best practice, to prefer Child-Window-relative coordinates:
`WRC.w18,56,153,57|$$COL|$$TOL`

© 2023 Theo Gottwald * https://www.smart-package.com

You can simply specify the coordinates In this format:
`WRC.x1,y1,x2,y2`
a real example would be:
`WRC.10,15,100,200`
Or you can specify a Rectangle, using Numbers like:
`WRC.10,15,100`
then `y2` is assumed to be the same as `y1`.

You can also use 3.6 Using Quadrant-Coordinates 356.
In this case the Line will be drawn from the upper-left corner straight to the lower right corner.

```
WRC.{D1,D2}|$$COL|$$TOL|$$TIM
WRC.{B1}|$$COL|$$TOL
```

In the same way you can unse the large letter "`W`" or "`C`" and write:

```
WRC.W|$$COL|$$TOL|$$TIM
WRC.C|$$COL|$$TOL
```

If omitted the Rectangle from the last localized Object or window is taken, same as "`W`".

P2 - (optional) numeric/variable contains the color value to scan for.

P3 - (optional) numeric/variable contains the Tolerance to be used when checking if a color is different. You can specify the color tolerance as a numeric value from 0 - `441673`, or you can specify a percentage from `0%` to `100%`. If you specify a percentage, just add the **%** Sign after the number. The color tolerance is the maximum allowed Color-Distance between two colors. The maximum color distance is **441673** that is the **distance between black and white**. If you omit this parameter, the system will use a zero Tolerance. In this case the colors are just "equal or not equal".

P4 - (optional) numeric/variable contains the minimum number of "Hits" that are pixels that have to be different to make the decision to become "`true`".

P5 - (optional) numeric/variable contains the timeout time in seconds. After this time, the command will set the timeout-flag and just let the script go on running. You can ask for the state of the timeout-flag to find out, if a timeout happened or not.

Example

```
'**********************************
' WRC.-Sample
'**********************************
' Use "Sample Application"
'
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WRC.|&H000|1|15|140
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.6
MBX.!
ENR.

'**********************************
' WRC.-Sample using the ! Operator
'**********************************
' Use "Sample Application"
'
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
WRC.|!&HFFFFFF|1|15|140
DMP.6
MBX.!
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's Pixel-Operations 1088
- 1.8.1 The "Client" and the "Non-Client" Parts of a window 156
- 3.6 Using Quadrant-Coordinates 356
- IRC. / NRC, - If Rect Color 1828
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801
- ILC. / NLC. - If Line Color 1805
- RRA. - Remember Rectangular Area 1836
- IRA. / NRA. - If Rect altered 1824
- WRA. - Wait Rect-Area Altered 1851
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7 Pattern-Search

These Commands locate Patterns that have been recorded before.

They are similar to the Pattern-Commands that are available with the ANA.-Command.

Here the most used Parameters can be set as "default" using the

SFP. - Command (Set Find Pattern Parameters) 1899
GFP. - Command (Get Find Pattern Parameters) 1869

therefore the Pattern-Commands can be used with only very few Parameters.

The Pattern Commands of this Group include:

JFP. / JNP. - Jump if Pattern (not) found 1881
IFP. - NFP. (Not) If Pattern Found 1873

LFP. - Locate Find Pattern 1890

and

CFP. - Case Find Pattern 2993

The CFP.-Command is documented in the Section with the SCS.-Command 2982

From all these Commands there is a version that can search for 2 Pattern in one run.
These are the Versions with a "T": LTP. , JTP, etc.

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.1 ! How to record Patterns

Supported Pattern-Sizes

The following Pattern-Sizes are supported:

```
4x16
4x32
4x64
8x8
8x16
8x32
8x64
9x9
10x10
11x11
12x12
16x8
24x8
32x4
32x8
32x16
40x8
48x8
56x8
64x6
64x8
96x4
96x8
98x6
128x4
128x8
```

The "Debug-Flag"

The Debug Flag has been constantly enhanced during development. Currently the following Options are supported:

1 - Switch Debug Console ON

2 - Use "Snap again" during Search. This will update the Snapshot using the Parameters from the last taken Snapshot.

4 - Use "X-Ray Search". This will start an enhanced search that is more color-independent and takes much longer time.
Especially together with Option "64 -Extended Search" it may find a lot more results.
Using this Option, the most upper Line and the most left Line are ignored.
Therefore this Option will reduce the effective Pattern-Size by -1 in X- and Y-Direction.
For example a Pattern that was size "8x8" will be a Pattern size "7x7", this is important when you specify needed "Pixelhits".

*© 2023 Theo Gottwald * https://www.smart-package.com*

8 - If added, this will disable one Layer of "Fuzzy-Logic". This Flag will make the Search more sensitive to small color-differences

16 - If used, this Flag will internally invert the Images. Use it preferably with Option "4 - X-Ray Search".

32 - If used, this Flag will internally complement the Images. Use it preferably with Option "4 - X-Ray Search".

64 - "Extended Search". This search uses longer time, but returns more results. Use it preferably with Option "4 - X-Ray Search"

Use Picture-File as Pattern

Generally you can use any picture file for the Pattern-Search commands.
Please be aware that the resolution of the picture in the Image Register must be one of the allowed Pattern Resolutions.
Please use only Picture-File Formats that do compress losseless.
Lossless compression reduces file size without removing any bits of information. These are for example: ".bmp", ".png".
Do not use: ".jpg".

For this specify a Filepath/Filename.

```
$$FIL=?pattern\MyPattern.bmp
LFP.$$FIL
```

Use Inline-Data Pattern

You can use Inline Data for the Pattern-Search commands.
Instead of specifying a Filepath/Filename, you just specify

```
LFP.ID:$$DTA
```

The term ID: tells the Robot that the following variable contains Inline-Data.
And therefore will only be carefully binary resolved.

Use Image-Register as Pattern

You can use a Picture that is in teh Image-Register for the the Pattern-Search commands.
Please be aware that the resolution of the picture in the Image Register must be one of the allowed Pattern Resolutions.
Instead of specifying a Filepath/Filename, you just specify

```
LFP.IR:($$Num+1)
```

The term IR: tells the Robot that the following variable is numeric and specifies the Image-Register Number to use..

How to record Patterns

When recording a Pattern you should take a few things into account if you want the Pattern to be found faster.

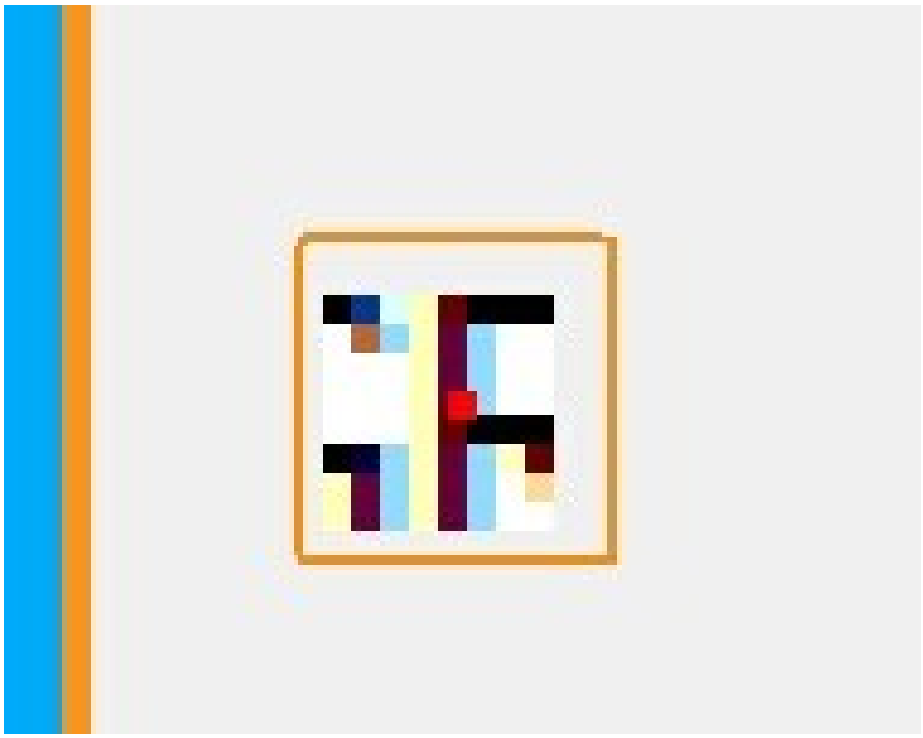
© 2023 Theo Gottwald * https://www.smart-package.com

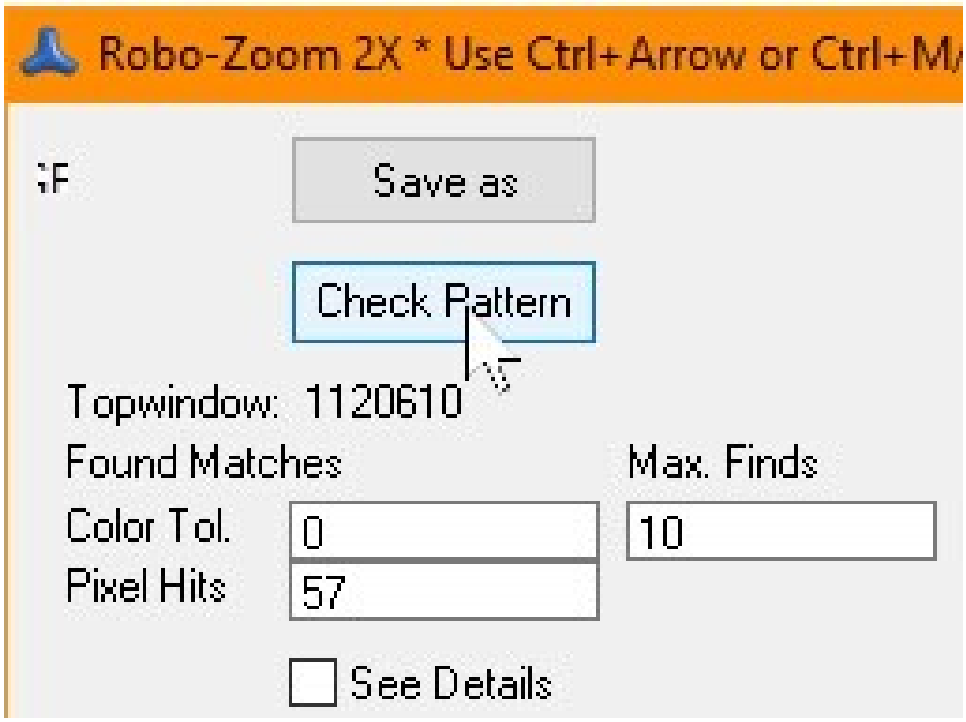
Here are some good and bad examples:

Here is a good Pattern

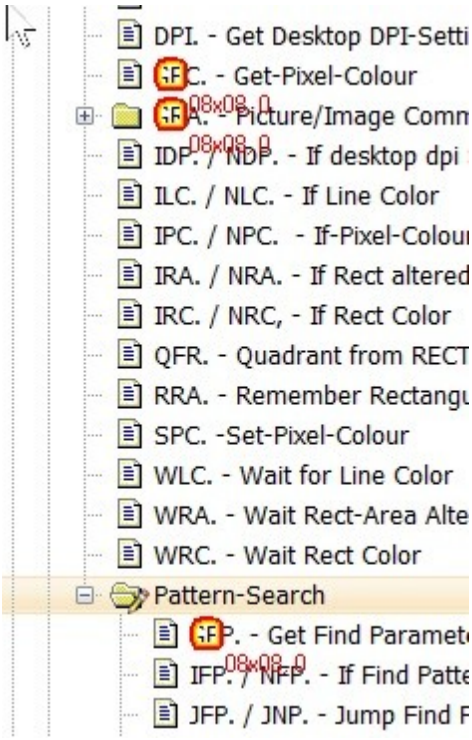
A good Pattern contains contrasted Pixels in the first Line and on the first 5 Pixesl.
Such Patterns will be found very fast.
You can test this if you press the "Check Pattern" button in the HE-Recorder.
A good Pattern should be found in less than a second on a good Computer. Even on a 4K Screen.

© 2023 Theo Gottwald * https://www.smart-package.com





© 2023 Theo Gottwald * https://www.smart-package.com

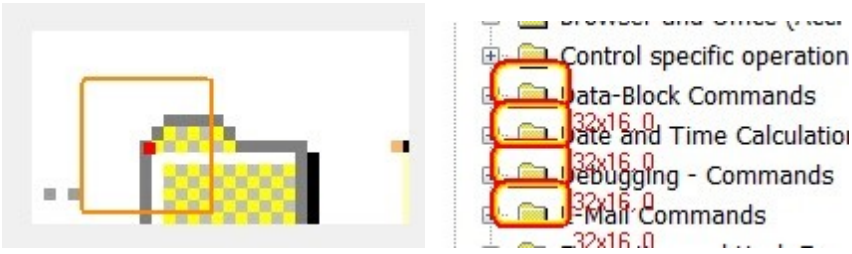


This is a good Pattern that will be found quickly.

Now lets see a *less good Pattern*.

This Pattern has two Speed-Penalties:
1. Its a large Pattern with 32x16 Points
2. The Top-Lines are all empty

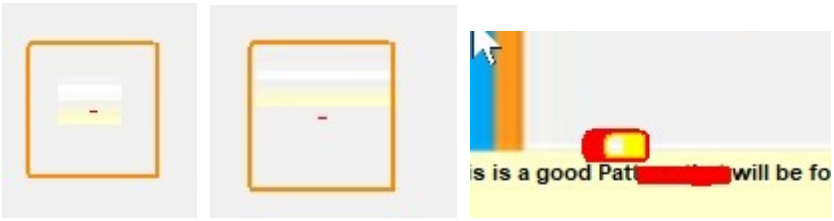
Still it will be found in a Second.
Please do not wonder if not all possible Finds are being marked.
Default the number of finds is now limited to 10.
The finds must not be sequential, due to the multi-threading core of the Find-Pattern Engine.
Also the sequence of the Finds is not guaranteed to always be in the same order.

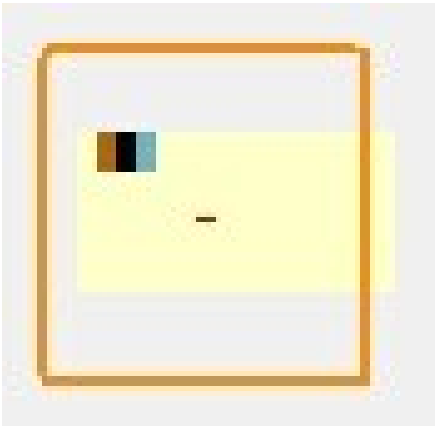


This Pattern is not so good as its larger and the Top-Lines all have the same colour.

A Pattern that is just few Lines is also not a very good Pattern.
Also choose the smallest possible Pattern-Size to optimize for Speed.

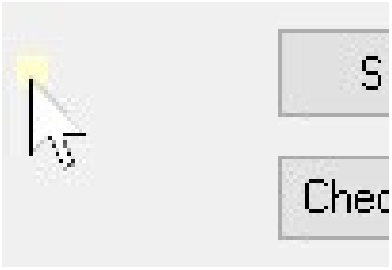
*© 2023 Theo Gottwald * https://www.smart-package.com*





Pattern that do not contain a lot of Information should try to have these in the upper left corner.

And finally this Pattern is a the worst case for the Pattern-Find Engine:



This Pattern is just the Background color.

A Pattern that is just a Background Color is not a Pattern and can not easily be matched.
The Pattern-Engine will search and find also this Pattern, but depending on how often the color is available, this may take a while.

Remarks
-

Limitations:
-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.2 GFP. - Get Find Parameter

MiniRobotLanguage (MRL)

GFP. Get Find Pattern Parameter

Read internal Parameters for the Find Pattern Commands

Intention

This command helps to manage the Standard-Parameters for the Find-Pattern Commands.
To have the Find-Pattern Commands use only **3 Parameters,** all less important Parameters can be set using the SFP.-Command.
The GFP-Command will help you to read these **additional Parameters and several more Parameters that are set by the System.**

Such Parameters are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

Use SFP. and GFP. to set and Read the default Parameters

```
' Set and Read the value of "MaxResults"
SFP.mr|12
GFP.mr|$$RES
MBX.$$RES
ENR.

' Set the maximum Number of used CPU-Cores to 8
SFP.CPU|8
GFP.CPU|$$RES
MBX.$$RES
```

Explanation for the Parameter-Names:

"MaxResults" or "Mr"
This will tell the Pattern-Engine to stop searching for more results after this number of results has been found.

"PixelHits" or "Ph"
This number will tell the Engine how many Pixels from the Target must match the Source Pattern (taking the Color-Tolerance into account)
so that the Location is been marked as a "Hit". This Parameter must be set depending on the Pattern Size. If you leave it at "0" a default, calculated value is taken (85%).

"ColorTolerance" or "Ct"

*© 2023 Theo Gottwald * https://www.smart-package.com*

This number will define the Color-Tolerance that the Pattern-Search Engine applies to each Pixel before marking it as "valid" or "Invalid".
The higher this value, the more possible results can be found.

"DebugMode" or "db"
This number can only be 0 or 1. If set to one, after there have been results found, a Console Window will appear and list all the found results together with some technical informations, like the "Pixelhits" that have matched for the Find. Using the Debugmode you can find the best values for Tolerance and Pixelhits by looking over the results. If nothing is found, possibly no Console Window will appear.

"CPU"
Number of CPU-Cores in the System. If change this, the Subprogrammes may use less CPU-Cores and work possibly slower. If you set it too high (Higher then the real number of available CPU-Cores) then the System may become overloaded and get unresponsible. Best leave at the default value that is calculated when the Robot starts.

"Snap_Border" or "sb"
This can be set to 0, 1 or 2 and will influence the Bordersize that is taken when Snapshoting a Window.
"0" - Client-Area only
"1" - Whole Window
"2" - Whole WIndow without Border
It depends on the Target which is the best setting.

XSize, YSize
This gives you the X- and Y-Resolution of the last Snapshot, using any of the Pattern-Commands that do Snapshot a Window. Including `ANA.Snap`.
There are no know use cases where you would like to change these numbers. Doing so may have unforeseeable Side-Effects on some commands.

"MovePoint", "MP"
This is the preset for the Point-Shift Parameter. By default it is "0,0". The Preset is not yet implemented in all Pattern-Commands.

"LastPattern", "LP"
This is will return the number of the Pattern that was found if a "multi-Pattern-Command" was used. It is working for WTP. and LTP.-Commands.
Using this, you can find out which of multiple Patterns was found for example with

```
WTP.$$PAA|$$PAB|30,40|60
GFP.LastPattern|$$PAT
MBX.$$PAT
```

This will only make sense, if only 1 result was found.
In this case, `$$PAT` will be 1 or 2 depending if the Pattern in `$$PAA` or `$$PAB` was found.

"LastCount", "LC"
This will return the number of found matches from the last Pattern-Command. It may not yet be implemented in all Pattern-Commands. It is working for WTP. and LTP.-Commands where its mostly needed.

"LastHits", "LH"
This will return the number of Pixelhits that were diagnosed with the last find. Of course this makes only sense if there was just 1 find.

```
WTP.$$PAA|$$PAB|30,40|60
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GFP.LH|$$PAT
MBX.Pixelhits: $$PAT
```

"LastSnap", "LS"
This will return the Number of the IR that got the last Snapshot (Read Only).

"FlagA", "FA"
This will return the Value of Flag "A". It has to do with Pattern Search. Generally the normal value is "0".
If set to 1,2 or 3 the Pattern-Operation check more Pixels before returning a result, this may lead to better yet may use longer time.

Syntax

GFP.P1[|P2]

Parameter Explanation

P1 - Parameter-Name this is the Name of the Parameter that is going to be changed. Please note that some Parameters-Names refer to "read-only" Parameters that can not be changed or may result in strange side-effects if you change them.

- MaxResults
- PixelHits
- ColorTolerance
- DebugMode
- HWnd
- Use_First
- Snap_Border
- CPU
- Snap_Border
- MovePoint
- LastPattern
- LastCount
- LastHits
- FlagA

Then there are several "Read Only" Parameter-Names
- LastSnap (RO)
- XSize (RO)
- YSize (RO)
- Result (RO)

P2 - opt. Variable The variable will return the value that is stored inside the given Parameter. If omitted TOS is used for the result.

Example
-

Remarks
-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:
The values are not checked for validity. Specifying nonsense values may have unforeseeable side-effects.

See also:

- SFP. - Set Find Parameter 1899

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.3 IFP. / NFP. - If Find Pattern

MiniRobotLanguage (MRL)

IFP. - If Find Pattern
NFP. - Not Find Pattern

Check if a specified Pattern can be found in an Image-Register (IR)

Intention

This command will just check if a specified Pattern can be found or not.

If the Pattern can be found, it will then find the Locations where a specified Object is in an Image.
And put the number of found occurrences on the TOS, as well as all occurrences limited by the Setting of MaxResults (See SFP.-Command).
It uses only 1 Pattern and **3 Parameter** which are similar to the Parameters in LFP.

NFP. is generally just the Opposite of IFP. technically just the Blocks of "IF .(Block 1). ELS. .(Block 2). EIF." are swapped for NFP.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The IFP./NFP.-Command will use **additional Parameters that can be changed using the SFP.-Command.**
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

Technically the IFP. is an enhanced Version of the WFP.-Command, as it has the same Parameters. With the difference that if you do NOT specify the Timeout-Value **P3**,

then WFP. will "wait forever" while IFP. will "return immediately". You can change this behavior with the P3-Value. Generally "0" means "wait forever" while "-1" means "return immediately".

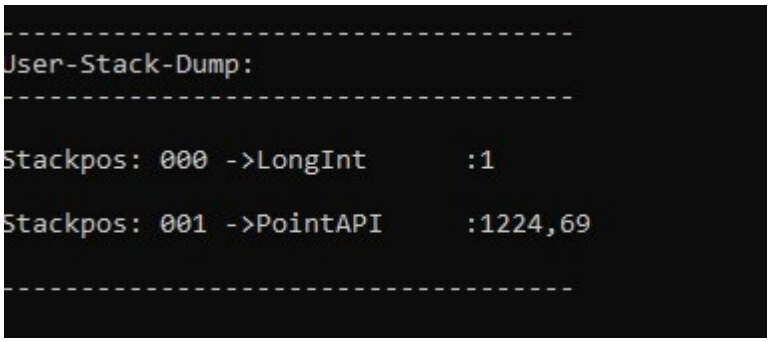
Please see the LFP.-Commands for more Details.

1. Use IFP with 3 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
SWP.0,0|2807,1800|T
$$LAA=?pattern\JK_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust
IFP.$$LAA|0,50
   MMV.
ELS.
   MBX.Pattern was not found.
EIF.
' And here the same Sample using NFP.
NFP.$$LAA|0,50
   MBX.Pattern was not found.
ELS.
   MMV.
EIF.
ENR.
```



If you Specify P3, then P3 is the number of findings, limited by thesetting of Max-Results (see SFP.-Command).
On the TOS you will find all the Results.

2. Advance Use of IFP. / NFP.

You can change more advanced Parameters that IFP. uses with the SFP.-Command.
You can use Inline-Data using "ID:$$DTA".
And you can use directly pictures in an Image-Register - for example 1 - using "IR:1"

```
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust
IFP.$$LAA|0,50
   MMV.
ELS.
   MBX.Pattern was not found.
EIF.
' And here the same Sample using NFP.
NFP.$$LAA|0,50
   MBX.Pattern was not found.
ELS.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    MMV.
  EIF.
  ENR.
```

Syntax

```
IFP.P1[|P2][|P3][|P4] ...
ELS. ... EIF.
NFP.P1[|P2][|P3][|P4] ...
ELS. ... EIF.
```

Parameter Explanation

P1 - Pattern File-Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
***or Specify "ID:*$$DTA" und directly use Inline-Data as Pattern.**

P2 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.

P3 - opt. Wait-Time in Seconds Command will wait until the time is over if that Pattern will appear.
If not specified the Command will return immediately. If 0 is specified the command will wait "Forever". If "-1" is specified the command will not wait and therefore return immediately.

P4 - opt. Variable for Result. This variable will contain the number of Finds.
If omitted the result will be placed on TOS.

The Command will leave a the number of finds in **P3** (if given) or the TOS.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".
The last found Position will automatically be set as the "Active Point".
If the Pattern does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

-

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:
The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.4 ITP. / NTP. - If Find any of Two Pattern

ITP.- / NFP. - If Find Pattern **Previous** 1088 **Top** 41 **Next** 1109

MiniRobotLanguage (MRL)

ITP. - If Find Pattern
NFP. - Not Find Pattern

Check if a specified Pattern can be found in an Image-Register (IR)

Intention

This command will check if at least one of two specified Pattern will be found in the specified **IR**.
You will find the number of the Pattern (1 or 2) that has appeared on the TOS below the Coordinates.

It will then find the Locations where the specified Object is and locate the Coordinates immediately.

The command can be given a Timeout Value, if this specified time is over the Command will exit and set the Timeout-Flag
and put a "0" on the TOS and use the "Not Found branch" after the "ELS." Command.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

If the Pattern can be found, it will then find the Locations where a specified Object is in an Image.
And put the number of found occurrences on the TOS, as well as all occurrences limited by the Setting of MaxResults (See SFP.-Command).
It uses only 2 Pattern and optional **3 Parameter** which are similar to the Parameters in `LTP.`

NTP. is generally just the Opposite of ITP. technically just the Blocks of "IF .(Block 1). ELS. .(Block 2). EIF." are swapped for NTP.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The `ITP./NFP.`-Command will use **additional Parameters that can be changed using the** `SFP.`**-Command**.
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

Technically the ITP. is an enhanced Version of the WTP.-Command, as it has the same Parameters. With the difference that if you do NOT specify the Timeout-Value

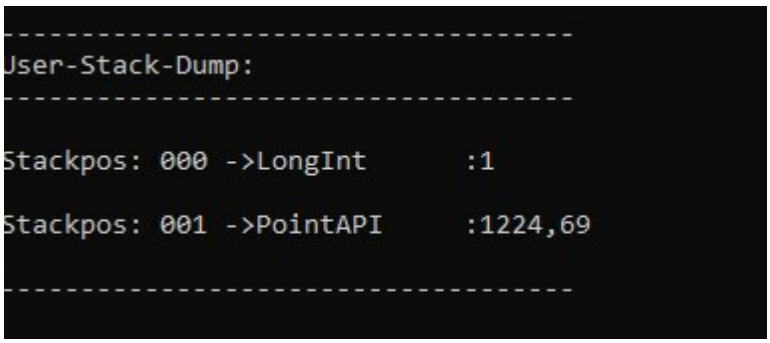
 *© 2023 Theo Gottwald * https://www.smart-package.com*

P3, then WTP. will "wait forever" while ITP. will "return immediately". You can change this behavior with the P3-Value. Genrally "0" means "wait forever" while "-1" means "return immediately".

Please see the LTP.-Commands for more Details.

1. Use ITP with 3 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
SWP.0,0|2807,1800|T
$$LAA=?pattern\JK_8.bmp
$$LAB=?pattern\JK_10.bmp
' P1/P2 - Pattern-Files, P3 - Target-Position adjust
ITP.$$LAA|$$LAB|0,50
   MMV.
ELS.
   MBX.Pattern was not found.
EIF.
' And here the same Sample using NFP.
NTP.$$LAA|$$LAB|0,50
   MBX.Pattern was not found.
ELS.
   MMV.
EIF.
ENR.
```



If you Specify P3, then P3 is the number of findings, limited by thesetting of Max-Results (see SFP.-Command).
On the TOS you will find all the Results.

2. Advance Use of ITP. / NTP.

You can change more advanced Parameters that ITP. uses with the SFP.-Command.
You can use Inline-Data using "ID:$$DTA".
And you can use directly pictures in an Image-Register - for example 1 - using "IR:1"

```
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust
ITP.$$LAA|0,50
   MMV.
ELS.
   MBX.Pattern was not found.
EIF.
' And here the same Sample using NFP.
NFP.$$LAA|0,50
   MBX.Pattern was not found.
ELS.
   MMV.
EIF.
ENR.
```

Syntax

ITP.P1|P2[|P3][|P4][|P5] ...
ELS. ... EIF.
NTP.P1|P2[|P3][|P4][|P5] ...
ELS. ... EIF.

Parameter Explanation

P1 - Pattern A File-Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
***or Specify "ID*:$$DTA" und directly use Inline-Data as Pattern.**

P2 - Pattern B File-Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
***or Specify "ID*:$$DTA" und directly use Inline-Data as Pattern.**

P3 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.

P4 - opt. Wait-Time in Seconds Command will wait until the time is over if that Pattern will appear.
If not specified the Command will return immediately. If 0 is specified the command will wait "Forever". If "-1" is specified the command will not wait and therefore return immediately.

P5 - opt. Variable for Result. This variable will contain the number of Finds.
If omitted the result will be placed on TOS.

© 2023 Theo Gottwald * https://www.smart-package.com

The Command will leave a the number of finds in **P5** (if given) or the TOS.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches. The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".
The last found Position will automatically be set as the "Active Point".
If the Pattern does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

-

Remarks

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.5 JFP. / JNP. - Jump Find Pattern

MiniRobotLanguage (MRL)

JFP. - Jump if Find Pattern
JNP. - Jump if Not Find Pattern

Check if a specified Pattern can be found in an Image-Register (IR) and if so Jump to a Label

Intention

This command check if a specified Pattern can be found or not. If the Condition is met, the Jump to the specified Label will be made.

If the Pattern can be found, it will then find the Locations where a specified Object is in an Image.
And put the number of found occurrences on the TOS, as well as all occurrences limited by the Setting of MaxResults (See SFP.-Command).
It uses only **3 Parameter** which are similar to the Parameters in LFP..

NFP. is generally just the Opposite of IFP. technically just the Blocks of "IF .(Block 1). ELS. .(Block 2). EIF." are swapped for NFP.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The IFP./NFP.-Command will use **additional Parameters that can be changed using the SFP.-Command**.
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

Technically the IFP. is an enhanced Version of the WFP.-Command, as it has the same Parameters. With the differnece that if you do NOT specify the Timeout-Value **P3**,
then WFP. will "wait forever" while IFP. will "return immediately". You can change this behavior with the P3-Value. Genrally "0" means "wait forever" while "-1" means "return immediately".

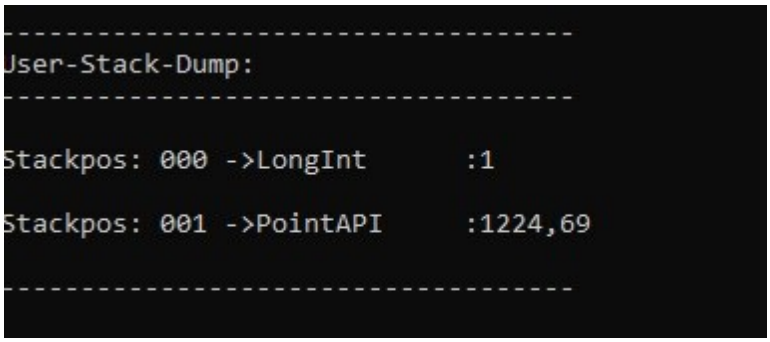
Please see the LFP.-Commands for more Details.

1. Use WFP with 3 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SWP.0,0|2807,1800|T
$$LAA=?pattern\JK_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust
IFP.$$LAA|0,50
   MMV.
ELS.
   MBX.Pattern was not found.
EIF.
' And here the same Sample using NFP.
NFP.$$LAA|0,50
   MBX.Pattern was not found.
ELS.
   MMV.
EIF.
ENR.
```



If you Specify P3, then P3 is the number of findings, limited by thesetting of Max-Results (see SFP.-Command).
On the TOS you will find all the Results.

2. Use IFP. and set the default Parameters with SFP. before

You can change more advanced Parameters that WFP. uses with the SFP.-Command.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust
IFP.$$LAA|0,50
   MMV.
ELS.
   MBX.Pattern was not found.
EIF.
' And here the same Sample using NFP.
NFP.$$LAA|0,50
   MBX.Pattern was not found.
ELS.
   MMV.
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

JFP.P1[|P2][|P3][|P4]
JNP.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - Pattern File-Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P2 - Label - Label where the Jump will go if the condition is met. With the SPR Labels can also be dynamic and in Variables.

P3 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.

P4 - opt. Wait-Time in Seconds Command will wait until the time is over if that Pattern will appear.
If not specified the Command will return immediately.
If 0 is specified the command will wait "Forever". If "-1" is specified the command will not wait and therefore return immediately, this is the default option.

The Command will leave the number of finds on the TOS.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".
The last found Position will automatically be set as the "Active Point".
If the Pattern does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

-

Remarks

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.6 JTP. / JTN. - Jump Two Pattern

JTP.- / JTN. - Jump If (n)one of Two Pattern **Previous** 1088 **Top** 41 **Next** 1109

MiniRobotLanguage (MRL)

JTP. - Jump if one of Two Pattern
JTN. - Jump if Two Pattern Not found

Check if one or more of two specified Pattern can be found in an Image-Register (IR) and if so Jump to a Label

Intention

This command check if a specified Pattern can be found or not. If the Condition is met, the Jump to the specified Label will be made using JTP.
JNT is just the opposite, it will jump in none of the two Patterns was found.

And put the number of found occurrences on the TOS, as well as all occurrences limited by the Setting of MaxResults (See SFP.-Command).
It uses only **3 Parameter** which are similar to the Parameters in LFP..

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The JTP./JTN.-Command will use **additional Parameters that can be changed using the SFP.-Command**.
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

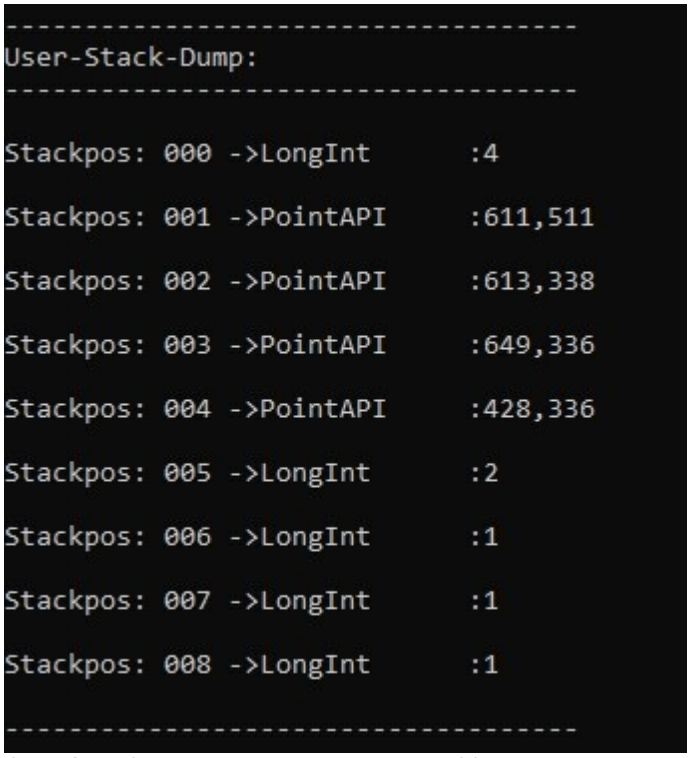
Technically the JTP. is an enhanced Version of the JFP.-Command, as it has the same Parameters but you can specify two Patterns.
Please see the LFP.-Commands for more Details.

1. Use WFP with 3 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|Chrome_Widge|Safir - Brav
SWP.0,1320|1098,668|T
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
SFP.ColorTolerance|0
SFP.PixelHits|0
SFP.DebugMode|0
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
JTP.$$PAA|$$PAB|out|0,0|10
MBX.Pattern found
ENR.
:out
MMV.
DMP.6
MBX.Pattern found
ENR.
```



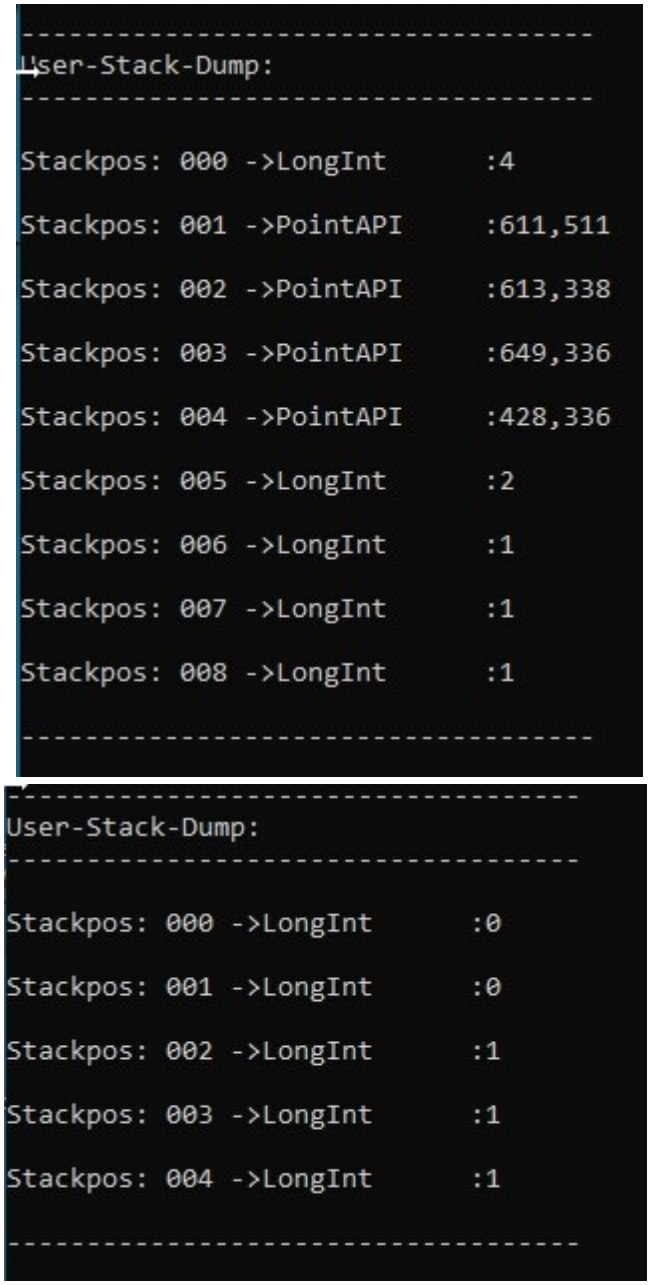
If you Specify P3, then P3 is the number of findings, limited by thesetting of Max-Results (see SFP.-Command).
On the TOS you will find all the Results.

Very useful is the negative Form of the Command. Assume taht you Click some Button and you expect a Menu to appear.
Now using this one Command, you can locate the Menu (or whatever the Pattern is). And in the same Command, you have a variable Waiting Time (to wait a moment if it appears), and if the Pattern does not appear you can jump back to Click again.
All in one Command.

```
STW.ct|Chrome_Widge|Safir - Brav
SWP.0,1320|1098,668|T
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
SFP.ColorTolerance|0
SFP.PixelHits|0
SFP.DebugMode|0
JTN.$$PAA|$$PAB|out|0,0|1
MMV.
DMP.6
MBX.Pattern found
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
:out
MBX.Pattern not found
ENR.
```



The first Picture shows the Stack from JTN. if the Pattern was faound. The second Picture shows the Stack if the Pattern was not found.

Syntax

JTP.P1|P2[|P3][|P4][|P5]
JTN.P1|P2[|P3][|P4][|P5]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - Pattern File A - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P2 - Pattern File B - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P3 - Label - Label where the Jump will go if the condition is met. With the SPR Labels can also be dynamic and in Variables.

P4 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.

P5 - opt. Wait-Time in Seconds Command will wait until the time is over if that Pattern will appear.
If not specified the Command will return immediately.
If 0 is specified the command will wait "Forever". If "-1" is specified the command will not wait and therefore return immediately, this is the default option.

The Command will leave the number of finds on the TOS.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".
The last found Position will automatically be set as the "Active Point".
If the Pattern does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

-

Remarks

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.7 LFP. - Locate Find Pattern

MiniRobotLanguage (MRL)

LFP. Locate and Find a Pattern

Get Locations of a specified Pattern-Object in an Image-Register (IR)

Intention

This command will find the Locations where a specified Object is in an Image. It uses only **3 Parameter** and is easy to use.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The LFP-Command will use **additional Parameters that can be changed using the SFP.-Command.**
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

To use the FindPattern-Command, there is additional Program called "Human-Eye-Recorder" available with this distribution.
See a Description here: ***Human Eye-Recorder*** 1140

Using the HE-Recorder the recording of Patterns is easy and fast.

Here is an important Notice: if you record a Pattern, and you use "Check Pattern" and the Pattern is not found, then you have "Mouseover Effects".
This means that the Color of the Object changes if Mouse is nearby or over the Object.
You may **not** see this change visually.
The Pattern Matcher will definitely see it and avoid the Pattern, if Color Tolerance is zwero.

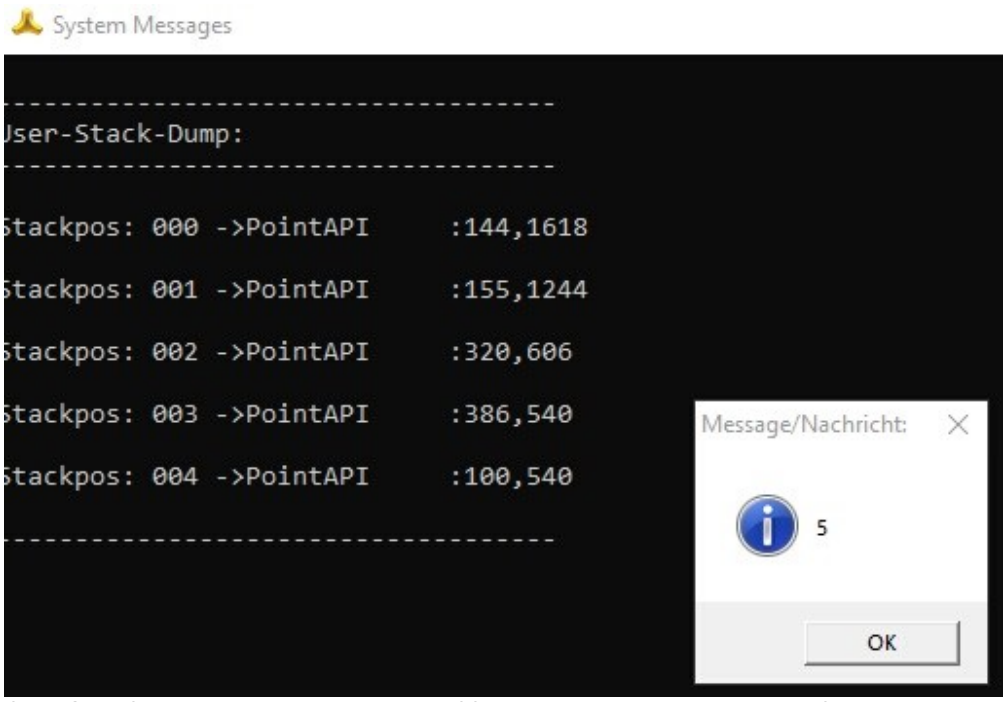
You can handle these Events best, if you first make a Screenshot from that Window and record the Pattern from the Screenshot.
Again, do never use a JPG or loosing Compression for working with the Pattern-Matcher Commands.
Due to Compression Artefacts the quality of the results is unknown. **Best use ".bmp" or ".png".**

Alternatively you can do the recording with an higher Color-Tolerance.
Which is not the best choice - but will work most often.

1. Use LFP with 3 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
STW.ct|PBWindowClass:0|JK - IDE 2.
$$LAA=?pattern\SP_8.bmp
' LFP. uses only 3 Parameters
LFP.$$LAA|0,40|$$RES
DMP.6
MBX.$$RES
ENR.
```

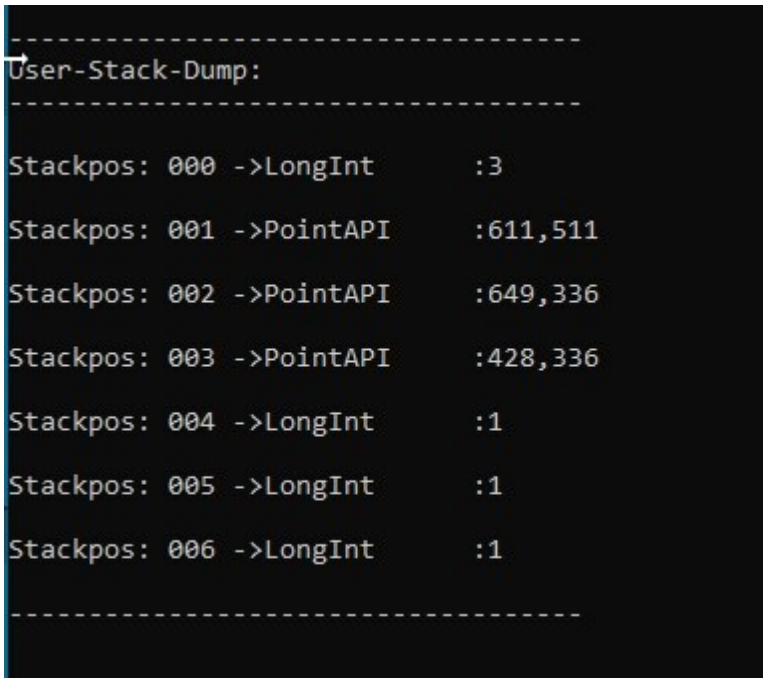


If you Specify P3, then P3 is the number of findings, limited by thesetting of Max-Results (see SFP.-Command).
On the TOS you will find all the Results.

If you omit the $$RES then you get the number of finds on TOS.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
$$LAA=?pattern\SP_8.bmp
' LFP. uses only 3 Parameters
LFP.$$LAA|0,40
DMP.6
MBX.$$RES
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



2. Use LFP. and set the default Parameters with SFP. before
You can change the Parameters that LFP. uses with the SFP.-Command.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
' LFP. uses only 3 Parameters but internally also uses the Parameters that can b
LFP.$$LAA|0,40|$$RES
DMP.6
MBX.$$RES
ENR.
```

Syntax

LFP.P1[|P2][|P3]

Parameter Explanation

P1 - Pattern File - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
***or Specify "ID*:$$DTA" und directly use Inline-Data as Pattern.**

P2 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.

© 2023 Theo Gottwald * https://www.smart-package.com

For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.
P3 - opt. Variable for Result. if omitted the result will be placed on TOS.

The Command will leave a the number of finds in P3 (if given) or the TOS.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".

The last found Position will automatically set as the "Active Point".

If the Pattern does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

-

Remarks

Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.
5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again. Check that there are no empty Lines on Top of the Pattern.
7. A pattern that is completely empty will not work as there is nothing to recognize in it.

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.8 LTP. - Locate Two Pattern

MiniRobotLanguage (MRL)

LTP. Locate and Find one or two Pattern

Get Locations of one or two specified Pattern-Objects in an Image-Register (IR)

Intention

This command will find the Locations where a specified Objects are in an Image. It uses only 4 **Parameter** and is easy to use.
Assume that you want to recognize the state in which a GUI is, that is not accessible to other commands.
Then you can record 2 Patterns and locate them using a single command.

If there are more then 2 Patterns that shall be taken into consideration, maybe use the SCS. and the CFP.-Command
it can work with an unlimited number of Patterns.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The LFP-Command will use **additional Parameters that can be changed using the SFP.-Command.**
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

To use the FindPattern-Command, there is additional Program called "Human-Eye-Recorder" available with this distribution.
See a Description here: ***Human Eye-Recorder*** 1140

Using the HE-Recorder the recording of Patterns is easy and fast.

Here is an important Notice: if you record a Pattern, and you use "Check Pattern" and the Pattern is not found, then you have "Mouseover Effects".
This means that the Color of the Object changes if Mouse is nearby or over the Object.
You may **not** see this change visually.
The Pattern Matcher will definitely see it and avoid the Pattern, if Color Tolerance is zwero.

You can handle these Events best, if you first make a Screenshot from that Window and record the Pattern from the Screenshot.
Again, do never use a JPG or loosing Compression for working with the Pattern-Matcher Commands.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Due to Compression Artefacts the quality of the results is unknown. **Best use ".bmp" or ".png".**

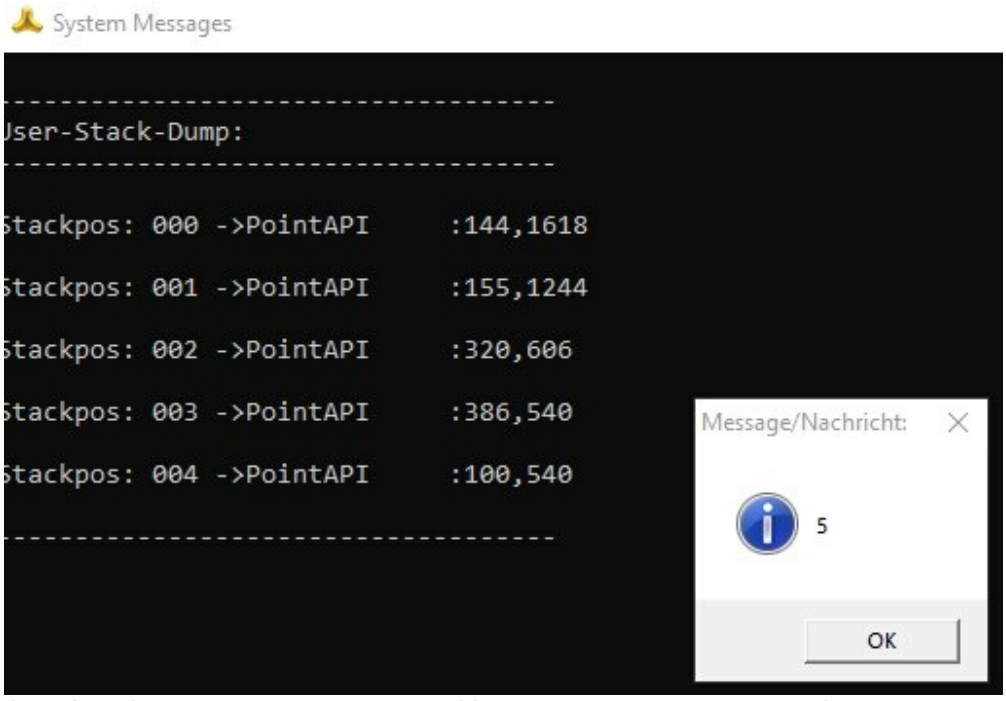
Alternatively you can do the recording with an higher Color-Tolerance.
Which is not the best choice - but will work most often.

Now you can find out which of multiple Patterns was found for example with

```
WTP.$$PAA|$$PAB|30,40|60
GFP.LastPattern|$$PAT
MBX.$$PAT
```

1. Use LTP with 4 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
' LTP. uses 4 Parameters
LTP.$$PAA|$$PAB|0,0|$$RES
DMP.6
MBX.$$RES
ENR.
```



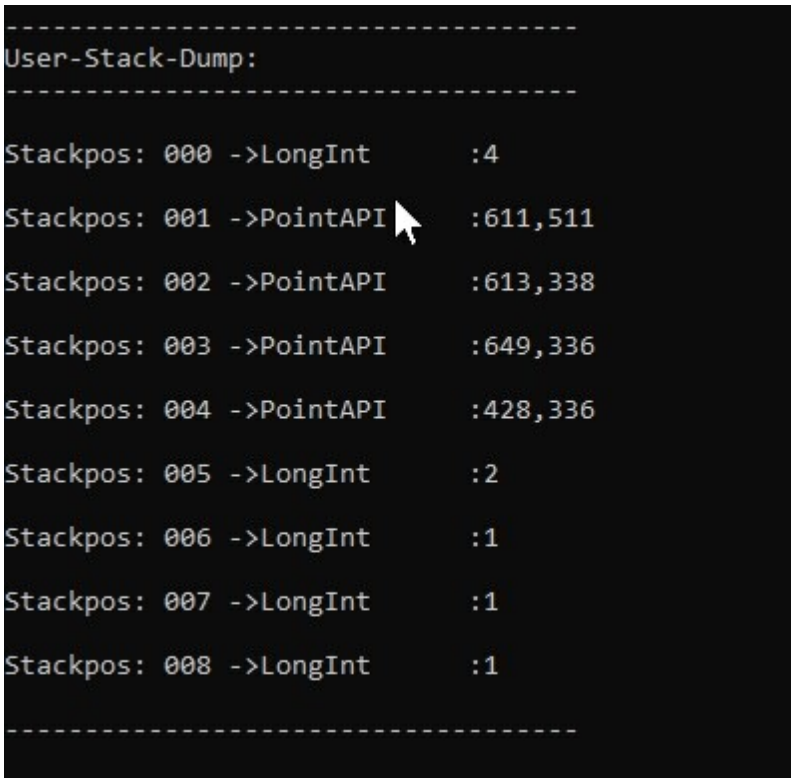
If you Specify P3, then P3 is the number of findings, limited by thesetting of Max-Results (see SFP.-Command).
On the TOS you will find all the Results.

If you omit the $$RES then you get the number of finds on TOS.

```
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
STW.ct|PBWindowClass:0|JK - IDE 2.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$LAA=?pattern\SP_5.bmp
' LTP. uses only 3 Parameters
LTP.$$PAA|$$PAB|0,0
DMP.6
MBX.$$RES
ENR.
```



2. Use LTP. and set the default Parameters with SFP. before
You can change the Parameters that LTP. uses with the SFP.-Command.

```
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
LTP.$$PAA|$$PAB|0,0|$$RES
DMP.6
MBX.$$RES
ENR.
```

Syntax

LTP.P1|P2[|P3][|P4]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - Pattern File A - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P2 - Pattern File B - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P3 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.
P4 - opt. Variable for Result. if omitted the result will be placed on TOS.

The Command will leave a the number of finds in P4 (if given) or the TOS.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".
The last found Position will automatically set as the "Active Point".

If the Pattern does not exist then you will get a result "0" on TOS. The program will **not generate an Error-Message**.

Example

-

Remarks

Speed Considerations:
1. The Command is "Multithreaded" and works best with a Multicore CPU.
2. The Command will need a lot of CPU-Power, it may also use the GPU at some point internally.
3. Generally the Command is very fast even in very large windows, as long as the Computer has a modern CPU.
4. The larger the Pattern, the slower the process.
5. Currently it may speed up the process, if you have the pattern touch the upper left corner. This may change in the future.
6. If a Pattern is very slow, just record another Pattern. And try this again. Check that there are no empty Lines on Top of the Pattern.
7. A pattern that is completely empty will not work as there is nothing to recognize in it.

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:
The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.9 SFP. - Set Find Parameter

MiniRobotLanguage (MRL)

SFP. Set Find Pattern Parameter

Set internal Parameters for the Find Pattern Commands

Intention

This command manages the Standard-Parameters for the Find-Pattern Commands.
To have the Find-Pattern Commands use only **3 Parameters,** all less important Parameters can be set using this Command.

The SFP-Command will set all **additional Parameters**.

These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

1. Use LFP. and set the default Parameters with SFP. before
You can change the Parameters that LFP. uses with the SFP.-Command.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
' LFP. uses only 3 Parameters but internally also uses the Parameters that can b
LFP.$$LAA|0,40|$$RES
DMP.6
MBX.$$RES
ENR.
```

Explanation for the Parameter-Names:

"MaxResults" or "Mr"
This will tell the Pattern-Engine to stop searching for more results after this number of results has been found.

"PixelHits" or "Ph"
This number will tell the Engine how many Pixels from the Target must match the Source Pattern (taking the Color-Tolerance into account)
so that the Location is been marked as a "Hit". This Parameter must be set depending on the Pattern Size. If you leave it at "0" a default, calculated value is taken (85%).

*© 2023 Theo Gottwald * https://www.smart-package.com*

"ColorTolerance" or "Ct"
This number will define the Color-Tolerance that the Pattern-Search Engine applies to each Pixel before marking it as "valid" or "Invalid".
The higher this value, the more possible results can be found.

"DebugMode" or "db"
This number can only be 0 or 1. If set to one, after there have been results found, a Console Window will appear and list all the found results together with some technical informations, like the "Pixelhits" that have matched for the Find. Using the Debugmode you can find the best values for Tolerance and Pixelhits by looking over the results. If nothing is found, possibly no Console Window will appear.

"CPU"
Number of CPU-Cores in the System. If change this, the Subprogrammes may use less CPU-Cores and work possibly slower. If you set it too high (Higher then the real number of available CPU-Cores) then the System may become overloaded and get unresponsible. Best leave at the default value that is calculated when the Robot starts.

"Snap_Border" or "sb"
This can be set to 0, 1 or 2 and will influence the Bordersize that is taken when Snapshoting a Window.
"0" - Client-Area only
"1" - Whole Window
"2" - Whole WIndow without Border
It depends on the Target which is the best setting.

XSize, YSize
This gives you the X- and Y-Resolution of the last Snapshot, using any of the Pattern-Commands that do Snapshot a Window. Including `ANA.Snap`.
There are no know use cases where you would like to change these numbers. Doing so may have unforeseeable Side-Effects on some commands.

"MovePoint", "MP"
This is the preset for the Point-Shift Parameter. By default it is "0,0". The Preset is not yet implemented in all Pattern-Commands.

"LastPattern", "LP"
This is will return the number of the Pattern that was found if a "multi-Pattern-Command" was used. It is working for WTP. and LTP.-Commands.
Using this, you can find out which of multiple Patterns was found for example with

```
WTP.$$PAA|$$PAB|30,40|60
GFP.LastPattern|$$PAT
MBX.$$PAT
```

This will only make sense, if only 1 result was found.
In this case, `$$PAT` will be 1 or 2 depending if the Pattern in `$$PAA` or `$$PAB` was found.

"LastCount", "LC"
This will return the number of found matches from the last Pattern-Command. It may not yet be implemented in all Pattern-Commands. It is working for WTP. and LTP.-Commands where its mostly needed.

"LastHits", "LH"
This will return the number of Pixelhits that were diagnosed with the last find. Of course this makes only sense if there was just 1 find.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
WTP.$$PAA|$$PAB|30,40|60
GFP.LH|$$PAT
MBX.Pixelhits: $$PAT
```

"LastSnap", "LS"
This will return the Number of the IR that got the last Snapshot (Read Only).

"FlagA", "FA"
This will Set the Value of Flag "A". It has to do with Pattern Search. Generally the normal value is "0".
If set to 1,2 or 3 the Pattern-Operation check more Pixels before returning a result, this may lead to better yet may use longer time.

Syntax

SFP.P1|P2

Parameter Explanation

P1 - Parameter-Name this is the Name of the Parameter that is going to be changed. Please note that some Parameters-Names refer to "read-only" Parameters that can not be changed or may result in strange side-effects if you change them.

- MaxResults
- PixelHits
- ColorTolerance
- DebugMode
- HWnd
- Use_First
- Snap_Border
- CPU
- Snap_Border
- MovePoint
- LastPattern
- LastCount
- LastHits
- FlagA

Then there are several "Read Only" Parameter-Names
- LastSnap (RO)
- XSize (RO)
- YSize (RO)
- Result (RO)

P2 - Value for Parameter The value that is to be stored inside the given Parameter.

The values are not checked for validity. Specifying nonsense values may have unforeseeable side-effects.

Example

```
' Set and Read the value of "MaxResults"
SFP.mr|12
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GFP.mr|$$RES
MBX.$$RES
ENR.

' Set the maximum Number of used CPU-Cores to 8
SFP.CPU|8
GFP.CPU|$$RES
MBX.$$RES
```

Remarks

-

Limitations:

The values are not checked for validity. Specifying nonsense values may have unforeseeable side-effects.

See also:

- GFP-Command 1869

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.10 WFP. - Wait For Pattern

MiniRobotLanguage (MRL)

WFP. Wait For Pattern

Wait for appearance of a Pattern and then get the Locations of a specified Pattern-Object in an Image-Register (IR)

Intention

This command will wait until a specified Pattern will appear or disappear.
It will then find the Locations where a specified Object is in an Image. It uses only **3 Parameter** and is easy to use.
The cammand can be given a Timeout Value, if this specified time is over the Command will exit and set the Timeout-Flag
and put a "0" on the TOS.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

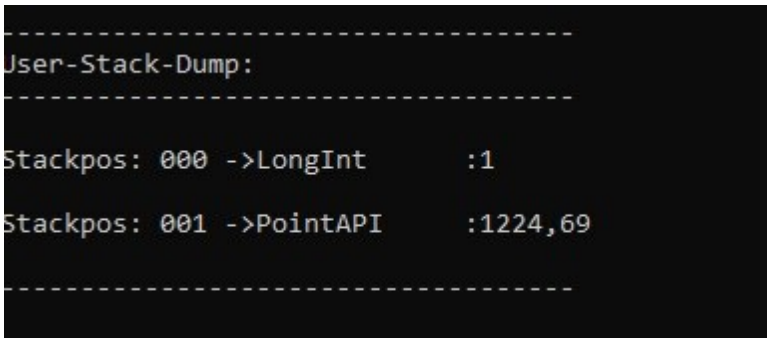
The WFP.-Command will use **additional Parameters that can be changed using the SFP.-Command.**
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

Please see the LFP.-Commands for more Details.

1. Use WFP with 3 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.BAct|PBWindowClass:0|JK - IDE 2.
SWP.0,0|2807,1800|T
$$LAA=?pattern\JK_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust, P3- Timeout in Seconds
WFP.$$LAA|0,50|60
MMV.
ENR.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*



If you Specify P3, then P3 is the timeout value..
On the TOS you will find all the Results.

2. Use WFP. and set the default Parameters with SFP. before

You can change more advanced Parameters that WFP. uses with the SFP.-Command.

```
STW.ct|PBWindowClass:0|JK - IDE 2.
' Set Maxmimum Results Number to 4
SFP.MaxResults|4
' Set Minimum Pixel-Hits to 63
SFP.PixelHits|63
' Set Color-Tolerance to 12
SFP.ColorTolerance|12
$$LAA=?pattern\SP_8.bmp
' P1 - Pattern-File, P2 - Target-Position adjust, P3- Timeout in Seconds
WFP.$$LAA|0,50|60
MMV.
ENR.
```

3. Use WFP. together with Image-Register and Inline-Data

You can use Inline-Data instead of pictures from files.

```
STW.ct|TFrmHelpMan|minirobot.hm
GSB.PattA
ANA.#FromVar|1|$$DTA
$$PAT=IR:1
SFP.ColorTolerance|0
SFP.PixelHits|57
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits1
MMV.
:Lab_NoHits1
ENR.


' You can also use Formula for the IR
WFP.IR:($$PAT+5)|0,0|60
```


© 2023 Theo Gottwald * https://www.smart-package.com

4. Use WFP. directly with Inline-Data

You can use Inline-Data instead of pictures from files.

```
STW.Act|TFrmHelpMan|minirobot.hm
GSB.PattA
$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min. Hits (Pixel) maximu
SFP.ColorTolerance|0
SFP.PixelHits|57
SFP.DebugMode|1
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits1
MMV.
:Lab_NoHits1
ENR.


' Alternativ you can also write:
WFP.ID:$$DTA|0,0|60
```

Syntax

WFP.[!]P1[|P2][|P3]

Parameter Explanation

! **-** If the ! is specified then the Command will wait for the Patterns to disappear, ("Wait for no Pattern")

P1 - Pattern File-Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
***or Specify "ID*:$$DTA" und directly use Inline-Data as Pattern.**

P2 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.

P3 - opt. Timeout value. If omitted the command will wait "Forever".

The Command will leave a the number of finds on the TOS, followed by all finds.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".

© 2023 Theo Gottwald * https://www.smart-package.com

The last found Position will automatically set as the "Active Point".

If the Pattern did not appear in teh given Time, then you will get a result "0" on TOS and the Timeout-Flag will be set.
The program will **not generate an Error-Message**.

Example

-

Remarks

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.7.11 WTP. - Wait For Two Pattern

MiniRobotLanguage (MRL)

WTP. Wait For Two Pattern

Wait for appearance of any of two Patterns and then get the Locations and put it on TOS.

Intention

This command will wait until at least one of two specified Pattern will appear or until none of these is found.
You will find the number of the Pattern (1 or 2) that has appeared on the TOS below the Coordinates.

It will then find the Locations where the specified Object is and locate the Coordinates immediately.

The command can be given a Timeout Value, if this specified time is over the Command will exit and set the Timeout-Flag
and put a "0" on the TOS.

This command uses internally multiple Levels (4 Levels) of Fuzzy-Logic.
It will search all of the currently localized Top-Window.

The WTP.-Command will use **additional Parameters that can be changed using the SFP.-Command.**
These are for example the **Color-Tolerance** and the number of "**needed Pixel-Hits**".
Hereby Pixel-Hits are the number of Pixels that do match the Pattern using the given Color-Tolerance.

Now you can find out which of multiple Patterns was found for example with

```
WTP.$$PAA|$$PAB|30,40|60
GFP.LastPattern|$$PAT
MBX.$$PAT
```

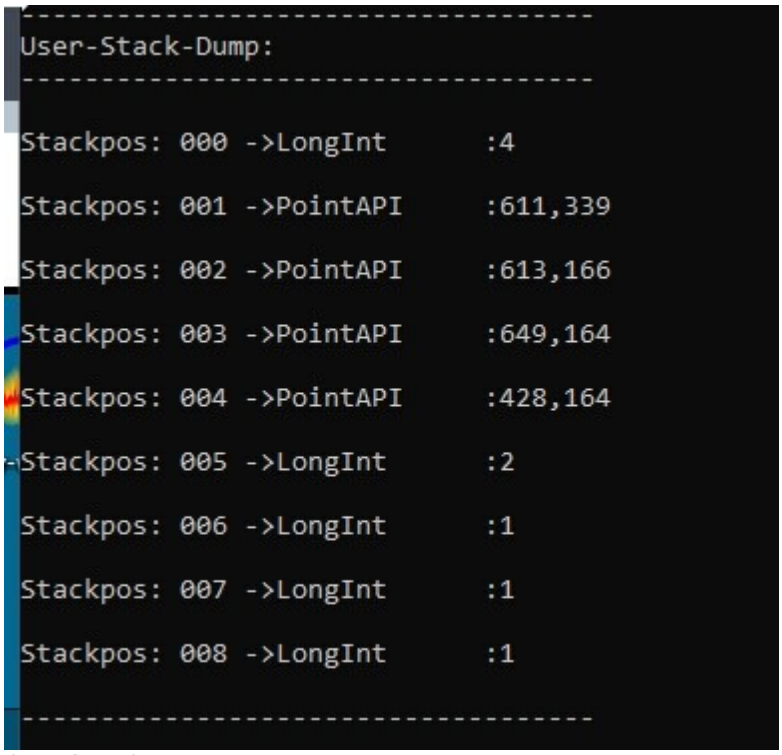
Please see the LTP.-Commands for more Details.

1. Use WTP with 4 Parameters and default settings
This will include all Screens on a Dual-Screen System. Here we search a Pattern on the complete Desktop.

```
STW.ct|Chrome_Widge|Safir - Brav
SWP.0,0|1098,668|T
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
WTP.$$PAA|$$PAB|0,0|60
' You can directly ofter the WTP. Command use MMV. or MLC. etc.
MMV.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
DMP.6
MBX.!
ENR.
```



If you Specify P4, then P4 is the timeout value.
On the Stack you will find all the Results limited by "MaxResults".
On TOS is the "Number of Results".
Followed by the Results.
FInally the Pattern Number that was found as the first result (Here it was Pattern Nr.2 / $$PAB).

2. Use WFP. and set the default Parameters with SFP. before

You can change more advanced Parameters that WFP. uses with the SFP.-Command.

```
STW.ct|Chrome_Widge|Safir - Brav
SWP.0,0|1098,668|T
VAR.$$PAA=?pattern\Py.bmp
VAR.$$PAB=?pattern\Pa.bmp
SFP.ColorTolerance|0
SFP.PixelHits|0
SFP.DebugMode|1
WTP.$$PAA|$$PAB|0,0|60
' You can directly use MMV. or Mouse-Move Command directly after the WTP.
MMV.
' This will show the Stack.
DMP.6
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

WTP.[!]P1|P2[|P3][|P4]

Parameter Explanation

! – If the ! is specified then the Command will wait for the Patterns to disappear, ("Wait for no Pattern")

P1 - Pattern File A - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P2 - Pattern File B - Name and Path
or Specify "IR:" and Image-Register number for Pattern from an Image-Register..
or Specify "ID:$$DTA" und directly use Inline-Data as Pattern.

P3 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.
P4 - opt. Timeout value. If 0 or omitted the command will wait "Forever". If "-1" then the Command will return immediately.

The Command will leave the Pattern-Number of the first find on the TOS, followed by all finds. On the TOS is the number of finds.

The Result "0" means "nothing found", a result of -1 means "Error", while a "1" or a higher number shows the number of found matches.
The maximum Number of found Locations is limited by the "Max-Results" Setting that can be changed using the "SFP.-Command".

The last found Position will automatically set as the "Active Point".

If the Pattern did not appear in the given Time, then you will get a result "0" on TOS and the Timeout-Flag will be set.
The program will **not generate an Error-Message**.

Example

-

Remarks

Do not use ".jpg" Images or images that are compressed with Artefacts. Generally use ".bmp" or ".png" Images. Otherwise the result will be hard to predict.

Limitations:

The resolution of the Pattern-Images that can be used is limited to those resolutions that can be saved with the "Human Eye-Recorder"-Tool.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- WFP. - Wait For Pattern 1903

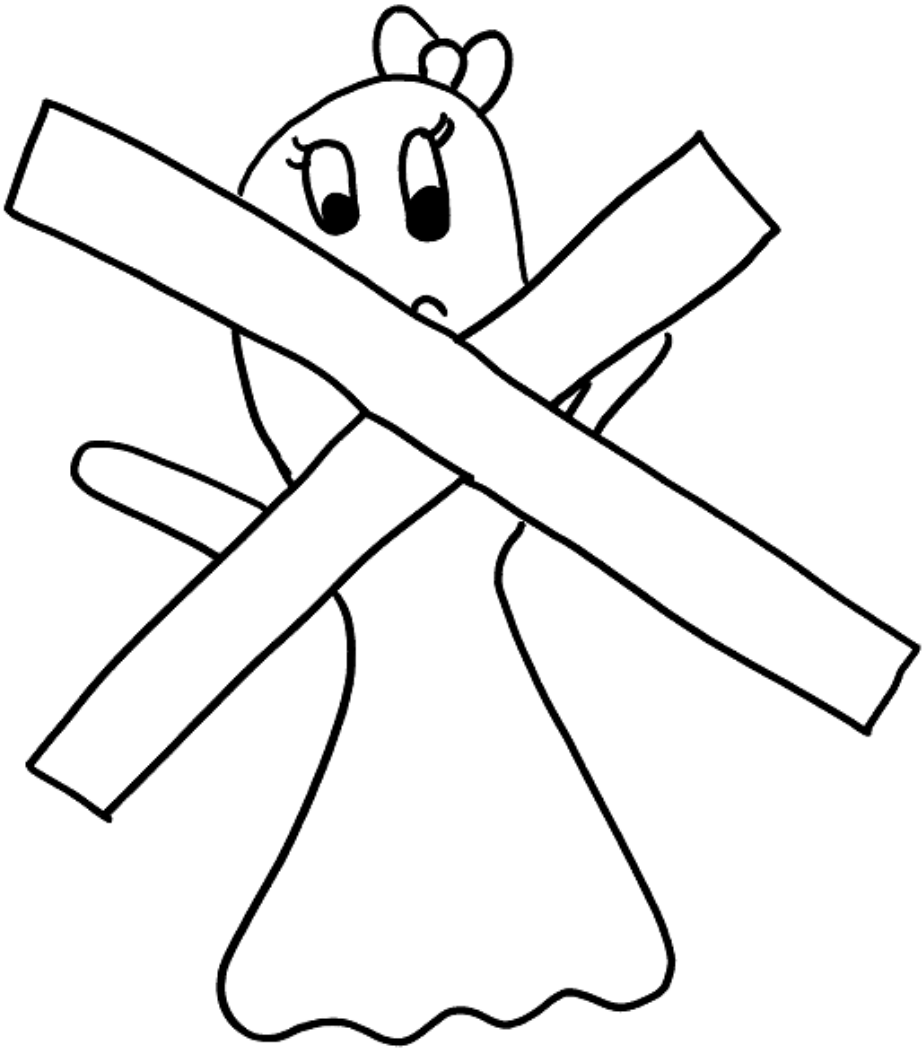
© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8 Undercover and Graphics

3.55.8.1 ! Smarty's Undercover Operations

Smarty's UnderCover Graphics **Previous** 3234 **Top** 41 **Next** 1914

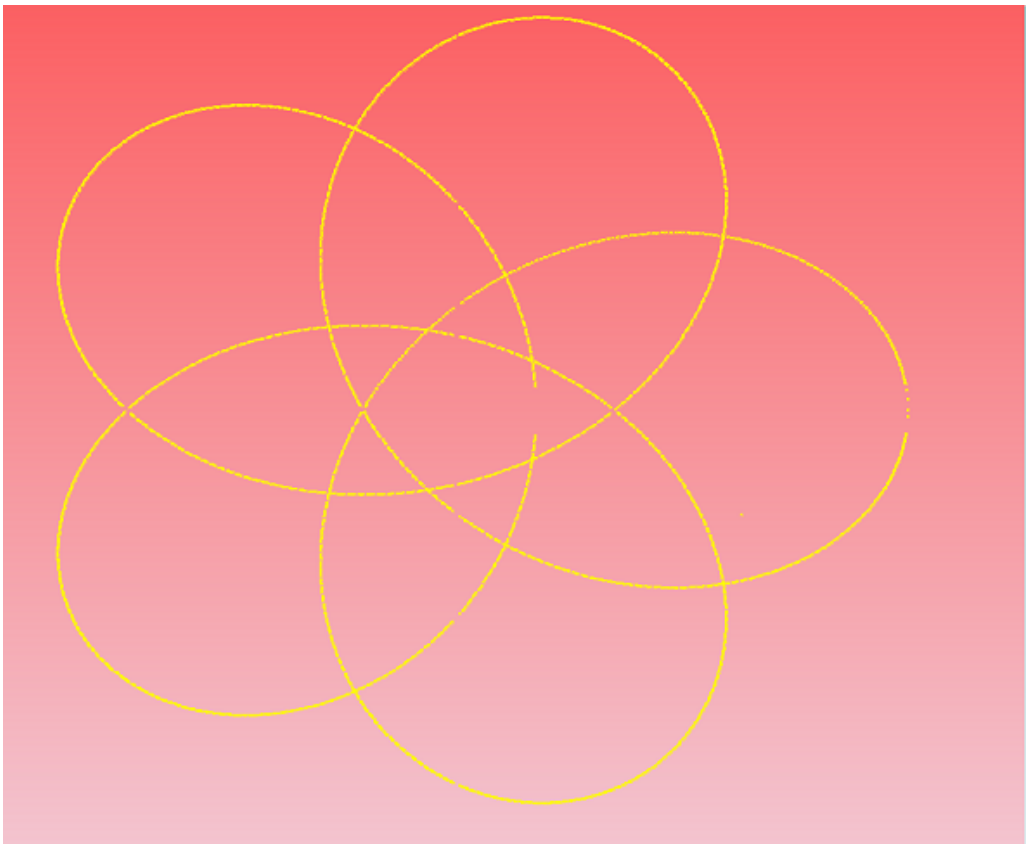
Smarty's Undercover Graphics Operations



Undercover Graphics can be used for these purposes:

1. They can protect your script from being interrupted by a User
2. They can hide your running installations/operations behind a desktop-cover
3. They can show a custom bitmap or graphic to communicate a message to the user
4. The cover "glas" can emulate a "Mouse disable" to some degree, as a user can only click onto the cover and can not disturb the setup process easily.
5. Use them to create/show mathematical figures (see below).

*© 2023 Theo Gottwald * https://www.smart-package.com*



Generally the following features (and more) are supported:

1. "**cover**" - generate a TopMost Form in front of the primary desktop
2. "**no blend**", "**no blend-in**","**no blend-out**" - optional soft fade-in and fade-out transitions
3. "**glas**" - completely invisible glas-form can prevent the user from clicking, while the user
   can see everything that happens below.
4. "**matte**" - Half Transparent cover can prevent the user from clicking, while still allowing to watch
   through it whats going on below.
5. "**desktop-copy**" - Create a desktop-Screen-Shot that is identical to the current desktop to
   completely hide all Operations below (undercover).
6. "**desktop bw**" - Create a desktop screens-hot and change it to grey (no colors).
7. "**cover**" - generate a non-transparent cover and color it with a solid background color of
   your choice
8. "**gradient**" - generate a non-transparent cover and color it with a smooth color transition of your choice
9. "**write**" - Print Text in any size, position and font on to the foreground graphic
10. "**print**" - print listings and scrolling text-messages
11. "**load image**" - load any "**.bmp**" file and resize it to fit the cover-screen.

Technical Note:

Blend-in is done generally always from the desktop. While the cover commands can be combined, it makes sense, to switch off fade-in/fade-out at some point, when combining features. Otherwise you may get unwanted blend-in effects between features.

*© 2023 Theo Gottwald * https://www.smart-package.com*

The Cover will not have any transparency or glas-effects on Windows 2000 or below, as these are not supported below Windows XP.

The Cover needs sufficient graphical resources. On heavily loaded systems with not enough graphical system resources, the cover may just not appear.If you open and close the cover too often, also the cover may just not appear anymore. Both of these issues can quickly be fixed by rebooting the computer.

Important note:
On Error: If your script runs into an Error, then the cover is removed immediately, even before the script has ended. This is to make sure that you can look for the source of the error.
The cover is removed in any error-case, even if the `USM.` (Use silent mode) command is used,

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.2 ! UCV. - Undercover Graphics

MiniRobotLanguage (MRL)

UCV. command

Undercover graphics

Intention

Undercover Graphics can be used for these purposes:

1. They can protect your script from being interrupted by a User
2. They can hide your running installations/operations behind a desktop-cover
3. They can show a custom bitmap or graphic to communicate a message to the user
4. The cover "glas" can emulate a "Mouse disable" to some degree, as a user can only click onto the cover and can not disturb the setup process easily.

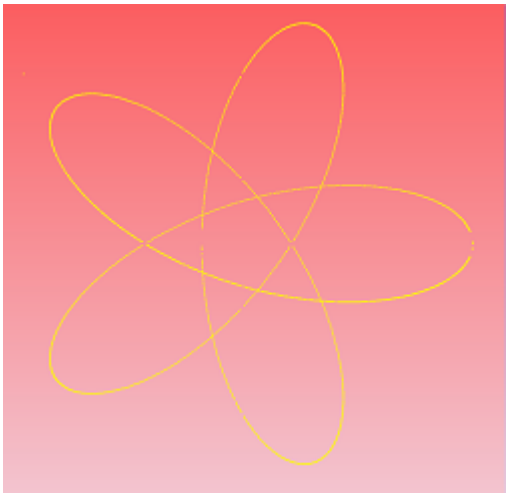
These and more features are supported:

1. "**cover**" - generate a TopMost Form in front of the primary desktop
2. "**no blend**", "**no blend-in**","**no blend-out**" - optional soft fade-in and fade-out transitions
3. "**glas**" - completely invisible galas-form can prevent the user from clicking, while the user can see everything that happens below.
4. "**matte**" - Half Transparent cover can prevent the user from clicking, while still allowing to watch through it whats going on below.
5. "**desktop-copy**" - Create a desktop-Screen-Shot that is identical to the current desktop to completely hide all Operations below (undercover).
6. "**desktop bw**" - Create a desktop screens-hot and change it to grey (no colors).
7. "**cover**" - generate a non-transparent cover and color it with a solid background color of your choice
8. "**gradient**" - generate a non-transparent cover and color it with a smooth color transition of your choice
9. "**write**" - Print Text in any size, position and font on to the foreground graphic
10. "**print**" - print listings and scrolling text-messages
11. "**load image**" - load any "**.bmp**" file and resize it to fit the cover-screen.
12. "**close**". Same like "UCV." without any Subcommand. WIll remove any cover

*© 2023 Theo Gottwald * https://www.smart-package.com*

immediately.
13. "**fillstyle**" - used to define the style (color and pattern) of the fill-operation used for the `UCV.box` and the `UCV.ellipse` command.
14. "**drawto**" - used to paint a line between two points.

As you can see, `UCV.` serves two targets:
1. You can hide your automatic script from user interaction
2. You can use the graphics commands to paint pictures and show them to your audience.



To achieve these goals, Smarty will
- create a "Cover"-Form that will be placed above all other non-TopMost Forms.
- eventually copy a desktop screenshot or anotehr graphic onto that form
- that form will prevent the user from being able to see or click anything that is below that form.*
- you can still use the Direct Engine 231 to work with Setup-Programs.
- the Compatible Engine 231 will not work if a Cover shields the desktop.

For the user this will happen completely transparent, he may not be aware what happens below the cover.

The "desktop-picture cover" is shown with the simple line:

```
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

You can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

You may set up a Cover-Script from multiple `UCV.`-commands (see Parameter Explanation below). Use them in a sequence that will:

- first specify a Modifier,
- then Generate the Cover of your choice
- then customize the Cover

Example:

```
' We use a Modifier to prevent fading
UCV.nb
' Now we start a Cover, a Gradient Cover
UCV.grd|&HFF00FF|&H00FF00
' We wait for 5 Seconds
PAU.5
' We customize the cover printing something on it
' at coordinates 10,300
UCV.write|Please wait!|10,300
PAU.5
' We close the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.[P1][|P2...|Px]

Parameter Explanation

P1 – `(optional)` Subcommand description
.
P2 .. Px – `(optional)` Parameters depending on the used Subcommand.

© 2023 Theo Gottwald * https://www.smart-package.com

```
The following Subcommands are supported:

Cover Modifiiers:
    "no blend","nb"
    "no blend-out","nbo"
    "no blend-in","nbi"
    "transparency","tra"
    "no transparency","ntr"

Generate cover:
    "desktop-copy","dc"
    "desktop bw","dbw"
    "glas"
    "matte","mat"
    "cover","cov"
    "gradient","grd"

Customize Cover:
    "background","bgr"
    "foreground","fgr"
    "print","prt"
    "plot","spo"
    "reset font","rsf"
    "set font","sft"
    "load image","ldi"
    "write","wri"
    "close","clo"
    "box"
    "drawto","line"
    "ellipse","eli"
    "fillstyle","fst"

You may set up a Cover-Script from multiple UCV.-commands.
Use the sequence that you
- first specify a Modifier,
- then Generate the Cover of your choice
- then customize the Cover

to remove a Cover, just use:

UCV.close

or

UCV.
```

Example

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will fade in slowly
' which is invisible if no on going
' changes on the desktop happen.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'***********************************
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.

'***********************************
' UCV.-Sample 2
' In this Mode, the cover will appear immediately.
'
'***********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.


'***********************************
' UCV.-Sample 3
' In this Mode, the cover will appear
' and then be faded to transparent.
'
'***********************************
'
UCV.dc
UCV.tra
PAU.10

' This line will remove the cover
UCV.

'***********************************
' UCV.-Sample 4
' In this Mode, the cover will appear
' without fading.
' Then we will change the background color
' and start printing a larger text.
'
'***********************************
'
$$TXT=Hallo my dear, try the PRINT command also ....
STR.CLONE|$$TXT|6000|$$TXT
UCV.nb
UCV.tra
UCV.cov
UCV.bgr|&HFF00FF
UCV.set font|futura t|40|||0
UCV.pri|$$TXT
PAU.4
UCV.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' UCV.-Sample 5
' Paint mathematical figure
'**********************************
'#SPI:Forcewrite
UCV.nb
UCV.grd|&HFF4545
' Set red foreground color
UCV.fgr|&HFFFF00
FOR.$$F00|0|45|0.002
  $$C00=0.7
  CAX.$$X00=COS(16*$$F00)+$$C00*COS(-4*$$F00)
  CAX.$$Y00=SIN(16*$$F00)+$$C00*SIN(-4*$$F00)
  CAL.$$Y00=$$Y00*260+700
  CAL.$$X00=260*$$X00+700
  IVV.$$Y00>0
    UCV.plot|$$X00,$$Y00
    UCV.drawto|($$X00+2),($$Y00+2)
    UCV.plot|$$X00,$$Y00
    UCV.drawto|($$X00),($$Y00+2)
    UCV.plot|$$X00,$$Y00
    UCV.drawto|($$X00+2),($$Y00
  EIF.
NEX.
PAU.1
ENR.
```

Remarks

You can use the command `CCV.` to make calculations with the RGB-Colors.

Limitations:

The Cover will not have any transparency or glas-effects on Windows 2000 or below, as these are not supported below Windows XP.

The Cover needs sufficient graphical resources. On heavily loaded systems with not enough graphical system resources, the cover may just not appear.If you open and close the cover too often, also the cover may just not appear anymore. Both of these issues can quickly be fixed by rebooting the computer.

Important note:
On Error: If your script runs into an Error, then the cover is removed immediately, even before the script has ended. This is to make sure that you can look for the source of the error.

The cover is removed in any error-case, even if the `USM.` (Use silent mode) command is used,

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's Undercover Operations 1911
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.3 bgr - background

UCV.background - change color of background Previous 1914 Top 41 Next 1925

MiniRobotLanguage (MRL)

UCV.background
UCV.bgr

Change color of cover background

Intention

This command will change the background color of an existing cover and also of an cover that is soon to be created.

General usage is simple:

```
' Make a compeltely green cover
UCV.bgr|&H00FF00
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' This will remove the Cover
UCV.
ENR.
```

You can print a text on the cover, like this:

```
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.
```

As you can see, the "`background`" command will also influence the background of the color of the "`print`" command.

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

WHile "Matte" is a "dark glas", you can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.background
UCV.bgr

Parameter Explanation

P1 - Color- value, for example 16711945 or in hexadezimal: `&HFF00FF` which are both
the same values.

Important notice: You can set the color values also to these three negative values:

-1 -> the DEFAULT foreground/background color is used
-2 -> the background is just not painted, allowing the content behind to become visible
-3 -> the existing color is not changed

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.nb
UCV.bgr|&HFF0000
UCV.cov
PAU.10
UCV.
ENR.


'**********************************
' UCV.-Sample 2
'
'**********************************
'
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.


'**********************************
' UCV.-Sample 3
'
'**********************************
'
UCV.cover
PAU.5
UCV.mat
PAU.5
UCV.
ENR.


'**********************************
' UCV.-Sample 4
' Write a text into the cover'
'**********************************
'
UCV.nb
UCV.cover
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

–

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.4 box

MiniRobotLanguage (MRL)

UCV.box

Draw a box onto the cover

Intention

This command will draw a box, colored and using a custom fillstyle, with optional round corners between two diagonal pixel-coordinates onto the cover. The box-outline is been drawn in the foreground color. The "`UCV.fillstyle`" command defines the fill-style inside the box. As this is a very general graphics command, it will enable you to draw all sorts of shapes that are rectangular or even a circle (make corners 100% round for this).

General usage is simple:

```
' Set foreground color
UCV.fgr|&HFF00FF
' Draw thwe line from 600,1 to 200,100
UCV.line|600,1|200,100
```

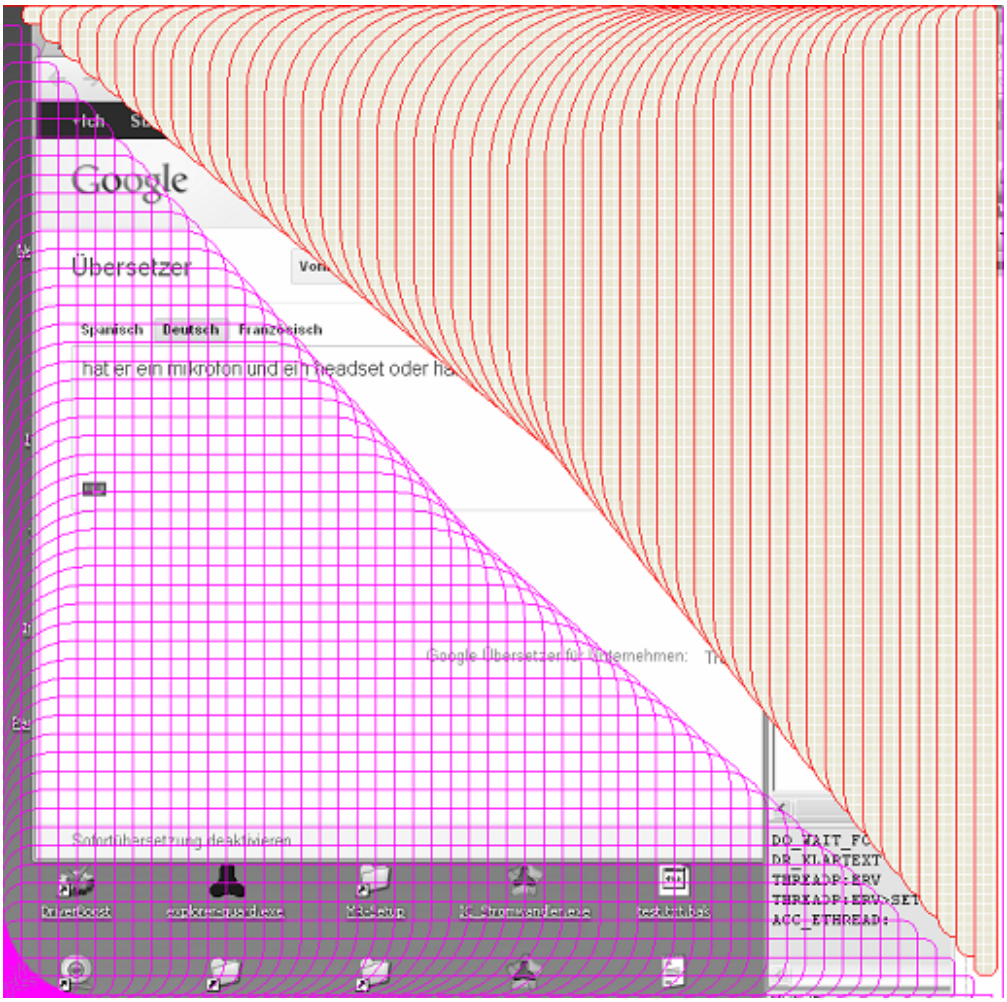
Points that have been set using the "`set point`" command can be used as starting points for the "`Line`" command. In that case you can omit the second coordinate pair.

```
' We set a single Pixel at 1,1 which will then
' be the starting point for the line.
UCV.set pixel|1,1
' you can use mathematical inline calculations
' for all coordinates
UCV.line|($$XCO*2.5),600


' Here is a complete example
'
UCV.nbi
UCV.dbw
FOR.$$COL|0|6
  ' Set foreground color
  UCV.fgr|&HFF0000
  ' Set fillcolor and fillstyle
  UCV.fls|$$COL|&H00FFFF
  ' Make it a circle (100% round corners)
  UCV.box|10,10|500,500|100
  PAU.0.1
NEX.
PAU.1
UCV.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
Here is another example:

$$XMA=800
$$STP=15
UCV.nbi
UCV.dbw

FOR.$$XCO|1|($$XMA+$$STP)|$$STP
  UCV.fgr|&HFF00FF
  UCV.fillstyle|4|-2
  UCV.box|0,$$XCO|$$XCO,$$XMA|40
  UCV.fgr|&HFF0000
  UCV.fillstyle|5|-3
  UCV.box|$$XCO,0|$$XMA,$$XCO|70
NEX.
PAU.5
ENR.
```

This is the output from the above Script:




```
' This line will remove the cover
UCV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

UCV.box|P1,P2[|P3,P4][|P5]

Parameter Explanation

P1,P2 - Screen-Pixel-Coordinates for the starting Point of the line to be drawn.

P3,P4 - (optional) Screen-Pixel-Coordinates for the ending-Point of the line to be drawn.
If omitted, the last used Coordinates are used. This way you can easily draw a figure
of lines.

P5 - (optional) Roundness of corners in percent from 0 to 100%. If you choose 100%, you may end up with a circle.

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
$$XMA=800
$$STP=15
UCV.nbi
UCV.dbw
UCV.fgr|&HFF00FF

FOR.$$XCO|1|($$XMA+$$STP)|$$STP
  UCV.fgr|&HFF00FF
  UCV.fillstyle|4|-2
  UCV.box|0,$$XCO|$$XCO,$$XMA|40
  UCV.fgr|&HFF0000
  UCV.fillstyle|5|-3
  UCV.box|$$XCO,0|$$XMA,$$XCO|70
NEX.
PAU.5
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.5 cov - cover

MiniRobotLanguage (MRL)

UCV.cover
UCV.cov

Create a unicolor, opaque and protective cover

Intention

This command will create a protective cover between the user and the desktop.
You can use it to prevent the user from seeing serial numbers, and you can prevent the user from interacting with your setup-processes.

By default the cover has a windows-button color, that could be a sort of grey.
You can however change the cover-color using the "`UCV.background`" command.

To achieve this goal, Smarty will
- create a "Cover"-Form that will be placed above all other Forms*.
- make that form half transparent and add a grey desktop picture to it.
- that form will prevent the user from being able to see or click anything that is below that form.*
- you can still use the Direct Engine 231 to work with Setup-Programs.
- the Compatible Engine 231 will not work if a Cover shields the desktop.

General usage is simple:

```
' Make a compeltely green cover
UCV.bgr|&H00FF00
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' This will remove the Cover
UCV.
ENR.
```

You can print a text on the cover, like this:

```
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.
ENR.
```

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

WHile "Matte" is a "dark glas", you can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.cover
UCV.cov

Parameter Explanation

-

Example

```
'*********************************
' UCV.-Sample
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'***********************************
'
UCV.nb
UCV.bgr|&HFF0000
UCV.cov
PAU.10
UCV.
ENR.


'***********************************
' UCV.-Sample 2
'
'***********************************
'
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.


'***********************************
' UCV.-Sample 3
'
'***********************************
'
UCV.cover
PAU.5
UCV.mat
PAU.5
UCV.
ENR.


'***********************************
' UCV.-Sample 4
' Write a text into the cover'
'***********************************
'
UCV.nb
UCV.cover
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.6 dbw - desktop bw

UCV.desktop bw - greyscaled desktop cover **Previous** 1929 **Top** 41 **Next** 1937

MiniRobotLanguage (MRL)

UCV.desktop bw
UCV.dbw

Create a greyscaled desktop cover (fade to grey)

Intention

This command is used to prevent the user from seeing what happens on the screen. You can use it to protect serial numbers or other confidential information during automated setup processes. As the Cover is grayscaled, the user will be aware that something happens.

To achieve this goal, Smarty will
- create a "Cover"-Form that will be placed above all other Forms*.
- copy a desktop screenshot onto that form
- that form will prevent the user from being able to see or click anything that is below that form.*
- you can still use the Direct Engine 231 to work with Setup-Programs.
- the Compatible Engine 231 will not work if a Cover shields the desktop.
- Smarty will slowly fade the cover colors to grey, depending on the use of the blending commands (nb,nbi,nbo)

For the user this will happen completely transparent, he can not see what happens below the cover.

The greyscaled "desktop-picture cover" is shown with the simple line:

```
UCV.dbw
PAU.10

' This line will remove the cover
UCV.
```

If you do not want to use the "fade to grey" effect, use this script:

```
UCV.nb
UCV.dbw
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```

Using the "`write`" command, you can write a text at any place into the desktop-cover. You can also change/choose the font for this text using the "`set font`" command.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
```

You can also use the "print" command while this makes less sense, because the print command has a fixed background.

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

You can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.desktop bw
UCV.dbw

Parameter Explanation

-

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will fade in slowly
' which is invisible if no on going
' changes on the desktop happen.
'**********************************
' Switch cover on
UCV.dbw
PAU.20
UCV.
ENR.

'**********************************
' UCV.-Sample 2
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.dbw
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample 3
' In this Mode, the cover will appear
' and then be faded to transparent.
'
'**********************************
'
UCV.dbw
UCV.tra
PAU.10

' This line will remove the cover
UCV.

'**********************************
' UCV.-Sample 4
' Write a text into the cover'
'**********************************
'
UCV.nb
UCV.dbw
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

–

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.7 dc - desktop-copy

UCV.desktop-copy - desktop screenshot on cover **Previous** 1933 **Top** 41 **Next** 1941

MiniRobotLanguage (MRL)

UCV.desktop-copy
UCV.dc

Copy desktop screenshot on TopMost Cover

Intention

This command is used to prevent the user from seeing what happens on the screen. You can use it to protect serial numbers or other confidential information during automated setup processes.

To achieve this goal, Smarty will
- create a "Cover"-Form that will be placed above all other Forms*.
- copy a desktop screenshot onto that form
- that form will prevent the user from being able to see or click anything that is below that form.*
- you can still use the Direct Engine 231 to work with Setup-Programs.
- the Compatible Engine 231 will not work if a Cover shields the desktop.

For the user this will happen completely transparent, he may not be aware what happens below the cover.

The "desktop-picture cover" is shown with the simple line:

```
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

Using the "`write`" command, you can write a text at any place into the desktop-cover. You can also change/choose the font for this text using the "`set font`" command.

```
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
```

You can also use the "print" command while this makes less sense, because the print command has a fixed background.

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

You can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.desktop-copy
UCV.dc

Parameter Explanation

-

Example

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will fade in slowly
' which is invisible if no on going
' changes on the desktop happen.
'**********************************
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.

'**********************************
' UCV.-Sample 2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample 3
' In this Mode, the cover will appear
' and then be faded to transparent.
'
'**********************************
'
UCV.dc
UCV.tra
PAU.10

' This line will remove the cover
UCV.

'**********************************
' UCV.-Sample 4
'**********************************
UCV.nb
UCV.dc
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911

© 2023 Theo Gottwald * https://www.smart-package.com

- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.8 eli - ellipse

MiniRobotLanguage (MRL)

UCV.ellipse
UCV.eli

Draw a box onto the cover

Intention

This command will draw a elliptical, colored, filled shape, between two diagonal pixel-coordinates onto the cover. The shape-outline is been drawn in the foreground color. The "`UCV.fillstyle`" command defines the fill-style inside the shape. As this is a very general graphics command, it will enable you to draw all sorts of shapes that are circular or elliptical. If you want to end up with a perfect circle, prefer the "`UCV.box`" command and make the corners 100% round for this.

General usage is simple:

```
' Set foreground color
UCV.fgr|&HFF00FF
' Draw a black filled ellisoid
UCV.ellipse|600,1|200,100
```

Points that have been defined using the "`set point`" command can be used as starting points for the "`Line`" command. In that case you can omit the second coordinate pair.

```
' We set a single Pixel at 1,1 which will then
' be the starting point for the ellipse.
UCV.nbi
UCV.dbw
UCV.fgr|&HFF00FF
UCV.set pixel|100,100
UCV.ellipse||200,100
PAU.5
```

Here is a more complex demo program:

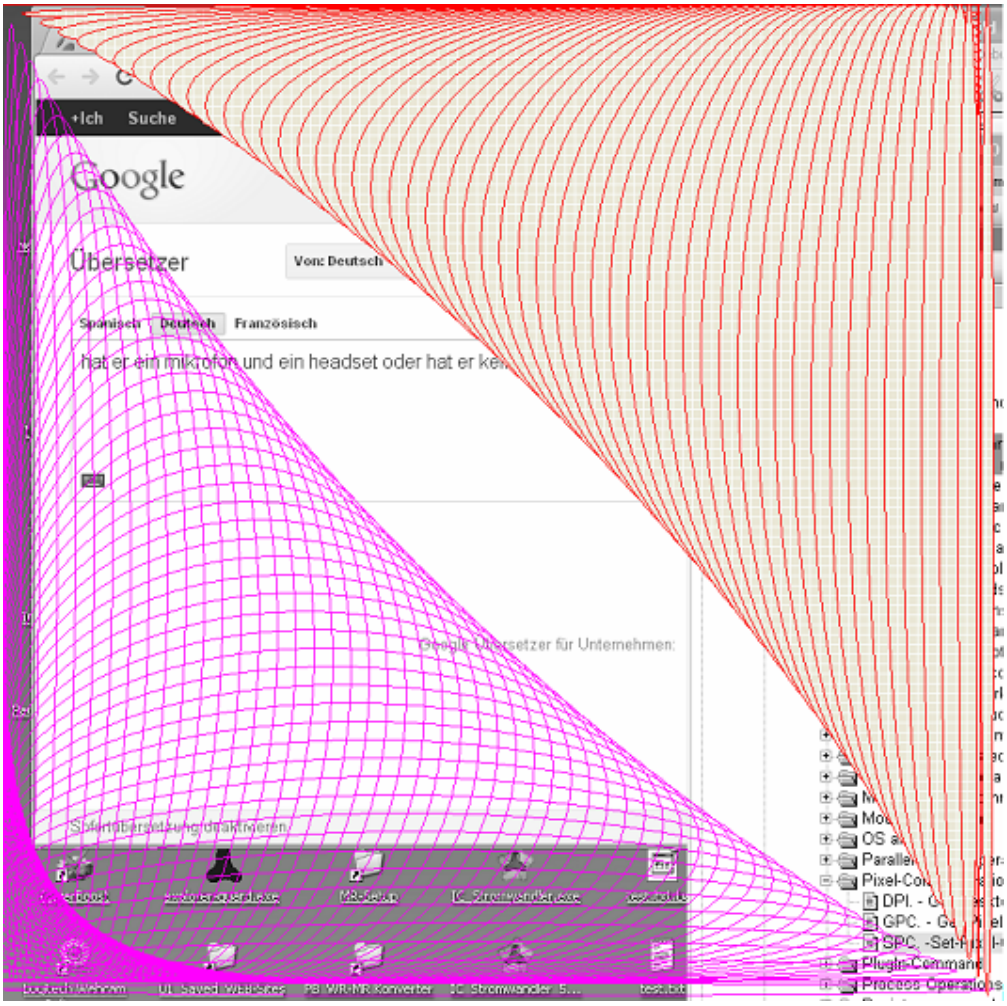
```
$$XMA=800
$$STP=15
UCV.nbi
UCV.dbw

FOR.$$XCO|1|($$XMA+$$STP)|$$STP
  UCV.fgr|&HFF00FF
  UCV.fillstyle|4|-2
  UCV.eli|0,$$XCO|$$XCO,$$XMA|40
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  UCV.fgr|&HFF0000
  UCV.fillstyle|5|-3
  UCV.eli|$$XCO,0|$$XMA,$$XCO|70
NEX.
PAU.5
ENR.
```

Below you see the output graphics. Note the effects of the

```
' Draw transparent shapes
UCV.fillstyle|4|-2
```




```
' Here is an example that will draw circular shapes
'
UCV.nbi
UCV.dbw
FOR.$$COL|0|6
  ' Set foreground color
  UCV.fgr|&HFF0000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  ' Set fillcolor and fillstyle
  UCV.fls|$$COL|&H00FFFF
  ' Make it a circle (100% round corners)
  UCV.eli|10,10|500,500
  PAU.0.1
NEX.
PAU.1
UCV.
ENR.

' This line will remove the cover
UCV.
```

Syntax

UCV.ellipse|P1,P2[|P3,P4]

Parameter Explanation

P1,P2 - Screen-Pixel-Coordinates for the starting Point of the line to be drawn.

P3,P4 - (optional) Screen-Pixel-Coordinates for the ending-Point of the line to be drawn.
If omitted, the last used Coordinates are used. This way you can easily draw a figure
of lines.

P5 - (optional) Roundness of corners in percent from 0 to 100%. If you choose 100%, you may end up with a circle.

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
$$XMA=800
$$STP=15
UCV.nbi
UCV.dbw
UCV.fgr|&HFF00FF

FOR.$$XCO|1|($$XMA+$$STP)|$$STP
  UCV.fgr|&HFF00FF
  UCV.fillstyle|4|-2
  UCV.box|0,$$XCO|$$XCO,$$XMA|40
  UCV.fgr|&HFF0000
  UCV.fillstyle|5|-3
  UCV.box|$$XCO,0|$$XMA,$$XCO|70
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEX.
PAU.5
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.9 fgr - foreground

UCV.foreground - change color of foreground **Previous** 1941 **Top** 41 **Next** 1949

MiniRobotLanguage (MRL)

UCV.foreground
UCV.fgr

Change color of foreground Print/Pixel/Paint commands

Intention

This command will change the foreground color of any soon to be used graphic command that will draw onto the cover (print/write/pixel etc.).

General usage is simple:

```
' Make a completely green cover
UCV.bgr|&H00FF00
' Set red foreground color
UCV.fgr|&HFF0000
UCV.cov
' Print in red onto the green background
UCV.print|Hello world!
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&HFFFF00
UCV.print|Hello world!
' This will remove the Cover
UCV.
ENR.
```

You can print a text on the cover, like this:

```
' Make a completely red cover
UCV.bgr|&HFF0000
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&HFFFF00
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.
```

As you can see, the "`background`" command will also influence the background of the color of the "`print`" command.

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

*© 2023 Theo Gottwald * https://www.smart-package.com*

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

WHile "Matte" is a "dark glas", you can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.foreground|P1
UCV.fgr|P1

Parameter Explanation

P1 - Color- value, for example 16711945 or in hexadezimal: `&HFF00FF` which are both
the same values.

Important notice: You can set the color values also to these three negative values:

-1 -> the DEFAULT foreground/background color is used
-2 -> the background is just not painted, allowing the content behind to become visible
-3 -> the existing color is not changed

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' UCV.-Sample
'
'**********************************
'
UCV.tra
UCV.nb
UCV.dc
UCV.fgr|&HFF00FF
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.

'**********************************
' UCV.-Sample 2
'
'**********************************
'
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.


'**********************************
' UCV.-Sample 3
'
'**********************************
'
UCV.cover
PAU.5
UCV.mat
PAU.5
UCV.
ENR.


'**********************************
' UCV.-Sample 4
' Write a text into the cover'
'**********************************
'
UCV.nb
UCV.cover
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.10 glas - suppress mouse and keyboard

UCV.glas - put a glas-screen betwe Previous 1945 Top 41 Next 1953

MiniRobotLanguage (MRL)

UCV.glas

Create a completely invisible, yet protective cover

Intention

This command is used to allow the user to see what happens on screen. While its still protective to the scripted process and prevents the user from clicking or doing any action on the protected process. Its just like if you put a glas between the desktop and the user-mouse

To achieve this goal, Smarty will
- create a "Cover"-Form that will be placed above all other Forms*.
- make that form completely transparent
- that form will prevent the user from being able to see or click anything that is below that form.*
- you can still use the Direct Engine 231 to work with Setup-Programs.
- the Compatible Engine 231 will not work if a Cover shields the desktop.

General usage is simple:

```
UCV.glas
' Here comes your script, we use a PAUSE command instead
PAU.10
' This will remove the Cover
UCV.
ENR.
```

You can switch from "matte" (50% transparent/grey glas) to "glas", like this:

```
UCV.mat
PAU.5
UCV.glas
PAU.5
UCV.
ENR.
```

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

You can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.glas

Parameter Explanation

-

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
'
UCV.glas
PAU.10
UCV.
ENR.


'**********************************
' UCV.-Sample 2
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.dbw
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample 3
' In this Mode, the cover will appear
' and then be faded to transparent.
'
'**********************************
'
UCV.dbw
UCV.tra
PAU.10

' This line will remove the cover
UCV.


'**********************************
' UCV.-Sample 4
' Write a text into the cover'
'**********************************
'
UCV.nb
UCV.dbw
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937

© 2023 Theo Gottwald * https://www.smart-package.com

- CCV. - Color Conversion 11117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.11 grd - gradient

UCV.gradient - fade between two colors **Previous** 1949 **Top** 41 **Next** 1958

MiniRobotLanguage (MRL)

UCV.gradient
UCV.grd

Fade between two colors on a cover

Intention

This command will change the background color of an existing cover or create an cover itself. It will paint the cover with an fading effect between two colors.
The effect can be horizontal or vertical.

Here is a simple example:

```
UCV.grd
PAU.5
ENR.
```

Below is the result, it uses the default gradient-colors.

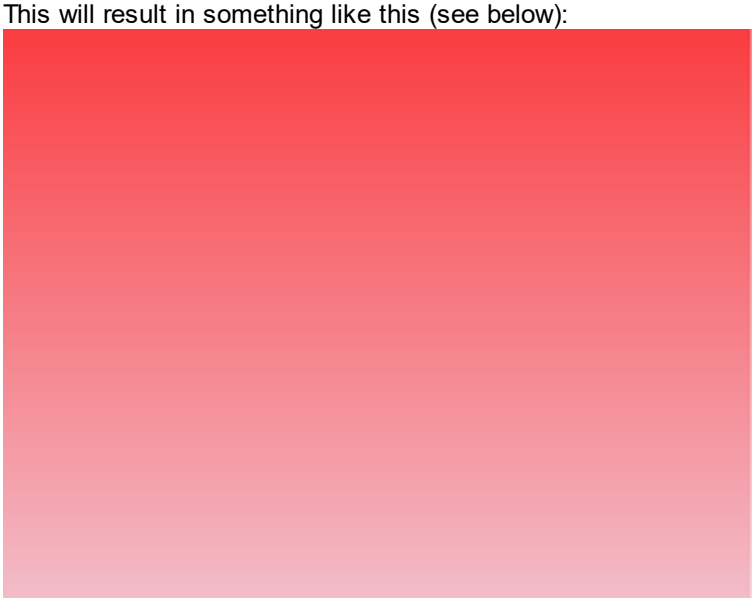
Next we will use a custom gradient color. Like this:

```
UCV.nbi
UCV.grd|&HFF0000
PAU.5
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
ENR.
```

This will result in something like this (see below):



Now we will specify a second gradient color. Lets say green:

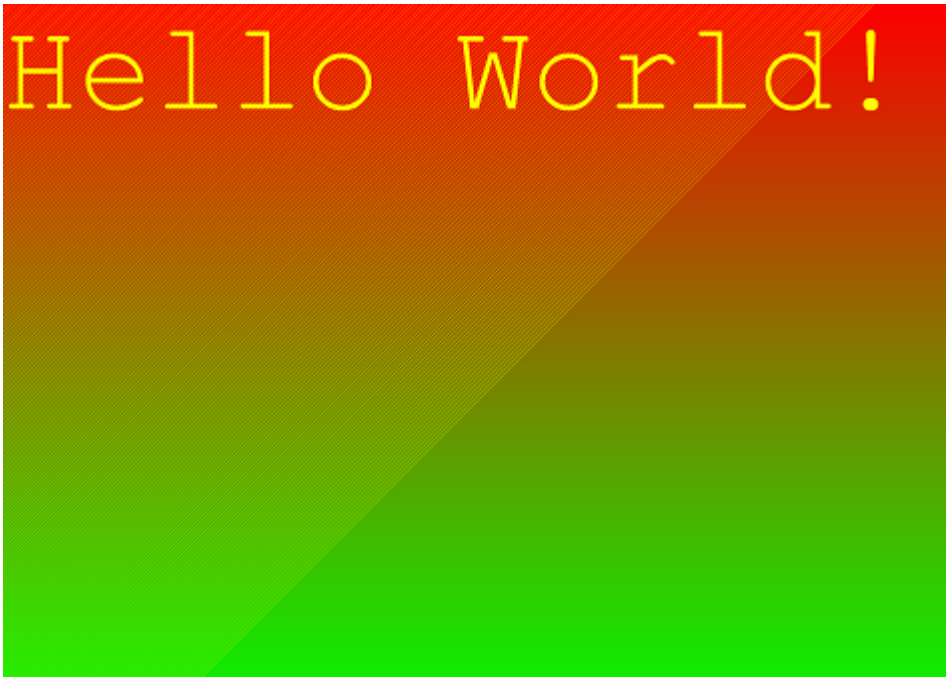
```
UCV.nbi
' red/green fading gradient
UCV.grd|&HFF0000|&H00FF00
PAU.5
ENR.
```

The following example shows that you can also write and paint on a gradient.

```
UCV.nbi
UCV.grd|&HFF0000|&H00FF00
UCV.fgr|&HFFFF00
UCV.write|Hello World!
FOR.$$XCO|1|2600|10
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  UCV.drawto|1,$$XCO|2*$$XCO,$$XCO*2
  UCV.drawto|$$XCO,1|1,$$XCO
NEX.
PAU.5
ENR.
```



If you specify a "1" as third number:

```
UCV.grd|&HFF0000|&H00FF00|1
```

then the gradient will be horizontal instead of vertical.

```
UCV.nbi
UCV.grd|&HFF0000|&H00FF00|1
UCV.fgr|&HFFFF00
UCV.write|Hello World!
FOR.$$XCO|1|2600|10
  UCV.drawto|1,$$XCO|2*$$XCO,$$XCO*2
  UCV.drawto|$$XCO,1|1,$$XCO
NEX.
PAU.5
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

UCV.gradient[|P1][|P2][|P3]
UCV.grd[|P1][|P2][|P3]

Parameter Explanation

P1 - (optional) Color-value for first gradient color.

P2 - (optional) Color-value for second gradient color.

P3 - (optional) 0/1 - switches between a horizontal and a vertical gradient.

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
'
UCV.nbi
UCV.grd|&HFF0000|&H00FF00
PAU.10
UCV.
ENR.


'**********************************
' UCV.-Sample 2
'
'**********************************
'
UCV.nbi
UCV.grd|&HFF0000|&H00FF00
UCV.fgr|&HFFFF00
UCV.write|Hello World!
FOR.$$XCO|1|2600|10
  UCV.drawto|1,$$XCO|2*$$XCO,$$XCO*2
  UCV.drawto|$$XCO,1|1,$$XCO
NEX.
PAU.5
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

_

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.12 ldi - load image

UCV.load image - load a BMP-Pictu Previous 1953 Top 41 Next 1962

MiniRobotLanguage (MRL)

UCV.load image
UCV.ldi

Load a BMP-Picture as cover background

Intention

This command will load an bitmap file (BMP-file) onto the cover. The file will automatically be resized to fit the cover.

General usage is simple, if the picture is in `?path\`, that is the folder, where the script itself is, you can just give the filename. Otherwise specify a complete path..

If you want to compile an executable, note that you have to include the picture file manually into the excutable using the

```
'#INC:?path\picture.bmp
```

command. There is currently no auto-include for images.

```
' this will load an image and show it with a fade-in effect
UCV.ldi|rubin.bmp
PAU.5
```

© 2023 Theo Gottwald * https://www.smart-package.com

You can also use the image as background and modify it using the `UCV.write` or other `UCV.`-commands.

```
UCV.nbi
UCV.ldi|rubin.bmp
' Paint on the picture
UCV.fgr|&HFF0000
FOR.$$XCO|1|2600|6
  UCV.drawto|1,$$XCO|2*$$XCO,$$XCO*2
  UCV.drawto|$$XCO,1|1,$$XCO
NEX.
PAU.5
ENR.
```

Syntax

UCV.load image
UCV.ldi

Parameter Explanation

P1 - variable/filename of the image that should be loaded. The image will automatically be re-sized to fit the entire cover.

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
'
UCV.nb
UCV.bgr|&HFF0000
UCV.cov
PAU.10
UCV.
ENR.


'**********************************
' UCV.-Sample 2
'
'**********************************
'
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.


'**********************************
' UCV.-Sample 3
'
'**********************************
'
UCV.cover
PAU.5
UCV.mat
PAU.5
UCV.
ENR.


'**********************************
' UCV.-Sample 4
' Write a text into the cover'
'**********************************
'
UCV.nb
UCV.cover
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.13 lin - line / drawto

MiniRobotLanguage (MRL)

UCV.line
UCV.lin
UCV.drawto

Draw a line onto the cover

Intention

This command will draw a line between two pixel-coordinates onto the cover. The line is been drawn in the foreground color. As this is a very general graphics command, it will enable you to draw all sorts of shapes.

General usage is simple:

```
' Set foreground color
UCV.fgr|&HFF00FF
' Draw thwe line from 600,1 to 200,100
UCV.line|600,1|200,100
```

Points that have been set using the "`set point`" command can be used as starting points for the "`Line`" command. In that case you can omit the second coordinate pair.

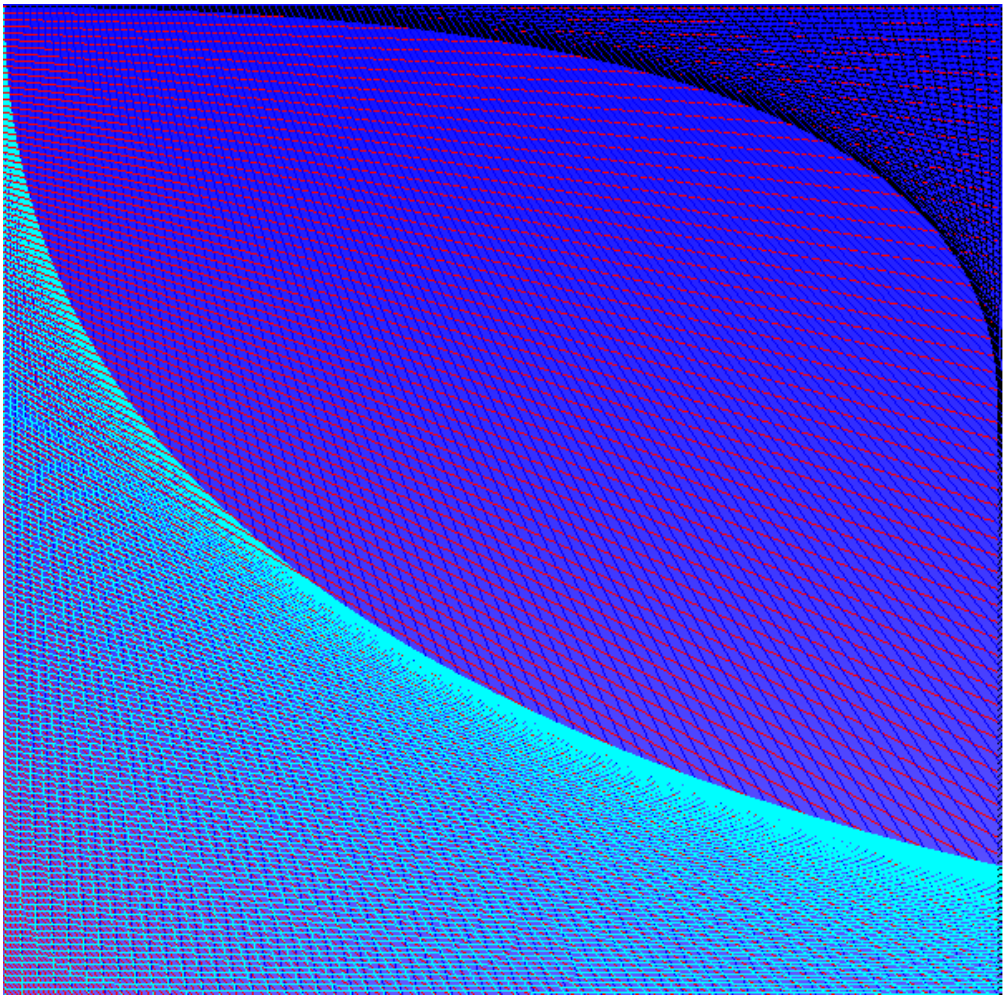
```
' We set a single Pixel at 1,1 which will then
' be the starting point for the line.
UCV.set pixel|1,1
' you can use mathematical inline calculations
' for all coordinates
UCV.line|($$XCO*2.5),600


' Here is a complete example using a gradient as background.
UCV.nb
UCV.gradient
FOR.$$XCO|1|600
  UCV.fgr|&HFF00FF
  UCV.line|600,$$XCO|($$XCO*2.5),1
NEX.
PAU.4
UCV.
ENR.ENR.
```

You can draw more complex example, like this:

```
' compile before running
UCV.nb
UCV.gradient
FOR.$$XCO|1|600
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  UCV.fgr|&H0000FF
  UCV.set pixel|$$XCO,1
  UCV.line|($$XCO*2.5),600
  UCV.fgr|&HFF0000
  UCV.line|1,$$XCO|600,($$XCO*2.5)
  UCV.fgr|&H00FFFF
  UCV.line|($$XCO*2.5),600|1,$$XCO
  UCV.fgr|&HFFFF
  UCV.line|600,$$XCO|($$XCO*2.5),1
NEX.
PAU.4
UCV.
ENR.
```



Syntax

UCV.line|P1,P2[|P3,P4]
UCV.lin|P1,P2[|P3,P4]
UCV.drawto|P1,P2

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1,P2 - Screen-Pixel-Coordinates for the starting Point of the line to be drawn.

P3,P4 - (optional) Screen-Pixel-Coordinates for the ending-Point of the line to be drawn.

If omitted, the last used Coordinates are used. This way you can easily draw a figure

of lines.

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
UCV.nbi
UCV.dbw
FOR.$$XCO|1|1200|4
  UCV.fgr|&HFF00FF
  UCV.line|1200,$$XCO|($$XCO*2.5),1
NEX.
PAU.4
UCV.
ENR.

'**********************************
' UCV.-Sample 2
' use Lines in random colors
'**********************************
'
UCV.nbi
UCV.dbw

FOR.$$XCO|1|1200|4
  UCV.fgr|&HFF00FF
  UCV.line|1200,$$XCO|($$XCO*2.5),1
  UCV.fgr|&HFFFF00
' Generate random colors
  RND.120|255|$$RED
  RND.120|255|$$GRN
  RND.120|255|$$BLU
  CAL.$$COL=$$RED+256*$$GRN+(256*256)*$$BLU
  UCV.fgr|$$COL
'-------------------------
  UCV.line|1200,($$XCO*2.5)
  UCV.line|($$XCO*2.5),1200
  UCV.line|($$XCO*2.5),1|($$XCO*2.5),1200
NEX.
PAU.4
UCV.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' UCV.-Sample 2
' Mathematical figure in random colors
'**********************************
'
$$XCE=800
$$YCE=500
UCV.nb
UCV.cov
UCV.bgr|&HFFFFFF
UCV.fgr|&H0000FF
UCV.plot|$$XCE,$$YCE
FOR.$$VAA|0|($$XCE*2)
  GSB.rndcol
  CAX.$$XCO=$$XCE+300*SIN($$VAA*3.14/180)
  CAX.$$YCO=$$YCE+300*COS($$VAA*3.15/180)
  UCV.drawto|$$XCO,$$YCO
NEX.
PAU.5

ENR.

:rndcol
RND.120|255|$$RED
RND.120|255|$$GRN
RND.120|255|$$BLU
CAL.$$COL=$$RED+256*$$GRN+(256*256)*$$BLU
UCV.fgr|$$COL
RET.

'**********************************
' UCV.-Sample 2
' Mathematical figure
'**********************************
$$XMA=800
$$STP=15
UCV.nbi
UCV.dbw
UCV.fgr|&HFF00FF

FOR.$$XCO|1|($$XMA+$$STP)|$$STP
  UCV.drawto|0,$$XCO|$$XCO,$$XMA
  UCV.drawto|$$XCO,0|$$XMA,$$XCO
NEX.
PAU.5
ENR.
```

Remarks

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.14 mat - matte, dark transparent cover

UCV.matte - put a dark glas-screen **Previous** 1962 **Top** 41 **Next** 1971

MiniRobotLanguage (MRL)

UCV.matte
UCV.mat

Create a dark, transparent yet protective cover

Intention

This command is used to allow the user to see what happens on screen. While its still protective to the scripted process and prevents the user from clicking or doing any action on the protected process. Its just like if you put a dark, half transparent glas between the desktop and the user-mouse.

To achieve this goal, Smarty will
- create a "Cover"-Form that will be placed above all other Forms*.
- make that form half transparent and add a grey desktop picture to it.
- that form will prevent the user from being able to see or click anything that is below that form.*
- you can still use the Direct Engine 231 to work with Setup-Programs.
- the Compatible Engine 231 will not work if a Cover shields the desktop.

General usage is simple:

```
UCV.matte
' Here comes your script, we use a PAUSE command instead
PAU.10
' This will remove the Cover
UCV.
ENR.
```

You can switch from glas to "matte" (50% transparent/grey glas), like this:

```
UCV.glas
PAU.5
UCV.mat
PAU.5
UCV.
ENR.
```

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

WHile "Matte" is a "dark glas", you can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.matte
UCV.mat

Parameter Explanation

-

Example

```
'***********************************
' UCV.-Sample
'
'***********************************
'
UCV.matte
PAU.10
UCV.
ENR.


'***********************************
' UCV.-Sample 2
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'***********************************
'
UCV.mat
PAU.5
UCV.glas
PAU.5
UCV.
ENR.


'***********************************
' UCV.-Sample 3
'
'***********************************
'
UCV.glas
PAU.5
UCV.mat
PAU.5
UCV.
ENR.


'***********************************
' UCV.-Sample 4
' Write a text into the cover'
'***********************************
'
UCV.nb
UCV.dbw
UCV.set font|futura t|100|||0
UCV.write|Hallo world!|10,200
PAU.3
' This line will remove the cover
PAU.5
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.15 nb - no blend

MiniRobotLanguage (MRL)

UCV.no blend
UCV.nb

Change fading behavior of the cover

Intention

This command is used to change the fading behavior for an cover. Without this command, a cover will slowly fade-in, and also slowly fade-out. If this command is use, the fading is completely disabled..

The "desktop-picture cover" is shown immediately without fading like this:

```
UCV.nb
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

Syntax

UCV.no blend
UCV.nb

Parameter Explanation

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample
' We will make the existing cover to disappear
' using a transparency value of 0.
' Then we print on the cover and just
' make it visible again.
'
'**********************************
'
UCV.nb
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

Remarks

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911

© 2023 Theo Gottwald * https://www.smart-package.com

- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.16 nbi - no blend-in

UCV.no blend-in - disable fading in Previous 1971 Top 41 Next 1977

MiniRobotLanguage (MRL)

UCV.no blend-in
UCV.nbi

Change fading behavior of the appearing cover

Intention

This command is used to change the fading behavior for an cover when it appears. Without this command, a cover will slowly fade-in. If this command is use, the fading-in is completely disabled.

The "desktop-picture cover" is shown immediately without fading like this:

```
UCV.nbi
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

Syntax

UCV.no blend-in
UCV.nbi

Parameter Explanation

-

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample
' We will make the existing cover to disappear
' using a transparency value of 0.
' Then we print on the cover and just
' make it visible again.
'
'**********************************
'
UCV.nbi
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

Remarks

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911

© 2023 Theo Gottwald * https://www.smart-package.com

- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.17 nbo - no blend-out

UCV.no blend-out - disable fading out Previous 1974 Top 41 Next 1980

MiniRobotLanguage (MRL)

UCV.no blend-out
UCV.nbo

Change fading behavior of the disappearing cover

Intention

This command is used to change the fading behavior for an cover when it disappears. Without this command, a cover will slowly fade-out. If this command is used, the fading-out is completely disabled.

The "desktop-picture cover" is shown without fade-in and then disappear without fade-out:

```
UCV.dc
PAU.10

' This line will remove the cover
UCV.nbo
UCV.
```

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

Syntax

UCV.no blend-in
UCV.nbi

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

Example

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-out
UCV.no blend-out
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample
' We will make the existing cover to disappear
' using a transparency value of 0.
' Then we print on the cover and just
' make it visible again.
'
'**********************************
'
UCV.nbi
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

Remarks

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.18 ntr - no transparency

UCV.no transparency - make cover opaque Previous 1977 Top 41 Next 1984

MiniRobotLanguage (MRL)

UCV.no transparency
UCV.ntr

Change transparency value of Cover to opaque

Intention

This command is used to change the transparency value of the screen cover to opaque.

This command is a modifier command. It is used to modify an existing cover, or to modify an cover that is soon to be created.

Usage is simple:

```
' Tell the cover to start transparent (50%)
UCV.tra

' Choose a greyscale desktop picture cover
UCV.dbw

' Wait 5 seconds
PAU.5

' No make the cover opaque
UCV.ntr

' Wait ten seconds
PAU.10

' close the cover
UCV.

' end the script
ENR.
```

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```


© 2023 Theo Gottwald * https://www.smart-package.com

You can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

A transparency value of 0 (zero) which is used after the cover has been created, will deactivate the cover while its still there. You can switch it on later just by giving another transparency value. Here is an example:

```
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.no transparency
UCV.ntr

Parameter Explanation

-

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
UCV.tra
UCV.dbw
PAU.5
UCV.ntr
PAU.10
UCV.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' UCV.-Sample 2
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.


'**********************************
' UCV.-Sample 3
' In this Mode, the cover will appear
' and then be faded to transparent.
'
'**********************************
'
UCV.dc
UCV.tra
PAU.10

' This line will remove the cover
UCV.

'**********************************
' UCV.-Sample 4
' We will make the existing cover to disappear
' using a transparency value of 0.
' Then we print on the cover and just
' make it visible again.
'
'**********************************
'
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

Remarks

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.19 pri - print

UCV.print - print listing Previous 1980 Top 41 Next 1987

MiniRobotLanguage (MRL)

UCV.print
UCV.pri

Print more text use vertical scrolling if needed

Intention

This command will print text in the actual foreground color. Use the "`UCV.foreground`" command to set the foreground color. "`UCV.print`" will use the font that you can set using the "`UCV.set font`" command.

General usage is simple:

```
' Make a completely green cover
UCV.bgr|&H00FF00
' Set red foreground color
UCV.fgr|&HFF0000
UCV.cov
' Print in red onto the green background
UCV.print|Hello world!
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&HFFFF00
UCV.print|Hello world!
' This will remove the Cover
UCV.
ENR.
```

You can print a text on the cover, like this:

```
' Make a completely red cover
UCV.bgr|&HFF0000
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&HFFFF00
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.
```

As you can see, the "`background`" command will also influence the background of the color of the "`print`" command.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

Print will automatically do a "word wrap". Print will place words that do not fit into the current line, automatically into the next line. Therefore avoid words that do not fit into a line, as these may not be printed at all. Use the "`UCV.write`" for such cases.

```
' This line will remove the cover
UCV.
```

Syntax

UCV.print|P1
UCV.pri|P1

Parameter Explanation

P1 - String/variable with the text that should be printed.

Example

```
'**********************************
' UCV.-Sample
' Shows scrolling text
'**********************************
'
$$TXT=Hallo my dear, try the PRINT command also ....
STR.CLONE|$$TXT|160|$$TXT
' no fading
UCV.nb
' set foreground color
UCV.fgr|&HFF0000
' set transparent background color
UCV.bgr|-1
UCV.mat
UCV.set font|futura t|40|||0
UCV.pri|$$TXT
PAU.4
UCV.
ENR.

'**********************************
' UCV.-Sample 2
'
'**********************************
'
' Make a completely red cover
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.print|Hello world
' This will remove the Cover
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.20 rsf - reset font

UCV.reset font - resete font for "print" and "write" Previous 1984 Top 41 Next 1990

MiniRobotLanguage (MRL)

UCV.reset font
UCV.rsf

Reset font for printing and writing to original values

Intention

This command enables you to reset the font-settings for printing and writing to the default settings. These are:

Font-Name = Courier New
Font-Size = X-Resolution/10
Font-Pitch = 0
Font-Style = 0
Font-Escapement = 0
Font-Charset = 0 (ANSI)

Once you have used the "`UCV.set font`" command, the chosen font is use for all print and write commands. Until you choose another font, or until you use this command to reset the values to the default values.
Use the "`UCV.foreground`" command to set the foreground color.

General usage is simple, here you can see an example that will set- and reset the font.

*© 2023 Theo Gottwald * https://www.smart-package.com*

It will generate a picture like the above picture.

```
UCV.nbi
' Make a completely red cover
UCV.bgr|&HFF0000
UCV.gra|&HFF00FF|&HFFFF00
' Switch writing color to yellow
UCV.fgr|&HFFFF00
' Set a font
UCV.font set|Futurat|164|18||350
UCV.write|Hello world!|350,620
' reset the font to default
UCV.rsf
' print the black shadow for a 3D Effect
UCV.fgr|&H000000
UCV.write|Hello world!|15,620
UCV.fgr|&H00FF00
UCV.write|Hello world!|17,625
PAU.5
' This will remove the Cover
UCV.
ENR.
```

Syntax

UCV.reset font
UCV.rsf

Parameter Explanation

-

Example

```
'**********************************
' UCV.-Sample
' Shows font setting and resetting
'**********************************
'
UCV.nbi
' Make a completely red cover
UCV.bgr|&HFF0000
UCV.gra|&HFF00FF|&H00FFFF
' Switch font color to yellow
UCV.fgr|&HFFFF00
UCV.font set|Futurat|164|18||350
UCV.write|Hello world!|350,620
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.rsf
UCV.fgr|&H000000
UCV.write|Hello world!|15,620
UCV.fgr|&H00FF00
UCV.write|Hello world!|17,625
PAU.5
' This will remove the Cover
UCV.
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.21 sft - set font

MiniRobotLanguage (MRL)

UCV.set font
UCV.sft

Choose alternative font for printing and writing

Intention

This command enables you to choose an alternative font for printing and writing. Once you have used the "`UCV.set font`" command, the chosen font is use for all print and write commands.

will print text in the actual foreground color. Use the "`UCV.foreground`" command to set the foreground color. "`UCV.print`" will use the font that you can set using the "`UCV.set font`" command.

General usage is simple, here you can see an example:

```
UCV.nb
' Make a completely red cover
UCV.bgr|&HFF0000
UCV.cov
' Switch font color to yellow
UCV.fgr|&HFFFF00
UCV.font set|-|164|18||-350
UCV.write|Hello world!|350,20
PAU.5
' This will remove the Cover
UCV.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

You can print a text on the cover, like this:

```
'
UCV.nbi
' Make a gradient cover
UCV.gra|&HFF00FF|&HFFFF00
' Switch font color to yellow
UCV.fgr|&HFFFF00
UCV.font set|FuturaT|164|18||350
UCV.write|Hello world!|350,620
PAU.5
' This will remove the Cover
UCV.
ENR.
```

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

UCV.set font[|P1][|P1][|P1][|P1][|P1]
UCV.sft[|P1][|P1][|P1][|P1][|P1]

Parameter Explanation

***P1 - (optional)* Fontname** - String/variable with the Font name e.g. "Arial", that should be used.

***P2 - (optional)* Fontsize** - Value/variable with the Fontsize e.g. 64. Font-Sizes should be in the Range between 4 and 260.

***P3 - (optional)* Fontstyle** - Value/variable with the Fontstyle:

0 - Normal
1 - Bold
2 - Italic
4 - Underline
8 - Strikeout
16 - Leading

You can use these values or add them and use a combination of them.

***P4 - (optional)* Fontpitch** - Value/variable with the Fontpitch:

0 - Default
1 - Fixed width font
2 - Variable width font
16 - Roman font (Times Roman...)
32 - Swiss font (Helvetica, Swiss...)
48 - Modern font (Pica, Courier...)
64 - Script font (Cursive...)
80 - Decorative (OldEnglish...)

You can use these values or add them and use a combination of them.

***P5 - (optional)* Escapement** - Value/variable with the Font-Escapement:
Specifies the angle, in tenths of degrees, between the character base line and the x axis.
Allows printing of text on an angle. If this value is not zero, any vertical scrolling is disabled.

P6 - (optional) ANSI-Charset - Value/variable with the Font-Charset.
Please use directly the number, combinations are impossible.
Here are some possible Charset-numbers.

```
0   - ANSI CharSet              162 - Turkish CharSet

1   - Default CharSet           177 - Hebrew CharSet
2   - Symbol CharSet            178 - Arabic CharSet
77  - Mac CharSet               186 - Baltic CharSet
128 - Shiftjis CharSet          204 - Russian CharSet
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
129 - Hangeul CharSet          222 - Thai CharSet
130 - Johab CharSet            238 - East Europe CharSet
136 - Chinese CharSet          255 - OEM CharSet
161 - Greek CharSet

More Infos, see here:
```

MSDN-CharSets

Any of the parameters can be omitted, in that case the previous value is been used.

Example

```
'**********************************
' UCV.-Sample
' Shows scrolling text
'**********************************
'
$$TXT=Hallo my dear, try the PRINT command also ....
STR.CLONE|$$TXT|160|$$TXT
' no fading
UCV.nb
' set foreground color
UCV.fgr|&HFF0000
' set transparent background color
UCV.bgr|-1
UCV.mat
UCV.set font|futurat|40|||0
UCV.pri|$$TXT
PAU.4
UCV.
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.55.8.22 spi - plot - set pixel

MiniRobotLanguage (MRL)

UCV.plot
UCV.set pixel
UCV.spi

Paint a single pixel onto the cover

Intention

This command will paint a single pixel at a pixel-coordinate, onto the cover. The pixel is been drawn in the foreground color. As this is a very general graphics command, it will enable you to draw all sorts of shapes from pixels.

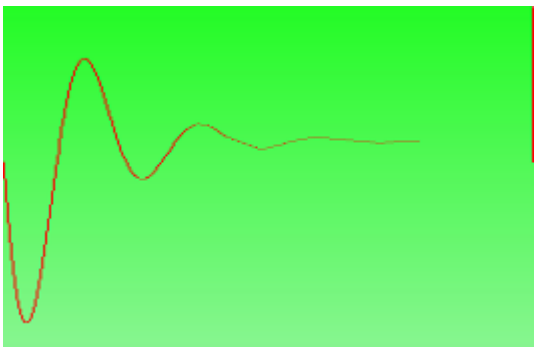
General usage is simple:

```
' Set foreground color
UCV.fgr|&HFF00FF
UCV.plot|600,1
```

Points that have been set using the "`set point`" command can be used as starting points for the "`Line`" command. In that case you can omit the second coordinate pair.

```
' We set a single Pixel at 1,1 which will then
' be the starting point for the line.
UCV.set pixel|1,1
' you can use mathematical inline calculations
' for all coordinates
UCV.line|($$XCO*2.5),600
```

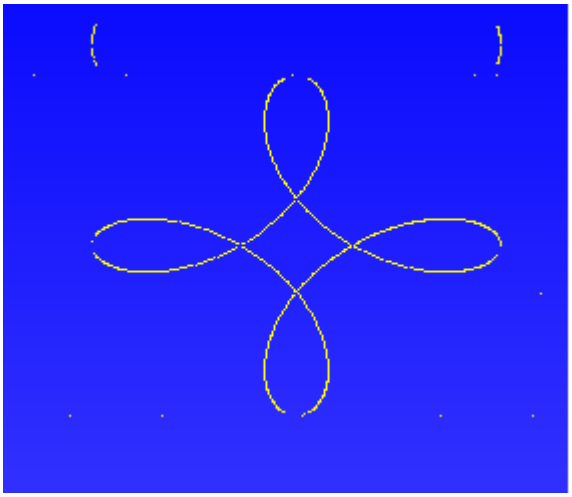
You can draw more complex examples, like this:



```
' compile before running
'#SPI:Forcewrite
UCV.nb
```

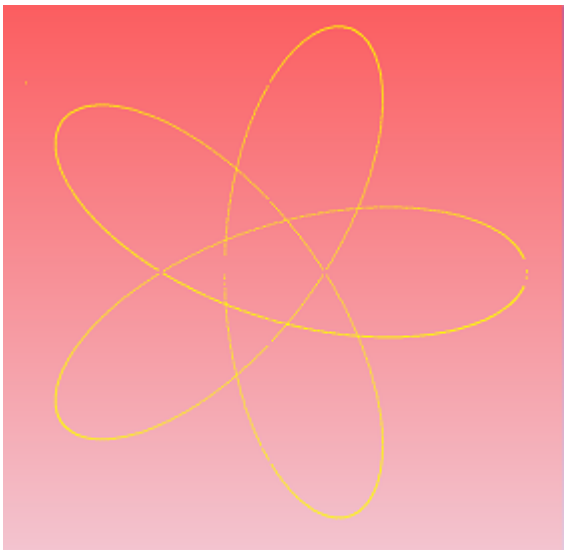

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.grd|&H00FF00
' Set red foreground color
UCV.fgr|&HFF0000
FOR.$$F00|0|16|0.01
  CAX.$$X00=2*$$F00
  CAX.$$Y00=EXP(-$$X00/4)*SIN($$X00)
  CAL.$$Y00=$$Y00*500+500
  CAL.$$X00=10*$$X00
  IVV.$$Y00>0
    UCV.drawto|$$X00,$$Y00
    UCV.plot|$$X00,$$Y00
  EIF.
NEX.
PAU.1
UCV.
ENR.


' The following script will also
' paint a mathematical figure
'#SPI:Forcewrite
UCV.nb
UCV.grd|&H0000FF
' Set red foreground color
UCV.fgr|&HFFFF00
FOR.$$F00|0|45|0.007
  $$C00=0.7
  CAX.$$X00=COS($$F00)+$$C00*COS(-3*$$F00)
  CAX.$$Y00=SIN($$F00)+$$C00*SIN(-3*$$F00)
  CAL.$$Y00=$$Y00*50+200
  CAL.$$X00=60*$$X00+200
  IVV.$$Y00>0
    'UCV.drawto|$$X00,$$Y00
    UCV.plot|$$X00,$$Y00
  EIF.
NEX.
PAU.1
ENR.
```



© 2023 Theo Gottwald * https://www.smart-package.com

```
'#SPI:Forcewrite
UCV.nb
UCV.grd|&HFF4545
' Set red foreground color
UCV.fgr|&HFFFF00
FOR.$$F00|0|45|0.002
  $$C00=0.7
  CAX.$$X00=COS(6*$$F00)+$$C00*COS(-4*$$F00)
  CAX.$$Y00=SIN(6*$$F00)+$$C00*SIN(-4*$$F00)
  CAL.$$Y00=$$Y00*260+700
  CAL.$$X00=260*$$X00+700
  IVV.$$Y00>0
    UCV.plot|$$X00,$$Y00
    UCV.drawto|($$X00+2),($$Y00+2)
    UCV.plot|$$X00,$$Y00
    UCV.drawto|($$X00),($$Y00+2)
    UCV.plot|$$X00,$$Y00
    UCV.drawto|($$X00+2),($$Y00
  EIF.
NEX.
PAU.1
ENR.
```



Syntax

UCV.plot|P1,P2
UCV.set pixel|P1,P2
UCV.spi|P1,P2

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1,P2 - Screen-Pixel-Coordinates for the starting Point of the line to be drawn.

Example

```
'**********************************
' UCV.-Sample
'
'**********************************
UCV.nbi
UCV.dbw
FOR.$$XCO|1|1200|4
  UCV.fgr|&HFF00FF
  UCV.line|1200,$$XCO|($$XCO*2.5),1
NEX.
PAU.4
UCV.
ENR.

'**********************************
' UCV.-Sample 2
' use Lines in random colors
'**********************************
'
UCV.nbi
UCV.dbw

FOR.$$XCO|1|1200|4
  UCV.fgr|&HFF00FF
  UCV.line|1200,$$XCO|($$XCO*2.5),1
  UCV.fgr|&HFFFF00
' Generate random colors
  RND.120|255|$$RED
  RND.120|255|$$GRN
  RND.120|255|$$BLU
  CAL.$$COL=$$RED+256*$$GRN+(256*256)*$$BLU
  UCV.fgr|$$COL
'-------------------------
  UCV.line|1200,($$XCO*2.5)
  UCV.line|($$XCO*2.5),1200
  UCV.line|($$XCO*2.5),1|($$XCO*2.5),1200
NEX.
PAU.4
UCV.
ENR.


'**********************************
' UCV.-Sample 2
' Mathematical figure in random colors
'**********************************
'
$$XCE=800
$$YCE=500
UCV.nb
UCV.cov
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.bgr|&HFFFFFF
UCV.fgr|&H0000FF
UCV.plot|$$XCE,$$YCE
FOR.$$VAA|0|($$XCE*2)
  GSB.rndcol
  CAX.$$XCO=$$XCE+300*SIN($$VAA*3.14/180)
  CAX.$$YCO=$$YCE+300*COS($$VAA*3.15/180)
  UCV.drawto|$$XCO,$$YCO
NEX.
PAU.5

ENR.

:rndcol
RND.120|255|$$RED
RND.120|255|$$GRN
RND.120|255|$$BLU
CAL.$$COL=$$RED+256*$$GRN+(256*256)*$$BLU
UCV.fgr|$$COL
RET.

'**********************************
' UCV.-Sample 2
' Mathematical figure
'**********************************
$$XMA=800
$$STP=15
UCV.nbi
UCV.dbw
UCV.fgr|&HFF00FF

FOR.$$XCO|1|($$XMA+$$STP)|$$STP
  UCV.drawto|0,$$XCO|$$XCO,$$XMA
  UCV.drawto|$$XCO,0|$$XMA,$$XCO
NEX.
PAU.5
ENR.
```

Remarks

-

Limitations:

PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.23 tra - transparency

UCV.transparency - make cover transparent **Previous** 1995 **Top** 41 **Next** 2005

MiniRobotLanguage (MRL)

UCV.transparency
UCV.tra

Change transparency value of Cover

Intention

This command is used to change the transparency value of the screen cover.
A value of 1 will make it completely transparent, a value of 255 will make it opaque.
All values between will change the cover visibility to any value between.

The transparency command is a modifier command. It is used to modify an existing cover, or to modify an cover that is soon to be created.

The "desktop-picture cover" is shown with the simple line:

```
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

The desktop-cover behavior can be changed/customized using the additional features:

```
UCV.transparency
UCV.no transparency
UCV.no blend
UCV.no blend-in
UCV.no blend-out
UCV.write
' this line will remove the cover
UCV.close
```

You can make the desktop-cover half transparent, by prefixing the script with:

```
UCV.tra
UCV.dc
PAU.10

' This line will remove the cover
UCV.
```

A transparency value of 0 (zero) which is used after the cover has been created, will deactivate the cover while its still there. You can switch it on later just by giving another transparency value. Here is an example:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

* PopUp-Menus and other TopMost Forms may be able to get temporarily in front of the Cover-form. This is due to windows-technical reasons and not a bug.

Syntax

UCV.transparency[|P1]
UCV.tra[|P1]

Parameter Explanation

P1 - `(optional)` transparency value between 1 and 255. If omitted 128 is used. A transparency value of 0 can only be used for a cover that already exists. In that case the cover will be deactivated. It can still be printed and painted on the cover, even though it seems not to be there. You can later make it visible just by changing the transparency value to a higher value.

Example

```
'**********************************
' UCV.-Sample
' In this Mode, the cover will fade in slowly
' which is invisible if no on going
' changes on the desktop happen.
'**********************************
' Switch cover on
UCV.desktop-copy
PAU.20
UCV.
ENR.

'**********************************
' UCV.-Sample 2
' In this Mode, the cover will appear immediately.
'
'**********************************
' prevent fade-in
UCV.no blend-in
' Switch cover on
UCV.desktop-copy
PAU.20
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.
ENR.


'**********************************
' UCV.-Sample 3
' In this Mode, the cover will appear
' and then be faded to transparent.
'
'**********************************
'
UCV.dc
UCV.tra
PAU.10

' This line will remove the cover
UCV.

'**********************************
' UCV.-Sample 4
' We will make the existing cover to disappear
' using a transparency value of 0.
' Then we print on the cover and just
' make it visible again.
'
'**********************************
'
UCV.cov
UCV.tra|0
UCV.bgr|&HFF00FF
UCV.pri|$$TXT
UCV.tra|126
PAU.4
UCV.
ENR.
```

Remarks

Any cover is been removed immediately, when the script-process ends or when the script runs into an error.

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937

© 2023 Theo Gottwald * https://www.smart-package.com

- CCV. - Color Conversion 11117

© 2023 Theo Gottwald * https://www.smart-package.com

3.55.8.24 wri - write

UCV.write - write text on cover Previous 2001 Top 41 Next 3572

MiniRobotLanguage (MRL)

UCV.write
UCV.wri

Print graphical text on to the cover

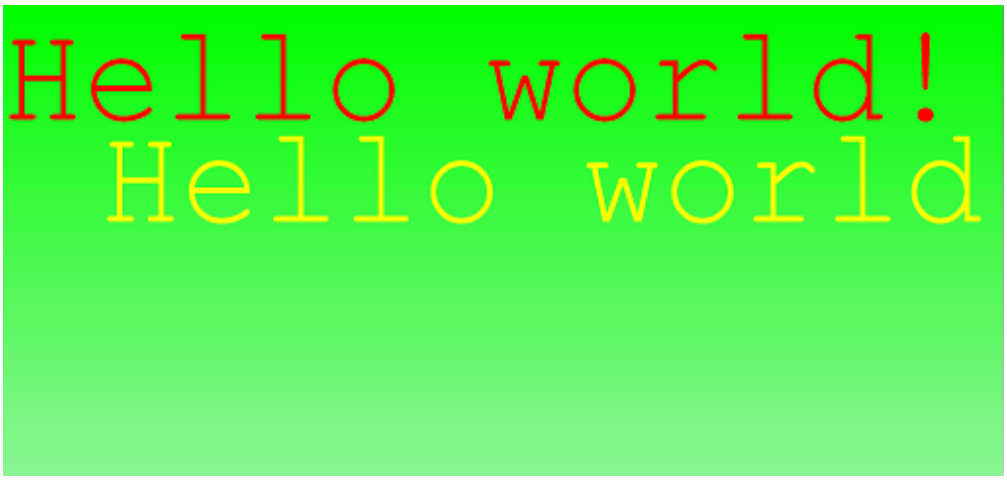
Intention

This command will print text in the actual foreground color. Use the "`UCV.foreground`" command to set the foreground color. "`UCV.write`" will use the font that you can set using the "`UCV.set font`" command.

General usage is simple:

```
UCV.nbi
UCV.grd|&H00FF00
' Set red foreground color
UCV.fgr|&HFF0000
' Print in red onto the green background
UCV.write|Hello world!
' We change color to yellow
UCV.fgr|&HFFFF00
UCV.wri|Hello world!|200,200
' This will remove the Cover
PAU.5
UCV.
ENR.
```

Will look like this:



*© 2023 Theo Gottwald * https://www.smart-package.com*

`UCV.write` will not automatically do a "word wrap". Therefore avoid words that do not fit into a line, as these may be cut off at the end otherwise. Use the "`UCV.print`" for such cases to have word wrap.

`UCV.write` will allow you to exactly specify the top,left corner of the text to print. Like this:

```
UCV.write|Hello world!|100,100
```

`UCV.write` will allow you further, to specify an alignment as a third parameter:

```
UCV.write|Hello world!|100,100|2
```

See below for details.

Syntax

UCV.write|P1[|P2,P3][|P4]
UCV.wri|P1|[|P2,P3][|P4]

Parameter Explanation

P1 - Text to write on to the Cover.

P2 - (optional) - X-Position value of the text to be printed, in pixels, screen coordinates.

P3 - (optional) - Y-Position value of the text to be printed, in pixels, screen coordinates.

P4 - (optional) - number between 1 and 4 that specifies the text-alignment. If omitted 1 is default.

```
LEFT       = 1
RIGHT      = 2
CENTER     = 3
JUSTIFY    = 4
```

Example

```
'**********************************
' UCV.-Sample
' Shows scrolling text
'**********************************
'
$$TXT=Hallo my dear, try the PRINT command also ....
STR.CLONE|$$TXT|160|$$TXT
' no fading
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
UCV.nb
' set foreground color
UCV.fgr|&HFF0000
' set transparent background color
UCV.bgr|-1
UCV.mat
UCV.set font|futura t|40|||0
UCV.wri|$$TXT
PAU.4
UCV.
ENR.

'**********************************
' UCV.-Sample 2
'
'**********************************
'
' Make a completely red cover
UCV.bgr|&H0000FF
UCV.cov
' Here comes your script, we use a PAUSE command instead
PAU.5
' Switch cover color to yellow
UCV.bgr|&H00FFFF
' Print on the cover screen
UCV.write|Hello world||2
' This will remove the Cover
UCV.
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's Undercover Operations 1911
- ! UCV. - Undercover Graphics 1937
- CCV. - Color Conversion 1117

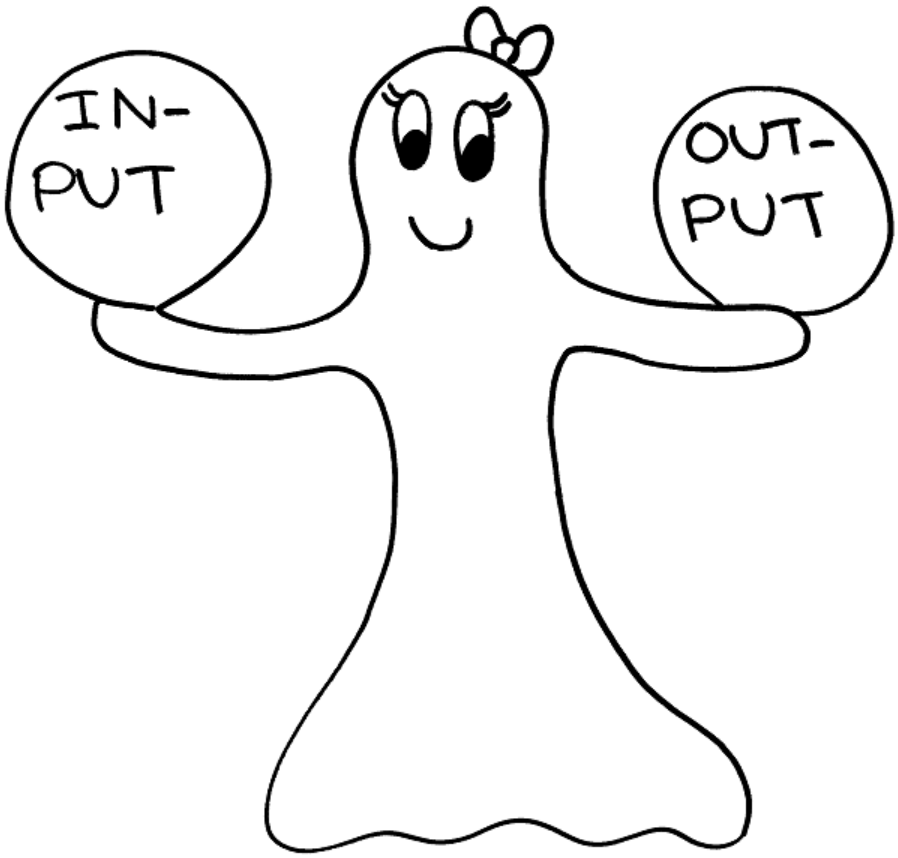
© 2023 Theo Gottwald * https://www.smart-package.com

3.56 Input- and Output commands

3.56.1 ! Smarty's Input and Output Commands

Smarty's Input and Output Commands Previous 2163 Top 41 Next 2009

Smarty's Input and Output Commands



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.56.2 CLP. - Clipboard - Operation

MiniRobotLanguage (MRL)

CLP. Command

Clipboard Operation

Intention

Return or set the Value of the System Clipboard(s).

Modes of operation:

```
1.    CLP.proto|$$MAIN[|$$CPT]
2.    CLP. (no parameter) put Clipboard -> TOS
3.    CLP.c - Clipboard Clear
4.    CLP.$$TXT - write Text to Clipboard
5.    CLP.r|$$RET|$$CPT
   ' read Text/Variable from Clipboard
6.    CLP.t|$$TXT|$$CPT
   ' write Variable-Content to Clipboard
7.    CLP.w|$$TXT|$$CPT
   ' wait for Clipboard-Change
```

Syntax

CLP.P1|P2[|P3]

Parameter Explanation

P1 - Type of Operation

Please specify only one prototype-letter for this command.
Successive letters will be ignored.
t - write Text to clipboard
r - read from Clipboard
c - Clear the Clipboard.
w - wait for Clipboard change.
P1 is maximum waiting-time in seconds or zero for endless.
P2 gets the new value from the clipboard.
The result can be in the original Variable P1 or if P3 is given, it will be placed in P3.

P2 - VAR with string or for result
P3 - optional: Clipboard Type to use.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Speed in Ticks:

This command uses typically around 1000 to 7000 Ticks heavily depending on Parameters.

Example

```
'**********************************
' CLP.-Sample 1
'**********************************
' Screenshot the desktop into Clipboard.


'**********************************
' CLP.-Sample 2
'**********************************
' Remove a picture from Clipboard
CLP.c||2
ENR.

'**********************************
' CLP.-DEMO
'**********************************
: §§TXT=http://www.it-berater.org

' When using no parameter, CLP. returns the Text-Clipboard-
Content on Stack
CLP.
PRT.Stack-Content:>$$000<
PRT.---------------------------
' When using one parameter, CLP. writes the String into the
Text-Clipboard
CLP.§§TXT
CLP.r|§§OUT
PRT.Clipboard: >§§OUT<
PRT.---------------------------
' Now we clear the Clipboard and use the systemvariable for
Clipboard
CLP.§§TXT
CLP.c
CLP.r|§§OUT
PRT.Clipboard: >$cpb$<>§§OUT<
PRT.---------------------------
' Starting Parallel-Robot that will change the clipboard
PRR.
  PAU.5
  CLP.t|Hallo!
  PAU.5
  CLP.t|Hey man!
  ENR.
PRE.
PRT. Waiting for Change!
CLP.§§TXT
CLP.w
PRT. Clipboard is: >$cpb$<
PRT.---------------------------
PRT. Now waiting for change with timeout 50 Seconds
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: §§TRIM=5
CLP.w|§§TIM|§§CIP
' The Changeflag is 0 if a timeout has occured and 1 if the
clipboard has changed in time
PRT.Clipboard Change to >§§CIP< Changeflag was: §§TIM
DMP.4
MBX.Ready
END.
```

Speed in Ticks:
This command uses typically around 7000 Ticks heavily depending on Parameters.

Remarks

The following Clipboard-Types can be used as third parameter (use the number!. If you are not sure for what you need this parameter, don't use it. Default value if omitted is 7.

```
TEXT =            1
BITMAP =          2
METAFILEPICT =    3
SYLK =            4
DIF =             5
TIFF =            6
OEMTEXT =         7
DIB =             8
PALETTE =         9
PENDATA =         10
RIFF =            11
WAVE =            12
UNICODETEXT =     13
ENHMETAFILE =     14
HDROP =           15
LOCALE =          16
DIBV5 =           17
MAX =             18 ' varies by Windows version
```

Limitations:

Use the ANA.ToClipboard-Command to place Pictures in the Clipboard.
To Read Pictures from the Clipboard, use the ANA.FromClipboard-Command.

See also:

- 3.1 Systemvariables 334

 © 2023 Theo Gottwald * https://www.smart-package.com

3.56.3 DDD. - Display Directory Dialog

MiniRobotLanguage (MRL)

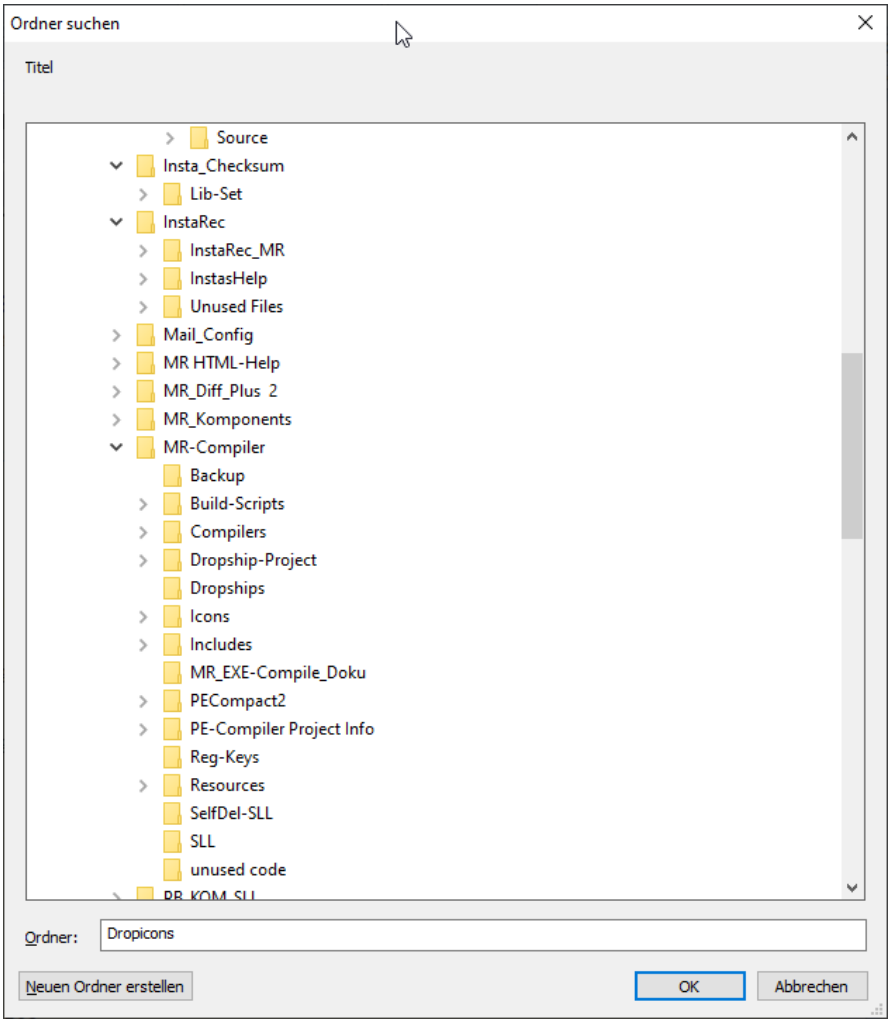
DDD. - Display Directory Dialog

Display Directory Selection Dialog-Box

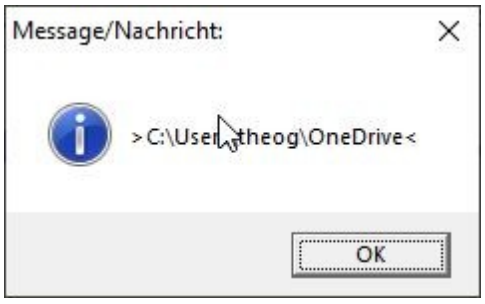
Intention

Generates a Dialog in which you can choose a Directory at Runtime of the Script.
Usage:

```
$$TIT=Titel
$$SFO=?path\
DDD.$$RET|$$TIT|$$SFO|?desktop\
MBX.>$$RET<
ENR.
```



*© 2023 Theo Gottwald * https://www.smart-package.com*



The result will be the Folder-Path without an "\" at the end.

Syntax

DDD.P1[|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - Variable to return the Folder-Path
P2 - (optional) - Titel. This Text/Explanation will be given to the User in the Dialog.
P3 - (optional) - Start-Path. The Dialog will start in this Folder.
P4 - (optional) - Default-Path. The Command will return this path, if the user hits "Cancel" or closes the Dialog.
If omitted, nothing is returned.
P5 - (optional) - Flags (add numbers to combine features)

1 Only returns file system directories. With any other selection, the OK button is grayed.
2 Does not include network folders below the domain level in the treeview control.

8 Only returns file system ancestors. With any other selection, the OK button is grayed.
16 Includes an edit control in the dialog box that allows the user to type the name of an item.
64 Provides the new user interface, a larger dialog box that can be resized. It also offers drag-and-drop capability within the dialog box, reordering, shortcut menus, new folders, delete, and other shortcut menu commands. This is the default style.
80 Use the new user interface, plus an edit box.
128 The dialog box can display URL's if 80 and 16384 are also set.
256 When this flag is combined with Flag 64, adds a usage hint to the dialog box in place of the edit box.
512 Do not include the "New Folder" button in the dialog box.
1024 When the selected item is a shortcut, return the PIDL of the shortcut itself rather than its target.
16384 The dialog box will display both files and folders.
32768 The dialog box can display shareable resources on remote systems. It is intended for applications that want to expose remote shares on a local system. Forthis, 64 flag must also be set.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' DDD.-Sample
'**********************************
' Titel
$$TIT=Titel
' Path where the Dialog starts
$$SFO=?path\
' ?desktop is the default return if the user hits "Cancel"
DDD.$$RET|$$TIT|$$SFO|?desktop\
MBX.>$$RET<
ENR.
```

Remarks

-

Limitations:

-

See also:

- DFD. - Display File Dialog 2015
- DSD. - Display Save-File Dialog 2021
- **SAY. - Say something** 2048
- **MSB. - Message Beep** 2040
- **MPS. - Music Play Speaker** 2536

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.4 DFD. - Display File Dialog

MiniRobotLanguage (MRL)

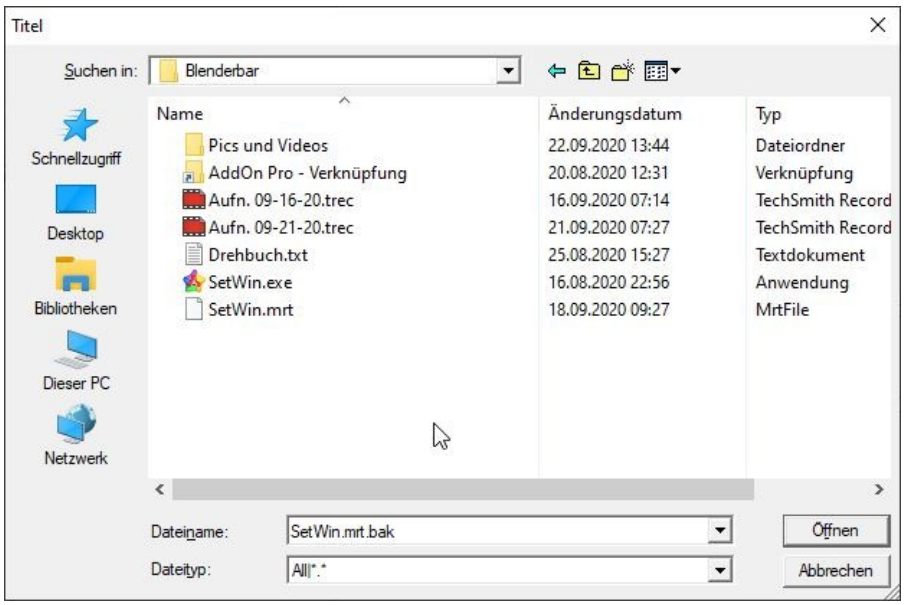
DFD. - Display File Dialog

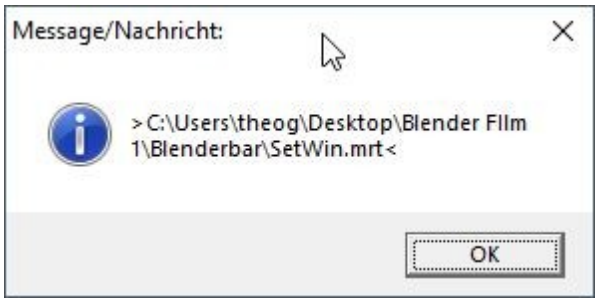
Display File Selection Dialog-Box

Intention

Generates a Dialog in which you can choose a one or more files at Runtime of the Script.
Usage:

```
$$TIT=Titel
$$SFO=?path\
DFD.$$RET|$$TIT|$$SFO|?desktop\
MBX.>$$RET<
ENR.
```

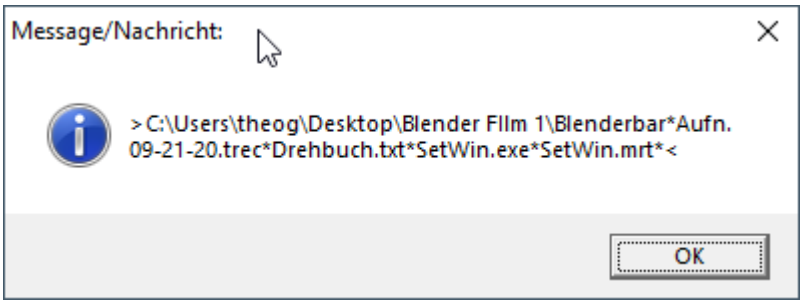




The result will be the File and -Path.
If you use "Multiselect" like this:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$TIT=Titel
$$SFO=?path\
' Use default style plus Multiselect
CAL.$$FLA=11081730+512
DFD.$$RET|$$TIT|$$SFO|$$FLA
MBX.>$$RET<
ENR.
```



You will get the many results like this, separated wit "*".

Syntax

DFD.P1[|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - Variable to return the FIle(s)
P2 - (optional) - Titel. This Text/Explanation will be given to the User in the Dialog.
P3 - (optional) - Start-Path. The Dialog will start in this Folder.
P4 - (optional) - **Flags** (see below, add the numbers to combine the features) set to "0" for default style.
P5 - (optional) - Default-Extension.
P6 - (optional) - Stringfilter. Sample: "BASIC|*.BAS;*.INC;*.BI|"

Flag Options:

2	Prompt for Overwriting existing Files.
8	Maintains current directory regardless of user selection.
16	The Help button is shown
256	File name not validated For invalid characters.
512 -	Allow multiple selections. Return values are "*"-terminated.
1024	Allow different Extensions
2048	Allows only valid paths and filenames.
4096 -	File name must exist.
8192 -	Allow file which does not exist.
16384	Returns network filename even if sharing violation occurs.
65536	File not created before the dialog Is closed.
131072	Hides and disables the network button.
2097152	Return Long Filenames
8388608 -	Dialog Is resizable.
524288 -	Explorer style interface (default state).
1048576	Returns name of shortcut file.

© 2023 Theo Gottwald * https://www.smart-package.com

11081730 - Default Style (%OFN_EXPLORER OR %OFN_FILEMUSTEXIST OR %OFN_LONGNAMES OR %OFN_OVERWRITEPROMPT OR %OFN_PATHMUSTEXIST OR %OFN_ENABLESIZING OR %OFN_NOTESTFILECREATE)

Example

```
'**********************************
' DFD.-Sample
'**********************************
' Titel
$$TIT=Titel
' Path where the Dialog starts
$$SFO=?path\
' ?desktop is the default return if the user hits "Cancel"
DDD.$$RET|$$TIT|$$SFO|?desktop\
MBX.>$$RET<
ENR.
```

Remarks

-

Limitations:

-

See also:

- DDD. - Display Directory Dialog 2012
- DSD. - Display Save-File Dialog 2021
- SAY. - Say something 2048
- MSB. - Message Beep 2040
- MPS. - Music Play Speaker 2536

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.5 DIT. - Dial Tone MFV

MiniRobotLanguage (MRL)

DIT. Command

Generate MFV-Dialtones through the Soundcard

Intention

Generates a Dial-Tone-Sound that can be attached to a telephone line to dial a number.
Together with its ability to speak (`SAY.`) and the ability to play Sound-Files (`MPA.`etc.), the Robot can call you on the phone and tell you what is going on.

Modes of operation:

1. Use `DIT.` just with the number and the default speed of 100 ms per Tone:
`DIT.1234567890`

2. Use `DIT.` and dial faster, using a shorter duration for each tone:
`DIT.1234567890|20`

3. Use `DIT.` and dial slower, using a longer duration for each tone:
`DIT.1234567890|420`

```
' You can misuse DIT. to make Music ...
' Music Mode:
FOR.$$CNT|1|2
  PRR.
    DIT.0943561954642210243|213
  PRE.
NEX.
PRR.
  DIT.094353246192210243|203
PRE.
PAU.10
ENR.
```

Syntax

DIT.P1[|P2]

Parameter Explanation

P1 - Number to dial. Following characters are supported:

 *© 2023 Theo Gottwald * https://www.smart-package.com*

0 1 2 3 4 5 6 7 8 9 * # A B C D

unsupported characters are ignored.

P2 - (optional) - if omitted, a default duration of 100 ms per Tone is used. If given, this

is the duration in ms that a Tone will stay.

Example

```
'***********************************
' DIT.-Orchestra
'***********************************
PRR.
  DIT.0334920243|203
PRE.
PRR.
  DIT.0334920243|203
PRE.
PRR.
  DIT.0334920243|203
PRE.
PRR.
  DIT.0334920243|203
PRE.
ENR.

'***********************************
' DIT.-Orchestra II
'***********************************
FOR.$$CNT|1|2
  PRR.
    DIT.094356192210243|203
  PRE.
NEX.
PRR.
  DIT.094356192210243|203
PRE.
PAU.10
ENR.
```

Remarks

-

Limitations:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

See also:

- SAY. - Say something 2048
- MSB. - Message Beep 2040
- MPS. - Music Play Speaker 2536

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.56.6 DSD. - Display Save-File Dialog

MiniRobotLanguage (MRL)

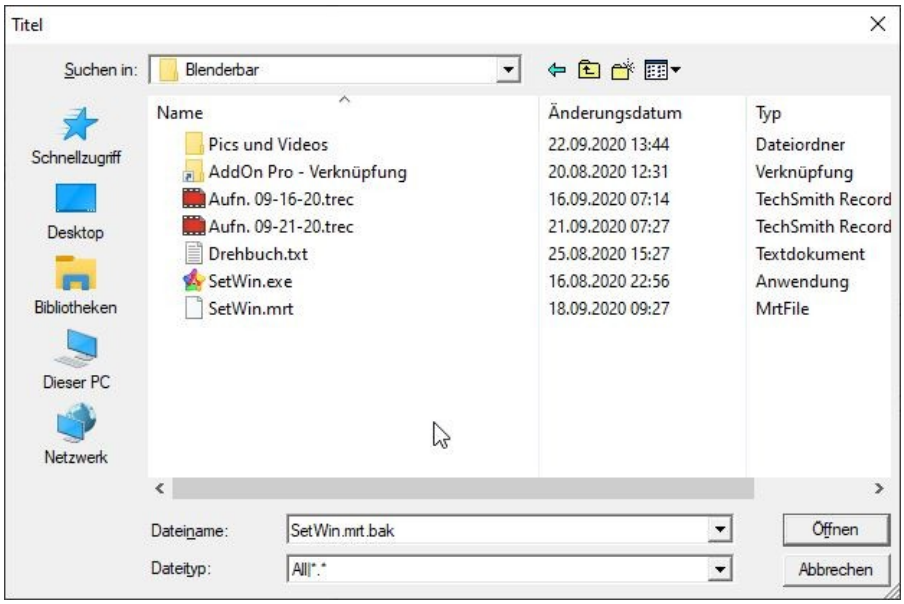
DSD. - Display Save-File Dialog

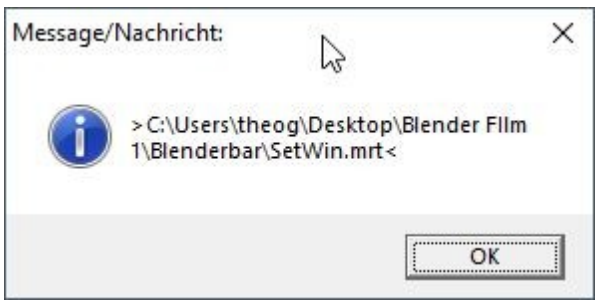
Display Save-File Selection Dialog-Box

Intention

Generates a Dialog in which you can choose a one or more files for Saving at Runtime of the Script. Usage:

```
$$TIT=Titel
$$SFO=?path\
DSD.$$RET|$$TIT|$$SFO|?desktop\
MBX.>$$RET<
ENR.
```



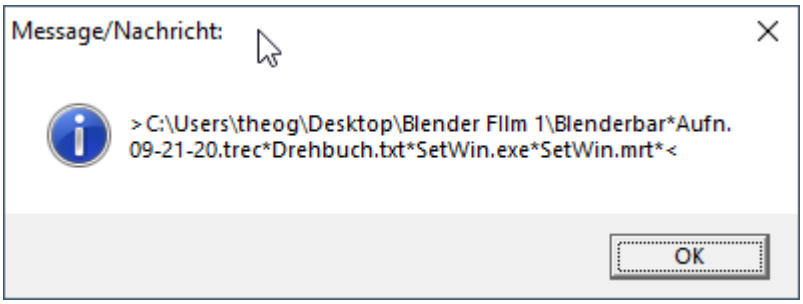


The result will be the File and -Path.
If you use "Multiselect" like this:

```
$$TIT=Titel
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$SFO=?path\
' Use default style plus Multiselect
CAL.$$FLA=11081730+512
DSD.$$RET|$$TIT|$$SFO|$$FLA
MBX.>$$RET<
ENR.
```



You will get the many results like this, separated wit "*".

Syntax

DSD.P1[|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - Variable to return the FIle(s)
P2 - (optional) - Titel. This Text/Explanation will be given to the User in the Dialog.
P3 - (optional) - Start-Path. The Dialog will start in this Folder.
P4 - (optional) - **Flags** (see below, add the numbers to combine the features) set to "0" for default style.
P5 - (optional) - Default-Extension.
P6 - (optional) - Stringfilter. Sample: "BASIC|*.BAS;*.INC;*.BI|"

Flag Options:

2	Prompt for Overwriting existing Files.
8	Maintains current directory regardless of user selection.
16	The Help button is shown
256	File name not validated For invalid characters.
512 -	Allow multiple selections. Return values are "*"-terminated.
1024	Allow different Extensions
2048	Allows only valid paths and filenames.
4096 -	File name must exist.
8192 -	Allow file which does not exist.
16384	Returns network filename even if sharing violation occurs.
65536	File not created before the dialog Is closed.
131072	Hides and disables the network button.
2097152	Return Long Filenames
8388608 -	Dialog Is resizable.
524288 -	Explorer style interface (default state).
1048576	Returns name of shortcut file.

© 2023 Theo Gottwald * https://www.smart-package.com

11081730 - Default Style (%OFN_EXPLORER OR %OFN_FILEMUSTEXIST OR %OFN_LONGNAMES OR %OFN_OVERWRITEPROMPT OR %OFN_PATHMUSTEXIST OR %OFN_ENABLESIZING OR %OFN_NOTESTFILECREATE)

Example

```
'**********************************
' DSD.-Sample
'**********************************
$$TIT=Titel
$$SFO=?path\
' Use default style plus Multiselect
CAL.$$FLA=11081730
DSD.$$RET|$$TIT|$$SFO|$$FLA
MBX.>$$RET<
ENR.
```

Remarks

-

Limitations:

-

See also:

- DDD. - Display Directory Dialog 2012
- DFD. - Display File Dialog 2015
- SAY. - Say something 2048
- MSB. - Message Beep 2040
- MPS. - Music Play Speaker 2536

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.7 GUP. - Get User Password

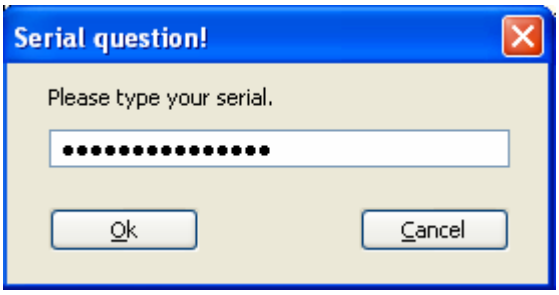
MiniRobotLanguage (MRL)

GUP. Command

Get User Password

Intention

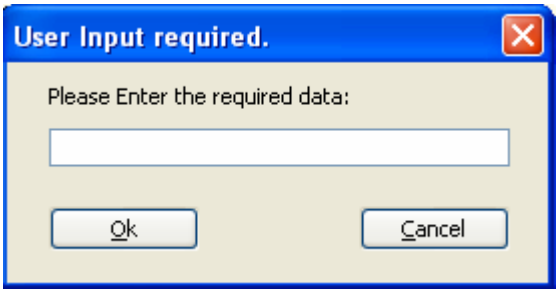
Get an input text/password from the user. The difference to GUT is, that a password-textbox is used. When the user types, it looks like this:



Usage is easy:

```
GUP.
```

will show a dialog like this:



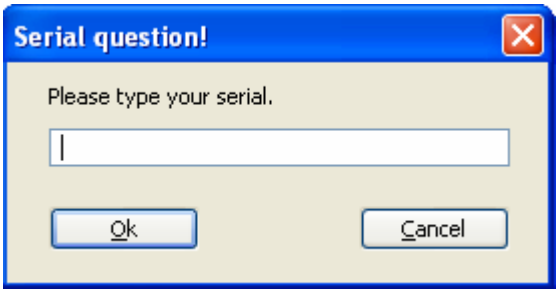
The cursor will blink, and the dialog will immediately take user input. The user can type up to 260 characters into the textbox. The result, will be place on TOS 104. You can try it using this sample script:

```
GUP.
STS.DUMP
MBX.Ready?
ENR.
```

Next you may want to change the Dialog-text and the dialog Headline.
These are parameters **P2** and **P3**:

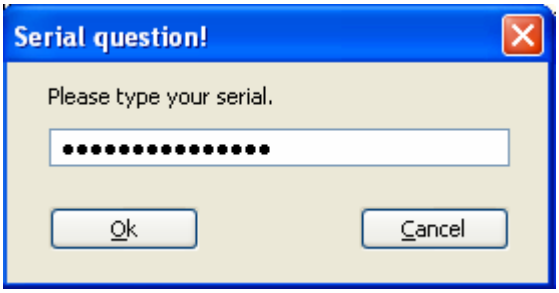
© 2023 Theo Gottwald * https://www.smart-package.com

```
GUP.Please type your serial.|Serial question!
```



You may want to add a "default text". Like the one you can see in the picture below.

```
GUP.Please type your serial.|Serial question!|email@email.com
```



And finally you can have the result in the variable, **P4**.
Here is a complete Sample:

```
' Get User Text
: $$TXT=Please enter your mailadress.
: $$HEA=
: $$DEF=yourmail@yourdomain.com
GUP.$$TXT|$$HEA|$$DEF|$$RES

' Remove Invalid characters
GFS.e|$$RES

' Display result
MBX.Your Mailadress is:$crlf$$$RES
ENR.
```

Syntax

GUP.[P1][|P2][|P3][|P4]

Parameter Explanation

P1 - ***(optional)*** Text/Explanation to the user. If omitted, the phrase "Please Enter the required data:" is used. If you really want to have no text, use a blank, like this

© 2023 Theo Gottwald * https://www.smart-package.com

```
' complete empty input-box
GUP. | |
```

P2 - (optional) Text, window title. If omitted the phrase "User Input required!" is used.

If you really want to have no text, use a blank, like this

```
GUP. |Serial question!
```

P3 - (optional) Text, default Text for the textbox. If omitted no default text is given.

P4 - (optional) Variable that will receive the user-text. If omitted, user-input is placed on TOS 104.

Example

```
'*****************************************************
'  GUT. / GUP. - Sample
'*****************************************************

' Get User Mailadress
: $$TXT=Please enter your mailadress.
: $$HEA=
: $$DEF=yourmail@yourdomain.com
GUT.$$TXT|$$HEA|$$DEF|$$MAI

' Remove Invalid characters
GFS.e|$$MAI

:again
' Get User Password
: $$TXT=Please enter your password.
: $$HEA=You MUST enter something!
: $$DEF=
GUP.$$TXT|$$HEA|$$DEF|$$RES
' Trim Spaces away
VBT.$$RES
' Convert to small letters
VTL.$$RES
' Jump back if empty string is given
IVV.$$RES=
  GTO.again
EIF.
' Display result
MBT.Your Mailadress is:$crlf$$$MAI$crlf$$crlf$Your password in
small letters is:$crlf$$$RES
ENR.
'*****************************************************
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

If you do want to have secret information to be typed in, use `GUP.` instead.

Limitations:

-

See also:

- **GUT. - Get User Text** 2028
- **DBP - Debug Print** 769
- DBM. - DeBugMode 765
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.56.8 GUT. - Get User Text

GUT. - Get User Text **Previous** 2024 **Top** 41 **Next** 2031

MiniRobotLanguage (MRL)

GUT. Command

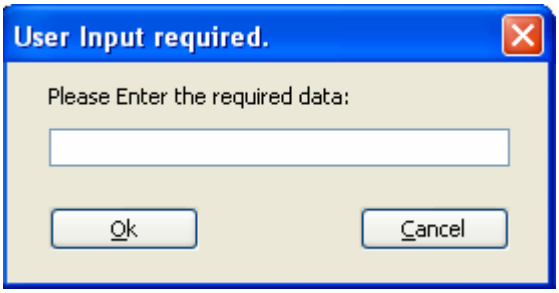
Get User Text

Intention

Get an input text from the user. Usage is easy:

```
GUT.
```

will show a dialog like this:

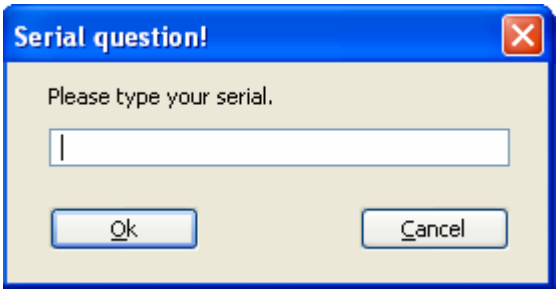


The cursor will blink, and the dialog will immediately take user input. The user can type up to 260 characters into the textbox. The result, will be place on TOS 104. You can try it using this sample script:

```
GUT.
STS.DUMP
MBX.Ready?
ENR.
```

Next you may want to change the Dialog-text and the dialog Headline.
These are parameters **P2** and **P3**:

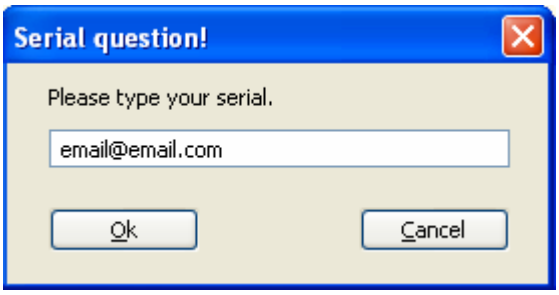
```
GUT.Please type your serial.|Serial question!
```



You may want to add a "default text". Like the one you can see in the picture below.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
GUT.Please type your serial.|Serial question!|email@email.com
```



And finally you can have the result in the variable, **P4**.
Here is a complete Sample:

```
' Get User Text
: $$TXT=Please enter your mailadress.
: $$HEA=
: $$DEF=yourmail@yourdomain.com
GUT.$$TXT|$$HEA|$$DEF|$$RES

' Remove Invalid characters
GFS.e|$$RES

' Display result
MBX.Your Mailadress is:$crlf$$$RES
ENR.
```

Syntax

GUT.[P1][|P2][|P3][|P4]

Parameter Explanation

P1 - ***(optional)*** Text/Explanation to the user. If omitted, the phrase "Please Enter the required data:" is used. If you really want to have no text, use a blank, like this

```
' complete empty input-box
GUT.
```

P2 - ***(optional)*** Text, window title. If omitted the phrase "User Input required!" is used.
If you really want to have no text, use a blank, like this

```
GUT.|Serial question!
```

P3 - ***(optional)*** Text, default Text for the textbox. If omitted no default text is given.

P4 - ***(optional)*** Variable that will receive the user-text. If omitted, user-input is placed on TOS 104.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
'  GUT. - Sample
'*****************************************************
' Get User Text
: $$TXT=Please enter your mailadress.
: $$HEA=
: $$DEF=yourmail@yourdomain.com
GUT.$$TXT|$$HEA|$$DEF|$$RES

' Remove Invalid characters
GFS.e|$$RES

' Display result
MBX.Your Mailadress is:$crlf$$$RES
ENR.'*****************************************************
```

Remarks

If you do want to have secret information to be typed in, use `GUP.` instead.

Limitations:

-

See also:

- **GUP. - Get User Password** 2024
- **DBP - Debug Print** 769
- DBM. - DeBugMode 765
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.9 MBT. - MessageBox Timed

MiniRobotLanguage (MRL)

MBT. Command

MessageBoX Timed

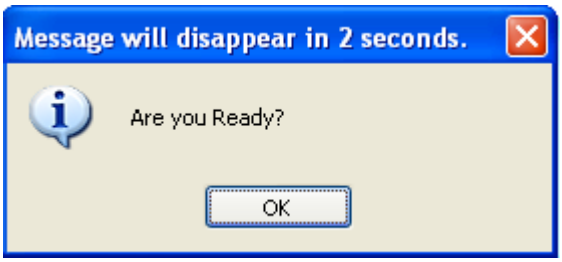
Intention

Display a message box containing a text string and optional title. This Messagebox will disappear automatically after an given amount of time.

The easiest usage is:

```
MBT.
```

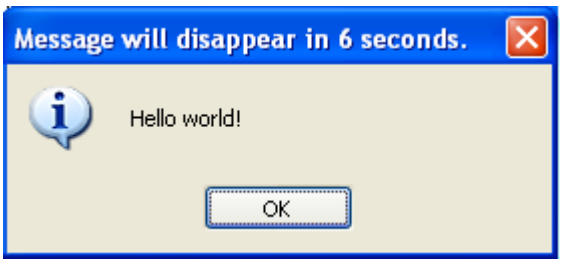
And this is the result:



You can give an Text for the Message like this:

```
MBT.Hello world!
```

And here is the result



And you can add a window title like this:

```
MBT.Hello world!|My Hello-Box will close in %t seconds.
```

which looks like this:

*© 2023 Theo Gottwald * https://www.smart-package.com*

Then you can change the number of seconds, the Messagebox will stay before it is automatically closed. In this case we choose 32 seconds:

```
MBT.Hello world!|My Hello-Box will close in %t seconds.|32
```

Next you can change the number of button or the appearance of the messagebox.

```
MBT.Hello world!|My Hello-Box will close in %t seconds.|32|48
```

And here are some results:

Flag: 32

Flag: 48

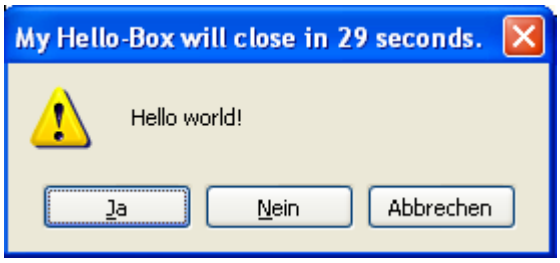
Flag: 16

© 2023 Theo Gottwald * https://www.smart-package.com

If you combine the flags, you get something like this:

```
MBT.Hello world!|My Hello-Box will close in %t seconds.|32|
(48+3)
```

Flag: 48+3



See more options below.

`MBX.` can also be used for Debugging. If you want to see the value of a variable at some place, you can just write:

```
' Check variable content
MBX.$$TXT
```

Important note: Robot-Messageboxes are always TopMost. They will be in front of all non TopMost windows. This is necessary, because the Editor can be set to TopMost, and in this case the Messageboxes must be shown on Top of the Editor to be seen.

Syntax

MBT.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - ***(optional)*** Text may contain systemvariables, like for example $crlf$, which will force a newline.

P2 - ***(optional)*** - Window-Title Text. If omitted, "Message/Nachricht" is used.

P3 - ***(optional)*** numeric Value, time in seconds, before Messagebox will be automatically closed.

P4 - ***(optional)*** numeric Value, see below.

Values for P2 change the number and Text of the displayed buttons:

00 Display OK button (default)

© 2023 Theo Gottwald * https://www.smart-package.com

```
01 OK and Cancel
02 Abort, Retry and Ignore
03 YES, NO and CANCEL
04 YES and NO
05 Retry and Cancel
06 Cancel, Retry and Continue

Add this to get a additional Help-Button:
12288 Help-Button

Add this values to change the Graphic displayed:
16 Serious problem
32 Question mark
48 Exclamation mark
64 Information (default)

Add this value to set the default button:

0   Default Button is Button Nr. 1
256 Default Button is Button Nr. 2
512 Default Button is Button Nr. 3
768 Default Button is Button Nr. 4

Add this value to change the Screen-Arrangement of the
Messagebox:

0       Application Modal
4096    System-Modal
8192    Taskmodal
262144  TopMost (default and always set.)
524288  Right align Window-Text
1048576 Right align Window-Title


Add this numbers to change Focus:
32768   No Focus
65536   Set Foreground Window
131072  Desktop
```

P5 - (optional) - Variable that will contain the result of the User-Input. If omitted, TOS 104 is used.

The returned values are:

OK - 1
Cancel - 2
Abort - 3
Retry - 4
Ignore - 5
Yes - 6
No - 7
TimeOut - 0

In case of Timeout, the result is zero (0) and the **Timeout-Flag is set.**

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
' A Messagebox will appear, Text is "Hallo"
' Title is Default Title
MBT.Hallo

' A Messagebox will appear, Text is "Hallo"
' Title is "MyTitel"
MBT.Hallo||MyTitle

: $$INP=My Hello-Box will close in %t seconds.
MBT.Hello world!|$$INP|5|(48+3)|$$TXT
MBT.$$TXT
ENR.
```

Remarks

-

Limitations:

The Parameters may not have the "|" inside as it is used a Separator.

See also:

- MBX. - MessageBox 2036
- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- PRT. PRinT 2045
- SAY. - Say something 2048

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.10 MBX. - MessageBox

MiniRobotLanguage (MRL)

MBX. Command

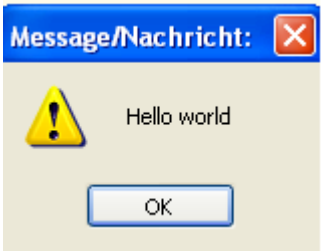
MessageBoX

Intention

Display a message box containing a text string and optional title. The easiest usage is:

```
MBX.Hello world
```

And this is the result:



Next you can add a window title like this:

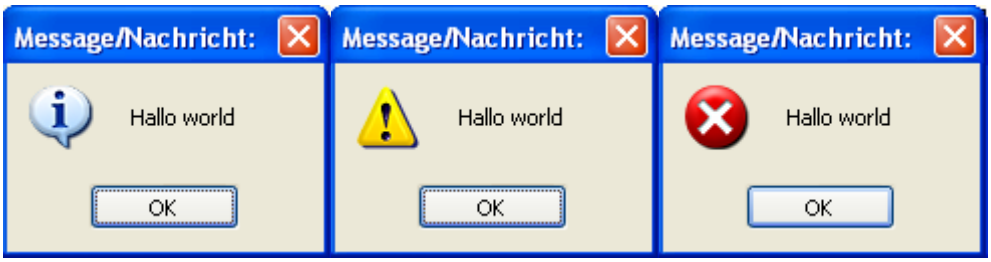
```
MBX.Hello world||Question
```

which looks like this:

Next you can change the number of button or the appearance of the messagebox.

```
MBX.Hello world|64
```


© 2023 Theo Gottwald * https://www.smart-package.com



MBX. can also be used for Debugging. If you want to see the value of a variable at some place, you can just write:

```
' Check variable content
MBX.$$TXT
```

Important note: Robot-Messageboxes are always TopMost. They will be in front of all non TopMost windows. This is necessary, because the Editor can be set to TopMost, and in this case the Messageboxes must be shown on Top of the Editor to be seen.

Syntax

MBX.P1[|P2][|P3][|P4]

Parameter Explanation

P1 **-** ***(optional)*** Text may contain systemvariables, like for example $crlf$, which will force a newline.

P2 **-** ***(optional)*** numeric Value, see below.

Values for P2 change the number and Text of the displayed buttons:

```
00 Display OK button (default)
01 OK and Cancel
02 Abort, Retry and Ignore
03 YES, NO and CANCEL
04 YES and NO
05 Retry and Cancel
06 Cancel, Retry and Continue

Add this to get a additional Help-Button:
12288 Help-Button

Add this values to change the Graphic displayed:
16 Serious problem
32 Question mark
48 Exclamation mark
64 Information (default)
```


© 2023 Theo Gottwald * https://www.smart-package.com

Add this value to set the default button:

```
0   Default Button is Button Nr. 1
256 Default Button is Button Nr. 2
512 Default Button is Button Nr. 3
768 Default Button is Button Nr. 4
```

Add this value to change the Screen-Arrangement of the Messagebox:

```
0       Application Modal
4096    System-Modal
8192    Taskmodal
262144  TopMost (default and always set.)
524288  Right align Window-Text
1048576 Right align Window-Title
```

Add this numbers to change Focus:

```
32768   No Focus
65536   Set Foreground Window
131072  Desktop
```

P3 - (optional) - Window-Title Text. If omitted, "Message/Nachricht" is used.

P4 - (optional) - Variable that will contain the result of the User--Input. If omitted, TOS[104] is used.

The returned values are:

OK - 1
Cancel - 2
Abort - 3
Retry - 4
Ignore - 5
Yes - 6
No - 7
TimeOut - 0

Example

```
' A Messagebox will appear, Text is "Hallo"
' Title is Default Title
MBX.Hallo

' A Messagebox will appear, Text is "Hallo"
' Title is "MyTitel"
MBX.Hallo||MyTitle

' Example using Variables
' Global Variable
: $$TXT=Number is
' Local Variable
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: §§NUM=12
MBX.$$TXT§§NUM|Number is ...
```

Remarks

there are 3 Versions, depending on the Number of Parameters:

```
1. MBX.[Variable(s) or Text]
2. MBX.[Variable(s) or Text]|Variable or Value
3. MBX.[Variable(s) or Text]|Variable or Value|[Variable(s) or
Text]
```

The Result will be put on TOS.

Limitations:

The Parameters may not have the "|" inside as it is used a Separator.

See also:

- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- PRT. PRinT 2045
- SAY. - Say something 2048

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.11 MSB. - Message Beep

MiniRobotLanguage (MRL)

MSB. Command

Message Beep through PC-Speaker

Intention

Generates a Signal Beep, through the PC-Speaker. Frequency and Duration can be customized. Useful for debugging. Useful to signal errors and states without showing a text.

Modes of operation:

1. Just use it without Parameter:
```
' Will produce a short Beep-Signal
MSB.
```

2. Use with Frequency
```
' Make a 100 Hz Sound for ~ 1 Second
MSB.100
```

3. Use default frequency (thats 800 Hz) and own Duration
```
MSB.|14
```

4. Use custom Frequency and Duration
```
MSB.600|10
```

5. Use default sound patterns:

```
' Generate a siren like sound
MSB.siren|780|650|2

'  Coin bouncing, sort of
MSB.bounce|760|280

' "Falling deep" like sound
MSB.fall|1780|150

' Switch between two tones
MSB.police|170|1780
```

Syntax

MSB.[P1][|P2][|P3][|P4]

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - (optional) - can be omitted, in that case the default frequency of 800 Hz is used, to generate a Beep Sound. If given, it can be

1.) a frequency between 1 Hz and 20000 Hz (depending on what your speaker can make that you can still hear). In that case **P2** will be a Duration in terms on 1/10 of a second.

2.) a Keyword from the following:

- **bounce** - use 2 numeric parameter
- **fall** - use 2 or 3 numeric parameter
- **siren** - use 2 or 3 numeric parameter
- **police** - use 2 or 3 numeric parameter

In this case, **P2** till **P4** will be the additional parameters for these predefined sound patterns.

P2 - (optional) - can be the duration of the frequency or a parameter for any of the above sound-patterns.

P3 - (optional) - if sound-patterns are used, this is one of their parameters.
P4 - (optional) - if sound-patterns are used, this is one of their optional parameters.

Example

```
'**********************************
' MSB. Sample
'**********************************
MSB.siren|780|650|4
MSB.police|170|1780|4
MSB.bounce|780|650
MSB.800|5
END.
```

Remarks

Use `PRR.` and `PRE.` to have your Melody play in the background.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- SAY. - Say something 2048
- DIT. - Dial Tone MFV 2018
- MPS. - Music Play Speaker 2536

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.56.12 OUT. - Send to Standard-Output

MiniRobotLanguage (MRL)

OUT. Command

Send String to the Standard-Output

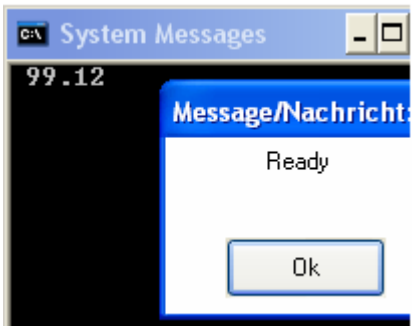
Intention

The OUT.-Command can be used to print a Line into the Debug-Console-Window or wherever Standard Output is directed to.

```
OUT. Hello World!
```

You can use variables.

```
: $$TIM=99.12
OUT.$$TIM
ENR.
```



This is the output to console window. Seethe Messagebox that prevents the console from being closed automatically, when the script ends.

Syntax

OUT.[P1]

Parameter Explanation

P1 - Variable or String

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'******************************************************
' EXAMPLE 1: PRT
'******************************************************
OUT.?path\

OUT.Hallo World

:over
MBX.Halt to read output
'******************************************************
```

Remarks

-

Limitations:

-

See also:

- DBP - Debug Print 769
- DBM. - DeBugMode 765
- MBX. - MessageBox 2036
- SAY. - Say something 2048

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.13 PRT. PRinT

PRT. - Print line in Debug-Console-Window **Previous** 2040 **Top** 41 **Next** 2048

MiniRobotLanguage (MRL)

PRT. Command

Print line in Debug-Console-Window

Intention

The `PRT`-Command can be used to print a Line into the Debug-Console-Window or into the Debug-window of the editor. Usage is as simple as:

```
PRT. Hello World!
```

You can use variables, or make a rounding of numbers.

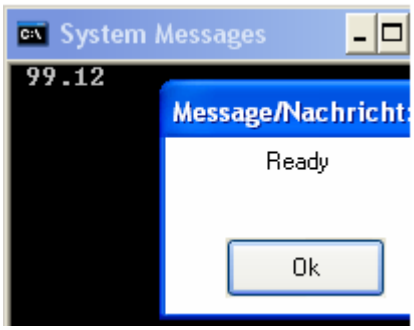
```
: $$TIM=99.123456789
PRT.$$TIM|rd
ENR.
```

This will Print the rounded value of:

```
[9]  99.12
```

into the Debug-window in the Editor. If you leave the d away, the `PRT.` will open a console window and print into this. You need a MBX.-command at the end, to prevent the console window from closing when the script ends. See this example:

```
: $$TIM=99.123456789
PRT.$$TIM|r
MBX.Ready
ENR.
```



This is the output to console window. Seethe Messagebox that prevents the console from being closed automatically, when the script ends.

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

PRT.[P1]|P2]

Parameter Explanation

P1 **-** Variable

P2 **-** Options

```
 Possible options are:
- |a do complete Variable Substitution
- |h Print Hex-Dump (Hex values of ASC-Characters)
- |d print in Debugwindow of Editor
- |r do a rounding on numbers (e.g. 99.1234 -> 99.12)
- |n do NO Specialfolders and SYSVAR-Processing
- |ra (both options)
```

Example

```
'*****************************************************
' EXAMPLE 1: PRT
'*****************************************************
PRT.?path\

PRT.

PRT.?path\|d

:over
MBX.Halt to read output
'*****************************************************
```

Remarks

-

Limitations:

-

See also:

- DBP - Debug Print 769
- DBM. - DeBugMode 765
- MBX. - MessageBox 2036
- SAY. - Say something 2048

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.14 SAY. - Say something

MiniRobotLanguage (MRL)

SAY. Command

Say something

Intention

This command will make the robot speak. And More. Let's start with an easy example.

```
: $$TXT=With a couple of simple program statements, you can
add
: $$TXT=$$TXT speech to your programs.
GUN.$$USR
ALB.$$USR|$$USR
: $$TXA=Hello $$USR, $$TXT
'
SAY.Look Mum, he is talking!{BREATH}
'
' The |w makes the robot wait until he has finished talking.
' Without the |w he would talk in the background.
'
SAY. $$TXA|w
ENR.
```

Per default all Speech runs in the background, while your script continues to do its job.
We have therefore specified the "`|w`"-Flag in this example.Because otherwise the script would end and you would not hear anything. **The talking stops, when the script ends.**

Alternatively you can have the robot talk while your script is running. In this case the talking happens in the background. Here is an example that show different uses.

```
SAY. This is a countdown!
PAU.1
FOR.$$CNT|10|3|-1
  SAY.$$CNT|w
NEX.
SAY.2
SAY.1
SAY.Now things can happen.|w
ENR.
```

The `SAY.` - command has multiple options. These are:

- Speak faster or slower using the `{SPEED x}` Tag. ***x*** is a number between -10 and 10, while 0 is default, normal Speed.
- Wait until Speech has finished, using the {WAIT} Tag without a number

*© 2023 Theo Gottwald * https://www.smart-package.com*

- Wait until speech has finished or a timeout is reached using the {WAIT ***x***}-Tag. Where x is a number that specifies the timeout in seconds or fractions of a second (like 0.5).
- Stop actual Speech using the {STOP}-Tag
- Repeat parts of the Saying using the {DO} and the {REPEAT ***x***}-Tag. Where x is the number how often you want it to be repeated. DO/REPEAT can be nested to unlimited depth and can contain other tags inside the text that is to be repeated.
- Let the voice make a small rest, using the {BREATH} Tag
- Let the voice make a rest for a specified time in seconds, using the {BREATH ***x***}–Tag. Where x is a number that specifies the time to rest in seconds or fractions of a second (like 0.5).
- Set the Mode, if the voice should "dictate" comma and punctuation or not. Using the {EXPAND x}-Tag. Where x is 0 to not expand punctuation, and 1 means that anything will be dictated.
- The {VOICES}-Tag will show you all installed voices that can be selected for Speaking.
- The {VOICE x}-Tag will select the Voice that the robot will talk with.
- The {PRIO x} Tag will set the Priority of the things to say. If there are multiple sentences in the Que, Sentences with a higher Priority will be said first and"overtake" other sentences in the Que. x is a number between 0 (default) and 2 (highest Priority).
- The {FILE}-Tag will switch into "Reciting Mode". It will have the Speech to recite a file for you.
- The {LINK}- Tag will have the Robot to read a WEB- or Intranet Site for you. Better don't use this one, as it reads the source-code of the site. This does really only make sense with special designed user-sites (intranet).
- Prefer using the {RECITE}-Tag. It will really read that article on a WEB-Site that is actually located.

Here are some examples:

Example 1: Say something and wait.

```
'*****************************************************
' EXAMPLE 1: SAY.
'*****************************************************
' Choose speaker voice
SAY.{voice Sam}
'
' Make robot not to speak Punctuation
SAY.{EXPAND 0}
'
' Set Speed to 1, that is a bit faster then normal
{SPEED 1}
'
' Say something .. and wait until all speech has been
finished.
SAY. Did you get me?{wait}
'
' Take an additional Breath ....
SAY.{BREATH}
'
SAY.OK!{BREATH}OK?{WAIT}
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
ENR.
```

Example 2: Show installed voices.

This example will show you which voices are currently installed on your Computer. You find these voices inside the variable `$$TXT`.

```
'*****************************************************
' EXAMPLE 2: SAY.
'*****************************************************
' Which Voices are installed?
' Print a List with all installed voices
SAY. {voices}|$$TXT
DBP.$$TXT
ENR.
```

Example 3: Check if a specified voice is installed.

To check whether a voice is installed or nut, just use the `IVC.`-command for this variable.

```
'*****************************************************
' EXAMPLE 3: SAY.
'*****************************************************
' Check whether a defined Voice is available or not
'
SAY. {voices}|$$TXT
IVC.$$TXT=Sam
  SAY. Sam is available|w
ELS.
  SAY. Sam is not available|w
EIF.
ENR.
```

Example 4: Recite a file.

```
'*****************************************************
' EXAMPLE 4: SAY.
'*****************************************************
'  You need a file name "text.txt" containing
' an english text on your desktop
'
: $$TXT=?desktop\test.txt
'
' Choose speaker voice
SAY.{voice Sam}
'
' Make robot not to speak Punctuation
SAY.{EXPAND 0}
'
' Set Speed to 1, thats a bit faster then normal
{SPEED 1}
'
' Read the content of that file
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SAY.{FILE 1}$$TXT|w
'
' Say something .. and wait until all speech has been
finished.
SAY. Did you get me?{wait}
ENR.
```

Example 5: Repeat parts of your text.

This example shows the use of the DO/REPEAT tags. This way you can inline make the Speaker repeat things thousand times without the need to write them thousand times.
Is your computer a blue-screen sinner? Have it repeat that prayer hundred times ... using DO/REPEAT. Don't forget to let it take a {BREATH} after that!

```
'*************************************************
' SAY.-Sample 5: DO / REPEAT / BREATH Tags
'*************************************************
: $$TXT=Per default {DO}all Speech runs in the background,
{BREATH}{REPEAT 2} while your script continues to do its job.
: $$TXT=$$TXT {BREATH} Using the Tags "DO" and "REPEAT"
enables you to inline repeat things multiple times!{wait}
'
SAY.{SPEED 2}$$TXT|w
ENR.
```

Example 6: Make it important.

The following example shows you how you can stop a on going speech using the STOP-Tag.
Also it shows how a PRIO 2-Speech will be spoken before a PRIO zero speech has ended.

```
'-------------------------------------------------------
' SAY.-Sample 6: PRIO / STOP
'-------------------------------------------------------
: $$TXT=Per default {DO}all Speech runs in the background,
{REPEAT 5} while your script continues to do its job.
: $$TXT=$$TXT Using the Tags "DO" and "REPEAT" enables you to
inline repeat things multiple times!
'
SAY.{SPEED 2}$$TXT
PAU.2
SAY.{PRIO 2}We stop it now!{wait} {STOP}
ENR.
```

Example 7: Recite a text from A WEB-site or dialog.

The following example shows you how you can make the robot to read a part of a WEB-site.
You just need to set the Actual Point (see MAP. - Move Active Point 2483).
The Robot will try to read the text at that place.

© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
' SAY.-Sample 7: RECITE
'-------------------------------------------------------------
' Will not really work because the news pages change often!
' This is just an example
'
' Locate the new-web-site in the Internet Explorer
' Don't try with Google Chrome or Opera, this will not work!
'
STW.ct|IEFrame|FOCUS Online - Nachrichten - Windows Internet
Explorer
SCW.nct|1|Frame Tab|{&NOTEXT:}
SCW.nct|1|TabWindowClass|FOCUS Online - Nachrichten - Windows
Interne
SCW.nct|1|Shell DocObject View|{&NOTEXT:}
SCW.nct|1|Internet Explorer_Server|{&NOTEXT:}
'
' Locate the article on the WEB-Site
'
SAO.rn|42|Christian Wulff
'
' Now have the robot read it for you.
SAY.{RECITE}|w
ENR.
```

Important Hint:
The `{File}` and `{LINK}` Tag switch the Speech into the selected Mode. After completion, they will automatically be switched off.

Important:

When the Robot program ends, speech will also be stopped. Therefore use the `{wait}` tag to make the robot wait for the end of his speech.

In case you do not hear anything, check your sound hardware and system sound settings. `SAY.` will only work, if your computer has working sound-hardware and the system settings for sound output must be made that you will be able to hear something.

If you want to select between different voices and languages, see the **remarks**-section below.

Syntax

SAY.[P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - (optional) Any text or variables you want the robot to say. If omitted the TOS is taken.

P2 - (optional) can be:

- **"w"**, if specified the robot will wait until things have been said. If omitted, talking will be in the background.

- **"r"**, if specified the robot will talk in RAW mode and ignore any commands in { ...}. Active Flags, Options like the speed and the selected voice will remain active.

Example

```
'*****************************************************
' EXAMPLE: SAY.
'*****************************************************

'  You need a file name "text.txt" containing
' an english text on your desktop
'
: $$TXT=?desktop\test.txt
' Choose speaker voice
SAY.{voice Sam}
' Read the textfile
SAY.{EXPAND 0}{SPEED 1}{FILE 1}$$TXT|w
SAY. Did you get me?{wait}
ENR.
```

Remarks

Internally SAY. uses the Windows TTS (Text-To Speech Engine). Per default it may support only one voice, thats the gold old "Microsoft Sam". Under Windows 7, it may be another voice. Now, how can you find out which "voices" are available on your system?

Use this script:

```
' Which Voices are installed?
SAY. {voices}|$$TXT
DBP.$$TXT
ENR.
```

What you see is maybe just one speaker/voice, for example (WinXP):

```
01  Microsoft Sam
```

and he has a pure english voice that may really sound a bit like an robot.

You can however make your computers speak german and french as well and get other voices. Amongst other offers, you can install the MS-Reader:

© 2023 Theo Gottwald * https://www.smart-package.com

1. MS-Reader Downloadlink

and then install the

2. Microsoft Text-to-Speech Package

There are multiple available for different languages. You can install all of these to get multiple additional speakers.

Now if you run the same script again, it may look somehow like this:

```
01  Microsoft Sam
02  LH Pierre
03  LH Stefan
04  LH Veronique
05  LH Anna
```

Now you can choose another voice using the voice of Stefan:

```
' Select Voice
SAY.{voice 03}
SAY."Hallo Matthias!{wait}"
ENR.
```

using the number 03 or you can choose by name:

```
' Select Voice
SAY.{voice Stefan}
SAY."Hallo Matthias!{wait}"
ENR.
```

Note that when choosing by name, that the second name is used, the first name must not be specified. "{`voice LH Stefan`} will not work. You have to drop the "`LH`" and just write `{VOICE Stefan}`.

Limitations:

-

See also:

- DBP - Debug Print 769
- DBM. - DeBugMode 765
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775

© 2023 Theo Gottwald * https://www.smart-package.com

3.56.15 SBT. - Show Bubble Text

MiniRobotLanguage (MRL)

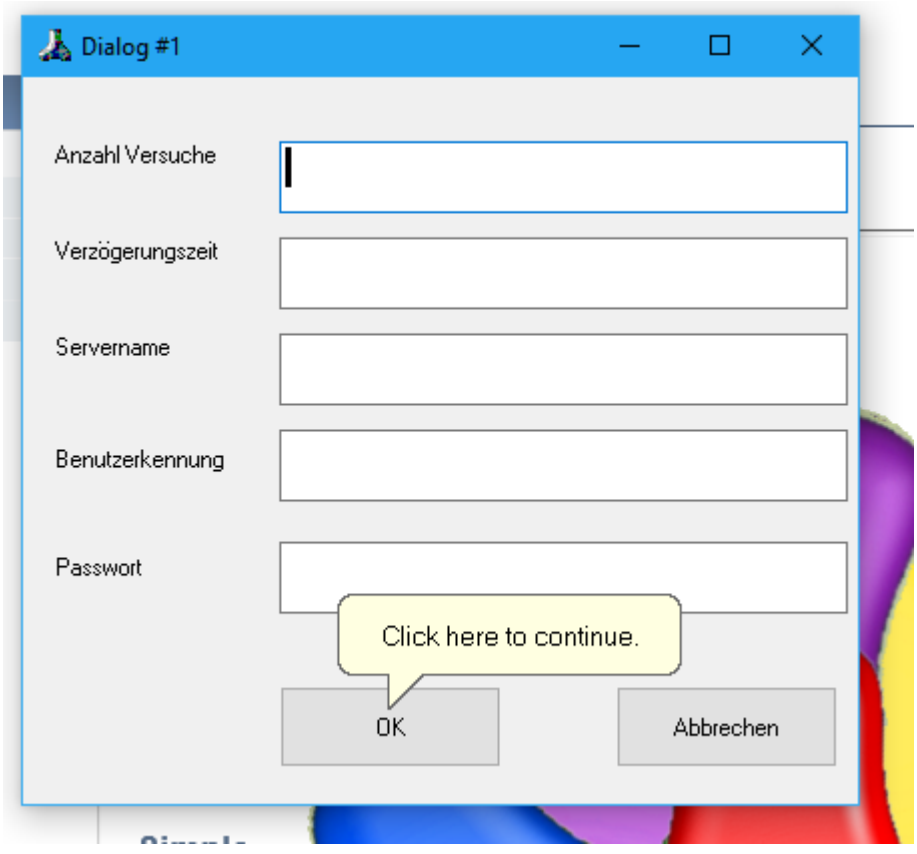
Show Bubble Text

Show Help Text "in a Bubble"

Intention

This command is used to display textual Information that can reference a control, or a point on Screen.
You can influence where the Bubble is displayed as well as the Text-Size.
You can also have that small "comic-arrow" or not. Let's see some examples.

```
' First we locate a Button
STW.ct|PBWindowClass:0|Dialog #1
SCW.nict|1|1009|Button|OK
' Next we how the Bubble-Text
SBT.Click here to continue.
' We need to rest, as the text runs in the Background
PAU.8
ENR.
```



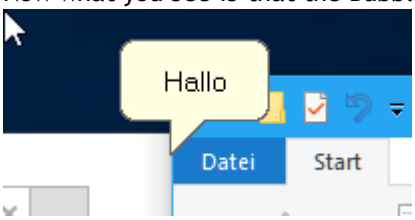
*© 2023 Theo Gottwald * https://www.smart-package.com*

Why do we need the "PAU.8" ?
It's needed because the Command - per default runs in the Background.
It starts to display the Bubble-Text, and WHILE the text is displayed the Script continues to run.
Now, if the Script ends - the Bubble will also be closed!

To prevent this you can add a second Parameter:

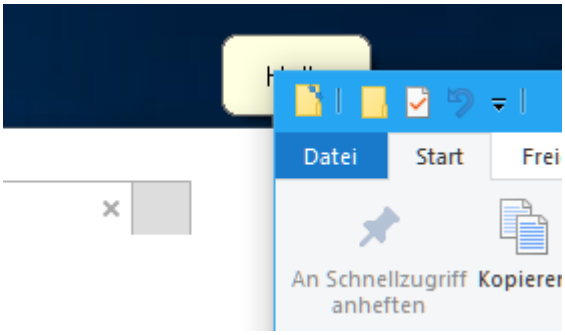
```
STW.ct|PBWindowClass:0|Dialog #1
SCW.nict|1|1009|Button|OK
SBT.Click here to continue.|#
ENR.
```

In this case you can leave away the "PAU.8" because the Bubble will stay for 6 Seconds (this is the default time, if no other time is specified).
Now what you see is that the Bubble is always "on Top". That's a default setting.



If you do not want that use that other Parameter "@":

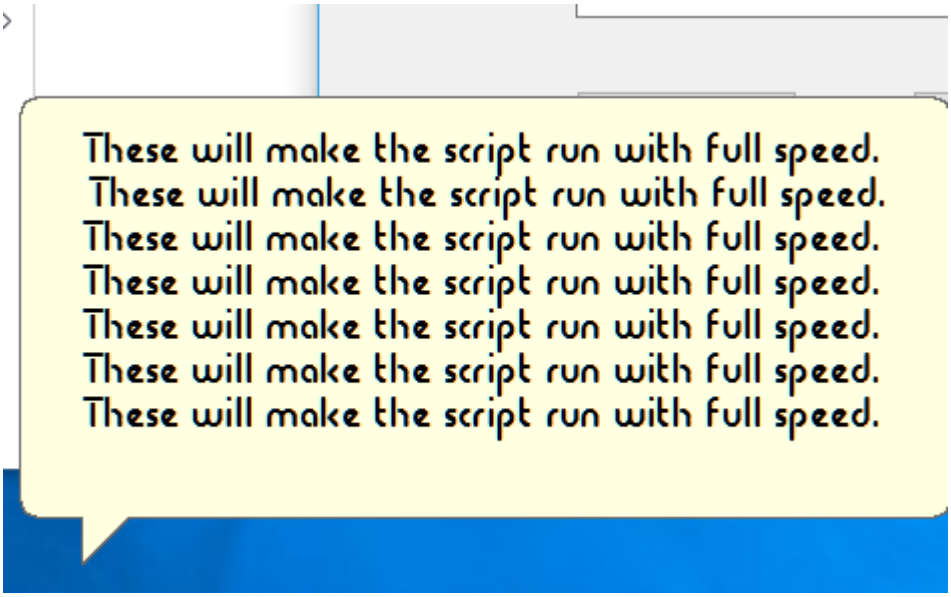
```
STW.ct|PBWindowClass:0|Dialog #1
SCW.nict|1|1009|Button|OK
SBT.Click here to continue.|#@
ENR.
```



Here is another example that uses a custom FontName:

```
$$XPO=-100
$$YPO=100
$$PAR=FS:14 t:10 cs:1 ax:$$XPO ay:$$YPO vc:1 FN:"Tomorrow People" #
```


© 2023 Theo Gottwald * https://www.smart-package.com

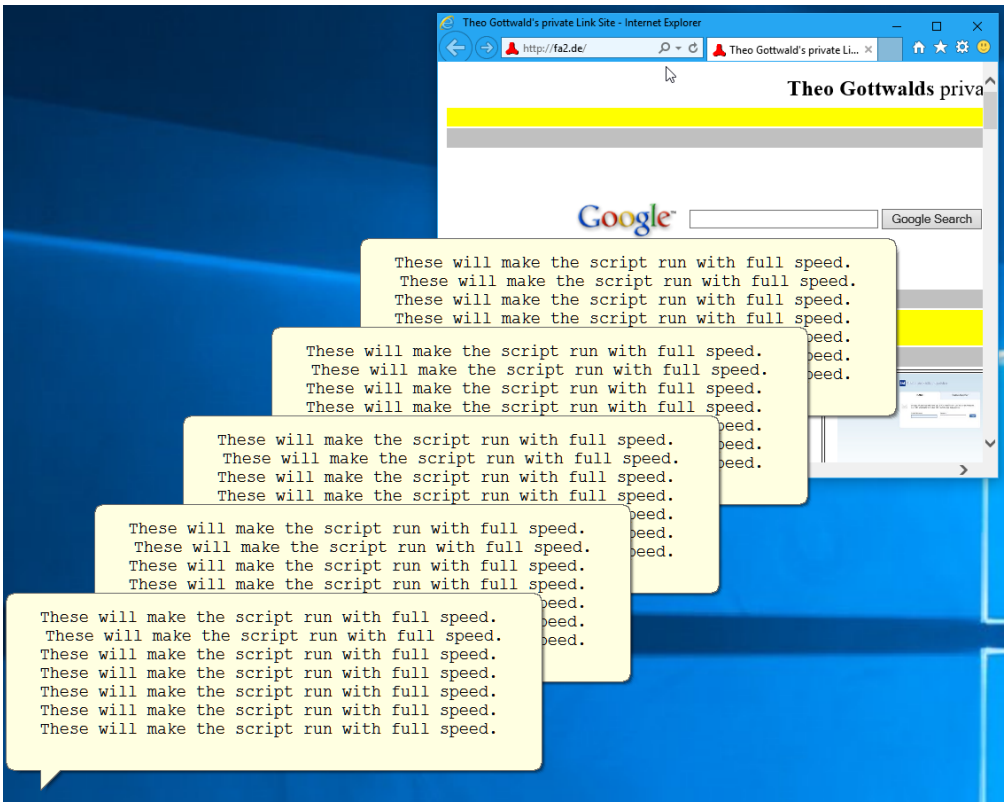


Following is a more complex Sample.

```
' More complex Sample
STW.ct|IEFrame| - Internet Explorer

$$TXT=These will make the script run with full speed. $crlf$
$$TXT=$$TXT $$TXT$$TXT$$TXT$$TXT$$TXT$$TXT

FOR.$$CNT|1|5
  CAL.$$XPO=(-120*$$CNT)
  CAL.$$YPO=100*$$CNT
  $$PAR=FS:14 t:10 cs:1 ax:$$XPO ay:$$YPO vc:1
  SBT.$$TXT|$$PAR
NEX.

PAU.40
ENR.
```

Which will produce this picture:

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

SBT.P1[|P2][#@]

Parameter Explanation

P1 - Text to be displayed

This can be a Text or a Text-Variable that will be displayed in the Bubble-Text.
It may contain $crlf$ - these will be treated as NewLine".

P2 - (optional) Flags and Parameters

@ - will prevent the "TOPMOST-Flag".

 # - will have the Script wait until the Bubble-time is over.

`T:(NUM)` - Time in Seconds, specify how long the Bubble-Text shall be displayed. The Bubble will anyway be closed when the Script ends! If not given the default time is 6 Seconds.
`FS:(NUM)` - FontSize: An number that specifies the size of the Font that shall be used.
`FN:"FONTNAME"` - Specify any Font that is on your Computer. Notice that the Fontname must be between " ".
`VC:(NUM)` - Vertical Center. The given number can be 0,1 or 2. Each number will have a different effect. 0 is the default.
`HC:(NUM)` - Horizontal Center. The given number can be 0,1 or 2. Each number will have a different effect. 0 is the default.

© 2023 Theo Gottwald * https://www.smart-package.com

AX:(NUM) - Adjusting X-Position. This value can be negative or positive. It will move the Bubble by this number of pixels left (if negative value) or right.

AY:(NUM)- Adjusting Y-Position. This value can be negative or positive. It will move the Bubble by this

MP: - Mouse Position. If specified then the Bubble will be generated at the Position where the Mouse Pointer is.

LA: - Left Align. If not specified the Text is always Centered in the Bubble.

CS: - Center on Screen. This will Center the Bubble on Screen. Use this if you have no Window or Button.

NC: - NoCorner. Will remove the "Comic-Arrow".

WB:(NUM) - WordBreak. Switches Wordbreak on. NUM is a numerical value, for example 32.

Using 32 will make the Text to appear in a Single line without Newlines. Other values may have other effects depending on the situation.

Example

```
'***********************************************
' SBT. - Sample
'***********************************************

  STW.ct|PBWindowClass:0|Dialog #1
  SCW.nict|1|1009|Button|OK
  SBT.Click here to continue.|#@
  ENR.

'***********************************************
  $$PAR=FS:12 t:10 cs:1 ax:$$XPO ay:$$YPO vc:1  @ wb:64
  $$TXT=Hallo All!
  SBT.$$TXT|$$PAR

'***********************************************

' More complex Sample
STW.ct|IEFrame| - Internet Explorer

$$TXT=These will make the script run with full speed. $crlf$
$$TXT=$$TXT $$TXT$$TXT$$TXT$$TXT$$TXT$$TXT

FOR.$$CNT|1|5
  CAL.$$XPO=(-120*$$CNT)
  CAL.$$YPO=100*$$CNT
    $$PAR=FS:14 t:10 cs:1 ax:$$XPO ay:$$YPO vc:1 FN:"Tomorrow
People" @
  SBT.$$TXT|$$PAR
NEX.

PAU.40
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- SHT. - Show Text 2061
- DBM. - DeBugMode 765
- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- DMP. - Dump internat Informations 775
- USM - Use silent Mode 786
- **OPT. - Optional Settings** 2406
- PAU. - Pause 783

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.56.16 SHT. - Show Text

MiniRobotLanguage (MRL)

Show Text

Show Textual Information to user

Intention

This command is used to display textual Information. You can influence where the Bubble is displayed as well as the Text-Size and Colors.
The following Script will display a "Hello World" on the upper right side of your screen. This way you can easily message something to your users.

```
' Let's start simple
  SHT.Hello World.
  PAU.8
  ENR.
```

Why do we need the "`PAU.8`" ?
It's needed because the Command - per default runs in the Background. It starts to display the Text, and WHILE the text is displayed the Script continues to run.
Now, if the Script ends - the Bubble will also be closed!

To prevent this you can add a second Parameter:

```
' Let's start simple
  SHT.Hello World.|#
  ENR.
```

This will make the Script wait until the message goes away. The default time for the message to disappear is **6 Seconds**.
How can we change that? We add another Parameter:

```
  SHT.Hello World.|# T:12
  ENR.
```

Now the Text will disappear after 12 **Seconds.** What else can we change?
We can change: **Colors, Text-Size and Font and Boldness** and also the **Position.**

To do this, the Parameters as shown below can be given.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
SHT.Click here to continue.|DOT: t: 5 BF:4 fs:115  fn:"Tomorrow People" cs:1
```

```
SHT.Click here to continue.|DOT: t: 5 BF:4 fs:115  fn:"Tomorrow People" cs:1 bc:
PAU.10
ENR.
```

Using other colors and custom fonts will produce this result. Notice the "Dot" on the left side?
It's from the "`DOT:`" Parameter.

```
' More compelx Sample
$$TXT=RED ALERT. $crlf$ Honeypot was hacked!!$crlf$Please check files.

FOR.$$CNT|1|5
  DIT.#
  $$PAR=FS:40 t:1 cs:1 bc:"0000F0"# lc:"00FF00"
  SHT.$$TXT|$$PAR
  DIT.2
NEX.
PAU.2
ENR.
```

Syntax

SHT.P1[|P2][#]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

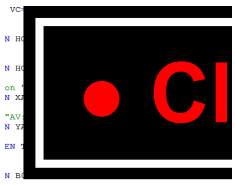
P1 - Text to be displayed

This can be a Text or a Text-Variable that will be displayed in the Bubble-Text. It may contain $crlf$ - these will be treated as NewLine".

P2 - (optional) Flags and Parameters

 # - will have the Script wait until the Bubble-time is over.

`T:(NUM)` - Time in Seconds, specify how long the Bubble-Text shall be displayed. The Bubble will anyway be closed when the Script ends! If not given the default time is 6 Seconds.
`FS:(NUM)` - FontSize: An number that specifies the size of the Font that shall be used.
`FN:"FONTNAME"` - Specify any Font that is on your Computer. Notice that the Fontname must be between " ".
`VC:` - Vertical Center. This will Center the Text vertically.
`HC:` - Horizontal Center. This will Center the Text horizontally.
`AX:(NUM)` - Adjusting X-Position. This value can be negative or positive. It will move the Text by this number of pixels left (if negative value) or right.
`AY:(NUM)` - Adjusting Y-Position. This value can be negative or positive. It will move the Text by this nu
`MP:` - Mouse Position. If specified then the Text will be generated at the Position where the Mouse Pointer is.
`CS:` - Center on Screen. This will Center the Text on Screen.
`BF:(NUM)` - **Bold-Faktor**, Number 0-9. The higher the number - the bolder the Text.
`DOT:1` - Will Paint an DOT at the left of the Text.



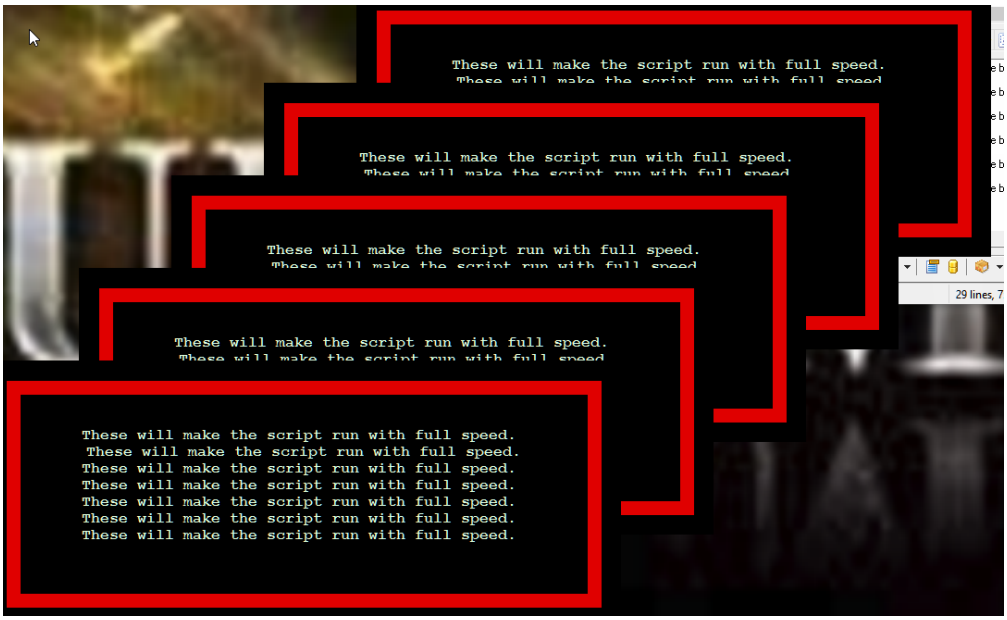
`BC:` - Background-Color for Text must be in Hexadezimal and in "". Format is: `BBGGRR`
Example: `BC:"000000"`
`LC:` - Line-Color for Surrounding Box must be in Hexadezimal and in "". Format is: `BBGGRR`
Example: `LC:"0000FF"`
`TC:` - Text-Color for Text must be in Hexadezimal and in "". Format is: `BBGGRR`
Example: `TC:"FFFFFF"`

Example

```
'***********************************************
' SBT. - Sample
'***********************************************

  SHT.Click here to continue.|DOT: t: 5 BF:4 fs:115  fn:"FUTURA BOLD" HC:1 VC:1 #
  ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'************************************************

'************************************************

' More complex Sample
$$TXT=These will make the script run with full speed. $crlf$
$$TXT=$$TXT $$TXT$$TXT$$TXT$$TXT$$TXT$$TXT

FOR.$$CNT|1|5
  CAL.$$XPO=(-100*$$CNT)
  CAL.$$YPO=100*$$CNT
  $$PAR=FS:12 t:10 cs:1 ax:$$XPO ay:$$YPO
  SHT.$$TXT|$$PAR
NEX.
PAU.2
ENR.
```



Remarks

-

Limitations:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

See also:

- SBT. - Show Bubble Text 2055
- DBM. - DeBugMode 765
- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- DMP. - Dump internat Informations 775
- USM - Use silent Mode 786
- **OPT. - Optional Settings** 2406
- PAU. - Pause 783

© 2023 Theo Gottwald * https://www.smart-package.com

3.57 Internet and Network

3.57.1 ! Smarty's Internet and Network Commands

Smarty's Internet and Network Commands **Previous** 2048 **Top** 41 **Next** 2067

Smarty's Internet and Network Commands

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.57.2 CFU. - Copy-From-URL

MiniRobotLanguage (MRL)

CFU. Command

Copy from Url

Intention

Reads a file from a WEB-URL via http. Can be used to test WEB-Sites and downloads. Can simulate different Browsers.

You may replace special-characters by a hexadecimal code to make them better readable. Spaces would to be %20 for example. Files that are hidden behind Pearl-Scripts can not be downloaded via `CFU`.

Alternative commands are: `DLF` or `DLV` for normal usage.

Syntax

CFU.P1[|P2[|P3][|P4][|P5][|P6]

Parameter Explanation

P1 – URL to read from WEB. Must be a URL ending with a filename.
P2 – ***(optional)*** Variable for the downloaded Data. If omitted, stack is used.
P3 – ***(optional)*** Timeout in Seconds.
P4 – ***(optional)*** Referer-String.
P5 – ***(optional)*** Set Browser-Type (numeric 1... 20)
P6 – ***(optional)*** Site-URL (Example: "www.fa2.de"). Only for special usage.

```
 ' Result auf Stack
 CFU.$$URL

' Result in $$RET
 CFU.$$URL|$$RET

' Result in $$RET + Timeout von $$TIM
 CFU.$$URL|$$RET|$$TIM

' plus Referer String aus$$REF
 CFU.$$URL|$$RET|$$TIM|$$REF

' plus Browser-Identification from $$BNR
 CFU.$$URL|$$RET|$$TIM|$$REF|$$BNR

' Leave a site-URL
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
CFU.$$URL|$$RET|$$TIM|$$REF|$$BNR|$$SIR
```

P5 - Browser-Types (User Agent Identification :

1. "MiniRobot PR0."
2. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)"
3. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; DT)"
4. "" (None - empty)
5. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en) Opera 8.0"
6. "msnbot/1.0 (+http://search.msn.com/msnbot.htm)"
7. "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
8. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en) Opera 8.01"
9. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; de) Opera 8.01"
10. "Mozilla/5.0 (Windows; U; Win98; hu-HU; rv:1.7.6) Gecko/20050318 Firefox/1.0.2"
11. "Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)"
12. "Mozilla/5.0 (Windows; U; Windows NT 5.0; sl-SI; rv:1.7.8) Gecko/20050511 Firefox/1.0.4"
13. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"
14. "YahooSeeker/1.2 (compatible; Mozilla 4.0; MSIE 5.5; yahooseeker at yahoo-inc dot com ; http://help.yahoo.com/help/us/shop/merchant/)"
15. "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; Win 9x 4.90)"
16. "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR"
17. Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) Chrome/0.A.B.C Safari/525.13
18. Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.0.10) Gecko/2009042316 Firefox/3.0.10
19. Mozilla/5.0 (X11; U; Linux i586; en-US; rv:1.7.3) Gecko/20040924 Epiphany/1.4.4 (Ubuntu)
20. Opera/9.80 (Macintosh; Intel Mac OS X; U; en) Presto/2.2.15 Version/10.00

Example

```
'**********************************
' CFU.-Test
'**********************************
: §§URL=http://www.it-berater.org/TheoDownloads/allinone cthm-map-pack 10.exe
VAR.§§FIL=?desktop\Map_Pack.exe
VAR.§§TIM=#dtime#
CFU.§§URL|§§RET
LEN.§§RET|§§LEN
PRT.§§LEN - Time: #dsince#
CTF.§§FIL|§§RET
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.
END.
```

Remarks

-

Limitations:

-

See also:

- DLF. - DownLoad-File 2074
- DLV. - Download-to-Variable 2076
- URL. - Split URL 3234

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.3 CNS. - Connect Net Share

MiniRobotLanguage (MRL)

CNS. Command

Connect Network Share

Intention

Can be used to makes a connection to a network resource and can redirect a local device to the network resource.

Just use the command and specify which Mapping you want:

```
CNS.\\MyComputer\Folder$

CNS.\\MyComputer\Folder$|MyLogin|MyPassword

' Make permanent connection
CNS.\\MyComputer\Folder$|MyLogin|MyPassword|1

' Make connection only until reboot (temporary)
CNS.\\MyComputer\Folder$|MyLogin|MyPassword|1
```

In case of error, the timeout-flag is set and the error-number is on the TOS.

Hint: Use the RDM. - Remove Drive Mapping 2112 command to remove a mapped drive.

Syntax

CNS.P1[|P2][|P3][|P4][|P5]

Parameter Explanation

P1 - Share to connect, example: ***\\computer\folder***

P2 - *(optional)* ***Username***

P3 - *(optional)* ***Password***

P4 - *(optional)*. 1/0 -If 1 is specified, the connection is permanent.
It will be restored at the next reboot.
If 0 or nothing is specified, the connection is temporary
and not saved in the user profile.

P5 - (optional). Variable that will return the result of the operation. In case of success the result will be zero. In case of error an error number is returned.

© 2023 Theo Gottwald * https://www.smart-package.com

If omitted TOS is used.

5 - ERROR_ACCESS_DENIED
The caller does not have access to the network resource.
This could also be a access-rights problem.

85 - ERROR_ALREADY_ASSIGNED
The local device specified is already connected to a network resource.

1200 - ERROR_BAD_DEVICE
The value specified is invalid.

67 - ERROR_BAD_NET_NAME
The value specified is not acceptable to any network resource provider because the resource name is invalid, or because the named resource cannot be located.

1204 - ERROR_BAD_PROVIDER
The value specified does not match any provider.

1223 - ERROR_CANCELLED
The attempt to make the connection was canceled by the user through a dialog box from one of the network resource providers, or by a called resource.

1208 - ERROR_EXTENDED_ERROR
A network-specific error occurred.

487 - ERROR_INVALID_ADDRESS
The caller passed a pointer to a buffer that could not be accessed.
This error should not appear, as it is an internal error.

87 - ERROR_INVALID_PARAMETER
This error is a result of providing wrong parameters, or the connection is to a network that requires the redirecting of a local device. Please check the variables.

86 - ERROR_INVALID_PASSWORD
The specified password is invalid.

234 - ERROR_MORE_DATA
Internal Error. If this error is returned, then no connection has been made.

259 - ERROR_NO_MORE_ITEMS
The operating system cannot automatically choose a local redirection because all the valid local devices are in use.

1202 - ERROR_DEVICE_ALREADY_REMEMBERED
The local device name has a remembered connection to another network resource.

1203 - ERROR_NO_NET_OR_BAD_PATH
The network path was either typed incorrectly, does not exist, or the network provider is not currently available. Please try retyping the path or contact your network administrator.

*© 2023 Theo Gottwald * https://www.smart-package.com*

1222 - ERROR_NO_NETWORK
The network is unavailable.

1219 - ERROR_SESSION_CREDENTIAL_CONFLICT
An attempt was made to establish a session to a network server, but there are already too many sessions established to that server.

1326 - ERROR_LOGON_FAILURE
Logon failure: user account restriction. Possible reasons are blank passwords not allowed, logon hour restrictions, or a policy restriction has been enforced.

Example

```
'-------------------------------------------------------------
' CNS. - No Pasword
'
$$PAT=\\XDRIVE\Daten$
CNS.$$PAT
ITO.
  PRT.error happend
EIF.
DMP.6
MBX.!
ENR.

' CNS. - with Pasword
'
$$PAT=\\XDRIVE\Daten$
CNS.$$PAT|Admin|Asjfds_55
ITO.
  PRT.error happend
EIF.
DMP.6
MBX.!
ENR.
```

Remarks

If the command fails, the Timeout-Flag is set, otherwise it will be cleared.

Limitations:

-

See also:

- MND. - Map Network Drive 2108

© 2023 Theo Gottwald * https://www.smart-package.com

- **RDM. - Remove Drive Mapping** 2112
- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- IVA. - If Volume available 1053

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.4 DLF. - DownLoad-File

MiniRobotLanguage (MRL)

DLF. Command

Download File

Intention

This command can be used to download a File or a WEB-Site from Internet to a local Filename. It an be used for all sorts of links.

Syntax

DLF.[P1][|P2[|P3][|P4]

Parameter Explanation

P1 - URL
P2 - Output Filename
P3 - (optional out) Variable for Success-flag 1/0.*
P4 - (optional in) Password for Download

Example

```
'**********************************
' DLF.-DEMO
'**********************************
VAR.$$FIX=?desktop\testfile2.zip
PRT.-------------------------
' We work from Variable to Stack if one Parameters are
specified.
' This is the filename of the downloaded file
PUS.$$FIX
' This is the URL of the file to download
VAR.$$URL=http://www.it-berater.org/download/down.pl?ID=1
DMP.4
DLF.$$URL
DMP.4
IVV.($tos$>800000)
  GSB.del
ELS.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  MBX. Could not download that file. File on Desktop contains
error message.
EIF.
PRT.-------------------------
' We work from VAR to VAR if two Parameters are specified.
Resulting Bytes are left on stack
' This is the URL of the file to download
VAR.§§URL=http://www.it-berater.org/download/down.pl?ID=1
DLF.§§URL|$$FIX
GSB.del
DMP.4
PRT.-------------------------
' We work from VAR to VAR if two Parameters are specified.
' This is the URL of the file to download
VAR.§§URL=http://www.it-berater.org/download/down.pl?ID=1
DLF.§§URL|$$FIX|§§BYT
PRT. Bytes loaded: §§BYT
PRT.-------------------------
DMP.
MBX.Ready
END.

:del
MBX. File is on your desktop - will be deleted now.
DEL.$$FIX
RET.
```

Remarks

-

Limitations:

-

See also:

- DLV. - Download-to-Variable 2076
- CFU. - Copy-From-URL 2067
- URL. - Split URL 3234

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.5 DLV. - Download-to-Variable

MiniRobotLanguage (MRL)

DLV. Command

Download to Variable

Intention

This command can be used to download a File or a WEB-Site from Internet to stack or variable. It an be used for all sorts of links.

Internally its identical to `DLF.`, just that it downloads to a variable and not to a file.

The timeout flag 334 is set according to success of operation.

Syntax

DLV.[P1][|P2[|P3][|P4]

Parameter Explanation

P1 - URL
P2 - ***(optional)*** Output Filename - if omitted is taken from TOS
P3 - (optional out) Variable for Success-flag 1/0.*
P4 - (optional in) Password for Download

Example

```
''***********************************
' DLV.-DEMO
'***********************************
PRT.-------------------------
' We work from Variable to Stack if one Parameters are
specified.
' This is the URL of the file to download
VAR.§§URL=http://www.it-berater.org/download/down.pl?ID=1
DLV.§§URL
DMP.4
STS.CLEA
PRT.-------------------------
' We work from VAR to VAR if two Parameters are specified.
Resulting Bytes are left on stack
' This is the URL of the file to download
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
VAR.§§URL=http://www.it-berater.org/download/down.pl?ID=1
DLV.§§URL|§§FIL
LEN.§§FIL|§§LEN
PRT.Downloaded File is: §§LEN Bytes.
DMP.4
PRT.-------------------------
' We work from VAR to VAR if two Parameters are specified.
' This is the URL of the file to download
VAR.§§URL=http://www.it-berater.org/download/down.pl?ID=1
DLV.§§URL|§§FIL|§§BYT
PRT. Bytes loaded: §§BYT
PRT.-------------------------
DMP.4
DMP.
MBX.Ready
END.
```

Remarks

-

Limitations:

-

See also:

- DLF. - DownLoad-File 2074
- CFU. - Copy-From-URL 2067
- URL. - Split URL 3234

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.6 FTP. - and more

3.57.6.1 FTP. - File Transfer Protokoll

MiniRobotLanguage (MRL)

FTP. Command

Dowload Files via FTP and More Internet Commands

Intention

This is a multi-purpose command.

This command can be used to
- download one or more File(s) via FTP to your local computer.
- test a download link if it works, without downloading the complete file.
- test a download link that uses a password, if it works, without downloading the complete file.

For a detailed description see the help pages of the subcommands.

Syntax

FTP.[Px...Px]

Parameter Explanation

Parameters depend on the used subcommand P1.
The following subcommands are available:

`"tdl","test download"`
-> Test a download-Link on any WEB-Site if it works.
For Details see here: tdl - Test Download 2078

`"tpw","test with password"`
-> Test a download-Link on a WEB-Site if it works. Using Login and Password.
For Details see here: tpw - test with password 2078

`"ftp","ftp download"`
-> Load one or many files down from a FTP-Server.
Can use a wildcard like "`*.jpg`"
For Details see here: fto - file transfer protocol 2080

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'**********************************
' FTP.-DEMO
'**********************************
$$URL=fa2.de
$$LOG=myftplogin
$$PWD=343retew&$dJ
$$PAT=site
$$FIM=*.*
$$LOC=?path\test\

FTP.ftp|$$URL|$$LOG|$$PWD|$$PAT|$$FIM|$$LOC|$$REA|$$REB|$$REC
PRT.----------------
PRT.REA=$$REA
PRT.----------------
PRT.REB=$$REB
PRT.----------------
PRT.REC=$$REC
PRT.----------------
MBX.Ready
ENR.
```

Remarks

-

Limitations:

-

See also:

- DLV. - Download-to-Variable 2076
- CFU. - Copy-From-URL 2067
- URL. - Split URL 3234
- fto - file transfer protocol 2080
- tdl - Test Download 2078
- tpw - test with password 2078

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.6.2 fto - file transfer protocol

MiniRobotLanguage (MRL)

FTP.ftp download
FTP.ftp

Dowload Files via FTP to a local folder

Intention

This command can be used to download a File or multiple files via the FTP-Protocol to your computer. You can use Login and Password and you can use wildcards to define which files you want to download.

```
' The ftp-server
$$URL=ftp-server.de

' Your Login
$$LOG=your_login
' Your Password
$$PWD=the-password
' The folder on the ftp-derver
$$PAT=site-folder

' File-Mask. Define which file(s) you want to download
' using wildcards: ?/*
$$FIM=*.*

' The local path where you want these files
' to downloaded to
$$LOC=?path\test\

FTP.ftp|$$URL|$$LOG|$$PWD|$$PAT|$$FIM|$$LOC
ENR.
```

The command does also give you a return on how things worked out.
For this you can specify three additional variables. Like this:

```
FTP.ftp|$$URL|$$LOG|$$PWD|$$PAT|$$FIM|$$LOC|$$REA|$$REB|$$REC
PRT.----------------
PRT.REA=$$REA
PRT.----------------
PRT.REB=$$REB
PRT.----------------
PRT.REC=$$REC
PRT.----------------
MBX.!
ENR.
```

The result could look like in teh picture below.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P7/$$REA shows how many files have been successfully transferred.
P8/$$REB shows you a textual Message if there were fatal Errors.
P9/$$REC contains a detailed list with all FTP transfers that were successful or not.
P9 will also list folders and files that can not be transferred due to right or other problems. Files that have been copied and counted will be in small letter and prefixed with a "OK:". Files that could not be transferred will be prefixed with a "ER:".
Folders can not be transferred, yet they are listed if there are folders.

```
----------------
REA=9
----------------
REB=Success
----------------
REC=OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\.htaccess
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\steeloverall.swf
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\steeloverall.swf_old
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\ASSETS
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\DOWNLOAD
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\favicon.ico
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\IMAGES
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\index.htm
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\index_old.php
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\logo.gif
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\MANAGER
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\ONLINE-HELP
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\robots.txt
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\TEMP
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\TEMPLATES
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\winrobots.jpg

----------------
```

If you specify an invalid remote-folder or no remote folder (leave **P4** empty), the root-folder that you have permission with you ftp-Login and Password, is used instead.
Therefore in most cases you may leave **P4** empty.

Example:
If you have on your FTP-Server this folder:

```
wrcom/site
```

and your FTP-Login and Password is for "wrcom", then you will always get the files from "wrcom" if you specify an invalid folder. To get the file4s that are inside "wrcom/site" you do have to specify "site" as remote-folder. Do not specify "wrcom/site" as this may not work.

Here is an example with just one file downloaded.

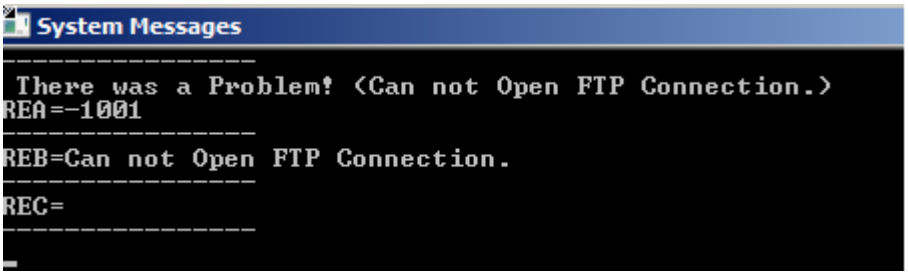
```
----------------
REA=1
----------------
REB=Success
----------------
REC=ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\CODE-CENTER_BKUP
OK:f:\00_mr\pb_minirobot\projects\ftp-test\test\favicon.ico
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\INSTALL____-----_____
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\PHPMYADMIN___-----___
ER:F:\00_MR\PB_MINIROBOT\PROJECTS\FTP-TEST\TEST\SITE

----------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

Here is what you get if you specify an invalid FTP server or if your firewall blocks the connection. As you can see, **Error-Numbers are negative**, this way you can easily find out, if there was a problem, using:

```
IVV.$$REA<0
  PRT. There was a Problem! ($$REB)
EIF.
```

It will look like this:



Syntax

FTP.ftp|P1|P2|P3|P4|P5|P6[|P7][|P8][|P9]

Parameter Explanation

P1 – **URL of FTP-Server**
P2 – ***Login for FTP-Server*** (Leave empty if none)
P3 – **Password** for FTP-Server (Leave empty if none)

P4 – **remote-folder** (Folder on the ftp-server), leave empty to use root-folder.
Please read comments about P4 in the text above!

P5 – **file-name/file-mask** specify which file(s) you want to download.
You can use wildcards * and "?".

P6 – **Local path** to save the transferred files to.

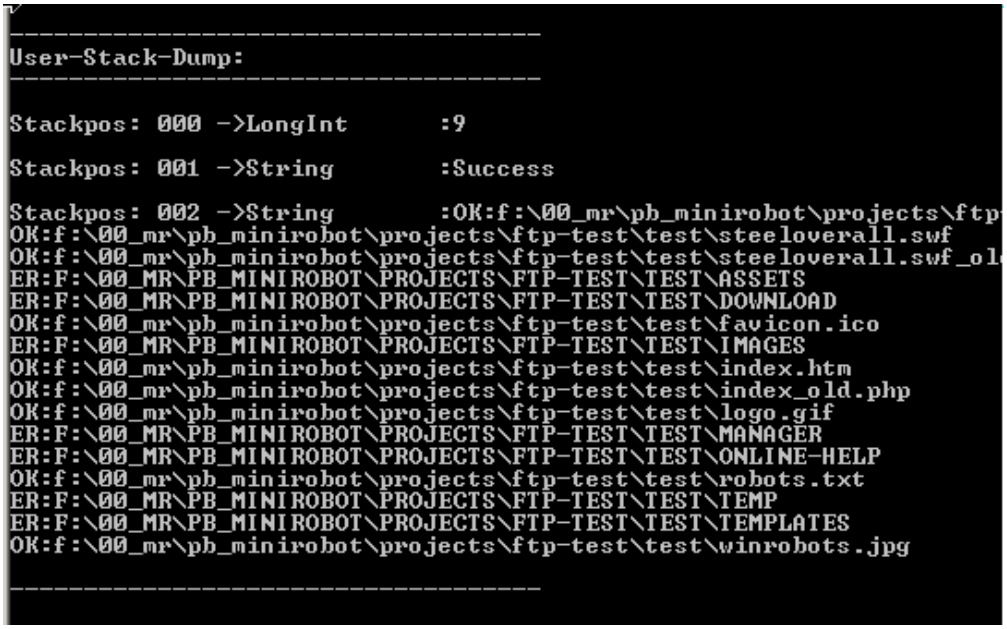
P7 – (***optional***) Returns an numeric code that shows success or failure of the operation. If omitted the number will be placed on TOS.
This numeric number will be positive in case of success and negative in case fatal of errors.
If **P7** is positive, the number is the number of files that have been transferred successfully.
if **P7** is negative, read **P8** for an explanation of the error that has happened.

P8 – (***optional***) Return the textual message that corresponds to the numerical number in **P7**.

© 2023 Theo Gottwald * https://www.smart-package.com

P9 - (***optional***) Return a `$crlf$` separated list with all files/ folders that have been processed. **P9** contains a detailed list with all FTP transfers that were successful or not.
P9 will also list folders and files that can not be transferred due to right or other problems. Files that have been copied and counted will be in small letter and prefixed with a "**OK:**". Files that could not be transferred will be prefixed with a "**ER:**".
Folders can not be transferred, yet they are listed if there are folders.

This is how the stack looks, if you do not specify **P7, P8** and **P9**.



Example

```
'**********************************
' FTP.-DEMO
'**********************************
' The ftp-server
$$URL=ftp-server.de
' Your Login
$$LOG=your_login
' Your Password
$$PWD=the-password
' The folder on the ftp-derver
$$PAT=site-folder
' File-Mask. Define which file(s) you want to download
' using wildcards: ?/*
$$FIM=*.*
' The local path where you want these files
' to downloaded to
$$LOC=?path\test\
FTP.ftp|$$URL|$$LOG|$$PWD|$$PAT|$$FIM|$$LOC|$$REA|$$REB|$$REC
PRT.----------------
PRT.REA=$$REA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IVV.$$REA<0
  PRT. There was a Problem! ($$REB)
EIF.
PRT.----------------
PRT.REB=$$REB
PRT.----------------
PRT.REC=$$REC
PRT.----------------
MBX.Ready
ENR.
```

Remarks

-

Limitations:

-

See also:

- DLV. - Download-to-Variable 2076
- CFU. - Copy-From-URL 2067
- URL. - Split URL 3234
- FTP. - File Transfer Protokoll 2078
- fto - file transfer protocol 2080
- tdl - Test Download 2078
- tpw - test with password 2078
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.6.3 tdl - Test Download

MiniRobotLanguage (MRL)

FTP.test download
FTP.tdl

Test a download link

Intention

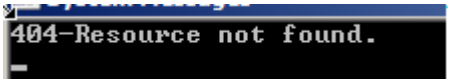
This command can be used to test a download-link, if it works or not.

Here is what we get with a valid URL:

```
$$URL=http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe
FTP.tdl|$$URL|$$RES|$$TXT
PRT.$$TXT
MBX.$$RES
ENR.
```

Here we have a valid server and an invalid link:

```
$$URL=http://download.microsoft.com/downloax/SQLServer2008-
KB2716433-x86.exe
FTP.tdl|$$URL|$$RES|$$TXT
PRT.$$TXT
MBX.$$RES
ENR.
```



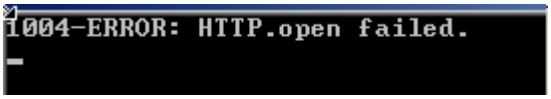
Now we will use an phantasy server and see what we get:

```
$$URL=http://download.fa2.de/download/9/2/7/927A4802-FC1C-
4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe
FTP.tdl|$$URL|$$RES|$$TXT
PRT.$$TXT
MBX.$$RES
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

At last we will specify an (invalid protocol) empty URL:

```
$$URL=
FTP.tdl|$$URL|$$RES|$$TXT
PRT.$$RES-$$TXT
MBX.Ready
ENR.
```



Syntax

FTP.tdl|P2[|P3][|P4]

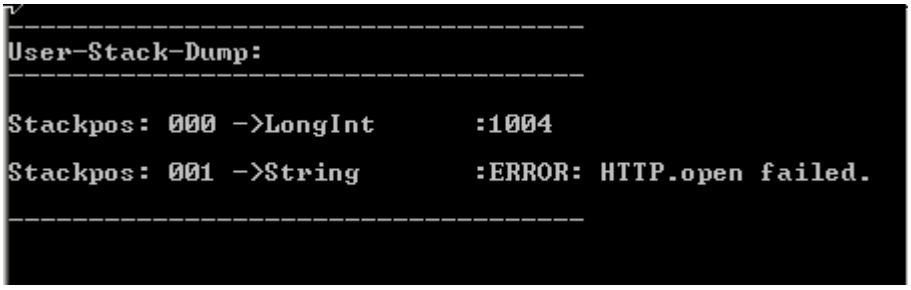
Parameter Explanation

P2 **-** URL - that shall be tested

P3 **-** (***optional***) Variable for a numerical result on success or not.
200 - is success, 404 - is resource not found, others see above.

P4 **-** (***optional***) Variable for a textual result on success or not.
Details see above.

If you omit the Stack will look somehow like this:



Example

```
'**********************************
' FTP.tdl.-DEMO
'**********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$URL=http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe
FTP.tdl$$URL|$$RES|$$TXT
PRT.$$TXT
IVV.$$RES!200
   MBX.There was a problem ($$RES)
EIF.
ENR.
RET.
```

Remarks

-

Limitations:

-

See also:

- DLV. - Download-to-Variable 2076
- CFU. - Copy-From-URL 2067
- URL. - Split URL 3234
- FTP. - File Transfer Protokoll 2078
- fto - file transfer protocol 2080
- tdl - Test Download 2078
- tpw - test with password 2078

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.6.4 tpw - test with password

MiniRobotLanguage (MRL)

FTP.test with password
FTP.tpw

Dowload Files via FTP and More Internet Commands

Intention

This command can be used to test a download-link, if it works or not.

Here is what we get with a valid URL:

```
$$LOG=YourLogin
$$PWD=Y?urPas-word
$$URL=http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe
FTP.tpw|$$URL|$$LOG|$$PWD|$$RES|$$TXT
PRT.$$TXT
MBX.$$RES
ENR.
```

Here we have a valid server and an invalid link:

```
$$LOG=YourLogin
$$PWD=Y?urPas-word
$$URL=http://download.microsoft.com/downloax/SQLServer2008-
KB2716433-x86.exe
FTP.tpw|$$URL|$$LOG|$$PWD|$$RES|$$TXT
PRT.$$TXT
MBX.$$RES
ENR.
```

Now we will use an phantasy server and see what we get:

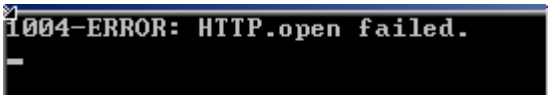
```
$$LOG=YourLogin
$$PWD=Y?urPas-word
$$URL=http://download.fa2.de/download/9/2/7/927A4802-FC1C-
4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
FTP.tpw|$$URL|$$LOG|$$PWD|$$RES|$$TXT
PRT.$$TXT
MBX.$$RES
ENR.
```

At last we will specify an (invalid protocol) empty URL:

```
$$LOG=YourLogin
$$PWD=Y?urPas-word
$$URL=
FTP.tpw|$$URL|$$LOG|$$PWD|$$RES|$$TXT
PRT.$$RES-$$TXT
MBX.Ready
ENR.
```



Syntax

FTP.tpw|P2[|P3][|P4]

Parameter Explanation

P2 – ***Download***-URL - that shall be tested

P3 – ***Login for WEB-Server*** (Leave empty if none)

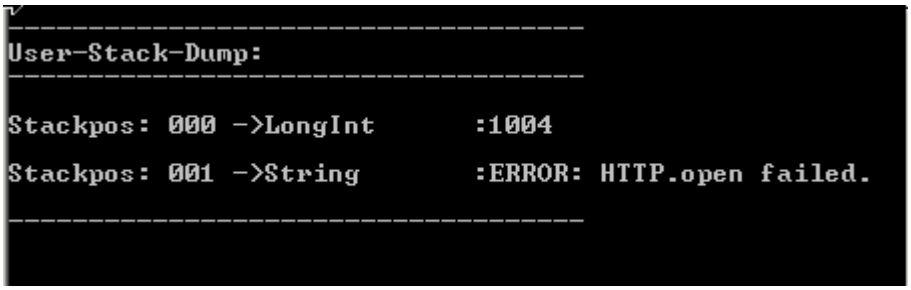
P4 – **Password *for WEB-Server*** (Leave empty if none)

P5 – (***optional***) Variable for a numerical result on success or not.
200 - is success, 404 - is resource not found, others see above.

P6 – (***optional***) Variable for a textual result on success or not.
Details see above.

If you omit the Stack will look somehow like this:

© 2023 Theo Gottwald * https://www.smart-package.com



Example

```
'**********************************
' FTP.tdl.-DEMO
'**********************************
$$LOG=
$$PWD=
$$URL=http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe
FTP.tdl$$URL|$$LOG|$$PWD|$$RES|$$TXT
PRT.$$TXT
IVV.$$RES!200
   MBX.There was a problem ($$RES)
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- DLV. - Download-to-Variable 2076
- CFU. - Copy-From-URL 2067
- URL. - Split URL 3234
- FTP. - File Transfer Protokoll 2078
- fto - file transfer protocol 2080
- tdl - Test Download 2078
- tpw - test with password 2078

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.7 GCN. - Get Computer Name

MiniRobotLanguage (MRL)

GCN. Command

Get Computer Name

Intention

Can be used to get the computer name of the actuals computer we are running on. '

Can also be used to get the hostname of any other computer, if we provide the IP address. Note that a DNS may be needed for this to work.

Syntax

GCN.[P1][|P2]

Parameter Explanation

P1 - (optional) Variable for Result. If omitted result is been placed on stack. Even if provided, there is additional Information on the stack.

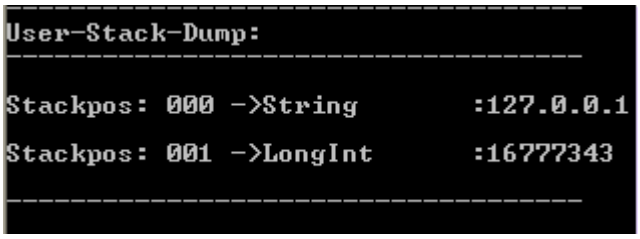
P2 - (optional). IP-Address of which we ant to get the Host-Name. If omitted the actual computer is been used.

```
' Will compute actual computers hostname and place
' informations on the stack.
GCN.

' Will compute actual computers hostname and place it
' in variable $$RES. There will be further informations
' on the stack.
GCN.$$RES

' Will compute another computers computers hostname and place
it
' in variable $$RES. There will be further informations
' on the stack.
' Note that a DNS may be needed for this to work.
: $$IPA=82.156.106.219
GCN.$$RES|$$IPA

' Will compute another computers computers hostname and place
it
' in on the stack.
```

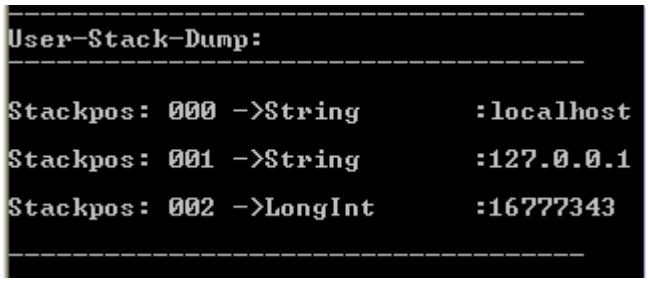

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Note that a DNS may be needed for this to work.
: $$IPA=82.156.106.219
GCN.|$$IPA

' Will compute another computers computers hostname and place
it
' in on the stack.
' You can provide the IP-Address in a numeric format also.
' Note that a DNS may be needed for this to work.
: $$IPA=16777343
GCN.|$$IPA
```

This is how the stack looks like, if you provide 127.0.0.1 or 16777343 as IP-Address, and also provide a variable for the hostname.



This is how the actual stack looks, if you do not provide a variable for the result. You see the hostname on TOS.



Example

```
'-------------------------------------------------------------
' GCN. - Get Computer Name
'
' Without Parameter
' Result on the stack
GCN.
DMP.6
MBX. Next with Result in variable

STS.CLEAR
GCN.$$IPA
DMP.6
MBX. The IP-Adress is $$IPA

STS.CLEAR
' Now we try to get "another" computers Name
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' from IP-Adress
: $$IPB=127.0.0.1
GCN.$$IPA|$$IPB
DMP.6
MBX. The IP-Adress is $$IPA

STS.CLEAR
' Now we try to get "another" computers Name
' from IP-Adress as number
: $$IPB=16777343
GCN.$$IPA|$$IPB
DMP.6
MBX. The IP-Adress is $$IPA

ENR.
```

Remarks

Limitations:

Depending on circumstances, a DNS-Server may be needed to get the host-name of IP addresses outside the local network.

See also:

- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.8 GIP. - Get IP Adress

MiniRobotLanguage (MRL)

GIP. Command

Get IP Address

Intention

Compute the IP Address of the actual computer. The code very simple:
`GIP.`

if multiple network cards, a index number can be given. Like this:
`GIP.2`

And this is how the stack will look in either case. The specific IP-numbers are just for example:

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->String        :192.168.178.56
Stackpos: 001 ->LongInt       :951232704

------------------------------------
```

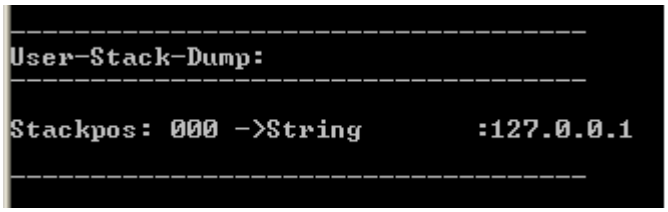
If you provide an host name/computer name, the command will translate the host name into the corresponding IP address. This can be a computer name from your network, as well as a internet resource. The code looks like this:
`GIP.www.fa2.de`

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->String        :82.165.106.219
Stackpos: 001 ->LongInt       :-613767854

------------------------------------
```

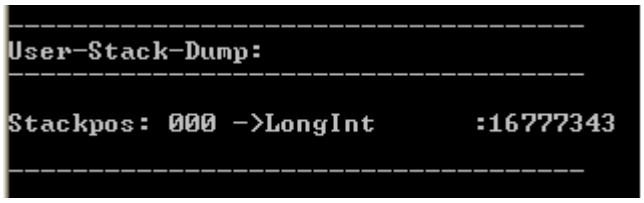
There is a second form of `GIP.` in which it can be used to translate a numeric IP address to a normal IP-address. Like this:
`GIP.>16777343`

and below you can see how the stack looks like.

*© 2023 Theo Gottwald * https://www.smart-package.com*



The opposite will also work, to translate a normal IP address to a numeric IP-address. You just write it like this:
`GIP.<127.0.0.1`



The stack now shows the IP in numeric form.

In all these cases you can provide a variable as second parameter. Like this:
```
GIP.|$$RES
GIP.2|$$RES
GIP.<127.0.0.1|$$RES
GIP.>16777343|$$RES
```

Then the value thats on the TOS will not be placed on the stack, but into the provided variable.

Syntax

GIP.[P1][|P2]

Parameter Explanation

P1 - (optional) host name, computer name or name of a internet resource that can be computed to an ip address. If omitted the ip address of the actual computer is been returned.

P2 - (optional) variable for the result

There is a second mode in which GIP. makes conversions between the numeric and the conventional form of an ip address. In these cases P1 must be prefixed with a ">" to convert a numeric IP to a standard IP. And if you prefix P1 with a "<", followed by Standard IP, this one will be translated into a numeric IP.

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' GIP. - Sample
STS.LOCAL
STS.CLEAR

GIP.
' show Actual Stack
DMP.6
MBX. Sample 1

STS.CLEAR

GIP.
' show Actual Stack
DMP.6
MBX. Sample 2

STS.CLEAR
GIP.1|$$REA
GIP.2|$$REB
' show Actual Stack
DMP.6
MBX.IP's are is $$REA-$$REB

STS.CLEAR
GIP.>16777343
DMP.6
MBX. IP Conversion done

STS.CLEAR
GIP.>16777343|$$RES
DMP.6
MBX. IP Conversion done: $$RES

STS.CLEAR
GIP.<127.0.0.1
DMP.6
MBX. IP Conversion 3 done

STS.CLEAR
GIP.<127.0.0.1|$$RES
DMP.6
MBX. IP Conversion 3 done $$RES

STS.CLEAR
GIP.www.fa2.de
DMP.6
MBX. Network Name resolved.
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- GUN. - Get User Name 2103
- GCN. - Get Computer Name 2091
- GPU. - Get Process User 2898

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.9 GNS. - Get Network Share

MiniRobotLanguage (MRL)

GNS. Command

Get Network Share

Intention

Can be used to retrieve the connection behind a drive-letter or specification.

```
' With a drive-letter it will retrieve the share
$$DRV=F:
GNS.$$DRV|$$ERG
' May return \\pcname\folder
DBP.$$ERG

' Without drive-letter will return all available shares
GNS.|$$ERG
DBP.$$ERG
```

In case of error, the timeout-flag is set and the error-number is on the TOS.
Otherwise the result is returned whether on TOS or in **P5.**

Hint: Use the RDM. - Remove Drive Mapping 2112 command to remove a mapped drive.

Syntax

GNS.[P1][|P2]

Parameter Explanation

P1 - ***(optional) Drive-Letter*** includes the ":" Example: `K:`
If no parameter is given, the result is a List of all shares, separated with "*".
Example:
`X:=\share1\folder*Y:=\share2\folder*Z:=\share3\folder`

P2 - (optional). Variable that will return the result of the operation. In case of success the result will be zero. In case of error an error number is returned. If omitted TOS is used.

5 - ERROR_ACCESS_DENIED
The caller does not have access to the network resource.
This could also be a access-rights problem.

85 - ERROR_ALREADY_ASSIGNED

© 2023 Theo Gottwald * https://www.smart-package.com

The local device specified is already connected to a network resource.

1200 - ERROR_BAD_DEVICE
The value specified is invalid.

67 - ERROR_BAD_NET_NAME
The value specified is not acceptable to any network resource provider because the resource name is invalid, or because the named resource cannot be located.

1204 - ERROR_BAD_PROVIDER
The value specified does not match any provider.

1223 - ERROR_CANCELLED
The attempt to make the connection was canceled by the user through a dialog box from one of the network resource providers, or by a called resource.

1208 - ERROR_EXTENDED_ERROR
A network-specific error occurred.

487 - ERROR_INVALID_ADDRESS
The caller passed a pointer to a buffer that could not be accessed.
This error should not appear, as it is an internal error.

87 - ERROR_INVALID_PARAMETER
This error is a result of providing wrong parameters, or the connection is to a network that requires the redirecting of a local device. Please check the variables.

86 - ERROR_INVALID_PASSWORD
The specified password is invalid.

234 - ERROR_MORE_DATA
Internal Error. If this error is returned, then no connection has been made.

259 - ERROR_NO_MORE_ITEMS
The operating system cannot automatically choose a local redirection because all the valid local devices are in use.

1202 - ERROR_DEVICE_ALREADY_REMEMBERED
The local device name has a remembered connection to another network resource.

1203 - ERROR_NO_NET_OR_BAD_PATH
The network path was either typed incorrectly, does not exist, or the network provider is not currently available. Please try retyping the path or contact your network administrator.

1222 - ERROR_NO_NETWORK
The network is unavailable.

1219 - ERROR_SESSION_CREDENTIAL_CONFLICT
An attempt was made to establish a session to a network server, but there are already too many sessions established to that server.

1326 - ERROR_LOGON_FAILURE

*© 2023 Theo Gottwald * https://www.smart-package.com*

Logon failure: user account restriction. Possible reasons are blank passwords not allowed, logon hour restrictions, or a policy restriction has been enforced.

Example

```
'-------------------------------------------------------------
' GNS. - No Parameters
'
GNS.
ITO.
  PRT.error happend
EIF.
DMP.6
MBX.!
ENR.

' GNS. - with Parameters
'
$$PAT=X:
GNS.$$PAT|$$ERG
ITO.
  PRT.error happend
EIF.
PRT.$$ERG
MBX.!
ENR.
```

Remarks

If the command fails, the Timeout-Flag is set, otherwise it will be cleared.

Limitations:

-

See also:

- MND. - Map Network Drive 2108
- CNS. - Connect Net Share 2070
- RDM. - Remove Drive Mapping 2112
- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- GVL. - Get Volume List 2105

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.10 GUD. - Get User and Domain

MiniRobotLanguage (MRL)

GUD. Command

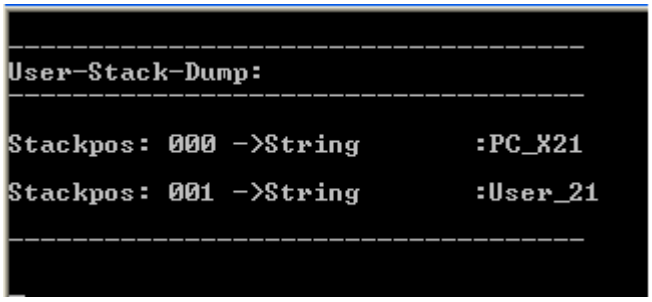
Get User and Domain Name

Intention

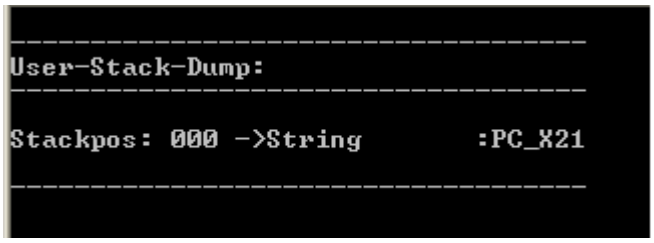
Get the name of the user that is actually running your script. When using the 1.1 '#RAU: - Run As User 271 it may be a good idea to find out, if the script really got the proper user.

`GUD.` provides you this information. It shows you, which user runs the actual script.

If you run `GUD.` without a parameter, the stack will get the information like in the picture below.



If you provide a variable for the User-Name, the damain information will be on the stack, and the user name in the provided variable. The stack will then look like this.



If you provide two variables, both values will be in the variables instead then on the stack.

Syntax

GUD.[P1][|P2]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** **(**optional) Variable for the user name. If omitted TOS is used.

P2 **-** **(**optional) variable for the domain/pc name. If omitted TOS is used.

Example

```
' Show GUN. -Get User Name Command
STS.LOCAL
STS.CLEAR

GUD.$$USR
STS.DUMP
MBX. User is: $$USR
ENR.
```

Remarks

Uses internally another system then GUN.

Limitations:

-

See also:

- GIP. - Get IP Adress 2094
- GCN. - Get Computer Name 2091
- GUN. - Get User Name 2103
- GPU. - Get Process User 2898

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.11 GUN. - Get User Name

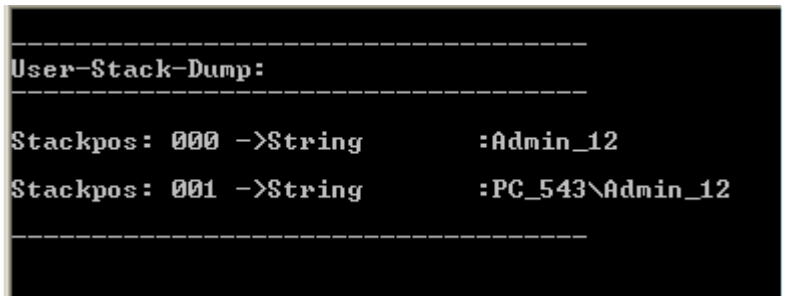
MiniRobotLanguage (MRL)

GUN. Command

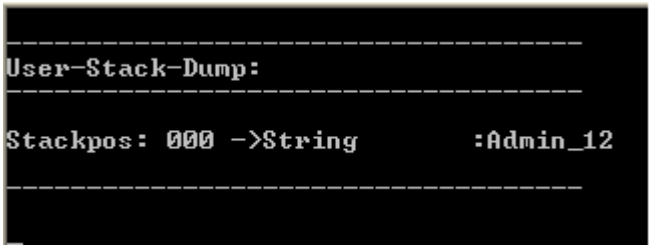
Get User Name

Intention

Get the name of the user that is actually running your script. When using the 1.1 '#RAU: - Run As User 271 it may be a good idea to find out, if the script really got the proper user.
GUN. provides you this information. It shows you, which user runs the actual script.
If you run GUN. without a parameter, the stack will get the information like in the picture below.



If you provide a variable for the User-Name, the user name will not be on the stack, but in this variable. The stack will then look like this.



Syntax

GUN.[P1][|P2]

Parameter Explanation

P1 **-** (optional) Variable for the result.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **-** **(**optional) the letter **N**, then another Algo is used.

Example

```
' Show GUN. -Get User Name Command
STS.LOCAL
STS.CLEAR

GUN.
' show Actual Stack
DMP.6
MBX. Sample 1

STS.CLEAR
GUN.$$RES
' show Actual Stack
DMP.6
MBX. Sample 2, Name is $$RES

ENR.
```

Remarks

-

Limitations:

-

See also:

- SID. - Security Identifier 2114
- GIP. - Get IP Adress 2094
- GCN. - Get Computer Name 2091
- GUD. - Get User and Domain 2101
- GPU. - Get Process User 2898

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.12 GVL. - Get Volume List

MiniRobotLanguage (MRL)

GVL. Command

Get Volume List

Intention

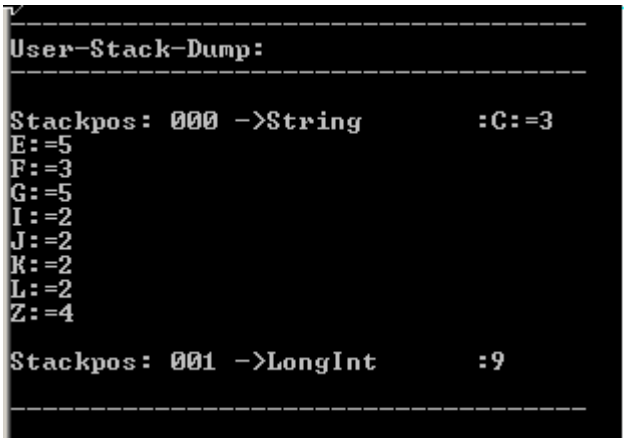
Can be used to retrieve a list of all available drive-letters (Volumes). Will also place the number of available Volumes on TOS.

```
' Without parameter will place a list of all Volumes on the
TOS
GVL.
' Show TOS
DMP.6
MBX.!
```

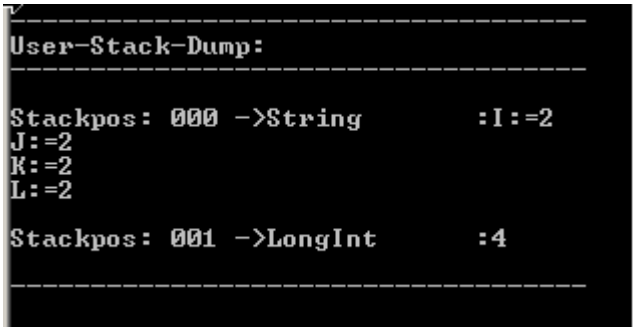
This is an possible result:

```
' Get the result in a variable
GVL.|$$ERG
DBP.$$ERG

' Change Format to numeric
GVL.n|$$ERG
DBP.$$ERG
```


*© 2023 Theo Gottwald * https://www.smart-package.com*




```
' Only return Volumes of a defined type
GVL.n2
' Show TOS
DMP.6
MBX.!
```



In case of error, the timeout-flag is set.

The command will always count the number of results, and place this number below TOS.

Syntax

GVL.[P1][|P2]

Parameter Explanation

P1 - ***(optional)*** numeric parameter, from 1-6 .
If specified only volumes of this type is listed.
if you prefix the number with an "n", you will get an numeric output.
If no parameter is given, a List with all available Volumes is returned.
They will be separated with `$crlf$`
These are the numbers you can choose from:

© 2023 Theo Gottwald * https://www.smart-package.com

```
DRIVE_UNKNOWN     = 0
DRIVE_NO_ROOT_DIR = 1
DRIVE_REMOVABLE   = 2
DRIVE_FIXED       = 3
DRIVE_REMOTE      = 4
DRIVE_CDROM       = 5
DRIVE_RAMDISK     = 6
```

P2 - (optional). Variable that will return the result of the operation. In case of success the result will be zero. In case of error an error number is returned. If omitted TOS is used.

Example

```
'-------------------------------------------------------------
' GVL. only volumes that are removable
' Result on TOS, and numeric output
'
GVL.n2
' Show TOS
DMP.6
MBX.!
ENR.
```

Remarks

-

Limitations:

-

See also:

- IVA. - If Volume available 1053
- **MND. - Map Network Drive** 2108
- **CNS. - Connect Net Share** 2070
- **RDM. - Remove Drive Mapping** 2112
- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- GNS. - Get Network Share 2098

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.13 MND. - Map Network Drive

MiniRobotLanguage (MRL)

MND. Command

Map Network Drive

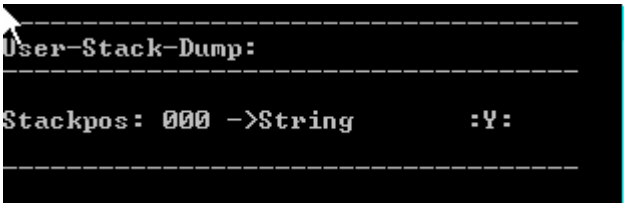
Intention

Can be used to assign a drive-letter to a network resource.
This can be done temporary, or you can specify that this is going to be connected on each reboot.

Just use the command and specify which Mapping you want:

```
' Add Mapping using automatic driveletter temporary
MND.\\MyComputer\Folder$
```

Will leave the automatically assigned driveletter on the TOS (**Y:**).




```
' Assign  Mapping for Drive F: permanent
RDM.F|1

' Remove Mapping for Drive F: permanent
' and verify it was removed
RDM.F|1|$$ERG
IVV.$$ERG=0
  MBX.Success
EIF.
```

Hint: Use the RDM. - Remove Drive Mapping 2112 command to remove a mapped drive.

Syntax

MND.P1[|P2][|P3][|P4][|P5][|P6]

Parameter Explanation

P1 - Share to connect, example: ***\\computer\folder***

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - *(optional)* Driveletter, for example: Y:

P3 - *(optional)* ***Username***

P4 - *(optional)* ***Password***

P5 - *(optional)*. 1/0 -If 1 is specified, the Mapping is permanent.
It will be restored at the next reboot.
If 0 or nothing is specified, the drive mapping is temporary
and not saved in the user profile.

P6 - (optional). Variable that will return the result of the operation. In case of success the result will be zero. In case of error an error number is returned. If omitted TOS is used.

5 - ERROR_ACCESS_DENIED
The caller does not have access to the network resource.
This could also be a access-rights problem.

85 - ERROR_ALREADY_ASSIGNED
The local device specified is already connected to a network resource.

1200 - ERROR_BAD_DEVICE
The value specified is invalid.

67 - ERROR_BAD_NET_NAME
The value specified is not acceptable to any network resource provider because the resource name is invalid, or because the named resource cannot be located.

1204 - ERROR_BAD_PROVIDER
The value specified does not match any provider.

1223 - ERROR_CANCELLED
The attempt to make the connection was canceled by the user through a dialog box from one of the network resource providers, or by a called resource.

1208 - ERROR_EXTENDED_ERROR
A network-specific error occurred.

487 - ERROR_INVALID_ADDRESS
The caller passed a pointer to a buffer that could not be accessed.
This error should not appear, as it is an internal error.

87 - ERROR_INVALID_PARAMETER
This error is a result of providing wrong parameters, or the connection is to a network that requires the redirecting of a local device. Please check the variables.

86 - ERROR_INVALID_PASSWORD
The specified password is invalid.

234 - ERROR_MORE_DATA

*© 2023 Theo Gottwald * https://www.smart-package.com*

Internal Error. If this error is returned, then no connection has been made.

259 - ERROR_NO_MORE_ITEMS
The operating system cannot automatically choose a local redirection because all the valid local devices are in use.

1202 - ERROR_DEVICE_ALREADY_REMEMBERED
The local device name has a remembered connection to another network resource.

1203 - ERROR_NO_NET_OR_BAD_PATH
The network path was either typed incorrectly, does not exist, or the network provider is not currently available. Please try retyping the path or contact your network administrator.

1222 - ERROR_NO_NETWORK
The network is unavailable.

1219 - ERROR_SESSION_CREDENTIAL_CONFLICT
An attempt was made to establish a session to a network server, but there are already too many sessions established to that server.

1326 - ERROR_LOGON_FAILURE
Logon failure: user account restriction. Possible reasons are blank passwords not allowed, logon hour restrictions, or a policy restriction has been enforced.

Example

```
'-------------------------------------------------------------
' MND. - No Pasword
'
$$PAT=\\XDRIVE\Daten$
MND.$$PAT
ITO.
  PRT.error happend
EIF.
DMP.6
MBX.!
ENR.

' MND. - with Pasword
'
$$PAT=\\XDRIVE\Daten$
MND.$$PAT||Admin|Asjfds_55
ITO.
  PRT.error happend
EIF.
DMP.6
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

If the command fails, the Timeout-Flag is set, otherwise it will be cleared.

Limitations:

-

See also:

- RDM. - Remove Drive Mapping 2112
- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- IVA. - If Volume available 1053

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.14 RDM. - Remove Drive Mapping

MiniRobotLanguage (MRL)

RDM. Command

Remove Mapped Network Drive

Intention

Can be used to remove a mapped network drive, temporary or permanently.
Just use the command and specify which Mapping to remove:

```
' Remove Mapping for Drive F: temporary
RDM.F

' Remove Mapping for Drive F: permanent
RDM.F|1

' Remove Mapping for Drive F: permanent
' and verify it was removed
RDM.F|1|$$ERG
IVV.$$ERG=0
  MBX.Success
EIF.
```

Hint: Use the MND. - Map Network Drive 2108 to add a Mapped Drive.

Syntax

RDM.P1[|P2][|P2]

Parameter Explanation

P1 - Driveletter, for example Y or M

P2 - (optional). 1/0 -If 1 is specified, teh Mapping is permanently removed.
If 0 or nothing is specified, the drive mapping is removed, but possibly restored at next reboot.

P3 - (optional). Variable that will return the result of the operation. In case of success the result will be zero. In case of error an error number is returned.

1206 - ERROR_BAD_PROFILE
The user profile is in an incorrect format.

1205 - ERROR_CANNOT_OPEN_PROFILE

*© 2023 Theo Gottwald * https://www.smart-package.com*

The system is unable to open the user profile to process persistent connections.

2404 - ERROR_DEVICE_IN_USE
The device is in use by an active process and cannot be disconnected.

1208 - ERROR_EXTENDED_ERROR
A network-specific error occurred.

2250 - ERROR_NOT_CONNECTED
The name specified is not a redirected device, or the system is not currently connected to the device specified by the parameter.

2401 - ERROR_OPEN_FILES
There are open files. This error should not occur, as the Command uses force to break the connection.

Example

```
'-------------------------------------------------------------
' RDM. - Get Computer Name
'
RDM.F
RDM.F|0|$$RES
ENR.
```

Remarks

If the command fails, the Timeout-Flag is set, otherwise it will be cleared.

Limitations:

-

See also:

- MND. - Map Network Drive 2108
- CNS. - Connect Net Share 2070
- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- IVA. - If Volume available 1053

© 2023 Theo Gottwald * https://www.smart-package.com

3.57.15 SID. - Security Identifier

MiniRobotLanguage (MRL)

SID. Command

Get Security Identifier* for actual or any User Name

Intention

Get the **SID** (Security Identifier)*of the current or any given User as Text String.

Example **SID:** `S-1-5-21-435374019-1570027373-1801674531-1013`

Usage is simple:

```
' Print current user SID as Text
SID.
DBP.$$000
ENR.

' Print Guest User SID
$$NAM=MYPC\Guest
SID.$$NAM
DBP.$$000
ENR.

' Print Guest User SID, use Variable for Result
$$NAM=MYPC\Guest
SID.$$NAM|$$RES
DBP.$$RES
ENR.
```

***Security Identifier** (commonly abbreviated **SID**) is a unique, immutable identifier of a user, user group, or other security principal.
A security principal has a single SID for life, and all properties of the principal, including its name, are associated with this SID.
This allows a principal to be renamed (for example, from "UserB" to "User21", without affecting the security attributes of objects that refer to the principal.

Syntax

SID.[P1][|P2]

Parameter Explanation

 *© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** **(**optional) Name of User to return the SID from. If omitted,
current user is taken.
P2 **-** **(**optional) Variable for the result.

Example

```
' Print Guest User SID, use Variable for Result
$$NAM=MYPC\Guest
SID.$$NAM|$$RES
DBP.$$RES
ENR.
```

Remarks

-

Limitations:

-

See also:

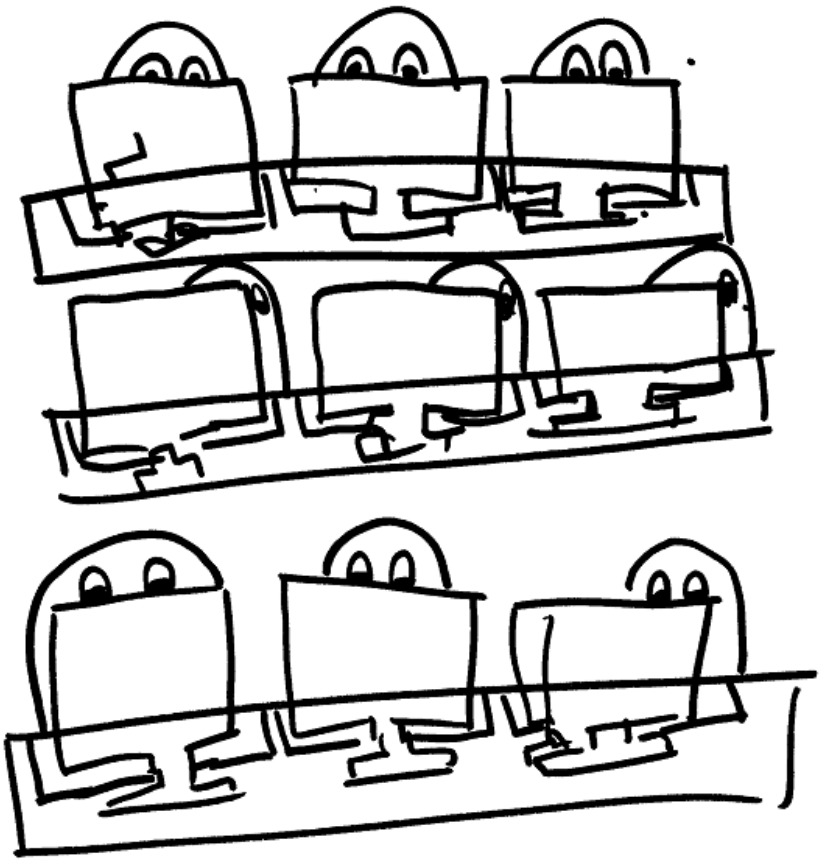
- GIP. - Get IP Adress 2094
- GCN. - Get Computer Name 2091
- GUD. - Get User and Domain 2101
- GPU. - Get Process User 2898

© 2023 Theo Gottwald * https://www.smart-package.com

3.58 IPC, InterKOM and Exit Code

3.58.1 ! Smarty's InterKOM Technology

Smarty's InterKOM Technology



Intention

What is InterKOM good for?

The SPR System is actually a system that contains multiple components. You have one ore more robots, a Plug In, the Mini-Browser, the Editor and in the future there will be a lot more components.

All of these components need to work together in a synchronized way. They need to communicate to each other. For example, the robot may want to ask the Browser "are you ready loading that page?"

*© 2023 Theo Gottwald * https://www.smart-package.com*

Synchronization and data exchange must be done in a very fast way and also it may be necessary to communicate even through cross boundaries of rights-management. This is not possible using actual windows messaging.

That is why we invented InterKOM.
InterKOM withing the SPR-System means "**Inter Process Communication**".

For example, if your Robot is running in another session you may still want it to send a debugging message to the Editor or to another robot instance.

As a base for actual and future "Inter Process Communication, InterKOM was made.
Its the newest technological step ahead in the SPR System.
Together with the new background technology that was added recently, InterKOM will build the base for interesting developements in the future.

It can interface and synchronize any thinkable executable that should be integrated into the SPR system in a lighning fast way.
And it works system-wide, for example you can send InterKOM Messages to robots that run in another session, which will not work using regular windows messaging.

Using `GIK.` you can read a number of defined values that can be read or written by any other InterKOM program also.
Its like you have a shared pool of variables. And as InterKOM is lightning fast, this is perfect for synchronization between all components.
In the same way, you can sent InterKOM-Messages using `SIK.`

For this to be possible, each program that joins InterKOM will get a "InterKOM" handle, that is like a postal address for InterKOM. In fact its nothing complicated.

The first program that joins InterKOM receives the handle "1", the second receives the handle "2" and so on. Its a number and currently the upper limit is 16.
This may change in future versions, as InterKOM's architecture could handle a lot more.
Its is currently the maximum number of possible InterKOM users and 16 robots and Editors and browsers are enough for now.

Each program that starts up, whether its a robot, and editor or an Mini-Browser, will immediately get an InterKOM handle. And that handle is called a "CUN" which means "communication unique number". Because its unique and because its purpose is to enable communication between the programs.

What are the InterKOM Commands?

Mostly InterKOM serves two purposes: Synchronization and Messaging.
All of these use the same commands:

- `WIK.` - Wait until an InterKOM event has happened.
- `SIK.` - Set an InterKOM event or send an InterKOM Message.
- `GIK.` - Get an InterKOM State or get an InterKOM Message.
- `IIK. / NIK.` - Check if an InterKOM Event has happened

*© 2023 Theo Gottwald * https://www.smart-package.com*

Do I have special system variables for use with InterKOM?

You have the following System-Variable for use with InterKOM:

`#cun#` – CUN of the actual robot that runs this script
`#ecun#` – CUN of the Editor that has started this robot - if any.
`#fcun#` – CUN of the robot that has started a Parallel-Robot (father-CUN)
`#bcun#` – CUN of the last started Mini-Browser (`LBR.`-Command only)
`#pcun#` – CUN of the last started Parallel-Robot (`PRR.`-Command)

What are the InterKOM Sync-Bytes?

For Synchronisation purposes, Interkom offers a number of Synchronisation Bytes. These Bytes are unique for each InterKOM Member. If you have for example 3 programs that use InterKOM, then any of this programs has the full set of InterKOM-Bytes.

If a program leaves InterKOM, these Bytes are being deleted automatically.
Let's take a look which synchronization bytes we have (for each InterKOM member):

- 1-250 Byte
  Bytes can hold any value from 0 to 255

- 1-250 Long
  Longs can hold any value from -2147483648 to 2147483647

- 1-250 Quad
  A Quad can hold any value from zero to 18446744073709551615.
  Each Quad can also hold a text of 8 letters.

- 1-250 Float
  A Float can hold positive or negative values that contain a fractional part.
  For example 1.234354 or such. They use very high precision internally.
  Higher then +/-1.7976931348623157E+308#

In fact InterKOM offers more then 250 Bytes of each sort. However these last bytes are reserved for internal usage of the robot system. Do not try to use them, or you may experience unknown problems.

As each InterKOM using program has a private, full set/pool of "own Sync-Bytes", you need to specify the CUN of the Member, who's bytes you want to read or change.

Was ist ein InterKOM Typ?

Any System-Component, like for example Robot, Editor or Mini-Browser will log on to InterKOM and tell its TYPE to InterKOM.
This way InterKOM always knows the Type of any running Module, and therefore you can search a CUN by TYPE. The actually defined Types are:

```
TYP unkonwn      = 0
TYP Robot        = 1
TYP Editor       = 2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TYP PlugIn       = 3
TYP Aoolication  = 4
TYP Mini Browser = 5
```

In case there are multiple Instances of, for example Robots, you can specify a number to select the one you want. Example:

```
$$TYP=1
$$NUM=2
GIK.cft|$$TYP|$$NUM|$$RES
DBP.Robot CUN is: $$RES / #cun#
```

Hint: InterKOM will also be know about the **Hwnd** and the **PID** of any attached Component.

Therefore you can also search a Component **CUN** by its **Hwnd** or its **PID**.

In the other direction, you can ask InterKOM for the Hwnd of a Component, if you have

the CUN. You can get the CUN most easy using System-Variables.

How will InterKOM Handle Messages internally?

InterKOM will que all Messages to a specific CUN, until the receiver CUN will use `GIK.` to receive these Messages. To prevent the Message-Cue from Overflow, it has a maximum value of >4000 Messages that are inserted into the que.
If you retrieve a message using `GIK.` , the message is been deleted from the Message Que. There is no need to get all Messages out before shutting down a Component. However when a Component ends, all left over message are being discarded.

An Example Script

```
'*********************************
' This example uses INTERKOM to signal a second Robot to
' end running in a looping.
'*********************************
'
' Run Parallel Robot
PRR.
  ' Get InterKOM CUN of father-robot
  $$CUN=#fcun#
  ' Here is the Loop
  DOL.
    ' Using GIK. we will read the Byte Nr.1 and put
    ' the result in $$RES
    GIK.Byte|1|$$CUN|$$RES
    PRT.received: $$RES
    PAU.0.5
  OOP.$$RES=10
  MBX.Ending.
  ENR.
PRE.
' Here is the Main Robot
PAU.0.2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Set the Pool-Byte to 10 this will make the Parallel-Robot
' end its loop.
SIK.Byte|1|10
' Receive the byte also
GIK.Byte|1||$$RES
DBP.received: $$RES
MBX.Signal sent.
ENR.

'*********************************
' Get any Message Sample-Script
'*********************************
$$CUN=#cun#
$$PCU=#pcun#
PRR.
  DOL.
    $$ERG=0
    GIK.ma.|$$ERG
    PRT.Message=$$ERG
    GSB.Lab_From
    IVS.$$ERG=Ende
      $$RES=1
    ELS.
      $$RES=0
    EIF.
    PRT.Looping with: $$RES
    PAU.1.1
  OOP.($$RES=1)
  ENR.
PRE.
PAU.5
$$PCU=#pcun#
SIK.message to|$$PCU|Ende
MBX.!
ENR.

' Find out from WHO the message is
' this is stored as ASC. in the first byte of the Message
:Lab_From
JIV.$$ERG=0|Lab_over
' THis line is more ore less cosmetic, ASC. could do without
GSS.1|1|$$ERG|$$FRO
ASC.$$FRO|$$NUM
PRT.Message is from CUN: $$NUM
GSS.2|0|$$ERG|$$ERG
:Lab_over
RET.
'-------------------------------------------------------------
ENR.


'*********************************
' Get Handle from CUN
'*********************************
'
$$CUN=#cun#
GIK.handle|$$CUN|$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.My Handle is: $$RES = #cbw#
ENR.
```

Limitations

The maximum Message size using InterKOM is currently around 650 kb this limitation may be increased or removed in future versions.

See also:

- GIK. - Get InterKOM 2122
- SIK. - Set InterKOM 2131
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.2 GIK. - Get InterKOM

MiniRobotLanguage (MRL)

GIK. Command

Get a IPC-Message or Signal from another Component

Intention

Using `GIK.` you can read a number of defined values that can be read or written by any other InterKOM program also. Its like you have a shared pool of variables. And as InterKOM is lightning fast, synchronization will easily be possible this way.

Using `SIK.` you can write to this shared memory, and at the same time other processes can read the writings using `GIK.` or `IIK.` and react on that.

`GIK.` is a multi-purpose command that includes multiple possible subcommands. Usage is simple:

```
' Get the value of Sync-Byte "1" (lowest) from CUN(-Member) 2
' and place it in Variable $$ERG
GIK.Byte|1|2|$ERG

' Get the value of Sync-Byte "250" from our own Bytes
' and place it on TOS
GIK.Byte|250

' Get the value of Sync-Byte "127" our own Bytes
' and place it in Variable $$ERG
' zero as CUN always evaluates to our own CUN
GIK.Byte|1|0|$ERG
```

Syntax

GIK.P1|P2[|P3][|P4]

Parameter Explanation

P1 – `Subcommand`, the following Subcommands are currently supported. Note that for all of the subcommands there is a long form that is better readable, as well as a short form, that can be used to write faster.

- "Byte","B."
  `GIK.Byte|(Byte-Nr.)[|(CUN)][|$$ERG]`
  `GIK.Byte|$$BNU|$$CUN|$$RES`

© 2023 Theo Gottwald * https://www.smart-package.com

- "Long","L."
  GIK.Long|(Byte-Nr.)[|(CUN)][|$$ERG]
  GIK.Long|$$BNU|$$CUN|$$RES

- "Quad","Q."
  GIK.Quad|(Byte-Nr.)[|(CUN)][|$$ERG]
  GIK.Quad|$$BNU|$$CUN|$$RES

- "Float","F."
  GIK.Float|(Byte-Nr.)[|(CUN)][|$$ERG]
  GIK.Float|$$BNU|$$CUN|$$RES

- "Use","U." - "InUse"-Flag from a defined CUN.
  If this returns 0 then the CUN with that number is not in use.
  GIK.Use|(CUN)[|$$ERG]

- "Message any","ma."
  GIK.Message any[|$$ERG]

- "Message from","mf."
  GIK.Message from|(Sender-CUN)[|$$ERG]

- "Handle","h."
  GIK.Handle|(CUN)[|$$ERG]

- "CUN from Handle","cfh.","cfh"
  GIK.cfh|(Handle)[|$$ERG]

- "CUN from PID","cfp.","cfp"
  GIK.cfp|(PID)[|$$ERG]

- "CUN from Type","cbt.","cbt"
  GIK.cft|(Type)[|$$NUM][|$$ERG]

Depending on the chosen Subcommand the further Parameters **P2** to **P4** differ and may be
needed or not.

P2 – Depending on the chosen Subcommand this Parameter may be the number of the synchronization element or may have another usage.

P3 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

P4 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

Example

```
' This example uses INTERKOM to signal a second Robot to
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' end running in a looping.
'
' Run Parallel Robot
PRR.
  ' Get InterKOM CUN of father-robot
  $$CUN=#fcun#
  ' Here is the Loop
  DOL.
    GIK.Byte|1|$$CUN|$$RES
    PRT.received: $$RES
    PAU.0.5
  OOP.$$RES=10
  MBX.Ending.
  ENR.
PRE.
' Here is the Main Robot
PAU.0.2
' Set the Pool-Byte to 10 this will make the Parallel-Robot
' end its loop.
SIK.Byte|1|10
' Receive the byte also
GIK.Byte|1||$$RES
DBP.received: $$RES
MBX.Signal sent.
ENR.
```

Remarks

-

Limitations:

Currently there are limitations to the maximum size that a message may have, using InterKOM. These are 650 kb for this release. This may change in the future.

Also the number of possible InterKOM Clients is limited to 16 at this time.

See also:

- ! Smarty's InterKOM Technology 2116
- SIK. - Set InterKOM 2131
- WIK. - Wait InterKOM 2142
- IIK. - If InterCOM Condition 2125

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.3 IIK. - If InterCOM Condition

MiniRobotLanguage (MRL)

IIK.- Conditional Statement

Check if an InterKOM Condition is true

Intention

Conditional statement.

Using `IIK.` you can check if and InterKOM Condition is true or not. This can be the value of a "Pool Byte" or if a specified Message has arrived.

Using the "Alive" Option, you can also check if other InterKOM Members are alive or not.

InterKOM is like you have a shared pool of variables. As InterKOM is lightning fast, synchronization will easily be possible this way.
For more Informations on InterKOM see here ! Smarty's InterKOM Technology 2116.

`IIK.` is a multi-purpose command that includes multiple possible subcommands. Usage is simple.

```
IIK.!Alive|3
  DBP. CUN 3 is been shut down.
EIF.

' And technically the same
NIK.Alive|3
  DBP. CUN 3 is been shut down.
EIF.
```

In this more complex example we start a Parallel-Robot to set a Synchronisation Number in the shared Pool. Note that we also use a variable to define the subcommand.

```
$$SBC=Float
SIK.$$SBC|1|6.5454

$$NUM=6.5453
IIK.$$SBC|1|>$$NUM
  DBP. Yes its >$$NUM
ELS.
  DBP. No its not >$$NUM
EIF.

$$NUM=12
IIK.$$SBC|1|>$$NUM
  DBP. Yes its >$$NUM
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
ELS.
  DBP. No its not >$$NUM
EIF.
ENR.
```

Syntax

IIK.[!]P1|P2[|P3][|P4]

Parameter Explanation

[!] – If you prefix the Subcommand with a "**!**" then the Logic is negated and the given
condition is true, if its false and false if its true.

P1 – `Subcommand`, the following Subcommands are currently supported. Note that for all of the subcommands there is a long form that is better readable, as well as a short form, that can be used to write faster.

- "Byte","B."
  `IIK.[!]Byte|Nr.|cond/value[|CUN]`
- "Long","L."
  `IIK.[!]Long|Nr.|cond/value[|CUN]`
- "Quad","Q."
  `IIK.[!]Quad|Nr.|cond/value[|CUN]`
- "Float","F."
  `IIK.[!]Float|Nr.|cond/value[|CUN]`
- "Use","u."
  `IIK.[!]Use|CUN`
- "Alive","a."
  `IIK.[!]Alive||CUN`
- "Message from","mf."
  `IIK.Message from|CUN[|$$MSG]`
- "Message any","ma."
  `IIK.Message any|[$$MSG]`

Depending on the chosen Subcommand the further Parameters **P2** to **P4** differ and may be
needed or not.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 – Depending on the chosen Subcommand this Parameter may be the number of the synchronization element or may have another usage.

P3 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

P4 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

Speed in Ticks:

This command uses typically around 550 Ticks for sending a Q-Message.

Example

```
' This example uses INTERKOM to signal a second Robot to
' end running in a looping.
'
$$SBC=Long
SIK.$$SBC|1|6
$$NUM=4
IIK.$$SBC|1|>$$NUM
  DBP. Yes its >$$NUM
ELS.
  DBP. No its not >$$NUM
EIF.
$$NUM=12
IIK.$$SBC|1|>$$NUM
  DBP. Yes its >$$NUM
ELS.
  DBP. No its not >$$NUM
EIF.
ENR.
```

Remarks

There are multiple examples on InterKOM Commands within the supplied Sample Scripts.

Limitations:

Currently there are limitations to the maximum size that a message may have, using InterKOM. These are 650 kb for this release. This may change in the future.

Also the number of possible InterKOM Clients is limited to 16 at this time.

*© 2023 Theo Gottwald * https://www.smart-package.com*

See also:

- ! Smarty's InterKOM Technology 2116
- GIK. - Get InterKOM 2122
- SIK. - Set InterKOM 2131
- WIK. - Wait InterKOM 2142

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.4 SEC. - Set Exit Code

MiniRobotLanguage (MRL)

SEC. Command

Set a Return code for use by the calling programm

Intention

This command will set the Return code for the Robot-Executable only.
The Return code is also known as Error Level.
You can call this command at any place in your script to set the Return code for the "Robot-Executable", that is the "`MR01.exe`".

This command **will NOT set the Return code for compiled Scripts**. Use the `SRV.`-Command for this. Even multiple times.

Only the last call will set the finally used return code.

Usage is simple:

```
' Set a Return Code of zero (0)
SEC.

' Set a Return Code of zero (0)
SEC.0

' Set a Return Code of 10005
: $$PRO=10005
SEC.$$PRO

' Add multiple results
SEC.($$PR1+$$PR2+$$PR3)
```

Syntax

SEC.[P1]

Parameter Explanation

P1 - **`(optional)`** `NUM` ,variable or formula in (..).
if omitted, default is 0. Will be the Return code of the compiled executable.

Example

```
' Set a Return Code of zero (0)
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
SEC.

' Set a Return Code of zero (0)
SEC.0

' Set a Return Code of 10005
: $$PRO=10005
SEC.$$PRO
```

Remarks

This command will only set the Return Value for the "**MR01.exe**".
To set the Error-Level of compiled Scripts (EXE-Mode) use the `SRV.`-command.
To end a script and set a Return-code for executables, use `END`.

Limitations:

-

See also:

- 1.B Error handling 224
- SRV. - Set Return Value (Executable) 2134
- END. - End Script 2362
- ENR. - End Return 2169

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.5 SIK. - Set InterKOM

MiniRobotLanguage (MRL)

SIK. Command

Set a IPC-Message or Signal for another Component

Intention

Using `SIK.` you can send a defined value that can be read by any other InterKOM program, for example using `GIK.`

Its like you have a shared pool of variables. As InterKOM is lightning fast, synchronization will easily be possible this way.

Using `SIK.` you can write to this shared memory, and at the same time other processes can read the writings using `GIK.` or `IIK.` and react on that.

`SIK.` is a multi-purpose command that includes multiple possible subcommands. Usage is simple:

```
'Here is an example using SIK. and GIK.

' Sent Byte 1 of Robots own "Pool" to 10
SIK.Byte|1|10

' Get the actual Value of "own Pool" value Nr.1
GIK.Byte|1||$$RES

' Print it
DBP.received: $$RES
```

Syntax

SIK.P1|P2[|P3][|P4]

Parameter Explanation

P1 – `Subcommand`, the following Subcommands are currently supported. Note that for all of the subcommands there is a long form that is better readable, as well as a short form, that can be used to write faster.

- "Byte","B."
  `SIK.Byte|Nr.|$$VAL[|CUN]`

*© 2023 Theo Gottwald * https://www.smart-package.com*

- "Long","L."
  `SIK.Long|Nr.|$$VAL[|CUN]`

- "Quad","Q."
  `SIK.Quad|Nr.|$$VAL[|CUN]`

- "Float","F."
  `SIK.Float|Nr.|$$VAL[|CUN]`

- "In Use","iu."
  `SIK.InUse|$$VAL[|CUN]`

- "Handle","h."
  `SIK.handle|$$VAL[|CUN]`

- "Message to","mt."
  `SIK.Message to|CUN|$$MSG`
  Technically a normal Message is a qued message, therefore this command does the same as the following command.

- "Qued Message","qms.","q-msg"
  `SIK.qued Message|CUN|$$MSG`
  Sends a Message to the Inbox of the receiver CUN. This message will be qued and the Inbox will be free for other Messages from other senders.

- "Direct Message","dms."
  `SIK.direct Message|CUN|$$MSG`
  A "Direct Message will not be qued, therefore it will block the Inbox of the receiving CUN until it is taken by the receiving CUN, using `GIK`..

Depending on the chosen Subcommand the further Parameters **P2** to **P4** differ and may be
needed or not.

P2 - Depending on the chosen Subcommand this Parameter may be the number of the synchronization element or may have another usage.

P3 - **`(optional)`** Depending on the chosen Subcommand this Parameter may be needed or not.

P4 - **`(optional)`** Depending on the chosen Subcommand this Parameter may be needed or not.

Example

```
' This example uses INTERKOM to signal a second Robot to
' end running in a looping.
'

' Run Parallel Robot
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRR.
  ' Get InterKOM CUN of father-robot
  $$CUN=#fcun#
  ' Here is the Loop
  DOL.
    GIK.Byte|1|$$CUN|$$RES
    PRT.received: $$RES
    PAU.0.5
  OOP.$$RES=10
  MBX.Ending.
  ENR.
PRE.
' Here is the Main Robot
PAU.0.2
' Set the Pool-Byte to 10 this will make the Parallel-Robot
' end its loop.
SIK.Byte|1|10
' Receive the byte also
GIK.Byte|1||$$RES
DBP.received: $$RES
MBX.Signal sent.
ENR.
```

Remarks

There are multiple examples on InterKOM Commands within the supplied Sample Scripts.

Limitations:

Currently there are limitations to the maximum size that a message may have, using InterKOM. These are 650 kb for this release. This may change in the future.

Also the number of possible InterKOM Clients is limited to 16 at this time.

See also:

- ! Smarty's InterKOM Technology 2116
- GIK. - Get InterKOM 2122
- WIK. - Wait InterKOM 2142
- IIK. - If InterCOM Condition 2125

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.6 SRV. - Set Return Value (Executable)

SRV. - Set Return Value (Error-Level) Previous 2131 Top 41 Next 2136

MiniRobotLanguage (MRL)

SRV. Command

Set a Return Error-Level Value in EXE-Mode

Intention

This command will set the Return code for compiled executable scripts. The Return code is also known as Error Level. You can call this command at any place in your script.

If called in "Exemode", `SRV.` will set a return code for a compiled executable. If not encapsuled in an executable, `SRV.` will set the Returncode like `SEC.` only for the "`MR01.exe`". If given no Parameter, a Return code of zero is used.

Even multiple times. However only the last call will set the finally used return code. Usage is simple:

```
' Set a Return Code of zero (0)
SRV.

' Set a Return Code of zero (0)
SRV.0

' Set a Return Code of 10005
: $$PRO=10005
SRV.$$PRO

' Add multiple results
SRV.($$PR1+$$PR2+$$PR3)
```

`SRV.` does NOT set a Return code for the "`MR01.exe`", use `SEC.` for this.

Syntax

SRV.[P1]

Parameter Explanation

P1 - **`(optional)`** `NUM` ,variable or formula in (..).
if omitted, default is 0. Will be the Return code of the compiled executable.

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Set a Return Code of zero (0)
SRV.

' Set a Return Code of zero (0)
SRV.0

' Set a Return Code of 10005
: $$PRO=10005
SRV.$$PRO
```

Remarks

This command will only set the Return Value for the "**MR01.exe**".
To set the Error-Level of compiled Scripts (EXE-Mode) use the `SRV.`-command.
To end a script and set a Return-code for executables, use `END`.

Limitations:

-

See also:

- 1.B Error handling 224
- SEC. - Set Exit Code 2129
- END. - End Script 2362
- ENR. - End Return 2169

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.7 TWM. - Text-Window-Message

MiniRobotLanguage (MRL)

TWM. Command

Text Window Message

Intention

Send a Text-Message to another running robot instance, which must be in receiving mode, using the `WFM.` command. Usage is simple:

```
' Sent Text to last localized Window use "19" as dwdata
TWM.Text

' Sent Text to robot $$CMH, use 19 (default) as dwdata
TWM.Text|$$CMH

' Sent Text to Windowhandle in $$MYW, use 21 as dwdata
TWM.Text|$$MYW|21
```

As the `PRV.` command is much more easy to use, for "Inter-Robot Communication", its the preferred way to exchange data between robots.

The `TWM.` / `WFM.` method is more complicated, and should be used primarily to communicate with external programs.

This command can be used to communicate (send messages) to WinRobots compatible automation-systems.

Syntax

TWM.P1[|P2][|P3]

Parameter Explanation

P1 - Text/String to send to the other robot.

P2 - ***(optional)*** Receiver/Target window handle (communications handle). If omitted, last localized window is been taken as receiver.

P3 - ***(optional)*** dwData, a number, which is 19 by default. It's recommended to ignore this number and omit this parameter, unless you now what you do.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Please note that you can get the window-handle of this instance by using the #cbw# Systemvariable.

Example

```
'*******************************************
' TWM./WFM. - Sample
' Shows that the Parallel-Robot starts with
' all variables as they have been at starting-time
' of the starting robot.
' It can also use all Subroutines in this file.
'*******************************************
VAR.$$INP=0
VAR.$$CMH=#cbw#
PRT. My Name is Robbi $$CMH.
'-----------------------------------------------------------
PRR.
  PRT.I am Parallel-Robot Robbi: #cbw#
  PRT. I am going to send to: $$CMH
  PAU.2
  TWM.#This is my message!#|$$CMH
  PRT.Parallelrobot Ending.
  PAU.12
  END.
PRE.$$RET|0
PRT. I am waiting for a Message from Robbi: $$RET
'-----------------------------------------------------------
WFM.md|$$WAI|19
MBX.I got: $$WAI
END.
'-----------------------------------------------------------
```

Remarks

To implement a communication between the SPR and an external program, you can use the following Sub-Programs (BASIC-Code). The communication is done using the WM_COPYDATA-Message.

```
'#############################################################
####
'#
'#############################################################
####
' T01 - hwnd, T02 - Text$,T05 - eigenes Empfänger-Handel, T03
- optionale Zahl,B01 - wenn >0 dann wird Postmessage statt
Sendmessage verwendet
FUNCTION PM_SendPM(BYVAL T01 AS DWORD,BYVAL T02 AS
STRING,BYVAL T05 AS DWORD,OPT BYVAL T03 AS LONG,OPT BYVAL B01
AS BYTE) AS LONG
    REGISTER R01 AS DWORD,R02 AS LONG
    LOCAL S01 AS STRING
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    ' T05 - Own Handle
    S01="TPMS"+MKDWD$(T05)+T02
     ' hTarget, dwdata-Zahl,Text$,Send/Post
    R02=PM_SendPrivateMsg(T01,T03,S01,B01)
     enx:
     FUNCTION = R02
END FUNCTION
'################################################################
####
'#
'################################################################
####
' Returns 0 oder 1 (sucess)
' Format: "TPMS"+MKDWD$(Sender)+MSG$
 FUNCTION PM_GetPM(BYREF PrivateMsg AS STRING, BYVAL wParam AS
LONG, BYVAL lParam AS LONG, BYREF sender AS DWORD,BYREF dwdata
AS LONG) AS LONG
   REGISTER R01 AS LONG
   LOCAL S01,S02 AS STRING
   dwdata=PM_GetPrivateMsg(S01,wParam,lParam)
   S02=LEFT$(S01,4)
   IF S02="TPMS" THEN
      S02=MID$(S01,5,4):Sender=CVDWD(S02)
      IF IsNoWin(Sender) THEN R01=0:GOTO ero
      PrivateMsg=MID$(S01,9)
   R01=1
   ELSE
     R01=0:GOTO ero
   END IF
   ero:
   FUNCTION = R01
 END FUNCTION
'################################################################
####
'#
'################################################################
####
'
FUNCTION PM_SendPrivateMsg(BYVAL hTarget AS LONG, BYVAL UseMsg
AS LONG, BYVAL Buff AS STRING,OPT BYVAL B01 AS BYTE) AS LONG
    REGISTER LenBuff AS LONG, R02 AS LONG
    DIM Buf(1) AS LOCAL BYTE
    LOCAL cds AS COPYDATASTRUCT
    LenBuff = LEN(Buff)
    REDIM Buf(LenBuff) AS BYTE
    POKE$ VARPTR(Buf(0)), Buff
    cds.dwData = UseMsg
    cds.cbData = UBOUND(Buf) - LBOUND(Buf) + 1
    cds.lpData = VARPTR(Buf(0))
    IF zero(B01) THEN
        R02 = SendMessage(hTarget, %WM_COPYDATA,
LenBuff,VARPTR(cds))
     ELSE
        PostMessage(hTarget, %WM_COPYDATA,
LenBuff,VARPTR(cds))
        R02=1
     END IF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    FUNCTION=R02
END FUNCTION
'#################################################################
####
'#
'#################################################################
####
FUNCTION PM_GetPrivateMsg(BYREF PrivateMsg AS STRING, BYVAL
wParam AS LONG, BYVAL lParam AS LONG) AS LONG
    LOCAL cds AS COPYDATASTRUCT
    DIM Buf(1) AS LOCAL BYTE
    REDIM Buf(wParam) AS BYTE
    CALL MoveMemory(cds, BYVAL lParam, SIZEOF(cds))
    CALL MoveMemory(buf(0), BYVAL cds.lpData, cds.cbData)
    PrivateMsg = PEEK$(VARPTR(buf(0)), wParam)
    FUNCTION = cds.dwData
END FUNCTION
'#################################################################
####
'#
'#################################################################
####
```

Limitations:

-

See also:

- WFM. - Wait-For-Message 2140
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- PRV. - Parallel-Robot send Variable 2651
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861
- TWM. - Text-Window-Message 2136

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.8 WFM. - Wait-For-Message

MiniRobotLanguage (MRL)

WFM. Command

Wait For Message

Intention

Wait for a Text-Message sent from another running robot instance, using `TWM`.
This command is only for advanced use.

Hint: You can directly manipulate variables in other running robots, using `PRV`.

Syntax

WFM.P1|P2[|P3]

Parameter Explanation

P1 **-** Prototypes

```
t - (in) timeout in ms
d - (out) dwdata
h - (out) sender window-handle - must be variable
s - (out) 1/0 if the timeout was flagged
m - (out) Text which has been received
f - (in) filter, if used only sender with these
     communication-handle will be accepted. All other
     messages will be dropped.
```

P2 **-** Data depending on Prototypes

P3 **-** Data depending on Prototypes

```
Please note that you can get the communication-handle of the
actual robot-instance by using the #cbw# Systemvariable.
The most easy way to get the communication handle from the
parent-robot is by variable inheritance (see example below).
You can also get a commandline-parameter with the handle from
the instance which has started this instance using $cmdl$.
This is the way to go if you use a WinRobots compatible robot
as father.
```

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'*******************************************
' TWM./WFM. - Sample
' Shows that the Parallel-Robot starts with
' all variables as they have been at starting-time
' of the starting robot.
' It can also use all Subroutines in this file.
'*******************************************
VAR.$$INP=0
VAR.$$CMH=#cbw#
PRT. My Name is Robbi $$CMH.
'-----------------------------------------------------------
PRR.
  PRT.I am Parallel-Robot Robbi: #cbw#
  PRT. I am going to send to: $$CMH
  PAU.2
  TWM.#This is my message!#|$$CMH
  PRT.Parallelrobot Ending.
  PAU.12
  END.
PRE.$$RET|0
PRT. I am waiting for a MEssage from Robbi: $$RET
'-----------------------------------------------------------
WFM.md|$$WAI|19
MBX.I got: $$WAI
END.
'-----------------------------------------------------------
```

Remarks

-

Limitations:

Actually `WFM.` can not receive messages from other, for example WinRobots compatible automation systems. It will only receive messages in the proper message format.

See also:

- TWM. - Text-Window-Message 2136
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- PRV. - Parallel-Robot send Variable 2651
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861

© 2023 Theo Gottwald * https://www.smart-package.com

3.58.9 WIK. - Wait InterKOM

MiniRobotLanguage (MRL)

WIK. Command

Wait for a IPC-Message or Signal

Intention

Using `WIK.` you can wait until a defined value or mesage is set/sent by any other InterKOM program, for example using `SIK.`

InterKOM is like you have a shared pool of variables. As InterKOM is lightning fast, synchronization will easily be possible this way.
For more Informations on InterKOM see here ! Smarty's InterKOM Technology 2116.

Using `SIK.` you can write to this shared memory, and at the same time other processes can read the writings using `GIK.` or wait for it using `WIK.` or `IIK.` and react on that.

`WIK.` is a multi-purpose command that includes multiple possible subcommands. Usage is simple.

```
' Wait unlimited time until our own Pool-Byte 1 contains
' a value that is >15
WIK.|Byte|1|>15

' Wait at most 15 seconds until our own Pool-Byte 1 contains
' a value that is NOT EQUAL 27
WIK.15|Byte|1|!27

' Wait at most 0.7 seconds until our own Pool-Long 1 contains
' a value that is smaller then 27012345
WIK.0.7|Long|1|<27012345

' Wait unlimited time
' until CUN from last started Parallelrobot is not more alive
$$PRR=#pcun#
WIK.|!Alive|$$PRR

' Note: This will do the same
WIK.!|Alive|$$PRR
```

In this more complex example we start a Parallel-Robot to set a Synchronisation Number in the shared Pool. Note that we also use a variable to define the subcommand.

```
$$SBC=Float
PRR.
  PAU.2
  SIK.$$SBC|1|6.5454|#fcun#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  MBX.Signal Sent!
  ENR.
PRE.

$$NUM=6.5453
WIK.15|$$SBC|1|>$$NUM
MBX.Ende
ENR.
```

Syntax

WIK.[!]P1|P2[|P3][|P4]

Parameter Explanation

[!] – If you prefix the time-value with a "**!**" then the Logic is negated and the given condition is true, if its false..

P1 – **[optional]** If given, this is the maximum time to wait for the condition to be true. If the condition does not become true, then the command will set the timeout-flag and just continue with the script. You can check the timeout-flag to immediately after this command, to see if an timeout has occurred.

P2 – `Subcommand`, the following Subcommands are currently supported. Note that for all of the subcommands there is a long form that is better readable, as well as a short form, that can be used to write faster.

- "Byte","B."
  `WIK.[!][max time]|Byte|Nr.|cond/value[|CUN]`
- "Long","L."
  `WIK.[!][max time]|Long|Nr.|cond/value[|CUN]`
- "Quad","Q."
  `WIK.[!][max time]|Quad|Nr.|cond/value[|CUN]`
- "Float","F."
  `WIK.[!][max time]|Float|Nr.|cond/value[|CUN]`
- "Use","u."
  `WIK.[!][max time]|Use|CUN`
- "Alive","a."
  `WIK.[!][max time]|Alive||CUN`
- "Message from","mf."
  `WIK.[!][max time]|Message from|CUN[|$$MSG]`

© 2023 Theo Gottwald * https://www.smart-package.com

- "Message any","ma."
  ```
  WIK.[!][max time]|Message any|[$$MSG]
  ```

Depending on the chosen Subcommand the further Parameters **P3** to **P5** differ and may be
needed or not.
If you prefix the Subcommand or the max-time value with a "**!**" this will negate the logic of the command. If you prefix both, the logic will be unchanged.

cond - WIK. supports the following conditions:

! - NOT

= - Is Equal to

> - Larger then

< - Smaller then

Hint: Combined Operators like "**>=**" are not supported. In this case, just use ! and "<"
and switch the operators. Because **A>=B** is the same as **NOT B<A**.
Example:

```
WIK.!|Long|$$NUM|<12
WIK.|Long|4|>$$NAU
WIK.5|Long|2|=$$NAC
WIK.!30|Long|$$NUM|=$$NAC
```

P2 – Depending on the chosen Subcommand this Parameter may be the number of the synchronization element or may have another usage.

P3 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

P4 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

P5 – **(optional)** Depending on the chosen Subcommand this Parameter may be needed or not.

Example

```
' This example uses INTERKOM to wait until a second Robot
' has ended.

' Run Parallel Robot
PRR.
  PAU.4
  ENR.
PRE.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' This Script will wait until the Parallel-Robot has ended
WIK.!|Alive|#pcun#
MBX.Ende
ENR.
```

Remarks

There are multiple examples on InterKOM Commands within the supplied Sample Scripts.

Limitations:

Currently there are limitations to the maximum size that a message may have, using InterKOM. These are 650 kb for this release. This may change in the future.

Also the number of possible InterKOM Clients is limited to 16 at this time.

See also:

- ! Smarty's InterKOM Technology 2116
- GIK. - Get InterKOM 2122
- SIK. - Set InterKOM 2131
- IIK. - If InterCOM Condition 2125

© 2023 Theo Gottwald * https://www.smart-package.com

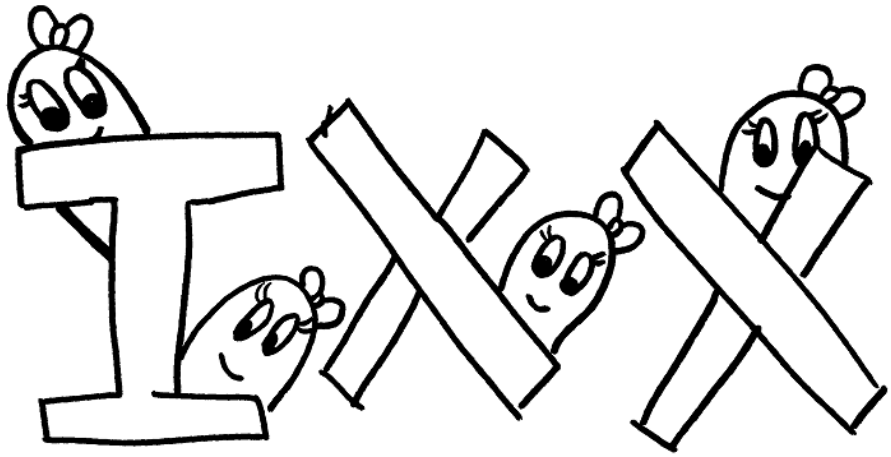
3.59 IXX. - Choose one of two cases

3.59.1 ! Smarty's IXX-Commands

Smarty's IXX-Commands

"Two sided decision". In other languages its called IF .. THEN ... ELSE ... ENDIF.
Here we have only 3 Letters so we use Ixx ... ELS. ... . Exx.

There are many more Ixx.-Commands not in this help-Chapter but in the other chapters.



Sometimes in life we have to make a decision. We must decide what we do next.
Decisions are most important because its in the moments of decision, that our life will be shaped.

The decisions we are talking about right now, are "**two-way-decisions**".

Do we choose A - or do we choose B?

Of course Smarty can also make decisions. And she likes to make decisions!
Her decisions are mostly based on what you can see on the screen.
But she can also decide based on texts, numerical values or system parameters.

Just take a look in the list of `IXX`.-commands.

There is one general thing to know. All `IXX`. Commands have a negative for that is called `NXX`.

You can learn more about decisions here:

1.6.1. Program Flow Control 121

Most of the `IXX`. are themed and therefore in the themed chapters.

*© 2023 Theo Gottwald * https://www.smart-package.com*

For example these IXX.-commands are outside the IXX-Chapter, but there are lots more.

- IEF. / NEF. - If - exist - File 1050
- ISC. - If Servic Command 2552
- ISI. / NSI. - If Service Installed 2557
- IIK. - If InterCOM Condition 2125
- IPC. / NPC. - If-Pixel-Colour 1816
- ILC. / NLC. - If Line Color 1805
- IRC. / NRC. - If Rect Color 1828
- IRA. / NRA. - If Rect altered 1824
- IAI. / NAI. - If-Application-Installed 2601
- IDC. / NDC. - If Date Condition 709
- IIK. - If InterCOM Condition 2125
- IRS. / NRS. - If Result available 434
- and much more ...

© 2023 Theo Gottwald * https://www.smart-package.com

3.59.2 EIF. - END IF

MiniRobotLanguage (MRL)

EIF. Command

End of an IF Block

Intention

Used in Conditional statements to close the statement-block.
Generally all `IXX.` commands support `ELS.` and `EIF.`

There is an more detailed explanation here: 1.6.1. Program Flow Control 121.

Syntax

EIF.

Parameter Explanation

-

Speed in Ticks:

This command uses typically around 3 to 9 Ticks.

Example

```
IEF.c:\MyFile.txt
     MBX. the file exists!
EIF.
```

Remarks

-

Limitations:

-

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

- 1.6.1. Program Flow Control 121
- ELS. - ELSE 2150

© 2023 Theo Gottwald * https://www.smart-package.com

3.59.3 ELS. - ELSE

MiniRobotLanguage (MRL)

ELS. Command

Else in IF Block

Intention

Used in Conditional statements to separate the blocks, being executed if the condition is met from the statement-block that is being executed when the condition is not met.

```
IEF.c:\MyFile.txt
   MBX. the file exists!
ELS.
   MBX. the file does not exist!
EIF.
```

There is an more detailed explanation here: 1.6.1. Program Flow Control 121.

Syntax

ELS.

Parameter Explanation

Speed in Ticks:

This command uses typically around 5 Ticks.

Example

```
See examples in the IFX Commands
```

Remarks

-

Limitations:

 *© 2023 Theo Gottwald * https://www.smart-package.com*

-

See also:

- 1.6.1. Program Flow Control 121
- EIF. - END IF 2148

© 2023 Theo Gottwald * https://www.smart-package.com

3.59.4 IFD. / NFD. - If File Date

MiniRobotLanguage (MRL)

IFD. Conditional Statement

If File Date

Intention

Check if a filedate is older then ... or newer then whatever you specify.
Used to compare version information from files.

In ***P4*** - Options, you specify which filedate to compare

```
a - last-access
c - creation date
w - last write (default)
```

Is the installed DLL or file older or newer?
This command will help you to solve this problem.

As parameters you can use the real filenames (with path) or the version-string.

```
' Syntax 1 does need | (separator-pipe)
IFD.?path\PR01.Exe|>=|?path\u9.exe|w
  PRT. File 1 is newer or same date
ELS.
  PRT. File 2 is newer
EIF.

' Syntax 2 does not need | (separator-pipe)
' but must therefore use Variables
IFD.$$FIA>$$FIB|c
  PRT. File 1 is newer
ELS.
  PRT. File 2 is newer
EIF.
```

You can also use the negative form of this statement: `NFD.` (If Not File Date)

Syntax

IFD.P1|P2|P3|P4]

Parameter Explanation

P1 - Filename 1 or version 1

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - compare-operation:
> < = ! => <=
P3 - Filename 2 or version 2

P4 - Options
a - last-access
c - creation date
w - last write (default, if omitted)

Example

```
''*******************************************
' IFD. - Sample
'*******************************************
'DBM.2
' Use Path to a existiing files
VAR.§§FIA=?exepath\MR01.Exe
VAR.§§FIB=?exepath\MRE.exe
PRT.=========================================
GFD.§§FIA
PRT.Date File 1: $$000
GFD.§§FIB|§§RES
PRT.Date File 2: §§RES

' Syntax 1 must use Variables
IFD.§§FIA>§§FIB|c
 PRT. File 1 is newer
ELS.
 PRT. File 2 is newer
EIF.

' Syntax 2 can use Text-Filenames
IFD.?path\PR01.Exe|>=|?path\u9.exe|w
 PRT. File 1 is newer or same date
ELS.
 PRT. File 2 is newer
EIF.

' We can compare files or File-Version
' Supported Operators:
' >,<,=,=>,<=,!
' In this case we compare date-strings
' these must have the format which is returned from
' GFD!
GFD.§§FIA|§§REA
GFD.§§FIB|§§REB
PRT. File 1: §§REA
PRT. File 2: §§REB
IFD.§§REA>§§REB|c
 PRT. Is the case
ELS.
 PRT. Is not the case
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.

PRT.=========================================
DMP.4
MBX.Ready
END.
```

Remarks

If you need to get the filedate, use `GFD.`

Limitations:

-

See also:

- **GFD. - Get File Date** 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137

© 2023 Theo Gottwald * https://www.smart-package.com

3.59.5 IFL. / NFL. - If File Language

MiniRobotLanguage (MRL)

IFL. / NFL. Conditional Statement

If File Language

Intention

Conditional Statement. Test if a specified **file** has the specified file-language property.

You can specify the right side of the equation in one of three formats:

```
t - Text-Result
c - Codepage
k - Keyboard code

' IFL.-Example
'
' First we define the file that is been analyzed.
'
$$FIL=?exepath\catseye.dll

' Now we use GFL. to display the file-language
' Information, in the Debug-Window of the Editor.
GFL.$$FIL|$$RES|c
DBP.$$RES

GFL.$$FIL|$$RES|t
DBP.$$RES

' In this case we use two variables
$$LAN=04E4
IFL.$$FIL=$$LAN
  DBP.Language is German
EIF.

' In this case we use a variable LEFT and a
' Numeric argument on the RIGHT
' We use the default option that is "|c"
' if no option is specified
IFL.$$FIL=0409
  DBP.Language is English I
EIF.

' This is just the same, but the |c is specified.
IFL.$$FIL=0409|c
  DBP.Language is German II
EIF.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Here we use a textual compare,
' You can use 3.2 Standard-Search Pattern
IFL.$$FIL=engli|t
  DBP.Language is English III
EIF.

GFL.$$FIL|$$RES|k
DBP.$$RES
END.
```

You can use NFL.(Not File Language) as the negative Form of the command.

Syntax

IFL.P1opP2[|P3] … ELS. … EIF.

Parameter Explanation

P1 - Variable contains a valid path and filename.
Must be a Variable, you can NOT specify
a filename inline as **P1**.

P2 - Variable or file-language description,
can be a variable or a description, depending on the selected
P3.

op - ***Operator***, can be: **=** ***(equal) or*** **!** **<>** **><** ***(not equal)***

P3 - can be one of these letters:

```
t - Text-Result
c - Codepage
k - Keyboard code
```

if omitted. "c" is the default option.

Example

```
'*******************************************
' IFL.-Example
'*******************************************
' First we define the file thar is been analyzed.
'
$$FIL=?exepath\catseye.dll

' Now we use GFL. to display the file-language
' Information, in the Debug-Window of the Editor.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GFL.$$FIL|$$RES|c
DBP.$$RES

GFL.$$FIL|$$RES|t
DBP.$$RES

' In this case we use two variables
$$LAN=04E4
IFL.$$FIL=$$LAN
  DBP.Language is German
EIF.

' In this case we use a variable LEFT and a
' Numeric argument on the RIGHT
' We use the default option that is "|c"
' if no option is specified
IFL.$$FIL=0409
  DBP.Language is English I
EIF.

' This is just the same, but the |c is specified.
IFL.$$FIL=0409|c
  DBP.Language is German II
EIF.

' Here we use a textual compare,
' You can use 3.2 Standard-Search Pattern
IFL.$$FIL=engli|t
  DBP.Language is English III
EIF.

GFL.$$FIL|$$RES|k
DBP.$$RES
END.
```

Remarks

`IFL.` and other IXX. can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX.` (Enumerations) to unlimited Depth.

Here are the numeric values for the country code:

```
0000 = Language Neutral
0400 = Process Default Language
0436 = Afrikaans
041C = Albanian
0401 = Arabic (Saudi Arabia)
0801 = Arabic (Iraq)
0C01 = Arabic (Egypt)
1001 = Arabic (Libya)
1401 = Arabic (Algeria)
1801 = Arabic (Morocco)
1C01 = Arabic (Tunisia)
2001 = Arabic (Oman)
2401 = Arabic (Yemen)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
2801 = Arabic (Syria)
2C01 = Arabic (Jordan)
3001 = Arabic (Lebanon)
3401 = Arabic (Kuwait)
3801 = Arabic (U.A.E.)
3C01 = Arabic (Bahrain)
4001 = Arabic (Qatar)
042B = Armenian. Unicode only (Windows 2000 and later).
044D = Assamese. Unicode only (Windows 2000 and later).
042C = Azeri (Latin)
082C = Azeri (Cyrillic)
042D = Basque
0423 = Belarussian
0445 = Bengali. Unicode only (Windows 2000 and later).
0402 = Bulgarian
0455 = Burmese
0403 = Catalan
0404 = Chinese (Taiwan Region)
0804 = Chinese (PRC)
0C04 = Chinese (Hong Kong SAR, PRC)
1004 = Chinese (Singapore)
1404 = Chinese (Macau)
041A = Croatian
0405 = Czech
0406 = Danish
0413 = Dutch (Netherlands)
0813 = Dutch (Belgium)
0409 = English (United States)
0809 = English (United Kingdom)
0C09 = English (Australian)
1009 = English (Canadian)
1409 = English (New Zealand)
1809 = English (Ireland)
1C09 = English (South Africa)
2009 = English (Jamaica)
2409 = English (Caribbean)
2809 = English (Belize)
2C09 = English (Trinidad)
3009 = English (Zimbabwe)
3409 = English (Philippines)
0425 = Estonian
0438 = Faeroese
0429 = Farsi
040B = Finnish
040C = French (Standard)
080C = French (Belgian)
0C0C = French (Canadian)
100C = French (Switzerland)
140C = French (Luxembourg)
180C = French (Monaco)
0437 = Georgian. Unicode only (Windows 2000 and later).
0407 = German (Standard)
0807 = German (Switzerland)
0C07 = German (Austria)
1007 = German (Luxembourg)
1407 = German (Liechtenstein)
0408 = Greek
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
0447 = Gujarati. Unicode only (Windows 2000 and later).
040D = Hebrew
0439 = Hindi. Unicode only (Windows 2000 and later).
040E = Hungarian
040F = Icelandic
0421 = Indonesian
0410 = Italian (Standard)
0810 = Italian (Switzerland)
0411 = Japanese
044B = Kannada. This is Unicode only (Windows 2000 and later).
0860 = Kashmiri (India)
043F = Kazakh
0457 = Konkani. This is Unicode only (Windows 2000 and later).
0412 = Korean
0812 = Korean (Johab)
0426 = Latvian
0427 = Lithuanian
0827 = Lithuanian (Classic)
042F = Macedonian
043E = Malay (Malaysian)
083E = Malay (Brunei Darussalam)
044C = Malayalam. Unicode only (Windows 2000 and later).
0458 = Manipuri
044E = Marathi. Unicode only (Windows 2000 and later).
0861 = Nepali (India). Unicode only (Windows 2000 and later).
0414 = Norwegian (Bokmal)
0814 = Norwegian (Nynorsk)
0448 = Oriya. Unicode only (Windows 2000 and later).
0415 = Polish
0416 = Portuguese (Brazil)
0816 = Portuguese (Standard)
0446 = Punjabi. Unicode only (Windows 2000 and later).
0418 = Romanian
0419 = Russian
044F = Sanskrit. Unicode only (Windows 2000 and later).
0C1A = Serbian (Cyrillic)
081A = Serbian (Latin)
0459 = Sindhi
041B = Slovak
0424 = Slovenian
040A = Spanish (Traditional Sort)
080A = Spanish (Mexican)
0C0A = Spanish (Modern Sort)
100A = Spanish (Guatemala)
140A = Spanish (Costa Rica)
180A = Spanish (Panama)
1C0A = Spanish (Dominican Republic)
200A = Spanish (Venezuela)
240A = Spanish (Colombia)
280A = Spanish (Peru)
2C0A = Spanish (Argentina)
300A = Spanish (Ecuador)
340A = Spanish (Chile)
380A = Spanish (Uruguay)
3C0A = Spanish (Paraguay)
400A = Spanish (Bolivia)
440A = Spanish (El Salvador)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
480A = Spanish (Honduras)
4C0A = Spanish (Nicaragua)
500A = Spanish (Puerto Rico)
0430 = Sutu
0441 = Swahili (Kenya)
041D = Swedish
081D = Swedish (Finland)
0449 = Tamil. Unicode only (Windows 2000 and later).
0444 = Tatar (Tatarstan)
044A = Telugu. Unicode only (Windows 2000 and later).
041E = Thai
041F = Turkish
0422 = Ukrainian
0420 = Urdu (Pakistan)
0820 = Urdu (India)
0443 = Uzbek (Latin)
0843 = Uzbek (Cyrillic)
042A = Vietnamese
```

Limitations:

-

See also:

- GFL.-Get File Language 1031
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- CTI. - Copy-To-Inifile 966
- CFI. - Copy-From_IniFile 951
- COP. - Copy File 957
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- SHF. - Shell-File-Operation 1082
- WFF. - Wait-For-File 946

© 2023 Theo Gottwald * https://www.smart-package.com

3.59.6 IFV. / NFV. If File Version

MiniRobotLanguage (MRL)

IFV. Conditional Statement

If File Version

Intention

Used to compare **version information** from files.
As parameters you can use the filenames (with path) or the version-string.

The negative Form of this statement is `NFV.`.

Syntax

IFV.P1|P2|P3

Parameter Explanation

P1 - Filename 1 or version 1
P2 - compare-operation:
> < = ! => <=
P3 - Filename 2 or version 2

Example

```
'*********************************************
' IFV. - Sample
'*********************************************
'DBM.2
' Use Path to a existiing files
VAR.$$FIA=?exepath\MR01.Exe
VAR.$$FIB=?exepath\MRE.exe
PRT.=======================================
GFV.$$FIA
PRT.Version File 1: $$000
GFV.$$FIB|$$RES
PRT.Version File 2: $$RES

' Syntax 1 must use Variables
IFV.$$FIA>$$FIB
 PRT. File 1 is newer
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
ELS.
 PRT. File 2 is newer
EIF.

' Syntax 2 can use Text-Filenames
IFV.?path\PR01.Exe|>=|?path\u9.exe
 PRT. File 1 is newer
ELS.
 PRT. File 2 is newer
EIF.

' We can compare files or File-Version
' Supported Operators:
' >,<,=,=>,<=,!
IFV.2.1.0.9|>|2.1.1.0
 PRT. Is the case
ELS.
 PRT. Is not the case
EIF.

PRT.=========================================
DMP.4
END.
```

Remarks

-

Limitations:

-

See also:

- GFD. - Get File Date 1023
- GFV. - Get File Version 1040
- IFD. / NFD. - If File Date 2152
- IFV. / NFV. If File Version 2161
- GSS. - GetSplitString 3137

© 2023 Theo Gottwald * https://www.smart-package.com

3.59.7 ILI. / NLI. - If LanguageID

MiniRobotLanguage (MRL)

ILI. Conditional Statement

If User or system LanguageID

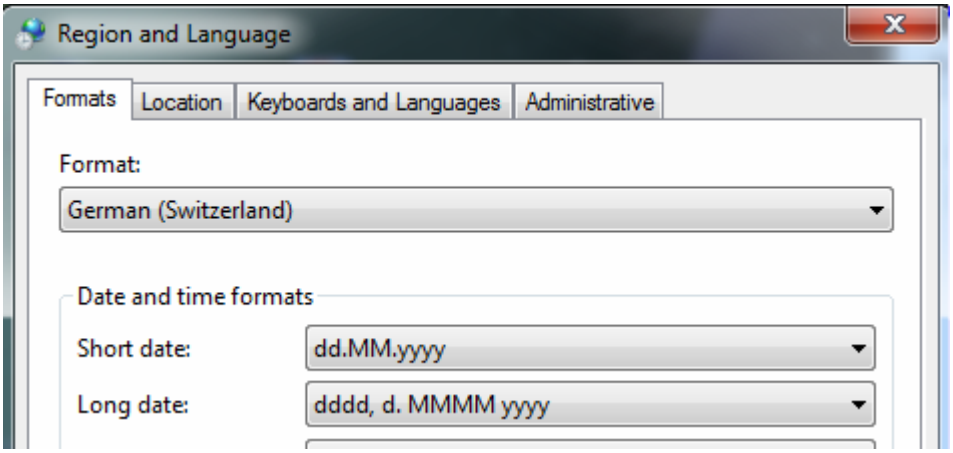
Intention

This command can be used to verify the language settings of the current user or system. Usage is simple:

```
' Try if we are running on a windows with german settings:
ILI.u|german
  DBP.We're on a german system
EIF.


' Try if we are running on a german windows:
ILI.s|german
  DBP.We're on a german system
EIF.
```

What is the difference between "u" and "s" ?

If you take a german windows and you set the Regional Format to Switzerland like this:



then you will get the following results:

```
' It uses german unicode settings
GLI.s
DBP.$$000
' result is 1031
ILI.s|1031
  DBP.True
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.

' This shows you that its set to swiss
GLI.u
DBP.$$000
' result is 2055
ILI.s|2055
  DBP.True
EIF.
```

You can use any combination of Language-ID's and languages descriptors:

```
ILI.u|german,english,1055
  DBP.Is valid
EIF.

ILI.u|1031,1055,1045
  DBP.Is valid
EIF.

NLI.s|fra,dan
  DBP.Is valid
ELS.
  DBP.wrong language!
EIF.
```

Important:
To get the Language ID a a number, use the `GLI.` - command.

Syntax

ILI.[P1][|P2]

Parameter Explanation

P1 - (optioal) defines the method which is used to get the Language ID.

U - Returns the language identifier for the current user locale.That is the Format settings (for date, time, currency, etc.) of current user in Regional and Language Options in Control Panel.
This is the default option is no letter is specified.

S - Returns the Language Identifier for the system locale. In fact you get the is the "Language for Non-Unicode programs" of current user.

C - an alternative system is used to get the Language ID for user. This supports more international properties providing a more culturally appropriate user experience.

© 2023 Theo Gottwald * https://www.smart-package.com

d - an alternative system is used to get the Language ID for the system (d),
This supports more international properties providing a more culturally appropriate
user experience.

P2 - (optioal) **Language-descriptor** or Language ID('s).

P2 can be any of these:

"core"
the english-speaking "core" countries:
USA, UK, Ireland, Australia, Canada, New Zealand
with the following Language-ID's: 1033,2057,3081,4105,5129,6153

"eng","englisch","english"
multiple english speaking countries, including these:
USA, UK, Ireland, Australia, Canada, New Zealand, South Africa, Jamaica, Caribbean, Belize, Trinidad, Zimbabwe, Philippines.

"ger","german","deutsch","deu"
all european german speaking countries:
Germany, Switzerland, Austria, Luxembourg, Liechtenstein
Using these Language ID's: 1031,2055,3079,4103,5127"

"spa","span","spanish","spanisch"
includes the 20 most important spanish speaking countries, worldwide.
Using these Language-IS's:
20490,19466,18442,17418,16394,15370,14346,13322,12298,11274,
10250,9226.8202,7178,6154,5130,4106,3082,2058,1034

"fra","french","franz","französisch"
Includes France, Belgium, Canada, Switzerland, Luxembourg, Monaco. Using these Language-ID's:
1036,2060,3084,4108,5132,6156"

"danish","dan","denmark"
Denmark - 1030

"sweden","swedish","schweden","schwedisch"
Sweden, Finland - 1053,2077

"ita","italy","italienisch","italian"
Italy, Switzerland - 1040,2064

© 2023 Theo Gottwald * https://www.smart-package.com

```
"dutch","nederland","niederländisch","netherlands"
Netherlands, Belgium - 1043,2067


"portuguese","por","port","portugal"
Portugal, Brazil - 1046,2070


"catalan"  - 1027
"basque"   - 1069
"galician" - 1110
```

or you can give any Language-ID or List of Language ID's separated by comma.

For a list of most language ID's see here:
Language-ID's 386

Example

```
'-----------------------------
'
'-----------------------------
ILI.u|german,english,1055
  DBP.Is valid
EIF.
ENR.
```

Remarks

You can find a more complete list with Language-ID's here:
Language-ID's 386

Often used Language-ID's (WEB-Link)

Limitations:

-

See also:

- GLI. - Get Language ID 2587
- GFI. - Get File Information 1026

© 2023 Theo Gottwald * https://www.smart-package.com

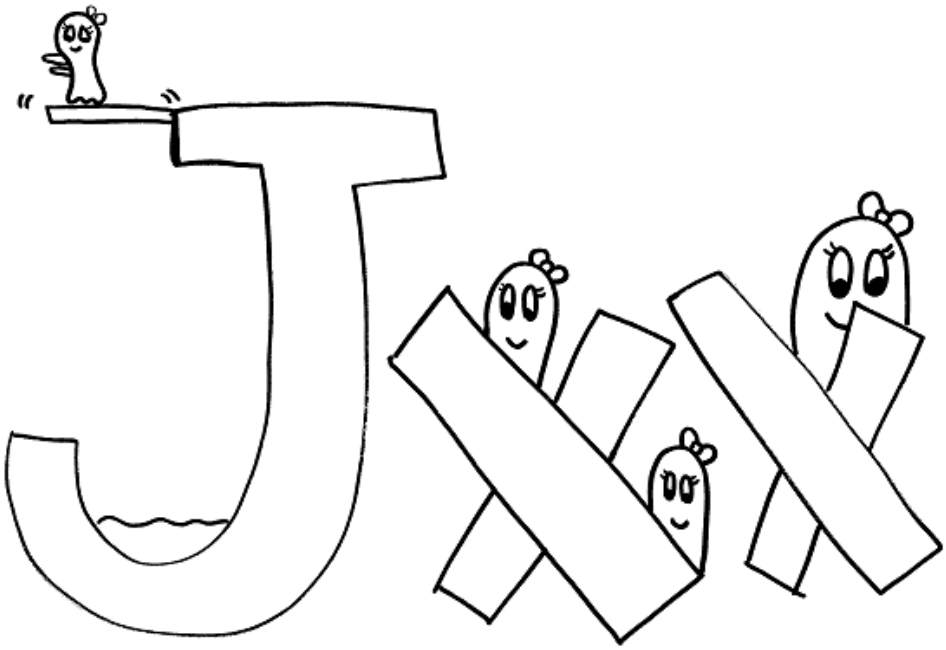
- GFD. - Get File Date 1023
- GFV. - Get File Version 1040

© 2023 Theo Gottwald * https://www.smart-package.com

3.60 JXX. - Jump - Instructions

3.60.1 ! Smarty's JXX-Commands

Smarty's JXX-Commands



JXX. are Jump commands.They can be:

- Conditional Jump-Commands
- Unconditional Jump-Commands

While the unconditional Jump-Commands will always jump to the given Label/Target, the conditional Jumps will only jump, if the condition is met. Else they will not jump.

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.60.2 ENR. - End Return

MiniRobotLanguage (MRL)

ENR. Command

Ends a script and Return to a calling script if there was one.

Intention

Mix between `END.` and `RTS.` Will execute an `RTS.` if possible, if not en `END.` will be executed.

End script execution and return to a calling script. Or if **P1** > 1 then return to the calling-script of the calling script of the ... (and so on).

Imagine you experience an error in a deeply nested subscript. Therefore you want to jump to a global error-handler in the Main script. Then you use the `ENR.(num)` option, to immediately jump up to the highest level.

Syntax

ENR.[P1]

Parameter Explanation

P1 - (optional) NUM , default 1. Count how many levels the "Return" should jump up.

Example

```
' Simply ends a Script or returns to a calling script.
ENR.

' Ends a Script and jumps TWO JNF.-Levels up.
ENR.2
```

Remarks

To end a script and set a Return-code, use `END.`
To set a Return code without ending the script, use `SEC.` (Set Exit Code).

If you try to Jump up to many JNF-Levels 2191 then the script will just end without an Error-Message. This is wanted behavior.

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Limitations:

-

See also:

- RTS. - ReTurn from Subroutine 2233
- 1.B Error handling 224
- END. - End Script 2362
- SEC. - Set Exit Code 2129
- JNF. - Jump New File 2191
- JNJ. - JumpNewFile-Jump to Label 2194

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.3 JIS. - Jump If String

MiniRobotLanguage (MRL)

JIS. Conditional Jump

Jump If String

Intention

This command can be used to test a string-variable against another string. Or it can be used to check if a string is empty or not. The easiest usage is like this:

```
JIS.$$TXT|Label
```

In this Mode, `JIS.` will jump to the Label, if the given string will evaluate to anything else the nothing (String-Len>0).

In Mode 2, you can compare string lexically. It looks like this:

```
' Test if the two Strings are equal
JIS.$$TXA=$$TXB|Label

' Test if the two Strings are unequal
JIS.$$TXA!$$TXB|Label

' Test if String P1 is lexically before String P3
JIS.$$TXA>$$TXB|Label

' Test if String P3 is lexically before String P1
JIS.$$TXA<$$TXB|Label
```

In this Mode **P1** and **P3** must be variables. These variables will only be resolved once. Specialfolders and Systemvariables will not be evaluated. This way the command is able to work also on binary data.

P2 must be any of the following 4 Operators.

>,<,=,!

When using the > or the < Operator, internally the strings are compared lexically.

For example, when using **>** , If the string in **P1** comes lexically before the string in **P3** then the evaluation will be true and the jump is been made.

Using **<** , If the string in **P3** comes lexically before the string in **P1** then the evaluation will be true and the jump is been made.

`JIS.` can be used as a conditional `JSR.` (Gosub like) if using ":" as a prefix to the Label). In normal Mode `JIS.` will always just make a `JMP.` (Goto like).

*© 2023 Theo Gottwald * https://www.smart-package.com*

JNS. generally evaluates to the opposite of JIS.

Hint: If you think that the evaluation is not correct, use

```
PRT.$$VAR|h
MBX.!
```

to check for invisible characters and Spaces. You will get a ASC-Dump like this, which shows all characters inside the variable with their Hex-ASC-Code.:

```
----------------------------------------
Len=7
(1)      48  61  6C  6C  6F  D  A
----------------------------------------
```

Syntax

```
JIS.P1|Label
JIS.P1|:Label

JIS.P1P2P3|Label
JIS.P1P2P3|:Label

JIS.P1|P2|P3|Label
JIS.P1|P2|P3|:Label
```

Parameter Explanation

P1 **-** variable, formula* in (..) or numeric value

P2 - condition sign: **>,<,=,!**
Use only one condition sign.

P3 - variable, formula* in (..) or numeric value

P4 – Label or variable that evaluates to a Label. If it contains the ":" a JSR./GSB. is done instead of a JMP.

Example

```
'************************************
' Simple JIS - Example
'************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: $$TXT=
' test if String is empty or not
JIS.$$TXT|Label
END.

:Label
MBX.String is not empty
ENR.

'************************************
' Check if a variable is "empty"
'************************************
: $$TXA=
JIS.$$TXA=|Label
END.

:Label
MBX.!
ENR.
```

Remarks

The available condition signs are: **>,<,=,!**

Use only one of these condition signs.
You can not combine condition signs like ">=" or "<>" etc.

Limitations:

-

See also:

- JNS. - Jump If No String 2200
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.4 JIT. - Jump if TimeOut

MiniRobotLanguage (MRL)

JIT. Conditional Jump

Jump If Timeout-Flag is set

Intention

This command can be used to test the TimeOut-Flag and Jump to a label if the TimeOut is set. Usage is simple:

```
STW.Xwt|0|nowi
JIT.ero
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
END.
```

You can use a variable that contains a label (dynamic jumping). Here is an example:

```
'************************************
' JIT - Example with dynamic jumping
'************************************
IVV.$$INP=1
  $$LAB=ero
ELSE.
  $$LAB=okp
EIF.
STW.Xwt|0|nowi
JIT.$$LAB
END.
:okp
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
END.
```

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

```
STW.Xwt|0|nowi
JIT.:ero
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
RET.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

JIT.P1
JIT.:P1

Parameter Explanation

P1 - Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like)
Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like)
Jump..

Example

```
'************************************
' JIT - Example
'************************************
STW.Xwt|0|nowi
JIT.ero
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
END.

'************************************
' Advanced JIT - Example (using :Label)
' this one makes a GSB. like-Jump.
'************************************
STW.Xwt|0|nowi
JIT.:ero
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
RET.

'************************************
' JIT - Example with dynamic jumping
'************************************
IVV.$$INP=1
  $$LAB=ero
ELSE.
  $$LAB=okp
EIF.
STW.Xwt|0|nowi
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
JIT.$$LAB
END.
:okp
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
END.
```

Remarks

There are two Systemvariables 334, representing the state of the timeout-flag:

```
#tio#    -        Timeout-Flag
#til#    -        Timeout-Line-Number
```

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- 3.1 Systemvariables 334
- JNT. - Fump If-No TimeOut 2203
- ITO. / NTO. - If TimeOut 2404
- STF. - Set TimeOut Flag 2410
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.5 JIV. - Jump-If-Variable

MiniRobotLanguage (MRL)

JIV. Conditional Jump

Jump If Variable

Intention

This command can be used to test a Variable for **>,<,=,!** against another variable or value or formula.

```
JIV.§§TAA<§§TAB|try
```

```
JIV.(§§TAA+5)<(§§TAB+3)|try
```

`JIV.` can be used as a conditional `JSR.` (Gosub like) if using ":" as a prefix to the Label). In normal Mode `JIV.` will always just make a `JMP.` (Goto like).

Syntax

```
JIV.P1P2P3|P4
JIV.P1P2P3|:P4
```

Parameter Explanation

P1 **-** variable, formula* in (..) or numeric value

P2 - condition sign: **>,<,=,!**
Use only one condition sign.

P3 - variable, formula* in (..) or numeric value

P4 – Label or variable that evaluates to a Label. If it contains the ":" a `JSR./GSB.` is done instead of a `JMP.`

```
*In case of JIV. the formula must never contain any of the
possible Sign-Operator (P2) or else the command may not be
properly interpreted.
```

Speed in Ticks:

This command uses typically around 250 Ticks.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'************************************
' Simple JIV - Example
'************************************
VAR.§§TAA=10
VAR.§§TAB=20
: §§LAB=try
:try
PRT.§§TAA
VIC.§§TAA
JIV.§§TAA<§§TAB|try
PRT.Hier
DMP.
MBX.!
END.


'************************************
' Advanced JIV - Example (using Formulas)
' as always, formulas must be in ().
'************************************
VAR.§§TAA=10
VAR.§§TAB=20
: §§LAB=try
:try
PRT.§§TAA
VIC.§§TAA
JIV.(§§TAA+5)<(§§TAB+3)|try
PRT.Hier
DMP.
MBX.!
END.
```

Remarks

The available condition signs are: **>,<,=,!**

Use only one of these condition signs.
You can not combine condition signs like ">=" or "<>" etc.

- Instead of **>=4** use **>3** instead.
- Instead of **<>** use **!**
- Instead of **<=5** use **<6**

The result is always the same.

Limitations:

JIV. does not have the Auto-Increment/ Auto-Decrement like for example JIZ.
If you need these, use JME. or JLE. instead.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.6 JIZ. - Jump-If-Zero

MiniRobotLanguage (MRL)

JIZ. Conditional Jump

Jump If Zero

Intention

This command can be used to test a Variable against zero and Jump to a label if the variable is zero. Here is a short example:

```
VAR.§§TST=0
:try
' just jump without other features
JIZ.§§TST|over
PRT.Hier
:over
DMP.
MBX.!
END.
```

Additionally the Command allows a "post-decrement" or alternatively a "post-increment" for easy Loop-Constructions. "Post-increment" will increment the given variable after the test inside the variable was done. Here is an example with "Post-increment":

```
' We'll count from -20 to 0
VAR.§§TST=-20
:try
JIZ.§§TST>over
PRT.Hier
JMP.try
:over
DMP.
MBX.!
END.
```

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an code-example:

```
' will jump to the Label "over" in Gosub-Mode
JIZ.$$NUM|:over

' we can use Labels that are in variables, too
: $$LAB=:over
JIZ.$$NUM|$$LAB
```

`JIZ.` is somehow the counterpart to JNZ. - Jump-if-not-zero 2206 .

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

JIZ.P1P2P3

Parameter Explanation

P1 - Must be a Variable, like $$AAA or §§BBB or $$001

P2 - ***can be an***
```
> increment-operator
< decrement operator or
| (Pipe Symbol) if no change of the value should happen
```

P3 - Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) (GOSUB like)
Jump is done instead of the normal JMP. - Jump to Label (GOTO like)
Jump..

Example

```
'************************************
' JIZ - Example
'************************************
VAR.§§TST=0
:try
JIZ.§§TST|over
PRT.Hier
:over
DMP.
MBX.!
END.

'************************************
' Advanced JIZ - Example (using :Label)
' this one makes a JSR.-Jump.
' Also we packed the Label into a Variable
'************************************
VAR.§§TST=0
: §§LAB=:over
:try
JIZ.§§TST|§§LAB
PRT.Here ...
DMP.
MBX.!
END.

:over
PRT.Was here first.
RET.
'************************************
' JIZ - Example with Post-Increment
'************************************
VAR.§§TST=-20
:try
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
JIZ.§§TST>over
PRT.Hier
JMP.try
:over
DMP.
MBX.!
END.

'************************************
' JIZ - Example with Post-Decrement
'************************************
VAR.§§TST=20
:try
JIZ.§§TST<over
PRT.Hier
JMP.try
:over
DMP.
MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.7 JLE. - Jump on Less or Equal

MiniRobotLanguage (MRL)

JLE. Conditional Jump

Jump on Less or Equal

Intention

This command can be used to test a Variable for being "**less or eqal**" against another variable, formula or value.

Additionally the Command allows a "post-decrement" or alternatively a "post-increment" for easy Loop-Constructions. "Post-increment" will increment the given variable after the test inside the variable was done. This can be used to construct simple Loops.
Here is an example with "Post-increment":

```
' Start-value
: §§NUM=1
: §§TAR=5
: §§LAB=lop
:lop
PRT.§§NUM
' 5 is the Loop-End
JLE.§§NUM>§§TAR|§§LAB
DMP.
MBX.!
END.
```

If you switch variables P1 and P2 you can test for "bigger then".

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

`JLE.` is the counterpart to JME. - Jump on More or Equal 2186.

Syntax

JLE.P1P2P3|P4

Parameter Explanation

P1 **-** must be variable

 *© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - can be an

```
> increment-operator
< decrement operator or
| (Pipe Symbol) if no change of the value should happen
```

P3 - variable or formula* in (..) or numeric value

P4 - Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like)
Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like)
Jump..

*In case of JLE. the formula must never contain any of the possible Sign-Operator (P2) or else the command may not be properly interpreted.

Example

```
'************************************
' Simple JLE - Example
'************************************
' Start-value
VAR.§§NUM=1
:lop
PRT.§§NUM
' 5 is the Loop-End
' > makes an post-increment to §§NUM
JLE.§§NUM>5|lop
DMP.
MBX.!
END.


'************************************
' Another JLE - Loop-Example
' using variables
'************************************
' Start-value
: §§NUM=1
: §§TAR=5
: §§LAB=lop
:lop
PRT.§§NUM
' 5 is the Loop-End
JLE.§§NUM>§§TAR|§§LAB
DMP.
MBX.!
END.

'************************************
' Another JLE - Loop-Example
' using variables and a formula
'************************************
' Start-value
: §§NUM=1
: §§TAR=5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: §§LAB=lop
:lop
PRT.§§NUM
' 5 is the Loop-End
JLE.§§NUM>(§§TAR+1)|§§LAB
DMP.
MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.8 JME. - Jump on More or Equal

MiniRobotLanguage (MRL)

JME. Conditional Jump

Jump on More or Equal

Intention

This command can be used to test a Variable for being "more or eqal" against another variable or value. Here is a simple example that also uses the "post-decrement" option to easily construct a loop.

```
' Start-value
VAR.$$NUM=10
'-------------------------
:lop
DBP.$$NUM
' We're using the post-decrement operator as P2
JME.$$NUM<1|lop
'-------------------------
END.
```

If you switch variables P1 and P2 you can test for "smaller then".

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

`JME.` is somehow the counterpart to JLE. - Jump on Less or Equal 2183 .

Syntax

JME.P1P2P3|P4

Parameter Explanation

P1 **-** must be variable

P2 **-** ***can be an***

```
> increment-operator
< decrement operator or
| (Pipe Symbol) if no change of the value should happen
```

P3 **-** variable or formula* in (..) or numeric value

P4 – Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like)

*© 2023 Theo Gottwald * https://www.smart-package.com*

Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like) Jump..

*In case of JME. the formula must never contain any of the possible Sign-Operator (P2) or else the command may not be properly interpreted.

Example

```
'************************************
' Simple JME - Example
' 1 is the Loop-End
' the < makes a post-decrement
'************************************
' Start-value
VAR.§§NUM=10
'-------------------------
:lop
PRT.§§NUM
JME.§§NUM<1|lop
'-------------------------
END.
```

```
'************************************
' Another JME - Loop-Example
' using variables
'************************************
' Start-value
: §§NUM=10
: §§TAR=5
: §§LAB=lop
:lop
PRT.§§NUM
' 5 is the Loop-End
VDC.§§NUM|2
JME.§§NUM|§§TAR|§§LAB
MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.60.9 JMP./GTO. - Jump to Label

JMP. - Jump to Label Previous 2186 Top 41 Next 2191

MiniRobotLanguage (MRL)

JMP./GTO. Unconditional Jump

JuMP to label

Intention

Continues Script Execution at the given Label (**GOTO**). You can type `GTO.` instead of `JMP.` both commands are technically equal.

You can jump to variables, if they contain valid labels at runtime.

Syntax

JMP.P1[|P2]

Parameter Explanation

P1 **-** `TXT (Label)`

P2 **-** `if Option "r" is given, a previous RET.`
`- will be removed from the JSR./GSB./RET. Stack.`

`JMP. may use variables, which must evaluate to a valid Label.`

Speed in Ticks:

This command uses typically between 10 to 20 Ticks.

Example

```
'*********************************************
' Example 1 - without Parameters,
' with local and global variables
'*********************************************
' This Example shows how to use JMP.
' Please run in Single-Step Mode/Editor.

: $$AAA=0
:A0
VIC.$$AAA
GTO.A$$AAA

:A1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP.LAB=A1
JMP.A0

:A2
DBP.LAB=A2
JMP.A0

:A3
MBX.Ready
```

Remarks

`JMP`. can use variables in the Parameter field.
You can type `GTO`. instead of `JMP`. both commands are technically equal.

Limitations:

There is no Limitation in the number of Labels you can Jump to. However, having thousands of Labels in your Script may slow down Script execution a bit.

See also:

- 1.6.1. Program Flow Control 121
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.10 JNF. - Jump New File

MiniRobotLanguage (MRL)

JNF. Unconditional Script-Call

Jump New File

Intention

Continues Script Execution in another File (Sub-Script).

Can use Parameters. You give the parameter separated with a | (Pipe) Symbol after the `JNF.`-call. Like this:

```
JNF.MySubscript.mrt|Param01|Param02
END.
```

In the Target script that is called, the parameters appear in:

`$$_00` -> number of given parameters (here 2)
`$$_01` -> first Parameter (here "Param01")
`$$_02` -> second Parameter (here "Param02")

You can use up to 25 Parameters in case you need them.
If you specify just Variables as Parameter, the Content is only copied,
and the Type of the Variable is preserved.

```
 ' Example
VAN.$$NUM=5.09
JNF.AddMe|$$NUM|$$NUX
ENR.

 ' File "AddME.mrt" in ?path\
VAO.$$PAA=$$_01
GVT.$$PAA|$$RES
MBX.We preserved the Variabletype: $$RES (=5, numeric)
ENR.
```

Can use script-local variables 366 that are local to each file.

`JNF.'s` can be nested to unlimited depth, even recursive.

Return to the calling Script with `RTS.`(ReTurn from Subroutine)-Command or `ENR.` (End-Return). This will restore the script-local variables of the calling script, including the parameter-variables.

If you need to leave a Sub-Program that was called with JNF. to a higher level then just the next, you can add a number behind the `RET.` Like this:

```
RET.2
```


© 2023 Theo Gottwald * https://www.smart-package.com

If you want to leave to the Top-Level in your script, just write:

```
RET.all
```

Syntax

JNF.P1[|P2..Px]

Parameter Explanation

P1 **-** Scriptfilename to call as subprogram
P2 - (optional) **-** up to 25 Parameters

Example

We need to have 2 Script Files for this simple Example.
Both need to be in the same folder.

The first Scriptfile, "First.txt" is the Main-Scriptfile.
It automatically defines ?path\ as its own path. That is, the folder where it is located.
It calls the second file "second.txt" which is also in "?path\" (in the same folder).

```
'*********************************************
' Example 1 - without Parameters,
' with local and global variables
'*********************************************
```

[First.txt]

```
: §§WHE=In First.txt
: $$GLO=1
PRT. $$WHE, Global=$$GLO
JNF.Second.txt
PRT. $$WHE, Global=$$GLO
END.
```

[Second.txt]

```
: §§WHE=In Second.txt
: $$GLO=2
PRT. $$WHE, Global=$$GLO
RTS.

'*********************************************
' Example 2 - with 2 Parameters
'*********************************************
```

[First.txt]

```
JNF.Second.txt|Hallo|$$GLO2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END.
```

[Second.txt]

```
PRT.§§_01 - §§_02
RTS.
```

Remarks

`§§_00` is the number of parameters given to the Script using `JNF.` or `JNJ.`
`§§_01 ... §§_25` are used for Parameters in the target Script file.

Limitations:

You can use up to 25 Parameters. If you need more, use Global variables or the Block-Commands (`SIB.` etc).

See also:

- **RTS. - ReTurn from Subroutine** 2233
- **ENR. - End Return** 2169
- 1.B Error handling 224
- END. - End Script 2362
- SEC. - Set Exit Code 2129
- JNJ. - JumpNewFile-Jump to Label 2194

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.11 JNJ. - JumpNewFile-Jump to Label

MiniRobotLanguage (MRL)

JNJ. Unconditional Script Call/Jump

Jump in New file and Jump to label

Intention

Continues Script Execution in another File optionally directly at a specified Label.

The second Parameter - if given - must be a Label, which is the Start-point in the other file.

Parameters P3 ..Px can be given as Parameter like in JNF. These parameters will be available in the other script file as §§_02 ...§§_25

Can use Parameters. You give the parameter separated with a | (Pipe) Symbol after the `JNJ`.-call. Like this:

```
JNJ.MySubscript.mrt|Startlabel|Param01|Param02
END.
```

In the Target script that is called, the parameters appear in:

```
Startlabel:
```

`§§_00` -> number of given parameters (here 2)
`§§_01` -> contains the Label (here "`Startlabel`")
`§§_02` -> first Parameter (here "Param01")
`§§_03` -> second Parameter (here "Param02")

You can use up to 24 Parameters in case you need them.

Can use script-local variables 366 that are local to each file.

`JNJ's and JNF.'s` can be nested to unlimited depth, even recursive.

Return to the calling Script with `RTS`.(ReTurn from Subroutine)-Command or `ENR`. (End-Return). This will restore the script-local variables of the calling script, including the parameter-variables.

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

JNJ.P1[|P2][|P3..Px]

Parameter Explanation

P1 **–** Scriptfilename
P2 - (optional) Label to Start at
P3 ... ***Px*** - up to 24 Parameters

Please note, that even if the "Goto 1st Parameter" has been done, §§_01 still contains the Label.
Therefore the first "real" Parameter is in §§_02.

Example

We need to have 2 Script Files for this simple Example.
Both need to be in the same folder.

The first Scriptfile, "First.txt" that is run defines ?path, which is the folder where it is. it calls the second file "second.txt".

```
'***********************************************
' Example 1 - without Parameters,
' with local and global variables
'***********************************************

[First.txt]
JNJ.Second.txt|Laba
PRT. Resultat ist: $$GLO
END.

[Second.txt]
:Labb
: $$GLO=2
JMP.Labc
:Laba
: $$GLO=3
:Labc
RTS.

'***********************************************
' Example 2 - with 2 Parameters
'***********************************************

[Main.txt]

JNJ.Sub.txt|LabC|Parameter 1
END.

[Sub.txt]

: §§AAA=This Command wont get executed!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:LabC
: §§AAA=This Command will get executed!
' The following Variable will get the Parameter 1, as §§_01
contains the Label.
: §§AAB=§§_02
RTS\
```

Remarks

`§§_00` is the number of parameters given to the Script using `JNF.` or `JNJ.`
`§§_01 ... §§_24` are used for Parameters in the target Script file.

Limitations:

You can use up to 24 Parameters with JNJ. If you need more, use Global variables or the Block-Commands (`SIB.` etc).

See also:

- **RTS. - ReTurn from Subroutine** 2233
- **ENR. - End Return** 2169
- **1.6.1. Program Flow Control** 121
- **JMP. - Jump to Label** 2189
- 1.B Error handling 224
- END. - End Script 2362
- SEC. - Set Exit Code 2129
- JNF. - Jump New File 2191

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.12 JNR. - Jump if no Result

MiniRobotLanguage (MRL)

JNR. Conditional Jump

Jump If No Result

Intention

This command can be used to test, if an specified Background- or Parallel-Robot Operation has emitted an result or not. Usage is simple:

```
SMB.
PAU.10
JRS.|result
MBX.Error we got no Result
ENR.
:result
MBX.Result is available
ENR.
```

You can use a variable that contains a label (dynamic jumping).
Here is another example:

```
SMB.
$$TIN=$$000
PAU.4
JNR.$$TIN|noresult
MBX.Yes we got an Result
ENR.
:noresult
MBX.No Result was available
ENR.
```

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

```
SMB.
:Retry
JNR.|:test
PAU.0.25
JNR.|Retry
ENR.
'--------------
:test
DBP. No Result yer!
PAU.0.1
RET.
```

`JRS.` is the positive version of `JNR.`

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

JNR.[P1]|P2

Parameter Explanation

P1 **-** ***(optional)*** Variable that evaluates to a valid Ticket-Number,
that is connected with a Background Operation or a Parallel Robot. If omitted
the last emitted Ticket-Number is used.

P2 **-** Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like)
Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like)
Jump..

Example

```
'************************************
' JNR - Example
'************************************
SMB.
:Retry
JNR.|:test
PAU.0.25
JNR.|Retry
ENR.
'--------------
:test
DBP. No Result yer!
PAU.0.1
RET.
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's Parallel Robot Operations 2632

© 2023 Theo Gottwald * https://www.smart-package.com

- JRS. - Jump if Result 2215
- GRS. - Get Result 430
- DRS. - Drop Result 427
- WRS. - Wait for Result 436
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- ! Smarty's Send-Mail 789
- SMB. Send Mail Background 799

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.13 JNS. - Jump If No String

MiniRobotLanguage (MRL)

JNS. Conditional Jump

Jump If No String

Intention

This command can be used to test a string-variable against another string. Or it can be used to check if a string is empty or not. The easiest usage is like this:

```
JNS.$$TXT|Label
```

In this Mode, `JNS.` will jump to the Label, if the given string will evaluate to nothing (String-Len=0).

In Mode 2, you can compare strings lexically. It looks like this:

```
' Test if the two Strings are NOT equal
' If they are equal JNS. will NOT Jump
JNS.$$TXA=$$TXB|Label

' Test if the two Strings are unequal
' If they are unequal JNS. will NOT Jump
JNS.$$TXA!$$TXB|Label

' Test if String P1 is lexically after String P3
JNS.$$TXA>$$TXB|Label

' Test if String P3 is lexically after String P1
JNS.$$TXA<$$TXB|Label
```

In this Mode **P1** and **P3** must be variables. These variables will only be resolved once. Specialfolders and Systemvariables will not be evaluated. This way the command is able to work also on binary data.

P2 must be any of the following 4 Operators.

>,<,=,!

When using the > or the < Operator, internally the strings are compared lexically.

Using **>** , If the string in **P3** comes lexically before the string in **P1** then the evaluation will be true and the jump is been made.

Using **<** , If the string in **P1** comes lexically before the string in **P3** then the evaluation will be true and the jump is been made.

© 2023 Theo Gottwald * https://www.smart-package.com

JNS. can be used as a conditional JSR. (Gosub like) if using ":" as a prefix to the Label). In normal Mode JNS. will always just make a JMP. (Goto like).

JIS. generally evaluates to the opposite of JNS.

Hint: If you think that the evaluation is not correct, use

```
PRT.$$VAR|h
MBX.!
```

to check for invisible characters and Spaces. You will get a ASC-Dump like this, which shows all characters inside the variable with their Hex-ASC-Code.:

```
----------------------------------------
Len=7
(1)      48  61  6C  6C  6F  D  A
----------------------------------------
```

Syntax

JNS.P1|Label
JNS.P1|:Label

JNS.P1P2P3|Label
JNS.P1P2P3|:Label

JNS.P1|P2|P3|Label
JNS.P1|P2|P3|:Label

Parameter Explanation

P1 - variable, formula* in (..) or numeric value

P2 - condition sign: **>,<,=,!**
Use only one condition sign.

P3 - variable, formula* in (..) or numeric value

P4 – Label or variable that evaluates to a Label. If it contains the ":" a JSR./GSB. is done instead of a JMP.

Speed in Ticks:

This command uses typically around 137 Ticks.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************
' Simple JNS. - Example
'************************************
: $$TXT=
' test if String is empty or not
JNS.$$TXT|Label
END.

:Label
MBX.String is empty
ENR.
```

Remarks

The available condition signs are: **>,<,=,!**

Use only one of these condition signs.
You can not combine condition signs like ">=" or "<>" etc.

Limitations:

-

See also:

- JIS. - Jump If String 2171
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.14 JNT. - Jump If-No TimeOut

MiniRobotLanguage (MRL)

JNT. Conditional Jump

Jump If Timeout-Flag is Not set

Intention

This command can be used to test the TimeOut-Flag and Jump to a label if the TimeOut is **Not** set. Usage is simple:

```
STW.Xwt|0|nowi
JNT.allOk
MBX.We are with an error
ENR.
'--------------
:ero
DBP. Everything is Ok
END.
```

You can use a variable that contains a label (dynamic jumping). Here is an example:

```
'************************************
' JNT - Example with dynamic jumping
'************************************
IVV.$$INP=1
  $$LAB=ero
ELSE.
  $$LAB=okp
EIF.
STW.Xwt|0|nowi
JNT.$$LAB
END.
:okp
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
END.
```

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

```
STW.Xwt|0|nowi
JNT.:ero
MBX.We are here
ENR.
'--------------
:ero
DBP. Anything is Ok!
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

JNT.P1

Parameter Explanation

P1 - Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like)
Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like)
Jump..

Example

```
'************************************
' JNT - Example
'************************************
STW.Xwt|0|nowi
JNT.allOk
MBX.We are with an error
ENR.
'--------------
:ero
DBP. Everything is Ok
END.

'************************************
' Advanced JNT - Example (using :Label)
' this one makes a GSB. like-Jump.
'************************************
STW.Xwt|0|nowi
JNT.:ero
MBX.We are here
ENR.
'--------------
:ero
DBP. Anything is Ok!
RET.

'************************************
' JNT - Example with dynamic jumping
'************************************
IVV.$$INP=1
  $$LAB=ero
ELSE.
  $$LAB=okp
EIF.
STW.Xwt|0|nowi
JNT.$$LAB
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:okp
MBX.We are here
ENR.
'--------------
:ero
DBP. AN Error has happened!
END.
```

Remarks

There are two Systemvariables 334, representing the state of the timeout-flag:

```
#tio#    -        Timeout-Flag
#til#    -        Timeout-Line-Number
```

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- 3.1 Systemvariables 334
- JIT. - Jump if TimeOut 2174
- ITO. / NTO. - If TimeOut 2404
- STF. - Set TimeOut Flag 2410
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.15 JNZ. - Jump-if-not-zero

MiniRobotLanguage (MRL)

JNZ. Conditional Jump

Jump is Not Zero

Intention

This command can be used to test a Variable for zero and Jump to a label if the variable is not zero. Here is a short example:

```
VAR.§§TST=-50
:try
VIC.§§TST
' just jump without other features
JNZ.§§TST|try
PRT.Hier
:over
DMP.
MBX.!
END.
```

Additionally the Command allows a "post-decrement" or alternatively a "post-increment" for easy Loop-Constructions. "Post-increment" will increment the given variable after the test inside the variable was done. Here is an example with "Post-increment":

```
VAR.§§TST=50
:try
DBP.Number is: §§TST
JNZ.§§TST<try
DBP.Hier
:over
DMP.
MBX.!
END.
```

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

```
' will jump to the Label "over" in Gosub-Mode
JNZ.$$NUM|:over

' we can use Labels that are in variables, too
: $$LAB=:over
JNZ.$$NUM|$$LAB
```

`JNZ.` is somehow the counterpart to JIZ. - Jump-If-Zero 2180.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

JNZ.P1P2P3

Parameter Explanation

P1 - Must be a Variable, like $$AAA or §§BBB or $$001

P2 - increment/decrement operator or just a | .

P3 - Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like)
Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like)
Jump..

Example

```
'************************************
' JNZ - Example
'************************************
VAR.§§TST=-50
:try
VIC.§§TST
' just jump without other features
JNZ.§§TST|try
PRT.Hier
:over
DMP.
MBX.!
END.

'************************************
' Advanced JNZ - Example (using :Label)
' the second one makes a JSR.-Jump.
' Also we packed the Label into a Variable
'************************************
VAR.§§TST=10
: §§LAB=try
:try
DBP.The number is §§TST
JNZ.§§TST<§§LAB
JNZ.§§TST|:over
PRT.Hier
DMP.
MBX.!
END.

:over
DBP.Was here with §§TST
VDC.§§TST
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.

'************************************
' JNZ - Example
' with Post-Increment
'************************************
VAR.§§TST=-50
:try
DBP.Number is: §§TST
' just jump without other features
JNZ.§§TST>try
DBP.Hier
:over
DMP.
MBX.!
END.

'************************************
' JNZ - Example
' with Post-Decrement
'************************************
VAR.§§TST=50
:try
DBP.Number is: §§TST
' just jump without other features
JNZ.§§TST<try
DBP.Hier
:over
DMP.
MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.16 JOR. - Jump On Repeat-Count

MiniRobotLanguage (MRL)

JOR. Conditional Jump

Jump On Repeat count

Intention

This command solves a special need. It automates the "Repeat count" of an repeated action. Imagine you have a special condition in your code.Something was not found or some kind of error-condition.
For example, a file was not found, or a window did not appear.

You decide that you want to try again, checking for the missing thing.
How often do you want to try looking for it?
That is the "Repeat count". Maybe only 3 times or 5 times. Maybe 1000 times. It's up to you.

The idea behind `JOR.` is:

- first make an endless loop (that endless tries whatever)
- then add an `RRC.` immediately before the endless-loop
- finally insert an JOR. into the endless loop and hence give it a exit condition

The general Usage is:

```
' Reset the Repeat Count first
RRC.
   :Label
  ' Try something, jump away if it works
' Leave the endless loop if the number of retries is exceeded.
JOR.(number of retries)|(error-label)
JMP.Label
```

Or you can also use a endless loop instead. This is a real, working example-code:

```
RRC.
  DOL.1
  ' Try something, jump away if it works
  ' Leave the endless loop if the number of retries is
exceeded.
  JOR.5|error
OOP.
ENR.

:error
MBX.An error happened.
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Technical Details:
The Repeat-count is not a global counter. Instead each JOR. has its own Repeat counter and its own RRC. Just one.

JOR. must be used together with its RRC. (Reset Repeat Count).
You can Repeat count a sub-program, that also repeat counts something

```
RRC.
    RRC.
    ...
    JOR.
JOR.
```

For this to work, the RRC. must come first and the JOR. must be in a following line.
In fact RRC. - JOR. is a sort of Loop-construction, that can be nested.

Syntax

JOR.P1|P2[|P3]

Parameter Explanation

P1 – number or variable containing a number which is the maximum repeat count, before the jump is done..

P2 – Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like) Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like) Jump..

P3 – ***(optional)*** must be variable, if given, is set to one in case the Jump was done,
else is set to zero.

Example

```
'************************************
' JOR - Example
'************************************
'DBM.2
' The following RRC. sets the Repeat-Counter to zero
RRC.
  :Labb
  DBP.Repeatin 1
' The JOR. will increase the Repeat Counter
' each time its called.
JOR.3|Laba
JMP.Labb
' This is where we Jump, if the Repeat-Counter
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' becomes bigger or equal to 3
:Laba
RRC.
  :Labc
  DBP.Repeat 2
JOR.5|Labd
JMP.Labc
:Labd
DBP.We are now here.
END.
```

Remarks

The RRC. builds internally a connection with the **next** JOR.
You can not call RRC. for example with JSR. or JNF.

The Preprocessor will only connect the RRC with the next successive JOR.
Therefore the RRC. must be in teh code above the JOR.

Limitations:

The Repeat counter is limited to a maximum of ~2 Billion retries.
The RRC. and JOR. must be in the same file.

See also:

- RRC. - Reset Repeat-Counter 2227
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.17 JRR. - Jump and Reset Return-Stack

MiniRobotLanguage (MRL)

JRR. Conditional Jump

Jump and Reset Returnstack

Intention

Error-handling in Subroutines. Imagine you have called a subroutine using `JSR.` Normally you would return from there using `RET.` And then continue your script. Now there is an error or special condition. You want to jump directly to the error-handler in the main-script. Or you want to just re-run the complete script from start?

If you do just that, there will be more and more useless entries on the so called "Return stack". That is the place, where each `JSR./GSB.` remembers where to jump back in case of a return (`RET.`). Now comes `JRR.`
With `JRR.` you can clear the `RET.`-Stack and make the robot forget all previous `GSB./JSR.` Usage is simple, use:

```
' This will clear the complete Return Stack
JRR.restart
```

You can also choose to remove only the the last **P2**-number of entries from the return stack if a number is given for **P2**.

```
' ... will only remove the last entry
 JRR.restart|1
```

And then you can just re-run the Script or do whatever is needed.

You can use `JRR.` on any of the following cases. Note in the case of `FEX.`-Instructions that you should prefer to use `EFX.` if you want to stop the enumeration.

- STW. - Description 3814 with the "W", or "V"-Option
- ! SCW - Search-Child-Window 3738 with the "W", or "V"-Option
- `ITW.` with the "W", or "V"-Option
- `ICW.` with the "W", or "V"-Option
- JIV. - Jump-If-Variable 2177 or
- any other of the `JIX. - Jumps`, when using the ":"-Label Option
- any of the FEX.-Enumeration Instructions, **important**: use `EFX.` in that case

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

JRR.P1[|P2]

Parameter Explanation

P1 – Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like) Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like) Jump..

P2 – if omitted the completed Return Stack is cleared. If a number is given, the number of entries is been removed from the return stack.

Example

```
'***********************************************
' Example 1 - without Parameters,
'***********************************************
: $$AAA=10
:restart
JSR.Laba
CAL.$$AAA=3+(9*$$AAA)+2.5
PRT. Resultat ist: $$AAA
END.

:Laba
IVV.$$AAA=10
  : $$AAA=9
  JRR.restart
EIF.
RET.

ENR.
```

Remarks

Alternatively you can use `JMP.retry|r`

Limitations:

-

See also:

- RRS. - Reset Return Stack 2230
- RET. - RETurn 2224

© 2023 Theo Gottwald * https://www.smart-package.com

- JSR. - Jump SubRoutine (Label) 2218
- JMP. - Jump to Label 2189
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.18 JRS. - Jump if Result

JRS. - Jump If Result Previous 2212 Top 41 Next 2218

MiniRobotLanguage (MRL)

JRS. Conditional Jump

Jump If Result

Intention

This command can be used to test, if an specified Background- or Parallel-Robot Operation has emitted an result or not. Usage is simple:

```
SMB.
PAU.10
JRS.|result
MBX.Error we got no Result
ENR.
:result
MBX.Result is available
ENR.
```

You can use a variable that contains a label (dynamic jumping).
Here is another example:

```
SMB.
$$TIN=$$000
PAU.4
JNR.$$TIN|noresult
MBX.Yes we got an Result
ENR.
:noresult
MBX.No Result was available
ENR.
```

The command can be switched from default mode (**Jump Mode**) into "**Gosub Mode**" if you prefix the Label-Parameter **P3** with a **colon**. Here is an example:

```
SMB.
:Retry
JNR.|:test
PAU.0.25
JNR.|Retry
ENR.
'--------------
:test
DBP. No Result yer!
PAU.0.1
RET.
```

`JNR.` is the negative version of `JRS.`

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

JRS.[P1]|P2
JRS.[P1]|:P2

Parameter Explanation

P1 **-** ***(optional)*** Variable that evaluates to a valid Ticket-Number, that is connected with a Background Operation or a Parallel Robot. If omitted the last emitted Ticket-Number is used.

P2 **-** Label or Variable that evaluates to a Label.
If it starts with a ":" (colon) a JSR. - Jump SubRoutine (Label) 2218 (GOSUB like) Jump is done instead of the normal JMP. - Jump to Label 2189 (GOTO like) Jump..

Example

```
'************************************
' JRS - Example
'************************************
SMB.
:Retry
PAU.0.25
JRS.|out
JMP.Retry
ENR.
'--------------
:out
DBP. Result Ok!
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Parallel Robot Operations 2632
- JNR. - Jump if no Result 2197
- GRS. - Get Result 430
- DRS. - Drop Result 427
- WRS. - Wait for Result 436
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.19 JSR./GSB. - Jump SubRoutine (Label)

JSR. - Jump SubRoutine (Label) Previous 2215 Top 41 Next 2224

MiniRobotLanguage (MRL)

JSR./ GSB. Unconditional Subroutine-Jump

Jump SubRoutine / GoSub

Intention

This is a Subroutine call (`GOSUB` - like). `JSR./GSB.` invoke a subroutine at the given Label, keep the Return-line on the Return Stack. It will returns to the statement following the `JSR./GSB.` after reaching a `RET.` - Command.

What is a subroutine?
A subroutine is a Sub-program in your Main-program. Mostly subroutines encapsule an specific part of the overall job of your script.

Note that it has **two names: GSB. and JSR.**.
These two are completely equal.

Usage is easy:

```
GSB.Label
' This is where we continue after the return
ENR.

:Label
' Do something here
RET.
```

Can i use Subroutine parameters?
Yes, the variables **$$_01** ... **$$_25** are hold the Subroutine Parameters.
The number of given parameters is available in **$$_00**.

If you specify just Variables as Parameter, the Content is only copied,
and the Type of the Variable is preserved.

```
 ' Example
 VAN.$$NUM=5.09
 GSB.AddMe|$$NUM|$$NUX
 ENR.
 :AddMe
VAO.$$PAA=$$_01
GVT.$$PAA|$$RES
MBX.We preserved the Variabletype: $$RES (=5, numeric)
RET.
```

Can i just jump out of a Subroutine?

*© 2023 Theo Gottwald * https://www.smart-package.com*

If you plan to end the script anyway, then **yes**.
If you plan to invoke other subroutines later, then better **not**.
In this case use the:

- JRR. - Jump and Reset Return-Stack 2212
- RRS. - Reset Return Stack 2230

commands to clean up the Return-stack before/while jumping out.

Speed in Ticks:

This command uses typically between 30 to 60 Ticks.

You can also call Subroutines in the same way like with `JSR./GSB.` with the following commands:

- STW. - Description 3814 with the "W", or "V"-Option
- ! SCW - Search-Child-Window 3738 with the "W", or "V"-Option
- `ITW.` with the "W", or "V"-Option
- `ICW.` with the "W", or "V"-Option
- JIV. - Jump-If-Variable 2177 or
- any other of the `JIX. - Jumps`, when using the ":"-Label Option

Syntax

JSR.P1[|P2..|Px]

Parameter Explanation

P1 – `TXT (Label) or Variable containing the Label.`
P2 ... Px -(optional) `Parameters for subroutine`

Example

```
'*********************************************
' Example 1 - without Parameters,
'*********************************************

DBM.2
JSR.Laba
CAL.$$AAA=3+(9*$$AAA)+2.5
PRT. Resultat ist: $$AAA
END.

:Laba
: $$AAA=10
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.

'*********************************************
' Example 2 - with Parameter
'*********************************************

DBM.2
: $$LAA=Laba
JSR.$$LAA|10
CAL.$$AAA=3+(9*$$AAA)+2.5
PRT. Resultat ist: $$AAA
END.

:Laba
: $$AAA=§§_01
PRT. $$AAA
RET.


'*********************************************
' Example 3 - with Parameter, uses Macros, GSB.
' and also returns Result.
'*********************************************
' Name; First Name; Street; ZIP-Code; ID-Number
'
$$PAT=?exeloc\Test.txt
$$DEL=,
$$FNA=James,Mary,John,Patricia,Robert,Jennifer,Michael,William,Elizabeth,David,B
$$FNA=$$FNA,Richard,Susan,Joseph,Jessica,Thomas,Sarah,Charles,Karen,Christopher,

$$LNA=JOHNSON,WILLIAMS,JONES,BROWN,DAVIS,MILLER,WILSON,MOORE,TAYLOR
$$LNA=$$LNA,ANDERSON,THOMAS,JACKSON,WHITE,HARRIS,MARTIN,THOMPSON
$$LNA=$$LNA,GARCIA,MARTINEZ
VAR.$$TIM=#time#
$$FIN=
FOR.$$LOP|1|255
  GSB.Get_Lastname
  GSB.Get_Firstname
  POP.$$TX1
  POP.$$TX2
  GSB.Get_Streetname
  POP.$$TX3
  GSB.Get_ZipCode
  POP.$$TX4
  $$IDN=$$LOP
  FMT.$$IDN|0000
  $$LIN=$$TX1$$DEL$$TX2$$DEL$$TX3$$DEL$$TX4$$DEL$$IDN
  $$FIN=$$LIN$crlf$$$FIN
NEX.
DBP.$$FIN
CTF.$$PAT|$$FIN
CAL.$$TIM=#time#-$$TIM|i
MBX. Operation took: $$TIM seconds.
ENR.
'============================================================
:Get_Firstname
VAO.$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
%Get_Random_Element $$FNA|$$RES
PUV.$$RES
RET.
'-------------------------------------------------------------
:Get_Lastname
VAO.$$RES
%Get_Random_Element $$LNA|$$RES
PUV.$$RES
RET.
'-------------------------------------------------------------
' P1 - List to use
' P2 - Variable for Result
: %Get_Random_Element 2
GSB.get_random_elem|§§§01|§§§02
END%

:get_random_elem
SAV.Save|$$OUT|$$LST|$$MIN|$$MAX|$$LAN
VAO.$$OUT=§§_02
VAR.$$LST=§§_01
%Get_Listcount $$LST
POP.$$MAX
$$MIN=1
RND.$$MIN|$$MAX|$$RES
SBD.$$LST|$$DEL|$$RES|$$LAN
VTM.$$LAN
VAI.$$OUT=$$LAN
SAV.Restore
RET.
'-------------------------------------------------------------
:Get_Streetname
SAV.Save|$$TXA|$$TXB|$$TXC
GSB.Get_Lastname
GSB.Get_Firstname
POP.$$TXA
POP.$$TXB
RND.1|99|$$NUM
$$TXC=$$TXA-$$TXB-Street. $$NUM
PUV.$$TXC
SAV.Restore
RET.
'-------------------------------------------------------------
:Get_ZipCode
SAV.Save|$$RES
RND.30000|92000|$$RES
PUV.$$RES
SAV.Restore
RET.
'-------------------------------------------------------------
: %Get_Listcount 1
GSB.Get_Listcount|§§§01
END%

:Get_Listcount
SAV.Save|$$CNT|$$PAR
$$PAR=§§_01
GSC.$$PAR|$$DEL|$$CNT|d
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PUV.$$CNT
SAV.Restore
RET.
'------------------------------------------------------------
```

Remarks

The variables §§_01 ... §§_25 are used for Parameters.
The number of given parameters is available in §§_00.

JSR. - unlike JNF. - does NOT open a local Variables/local Stack-Space for local variables.

Therefore Variables in a Subprogram called with JSR. are currently not explicitly local.

You can also call Subroutines in the same way like with JSR./GSB. with these commands:

- STW. - Description 3814 with the "W", or "V"-Option
- ! SCW - Search-Child-Window 3738 with the "W", or "V"-Option
- ITW. with the "W", or "V"-Option
- ICW. with the "W", or "V"-Option
- JIV. - Jump-If-Variable 2177 or
- any other of the JIX. - Jumps, when using the ":"-Label Option
- any of the FEX.-Enumeration Instructions

Limitations:

There is no depth limit for Subroutine calls. recursive calls are allowed.

See also:

- RET. - RETurn 2224
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JIN. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JNF. - Jump New File 2191
- RTS. - ReTurn from Subprogram 2233

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.20 RET. - RETurn

MiniRobotLanguage (MRL)

RET. Statement

RETurn from Gosub

Intention

Return from a Subroutine that was called using any of the statements below.

Returns to the line/statement following the Subroutine call.

A subroutine can be called with any of the following Instructions / Instruction-groups:

- JSR. - Jump SubRoutine (Label) 2218 (Same as `GSB.`)
- STW. - Description 3814 with the "W", or "V"-Option
- ! SCW - Search-Child-Window 3738 with the "W", or "V"-Option
- `ITW.` with the "W", or "V"-Option
- `ICW.` with the "W", or "V"-Option
- JIV. - Jump-If-Variable 2177 or
- any other of the `JIX. - Jumps`, when using the ":"-Label Option
- any of the FEX.-Enumeration Instructions

Syntax

RET.[P1]

Parameter Explanation

P1 - (optional) how many level you want to jump up. If omitted, the RET. Instruction will bring you just up one level. If you want to get to the Top-Level, for example in case of errors, you can write:

```
RET.all
```

this will bring you in one strike to the Top-Level of the Script. Note that all levels below have been removed from the Subprogram-Return Stack.

Speed in Ticks:

This command uses typically between 10 to 20 Ticks.

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'*********************************************
' Example 1 - without Parameters,
'*********************************************

[First.txt]
DBM.2
JSR.Laba
CAL.$$AAA=3+(9*$$AAA)+2.5
PRT. Resultat ist: $$AAA
END.

:Laba
: $$AAA=10
RET.

'*********************************************
' Example 2 - with Parameter
'*********************************************

DBM.2
: $$LAA=Laba
JSR.$$LAA|10
CAL.$$AAA=3+(9*$$AAA)+2.5
PRT. Resultat ist: $$AAA
END\

:Laba
: $$AAA=§§_01
PRT. $$AAA
RET.
```

Remarks

Please do not intermix `RET.` with `RTS`, because `RTS` is used to Return from Subprograms which are called with `JNF`.

Limitations:

There are no relevant depth limitations or limitations about recursive functions.

See also:

- JSR. - Jump SubRoutine (Label) 2218
- RTS. - ReTurn from Subprogram 2233
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JNZ. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209

© 2023 Theo Gottwald * https://www.smart-package.com

- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JNF. - Jump New File 2191

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.21 RRC. - Reset Repeat-Counter

MiniRobotLanguage (MRL)

RRC. Command

Reset Repeat Counter

Intention

`RRC.` will zero the Repeat-Count of the following/next JOR. - Jump On Repeat-Count 2209 in the code.

This command solves a special need. It automates the "Repeat count" of an repeated action. Imagine you have a special condition in your code.Something was not found or some kind of error-condition.
For example, a file was not found, or a window did not appear.

You decide that you want to try again, checking for the missing thing.
How often do you want to try looking for it?
That is the "Repeat count". Maybe only 3 times or 5 times. Maybe 1000 times. It's up to you.

The idea behind `RRC./JOR.` is:

- first make an endless loop (that endless tries whatever)
- then add an `RRC.` immediately before the endless-loop

finally insert an JOR. into the endless loop and hence give it a exit condition

The general Usage is:

```
' Reset the Repeat Count first
RRC.
   :Label
  ' Try something, jump away if it works
' Leave the endless loop if the number of retries is exceeded.
JOR.(number of retries)|(error-label)
JMP.Label
```

Or you can also use a endless loop instead. This is a real, working example-code:

```
RRC.
  DOL.1
  ' Try something, jump away if it works
  ' Leave the endless loop if the number of retries is
exceeded.
  JOR.5|error
OOP.
ENR.

:error
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
MBX.An error happened.
END.
```

Technical Details:
The Repeat-count is not a global counter. Instead each JOR. has its own Repeat counter and its own RRC. Just one.

JOR. must be used together with its RRC. (Reset Repeat Count).
You can Repeat count a sub-program, that also repeat counts something

```
RRC.
    RRC.
    ...
    JOR.
JOR.
```

For this to work, the RRC. must come first and the JOR. must be in a following line.
In fact RRC. - JOR. is a sort of Loop-construction, that can be nested.

Syntax

RRC .

Parameter Explanation

```
No parameters are required with this command.
```

Example

```
'************************************
' JOR - Example
'************************************
RRC.
  DOL.1
  ' Try something, jump away if it works
  JOR.5|error
OOP.

ENR.

:error
MBX.An error happened.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The RRC. builds internally a connection with the **next** JOR.
You can not call RRC. for example with JSR. or JNF.

The Preprocessor will only connect the RRC with the next successive JOR.
Therefore the RRC. must be in the code above the JOR.

Limitations:

The Repeat counter is limited to a maximum of ~2 Billion retries.

See also:

- JOR. - Jump On Repeat-Count 2209
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JNZ. - Jump if Not zero 2206
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183
- JMP. - Jump to Label 2189
- JSR. - Jump SubRoutine (Label) 2218
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.22 RRS. - Reset Return Stack

MiniRobotLanguage (MRL)

RRS. Command

Reset Return Stack

Intention

Error-handling in Subroutines, called with `GSB./JSR.` or equivalent Jump-Subroutine-Instructions, like:

- STW. - Description 3814 with the "W", or "V"-Option
- ! SCW - Search-Child-Window 3738 with the "W", or "V"-Option
- `ITW.` with the "W", or "V"-Option
- `ICW.` with the "W", or "V"-Option
- JIV. - Jump-If-Variable 2177 or
- any other of the `JIX. - Jumps`, when using the ":"-Label Option

Imagine you have called a subroutine using `JSR.`
Normally you would return from there using `RET.` And then continue your script.
Now there is an error or special condition. You want to jump directly to the error-handler in the main-script. Or you want to just re-run the complete script from start?

If you do just that, there will be more and more useless entries on the so called "Return stack". That is the place, where each `JSR./GSB.` remembers where to jump back in case of a return (`RET.`). Now comes `RRS. and JRR.`

With `RRS.` you can clear the `RET.`-Stack and make the robot forget all previous `GSB./JSR.` Usage is simple, use:

```
' This will clear the complete Return Stack
RRS.
```

You can also choose to remove only the the last **P1**-number of entries from the return stack if a number is given for **P1**.

```
' ... will only remove the last entry
 RRS.1
```

Then you can just re-run the Script or do whatever is needed.

You can use `RRS.` on any of the following cases. Note in the case of `FEX.`-Instructions that you should prefer to use `EFX.` if you want to stop the enumeration.

- STW. - Description 3814 with the "W", or "V"-Option
- ! SCW - Search-Child-Window 3738 with the "W", or "V"-Option
- `ITW.` with the "W", or "V"-Option

*© 2023 Theo Gottwald * https://www.smart-package.com*

- ICW. with the "W", or "V"-Option
- JIV. - Jump-If-Variable 2177 or
- any other of the JIX. - Jumps, when using the ":"-Label Option
- any of the FEX.-Enumeration Instructions, **important**: use EFX. in that case

Syntax

RRS.[P1]

Parameter Explanation

P1 **-** number of Entries to be removed from the Return Stack.

Example

```
'************************************
' RRS. Example
'************************************
DBM.2
$$LOP=1

:retry
DOL.$$LOP=1

JSR.suba
OOP.


MBX.?
END.

:suba
IVV.$$ERR=1
 RRS.
 JMP.retry
ELS.
 JSR.sub
EVV.
' return normal way
RET.

:sub
' Do something
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Alternatively you can use `JMP.retry|r`

Limitations:

-

See also:

- JRR. - Jump and Reset Return-Stack 2212
- JSR. - Jump SubRoutine (Label) 2218
- RET. - RETurn 2224
- JMP. - Jump to Label 2189
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JNZ. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183

© 2023 Theo Gottwald * https://www.smart-package.com

3.60.23 RTS. - ReTurn from Subprogram

MiniRobotLanguage (MRL)

RTS. Statement

ReTurn from Subprogram to calling script

Intention

Used together with JNF. - Jump New File 2191 to call Subprograms.
`RTS.` will return program flow to the calling Script.

If you need to leave a Sub-Program that was called with `JNF.` to a higher level then just the next, you can add a number behind the `RET.` Like this:

```
RTS.2
```

If you want to leave to the Top-Level in your script, just write:

```
RTS.all
```

Syntax

RTS.[P1]

Parameter Explanation

P1 **-** `Numeric, a variable or the term "all"`

Example

```
We need to have 2 Script Files for this simple Example. Both
need to be in the same folder. The first Scriptfile,
"First.txt"  that is run defines ?path, which is the folder
where it is. it calls the second file "second.txt".

[First.txt]
: $$WHE=In First.txt
: $$GLO=1
PRT. $$WHE, Global=$$GLO
JNF.Second.txt
PRT. $$WHE, Global=$$GLO
END.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
[Second.txt]
: §§WHE=In Second.txt
: $$GLO=2
PRT. $$WHE, Global=$$GLO
RTS.
```

Remarks

Please do not intermix `RTS,.` with `RET.`
`RET.` is used to Return from Subroutines which are called with `JSR./GSB.`

Limitations:

There are now relevant limitations about recursion depth.

See also:

- JNF. - Jump New File 2191
- JMP. - Jump to Label 2189
- RET. - RETurn 2224
- JIV. - Jump-If-Variable 2177
- JIZ. - Jump-If-Zero 2180
- JNZ. - Jump if Not zero 2206
- JOR. - Jump On Repeat-Count 2209
- JME. - Jump on More or Equal 2186
- JLE. - Jump on Less or Equal 2183

© 2023 Theo Gottwald * https://www.smart-package.com

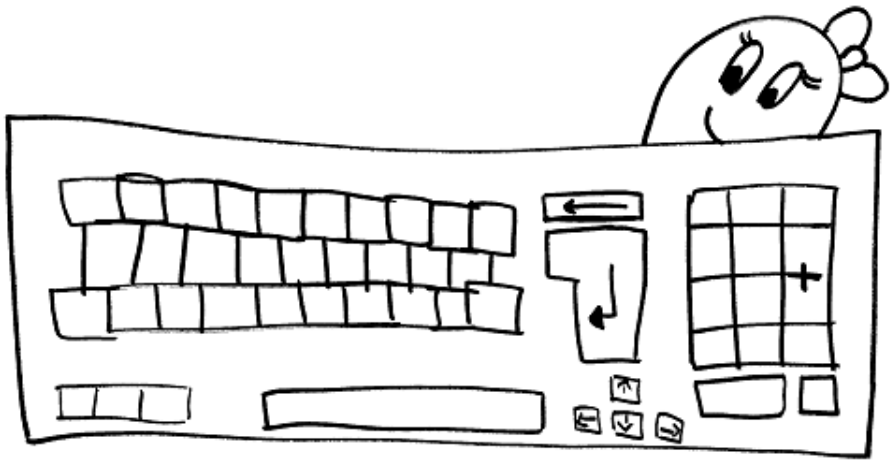
3.61 Keyboard Typing Simulation

3.61.1 ! Keyboard-Keycode-List

MiniRobotLanguage (MRL)

Keycode-List

Key Code List



Intention

These Keycodes are available with `KYD.`, `KYU.`, `SKP.` and other Compatible Engine 231 Keyboard-Typing Commands.

This is the most compatible and "Low Level" Keyboard simulation we have. Using these commands, you can even make window to bring up the Task-Manager, by simulating a Key-press:

```
' Bring Windows Task-Manager up
KYD.{STRG},{SHIFT},{ESC}
PAU.0.100
KYU.{STRG},{SHIFT},{ESC}
ENR.

' This example will open a Menu in Photoimpact
' using the SHFT-CTRL-S Key-Combination
'
STW.ct|IE3_CLASS|PhotoImpact -
SCW.nc|1|Afx:400000:0:10013:0:0
KYD.{SHIFT},{CTRL}
SKP.s
KYU.{SHIFT},{CTRL}
PAU.0.2
RET.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Keycode-List

Low-Level-Keycodes:

Special-Sequence	Keycode	Simulates:
{LMB}	1	Left MouseButton
{RMB}	2	Right MouseButton
{CANCEL}	3	Click on CANCEL (see 30)
{MMB}	4	Middle Mouse Button
{MSX1}	5	Mouse-X1 Key
{MSX2}	6	Mouse-X2 Key
{BS},{BACKSPACE}	8	BACKSPACE (deletes to left side)
{TAB}	9	TAB
{DEL},{CLEAR}	12	Delete-Key (deletes to right side), see also 46
{RETURN}	13	RETURN-Key
{SHIFT}	16	SHIFT-Key (general), for L/R see 160,161
{CTRL},{STRG}	17	CTRL-Key (general), for L/R see 162,163
{ALT},{MENU}	18	ALT-/MENU-Key (general), for L/R see 164 and 165
{PAUSE}	19	PAUSE-Key
{CAPSLOCK}	20	CAPS-LOCK - Key
{KANA}	21	KANA-Key,HANGUL-Key
{JUNJA}	23	JUNJA-Key,
{FINAL}	24	FINAL-Key
{KANJI}	25	KANJI-Key, HANJA-Key
{ESC}	27	ESCAPE-Key
{CONVERT}	28	CONVERT-Key
{NOCONVERT}	29	NOCONVERT-Key
{ACCEPT}	30	ACCEPT-Button (see 3)
{MODECHANGE}	31	MODECHANGE-Key
{SPACE}	32	SPACE-key
{PAGEUP},{PGUP]	33	PAGE-UP Key
{PAGEDOWN},{PGDN}	34	PAGE-DOWN
{END}	35	END-Key
{POS1},{HOME}	36	HOME/POS1-Key
{LEFT}	37	Left Arrow-Key
{UP}	38	Up Arrow-Key
{RIGHT}	39	Right Arrow-Key
{DOWN}	40	Down Arrow-Key
{SELECT}	41	
{PRINT}	41	
{EXECUTE}	43	
{SNAPSHOT},{SNAP}	44	Print Screen-Key
{INSERT}	45	INSERT-Key
{DELETE}	46	DELETE-Key
{HELP}	47	

© 2023 Theo Gottwald * https://www.smart-package.com

```
  -                 -  48-57 Numbers 0-9

{KP_01}             -  48  - Keypad-Modifier, together with
                             SHIFT (160) will be
                             Specialcharacter "*" on the
Keypad
{KP_02}             -  56  - Keypad-Modifier, together with
                             SHIFT (160) will be
                             Specialcharacter "(" on the
Keypad
{KP_03}             -  57  - Keypad-Modifier, together with
                             SHIFT (160) will be
                             Specialcharacter ")" on the
Keypad
  -                 -  65-90 Alphabet letter a-z
{LWIN}              -  91  - Left Windows-Button
{RWIN}              -  92  - Right Windows Button
{APPS}              -  93  - APPS-Key, often left to the right
                             CTRL-Key. Opens Menus.
{SLEEP}             -  95
{NUM_0}             -  96  - Numerical Key-Pad Number 0
{NUM_1}             -  97  - Numerical Key-Pad Number 1
{NUM_2}             -  98  - Numerical Key-Pad Number 2
{NUM_3}             -  99  - Numerical Key-Pad Number 3
{NUM_4}             -  100 - Numerical Key-Pad Number 4
{NUM_5}             -  101 - Numerical Key-Pad Number 5
{NUM_6}             -  102 - Numerical Key-Pad Number 6
{NUM_7}             -  103 - Numerical Key-Pad Number 7
{NUM_8}             -  104 - Numerical Key-Pad Number 8
{NUM_9}             -  105 - Numerical Key-Pad Number 9
{NUM_MUL}           -  106 - Numerical Key-Pad Multiplication
{NUM_ADD}           -  107 - Numerical Key-Pad Addition
{NUM_SEP}           -  108 - Numerical Key-Pad Separator Key
{NUM_SUB}           -  109 - Numerical Key-Pad Subtraction
{NUM_DOT}           -  110 - Numerical Key-Pad Decimal Point
{NUM_DIV}           -  111 - Numerical Key-Pad Divide Key
{F1}                -  112 - Function Key F1
{F2}                -  113 - Function Key F2
{F3}                -  114 - Function Key F3
{F4}                -  115 - Function Key F4
{F5}                -  116 - Function Key F5
{F6}                -  117 - Function Key F6
{F7}                -  118 - Function Key F7
{F8}                -  119 - Function Key F8
{F9}                -  120 - Function Key F9
{F10}               -  121 - Function Key F10
{F11}               -  122 - Function Key F11
{F12}               -  123 - Function Key F12
{F13}               -  124 - Function Key F13
{F14}               -  125 - Function Key F14
{F15}               -  126 - Function Key F15
{F16}               -  127 - Function Key F16
{F17}               -  128 - Function Key F17
{F18}               -  129 - Function Key F18
{F19}               -  130 - Function Key F19
{F20}               -  131 - Function Key F20
{F21}               -  132 - Function Key F21
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
{F22}               -  133 - Function Key F22
{F23}               -  134 - Function Key F23
{F24}               -  135 - Function Key F24
{NUM},{NUMLOCK}     -  144 - NUMLOCK-Key
{SCROLLOCK}         -  145 - SCROLLOCK-Key
{LSHIFT}            -  160 - Left SHIFT Key
{RSHIFT}            -  161 - Right SHIFT Key
{LCTRL},{LSTRG}     -  162 - Left CTRL-Key
{RCTRL},{RSTRG}     -  163 - Right CTRL-Key
{LALT},{LMENU}      -  164 - Left ALT-Key
{RALT},{RMENU}      -  165 - Right ALT-Key, ALT Gr.
```

Hint: To get **`ALT Gr`**, you need to use the two Keys: `165 and 162`. See example below.

```
{BROWSEBACK},{BBACK}     -  166
{BROWSERFWD},{BFWD}      -  167
{REFRESH},{BREFRESH}     -  168
{BROWSERSTOP},{BSTOP}    -  169
{BROWSERSEARCH},{BSEARCH}-  170
{BROWSERFAVS},{BFAVS}    -  171
{BROWSERHOME},{BHOME}    -  172
{MUTE}                   -  173
{VOLDN},{VOLUMEDOWN}     -  174
{VOLAUP},{VOLUMEUP}      -  175
{NTRACK},{NEXTTRACK}     -  176
{PTRACK},{PREVTRACK}     -  177
{MEDIASTOP},{MSTOP}      -  178
{MPAUSE},{PLAYPAUSE}     -  179
{MAIL},{LAUCHNMAIL}      -  180
{MSELECT},{MEDIASELECT}  -  181
{LAUCHAPP1}              -  182 - Launch Application 1 (may be
                                  Windows-Explorer)
{LAUCHAPP2}              -  183 - Launch Application 2 (may be
                                  Windows-Calculator)
{OEM_1},{UE},{UML_UE}    -  186- German "Ü"-Umlaut
{OEMPlus}                -  187
{OEMComma}     ","       -  188
{OEMMinus}               -  189
{OEMPeriod}              -  190
{OEM_2} "#"              -  191
{OEM_3},{OE},{UML_OE}    -  192 - German "Ö"-Umlaut
{OEM_4}                  -  193
{OEM_5}                  -  194
{OEM_6}                  -  195
{OEM_7}                  -  196
{SS},{GER_SS}            -  219 - German "ß"-Umlaut
{HIGH}         "^"       -  220
{AE},{UML_AE}            -  222- German "Ä"-Umlaut
{PIPE}         "|"       -  226 - the "<"-Key, below the "a",
                                  only on german keyboards.
{PROCESS}                -  229
{ATTN}                   -  246
{CRSEL}                  -  247
{EXSEL}                  -  248
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
{EREOF}                 -  249
{PLAY}                  -  250
{ZOOM}                  -  251
{PA1}                   -  253
{OEMCLEAR}              -  254
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.61.2 GEK. - Get Extended Key-State

MiniRobotLanguage (MRL)

GEK. Command

Get Extended Key-State

Intention

The GEK.-Command has two modes of operation.

In **Mode 1**, it can be used to get the state of NUMLOCK, CAPSLOCK and SCROLLLOCK.
What you get is really the "**state of the keyboard**". Not "the state of the button". GEK.-returns the "real thing", not just the key-states that you will get with GKS. The keystates from GES. just show you, if actually the NUMLOCK-Key is up or down. GEK. will really get the state of the keyboard about NUMLOCK, CAPSLOCK, SCROLLLOCK etc. When choosing between GKS. and GEK. you need to understand the difference between these commands.

In **Mode 2**, it can be used to retrieve the state of the SHIFT, CTRL or the ALT-Key and others, see list below. In this case you really check whether the given Key is down or not.

Note: Even if CAPSLOCK is activated, you will get a "0" if you check for SHIFT, because that key is actually not down itself.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

GEK.P1[|P2]

Parameter Explanation

P1 - VAR or numeric

P2 - ***(optional)*** Variable. If omitted TOS 104 is used for result.

```
Parameter-List for P1:

SHIFT          - any shift key (keyboard Shift-state)
LSHIFT         - only left Shift-Key
RSHIFT         - only right Shift-Key
STRG,CTRL      - any Ctrl key (keyboard Ctrl-state)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
LSTRG,LCTRL   - only left Ctrl-Key
RSTRG,RCTRL   - only right Ctrl-Key
MENU,ALT      - any ALT key (keyboard ALT-state)
LMENU,LALT    - only left ALT-Key
RMENU,RALT    - only right ALT-Key
NUMLOCK,NUM   - keyboard NUMLOCK-State
CAPSLOCK,CAPS - keyboard CAPSLOCK-State
SCROLLOCK,SCROL,SCROLL - keyboard SCROLLLOCK-State
```

Example

```
'*****************************************************
' EXAMPLE 1: GEK.
'*****************************************************
' Please run this script only in the editor
' Then press the SHIFT-Key (left or right) while its running.
DOL.1
  GEK.SHIFT|$$OUT
  DBP.$$OUT
OOP.

'*****************************************************
' EXAMPLE 2: GEK.
'*****************************************************
' Please run this script only in the editor
' Then press the NUMLOCK-Key (left or right) while its
running.
DOL.1
  GEK.NUM|$$OUT
  DBP.$$OUT
OOP.

'*****************************************************
' EXAMPLE 3: GEK.
'*****************************************************
' Please run this script only in the editor
' Then press the SHIFT-Key (left or right) while its running.
DOL.1
  GEK.CAPSLOCK
  DBP.$$000
OOP.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- SEK. - Set Extended Key-State 2240
- GKS. - Get Key-State 2243
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.3 GKS. - Get Key-State

MiniRobotLanguage (MRL)

GKS. Command

Get Key-State

Intention

The GKS.-Command can be used to find out if a specified key on the keyboard is actually pressed or released. Unlike GEK., the GKS.-Command will return the state of the physical keyboard key, and tell you if its up or down (0/1).

GKS. has three modes of operation. In the first Mode, you can specify the button using all of the Low-Level Keycodes see here:
Keyboard-Keycode-List 2235.

In the second Mode, you can type a **#** followed by a decimal number or variable that contains the number. Like this:

```
' This will check the state of the left SHIFT-Key
GKS.#160|$$RES
DBP.$$RES

' This will get the state of the left SHIFT-Key
' You can omit the {} here if you want.
GKS.SHIFT|$$RES
DBP.$$RES

' This will get the state of the left SHIFT-Key
GKS.{SHIFT}|$$RES
DBP.$$RES
```

In the third Mode, you define the key directly, using **"a"** or **"B"**. Here is an example:

```
' This will get the state of the a-Key
GKS."a"|$$RES
DBP.$$RES
```

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

GKS.P1[|P2]

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 **-** VAR or numeric, or Keycode see below.

P2 **-** ***(optional)*** Variable. If omitted TOS[104] is used for result.

For **P1**, you can use all Low-Level-Keycodes: see here:

Keyboard-Keycode-List[2235]

Example

```
'*****************************************************
' EXAMPLE 1: GKS.
'*****************************************************
' Run this example from the editor and
' press the "a" button while its running
'
DOL.1
  GKS."A"|$$RES
  DBP.$$RES
  JNZ.$$RES|out
OOP.
:out
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control[194]
- ! Keyboard-Keycode-List[2235]
- GEK. - Get Extended Key-State[2240]
- SEK. - Set Extended Key-State[2240]
- SKP. Send-Key-Press[2289]
- SKT. - Send-Key-Text[2293]
- KYD. - Key Down[2256]
- KYU. - Key Up[2260]
- RST. - Replace Selected Text[2264]
- SDT. - SenD-window-Text[2278]
-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.4 IEK. / NEK. - If Extended Key-State

MiniRobotLanguage (MRL)

IEK. Conditional Statement

If Extended Key-State

Intention

The IEK.-Command has two modes of operation.

In **Mode 1**, it can be used to react on the state of NUMLOCK, CAPSLOCK and SCROLLLOCK. What you get is really the "**state of the keyboard**". Not "the state of the button". IEK.-returns the "real thing", not just the key-states that you will get with IKS.

The keystates from IES. just show you, if actually the NUMLOCK-Key is up or down. IEK. will really get the state of the keyboard about NUMLOCK, CAPSLOCK, SCROLLLOCK etc. When choosing between IKS. and IEK. you need to understand the difference between these commands.

In **Mode 2**, it can be used to react on the state of the SHIFT, CTRL or the ALT-Key and others, see list below. In this case you really check whether the given Key is down or not.

Note: If CAPSLOCK is activated and if you check for SHIFT, then IEK. will evaluate "false", because that key is actually not down itself.

NEK. is the negative form of the command.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

IEK.P1 … ELS. … EIF.

Parameter Explanation

P1 **-** VAR or numeric

Parameter-List for **P1**:

```
SHIFT          - any shift key (keyboard Shift-state)
LSHIFT         - only left Shift-Key
RSHIFT         - only right Shift-Key
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STRG,CTRL     - any Ctrl key (keyboard Ctrl-state)
LSTRG,LCTRL   - only left Ctrl-Key
RSTRG,RCTRL   - only right Ctrl-Key
MENU,ALT      - any ALT key (keyboard ALT-state)
LMENU,LALT    - only left ALT-Key
RMENU,RALT    - only right ALT-Key
NUMLOCK,NUM   - keyboard NUMLOCK-State
CAPSLOCK,CAPS - keyboard CAPSLOCK-State
SCROLLOCK,SCROL,SCROLL - keyboard SCROLLLOCK-State
```

Example

```
'*****************************************************
' EXAMPLE 1: IEK.
'*****************************************************
' Please run this script only in the editor
' Then press the SHIFT-Key (left or right) while its running.
DOL.1
  IEK.SHIFT
     DBP.SHIFT is pressed!
  EIF.
OOP.

'*****************************************************
' EXAMPLE 2: NEK.
'*****************************************************
' Please run this script only in the editor
' Then press the NUMLOCK-Key (left or right) while its
running.
DOL.1
  NEK.NUM
     DBP.NUMLOCK is NOT pressed!
  EIF.
OOP.

'*****************************************************
' EXAMPLE 3: IEK.
'*****************************************************
' Please run this script only in the editor
' Then press the SHIFT-Key (left or right) while its running.
DOL.1
  IEK.CAPSLOCK
     DBP.CAPSLOCK is down!
  EIF.
OOP.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- GKS. - Get Key-State 2243
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.5 IKS. / NKS. - If Key-State

MiniRobotLanguage (MRL)

IKS./NKS. Conditional Statement

If Key-State

Intention

The `IKS.`-Command can be used to find out if a specified key on the keyboard is actually pressed or released. Unlike `IEK.`, the `IKS.`-Command will evaluate the state of the physical keyboard key, and tell you if its up or down and react on that.

`IKS.` has three modes of operation. In the first Mode, you can specify the button using all of the Low-Level Keycodes see here:
Keyboard-Keycode-List 2235.

In the second Mode, you can type a **#** followed by a decimal number or variable that contains the number. Like this:

```
' This will check the state of the left SHIFT-Key
IKS.#160
  DBP.$$RES
EIF.


' This will get the state of the left SHIFT-Key
' You can omit the {} here if you want.
IKS.SHIFT|
  DBP.$$RES
EIF.

' This will get the state of the left SHIFT-Key
IKS.{SHIFT}
  DBP.$$RES
EIF.
```

In the third Mode, you define the key directly, using **"a"** or **"B"**. Here is an example:

```
' This will get the state of the a-Key
IKS."a"
  DBP.$$RES
EIF.
```

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

IKS.P1[|P2]

Parameter Explanation

P1 - VAR or numeric, or Keycode see below.

For **P1**, you can use all Low-Level-Keycodes: see here:
Keyboard-Keycode-List 2235

Example

```
'*****************************************************
' EXAMPLE 1: GKS.
'*****************************************************
' Run this example from the editor and
' press the "a" button while its running
'
DOL.1
  IKS."A"
     GTO.out
  EIF.
OOP.
:out
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- GEK. - Get Extended Key-State 2240
- SEK. - Set Extended Key-State 2240
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264

© 2023 Theo Gottwald * https://www.smart-package.com

- SDT. - SenD-window-Text 2278
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.6 KBC. / KBI. - KeyBoard Command

MiniRobotLanguage (MRL)

KBC./KBI. Command

Keyboard Commands

Intention

The KBC.-Command can be used to simulate Keyboard typing. The Command has multiple predefined options and makes automated typing easier for you. Using KBC. you - for example - can mark all text in an Textbox. Or you can delete all Text in a Textbox.
It generally uses the compatible Engine.

The KBI.-Command works exactly as the KBC.-Command but uses another Typing-Technology that is ***faster***, yet sometimes not as compatible.
KBI. will put things directly into the Keyboard-Input Buffer so typing is faster

So possibly try using the KBI. - because its faster - and if it does not work just use KBC.

There are currently these options:

```
' This option will mark all text in the currently localized
text-control or web-control
' will work in Firefox or internet explorer
KBC.mark all

' This option will mark and delete all text in the currently
localized text-control or web-control
' will work in Firefox or internet explorer
KBC.del all

' This option will read all text in the currently localized
text-control or web-control
' will then compare to the text given as P2 (here $$TXT).
' If they do NOT match then it will delete this text and type
the text from P2
$$TXT=Password
KBC.Type If|$$TXT

The following options simulate the Keyboard:
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  KBC.Copy     -> -> strg-c
  KBC.Copy All -> strg-a strg-c
  KBC.Paste    -> strg-v[|n]   n - is a number that will specify how often to
press strg-v
  KBC.Paste Over -> strg-a strg-v
  KBC.Cut      -> strg-x
  KBC.Cut All -> strg-a strg-x
  KBC.Repeat   -> strg-y[|n]   n - is a number that will specify how often to
press strg-y
  KBC.Undo     -> strg-z[|n]   n - is a number that will specify how often to
press strg-z
  KBC.Italic   -> strg-i
  KBC.Bold     -> strg-b
  KBC.Underline  -> strg-u
  KBC.from Right[|n] -> {END} (Number of {RIGHT})
  KBC.from Left[|n] -> {HOME} (Number of {LEFT})
  KBC.Mark from Right[|n] -> {HOME}{SHIFT} (Number of {RIGHT})
  KBC.Mark from Left[|n] -> {END}{SHIFT} (Number of {LEFT})
  KBC.Mark to Right -> {HOME}{SHIFT}
  KBC.Mark to Left -> {END}{SHIFT}
```

There is a new series of `H-XXXX` Commands.
These commands use `{HOME}{SHIFT}{END}` instead of `strg-a`
to mark the complete Text in a Textbox. This is in some cases more compatible.
There are:

```
  KBC.H-Copy All
  KBC.H-Mark All
  KBC.H-Del All
  KBC.H-Cut All
  KBC.H-Paste Over
```

As it is just like if "**somebody is really typing**", it has all advantages and all limitations from that.

- The target control must have Input focus
- the target window must be in front
- you may need to wait a bit (`PAU.0.5`) after typing until the control is ready
- `KBC.` its slower then any direct engine 231 command
- but its highly compatible
- it understands special keyboard-actions like ctrl- or Shift or Alt-Key (see below).
- It can not change the text of buttons or controls like `SDT..`

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

KBC.P1[|P2]

 © 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - Prototype any of these:

- ***KBC.Type If*** *-> will read the text at the currently localized place and compare it with* **P2.**
  If they do not match it will make "del all" and type the new text just like SKP. would.
- ***KBC.Mark All*** *-> this will click into and type ctrl-a to mark the whole content of a browser-*
  or textcontrol that is currently localized.
- ***KBC.H-Mark All*** *-> this will click into and type {SHIFT} {HOME} {END} to mark the whole content*
  of a browser- or textcontrol that is currently localized.
- ***KBC.Del All*** *-> will call* ***"mark all"*** *and then type a Backspace just like* ***SKP.{BS}.***
  This should delete all content of a text-control even in a browser.
- ***KBC.H-Del All*** *-> will call* ***"h-mark all"*** *and then type a Backspace just like* ***SKP.{BS}.***
  This should delete all content of a text-control even in a browser.
- ***KBC.From Left*** *-> will click into the Textbox, type "HOME" to go to left side.*
  Then type "RIGHT" as many times, as the **P2***-Parameter specifies.*
- ***KBC.From Right*** *-> will click into the Textbox, type "END" to go to right side.*
  Then type "LEFT" as many times, as the **P2***-Parameter specifies.*
- ***KBC.Mark from Left*** *-> will click into the Textbox, type "HOME","SHIFT" to go to left side.*
  Then type "RIGHT" as many times, as the **P2***-Parameter specifies.*
- ***KBC.Mark from Right*** *-> will click into the Textbox, type "END","SHIFT" to go to right side.*
  Then type "LEFT" as many times, as the **P2***-Parameter specifies.*
- ***KBC.Copy -> strg-c***
- ***KBC.Copy All -> strg-a strg-c***
- ***KBC.H-Copy All -> like "h-mark all" and "strg-c"***
- ***KBC.Paste -> strg-v[\|n]***
- ***KBC.Paste Over -> strg-a strg-v***
- ***KBC.H-Paste Over -> like "h-mark all" and "strg-v"***
- ***KBC.Cut -> strg-x***
- ***KBC.Cut All -> strg-a strg-x***
- ***KBC.H-Cut All -> like "h-mark all" and "strg-x"***
- ***KBC.Repeat -> strg-y[\|n]***
- ***KBC.Undo -> strg-z[\|n]***
- ***KBC.Italic -> strg-i***
- ***KBC.Bold -> strg-b***
- ***KBC.Underline -> strg-u***

© 2023 Theo Gottwald * https://www.smart-package.com

- ***KBC.Mark to Left -> "SHIFT" + "Home" (no Click)***
- ***KBC.Mark to Right -> "SHIFT" + "END" (no Click)***

P2 - (optional) Depending on the Subcommand you can specify a second Parameter.
With "`Type If`" this is a Text, with other Subcommands, this will be a number "how often"
to repeat the specified Typing-Action. If **P2** is omitted, it is assumed to be 1.

Example

```
' Samples for KBC.
KBC.mark all
KBC.del all
KBC.Type If|Login-Text
KBC.Paste|5
KBC.from Left|6
END.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- SKP. Send-Key-Press 2289
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- WNK. - Wait-No-Key 2296
- AGR. - Accessibility Get Rect 564

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.7 KYD. - Key Down

MiniRobotLanguage (MRL)

KYD. Command

Key Down Simulation

Intention

The `KYD.`.-Command can be used to simulate Keyboard button press.

It's the most compatible and "Low Level" Keyboard simulation we have. Using it, you can even make window to bring you the Task-Manager up, by simulating a Key-press:

```
' Bring Windows Task-Manager up
KYD.{STRG},{SHIFT},{ESC}
PAU.0.100
KYU.{STRG},{SHIFT},{ESC}
ENR.
```

`KYD.` understands three sorts of parameters. These are:

- Special Sequences in {..} like `{SHIFT}`
- Letters and numbers, in `"."` like `"A"` or `"B"`. There is no difference between upper- and lowecase letters, because the letter size depends on the state of the `SHIFT-Key`..
- dezimal numbers like `65` (would be the same like `"A" or "a"`)

You can NOT use multiple letters between the `"."`.

All these Parameters can be used in a single statement. They must be separated by comma. Here is an example:

```
' Open "File Open Dialog" in WordPad
STW.ct|WordPadClass|- WordPad
KYD.{STRG},"o"
KYU.{STRG},"o"
ENR.
```

Here is the same example using numbers:

```
' Open "File Open Dialog" in WordPad
STW.ct|WordPadClass|- WordPad
KYD.17,79
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' You can mix numbers and chars
KYU.17,"o"
ENR.
```

Important: Always use `KYD.` and `KYU.` paired. Otherwise you will end up with Keys pressed after your script has ended.

Advanced Hint:
If you have Problems to delete the Content in a Browser-Edit Control, you can use this combination:

```
'-----------
:mark_all
MLI.
PAU.0.1
KYD.{STRG}
SKP.a
PAU.0.1
KYU.{STRG}
PAU.0.2
RET.
'-----------
:del_all
GSB.mark_all
SKP.{BS}
RET.
'-----------
```

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

KYD.P1

Parameter Explanation

P1 - VAR or STRING

All Low-Level-Keycodes: see:
Keyboard-Keycode-List 2235

Here are **just some** often used keycodes:

Special-Sequence	Keycode	Simulates:
{BS},{BACKSPACE}	8	BACKSPACE (deletes to left side)
{TAB}	9	TAB
{DEL},{CLEAR}	12	Delete-Key (deletes to right side),

© 2023 Theo Gottwald * https://www.smart-package.com

```
                                  see also 46
{RETURN}              -  13  -    RETURN-Key
{SHIFT}               -  16  -    SHIFT-Key (general), for L/R see
                                  160,161
{CTRL},{STRG}         -  17  -    CTRL-Key (general), for L/R see
                                  162,163
{ALT},{MENU}          -  18  -    ALT-/MENU-Key (general), for L/R
                                  see 164 and 165
{POS1},{HOME}         -  36  -    HOME/POS1-Key
{LEFT}                -  37  -    Left Arrow-Key
{UP}                  -  38  -    Up Arrow-Key
{RIGHT}               -  39  -    Right Arrow-Key
{DOWN}                -  40  -    Down Arrow-Key
{INSERT}              -  45  -    INSERT-Key
{DELETE}              -  46  -    DELETE-Key
{LWIN}                -  91  -    Left Windows-Button
{RWIN}                -  92  -    Right Windows Button
{LSHIFT}              -  160 -    Left SHIFT Key
{RSHIFT}              -  161 -    Right SHIFT Key
{LCTRL},{LSTRG}       -  162 -    Left CTRL-Key
{RCTRL},{RSTRG}       -  163 -    Right CTRL-Key
{LALT},{LMENU}        -  164 -    Left ALT-Key
{RALT},{RMENU}        -  165 -    Right ALT-Key, ALT Gr.
```

Hint: To get **`ALT Gr`**, you need to use the two Keys: `165 and 162`. See example below.

```
{OEM_1},{UE},{UML_UE}     -  186 - German "Ü"-Umlaut
{OEM_3},{OE},{UML_OE}     -  192 - German "Ö"-Umlaut
{SS},{GER_SS}             -  219 - German "ß"-Umlaut
{AE},{UML_AE}             -  222 - German "Ä"-Umlaut
{PIPE}                    -  226 - the "<"-Key, below the "a",
                                   only on german keyboards.
```

Example

```
' Open Taskmanager
KYD.{LCTRL},{LSHIFT},{ESC}
PAU.0.01
KYU.{LCTRL},{LSHIFT},{ESC}
ENR.

' Open "File Open Dialog" in WordPad
STW.ct|WordPadClass|- WordPad
KYD.{STRG},"o"
KYD.{STRG},"o"
ENR.

' Internally the right ALT-Key also presses the left CTRL-Key.
KYD.{LCTRL},{RALT}
PAU.0.1
KYU.{LCTRL},{RALT}
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

You can not simulate `{STRG},{RALT},{DELETE}` to get the or you can call the Task-Manager like this Taskmanager:

```
' Open Taskmanager using Keyboard.
KYD.{LCTRL},{LSHIFT},{ESC}
PAU.0.01
KYU.{LCTRL},{LSHIFT},{ESC}
ENR.
```

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- SKP. Send-Key-Press 2289
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.8 KYU. - Key Up

MiniRobotLanguage (MRL)

KYU. Command

Key Up Simulation

Intention

The `KYU.`-Command can be used to simulate Keyboard button release.

Important: Always use `KYD.` and `KYU.` paired. Otherwise you will end up with Keys pressed after your script has ended.

It's the most compatible and "Low Level" Keyboard simulation we have. Using it, you can even make window to bring you the Task-Manager up, by simulating a Key-press:

```
' Bring Windows Task-Manager up
KYD.{STRG},{SHIFT},{ESC}
PAU.0.100
KYU.{STRG},{SHIFT},{ESC}
ENR.
```

`KYD.` understands three sorts of parameters. These are:

- Special Sequences in {..} like `{SHIFT}`
- Letters and numbers, in `"."` like `"A"` or `"B"`. There is no difference between upper- and lowecase letters, because the letter size depends on the state of the `SHIFT-Key`..
- dezimal numbers like `65` (would be the same like `"A" or "a"`)

You can NOT use multiple letters between the `"."`.

All these Parameters can be used in a single statement. They must be separated by comma. Here is an example:

```
' Open "File Open Dialog" in WordPad
STW.ct|WordPadClass|- WordPad
KYD.{STRG},"o"
KYU.{STRG},"o"
ENR.
```

Here is the same example using numbers:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Open "File Open Dialog" in WordPad
STW.ct|WordPadClass|- WordPad
KYD.17,79
' You can mix numbers and chars
KYU.17,"o"
ENR.
```

KYU. knows all keys that have been pressed by the **LAST** KYD.-Instruction.
You can release them all in the right order, by writing:

```
KYU.all
```

For example:

```
' Bring Windows Task-Manager up
KYD.{STRG},{SHIFT},{ESC}
PAU.0.100
' release Keys in order
KYU.{ESC},{SHIFT},{STRG}
ENR.
```

is exactly the same as:

```
' Bring Windows Task-Manager up
KYD.{STRG},{SHIFT},{ESC}
PAU.0.100
' release Keys in order LILO
KYU.all
ENR.
```

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

KYD.P1

Parameter Explanation

P1 - VAR or STRING

All Low-Level-Keycodes: see:
Keyboard-Keycode-List 2235

Here are **just some** often used keycodes:

Special-Sequence	Keycode	Simulates:

© 2023 Theo Gottwald * https://www.smart-package.com

```
{BS},{BACKSPACE}     -  8   -  BACKSPACE (deletes to left side)
{TAB}                -  9   -  TAB
{DEL},{CLEAR}        -  12  -  Delete-Key (deletes to right
side),
                               see also 46
{RETURN}             -  13  -  RETURN-Key
{SHIFT}              -  16  -  SHIFT-Key (general), for L/R see
                               160,161
{CTRL},{STRG}        -  17  -  CTRL-Key (general), for L/R see
                               162,163
{ALT},{MENU}         -  18  -  ALT-/MENU-Key (general), for L/R
                               see 164 and 165
{POS1},{HOME}        -  36  -  HOME/POS1-Key
{LEFT}               -  37  -  Left Arrow-Key
{UP}                 -  38  -  Up Arrow-Key
{RIGHT}              -  39  -  Right Arrow-Key
{DOWN}               -  40  -  Down Arrow-Key
{INSERT}             -  45  -  INSERT-Key
{DELETE}             -  46  -  DELETE-Key
{LWIN}               -  91  -  Left Windows-Button
{RWIN}               -  92  -  Right Windows Button
{LSHIFT}             -  160 -  Left SHIFT Key
{RSHIFT}             -  161 -  Right SHIFT Key
{LCTRL},{LSTRG}      -  162 -  Left CTRL-Key
{RCTRL},{RSTRG}      -  163 -  Right CTRL-Key
{LALT},{LMENU}       -  164 -  Left ALT-Key
{RALT},{RMENU}       -  165 -  Right ALT-Key, ALT Gr.
```

Hint: To get **`ALT Gr`**, you need to use the two Keys: `165 and 162`. See example below.

```
{OEM_1},{UE},{UML_UE}    -  186 - German "Ü"-Umlaut
{OEM_3},{OE},{UML_OE}    -  192 - German "Ö"-Umlaut
{SS},{GER_SS}            -  219 - German "ß"-Umlaut
{AE},{UML_AE}            -  222 - German "Ä"-Umlaut
{PIPE}                   -  226 - the "<"-Key, below the "a",
                                  only on german keyboards.
```

Example

```
' Open Taskmanager
KYD.{LCTRL},{LSHIFT},{ESC}
PAU.0.01
KYU.{LCTRL},{LSHIFT},{ESC}
ENR.

' Open "File Open Dialog" in WordPad
STW.ct|WordPadClass|- WordPad
KYD.{STRG},"o"
KYD.{STRG},"o"
ENR.

' Internally the right ALT-Key also presses the left CTRL-Key.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
KYD.{LCTRL},{RALT}
PAU.0.1
KYU.{LCTRL},{RALT}
ENR.
```

Remarks

-

Limitations:

You can not simulate `{STRG},{RALT},{DELETE}` to get the Taskmanager, as these Keys are intercepted below application-Level. You can however use this one:

```
' Open Taskmanager using Keyboard.
KYD.{LCTRL},{LSHIFT},{ESC}
PAU.0.01
KYU.{LCTRL},{LSHIFT},{ESC}
ENR.
```

See also:

- KYD. - Key Down 2256
- ! Keyboard-Keycode-List 2235
- 1.9.1.6 Edit-Textbox Control 194
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- SKP. Send-Key-Press 2289
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.9 RST. - Replace Selected Text

MiniRobotLanguage (MRL)

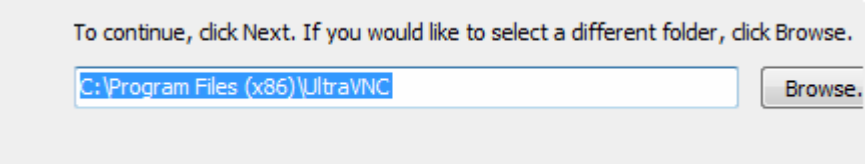
RST. Command

Replace Selected Text

Intention

The `RST.`-Command can be used to set a textbox to a specified text. It will change the Text of the actual localized Top-/or Child-window to the text that you specify.

Here is an example for a textbox.

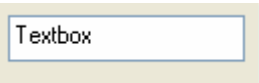


`RST.` uses the Direct engine 231 and therefore does not interpret any special Keyboard sequences like `SKP`. Using `RST.` the text is been send directly to the window and will replace the part of the text that is specified with the numeric parameters.

The command will select the Text given with the Parameters **P2**,**P3** and then replace this Text (-Part) with **P1**. The unselected Text remains as before.

To signal that all Text until the last Character should be replaced, the **P2** has to be "**-1**" which means " until End of Text".

Here is an example. This is a textbox in our Test-Application.

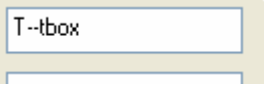


Now we start this small script:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nc|2|Edit
RST.--|1,3
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

And this is the textbox after the script:



As you can see the numeric argument is zero-based. You can use variables and formulas in (..) for the numeric arguments:

```
RST.--|1,(2+1)
```

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

RST.P1[|P2,P3]

Parameter Explanation

P1 - Text that is sent to the textbox

P2 - ***(optional) zero based*** character number, start of selection.
If omitted 0 is the default.

P3 - ***(optional)*** character number, end of selection.
If omitted, last character (-1) is the default.

Example

```
' Replace the whole Text with the new Text
RST.$$TXT|0,-1

' Replace the whole Text with the new Text
RST.$$TXT

' Clear Text
RST.

' Replace Letters 2 to 5 with the new Text
RST.$$TXT|2,5

' Replace Letters 1 to 3 with ---
RST.---|1,(2+1)

'********************************
' RST. - Sample
'********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
EXE.?exepath\Sample Apps\TestApp_01.exe

STW.ct|PBWindowClass:0|TestApp
SCW.nc|2|Edit
RST.We can write into a  Text-box

ENR.
```

Remarks

Direct Engine command.

Limitations:

`RST.` will only work on windows. It will normally not work with WEB-based or Office-Controls.

See also:

- 1.9.1.6 Edit-Textbox Control 194
- SDT. - SenD-window-Text 2278
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- WNK. - Wait-No-Key 2296
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.10 SDA. / SDU. - Send Input ANSI/UNICODE

MiniRobotLanguage (MRL)

SDA./SDU. Command

Send Keypresses using a different method that is between Direct and Compatible Engine.

Intention

Technically SDA. / SDU. places the characters directly the keyboard buffer.So they do the same.
. The only difference is that SDU. contains Optimizations for sending UNICODE-Text. It may only make sense in Applications who can receive such special Text.
This SendInput Technology is faster then other Compatible Engine Typing Commands. But its slower then the Direct-Engine Commands. So its some where in between. Use it if you can use it for your usage.

The `SDA. and the SDU.`-Command

```
STW.ct|WordPadClass|Dokument - WordPad
'SWP.2605,585|539,283|T
SCW.nc|1|RICHEDIT50W|{&NOTEXT:}

' This is our Text
$$TXT=Hallo Welt{ENTER}
' We look at the Hexdump
PRT.$$TXT|h
'  Now we make it UNICODE
ATU.$$TXT
' We look at the Hexdump
PRT.$$TXT|h
' Click into Wordpad
MLM.w|423,139
' Do the typing
SDU.$$TXT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SDA.$$TXT
MBX.Look
ENR.
```

Then Windows will send the characters to the application that currently has the Input-focus.

Apart from normal Ascii characters, SDA. / SDU. can also handle some special characters that are unique just for these commands:

```
'
' These special Tokens are currently implemented:
{CTRL_D} + {CTRL_U}
{ALT_D} +  {ALT_U}
{SHIFT_D} +{SHIFT_U}
{WIN_D} + {WIN_U}

{CTRL}
{ALT}
{SHIFT}
{WIN}

{BACKSPACE},{ENTER},{INSERT},{DELETE}
{RIGHT},{LEFT},{UP},{DOWN}
{HOME}, {END}
{TAB}, {ESCAPE}
{PGUP},{PGDN}
{F1}, {F2}, {F3}, {F4}, {F5}, {F6}, {F7}, {F8}, {F9}, {F10

{APPS}
{NUMLOCK}

' The {.._D} are the 'key down' codes.
' The {.._U} are the 'key up' codes.
'
'Examples of use:
'SDA.This is a TEST
'SDU.12{BACKSPACE}{BACKSPACE}abcdef{ENTER}
'SDA.{CTRL_D}ac{CTRL_U}" simulates a CTRL-a  CTRL-c key co
'SDU.{CTRL}a{CTRL}c" simulates a CTRL-a  CTRL-c key combin
' You can use Variables:
' SDA.$$TXT
```


© 2023 Theo Gottwald * https://www.smart-package.com

These commands have their own specific Special Characters. And They work with an own internal System, therefore an intermixing with other such commands may not be a good idea.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SDA.P1
SDU.P1

Parameter Explanation

P1 – `VAR or STRING` with xharacters and Special Tokens, see above.

Example

```
' Type into Ultraedit
STW.Act|Afx:|UltraEdit-32
SCW.c|EditControl
SAW.
VAR.$$TXA=Hallo Welt{ENTER}
SDA.$$TXA{ENTER}
END.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- SKP. Send-Key-Press 2289
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278

© 2023 Theo Gottwald * https://www.smart-package.com

- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.11 SDH. - Send Keys Hexcode

MiniRobotLanguage (MRL)

SDH. Command

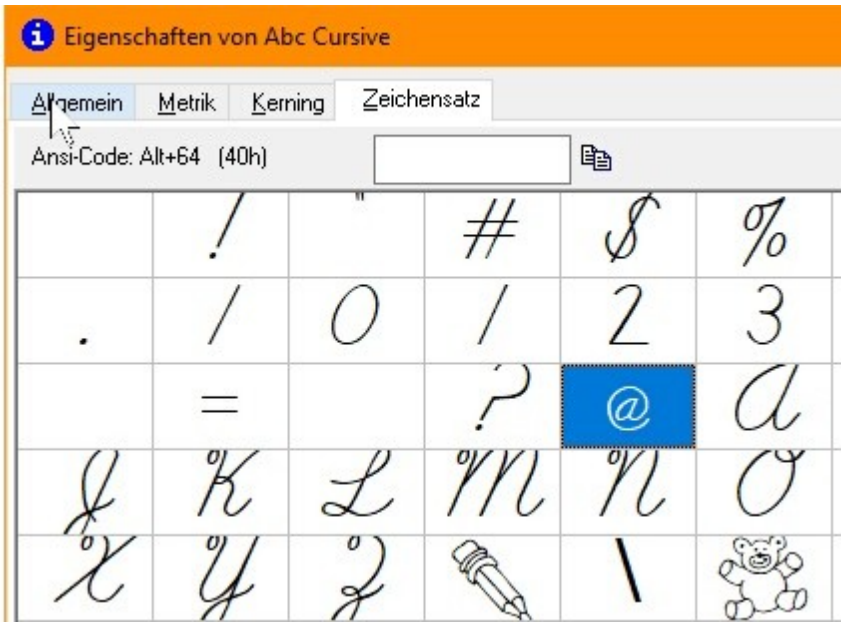
Send Keys delimited by a comma using Numbers

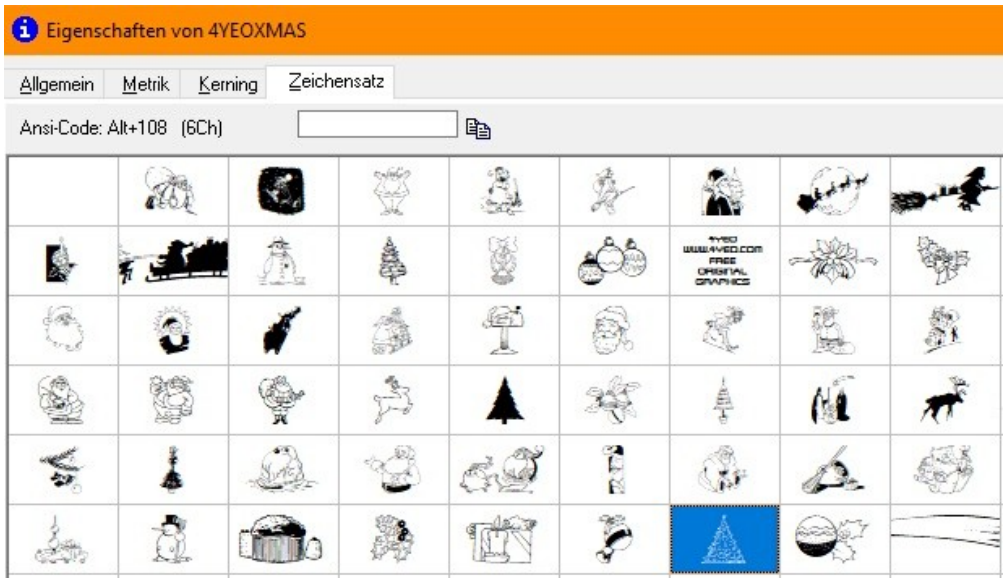
Intention

Do you remember using the Keypad to enter special characters?
In the Font-Description you find for each character the Number that you will need to type - together with the left "ALT"-Key.

Here is an example, if you hold the "Left-ALT" and then press "64"on the Num-Pad and then leave the ALT-Key, a "@" Sign will appear.
Of course the @-Sign can be found more easy, but there are other fonts and even foreign language signs where this becomes handy.

*© 2023 Theo Gottwald * https://www.smart-package.com*





The `SDH.`-Command isperfectly compatible and will accept "," so you can write:

```
SDH.144,145,165
or you can use other numeric Systems including Hexadezimal

SDH.&H89,&H54,&H3A,33,&B10101100,&O123

' Mathematical Formulas and Variables are also supported -
SDH.(143+1),(145-1),144

' using the supported ! > < = Option, you can even code "I
```

with as many characters as you need.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

SDH.P1

Parameter Explanation

P1 – `VAR or STRING` with comma separated numbers that define the characters to type. You can use decimal numbers or other numeric Systems including Hexadezimal, Octal or Binary like this:

```
SDH.&H89,&H54,&H3A,33,&B10101100,&O123,(144+$$ZPO*(3+$$O
```

Example

```
' Type into Ultraedit
STW.Act|Afx:|UltraEdit-32
SCW.c|EditControl
SAW.
VAR.$$TXA=65,66,657
SDH.$$TXA,89,45
END.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- SKP. Send-Key-Press 2289
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.12 SDK. - SendKeys VB

MiniRobotLanguage (MRL)

SDK. Command

Send Keys with VB-Rules

Intention

The `SDK.`-Command can be used to simulate Keyboard typing. SDK. is like if you really type something into the keyboard. This makes it compatible with real typing. It is not as compatible as it is SKI. or SKP. or KYU. and KYD.
Therefore it is faster. It is not as fast as SDT. its like in between SKI. and SDT.

`SDK.` will work on any place, where you can also enter text using the keyboard.It will work also in browsers, Java- and WEB-based controls.
As it is just like if "**somebody is really typing**", it has all advantages and all limitations from that.

SDK. has a different Syntax then the other Commands as its based on another System (VB-like Sendkeys).

- The target control must have Input focus
- the target window must be in front
- you may need to wait a bit (`PAU.0.5`) after typing until the control is ready
- `SDK.` its slower then any direct engine 231 command
- but its highly compatible
- it understands special keyboard-actions like ctrl- or Shift or Alt-Key (see below).
- It can not change the text of buttons or controls like `SDT.`

`SDK.` **emulates most of the same rules as the VisualBasic Sendkeys-Command.**

```
VAR.$$TXA=abc$crlf$cba
SDK.$$TXA
```

You can send these codes Special-Codes:

```
  Generally as a Prefix:
  "+" is SHIFT
  "^" is CTRL
  "%" is ALT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Use "(" and ")" to GROUP Shifted Keys.
Like "+(ab)"

{ENTER}
{ESC}
{UP}  {DOWN}  {LEFT}  {RIGHT}
{BACKSPACE} or {BS}
{HOME}  {END}
{PGUP}  {PGDN}
{TAB}
{DEL} or {DELETE}
{INS} or {INSERT}
{WIN}
{PRTSC}
{CAPSLOCK}
{NUMLOCK}
{SCROLLLOCK}
{HELP}
{PAUSEKEY}
{BREAK}
{LCONTROL} {RCONTROL}
{LSHIFT}{RSHIFT}
{ATTN}
{LMENU} {RMENU}
{F1}-{F16}
{NOTHING} '  so you can use Alt etc. without any Letter: %{NOTHING}

{SLEEP 1000} <-- 1000 miliseconds Pause
{PAUSE 3.5} <-- Pauses 3.5 seconds (while processing event messages)
{App WindowName} <-- switches to this window (does NOT support any wildcards, ne
{LMOUSECLICK X Y} <-- Left button mouse click at x/y screen position
{RMOUSECLICK X Y} <-- Right button mouse click at x/y screen position
{MMOUSECLICK X Y} <-- Middle button mouse click at x/y screen position
{CHR 123} <-- Send an ascii code 123 (same as pressing Alt-123 example: {CHR 65}
{AppActivate WindowName} will switch focus to the WindowName you specify
```

Procede characters with + for Shift, ^ for Ctrl, % for Alt example: `^p` would be `Ctrl-p`

You can also group characters for shift/ctrl/alt with parentheses ( ) example: `%(fx)` would be `Alt-F Alt-X`

To send these special characters `{+^%~` (as plain characters enclose them in braces example: `{{}` would just send {

To send a key multiple times inclose it in braces with a number like `{h 3}` example: `{DEL 41}` would send DELETE key 41 times

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SDK.P1

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** VAR or STRING

Example

```
' Type into Ultraedit
STW.Act|Afx:|UltraEdit-32
SCW.c|EditControl
SAW.
VAR.$$TXA=abc$crlf$cba
SDK.$$TXA
END.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- SKP. Send-Key-Press 2289
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.13 SDT. - SenD-window-Text

MiniRobotLanguage (MRL)

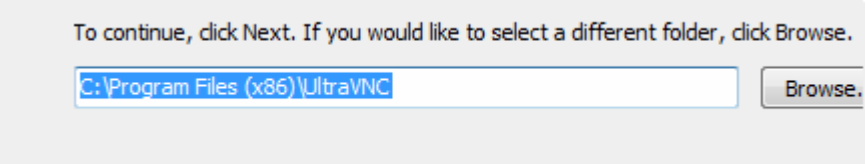
SDT. Command

SenD windowText

Intention

The SDT.-Command can be used to set a textbox to a specified text. It will change the Text of the actual localized Top-/or Child-window to the text that you specify.

Here is an example for a textbox.



SDT. uses the Direct engine 231 and is therefore much (50 times?) faster then SKP. It does not interpret any special Keyboard sequences like SKP.
Using SDT. the text is been send directly to the window and will replace the part of the text that is specified with the numeric parameters.
As it does not really simulate the keyboard like the Compatible Engine 231 commands, it can not be used to open menu's or like that. It's perfect to change the text of a Top- or child-window.

Usage is as simple as possible. To clear a Textbox write:

```
SDT.
```

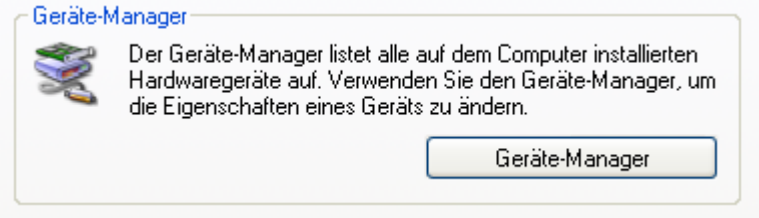
and to send a text into a textbox, write:

```
SDT.(your text)
```

SDT. can also change the text of statics or buttons as long as it finds a window control as target. Take this for example:

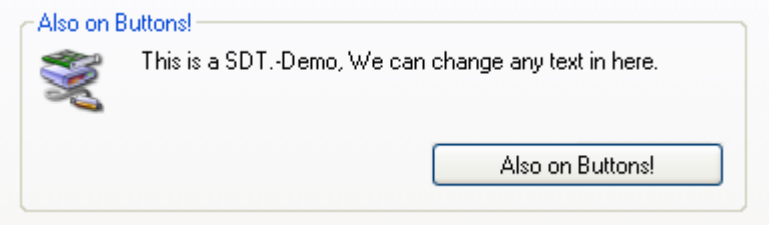
Before:

 *© 2023 Theo Gottwald * https://www.smart-package.com*



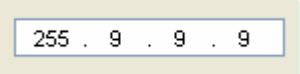

```
We use this script (german WinXP, SP3)

STW.ct|#32770|Systemeigenschaften
SCW.nct|1|#32770|Hardware
SCW.nct|1|static|Der Geräte-Manager listet.
SDT. This is a SDT.-Demo, We can change any text in here.

FOR.$$CNT|1|2
  SCW.ctT|#32770|Hardware
  SCW.Sct|Button|Geräte-Manager
  SDT.Also on Buttons!
NEX.
ENR.
```

And this is how it looks after:



You can set the text in a SysIP-Control directly:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|SysIPAddress32|192.168.0.1
' Set the SysIP-Control in one strike
SDT.255.9.9.9
```



Note: `SDT.` will not work in Office-, Java- or WEB-based controls.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SDT.P1

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** VAR or STRING

Example

```
'********************************
' SDT. - Sample
'********************************
'
EXE.?exepath\Sample Apps\TestApp_01.exe

STW.ct|PBWindowClass:0|TestApp
SCW.nc|2|Edit
SDT.We can write into a  Text-box

STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|SysIPAddress32|192.168.0.1
' Set the SysIP-Control in one strike
SDT.255.9.9.9

STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
SDT. change a button-text

STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Button|Check1
SDT.change this also
ENR.
```

Remarks

-

Limitations:

SDT. will only work on windows. It will normally not work with WEB-based or Office-Controls.

See also:

- 1.9.1.6 Edit-Textbox Control 194
- RST. - Replace Selected Text 2264
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.14 SEK. - Set Extended Key-State

MiniRobotLanguage (MRL)

SEK. Command

Set Extended Key-State

Intention

The SEK.-Command has two modes of operation.

In **Mode 1**, it can be used to set the state of NUMLOCK, CAPSLOCK and SCROLLLOCK.
What you set is really the "**state of the keyboard**". Not "the state of the button".

SEK.-will change the "real thing" exactly to the value that you provide in **P2**.

```
' Will set CAPSLOCK to ON
SEK.CAPSLOCK|1
```

In **Mode 2**, it can be used to set the state of the SHIFT, CTRL or the ALT-Key and others, see detailed list below. In this case you really change the state of the key, just like if you press or release the key physically.
Of course, if CAPSLOCK is activated, you will get a "1" if you check for SHIFT, even if you try to release either of the SHIFT keys.

```
' Will release the left SHIFT Button
SEK.LSHIFT|0
' Will press the right SHIFT Button
SEK.RSHIFT|1
```

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SEK.P1[|P2]

Parameter Explanation

P1 **-** VAR or numeric

P2 **-** ***(optional) numeric,*** variable or value 0/1 . If omitted 0 is used.

Parameter-List for **P1:**

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
SHIFT          - any shift key (keyboard Shift-state)
LSHIFT         - only left Shift-Key
RSHIFT         - only right Shift-Key
STRG,CTRL      - any Ctrl key (keyboard Ctrl-state)
LSTRG,LCTRL    - only left Ctrl-Key
RSTRG,RCTRL    - only right Ctrl-Key
MENU,ALT       - any ALT key (keyboard ALT-state)
LMENU,LALT     - only left ALT-Key
RMENU,RALT     - only right ALT-Key
NUMLOCK,NUM    - keyboard NUMLOCK-State
CAPSLOCK,CAPS - keyboard CAPSLOCK-State
SCROLLOCK,SCROL,SCROLL - keyboard SCROLLLOCK-State
```

Example

```
'*****************************************************
' EXAMPLE 1: SEK.
'*****************************************************
SEK.CAPSLOCK|1
GEK.SHIFT|$$STA
DBP.$$STA
GEK.LSHIFT|$$STA
DBP.$$STA
GEK.RSHIFT|$$STA
DBP.$$STA
SEK.CAPSLOCK|0
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- GEK. - Get Extended Key-State 2240
- GKS. - Get Key-State 2243
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.15 SKI. - Send Key-Input

MiniRobotLanguage (MRL)

SKI. Command

Send Key Input

Intention

The `SKI.`-Command can be used to simulate Keyboard typing. SKI. is like if you really type something into the keyboard. This makes it very compatible with real typing 231.

`SKI.` uses internally `KYU.` and `KYD.` and can therefore intermixed with these Commands. It can also understand the Syntax from these Commands..
`SKI.` will work on any place, where you can also enter text using the keyboard.It will work in browsers, Java- and WEB-based controls.
Unlike with KYD. and KYU. you can type complete words into SKI. SKI. will internally format it and type it.

```
SKI.Hallo Welt
```

SKI. uses a special "Character by Character" Syntax. It will really type "Letter by letter". This is slower but it has its advantages in safety.

Here is an Example that will type something into Notepad.

As the ","-Sign is used as a delimiter for SKI. it can not be sent to the target window using SKI.
You can use the special Keycode `{OEMComma}` to send a Comma using SKI.

```
STW.ct|WordPadClass|Dokument - WordPad
SCW.nc|1|RICHEDIT50W|{&NOTEXT:}
MLM.w|423,139
SKI.This,{OEMComma},Hallo
SEK.{LALT}|1
SKI.{NUM_1},{NUM_4},{NUM_4}
SEK.{LALT}|0
ENR.

You can also use this {DN_ and {UP_ Option to just do it all in SKI.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
STW.ct|WordPadClass|Dokument - WordPad
'SWP.2605,585|539,283|T
SCW.nc|1|RICHEDIT50W|{&NOTEXT:}
MLM.w|423,139
SKI.{DN_LALT},{NUM_1},{NUM_4},{NUM_4},{UP_LALT}
ENR.

'While especially this example could have been done more easy using SDH.
'Like this:

SDH.144
```

As it is just like if "**somebody is really typing**", it has all advantages and all limitations from that.

- The target control must have Input focus
- the target window must be in front
- you may need to wait a bit (`PAU.0.5`) after typing until the control is ready
- `SKI.` its slower then any direct engine 231 command
- but its highly compatible
- it understands special keyboard-actions like ctrl- or Shift or Alt-Key (see below).
- It can not change the text of buttons or controls like `SDT.`.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SKI.P1

Parameter Explanation

P1 - `VAR or STRING can contain all Low-Level-Keycodes, may also contain Letters or words. All must be separated by ",". See:`

Keyboard-Keycode-List 2235

SKI. supports the prefix of `{DN_ and {UP_` for all cases where you only want to press a button down and hold it, or release a pressed button.

Note:

1.You can use this command to open Menu-Items using the Keyboard.

© 2023 Theo Gottwald * https://www.smart-package.com

2. Always use small letters. In the context of this command, letters are generally small, unless additionally the SHIFT-Key is pressed.

Example

```
STW.ct|WordPadClass|Dokument - WordPad
SCW.nc|1|RICHEDIT50W|{&NOTEXT:}
MLM.w|423,139
SKI.{DN_LALT},{NUM_1},{NUM_4},{NUM_4},{UP_LALT}
ENR.


' SKI. can be used with Variables
STW.ct|WordPadClass|Dokument - WordPad
SCW.nc|1|RICHEDIT50W|{&NOTEXT:}
MLM.w|423,139
$$TXT={DN_LALT},{NUM_1},{NUM_4},{NUM_4},{UP_LALT}
SKI.$$TXT
ENR.




SKI.$$TXT
SKI.Hallo,64,{ENTER}
```

Remarks

For easier Typing without Commas between Elements you can also try SKP. or SKT.

Limitations:

SKI. works on the lowest Level that the robot can reach. If there are windows which you can not type in, please make sure that you have Admin-Rights for the Robot-Executable/SPRE.

```
' This will not work, as SKP. internally will undo the Effects of SEK.
SEK.LALT|1
SKP.{NUM_1}{NUM_4}{NUM_4}
SEK.LALT|0

' This will work, as KYD. and KYU. are specially made for this.
KYD.{LALT}
KYD.{NUM_1}
PAU..1
KYU.{NUM_1}
PAU..1
KYD.{NUM_4}
PAU..1
KYU.{NUM_4}
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU..1
KYD.{NUM_4}
PAU..1
KYU.{NUM_4}
KYU.{LALT}

' OR USE
' This will also work
SEK.LALT|1
SKI.{NUM_1},{NUM_4},{NUM_4}
SEK.LALT|0
```

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- **SKT. - Send-Key-Text** 2293
- **KYD. - Key Down** 2256
- **KYU. - Key Up** 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- SKT. - Send-Key-Text 2293
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.16 SKP. Send KeyPress

MiniRobotLanguage (MRL)

SKP. Command

Send Key Press

Intention

The `SKP.`-Command can be used to simulate Keyboard typing. SKP. is like if you really type something into the keyboard. This makes it very compatible with real typing 231.
Please note that for Safety-of-Typing-Consideration SKP. can not be intermixed with - for example SEK.-Command.
There might be some some cases that will require the use of KYU. / KYD. and SDI.-Command. Use these if it will not work with SKP.

`SKP.` will work on any place, where you can also enter text using the keyboard.It will work also in browsers, Java- and WEB-based controls.

As it is just like if "**somebody is really typing**", it has all advantages and all limitations from that.

- The target control must have Input focus
- the target window must be in front
- you may need to wait a bit (`PAU.0.5`) after typing until the control is ready
- `SKP.` its slower then any direct engine 231 command
- but its highly compatible
- it understands special keyboard-actions like ctrl- or Shift or Alt-Key (see below).
- It can not change the text of buttons or controls like `SDT.`.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SKP.P1

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - VAR or STRING can contain all Low-Level-Keycodes: see:

Keyboard-Keycode-List 2235

SKP. understands **additionally** to these commands some special commands. To understand this better, let's see an example.

```
SKP.A{RETURN}
```

In this example, the SKP. will technically press each of the buttons and then release each of the buttons. Like this:

1. Press A
2. Release A
3. Press {RETURN}
4. Release {RETURN}

Now you may want to just press and hold a button down. And not release it immediately.

For example the SHIFT or the ALT-Button. This is done, by prefixing ANY of the Low-Level Key-code with a

```
{DN_...}   -  only push that button down
{UP_...}   -  only release that button
```

for example:

```
SKP.{DN_SHIFT}abc{UP_SHIFT}
```

will type:

```
ABC
```

in large letters, as we are holding down the SHIFT-Key in this example.
In the same way, you can open Menus or select menu-items.

Note:

1. To call Menu-Items using the Keyboard, prefer KYD. and KYU. especially if the CTRL, or ALT-Key is needed..

2. When combining Keys with SHIFT, always use small letters. Large letters will automatically be SHIFTED when typed.

3. When **combining Keys with CTRL or ALT,** things **may not work as expected** because safety of typing* will internally interfere with pre-pressed keys.
   Use KYD. and KYU. for such Combinations. .

4. SKP. will switch off CAPSLOCK and NUMLOCK before it starts typing. It will restore the previous state of CAPSLOCK and NUMLOCK when its ready with typing..

© 2023 Theo Gottwald * https://www.smart-package.com

5. SKP. will do a complete variable-resolution. It will resolve Specialfolders as well as Systemvariables before it starts typing.

---
* Safety of typing. Internal process to make sure that no CTRL and other keys are pressed on the physical keyboard, while we are simulating keystrokes.

Example

```
SKP..$$TXT
SKP. Hallo
```

Remarks

If you need more control over the keyboard, you can use KYD. and KYU.

Limitations:

SKP. will generally type into the window that has the Input-Focus. Also the state of the SHIFT-Key or the SCROLLLOCK or the NUMLOCK-Key can influence SKP. Therefore don't type while SKP. is at work.

```
' Some things may not work with SKP.
' For example:

' This will not work, as SKP. internally will undo the Effects of SEK.
SEK.LALT|1
SKP.{NUM_1}{NUM_4}{NUM_4}
SEK.LALT|0

' This will work, as KYD. and KYU. are specially made for this.
KYD.{LALT}
KYD.{NUM_1}
PAU..1
KYU.{NUM_1}
PAU..1
KYD.{NUM_4}
PAU..1
KYU.{NUM_4}
PAU..1
KYD.{NUM_4}
PAU..1
KYU.{NUM_4}
KYU.{LALT}

' OR USE
' This will also work
SEK.LALT|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SDI.{NUM_1},{NUM_4},{NUM_4}
SEK.LALT|0
```

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- **SKT. - Send-Key-Text** 2293
- **KYD. - Key Down** 2256
- **KYU. - Key Up** 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- SKT. - Send-Key-Text 2293
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.17 SKT. - Send - Key-Text

MiniRobotLanguage (MRL)

SKT. Command

Send Key Text

Intention

The `SKT.`-Command can be used to simulate Keyboard typing. SKP. is like if you really type something into the keyboard. This makes it very compatible with real typing 231.

`SKT.` will work on any place, where you can also enter text using the keyboard.It will work also in browsers, Java- and WEB-based controls.

As it is just like if "**somebody is really typing**", it has all advantages and all limitations from that.

- The target control must have Input focus
- the target window must be in front
- you may need to wait a bit (`PAU.0.5`) after typing until the control is ready
- `SKT.` its slower then any direct engine 231 command
- but its highly compatible
- it understands special keyboard-actions like ctrl- or Shift or Alt-Key (see below).
- It can not change the text of buttons or controls like `SDT.`.

`SKT.` - unlike `SKP.` does NOT understand any Special-Sequences like `{RETURN}` or `{SHIFT}` . It will just try to type whatever it gets. SKT. will not interpret any systemvariables or specialfolders. It will jut resolve a given variable and then start typing.

You can not make a "`NEWLINE`" with `SKT.` using `{RETURN}`. But you can make a NEWLINE like this:

```
VAR.$$TXA=abc$crlf$cba
SKT.$$TXA
```

Because `$crlf$` will be converted to `ASCII` using the `VAR.`-Command and `SKT.` will just type it. In some editors you will even get two newlines, one for the `$cr$` and one for the `$lf$`.

*© 2023 Theo Gottwald * https://www.smart-package.com*

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

SKT.P1

Parameter Explanation

P1 **-** VAR or STRING (NO specialfolders, NO systemvariables)

Example

```
' Type into Ulktraedit
STW.Act|Afx:|UltraEdit-32
SCW.c|EditControl
SAW.
VAR.$$TXA=abc$crlf$cba
SKT.$$TXA
END.
```

Remarks

-

Limitations:

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- ! Keyboard-Keycode-List 2235
- SKP. Send-Key-Press 2289
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
- WNK. - Wait-No-Key 2296

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.61.18 WNK. - Wait-No-Key

MiniRobotLanguage (MRL)

WNK. Command

Wait until No Key is pressed

Intention

The `WNK.`-Command can be used to wait until the user has stopped typing. If you specify a time, then the user must not press any key for that time, before `WNK.` will let the robot continue.

You can read more about Keyboard-Simulation and Text-Boxes in the chapter:
1.9.1.6 Edit-Textbox Control 194

Syntax

WNK.[P1]

Parameter Explanation

P1 **-** `optional VAR or numeric,`

`WNK.1.5 for examples, will wait until the user has not pressed any key for maximum 1.5 seconds.`

Example

```
'*****************************************************
' EXAMPLE 1: WNK
'*****************************************************
WNK.
WNK.0.2
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1.9.1.6 Edit-Textbox Control 194
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293
- KYD. - Key Down 2256
- KYU. - Key Up 2260
- RST. - Replace Selected Text 2264
- SDT. - SenD-window-Text 2278
-

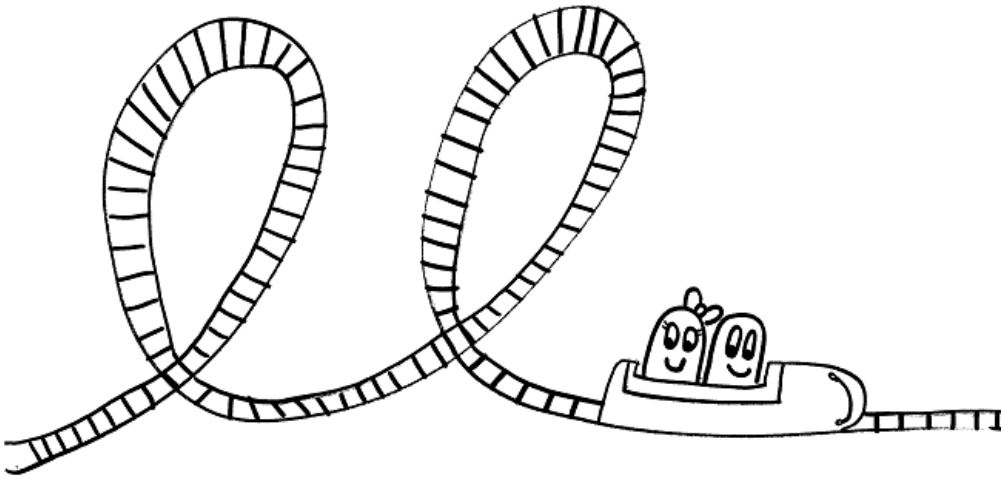
© 2023 Theo Gottwald * https://www.smart-package.com

3.62 Loops, and repeated action

3.62.1 ! Smarty's Loops

Smarty's Loops and repeated Commands **Previous** 2296 **Top** 41 **Next** 2299

Smarty's Loops and repeated Commands



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.62.2 DOL. - OOP. - Conditional Loop

MiniRobotLanguage (MRL)

DOL. ... OOP. Commands

DO ... LOOP, WHILE ... WEND or REPEAT ... UNTIL

Intention

Repeat a group of instruction in different ways. Lets start with the endless loop.

```
' Endless loop
DOL.1
...
OOP.
:exit
```

You can construct a "WHILE ... WEND" Loop.

```
' WHILE ... Loop
DOL.($$CON=1)
...
OOP.
```

This loop will be repeated WHILE ($$CON=1). In fact the formula in the condition is calculated. If the result is not zero, the loop continues running. Otherwise the loop is been left. Think of it as a "WHILE cond ... WEND" Loop.
This type of loop will never be entered, when the condition is not met. It will continue running, while the condition is not zero.

You can make a "REPEAT UNTIL" loop. Just write the condition at the bottom, after the OOP. Like this:

```
' REPEAT ... UNTIL - Loop
DOL.
...
OOP.($$CON=0)
```

This loop will run UNTIL the condition is met. The condition formula is been calculated. If the result is not zero, the loop execution continues. This type of loop will run UNTIL the condition is not zero. Also its guaranteed that the loop inside is executed at least once.

Technically you can even combine both loops and give two exit conditions. One at the top, and one at the bottom. While this special loop does not make much sense, in real life.

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

DOL. ... OOP.
DOL.P1 ... OOP.
DOL. ... OOP.P1

Parameter Explanation

P1 - Condition of loop execution

```
' You can make an endless loop like this
DOL.1
...
OOP.

' Condition on Top of Loop, prevents entering of Loop if not
met.
' Condition at top is WHILE ... WEND condition.
DOL.$$AAA<9
...
OOP.

' Condition on Bottom of Loop, guarantees
' loop is entered at least once.
' condition at bottom is "REPEAT ... UNTIL cond"
' when condition is true, loop is left!
DOL.
...
OOP.$$AAA<9

' You can have cond at Top and Bottom, too.
DOL.$$AAA<9
...
OOP.$$AAA<9
```

Speed in Ticks:

This command uses around 19 Ticks for a simple "DOL.1". With difficult to calculate Conditions as Parameters its a bit more.

Example

```
VAR.§§FIA=?pfad\myfile1.txt
VAR.§§FIB=?pfad\myfile2.txt
' This variable will be set to 1 if the file is found
: §§XIT=0
' Here is our endless loop
DOL.§§XIT=0
  PRT. I am here!
' We test for File1
  IEF.§§FIA
    : §§XIT=1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  EIF.
' We test for File2
  IEF.§§FIB
    : §§XIT=1
  EIF.
  ' We'll give the CPU a tenth of a second sleep time
  ' always do this in endless loops to save CPU time.
  ' for other running programs
  PAU.0.1
OOP.
:exit
MBX. A file appeared!
ENR.
```

Remarks

While technically not needed, its recommend that you put the condition in formula brackets (..).

Limitations:

There are **no limitations** about nesting of Loop and other constructs in any way. The condition formula can be as complex as you want. It will be calculated using the same rules like in the CAL. - Instruction.

See also:

- 6. Program Flow Control 121
- 1.6.2. Looping around 126
- FOR. - NEX. Counting Loop 2302
- 1.5. Features and Hints 80

© 2023 Theo Gottwald * https://www.smart-package.com

3.62.3 FOR. - NEX. Counting Loop

MiniRobotLanguage (MRL)

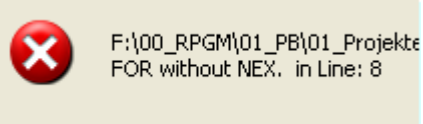
FOR. Command

For NEX Counting Loop

Intention

If you don't know the terms "Loop" and have no idea what a "FOR ... NEXT" Loop is, please read the section: 1.6.2. Looping around 126.
This section describes the basics of Loops and the `FOR ... NEX` - Loop.

The `FOR. ... NEX.` commands belong together and need to be paired in your script.
For each FOR. there must be a matching `NEX.` If that is not the case, you will get this error:



There are basically 2 Modes of operation for the `FOR. ... NEX.` – command.

Mode 1: Floating Point-Mode:
Imagine you want to count from 0.1 to 0.5 in steps of 0.02. Let's try that:

```
' The 80-bit Floating Point Loop
FOR.$$CNT|0.1|0.5|0.01
  DBP.$$CNT
NEX.
ENR.
```

```
[9] .45
[9] .46
[9] .47
[9] .48
[9] .49
[9] .5
MRE: Robot ended:
```

You see that its easy.

Once the robot finds a "**0.x**" or a variable, anywhere in your FOR. - line, he will choose internally an 80-Bit IEE Floating Point variable with highest precision.
You get up to 16 significant digits.
It has a theoretical range of: +/- 3.4*10^-4932 to 1.2*10^4932.

This happens automatically, you do not need to think about it.

© 2023 Theo Gottwald * https://www.smart-package.com

Mode 2 - 64 bit INTEGER Mode:
In case the robot finds a line like this:

```
FOR.$$CNT|1|13
```

or like this:

```
FOR.$$CNT|1000000000000|1000000000010
```

He will choose the 64-bit Integer mode. In this Mode you have Integers in the range of: **-2^63 to 2^63 -1 that is +/- 64 bit.**

In any of the twi Modes, you can specify a negative step.

```
FOR.$$CNT|13|1|-1

FOR.$$CNT|0.1|0.3|-.001
```

both are completely valid.

Can you jump out of a loop before it has ended?
Yes, you can. Generally jumping out of loops anytime is completely legal.

However this specific Loop will keep the loop-variable an will not reinitialize it, because the loop did not yet end.

Therefore, in case you want to use that specific loop again, you have to add one line of code. It's not more then a simple variable assignment. Do it, then jumping out is completely legal. Here is an example that shows the needed manual loop-variable initialization.

```
:again
: $$CNT=13
FOR.$$CNT|13|1|-1
JMP.out
NXT.
JMP.again
```

This way, you manually reset the Loop-Variable. And that's all to know about jumping around.

How about some speed considerations?

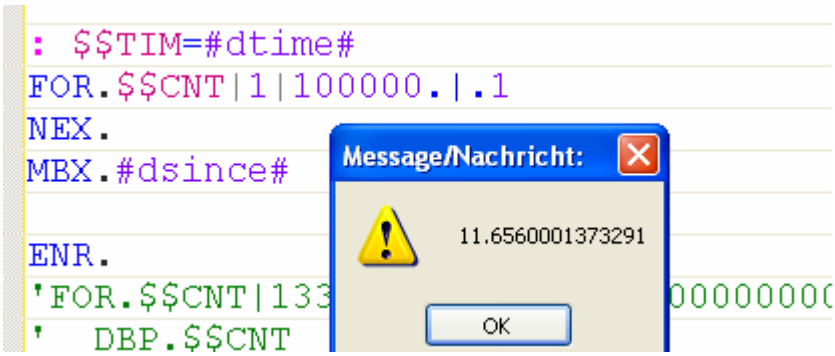
Now lets take a quick look on execution speed. How fast are such loops processed?

Let's try the floating point loop with **1 Million iterations**!
We use this script:

```
: $$TIM=#dtime#
FOR.$$CNT|1|100000.|.1
NEX.
MBX.#dsince#
```

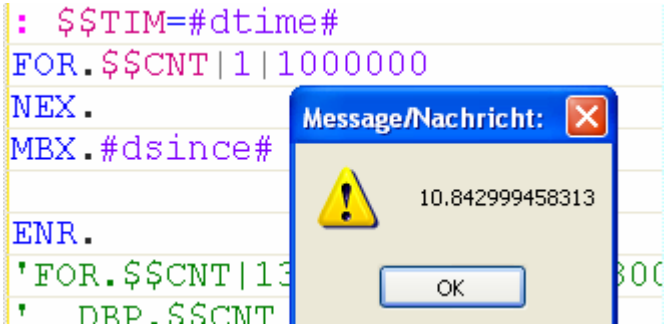

© 2023 Theo Gottwald * https://www.smart-package.com

Of course we need to "compile" the script to get full speed.



It runs in **11.6 seconds!***

Let's see the Integer Mode-Loop.



It needs **10.8 seconds for 1 Million iterations*.** This will show you, that Looping comes to you at a very low price in terms of speed!

*These numbers depend heavily on your PC-system. We give them just as an example.

Syntax

FOR.P1|P2[|P3][|P4]

Parameter Explanation

P1 - counter-variable - VAR
P2 - start-number - NUM
P3 - end-number - NUM
P4 - ***(optional)*** step - NUM

Speed in Ticks:

This command uses typically around 250 Ticks.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1: CTI. Write INI-File
'*****************************************************
'DBM.2
JNF.Move-Console.txt
VAR.§§FIL=?desktop\one.ini
VAR.§§SEC=Section
VAR.§§KEY=Key
VAR.§§VAL=Value
CTI.§§FIL|§§SEC|§§KEY|§§VAL|§§RES
IVV.§§RES=1
PRT. Ini was written with success!
ELS.
PRT. Ini was NOT written with success!
EIF.
' Now we write without check
CTI.§§FIL|§§SEC|§§KEY|§§VAL

' Now we read the file
EXO.§§FIL

MBX. Now we delete this key!
CTI.§§FIL|§§SEC|§§KEY
MBX. Now we delete the whole section!
CTI.§§FIL|§§SEC

END.'*************************************
' Sample 1: FOR. / NEX
'*************************************
FOR.§§AAA|1|100
' Loops from 1 to 100
NEX.

FOR.§§AAA|1|100|2
' Loops from 1 to 100 with increment 2
NEX.

FOR.§§AAA|100|1
' Loops from 100 to 1 step -1 can be omitted
NEX.

: $$TAR=100
FOR.$$AAA|$$TAR|1|-0.5
' Loops from 100 to 1 step -0.5
NEX.

FOR.§§AAA|$$STA|$$TAR|§§STP
' Loops from $$STA to $$TAR step §§STP
NEX.
END.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

FOR. ... NEX. can be nested to unlimited depth.
There is 1:1 paired connection between each FOR ... and the corresponding NEX.

If you do not leave a FOR.-Loop at the NEX.-Command, then the LOOP will stay "active". For example, if you jump out of a FOR.-Loop during the Loop-Run.
If you go into this Loop later, the Robot may be irritated and not reset the Loop-Variable because he thinks "That Loop is still active".
This is normal behvior, because you can also call Subprogrammes or Sub-Skripts from inside a FOR.-Loop.

Therefore in such cases i recommend to add a Initialization to any FOR.-Loop where you want to jump out before the end of the loop. Like this:

```
' Loop Initilization
VAR.$$LOP=1
FOR.$$LOP|1|99
```

Limitations:

If you **jump out** of a FOR...NEX-Loop **before it has ended**, the Loop-Counter Variable will keep the last value in case you call that same loop again.

In this case, the loop will not start, like normal, but just continue Looping from where it left of, when you jumped out.

Therefore, if you jump out of a FOR. ... NEX:-Loop. And if you want to call that same Loop again, you have to manually initialize the Loop-Variable, immediately before the loop.

Do not jump out of a FOR. - NEX. Loop using JMP. or any Jump, unless you make a manual Loop-initialization. Like in the example below.

```
'****************************************
' Sample Jump out of FOR. / NEX
' before End
'****************************************
: §§PAR=Teststring

' The following Line is necessary if you want to
' jump out of the Loop using JMP. or such commands.
' Otherwise the FOR. would not initialize the
' variable the next time it is used.
'
' This is the needed line:
: §§LOP=1
FOR.§§LOP|1|§§EIM
 ' This command will jump to Label "eoa"
 SAO.nrToN|§§PAR|&H028|eoa|§§LOP
 MMV.
NEX.
:eoa
```


© 2023 Theo Gottwald * https://www.smart-package.com

Doing like this, it is completely legal to jump out of as many Loops as you want.

See also:

- 1.6.2. Looping around 126
- 6. Program Flow Control 121
- DOL. - OOP. - Conditional Loop 2299

© 2023 Theo Gottwald * https://www.smart-package.com

3.62.4 NEX. - Bottom of FOR.-Loop

MiniRobotLanguage (MRL)

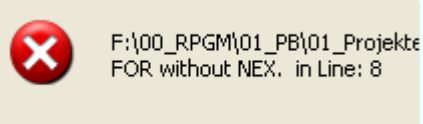
NEX. Command

End of Loop

Intention

Used only together with the FOR. command.

The `FOR. ... NEX.` commands belong together and need to be paired in your script.
For each FOR. there must be a matching `NEX.` If that is not the case, you will get this error:



If you don't know the term "Loop" and have no idea what a "FOR ... NEXT" Loop is, please read the section: 1.6.2. Looping around 126.
This section describes the basics of Loops and the `FOR ... NEX` - Loop.

There are basically 2 Modes of operation for the `FOR. ... NEX.` – command.

Mode 1: Floating Point-Mode:
Imagine you want to count from 0.1 to 0.5 in steps of 0.02. Let's try that:

```
' The 80-bit Floating Point Loop
FOR.$$CNT|0.1|0.5|0.01
  DBP.$$CNT
NEX.
ENR.
```

```
[9] .45
[9] .46
[9] .47
[9] .48
[9] .49
[9] .5
MRE: Robot ended:
```

You see that its easy.

Once the robot finds a "**0.x**" or a variable, anywhere in your FOR. - line, he will choose internally an 80-Bit IEE Floating Point variable with highest precision.

© 2023 Theo Gottwald * https://www.smart-package.com

You get up to 16 significant digits.
It has a theoretical range of: +/- 3.4*10^-4932 to 1.2*10^4932.

This happens automatically, you do not need to think about it.

Mode 2 - 64 bit INTEGER Mode:
In case the robot finds a line like this:

```
FOR.$$CNT|1|13
```

or like this:

```
FOR.$$CNT|1000000000000|1000000000010
```

He will choose the 64-bit Integer mode. In this Mode you have Integers in the range of: **-2^63 to 2^63 -1 that is +/- 64 bit.**

In any of the two Modes, you can specify a negative step.

```
FOR.$$CNT|13|1|-1

FOR.$$CNT|0.1|0.3|-.001
```

both are completely valid.

Can you jump out of a loop before it has ended?
Yes, you can. Generally jumping out of loops anytime is completely legal.

However this specific Loop will keep the loop-variable an will not reinitialize it, because the loop did not yet end.

Therefore, in case you want to use that specific loop again, you have to add one line of code. It's not more then a simple variable assignment. Do it, then jumping out is completely legal. Here is an example that shows the needed manual loop-variable initialization.

```
:again
: $$CNT=13
FOR.$$CNT|13|1|-1
JMP.out
NXT.
JMP.again
```

This way, you manually reset the Loop-Variable. And that's all to know about jumping around.

How about some speed considerations?

Now lets take a quick look on execution speed. How fast are such loops processed?

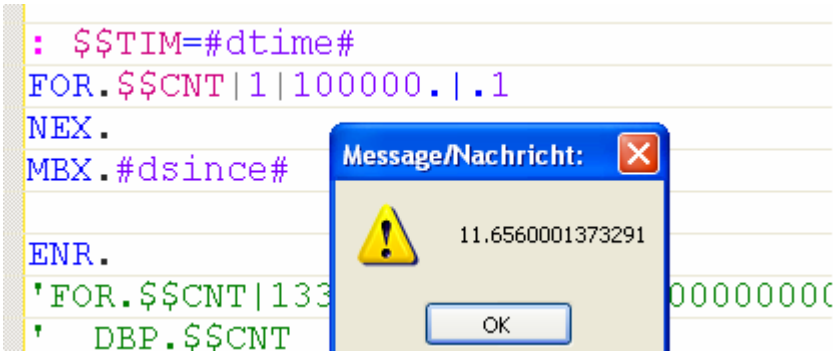
Let's try the floating point loop with **1 Million iterations**!
We use this script:

```
: $$TIM=#dtime#
```


© 2023 Theo Gottwald * https://www.smart-package.com

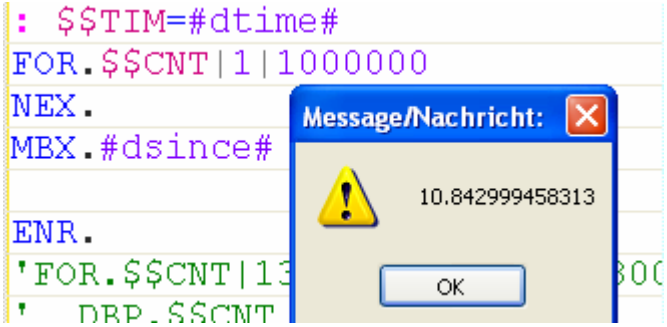
```
FOR.$$CNT|1|100000.|.1
NEX.
MBX.#dsince#
```

Of course we need to "compile" the script to get full speed.



It runs in **11.6 seconds!***

Let's see the Integer Mode-Loop.



It needs **10.8 seconds for 1 Million iterations*.** This will show you, that Looping comes to you at a very low price in terms of speed!

*These numbers depend heavily on your PC-system. We give them just as an example.

Syntax

NEX .

Parameter Explanation

Speed in Ticks:

This command uses typically around 5 Ticks.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
: $$TIM=#dtime#
FOR.$$CNT|1|1000000
NEX.
MBX.#dsince#
```

Remarks

See the suggested Links below to get more information on `FOR .. NEX.` and Loops.

Limitations:

-

See also:

- 1.6.2. Looping around 126
- 6. Program Flow Control 121
- DOL. - OOP. - Conditional Loop 2299
- FOR. - NEX. Counting Loop 2302

© 2023 Theo Gottwald * https://www.smart-package.com

3.62.5 OOP. - Bottom of DOL.-Loop

MiniRobotLanguage (MRL)

OOP. Command

Bottom of DOL.-Loop

Intention

The OOP. is a Loop-Comand. It can construct all sorts of Loops, depending if you add a conditional parameter or not..
Used in conjunction with the `DOL.` command. For Details see here:
DOL. - OOP. - Conditional Loop 2299

Syntax

```
OOP.[P1]
```

Parameter Explanation

Speed in Ticks:
This command uses typically around 20 Ticks.

Example

```
See examples in the DOL Command
```

Remarks

-

Limitations:

-

See also:

- DOL. - OOP. - Conditional Loop 2299

 *© 2023 Theo Gottwald * https://www.smart-package.com*

- 6. Program Flow Control 121
- 1.6.2. Looping around 126
- FOR. - NEX. Counting Loop 2302
- 1.5. Features and Hints 80

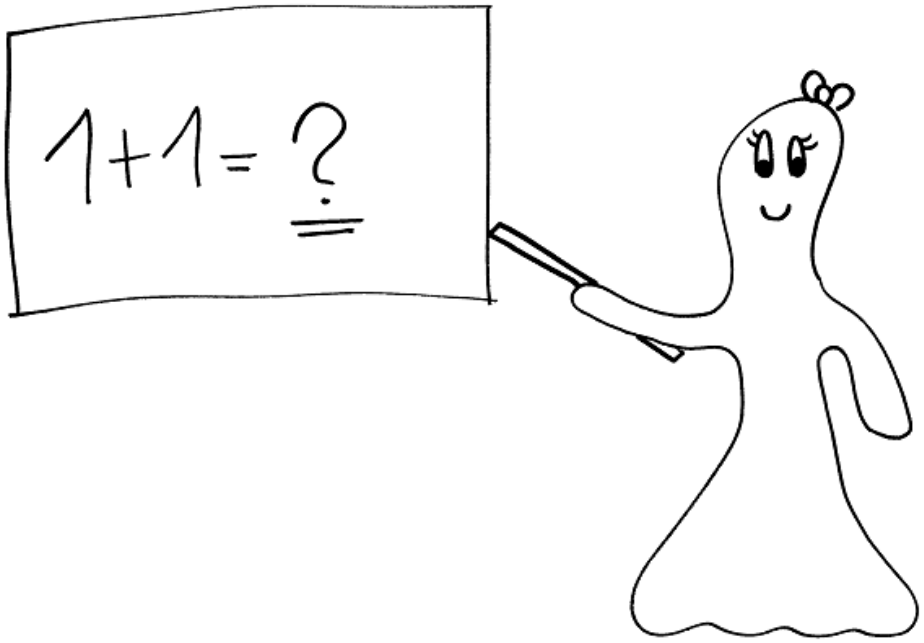
© 2023 Theo Gottwald * https://www.smart-package.com

3.63 Mathematical and algorithmic operations

3.63.1 ! Smarty's mathematical commands

Smarty's mathematical commands Previous 2312 Top 41 Next 2315

Smarty's mathematical and algorithmic operations



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.63.2 CAL. - mathematical CALculation

MiniRobotLanguage (MRL)

CAL. Command

mathematical CALculation

Intention

Solve mathematical calculations. With the `CAL.` Command, you can assign a Variable the result of a mathematical Calculation. And it can help you to prepare complicated decisions. Here is a simple example:

```
CAL.$$RES=3+9
MBX. The result of 3+9 is: $$RES
```

The `CAL.` command knows how to solve (..). You can also do this:

```
CAL.$$RES=3+(9*5)+(8+3/3)
MBX. The result is: $$RES
```

What is the result of this last calculation?
Is it 57 or is it 51.66666 ?

The correct answer is 57. Because calculations with * or / have precedence before + or -.
In fact this is calculated:

```
CAL.$$RES=3+(9*5)+(8+(3/3))
```

And that is the reason, why the correct result is 57. As you can see CAL. knows that. It will calculate mathematically correct.Here are some more advantages of CAL.

- You can use as complicated formulas as you want.
- you can use as many nested brackets as you like.
- get extended precision in all calculation
- include decisions into your calculations
- custom rounding of results
- very fast (~10000 calculations per second)

`CAL` includes a mechanism to embed decisions into the formula. Here is an example:

```
: $$MAA=12
: $$MAB=9
' Calculate an unlimited complicated decision into just one
result:
CAL.$$RES=($$MAA>9)+($$MAB<5)
' Here you get an result in $$RES that is 0,1 or 2
' Now you can make a decision:
IVV.$$RES=0
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
     ' Do something
ELS.
     ' Do something else
EIF.
```

CAL uses "Operator Precedence" and understands negative or positive signed numbers.

Syntax

CAL.P1=[N/V][Operator][...][|P2]

Parameter Explanation

[N/V] = Numerical Value or variable containing a numerical value

[Operator] = **+ - * / \ ^ %**

+ Addition
- Subtraction
* Multiplication
/ Division
\ Integer-Division
% is the "Modulo Operator"
^ power

The result of these numeric operations is a always a Floating-Point Number.

The following Operators enable you to **include decisions** into the formula.
! unequal -Operator
= Equal-Operator
< smaller then
> greater then

The Result of these logical operation is always 0 or 1.
Unlike elsewhere, the Operators do NOT work binary, but on decision true (1) or false (0). See Examples.
P2 - optional Options. Using these Options, you can round or truncate the result of the calculation to an integer. Using this Options, you can use the CAL-Command to round one variable into another one.

CAL uses "Operator Precedence" as follows:

1. The logical Decisions: =!<> are calculated first with equal priority.
2. The ^ Operator is been calculated before any other numerical calculation.
3. The "*/\%" Operators are calculated with equal precedence each after the other next.

© 2023 Theo Gottwald * https://www.smart-package.com

4. Finally the "+-" are calculated.

Custom rounding of results:

CAL. without any Option specified uses 80 bit Extended Precision for all Calculations.
Using the Options, you can manipulate the result of the operation:

```
- i (INTEGER)
- c (CEIL)
- x (FIXPOINT)
- f (FRACTIONAL)
- a (ABSOLUTE)
- r (ROUND) - makes a rounding to 2 decimal places (money ...)
- s (SIGN)
- t (SQR - Square Root)
- u (SIN)
- v (COS)
- w (LOG) - in this special case, the function checks
internally, if the result is not zero. Only then the LOG will
be calculated. Otherwise the result is zero.
```

In Detail:
- "**r**" Rounding is done according to the "banker's rounding" principle: if the fractional digit being rounded off is exactly five, with no trailing digits, the number is rounded to the nearest even number. This provides better results, on average, than the simple "round up at five" approach.
- "**x**" FIX strips off the fractional part of its argument, and returns the integer part. Unlike INT, FIX does not perform any form of rounding or scaling.
- "**i**" INT rounds numeric_expression to the largest integer-class value that is less than or equal to result.
- "**c**" The CEIL function rounds upward, returning the smallest integer value that is greater than or equal to result. For example, CEIL with 1.5 would places the value 2 into Result.
- "**f**" FRAC returns the number after the decimal point of a floating-point number or expression. FRAC rounds the result of fit the precision of the target h, as per IEEE specifications.
- "**a**" The absolute value of a number is its non-negative value. For example, the absolute value of -3 is 3, and the absolute value of +3 is also 3. The absolute value of 0 is 0.
- "**s**" If numeric_expression is positive, SGN returns 1. If numeric_expression is zero, SGN returns 0. If numeric_expression is negative, SGN returns -1.

Speed in Ticks:
This depends heavily on teh used Formula. A rather simple Formula with additional rounding to Integer uses around 260 Ticks.

Example

```
Eval-Example:
' The variable contains a formula now.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$FOM=3*5+9
CAL.$$RES=$$FOM

VAR.$$BBB=0.5
CAL.$$AAA=123.5*$$BBB+0.9|iar

'Example 2:
' Please note that the second part ($$BBB+.5)*($$AAA=1)
' evaluates to zero, when $$AAA=0 because of the use of
' the "=" Operator.
VAR.$$BBB=-5
CAL.$$AAA=(3^-19)+($$BBB+.5)*($$AAA=1)
PRT.L-RESULT:$$AAA

CAL.(4^19)+($$BBB+.5)|i
POP.$$AAA
PRT.Q-RESULT:$$AAA

CAL.(4^19)+($$BBB+.5)
POP.$$AAA
PRT.RESULT:$$AAA

CAL.(4^19)+($$BBB+.5)|r
POP.$$AAA
PRT.RESULT:$$AAA
DMP.

END.
```

Remarks

Depending on complexity of the formula, the CAL.-Command can calculate several thousands of calculations each seconds. The typical speed is between 4000 to 10000 formulas per second on a 3.00 GHz machine.

Limitations:

Does it have limitations? Lets see ...
Doing this:

```
CAL.$$RES=3+(9*5)+((8+3)/3)
```

we get this result:

```
51.6666666666667
00.1234567890ABC
```

We have extended precision. As you can see, we get 12 significant numbers after the "." and a last one that is rounded. My other scientific calculator just gives me 7.

CAL. calculates Formulas of unlimited complexity, there is **no** Limitation in:

© 2023 Theo Gottwald * https://www.smart-package.com

- total Length of the Formula
- How many () are used and nested.
other then the normal 2 GB Limit for total variable storage.

`CAL` works internally with Extended precision (FPU-Registers). They have a range of approximately **+/- 3.4x10^-4932 to 1.2x10^4932**, and offer between a total of 16-18 digits of precision.

See also:

- **3.5 Hexadezimal, Decimal and Binary and more** 352
- **CAX. - Calculate Extended** 2320
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- RND. - RandDom Number 2348
- VIC. - Variable Increment 2352
- VDC. - Variable Decrement 2350

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.63.3 CAX. - Calculate Extended

MiniRobotLanguage (MRL)

CAX. Command

extended scientific CALculations

Intention

Solve scientific mathematical calculations. With the `CAX.` Command, you can assign a Variable the result of a mathematical Calculation. Unlike `CAL.`, `CAX.` is a full featured scientific calculator.

Here are some of the features:

- Using extended precision, 16 to 18 digits of precision is obtained.
- get extended precision in all calculation
- custom rounding of results
- fast (~500 calculations per second)
- in case of errors, Error-Messages on TOS 104
- Large set of trigonometric functions
- Large set of scientific constants and conversions already built in (see below).
- Input of angles can be in degrees or radians.
- Trigonometric output using the ARC functions will always be in radians.
- You can transform angles to radians and vice versa using the built-in conversions.
- You can use as complicated formulas as you want.
- you can use as many nested brackets as you like.
- finally: Factorials can be calculated up to 1754! without overflow!

Supported Operators and expressions:

`+ - * / \ ^ ! ( ) < = > <> >= <=`

Operator precedence:
The Operator Precedence follows as closely as possible the common standards.
The only exception is the unary negation operator which does not have a special priority
following exponentiation.
Operations inside parentheses always have the highest priority and are always evaluated first. Thus by using parentheses in the right places you can always obtain the operator precedence you need for your calculation.

The following operator precedence is used in the program:

```
1. parentheses ( )
2. unary operators (Factorials (!) )
3. one argument functions
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
4. exponentiation (^) (power expressions)
5. multiplication (*) and floating-point division (/)
6. integer division (\)
7. modulo (MOD)
8. addition (+), subtraction (-)
9. relational operators (<, <=, =, >=, >, <> )
10. NOT, ISFALSE and ISTRUE
11. AND
12. OR and XOR (exclusive OR)
13. EQV (equivalence)
14. IMP (implication)
```

Trigonometric functions:

```
"ARCSINH() - Arc (Inverse) Hyperbolic Sine"
"ARCCOSH() - Arc (Inverse) Hyperbolic Cosine"
"ARCTANH() - Arc (Inverse) Hyperbolic Tangent"
"ARCSECH() - Arc (Inverse) Hyperbolic Secant"
"ARCCSCH() - Arc (Inverse) Hyperbolic Cosecant"
"ARCCOTH() - Arc (Inverse) Hyperbolic Cotangent"
"ARCSIN()  - Arc (Inverse) Sine"
"ARCCOS()  - Arc (Inverse) Cosine"
"ARCSEC()  - Arc (Inverse) Secant"
"ARCCSC()  - Arc (Inverse) Cosecant"
"ARCCOT()  - Arc (Inverse) Cotangent"
"SINH()    - Hyperbolic Sine","TANH() - Hyperbolic Tangent"
"SECH()    - Hyperbolic Secant"
"CSCH()    - Hyperbolic Cosecant"
"COTH()    - Hyperbolic Cotangent"
"SEC()     - Secant"
"CSC()     - Cosecant"
"COT()     - Cotangent"
"SIN()     - Sine","COS() - Cosine"
"TAN()     - Tangent"
"ATN()     - Arc (Inverse) Tangent"
"LN()      - Natural Log (base E)"
"EXP2()    - Exponentiation of 2 or Antilog 2"
"EXP10()   - Exponentiation of 10 or Antilog 10"
"EXP()     - Exponentiation of E or Antilog E"
"SQR()     - Square Root","LOG10() - Common Log (base 10)"
"ABS()     - Absolute Value"
"LOG2()    - Log (base 2)"
"LOG()     - Natural Log (base E)"
```

Logical Operations:

```
"AND"       - Logical AND
"XOR"       - Logical XOR (Exclusive OR)
"OR"        - Logical OR
"MOD"       - MOD (Modulo)
"IMP"       - Logical IMP (Implication)
"EQV"       - Logical EQV (Equivalence)
"NOT"       - NOT
"ISTRUE"    - Logical Truth
"ISFALSE"   - Logical Falsity
"CEIL()"    - Ceiling
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  "FRAC()"    - Fractional Part
  "FIX()"     - Truncate To Integer
  "SGN()"     - Sign
```

Important:

While FALSE is always zero, Logical **TRUTH** is been flagged with **-1.**

Usage example:

```
: $$NUA=36
CAX.$$NUM=SIN($$NUA)+5
DBP.Result is $$NUM in Radians

: $$NUA=36
CAX.$$NUM=SIN($$NUA)+5|#
DBP.Result is $$NUM in Degree
```

Know Constants:

A number of constants and conversions are available, making this math evaluator an advanced scientific calculator.

Let us start with PI and Euler:

```
 "PI#"
3.14159..
 "E#"
2.71828..
```

Known Conversion Constants:

```
 "Foot_TO_meter"                          = "*1/
3.28084"
 "Meter_TO_foot"                          , "*
3.28084"
 "Inch_TO_centimeter"                     , "* 2.54"
 "Centimeter_TO_inch"                     , "*1 /
2.54"
 "Kilometer_TO_mile"                   , "*1 /
1.609344"
 "Mile_TO_kilometer"                      , "*
1.609344"
 "Inch_TO_foot"                           , "*1 /
12"
 "Foot_TO_inch"                           , "* 12"
 "Yard_TO_meter"                       , "*1 /
1.093613"
 "Meter_TO_yard"                          , "*
1.093613"
 "Fathom_TO_meter"                        , "*
1.8288"
 "Meter_TO_fathom"                        , "*1 /
1.8288"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 "Mile_TO_light-year"                          , "*1 /
5880000000000"
 "Light-year_TO_mile"                          , "*
5880000000000"
 "Parsec_TO_light-year"                               , "*
3.261643"
 "Light-year_TO_parsec"                        , "*1 / 3.261643"
 "Square_ft_TO_square_m"                       , "*1 / 10.76391"
 "Square_m_TO_square_ft"                              , "*
10.76391"
 "Square_in_TO_square_cm"                             , "*
6.4516"
 "Square_cm_TO_square_in"                             , "*1 /
6.4516"
 "Hectare_TO_acre"                                    , "*
2.471054"
 "Acre_TO_hectare"                             , "*1 / 2.471054"
 "Pound_TO_kilogram"                           , "*1 / 2.204623"
 "Kilogram_TO_pound"                                  , "*
2.204623"
 "Ton_(metric)_TO_Kilogram"                           , "* 1000"
 "Kilogram_TO_ton_(metric)"                           , "*1 /
1000"
 "Ton_(US)_TO_Kilogram"                               , "*
907.18474"
 "Kilogram_TO_ton_(US)"                        , "*1 /
907.18474"
 "Ton_(UK)_TO_Kilogram"                        , "* 1016.046909"
 "Kilogram_TO_ton_(UK)"                        , "*1 /
1016.046909"
 "Ounce_(avoirdupois)_TO_gram"                        , "*
28.349551"
 "Gram_TO_ounce_(avoirdupois)"                 , "*1 /
28.349551"
 "Ounce_(troy)_TO_gram"                               , "*
31.103508"
 "Gram_TO_ounce_(troy)"                        , "*1 /
31.103508"
 "Fahrenheit_TO_Celsius"                      , "*1 / 1.8 - 32 /
1.8"
 "Celsius_TO_Fahrenheit"                              , "* 1.8 +
32"
 "Celsius_TO_Kelvin"                                  , "*1 +
273.16"
 "Kelvin_TO_Celsius"                                  , "*1 -
273.16"
 "Gallon_(US_dry)_TO_liter"                           , "*
4.404884"
 "Liter_TO_gallon_(US_dry)"                   , "*1 / 4.404884"
 "Gallon_(US_liquid)_TO_liter"                        , "*
3.785412"
 "Liter_TO_gallon_(US_liquid)"                , "*1 / 3.785412"
 "Quart_(US_dry)_TO_gallon_(US_dry)"                  , "*1 / 4"
 "Gallon_(US_dry)_TO_quart_(US_dry)"                  , "* 4"
 "Pint_TO_liter_(US_dry)"                             , "*
0.55061"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 "Liter_TO_pint_(US_dry)"                    , "*1 /
0.55061"
 "Pint_TO_liter_(US_liquid)"                   , "*
0.473176"
 "Liter_TO_pint_(US_liquid)"                 , "*1 /
0.473176"
 "Cubic_ft_TO_cubic_m"                       , "*1 /
35.314667"
 "Cubic_m_TO_cubic_ft"                         , "*
35.314667"
 "Horsepower_(elec.)_TO_watt"                  , "*
745.7"
 "Watt_TO_horsepower_(elec.)"                  , "*1 /
745.7"
 "Horsepower_(metric)_TO_watt"                 , "*
735.499"
 "Watt_TO_horsepower_(metric)"                , "*1 /
735.499"
 "BTU/hour_TO_watt"                            , "*
0.293071"
 "Watt_TO_BTU/hour"                           , "*1 /
0.293071"
 "Kilowatt_TO_watt"                            , "* 1000"
 "Watt_TO_kilowatt"                            , "*1 /
1000"
 "Kph_TO_mph"                                 , "*1 /
1.609344"
 "Mph_TO_kph"                                  , "*
1.609344"
 "Knot_TO_mph"                                 , "*
1.150778"
 "Mph_TO_knot"                                , "*1 /
1.150778"
 "Meter/sec_TO_ft/sec"                         , "*
3.28084"
 "Ft/sec_TO_meter/sec"                        , "*1 /
3.28084"
 "Second_TO_minute"                            , "*1 /
60"
 "Minute_TO_second"                            , "* 60"
 "Minute_TO_hour"                              , "*1 /
60"
 "Hour_TO_minute"                              , "* 60"
 "Minute_TO_day"                               , "*1 /
1440"
 "Day_TO_minute"                               , "* 1440"
 "Day_TO_hour"                                 , "* 24"
 "Hour_TO_day"                                 , "*1 /
24"
 "Psi_TO_pascal"                               , "*
6894.757"
 "Pascal_TO_psi"                             , "*1 /
6894.757"
 "Psi_TO_atmosphere"                           , "*
0.068046"
 "Atmosphere_TO_psi"                         , "*1 /
0.068046"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 "Psi_TO_kg/sqcm"                                , "*
0.070307"
 "Kg/sqcm_TO_psi"                               , "*1 /
0.070307"
 "Degrees_TO_radians"                    , "*
1.74532925199432958E-2"
 "Radians_TO_degrees"                , "*1 /
1.74532925199432958E-2"
```

Usage example:

```
GUT.How many hours?|||$$TXT
CAX.$$NUM=$$TXT Hour_TO_minute
DBP.These are $$NUM Minutes.
```

More known Conversions:

```
 "Acceleration_of_gravity_(m/sec²)"                ,
"9.80665"
 "Dry_air_density_at_STP_(kg/m³)"                  , "1.293"
 "Light_speed_in_a_vacuum_(m/sec)"                 ,
"299792458"
 "Solar_constant_(watts/m²)"                       , "1340"
 "Sound_speed_at_STP_(m/sec)"                      , "331.4"
 "Standard_atmosphere_(nt/m²)"                     , "101300"
 "Earth_equatorial_radius_(m)"                     ,
"6378000"
 "Earth_escape_velcity_(m/sec)"                    , "11186"
 "Earth_magnetic_dipole_moment_(amp-m²)"           ,
"6.400E21"
 "Earth_mass_(kg)"                                 ,
"5.983E24"
 "Earth_mean_angular_rotational_speed_(rads/sec)"  ,
".0000729"
 "Earth_mean_density_(kg/m³)"                      , "5522"
 "Earth_mean_orbital_speed_(m/sec)"                , "29770"
 "Earth_polar_radius_(m)"                          ,
"6357000"
 "Earth_volume_(m³)"                               ,
"1.087E21"
 "Alpha_particle_mass_(kg)"                       ,
"6.64465598E-27"
 "Electron_charge-mass_ratio_(coul/kg)"           ,
"175881961000"
 "Electron_rest_mass_(kg)"                        ,
"9.10938188E-31"
 "Muon_rest_mass_(kg)"                            ,
"1.88353109E-28"
 "Neutron_rest_mass_(kg)"                         ,
"1.67492716E-27"
 "Proton_rest_mass_(kg)"                          ,
"1.67262158E-27"
 "Atomic_mass_constant_(kg)"                      ,
"1.66053873E-27"
 "Avogadro_constant_(1/mole)"                     ,
"6.02214199E23"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 "Boltzmann_constant_(joule/K)"                  ,
"1.3806503E-23"
 "Elementary_charge_(coul)"                     ,
"1.602176462E-19"
 "Faraday_constant_(coul/mole)"                 , "96485.3415"
 "Gravitational_constant_(nt-m²/kg²)"           ,
".000000000066726"
 "Permeability_constant_(henry/m)"              ,
"1.25663706143E-6"
 "Permittivity_constant_(farad/m)"             ,
"8.85418781762E-12"
 "Planck_constant_(joule-sec)"                  , "6.62606876E-
34"
 "Rydberg_constant_(1/m)"                       ,
"10973731.568549"
 "Stefan-Boltzmann_constant_(watt/m²K4)"        ,
".000000056704"
 "Universal_gas_constant_(joule/K-mole)"        , "8.314472"
 "Wien_displacement_constant_(m-K)"             , ".00290"
```

Input of angles can be in degrees or radians, depending if you specify the "#" as an additional flag. However, trigonometric output using the ARC functions will always be in radians. You can transform angles to radians and vice versa.

Here is a simple example:

```
CAX.$$RES=3+9
MBX. The result of 3+9 is: $$RES
```

In case of errors, the result will be zero and you will find an error-message on the TOS[104]. Also the Timeout-Flag will be set to show you there was an error. Here is an example:

```
CAX.$$NUM=SIN((9)
DBP.Result is $$NUM

ITO.
  DBP.There was an Error:$$000
EIF.
ENR.
```

```
Stackpos: 000 ->String      :   Error in Expression: Parentheses mismatch
--------------------------------
```

Important:
Expressions and functions should be entered strictly according to standard algebraic syntax rules.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

CAX.P1=[N/V][Operator][...][|P2]

Parameter Explanation

[N/V] = Numerical Value or variable containing a numerical value

[Operator] = **+ - * / \ ^ %**

```
+ Addition
- Subtraction
* Multiplication
/ Division
\ is the "Modulo Operator"
^ power
! Factorial
```

The result of these numeric operations is a always a Floating-Point Number.

The following Operators enable you to **include decisions** into the formula.

```
<>  unequal -Operator
=   Equal-Operator
<   smaller then
>   greater then
<=  smaller then
>=  greater then
```

The Result of these logical operation is always 0 or -1.

Logical Operations:

```
"AND"       - Logical AND
"XOR"       - Logical XOR (Exclusive OR)
"OR"        - Logical OR
"MOD"       - MOD (Modulo)
"IMP"       - Logical IMP (Implication)
"EQV"       - Logical EQV (Equivalence)
"NOT"       - NOT
"ISTRUE"    - Logical Truth, anything else
              then 0 will result in -1
"ISFALSE"   - Logical Falsity, only 0 will result in -1
"CEIL()"    - Ceiling
"FRAC()"    - Fractional Part
"FIX()"     - Truncate To Integer
"SGN()"     - Sign
```

Important:

While **FALSE** is always **zero**, Logical **TRUTH** is been flagged with **-1.** This is different to the CAL.-command which flags truth with 1. Example:

© 2023 Theo Gottwald * https://www.smart-package.com

```
CAX.$$NUM=(5>1)
DBP.Result is $$NUM
' Result is -1

CAL.$$NUM=(5>1)
DBP.Result is $$NUM
' Result is 1
```

P2 - optional Rounding Options. Using these Options, you can round or truncate the result of the calculation to an integer in many way.

Custom rounding of results:

`CAX.` without any Option specified uses 80 bit Extended Precision for all Calculations.
Using the Options, you can manipulate the result of the operation:

```
- i (INTEGER)
- c (CEIL)
- x (FIXPOINT)
- f (FRACTIONAL)
- a (ABSOLUTE)
- r (ROUND) - makes a rounding to 2 decimal places (money ...)
- s (SIGN)
- # (Radian to Degree switch), if omitted Radian is default
```

In Detail:
- "**r**" Rounding is done according to the "banker's rounding" principle: if the fractional digit being rounded off is exactly five, with no trailing digits, the number is rounded to the nearest even number. This provides better results, on average, than the simple "round up at five" approach.
- "**x**" FIX strips off the fractional part of its argument, and returns the integer part. Unlike INT, FIX does not perform any form of rounding or scaling.
- "**i**" INT rounds numeric_expression to the largest integer-class value that is less than or equal to result.
- "**c**" The CEIL function rounds upward, returning the smallest integer value that is greater than or equal to result. For example, CEIL with 1.5 would places the value 2 into Result.
- "**f**" FRAC returns the number after the decimal point of a floating-point number or expression. FRAC rounds the result of fit the precision of the target h, as per IEEE specifications.
- "**a**" The absolute value of a number is its non-negative value. For example, the absolute value of -3 is 3, and the absolute value of +3 is also 3. The absolute value of 0 is 0.
- "**s**" If numeric_expression is positive, SGN returns 1. If numeric_expression is zero, SGN returns 0. If numeric_expression is negative, SGN returns -1.

Example

```
'#####################################
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' CAX. - Examples
'#####################################
'
CAX.$$NUM=PI#|c
DBP.Result is $$NUM
' Result is 4
'
CAX.$$NUM=PI#|i
DBP.Result is $$NUM
' Result is 3
'
CAX.$$NUM=SGN(-0.43)
DBP.Result is $$NUM
' Result is -1
'
CAX.$$NUM=ISFALSE(-0.43)
DBP.Result is $$NUM
' Result is 0

END.
```

Remarks

Depending on complexity of the formula, the CAX.-Command can calculate several hundreds of calculations each seconds. The typical speed is between 100 to 500 formulas per second on a *3.00 GHz* machine.

Note that the MOD Operator (`\,%`) is different, and the result of logical operations are different (-1/1) in `CAX.` and `CAL`. Therefore be careful when switching formulas over.

Limitations:

`CAX` works internally with Extended precision (FPU-Registers). They have a range of approximately **+/- 3.4x10^-4932 to 1.2x10^4932**, and offer between a total of 16-18 digits of precision.

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- CAL. - mathematical CALculation 2315
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- RND. - RandDom Number 2348
- VIC. - Variable Increment 2352
- VDC. - Variable Decrement 2350

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.4 CST. - Coordinate System Transform

MiniRobotLanguage (MRL)

CST. Command

Coordinate System Transform

Intention

The `CST`-Command can transform Coordinates between different coordinate systems.

Currently we can use three coordinate systems:

```
1. Screen Coordinates
2. Window Coordinates (these include the Non-Client Area)
3. Client Area Coordinates
```

For more details on these coordinate systems see here: 3.6 Using Quadrant-Coordinates 356

Using `CST.` you can transfer Point Coordinates like:

```
120,100
```

as well as Rectangle-Coordinates, like

```
120,100,200,400
```

from any of the three above coordinate systems to any of the three coordinate systems.
Here are some examples:

```
' Use Coordinates from Stack, Store Result on Stack
' Window handle is in $$MYW
' Transform Client -> Window Coordinates
CST.cw||$$MYW

' Use actual window as reference
' Transform Screen -> Client Coordinates
CST.s to c|$$REC

' Use Coordinates from Stack, Store Result on Stack
' Transform Window -> Screen Coordinates
CST.w_to_s

And some more examples:

CST.sc|$$REC|$$MYW

CST.cw|$$REC|$$MYW
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
CST.s to c|$$REC|$$MYW

CST.c_to_w|$$REC|$$MYW

CST.sc|$$REC|$$MYW
```

Syntax

CST.P1[|P2][|P3]

Parameter Explanation

P1 **-** Prototype that describes which transformation to do: The following are available:

```
"s to w","sw","s_to_w" - Screen Coordinates to Window coordinates
"s to c","sc","s_to_c" - Screen Coordinates to Client coordinates
"c to s","cs","c_to_s" - Client Coordinates to Screen coordinates
"c to w","cw","c_to_w" - Client Coordinates to window coordinates
"w to c","wc","w_to_c" - Window Coordinates to Client coordinates
"w to s","ws","w_to_s" - Window Coordinates to Screen coordinates
```

Any of the three possible prototypes for each transformation is valid, you can choose how much you want to type.

P2 **-** ***(optional)*** Variable that contains whether Point- or Rectangle Coordinates. They have to be in one of these formats:.

a) `x1,y1`

b) `x1,y1,x2,y2`

If you omit **P2**, the command will try to get the coordinates from the TOS.

P3 **-** ***(optional)*** Variable that contains the handle of the window, that should be used for the transformation. If you omit **P3** then the actual window-handle will be taken. If there is no valid actual window, the desktop-window (Screen coordinates) will be used.

Example

```
' Use the Testapplication
' for this Sample Script
STW.ct|MyClassName|Sample Application
HTV.$$MYW
SCW.nct|1|Edit|{&NOTEXT:}
MMV.
GAR.$$REC
DBP.$$REC
' Convert Screen->Client
CST.sc|$$REC|$$MYW
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Client->Window
CST.cw|$$REC|$$MYW
DBP.$$REC
' Window->Client
CST.wc|$$REC|$$MYW
' Client -> Screen
CST.cs|$$REC|$$MYW
DBP.$$REC
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.6 Using Quadrant-Coordinates 356
- 1.8. Selecting the Application (Topwindow) 150
- 1.9. Locating Child-Windows 162
- ! SCW - Search-Child-Window 3738
- !. STW. - Description 3814
- GAR. - Get actual Rectangle 3619
- GRM. - Get Rect Middle-Point 2337
- MAP. - Move Active Point 2483

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.5 FZC. - Fuzzy Compare

MiniRobotLanguage (MRL)

FZC. Command

Fuzzy Compare

Intention

Compare two strings with one of two alternative fuzzy logic algorithms.

`FZC.` in Mode 1 uses a percentage from 0 to 100.
`FZC.` in Mode 2 (use P3, "m"-Flag) calculates a error number starting from 0 if two strings are equal.

In both modes, you will find the internally calculated "Match-value" on TOS.
The firs example will evaluate to 95% and therefore leave a 95 on the TOS.

```
$$TXB=ABCDEFGHIJK
$$TXA=ABCEFGHIJK

FZC.$$TXA|$$TXB
DBP.$$000
ENR.
```

In Mode 2 we deal with "typing errors". Take this example:

```
$$TXB=ABCDEFGHIJK
$$TXA=ABCEFGHIJK

FZC.$$TXA|$$TXB|m
DBP.$$000
ENR.
```

Will drop a "`1`" on TOS. It indicates that if you type one character at the right place, these strings could be equal. Note that a missing or deleted char is always one typing error.

If you want to base a decision on the result of the comparison, prefer
IFC. - If Fuzzy Compare 2340 .

Syntax

FZC.P1|P2[|P3]

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** Variable A to compare

P2 **-** Variable B to compare

P4 **-** ***(optional)*** can be omitted or can be "m". In that case another algorithm is used to compare **P1** and **P2**.

In detail the difference is:

- Algorithm 1 (no "m"-Flag):
  Result on TOS is a percentage number from **0%** to **100%**.
  P1 matches P2 for the given percentage.

- Algorithm 2 ("m"-Flag, **P4** is used):
  Result is a error number, similar to wrong letter,
  or missing letter. If Algorithm 2 is chosen, then
  P1 matches P2 taking into account the resulting
  number of "typing errors".

Example

```
'**********************************
' FZC.-Test, Algo 1
'**********************************
$$TXB=ABCDEFGHIJK
$$TXA=ABCEFGHIJK

FZC.$$TXA|$$TXB
DBP.$$000
ENR.

'**********************************
' FZC.-Test, Algo 2
'**********************************
$$TXB=ABCDEFGHIJK
$$TXA=ABCEFGHIJK

FZC.$$TXA|$$TXB|m
DBP.$$000
ENR.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.5. Features and Hints 80
- IFC. - If Fuzzy Compare 2340
- IVP./ NVP. - If-Variable-Pattern 3529
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- IVV. / NVV. - If Variable Value 3534

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.6 GRM. - Get Rect Middle-Point

GRM. - Get Rectangle Middle-Point Previous 2334 Top 41 Next 2340

MiniRobotLanguage (MRL)

GRM. Command

Get Rect Middlepoint

Intention

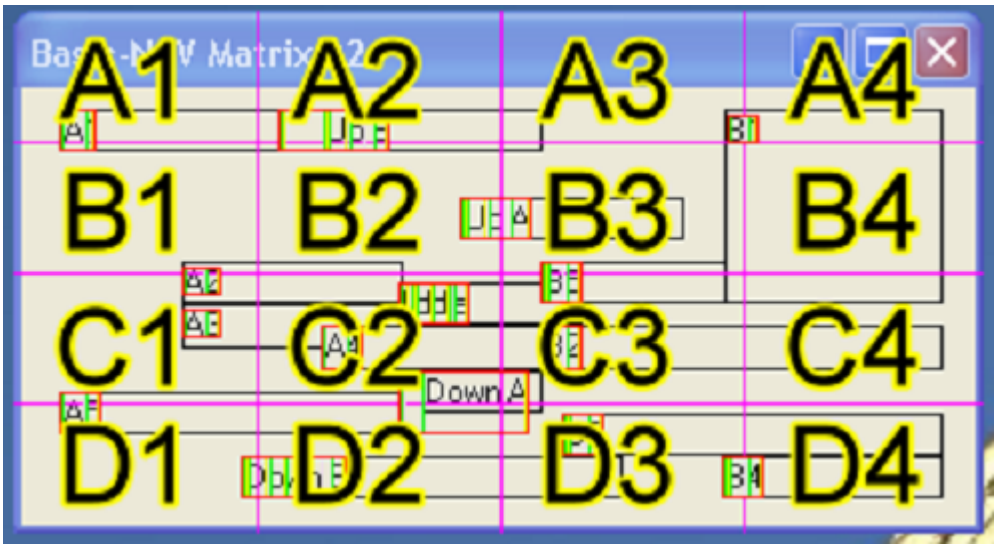
The GRM-Command will simply take any Rectangular Area, and calculate the Middle-Point of it, so you can move the Mouse there.

```
$$REC=120,200,220,300
GRM.$$REC|$$MID
DBP.$$MID
' Result is 170,250
```

GRM. understands Quadrant Coordinates, see 3.6 Using Quadrant-Coordinates 356. It will take the last localized Object as base for the Quadrants. This way, you can split any type of objects or control in an resolution independent way into 16 Quadrants!

Here is an example for the whole window:

```
STW.ct|MyClassName|Sample Application
' We use the actual window-Rect and get the
' Quadrant D1 Middle-Point this way.
GRM.{D1}
' Show TOS
DBP.$$000
ENR.
```



*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

GRM.[P1][|P2]

Parameter Explanation

P1* - *(optional) Input-Rectangle. Format is: `x1,y1,x2,y2` if omitted,
`GRM.` uses the rectangle on TOS, or if this not an Rectangle in the above format,
it will use the actual window rectangle.

`GRM.` understands the 3.6 Using Quadrant-Coordinates 356. Quadrant Coordinates.
It will take the last localized Object as base for the Quadrants.

P2* - *(optional) Variable that will receive the result of the operation in the format:
`x1,y1`
If you omit **P2**, the command will store the coordinates on the TOS.

Example

```
' Use the Testapplication
' for this Sample Script
$$REC=120,200,220,300
GRM.$$REC|$$MID
DBP.$$MID
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.6 Using Quadrant-Coordinates 356
- 1.8. Selecting the Application (Topwindow) 150
- 1.9. Locating Child-Windows 162
- ! SCW - Search-Child-Window 3738

© 2023 Theo Gottwald * https://www.smart-package.com

- !. STW. - Description 3814
- CST. - Coordinate System Transform 2331
- GAR. - Get actual Rectangle 3619
- MAP. - Move Active Point 2483

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.7 IFC. - If Fuzzy Compare

MiniRobotLanguage (MRL)

IFC. Conditional Statement

If Fuzzy Compare

Intention

Conditional Statement.
Test if a specified string evaluates true with a given other String. having regard to one of two alternative fuzzy logic algorithm.

`IFC.` in Mode 1 uses a percentage from 0 to 100.
`IFC.` in Mode 2 (use P4, "m"-Flag) uses a maximum error number starting from 0.

In both modes, you will find the internally calculated "Match-value" on TOS.

```
' Example that will NOT match
' change the percentaqe value so that it will match.
'
$$TXB=ABCDEFGHIJK
$$TXA=ABCDEFGH

IFC.$$TXA|$$TXB|90
  DBV.Yes, we match.
ELS.
  DBP. This matches only to $$000 percent.
EIF.
ENR.
```

In Mode 2 we deal with "typing errors". Take this example:

```
$$TXB=ABCDEFGHIJK
$$TXA=ABCEFGHK

IFC.$$TXA|$$TXB|2|m
  DBV.Yes, we match.
ELS.
  DBP. This matches only to $$000 errors.
EIF.
ENR.
```

Note that a missing or deleted char is always one typing error.
Using algo 2, you will always find the internally calculated number of typing errors on the TOS.

`NWC.` is the negative Form of this statement.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IFC.P1|P2|P3 … ELS. … EIF.

Parameter Explanation

P1 - Variable A to compare

P2 - Variable B to compare

P3 - numerical argument, depends on used algorithm.

P4 - ***(optional)*** can be omitted or can be "m". In that case another algorithm is used to compare **P1** and **P2**.

In detail the difference is:

- Algorithm 1 (no "m"-Flag):
  P3 is a percentage number from **0%** to **100%**.
  If Algorithm 1 is chosen, then
  P1 must match P2 for the given percentage.

- Algorithm 2 ("m"-Flag, **P4** is used):
  P3 is a maximum error number, similar to wrong letter,
  or missing letter.
  If Algorithm 2 is chosen, then
  P1 must match P2 taking into account the given
  maximum number of "typing errors".

IFC.'s can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'***********************************
' IWC.-Test, Algo 1
'***********************************
$$TXB=ABCDEFGHIJK
$$TXA=ABCDEFGH

IFC.$$TXA|$$TXB|90
  DBV.Yes, we match.
ELS.
  DBP. This matches only to $$000 percent.
EIF.
ENR.

'***********************************
' IWC.-Test, Algo 1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' Will match to 90%
$$TXB=ABCDEFGHIJK
$$TXA=ABCDEFGHI

IFC.$$TXA|$$TXB|90
  DBV.Yes, we match.
EIF.


'**********************************
' IWC.-Test, Algo 2
'**********************************
'
' This will NOT match
$$TXB=ABCDEFGHIJK
$$TXA=ABCEFGHK

IFC.$$TXA|$$TXB|2|m
  DBV.Yes, we match.
ELS.
  DBP. This matches only to $$000 errors.
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- 1.5. Features and Hints 80
- FZC. - Fuzzy Compare 2340
- IVP./ NVP. - If-Variable-Pattern 3529
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- IVV. / NVV. - If Variable Value 3534

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.8 IVN. - If Variable Numeric

MiniRobotLanguage (MRL)

IVN./NVN. Conditional Statement

Test if Content of a Variable is numeric or not

Intention

Conditional Statement.
Test if a Variable has numeric content or not. This Conditional Statement has 4 Options.

Mode 1:

`IVN.` in Mode 1 the Parameters get prefixed with a "#".
Doing so, the IVN.uses a Simple character Compare. It will check if the Variable contains only allowed numerical characters, these are: "0123456789".
In this mode IVN. will only accept pure decimal numeric numbers as valid.
Sample:

```
$$TXA=099
IVN.#$$TXA
  MBX.Its numeric!
ELS.
  MBX.Its Not numeric!
EIF.
ENR.
```

Using

```
$$TXA=#099
IVN.$$TXA
```

will not work as the # is part of the Command. Otherwise it would compromise the commands safety.

Mode 2:

`IVN.` in Mode 1 the Parameters get prefixed with a "@".
Doing so, the IVN.uses a Simple character Compare. It will check if the Variable contains only allowed numerical characters, these are: "0123456789.".
In this mode IVN. will only accept pure decimal numeric numbers as valid.
Sample:

```
$$TXA=0.099
IVN.@$$TXA
  MBX.Its numeric!
ELS.
  MBX.Its Not numeric!
EIF.
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Using

```
$$TXA=#1.099
IVN.$$TXA
```

will not work as the @ is part of the Command. Otherwise it would compromise the commands safety.

Mode 3:

`IVN.` in Mode 1 the Parameters get prefixed with a "*".

Doing so, the IVN.uses a Simple character Compare. It will check if the Variable contains only allowed numerical characters, these are: "0123456789." and you can prefix the number with + or −.

In this mode IVN. will only accept pure decimal numeric numbers as valid.

Sample:

```
$$TXA=*-0.99
IVN.*$$TXA
  MBX.Its numeric!
ELS.
  MBX.Its Not numeric!
EIF.
ENR.
```

Using

```
$$TXA=*+19.099
IVN.$$TXA
```

will not work as the * is part of the Command. Otherwise it would compromise the commands safety.

Mode 4: Tolerance Mode - interpret as number.

`IFC.` in Mode 4, you do not use any Prefix. In this Mode the Robot will try to interpret the variable content as a number.

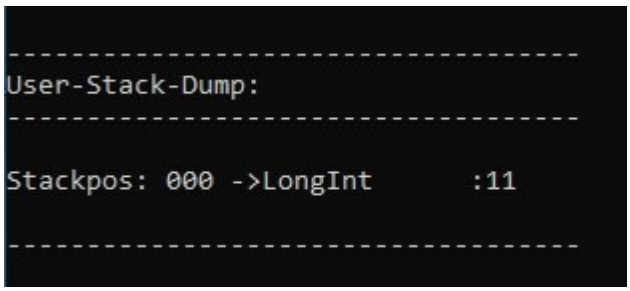
In this mode all sorts of numbers, Floating Point, Hexadecinmal, Octal, Binary ... and even Formulas in (. .) will be recognized.

```
$$TXA=&H09A9
IVN.$$TXA
  MBX.Its numeric!
ELS.
  MBX.Its Not numeric!
EIF.
ENR.


' Floating Point Number
$$TXA=1.1E09
IVN.$$TXA

' Hexadecimal Numbers prefixed with "&H"
$$TXA=&H3A9
IVN.$$TXA
```

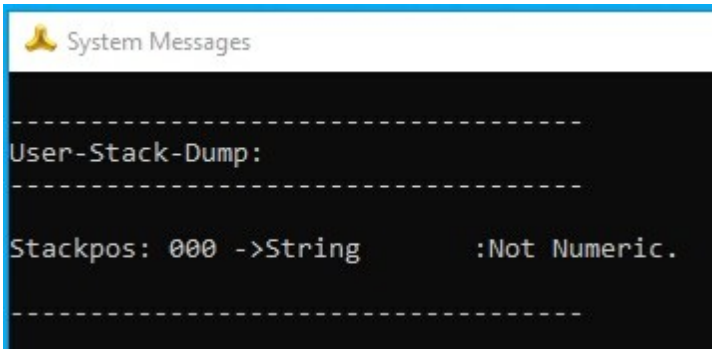

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Octal Numbers prefixed with "&O"
$$TXA=&O371
IVN.$$TXA

' Binary Numbers prefixed with "&B"
$$TXA=&B1011
IVN.$$TXA

' Even Formulas can be used:
$$TXA=(&H9+&O2)
IVN.$$TXA
  MBX.Its numeric!
ELS.
  MBX.Its Not numeric!
EIF.
DMP.6
MBX.$tos$
ENR.
```

In both modes, you will find the internally calculated "Match-value" on TOS.
This is the TOS after the Script above.



You will find the resulting Number on TOS.



If no number was found you will find "Not Numeric" on TOS.

Speed-Consideration:

© 2023 Theo Gottwald * https://www.smart-package.com

```
047 0000000000000045 0001 VAR.$$MSG=3. Teste IVN. "If Variable Numeric @4".
048 0000000000000133 0122 FOR.$$PAA|0|15|.125
049 0000000000000054 0121 IVN.@$$PAA
050 0000000000000148 0121 GSB.Check_Val|$$PAA
051 0000000000000017 0121 ELS.
052 0000000000000000 0000 GTO.Lab_failed
053 0000000000000004 0121 EIF.
054 0000000000000007 0122 NEX.
055 0000000000000000 0001 DMP.Speed
056 0000000000000000 0000 MBX.!
```

IVN. and NVN are really fast and use only 54 to 170 Ticks to execute. Therefore you can execute >250.000 of these commands in one second.

NVN. is the negative Form of this statement.

Syntax

IVN.[#/@/*]P1 … ELS. … EIF.
NVN.[#/@/*]P1 … ELS. … EIF.

Parameter Explanation

\# - ***optional Prefix*** If this Prefix is used then the Content of the Variable will only be checked against "0123456789". The condition will evaluate false if any other characters are inside the variable.
@ - ***optional Prefix*** If this Prefix is used then the Content of the Variable will only be checked against "0123456789.". The condition will evaluate false if any other characters are inside the variable.
* - ***optional Prefix*** If this Prefix is used then the Content of the Variable will only be checked against "0123456789." and Prefix of + and − will be allowed.. The condition will evaluate false if any other characters are inside the variable.

P1 - Variable to check if its numeric.

After the Operation you will find the numeric value on TOS. If no value could be found, you will find "Not Numeric." on TOS.

Please note that a leading "#","@" or "*" inside the Variable will not be interpreted as a Prefix but will lead to the result being "not numerical" as it should be.
IVN.'s can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'***********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' IVN.-Test
'*********************************
FOR.$$PAA|-5|5|.125
  IVN.$$PAA
    GSB.Check_Val|$$PAA
  ELS.
    GTO.Lab_failed
  EIF.
NEX.
ENR.
' Prüft Value auf TOS gegen Angabe
:Check_Val
SAV.Save|$$SEB
VAV.$$SEB=§§_01
POP.$$SEA
PRT.$$SEB-"$$SEA"
JIS.$$SEA!$$SEB|Lab_failed
SAV.Restore
RET.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- 1.5. Features and Hints 80
- FZC. - Fuzzy Compare 2340
- **IVP./ NVP. - If-Variable-Pattern** 3529
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- IVV. / NVV. - If Variable Value 3534

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.9 RND. - RandDom Number

MiniRobotLanguage (MRL)

RND. Command

RaNDom number

Intention

This command can be used to get random numbers within a known range.
Using `RND.` without any parameter will Re-Seed the internal `RND`-Generator.

The internal Random number generator will always return the same sequence of random numbers. If you do not want this, just add a

RND.
at the start of your script. This will re-seed the random number generator and start a new practically unpredictable random number sequence.

Syntax

RND.[P1][|P2][|P3]

Parameter Explanation

P1 **-** `minimum value (can be negative), variable or formula in (..).`

P2 **-** `Maximum Value (can be negative), variable or formula in (..).`

P3 **-** `Variable to return the Random-Value.If omitted Value is been placed on` TOS 104.

Example

```
'************************************
' RND - DEMO
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'************************************
' Re-Seed the RND-numbers generator
' do this to prevent to always get the same sequence of
' random numbers.
RND.
FOR.$$AAA|1|20
' Generate Random Numbers between 1 and 9
  RND.1,9,$$REB
  PRT.[$$AAA]->$$REB|d
NEX.
END.

'************************************
' RND - DEMO 2
'************************************
'DBM.2
FOR.$$AAA|1|20
  RND.-3|-1|$$REB
  PRT.[$$AAA]->$$REB|d
NEX.
END.
```

Remarks:

The Parameter **P1** can be **negative**, for example, if you want Random-Numbers between 5 and 5.

Limitations:

This is not a cryptographic/scientific random number generator.

See also:

- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- 3.1 Systemvariables 334

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.10 VDC. - Variable Decrement

MiniRobotLanguage (MRL)

VDC. Command

Variable DeCrement

Intention

Decrement the Value inside any specified Global or Local Variable or Stack.

"Decrement" means that the number is decreased by 1 or if specified, the number in **P2**.

Syntax

VDC.P1[|P2]

Parameter Explanation

P1 **-** `Variable containing any NUM`

P2 **-** `optional, value that is subtracted from Variable, default is 1.`

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAR.$$NUM=50
VDC.$$NUM
VDC.$$NUM|15
END.
```

Remarks

-

Limitations:

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

See also:

- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.11 VIC. - Variable Increment

VIC. - Variable Increment Previous 2350 Top 41 Next 2354

MiniRobotLanguage (MRL)

VIC. Command

Variable InCrement

Intention

Increment the Value inside any local or Global Variable or Stack.

"Increment" means that the number is increased by 1 or if specified, the number in **P2**.

Syntax

VIC P1[|P2]

Parameter Explanation

P1 **-** `Variable containing any NUM`

P2 **-** `optional, value that is added to P1, default is 1.`

Speed in Ticks:

This command uses typically around 170 Ticks.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAR.$$NUM=50
VIC.$$NUM
VÎC.$$NUM|15
END.
```

Remarks

-

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Limitations:

-

See also:

- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.12 VTB. - Variable to Binary

VTB. - Variable To Binary Previous 2352 Top 41 Next 2356

MiniRobotLanguage (MRL)

VTB. Command

Variable To Binary

Intention

Convert a numeric value into the Binary number-System, optional you can specify the length of the output string.

This very simple example

```
$$NUM=655345
VTB.$$NUM
DBP.$$NUM
```

Will give the result: "1001111111111110001"

In this example we say that we want the output to be formatted to 40 digits.

```
$$NUM=655345
VTB.$$NUM||40
DBP.$$NUM
ENR.
```

Therefore the result is "0000000000000000000010011111111111110001"

In both cases, we did not specify **P2** (Target variable, therefore the result was placed in the source variable.

For details on how to prefix Hexadecimal, Octal or Binary numbers, see
3.5 Hexadezimal, Decimal and Binary and more 352

Syntax

VTB.P1[|P2][|P3]

Parameter Explanation

P1 - Variable containing any NUM

P2 - (optional) Variable for the result, if omitted result is been stored in Source variable.

P3 - (optional) optional, value that is used as a fix length of the output string.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Empty places are filled with "0"'s.
If you specify less digits then needed, the number will be clipped from the right side,

Example:

```
$$NUM=655345
VTB.$$NUM|$$TAR|2
```

will give you "01" as result (instead of 1001111111111110001)

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
' Binary to hexadecimal
$$NUM=&B1001111111111110001
VTH.$$NUM
DBP.$$NUM

' Hexadecimal to decimal (unsigned)
$$NUM=&H0$$NUM
VTN.$$NUM|$$TAR|6
DBP.$$TAR
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.13 VTH. - Variable to Hexadecimal

MiniRobotLanguage (MRL)

VTH. Command

Variable To Hexadecimal

Intention

Convert a numeric value into the Hexadecimal-System, optional you can specify the length of the output string.

This very simple example

```
$$NUM=655345
VTH.$$NUM
DBP.$$NUM
```

Will give the result: `"9FFF1"`

In this example we say that we want the output to be formatted to 15 digits.

```
$$NUM=655345
VTH.$$NUM||15
DBP.$$NUM
ENR.
```

Therefore the result is `"00000000009FFF1"`

In both cases, we did not specify **P2** (Target variable, therefore the result was placed in the source variable.

For details on how to prefix Hexadecimal, Octal or Binary numbers, see
3.5 Hexadezimal, Decimal and Binary and more 352

Syntax

VTH.P1[|P2][|P3]

Parameter Explanation

P1 - Variable containing any NUM

P2 - (optional) Variable for the result, if omitted result is been stored in Source variable.

P3 - (optional) optional, value that is used as a fix length of the output string.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Empty places are filled with "0"'s.
If you specify less digits then needed, the number will be clipped from the right side,

Example:

```
$$NUM=655345
VTH.$$NUM|$$TAR|2
```

will give you "F1" as result (instead of 9FFF1)

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
' Binary to hexadecimal
$$NUM=&B10011111111111110001
VTH.$$NUM
DBP.$$NUM

' Hexadecimal to decimal (unsigned)
$$NUM=&H0$$NUM
VTN.$$NUM|$$TAR|6
DBP.$$TAR
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTB. - Variable to Binary 2354
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

3.63.14 VTN. - Variable Numeric

VTN. - Variable To Numeric (Decimal) **Previous** 2356 **Top** 41 **Next** 2361

MiniRobotLanguage (MRL)

VTN. Command

Variable To Numeric (Dezimal)

Intention

Convert a numeric value into the Decimal number-System, optional you can specify the length of the output string.

This very simple example

```
$$NUM=&B1001111111111110001
VTN.$$NUM
DBP.$$NUM
```

Will give the result: `"655345"`

In this example we say that we want the output to be formatted to 15 digits.

```
$$NUM=&H09FFF1
VTN.$$NUM||15
DBP.$$NUM
ENR.
```

Therefore the result is `"000000000655345"`

In both cases, we did not specify **P2** (Target variable, therefore the result was placed in the source variable.

For details on how to prefix Hexadecimal, Octal or Binary numbers, see
3.5 Hexadezimal, Decimal and Binary and more 352

Syntax

VTN.P1[|P2][|P3]

Parameter Explanation

P1 - Variable containing any NUM

P2 - (optional) Variable for the result, if omitted result is been stored in Source variable.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P3 - (optional) optional, value that is used as a fix length of the output string.
Empty places are filled with "0"'s.
If you specify less digits then needed, the number will be clipped from the right side,

Example:

```
$$NUM=655345
VTN.$$NUM|$$TAR|2
```

will give you "45" as result (instead of `655345`)

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
' Binary to hexadecimal
$$NUM=&B10011111111111110001
VTH.$$NUM
DBP.$$NUM

' Hexadecimal to decimal (unsigned)
$$NUM=&H0$$NUM
VTN.$$NUM|$$TAR|6
DBP.$$TAR
ENR.
```

Remarks

Please always use the proper prefix for non-decimal numbers.
More details on this, see here:
3.5 Hexadezimal, Decimal and Binary and more 352

Limitations:

-

See also:

- 3.5 Hexadezimal, Decimal and Binary and more 352
- VTH. - Variable to Hexadecimal 2356
- VDC. - Variable Decrement 2350
- CAL. - mathematical CALculation 2315
- VAR. - Variable Set Value/Clear 3562
- VAN. - Variable Numeric 3553

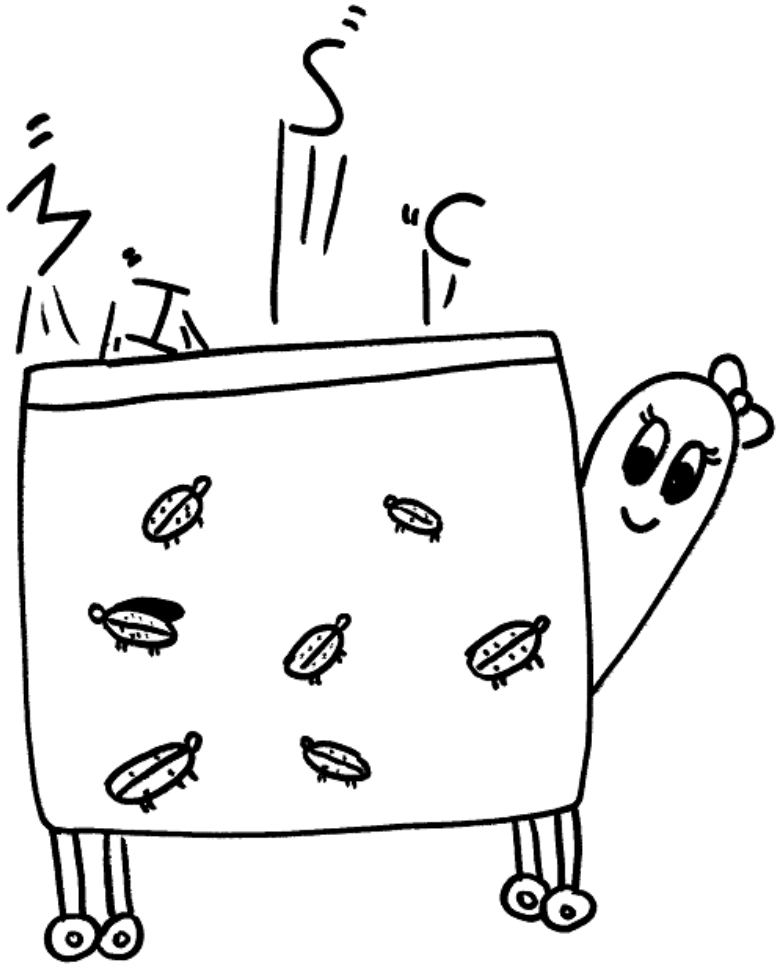
© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.64 Miscellaneous commands

3.64.1 ! Smarty's MISC-Operations

Smarty's MISC-Operations



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.64.2 END. - End Script

MiniRobotLanguage (MRL)

END. Command

Ends a script

Intention

End script execution immediately and optionally set a return-code.

If given a Parameter, and called in "Exemode", End will set a return code for a compiled executable. If given a parameter not encapsuled in an executable, `END.` will set the Returncode like `SEC.` only for the "`MR01.exe`".

Syntax

END.[P1]

Parameter Explanation

P1 – NUM containing the return-value (Error-Code) for the calling application.

Example

```
' Simply ends a Script
END.

' Ends a Script and gives a return-code of 10
' for compiled executables
END.10
```

Remarks

To set a Return code without ending the script, use `SEC.` (Set Exit Code).
If no Return code is given, the return code is zero (0) or the last code that was given to an `SEC.`-command.
Use the `ENR.` (End Return) command if you plan to call the script from another script using `JNF.`

Limitations:

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

See also:

- 1.B Error handling 224
- END. - End Script 2362
- ENR. - End Return 2169
- SRV. - Set Return Value (Executable) 2134
- SEC. - Set Exit Code 2129

© 2023 Theo Gottwald * https://www.smart-package.com

3.64.3 UNI. - Universal Command

MiniRobotLanguage (MRL)

UNI. Command

Name for multiple Subprogrammes that are currently in testing

Intention

You can use these commands but there is no warranty that they will be working in the same way in the future.
They are "subject to Improvement". Some of the Examples below are in "MPR-Skript", they use a "\" instead of the "." so they may need be changed to SPR-Skript in parts.
Many of the commands are already implemented in native SPR-Commands.

Syntax

UNI.subcommand|[Parameters]

Parameter Explanation

Paramenter - the Parameters depend on the used Subcommand

B E T A

GRN - Dokumentation

Das UNI.- Mnemonic ist als Universal-Mnemonic angelegt und eröffnet eine Reihe von fortschrittlichen neuen Möglichkeiten.
Die Hauptaufgabe ist dabei die graphische Erkennung von Objekten in Fenstern, wie dies typischerweise in JAVA-Programmen und JAVA-Setups erforderlich ist.
Darüberhinaus implementiert UNI. auch noch ergänzende Möglichkeiten im Umfeld der Bedienung solcher Programme.

Die Syntax ist generell wie folgt:

UNI.[Command-Name][..|Parameter1][..|Parameter2][..|Parameter3][>Variable]

Dabei sind die Command-Names in unterschiedliche Gruppen eingeteilt:

g_ ... graphische Commands

 *© 2023 Theo Gottwald * https://www.smart-package.com*

s_ ... String (Zeichenketten-Befehle)
d_ ... Befehle, die neue Datenklassen implementieren
m_ ... mathematische und Datums-Befehle

```
/*****************************************************************************************************
*****************************/
```

Beispiel: UNI.s_unwrap

```
'? VAR=Variable
VAR\$$STR=a<end>c
UNI.s_unwrap|$$STR|ac>$$RES
' $$RES wäre nun "<end>" da die "linke Klammer 'a'
'                   und die rechte Klammer 'b"' entfernt wurden.
'? MBX=MessageBoX
MBX\$$RES
```

```
/*****************************************************************************************************
*****************************/
```

```
/*****************************************************************************************************
*****************************/
```

Beispiel: UNI.m_calc

```
UNI.m_calc|13/2|x>$$RES
MBX\$$RES
```

```
/*****************************************************************************************************
*****************************/
```

```
#############################################################################
###############################################
#############################################################################
###############################################
#############################################################################
###############################################
```

```
'*****************************************************************************
Graphical Recognition Commands
'*****************************************************************************
```

Diese Befehle dienen dazu, in einem Setup oder einer anderen Programm-Maske Elemente zu lokalisieren oder auf das Erscheinen dieser Elemente zu warten.
Falls dazu Text verwendet wird, wird dazu in zwei Schritten verfahren:
1. Analyze (Parameter)
2. Locate Object

Alternativ kann auch zusätzlich navigiert werden. Hierbei wird ein Objekt lokalisiert und dann mittels der "g_navigate" Anweisung ein Objekt angefahren.
Dabei wäre die Verfahrensweise dann so:

© 2023 Theo Gottwald * https://www.smart-package.com

1. Analyze
2. Locate
3. Navigate ...

Unabhängig davon, was lokalisiert wurde, stehen die Koordinaten der Kanten des lokalisierten Elementes jeweils auch über die WinRobots-Spezialvariablen zur Verfügung:
il = Linkskoordinate (ItemLeft),
it = Obenkoordinate (ItemTop),
ir = Rechtskoordinate (ItemRight),
ib = Untenkoordinate (ItemBottom),
iw = Breite (ItemWidth),
ih = Höhe (ItemHeight),
(Siehe Hilfe "Spezialordner")

Wird eine Farbdifferenz oder ein Farbkontrast verlangt, ist die übliche Reichweite 0-765.
0 - wäre die genau gleiche Farbe, während die Maximaldifferenz bzw. der Maximalkontrast (also weiss-schwarz) 765 ist.

****
"ana.","g_analyze"
****

Dieser Befehl analysiert die Fensteroberfläche. Dies ist notwendig zur Vorbereitung späterer Suchbefehle.
Der Befehl teilt die Oberfläche des lokalisierten Fensters in Quadranten ein, und ermöglicht die Angabe in welchen der Quadranten gesucht werden soll.
Zudem ermöglicht der Befehl eine graphische Anzeige, welche Objekte mit den späteren Suchbefehlen lokalisiert werden können.

Der Befehl benötigt dazu eine Anzahl Parameter:
' TF:Textfarbe, Defaultwert ist 0 = schwarz - optional
' TL:Toleranzwert, Defaultwert 150
' PD:Pixeldistanz, 4-48, Defaultwert 6 - optional
' WH:Window Handle - optional, Default ist das zuletzt lokalisierte Fenster
' MS:Feldmaske, Default 0 - keine Einschränkung
' FL:Flags 0/1 - Picture-Mode Rest, s.u.
' UNI.g_analyze|[TF:0][,PD:4-48][,TL:150][,WH:32423][,MS:Feldmaske]

Dabei wird die Maske wie folgt in 16 Felder eingeteilt:
A1,A2,A3,A4
B1,B2,B3,B4
C1,C2,C3,C4
D1,D2,D3,D4

Damit kann man angeben, in welchen Feldern später gesucht werden soll.
Ebenso kann man angeben, ob Objekte vollständig oder nur überwiegend in den angegebenen Feldern liegen müssen oder ob es genügt, wenn das Objekt partiell im Feld liegt.
Diese Angaben macht man bei der Vorbereitung der Maske mittels "g_mask".

© 2023 Theo Gottwald * https://www.smart-package.com

Beim Parameter MS: hat man die Wahl, die Felder entweder direkt anzugeben, dazu müssen Klammern wie folgt verwendet werden:
' UNI.g_analyze|MS:(A1,A2,A3,p)

oder man kann eine vordefinierte Maske aus einer Variablen benutzen, die zuvor mittels "g_mask" vorbereitet wurde.
Dieses Verfahren erlaubt es diverse Masken in Variablen vorzuhalten.
VAR\$$ARS=A1,A2
' Feldmaske vorbereiten und in $$TXT ablegen
UNI.g_mask|$$ARS|p>$$TXT
UNI.g_analyze|PD:8,FL:12,TL:50,MS:$$TXT

Beispiele:

' Direktangabe der Maske
' UNI.g_analyze|MS:(A1,A2,A3,p),TL:100

' repeat last analyze run with same parameters
' UNI.g_analyze|R
' or
' UNI.g_analyze|Repeat

' Komplettes Beispiel
' Diese Felder werden durchsucht
VAR\$$ARS=A1,A2
' Feldmaske vorbereiten und in $$TXT ablegen
UNI.g_mask|$$ARS|p>$$TXT

STW\shimgvw:cpreviewwnd\
UNI.g_analyze|PD:8,FL:12,TL:50,MS:$$TXT
UNI.fba.|Summen
MMV\

Flags - Bedeutung:
' 1 - Win7-Mode
' 2 - ToGrey
' 4 - Cleartype Mode
' 8 - Debug Mode, Draw Letterboxes
' 16 - Invert Boxes
' 32 - Draw Fields

****
"msk.","g_mask"
****

Mit diesem Befehl kann man den Suchbereich in der Maske auf bestimmte Felder einschränken.
Dabei wird die Maske/das Suchfenster wie folgt in 16 Felder eingeteilt:
A1,A2,A3,A4
B1,B2,B3,B4
C1,C2,C3,C4

© 2023 Theo Gottwald * https://www.smart-package.com

D1,D2,D3,D4

Man kann nun angeben welche Felder für die spätere Suche initialisiert werden sollen, alle anderen werden ignoriert.
Dazu kommt eine Angabe, ob Objekte vollständig, überwiegend oder nur teilweise in den angegebenen Feldern liegen müssen.
Es genügt dazu die Angabe eines Buchstabens, dieser gilt immer für alle Felder:

p - partiell, f - full, m - much (bedeutet mehr als die Hälfte muss "inside" des Feldes sein.

Für die Spezifikation der Felder gibt es prinzipiell zwei Möglichkeiten. Die additive und die subtraktive.

Die additive Maskenangabe:
Hierbei werden einfach die Felder aufgelistet, die durchsucht werden sollen.

Beispiel:
' gesucht wird in A1 und A2, die Objekte müssen nur "p"artiell in den Feldern enthalten sein.
VAR\$$ARS=A1,A2
UNI.g_mask|$$ARS|p>$$TXT

Die subtraktive Maskenangabe:
Hierbei werden nur die Felder aufgelistet, die NICHT durchsucht werden sollen. ALLE anderen werden durchsucht.
Wird ein Feld als "subtraktiv" angegeben, sind automatisch alle weiteren Angaben auch subtraktiv!

Beispiel:
' gesucht wird in allen Feldern außer A1 und A2, die Objekte müssen vollständig in den Feldern enthalten sein (wegen des "f").
VAR\$$ARS=!A1,A2
UNI.g_mask|$$ARS|f>$$TXT

****
"usp.","g_use_place"
****
Mit diesem Befehl kann ein beliebiger Punkt in der Maske spezifiziert werden, von wo aus dann mit dem "g_navigate" Befehl ein Element angefahren werden kann.
Die Angaben erfolgen in fensterrelativen Koordinaten, die aus dem Windows Analyzer ermittelt werden können.

Beispiel:
' Element "Standardmappe" lokalisieren
UNI.g_use_place|389,292
UNI.g_navigate|RA|1

Beispiel:
VAR\$$ARS=!D1
UNI.g_mask|$$ARS|p>$$TXT
STW\shimgvw:cpreviewwnd\

© 2023 Theo Gottwald * https://www.smart-package.com

UNI.g_analyze|FL:6,PD:6,PM:6,TL:10
UNI.g_use_place|389,292
UNI.g_navigate|RA|1
MMV\

****
"nav.","g_navigate"
****
Dieser Befehl dient dazu, von einem lokalisierten Objekt aus ein danebenliegendes anderes Objekt anzufahren.
Dies ist sinnvoll, um auch solche Objekte anzufahren, die man anders nicht sicher lokalisieren kann.
g_navigate berücksichtigt intern den Abstand zwischen dem aktuell lokalisierten Element und den in Frage kommenden Elementen.
Daher kann man auch noch eine number angeben, die angibt das wievielte Element (rechts, links etc.) man anfahren will.
Dies kann zum Beispiel genutzt werden, um in Tabellen Elemente zu überspringen.

UNI.g_navigate|[Richtung]|[number][|cc-Wert]

Für die Richtung gibt es folgende Parameter:
A - All (jede Richtung)
R - rechts
L - links
U - Oben
D - Unten
Sowie Kombinationen:
LA, RA, UA, DA.

Der Unterschied liegt darin, dass ein Feld bei der Angabe von L,R,U,D innerhalb der jeweiligen Kante liegen muss um berücksichtigt zu werden.
Wird zusätzlich ein "A" angegeben, dann kann das Element "irgendwo" auf der jeweiligen Seite liegen.
In diesem Fall wird nur noch der absolute Abstand der Elemente (Kante zu Kante) berücksichtigt.

Beispiel:
' Das nächstliegende Element rechts vom aktuellen wird angefahren
UNI.g_navigate|RA

' Das viert-nächstliegende Element (A(ll) = Umkreis) wird lokalisiert
UNI.g_navigate|A|4

****
"fba.","g_find_bap"
****
Der Befehl g_find_bap dient dazu, in der Anwendung "Bank ArbeitsPlatz" (Fiducia) Elemente zu lokalisieren.
Die Schriftgröße kann dabei zwischen 7 und 14 Punkte variieren. Es wird direkt der jeweilige Text angegeben, der angefahren werden soll.

*© 2023 Theo Gottwald * https://www.smart-package.com*

UNI.g_find_bap|Suchbegriff|No.|Toleranzzahl für Textvergleich

Die numbernangabe ist dann wichtig, wenn man nicht den ersten passenden, sondern einen anderen passenden Begriff finden will.
Die "Toleranzzahl für Textvergleich" ist per default 2 und je höher (max. 8) desto toleranter ist der Textmustervergleich. Dieser Wert ist unabhängig von der Farbtoleranz, die bei "g_analyze" verwendet wird.

Beispiel:
```
UNI.g_find_bap|Standardmappe
MMV\
```

Beispiel mit Angabe eine alternativen Font-Namens:

```
UNI.fba.|Cancel01|1|4|Courier New
ITO\
  OPR\0,0
  SBT\2|Not found.
ELSE
  MMV\
ETO\

MBX\!
@
```

****
"wba.","g_wait_find_bap"
****
Dieser Befehl dient dazu, auf das Erscheinen eines Suchbegriffes in der Maske zu warten.

UNI.g_wait_find_bap|Suchbegriff|No.|Toleranzzahl für Textvergleich|Maximale Wartezeit in Sekunden

Es kann eine maximale Wartezeit in Sekunden angegeben werden; fehlt diese Angabe wird 30 angenommen.

Beispiel:
```
' Es wird 6 Minuten auf das Erscheinen eines bestimmten Textes gewartet.
UNI.g_wait_find_bap|Übersicht|1|2|360
' bewirkt das Gleiche
UNI.g_wait_find_bap|Übersicht|||360
```

****
"set.","g_settings"
****
Dieser Befehl dient dazu interne Parameter zu setzen. Bei Angabe von:
TE:0 oder TE:1
wird zur Textfindung mittels "fba." oder "ftx." jeweils ein anderes Vergleichsverfahren verwendet.
Es ist denkbar dass zukünftig andere Verfahren dazukommen.
Das Standardverfahren ist derzeit TE:0.

© 2023 Theo Gottwald * https://www.smart-package.com

Dieses Verfahren kann bei Fenstern mit vielen graphischen Mustern fälschlicherweise kleinen Text finden.
TE:1 arbeitet viel exakter, findet evtl. jedoch Text der nicht ganz genau übereinstimmt nicht.

Beispiel:
STW\shimgvw:cpreviewwnd\
UNI.g_analyze|PD:4,FL:1,TL:1,TF:&H0,MS:$$TXT

UNI.set.|TE:1
UNI.fba.|Bestellung
MMV\

UNI.set.|TE:0
UNI.fba.|Belegübersicht ein
MMV\

eine weitere Option ist "RC:". Dieses Setting bestimmt die Auswahl der zu lesenden Zeichen für den "rab." Befehl.
Es stehen folgende Subsets und jede Kombination daraus zur Auswahl:

US-Letters_small: 1
US-Letters_big: 2
german Umlauts: 4
Zahlen: 8
Signs: 16
Kernings: 32

' 39 = Kernings + Umlauts + small Letters + big Letters
UNI.set.|RC:39
UNI.rab.
MMV\

Der defaukt Wert für RC ist 39. Sollten in einem Block zum Beispiel NUR Zahlen gelesen werden, kann man das Setting auf "8" setzen.

Eine weitere Option ist "FA:" - "Font Attributes". man kann hier auswählen ob auch Text der:
1 - Normal
2 - Bold
4 - Kursiv
6 - Bold und Kursiv
7 - Normal, Bold und Kursiv
in die Suche mit einbezogen werden soll. Dies kann Einfkluss auf die Rechenzeit der Lesebefehle haben.

Die Option: "SS:" - "smallest Size" ermöglicht die Angabe einer minimalen Font-Suchgröße.
Intern ist die minimale Suchgröße auf den Wert "6" beschränkt. Kleinere Werte werden nicht berücksichtigt.

© 2023 Theo Gottwald * https://www.smart-package.com

Die Option: "LS:" - "Largest Size" ermöglicht die Angabe einer maximalen Font-Suchgröße.
Intern ist die maximale Font-Suchgröße auf den Wert "22" festgelegt. Größere Werte werden nicht berücksichtigt.

****
"gfa.","g_get_found_size"
****
Dieser Befehl kann erst angewendet werden, nachdem ein Textsuch-Befehl wie afb. oder ftx. verwendet wurde.
Er gibt als Rückgabewert eine interne Schriftgröße zurück, in zwei verschiedenen Skalierungen, unter der der jeweilige Text lokalisiert wurde.
Zudem eine dritte Zahl zurückgegeben, dabei handelt es sich um einen Bold-Wert.

```
VAR\$$ARS=!A1

UNI.g_mask|$$ARS|p>$$TXT
STW\shimgvw:cpreviewwnd\
'GCW\\1\shimgvw:czoomwnd\*|w74994
UNI.g_analyze|PD:6,FL:2,TL:10,TF:0,MS:$$TXT
UNI.fba.|Abbrechen||1
NTO\
  UNI.gfa.>$$TXT
  MBX\$$TXT
ETO\
@
```

****
"ftx.","g_find_text","graphic find text"
****
Dient zum Finden von Texten in JAVA-Setup-Programmen.

****
"wtx.","g_wait_find_text"
****

UNI.wtx.|(zu suchender Text[optional: number bei mehreren][|optional Toleranz]

****
"rab.","g_read_actual_block"
****
Dieser Befehl dient dazu den Text im aktuell lokalisierten Block zu lesen. Der Text wird dabei in einer Variable oder in $v1$ zurückgegeben.
Dieser Befehl setzt vorraus, dass ein Block lokalisiert wurde, egal mit welchen Befehlen (Beispiel: "g_find_by_number", oder "g_navigate" oder zum Beispiel g_scan_object_right).

Der Befehl wird im einfachsten Fall so aufgerufen:

```
' Es wird die interne Standardschrift verwendet
UNI.rab.>$$TXT
```


© 2023 Theo Gottwald * https://www.smart-package.com

MBX\$$TXT

Man kann das Ergebnis des Lesenbefehls mit den Patternmatch-Funktionen (wie "s_wildcard_compare") gegen einen gesuchten Text "matchen", und so fehlertolerant lesen.
Dabei sollten sowohl normal,bold,kursiv als auch beides einigermassen erkannt werden.

Es ist möglich ein alternatives "Schriftarten-Set" anzugeben. Dabei kann man entweder den Namen einer Schriftart oder einfach eine number angeben.
Diese numbern bedeuten dabei folgendes:

"1" = Standardfont
"2" = "Arial"
"3" = "Times New Roman"
"4" = "OCR-A"
"5"= "Verdana"
"6" = "Symbol"

Beispielskript 1:
' Es wird die Schriftart 2 (Arial) verwendet
UNI.rab.|2>$$TXT
MBX\$$TXT

Jeweils beim ersten Aufruf des Befehls mit einem bestimmten "Font" erzeugt der Befehl eine Datendate in "?exepfad\".
So eine Datei ist für jeden Font der gelesen werden soll erforderlich.
Dies kann bis zu 10 Sekunden dauern, diese Datei ist zwischen 1 MB und 3 MB groß je nach Font.
In dieser Zeit wird auf dem System (im Installationsverzeichnis (Zugriff muss erlaubt sein!) ein Datenfile erzeugt, welches für den Befehl notwendig ist. Es hat ca. 1.3 MB.
Wenn Sie das Datenfile haben, muss es nicht mehr neu erzeugt werden.
Sie können das Datenfile auch kopieren und mittels
'#INC:?exepfad\Catseye_4233042457.dat
In die EXE einfügen.

Dadurch ersparen Sie sich die 6 Sekunden Wartezeit beim allerersten Aufruf des Befehl.
Kurz: Diese Datei wird zum Lesen von BAP benötigt, falls sie nicht da ist wird sie erzeugt.

Noch in wichtiger Hinweis für die Tester:
Dieses Datenfile enthält Daten zu den zu lesenden Zeichen.
Sie müssen dieses Datenfile (bis zur Finalen Version) bei jedem Update NEU erzeugen, da es sein kann dass jeweils Zeichen dazukommen.

Beispielskript 2:
VAR\$$ARS=!D1,C1,B1

UNI.g_mask|$$ARS|p>$$TXT
STW\shimgvw:cpreviewwnd\
'GCW\\1\shimgvw:czoomwnd*|w74994

© 2023 Theo Gottwald * https://www.smart-package.com

UNI.g_analyze|PD:5,FL:2,TL:0,TF:0,MS:$$TXT

UNI.fba.|SB-Karte erfassen
MPO\
UNI.rab.>$$TXT
MBX\$$TXT
@

****
"fbn.","g_find_by_number"
****
Dieser Befehl kann zum Lokalisieren von Ojekten verwendet werden, die derzeit nicht gelesen werden können, jedoch als Blöcke angezeigt werden.
Dazu wählt man den erwarteten Quadranten aus und gibt die Blocknumber an.

UNI.fbn.|(Nr. zu suchendes Element)

Beispiel:
VAR\$$ARS=D3,D4
UNI.g_mask|$$ARS|p>$$TXT
STW\shimgvw:cpreviewwnd\
UNI.g_analyze|PD:6,FL:12,TL:10,TF:0,MS:$$TXT
UNI.fbn.|10
MMV\

****
"wbn.","g_wait_find_by_number"
****
Dieser Befehle entspricht "fbn." in der Version mit "wait".
Man kann damit zum Beispiel auf das Erscheinen von Buttons warten.

UNI.wbn.|(Nr. zu suchendes Element)|(Timeout in Sekunden)

****
"gqn.","get_quadrant_rect_by_name"
****
Dient zum Ermitteln des Rechtecks (Koordinaten) eines Feldes.
Das Kommando lokalisiert das Feld, der aktuelle Punkt wird in die Mitte des Feldes gelegt.
Die Koordinaten der Kanten des Feldes stehen dann über die WinRobots-Spezialvariablen zur Verfügung:
il = Linkskoordinate (ItemLeft),
it = Obenkoordinate (ItemTop),
ir = Rechtskoordinate (ItemRight),
ib = Untenkoordinate (ItemBottom),
iw = Breite (ItemWidth),
ih = Höhe (ItemHeight),
(Siehe Hilfe "Spezialordner").

Beispiele:
UNI.get_quadrant_rect_by_name|A1
MMV\

© 2023 Theo Gottwald * https://www.smart-package.com

```
UNI.get_quadrant_rect_by_name|D4
MMV\
```

****
"sco.","g_scan_object_at_point"
****
Dieser Befehl ermittelt das umschreibende Rechteck und den Mittelpunkt eines Objektes, typischerweise einer Textbox.
Dazu wird ein beliebiger Punkt innerhalb des Objektes angegeben, sowie optional eine Farbe und eine Farbtoleranz.
Wird als Farbe -1 angegeben, wird die Farbe an dem angegebenen Punkt als Referenzfarbe verwendet, wird die Farbtoleranz weggelassen, wird 180 als Defaultwert verwendet.
Der Befehl benötigt vorweg ein "g_analyze".

Beispiel:

```
STW\shimgvw:cpreviewwnd\
UNI.g_analyze|PD:5,FL:1,TL:1,TF:&H8C5317

' Hier ist Weiss die Referenzfarbe, die Farbtoleranz ist 180
UNI.sco.|&HFFFFFF|333,242
MMV\

' Hier wird die Referenzfarbe ermittelt und die Farbtoleranz beträgt 100
UNI.sco.|-1,100|333,242
MMV\
```

****
"scr.","g_scan_object_right"
"scl.","g_scan_object_left"
"scd.","g_scan_object_down"
"scu.","g_scan_object_up"
****
Die Befehle dienen dazu, rechts/links/oben/unten liegende Textboxen zu lokalisieren.

Diese Lokalisierung erfolgt anders als die reguläre Lokalisierung der Texte.Daher ist eine Navigation mittels "g_navigate" nur von diesen Objekten zu anderen Objekten möglich.
Eine Navigation zu diesen Objekten ist nur mittels dieser Befehle möglich, nicht mittels "g_navigate".
Ausgenommen davon ist jeweils das zuletzt lokalisierte Objekt.
Es nimmt einen Sonderstatus ein, damit von diesem Objekt aus zu einem Textobjekt navigiert werden kann.

Der Befehl scannt nach rechts/links/oben/unten bis er ein Objekt/Punkt/Linie findet, die einen Farbkontrast zum Hintergrund in der angegebenen Stärke aufweist.
Er versucht dann den Umfang des Objektes zu ermitteln.

© 2023 Theo Gottwald * https://www.smart-package.com

UNI.scr.|(Kontrastwert beim Rechts-Scan]|[inside/outside]|[Farbtoleranz für Objekterkennung]|[Scanweite in Pixel oder 0]

+ Kontrastwert beim Scan - Wertbereich: 1-765, wird 0 angegeben, so wird der Defaultwert von 180 verwendet.
+ inside/outside Point - Wertebereich 0/1...10 - Falls die Objekte eine Umrandung haben, kann man durch Angabe von "1" erreichen, dass ein Punkt innerhalb des Objektes zur Ermittlung des Umfangs herangezogen wird.
Dies ist bei Tabellen erforderlich, da ein Scan des Randes dann die gesamte Tabelle erfassen wird.
Navigiert man Checkmarks, kann ein Wert von 2 hilfreich sein, da damit der Rand übersprungen wird.
+ Farbtoleranz für Objekterkennung - es ist möglich hier eine Farbtoleranz zwischen 0 und 765 anzugeben. Dadurch werden auch Objekte, die Farbübergänge enthalten, korrekt ermittelt.
Default ist der gleiche Wert wie beim Kontrastwert.
+ Scanweite: Default ist 1/2 der Bildbreite. Die Zahl rechnet ab Außenkante des aktuell lokalisierten Elementes.
Der Befehl benötigt ein g_analyze und ein lokalisiertes aktuelles 9er-Element.

Beispiel:

```
VAR\$$ARS=!B4
UNI.g_mask|$$ARS|p>$$TXT

STW\shimgvw:cpreviewwnd\
UNI.g_analyze|PD:5,FL:1,TL:1,TF:&H8C5317,MS:$$TXT
' Von Text
UNI.fba.|Alle Firmen
MMV\
' Zu Checkmark
UNI.scl.|230|1|180
MMV\
' Zu Checkmark darunter
UNI.scd.|130|2|180
MMV\
' Zu Checkmark darunter
UNI.scd.|130|2|180
MMV\

@

' Beispiel feststellen ob ein Checkmark gesetzt ist:
VAR\$$ARS=!B1
' Verwenden Sie 2, 10 oder 26 als Flag!
CAL\$$FLG=3+0
UNI.g_mask|$$ARS|p>$$TXT
STW\shimgvw:cpreviewwnd\
UNI.g_analyze|PD:6,FL:$$FLG,TL:0,TF:&H0,MS:$$TXT
UNI.fba.|telefonisch durchgeführt
UNI.g_scan_object_left|80|2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Aktueller Punkt ist nun in der Mitte der Checkmark
' Hole aktuellen Punkt (Mitte Checkmark)
VAR\$$XPA=$ix$
VAR\$$YPA=$iy$

' kleines Rechteck drumherum ermitteln
CAL\$$XLI=$$XPA-3
CAL\$$XRE=$$XPA+3
CAL\$$YOB=$$YPA-3
CAL\$$YUN=$$YPA+3

' Toleranz für IRC auf 20% Festlegen
CTO\20
IRC\$$XLI,$$YOB,$$XRE,$$YUN|&H0
  MBX\Gesetzt
ELSE
  MBX\Nicht gesetzt
ERC\


##############################################################################
##############################################
##############################################################################
##############################################
##############################################################################
##############################################
'*****************************************************************************
Datentypen
'*****************************************************************************
Der Stapel / Stack

****************************
d_init_stack
****************************
Um verschachtelte Ausdrücke zu parsen, ist ein Stapel-Datentyp sehr hilfreich.

Der "Stack", zu Deutsch "Stapel", ist ein sogenannter LIFO-Datentyp. Das bedeutet
"Last-In-First-Out".
Dasjenige Element, welches man zuletzt hinzugefügt hat, ist dasjenige, das man
zuerst wieder zurückbekommt.
Vergleichbar mit den Karten auf einem Kartenspiel. Die oberste Karte ist die letzte
Karte (Last IN), die auf den Kartenstapel gepackt wurde,
so ist sie automatisch die erste Karte (first out), die man von OBEN auch wieder
herunternehmen kann.

Wir alle kennen solche Stapel aus dem täglichen Arbeitsleben: das Ein- und
Ausgangskörbchen in jedem Büro ist solch ein Stapel.
Um solche Vorgänge script-technisch nachzubilden gibt es nun den Stapel-Datentyp.
Es gibt also nicht nur einen Stapel, sondern viele Stapel.
Um das Scripting damit übersichtlich zu gestalten, bekommen diese Stapel vom User
jeweils Namen, damit diese vom System ansprechbar werden.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Hinweis: Wenn man sich z.B. vom Spielkartenstapel die oberste Karte herunternimmt, so ist diese EXKLUSIV in der Hand und auf dem Stapel steht die nächste Karte zur Verfügung. Das ist das normale Verhalten, das wir alle kennen.
Mit dem Parameter "N", kann man dafür sorgen, dass man einen Blick auf die oberste Karte erhält, diese bleibt aber liegen.
Somit hat man quasi eine Kopie der Karte vor Augen.
Um nun wirklich an die nächste Karte vom Stapel zu bekommen, muss man nun manuell dem System sagen, dass die oberste Karte gelöscht werden soll (d_del_stack_item).

Wenn man einen Stapel verwenden möchte, muss man diesen zunächst benennen. Dazu dient der folgende Befehl:

UNI.d_init_stack|(Name des Stapels)|(Autodeletefunktion an/aus) durch N oder weglassen

Ab dann steht ein Stapel mit diesem Namen zur Verfügung.
Es gibt die Möglichkeit, als weiteren Parameter das "N" wie "NoAutoDelete" anzugeben.
Ist dieser Parameter gesetzt, dann wird beim Abheben eines Elementes von diesem Stack dieses Element nicht automatisch gelöscht.
Existiert ein Stack dieses Namens bereits, löscht der Befehl den Stack und setzt das "N"-Bit neu.
Nach dem Erzeugen eines Stapels, wird dieser Stapel automatisch selektiert. Das bedeutet, dass "Push" und "Pop" Befehle dann auf diesen Stapel angewendet werden.
Beispiel:

```
UNI.d_init_stack|Loops

' Im folgenden Beispiel wird die "N" Option verwendet.
' Bei diesem Stack werden beim Abheben die Elemente nicht automatisch gelöscht.
' Dies muss man dann explizit mit "d_del_stack_item" tun, da die N Option aktiv ist.
UNI.d_init_stack|Loops|N
```

****
d_select_stack
****
Zum Beispiel beim Parsen von XML genügt ein Stack nicht, man benötigt mehrere Stacks.
Um nun festzulegen, welchen der Stacks man aktuell verwendet, wird dieser Befehl benötigt.
Beispiel:

```
UNI.d_init_stack|Loops
' Der neuerliche "d_init_stack" selektiert automatisch den Stack "Ferz"
UNI.d_init_stack|Ferz
' Nun benötigt man den Loops-Stack; also selektiert man diesen
UNI.d_select_stack|Loops

' Löscht das oberste Item von dem gerade selektierten Stack.
UNI.d_del_stack_item
```


© 2023 Theo Gottwald * https://www.smart-package.com

' Löscht die obersten 4 Items von dem gerade selektierten Stack.
UNI.d_del_stack_item||4

****
d_push_on_stack
****
Und wie kommen die Items nun auf den Stack? Mit "d_push_on_stack" natürlich.
Die Syntax lautet:
UNI.d_push_on_stack|(Item-String)[|(Item-String 2) ...

Beispiel:
UNI.d_init_stack|Loops
' Wir legen drei Items auf den neu erstellten Stapel
UNI.d_push_on_stack|Line 150|Line 160|Line 170

****
d_pop_from_stack
****
Sinn macht das Ganze natürlich erst, wenn wir die Items auch wieder vom Stack zurückbekommen. Das geht mit:

UNI.d_pop_from_stack>$$ITM

Man hat die Möglichkeit hier noch den Namen eines Stapels anzugeben:

UNI.d_pop_from_stack|Loops>$$ITM

Dann wird das Item von diesem Stapel genommen. Der Stapel wird aber nicht selektiert.
Ein nachfolgender

UNI.d_pop_from_stack>$$ITM

wird also das Item vom derzeit selektierten Stapel nehmen.

****
d_del_stack_item
****
Will man Items vom Stapel entfernen, zum Beispiel weil die Option "N" beim Deklarieren des Stacks (siehe: d_init_stack) verwendet wurde, so ist dies der Befehl der Wahl. Man kann optional noch den Namen eines Stacks und die Anzahl der Items, die gelöscht werden sollen, angeben.

' Löscht das oberste Item vom aktuellen Stack
UNI.d_del_stack_item

' Löscht das oberste Item vom Stack "Loops"
UNI.d_del_stack_item|Loops

' Löscht die obersten 5 Items vom Stack "Loops"
UNI.d_del_stack_item|Loops|5

© 2023 Theo Gottwald * https://www.smart-package.com

' Löscht die obersten 5 Items vom aktuell selektierten Stack
UNI.d_del_stack_item||5

Der Befehl gibt auf Wunsch die Anzahl der noch auf dem Stack verbleibenden Elemente zurück.
UNI.d_del_stack_item||5>$$CNT

****
d_get_stack_count
****
Gibt die Anzahl der Elemente auf dem jeweiligen Stapel zurück. Gibt man keinen Stapelnamen an, wird der aktuelle Stapel verwendet.
Gibt man den Namen eines Stapels an, wird dieser verwendet.

' Hier bezieht sich der Befehl auf den selektierten Stack
UNI.d_get_stack_count>$$CNT
' Hier bezieht sich der Befehl auf den Stack mit Namen "Loops"
UNI.d_get_stack_count|Loops>$$CNT

****
d_erase_stack
****
Diese Funktion löscht den Stack und zwar auch den Stack-Namen. Er kann danach für einen neuen Stack erneut mit "d_init_stack" verwendet werden.
Beim Beenden des Scripts werden alle Stackspeicher sowieso freigegeben, dies ist also nicht unbedingt manuell erforderlich.
Bitte bei diesem Befehl den Namen des zu löschenden Stacks stets explizit angeben.

UNI.d_erase_stack|Loops

' Im Folgenden erhält $$STK die Anzahl noch deklarierter Stacks.
UNI.d_erase_stack|Loops>$$STK

##
##
##
##
##
##

'**
' String-Befehle
'**
****
s_unwrap,unw
****
"Unwrap" auf Deutsch "Auspacken".
Entfernt linke und rechte Klammer. Diese müssen angegeben werden.
Folgt ein weiterer Parameter "a" oder "a", dann wird sichergestellt, dass an beiden Seiten jeweils alle Klammern entfernt werden.
Das Ergebnis wird als String zurückgegeben.

© 2023 Theo Gottwald * https://www.smart-package.com

UNI.s_unwrap|$$STR|()>$$RES
UNI.s_unwrap|$$STR|{}>$$RES
UNI.s_unwrap|$$STR|<>>$$RES

****
s_wrap,unw
****
"wrap" auf Deutsch "Einpacken".
Fügt linke und rechte Klammer zu einem String hinzu. Diese müssen angegeben werden.
Folgt ein weiterer Parameter "n" oder "no doubles", dann wird sichergestellt, dass an beiden Seiten jeweils nur eine Klammer ist.
In diesem Fall werden evtl. bereits vorhandene Einklammerungen entfernt, bevor diese Klammer hinzugefügt wird.
Das Ergebnis wird als String zurückgegeben.

UNI.s_wrap|$$STR|()>$$RES
UNI.s_wrap|$$STR|{}|no doubles>$$RES
UNI.s_wrap|$$STR|<>|n>$$RES

****
s_replace_word
****
Word-Replace Funktion. Ersetzt nur ganze Wörter (abgetrennt mit Leerzeichen), keine Wörter in Wörtern.
Falls der Zusatz "colon" als 4. Parameter gegeben wurde, werden Wörter innerhalb von " " (Gänsefüsschen / ASC 34) ignoriert.

Beispiele:
VAR\$$STR=Halleluja in der Halle. Er rief "Halleluja in der Halle".
UNI.s_replace_word|$$STR|Halle|Raum|colon
UNI.s_replace_word|$$STR|Halle|Raum
' Ergebnis in $v1$
UNI.s_replace_word|$$STR|Halle|Raum|colon>$$RES
UNI.s_replace_word|$$STR|Halle|Raum>$$RES
' Ergebnis in $$RES

****
s_no_case_instr
****
Diese Funktion sucht eine Zeichenkette in einer größeren Zeichenkette. Dabei wird Groß-/Kleinschreibung ignoriert.
Es gibt die Möglichkeit, die Suchrichtung durch Angabe von "back" als drittem Parameter umzukehren.
In einem vierten Parameter kann eine Startposition angegeben werden.

Beispiele:
' Diese beiden sind identisch in der Wirkung, Ergebnis jeweils in $v1$
UNI.s_no_case_instr|$$STR|findme|f|1
UNI.s_no_case_instr|$$STR|findme
' Ergebnise hier in Variable
UNI.s_no_case_instr|$$STR|findme|back|5>$$RES

© 2023 Theo Gottwald * https://www.smart-package.com

UNI.s_no_case_instr|$$STR|findme|forward|88>$$RES

****
s_equalcase_replace
****
Diese Funktion ersetzt das erste oder alle (Parameter 5) Vorkommen von (Parameter 2) in der Zeichenkette (Parameter 1) ab Position (Parameter 4) durch (Parameter 3). Dabei spielt Groß- und Kleinschreibung keine Rolle. Die originale Groß- und Kleinschreibung der Angaben wird in dem Ersetzungsvorgang trotzdem beibehalten.
Die Angaben der Zeichenketten geschehen wie folgt:
UNI.s_equalcase_replace|(String, in dem ersetzt werden soll)|(String, der ersetzt werden soll)|(String, der ersatzweise eingefügt wird)[|optional Startposition][|optional all]

Beispiele:
VAR\$$STR=Hallo Peter.
VAR\$$IST=Peter
VAR\$$SOL=Franz
UNI.s_equalcase_replace|$$STR|$$IST|$$SOL>$$TAR
UNI.s_equalcase_replace|$$STR|$$IST|$$SOL|1|all>$$TAR
UNI.s_equalcase_replace|$$STR|$$IST|$$SOL|5>$$TAR
UNI.s_equalcase_replace|$$STR|$$IST|$$SOL|15|all
' Im letzteren Fall ist das Ergebnis in $v1$, in den anderen Beispielen ist es in der Zielvariablen $$TAR.

****
s_case_sensitive_replace
****
Diese Funktion ersetzt das erste oder alle (Parameter 5) Vorkommen von (Parameter 2) in der Zeichenkette (Parameter 1) ab Position (Parameter 4) durch (Parameter 3). Dabei werden Groß- und Kleinschreibung beachtet. Die originale Groß- und Kleinschreibung der Angaben wird in dem Ersetzungsvorgang stets beibehalten.
Die Angaben der Zeichenketten geschehen wie folgt:
UNI.s_case_sensitive_replace|(String, in dem ersetzt werden soll)|(String, der ersetzt werden soll)|(String, der ersatzweise eingefügt wird)[|optional all]

Beispiele:
VAR\$$STR=Hallo Peter.
VAR\$$IST=Peter
VAR\$$SOL=Franz
UNI.s_case_sensitive_replace|$$STR|$$IST|$$SOL>$$TAR
UNI.s_case_sensitive_replace|$$STR|$$IST|$$SOL|all>$$TAR
UNI.s_case_sensitive_replace|$$STR|$$IST|$$SOL>$$TAR
UNI.s_case_sensitive_replace|$$STR|$$IST|$$SOL|all
' Im letzteren Fall ist das Ergebnis in $v1$, in den anderen Beispielen ist es in der Zielvariablen $$TAR.

****
s_generate_rnd_string
Transfered to STR.GENERATE
****

© 2023 Theo Gottwald * https://www.smart-package.com

Diese Funktion erzeugt einen Zufallsstring aus den Zeichen in Zeichenkette (Parameter 1) mit der Länge (Parameter 2) oder (Defaultwert) 32.
Dabei kann man (Parameter 1) auch weglassen. Dann werden alle Groß- und Kleinbuchstaben sowie das "-" verwendet.
Alternativ kann man als (Parameter 1) folgende Angaben machen:
- (beliebige Zeichenkette - diese Zeichen werden verwendet)
- leer - es werden Klein- und Großbuchstaben und das "-" verwendet
- "A" die Zeichenkette wird nur aus Großbuchstaben gebildet.
- "a" die Zeichenkette wird nur aus Kleinbuchstaben gebildet.
- "A-" die Zeichenkette wird nur aus Großbuchstaben und dem "-" Zeichen gebildet.
- "a-" die Zeichenkette wird nur aus Kleinbuchstaben und dem "-" Zeichen gebildet.

Beispiele:
' Es werden Klein- und Großbuchstaben und das "-" verwendet, Ergebnis in $$RES mit Länge 32
UNI.s_generate_rnd_string||>$$RES

' Es werden nur die Zeichen aus $$LET verwendet, die Länge des Ergebnisstrings ist hier 45
VAR\$$LET=%&_ABCDefgh
UNI.s_generate_rnd_string|$$LET|45>$$RES

' Es werden hier Großbuchstaben und das "-" Zeichen verwendet. Die Länge des Ergebnisstrings ist hier 8000 Zeichen
VAR\$$LET=A-
UNI.s_generate_rnd_string|$$LET|8000>$$RES

****
s_scan_for_number
****
Diese Funktion ermittelt die erste Ziffer in einer Zeichenkette (Parameter 1), beginnend bei Position (Parameter 2).
Rückgabe ist die Position der ersten Ziffer (0-9).

Beispiele:
VAR\$$STR=<Fontsize123>
UNI.s_scan_for_number|$$STR>$$POS
' $$POS enthält nun die Position der "1" also 10.

' Im Folgenden startet der Scan ab Position 6:
UNI.s_scan_for_number|$$STR|6>$$POS

****
s_fuzzy_compare
****
Diese Funktion vergleicht zwei Zeichenketten. Dabei wird ein gradueller Vergleich angewendet, der entweder:
- eine Übereinstimmung von 0% bis 100% zurückliefert (als Zahl 0 bis 100)
- oder (wenn Parameter 3 = "mistakes") die Anzahl der Tippfehler (Mistakes) zwischen den beiden Strings zurückgibt.

© 2023 Theo Gottwald * https://www.smart-package.com

Beispiele:
VAR\$$TXA=MyCommands
VAR\$$TXB=MyCummands
UNI.s_fuzzy_compare|$$TXA|$$TXB|mistakes>$$RES
' $$RES ist nun 1, da sich der Schreiber einmal vertippt hat.

UNI.s_fuzzy_compare|$$TXA|$$TXB>$$RES
' $$RES ist nun 90, da die Übereinstimmung zwischen den beiden Strings 90% beträgt.

****
s_wildcard_compare
****
Texte einfach nach dem Schema "A = B ?" zu vergleichen, erfordert nicht besonders viel Intelligenz.
Oftmals will man wissen, ob Dinge ähnlich sind, oder bestimmte Bestandteile enthalten sind.
Dazu können Wildcards herangezogen werden.

Die Vergleichsfunktion s_wildcard_compare vergleicht einen Wildcard-String (Parameter 1) mit einem zweiten String (Parameter 2).
Das Ergebnis ist 1 (das bedeutet Übereinstimmung) oder 0 (die Strings stimmen nicht überein).

Es gibt drei verschiedene Wildcard-Sets.

Set 1:

Als Wildcards werden in diesem Fall primär die zwei Joker-Zeichen aus DOS-Zeiten verwendet.
 * - ersetzt beliebige Zeichen
 ? - ersetzt ein Zeichen

Beispiel:
VAR\$$TXA="Jan?a*"
VAR\$$TXB="Januar"
UNI.s_wildcard_compare|$$TXA|$$TXB>$$RES

' Man kann in allen Fällen JIZ\ (Jump-If-Zero) verwenden, um das Resultat auszuwerten.
JIZ\$$RES:Label
' $$RES wäre hier 1, da der Vergleich erfolgreich ist. Andernfalls ist das Ergebnis 0.

Set 2:

Falls der Zusatz "numeric" (Parameter 3) angegeben wird, dann wird zusätzlich das Zeichen
 # - ersetzt eine Ziffer
ausgewertet.

' der Zusatz "numeric" bewirkt, dass auch das # Zeichen ausgewertet wird.

© 2023 Theo Gottwald * https://www.smart-package.com

UNI.s_wildcard_compare|$$TXA|$$TXB|numeric>$$RES
JIZ\$$RES:Label

Set 3:

' der Zusatz "all" bewirkt, dass die AGR\-Stringpatterns verwendet werden. Diese sind unten aufgeführt.
Beispiel:

VAR\$$TXA="{Janner&TOLERANT:&DOCAPS:60}"
UNI.s_wildcard_compare|$$TXA|$$TXB|all>$$RES
JIZ\$$RES:Label

Pattern	Abkürzung	Beispiel	
{&DOCAPS:}	{&0:}	{&DOCAPS:ExPE}	-> Groß-/Kleinschreibung beachten.
{&OR:}	{&1:}	{Text1&OR:Text2}	-> A oder B oder beides
{&XOR:}	{&2:}	{Text1&XOR:Text2}	-> A oder B
{&AND:}	{&3:}	{Text1&AND:Text2}	-> A und B; Reihenfolge beliebig
{&ANDTHEN:}	{&4:}	{Text1&ANDTHEN:Text2}	-> A dann B
{&FUZZY:}	{&5:}	{{Text1}&FUZZY:[value]}	-> ungefähr A
{&TOLERANT:}	{&6:}	{{Text1}&TOLERANT:[value]}	-> A zu min. X%
{&NOT:}	{&7:}	{&NOT:{Text1}}	-> darf nicht vorkommen
{&EXACT:}	{&8:}	{&EXACT:Text1}	-> genau A (Ohne Groß/Klein)
{&NOTEXT:}	{&9:}	{&9:}	
{&t:}			-> Always True
{&f:}			-> Always False

Alle Pattern können auch mit der Abkürzung verwendet werden, also statt &EXACT: kann man einfach &0: schreiben.
Die einfachen Operatoren (&OR, &XOR, &AND, &ANDTHEN) können beliebig oft wiederholt werden, also
{Text1&OR:Text2&OR:Text3&OR:Text4&OR:Text5... usw.}
ist möglich.

Der &NOT: Operator bezieht sich auf das Nichtvorhandensein von Text; er dient nicht dazu, Pattern-Berechnungen zu negieren. Also

&NOT:{text1&AND:text2}

funktioniert so nicht, allerdings kann man die aufgelöste Variante verwenden:

{&NOT:text1}&AND:{&NOT:text2}&AND:{&NOT:text3} ... etc.

Dies funktioniert mit allen einfachen Operatoren.

© 2023 Theo Gottwald * https://www.smart-package.com

Es gibt keine Beschränkung auf 2 Operatoren wie ansonsten bei den WinRobots-Suchmustern, also beispielsweise AFO\(?), STW\ etc. Dadurch sind beliebig viele Internationalisierungen im gleichen Script denkbar.

Das {&EXACT:Text1}-Pattern ersetzt das WinRobots Standard-Suchmuster "Text1&&" (&& am Ende).

Prinzipiell sind auch Verschachtelungen möglich, solange die Klammern korrekt gesetzt werden, also zum Beispiel:

{{{Text1&ORText2}&ANDTHEN:{Text1&ORText4}}&ANDTHEN:Text2}

Neu hinzugekommen ist der fortschrittliche "Tippfehler-Operator"; Beispiel:

{Text1&FUZZY:5}

Die Zahl hinter dem "&FUZZY:" gibt an, wie oft man sich vertippen darf, so dass der Text trotzdem erkannt wird.

Ebenso neu hinzugekommen ist der "&TOLERANT" Operator. Hierbei wird eine %-Zahl angegeben, die die Wortähnlichkeit angibt, ab der der Text der Bedingung entspricht.

Beispiel:
{text1&TOLERANT:89}

{text1&TOLERANT:&DOCAPS:89}

Achtung: Folgendes funktioniert nicht (und macht auch keinen Sinn)
{{text1&OR:text2}&TOLERANT:&DOCAPS:89}
da der "&TOLERANT" Operator nur Texte auswerten kann.

Bei allen Operatoren kann jeweils das "&DOCAPS:" verwendet werden. Dadurch wird die jeweilige Klammerebene auf Groß-/Klein-Schrift umgeschaltet, berücksichtigt dann also Groß- und Kleinschreibung.

##
##
##
##
##
##

****
m_calc
****

Diese Funktion ermöglicht das Berechnen von beliebigen, geklammerten Ausdrücken. Ebenso werden die üblichen Rechenregeln berücksichtigt (Punkt vor Strich etc.). Zusätzlich kann das Ergebnis auf vielfache Art und Weise gerundet werden.

Es gibt die folgenden Operatoren (Auflistung in der Reihenfolge der Priorität):

© 2023 Theo Gottwald * https://www.smart-package.com

' Logische Operatoren - Allerhöchste Priorität
> Größer als
< Kleiner als
= Gleich
! Nicht gleich

^ Hoch - Zweithöchste Priorität

* Multiplikation
% Modulo-Division
/ Division
\ Rest von Division ermitteln

+ Addition
- Subtraktion

Beispiel:
UNI.m_calc|13\2>$$RES
MBX\$$RES

Das Ergebnis des Befehls kann nun auf vielfache Weise nachbearbeitet werden.
Dazu stehen unter anderem folgende Möglichkeiten zur Verfügung:

|i - Integer, rundet auf die nächste Ganzzahl, die gleich oder größer als das Ergebnis ist
|c - rundet auf die nächst kleinere Ganzzahl, die gleich oder kleiner als das Ergebnis ist
|x - schneidet einfach alle Nachkommastellen ab
|f - fractiona, gibt nur die Nachkommastellen zurück
|a - Absolutwert: Schneidet ein evtl. negatives Vorzeichen weg.
|r - rundet auf 2 Dezimalstellen
|s - Sign, gibt 1,0 oder -1 zurück, je nachdem welches Vorzeichen das Ergebnis hatte

Die Zeichen können auch kombiniert werden und werden dann der Reihe nach verarbeitet.

Beispiele:
UNI.m_calc|13\2|x>$$RES
MBX\$$RES

UNI.m_calc|-13\2|fa>$$RES
MBX\$$RES

UNI.m_calc|-13\2|xa>$$RES
MBX\$$RES

VAR.$$NAM=10
VAR.$$FOR=(100*$$NUM)+20
UNI.m_calc\$$FOR|ia>$$ERG
MBX.Das Ergebnis ist: $$ERG

© 2023 Theo Gottwald * https://www.smart-package.com

****
m_eval
****

Dieser Befehl implementiert einen wissenschaftlichen Formel-Rechner, der relativ langsam arbeitet aber kaum Wünsche offen lässt.
Der Rechner ist derzeit nur in englisch implementiert.
Im Fehlerfall erhält man eine aussagekräftige Fehlermeldung, statt des Ergebnisses.

[englisch]
Using extended precision, 18 digits of precision is obtained. Factorials are computed up to 1754! without overflow.

Expressions and functions should follow strictly to standard algebraic syntax rules.
Operators and expressions which can be used are:

+ - * / \ ^ ! ( ) < = > <> >= <=

The Operator Precedence follows algebraic rules.
The only exception is the unary negation operator which in this program does not have a special priority following exponentiation.

However, operations inside parentheses always have the highest priority and are always evaluated first.
Thus by using parentheses in the right places you can always obtain the operator precedence you need for your purpose.

The following operator precedence is used in the program:
1. parentheses ( )
2. unary operators (Factorials (!))
3. one argument functions
4. exponentiation (^) (power expressions)
5. multiplication (*) and floating-point division (/)
6. integer division (\)
7. modulo (MOD)
8. addition (+), subtraction (-)
9. relational operators (<, <=, =, >=, >, <>)
10. NOT, ISFALSE and ISTRUE
11. AND
12. OR and XOR (exclusive OR)
13. EQV (equivalence)
14. IMP (implication)

These are the implemented trigonometric functions:

"ARCSINH() - Arc (Inverse) Hyperbolic Sine"
"ARCCOSH() - Arc (Inverse) Hyperbolic Cosine"
"ARCTANH() - Arc (Inverse) Hyperbolic Tangent"
"ARCSECH() - Arc (Inverse) Hyperbolic Secant"
"ARCCSCH() - Arc (Inverse) Hyperbolic Cosecant"
"ARCCOTH() - Arc (Inverse) Hyperbolic Cotangent"
"ARCSIN() - Arc (Inverse) Sine"

© 2023 Theo Gottwald * https://www.smart-package.com

```
"ARCCOS() - Arc (Inverse) Cosine"
"ARCSEC() - Arc (Inverse) Secant"
"ARCCSC() - Arc (Inverse) Cosecant"
"ARCCOT() - Arc (Inverse) Cotangent"
"SINH() - Hyperbolic Sine","TANH() - Hyperbolic Tangent"
"SECH() - Hyperbolic Secant"
"CSCH() - Hyperbolic Cosecant"
"COTH() - Hyperbolic Cotangent"
"SEC() - Secant"
"CSC() - Cosecant"
"COT() - Cotangent"
"SIN() - Sine","COS() - Cosine"
"TAN() - Tangent"
"ATN() - Arc (Inverse) Tangent"
"LN() - Natural Log (base E)"
"EXP2() - Exponentiation of 2 or Antilog 2"
"EXP10() - Exponentiation of 10 or Antilog 10"
"EXP() - Exponentiation of E or Antilog E"
"SQR() - Square Root","LOG10() - Common Log (base 10)"
"ABS() - Absolute Value"
"LOG2() - Log (base 2)"
"LOG() - Natural Log (base E)"
"AND - Logical AND"
"XOR - Logical XOR (Exclusive OR)"
"OR - Logical OR"
"MOD - MOD (Modulo)"
"IMP - Logical IMP (Implication)"
"EQV - Logical EQV (Equivalence)"
"NOT - NOT","ISTRUE - Logical Truth"
"ISFALSE - Logical Falsity"
"CEIL() - Ceiling"
"FRAC() - Fractional Part"
"FIX() - Truncate To Integer"
"SGN() - Sign"
"PI# - Constant (3.14159..)"
"E# - Constant (2.71828..)"
```

A number of constants and conversions are available, making this math evaluator an advanced scientific calculator.
These "known constants" include:

```
"Foot_TO_meter"                 = "*1/ 3.28084"
"Meter_TO_foot"                , "* 3.28084"
"Inch_TO_centimeter"             , "* 2.54"
"Centimeter_TO_inch"             , "*1 / 2.54"
"Kilometer_TO_mile"              , "*1 / 1.609344"
"Mile_TO_kilometer"              , "* 1.609344"
"Inch_TO_foot"                  , "*1 / 12"
"Foot_TO_inch"                  , "* 12"
"Yard_TO_meter"                  , "*1 / 1.093613"
"Meter_TO_yard"                  , "* 1.093613"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"Fathom_TO_meter"                    , "* 1.8288"
"Meter_TO_fathom"                    , "*1 / 1.8288"
"Mile_TO_light-year"                 , "*1 / 5880000000000"
"Light-year_TO_mile"                 , "* 5880000000000"
"Parsec_TO_light-year"               , "* 3.261643"
"Light-year_TO_parsec"               , "*1 / 3.261643"
"Square_ft_TO_square_m"              , "*1 / 10.76391"
"Square_m_TO_square_ft"              , "* 10.76391"
"Square_in_TO_square_cm"             , "* 6.4516"
"Square_cm_TO_square_in"             , "*1 / 6.4516"
"Hectare_TO_acre"                    , "* 2.471054"
"Acre_TO_hectare"                    , "*1 / 2.471054"
"Pound_TO_kilogram"                  , "*1 / 2.204623"
"Kilogram_TO_pound"                  , "* 2.204623"
"Ton_(metric)_TO_Kilogram"           , "* 1000"
"Kilogram_TO_ton_(metric)"           , "*1 / 1000"
"Ton_(US)_TO_Kilogram"               , "* 907.18474"
"Kilogram_TO_ton_(US)"               , "*1 / 907.18474"
"Ton_(UK)_TO_Kilogram"               , "* 1016.046909"
"Kilogram_TO_ton_(UK)"               , "*1 / 1016.046909"
"Ounce_(avoirdupois)_TO_gram"        , "* 28.349551"
"Gram_TO_ounce_(avoirdupois)"        , "*1 / 28.349551"
"Ounce_(troy)_TO_gram"               , "* 31.103508"
"Gram_TO_ounce_(troy)"               , "*1 / 31.103508"
"Fahrenheit_TO_Celsius"              , "*1 / 1.8 - 32 / 1.8"
"Celsius_TO_Fahrenheit"              , "* 1.8 + 32"
"Celsius_TO_Kelvin"                  , "*1 + 273.16"
"Kelvin_TO_Celsius"                  , "*1 - 273.16"
"Gallon_(US_dry)_TO_liter"           , "* 4.404884"
"Liter_TO_gallon_(US_dry)"           , "*1 / 4.404884"
"Gallon_(US_liquid)_TO_liter"        , "* 3.785412"
"Liter_TO_gallon_(US_liquid)"        , "*1 / 3.785412"
"Quart_(US_dry)_TO_gallon_(US_dry)"  , "*1 / 4"
"Gallon_(US_dry)_TO_quart_(US_dry)"  , "* 4"
"Pint_TO_liter_(US_dry)"             , "* 0.55061"
"Liter_TO_pint_(US_dry)"             , "*1 / 0.55061"
"Pint_TO_liter_(US_liquid)"          , "* 0.473176"
"Liter_TO_pint_(US_liquid)"          , "*1 / 0.473176"
"Cubic_ft_TO_cubic_m"                , "*1 / 35.314667"
"Cubic_m_TO_cubic_ft"                , "* 35.314667"
"Horsepower_(elec.)_TO_watt"         , "* 745.7"
"Watt_TO_horsepower_(elec.)"         , "*1 / 745.7"
"Horsepower_(metric)_TO_watt"        , "* 735.499"
"Watt_TO_horsepower_(metric)"        , "*1 / 735.499"
"BTU/hour_TO_watt"                   , "* 0.293071"
"Watt_TO_BTU/hour"                   , "*1 / 0.293071"
"Kilowatt_TO_watt"                   , "* 1000"
"Watt_TO_kilowatt"                   , "*1 / 1000"
"Kph_TO_mph"                         , "*1 / 1.609344"
"Mph_TO_kph"                         , "* 1.609344"
"Knot_TO_mph"                        , "* 1.150778"
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

"Mph_TO_knot" , "*1 / 1.150778"
"Meter/sec_TO_ft/sec" , "* 3.28084"
"Ft/sec_TO_meter/sec" , "*1 / 3.28084"
"Second_TO_minute" , "*1 / 60"
"Minute_TO_second" , "* 60"
"Minute_TO_hour" , "*1 / 60"
"Hour_TO_minute" , "* 60"
"Minute_TO_day" , "*1 / 1440"
"Day_TO_minute" , "* 1440"
"Day_TO_hour" , "* 24"
"Hour_TO_day" , "*1 / 24"
"Psi_TO_pascal" , "* 6894.757"
"Pascal_TO_psi" , "*1 / 6894.757"
"Psi_TO_atmosphere" , "* 0.068046"
"Atmosphere_TO_psi" , "*1 / 0.068046"
"Psi_TO_kg/sqcm" , "* 0.070307"
"Kg/sqcm_TO_psi" , "*1 / 0.070307"
"Degrees_TO_radians" , "* 1.74532925199432958E-2
"Radians_TO_degrees" , "*1 / 1.74532925199432958E-2

Here are the known conversions:

"Acceleration_of_gravity_(m/sec²)" , "9.80665"
"Dry_air_density_at_STP_(kg/m³)" , "1.293"
"Light_speed_in_a_vacuum_(m/sec)" , "299792458"
"Solar_constant_(watts/m²)" , "1340"
"Sound_speed_at_STP_(m/sec)" , "331.4"
"Standard_atmosphere_(nt/m²)" , "101300"
"Earth_equatorial_radius_(m)" , "6378000"
"Earth_escape_velcity_(m/sec)" , "11186"
"Earth_magnetic_dipole_moment_(amp-m²)" , "6.400E21"
"Earth_mass_(kg)" , "5.983E24"
"Earth_mean_angular_rotational_speed_(rads/sec)" , ".0000729"
"Earth_mean_density_(kg/m³)" , "5522"
"Earth_mean_orbital_speed_(m/sec)" , "29770"
"Earth_polar_radius_(m)" , "6357000"
"Earth_volume_(m³)" , "1.087E21"
"Alpha_particle_mass_(kg)" , "6.64465598E-27"
"Electron_charge-mass_ratio_(coul/kg)" , "175881961000"
"Electron_rest_mass_(kg)" , "9.10938188E-31"
"Muon_rest_mass_(kg)" , "1.88353109E-28"
"Neutron_rest_mass_(kg)" , "1.67492716E-27"
"Proton_rest_mass_(kg)" , "1.67262158E-27"
"Atomic_mass_constant_(kg)" , "1.66053873E-27"
"Avogadro_constant_(1/mole)" , "6.02214199E23"
"Boltzmann_constant_(joule/K)" , "1.3806503E-23"
"Elementary_charge_(coul)" , "1.602176462E-19"
"Faraday_constant_(coul/mole)" , "96485.3415"
"Gravitational_constant_(nt-m²/kg²)" , ".000000000066726"
"Permeability_constant_(henry/m)" , "1.25663706143E-6"
"Permittivity_constant_(farad/m)" , "8.85418781762E-12"

© 2023 Theo Gottwald * https://www.smart-package.com

"Planck_constant_(joule-sec)" , "6.62606876E-34"
"Rydberg_constant_(1/m)" , "10973731.568549"
"Stefan-Boltzmann_constant_(watt/m²K4)" , ".000000056704"
"Universal_gas_constant_(joule/K-mole)" , "8.314472"
"Wien_displacement_constant_(m-K)" , ".00290"

Input of angles can be in degrees or radians, depending if you specify the "#" as an additional flag.
However, trigonometric output using the ARC functions will always be in radians.
You can transform angles to radians and vice versa.

Beispiel:

```
VAR\$$FOR=SIN(1)*Inch_TO_centimeter
' Die Angabe des |# bewirkt Winkelgerechnungen in "Degree" statt "Radian"
UNI.m_eval|$$FOR|#
VAR\$$RES=$v1$
MBX\$$RES

VAR\$$FOR=(5*Inch_TO_centimeter)*Acceleration_of_gravity_(m/sec²)
UNI.m_eval|$$FOR|#>$$RES
MBX\$$RES
```

****
m_is_leapyear
****
Die Funktion ermittelt, ob das angegebene Jahr (im Bereich 0 bis 5000) ein Schaltjahr ist oder nicht.
Die Rückgabe ist dementsprechend 1 oder 0.
Fehlt die Jahresangabe, dann wird das aktuell auf dem PC eingestellte Datum zugrunde gelegt.

Beispiel:

```
UNI.m_is_leapyear|1968>$$RES
UNI.m_is_leapyear|2015
VAR\$$RES=$v1$
' Im Folgenden wird das aktuelle Jahr zugrunde gelegt, da die Jahresangabe fehlt:
UNI.m_is_leapyear>$$RES
```

****
m_get_time
****
Die Funktion "m_get_time" ermittelt Zeit und/oder Datum. Darüber hinaus können auch Datumsberechnungen ausgeführt werden.
Da es hierbei wichtig ist, dass man genau das gewünschte Format erhält, unabhängig von den jeweiligen Ländereinstellungen, kann man hier das Format selbst angeben.
Dies geschieht unter Verwendung der folgenden Buchstaben:
y - Jahr
m - Monat

© 2023 Theo Gottwald * https://www.smart-package.com

d - Tag
h - Stunde
n - Minute
s - Sekunde
t - Tick (Mikrosekunde)
Ein einfacher Aufruf der Funktion sieht so aus:
UNI.m_get_time|dd>$$RES
Ergebnis: 23 - wenn gerade der 23.te des Monats ist.

UNI.m_get_time|dddd>$$RES
Ergebnis: In diesem Fall 0023 da vier "d" angegeben wurden.

' Ein komplettes Datum erhält man so:
UNI.m_get_time|dd/mm/yyyy>$$RES
Ergebnis in $$RES ist hier zum Beispiel: 03/02/2011

Auf die gleiche Weise kann man die Zeit ermitteln:
UNI.m_get_time|nn>$$RES
Ergebnis wäre zum Beispiel 03 (falls es gerade 7:03 ist).

UNI.m_get_time|nn-ss-tttt>$$RES
' Hier erhält man zum Beispiel Minuten, Sekunden und Ticks.

' Zeitpunkt in 4 Jahren in einem User-definierten Format
UNI.m_get_time|dd/mm/yyyy|AY:4>$$RES

Die Funktion bietet außerdem die Möglichkeit, ausgehend von der jetzigen Zeit Datums-Berechnungen auszuführen.
Dazu kann man als zweiten Parameter angeben, welcher Betrag von Tag, Monat, Jahr etc. abgezogen oder hinzuaddiert werden soll.
Diese Angabe erfolgt durch Buchstabenkombinationen von "A" (Add) oder "S" (Subtract) und dem jeweiligen Zeitbuchstaben wie folgt:

AY:(Zahl) ' Jahre hinzuzählen
SY:(Zahl) ' Jahre abziehen

AM:(Zahl) ' Monate hinzuzählen
SM:(Zahl) ' Monate abziehen

AD:(Zahl) ' Tage hinzuzählen
SD:(Zahl) ' Tage abziehen

AH:(Zahl) ' Stunden hinzuzählen
SH:(Zahl) ' Stunden abziehen

AN:(Zahl) ' Minuten hinzuzählen
SN:(Zahl) ' Minuten abziehen

AS:(Zahl) ' Sekunden hinzuzählen
SS:(Zahl) ' Sekunden abziehen

AT:(Zahl) ' Ticks hinzuzählen

© 2023 Theo Gottwald * https://www.smart-package.com

ST:(Zahl) ' Ticks abziehen

Beispiele:
UNI.m_get_time|yy-mm-dd|AH:200,SN:400>$$RES
' Es werden zum aktuellen Datum 200 Stunden hinzugezählt und dann 400 Sekunden abgezogen.
' Das Ergebnis wird im gewünschten Format ausgegeben.

UNI.m_get_time|dd;mm;hh|AD:50,SH:90>$$RES
' Es werden zum aktuellen Datum 50 Tage hinzugezählt und dann 90 Stunden abgezogen.
' Das Ergebnis wird im gewünschten Format ausgegeben in die Variable $$RES.

Der Befehl kann auch für Datumsberechnungen verwendet werden.
Dazu wird die Startzeit mit den folgenden Optionen geändert:

NY: (geänderte Jahreszahl 0 ... 5000)
NM: (Monatszahl: 1 ... 12)
ND: (Tag: 1 ... 31)
NN: (Minute: 0 ... 59)
NS: (Sekunde: 0 ... 59)
NT: (Millisekunde: 0 ... 999)

Beispiel:
UNI.m_get_time|yy-mm-dd|NY:2013,NM:5,ND:5,AH:200,SN:400>$$RES

Man kann so auch einfach ein Datum formatieren (ohne Berechnung):
UNI.m_get_time|yy-mm-dd|NY:2013,NM:5,ND:5>$$RES

****
"dtn.","m_date_to_number"
****
Der Befehl wandelt ein Datum in eine 32 bit-Zahl um. Dadurch können Datumsvergleiche einfach mittels IVB\ erfolgen.

Beispiel 1:
UNI.dtn.|15-03-2005
VAR\$$DAT=$v1$
UNI.ntd.|$$DAT
MBX\$v1$

Beispiel 2:
UNI.dtn.|15-03-2005
VAR\$$DAA=$v1$

UNI.dtn.|19-04-2005
VAR\$$DAB=$v1$

UNI.dtn.|20-04-2005
VAR\$$DAC=$v1$

IVB\$$DAC|$$DAA|$$DAB

© 2023 Theo Gottwald * https://www.smart-package.com

```
  MBX\Datum ist innerhalb des Zeitraumes
ELSE
  MBX\Datum ist nicht innerhalb des Zeitraumes
EVB\
```

Beispiel 3:

```
' Lässt man das Datum weg, wird das heutige "Local Time" Datum verwendet.
UNI.dtn.|
VAR\$$DAA=$v1$

UNI.dtn.|17-04-2011
VAR\$$DAB=$v1$

CAL\$$DIF=$$DAB-$$DAA
MBX\$$DAA-$$DAB--$$DIF
```

****
"ntd.","m_number_to_date"
****
Der 15-03-2011 ergibt zum Beispiel 149823.
Der 16-04-2011 ist 32 Tage später ergibt also 149855.
Auf diese Weise kann man auch leicht DAtumsberechnungen ausführen, auf Tagesbasis.
Mit der "number_to_date" Funktion kann man dann die berechneten Werte wieder in Reale Daten zurückumwandeln.

****
"rdi.","m_return_date_info"
****
Returns Date-Info (Systemunabhängig). Je nach Angabe eines Zahlenparameters werden unterschiedliche Informationen zum aktuellen Zeitpunkt ermittelt und zurückgegeben:

0 - dayname,
1 - Monthname,
2 - Datenum,
3 - yearnum,
4 - time,
5 - Time + Milliseconds,
6 - DayOfWeek (as number 0-6),
7 - Datum (deutsche Schreibweise),
8 - Wochentag (immer Englisch!),
9 - Wochetag (nur 2 Buchstaben, gross)
10 - Zeit - nur Stunde und Minute

Beispielskript Datumsfunktionen:

```
' Diese Angabe könnte in einer INI-Datei stehen.
' Sie sagt aus in welchen zeiträumen ein Unterskript ausgeführt werden soll (oder nicht)
VAR\$$INT=Mo*7:00-12:00,Di*9:00-10:00
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FEM\1,|$$INT|DoIt|$$INA
MBX\Fertig
@

:DoIt
' Heutiger Tag als Zahl
UNI.m_return_date_info|6>$$DAN

' Erste Zwei Buchstaben aus Ini-Zeile sind der Tag
GSS\1,2|$$INA>$$DAY
GSB\Daynum
' Wenn nicht heutiger Tag gemeint ist
JIV\$$RES!$$DAN:enz

' Ab hier Uhrzeit prüfen
' Aktuelle Zeit holen
UNI.m_return_date_info|10>$$TIM

' Zeitangaben holen
SBD\|$$INA|*|2>$$TMS

' In von / bis aufspalten
SBD\|$$TMS|-|1
' Von-Zeit
VAR\$$TMA=$v1$
' Bis Zeit
VAR\$$TMB=$v2$

UNI.m_verify_time_format|$$TMA>$$TMA
UNI.m_verify_time_format|$$TMB>$$TMB
UNI.m_verify_time_format|$$TMS>$$TMS

UNI.m_time_to_number|$$TMA>$$TAN
UNI.m_time_to_number|$$TMB>$$TBN
UNI.m_time_to_number|$$TMS>$$TSN

IVB\$$TSN|$$TAN|$$TBN
  ' Zeit ist innerhalb der genannten Parameter
ELSE
  ' Zeit ist nicht innerhalb der genannten Parameter
EVB\


:enz
@

:Daynum
SLC\$$DAY
CSE\MO
   VAR\$$RES=1
CSE\DI,TU
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  VAR\$$RES=2
CSE\MI,WE
  VAR\$$RES=3
CSE\DO,TH
  VAR\$$RES=4
CSE\FR
  VAR\$$RES=5
CSE\SA
  VAR\$$RES=6
CEL\
  VAR\$$RES=7
ESL\
RET\
```

****
"vdf.","m_verify_date_format"
****

Der Befehl stellt sicher, dass ein String das notwendige deutsche Datumsformat DD-MM-YYYY hat.
Es wird auch geprüft ob die Zahlen im erforderlichen Bereich sind.
Der String wird falls erforderlich um führende "0" ergänzt und so zurückgegeben.

Beispiel:

```
UNI.vdf.|41-2-99
MBX\$v1$

' Da 41 keine gültige Datumsangabe wirde der Wert auf 31 reduziert.
Ausgabe: 31-02-0099
```

****
"vtf.","m_verify_time_format"
****

Der Befehl stellt sicher, dass ein String das notwendige deutsche Zeitformat HH:MM:SS hat.
Es wird auch geprüft ob die Zahlen im erforderlichen Bereich sind.
Der String wird falls erforderlich um führende "0" ergänzt und so zurückgegeben.

Beispiel:

```
UNI.vtf.|24-1-9
MBX\$v1$
@
```

Rückgabe: 23:01:09

****
"ttn.","m_time_to_number"
****

Erzeugt eine Zahl, die die Anzahl der Dekunde nseit Mitternacht angibt.

© 2023 Theo Gottwald * https://www.smart-package.com

Dadurch lassen sich Zeitabfragen leicht via IVB\ ausführen.
Ebenso werden Zeitberechnungen mit Sekunden vereinfacht.

Beispiel:

```
UNI.ttn.|10:09:00
VAR\$$DAT=$v1$
MBX\$$DAT
' Ausgabe: 36540

UNI.ntt.|$$DAT
MBX\$v1$
' Ausgabe: 10:09:00
@
```

****
"ntt.","m_number_to_time"
****
Gegenfunktion zu "m_time_to_number". Nach einer Zeitberechnung läßt sich damit die Anzahl der Sekunden wieder in die Tageszeit umrechnen.

Beispiel:

```
UNI.ttn.|10:09:00
VAR\$$DAT=$v1$
MBX\$$DAT
' Ausgabe: 36540

UNI.ntt.|$$DAT
MBX\$v1$
' Ausgabe: 10:09:00
@
```

##
##
##
##
##
##
Mit diesen Befehlen können die Robots untereinander kommunizieren.

****
"csi.","c_create_alive_signal"
****
Erzeugt ein systemweites Signal, das von einem anderen Robot via "c_try_signal" abgefragt werden kann.
Signale werden automatisch entfernt, wenn der Robot, der das Signal gesetzt hat beendet wird.
Die Rückgabe ist der Name des Signals, falls ein Name angegeben wurde, dieser.
Falls kein Signal erstelltw erden konnte, ist die Rückgabe ein Minuszeichen/Bindestrich "-".

****

© 2023 Theo Gottwald * https://www.smart-package.com

"tsi.","c_try_alive_signal"
****
Fragt ob ein systemweites Signal gesetzt wurde. Rückgabe ist 1/0 je nach dem ob das Signal von irgendwoher gesetzt wurde oder nicht.

****
"rsi.","c_remove_alive_signal"
****
Entfernt ein systemweites Alive-Signal. Rückgbae ist 0/1 je nach dem ob das Signal entfernt werden konnte oder nicht.
ACHTUNG: Ein globales "Alive Signal" kann NUR von der robot-Instanz gelöscht werden, die es erzeugt hat.
Wenn der Robot sich beendet (egal wie) wird automatisch auch das "Alive Signal" gelöscht.

****
"wsi.","c_wait_alive_signal"
****
Wartet bis das angegebene Signal erzeugt wird oder verschwindet.
Es kann als weiterer Parameter ein Timeout angegeben werden.
Falls das Timeout erreicht wird, wird das ITO\ (Timeout-Flag) gesetzt, ansonsten ist die Rückgabe "1".
Falls auf das Verschwinden eines Signals gewartet wird, muss das Signal beim Aufruf des Befehls bereits vorhanden sein,
sonst ist die Bedingung sofort erfüllt.

Beispiele:

```
' Signalname "RUN", 10 Sekunden Timeout ob Signal erscheint
UNI.c_wait_alive_signal|RUN|10
ITO\
  MBX\Signal nicht erschienen!
ETO\
@

' Signalname "RUNX", 360 Sekunden Timeout ob Signal verschwindet
UNI.c_wait_alive_signal|!RUNX|360
ITO\
  MBX\Signal nicht verschwunden!
ETO\
@
```

****
"csf.","c_create_signal_flag"
****
Erzeugt ein systemweites Flag, das von einem anderen Robot via "c_try_signal_flag" abgefragt werden kann.
Anders als Signale, werden Flags nicht automatisch entfernt, wenn der Robot, der das Flag gesetzt hat beendet wird.
Ausserdem kann ein Flag von jedem Robot gelöscht werden, nicht nur von dem, der es erzeugt hat.

© 2023 Theo Gottwald * https://www.smart-package.com

Die Rückgabe ist der Name des Signals, falls ein Name angegeben wurde, dieser.
Falls kein Signal erstellt erden konnte, ist die Rückgabe ein Minuszeichen/Bindestrich "-".

```
' Robot 1
UNI.csf.|Theo
TDL\500
UNI.rsf.|Theo
CAS\2
@

' Robot 2
UNI.wsf.|Theo
UNI.wsf.|!Theo
@
```

****
"tsf.","c_try_signal_flag"
****
Fragt ob ein systemweites Flag gesetzt wurde. Rückgabe ist 1/0 je nach dem ob das Flag von irgendwoher gesetzt wurde oder nicht.

****
"rsi.","c_remove_alive_signal"
****
Entfernt ein systemweites Flag-Signal. Rückgbae ist 0/1 je nach dem ob das Signal entfernt werden konnte oder nicht.
Ein globales "Flag Signal" kann von jeder Robot-Instanz gelöscht werden.
Ein "Flag Signal" ist permanent und besteht auch weiter, wenn der Robot sich beendet, der das "Flag Signal" gesetz hat.

****
"wsf.","c_wait_signal_flag"
****
Wartet bis das angegebene Flag Signal erzeugt wird oder verschwindet. Es kann als weiterer Parameter ein Timeout angegeben werden.
Falls das Timeout erreicht wird, wird das ITO\ (Timeout-Flag) gesetzt, ansonsten ist die Rückgabe "1".
Falls auf das Verschwinden eines Signals gewartet wird, muss das Signal beim Aufruf des Befehls bereits vorhanden sein, sonst ist die Bedingung sofort erfüllt.

Example

```
STW.ct|applicationframewindow|fotos
SPC.c|applicationframeinputsinkwindow
UNI.g_analyze|PD:8,FL:12,TL:50
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

_

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.64.4 IEM. / NEM. - If Execution-Mode

MiniRobotLanguage (MRL)

IEM. Conditional Statement

If Execution Mode

Intention

Conditional Statement.

Use this statement to test if:

- if the Robot is running as a compiled executable.
- if the robot is been started as a parallel-robot.
- if the robot is running under Editor control.

`NEM.` - is the negative form of this statement.

Syntax

IEM.[P1]

Parameter Explanation

P1 - (optional) Condition to test for. If omitted, "1" is used.

Can be any of these:

"1","exe","executable"
Evaluate TRUE if the script is running as a compiled executable. This is the default value.

"2","editor"
Evaluate TRUE if the script is running from inside and under control of the Editor.

"3","father","sub-robot","subrobot"
Evaluate TRUE if the script is running as a Parallelrobot, using the PRR./PRE. Commands.

Example

```
'**********************************
' IEM.-Sample
'**********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
IEM.1
  $$PAT=?exeloc\
ELS.
  $$PAT=?path\
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- 3.1 Systemvariables 334

© 2023 Theo Gottwald * https://www.smart-package.com

3.64.5 ITO. / NTO. - If TimeOut

MiniRobotLanguage (MRL)

ITO. Conditional Statement

If TimeOut-Flag set

Intention

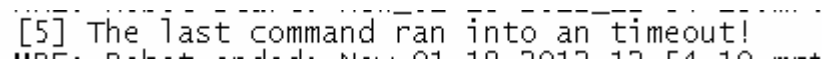
Conditional Statement.

Test if the Timeout-Flag #tio# 334 has been set by the previous command.

A command may set the timeout flag under error conditions, or when it runs into a *TimeOut*. Here is an example:

```
' The following STW. will run into an TimeOut
STW.Xwt|0|no-such-window
ITO.
  DBP.The last command ran into an timeout!
EIF.
ENR.
```

If you run the script, this is what you see in the Debug-window of the editor.



Many other commands also set the TimeOut-Flag in case of error-conditions. As well, mos commands clear the TimeOut-Flag if there are no error-conditions, and no timeout. Therefore always check the TimeOut-Flag immediately after the command that you want to test for success.

You can call ITO. with a parameter. In this case the value will overwrite the TimeOut-Flag and force (1) true or (0) false evaluation.
A value of 2 or if the variable is empty will make ITO. to do its normal job and check the TimeOut-Flag..

ITO. will always leave the TimeOut-Flag cleared after it has been called.

There are two Systemvariables 334, representing the state of the timeout-flag:

```
#tio#    -        Timeout-Flag
#til#    -        Timeout-Line-Number
```

NTO. is the negative Form of ITO.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

ITO.[P1]

Parameter Explanation

P1 - (optional) variable that evaluates to "1" or "0". In this case the value will overwrite the TimeOut-Flag and force (1) true or (0) false evaluation.
A value of 2 or if the variable is empty will make `ITO.` to do its normal job and check the TimeOut-Flag.

Example

```
'**********************************
' ITO.-Sample
'**********************************
' The following STW. will run into an TimeOut
STW.Xwt|0|no-such-window
ITO.
  DBP.The last command ran into an timeout!
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- JIT. - Jump if TimeOut 2174
- JNT. - Fump If-No TimeOut 2203
- STF. - Set TimeOut Flag 2410

© 2023 Theo Gottwald * https://www.smart-package.com

3.64.6 OPT. - Optional Settings

MiniRobotLanguage (MRL)

OPT. Command

Optional Settings

Intention

The OPT-Command enables you, to set, reset or retrieve general system-wide settings.
The three Modes of Operation are:

1. Change Settings:

```
OPT.HM|25
```

2. Reset Setting to default value:

```
OPT.HM|reset
```

3. Retrieve actual Setting to TOS 104:

```
OPT.HM|show
```

Here is an example-script

```
OPT.HM|show
MBX.The Human MouseMove Flag is: $$000
ENR.
```

In the same way, you can change that value and turn "Human Mouse Move" off:

```
OPT.HM|0
ENR.
```

or reset it to its default value:

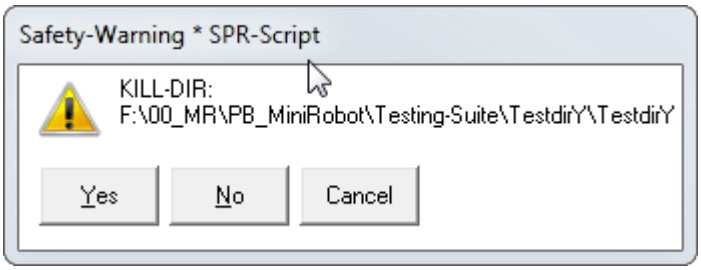
```
OPT.HM|reset
ENR.

' Another example

OPT.FIF|128
ENR.

' Or reset to default value
OPT.FIF|reset
ENR.
```

Using the

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
OPT.SAFEFILE|1
```

will bring up such a Messagebox for destructive File-Operations like `DEL.`, `RMD.` `REN.` and `DCP.`



You can choose between

- "**Yes**" - this will start the requested file-operation
- "**No**" - this will prevent the requested file-operation and set the Timeout-Flag
- "**Cancel**" - this will prevent the requested file-operation and set the Timeout-Flag and end the script.

Syntax

OPT.P1|P2

Parameter Explanation

```
P1 -  1 - HM  = "Human Mousemove", value 0 (off) to 45 (very
strong)
      2 - WII = "Wait for Input Idle" Value, in seconds.
      3 - MaxVarRes = Maximal Number of variable resolutions;
          default = 255.
      4 - STSZ = Standard Searchtime for Top-Windows in
seconds,
           default = 60 Sec.
      5 - SCSZ = Standard Searchtime for Child-Windows in
seconds,
          default = 30 Sec.
      6 - WS  =  Standard-Fenstergröße bei EXE-Starts.
      7 - ERN = Last Error Number and Text. Using the "Set"-
Mode
          with this, will only set the error-number. The error
          text will be "Synthetic Error"
      8 - SILENT = Internal Number that shows the actual
"silent"
          state. Changing this number will change the "Silent-
Mode"
          behavier, details see USM. 786
      9 - DIP, the acual DIP-value. CHanging this number will
          override the last setting with the DIP. 773-command.
     10 - Default ExitCode. Changing this value will override
the
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
          last setting with SEC. - Set Exit Code2129 and other
commands.
     11 - REG64 - Switch 64 bit Registry on
     12 - REG32 - Switch 64 bit Registry off
     13 - FIF - Set maximum Search Pattern Size for the FIF.-
Command
          default = 256.
    14 - ENCRYPT .Standard Strenght (Rounds) for TEA-
 Encryption/Decryption
    15 - SPLITCOMP - Setting for FIL.SPLIT-Command. If set,
 command will use encryption and compression.
    16 - SAFEFILE - Setting for RMD. and DEL. etc. If set, a
 WARNING Messagebox will appear
         before any such destructive file-Operation.
    17 - PWSWAIT - Waiting time for Powershell, default is 60
 Seconds.
```

Example

```
'***********************************
' OPT.-Demo
'***********************************
' "show" will place the result on the TOS
OPT.HM|show
MBX.The Human MouseMove Flag is: $$000
ENR.
'***********************************

'***********************************
' OPT.-Demo
'***********************************
' "show" will place the result on the TOS
OPT.FILESAFE|1
' Now before the folder is been deleted, a warning MSGBOX will
appear.
RMD.d:\folder
ENR.
'***********************************
```

Remarks

Be careful when using this command. If you set wrong values here, your scripts will fail without visible errors.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- DBP - Debug Print 769
- DBV. - Debug Print Variable 771
- DBO. - Debug Raw Output 767
- DIP. - Delayed Instruction Processing 773
- USM - Use silent Mode 786
- DMP. - Dump internat Informations 775
- MLC - Mouse-Left-Click (Invisible) 2448
- MRC. / MRI. - Mouse-Right-Click 2472
- VAV. - Variables with Variables 3566

© 2023 Theo Gottwald * https://www.smart-package.com

3.64.7 STF. - Set TimeOut Flag

MiniRobotLanguage (MRL)

STF. Command

Set TimeOut Flag

Intention

The `STF.`-Command can be used to manually set the timeout flag to simulate a timeout condition.

```
' This will set the timeout-flag
STF.1

' This will clear the timeout-flag
STF.0
```

There are two Systemvariables 334, representing the state of the timeout-flag:

```
#tio#    -        Timeout-Flag
#til#    -        Timeout-Line-Number
```

Syntax

STF.P1

Parameter Explanation

P1 **-** ***(optional)*** VAR or number.If omitted, the timeout-flag will be cleared.

Example

```
' This will set the timeout-flag
STF.1

' This will clear the timeout-flag
STF.0
```

Remarks

*© 2023 Theo Gottwald * https://www.smart-package.com*

There are two Systemvariables 334, representing the state of the timeout-flag:

```
#tio#    -        Timeout-Flag
#til#    -        Timeout-Line-Number
```

Limitations:

See also:

- 1.6.1. Program Flow Control 121
- JIT. - Jump if TimeOut 2174
- JNT. - Fump If-No TimeOut 2203
- ITO. / NTO. - If TimeOut 2404
- 3.1 Systemvariables 334

© 2023 Theo Gottwald * https://www.smart-package.com

3.65 Mouse-Operations

3.65.1 ! Smarty's Mouse-Operations

Smarty's Mouse-Operations Previous 2410 Top 41 Next 2431

Smarty's Mouse-Operations

Mouse-Positioning Engine Prototypes (Options for all Mouse-Mov

The use of the Mouse Positioning Engine is Simple. Just type `MMV.MLC.` or whatever you want to do.
This may be enough in 90% of all cases.

Then there may be a few advanced cases where you need more Features to "hit the point".
For these cases where you need more fine-tuning, you do get a lot of Options through the use of the following Prototypes (detailed Explanation see below!):
Generally in any Mouse-Command you can specify one or more Prototypes. And additionally you can specify coordinates or Quadrant-Coordinates.
This system was developed over the years to give you maximum flexibility. Here is an example.

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dua

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning, these 3 can not be combined with themselves:

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and y
"b" - last mouse-position, before last Move.
"p" - relativ to current "Active Point". (Default)
"l" - Last "Active Point".
```

Normally these commands just "Act" on the current "Active Point", but they do NOT Move the current "Active Point".
If you want to do so, just add the `"x"`-Prototype.

```
"x" - Set Active Point at this position.

Example:
 MMV.oex|-20,90 {A2-B4}
```

There is no rule about the order of the prototypes. Therefore:

```
 MMV.oex|-20,90 {A2-B4}
```

and

```
 MMV.xeo|-20,90 {A2-B4}
```

are both functionally equal.

Following is a more "in detail" Description:

Mouse-Positioning Engine

Generally the Location where the Mouse will do its next Action is called the "Activity Point" or in short: "Active Point".
The "Active Point" is normally set by other commands that localize a window or object.
In each of these cases, the Active Point is the Middle-point of these rectangular Areas.
Therefore a Mouse Operation will normally start with the Selection of an Application-Window (called "Top-Windows").

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.t|Title
```

This will "Locate the Window". Internally the Coordinates of the Window are stored, and the Point in the Middle of the Window becomes the "Active Point".
Now we can choose a Button. For example with:

```
SCW.c|Button
```

Next we can Move the Mouse there:

```
MMV.
```

This is the general Principle how the Mouse-Position Engine works.

1. Choose the Application-Window (`STW.`)
2. Choose the Button/Control etc. (`SCW./AGR.` etc.)
3. Move the Mouse or Click at that Place.

Now so far the easy part. You can do that in 90% of all cases this is all you need.
For the rest of cases,we have added superior possibilities to fine-tune Mouse-Positioning and Clicks.

You can use `LBO.` or `RBO.` to select a side of the target-Rectangle.
In case you do not want to click in the Middle, but rather left side, or right side.

```
STW.At|TestApp
AGR.43|Button 2
' Will change the RECT to the Left Side
LBO.
MMV.o|{A1}
ENR.
```

As `LBO.` will change the RECT, it can be combined with all fine-tuning Options, like numeric Input or Quadrant-Coordinates.
In such a case, the Engine will first calculate the Quadrant and then add the numeric input to the Quadrant-Middlepoint.

```
STW.At|TestApp
AGR.43|Button 2
' Will change the RECT to the Left Side
LBO.
MMV.o|-20,10{G4}
ENR.
```

Let's take a look on this line in detail:

```
MMV.o|-20,10{G4}
```

We use the Option "`o`" this says "use the rectangle from the last object". The last Object was the Button, that we have localized using "`AGR.`"
If we would not write anything, the Mouse would drive to 0,0 - which is the upper left corner of the Button.
Now we fine-tune the Position relative to the upper-left-corner. We user a Quadrant-Coordinate `{G4}`.
For this the current Rectangle is been split into 8 pieces. They get Names from "`A1`" to "`H8`".

© 2023 Theo Gottwald * https://www.smart-package.com

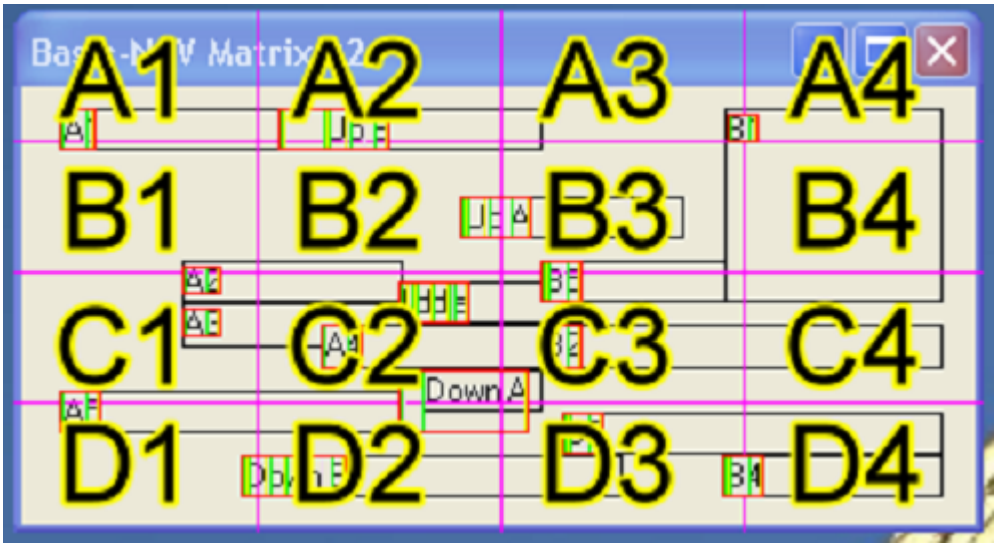
They are sorted like in the picture below. And the Mouse will drive to the Centerpoint of this Rectangle "G4".
And now the numeric coordinates are added to this Position in the last step.

You can use Quadrant-Coordinates for fine-tuning.

All Quadrant-Coordinates in the Mouse-Position Engine are 1/8 th Quadrants.

They go from **A-H** and from **1-8**

This is different from the Standard-Quadrants which are 1/4 Quadrant (see below).



Here is an Example:

```
STW.At|TestApp
AGR.43|Button 2
MMV.o|{H7}
ENR.
```

Where it all started: Screen-Coordinates

Generally a Computer--Screen is made up of "Pixels". Depending on your Screen-Resolution, you may have for example

1920 Pixels horzontally and
1080 Pixels in vertical direction.
which would make your Screen to be build from a total of 1920x1080=2073600 Pixels.

Now you can Simply move the mouse by giving it the Coordinates of an Pixel.
For example, if you want the Mouse to drive to the Center of the Screen, you can just type:

```
' This will Move the Mouse to these Screen Coordinates
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MMV.960,540
' This will just end the Script
ENR.
```

This will Move the Mouse-pointer to these "Screen Coordinates".

Please note that in all Examples, you can just replace the Command `MMV.(Mouse Move)` with the Command `MLC. (Mouse Left Click)`
to make the Robot Click to that place, instead of just Moving there.

```
' This will Move the Mouse to these Screen Coordinates and make a Left Mouse-Cli
MLC.960,540
' This will just end the Script
ENR.
```

Depending on your Screen-Resolution, this place "`960,540`" may, or may not be the Centerpoint of your Screen.

Most often we do not use Screen-Coordinates, because we do not know the Screen-Resolution,
and we do not know "where on the Screen" our "Button" (or whatever we want to click) really is.

So whats the next step?

Better: Window-Coordinates

If we tell the Robot which "Window" we like to click into, we can use "Window-Coordinates".
Unlike Screen-Coordinates, these start at the upper left corner of the window.

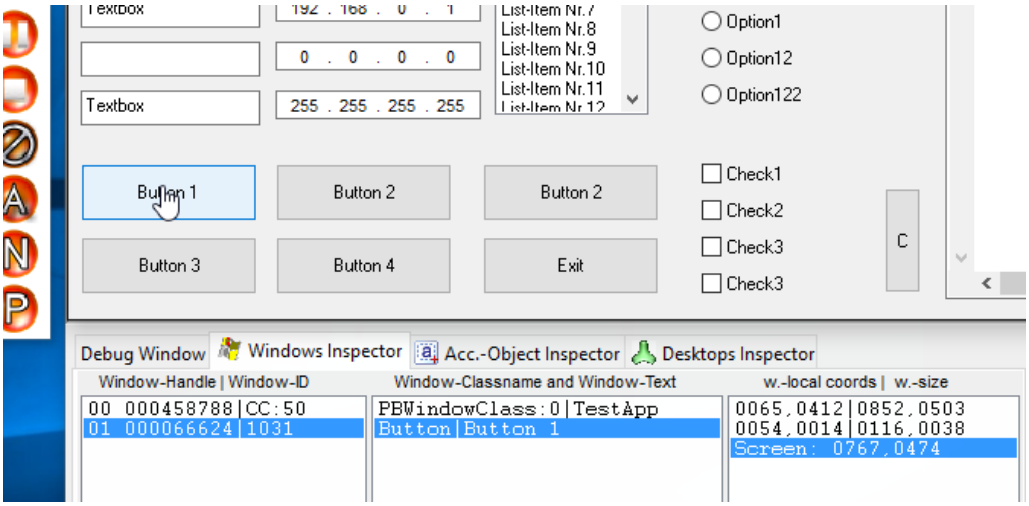
And therefore the window can be anywhere on the Screen, we will always be right.
As long, as the Button we want to click did not move or change its size.

Now what you need to know that we deal with many sorts of "windows" here.
A window is the Application window itself - we call it a "Top-Level Window".
Another windows is any of the Control or buttons here.

Not every Button must be a window. But if it is a window you will see it in the "Windows-Inspector",
see picture below. The "Windows-Inspector" also shows you the "Window-Coordinates" (right side:
- 65,412 - for the Application "Top-Level-Window" and
- 54,14 for the "Button-Control-Window"
below are the Screen coordinates (767,474).

On the right side of these coordinates you can see the size of these windows in X-Pixel,Y-Pixel.

© 2023 Theo Gottwald * https://www.smart-package.com



You can Move the Mouse using Window-Coordinates like this:

```
STW.t|TestApp
MMV.w
ENR.
```

This will move the Mouse to the upper left corner (0,0 - Position) of the last localized (Top- or Child-)Window.
You can specify own Window Coordinates as well:

```
STW.t|TestApp
MMV.w|213,200
ENR.
```

Then the Mouse will move to these "Window-relative Coordinates". There is more. You can do inline Calculations with Mouse-Coordinates:

```
STW.t|TestApp
MMV.w|(22+8),(33-5)
ENR.
```

You can also use Variables and Formulas inline, for example to paint something:

```
STW.t|TestApp
FOR.$$LOP|1|29
  MMV.w|($$LOP+8),($$LOP*2-5)
NEX.
ENR.
```

In Addition to the "w"-Prototype there are the two Prototypes "t" and "k". These will work with the last localized
"w"- Any Window (Last localized)
"t" - Top-Window
"k" -Child-Window (Button or GUI-Control)

There is one Extension to these Prototypes, that is the
"c" - Client-Area

© 2023 Theo Gottwald * https://www.smart-package.com

If you specify the "c" additionally, the Mouse will take the Client Area as Origin. That excludes the blue Part (see Picture below) for Client-Area and Non-Client-Area.
Try this and see where the Mouse goes.

```
STW.t|TestApp
MMV.wc|
ENR.
```

Client-Area, Mouse-Position, Mouse-Click and the Active Point

Generally when you select an Application-Window, the Program will internally "locate" the window and remember its coordinates.

In this picture you can see the 3 important Elements:

1. (red) The rectangle around the whole Window
2. (blue) then "Non-Client Area" that is the Area that is occupied by the Title-bar, Scrollbar and resizing gadgets.
   This part is different depending on Settings and windows-version.
3. The Centerpoint of the Window.

Simply by using the Command `STW.` - here by example

```
STW.t|Testform
```

you select an Window, and the Robot will automatically "locate it". The robot will find the window and "internally calculate the Centerpoint".
This point in the center of an located window (or object) is called the "Active Point".
You can not see the "Active Point", its just the Point where the Mouse is going to drive, or where you will Click next time.

At this time, when you use `STW.`, the Mouse does not move. The robot just internally "locates" the window and calculates the Centerpoint.

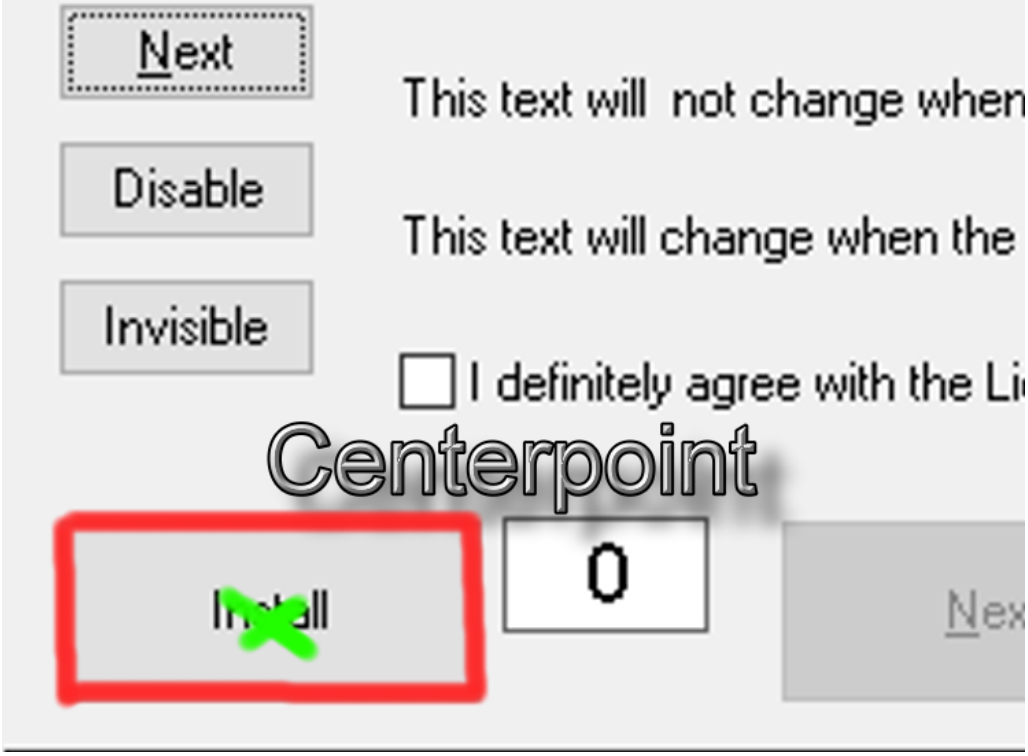
© 2023 Theo Gottwald * https://www.smart-package.com

The same is possible with a button or othe GUI-Elements. Any such Element has a surounding rectangle.
And a Centerpoint.

Using for example:

```
SCW.ct|Button|Button 1
```

you will locate this Button, and the Robot will localize its coordinates and calculate the Centerpoint.
Which is then the"Active Point".



Now you can just write

```
MMV.
```

and the Mouse will drive to the "Active Point", that is the Point in the Center of the last "localized Object".

Alltogether the Script will look as simple as this:

```
STW.t|TestApp
SCW.ct|Button|Button 1
MMV.
```

This will Move the Mouse over the Button-Centerpoint. Now there are more Options. Remember, if you use:

```
STW.t|TestApp
SCW.ct|Button|Button 1
MMV.w|10,10
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

this will move the Mouse 10,10 Points below and right the Origin (Upper,Left Corner) of the localized Button.

Move around the Centerpoint using "r"

What if you want to move 10,10 right and down of the Centerpoint?
Then you specify the "r"-Option.

```
STW.t|TestApp
SCW.ct|Button|Button 1
MMV.wr|10,10
ENR.
```

And if you want to go "Left and up"?
Then use negative Coordinates:

```
STW.t|TestApp
SCW.ct|Button|Button 1
MMV.wr|-10,-10
ENR.
```

Move Active Point using "x"

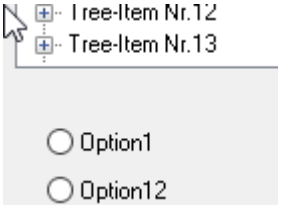
Normally when you do that, the Mouse drives there and also makes a Click if you say so,
but it does not Move the Active Point - that is the Point where the next Mouse-Action occurs.

If you want to Move the "Active Point" to that Location, specify the
"x" . Option - Move Active Point to th specified Location.

Checkmark-Position using "d" and "e"

Sometimes you do not want the Centerpoint of your Rectangle. For example see the Treeview and Checkmark below.
To click the Checkmark, you may want to click on the Left side in the vertical Middle of the Rectangular Area.
To get the Mouse there use the
- "d" Checkmark-Position Left Side
- "e" Checkmark-Position Right Side

Options.



Note the Options: "d","e","r" can not be combined.
Also these options require an Base-Rect from "w","t","k" (or "o","g","a" see below) and will not work without these.

© 2023 Theo Gottwald * https://www.smart-package.com

Other Rectangular Object

You can locate other rectangular Objects, using "Accessibilty"-Commands (called "E8") or graphical Commands (called "E9").
For these there are also Prototypes available.

"a" - last localizes Accessible Object
"g" - last localized graphical Object
"o" - any Object of these Two or the last localized window-object

Usage and Options are the same as with the window-Objects before:

```
STW.t|TestApp
AGR.43|Button 1
MMV.
ENR.
```

Also you can use the options "r","d" and "e" for these Objects. For eaxmple:

```
STW.t|TestApp
AGR.43|Button 1
MMV.ar
PAU.1
MMV.ar|10,122
ENR.
```

Point-based Objects

Unlike the Objects above, these "Sources" are not rectangular, but just "points" on the Screen.
Because they are not rectangular, there is no way to calculate an Centerpoint or to use the "d" or "e" Options.
You can anyway still correct these points with your own additional Coordinates.

These "Point-based Objects" are:

"m" - actual Mouse-Position
"b" - last Mouse-Position
"p" - Active Point
"l" - Last Active Point
and special:
"x" - set Active Point

You can not combine these with each other and also not mix these with the other Prototypes above.

```
STW.t|TestApp
AGR.43|Button 1
' Move Mouse to the Button
MMV.
' Now you have 3 Seconds time to move the mouse anywhere
PAU.3
MMV.m|10,100
' Now again you have 3 Seconds to move the Mouse around
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU.3
MMV.b
ENR.
```

The "m"-Option makes sense, if you want to act from the Point "where the mouse is".
It may be there from a robot- or from a User mouse move.

The "b"-Option is similar. It remembers the last Position, where the Robot has moved the Mouse to.
If the user moves the mouse away, you can revert it to the "last robot-position" using the "b"-Option.

The "p"-Option is the same like if you write nothing. The Mouse will just move to the current "Active Point".
The "l"-Option is the previous "Active Point". The one before the actual.
Here is an example-Skript. The Mouse will move to the Button 2, and then move into the Window-Center,
because that is the place where the last Active Point was.

```
STW.At|TestApp
AGR.43|Button 2
' Move Mouse to the Button
MMV.p
PAU.3
MMV.l
ENR.
```

All possible Mouse-Parameters:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dua

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning. These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and y
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".
```

Normally these commands just "Act" on the current "Active Point", but they do NOT Move the current "Active Point".
If you want to do so, just add the `"x"`-Prototype.

```
"x" - Set Active Point at this position.
```

Important to know:

Moving the Mouse to a Point does not automatically move the actual Point.
To do this, use additionally the "x" Parameter. This is different from earlier versions.

```
Example:

STW.At|TestApp
MPO.o|57,415
MLI.
```

This does not make sense because, if you want to Click at 57,415 in the Topwindow, you would simply write:

```
STW.At|TestApp
MLI.o|57,415
```

In earlier versions this would have clicked at 57,415 - in Top-Window-Coordinates anyway.

From now, it will do what the Script says:
- Locate the Top-Window and set the actual Point in the Center of the Top-Window.
- Move the Mouse to that Coordinate
- Click into the Actual Point (that is still in the Center of the Top-Window)

If you depend on using an not needed `MMV.` or `MPO.` you can then add the "x"-Option.
This will move the "Active Point" and have the desired effect.

```
STW.At|TestApp
MPO.ox|57,415
MLI.
```

The `MPO.` and `MMV.` does only "Move the Mouse-Pointer" it does not move the actual point.
For this use the `MAP.` - Command or the "x"-Option with `MMV./ MPO.`.

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.2 CPC. - Convert Pixel Coordinate

MiniRobotLanguage (MRL)

CPC. Command

CPC. - Convert Pixel Coordinate

Intention

The CPC.-Command helps you to convert between Coordinates that are scalend or unscaled due to Windows Size-Settings.
As you can see in the picture below, the Setting is "125%".
This will lead to a shift of Pixel-Coordinates in some dpi-aware Applications.
Generally if you make things by 125% larger, the virtual Screen-Resolution is decreased.
To get back from 125% to normal, you could use the formula: `X=X/100*125`
This is what the "Scale-Command" does internally.
To Unscale the formula is as simple: `X=X/125*100`
This is done under the hood if you choose an Unscale-command.

The robot detects the current dpi-/scaling settings at the start of the script.
Therefore changing the DPI-Scaling while the script is running may prevent the script from correct scaling.

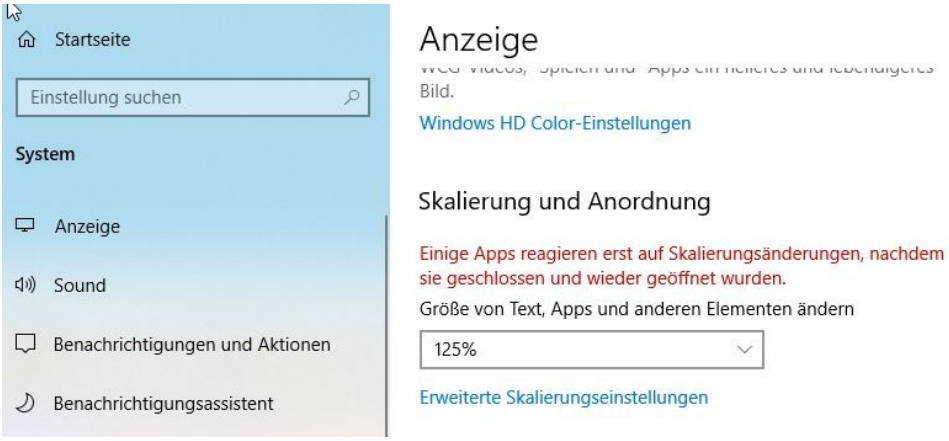
To get nearly the same effect you could just choose a lower Screen Resolution. This will make less problems with pixel-based automations.

The robot tries to automatically scale/unscale the situation, but if this does not work, you will need to manually correct the coordinates using the `CPC.` Commands.
CPC. will use scaled or unscaled input in multiple ways and scale/unscale it with the current settings.

Possible Uses are:
- Coordinate based Script was recorded in scaled Mode and now is going to be played in unscaled mode
- Coordinate based Script was recorded in 100% and is now going to be played on a Client with Scaled coordinates

As said, in many cases it will just work, buit in some cases you may need to correct the coordinates.

© 2023 Theo Gottwald * https://www.smart-package.com



What is the "Active Point"?
The "active Point" is the "Focus Point" of the robot. It is the point "On screen", where the next robot action - for example a mouse-click - will happen.

The CPC.-command can move the active point in the conversion process. For this use the "x" option.
The CPC.-command can also convert Point- and Rect-Coordinates (Variables) in one step.

A complete sequence could look like this:

```
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.nc|1|TcxButtonEdit
' This will just move the active point and put also the new
coordinates on the stack
CPC.uax
MMV.
ENR.
```

Syntax

CPC.[P1][|P2][|P3]...

Parameter Explanation

P1 – (optional) Prototype-String with any of the following options

a - Use actual Point for input coordinate
s - Scale Operation (you have the coordinate from 100% and want the coordinate from for example 125%)
u - UnScale-up Operation (you have the coordinate from for example 125% and want the coordinate from 100%)

*© 2023 Theo Gottwald * https://www.smart-package.com*

p - use Point Input Variable. This is one Variable with both coordinates separated by "," e.g. see VAP. 3560 command

r - use RECT as Input. Will also use a RECT for Output. if specified together with "a" then "Actual Rect" (from last window or object) will be taken, otherwise RECT in Variable **P2**.

x - If specified, the result of the Operation will be set as active Point/active Rect. Using Prototype "arx" for example

f - just get the scaling factor, if one Variable P" is given, result is Point-Format, if two variables are there, P2 is X and P3 is Y-Factor

Generally the Command looks how many Output and Input Variables are there and tries to find out, if the result should be given as Point or in two variables as coordinates.

Using the "r"-Rect Option, you can also specify 4 variables to get each coordinate in one variable (left,top,right,bottom).

P2 .. PX – (optional) Depending on the used Prototypes, **P2 ... P5** may be Input or Output-Variables.

Example

```
'**********************************
' CPC.
'**********************************
GSB.Lab_Get_Factor

' Otion "f"
§§STA=50
' Resultat zum Beispiel 1.25 - 1.25
CPC.f|$$XSD,$$YSD
JIV.$$XSD!$$XAL|ero
JIV.$$YSD!$$YAL|ero

§§STA=60
' Option "fp" mit Point-Variable
CPC.f|$$XSA
%Point_to_Vars $$XSA|$$XSD|$$YSD
JIV.$$XSD!$$XAL|ero
JIV.$$YSD!$$YAL|ero

' Option "u" und "s" mit extra Rückgabevariablen
§§STA=70
GSB.Vorbereiten
CPC.u|$$XPA,$$YPA|$$XSD,$$YSD
'MBX.$$XSD,$$YSD --- $$XPA,$$YPA --- $$XPB,$$YPB
JIV.$$XSD!$$XPB|ero
JIV.$$YSD!$$YPB|ero

§§STA=80
GSB.Vorbereiten
CPC.s|$$XPA,$$YPA|$$XSD,$$YSD
JIV.$$XSD!$$XPC|ero
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
JIV.$$YSD!$$YPC|ero

' Option "u" und "s" ohne extra Rückgabevariablen
§§STA=90
GSB.Vorbereiten
CPC.u|$$XPA,$$YPA
JIV.$$XPA!$$XPB|ero
JIV.$$YPA!$$YPB|ero

§§STA=100
GSB.Vorbereiten
CPC.s|$$XPA,$$YPA
JIV.$$XPA!$$XPC|ero
JIV.$$YPA!$$YPC|ero

' Option "u" und "s" mit POINT und extra Rückgabevariablen
§§STA=110
GSB.Vorbereiten
CPC.up|$$POA|$$XSD
JIV.$$XSD!$$POB|ero

§§STA=120
GSB.Vorbereiten
CPC.sp|$$POA|$$XSD
JIV.$$XSD!$$POC|ero

' Option "u" und "s" mit POINT und ohne Rückgabevariablen
§§STA=130
GSB.Vorbereiten
CPC.up|$$POA
JIV.$$POA!$$POB|ero

§§STA=140
GSB.Vorbereiten
CPC.sp|$$POA
JIV.$$POA!$$POC|ero

' Option "u" und "s" mit POINT und ohne Rückgabevariablen und
"x"
§§STA=130
GSB.Vorbereiten
CPC.upx|$$POA
JIV.$$POA!$$POB|ero
MMV.
GMP.ms|$$POM
CPC.sp|$$POM
JIV.$$POA!$$POB|ero

§§STA=140
GSB.Vorbereiten
CPC.spx|$$POA
JIV.$$POA!$$POC|ero
MMV.
GMP.ms|$$POM
CPC.sp|$$POM
JIV.$$POA!$$POC|ero
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.READY
ENR.

'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Lab_passed
SAV.Restore
$$RES=108
ENR.
'-------------------------------------------------------------
:Vorbereiten
IVV.$$XAL=0
  GSB.Lab_Get_Factor
EIF.
VAN.$$XPA=300
VAN.$$YPA=250
VAP.$$POA=$$XPA,$$YPA
' Unscaled
CAL.$$XPB=300/$$XAL*100
CAL.$$YPB=250/$$YAL*100
VAP.$$POB=$$XPB,$$YPB
' Scaled
CAL.$$XPC=300/100*$$XAL|i
CAL.$$YPC=250/100*$$YAL|i
VAP.$$POC=$$XPC,$$YPC

'MBX.Scaled: $$POC   Unscaled: $$POB
RET.
'-------------------------------------------------------------
:ero
' Fehler aufgetreten
$$ERO=§§STA
MBX. Fehler in $script$ $crlf$bei $$ERO
$$RES=0
ENR.
'-------------------------------------------------------------
:Lab_Get_Factor
DPI.s|$$XSC,$$YSC
CAL.$$XAL=100*$$XSC
CAL.$$YAL=100*$$YSC
RET.
'-------------------------------------------------------------
' Convert Point-Variable to 2 Variables

: %Point_to_Vars 3
VAO.§§§02
VAO.§§§03
GSB.Lab_PTV|§§§01|§§§02|§§§03
END%

:Lab_PTV
SAV.Save|$$P01|$$P02|$$P03
VAP.$$P01=§§_01
GES.b|$$P01|,|$$P02
GES.d|$$P01|,|$$P03
VAI.§§_02=$$P02
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAI.$$_03=$$P03
SAV.Restore
RET.
```

Remarks

-

Limitations:

-

See also:

- LBO. - Left Border 2506
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.3 Automated Clicks

3.65.3.1 BCS. - Button Click Special

MiniRobotLanguage (MRL)

BCS. Command

Button Click Special

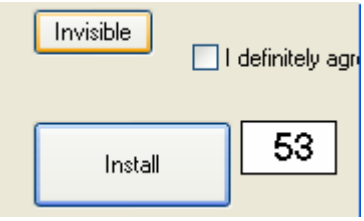
Intention

These command simulates a Left-Click with the Mouse in one of several ways.
The most important difference from BSC. to other Commands is, that it will watch if the Click "will do something".
And will click again if it did not do something. It's a Click in an built in Loop.
By default, BCS. want the Button or Window that we are clicking, to disappear.
To be closed, to have an invalid Window-Handle. As it happens when you close a Window.

`BCS.` will click, then wait a short time.
Next the Target button/window is checked on a reaction to the Mouse-click.
If there is no reaction, `BCS.` will click again, until there is an reaction.
Here is an example:

```
' Start the BCS.-Testapp from your Sample-Apps folder
STW.wct|45|PBWindowClass:0|BCS - Testform
SCW.wctid|45|Button|Install|1012|1
SFW.
BCS.
```

This is what you get:



`BCS.` just keeps clicking on "Install". Because the window does not disappear.
The counter on the right side increases each time, when the button is clicked, in this case it shows 53 and `BCS.` will not stop if we don't stop it.

For example, the Target window should disappear shortly after the Mouse-Click.
Or the text on a button can change after a mouse-click.

If the specified reaction of the window does not occur, BCS. will repeat the action.
`BCS.`is somehow a "Mouse-click in a Loop".

*© 2023 Theo Gottwald * https://www.smart-package.com*

BCS. is used in automated Software installations, to make sure things work as reliable as possible.

To test the features of BCS. you can use a provided BCS.-Testapplication in the "TestApps folder" in the installation folder.

Syntax

BCS.[Prototype ...][|Params ...]

Parameter Explanation

Prototype - (optional) prototype values

```
Prototype - (optional) prototype coordinates
Parameter - (optional) Parameter/timeout
```

```
   Prototype can be omitted. Alternatively you can use these:
   a - Button must become invisible or disabled
   e - Button must become disabled
   h - Parameter is Window-handle
   p - clicks until Topwindow disappears
   t - timeout in ms [+PARAMETER]
   v - Button must become invisible
   s - Button-Text must change (works only with Buttons where
   CFW. can read
        the text.)
   n - Max-Count of Clicks before Timeout is generated.
        [+PARAMETER]
   w - time between clicks in ms. Default is 100 ms


   l - MLE. - Mouse Left Event Click is used
   o - MLC.
   i - MLI. - Like MLC. but much faster
   c - CLW. - Close Window is used (Window-Message)
   r - MRM. - Compatible right Click.
   m - MLM. - Compatible like MLC. but faster due to a Jump.
   u - MDC. Double Click
   Mixed:
   q - MLM. and MLE. Clicks are used.

   Adding multiple Prototypes, all of the choosen Clicks were
   used.
   Example: BCS.lr will do all two Clicks at the same time.
   Option "q" will do exchange the two Click-Types and do one
   after the other.
```

Parameter - (optional) Parameter/timeout

© 2023 Theo Gottwald * https://www.smart-package.com

Some Prototypes needs a Parameter, they are marked with [+PARAMETER].

Example

```
'*********************************************
' BCS.-Sample
'*********************************************
' This will create some messageboxes in the background.
FOR.§§LOP|1|8
PRR.
:ag
MBX.!
PRT.Ending.
END.
PRE.
NEX.
PAU.1
'-------------------------
' Here comes the BCS.
' This will click away the generated Msg.-Boxes
FOR.§§LOP|1|8
STW.t|Message
SCW.c|button
BCS.
' BCS.pw|250
NEX.
MBX.Ende
END.
'*********************************************
```

Remarks

BCS. is provided to make automated tasks as reliable as possible. If you experience problems in automated Setup tasks, especially, that buttons are clicked too often. Look if there are BCS. involved and replace them with MLE. or use other options.

BCS. can be used best, when the window that is clicked will and must disappear after the click. If this is not the case, BCS. may click a button to often and should maybe not used at that place.

Limitations:

-

See also:

- CAW. - Close-All-Windows 3586

© 2023 Theo Gottwald * https://www.smart-package.com

- BSC. - Button Special Click 2435
- MLE. - Mouse-Left-Click Event 2492
- DBC. - Do-Button-Clicks 2863
- **OPT. - Optional Settings** 2406

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.65.3.2 BSC. - Button Special Click

MiniRobotLanguage (MRL)

BSC. Command

Button Special Click

Intention

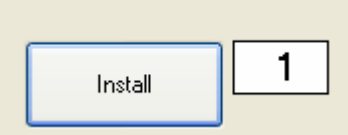
These command simulates a Left-Click with the Mouse in one of several ways.
The most important difference from BSC. to other Commands is, that it will watch if the Click "will do something".
And will click again if it did not do something. It's a Click in an built in Loop.
By default, BSC. will watch, if the Button or Window changes its Text, gets invisible, gets disabled or will disappear.

BSC. will click, then wait a short time.
Next the Target button as well as the target window, is checked on a reaction to the Mouse-click.

If there is no reaction, BSC. will click again, until there is an reaction.
Here is an example:

```
' Start the BCS.-Testapp from your Sample-Apps folder
STW.wct|45|PBWindowClass:0|BCS - Testform
SCW.wctid|45|Button|Install|1012|1
SFW.
BSC.
```

This is what you get:



BCS. just clicked one time on "Install". While the window did not disappear, the counter on the right side changed. and that made BSC. to stop clicking.

BSC. will react on changes in the text of the target form. As well as on anything that BCS. would react to.

If the specified reaction of the window does not occur, BSC. will repeat the action.
BSC.is somehow a "Mouse-click in a Loop".

BSC. is used in automated Software installations, to make sure things work as reliable as possible.

To test the features of BSC. you can use a provided BCS.-Testapplication in the "TestApps folder" in the installation folder.

*© 2023 Theo Gottwald * https://www.smart-package.com*

You can also use it on the normal TestApp_01.exe and see the results.

Syntax

BSC.[Prototype ...][|Params ...]

Parameter Explanation

Prototype - (optional) prototype values

```
Prototype - (optional) prototype coordinates
Parameter - (optional) Parameter/timeout

   Prototype can be omitted. Alternatively you can use these:
   a - Button must become invisible or disabled
   e - Button must become disabled
   h - Parameter is Window-handle
   p - clicks until Topwindow disappears
   t - timeout in ms [+PARAMETER]
   v - Button must become invisible
   s - Button-Text must change (works only with Buttons where
   CFW. can read
        the text.)
   n - Max-Count of Clicks before Timeout is generated.
        [+PARAMETER]
   w - time between clicks in ms. Default is 100 ms


   l - MLE. - Mouse Left Event Click is used
   o - MLC.
   i - MLI. - Like MLC. but much faster
   c - CLW. - Close Window is used (Window-Message)
   r - MRM. - Compatible right Click.
   m - MLM. - Compatible like MLC. but faster due to a Jump.
   u - MDC. Double Click
   Mixed:
   q - MLM. and MLE. Clicks are used.

   Adding multiple Prototypes, all of the choosen Clicks were
   used.
   Example: BCS.lr will do all two Clicks at the same time.
   Option "q" will do exchange the two Click-Types and do one
   after the other.
```

Parameter - (optional) Parameter/timeout

Some Prototypes needs a Parameter, they are marked with [+PARAMETER].

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*********************************************
' BSC.-Sample
'*********************************************
' This will create some messageboxes in the background.
FOR.§§LOP|1|8
PRR.
:ag
MBX.!
PRT.Ending.
END.
PRE.
NEX.
PAU.1
'-------------------------
' Here comes the BCS.
' This will click away the generated Msg.-Boxes
FOR.§§LOP|1|8
 STW.t|Message
 SCW.c|button
 BSC.
NEX.
MBX.Ende
END.
'*********************************************
```

Remarks

BSC. is provided as an intelligent mouse-click, that will make automated tasks as reliable as possible. If you experience problems in automated Setup tasks, especially, that buttons are clicked too often. Look if there are BSC. involved and replace it with MLE. or use other options in BSC.

BSC. can be used best, when the window that is clicked will change its text after the click. But also if it will disappear after the click. If this is not the case, BSC. may click a button to often and should maybe not used at that place.

Limitations:

-

See also:

- CAW. - Close-All-Windows 3586
- BCS. - Button Click Special 2431
- MLE. - Mouse-Left-Click Event 2492
- DBC. - Do-Button-Clicks 2863

© 2023 Theo Gottwald * https://www.smart-package.com

- OPT. - Optional Settings 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4 Compatible Engine

3.65.4.1 GMP. - Get-Mouse-Position

MiniRobotLanguage (MRL)

GMP. Command

Get Mouse Position

Intention

Returns the position of the Mouse-Pointer or of the internal "Actual Point" which is not equal.
Depending on the used prototypes the command gives the coordinates in screen or window relative.

P1 is mostly two letter. First specify if you want the Mousepointer-Position or the "actual point".

Example:

```
GMP.mw|$$RET
```

or if you specify "k" then this command requires two variables for holding the resulting `x` and `y` data.

```
GMP.mwk|$$REX,$$REY
```

In case you want the "actual point" instead of the "Mouse pointer position", just use the "a" directive instead of the "m".

Syntax

GMP.P1|P2[,P3]

Parameter Explanation

P1

```
These are the source-prototypes

 a - get "actual point" in Screencoords if no "c" or "t"
Option is used
 m - get mouse-position
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

If whether "a" nor "m" is specified, "m" is the default.
the following prototypes define the coordinate system being used for the result:

```
 s - screen coordinates (default)
 t - get point in actual Top-window coordinates
 c - get point in actual window coordinates
 w - last (whatever) window-coordinates
```

Additional you can specify

```
 r - client coordinates of the selected window
 k - output to 2 variables instead of one
```

P2 **-** Variable for result or if "k" is used, for result #1, the x-coordinate
P3 **-** Variable for result #2, the y-coordinate

Example

```
'**********************************
' Example 1
' with 2 data variables for result
'**********************************
' Get Mousepointer
GMP.km|§§XPO,§§YPO
PRT. Mouse is at: §§XPO,§§YPO
DMP.


:over
MBX.Halt to read output
END.


'**********************************
' Example 2
'**********************************
' Get Mousepointer
GMP.§§MPO
PRT. Mouse is at: §§MPO
DMP.
END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- MMV - Mouse MoVe 2466
- MPO. - Mouse-POsition move
- LAP. - Locate at Point 599
- GAI. - Get-Acc-Information 591

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.2 MDC. - Mouse Double-left Click (Invinsible)

MiniRobotLanguage (MRL)

MDC. Command

Mouse Double Click (compatible engine with move)

Intention

`MDC.` simulates a Mouse-Move to, and a Double-Click at the target Position.

This Command simulates a Left-Double Click with the Mouse, using the Compatible Engine 231.
This command will simulate a real Mouse-Move, even with a "Human touch", if this is been set, using

```
OPT.HM|35
```

This is a are very realistic Mouse Move and a Mouse Double-Click at the target.

`MDC.` has therefore all the strong and all the weak sides of a typical Compatible Engine 231 command.

Then the mouse pointer will move to the current "Active Point" and do the Click.
In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Some Hints:
If you need just one Compatible Engine 231 click, use `MLC.`
If you prefer the mouse to "Jump" instead of slowly moving to the target, use `MDI.`
In case you can use the Direct Engine 231, `MDE.` is the command for you..

Syntax

```
MDC.[P1|][P2,P3]
```

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:
GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".
```

Normally these commands just "Act" on the current "Active Point", but they do NOT
Move the current "Active Point".
If you want to do so, just add the `"x"`-Prototype.

```
"x" - Set Active Point at this position.

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** **-** (optional) x,y - coordinates

```
     numeric coordinates, separated by comma. You can use
     variable and Formula in (..). You can also use a
     Point-Variable instead of two coordinates.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' MDC.-Demo
'**********************************
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
MDC.p|(11*5),(-1*5)
ENR.
END.
```

Remarks

Mostly you do not need to specify the coordinates with `MMV.` or `MLI.` but you can directly write the coordinates into `MLC.`, `MLI.`, `MRC.` and MRI. Please see also `BCS.`

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MDE. - Mouse DoubleCLick Event 2487
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.3 MDI. - Mouse Double-Click Invisible

MiniRobotLanguage (MRL)

MDI. Command

Mouse Double Click Invisible (compatible engine with jump)

Intention

`MDI.` simulates a Mouse-Jump to, and a Double-Click at the target Position.

This Command simulates a Left-Double Click with the Mouse, using the Compatible Engine 231.
This command will simulate a Mouse-Jump. In opposite to a Mouse-Move, the Mouse will jump to the target, click and then jump back to the previous position.

`MDI.` uses the Compatible Engine 231 for the Mouse-Click. It's faster then `MDC.` as the Mouse will not move slowly but Jump to the target.

Using `MDI.` the mouse pointer will jump to the current "Active Point" and do the Double-Click. In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here: MAP - Move Active Point 2483.

Some Hints:

- If `MDI.` seems not to work on your target window, try `MDC.` `MDC.` is a bit more compatible, for those applications that need a Mouse-move at the target position before the click.
- In case you can use the Direct Engine 231, `MDE.` is the command for you..

Syntax

```
MDI.[P1|][P2,P3]
```

Parameter Explanation

P1 – (optional) prototype coordinates

```
Prototype can be omitted or any of:
```

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'**********************************
' MDI.-Demo
'**********************************
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MDI.p|(11*5),(-1*5)
ENR.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MDE. - Mouse DoubleCLick Event 2487
- MDC - Mouse Double-left Click (Invinsible) 2442
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.4 MLC. - Mouse-Left-Click (Invisible)

MiniRobotLanguage (MRL)

MLC. Command

Mouse Left Click (compatible engine with move)

Intention

`MLC.` simulates a Mouse-Move to, and a Left-Click at the target Position.

This Command simulates a Left-Click with the Mouse, using the Compatible Engine 231.

The usage of the command is simple, you can just write:

```
MLC.
```

Then the mouse pointer will move to the current "Active Point" and do the Click.
In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

This command will simulate a real Mouse-Move, even with a "Human touch", if this is been set, using

```
OPT.HM|35
```

This is a are very realistic Mouse Move and a Mouse Left-Click at the target.

`MLC.` has therefore all the strong and all the weak sides of a typical Compatible Engine 231 command.

`MLC.` is a real Mouse-Click, which need the window to be on top of other windows.
Also it will need a mouse-pointer and a desktop.

You can find some more details on the "r"-prototype here: MAP - Move Active Point 2483.

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

MLC.[P1|][P2,P3]

Parameter Explanation

P1 - (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.

"x" - Set Active Point to the Command-position.

Example:
' Move Mouse and Active Point
MLC.wrx|-358,207
```


© 2023 Theo Gottwald * https://www.smart-package.com

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'**********************************
' MLC.-Demo
'**********************************
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|Install
MLC.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.5 MLD. - Mouse left down

MiniRobotLanguage (MRL)

MLD Command

Mouse Left Down (compatible engine with move)

Intention

For Drag'n Drop operations, you may need to just press a mouse-button down.
Then move the mouse and release later it.

There are two commands for this purpose:

- `MLD.` - Compatible Engine 231
- `MLU.` - Compatible Engine 231

These command should only be used paired, as otherwise you may end up with a "pressed left mouse-button". Using `MLD.` the left Mouse-Button will be pressed until you physical press and release it again, or until you make another left mouse-click.

You can use `MLD.` and `MLU.` for Drag'n Drop operations in programs or on your desktop.

Using `MLD.`, Then the mouse pointer will move to the current "**Active Point**" and do the click-down using thew the left mouse-button.
In most cases the "Active Point" is been set, by localizing a button or control or object.
In these cases, the "Active Point" is in the center of the rectangular button/window/object.
For Drag'n Drop operations, you may want to move the "Active Point" manually. You can doi that easily, using the provided prototypes.

You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Syntax

MLD .

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
Prototype can be omitted or any of:
```

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'*****************************
' Drag'n Drop Sample
'*****************************
' This script will drag an desktop-
' Icon named "Furmark" to the right
' and drop it then
STW.ct|Progman|Program Manager
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SCW.nct|1|SHELLDLL_DefView|{&NOTEXT:}
SCW.nct|1|SysListView32|FolderView
SAO.rn|34|FurMark
MLD.
MMV.r|300,100
MLU.
ENR.
```

Remarks

The alternative to this command is `MDM.` which uses the Direct Engine 231. It can not be used for Drag'n Drop operations.

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLU. - Mouse left up 2463
- MDM. - Mouse Down Message 2490
- MUM. - Mouse Up Message 2498
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.6 MLI. - Mouse Left Click quIckly (Invinsible)

MLI. - Mouse-Left-click quIckly (Invisible) **Previous** 2492 **Top** 41 **Next** 2457

MiniRobotLanguage (MRL)

MLI. Command

Mouse Left Click Invisible (compatible engine with jump)

Intention

`MLI.` simulates a Mouse-Jump to, and a Left-Click at the target Position.
The mouse-pointer will then jump back to where it was before the click.

This Command simulates a Left-Click with the Mouse, using the Compatible Engine 231.
This command will simulate a Mouse-Jump. In opposite to a Mouse-Move, the Mouse will jump to the target, click and then jump back to the previous position.

`MLI.` uses the Compatible Engine 231 for the Mouse-Click. It's faster then `MLC.` as the Mouse will not move slowly but Jump to the target.

Using `MLI.` the mouse pointer will jump to the current "Active Point" and do the Left-Click. In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Some Hints:

- If `MLI.` seems not to work on your target window, try `MLC.` or `MLM. MLC.` and `MLM.` are a bit more compatible, for those applications that need a Mouse-move at the target position before the click.
- In case you can use the Direct Engine 231, `MLE.` is the command for you..

Syntax

MLI.[P1|][P2,P3]

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.

"x" - Set Active Point to the Command-position.

Example:
' Move Mouse and Active Point
MLC.wrx|-358,207
```

P2, ***P3*** **-** (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' MLI.-Demo
'**********************************
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
MLI.r|(11*5),(-1*5)
ENR.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLC - Mouse-Left-Click (Invisible) 2448
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MDI. - Mouse Double-Click Invisible 2445
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.7 MLM. - Mouse Left Click Move (Invinsible)

MiniRobotLanguage (MRL)

MLM. Command

Mouse Left click Move (compatible engine with jump and small move)

Intention

`MLM.` simulates a Mouse-Jump to, and a Left-Click at the target Position.
Unlike `MLI.` the mouse-pointer will not jump back to where it was before the click.

`MLM` is somehow like an `MLI.` with a small Mouse-Move at the target position.
It can be used where `MLC.` works. but `MLI.` does not.

This Command simulates a Left-Click with the Mouse, using the Compatible Engine 231.
This command will also simulate a Mouse-Jump. In opposite to a Mouse-Move, the Mouse will jump to the target, click and remain in the target position.

`MLM.` uses the Compatible Engine 231 for the Mouse-Click. It's faster then `MLC.` as the Mouse will not move slowly, but Jump to the target.

Using `MLM.` the mouse pointer will jump to the current "Active Point" and do the Left-Click. In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Some Hints:

- For automated installations use `BSC. and BCS.` where possible.
- In case you can use the Direct Engine 231, `MLE.` is the command for you..

Using `MLM.` you get a real Mouse-Click, which needs the window to be on top of other windows, unlike if you use Mouse-Events (See `MLE.`).

Syntax

```
MLM.[P1|][P2,P3]
```

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** **-** (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'*********************************
' MLM.-Demo
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|Install
MLM.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLC - Mouse-Left-Click (Invisible) 2448
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.8 MRM. - Mouse Right Click Move (Invinsible)_2

MLM. - Mouse-Left-click Move (Invisible) **Previous** 2454 **Top** 41 **Next** 2463

MiniRobotLanguage (MRL)

MRM. Command

Mouse Left click Move (compatible engine with jump and small move)

Intention

`MRM.` simulates a Mouse-Jump to, and a Right-Click at the target Position.
Unlike `MRI.` the mouse-pointer will not jump back to where it was before the click.

`MRM.` is somehow like an `MRI.` with a small Mouse-Move at the target position.
It can be used where `MRC.` works. but `MRI.` does not.

This Command simulates a Left-Click with the Mouse, using the Compatible Engine 231.
This command will also simulate a Mouse-Jump. In opposite to a Mouse-Move, the Mouse will jump to the target, click and remain in the target position.

`MRM..` uses the Compatible Engine 231 for the Mouse-Click. It's faster then `MLC.` as the Mouse will not move slowly, but Jump to the target.

Using `MRM..` the mouse pointer will jump to the current "Active Point" and do the Left-Click. In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Using `MRM..` you get a real Mouse-Click, which needs the window to be on top of other windows, unlike if you use Mouse-Events (See `MRE.`).

Syntax

MRM..[P1|][P2,P3]

Parameter Explanation

P1 – `(optional) prototype coordinates`

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
Prototype can be omitted or any of:
```

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'**********************************
' MRM..-Demo
'**********************************
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|Install
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MRM.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLC - Mouse-Left-Click (Invisible) 2448
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.9 MLU. - Mouse left up

MiniRobotLanguage (MRL)

MLU Command

Mouse Left Up (compatible engine with move)

Intention

For Drag'n Drop operations, you may need to just press a mouse-button down. Then move the mouse and release later it.

There are two commands for this purpose:

- `MLD.` - Compatible Engine 231
- `MLU.` - Compatible Engine 231

These command should only be used paired, as otherwise you may end up with a "pressed left mouse-button". Using `MLD.` the left Mouse-Button will be pressed until you physical press and release it again, or until you make another left mouse-click.

You can use `MLD.` and `MLU.` for Drag'n Drop operations in programs or on your desktop.

Using `MLU.`, Then the mouse pointer will move to the current "**Active Point**" and do the release the left mouse-button.
In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object. For Drag'n Drop operations, you may want to move the "Active Point", which is easy using the provided prototypes. You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Syntax

MLU .

Parameter Explanation

`Prototype can be omitted or any of:`

GROUP I (based on rectangular Area) Prototypes

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'****************************
' Drag'n Drop Sample
'****************************
' This script will drag an desktop-
' Icon named "Furmark" to the right
' and drop it then
STW.ct|Progman|Program Manager
SCW.nct|1|SHELLDLL_DefView|{&NOTEXT:}
SCW.nct|1|SysListView32|FolderView
SAO.rn|34|FurMark
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MLD.
MMV.p|300,100
MLU.
ENR.
```

Remarks

The alternative to this command is `MUM.` which uses the Direct Engine[231]. It can not be used for Drag'n Drop operations.

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine[231]
- MLD. - Mouse left down[2451]
- MDM. - Mouse Down Message[2490]
- MUM. - Mouse Up Message[2498]
- MLE. - Mouse-Left-Click Event[2492]
- MLM - Mouse Left click Move (Invinsible)[2457]
- MLI - Mouse Left Click quIckly (Invinsible)[2454]
- MRC. / MRI. - Mouse-Right-Click[2472]
- MMV - Mouse MoVe[2466]
- **OPT. - Optional Settings**[2406]

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.10 MMV. - Mouse MoVe

MiniRobotLanguage (MRL)

MMV. Command

Mouse MoVe (compatible engine and direct engine with move)

Intention

This Command move the mouse from the current position to the current "active point".

What is the "Active Point"?
The "active Point" is the "Focus Point" of the robot. It is the point "On screen", where the next robot action - for example a mouse-click - will happen.

How do i set the "active point"?
Normally you locate a button, window or object. These are all rectangular areas. The robot will set the "active point" to the mIdpoint of these areas.

Can i influence the "active point" in any way?
This is where `MMV.` enters the game. Using `MMV.` or `MPO.` or `MAP.` you can set or move the "**active point**". In case you use `MMV.` with parameters, `MMV.` will not only move the mouse, but also move the "Active Point".

The difference between these commands is, that

- `MAP.` will only move the "active point"
- `MPO.` will also jump with the mouse to the new "active point"
- `MMV.` will make a mouse-move to the new "active point"

`MMV.` will use the "Human mouse move" feature if it is activated. In that case it will not move the mouse in a straight way, but simulate a "human like" mouse move. To switch this feature off, use the **OPT. - Optional Settings** 2406 command.

In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

MMV.[P1|][P2,P3]

Parameter Explanation

P1 - (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' MMV./MPO. -Demo
'**********************************
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|Install
MMV.
END.
```

Remarks

Mostly you do not need to specify the coordinates with MMV2466. because you can directly write the coordinates into the Mouse-Click commands, like these following:
MLC - Mouse-Left-Click (Invisible)2448
MLI - Mouse Left Click qulckly (Invinsible)2454
MRC. / MRI. - Mouse-Right-Click2472
MLM - Mouse Left click Move (Invinsible)2457
and others.

MMV. is also good for debugging, as it will just move the Mouse to the target area, and will not click any button.

Limitations:

See also:

- GMP. - Get-Mouse-Position2439
- MAP - Move Active Point2483
- MPO. - Mouse-POsition move2469
- **OPT. - Optional Settings**2406
- 1.C Direct Engine and Compatible Engine231
- MLE. - Mouse-Left-Click Event2492
- MLM - Mouse Left click Move (Invinsible)2457
- MLI - Mouse Left Click qulckly (Invinsible)2454
- MRC. / MRI. - Mouse-Right-Click2472
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.11 MPO. - MousePosition

MiniRobotLanguage (MRL)

MPO. Command

Mouse Position Move

Intention

This Command will make the mouse jump immediately from the current position to the current "active point".

What is the "Active Point"?
The "active Point" is the "Focus Point" of the robot. It is the point "On screen", where the next robot action - for example a mouse-click - will happen.

How do i set the "active point"?
Normally you locate a button, window or object. These are all rectangular areas. The robot will set the "active point" to the mIdpoint of these areas.

Can i influence the "active point" in any way?
This is where `MPO.` enters the game. Using `MMV.` or `MPO.` or `MAP.` you can set or move the "**active point**". In case you use `MPO.` with parameters, `MPO.` will not only move the mouse, but also move the "Active Point".

The difference between these commands is, that

- `MAP.` will only move the "active point"
- `MPO.` will also jump with the mouse to the new "active point"
- `MMV.` will make a mouse-move to the new "active point"

In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Syntax

```
MPO.[P1|][P2,P3]
```

Parameter Explanation

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** **-** (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Speed in Ticks:
This command uses typically around 900.000 Ticks.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' MMV./MPO. -Demo
'**********************************
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|Install
MPO.
END.
```

Remarks

Mostly you do not need to specify the coordinates with MPO. because you can directly write the coordinates into the Mouse-Click commands, like these following:
MLC - Mouse-Left-Click (Invisible) 2448
MLI - Mouse Left Click qulckly (Invinsible) 2454
MRC. / MRI. - Mouse-Right-Click 2472
MLM - Mouse Left click Move (Invinsible) 2457
and others.

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.12 MRC. - Mouse-Right-Click

MiniRobotLanguage (MRL)

MRC. Command

Mouse Right Click (Invinsible)

Intention

`MRC.` simulates a Mouse-Move to, and a Right-Click at the target Position.

This Command simulates a Right-Click with the Mouse, using the Compatible Engine 231.
It can be used to call all sorts of PopUp-Menus. To answer the upcoming popup-menus, see the SMH. - Set Menu Hook 660 command.

The usage of the command is simple, you can just write:

```
MRC.
```

Then the mouse pointer will move to the current "Active Point" and do the Click.
In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here: MAP - Move Active Point 2483.

This command will simulate a real Mouse-Move, even with a "Human touch", if this is been set, using

```
OPT.HM|35
```

This is a are very realistic Mouse Move and a Mouse Right-Click at the target.

`MRC.` has therefore all the strong and all the weak sides of a typical Compatible Engine 231 command.

`MRC.` is a real Mouse-Click, which needs the window to be on top of other windows.
Also it will need a mouse-pointer and a desktop.

You can find some more details on the "r"-prototype here: MAP - Move Active Point 2483.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

MRC.[P1|][P2,P3]

Parameter Explanation

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, ***P3*** **-** (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use
variable and Formula in (..). You can also use a

© 2023 Theo Gottwald * https://www.smart-package.com

```
     Point-Variable instead of two coordinates.
```

Mostly you do not need to specify the coordinates with MRC. or MRI. but you locate the position with STW., SCW., SAO. or MPO. etc.

Example

```
'*****************************************
' SMH./MRC._Sample
'*****************************************
' Select the TestApp
STW.ct|PBWindowClass:0|TestApp

' Select the Textbox
SCW.nc|3|edit

' Tell Hook that soon a Menu will pop up, he should click
' Einfügen (engl. "Paste")
SMH.Einfügen

' Now we move the Mouse to the Textbox and click the
' right Mousebutton
MRC.

' This Pause is needed, because if the script ends, the SMH.
' will also end, before the Menu is been clicked!
PAU.2

ENR.
```

Remarks

MRC / MRI. is the right Command to open a PopUp-Menu, because this may not work in all cases with MRE as expected.

Limitations:

See also:

- 1.C Direct Engine and Compatible Engine 231
- SMH. - Set Menu Hook 660
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457

© 2023 Theo Gottwald * https://www.smart-package.com

- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.13 MRI. - Mouse Right Click Invisible

MiniRobotLanguage (MRL)

MRI. Command

Mouse Right Click Invinsible

Intention

`MRI.` simulates a Mouse-Jump to, and a Right-Click at the target Position. After the Click, the Mouse jumps back to her original position where she was before the `MRI.`.

In short MRI. does this:

1. Jump to target (active point)
2. Click left mouse button via Compatible Engine 231
3. Jump back to original position

`MRI.` is somehow between the `MRC.` that does a "slow" mouse-move, and the fast `MRE.` which does not use the mouse.

`MRI.` simulates a Right-Click with the Mouse, using the Compatible Engine 231.

It can be used to call all sorts of PopUp-Menus. To answer the upcoming popup-menus, see the SMH. - Set Menu Hook 660 command.

The usage of the command is simple, you can just write:

```
MRI.
```

Then the mouse pointer will move to the current "Active Point" and do the Click.
In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

This command will simulate a real Mouse-Move, even with a "Human touch", if this is been set, using

```
OPT.HM|35
```

This is a are very realistic Mouse Move and a Mouse Right-Click at the target.

`MRI.` has therefore most the strong and all the weak sides of a typical Compatible Engine 231 command.

*© 2023 Theo Gottwald * https://www.smart-package.com*

MRI. is a real Mouse-Click, which needs the window to be on top of other windows.
Also it will need a mouse-pointer and a desktop.

You can find some more details on the "r"-prototype here: MAP - Move Active Point2483.

Syntax

MRI.[P1|][P2,P3]

Parameter Explanation

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)
"u" - Move down by the size of the Windows-Titlebar
"v" - Move up by the size of the Windows-Titlebar

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```


© 2023 Theo Gottwald * https://www.smart-package.com

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Mostly you do not need to specify the coordinates with MRC. or MRI. but you locate the position with STW., SCW., SAO. or MPO. etc.

Example

```
'**********************************
' MRC.-Demo
'**********************************
' Click inside Internet Explorer
STW.ct|ieframe|microsoft
MRI.
END.
```

Remarks

MRC / MRI. is the right Command to open a PopUp-Menu, because this may not work as expected in all cases with MRE..

Limitations:

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431
-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.4.14 MWR. - Mouse Wheel Roll

MiniRobotLanguage (MRL)

MWR. Command

Mouse Wheel Roll (compatible engine)

Intention

`MWR.` simulates a Mouse-Wheel Scroll forward or backward.
This Command simulates the Mouse Wheel using the Compatible Engine 231.
Therefore the Mouse must be positioned over the target application window.

The usage of the command is simple, you can just write:

```
MPO.
MWR.120
or
MWR.-120
```

Then the mouse wheel will do the scroll either forward (120) or backward (-120).

In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.

You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

This command will simulate a real Mouse-Wheel.
Therefore the Mouse pointer needs to be over the target window.

`MWR--.` has therefore all the strong and all the weak sides of a typical Compatible Engine 231 command.

`MWR.` is a real Mouse-Roll, therefore it need the window to be on top of other windows. Also it will need a mouse-pointer and a desktop.

You can find some more details on the "r"-prototype here: MAP - Move Active Point 2483.

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

MWR.P1

Parameter Explanation

P1 **-** (optional) Variable or Value

Generally **P1** is a number that is either positive or negative.

Positive numbers scroll the Mouse wheel forward (away from the user)
negative numbers scroll the Mouse wheel backwards (direction to the user).

You can scroll in very fine steps.
A normal Mouse-Wheel scroll with a standard Mouse is 120.
If you want to simulate two steps with a Mouse Wheel, therefore you would scroll by 240.

Example:

```
' Scrol the Mouse one Step forward
MWR.120
' Scrol the Mouse one Step backward
MWR.-120
' Tile the Mouse one Step right
MWT.120
' Tile the Mouse one Step left
MWT.-120
```

Example

```
'**********************************
' MWR.-Demo
'**********************************
STW.ct|IEFrame|Facebook - Internet Explorer
SCW.ct|Internet Explorer_Server|{&NOTEXT:}
' Scroll IE nach oben
MPO.
MWR.120
' Scroll IE nach unten
MWR.-120
PAU.1
ENR.
END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MWT. - Mouse Wheel Tile 2516
- MWE. - Mouse Wheel Event 2500
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5 Direct Engine

3.65.5.1 MAP. - Move Active Point

MiniRobotLanguage (MRL)

MAP. Command

Move active point (direct engine)

Intention

To understand this command, you need to know the term "Active Point".

What is the "Active Point"?
The "active Point" is the "Focus Point" of the robot. It is the point "On screen", where the next robot action will happen.

How do i set the "active point"?
Normally you locate a button, window or object. These are all rectangular areas. The robot will set the "active point" to the point in the middle of these rectangular areas.

Can i influence the "active point" in any way?
This is where `MAP.` enters the game. Using `MAP.` you can set or move the "**active point**".

`MAP.` is somehow ike `MPO.` with the option "x".
But even if the "x" is not specified, `MAP.` will set the Active Point and NOT move the Mouse.
`MAP.` does only move the "**active Point**", it does not move the mouse cursor.
The most important usage is **to move the AP relative to where its right now.**

Relative moves do not count from 0,0, but from where the AP is. This way, you can locate a button and then move the AP 10 pixel to the right from the button center.
Here is an example-script:

```
STW.ct|PBWindowClass:0|TestApp
' We locate Button 2
SCW.nct|2|Button|Button 2
' We move the AP from the button center 100 points to the left
MAP.p|-100,0
' We move the Mouse to the AP
MMV.
' We move the AP from the current position 200 points to the
right
MAP.p|200
MMV.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Move the AP 100 up, and 100 left
MAP.p|-100,-100
MMV.
' Move AP for 100 points down, we are back in the button
center!
MAP.p|0,100
MMV.
ENR.
```

The next thing that is often used, is a "window relative coordinate". It counts from the upper left corner of the actually located button/Top- or Child-window.
The advantage over screen coordinates is, that these coordinates move with the window, if the window moves. The Editor will sometimes record "window local coordinates".
Here is an example:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
MAP.w|10,10
MLC.
ENR.
```

Note that you can use Formulas in (..) as parameters:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
VAR.$$NUM=5
MLC.p|(15*$$NUM),(-23*$$NUM)
```

Syntax

MAP.[P1|][P2,P3]

Parameter Explanation

P1 **-** (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Important:
In case you use `MAP.` without any Parameter, then `MAP.` will try to get the TOS-Content (Top of Stack). If it is a Single Point, like
`100,200`
then `MAP.` will locate that point (as Screen coordinates). In case on the Stack is a RECT, like this: `100,200,150,400`
then the Middle-Point of that RECT (in Screen coordinates) is been calculated and located as actual point.

This way you can more easily use MAP. together with `GRM.` and `GAR.` commands.
And immediately Move the Mouse there.

```
STW.ct|MyClassName|Sample Application
GAR.
GRM.
MAP.
MMV.
ENR.
```

Example

```
'*********************************
' MMV./MAP. - Sample
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Please start the Testapp
'
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
MAP.p|-100
MMV.
MAP.p|200
MMV.
MAP.p|-100,-100
MMV.
MAP.p|0,100
MMV.
ENR.
```

Remarks

This Command has been changed.

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5.2 MDE. - Mouse Double Click Event

MiniRobotLanguage (MRL)

MDE. Command

Mouse Double-Click Event (direct engine)

Intention

`MDE.` simulates a Double-Click Event at the target Position. It does not move or even use the mouse-pointer.

This Command simulates a "Mouse-Double Click Event", using the Direct Engine 231.

`MDE.` has therefore all the strong and all the weak sides of a typical Direct Engine 231 command.

`MDE.` supports all the prototypes which are useful for Mouse- and "Active Point"-positioning.
You can find some more details on the "r"-prototype here: MAP - Move Active Point 2483.

Some Hints:

- If `MDE.` seems not to work on your target window, try `MDI.`
- or in case you want a mouse move, use `MDC.`

Syntax

MDE.[P1|][P2,P3]

Parameter Explanation

P1 – `(optional) prototype coordinates`

`Prototype can be omitted or any of:`

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.

"x" - Set Active Point to the Command-position.

Example:
' Double Click Event and Active Point Move
MDE.wrx|-358,207
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'*********************************
' MDC.-Demo
'*********************************
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
MDC.p|(11*5),(-1*5)
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

–

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MDC - Mouse Double-left Click (Invinsible) 2442
- MDI. - Mouse Double-Click Invisible 2445
- MLU. - Mouse left up 2463
- MLD. - Mouse left down 2451
- MDM. - Mouse Down Message 2490
- MUM. - Mouse Up Message 2498
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5.3 MDM. - Mouse Down Message

MiniRobotLanguage (MRL)

MDM. Command

Mouse Down Message (direct engine)

Intention

For few operations, you may need to just press a mouse-button down and release it later.

There are two commands for this purpose:

- `MDM.` - Direct Engine 231
- `MLD.` - Compatible Engine 231

`MDM.` has all the strong and all the weak sides of a typical Direct Engine 231 command.

`MDM.` can simulate all the three mouse buttons, depending on parameter **P1**.

The `MDM.` Event will take place at the position of the current "Active Point".
You can find some more details on the "Active Point" here: MAP - Move Active Point 2483.

Some Hints:

- If `MDM.` seems not to work on your target window, try `MLD.`
- You can not use `MDM.` for Drag'n Drop operation, as the direct engine does not support that.
- For Drag'n Drop operations, always use: `MLD.`, `MMV./MPO.` and `MLU.`

Syntax

MDM.[P1]

Parameter Explanation

P1 - Prototypes:

```
1 = left mouse button will be pressed
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  m = then middle button will be pressed (press on wheel)
  r = right button will be pressed
```

Example

```
' Will click the right Mouse-Button
' on the WinXP-Desktop
' on an Icon with the name "Furmark"
STW.ct|Progman|Program Manager
SCW.nct|1|SHELLDLL_DefView|{&NOTEXT:}
SCW.nct|1|SysListView32|FolderView
SAO.rn|34|FurMark
MDM.r
MUM.r
ENR.
```

Remarks

-

Limitations:

-

See also:

- MUM. - Mouse Up Message 2498
- MLD. - Mouse left down 2451
- MLU. - Mouse left up 2463
- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5.4 MLE. - Mouse-Left-Click Event

MiniRobotLanguage (MRL)

MLE. Command

Mouse-Left-Click Event (direct engine)

Intention

This Command is the standard clicking command in automated software-installations.

MLE. will simulate a Left-Click (MLE.) with the Mouse using the direct engine 231.

The MLE. Command will not really move the mouse and do the click. It will instead tell the the target application "that there was a mouse-click".

Therefore the target window does not need to be on top. It can even be hidden or outside of the area where the mouse can move. MLE. can click buttons which are disabled, and it can even click on invisible controls.

Using MLE. then the mouse pointer will not be used. However the "action" will take place at the current "Active Point". In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Hints:

- Like all direct engine 231 commands, MLE. depends on the cooperation of the target application. In some - especially JAVA-based) applications, the MLE. will not work.
- In those cases where MLE. do not works as expected, use MLC.or MLI. or MLM. .
- If you have the problem that MLE. seems to be eaten by the target application sometimes, use BSC. or BCS.
- MLE. may not work on Buttons, when you have just located the Top-Window. This is normal Application behavior.

Syntax

 *© 2023 Theo Gottwald * https://www.smart-package.com*

MLE.[P1|][P2,P3]

Parameter Explanation

P1 - (optional) prototype coordinates

Prototype can be omitted or any of:

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.

"x" - Set Active Point to the Command-position.

Example:
' Left Click Event and Active Point Move
MLE.wrx|-358,207
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use

© 2023 Theo Gottwald * https://www.smart-package.com

```
variable and Formula in (..). You can also use a
Point-Variable instead of two coordinates.
```

Example

```
'**********************************
' MLE.-Demo
'**********************************
STW.ct|PBWindowClass:0|TestApp
SCW.nct|2|Button|Button 2
MLE.
MLE.r|(11*5),(-1*5)
ENR.
```

Remarks

For automated Installations prefer `BSC.` where possible.

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5.5 MRE. - Mouse-Right-Click Event

MiniRobotLanguage (MRL)

MRE. Command

Mouse-Right-Click Event

Intention

`MRE.` will simulate a Right-Click with the Mouse using the direct engine 231.

The `MRE.` Command will not really move the mouse and do the click. It will instead tell the the target application "that there was a mouse-click".

Therefore the target window does not need to be on top. It can even be hidden or outside of the area where the mouse can move. `MRE.` can click buttons which are disabled, and it can even click on invisible controls.

When using `MRE.`, the mouse pointer will not be used. However the "action" will take place at the current "Active Point". In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.
You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Hints:

- Like all direct engine 231 commands, `MRE.` depends on the cooperation of the target application. In some - especially JAVA-based) applications, the `MRE.` may not work.
- In those cases where `MRE.` does not works as expected, use `MRC.`or `MRI.`

Syntax

MRE.[P1|][P2,P3]

Parameter Explanation

P1 – `(optional) prototype coordinates`

`Prototype can be omitted or any of:`

 *© 2023 Theo Gottwald * https://www.smart-package.com*

GROUP I (based on rectangular Area) Prototypes

```
"w" - choose RECT from last located Window (Child-/ or Top-Window)
"t" - choose RECT from last located Top-Window
"k" - choose RECT from last located Child-Window

"a" - choose RECT from last located Accessible Object (E8) (using SAO.,AGR.,NAV.
"g" - choose RECT from last located graphical Object (E9) (using QFR,, ANA., UNI.
"o" - choose RECT from last located Object/Window (any sort of)
"s" - use Screen-Coordinates, these will go over the whole Screen, including Dual

for these "Base-Prototypes" you can specify additionally fine-tuning using:
 - Quadrant coordinates like:
 MMV.o|{A2-B4}
 - numeric numbers (positive and negative)
 MMV.o|-20,90
and you can combine this:
 MMV.o|-20,90 {A2-B4}
```

also there are three Prototypes that will do sort of "left-right" or "Mid" Positioning.
These 3 can not be combined with each other.

```
"d" - Sort of LBO. - Positioning on the Left side of the current Rectangle.
"e" - Sort of RBO. - Positioning on the Right side of the current Rectangle.
"r" - Midpoint of the last RECT (default)

' Example (Combination):
 MMV.oe|-20,90 {A2-B4}
```

GROUP II (based on Point) Prototypes

```
"m" - actual mouse-position. Usefull if the user moves the mouse somewhere and yo
"b" - last mouse-position, before last Move.
"p" - current "Active Point". (Default)
"l" - Last "Active Point".

If prototype is omitted, Screen-Coordinates are taken.

"x" - Set Active Point to the Command-position.

Example:
' Right Click Event and Active Point Move
MRE.wrx|-358,207
```

P2, P3 - (optional) x,y - coordinates
numeric coordinates, separated by comma. You can use variable and Formula in (..). You can also use a Point-Variable instead of two coordinates.

Example

```
'*********************************
' MRE.-Demo
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|ieframe|microsoft
MRE.

END.
```

Remarks

Mostly you do not need to specify the coordinates with `MMV.`or `MLI.` but you can directly write the coordinates into `MRE.`, `MRI.`, and `MRC.`.

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- SMH. - Set Menu Hook 660
- MRI. - Mouse Right Click Invisible 2476
- MRC. / MRI. - Mouse-Right-Click 2472
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5.6 MUM. - Mouse Up Message

MiniRobotLanguage (MRL)

MUM. Command

Mouse Up Message

Intention

For few operations, you may need to just press a mouse-button down and release it later.

There are two commands for this purpose:

- `MDM.` - Direct Engine 231
- `MLD.` - Compatible Engine 231

`MUM.` has all the strong and all the weak sides of a typical Direct Engine 231 command.

`MUM.` can simulate all the three mouse buttons, depending on parameter **P1**.

The `MUM.` Event will take place at the position of the current "Active Point".
You can find some more details on the "Active Point" here: MAP - Move Active Point 2483.

Some Hints:

- If `MUM.` seems not to work on your target window, try `MLU.`
- You can not use `MUM.` for Drag'n Drop operation, as the direct engine does not support that.

For Drag'n Drop operations, always use: `MLD.`, `MMV./MPO.` and `MLU.`

Syntax

MUM.[P1]

Parameter Explanation

P1 - Prototypes:

```
l = left mouse button will be pressed
m = then middle button will be pressed (press on wheel)
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
r = right button will be pressed
```

Example

-

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MDM. - Mouse Down Message 2490
- MLD. - Mouse left down 2451
- MLU. - Mouse left up 2463
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.5.7 MWE. - Mouse Wheel Event

MiniRobotLanguage (MRL)

MWE. Command

Mouse-Wheel Event (direct engine)

Intention

This Command is the standard command to scroll the Mouse Wheel.
It can scroll in any direction, either forward/backward, or left/right.

`MWE.` will simulate a the Mouse Wheel Event using the direct engine 231.

The `WME.` Command will not really move the mouse and do the Scrolling.
It will instead tell the the target application "that there was a mouse-wheel movement".

Therefore the target window does not need to be on top. It can even be hidden or outside of the area where the mouse can move.
`MWE.` can scroll in Windows that are disabled, and it can even work on invisible windows.

Using `WME.` then the mouse pointer will not be used. However the "action" will take place at the current "Active Point". In most cases the "Active Point" is been set, by localizing a button or control or object. In that case, the "Active Point" is in the center of the rectangular object.

You will definitely need to localize an active application/window before using this command.

You can find some more details on the "Active Point" and how to move it, here:
MAP - Move Active Point 2483.

Hints:

- Like all direct engine 231 commands, `MWE.` depends on the cooperation of the target application. In some - especially JAVA-based) applications, the `MWE.` will not work.
- In those cases where `MWE.` does not works as expected, use `MWR.` or `MWT.`

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

MWE.[P1]

Parameter Explanation

P1 **-** (optional) Variable or Value

Generally **P1** is a number that is either positive or negative.
Additionally, you can prefix P1 with the following letters to change the functionality:

H – Switch to Horizontal Mode, Tile Mouse-Wheel instead of scrolling.
Negative numbers will tile left, Positive numbers will tile right.

A – First XButton is pressed while the Wheel is rotated
B – Second XButton is pressed while the Wheel is rotated
C – ctrl (strg) is pressed on the Keyboard while the Wheel is rotated
D – Left Mouse Button is pressed while the Wheel is rotated
E – Right Mouse Button is pressed while the Wheel is rotated
F – Middle Mouse Button is pressed while the Wheel is rotated

Note that it does not make any difference whether you prefox the number with the letter, or if you use the letter as suffix.

```
MWE.H120
MWE.120H
```

is both completely equal.

Positive numbers scroll the Mouse wheel forward (away from the user)
negative numbers scroll the Mouse wheel backwards (direction to the user).

You can scroll in very fine steps.
A normal Mouse-Wheel scroll with a standard Mouse is 120.
If you want to simulate two steps with a Mouse Wheel, therefore you would scroll by 240.

Example:

```
' Scrol the Mouse one Step forward
MWE.120
' Scrol the Mouse one Step backward
MWE.-120
' Tile the Mouse one Step right
MWE.H120
' Tile the Mouse one Step left
MWE.-120H
```

Example

```
'***********************************
' MWE.-Demo
'***********************************
' Scroll Google Chrome Browser
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|Chrome_WidgetWin_1|- Google Chrome
MWE.120H
PAU.1
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MWR. - Mouse Wheel Roll 2480
- MWT. - Mouse Wheel Tile 2516
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.6 IMO. / NMO. - If Mouse Option

MiniRobotLanguage (MRL)

IMO. Conditional Statement

If Mouse Option

Intention

Conditional Statement. Test several parameters about the mouse. You can test the following states:

- Is Mouse installed in the current system?
- Does the Mouse have a Mouse Wheel?
- Is the Left Button of the Mouse Up / Down?
- Is the Middle Button of the Mouse Up / Down?
- Is the Right Button of the Mouse Up / Down?

To make these tests, you specify a number or a keyword behind the `IMO.`-command.

Example:

```
IMO.IsMouse
  DBP.Yes we have a mouse
ELS.
  DBP.No we have no mouse
EIF.
```

In the same way you could have specified a number instead, which would do just the same:

```
IMO.1
  DBP.Yes we have a mouse
ELS.
  DBP.No we have no mouse
EIF.
ENR.
```

To test for a Mousewheel, you would similarly write:

```
IMO.iswheel
  DBP.Yes we have a mouse-wheel
ELS.
  DBP.No we have no mouse-wheel
EIF.
```

and you can even combine all the tests like this:

```
IMO.iswheel,ismouse
  DBP.Yes we have a mouse+wheel
ELS.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  DBP.No we have no mouse+wheel
EIF.
```

or use the numeric way - please always use the (...) for formulas:

```
IMO.(1+4)
  DBP.Yes we have a mouse+wheel
ELS.
  DBP.No we have no mouse+wheel
EIF.
```

or add the numbers yourself:

```
IMO.5
  DBP.Yes we have a mouse+wheel
ELS.
  DBP.No we have no mouse+wheel
EIF.
```

or separate them by comma:

```
IMO.1,4
  DBP.Yes we have a mouse+wheel
ELS.
  DBP.No we have no mouse+wheel
EIF.
```

Which will al do the same. Below you will find a list with the keywords and Options you can test for.

Syntax

IMO.P1[,P2...PX] … ELS. … EIF.

Parameter Explanation

P1 - Parameter that specifies one or more conditions to test for.
You can either specify the keyword or the number. You can
also specify multiple keywords or numbers, separated by comma. You can also add the numbers when you specify the condition.

```
"ismouse"       1   - TRUE if Mouse installed.
"nomouse"       2   - TRUE if no Mouse installed.
"iswheel"       4   - TRUE if Mouse Wheel installed.
"nowheel"       8   - TRUE if no Mouse Wheel installed.
"islmbdown"     16  - TRUE if Left Mouse Button (LMB) down.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
      "islmbup"       32  - TRUE if Left Mouse Button (LMB)
up.
      "ismmbdown"     64  - TRUE if Middle Mouse Button (MMB)
down.
      "ismmbup"       128 - TRUE if Middle Mouse Button (MMB)
up.
      "isrmbdown"     256 - TRUE if Right Mouse Button (RMB)
down.
      "isrmbup"       512 - TRUE if Right Mouse Button (RMB)
up.
```

Example

```
'**********************************
' IMO.-Sample
'**********************************
IMO.iswheel,ismouse
  DBP.Yes we have a mouse+wheel
ELS.
  DBP.No we have no mouse+wheel
EIF.
END.
```

Remarks

This command may get additional options in future updates.

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- GMP. - Get-Mouse-Position 2439
- IPC. / NPC. - If-Pixel-Colour 1816
- GPC. - Get-Pixel-Colour 1801

© 2023 Theo Gottwald * https://www.smart-package.com

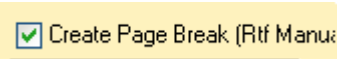
3.65.7 LBO. - Left Border

MiniRobotLanguage (MRL)

LBO. Command

Left Border Checkmark Operation

Intention



See the Checkmark left of the Text.

In cases where they can not be directly located, you can locate them using the LBO. command.

What is the "Active Point"?
The "active Point" is the "Focus Point" of the robot. It is the point "On screen", where the next robot action - for example a mouse-click - will happen.

The `LBO.`-command will not move the mouse but just set the active point.

A complete sequence could look like this:

```
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.nct|1|TcxPageControl|{&NOTEXT:}
SCW.nct|1|TcxTabSheet|EDOC
SCW.nc|1|TcxButtonEdit
LBO.
MMV.
ENR.
```

Using `LBO.` without any parameter will address a rectangular Button **on the leftt side inside** the last located rectangle. To address the checkmark in teh picture above you will need to specify "**A**" as an additional second parameter **P2.**

If you specify a number as **P1** , `LBO.` will move the active point to inside the rectangular area.

X-Position = Number x height

For example:

```
LBO.5
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

would be nearly in the center of the Text-Control.
If you specify an additional Parameter like this:

```
LBO.1|A
```

the active point will be moved at the right side to outside of the control.
For example to press that checkmark on the picture above.

You can finetune the X-Position if you specify additionally "Q" (Quarter) and "H" (Half) as third parameter. They will increase the Position just a bit.

As this Method works relative to the height of the control, it is safer then to move the mouse using point coordinates.

Syntax

LBO.[P1][|P2]

Parameter Explanation

P1 – ***(optional)*** Number of "height units" to step inside or outside of the selected rectangle/control. If omitted, "1" is used as default value.

P2 – ***(optional)*** One of these letters:
A - change direction to outside, if missing direction is to inside of the rectangle.
H - Half, add just a few points to the position
Q - Quarter, add just a few points to the position

Example

```
'**********************************
' MMV./MPO. -Demo
'**********************************
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.nct|1|TcxPageControl|{&NOTEXT:}
SCW.nct|1|TElButtonEdit|{&NOTEXT:}
LBO.1|A
MMV.
RBO.3
END.
```

Remarks

You can use multiple `RBO.`'s and `LBO.`'s to find your desired, final target position.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- RBO. - Right Border 2519
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click quIckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.8 MOS. - Mouseless Object Select

MiniRobotLanguage (MRL)

MOS. Command

Mouseless Object Select (direct engine / Accessibility / focussless technology)

Intention

The `MOS` -Command can be used to simulate a many different actions, like a mouse-click and other actions without really using or moving the mouse.

It is based on accessibility and will preferably work with objects that have been located using `SAO.`, `NAV.` or `AGR.`, like in this example:

```
STW.ct|PBWindowClass:0|BCS - Testform
AGR.43|Install
MOS.
```

`MOS.` is perfect for clicking for example Links in the WEB-Browser, even if they are actually not visible.

To use `MOS.`, you need to locate these elements first for example, using the `SAO.`, `NAV.` or `AGR.` Command.

`MOS.` understands a "subscript" that can be given as command-line. For example:

```
STW.ct|PBWindowClass:0|BCS - Testform
AGR.43|Install
MOS.WIN.MPO.DMO.WII.
ENR.
```

You can find a list with these "Subscript-commands" below. In our example the following will happen:

```
MOS.WIN.MPO.DMO.WII.
```

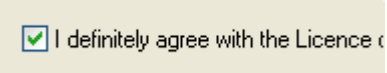
`WIN.` - we set the internal value for `WII.` to infinte
`MPO.` - the mouse jumps to the target position (it will jump back later!)
`DMO.` - we'll click two times (mouse-less - its just a demo)
`WII.` - we will wait until the application is ready again for new actions

You can try this script with the supplied BCS-Testapplication.

*© 2023 Theo Gottwald * https://www.smart-package.com*

MOS. has a special feature for checkmarks. We'll now use it on our BCS-Testform. This feature comes with the two "sub-commands" BTC. and BTS. Please do not intermix these "sub-commands" with the real BTC. command.
They have just the same name, but are different things..In this case, BTC. - means "Button clear" and BTS. means "Button Set". These two wubcommands are specially to have a checkmark or radio button SET or CLEARED.
The BTC. sub-command must be followed by a mouse-click-sub-command to be complete.
This way you can choose which kind if mouse-click you want to do. It will look like this:

```
STW.ct|PBWindowClass:0|BCS - Testform
AGR.44|agree
MOS.BTS.MLE.
ENR.
```



And to have that checkmark cleared, you would use:

```
' direct engine click
MOS.BTC.MLE.
```

or as alteranatice

```
' compatible engine click.
MOS.BTC.MLI.
```

Note that in case its a real child-window and not an web-based object, you can also use SCW. instead of AGR. or SAO. to locate it like this:

```
STW.ct|PBWindowClass:0|BCS - Testform
SCW.ct|button|agree
MOS.BTS.MLE.
ENR.
```

Using any kind of mouse clicks,.this will also work as expected. However if you want to use the MOS. (mouseless operation) sub-command, you need to have the target located using AGR. or SAO. or NAV. Anything else may not work. Especially SCW. would not work in this case.

```
STW.ct|PBWindowClass:0|BCS - Testform
AGR.44|agree
'This will also work
MOS.BTC.MOS.
ENR.
```

After execution, the MOS. - Command will drop a 0 or 1 onto the stack, to show if the command was successful (1) or not (0). If the command was not successful then a possible reason is that the control does internally not support the used interface.

MOS. has a lot of options/sub-commands, see below. The options from "**a**" to "**i**" and "**k**", are mostly for selecting items in List-controls. The other sub-commands are Mouse-operations and Waiting operations.

© 2023 Theo Gottwald * https://www.smart-package.com

Using our TestApp_01.exe, here is an example that shows how to use the sub-commands to select the items: 4,5 and 8:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|ListBox|List-Item Nr.1
SAO.rn|34|List-Item Nr.4
MOS.TAS.
SAO.rn|34|List-Item Nr.5
MOS.CCS.
SAO.rn|34|List-Item Nr.8
MOS.CCS.
ENR.
```

In Listboxes, the items must not be visible to become selected.
You can also select item 18, which will then get visible.

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|ListBox|List-Item Nr.1
SAO.rn|34|List-Item Nr.4
MOS.TAS.
SAO.rn|34|List-Item Nr.5
MOS.CCS.
SAO.rn|34|List-Item Nr.18
MOS.CCS.
ENR.
```

Note that `CCS.` will extend (add items to) an existing selection, while `TAS.` will remove all current selections and start with a new selection. `MOS.` may also be useful with other controls to select and unselect items.

Syntax

MOS.P1

Parameter Explanation

P1 - Optional subscript (inline)-commands:

```
 "TAF" - "a" ' Take Focus
 "TAS" - "b" ' Take Selection
 "ESE" - "c" ' Ext. Select
 "ASE" - "d" ' Add to Selection
 "RSE" - "e" ' Remove Selection
 "SSE" - "f" ' Strong Select
 "SHC" - "g" ' Shift-Click
 "CCS" - "h" ' ctrl-click - select
 "CSE" - "i" ' ctrl-click cancel selection
 "MER" - "j" ' MER at Pos.
 "MSE" - "k" ' do Multiselect
 "MOS" - "m" ' Mouseless Operation Select
 "DMO" - "n" ' Double Mouseless Operation Select
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"MLI" - "o" ' MLI at place
"DMR" - "p" ' Double Mouseless Operation Select with rest
"MPO" - "q" ' Mousepointer Position Set
"MSO" - "r" ' Mouse Operation
"MDO" - "s" ' Mouse Double Operation
"WIN" - "t" ' Set Wait-Input Idle- Time to infinte
"MDC" - "u" ' MDC at place
"MRC" - "v" ' MRC at place
"WII" - "w" ' Wait-for-Input-Idle (max. 5 Seconds)
"MLE" - "x" ' MLE.
"SMP" - "y" ' Set Mousepos Mode
"BTP" - "z" ' Set Left-ControlPosition like for Checkmark
"BTC" - "0" ' check State before click and click
              if actually state=0 (check it)
"BTS" - "1" ' check State before click and click if actually
              state=1 (uncheck it)
```

Example

```
'----------------------------------------
' MOS-Example 1
' This example Script will click a
' button in the BCS-Testapp
'----------------------------------------
DBM.2
STW.ct|PBWindowClass:0|BCS - Testform
AGR.43|Install
MOS.WIN.MPO.DMO.WII.
ENR.
END.
```

Remarks

You can use these sub-commands to make extended selections in List-Objects like Listview or Listbox.

Limitations:

Please note that not all List-Controls support this command, while most other elements in windows or IE-Browser support `MOS`..

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466

© 2023 Theo Gottwald * https://www.smart-package.com

- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.9 MWC. - Mouse Window Click

MiniRobotLanguage (MRL)

MWC. Command

Mouse Window Click

Intention

This command will do nearly the same like `MLE.` Internally it uses a special method that will work, even if a Standard `MLE.` would fail. It is only for very special use.
If you use the numerical parameters, note that these are experimental and therefore undocumented, unlike
0 - `MLE.` and
15 - `MLC.`

```
' Start the BCS.-Testapp from your Sample-Apps folder
STW.wct|45|PBWindowClass:0|BCS - Testform
SCW.wctid|45|Button|Install|1012|1
MWC.
```

Syntax

MWC.[P1]

Parameter Explanation

P1 - (optional) - number to specify details on how to click.

```
1 - check                  5,9 - 1+
2 - button HILITE          6,10 - 2+
3 - Button DE              7,11,14 - 3+  13 -DoubleLCLick
4 - Click                  8,12 - 4+

0  - MLE. - like
15 - MLC - like
```

Example

```
'*****************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Demo for MWC.
'****************************************
' Start the BCS.-Testapp from your Sample-Apps folder
STW.wct|45|PBWindowClass:0|BCS - Testform
SCW.wctid|45|Button|Install|1012|1
MWC.
END.
```

Remarks

-

Limitations:

Currently this method is experimental and only implemented for x32. Under x64 the `MWC.` will make a normal mouse-click, like the other Mouse-clicks.

See also:

- 1.C Direct Engine and Compatible Engine 231
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406
- BSC. - Button Special Click 2435
- BCS. - Button Click Special 2431

© 2023 Theo Gottwald * https://www.smart-package.com

3.65.10 MWT. - Mouse Wheel Tile

MiniRobotLanguage (MRL)

MWT. Command

Mouse Wheel Tile (compatible engine)

Intention

`MWT.` simulates a Mouse-Wheel Scroll left or right.
This Command simulates the Mouse Wheel using the Compatible Engine 231.
Therefore the Mouse must be positioned over the target application window.

The usage of the command is simple, you can just write:

```
MPO.
MWT.120
or
MWT.-120
```

Then the mouse wheel will do the scroll either forward (120) or backward (-120).

In most cases the "Active Point" is been set, by localizing a button or control or object.
In that case, the "Active Point" is in the center of the rectangular object.

You can find some more details on the "Active Point" and how to move it, here: MAP - Move Active Point 2483.

This command will simulate a real Mouse-Wheel.
Therefore the Mouse pointer needs to be over the target window.

`MWR--.` has therefore all the strong and all the weak sides of a typical Compatible Engine 231 command.

`MWT.` is a real Mouse-Roll, therefore it need the window to be on top of other windows. Also it will need a mouse-pointer and a desktop.

You can find some more details on the "r"-prototype here: MAP - Move Active Point 2483.

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

MWT.P1

Parameter Explanation

P1 **-** (optional) Variable or Value

Generally **P1** is a number that is either positive or negative.

Positive numbers scroll the Mouse wheel forward (away from the user)
negative numbers scroll the Mouse wheel backwards (direction to the user).

You can scroll in very fine steps.
A normal Mouse-Wheel scroll with a standard Mouse is 120.
If you want to simulate two steps with a Mouse Wheel, therefore you would scroll by 240.

Example:

```
' Scrol the Mouse one Step forward
MWR.120
' Scrol the Mouse one Step backward
MWR.-120
' Tile the Mouse one Step right
MWT.120
' Tile the Mouse one Step left
MWT.-120
```

Example

```
'**********************************
' MWT.-Demo
'**********************************
STW.ct|IEFrame|Facebook - Internet Explorer
SCW.ct|Internet Explorer_Server|{&NOTEXT:}
' Scroll IE nach oben
MPO.
MWT.120
' Scroll IE nach unten
MWT.-120
PAU.1
ENR.
END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1.C Direct Engine and Compatible Engine 231
- MWR. - Mouse Wheel Roll 2480
- MWE. - Mouse Wheel Event 2500
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466
- **OPT. - Optional Settings** 2406

© 2023 Theo Gottwald * https://www.smart-package.com

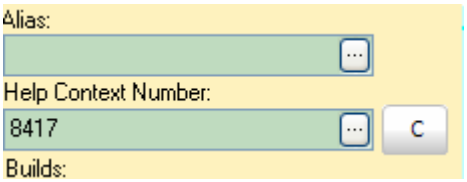
3.65.11 RBO. - Right Border

MiniRobotLanguage (MRL)

RBO. Command

Right Border Operation

Intention



See the Buttons inside the Text-Box, on the right side.
Also see the Button on the right of the Text-Box below.

In cases where they can not be directly located, you can locate them using the `RBO.` command.

What is the "Active Point"?
The "active Point" is the "Focus Point" of the robot. It is the point "On screen", where the next robot action - for example a mouse-click - will happen.

The `RBO.`-command will not move the mouse but just set the active point.

A complete sequence could look like this:

```
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.nct|1|TcxPageControl|{&NOTEXT:}
SCW.nct|1|TcxTabSheet|EDOC
SCW.nc|1|TcxButtonEdit
RBO.
MMV.
ENR.
```

Using `RBO.` without any parameter will address a rectangular Button **on the right side inside** the last located rectangle (in this cas the Text-Box). For example the upper button in the picture above.

If you specify a number as **P1** , `RBO.` will move the active point to inside the rectangular area.
X-Position = Number x height

For example:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
RBO.5
```

would be nearly in the center of the Textbox-Control.
If you specify an additional Parameter like this:

```
RBO.1|A
```

the active point will be moved at the right side to outside of the control.
For example to press that "C"-Button on the picture above.

You can finetune the X-Position if you specify additionally "Q" (Quarter) and "H" (Half) as third parameter. They will increase the Position just a bit.

As this Method works relative to the height of the control, it is safer then to move the mouse using point coordinates.

Syntax

RBO.[P1][|P2]

Parameter Explanation

P1 – ***(optional)*** Number of "height units" to step inside or outside of the selected rectangle/control. If omitted, "1" is used as default value.

P2 – ***(optional)*** One of these letters:
A - change direction to outside, if missing direction is to inside of the rectangle.
H - Half, add just a few points to the position
Q - Quarter, add just a few points to the position

Example

```
'**********************************
' MMV./MPO. -Demo
'**********************************
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.nct|1|TcxPageControl|{&NOTEXT:}
SCW.nct|1|TElButtonEdit|{&NOTEXT:}
RBO.1|A
MMV.
RBO.3
END.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

You can use multiple `RBO.`'s and `LBO.`'s to find your desired, final target position.

Limitations:

-

See also:

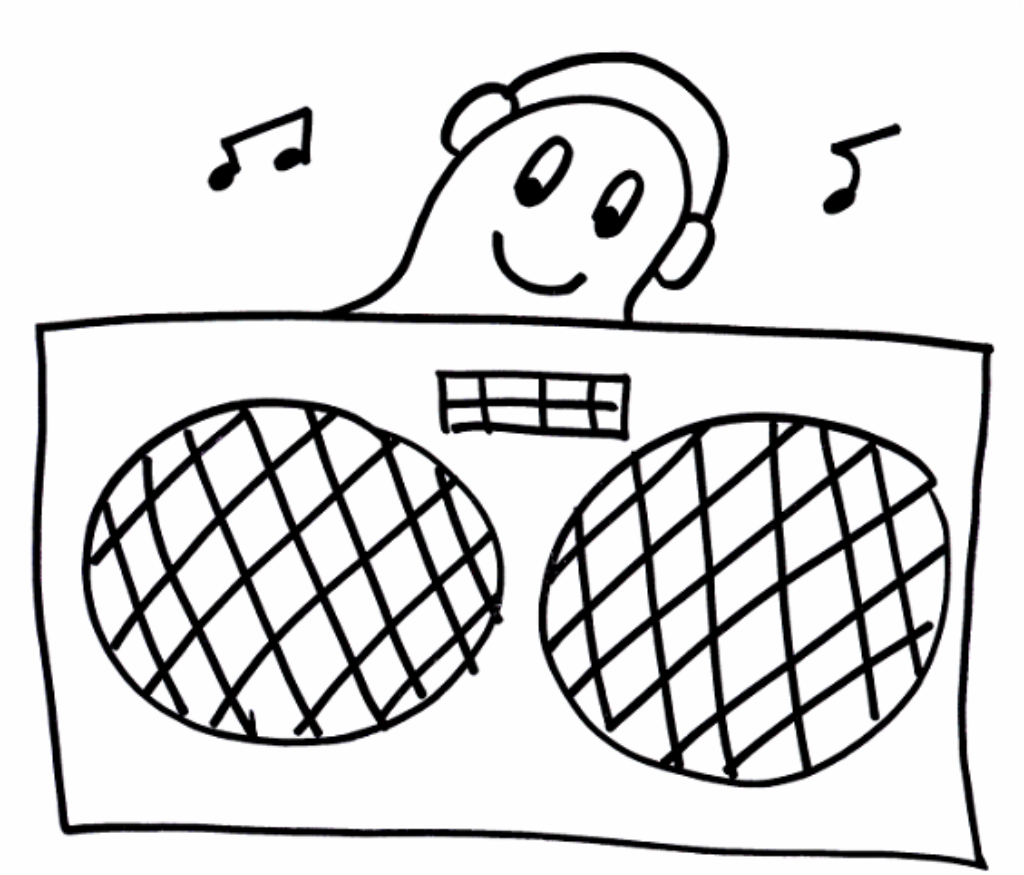
- LBO. - Left Border 2506
- MLE. - Mouse-Left-Click Event 2492
- MLM - Mouse Left click Move (Invinsible) 2457
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MRC. / MRI. - Mouse-Right-Click 2472
- MMV - Mouse MoVe 2466

© 2023 Theo Gottwald * https://www.smart-package.com

3.66 Multimedia Commands

3.66.1 ! Smarty's Multimedia Commands

Smarty's Multimedia Commands



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.66.2 MPA. - Music Player

MiniRobotLanguage (MRL)

MPA. Command

Music Player

Intention

`MPA.` is a "Music Player". You give it a piece of Sound, and it will play it.

Simple Example:

```
MPA.MySong.mp3
```

`MPA.` can also accept a List of Audio Files (Play-List).
The Songs must be separated using a "**>**" Sign. Example:

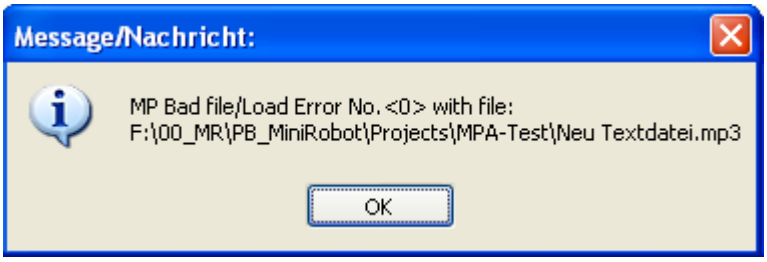
```
MPA.MySong.mp3>MySong2.mp3>MySong3.mp3
```

Of course this makes more sense using a variable:

```
$$PLI=MySong.mp3>MySong2.mp3>MySong3.mp3
MPA.$$PLI
```

You can add a third parameter to test if a file can be played or not.
Example:

```
$$LID=?pfad\Neu Textdatei.mp3
MPA.$$LID|$$RES
STS.DUMP
MBX.$$RES
```



The Music-Player will not pop up any error-messages, instead place all errors whether in this variable or on TOS, if no variable is given.

Some Hints:

- Use `MPB.` in case the Music should play in the background
- Use `SVL.` to change the Overall Volume and the Mute-State
- Use `MPC.` and `MPB.` for Sound effect like faster or slower Playback rate etc.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

MPA.[P1][|P2]

Parameter Explanation

P1 **-** Filename or Playlist (=Filenames, separated with ">")

P2 **-** (optional) Variable, receives Error-Messages if there are Errors during playback.

Example

```
'**********************************
' MPA.-Sample
'**********************************
' Repeat a Song endless
DOL.1
  MPA.MySong.mp3
OOP.
END.
```

Remarks

-

Limitations:

This command makes use of the Windows "**Direct Show**" **API**. If Direct X / Direct Show is not installed on the target PC, these commands will not work. If you have a working Windows Media Player,on the system, then everything will work as expected. Generally all Audio-Formats, that are supported with the Windows Media Player, should work with this command also.

See also:

- SVL. - Set Volume 2539
- MPB. - Music Play Background 2526
- MPC. - Music Player Command 2530

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.66.3 MPB. - Music Play Background

MiniRobotLanguage (MRL)

MPB. Command

Music Play Background

Intention

The `MPB.` command acts as a "Background Audio Player". It is part of a set of commands that includes `MPC.` - Music Player Commands. These commands enable you to play music or any audio in the background. For instance, you can use it to play music during an automated installation process.

However, its uses aren't limited to music playback. For instance, you can use it to play a "police siren" sound during alarm conditions.

To utilize the `MPB.`, you need to feed it an audio file. As it operates in the background, a separate command is required to check if it's ready to play the audio. If we neglect to do so, we won't hear any audio because when the script concludes, `MPB.` also terminates any ongoing playback

Simple Example:

```
' Start the PLayback, here we assume that the song is located
' in ?path\, that is where the scriptfile is.
MPB.MySong.mp3
' Now we wait until the sound was played
WRS.
' We drop the result
DRS.
' We end the script.
ENR.
```

`MPB.` can also accept a list of Audio Files (Play-List).

The Songs must be separated using a "**>**" Sign. Example:

```
MPB.MySong.mp3>MySong2.mp3>MySong3.mp3
```

Of course this makes more sense using a variable:

```
$$PLI=MySong.mp3>MySong2.mp3>MySong3.mp3
MPA.$$PLI
```

You can add a third parameter to get the "Ticket-Number" of the background task.
Using the "Ticket Number" you can control the background task.
More details on how to work with a "Ticket number" and background tasks, see:

"Ticket Numbers" and Background Tasks 2632

*© 2023 Theo Gottwald * https://www.smart-package.com*

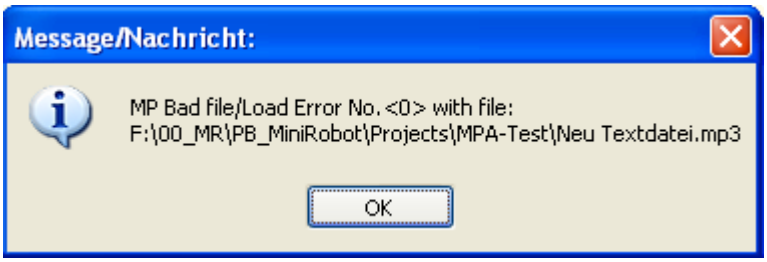
Using the ticket number and the `MPC.` - command, you can influence the behavior of the playing song.

```
' Start the Playback
MPB.$$LIB
' Set the Playback speed to double speed
MPC.|setrate|2
' Decrease the volume a bit (Range is from 100 - loud, to zero
' - sound off)
MPC.|svol|90

' Now we wait until the sound was played
WRS.
' We drop the result
DRS.
' We end the script.
ENR.
```

If there was an error during Playback, you will get it as a result of the background task.
Use `GRS.` for that purpose.

Example:

```
$$LID=?pfad\Neu Textdatei.mp3
MPB.$$LID|$$RES
WRS.
GRS.|$$RES
MBX.$$RES
```



The Music-Player will not pop up any error-messages, instead give the error back as a result.

Some Hints:

- Use `MPA.` in case the Music should play in the background
- Use `SVL.` to change the Overall Volume and the Mute-State
- Use `MPC.` and `MPB.` for Sound effect like faster or slower Playback rate etc.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

MPB.[P1][|P2]

Parameter Explanation

P1 – Filename or Playlist (=Filenames, separated with ">")

P2 – (optional) Variable, receives Ticket-Number for Result (`GRS.`)

Example

```
'**********************************
' MPB.-Sample
'**********************************
' Start the PLayback
MPB.MySong.mp3
GSB.Fade
' Now we wait until the sound was played
WRS.
' We drop the result
DRS.
' We end the script.
END.
:Fade
' Fade Balance from left to right
FOR.$$CNT|-10000|10000|100
  MPC.|sbal|$$CNT
NEX.
' Reset Balance
MPC.|sbal|0
RET.
```

Remarks

-

Limitations:

This command makes use of the Windows "**Direct Show**" **API**. If Direct X / Direct Show is not installed on the target PC, these commands will not work. If you have a working Windows Media Player,on the system, then everything will work as expected. Generally all Audio-Formats, that are supported with the Windows Media Player, should work with this command also.

*© 2023 Theo Gottwald * https://www.smart-package.com*

See also:

- SVL. - Set Volume 2539
- MPA. - Music Player 2523
- MPC. - Music Player Command 2530

© 2023 Theo Gottwald * https://www.smart-package.com

3.66.4 MPC. - Music Player Command

MiniRobotLanguage (MRL)

MPC. Command

Music Player Commands

Intention

MPC. functions as a companion to the MPB. command. It allows you to modify the playback parameters once the audio is playing through MPB.
Therefore, prior to using MPC., it's crucial that you're familiar with the workings of MPB.
The first parameter needed for using MPC. is the "Ticket Number".
By using the "Ticket Number", you gain control over the associated background task.
More information on handling "Ticket Numbers" and managing background tasks can be found at:

"Ticket Numbers" and Background Tasks 2632

Using the ticket number and the MPC. - command, you can influence the behavior of the playing song.

```
' Start the Playback
MPB.$$LIB
' Set the Playback speed to double speed
MPC.|setrate|2.5
' Decrease the volume a bit (Range is from 100 - loud, to zero
' - sound off)
MPC.|svol|90

' Fade Balance from left to right
FOR.$$CNT|-10000|10000|100
  MPC.|sbal|$$CNT
NEX.
' Reset Balance
MPC.|sbal|0

' Now we wait until the sound was played
WRS.
' We drop the result
DRS.
' We end the script.
ENR.
```

You can change values dynamically to achieve Sound effects.
Note that such effects may be different on different fast computers.

```
$$LIB=?path\Lied.mp3
' Start the Playback
MPB.$$LIB
' Now we wait until the sound was played
```

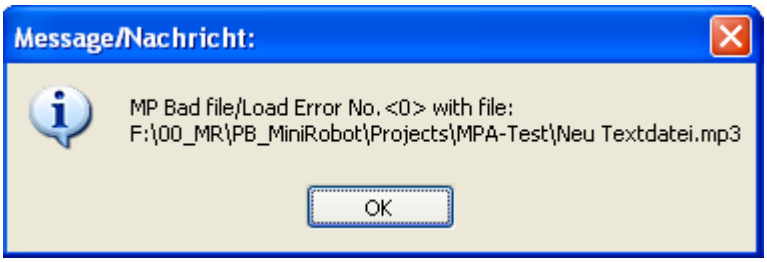

© 2023 Theo Gottwald * https://www.smart-package.com

```
MPC.|setrate|1.3
MPC.|svol|90
MPC.|gd|$$RET
' Change position and volume dynamically
FOR.$$CNT|0|$$RET|1000
  MPC.|sp|$$CNT
  FOR.$$VOL|100|77|-1
    MPC.|sv|$$VOL
  NEX.
NEX.
MPC.|stop
' Reset Balance
MPC.|sbal|0
ENR.
```

If there was an error during Playback, you will get it as a result of the background task.
Use `GRS.` for that purpose.

Example:

```
$$LID=?pfad\Neu Textdatei.mp3
MPB.$$LID|$$RES
WRS.
GRS.|$$RES
MBX.$$RES
```



The Music-Player will not pop up any error-messages, instead give the error back as a result.

Some Hints:

- Use `MPA.` in case the Music should play in the background
- Use `SVL.` to change the Overall Volume and the Mute-State
- Use `MPC.` and `MPB.` for Sound effect like faster or slower Playback rate etc.

Syntax

MPC.[P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** Sub-Command. Out of the following:

"Stop" - Stops Playback from the selected background task immediately.

"Pause" - Pauses Playback from the selected background task immediately.
Continue playing using the "Play" Sub-Command.

"Play" - Start Playback of a loaded or Paused Audio-File, from the
selected background task.

"setrate","set rate","set playback rate","sr"
Change the Playback Rate/Speed of the currently selected background task.
The playback rate is expressed as a ratio of the normal speed. Thus, 1.0 is normal
playback speed, 0.5 is half speed, and 2.0 is twice the original speed..

"setpos","set position","sp"
Change the Playback Position of the currently selected background task.
The playback Position is expressed is a 64-bit integer value. To obtain the
maximum absolute position, use the "get duration" command.

"set end position","setend","setendpos","se"
Change the Playback-End Position of the currently selected background task.
The playback End-Position is expressed is a 64-bit integer value. To obtain the
maximum absolute position, use the "get duration" command.

"move","from here","mv"
Change the Playback-Current Position relative to the current position,
of the currently selected background task.
The playback Current-Position is expressed is a 64-bit integer value. To obtain the
maximum absolute position, use the "get duration" command.
Example:

```
MPC.$$TIN|mv|1000
```

"moveend","move end","end from here","me"
Change the Playback-End Position relative to the current position,
of the currently selected background task.
The playback End-Position is expressed is a 64-bit integer value. To obtain the
maximum absolute position, use the "get duration" command.
Example:

```
MPC.$$TIN|mv|1000
```

"get duration","gd","gdur"
Get the Playback-Duration of the currently selected background task.

© 2023 Theo Gottwald * https://www.smart-package.com

The playback Duration is a 64-bit integer value.
Example:

```
MPC.$$TIN|gd|$$RET
DBP.$$RET
```

"get position","gpos","gp"

Get the Playback-Duration of the currently selected background task.
The playback Duration is a 64-bit integer value.
Example:

```
MPC.$TIN|gp|$$RET
DBP.$$RET
```

"load"

"set volume","svol","setvolume","sv"

Set the Playback-Volume in a Range from 0 to 1000, of the currently selected background task. The volume steps are not linear, this way you can set very soft music / sounds more accurate and have less steps for loud sounds.
Example:

```
MPC.$TIN|sv|100
```

"set balance","sbal","setbalance","sb"

Set the Playback-Balance in a Range from -10000 to 10000, of the currently selected background task.
The Balance (left/right speaker) is equal when you set the value to zero.
If you set it to -10000 then only the right speaker is used. If you set it to 10000 then only the left speaker is used. Example:

```
MPC.$TIN|sb|10000
MPC.$TIN|sb|-10000
```

"get volume","gvol","gv"

Get the Playback-Volume in a Range from 0 to 1000, of the currently selected background task. The volume steps are not linear, this way you can set very soft music / sounds more accurate and have less steps for louder sounds.
Example:

```
MPC.$TIN|gv|$$RES
DBP.$$RES
```

"get balance","gbal","gb"

Get the Playback-Balance in a Range from -10000 to 10000, of the currently

© 2023 Theo Gottwald * https://www.smart-package.com

selected background task.
The Balance (left/right speaker) is equal when you set the value to zero.
If you set it to -10000 then only the right speaker is used. If you set it to 10000
then only the left speaker is used. Example:

```
MPC.$TIN|gb|$$RES
DBP.$$RES
```

P2 – (optional) Variable, receives or contains values, depending on the chosen Sub-Command.

Example

```
'**********************************
' MPB. / MPC.-Sample
'**********************************
' Start the PLayback
MPB.$$LIB
' Now we wait until the sound was played
MPC.|setrate|1.3
MPC.|svol|90
MPC.|gd|$$RET
' Change position and volume dynamically
FOR.$$CNT|0|$$RET|500
  MPC.|sp|$$CNT
  FOR.$$VOL|100|70|-1
    MPC.|sv|$$VOL
  NEX.
NEX.
MPC.|stop

RET.
```

Remarks

-

Limitations:

This command makes use of the Windows "**Direct Show**" **API**. If Direct X / Direct Show is not installed on the target PC, these commands will not work. If you have a working Windows Media Player,on the system, then everything will work as expected. Generally all Audio-Formats, that are supported with the Windows Media Player, should work with this command also.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- SVL. - Set Volume 2539
- MPA. - Music Player 2523
- MPB. - Music Play Background 2526

© 2023 Theo Gottwald * https://www.smart-package.com

3.66.5 MPS. - Music Play Speaker

MiniRobotLanguage (MRL)

MPS. Command

Music Play through PC-Speaker

Intention

Note-setting command. Did you ever had the idea to compose a melody on your own? This is the way you can start.

The `MPS.`-command generates Beep-like Music-Signals, through the PC-Speaker. Frequency and Duration can be customized using musical symbols.

Its also useful for debugging. and to signal errors and system states without showing a text.

You have some sort of musical notation to input the sound.
Here is an example:

```
' Octave 1, Play a note, make a Pause, Play another Note
MPS.O1 A+ P A-

' Octave 4, Play a note, make a Pause, Play another Note,
Octave 1
MPS.O4 A+ P A-
```

Here is a larger example:

```
$$TOA=mnt120o4d7d9g9g9b9b9>d6<b5p9>
$$TOA=$$TOA+d7e9d9c9<b9>d9c6<a5p9d7d9f#9f#9a9a9>c6<a9a5p9
$$TOA=$$TOA+>d7e9d9c9<b9a9g3d9d9d9g6b5d9d9d9g6b5p9g7g9f#9f#
$$TOA=$$TOA+9e9e9d5d9d9d9f#6a7d9d9d9f#6a5p9>d7e9d9c9<b9a9g7p7d
$$TOA=$$TOA+9d9d9g6b7d9d9d9g6b5p9g7g9f#9f#9e9e9d5d9d9d9f#6a7d
$$TOA=$$TOA+9d9d9f#6a5p9>d7e9d 9c9<b9a9g3
MPS.$$TOA
ENR.
```

Some musical settings (`T,O,<,>,L,MN,MS,ML`) will stay persistent between multiple calls to the command as long as the script is running.

Syntax

MPS.P1

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - (optional) - Sequence of musical notes and dates that form a melody.

Use these letters:

```
A,A+,A#
B-,B
C, C+, C#
D-, D, D+,D#
E-,E
F,F+,F#
G,G+,G#

P - fixed Pause
```

and the special letters

```
N - followed by value between 1 and 90 (Note)
T - followed by value between 32 and 255
(Takt)
L - followed by value between 1 and 64
(Length)
O - followed by value between 0 and 6 (Oct)
< - decrease Oct by following number
> - increase Oct by following number
```

and these symbols

```
MN,MS,ML
```

to write you own compositions.

Some musical settings (`T,O,<,>,L,MN,MS,ML`) will stay persistent between multiple calls to the command as long as the script is running.

Example

```
'**********************************
' MPS. Sample "Joy to the world"
'**********************************
$$TAO=mst255O2L2>DL4C#.L8<BL2A.L4GL2F#ED.
$$TAO=$$TAO+L4AL2B.L4B>L2C#.L4C#L2D.L4DDC#<BAA.
$$TAO=$$TAO+L8GL4F#>DDC#<BAA.L8GL4F#F#F#F#L8F#GL
$$TAO=$$TAO+2A.L8GF#L4EEEL8EF#L2G.
$$TAO=$$TAO+L8F#EL4D>L2DL4<BA.L8GL4F#GL 2F#EL1D
MPS.$$TAO
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END.
```

Remarks

Use `PRR.` and `PRE.` to make your Melody play in the background.

Limitations:

-

See also:

- MSB. - Message Beep 2040
- SAY. - Say something 2048
- DIT. - Dial Tone MFV 2018
- SVL. - Set Volume 2539
- MPA. - Music Player 2523
- MPB. - Music Play Background 2526
- MPC. - Music Player Command 2530
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.66.6 SVL. - Set Volume

MiniRobotLanguage (MRL)

SVL. Command

Set the Soundcard/Mixer Volume (OS independent)

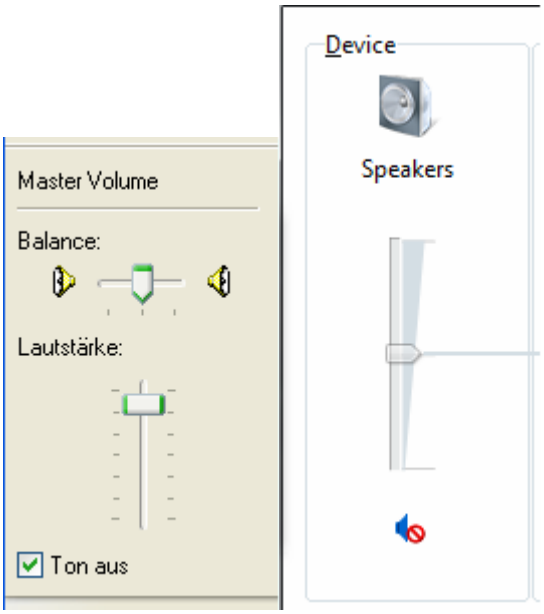
Intention

Get/Set the Soundcard/Mixer Master-Volume.
Get/Set the Mute-State of the Master- Volume.

Modes of operation:

1. Mute Master Volume

```
SVL.Set Mute|1
```



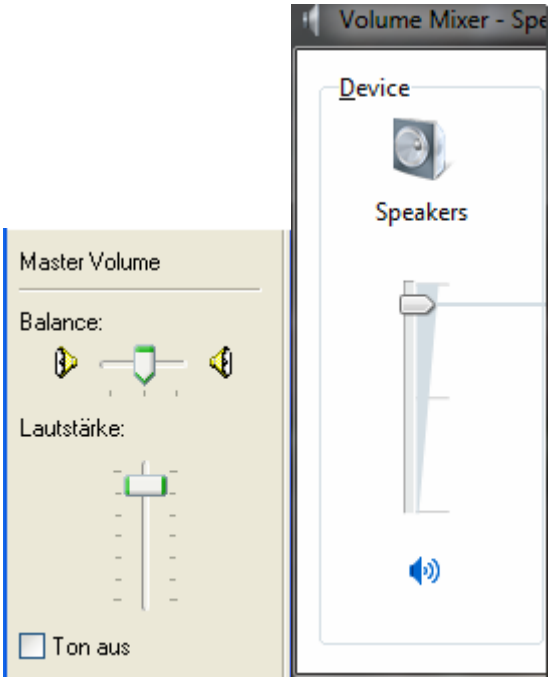
These two pictures show the result under XP (left picture) and Windows 7/x64 (right picture).
You will also get an `OK` on TOS. In case the operation would fail, you will get an error-message on TOS instead, and you will get the timeout-flag set, which is cleared otherwise.

```
------------------------------------
User-Stack-Dump:
------------------------------------
Stackpos: 000 ->String       :OK
------------------------------------
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Now lets go the opposite way and UnMute the sytem.

```
SVL.Set Mute|0
```



And voila' the Sound is back!

2. You can find out, if a system is muted or not.

```
SVL.Get Mute|$$RES
DBP.$$RES
```

If you get a "`1`" back, the system is muted, otherwise you'll get a "`0`" in `$$RES`. And that means that the system is not muted, sound is ON.

3. You can get the Master Volume.

```
' Get the Master Volume
SVL.get master volume|$$RES
DBP.$$RES
```

No matter, if its XP or Windows 7, the result is always a number in the range from 0 to 100. About these numbers, 100 is always the maximum volume, and 0 is total silence.

4. Set/change the Master Volume:

```
' Set Master Volume to 95 (Range is 0 to 100)
SVL.Set Master Volume|95
```

as always you will get a "`OK`" on the TOS if the operation succeeds.

© 2023 Theo Gottwald * https://www.smart-package.com

So far, all commands are perfectly transparent between different windows OS.

The following commands may show different results between different OS.

5. Find out how many channels we can control:

```
SVL.get channel count|$$RES
DBP.$$RES
ENR.
```

The result may be different between XP and higher Windows versions.

For Windows XP, I get the following Results. This may be slightly different on your actual system, depending on the availability of such Sound-Hardware.

```
Channel 1: Master Volume
Channel 2: Wave Volume
Channel 3: CD Volume
Channel 4: SW Synthesizer
Channel 5: Front Speaker
```

In Vista/Windows 7 and higher you should actually avoid these functions, as the result is unpredictable.

6. Get Channel Volume

```
' Get Volume from Channel No. 2
SVL.Get Channel Volume|2|$$RES
DBP.$$RES
```

7. Set Channel Volume

```
' Set Volume from Channel No. 2
SVL.Set Channel Volume|2|55
DBP.$$000
```

8. Get Channel Mute

```
SVL.Get Channel Mute|2|$$RES
DBP.$$RES
```

9. Set Channel Mute

```
SVL.Set Channel volume|2|1
DBP.$$RES
```

"1" will Mute the Sound-Channel, while "0" will UnMute the Sound-Channel.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

SVL.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Subcommand - choose any of the following Subcommands. This will specify what to do:

"gm","get mute" - use P2 for the Result (Variable)
"sm","set mute" - use P2 for the Value (1 - Mute / 0 - UnMute)

"gmv","get master volume" - use P2 for the Result (Variable)
"smv","set master volume" - use P2 for the Volume Value

"gc","get channel count" - use P2 for the Return Value (Variable)

' Avoid the following Subcommands under Vista and higher OS.

"gcv","get channel volume" - use **P2** for the channel number and
P3 for the Return Value (Variable)

"scv","set channel volume" - use **P2** for the channel number and
P3 for the Volume Value.

"gcm","get channel mute" - use **P2** for the channel number and
P3 for the Return Value (Variable)

"scm","set channel mute" - use **P2** for the channel number and
P3 for the Mute State (1 - Mute / 0 - UnMute)

P2 - (optional) - additional Parameter, depending on the used Subcommand.
P3 - (optional) - additional Parameter, depending on the used Subcommand.

Example

```
'**********************************
' Mute Volume-Sample
'**********************************
' Mute the PC
SVL.Set Mute|1

' UnMute the PC
SVL.Set Mute|0
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

The major functions like "**Mute/UnMute**", "Set Master Volume" etc. are expected to work transparently on any windows OS. From Windows 2000 to Windows 8. If the operation succeeds, you will find an "OK" on TOS. Otherwise an error message.

Limitations:

See the comments about which functions to avoid, if using an Vista or higher windows version.

See also:

- SAY. - Say something 2048
- DIT. - Dial Tone MFV 2018
- MPS. - Music Play Speaker 2536
- MPA. - Music Player 2523
- MPB. - Music Play Background 2526
- MPC. - Music Player Command 2530

© 2023 Theo Gottwald * https://www.smart-package.com

3.67 NT Service commands

3.67.1 !Service Commands

NT Service Command codes Previous 2539 Top 41 Next 2550

MiniRobotLanguage (MRL)

Service Command codes

Manage NT Services

General Information about the Service commands

The service command(s) need a service name to specify the service we want to deal with.
A service has two names:

1. the "display name" (dn:) - that is the name you see in the service manager console

2. the "service name" (sn:) - that is the name you can see in the service properties, only.

You can see an example in the picture below.

© 2023 Theo Gottwald * https://www.smart-package.com

Generally we are using the "service name" for all service commands.
If you prefer to specify a "display name" 8or a part of it), prefix it with a "dn:".

Then the command will internally try to find the matching service name for that service.

You do not need to specify the full "service-name" or "display-name". You can use the
3.2 Standard-Search Pattern 341
to specify the name of the service.

To make clear, which service name you are using, you can prefix it with:

"**dn:**" if the "display-name" is been specified
or
"**sn:**" if you have specified the "service name".

As noted, all commands use the "Standard search patterns", therefore you can just give a unique part of the service name or specify a pattern. Here are some examples:

If you use:

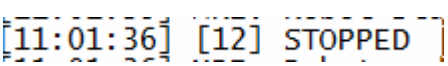
```
SER.grs|bitlock
DBP.$$000
```

the service will place an error message on TOS, that is "`Error: No such service.`".
You can see the Error-Message in the Debug-Output of the Editor. The reason is, that "bitlock" is a part of the "display name" of the service, but generally we search for the "service name". And for the "Bitlocker Service", the proper "service name" is "`BIDESVC`" as you can see above.

Therefore we could specify:

```
SER.grs|BDESVC||$$RES
DBP.$$RES
```

and would get the proper result:



if that service is available and actually stopped. Of course we could also use the "display name", if we prefix it with a "dn:" and write:

© 2023 Theo Gottwald * https://www.smart-package.com

```
SER.grs|dn:bitloc||$$RES
DBP.$$RES
```

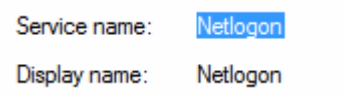
which will also get the expected result ("STOPPED").

As you can see, there is no need to type the whole, complete service name, as we can use the Search Pattern Engine here. That is why "bitloc" in the above example will work just fine.

Here are two things you should take care for:

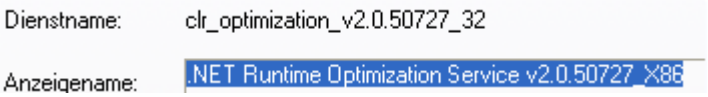
1. On international computers the "display name" of a service may differ from country to country. For example, the "Netlogon" Service shows up on a german computer as

where it shows up on a english system as expected:



Therefore prefer to use the "service name" whenever your script may touch international clients computers.

2. Make sure that the search pattern, that you specify will really hit the proper service. Therefore specify the target service as close as possible. The good thing about the Pattern Engine is, that you can omit version information that are really obsolete, for example, see the picture below.



You could use "`clr_optim&ANDTHEN:v2.`" to specify such a service using a pattern. Like this:

```
SER.dnf|clr_optim&ANDTHEN:v2.||$$RES
DBP.$$RES
```

and you will get the expected result.

To verify if a service is available or running is one thing, and it will most often be done using the `ISC.` command. To change service attributes or to restart a service is the main job of the `SER.` command.

What about "state groups"?

© 2023 Theo Gottwald * https://www.smart-package.com

Imagine you want to start a service and therefore you want to check if the service is already running. You can do that in several ways. Now you check if the state of the service is "run" and its not. Are you going to run that service next?
What if the state of the service was "start pending" which is as good as "run" but is in fact another state?

To solve this issue, we have so called state groups for some instructions.
These have only 3 states, which are represented by 3 numeric values:

```
"stop","stopped","stop pending"                                  =
1
"running","run","start pending","continue pending"          = 4
"pause","paused","pause pending"                                 = 7
```

Using these you can easily check, if you really need to start or stop or continue a service.

"Run States" to set using

```
SER.set run state|(Service name)|(target state number)

START SERVICE                = 0
STOP SERVICE                 = 1
PAUSE SERVICE                = 2
CONTINUE SERVICE             = 3
UPDATE SERVICE MANAGER       = 4
```

"Run States" as in return from

```
SER.get run state numeric|(Service name)|$$RES

STOPPED          = 1
STARTING         = 2
STOPPING         = 3
RUNNING          = 4
CONTINUE PENDING = 5
PAUSE PENDING    = 6
PAUSED           = 7
UNKNOWN          = any other number
```

Some commands, like ISC. / NSC. support textual state, these are:

```
"stop","stopped"       = 1
"start pending"        = 2
"stop pending"         = 3
"running","run"        = 4
"continue pending"     = 5
"pause pending"        = 6
"pause","paused"       = 7
```

Some commands support textual state and a "state group":

```
"stop","stopped","stop pending"                                  =
1
"running","run","start pending","continue pending"          = 4
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"pause","paused","pause pending"                              = 7
```

Possibly Actions to take an Service failure:

```
SC_ACTION_NONE                = 0
SC_ACTION_RESTART             = 1
SC_ACTION_REBOOT              = 2
SC_ACTION_RUN_COMMAND         = 3
```

Service-Types

```
SER.gst|rasaut|$$RES

SERVICE_KERNEL_DRIVER                = 1
SERVICE_FILE_SYSTEM_DRIVER           = 2
SERVICE_ADAPTER                      = 4
SERVICE_RECOGNIZER_DRIVER            = 8
SERVICE_DRIVER                       = 11
SERVICE_WIN32_OWN_PROCESS            = 16
SERVICE_WIN32_SHARE_PROCESS          = 32
SERVICE_WIN32                        = 48
SERVICE_INTERACTIVE_PROCESS          = 256
SERVICE_TYPE_ALL                     = 319
```

Service Start Modes

```
SERVICE_AUTO_START      (Automatisch)      = 2
SERVICE_DEMAND_START    (Manuell)          = 3
SERVICE_DISABLED        (Deaktiviert)      = 4

also these two states which are in the SM not available,
and may not work with most services:

SERVICE_BOOT_START                         = 0
SERVICE_SYSTEM_START                       = 1
```

Textual Service Start Modes

```
(parsing is NOT case-sensitive)

"Auto","Auto Start","Automatisch"          = 2
"Demand","on Demand Start","Manuell"       = 3
"dis","Disabled","Deaktiviert"             = 4
"boot","Boot Start"                        = 0
"system","System Start"                    = 1
```

Example

```
'*********************************
' SER.
'*********************************
SER.dnf|yaho||$$RES
DBP.$$RES
SER.snf|yaho||$$RES
DBP.$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SER.grs|yaho||$$RES
DBP.$$RES
SER.grn|yaho||$$RES
DBP.$$RES
ENR.
'*********************************
```

Remarks

-

Limitations:

-

See also:

- FES. - For Each Service 2550
- SER. - Service Command 2559
- ISC. - If Servic Command 2552
- WSC. - Wait Service Command 2566
- ISI. / NSI. - If Service Installed 2557

© 2023 Theo Gottwald * https://www.smart-package.com

3.67.2 FES. - For Each Service

MiniRobotLanguage (MRL)

FES. Enumeration Command

For each Service

Intention

Enumerate system services. Using `FES.` your Skript can go "through all services" and check them "one by one" for whatever conditions you want.

Syntax

FES.P1|P2[|P3][|P4][|P5]

Parameter Explanation

P1 - Label where the for each will jump "for each element"
P2 - Temp. Variable, updated by FEX each round.
P3 - ***(optional)*** Flags:
"a" - enum and pattern by display name, else use service name
"b" - enum **SERVICE_RUNNING**
"c" - enum **SERVICE_STOPPED**
"d" - enum **SERVICE_PAUSED**
"e" - enum ***SERVICE_CONTINUE_PENDING***
"f" - enum ***SERVICE_STOP_PENDING***
"g" - enum ***SERVICE_PAUSE_PENDING***
"h" - enum ***SERVICE_START_PENDING***

You can use any combination of the above flags. The flags can be given in any order.

P4 - ***(optional)*** optional Standard Search Pattern
P5 - ***(optional)*** PC-Name, if omitted local machine is used.

Example

```
'*********************************************
' FES. usingtwo flags
'*********************************************
'
FES.doit|$$TXT|ac
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.

:doit
DBP.$$TXT
RET.
ENR.
```

Remarks

`FEM.` is a Member of the FEX-Family "For Each X". They are used for easy enumeration

Limitations:

-

See also:

- 3.1 Systemvariables 334 especially: `#fex#, #fxd#` . etc.
- 1.6.1. Program Flow Control 121
- FEF. - For-Each-File(/Dir) 924
- FEL - For-Each-Line 927
- FEP. - For-Each-Process 932
- FEV. - For-Each-Variable(-Line) 937
- !Service Commands 2544
- SER. - Service Command 2559
- ISC. - If Servic Command 2552
- WSC. - Wait Service Command 2566
- ISI. / NSI. - If Service Installed 2557

© 2023 Theo Gottwald * https://www.smart-package.com

3.67.3 ISC. - If Servic Condition

MiniRobotLanguage (MRL)

ISC. Conditional Statement

If Service Condition

Intention

This command can be used to manage System Services. You can check almost all conditions of system services. If they are installed, running or not. You can check the Starting Mode of a service and much more.

To specify a service you will need to give the "service name". Please see
How to get the proper service name 2544
on which choices you have to specify the service you want to deal with, in a way that it will really work for you, as expected.

ISC. / NSC. support textual state, these are:

```
"stop","stopped"        = 1
"start pending"         = 2
"stop pending"          = 3
"running","run"         = 4
"continue pending"      = 5
"pause pending"         = 6
"pause","paused"        = 7
```

If you use textual Run-States please do only specify one Run-State.
If you use numeric run-states, you can specify multiple Run-States.
For example:

```
   ISC.sta|dn:yah||2,4,3
but
   ISC.sta|dn:yah||running
```

Syntax

ISC.P1|P2[|P3][|P4][|P5]

Details of the parameters depend on the used Subcommand, see below.

Parameter Explanation

P1 - Subcommand

' Check the "running state" of a service
`"sta" or "state" or "any state"`

© 2023 Theo Gottwald * https://www.smart-package.com

Evaluate, if the "running state" of the service **P2 (you must specify the service name)**
is equal to the state given in **P4**. For P4, you can specify a numeric value or a textual
parameter like these: !Service Commands 2544
You can specify multiple values if they are separated with ",", like this:

```
 ISC.sta|dn:yah||2,4,3

' Example:
ISC.sta|dn:yah||stop
   DBP.Service is stopped.
ELS.
   DBP.Service is not stopped
EIF.

ISC.sta|dn:yah|||2,4
```

' Check the "running state group" of a service
"stg" or "state group"
Evaluate, if the "running state" of the service **P2 (you must specify the service name)**
is equal to the state given in **P4**. For **P4**, you can specify a numeric value or a textual
parameter, details see !Service Commands 2544

```
' Examples:
ISC.stg|dn:yah||stop
   DBP.Service is stopped or stop is pending.
ELS.
   DBP.Service is not stopped, and stop is not pending
EIF.
```

' Check for multiple "running state" of a service
"ans" or "any state"
Evaluate, if the "running state" of the service **P2 (you must specify the service name)**
is equal to the state given in **P4**. For P4, you can specify a numeric value or a textual
parameter like these: !Service Commands 2544

```
' Examples:
' Service may not have any of these states
NSC.ans|dn:yah||2,3,4
   DBP.Here
EIF.

' Service should have any of these states
ISC.ans|dn:yah||2,3,RUN
   DBP.Here
EIF.
```

' Check the start mode of a service

© 2023 Theo Gottwald * https://www.smart-package.com

```
  "sm" or "start mode"
```
Evaluate the "Start Mode" of a specified service and compare with a specified start-Mode **P4**. **P4** can be a number from 2 to 4 or a phrase.
See a list of all numbers, phrases and their meanings here: !Service Commands 2544

```
  ' Examples:
ISC.sm|dn:yah||demand
   DBP.Start Mode is "on demand"
ELS.
   DBP.Start Mode is not "on demand"
EIF.
DBP.$$000
ENR.
```

Further tests:

```
"run","running"
Test if Run State is 4 (RUN)
ISC.run|NetLog

"stop","stopped"
Test if Run State is 1 (STOP)
ISC.stop|bdsvc

"pau","paused"
Test if Run State is 7 (PAUSE)
ISC.pau|servicename

"dis","disabled","deaktiviert"
Test if Start-Mode is 4 (disabled)
ISC.dis|Logon

"auto","automatisch","autostart"
Test if Start-Mode is 2 (AUTO)
ISC.auto|sn

"manuell","demand","on demand"
Test if Start-Mode is 3 (on demand)
ISC.demand|sn

"pen","pending"
Test if the service Run State is in any
"Pending State". These are any of the states:
 "Start pending", "Stop pending", "Continue pending" or
"Pause pending".
ISC.pen|sn

"pen-sta","start pending"
Test if the service is "Start pending" or "Continue pending"
ISC.pen-sta|sn

"pen-sto","stop pending"
Test if the service is "Stop pending" or "Pause pending"
ISC.pen-sto|sn
```

***P2* - Service-Name**

© 2023 Theo Gottwald * https://www.smart-package.com

P3 - Computer-Name or empty. In that case the local computer is used.

P4 - (optional) Meaning of P4 depends on used Subcommand.

What happens in case of an error?
If the specified service is not found, the Error-Message "Error: No such service." is been placed on TOS and the condition is evaluated to FALSE..
In case there are other errors during the processing of the command, the error-message is been placed on TOS and the condition is evaluated to FALSE.
The error-message will always be prefixed with an "Error: " to indicate that it is an error from the services-engine.
Here is an example for such an error-message:
"Error: The requested access to the services database was denied."

Example

```
'*********************************
' ISC.
'*********************************
ISC.sm|dn:yah||demand
   DBP.State is
ELS.
   DBP.State is not
EIF.
' Display Error if we got one
DBP.$$000
ENR.END.
'*********************************

'*********************************
' ISC.
' Error - will be displayed
'*********************************
ISC.sm|dn:yahs||demand
  DBP.State is
ELS.
  DBP.State is not
EIF.
' We'll see the error that the service does not exist.
DBP.$$000
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- !Service Commands 2544
- SER. - Service Command 2559
- WSC. - Wait Service Command 2566
- ISI. / NSI. - If Service Installed 2557
- FES. - For Each Service 2550

© 2023 Theo Gottwald * https://www.smart-package.com

3.67.4 ISI. / NSI. - If Service Installed

MiniRobotLanguage (MRL)

ISI. Conditional Statement

If Service installed

Intention

This command can be used to verify if a System Services is installed. You can check if a service is available on a specified (local?) computer system.

To specify a service you will need to give the "service name". Please see
How to get the proper service name 2544
on which choices you have to specify the service you want to deal with, in a way that it will really work for you, as expected.

```
ISI.wsearch|mypc
   DBP.service is available
ELS.
   DBP.no such service on mypc
EIF.
' Display Error if we got one
DBP.$$000
ENR.
```

To uninstall a Service use the
`SER.uni|sn|pc-name`
command.

Syntax

```
ISI.P1[|P2]
NSI.P1[|P2]
```

Parameter Explanation

P1 **- Service-Name**

P2 **-** ***(optional) Computer-Name*** or empty. In that case the local computer is used.

What happens in case of an error?

*© 2023 Theo Gottwald * https://www.smart-package.com*

If the specified service is not found, the Error-Message "Error: No such service." is been placed on TOS and the condition is evaluated to FALSE..
In case there are other errors during the processing of the command, the error-message is been placed on TOS and the condition is evaluated to FALSE.
The error-message will always be prefixed with an "Error: " to indicate that it is an error from the services-engine.
Here is an example for such an error-message:
"Error: The requested access to the services database was denied."

Example

```
'*********************************
' ISI.
'*********************************
ISI.dn:yah
   DBP.service is available
ELS.
   DBP.no such service
EIF.
' Display Error if we got one
DBP.$$000
ENR.
'*********************************
```

Remarks

-

Limitations:

-

See also:

- !Service Commands 2544
- SER. - Service Command 2559
- ISC. - If Servic Command 2552
- WSC. - Wait Service Command 2566
- FES. - For Each Service 2550

© 2023 Theo Gottwald * https://www.smart-package.com

3.67.5 SER. - Service Command

MiniRobotLanguage (MRL)

SER. Command

Manage NT Services

Intention

This command can be used to manage System Services. You can start and stop services.
You can install and uninstall a Service. And you can check if they are running or not.
To specify a service you will need to give the "**service name**" (sn). You can also specify the "display name" (dn), if you prefix it with "**dn:**". Please see
How to get the proper service name 2544
on which choices you have to specify the service you want to deal with, in a way that it will really work for you, as expected.

Using the `SER.` - command, you can:

```
' Get the display-name of a service
 SER.dn from sn|sn|pc-name[$$RES]

' Get the service-name from the display name (or a part of it)
SER.sn from dn|sn|pc[|$$RES]

' Get the "Run-State" of a service
' See if its running, stopped or whatever.
' Result in Textform.
SER.get run state|sn|pc|$$RES

' Get the "Run-State" of a service
' See if its running, stopped or whatever.
' Result is numeric.
SER.get run state numeric|sn|pc|$$RES

' Set the "Run-State" of a service
' Start it, Stop it etc.
SER.set run state|sn|pc|$$RST

' Stop service "All"
' This command will stop a service and - if needed -
' all "dependend services", which could prevent the service
' from beeign stopped.
' This is very useful, in case you want to unistall a service.
SER.stop service all|sn|pc[|$$RES]

' Get "Start-Mode" of a service
' For a list of possible "Start Modes"
' see here:  !Service Commands 2544
SER.get start mode|sn|pc|$$RET
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Set "Start Mode" of a service
' For a list of possible "Start Modes"
' see here:  !Service Commands 2544
SER.set start mode|sn|pc|sm[||$$RES]

' Get "Service Type", get a technical Info on the type of a
service
' For a list of possible "Service Types"
' see here:  !Service Commands 2544
 SER.get_service_type|sn|pc[||$$RET]

' Restart a service
' On a "Restart", a service is been stopped first
' and then started again.
'
SER.restart service|sn|pc[||$$RES]

' Install a service
SER.inst|Apple2|$$PAT|appl

' Uninstall a service
' internally also stops the service and all dependend services
' like a SER.stop service all|sn|pc
SER.uninstall|sn:Apple2
```

Syntax

SER.P1|P2[|P3][|P4][|P5]

Details of the parameters depend on the used Subcommand, see below.

Parameter Explanation

P1 - Subcommand

' Evaluate the display name of a service
"dnf" or "dn from sn"
Evaluate the "display name" of the service **P2 (you must specify the service name)**
into variable **P3.**

```
' Examples:
 SER.dnf|sky
 SER.dnf|sky||$$RES
```

' Evaluate the service name of a service
"snf" or "sn from dn"
Evaluate the "service name" of the service **P2 (you must specify the display name)**
into variable **P3.**

```
' Examples:
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SER.snf|ya&AND:update
SER.snf|ya&AND:update||$$RES
```

' Evaluate the run state of a service with textual result

```
"grs" or "get run state"
```

Evaluate the "run state" of a specified service. You will get a result like "STOPPED" or "RUNNING". See a list of all states here:
!Service Commands 2544

```
' Examples:
SER.grs|BDESVC
SER.grs|BDESVC||$$RES
```

' Evaluate the run state of a service with numerical result

```
"grn" or "get run state numeric"
```

Evaluate the "run state" of a specified service. You will get a numeric result, that is a number from 1 to 7. See a list of all numbers and their meanings here:
!Service Commands 2544

```
' Examples:
SER.grn|BDESVC
SER.grn|BDESVC||$$RES
```

' Stop a service, and stop all dependend services

```
"sta" or "stop service all"
```

This command will stop (halt) a service.
If the service can not be stopped because there are dependant services still running, they will also be stopped.
This is your command for safely stopping a service.
It will be called internally before uninstalling a service, using "uninstall", to make sure the serivce is not running before Uninstallation.

```
SER.stop service all|sn|pc[||$$RES]
```

' Set the run state of a service, using a number (from list below)

```
"srs" or "set_run_state"
```

This command will stop or start a service. Use the value 0 to start a service.
The command will take care internally if a service is disabled, and do whatever is needed to start or stop the service, yet keeping it disabled. Means: Using this command, you can start a service, even though if its disabled.

```
' Start a service
SER.srs|dn:ya&AND:update||0
' Stop a service
SER.srs|dn:update||1
```

' Evaluate the start mode of a service with numerical result

```
"gsm" or "get start mode"
```

Evaluate the "Start Mode" of a specified service. You will get a numeric result, that is a number from 0 to 4. See a list of all numbers and their meanings here:
!Service Commands 2544

```
' Examples:
SER.gsm|dn:ya&AND:update
SER.gsm|dn:ya&AND:update||$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

' Set the start mode of a service with numerical result

"ssm" or "set start mode"

Set the "Start Mode" of a specified service. Use a numeric code, that is a number from 0 to 4 to specify the startmode.See a list of all numbers and their meanings here:

!Service Commands 2544. The operation may provide a result of 0 = failure, or 1 = success.

```
' Examples:
' Disable the service
SER.ssm|dn:ya&AND:update||4
' Set startmode to AUTOMATIC START
' and get result of operation in Variable $$RES
SER.ssm|dn:ya&AND:update||2|$$RES
```

' Evaluate the service type of a service with numerical result

"gst" or "get service type"

This command will provide you with a technical info on the selected service.
The result will be in textual form. The example:

```
SER.gst|dn:ya&AND:update
```

will provide this: "SERVICE_WIN32_OWN_PROCESS".
See here for a list of possible results: !Service Commands 2544.

' Restart a service

"rst" or "restart service"

This command will end and then restart the selected service.
The result will be on TOS or in teh given variable. Here is an example:

```
SER.rst|dn:Netlog
SER.rst|dn:Netlog||$$RET
```

will restart the service like if you choose "restart" in the Menu of the service-management console. Using this command, you can restart a service, even if its *disabled*.

' Uninstall a service

"uni" or "uninstall"

This command will try to uninstall/kill the service with that name.
It will first try to stop the service and all dependend services.
This is a special feature. Use with caution and try usage first in a safe environment.

```
SER.uni|sn|pc-name
```

' Install a service

"inst" or "install"

This command will try to install the service with that name and executable.
This is a special feature, and there is no warranty that this will work as expected.
Use with caution and try usage first in a safe environment.
You can remove a installed service using the above "Uninstall service".
Therefore the command is somehow safe.
The newly installed service does not start automatically, use the SER. command to set its proper parameters.

© 2023 Theo Gottwald * https://www.smart-package.com

```
SER.inst|Service Display Name|Path of Service Exercutable|
Service Name[|PC-Name]
```

P2 **- Service-Name**

P3 **-** ***Computer-Name*** or empty. In that case the local computer is used.

P4 **-** ***(optional)*** `Variable for Result of Operation, if omitted TOS is used.`

What happens in case of an error?
If the specified service is not found, the Error-Message `"Error: No such service."` is been placed on TOS.
In case there are other errors during the processing of the command, the return variable will be empty and the error-message is been placed on TOS.
The error-message will always be prefixed with an "`Error: `" to indicate that it is an error from the services-engine.
Here is an example for such an error-message:
`"Error: The requested access to the services database was denied."`

Example

```
'*********************************
' SER.
'*********************************
SER.dnf|clr_optim&ANDTHEN:v2.||$$RES
DBP.$$RES
SER.snf|yaho||$$RES
DBP.$$RES
SER.grs|yaho||$$RES
DBP.$$RES
END.
'*********************************
' Path to the service-executable
VAF.$$PAT=F:\00_MR\MR_Diff_Plus\release\Diff-Tool.exe
' Install as a service
SER.inst|Apple2|$$PAT|appl
MBX.!
' UNinstall the service
SER.uninstall|dn:Apple2
END.

'*********************************
' SER.
' Install-Service Sample Script
'*********************************
$$NAM=ComWdienst
$$PAT=?ws\CoinSvc.EXE
$$SRC=?path\CoinSvc.EXE
:again
IEF.$$PAT
  MBX.Bitte den vorhandenen Dienst erst Deinstallieren!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GTO.enx
EIF.
NSI.$$NAM
  COP.$$SRC|$$PAT
  SER.install|ComWdienst|$$PAT|$$NAM
  SER.restart service|$$NAM
ELS.
  MBX.Es ist eine ältere Version des Dienstes
installiert.$crlf$
  SER.stop service all|$$NAM
  SER.uninstall|$$NAM
  GTO.again
EIF.
:enx
ENR.

'*********************************
' SER.
' Uninstall-Service Sample Script
'*********************************
$$NAM=ComWdienst
$$PAT=?ws\CoinSvc.EXE
IEF.$$PAT
ISI.$$NAM
  SER.stop service all|$$NAM
  SER.uninstall|$$NAM
  DEL.$$PAT
ELS.
  MBT.$$NAM gibts hier nicht als Dienst.
EIF.
ELS.
  MBT.Münzdienst ist hier nicht in $$PAT$crlf$installiert.
EIF.
ENR.
```

Remarks

-

Limitations:

In some cases the SER.-Command will lead to a program crash. In this case you can fix the problem using

```
SER.srs|WEBZES_SERVER_NODE_01||0
```

instead use

```
EXC.?path\Service_start.bat|?path\
WPT.#pid#
```

Inside the batch file use

```
net start servicename
```

or

```
sc config servicename start= auto
```


© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- !Service Commands 2544
- ISC. - If Servic Command 2552
- WSC. - Wait Service Command 2566
- ISI. / NSI. - If Service Installed 2557
- FES. - For Each Service 2550

© 2023 Theo Gottwald * https://www.smart-package.com

3.67.6 WSC. - Wait Service Command

MiniRobotLanguage (MRL)

WSC. Wait for Command

Manage NT Services

Intention

This command can be used to manage System Services. You can wait until they are running or stopped. You can even wait for the Starting Mode of a service and much more.

To specify a service you will need to give the "service name". Please see
How to get the proper service name 2544
on which choices you have to specify the service you want to deal with, in a way that it will really work for you, as expected.

Syntax

WSC.P1|P2[|P3][|P4][|P5]

Details of the parameters depend on the used Subcommand, see below.

Parameter Explanation

P1 - Subcommand

' Wait for a specified "running state" of a service
"sta" or "state"
Wait until a service is paused, started or stopped. The command checks if the service
has the "running state" that is specified in **P4**. For **P4**, you can specify a numeric value
or a textual parameter like these: !Service Commands 2544
You can specify multiple values if they are separated with "," like this:

```
WSC.sta|dn:yah|||stop,stop pending
```

```
 ' Examples:
' Wait unlimited time until service is stopped
 WSC.sta|dn:yah||stop

' Wait 60 seconds time until service is started
 WSC.sta|dn:yah||start
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Check timeout flag
  JIT.error
```

' Wait for a specified "running state group" of a service
"stg" or "state group"
Evaluate, if the "running state" of the service **P2 (you must specify the service name)**
is equal to the state given in **P4**. For **P4**, you can specify a numeric value or a textual
parameter, details see !Service Commands 2544

```
' Examples:
WSC.stg|dn:yah|||stop
```

' Wait for any of multiple "running state" of a service
"ans" or "any state"
Evaluate, if the "running state" of the service **P2 (you must specify the service name)**
is equal to the state given in **P4**. For P4, you can specify a numeric value or a textual
parameter like these: !Service Commands 2544

```
' Examples:
' Service may not have any of these states
WSC.ans|dn:yah|!|2,3,4

' Service should have any of these states
WSC.ans|dn:yah||2,3,RUN
```

' Wait for a specified start mode of a service
"sm" or "start mode"
Evaluate the "Start Mode" of a specified service and compare with a specified start-Mode **P4**. **P4** can be a number from 2 to 4 or a phrase.
See a list of all numbers, phrases and their meanings here: !Service Commands 2544

```
' Examples:
WSC.sm|dn:yah|||demand
```

Further tests:

```
"run","running"
Wait until  Run State is 4 (RUN)
WSC.run|NetLog

' Wait until service is NOT running
WSC.run|yah||!

"stop","stopped"
Test if Run State is 1 (STOP)
WSC.stop|bdsvc

"pau","paused"
Test if Run State is 7 (PAUSE)
WSC.pau|servicename
```


© 2023 Theo Gottwald * https://www.smart-package.com

"dis","disabled","deaktiviert"
Test if Start-Mode is 4 (disabled)
WSC.dis|Logon

"auto","automatisch","autostart"
Test if Start-Mode is 2 (AUTO)
WSC.auto|sn

"manuell","demand","on demand"
Test if Start-Mode is 3 (on demand)
WSC.demand|sn

"pen","pending"
Test if the service Run State is in any
"Pending State". These are any of the states:
"Start pending", "Stop pending", "Continue pending" or
"Pause pending".
WSC.pen|sn

"pen-sta","start pending"
Test if the service is "Start pending" or "Continue pending"
WSC.pen-sta|sn

"pen-sto","stop pending"
Test if the service is "Stop pending" or "Pause pending"
WSC.pen-sto|sn

***P2* - Service-Name**

P3* - *Computer-Name or empty. In that case the local computer is used.

P4* - *(optional) Waiting time in seconds, if prefixed with a "!", the command will wait for NOT the given condition.

```
' Wait 10 seconds for service to become disabled
WSC.sm|yah||10|dis

' Wait 10 seconds for service to NOT be disabled
WSC.sm|yah||!10|dis
```

P5* - *(optional) Meaning of P4 depends on used Subcommand.

What happens in case of an error?
If the specified service is not found, the Error-Message "Error: No such service." is been placed on TOS and the condition is evaluated to FALSE..
In case there are other errors during the processing of the command, the error-message is been placed on TOS and the condition is evaluated to FALSE.
The error-message will always be prefixed with an "Error: " to indicate that it is an error from the services-engine.
Here is an example for such an error-message:
"Error: The requested access to the services database was denied."

© 2023 Theo Gottwald * https://www.smart-package.com

If an timeout occurs, the timeout flag will be set and the script will continue its operation.
You can check the timeout-flag with the `ITO./NTO.` or the `JIT./JNT.` commands, or use the `#tio#` System-Variable 334.

Example

```
'*********************************
' WSC.
'*********************************
WSC.sm|dn:yah|30|demand
ITO.
   DBP.Timeout occured
ELS.
   DBP.things happened in time
EIF.
' Display Error if we got one
DBP.$$000
ENR.END.
'*********************************
```

Remarks

-

Limitations:

-

See also:

- !Service Commands 2544
- SER. - Service Command 2559
- ISC. - If Servic Command 2552
- ISI. / NSI. - If Service Installed 2557
- FES. - For Each Service 2550

© 2023 Theo Gottwald * https://www.smart-package.com

3.68 OS and Software

3.68.1 ! Smarty's OS & Software Ops

Smarty's OS and Software Operations

Previous 2566 Top 41 Next 2572

Smarty's OS and Software Operations

Intention

...

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations

-

See also:

–

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.2 ADV. - Get Advanced Operator Privilege

MiniRobotLanguage (MRL)

ADV. Command

Get ADVanced Operator Privilege

Intention

This command gives and releases advances (systemcritical!) Privileges to the Robot.

Use command „CPR“ together with command „ADV“ to be able to close every process, including critical system-processes.
Please **be careful by using** this command, or else you may crash the system and damage files on your harddisk, or loose them forever.

Syntax

ADV.[Privilege]

Parameter Explanation

P1 - Privilege(s)
If P1 is omitted (empty) then the Privileges to close any Process are given (same as "k").
otherwise P1 must be a String of these letters:

```
b - Backup/read all files
c - Changenotify
d - ShutDown
e - remote Shutdown
f - Increase Quota
g - Systemtime "SeSystemtimePrivilege"
h - TimeZone "SeTimeZonePrivilege"
i - SystemEnvironment "SeSystemEnvironmentPrivilege"
j - "SeCreateSymbolicLinkPrivilege"
k - Killprocess "SeDebugPrivilege"
r - "SeRestorePrivilege"
s - "SeSecurityPrivilege"
t - "SeTakeOwnershipPrivilege"
u - "SeRelabelPrivilege"
v - "SeManageVolumePrivilege"
```

Example

```
'**********************************
' ADV.-Demo
'**********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Takeover Backup Privileges
ADV.b
' Result of the Operation is on the stack 0/1
DMP.6
MBX.Ready

'-------------------------------------------------------------
ENR.


'**********************************
' ADV.-Sample 2
'**********************************
' Takeover multiple Privileges in one command
ADV.brts
' Result of the Operation is on the stack 0/1
DMP.6
MBX.Ready

'-------------------------------------------------------------
ENR.
```

Remarks

Using this command you will be able to close any, even critical system-processes.
Please do not experiment with this command, as it may damage files on your harddisk and crash your system.
This command removes system-critical windows-protections and should only be used where you are sure you have to do.

Limitations:

-

See also:

- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.3 CPL. - Control-Panel

MiniRobotLanguage (MRL)

CPL. Command

Control Panel

Intention

This command will start Applets from System's Control panel.

Syntax

CPL.[P1]

Parameter Explanation

There are three way to use CPL.

1. CPL. without any parameter calls the System Control Panel itself.
2. CPL. #XY (XY= any Number from 1 to 111) calls System-Applets.
3. CPL. Name-of-Aplett

Not all CPL's which are accessible via numeric parameter are also accesible via string-parameter nd vice versa.

Possible string parameters include following german and english words.
The interpreter tries to find the closest match in case you misspelled a word.

```
CPL. my computer

- "my computer","arbeitsplatz"
- "audio","audio"
- "screensaver","bildschirmschoner"
- "borland","borland"
- "appearence","darstellung"
- "date/time","Datum/Zeit"
- "dun","DFUE"
- "printer","Drucker"
- "accessibility","Eingabehilfen"
- "settings","Einstellungen"
- "exchange","exchange"
- "hardware","hardware"
- "background","Hintergrund"
- "infrared","Infrarot"
- "internet","internet"
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
- "joystick","joystick"
- "sounds","Klaenge"
- "countries","Laender"
- "mouse","maus"
- "midi","midi"
- "modem","modem"
- "multimedia","multimedia"
- "music cd","Musik Cd"
- "network","Netzwerk",
- "odbc","odbc"
- "folder",ordner"
- "pcmcia","pcmcia"
- "power","energie"
- "scanner","scanner"
- "systemfonts","Schriftarten"
- "scsi","scsi"
- "setup","setup"
- "software","software"
- "system","system"
- "keyboard","Tastatur"
- "phone","Telefon"
- "video","video"
- "manager","manager"
- "Certificate Manager"
- "Index Service","Indexdienst"
- "Computer Management","Computerverwaltung"
- "Disk Management","Datentraegerverwaltung"
- "Disk Defrag","Defragmentierung"
- "Event Viewer","Ereignisanzeige"
- "Shared Folders","Freigaben"
- "Group Policies","Gruppenrichtlinien-Editor"
- "Local Users and Groups","Lokale Benutzer und Gruppen"
- "Removable Media Management","Wechselmedienverwaltung"
- "Performance Monitor","Leistungsmonitor"
- "Local Security Settings","Sicherheitseinstellungen"
- "Services","Diensteverwaltung"
- "Component Services","Komponentendienste"
- "Resultant Policies","Policies"
- "QoS Admission Control","QoS Admission Control"
- "Domain Controller Security Policy","Domain Controller
Security Policy"
- "Advanced Network Settings","Erweiterte
Netzwerkeinstellungen"
- "Distributed File System","Verteiltes Dateisystem"
- "DHCP","DHCP"
- "DNS","DNS"
- "Active Directory Domains and Trusts","Active Directory
Domains and Trusts"
- "Domain Security Policy","Domain Security Policy"
- "Active Directory Users and Computers","Active Directory
Users and Computers"
- "AD Sites and Services","AD Sites and Services"
- "Internet Authentication Service","Internet Authentication
Service"
- "Internet Information Services","Internet Information
Services"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
- "Removable Storage Operator Requests","Removable Storage
Operator Requests"
- "Routing and Remote Access","Routing and Remote Access"
- "Remote Storage","Remote Speicher"
- "Active Directory Schema","Active Directory Schema"
- "Security Templates","Security Templates"
- "Telephony","Telephony"
- "Terminal Services Configuration","Terminal Services
Configuration"
- "WINS","WINS"
- "Windows Management Infrastructure","Windows Management
Infrastructure"
- "Performance","Performance"
- "Task Scheduler","Geplante Tasks"
- "Bluetooth","Bluetooth"
- "Windows Updates","Windows Updates"
- "Speech Control","Sprachsynthese"
- "Tablet","Tablet"
- "XP-Firewall","XP-Firewall"
- "UPS","UPS"
- "Postoffice","Postoffice"
- "FindFast Settings","FindFast Settings"
- "Usermgr 2","Usermgr 2"
- "RecycleBin","Papierkorb"
- "NetworkNeighborhood","Netzwerkumgebung"
```

Following numeric parameters are supported at time of writing:

```
#1 -> Control Panel 1
#2 -> Accessability Properties (Keyboard)
#3 -> Accessability Properties (Sound)
#4 -> Accessability Properties (Display)
#5 -> Accessability Properties (Mouse)
#6 -> Accessability Properties (General)
#7 -> Add/Remove Programs Properties (Install/Uninstall)
#8 -> Add/Remove Programs Properties (Install/Uninstall)
#9 -> Add/Remove Programs Properties (Windows Setup)
#10 -> Add/Remove Programs Properties (Startup Disk)
#Programzugriffe und Standards festlegen
#11 -> Regional Settings Properties (Regional Settings)
#12 -> Regional Settings Properties (Number)
#13 -> Regional Settings Properties (Currency)
#14 -> Regional Settings Properties (Time)
#15 -> Regional Settings Properties (Date)
#16 -> Regional Settings Properties (Input Locales)
#17 -> Mouse Properties
#18 -> Mouse Properties
#19 -> Mouse Properties
#20 -> Mouse Properties
#21 -> Keyboard Properties
#22 -> Printers
#23 -> Fonts
#24 -> Add Printer Wizard
#25 -> Microsoft Exchange Profiles (General). Mail and Fax
Options
#26 -> Multimedia Properties (Audio)
#27 -> Multimedia Properties (Video)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
#28 -> Multimedia Properties (MIDI)
#29 -> Multimedia Properties (CD Music)
#30 -> Multimedia Properties (Advanced)
#31 -> Sounds Properties
#32 -> Modem Properties (General)
#33 -> Network (Configuration) Win 9x
#34 -> Network (Configuration) Win NT, XP (tested)
#35 -> System Properties (Device Manager)
#36 -> System Properties (Hardware Profiles)
#37 -> System Properties (Performance)
#38 -> System Properties Dialog
#39 -> Systemeigenschaften Recovery
#40 -> Dialog "Automatische Updates"
#41 -> Microsoft Workgroup Postoffice Admin
#42 -> View Printers
#43 -> View Fonts
#44 -> Pick a Time Zone Dialog
#45 -> Open "About Window Window"
#46 -> Install New Hardware Wizard
#47 -> Dial-up Networking Wizard
#48 -> Change screensaver Dialog
#49 -> Show "Map Network Drive" Window
#50 -> Show "Disconnect Network Disk" Window
#51 -> Display The BSOD (blue screen of death)Window
#52 -> Add/remove programs
#53 -> Date/time settings
#54 -> ODBC settings
#55 -> Choose & Print Test Page Of Current Printer
#56 -> Open unknown file
#57 -> Run Diskcopy Dialog
#58 -> Show "Format Disk (A)" Window
#59 -> (General): Find Fast Properties
#60 -> (General):Internet Properties
#61 -> (Security):Internet Properties
#62 -> (Content):Internet Properties
#63 -> (Connection):Internet Properties
#64 -> (Programs):Internet Properties
#65 -> (Advanced):Internet Properties
#66 -> Joystick Properties (Joystick)
#67 -> Password Properties (Change Passwords)
#68 -> System Properties (General)
#69 -> Freigabe erstellen Dialog. ( Manage Shares Dialog)
#70 -> Server Properties
#71 -> COM Ports Settings
#72 -> Add New Hardware Wizard
#73 -> Date/Time Properties
#74 -> Choose Time Zone:
#75 -> Telephony Settings (WinNT)
#76 -> Themes Settings
#77 -> Open third tab (first tab is 0)
#78 -> UPS Settings (General): Uninteruptable Power Supply
Properties WinNT
#79 -> Microsoft Mail Postoffice Settings / Microsoft
Workgroup Postoffice Admin
#78 -> Open Network Neighborhood
#79 -> XP-Firewall
#80 -> Systemsteurung Verwaltung
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
#81 -> Telephony
#82 -> Keyboard
#83 -> Tablett- und Stift
#84 Spreech
#85 fonts
#86 scanner and camera
#87 sprachen- und regionen
#88 Ordneroptionen
#89 netzwerkverbindungen
#90 Netzwerkinstallationsassistent
#91 Mouse
#92 Drucker
#93 Geplante Tasks
#94 Infrarot
#95 date
#96 date
#97 bluetooth
#98 Benutzerkonten (XP-Version, bunt
#99 Benutzerkonten-2 Profi-Version, grau
#100 Benutzerkonten 3 - XP Version, bunt
#101 Automatische Updates
#102 Anzeige-Eigenaschaften
#103 Anzeige-Eigenaschaften
#104 Arbeitsplatz öffnen
#105 Netzwerkverbindungen
#106 Drucker und Faxgeräte
#107 Task Schedulter
#108 -> Display Properties (Background) #-> WR:
CPL\Hintergrund
#109 -> Display Properties (Screen Saver)
#110 -> Display Properties (Appearance)
#111 -> Display Properties (Settings)
```

P1 - Name of Applet or Applet # and number (see above)

Example

```
' CPL with a name
CPL.Advanced Network Settings

' Open Internet Security Settings
CPL.#61

' Open Microsoft Exchange Profiles (General)
CPL.#25
END.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- LIB. - Library 2614

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.4 GEN. - Get-EXE-Number

MiniRobotLanguage (MRL)

GEN. Command

Get Exe Number

Intention

Get the number of times an EXE-File is actually being executed.

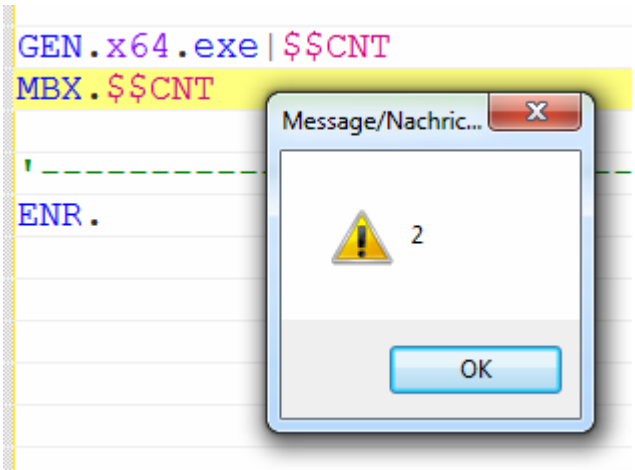
Assume there are two x64.exe running. The task-manager will show them:

x64.exe	1768	Admin	00	1.764 K	x64
x64.exe	1780	Admin	00	1.760 K	x64

Then you call the script:

```
GEN.x64.exe|$$CNT
MBX.$$CNT
ENR.
```

and you get this result (see picture below).



Syntax

GEN.[P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** Filename with or without path, pattern possible
P2 **-** Variable to return the result

Example

```
'***********************************
' GEN.-Demo
'***********************************
GEN.x64.exe|$$CNT
MBX.$$CNT
END.
```

Remarks

x64 tested.

Limitations:

-

See also:

- 1.7. Application and PID 137
- GPI. - Get Process Information 2893
- IEP. / NEAP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.5 GEP. - Get-Executable-Path

MiniRobotLanguage (MRL)

GEP. Command

Get Executable Path

Intention

Returns the Path of Applications for registered File-Extensions like ".txt" or ".doc" etc.
The given parameter is a filename, however the file does not need to exist.
The given filename needs an valid three letter File-Extension at its end.

Usage is as simple as:

```
: §§FIL=?desktop\MyWord.doc
GEP.§§FIL|§§PAT
DBP.The executable Path for .DOC-files is:$crlf$ §§PAT
ENR.
```

The result will look somehow like this:

```
[22:01:44] [15] The executable Path for .DOC-files is:
 C:\Programme\Microsoft Office\OFFICE11\WINWORD.EXE
[22:01:44] MRE: Robot ended: test.mrt
```

This way you can find out, which program is registered on a target system to a file-extension. Try this on your system:

```
: §§FIL=MyWord.html
GEP.§§FIL|§§PAT
DBP.The executable Path for .html-files is:$crlf$ §§PAT
' You can use the result with other commands.
EXE.§§PAT|www.fa2.de
ENR.
```

Syntax

GEP.P1[|P2]

Parameter Explanation

P1 - Document-Filename (can't be a hyperlink)
P2 - Variable for Executable-Path. If omitted Result
will be put on TOS

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************************
' Example 1 - GEP.
'************************************************
GEP.?desktop\MR.txt
' Write output in Editor-Debug Area
DBP.The executable Path for Text-files is: $crlf$
$tos$
MBX.Ready
END.
'************************************************
' Example 2 - GEP.
'************************************************
' Variables and specialfolder can be used
: §§FIL=MyWord.html
GEP.§§FIL|§§PAT
DBP.The executable Path for .html-files is:$crlf$
§§PAT
EXE.§§PAT|www.fa2.de
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.7. Application and PID 137
- 3.3 Specialfolders 345
- EXR. - Executer-from-Registry 902
- EXO. - Execute Open Document 895
- EXS. - Execute Shell 905

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.6 GII. - Get-Installation-Information

MiniRobotLanguage (MRL)

GII. Command

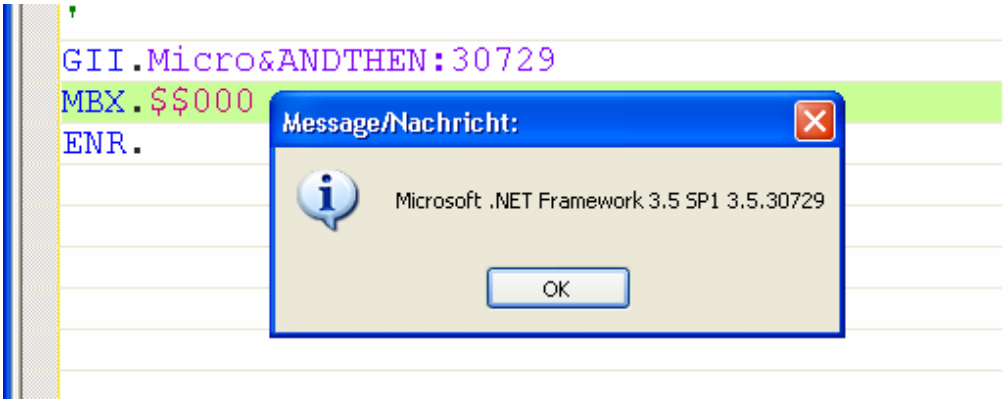
Get Installation Information

Intention

This command can be used to get the Uninstall-String for a specified application.

To find out if the application is installed, use `IAI.` See `IAI.` for more details.

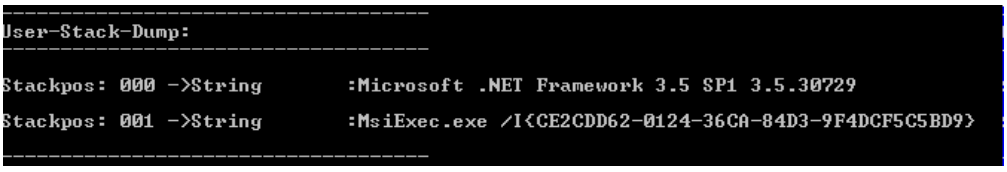
It will retrieve the complete Application name or the requested uninstall information, if they are available.



The following Script will retrieve the "modify String" of the MS Framework with the given Version-Number:

```
GII.Micro&ANDTHEN:30729|m
DMP.6
ENR.
```

This is the result on the Stack:



If we try this (note the "`|s`" at the end:

```
GII.Micro&ANDTHEN:30729|s
DMP.6
```

the the stack will be empty, because there is no silent Uninstallation for this patch. We can use a variable instead of the stack.

© 2023 Theo Gottwald * https://www.smart-package.com

```
GII.Micro&ANDTHEN:30729|$$RES|u
DBP.$$RES
ENR.
```

Andthe result is:

```
[12] Microsoft .NET Framework 3.5 SP1 3.5.30729
```

while the uninstall string itself will be on the stack.

```
-----------------------------------
User-Stack-Dump:
-----------------------------------
Stackpos: 000 ->String       :MsiExec.exe /I{CE2CDD62-0124-36CA-84D3-9F4DCF5C5BD9}
-----------------------------------
```

The command works transparently with both x32 and x64 applications.

Standard-Search Pattern 341 can be used.

Syntax

GII.P1[|P2][|P3]

Parameter Explanation

P1 – Application to test if installed
P2 – Variable for Result if found.
P3 - letter specifies what to look for.

s - return command-line for silent Uninstallation
u - return command-line for standard Uninstallation.
m - return command-line for standard Modify-Path (Add-/Remove Setup)
a - return Standard Application Name

Example

```
'*********************************************
' IAI./GII. - Sample
'*********************************************
'DBM.2
PRT.=================================================
IAI.office
 VAR.§§APP=$$000
 PRT. §§APP is installed.
ELS.
 PRT. No such Application installed.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
PRT.=====================================================
IAI.office|m
  VAR.§§APP=$$000
  PRT. §§NAM can be modified like this:$crlf$ §§APP
ELS.
  PRT. Application is not installed or cannot be modified.
EIF.
PRT.=====================================================
END.
```

Remarks

The command works transparently with both x32 and x64 applications.
Technically the command reads the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
```

to get these information.

`IAI.` as well as `GII.` as well as "InstaRec" build the Name of the Program using a Combination of the "`DisplayName`" and the "`DisplayVersion`" Properties. Therefore you can more easily separate which Software you want to deal with.

Limitations:

This command looks for standard information details at:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
```

If the application being tested for installation did not register this information, the command will not be able to retrieve installation information.

See also:

GEP. - Get-Executable-Path 2582
IAI. / NAI. - If-Application-Installed 2601
EXU. - Execute Uninstall 910

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.7 GLI. - Get Language ID

MiniRobotLanguage (MRL)

GLI. Command

Get User or system LanguageID

Intention

This command will get the language settings of the current user or system. Usage is simple:

```
' Will place the current users language settings on the TOS
GLI.
DBP.$$000
```

May print something like this:

```
12:14:14] [9] 1031
```

In the Editors debug window, if you are running on a german system.
To get deeper into international settings, you can supply parameters.
In fact, what is a german system?

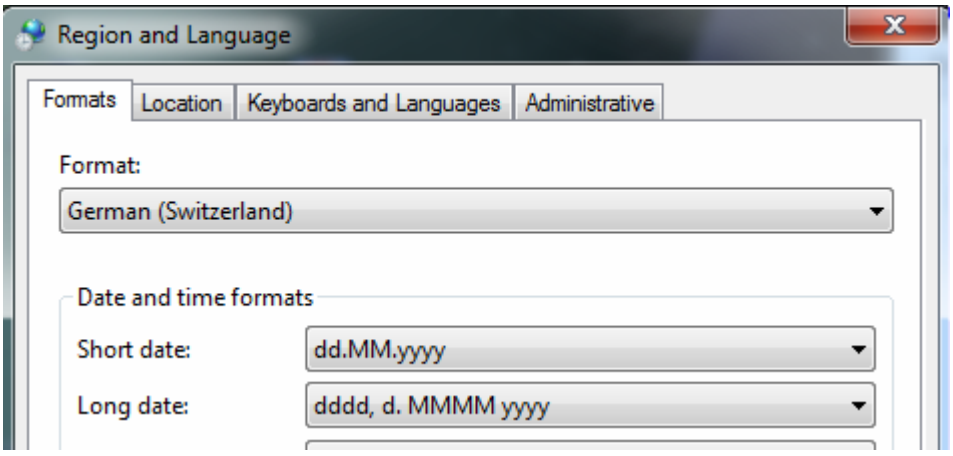
You can supply a variable for the result:

```
GLI.u|$$RET
DBP.$$RET
ENR.
```

If you need a very detailed locale, prefer using options `c` or `d`.

Where is the difference between these settings?

If you take a german windows and you set the Regional Format to Switzerland like this:



 © 2023 Theo Gottwald * https://www.smart-package.com

then you will get the following results:

```
' It uses german unicode settings
GLI.s
DBP.$$000
' result is 1031

' This shows you that its set to swiss
GLI.u
DBP.$$000
' result is 2055
```

Here is a table with some language-id's.

1025	Arabic	1041	Japanese
1028	Chinese - Traditional	1042	Korean
1029	Czech	1043	Dutch
1030	Danish	1044	Norwegian
1031	German	1045	Polish
1032	Greek	1046	Portuguese - Brazilian
1033	English	1049	Russian
1034	Spanish	1053	Swedish
1035	Finnish	1054	Thai
1036	French	1055	Turkish
1037	Hebrew	2052	Chinese - Simplified
1038	Hungarian	2070	Portuguese
1040	Italian	3076	Chinese - Hong Kong SAR

You can find a more complete list with Language-ID's here:
Language-ID's

Important:
To check if a system is of a specified language,
`ILI.`
is the command of your choice.

Syntax

GLI.[P1][|P2]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - (optioal) defines the method which is used to get the Language ID.

U - Returns the language identifier for the current user locale.That is the Format settings (for date, time, currency, etc.) of current user in Regional and Language Options in Control Panel.
This is the default option is no letter is specified.

S - Returns the Language Identifier for the system locale. In fact you get the is the "Language for Non-Unicode programs" of current user.

C - an alternative system is used to get the Language ID for user. This supports more international properties providing a more culturally appropriate user experience.

d - an alternative system is used to get the Language ID for the system (d), This supports more international properties providing a more culturally appropriate user experience.

P2 - (optioal) Variable for the result. If omitted, TOS is used.

Example

```
'-----------------------------
'
'-----------------------------
GLI.d|$$RET
DBP.$$RET
ENR.
```

Remarks

You can find a more complete list with Language-ID's here:

Language-ID's (Technet)

Often used Language-ID's

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ILI. / NLI. - If LanguageID 2163
- GLI. - Get Language ID 2587
- GFI. - Get File Information 1026
- GFD. - Get File Date 1023
- GFV. - Get File Version 1040

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.68.8 GOS. - Get OS Information

MiniRobotLanguage (MRL)

GOS. Command

Get Operation System Information

Intention

Get numerical and textual information about the Operating System on which the script is actually running.

```
GOS.
STS.DUMP
MBX.Ready
```

will leave the following Information on the TOS if it runs on Windows XP:

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->String       :Windows XP

Stackpos: 001 ->String       :Service Pack 3

Stackpos: 002 ->String       :01-16-2012     09:47:58
OsString =      Windows XP
OsLong =       9
     dwMajorVersion =       5
     dwMinorVersion =       1
OsFamilyString =      Windows NT, 3 = NT 3/3.1, 4 = NT, 5 = 2000/XP/2003 Server, 6 = Vista and Windows 7
OsFamilyLong =       2
     dwPlatformId =       2
dwBuildNumber =       2600
Build Name =
Service Pack =      Service Pack 3
Os Is 64 bit =       0

Stackpos: 003 ->LongInt      :9

------------------------------------
```

If the same script runs on **Windows 7/x64** the stack 369 will look like this:

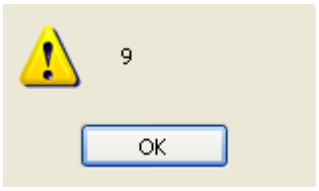
```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->String       :Windows 7

Stackpos: 001 ->String       :

Stackpos: 002 ->String       :01-17-2012     22:46:21
OsString =      Windows 7
OsLong =       12
     dwMajorVersion =       6
     dwMinorVersion =       1
OsFamilyString =      Windows NT, 3 = NT 3/3.1, 4 = NT, 5 = 2000/XP/2003 Server, 6 = Vista and Windows 7
OsFamilyLong =       2
     dwPlatformId =       2
dwBuildNumber =       7600
Build Name =
Service Pack =
Os Is 64 bit =       1

Stackpos: 003 ->LongInt      :12

------------------------------------
```

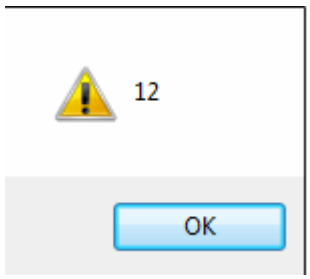
You can specify variables as parameters. In that case some of the information will be placed in these variables instead. Here is an example script:

```
GOS.$$NUM
STS.DUMP
MBX.$$NUM
```


© 2023 Theo Gottwald * https://www.smart-package.com

Again for windows XP, SP3 the stack will look just like before. However you get a value of 9 into the variable $$NUM. Now lets run the same script under Windows 7/x64. Again the stack is the same as before but you get a value of 12 (see right picture below) as numeric OS information.

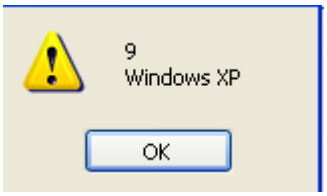


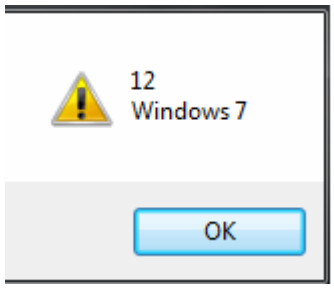


Lets see the last mode of operation, this is when we give two variables to GOS.

```
GOS.$$NUM|$$TXT
STS.DUMP
MBX.$$NUM$crlf$$$TXT
```

This is what we get for windows XP. And the picture on the right shows the result for Windows 7.





As we can see, the second variable contains the textual name of the operating system.

Syntax

GOS.[P1][|P2]

Parameter Explanation

P1 - Variable to take numeric OS-Information

You get these Numbers on the Stack/Var or in **P1:**

1 - Windows 3.1
2 - Windows 95
3 - Windows 98
4 - Windows ME

© 2023 Theo Gottwald * https://www.smart-package.com

```
 5  - Windows NT 3
 6  - Windows NT 3.1
 7  - Windows NT 4
 8  - Windows 2000
 9  - Windows XP
 10 - Windows 2003
 11 - Windows Vista
 12 - Windows 7
 13 - Windows 8
 14 - Windows 9
 15 - Windows 10

P2  - Variable to take textual OS-Information
```

Example

```
'*********************************************
' GOS-Example
'*********************************************
'
GOS.
PRT. OS-Version: $$000

GOS.$$NUM
PRT.numeric OS-information: $$NUM

GOS.-|$$TXT
PRT.Textual OS-information:$crlf$$$TXT

GOS.$$NUM|-
PRT.Textual OS-information:$crlf$$$000
PRT. Numerical Info: $$NUM

MBX.Klick me.
'-------------------------
END.
```

Remarks

To test if your script runs under a 64-bit OS, you can use the `IOS.x64` statement.

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- IOS. / NOS. - If-Operating System 2607

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.9 GSD. - Get Session Data

MiniRobotLanguage (MRL)

GSD. Command

Get Session Data (name and Info)

Intention

This command will get information about running OS Sessions. It may be of use especially for Windows Terminal servers. `GSD.` can return the number of sessions, as well as the name and state of a specified session ID.

`GSD.` may return one of these states for a given session ID:

```
0 -   User logged on to WinStation
1 -   WinStation connected to client
2 -   In the process of connecting to client
3 -   Shadowing another WinStation
4 -   WinStation logged on without client
5 -   Waiting for client to connect
6 -   WinStation is listening for connection
7 -   WinStation is being reset
8 -   WinStation is down due to error
9 -   WinStation in initialization
```

There are four way to use GSD.

1. `GSD.` without any parameter will place two values on TOS:

- the session identifier of the current physical console session on TOS.
- the number of sessions below TOS. Example:

```
GSD.
DBP.$$000
DBP.$$000
```

will return:under Windows XP (run from Editor)

```
0 - current Session ID
1 - total sessions on this machine
```

will return:under Vista and higher, for example (run from Editor):

```
1 - current Session ID
1 - total sessions on this machine
```

2. GSD. with one parameter, like this: `GSD.$$NUM`

© 2023 Theo Gottwald * https://www.smart-package.com

will return the session number in the given variable. Example:

```
GSD.$$NUM
DBP.$$NUM
```

3. `GSD.` with two parameters. Like this:

```
  GSD.session ID|$$RET
```

will return the run-state of the session with the given session id. Example:

```
GSD.0|$$RST
DBP.$$RST
```

4.GSD. with 3 parameters, like this:

```
GSD.session ID|$$RET|$$NAM
```

Will return the run-state as well as the session name of the session with the given session ID.

Syntax

GSD.[P1][|P2][|P3]

Parameter Explanation

P1 - ***(optional)*** **- Session ID,** if omitted the Session ID for the current session is been evaluted and placed on TOS. See examples above.

`GSD.` may return one of these states for a given session ID:

```
     0 -   User logged on to WinStation
     1 -   WinStation connected to client
     2 -   In the process of connecting to client
     3 -   Shadowing another WinStation
     4 -   WinStation logged on without client
     5 -   Waiting for client to connect
     6 -   WinStation is listening for connection
     7 -   WinStation is being reset
     8 -   WinStation is down due to error
    9 -   WinStation in initialization
```

P2 - ***(optional)*** - Variable for Result. For Details see examples above.

P3 - ***(optional)*** - Variable for Result. For Details see examples above.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'
GSD.
DBP.$$000
END.
```

Remarks

-

Limitations:

-

See also:

- SID. - Security Identifier 2114

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.10 GUS. - Get Usage State

MiniRobotLanguage (MRL)

GUS. Command

Get Desktop name and Info if script runs without user logged on

Intention

This command will get the current desktop of the script as well as the current input-desktop of the user.
This way the command can find out, if a user is logged on or not.

There are 3 ways to use GUS.

1. `GUS.` without any parameter will place a values on TOS:

0 - the user is currently logged on
1 - the script runs at logon screen
technically: the input-desktop is NOT the desktop which the script runs on.

2. `GUS.` with one parameter. Like this:

will return the result of the operation on TOS. Example:

```
GUS.log-off state
DBP.$$000
```

3.GUS. with 2 parameters, like this:

```
GUS.log-off state|$$RET
```

Will return the result of the operation in the given variable. Example:

```
GUS.log-off state|$$RET
DBP.$$RET
```

Hint:
To find out, if a user has just locked his client, you can use `IEP.` to check if a process "`explorer.exe`" is currently running.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

GUS.[P1][|P2]

Parameter Explanation

P1 - (optioal) Subcommand, one of the following:

"acs-id","ai"
returns The session identifier of the session that is attached to the physical console. If there is no session attached to the physical console, (for example, if the physical console session is in the process of being attached or detached), the return value is a value that is not one of the valid session-id's which you can get using the GSD.- command.

"ir","is
The return code to this subcommand consists of two parts:
Part I: If the calling process is associated with a Terminal Services client session, the return value is 1. If the calling process is associated with the Terminal Services console session, the return value is 0.
Part II: is used in a Terminal Services environment to determine if the current Terminal Server session is being remotely controlled. Its value is 1 if the current session is remotely controlled; otherwise, 0.

"dn","desktop name"
Returns the name of the desktop which is the desktop of the current running script.

"id","input desktop"
Returns the name of the desktop that is currently connected with the mouse and screen.

"log-off state","los"
Checks if the two desktops have the same name - or not. In that case, its possible that the station is logged off. Note that this will only work as long as the user and the script run in the same session. If that is not the case, both could use desktops with the same name, while being different desktops. If necessary, check the session ID also, using GSD.

P2 - (optioal) Variable for the result. If omitted, TOS is used.

Example

```
'-----------------------------
'
'-----------------------------
GUS.log-off state|$$RET
DBP.$$RET
END.


'-----------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
'-----------------------------
SER.restart|dn:interactiv
DOL.1
  GUS.los
  PRT.los=$$000
  GUS.id
  PRT.id=$$000
  GUS.dn
  PRT.dn=$$000
  PAU.3
OOP.
ENR.
```

Remarks

There are huge differences about the handling of desktops and sessions, before and after windows Vista.
After Vista, all system services run in session 0, that does not have a interactive desktop. The user desktop is in session 1.
Before Vista, services and user-desktop were in session 0. Desktops will always have different names, unless they are not in the same session. Mostly there is a desktop named "default" and a desktop named "WinLogon" in each session.
If no user is logged on, then the Input-Desktop will possibly be the "WinLogon".

Limitations:

-

See also:

- GSD. - Get Session Data 2595
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.11 IAI. / NAI. - If-Application-Installed

MiniRobotLanguage (MRL)

IAI. Conditional Statement

If Application Installed

Intention

This command can be used to find out if a specified application is installed. It can also be used, to find out if an program has a specified "Uninstall-Option" like "silent Uninstall" or "Modify/Repair" or not. Standard-Search Pattern 341 can be used.

The most important usage is, to find out, if a specified application is installed.
The following script will do that:

```
IAI.mozil&ANDTHEN:Fire
  VAR.$$APP=$$000
  DBP. $$APP is installed.
ELS.
  DBP. No such Application installed.
EIF.
```

May result in this output:

```
[8]  Mozilla Firefox 9.0 (x86 en-GB) is installed.
```

if that program is really installed on that particular PC. You can keep it simple. You can just give a part of what you want.

```
IAI.firefo
  STS.DUMP
ELS.
  DBP. No such Application installed.
EIF.
MBX.!
```

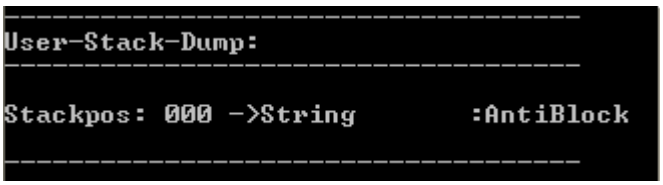
Take a look at these programs:

They both contain the term "block". Lets search them. If we just write:

```
' Without any option we search for a program NAME
IAI.block
  STS.DUMP
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

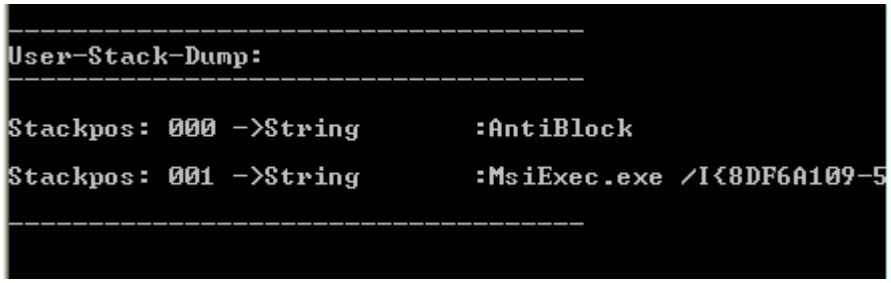
We search for ANY program with that term "block" in name. Thats the way how we check, if a program is installed. See the result in the picture below.



Now we want to know **if that program has a option to be uninstalled**. We just add an "`|u`" for this.

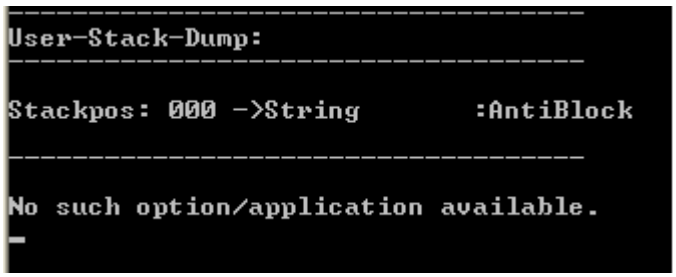
```
IAI.block|u
  STS.DUMP
EIF.
```

The result is as expected. Because that option is available.



Now we add a check, if an option for a silent uninstall is available.

```
IAI.Antiblock
' $$000 is the same as $tos$
  IAI.$$000|s
    STS.DUMP
  ELS.
    STS.DUMP
    PRT.No such option/application available.
  EIF.
EIF.
```



This way you can check, if an application is installed, and also if it has an option to be uninstalled.

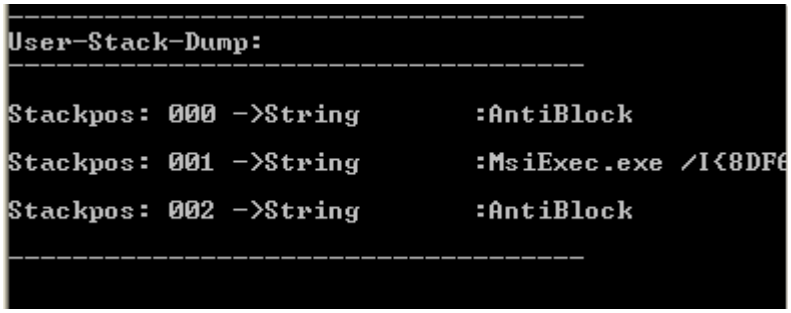
You can dig more deep using `IAI`. You can say "Tell me if there is ANY available Uninstall Option" for this program. We use the "`|a`" Option for this.

```
IAI.Antiblock
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
  IAI.$tos$|a
    STS.DUMP
  ELS.
    STS.DUMP
    PRT.No such option/application available.
  EIF.
EIF.
```

Now the robot looks if there is a
- silent-uninstall available.
- a normal uninstall-string available.
- a repair/modify string available
if any of these is found (in that order!), it is returned.



You can also say "look only for the silent and the uninstall string". Like this:

```
IAI.$tos$|su
```

Then it will only return the silent or the uninstall string. Or evaluate to FALSE (jump to ELS.) if they are not available.

Note that there is a priority, if more then one of these strings is available.
The order of search is always the same: silent,uninstall,modify. The first found will be returned.

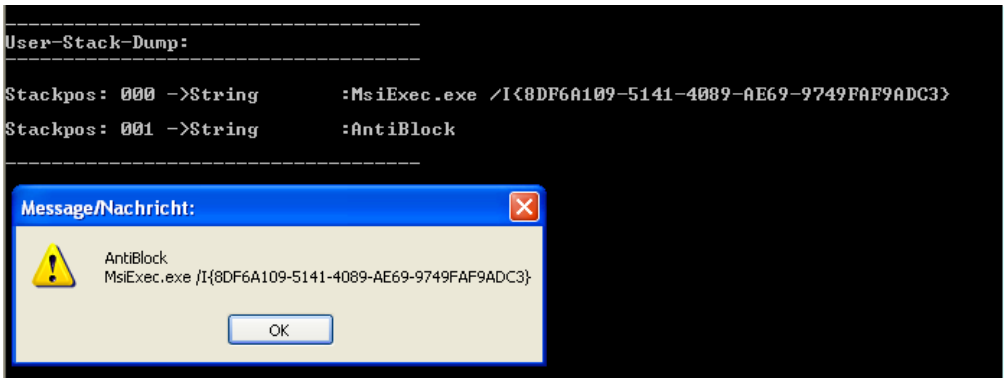
If you specify one of the flags u,m or s, then the specified "uninstall-string" must be available to be TRUE. Otherwise `IAI.` will decide FALSE, even if the application itself is installed.

There is one more thing here. Watch this script, and the result.

```
IAI.Antiblock
  IAI.$$000|$$TXT|a
    STS.DUMP
    MBX.$$TXT+$crlf$+$tos$
  ELS.
    STS.DUMP
    PRT.No such option/application available.
  EIF.
EIF.
```

As you can see in the picture below, the name of the found application can be directly returned to a variable, instead of the TOS (Top of Stack).

© 2023 Theo Gottwald * https://www.smart-package.com



NAI. is the negative form of this statement.

```
NAI.Antiblock
  GII.Antiblock
  STS.DUMP
  MBX.Ready
EIF.
END.

++++++++++++++++++++++++++++++++++++++++++++++++
```

IMPORTANT:

To verify, if a specified Patch is installed, you can not use IAI.
Instead use this Skript

```
$$HOT=KB4016871
$$TXT=?path\$$HOT
$$VAR=wmic qfe get hotfixid | find "$$HOT" > "$$TXT"
$$ZIE=?path\$$HOT.bat
CTF.$$ZIE|$$VAR
EXC.$$ZIE
$$PID=#pid#
WPT.$$PID
FIL.Length|$$TXT|$$RES
IVV.$$RES!0
  MBX.Patch $$HOT installiert.
ELS.
  MBX.Patch $$HOT ist nicht installiert.
EIF.
ENR.
```

Syntax

IAI.P1[|opt]
IAI.P1|P2[|opt]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** Application to test if installed
P2 **-** Variable for Result if found.

opt
```
 s - try command-line for silent Uninstallation
 u - try command-line for standard Uninstallation
 m - try command-line for Modify/Repair

 a - (any) get the first uninstall string available.
     this option will try the availability of the
     uninstall-Information in this order:
     - silent
     - uninstall
     - modify/repair
     It will return the first available uninstall string
     on the TOS.
```

Example

```
'*********************************************
' IAI./NAI. - Sample
'*********************************************
'DBM.2
PRT.=================================================
IAI.{office&ANDTHEN:Prof&ANDTHEN:2003}&AND:{&NOT:Visio}
 VAR.§§APP=$$000
 PRT. §§APP is installed.
ELS.
 PRT. No such Application installed.
EIF.
PRT.=================================================
NAI.office
 PRT. No such Application installed.
ELS.
 VAR.§§APP=$$000
 PRT. §§APP is installed.
EIF.
PRT.=================================================
END.
```

Remarks

The `IAI.` command works transparently with both x32 and x64 applications.
Technically the command reads the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\U
ninstall\
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

to get these information.

`IAI.` as well as `GII.` as well as "InstaRec" build the Name of the Program using a Combination of the "`DisplayName`" and the "`DisplayVersion`" Properties. Therefore you can more easily separate which Software you want to deal with.

Limitations:

This command looks for standard information details at:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
```

If the application being tested for installation did not register this information, the command will not be able to retrieve installation information.

See also:

- GII. - Get-Installation-Information 2584
- EXU. - Execute Uninstall 910

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.12 IOS. / NOS. - If-Operating System

MiniRobotLanguage (MRL)

IOS. Conditional Statement

If Operating System

Intention

Conditional Statement. Test if the actual system meets the OS criteria you can specify. This way you can tell your script to do something else on different OS. Usage is very simple. You just write IOS. and the OS you want to test for.

```
IOS.Win7
  DBP. We are in Windows 7!
EIF.
```

Also, you can specify multiple OS you want to support. In this case you just write them all separated by a comma.

```
IOS.WinXP,Vista,Win2K
  DBP. We are here.
EIF.
```

But life is not always so simple. Maybe your script will only run on anything below Windows Vista? In this case you just write

```
IOS.<Vista
  DBP. We are below Windows Vista
EIF.
```

And if Windows Vista is included, you can add a "=".

```
IOS.<=Vista
  DBP. We are below Windows Vista or in Vista.
EIF.
```

All of these features can be combined, for those seldom cases when needed. In this case Windowx XP and WIndows Vista would not be served.

```
IOS.=>Win7,<Win2K
  DBP. We are not in WinXP.
EIF.
```

Also instead of writing a OS-Name, you can use the OS number (below) and prefix it with a #Sign. Like this.

```
IOS.<#11
  DBP. We are below Windows Vista
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

You can also use the command to check if the OS is x64 or not. Please do not combine this with other features:

```
IOS.x64
  DBP. Yes its x64
EIF.
```

The negative Form of this statement is `NOS.`

Syntax:

```
IOS.P1
 ' Block is executed if condition is met.
ELS.
 ' Block is executed if condition is not met.
EIF.
```

Parameter Explanation:

P1 - (optional) can be:

```
For Windows 2000:
    "windows 2000","w2k","win2000","win2k","w2000"

For Windows XP:
    "winxp","wxp","xp","windows xp","win xp"

For Window 2003:
    "windows 2003 server","ws2003","w2003"

For Windows Vista
    "windows vista","vista"

For Windows 7:
    "windows 7","w7","win7"

For Windows 8:
    "windows 8","w8","win8"

For Windows 9:
    "windows 9","w9","win9"

For Windows 10:
    "windows 10","w10","win10"

For Windows 3.1:
   "windows 3.1","w31"

For Windows 95:
    "windows 95","win95","w95"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
For Windows 98:
    "windows 98","win98","w98"

For Windows ME:
    "windows me","winme","me"

For Windows NT 3:
    "windows nt 3","win nt3","nt3"

For Windows NT 3.1:
    "windows nt 3.1","win nt31","nt31"

For Windows NT 4:
    "windows nt 4","nt4","win nt4"

For any Windows NT:
    "windows nt","nt","win nt"

For any Windows 9x:
    "windows 9x","win9x","9x","w9x"

You can prefix the OS version with these Signs:

> - later then
= - equal
< - earlier then
```

to specify that all OS versions before or after an specified OS version are Ok.

```
IOS.<=Win7
  DBP. We are under Win7 or below!
EIF.
```

You can also specify a number for the OS. In that case prefix the number with a "#". Do not use spaces between the number and the "#".The syntax is #**(number of OS)**. Here are the numbers for the OS-versions.

1 - Windows 3.1
2 - Windows 95
3 - Windows 98
4 - Windows ME
5 - Windows NT 3
6 - Windows NT 3.1
7 - Windows NT 4
8 - Windows 2000
9 - Windows XP
10 - Windows 2003
11 - Windows Vista
12 - Windows 7
13 - Windows 8
14 - Windows 9
15 - Windows 10

For example: **#9** will be Windows XP.
#12 would be Windows 7.

```
' You could write
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 IOS.#12,#9
  PRT. We're running Windows 7 or WinXP!
 EIF.

' which is functional the same like if you write
 IOS.Win7,XP
  PRT. We're running Windows 7 or WinXP!
 EIF.
```

You can use this special feature to check if you are running under `WOW64` (x64 Windows)

```
IOS.x64
  DBP. We are running on a x64 OS.
EIF.
```

Example:

```
'*********************************************
' IOS-Example
'*********************************************
'
 IOS.VISTA,XP
  MBX. We're running VISTA or WinXP!
 EIF.

'-------------------------
END.


'*********************************************
' IOS-Example
'*********************************************
'
GOS.x64
DBP.$$000
DBP.$$000
IOS.x64
  DBP.Ja
EIF.
ENR.
```

Remarks:

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.6.1. Program Flow Control 121
- GOS. - Get OS Information 2591

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.13 IUA. / NUA. - If User is Administrator

MiniRobotLanguage (MRL)

IUA. / NUA. Conditional statement

Check if User is Admin

Intention

Can be used to verify that the current running script had admin privileges. Internally the command will check, if the current user is a member of the Administrator group.

Syntax

```
IAU. ... ELS. ... EIF.
```

Parameter Explanation

This command does not support any parameters.

Example

```
'-------------------------------------------------------
' IUA.
'
' Without Parameter!
'-------------------------------------------------------
'
IUA.
   DBP. User is Admin.
ELS.
   DBP. User is not Admin.
EIF.
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- GIP. - Get IP Adress 2094
- GUN. - Get User Name 2103
- GUS. - Get Usage State 2598
- GPU. - Get Process User 2898
- GUD. - Get User and Domain 2101

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.14 LIB. - Library

MiniRobotLanguage (MRL)

LIB. Command

LIBrary

Intention

Multiple Usage Command.

Runs a lot of system Applets, Wizzards etc. see List below.
You can use the phrase or the number as **P1**.

Example:

```
LIB.HotPlug
```

will do the same as

```
LIB.19
```

Syntax

LIB.P1[|P2][|P3]

Parameter Explanation

P1 **-** [VAR with] command-number or Command-name
P2 **-** optional, VAR with parameter, depends on command number.
P3 **-** optional, VAR with parameter, depends on command number.

```
  1  -  Control-Panel          -> Run Control-Panel
  2  -  Add-Printer            -> Run Add Printer Wizard
  3  -  View Printers          -> Open Printers CPL (XP,Win 7)
  4  -  Exchange-Profiles      -> Run Microsoft Exchange
Profiles
  5  -  Postoffice-Admin       -> Run  Microsoft Workgroup
                                  Postoffice Admin

  6  - Open With               -> Open-With Dialog, you can
choose
       the File Associations for a file. Filename as parameter
P2.
  7  - Run-Registered          -> Filename as parameter P2.
       Will run a file with the registered application.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  8  - View-Fonts              -> Open FONTS Folder (XP,Win 7)
  9  - About-Windows           -> Open "About Window Window"

  10 - Time-Zone               -> Pick a Time Zone Dialog
  11 - Time-Set                -> Set-Time Dialog
  12 - Lock-Computer           -> Lock the Workstation
  13 - Diskcopy                -> Diskcopy Dialog, only W2k,XP
  14 - System-Properties       -> Open "System Properties"
Dialog
  15 - Stored-Passwords        -> opens Dialog: "Stored
usernames
                                  and passwords"
  16 - INF-Install (filename)  -> Install INF-File P2
                                  Filename inkl. Path
  17 - Install-INF-Section     -> Install INF-Section P2
  18 - Default-Printer         -> Set Standard-Printer P2
  19 - HotPlug                 -> Shows "Safely remove
Hardwae"
                                  Dialog
  20 - Action-Center           -> Opens the security (XP) or
                                  "Action Center" (Win7)
  21 - Flip-3D                 -> Start Flip-3D (Only Vista
                                   and Win7)
  22 - Install-SCR             -> Install Screensaver P2
(untested)
  23 - Tired-PC                -> Suspend PC, P2 can be  0 or
1
  24 - Print-Html              -> Print HTML-File P2
  25 - New-Briefcase           -> Create new briefcase, uses
                                   Parameters P2 and P3 (XP
only)
  26 - Aero-ON                 -> Switch AERO-GLAS On (Vista
only)
  27 - Aero-OFF                -> Switch AERO-GLAS Off (Vista
only)
  28 - Format-Drive            -> Show "Format Drive" Dialog,
                                  XP only
  29 - Software-CPL            -> Show "Add remove programs"
dialog
  30 - ODBC-Settings           -> Show"ODBC-Data Source
                                  Administrator" Dialog
  31 - Open-URL                -> Open URL P2 in default
browser

  32 - Open-Scrap (filename)   -> Open a scrap document -
                                  1 Parameter. only XP.

There are more Options (till No. 68) but not yet documented.
```

Example

```
'*****************************************************
' EXAMPLE 1: LIB.
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
LIB.HotPlug
ENR.
```

Remarks

-

Limitations:

-

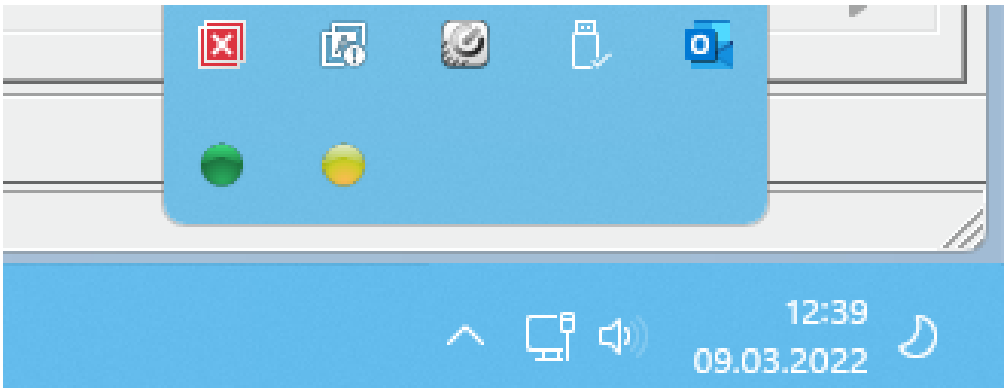
See also:

- CPL. - Control-Panel 2574

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15 Systray-Icon Commands

Using Systray-Icon Commands, you can give your Script an Systray Icon or change the Icon to show a changed State.



© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.1 SYT.Add Icon To

MiniRobotLanguage (MRL)

SYT.Add Own Icon

Add a Tooltip-Icon to your Script and bind it to another window-handle

Intention

This Command will Add an Icon in the Systray for your running Script.
The Icon will be registered to another window-Handle.
This way you can have multiple Icons within your Script.

Example:

```
$$ICA=?path\Blue.ico
' Locate any other window
STW.ct|ThunderRT6FormDC|Screenshot
' Get the Window-Handle
$$MYW=#hwn#
' Bind the Icon
SYT.ai|$$MYW|$$ICA
```

The Icon will automatically disappear when your Script ends. You can only bind Icons to valid Window-Handles.

Syntax

SYT.Add Icon To|P1|P2

Parameter Explanation

P1 **-** [VAR with] Window-Handle of the target Window. This can be any valid Window-Handle.
P2 **-** Filename of the Icon (.ico)

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
PAU.5
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.2 SYT.Add Own Icon

MiniRobotLanguage (MRL)

SYT.Add Own Icon

Add a Tooltip-Icon to your Script

Intention

This Command will Add an Icon in the Systray for your running Script.

Example:

```
$$ICA=?path\Blue.ico
SYT.oi|$$ICA
```

The Icon will automatically disappear when your Script ends.

Syntax

SYT.Add Own Icon|P1

Parameter Explanation

P1 **-** [VAR with] Icon-Filename

Example

```
'*******************************************************
' EXAMPLE 1:
'*******************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
PAU.5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.3 SYT.Change Any Icon

MiniRobotLanguage (MRL)

SYT.Change Any Icon

Change the Tooltip-Icon of your Script

Intention

This Command will change an Icon in the Systray to another Icon, for your running Script.
It will work for the Icons that have bin bound to another Window-Handle.
It will only work for Icons that have been created using the SYT.-Commands..

Example:

```
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
' Locate any Target-Window
STW.c|iexplore
' Get Handle
$$HWN=#hwn#
' Start with a Blue Icon
SYT.ai|$$HWN|$$ICA
PAU.1
'Change to Red-Icon
SYT.ca|$$HWN|$$ICB
```

The Icon will automatically disappear when your Script ends.

Syntax

SYT.Change Any Icon|P1|P2

Parameter Explanation

P1 **-** [VAR with] valid Window-Handle
P2 **-** [VAR with] Icon-File

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
PAU.5
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.4 SYT.Change Own Icon

MiniRobotLanguage (MRL)

SYT.Change Own Icon

Change the Tooltip-Icon for your Script

Intention

This Command will change an available Icon to another Icon in the Systray for your running Script.

Example:

```
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
' Start with a Blue Icon
SYT.oi|$$ICA
PAU.1
'Change to Red-Icon
SYT.co|$$ICB
```

The Icon will automatically disappear when your Script ends.

Syntax

SYT.Change Own Icon|P1

Parameter Explanation

P1 **-** [VAR with] Filename of Icon-File

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
PAU.5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.5 SYT.Has Icon

MiniRobotLanguage (MRL)

SYT.Has Icon

Verify if a Window-Handle has an Icon from our Script.

Intention

This Command will just check the internal Data to find out if an Window Handle has been used to bind an Icon to it from our Script.
The result will be zero if there is no Icon, or if the Window-Handle is invalid.
If there is a valid Icon under the given Handle, the result will be >0.

Example:

```
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
```

Syntax

SYT.Has Icon|P1|P2

Parameter Explanation

P1 **-** [VAR with] valid Window-Handle
P2 **-** Variable that will take the result. You can also specify "-" to have the result on TOS.

```
SYT.hi|6554616|$$RES
SYT.hi|6554616|-
```

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
PAU.5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.6 SYT.Own Tooltip

MiniRobotLanguage (MRL)

SYT.Own Tooltip

Add a Tooltip to your Systray Icon.

Intention

This Command will Add an Tooltip to the Systray Icon that has been created using the SYT.-Command.

Example:

```
$$ICA=?path\Blue.ico
SYT.oi|$$ICA
SYT.ot|Here is my Tooltip.
```

The Icon will automatically disappear when your Script ends.

Syntax

SYT.Own Tooltip|P1

Parameter Explanation

P1 **-** [VAR with] Text for Tooltip.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
PAU.5
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.68.15.7 SYT.Tooltip

MiniRobotLanguage (MRL)

SYT.Tooltip

Add a Tooltip to your Systray-Icon

Intention

This Command will Add an Toolltip to the Systray Icon for your running Script.

Example:

```
$$ICA=?path\Blue.ico
' Locate any other window
STW.ct|ThunderRT6FormDC|Screenshot
' Get the Window-Handle
$$MYW=#hwn#
' Bind the Icon
SYT.ai|$$MYW|$$ICA
SYT.tp|$$MYW|Zhis is a Icon Tooltip.
```

The Icon will automatically disappear when your Script ends.

Syntax

SYT.Tooltip|P1|P2

Parameter Explanation

P1 - [VAR with] Valid Window Handle
P2 - optional, VAR with parameter, depends on command number.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
'
$$ICA=?path\Blue.ico
$$ICB=?path\Red.ico
$$ICC=?path\Green.ico
$$ICD=?path\Yellow.ico
SYT.oi|$$ICA
' Need to localize an Application window here
STW.c|explorer
$$MYW=#hwn#
SYT.ai|$$MYW|$$ICB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU.5
SYT.co|$$ICC
PAU.5
SYT.ca|$$MYW|$$ICD
SYT.hi|$$MYW|$$RES
DBP.Has Icon: $$RES
SYT.hi|#cbw#|$$RES
DBP.Own Icon: $$RES
SYT.hi|6554616|$$RES
DBP.Own Icon: $$RES
PAU.5
SYT.ot|Hallo Tooltip.
SYT.tp|$$MYW|Hallo App.
PAU.1
MBT.Hallo
ENR.
```

Remarks

-

Limitations:

-

See also:

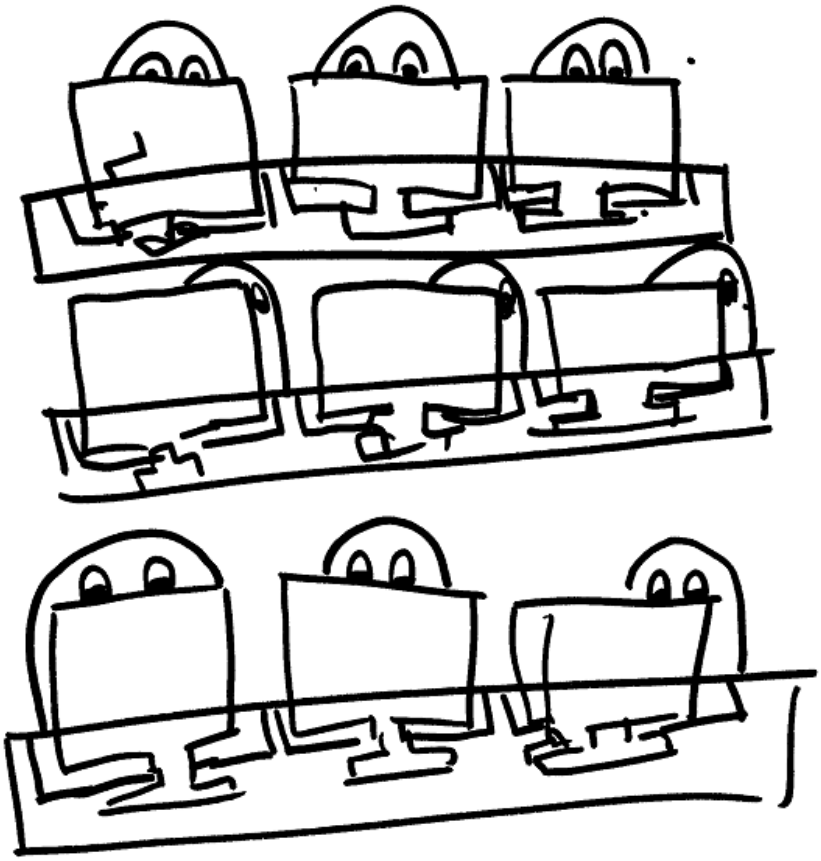
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.69 Parallel-Robot Operations

3.69.1 ! Smarty's Parallel Robot Operations

Smarty's Parallel Robot Operations



Intention

What are Parallel-Robot Operations and Background Processing good for?

Take a look at the picture above. As you can see, many "Smarty's" work at the same time.
Having several "Smarty's" work at the same time can save you processing time.

Especially if you have a Multi-Core-CPU, then each of the Tasks uses his own CPU-Core and therefore makes perfect use of your hardware.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Using the Parallel-Robot feature, you can run a lot of things in the background. Some examples are:

- start large applications in the background
- check and handle complicated pop-up-windows in the background (prefer `DBC.`)
- split large scripts in smaller portions that run parallel
- use all cores of your computer (Multi-Core/CPU usage)
- have things done in shorter time

Is there a difference between a Parallel-Robot and a Background Operation?

Yes, there is a technical difference. A Parallel Robot will run as a new process with an own PID. It will be visible in the Windows Task Manager as a separate process.
A Background Operation starts as a so called "thread" and therefore has the same `PID`.
Background Operations are more efficient in terms of resources.
There is no problem to have 100, 200 or 400 and maybe more Background Operations running at the same time.
Parallel running robots need more system resources therefore you will not use so many under normal circumstances. The main difference is this:
A Background Operation is already defined and somehow "hard-wired". Therefore it will give you an Operation Result - or not.
A Parallel Robot is not hard-wired, it will do just what you want it to do. It will only give you an Operation Result, if you use the `PRS.`-command. We'll come back on this topic later.

How can i start a Parallel-Robot?

It's easy. Technically it is just a pair of `PRR.` and `PRE.` A short example could look like this:

```
PRR.
  ' Script that the parallel-robot will execute
  ' This is the necessary END for the Parallel robot
  END.
PRE.$$RET
' Here the Main-Robot continues with his script
' $$RET contains the window-handle of the Parallel-Robot
```

There are different sorts of Background-Jobs and Parallel Robots. For most of them, you can use the commands:

- `WRS.` - Wait until the Process has ended / is ready.
- `GRS.` - Get the Result of that Bavkground process, if there is any.
- `DRS.` - Drop the result of a Background-Operation.
- `IRS. / NRS.` - Check if an Result is waiting for you
- `JRS. / JNS.` - Jump if an Result is/not available
- `PRS.` - Set the Result of an Parallel-Robot Operation.

Can i receive an result from calculations inside parallel running Robots?

Yes - you can!

© 2023 Theo Gottwald * https://www.smart-package.com

Second answer: In case of an Parallel-Robot you have to sent it your self - of course. In case of other Background-Operations, (like `SMB.`) the command will sent the result of the operation by itself automatically.

You can receive a calculation Result from a Parallel-Robot, using the Ticket-Number and the commands `WRS. / GRS.` etc. See this example:

```
' We start the Parallel Robot
PRR.
  ' This is the Result the Parallel-Robot sends back to us.
  ' Note that we did not specify an ticket number in P1,
  ' because it will be sent to his "father-robot" in this
case.
  PRS.|Hallo!
  ENR.
PRE.
' This is the Stack after the Parobot was started
STS.DUMP
' We wait for the Parobot to end.
WRS.
' And get the calculation result
GRS.
' This is the Stack after the Parobot sent his result
STS.DUMP
MBX.!
ENR.
```

To see, how you get an Result from a Background-Operation, take a look at

SMB. - Send Mail Background 799

or

MPB. - Music Play Background

Can i receive an result from multiple parallel running Robots?

Yes, here is an example:

```
PRR.
  GUT.Please give Result for Robot 1
  PRS.|Result from Robot 1 is: $$000
  ENR.
PRE.
$$ROA=$$001
PRR.
  GUT.Please give Result for Robot 2
  PRS.|Result from Robot 2 is: $$000
  ENR.
PRE.
$$ROB=$$001
STS.DUMP
WRS.$$ROA,$$ROB
GRS.$$ROA
GRS.$$ROB
STS.DUMP
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

You will need to put some text in the two input-boxes, when you run that example.

Important Note:

The parallel-robot will inherit all variables that the starting robot knows at the starting time.

In other words:
All variables and Stack-content will be available on the Parallel-Robot immediately. But from then each robot continues his work in his own address-space and with its own variables. Technically it looks like this:

```
$$MYA=1001
PRR.
  ' Script that the parallel-robot will execute
  ' This is the END for the Parallel robot
  MBX.$$MYA = 1001
  $$MYA=99
  END.
PRE.$$RET
' Here the Main-Robot continues with his script
' Result will be 1001
MBX.MYA
```

The Main Robot will not execute anything that is between the `PRR.` and the `PRE.`

And how do i know that the operation was finished and there is an Result available?

There are several ways to find that out. The easiest is to simply wait for the background Operation to finish, using the `WRS.`-command (Wait for Result). You can also use the `IRS. / NRS.` - Conditional statement.
And theoretically you could call `GRS.` and check the timeout-flag after that. If it was set, there was no such result available.

Important Note:
Any result behind an "ticket-number" is stored as long, until you drop it using `DRS.` You do not need to wait until the Operation is finished, to drop a result. If you do not need the result of an Operation, you can drop it immediately after you started the background-task. In this case, the robot will internally note that the result is been dropped and drop it immediately when it becomes available.

There is an exception:
Of course you only need to drop the Result of an Parallel-Robot (started with `PRR. / PRE.`) , if you use the `PRS,`-command to return a Result. If no Result is returned, it does not need to be dropped.

Use these commands together with any Background Operation or Parallel-Robot Operation:

- `GRS.` - Get Result of Operation
- `WRS.` - Wait until (one or more) Operation(s) is/are finished

© 2023 Theo Gottwald * https://www.smart-package.com

- DRS. - Drop Result of Operation if you do not need it.
- IRS. / NRS. Check if an Result is available
- JRS. / JNS. - Jump if an Result is/not available

Use this command:
- PRS. - Set the Result of an Parallel-Robot Operation.

only for Parallel Robots, to set the Operation Result..

Background Operation Example, using SMB.:

```
' Assume that a file "?path\MailConfig.dat"
' is available
SMP.
' It makes mostly sense with attachements
' SMP.attachment=?path\Send-Mail.zip
' Send the Mail, get the ticket number
SMB.
' If we do not specify a ticket number,
' WRS. will automatically remember the last ticket number
' and use it.
WRS.
' If we do not specify a ticket number,
' GRS. will automatically remember the last ticket number
' and use it.
GRS.
DBP.$$000
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
' and use it.
' DRS. free's the Memory location that was used to hold the
' Result of the background operation.
DRS.
ENR.
```

Here is an Sample that shows a Parallel-running Script on a "Private Desktop".
Please note that each PD has its own address space.
Therefore an Robot can not easily communicate with another Robot on a PD.
Several communications features may not work on a PD.
However the ! Smarty's InterKOM Technology 2116 should work.

```
 ' This Script will create Desktop and start a
 ' new Script on that Desktop using the PRR.-Command
 '
 PDC.Create Desktop with Explorer|DTF
 PDC.Switch to Desktop|DTF
 PRR.0|DTF
   MBX.I am on the DTF-Desktop
   PDC.Goto Default
 PRE.
 MBX.End the Script
 ENR.
```

Limitations

© 2023 Theo Gottwald * https://www.smart-package.com

The Parallel-Robot does NOT inherit the E-Mail Configuration for `SMA./SMB`.
Remember that you can send E-Mails in the background using `SMB.` without explicit starting a Parallel Robots.

There are no limitations about Parallel-Robots starting other Parallel Robots, or Background-Tasks.

There are limitations, running Parallel-running Script's on a "Private Desktop".
Each PD has its own address space.
Therefore an Robot can not easily communicate with another Robot on a PD.
Several communications features may not work on a PD.
However the ! Smarty's InterKOM Technology 2116 should work.

See also:

- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- GRS. - Get Result 799
- SMB. Send Mail Background 799

© 2023 Theo Gottwald * https://www.smart-package.com

3.69.2 PRE. - Parallelrobot-Run-End

MiniRobotLanguage (MRL)

PRE. Command

Parallel Robot run End

Intention

The command PRR. can be used to start a parallel running robot. For this purpose it starts a block containing the script for the new robot. The PRE. command will end this script-block.
PRR. and PRE. must always be used in Pairs. They can be nested, a Parallel-Robot can start another parallel robot and so on.
The parallel-robot will knows all variables that the starting robot knows at the starting time.
From that time each robot works in his own address-space and with its own variables.
Technically it looks like this:

```
PRR.
  ' Script that the parallel-robot will execute
  ' This is the END for the Parallel robot
  END.
PRE.$$RET
' Here the Main-Robot continues with his script
```

The Main Robot will not execute anything that is between the PRR. and the PRE.

The PRE. is not an END.-command for the Parallel-Robot.
The PRE. just tells the Main robot where to continue, and it gives the communication handle of the parallel-robot to the main-robot.If you want the Parallel-Robot to end at the PRE., place an END. command there. If you do not place an END. command there, the parallel-robot will walk over the PRE. into the Main-robots-script and continue running it.

In fact the Parallel-Robot starts with a 1:1 copy of the full script including all subroutines. The parallel robot will also start with a copy of all variables that the main robot had at this time. In fact he is more a "Clone" then a brother to the main-robot. As a result, a parallel robot can call all subroutines in the script, because he has a complete 1:1 copy of the script..

More Infos:
You can send variables to another running robot using PRV. Unless the Robot is on another "Private Desktop".
You can wait for a variable from another running robot using WFV.
You can set a global Signal that multiple running Robots will see, using SGS. and check if such a signal exist, using IGS.

Here is an Example that shows how you can wait for 5 Parallel Processes to end.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Click on the 5 Messageboxes and see what happens.

```
VAO.$$TIB=
FOR.$$LOP|1|5
' Run 5 Robots
  PRR.0
    MBX.!
    END.
  PRE.$$TIC
'$$TIC is the Window-Handle of the new Robot for use with PRV.
' The Kom-Handle is on TOS for use with WRS.
  VAR.$$TID=$tos$
    ' Collect the Tickets separated with ","
  $$TIB=$$TIB,$$TID
  PRT.$$TIB
NEX.
' Wait until they end
WRS.!$$TIB
ENR.
```

Syntax

PRE.[P1][|P2]

Parameter Explanation

P1 - (out) - will be given the WIndow-Handle of the started Robot. Which is the same as a window-handle. The Communicationshandle will be placed on TOS for use with WRS..

P2 - (in) - Timeout in sec. to wait for the robot to start.

```
-----------------------------------
User-Stack-Dump:
-----------------------------------
Stackpos: 000 ->LongInt      :658028
Stackpos: 001 ->String       :H658028
-----------------------------------
```

You will also find these values on the Stack:
1. Window-Handle-New Robot.
2. On the second stack place is the "Ticket-Number" for use with WRS. / GRS. / IRS.
technically, its the same number, but prefixed with a "H".

Example

```
'*********************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' PRR./PRE./PRV. - Sample 1
' Shows that the Parallel-Robot starts with
' all variables as they have been at starting-time
' of the starting robot.
' It can also use all Subroutines in this file.
'*******************************************
VAR.$$INP=0
VAR.$$TXT=Peter
PRT. I am the Main Robot.
'-------------------------------------------------------------
PRR.
  PRT.I am Parallel-Robot-Nr. #pri# from #prc#.
  PRT.I am Son #pri# of $$TXT.
  :Loop
  WFV.$$INP|10
  PRT. Variable set to: $$INP
  GSB.wait
  'JIV.$$INP=1|out
  JMP.Loop
  :out
  END.
PRE.$$RET|0
GRS.$$RET|$$RES

'-------------------------------------------------------------
PRT.KOM-H: $$RET
PRT.I have #prc# running parallel robots and PlugIns.
PAU.1
FOR.$$MYL|1|300
  VAN.$$INP=1.342587459
  PRV.$$RET|$$INP
  VAC.$$INP=12,31,(2*45),(5+67)
  PRV.$$RET|$$INP
  VAP.$$INP=142,32
  PRV.$$RET|$$INP
  PAU.0.23
NEX.
' This will end the parallel robot
VAN.$$INP=1
PRV.$$RET|$$INP
PAU.2
DMP.
MBX.!
END.
'-------------------------------------------------------------
:wait
PAU.0.5
RET.
'-------------------------------------------------------------
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The Parallel-Robot starts with the full script. Therefore do not forget the `END.` before the `PRE.` Otherwise the Parallelrobot will just walk over the `PRE.` into the Script of the starting robot. Then both will try to do the same script at the same time.

Limitations:

It is possible to start more then one Parallel-running robot, **actually up to 8**.
Each robot uses quite some system resources/memory therefore you should not run too many, depending on your CPU number and memory capacity.
Each robot runs independently from the moment he starts with his script.

It may or may not be possible to use this feature to install multiple software products at the same time. This depends on the used installer and the setup-program.

See also:

- ! Smarty's Parallel Robot Operations 2632
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- PRV. - Parallel-Robot send Variable 2651
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861
- WFM. - Wait-For-Message 2140
- TWM. - Text-Window-Message 2136

© 2023 Theo Gottwald * https://www.smart-package.com

3.69.3 PRR. - Parallel-Robot-Run

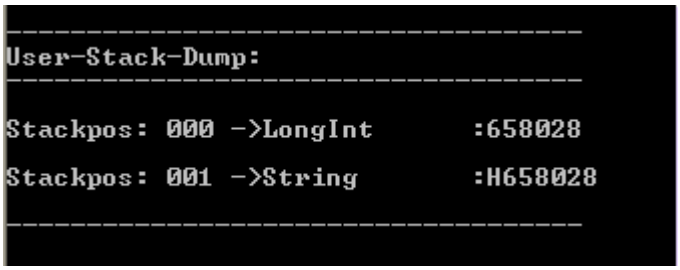
MiniRobotLanguage (MRL)

PRR. Command

Parallel Robot Run

Intention

The command `PRR.` can be used to start a parallel running robot. For this purpose it starts a block containing the script for the new robot. The `PRE.` command will end this script-block.
`PRR.` and `PRE.` must always be used in Pairs. They can be nested, a Parallel-Robot can start another parallel robot and so on.



This is the stack, after a successful call to `PRR. / PRE.`
At the top-level it contains the Window-Handle of the just started Robot.
Below is a valid "Ticket-Number" that can be used together with `GRS. / WRS. / IRS. / PRS / JRS. etc..`

Parameters:
Setting **P1** to 0 (end or 1 (survive), tells the Parallel running robot if he should end, if his father dies, or if he should survive anyway. If you omit **P1**, the parallel-robot will also end, when the robot that started it, ends.

More Infos:
You can send variables to another running robot using `PRV`.
You can wait for a variable from another running robot using `WFV`.
Both of these may NOT work if the Robot runs on another Desktop.

You can set a global Signal that multiple running Robots will see, using `SGS.` and check if such a signal exist, using `IGS.`

Here is an Sample that shows a Parallel-running Script on a "Private Desktop".
Please note that each PD has its own address space.
Therefore an Robot can not easily communicate with another Robot on a PD.
Several communications features may not work on a PD.
However the ! Smarty's InterKOM Technology 2116 should work.

```
' This Script will create Desktop and start a
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' new Script on that Desktop using the PRR.-Command
'
PDC.Create Desktop with Explorer|DTF
PDC.Switch to Desktop|DTF
PRR.0|DTF
  MBX.I am on the DTF-Desktop
  PDC.Goto Default
PRE.
MBX.End the Script
ENR.
```

Here is an Example that shows how you can wait for 5 Parallel Processes to end.
Click on the 5 Messageboxes and see what happens.

```
VAO.$$TIB=
FOR.$$LOP|1|5
' Run 5 Robots
  PRR.0
    MBX.!
    END.
  PRE.$$TIC
  ' Collect the Tickets separated with ","
  $$TIB=$$TIB,$$TIC
NEX.
' Wait until they end
WRS.!$$TIB
ENR.
```

Syntax

PRR.[P1][|P2]

Parameter Explanation

P1 **-** `(in) - 1/0 shall parallel-Robot survive when the starting robot ends?`

P2 **-** `(in/optional)` - Name of the existing "Private Desktop" where the Script shall be started.
if omitted, the Script will always start on the Desktop where the "Father-Script" is running.

Example

```
'*********************************************
' PRR.-Sample
'*********************************************
'
PRR.
  PAU.5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  KPN.TestApp_01.exe|a
  ENR.
PRE.
' EXB. - Beispiel
VAR.$$LNK=?exepath\Sample Apps\TestApp_01.exe
EXB.$$LNK
DBP.Exit-Code: $tos$
ENR.


'*********************************************
 ' This script will create a Private Desktop
 ' and run a Script on that PD.
 ' It will use PRR. for this.
 ' Important: The Messagebox will appear on the PD!
'*********************************************
 GSB.Test
 PDC.Create Desktop no Explorer|DTB
 PRR.0|DTB
   GSB.Test
   ENR.
 PRE.
 MBX.!
 ENR.
 '---------------------------------------
 :Test
 ICD.DTB
   MBX.Running on DTB.
 ELS.
   MBX.Not running on DTB.
 EIF.
 RET.
 ENR.


'*********************************************
' PRR./PRE./WFV. - Sample 1
' Shows that the Parallel-Robot starts with
' all variables as they have been at starting-time
' of the starting robot.
' It can also use all Subroutines in this file.
'*********************************************
VAR.$$INP=0
VAR.$$TXT=Peter
PRT. I am Robot-Nr. #pri# from #prc#.
'-----------------------------------------------------------
PRR.
  PRT.I am Parallel-Robot-Nr. #pri# from #prc#.
  PRT.I am Son #pri# of $$TXT.
  DOL.1
    : $$INP=99
    PRT.Waiting for Variable
    GSB.wait
    PRT.Input=$$INP
    JIV.$$INP=0|out
    WFV.$$INP
    PRT. Variable set to: $$INP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    JIV.$$INP=0|out
  OOP.
  :out
  PRT.Parallelrobot Ending
  PAU.2
  END.
PRE.$$RET|0
PRT. I am Robot-Nr. #pri# from #prc#.
'-------------------------------------------------------------
PAU.3
FOR.$$MYL|1|10
  GSB.wait
  VAN.$$INP=1.342587459
  PRV.$$RET|$$INP
NEX.
' END Paralel Robot
VAN.$$INP=0
PRV.$$RET|$$INP
PAU.8
DMP.
END.
'-------------------------------------------------------------
:wait
PAU.0.1
RET.
```

Remarks

The Parallel-Robot starts with the full script. Therefore do not forget the END. before the `PRE.` Otherwise the Parallel robot will just walk over the `PRE.` into the Script of the starting robot.
Then both will try to do the same script at the same time. Depending on the Setting in the `PRR.` command,the parallel running robots will automatically be killed when the main-script that started the robot, ends.

Limitations:

It is possible to start more then one Parallel-running robot, actually up to 8.

Each robot uses quite some system resources/memory therefore you should not run too many, depending on your CPU number and memory capacity.

Each robot runs independently from the moment he starts with his script.

When using `PRR.` the Parallel-Robot will not inherit the e-Mail Configuration (for use with `SMA.`) from the Starting Robot. To send Mails in the Background, use `SMB.` instead.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- PRV. - Parallel-Robot send Variable 2651
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861
- WFM. - Wait-For-Message 2140
- TWM. - Text-Window-Message 2136

© 2023 Theo Gottwald * https://www.smart-package.com

3.69.4 PRS. - Parallel Robot Set Resukt

MiniRobotLanguage (MRL)

PRS. - Parallel Robot Set Result

Set the Result from a Parallel Running Robot Operation

Intention

Using `PRR.` and `PRE.` you can start multiple Robots to do several things at the same time.
It's just like you have a business and several employees working at the same time for you.

Now comes the time that you want to know what they were doing the whole day. You want the "Results". How do they tell you the results of their jobs?

There are two ways, a parallel running robot can send his results to the Main Script. First can use `PRV.` to simply change a variable in the Main Script to whatever value you want.
Second you can place a "Result" with your "Ticket Number" in the Main Script, and that is what `PRS.` is good for.

The system works like this:

1. Start a Parallel Robot using `PRR.` and `PRE.` and get a "ticket number"
2. keep the ticket number in a variable
3. do whatever you want ...
4. The Parallel Robot will process its script and get a Result.
5. And call `PRS.` this will transfer the Result to the Main Script
6. The Mainscript will call `WRS.` with that "ticket number" as parameter
7. `WRS.` will wait until the background Operation has finished
8. and then call `GRS.` to get the result of the Parallel Robot

Hint: If you have just one Parallel-Robot running, you do not need to specify an ticket number. The system always remembers the and uses the ticket number from the last background operation.

Example:

```
' We start a Parallel Robot
PRR.
   ' Do any Calculation or whatever here
   VAL.$$CAL=9*99
   PRS. My Result is $$CAL
   ENR.
PRE.
' If we do not specify a ticket number,
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' WRS. will automatically remember the last ticket number
' and use it.
WRS.
' If we do not specify a ticket number,
' GRS. will automatically remember the last ticket number
' and use it. Here we get the Result from the Parallel Robot.
GRS.
DBP.$$000
' If we do not specify a ticket number,
' DRS. will automatically remember the last ticket number
' and use it.
' DRS. free's the Memory location that was used to hold the
' Result of the background operation.
DRS.
ENR.
```

How do i send / collect results from multiple parallel running robots/operations?

Here is an example:

```
PRR.
  GUT.Please give Result for Robot 1
  PRS.|Result from Robot 1 is: $$000
  ENR.
PRE.
$$ROA=$$001
PRR.
  GUT.Please give Result for Robot 2
  PRS.|Result from Robot 2 is: $$000
  ENR.
PRE.
$$ROB=$$001
STS.DUMP
WRS.$$ROA,$$ROB
GRS.$$ROA
GRS.$$ROB
STS.DUMP
MBX.!
ENR.
```

You will need to put some text in the two input-boxes, this text is the result from each of the two parallel robot operations.

As you can see, the `WRS.` will wait until both Results are available. And then present them to you.

Syntax

PRS.[P1][|P2]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - ***(optional)*** Variable that contains the own "Ticket-Number". In most cases, you can omit this parameter, then the own Ticket-Number is used.

P2 - ***(optional)*** Variable that contains the Result of the background job, that is to be
sent to the Main Script..

Example

```
'**********************************
' PRS.-Sample
'**********************************
'
PRR.
  GUT.Please give Result for Robot 1
  PRS.|Result from Robot 1 is: $$000
  ENR.
PRE.
$$ROA=$$001
PRR.
  GUT.Please give Result for Robot 2
  PRS.|Result from Robot 2 is: $$000
  ENR.
PRE.
$$ROB=$$001
STS.DUMP
WRS.$$ROA,$$ROB
GRS.$$ROA
GRS.$$ROB
STS.DUMP
MBX.!
ENR.
```

Remarks

If you use `PRS.`, always drop the result later in the Main-Script, using `DRS.`.
Otherwise it will be kept in Memory until the Script ends.

Limitations:

There is no technical limit in the size of the result, other then available Memory and System Resources.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's Parallel Robot Operations 2632
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- GRS. - Get Result 430
- DRS. - Drop Result 427
- WRS. - Wait for Result 436
- JRS. - Jump if Result 2215
- JNR. - Jump if no Result 2197
- IRS. / NRS. - If Result available 434

© 2023 Theo Gottwald * https://www.smart-package.com

3.69.5 PRV. - Parallel-Robot send Variable

MiniRobotLanguage (MRL)

PRV. Command

Parallel Robot send Variable

Intention

This command can be used to send variables to another running robot.

Its best if the parallel robot has been started using the `PRR./PRE.` command, because then you have his *communication-Handle*.

However parallel robots can be started anyway, as long you get his communication handle. The communication handle of a particular robot instance is inside the system-variable: #cbw# 334 .

Advanced technical note: If you have the PID 137 of a parallel-robot, you can locate his (invisible) window using `STW.` and the the "p" flag and then use the window-handle as *communication-handle*.

More Infos:
You can wait for a variable from another running robot using the command `WFV.` However, this is not necessary. A running robot will receive the value of a variable that is sent to him with or without `WFV.`, anyway. This processing takes place in the execution time between two command-lines.

Here is a very simple example, which looks larger then it is, because its heavily documented.

```
' We set the variable $$NUM to 1 in the Main-Robot
' The Parallel robot will inherit all variables from the Main
' Robot on start this way he knows that $$NUM=1 also.
: $$NUM=1
' Here we start the Parallel-Robot
PRR.
 ' This is an endless-loop (condition 1 is always true)
  DOL.1
    ' We make a small rest (variables are received in resting
time)
    PAU.0.1
' We check if the variable $$NUM changes its value to "0",
' if so we leave
    JIZ.$$NUM|out
' As long as $$NUM is not "0", we PRINT something and
' continue Looping.
    PRT.Running until i get zero ...
  OOP.
  ' Here we go when we got zero
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  :out
  ' Then you get a Message-Box
  MBX.Got a zero, leaving
  ' ANd we end the parallel robot
  END.
PRE.$$CMH
' This the way of the Main-Robot
' We let him rest for 2 seconds, then signal the Parallel
robot
' to end
PAU.2
' Now we have the Robot-Communication handle insde $$CMH
: $$CON=0
' We set $$NUM on the Parallel Robot to zero
PRV.$$CMH|$$NUM|$$CON
' We wait for 3 seconds and then
PAU.3
' End the script
END.
```

Syntax

PRV.P1|P2[|P3]

Parameter Explanation

P1 **-** (in) Window/Communicationhandle of a another Robots-Instance

P2 **-** (in) Name of a Variable which shall be sent to the other robot.

P3 **-** (in) Content which shall be sent to the other robot to be placed inside the given VAR-Name **P2**. If omitted content of **P2** is sent.

If only two parameters are given, the second parameter will be taken as variable-name and as content.

Therefore:
PRV.$$RET|$$INP
is equal to:
PRV.$$RET|$$INP|$$INP

Important: Please note that **P2 is taken "as is"**, it can not be inside another variable!

Alternatively use this:
PRV.$$WHN|$$NAM|$$CON

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*******************************************
' PRR./PRE./PRV. - Sample 1
' Shows that the Parallel-Robot starts with
' all variables as they have been at starting-time
' of the starting robot.
' It can also use all Subroutines in this file.
'*******************************************
VAR.$$INP=0
VAR.$$TXT=Peter
PRT. I am the Main Robot.
'-----------------------------------------------------------
PRR.
  PRT.I am Parallel-Robot-Nr. #pri# from #prc#.
  PRT.I am Son #pri# of $$TXT.
  :Loop
  WFV.$$INP|10
  PRT. Variable set to: $$INP
  GSB.wait
  'JIV.$$INP=1|out
  JMP.Loop
  :out
  END.
PRE.$$RET|0
GRS.$$RET|$$RES

'-----------------------------------------------------------
PRT.KOM-H: $$RET
PRT.I have #prc# running parallel robots and PlugIns.
PAU.1
FOR.$$MYL|1|300
  VAN.$$INP=1.342587459
  PRV.$$RET|$$INP
  VAC.$$INP=12,31,(2*45),(5+67)
  PRV.$$RET|$$INP
  VAP.$$INP=142,32
  PRV.$$RET|$$INP
  PAU.0.23
NEX.
' This will end the parallel robot
VAN.$$INP=1
PRV.$$RET|$$INP
PAU.2
DMP.
MBX.!
END.
'-----------------------------------------------------------
:wait
PAU.0.5
RET.
'-----------------------------------------------------------
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

It is possible to start more then one Parallel-running robot.
Each robot uses quite some system resources/memory therefore you should not run too many, depending on your CPU and memory capacity.
Each robot runs independently from the moment it starts.

Limitations:

There are limitations, running Parallel-running Script's on a "Private Desktop".
Each PD has its own address space.
Therefore an Robot can not easily communicate with another Robot on a PD.
Several communications features may not work on a PD.
However the ! Smarty's InterKOM Technology 2116 should work.

See also:

- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- PRS. - Parallel Robot Set Resukt 2647
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861
- WFM. - Wait-For-Message 2140
- TWM. - Text-Window-Message 2136

© 2023 Theo Gottwald * https://www.smart-package.com

3.69.6 WFV. - Wait-For-VariableChange

MiniRobotLanguage (MRL)

WFV. Command

Wait For Variablechange

Intention

This command can be used to wait for receipt of variables from another running robot.
It can also be used to synchronize running robots.

`WFV.` will use the name of the variable it has as got, as a reference. And then wait until anything is sent to this variable. It does not just wait for any variable, and it does not matter, if the variable value changes or not.

Think a moment, what can happen. If your script runs in a loop, with a `WFV.` and the message for that particular variable comes while your script is outside `WFV.` then your `WFV.` may wait forever for that `PRV.`-message.

Therefore use `WFV.` only if you really need it, and be sure that your parallel robot is inside the `WFV.` when the `PRV.` is done.
Using `WFV.` without such care can lead to a hanging script. If you are not sure, use the timeout-option **P2**. This will prevent a script from hanging.

Important:
`WFV.` **is not needed for a robot to receive variables from another robot.** Any robot receives variables that have been sent to him anyway, also without `WFV.`. This is done between the processing of the commands (code-lines).

More Infos:
You can send variables to another running robot using `PRV.`
You can wait for a variable from another running robot using `WFV.`
You can set a global Signal that multiple running Robots will see, using `SGS.` and check if such a signal exist, using `IGS.`

Syntax

WFV.P1[|P2]

Parameter Explanation

P1 **-** `(in) - Name of a Variable which shall be received from the other robot.`

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 **-** (in) - Timeout time in seconds. If omitted or 0, endless time is used.

Please note that P1 is taken "as is" and not resolved.

Example

```
'*********************************************
' PRR./PRE./PRV. - Sample 1
' Shows that the Parallel-Robot starts with
' all variables as they have been at starting-time
' of the starting robot.
' It can also use all Subroutines in this file.
'*********************************************
VAR.$$INP=0
VAR.$$TXT=Peter
PRT. I am the Main Robot.
'-----------------------------------------------------------
PRR.
  PRT.I am Parallel-Robot-Nr. #pri# from #prc#.
  PRT.I am Son #pri# of $$TXT.
  :Loop
  WFV.$$INP|10
  PRT. Variable set to: $$INP
  GSB.wait
  'JIV.$$INP=1|out
  JMP.Loop
  :out
  END.
PRE.$$RET|0
GRS.$$RET|$$RES

'-----------------------------------------------------------
PRT.KOM-H: $$RET
PRT.I have #prc# running parallel robots and PlugIns.
PAU.1
FOR.$$MYL|1|300
  VAN.$$INP=1.342587459
  PRV.$$RET|$$INP
  VAC.$$INP=12,31,(2*45),(5+67)
  PRV.$$RET|$$INP
  VAP.$$INP=142,32
  PRV.$$RET|$$INP
  PAU.0.23
NEX.
' This will end the parallel robot
VAN.$$INP=1
PRV.$$RET|$$INP
PAU.2
DMP.
MBX.!
END.
'-----------------------------------------------------------
:wait
PAU.0.5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'------------------------------------------------------------
```

Remarks

-

Limitations:

It is possible to start more then one Parallel-running robot.
Each robot uses quite some system resources/memory therefore you should not run too many, depending on your CPU number and memory capacity.
Each robot runs independently from the moment he starts with his script.

There are limitations, running Parallel-running Script's on a "Private Desktop".
Each PD has its own address space.
Therefore an Robot can not easily communicate with another Robot on a PD.
Several communications features may not work on a PD.
However the ! Smarty's InterKOM Technology 2116 should work.

See also:

- **PRV. - Parallel-Robot send Variable** 2651
- PRR. - Parallel-Robot-Run 2642
- PRE. - Parallelrobot-Run-End 2638
- WFV. - Wait-For-VariableChange 2655
- SGS - Set Global Signal (Mutex) 861
- WFM. - Wait-For-Message 2140
- TWM. - Text-Window-Message 2136

© 2023 Theo Gottwald * https://www.smart-package.com

3.70 PDC. - Private Desktop Commands

3.70.1 !PDC. - Private Desktop Command

MiniRobotLanguage (MRL)

PDC. - Manage private Desktops

Used to start the Automation on a private Desktop (Hidden Desktop)

Intention

Generally the **PDC.** - command is used to manage the "Private Desktops" from Script.
You can add or remove "Private Desktops". You can switch to - or from a "Private Desktop".

Using PDC. The SPR can do some things with PD's, these are:
1. Create PD's
2. Destroy PD's
3. Switch between PD's
4. Run Programs or Scripts on any PD, also with "Run as User"
5. Check if a PD exists or not

The SPR can NOT work itself on PD's. It will always stay with the desktop where it was run. This is due to Limitiations in windows.
To do this you will need to start a dedicated SPR on this PD using:

```
PDC.Run on Desktop WFI|PD Name|compiled Script.exe
```

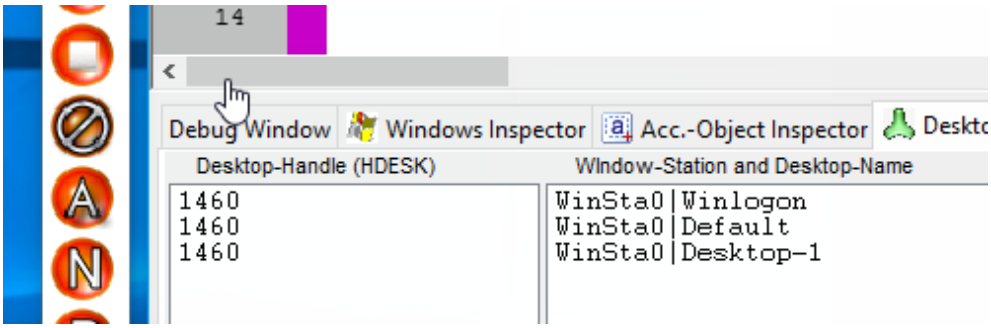
Aside of the PDC. - Command, you have the new **'#HID:** 283 Compiler Directive to start compiled executables on the private Desktop. Also the Command PRR. - Parallel-Robot-Run 2642 has been enhanced to start Scripts on PD's.

Even if you Switch to a PD, all Robot-Commands will be executed on the Desktop where the Robot was originally started.

Do i have "Private Desktops"?
If you have created private Desktops, these will be displayed in the SPR-Editor, in the "Desktops-Inspector". In this Example, you can see a "private Desktop" named "Desktop-1".

The Desktop with the name "Default" is always the "Normal User Desktop".
Its the Desktop which Windows will give you when it starts.
Its the Desktop where you work normally and where everything works as expected.

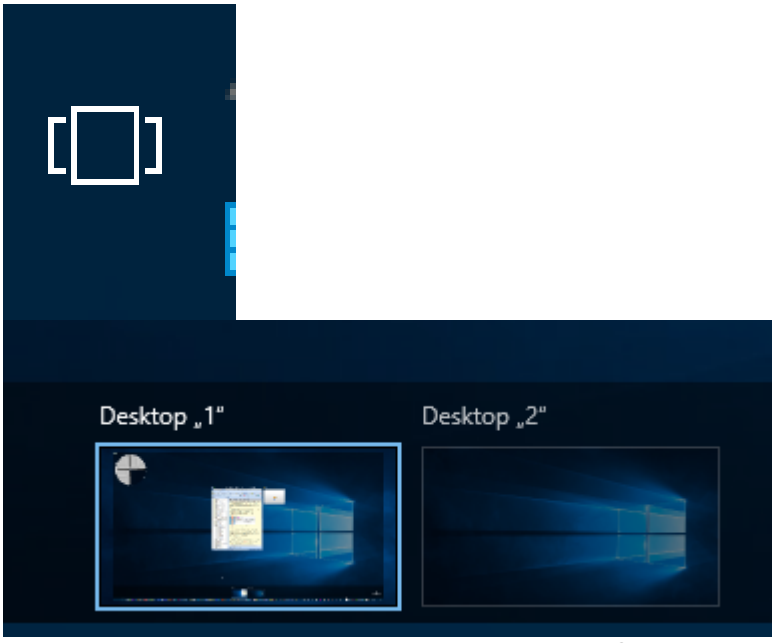
*© 2023 Theo Gottwald * https://www.smart-package.com*



The SPR-Editor shows a new private Desktop with the name "Desktop-1".

What are "private Desktops" good for?
You can run Scripts on "private Desktops". These Scripts can use the Mouse-Pointer.
And while these Scripts run, you - or an other User - can still work and use the Computer.

What is the difference to "Windows virtual Desktops" ?
Theoretically you can also use "private Desktops" like the new "virtual Desktops" that are built into Windows 10. But for this we recommend to prefer the Windows 10 Desktops, as you can move Windows
from one "virtual Desktop" to another. This is not possible with "Real Desktops". Therefore for human use the "virtual Desktops" may be best.



Virtual Desktops are built into Windows 10. These are NOT the Desktops we use for the SPR.

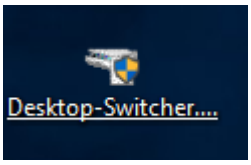
On the other side, these "virtual Desktops" do not support running independent Scripts.
You can not run a Script that uses the Mouse on such a "virtual Desktop".
It will not work due to the nature of these "virtual Desktops".

Do NOT intermix our "Real Private Desktops" with these "virtual Desktops".

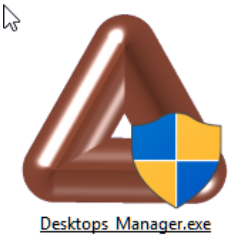
*© 2023 Theo Gottwald * https://www.smart-package.com*

These are more like "Desktop simulations". As said, the reason is that you can move windows
from one Desktop to another. This is not possible with "Private Desktops".
They are technically separated from each other.
Private Desktops have an own Mousepointer and own "Input Focus".

How can i manage "private Desktops" aside from Scripts?
To take a look on an private Desktop - for example to check for error-Messages - as well, as to Switch back from an "private Desktop" to the "Default Desktop",
you may need to use the "Switch Desktop"-Tool that is provided together with this SPR-Release.



You will find the Tool "Desktop-Switcher.exe" in the Startmenu.



Also you will find the Tool "Desktops Manager.exe" in the Startmenu.

Both Tools provide you with the ability to manage "private Desktops".
For example you can
- add a private Desktop,
- or remove a private Desktop.
You can also start an Executable "on a private Desktop".

Here is an Sample-Script that will switch to a private Desktop and then after 5 Seconds switch back to the "Default Desktop".

```
PDC.Create Desktop with Explorer|Desktop-1
PDC.Switch to Desktop|Desktop-1
PAU.5
PDC.Goto Default
ENR.
```

Important: You may not always be able to delete a private Desktop after Usage.
The "Private Desktop" will be gone after the next reboot.

Windows itself provides two Standard-Desktops, these are:

```
Default
  Winlogon
```

neither of these can be closed.

© 2023 Theo Gottwald * https://www.smart-package.com

Additional to the PDC-Commands there are two Query-Commands:
ICD./NCD. - If/Not Current Desktop 2666
IPD./NPD. - If/Not Private Desktop 2669

Syntax:

PDC.P1[|P2][|P3]

Parameter Explanation:

P1 - Subcommand, this can be any of the following:

```
"create desktop with explorer","cde"
"create desktop no explorer","cdn"
"switch to desktop","swd"
"set user","su"
"reset user","ru"
"run on desktop wfi","ruda"
"run on desktop wfo","rudb"
"close desktop","cld"
"get current desktop name","gdn"
"goto default","gd"
```

The Abbreviations like "gd" for "Goto Default" are functionally equal to the original command.

P2 - ...

Example:

```
'----------------------------------------
' This script will create a PD
' and destroy it later
'----------------------------------------
'
VAR.$$PRO=?progs\Smart Package Robot\Desktops_Manager.exe
PDC.Create Desktop with Explorer|DTA
PDC.Run on Desktop WFI|DTA|$$PRO
PDC.Switch to Desktop|DTA
' The following Bubble-Text appears on the original Desktop
SBT.Waiting a bit ...|FS:24 CS:1 T:15
PAU.5
PDC.Goto Default
PAU.10
' Close PD
PDC.Close Desktop|DTA
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'----------------------------------------
' Create Desktop / Switch to Desktop
'----------------------------------------
 PDC.Create Desktop with Explorer|Desktop-1
 PDC.Switch to Desktop|Desktop-1
 PAU.5
 PDC.Goto Default
 ENR.



'----------------------------------------
' Compile-Time Commands
'----------------------------------------
'
' The following line will make the compiled
' executable start up on the private Desktop "Desktop-2".
' It will only work if the Script has full admin-rights.
' The command will switch to the private Desktop.
' After you click the Messagebox, the private Desktop will be
removed.
'
'#HID:2|scn
'
MBX.I am the Smart Package Robot on Desk 2!
END.
```

Remarks:

'#HID: does work with '#RAU:.

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops.
To start SPR Scripts on a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.
This feature needs Admin-Rights!
This feature is currently a bit experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops.
For Limitations in the Windows-Explorer itself, not all Actions will be possible on "private Desktops".
Sometimes System-Windows - like the Start-Menu - will not open on the "private Desktop" but on the Default Desktop.

DANGER: Using the "Close Desktop"-Option will forcefully terminate all processes that are bound to the Desktop that is to be closed!

Also note: Before the new created Desktop can be closed, the Script switches to the new Desktop.
When the new Desktop has been closed, the Script switches to the Default desktop.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.2 HDC. - Hidden Desktop Control

MiniRobotLanguage (MRL)

HDC. - Hidden Desktop Control

Enable or Disable Hidden Desktop Access

Intention

The HDC.-Command can be used to manually enable or disable the possibility to switch to the "Hidden Desktop".
The "Hidden Desktop" is the System-Desktop. Its not a "private Desktop".

```
' This will enable the Access to the Hidden Desktop
HDC.1

' This will disbale the possibility to access the Hidden
Desktop
' Robot Executables can still work on the Hidden Desktop.
HDC.0
```

Syntax

HDC.[P1]

Parameter Explanation

P1 - (optional) can be one of the following options:

e - "e"nable Hidden Desktop Access. You can also write "1" instead.

d - "d"isable Hidden Desktop Access. You can also write "0" instead.
Use this Option in your Client-Script to prevent the User from being able
to enter the "Hidden Desktop".

r - "r"estart. This is the same like "e" but stronger but can be slow!
Internally the "UI0Detect" service is restarted. You can also write "2" instead.

f - "f"orbid. This will totaly disable the option to use the Hidden Desktop.
We recommend to not use this option as it can also influence Robot-Scripts.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
HDC.disable
HDC.d
HDC.0

or

HDC.e
HDC.1
HDC.enable
```

Remarks

Internally it will start or stop the "`UI0Detect`" Service.

Limitations:

See also:

- EXV. - Execute via System-Desktop 912
- 1.0.1 '#EXV: - Execute on Hidden Desktop 280

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.3 ICD./NCD. - If/Not Current Desktop

MiniRobotLanguage (MRL)

ICD./NCD. - If/Not Current Desktop

Used to verify if the current Script runs on the specified Desktop or not

Intention

This Command will find out, if the running Script runs on the specified "private Desktop" (PD) or not.
For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

Usage is easy. You can just check whether your runs on that PD or not.

```
ICD.DTB
  MBX.Running on DTB.
ELS.
  MBX.Not running on DTB.
EIF.
```

Its possible to specify more then one Desktop-name (no limit).
In this case "OR" logic is applied. If the Script runs on "any of" these Desktops, the condition evaluates to true for ICD. and to false for NCD.

```
' Test if the Script runs whether on Desktop "DTA" or "DTB"
or on "Default"
ICD.DTA,DTB,Default
  MBX.Running on DTB.
ELS.
  MBX.Not running on DTB.
EIF.
```

Windows itself provides two Standard-Desktops, these are:

```
Default
Winlogon
```

Important:
Even if you Switch to a PD, all Robot-Commands will still be executed on the Desktop where the Robot was originally started. See below for an example on how to start a Script on a PD.

Syntax:

```
ICD.[P1][,P2][,Px]
```


© 2023 Theo Gottwald * https://www.smart-package.com

NCD.[P1][,P2][,Px]

Parameter Explanation:

P1 - The name of the private Desktop that we want to check for..

Example:

```
'----------------------------------------
' This script will create a PD
' and demonstrate the ICD.-Query on and not on
' the new desktop.
'
GSB.Test
PDC.Create Desktop no Explorer|DTB
PRR.0|DTB
  GSB.Test
  ENR.
PRE.
MBX.!
ENR.
'----------------------------------------
:Test
ICD.DTB
  MBX.Running on DTB.
ELS.
  MBX.Not running on DTB.
EIF.
RET.
ENR.
```

Remarks:

-

Limitations:

This Feature may require Windows Vista or higher.

See also:

- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658

© 2023 Theo Gottwald * https://www.smart-package.com

- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290
- PRR. - Parallel-Robot-Run 2642

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.4 IPD./NPD. - If/Not Private Desktop

MiniRobotLanguage (MRL)

IPD./NPD. - If/Not Private Desktop

Used to verify if a private Desktop exist or not

Intention

This Command will find out, if a "private Desktop" (PD) exists or not.
For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

To check if the PD "MyPD" exists, you can simply type:

```
$$DSK=MyPD
IPD.$$DSK
  MBX.$$DSK exists.
ELS.
  MBX.$$DSK not exists.
EIF.
ENR.
```

Syntax:

```
IPD.[P1]
NPD.[P1]
```

Parameter Explanation:

P1 - (in) The name of the private Desktop. If omitted, the name "Default" is taken.

Example:

```
'============================================================
' This Script will verify if a specified Private Desktop
' exists or not
'
VAR.$$DSK=DTX
GSB.Test
PDC.Create Desktop no Explorer|$$DSK
GSB.Test
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
ENR.
'----------------------------------------
:Test
IPD.$$DSK
  MBX.$$DSK exists.
ELS.
  MBX.$$DSK not exists.
EIF.
RET.
ENR.
```

Remarks:

-.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.5 PDC.Close Desktop

MiniRobotLanguage (MRL)

PDC.Close Desktop

Used to destroy a private Desktop and all programs running on that PD (Hidden Desktop)

Intention

Generally we do not recommend to use this command, unless you have reasons to remove the PD.
In any normal use, you can just leave the PD, it will be removed at the next reboot anyway.

Desktops can not be closed if there are programs running.
Therefore the command will try to close all programs on that Desktop.
Using this command will close all Programs running on a "Private Desktop" without further notice.
Even then, it may not be possible to close a "Private Desktop".

Using this Command may close a"private Desktop" (PD) if possible.
If not, a "empty Desktop" without anything will remain until the next reboot of the PC.

For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

To create and destroy a PD you can simply type:

```
PDC.Create Desktop with Explorer|Desktop-1
MBX.Private Desktop created
PDC.close desktop|Desktop-1
MBX.Private Desktop destroyed
ENR.
```

The changes will be visible in the SPR-Editor's "Desktop Inspector".

Important:

When trying to close a PD, the Skript will need to switch to this PD and then close all Processes that are connected to the PD!
This command may have therefore unexpected side-effects! Also the Screen may flicker.
Under special circumstances this comamnd may also close programs that run on another desktop.

Windows itself provides two Standard-Desktops, these are:

```
Default
Winlogon
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

neither of these can be closed.

Syntax:

PDC.Close Desktop|P1
PDC.CDL|P1

Parameter Explanation:

P1 - The name of the private Desktop that shall be closed.

Example:

```
'---------------------------------------
'
PDC.Create Desktop with Explorer|Desktop-1
MBX.Private Desktop created
PDC.close desktop|Desktop-1
MBX.Private Desktop destroyed
ENR.
```

Remarks:

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops.
To create or destroy a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- PRR. - Parallel-Robot-Run 2642
- PDC_General 2658
- EXV. - Execute via System 912

© 2023 Theo Gottwald * https://www.smart-package.com

- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.6 PDC.Create Desktop no Explorer

MiniRobotLanguage (MRL)

PDC.Create Desktop no Explorer

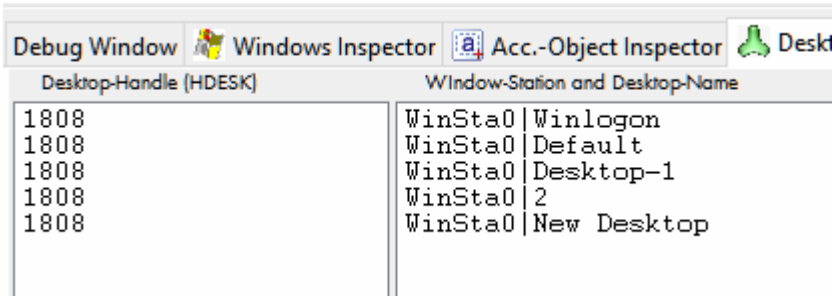
Used to create a private Desktop (Hidden Desktop)

Intention

This Command will create e a new "private Desktop" (PD) without running the windows Explorer on this new desktop.
It only makes sense, to start a Program on that PD, immediately after its creation.
.
For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.
Here is a Sample-Script.

```
VAR.$$PRO=?progs\Smart Package Robot\Desktops_Manager.exe
PDC.Create Desktop no Explorer|DTA
PDC.Run on Desktop WFI|DTA|$$PRO
PDC.Switch to Desktop|DTA
PAU.5
PDC.Goto Default
PDC.Close Desktop|DTA
MBX.!
ENR.
```

The result will be visible in the SPR-Editor's "Desktop Inspector".



Important:

PD's that are created without an "Explorer" may automatically close after the Script (and otehr Tasks running there) have ended!
Therefore they will only be visible as long as the Script is running.

Syntax:

*© 2023 Theo Gottwald * https://www.smart-package.com*

PDC.Create Desktop no Explorer|P1
PDC.CDN|P1

Parameter Explanation:

P1 - The name of the new private Desktop.

Example:

```
'---------------------------------------
'
VAR.$$PRO=?progs\Smart Package Robot\Desktops_Manager.exe
PDC.Create Desktop no Explorer|DTA
PDC.Run on Desktop WFI|DTA|$$PRO
PDC.Switch to Desktop|DTA
PAU.5
PDC.Goto Default
PDC.Close Desktop|DTA
MBX.!
END.
```

Remarks:

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664

© 2023 Theo Gottwald * https://www.smart-package.com

- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.7 PDC.Create Desktop with Explorer

MiniRobotLanguage (MRL)

PDC.Create Desktop with Explorer

Used to create a private Desktop (Hidden Desktop)

Intention

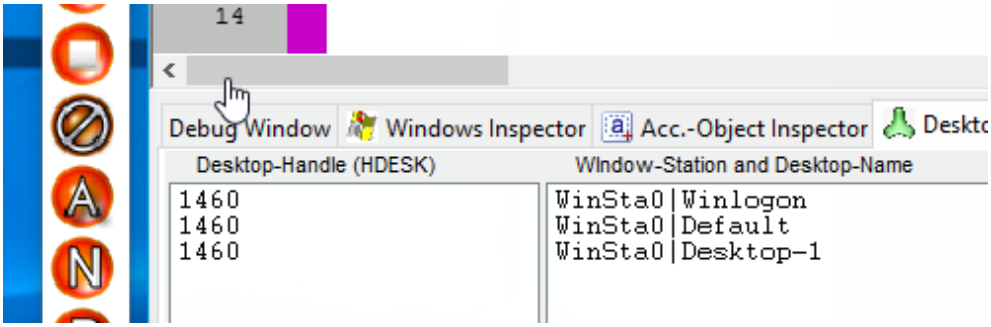
This Command will create e a new "private Desktop" (PD).
The creation of the Private Desktop is rather fast, while the loading of the Explorer will take some time.
This can continue even after the command was ready.
Even though an Explorer was loaded on the "Private Desktop", it may not work in all parts as expected.
That's due to the fact that the Explorer itself is not prepared to work with PD's.

For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

To just create a PD you can simply type:

```
PDC.Create Desktop with Explorer|Desktop-1
```

The result will be visible in the SPR-Editor's "Desktop Inspector".



Syntax:

PDC.Create Desktop with Explorer|P1

© 2023 Theo Gottwald * https://www.smart-package.com

PDC . CDE | P1

Parameter Explanation:

P1 - The name of the new private Desktop.

Example:

```
'----------------------------------------
'
PDC.Create Desktop with Explorer|New Desktop
END.
```

Remarks:

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.8 PDC.Get current Desktop Name

PDC.Create Desktop no Explorer Previous 2677 Top 2658 Next 2682

MiniRobotLanguage (MRL)

PDC.Get current Desktop Name

Used to get the Name of the Desktop where a Script is running

Intention

This Command will find out the Name of the "private Desktop" (PD) where the Script is running.
For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

There are two way to use this command::

```
PDC.Get Current Desktop Name
MBX.$tos$
ENR.

' Or

PDC.Get Current Desktop Name|$$NAM
MBX.$$NAM
ENR.
```

Important:

Switching to a desktop using `PDC.Switch to Desktop|(PD-Name)`
will not change the value that is returned with `PDC.Get Current Desktop Name`

The reason is, that the Script itself ist always running at the desktop where it was started,
In fact the return of this command is "The desktop where the Script was started."

```
PDC.Create Desktop with Explorer|DTA
PDC.Switch to Desktop|DTA
PDC.Get Current Desktop Name
MBX.$tos$
ENR.
```

Switching the "View" does not make the Script to do things on another Desktop.

To do this you need to run a Script on that other Desktop.
You can do this for example using `'#HID:` or using the new enhanced `PRR./PRE.-` command.

Syntax:

© 2023 Theo Gottwald * https://www.smart-package.com

PDC.Get current Desktop Name[|P1]
PDC.GDN[|P1]

Parameter Explanation:

P1 - The name of the private Desktop where the Script is running. If omitted, the result is placed on TOS.

Example:

```
'---------------------------------------
'
PDC.Get Current Desktop Name|$$TXT
MBX.The name of the PD where the Script was started is: $$TXT
END.
```

Remarks:

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.9 PDC.Goto Default

MiniRobotLanguage (MRL)

PDC.Goto Default

Used to switch View from a private Desktop (Hidden Desktop) to the Default Desktop

Intention

This Command is similar to `PDC.Switch to Desktop`, just that the Destination of the Switch is fixed, and is the "Default Desktop". This is the Original Desktop that was created by the OS on Startup.

For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

```
PDC.Create Desktop with Explorer|DTA
PDC.Switch to Desktop|DTA
PAU.3
PDC.Goto Default
ENR.
```

Important:

You can switch to the "Default Desktop" using this command, even when the Script was run on another PD.
However all commands that the Script executes will always be executed on the desktop where the Script was originally started.
Switching the "View" does not make the Script to do things on another Desktop.
To do this you need to run a Script on that other Desktop.
You can do this for example using `'#HID:` or using the new enhanced `PRR./PRE.-` command.

Syntax:

```
PDC.Goto Default
PDC.GD
```

Parameter Explanation:

-

Example:

```
'----------------------------------------
PDC.Create Desktop with Explorer|DTA
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
PDC.Switch to Desktop|DTA
PAU.3
PDC.Goto Default
ENR.
```

Remarks:

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.10 PDC.Run on Desktop WFI/WFO

MiniRobotLanguage (MRL)

PDC.Run on Desktop WFI/WFO

Used to start a program on a private Desktop (Hidden Desktop)

Intention

This Commands will start a program on an existing "private Desktop" (PD).
The PD must exist, if the Desktop does not exist, the Program will not start the executable.
For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

There are two versions of this command:

```
PDC.Run on Desktop WFI|DTB|?path\Desktops_Manager.exe
```
and
```
PDC.Run on Desktop WFO|DTB|?path\Desktops_Manager.exe
```

In most cases there will be no big difference between either of these.
Technically the "WFI"-Version makes an "Wait for Input Idle" after the executable was started.
While the "WFO"-Version makes an "Wait for Single Object".
Depending on the Type of program you are starting any of these may be the perfect System,
to get the Moment, when the Program is really running.
In most cases you can choose the "WFO"-Version, and if it will block or not be like you want, choose the WFI-Version.

To run an executable on a PD you can simply type:

```
PDC.Create Desktop no Explorer|DTB
PDC.Run on Desktop WFO|DTB|?path\Desktops_Manager.exe
PDC.Switch to Desktop|DTB
PRR.1|DTB
  MBX.Click to proceed
  PDC.Goto Default
  ENR.
PRE.
ENR.
```

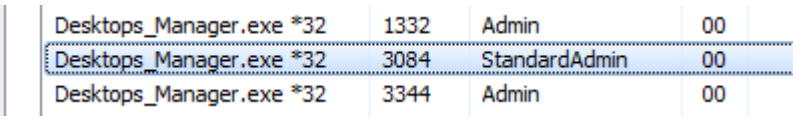
To run an executable on a PD with another User you can simply type:

```
'---------------------------------------
' We are starting a ParallelRobot on the "new Desktop"
' to make the Switchback more easy.
' Here we use the PDC.Setuser to start something with another
User
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
PDC.Create Desktop no Explorer|DTB
PDC.Set User|StandardAdmin|Pass2
PDC.Run on Desktop WFO|DTB|?path\Desktops_Manager.exe
PDC.Switch to Desktop|DTB
PRR.1|DTB
  MBX.Click to proceed
  PDC.Goto Default
  ENR.
PRE.
ENR.
```

In the Task-Manager you can see the success:



Desktops_Manager.exe *32	1332	Admin	00
Desktops_Manager.exe *32	3084	StandardAdmin	00
Desktops_Manager.exe *32	3344	Admin	00

This will also work on the "Default" Desktop.

```
PDC.Set User|StandardAdmin|Pass2
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
```

Syntax:

PDC.Run on Desktop WFI|P1|P2[|P3]
PDC.Ruda|P1|P2[|P3]

PDC.Run on Desktop WFO|P1|P2[|P3]
PDC.Rudb|P1|P2[|P3]

Parameter Explanation:

P1 - (in) The name of the new private Desktop.
P2 - (in) The name and path of the executable.
P3 - (in,optional) a number >0 that specified a waiting time in ms.
This waiting-time is applied after the executable was started.

Example:

```
'---------------------------------------
' We are starting a ParallelRobot on the "new Desktop"
' to make the Switchback more easy.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
PDC.Create Desktop no Explorer|DTB
PDC.Run on Desktop WFO|DTB|?path\Desktops_Manager.exe
PDC.Switch to Desktop|DTB
PRR.1|DTB
  MBX.Click to proceed
  PDC.Goto Default
  ENR.
PRE.
ENR.

'---------------------------------------
' We are starting a ParallelRobot on the "new Desktop"
' to make the Switchback more easy.
' Here we use the PDC.Setuser to start something with another
User
'
PDC.Create Desktop no Explorer|DTB
PDC.Set User|StandardAdmin|Pass2
PDC.Run on Desktop WFO|DTB|?path\Desktops_Manager.exe
PDC.Switch to Desktop|DTB
PRR.1|DTB
  MBX.Click to proceed
  PDC.Goto Default
  ENR.
PRE.
ENR.
```

Remarks:

The `PDC.SetUser`-Feature is supported.

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Set User / Reset User 2688
- PDC.Switch to Desktop 2691

© 2023 Theo Gottwald * https://www.smart-package.com

- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.11 PDC.Set User / Reset User

MiniRobotLanguage (MRL)

PDC.SetUser / Reset User

Used to choose another User for Running a Executable using the PDC.-Commands

Intention

This Command will only work together with
PDC.Run on Desktop WFI/WFO 2684

For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

Here is an example.

```
' This command will jsut close all running "Desktop-Managers"
' on this Desktop.
CAW.FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager

' Here we choose a User called "StandardAdmin" with a Password of "Pass2"
PDC.Set User|StandardAdmin|Pass2

' New we start 3 Instances of a executable that must be available
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe

' Here we reset the User to the User that started the Script
PDC.Reset User
' This waiting time is only needed to get things in the
' Task Manager in the right order.
PAU.1
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
ENR.
```

The result is visible in the Task-Manager.

Desktops_Manager.exe...	1056	StandardAdmin	00
Desktops_Manager.exe...	1364	StandardAdmin	00
Desktops_Manager.exe...	3352	StandardAdmin	00
Desktops_Manager.exe...	3820	Admin	00

Syntax:

PDC.SetUser|P1|P2[|P3]
PDC.SU|P1|P2[|P3]

*© 2023 Theo Gottwald * https://www.smart-package.com*

PDC.ResetUser
PDC.RU

Parameter Explanation:

P1 - (in) Username
P2 - (in) Password
P3 - (in,optional) Domain - if omitted the PC-name is used internally

Example:

```
'----------------------------------------
'
$$LOG=(your Username)
$$PAS=(your Password)
$$DOM=(selected Domain)
CAW.FORM_DESKTOPS_MANAGER_FRMMAIN_CLASS:0|Desktops-Manager
(c) 2016 Theo Gottwald
PDC.Set User|$$LOG|$$PAS|$$DOM
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
PDC.Reset User
PAU.1
PDC.Run on Desktop WFO|Default|?path\Desktops_Manager.exe
ENR.
VAR.$$DSK=DTX
GSB.Test
PDC.Create Desktop no Explorer|$$DSK
GSB.Test
ENR.
```

Remarks:

Some tasks may not work without the selected User having Admin-Rights.

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666

© 2023 Theo Gottwald * https://www.smart-package.com

- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Switch to Desktop 2691
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.70.12 PDC.Switch to Desktop

MiniRobotLanguage (MRL)

PDC.Switch to Desktop

Used to switch the View to a private Desktop (Hidden Desktop)

Intention

This Command will switch the View to a "private Desktop" (PD).
This way you can see whatever happens on that PD.

IMPORTANT:
Even if you Switch the View to a PD, all Robot-Commands will be executed on the Desktop where the Robot was originally started. This command does only switch the View.

For more Details on "Private Desktops" (PD) see here: PDC - General Information 2658
.

To just create a PD and Switch there you can simply type:

```
PDC.Create Desktop no Explorer|Desktop-1
PDC.Switch to Dektop|Desktop-1
PAU.2
PDC.Goto Default
ENR.
```

Here is another example that will switch to and back from a new created Desktop.

```
PDC.Create Desktop with Explorer|DTF
PDC.Switch to Desktop|DTF
PRR.0|DTF
  MBX.I am on the DTF-Desktop
  PDC.Goto Default
PRE.
MBX.End the Script
ENR.
```

Syntax:

```
PDC.Switch Desktop|P1
PDC.swd|P1
```

Parameter Explanation:

P1 - The name of the private Desktop.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example:

```
'----------------------------------------
' This Script will creat e Desktop and start a
' new Script on that Desktop using the PRR.-Command
'
PDC.Create Desktop with Explorer|DTF
PDC.Switch to Desktop|DTF
PRR.0|DTF
  MBX.I am on the DTF-Desktop
  PDC.Goto Default
PRE.
MBX.End the Script
ENR.
```

Remarks:

This feature is currently experimental, therefore not all features of the SPR and the SPR-Editor may work as expected on private Desktops. To create a private Desktop, the Executable needs Admin-Rights.

Limitations:

This Feature may require Windows Vista or higher.

See also:

- PRR. - Parallel-Robot-Run 2642
- 2.1.5 '#HID: - Hidden Desktop Scripts 283
- PDC-General 2658
- ICD./NCD. - If/Not Current Desktop 2666
- IPD./NPD. - If/Not Private Desktop 2669
- PDC.Close Desktop 2671
- PDC.Create Desktop no Explorer 2674
- PDC.Create Desktop with Explorer 2677
- PDC.Get current Desktop Name 2679
- PDC.Goto Default 2682
- PDC.Run on Desktop WFI/WFO 2684
- PDC.Set User / Reset User 2688
- EXV. - Execute via System 912
- 2.1.4 '#EXV: - Execute on System Desktop 280
- HDC. - Hidden Desktop Control 2664
- '#RAU: Run as User 271
- #ONCE / # OEND - Multiple Include Protection 251
- 2.1.8 '#INC: - Include Files and Folders into the Executable 290

© 2023 Theo Gottwald * https://www.smart-package.com

3.71 PlugIn-Commands

3.71.1 ! Smarty's PlugIn's

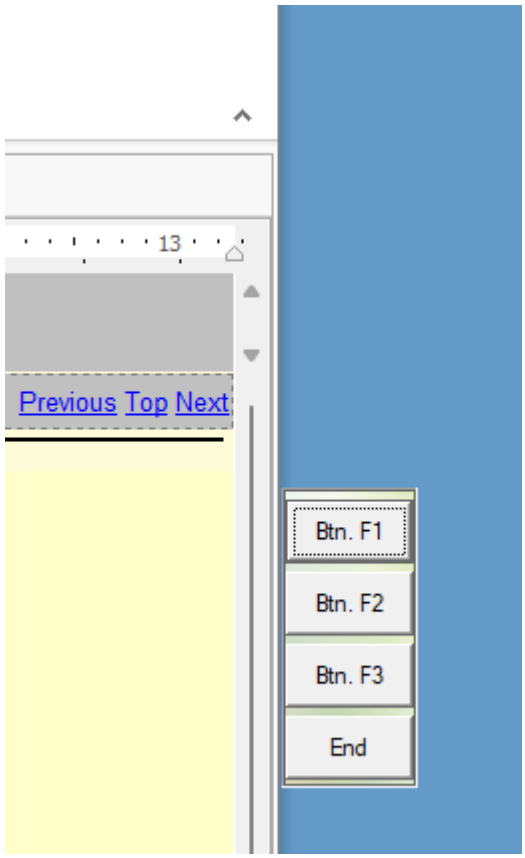
Smarty's PlugIn Commands **Previous** 1856 **Top** 41 **Next** 2702

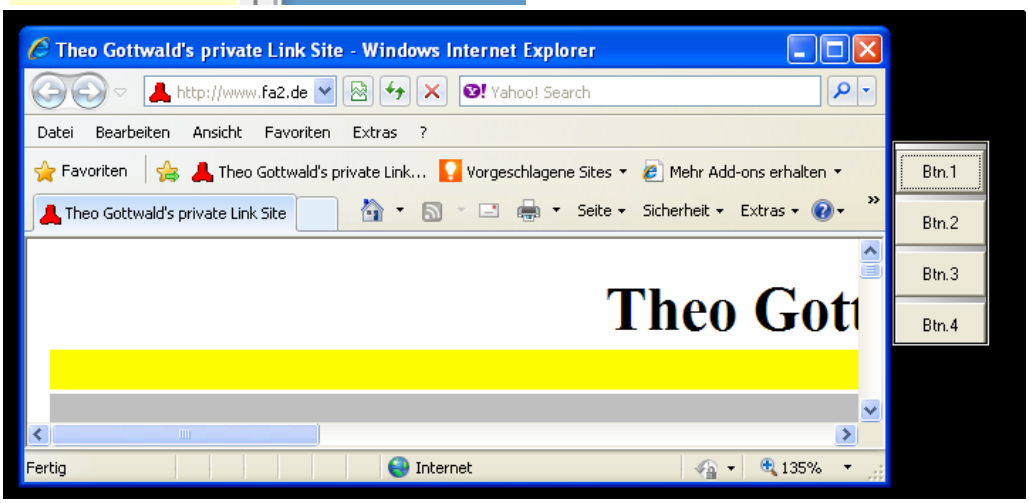
Smarty's PlugIn Commands

The "PlugIn01.exe" is an configurable Buttonbar.
It can be attached to any Window.
If you press the Buttons you can execute Smart Package Robot-Scripts directly in any Software.

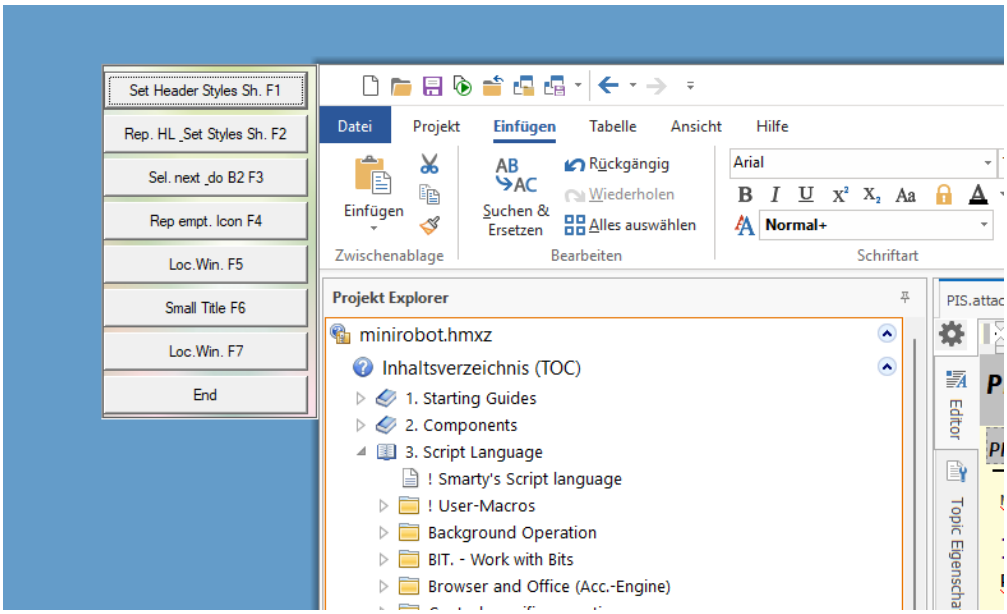
Examples:

*© 2023 Theo Gottwald * https://www.smart-package.com*

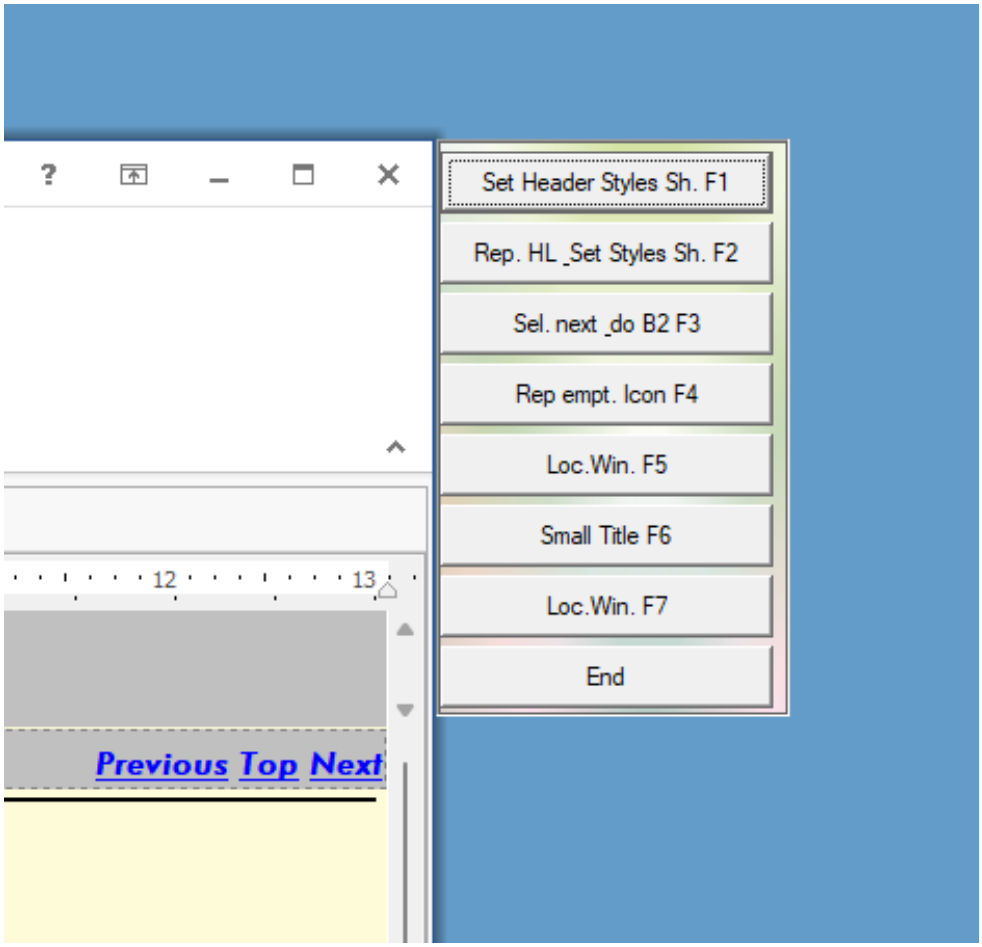




© 2023 Theo Gottwald * https://www.smart-package.com



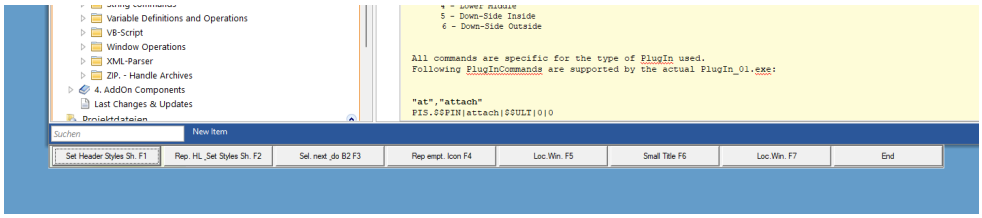

```
PIS.$$PLG|sv
PIS.$$PLG|attach|$$HWN|1|1
```




```
PIS.$$PLG|sv
PIS.$$PLG|attach|$$HWN|0|1
```

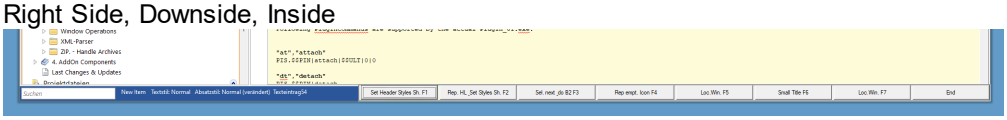
Left Side, Downside, Outside

© 2023 Theo Gottwald * https://www.smart-package.com



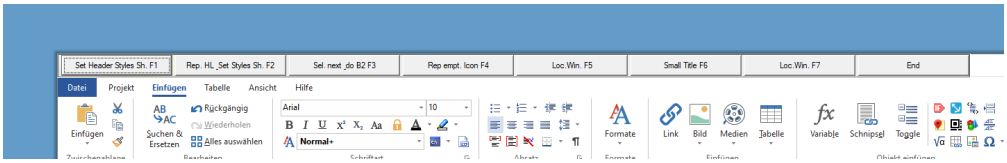

```
PIS.$$PLG|sh
PIS.$$PLG|attach|$$HWN|5|6
```

Right Side, Downside, Inside



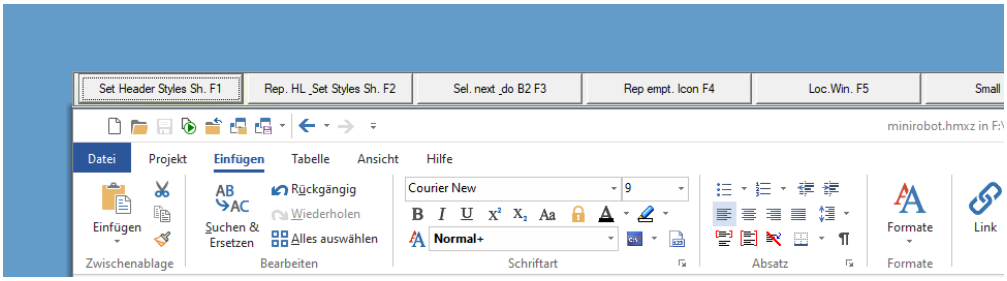
```
PIS.$$PLG|sh
PIS.$$PLG|attach|$$HWN|6|5
```

Top-Side Inside




```
PIS.$$PLG|sh
PIS.$$PLG|attach|$$HWN|5|1
```

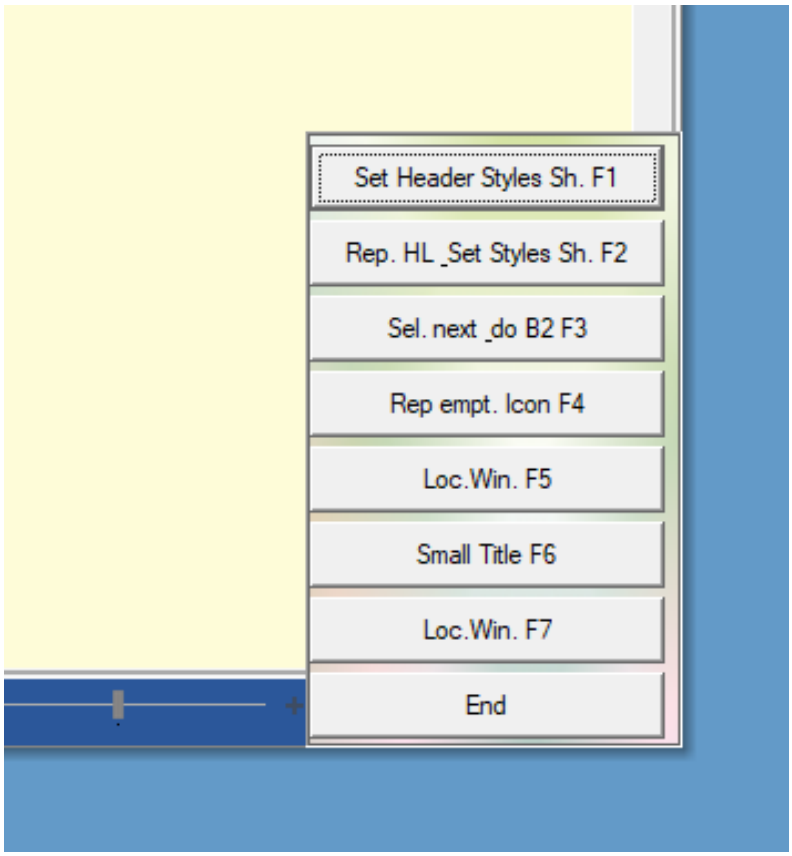
Top-Side Outside




```
PIS.$$PLG|sh
PIS.$$PLG|attach|$$HWN|5|7
```

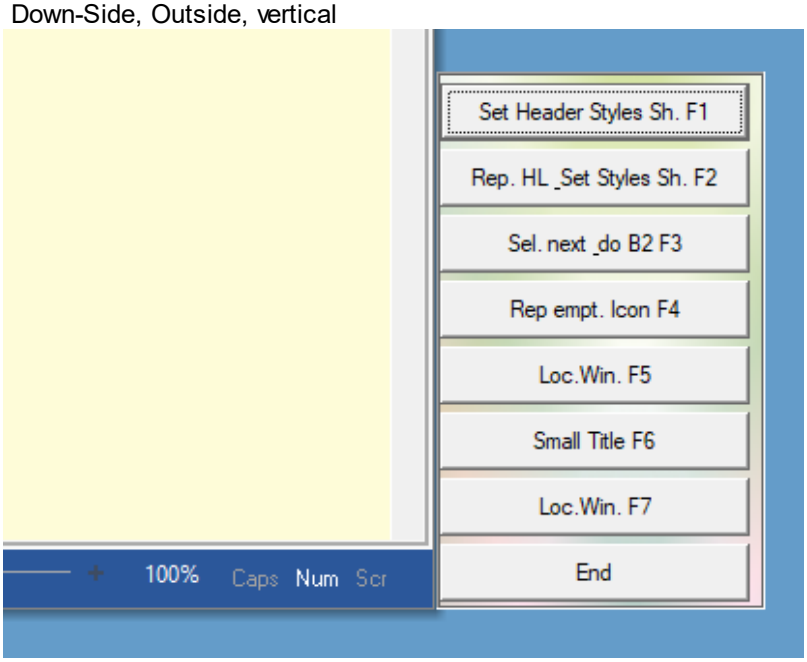
Down-Side, Inside, vertical

© 2023 Theo Gottwald * https://www.smart-package.com




```
PIS.$$PLG|sv
PIS.$$PLG|attach|$$HWN|6|5
```

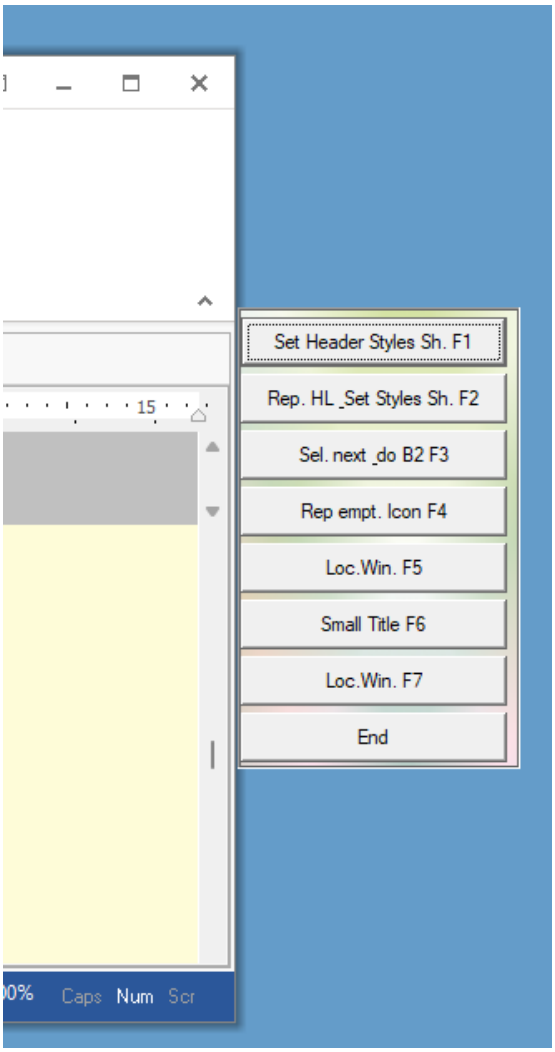
Down-Side, Outside, vertical




```
PIS.$$PLG|sv
PIS.$$PLG|attach|$$HWN|2|5
```

Right-Side, Middle, Outside, vertical

© 2023 Theo Gottwald * https://www.smart-package.com




```
PIS.$$PLG|sv
PIS.$$PLG|attach|$$HWN|2|3
```

Code-Example Button-Bar for "Help and Manual 8"

```
'Creation date: 02-08-2022 at 07:53:12
'============================================================
'#EXE:?path\
'#SPI:ForceWrite

' This will run the NoLimits2 in case its not yet running
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=4
PIS.$$REA|sbn|$$BUN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' use any window to attach the plugin
PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|0|0

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.

$$FIL=XY.hmxz
GEP.$$FIL|$$PAT
MBX.$$PAT
EXE.$$PAT
SAV.Restore
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'------------------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.4
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'------------------------------------------------------------
:doButton01
DBP.B1 pressed!
RET.
'------------------------------------------------------------
:doButton02
DBP. B2 pressed!
RET.
'------------------------------------------------------------
:doButton03
DBP. B3 pressed!
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
ENR.
'=======================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'-------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.71.2 PIL. - PlugIn Launch

MiniRobotLanguage (MRL)

PIL. Command

PlugInLaunch

Intention

The `PIL`-Command launches a PlugIn. A PlugIn is (for example) a button-bar like the one in this picture:

A PlugIn is a separate Program which communicates with the robot.

There may be multiple PlugIn's running at the same time

At the moment of this writing the only Plugin avaiable is PlugIn_01.exe - thats a sort of button-bar.

Syntax

PIL.P1|P2

Parameter Explanation

P1 - `variable which becomes the PlugIn Index-No`
P2 - `PlugIn typ-number, actually always 01`

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'------------------------------------------------------------
'SPR Script-file: NoLimits 2 AddOn
'Purpose:
'Creation date: 12-17-2015,22:47:50
'------------------------------------------------------------

' This will run the NoLimits2 in case its not yet running
STW.ctwV|NL3D_MAIN_|NoLimits 2|1|runnl2
' Possible Commands are: SWP. | SWS.
SWP.5,2|1527,1112|T
HTV.$$ULT

PIL.$$REA|01
' use any window to attach the plugin

PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show

' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|0|0

' Set Button Text
PIS.$$REA|sBt|1|Btn. F1
PIS.$$REA|sBt|2|Btn. F2
PIS.$$REA|sBt|3|Btn. F3
PIS.$$REA|sBt|4|Btn. End

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|sbn|4

' Farben für Mouseover
$$COM=&HFF00FF
$$COL=&H00FFFF

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.

PAU.2
:enz
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
' Starte NL2
:runnl2
CHD.?cstartprog\NoLimits$sp$2\
EXO.?cstartprog\NoLimits$sp$2\NoLimits$sp$2.lnk
STW.ct|NL3D_MAIN_{7F825CE1-21E4-4C1B-B657-DE6FCD9AEB12}|
NoLimits 2
RET.
'-------------------------------------------------------
:react

' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA

' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB

PRT. Button pressed: §§TMA
PRT. Action-code: §§TMB

GSB.TestMo

' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.4
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAE.
' PRT.Undefined Button-Number: §§TMA
    : $$RET=
  ESC.
EIF.
RET.
'-------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------
:doButton01
PRT.B1 pressed
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------
:doButton02
DBP. B2 pressed
RET.
'-------------------------------------------------------
:doButton03
DBP. B3 pressed
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'-------------------------------------------------------
:TestMo
' react on Mouseover
IVS.$$TMB=!M!O!
  PIS.$$PIN|sbbc|$$TMA|$$COM
  GTO.MOut
EIF.

' react on Mouse-Leaves Button Area
IVS.$$TMB=!M!L!
  PIS.$$PIN|sbbc|$$TMA|$$COL
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------
```

Remarks

You can use up to 5 Plugins. Just start them, each will get another Index.

Limitations:

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.3 PIW. - PlugInWait

MiniRobotLanguage (MRL)

PIW. Command

Plug In Wait

Intention

The `PIW.`-Command waits for a Message from a specified PlugIn.
Thats a separate Program which communicates with the robot.
There may be different PlugIns.

At the moment of this writing the only Plugin avaiable is PlugIn_01.exe - thats a sort of button-bar.

The `PIW.` waits till a button is pressed or a mouseover event happened then gives the result back in **P2**. **P1** contains the PlugIn Index from `PIL`

Syntax

```
PIW.P1|P2[|P3]
```

Parameter Explanation

P1 **-** variable with the the PlugIn Index-No. from **PIL**.

P2 **-** Variable which holds the number of the event that happened
to the PlugIn.

P3 **-** ***(optional)*** Timeout-Value. After this time in seconds PIW. is left (without any Event) and the Timeout-Flag is set.

PIW. returns a number of Messages in **P2** from the PlugIn:

```
' 12345 represents the window to which the Plugin has been
attached using PIS.$$PIN|attach

' Close is returned when the window which the Plugin is
attached to has been closed/does not respond.
 12345 !Closed!
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' Iconic is returned, when the window which the Plugin is
attached to has been iconized.
 12345 !Iconic!

' When the window becomes normal sized from being iconized,
the Plugin reports
 12345 !nsize!

' invisible is returned the window which the Plugin is
attached to has been made invisible using MWI*.
 12345 !invisible!

' When the window becomes visible from being invisible, the
Plugin reports
 12345 !visible!

' disabled is returned the window which the Plugin is attached
to has been made disabled using MWE*.
 12345 !disabled!

' When the window becomes visible from being invisible, the
Plugin reports
 12345 !enabled!

' normal is returned, if the window which has been iconized or
made invisible or was disabled becomes "usable" again. Usable
is a window which is enabled, visible and not iconized.
 12345 !normal!

' "D"ouble-"P"ressed is been returned on a Doubleclick on that
Button
' The number 5 represents in this case the number for the
button
' which has been pressed.
 5 !P!D!

' "P"ressed is been returned on a Click on that Button with
left Mousebutton
' The number 5 represents in this case the number for the
button
' which has been pressed.
 5 !P!

' "R"ight is been returned on a Click on that Button with
Right Mousebutton
' The number 5 represents in this case the number for the
button
' which has been pressed.
 5 !R!

' Mouse Over is been returned if the Mouse is over the
specified
' button. In fact the Message is returned once the Mouse
enters the
' button area.
' The number 5 represents in this case the number for the
button
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' which area has been entered.
 5 !M!O!

' Mouse Left is been returned if the Mouse has left the
specified
' button area with a mouse move. In fact the Message is
returned
' once the Mouse leaves the button area.
' The number 5 represents in this case the number for the
button
' which area has been left.
 5 !M!L!

' The OK Message is been sent in Response to commands given to
' the PlugIn as a Reply.
' OK
```

Additionally there custom responses from the PlugIn depending on the command being sent.
With **PIS.** for example **"gethandle"**.

---
* not implemented at the time of this writing.

Example

```
' The following code is symbolic, can not run like this,
'
DOL.1
' Wait for PlugIn Event, Returncode in $$RET
 PIW.$$PIN|$$RET

JSR.react
OOP.

'-------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.$$RET|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.$$RET|$sp$|2|§§TMB

PRT. Button pressed: §§TMA
PRT. Action-code: §§TMB

' react on Mouseover
IVS.§§TMB=!M!O!
 PIS.$$PIN|sbbc|§§TMA|$$COM
EIF.

' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
 PIS.$$PIN|sbbc|§§TMA|$$COL
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' react on Buttonpress
IVS.§§TMB=!P!
 SCS.§§TMA
 CAN.5
 JMP.enx
 CAN.1
 JSR.doButton01
 CAN.2
 JSR.doButton02
 CAN.3
 JSR.doButton03
 CAN.4
 JSR.doButton04
 CAE.
' PRT.Undefined Button-Number: §§TMA
 : $$RET=
 ESC.
EIF.
RET.
```

Remarks

You can use up to 5 Plugins. Just start them, each will get another Index.

Limitations:

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4 PIS. - PlugIn Send

3.71.4.1 ! Buttonbar for Blender-Sample Code

MiniRobotLanguage (MRL)

Buttonbar for Blender-Sample Code

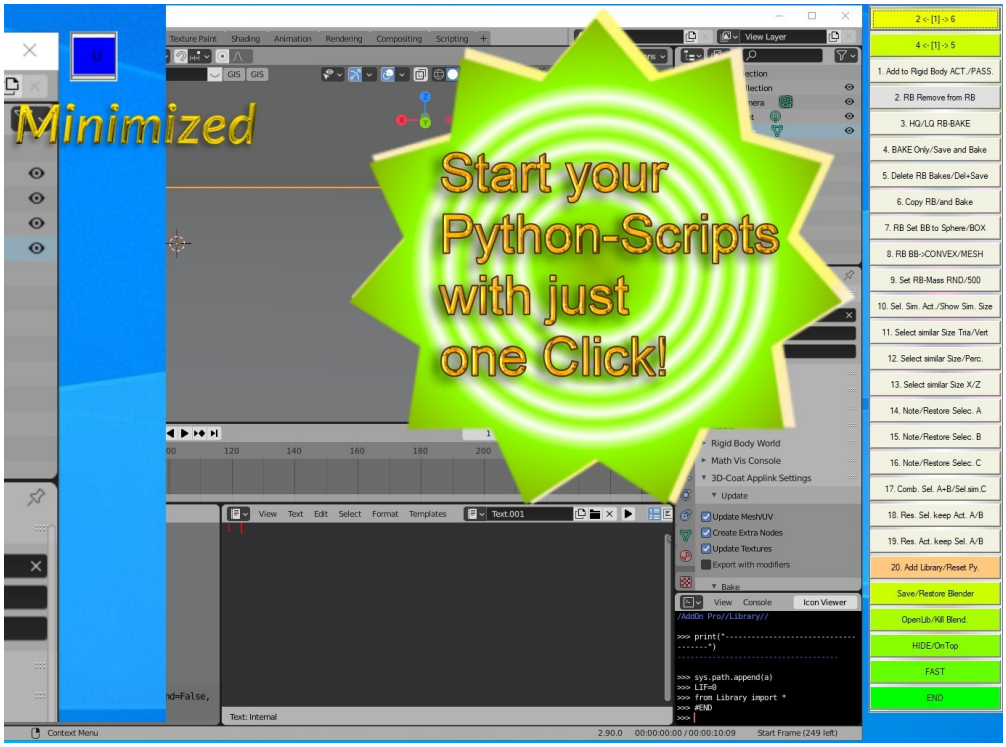
This Script is a commercial Blender AddOn

Intention

Using the SPR and its PlugIn we can realize a "Companion" for any Software that accepts Mouse and Keyboard Input.
The Code below realizes the Blenderbar Pro AddOn.It was developed over time so it does not yet use all the new Features for Textfiles.
However the idea to implement them has to do with these tasks.

The AddOn is available at Blendermarket here:
https://blendermarket.com/products/blender-quickbuttons-buttonbar

but as you have the SPR you can just compile it yourself.



*© 2023 Theo Gottwald * https://www.smart-package.com*

Code Explanation

Example

```
'
'SPR Script-file: Blender-AddOn
'Purpose: V.5
'Author:Theo Gottwald
'Creation date: 26-08-2020 at 12:19:34
'============================================================
'#EXE:?path\Quickbuttons.exe
'#SPI:ForceWrite
'#SPI:NoMutex
'#ICS:32
'#MAN:ai
'
'MRK.1

NAI.Blender
  MBX.Blender is not installed on your System.
  GTO.enz
EIF.

' Debugmodus bei Problemen
VAR.$$CMD=$cmdexe$
JNS.$$CMD|nodeb
VBT.$$CMD
VTL.$$CMD
IVV.$$CMD="-deb"
  DBM.2
EIF.
:nodeb

GSB.Definitions
' use any window to attach the plugin
GSB.FindBlender

GSB.CheckSlow

' Maximale Anzahl Layer (Button-Ebenen)
GSB.Count_Layers
POP.$$LAM
' Bigstep ausrechnen
CAL.$$BIO=$$LAM/2|i
GSB.FirstRun

GSB.Init_plugin
' Setze Texte der Buttons
PUV.$$LAY
GSB.Init_Button_Layer

' Make the PlugIn visible
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PIS.$$REA|show

DOL.1
  GSB.CloseIfDestroyed
  PIW.$$REA|$$RET
  GSB.Auswertung
  PAU.0.1
OOP.

PAU.2
:enz
END.
'-------------------------------------------------------------
:CloseIfDestroyed
NEW.$$ULT
  GTO.enz
EIF.
RET.

'-------------------------------------------------------------
' Switch Layers
'-------------------------------------------------------------
:NextLayer
' Falls nur ein Layer
JIV.$$LAM=1|enl
$$A00=$$LAY
VIC.$$LAY
:Layerpro
IVV.$$LAY>$$LAM
  CAL.$$LAY=$$LAY-$$LAM
EIF.
IVV.$$LAY<1
  CAL.$$LAY=$$LAY+$$LAM
EIF.
:NewLayer
PUS.$$LAY
GSB.Init_Button_Layer
:enl
RET.
'-------------------------------------------------------------
:FastLayer
' Falls nur ein Layer
JIV.$$LAM=1|enl
$$A00=$$LAY
CAL.$$LAY=$$LAY+2
GTO.Layerpro
RET.
'-------------------------------------------------------------
' Bei Doppelklick
:RevLayer
' Falls nur ein Layer
JIV.$$LAM=1|enl
$$A00=$$LAY
CAL.$$LAY=$$LAY-1
IVV.$$LAY=0
  VAN.$$LAY=$$LAM
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GTO.Layerpro
RET.
'-------------------------------------------------------------
' Input: number der Button-Ebene auf STACK
:Init_Button_Layer
GSB.CountTextFiles
GSB.SetButtonText
' Tooltip-Paralelrobot informieren
$$MSG=GDA.$$TBN-$$LAY
SIK.qued Message|$$TIA|$$MSG
RET.
'-------------------------------------------------------------
' Count Textfiles per Button-Ebene
' Input Ebene Number on Stack
' Output Number of Buttons in Ebene on Stack
:CountTextFiles
SAV.Save|$$LOP|$$NTX|$$PAT
GSB.GenLFP
POP.$$PAT
$$PBT=$$PATButton
VAN.$$BNU=0
FOR.$$LOP|1|99
  VAR.$$NTX=$$LOP
  GSB.ConstructCompleteFN
  IEF.$$NFN
    VAN.$$BNU=$$LOP
  ELS.
    GTO.LoopExit
  EIF.
NEX.
:LoopExit
PUV.$$BNU
SAV.Restore
RET.

'-------------------------------------------------------------
' $$NTX ist die Zahl die eingesetzt wird
:Replace
FMT.$$NTX|00
VAR.$$NFN=$$PBT
RPL.$$NFN|XX|$$NTX
RET.
'-------------------------------------------------------------
' $$RMB=[OnRightMouseButton!]
'-------------------------------------------------------------
:RMB_Process
TXT.Find_Line_with|$$RMB|$$POS
IVV.$$POS>0
  TXT.Get_after_Line|$$POS|$$BTR
  TXT.Get_before_Line|$$POS|$$BTL
ELS.
 TXT.Get_pure_Text|$$BTT
EIF.
IVV.$$RMF=1
    VAR.$$BTT=$$BTR
  ELS.
    ACH.$$BTL|$$CRL|2|$$BTT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  EIF.
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:writeCMD
SCW.h|$$ULT
GSB.GetCoords
GSB.ReplaceSV
GSB.clickType
:doNotType
RET.

'-------------------------------------------------------------
' Click and Type into Blender
'-------------------------------------------------------------
:clickType
SFW.$$ULT
SAW.$$ULT
GMP.msk|$$MSX|$$MSY
MLE.$$CLX,$$CLY
MDI.$$CLX,$$CLY
WII.5|$$PID
MLM.$$CLX,$$CLY
GSB.typeIt
MPO.s|$$MSX,$$MSY
RET.
'-------------------------------------------------------------
' Here we are going to type into the Python Console Window
:typeIt
IVV.$$SLO=1
  SKP.$$CMD
ELS.
  CLP.r|$$CLP
  $$CMD=$$CMD$crlf$
  CLP.t|$$CMD
  KYD.{STRG},"v"
  PAU.150|ms
  KYU.{STRG},"v"
  PAU.100|ms
EIF.
CLP.t|$$CLP
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------

' Replace all those Special Variables like [?lib] etc. with
the final content
:ReplaceSV
IVC.$$BTT=filename=
  ' Get First Line (Buttonname)
  SBD.$$BTT|$crlf$|1|$$FIA
  SBD.$$BTT|$crlf$|2|$$FIB
  SBD.$$BTT|$crlf$|3|$$FIC
  ' Hole Pfad nach =
  SBD.$$FIB|2|=|$$FID
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  VAR.$$BTT=$$FIA$crlf$$$FIB$crlf$$crlf$$$FIC
EIF.
NVC.$$BTT=[
  GTO.NoSpecial
EIF.
IVC.$$BTT=$$DFD
  VAR.$$POS=$tos$
  GSB.FileDialog
EIF.
' Work on [?lib]
IEF.$$LIA\
  RPL.$$BTT|$$COD|$$LIB
EIF.
IVC.$$BTT=$$UNF
  RND.0|9999|$$RAN
  FMT.$$RAN|0000
  VBT.$$RAN
  RPL.$$BTT|$$UNF|$$RAN|eb
EIF.
' Work on [INP]
IVC.$$BTT=$$COI
  VAR.$$POS=$tos$
  GSB.SpecialOpt
EIF.
' Work on [INP1] etc.
IVC.$$BTT=$$CIN
  GSB.SpecialOptB
EIF.
:NoSpecial
' Prevent too many Linefeeds in Result
$$CMD=$crlf$$$BTT$crlf$
RPL.$$CMD|$crlf$$crlf$|$crlf$
RET.
'-------------------------------------------------------------
'[?File?]
:FileDialog
CAL.$$TMB=$$POS+7
STR.DELETE|$$BTT|$$POS|$$TMB
CLP.r|$$CLI
VRT.$$CLI|\
$$CLI=$$CLI\
IEF.$$CLI
  $$STA=$$CLI
ELS.
  $$STA=?desktop\
EIF.
:retry
DFD.$$FID|Please choose the File|$$STA
JIT.retry
NEF.$$FID
  MBX.File does not exist. Please try again.||File-Warning.
  GTO.retry
EIF.
STR.INSERT|$$BTT|$$POS|$$FID
RET.
'-------------------------------------------------------------
'-------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' $$ALJ -> Letztes Input
' [ in $$POS
:SpecialOpt
VAR.$$BTT=$$BTT$crlf$
' Input in $$ALI
CAL.$$TMC=$$POS+4
' [INP] löschen
STR.DELETE|$$BTT|$$POS|$$TMC
$$EOL=$crlf$
VAR.$$WHA=
STR.FIND|$$BTT|$$POS|$$EOL
' $$WHE ist nun Zeilenende hinter [INP]
VAR.$$WHE=$tos$
' Falls keine Parameter hinter [INP] sind
VAR.$$ALK=
IVV.$$WHE=$$POS
  $$WHA=
  VAR.$$WHB=$$WHA
ELS.
' $$WHA contains all Characters behind the [INP] until EOL
  GSS.$$POS|($$WHE-1)|$$BTT|$$WHA
' Aus Quellstring löschen
  STR.DELETE|$$BTT|$$POS|($$WHE-1)
'Alles hinter "!" wird als Default-Value genommen
  IVC.$$WHA=!
    VAR.$$POT=$tos$
    GSS.1|($$POT-1)|$$WHA|$$WHB
    GSS.($$POT+1)|0|$$WHA|$$ALK
  ELS.
    VAR.$$WHB=$$WHA
  EIF.
EIF.
LEN.$$ALJ|$$LEN
IVV.§§TMC=§§TML
  IVV.$$LEN<2
    VAR.$$ALJ=$$ALK
  EIF.
EIF.
RPL.$$BTT|$crlf$$crlf$|$crlf$
RPL.$$BTT|$sp$$sp$|$sp$
GUT.$$WHB||$$ALJ|$$ALI
PAU..5
VBT.$$ALI
VAR.$$ALJ=$$ALI
VBT.$$ALI|,
  ' $$ALC enthält Anzahl der ","
GSC.$$ALI|,|$$ALC
  ' Anzahl der übergeben Parameter in $$ALC
VIC.$$ALC
VAR.§§TML=§§TMC
RET.
'-------------------------------------------------------------
' Work on [INP1]
'          12345
:SpecialOptB
NVC.$$BTT=$$CIN
  GTO.NoSpec
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
VAR.$$POA=$tos$
CAL.$$POE=$$POA+5
CAL.$$POF=$$POE+1
' Testen ob korrekte Syntax [INP1] ... [INP9]
GSS.$$POE|$$POE|$$BTT|$$TMA
IVS.$$TMA!]
  GTO.NoSpec
EIF.
' Holen der Zahl hinter INP in $$NUA
CAL.$$POG=$$POA+4
GSS.$$POG|$$POG|$$BTT|$$NUA
IVV.$$NUA|<|1
  GTO.NoSpec
EIF.
' Ist Parameter mit der number übergeben worden?
IVV.$$NUA|>|$$ALC
  GTO.NoSpec
EIF.
' Parameter holen in $$NUB
SBD.$$ALI|,|$$NUA|$$NUB
' Im Original ersetzen
VAR.$$NUC=[INP$$NUA]
RPL.$$BTT|$$NUC|$$NUB
' Können etliche Paramter in jeder Zeile sein
GTO.SpecialOptB
:NoSpec
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------

'------------------------------------------------------------
'
'------------------------------------------------------------
' Resultat in $$NOE und auf Stack
:Count_Layers
VAN.$$NOE=0
FOR.$$LOP|1|9
  PUS.$$LOP
  GSB.GenLFP
  POP.$$DIN
  IEF.$$DIN
    VIC.$$NOE
  ELS.
    GTO.Outc
  EIF.
NEX.
:Outc
PUV.$$NOE
RET.
'------------------------------------------------------------
' Layernumber in $$LAY
' Buttonnumber in $$NTX
' Rückgabe in $$NFN
:ConstructCompleteFN
FMT.$$LAY|0
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
FMT.$$NTX|00
VAR.$$NFN=$$LAP$$LAY\Button$$NTX.txt
RET.
' Generate Layer-Folder-Name
' Input Number of Layer on Stack
' Output: Folder Name like BE2 on Stack
:GenLFP
POP.$$A01
FMT.$$A01|1
VAR.$$A00=$$LAP$$A01\
VBT.$$A00
PUV.$$A00
RET.
'-------------------------------------------------------------

'-------------------------------------------------------------

'-------------------------------------------------------------
'
'-------------------------------------------------------------
' Check if First Run
:FirstRun
' Default button-Text
VAR.$$NTX=1
VAR.$$LAY=1
GSB.ConstructCompleteFN
NEF.$$NFN
  GSB.FirstRunB
EIF.
RET.
'-------------------------------------------------------------
:FirstRunB
$$PBT=$$OFN
' Create 9 Button_Files
FOR.$$LOP|1|9
  VAR.$$DEF=#Button $$LOP$crlf$$$CMA
  VAR.$$NTX=$$LOP
  VAR.$$NTX=$$LOP
  VAR.$$LAY=1
  GSB.ConstructCompleteFN
  NEF.$$NFN
    CTF.$$NFN|$$DEF
  EIF.
NEX.
RET.
'-------------------------------------------------------------
' FAST and SLOW
'-------------------------------------------------------------
:SetOpt
CAL.$$SLO=1-$$SLO
IVV.$$SLO=1
  NEF.$$SLF
    CTF.$$SLF|Slowmode
  EIF.
ELS.
  DEL.$$SLF
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.SetTxt
RET.
'-------------------------------------------------------------
' Prüfe auf SLOW.txt
:CheckSlow
IEF.$$SLF
  $$SLO=1
ELS.
  $$SLO=0
EIF.
RET.
'-------------------------------------------------------------
:SetTxt
IVV.$$SLO=1
  $$SPD=SLOW
ELS.
  $$SPD=FAST
EIF.
PIS.$$REA|sBt|$$SP2|$$SPD
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:SaveWinSizeIf
CFI.$$SEI|Blender|SizeSet|$$SEJ
IVV.$$SEJ=0
  GSB.SaveWinSize
EIF.
RET.
'-------------------------------------------------------------
:SaveWinSize
SCW.h|$$ULT
GWP.$$XPO,$$YPO|$$XSZ,$$YSZ
CTI.$$SEI|Blender|XPos|$$XPO
CTI.$$SEI|Blender|YPos|$$YPO
CTI.$$SEI|Blender|XSize|$$XSZ
CTI.$$SEI|Blender|YSize|$$YSZ
CTI.$$SEI|Blender|SizeSet|1
RET.
'-------------------------------------------------------------
:RestoreWinSize
CFI.$$SEI|Blender|SizeSet|$$SEJ
IVV.$$SEJ=1
  CFI.$$SEI|Blender|XPos|$$XPO
  CFI.$$SEI|Blender|YPos|$$YPO
  CFI.$$SEI|Blender|XSize|$$XSZ
  CFI.$$SEI|Blender|YSize|$$YSZ
  SCW.h|$$ULT
  SWP.$$XPO,$$YPO|$$XSZ,$$YSZ
EIF.
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:RunIt
IEF.$$DMM
  EXS.$$DMM
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  MBX.Sorry ...$crlf$$$DMM$crlf$ not found.
EIF.
RET.
'-------------------------------------------------------------
:RunButtonswitcher
VAR.$$DMM=?exeloc\SwitchButton.exe
GSB.RunIt
RET.
'-------------------------------------------------------------
:RunDesktops
VAR.$$DMM=?exeloc\Desktops_Manager.exe
GSB.RunIt
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Definitions
' Get Screen DPI
DPI.s|$$DPX,$$DPY
VAF.$$SLF=?exeloc\Slow.txt
VAF.$$WIF=?exeloc\Blendersize.txt
' The Prefix of the Directories with the Buttons
VAR.$$LYN=BE
'VAF.$$OFN=?exeloc\$$LYN?\Button**.txt
VAR.$$LAP=?exeloc\$$LYN
VAR.$$LIA=?exeloc\Library
VAR.$$LIB=?exeloc\
RPL.$$LIB|\|/
RPL.$$LIB|//|/
RPL.$$LIB|/|//|b
VAR.$$ADL=Add Library
' This will be replaced with the path to the Library Folder
VAR.$$COD=[?lib]
VAR.$$DFD=[?File?]
VAR.$$FOD=filename=
' Sample Code for Sample-Buttons
VAR.$$CMA=bpy.ops.ptcache.free_bake_all()$crlf$bpy.ops.ptcache
.bake_all(bake=True)$crlf$
' The Delimiter between the Python-Code for left and right
Mousebutton
VAR.$$RMB=[OnRightMouseButton!]
' This will be replaced with a 4 digit random number
VAR.$$UNF=[?RND4?]
' Dispaly User-Input Box. Input is comma separated and goes to
[INP1],[INP2]...
VAR.$$COI=[INP]
' This is the prefix for [INP1][INP2]...
VAR.$$CIN=[INP
' Aktueller Button-Layer (Button-Ebene)
VAN.$$LAY=1
' Default Input ist leer
VAR.$$ALJ=
' Button-Farben
$$COM=&HA2A2A2
$$COL=&HA0A0A0
' Button-Ordner 1 erstellen falls nicht vorhanden
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$FOA=?exeloc\BE1\
NEF.$$FOA
  MKD.$$FOA
EIF.
' Last Button-Number
VAR.$$TML=0
' Immer 2 Top-Buttons
VAN.$$TOB=2
' Settings about the BB
VAR.$$SEI=?exeloc\Settings.ini
VAR.$$HID=0
VAR.$$CRL=$crlf$
RET.
'------------------------------------------------------------
:FindBlender
NEP.Blender.exe
  :RunOwn
  GEP.XXXXX.blend|$$BIS
  EXE.$$BIS|-con
  $$PID=#pid#
  STW.BctoXwp|Ghost_Windowclass|Blender|noBlend|250|$$PID
ELS.
' Falls Blender auf anderem Desktop läuft oder minimized ist
  STW.BctoXw|Ghost_Windowclass|Blender|RunOwn|5
  IWI.t
    GTO.RunOwn
  EIF.
  $$PID=#pid#
EIF.
' Store the window handle in a variable
HTV.$$ULT

CFI.$$SEI|Blender|SizeSet|$$SEJ
IVV.$$SEJ=1
  GSB.RestoreWinSize
ELS.
' Neue Fenstergröße ausrechnen
  CAL.$$NSX=#wsx#/100*90|i
  CAL.$$NSY=#wsy#/100*90|i
  MBX. $$NSX-$$NSY
  SWP.0,0|$$NSX,$$NSY|F
   ' Blender-WindSize speichern wenn noch nicht geschehen
  GSB.SaveWinSizeIf
EIF.

' Treat Console-Window
STW.Xctwop|ConsoleWindowClass|blender|0|noCons|$$PID
SWP.rb,bb||T
:noCons

:GetCoords
' Koordinaten des Blender Fensters unten rechts
VAR.$$RIG=#ir#
VAR.$$BOT=#ib#

' Klick-Koordinaten
CAL.$$CLX=$$RIG-100
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CAL.$$CLY=$$BOT-100
RET.
'-------------------------------------------------------------

'-------------------------------------------------------------
:noBlend
MBX.I did not find the Main-Window from Blender.$crlf$Please
get support.$crlf$Program ends.
END.
RET.
'-------------------------------------------------------------
' Here is the Management for the System-Buttons
'-------------------------------------------------------------
' Input: Number of Buttons on Stack
' $$TBN - total Button Number wird gesetzt auf totale Button-
Anzahl inkl. END und Switch
'
:SetButtonText
' Zeiger auf 1. Button wird immer erhöht
VAN.$$PBZ=1
' Button Anzahl ohne Systembutton
POP.$$BNU
' Nur auf Layer 1 die Buttons END und FAST
CAL.$$LAI=$$PBZ-1
IVV.$$LAY!1
  CAL.$$TBN=$$BNU+3
ELS.
  CAL.$$TBN=$$BNU+7
  'Zwei Switch Button auf L1 und 4 System-Buttons
EIF.

' Buttionanzahl setzen
PIS.$$REA|sbn|$$TBN

' gelber Switch Layer-Button
GSB.Paint_SwitchLayerButton

GSB.Paint_SwitchFast

' Button Texte setzen
FOR.$$LOP|1|$$BNU
  VAR.$$NTX=$$LOP
  GSB.ConstructCompleteFN
  IEF.$$NFN
    CAL.$$FIB=$$PBZ
    TXT.Load_Textfile|$$NFN
    TXT.Get_Line_Trimmed|1|$$BTX
    VLT.$$BTX|#
    PIS.$$REA|sBt|$$FIB|$$LOP. $$BTX
    GSB.SetBtCol
  EIF.
  VIC.$$PBZ
NEX.

' Spezialbuttons
IVV.$$LAY=1
' Add Systen-Button with Multi-Functions
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GSB.MultiButton

' Add Openlib-Button
  GSB.OpenLib
  GSB.AddHide
' Add END and FAST if Layer 1
  GSB.AddEFast
EIF.
GSB.AddEnd
RET.
'-------------------------------------------------------------
:SetBtCol
IVC.$$BTX=$$ADL
  ' Adlib-Farbe ' Orange  - &H00AAFF/80CAFF  rot - &HE2E2FF
drk.blue: &H80CAFF
  IVV.$$ADO=0
    $$COL=&H80CAFF
    $$ADO=1
  ELS.
    $$COL=&HFFFAA0
  EIF.
ELS.
  IVC.$$BTT=$$RMB
    $$COL=&HE2F2F2
  ELS.
    $$COL=&HE0E0E0
  EIF.
EIF.
:SetCol
PIS.$$REA|sbbc|$$FIB|$$COL
RET.
'-------------------------------------------------------------
:Set_Fast_Values
SCS.$$LAM
CAN.1
  $$LAE=1
  $$LAD=1
  GTO.Paint_Fast
CAN.2
  $$LAE=1
  $$LAD=2
  GTO.Paint_Fast
CAE.
  $$BIU=2
ESC.
' rechte Seite
CAL.$$LAD=$$LAY+$$BIO
IVV.$$LAD>$$LAM
  CAL.$$LAD=$$LAD-$$LAM
EIF.
' linke Seite
CAL.$$LAE=$$LAY-$$BIU
IVV.$$LAE<1
  CAL.$$LAE=$$LAE+$$LAM
EIF.
:Paint_Fast
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:Paint_SwitchFast
$$SLC=$$PBZ
GSB.Set_Fast_Values
VAV.$$TXA=$$LAD <- [$$LAY] -> $$LAE
PIS.$$REA|sBt|$$SLC|$$TXA
PIS.$$REA|sbbc|$$SLC|&H00FFCC
' Zeiger auf nächsten Button im Plugin
VIC.$$PBZ
RET.
'-------------------------------------------------------
:Paint_SwitchLayerButton
$$SLB=$$PBZ
CAL.$$LAD=$$LAY+1
IVV.$$LAD>$$LAM
  $$LAD=1
EIF.
CAL.$$LAE=$$LAY-1
IVV.$$LAE<1
  $$LAE=$$LAM
EIF.

VAV.$$TXA=$$LAD <- [$$LAY] -> $$LAE
PIS.$$REA|sBt|$$SLB|$$TXA
' Orange  - &H00AAFF
PIS.$$REA|sbbc|$$SLB|&H00FFEE
' Zeiger auf nächsten Button im Plugin
VIC.$$PBZ
RET.
'-------------------------------------------------------
' UNUSED
'  $$TXA=OpenLib
'  $$TXA=Save Blender-Size
'  $$TXA=Restore Blender-Size
'  $$TXA=Run Desktops Manager
'  $$TXA=Run ButtonSwitcher
'-------------------------------------------------------
:MultiButton
' Spezialbutton 1 (SP1-SP4)
VAN.$$SP4=$$PBZ
$$TXA=Save/Restore Blender
PIS.$$REA|sBt|$$SP4|$$TXA
PIS.$$REA|sbbc|$$SP4|&H00FFCC
VIC.$$PBZ
RET.
'-------------------------------------------------------
:OpenLib
' Spezialbutton 1 (SP1-SP4)
VAN.$$SP3=$$PBZ
$$TXA=OpenLib/Kill Blend.
PIS.$$REA|sBt|$$SP3|$$TXA
PIS.$$REA|sbbc|$$SP3|&H00FFAA
VIC.$$PBZ
RET.
'-------------------------------------------------------
:AddEFast
' Fast-Button
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAN.$$SP2=$$PBZ
PIS.$$REA|sBt|$$SP2|Fast
PIS.$$REA|sbbc|$$SP2|&H00FF66
' Text für FAST-Button
GSB.SetTxt
VIC.$$PBZ
RET.
'-------------------------------------------------------------
:AddHide
' Hide-Button
VAN.$$SP5=$$PBZ
PIS.$$REA|sBt|$$SP5|HIDE/OnTop:$$I09
PIS.$$REA|sbbc|$$SP5|&H00FF88
VIC.$$PBZ
RET.
'-------------------------------------------------------------
:AddEnd
' END-Button
VAN.$$SP1=$$PBZ
PIS.$$REA|sBt|$$SP1|END
PIS.$$REA|sbbc|$$SP1|&H00FF00
VIC.$$PBZ
RET.
'-------------------------------------------------------------
:Lab_OnTop
CAL.$$I09=1-$$I09
PIS.$$REA|stm|$$I09
PIS.$$REA|sBt|$$SP5|HIDE/OnTop:$$I09
RET.
'-------------------------------------------------------------
:Lab_HIDE
VAN.$$HID=1
PIS.$$REA|sbn|1
PIS.$$REA|sbhs|40
PIS.$$REA|sBt|1|U
PIS.$$REA|sbbc|1|&HFF0000
RET.
'-------------------------------------------------------------
:Lab_UnHide
PIS.$$REA|sbhs|$$I08
GSB.Init_Button_Layer
VAN.$$HID=0
RET.
'-------------------------------------------------------------
' Tooltips nur wenn nicht "OnTop"
:Tooltips
IVV.$$I10=1
  IVV.$$I09=0
    PIS.$$REA|gmb
    VAN.$$ATT=$tos$
    PIS.$$REA|gh|$$ATT
    VAN.$$ATU=$tos$
  ' Button-Nr. - Layer-Nr. - Buttonhwnd.
    $$MSG=ENT.$$ATT-$$LAY-$$ATU
    SIK.qued Message|$$TIA|$$MSG
    $$ATM=$$ATL
  EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
RET.
'-------------------------------------------------------------
:OnLeave
IVV.$$I10=1
  IVV.$$I09=0
    PIS.$$REA|gmb
    VAN.$$ATL=$tos$
    $$MSG=LEA.$$ATT-$$LAY
    SIK.qued Message|$$TIA|$$MSG
  EIF.
EIF.
RET.
'-------------------------------------------------------------
:Auswertung
'DBP.$$RET
' Get Button number out of $$RET into §§TMA
SBD.$$RET|$sp$|1|§§TMA-$$LAY
' Button Action into §§TMB
SBD.$$RET|$sp$|2|§§TMB
JIS.§§TMB=!normal!|enk
JIS.§§TMB=!Closed!|enk
JIS.§§TMB=!Iconic!|enk
JIS.§§TMB=!nsize!|enk
'-------------------------------------------------------------
' HIDE-Button
IVV.$$HID=1
  IVV.§§TMA=1
    GSB.Lab_UnHide
  EIF.
  GTO.enk
EIF.
' Tooltips
IVS.§§TMB=!M!O!
  GSB.Tooltips
  GTO.enk
EIF.
IVS.§§TMB=!M!L!
  GSB.OnLeave
  GTO.enk
EIF.

'-------------------------------------------------------------
' Normal Click LMB
SCS.§§TMB
CAS.!P!
  VAR.$$RMF=0
  GTO.More
' react on right Buttonpress
CAS.!R!
' Switch Button-Layer
  VAR.$$RMF=1
  GTO.More
CAS.!P!D!
' Left Doubleclick
' Switch Button-Layer fast
  IVV.§§TMA=$$SLB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    GSB.FastLayer
    MSB.4170|1
    GTO.enk
  EIF.
CAE.
' Doppelklick woanders wie Klick
  $$RMF=0
  GTO.More
ESC.

' Für alle Layer gemeinsam
:More
  ' Switch Button-Layer
SCS.§§TMA
CAN.$$SLB
  IVV.$$RMF=1
    GSB.RevLayer
    GTO.enk
  ELS.
    GSB.NextLayer
    GTO.enk
  EIF.
CAN.$$SLC
  GSB.Set_Fast_Values
  IVV.$$RMF=0
    CAL.$$LAY=$$LAD
  ELS.
    CAL.$$LAY=$$LAE
  EIF.
  GSB.NewLayer
  GTO.enk
ESC.

' END abprüfen
IVV.§§TMA=$$SP1
  GTO.enz
EIF.

:OnlyOneSB
IVV.$$LAY=1
  SCS.§§TMA
  CAN.$$SP5
    IVV.$$RMF=1
      GSB.Lab_OnTop
    ELS.
      GSB.Lab_HIDE
    EIF.
  CAN.$$SP4
    IVV.$$RMF=0
      GSB.SaveWinSize
    ELS.
      GSB.RestoreWinSize
    EIF.
    GTO.enk
  CAN.$$SP3
    IVV.$$RMF=1
      GSB.KillBlend
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    ELS.
   ' OpenLib nur in Button-Layer 1
      IEF.$$LIA\
        EXS.$$LIA\
      EIF.
    EIF.
    GTO.enk
  CAN.$$SP2
    ' FAST-Button
    GSB.SetOpt
  CAE.
    GTO.DoMyButtons
  ESC.
  GTO.enk
EIF.

:DoMyButtons
' Alle Anderen Buttons hier
' $$TOB - Anzahl der Top-Buttons
CAL.$$NTX=§§TMA-$$TOB
JIV.$$NTX>$$SP1|enk
'MBX.$$NTX$crlf$§§TMA$crlf$$$TOB
GSB.ConstructCompleteFN
NEF.$$NFN
  MBX.File "$$NFN" can not be loaded.
  GTO.enk
EIF.
GSB.GetTextFile
GSB.RMB_Process
GSB.writeCMD
:enk
RET.
'-------------------------------------------------------------
' Tested Ready
'-------------------------------------------------------------
:KillBlend
MBT.Press Ok to Kill Blender-Prozess!|Want Kill Blender?|32|
(4+48+256)|$$RES
IVV.$$RES=6
  CPR.$$PID|5000
EIF.
GTO.FindBlender
RET.
'-------------------------------------------------------------
:Init_plugin
PIL.$$REA|01
PIS.$$REA|hide
PIS.$$REA|GetHandle
$$PIH=$tos$
NEF.$$SEI
' ButtonHdistance
  VAN.$$I00=4
  CTI.$$SEI|Buttons|ButtonHdistance|$$I00
  VAR.$$I01=sv
  CTI.$$SEI|Position|show|$$I01
  VAN.$$I02=-2
  CTI.$$SEI|Position|Adjust_A|$$I02
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  VAN.$$I04=-5
  CTI.$$SEI|Position|Adjust_B|$$I04
' attach the Plugin to the window
' 1 - Links 2 - Rechts 3 - links inside, 4 - rechts inside
' 5  - links inside bündig, 6 -rechts inside bündig
'
  VAN.$$I05=2
  CTI.$$SEI|Position|Side_H|$$I05
  VAN.$$I06=1
  CTI.$$SEI|Position|Side_V|$$I06
  VAN.$$I07=4
  CTI.$$SEI|Buttons|ButtonVdistance|$$I07
  VAN.$$I08=180
  CTI.$$SEI|Buttons|Length|$$I08
  VAN.$$I09=1
  CTI.$$SEI|Position|OnTop|$$I09
  VAN.$$I10=1
  CTI.$$SEI|Tooltips|Enabled|$$I10
ELS.
' Buttondistance
  VAN.$$I00=4
  CFI.$$SEI|Buttons|ButtonHdistance|$$I00
  VAR.$$I01=sv
  CFI.$$SEI|Position|show|$$I01
  VAN.$$I02=-2
  CFI.$$SEI|Position|Adjust_A|$$I02
  VAN.$$I04=-5
  CFI.$$SEI|Position|Adjust_B|$$I04
' attach the Plugin to the window, 2 - right side, 1 - ganz
oben
  VAN.$$I05=2
  CFI.$$SEI|Position|Side_H|$$I05
  VAN.$$I06=1
  CFI.$$SEI|Position|Side_V|$$I06
  VAN.$$I07=4
  CFI.$$SEI|Buttons|ButtonVdistance|$$I07
  VAN.$$I08=180
  CFI.$$SEI|Buttons|Length|$$I08
  VAN.$$I09=0
  CFI.$$SEI|Position|OnTop|$$I09
  VAN.$$I10=1
  CFI.$$SEI|Tooltips|Enabled|$$I10
EIF.
' Plugin vertikal anordnen Buttondistanz 4 Pixel
PIS.$$REA|sbhd|$$I00
PIS.$$REA|$$I01
PIS.$$REA|scf|$$I02|0|$$I04
PIS.$$REA|attach|$$ULT|$$I05|$$I06
PIS.$$REA|hide
' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|$$I07
' Setze Breite der Buttons
PIS.$$REA|sbhs|$$I08
PIS.$$REA|stm|$$I09
IVV.$$I10=1
  GSB.Parallel
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'-------------------------------------------------------------
'-------------------------------------------------------------
' Helps with Plugin and Tooltipps
:Parallel
PRR.0
  $$ROH=#cbw#
  GDF.hp|$$ROH|$$PII
  VAR.$$RUN=1
  DOL.($$RUN=1)
    WIK.|Message Any|$$MSG
    GSS.2|0|$$MSG|$$MSG
    JIS.$$MSG=EXIT|enc
    GSS.1|3|$$MSG|$$CMD
    GSS.5|0|$$MSG|$$PAR
    SCS.$$CMD
    CAS.LEA
      GSB.PRR_Close_TT
    CAS.ENT
      GSB.PRR_Set_TT
    CAS.GDA
      GSB.PRR_GetData
    ESC.
  OOP.
  :enc
  ENR.
PRE.$$TIC
VAR.$$TIA=#pcun#
RET.
'-------------------------------------------------------------
' Tooltip-Engine
'-------------------------------------------------------------
' bekommt Anzahl der Buttons und Layernumber
:PRR_GetData
BCH.$$PAR|-||$$TBN
ACH.$$PAR|-||$$LAY
CAL.$$SP1=$$TBN
IVV.$$LAY=1
  CAL.$$SP2=$$TBN-1
  CAL.$$SP3=$$TBN-2
  CAL.$$SP4=$$TBN-3
  CAL.$$SP5=$$TBN-4
EIF.
RET.
'-------------------------------------------------------------
:PRR_Set_TT
BCH.$$PAR|-||$$BUT
ACH.$$PAR|-||$$LAY
ACH.$$PAR|-|2|$$HWN
'PRT.PL: Layer $$LAY --  Hwnd:$$HWN --- Button-Nr.$$BUT
IVV.$$LTT!$$BUT
  ' Switch Buttons
  IVV.$$BUT<3
    $$BTX=Use this Buttons to switch between the Button-
Layers.$crlf$
    $$BTX=$$BTXThey are made in a way that you will reach any
Layer within 1-2 Clicks.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    GTO.PRR_SetText
  EIF.
  CAL.$$NTX=$$BUT-2
  IVV.$$BUT=$$SP1
    $$BTX=*END* This Button will close the Buttonbar.
    GTO.PRR_SetText
  EIF.

  IVV.$$LAY=1
    SCS.$$BUT
    CAN.$$SP2
      $$BTX=*FAST* Here you can set the Typing-Speed of the
Buttonbar to FAST or to SLOW.
    CAN.$$SP3
      $$BTX=LMB:*HIDE*/Minimze Buttonbar.$crlf$RMB: Change
"OnTop"-Behavior from Buttonbar.$crlf$1 - BB is on Top (no
Tooltips).$crlf$0 - BB is "Normal", here Tooltipps are
possible.
    CAN.$$SP4
      $$BTX=LMB:*Openlib* this will open the Library-Folder in
an Explorer Window.$crlf$RMB:*Kill Blender* this will kill the
Blender-Process which the Buttonbar hangs on.$crlf$No Danger
for other running Blender Versions.
    CAN.$$SP5
      $$BTX=LMB:*SAVE* This will save the current Size and
Position of the Blender Window.$crlf$Into the Settings.ini-
File.$crlf$RMB:*RESTORE* this will restore the Blender Windows
Size and Position that was saved before.
    CAE.
      GTO.PRR_Getfile
    ESC.
    GTO.PRR_SetText
  ELS.
    GTO.PRR_Getfile
  EIF.
  :PRR_Getfile
  GSB.ConstructCompleteFN
  IEF.$$NFN
    GSB.GetTextFile
    GSB.PRR_TT
    GTO.PRR_SetText
  EIF.
EIF.
RET.
'-------------------------------------------------------------
:PRR_Close_TT
STW.ABcwXop|Ballontext|0|NoBall|$$PII
CLW.
GTO.PRR_Close_TT
:NoBall
RET.
'-------------------------------------------------------------
:GetTextFile
TXT.Load_Textfile|$$NFN
TXT.Get_pure_Text|$$BTT
TXT.Replace_in_Text|**_$crlf$|**
'CFF.$$NFN|$$BTT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'VLT.$$BTT|$$CRL$sp$
'RPL.$$BTT|**_$crlf$|**|e
RET.
'------------------------------------------------------------
:PRR_TT
SBD.$$BTT|$$CRL|2|$$BTX
RPL.$$BTX|**|$$CRL|e
RET.
'------------------------------------------------------------
:PRR_SetText
VAR.$$PAI=FS:10 T:50
SCW.h|$$HWN
SBT.$$BTX|$$PAI
$$LTT=$$PAR
RET.
'------------------------------------------------------------

'------------------------------------------------------------
END.
```

Remarks

-

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.2 ! Buttonbar for HM8-Sample Code

MiniRobotLanguage (MRL)

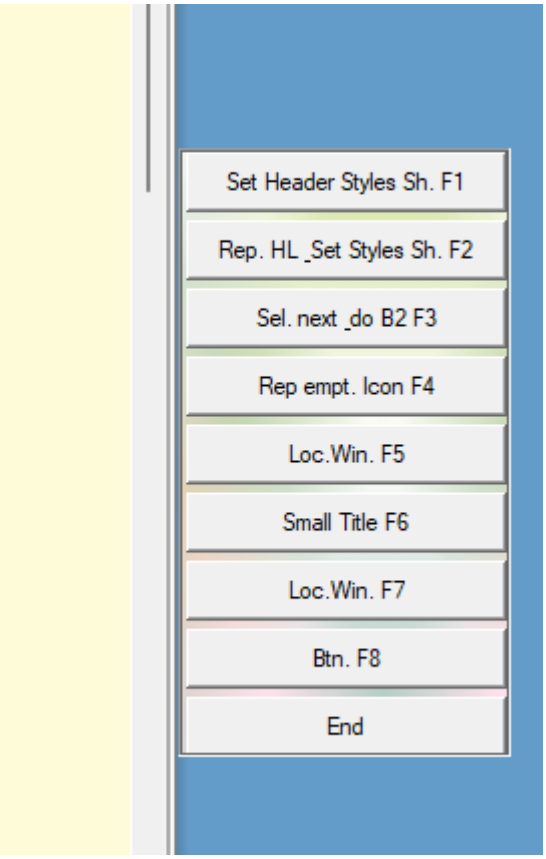
Buttonbar for Blender-Sample Code

This Script is a commercial Blender AddOn

Intention

Using the SPR and its PlugIn we can realize a "Companion" for any Software that accepts Mouse and Keyboard Input.

The Code below realizes an AddOn Buttonbar for "Help and Manual 8".
The functions may mostly be very special to my usage, but you can clearly see the general structure.



ButtonBar attached to the HM8 Window

Code Explanation

 *© 2023 Theo Gottwald * https://www.smart-package.com*

-

Example

```
'
'#################################################################
##########
' Button-Bar for Help and Manual 8
' As far ass Patterns are used, HM8 must be set to 100% Zoom
'#################################################################
##########
'#EXE:?path\
'#SPI:ForceWrite
'
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=9
PIS.$$REA|sbn|$$BUN

' use any window to attach the plugin
PIS.$$REA|sbhd|4


' Make the PlugIn visible
PIS.$$REA|show

$$DIR=v
IVS.$$DIR=v
' attach the Plugin to the window
  PIS.$$REA|attach|$$ULT|2|2
  PIS.$$REA|sv
  PIS.$$REA|SetcorrectionFactor|-2|-4|4|-4
ELS.
  PIS.$$REA|attach|$$ULT|5|6
' Plugin vertikal anordnen
  PIS.$$REA|sh
EIF.



$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|2|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7
' Set Distance between buttons (vertical in Points)
'PIS.$$REA|sbvd|4

PIS.$$REA|SetButtonsVDist|4

PIS.$$REA|SetTopMost|1

' Endlos-Schleife
DOL.1
  PIW.$$REA|§§BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.

SAV.Restore
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'-------------------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.$$BUN
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAN.4
    JSR.doButton04
  CAN.5
    JSR.doButton05
  CAN.6
    JSR.doButton06
  CAN.7
    JSR.doButton07
  CAN.8
    DBP.8 pressed!
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'-------------------------------------------------------------
' Do Button 1
'-------------------------------------------------------------
:doButton01
GSB.Set_Style_Headline
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.1
MLM.
GSB.Move_Mouse_To_B1
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
' Do Button 2
'-------------------------------------------------------
:doButton02
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.0.25
MLM.
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------
' Do Button 3
'-------------------------------------------------------
' Jump to next and then Replace Headers
:doButton03
GSB.Locate_below_Selected
PAU.1
MLM.
PAU.1
WII.6
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.Move_Mouse_To_Button|3
RET.
'-------------------------------------------------------
' Do Button 4
'-------------------------------------------------------
:doButton04
GSB.Locate_false_Icon
GSB.Locate_Icon_change_Menu
GSB.Locate_right_Icon
GSB.Move_Mouse_To_Button|4
RET.
'-------------------------------------------------------
' Do Button 5
'-------------------------------------------------------
:doButton05
FOR.$$LL5|1|8
  GSB.doButton03
  PAU.1
  IEK.LSHIFT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    ENR.
  EIF.
NEX.
GSB.Move_Mouse_To_Button|5
RET.
'-------------------------------------------------------------
' Do Button 6
'-------------------------------------------------------------
:doButton06
GSB.Do_Mark_MRL
GSB.Set_Style_Small_Page_Title
GSB.Locate_Topline_Start
MLC.
GSB.Move_Mouse_To_Button|6
RET.
'-------------------------------------------------------------
' Do Button 7
'-------------------------------------------------------------
:doButton07
GSB.Locate_TW
GSB.Move_Mouse_To_Button|7
RET.
'-------------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
ENR.
'=============================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------------
:WaitShift
WNK.
:Laba
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'-------------------------------------------------------------
:Locate_TW
IEW.$$ULT
  STW.h|$$ULT
ELS.
  STW.ct|TFrmHelpMan|Help+Manual
  HTV.$$ULT
  SWP.1090,130|1414,1665|T
EIF.
RET.
'-------------------------------------------------------------
:Open_Style_Menu
GSB.Locate_TW
'SWP.1666,287|297,93|T
'MMV.t|428,125
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControlSite|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControl|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupBarControl|Schriftart
MLI.w|117,61
WII.
PAU.0.1
RET.
'-------------------------------------------------------------
:Set_Font_Style
$$STL=§§_01
STW.Ac|TdxBarDropdown
AGR.34!WII.MOS.|$$STL||1|3|60
RET.
'-------------------------------------------------------------
:Do_Mark_MRL
GSB.Locate_Small_MRL
SKP.{END}
KYD.{LSHIFT}
SKP.{HOME}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------
:Do_Mark_Line
GSB.Locate_TW
SKP.{HOME}
KYD.{LSHIFT}
SKP.{END}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:Locate_Headline
GSB.Locate_TW
MLI.t|584,249
RET.
'-------------------------------------------------------
:Locate_Sub_Headline
GSB.Locate_TW
MLI.t|835,300
RET.
'-------------------------------------------------------
:Locate_right_Sub_Headline
GSB.Locate_TW
MLC.t|1126,298
RET.
'-------------------------------------------------------
:Mark_Headline
GSB.Locate_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------
:Mark_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------
:Mark_right_Sub_Headline
GSB.Locate_right_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------
:Mark_Full_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------
:Set_Style_Line_Style
GSB.Open_Style_Menu
GSB.Set_Font_Style|LineStile
RET.
'-------------------------------------------------------
:Set_Style_Small_Page_Title
GSB.Open_Style_Menu
GSB.Set_Font_Style|Small Page Title
RET.
'-------------------------------------------------------
:Set_Style_Headline
GSB.Mark_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
RET.
'-------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Set_Style_Sub_Headline
GSB.Mark_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading2
RET.
'-------------------------------------------------------------
:Set_Style_right_Sub_Headline
GSB.Mark_right_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading3
RET.
'-------------------------------------------------------------
:Move_Mouse_To_Button
$$STL=§§_01
VBT.$$STL
$$TXT=F$$STL
STW.ct|Form1_Class|MR
SCW.ct|ButtonEx_Class32|$$TXT
MPO.
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B1
GSB.Move_Mouse_To_Button|1
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B2
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Locate_selected_TV_Element
GSB.Locate_TW
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_7335.bmp
GSB.Inline_8970
VAR.$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:64 . We use -10%, P5 - Debugmode
0/1
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|57
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_7335
MMV.
RET.
'-------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  158 Bytes in  1 Lines.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Inline_8970
VDA.$$DTA|$$FIN=
  ?
MAAEOAQAASlHMfhfYzrQoxYI/Ftc3mnBtSjYSYBfiLaPgfbNgC2KBU1eeQhLy6
0MDuXfpRQj0rcuPtPBTGlvGu+z7cJO6C44/EnytF8yVNA+C1z1msKAtTqlmALt
GTSnMagDrPLpMHfmKmu/PGAAMCwhGVeUB
VDE.
RET.
'-------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_7335
MBX.Pattern <P_7335.bmp> not found.
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------

'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Locate_below_Selected
GSB.Locate_TW
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_7767.bmp
GSB.Inline_4035
VAR.$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:64 . We use -10%, P5 - Debugmode
0/1
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|57
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_7767
MPO.
RET.
'-------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_7767
MBX.Pattern <P_7767.bmp> not found.
RET.
'-------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  176 Bytes in  2 Lines.
:Inline_4035
VDA.$$DTA|$$FIN=
  ?
MAAEOAQAASlHMfhfYzgSkdHk/Ftc3mnBtSjbWqlA6LqThfbNgGIBNq1ifQhK67
kQRhd6S7rg6w+M4+phKZ88dX5Ceb4pRYGw/jP1odDNXOdAJNadnvmsd50FlzH8
eGH1C4K0umf1LEQGH80MiEAAJBP8uFbfuhuaT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  +HAC5NKvqmUB
VDE.
RET.

'------------------------------------------------------------
'
'------------------------------------------------------------



'------------------------------------------------------------
'
'------------------------------------------------------------
:Locate_false_Icon
GSB.Locate_TW
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_3630.bmp
GSB.Inline_3528
VAR.$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:64 . We use -10%, P5 - Debugmode
0/1
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|64
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_3630
MRI.
RET.
'------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_3630
MBX.Pattern <P_3630.bmp> not found.
RET.
'------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  284 Bytes in  2 Lines.
:Inline_3528
VDA.$$DTA|$$FIN=
  ?
MAAEOAQAASlF8bSHYBphXC9xgHyB6ivXH1nWYS+sYAFmXMobDia4QBMtGskeKO
YP6b4lGFQWpYq/uQo3z4fiYwLrIvMpnfgXqCEGBoXhwSBqddgeEx6Zl/zZX1yD
dgstU9u2J7Gzh+keW/8n0DziM9W9TR5bJULrf
  K/ByW5ot1u7OaYPvmufG+hB6dK/YGnvxzA7NxC79ZFGjmrWs2l+DTBUD5Uug
qFggIAAL9BYDL6jU22F2bgQwJ6bazg8168RuOrFhgEno9E4pJd+VDnxyQQE=
VDE.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'------------------------------------------------------------
'
'------------------------------------------------------------
:Locate_Icon_change_Menu
STW.c|TDXRibbonPopup
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_4134.bmp
GSB.Inline_7893
VAR.$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:128 . We use -10%, P5 - Debugmode
0/1
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|115
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_4134
MPO.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  378 Bytes in  3 Lines.
:Inline_7893
VDA.$$DTA|$$FIN=
  ?
MAAEOAgAASlEcTCPYthDB0N/zfCvTaAaNcwkJCQkJiclwh+ykrmEBgOlFDzqQH
+sam+dIeKy5C0JbASdrgTXeJl+Cy1UnneCoWcc7ymSyjz3FPBPEEOA1KfSf4ru
xxhcNLd4NAKjYfEuqeB3Q5M3xCtXY3wDQsU5h
  nlAungu0VmLgngs0fQ+P+j4KjDe8GF4i8b1wRqb2ecSYjRgcm9iGikgNM8UZ
wOzhuYqW1mrA+cTwBKvcRxZsC+/IlpSowd/CB3AO2ASn5oTMe3u3A3/Ua36ylS
Yq13valENQxO55dPOnHZ/Q5OWmE9Wm3obCLeAX
  0ycgoQOylaZrgnyyU/1BITr03xTqdKoKWkcWBTArGL/S2t5r3PAB
VDE.
RET.

'------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_4134
MBX.Pattern <P_4134.bmp> not found.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------

'------------------------------------------------------------
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'------------------------------------------------------------
:Locate_right_Icon
STW.c|TDXRibbonDrop
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_0736.bmp
GSB.Inline_5082
VAR.$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:64 . We use -10%, P5 - Debugmode
0/1
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|57
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|0,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_736
MLC.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  256 Bytes in  2 Lines.
:Inline_5082
VDA.$$DTA|$$FIN=
  ?
MAAEOAQAASlE0fiPYojaIkFksofsrHLzSYwEMI2DupSxPFpRGoxlW/K6vBCiYs
9/tb4VBDoVshD+ORQ+63LN/cjN2pi28Slu3LJr3SntQodlcgW6EcAG21dfb5YW
wKSUfsivsUMtUGE9LXCCZeTIQts1fPhE5MmdA
  LZqQozOATKpVnv6Rdo1zCAgVVXL4c3fztfeJI4ldaD1pZD8Sum4EZBIGi/PE
5nUtlEMoo7A8TGFh5P9ragZ/6nfOdwE=
VDE.
RET.
'------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_736
MBX.Pattern <P_0736.bmp> not found.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------

:Locate_Small_MRL
GSB.Locate_TW
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_2226.bmp
GSB.Inline_8713
VAR.$$PAT=ID:$$DTA
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:192 . We use -10%, P5 - Debugmode
0/1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|172
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|60,0|60
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_2226
MLC.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  588 Bytes in  4 Lines.
:Inline_8713
VDA.$$DTA|$$FIN=
  ?
MAAEOAwAASlFsSBrYfuxres3rZKAVC4IPFDcuS3dlJLnnqwk0AKNcmzYvEneCN
F75m7uuvzUiv+FGDqAZK1M3oPx+8MkHWIUBikKVrzeyBxQ3rpGibARJRG45bop
EUFSJGnqLG5O62zrXB3o9HHCvfLdGs51Y3vFL
  xqPD74Zi2QwLbWzVsBKIoS58imR/8vdLCQjPgsGz4m+56olo/Sjz9BEGRtsk
GbEUqcQ07wJl5RD5lHeEwCvw5JKEgIK+jM38GxbFuMh5sAMYO0NISQOr7VbIzd
BkMCZsVMBiFtJiQU9FP4CSvL2iUysrmDhxCjYn
  2l4bPZIOyhOn6MqKdDqWKJnA6wQ9hqF2xsdkMlryz26px52eWjy2h2MSEL6b
/76fDuPCJYPUXeYcuvv8tbZaHZtsUYC4ILiN5r4IBSLRSAuMZU1vrLslpSCQuf
THTOStYcdBG2gFSD+CArBYAyWeFC3fgFjMi2Oy
  DGF4vN0vKdwYFIxIVjXtlnbr31mt5dZPVaddmnfLz0gpGTpISyim3lycfiyk
eeLBWjJXfKSr8D+S+k6U+XE1F5RH/ib2V4oMGAE=
VDE.
RET.
'------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_2226
MBX.Pattern <P_2226.bmp> not found.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------



'------------------------------------------------------------
'
'------------------------------------------------------------
:Locate_Topline_Start
GSB.Locate_TW
' Please adapt the filename of the saved Pattern
'VAR.$$PAT=?pattern\P_5564.bmp
GSB.Inline_4
VAR.$$PAT=ID:$$DTA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' P1 - Target, P2 - Pattern, P3 - Color-Tolerance, P4 - min.
Hits (Pixel) maximum Number:64 . We use -10%, P5 - Debugmode
0/1
'SFP.Snap_Border|1
SFP.ColorTolerance|0
SFP.PixelHits|57
SFP.DebugMode|0
' Patternfile | Point-Shift| Timeout 60 Seconds
WFP.$$PAT|30,0|5
' We take away the number of Hits from TOS
POP.$$NOH
' If no Hits, jump away to Lab_NoHits.
JIZ.$$NOH=0|Lab_NoHits_5564
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
' .bmp from HE-Recorder
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  138 Bytes in  1 Lines.
:Inline_4
VDA.$$DTA|$$FIN=
  ?
MAAEOAQAASlGseyTYXhw1dWgkPbIMMW3DbAOw1aeeO20b4CJPvTOzqXjsFcet5
AHMJ3jOQgIwQ7jveo50iUkpmUGtWyAjHD0Tj/F1to6YforqXvDW6Mft/JFFLnL
+ZPV3xSd0AQ==
VDE.
RET.
'------------------------------------------------------------
' The following Part will be called in case of Error.
:Lab_NoHits_5564
MBX.Pattern Topline-Start not found.
RET.
'------------------------------------------------------------
'
'------------------------------------------------------------
'
'------------------------------------------------------------
ENR.
```

Remarks

-

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702

© 2023 Theo Gottwald * https://www.smart-package.com

- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.3 ! PIS. - PlugIn Send

MiniRobotLanguage (MRL)

PIS. Command

PlugInSend

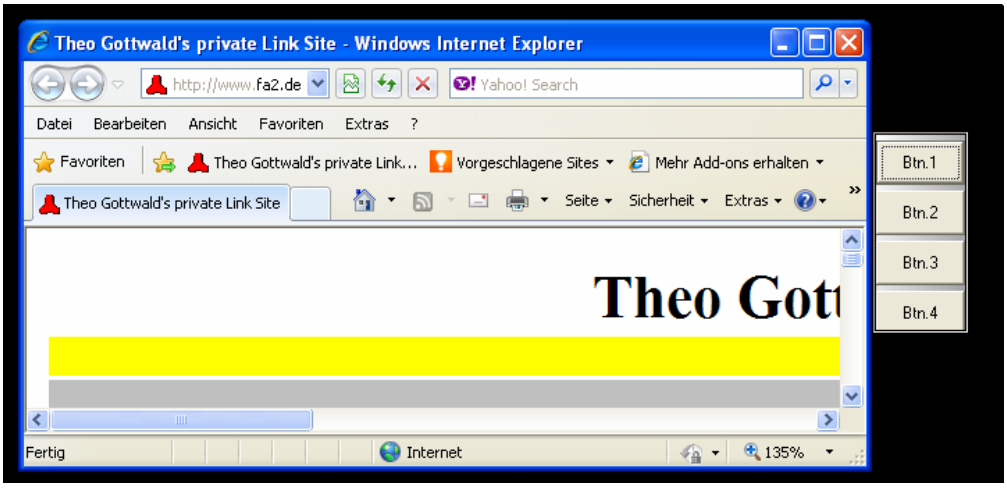
Intention

The PIS-Command sends Commands to a PlugIn.

Using `PIS.` you can

- change the Text on the Buttons,
- attach a picture to the buttons,
- resize the Plugin or
- attach the PlugIn to any window
- and much more.

You can also decide whether the PlugIn Buttons should be arranged vertical or horizontal.



This is a vertical PlugIn 01. It will be arranged vertical, using:

```
' Plugin vertical
PIS.$$REA|sv
```

And in the picture below you see a horizontal PlugIn.
The code for a horinzontal PlugIn is:

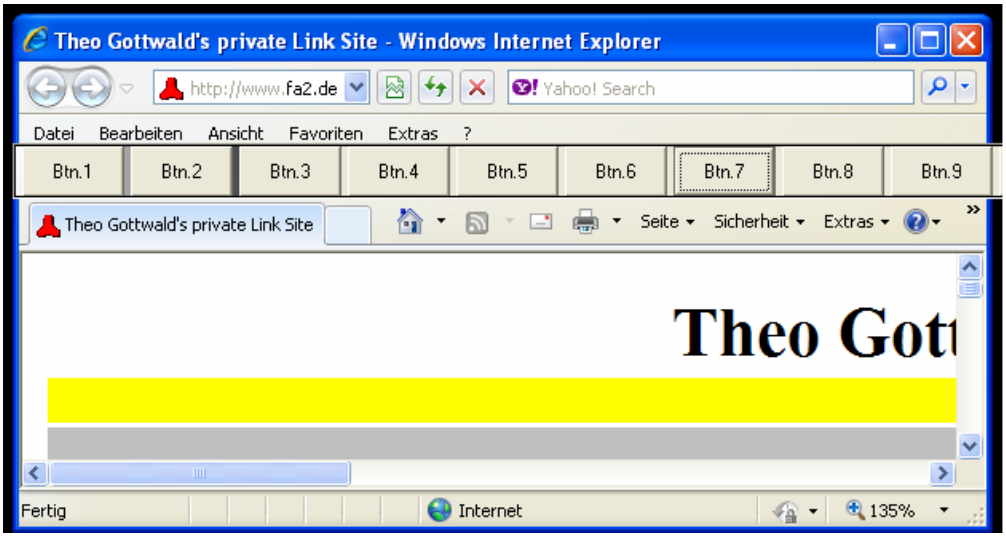
```
' Plugin horizontal
PIS.$$REA|sh
```

Important note:

Select the arrangement (horizontal/vertical) immediately after the PlugIn Launch.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Then add all needed Buttons. If you need to change the arrangement later, reduce the button-number to 1 and then after changing, add additional buttons.



Syntax

PIS.P1|P2[|P3...Px]

Parameter Explanation

P1 - variable which contains the PlugIn Index-No.

P2 - Command which shall be sent to the PlugIn

All commands are specific for the type of PlugIn used.
Following PlugInCommands are supported by the actual
PlugIn_01.exe:

```
"at","attach"
' $$I06 attach the Plugin to the window,
' 2 - right side, 1 - on Top
' §§I05
' 1 - Left side outside; 2 - Right side outside;
' 3 - left side inside, 4 - right side inside;
' 5  - left side inside bündig, 6 - right side inside bündig
'
PIS.$$PIN|attach|$$ULT|$$I05|$$I06
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"dt","detach"
PIS.$$PIN|detach

"sbt","setbuttontext"
PIS.$$PIN|sbt|$$NUM|$$TXT

"gh","gethandle" ' Handle is returned on TOS
PIS.$$PIN|gh

"sp","setpicture"
PIS.$$PIN|sp|1|24_about.bmp
PIS.$$PIN|sp|2|24_about.ico
PIS.$$PIN|sp|19|24_about.cur
' This will set all pictures to
' "24_about_1.cur" til "24_about_20.cur"
PIS.$$PIN|sp|all|24_about_%n.cur

"sbhvs","setbuttonshvsize"
PIS.$$PIN|sbhvs|$$HSI|$$VSI

"sbvs","setbuttonsvsize"
PIS.$$PIN|sbvs|$$VSI

"sbvd","setbuttonsvdist"
PIS.$$PIN|sbvd|$$VDI

"sbhs","setbuttonshsize"
PIS.$$PIN|sbhs|$$HSI

"sbhd","setbuttonshdist"
PIS.$$PIN|sbvd|$$VDI

"vis","show"
PIS.$$PIN|vis

"inv","hide"
PIS.$$PIN|hide

"sh","sethorizontal"
PIS.$$PIN|sh

"sv","setvertikal"
PIS.$$PIN|sv

"scf","setcorrectionfactor"
PIS.$$PIN|scf|$$I02|0|$$I04

"sbn","setbuttonnumber"
PIS.$$PIN|sbn|6

"sbfc","setbuttonforecolor"
PIS.$$PIN|sbfc|1|&hff00ff

"sbbc","setbuttonbackcolor"
PIS.$$PIN|sbbc|1|&h0ff00
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
"sbtr","setbuttonstransparent"
PIS.$$PIN|sbtr|ALL|1

"bd","buttondisable"
PIS.$$PIN|bd|all

"be","buttonenable"
PIS.$$PIN|be|2

' Parameters can be 1 or 0
' the PlugIn will be on Top or not
"stm","switchtopmost"
PIS.$$PIN|stm|1

"sas","setautorunsynchron"
PIS.$$PIN|sas|?pfad\myprog.exe

"saa","setautorunasynchron"
PIS.$$PIN|saa||?pfad\myprog.exe

"sasnd","setautorunsynchronnodisable"
PIS.$$PIN|sasnd|?pfad\myprog.exe

"saand","setautorunasynchronnodisable"
PIS.$$PIN|saand|?pfad\myprog.exe

"mm","mousemesage"
PIS.$$PIN|all|1
 0 - alle Messages,
 1 - Message nicht bei Autorun,
 2 - keine dblclick,
 4 - keine singleklick,
 8 - keine mouseover,
16 - keine Mouseleftmessages

"stm","switchtopmost"
PIS.$$PIN|stm|1
' Can be 0 or 1 to select between Top- and TOPMOST

"swa","setwindowalert"
PIS.$$PIN|swa|1|ieframe
' Can be 0 or 1 to enable or disable the Alerting-Message when
another window with the given classname appears in the
foreground.

"sha","setautohide"
PIS.$$PIN|sah|1
' Can be 0 or 1 to select between hide or not-hide PlugIn
Buttons automatically when the Host-window dissapears.
Autohide is ON by default.
```

P3 **-** Parameters will be sent to the Plugin, they depend on the command **P1**.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*********************************
' Vertical Plugin-Sample
'*********************************
'VAR.$$PIC=24_about.bmp
EXO.iexplore
PIL.$$REA|01
' use any window to attach the plugin
STW.t|explorer
' Store the window handle in a variable
HTV.$$ULT

PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show

' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|0|0

' Set text on Button 1
FOR.$$CNT|1|20
  PIS.$$REA|sBt|$$CNT|Btn.$$CNT
NEX.

' Show a picture on Button 2
'PIS.$$REA|sp|2|$$PIC

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|sbn|1

' This double Loop will Show more or less buttons
FOR.$$LOA|1|2
  FOR.$$LOP|1|20
    PRT.->$$LOP
    PIS.$$REA|sbn|$$LOP
    PAU..2
  NEX.
NEX.

PAU.2
:enz
END.
```

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.

© 2023 Theo Gottwald * https://www.smart-package.com

To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
```
PIS.$$PIN|sbt|all|Button %n
```
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-
-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.4 PIS.attach/detach

MiniRobotLanguage (MRL)

PIS.attach/detach Command

PlugInSend attach/detach

Intention

The `PIS.attach`-Command will attach a Buttonbar to a Target Window.
The `PIS.detach`-Command will detach a Buttonbar from the Target Window. After that the Buttonbar can be moved around.

"at","attach"
```
PIS.$$PIN|attach|$$ULT|0|0
```

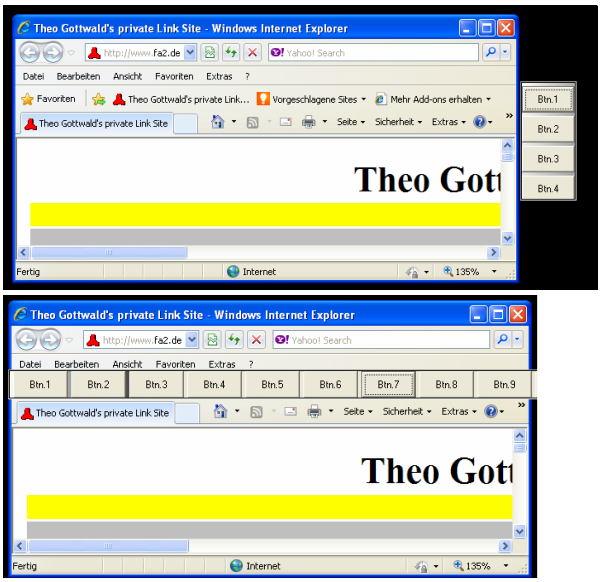
"dt","detach"
```
PIS.$$PIN|detach
```

Using the "attach/detach" Subcommand, you can define whether and where a Plugin is attached to an existing window.
To clearly specify the Position of the Plugin, you need to decide whether the PlugIn Buttons should be arranged vertical or horizontal.
This is the first step. For this you use the "sv" or the "sh" Subcommand.

This is a vertical PlugIn 01. It will be arranged vertical, using:
```
' Plugin vertical
PIS.$$REA|sv
```

And in the picture below you see a horizontal PlugIn.
The code for a horinzontal PlugIn is:
```
' Plugin horizontal
PIS.$$REA|sh
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

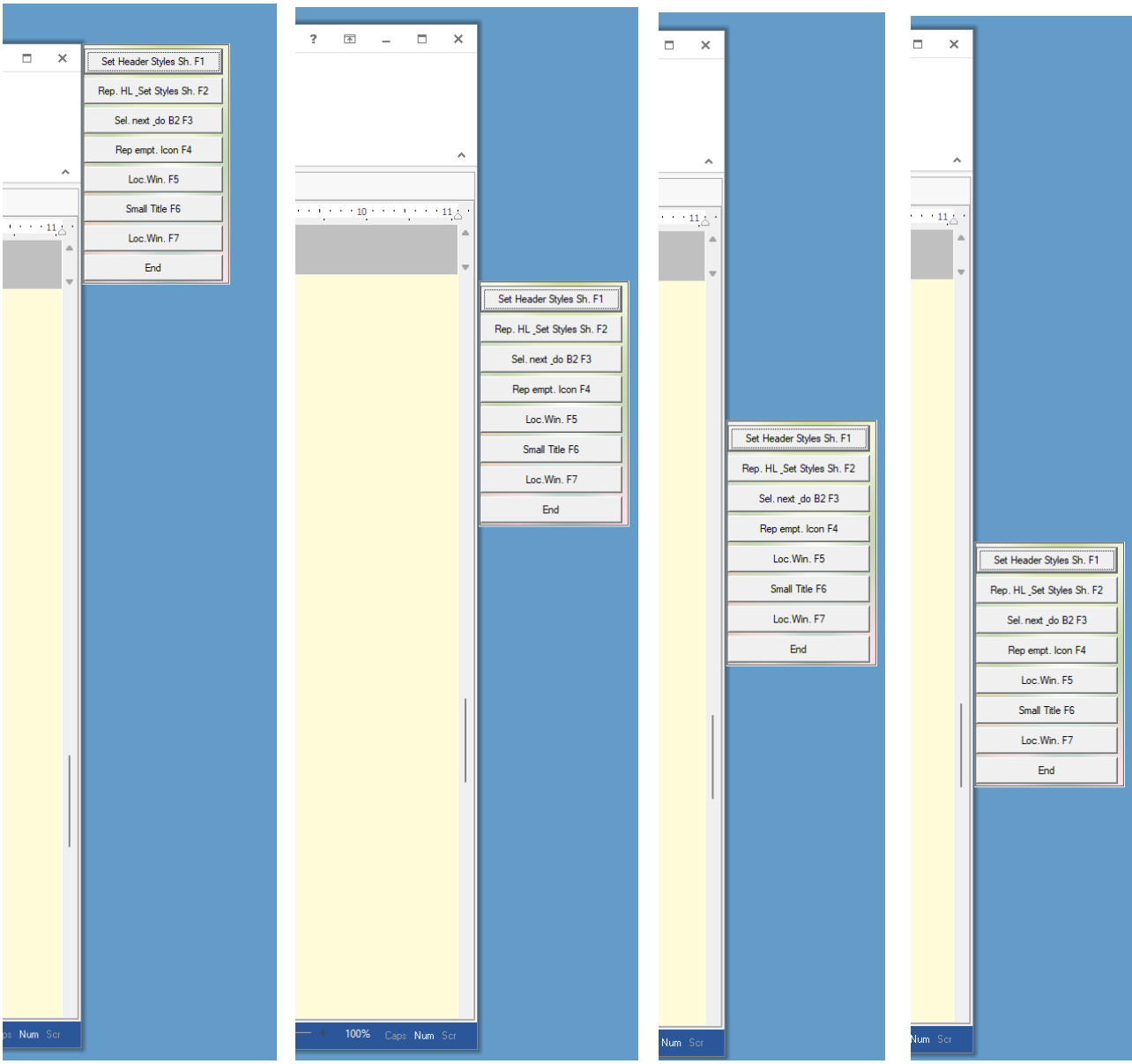


Here you can see a vertical and horizontal Plugin-Buttonbar. Below you will find more Samples.

The following are Samples using a vertical Buttonbar:

The vertical Positions are calculated depending on the PlugIn-Size, and the Target-Window Size.

*© 2023 Theo Gottwald * https://www.smart-package.com*



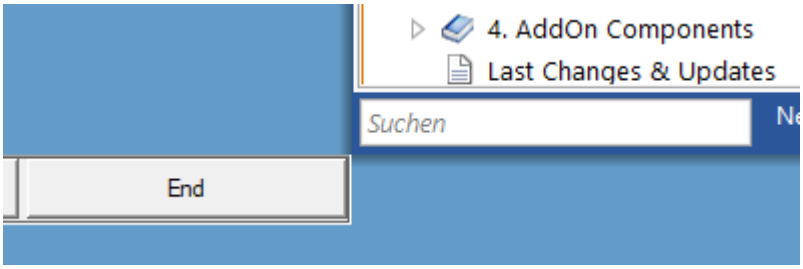
X= 1 2 3 4

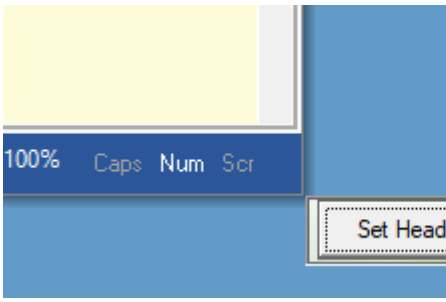
```
PIS.$$PLG|sv
PIS.$$PLG|attach|$$HWN|2|X
```

The following are Samples using a horizontal Buttonbar:

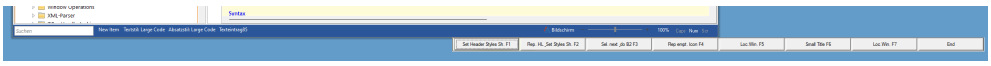
```
PIS.$$PLG|sh
PIS.$$PLG|attach|$$HWN|X|6
below you find the Values for "X"
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



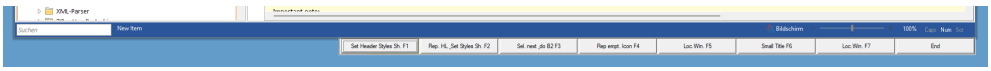


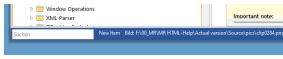
X= 1 2



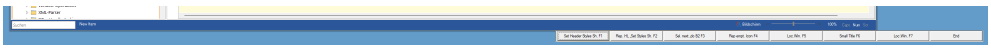
X= 4

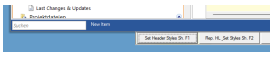
X= 5



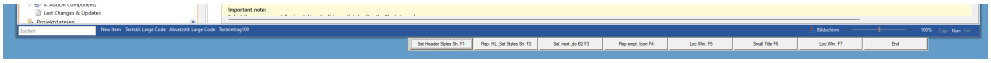


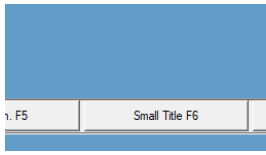
X= 6



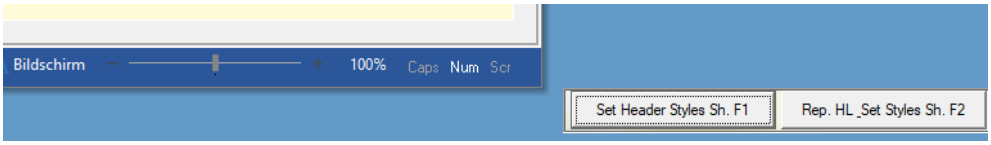


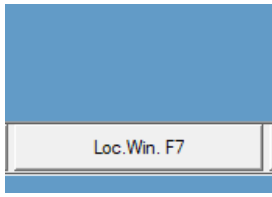
X= 8



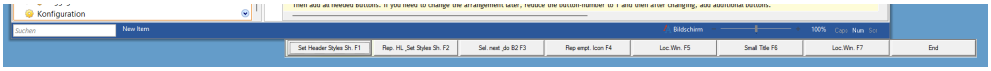


X= 10



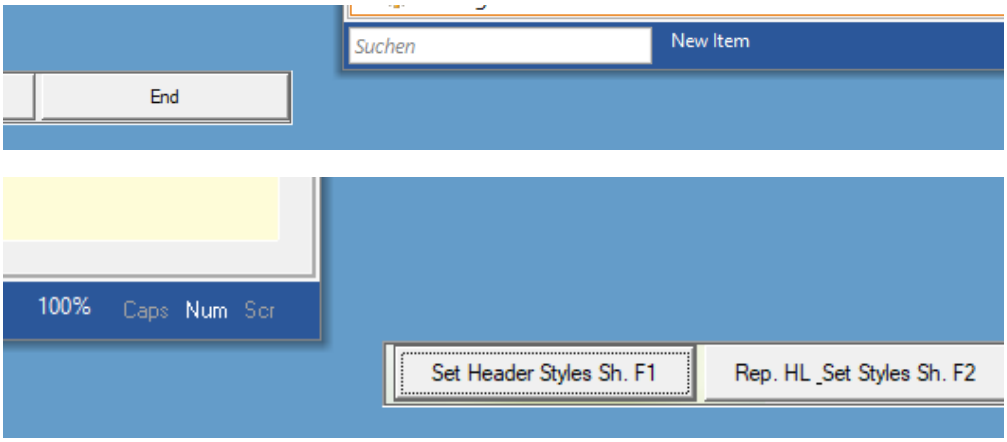


X= 12 -> 16 Pixel Distance to the right Border



X= 14 (right Anchor)

© 2023 Theo Gottwald * https://www.smart-package.com



X= 15 -> 32 Pixel Distance to the right Border

Important note:
Select the arrangement (horizontal/vertical) immediately after the PlugIn Launch. Then add all needed Buttons. If you need to change the arrangement later, reduce the button-number to 1 and then after changing, add additional buttons.

Syntax

PIS.P1|attach|P2|P3|P4
PIS.P1|at|P2|P3|P4

PIS.P1|detach
PIS.P1|dt

Parameter Explanation

P1 - Variable which contains the PlugIn Index-No.
P2 - Window-Handle of Target Window
P3 - Position of Buttonbar relative to Target-Window
1 - Left Side Outside the Window
2 - Right Side Outside the Window (default value, if omitted or zero)
3 - Left Side Inside the Window with Distance
4 - Right Side Inside the Window with Distance
5 - Left Side Inside the Window
6 - Right Side Inside the Window
... 16 (see above)

P4 - X/Y-Position of Buttonbar relative to Side

© 2023 Theo Gottwald * https://www.smart-package.com

```
1 - Top-Side, Inside
2 - Top Quarter
3 - Middle
4 - Lower Middle
5 - Down-Side Inside
6 - Down-Side Outside
7 - Top-Side Outside
... 15  (see above)
```

Some of the values set The anchor on the left side of the Target Window, some set the Anchor on the right side.

Example

```
'*********************************
' Vertical Plugin-Sample
'*********************************
'VAR.$$PIC=24_about.bmp
EXO.iexplore
PIL.$$REA|01
' use any window to attach the plugin
STW.t|explorer
' Store the window handle in a variable
HTV.$$ULT

PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show

' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|0|0

' Set text on Button 1
FOR.$$CNT|1|20
  PIS.$$REA|sBt|$$CNT|Btn.$$CNT
NEX.

' Show a picture on Button 2
'PIS.$$REA|sp|2|$$PIC

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|sbn|1

' This double Loop will Show more or less buttons
FOR.$$LOA|1|2
  FOR.$$LOP|1|20
    PRT.->$$LOP
    PIS.$$REA|sbn|$$LOP
    PAU..2
  NEX.
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU.2
:enz
END.
```

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Technically the Anchoring Points are defined like this:

```
x,y - size of Plugin-Buttonabar
nleft means the Anchor is on the left side of the Target Window, nRight mean

CASE 1:xt=r2.nleft-x ' links ausserhalb des Fensters
CASE 2:xt=r2.nright+1 ' rechts ausserhalb des fenster
CASE 3:xt=r2.nleft+(x/2) ' links mitte
CASE 4:xt=r2.nright-(x/2) ' rechts mitte
CASE 5:xt=r2.nleft ' links innerhalb
CASE 6:xt=r2.nright-x ' rechts innerhalb
CASE 7:xt=r2.nright-(x/4) ' rechts mitte
CASE 8:xt=r2.nright-((x/4)*3) ' rechts mitte
CASE 9:xt=r2.nleft+(x/4) ' rechts mitte
CASE 10:xt=r2.nleft+((x/4)*3) ' rechts mitte
CASE 7:xt=r2.nright+(x/4) ' rechts mitte
CASE 8:xt=r2.nright+((x/4)*3) ' rechts mitte
CASE 9:xt=r2.nleft-(x/4) ' rechts mitte
CASE 10:xt=r2.nleft-((x/4)*3) ' rechts mitte
CASE 11:xt=r2.nleft-((x/8)*7)
CASE 12:xt=r2.nright+16
CASE 13:xt=r2.nleft-x-16
CASE 14:xt=r2.nright-((x/8)*7)
CASE 15:xt=r2.nleft-x-32
CASE 16:xt=r2.nright+32

CASE 1:yt=r2.ntop ' Fenster oben anlegen, innerhalb
CASE 2:yt=r2.ntop+(T08/4) ' oberes viertel
CASE 3:yt=r2.ntop+(T08/2)-(y/2) ' mitte anlegen
CASE 4:yt=r2.ntop+(T08/2) ' mitte unten anlegen
CASE 5:yt=r2.nbottom-y ' unten ausserhalb
CASE 6:yt=r2.nbottom ' unten ausserhalb anlegen
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CASE 7:yt=r2.ntop-(y+1) ' Fenster oben anlegen, ausserhalb
CASE 8:yt=r2.ntop+(T08/4)-(y/2) ' oberes viertel minus Pluginsize
CASE 9:yt=r2.ntop+(T08/8) ' oberes achtel
CASE 10:yt=r2.ntop+(T08/8)-(y/2) ' oberes achtel-Plugin-Size
CASE 11:yt=r2.ntop+(T08/8)+(y/2) ' oberes achtel+Plugin-Size
CASE 12:yt=r2.ntop+(T08/4)-(y/2) ' oberes viertel minus Pluginsize
CASE 13:yt=r2.nbottom-(T08/8)-y ' oberes achtel
CASE 14:yt=r2.nbottom-(T08/8)-(y*1.5) ' oberes achtel-Plugin-Size
CASE 15:yt=r2.nbottom-(T08/8)-y ' oberes achtel+Plugin-Size
```

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.5 PIS.ButtonEnable / ButtonDisable

PIS.ButtonEnable / ButtonDisable - be/bd Previous 2702 Top 41 Next 2706

MiniRobotLanguage (MRL)

PIS.ButtonEnable Command
PIS.ButtonDisable Command

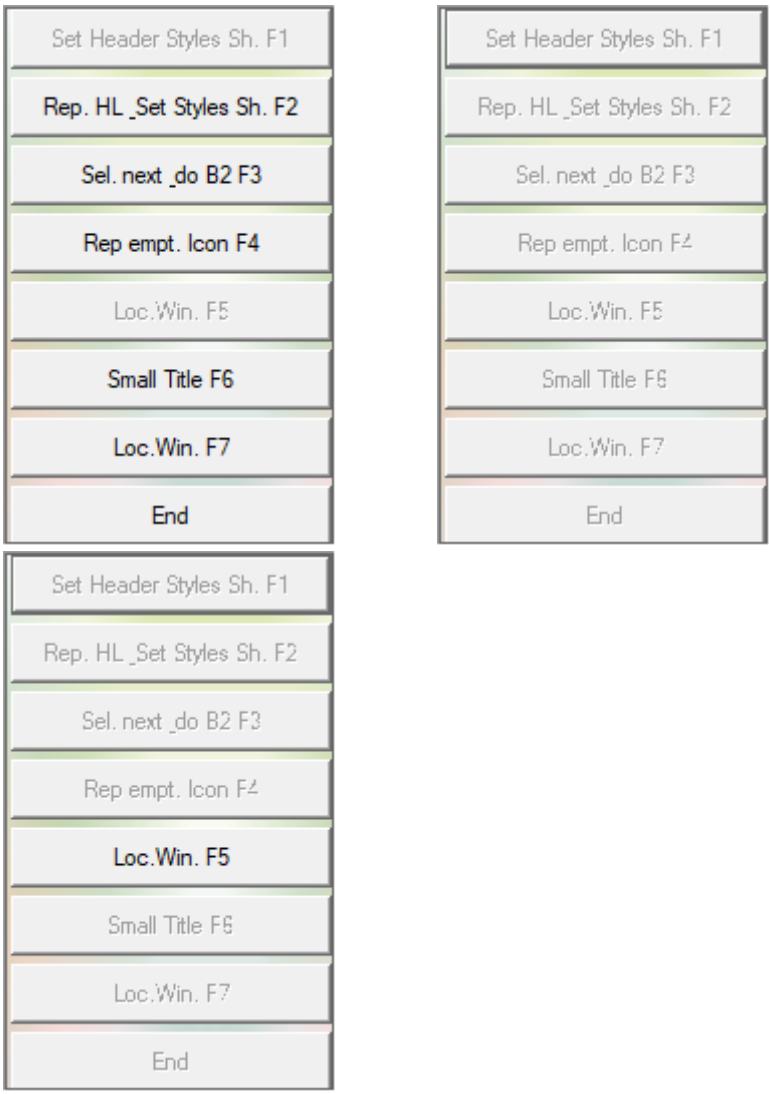
Enable or disable Buttons of the specified Buttonbar

Intention

The `PIS.ButtonEnable / ButtonDisable` -Command will enable or disable the specified button.
If you specify "all" instead of a Button-Number, then all Buttons will be enabled or disabled.

```
PIS.$$REA|ButtonDisable|1
PIS.$$REA|ButtonDisable|5
PIS.$$REA|ButtonDisable|all
PIS.$$REA|ButtonEnable|5
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Buttonbar with enabled, and disabled Buttons.

Syntax

```
PIS.P1|ButtonEnable|P2
PIS.P1|be|P2

PIS.P1|ButtonDisable|P2
PIS.P1|bd|P2
```

Parameter Explanation

P1 - Variable which contains the *PlugIn Index-No.*

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **-** value (0-255)- the transparency Values range from 0 to 255,
where 0 is completely transparent (invisible) and 255 is solid (not transparent).

All commands are specific for the type of PlugIn used.

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.6 PIS.GetHandle

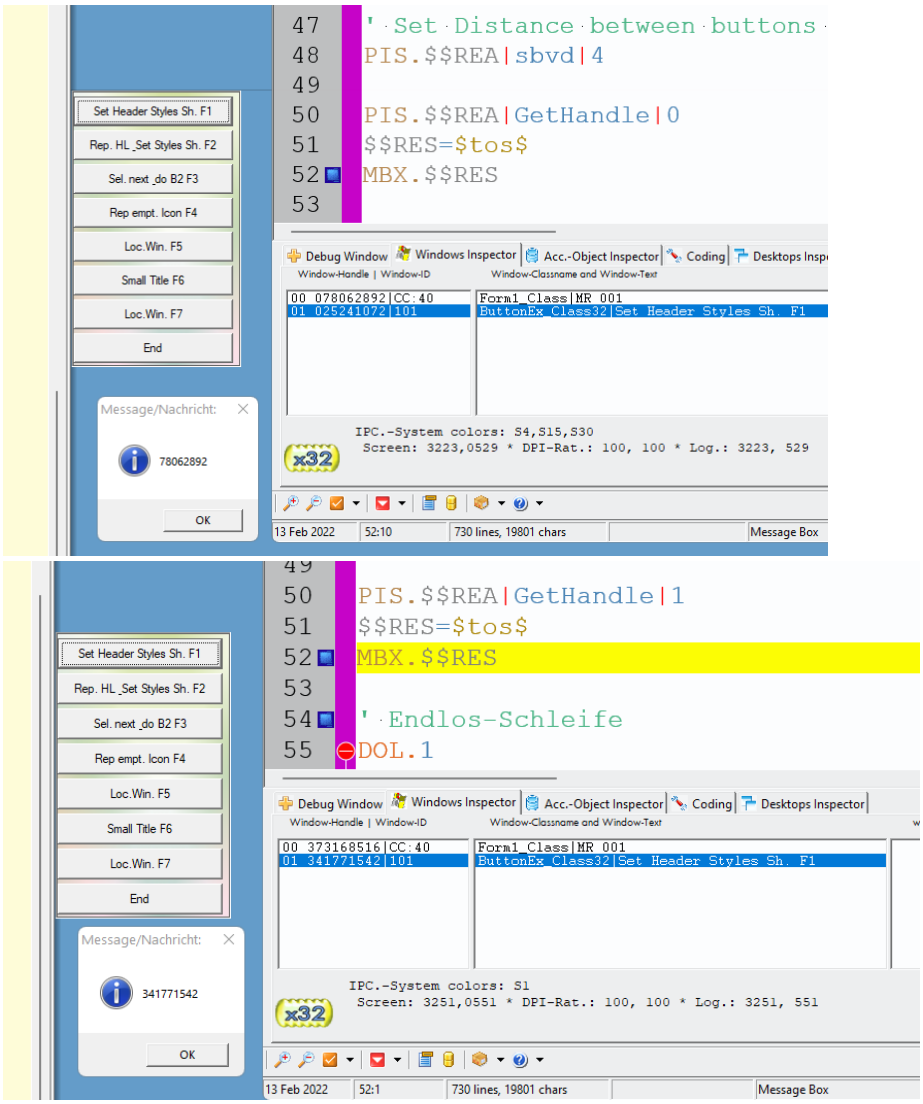
MiniRobotLanguage (MRL)

PIS.GetHandle Command

Get the Window-Handle from the Buttonbar

Intention

The `PIS.GetHandle` -Command will return the Window-Handle from the specified Buttonbar.
Using **P2** you can specify a number from 1 to (number of buttons) to get the Windowhandle from that Button.
The result is always placed on TOS.



Get Windowhandle from Buttonbar (Tophandle)
Get Windowhandle from Buttonbar (Handle from Button 1)

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

PIS.P1|GetHandle|P2
PIS.P1|gH|P2

Parameter Explanation

P1 - `Variable` which contains the *`PlugIn Index-No.`*

P2 - *`Button Number`*, starting with "1" to get the Button-Windowhandle, if omitted or 0, the returned handle is from the Top-Window of the Buttonbar

The result is always placed on TOS.

Example

```
'*********************************
' Vertical Plugin-Sample
'*********************************
'
'SPR Script-file: New Script_1
'Purpose:
'Author:
'Creation date: 02-08-2022 at 07:53:12
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
' This will run the NoLimits2 in case its not yet running
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=8
PIS.$$REA|sbn|$$BUN

' use any window to attach the plugin
PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|2|2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|GetHandle|1
$$RES=$tos$
MBX.The handle is $$RES

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'-------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.

SAV.Restore
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'------------------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.$$BUN
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAN.4
    JSR.doButton04
  CAN.5
    JSR.doButton05
  CAN.6
    JSR.doButton06
  CAN.7
    JSR.doButton07
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'------------------------------------------------------------
' Do Button 1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:doButton01
GSB.Set_Style_Headline
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.1
MLM.
GSB.Move_Mouse_To_B1
RET.
'-------------------------------------------------------
' Do Button 2
'-------------------------------------------------------
:doButton02
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.0.25
MLM.
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------
' Do Button 3
'-------------------------------------------------------
' Jump to next and then Replace Headers
:doButton03
GSB.Locate_below_Selected
PAU.1
MLM.
PAU.1
WII.6
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.Move_Mouse_To_Button|3
RET.
'-------------------------------------------------------
' Do Button 4
'-------------------------------------------------------
:doButton04
GSB.Locate_false_Icon
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Locate_Icon_change_Menu
GSB.Locate_right_Icon
GSB.Move_Mouse_To_Button|4
RET.
'-------------------------------------------------------------
' Do Button 5
'-------------------------------------------------------------
:doButton05
FOR.$$LL5|1|8
  GSB.doButton03
  PAU.1
  IEK.LSHIFT
    ENR.
  EIF.
NEX.
GSB.Move_Mouse_To_Button|5
RET.
'-------------------------------------------------------------
' Do Button 6
'-------------------------------------------------------------
:doButton06
GSB.Do_Mark_MRL
GSB.Set_Style_Small_Page_Title
GSB.Locate_Topline_Start
MLC.
GSB.Move_Mouse_To_Button|6
RET.
'-------------------------------------------------------------
' Do Button 7
'-------------------------------------------------------------
:doButton07
GSB.Locate_TW
GSB.Move_Mouse_To_Button|7
RET.
'-------------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
'============================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'------------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'------------------------------------------------------------
:Locate_TW
IEW.$$ULT
  STW.h|$$ULT
ELS.
  STW.ct|TFrmHelpMan|Help+Manual
  HTV.$$ULT
  SWP.1090,130|1414,1665|T
EIF.
RET.
'------------------------------------------------------------
:Open_Style_Menu
GSB.Locate_TW
'SWP.1666,287|297,93|T
'MMV.t|428,125
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControlSite|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControl|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupBarControl|Schriftart
MLI.w|117,61
WII.
PAU.0.1
RET.
'------------------------------------------------------------
:Set_Font_Style
$$STL=§§_01
STW.Ac|TdxBarDropdown
AGR.34!WII.MOS.|$$STL||1|3|60
RET.
'------------------------------------------------------------
:Do_Mark_MRL
GSB.Locate_Small_MRL
SKP.{END}
KYD.{LSHIFT}
SKP.{HOME}
KYU.{LSHIFT}
PAU.1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'-------------------------------------------------------------
:Do_Mark_Line
GSB.Locate_TW
SKP.{HOME}
KYD.{LSHIFT}
SKP.{END}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------

'-------------------------------------------------------------
:Locate_Headline
GSB.Locate_TW
MLI.t|584,249
RET.
'-------------------------------------------------------------
:Locate_Sub_Headline
GSB.Locate_TW
MLI.t|835,300
RET.
'-------------------------------------------------------------
:Locate_right_Sub_Headline
GSB.Locate_TW
MLC.t|1126,298
RET.
'-------------------------------------------------------------
:Mark_Headline
GSB.Locate_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_right_Sub_Headline
GSB.Locate_right_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Full_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Set_Style_Line_Style
GSB.Open_Style_Menu
GSB.Set_Font_Style|LineStile
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
:Set_Style_Small_Page_Title
GSB.Open_Style_Menu
GSB.Set_Font_Style|Small Page Title
RET.
'-------------------------------------------------------------
:Set_Style_Headline
GSB.Mark_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
RET.
'-------------------------------------------------------------
:Set_Style_Sub_Headline
GSB.Mark_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading2
RET.
'-------------------------------------------------------------
:Set_Style_right_Sub_Headline
GSB.Mark_right_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading3
RET.
'-------------------------------------------------------------
:Move_Mouse_To_Button
$$STL=§§_01
VBT.$$STL
$$TXT=F$$STL
STW.ct|Form1_Class|MR
SCW.ct|ButtonEx_Class32|$$TXT
MPO.
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B1
GSB.Move_Mouse_To_Button|1
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B2
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Locate_selected_TV_Element
' Code removed
RET.
'-------------------------------------------------------------
:Locate_below_Selected
' Code removed
RET.
'-------------------------------------------------------------
:Locate_false_Icon
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Icon_change_Menu
' Code removed
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'------------------------------------------------------------
:Locate_right_Icon
' Coder removed
RET.
'------------------------------------------------------------
:Locate_Small_MRL
' Code removed
RET.
'------------------------------------------------------------
:Locate_Topline_Start
' Code removed
RET.
'------------------------------------------------------------
ENR.
```

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706

© 2023 Theo Gottwald * https://www.smart-package.com

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.7 PIS.GetMouseButton

MiniRobotLanguage (MRL)

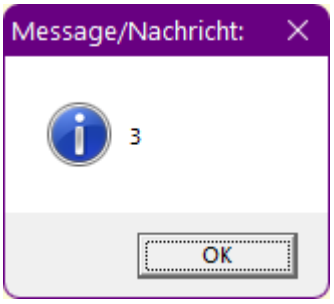
PIS.GetMouseButton Command

Get the Number of the Button where the Mouse is (Mouseover)

Intention

The `PIS.GetHandle` -Command will return the Window-Handle from the specified Buttonbar or the Button where the Mouse is currently over.
It can be used to react to Mouse-Over Commands.
If the Mouse is not over a Buttonbar-Button, the result is "0" The result is always placed on TOS.

```
PIS.$$REA|GetMouseButton
$$RES=$TOS$
```



Syntax

```
PIS.P1|GetMouseButton
PIS.P1|gmb
```

Parameter Explanation

P1 – `Variable` which contains the *`PlugIn Index-No.`*

Will return the number of the Button over which the Mouse is currently.
If the Mouse is not over a Buttonbar-Button, the result is "0" The result is always placed on TOS.

Example

```
'*********************************
' Vertical Plugin-Sample
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
'SPR Script-file: New Script_1
'Purpose:
'Author:
'Creation date: 02-08-2022 at 07:53:12
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
' This will run the NoLimits2 in case its not yet running
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=8
PIS.$$REA|sbn|$$BUN

' use any window to attach the plugin
PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|2|2

$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|GetHandle|1
$$RES=$tos$
MBX.The handle is $$RES

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'-------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.

SAV.Restore
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'-------------------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.$$BUN
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAN.4
    JSR.doButton04
  CAN.5
    JSR.doButton05
  CAN.6
    JSR.doButton06
  CAN.7
    JSR.doButton07
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'-------------------------------------------------------------
' Do Button 1
'-------------------------------------------------------------
:doButton01
GSB.Set_Style_Headline
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.1
MLM.
GSB.Move_Mouse_To_B1
RET.
'-------------------------------------------------------------
' Do Button 2
'-------------------------------------------------------------
:doButton02
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.0.25
MLM.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
' Do Button 3
'-------------------------------------------------------------
' Jump to next and then Replace Headers
:doButton03
GSB.Locate_below_Selected
PAU.1
MLM.
PAU.1
WII.6
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.Move_Mouse_To_Button|3
RET.
'-------------------------------------------------------------
' Do Button 4
'-------------------------------------------------------------
:doButton04
GSB.Locate_false_Icon
GSB.Locate_Icon_change_Menu
GSB.Locate_right_Icon
GSB.Move_Mouse_To_Button|4
RET.
'-------------------------------------------------------------
' Do Button 5
'-------------------------------------------------------------
:doButton05
FOR.$$LL5|1|8
  GSB.doButton03
  PAU.1
  IEK.LSHIFT
    ENR.
  EIF.
NEX.
GSB.Move_Mouse_To_Button|5
RET.
'-------------------------------------------------------------
' Do Button 6
'-------------------------------------------------------------
:doButton06
GSB.Do_Mark_MRL
GSB.Set_Style_Small_Page_Title
GSB.Locate_Topline_Start
MLC.
GSB.Move_Mouse_To_Button|6
RET.
'-------------------------------------------------------------
' Do Button 7
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:doButton07
GSB.Locate_TW
GSB.Move_Mouse_To_Button|7
RET.
'-------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
ENR.
'=======================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'-------------------------------------------------------
:Locate_TW
IEW.$$ULT
  STW.h|$$ULT
ELS.
  STW.ct|TFrmHelpMan|Help+Manual
  HTV.$$ULT
  SWP.1090,130|1414,1665|T
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'-------------------------------------------------------------
:Open_Style_Menu
GSB.Locate_TW
'SWP.1666,287|297,93|T
'MMV.t|428,125
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControlSite|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControl|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupBarControl|Schriftart
MLI.w|117,61
WII.
PAU.0.1
RET.
'-------------------------------------------------------------
:Set_Font_Style
$$STL=§§_01
STW.Ac|TdxBarDropdown
AGR.34!WII.MOS.|$$STL||1|3|60
RET.
'-------------------------------------------------------------
:Do_Mark_MRL
GSB.Locate_Small_MRL
SKP.{END}
KYD.{LSHIFT}
SKP.{HOME}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------
:Do_Mark_Line
GSB.Locate_TW
SKP.{HOME}
KYD.{LSHIFT}
SKP.{END}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------

'-------------------------------------------------------------
:Locate_Headline
GSB.Locate_TW
MLI.t|584,249
RET.
'-------------------------------------------------------------
:Locate_Sub_Headline
GSB.Locate_TW
MLI.t|835,300
RET.
'-------------------------------------------------------------
:Locate_right_Sub_Headline
GSB.Locate_TW
MLC.t|1126,298
RET.
'-------------------------------------------------------------
:Mark_Headline
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Locate_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_right_Sub_Headline
GSB.Locate_right_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Full_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Set_Style_Line_Style
GSB.Open_Style_Menu
GSB.Set_Font_Style|LineStile
RET.
'-------------------------------------------------------------
:Set_Style_Small_Page_Title
GSB.Open_Style_Menu
GSB.Set_Font_Style|Small Page Title
RET.
'-------------------------------------------------------------
:Set_Style_Headline
GSB.Mark_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
RET.
'-------------------------------------------------------------
:Set_Style_Sub_Headline
GSB.Mark_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading2
RET.
'-------------------------------------------------------------
:Set_Style_right_Sub_Headline
GSB.Mark_right_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading3
RET.
'-------------------------------------------------------------
:Move_Mouse_To_Button
$$STL=§§_01
VBT.$$STL
$$TXT=F$$STL
STW.ct|Form1_Class|MR
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SCW.ct|ButtonEx_Class32|$$TXT
MPO.
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B1
GSB.Move_Mouse_To_Button|1
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B2
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Locate_selected_TV_Element
' Code removed
RET.
'-------------------------------------------------------------
:Locate_below_Selected
' Code removed
RET.
'-------------------------------------------------------------
:Locate_false_Icon
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Icon_change_Menu
' Code removed
RET.
'-------------------------------------------------------------
:Locate_right_Icon
' Coder removed
RET.
'-------------------------------------------------------------
:Locate_Small_MRL
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Topline_Start
' Code removed
RET.
'-------------------------------------------------------------
ENR.
```

Remarks

*© 2023 Theo Gottwald * https://www.smart-package.com*

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
```
PIS.$$PIN|sbt|all|Button %n
```
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-
-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.8 PIS.GetMouseButtonHandle

MiniRobotLanguage (MRL)

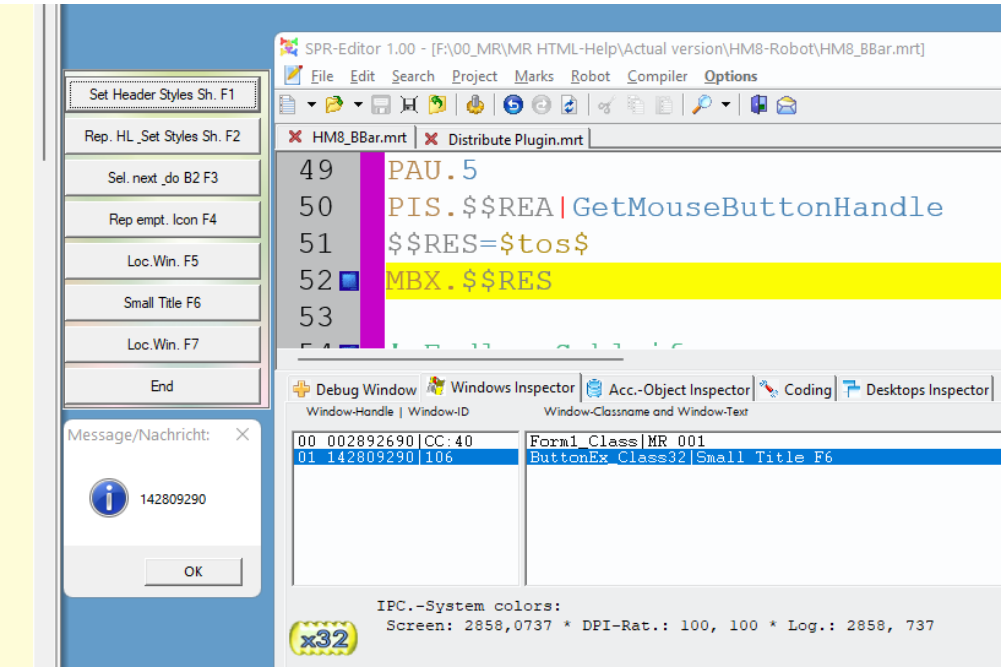
PIS.GetMouseButtonHandle Command

Get the Handle of the Plugin-Button where the Mouse is (Mouseover)

Intention

The `PIS.GetHandle` -Command will return the Window-Handle from the specified Buttonbar.
Using **P2** you can specify a number from 1 to (number of buttons) to get the Windowhandle from that Button.
The result is always placed on TOS.

```
PIS.$$REA|GetMouseButtonHandle
$$RES=$TOS$
```



Syntax

```
PIS.P1|GetMouseButtonHandle
PIS.P1|gmbh
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

P1 - `Variable` which contains the *`PlugIn Index-No.`*

The result is always placed on TOS. It will be "0" if the Mouse is not over any of the Buttons, otherwise the number of the Button is returned.

Example

```
'********************************
' Vertical Plugin-Sample
'********************************
'
'SPR Script-file: New Script_1
'Purpose:
'Author:
'Creation date: 02-08-2022 at 07:53:12
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
' This will run the NoLimits2 in case its not yet running
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=8
PIS.$$REA|sbn|$$BUN

' use any window to attach the plugin
PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|2|2

$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|GetHandle|1
$$RES=$tos$
MBX.The handle is $$RES

' Endlos-Schleife
DOL.1
  PIW.$$REA|§§BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SAV.Restore
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'-------------------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.$$BUN
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAN.4
    JSR.doButton04
  CAN.5
    JSR.doButton05
  CAN.6
    JSR.doButton06
  CAN.7
    JSR.doButton07
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'-------------------------------------------------------------
' Do Button 1
'-------------------------------------------------------------
:doButton01
GSB.Set_Style_Headline
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.1
MLM.
GSB.Move_Mouse_To_B1
RET.
'-------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Do Button 2
'-------------------------------------------------------------
:doButton02
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.0.25
MLM.
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
' Do Button 3
'-------------------------------------------------------------
' Jump to next and then Replace Headers
:doButton03
GSB.Locate_below_Selected
PAU.1
MLM.
PAU.1
WII.6
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.Move_Mouse_To_Button|3
RET.
'-------------------------------------------------------------
' Do Button 4
'-------------------------------------------------------------
:doButton04
GSB.Locate_false_Icon
GSB.Locate_Icon_change_Menu
GSB.Locate_right_Icon
GSB.Move_Mouse_To_Button|4
RET.
'-------------------------------------------------------------
' Do Button 5
'-------------------------------------------------------------
:doButton05
FOR.$$LL5|1|8
  GSB.doButton03
  PAU.1
  IEK.LSHIFT
    ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  EIF.
NEX.
GSB.Move_Mouse_To_Button|5
RET.
'-------------------------------------------------------
' Do Button 6
'-------------------------------------------------------
:doButton06
GSB.Do_Mark_MRL
GSB.Set_Style_Small_Page_Title
GSB.Locate_Topline_Start
MLC.
GSB.Move_Mouse_To_Button|6
RET.
'-------------------------------------------------------
' Do Button 7
'-------------------------------------------------------
:doButton07
GSB.Locate_TW
GSB.Move_Mouse_To_Button|7
RET.
'-------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
ENR.
'=======================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'-------------------------------------------------------
:Locate_TW
IEW.$$ULT
  STW.h|$$ULT
ELS.
  STW.ct|TFrmHelpMan|Help+Manual
  HTV.$$ULT
  SWP.1090,130|1414,1665|T
EIF.
RET.
'-------------------------------------------------------
:Open_Style_Menu
GSB.Locate_TW
'SWP.1666,287|297,93|T
'MMV.t|428,125
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControlSite|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControl|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupBarControl|Schriftart
MLI.w|117,61
WII.
PAU.0.1
RET.
'-------------------------------------------------------
:Set_Font_Style
$$STL=§§_01
STW.Ac|TdxBarDropdown
AGR.34!WII.MOS.|$$STL||1|3|60
RET.
'-------------------------------------------------------
:Do_Mark_MRL
GSB.Locate_Small_MRL
SKP.{END}
KYD.{LSHIFT}
SKP.{HOME}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------
:Do_Mark_Line
GSB.Locate_TW
SKP.{HOME}
KYD.{LSHIFT}
SKP.{END}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------

'-------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Locate_Headline
GSB.Locate_TW
MLI.t|584,249
RET.
'-------------------------------------------------------------
:Locate_Sub_Headline
GSB.Locate_TW
MLI.t|835,300
RET.
'-------------------------------------------------------------
:Locate_right_Sub_Headline
GSB.Locate_TW
MLC.t|1126,298
RET.
'-------------------------------------------------------------
:Mark_Headline
GSB.Locate_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_right_Sub_Headline
GSB.Locate_right_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Full_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Set_Style_Line_Style
GSB.Open_Style_Menu
GSB.Set_Font_Style|LineStile
RET.
'-------------------------------------------------------------
:Set_Style_Small_Page_Title
GSB.Open_Style_Menu
GSB.Set_Font_Style|Small Page Title
RET.
'-------------------------------------------------------------
:Set_Style_Headline
GSB.Mark_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
RET.
'-------------------------------------------------------------
:Set_Style_Sub_Headline
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
GSB.Mark_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading2
RET.
'-------------------------------------------------------------
:Set_Style_right_Sub_Headline
GSB.Mark_right_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading3
RET.
'-------------------------------------------------------------
:Move_Mouse_To_Button
$$STL=§§_01
VBT.$$STL
$$TXT=F$$STL
STW.ct|Form1_Class|MR
SCW.ct|ButtonEx_Class32|$$TXT
MPO.
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B1
GSB.Move_Mouse_To_Button|1
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B2
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Locate_selected_TV_Element
' Code removed
RET.
'-------------------------------------------------------------
:Locate_below_Selected
' Code removed
RET.
'-------------------------------------------------------------
:Locate_false_Icon
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Icon_change_Menu
' Code removed
RET.
'-------------------------------------------------------------
:Locate_right_Icon
' Coder removed
RET.
'-------------------------------------------------------------
:Locate_Small_MRL
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Topline_Start
' Code removed
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
ENR.
```

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:

```
PIS.$$PIN|sbt|all|Button %n
```

The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.9 PIS.MouseMessageFilter

MiniRobotLanguage (MRL)

PIS.MouseMessageFilter Command

Set/Change which Mouse-Messages you get inside of PIW. for each Button.

Intention

The `PIS.MouseMessageFilter` -Command will let you decide for each Button, which Mouse-Messages you want to get - or not.
Generally a "Mouse-Event" can be a Button-Click, left or right Mousebutton, or it can just be an Mouseover, see the `PIW.-Command` for Details on which Mouse-Events the Robot supports.
Now each Mouse-Event needs your programs attention. Therefore you can now filter which Mouse-Event should be sent or not.
This is done, using the `PIS.MouseMessageFilter` -Command.
You define the filter once and then you will only get those messages that you really want.

0 - receive alle Messages (reset Filter)
1 - No Messages for Autorun-Executable-Events
2 - No DoubleClick-Messages
4 - No Singleclick Messages
8 - No Mouseover Messages
16 - No Mouse-Left-Messages (Mouse leaving the Button = End of Mouse-Over)

You just add the Numbers that you want to filter out, and call the

```
PIS.$$REA|MouseMessageFilter|8|(8+16+1)
' or add the numbers yourself:
PIS.$$REA|MouseMessageFilter|8|25
```

Command.

If you want to reset the filter to get all Messages, use:

```
PIS.$$REA|MouseMessageFilter|8|0
```

Here is an example:

```
  $$PAT=?path\Hallo.exe
PIS.$$REA|MouseMessageFilter|8|1
PIS.$$REA|setautorunsynchron|8|$$PAT

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
```

Syntax

PIS.P1|MouseMessageFilter|P2|P3
PIS.P1|Mmf|P2|P3

Parameter Explanation

P1 - Variable which contains the *PlugIn Index-No.*

P2 - *Button Number*, starting with "1" to 40

P3 - ***value for the Filter*** **(0 to 31) see description above.**

All commands are specific for the type of PlugIn used.

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:

© 2023 Theo Gottwald * https://www.smart-package.com

```
PIS.$$PIN|sbt|all|Button %n
```

The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.10 PIS.SetAutoHide

MiniRobotLanguage (MRL)

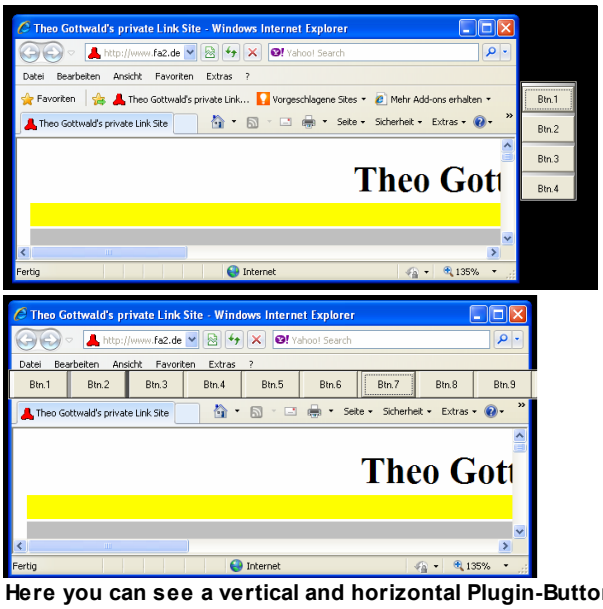
PIS.SetAutoHide Command

PlugInSend Set Autohide on/off

Intention

The `PIS.SetAutoHide`-Command will set the Behavior of the Buttonbar, in case the Target-Window becomes unavailable.
If the Target Window gets disabled, invisible or is minimized, then the Buttonbar will automatically become invisible.
In this case it will automatically return to visibility when the target window is back.
This Behavior can be influenced with the command "`PIS.$$PIN|SetAutoHide|0/1`" Command.

```
' Will switch off the Autohide-Feature
PIS.$$PIN|SetAutoHide|0
' Will switch on the Autohide-Feature
PIS.$$PIN|SetAutoHide|1
```



Here you can see a vertical and horizontal Plugin-Buttonbar. Below you will find more Samples.

Important note:
If you use `PIS.$$PIN|Hide` to make the Buttonbar invisible, the AutoHide Feature will NOT make it visible.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

PIS.P1|SetAutoHide|P2
PIS.P1|Sah|P2

Parameter Explanation

P1 **-** `Variable which contains the PlugIn Index-No.`
P2 **-** `0/1 - Switch on(off the AutoHide-Feature`

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.11 PIS.SetAutoRunSynchron / SetAutoRunASynchron

PIS.SetAutoRunSynchron / SetAutoRunASynchron Previous 2702 Top 41 Next 2706

MiniRobotLanguage (MRL)

PIS.SetAutoRunSynchron PIS.SetAutoRunASynchron Command

PlugInSend Run Executable on Button-Press (deactivate Button for the time)

Intention

Using the `PIS.SetAutoRunSynchron / SetAutoRunASynchron`, you can make the Buttonbar
directly start an executable. You will still be notified in the `PIW.`-Command that the Button has been pressed,
this will happen after the executable has been started.

So what is the difference between running an executable ASynchron or Synchron?
If you run it "ASynchron" the Plugin will just run the executable and immediately continue what its doing,
NOT waiting for the executable to end its task.
If you run it "Synchron" the Plugin will run the executable and wait for the executable to end its task.
In that time, the Button will be disabled.

```
$$PAT=?path\Hallo.exe
' Define Button 7 to start the Hallo.exe "ASynchroneous"
PIS.$$REA|SetAutoRunASynchron|7|$$PAT
' Define Button 8 to start the Hallo.exe "Synchroneous"
PIS.$$REA|setautorunsynchron|8|$$PAT
```

Syntax

PIS.P1|SetAutoRunSynchron|P2|P3
PIS.P1|Sas|P2|P3

PIS.P1|SetAutoRunASynchron|P2|P3
PIS.P1|Saa|P2|P3

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** Variable which contains the PlugIn Index-No.
P2 **-** ***Value,*** Button Number
P3 **-** Variable with the complete Path of the Executable to start on the press of the Button.

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.12 PIS.SetAutoRunSynchronNoDisable / SetAutoRunASynchronNoDisable

MiniRobotLanguage (MRL)

PIS.SetAutoRunSynchronNoDisable PIS.SetAutoRunASynchronNoDisable

PlugInSend Run Executable on Button-Press (do NOT deactivate Button for the time)

Intention

Using the `PIS.SetAutoRunSynchronNoDisable / SetAutoRunASynchronNoDisable`,
you can make the Buttonbar directly start an executable. You will still be notified in the
`PIW.`-Command that the Button has been pressed, this will happen after the executable has been started.

So what is the difference between running an executable ASynchron or Synchron?
If you run it "ASynchron" the Plugin will just run the executable and immediately continue what its doing,
NOT waiting for the executable to end its task.
If you run it "Synchron" the Plugin will run the executable and wait for the executable to end its task.
In that time, the Button will NOT be disabled.

```
$$PAT=?path\Hallo.exe
' We do not want a Mousemessage in PIW. for Button 8 so we
PIS.$$REA|MouseMessageFilter|8|1

' Define Button 7 to start the Hallo.exe "ASynchroneous"
PIS.$$REA|SetAutoRunASynchronNoDisable|7|$$PAT
' Define Button 8 to start the Hallo.exe "Synchroneous"
PIS.$$REA|setautorunsynchronNoDisable|8|$$PAT
```

Generally a Mouse-Message will be sent to the PIW.
If you do not want to get a Message that the Button was pressed, then you can use the

```
PIS.$$REA|MouseMessageFilter|8|1
```

Command to decide which messages you want or not.
Value "1" means "No messages for Running Executables".
Value "0" will clear the filter and then you will get all messages.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

PIS.P1|SetAutoRunSynchron|P2|P3
PIS.P1|Sas|P2|P3

PIS.P1|SetAutoRunASynchron|P2|P3
PIS.P1|Saa|P2|P3

Parameter Explanation

P1 **-** Variable which contains the PlugIn Index-No.
P2 **-** ***Value,*** Button Number
P3 **-** Variable with the complete Path of the Executable to start on the press of the Button.

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.13 PIS.SetButtonForeColor / SetButtonBackColor

MiniRobotLanguage (MRL)

PIS.SetButtonForeColor Command
PIS.SetButtonBackColor Command

PlugInSend Set Color of of Buttons

Intention

The `PIS.SetButtonForeColor / SetButtonBackColor`-Command will set the color of the Buttons.

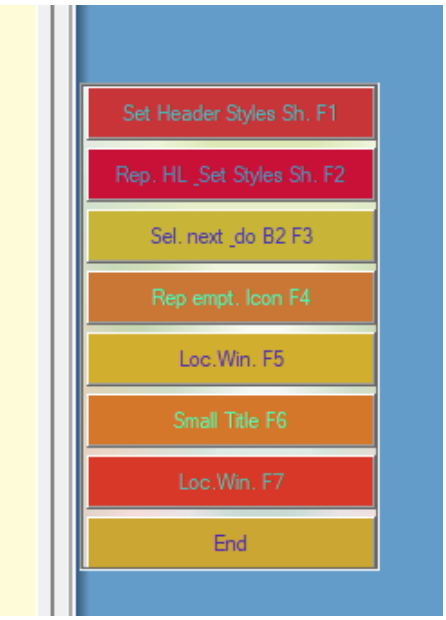
Using `PIS.SetButtonForeColor` you can specify the color in which the Text will appear on the Button.
Using `PIS.SetButtonBackColor` you can specify the Background color of the Button.
The color is generally specified as a decimal or hexadecimal number like "`&h123456`".
On this number the Order is:"`&HRRGGBB`" each of the pairs is 0-FF (hex).
For any of these commands, you can specify "`default`" instead, then the Windows-default color will be chosen.

```
PIS.$$REA|SetButtonForeColor|4|&Hffa0b3
PIS.$$REA|SetButtonForeColor|5|default

PIS.$$REA|SetButtonBackColor|4|&HF0A0B3
PIS.$$REA|SetButtonBackColor|5|default
```

```
:Again
FOR.$$LU2|1|255|1
  RND.1|$$BUN|$$BTN
  RND.0|255|$$GRN
  VTH.$$GRN|$$GRN
  VTH.$$LU2|$$RED
  CAL.$$HLU=255-$$LU2
  VTH.$$HLU|$$BLU
  VAR.$$BGC=&H$$RED$$GRN$$BLU
  PIS.$$REA|SetButtonBackColor|$$BTN|$$BGC
  ANA.Complementary_Color|$$BGC|$$FOR
  PIS.$$REA|SetButtonForeColor|$$BTN|$$FOR
  PAU.0.01
NEX.
MBX.!1
GTO.Again
```

Samples for Colored Buttons, using the Script above.

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

```
PIS.P1|SetButtonForeColor|P2|P3
PIS.P1|Sbfc|P2|P3

PIS.P1|SetButtonBackColor|P2|P3
PIS.P1|Sbbc|P2|P3
```


© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** `Variable which contains the PlugIn Index-No.`
P2 **-** `value - Button-Number of the Button the color should be changed.`
P3 **-** `value - Color-Value "&Hffa0B1" in Format "&HRRGGBB" or the text: "default"`

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.14 PIS.SetButtonNumber

MiniRobotLanguage (MRL)

PIS.SetButtonNumber Command

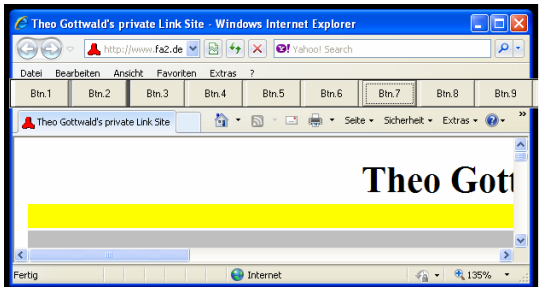
PlugInSend Set Number of Buttons

Intention

The `PIS.SetButtonNumber`-Command will set the number of visible Buttons. Generally the Plugin has 40 Buttons, but only the specified Number of buttons will be available to the User.

```
' Will set the number of visible Buttons to 4
PIS.$$PIN|SetButtonNumber|4
```

```
' Will set the number of visible Buttons to 9
PIS.$$PIN|SetButtonNumber|9
```



Syntax

PIS.P1|SetButtonNumber|P2
PIS.P1|Sbn|P2

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** `Variable which contains the PlugIn Index-No.`
P2 **-** `value - number of visible Buttons (1-40)`

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.15 PIS.SetButtonsHDist / SetButtonsVDist

MiniRobotLanguage (MRL)

PIS.SetButtonsHDist Command
PIS.SetButtonsVDist Command

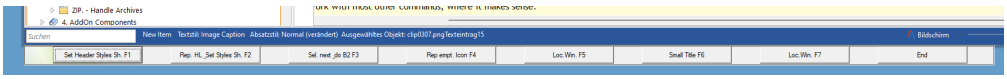
Set/Change Picture on Plugin-Buttons

Intention

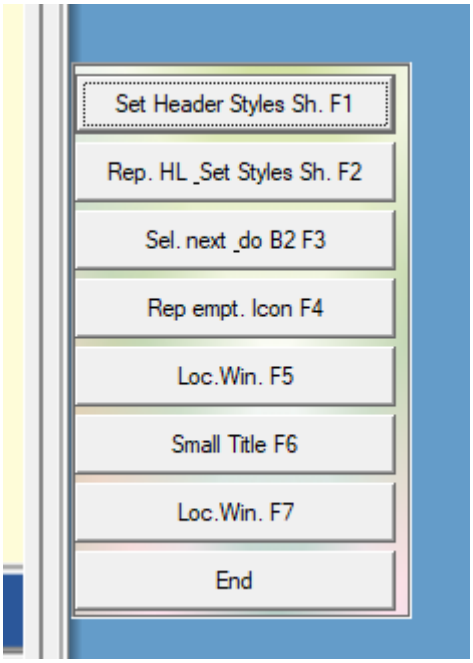
The `PIS.SetButtonHDist` -Command will set or change the horizontal distance between the Buttons on a horizontal Buttonbar.
Please do not use values smaller then 4 to avoid painting problems.

```
PIS.$$REA|SetButtonsHDist|4
```

```
PIS.$$REA|SetButtonsHDist|50
```

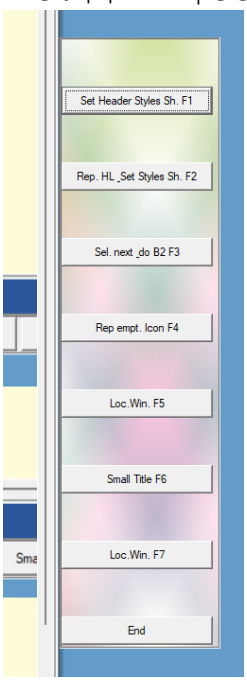


```
PIS.$$REA|SetButtonsVDist|4
```



*© 2023 Theo Gottwald * https://www.smart-package.com*

PIS.$$REA|SetButtonsVDist|50



Syntax

PIS.P1|SetButtonsHDist|P2
PIS.P1|sbhd|P2

PIS.P1|SetButtonsVDist|P2
PIS.P1|sbvd|P2

Parameter Explanation

P1 - Variable which contains the *PlugIn Index-No.*

P2 - ***Distance between*** *Button in Pixel, if 0, the Value will not be changed.*
Values below 4 are not recommended because these can lead to optical painting problems.

All commands are specific for the type of PlugIn used.

Example

-

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.16 PIS.SetButtonsHVSize

PIS.sbhvs - SetButtonsHVSize, sbhs, sbvs Previous 2702 Top 41 Next 2706

MiniRobotLanguage (MRL)

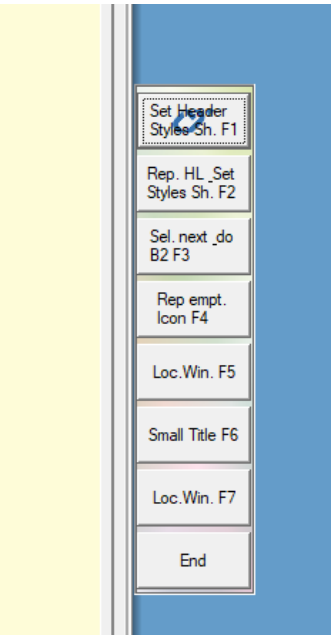
PIS.SetButtonsHVSize Command
PIS.SetButtonsHSize Command
PIS.SetButtonsVSize Command

Set/Change Size of all Plugin-Buttons

Intention

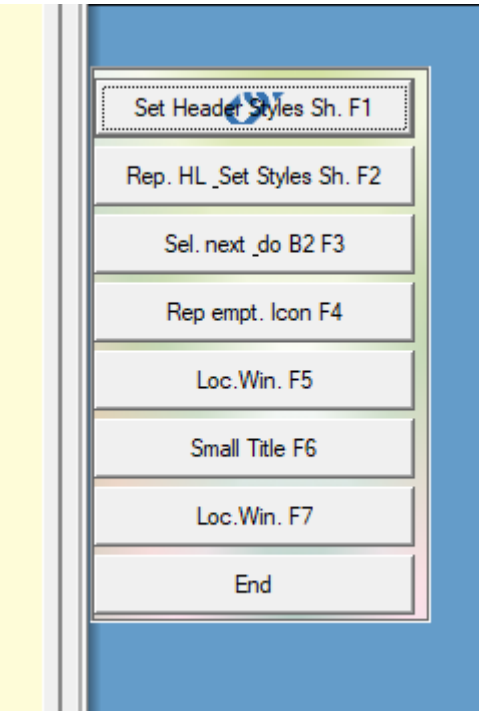
The `PIS.SetButtonsHVSize` -Command will set or change the Size of the Buttons from the specified Buttonbar.
You can Re-Size the Buttons any time.

```
PIS.$$REA|SetButtonsHVSize|80|40
' Here we size the Buttons 80x40
```




```
PIS.$$REA|SetButtonsHVSize|160|30
' Here we size the Buttons 160x30
```


© 2023 Theo Gottwald * https://www.smart-package.com




```
' If you specify a "0" as Size-Parameter, the value is not
PIS.$$REA|SetButtonsHVSize|80|0
```

Syntax

PIS.P1|SetButtonsHVSize|P2|P3
PIS.P1|sbhvs|P2|P3

PIS.P1|SetButtonsHSize|P2
PIS.P1|sbhs|P2

PIS.P1|SetButtonsVSize|P2
PIS.P1|sbvs|P2

Parameter Explanation

P1 - Variable which contains the *PlugIn Index-No.*

© 2023 Theo Gottwald * https://www.smart-package.com

***P2* - *Value*, first Size-Parameter in Pixel, if this value is "0" Size is not changed**

only for the "sbhvs" - Command:
***P3* - *Value*, second Size-Parameter in Pixel, if this value is "0" Size is not changed**

All commands are specific for the type of PlugIn used.

Example

```
' This code is just a Syntax-Sample and will not run
PIS.$$REA|SetButtonsHVSize|80|40
PIS.$$REA|SetButtonsHVSize|270|90
PIS.$$REA|SetButtonsHSize|270
PIS.$$REA|SetButtonsVSize|90
```

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.17 PIS.SetButtonText

MiniRobotLanguage (MRL)

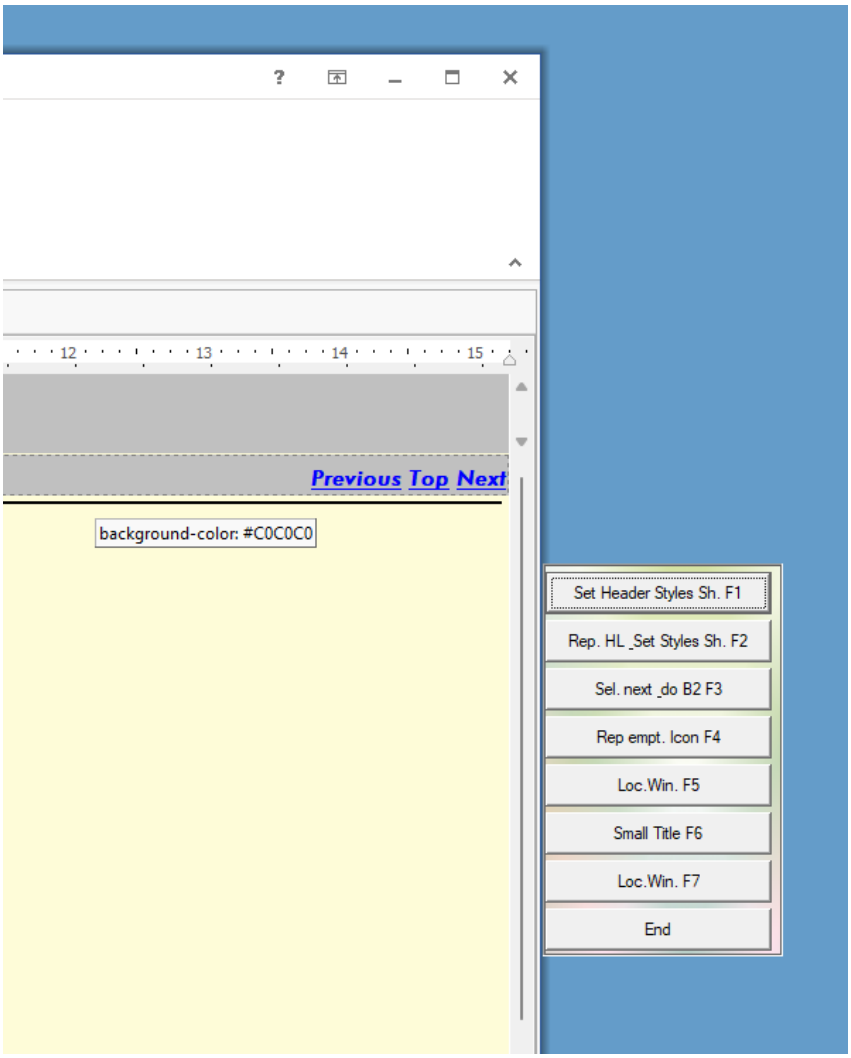
PIS.SetButtonText Command

Set/Change Text on Plugin-Buttons

Intention

The `PIS.SetButtonText` -Command will set or change the Text that is displayed on a Button from the specified Buttonbar.

*© 2023 Theo Gottwald * https://www.smart-package.com*



© 2023 Theo Gottwald * https://www.smart-package.com

```
' Return PlugIn Index Number
PIL.$$REA|01
' Set Button-Number
VAN.$$BUN=8
PIS.$$REA|sbn|$$BUN
' use any window to attach the plugin
PIS.$$REA|sbhd|4
' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|2|2

$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7
```

Syntax

PIS.P1|SetButtonText|P2|P3
PIS.P1|sBt|P2|P3

Parameter Explanation

P1 - Variable which contains the *PlugIn Index-No.*

P2 - *Button Number*, starting with "1"

P3 - ***Text*** **that should appear on the Button**

All commands are specific for the type of PlugIn used.

© 2023 Theo Gottwald * https://www.smart-package.com

```
  PIS.$$REA|sBt|$$BUN|End
  PIS.$$REA|sBt|1|Set Header Styles Sh. F1
```

Example

```
'*********************************
' Vertical Plugin-Sample
'*********************************
'VAR.$$PIC=24_about.bmp
EXO.iexplore
PIL.$$REA|01
' use any window to attach the plugin
STW.t|explorer
' Store the window handle in a variable
HTV.$$ULT

PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show

' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|0|0

' Set text on Button 1
FOR.$$CNT|1|20
  PIS.$$REA|sBt|$$CNT|Btn.$$CNT
NEX.

' Show a picture on Button 2
'PIS.$$REA|sp|2|$$PIC

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|sbn|1

' This double Loop will Show more or less buttons
FOR.$$LOA|1|2
  FOR.$$LOP|1|20
    PRT.->$$LOP
    PIS.$$REA|sbn|$$LOP
    PAU..2
  NEX.
NEX.

PAU.2
:enz
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.18 PIS.SetcorrectionFactor

MiniRobotLanguage (MRL)

PIS.SetcorrectionFactor Command

Set/Change Picture on Plugin-Buttons

Intention

The `PIS.SetcorrectionFactor` -Command will set or change the exact Position of the Border of the specified Buttonbar.

This is the Buttonbar-PlugIn without any corrections. If it does not look like this, you can make Corrections to the Form-Borders.

`PIS.$$PIL|SetcorrectionFactor|-10` will make the Buttonbar shorter.

`PIS.$$P` will make the Buttonbar larger.

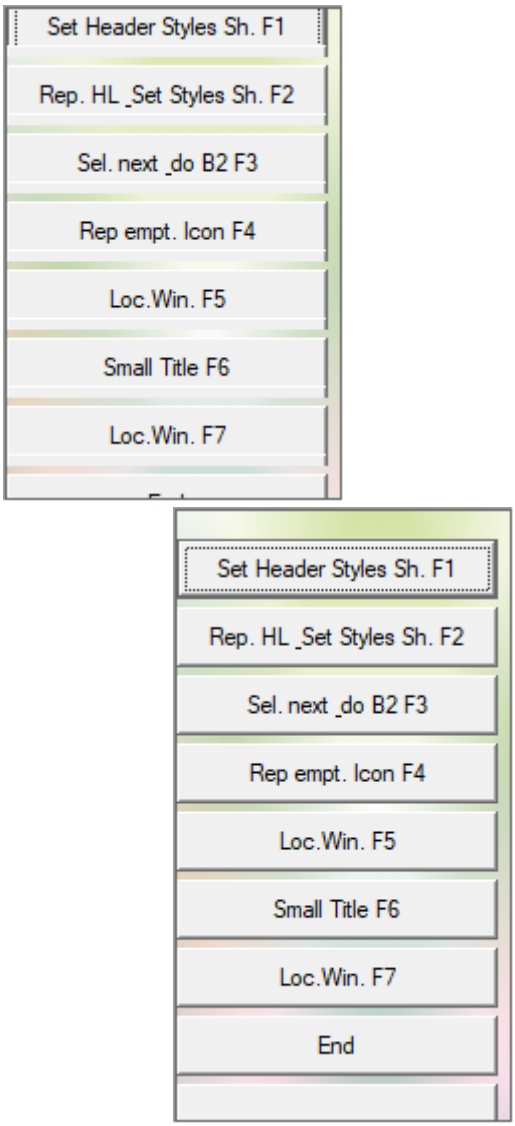
*© 2023 Theo Gottwald * https://www.smart-package.com*

`PIS.$$PIL|SetcorrectionFactor|0|-20` `PIS.$$PI`

will make the Buttonbar shorter.

will make the Buttonbar larger. Hidden Buttons may get visible (there is a total of 40 Buttons).

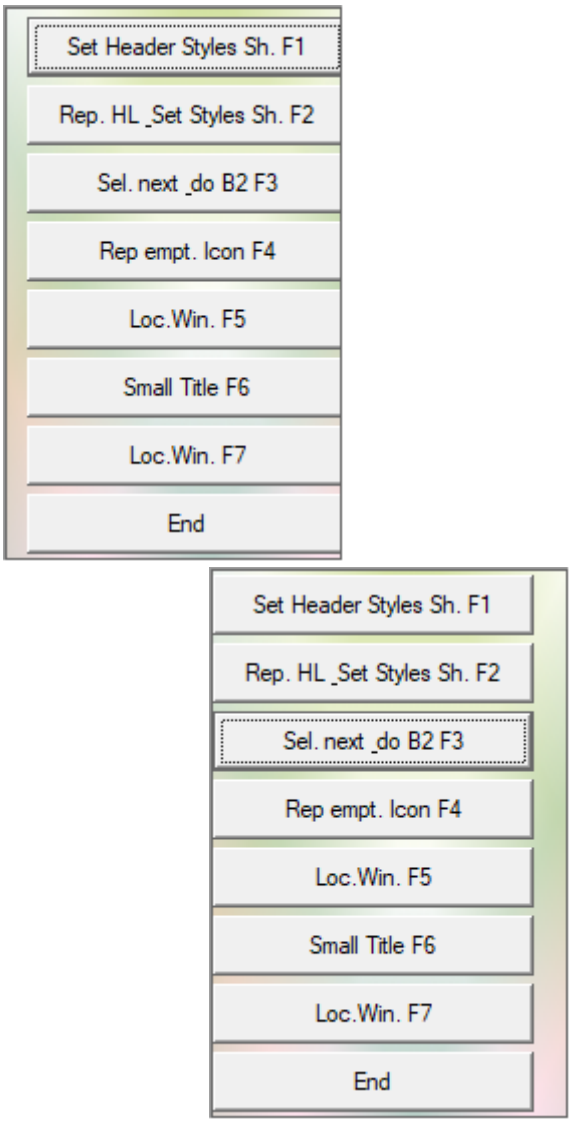
© 2023 Theo Gottwald * https://www.smart-package.com



`PIS.$$REA|SetcorrectionFactor|0|0|-10|0` will make the Buttonbar smaller to the right side.

`PIS.$$REA|S` will make the Buttonbar larger to the right side.

© 2023 Theo Gottwald * https://www.smart-package.com

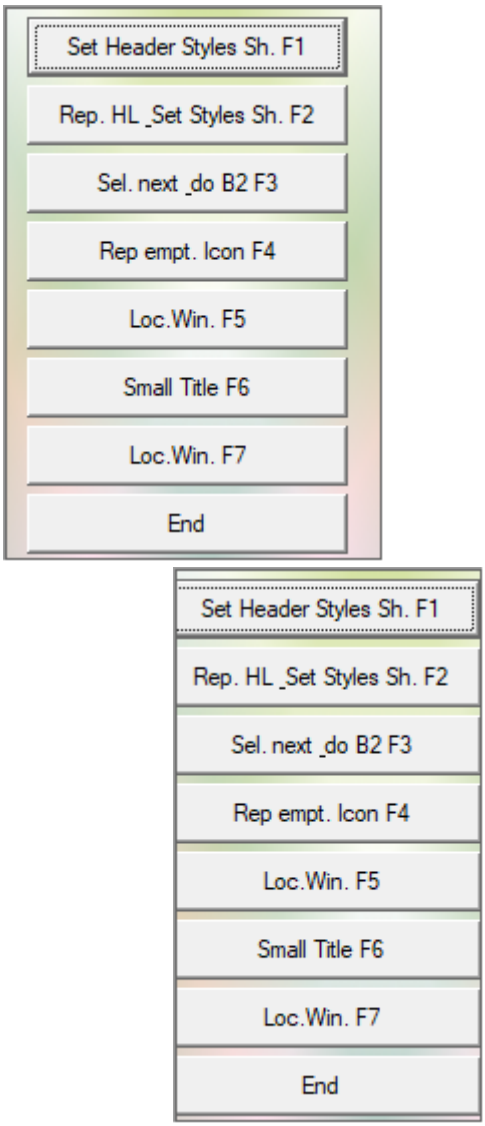


`PIS.$$REA|SetcorrectionFactor|0|0|0|20` `PIS.$$REA|S`

will make the Buttonbar smaller.

will make the Buttonbar larger.

© 2023 Theo Gottwald * https://www.smart-package.com



Hint:
Its recommended, to set the Correction-Factor only once after the PlugIn was created. Setting it multiple times may sometimes bring the Plugin-System internally into disorder.

Syntax

```
PIS.P1|SetcorrectionFactor|P2[|P3][|P4][|P5]
PIS.P1|scf|P2[|P3][|P4][|P5]
```

Parameter Explanation

P1 - Variable which contains the *PlugIn Index-No.*

© 2023 Theo Gottwald * https://www.smart-package.com

P2 *- Buttons-Position - Upper Border Space Correction, can be negative.*
All Buttons will move up or down by this number of Pixels. The normal range is between -10 and +10."0" is the default.

P3 *- Buttons-Position - Upper and Bottom Border Space Correction, can be negative.*
All Buttons will move up or down by this number of Pixels. The normal range is between -10 and +10."0" is the default.

P4 *- Buttons-Position - Left Border Space Correction, can be negative.*
All Buttons will move left or right by this number of Pixels. The normal range is between -10 and +10."0" is the default.

P5 *- Buttons-Position - Left and Right Border Space Correction, can be negative.*
All Buttons will move left or right by this number of Pixels. The normal range is between -10 and +10."0" is the default.

All commands are specific for the type of PlugIn used.

Example

```
' Button Bar with only very small Borders
PIS.$$REA|SetcorrectionFactor|-2|-4|4|-4
```

Remarks

You can use up to 5 Plugins at the same time.

© 2023 Theo Gottwald * https://www.smart-package.com

Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-
-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.19 PIS.SetPicture

MiniRobotLanguage (MRL)

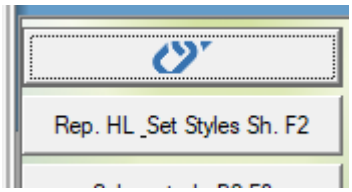
PIS.SetPicture Command

Set/Change Picture on Plugin-Buttons

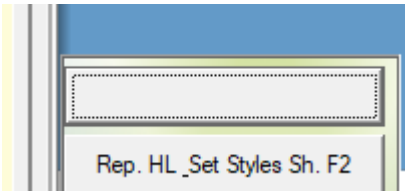
Intention

The `PIS.SetPicture` -Command will set or change the Picture that is displayed on a Button from the specified Buttonbar.
It can also Re-Size a picture during the loading to any size you specify.

```
$$FIX=?path\PA_32x16.bmp
PIS.$$REA|SetPicture|1|$$FIX|32|5
```




```
PIS.$$REA|SetPicture|1|
```
will remove the Picture.



For this Command, only the following File-Formats are supported:

- `.BMP.`
- `.ICO`
- `.CUR`

Syntax

PIS.P1|SetPicture|P2|P3[|P4|P5]
PIS.P1|sP|P2|P3[|P4|P5]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** Variable which contains the *PlugIn Index-No.*

P2 **-** *Button Number*, starting with "1"

P3 **- Filename (incl. Path) of the Picture to load.**

P4 - if 0 this Parameter is unused. Otherwise the X-Size of the Picture. (Resize-X)

P5 - if 0 this Parameter is unused. Otherwise the Y-Size of the Picture. (Resize-Y)

All commands are specific for the type of PlugIn used.

Example

```
' This code is just a Syntax-Sample and will not run
$$FIX=?path\PA_32x16.bmp
PIS.$$REA|SetPicture|1|$$FIX
' Remove Picture from Button
PIS.$$REA|SetPicture|1|
' Remove Text from Button
PIS.$$REA|sBt|1|
```

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.20 PIS.SetSetHorizontal / SetVertical

MiniRobotLanguage (MRL)

PIS.SetHorizontal Command
PIS.SetVertical Command

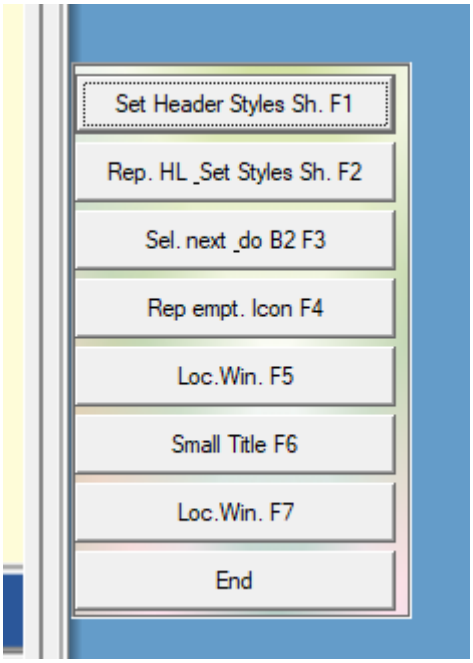
Set/Change the Button-Stacking Direction of the Plugin-Buttons

Intention

The `PIS.SetHorizontal ( PIS.SetVertical` -Command will set or change the stacking direction of the Buttons in the Buttonbar.

```
PIS.$$REA|SetHorizontal
```

```
PIS.$$REA|SetVertical
```



Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

PIS.P1|SetHorizontal
PIS.P1|sh

PIS.P1|SetVertical
PIS.P1|sv

Parameter Explanation

P1 **-** `Variable which contains the` *`PlugIn Index-No.`*

`All commands are specific for the type of PlugIn used.`

Example
-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-
-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.21 PIS.SetTopMost

MiniRobotLanguage (MRL)

PIS.SetTopMost Command

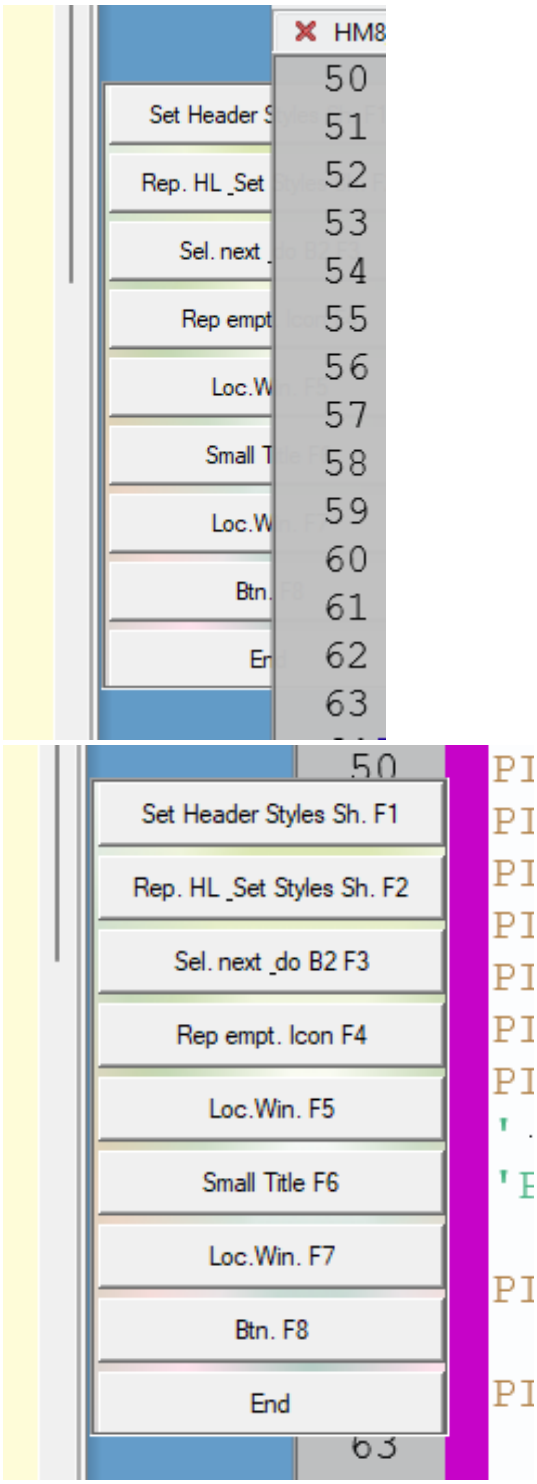
Set the Buttonbar to TopMost or not

Intention

The `PIS.SetTopMost` -Command will change the way, the Buttonbar is reacting if overlapped from other windows.
Generally the Buttonbar can be "TopMost" which is always above all other windows (unless they are also TopMost).
You can switch this TopMost Behavior on or Off using this Command.

```
PIS.$$REA|SetTopMost|0          PIS.$$REA|SetTopMost|1
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

```
PIS.P1|SetTopMost|P2
PIS.P1|stm|P2
```


© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** `Variable` which contains the *`PlugIn Index-No.`*

P2 **-** `value 0 or 1. 1- TopMost, 0 - On Top but NOT TopMost`

Example

```
'********************************
' Vertical Plugin-Sample
'********************************
'
'SPR Script-file: New Script_1
'Purpose:
'Author:
'Creation date: 02-08-2022 at 07:53:12
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=8
PIS.$$REA|sbn|$$BUN

' use any window to attach the plugin
PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|2|2

$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|GetHandle|1
$$RES=$tos$
MBX.The handle is $$RES

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.

SAV.Restore
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'-------------------------------------------------------
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.$$BUN
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAN.4
    JSR.doButton04
  CAN.5
    JSR.doButton05
  CAN.6
    JSR.doButton06
  CAN.7
    JSR.doButton07
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'-------------------------------------------------------
' Do Button 1
'-------------------------------------------------------
:doButton01
GSB.Set_Style_Headline
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.1
MLM.
GSB.Move_Mouse_To_B1
RET.
'-------------------------------------------------------
' Do Button 2
'-------------------------------------------------------
:doButton02
GSB.Mark_Headline
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.0.25
MLM.
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
' Do Button 3
'-------------------------------------------------------------
' Jump to next and then Replace Headers
:doButton03
GSB.Locate_below_Selected
PAU.1
MLM.
PAU.1
WII.6
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.Move_Mouse_To_Button|3
RET.
'-------------------------------------------------------------
' Do Button 4
'-------------------------------------------------------------
:doButton04
GSB.Locate_false_Icon
GSB.Locate_Icon_change_Menu
GSB.Locate_right_Icon
GSB.Move_Mouse_To_Button|4
RET.
'-------------------------------------------------------------
' Do Button 5
'-------------------------------------------------------------
:doButton05
FOR.$$LL5|1|8
  GSB.doButton03
  PAU.1
  IEK.LSHIFT
    ENR.
  EIF.
NEX.
GSB.Move_Mouse_To_Button|5
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
' Do Button 6
'-------------------------------------------------------------
:doButton06
GSB.Do_Mark_MRL
GSB.Set_Style_Small_Page_Title
GSB.Locate_Topline_Start
MLC.
GSB.Move_Mouse_To_Button|6
RET.
'-------------------------------------------------------------
' Do Button 7
'-------------------------------------------------------------
:doButton07
GSB.Locate_TW
GSB.Move_Mouse_To_Button|7
RET.
'-------------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
ENR.
'=============================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
:WSOut
RET.
'-------------------------------------------------------------
:Locate_TW
IEW.$$ULT
  STW.h|$$ULT
ELS.
  STW.ct|TFrmHelpMan|Help+Manual
  HTV.$$ULT
  SWP.1090,130|1414,1665|T
EIF.
RET.
'-------------------------------------------------------------
:Open_Style_Menu
GSB.Locate_TW
'SWP.1666,287|297,93|T
'MMV.t|428,125
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControlSite|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControl|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupBarControl|Schriftart
MLI.w|117,61
WII.
PAU.0.1
RET.
'-------------------------------------------------------------
:Set_Font_Style
$$STL=§$_01
STW.Ac|TdxBarDropdown
AGR.34!WII.MOS.|$$STL||1|3|60
RET.
'-------------------------------------------------------------
:Do_Mark_MRL
GSB.Locate_Small_MRL
SKP.{END}
KYD.{LSHIFT}
SKP.{HOME}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------
:Do_Mark_Line
GSB.Locate_TW
SKP.{HOME}
KYD.{LSHIFT}
SKP.{END}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------

'-------------------------------------------------------------
:Locate_Headline
GSB.Locate_TW
MLI.t|584,249
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
:Locate_Sub_Headline
GSB.Locate_TW
MLI.t|835,300
RET.
'-------------------------------------------------------------
:Locate_right_Sub_Headline
GSB.Locate_TW
MLC.t|1126,298
RET.
'-------------------------------------------------------------
:Mark_Headline
GSB.Locate_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_right_Sub_Headline
GSB.Locate_right_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Full_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Set_Style_Line_Style
GSB.Open_Style_Menu
GSB.Set_Font_Style|LineStile
RET.
'-------------------------------------------------------------
:Set_Style_Small_Page_Title
GSB.Open_Style_Menu
GSB.Set_Font_Style|Small Page Title
RET.
'-------------------------------------------------------------
:Set_Style_Headline
GSB.Mark_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
RET.
'-------------------------------------------------------------
:Set_Style_Sub_Headline
GSB.Mark_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading2
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
:Set_Style_right_Sub_Headline
GSB.Mark_right_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading3
RET.
'-------------------------------------------------------
:Move_Mouse_To_Button
$$STL=§§_01
VBT.$$STL
$$TXT=F$$STL
STW.ct|Form1_Class|MR
SCW.ct|ButtonEx_Class32|$$TXT
MPO.
RET.
'-------------------------------------------------------
:Move_Mouse_To_B1
GSB.Move_Mouse_To_Button|1
RET.
'-------------------------------------------------------
:Move_Mouse_To_B2
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------
'
'-------------------------------------------------------
:Locate_selected_TV_Element
' Code removed
RET.
'-------------------------------------------------------
:Locate_below_Selected
' Code removed
RET.
'-------------------------------------------------------
:Locate_false_Icon
' Code removed
RET.
'-------------------------------------------------------
:Locate_Icon_change_Menu
' Code removed
RET.
'-------------------------------------------------------
:Locate_right_Icon
' Coder removed
RET.
'-------------------------------------------------------
:Locate_Small_MRL
' Code removed
RET.
'-------------------------------------------------------
:Locate_Topline_Start
' Code removed
RET.
'-------------------------------------------------------
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:

```
PIS.$$PIN|sbt|all|Button %n
```

The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.22 PIS.SetTransparent

MiniRobotLanguage (MRL)

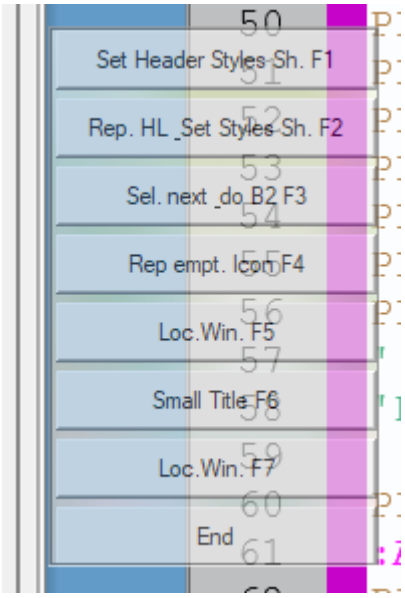
PIS.SetTransparent Command

Set/Change Transparency Level of Buttonbar

Intention

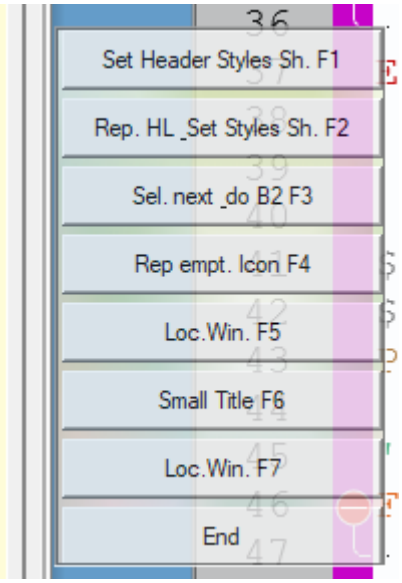
The `PIS.SetTransparent` -Command will set or change the Level of Transparency for the specified Buttonbar.
It can not make single Buttons Transparent, the value will always affect the whole Buttonbar.
The Transparency Values range from 0 to 255, where 0 is completely transparent (invisible) and 255 is solid (not transparent).
You can use Transparency and change the Buttons Background color.

```
PIS.$$PIL|SetTransparent|165
```




```
PIS.$$PIL|SetTransparent|210
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

PIS.P1|SetTransparent|P2
PIS.P1|str|P2

Parameter Explanation

P1 **-** Variable which contains the *PlugIn Index-No.*

P2 **-** value (0-255)- the transparency Values range from 0 to 255,
where 0 is completely transparent (invisible) and 255 is solid (not transparent).

All commands are specific for the type of PlugIn used.

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:

© 2023 Theo Gottwald * https://www.smart-package.com

Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-
-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.23 PIS.SetWindowAlert

MiniRobotLanguage (MRL)

PIS.SetWindowAlert Command

Set an Alarm, if a window with the same class comes in front

Intention

The `PIS.SetWindowAlert` -Command will set the Plugi to send an "`!Foreground!`" Message to the "`PIW.`" Command,
if a Topwindow with the specified Window-Class will get in Front.

What is it good for?
Assume you are working in a Chrome-Sort of Browser, and while you are working another Browser pops up in front.
To prevent the robot to work "in the wrong window", you will now get an "Alert Message".
Then you can react and possibly minimize that window, or do whatever you want with it.

```
' This will activate the Alert
PIS.$$PIL|SetWindowAlert|1|Chrome_Widget_Win

' This will deactivate the Alert
PIS.$$PIL|SetWindowAlert|0
```

Syntax

```
PIS.P1|SetWindowAlert|P2[|P3]
PIS.P1|swa|P2[|P3]
```

Parameter Explanation

P1 - `Variable` which contains the *`PlugIn Index-No.`*

P2 - `value 0 or 1. 1 -> Enable Alert, 0 -> Disable Alert`

P3 - `Window-Class - can use all sorts of Standard-Search Patterns, needed to enable the Alert.`

`In case of Alert you will get a Message in the PIW.-Command.`

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'********************************
' Vertical Plugin-Sample
'********************************
'
'SPR Script-file: New Script_1
'Purpose:
'Author:
'Creation date: 02-08-2022 at 07:53:12
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
GSB.RunIt
'SWP.700,498|1612,165|T

' Return PlugIn Index Number
PIL.$$REA|01

' Set Button-Number
VAN.$$BUN=8
PIS.$$REA|sbn|$$BUN

' use any window to attach the plugin
PIS.$$REA|sbhd|4

' Plugin vertikal anordnen
PIS.$$REA|sv
' Make the PlugIn visible
PIS.$$REA|show
' attach the Plugin to the window
PIS.$$REA|attach|$$ULT|2|2

$$HSI=160
$$VSI=30
PIS.$$PIN|sbhvs|$$HSI|$$VSI

' Set Button Text
FOR.$$LO2|1|$$BUN
  PIS.$$REA|sBt|$$LO2|Btn. F$$LO2
NEX.
PIS.$$REA|sBt|$$BUN|End
PIS.$$REA|sBt|1|Set Header Styles Sh. F1
PIS.$$REA|sBt|2|Rep. HL & Set Styles Sh. F2
PIS.$$REA|sBt|3|Sel. next & do B2 F3
PIS.$$REA|sBt|4|Rep empt. Icon F4
PIS.$$REA|sBt|5|Loc.Win. F5
PIS.$$REA|sBt|6|Small Title F6
PIS.$$REA|sBt|7|Loc.Win. F7

' Set Distance between buttons (vertical in Points)
PIS.$$REA|sbvd|4

PIS.$$REA|GetHandle|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$RES=$tos$
MBX.The handle is $$RES

' Endlos-Schleife
DOL.1
  PIW.$$REA|$$BTN|4
  ITO.
    PAU.0.1
    GTO.MLop
  EIF.
' Button wurde geklickt oder Mouseover
  GSB.react
  :MLop
OOP.
PAU.2
:enz
END.
'------------------------------------------------------------
' Starte NL2
:RunIt
$$PRO=HELPMAN.EXE
NEP.$$PRO
  STW.wctgX|0|TFrmHelpMan|Help+Manual|Locate_First
ELS.
  GTO.Locate_First
EIF.

SAV.Save|$$FIL|$$PAT
IAI.Help+Manual 8|s
  $$FIL=$$001
  BLB.$$FIL|$$FIL
  BLB.$$FIL|$$FIL
  VBT.$$FIL
  VBT.$$FIL|"
  $$FIN=$$FIL\$$PRO
  DBP.$$FIN
  NEF.$$FIN
    $$ERO=HM8 nicht gefunden
    GTO.Error
  EIF.
  EXE.$$FIN
  PAU.3
  GTO.RunIt
ELS.
  $$ERO=HM8 nicht installiert.
  GTO.Error
EIF.

SAV.Restore
:Locate_First
STW.ctw|TFrmHelpMan|Help+Manual|45
'SWP.700,498|1612,165|T
HTV.$$ULT
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
HTV.$$RIB
RET.
'------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:react
' Get Button number out of $$RET into §§TMA
SBD.§§BTN|$sp$|1|§§TMA
' Button Action into §§TMB
SBD.§§BTN|$sp$|2|§§TMB
'PRT. Button pressed: §§TMA
'PRT. Action-code: §§TMB
GSB.TestMo
' react on Buttonpress
IVS.§§TMB=!P!
  SCS.§§TMA
  CAN.$$BUN
    JSR.doEnd
  CAN.1
    JSR.doButton01
  CAN.2
    JSR.doButton02
  CAN.3
    JSR.doButton03
  CAN.4
    JSR.doButton04
  CAN.5
    JSR.doButton05
  CAN.6
    JSR.doButton06
  CAN.7
    JSR.doButton07
  CAE.
    MBX.Button-Number: §§TMA is not defined.
    : $$RET=
  ESC.
EIF.
RET.
'-------------------------------------------------------------
' Do Button 1
'-------------------------------------------------------------
:doButton01
GSB.Set_Style_Headline
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.1
MLM.
GSB.Move_Mouse_To_B1
RET.
'-------------------------------------------------------------
' Do Button 2
'-------------------------------------------------------------
:doButton02
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.WaitShift
GSB.Locate_below_Selected
PAU.0.25
MLM.
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
' Do Button 3
'-------------------------------------------------------------
' Jump to next and then Replace Headers
:doButton03
GSB.Locate_below_Selected
PAU.1
MLM.
PAU.1
WII.6
GSB.Mark_Headline
SDK.^v
GSB.Mark_Headline
PAU.0.5
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
GSB.Set_Style_Sub_Headline
GSB.Set_Style_right_Sub_Headline
'
GSB.Move_Mouse_To_Button|3
RET.
'-------------------------------------------------------------
' Do Button 4
'-------------------------------------------------------------
:doButton04
GSB.Locate_false_Icon
GSB.Locate_Icon_change_Menu
GSB.Locate_right_Icon
GSB.Move_Mouse_To_Button|4
RET.
'-------------------------------------------------------------
' Do Button 5
'-------------------------------------------------------------
:doButton05
FOR.$$LL5|1|8
  GSB.doButton03
  PAU.1
  IEK.LSHIFT
    ENR.
  EIF.
NEX.
GSB.Move_Mouse_To_Button|5
RET.
'-------------------------------------------------------------
' Do Button 6
'-------------------------------------------------------------
:doButton06
GSB.Do_Mark_MRL
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
GSB.Set_Style_Small_Page_Title
GSB.Locate_Topline_Start
MLC.
GSB.Move_Mouse_To_Button|6
RET.
'-------------------------------------------------------------
' Do Button 7
'-------------------------------------------------------------
:doButton07
GSB.Locate_TW
GSB.Move_Mouse_To_Button|7
RET.
'-------------------------------------------------------------
:TestMo
' react on Mouseover
IVS.§§TMB=!M!O!
  '
  GTO.MOut
EIF.
' react on Mouse-Leaves Button Area
IVS.§§TMB=!M!L!
  '
  GTO.MOut
EIF.
RET.
' On Mouseover return directly to Mainloop (RET.2)
:MOut
RET.2
'-------------------------------------------------------------
:doEnd
END.
'-------------------------------------------------------------
:Error
MBX.$$ERO
GTO.doEnd
ENR.
'=============================================================
' These Subprogramms are unused
:GetMousePos
GSB.WaitShift
GMP.mw|§§POS
PRT.Mouse is: §§POS
RET.
'-------------------------------------------------------------
:WaitShift
WNK.
:Laba
IEK.LSHIFT
  GTO.WSOut
ELS.
  PAU.0.1
  GTO.Laba
EIF.
:WSOut
RET.
'-------------------------------------------------------------
:Locate_TW
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IEW.$$ULT
  STW.h|$$ULT
ELS.
  STW.ct|TFrmHelpMan|Help+Manual
  HTV.$$ULT
  SWP.1090,130|1414,1665|T
EIF.
RET.
'-------------------------------------------------------------
:Open_Style_Menu
GSB.Locate_TW
'SWP.1666,287|297,93|T
'MMV.t|428,125
SCW.nct|1|TdxHMRibbon|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControlSite|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupsDockControl|{&NOTEXT:}
SCW.nct|1|TdxRibbonGroupBarControl|Schriftart
MLI.w|117,61
WII.
PAU.0.1
RET.
'-------------------------------------------------------------
:Set_Font_Style
$$STL=§§_01
STW.Ac|TdxBarDropdown
AGR.34!WII.MOS.|$$STL||1|3|60
RET.
'-------------------------------------------------------------
:Do_Mark_MRL
GSB.Locate_Small_MRL
SKP.{END}
KYD.{LSHIFT}
SKP.{HOME}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------
:Do_Mark_Line
GSB.Locate_TW
SKP.{HOME}
KYD.{LSHIFT}
SKP.{END}
KYU.{LSHIFT}
PAU.1
RET.
'-------------------------------------------------------------

'-------------------------------------------------------------
:Locate_Headline
GSB.Locate_TW
MLI.t|584,249
RET.
'-------------------------------------------------------------
:Locate_Sub_Headline
GSB.Locate_TW
MLI.t|835,300
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
:Locate_right_Sub_Headline
GSB.Locate_TW
MLC.t|1126,298
RET.
'-------------------------------------------------------------
:Mark_Headline
GSB.Locate_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_right_Sub_Headline
GSB.Locate_right_Sub_Headline
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Mark_Full_Sub_Headline
GSB.Locate_Sub_Headline
SDK.^a
SDK.^a
'GSB.WaitShift
RET.
'-------------------------------------------------------------
:Set_Style_Line_Style
GSB.Open_Style_Menu
GSB.Set_Font_Style|LineStile
RET.
'-------------------------------------------------------------
:Set_Style_Small_Page_Title
GSB.Open_Style_Menu
GSB.Set_Font_Style|Small Page Title
RET.
'-------------------------------------------------------------
:Set_Style_Headline
GSB.Mark_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading1
RET.
'-------------------------------------------------------------
:Set_Style_Sub_Headline
GSB.Mark_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading2
RET.
'-------------------------------------------------------------
:Set_Style_right_Sub_Headline
GSB.Mark_right_Sub_Headline
GSB.Open_Style_Menu
GSB.Set_Font_Style|Heading3
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
'-------------------------------------------------------------
:Move_Mouse_To_Button
$$STL=§§_01
VBT.$$STL
$$TXT=F$$STL
STW.ct|Form1_Class|MR
SCW.ct|ButtonEx_Class32|$$TXT
MPO.
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B1
GSB.Move_Mouse_To_Button|1
RET.
'-------------------------------------------------------------
:Move_Mouse_To_B2
GSB.Move_Mouse_To_Button|2
RET.
'-------------------------------------------------------------
'
'-------------------------------------------------------------
:Locate_selected_TV_Element
' Code removed
RET.
'-------------------------------------------------------------
:Locate_below_Selected
' Code removed
RET.
'-------------------------------------------------------------
:Locate_false_Icon
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Icon_change_Menu
' Code removed
RET.
'-------------------------------------------------------------
:Locate_right_Icon
' Coder removed
RET.
'-------------------------------------------------------------
:Locate_Small_MRL
' Code removed
RET.
'-------------------------------------------------------------
:Locate_Topline_Start
' Code removed
RET.
'-------------------------------------------------------------
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.71.4.24 PIS.Show/Hide

MiniRobotLanguage (MRL)

PIS.Show/Hide Command

PlugInSend Show/Hide

Intention

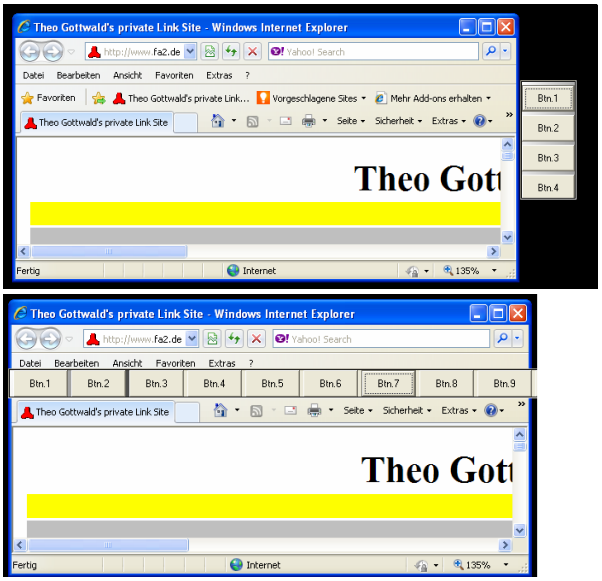
The `PIS.Show`-Command will make the Buttonbar visible.
The `PIS.Hide`-Command will make the Buttonbar invisible.

`"sho","Show"`
`PIS.$$PIN|Show`

`"Hid","Hide"`
`PIS.$$PIN|Hide`

Using the "Show/Hide" Subcommand, you can make the Buttonbar invisible.
For example, to prevent the user from pressing Buttons.

If the Target Window gets disabled, invisible or is minimized, then the Buttonbar may automatically become invisible.
In this case it will automatically return to visibility when the target window is back.
This Behavior can be influenced with the command "`PIS.$$PIN|SetAutoHide|0/1`" Command.



Here you can see a vertical and horizontal Plugin-Buttonbar. Below you will find more Samples.

© 2023 Theo Gottwald * https://www.smart-package.com

Important note:
If the Target Window is minimized, the Buttonbar will also disappear. Once the Target Window comes back, the Buttonbar will also come back.
This Behavior can be influenced with the command "`PIS.$$PIN|SetAutoHide|0/1`" Command.

Syntax

```
PIS.P1|Show
PIS.P1|Sho

PIS.P1|Hide
PIS.P1|Hid
```

Parameter Explanation

P1 **-** `Variable which contains the PlugIn Index-No.`

Example

-

Remarks

You can use up to 5 Plugins at the same time.
Just start them, each will get another Index.
To communicate with each PlugIn, you just give the PlugIn-Index to each command.

Button-Colors and transparency:
Please note that if you set Buttons transparent, that they can not have a background-color at the same time.

To set multiple Buttontext at once you can use:
`PIS.$$PIN|sbt|all|Button %n`
The "%n" is replaced with the number of that button.
The "all" and %n-scheme will also work with most other commands, where it makes sense.

Limitations:-

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- PIL. - PlugIn Launch 2702
- PIS. - PlugIn Send 2749
- PIW. - PlugInWait 2706
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.72 PopUp - Handling

3.72.1 DBC. - Do-Button-Clicks

DBC. - Do-Button-Clicks Previous 2706 Top 41 Next 2881

MiniRobotLanguage (MRL)

DBC. Command

Do Button Clicks

Intention

DBC. is a Parallel running Button-Clicker Robot.

It will click on PopUp-Messageboxes that may or may not appear during automated installations.

Imagine a automated Setup-Script. Sometimes a Messagebox appears "This folder already exists". The problem in such cases is, that you have to deal with Messageboxes, which:

- appear but not always
- can appear more often then once (10 or 100 times?)
- may not appear at all
- may have slightly varying texts
- must be clicked or closed

The solution to this problem is the DBC.-command.

It runs in the background and when a specified message box appears, the DBC. will click or close it. If the Messagebox will not appear, the DBC. will have no effect on the running script.

If the messagebox will come 100 time, it will be clicked 100 times.

DBC. is a special sort of background-task and can only be accessed with the DBC. - command.

Syntax

```
DBC.P1|P2|P3|P4[|P5|P6..|P10]
```

Parameter Explanation

Leave unneeded Parameters empty.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
P1 - a  = add DBC: start DBC. task
     r  = remove DBC: remove DBC. task

P2 - Variable: returns the DBC.-Handle for later removing

P3 - Title-Text of the Top-Window of the messagebox

P4 - Button-Text of the button to click on

P5 - Text of a static text inside the Messagebox

P6 - PID - Process ID of Process behind the Messagebox

P7 - action-code, normally
     1 = MLE. (Direct Engine Click)
     2 = close TopWindow
     4 = MLM. (compatible Engine)
     8 - MLI. (compatible Engine)
    16 - Close spezial

P8 - class of the Top-Window, if omitted "#32770" is default

P9 - class of the button-control, if omitted "button" is
default

P10 - class of the static-text. If omitted, "static" or
"syslink" is default
```

Example

```
'***********************************************
' DBC.-Sample 1
'***********************************************
' This will create some messageboxes in the background.
FOR.$$LOP|1|10
  PRR.
:ag
    MBX.This is a Text.
    END.
  PRE.
NEX.
PAU.1
'-------------------------------------------------------------
' Here we start the DBC.
DBC.a|$$RUK|remote&OR:Nachricht|Ok|This is a||1|#32770|button
PAU.2
MBX.Ende
END.

'***********************************************
' DBC.-Sample 2
'***********************************************
' This will create some messageboxes in the background.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRR.
FOR.§§LOP|1|180
MBX.Hallo||Nachricht-§§LOP
NEX.
END.
PRE.
'-------------------------
DBC.a|$$UK|remote&OR:Nachricht|Ok
PRT.My ID is:$$RUK
PAU.15
MBX.Ende
END.
'*********************************************
```

Remarks

`DBC.` ist not the "Mouseclick".-Command. If you just want to click on a button, use `MLC.`,`MLE.`,`MLM.`, or `MLI.`

The `DBC.`-Command is a background-Task for suddenly appearing Messageboxes.

The Action Parameter 16 (`MLM.`) generates a small Move on the target button, before the Click (`MLI.`). This way the command is more compatible, with few difficult application.

Limitations:

There is no limit in the number of `DBC.`-commands that can be active at the same time.

See also:

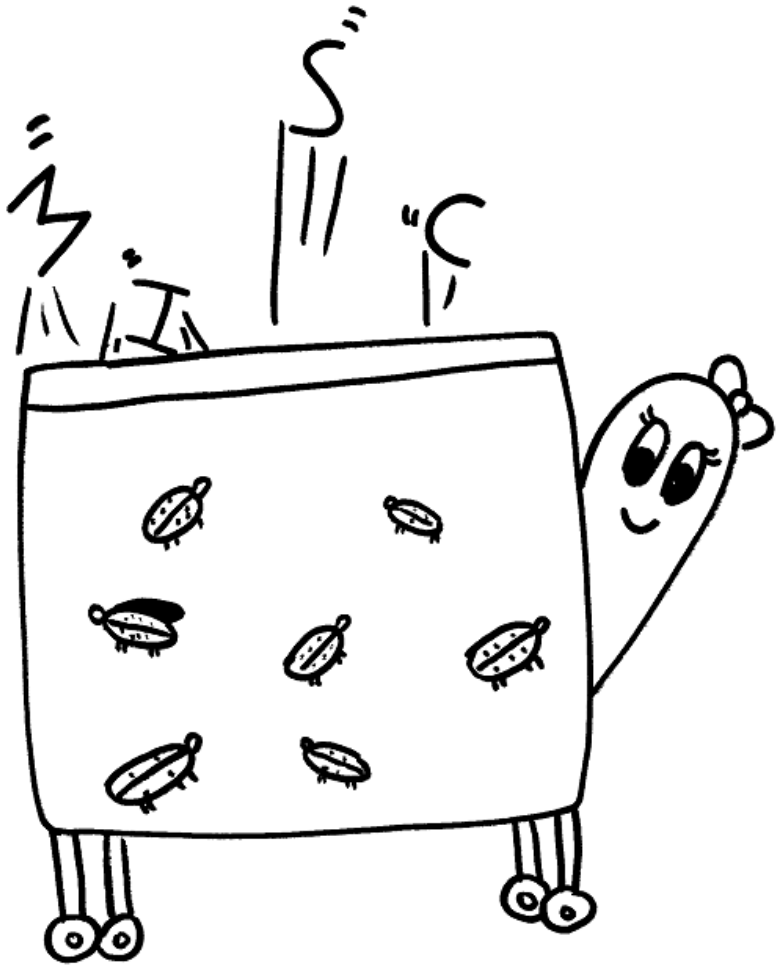
- MLE. - Mouse-Left-Click Event 2492
- BSC. - Button Special Click 2435
- CLW. - Close-Window 3592
- MLI - Mouse Left Click qulckly (Invinsible) 2454
- MLM - Mouse Left click Move (Invinsible) 2457
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.73 Powershell Inline

3.73.1 !Smarty's Powershell Inline

Smarty's Powershell-Support



Here is an Introduction to Powershell:
Getting started with Powershell

See also here:
Powershell Reference

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
An A-Z Index of Windows PowerShell commands

        %                       Alias for ForEach-Object
        ?                       Alias for Where-Object
```

A

```
    Get-Acl                     Get permission settings for a file or registry ke
    Set-Acl                     Set permissions
        Active Directory        Account, Computer, Group and User cmdlets
    Get-Alias           gal     Return alias names for Cmdlets
 Import-Alias          ipal     Import an alias list from a file
    New-Alias           nal     Create a new alias.
    Set-Alias           sal     Create or change an alias
 Compress-Archive               Create a new archive/zipped file [PS 5+]
   Expand-Archive               Extract files from an archive (zipped) file [PS 5
   Get-AuthenticodeSignature    Get the signature object associated with a file
   Set-AuthenticodeSignature    Place a signature in a .ps1 script or other file
```

B

```
      Backup-GPO                Backup group policy objects (GPOs)
      Begin                     Function BEGIN block
      BITS                      Background Intelligent Transfer Service cmdlets
      Break                     Exit a program loop
```

C

```
     Catch                      Handle a terminating error within a scriptblock
   Set-Location  cd/chdir/sl    Set the current working location
   Get-ChildItem   dir/ls/gci   Get child items (contents of a folder or registry
     Clear-Host    clear/cls    Clear the screen
     Clear-Item          cli    Remove content from a variable or an alias
       Get-Command       gcm    Retrieve basic information about a command
   Measure-Command              Measure running time
     Trace-Command              Trace an expression or command
        Add-Computer            Add a computer to the domain
 Checkpoint-Computer            Create a system restore point (XP)
     Remove-Computer            Remove the local computer from a workgroup or dom
    Restart-Computer            Restart the operating system on a computer
    Restore-Computer            Restore the computer to a previous state
       Stop-Computer            Stop (shut down) a computer
Reset-ComputerMachinePassword   Reset the machine account password for the comput
 Test-ComputerSecureChannel     Test and repair the secure channel to the domain
     Add-Content          ac    Add to the content of the item
     Get-Content cat/type/gc    Get content from item (specific location)
     Set-Content          sc    Set content in the item (specific location)
   Clear-Content         clc    Remove content from a file/item
         Continue               Skip just this iteration of a loop
     Get-Command         gcm    Get basic information about cmdlets
  Invoke-Command         icm    Run command
   Enable-ComputerRestore       Enable System Restore on a drive
   Disable-ComputerRestore      Disable System Restore on a drive
  Get-ComputerRestorePoint      Get the restore points on the local computer
 Test-Connection                Ping one or more computers
   ConvertFrom-CSV              Convert object properties (in CSV format) into CS
   ConvertTo-CSV                Convert .NET Framework objects into CSV variable-
   ConvertTo-Html               Convert the input into an HTML table
   ConvertTo-Xml                Convert the input into XML
   ConvertFrom-SecureString     Convert a secure string into an encrypted standar
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   ConvertTo-SecureString     Convert an encrypted standard string into a secur
   Copy-Item     copy/cp/ci   Copy an item from a namespace location
   Export-Counter             Export Performance Counter data to log files
      Get-Counter             Get performance counter data
   Import-Counter             Import performance counter log files
   Get-Credential             Get a security credential (username/password)
   Get-Culture                Get region information (language and keyboard lay
```

D

```
   Get-ChildItem   Dir/ls/gci Get child items (contents of a folder or registry
   Get-Date                   Get current date and time
   Set-Date                   Set system time on the host system
   Remove-Item  Del/erase/rd/rm/rmdir   Delete an item
   Compare-Object diff/compare   Compare the properties of objects
   Do                         Loop while a condition is True
     Get-DscConfiguration     Get the current config. of a node
     Get-DscLocalConfigurationManager   Get Local Config Manager settings
     Get-DscResource          Get Desired State Config. resources from a comput
     New-DSCCheckSum          Create checksum files for DSC docs/resources
   Start-DscConfiguration     Apply Desired State config to nodes
```

E

```
         End                  Function END block
      Get-Event               Get events in the PowerShell event queue
      New-Event               Create a new event
   Remove-Event               Delete events from the event queue
  Unregister-Event            Cancel an event subscription
     Wait-Event               Wait until a particular event is raised
   Clear-EventLog             Delete all entries from an event log
     Get-Eventlog             Get event log data (2003)
   Limit-EventLog             Limit the size of the event log
     New-Eventlog             Create a new event log and a new event source
   Remove-EventLog            Delete an event log
    Show-EventLog             Display an event log
   Write-EventLog             Write an event to an event log
     Get-WinEvent             Get event log data (Vista+)
   Get-EventSubscriber        Get event subscribers
Register-EngineEvent          Subscribe to PowerShell events
Register-ObjectEvent          Subscribe to .NET events
   Register-WmiEvent          Subscribe to a WMI event
   Get-ExecutionPolicy        Get the execution policy for the shell
   Set-ExecutionPolicy        Change the execution policy (user preference)
   Export-Alias         epal  Export an alias list to a file
   Export-Clixml              Produce a clixml representation of PowerShell obj
   Export-Console             Export console configuration to a file
   Export-Csv          epcsv  Export to Comma Separated Values (spreadsheet)
   Exit-PSSession       Exit  Exit PowerShell (or exit a script)
```

F

```
   -F operator                Format operator
   Unblock-File               Unblock files downloaded from the Internet
   ForEach-Object    foreach  Loop through each item in the pipeline ( % )
   ForEach                    Loop through each item in a collection
   ForEach method             Loop through each item in a collection
   For                        Loop through items that match a condition
   Format-Custom         fc   Format output using a customized view
   Format-List           fl   Format output as a list of properties, each on a
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   Format-Table          ft   Format output as a table
   Format-Wide           fw   Format output as a table listing one property onl
  Export-FormatData           Save formatting data from the current session
    Get-FormatData            Get the formatting data in the current session
```

G

```
   Get-Item              gi   Get a file/registry object (or any other namespac
   Get-ChildItem   dir/ls/gci Get child items (contents of a folder or registry
    Backup-GPO               Backup group policy objects (GPOs)
   Restore-GPO               Restore one or all GPOs from a GPO backup
    Import-GPO               Import Group Policy settings into a specified GPO
```

H

```
   Get-Help            help   Open the help file
     Add-History             Add entries to the session history
   Clear-History       clhy   Delete entries from the session history
     Get-History  history/h/ghy Get a listing of the session history
  Invoke-History     r/ihy    Invoke a previously executed Cmdlet
     Get-Host                 Get host information (PowerShell Version and Regi
   Clear-Host      clear/cls  Clear the screen
    Read-Host                 Read a line of input from the host console
   Write-Host                Write customized output to the host/screen
     Get-HotFix               Get Installed hotfixes
```

I

```
   if                        Conditionally perform a command
   Import-Clixml             Import a clixml file and rebuild the PS object
   Import-Csv         ipcsv  Take values from a CSV list and send objects down
   Invoke-Command            Run commands on local and remote computers
   Invoke-Expression    iex  Run a PowerShell expression
      Get-Item           gi   Get a file object or get a registry (or other nam
   Invoke-Item           ii   Invoke an executable or open a file (START)
      New-Item   md/mkdir/ni  Create a new item in a namespace
   Remove-Item  rm/del/erase/rd/ri/rmdir   Remove an item
      Set-Item           si   Change the value of an item
    Clear-ItemProperty  clp   Remove the property value from a property
     Copy-ItemProperty  cpp   Copy a property along with it's value
      Get-ItemProperty   gp   Retrieve the properties of an object
     Move-ItemProperty   mp   Move a property from one location to another
      New-ItemProperty        Set a new property
   Remove-ItemProperty   rp   Remove a property and its value
   Rename-ItemProperty  rnp   Renames a property at its location
      Set-ItemProperty   sp   Set a property at the specified location to a spe
```

J

```
       Get-Job          gjb   Get PowerShell background jobs that are running
   Receive-Job         rcjb   Get PowerShell background job results
    Remove-Job          rjb   Delete a PowerShell background job
     Start-Job         sajb   Start a PowerShell background job
      Stop-Job         spjb   Stop a PowerShell background job
      Wait-Job          wjb   Wait for a background job
   Job Trigger cmdlets       Get/Set Scheduled job triggers
```

K

```
   Stop-Process    kill/spps  Stop a running process
```

L

```
Update-List                  Add and remove items from a collection
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  Get-Location    pwd / gl   Get and display the current location
  Pop-Location        popd   Set the current working location from the stack
 Push-Location       pushd   Push a location to the stack
  Set-Location  cd/chdir/sl  Set the current working location
```

M

```
 Send-MailMessage            Send an email message
  Add-Member                 Add a member to an instance of a PowerShell objec
  Get-Member            gm   Enumerate the properties of an object
   Get-Module          gmo   Get the modules imported to the session
Import-Module         ipmo   Add a module to the session
   New-Module          nmo   Create a new dynamic module (only in memory)
Remove-Module          rmo   Remove a module from the current session
Export-ModuleMember          Export specific module members
  Move-Item      mv/move/mi  Move an item from one location to another
```

O

```
 Compare-Object diff/compare Compare the properties of objects
   Group-Object       group  Group objects that contain the same value
 Measure-Object              Measure the properties of an object
     New-Object              Create a new .Net object
  Select-Object      select  Select properties of objects
    Sort-Object        sort  Sort objects by property value
   Where-Object              Filter the objects passed along the command pipel
  Out-Default                Send output to default
  Out-File                   Send output to a file
  Out-GridView         ogv   Send output to an interactive table
  Out-Host              oh   Send output to the host
  Out-Null                   Send output to null
  Out-Printer           lp   Send the output to a printer
  Out-String                 Send objects to the host as strings
```

P

```
          Param              Script Parameters
          Pause              Pause and display the message "Press Enter to con
          Powershell         Launch a PowerShell session
  Convert-Path        cvpa   Convert a ps path to a provider path
     Join-Path               Combine a path and child-path
  Resolve-Path        rvpa   Resolves the wildcards in a path
    Split-Path               Return part of a path
     Test-Path               Return true if the path exists, otherwise return
Get-Pfxcertificate          Get pfx certificate information
   Pop-Location       popd   Set the current working location from the stack
  Push-Location      pushd   Push a location to the stack
        Process              Function PROCESS block
    Get-Process     ps/gps   Get a list of processes on a machine
  Debug-Process              Attach a debugger to a running process
  Start-Process  start/saps  Start one or more processes
   Stop-Process   kill/spps  Stop a running process
   Wait-Process              Wait for a process to stop
  Enable-PSBreakpoint  ebp   Enable a breakpoint in the current console
 Disable-PSBreakpoint  dbp   Disable a breakpoint in the current console
     Get-PSBreakpoint  gbp   Get the currently set breakpoints
     Set-PSBreakpoint  sbp   Set a breakpoint on a line, command, or variable
  Remove-PSBreakpoint  rbp   Delete breakpoints from the current console
     Get-PSDrive       gdr   Get drive information (DriveInfo)
     New-PSDrive   mount/ndr Create a mapped network drive.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   Remove-PSDrive       rdr   Remove a provider/drive from its location
      Get-PSProvider          Get information for the specified provider
      Set-PSdebug             Turn script debugging on or off
  Disable-PSRemoting          Disable remote session configuration on the local
   Enable-PSRemoting          Run PowerShell commands on remote computers
    Enter-PSSession  etsn     Start an interactive session with a remote comput
     Exit-PSSession  exsn     End an interactive session with a remote computer
   Export-PSSession  epsn     Import commands and save them in a PowerShell mod
      Get-PSSession   gsn     Get the PSSessions in the current session
   Import-PSSession  ipsn     Import commands from another session
      New-PSSession   nsn     Create a persistent connection to a local or remo
   Remove-PSSession   rsn     Close PowerShell sessions
    Disable-PSSessionConfiguration  Disable session configurations on the local
     Enable-PSSessionConfiguration  Enable session configurations on the local
        Get-PSSessionConfiguration  Get the registered PS session configuration
   Register-PSSessionConfiguration  Create and register a new PS session config
        Set-PSSessionConfiguration  Change properties of a registered session c
 Unregister-PSSessionConfiguration  Delete registered PS session configuration
        New-PSSessionOption         Advanced options for a PSSession
   Add-PsSnapIn        asnp   Add snap-ins to the console
   Get-PsSnapin        gsnp   List PowerShell snap-ins on this computer
   Remove-PSSnapin     rsnp   Remove PowerShell snap-ins from the console
```

Q

```
   Quest AD cmdlets           Read and write to Active Directory
```

R

```
   Get-Random                 Get a random number
   Read-Host                  Read a line of input from the host console
   Remove-Item  rm/del/erase/rd/ri/rmdir   Remove an item
   Rename-Item      ren/rni   Change the name of an existing item
   Rename-ItemProperty        Rename a property of an item
   Return                     Exit the current scope, (function, script, or scr
   Run/Call             &     Run a command (call operator)
```

S

```
   Scheduler cmdlets          Get/Set scheduled jobs
   Select-Object     select   Select properties of objects
   Select-XML                 Find text in an XML string or document
   Send-MailMessage           Send an email message
       Get-Service      gsv   Get a list of services
       New-Service            Create a new service
   Restart-Service            Stop and then restart a service
    Resume-Service            Resume a suspended service
       Set-Service            Change the start mode/properties of a service
     Start-Service     sasv   Start a stopped service
      Stop-Service     spsv   Stop a running service
   Suspend-Service            Suspend a running service
   Sort-Object         sort   Sort objects by property value
   Set-StrictMode             Enforce coding rules in expressions & scripts
   Start-Sleep        sleep   Suspend shell, script, or runspace activity
   Switch                     Multiple if statements
   ConvertFrom-StringData     Convert a here-string into a hash table
   Select-String              Search through strings or files for patterns
```

T

```
   Tee-Object           tee   Send input objects to two places
   New-Timespan               Create a timespan object
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   Trace-Command              Trace an expression or command
   Get-Tracesource            Get components that are instrumented for tracing.
   Set-Tracesource            Trace a PowerShell component
     Start-Transaction        Start a new transaction
  Complete-Transaction        Commit the transaction
       Get-Transaction        Get information about the active transaction
       Use-Transaction        Add a command or expression to the transaction
      Undo-Transaction        Roll back a transaction
   Start-Transcript           Start a transcript of a command shell session
    Stop-Transcript           Stop the transcription process
         Trap                 Handle a terminating error
         Try ... Catch        Handle a terminating error within a scriptblock
     Add-Type                 Add a .NET Framework type to a PowerShell session
   Update-TypeData            Update extended type configuration
```

U

```
   Get-Uiculture              Get the ui culture information
       Unblock-File           Unblock files downloaded from the Internet
   Get-Unique           gu    Get the unique items in a collection
    Update-Formatdata         Update and append format data files
    Update-Help               Download and install help files
    Update-Typedata           Update the current extended type configuration
```

V

```
   Clear-Variable       clv   Remove the value from a variable
     Get-Variable        gv   Get a PowerShell variable
     New-Variable        nv   Create a new variable
  Remove-Variable        rv   Remove a variable and its value
     Set-Variable    set/sv   Set a variable and a value
```

W

```
Checkpoint-WebApplicationMonitoring Create a checkpoint for an IIS web app
 Get-WebApplicationMonitoringStatus Get the monitoring status of web apps
 New-WebServiceProxy          Create a Web service proxy object
   Where-Object      where/?  Filter input from the pipeline
   Where method               Filter objects from a collection
   While                      Loop while a condition is True
 Add-WindowsFeature           Install roles, role services, and features
 Get-WindowsFeature           Retrieve roles, role services, and features
 Remove-WindowsFeature        Remove roles, role services, and features
   Write-Debug                Write a debug message to the host display
   Write-Error                Write an object to the error pipeline
   Write-Host                 Display text on screen
   Write-Output   write/echo  Write an object to the pipeline
   Write-Progress             Display a progress bar
   Write-Verbose              Write a string to the host's verbose display
   Write-Warning              Write a string in reverse video to the display
    Set-WmiInstance           Create or update an instance of an existing WMI c
 Invoke-WmiMethod      iwmi   Call WMI methods
    Get-WmiObject      gwmi   Get WMI class information
 Remove-WmiObject      rwmi   Delete an instance of a WMI class
    Connect-WSMan             Connect to the WinRM service on a remote computer
 Disconnect-WSMan             Disconnect from the WinRM service on a remote com
       Test-WSMan             Test if a computer is setup to receive remote com
  Invoke-WSManAction          Invoke an action on a specified object
 Disable-WSManCredSSP         Disable Credential Security Service Provider (SSP
  Enable-WSManCredSSP         Enable Credential SSP authentication
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    Get-WSManCredSSP          Get the Credential SSP configuration
  New-WSManInstance           Create a new instance of a management resource
  Get-WSManInstance           Display management information (XML or value)
  Set-WSManInstance           Modify the management information related to a re
Remove-WSManInstance          Delete a management resource instance
  Set-WSManQuickConfig        Configure the local computer for remote managemen
  New-WSManSessionOption      Options for WSMan commands
```

Z

```
  Zipfile                     Compress or Extract zip files
      #                       Comment / Remark
      . (source)              Run a command script in the current shell
      & (call)                Run a command, script or function
      %                       Alias for ForEach-Object
      --%                     Stop parsing input
      ?                       Alias for Where-Object
      $variable = "value"     Define a variable  also: ${n!a#me} = "value"
      $_                      The current pipeline object
      @(...)                  Force an expression to be evaluated as an array
```

For Details see Powershell-Command Reference

© 2023 Theo Gottwald * https://www.smart-package.com

3.73.2 PWS. - Powershell-Script Start

MiniRobotLanguage (MRL)

PWS. Command

Powershell Script Start

Intention

This command is used to start a block with VB-Script. It is always used paired with `VBE.`

The statements between a `VBS.` and a `VBE.` are ignored by the SPR-Preprocessor. They are also ignored by the "Pretty-Printer" in the Editor.

`VBS.` and `VBE.` enable you, to include a VB-Script or Java-Script Code inline into the SPR-Script. You can use it to check for System-Objects (See Sample 2 below), or you can use it to call WMI-Functions.

You can also transfer Variable-Content into `VB-Script`-Variables (see example 1 below). And use them in your VB-Script.

And with `VBE.` you can get a single result value from the VB-Script back into your SPR-Script (see example 2 below).

This Sample-Script shows how to get Parameters into the Powerscript and how to get the result out to the SPR.

```
' This is the Powershell-Call with the Parameters
PWS.||Theo|Peter und Franz|Ralf|Heribert|Jürgen

param ($var1, $var2, $var3, $var4, $var5)
Write-host "var1 =" $var1
Write-host "var2 =" $var2
Write-host "var3 =" $var3
Write-host "var4 =" $var4
Write-host "var5 =" $var5

PWE.$$OUT
' Display the Output from the Powershell in a Messagebox
MBX.$$OUT
ENR.
```

Syntax

PWS.P1|P2[|P3...|Px]

...

*© 2023 Theo Gottwald * https://www.smart-package.com*

PWE.

Parameter Explanation for VBS

P1 - (optional,in) Addition to Powershell call. Could be something like
"-ExecutionPolicy Undefined ". Normally leave just empty.

P2 - (optional,in) Reserved for STDIN. Currently unused, leave empty.

P3-Px - (in) - these are the Parameters to transfer into the Powershell-Script. You can use up to 30 KB for all Parameters together.
In case you need more, use temporary files.

Example

```
'*********************************************
' PWS./PWE. - Sample
' with Parameters
'*********************************************
' This is the Powershell-Call with the Parameters
PWS.||Theo|Peter und Franz|Ralf|Heribert|Jürgen
param ($var1, $var2, $var3, $var4, $var5)
Write-host "var1 =" $var1
Write-host "var2 =" $var2
Write-host "var3 =" $var3
Write-host "var4 =" $var4
Write-host "var5 =" $var5
PWE.$$OUT
' Display the Output from the Powershell in a Messagebox
MBX.$$OUT
ENR.
```

```
'*********************************************
' PWS./PWE. - Sample
' Get and say the BIOS-Version 2 times.
'*********************************************
PWS.
param( [string]$strComputer = "." )
$colItems = get-wmiobject -class "Win32_BIOS" -namespace
"root\CIMV2" -computername $strComputer
foreach ($objItem in $colItems)
          {
write-host "Name : "       $objItem.Name
write-host "Version : "    $objItem.Version
write-host "BiosVersion: " $objItem.BiosVersion
write-host "SMBIOSBIOS Version : " $objItem.SMBIOSBIOSVersion
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
write-host "Caption: "      $objItem.Caption
write-host "Manufacturer : " $objItem.Manufacturer

Add-Type -AssemblyName System.speech
$speak = New-Object System.Speech.Synthesis.SpeechSynthesizer
$speak.Speak($objItem.SMBIOSBIOSVersion)
$speak.Speak($objItem.SMBIOSBIOSVersion)
}
PWE.$$OUT
MBX.$$OUT
ENR.
```

Remarks

-

Limitations:

The Powershell-Scripts will be written to the harddisk prior to execution. Therefore they are not permanently crypted together
with the SPR-Script when using the `'#CRY:`-Option.

There are no Limitations about Multithreading here.

See also:

- PWE. - Powershell-Script End 2877
- VBE. - VB-Script End

© 2023 Theo Gottwald * https://www.smart-package.com

3.73.3 PWE. - Powershell-Script End

Powershell Inline

PWE. - Powershell-Script-End Previous 2866 Top 41 Next 2874

MiniRobotLanguage (MRL)

PWE. Command

Powershell Script End

Intention

This command is used to end a block with VB-Script. It is always used paired with `VBS.`
The statements between a `VBS.` and a `VBE.` are ignored by the SPR-Preprocessor. They are also ignored by the "Pretty-Printer" in the Editor.

`VBS.` and `VBE.` enable you, to include a VB-Script or Java-Script Code inline into the SPR-Script. You can use it to check for System-Objects (See Sample 2 below), or you can use it to call WMI-Functions.

You can also transfer Variable-Content into `VB-Script`-Variables (see example 1 below). And use them in your VB-Script.

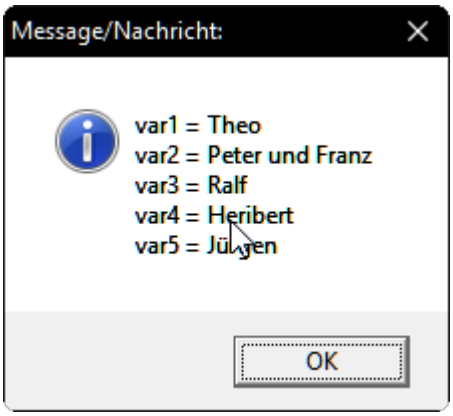
And with `VBE.` you can get a single result value from the VB-Script back into your SPR-Script (see example 2 below).

This Sample-Script shows how to get Parameters into the Powerscript and how to get the result out to the SPR.

```
' This is the Powershell-Call with the Parameters
PWS.||Theo|Peter und Franz|Ralf|Heribert|Jürgen

param ($var1, $var2, $var3, $var4, $var5)
Write-host "var1 =" $var1
Write-host "var2 =" $var2
Write-host "var3 =" $var3
Write-host "var4 =" $var4
Write-host "var5 =" $var5

PWE.$$OUT
' Display the Output from the Powershell in a Messagebox
MBX.$$OUT
ENR.
```

The Output from the Script looks like this:

 © 2023 Theo Gottwald * https://www.smart-package.com



Syntax

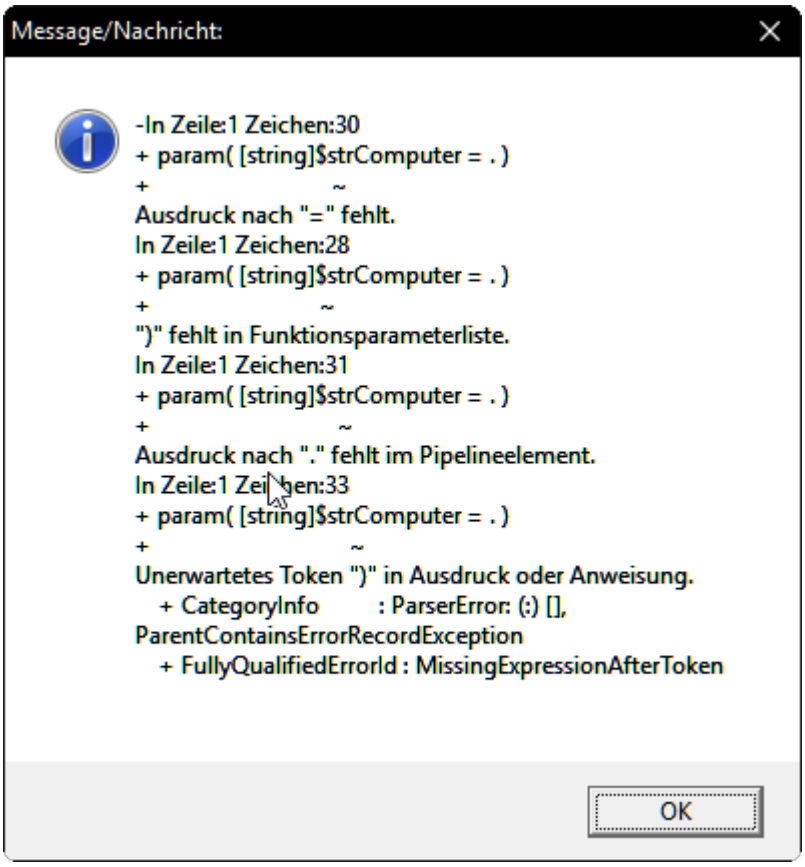
```
PWS.
...
PWE.[P1][|[P2]
```

Parameter Explanation for VBE

P1 **-** (out,optional) - will hold to the result of the Powerscript (STDOUT).
P2 **-** (out,optional) - will Get some Error-Information as far as its not in **P1**.
Like Commandline-Errors.

This is a Errormessage like you can get it in **P2**.

© 2023 Theo Gottwald * https://www.smart-package.com



Example

```
'*********************************************
' PWS./PWE. - Sample
' with Parameters
'*********************************************
' This is the Powershell-Call with the Parameters
PWS.||Theo|Peter und Franz|Ralf|Heribert|Jürgen
param ($var1, $var2, $var3, $var4, $var5)
Write-host "var1 =" $var1
Write-host "var2 =" $var2
Write-host "var3 =" $var3
Write-host "var4 =" $var4
Write-host "var5 =" $var5
PWE.$$OUT
' Display the Output from the Powershell in a Messagebox
MBX.$$OUT
ENR.



'*********************************************
' PWS./PWE. - Sample
' Get and say the BIOS-Version 2 times.
'*********************************************
PWS.
param( [string]$strComputer = "." )
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$colItems = get-wmiobject -class "Win32_BIOS" -namespace
"root\CIMV2" -computername $strComputer
foreach ($objItem in $colItems)
          {
write-host "Name : "        $objItem.Name
write-host "Version : "     $objItem.Version
write-host "BiosVersion: " $objItem.BiosVersion
write-host "SMBIOSBIOS Version : " $objItem.SMBIOSBIOSVersion
write-host "Caption: "      $objItem.Caption
write-host "Manufacturer : " $objItem.Manufacturer

Add-Type -AssemblyName System.speech
$speak = New-Object System.Speech.Synthesis.SpeechSynthesizer
$speak.Speak($objItem.SMBIOSBIOSVersion)
$speak.Speak($objItem.SMBIOSBIOSVersion)
}
PWE.$$OUT
MBX.$$OUT
ENR.
```

Remarks

-

Limitations:

The Powershell-Scripts will be written to the harddisk prior to execution. Therefore they are not permanently crypted as is
the SPR-Script, when using the `'#CRY:`-Option.

There are no Limitations about Multithreading here.

See also:

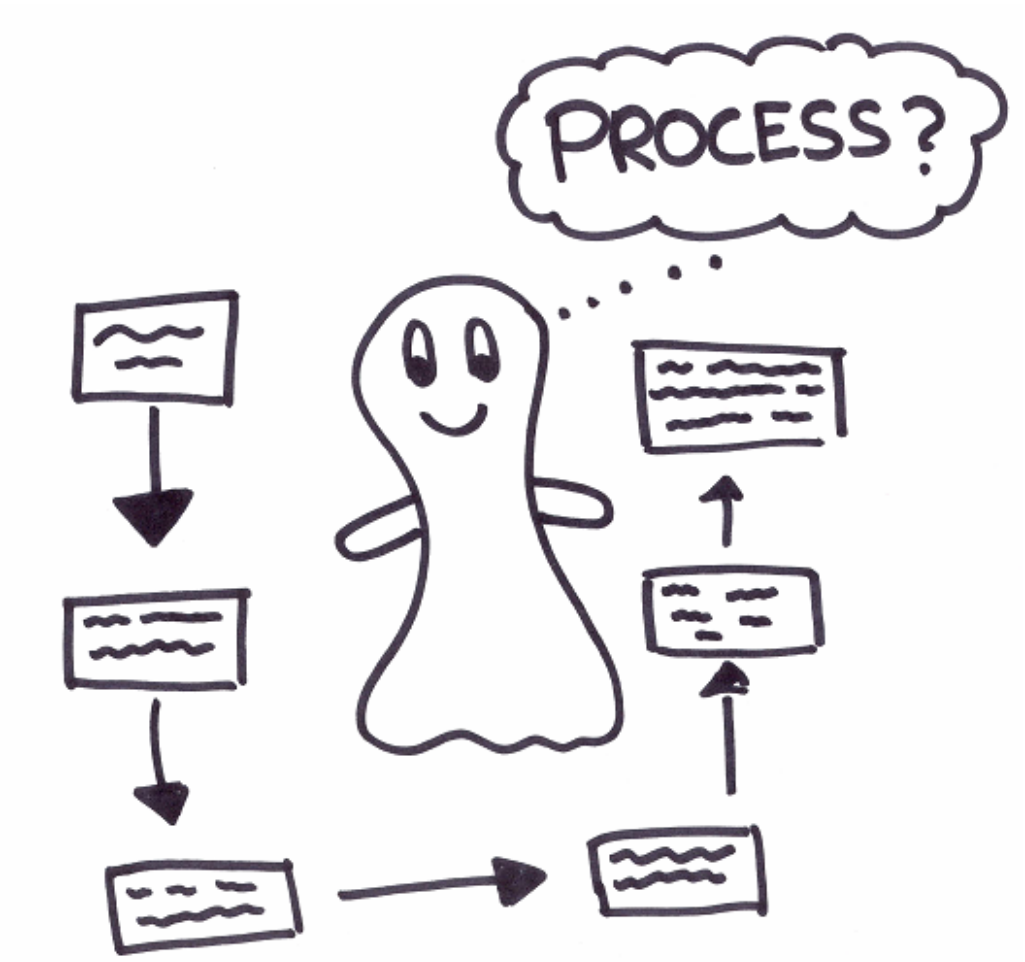
- PWS. - Powershell-Script Start 2874
- VBS. - VB-Script Start

© 2023 Theo Gottwald * https://www.smart-package.com

3.74 Process Operations

3.74.1 ! Smarty's Process Commands

Smarty's Process Commands



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.74.2 CPR. - Close-PRocess

MiniRobotLanguage (MRL)

CPR. Command

Close Process

Intention

Close Process by Process-ID (#pid#). This command will end a running process.
This is done in 2 stages.
First the command first tries to send a "Close-Message" to the Process-Mainwindow.
It then waits for a specified time.
Second, if the process does not end within the given waiting time, (if it hangs), the process is forcefully closed.

CPR. can be use in three ways:

```
' This will close the process from the last localized window
CRP.

' This will close process with PID in $$PID and a
' waiting time of 1 second (default waiting time).
' CPR.$$PID

' This will close the process from the last localized window
' with a waiting time of $$WAT milliseconds.
' CPR.|$$WAZ


' This will close the process with the given PID ($$PID)
' with a waiting time of $$WAT milliseconds.
' CPR.$$PID|$$WAZ

' This will immediately, forcefully close the process
' with the given PID ($$PID)
' CPR.$$PID|0
```

Syntax

CPR.P1|P2

Parameter Explanation

P1 - PID of the process to close

P2 - Waiting time in ms before forcefully closing the process.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
' CPR.-Demo
'**********************************
DBM.2
' Get the PID using IEP.command
IEP.Testapp_01.exe
' Close the App with 5 Seconds waiting time
  CPR.#pid#|5000
ELS.
  MBX. That Application is not running!
EIF.
ENR.
```

Remarks

Tested: XP, Win7/x64

Limitations:

-

See also:

- 1.7. Application and PID 137
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- CLW. - Close-Window 3592
- CAW. - Close-All-Windows 3586
- CTR. - Close-ThRead 2884
- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904
- GDF. - Get-Data-From 2889
- GPU. - Get Process User 2898

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.3 CTR. - Close-ThRead

MiniRobotLanguage (MRL)

CTR. Command

Close Thread

Intention

Close Thread. Parameter Thread-ID (#tid#) is optional.
Otherwise #tid# is taken.

Will end a running Thread immediately and forcefully

Syntax

CTR.[P1]

Parameter Explanation

P1 – ThreadID (#tid#)

Example

```
'**********************************
' CTR.-Demo
'**********************************
' Internet Explorer must be started
: §§LNK=http://www.fa2.de/
EXS.iexplore
STW.c|ieframe
' #tid# is set automatically by the STW.
' Be shure to use VAR. and not : because of
Specialfolder/Systemvariable-Substitution
VAR.§§TID=#tid#
DBP.Thread-ID is: §§TID
SWP.0,0
DBP.We'll kill the IE-Thread in 3 Seconds.
PAU.3
CTR.§§TID
MBX.Ready
END.
```

Remarks

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

Limitations:

-

See also:

- 1.7. Application and PID 137
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- CLW. - Close-Window 3592
- CAW. - Close-All-Windows 3586
- CTR. - Close-ThRead 2884
- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904
- GDF. - Get-Data-From 2889
- GPU. - Get Process User 2898

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.4 FPI. - Filepath to PID

MiniRobotLanguage (MRL)

FPI. Command

File-Path to PID and reverse

Intention

If you specify a File-path, the command will give you the corresponding PID 137.

```
VAR.$$FIL=?exepath\Sample Apps\TestApp_01.exe
FPI.$$FIL|$$PID
DBP.PID is:$$PID
```

You can provide a File-path or not. And just specify a filename. In all cases, you can use 3.2 Standard-Search Pattern 341, see this valid example:

```
VAR.$$FIL=Testapp&AND:01.exe
FPI.$$FIL|$$PID
DBP.$$PID
ENR.
```

In case you prefer to use the good old WIldcards, ***** and **?**, do it like this:

```
VAR.$$FIL=&WILDC:Testapp*01.exe
FPI.$$FIL|$$PID
DBP.$$PID
ENR.
```

If there are multiple Instances of this program running, you can additionally specify a Instance number:

```
VAR.$$FIL=?exepath\Sample Apps\TestApp_01.exe
' Will get the PID of the second Instance of that program
FPI.$$FIL|2|$$PID
DBP.PID is:$$PID
```

You can also go the opposite direction and get the File-Path of a PID. Like this:

```
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Static|This is a Text in the TestApp.
VAR.$$PID=#pid#
FPI.>$$PID|$$FIL
DBP.$$FIL
ENR.
```

This example will Print the Path to the Sample Application.

Note: This command duplicates a function from `GDF`. We made this as extra command, to make working with processes easier to use and understand.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

FPI.P1[|P2][|P3]

Parameter Explanation

P1 - Filename (with or without path). If it starts with a ">" then
a numeric **PID** must follow. You can use variables and also a
formula in (..).

P2 - ***(optional)*** Variable for the result, if two parameters are specified. If three parameters are given, this is the number of the executable to search for, and P3 is the variable for the result.

P3 - ***(optional)*** Variable for the result.

Example

```
'**********************************
' FPI.-Demo
'**********************************
' STW. will automatically set #pid#
' to the PID of the program that created the window.
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Static|This is a Text in the TestApp.
VAR.$$PID=#pid#
' Get the Path of a program
FPI.>$$PID|$$FIL
DBP.$$FIL
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- **1.7. Application and PID** 137
- GPU. - Get Process User 2898
- GPI. - Get Process Information 2893
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- GDF. - Get-Data-From 2889
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.5 GDF. - Get-Data-From

MiniRobotLanguage (MRL)

GDF. Command

Get Data From

Intention

When your scripts get more advanced, then its time to understand this command. Understanding how to use GDF. can add wings to your scripting abilities.

Be sure that you have read and understand the chapter about 1.7. Application and PID 137.

The most often used case for GDF. is to find out which process opened a window. Here is Sample Script that will show the executable behind the localized window,

```
STW.c|Chrome_WidgetWin_1
GDF.af|$$EXE
MBX.$$EXE
```

GDF. converts your scripting resources between each other. Just give it what you have, and you get what you need. Whatever you have, window handles, PID's or DLL names.
Feed it to GDF. and get what ever you need.

What is the EXE-Filepath od file with PID 123?
What is the PID of the Program that own the window we have localized?
What is the EXE-File-Path of the program that has the window handle number 12345?
What is the window handle of a program with PID 1234?

GDF. is the answer to all these questions. GDF. uses the prototype parameter system 73.
Be sure that you have read that chapter and understood how it works..

Imagine a box where you put in any information you have (PID, Path, handle ...) and you can get out any information you want. That is GDF.

GDF. deals with: Window-Handle, Process-ID and EXE-File-Path. The window handle must be a valid handle of an existing window. The file-path must be from an process or module (DLL) that is currently running. When asking for a DLL, the filename must end with ".dll". You can use the 3.2 Standard-Search Pattern 341 for the filename on the source side. You can add a ">2" to the filename if you want the second fitting result.Of course any other number after the ">" will also do what you expect.

GDF. takes what you provide and calculates the rest you need. It can calculate multiple results at one time at the destination side. For example:

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$FIL=?exepath\MR01.exe
GDF.fph|$$FIL|$$PID|$$HWN
MBX.$$FIL - $$PID - $$HWN
ENR.
```

The first letter is always the source. Thats what you put in. All following letters in the prototype field are destination prototypes. In this example we have provided the "f" as source. We have given the filepath. `?exepath\` is the 3.3 Specialfolders 345 shortcut to the installation directory. Therefore `$$FIL` leads to the actual running robot itself. This is what we put in. And the following two prototypes, these are "p" and "h" is what we expect to get back. "p" - is the Process ID (PID). And "h" is the window handle - or zero if the process has no window. For this we add two variables to be filled with these values.
`$$PID` for the PID and `$$HWN` for the window handle.

`GDF.` works with any process in the Task-Manager, no matter if that process has a window or not. If its running on a server OS, the program must start with sufficient rights to collect these information.

If you add prototypes and you do not give variables to put them in, the result will be placed on the actual stack.

If you want the PID from the actual localized window, you just write:

```
' We want the PID of the actual localized window(-prozess)
' into $$PID
GDF.ap|$$PID


' You can take a PID and localize a window with it!
: $$PID=1234
GDF.pa|$$PID
```

Syntax

GDF.P1|P2[|P3]..[|PX]

Parameter Explanation

Possible prototype-letters **for source:**

```
' a - the actual lokalized window is been used,
'     needs no input parameter therefore.
'
' f - filepath (Exe-Path or DLL Path), needs Parameter in
source.
'     When used in source, you can add a ">" Sign to tell
which of
'     many files with the same name you want.
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' h - window handle (Child- or Top-Window), needs input
parameter.
'
' p - Process ID (PID), needs input parameter.
'
' q - Process ID (PID) is taken from #pid#,
'     needs no parameter therefore.
```

Possible prototype-letters **for destination:**

```
' a - the window is been localized internally
'     like with SCW.h|$$HWN.
'
' f - filepath (Exe-Path or DLL Path). You can give a
parameter
'     with a variable then the variable will be filled with
'     the filepath. If you omit the variable, the filepath is
'     been place on the actual stack.
'
' h - window handle (Child- or Top-Window), you can
'     give a parameter with a variable. In that case
'     the variable will be filled with the window handle.
'     If you omit the variable, the window handle
'     is been place on the actual stack.
'
' p - Process ID (PID), you can give a parameter with
'     a variable. In that case the variable will be filled
'     with the Process ID (PID). If you omit the variable,
'     the Process ID (PID) is been place on the actual stack.
'
' q - Process ID (PID) that is actually in #pid#, needs no
'     parameter. If used as destination, #pid# is set with
'     the result.
```

P1 - Prototype field. First letter is source prototype, all following letters are destination-prototypes

P2 ... PX – Input- and output variables according to prototype

Example

We'll look at some scenarios when using `GDF`. Think there are two application, named "x64.exe".We want the the `PID`'s int two variables `$$PIA` and `$$PIB`.
To do this we need the information, that we can add a ">" Sign and a number behind a filename-parameter. This way we can decide which of multiple executable with the same name we want to get the PID for.

Lets go. Here is our Script.

© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$FIA=x64.exe>1
VAR.$$FIB=x64.exe>2
GDF.fp|$$FIA|$$PIA
GDF.fp|$$FIB|$$PIB
DBP.$$PIA and $$PIB
MBX.Ready
```

and this is the result in the Editor-Debug Area:

```
[12:20:39] MRE: File saved as: 'C:\Program Files (x86)\MiniRc
[12:20:39] MRE: Robot start: IEP.mrt (On line #1)
[12:20:39] [15] 2768 and 2084
```

We got the PID's of both executables. No matter that they had the same filename. The [15] is from the `DBP.` command. It shows us that it was "Line 15" in the script that sent us this message. This information is useful for debugging. You will immediately know which `DBP.` sent you a message.

Remarks

x64 tested.

Limitations:

-

See also:

- 1.7. Application and PID 137
- GPU. - Get Process User 2898
- GPI. - Get Process Information 2893
- IEP. / NEAP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.6 GPI. - Get Process Information

MiniRobotLanguage (MRL)

GPI. Command

Get Process Information

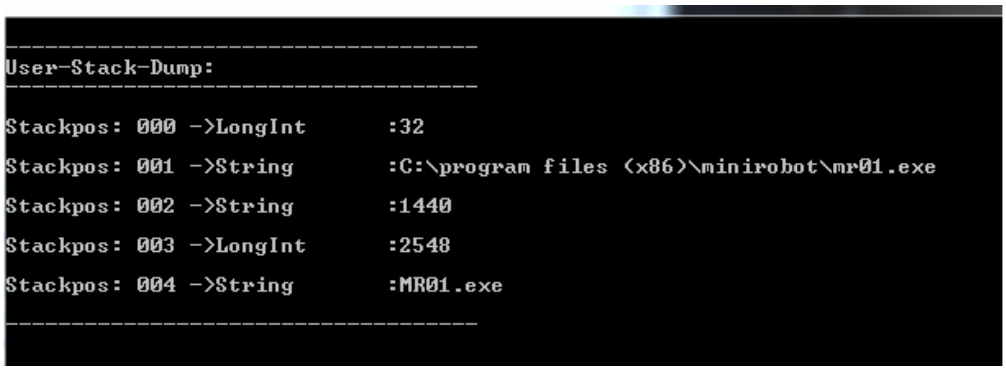
Intention

This command will provide you with informations about a specified process.

If you just call it without any parameters, it will examine the robot program itself. The code is very simple:

```
GPI.
```

And as result, the stack will look like this:



The order of the information is:

- 32 - shows you the process is an x32 process. if it would be a 64, it would be a x64 process.
- c.\ ... - thats the complete path of the examined process
- 1440 - this is the PID of the parent process. This is the process that has started the executable.
- 2548 - this is the PID of the examined process itself
- MR01.exe - this is just the process name. Like you see it in the task manager.

Now lets examine another process.The code is:

```
STS.CLEAR
STS.LOCAL

GPI.x64.exe
MBX. Process ID is: #pid#

ENR.
```

Now this is the stack:

 © 2023 Theo Gottwald * https://www.smart-package.com

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt       :64
Stackpos: 001 ->String        :C:\users\admin\desktop\x64.exe
Stackpos: 002 ->String        :1804
Stackpos: 003 ->LongInt       :2768
Stackpos: 004 ->String        :x64.exe

------------------------------------
```

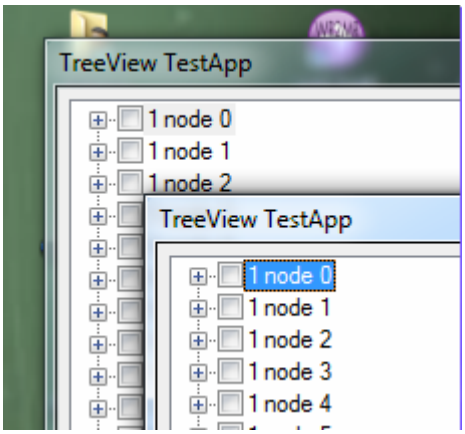
Again, the order of the information is:

- 64 - shows you the process is an x64process.
- c.\ ... - thats the complete path of the examined process
- 1804 - this is the PID of the parent process. This is the process that has started the executable. In fact its the "explorer.exe", see picture below.
- 2768 - this is the PID of the examined process itself
- x64.exe - this is just the process name. Like you see it in the task manager.

explorer.exe	1804	Admin	02	12.852 K	Windov
MRE.exe *32	1440	Admin	00	8.768 K	Mini-Ro
process.exe *32	1172	Admin	00	620 K	process

If you have multiple EXE running with the same name, you can specify a number.

For example, if you have two equal x64.exe applications.




```
STS.CLEAR
STS.LOCAL
GPI.x64.exe|1
GPI.x64.exe|2
DMP.6
MBX.Sample 1

You will get this stack:
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->LongInt      :64
Stackpos: 001 ->String       :C:\users\admin\desktop\x64.exe
Stackpos: 002 ->String       :1804
Stackpos: 003 ->LongInt      :2084
Stackpos: 004 ->String       :x64.exe
Stackpos: 005 ->LongInt      :64
Stackpos: 006 ->String       :C:\users\admin\desktop\x64.exe
Stackpos: 007 ->String       :1804
Stackpos: 008 ->LongInt      :2768
Stackpos: 009 ->String       :x64.exe

-----------------------------------
```

The Systemvariable **#pid#** is always set to the EXE's Process-ID. Please note that **#tid#** is always set to zero.

You can specify a full path, or only the filename or a part of it.

If a process is found, valuable information is placed on the actual stack. These include the full name and path of the executable.

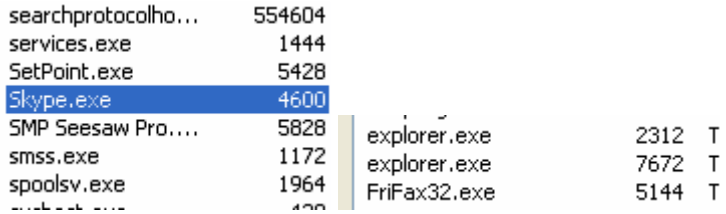
Below are two snapshots of the windows-taskmanager. The executable "Skype.exe" has the PID (process-id) 4600. Its on the stack on position 02 in the picture above.

The 2312 which is on position 01 (picture above) is the parent-process, that is the process which has started "skype.exe".

In the task manager, we can see that this is the PID of the windows-explorer. This way you can find out which process has started another process.

If you give GPI. a variable as third parameter, then the stack will not get the full path of the found file. You will find it in the given variable instead.

Please note that the stack organization (see 2.4 The global and local Stack 369) as Que or as Stack will influence the sequence in which the items appear on the stack.



You can also use `GPI.` to get information about a process, if you just have the PID. For this you have two choices. Here is the first:

```
GPI.>4600
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt       :32
Stackpos: 001 ->String        :C:\programme\skype\phone\skype.exe
Stackpos: 002 ->LongInt       :4600
------------------------------------
```

Alternative there is a slower, yet more in depth method. Like this:

```
GPI.<4600
```

```
------------------------------------
User-Stack-Dump:
------------------------------------

Stackpos: 000 ->LongInt       :32
Stackpos: 001 ->LongInt       :4600
Stackpos: 002 ->String        :C:\programme\skype\phone\skype.exe
Stackpos: 003 ->String        :skype.exe
Stackpos: 004 ->String        :C:\windows\explorer.exe
Stackpos: 005 ->String        :2312
------------------------------------
```

The stack shows you the following information:

- 32 - its a x32 process. Otherwise it would be a 64 if its a x64 process.
- 4600 - the PID of the examined process
- c:\... the full path of the process.
- skype.exe - just the name of the process
- the name and .path of the parent process that started this process
- the PID of the parent process

You can read more details on how Applications and PID's in the chapter 1.7. Application and PID 137.

Syntax

GPI.P1[|P2][|P3]

Parameter Explanation

There are three Usage Mods.
"process name mode"
P1 - Variable or Exe-Filename, with or without path. Pattern can be used.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **-** Variable/Number/Formula - Number of result (take fist, second etc. fitting result)
P3 **-** Variable if given gets the full path of the found exe

"PID Info Mode 1": Get basic infos about a process.
P1 **-** Starting with a ***">"***-Sign this indicates that we are in this Mode. Now follows a PID. Description see above.

"PID Info Mode 2": Get detailed Infos about a process.
P1 **-** Starting with a ***"<"***-Sign this indicates that we are in this Mode. Now follows a PID. Description see above.

Example

```
'***********************************
' GPI.-Demo
'***********************************
STS.CLEAR
STS.LOCAL

GPI.
DMP.6
MBX.Sample 1
```

Remarks

x64 tested.

Limitations:

no Limitations.

See also:

- 1.7. Application and PID 137
- GPU. - Get Process User 2898
- FEP. - For-Each-Process 932
- GDF. - Get-Data-From 2889
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.7 GPU. - Get Process User

MiniRobotLanguage (MRL)

GPU. Command

PID to User-Login Information (User-Name and Domain)

Intention

If you specify a PID 137, the command will provide you with the user-name and Domain information of the user that started the program. This command can be used to find out, which user has locked a workstation.

```
VAR.$$FIL=?exepath\Sample Apps\TestApp_01.exe
FPI.$$FIL|$$PID
DBP.PID is:$$PID
' Now we get the Username and Domain from the user that
started
' the program with that PID
GPU.$$PID
' Its all on the stack
STS.DUMP
MBX.!
ENR.
```

You can provide variables then the results will be in these variables. See this example:

```
VAR.$$FIL=Testapp&AND:01.exe
FPI.$$FIL|$$PID
GPU.$$PID|$$USR|$$DOM
DBP. Username is: $$USR
DBP.Domain is: $$DOM
ENR.
```

Hints:

1. Use the `FPI.`-command to get the `PID` from a filename.
2. To find out if a user is loged in and has just "Locked" his workstation, you can get the `PID` from "Explorer.exe", using `FPI.` and if you get one, find out which user is still logged on using `GPU.`.

Syntax

GPU.P1[|P2][|P3]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - PID of the process to analyze

P2 - (optional) Variable for the resulting Username. If omitted,
TOS is used.

P3 - (optional) Variable for the resulting Domain information.
If omitted, TOS is used.

Example

```
'**********************************
' GPU.-Demo
'**********************************
' STW. will automatically set #pid#
' to the PID of the program that created the window.
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Static|This is a Text in the TestApp.
VAR.$$PID=#pid#
' Get the Path of a program
FPI.>$$PID|$$FIL
GPU.$$PID|$$USR|$$DOM
DBP. Username is: $$USR
DBP.Domain is: $$DOM
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.7. Application and PID 137
- FPI. - Filepath to PID 2886
- GPI. - Get Process Information 2893
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- GDF. - Get-Data-From 2889
- WPT. - Wait-Process-Termination 2913

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.8 IEP. / NEP. - If-Exists-Process

MiniRobotLanguage (MRL)

IEP. Conditional Statement

If Exist Process

Intention

Conditional Statement. Test if a specified process is running.

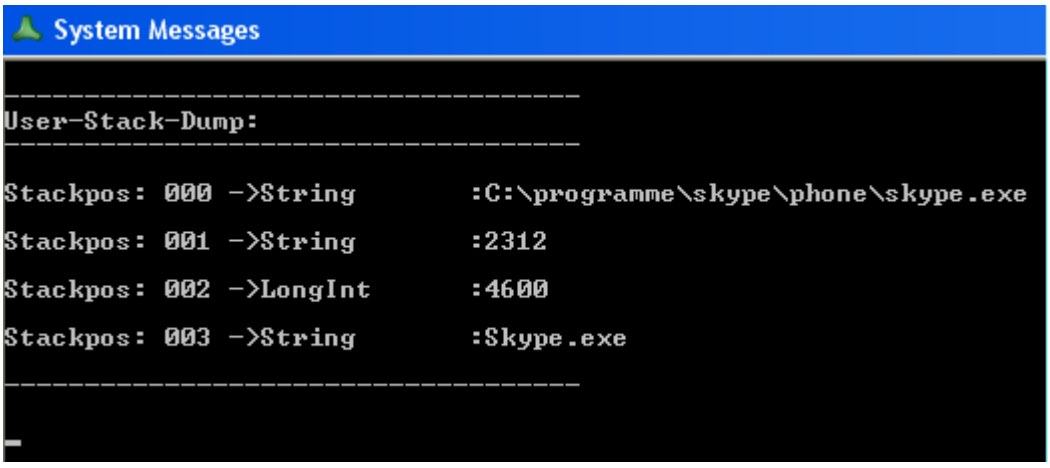
```
IEP.x64.exe
   MBX. Process ID is: #pid#
EIF.
```

If you have multiple EXE running with the same name, you can specify a number. If the EXE is found, the #pid# is set to the EXE's Process-ID. Please note that #tid# is always set to zero.

You can specify a full path, or only the filename or a part of it. If you search for a dll, then the extension ".dll" must be included in the pattern.

You can use `NEP.`, the negative Form of this command (`NEP.` - Not Exists Process).

If a process is found, valuable information is placed on the actual stack. These include the full name and path of the found executable.

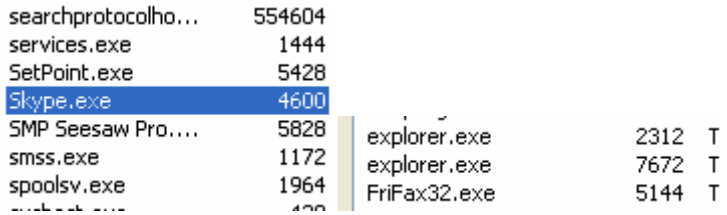


Below are two snapshots of the windows-taskmanager. Skype.exe has the PID (process-id) 4600. Its on the stack on position 02 in the picture above.
The 2312 which is on position 01 (picture above) is the parent-process, that is the process which has started skype.exe.
In the task manager, we can see that this is the PID of the windows-explorer. This way you can find out which process has started another process.

*© 2023 Theo Gottwald * https://www.smart-package.com*

If you give IEP. a variable as third parameter, then the stack will not get the full path of the found file. You will find it in the given variable instead.

Please note that the 2.4 The global and local Stack 369 as Que or as Stack will influence the sequence in which the items appear on the stack.



You can also use IEP. to verify, if a process with a specified **PID is alive** or not.
To do this, the Syntax is:

```
STS.GLOBAL
IEP.>1234
   MBX.PID 1234 (Filename=$$000) is alive!
EIF.
```

You can use IEP. also to check if one or one of many PID's is alive or is not alive. There two Modes where you can give a PID as parameter and test if this process is alive.

You can read more details on how IEP works in the chapter 1.7. Application and PID 137.

Syntax

IEP.P1[|P2][|P3] ... ELS. ... EIF.

Parameter Explanation

There are two Usage Mods.

"If_exist_Process-Mode":
P1 – Variable/Exe-/DLL-Filename, with or without path. Pattern can be used.

P2 - Variable/Number/Formula - Number of result (take fist, second etc. fitting result)

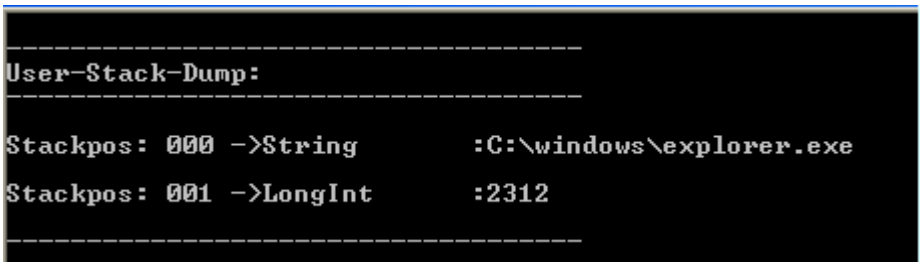
P3 - Variable if given gets the full path of the found exe/dll

"Is_Process_Alive-Mode":
P1 – Starting with a ***">"***-Sign this indicates that we are in this Mode. Now follow one or more PID's separated by ",". The rule is: if ONE of these PID's is alive, the data of this process is been place on stack, and the condition evaluates to true. Only if none of the given PID's is alive the condition evaluates to false.

© 2023 Theo Gottwald * https://www.smart-package.com

In this Mode you'll find some information about the process that is alive on the stack. See following picture. On TOS you see the full path of the process. And below you see the PID of the process. Also `#pid#` is set to the alive process.

```
' Example "Is process alive Mode"
IEP.>$$PIA,$$PIB,$$PIC
...
EIF.
```



"Is_Process_Dead-Mode":

P1 – Starting with a ***"<"***-Sign this indicates that we are in this Mode. Now follow one or more PID's separated by ",". The rule is: if ONE of these PID's is NOT alive, then the PID of this process is been place on stack, and the condition evaluates to true. Only if all of the given PID's are alive then the condition evaluates to false. Also #pid# is set to the last process that was found alive. If multiple PID's given and the first one is dead, `#pid#` is not changed.

```
' Example "Is process dead Mode"
IEP.<$$PIA,$$PIB,$$PIC
...
EIF.
```

The negative Form of `IEP.` is `NEP.` **`NEP.` is available for all Modes above.**

Example

```
'**********************************
' IEP.-Demo (IsProcess Alive)
'**********************************
' This first IEP. Shows the normal Mode, we ask for
' a specified process to exist
' we will get the PID of that process into #pid#
IEP.Explorer.exe
  ' Now we have a valid PID
  : $$PID=#pid#
  ' Now we can test if the PID is "alive"
  IEP.>$$PID
    DBP. The Process $$000 is alive
  EIF.
EIF.
ENR.

'**********************************
' IEP.-Demo (IsProcess dead)
'**********************************
' We use a Search-Pattern this time
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IEP.{{Ex&THEN:rer.exe}&OR:Explo}
  : $$PID=#pid#
  ' Now we have a valid PID
  ' Now we can test if the PID is not alive
  IEP.<$$PID,99,45,34,222
    DBP. At least one of the Processes 99,45,34,222 is not
alive
    DBP.Its PID:$$000
  EIF.
EIF.
ENR.
```

Remarks

The command works transparently with both x32 and x64 applications.

`IEP.` and other `IXX.` can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX.` (Enumerations) to unlimited Depth.

Limitations:

no Limitations.

See also:

- 1.6.1. Program Flow Control 121
- 1.7. Application and PID 137
- GPI. - Get Process Information 2893
- FEP. - For-Each-Process 932
- GDF. - Get-Data-From 2889
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.9 KPN. - Kill Process by Name

KPN. - Kill Process by Name Previous 2900 Top 41 Next 2907

MiniRobotLanguage (MRL)

KPN. Command

Kill Process by Name

Intention

This command is used to kill all processes with a specified name. For example, if you want to kill the "Internet Explorer" you could use:

```
' will close one IE
KPN.iexplore.exe

' Will close all IE's
KPN.iexplore.exe|a
```

Here is an complete working example.

```
KPN.iexplore.exe|a
WII.5000
FOR.$$CNT|1|20
  EXS.iexplore|www.fa2.de
NEX.
MBX.Click Me to Close all IE's
' Here we do NOT use background Mode
KPN.iexplore.exe|a
PAU.5
ENR.
```

KPN. has a **background mode**. In this Mode, the script will continue running while KPN. will close these processes in the background, while your script continues.. In short it looks like this.

```
KPN.iexplore.exe|ab
' Give it a bit time to work in the background
PAU.4
```

A complete working example looks like this:

```
KPN.iexplore.exe|a
WII.5000
FOR.$$CNT|1|20
  EXS.iexplore|www.fa2.de
NEX.
MBX.Click Me to Close all IE's
' Here we use background Mode
KPN.iexplore.exe|ab
PAU.5
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

You can also specify PID's that will be ignored (not closed). This would look like this:

```
KPN.iexplore.exe|ax=123,$$PID,2134
KPN.iexplore.exe|abx=123,345,$$PID
```

Syntax

KPN.P1[|P2]

Parameter Explanation

P1 **-** Processname:
  name or part of a name or path of a process.
  You can use the standard search patterns.
  You can give a full path or just the process name or a part of it.

P2 **-** Options:
  Generally the first process fitting that name will be ended.
  If you have more processes with the same name and you want to end
  them all, use the
       a - Flag: End All processes fitting that name

  You can add exceptions which will NOT be killed using the
       x= option.
  For this, just give all the PID's (Process ID's) of the processes
  that should not be killed.

  You can use the GDF. - command to get a PID from a window.

Important Notice: If you use KPN. at the end of your script, you may need to add a PAU. - command because KPN. works in the background!
If the script ends, the KPN.-background process would also end before its finished.

See this example:

```
KPN.iexplore.exe|a
CAW.ieframe
WII.5000
FOR.$$CNT|1|20
  EXS.iexplore|www.fa2.de
  STS.DUMP
  PAU..1
NEX.
STS.DUMP
MBX.Click to CLose all IE's
KPN.iexplore.exe|a
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Example

```
'*********************************************
' KPN. Examples
'*********************************************
' close all Internet explorers
KPN.iexplore.exe|a

STW.c|ieframe
GDF.hp|$$PID

' close all Internet explorers except the one we localized
above
KPN.iexplore.exe|ax=$$PID


'*********************************************
```

Remarks

Prefer CLW. and CAW. before using KPN. While KPN. is faster, its also more violent. CLW. is less violent to the process and makes sure that all process resources are properly freed. KPN. is similar then if you "shoot a process" using the windows task manager. If you can't shoot a process using the windows task manager, KPN. may also fail.

Limitations:

-

See also:

- CLW. - Close-Window 3592
- CAW. - Close-All-Windows 3586
- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904
- IEP. / NEP. - If-Exists-Process 2900
- GDF. - Get-Data-From 2889
- GPI. - Get Process Information 2893
- GPU. - Get Process User 2898
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.10 MRK. - MiniRobotKill

MiniRobotLanguage (MRL)

MRK. Command

Mini Robots Kill

Intention

These command is used to kill all processes with the name "`MR01.exe`" that may be currently running on that system, except the actual running robot.

This command will make sure that no Instances of the Robot-Executable are leftover on the system.

Syntax

MRK.

Parameter Explanation

no parameters

Example

```
'************************************************
' MRK.
'************************************************

' Kill all running robots
MRK.


'************************************************
```

Remarks

-

Limitations:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

See also:

- 1.7. Application and PID 137
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- CLW. - Close-Window 3592
- CAW. - Close-All-Windows 3586
- CTR. - Close-ThRead 2884
- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904
- GDF. - Get-Data-From 2889

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.11 WII. - Wait-for-Input-Idle

MiniRobotLanguage (MRL)

WII. Command

Wait for Input Idle

Intention

Halts script execution until a specified Process is ready to process Input.
This Command can be seen as a dynamic Pause-Statement, unlike a fixed statement which waits always for a fixed time.

If this process is a console application or does not have a message queue, **WaitForInputIdle** returns immediately.

Syntax

WII.[P1][|P2]

Parameter Explanation

P1 **-** time to pause (maximum) in seconds

P2 **-** (optional) Process-ID of the process to wait for.
If omitted, actual window-process is taken.

Please note that some old programs do not support WII. and therefore may hang if WII. is used on them.

Example

```
'************************************
' WII - Demo
'************************************
'DBM.2
: §§LNK=http://www.fa2.de/
EXH.index.html|$$HWA|§§LNK
PRT.Handle No. 1 is $$HWA
STW.h|$$HWA
WII.
:over
END.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

-

Limitations:

-

See also:

- OPT. - Optional Settings 2406
- IEP. / NEP. - If-Exists-Process 2900
- GDF. - Get-Data-From 2889
- CPR. - Close-PRocess 2882
- GPI. - Get Process Information 2893
- KPN. - Kill Process by Name 2904
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.12 WPR. - Wait-Process-Running

MiniRobotLanguage (MRL)

WPR. Command

Wait Process Running

Intention

Halts script execution until a specified process has been started.

See: 1.7. Application and PID 137

Syntax

WPR.[P1][|P2]

Parameter Explanation

P1 - Filename (with or without path) of the EXE which shall be monitored

P2 - Timeout (0 = infinite) if omitted 0 is taken

Example

```
'**********************************
' WPR.-Demo
'**********************************
DBM.2
' First closing all IE's
:retry
IEP.iexplore.exe
 CPR.0|#pid#
 JMP.retry
EIF.
MBX. Please click Ok. then start the Internet Explorer
WPR.iexplore.exe
MBX.IE running!
'**********************************
' WPR.-Demo 2 with Timeout of 3 Seconds
'**********************************
' First closing all IE's
:retrb
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
IEP.iexplore.exe
 CPR.0|#pid#
 JMP.retrb
EIF.
MBX. Please click Ok. then start the Internet Explorer
WPR.iexplore.exe|3
MBX.IE running or Timeout!
:over
MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- 1.7. Application and PID 137
- GPI. - Get Process Information 2893
- GPU. - Get Process User 2898
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- GDF. - Get-Data-From 2889
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.74.13 WPT. - Wait-Process-Termination

MiniRobotLanguage (MRL)

WPT. Command

Wait Process Termination

Intention

Halts script execution until a specified process has been ended.

See: 1.7. Application and PID 137

Syntax

WPT.[P1][|P2]

Parameter Explanation

P1 – PID

P2 – Timeout (0 = infinite) if omitted 0 is taken

Example

```
'************************************
' WPT - Demo
'************************************
'DBM.2
: $$LNK=http://www.fa2.de/
EXH.index.html|$$HWA|$$LNK
PRT.Handle No. 1 is $$HWA
STW.h|$$HWA
MBX. Please click Ok. then close the Internet Explorer
WPT.#pid#
MBX. Next please do it again within 5 Seconds
EXH.index.html|$$HWA|$$LNK
PRT.Handle No. 1 is $$HWA
STW.h|$$HWA
WPT.#pid#|5
PRT.Closed or timeout happened
:over
MBX.!
END.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

-

Limitations:

-

See also:

- 1.7. Application and PID 137
- GPI. - Get Process Information 2893
- GPU. - Get Process User 2898
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- GDF. - Get-Data-From 2889
- WPR. - Wait-Process-Running 2911
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.75 Registry-commands

3.75.1 ! Smarty's Registry Commands

Smarty's Registry Commands

Previous 2913 Top 41 Next 2916

Smarty's Registry Commands

© 2023 Theo Gottwald * https://www.smart-package.com

3.75.2 DID. - Decode (MS-)Product-ID

MiniRobotLanguage (MRL)

DID. Command

Decode (MS-) product-ID

Intention

Find and decode MS-Product-ID from Registry. Decoded Product-ID's follow the Pattern XXXX-XXXX-XXXX-XXXX. In the registry they are stored in an encrypted way. The parameter is a Registry part that is searched for a Key named "DigitalProductId" and "ProductName".
If the given Product and the Product-ID to that Product can not be found in registry, then the command will return the next reachable Pruduct-ID.

Syntax

DID.[P1][|P2]

Parameter Explanation

When no parameter is given then the value will be read from the stack and the result will be placed on the stack (TOS)

P1 - ***(in)*** Registry Key to read.
P2 - ***(out)*** Variable that will take the result-key.

Example

```
'*******************************************
' DID. - Sample
'*******************************************
PRT.=========================================================
=====
' Using the full path from RegEdit
: $$KEY=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet
Explorer\Registration
DID.$$KEY
PRT.$$000
PRT.=========================================================
=====
Using a half-Path without HKLM works, too.
: $$KEY=SOFTWARE\Microsoft\Windows NT\CurrentVersion
DID.$$KEY
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
PRT.From Stack: $$000
PRT.============================================================
=====
' in this case we specify only the relevant part.
: §§KEY=Windows NT\CurrentVersion
DID.§§KEY|§§RES
PRT.In Variable: §§RES
PRT.============================================================
=====
' Office-Reg (Example)...
: §§KEY=Office\11.0\Registration\
DID.§§KEY|§§RES
PRT.In Variable: §§RES
PRT.============================================================
=====
DMP.4
END.
```

Remarks

-

Limitations:

`DID.` is currently limited to 32 bit applications.

See also:

- IER. / NER. - If-Exist-Registry 2918
- RER. - Registry Entry Read 2925

© 2023 Theo Gottwald * https://www.smart-package.com

3.75.3 IER. / NER. - If-Exist-Registry

MiniRobotLanguage (MRL)

IER. Conditional Statement

If Exist Registry

Intention

This command can be used to find out, if a specified registry Key, Entry or Value is available. Standard-Search Pattern can not be used.

You can use this statement on x64 systems. In this case, you have the 32- and the 64-bit registry. You can choose which to use with the `OPT.` - command.

```
' Switch access to x64-registry
OPT.reg64|1
```

or

```
' Switch access to x32-registry on x64 systems
OPT.reg32|1
```

Syntax

IER.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - Registry-path as if you copy it from a ".reg"-file.
P2 - Entry-name if given
P3 - Value if given.
P4 - Type of Entry - if given.

```
Type of Entry is a numeric argument as follows:
'  0 - REG_NONE    ' No value type
'  1 - REG_SZ   ' Unicode nul terminated string
'  2 - REG_EXPAND_SZ ' Unicode nul terminated string plus
Environment-Var
'  3 - REG_BINARY    ' Free form binary
'  4 - REG_DWORD   ' 32-bit number
'  4 - REG_DWORD_LITTLE_ENDIAN' 32-bit number (same as %
REG_DWORD)
'  5 - REG_DWORD_BIG_ENDIAN ' 32-bit number
'  6 - REG_LINK              ' Symbolic Link (unicode)
'  7 - REG_MULTI_SZ         ' Multiple Unicode strings
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'  8 - REG_RESOURCE_LIST ' Resource list in the resource map
'  9 - REG_FULL_RESOURCE_DESCRIPTOR
' 10 - REG_RESOURCE_REQUIREMENTS_LIST
' 11 - REG_QWORD      ' 64-bit number
' 11 - REG_QWORD_LITTLE_ENDIAN  ' (same as %REG_QWORD)
```

Example

```
''*****************************************
' IER. - Sample
'*****************************************
'DBM.2
PRT.=====================================================
:§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVe
rsion\Internet Settings\SO\MISC
IER.§§REG
 PRT. Result: Exist
ELS.
 PRT. Result: Doesn't Exist or no Access-Rights.
EIF.
DMP.4
PRT.=====================================================
:
§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
sion\Internet Settings\SO\MISC
: §§KEY=Type
IER.§§REG|§§KEY
 PRT. Result: Exist
ELS.
 PRT. Result: Doesn't Exist or no Access-Rights.
EIF.
DMP.4
PRT.=====================================================
:
§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
sion\Internet Settings\SO\MISC
: §§KEY=Type
: §§VAL=group
IER.§§REG|§§KEY|§§VAL
 PRT. Result: Exists and is a match
ELS.
 PRT. Result: Doesn't Exist or no Access-Rights or Value does
not match.
EIF.
DMP.4
PRT.=====================================================
' This is a special case because we give 4 parameter and value
is "-"
' in this case value is not evaluated, just the type is
compared
' if value is diffrent from "-" it will also be compared.
:
§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
sion\Internet Settings\SO\MISC
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: §§KEY=Type
: §§VAL=-
: §§TYP=2
IER.§§REG|§§KEY|§§VAL|§§TYP
 PRT. Result: Exists and is a match
ELS.
 PRT. Result: Doesn't Exist or no Access-Rights or Type does
not match.
EIF.
DMP.4
PRT.=================================================
END.
```

Remarks

`IEP`. and other IXX. can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX`. (Enumerations) to unlimited Depth.

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- OPT. - Optional Settings 2406
- IER. / NER. - If-Exist-Registry 2918
- FER. - For-Each-Registry 935
- RER. - Registry Entry Read 2925
- REW. - Registry-Entry-Write 2928
- RDI. - Registry-Delete-Item 2921
- RDT. - Registry-DelTree 2923

© 2023 Theo Gottwald * https://www.smart-package.com

3.75.4 RDI. - Registry-Delete-Item

MiniRobotLanguage (MRL)

RDI. Command

Registry Delete Item

Intention

Delete Registry Items. Depending on parameters this can be a Registry Key or Value. Registry Keys with Subkeys return an API-Error number 372 on Stack.

Therefore this command is less sensitive to script errors compared to `RDT.`, which would delete whatever it can get, in case the parameters are like that.

In case of success, the number 0 is been placed on stack, otherwise an API-Error is placed on stack.

For a list of API-Errors see API-Errors 372.

Syntax

RDI.P1[|P2]

Parameter Explanation

P1 - (in) Registry Key to delete

P2 - (in) Registry Entry-Name to remove

```
When P2 is given the P1 Parameter changes his function to:
     (in) Registry Key from which the registry entry-name to
remove
```

Example

```
'*********************************************
' REW./RDI. - Sample
'*********************************************
'DBM.2
PRT.=========================================
' First we write some Keys ...
:
$$REG=HKEY_LOCAL_MACHINE\SOFTWARE\A_registry_test\Subkey03\A\B
\C
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
REW.§§REG|MYTestItem|Test-Value
MBX.!
' Now we delete the Entry
:
§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\A_registry_test\Subkey03\A\B
\C
RDI.§§REG|MYTestItem
' Now we delete the Key
:
§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\A_registry_test\Subkey03\A\B
\C
RDI.§§REG
DMP.4
END.
```

Remarks

-

Limitations:

This command is currently limited to the x32-Registry.

See also:

- 1.6.1. Program Flow Control 121
- IER. / NER. - If-Exist-Registry 2918
- FER. - For-Each-Registry 935
- RER. - Registry Entry Read 2925
- REW. - Registry-Entry-Write 2928
- RDI. - Registry-Delete-Item 2921
- RDT. - Registry-DelTree 2923

© 2023 Theo Gottwald * https://www.smart-package.com

3.75.5 RDT. - Registry-DelTree

MiniRobotLanguage (MRL)

RDT. Command

Registry Delete Tree

Intention

Delete Registry Items, and any subitems.

Please use RDT. with extreme caution and on your own danger.

In case of success 0 is been placed on stack, otherwise an API-Error 372 is placed on stack.

For safety reasons, prefer `RDI.` where possible to delete single Items from Registry.

Syntax

RDT.P1

Parameter Explanation

P1 **-** `(in) Registry Key to delete`

Example

```
'*********************************************
' REW./RDT. - Sample
'*********************************************
'DBM.2
PRT.=========================================
' First we write some Keys ...
:
$$REG=HKEY_LOCAL_MACHINE\SOFTWARE\A_registry_test\Subkey03\A\B
\C
REW.$$REG
' Now we delete them
: $$REG=HKEY_LOCAL_MACHINE\SOFTWARE\A_registry_test\Subkey03
RDT.$$REG
DMP.4
END.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

-

Limitations:

This command is currently limited to the x32-Registry.

See also:

- IER. / NER. - If-Exist-Registry 2918
- FER. - For-Each-Registry 935
- RER. - Registry Entry Read 2925
- REW. - Registry-Entry-Write 2928
- RDI. - Registry-Delete-Item 2921
- RDT. - Registry-DelTree 2923

© 2023 Theo Gottwald * https://www.smart-package.com

3.75.6 RER. - Registry Entry Read

MiniRobotLanguage (MRL)

RER. Command

REgistry entry Read

Intention

Read Registry Entries and values of any datatype..

If this command fails, it will return a API-Error number on the stack.
If the command succeeds, the value on the stack is zero.

```
VAR.$$REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Curren
tVersion\Uninstall\Adobe Acrobat 5.0\
RER.$$REG|DisplayName|$$RES
DBP. Result: >$$RES<
' Now we read a value and get its Type...
RER.$$REG|DisplayName|$$RES|$$TYP
DBP. Result: >$$RES<$$TYP>
END.
```

You can use this command on x64 systems. In this case, you have the 32- and the 64-bit registry. You can choose which to use with the `OPT.` - command.

```
' Switch access to x64-registry
OPT.reg64|1
```

or

```
' Switch access to x32-registry on x64 systems
OPT.reg64|1
```

Syntax

RER.P1|P2|P3[|P4]

Parameter Explanation

P1 **-** (in) Registry Key to read

P2 **-** (in) Name of Reg - Entry which is been read.

P3 **-** (out) Result: Content Value of Reg-Entry which is been read.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
P4 -  (out) numeric value which shows the Datatype of value
which
      is been read. Details  see below.

Here is the numeric code for the datatype-value (P4):
 0 - 'REG_NONE No defined value type.
 1 - 'REG_SZ A null-terminated string.
      It will be a Unicode or ANSI string, depending on
whether
      you use the Unicode or ANSI functions.
 2 - 'REG_EXPAND_SZ A null-terminated string that contains
      unexpanded references to ' environment variables (for
example, "%PATH%").
      It will be a Unicode or ANSI string depending on
      whether you use the Unicode or ANSI functions.
 3 - 'REG_BINARY Binary data in any form.
 4 - 'REG_DWORD A 32-bit number.
 5 - 'REG_DWORD_BIG_ENDIAN A 32-bit number in big-endian
format.
      In big-endian format,
      ' the most significant byte of a word is the low-order
byte.
 6 - 'REG_LINK A Unicode symbolic link.
 7 - 'REG_MULTI_SZ An array of null-terminated strings,
      terminated by two null characters.
 8 - 'REG_RESOURCE_LIST A device-driver resource list.
```

Example

```
DBP.=========================================================
=====
' First we read a value ...
:
$$REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
sion\Uninstall\Adobe Acrobat 5.0\
RER.$$REG|DisplayName|$$RES
DBP. Result: >$$RES<
DBP.=========================================================
=====
' Now we read a value and get its Type...
RER.$$REG|DisplayName|$$RES|$$TYP
DBP. Result: >$$RES<$$TYP>
DBP.=========================================================
=====
END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- IER. / NER. - If-Exist-Registry 2918
- OPT. - Optional Settings 2406
- FER. - For-Each-Registry 935
- RER. - Registry Entry Read 2925
- REW. - Registry-Entry-Write 2928
- RDI. - Registry-Delete-Item 2921
- RDT. - Registry-DelTree 2923

© 2023 Theo Gottwald * https://www.smart-package.com

3.75.7 REW. - Registry-Entry-Write

MiniRobotLanguage (MRL)

REW. Command

REgistry entry Write

Intention

Creates Registry Keys to unlimited depth, creates Entries of any Datatype and value.

If this command fails, it will return a API-Error number on the stack.
If the command succeeds, the value on the stack is zero.

You can use this command on x64 systems. In this case, you have the 32- and the 64-bit registry. You can choose which to use with the `OPT.` - command.

```
' Switch access to x64-registry
OPT.reg64|1
```

or

```
' Switch access to x32-registry on x64 systems
OPT.reg64|1
```

Syntax

REW.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - Registry Key to create

P2 - Name of empty Reg - Entry which is been created.

P3 - Content Value of Reg-Entry which is been created.

P4 - Datatype of value which is been created. This is numeric see below.

```
here is the numeric code for the datatype-value:
 0 - 'REG_NONE No defined value type.
 1 - 'REG_SZ A null-terminated string.
      It will be a Unicode or ANSI string, depending on
      whether you use the Unicode or ANSI functions.
 2 - 'REG_EXPAND_SZ A null-terminated string that
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
       contains unexpanded references to
       ' environment variables (for example, "%PATH%").
       It will be a Unicode or ANSI string depending on
       whether you use the Unicode or ANSI functions.
  3 - 'REG_BINARY Binary data in any form.
  4 - 'REG_DWORD A 32-bit number.
  5 - 'REG_DWORD_BIG_ENDIAN A 32-bit number in big-endian
 format.
       In big-endian format,
       ' the most significant byte of a word is the low-order
 byte.
  6 - 'REG_LINK A Unicode symbolic link.
  7 - 'REG_MULTI_SZ An array of null-terminated strings,
       terminated by two null characters.
  8 - 'REG_RESOURCE_LIST A device-driver resource list.
```

Example

```
'*******************************************
' REW. - Sample
'*******************************************
'DBM.2
  $$REG=HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Serv
ices
  $$REG=\$$REGlanmanserver\parameters\
  DBP. Result: Exist

  $$MYN=AutoShareWks
  $$VAL=0
 REW.$$REG|$$MYN|$$VAL|4


 PRT.=========================================================
 =====
 ' First we write some Keys ...
 :
 §§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
 sion\Uninstall\AddressBook\NewA\NEWB\
 REW.§§REG
 DMP.4
 PRT.=========================================================
 =====
 ' Now we write some Keys AND a empty value ...
 :
 §§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
 sion\Uninstall\AddressBook\NewA\NEWB\
 REW.§§REG|MYVAL
 DMP.4
 PRT.=========================================================
 =====
 ' Now we write some Keys AND a REG_SZ value ...
 :
 §§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
 sion\Uninstall\AddressBook\NewA\NEWB\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: §§MYN=Val_Name
: §§VAL=Content
REW.§§REG|§§MYN|§§VAL
DMP.4
PRT.=============================================================
=====
' Now we write some Keys AND a DWORD value ...
:
§§REG=HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVer
sion\Uninstall\AddressBook\NewA\NEWB\
: §§MYN=Val_Number
: §§VAL=123456
REW.§§REG|§§MYN|§§VAL|4
DMP.4

END.
```

Remarks

-

Limitations:

-

See also:

- IER. / NER. - If-Exist-Registry 2918
- OPT. - Optional Settings 2406
- FER. - For-Each-Registry 935
- RER. - Registry Entry Read 2925
- REW. - Registry-Entry-Write 2928
- RDI. - Registry-Delete-Item 2921
- RDT. - Registry-DelTree 2923

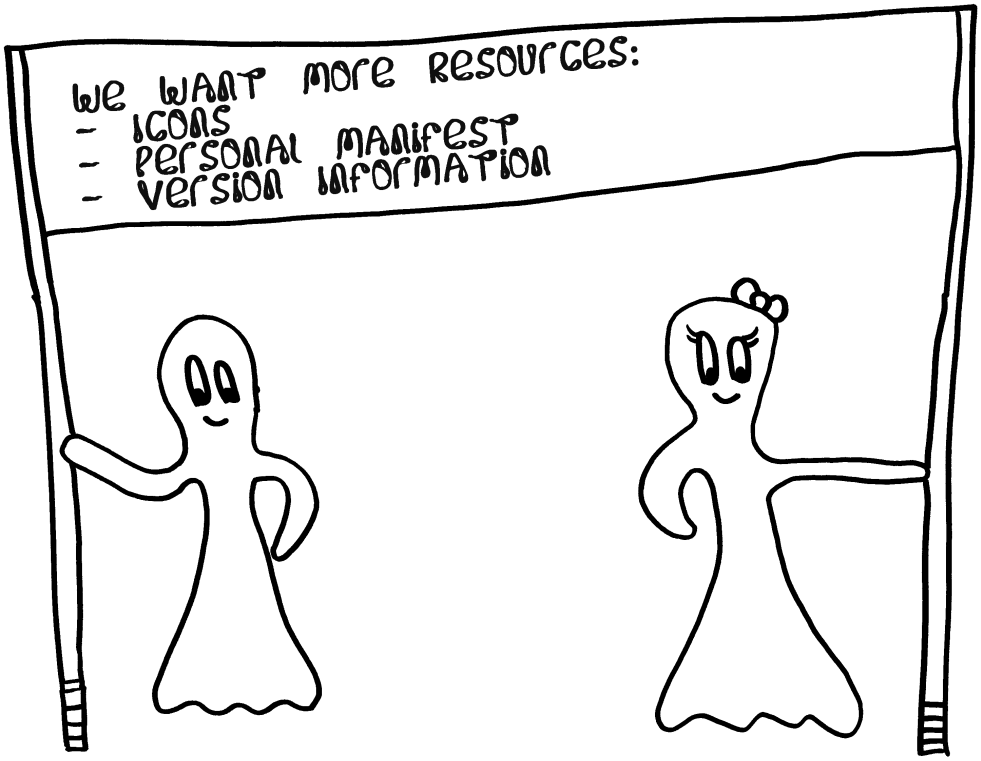
© 2023 Theo Gottwald * https://www.smart-package.com

3.76 RES. - EXE-Resourcen managen

3.76.1 ! Resourcen Commands

Smarty's Ressourcen-Commands Previous 937 Top 41 Next 946

Smarty's Ressourcen Commands



Ressources in our sense are:

- Custom Icons
- personal Manifest
- Changing or adding Version-Info

While developing this task, we found that even some other commercial tools produce incorrect results.
We have therefore added the worldwide best we could find::

*© 2023 Theo Gottwald * https://www.smart-package.com*

This engine is doing all the complicated tasks internally.
You don't need to care for the details.

Anyway, handling Resources is a direct change in an executable file.
You will need to know what you are doing.

IMPORTANT TIPPS:
Please take care of these points.

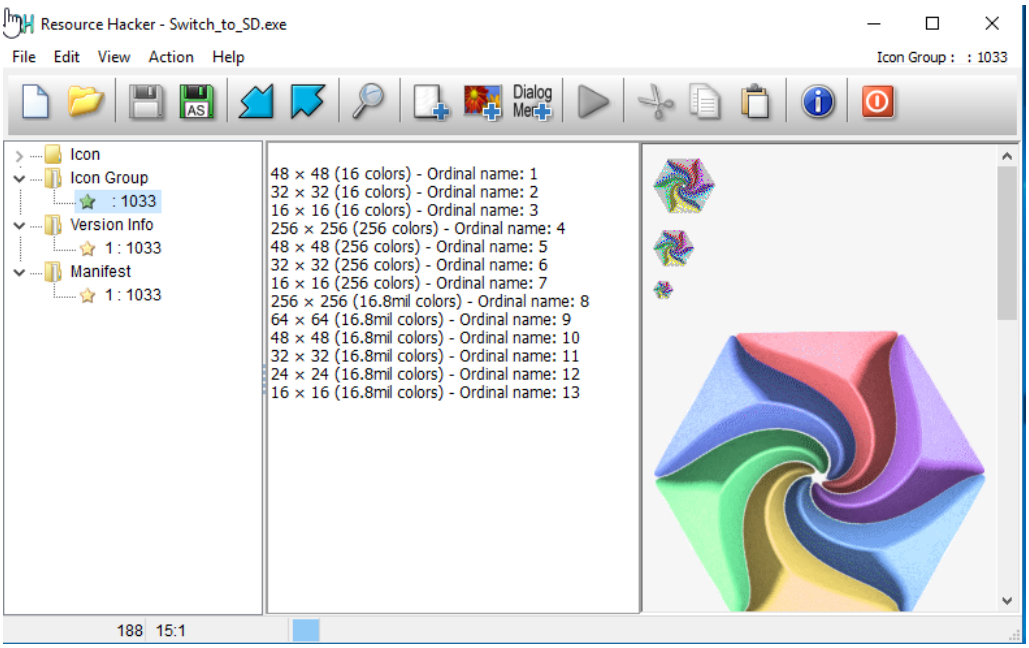
1. Files may be protected or copyrighted
2. Signed files may loose their signature (but you may sign them yourself after the process)
3. If the process fails for whatever reason (Antivirus .. etc.) the result may be an unusable file!
Especially for this reason, always work on a copy of the original file!
4. Using the File-Version changing Commands will remove all file-version information that have been there before,
and replace them with your given data.

To make manual changes on your Resources without using an SPR-Script, you can take a look on this Freeware-Tool:
Resourcehacker(TM) by Angus Johnsen

We may use some Screenshots from this Tool, in this help-file. Here is one:

On the left side you can see that we have three types of Resources:
- The Icons
- The Manifest
- The Version-Information

© 2023 Theo Gottwald * https://www.smart-package.com



Using the `RES.`-Command, you can:

- extract Resources from EXE and DLL-Files:
- extract Icons
- extract Version-Information in all languages
- extract the used Manifest

- change Resources in an EXE or DLL-File:
- change the used Icon
- replace the used Manifest
- replace the Version-Information

- And finally, if there is no resources attached to an executable, you can attach your own Ressources
- attach an Icon
- attach a Manifest
- add Version Information

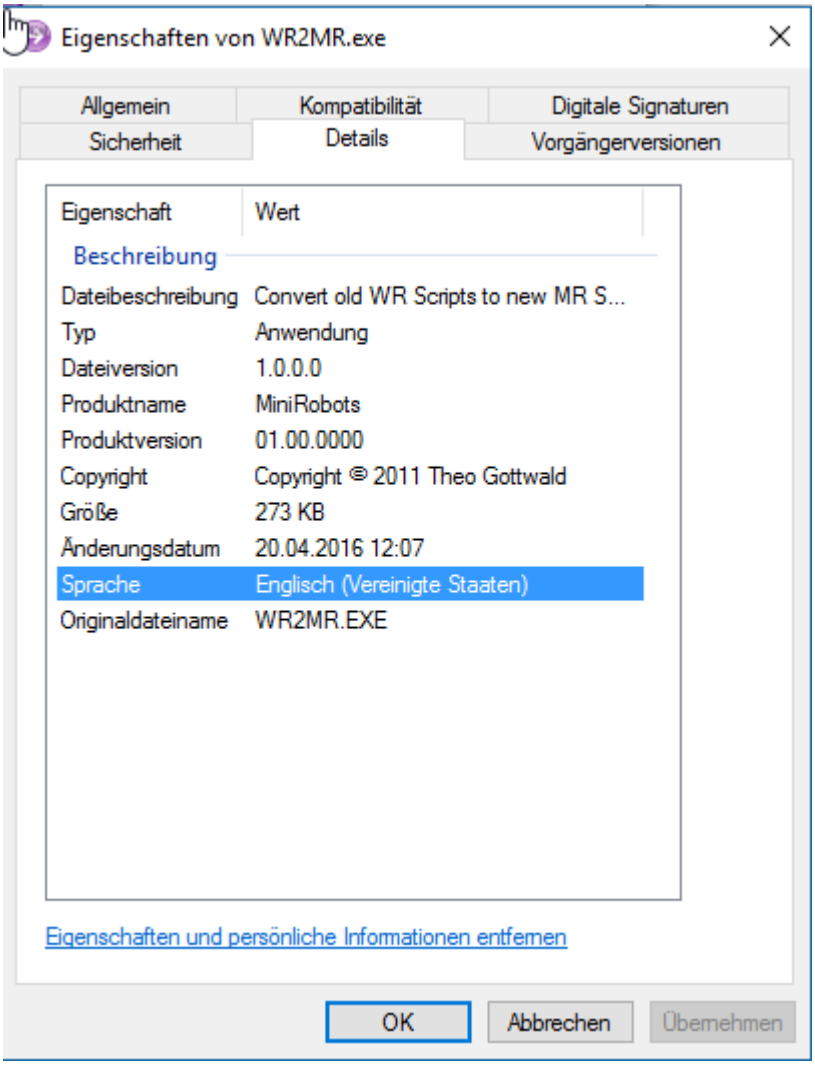
- additionally you can "Refresh" the explorer view. Because there is one more thing to know.

CHANGING the Icon may not show up immediately in the Windows Explorer, because of the "old Icon" still being stored in the windows "Icon Cache".

You can use several ways to check which Icon is really attached to a file.
- Use the available RES.-Command
- Copy the file to another place with a new Name
- Open the "File-Properties", they will show the most actual Icon
- press and hold down the left strg/ctrl-Key and then turn the Mouse-Wheel. Changing the Icon-Size may also reveal the "real Icon".

CHANGING the Version Information, make sure to use the Language that is displayed.
Otherwise you may end up with empty version information.

*© 2023 Theo Gottwald * https://www.smart-package.com*



Especially when changing "Version Information" we strongly recommend to work with a Backup" and not with the original File.
For this several Commands have an "Backup-Option", already built in".

You can also apply any of these methods to groups of files, using `FEF`.

Here is a code Sample that will change the Icon of a EXE-File.

```
'=============================================================
   ' Change the Icon of an EXE-File
   $$PAB=?path\WR2MR.exe
   $$ICO=?path\My.ico
   RES.Change Icon|$$PAB|$$ICO
ENR.
'=============================================================
   ' Change the Manifest of an EXE-File
   $$PAB=?path\WR2MR.exe
   $$MAN=?path\My.xml
   RES.Change Manifest|$$PAB|$$MAN
ENR.
'=============================================================
```


© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.2 RES. - Manage Resources

MiniRobotLanguage (MRL)

RES. Command

Manage Ressources like Icons, Manifest and Version Information

Intention

This command is used to change or add the Icon, Manifest or Version Information of Executable or DLL-Files.

Syntax

RES.P1|P2[|P3...|Px]

Parameter Explanation

P1 - Command can be any of these

```
Refresh Explorer
Change Icon
Change Manifest
Change Version
Change Version EN
Change Version DE
Extract All
Extract Icon
Icons Path
Extract Manifest
Extract Version
Last Version
Last Manifest
```

P2-Px - Parameters depending on the given Command.

Example

```
'*******************************************
RES.Refresh Explorer
END.

'*******************************************
VAR.$$EXE=Hello.exe
VAR.$$FIA=?path\EXEs\$$EXE
RES.Extract Version|$$FIA|?path\
ENR.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

'**

Remarks

Please take care of these points.

1. Icon-Files may be copyrighted.

2. Signed files may loose their signature (but you may sign them yourself after the process).

3. If the process fails for whatever reason (Antivirus .. etc.) the result may be an unusable file!
   Especially for this reason, always work on a copy of the original file!

4. Using the File-Version changing Commands will in some cases (for example, if there is wrong aligned version data in the target file)
   remove all file-version information that have been there before, and replace them with your given data. Therefore always work on a copy of
   the file to test what you are going to make changes on.

Limitations:

Note that the Explorer has its own logic in displaying version Information, dependent on the windows version.
This may get into your way when you have multi-language resources.

See also:

- ! Ressourcen Commands 2931
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.3 RES.Change Icon

MiniRobotLanguage (MRL)

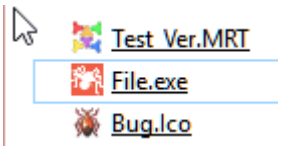
RES.Change Icon

Change the Icons of an executable or DLL-File

Intention

This command is used to change the Icon of an Executable or DLL-File.
As Source Icon, you can specify whether an ".ico"-File, or you can specify another Executable or DLL.
In this case the program will first extract the icon and then use it for the Executable.

Lets make an Example. We have an Folder like this:

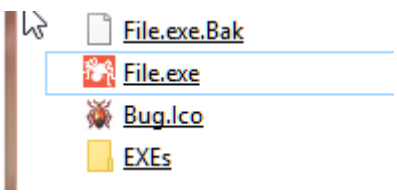


Here is our Script.

```
17  VAR.$$SRC=?path\Bug.ico
18  VAR.$$TAR=?path\File.exe
19
20  RES.Change Icon|$$TAR|$$SRC|1
21  ENR.
22
```

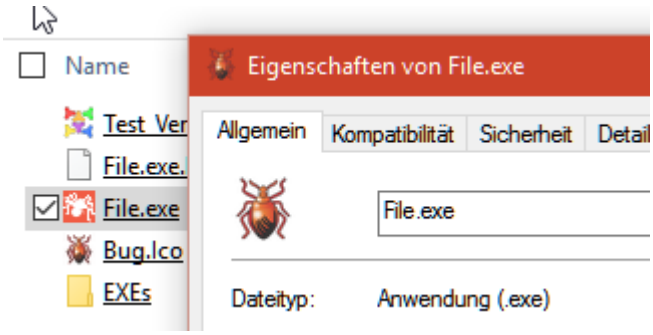
Now we run the Script. The folder will just change a bit and look like this.
For technical reasons the Icon seems unchanged.
Don't worry the Icon was changed, the Explorer just did not realize it yet, due to cache-effects.

You may also notice that the was made an Backup of the Target-File named "`File.exe.Bak`".
This is because we specified the "1" as third Parameter.
It is recommended to always work on a copy of the file.
Doing it this way, you always have a working copy of the Target-File as Backup.



© 2023 Theo Gottwald * https://www.smart-package.com

Now we use the "Properties" to see which Icon is really in there.



And you see, that the Icon has been "implanted".

Important Note: You must specify an valid "`.ico`"-File as Source-Parameter.
You can **not** specify an EXE-File or an DLL-File, even if this one may contain an Icon.
If you specify any other filetype the result is unpredictable.
In our tests, most often the target file will just not have any Icon after the operation.

Syntax

```
RES.Change Icon|P1|P2[|P3]
RES.CI|P1|P2[|P3]
```

Parameter Explanation

P1 - Path of `EXE-File or DLL-File`
P2 - Path of `Icon-File`
P3 - (num) **0/1/2/3 - Flag (default is 0)**
0 - no Backup, Refresh Explorer
1 - make Backup before changing file, Refresh Explorer
2 - do not "Refresh Explorer" do not make a Backup
3 - do not "Refresh Explorer", but make a Backup

Example

```
'***********************************************
' Path to the Icon File
VAR.$$SRC=?path\Bug.ico

' Path to the File that will get this Icon.
VAR.$$TAR=?path\File.exe

RES.Change Icon|$$TAR|$$SRC|1
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Please take care of these points.

1. Files may be protected or copyrighted
2. Signed files may loose their signature (but you may sign them yourself after the process)
3. If the process fails for whatever reason (Antivirus .. etc.) the result may be an unusable file!

Especially for this reason, always work on a copy of the original file!

Limitations:

-

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.4 RES.Change Manifest

MiniRobotLanguage (MRL)

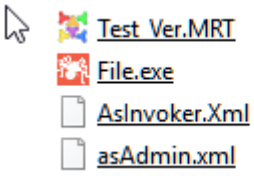
RES.Change Manifest

Change the Manifest of any executable

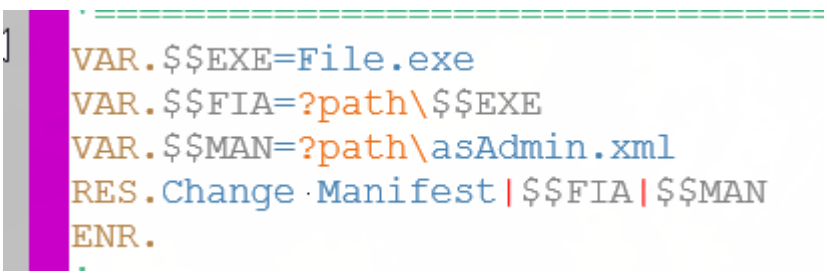
Intention

This command is used to attach an Manifest(.xml) File of your choice to an executable.
The original Manifest - if there was one before - will be removed during the process.

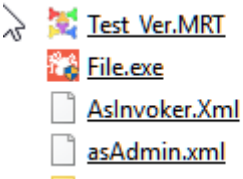
Lets make a Sample. Here we have an executable and an manifest file.



Here is our Script. We run it.



Now the folder looks like this:

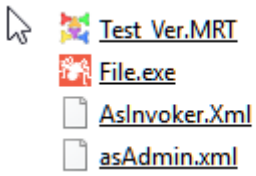


As you can see, the Manifest was implanted into the Executable. Now we remove it and make it to "asInvoker".
Here is our Script:

```
VAR.$$EXE=File.exe
VAR.$$FIA=?path\$$EXE
VAR.$$MAN=?path\asInvoker.xml
' You can use the short Form "cm" for Change Manifest, also.
RES.CM|$$FIA|$$MAN
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

We run it and ... we are back!



Syntax

RES.Change Manifest|P1|P2[|P3]
RES.CM|P1|P2[|P3]

Parameter Explanation

P1 - Path of `EXE-File or DLL-File`
P2 - Path of Manifest-File (must be an .xml-File)
P3 - (num) **0/1/2/3 - Flag (default is 0)**
0 - no Backup, Refresh Explorer
1 - make Backup before changing file, Refresh Explorer
2 - do not "Refresh Explorer" do not make a Backup
3 - do not "Refresh Explorer", but make a Backup

Example

```
'*********************************************
VAR.$$EXE=File.exe
VAR.$$FIA=?path\$$EXE
VAR.$$MAN=?path\asInvoker.xml
RES.Change Manifest|$$FIA|$$MAN
ENR.
```

Remarks

Please take care of these points.

1. Files may be protected or copyrighted
2. Signed files may loose their signature (but you may sign them yourself after the process)
3. If the process fails for whatever reason (Antivirus .. etc.) the result may be an unusable file!
Especially for this reason, always work on a copy of the original file!

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.3.0. '#MAN: - Manifest for Executable 321

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.5 RES.Change Version

MiniRobotLanguage (MRL)

RES.Change Version

Add or change Version Information in an Executable File or DLL

Intention

This command is used to Add or Change Version Information in an Executable File or DLL.
Adding or changing is supported, even Multi-Language Resources can be added or changed.

Here is an Example:

```
   ' Sample changing Version Information
' Name and Path of the File into variable
VAR.$$FIA=?path\File.exe
' Tell command what do change
VAR.$$VER=LangId = 040704B0|ProductName=Package Robot|
FileVersion=1.2.3.4.56|MainLangId = 040704B0
' Do changes
RES.Change Version|$$FIA|$$VER|1
ENR.
```

The **special Version for German**:
`RES.Change Version` **`DE`**`|$$FIA|$$VER`
will internally prefix the String with a "`LangId = 040704B0|`" and add a "`|MainLangId=040704B0`" at the end.

The **special Version for English:**
`RES.Change Version` **`EN`**`|$$FIA|$$VER`
will internally prefix the String with a "`LangId = 040904B0|`" and add a "`|MainLangId=040904B0`" at the end.

Note that the proper use of "MainLangID = xxxxxx" and "LangId = xxxxxx" is important.

Technically:
"**`MainLangId`**" defines the "`VarFileInfo`" Block of the original Resources-File, that was used when the Executable was created..

```
BLOCK "VarFileInfo"
{
VALUE "Translation", 0x0407 0x04B0
VALUE "Translation", 0x0409 0x04B0
}
```

"**`LangId`**" defines a block inside the **`"StringFileInfo"`** Block of the original Resources-File, that was use

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
BLOCK "StringFileInfo"
{
BLOCK "040704B0"
{
VALUE "Comments", "[-De-] Comments"
VALUE "CompanyName", "[-De-] CompanyName"
…
}
BLOCK "040904B0"
{
VALUE "Comments", "[-En-] Comments"
VALUE "CompanyName", "[-En-] CompanyName"
…
}
}
```

About Lang-charset and string-name

Its important to use the proper `MainLangId=xxxxxxxx`. The values for all countries can be found in MSDN here:

MSDN StringFileInfo BLOCK

The values for the charactersets are in Decimal on the MSDN Page. These need to be converted to Hexadezimal.

Take something like <`BLOCK "040904E4"`>

note that the first half "`0490`" correspond to *`U.S. English`*, The second half "`04E4`" correspond to Multilingual (Decimal `1252`).

Here you would write: `MainLangId=049004E4`

Syntax

```
RES.Change Version|P1|P2[|
P3...|Px]
RES.Change Version en|P1|P2[|
P3...|Px]
RES.Change Version de|P1|P2[|
P3...|Px]
```

There are the following, functional equal short forms available:

```
RES.CV|P1|P2[|P3...|Px]
RES.CE|P1|P2[|P3...|Px]
RES.CD|P1|P2[|P3...|Px]
```

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - Path of `EXE-File or DLL-File`
P2 - See Description below.
P3 - (num) **0/1/2/3 - Flag (default is 0)**
0 - no Backup, Refresh Explorer
1 - make Backup before changing file, Refresh Explorer
2 - do not "Refresh Explorer" do not make a Backup
3 - do not "Refresh Explorer", but make a Backup

Description for P2:

P2 - Contains one or more "Subcommands", all in one String. All spearated by the Pipe-Symbol "|".
You may start with:
`LangId = 040704B0|`

then follow with one or more of the Parmeters below.

finally write:
`|MainLangId = 040704B0`
the given number is in Detail explained here: MSDN StringFileInfo BLOCK .
It starts with a 4 digit Country-code (in Hexadezimal) then 4 digits characterset (Hexadezimal also!).

Inside follow one or more of these Version-Strings followed with an "=" and a value:
- `FileDescription`
- `FileVersion`
- `ProductName`
- `ProductVersion`
- `LegalCopyright`
- `LegalTrademarks`
- `OriginalFilename`
- `CompanyName`
- `Comments`

All of these must be followed by an "=" and then the String that shall be assigned to this Version-Info.
If it is followed by more other "Subcommands" you need to delimit them with the Pipe-Symbol "|".

Example:

```
LangId = 040704B0|ProductName=Package Robot|FileVersion=1.2.3.4.56|MainLangId =
```

Example

```
'*********************************************
   ' Sample changing Version Information
VAR.$$EXE=File.exe
VAR.$$FIA=?path\$$EXE
VAR.$$VER=LangId = 040704B0|ProductName=Package Robot|
FileVersion=1.2.3.4.56|MainLangId = 040704B0
RES.Change Version|$$FIA|$$VER|1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Remarks

Please take care of these points.

1. Files may be protected or copyrighted
2. Signed files may loose their signature (but you may sign them yourself after the process)
3. If the process fails for whatever reason (Antivirus .. etc.) the result may be an unusable file!
Especially for this reason, always work on a copy of the original file!
4. Using the File-Version changing Commands will remove all file-version information that have been there before,
and replace them with your given data.

Limitations:

-

See also:

- MSDN StringFileInfo BLOCK
- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.4.0. '#VER: - Version-Informations 328

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.6 RES.Refresh Explorer

MiniRobotLanguage (MRL)

RES.Refresh Explorer

Refresh the Explorer-View after changing Ressources like Icons, Manifest and Version Information

Intention

This command is used to Refresh the Explorer-View after changing Ressources like Icons, Manifest and Version Information.

Syntax

```
RES.Refresh Explorer
RES.RE
```

Parameter Explanation for VBE

-

Example

```
'*********************************************
' Refresh the Explorer View
RES.Refresh Explorer
END.
```

Remarks

Please take care of these points.

1. Files may be protected or copyrighted
2. Signed files may loose their signature (but you may sign them yourself after the process)
3. If the process fails for whatever reason (Antivirus .. etc.) the result may be an unusable file!
Especially for this reason, always work on a copy of the original file!
4. Using the File-Version changing Commands will remove all file-version information that have been there before,
and replace them with your given data.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Limitations:

Due to Windows and Explorer Properties, this may not always work as expected.

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.7 RES.Extract All

MiniRobotLanguage (MRL)

RES.Extract All

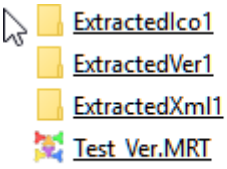
Extract Resources from a given File

Intention

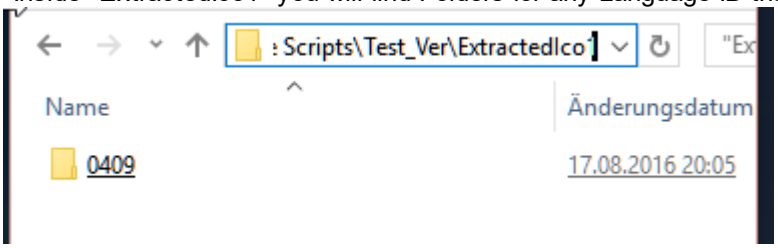
This command is extract Resources from an EXE- or DLL-File.
These Resources will be stored in Folders in a given path.

```
' Example
' This is the path to the Source-File to be extracted
 VAR.$$FIA=?path\EXEs\Hello.exe
' The second Parameter is the Path to the root of all "Target Folders".
 RES.Extract All|$$FIA|?path\
```

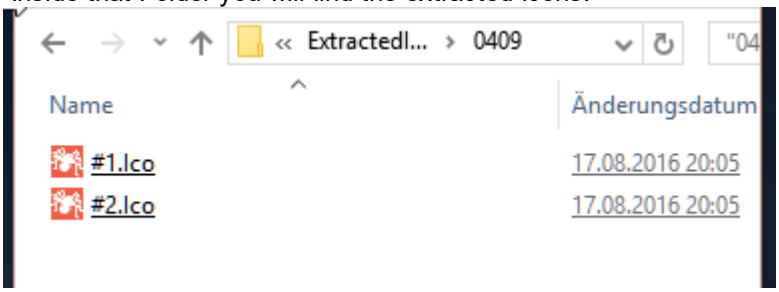
This Script will create these folders in `?path\`.



Inside "ExtractedIco1" you will find Folders for any Language ID that had Icons.

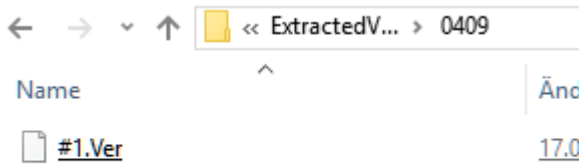


Inside that Folder you will find the extracted Icons.



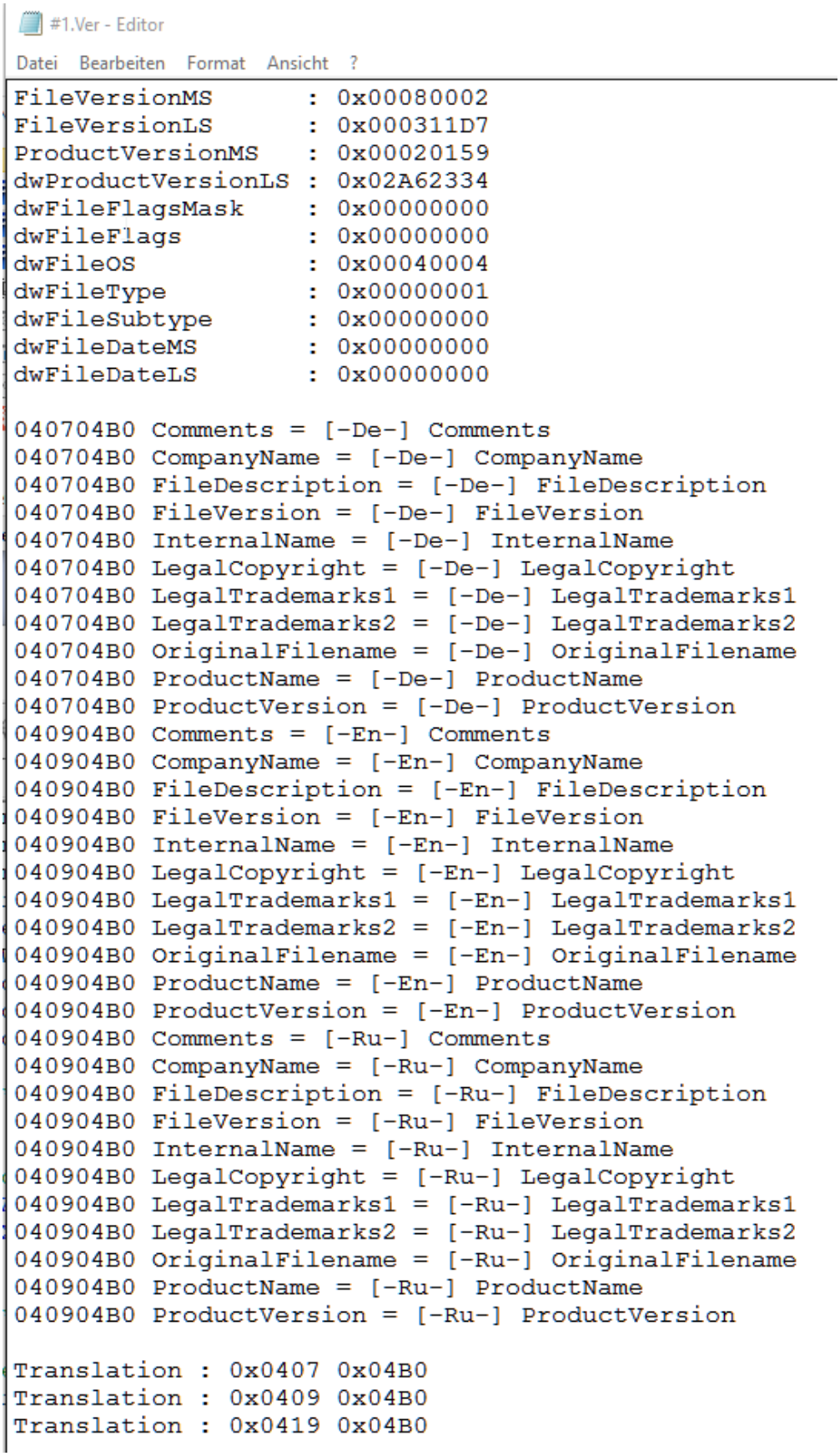
The same applies to the "ExtractedVer1" Folder.
Inside you will first find an folder for each language-ID.
And inside this Folder you find the Version Information for each Language-ID.

© 2023 Theo Gottwald * https://www.smart-package.com



In most cases the generated "`.ver`"-File will look somehow like the file below.
It has an own "Pseudocode"-Format.
It is not possible to import this file anywhere. Its just for reading and analyzing.

*© 2023 Theo Gottwald * https://www.smart-package.com*



© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

RES.Extract All|P1|P2
RES.EA|P1|P2

Parameter Explanation

P1 - Path of `EXE-File or DLL-File`
P2 - Path of Target-Folder

Example

```
'*********************************************
'#EXE:?path\
'#SPI:ForceWrite
' The following files will be inserted into the Executable.
'#INC:?path\EXEs\Hello.exe

VAR.$$FIA=?path\EXEs\Hello.exe
RES.Extract All|$$FIA|?path\
ENR.


'*********************************************
    ' Here is a more complex Sample that uses "RES.LastVersion" to display the las
 VAR.$$EXE=Hello.exe
 VAR.$$FIA=?path\EXEs\$$EXE
 IEF.$$FIA
   RES.Extract Version|$$FIA|?path\
  ' Display Path to Version-File
   MBX.$$TXT

  RES.Last Version|$$TXA
   ' Display Path to Version-File
   MBX.$$TXA
   ' Display Content of Version File
   VAR.$$TXB=$tos$
   MBX.$$TXB
 EIF.
 ENR.
```

Remarks

-

Limitations:

Note that the Explorer has its own logic in displaying version Information, dependent on the windows version.
This may get into your way when you have multi-language resources.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.8 RES.Extract Icon

MiniRobotLanguage (MRL)

RES.Extract Icon

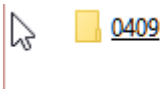
Extract Icons from a specified EXE- or DLL-File

Intention

This command is extract Icon-Resources from an EXE- or DLL-File.
These Resources will be stored in Folders with the name of the country code, in a given path.

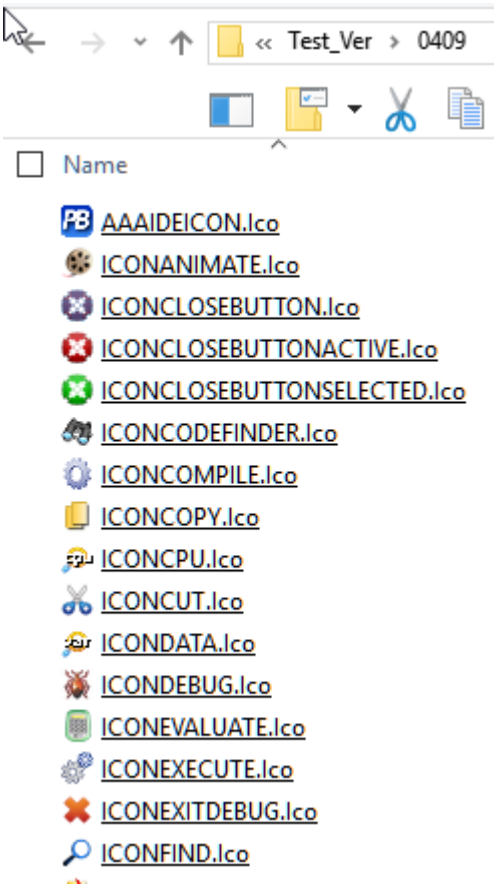
```
' Example
' This is the path to the Source-File to be extracted
 VAR.$$FIA=?path\EXEs\Hello.exe
' The second Parameter is the Path to the root of all "Target Folders".
 RES.Extract Icon|$$FIA|?path\
 MBX.$tos$
 ENR.
```

This Script will create these folder with the name of the LanguageID of the Icons-Resource in `?path\`.



Inside that Folder you will find the extracted Icons.

© 2023 Theo Gottwald * https://www.smart-package.com



The name and path of the last saved Icon will be placed on TOS.

Syntax

RES.Extract Icon|P1|P2
RES.EI|P1|P2

Parameter Explanation

P1 - Path of `EXE-File or DLL-File`
P2 - Path of Target-Folder

Example

```
'*********************************************
'#EXE:?path\
'#SPI:ForceWrite
' This is the path to the Source-File to be extracted
    VAR.$$FIA=?path\EXEs\Hello.exe
   ' The second Parameter is the Path to the root of all "Target Folders".
    RES.Extract Icon|$$FIA|?path\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Last Icon Path is now on the TOS
MBX.$tos$
ENR.
```

Remarks

-

Limitations:

Note that there is currently no way to extract only a specified Icon.

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.9 RES.Extract Manifest

MiniRobotLanguage (MRL)

RES.Extract Manifest

Extract Manifest-Resources from a given EXE- or DLL-File

Intention

This command is extract Manifest-Resources from an EXE- or DLL-File.
These Resources will be stored in Folders with the name of the country code, in a given path.

Here is a Sample-Script.

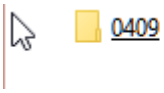
```
  ' This is the path to the Source-File to be extracted
    VAR.$$FIA=?path\EXEs\Hello.exe
   ' The second Parameter is the Path to the root of all "Target Folders".
 RES.Extract Manifest|$$FIA|?path\
MBX.$tos$
  ' To analyze the Manifest in the Script, you can use
RES.Last Manifest|$$TXT
VAR.$$TXA=$tos$
   ' The Manifest is now in $$TXA
MBX.$$TXA
ENR.
```

This Script will display fist this Messagebox with the path to the ".Xml"-File.:

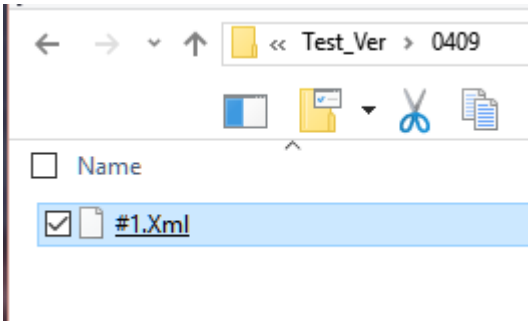
Secondly it will display a Messagebox with the Content of the ".Xml"-file.
The following picture shows only a part.

*© 2023 Theo Gottwald * https://www.smart-package.com*

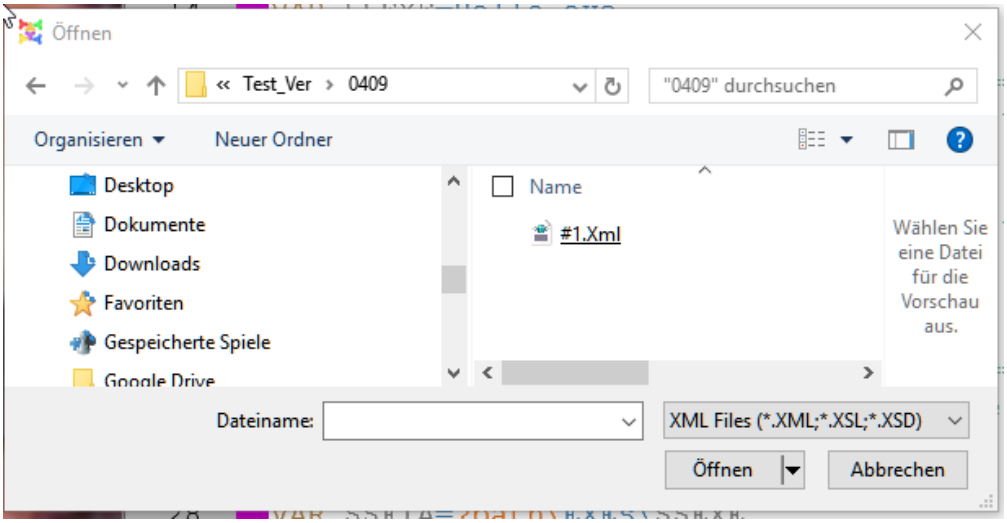
This Script will create these folder with the name of the LanguageID of the Manifest-Resource in `?path\`.



Inside that Folder you will find the extracted Manifest as a "`.XML`"-File.

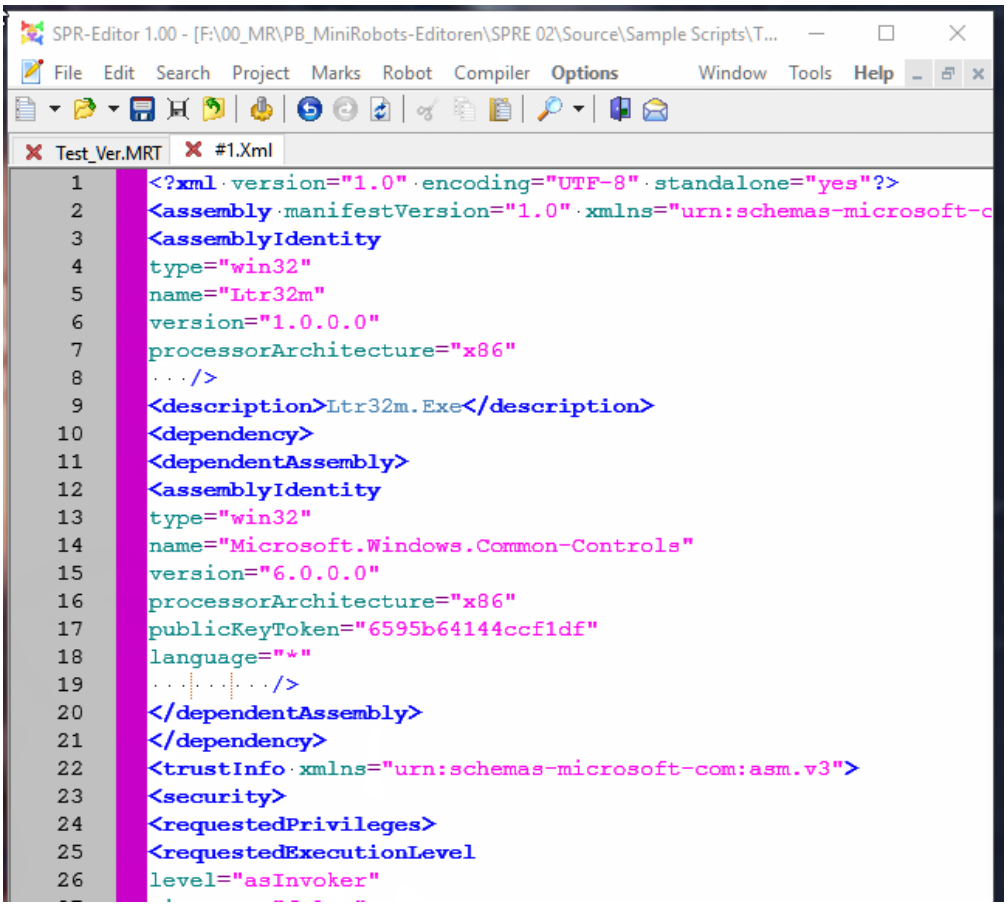


You can Load XML-Files with SPRE, just change the Filter to "XML-Files":



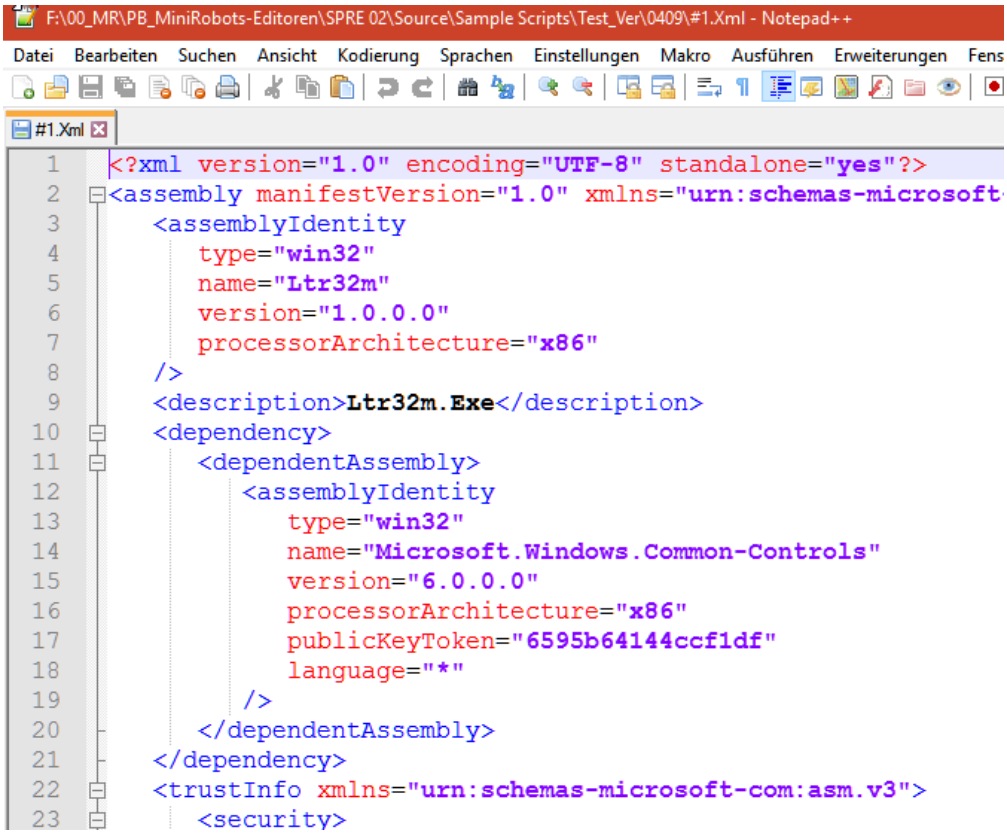
© 2023 Theo Gottwald * https://www.smart-package.com

This may look somehow like here:

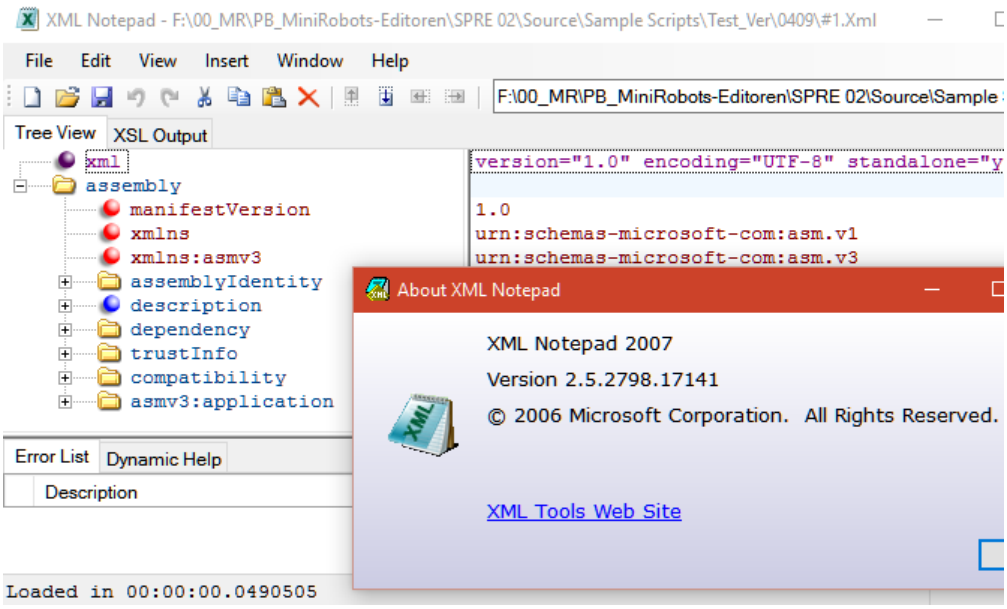


You can also open XML-Files with some advanced Text-Editor Programs. Here "Notepad++".

© 2023 Theo Gottwald * https://www.smart-package.com



Another Option is the "XML-Notepad".



Syntax

RES.Extract Manifest|P1|P2

RES.EM|P1|P2

Parameter Explanation

P1 - Path of `EXE-File or DLL-File`
P2 - Path of Target-Folder

Example

```
'*********************************************
'#EXE:?path\
'#SPI:ForceWrite
' The following files will be inserted into the Executable.
'#INC:?path\EXEs\Hello.exe
    ' This is the path to the Source-File to be extracted
     VAR.$$FIA=?path\EXEs\Hello.exe
    ' The second Parameter is the Path to the root of all "Target Folders".
  RES.Extract Manifest|$$FIA|?path\
 MBX.$tos$
 ' To analyze the Manifest in the Script, you can use
 RES.Last Manifest|$$TXT
 VAR.$$TXA=$tos$
 ' The Manifest is now in $$TXA
 MBX.$$TXA
 ENR
```

Remarks

-

Limitations:

-

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.10 RES.Extract Version

MiniRobotLanguage (MRL)

RES.Extract Version

Extract Resources from a given File

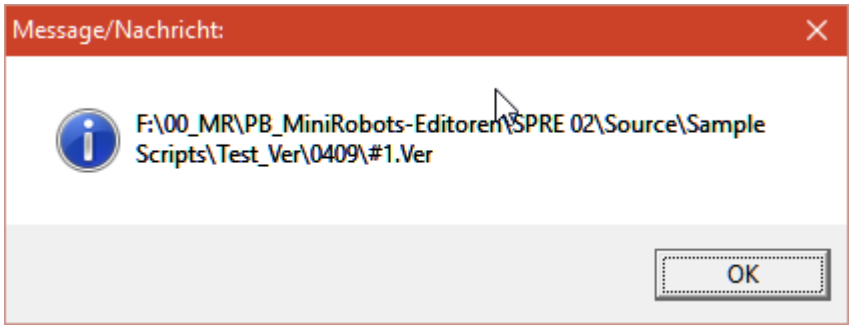
Intention

This command is extract Resources from an EXE- or DLL-File.
These Resources will be stored in Folders with the name of the country code, in a given path.

Here is a Sample-Script.

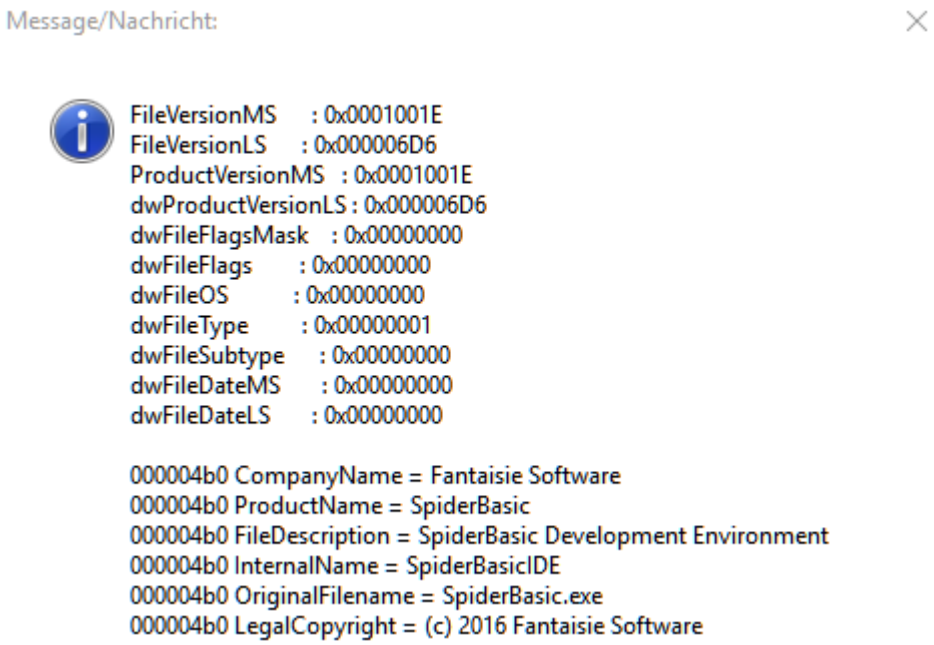
```
' This is the path to the Source-File to be extracted
VAR.$$FIA=?path\File.exe
' The second Parameter is the Path to the root the "Target Folder".
RES.Extract Version|$$FIA|?path\
MBX.$tos$
' To analyze the Version-Information in the Script, you can use
RES.Last Version|$$TXT
VAR.$$TXA=$tos$
' The Version-Info is now in $$TXA
MBX.$$TXA
ENR.
```

This Script will display fist this Messagebox with the path to the "`.Ver`"-File.:

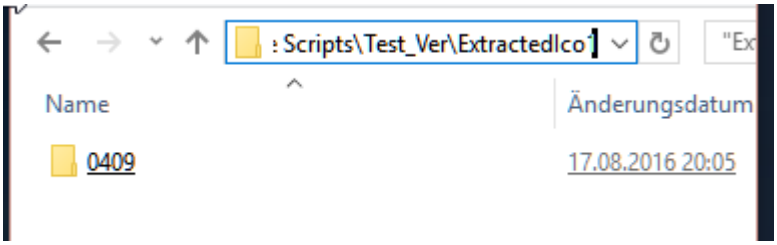


Secondly it will display a Messagebox with the Content of the "`.Ver`"-file.
The following picture shows only a part.

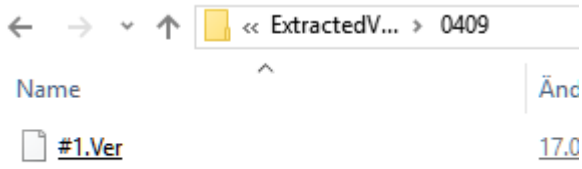
*© 2023 Theo Gottwald * https://www.smart-package.com*



This Script will create these folder with the name of the LanguageID for each available Version-Resource, in `?path\`.

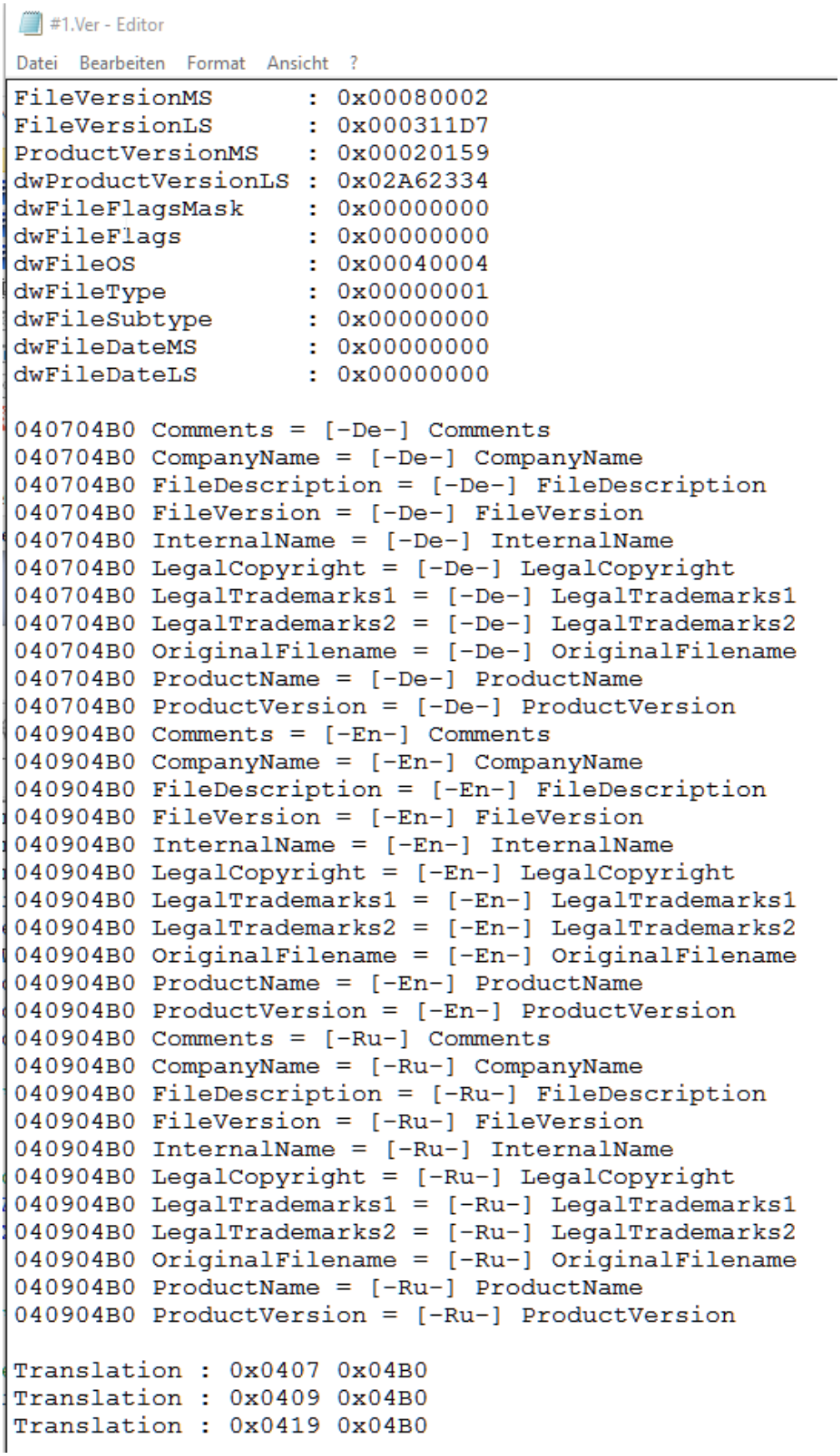


Inside that Folder you will find the extracted Version Information for each Language-ID.



In most cases the generated "`.ver`"-File will look somehow like the file below.
It has an own "Pseudocode"-Format.
It is not possible to import this file anywhere. Its just for reading and analyzing.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

RES.Extract Version|P1|P2
RES.EV|P1|P2

Parameter Explanation

P1 - Path of `EXE-File or DLL-File`
P2 - Path of Target-Folder

Example

```
'*********************************************
'#EXE:?path\
'#SPI:ForceWrite
' The following files will be inserted into the Executable.
'#INC:?path\EXEs\File.exe
 VAR.$$FIA=?path\File.exe
 ' The second Parameter is the Path to the root the "Target Folder".
 RES.Extract Version|$$FIA|?path\
ENR.

'*********************************************
    ' Here is a more complex Sample that uses "RES.LastVersion" to display the las
 ' This is the path to the Source-File to be extracted
 VAR.$$FIA=?path\File.exe
 ' The second Parameter is the Path to the root the "Target Folder".
 RES.Extract Version|$$FIA|?path\
 MBX.$tos$
 ' To analyze the Version-Information in the Script, you can use
 RES.Last Version|$$TXT
 VAR.$$TXA=$tos$
 ' The Version-Info is now in $$TXA
 MBX.$$TXA
 ENR.
```

Remarks

-

Limitations:

Note that the Explorer has its own logic in displaying version Information, dependent on the windows version.
This may get into your way when you have multi-language resources.

See also:

- ! Ressourcen Commands 2931

© 2023 Theo Gottwald * https://www.smart-package.com

- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Last Icon 2978
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.4.0. '#VER: - Version-Informations 328

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.11 RES.Last Manifest

MiniRobotLanguage (MRL)

RES.Last Manifest

Get the Path and Data from the last extracted Manifest

Intention

This command is used to get the Data from the last extracted Manifest.
The command will get the Path to the Manifest-File into a Variable, and the Data of the File on the Top Of Stack `$tos$`.
To see the effect of the Program, its necessary to first "extract" an Manifest.
The command can also be used after an "`RES.Extract All`".

Here is a Sample-Script.

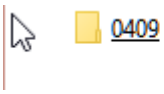
```
' This is the path to the Source-File to be extracted
   VAR.$$FIA=?path\EXEs\Hello.exe
  ' The second Parameter is the Path to the root of all "Target Folders".
 RES.Extract Manifest|$$FIA|?path\
MBX.$tos$
' To analyze the Manifest in the Script, you can use
RES.Last Manifest|$$TXT
VAR.$$TXA=$tos$
' The Manifest is now in $$TXA
MBX.$$TXA
ENR.
```

This Script will display fist this Messagebox with the path to the "`.Xml`"-File.:

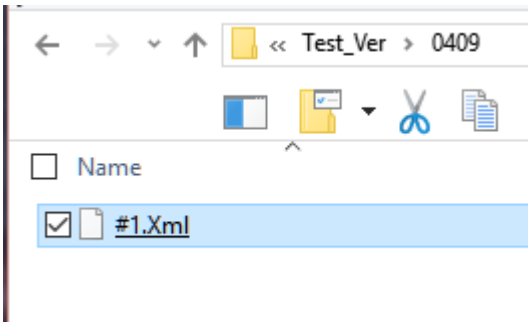
Secondly it will display a Messagebox with the Content of the "`.Xml`"-file.
The following picture shows only a part.

 *© 2023 Theo Gottwald * https://www.smart-package.com*

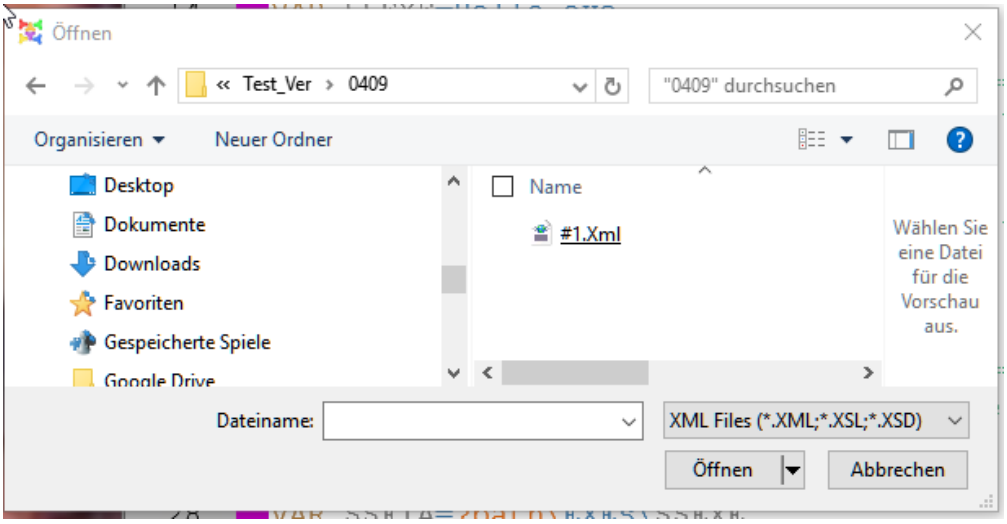
This Script will create these folder with the name of the LanguageID of the Manifest-Resource in `?path\`.



Inside that Folder you will find the extracted Manifest as a "`.XML`"-File.

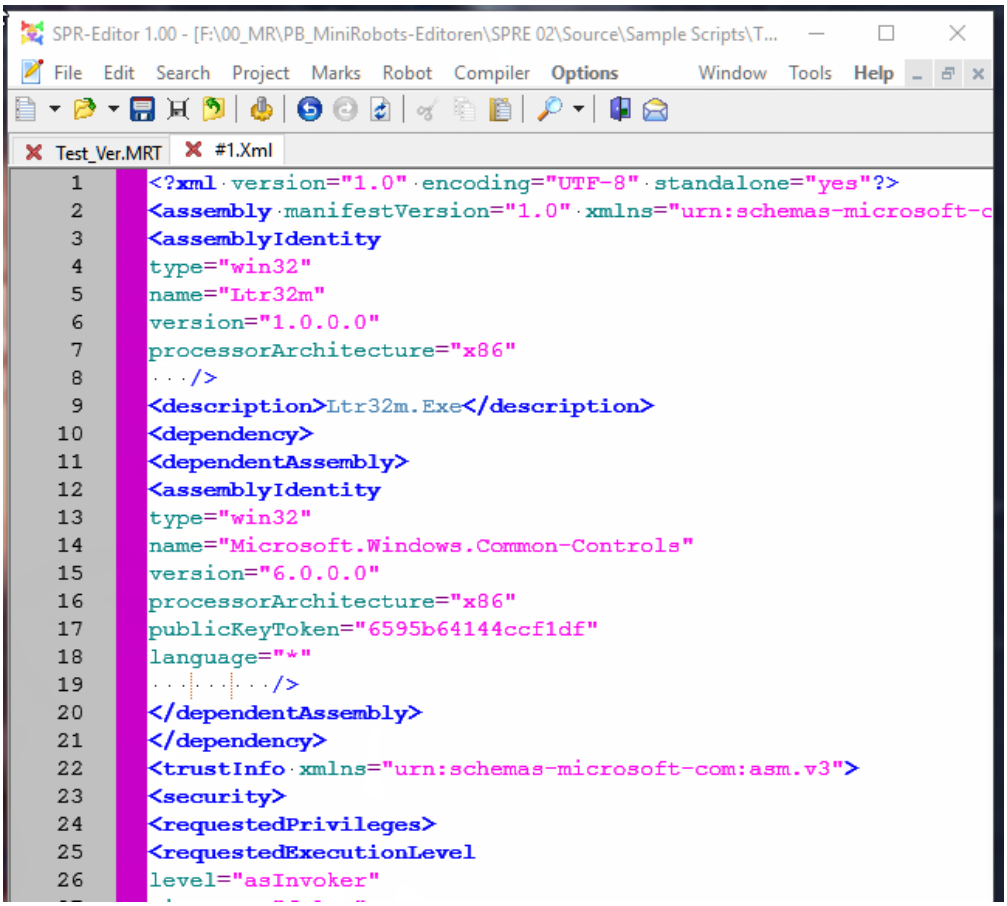


You can Load XML-Files with SPRE, just change the Filter to "XML-Files":



This may look somehow like here:

© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

RES.Last Manifest|P1
RES.LM|P1

Parameter Explanation

P1 – Variable that will get the Path to the last extracted Manifest-File.

Example

```
' This is the path to the Source-File to be extracted
    VAR.$$FIA=?path\EXEs\Hello.exe
   ' The second Parameter is the Path to the root of all "Target Folders".
 RES.Extract Manifest|$$FIA|?path\
MBX.$tos$
' To analyze the Manifest in the Script, you can use
RES.Last Manifest|$$TXT
VAR.$$TXA=$tos$
' The Manifest is now in $$TXA
MBX.$$TXA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Version 2973
- 2.3.0. '#MAN: - Manifest for Executable 321

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.12 RES.Last Version

MiniRobotLanguage (MRL)

RES.Last Version

Get the Path and Data from the last extracted Version Information

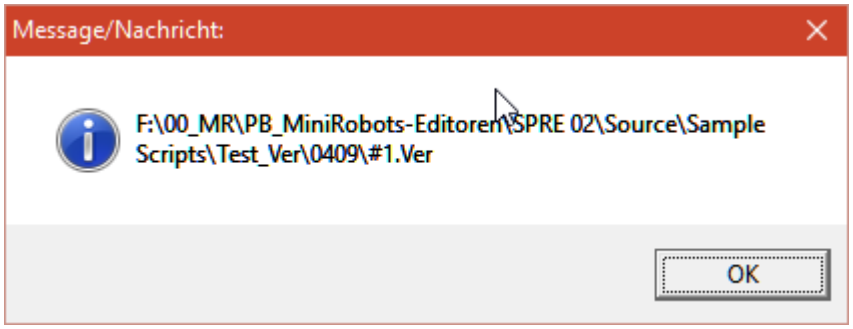
Intention

This command is used to get the Data from the last extracted Version Information.
The command will get the Path to the last extracted Version-File into a Variable, and the Data of the File on the Top Of Stack `$tos$`.
To see the effect of the Program, its necessary to first "extract" an Version Information.
The command can also be used after an "`RES.Extractr All`".

Here is a Sample-Script.

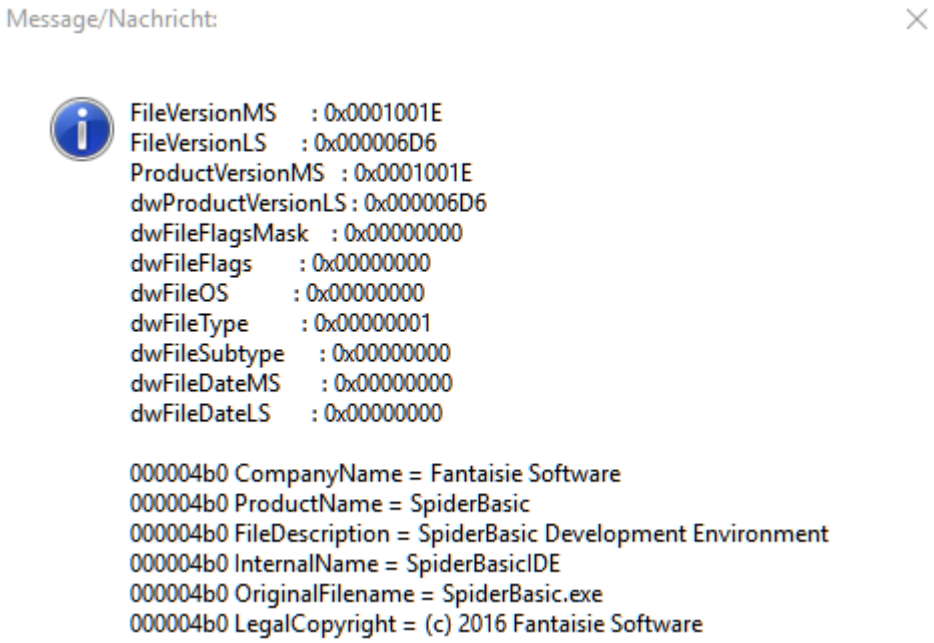
```
' This is the path to the Source-File to be extracted
VAR.$$FIA=?path\File.exe
' The second Parameter is the Path to the root the "Target Folder".
RES.Extract Version|$$FIA|?path\
MBX.$tos$
' To analyze the Version-Information in the Script, you can use
RES.Last Version|$$TXT
VAR.$$TXA=$tos$
' The Version-Info is now in $$TXA
MBX.$$TXA
ENR.
```

This Script will display fist this Messagebox with the path to the "`.Ver`"-File.:

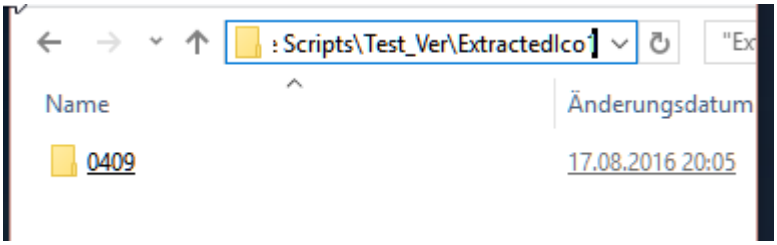


Secondly it will display a Messagebox with the Content of the "`.Ver`"-file.
The following picture shows only a part.

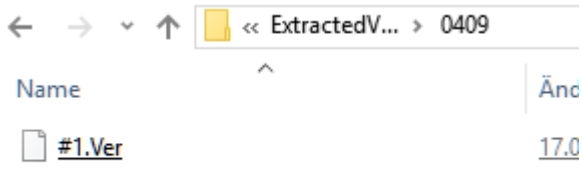
© 2023 Theo Gottwald * https://www.smart-package.com



This Script will create these folder with the name of the LanguageID for each available Version-Resource, in `?path\`.

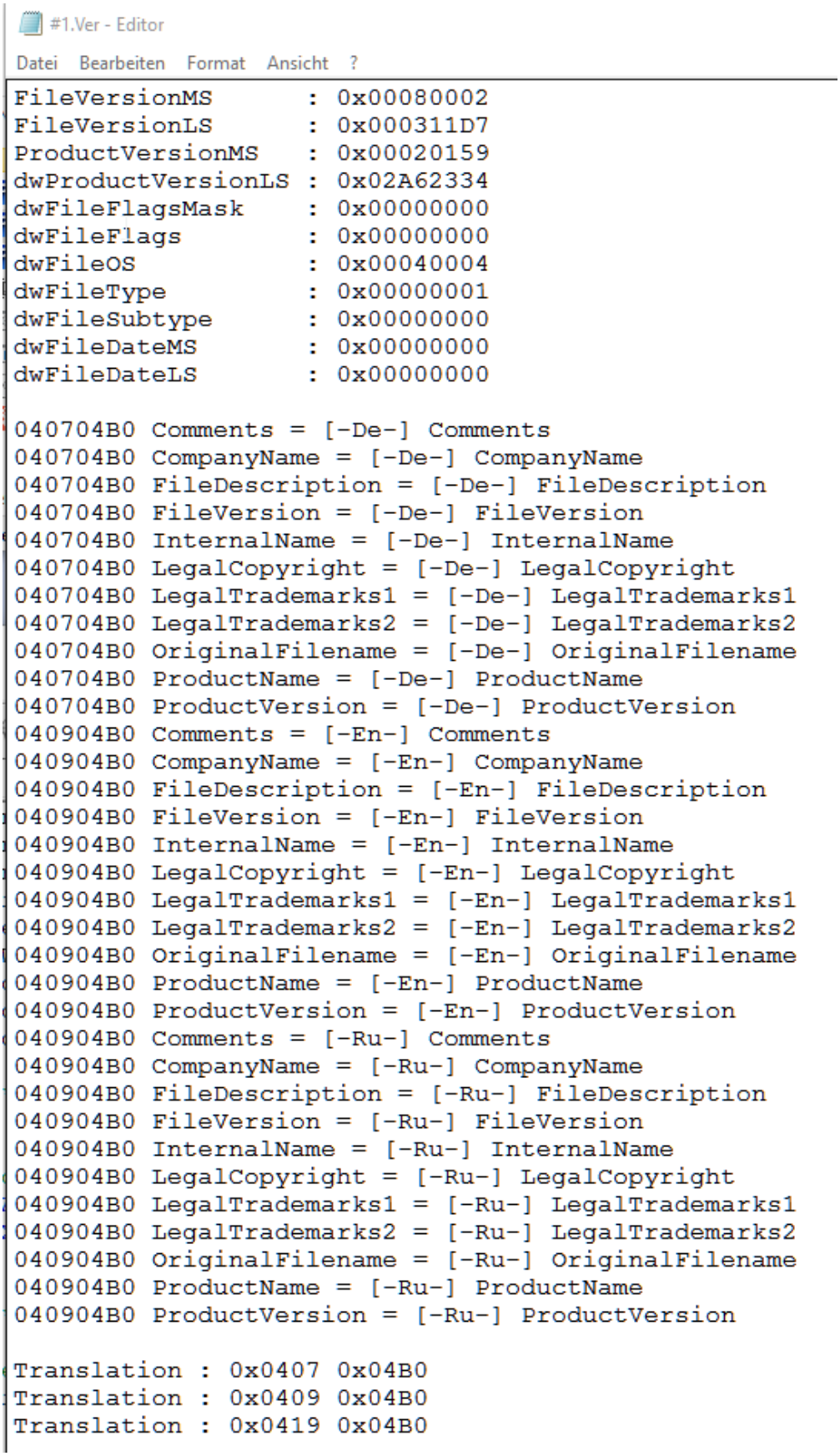


Inside that Folder you will find the extracted Version Information for each Language-ID.



In most cases the generated "`.ver`"-File will look somehow like the file below.
It has an own "Pseudocode"-Format.
It is not possible to import this file anywhere. Its just for reading and analyzing.

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

RES.Last Version|P1
RES.LV|P1

Parameter Explanation

P1 – Variable that will get the Path to the last extracted Manifest-File.

Example

```
' This is the path to the Source-File to be extracted
   VAR.$$FIA=?path\EXEs\Hello.exe
  ' The second Parameter is the Path to the root of all "Target Folders".
 RES.Extract Version|$$FIA|?path\
MBX.$tos$
' To analyze the Version in the Script, you can use
RES.Last Version|$$TXT
VAR.$$TXA=$tos$
' The Version-Data is now in $$TXA
MBX.$$TXA
ENR.
```

Remarks

-

Limitations:

-

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Icon 2978
- RES.Last Manifest 2969
- 2.4.0. '#VER: - Version-Informations 328

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.76.13 RES.Last Icon

MiniRobotLanguage (MRL)

RES.Last Icon

Get the Path and Data from the last extracted Icon

Intention

This command is used to get the Path and Data from the last extracted Icon.
This command is available rather for completeness, it may not be as useful as the other Commands (`RES.Last Version` for example).
The command will get the Path to the last extracted Icon-File into a Variable, and the Data of the File on the Top Of Stack `$tos$`.
To see the effect of the Program, its necessary to first "extract" an Icon.
The command can also be used after an "`RES.Extractr All`"

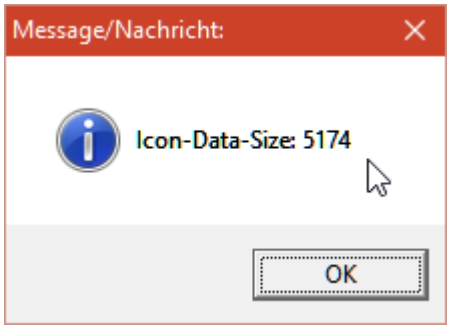
Here is a Sample-Script.

```
' This is the path to the Source-File to be extracted
VAR.$$EXE=Hello.exe
VAR.$$FIA=?path\File.exe
' The second Parameter is the Path to the root of all "Target
Folders".
RES.Extract Icon|$$FIA|?path\
MBX.$tos$
' To analyze the Icon in the Script, you can use
RES.Last Icon|$$TXT
VAR.$$TXA=$tos$
' The Icon is now in $$TXA
LEN.$$TXA|$$TXA
MBX.Icon-Data-Size: $$TXA
ENR.
```

This Script will display fist this Messagebox with the path to the "`.ico`"-File.:

© 2023 Theo Gottwald * https://www.smart-package.com

Secondly it will display a Messagebox with the Size of the ".ico"-file.



In most cases, multiple Icons are extracted. This command will only return the Path to the last Icon, that was extracted.
To get Informations about the other icons, it makes sense to use the result from this command and use other commands.
Here is an example that will display all path's from all extracted Icons

```
 This is the path to the Source-File to be extracted
VAR.$$EXE=Hello.exe
VAR.$$FIA=?path\File.exe
' The second Parameter is the Path to the root of all "Target Folders".
RES.Extract Icon|$$FIA|?path\
MBX.$tos$
' To analyze the Icon in the Script, you can use
RES.Last Icon|$$TXT
VAR.$$TXA=$tos$
' The Icon is now in $$TXA
LEN.$$TXA|$$TXA
MBX.Icon-Data-Size: $$TXA
GFT.h|$$TXT|$$PAT
MBX.$$PAT
FEF.DoIt|$$TMP|$$PAT
MBX.Ready
ENR.

:DoIt
PRT.$$TMP
RET.
ENR.
```

In case the result has a multi-language resource, you can use the Script below..

```
' This is the path to the Source-File to be extracted
VAR.$$EXE=Hello.exe
VAR.$$FIA=?path\File.exe
' The second Parameter is the Path to the root of all "Target Folders".
RES.Extract Icon|$$FIA|?path\
MBX.$tos$
' To analyze the Icon in the Script, you can use
RES.Last Icon|$$TXT
VAR.$$TXA=$tos$
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' The Icon is now in $$TXA
LEN.$$TXA|$$TXA
MBX.Icon-Data-Size: $$TXA
GFT.c|$$TXT|$$PAT
' Only needed for Multi-Language Resources
BLB.$$PAT|$$RES
MBX.$$PAT$crlf$->$$RES
FEF.DoIt|$$TMP|$$RES
MBX.Ready
ENR.

:DoIt
PRT.$$TMP
RET.
ENR.
```

Syntax

RES.Last Icon|P1
RES.LI|P1
RES.Icon Path|P1
RES.IP|P1

Parameter Explanation

P1 – Variable that will get the Path to the last extracted Icon-File.

Example

```
' This is the path to the Source-File to be extracted
VAR.$$EXE=Hello.exe
VAR.$$FIA=?path\File.exe
' The second Parameter is the Path to the root of all "Target Folders".
RES.Extract Icon|$$FIA|?path\
MBX.$tos$
' To analyze the Icon in the Script, you can use
RES.Last Icon|$$TXT
VAR.$$TXA=$tos$
' The Icon is now in $$TXA
LEN.$$TXA|$$TXA
MBX.Icon-Data-Size: $$TXA
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Ressourcen Commands 2931
- RES. - Manage Resources 2936
- RES.Refresh Explorer 2948
- RES.Change Icon 2938
- RES.Change Manifest 2941
- RES.Change Version 2944
- RES.Extract All 2950
- RES.Extract Icon 2955
- RES.Extract Manifest 2958
- RES.Extract Version 2964
- RES.Last Manifest 2969
- RES.Last Version 2973
- 2.2.1 '#ICO: - Choose own Icon 313
- 2.2.2 '#ICF: - Icon from Icon-Folder 316
- 2.2.3 '#ICS: / '#ICS: Icon-Set 318

© 2023 Theo Gottwald * https://www.smart-package.com

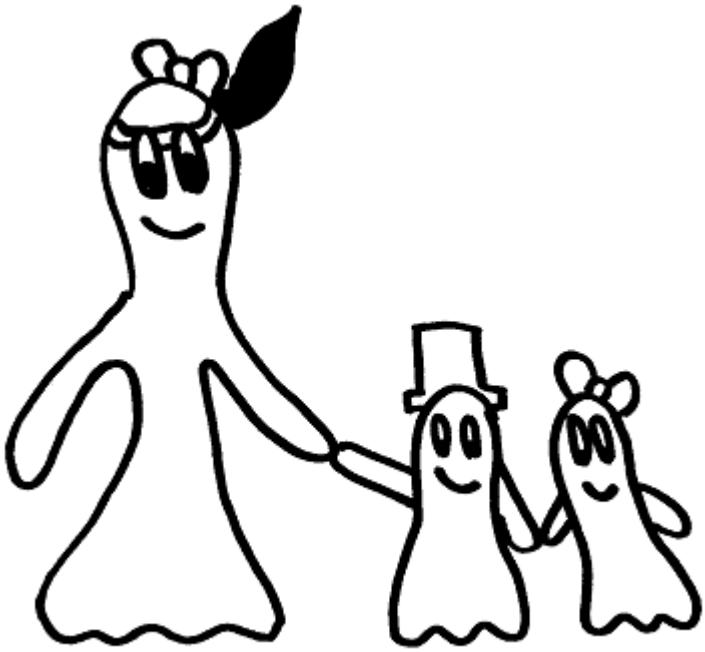
3.77 SCS. - Select from Mutiple Cases

3.77.1 ! Select from multiple choices

Smarty's Multiple Choice Commands

Previous 2146 Top 41 Next 2983

Smarty's Multiple Choice Commands



- 1.6.1. Program Flow Control 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.2 1. SCS. - Select Case

MiniRobotLanguage (MRL)

SCS. Command

Select CaSe

Intention

This command implements a multiple-choice decision ("SELECT CASE"). You have 3 ways to use this Command:

1. Use it together with CAS. .. CAE. and ESC. / ESL. to Test for Textual or Character Strings.

See example below.

```
$$TXT=Karl
SCS.$$TXT
CAS.Karl
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will come if no other choice was true
ESC.
MBX.Ready
ENR.
```

2. Use it together with CAN. .. CAE. and ESC. / ESL. to Test for numbers or formulas (numerical compare).

See example below.

```
$$NUM=14
SCS.$$NUM
CAN.14
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3. Use it together with CFP. .. CAE. and ESC. / ESL. to Test for Search Patterns 1862 in the current localized Top-Window.

See example below.

```
STW.Act|TFrmHelpMan|- Help
$$PA1=?pattern\Tabelle.bmp
$$PA2=?pattern\SCS02.bmp
$$PA3=?pattern\JK_8.bmp
SCS.1
 CFP.$$PA1
  MBX.Pattern found: $$PA1
 CFP.$$PA2
  MBX.Pattern found: $$PA2
 CFP.$$PA3
  MBX.Pattern found: $$PA3
ESL.
MBX.OK
ENR.
```

Syntax

SCS.P1

Parameter Explanation

P1 **-** String which shall be compared to the CAS./CAN.-Cases.

Example

```
'*****************************************
' Simple Demo for SCS., CAS., CAE., ESC.
'*****************************************
: §§TXT=Karl Kovse
SCS.§§TXT
CAS.Karl
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready
ENR.

'*****************************************
' Nested Sample for SCS., CAS., CAE., ESC. and CAN.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************
FOR.$$LOP|1|20
  VAR.$$TXT=Peter$$LOP
  VAR.$$NUM=500
  PRT.TXT=$$TXT
  SCS.$$TXT
  CAS.Was first case.
    PRT.First Case
' This one will come in the twentieth
  CAS.{Rufus&OR:Peter2}
    PRT.Was "Peter2..."
' You can jump out of the structure directly with GOTO.
' And you can nest unlimited many SCS.
    SCS.$$NUM
    CAN.(100*5)
      PRT.You can nest it unlimited.
    CAN. 499
  ' You can even intermix CAS. / CAN.
    CAS. Intermixed
    CAE.
      PRT.THis will not come
    ESC.
    JMP.otu
  CAS.Third
    PRT.Was Third
  CAS.{Peter&AND:3}
    PRT.Was "Peter" and "3"
' If you need a perfect match ...
  CAS.{&EXACT:Theo}
    PRT.This one is Theo.
  CAS.Fourth.
    PRT.Was this one.
' CASE ELSE
  CAE.
    PRT.Was nothing of all.
' END SELECT
  ESC.
  :otu
NEX.
MBX.Ending
:enz
ENR.
'-----------------------------------------------------------
```

Remarks

Please do not write code between the `SCS.` and the first following `CAS./CAN.`
While this code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

There are no relevant nesting limitations. if you do not understand some behavior of the CAS., then please take a look on the "Standard Search Patterns".
These are used inside the CAS.

See also:

- **1.6.1. Program Flow Control** 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.3 2. CAS. - Case-String

MiniRobotLanguage (MRL)

CAS. Command

Case String

Intention

This command can be used to script multiple-choice decisions.
It has to be used together with:

`SCS., CAN., CAE., and ESC..`

See example below.

```
: §§TXT=Karl Kovse
SCS.§§TXT
CAS.Karl
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready
ENR.
```

Syntax

CAS.[P1]

Parameter Explanation

P1 - String which shall be compared to the `SCS.`-String. Can use Standard-Search Patterns 341 like:
`{Yes&OR:Qui}` see chapter with explanation of standard search patterns.
If **P1** is omitted, only an empty string will meet the condition.

Example

```
'*****************************************
' Demo for SCS., CAS., CAE., ESC. and CAN.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************
' SCS.-Test
'DBM.2
'-------------------------
FOR.$$LOP|1|20
VAR.$$TXT=Peter$$LOP
VAR.$$NUM=500
PRT.TXT=$$TXT
SCS.$$TXT
CAS.Was first case.
PRT.First Case
' This one will come in the twentieth
CAS.{Rufus&OR:Peter2}
PRT.Was "Peter2..."
' You can jump out of the structure directly with GOTO.
' And you can nest unlimited many SCS.
SCS.$$NUM
CAN.(100*5)
PRT.You can nest it unlimited.
CAN. 499
' You can even intermix CAS. / CAN.
CAS. Intermixed
CAE.
PRT.THis will not come
ESC.
JMP. otu
CAS.Third
PRT.Was Third
CAS.{Peter&AND:3}
PRT.Was "Peter" and "3"
' If you need a perfect match ...
CAS.{&EXACT:Theo}
PRT.This one is Theo.
CAS.Fourth.
PRT.Was this one.
' CASE ELSE
CAE.
PRT.Was nothing of all.
' END SELECT
ESC.
:otu
NEX.
MBX.Ending
:enz
END.
```

Remarks

Please do not write code between the SCS. and the first following CAS./CAN. While this code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

There are no relevant nesting limitations. if you do not understand some behavior of the CAS., then please take a look on the "Standard Search Patterns". These are used inside the CAS.

See also:

- **3.2 Standard-Search Pattern** 341
- **1.6.1. Program Flow Control** 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.4 3. CAN. - Case-Numeric

MiniRobotLanguage (MRL)

CAN. Command

Case Numeric

Intention

This command can be used to script multiple-choice decisions using numerical choice.
It has to be used together with:

`CAS.`, `SCS..`, `CAE.`, and `ESC..`

See example below.

```
$$NUM=3
GSB.decide
$$NUM=7
GSB.decide
$$NUM=4
GSB.decide

ENR.
*---------------------
:decide
SCS.$$NUM
CAN.2
  DBP.I am here with 2
CAN.(2+1)
  DBP.I am here with 3
CAN.4
  DBP.I am here with 4
CAE.
  DBP. Anything else
ESC.
RET.
```

It will also work with Floating Point numbers:

```
$$NUM=6.012
SCS.$$NUM
CAN.6
  DBP.I am here with 2
CAN.(5+1)
  DBP.I am here with 3
CAN.(6+0.012)
  DBP.I am here with 6.012
CAE.
  DBP. Anything else
ESC.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Syntax

CAN . P1

Parameter Explanation

P1 - Numeric value or variable which shall be compared to the `SCS`.-Parameter. Can use Formula in Brakets like: `($$NUM*$$EXP+($$REZ))`
which is evaluated at runtime. Actually the result is compared as string to the Parameter from `SCS`. Supports Floating-Point or Integer Numeric types.

Speed in Ticks:

This command uses typically around 18(fixed number as Parameter) to 130 Ticks (Variable as Parameter). If you use difficult Formula, even more.

Example

```
'*****************************************
' Demo for SCS., CAS., CAE., ESC. and CAN.
'*****************************************
' SCS.-Test
'DBM.2
'-----------------------------------------------------------
FOR.$$LOP|1|20
  VAR.$$TXT=Peter$$LOP
  VAR.$$NUM=500
  PRT.TXT=$$TXT
  SCS.$$TXT
  CAS.Was first case.
    PRT.First Case
' This one will come in the twentieth
  CAS.{Rufus&OR:Peter2}
    PRT.Was "Peter2..."
' You can jump out of the structure directly with GOTO.
' And you can nest unlimited many SCS.
    SCS.$$NUM
    CAN.(100*5)
      PRT.You can nest it unlimited.
    CAN. 499
' You can even intermix CAS. / CAN.
    CAS. Intermixed
    CAE.
      PRT.THis will not come
    ESC.
    JMP. otu
  CAS.Third
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    PRT.Was Third
  CAS.{Peter&AND:3}
    PRT.Was "Peter" and "3"
' If you need a perfect match ...
  CAS.{&EXACT:Theo}
    PRT.This one is Theo.
  CAS.Fourth.
    PRT.Was this one.
' CASE ELSE
  CAE.
    PRT.Was nothing of all.
' END SELECT
  ESC.
  :otu
NEX.
MBX.Ending
:enz
END.
```

Remarks

please do not write code between the SCS. and the first following CAS./CAN. While this code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

There are no relevant nesting limitations.Please always make formulas into () or they may not be calculated.

See also:

- 1.6.1. Program Flow Control 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.5 4. CFP. - Case-Pattern

MiniRobotLanguage (MRL)

CFP. Command

Case Find Pattern

Intention

This command can be used to script multiple-choice decisions using a graphical Pattern 1862.
It has to be used together with:

`SCS., CAE., and ESC. and ESL.`
It can be nested to any depth.

The Parameter behind `SCS.` will specify how `CFP.` will work.
There are 3 choices:

`SCS.0`
The given Pattern must NOT be found for the condition to be true.

`SCS.1 or SCS.(positive number)`
The given Pattern must be found at least as often as the number specifies, for the condition to be true.

`SCS.-1 or SCS.(negative number)`
The given Pattern must be found exactly as often as the number specifies, for the condition to be true.

See example below.

```
STW.Act|TFrmHelpMan|- Help
$$PA1=?pattern\Tabelle.bmp
$$PA2=?pattern\SCS02.bmp
$$PA3=?pattern\JK_8.bmp
SCS.1
 CFP.$$PA1
  MBX.Pattern found: $$PA1
 CFP.$$PA2
  MBX.Pattern found: $$PA2
 CFP.$$PA3
  MBX.Pattern found: $$PA3
ESL.
MBX.OK
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

CFP.P1[|P2][|P3]

Parameter Explanation

P1 - Filename of Pattern-File for details on how to use Pattern-Files see the LFP.-Command.
The Pattern-File does not in any way relate to the Parameter given with SCS.

P2 - opt. Destination Point Shift, will be used to move the target point by a specified amount of Pixels.
For example, if the Target is found at 100,50 and you specify here -10,40 then the final Point will be at 90,90.
P3 - opt. Variable for Result. if omitted the result will be placed on TOS.

Speed in Ticks:

This is a slow command, which is possibly in the range of 0.2 to 1 seconds, depending on the Pattern-Size and complexity.

Example

```
'*****************************************
' Demo for SCS., CAS., CAE., ESC. and CAN.
'*****************************************
' SCS.-Test
'DBM.2
'-------------------------------------------------------------
FOR.$$LOP|1|20
  VAR.$$TXT=Peter$$LOP
  VAR.$$NUM=500
  PRT.TXT=$$TXT
  SCS.$$TXT
  CAS.Was first case.
    PRT.First Case
' This one will come in the twentieth
  CAS.{Rufus&OR:Peter2}
    PRT.Was "Peter2..."
' You can jump out of the structure directly with GOTO.
' And you can nest unlimited many SCS.
    SCS.$$NUM
    CAN.(100*5)
      PRT.You can nest it unlimited.
    CAN. 499
' You can even intermix CAS. / CAN.
    CAS. Intermixed
    CAE.
      PRT.THis will not come
    ESC.
    JMP. otu
  CAS.Third
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    PRT.Was Third
  CAS.{Peter&AND:3}
    PRT.Was "Peter" and "3"
' If you need a perfect match ...
  CAS.{&EXACT:Theo}
    PRT.This one is Theo.
  CAS.Fourth.
    PRT.Was this one.
' CASE ELSE
  CAE.
    PRT.Was nothing of all.
' END SELECT
  ESC.
  :otu
NEX.
MBX.Ending
:enz
END.
```

Remarks

please do not write code between the SCS. and the first following CAS./CAN. While this code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

There are no relevant nesting limitations.Please always make formulas into () or they may not be calculated.

See also:

- 1.6.1. Program Flow Control 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.6 5. CAE. - Case-Else

MiniRobotLanguage (MRL)

CAE. Command

Case Else

Intention

This command can be used together with:

- SCS.
- CAN.
- ESC.
- CAS.

to build multiple-choice constructs.

CAE always evaluates to TRUE if none of the previous constructs has evaluated TRUE.

See example below.

```
: §§TXT=Karl Kovse
SCS.§§TXT
CAS.Karl
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready
ENR.
```

Syntax

CAE.

Parameter Explanation

This command does not need any parameters.

Speed in Ticks:

© 2023 Theo Gottwald * https://www.smart-package.com

This command uses typically around 14 Ticks.

Example

```
'*****************************************
' Demo for SCS., CAS., CAE., ESC. and CAN.
'*****************************************
' SCS.-Test
'DBM.2
FOR.$$LOP|1|20
  VAR.$$TXT=Peter$$LOP
  VAR.$$NUM=500
  PRT.TXT=$$TXT
  SCS.$$TXT
  CAS.Was first case.
    PRT.First Case
' This one will come in the twentieth
  CAS.{Rufus&OR:Peter2}
    PRT.Was "Peter2..."
' You can jump out of the structure directly with GOTO.
' And you can nest unlimited many SCS.
    SCS.$$NUM
    CAN.(100*5)
      PRT.You can nest it unlimited.
    CAN. 499
' You can even intermix CAS. / CAN.
    CAS. Intermixed
    CAE.
      PRT.THis will not come
    ESC.
    JMP. otu
  CAS.Third
    PRT.Was Third
  CAS.{Peter&AND:3}
    PRT.Was "Peter" and "3"
' If you need a perfect match ...
  CAS.{&EXACT:Theo}
    PRT.This one is Theo.
  CAS.Fourth.
    PRT.Was this one.
' CASE ELSE
  CAE.
    PRT.Was nothing of all.
' END SELECT
  ESC.
  :otu
NEX.
MBX.Ending
:enz
END.
```

Speed in Ticks:

This command uses typically around 15 Ticks.

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Please do not write code between the SCS. and the first following CAS./CAN.
While this code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

There are no relevant nesting limitations.

Here is a general hint:
Please always make formulas into (...) to make sure they will be calculated.

See also:

- 1.6.1. Program Flow Control 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.7 6. ESC. - End-Select-Case

MiniRobotLanguage (MRL)

ESC. Command

End Select Case

This command can be used together with:

- SCS.
- CAN.
- CAE.
- CAS.

to build multiple-choice constructs.

ESC. (End Select Case) will close the multiple choice decision block.

See example below.

```
: §§TXT=Karl Kovse
SCS.§§TXT
CAS.Karl
  PRT.This will come
CAS.Peter
  PRT.This will not come
CAE.
  PRT. This will not come
ESC.
MBX.Ready
ENR.
```

Syntax

ESC.

Parameter Explanation

this command takes no parameters.

Speed in Ticks:

This command uses typically around 7 Ticks.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*****************************************
' Demo for SCS., CAS., CAE., ESC. and CAN.
'*****************************************
' SCS.-Test
'DBM.2
FOR.$$LOP|1|20
  VAR.$$TXT=Peter$$LOP
  VAR.$$NUM=500
  PRT.TXT=$$TXT
  SCS.$$TXT
  CAS.Was first case.
    PRT.First Case
' This one will come in the twentieth
  CAS.{Rufus&OR:Peter2}
    PRT.Was "Peter2..."
' You can jump out of the structure directly with GOTO.
' And you can nest unlimited many SCS.
    SCS.$$NUM
    CAN.(100*5)
      PRT.You can nest it unlimited.
    CAN. 499
' You can even intermix CAS. / CAN.
    CAS. Intermixed
    CAE.
      PRT.THis will not come
    ESC.
    JMP. otu
  CAS.Third
    PRT.Was Third
  CAS.{Peter&AND:3}
    PRT.Was "Peter" and "3"
' If you need a perfect match ...
  CAS.{&EXACT:Theo}
    PRT.This one is Theo.
  CAS.Fourth.
    PRT.Was this one.
' CASE ELSE
  CAE.
    PRT.Was nothing of all.
' END SELECT
  ESC.
  :otu
NEX.
MBX.Ending
:enz
END.
```

Remarks

Please do not write code between the `SCS.` and the first following `CAS./CAN.`

© 2023 Theo Gottwald * https://www.smart-package.com

While the code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

-

There are no relevant nesting limitations.
Please always make formulas into () or they may not be calculated.

See also:

- 1.6.1. Program Flow Control 121
- 1. SCS. - Select Case 2983
- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.77.8 7. ESL. - End Select Loop

ESC. - End-Select-Case Previous 2996 Top 41 Next 3005

MiniRobotLanguage (MRL)

ESL. Command

End Select Loop

The ESL.-Commandreplaces the ESC.-Command in a SCS. ... CAE. ... ESC. Construct.
In case the Program did not match ANY of the given Conditions, the ESL. will jump back to the SCS.
If there was a condition, ESL. will exit the Construct and continue with the Script.

Do not use CAE. if you use ESL. Otherwise the ESL. will take the CAE. as a valid Option and end the Construct.

This command can be used together with:

- SCS.
- CAN.
- CAE. - do not use CAE. if you use ESL.
- CAS.
- CFP.

to build multiple-choice constructs.

ESC. (End Select Case) will close the multiple choice decision block.

See example below.

```
STW.Act|TFrmHelpMan|- Help
$$PA1=?pattern\Tabelle.bmp
$$PA2=?pattern\SCS02.bmp
$$PA3=?pattern\JK_8.bmp
SCS.1
 CFP.$$PA1
  MBX.Pattern found: $$PA1
 CFP.$$PA2
  MBX.Pattern found: $$PA2
 CFP.$$PA3
  MBX.Pattern found: $$PA3
ESL.
MBX.OK
ENR.
```

Syntax

ESL.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Parameter Explanation

this command takes no parameters.

Speed in Ticks:

-

Example

```
'*****************************************
' ESL. Sample
'*****************************************

  STW.Act|TFrmHelpMan|- Help & Manual

  $$PA1=?pattern\Tabelle.bmp
  $$PA2=?pattern\SCS02.bmp
  $$PA3=?pattern\JK_8.bmp
  NEF.$$PA1
    MBX.File existiert nicht: $$PA1
  EIF.
  SCS.1
  CFP.$$PA3
    MBX.PA3
  CFP.$$PA1
    MBX.PA2
  ESL.
  MBX.OK
  ENR.
```

Remarks

Please do not write code between the `SCS.` and the first following `CAS./CAN./CFP.`
While the code will actually be executed, you could get memory-leaks if you jump out of the construct from that place.

Limitations:

-
There are no relevant nesting limitations.
Please always make formulas into () or they may not be calculated.

See also:

- 1.6.1. Program Flow Control 121
- 1. SCS. - Select Case 2983

© 2023 Theo Gottwald * https://www.smart-package.com

- 2. CAS. - Case-String 2987
- 3. CAN. - Case-Numeric 2990
- 4. CAE. - Case-Else 2996
- 5.CFP. - Case Find Pattern 2993
- 6.ESL. - End Select Loop 3002
- 7.ESC. - End Select Case 2999

© 2023 Theo Gottwald * https://www.smart-package.com

3.78 Stack-Operations

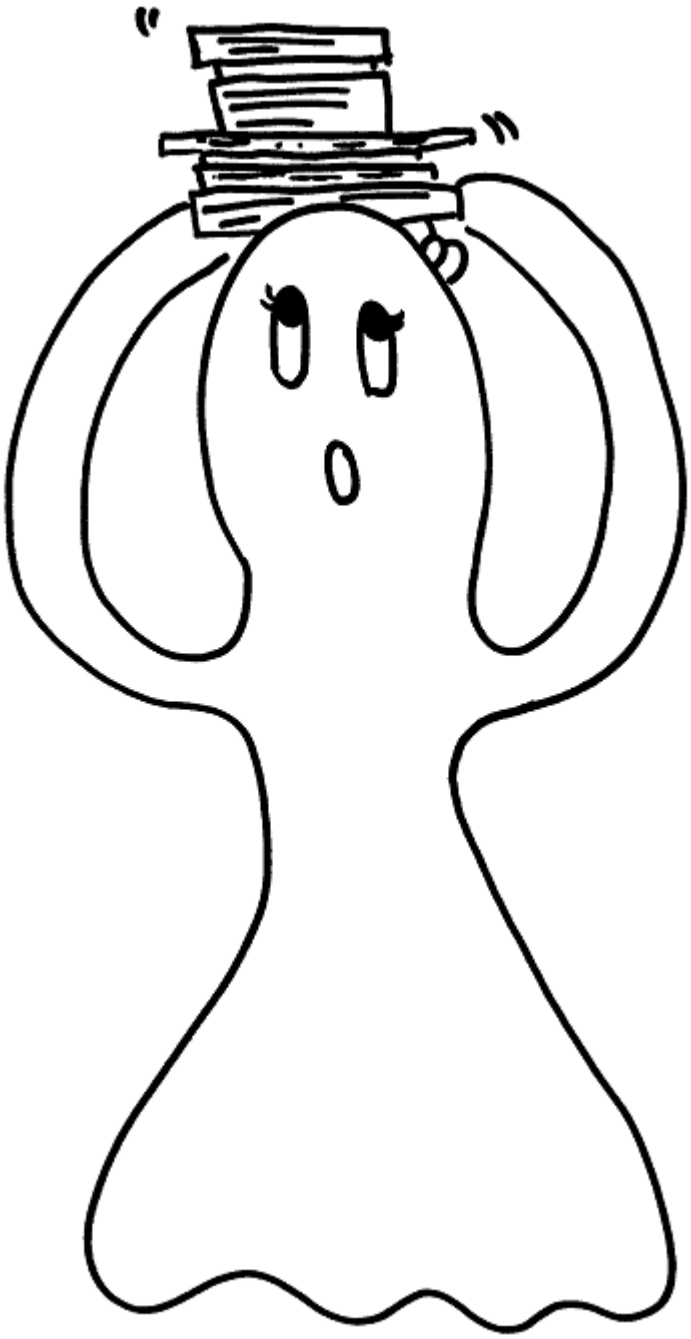
3.78.1 ! Smarty's Stack Commands

Smarty's Stack Commands

Previous 2999 Top 41 Next 3007

Smarty's Stack Commands

© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com

3.78.2 POP. - POP Variable from Stack

POP. - POP Variables from User-St Previous 3005 Top 41 Next 3010

MiniRobotLanguage (MRL)

POP. Command

POP variable from user stack, without resolution

Intention

Takes off the Top of the current User-Stack and trim left and right side, then put it into the Variable in **P1**.
Will resolve Variables which are on the stack.
Takes the TOS into the Variable behind `POP`. Counterpart to `PUS.`, `PUV.` `PSS.`, `PNS.`, and `PVS`.- these will Push Variables onto Stack with different sorts of Variable-Resolution.
If you do not want the content to be trimmed, use POV. instead.

You can `POP.` any Stack Content into the variable.
While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. `PUV.` `PSS.`, `PNS` do just that.
To avoid Variable Resolution when taking things from the Stack, you can use instead of POP. you can use the POV. 3010-Command the following code with VAO. and $$000.

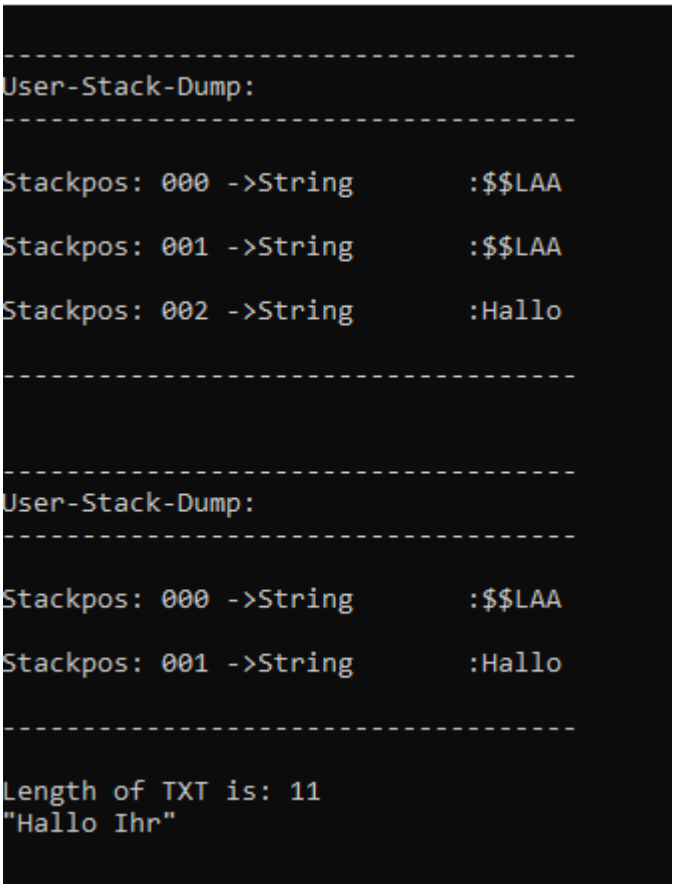
```
VAO.$$VAR=$$000
```

PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
PUV. - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with |"|". Just the whole Parameter will be placed on TOS.

You can try this Script and see the stack and the return.

```
$$LAA="Hallo Ihr"
PVS.Hallo|$$LAA|$$LAA
DMP.6
POP.$$TXT
DMP.6
LEN.$$TXT|$$LEN
PRT.Length of TXT is: $$LEN
PRT.$$TXT
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

POP.P1

Parameter Explanation

P1 - `Variable` that will take the result from TOS: The result will be trimmed and variables will be resolved.

If you omit the needed Parameter, POP. will set the timeout-flag and just continue to execute without doing anything.

`POP.` will only take 1 Parameter, if you need more use `PUV.`

```
The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which
depends on Settings done with STS.-Command.

If the Popped items are deleted from stack or not also depends
on the Settings done with STS.

Please note, that Special Folders are resolved before
Variables, therefore Special-Folders that may be unresolved
inside Variables wont be touched.
```

Speed in Ticks:

© 2023 Theo Gottwald * https://www.smart-package.com

This command uses typically around 90 Ticks.

Example

```
' Pop Variable from User-Stack (with resolution)
PRT.Items on Stack: #tos#
: $$TXA=Laba
: $$TXB=Lax
DBP. Variables Content (1): $$TXA-$$TXB
PUV.$$TXA|$$TXB
GSB.tos
POP.$$TXC
DBP.$$TXC
GSB.tos
MBX.Ready
END.
:tos
DMP.4
DBP. Items on Stack: #tos#
DBP. Item on TOS: <$dtos$>
RET.
```

Remarks

-

Limitations:

-

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.3 POV. - POp Variables from User-Stack

MiniRobotLanguage (MRL)

POV. Command

Pop Variables from user stack

Intention

Takes up to 25 variables, which Content is been taken from the Stack/the Que.

Counterpart to `PUV.` - Push Variables onto Stack. You can however pop any Stack Content into variables.

You can `PUV.` any Stack Content into the variable(s).

While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. `PUV. PSS., PNS` do just that.
The POP.-Command will do a variable resolution and trim the result.
To avoid Variable Resolution and trimming, when taking things from the Stack, you can use POV. or this:

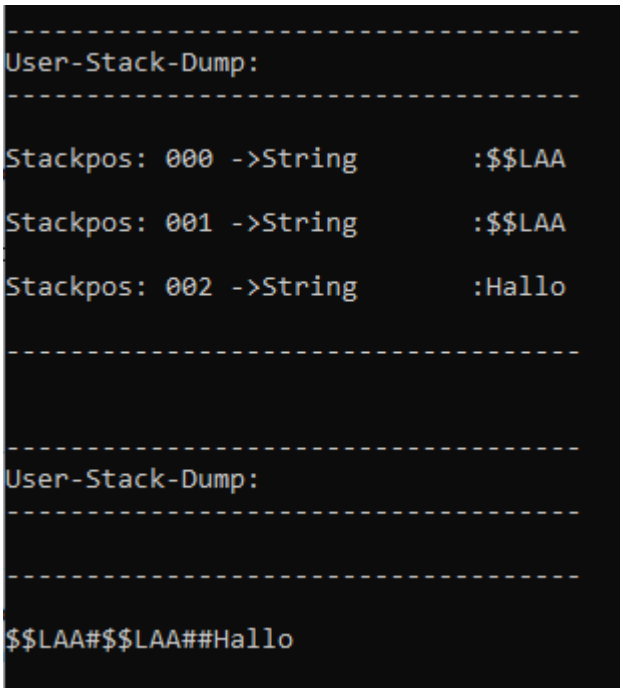
```
VAO.$$VAR=$$000
```

PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
PUV. - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with |"|". Just the whole Parameter will be placed on TOS.

Using teh following Script will bring you this Stack and Output:

```
$$LAA="Hallo Ihr"
PVS.Hallo|$$LAA|$$LAA
DMP.6
POV.$$TXT|$$TXA|$$TXB
DMP.6
PRT.$$TXT#$$TXA##$$TXB
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

POV.[P1][|P2...|P25]

Parameter Explanation

P1 **-** Variable

P2 **-** Variable P2 to Px, separated by comma or |

The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which depends on Settings done with STS.-Command.

If the popped items are being deleted from stack, depends on the Settings done with STS.

Example

```
' Pop Variables Content from User-Stack
'DBM.2
DBP.Items on Stack: #tos#
: $$TXA=Laba
: $$TXB=Lax
DBP. Variables Content (1): $$TXA-$$TXB
PUV.$$TXA|$$TXB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP. Items on Stack: #tos#
DBP. Item on TOS: <$dtos$>
DMP.4
: $$TXA=
: $$TXB=
DBP. Variables have been cleared (2): $$TXA-$$TXB
DMP.4
POV.$$TXB|$$TXA
DBP. Variables Content after POV. (3): $$TXA-$$TXB
MBX.Ready
END.
```

Remarks

-

Limitations:

25 Variables is the maximum number of possible parameters.

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POP. - POP Variable from Stack 3007

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.4 PUS. - Push Parameter onto Stack/Que

PUS. - PUsh Parameter on the User-Stack **Previous** 3010 **Top** 41 **Next** 3019

MiniRobotLanguage (MRL)

PUS. Command

Push parameter onto User Stack

Intention

Pushes the complete parameter on top of the User-Stack.
Variables or special-folders are not processed.

You can `PUV.` any Stack Content into the variable(s).

While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. `PUV. PSS., PNS` do just that.
To avoid Variable Resolution when taking things from the Stack, you can use POV. 3010 or

```
VAO.$$VAR=$$000
```

PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
PUV. - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with |"|". Just the whole Parameter will be placed on TOS.

Syntax

```
PUS.[P1]
```

Parameter Explanation

P1 – `String`

```
The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which
depends on settings done with STS.-Command.

If the popped items are deleted from stack or not also depends
on the settings done with STS.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Speed in Ticks:

This command uses typically around 23 Ticks.

Example

```
DBM.2
PRT.Items on Stack: #tos#
: $$TXA=Laba
: $$TXB=Lax
PRT. Variables Content (1): $$TXA-$$TXB
PUS.$$TXA|$$TXB
PRT. Items on Stack: #tos#
PRT. Item on TOS: <$dtos$>
DMP.4
END.
```

Remarks

-

Limitations:

-

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.5 PVS. - Push Variables on Stack/Que

MiniRobotLanguage (MRL)

PVS. Command

Push Variables onto the User Stack without resolving them

Intention

Pushes the Variabler on top of the User-Stack.
Variables or special-folders are not processed.

You can `PUV.` any Stack Content into the variable(s).

While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. does, `PUV.` `PSS.`, `PNS.` also do that.
PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
PUV. - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with |"|". Just the whole Parameter will be placed on TOS.

Using PVS. the Stack will look like this:

```
$$LAA="Hallo Ihr"
PVS.$$LAA
DMP.6
```

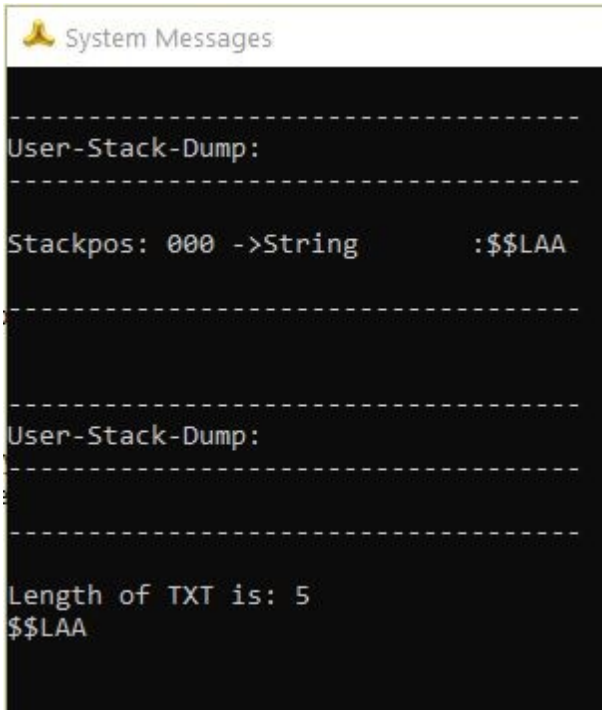
```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->String       :$$LAA

-----------------------------------
```

If you want to retrieve the Variable unresolved from the Stack, you can use VAQ 3556. and $$000.

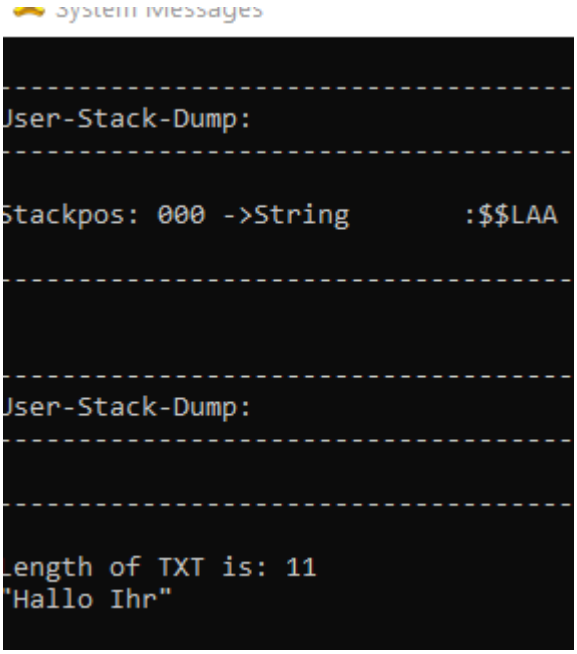
```
$$LAA="Hallo Ihr"
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
PVS.$$LAA
DMP.6
VAO.$$TXT=$$000
DMP.6
LEN.$$TXT|$$LEN
PRT.Length of TXT is: $$LEN
PRT.$$TXT
MBX.!
```



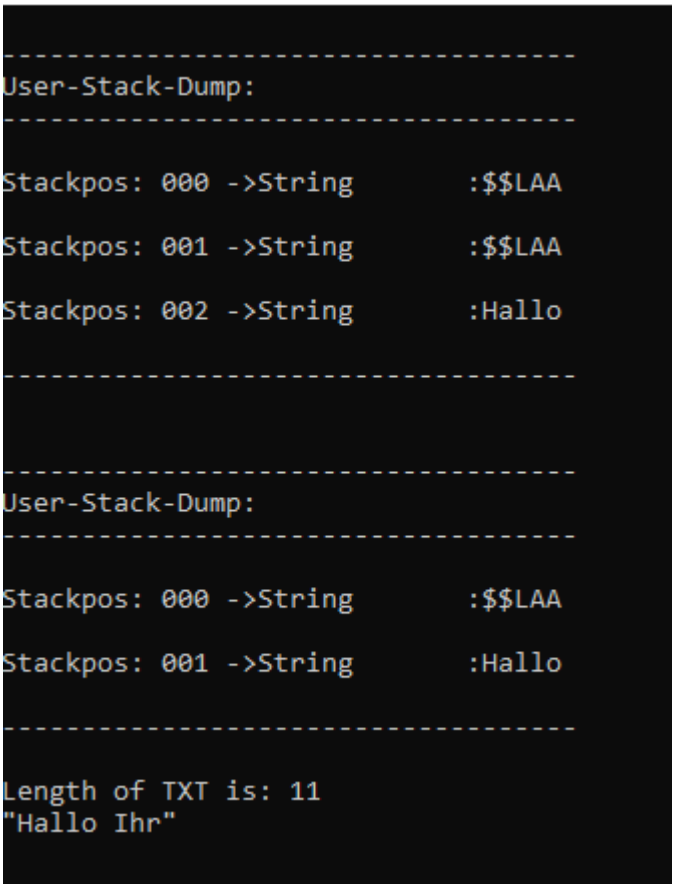
If you use POP. instead of VAO., the stack will look like that, and the Variable will be resolved:



© 2023 Theo Gottwald * https://www.smart-package.com

You can use multiple Parameters, each of these will be placed on the Stack.

```
$$LAA="Hallo Ihr"
PVS.Hallo|$$LAA|$$LAA
DMP.6
POP.$$TXT
DMP.6
LEN.$$TXT|$$LEN
PRT.Length of TXT is: $$LEN
PRT.$$TXT
```



Syntax

PVS.P1[|Px ...|P25]

Parameter Explanation

P1 - (Variable or Text) will be placed directly on TOS without resolution,
P2 ...P25 - (optional) Each of these will be placed on the Stack.

© 2023 Theo Gottwald * https://www.smart-package.com

```
The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which
depends on settings done with STS.-Command.

If the popped items are deleted from stack or not also depends
on the settings done with STS.
```

Example

```
DBM.2
PRT.Items on Stack: #tos#
: $$TXA=Laba
: $$TXB=Lax
PRT. Variables Content (1): $$TXA-$$TXB
PVS.$$TXA|$$TXB
PRT. Items on Stack: #tos#
PRT. Item on TOS: <$dtos$>
DMP.4
END.
```

Remarks

-

Limitations:

-

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.6 PUV. PUsh variables on the User-Stack

MiniRobotLanguage (MRL)

PUV. Command

PUsh Variable(s) on the user stack

Intention

Takes up to 25 variables, which content is been pushed on Top of the User-Stack/the Queue. Counterpart to `POV.` (`POP` Variables from Stack).

You can Push any variables onto the stack. PUV. will do a type-dependend variable resolution. This will only work, if the Variables have been defined with a type. Otherwise the resolution will be a Binary-compatible Variable-resolution.

You can `PUV.` any Stack Content into the variable(s).

While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. `PUV. PSS., PNS` do just that.
PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
`PUV.` - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT 3477.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with |"|". Just the whole Parameter will be placed on TOS.

Sample-Code and a Stack-Printout:

```
$$LAA="Hallo you "
PUV.People|$$LAA|$$LAA
DMP.6
POV.$$TXA|$$TXB|$$TXC
DMP.6
PRT.$$TXA#$$TXB##$$TXC
MBX.!
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

PUV.P1[|P2]...[|P25]

Parameter Explanation

P1 **-** Variable

P2 - Px **-** Variables

The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which depends on Settings done with STS.-Command.

If the Popped items are deleted from stack or not also depends on the Settings done with STS.

Speed in Ticks:

This command uses around 150 Ticks with one Parameter.

Example

```
' Pop Variables Content from User-Stack
'DBM.2
DBP.Items on Stack: #tos#
: $$TXA=Laba
: $$TXB=Lax
DBP. Variables Content (1): $$TXA-$$TXB
PUV.$$TXA|$$TXB
DBP. Items on Stack: #tos#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP. Item on TOS: <$dtos$>
DMP.4
: $$TXA=
: $$TXB=
DBP. Variables have been cleared (2): $$TXA-$$TXB
DMP.4
POV.$$TXB,$$TXA
DBP. Variables Content after POV. (3): $$TXA-$$TXB
MBX.Ready
END.
```

Remarks

-

Limitations:

25 Variables is the maximum number of possible parameters.

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.7 PNS. Push Numeric variables on the User-Stack

PNS. - PUsh Numeric Variables on the User-Stack **Previous** 3013 **Top** 41 **Next** 3030

MiniRobotLanguage (MRL)

PNS. Command

PUsh Numeric Variable(s) on the user stack

Intention

Takes up to 25 variables, which content is been pushed on Top of the User-Stack/the Queue. Counterpart to `POV.` (`POP` Variables from Stack).
As the command tries to resolve Numbers and Formulas, you can get a "0" or "1" result if you use it for strings taht can in any way be resolved to a formula. Therefore Use PSS. for Strings and PNS. for numbers.

You can Push any variables onto the stack. PUV. will do a type-dependend variable resolution. This will only work, if the Variables have been defined with a type. Otherwise the resolution will be a Binary-compatible Variable-resolution.

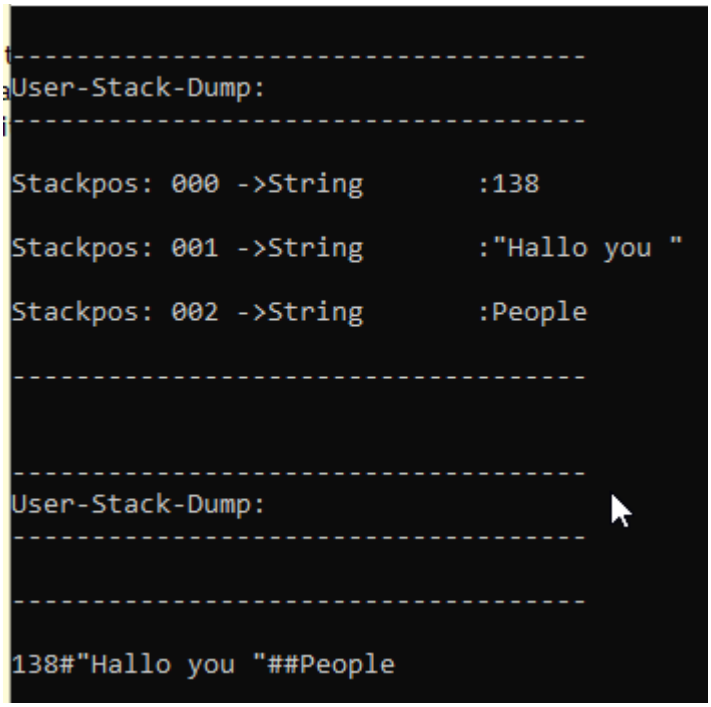
You can `PUV.` any Stack Content into the variable(s).

While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. `PUV.` `PSS.`, `PNS` do just that.
PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
`PUV.` - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT 3477.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with `|"|"`. Just the whole Parameter will be placed on TOS.

Sample-Code and a Stack-Printout:

```
$$LAA="Hallo you "
$$LAB=(1+2+3*45)
PNS.People|$$LAA|$$LAB
DMP.6
POV.$$TXA|$$TXB|$$TXC
DMP.6
PRT.$$TXA#$$TXB##$$TXC
MBX.!
```

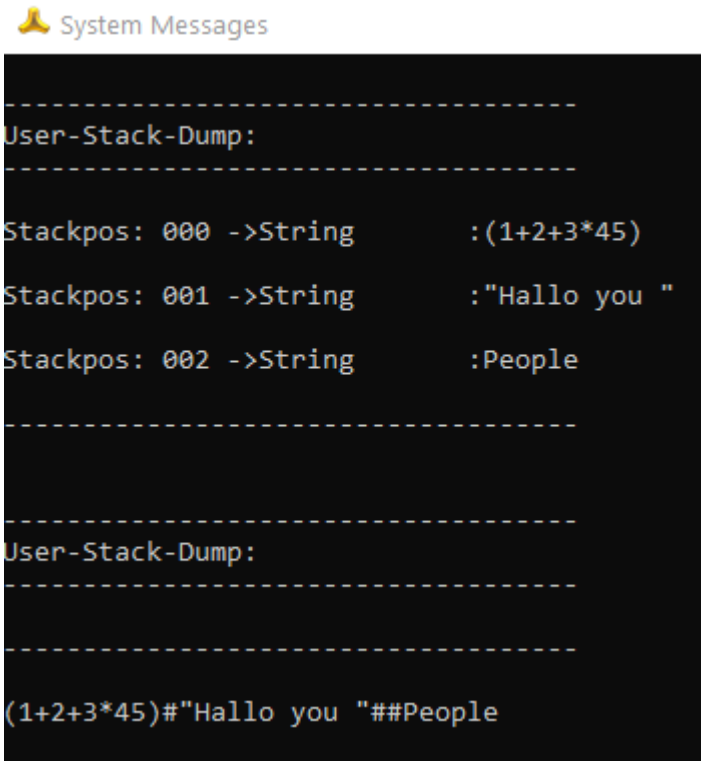

*© 2023 Theo Gottwald * https://www.smart-package.com*



Please compare the Result with the Results from PSS.

```
$$LAA="Hallo you "
$$LAB=(1+2+3*45)
PSS.People|$$LAA|$$LAB
DMP.6
POV.$$TXA|$$TXB|$$TXC
DMP.6
PRT.$$TXA#$$TXB##$$TXC
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

PNS.[P1][|P2]...[|PX]

Parameter Explanation

P1 **-** Variable

P2 - Px - (optinal) more Variables

The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which depends on Settings done with STS.-Command.

If the Popped items are deleted from stack or not also depends on the Settings done with STS.

Example

```
' Pop Variables Content from User-Stack
'DBM.2
DBP.Items on Stack: #tos#
: $$TXA=(1*10+23)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
: $$TXB=+555 E01
DBP. Variables Content (1): $$TXA-$$TXB
PNS.$$TXA|$$TXB
DBP. Items on Stack: #tos#
DBP. Item on TOS: <$dtos$>
DMP.4
: $$TXA=
: $$TXB=
DBP. Variables have been cleared (2): $$TXA-$$TXB
DMP.4
POV.$$TXB,$$TXA
DBP. Variables Content after POV. (3): $$TXA-$$TXB
MBX.Ready
END.
```

Remarks

-

Limitations:

25 Variables is the maximum number of possible parameters.

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.8 PSS. Push String variables on the User-Stack

PSS - PUsh String Variables on the User-Stack **Previous** 3013 **Top** 41 **Next** 3030

MiniRobotLanguage (MRL)

PSS. Command

PUsh String Variable(s) on the user stack

Intention

Takes up to 25 String/Text-Variables, which content is been pushed on Top of the User-Stack/the Queue. Counterpart to `POV.` (`POP` Variables from Stack).

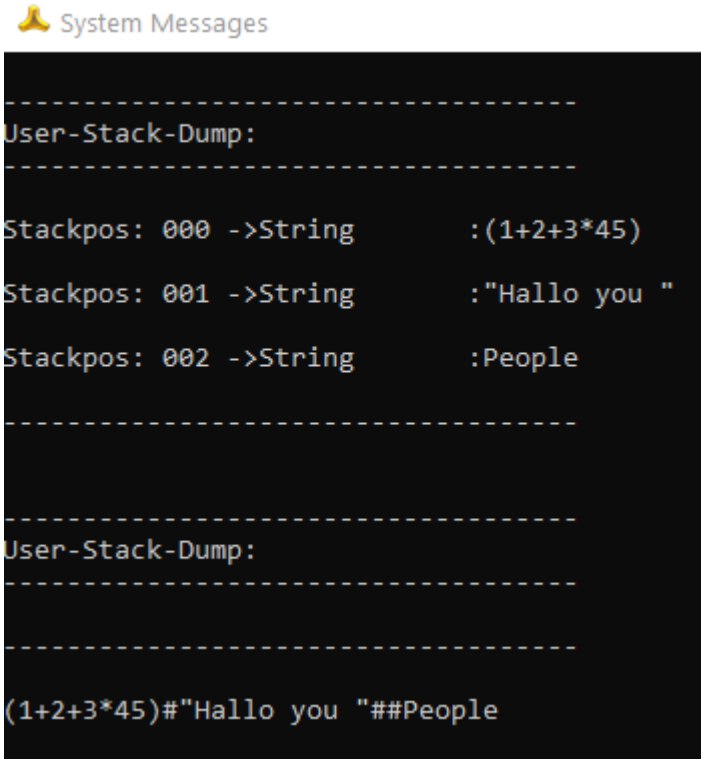
You can Push any variables onto the stack. PSS. will do a String-type variable resolution.

While `PVS.` and PUS. explicitly does NOT resolve variables and special-folder, POP. `PUV.` `PSS.`, `PNS` do just that.
PVS. - Put Variable-Name on Stack without resolving anything
PSS. - Resolve Variables and Special-Folders before putting the Variable on Stack
PNS. - Do a numerical resolution and calculate Formulas in the Variable (if they are in ()). Then put the result on TOS.
`PUV.` - Do a Variable-Type dependent Variable resolution. Variables that are defined in some way have a Variable-Type (See GVT 3477.-Command)
PUS. - Put Variable-Name on Stack without resolving anything. Unlike PVS., Variables will not be split if separated with |"|". Just the whole Parameter will be placed on TOS.

Sample-Code and a Stack-Printout:

```
$$LAA="Hallo you "
$$LAB=(1+2+3*45)
PSS.People|$$LAA|$$LAB
DMP.6
POV.$$TXA|$$TXB|$$TXC
DMP.6
PRT.$$TXA#$$TXB##$$TXC
MBX.!
```

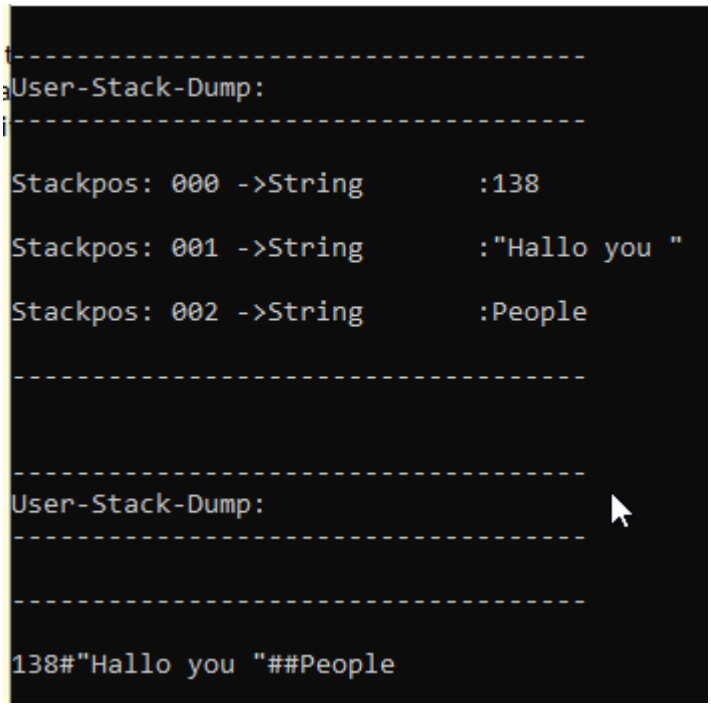

*© 2023 Theo Gottwald * https://www.smart-package.com*



Please compare the results with those from the PNS. - Command.

```
$$LAA="Hallo you "
$$LAB=(1+2+3*45)
PNS.People|$$LAA|$$LAB
DMP.6
POV.$$TXA|$$TXB|$$TXC
DMP.6
PRT.$$TXA#$$TXB##$$TXC
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

PSS.P1[|P2]...[|P25]

Parameter Explanation

P1 **-** Variables with Text or String COntent. Should not be binary content that may eventually get Variable-Expanded.

P2 - Px - (optional) More Variables

The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which depends on Settings done with STS.-Command.

If the Popped items are deleted from stack or not also depends on the Settings done with STS.

Example

```
' Pop Variables Content from User-Stack
'DBM.2
DBP.Items on Stack: #tos#
: $$TXA=Laba
: $$TXB=Lax
DBP. Variables Content (1): $$TXA-$$TXB
PSS.$$TXA|$$TXB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DBP. Items on Stack: #tos#
DBP. Item on TOS: <$dtos$>
DMP.4
: $$TXA=
: $$TXB=
DBP. Variables have been cleared (2): $$TXA-$$TXB
DMP.4
POV.$$TXB,$$TXA
DBP. Variables Content after POV. (3): $$TXA-$$TXB
MBX.Ready
END.
```

Remarks

-

Limitations:

25 Variables is the maximum number of possible parameters.

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.9 STJ. - STack Job

MiniRobotLanguage (MRL)

STJ. Command

STack Job

Intention

This Command is the universal Command to manage the User-Stack/Que. 104

- You can exchange stack items (SWAP).
- You can duplicate Items (DUPL).
- You can delete items (REMO)
- You can insert items anywhere in the stack (INSE)

Usage is like this:

```
' This command will swap Stack positions 0 and 2
STJ.SWAP|0|2
```

Now how does the Stack look before and after?
We assume that there are 10 Elements on the Stack, we'll just look at the first 4 Stack positions. This is the stack before the command.

Note: We will use this Stack as the start-stack for all of the following Stack-commands examples.

```
-------------------------------------
User-Stack-Dump:
-------------------------------------

Stackpos: 000 ->String        :Stackplace 10
Stackpos: 001 ->String        :Stackplace 9
Stackpos: 002 ->String        :Stackplace 8
Stackpos: 003 ->String        :Stackplace 7
```

picture 1 - the starting stack before the stack-commands

Now we will swap elements 0 and 2 using the command:

```
STJ.SWAP|0|2
```

And then the stack looks like this:

© 2023 Theo Gottwald * https://www.smart-package.com

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->String        :Stackplace 8
Stackpos: 001 ->String        :Stackplace 9
Stackpos: 002 ->String        :Stackplace 10
Stackpos: 003 ->String        :Stackplace 7
```

You can see that the elements are exchanged.
Using the SWAP Command, you can exchange ANY of the Stack-Elements.

Also you can exchange multiple elements in just one command.
The parameters are always used in pairs. If one parameter is missing, the TOS is exchanged with it.

For example:

```
STJ.SWAP|1|2|3|4
```

Will result in this:

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->String        :Stackplace 10
Stackpos: 001 ->String        :Stackplace 8
Stackpos: 002 ->String        :Stackplace 9
Stackpos: 003 ->String        :Stackplace 6
Stackpos: 004 ->String        :Stackplace 7
```

Now lets add just one parameter to exchange it with the TOS.

```
STJ.SWAP|1|2|3|4|4
```

This what we get:

```
-----------------------------------
User-Stack-Dump:
-----------------------------------

Stackpos: 000 ->String        :Stackplace 7
Stackpos: 001 ->String        :Stackplace 8
Stackpos: 002 ->String        :Stackplace 9
Stackpos: 003 ->String        :Stackplace 6
Stackpos: 004 ->String        :Stackplace 10
```

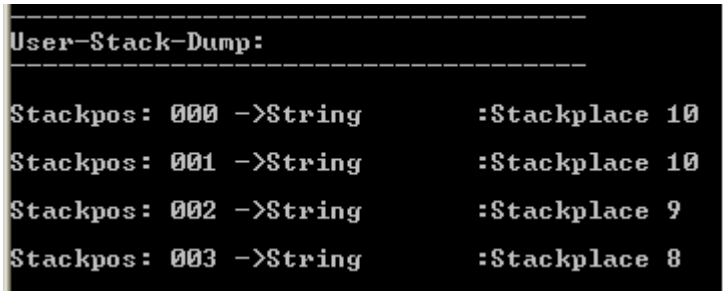
You can exchange up to 20 Stack-elements in the same command!
All exchanges are done from left to right, one after the other.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Now lets **duplicate a Stack element**. The stack before the command is the same like in *picture 1*. In the first example we are going to DUPLicate the TOS (Top of Stack Element).
Because this is the most commonly used case. The command is:

```
STJ.DUPL|0
```

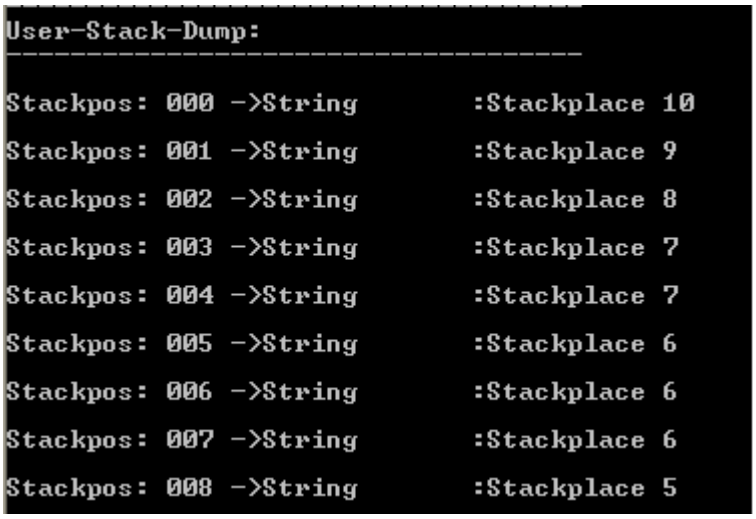
And the result is as expected:



Again you can use up to 20 parameters to duplicate many stack elements with just one command. We will jsut use three duplications here.

```
STJ.DUPLICATE|3|4|4
```

which will result in this:



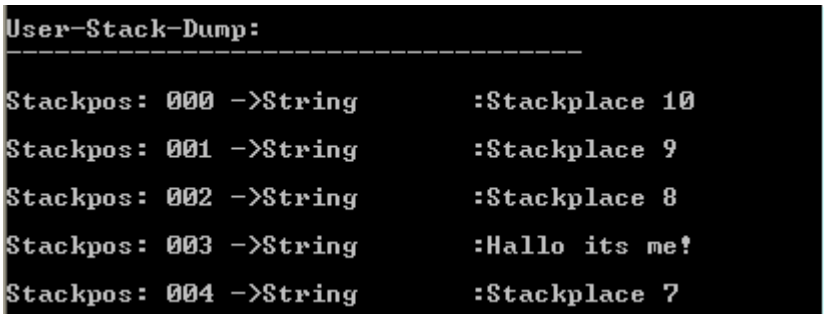
Internally the duplicates are done in a way to avoid shifting effect.

Next let's INSERT something. The stack before the command is the same like in *picture 1*. In the first example we are going to INSErt something on the TOS (Top of Stack).
Because this is the most commonly used case. The command is:

```
STJ.INSERT|3|Hallo its me!
```

The result is as expected:

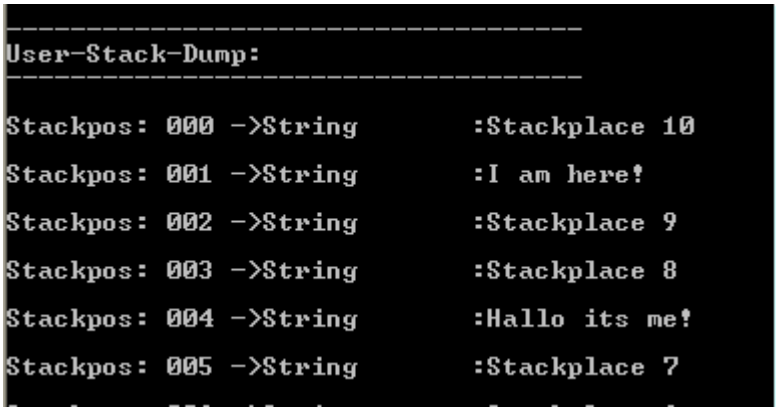
© 2023 Theo Gottwald * https://www.smart-package.com



As always with `STJ.` you can add more parameters. Generally they are used in pairs of two. First the number, then the String. Objects will always be inserted as string elements currently. If you omit the second parameter, or if you leave it empty, an empty string will be inserted. Here is an example with just two elements:

```
STJ.INSERT|3|Hallo its me!|1|I am here!
```

The result is:



You can add more parameters, up to ten elements can be inserted with just one command. INSErts will **not** be processed in strict order from left to right.
They will be processed in a way that "Shifting effects" that will not occur.
An element that you want to have inserted at place 2 will be at place 2 when you dump the stack after the operation. Of course inserting multiple elements at the same place will always shift elements to another place. For example:

```
STJ.INSE|4|Me at place 4||4|Me too at place 4|2|Me at place 2
```

If this would be processed from "Left to right", we would have two inserts at place 4 and then we would insert an element at place 2. This would shift our previous place 4 element to place 5!

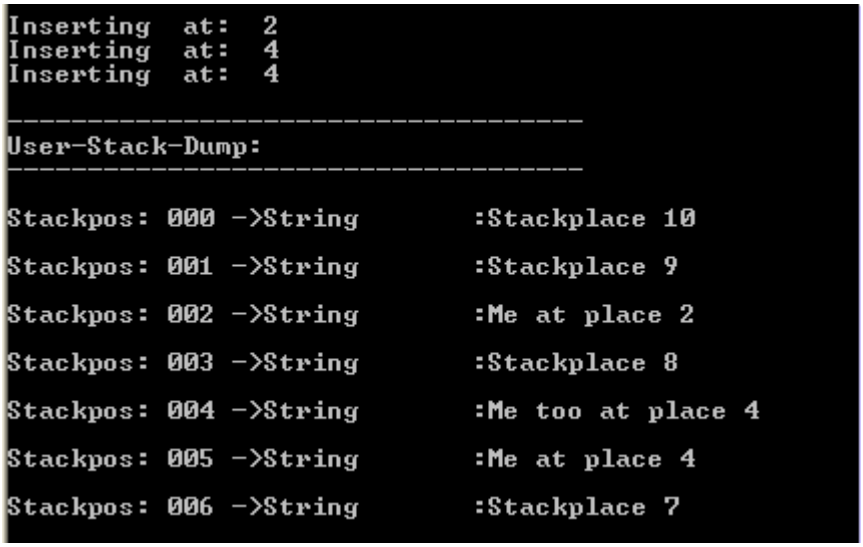
Therefore the INSERT command handles it internally different. The result of this command will be, as if you had done this:

```
STJ.INSE|2|Me at place 2|4|Me at place 4||4|Me too at place 4
```

As you can see below all INSERTS are made in a way that shifting effects are avoided, and that you get what you want.

Below you see the result.

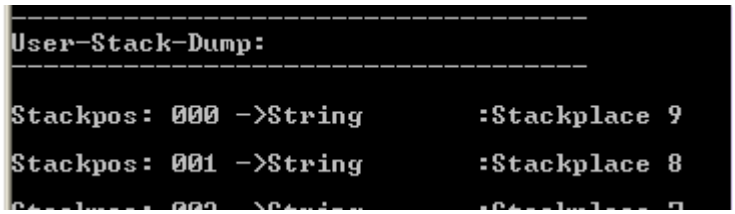
*© 2023 Theo Gottwald * https://www.smart-package.com*



If you got until here, the REMOve command may be already clear to you. Let us anyway take a look at some examples. As always we start with our stack like in *picture 1*. This is the easiest way. We remove just the TOS-element:

```
STJ.REMOVE|0
```

The result is as expected:



Now it gets more interesting. Imagine we would remove the elements just in this order:

```
STJ.REMOVE|0|2
```

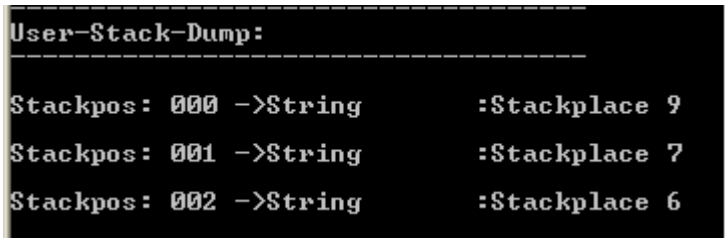
Then internally after the removal of element 0 (TOS), the former element 2 would shift one up and be element 1. If we would go on to remove element 2, we would really remove element 3!

To avoid this, internally the elements are removed from highest to lowest NOT in the order you specify them in the command This avoids confusion through shifting effect.

Important: When the actual stack is switched to a Que, then the internal removal order is automatically reversed. This is true for the REMOVE as well as for the DUPLICATE command.

And here is the result as expected:

 *© 2023 Theo Gottwald * https://www.smart-package.com*



This way, its much easier for you to specify the right elements to be removed. You do not need to care for the right order of removal to avoid shifting effects.

Therefore if you write:

```
STJ.REMOVE|0|2|4
```

exactly the same will happen, as if you write:

```
STJ.REMOVE|4|2|0
```

because the elements are internally sorted, and removed in the right order that avoids "shifting effects". Still you can do something like this:

```
 STJ.REMOVE|0|0|0
```

Which will just delete the first three elements from the Stack.

Important: it does not matter, if you write:

```
STJ.DUPL|0|1
```

or if you write

```
STJ.DUPLICATE|0|1
```

Therefore you can decide if you want to save a bit typing or if you prefer readability.

Syntax

STJ. P1|P2[..|PX]

Parameter Explanation

P1 **-** Stack-Command:

```
SWAP - Exchange given Elements
DUPL - Duplicate given Element(s)
REMO - Remove given Element(s)
INSE - Insert String/Variable into Stack-Position
```

P2 **-** VAR/NUM - depend on used Stack-Command
PX **-** VAR/NUM depend on used Stack-Command

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
STS.GLOB
' Put 10 Items on global Stack
FOR.$$A01|1|10
  VAR.$$B00=Stackplace $$A01
  PUV.$$B00
NEX.
STS.DUMP
DBM.2
: $$B00=3
: $$B01=2
: $$B02=1
' Swap Items 3 with 2
STJ.SWAP|$$B00|$$B01
STS.DUMP
' Insert a 2 (=$$B01) below TOS and insert "Hallo" at Stack-
Pos. 3
STJ.INSE|0|$$B01|$$B00|Hallo
STS.DUMP
' Remove Items 9,8 and 4 from Stack
STJ.REMO|9|8|4
STS.DUMP
' Duplicate Items 2 and 3
STJ.DUPL|$$B01|$$B00
STS.DUMP
DMP.
MBX.Ready
END.
```

Remarks

If you add multiple Parameters, they will not be processed strictly from left to right. Instead they will internally be reordered to avoid "Shifting effects".

Limitations:

-

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019

© 2023 Theo Gottwald * https://www.smart-package.com

- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 8010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.10 STS. - Set STack Settings

MiniRobotLanguage (MRL)

STS. Command

Set sTack Settings

Intention

This Command is the universal Command to manage the User-Stack/Que.

Syntax

STS.P1[|P2]...[|Px]

Parameter Explanation

P1 **-** TXT or VAR
P2 - Px **-** TXT or VAR

```
The following Stack-Commands can be given as Parameter and
will be executed one after the other:

LOCAL - Switches all Commands to "LOCAL STACK"
GLOBAL - Switches all Commands to "GLOBAL STACK"
LIFO - Switches the actual Stack to LIFO*1 (Stack)
FIFO - Switches the actual Stack to FIFO*2 (Que)
SETNODEL - Switches the actual Stack to "No Delete"*3
SETDEL - Switches the actual Stack to "Delete"*4
STANDARD - Resets the Stack-Engine to Standard-Settings.*5
CLEAR - Clear the actual Stack, delete all Elements.
RESET - Clear Stack and Reset Settings
DUMP - Dump actual Stack into Console-Window.
SWAP - Exchange TOS and Stack Element below TOS.
DUPLICATE - Duplicate "Top-of-Stack"-Element.
REMOVE - Remove Element at TOS
```

You can write the first 4 letters of the Parameter or use the full word for better readability of the script.

Example

```
DBM.2
VAR.$$B00=Stackplace 2
' You can write the full words, but ...
STS.GLOBAL
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
VAR.$$B01=This will be duplicated
PUV.$$B00|$$B01
' The first 4 letters are enough
STS.DUMP,DUPL,DUMP,REMO,DUMP,DUPL,DUMP,REMO,REMO,DUMP
DMP.
MBX.Ready
ENR.
```

Remarks

*1 "LAST IN FIRST OUT"
*2 "FIRST IN FIRST OUT"
*3 Elements are NOT deleted when taken from Stack.
*4 Elements are deleted when taken from Stack. (Standard)
*5 Standard-Settings: LIFO, DELETE, MAX-Size=25 Elements.

Limitations:

There are no built-in Limitations to the number of Stack-Commands in a line.
Use the STV.-command to change the maximum number of stack elements.

See also:

- The global and local Stack 366
- STS. - Set STack Settings 3038
- STJ. - STack Job 3030
- STV. - Stack To/from Variable 3040
- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.78.11 STV. - Stack To/from Variable

MiniRobotLanguage (MRL)

STV. Command

STack from/to Variable

Intention

`STV.` saves the whole actual stack into a variable or restores the whole stack from a variable. You can also use this command to change the maximum number of stack elements.
This will not only get the Top-Element from the stack, but will save the complete stack into the variable in a binary format.
This will only be useful to restore the stack later from the variable.

Syntax

STV. P1|P2

Parameter Explanation

P1
- TXT or VAR

P2
- VAR

```
The following Stack-Commands can be given as Parameter and
will be executed one after the other:

TOVAR - Transfers the complete Stack into a variable.
TOVCLEAR - like TOVA, but also clears the stack.
FROM - restores a stack from Variable VAR.
SWAP - Exchange given Element and TOS
DUPLICATE - Duplicate given Element
REMOVE - Remove given Element
SIZE - change the maximum stack size (limited only by Memory)
```

You can write the first four letters or use the full word.

Example

```
DBM.2
: $$A00=1
VAR.$$B00=Stackplace 2
JMP. over
STS.GLOB
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
PUV.$$B00
STS.DUMP
' Following command copies the Global stack into $$AAA
STV.TOVA|$$AAA
STS.DUMP|CLEA|DUMP
STV.FROM|$$AAA
STS.DUMP
:over
STS.LOCA|DUMP
PUV.$$B00
STS.DUMP
' Following command copies the Local stack into $$AAA
STV.TOVA|$$AAA
STS.DUMP
STS.CLEA|DUMP
STV.FROM|$$AAA
STS.DUMP
DMP.
END.
```

Remarks

When using FROM, please note that stack-content is been completely replaced by the content in VAR When using TOVC the stack content is been cleared while there is a copy in VAR.

Limitations:

See also:

- The global and local Stack 366
- STJ. - STack Job 3030
- STS. - Set STack Settings 3038
- PUS. - Push Parameter onto Stack/Que 3013
- PUV. PUsh variables on the User-Stack 3019
- POP. - POP Variable from Stack 3007
- POV. - POp Variables from User-Stack 3010

© 2023 Theo Gottwald * https://www.smart-package.com

3.79 String commands

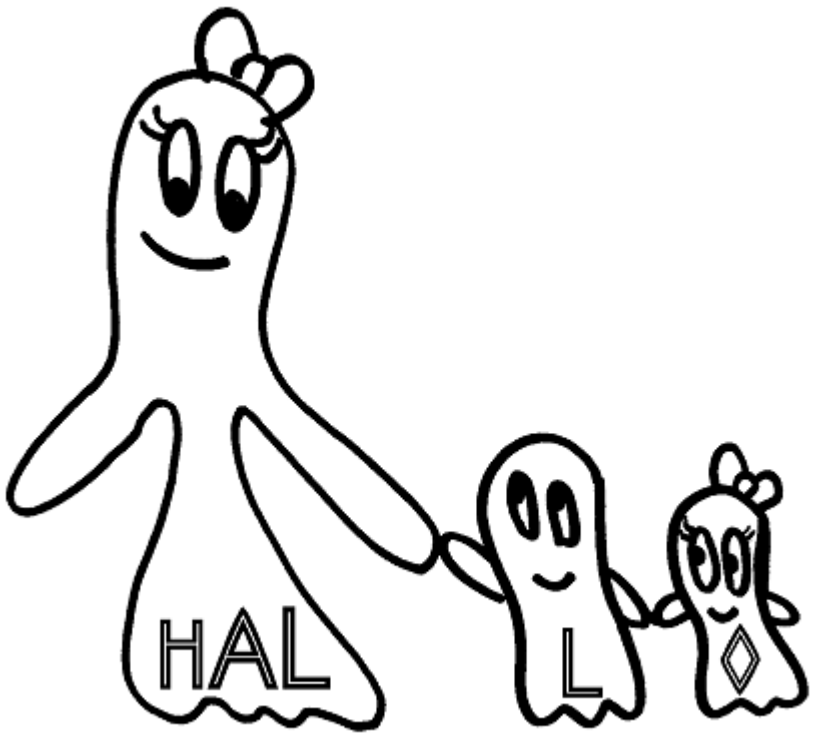
3.79.1 ! Smarty's String Operations

Smarty's String Operations Previous 3040 Top 41 Next 3043

Smarty's String Operations

BASIC was popular for handling its Memory in "Strings".
Strings can be used for text-data, but they can also be used for any type of binary data.

There is no need for you to reserve or free memory. The Robot will do it under the hood.



*© 2023 Theo Gottwald * https://www.smart-package.com*

3.79.2 ACH. - After Character

MiniRobotLanguage (MRL)

ACH. Command

After Character

Intention

Split a file-path or string (text) at any character. Return the Part after the specified character. You can specify a character/string and optional an index number.
The Index Number is used to count all occurrences of that character and select the specified character. If you have a string:

```
http://www.fa2.de/index.html
```

You can do several things using `ACH.` For example:

```
ACH.$$TXT|.
' Result on TOS will be: "de/index.html"
```

You can specify more parameters and do this:

```
ACH.$$TXT|/|3
' Result on TOS will be: "index.html"
```

You can use it to get the extension of a filename:

```
$$FIL=C:\Folder\MyFile.txt
ACH.$$FIL|.||$$RET
DBP.$$RET
```

Syntax

ACH.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Variable with string to split

P2 - Character or string to split variable at.
Can not be the "|" (Pipe Symbol).

P3 - ***(optional)*** Index number of the element that defines the splitting point. If the number is negative (-1,-5 etc.) then

© 2023 Theo Gottwald * https://www.smart-package.com

the elements are counted from right to left (in reverse order).

P4 - (optional) Variable for Result, if omitted TOS is used.

Example

```
'*********************************************
' Sample: ACH
'*********************************************
'
$$FIL=C:\Folder\MyFile.txt
ACH.$$FIL|.||$$RET
DBP.$$RET

 $$TES=Hallo.txt
ACH.$$TES|.||$$RES
DBP.$$RES

$$TES=Hallo.txt
BCH.$$TES|.||$$RES
DBP.$$RES
```

Remarks

-

Limitations:

-

See also:

- **BCH. - Before Character** 3049
- **ALB. - After last Backslah** 3045
- BLB - Before last Backslash 3051
- GFT. - Get-File-Token 1037
- GSS. - GetSplitString 3137
- VRT. - Variable-Right-Trim 3231
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.3 ALB. - After last Backslah

MiniRobotLanguage (MRL)

ALB. Command

After Last Backslash

Intention

Split a file-path or string (text) at the slash or backslash. Return the Part after the last backslash or slash in the string. The effect is very simple. If you have a string:

```
http://www.fa2.de/index.html
```

ALB. will return all after the last slash/backslash, in this case we get

```
index.html
```

As it supports the backslash "\" as well as the slash "/", it can be used on windows-pathes as well as on web-URLs.

Syntax

ALB.[P1][|P2]

Parameter Explanation

P1 - Path or hyperlink or string as text or as variable

P2 - Variable for Result. If omitted, Result is been put on TOS.

Example

```
'*********************************************
' Sample: ALB
' AWT at 2011-07-07
'*********************************************
'
' fill variable $$MYP
VAR.$$MYP=?pfad\AWT.dll
'
' use ALB to get the filename into $$SOL
ALB.$$MYP|$$SOL
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
PRT.Variable MYP is: §§MYP
PRT.Variable SOL is: §§SOL

' Another example
VAR.§§FIL=C:\MyDir\Mydir2
' In this case the result will be placed on the TOS
ALB.§§FIL

' Use VRT. if you have pathes with a "\" at the end
VAR.§§FIL=C:\MyDir\Mydir2\
' In this case the result will be placed on the TOS
VRT.§§FIL|\
ALB.§§FIL

DMP.6

MBX.Ready
'-------------------------------------------------------------
```

Remarks

-

Limitations:

-

See also:

- **ACH. - After Character** 3043
- **BCH. - Before Character** 3049
- BLB - Before last Backslash 3051
- GFT. - Get-File-Token 1037
- GSS. - GetSplitString 3137
- VRT. - Variable-Right-Trim 3231
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.4 ASC. - Get ASCII - Code

MiniRobotLanguage (MRL)

ASC. Command

Get ASCII Code

Intention

This command puts the ASCII-Code for the character or chain of characters in a variable.

Syntax

ASC.[P1] [|P2] [|P3]

Parameter Explanation

P1 - Input-Parameter, if omitted TOS is taken

P2 - Output-variable, if omitted TOS is takeen

P3 - optional Index (Increment)

Example

```
'**********************************
' ASC.-DEMO
'**********************************
'DBM.2
PRT.-------------------------
' We work from Stack to Stack if no
' Parameters are specified.
: §§TXT=NBC
PUV.§§TXT
ASC.
PRT.We get the ASCII-Code for "N" thats >§§000<
PRT.-------------------------
' We work from Variable to Stack if one
' Parameters is specified.
: §§TXT=MUM
ASC.§§TXT
PRT.We get the ASCII-Code for M >§§000<
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.-------------------------
' We work from Variable to Variable if
' two Parameters are specified.
: §§TXT=CMD
ASC.§§TXT|§§LFN
PRT.We get the ASCII-Code for C >§§LFN<
PRT.-------------------------
' We work from Variable to Variable
' and use thirdparameter as additional index.
: §§TXT=abcdefg
: §§IND=3
ASC.§§TXT|§§LFN|§§IND
PRT.We get the ASCII-Code for "c" >§§LFN<
PRT.-------------------------
MBX.Ready
END.
```

Remarks

-

Limitations:

-

See also:

- CHR. - Get CHaRacter 3062
- GSS. - GetSplitString 3137
- ! STR. - String Command 3150
- ASCI-Codes (WEB-Link)
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.5 BCH. - Before Character

MiniRobotLanguage (MRL)

BCH. Command

Before Character

Intention

Split a file-path or string (text) at any character. Return the Part before the specified character. You can specify a character/string and optional an index number.
The Index Number is used to count all occurrences of that character and select the specified character. If you have a string:

```
http://www.fa2.de/index.html
```

You can do several things using `BCH.` For example:

```
BCH.$$TXT|.
' Result on TOS will be: "http://www.fa2"
```

You can specify more parameters and do this:

```
BCH.$$TXT|/|3
' Result on TOS will be: "http://www.fa2.de"
```

You can use it to get all but the extension of a filename:

```
$$FIL=C:\Folder\MyFile.txt
BCH.$$FIL|.||$$RET
DBP.$$RET
```

Syntax

BCH.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Variable with string to split

P2 - Character or string to split variable at.
Can not be the "|" (Pipe Symbol).

P3 - ***(optional)*** Index number of the element that defines the

*© 2023 Theo Gottwald * https://www.smart-package.com*

splitting point. If the number is negative (-1,-5 etc.) then
the elements are counted from right to left (in reverse order).

P4 - (optional) Variable for Result, if omitted TOS is used.

Example

```
'*********************************************
' Sample: BCH
'*********************************************
'
$$FIL=C:\Folder\MyFile.txt
BCH.$$FIL|.||$$RET
DBP.$$RET

$$TES=Hallo.txt
ACH.$$TES|.||$$RES
DBP.$$RES

$$TES=Hallo.txt
BCH.$$TES|.||$$RES
DBP.$$RES
```

Remarks

-

Limitations:

-

See also:

- ACH. - After Character 3043
- ALB. - After last Backslah 3045
- BLB - Before last Backslash 3051
- GFT. - Get-File-Token 1037
- GSS. - GetSplitString 3137
- VRT. - Variable-Right-Trim 3231
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.6 BLB. - Before last Backslash

MiniRobotLanguage (MRL)

BLB. Command

Before Last Backslash

Intention

Split a file-path or string (text) at the slash or backslash. Return the Part before the last backslash or slash in the string. The effect is very simple. If you have a string:

```
http://www.fa2.de/index.html
```

BLB. will return all thats before the last slash/backslash, in this case we get

```
http://www.fa2.de
```

As it supports the backslash "\" as well as the slash "/", it can be used on windows-pathes as well as on web-URLs.

Syntax

BLB.[P1][|P2]

Parameter Explanation

P1 - Path or hyperlink or String

P2 - Variable for Result. If omitted, Result is been put on TOS.

Example

```
'*********************************************
' Sample: BLB
'
'*********************************************
'
' fill variable $$MYP
VAR.$$MYP=?pfad\AWT.dll
'
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' use ALB to get the path into §§SOL
BLB.§§MYP|§§SOL
PRT.Before BLB: §§MYP
PRT.After BLB: §§SOL

' Another example
VAR.§§FIL=C:\MyDir\Mydir2
' In this case the result will be placed on the TOS
BLB.§§FIL

' Use VRT. if you have pathes with a "\" at the end
VAR.§§FIL=C:\MyDir\Mydir2\
' In this case the result will be placed on the TOS
VRT.§§FIL|\
BLB.§§FIL

DMP.6

MBX.Ready
ENR.
'-------------------------------------------------------------
```

Remarks

-

Limitations:

-

See also:

- **ALB - After last Backslah** 3045
- ACH. - After Character 3043
- BCH. - Before Character 3049
- GFT. - Get-File-Token 1037
- GSS. - GetSplitString 3137
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.7 Change case (upper/ lowercase)

3.79.7.1 VTL. - Variable to Lowercase

MiniRobotLanguage (MRL)

VTL. Command

Variable To Lowercase

Intention

This command can be used to make the string-content of a variable to lowercase.

Example:

```
: §§TXT=HaLlO PeTeR
VTL.§§TXT
DBP.§§TXT
' Result is "hallo peter"
```

In the second operation mode you can give a target variable. This way the original variable will not be overwritten.

```
: §§TXT=HaLlO PeTeR
VTL.§§TXT|$$TAR
DBP.$$TAR
' Result is "hallo peter"
```

Syntax

VTL.[P1][|P2]

Parameter Explanation

P1 **-** ***(optional)*** Source and Target-variable, if omitted, TOS is used.

P2 **-** ***(optional)*** if given then **P1** is source and **P2** is destination

Example

```
'************************************
' VTU./VTL./VTM.- DEMO
'************************************
: §§TXT=Hallo Peter
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
VTU.§§TXT
PRT.§§TXT

VTL.§§TXT
PRT.§§TXT

VTM.§§TXT
PRT.§§TXT

' Using two Parameters
: §§TXT=Hallo Peter

VTU.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

VTL.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

VTM.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VBT. - Variable-Both-sides-Trim** 3225
- **VLT - Variable-Left-Trim** 3228
- **VRT. - Variable-Right-Trim** 3231
- **VTU. - Variable to Uppercase** 3059
- **VRD. - Variable-Remove-Doubles** 3076
- **VTM. - Variable to Mediumcase** 3056
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.7.2 VTM. - Variable to Mixedcase

MiniRobotLanguage (MRL)

VTM. Command

Variable To Mediumcase

Intention

This command can be used to make the string-content of a variable to Medium-Case.
In Mixed-Case the first letter of each word is always a uppercase letter, all successive letters are lowercase letters.

Example:

```
: §§TXT=HaLlO PeTeR
VTM.§§TXT
DBP.§§TXT
' Result is "Hallo Peter"
```

In the second operation mode you can give a target variable. This way the original variable will not be overwritten.

```
: §§TXT=HaLlO PeTeR
VTM.§§TXT|$$TAR
DBP.$$TAR
' Result is "Hallo Peter"
```

Syntax

VTM.[P1][|P2]

Parameter Explanation

P1 **-** ***(optional)*** Source and Target-variable, if omitted, TOS is used.

P2 **-** ***(optional)*** if given then **P1** is source and **P2** is destination

Example

```
'************************************
' VTU./VTL./VTM.- DEMO
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'************************************
: §§TXT=Hallo Peter
VTU.§§TXT
PRT.§§TXT

VTL.§§TXT
PRT.§§TXT

VTM.§§TXT
PRT.§§TXT

' Using two Parameters
: §§TXT=Hallo Peter

VTU.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

VTL.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

VTM.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VBT. - Variable-Both-sides-Trim** 3225
- **VLT - Variable-Left-Trim** 3228
- **VRT. - Variable-Right-Trim** 3231
- **VTU. - Variable to Uppercase** 3059
- **VTL. - Variable to Lowercase** 3053
- **VRD. - Variable-Remove-Doubles** 3076
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

© 2023 Theo Gottwald * https://www.smart-package.com

- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.7.3 VTU. - Variable to Uppercase

VTU. - Variable to Uppercase **Previous** 3056 **Top** 41 **Next** 3067

MiniRobotLanguage (MRL)

VTU. Command

Variable To Uppercase

Intention

This command can be used to make the string-content of a variable to uppercase.

Example:

```
: §§TXT=HaLlO PeTeR
VTU.§§TXT
DBP.§§TXT
' Result is "HALLO PETER"
```

In the second operation mode you can give a target variable. This way the original variable will not be overwritten.

```
: §§TXT=HaLlO PeTeR
VTU.§§TXT|$$TAR
DBP.$$TAR
' Result is "HALLO PETER"
```

Syntax

VTU.[P1][|P2]

Parameter Explanation

P1 - ***(optional)*** Source and Target-variable, if omitted, TOS is used.

P2 - ***(optional)*** if given then **P1** is source and **P2** is destination

Example

```
'************************************
' VTU./VTL./VTM.- DEMO
'************************************
: §§TXT=Hallo Peter
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
VTU.§§TXT
PRT.§§TXT

VTL.§§TXT
PRT.§§TXT

VTM.§§TXT
PRT.§§TXT

' Using two Parameters
: §§TXT=Hallo Peter

VTU.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

VTL.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

VTM.§§TXT|§§ERG
PRT.§§TXT -> §§ERG

MBX.!
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VBT. - Variable-Both-sides-Trim** 3225
- **VLT - Variable-Left-Trim** 3228
- **VRT. - Variable-Right-Trim** 3231
- **VRD. - Variable-Remove-Doubles** 3076
- **VTL. - Variable to Lowercase** 3053
- **VTM. - Variable to Mediumcase** 3056
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.8 CHR. - Get CHaRacter

MiniRobotLanguage (MRL)

CHR. Command

Get Character

Intention

This command can be used to get the String-Character for a one or more specified ASCII-Codes. Usage is easy.

```
CHR.$$INP|$$OUT
```

Syntax

CHR. [P1][|P2[|P3]

Parameter Explanation

If 3 Parameters are used, the command will put a sequence (P1 to P3) into P2.

P1 – Input-Parameter, if omitted TOS is taken
P2 – Output-variable, if omitted TOS is taken
P3 - optional. If given all CHR(P1 to P3) are filled into VAR in P2

Example

```
'**********************************
' CHR.-DEMO
'**********************************
'DBM.2
PRT.-------------------------
' We work from Stack to Stack if no
' Parameters are specified.
VAN.$$NUM=65
PUV.$$NUM
DMP.4
CHR.
PRT.We get the letter A: >$$000<
PRT.-------------------------
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' We work from Variable to Stack if one
' Parameters is specified.
: §§TXT=66
CHR.§§TXT
PRT.We get the letter B: >$$000<
PRT.-------------------------
' We work from Variable to Variable if two
' Parameters are specified.
: §§TXT=67
CHR.§§TXT|§§LFN
PRT.We get the letter C >§§LFN<
PRT.-------------------------
' We work from Variable to Variable and use third
' parameter to generate a sequence of ASC-Chars.
' The Chars from 65 to &8 will be put into §§LFN.
: §§TXT=65
: §§IND=68
CHR.§§TXT|§§LFN|§§IND
PRT.We get the letters from A to D >§§LFN<
PRT.-------------------------
END.
```

Remarks

Parameters are range-checked. Numbers lower then 0 will be set to 0, numbers higher then 255 will be set to 255.

Limitations:

-

See also:

- ASC. - Get ASCII - Code 3047
- GSS. - GetSplitString 3137
- ! STR. - String Command 3150

- ASCI-Codes (WEB-Link)

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.9 CSP. - Clip String Part

MiniRobotLanguage (MRL)

CSP. Command

Clip String Part

Intention

Returns a part of a string/text with Text at letf and(or right side removed.
If you have a text/string and you want to clip some letters from left or right side, think of `CSP.` Usage is easy:

```
VAR.$$TXT=Hello name is Peter, Tel.12345
' Now we are going to Clip some letters from Left side
' We put the result in $$RES
CSP.5|$$TXT|$$RES
DBP.$$RES
' The result is "name is Peter, Tel.12345"


' We will now clip 5 characters from the right side
' and put the result into the same variable as the Source
VAR.$$TXT=Hello name is Peter, Tel.12345
CSP.,5|$$TXT
DBP.$$TXT
' The result is "llo name is Peter, Tel."


' We can also clip text at both sides at the same time
VAR.$$TXT=Hello name is Peter, Tel.12345
CSP.3,8|$$TXT|$$RES
DBP.$$RES
' The result is: "lo name is Peter, T"
```

Syntax

CSP.P1|P2[|P3]

Parameter Explanation

P1 – Clip-Position(s), Format:
(number of characters to clip from left side),(number of characters to clip from right side)

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - Input variable, variable with Source-String, if P3 is missing, this variable will also be
used as Target variable.

P3 - ***(optional)*** Target-Variable, receives the result.

Example

```
'**********************************
' CSP.-Sample
'**********************************
VAR.$$TXT=Hello name is Peter, Tel.12345
' Now we are going to Clip some letters from Left side
' We put the result in $$RES
CSP.5|$$TXT|$$RES
DBP.$$RES

' We will now clip 5 characters from the right side
' and put the result into the same variable as the Source
VAR.$$TXT=Hello name is Peter, Tel.12345
CSP.,5|$$TXT
DBP.$$TXT

' We can also clip text at both sides at the same time
VAR.$$TXT=Hello name is Peter, Tel.12345
CSP.3,8|$$TXT|$$RES
DBP.$$RES
'
ENR.
```

Remarks

Binary compatible variable resolution is used for the Source-Part.

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **GSS. - GetSplitString** 3137
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- ACH. - After Character 3043
- BCH. - Before Character 3049
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.10 Delimited Strings

3.79.10.1 GES. - Get-Extracted-String

MiniRobotLanguage (MRL)

GES. Command

Get Extracted String

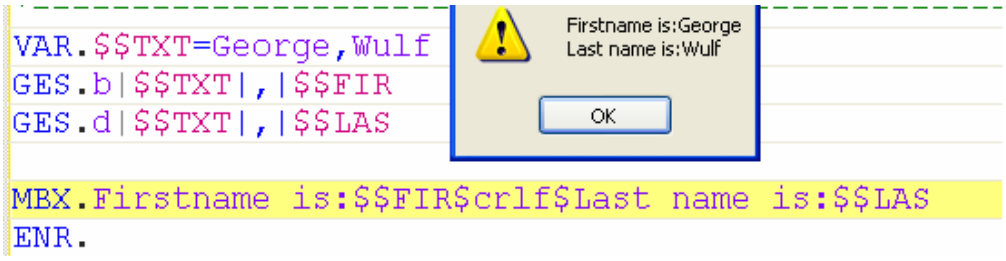
Intention

Simply said: "**Splits a string into two parts**". More exact; you have text or string that consists of basically two parts. These parts are delimited by one or more known delimiter-character(s). This is the situation where you should think of `GES.`.

Returns the portion of a string - leading up to / or following - the first occurrence of a specified character or string. Let us see an easy example. We got a text and we want to have the first and the last name separated.

```
' We got this text
VAR.$$TXT=George,Wulf

' Here we get the first string before the delimiter
GES.b|$$TXT|,|$$FIR

' Here we get the next string after the delimiter
GES.d|$$TXT|,|$$LAS

' This is to display the result
MBX.Firstname is:$$FIR$crlf$Last name is:$$LAS
ENR.
```

And this is the result:



The modes of operation can be separated into two groups:

a,b,c,d - these work on whether a single split-string character, or on a split string of any length is given in **P3.**

e,f,g,h - these treat the split-string **P3** as a group of single characters and test for each of them.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

GES.P1|P2[|P3][|P4]

Parameter Explanation

P1 **-** Prototype

Please use only one single prototype for this command.

a - get numeric result: position of last character before the split-string..

b - return the extracted string which is before the split-character..

c - get numeric result: position of next character after the split-string..

d - return the extracted string which is after the split-character.

e - get numeric result: position of last character before any of the chars in the split-string..

f - return the extracted string which is before any of the chars in the split-string.

g - get numeric result: position of next character after any of the chars in the split-string..

h - return the extracted string which is after any of the chars in the split-string.

Please use only one prototype for this command. The result can be in the original Variable P2 or if P4 is given, it will be placed in P4.

P2 **-** VAR for Source/Destination
P3 **-** VAR with Split-String or Character.
P4 **-** VAR forResult. If ommited Result is been placed in Source-Variable P2.

Example

```
'**********************************
' GES.-Demo
'**********************************
'DBM.2
'JNF.Move-Console.txt
: §§TXT=http://www.space web.com/links.htm
' When using 2 Parameters, the $sp$ is the default Split-Char
' Here we get anything till the first space
PRT. Before:>§§TXT<
GES.b|§§TXT
PRT. After :>§§TXT<
PRT.---------------------------
: §§TXT=http://www.space web.com/links.htm
' Now we us the / Sign as Split-char and after
PRT. Before:>§§TXT<
GES.b|§§TXT|/
PRT. After :>§§TXT<
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.--------------------------
: §§TXT=http://www.space web.com/links.htm
PRT. Before:>§§TXT<
GES.a|§§TXT|/|§§NUM
PRT. After :>§§TXT< -<§§NUM>
PRT.--------------------------
: §§TXT=http://www.space web.com/links.htm
PRT. Before:>§§TXT<
GES.c|§§TXT|$sp$|§§NUM
PRT. After :>§§TXT< -<§§NUM>
PRT.--------------------------
END.
```

Remarks

To split mutiple delimited strings, use SBD. 3070 or SBM. 3073

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.10.2 SBD. - String between Delimiter

MiniRobotLanguage (MRL)

SBD. Command

String Between Delimiter

Intention

Return a delimited part of a string. A example for a delimited string is a "csv" file from Excel. It could look like this:

```
VAR.$$TXT=Name,Last Name,Zip-Code
```

In this case the comma is the delimiter. Using the `SBD.`-command you can get any part of such delimited strings. Usage is like this:

```
VAR.$$TXT=Name,Last Name,Zip-Code
SBD.$$TXT|,|2|$$RES
DBP.$$RES
```

Will return "Last name", as its the second part in the delimited string.

There is another mode of operation. Imagine the delimiter is a PIPE-Sign "|". Now the PIPE-Sign is the delimiter for the command itself, how could we do that?

We can do that with the second operation mode, we use **P5**. When we specify **P5** then we can change the delimiter for the command itself. It's as easy as this:

```
' Here we have a string that is delimited with PIPE's
VAR.$$TXT=Name|Last Name|Zip-Code
' We tell SBD. that we use the comma as command-delimiter (see
P5)
SBD.,$$TXT,|,2,$$RES
DBP.$$RES
```

As alternative, you can use the `RPL.`-command and replace all Pipes with something else before using `SBD.`

In this case only one `SBD.` command is used to extract multiple parts of a delimited string to five variables: `$$V01` to `$$V05`.

```
'#EXE:?path\
'#SPI:ForceWrite

VAS.$$SRC=12*34*45*67*89
' We remove leading and trailing delimiters first
VBT.$$SRC|*

' We replace double delimiters next
$$REP=**
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$WIT=*
RPL.$$SRC|$$REP|$$WIT

' Get number of delimited elements
GSC.$$SRC|*|$$CNT|d

' Split the delimited parts
FOR.$$LOP|1|$$CNT
  SBD.$$SRC|*|$$LOP|$$RES
' Construct output variable
  VAN.$$VON=$$LOP
  FMT.$$VON|00
  VAR.$$VAR=$$V$$VON
' Store into output variable
  VAI.$$VAR=$$RES
NEX.

' Print the 5 Variables
PRT.$$V01
PRT.$$V02
PRT.$$V03
PRT.$$V04
PRT.$$V05
MBX.!
ENR.
```

Syntax

SBD.[P5]P1|P2|P3[|P4]

Parameter Explanation

P1 - STR Source string from which the part should be extracted.

P2 - Delimiter inside the source string.

P3 - Index of delimited string-item to be extracted

P4 - optional Variable for result. If omitted, result goes to TOS.

P5 - optional SINGLE CHAR. Delimiter for SBD itself. Cannot be "$"
or "§". If omitted its "|".

Example

```
' Example for SBD. Result is "das"
: $$TAR=Hallo Peter das ist ein Satz.
: $$IDX=3
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SBD.$$TAR|$SP$|$$IDX|$$TAR
PRT.Result is: $$TAR
DMP.
END.

' Example 2 for SBD. Result is "das"
: $$TAR=Hallo|Peter|das|ist|ein|Satz.
: $$IDX=3

SBD.!$$TAR!|!$$IDX!$$TAR
PRT.Result is: $$TAR
DMP.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
-
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.10.3 SBM. - String between Delimiter-Multiple

MiniRobotLanguage (MRL)

SBM. Command

String Between delimiter Multiple

Intention

Return a delimited part of a string. You can specify multiple parts to be returned in one string. For example:

```
$$VAR=12*3*45*5*67
SBM.$$VAR|*|2|4|$$RES

$$RES will be 3*45*5
```

A example for a delimited string is a "csv" file from Excel. It could look like this:

```
VAR.$$TXT=Name,Last Name,Zip-Code
```

In this case the comma is the delimiter. Using the `SBM.`-command you can get any part of such delimited strings. Usage is like this:

```
VAR.$$TXT=Name,Last Name,Zip-Code
SBM.$$TXT|,|2|3|$$RES
DBP.$$RES
```

Will return "Last name,Zip-Code,", as its the second and the third part in the delimited string. Note that you will find any delimited part in the result aagain delimited with delimiters, even if there were no such delimiters in the source string. This is done to be compatible with empty delimited parts.

There is another mode of operation. Imagine the delimiter is a PIPE-Sign "|". Now the PIPE-Sign is the delimiter for the command itself, how could we do that?

We can do that with the second operation mode, we use **P5**. When we specify **P5** then we can change the delimiter for the command itself. It's as easy as this:

```
' Here we have a string that is delimited with PIPE's
VAR.$$TXT=Name|Last Name|Zip-Code
' We tell SBD. that we use the comma as command-delimiter (see
P5)
SBM.,$$TXT,|,2,$$RES
DBP.$$RES
' Will print: "Last Name|Zip-Code|"
```

As alternative, you can use the `RPL.`-command and replace all Pipes with something else before using `SBM.`

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

SBM.[P6]P1|P2|P3|P4[|P5]

Parameter Explanation

P1 - STR Source string from which the part should be extracted.

P2 - Delimiter inside the source string.

P3 - Start-Index of delimited string-item to be extracted

P4 - End-Index of delimited string-item to be extracted

P5 - Variable for result. If omitted, result goes to TOS.

P6 - optional SINGLE CHAR. Delimiter for SBD itself.
Cannot be "$" or "§". If omitted its "|".

Example

```
'**********************************
' SBM.-Demo
'**********************************
DBM.2
VAR.$$URA=//Local/www.com/it/berater/org
SBM.$$URA|/|3|4|$$ERG
PRT. Ergebnis von SBM: $$ERG
:enx
MBX.!
END.
```

Remarks

If you have a delimited String and you want to spit all the delimited parts -each into an separate variable, best use `SBD.` and copy it several times for each delimited part.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.5.2 Working with Text-Strings 110
- SBD. - String between Delimiter 3070
- FEM. - For Each Member 929
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- SRT. - Sort Items 3147
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- ! STR. - String Command 3150
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.10.4 VRD. - Variable-Remove-Doubles

MiniRobotLanguage (MRL)

VRD. Command

Variable Remove Doubles

Intention

This command can be used to make strings have only single delimiters.

The command has two modes: Text-mode and Binary mode.
In Text mode ASCII 32 and ASC 9 (TAB's and SPACES) are processed to single Spaces in one Run. In text-mode, the string is not allowed to contain ASCII(0).

In Binary Modes all characters are allowed.

Syntax

VRD.[P1][|P2]

Parameter Explanation

P1 - is Source and Target-variable

P2 - is obsolete for Text-Trimming.
  In that case it just trims away all spaces left and right.
  If Parameter 2 is "all" the all characters from CHR$(0 to 32)
  will be trimmed.

Example

```
''************************************
' VRD/VBT/SBD - DEMO
'************************************
VAR.§§LIN=12     345     6789
' We first remove the double delimiters in Textmode
VRD.§§LIN
PRT.>§§LIN<
VAR.§§LIN=1234aa4567bb34567
' We now remove double delimiters in binary Mode
VRD.§§LIN|abc
PRT.>§§LIN<
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.!
END.
```

Remarks

Often used together with `SBD.` or `SBM.` or `FEM.`

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VBT. - Variable-Both-sides-Trim** 3225
- **VLT - Variable-Left-Trim** 3228
- **VRT. - Variable-Right-Trim** 3231
- **VTU. - Variable to Uppercase** 3059
- **VTL. - Variable to Lowercase** 3053
- **VTM. - Variable to Mediumcase** 3056
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11 DIC.-Delimited Item Commands

MiniRobotLanguage (MRL)

DIC. Command

Delimited Item Commands

These commands are for Excel- and CSV-Files.

You can
- remove Columns,
- Insert Columns,
- Swap Columns.
- return only those Columns you want
- return only those Columns you want in the order you want

In real life many information is stored in Excel-Files, and often we have to work with these.

Using `SBD.` it was always possible to use with Excel-CSV-Files.
Now using the new `DIC.`-Commands, many tasks that needed to be programmed before,
can now done with just 1 Line of Code.
.
The `DIC.`-Command works on "Line-by-Line" Base.
Therefore you will normally load the "csv"-File using the `TXT.`-Commands.
And then work on it "Line-by-Line" using the `DIC.`-Commands.

```
DIC.Swap|$$DEL|$$LIN|$$INA|$$INB|$$RES
DIC.Insert|$$DEL|$$LIN|$$IND|$$RES
DIC.remove|$$DEL|$$LIN|$$IND|$$RES
DIC.Get_in_Order|$$DEL|$$LIN|$$IND|$$RES
DIC.Get_Fields|$$DEL|$$LIN|$$IND|$$RES
```

Now here are some Samples for the DIC.-Commands:

```
' Get only the specified fields in the specified Order
VAR.$$DEL=;
VAR.$$LIN=AA;BB;CC;DD;EE;FF
VAR.$$IND="C","E",1,6,4
DIC.Geo|$$DEL|$$LIN|$$IND|$$RES
' returns:  CC;EE;AA;FF;DD
DBP.$$RES
ENR.


' Get only the specified fields in original Order
VAR.$$DEL=;
VAR.$$LIN=AA;BB;CC;DD;EE;FF
VAR.$$IND="A","E",2
DIC.Get|$$DEL|$$LIN|$$IND|$$RES
' returns: AA;BB;EE
DBP.$$RES
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Swap the two Columns $$INA and $$INB
VAR.$$DEL=;
VAR.$$LIN=AA;BB;CC;DD;EE;FF
VAR.$$INA="A"
VAR.$$INB="E"
' returns: EE;BB;CC;DD;AA;FF
DIC.Swap|$$DEL|$$LIN|$$INA|$$INB|$$RES
DBP.$$RES
ENR.


' Insert Columns before the specified places
VAR.$$DEL=;
VAR.$$LIN=AA;BB;CC;DD;EE;FF
VAR.$$IND="A","E"
DIC.Insert|$$DEL|$$LIN|$$IND|$$RES
' returns: ;AA;BB;CC;DD;;EE;FF
DBP.$$RES
ENR.


' remove specified Columns
VAR.$$DEL=;
VAR.$$LIN=AA;BB;CC;DD;EE;FF
VAR.$$IND="A","E",3
DIC.remove|$$DEL|$$LIN|$$IND|$$RES
' returns: BB;DD;FF
DBP.$$RES
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.1 DIC.Eval_Column_List

MiniRobotLanguage (MRL)

DIC.Eval_Column_List

Convert and sort Description of delimited Fields.

Intention

Assume you have several X-Coordinate from A Spreadsheet like this: `"AA" but also like "A-C" or "3-7".`
Now you want to convert it to its numeric representation which in this case would be 27.
This function will NOT remove fields that are multiple times specified.

```
VAR.$$LIN="1","2-5","Y-AB","J-L",21,"AC-AF","JK-JZ"

' Now we call the command
DIC.ecl|$$LIN|$$RES

' And Debug-Print the Result 27
DBP.$$RES

ENR.

' The Result
'  1,2,3,4,5,10,11,12,21,25,26,27,28,29,30,31,32,271,272,273,274,275,276,277,278
```

Syntax

DIC.ecl|P1[|P2]
DIC.Eval_Column_List|P1[|P2]

Parameter Explanation

P1 – Field Index can be a number like 1,2,3 or a Letter like A, AB. Letters must not be enclosed in " ".
You can also specify the "-" Sign to specify a Range. Like: "A-C". This will be interpreted as "A","B","C".
P2 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'*********************************
'
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

This Command does somehow the same like DIC.Get_Field_Index. But unlike this command, DIC.L2N will not accept Fields in " ".

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.2 DIC.Format_Numeric

MiniRobotLanguage (MRL)

DIC.Format_Numeric

Format delimited Fields in a Line in the same way like with FMT.

Intention

Assume you a line containing some Floating-Point Numbers. Now if you convert them to TAB-separated Text, they may be badly formatted if they do not have an equal length.
Use this Command to format specified Numeric Data in a line.

```
VAR.$$LIN=2022/07/19$tab$03:44:06$tab$-32.4$tab$-12.0$tab$-12.0$tab$-17.7$tab$36
VAR.$$DEL=$tab$
VAR.$$IND=4,5,6,7
' For more Options see FMT.-Command
VAR.$$FMT=00.00

' Now we call the command
DIC.Format_Numeric|$$LIN|$$DEL|$$IND|$$FMT|$$RES

' And Debug-Print the Result 27
DBP.$$LIN$crlf$$$RES

ENR.
```

Result:

```
[22:08:18] 2022/07/19        03:44:06 -32.4     -12.0     -12.0     -17.7     36.7      377
2022/07/19        03:44:06 -32.4     -12.00    -12.00    -17.70    36.70     377
```

Syntax

DIC.fnu|P1|P2|P3|P4[|P5][|P6]
DIC.Format_Numeric|P1|P2|P3|P4[|P5][|P6]

Parameter Explanation

P1 – Field Index can be a number like 1,2,3 (makes no sense anyway) or a Letter like "A", "AB". Letters must be enclosed in " ".
P2 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'************************************
'
'************************************
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.3 DIC.Get_Field_Index

MiniRobotLanguage (MRL)

DIC.Get_Field_Index

Convert Description of delimited Field to numbers.

Intention

Assume you have a XCoordinate from A Spreadsheet like this: "AA".
Now you want to convert it to its numeric representation which in this case would be 27.

```
VAR.$$LIN=AA

' Now we call the command
DIC.gfi|$$LIN|$$RES

' And Debug-Print the Result 27
DBP.$$RES

ENR.
```

Syntax

DIC.gfi|P1[|P2]
DIC.Get_Field_Index|P1[|P2]

Parameter Explanation

P1 – Field Index can be a number like 1,2,3 (makes no sense anyway) or a Letter like "A", "AB". Letters must be enclosed in " ".
P2 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'************************************
'
'************************************
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.4 DIC.Get_Fields

MiniRobotLanguage (MRL)

DIC.Get_Fields

Get specified Fields from a Delimited String in the Original Order

Intention

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-bs-line out of a csv.-file

Now, you decide that you only need the Columns 1,2, and 5.
Note that with this command, you can specify the Columns in any order, the result will be the same.

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' This is the Index-List with the Elements that we want to get in the result
VAR.$$IND="A","E",2

' Now we call the command
DIC.Get|$$DEL|$$LIN|$$IND|$$RES

' And Debug-Print the Result
DBP.$$RES

' Result:
' BB;DD;FF
ENR
```

And this is the result: **`BB;DD;FF`**

Syntax

DIC.Get_Fields|P1|P2|P3[|P4]

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P2**.
P2 – Line with delimited Items.

© 2023 Theo Gottwald * https://www.smart-package.com

P3 – Variable with Index of Fields to get. Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.
P4 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'**********************************
' DIC.Get - Sample
'**********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' This is the Index-List with the Elements that we want to get in the result
  VAR.$$IND="A","E",2

  ' Now we call the command
  DIC.Get|$$DEL|$$LIN|$$IND|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  ' BB;DD;FF
  ENR
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.5 DIC.Get_in_Order

MiniRobotLanguage (MRL)

DIC.Get_in_Order

Get specified Fields from a Delimited String in the Order of the List-Parameter

Intention

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you only need the Columns 3,5,1,6 and 4 in this order.
Note that with this command, you can specify the Columns in the order that will be in the result.

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' This is the Index-List with the Elements that we want to get in the result
VAR.$$IND="C","E",1,6,4

' Now we call the command
DIC.Geo|$$DEL|$$LIN|$$IND|$$RES

' And Debug-Print the Result
DBP.$$RES
ENR
```

And this is the result: **`CC;EE;AA;FF;DD`**

Syntax

DIC.Get_in_Order|P1|P2|P3[|P4]

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P2**.
P2 – Line with delimited Items.
P3 – Variable with Index of Fields to get. Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.

© 2023 Theo Gottwald * https://www.smart-package.com

P4 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'**********************************
' DIC.Geo - Sample
'**********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' This is the Index-List with the Elements that we want to get in the result
  VAR.$$IND="C","E",1,6,4

  ' Now we call the command
  DIC.Geo|$$DEL|$$LIN|$$IND|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  ' CC;EE;AA;FF;DD
  ENR
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.6 DIC.Insert

MiniRobotLanguage (MRL)

DIC.Insert

Insert Fields before specified places into a Delimited String

Intention

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you want to insert Columns before "A" and before "E".

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' This is the Index-List with the Elements where we want to insert new fiels bef
VAR.$$IND="A","E"

' Now we call the command
DIC.Insert|$$DEL|$$LIN|$$IND|$$RES

' And Debug-Print the Result
DBP.$$RES
ENR
```

And this is the result: **`;AA;BB;CC;DD;;EE;FF`**

Syntax

DIC.Insert|P1|P2|P3[|P4]

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P2**.
P2 – Line with delimited Items.
P3 – Variable with Index of Fields to Insert Fields before. Technically the command will just place a delimiter in front of these fields.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.

© 2023 Theo Gottwald * https://www.smart-package.com

P4 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'**********************************
' DIC.Insert - Sample
'**********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' This is the Index-List with the Elements where we want to insert new fiels bef
  VAR.$$IND="A","E"

  ' Now we call the command
  DIC.Insert|$$DEL|$$LIN|$$IND|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  '  ;AA;BB;CC;DD;;EE;FF
  ENR
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.7 DIC.Letter_to_Number

MiniRobotLanguage (MRL)

DIC.Letter_to_Number

Convert Description of delimited Field to numbers.

Intention

Assume you have a XCoordinate from A Spreadsheet like this: "AA".
Now you want to convert it to its numeric representation which in this case would be 27.

```
VAR.$$LIN=AA

' Now we call the command
DIC.l2n|$$LIN|$$RES

' And Debug-Print the Result 27
DBP.$$RES

ENR.
```

Syntax

DIC.l2n|P1[|P2]
DIC.Letter_to_Number|P1[|P2]

Parameter Explanation

P1 – Field Index can be a number like 1,2,3 (makes no sense anyway) or a Letter like A, AB. Letters must not be enclosed in " ".
P2 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'*************************************
'
'*************************************
```

Remarks

This Command does somehow the same like DIC.Get_Field_Index. But unlike this command, DIC.L2N will not accept Fields in " ".

© 2023 Theo Gottwald * https://www.smart-package.com

This Command does not need Excel to be installed on the System.

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.8 DIC.Number_to_Letter

MiniRobotLanguage (MRL)

DIC.Number_to_Letter

Convert Description of delimited Field to numbers.

Intention

Assume you have a XCoordinate from A Spreadsheet like this: "AA".
Now you want to convert it to its numeric representation which in this case would be 27.

```
VAR.$$LIN=285

' Now we call the command
DIC.l2n|$$LIN|$$RES

' And Debug-Print the Result JY
DBP.$$RES

ENR.
```

Syntax

DIC.n2l|P1[|P2]
DIC.Number_to_Letter|P1[|P2]

Parameter Explanation

P1 – Field Index can be a number like 1,2,33
P2 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'*********************************
'
'*********************************
```

Remarks

This Command does not need Excel to be installed on the System..

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.9 DIC.Remove

MiniRobotLanguage (MRL)

DIC.Remove

Remove specified Fields from a Delimited String

Intention

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you want to remove Columns 1,3, and 5.

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' This is the Index-List with the Elements where we want to insert new fiels bef
VAR.$$IND="A","E",3

' Now we call the command
DIC.remove|$$DEL|$$LIN|$$IND|$$RES

' And Debug-Print the Result
DBP.$$RES
ENR
```

And this is the result: **`BB;DD;FF`**

Syntax

DIC.Remove|P1|P2|P3[|P4]

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`.
This is the specified delimiter that delimits the fields in the specified Line **P2**.
P2 – Line with delimited Items.
P3 – Variable with Index of Fields to remove.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.
P4 – (optional) Variable for Result. If omitted result is placed on TOS.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
' DIC.Remove - Sample
'**********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' This is the Index-List with the Elements where we want to insert new fiels bef
  VAR.$$IND="A","E",3

  ' Now we call the command
  DIC.Remove|$$DEL|$$LIN|$$IND|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  '  BB;DD;FF

  ENR
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.1(DIC.Set_Multiple

MiniRobotLanguage (MRL)

DIC.Set_Multiple

Set specified Fields from a Delimited String in the Order of the List-Parameter to other specified values

Intention

Assume you have a TAB-delimited Line like this:

```
2022/07/19 03:44:06    -32.4 -12.00      -12.00      -17.70
     36.70 377
```

Now, you decide that you want to change the Columns
`4,6,5,7` in this order to this values:
`-45,+56.00,.15.00,5`

Note that with this command, you can specify the Columns in the order that you specify the values in **P4**.

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items and will be changed
VAR.$$LIN=2022/07/19$tab$03:44:06$tab$-32.4$tab$-12.00$tab$-12.00$tab$-17.70$tab

' This is the Index-List with the Elements that we want to set the values for.
' The Order is not iumportant, just that the values in P5 will be matched 1:1 to
VAR.$$IND=4,6,5,7
' These are the values that will be set to these fields
VAR.$$VAL=-45,+56.00,.15.00,5

' Now we call the command
DIC.sem|$$LIN|$$DEL|$$IND|$$VAL

' And Debug-Print the Result
DBP.$$LIN
ENR.
```

And this is the result:

```
2022/07/19   03:44:06      -32.4   -45    .15.00  +56.00  5      377
```

Before: `2022/07/19 03:44:06 -32.4 -12.00 -12.00 -17`

After: `2022/07/19 03:44:06 -32.4 -45 .15.00 +56.00 5 377`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

DIC.Set_Multiple|P1|P2|P3|P4[|P5]

Parameter Explanation

P1 – Line with delimited Items.
P2 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P1**.
P3 – Variable with Index of Fields to be changed. Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to change.
IMPORTANT: The number of Elements in **P3** must 1:1 match the number of Elements in **P4** else the Command will run into an error and return an empty result.
P4 – Variable with values to set into the fields that are specified by P3.
P5 – (optional) Variable for Result. If omitted result is placed in **P1** overwriting the original line..

Example

```
'***********************************
' DIC.Sem - Sample
'***********************************
  VAR.$$LIN=2022/07/19$tab$03:44:06$tab$-32.4$tab$-12.0$tab$-12.0$tab$-17.7$tab$36
  VAR.$$DEL=$tab$
  VAR.$$IND=4,6,5,7
  VAR.$$VAL=-45,+56.00,.15.00,5
  ' For more Options see FMT.-Command
  VAR.$$FMT=00.00

  ' Now we call the command
  DIC.Format_Numeric|$$LIN|$$DEL|$$IND|$$FMT|$$RES
  $$RET=$$RES
  DIC.sem|$$RET|$$DEL|$$IND|$$VAL

  VAR.$$REF=$$RET
  VAR.$$VAL=XI
  VAR.$$INX=5
  DIC.set|$$REF|$$DEL|$$INX|$$VAL

  ' And Debug-Print the Result 27
  DBP.$crlf$$$LIN$crlf$$$RES$crlf$$$RET$crlf$$$REF

  ' Output:
  ' 2022/07/19   03:44:06    -32.4 -12.0       -12.0       -17.7       36.7  377
  ' 2022/07/19   03:44:06    -32.4 -12.00      -12.00      -17.70      36.70 377
  ' 2022/07/19   03:44:06    -32.4 -45         .15.00      +56.00      5     377
  ' 2022/07/19   03:44:06    -32.4 -45         XI          +56.00      5     377
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
'============================================================
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.1 DIC.Set_Value

MiniRobotLanguage (MRL)

DIC.Set_Value

Set a Value of a specified Field from a Delimited String to another specified value

Intention

Assume you have a TAB-delimited Line like this:

```
2022/07/19 03:44:06    -32.4 -45    .15.00      +56.00      5
     377
```

Now, you decide that you want to change the Column `5` to this value: `XI`

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items and will be changed
VAR.2022/07/19 03:44:06$tab$-32.4$tab$-45$tab$.15.00$tab$+56.00$tab$5$tab$377

' This is the Index of the Elements that we want to set the value for.
VAR.$$INX=5
' This is the values that will be set into the field
VAR.$$VAL=XI

' Now we call the command
DIC.set|$$REF|$$DEL|$$INX|$$VAL

' And Debug-Print the Result
DBP.$$LIN
ENR.
```

And this is the result:

```
2022/07/19   03:44:06      -32.4   -45     XI      +56.00  5       377
```

Before:

```
2022/07/19  03:44:06 -32.4    -45  .15.00   +56.00
```

After:

```
2022/07/19 03:44:06 -32.4    -45  XI
+56.00   5    377
```

Syntax

DIC.Set_Value|P1|P2|P3|P4[|P5]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 – Line with delimited Items.
P2 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P1**.
P3 – Variable with Index of Field to be changed. Can be Single number like 1 or can be an Letter like "A".
P4 – Variable with values to set into the field that is specified by **P3**.
P5 – (optional) Variable for Result. If omitted result is placed in **P1** overwriting the original line..

Example

```
'**********************************
' DIC.Set - Sample
'**********************************
  VAR.$$LIN=2022/07/19$tab$03:44:06$tab$-32.4$tab$-12.0$tab$-12.0$tab$-17.7$tab$36
  VAR.$$DEL=$tab$
  VAR.$$IND=4,6,5,7
  VAR.$$VAL=-45,+56.00,.15.00,5
  ' For more Options see FMT.-Command
  VAR.$$FMT=00.00

  ' Now we call the command
  DIC.Format_Numeric|$$LIN|$$DEL|$$IND|$$FMT|$$RES
  $$RET=$$RES
  DIC.sem|$$RET|$$DEL|$$IND|$$VAL

  VAR.$$REF=$$RET
  VAR.$$VAL=XI
  VAR.$$INX=5
  DIC.set|$$REF|$$DEL|$$INX|$$VAL

  ' And Debug-Print the Result 27
  DBP.$crlf$$$LIN$crlf$$$RES$crlf$$$RET$crlf$$$REF

  ' Output:
  ' 2022/07/19   03:44:06    -32.4 -12.0       -12.0       -17.7       36.7  377
  ' 2022/07/19   03:44:06    -32.4 -12.00      -12.00      -17.70      36.70 377
  ' 2022/07/19   03:44:06    -32.4 -45         .15.00      +56.00      5     377
  ' 2022/07/19   03:44:06    -32.4 -45         XI          +56.00      5     377


  ENR.
  '=========================================================
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.11.1 DIC.Swap

MiniRobotLanguage (MRL)

DIC.Swap

Exchange two Fields inside a Delimited String

Intention

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you want to insert Columns before "A" and before "E".

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' Theseare the two fields to be exchanged
VAR.$$INA="A"
VAR.$$INB="E"

' Now we call the command
DIC.Swap|$$DEL|$$LIN|$$INA|$$INB|$$RES

' And Debug-Print the Result
DBP.$$RES
ENR
```

And this is the result: **`EE;BB;CC;DD;AA;FF`**

Syntax

DIC.Swap|P1|P2|P3|P4[|P5]

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P2**.
P2 – Line with delimited Items.

© 2023 Theo Gottwald * https://www.smart-package.com

P3 – Variable with Index of Fields A to Swap with Field in **P4**.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.
P4 – Variable with Index of Fields B to Swap with Field in **P3**.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.
P5 – (optional) Variable for Result. If omitted result is placed on TOS.

Example

```
'**********************************
' DIC.Swap - Sample
'**********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' Theseare the two fields to be exchanged
  VAR.$$INA="A"
  VAR.$$INB="E"

  ' Now we call the command
  DIC.Swap|$$DEL|$$LIN|$$INA|$$INB|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  '  EE;BB;CC;DD;AA;FF
  ENR
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070

© 2023 Theo Gottwald * https://www.smart-package.com

- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.12 Format Strings

3.79.12.1 FMT. - Format Numeric Output

MiniRobotLanguage (MRL)

FMT. Command

Format Numeric Output

Intention

This command can be used to format numeric output according to a format-prototype contained in **P2**. `FMT.` has 3 modes of operation.

In Mode 1, you can just specify a number of digits. Here are some examples:

```
$$NUM=25.22222
FMT.$$NUM|3
DBP.$$NUM
' result is "25.2"

$$NUM=125.22222
FMT.$$NUM|3
DBP.$$NUM
' result is "125"

$$NUM=125.22222
FMT.$$NUM|(4+1)
DBP.$$NUM
' result is "125.22"
```

In Mode 2 you can specify a pattern which will be used for the formatting. It must contain non numeric characters, that are all characters except "`0123456789`", while not being a formula in (..) .

Example:

```
$$NUM=125.22222
FMT.$$NUM|00.000
DBP.$$NUM
' Result is "125.222"

$$NUM=125.22223
FMT.$$NUM|0+00
DBP.$$NUM
' Result is "1+25"

$$NUM=082324
FMT.$$NUM|00\:00\:00
DBP.$$NUM
' Result is "08:23:24"
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

FMT.[P1][|P2]

Parameter Explanation

P1 - Source and Target-variable, must contain a number
or formula in (..).

P2 - There are four modes of operation, depending on the
format of ***P2***.

First: You can specify a pure numeric parameter, then
this will be the number of digits in the output.

Second: You can use one pattern to format the output
according to this pattern.

Third: You can specify two patterns, separated by a
colon.
Then the first one will be used for positive numbers,
the second one will specify the format for negative
numbers.

Fourth: You can specify tthree patterns, each separated
by a colon. Then the first pattern will be used to
format
positive numbers, the second pattern will specify the
format
for negative numbers. And the third pattern will specify
the
format of a zero.

The following table shows the characters you can use to
create the user-defined format strings (patterns) and
what
they are good for.

(empty) - If P2 is omitted, the number is just trimmed
at both sides, similar to VBT.

0 - digit placeholder. FMT. will either insert a digit
or
a 0. If there is a digit in num_expression in the
position
where the 0 appears in the pattern string, FMT. return
that
digit. Otherwise FMT. return a "0".
If the number being formatted has fewer digits than
there are

© 2023 Theo Gottwald * https://www.smart-package.com

```
      zeros (on either side of the decimal point) in the
pattern,
      leading or trailing zeros are added.
      If the number has more digits to the right of the
decimal
      point than there are zeros to the right of the decimal
point
      in the format expression, then the number is rounded to
as
      many decimal places as there are zeros in the mask.

      If the number has more digits to the left of the decimal
      point than there are zeros to the left of the decimal
point
      in the pattern, then the extra digits are displayed
without
      truncation.

      If the numeric value is negative, the negation symbol
will
      be treated as a decimal digit. Therefore take care, when
      displaying negative values with this placeholder style.
      In such cases, it is recommended that multiple patterns
are
      be used.

     # - digit placeholder. If there is a digit for this
position,
      FMT. replaces this placeholder with that digit, with
nothing,
      or with a user-specified character. Unlike the 0 digit
      placeholder, if the numeric value has the fewer digits
than
      there are # characters on either side of the decimal
      placeholder, FMT. will either omit this character
position
      from the final formatted string; or substitute a
      user-specified replacement character if one has been
defined
      (see the asterisk (*) character for more information).
      To specify leading spaces, prefix the mask with "* "
      (asterisk and a space character).

      PUS.0.23
      FMT.|*=##.###
      DBP.$$000
      ENR.
      ' Result is "===0.23="

   . (period-sign)  Decimal placeholder. Determines the
position
      of the decimal point in the result.

      If any numeric field is specified to the left of the
decimal
      point, at least one digit will always result, even if
only a
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
      zero.  The zero is not considered to be a "leading" zero
if
      it is the only digit to the left of the decimal.
Placing
      more than one period character in the fmt$ string will
      produce undefined results.

   %  (percent-sign) Percentage placeholder. FMT. multiplies
      the specified source number with 100, and adds a
trailing
      percent symbol.  For example:

      VAN.$$NUM=0.2
      FMT.$$NUM|0.0%
      DBP.$$NUM
      ENR.
     ' Result is 20.0%

,  (comma-sign) Thousands separator. Used to separate
thousands
   from hundreds within a number that has four or more digits
to
   the left of the decimal point.  In order to be recognized
as a
   format character, the comma must be placed immediately
after a
   digit placeholder character (also see Restrictions below).

   VAN.$$NUM=2312.56
   FMT.$$NUM|$00,000.00
   DBP.$$NUM
   ENR.
   ' Result is $02,312.56

*x  (asterisk + specified character of your choice)
   Digit placeholder and fill-character.
   Instructs FMT. to insert a digit or character "x" in that
   position. If there is a digit in num_expression at the
position
   where the * appears in the format string, that digit is
used.
   Otherwise, the "x" character is used (where "x" represents
your
   specified character). The *x specifier acts as two digit
(#)
   fields.

   VAN.$$NUM=8888.9
   FMT.$$NUM|$**####,.00
   DBP.$$NUM
   ENR.
   ' Result is $**8,888.90
"  (double-quote) Quoted string. FMT. treats all characters up
to
   the next quotation mark as-is, without interpreting them as
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   digit placeholders or format characters.  Also see
backslash.

   VAN.$$NUM=54.3
   VAR.$$FOR="x="#
   FMT.$$NUM|$$FOR
   DBP.$$NUM
   ENR.
   ' Result is "x=54"
```

\x (backslash) Escaped character prefix. FMT. treats the character
"x" immediately following the backslash (\) as a literal character rather than a digit placeholder or a formatting character. Many characters in a mask have a special meaning and cannot be used as literal characters unless they are preceded by a backslash.
The backslash itself is not copied. To display a backslash, use two backslashes (\\). To display a literal double-quote, use two double-quote characters.
To simplify the mask string for common numeric formats, FMT. permits the dollar symbol, the left and right parenthesis symbols, the plus and minus symbols, and the space character ("$()+- ") to pass through from the mask string into the formatted output string, without requiring an escape (\) prefix character.

Example

```
'************************************
' FMT.-Sample
'************************************
' Format up to 6 digits
$$NUM=125.22223
FMT.$$NUM|(4+2)
DBP.$$NUM
END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

You cannot pass a non-numeric expression or text-string to be formatted.
Also you should not place more than one decimal point in the pattern.

`FMT.` can return the maximum possible number of digits (that is up to 4932 for Extended-precision); however, the resulting digits will be meaningless beyond the actual precision.
Consequently, the specified value may produce formatted strings that are wider than the length of the specified pattern, for example:

A$ = FORMAT$(3e30!, "#,###") ' returns 41 characters

Rounding, if necessary, is implemented by the "banker's rounding" principle: if the fractional digit being rounded off is exactly five, with no trailing digits, the number is rounded to the nearest even number. This provides better results, on average, and follows the IEEE standard. For example:

```
VAN.$$NUM=3e30
VAR.$$FOR=#,###
FMT.$$NUM|$$FOR
DBP.$$NUM
ENR.

Result is:
3,000,000,000,000,000,000,000,000,000,000,000
```

See also:

- 1.5.2 Working with Text-Strings 110
- VSL. - Variable Set Left 3113
- VSM. - Variable Set Middle 3116
- VSR. - Variable Set Right 3119
- FMT. - Format Numeric Output 3107
- VBT. - Variable-Both-sides-Trim 3225
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.12.2 VSL. - Variable Set Left

MiniRobotLanguage (MRL)

VSL. Command

Variable Set Left Aligned

Intention

This command can be used to re-format a string in a way that it will fit into a table, left-aligned. You can also specify the fill-character for the alignment.
Here is an example, we use the "*" as fill-character:

```
$$TXT=Peter
VSL.$$TXT|10|*
DBP.<$$TXT>
' Result is <Peter*****>
```

In the second operation mode you can omit the fill-character, in this case just a SPACE (`$sp$`) is used as fill-character. It looks like this:

```
$$TXT=Peter
VSL.$$TXT|10
DBP.<$$TXT>
' Result is <Peter     >
```

Syntax

VSL.P1[|P2][|P3]

Parameter Explanation

P1 - Source and Target-variable.

P2 - ***(optional)*** if given then **P1** specifies the number of fill-characters to be used. If omitted, 20 is used as default.

P3 - ***(optional)*** If specified, this is the fill-character to be used. If omitted, SPACE (ASCII 32) is used as default.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'************************************
' VTU./VTL./VTM.- DEMO
'************************************
'
$$TXT=Peter
VSL.$$TXT|10|-
DBP.<$$TXT>
$$TXT=Peter
VSM.$$TXT|10|#
DBP.<$$TXT>
$$TXT=Peter
VSR.$$TXT|10|+
DBP.<$$TXT>

$$NUM=125.22222
FMT.$$NUM|00.000
DBP.$$NUM
ENR.
```

Will give this output:

```
<Peter----->
<##Peter###>
<+++++Peter>
125.222
```

Remarks

-

Limitations:

You can not specify an Null-String (empty variable) as Fill-character. Trying to do so, will make that `$sp$` will be used for the alignment.

See also:

- 1.5.2 Working with Text-Strings 110
- VSL. - Variable Set Left 3113
- VSM. - Variable Set Middle 3116
- VSR. - Variable Set Right 3119
- FMT. - Format Numeric Output 3107
- VBT. - Variable-Both-sides-Trim 3225
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.12.3 VSM. - Variable Set Middle

MiniRobotLanguage (MRL)

VSM. Command

Variable Set Middle (Center-) Aligned

Intention

This command can be used to re-format a string in a way that it will fit into a table, center-aligned. You can also specify the fill-character for the alignment.
Here is an example, we use the "*" as fill-character:

```
$$TXT=Theo
VSM.$$TXT|10|*
DBP.<$$TXT>
' Result is <***Theo***>
```

In the second operation mode you can omit the fill-character, in this case just a SPACE (`$sp$`) is used as fill-character. It looks like this:

```
$$TXT=Peter
VSM.$$TXT|10
DBP.<$$TXT>
' Result is <  Peter   >
```

Syntax

VSM.P1[|P2][|P3]

Parameter Explanation

P1 - Source and Target-variable.

P2 - ***(optional)*** if given then **P1** specifies the number of fill-characters to be used. If omitted, 20 is used as default.

P3 - ***(optional)*** If specified, this is the fill-character to be used. If omitted, SPACE (ASCII 32) is used as default.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************
' VTU./VTL./VTM.- DEMO
'************************************
'
$$TXT=Peter
VSL.$$TXT|10|-
DBP.<$$TXT>
$$TXT=Peter
VSM.$$TXT|10|#
DBP.<$$TXT>
$$TXT=Peter
VSR.$$TXT|10|+
DBP.<$$TXT>

$$NUM=125.22222
FMT.$$NUM|00.000
DBP.$$NUM
ENR.
```

Will give this output:

```
<Peter----->
<##Peter###>
<+++++Peter>
125.222
```

Remarks

-

Limitations:

You can not specify an Null-String (empty variable) as Fill-character. Trying to do so, will make that `$sp$` will be used for the alignment.

See also:

- 1.5.2 Working with Text-Strings 110
- VSL. - Variable Set Left 3113
- VSM. - Variable Set Middle 3116
- VSR. - Variable Set Right 3119
- FMT. - Format Numeric Output 3107
- VBT. - Variable-Both-sides-Trim 3225
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.12.4 VSR. - Variable Set Right

MiniRobotLanguage (MRL)

VSR. Command

Variable Set Right Aligned

Intention

This command can be used to re-format a string in a way that it will fit into a table, right -aligned. You can also specify the fill-character for the alignment.
Here is an example, we use the "*" as fill-character:

```
$$TXT=Theo
VSR.$$TXT|30|*
DBP.<$$TXT>
' Result is <******Theo>
```

In the second operation mode you can omit the fill-character, in this case just a SPACE (`$sp$`) is used as fill-character. It looks like this:

```
$$TXT=Peter
VSR.$$TXT|10
DBP.<$$TXT>
' Result is <     Peter>
```

Syntax

VSR.P1[|P2][|P3]

Parameter Explanation

P1 - Source and Target-variable.

P2 - ***(optional)*** if given then **P1** specifies the number of fill-characters to be used. If omitted, 20 is used as default.

P3 - ***(optional)*** If specified, this is the fill-character to be used. If omitted, SPACE (ASCII 32) is used as default.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************
' VTU./VTL./VTM.- DEMO
'************************************
'
$$TXT=Peter
VSL.$$TXT|10|-
DBP.<$$TXT>
$$TXT=Peter
VSM.$$TXT|10|#
DBP.<$$TXT>
$$TXT=Peter
VSR.$$TXT|10|+
DBP.<$$TXT>

$$NUM=125.22222
FMT.$$NUM|00.000
DBP.$$NUM
ENR.
```

Will give this output:

```
<Peter----->
<##Peter###>
<+++++Peter>
125.222
```

Remarks

-

Limitations:

You can not specify an Null-String (empty variable) as Fill-character. Trying to do so, will make that `$sp$` will be used for the alignment.

See also:

- 1.5.2 Working with Text-Strings 110
- VSL. - Variable Set Left 3113
- VSM. - Variable Set Middle 3116
- VSR. - Variable Set Right 3119
- FMT. - Format Numeric Output 3107
- VBT. - Variable-Both-sides-Trim 3225
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.13 GFS. - Get-Filtered-String

MiniRobotLanguage (MRL)

GFS. Command

Get Filtered String

Intention

This command implements a string-filter. A string-filter is a thing that filters a string in a way that only allowed chars are left in that string.
While some uses of this command are simple, the command itself implements a complex string processor that can be customized to filter your strings in many ways.

IMPORTANT: You need to additionally specify the "z" or "x" or "y" or "o","p" Prototype to get the filtered result calculated.
Please specify these in the right order, as they are processed one after the other from left to right.

Example:

```
  ' Will work
GFS.tz|$$TXT

  ' Will NOT work, because z is used before t is processed
GFS.zt|$$TXT
```

Mode 1: the Keep Chars-Filter

Let's see an example. We use two prototypes here:
t - select filter chars for "time string"
z - keep only filter-chars

Note that the **sequence of the prototypes is important** here. The string processor **evaluates the prototypes from left to right**. We have first to select the "filter chars" then we can use them in a filter-set.

© 2023 Theo Gottwald * https://www.smart-package.com

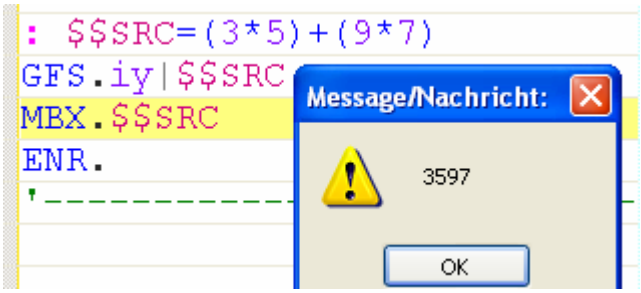
Thats why you can not write::

```
GFS.zt|$$SRC
```

If you do so,.the result will be an empty string.

Lets see another example.

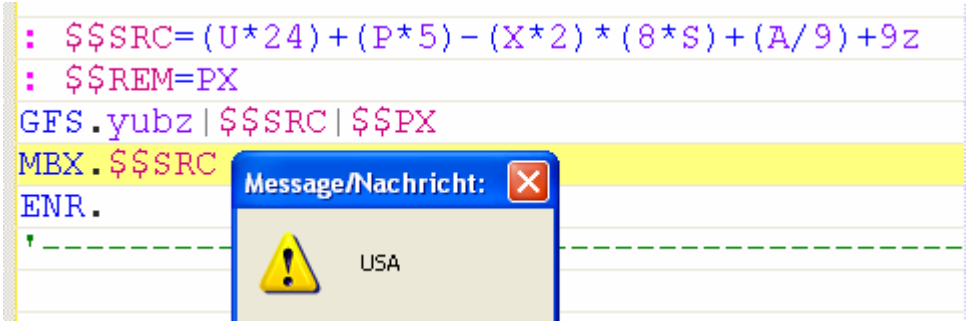
Mode 2: the Remove Chars-Filter



In the RCF-Mode, the selected chars will be removed from the string. In this example we select the i-prototype which is `"E.,()<>+-/\*^"` (see below). Now all chars that are inside that filter-char-set will be removed from the source string. You can see the result in the messagebox in the picture above.

Mode 3: the Combined Mode-Filter

This mode is just a combination of the two modes above. You can combine as many filters as you want in a single `GFS.`-Instruction. In this example we give a third parameter `$$REM` that is automatically added to the filter-chars at the start. The following example implements two filters.



When the filter starts, it takes the `$$REM` and adds it to the filter-char. It does this because its the third parameter.
Next it comes to the "y"-prototype and executes an "Remove all filter-chars"-Filter (RCF).
Then it comes to the "u"-prototype that will clear the filter-chars-list. We do this just for this example, it would not really be necessary here.
The next prototype is the "b", it adds all uppercase english letters to the filter-chars.
And finally we execute the "z"-prototype that is a "KCF"- "Keep Only Filter-Chars". It will remove all chars that are NOT in the filter-char list. As a result we get the string USA, as you can see in the picture.

Mode 4: The two special filters

© 2023 Theo Gottwald * https://www.smart-package.com

There are two special filters included which are not influenced by the filter-char list and the other commands. These two prototypes will just work on the given string and immediately process it. These are the "o" and the "p"-prototype. They serve a special purpose.
The **"o" - prototype** will remove all ASCII-Chars below 32 from your given input string.
To understand the next prototype you will need to know that the normal "Space"-Character has the ASCII of 32. But there is another one which look equal and has an ASCII of 160. This one can make processing of strings for you difficult because you may think "Its an Space" but in reality its an ASCII 160 that is in fact not an Space.
To overcome that problem, **the filter option "p"** will replace any ASCII 160 to an ASCII of 32.

Mode 5: Use a custom filter string from stack

We have already shown how to give a custom filter-char-set, using a third parameter. You can add a value from stack, using the "c" prototype. See the example in the next picture.

You can push the current filter-char-set at any time to the stack, using the "q" prototype. You can also put the current input-string onto the stack, using the "r"-prototype.

```
: $$SRC=(U*24)+(P*5)-(X*2)*(8*S)+(A/9)+9z
: $$REM=PX
PUV.$$REM
STS.DUMP
GFS.cyubz|$$SRC|$$PX
MBX.$$SRC
ENR.
```

Important note:
As this commands implements a "string processor" the sequence of the prototypes is important, unlike in most other commands. The prototype are processed in the order they are provided.

Syntax

GFS.P1[|P2][|P3]

Parameter Explanation

P1 - Prototypes.

Options:

© 2023 Theo Gottwald * https://www.smart-package.com

```
a - force binary compatible resolution for string parameter*

Filter-Char prototypes:
b - "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
c - Add "custom" TOS to Filter-chars
d - add "0123456789"

' add Chars that are allowend in an e-mailadress before the @
e - "!#$%&'*+-/=?^_`.{|}~"

f - add "0123456789."

' add german Umlauts
g - "äöüßÄÜÖß"

' add hex-numbers
h - "aAbBcCdDeEfF0123456789"

' add mathematical symbols
i - "E.,()<>+-/\*^"

' add grammatic text symbols
j - "!?.,;-()"

' "k" - add english letter-set
k - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

l- add german letter set
"abcdefghijklmnopqrstuvwxyzäöüABCDEFGHIJKLMNOPQRSTUVWXYZÄÖÜß"

m - add valid Filename-chars excluding the \ and :
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß[]
{}@.1234567890_-;:+#§%&()=! "

n - Add ":\"

'  add "small letters"
s - "abcdefghijklmnopqrstuvwxyz"

' add time-symbols
t - ":0123456789"

Special filters (Immediately executed)

o - Remove ASC 0 - 31 from input-string (immediately)
p - Replace ASC 160 with ASCII 32 (immediately)

r - current filter-charset -> TOS
q - current input-string on TOS
u - empty/clear filter-char set
v - valid Pathnames; contains all chars that are valid
    in a pathname.

w - valid filenames
x - Remove exact match of filter-char-st from input-string
y - Remove any letter in filter-char-st from Input-string
z - Keep any letter that is in filter-char set
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    and remove all other letters.

P2 - (optional) Variable with input-String that is
     going to be filtered. If given, result will be in
     same variable.
     If omitted, string is taken from TOS. In this case,
     filtered string is also put n TOS.

P3 - (optional) Filter-Char-String. If given, will be added
      to Filter-Char-Set at start of operation.

* binary compatible variable resolution:
Only the given variable name is resolved. Special-variables or
variable names inside the variable are not resolved.
```

Example

```
'************************************
' GFS.-Test
'************************************
: $$SRC=(U*24)+(P*5)-(X*2)*(8*S)+(A/9)+9z
: $$REM=PX
PUV.$$REM
STS.DUMP
GFS.cyubz|$$SRC|$$PX
MBX.$$SRC
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.14 GRE. - Get Regular Expression

MiniRobotLanguage (MRL)

GRE. Command

Get Regular Expressions

Intention

Enables you to extract Parts from Strings or test if a string matches a regular expression.

What are regular expressions good for?

Ok, they are a challenge if you use them first time. But Challenges are something positive. Catch them. At the end of a challenge you will know yourself better - and you will have new abilities.

When you know about regular expressions, you can do things the easy way. Things, you would have never thought about before.
The following line will extract the E-Mail adress from the line.

```
' Use Regular Expression to extract an e-mail address
: $$NUM=2
: $$MAN=please send an email to info@it-berater.org
VAS.$$MSK=([a-z0-9._/+-]+)(@[a-z0-9.-]+)
' This will do the Job.
GRE.$$NUM,$$MAN,$$MSK
STS.DUMP
DMP.
END.
```

See Regular Expression 381

Syntax

GRE.[P1],P2,P3

Parameter Explanation

P1* - *(optional) `Start of Stringscan. This can be a Variable containing a number or a plain number.`

P2 - `VAR` with Main String. Can only be a Variable.

P3 - `VAR` with Regular Expression. This Variable contains the regular Expression.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 and **P3** must be Variables. You cannot just write the Regular Expression in the Parameter-Field because the **$** and the | are special-characters in Regular Expressions.

Please note that for this command, you have to use the "," instead of the "|" to separate the Parameters.

Example

```
' Use Regular Expression to extract an e-mail address
: $$NUM=2
: $$MAN=please send an email to info@it-berater.org
VAS.$$MSK=([a-z0-9._/+-]+)(@[a-z0-9.-]+)
GRE.$$NUM,$$MAN,$$MSK
STS.DUMP
DMP.
END.

' Use Regular Expression to extract the value
VAS.$$MSK=\$[0-9.,]+
: $$MAN=Der Schuldbetrag ist $39.75 plus Steuern
GRE.$$NUM,$$MAN,$$MSK
STS.DUMP
DMP.
END.
```

Remarks

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GRR. - Get Regular Replace 3131
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.15 GRR. - Get Regular Replace

MiniRobotLanguage (MRL)

GRR. Command

Get Regular Replace

Intention

`GRR.` Enables you to replace Parts from Strings which match a regular expression.

What is it really good for?

Using "regular expressions" you can do things that would be difficult to realize in other ways. Take our example. You want to replace all e-mail addresses in a document automatically with the term "[e-mail]. Sounds difficult?

Is easy:

```
' Use Regular Expression to replace an e-mail adress
: $$NUM=2
: $$MAC=please send an email to info@it-berater.org
: $$MSK=([a-z0-9._/+-]+)(@[a-z0-9.-]+)
: $$REP=[E-Mail]

DBP.Before:
DBP.$$MAC

GRR.$$NUM,$$MAC,$$MSK,$$REP

DBP.After Replace
DBP.$$MAC
END.
```

See Regular Expression 381

Syntax

GRR.[P1],P2,P3,P4

Parameter Explanation

P1 **- *(optional)*** `NUM/VAR`

— *Start of Stringscan.*

— This can be a Variable containing a number or a number.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 **-** VAR
— Main String. Can only be a Variable of any Type.
— Non-String-Types will be converted to a Textstring.

P3 **-** VAR Regular Expression. This Variable contains the regular Expression.

P4 **-** VAR

P2 and **P3** must be Variables.
You cannot just write the Regular Expression in the Parameter-Field as the $ and the | are special-characters in Regular Expressions.

Please note that for this command, you have to use the "," instead of the "|" to separate the Parameters.

Example

```
' Use Regular Expression to replace an e-mail adress
: $$NUM=2
: $$MAC=please send an email to info@it-berater.org
: $$MSK=([a-z0-9._/+-]+)(@[a-z0-9.-]+)
: $$REP=[E-Mail]

DBP.Before:
DBP.$$MAC

GRR.$$NUM,$$MAC,$$MSK,$$REP

DBP.After Replace
DBP.$$MAC
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- GRE. - Get Regular Expressio 3128
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067

© 2023 Theo Gottwald * https://www.smart-package.com

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.16 GSC. - Get String Count

MiniRobotLanguage (MRL)

GSC. Command

Get Substring/Delimiter Count

Intention

Returns the number of occurences of a given substring or delimiter within a main-string.
There are several modes of operation. General usage is easy:

```
GSC.$$MAI|$$SUB
DBP.Result is: $$000
```

Which will get the number of delimiters. You can also count the number of delimited elements by adding the flag "**d**".

```
GSC.$$MAI|$$SUB||d
DBP.Result is: $$000
```

If you add the flag "**a**" the nany of the elemnts in P2 will be treated as delimiter. Like this:

```
$$MAI=Ralf,Peter;Mark.Paul
$$SUB=;,.
GSC.$$MAI|$$SUB||da
DBP.Result is: $$000
```

Syntax

GSC.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Main-String
P2 - Sub-String/Delimiter
P3 - ***(optional)*** Variable for the result of the operation.
If omitted, TOS is used.
P4 - ***(optional)*** Flags. Can be any or all of the following letters:

"a" - then ANY of the characters in P2 is been treated as delimiter/string to count.
"d" - then elements are counted, rather then delimiters, result will be higher.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
"e" - equal-case-mode is switched on
```

Example

```
'**********************************
' GSC.-Sample
' Count delimited Elements
'**********************************
' Use Elements-Coint Flag: "d"
$$MAI=Haus;Ochse;Karl
$$SUB=;
GSC.$$MAI|$$SUB||d
DBP.Result is: $$000
ENR.

'**********************************
' GSC.-Sample 2
' Count delimiters
'**********************************
' Use Elements-Coint Flag: "d"
$$MAI=Haus;Ochse;Karl
$$SUB=;
GSC.$$MAI|$$SUB
DBP.Result is: $$000
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067

© 2023 Theo Gottwald * https://www.smart-package.com

- LEN. - Length-of-String 3140
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.17 GSS. - GetSplitString

MiniRobotLanguage (MRL)

GSS. Command

Get Split String

Intention

Returns a part of a string/text. If you have a text/string and you want just a part of it, think of `GSS.` Usage is easy:

```
VAR.$$TXT=My name is Peter, Tel.12345
DBP.$$TXT
' The result is the complete text

' Now we are going to split this string into pieces
' We put the left two characters into $$RES
GSS.1|2|$$TXT|$$RES
DBP.$$RES
' The result is "My"

' We put the right five characters into $$RES
GSS.-5|-1|$$TXT|$$RES
DBP.$$RES
' The result is "12345"


' We put characters 3 to 8  into $$RES
GSS.3|8|$$TXT|$$RES
DBP.$$RES
' The result is "name"
```

Syntax

GSS.P1|P2[|P3][|P4]

Parameter Explanation

P1 - Starting position
P2 - Ending position
P3 - Source Variable
P4 - Target Variable

*© 2023 Theo Gottwald * https://www.smart-package.com*

There are three versions of GSS. depending on how many parameters are specified.
The first two position parameters must always be given.

If **P1** or **P2** are negative, the character is counted starting from end of source-string backwards to the first character on the string.

GSS.P1|P2

The Source String is taken from **TOS**.
P1 gives the first character of the substring.
P2 gives the last character of the substring to extract.
The Result is been placed on **TOS**.

GSS.P1|P2|P3

The Source String is taken from **P3**.
P1 gives the first character of the substring.
P2 gives the last character of the substring to extract.
The Result is been placed on **TOS**.

GSS.P1|P2|P3|P4

The Source String is taken from **P3**.
P1 gives the first character of the substring.
P2 gives the last character of the substring to extract.
P4 takes the Result.

For all three versions of GSS.:
If **P1** or **P2** are negative, the character is been counted from end of source-string.

Example

```
'**********************************
' GSS.-DEMO
'**********************************
' GSS needs at least two parameters
' String total len=11
: §§TXT=ABCDEFGHIJK
PUV.§§TXT
DMP.4
GSS.2|3
DMP.4
PRT.-------------------------
PUV.§§TXT
DMP.4
GSS.-6|-4
DMP.4
PRT.-------------------------
PUV.§§TXT
DMP.4
: §§NUM=4
GSS.(§§NUM+1)|(-§§NUM)|§§SRC|§§RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.Result: >§§RES<
DMP.
END.
```

Remarks

Variables inside the source are only resolved if the length of the source string is 5. Therefore it can safely be used fort for Binary data.
For the same reason, Special-Characters etc. are not resolved, to avoid unwanted changes in binary data..

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- ACH. - After Character 3043
- BCH. - Before Character 3049
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.18 LEN. - Length-of-String

MiniRobotLanguage (MRL)

LEN. Command

LENgth of string

Intention

This command returns the length of a string in bytes. Usage is very intuitive:

```
: §§TXT=ABCDEFGHIJK

' Will return 11 on TOS
LEN.§§TXT
DBP.$$000


' Will return 11 in variable $$RES
LEN.§§TXT|$$RES
DBP.$$RES
```

Syntax

LEN.[P1][|P2]

Parameter Explanation

P1 - String to get Len from.
P2 - ***(optional)*** Variable to return the String-Len

All Parameters can be omitted. In each case the Source or Destination is TOS.

Example

```
'**********************************
' LEN.-Sample
'**********************************
: §§TXT=ABCDEFGHIJK
' Will return 11 on TOS
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
LEN.$$TXT
DBP.$$000
'
' Will return 11 in variable $$RES
LEN.$$TXT|$$RES
DBP.$$RES

END.

'**********************************
' LEN.-Test
'**********************************
VAR.$$TXT=12345678
PUV.$$TXT
DMP.4
' String from TOS, LEN-> TOS
LEN.
DMP.4
' Len->TOS
LEN.$$TXT
DMP.4
LEN.$$TXT|$$RET
PRT.$$RET
DMP.
END.
```

Remarks

-

Limitations:

The theoretical maximum length of a variable /text / string is 2 GB. In real life, it is less, because variables need string memory and that must not be fragmented. After all the maximum size of a variable on your system depends on the amount of RAM that is available, and it depends on how fragmented your memory is. Under normal circumstances several hundred MegaBytes should be no problem.

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

© 2023 Theo Gottwald * https://www.smart-package.com

- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.19 RPL. - RePLace in String

MiniRobotLanguage (MRL)

RPL. Command

RePLace in string

Intention

This command can be used to replace parts of a string with another.

There are several options, which can be handy, if you want to replace in binary strings or for special tasks.

Syntax

RPL.P1|P2|P3[|P4]

Parameter Explanation

P1 - Source and Destination
P2 - Replace-String (this one is replaced)
P3 - with-String (this is what it is replaced with)
P4 - Options, see below:

You can only specify one flag for RPL. as **P4**.
These are the flags for P4:

a - "ANY" & binary-compatible.
Using this one you will replace single letters
with single letters.

b - binary compatible does not resolve special-folders or
system-variables

c - "ANY" using this one you will replace single letters
with single letters. Not binary compatible.

e - Equalcase-Replace. Replace is done in Equalcase-Mode,
the Result will anyway keep its case.

f - fast replace. Default mode if no option is specified.

w - "Whole Words Algo"-Mode. in this mode only words are
replaces
and only if they are not enclosed in " ".

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
First Parameter is a Variable, local or Global.
If invalid ("-") for example, Stack is used.
```

Speed in Ticks:

This command uses typically around 170 to 230 Ticks heavily depending on Parameters.

Example

```
'**********************************
' RPL.-Test
'**********************************
PRT. Do a Fast-Replace
: $$TXT=Hallo "PETER",Peter, PETER when PETERSCHMIDT...
: §§REP=PETER
: §§WIT=RALF
PRT.$$TXT
RPL.$$TXT|§§REP|§§WIT
PRT.$$TXT
PRT.------------------------------
PRT. Do a Equalcase-Replace
: $$TXT=Hallo "PETER",Peter, PETER when PETERSCHMIDT...
: §§REP=PETER
: §§WIT=RALF
PRT.$$TXT
RPL.$$TXT|§§REP|§§WIT|e
PRT.$$TXT
PRT.------------------------------
PRT. Use "whole-word Algo"
: $$TXT=Hallo "PETER",Peter, PETER when PETERSCHMIDT...
: §§REP=PETER
: §§WIT=RALF
PRT.$$TXT
RPL.$$TXT|§§REP|§§WIT|w
PRT.$$TXT
PRT.------------------------------
PRT. Use "any matchstring"
: $$TXT=Hallo "PETER",Peter, PETER when PETERSCHMIDT...
: §§REP=PETER
: §§WIT=RALF
PRT.$$TXT
RPL.$$TXT|§§REP|§§WIT|c
PRT.$$TXT
PRT.------------------------------
DMP.
END.

'**********************************
' RPL.-Sample
' Here we use RPL to replace Text in
' a line-based Textfile
' We search for a Date like 11.?7.*
' and replace it with 12.08.2015
'**********************************
$$FIL=?path\Test.txt
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$TAR=?path\Target.txt
' Delete it if it existed before
DEL.$$TAR
' We search for the line with the date
$$SRC=11.?7.*
' New date - we replace the old one with this
$$NEW=12.08.2015
FIF.tww|$$FIL|$$SRC
JIZ.$$000|Lab_nf
$$LIN=$$000
$$STA=$$000
$$END=$$000
$$STR=$$000
$$TRE=$$000
PRT.String: $$STR Found in Line: $$LIN
PRT.From Character $$STA to $$END
GLC.$$FIL|$$REA
' Line where String is minus 1
CAL.$$TOL=$$LIN-1
' Copy all lines until the wanted Line
FOR.$$COP|1|$$TOL
  LFF.$$FIL|$$COP|$$RES
   ' We use the delimiter that was found in the source file
  $$TXT=$$RES$$TRE
  ATF.$$TAR|$$TXT
NEX.
   ' Here is the Line with the date
LFF.$$FIL|$$LIN|$$TXA
RPL.$$TXA|$$STR|$$NEW
$$TXB=$$TXA$$TRE
ATF.$$TAR|$$TXB
' Now copy Rest of file
' Line where String is plus 1
CAL.$$TOL=$$LIN+1
FOR.$$COP|$$TOL|$$REA
  LFF.$$FIL|$$COP|$$RES
   ' We use the delimiter that was found in the source file
  $$TXT=$$RES$$TRE
  ATF.$$TAR|$$TXT
NEX.

:Lab_nf
MBX.!
ENR.
```

Remarks

If you use the ANY option, within MainString, each occurrence of each character in MatchString will be replaced with the corresponding character in NewString. In this case, MatchString and NewString must be the same length, because there is a one-to-one relation.

"|w" - replaces only whole words (no words in word, no words in qoutes) in text but not within double quotes

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

Due to speed-reasons, "|w"-Algo has the limitation, that the result-string cannot exceed the source in size by more then 100 KB.
If you need more then just append enough spaces to the source string before.
This limitation should never be a problem in practical use.

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.20 SRT. - Sort Items

MiniRobotLanguage (MRL)

SRT. Command

SoRT delimited items

Intention

This command is used to sort delimited items alphabetically.

Items in this definitions are members of a string, delimited with a known delimiter. In a Text-file which has been loaded, the delimiter would be `$crlf$` for example.

Usage can be as easy as calling:

```
VAR.$$TXT=Name Last-Name Zip-Code
SRT.$$TXT
DBP.$$TXT
```

Now for special reasons you can specify more parameters. For example:

```
VAR.$$TXT=Name Last-Name Zip-Code
SRT.$$TXT|$sp$|d
DBP.$$TXT
' returns "Last-Name Name Zip-Code"
```

or change to ascending sort-order:

```
VAR.$$TXT=Name Last-Name Zip-Code
SRT.$$TXT|$sp$|a
DBP.$$TXT
' returns "Zip-Code Name Last-Name"
```

Syntax

SRT.P1[|P2][|P3][|P4]

Parameter Explanation

P1 - String with delimited elements to sort.

P2 - ***(optional)*** Delimiter-Char

P3 - ***(optional)*** "a" or "d" - ascending sort or descending sort-order

*© 2023 Theo Gottwald * https://www.smart-package.com*

P4 - ***(optional)*** "Tag-String".
This string is internally generated and shows how the source
string has been sorted.

If you omit the P2 delimiter, the command will try to find it automatically.

Example

```
' Keyword-Sorter
' This script reads a file with keywords
' it replaces the delimiter and removes double delimiters
' and double Keywords.
' It will then sort the keywords and save them in another file.
'
' Its an example for RPL., FEM. and SRT.
'
'-------------------------

': §§SRF=Gesamt-Original
: §§SRF=reinigungsmittel

: §§FIL=?pfad\§§SRF.txt
: §§TAA=?pfad\§§SRF-csv.csv
: §§TAB=?pfad\§§SRF-txt.txt
VAR.§§SON=$crlf$$tab$
VAR.§§SOA=$sp$
: §§TMP=
: §§COA=
DEL.§§TAR
CFF.§§FIL|§§CON
RPL.§§CON|,|;
RPL.§§CON|  |§§SOA|d
RPL.§§CON|§§SON|;|a
RPL.§§CON|;;|;
RPL.§§CON|; ;|;
RPL.§§CON|;;|;
SRT.§§CON|;
FEM.DoIt|§§MEM|§§CON|;
CTF.§§TAA|§§COA
RPL.§§COA|;|,
CTF.§§TAB|§§COA
END.

:DoIt
IVV.§§TMP!§§MEM
 PRT.Adding: §§MEM
  VAR.§§COA=§§COA;§§MEM
ELS.
 PRT. Deleting: §§MEM
EIF.
VAR.§§TMP=§§MEM
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- ! STR. - String Command 3150
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21 STR.- String Command

3.79.21.1 ! STR. - String Command

MiniRobotLanguage (MRL)

STR. Command

String commands

Intention

This command contains a set of String-Commands. You can use it mainly to change binary strings. For this, variables will only be resolved one time (binary compatible). specialfolders and systemvariables are will be evaluated, to prevent unwanted changes in binary data.

The command has several modes of operation. Each of these is like a separate command.

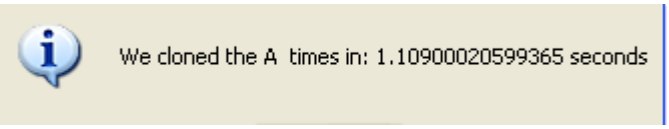
- CLONE - Mode 3150. This Mode will create a new String. out of a second string. The result is a string, consisting of multiple copies of the second specified string.
- DELETE - Mode 3150: Delete bytes/chars from inside the string.
- FIND - Mode 3176: Search for a String inside a String. Result will be on TOS. String is NOT changed in this Mode.
- INSERT - Mode 3192: This Mode will insert a String at a place into the Source String.
- OVERWRITE - Mode: 3205 This Mode will patch the Source String with a second Patch String. It will "overwrite" the Source String, while it will never change the total length of the String.
- PATCH - Mode: 3208 This Mode combines the DELETE and the INSERT Mode. Unlike the OVERWRITE-Mode, this Mode can change the length of a String.
- REVERSE - Mode 3212: This will reverse all bytes in a String. The last byte will be the first byte.
- GENERATE 3188 RANDOM STRING - Mode: This will generate a random string from specified length and of specified characters.
- STR.CHARS A 3155 - OTP-PRNG. This CPRNG is based on the ANSI X9.17 standard for secure key generation and uses Rijndael as its cryptographic base algorithm.
- STR.CHARS B 3159 This is an enhanced version of CHARS A. It uses the same PRNG. This version will create between 10 and 20 Pads using CHARS A. Then all of these will be combined using the XOR.
- STR.CHARS C 3163 This CPRNG is based on the recursion-with-carry generator, invented by George Marsaglia. Estimated Cycle length is 3 x 10^47.
- STR.CHARS D 3167 This is an enhanced version of CHARS C. This version will create a Pad with the algo from STR.CHARS A 3155 first. It will then created between 4 and 8 Pads using the algo from `CHARS C`. Then all of these Pads will be combined using the XOR to get a final result.

*© 2023 Theo Gottwald * https://www.smart-package.com*

- STR.MAKE PAD 3195 MAKE PAD has the same Syntax like GENERATE RANDOM STRING - but uses internally another PRNG (Pseudo Random Number Generator) with a Periode length of about 2^60 which is in numbers (rough estimation) 1152921504000000000 before the sequence will repeat. Therefore the result should be safe for cryptographic uses.This Mode will generate a random String of specified length and from specified characters. You can use it to generate passwords.
- STR.FROM BASE64 URL 3182 will convert the MIME/BASE64 URL data that is contained in a variable, back to the original-Format. The result will be in the same variable or in a Target variable if given.
- STR.TO BASE64 URL 3219 will convert the text or binary data that is contained in a variable, to the BASE64 URL-Format. The result will be in the same variable or in a Target variable if given.
- STR.FROM HEXASCI 3185 will convert the HEXASCII-text that is contained in a variable, to the Original-Format. The result will be in the same variable or in a Target variable if given.
- STR.TO HEXASCI 3222 will convert the text or binary data that is contained in a variable, to the HEXASCII-Format. The result will be in the same variable or in a Target variable if given.
- STR.MIME TO TEXT 3198 will convert the MIME/BASE64 data that is contained in a variable, back to the original-Format. The result will be in the same variable or in a Target variable if given.
- STR.TEXT TO MIME 3215 will convert the text or binary data that is contained in a variable, to the MIME-Format. The result will be in the same variable or in a Target variable if given.

Timing considerations

```
' Timing considerations
: $$TIM=#dtime#
: $$TXT=A
: $$NUM=30000000
STR.CLONE|$$SRC|$$NUM|$$TXT
LEN.$$SRC|$$LEN
VAR.$$TIM=#dsince#
MBX.We cloned the A $tos$ times in: $$TIM seconds
```



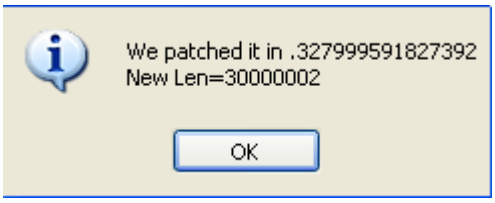
To generate a string of 29.3 MB we needed about 1 second. The times on different systems may vary, the times on my testing systems are only to give you a rough idea.

Now the STR.PATCH is possibly our slowest operation. Let's see how slow it is.

```
VAR.$$NEW=BB
VAR.$$POS=4
VAR.$$DEL=0
VAR.$$TIM=#dtime#
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
VAR.$$TIM=#dsince#
```

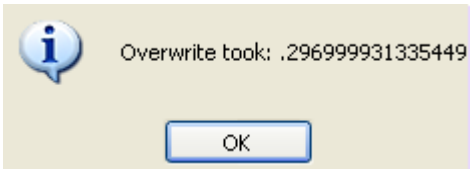

© 2023 Theo Gottwald * https://www.smart-package.com

```
LEN.$$SRC|$$LEN
MBX.We patched it in $$TIM$crlf$New Len=$$LEN
```



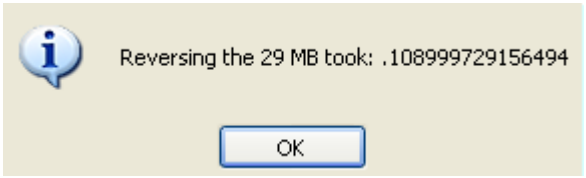
We needed 0.3 of a second to patch the 29 MB String. Now let's see how much it would be if we use `STR.OVERWRITE`.

```
VAR.$$TIM=#dtime#
STR.OVERWRITE|$$SRC|$$POS|$$NEW
VAR.$$TIM=#dsince#
MBX.Overwrite took: $$TIM
ENR.
```



Looks like there is no large difference in practical use.
Finally let's reverse that string and look how long it takes.

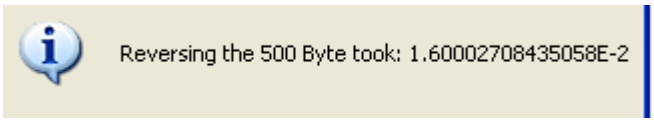
```
VAR.$$TIM=#dtime#
STR.REVERSE|$$SRC|$$POS|$$NEW
VAR.$$TIM=#dsince#
MBX.Reversing the 29 MB took: $$TIM
```



Seems not to be a big deal. Looks like you can make ten times that in a second. This should give you a rough approximation of execution times of these String commands.

Lets end with something simple.
Let's revers a string with just 500 bytes and measure this.

```
VAR.$$TIM=#dtime#
STR.REVERSE|$$SRC
VAR.$$TIM=#dsince#
MBX.Reversing the 500 Byte took: $$TIM
```



© 2023 Theo Gottwald * https://www.smart-package.com

In other words: 0.016 seconds. You can do that 62 times in a second.

General Hints:
To Replace bytes in a String, use the RPL. - RePLace in String 3143 command.
To extract a part out of a String, use the GSS. - GetSplitString 3137 command.
If your String is a delimited String (like a .csv-file), use:

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

To filter a String, use: GFS. - Get-Filtered-String 3122.

Syntax

STR.P1[|P2][|P3][|P4]

Parameter Explanation

P1 **-** MODE of operation

P2 **-** Source and (if no P2) destination variable

P3 **-** numeric parameter, will be completely resolved

P4 **-** depend on the Mode of operation

Example

```
'************************************
' STR.
'************************************
VAR.$$SRC=Alles alt macht der Mai!
VAR.$$NEU=Neu
VAR.$$ALT=alt
STR.FIND|$$SRC|1|$$ALT
VAR.$$POS=$tos$
DBP.Patching at: $$POS
STR.OVERW|$$SRC|$$POS|$$NEU
DBP.$$SRC
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Variables inside the source are only resolved once. Therefore it can be safely used fort for Binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data..

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- GFS. - Get-Filtered-String 3122
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.2 STR.CHARS A

MiniRobotLanguage (MRL)

STR.Chars A

Generate a string with random Bytes

Intention

The used PRNG (Pseudo Random Number Generator) provides a source of cryptographically strong random 'bytes for use in creating session keys, One-TIme Pads and other similar purposes.
This CPRNG is based on the ANSI X9.17 standard for secure key generation and uses Rijndael as its cryptographic base algorithm.
A cryptographically strong PRNG produces values which are not predictable.
This quality separates such generators from other software random number 'generators, which always produce a predictable stream of values regardless 'of the size of a generator's period. If cryptographic strength is not 'required, a CPRNG constitutes overkill.
However, if cryptographic strength is in fact required, only a generator designed specifically for such use is acceptable.

The command itself looks like this:

```
STR.CHARS A|$$CNT|$$RES
```

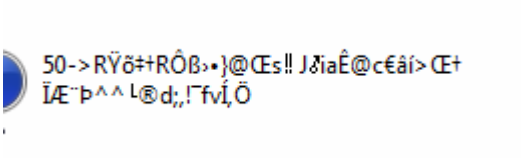
And here is a small example:

```
: $$NUM=5
STR.CHARS A|$$NUM|$$RES
MBX.$$RES
```

This is a larger example which produces a Pad of size 50.

```
: $$CNT=50
STR.CHARS A|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

It brings up this output:



Use the CTF. - Copy-To-File 964 to write it to a USB-Stick or Harddisk.

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

STR.CHARS A|P1[|P2]

Parameter Explanation

P1 **-** length of desired random string in bytes. If omitted, a default length of 32 byte length is used.

P2 **-** destination variable, if omitted TOS will be used for the result.

Example

```
'************************************
' STR.CHARS A
'************************************
: $$CNT=50
STR.CHARS A|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

Remarks

While we have made any effort to produce PRNG's of highest quality, there is no absolute warranty whatever, that these randomly generated strings are safe for use with cryptographic or safety software, or even for password-generation on a specific computer-hardware. Therefore you have to use them on your own risk.

Below you can see the results of the so called "ENT-Test Suite" for OTP's generated with this PRNG.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **STR.CHARS B** 3159
- **STR.CHARS C** 3163
- **STR.CHARS D** 3167
- CTF. - Copy-To-File 964
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- GEC. - Get Encryption 825
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929

© 2023 Theo Gottwald * https://www.smart-package.com

- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.3 STR.CHARS B

MiniRobotLanguage (MRL)

STR.Chars B

Generate a string with random Bytes

Intention

The used PRNG (Pseudo Random Number Generator) provides a source of cryptographically strong random 'bytes for use in creating session keys, One-TIme Pads and other similar purposes.
This CPRNG is based on the ANSI X9.17 standard for secure key generation and uses Rijndael as its cryptographic base algorithm.
A cryptographically strong PRNG produces values which are not predictable.
This quality separates such generators from other software random number 'generators, which always produce a predictable stream of values regardless 'of the size of a generator's period. If cryptographic strength is not 'required, a CPRNG constitutes overkill.
However, if cryptographic strength is in fact required, only a generator designed specifically for such use is acceptable.

This version will create between 10 and 20 Pads using the above algo. Then all of these will be combined using the XOR. Therefore an attack based on quality of Random numbers looks impossible from my standpoint.

The command itself looks like this:

```
STR.CHARS B|$$CNT|$$RES
```

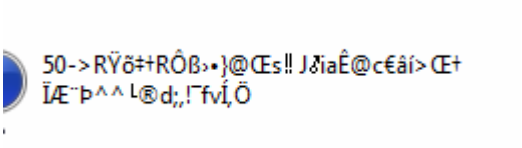
And here is a small example:

```
: $$NUM=5
STR.CHARS B|$$NUM|$$RES
MBX.$$RES
```

This is a larger example which produces a Pad of size 50.

```
: $$CNT=50
STR.CHARS B|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

It brings up this output:



© 2023 Theo Gottwald * https://www.smart-package.com

Use the CTF. - Copy-To-File 964 to write it to a USB-Stick or Harddisk.

Syntax

STR.CHARS B|P1[|P2]

Parameter Explanation

P1 **-** length of desired random string in bytes. If omitted, a default length of 32 byte length is used.

P2 **-** destination variable, if omitted TOS will be used for the result.

Example

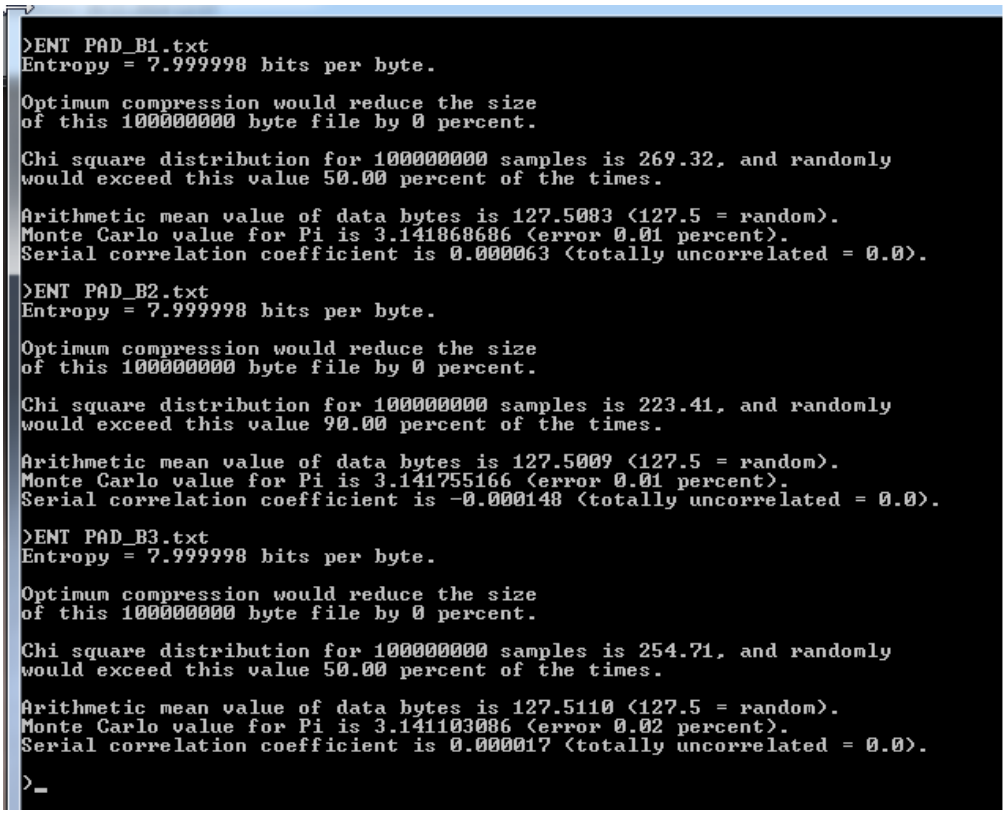
```
'************************************
' STR.CHARS B
'************************************
: $$CNT=50
STR.CHARS B|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

Remarks

While we have made any effort to produce PRNG's of highest quality, there is no absolute warranty whatever, that these randomly generated strings are safe for use with cryptographic or safety software, or even for password-generation on a specific computer-hardware. Therefore you have to use them on your own risk.

Below you can see the results of the so called "ENT-Test Suite" for OTP's generated with this PRNG.

© 2023 Theo Gottwald * https://www.smart-package.com



Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **STR.CHARS A** 3155
- **STR.CHARS C** 3163
- **STR.CHARS D** 3167
- CTF. - Copy-To-File 964
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- GEC. - Get Encryption 825
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929

© 2023 Theo Gottwald * https://www.smart-package.com

- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.4 STR.CHARS C

MiniRobotLanguage (MRL)

STR.Chars C

Generate a string with random Bytes

Intention

The used PRNG (Pseudo Random Number Generator) provides a source of cryptographically strong random 'bytes for use in creating session keys, One-TIme Pads and other similar purposes.
This CPRNG is based on the recursion-with-carry generator, invented by George Marsaglia.
Estimated Cycle length is 3 x 10^47.
This PRNG Passes All Test in ENT and in DIEHARD
ENT Can be found at: http://www.fourmilab.ch/random/
DIEHARD Can be found at: http://stat.fsu.edu/~geo/diehard.html

The command itself looks like this:

```
STR.CHARS C|$$CNT|$$RES
```

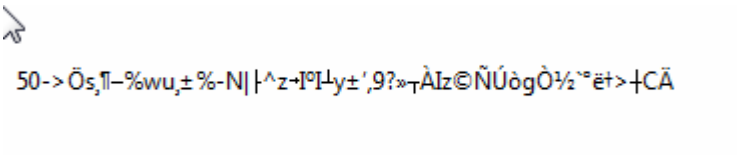
And here is a small example:

```
: $$NUM=5
STR.CHARS C|$$NUM|$$RES
MBX.$$RES
```

This is a larger example which produces a Pad of size 50.

```
: $$CNT=50
STR.CHARS C|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

It brings up this output:



Use the CTF. - Copy-To-File 964 to write it to a USB-Stick or Harddisk.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

STR.CHARS C|P1[|P2]

Parameter Explanation

P1 **-** length of desired random string in bytes. If omitted, a default length of 32 byte length is used.

P2 **-** destination variable, if omitted TOS will be used for the result.

Example

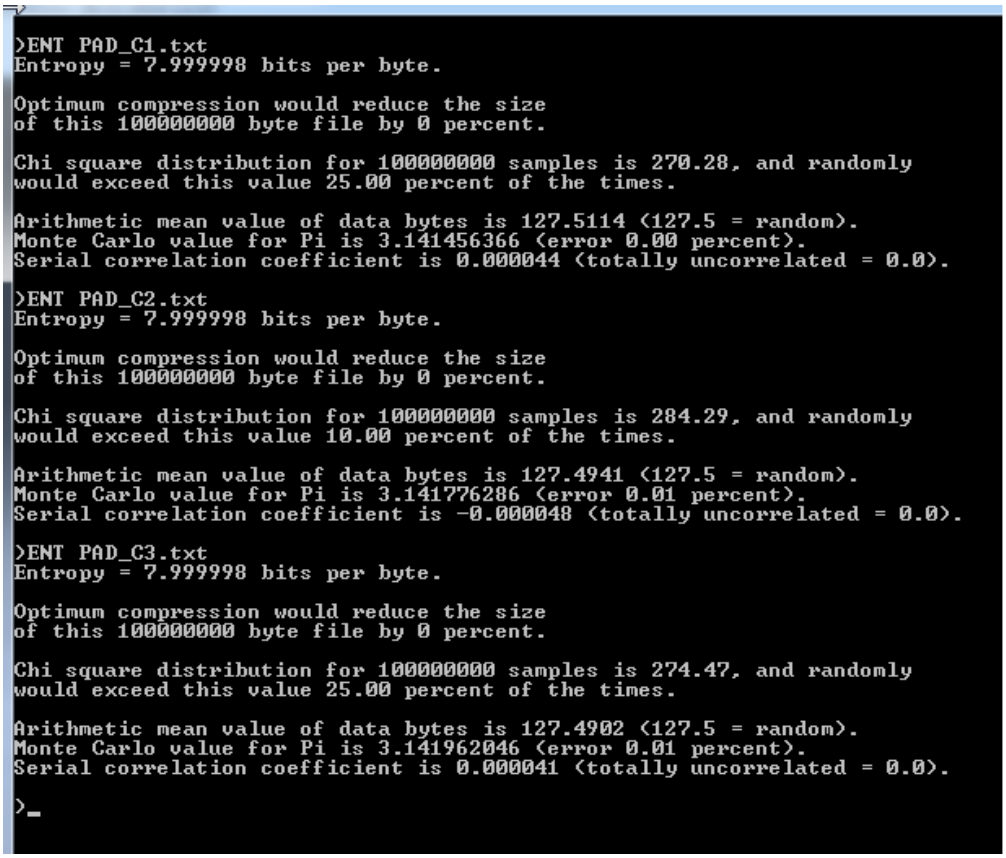
```
'************************************
' STR.CHARS C
'************************************
: $$CNT=50
STR.CHARS C|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

Remarks

While we have made any effort to produce PRNG's of highest quality, there is no absolute warranty whatever, that these randomly generated strings are safe for use with cryptographic or safety software, or even for password-generation on a specific computer-hardware. Therefore you have to use them on your own risk.

Below you can see the results of the so called "ENT-Test Suite" for OTP's generated with this PRNG.

© 2023 Theo Gottwald * https://www.smart-package.com



Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **STR.CHARS A** 3155
- **STR.CHARS B** 3159
- **STR.CHARS D** 3167
- CTF. - Copy-To-File 964
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- GEC. - Get Encryption 825
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

© 2023 Theo Gottwald * https://www.smart-package.com

- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.5 STR.CHARS D

MiniRobotLanguage (MRL)

STR.Chars D

Generate a string with random Bytes

Intention

The used PRNG (Pseudo Random Number Generator) provides a source of cryptographically strong random 'bytes for use in creating session keys, One-TIme Pads and other similar purposes.
This CPRNG is based on the recursion-with-carry generator, invented by George Marsaglia.
Estimated Cycle length is 3 x 10^47.
This PRNG Passes All Test in ENT and in DIEHARD
ENT Can be found at: http://www.fourmilab.ch/random/
DIEHARD Can be found at: http://stat.fsu.edu/~geo/diehard.html

This version will create a Pad with the algo from STR.CHARS A 3155 .
And will then created between 4 and 8 Pads using the above algo.
Then all of these Pads will be combined using the XOR to get a final result.
Therefore an attack based on quality of Random numbers looks impossible from my standpoint.

The command itself looks like this:

```
STR.CHARS C|$$CNT|$$RES
```

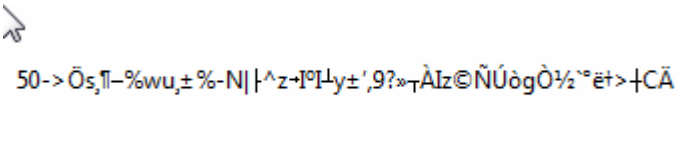
And here is a small example:

```
: $$NUM=5
STR.CHARS C|$$NUM|$$RES
MBX.$$RES
```

This is a larger example which produces a Pad of size 50.

```
: $$CNT=50
STR.CHARS C|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

It brings up this output:



50->Õs¶–%wu±%-N|├^z→Iº╚╜y±'‚9?»┬ÀIz©ÑÚògÒ½`°ë†>┼CÃ

© 2023 Theo Gottwald * https://www.smart-package.com

Use the CTF. - Copy-To-File 964 to write it to a USB-Stick or Harddisk.

Syntax

STR.CHARS D|P1[|P2]

Parameter Explanation

P1 **-** length of desired random string in bytes. If omitted, a default length of 32 byte length is used.

P2 **-** destination variable, if omitted TOS will be used for the result.

Example

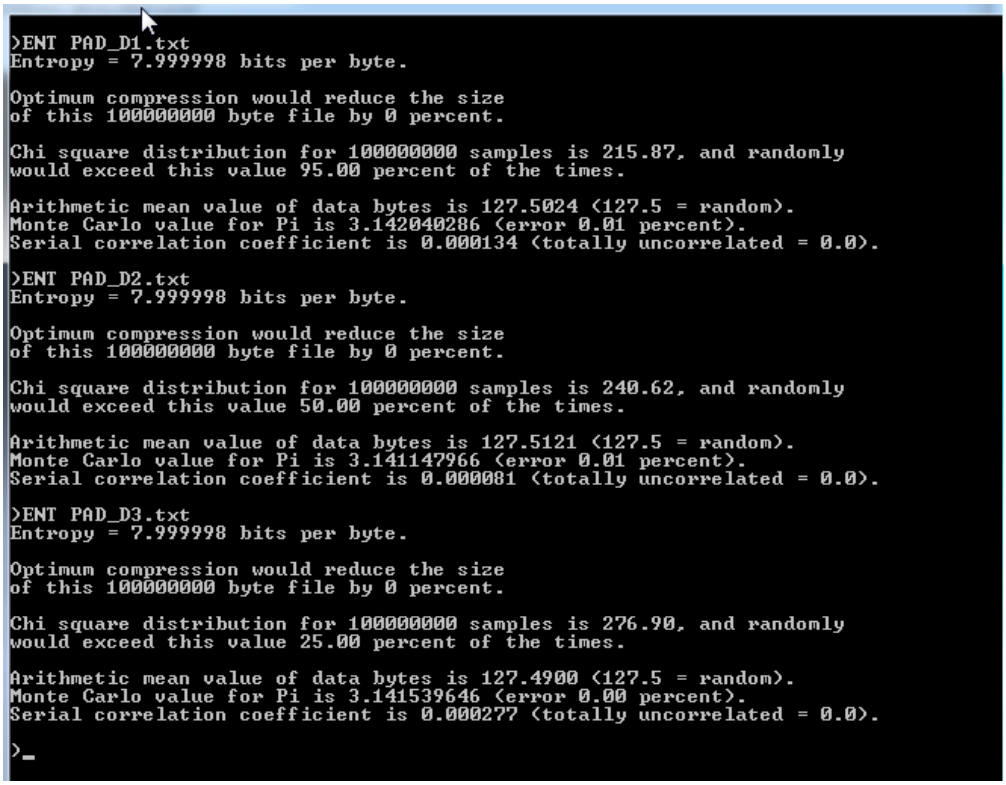
```
'************************************
' STR.CHARS D
'************************************
: $$CNT=50
STR.CHARS D|$$CNT|$$RES
LEN.$$RES|$$LEN
MBX.$$LEN->$$RES
ENR.
```

Remarks

While we have made any effort to produce PRNG's of highest quality, there is no absolute warranty whatever, that these randomly generated strings are safe for use with cryptographic or safety software, or even for password-generation on a specific computer-hardware. Therefore you have to use them on your own risk.

Below you can see the results of the so called "ENT-Test Suite" for OTP's generated with this PRNG.

© 2023 Theo Gottwald * https://www.smart-package.com



Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **STR.CHARS A** 3155
- **STR.CHARS B** 3159
- **STR.CHARS C** 3163
- CTF. - Copy-To-File 964
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- GEC. - Get Encryption 825
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.6 STR.CLONE

MiniRobotLanguage (MRL)

STR.CLONE

String commands

Intention

CLONE - Mode. This command will create a new String. by appending a specified second string for a given number of times. The result is a string, consisting of multiple copies of the second specified string. Here is an example:

```
VAR.$$NEU=Neu
STR.CLONE|$$SRC|5|$$NEU
MBX.$$SRC
ENR.
```

And this is the result:

We have created a new String that consists of 5 times the given second string. In the same way, you can make a string that is filled with SPACES or zero'es.

```
VAR.$$NEU=$nul$
STR.CLONE|$$SRC|3|$$NEU
MBX.$$SRC
ENR.
```

or

```
VAR.$$NEU=$sp$
STR.CLONE|$$SRC|3|$$NEU
MBX.<$$SRC>
ENR.
```

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

STR.CLONE|P1|P2|P3

Parameter Explanation

P1 **-** destination variable

P2 **-** numeric parameter, will be completely resolved

P3 **-** Clone-String

Example

```
'************************************
' STR.CLONE
'************************************
VAR.$$NEU=Neu
STR.CLONE|$$SRC|5|$$NEU
MBX.$$SRC
ENR.
ENR.
```

Remarks

Variables inside the source are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data.

To Replace bytes in a String, use the RPL. - RePLace in String 3143 command.
To extract a part out of a String, use the GSS. - GetSplitString 3137 command.
If your String is a delimited String (like a .csv-file), use:

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531

© 2023 Theo Gottwald * https://www.smart-package.com

- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- GFS. - Get-Filtered-String 3122
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.7 STR.DELETE

MiniRobotLanguage (MRL)

STR.DELETE

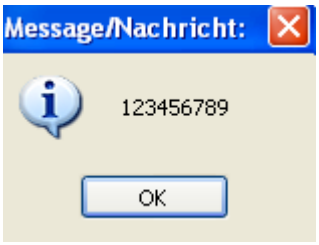
String commands

Intention

DELETE - Mode: Delete bytes/chars from inside the string. In this example, we are going to delete the Letters "AB" out of this string.

```
VAR.$$SRC=12345AB6789
STR.DELETE|$$SRC|6|7
MBX.$$SRC
ENR.
```

The result is as expected:



The string is deleted from Byte 6 to Byte 7.

Syntax

STR.DELETE|P1|P2|P3

Parameter Explanation

P1 **- Source and destination** variable

P2 **-** numeric parameter, will be completely resolved.
Start Position, first byte that is been deleted.

P3 **-** numeric parameter, will be completely resolved. End position.
Last byte that is been deleted.

if any of the numerical parameters evaluates to 0, nothing is been deleted.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************
' STR.
'************************************
VAR.$$SRC=12345AB6789
STR.DELETE|$$SRC|6|7
MBX.$$SRC
ENR.
```

Remarks

Variables inside the source **P1** are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data..

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- GFS. - Get-Filtered-String 3122
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.8 STR.FIND

MiniRobotLanguage (MRL)

STR. FIND - Mode

String commands

Intention

FIND - Mode: Search for a String inside a String. The Result will be placed on TOS 104
.
The Source-String is NOT changed in this Mode.

The result is 0 if the find-string is not found.
The result will be a number between zero and the length of the source-string if the find-string is found in the source string. Here is an example:

```
VAR.$$SRC=12345AB6789
VAR.$$FND=AB
STR.FIND|$$SRC|1|$$FND
VAR.$$POS=$tos$
DBP.Found at: $$POS
ENR.
```

We get the expected result:

```
[13] Found at: 6
```

FIND will always return the first occurrence after the given starting position or zero if nothing was found. If the starting position is zero, starting position 1 is used.

if the starting position is negative, for example -1, special rules apply.

- -1 - start search at the end of the string
- -2 - start search at the second byte from the end of the string
- and so on. Note that in any case, the result is the absolute position of a found item or zero.

Here is an example:

```
VAR.$$SRC=12345AB67AB89
VAR.$$FND=AB
STR.FIND|$$SRC|-1|$$FND
VAR.$$POS=$tos$
DBP.Found at: $$POS
ENR.
```

The result is the second "AB" in the Source-String, starting at byte-position 10.

```
[13] Found at: 10
```


© 2023 Theo Gottwald * https://www.smart-package.com

FIND is case-sensitive. If you do not want that, use `VTU.` on both the source and the Find-String.

To Replace bytes in a String, use the RPL. - RePLace in String 3143 command.
To extract a part out of a String, use the GSS. - GetSplitString 3137 command.
If your String is a delimited String (like a .csv-file), use:

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

Syntax

STR.FIND|P1|P2|P3

Parameter Explanation

P1 **-** Source variable

P2 **-** numeric parameter, will be completely resolved. Starting position for search, can be negative to search from the end of the string (reverse direction). Binary compatible variable resolution.

P3 **-** String to search for. Binary compatible variable resolution.

Example

```
'************************************
' STR.FIND
'************************************
VAR.$$SRC=12345AB6789
VAR.$$FND=AB
STR.FIND|$$SRC|1|$$FND
VAR.$$POS=$tos$
DBP.Found at: $$POS
ENR.

'************************************
' STR.FIND, search from end of string
'************************************
VAR.$$SRC=12345AB67AB89
VAR.$$FND=AB
STR.FIND|$$SRC|-1|$$FND
VAR.$$POS=$tos$
DBP.Found at: $$POS
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Variables inside the source **P1** are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data.

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.9 STR.FIND ANY

MiniRobotLanguage (MRL)

STR.fay
STR.Find Any

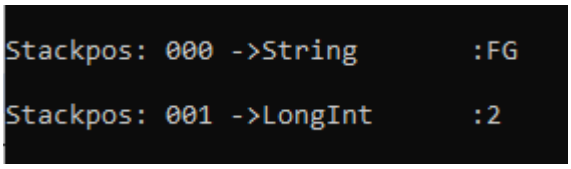
General Multi-Instring Command, will search for multiple Terms in one run.

Intention

This command is used to test if one of multiple Terms is inside a given String.
It will return the found Term and the Number of the term.
It will not return where the term was found.
It will only return the first found term.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdefg>$crlf$
$$SEA="XBC","FG"
$$POS=1
$$NUM=3
STR.fay|$$TXT|$$SEA|$$POS|p
DMP.6
MBX.!
ENR.
```

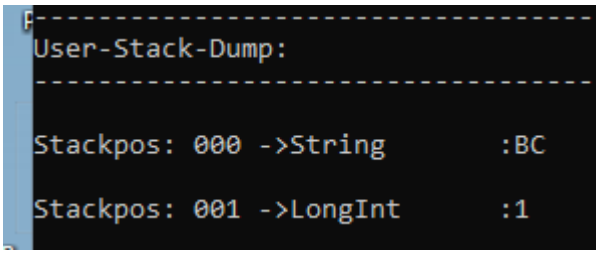
Result:



The Term FG has been found, which is the second Term.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdefg>$crlf$
$$SEA="BC","FG"
$$POS=1
$$NUM=3
STR.fay|$$TXT|$$SEA|$$POS|p
DMP.6
MBX.!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



The Term BC has been found, which is the First Term.

Syntax

STR.fay|P1|P2|P1|P2
STR.Find Any|P1|P2|P1|P2

Parameter Explanation

P1 - **(Mainstring,Text)** Variable with Text (Mainstring).
P2 - **(Input,Text)** Variable with the Terms that are searched. All Terms must be delimited using **" "** and separated using **,**
Example: `$$SEA="Term 1","/A","/D"`
P3 - **(Input,Numeric)** Variable with Starting-Position from where the Search will start.
P4 - **(Input,Character)** Optional Search Type:
p - default search "perfect Match"
e - Equalcase Search
w - Standard Search Pattern Search

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_after_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.1 STR.FROM BASE64 URL

MiniRobotLanguage (MRL)

STR.FROM BASE64 URL

String commands

Intention

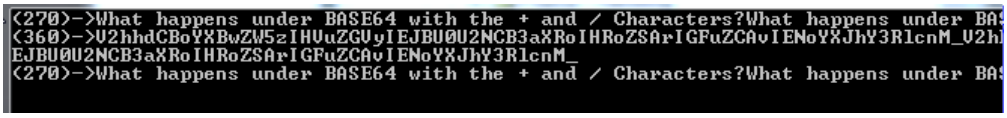
To transfer binary data in written form, all "non-printable characters" like ASCII 12,13 and others need to be encoded in printable Form.
For this generally the MIME-Format can be used. The MIME-Format is also called "BASE 64" Format. There is a special case where we do not want to have "+" and "/" signs in the result. This is the so called "BASE64 URL-Format".

`STR.FROM BASE64 URL` will convert the MIME/BASE64 URL data that is contained in a variable, back to the original-Format. The result will be in the same variable or in a Target variable if given.

The command itself looks like this:

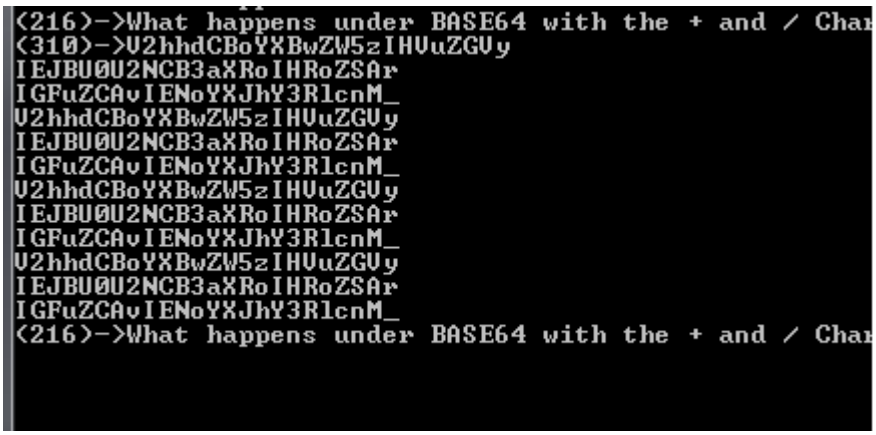
```
' $$SRC - Binary or Text-Data to convert to MIME
' $$TAR - MIME-Format Result
STR.FROM BASE64 URL|$$SRC|$$TAR
```

The Result looks somehow like you see below.
Its a Screenshot from the Examples below (see Examples-Section).



As you can see the MIME encoded Binary is 360 bytes long, while the original size was 270 bytes. That is generally 134% the size of the original size.

If the MIME-Text contains linefeeds ($crlf$), these will be ignored.



 © 2023 Theo Gottwald * https://www.smart-package.com

This is the revers command to STR.TO BASE64 URL 3219.

Syntax

STR.FROM BASE64 URL|P1[|P2]

Parameter Explanation

P1 - source variable. If **P2** is missing then its also the destination variable

P2 - ***(optional)*** destination variable. If omitted the result is placed in the variable in **P1**.

Example

```
'************************************
' STR.TO BASE64 URL and
' STR.FROM BASE64 URL
' using one Input-Output-Variable
'************************************
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO BASE64 URL|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.FROM BASE64 URL|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
MBX.It works!


$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO BASE64 URL|$$SRC||24
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.FROM BASE64 URL|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC

MBX.It works!
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- STR.TO BASE64 URL 3219
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- STR.GENERATE - Mode 3188
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- VAR. - Variable Set Value/Clear 3562
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.1 STR.FROM HEXASCI

MiniRobotLanguage (MRL)

STR.FROM HEXASCI

String commands

Intention

To transfer binary data in written form, all "non-printable characters" like ASCII 12,13 and others need to be encoded in printable Form.
HEXASCII is a Format, which encodes ASC-Codes with their ASCII-value in Hexadecimal Format.

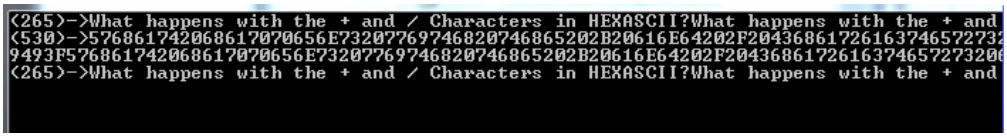
```
A -> 1E
B -> 30
```

etc.
`STR.FROM HEXASCI` will convert the HEXASCII-text that is contained in a variable, to the Original-Format. The result will be in the same variable or in a Target variable if given.

The command itself looks like this:

```
' $$SRC - Binary or Text-Data to convert to MIME
' $$TAR - MIME-Format Result
STR.FROM HEXASCI|$$SRC|$$TAR
```

The Result looks somehow like you see below.
Its a Screenshot from the Examples below (see Examples-Section).



As you can see the HEXASCII encoded Binary is 530 bytes long, while the original size was 265 bytes. That is generally 200% the size of the original size.
Therefore its less efficient then MIME Encoding,

This is the reverse command to STR.TO HEXASCI 3222 .

Syntax

STR.TO HEXASCI|P1[|P2]

Parameter Explanation

P1 - source variable. If **P2** is missing then its also the destination variable

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - (optional) destination variable. If omitted the result is placed in the variable in **P1**.

Example

```
'************************************
' STR.TO HEXASCI and
' STR.FROM HEXASCI
'************************************
$$SRC=What happens with the + and / Characters in HEXASCII?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO HEXASCI|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.FROM HEXASCI|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
MBX.It works!
ENR.

'************************************
' STR.TO HEXASCI and
' STR.FROM HEXASCI
'************************************
$$SRC=What happens with the + and / Characters in HEXASCII?
$$SRC=$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO HEXASCI|$$SRC|$$TAR
LEN.$$TAR|$$LEN
PRT.($$LEN)->$$TAR
STR.FROM HEXASCI|$$TAR|$$ERG
LEN.$$ERG|$$LEN
PRT.($$LEN)->$$ERG
MBX.It works!
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- 1.5.2 Working with Text-Strings 110
- STR.TO HEXASCI 3222
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- STR.GENERATE - Mode 3188
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- VAR. - Variable Set Value/Clear 3562
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.1 STR.GENERATE

MiniRobotLanguage (MRL)

STR.GENERATE RANDOM

String commands

Intention

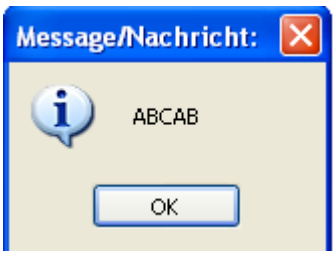
GENERATE RANDOM STRING - Mode: This Mode will generate a random String of specified length and from specified characters. You can use it to generate passwords.
The sequence lenght is about 2^32 for this PRNG.
For true cryptpgraphical random numbers and "One Time Pads" use other options, for example the `CHAR D - Option`.

The command itself looks like this:
`STR.GENERATE|$$SRC|$$CNT|$$CHR`

And here is a small example:

```
: $$TXT=ABC
: $$NUM=5
STR.GENERATE|$$SRC|$$NUM|$$TXT
MBX.$$SRC
STR.GENERATE|$$SRC
MBX.$$SRC
ENR.
```

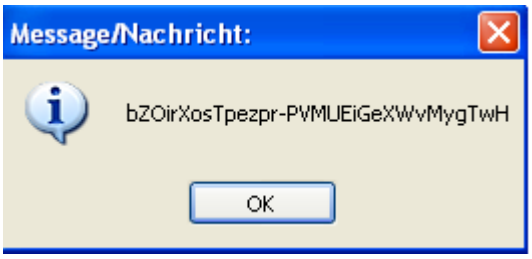
Which will bring up a Message-Box like this ..



Pic.1 Result from:
`STR.GENERATE|$$SRC|$$NUM|$$TXT`

and a Message-Box like this:

© 2023 Theo Gottwald * https://www.smart-package.com



Pic.2 Result from:
`STR.GENERATE|$$SRC`

Syntax

STR.GENERATE|P1[|P2][|P3]

Parameter Explanation

P1 **-** source and destination variable

P2 **-** ***(optional)*** length of desired random string in bytes. If omitted, a default length of 32 byte length is used.

P3 **-** ***(optional)*** variable with all characters that should be used for the random string generation.
Alternatively you can use some small letters to define the character-Set for **P3**.

- empty/if **P3** is omitted, then the whole charset from 255 ASC-Characters will be used.

"a" - all ASC-Character (0-255)
"b" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-?#+*"
"c" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-?"
"d" - "abcdefghijklmnopqrstuvwxyzöüäßABCDEFGHIJKLMNOPQRSTUVWXYZÄÜÖ"
"e" - "abcdefghijklmnopqrstuvwxyz"
"f" - "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
"g" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß'[]{}@|<>.1234567890_\-;:*+#§%&/()=! "
"h" - "0123456789ABCDEF"
"i" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß'´`[]{}@.1234567890_-;+#§%&()=! "
"k" - ":\abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß'[]{}@.1234567890_-;+#§%&()=! "

© 2023 Theo Gottwald * https://www.smart-package.com

```
"m" -
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ.12345678
90_\ "
"o" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ-"
"n" - "0123456789"
"p" - "\/:*?<>|"+ASC(34) (Double-Quotes)
"x" -
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß[]
{}@<>.1234567890_\-;:*+#§%&/()=!"
```

Example

```
'************************************
' STR.GENERATE
'************************************
: $$TXT=ABC
: $$NUM=5
STR.GENERATE|$$SRC|$$NUM|$$TXT
MBX.$$SRC
STR.GENERATE|$$SRC
MBX.$$SRC
ENR.
```

Remarks

There is no warranty whatever, that these randomly generated strings are safe for use with cryptographic or safety software, or even for passwords. Use them on your own risk.

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- GEC. - Get Encryption 825
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122

© 2023 Theo Gottwald * https://www.smart-package.com

- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.13 STR.INSERT

MiniRobotLanguage (MRL)

STR.INSERT

String commands

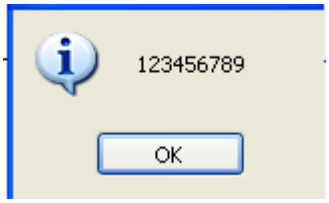
Intention

INSERT - Mode: This Mode will insert a String at a specified place into the Source String.
This command can change the length of a string.

Used on a large number of large strings, it will be slower then the `STR.OVERWRITE` command.

```
' Source and Target string
VAR.$$SRC=1234589
' String to insert
VAR.$$NEW=67
' Position where string will be inserted
VAR.$$POS=6
' This is where the magic happens
STR.INSERT|$$SRC|$$POS|$$NEW
' Here we get a message-box
MBX.$$SRC
ENR.
```

The result is:



To Replace bytes in a String, use the RPL. - RePLace in String 3143 command.
To extract a part out of a String, use the GSS. - GetSplitString 3137 command.
If your String is a delimited String (like a .csv-file), use:

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

Syntax

STR.INSERT|P1|P2|P3

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** ***source and*** destination variable

P2 **-** numeric parameter, will be completely resolved.
Insert-Position is before the given byte-position.
The inserted string will start at that position.
If P2 is 1 or smaller then one, P3 will be appended
before the start of the source-string. if P2 is larger
then the length of the source-string, P3 will be appended
to the end of the string.

P3 **-** ***INSERT***-String. This string will be inserted into
the source string.

Example

```
'************************************
' STR.INSERT/FIND
'************************************
VAR.$$SRC=1234589
VAR.$$FND=5
VAR.$$NEW=67
STR.FIND|$$SRC|1|$$FND
VAR.$$POS=$tos$
VIC.$$POS
STR.INSERT|$$SRC|$$POS|$$NEW
MBX.$$SRC
ENR.

'************************************
' STR.INSERT
'************************************
VAR.$$SRC=1234589
VAR.$$NEW=67
VAR.$$POS=6
STR.INSERT|$$SRC|$$POS|$$NEW
MBX.$$SRC
ENR.
```

Remarks

Variables inside the source **P1** are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data.

Limitations:

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- GFS. - Get-Filtered-String 3122
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.14 STR.MAKE PAD

MiniRobotLanguage (MRL)

STR.Make Pad

Generate a highly random number

Intention

MAKE PAD has the same Syntax like GENERATE RANDOM STRING - but uses internally another PRNG (Pseudo Random Number Generator) with a Periode length of about 2^60 which is in numbers (rough estimation) 1152921504000000000 before the sequence will repeat. Therefore the result should be safe for cryptographic uses.

This Mode will generate a random String of specified length and from specified characters. You can use it to generate passwords.

For generation of Tables with random numbers in the full ASC-Range), cryptographical random numbers and "One Time Pads" prefer other options, for example the `CHAR D - Option`.

The command itself looks like this:

```
STR.MAKE PAD|$$SRC|$$CNT|$$CHR
```

And here is a small example:

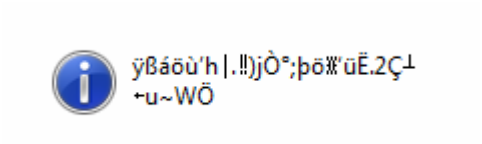
```
: $$TXT=ABC
: $$NUM=5
STR.MAKE PAD|$$SRC|$$NUM|$$TXT
MBX.$$SRC
```

Which will bring up a Message-Box like this ..

Pic.1 Result from:

```
STR.MAKE PAD|$$SRC|$$NUM|$$TXT
```

and a Message-Box like this:



© 2023 Theo Gottwald * https://www.smart-package.com

Pic.2 Result from:
STR.MAKE PAD|$$SRC

Syntax

STR.MAKE PAD|P1[|P2][|P3]

Parameter Explanation

P1 **-** source and destination variable

P2 **-** ***(optional)*** length of desired random string in bytes. If omitted, a default length of 32 byte length is used.

P3 **-** ***(optional)*** variable with all characters that should be used for the random string generation.
Alternatively you can use some small letters to define the character-Set for **P3**.

- empty/if **P3** is omitted, then the whole charset from 255 ASC-Characters will be used.

"a" - all ASC-Character (0-255)
"b" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-?#+*"
"c" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-?"
"d" - "abcdefghijklmnopqrstuvwxyzöüäßABCDEFGHIJKLMNOPQRSTUVWXYZÄÜÖ"
"e" - "abcdefghijklmnopqrstuvwxyz"
"f" - "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
"g" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß'[]{}@|<>.1234567890_\-;:*+#§%&/()=! "
"h" - "0123456789ABCDEF"
"i" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß'´`[]{}@.1234567890_-;+#§%&()=! "
"k" - ":\abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß'[]{}@.1234567890_-;+#§%&()=! "
"m" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ.1234567890_\ "
"o" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ-"
"n" - "0123456789"
"p" - "\/:*?<>|"+ASC(34) (Double-Quotes)
"x" - "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZüöäÜÖÄß[]{}@<>.1234567890_\-;:*+#§%&/()=!"

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'************************************
' STR.MAKE PAD
'************************************
: $$TXT=ABC
: $$NUM=5
STR.MAKE PAD|$$SRC|$$NUM|$$TXT
MBX.$$SRC
STR.MAKE PAD|$$SRC
MBX.$$SRC
ENR.
```

Remarks

While we have made any effort to produce PRNG's of highest quality, there is no warranty whatever, that these randomly generated strings are safe for use with cryptographic or safety software, or even for password-generation on a specific computer-hardware. Use them on your own risk.

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- GEC. - Get Encryption 825
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.1 STR.MIME TO TEXT

MiniRobotLanguage (MRL)

STR.MIME TO TEXT

String commands

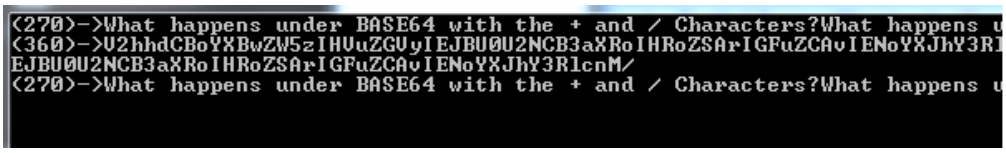
Intention

To transfer binary data in written form, all "non-printable characters" like ASCII 12,13 and others need to be encoded in printable Form.
For this generally the MIME-Format can be used. The MIME-Format is also called "BASE 64" Format.
`STR.MIME TO TEXT` will convert the MIME/BASE64 data that is contained in a variable, back to the original-Format. The result will be in the same variable or in a Target variable if given.

The command itself looks like this:

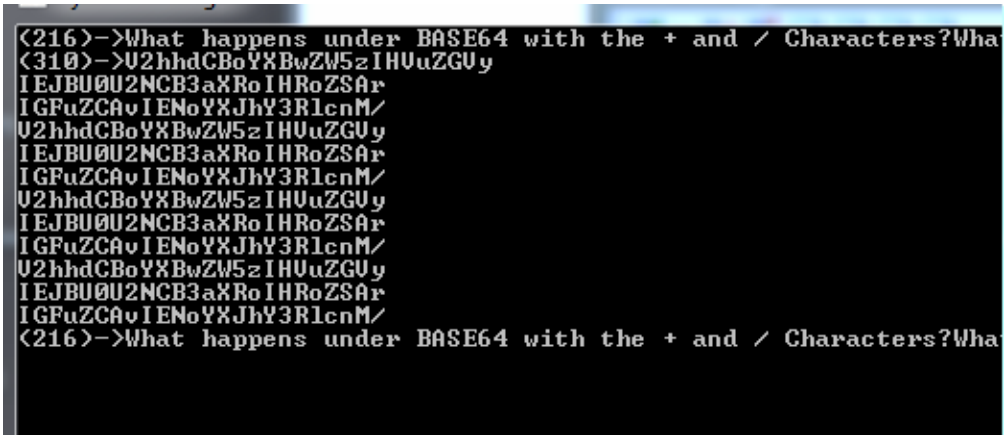
```
' $$SRC - Binary or Text-Data to convert to MIME
' $$TAR - MIME-Format Result
STR.MIME TO TEXT|$$SRC|$$TAR
```

The Result looks somehow like you see below.
Its a Screenshot from the Examples below (see Examples-Section).



As you can see the MIME encoded Binary is 360 bytes long, while the original size was 270 bytes. That is generally 134% the size of the original size.

If the MIME-Text contains linefeeds ($crlf$), these will be ignored.



 © 2023 Theo Gottwald * https://www.smart-package.com

This is the revers command to STR.TEXT TO MIME3215.

Syntax

STR.MIME TO TEXT|P1[|P2]

Parameter Explanation

P1 **-** source variable. If **P2** is missing then its also the destination variable

P2 **-** ***(optional)*** destination variable. If omitted the result is placed in the variable in **P1**.

Example

```
'************************************
' STR.MIME TO TEXT and
' STR.TEXT TO MIME
' using one Input-Output-Variable
'************************************
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TEXT TO MIME|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.MIME TO TEXT|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC

$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TEXT TO MIME|$$SRC||64
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.MIME TO TEXT|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC

MBX.It works!
ENR.

'************************************
' STR.MIME TO TEXT and
' STR.TEXT TO MIME
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' with a 40 Characters per Line Output
'************************************
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
STR.TEXT TO MIME|$$SRC|$$ERG|40
LEN.$$ERG|$$LEN
PRT.($$LEN)->$$ERG
$$SRC=$$ERG
STR.MIME TO TEXT|$$SRC|$$ERG
LEN.$$ERG|$$LEN
PRT.($$LEN)->$$ERG
MBX.It works!
ENR.


'************************************
' STR.MIME TO TEXT and
' STR.TEXT TO MIME
' using one Variable only
' with a 40 Characters per Line Output
'************************************
'
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TEXT TO MIME|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.MIME TO TEXT|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
MBX.It works!
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- STR.TEXT TO MIME 3215
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182

© 2023 Theo Gottwald * https://www.smart-package.com

- Variables with Inline Data 3516
- STR.GENERATE - Mode 3188
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.79.21.1 STR.NINSTR

MiniRobotLanguage (MRL)

STR.nis
STR.NInstr

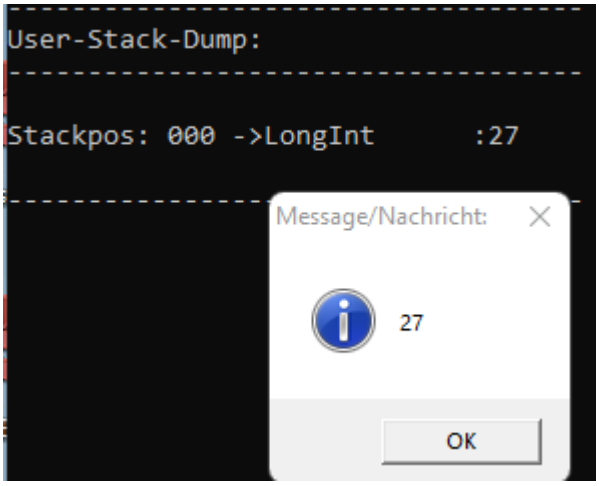
Instring Command, will search for the n'th search-Terms in one run.

Intention

This command is used to find the n'th match of a search Termin in Mainstring.
It will return the Position of the found Term.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdEFg>$crlf$
$$SEA=EF
$$POS=1
$$NUM=2
STR.nis|$$TXT|$$SEA|$$NUM|$$POS|$$RES
DMP.6
MBX.$$RES
ENR.
```

Result:



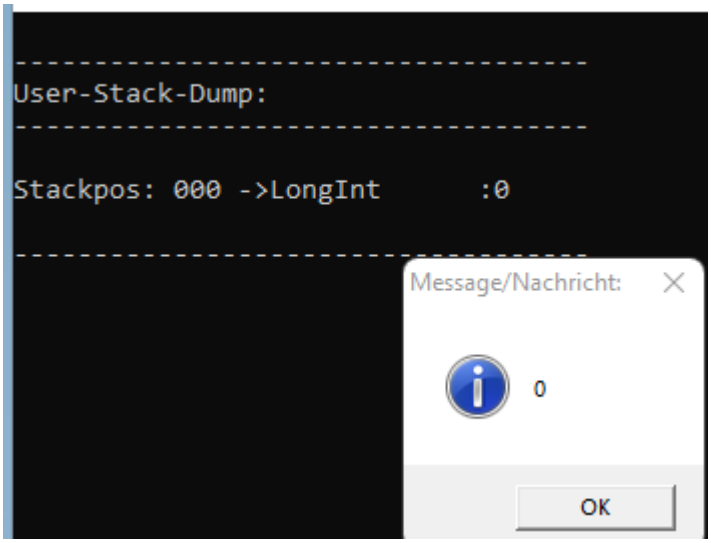
The Text was found at Position 27.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdEFg>$crlf$
$$SEA=ef
$$POS=1
$$NUM=2
STR.nis|$$TXT|$$SEA|$$NUM|$$POS|$$RES
DMP.6
MBX.$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
```

Result:



The Text "ef" was not found.

Syntax

STR.nis|P1|P2|P3[|P4][|P5]
STR.NInstr|P1|P2|P3[|P4][|P5]

Parameter Explanation

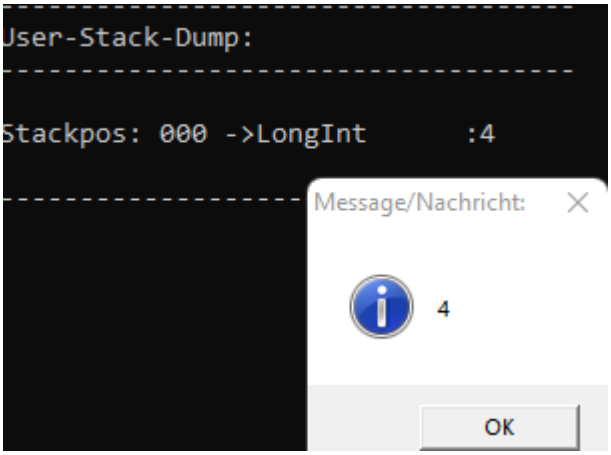
P1 - **(Mainstring,Text)** Variable with Text (Mainstring).
P2 - **(Input,Text)** Variable with the Text to search for.
P3 - **(Input,Numeric)** Variable with Number of the wanted appearance of the search term.
P4 - **(Input,Numeric)** Variable with Starting-Position from where the Search will start.
P5 - **(Output,Numeric)** Variable that will contain the result. The result will also be put on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: STR.nis
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdEFg>$crlf$
$$SEA=CD
$$POS=1
$$NUM=1
STR.nis|$$TXT|$$SEA|$$NUM|$$POS|$$RES
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
DMP.6
MBX.$$RES
ENR.
```



Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.1 STR.OVERWRITE

MiniRobotLanguage (MRL)

STR.OVERWRITE

String commands

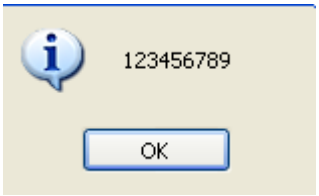
Intention

OVERWRITE - Mode: This Mode will patch the Source String with a second Patch String. It will "overwrite" the Source String, while it will never change the total length of the Source String.

Here is an example:

```
VAR.$$SRC=12345XX89
VAR.$$NEW=67
VAR.$$POS=6
STR.OVERW|$$SRC|$$POS|$$NEW
MBX.$$SRC
ENR.
```

We want to OVERWRITE the two "XX" in the Source String with the missing numbers "67".
The result is just that. In real life, you can use this command for example to patch binary files. OVERWRITE will take care that the overall length of the file is never changed, and that no content in the file is been moved. It will just "overwrite" some content with your content.

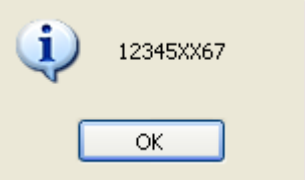


You can specify negative numbers as positional parameter. In this case, the position will be calculated from the end of the source string:

```
VAR.$$SRC=12345XX89
VAR.$$NEW=67
' - 2 => Second byte from end of the string
VAR.$$POS=-2
STR.OVERW|$$SRC|$$POS|$$NEW
MBX.$$SRC
ENR.
```

Like in this example.

© 2023 Theo Gottwald * https://www.smart-package.com



To Replace bytes in a String, use the RPL. - RePLace in String 3143 command.
To extract a part out of a String, use the GSS. - GetSplitString 3137 command.
If your String is a delimited String (like a .csv-file), use:

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

Syntax

STR.OVERWRITE|P2|P3

Parameter Explanation

P1 **-** source and destination variable

P2 **-** numeric parameter, will be completely resolved. This is the start-byte. Its the first byte that will be overwritten with the target string **P3**. You can specify a negative value for P2 then the position will be calculated from the end of the string (reverse direction).

P3 **-** Overwrite-String, must have a length larger then zero.

Example

```
'************************************
' STR.OVERWRITE
'************************************
VAR.$$SRC=12345XX89
VAR.$$NEW=67
VAR.$$POS=6
STR.OVERW|$$SRC|$$POS|$$NEW
MBX.$$SRC
ENR.

'************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' STR.OVERWRITE, negative position
'************************************
VAR.$$SRC=12345XX89
VAR.$$NEW=67
VAR.$$POS=-2
STR.OVERW|$$SRC|$$POS|$$NEW
MBX.$$SRC
ENR.
```

Remarks

Variables inside the source **P1** are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data..

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.11 STR.PATCH

MiniRobotLanguage (MRL)

STR.PATCH

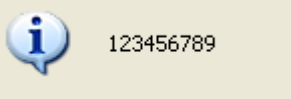
String commands

Intention

PATCH - Mode: This Mode combines the DELETE 3174 and the INSERT Mode 3192 into one command. Unlike the OVERWRITE-Mode 3205, this Mode can change the length of a String if needed. Here is a simple example:

```
VAR.$$SRC=12345XX89
' These characters will be inserted
VAR.$$NEW=67
' Starting position is byte 6
VAR.$$POS=6
' DELETE 2 characters
VAR.$$DEL=2
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.
```

The result is as expected:



You can also use a negative P2, in that case the position will be calculated from the end of the string (reverse direction). Like this:

```
VAR.$$SRC=12345XX89
VAR.$$NEW=67
VAR.$$POS=-4
VAR.$$DEL=2
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.
```

The **result is the same like before**, as the position "-4" is the same character then the position "6", if the total string length is 9.

You can use the `STR.PATCH` command also as a replacement for the `STR.DELETE` command, if you do specify an empty Insert-parameter **P4**. That works, because if **P4** is empty then the INSERT-Part will be skipped and only the DELETE will be made. Like this:

```
VAR.$$SRC=12345XX89
```


© 2023 Theo Gottwald * https://www.smart-package.com

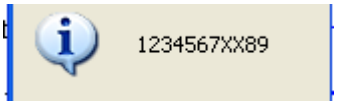
```
VAR.$$NEW=
VAR.$$POS=-4
VAR.$$DEL=2
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.
```

The result is this:

In the same way, you can have the DELETE-Part skipped, if you specify a zero for **P3**. This is the number of characters to be deleted. Of course if **P3** is zero, nothing is going to be deleted. Instead the complete DELETE part is being skipped. In that case the INSERT Part will be executed, if **P4** is not empty. Here is an example:

```
VAR.$$SRC=12345XX89
VAR.$$NEW=67
VAR.$$POS=-4
VAR.$$DEL=0
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.
```

The result is just the INSERTed string, and nothing deleted.



Using the `STR.PATCH` command on many large strings (>100 MB) may be significantly slower then using the `STR.OVERWRITE` command. Therefore prefer the `STR.OVERWRITE`, if the String that is inserted has the same length as the string that is been deleted.

Syntax

STR.PATCH|P1|P2|P3|P4

Parameter Explanation

P1 - source and destination variable

P2 - numeric parameter, will be completely resolved.
Starting position of the DELETE/INSERT operations.

P3 - numeric parameter, will be completely resolved. Number of bytes to delete, starting at the position **P2**. If **P3** evaluates to zero, nothing is deleted. Then only the

© 2023 Theo Gottwald * https://www.smart-package.com

INSERT is been done.

P4 **-** Insert-String, String that is inserted. If Length of **P4** evaluates to zero, nothing will be inserted. In this case the PATCH command will work somehow like DELETE (**P2** is count here, while it is a position in DELETE).

Example

```
'************************************
' STR.PATCH
'************************************
VAR.$$SRC=12345XX89
' These characters will be inserted
VAR.$$NEW=67
' Starting position is byte 6
VAR.$$POS=6
' DELETE 2 characters
VAR.$$DEL=2
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.


'************************************
' STR.PATCH (INSERT MODE)
'************************************
VAR.$$SRC=12345XX89
VAR.$$NEW=67
VAR.$$POS=-4
VAR.$$DEL=0
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.


'************************************
' STR.PATCH (DELETE MODE)
'************************************
VAR.$$SRC=12345XX89
VAR.$$NEW=
VAR.$$POS=-4
VAR.$$DEL=2
STR.PATCH|$$SRC|$$POS|$$DEL|$$NEW
MBX.$$SRC
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Variables inside the source **P1** and **P4** are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data..

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.79.21.1 STR.REVERSE

MiniRobotLanguage (MRL)

STR.REVERSE

String commands

Intention

REVERSE - Mode: This will reverse all bytes in a String. The last byte will be the first byte.

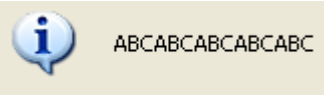
The command itself looks like this:

```
STR.REVERSE|$$SRC
```

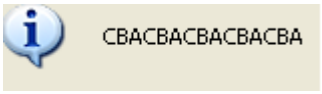
And here is a small example:

```
: $$TXT=ABC
: $$NUM=5
STR.CLONE|$$SRC|$$NUM|$$TXT
MBX.$$SRC
STR.REVERSE|$$SRC
MBX.$$SRC
ENR.
```

Which will bring up a Message-Box before the reversing:



and a Message-Box after the string was reversed:



Looking at these clearly shows what we have been doing here.

To Replace bytes in a String, use the RPL. - RePLace in String 3143 command.
To extract a part out of a String, use the GSS. - GetSplitString 3137 command.
If your String is a delimited String (like a .csv-file), use:

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

STR.REVERSE|P1

Parameter Explanation

P1 **-** source and destination variable

Example

```
'************************************
' STR.REVERSE
'************************************
: $$TXT=ABC
: $$NUM=5
STR.CLONE|$$SRC|$$NUM|$$TXT
MBX.$$SRC
STR.REVERSE|$$SRC
MBX.$$SRC
ENR.
```

Remarks

Variables inside the source **P1** are only resolved once. Therefore it can be safely used for binary data. Specialfolders and systemvariables etc. are not resolved, to avoid unwanted changes in binary data.

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.2 STR.TEXT TO MIME

MiniRobotLanguage (MRL)

STR.TEXT TO MIME

String commands

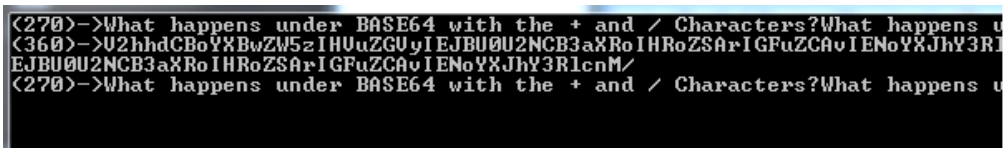
Intention

To transfer binary data in written form, all "non-printable characters" like ASCII 12,13 and others need to be encoded in printable Form.
For this generally the MIME-Format can be used. The MIME-Format is also called "BASE 64" Format.
`STR.TEXT TO MIME` will convert the text or binary data that is contained in a variable, to the MIME-Format. The result will be in the same variable or in a Target variable if given.

The command itself looks like this:

```
' $$SRC - Binary or Text-Data to convert to MIME
' $$TAR - MIME-Format Result
STR.TEXT TO MIME|$$SRC|$$TAR
```

The Result looks somehow like you see below.
Its a Screenshot from the Examples below (see Examples-Section).



As you can see the MIME encoded Binary is 360 bytes long, while the original size was 270 bytes. That is generally 134% the size of the original size.

There is an additional third parameter, that allows for a maximum line-length, after which a `$crlf$` will be inserted (newline). If this parameter is zero, no `$crlf$` will be inserted. Otherwise, it may look like this:

```
STR.TEXT TO MIME|$$SRC|$$TAR|24
```

or

```
STR.TEXT TO MIME|$$SRC||24
```

The Result now contains a linefeed ($crlf$) after each line with the given number of characters.

© 2023 Theo Gottwald * https://www.smart-package.com

Now the Ratio is 144% because of the additional Newlines that have been added. They will be ignored when decoding MIME back to the original Text or Binary.

This is the reverse command to STR.MIME TO TEXT 3198.

Syntax

STR.TEXT TO MIME|P1[|P2][|P3]

Parameter Explanation

P1 **-** source variable. If **P2** is missing then its also the destination variable

P2 **-** ***(optional)*** destination variable. If omitted the result is placed in the variable in **P1**.

P3 **-** ***(optional)*** numerical parameter. If given, a $crlf$ (Newline) is placed after the given number of characters.

Example

```
'************************************
' STR.MIME TO TEXT and
' STR.TEXT TO MIME
' using one Input-Output-Variable
'************************************
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TEXT TO MIME|$$SRC
LEN.$$SRC|$$LEN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.($$LEN)->$$SRC
STR.MIME TO TEXT|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC

$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TEXT TO MIME|$$SRC||64
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.MIME TO TEXT|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC

MBX.It works!
ENR.

'************************************
' STR.MIME TO TEXT and
' STR.TEXT TO MIME
' with a 40 Characters per Line Output
'************************************
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
STR.TEXT TO MIME|$$SRC|$$ERG|40
LEN.$$ERG|$$LEN
PRT.($$LEN)->$$ERG
$$SRC=$$ERG
STR.MIME TO TEXT|$$SRC|$$ERG
LEN.$$ERG|$$LEN
PRT.($$LEN)->$$ERG
MBX.It works!
ENR.


'************************************
' STR.MIME TO TEXT and
' STR.TEXT TO MIME
' using one Variable only
' with a 40 Characters per Line Output
'************************************
'
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$$SRC$
$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TEXT TO MIME|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.MIME TO TEXT|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.It works!
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- Variables with Inline Data 3516
- STR.MIME TO TEXT 3198
- **STR.TO BASE64 URL** 3219
- **STR.FROM BASE64 URL** 3182
- STR.GENERATE - Mode 3188
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.2 STR.TO BASE64 URL

MiniRobotLanguage (MRL)

STR.TO BASE64 URL

String commands

Intention

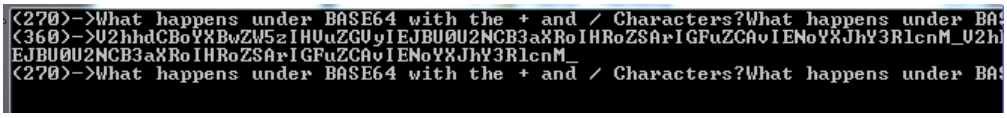
To transfer binary data in written form, all "non-printable characters" like ASCII 12,13 and others need to be encoded in printable Form.
For this generally the MIME-Format can be used. The MIME-Format is also called "BASE 64" Format. There is a special case where we do not want to have "+" and "/" signs in the result. This is the so called "BASE64 URL-Format".

`STR.TO BASE64 URL` will convert the text or binary data that is contained in a variable, to the BASE64 URL-Format. The result will be in the same variable or in a Target variable if given.

The command itself looks like this:

```
' $$SRC - Binary or Text-Data to convert to MIME
' $$TAR - MIME-Format Result
STR.TO BASE64 URL|$$SRC|$$TAR
```

The Result looks somehow like you see below.
Its a Screenshot from the Examples below (see Examples-Section).



As you can see the MIME encoded Binary is 360 bytes long, while the original size was 270 bytes. That is generally 134% the size of the original size.

There is an additional third parameter, that allows for a maximum line-length, after which a `$crlf$` will be inserted (newline). If this parameter is zero, no `$crlf$` will be inserted. Otherwise, it may look like this:

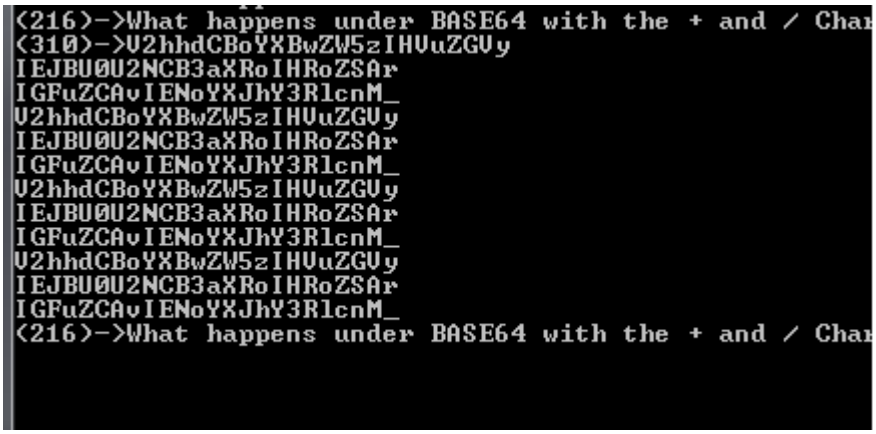
```
STR.TO BASE64 URL|$$SRC|$$TAR|24
```

or

```
STR.TO BASE64 URL|$$SRC||24
```

The Result now contains a linefeed ($crlf$) after each line with the given number of characters.

© 2023 Theo Gottwald * https://www.smart-package.com



Now the Ratio is 144% because of the additional Newlines that have been added. They will be ignored when decoding MIME back to the original Text or Binary.

This is the reverse command to STR.FROM BASE64 URL 3182.

Syntax

STR.TO BASE64 URL|P1[|P2][|P3]

Parameter Explanation

P1 **-** source variable. If **P2** is missing then its also the destination variable

P2 **-** ***(optional)*** destination variable. If omitted the result is placed in the variable in **P1**.

P3 **-** ***(optional)*** numerical parameter. If given, a $crlf$ (Newline) is placed after the given number of characters.

Example

```
'************************************
' STR.TO BASE64 URL and
' STR.FROM BASE64 URL
' using one Input-Output-Variable
'************************************
$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO BASE64 URL|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STR.FROM BASE64 URL|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
MBX.It works!


$$SRC=What happens under BASE64 with the + and / Characters?
$$SRC=$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO BASE64 URL|$$SRC||24
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.FROM BASE64 URL|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC

MBX.It works!
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- STR.GENERATE - Mode 3188
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- VAR. - Variable Set Value/Clear 3562
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.21.2 STR.TO HEXASCI

MiniRobotLanguage (MRL)

STR.TO HEXASCI

String commands

Intention

To transfer binary data in written form, all "non-printable characters" like ASCII 12,13 and others need to be encoded in printable Form.
HEXASCII is a Format, which encodes ASC-Codes with their ASCII-value in Hexadecimal Format.

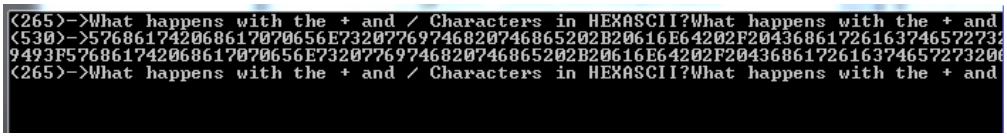
```
A -> 1E
B -> 30
```

etc.
`STR.TO HEXASCI` will convert the text or binary data that is contained in a variable, to the HEXASCII-Format. The result will be in the same variable or in a Target variable if given.

The command itself looks like this:

```
' $$SRC - Binary or Text-Data to convert to MIME
' $$TAR - MIME-Format Result
STR.TO HEXASCI|$$SRC|$$TAR
```

The Result looks somehow like you see below.
Its a Screenshot from the Examples below (see Examples-Section).



As you can see the HEXASCII encoded Binary is 530 bytes long, while the original size was 265 bytes. That is generally 200% the size of the original size.
Therefore its less efficient then MIME Encoding,

This is the reverse command to STR.FROM HEXASCI 3185.

Syntax

STR.TO HEXASCI|P1[|P2]

Parameter Explanation

P1 - source variable. If **P2** is missing then its also the destination variable

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - (optional) destination variable. If omitted the result is placed in the variable in **P1**.

Example

```
'************************************
' STR.TO HEXASCI and
' STR.FROM HEXASCI
'************************************
$$SRC=What happens with the + and / Characters in HEXASCII?
$$SRC=$$SRC$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO HEXASCI|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.FROM HEXASCI|$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
MBX.It works!
ENR.

'************************************
' STR.TO HEXASCI and
' STR.FROM HEXASCI
'************************************
$$SRC=What happens with the + and / Characters in HEXASCII?
$$SRC=$$SRC$$SRC$$SRC$$SRC
LEN.$$SRC|$$LEN
PRT.($$LEN)->$$SRC
STR.TO HEXASCI|$$SRC|$$TAR
LEN.$$TAR|$$LEN
PRT.($$LEN)->$$TAR
STR.FROM HEXASCI|$$TAR|$$ERG
LEN.$$ERG|$$LEN
PRT.($$LEN)->$$ERG
MBX.It works!
ENR.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1.5.2 Working with Text-Strings 110
- **STR.FROM HEXASCI** 3185
- STR.TO BASE64 URL 3219
- STR.FROM BASE64 URL 3182
- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- STR.GENERATE - Mode 3188
- GEC. - Get Encryption 825
- GMD. - Get Message Digest 831
- SHA. - Safer Hash Algo 840
- VAR. - Variable Set Value/Clear 3562
- RPL. - RePLace in String 3143
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- GFS. - Get-Filtered-String 3122
- GES. - Get-Extracted-String 3067
- LEN. - Length-of-String 3140

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.79.22 Trim String

3.79.22.1 VBT. - Variable-Both-sides-Trim

MiniRobotLanguage (MRL)

VBT. Command

Variable Both sides Trim

Intention

This command can be used to Trim Spaces from both sides of a variable-value.

By using the optional parameters you can trim any chars from both sides of the variable value.

Example:

```
VAR.§§LIN=  345  Hallo Marco
' We first remove the outer Spaces
VBT.§§LIN
DBP.>§§LIN<
' Result is: >345  Hallo Marco<
```

Now we use the Parameter **P2.**

```
VAR.§§LIN=  345  Hallo Marco
' $sp$ is a system-variable that is resolved to a SPACE
character
VBT.§§LIN|$sp$345
DBP.>§§LIN<
' Result is: >Hallo Marco<
```

Now we use the third mode, for this Parameter **P2 is the term "all"**

```
VAR.§§LIN=  345  Hallo Marco $crlf$
' "all" means all characters with ASCII codes below 33
VBT.§§LIN|all
DBP.>§§LIN<
' Result is: >345  Hallo Marco<
```

Syntax

VBT.[P1][|P2]

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 - ***(optional)*** Source and Target-variable, if omitted, TOS is used.

P2 - ***(optional)*** if given, ANY of the characters in **P2** will be trimmed at both sides of **P1**

If Parameter 2 is "all" the characters from ASCI(0 to 32) will be trimmed from both sides of **P1**.

Example

```
''************************************
' VBT/SBD - DEMO
'************************************
VAR.§§LIN=  345  Hallo Marco
' We first remove the outer Spaces
VBT.§§LIN
PRT.>§§LIN<
' Now we remove the outer numbers and spaces
VBT.§§LIN|123 4567890
PRT.>§§LIN<
SBD.§§LIN|$sp$|4|§§ERG
MBX.§§ERG
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VLT - Variable-Left-Trim** 3228
- **VRT. - Variable-Right-Trim** 3231
- **VTU. - Variable to Uppercase** 3059
- **VTL. - Variable to Lowercase** 3053
- **VTM. - Variable to Mediumcase** 3056
- **VRD. - Variable-Remove-Doubles** 3076
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067

© 2023 Theo Gottwald * https://www.smart-package.com

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.22.2 VLT. - Variable-Left-Trim

MiniRobotLanguage (MRL)

VLT. Command

Variable Left sides Trim

Intention

This command can be used to Trim Spaces from the left side of a variable-value.

By using the optional parameters you can trim any chars from the left side of the variable value.

Example:

```
VAR.§§LIN=  345  Hallo Marco
' We first remove the outer Spaces
VLT.§§LIN
DBP.>§§LIN<
' Result is: >345  Hallo Marco <
```

Now we use the Parameter **P2.**

```
VAR.§§LIN=  345  Hallo Marco
' $sp$ is a system-variable that is resolved to a SPACE
character
VLT.§§LIN|$sp$345
DBP.>§§LIN<
' Result is: >Hallo Marco <
```

Now we use the third mode, for this Parameter **P2 is the term "all"**

```
VAR.§§LIN= $crlf$ 345  Hallo Marco
' "all" means all characters with ASCII codes below 33
VLT.§§LIN|all
DBP.>§§LIN<
' Result is: >345  Hallo Marco<
```

Syntax

VLT.[P1][|P2]

Parameter Explanation

P1 **- *(optional)*** Source and Target-variable, if omitted,

© 2023 Theo Gottwald * https://www.smart-package.com

TOS is used.

P2 - ***(optional)*** if given, ANY of the characters in **P2** will be trimmed at the left side of **P1**

If Parameter 2 is "all" the characters from ASCI(0 to 32) will be trimmed from the left side of **P1**.

Speed in Ticks:

This command uses typically around 290 Ticks.

Example

```
''************************************
' VLT/VBT/SBD - DEMO
'************************************
'DBM.2

VAR.§§LIN=  345  Hallo Marco
' We first remove the outer Spaces
VBT.§§LIN
PRT.>§§LIN<
' Now we remove the outer numbers and spaces
VLT.§§LIN|123 4567890
PRT.>§§LIN<
SBD.§§LIN|$sp$|4|§§ERG
MBX.§§ERG
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VBT. - Variable-Both-sides-Trim** 3225
- **VRT. - Variable-Right-Trim** 3231
- **VTU. - Variable to Uppercase** 3059
- **VTL. - Variable to Lowercase** 3053
- **VTM. - Variable to Mediumcase** 3056
- **VRD. - Variable-Remove-Doubles** 3076
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.22.3 VRT. - Variable-Right-Trim

MiniRobotLanguage (MRL)

VRT. Command

Variable Right sides Trim

Intention

This command can be used to Trim Spaces from the right side of a variable-value.

By using the optional parameters you can trim any chars from the right side of the variable value.

Example:

```
VAR.$$LIN=  345  Hallo Marco
' We first remove the outer Spaces
VRT.$$LIN
DBP.>$$LIN<
' Result is: >345  Hallo Marco<
```

Now we use the Parameter **P2.**

```
VAR.$$LIN=   Hallo Marco   345
' $sp$ is a system-variable that is resolved to a SPACE
character
VRT.$$LIN|$sp$345
DBP.>$$LIN<
' Result is: >Hallo Marco   345<
```

Now we use the third mode, for this Parameter **P2 is the term "all"**

```
VAR.$$LIN=345  Hallo Marco $crlf$
' "all" means all characters with ASCII codes below 33
VRT.$$LIN|all
DBP.>$$LIN<
' Result is: >345  Hallo Marco<
```

Syntax

VRT.[P1][|P2]

Parameter Explanation

P1 - ***(optional)*** Source and Target-variable, if omitted, TOS is used.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **-** ***(optional)*** if given, ANY of the characters in **P2** will be trimmed from the right side of **P1**

If Parameter 2 is "all" the characters from ASCI(0 to 32) will be trimmed from the right side of **P1.**

Example

```
''************************************
' VRT/VBT/SBD - DEMO
'************************************
'DBM.2

VAR.§§LIN=  Hallo Marco 345
' We first remove the outer Spaces
VBT.§§LIN
PRT.>§§LIN<
' Now we remove the outer numbers and spaces
VRT.§§LIN|123 4567890
PRT.>§§LIN<
SBD.§§LIN|$sp$|2|§§ERG
MBX.§§ERG
END.
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- **VBT. - Variable-Both-sides-Trim** 3225
- **VLT - Variable-Left-Trim** 3228
- **VRD. - Variable-Remove-Doubles** 3076
- **VTU. - Variable to Uppercase** 3059
- **VTL. - Variable to Lowercase** 3053
- **VTM. - Variable to Mediumcase** 3056
- **GFS. - Get-Filtered-String** 3122
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067

© 2023 Theo Gottwald * https://www.smart-package.com

- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140
- RPL. - RePLace in String 3143

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.23 URL. - Split URL

MiniRobotLanguage (MRL)

URL. Command

split URL

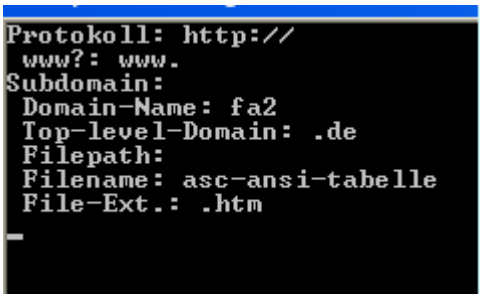
Intention

Splits an internet URL into its different parts.

Example:

```
: $$NUM=1
: $$MAN=http://www.fa2.de/asc-ansi-tabelle.htm
' Output is placed inside these variables in order
URL.$$MAN|$$V00|$$V01|$$V02|$$V03|$$V04|$$V05|$$V06|$$V07
PRT.Protokoll: $$V00
PRT. www?: $$V01
PRT.Subdomain: $$V02
PRT. Domain-Name: $$V03
PRT. Top-level-Domain: $$V04
PRT. Filepath: $$V05
PRT. Filename: $$V06
PRT. File-Ext.: $$V07
MBX.!
END.
```

And here is the result:



Syntax

URL.P1[|P2 ... P9]

Parameter Explanation

P1 - (in) - URL to split into parts

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 - (out) - protocol (http://, ftp:// etc.)

P3 - (out) - www. if there or empty

P4 - (out) - Sub-Domains if any

P5 - (out) - Domain-Name

P6 - (out) - Top-Level-Domain

P7 - (out) - Path

P8 - (out) - Filename

P9 - (out) - File-Extension

The Stack being used is the actual User-Stack.
It can be the Local User Stack or the Global User-Stack, which depends on Settings done with STS.-Command.

If the Popp'ed items are deleted from stack or not also depends on the Settings done with STS.

Example

```
'********************************
' URL. Sample 1
'********************************
: $$NUM=1
: $$MAN=http://www.it-berater.org
' All output is placed on Stack in reverse order
URL.$$MAN
DMP.4
MBX.!
END.

'********************************
' URL. Sample 2
'********************************
: $$NUM=1
: $$MAN=http://www.it-berater.org
' Output is placed inside these variables in order
URL.$$MAN|$$V00|$$V01|$$V02|$$V03|$$V04|$$V05|$$V06|$$V07
PRT.Protokoll: $$V00
PRT. www?: $$V01
PRT.Subdomain: $$V02
PRT. Domain-Name: $$V03
PRT. Top-level-Domain: $$V04
PRT. Filepath: $$V05
PRT. Filename: $$V06
PRT. File-Ext.: $$V07
MBX.!
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- GFT. - Get-File-Token 1037
- GSS. - GetSplitString 3137
- ! STR. - String Command 3150

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.24 Unicode-Conversion

3.79.24.1 ATU. - ANSI to Unicode

MiniRobotLanguage (MRL)

ATU. Command

ANSI to Unicode

Intention

This command returns Text in P1 converted to Unicode 2-Byte Format.

```
' Use it with 1 or 2 Parameters
$$TXT=ABCDEFGHIJK
PRT.$$TXT|h
ATU.$$TXT
PRT.$$TXT|h
MBX. Ready

' Or use it like this
$$TXT=ABCDEFGHIJK
ATU.$$TXT|$$TAR
PRT.$$TXT|h
PRT.$$TAR|h
MBX. Ready
```

Syntax

ATU.[P1][|P2]

Parameter Explanation

P1 **-** ***Input*** String in Variable. If **P2** is omitted the result will also be placed in **P1**.
P2 **-** ***(optional)*** Variable to return the Unicode-Result.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

_

Remarks

This command is the counterpart of the UTA. - Command.

Limitations:

The theoretical maximum length of a variable /text / string is 2 GB. In real life, it is less, because variables need string memory and that must not be fragmented. After all the maximum size of a variable on your system depends on the amount of RAM that is available, and it depends on how fragmented your memory is. Under normal circumstances several hundred MegaBytes should be no problem.

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.79.24.2 UTA. - Unicode to ANSI

MiniRobotLanguage (MRL)

UTA. Command

Unicode to ANSI

Intention

This command returns the Unicode-Text in **P1** to 1-Byte ANSI Format.

```
' Use it with 1 or 2 Parameters
$$TXT=ABCDEFGHIJK
PRT.$$TXT|h
ATU.$$TXT
PRT.$$TXT|h
UTA.$$TXT
PRT.$$TXT|h
MBX. Ready

' Or use it like this
$$TXT=ABCDEFGHIJK
ATU.$$TXT|$$TAR
PRT.$$TXT|h
PRT.$$TAR|h
UTA.$$TAR|$$TXT
PRT.$$TXT|h
MBX. Ready
```

Syntax

UTA.[P1][|P2]

Parameter Explanation

P1 **-** ***Input*** String in Variable. If **P2** is omitted the result will also be placed in **P1**.
P2 **-** ***(optional)*** Variable to return the ANSI-Result.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

_

Remarks

This command is the counterpart of the ATU. - Command.

Limitations:

The theoretical maximum length of a variable /text / string is 2 GB. In real life, it is less, because variables need string memory and that must not be fragmented. After all the maximum size of a variable on your system depends on the amount of RAM that is available, and it depends on how fragmented your memory is. Under normal circumstances several hundred MegaBytes should be no problem.

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- ! STR. - String Command 3150
- GSS. - GetSplitString 3137
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.80 TXT. Text-File Engine

MiniRobotLanguage (MRL)

TXT. the Text-File Engine

This Engine can parse and change Textfiles or CSV-Files very fast (Example 1600 Lines per Second).

Intention

Assume that you want to do something with a Text- or csv-File.
You want to read "Line-by-Line" and write something out of this.
Maybe remove some fields or sort something.

Normally you you could use the Commands like `CFF.`, `LFF.`, `FEL.`, `SBD`, etc.
This will work and will also be sufficient fast in most cases.

But there are cases when you have large files with 50.000 or more Lines,
when its just gets very slow, reading Line-by-Line from the harddisk.

Now here comes the **TXT.-Engine**.
The `TXT.`- Commands will read the whole file into memory and do all changes and stuff also in memory.
As it does not touch the harddrive during the operation, all commands will be done with full speed.

The `TXT.Command` can also load and convert Unicode-Textfiles.
All loaded Textfiles must be converted to `$crlf$`-ending Lines and to ANSI.
For this there are commands available.

If you work on csv-Files, please also take a look on the DCI.-Command 2012.
Using `TXT.` together with `DCI.` you can do all sorts of stuff with Csv.-Files in very short time,
and get very fast results.

Some commands like TXT.Sort... and similar have been **optimized for Speed and tested with Files up to 360.000 Lines.**

Code-Sample

```
' This Sample is a productive Script which is just a Code-Sample.
' It may only be partidally useful for someone else.
'
'SPR Script-file: Dr_Andreas_Debit_Sample
'Purpose:
'Author: Dr.Andreas
'Creation date: 04-28-2022 at 10:21:39
'=========================================================
'#EXE:?path\
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'#SPI:ForceWrite
' Zeilentrenner ist nur $CR

' wieviel treffer der pattern gab es in der datei
' countervarianle $$TRE = TREFFER
$$TRE=0

$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  VAR.$$FIA=?exeloc\BEISPIEL_1.csv
EIF.
VAR.$$FIB=?exeloc\Tempfile_2.csv
VAR.$$OUF=?exeloc\Result_#date#_awt.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.


' Feldnummer vonFeld "U" ist 21
$$MAF=21
$$PTA=1500.00
$$PTB=0.01
$$FIK=?exeloc\Test.txt
$$LIA="Q","M","K","I","H",F"
'-------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$NUL
' Zeilentrenner umwandeln
TXT.crp
TXT.Get_Line_Count|$$NUL
IVV.$$NUL<5
  $$ERR=Ist die Datei leer?
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
'-------------------------------------------------------------

' Finde Ende des Headers
$$HEA=;;Konto;Belegnr;Ar;;Datum;
TXT.Find_Line_with|$$HEA|1|$$LIH
VIC.$$LIH
IVV.$$LIH=0
  $$ERR=Header nicht gefunden.
  GTO.ero
EIF.
TXT.Get_to_Start|$$LIH|$$HEB
'-------------------------------------------------------------
' Header schreiben
'VAR.$$HEC=$$HEB$crlf$
' leere Datei erzeugen
VAR.$$HEC=$crlf$
CTF.$$OUF|$$HEC

'-------------------------------------------------------------
TXT.Replace|$crlf$|$cr$
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:MAIN
' zum testen hier $$NUL patchen
'
'$$NUL=2000
FOR.$$LIN|1|$$NUL
  ' Die folgende Zeile ist unnötig
  ' aber verdoppelt die Geschwindigkeit hier von 78 Sek -> 33 Sek.
  ' diese routine gibt einen optimierten $$LIN zurück
  GSB.GetNextFit

  GSB.Test_Lit

  ' wenn der jump ausgeloest wird,
  ' d.h. es wurde kein pattern gefunden
  ' so ueberspringt er "nur" den folgenden
  ' code. ein if then else haette es auch getan :)
  JIZ.$$ERG|Loop

  ' Zeile gefunden, bestimme Position des Start-Debitors
  CAL.$$POY=$$LIN-8
  $$POY=$$LIN
  ' Hole Ende-Debitor nach $$LEB
  GSB.Find_Deb_End

  'DBP. Zeilen: $$POY bis $$LEB
  ' 828-848 wären OK
  VIC.$$TRE

  TXT.gtls|$$POY|$$LEB|$$PAA
  GSB.Lab_Remove_Cols
  ''$$PAB=$$PAA$crlf$$crlf$########## $$TRE ##########$crlf$$crlf$
  $$PAB=$$PAA$crlf$########## $$TRE ##########$crlf$
  ATF.$$OUF|$$PAB
  :Loop
NEX.

:enx
CAL.$$TMA=#dsince#|i
MBX.Programm wird beendet nach $$TMA Sekunden.
ENR.
:ero
MBX.$$ERR
GTO.enx
'-------------------------------------------------------------
' Remove unwanted Columns
'
:Lab_Remove_Cols
TXT.Push_Intern
TXT.Load_pure_Text|$$PAA|$$FIK
TXT.Delimited_Items_Remove|;|$$LIA
TXT.Get_pure_Text|$$PAA
TXT.Pop_Intern
RET.
'-------------------------------------------------------------
:Find_Deb_End
$$DEB=Debitor
$$LEB=$$LIN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Research
IVV.$$LEB>$$NUL
  $$END=1
  $$LEN=$$NUL
  GTO.LeaveResearch
EIF.
' finde die Zeile mit dem Suchpattner $$DEB
' also hier als Content DEBITOR
'
TXT.Find_Line_with|$$DEB|$$LEB|$$DEH

'TXT.Get_Line_Trimmed|$$DEH|$$DET
'SBD.$$DET|;|1|$$DEA

TXT.get_delimited_item_from_line|$$DEH|;|1|$$DEA
VBT.$$DEA|$lf$$sp$
'PRT.<$$DEA>|h
IVS.$$DEA!$$DEB
  CAL.$$LEB=$$DEH+1
  GTO.Research
EIF.
CAL.$$LEB=$$DEH-1
'PRT.Found at:$$LEB
'MBX.!
:LeaveResearch
RET.
'============================================================
' prüfe nun ob in der zeile die Pattern
' $$PTA bzw $$PTB vorkommen
' wenn ja, setze dir einen Merker (die 1)
' in die Variable $$ERG
'
:Test_Lit
'TXT.Get_Line_Trimmed|$$LIN|$$LIT
'SBD.$$LIT|;|$$MAF|$$RES
'
TXT.get_delimited_item_from_line|$$LIN|;|$$MAF|$$RES
$$ERG=0
IVV.$$RES=$$PTA
  $$ERG=1
EIF.
IVV.$$RES=$$PTB
  $$ERG=1
EIF.
RET.
'============================================================
' ermittel die zeilenummern wo die
' suchpattern $$PTA und $$PTB vorkommen
'
' der kleinere zeilenfundort wird in $$LIN
' geladen, sofern er nicht 0 ist
'
' so kann eine nachfolgende analyse dirkt
' bei $$LIN starten
'
:GetNextFit
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CAL.$$SEB=$$LIN+1
TXT.Find_Line_with|;$$PTA|$$SEB|$$FRA
TXT.Find_Line_with|;$$PTB|$$SEB|$$FRB
'PRT.Got: $$FRA - $$FRB from $$SEB
IVV.$$FRA<$$FRB
  JIZ.$$FRA|DoFrb
  $$LIN=$$FRA
ELS.
  :DoFrb
  NVV.$$FRB=0
    $$LIN=$$FRB
  ELS.
    $$LIN=$$NUL
  EIF.
EIF.
:Leave
'PRT.$$LIN
RET.
'============================================================
```

Code Explanation

-

Example

```
'

'------------------------------------------------------------
END.
```

Remarks

-

Limitations:-

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.1 !Sample-Code: Indentation Checker

TXT.-Commands-Sample Code Previous 2702 Top 41 Next 2706

MiniRobotLanguage (MRL)

Indentation Checker for Python Code

This Script is a Indentation Checker for Python Scripts

Intention

As PYTHON uses a "Block Indentation Syntax", which even checks if you use TAB's or SPACES,
its very easy to make indentation errors.
And depending on the type of Editor you will not even see then because TABs and Spaces are both generally invisible.
Going through a thousands of code lines checking the indentation manually is also not a good solution.
Why not let it the SPR do?
Due to its very easy block-Syntax he does not have this kind of problems, and when compiled he is as fast as Python Code or even faster.
Using the SPR and its PlugIn we can realize a Indentation Checker for Python Scripts quickly.
The Code below is an example for that.
After changing its structure from FEL. to the new TXT.-Commands, it got a speedup fo ten times!
Currently it takes just about 1 second for checking 1000 Lines of Code.

```
'#EXE:?path\
'#SPI:ForceWrite
'#ICS:39
NEM.
  'GTO.enx
  VAR.$$INF=?path\V2.py
ELS.
  VAR.$$INF=$cmdexe$
EIF.
VAR.$$TIA=#time#

VAR.$$OUF=?exeloc\Backup.py
DEL.$$OUF
NEF.$$INF
  VAR.$$INF=?exeloc\Library.py
EIF.
NEF.$$INF
  MBX.Can not open: $$INF
  GTO.enx
EIF.
PRT.Lib-Checker started: $now$
PRT.-------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$DEF=def$sp$
VAR.$$SP3=$sp$$sp$$sp$
VAR.$$TAB=$tab$
VAR.$$DEE=$crlf$
VAR.$$DEL=$tab$$sp$
VAR.$$LFF=$lf$
VAR.$$CRC=$cr$
VAR.$$CRL=$crlf$
' Read File
TXT.Load_Textfile|$$INF|$$ANZ
TXT.Save|$$OUF
VAR.$$IDL=0
VAR.$$ADL=0
VAN.$$BLM=0
FOR.$$CNT|1|$$ANZ
  TXT.get_line|$$CNT|$$LIN
  GSB.CheckLine
NEX.
TXT.Save
CAL.$$TIB=(#time#-$$TIA)|i
MBX.Ready in $$TIB Seconds.
:enx
ENR.
'============================================================
' Check for:
' $$IDL = Ident-Level (Anzahl nötiger Spaces)
' NoSPaces links und rechts
' $$LIN - Original
' $$LIA - nur links getrimmt
' $$LIB - nur rechte Seite getrimmt
' $$LIX - Korrigiert
:CheckLine
GSB.GenLIV

GSB.CheckBlockComments

' Blockkommentare alles überlaufen
CAL.$$BLO=$$BLM+$$BLN
IVV.$$BLO>0
  VAR.$$LIX=$$LIB
  GTO.comment
EIF.


' Leerzeile
LEN.$$LIA|$$LEA
IVV.$$LEA=0
  VAR.$$LIX=
  GTO.enl
EIF.

' Prüfe Kommentare
GSS.1|1|$$LIC|$$LAS
IVS.$$LAS=#
  $$LIX=$$LIC
  GTO.comment
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Vorprüfungen
GSB.CheckIMaxdentLevelChange
GSB.CountIdentofLine
GSB.CheckIdentofLine

:enl
IVS.$$LIB!$$LIX
  PRT.Changes in Line: $$CNT
  PRT.($$BLM)-($$IDL)<-$$LIX
EIF.
:comment
VAR.$$IDL=$$ADL
TXT.set_line_to|$$LOP|$$LIX
RET.

'-------------------------------------------------------------
:CheckIdentofLine

RET.
'-------------------------------------------------------------
:CountIdentofLine
VAR.$$SPA=0
VAR.$$TAA=0
VAR.$$FNI=
VAR.$$LOA=0
FOR.$$LOA|1|$$LEB
  GSS.$$LOA|$$LOA|$$LIB|$$LAS
' Wurde Leerzeichen gefunden?
  IVS.$$LAS=$tab$
    VIC.$$TAA
  ELS.
    IVS.$$LAS=$sp$
      VIC.$$SPA
    ELS.
      GTO.others
    EIF.
  EIF.
NEX.
:others
JIV.$$LOP=1|NoIdent
IVV.$$SPA>0
  IVV.$$TAA>0
    PRT.Ident in Line $$CNT is wrong (Tabs and Spaces) please check it.
    VAR.$$TAN=$$TAA
  ELS.
    CAL.$$TAN=$$SPA/4
  EIF.
  GSB.GenwithTabs
ELS.
  IVV.$$TAA>0
    IVV.$$TAA>$$IDL
      PRT.Possible Ident-Error in Line $$CNT (too much idented)?
      VAR.$$TAN=$$IDL
    ELS.
      VAR.$$TAN=$$TAA
      VAR.$$IDL=$$TAA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    EIF.
    GSB.GenwithTabs
  ELS.
    GTO.NoIdent
  EIF.
EIF.
RET.
'-------------------------------------------------------------
:NoIdent
VAR.$$LIX=$$LIA
VAR.$$IDL=0
RET.
'-------------------------------------------------------------
' Erzeuge Indent mit Tabs
:GenwithTabs
STR.CLONE|$$FID|$$TAN|$$TAB
VAR.$$LIX=$$FID$$LIA
RET.
'-------------------------------------------------------------
:CheckIMaxdentLevelChange
GSS.-1|-1|$$LIA|$$LAS
IVS.$$LAS=:
  VIC.$$ADL
EIF.
GSS.1|4|$$LIA|$$LAS
IVS.$$LAS=$$DEF
  $$IDL=0
EIF.
RET.
'-------------------------------------------------------------
:CheckBlockComments
GSS.1|3|$$LIC|$$LAS
' Blockkommentare
IVS.$$LAS="""
  CAL.$$BLM=1-$$BLM
ELS.
  IVS.$$LAS='''
    CAL.$$BLM=1-$$BLM
  EIF.
EIF.
IVS.$$LIC=#region INFO
  VAN.$$BLN=0
ELS.
  IVS.$$LIC=#endregion
    IVV.$$BLN=1
      VAN.$$BLN=0
    EIF.
  EIF.
EIF.
RET.
'-------------------------------------------------------------
:GenLIV
VAR.$$LIB=$$LIN
VRT.$$LIB|$$DEL
VAR.$$LIC=$$LIN
VBT.$$LIC
VAR.$$LIA=$$LIB
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  VLT.$$LIA|$$DEL
  LEN.$$LIB|$$LEB
  RET.
```

Code Explanation

Example

```
'

'------------------------------------------------------------
END.
```

Remarks

-

Limitations:-

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.2 !TXT. - Text File Engine

MiniRobotLanguage (MRL)

TXT.-Commands

Text-File Operations commands

Intention

These commands are the easy way to work through Textfiles.
What you can do:

1. Load and Save Textfiles,csv-Files
2. Process Textfiles from an internal Cache without need to access the Filesystem
3. Process csv- or delimited Textfiles in multiple ways with a high Speed. Typical real life example are about 1600 - 2000 Lines per second.
4. Store Textfile in Variable (with or without Filename)
5. Restore Textfile from Variable (with or without Filename)
6. Store TXT.-Files in an internal Memory and restore it from there using the TXT.Push_Intern / TXT.Pop_Intern Commands.
7. Remove, insert or Swap colums in a csv-File.
8. Get the Linenumber of a Byte-Position in a Textfile
9. Get the complete Line of specified a Linenumber
10. Get all Lines from specific line to the start or end of text
11. Get the start or end of a Line from a specified Byte-Position
12. Change specfied Lines (replace with a specified line)
13. Get Textfile from the LOF./LFT.-Command or
14. Move a textfile to the LOF./LFT.-Command
15. several Replace Operations and more.

Here is a Quick and incomplete Overview on the Commands:

"ltf","load_textfile"

Load a Textfile int othe internal Buffer for Processing

```
' $$FIL - Filename
' $$RES (optional) Returns Number of loaded lines
TXT.Load_Textfile|$$FIL[|$$RES]
```

"clr","clear"

Clear the internal Buffer

```
TXT.Clear
```

"gfn","get_filename"

```
' Get Filename of File in Buffer
TXT.gfn|$$FIL
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

"gtl","get_lenght"

```
' Get lenght of loaded Text in Bytes
  TXT.gtl|$$LAC
```

"tov","to_var"

```
' Move Textfile including  Filename to Variable
  TXT.To_Var|$$VAR
```

"toc","to_var_and_clear"

```
' Move Textfile including  Filename to Variable and clear internal Buffer
  TXT.To_Var|$$VAR
```

"frv","from_var"

```
' Get Textfile including Filename from Variable
  TXT.From_Var|$$VAR
```

"lpt","load_pure-Text"

```
' Load Pure-Textfile and Filename from 2 Variables
  TXT.lpt|$$TXT[|$$FIL]
```

"gpt","get_pure_text"

```
' "Get Pure Text". Only Textfile into $$TXT. If available, $$FIL returns the Fil
  TXT.gpt|$$TXT[|$$FIL]]
```

"glc","get_line_count"

```
' "Get Line Count". Get number of loaded lines.
  TXT.glc|$$CNT
```

"gel","get_line"

```
' Get Line Number $$NUM into $$LIN
' TXT.gel|$$NUM|$$LIN
```

"gll","get_line_lefttrim"

```
' Return Line $$LIN, like "gel" but left side trimmed (all ASC 0-32 removed)
  TXT.gll|$$LIN|$$RES
```

"glt","get_line_trimmed"

```
' Return Line $$LIN, like "gel" but both sides trimmed (all ASC 0-32 removed)
  TXT.glt|$$LIN|$$RES
```

"slt","set_line_to"

```
' Overwrite Line $$NUM with new Line $$NEW (Replace Line)
  TXT.slt|$$NUM|$$NEW
```

"glp","get_line_position"

```
' Get position of first and last character of line $$NUM into $$POS and last int
  TXT.glp|$$NUM|$$POS|$$POE
```

"lnp","linenumber_from_position"

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Get linenumber from Byte-Position $$POS into $$LIN
  TXT.lnp|$$POS|$$LIN
```

"lsp","line_Start_end_by_position"

```
' Get start/end of Line by Byte-Position in Text
  TXT.lsp|$$LIN|$$STR|$$END
```

"flw","find_line_with"

```
' Find Line with String. Search first occurence of $$SEA in Textfile, return the
' Falls angegeben suche ab (inkl.) Zeile $$STA
  TXT.flw|$$SEA|$$LIN[|$$STA]
```

"fit","Find_Text"

```
' $$SEA - Searchstring
' $$FRO - Search from (B)yte or (L)ine
' Returns:
' |$$POT - Postion in Text
' |$$POL - Postion in Line
' |$$LIN - Complete Line
' |$$LNN - Linenumber
  TXT.fit|$$SEA|$$FRO[|$$POT][|$$LIN][|$$LNN]
```

"gbl","get_before_line"

```
' "Get Before Line". Get all Text that is before and without the specified line
  TXT.gbl|$$LIN|$$BLT
```

"gal","get_after_line"

```
' "Get After Line". Get all Text that is After and without the specified line in
  TXT.gal|$$LIN|$$ALT
```

"gts","get_to_start"

```
' Get all Text from and including the specified line until the start of the text
  TXT.gts|$$LIN|$$TXT
```

"gte","get_to_end"

```
' Get all Text from and including the specified line until the end of the text
  TXT.gte|$$LIN|$$TXT
```

"sav","save_file"

```
' Save Text under the Filename with which it was loaded from, or if specified wi
' Leaves a result on the TOS, NTFSD-Enabled
  TXT.Save[|$$FIL]
```

"gfl","get_from_lof"

```
' Get from LOF # will load Textfile from LOF. Index Number $$IND
  TXT.get_from_lof|$$IND
```

"tol","to_lof"

```
' To lof cache # will put Textfile into LOF. Cache Index Number $$IND
  TXT.To_Lof|$$IND
```


© 2023 Theo Gottwald * https://www.smart-package.com

"rit","replace_in_text"

```
' "replace_in_text". Do a replace over the complete text using a Equalcase-Repla
' TXT.rit|$$OLD|$$NEW
```

"ril","replace_in_line"

```
' "replace_in_line". Do a replace only in the specified Line using a Equalcase-R
  TXT.ril|$$LIN|$$OLD|$$NEW[|$$STA]
```

"rep","Replace"

Search a Text in the TXT.-internal Buffer and replace it with a given other sequence.
Will replace all occurrences.

"tra","Translate_Chars"

Using TXT.tra there is a 1:1 relation between the characters in $$OLD and the characters in $$NEW.
So if a character in $$OLD is found its replaced with the corresponding Character in $$NEW
Therefore both - $$OLD and $$NEW - MUST have the same lenght, esle there will be no changes and you will get the Timeout Flag set.

```
TXT.tra|$$OLD|$$NEW[|$$STA][|$$END]
```

"luf","load_unicode_file"

```
' Load and convert Unicode-File to ANSI
  TXT.luf|$$FIL
```

"sau","save_as_unicode"

```
' Convert ANSI-File to Unicode and save (using default codepage)
  TXT.sau[|$$FIL]
```

"ctu","convert_to_unicode"

```
' Convert ANSI-String to Unicod with default Codepage
  TXT.cta|$$ANS|$$UNI
```

"cta","convert_to_ansi"

```
' Convert Unicod with default Codepage to ANSI-String
  TXT.cta|$$UNI|$$ANS
```

See also:

- LOF. - Load-Text-File 1063
- LFT. - Line-From-Text 1060

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3 Conversions

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.1 Convert_to_ANSI

MiniRobotLanguage (MRL)

TXT.cta
TXT.Convert_to_ANSI

This command will do several converts from Unicode to ANSI.

Intention

This command has three modes of Operation:

```
TXT.cta
```

Without Parameters it will convert the TXT.-internal Buffer to ANSI.

```
TXT.cta|$$TXT
```

Called with **P1**, it will convert the content of **P1** to ANSI and store it on the TOS.

```
TXT.cta|$$TXT|$$UNI
```

Called with **P1 and P2**, it will convert the content of **P1** to ANSI and store it in P2.

If you convert files from Unicode to ANSI and back, please note that due the nature of this process, some international characters may get lost.

```
TXT.Convert_to_Unicode|$$FIL
MBX.Operation was succesfull, Result: $tos$
```

There is no checking if the file or Text was Unicode before. Therefore converting an ANSI File to ANSI may lead to a loss of half the data.

Syntax

TXT.cta[|P1][|P2]
TXT.Convert_to_ANSI[|P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - (Input, optional) Variable with Unicode-Content. If omitted the TXT.-internal Buffer is been used.

P2 - (Output) Variable with the result of the operation. If omitted the result is placed on TOS.

Example

```
'******************************************************
' EXAMPLE 1: TXT.luf  and TXT.cta
'******************************************************
VAR.$$LAA=?path\..\Pics\Outlook.msg
TXT.luf|$$LAA
TXT.cta
TXT.gpt|$$TXT
PRT.$$TXT
MBX.Ready
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- Split File 1001
- Join File 994
- OPT. - Optional Settings 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.2 Convert_to_Unicode

MiniRobotLanguage (MRL)

TXT.ctu
TXT.Convert_to_Unicode

This command will do several converts from ANSI to Unicode.

Intention

This command has three modes of Operation:

```
TXT.ctu
```

Without Parameters it will convert the TXT.-internal Buffer to Unicode.

```
TXT.ctu|$$TXT
```

Called with **P1**, it will convert the content of **P1** to Unicode and store it on the TOS.

```
TXT.ctu|$$TXT|$$UNI
```

Called with **P1 and P2**, it will convert the content of **P1** to Unicode and store it in P2.

If you convert files from Unicode to ANSI and back, please note that due the nature of this process, some international characters may get lost.

```
TXT.Convert_to_Unicode|$$FIL
MBX.Operation was succesfull, Result: $tos$
```

Syntax

TXT.ctu[|P1][|P2]
TXT.Convert_to_Unicode[|P1][|P2]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - (Input, optional) Variable with ANSI-Content. If omitted the TXT.-internal Buffer is been used.

P2 - **(Output)** Variable with the result of the operation. If omitted the result is placed on TOS.

Example

```
'******************************************************
' EXAMPLE 1: TXT.luf  and TXT.ctu
'******************************************************
VAR.$$LAA=?path\..\Pics\Outlook.msg
TXT.ltf|$$LAA
TXT.ctu
TXT.gpt|$$TXT
PRT.$$TXT
MBX.Ready
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- Split File 1001
- Join File 994
- OPT. - Optional Settings 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.3 CR_to_CRLF

MiniRobotLanguage (MRL)

TXT.cr_to_crlf
TXT.crp

Replace all cr (ASC 13) from the TXT.-internal Buffer with a new Line ($crlf$ ASC 13,10).

Intention

This command will do a Replace of all ASC 13 Characters (CR) with ASC(13,10) $crlf$ in the complete loaded Text.
Some files - for example csv-files that have been exported from MS-Office Excel, may not contain a $crlf$ at the end of each line,
but may contain just a `$cr$` instead.
To work with these files, we must change the Line-End to `$crlf$`.
If there are zero `$lf$` in the file, the `TXT.Load_Textfile` Command may do that automatically.
Sometimes if there are still a few `$lf$` inside the text, then you need to do that manually.

```
' Change Line Number $$NUM into $$NEW
TXT.Load_Textfile|$$FIA|$$NUM
TXT.crp
TXT.Get_Line_Count|$$NUM
IVV.$$NUL<5
  $$ERR=Ist die Datei leer?
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
```

Syntax

TXT.cr_to_crlf
TXT.crp

Parameter Explanation

-

Example

```
'*****************************************************
' EXAMPLE 1: TXT.crp
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  VAR.$$FIA=?exeloc\BEISPIEL_1.csv
EIF.

NEF.$$FIA
  $$ERR=Sourcefile: $$FIA$crlf$ is missing.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$NUL
TXT.crp
TXT.Get_Line_Count|$$NUL
IVV.$$NUL<5
  $$ERR=File is empty.
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.4 CRLF_to_CR

MiniRobotLanguage (MRL)

TXT.crlf_to_cr
TXT.crr

Replace all $crlf$ (ASC 13,10) from the TXT.-internal Buffer with a cr (ASC 13).

Intention

This command will do a Replace of all ASC 13,10 Characters (`$crlf$`) with ASC(13) `$cr$` in the complete loaded Text.
Some files - for example csv-files that have been exported from MS-Office Excel, may not contain a $crlf$ at the end of each line,
but may contain just a `$cr$` instead. To work with these files, we must change the Line-End to `$crlf$`.
If there are zero `$lf$` in the file, the `TXT.Load_Textfile` Command may do that automatically.
Sometimes if there are still a few `$lf$` inside the text, then you need to do that manually.
Now when you want to convert it back, before you save the file, you can use this command.

```
' Change Line Number $$NUM into $$NEW
TXT.Load_Textfile|$$FIA|$$NUM
TXT.crp
TXT.Get_Line_Count|$$NUM
IVV.$$NUL<5
  $$ERR=Ist die Datei leer?
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
TXT.crr
TXT.Save|$$FIA
```

Syntax

TXT.crlf_to_cr
TXT.crr

Parameter Explanation

-

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'******************************************************
' EXAMPLE 1: TXT.crr
'******************************************************
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  VAR.$$FIA=?exeloc\BEISPIEL_1.csv
EIF.

NEF.$$FIA
  $$ERR=Sourcefile: $$FIA$crlf$ is missing.
  GTO.ero
EIF.
'------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$NUL
TXT.crp
TXT.Get_Line_Count|$$NUL
IVV.$$NUL<5
  $$ERR=File is empty.
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
TXT.crr
TXT.Save|$$FIA
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355

© 2023 Theo Gottwald * https://www.smart-package.com

- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.5 CRLF_to_LF

MiniRobotLanguage (MRL)

TXT.crlf_to_lf
TXT.lfr

Replace all $crlf$ (ASC 13,10) from the TXT.-internal Buffer with a lf (ASC 10).

Intention

This command will do a Replace of all ASC 13,10 Characters (`$crlf$`) with ASC(10) `$lf$` in the complete loaded Text.
Some files - for example csv-files that have been exported from MS-Office Excel, may not contain a $crlf$ at the end of each line,
but may contain just a `$cr$` instead. To work with these files, we must change the Line-End to `$crlf$`.
If there are zero `$cr$` in the file, the `TXT.Load_Textfile` Command may do that automatically.
Sometimes if there are still a few `$lf$` inside the text, then you need to do that manually.
Now when you want to convert it back, before you save the file, you can use this command.

```
' Change Line Number $$NUM into $$NEW
TXT.Load_Textfile|$$FIA|$$NUM
TXT.lfp
TXT.Get_Line_Count|$$NUM
IVV.$$NUL<5
  $$ERR=Ist die Datei leer?
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
TXT.lfr
TXT.Save|$$FIA
```

Syntax

TXT.crlf_to_lf
TXT.lfr

Parameter Explanation

-

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.crr
'*****************************************************
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  VAR.$$FIA=?exeloc\BEISPIEL_1.csv
EIF.

NEF.$$FIA
  $$ERR=Sourcefile: $$FIA$crlf$ is missing.
  GTO.ero
EIF.
'-------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$NUL
TXT.lfp
TXT.Get_Line_Count|$$NUL
IVV.$$NUL<5
  $$ERR=File is empty.
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
TXT.lfr
TXT.Save|$$FIA
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355

© 2023 Theo Gottwald * https://www.smart-package.com

- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.6 LF_to_CRLF

MiniRobotLanguage (MRL)

TXT.lf_to_crlf
TXT.lfp

Replace all lf (ASC 10) from the TXT.-internal Buffer with a new Line ($crlf$ ASC 13,10).

Intention

This command will do a Replace of all ASC 10 Characters (LineFeed) with ASC(13,10) $crlf$ in the complete loaded Text.
Some files - for example csv-files that have been exported from MS-Office Excel, may not contain a $crlf$ at the end of each line,
but may contain just a `$lf$` instead.
To work with these files, we must change the Line-End to `$crlf$`.
If there are zero `$cr$` in the file, the `TXT.Load_Textfile` Command may do that automatically.
Sometimes if there are still a few `$cr$` inside the text, then you need to do that manually.

```
' Change Line Number $$NUM into $$NEW
TXT.Load_Textfile|$$FIA|$$NUM
TXT.lfp
TXT.Get_Line_Count|$$NUM
DBP.$$NUL Lines loaded.
```

Syntax

TXT.lf_to_crlf
TXT.lfp

Parameter Explanation

-

Example

```
'*****************************************************
' EXAMPLE 1: TXT.lfp
'*****************************************************
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  VAR.$$FIA=?exeloc\BEISPIEL_1.csv
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.

NEF.$$FIA
  $$ERR=Sourcefile: $$FIA$crlf$ is missing.
  GTO.ero
EIF.
'-------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$NUL
TXT.lfp
TXT.Get_Line_Count|$$NUL
IVV.$$NUL<5
  $$ERR=File is empty.
  GTO.ero
EIF.
DBP.$$NUL Zeilen geladen
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.3.7 Translate_Chars

MiniRobotLanguage (MRL)

TXT.tra
TXT.Translate_Chars

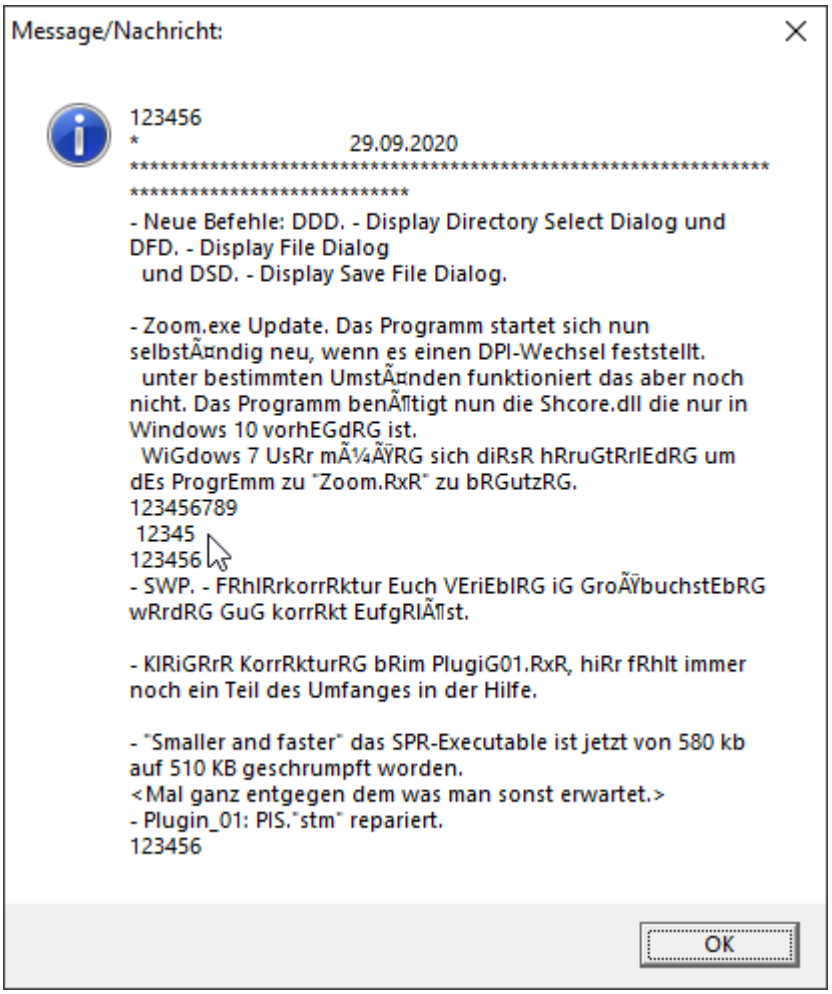
Translate Characters in the TXT.-internal Buffer

Intention

Using TXT.tra there is a 1:1 relation between the characters in $$OLD (P1) and the characters in $$NEW (P2).
So if a character in $$OLD is found it is replaced with the corresponding Character in $$NEW.
Therefore both - $$OLD and $$NEW - MUST have the same lenght, else there will be no changes and you will get the Timeout Flag set.

```
'Translate_Chars
' a -> E
' n -> G
' e -> R
' Looks like this:
TXT.tra|ane|EGR|520|800

VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' This example shows all features of TXT.tra
TXT.tra|ane|EGR|520|800
' You can also see that the Translate_Chars starts a Position
520, as specified and ends at Position 800.
' So the start of the text is unchanged and also the end.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

```
TXT.tra|P1|P2[|P3][|P4]
TXT.Translate_Chars|P1|P2[|P3]
[|P4]
```

Parameter Explanation

P1 – **`(Input)`** Variable with Characters to translate. Using TXT.tra each occurrence of each character in **P1** will be replaced with the corresponding character in **P2**.
because of that, **P1** and **P2** must be the same length. There is a one-to-one correspondence between their characters.

P2 – **`(Input)`** Variable with the String that shall replace the characters in **P1**.

P3 – **`(`***optional, numeric***`)`** Variable with a Byte-Position in the TXT.-Buffer, if given the Translate_Chars will start at that Position.
If omitted the Text is been searched and replaced from Position 1.

P4 – **`(`***optional, numeric***`)`** Variable with a Byte-Position in the TXT.-Buffer, if given the Translate_Chars will end at that Position.
If omitted the complete Text is been searched and replaced to the end.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
VAN.$$STA=100
'Translate_Chars all "a" with "e" and "s" with "f" in internal
TXT.-Buffer starting from Byte-Position 100
VAR.$$OLD=as
VAR.$$NEW=ef
TXT.rep|$$OLD|$$NEW|$$STA
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4 CSV-Related Commands

Here is a Template that you can use to process .csv-Files.

```
'SPR Script-file: CSV-Template
'Purpose: Just add your code and you are done.
'Author: Theo Gottwald
'Creation date: 12-15-2021 at 16:41:27
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?_x_tr_sl=en&_x_tr_tl=de&_x_tr_h
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  VAR.$$FIA=?exeloc\Sources\MaxSize.CSV
EIF.
' This is the Output-File
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
'Sample Script follows
$$IND="L"
$$STA=2
$$DEL=;
TXT.sort_by_column_letter|$$STA|$$IND|$$DEL|a

$$SEA="Batterie"
' We use Standard Search Pattern
TXT.gcb|$$IND|$$STA|$$SEA|$$DEL|e

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' The following coe will save the results
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000
CAL.$$LIF=($$LIC*1000)/$$TMA|i
$$TXA=$crlf$Processing $$LIF Lines per Second.
IVV.$$TMA>1000
  CAL.$$TMA=$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.$$TXA
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.$$TXA
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'#############################################################################
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.1 Delimited_Items_Insert

MiniRobotLanguage (MRL)

TXT.Delimited_Items_Insert

Insert Fields before specified places into a Delimited String, process the whole TXT.-Buffer with all lines in one run.

Intention

Technically this works like the DIC.Insert-Command.
It will however be applied to all Lines in the TXT.-Buffer.
Therefore you do not need to specify a Line, or target. Target is the complete loaded file.

This Sample is from the DIC.-Insert Command.

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you want to insert Columns before "A" and before "E".

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' This is the Index-List with the Elements where we want to insert new fiels bef
VAR.$$IND="A","E"

' Now we call the command
DIC.Insert|$$DEL|$$LIN|$$IND|$$RES

' And Debug-Print the Result
DBP.$$RES
ENR
```

And this is the result: **`;AA;BB;CC;DD;;EE;FF`**

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

TXT.dic|P1|P2
TXT.Delimited_Items_Insert|P1|P2

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`. This is the specified delimiter that delimits the fields in the specified Line **P2**.
P2 – Variable with Index of Fields to Insert Fields before. Technically the command will just place a delimiter in front of these fields.
Can be Single number like 1 or can be an Letter like "A". You can give a **comma-separated list with Numbers of the fields to get**.

Example

```
'***********************************
' DIC.Insert - Sample
'***********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' This is the Index-List with the Elements where we want to insert new fiels bef
  VAR.$$IND="A","E"

  ' Now we call the command
  DIC.Insert|$$DEL|$$LIN|$$IND|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  '  ;AA;BB;CC;DD;;EE;FF
  ENR
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.2 Delimited_Items_Remove

MiniRobotLanguage (MRL)

TXT.Delimited_Items_Remove

Remove specified Fields from a Delimited String, similar to DIC.Remove, but process the whole TXT.-Buffer, not just a line.

Intention

This command does the same like DIC.Remove 3096 but will process the whole TXT.-Buffer "Line by Line".
Therefore it does not need an Line-Specifier.

The following Sample is from DIC.Remove 3096
Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you want to remove Columns 1,3, and 5.

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF

' This is the Index-List with the Elements where we want to insert new fiels bef
VAR.$$IND="A","E",3

' Now we call the command
DIC.remove|$$DEL|$$LIN|$$IND|$$RES

' And Debug-Print the Result
DBP.$$RES
ENR
```

And this is the result: **`BB;DD;FF`**

Syntax

TXT.dir|P1|P2

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`.

© 2023 Theo Gottwald * https://www.smart-package.com

This is the specified delimiter that delimits the fields in the specified Line **P2**.

P2 – Variable with Index of Fields to remove.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.

Example

```
'**********************************
' DIC.Remove - Sample
'**********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' This is the Index-List with the Elements where we want to insert new fiels bef
  VAR.$$IND="A","E",3

  ' Now we call the command
  DIC.Remove|$$DEL|$$LIN|$$IND|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  '  BB;DD;FF

  ENR
```

Remarks

-

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929

© 2023 Theo Gottwald * https://www.smart-package.com

- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.3 Delimited_Items_Swap

MiniRobotLanguage (MRL)

TXT.Delimited_Items_Swap
TXT.dis

Exchange two Fields inside a Delimited String, similar to DIC.Swap, but process the whole TXT.-Buffer, not just a line.

Intention

This command does the same like DIC.Swap 3104 but will process the whole TXT.-Buffer "Line by Line".
Therefore it does not need an Line-Specifier.

The following Sample is from DIC.Swap 3104

Assume you have a Line like this:
`AA;BB;CC;DD;EE;FF`
as you can read it Line-by-line out of a csv.-file

Now, you decide that you want to insert Columns before "A" and before "E".

```
' This is the delimiter character
VAR.$$DEL=;

' This is the Line that has delimited items
VAR.$$LIN=AA;BB;CC;DD;EE;FF$crlf$
VAV.$$LIM=$$LIN$$LIN$$LIN
TXT.Load_Pure_Text|$$LIM
' Theseare the two fields to be exchanged
VAR.$$INA="A"
VAR.$$INB="E"

' Now we call the command
TXT.dis|$$DEL|$$INA|$$INB

' And Save the Result
$$OUT=?exeloc\Result.txt
TXT.Save_File|$$OUT
ENR.
```

And this is the result:

EE;BB;CC;DD;AA;FF
EE;BB;CC;DD;AA;FF
EE;BB;CC;DD;AA;FF

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

TXT.Dis|P1|P2|P3
TXT.Delimited_Items_Swap|P1|P2|P3

Parameter Explanation

P1 – Delimiter, can be a Single character like semicolon or multiple characters like `$crlf$`.
This is the specified delimiter that delimits the fields..
P2 – Variable with Index of Fields A to Swap with Field in **P3**.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.
P3 – Variable with Index of Fields B to Swap with Field in **P2**.
Can be Single number like 1 or can be an Letter like "A". You can give a comma-separated list with Numbers of the fields to get.

Example

```
'***********************************
' DIC.Swap - Sample
'***********************************
  ' This is the delimiter character
  VAR.$$DEL=;

  ' This is the Line that has delimited items
  VAR.$$LIN=AA;BB;CC;DD;EE;FF

  ' Theseare the two fields to be exchanged
  VAR.$$INA="A"
  VAR.$$INB="E"

  ' Now we call the command
  DIC.Swap|$$DEL|$$LIN|$$INA|$$INB|$$RES

  ' And Debug-Print the Result
  DBP.$$RES

  ' Result:
  '  EE;BB;CC;DD;AA;FF
  ENR
```

Remarks

This command is currently not optimized for very large Tables with 100.000's of lines. Using it for very large tables may lead to significant waiting times.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- 1.5.2 Working with Text-Strings 110
- VAR. - Variable Set Value/Clear 3562
- IVS. / NVS. - If-Variable-String 3531
- GSS. - GetSplitString 3137
- SRT. - Sort Items 3147
- ! STR. - String Command 3150
- GES. - Get-Extracted-String 3067
- SBD. - String between Delimiter 3070
- SBM. - String between Delimiter-Multiple 3070
- FEM. - For Each Member 929
- LEN. - Length-of-String 3140

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.4 Filter_Lines

MiniRobotLanguage (MRL)

TXT.flt
TXT.Filter_Lines

Only keep Lines that contain specified Words in a specified Column.

Intention

This command is useful if you have to do with csv-files. Assume you have a List with Members of an Organisation.
Now you want to only keep certain Members in your Result List, for example those from a specified Country and with specified Occupation.
Like this one: WEF-Members (Internet-Link)

In Column "F" there you can read the profession, and in Column "L" the Country. Like this: (Sample)

	A	B	C	D	E	F	G	H	I	J	K	L	M
	YEAR	First name	Family name	Link bio	Sex	Occupation /	Organisation	Political/Gov	Category	Past / other	University	Country	Region
	2001 GLT	Andrew	McLaughlin	– no bio	M	– no info			-			?	
	1996 GLT	Michael J.	Smith	– no bio	M	– no info	?		?			?	?
	1993 GLT	Martin A.	O'Neill	– no bio	M	President an	Hi-Touch Int		-	Business		?	?
	2016 YGL	Hassina	Syed	Hassina Syed	F	Womens Rig	Chamber of Commerce			Activist		Afghanistan	Asia
	2021 YGL	Mohammad	Omaid	Mohammed	M	Director	Azizi Bank			Banking		Afghanistan	Asia
	2011 YGL	Fawad	Ahmad Musl	– no bio	M	President an	Jahaan Techr		-	Business		Afghanistan	Asia

Now you just want all "Politician" and other "Actors" from "Germany" or "Poland".
This is the Script (active Part, full Script see below at Example):

```
$$IND="L"
$$SEA="Germany","Poland"
$$STA=2
$$DEL=;
' We use Stadard Search Pattern
TXT.flt|$$IND|$$SEA|$$STA|$$DEL|w
TXT.dbp

$$IND="F"
$$SEA="Politician","Actor"
$$STA=2
$$$DEL=;
' We use Stadard Search Pattern
TXT.flt|$$IND|$$SEA|$$STA|$$DEL|e
```

Here is a Part of the Result.

13	2015 YGL	Franziska 'Sk	Keller	Ska Keller	F	Politician	European Pa	MEP, Leader	Political & Government		Germany	Europe
14	2016 YGL	Jens	Spahn	Jens Spahn	M	Politician	Government	Federal Mini	Political & Government		Germany	Europe
15	2020 YGL	Annalena	Baerbock	Annalena Ba	F	Politician	Federal Fore	Minister of F	Political & Government		Germany	Europe
16	2012 YGL	Daniel	Bahr	Daniel Bahr	M	Politician	Government	Federal Mini	Politics & Government		Germany	Europe
17	1993 GLT	Jan Maria	Rokita	Jan Rokita	M	Politician	Parliament c	Former Mem	Political & Government		Poland	Europe
18	1994 GLT	Aleksander	Kwasniewsk	Aleksander K	M	Politician	Government	Government	Political & Government	Gdansk	Poland	Europe
19	1995 GLT	Waldemar	Pawlak	Waldemar Pa	M	Politician	Government	Prime Minist	Political & G	Deputy Prime Minister an	Poland	Europe

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

TXT.flt|P1|P2[|P3]
TXT.Filter_Lines|P1|P2[|P3]

Parameter Explanation

P1 **- (Input)** Column number which must contain the specified words in **P3**.
P2 **- (Input)** Variable with List of item to search for.The List must contain the words enclosed in "" and separated with ",".
Only Lines with any of the specified words in the Column P1 will be left in the final result.
Example: `"Fox","Wulf",Neutron"`
P3 **- (Input)** Startline, if omitted default is 2 (to skip Headerline). First upper Line is Line 1. Lines before the Startline will be in the result untouched.
P4 **- (Input)** Variable with delimiter
P5 **- (Input)** **Flag**, can be any of these: "**p**" (default) - normal compare, "**w**" - compare using SPR-Standard-Search Patterns,
"**e**" - Equalcase Compare, "**c**" - contains, "**ce**" - contains equalcase, "**x**" - Exclusive Flag (do NOT list Lines before the Startline)

Example

```
'*****************************************************
'
'*****************************************************
  ' csv-File from:
  ' https://maloneinstitute-org.translate.goog/wef?_x_tr_sl=en&_x_tr_tl=de&_x_tr_h
  '
  $$TMA=#dtime#
  IEM.
    VAR.$$FIA=$cmdexe$
  ELS.
    VAR.$$FIA=?exeloc\WEF-Members.CSV
  EIF.
  VAR.$$OUF=?exeloc\Result_#date#.csv
  NEF.$$FIA
    $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
    GTO.ero
  EIF.
  '-------------------------------------------------------------
  TXT.Load_Textfile|$$FIA|$$LIC
  ' Convert $LF to $crlf$ if needed.
  'TXT.lfp
  TXT.Get_Line_Count|$$LIC
  IVV.$$LIC<5
    $$ERR=File is empty?
    GTO.ero
  EIF.
  PRT.$$LIC Lines loaded.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here
$$IND="L"
$$SEA="Germany","Poland"
$$STA=2
$$DEL=;
' We use Stadard Search Pattern
TXT.flt|$$IND|$$SEA|$$STA|$$DEL|w
TXT.dbp

$$IND="F"
$$SEA="Politician","Actor"
$$STA=2
$$$DEL=;
' We use Stadard Search Pattern
TXT.flt|$$IND|$$SEA|$$STA|$$DEL|e

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'############################################################################
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.5 Find_first_in_Column

MiniRobotLanguage (MRL)

TXT.Fic
TXT.Find_first_in_Column

Find the first Entry in a specified column with a specified value.

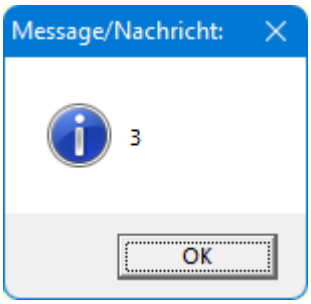
Intention

This command is useful if you have to do with csv-files.
Most often this can be used after you have sorted the csv-File.
After Sorting, you want a complete Block of Entried. For example, if you have this File:

	A	B	C	D	E	F	G	H	I	J	K	L
1	993 GLT	Angela	Merkel	Angela Merk	F	Politician	Federal Char	Chancellor o	Political &	Government		Germany
2	1993 GLT	Claudia	Nolte (Crawf	Claudia Nolt	F	Politician	CDU	Ex-Federal M	Political &	Government		Germany
3	1993 GLT	Jan Maria	Rokita	Jan Rokita	M	Politician	Parliament c	Former Mem	Political &	Government		Poland
4	1994 GLT	Aleksander	Kwasniewsk	Aleksander K	M	Politician	Government	Government	Political &	Government	Gdansk	Poland
5	1995 GLT	Waldemar	Pawlak	Waldemar Pa	M	Politician	Government	Prime Minist	Political & G	Deputy Prime Minister an		Poland
6	1995 GLT	Christian	Wulff	Christian Wu	M	Politician	Office of Pre	President of	Political &	Government		Germany
7	2005 YGL	Silvana	Koch-Mehrin	Silvana Koch	F	Politician	European Pa	CEO of Wom	Political & G	Former MEP, EU Represer		Germany

then you have in Column "L" a block of 3 Entries which are from Poland.
The `TXT.Fic` will find the most upper of these entries, starting from a given Start-Line.

```
' This is the short version of the Sample Script, the full version see below.
$$IND="L"
$$SEA="Poland"
$$STA=1
$$DEL=;
' We use Stadard Search Pattern
TXT.fic|$$IND|$$STA|$$SEA|$$DEL|w
VAR.$$RES=$$000
MBX.$$RES
ENR.
```



Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

TXT.Fic|P1|P2[|P3][|P4][|P5]
TXT.Find_first_in_Column|P1|P2[|P3][|P4][|P5]

Parameter Explanation

P1 **- (Input)** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
Can be `"C" or "AA"` - must be a Single Item.
P2 **- (Input)** Variable with Startline, default is 2 (to skip an Header).
P3 **- (Input)** Variable with one or more Words to compare to. These must be quoted and separated with ",".
Example: `"Headline","Party","Column"`
P4 **- (Input)** Delimiter, if omitted Semicolon is used.
P5 **- (Input)** **Flag**, can be any of these: "**p**" (default) - normal compare, "**w**" - compare using SPR-Standard-Search Patterns,
"**e**" - Equalcase Compare, "**c**" - contains, "**ce**" - contains equalcase

Example

```
'*****************************************************
'
'*****************************************************

  $$TMA=#dtime#
  IEM.
    VAR.$$FIA=$cmdexe$
  ELS.
    VAR.$$FIA=?exeloc\Sources\Blocktest.CSV
  EIF.
  VAR.$$OUF=?exeloc\Result_#date#.csv
  NEF.$$FIA
    $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
    GTO.ero
  EIF.
  '-----------------------------------------------------------
  TXT.Load_Textfile|$$FIA|$$LIC
  ' Convert $LF to $crlf$ if needed.
  'TXT.lfp
  TXT.Get_Line_Count|$$LIC
  IVV.$$LIC<5
    $$ERR=File is empty?
    GTO.ero
  EIF.
  PRT.$$LIC Lines loaded.
  '++++++++++++++++++++++++++++++
  '++++++++++++++++++++++++++++++
  ' Place your Script here
  $$IND="L"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$SEA="Poland"
$$STA=1
$$DEL=;
' We use Stadard Search Pattern
TXT.fic|$$IND|$$STA|$$SEA|$$DEL|w
VAR.$$RES=$$000
MBX.$$RES
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'#############################################################################
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.6 Find_last_in_Column

MiniRobotLanguage (MRL)

TXT.Flc
TXT.Find_last_in_Column

Find the first Entry in a specified column with a specified value.

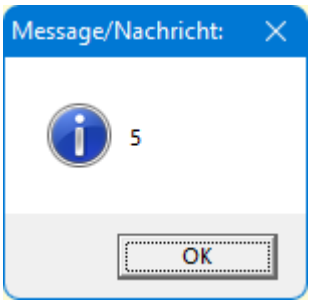
Intention

This command is useful if you have to do with csv-files.
Most often this can be used after you have sorted the csv-File.
After Sorting, you want a complete Block of Entried. For example, if you have this File:

	A	B	C	D	E	F	G	H	I	J	K	L
1	993 GLT	Angela	Merkel	Angela Merk	F	Politician	Federal Char	Chancellor o	Political &	Government		Germany
2	1993 GLT	Claudia	Nolte (Crawf	Claudia Nolt	F	Politician	CDU	Ex-Federal M	Political &	Government		Germany
3	1993 GLT	Jan Maria	Rokita	Jan Rokita	M	Politician	Parliament c	Former Mem	Political &	Government		Poland
4	1994 GLT	Aleksander	Kwasniewsk	Aleksander K	M	Politician	Government	Government	Political &	Government	Gdansk	Poland
5	1995 GLT	Waldemar	Pawlak	Waldemar Pa	M	Politician	Government	Prime Minist	Political & G	Deputy Prime Minister an		Poland
6	1995 GLT	Christian	Wulff	Christian Wu	M	Politician	Office of Pre	President of	Political &	Government		Germany
7	2005 YGL	Silvana	Koch-Mehrin	Silvana Koch	F	Politician	European Pa	CEO of Wom	Political & G	Former MEP, EU Represer		Germany

then you have in Column "L" a block of 3 Entries which are from Poland.
The `TXT.Fic` will find the most upper of these entries, starting from a given Start-Line.

```
' This is the short version of the Sample Script, the full version see below.
$$IND="L"
$$SEA="Poland"
$$STA=1
$$DEL=;
' We use Stadard Search Pattern
TXT.flc|$$IND|$$STA|$$SEA|$$DEL|w
VAR.$$RES=$$000
MBX.$$RES
ENR.
```



Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

TXT.Flc|P1|P2[|P3][|P4][|P5]
TXT.Find_last_in_Column|P1|P2[|P3][|P4][|P5]

Parameter Explanation

P1 **- (Input)** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
Can be `"C" or "AA"` - must be a Single Item.

P2 **- (Input)** Variable with Startline, default is 2 (to skip an Header).

P3 **- (Input)** Variable with one or more Words to compare to. These must be quoted and separated with ",".
Example: `"Headline","Party","Column"`

P4 **- (Input)** Delimiter, if omitted Semicolon is used.

P5 **- (Input)** **Flag**, can be any of these: "**p**" (default) - normal compare, "**w**" - compare using SPR-Standard-Search Patterns,
"**e**" - Equalcase Compare, "**c**" - contains, "**ce**" - contains equalcase

Example

```
'*****************************************************
'
'*****************************************************

  $$TMA=#dtime#
  IEM.
    VAR.$$FIA=$cmdexe$
  ELS.
    VAR.$$FIA=?exeloc\Sources\Blocktest.CSV
  EIF.
  VAR.$$OUF=?exeloc\Result_#date#.csv
  NEF.$$FIA
    $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
    GTO.ero
  EIF.
  '-----------------------------------------------------------
  TXT.Load_Textfile|$$FIA|$$LIC
  ' Convert $LF to $crlf$ if needed.
  'TXT.lfp
  TXT.Get_Line_Count|$$LIC
  IVV.$$LIC<5
    $$ERR=File is empty?
    GTO.ero
  EIF.
  PRT.$$LIC Lines loaded.
  '++++++++++++++++++++++++++++++
  '++++++++++++++++++++++++++++++
  ' Place your Script here
  $$IND="L"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$SEA="Poland"
$$STA=1
$$DEL=;
' We use Stadard Search Pattern
TXT.flc|$$IND|$$STA|$$SEA|$$DEL|w
VAR.$$RES=$$000
MBX.$$RES
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'#############################################################################
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.7 Get_any_Delimited_Item_from_Line

MiniRobotLanguage (MRL)

TXT.gdal
TXT.Get_any_delimited_Item_from_Line

Use a Line with the specified Number from the TXT.-internal Buffer and extract delimited Items from that Line.
Here multiple Single-Character Delimiter-characters can be specified.

Intention

This command is useful if you have to do with csv-files.
This Command can be used to work on a "Line by line" base through a file working with delimited Items.
The Command will extract single delimited Items but can also be used to extract multiple Items in one run.
The Items will be returned in their normal order, no Matter in which order the Items are specified.
Internally this Command works somehow (not exactly) like DIC.Get_in_Order 3088.

```
' Assume Line 10 Line could be AA;BB;CC;DD;EE;FF
VAN.$$LIN=10
VAR.$$DEL=;
VAR.$$IND="A","E",2
TXT.get_any_delimited_item_from_line|$$LIN|$$DEL|$$IND|$$RES
DBP.$$RES

' Result:
' BB;DD;FF
```

Syntax

TXT.gdal|P1|P2|P3[|P4]
TXT.Get_any_delimited_Item_from_Line|P1|P2|P3[|P4]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Input)** Variable with delimiter. Delimiter can be one or multiple characters.
P3 - **(Input)** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
The Items will be returned in their normal order, no Matter in which order the Items are specified.

```
1,3,"C","AA"
```

P4 - **(Output)** Variable for Result

Example

```
'*****************************************************
'
'*****************************************************
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.8 Get_any_Delimited_Item_from_Text

MiniRobotLanguage (MRL)

TXT.gdat
TXT.Get_any_delimited_Item_from_Text

Use the complete TXT.-internal Buffer and extract delimited Items from there. Here multiple Single-Character Delimiter-characters can be specified.

Intention

This command is useful if you have to do with csv-files.
This Command can be used to work on a "Line by line" base through a file working with delimited Items.
The Command will extract single delimited Items but can also be used to extract multiple Items in one run.
The Items will be returned in their normal order, no Matter in which order the Items are specified.
Internally this Command works somehow (not exactly) like DIC.Get_in_Order 3088.

```
' Assume Line 10 Line could be AA;BB;CC;DD;EE;FF
VAN.$$LIN=10
VAR.$$DEL=;
VAR.$$IND="A","E",2
TXT.get_any_delimited_item_from_line|$$LIN|$$DEL|$$IND|$$RES
DBP.$$RES

' Result:
' BB;DD;FF
```

Syntax

TXT.gdat|P1|P2[|P3]
TXT.Get_any_delimited_Item_from_Text|P1|P2[|P3]

Parameter Explanation

P1 **-** **(Input)** Variable with delimiter. Delimiter can be one or multiple characters.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **-** **`(Input)`** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
The Items will be returned in their normal order, no Matter in which order the Items are specified.

```
1,3,"C","AA"
```

P3 **-** **`(Output)`** Variable for Result

Example

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.9 Get_Column_Block

MiniRobotLanguage (MRL)

TXT.Gcb
TXT.Get_Column_Block

Find the first Entry in a specified column with a specified value.

Intention

This command is useful if you have to do with csv-files.
Most often this can be used after you have sorted the csv-File.
After Sorting, you want a complete Block of Entried. For example, if you have this File:

	A	B	C	D	E	F	G	H	I	J	K	L
1	993 GLT	Angela	Merkel	Angela Merk	F	Politician	Federal Char	Chancellor o	Political & Government			Germany
2	1993 GLT	Claudia	Nolte (Crawf	Claudia Nolt	F	Politician	CDU	Ex-Federal M	Political & Government			Germany
3	1993 GLT	Jan Maria	Rokita	Jan Rokita	M	Politician	Parliament c	Former Mem	Political & Government			Poland
4	1994 GLT	Aleksander	Kwasniewsk	Aleksander K	M	Politician	Government	Government	Political & Government		Gdansk	Poland
5	1995 GLT	Waldemar	Pawlak	Waldemar Pa	M	Politician	Government	Prime Minist	Political & Gi	Deputy Prime Minister an		Poland
6	1995 GLT	Christian	Wulff	Christian Wu	M	Politician	Office of Pre	President of	Political & Government			Germany
7	2005 YGL	Silvana	Koch-Mehrin	Silvana Koch	F	Politician	European Pa	CEO of Wom	Political & Gi	Former MEP, EU Represer		Germany

then you have in Column "L" a block of 3 Entries which are from Poland.
The `TXT.gcb` will give you the complete Block from the first to the last found entry with the given Specifications.

```
' This is the short version of the Sample Script, the full version see below.
$$IND="L"
$$SEA="Poland"
$$STA=1
$$DEL=;
' We use Standard Search Pattern
TXT.gcb|$$IND|$$STA|$$SEA|$$DEL|w
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1993 GLT	Jan Maria	Rokita	Jan Rokita	M	Politician	Parliament c	Former Mem	Political & Government			Poland	Europe
2	1994 GLT	Aleksander	Kwasniewsk	Aleksander K	M	Politician	Government	Government	Political & Government		Gdansk	Poland	Europe
3	1995 GLT	Waldemar	Pawlak	Waldemar Pa	M	Politician	Government	Prime Minist	Political & Gi	Deputy Prime Minister an		Poland	Europe

Syntax

TXT.Gcb|P1|P2[|P3][|P4][|P5]
TXT.Get_Column_Block|P1|P2[|P3][|P4][|P5]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **- (Input)** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
Can be "C" or "AA" - must be a Single Item.

P2 **- (Input)** Variable with Startline, default is 2 (to skip an Header).

P3 **- (Input)** Variable with **one or more** Words to compare to. These must be quoted and separated with ",".
Example: "Headline","Party","Column"

P4 **- (Input)** Delimiter, if omitted Semicolon is used.

P5 **- (Input)** **Flag**, can be any of these: "**p**" (default) - normal compare, "**w**" - compare using SPR-Standard-Search Patterns,
"**e**" - Equalcase Compare, "**c**" - contains, "**ce**" - contains equalcase

Example

```
'******************************************************
'
'******************************************************

  $$TMA=#dtime#
  IEM.
    VAR.$$FIA=$cmdexe$
  ELS.
    VAR.$$FIA=?exeloc\Sources\Blocktest.CSV
  EIF.
  VAR.$$OUF=?exeloc\Result_#date#.csv
  NEF.$$FIA
    $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
    GTO.ero
  EIF.
  '------------------------------------------------------------
  TXT.Load_Textfile|$$FIA|$$LIC
  ' Convert $LF to $crlf$ if needed.
  'TXT.lfp
  TXT.Get_Line_Count|$$LIC
  IVV.$$LIC<5
    $$ERR=File is empty?
    GTO.ero
  EIF.
  PRT.$$LIC Lines loaded.
  '++++++++++++++++++++++++++++++++
  '++++++++++++++++++++++++++++++++
  ' Place your Script here
  $$IND="L"
  $$SEA="Poland"
  $$STA=1
  $$DEL=;
  ' We use Standard Search Pattern
  TXT.gcb|$$IND|$$STA|$$SEA|$$DEL|w
  '++++++++++++++++++++++++++++++++
  '++++++++++++++++++++++++++++++++
  ' These lines process and save the result.
  'TXT.lfr
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'############################################################################
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.10 Get_Delimited_Item_from_Line

MiniRobotLanguage (MRL)

TXT.gdil
TXT.Get_delimited_Item_from_Line

Use a Line with the specified Number from the TXT.-internal Buffer and extract delimited Items from that Line.
Here a multicharacter delimiter can be specified.

Intention

This command is useful if you have to do with csv-files.
This Command can be used to work on a "Line by line" base through a file working with delimited Items.
The Command will extract single delimited Items but can also be used to extract multiple Items in one run.
The Items will be returned in their normal order, no Matter in which order the Items are specified.
Internally this Command works like DIC.Get_in_Order 3088.

```
' Assume Line 10 Line could be AA;BB;CC;DD;EE;FF
VAN.$$LIN=10
VAR.$$DEL=;
VAR.$$IND="A","E",2
TXT.get_delimited_item_from_line|$$LIN|$$DEL|$$IND|$$RES
DBP.$$RES

' Result:
' BB;DD;FF
```

Syntax

TXT.gdil|P1|P2|P3[|P4]
TXT.Get_delimited_Item_from_Line|P1|P2|P3[|P4]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** **`(Input,numeric)`** Variable with Line-Number.
P2 **-** **`(Input)`** Variable with delimiter
P3 **-** **`(Input)`** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
The Items will be returned in their normal order, no Matter in which order the Items are specified.

```
1,3,"C","AA"
```

P4 **-** **`(Output)`** Variable for Result

Example

```
'*****************************************************
'
'*****************************************************
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.11 Get_Delimited_Item_from_Text

MiniRobotLanguage (MRL)

TXT.gdit
TXT.Get_delimited_Item_from_Text

Use the complete TXT.-internal Buffer and extract delimited Items from there. Here a multicharacter delimiter can be specified.

Intention

This command is useful if you have to do with csv-files.
This Command can be used to work on a "Line by line" base through a file working with delimited Items.
The Command will extract single delimited Items but can also be used to extract multiple Items in one run.
The Items will be returned in their normal order, no Matter in which order the Items are specified.
Internally this Command works like DIC.Get_in_Order 3088 .

```
' Assume the Textbuffer contains:"AA;BB;CC;DD;EE;FF"
VAN.$$LIN=10
VAR.$$DEL=;
VAR.$$IND="A","E",2
TXT.get_delimited_item_from_text|$$LIN|$$DEL|$$IND|$$RES
DBP.$$RES

' Result:
' BB;DD;FF
```

Syntax

TXT.gdit|P1|P2[|P3]
TXT.Get_delimited_Item_from_Text|P1|P2[|P3]

Parameter Explanation

P1 **-** **(Input)** Variable with delimiter

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **-** **`(Input)`** Variable with Index of item to get. Can also be a List of Items to get. The List can contain whether the Column-Number or the Letter in "".
The Items will be returned in their normal order, no Matter in which order the Items are specified.

```
1,3,"C","AA"
```

P3 **-** **`(Output)`** Variable for Result

Example

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.12 Keep_Columns

MiniRobotLanguage (MRL)

TXT.kec
TXT.Keep_Columns

Get Line with the specified Number from TXT.-internal Buffer into Variable.

Intention

This command is used to extract Columns from out of a csv-file.
You can use the Minus-Sign to specify a Range like this: "A-C" or "2-AC".
The columns will stay in their order, with those not specified, removed.
The original Buffer-COntent wil lautomati

```
$$TXA=A;B;C;D;E;F$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB
$$IND="2-4"
$$DEL=;
TXT.kec|$$IND|$$DEL
DBP.Main Buffer
TXT.dbp
ENR.

' Result:
B;C;D
B;C;D
B;C;D
B;C;D
```

```
' Generate 5 Lines of Text A
$$TXA=A;B;C;D;E;F;G;H;F$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB
$$IND="1","3-5"
$$DEL=;
TXT.kec|$$IND|$$DEL
DBP.Main Buffer
TXT.dbp
ENR.


A;C;D;E
A;C;D;E
A;C;D;E
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
A;C;D;E
```

Syntax

TXT.kec|P1[|P2]
TXT.Keep_Columns|P1[|P2]

Parameter Explanation

P1 **-** **`(Input,numeric)`** List with Columns to keep. Looks like this: `1,3,5,"A",B-E"`. Single Numbers do not necessary need to use `""`.
P2 **-** **`(Output)`** Variable will contain the delimiter. If omitted, the default delimiter is the semicolon.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gel
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
CFF.$$LAA|$$RES
TXT.lpt|$$RES|$$LAA
' Now we get only Line 5
TXT.gel|5|$$LIN
MBX.Line 5 is: $$LIN
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.13 Repair_CSV

MiniRobotLanguage (MRL)

TXT.Rer
TXT.Repair_CSV

Repair loaded CSV-File, especially fixe opened " " that have not been closed.

Intention

This command is used repair CSV-Files.Sometime you may see that parts of a CSV-File are not displayed when loaded into Excel etc.
This can happen, if there are opened, and not closed "" inside the file.
This command will either close the "" or remove them if they are open.
If **P2** i snot "0" then it will also correct the number of used delimiters per Line to this specified number.

```
' Repair_csv
$$DEL=;
$$ANZ=7
TXT.Rer|$$DEL|$$ANZ
```

Syntax

TXT.Rer|P1[|P2]
TXT.Repair_CSV|P1[|P2]

Parameter Explanation

P1 - **`(Input,numeric)`** Variable with Delimiter-Character.
P2 - **`(Output)`** Variable will contain the number of delimiters per Line, or "0". If given lines with too less delimiters will be completed by adding delimiter characters.

Example

```
'*****************************************************
'
'*****************************************************
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.14 Sort_by_Column_Float

MiniRobotLanguage (MRL)

TXT.Sbcf
TXT.Sort_by_Column_Float

Treat the internal Buffer as a csv-file and Sort the File "by Column"

Intention

This command is used to work with csv-Files. It will Sort the internal Buffer "by a Column".
Float-Numbers must be treated in a special way to get a correct result.
For example, 4,00E-02 is the same like 0.004 and would alphabetically or even as Integer not properly be sorted.
The same applies to 2.00E+04 which equals 4000 and will also be sorted in the right way using this command.
Here we are going to sort by Column "D", ignoring the Headline.
The german separator "," (e.g. **2,5**) will be replaced with the international "." (e.g. **2.5**) before sorting.

Important: All Data here is fictional and just for demonstrating the SPR-Commands.

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ulrike	Nieter	– no bio	0,8	Germany
3	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

```
$$IND="A"
' Start in Line 2 to keep the header
$$STA=2
TXT.sort_by_column_Float|$$STA|$$IND|;|a
```

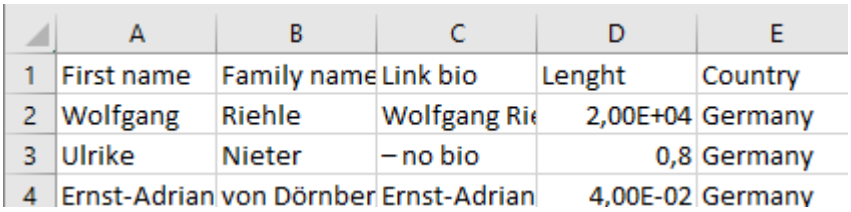
Result:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
3	Ulrike	Nieter	– no bio	0,8	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

```
$$IND="A"
' Start in Line 2
$$STA=2
$$IND="D"
TXT.sort_by_column_Float|$$STA|$$IND|;|d
```


© 2023 Theo Gottwald * https://www.smart-package.com

Result:



	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany
3	Ulrike	Nieter	– no bio	0,8	Germany
4	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany

This command has been optimized for Speed. Even large Files with 40 MB in Size should be sorted within 1-2 Seconds or less.

Syntax

TXT.Sbcf|P1|P2[|P3][|P4]
TXT.Sort_by_Column_Float|P1|P2[|P3][|P4]

Parameter Explanation

P1 **- (Input,numeric)** Start-Line, because in Line 1 is most often a Header, StartI_line is most often and per default Line 2.
P2 **- (Input)** Sorting Column. Something like a number like **3** or a letter in **"C"**.
P3 **- (Input)** Variable with delimiter. Delimiter can be one or multiple characters. If omitted **;** is used.
P4 **- (Input) Flags** a/d/l - **A**scending / **D**escending sort, **L** - equalcase sort (uses Lowercase internal)

Example

```
'*****************************************************
' Template for Testing TXT.-Commands
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
' put your Filename here
  VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="A"
TXT.sort_by_column_letter|2|$$IND|;|d

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000
CAL.$$LIF=($$LIC*1000)/$$TMA|i
$$TXA=$crlf$Processing $$LIF Lines per Second.
IVV.$$TMA>1000
  CAL.$$TMA=$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.$$TXA
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.$$TXA
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'#############################################################
##########
```

Remarks

Optimized for speed. Can sort very large files in a Second..
`$crlf$` (CarriageReturn/LineFeed) at the end of the loaded File will be stripped before Sorting.
This was necessary to prevent empty lines to get sorted into the result.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.4.15 Sort_by_Column_Integer

MiniRobotLanguage (MRL)

TXT.Sbci
TXT.Sort_by_Column_Integer

Treat the internal Buffer as a csv-file and Sort the File "by Column"

Intention

This command is used to work with csv-Files. It will Sort the internal Buffer "by a Column".
Integer Numbers must be treated in a special way to get a correct result.
For example, 900 would alphabetically be sorted after 1000 in a pure "Letter sort", therefore this command is the solution.
Here we are going to sort by Column "A", ignoring the Headline.

Important: All Data here is fictional and just for demonstrating the SPR-Commands.

	A	B	C	D	E
1	First name	Family name	Link bio	Salary	Country
2	Wolfgang	Riehle	Wolfgang Rie	9000	Germany
3	Ulrike	Nieter	– no bio	1200	Germany
4	Ernst-Adrian	von Dörnber	Ernst-Adrian	79000	Germany

```
$$IND="A"
' Start in Line 2 to keep the header
$$STA=2
TXT.sort_by_column_Integer|$$STA|$$IND|;|a
```

Result:

	A	B	C	D	E
1	First name	Family name	Link bio	Salary	Country
2	Ulrike	Nieter	– no bio	1200	Germany
3	Wolfgang	Riehle	Wolfgang Rie	9000	Germany
4	Ernst-Adrian	von Dörnber	Ernst-Adrian	79000	Germany

```
$$IND="A"
' Start in Line 2
$$STA=2
$$IND="D"
TXT.sort_by_column_Integer|$$STA|$$IND|;|d
```

Result:

© 2023 Theo Gottwald * https://www.smart-package.com

	A	B	C	D	E
1	First name	Family name	Link bio	Salary	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	79000	Germany
3	Wolfgang	Riehle	Wolfgang Rie	9000	Germany
4	Ulrike	Nieter	– no bio	1200	Germany

This command has been optimized for Speed. Even large Files with 40 MB in Size should be sorted within 1-2 Seconds or less.

Syntax

TXT.Sbci|P1|P2[|P3][|P4]
TXT.Sort_by_Column_Integer|P1|P2[|P3][|P4]

Parameter Explanation

P1 **- `(Input,numeric)`** Start-Line, because in Line 1 is most often a Header, StartI_line is most often and per default Line 2.
P2 **- `(Input)`** Sorting Column. Something like a number like **3** or a letter in **"C"**.
P3 **- `(Input)`** Variable with delimiter. Delimiter can be one or multiple characters. If omitted **;** is used.
P4 **- `(Input) Flags`** a/d/l - **A**scending / **D**escending sort, **L** - equalcase sort (uses Lowercase internal)

Example

```
'*****************************************************
' Template for Testing TXT.-Commands
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
' put your Filename here
  VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GTO.ero
EIF.
'-------------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
' Place your Script here

$$IND="A"
TXT.sort_by_column_letter|2|$$IND|;|d

'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000
CAL.$$LIF=($$LIC*1000)/$$TMA|i
$$TXA=$crlf$Processing $$LIF Lines per Second.
IVV.$$TMA>1000
  CAL.$$TMA=$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.$$TXA
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.$$TXA
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'#############################################################
##########
```

Remarks

Optimized for speed. Can sort very large files in a Second..
`$crlf$` (CarriageReturn/LineFeed) at the end of the loaded File will be stripped before Sorting.
This was necessary to prevent empty lines to get sorted into the result.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.80.4.16 Sort_by_Column_Letter

MiniRobotLanguage (MRL)

TXT.Sbcl
TXT.Sort_by_Column_Letter

Treat the internal Buffer as a csv-file and Sort the File "by Column"

Intention

This command is used to work with csv-Files. It will Sort the internal Buffer "by a Column".
Here we are going to sort by Column "A", ignoring the Headline.

	A	B	C	D	E
	First name	Family name	Link bio	Sex	Country
2	Ulrike	Nieter	– no bio	F	Germany
3	Ernst-Adrian	von Dörnber	Ernst-Adrian	M	Germany
4	Wolfgang	Riehle	Wolfgang Rie	M	Germany

```
$$IND="A"
' Start in Line 2 to keep the header
$$STA=2
TXT.sort_by_column_letter|$$STA|$$IND|;|a
```

Result:

	A	B	C	D	E
1	First name	Family name	Link bio	Sex	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	M	Germany
3	Ulrike	Nieter	– no bio	F	Germany
4	Wolfgang	Riehle	Wolfgang Rie	M	Germany

```
$$IND="A"
' Start in Line 2
$$STA=2
TXT.sort_by_column_letter|$$STA|$$IND|;|d
```

Result:

	A	B	C	D	E
1	First name	Family name	Link bio	Sex	Country
2	Wolfgang	Riehle	Wolfgang Rie	M	Germany
3	Ulrike	Nieter	– no bio	F	Germany
4	Ernst-Adrian	von Dörnber	Ernst-Adrian	M	Germany

© 2023 Theo Gottwald * https://www.smart-package.com

This command has been optimized for Speed. Even large Files with 40 MB in Size should be sorted within 1-2 Seconds or less.
I have tested with a file that contained >350.000 Lines and it sorted instantly.

Syntax

TXT.Sbcl|P1|P2[|P3][|P4]
TXT.Sort_by_Column_Letter|P1|P2[|P3][|P4]

Parameter Explanation

P1 **- (Input,numeric)** Start-Line, because in Line 1 is most often a Header, StartI_line is most often and per default Line 2.
P2 **- (Input)** Sorting Column. Something like a number like **3** or a letter in **"C"**.
P3 **- (Input)** Variable with delimiter. Delimiter can be one or multiple characters. If omitted **;** is used.
P4 **- (Input) Flags** a/d/l - **A**scending / **D**escending sort, **L** - equalcase sort (uses Lowercase internal)

Example

```
'*****************************************************
' Template for Testing TXT.-Commands
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
' put your Filename here
  VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
' Place your Script here

$$IND="A"
TXT.sort_by_column_letter|2|$$IND|;|d

'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000
CAL.$$LIF=($$LIC*1000)/$$TMA|i
$$TXA=$crlf$Processing $$LIF Lines per Second.
IVV.$$TMA>1000
  CAL.$$TMA=$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.$$TXA
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.$$TXA
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
'############################################################
##########
```

Remarks

Optimized for speed. Can sort very large files in a Second..
`$crlf$` (CarriageReturn/LineFeed) at the end of the loaded File will be stripped before Sorting.
This was necessary to prevent empty lines to get sorted into the result.

Limitations:

-

See also:

- TXT.Load Textfile 3418

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.80.5 Find- and Replace Operations

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.5.1 Find_Line_with

MiniRobotLanguage (MRL)

TXT.flw
TXT.Find_Line_with

Search a Text in the TXT.-internal Buffer and return Line-Number and Byte-Position where it was found.

Intention

This is the simple search, it will just return the most needed results.
This command is used to get the Line-Number and Byte-Positions of a Search-String,
that was found in the Text that is in the Internal TXT-Buffer, into Variables.
As slightly faster Alternative to `TXT.flw`, this command will return all needed Information to replace the text or work with it in any way.

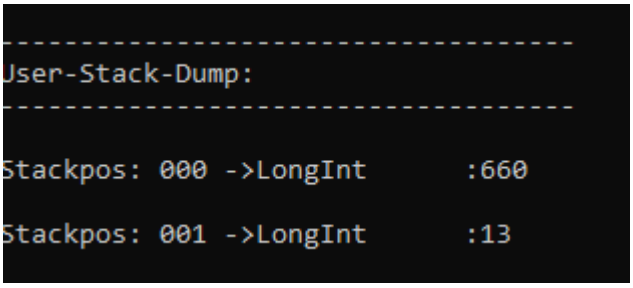
```
$$SEA=SWP.
$$LAB=0
TXT.flw|$$SEA|0|$$RES|$$LAB
DMP.6
MBX.$$RES-$$LAB
ENR.
```

If you use the command without Output-Variables, the result will be on the stack in reverse order.

```
TXT.flw|$$SEA
```

What you see here is:

TOS:	Byte-Position of found Text in whole Text
TOS+1:	Line-Number where the Text was found



Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

TXT.flw|P1[|P2][|P3][|P4]
TXT.Find_Line_with|P1[|P2][|P3][|P4]

Parameter Explanation

P1 **-** **`(Input)`** Variable with String to search for.
P2 **-** **`(numeric)`** Variable will contain a number or 0, this the Starting-Line for the search. The search will at the first byte of the Line.
If this is 0 or 1, the complete loaded Textfile will be searched.
P3 **-** **`(Output)`** Variable will contain the Line-Number of the Line that contains the Search String.
If omitted, the result is placed on TOS.
P4 **-** **`(Output)`** Variable will contain the Byte-Position of the fist-Byte of the Search string in the Text.
If omitted, the result is placed on TOS.

Example

```
'******************************************************
' EXAMPLE 1:
'******************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
VAN.$$LAC=0
$$SEA=SWP.
$$LAB=0
TXT.flw|$$SEA|0|$$RES|$$LAB
JIV.$$RES!13|Lab_failed
JIV.$$LAB!660|Lab_failed
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015

© 2023 Theo Gottwald * https://www.smart-package.com

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.5.2 Find_Text

MiniRobotLanguage (MRL)

TXT.fit
TXT.Find_Text

Search a Text in the TXT.-internal Buffer and return various Information where it was found.
Can also do a reverse search, see P2.

Intention

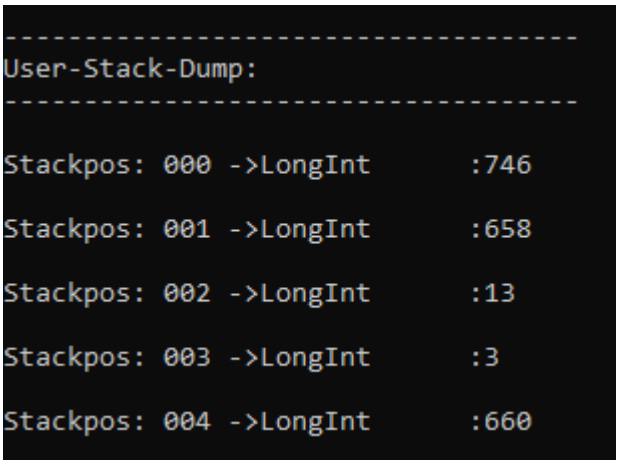
This is the extended search command. it will return detailled results.
This command is used to get the Byte-Positions of Start- and End of a specified Line, from the Text that is in the Internal TXT-Buffer into Variables.
Unlike TXT.glp, it works from a Byte-Position. So you say "Byte 42" and the command will return the start- and End-Byte of the line,
that contains Byte 43. Variable **P2** will contain the Start-Byte of the Line. Variable **P3** will contain the Last-Byte of the Line before and excluding the `$crlf$`.
This Command can be used to work on a Byte-Base through a file without accessing the file on the harddrive.

```
$$SEA=SWP.
$$LAB=0
TXT.fit|$$SEA|0|$$RES|$$LAB
DMP.6
MBX.Was found at Byte-Pos: $$RES$crlf$ Byte-Position in the
Line:$$LAB
ENR.


TXT.fit|$$SEA
```

So you get following Information:

TOS: End-Byte of Line with the found text
TOS+1: Start-Byte of Line with the found text
TOS+2: Linenumber where the Text was found, or "0" if not found
TOS+3: Byte-position of the found-Text in the Line (Text starts here at character 3 in the Line 13)
TOS+4: Byte-Position of the found Text in the whole-text

*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

TXT.fit|P1[|P2][|P3][|P4][|P5][|P6][|P7]
TXT.Find_Text|P1[|P2][|P3][|P4][|P5][|P6][|P7]

Parameter Explanation

P1 - **(Input)** Variable with String to search for.
P2 - **(numeric or string)** Starting-Position and Flags "**B**" (Byte) and "**L**" (Line). If omitted "L" is taken as default.
If "**R**" is specified this will activate "Reverse Search".
Variable will contain a number or 0 can also contain "L" or "B" or "R",
This the Starting-Line Number for the search. The search will start at the first byte of the specified Line.
If this contains a letter "**B**" then the number will be taken as a Byte-Position, where to start the search.
If this contains a letter "**R**" then a reverse search is performed.
If the number is a Byte-Position (using "B"), it will be counted counted from the end of the file and it will perform a Reverse search,
from the end of the file towards the start.
If the number is a Linenumber, the linenumber will also be reversed, eg. the last line is Line 1.
If this is 0 or 1, the complete loaded Textfile will be searched.
If no letter is given, the default way to interpret this number is "L" (as Linenumber).

P3 - **(Output)** Variable will contain the Byte-Position of the Search String in the whole Text.If omitted, the result is placed on TOS.
If this result is "0" nothing was found.

P4 - **(Output)** Variable will contain the Byte-Position of the Search String in the Line where it was found.
If omitted, the result is placed on TOS. If this result is "0" nothing was found.

© 2023 Theo Gottwald * https://www.smart-package.com

P5 **-** **(Output)** Variable will contain the Line-Number of the Line that contains the Search String or 0 if nothing was found.
If omitted, the result is placed on TOS.

P6 **-** **(Output)** Variable will contain the Byte-Position of the fist-Byte of the Line where the Search string was found.
If omitted, the result is placed on TOS.

P7 **-** **(Output)** Variable will contain the Byte-Position of the Last-Byte of the Line where the Search string was found.
Does not include the `$crlf$` at the End of the Line.If omitted, the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
VAN.$$LAC=0
$$SEA=SWP.
$$LAB=0
TXT.fit|$$SEA|0|$$RES|$$LAB
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.5.3 Replace

MiniRobotLanguage (MRL)

TXT.rep
TXT.Replace

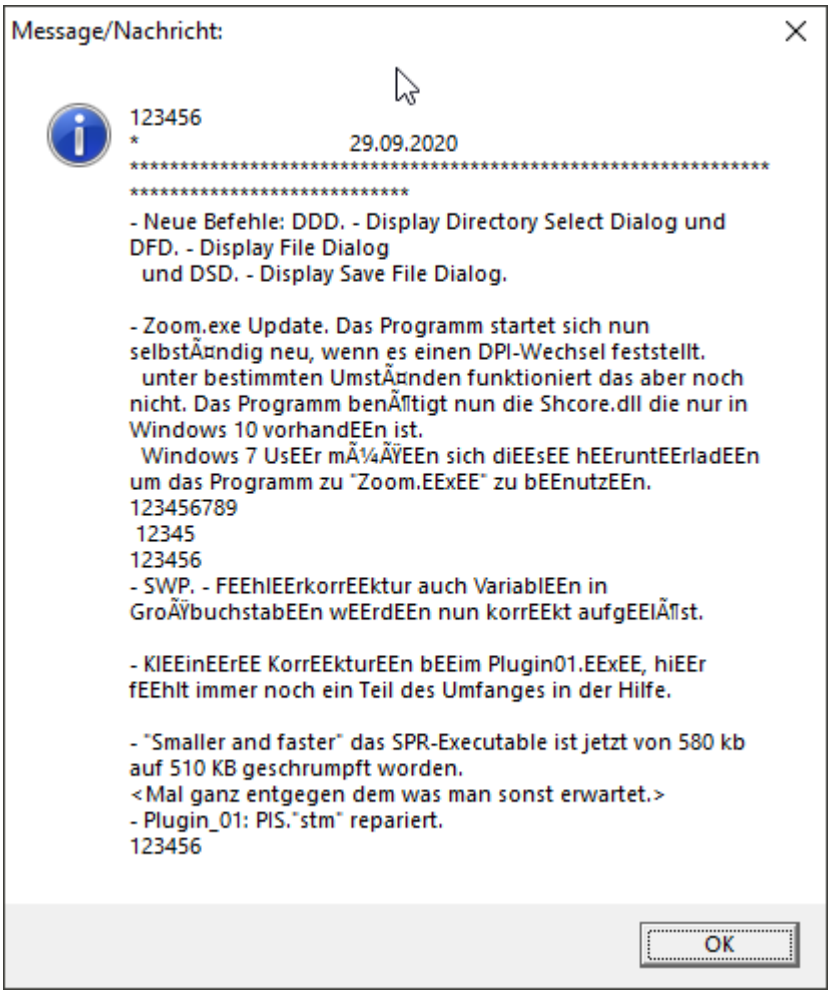
Search a Text in the TXT.-internal Buffer and replace it with a given other sequence.

Intention

This command is used to replace all occurrences of a Search-String in the TXT.-Buffer with another String.

```
'Replace all "e" with "ef" in internal TXT.-Buffer
VAR.$$LAB=e
VAR.$$LAC=ef
TXT.rep|$$LAB|$$LAC
ENR.

VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' This example shows all features of TXT.rep
' We are running in 2 times so its clear that as this is NOT
an Equalcase-Replace,
' the "EE" will not be treated as "ee" and therefore also not
replaced.
' This is teh differnece to TXT.rit
TXT.rep|e|EE|520|800
TXT.rep|e|EE|520|800
' You can also see that the replace starts a Position 520, as
specified and ends at Position 800.
' So the start of the text is unchanged and also the end.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

```
TXT.rep|$$SEA|$$REP[|$$STA]
[|$$END]
TXT.Replace|P1|P2[|P3][|P4]
```

Parameter Explanation

P1 – **`(Input)`** Variable with String to search for.
P2 – **`(Input)`** Variable with the String that shall replace the String in **P1**.
P3 – **`(`***optional, numeric***`)`** Variable with a Byte-Position in the TXT.-Buffer, if given the replace will start at that Position.
If omitted the Text is been searched and replaced from Position 1.
P4 – **`(`***optional, numeric***`)`** Variable with a Byte-Position in the TXT.-Buffer, if given the replace will end at that Position.
If omitted the complete Text is been searched and replaced to the end.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
VAN.$$STA=100
'Replace all "e" with "ef" in internal TXT.-Buffer starting
from Byte-Position 100
VAR.$$OLD=e
VAR.$$NEW=ef
TXT.rep|$$OLD|$$NEW|$$STA
ENR.
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Replace_in_Text 3344
- TXT.Replace 3338
- TXT.Replace_in_Line 3338
- Push_Intern 3444
- Pop_Intern 3338
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.5.4 Replace_in_Line

MiniRobotLanguage (MRL)

TXT.ril
TXT.Replace_in_Line

Search a Text in the TXT.-internal Buffer in a specified Line and replace just the first match with a given other string.

Intention

This command is specific in that it will only replace the first match of **P2** in the specified Line with **P3**.
Also you can specify a starting Position in **P4**.
If you want to replace all occurrences in a Line, you can get the Line-Position using TXT.glp and then use TXT.rep.
Or just call TXT.ril multiple TImes until the TOS returns "0".
The command will leave a "0" or a "1" on the TOS, depending if there was a replace or not.

```
'Replace all "e" with "ef" in internal TXT.-Buffer
VAR.$$LAB=e
VAR.$$LAC=ef
TXT.ril|13|$$LAB|$$LAC|15
ENR.
```

```
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' This example shows all features of TXT.ril
' We are running in 2 times so its clear that due to the
Equalcase-Replace,
' also the "EE" will be treated as "ee" and replaced.
TXT.ril|e|EE|520
TXT.ril|e|EE|520
' You can also see that the replace starts a Position 520, as
specified.
' So the start of the text is unchanged.
```

Syntax

TXT.ril|$$LIN|$$SEA|$$REP[|$$STA]

TXT.Replace_in_Line|P1|P2|P3[|P4]

Parameter Explanation

P1 **-** **(Input,numeric)** Variable with Line-Number of the Line that shall be searched for something. The first Line is always Line 1.
P2 **-** **(Input)** Variable with String to search for.
P3 **-** **(Input)** Variable with the String that shall replace the String in **P2**.
P4 **-** **(*optional, numeric*)** Variable with a Byte-Position in the TXT.-Buffer, if given the replace will start at that Position.
If omitted the complete Text is been searched and replaced.

TOS: The TXT.ril returns a value of "0" on the TOS if there was no match found.
If there was something replaced, a "1" is been returned on the TOS.

Example

```
'*****************************************************
' EXAMPLE 1:
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
VAN.$$STA=100
'Replace all "e" with "ef" in internal TXT.-Buffer starting
from Byte-Position 100
VAR.$$OLD=e
VAR.$$NEW=ef
TXT.ril|13|$$OLD|$$NEW|$$STA
ENR.
```

Remarks

Will replace only the FIRST occurrences of the Search-String.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Replace_in_Text 3344
- TXT.Replace 3338
- TXT.Replace_in_Line 3341

© 2023 Theo Gottwald * https://www.smart-package.com

- Push_Intern 3444
- Pop_Intern 3341
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.5.5 Replace_in_Text

MiniRobotLanguage (MRL)

TXT.rit
TXT.Replace_in_Text

Search a Text in the TXT.-internal Buffer and replace it with a given other sequence, using an Equalcase-Search Algo.

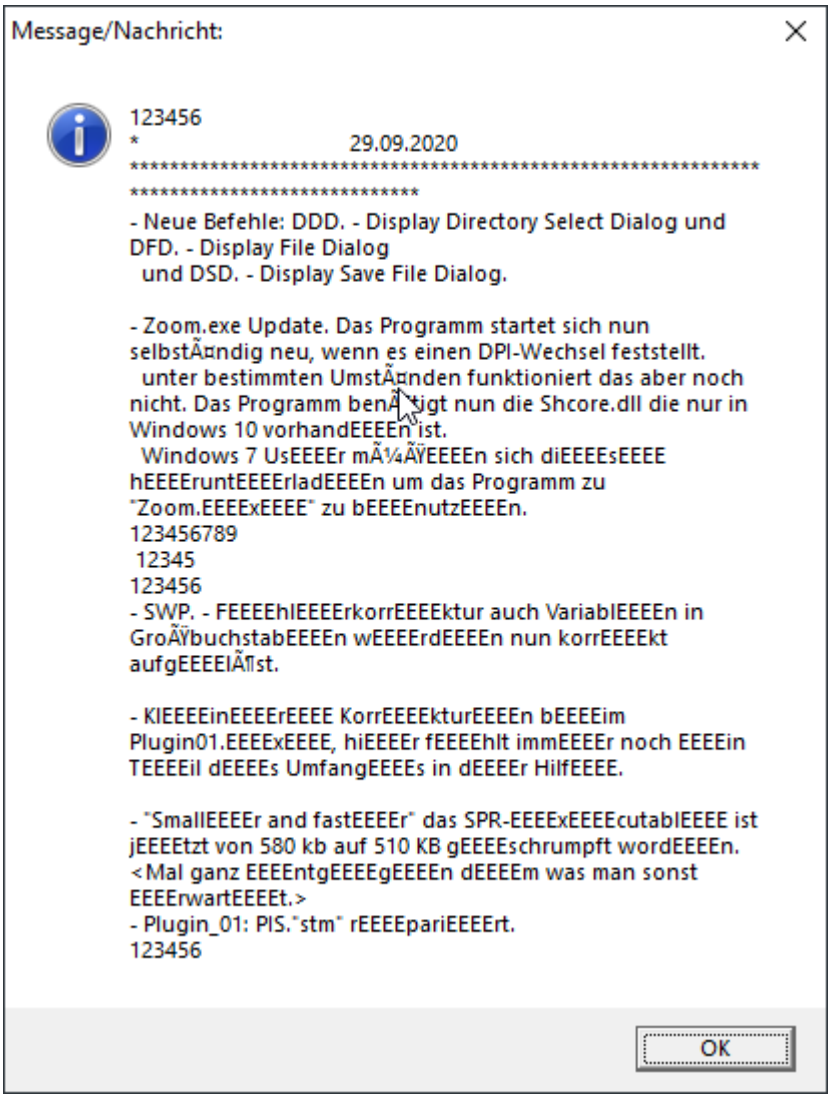
Intention

This command is used to replace all occurrences of a Search-String in the TXT.-Buffer with another String.
It uses an Equalcase-Algorhytm so it will search "equal-case" but the replace will be done with the proper case.

```
'Replace all "e" with "ef" in internal TXT.-Buffer
VAR.$$LAB=e
VAR.$$LAC=ef
TXT.rit|$$LAB|$$LAC
ENR.
```

```
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' This example shows all features of TXT.rit
' We are running in 2 times so its clear that due to the
Equalcase-Replace,
' also the "EE" will be treated as "ee" and replaced.
TXT.rit|e|EE|520
TXT.rit|e|EE|520
' You can also see that the replace starts a Position 520, as
specified.
' So the start of the text is unchanged.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*



Syntax

TXT.rit|$$SEA|$$REP[|$$STA]
TXT.Replace_in_Text|P1|P2[|P3]

Parameter Explanation

P1 – **`(Input)`** Variable with String to search for.
P2 – **`(Input)`** Variable with the String that shall replace the String in **P1**.
P3 – **`(`***optional, numeric***`)`** Variable with a Byte-Position in the TXT.-Buffer, if given the replace will start at that Position.
If omitted the complete Text is been searched and replaced.

Example

```
'*******************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' EXAMPLE 1:
'*************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
VAN.$$STA=100
'Replace all "e" with "ef" in internal TXT.-Buffer starting
from Byte-Position 100
VAR.$$OLD=e
VAR.$$NEW=ef
TXT.rit|$$OLD|$$NEW|$$STA
ENR.
```

Remarks

Will replace ALL occurences of the Search-String. Uses an Equalcase-Algo so the search-string will be found whether in lower- or in Upper-case..

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Replace_in_Text 3344
- TXT.Replace 3338
- TXT.Replace_in_Line 3344
- Push_Intern 3444
- Pop_Intern 3344
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6 Get Lines and Text

Sample Script

```
NEM.
  $$SRC=?exeloc\History.csv
ELS.
  $$SRC=$cmdexe$
EIF.

$$DES=?exeloc\Converted_#datetime#.txt

NEF.$$SRC
  MBX.File existiert nicht!
  GTO.enx
ELS.
  TXT.Load_Textfile|$$SRC|$$RES
  DBP.Loaded $$RES Lines
EIF.

TXT.Get_Line_Count|$$NUM

VAR.$$TIM=#dtime#

' Linie zwischen Header und Rest
VAN.$$LAN=5
VAR.$$DEL=$tab$
VAR.$$DEA=;
VAR.$$REM=all
VAR.$$GES=

FOR.$$LU2|1|$$LAN
  JIV.$$LU2=1|Nex
  IVV.$$LU2=5
    $$SMA=$crlf$Datum$tab$$tab$Zeit$tab$$tab$$tab$Vakuum$ta
  ELS.
    TXT.Get_Line_Trimmed|$$LU2|$$SMA
  EIF.
  VBT.$$SMA
  JNS.$$SMA|Nex
  VAS.$$GES=$$GES$$SMA$crlf$
  :Nex
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
NEX.
VRT.$$GES|$$REM
VAS.$$GES=$$GES$crlf$
CTF.$$DES|$$GES

' Part 2 - Data
TXT.gal|$$LAN|$$GES
'RPL.$$GES|$$DEL|;
TXT.Load_pure_Text|$$GES|-
TXT.Rer|$$DEL|8

TXT.Get_Line_Count|$$NUM
$$IND=1,2,8,3,4,5,6,7
$$INA=3,4,5,6,7,8
$$INE=4,5,6,7
$$FMT=00.00
VAN.$$LAT=0
VDC.$$NUM
FOR.$$LIC|1|$$NUM
  TXT.Get_Line_Trimmed|$$LIC|$$LIA
  'PRT. # $$LIA #
  DIC.Geo|$$DEL|$$LIA|$$IND|$$LIB
  'PRT. # $$LIB #
  DIC.fnu|$$LIB|$$DEL|$$INE|$$FMT
  PRT.$$LIC - $$LIB

  DIC.Insert|$$DEL|$$LIB|$$INA|$$LID
  TXT.Set_Line_To|$$LIC|$$LID
  ' Store last time
  $$LAT=$$TIB
NEX.
'$$FIE=3,4,8
'TXT.Delimited_Items_Insert|$$DEL|$$FIE

TXT.gpt|$$GES|$$FIL
VAR.$$OUT=$$GES
ATF.$$DES|$$OUT

:enx
VAR.$$TIM=#dsince#
CAL.$$LPS=$$RES/$$TIM|i
CAL.$$TIM=$$TIM|i
MBT.Done in $$TIM Sec.$crlf$ Lines per Sec.: $$LPS
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'================================================
```

TAB (ASC(9)) separated Sample Data from a Freeze-Dryer-Log.

```
history data is exported  only for reference.
Export time:2022/07/29 07:40


date   time
2022/07/19 03:43:06 -32.4    -12.0    -12.0    -17.7    36.
2022/07/19 03:44:06 -32.4    -12.0    -12.0    -17.7    36.
2022/07/19 03:45:06 -32.4    -11.8    -12.0    -17.7    36.
2022/07/19 03:46:06 -32.3    -12.1    -12.1    -17.7    36.
2022/07/19 03:47:06 -32.4    -11.9    -12.0    -17.7    36.
2022/07/19 03:48:06 -32.4    -11.8    -12.0    -17.7    36.
2022/07/19 03:49:06 -32.4    -12.0    -12.0    -17.7    36.
2022/07/19 03:50:06 -32.4    -12.0    -12.1    -17.7    36.
2022/07/19 03:51:06 -32.4    -11.9    -12.0    -17.7    37.
2022/07/19 03:52:06 -32.3    -12.0    -12.0    -17.6    36.
2022/07/19 03:53:06 -32.3    -12.0    -12.0    -17.6    36.
2022/07/19 03:54:06 -32.4    -11.8    -12.0    -17.7    36.
2022/07/19 03:55:06 -32.4    -12.0    -12.0    -17.7    36.
2022/07/19 03:56:06 -32.4    -11.9    -12.0    -17.6    37.
2022/07/19 03:57:06 -32.4    -11.9    -12.0    -17.7    36.
2022/07/19 03:58:06 -32.4    -12.0    -12.0    -17.6    36.
2022/07/19 03:59:06 -32.4    -12.1    -12.0    -17.7    36.
2022/07/19 04:00:06 -32.4    -12.0    -12.0    -17.6    37.
2022/07/19 04:01:06 -32.4    -11.9    -12.0    -17.6    36.
2022/07/19 04:02:06 -32.4    -12.0    -12.0    -17.6    36.
2022/07/19 04:03:06 -32.3    -12.0    -12.0    -17.6    36.
```

TAB (ASC(9)) separated Result Data from a Freeze-Dryer-Log, us

```
Export time:2022/07/29 07:40

Datum      Zeit         Vakuum   Kühlraum     Abl.1
2022/07/19 03:43:06     377      -32.40       -12.00
2022/07/19 03:44:06     377      -32.40       -12.00
2022/07/19 03:45:06     377      -32.40       -11.80
2022/07/19 03:46:06     377      -32.30       -12.10
2022/07/19 03:47:06     377      -32.40       -11.90
2022/07/19 03:48:06     377      -32.40       -11.80
```


© 2023 Theo Gottwald * https://www.smart-package.com

2022/07/19	03:49:06	377	-32.40	-12.00
2022/07/19	03:50:06	377	-32.40	-12.00
2022/07/19	03:51:06	377	-32.40	-11.90
2022/07/19	03:52:06	387	-32.30	-12.00
2022/07/19	03:53:06	377	-32.30	-12.00
2022/07/19	03:54:06	377	-32.40	-11.80
2022/07/19	03:55:06	385	-32.40	-12.00
2022/07/19	03:56:06	387	-32.40	-11.90
2022/07/19	03:57:06	387	-32.40	-11.90
2022/07/19	03:58:06	387	-32.40	-12.00
2022/07/19	03:59:06	387	-32.40	-12.10
2022/07/19	04:00:06	387	-32.40	-12.00
2022/07/19	04:01:06	387	-32.40	-11.90
2022/07/19	04:02:06	387	-32.40	-12.00
2022/07/19	04:03:06	387	-32.30	-12.00

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.80.6.1 Get_after_Line

MiniRobotLanguage (MRL)

TXT.gal
TXT.Get_after_Line

Get all Text that is after a specified Line, excluding the specified line.

Intention

This command is used to get all Text after a specified Line from the Text that is in the Internal TXT-Buffer, into a Variable.
It will not include the specified Line itself. Variable P2 will contain the Text after the Line Nr. P1.

```
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gal|$$NUM|$$LEF
LEN.$$LEF|$$LEN
'JIV.$$LEN!238|Lab_failed
```

Syntax

TXT.gal|P1[|P2]
TXT.Get_after_Line|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Text. If omitted, the line is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gal|$$NUM|$$LEF
LEN.$$LEF|$$LEN
MBX.Len of returned Text is:$$LEN
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.2 Get_before_Line

MiniRobotLanguage (MRL)

TXT.gbl
TXT.Get_before_Line

Get all Text that is before a specified Line, excluding the specified line.

Intention

This command is used to get all Text before a specified Line from the Text that is in the Internal TXT-Buffer, into a Variable.
It will not include the specified Line itself. Variable P2 will contain the Text before the Line Nr. P1.

```
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gbl|$$NUM|$$LEF
LEN.$$LEF|$$LEN
'JIV.$$LEN!238|Lab_failed
```

Syntax

TXT.gbl|P1[|P2]
TXT.Get_before_Line|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Text. If omitted, the line is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gbl
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gbl|$$NUM|$$LEF
LEN.$$LEF|$$LEN
MBX.Len of returned Text is:$$LEN
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.3 Get_Line

MiniRobotLanguage (MRL)

TXT.gel
TXT.Get_Line

Get Line with the specified Number from TXT.-internal Buffer into Variable.

Intention

This command is used to get a specified Line from the Text that is in the Internal TXT-Buffer into a Variable.
Variable P2 will contain the Line Nr. P1.

This Command can be used to work on a "Line by line" base through a file without accessing the file on the harddrive.

```
' Get Line Number $$NUM into $$LIN
$$NUM=10
TXT.gel|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
```

Syntax

TXT.gel|P1[|P2]
TXT.Get_Line|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Line. If omitted, the line is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gel
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
CFF.$$LAA|$$RES
TXT.lpt|$$RES|$$LAA
' Now we get only Line 5
TXT.gel|5|$$LIN
MBX.Line 5 is: $$LIN
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.4 Get_Line_Count

MiniRobotLanguage (MRL)

TXT.glc
TXT.Get_Line_Count

Return the number of lines of the currently loaded File inside the TXT.-internal Buffer

Intention

This command is used to return the number of lines of the Textfile, that was loaded into the internal Buffer of the TXT.-Commands.

```
' Load Textfile into intrnal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
TXT.glc|$$RES
MBX. The loaded File has $$RES Lines.
```

Syntax

TXT.glc|P1
TXT.Get_Line_Count|P1

Parameter Explanation

P1 - **(Output)** Variable with the result of the operation.
On error a "" is returned also the Timeout-Flag is set.
On success, the Lenght in Bytes/Charcters of the currently loaded file is been returned.
If omitted the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.gtl etc.
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
TXT.clr
TXT.glc|$$LAB
JIV.$$LAB!0|Lab_failed
TXT.gtl|$$LAC
JIV.$$LAC!0|Lab_failed
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- TXT.Get_Line_Count 3357
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.5 Get_Line_lefttrim

MiniRobotLanguage (MRL)

TXT.gel
TXT.Get_Line_lefttrim

Get Line with the specified Number from TXT.-internal Buffer into Variable and trim Spaces from left side.

Intention

This command is used to get a specified Line from the Text that is in the Internal TXT-Buffer into a Variable.
It will also trim all Spaces and TAB's from the left side.
Variable P2 will contain the Line Nr. P1.

This Command can be used to work on a "Line by line" base through a file without accessing the file on the harddrive.

```
' Get Line Number $$NUM into $$LIN
$$NUM=10
TXT.gll|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
```

Syntax

TXT.gll|P1[|P2]
TXT.Get_Line_lefttrim|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Line. If omitted, the line is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gel
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
CFF.$$LAA|$$RES
TXT.lpt|$$RES|$$LAA
' Now we get only Line 5
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TXT.gll|5|$$LIN
MBX.Line 5 is: $$LIN
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- TXT.get_line_lefttrim 3359
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.6 Get_Line_Position

MiniRobotLanguage (MRL)

TXT.glp
TXT.Get_Line_Position

Get Byte-Position of Line with the specified Number from TXT.-internal Buffer into Variables.

Intention

This command is used to get a specified Line-Position from the Text that is in the Internal TXT-Buffer into a Variables.
Variable **P2** will contain the number of the first byte of Line Nr. **P1**. Variable P3 will contain the number of the last character of Line **P1**,
the following `$crlf$` is not included. Threrefore if a line is followed by `$crlf$` then these bytes will come after **P3**.

```
' Get Line Number $$NUM into $$LIN
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.lft|$$LAA
$$NUM=15
' Get star- and ENd-Byte from Line 15
TXT.glp|$$NUM|$$STA|$$END
CFF.$$LAA|$$TXT
GSS.$$STA|$$END|$$TXT|$$LAE
' Line is now in $$LAE
LEN.$$LAE|$$RES
MBX.Line 10 is: $crlf$ $$LIN
```

Syntax

TXT.glp|P1|P2|P3
TXT.Get_Line_Position|P1|P2|P3

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Start-Byte of the specified Line. If omitted, the result is placed on TOS.
P3 - **(Output)** Variable will contain the End-Byte of the specified Line. If omitted, the result is placed on TOS.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*****************************************************
' EXAMPLE 1: TXT.glp
'*****************************************************
' Get Line Number $$NUM into $$LIN
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.lft|$$LAA
$$NUM=15
' Get star- and ENd-Byte from Line 15
TXT.glp|$$NUM|$$STA|$$END
CFF.$$LAA|$$TXT
GSS.$$STA|$$END|$$TXT|$$LAE
' Line is now in $$LAE
LEN.$$LAE|$$RES
MBX.Line 10 is: $crlf$ $$LIN
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.7 Get_Line_trimmed

MiniRobotLanguage (MRL)

TXT.glt
TXT.TXT.Get_Line_trimmed

Get Line with the specified Number from TXT.-internal Buffer into Variable and trim Spaces from both sides.

Intention

This command is used to get a specified Line from the Text that is in the Internal TXT-Buffer into a Variable.
It will also trim all Spaces and TAB's from both sides.
Variable P2 will contain the Line Nr. P1.

This Command can be used to work on a "Line by line" base through a file without accessing the file on the harddrive.

```
' Get Line Number $$NUM into $$LIN
$$NUM=10
TXT.glt|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
```

Syntax

TXT.glt|P1[|P2]
TXT.Get_Line_trimmed|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Line. If omitted, the line is placed on TOS.

Speed in Ticks:

This command uses typically around 250 Ticks.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gel
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
CFF.$$LAA|$$RES
TXT.lpt|$$RES|$$LAA
' Now we get only Line 5
TXT.glt|5|$$LIN
MBX.Line 5 is: $$LIN
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- TXT.get_line_lefttrim 3359
- TXT.Get_Line_trimmed 3363
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.8 Get_Text_Block

MiniRobotLanguage (MRL)

TXT.Get_Text_Block
TXT.Gtb

Return a Part from the TXT.-internal Buffer into a Variable, specified with a Start- and End-Byte.

Intention

This command is used to get a specified Part from the Text that is in the Internal TXT-Buffer into a Variable.
Variable **P1** will contain the starting Byte number and **P2** will contain the last Byte number.

```
' Get Lines from 10 to 90 out of the TXT.-Buffer
TXT.gtb|10|900|$$LIS
PRT.$$LIS
MBX.!
```

Syntax

TXT.Gtb|P1|P2[|P3]
TXT.Get_Text_Block|P1|P2[|P3]

Parameter Explanation

P1 - **`(Input,numeric)`** Variable with starting Byte-Number.
P2 - **`(Input)`** Variable with the ending Byte-Number.
P3 - **`(Output)`** Variable for the result.

Example

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- TXT.Load_Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.9 Get_Text_Lines

MiniRobotLanguage (MRL)

TXT.Get_Text_Lines
TXT.Gtls

Return a number of Lines staring from the specified Line-Number to the second specified Line-Number from the TXT.-internal Buffer.

Intention

This command is used to get a specified Part from the Text that is in the Internal TXT-Buffer into a Variable.
Variable **P1** will contain the Line Nr. of the first Line, and **P2** will contain the Line-Number of the last line.

```
' Get Lines from 10 to 90 out of the TXT.-Buffer
TXT.gtls|10|90|$$LIS
PRT.$$LIS
MBX.!
```

Syntax

TXT.Gtls|P1|P2[|P3]
TXT.Get_Text_Lines|P1|P2[|P3]

Parameter Explanation

P1 - **(Input,numeric)** Variable with starting Line-Number.
P2 - **(Input)** Variable with the ending Line-Number.
P3 - **(Output)** Variable for the result.

Example

```
'******************************************************
'
'******************************************************
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- TXT.Load_Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.10 Get_To_End

MiniRobotLanguage (MRL)

TXT.gte
TXT.Get_to_End

Get all Text that is after a specified Line, including the specified line.

Intention

This command is used to get all Text that is after a specified Line, including the specified Line, from the Text that is in the Internal TXT-Buffer, into a Variable. Variable P2 will contain the result Text include and after the Line Nr. **P1**.

```
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gte|$$NUM|$$LEF
LEN.$$LEF|$$LEN
'JIV.$$LEN!238|Lab_failed
```

Syntax

TXT.gte|P1[|P2]
TXT.Get_to_End|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Text. If omitted, the line is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gte
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA
' Get all Text after Line 10 into $$LEF
$$NUM=10
TXT.gte|$$NUM|$$LEF
LEN.$$LEF|$$LEN
MBX.Len of returned Text is:$$LEN
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.11 Get_To_Start

MiniRobotLanguage (MRL)

TXT.gts
TXT.Get_to_Start

Get all Text that is before a specified Line, including the specified line.

Intention

This command is used to get all Text before a specified Line from the Text that is in the Internal TXT-Buffer, into a Variable.
It will include the specified Line itself. Variable P2 will contain the Text before and include the Line Nr. P1.

```
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gts|$$NUM|$$LEF
LEN.$$LEF|$$LEN
'JIV.$$LEN!238|Lab_failed
```

Syntax

TXT.gts|P1[|P2]
TXT.Get_to_Start|P1[|P2]

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Output)** Variable will contain the Text. If omitted, the line is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gts
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA
' Get all Text before Line 10 into $$LEF
$$NUM=10
TXT.gts|$$NUM|$$LEF
LEN.$$LEF|$$LEN
MBX.Len of returned Text is:$$LEN
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.12 Line_Start_End_by_Position

MiniRobotLanguage (MRL)

TXT.lsp
TXT.Line_Start_End_by_Position

Get Byte-Positions of Line Start and End using an Byte-Position as Input, from TXT.-internal Buffer into Variables.

Intention

This command is used to get the Byte-Positions of Start- and End of a specified Line,
from the Text that is in the Internal TXT-Buffer into Variables.
Unlike TXT.glp, it works from a Byte-Position. So you say "Byte 42" and the command will return the start- and End-Byte of the line,
that contains Byte 43. Variable **P2** will contain the Start-Byte of the Line. Variable **P3** will contain the Last-Byte of the Line before
and excluding the `$crlf$`.
This Command can be used to work on a Byte-Base through a file without accessing the file on the harddrive.

```
' Byte 35 is in Line 2
TXT.lnp|35|$$LAC
JIV.$$LAC!2|Lab_failed
' Get Line 2 Start and End-Position
TXT.glp|$$LAC|$$LAD|$$LAE
JIV.$$LAB!9|Lab_failed
JIV.$$LAC!56|Lab_failed
' Get Byte 35 Start and End-Position (of that Line containing
Byte 35)
TXT.lsp|35|$$LAB|$$LAC
JIV.$$LAB!9|Lab_failed
JIV.$$LAC!56|Lab_failed
ENR.
```

Syntax

TXT.lsp|P1|P2|P3
TXT.Line_Start_End_by_Position|P1|P2|P3

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** **`(Input,numeric)`** Variable with Byte Position.
P2 **-** **`(Output)`** Variable will contain the Start-Byte of the Line that contains this Byte. If omitted, the result is placed on TOS.
P3 **-** **`(Output)`** Variable will contain the End-Byte of the Line. If omitted, the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.glp
'*****************************************************
' Get Line Number $$NUM into $$LIN
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.lnp|100|$$LIN
TXT.lsp|100|$$STA|$$END
MBX.Byte 100 is in line $$LIN from: $$STA to $$END
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.6.13 Linenumber_from_Position

MiniRobotLanguage (MRL)

TXT.lnp
TXT.Linenumber_from_Position

Get Line-Number in TXT.-internal Buffer from a Byte-Position.

Intention

This command is used to get a specified Line-Number from a Byte-Position, from the Text that is in the Internal TXT-Buffer.
Variable **P2** will contain the Line Nr. **P1** contains the Byte-Position.
For example, you have found something at Byte Postion 120 and now you want to know "Which Line is it?".
This Command can be used to work with Bytes on a "Line by line" base through a file without accessing the file on the harddrive.

```
' Get Line Number from byte 5
TXT.lnp|5|$$LAC
MBX.Byte 5 is in Line Nr.$$LAC
```

Syntax

TXT.lnp|P1[|P2]
TXT.Linenumber_from_Position|P1[|P2]

Parameter Explanation

P1 **-** **(Input,numeric)** Variable with Byte-Position.
P2 **-** **(Output)** Variable will contain the Line-Number.If omitted, the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gel
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA
$$BYT=5
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
TXT.lnp|$$BYT|$$LIN
' Now we get the Linenumber from the line that contains Byte 5
MBX.Byte 5 is in Line: $$LIN
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.Get_pure_Text 3455
- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Load Unicode_File 3420
- TXT.Load_pure_Text 3376
- TXT.Save_File 3424
- Push_Intern 3444
- Pop_Intern 3376
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7 Insert/Delete Text

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.1 Delete_Characters

MiniRobotLanguage (MRL)

TXT.dch
TXT.Delete_Characters

Delete Text at the specified Byte-Position in the actual TXT.-Buffer.

Intention

This command is used to delete Bytes or Text from the internal Text-Buffer at any Position.
The Text is deleted and the remaining Text is moved up.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$POS=2
$$NUM=3
TXT.dch|$$POS|$$NUM
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<DEFG>
<ABCDEFG>
<ABCDEFG>
```

Syntax

TXT.dch|P1|P2
TXT.Delete_Characters|P1|P2

Parameter Explanation

P1 **- (Input,Numeric)** Variable with Text that will be Inserted at Position **P2**.
P2 **- (Input,Numeric)** Variable with Line-Number, before which the Text will be inserted.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1: TXT.dch
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$POS=2
$$NUM=3
TXT.dch|$$POS|$$NUM
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.2 Delete_Line

MiniRobotLanguage (MRL)

TXT.dli
TXT.Delete_Line

Delete a complete Line, including the $crlf$ at the end of the Line at the specified Position in the actual TXT.-Buffer.

Intention

This command is used to delete a complete Line, including the following `$crlf$` from the internal Text-Buffer.
The Line is deleted and the remaining Text is moved up.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$POS=2
TXT.dli|$$POS
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<ABCDEFG>
<ABCDEFG>
```

Syntax

TXT.dli|P1
TXT.Delete_Line|P1

Parameter Explanation

P1 - **`(Input,Numeric)`** Variable with Line-Number of Line that will be deleted.

Example

```
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' EXAMPLE 1: TXT.dli
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$POS=2
TXT.dli|$$POS
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.3 Delete_Lines

MiniRobotLanguage (MRL)

TXT.dls
TXT.Delete_Lines

Delete multiple Lines, including the $crlf$ at the end of the Lines, from Line P1 to Line P2.

Intention

This command is used to delete Lines, starting with Line **P1** until and including Line **P2**.
The Lines are deleted, including the following `$crlf$` and the remaining Text is moved up.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdefg>$crlf$
TXT.Load_Pure_Text|$$TXT
$$POS=2
$$NUM=3
TXT.dls|$$POS|$$NUM
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<ABCDEFG>
```

Syntax

TXT.dls|P1|P2
TXT.Delete_Lines|P1|P2

Parameter Explanation

P1 **- (Input,Numeric)** Variable with Text that will be Inserted at Position **P2**.
P2 **- (Input,Numeric)** Variable with Line-Number, before which the Text will be inserted.

Example

```
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' EXAMPLE 1: TXT.dls
'*************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<123456>$crlf$<abcdefg>$crlf$
TXT.Load_Pure_Text|$$TXT
$$POS=2
$$NUM=3
TXT.dls|$$POS|$$NUM
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.4 Insert_Line_after_Line

MiniRobotLanguage (MRL)

TXT.ilal
TXT.Insert_Line_after_Line

Insert a Line after the specified Line in the actual TXT.-Buffer.

Intention

This command is used to Insert Bytes or Text into the internal Text-Buffer at any Position.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_after_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<ABCDEFG>
<ABCDEFG>
123
<ABCDEFG>
```

Syntax

TXT.ilal|P1|P2
TXT.Insert_Line_after_Line|P1|P2

Parameter Explanation

P1 - **`(Input,Text)`** Variable with Text that will be Inserted at Position **P2**.
P2 - **`(Input,Numeric)`** Variable with Line-Number, before which the Text will be inserted.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_after_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.5 Insert_Line_at_End

MiniRobotLanguage (MRL)

TXT.ilae
TXT.Insert_Line_at_End

Insert a Line after the specified Line in the actual TXT.-Buffer.

Intention

This command is used to append the internal Text with - for example - an Footer.
In some cases you will want to add an `$crlf$` before the Footer so that its not appended to the last line.
Using this command a `$crlf$` will be automatically appended to the Footer-Text on the right side.
It will also append an `$crlf$` automatically to the Left side, if there is no `$crlf$` after the Text.
Therefore each call to this command will add a new Line.

Before:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
3	Ulrike	Nieter	– no bio	0,8	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

```
$$HEA=Footer-Line
TXT.Insert_Line_at_End|$$HEA
```

After:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ulrike	Nieter	– no bio	0,8	Germany
3	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany
5	Footer-Line				

Syntax

TXT.ilae|P1
TXT.Insert_Line_at_End|P1

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **- `(Input,Text)`** Variable with Text that will be appended to the Text.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  'VAR.$$FIA=?exeloc\Sources\Politician.CSV
  'VAR.$$FIA=?exeloc\Sources\WEF-Members.CSV
  VAR.$$FIA=?exeloc\Sources\4Zeiler.CSV
  'VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="D"
TXT.sort_by_column_Float|2|$$IND|;|a

$$HEA=Headline$crlf$
TXT.Insert_Text_at_Start|$$HEA

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.6 Insert_Line_at_Pos

MiniRobotLanguage (MRL)

TXT.ilap
TXT.Insert_Line_at_Pos

Insert a Line (with an $crlf$ at the end) at the specified Byte-Position in the actual TXT.-Buffer.

Intention

This command is used to Insert a Text-Line including the needed `$crlf$` before and after the Text into the internal Text-Buffer at any Position.
Let say that you have the Text:

```
' Insert Text/Bytes at any Place inside the TXT.Buffer
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_at_Pos|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

then the result might look like this:

```
<
123
ABCDEFG>
<ABCDEFG>
<ABCDEFG>
```

Syntax

TXT.ilap|P1|P2
TXT.Insert_Line_at_Pos|P1|P2

Parameter Explanation

P1 **- (Input,Text)** Variable with Line that will be Inserted at Position **P2**. The `$crlf$` will be automatically appended.

© 2023 Theo Gottwald * https://www.smart-package.com

P2 **- `(Input,Numeric)`** Variable with Byte-Position, where the Text will be inserted. The Text that was before at this position will be moved to the right side.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ilap
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_at_Pos|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.7 Insert_Line_at_Start

MiniRobotLanguage (MRL)

TXT.ilas
TXT.Insert_Line_at_Start

Insert Text and $crlf$ before the actual TXT.-Buffer Content.

Intention

This command is used to prepend the internal Text with - for example - an Headline. An `$crlf$` will automatically be added to the Headline so that its not prepended to the first line.

Before:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
3	Ulrike	Nieter	– no bio	0,8	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

```
$$HEA=Headline
TXT.Insert_Text_at_Start|$$HEA
```

After:

	A	B	C	D	E
1	Headline				
2	First name	Family name	Link bio	Lenght	Country
3	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
4	Ulrike	Nieter	– no bio	0,8	Germany
5	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

Syntax

TXT.ilas|P1
TXT.Insert_Line_at_Start|P1

Parameter Explanation

P1 **-** **`(Input,Text)`** Variable with Text that will be prepended to the Text.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  'VAR.$$FIA=?exeloc\Sources\Politician.CSV
  'VAR.$$FIA=?exeloc\Sources\WEF-Members.CSV
  VAR.$$FIA=?exeloc\Sources\4Zeiler.CSV
  'VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="D"
TXT.sort_by_column_Float|2|$$IND|;|a

$$HEA=Headline
TXT.Insert_Line_at_Start|$$HEA

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.8 Insert_Line_before_Line

MiniRobotLanguage (MRL)

TXT.ilbl
TXT.Insert_Line_before_Line

Insert a Line before the specified Line in the actual TXT.-Buffer.

Intention

This command is used to Insert Bytes or Text into the internal Text-Buffer at a specified Line-Number Position.
A `$CRLF$` will automatically be appended to the Line.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_before_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<ABCDEFG>
123<ABCDEFG>
<ABCDEFG>
```

Syntax

TXT.ilbl|P1|P2
TXT.Insert_Line_before_Line|P1|P2

Parameter Explanation

P1 **-** **`(Input,Text)`** Variable with Text that will be Inserted at Position **P2**.
P2 **-** **`(Input,Numeric)`** Variable with Line-Number, before which the Text will be inserted.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Line_before_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.9 Insert_Text_after_Line

MiniRobotLanguage (MRL)

TXT.itbl
TXT.Insert_Text_after_Line

Insert Text after the specified Line in the actual TXT.-Buffer.

Intention

This command is used to Insert Bytes or Text into the internal Text-Buffer at any Position.

```
' Insert Text/Bytes at any Place inside the TXT.Buffer
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Text_after_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<ABCDEFG>
<ABCDEFG>123
<ABCDEFG>
```

Syntax

TXT.itbl|P1|P2
TXT.Insert_Text_after_Line|P1|P2

Parameter Explanation

P1 - **`(Input,Text)`** Variable with Text that will be Inserted at Position **P2**.
P2 - **`(Input,Numeric)`** Variable with Line-Number, after which the Text will be inserted.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  'VAR.$$FIA=?exeloc\Sources\Politician.CSV
  'VAR.$$FIA=?exeloc\Sources\WEF-Members.CSV
  VAR.$$FIA=?exeloc\Sources\4Zeiler.CSV
  'VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="D"
TXT.sort_by_column_Float|2|$$IND|;|a

$$HEA=Headline$crlf$
TXT.Insert_Text_at_Start|$$HEA

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.10 Insert_Text_at_End

MiniRobotLanguage (MRL)

TXT.itae
TXT.Insert_Text_at_End

Insert Text after the actual TXT.-Buffer Content.

Intention

This command is used to append the internal Text with - for example - an Footer. In some cases you will want to add an `$crlf$` before the Footer so that its not appended to the last line.

Before:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
3	Ulrike	Nieter	– no bio	0,8	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

```
$$HEA=Footer-Line
TXT.Insert_Text_at_End|$$HEA
```

After:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ulrike	Nieter	– no bio	0,8	Germany
3	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany
5	Footer-Line				

Syntax

TXT.itae|P1
TXT.Insert_Text_at_End|P1

Parameter Explanation

P1 **-** **`(Input,Text)`** Variable with Text that will be appended to the Text.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  'VAR.$$FIA=?exeloc\Sources\Politician.CSV
  'VAR.$$FIA=?exeloc\Sources\WEF-Members.CSV
  VAR.$$FIA=?exeloc\Sources\4Zeiler.CSV
  'VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="D"
TXT.sort_by_column_Float|2|$$IND|;|a

$$HEA=Headline$crlf$
TXT.Insert_Text_at_Start|$$HEA

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.11 Insert_Text_at_Pos

MiniRobotLanguage (MRL)

TXT.itap
TXT.Insert_Text_at_Pos

Insert Text at the specifies Byte-Position in the actual TXT.-Buffer.

Intention

This command is used to Insert Bytes or Text into the internal Text-Buffer at any Position.
After each Line there are 2 Byte with `$crlf$` that delimit the Lines.
These will also be counted for the Pos.-Value.

```
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=4
TXT.Insert_Text_at_pos|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

```
Result is:
```

```
<AB123CDEFG>
<ABCDEFG>
<ABCDEFG>
```

Syntax

TXT.itap|P1|P2
TXT.Insert_Text_at_Pos|P1|P2

Parameter Explanation

P1 **- `(Input,Text)`** Variable with Text that will be Inserted at Position **P2**.
P2 **- `(Input,Numeric)`** Variable with Byte-Position, where the Text will be inserted. The Text that was before at this position will be moved to the right side.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  'VAR.$$FIA=?exeloc\Sources\Politician.CSV
  'VAR.$$FIA=?exeloc\Sources\WEF-Members.CSV
  VAR.$$FIA=?exeloc\Sources\4Zeiler.CSV
  'VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="D"
TXT.sort_by_column_Float|2|$$IND|;|a

$$HEA=Headline$crlf$
TXT.Insert_Text_at_Start|$$HEA

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBT.$$TXT|Programm Ende|9|32
ENR.
'------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.12 Insert_Text_at_Start

MiniRobotLanguage (MRL)

TXT.itas
TXT.Insert_Text_at_Start

Insert Text before the actual TXT.-Buffer Content.

Intention

This command is used to prepend the internal Text with - for example - an Headline. In most cases you will want to add an `$crlf$` to the Headline so that its not prepended to the first line.

Before:

	A	B	C	D	E
1	First name	Family name	Link bio	Lenght	Country
2	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
3	Ulrike	Nieter	– no bio	0,8	Germany
4	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

```
' Get all Text before Line 10 into $$LEF
$$HEA=Headline$crlf$
TXT.Insert_Text_at_Start|$$HEA
```

After:

	A	B	C	D	E
1	Headline				
2	First name	Family name	Link bio	Lenght	Country
3	Ernst-Adrian	von Dörnber	Ernst-Adrian	4,00E-02	Germany
4	Ulrike	Nieter	– no bio	0,8	Germany
5	Wolfgang	Riehle	Wolfgang Rie	2,00E+04	Germany

Syntax

TXT.itas|P1
TXT.Insert_Text_at_Start|P1

Parameter Explanation

P1 - **`(Input,Text)`** Variable with Text that will be prepended to the Text.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
' csv-File from:
' https://maloneinstitute-org.translate.goog/wef?
_x_tr_sl=en&_x_tr_tl=de&_x_tr_hl=de&_x_tr_pto=wapp
'
$$TMA=#dtime#
IEM.
  VAR.$$FIA=$cmdexe$
ELS.
  'VAR.$$FIA=?exeloc\Sources\Politician.CSV
  'VAR.$$FIA=?exeloc\Sources\WEF-Members.CSV
  VAR.$$FIA=?exeloc\Sources\4Zeiler.CSV
  'VAR.$$FIA=?exeloc\Sources\Preise_Sample.CSV
EIF.
VAR.$$OUF=?exeloc\Result_#date#.csv
NEF.$$FIA
  $$ERR=Die Quelldatei: $$FIA$crlf$existiert nicht.
  GTO.ero
EIF.
'-----------------------------------------------------------
TXT.Load_Textfile|$$FIA|$$LIC
' Convert $LF to $crlf$ if needed.
'TXT.lfp
TXT.Get_Line_Count|$$LIC
IVV.$$LIC<5
  $$ERR=File is empty?
  GTO.ero
EIF.
PRT.$$LIC Lines loaded.
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' Place your Script here

$$IND="D"
TXT.sort_by_column_Float|2|$$IND|;|a

$$HEA=Headline$crlf$
TXT.Insert_Text_at_Start|$$HEA

'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
' These lines process and save the result.
'TXT.lfr
TXT.Save_File|$$OUF
EXO.$$OUF
'++++++++++++++++++++++++++++++++
'++++++++++++++++++++++++++++++++
:enx
CAL.$$TMA=#dsince#*1000|i
IVV.$$TMA>1100
  CAL.$$TMA=$$$TMA/1000
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  $$TXT=Programm wird beendet nach $$TMA Sekunden.
ELS.
  $$TXT=Programm wird beendet nach $$TMA ms.
EIF.
MBT.$$TXT|Programm Ende|9|32
ENR.
'-------------------------------------------------------------
:ero
PRT.$$ERR
GTO.enx
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.13 Insert_Text_before_Line

MiniRobotLanguage (MRL)

TXT.itbl
TXT.Insert_Text_before_Line

Insert Text before the specified Line in the actual TXT.-Buffer.

Intention

This command is used to Insert Bytes or Text into the internal Text-Buffer at any Position.

```
' Insert Text/Bytes at any Place inside the TXT.Buffer
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Text_before_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Result:

```
<ABCDEFG>
123<ABCDEFG>
<ABCDEFG>
```

Syntax

TXT.itbl|P1|P2
TXT.Insert_Text_before_Line|P1|P2

Parameter Explanation

P1 **-** **`(Input,Text)`** Variable with Text that will be Inserted at Position **P2**.
P2 **-** **`(Input,Numeric)`** Variable with Line-Number, before which the Text will be inserted.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1: TXT.gal
'*****************************************************
$$OUT=?exeloc\Result.txt
$$TXT=<ABCDEFG>$crlf$<ABCDEFG>$crlf$<ABCDEFG>$crlf$
TXT.Load_Pure_Text|$$TXT
$$HEA=123
$$POS=2
TXT.Insert_Text_before_Line|$$HEA|$$POS
TXT.Save_File|$$OUT
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.14 Set_Line_to

MiniRobotLanguage (MRL)

TXT.slt
TXT.Set_Line_to

Replace a Line with the specified Number from the TXT.-internal Buffer with a new Line.

Intention

This command is used to get a specified Line from the Text that is in the Internal TXT-Buffer into a Variable.
Variable P2 will contain the Line Nr. P1.

This Command can be used to work on a "Line by line" base through a file without accessing the file on the harddrive.

```
' Change Line Number $$NUM into $$NEW
$$NUM=10
VAR.$$NEW=Hallo
TXT.gel|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
TXT.slt|$$NUM|$$NEW
TXT.gel|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
```

Syntax

TXT.slt|P1|P2
TXT.Set_Line|P1|P2
TXT.Set_Line_to|P1|P2

Parameter Explanation

P1 - **(Input,numeric)** Variable with Line-Number.
P2 - **(Input)** Variable with the new Line that is going to replace the current line.

Example

```
'*************************************************
' EXAMPLE 1: TXT.gel
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' Change Line Number $$NUM into $$NEW
$$NUM=10
VAR.$$NEW=Hallo
TXT.gel|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
TXT.slt|$$NUM|$$NEW
TXT.gel|$$NUM|$$LIN
MBX.Line 10 is: $crlf$ $$LIN
ENR.
```

Remarks

-

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.7.15 Set_Text_Lines

MiniRobotLanguage (MRL)

TXT.stl
TXT.Set_Text_Lines

Replace a number of Lines with the P3

Intention

This command is used to replace a block of Text-Lines from (and including) Line **P1** to (and including) Line **P2** with the content of **P3**.

```
' Untested
```

Syntax

```
TXT.stl|P1|P2|P3
TXT.Set_Text_Lines|P1|P2|P3
```

Parameter Explanation

P1 - **`(Input,numeric)`** Variable with Line-Number of the first Line that will be replaced.
P2 - **`(Input,numeric)`** Variable with Line-Number of the last Line that will be replaced.
P2 - **`(Input)`** Variable with the new Lines or Text that will be inserted.

Example

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- TXT.Get_Line 3355
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- TXT.Set_Line_to 3411

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.8 Load / Save Commands

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.8.1 Clear

MiniRobotLanguage (MRL)

TXT.clr
TXT.Clear

Clear the internal Buffer where a file was loaded with TXT.ltf

Intention

This command is used to clear the internal Buffer, into which a Textfile was loaded, using the TXT.ltf-Command.
This command will just free the memory that was used to store the Textfile.
You do not need to use this command, as the SPR will freee all memory automatically:
- when you load a new file into the Buffer
- when the Script and the SPR-ends.

```
' Load Textfile into intrnal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
MBX.Operation was succesfull: $$RES
TXT.Clear
```

Syntax

TXT.clr
TXT.Clear

Parameter Explanation

```
Does not need any Parameters
```

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.clr and TXT.gtl
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TXT.ltf|$$LAA|$$RES
TXT.clr
TXT.gtl|$$LAB
JIV.$$LAB!0|Lab_failed
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.8.2 Load Text File

MiniRobotLanguage (MRL)

TXT.ltf
TXT.Load_Textfile

Load a Textfile into the internal Buffer for further Operations using other TXT.-Commands

Intention

This command is used to load a Textfile into the internal Buffer of the TXT.-Commands.
From there you can do all sorts of operations on the file without accessing the harddrive.

```
' Load Textfile into intrnal Buffer for further TXT.-Command
Operations
'$$RES is 0 in case of error
' returns number of loaded lines in case of success
TXT.Load_Textfile|$$FIL|$$RES
MBX.Operation was succesfull, Lines loaded: $$RES
```

Syntax

TXT.ltf|P1[|P2]
TXT.Load_Textfile|P1[|P2]

Parameter Explanation

P1 - Filename and Path of the Textfile to load.

P2 - **(Output)** Variable with the result of the operation.
On error a "0" is returned also the Timeout-Flag is set.
On succes, the number of loaded lines is returned.
If omitted the result is placed on TOS.

Speed in Ticks:

This command depends heavily on teh general file-system speed and the size of the Textfile. The command uses typically around 2500 Ticks with a small Textfile.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.gpt and TXT.gtl
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
TXT.gpt|$$RES
VAN.$$LAB=0
LEN.$$RES|$$LAB
JIV.$$LAB!1040|Lab_failed
VAN.$$LAC=0
TXT.gtl|$$LAC
JIV.$$LAB!1040|Lab_failed
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Load Unicode_File 3420
- TXT.Load_pure_Text 3418
- TXT.Save_File 3424
- Push_Intern 3444
- Pop_Intern 3418
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.8.3 Load Unicode_File

MiniRobotLanguage (MRL)

TXT.luf
TXT.Load_Unicode_File

Load a Unicode-Textfile into the internal Buffer and convert to ANSI for further Operations using other TXT.-Commands

Intention

This command is used to load a Unicode-Textfile - for example a ".msg"-file - into the internal Buffer of the TXT.-Commands.
It will also convert the file to ANSI-Characterset, so it can be used like a normal ANSI-File.
From there you can do all sorts of operations on the file.
You can later use the "TXT.sau" Command to save it as Unicode back to the original Filename
Please note that due the nature of this process, some international characters may get lost.

```
' Load Unicode-Textfile into internal Buffer for further TXT.-
Command Operations
'$$RES is 0 in case of error
' returns number of loaded lines in case of success
TXT.Load_Unicode_File|$$FIL|$$RES
MBX.Operation was succesfull, Lines loaded: $$RES
```

Syntax

TXT.luf|P1[|P2]
TXT.Load_Unicode_File|P1[|P2]

Parameter Explanation

P1 - Filename and Path of the Textfile to load.

P2 - **(Output)** Variable with the result of the operation.
On error a "0" is returned also the Timeout-Flag is set.
On succes, the number of loaded lines is returned.

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
    If omitted the result is placed on TOS.
```

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.gpt and TXT.gtl
'*****************************************************
VAR.$$LAA=?path\..\Pics\Outlook.msg
TXT.luf|$$LAA|$$RES
TXT.gpt|$$TXT
PRT.$$TXT
MBX.Ready
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Load Unicode_File 3420
- TXT.Load_pure_Text 3420
- TXT.Save_File 3424
- Push_Intern 3444
- Pop_Intern 3420
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.8.4 Save_as_Unicode

MiniRobotLanguage (MRL)

TXT.sau
TXT.Save_as_Unicode

Load a Unicode-Textfile into the internal Buffer and convert to ANSI for further Operations using other TXT.-Commands

Intention

This command is used to save the TXT%.-internal Buffer as a Unicode-Textfile - for example a ".msg"-file.
It will not do any changes to the internal FIle, just save it as Unicode.
If you convert files from Unicode to ANSI and back, please note that due the nature of this process, some international characters may get lost.

```
' Load Unicode-Textfile into internal Buffer for further TXT.-
Command Operations
TXT.Save_as_Unicode|$$FIL
MBX.Operation was succesfull, Result: $tos$
```

Syntax

TXT.sau|P1[|P2]
TXT.Save_as_Unicode|P1[|P2]

Parameter Explanation

P1 **-** ***(Input, optional)*** Filename and Path of the Textfile to Save it to. If omitted the original filename where it was loaded is been used.

P2 **-** **(Output)** Variable with the result of the operation. On error a "0" is returned also the Timeout-Flag is set.
On succes, the number of loaded lines is returned. If omitted the result is placed on TOS.

TOS: Calling TXT.SAV will leave a 0 or a 1 on the TOS that will show if the Operation was successful (1) or failure (0).

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.luf  and TXT.sau
'*****************************************************
VAR.$$LAA=?path\..\Pics\Outlook.msg
TXT.sau|$$LAA
TXT.gpt|$$TXT
PRT.$$TXT
MBX.Ready
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Save_as_Unicode 3422
- Push_Intern 3444
- Pop_Intern 3422
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.8.5 Save_File

MiniRobotLanguage (MRL)

TXT.Sav
TXT.Save
TXT.Save_File

Save the TXT.-internal Buffer into the file where it was loaded, or to a specified File-Path.

Intention

This command is used to save the internally stored file back to the Harddrive.
Default it will use the same Filename with which it has originally been loaded.
As the internal File can be changed with several commands, this is a way to cahnge Files "In Memory" in a very fast way.

```
TXT.tov|$$LAD
DEL.$$LAA
WFF.<>$$LAA|10
PAU.0.1
TXT.sav|$$LAA
WFF.$$LAA|10
ITO.
  GTO.Lab_failed
EIF.
DEL.$$LAA
WFF.<>$$LAA
PAU.0.1
TXT.Save_File
WFF.$$LAA|10
ITO.
  GTO.Lab_failed
EIF.
```

Syntax

TXT.Sav[|P1]
TXT.Save[|P1]
TXT.Save_File[|P1]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **- (Input)** Variable with the Filename and Path where the Textfile shall be saved.

If omitted, the same Path where the file was loaded is used, previous version is overwritten without asking..

Example

```
'******************************************************
' EXAMPLE 1: TXT.tov  and TXT.fov etc.
'******************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
TXT.sav|$$LAA
WFF.$$LAA|10
 ITO.
   GTO.Lab_failed
 EIF.
 DEL.$$LAA
 WFF.<>$$LAA
 PAU.0.1
 TXT.Save_File
 WFF.$$LAA|10
 ITO.
   GTO.Lab_failed
 EIF.
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Save_as_Unicode 3422
- Push_Intern 3444
- Pop_Intern 3424
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9 Push / Pop

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.1 Append_from_Internal

MiniRobotLanguage (MRL)

TXT.afi
TXT.Append_from_Internal

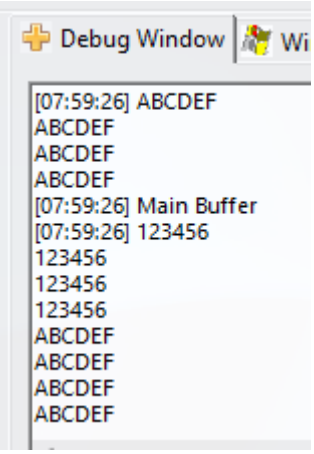
Append the Content of an internal Place that is numbered from 0 to 9 to the TXT.-internal Buffer.

Intention

This command is used to Append either the TXT.-Buffer or a specified Variable to the specified Memory (0-9).
Later the whole Memory can be restored using the `TXT.Pop_Intern` or the `TXT.Pop_Keep` - Command.
The Content of the TXT.-Buffer will NOT be changed using this Command.

```
' Generate 5 Lines of Text A
$$TXA=ABCDEF$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Generate 5 Lines of Text B
$$TXC=123456$crlf$
$$TXD=$$TXC$$TXC$$TXC$$TXC

' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB
' Save Reserve to internal Memory Place "0"
TXT.Push_Intern|0

' Load second Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXD

' Append it
TXT.afi|0
DBP.Internal Memory 0
TXT.dbp|0
DBP.Main Buffer
TXT.dbp
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

TXT.afi[|P1]
TXT.Append_from_Internal[|P1]

Parameter Explanation

P1 **-** **(Input)** Number or Variable with number of Internal Memory. Can be any Number between 0 and 9.
If omitted the place "0" is used.

P2 **-** **(Input,optional)** Text. If given this will be appended to the Internal Memory. If omitted the current TXT-Buffer will be appended.

Example

```
'******************************************************
'
'******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Append_to_Internal 3444
- Pop_Intern 3428
- Push_Keep 3446
- Pop_Keep 3437

© 2023 Theo Gottwald * https://www.smart-package.com

- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.2 Append_to_Internal

MiniRobotLanguage (MRL)

TXT.api
TXT.Append_to_Internal

Append the Content of the TXT.-internal Buffer to an internal Place that is numbered from 0 to 9.

Intention

This command is used to Append either the TXT.-Buffer or a specified Variable to the specified Memory (0-9).
Later the whole Memory can be restored using the `TXT.Pop_Intern` or the `TXT.Pop_Keep` - Command.
The Content of the TXT.-Buffer will NOT be changed using this Command.

```
' Generate 5 Lines of Text A
$$TXA=ABCDEF$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Generate 5 Lines of Text B
$$TXC=123456$crlf$
$$TXD=$$TXC$$TXC$$TXC$$TXC

' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB

' Save Reserve to internal Memory Place "0"
TXT.Push_Intern|0

' Load second Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXD

' Append it to first Text
TXT.api|0
TXT.dbp|0
ENR.
```

Syntax

TXT.api[|P1]
TXT.Append_to_Internal[|P1]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **- (Input)** Number or Variable with number of Internal Memory. Can be any Number between 0 and 9.
If omitted the place "0" is used.

P2 **- (Input,optional)** Text. If given this will be appended to the Internal Memory. If omitted the current TXT-Buffer will be appended.

Example

```
'******************************************************
'
'******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Append_to_Internal 3444
- Pop_Intern 3431
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.3 Clear_Intern

MiniRobotLanguage (MRL)

TXT.Cls
TXT.Clear_Intern

Clear the internal Places that may contain TXT.-Buffer-Backups and that are numbered from 0 to 9.
You can specify a number to clear that specified place. If you specify no number all places are cleared.

Intention

This command is used to clear those internal storage-places that are numbered from 0 to 9.
You can specify a number to clear that specified place. If you specify no number all places are cleared.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.Puk|1
' We have now a Backup of the Textfile in Internal Storage
"1".
' It can be restored - overwriting the current Buffer-Content
using:
TXT.Pop|1
' Clear all internal places
TXT.cls
```

Syntax

TXT.Cls[|P1]
TXT.Clear_Intern[|P1]

Parameter Explanation

P1 **- (Input)** Number or Variable with number. Can be any Number between 0 and 9.
If omitted all internal places are cleared.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Push_Intern 3444
- Pop_Intern 3433
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.4 Pop_Intern

MiniRobotLanguage (MRL)

TXT.Pop
TXT.Pop_Intern

Get the TXT.-internal Buffer from an internal Place that is numbered from 0 to 9.
The Content of the internal Place will be cleared.

Intention

This command is used to restore a complete Backup of the currently loaded Text from an internal Variable that is numbered from 0 to 9.
Therefore the Text must have been "saved" using the `TXT.Push_Intern` or the `TXT.Push_Keep` - Command.
The Content of the internal Place will be cleared after it was moved back into TXT.-Buffer.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.Pus|1
' We have now a Backup of the Textfile in Internal Storage
"1".
' It can be restored - overwriting the current Buffer-Content
using:
TXT.Pop|1
```

Syntax

TXT.Pop[|P1]
TXT.Pop_Intern[|P1]

Parameter Explanation

P1 **- (Input)** Number or Variable with number. Interger Number between 0 and 9.
If omitted the place "0" is used.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*******************************************************
'
'*******************************************************
```

Remarks

TXT.Pop is much faste compared with TXT.tov. Therefore in Scripts that should run as fast as possible,
this command is the better choice.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Push_Intern 3444
- Pop_Intern 3435
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.5 Pop_Keep

MiniRobotLanguage (MRL)

TXT.Pok
TXT.Pop_Keep

Copy the TXT.-internal Buffer from an internal Place that is numbered from 0 to 9.
The Content of the internal Place will not be altered.

Intention

This command is used to restore a complete Backup of the currently loaded Text from an internal Variable that is numbered from 0 to 9.
Therefore the Text must have been "saved" using the `TXT.Push_Intern` or the `TXT.Push_Keep` - Command.
The Content of the internal Place will not be altered, after it was moved back into TXT.-Buffer.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.Pus|1
' We have now a Backup of the Textfile in Internal Storage
"1".
' It can be restored - overwriting the current Buffer-Content
using:
TXT.Pop|1
```

Syntax

TXT.Pok[|P1]
TXT.Pop_Keep[|P1]

Parameter Explanation

P1 **- (Input)** Number or Variable with number. Can be any Number between 0 and 9.
If omitted the place "0" is used.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Push_Intern 3444
- Pop_Intern 3437
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.6 Prepend_from_Internal

MiniRobotLanguage (MRL)

TXT.pfi
TXT.Prepend_from_Internal

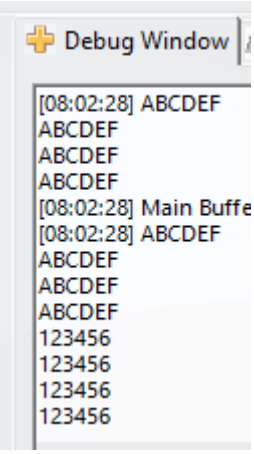
Prepend the Content of an internal Place that is numbered from 0 to 9 to the TXT.-internal Buffer.

Intention

This command is used to Prepend either the TXT.-Buffer or a specified Variable to the specified Memory (0-9).
Later the whole Memory can be restored using the `TXT.Pop_Intern` or the `TXT.Pop_Keep` - Command.
The Content of the TXT.-Buffer will NOT be changed using this Command.

```
' Generate 5 Lines of Text A
$$TXA=ABCDEF$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Generate 5 Lines of Text B
$$TXC=123456$crlf$
$$TXD=$$TXC$$TXC$$TXC$$TXC

' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB
' Save Reserve to internal Memory Place "0"
TXT.Push_Intern|0

' Load second Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXD

' Append it
TXT.pfi|0
DBP.Internal Memory 0
TXT.dbp|0
DBP.Main Buffer
TXT.dbp
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com



Syntax

TXT.pfi[|P1]
TXT.Prepend_from_Internal[|P1]

Parameter Explanation

P1 **-** **(Input)** Number or Variable with number of Internal Memory. Can be any Number between 0 and 9.
If omitted the place "0" is used.

P2 **-** **(Input,optional)** Text. If given this will be appended to the Internal Memory. If omitted the current TXT-Buffer will be appended.

Example

```
'******************************************************
'
'******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Append_to_Internal 3444
- Pop_Intern 3439
- Push_Keep 3446
- Pop_Keep 3437

© 2023 Theo Gottwald * https://www.smart-package.com

- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.7 Prepend_to_Internal

MiniRobotLanguage (MRL)

TXT.Ppi
TXT.Prepend_to_Internal

The Content of the TXT.-Buffer will be prepended to the Text that is already stored in the internal Memory.

Intention

This command is used to prepend either the TXT.-Buffer or a specified Variable to the specified Memory (0-9).
Later the whole Memory can be restored using the `TXT.Pop_Intern` or the `TXT.Pop_Keep` - Command.
The Content of the TXT.-Buffer will NOT be changed using this Command.

```
' Generate 5 Lines of Text A
$$TXA=ABCDEF$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Generate 5 Lines of Text B
$$TXC=123456$crlf$
$$TXD=$$TXC$$TXC$$TXC$$TXC

' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB

' Save Reserve to internal Memory Place "0"
TXT.Push_Intern|0

' Load second Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXD

' Prepend it to first Text
TXT.ppi|0
TXT.dbp|0
ENR.
```

Syntax

TXT.Ppi[|P1]
TXT.Prepend_to_Internal[|P1]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **- (Input)** Number or Variable with number of Internal Memory. Can be any Number between 0 and 9.
If omitted the place "0" is used.

P2 **- (Input,optional)** Text. If given this will be appended to the Internal Memory. If omitted the current TXT-Buffer will be appended.

Example

```
'******************************************************
'
'******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Prepend_to_Internal 3444
- Pop_Intern 3442
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.8 Push_Intern

MiniRobotLanguage (MRL)

TXT.Pus
TXT.Push_Intern

Move the Content of the TXT.-internal Buffer into a internal Place that is numbered from 0 to 9.
The Content of the TXT.-Buffer will be deleted after the Content has been moved.

Intention

This command is used to make a complete Backup of the currently loaded Text into an internal Variable that is numbered from 0 to 9.
Therefore the Text can be restored using the `TXT.Pop_Intern` or the `TXT.Pop_Keep` - Command.
The Content of the TXT.-Buffer will be cleared after the Content has been moved to the internal place.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.Push_Keep|0
' We have now a Backup of the Textfile in internal Place "0".
' It can be restored - overwriting the current Buffer-Content
using:
TXT.Pop_Keep|$$LAD
```

Syntax

TXT.Pus[|P1]
TXT.Push_Intern[|P1]

Parameter Explanation

P1 **- (Input)** Number or Variable with number. Can be any Number between 0 and 9.
If omitted the place "0" is used.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Push_Intern 3444
- Pop_Intern 3444
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.9.9 Push_Keep

MiniRobotLanguage (MRL)

TXT.Puk
TXT.Push_Keep

Copy the TXT.-internal Buffer into a internal Place that is numbered from 0 to 9.
The Content of the TXT.-Buffer will not be altered.

Intention

This command is used to make a complete Backup of the currently loaded Text into an internal Variable that is numbered from 0 to 9.
Therefore the Text can be restored using the `TXT.Pop_Intern` or the `TXT.Pop_Keep` - Command.
The Content of the TXT.-Buffer will be unchanged, after the Content has been copied to the internal place.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.Push_Keep|0
' We have now a Backup of the Textfile in internal Place "0".
' It can be restored - overwriting the current Buffer-Content
using:
TXT.Pop_Keep|$$LAD
```

Syntax

TXT.Puk[|P1]
TXT.Push_Keep[|P1]

Parameter Explanation

P1 **- (Input)** Number or Variable with number. Can be any Number between 0 and 9.
If omitted the place "0" is used.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*******************************************************
'
'*******************************************************
```

Remarks

-

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- Push_Intern 3444
- Pop_Intern 3446
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10 To LOF / To Var

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.1 From_Var

MiniRobotLanguage (MRL)

TXT.fov
TXT.From_Var

Restore the TXT.-internal Buffer from a Variable using an internal Format, that includes additional Informations, like the Filenname.

Intention

This command is used to restore the Internal TXT-Buffer "as if the file would have been loaded".
The content of the Variable must be created with the `TXT.To_Var` or the `To_Var_and_Clear` Command.
This Command-Set can be used to quickly switch between several Text-Files without loading them each time new from the harddrive.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.toc|$$LAD
' We have now a Backup of the Textfile in $$LAD.
' It can be restored - overwriting the current Buffer-Content
using:
TXT.fov|$$LAD
```

To transfer pure Text from a Variable into the Buffer, use the Command `TXT.Load_Pure_Text`.

Syntax

TXT.fov|P1
TXT.from_Var|P1

Parameter Explanation

P1 **-** **`(Input)`** Variable with the Textfile, this one must have been created using the `tov` or `toc`-command.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.tov  and TXT.fov etc.
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' Reserve speichern
TXT.toc|$$LAD
' We have now a Backup of the Textfile in $$LAD.
' It can be restored - overwriting the current Buffer-Content using:
TXT.fov|$$LAD
ENR.


'*****************************************************
' EXAMPLE 2: TXT.tov  and TXT.fov etc.
' Subprogram that removes Columns
'*****************************************************
:Lab_Remove_Cols
$$FIK=?exeloc\Test.txt
$$LIA="Q","M","K","I","H",F"
TXT.toc|$$BAK
TXT.Load_pure_Text|$$PAA|$$FIK
TXT.Delimited_Items_Remove|;|$$LIA
TXT.Get_pure_Text|$$PAA
TXT.fov|$$BAK
TXT.Get_Line_Count|$$NUL
RET.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.2 Get_from_LOF

MiniRobotLanguage (MRL)

TXT.gfl
TXT.Get_from_LOF

Restore the TXT.-internal Buffer from the File that was loaded with the LOF. 1063-Command
that includes additional Informations, like the Filenname.

Intention

This command is used to restore the Internal TXT-Buffer "as if the file would have been loaded" it takes the file and additional Informations from the LOF.-Command 1063.
As the LOF-Command can hold up to 128 Textfiles "in Memory", you will need to specify the "Index number" which the LOF.-Command has assigned to that file at Loading-Time.
The Text-File and Data will be transfered into the TXT.-internal Buffer. The LOF.-Buffer will be cleared!

```
$$MSG=Teste "to_lof"
' Clear LOF:-Buffer
LOF.!$$IND
LFT.$$IND|-1|$$LAB
JNZ.$$LAB|Lab_failed

' Transfer Text to LOF.-Buffer, clear TXT.-Buffer
TXT.tol|$$IND
LFT.$$IND|-1|$$RES
JIV.$$RES!20|Lab_failed

' Restore TXT.-Buffer from LOF.-Buffer, clear LOF.-Buffer
TXT.gfl|$$IND
TXT.gtl|$$LAB
JIV.$$LAB!1040|Lab_failed
```

Syntax

TXT.gfl|P1
TXT.Get_from_LOF|P1

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **- (Input)** Variable with the LOF-INDEX-Number of the Textfile that shall be transfered.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.tol  and TXT.gfl etc.
'*****************************************************
$$MSG=Teste "to_lof"
' Clear LOF:-Buffer
LOF.!$$IND
LFT.$$IND|-1|$$LAB

' Transfer Text to LOF.-Buffer, clear TXT.-Buffer
TXT.tol|$$IND
LFT.$$IND|-1|$$RES

' Restore TXT.-Buffer from LOF.-Buffer, clear LOF.-Buffer
TXT.gfl|$$IND
TXT.gtl|$$LAB
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.
The way the lines are delimited should be $crlf$, in some cases the loader will convert $cr or $lf-files to $crlf$ files.
To properly work with these commands, make sure that you work with this type of files that are $crlf$ delimited.

Limitations:

-

See also:

- LOF-Load Textfile 1063
- LFT-Line from Text 1060
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957

© 2023 Theo Gottwald * https://www.smart-package.com

- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.3 Get_pure_Text

MiniRobotLanguage (MRL)

TXT.gpt
TXT.Get_pure_Text

Copy the TXT.-internal Buffer into two Variables using normal Text-Format in Variable 1,
and the Filename in Variable 2.

Intention

This command is used to get the Text that is in the Internal TXT-Buffer into 2 Variables.
Variable P1 will contain the "pure Text". Variable P2 will receive the filename and Path.

This Command-Set can be used together with `TXT.lpt` to quickly switch between several Text-Files
without loading them each time new from the harddrive.

```
CFF.$$FIL|$$RES
' Load Pure Text
TXT.lpt|$$RE|$$FIL
' get Pure Text
TXT.gpt|$$RES|$$FIL
```

Syntax

TXT.gpt|P1[|P2]
TXT.Get_pure_Text|P1[|P2]

Parameter Explanation

P1 - **(Output)** Variable with the Textfile.
P2 - **(Output)** Variable will contain the Path and FIlename.

Example

```
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' EXAMPLE 1: TXT.lpt  and TXT.fov etc.
'*************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
CFF.$$LAA|$$RES
TXT.lpt|$$RES|$$LAA
TXT.gfn|$$LAB
NVC.$$LAB=Test.txt
  GTO.Lab_failed
EIF.
TXT.gtl|$$LAB
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- TXT.Load_pure_Text 3457
- TXT.Get_pure_Text 3455
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.4 Load_pure_Text

MiniRobotLanguage (MRL)

TXT.lpt
TXT.Load_pure_Text

Restore the TXT.-internal Buffer from two Variables using normal Text-Format in Variable 1,
and the Filename in Variable 2.

Intention

This command is used to set the Internal TXT-Buffer "as if the file would have been loaded" from 2 Variables.
Variable P1 must be the "pure Text". Variable P2 must contain the filename and Path.
The content of the Variable can be created with the `TXT.gpt` or the `Get_pure_Text` Command.
Or it can just be an Text that was loaded with CFF. and the filename and Path in Variable P2.
This Command-Set can be used to quickly switch between several Text-Files without loading them each time new from the harddrive.

```
CFF.$$FIL|$$RES
' Load Pure Text
TXT.lpt|$$RE|$$FIL
' get Pure Text
TXT.gpt|$$RES|$$FIL
```

Syntax

TXT.lpt|P1|P2
TXT.Load_pure_Text|P1|P2

Parameter Explanation

P1 **-** **`(Input)`** Variable with the Textfile, this one must have been created using the `tov` or `toc`-command.

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'*****************************************************
' EXAMPLE 1: TXT.lpt  and TXT.fov etc.
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
CFF.$$LAA|$$RES
TXT.lpt|$$RES|$$LAA
TXT.gfn|$$LAB
NVC.$$LAB=Test.txt
  GTO.Lab_failed
EIF.
TXT.gtl|$$LAB
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- !TXT. - Text File Engine 3252
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- TXT.Load Unicode_File 3420
- TXT.Load_pure_Text 3457
- TXT.Save_File 3424
- Push_Intern 3444
- Pop_Intern 3457
- Push_Keep 3446
- Pop_Keep 3437
- Clear_Intern 3433
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.To_Var 3462
- TXT.From_Var 3449
- TXT.To_Var_and_Clear 3464
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.5 To_LOF

MiniRobotLanguage (MRL)

TXT.tol
TXT.To_LOF

Move the TXT.-internal Buffer to the Buffer of the LOF. 1063-Command that includes additional Informations, like the Filenname. Clear TXT.-Buffer.

Intention

This command is used to transfer (move) the TXT-Buffer "as if the file would have been loaded" into the specified Buffer of the LOF.-Command 1063.
As the LOF-Command can hold up to 128 Textfiles "in Memory", you will need to specify the "Index number" which shall be used.
The Text-File and Data will be transfered into the LOF-Buffer. The TXT.-internal -Buffer will be cleared!
If you use a LOF.-Buffer-Inde4x that is already "in use", the file that was loaded with LOF. will be lost and replaced with the File that is transfered from the TXT.-Buffer.
The Index-Numbers for the LOF.-internal FIle-Cache are in the range from 1 to 128, so i recommend that you just pick a free number out of these if you need to use a new Index--Number.
Possibly in many cases you may just use the same number that was used to transfer the file from LOF. into the TXT.-Buffer.

```
$$MSG=Teste "to_lof"
' Clear LOF:-Buffer
LOF.!$$IND
LFT.$$IND|-1|$$LAB
JNZ.$$LAB|Lab_failed

' Transfer Text to LOF.-Buffer, clear TXT.-Buffer
TXT.tol|$$IND
LFT.$$IND|-1|$$RES
JIV.$$RES!20|Lab_failed

' Restore TXT.-Buffer from LOF.-Buffer, clear LOF.-Buffer
TXT.gfl|$$IND
TXT.gtl|$$LAB
JIV.$$LAB!1040|Lab_failed
```

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

TXT.tol|P1
TXT.To_LOF|P1

Parameter Explanation

P1 **- (Input)** Variable with the LOF-INDEX-Number where the Textfile that shall be transfered to.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.tol  and TXT.gfl etc.
'*****************************************************
$$MSG=Teste "to_lof"
' Clear LOF:-Buffer
LOF.!$$IND
LFT.$$IND|-1|$$LAB

' Transfer Text to LOF.-Buffer, clear TXT.-Buffer
TXT.tol|$$IND
LFT.$$IND|-1|$$RES

' Restore TXT.-Buffer from LOF.-Buffer, clear LOF.-Buffer
TXT.gfl|$$IND
TXT.gtl|$$LAB
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.
The way the lines are delimited should be $crlf$, in some cases the loader will convert $cr or $lf-files to $crlf$ files.
To properly work with these commands, make sure that you work with this type of files that are $crlf$ delimited.

Limitations:

-

See also:

- LOF-Load Textfile 1063
- LFT-Line from Text 1060
- TXT.Load Textfile 3418
- TXT.Clear 3416
- TXT.From_Var 3449
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015

© 2023 Theo Gottwald * https://www.smart-package.com

- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.6 To_Var

MiniRobotLanguage (MRL)

TXT.tov
TXT.To_Var

Copy the TXT.-internal Buffer into a Variable
in a internal Format, that includes additional Informations, like the Filenname

Intention

This command is used to make a complete Backup of the currently loaded Text into a Variable.
Therefore the Text can be restored using the `TXT.From_Var`-Command.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.tov|$$LAD
' We have now a Backup of the Textfile in $$LAD.
' It can be restored - overwriting the current Buffer-Content
using:
TXT.fov|$$LAD
```

Syntax

TXT.tov|P1
TXT.To_Var|P1

Parameter Explanation

P1 - **(Output)** Variable with the result of the operation.
On error a "" is returned also the Timeout-Flag is set.
On success, the Lenght in Bytes/Charcters of the currently loaded file is been returned.
If omitted the result is placed on TOS.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.gtl etc.
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' Reserve speichern
TXT.tov|$$LAD
' We have now a Backup of the Textfile in $$LAD.
' It can be restored - overwriting the current Buffer-Content
using:
TXT.fov|$$LAD
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.10.7 To_Var_and_Clear

MiniRobotLanguage (MRL)

TXT.toc
TXT.To_Var_and_Clear

Move the TXT.-internal Buffer into a Variable using an internal Format, that includes additional Informations, like the Filenname

Intention

This command is used to make a complete Backup of the currently loaded Text into a Variable.
After doing so, it will call the TXT.Clear-Command to clear the internal Buffer.
However the Text can be restored using the `TXT.From_Var`-Command.

```
' Load Textfile into internal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
' Reserve speichern
TXT.toc|$$LAD
' We have now a Backup of the Textfile in $$LAD.
' It can be restored - overwriting the current Buffer-Content
using:
TXT.fov|$$LAD
```

Syntax

TXT.toc|P1
TXT.To_Var_and_Clear|P1

Parameter Explanation

P1 - **(Output)** Variable with the result of the operation.
On error a "" is returned also the Timeout-Flag is set.
On success, the Lenght in Bytes/Charcters of the currently loaded file is been returned.
If omitted the result is placed on TOS.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.toc etc.
'*****************************************************
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
' Reserve speichern
TXT.toc|$$LAD
' We have now a Backup of the Textfile in $$LAD.
' It can be restored - overwriting the current Buffer-Content
using:
TXT.fov|$$LAD
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- TXT.To_Var_and_Clear 3464
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.11 Other Commands

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.11.1 DebugPrint

MiniRobotLanguage (MRL)

TXT.dbp
TXT.DebugPrint

Dump/Print the internal Buffer into the SPRE-Debug-Window

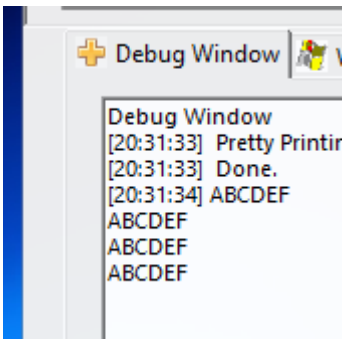
Intention

This command makes only sense when **used from the SPR-Editor**.
It will print the complete TXT.-Buffer into the Debug-Window of the SPRE.

There are two ways to use the command:

1. Do not specify any Parameter.
In this case the command will Print the complete TXT.-Buffer in the Debug-Window of the SPRE.

```
' Generate 5 Lines of Text A
$$TXA=ABCDEF$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Generate 5 Lines of Text B
$$TXC=123456$crlf$
$$TXD=$$TXC$$TXC$$TXC$$TXC

' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB
TXT.dbp
' Save Reserve to internal Memory Place "0"
TXT.Push_Intern|0
ENR.
```

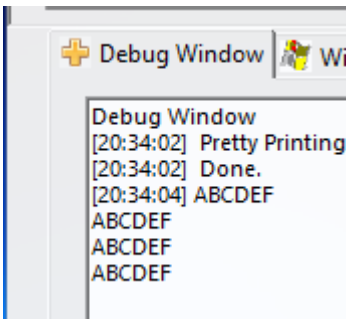


2. Specify an Numeric Parameter P1 (0-9)
In this case the Command will Print the content of the Internal Memory Locations (IML) 0-9.

```
' Generate 5 Lines of Text A
```

 © 2023 Theo Gottwald * https://www.smart-package.com

```
$$TXA=ABCDEF$crlf$
$$TXB=$$TXA$$TXA$$TXA$$TXA
' Generate 5 Lines of Text B
$$TXC=123456$crlf$
$$TXD=$$TXC$$TXC$$TXC$$TXC

' Load first Text into TXT.-Buffer
TXT.Load_Pure_Text|$$TXB
' Save Reserve to internal Memory Place "0"
TXT.Push_Intern|0
TXT.dbp|0
ENR.
```



Syntax

TXT.dpb
TXT.DebugPrint

Parameter Explanation

P1 **-** **(Input)** Number or Variable with number. Integer Number between 0 and 9.
If omitted the command will print the TXT.-Main Buffer.

Example

```
'*****************************************************
'
'*****************************************************
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- SHF. - Shell-File-Operation 1082
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.80.11.2 Get_Filename

MiniRobotLanguage (MRL)

TXT.gfn
TXT.Get_Filename

Return Path and Filename of the currently loaded File inside the TXT.-internal Buffer

Intention

This command is used to return Path and Filename of the Textfile, that was loaded into the internal Buffer of the TXT.-Commands.

```
' Load Textfile into intrnal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
TXT.gfn|$$RES
MBX.File loaded:$crlf$ $$RES
```

Syntax

TXT.gfn|P1
TXT.Get_Filename|P1

Parameter Explanation

P1 - **(Output)** Variable with the result of the operation.
On error a "" is returned also the Timeout-Flag is set.
On success, the Filename (incl. Path) of the currently loaded file is been returned.
If omitted the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.gfn etc.
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
TXT.gtl|$$LAB
JIV.$$LAB!0|Lab_failed
TXT.gfl|$$IND
TXT.gtl|$$LAB
JIV.$$LAB!1040|Lab_failed
TXT.gfn|$$LAB
NVC.$$LAB=Test.txt
  GTO.Lab_failed
EIF.
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- MDF. - Make-Direktory-File 1066
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.80.11.3 Get_Lenght

MiniRobotLanguage (MRL)

TXT.gtl
TXT.Get_Lenght

Return the lenght in Bytes/Characters of the currently loaded File inside the TXT.-internal Buffer

Intention

This command is used to return the Length in Bytes of the Textfile, that was loaded into the internal Buffer of the TXT.-Commands.

```
' Load Textfile into intrnal Buffer for further TXT.-Command
Operations
'($$RES is 0/1 success)
TXT.Load_Textfile|$$FIL|$$RES
TXT.gtl|$$RES
MBX.File loaded has a lenght of $$RES Bytes.
```

Syntax

TXT.gtl|P1
TXT.Get_Lenght|P1

Parameter Explanation

P1 - **(Output)** Variable with the result of the operation.
On error a "" is returned also the Timeout-Flag is set.
On success, the Lenght in Bytes/Charcters of the currently loaded file is been returned.
If omitted the result is placed on TOS.

Example

```
'*****************************************************
' EXAMPLE 1: TXT.ltf  and TXT.gtl etc.
'*****************************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$LAA=?path\..\Pics\Test.txt
TXT.ltf|$$LAA|$$RES
TXT.gtl|$$LAB
JIV.$$LAB!0|Lab_failed
TXT.gfl|$$IND
TXT.gtl|$$LAB
JIV.$$LAB!1040|Lab_failed
TXT.gfn|$$LAB
NVC.$$LAB=Test.txt
  GTO.Lab_failed
EIF.
ENR.
```

Remarks

Can acess very large pathes up to 32767 characters due to the use of **NTFS-direct**.

Limitations:

-

See also:

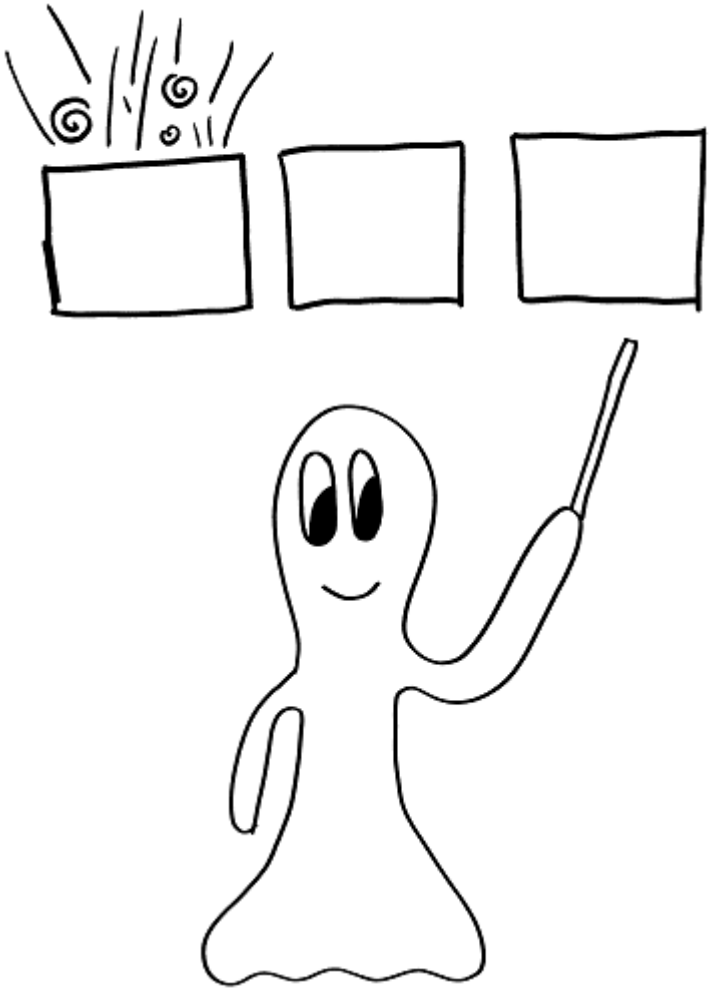
- TXT.Load Textfile 3418
- TXT.Clear 3416
- **OPT. - Optional Settings** 2406
- FIL.Generate numbered Name 1012
- FIL.Move numbered File 1015
- ATF - Append-to-File 946
- CFF. - Copy-From-File 948
- CTF - Copy-To-File 964
- COP. - Copy File 957
- TXT.Get_Filename 3470
- TXT.Get_Lenght 3472
- MKD. - Make-Directory 1068
- GLC. - Get-Line-Count 1043
- IEF. / NEF. - If - exist - File 1050
- LFF - Line-From-File 1056
- WFF. - Wait-For-File 946
- REN. - Rename File/Dir 1074

© 2023 Theo Gottwald * https://www.smart-package.com

3.81 Variable Definitions and Operations

3.81.1 ! Smarty's Variable Commands

Smarty's Variable Commands



Variables simple and complex in once?

What are Variables?

Generally with the SPR, a Variable is always 5 Characters long.
It starts with "$$" or with "§§".
The next 3 characters, can be any alphanumeric characters.
A typical variable is "$$TXT" or "$$NUM".

At runtime the Variable is been replaced with its internal content.

Therefore the Use of Variables is as Simple as:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
VAR.$$TXT=Peter
MBX.Hallo $$TXT
ENR.
```

In the same way you can use Variables for calculations.

```
VAR.$$NUA=5
VAR.$$NUB=7
CAL.$$NUC=$$NUA+$$NUB
' Result is 12
MBX.Result is $$NUC
ENR.
```

If you use Variables inside Variables, or Variables as Parameters for Subprograms, this is where it gets more complex.
Try this example:

```
' Variable Definition
VAR.$$TX1=5
' Numeric Variable Definition
VAN.$$NUM=1
' We generate a Variable-Name inside $$IND
VAR.$$IND=$$TX$$NUM
' This command is "Variable Indirect", we assign to the Variable Name inside t
VAI.$$IND=21
MBX.$$TX1
ENR.
```

Internally Variable Resolution is done ion several ways:
- String-Variables are replaced
- Numeric Variables and formulas are solved
- Special-folders are replaced
- System-Variables are replaced

Not every command does ANY of the above Variable-resolutions.
The reason is that if a Binary content may contain Variable-Names or formulas inside, it shall not be resolved.
That's the reason why there is a bunch of different Variable-Resolution Commands.

Starting with `VAB.` that does not resolve anything on the right side.
Next is `VAO.` that does resolve just "once"
Then comes the normal `VAR.` that does resolve most things.

And there is for example `VAI.` that assign something to the Variable, which name is inside the given Variable.
You can use it in Subprograms or Macros to return results in Variables.

Internally Variables have a Variable-Type. Most often you have not to deal with that.
Using `GVT.` you are able to find out what Type a variable has.
Using this information, you can then decide what to do with the Variable-Content.

Also take a look at `VBA.` that can save you a lot of trouble when dealing with unknown Content.

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.2 GVT. - Get Variable Type

MiniRobotLanguage (MRL)

GVT. Command

Get Variable Type

Intention

This command can be used to find out which Vartype was used for the given Variable In **P1**. There are generally 8 Types of variables that are available for Scripting (Type 7 should never happen):

```
   1 = Stringvariable (ASC-Text or Data)
   2 = LongInteger (32 - bit signed Number)
   3 = PointAPI (2 Integer Coordinates, e.g. X and Y)
   4 = Quad (64 Bit Signed Integer)
   5 = Extended Precission FLoating Point Number
   6 = Rect
   7 = unknown Stringtype
   8 = Empty string / Variable invalid and can not be resolved.

' Example:

' Here we define some Variables
' Extended FP Variable (Type 5)
VAN.$$ERT=15.05

' String Variable - (Type 1)
VAR.$$RTT=A

' Extended FP Variable (Type 5)
VAN.$$XTT=0.015

' Point-API Variable (Type 3)
VAP.$$MYP=12,13

GVT.$$ERT
DBP.$tos$

GVT.$$RTT|$$RES
DBP.$$RES

GVT.$$XTT|$$RES
DBP.$$RES

GVT.$$MYP|$$RES
DBP.$$RES

' This variable is undefined you will get a
' result zero, and the timeout flag will be set.
'
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
GVT.$$MYP|$$RES
DBP.$$RES

ENR.
```

Note that currently Variables will normally loose their Variable-Type (and get Type 1 String) if they are assigned to another Variable. You can avoid this by using the VAO.-Command:

```
 ' This will preserve the Variable Type
 VAO.$$ABC=$$EFG
 VAO.$$ABC=§§_01
```

Syntax

GVT.P1[|P2]

Parameter Explanation

P1 - Variable-name of Source and Target variable

P2 - ***(optional)*** Variable which will receive the result. If omitted the result is placed on TOS.

Example

```
'*********************************
' GVT.-Sample
'*********************************
VAN.$$PAA=5.56
VAN.$$PAB=7
GSB.Add|$$PAA|$$PAB
MBX.Result is: $$RES

VAR.$$PAA=Max
VAR.$$PAB=Müller
GSB.Add|$$PAA|$$PAB
MBX.Result is: $$RES

END.
:Add
' We use VAO. to preserve the Variable Type
VAO.$$TGA=§§_01
GVT.$$TGA|§§TYP
DBP. Vartype is §§TYP.
CAL.§§TYA=(§§TYP=2)+(§§TYP=4)+(§§TYP=5)
IVV.§§TYP=1
  $$RES=$$PAA,$$PAB
EIF.
IVV.§§TYA=1
  CAL.$$RES=$$PAA+$$PAB
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
RET.
ENR.
'*********************************


'*********************************
' GVT.-Sample 2
'*********************************
VAN.$$PAA=5.56
VAN.$$PAB=7
VAO.$$REA
%Add $$PAA|$$PAB|$$REA
MBX.Result is: $$REA

VAR.$$PAA=Max
VAR.$$PAB=Müller
VAO.$$REA
%Add $$PAA|$$PAB|$$REA
MBX.Result is: $$REA
END.
'------------------------------------------------------------
: %Add 3
GSB.Add|§§§01|§§§02|§§§03
END%

:Add
SAV.Save|$$ERG|$$TGA|$$TYP|$$PAA|$$PAB
VAR.$$PAA=§§_01
VAR.$$PAB=§§_02
VAO.$$ERG=§§_03
DBP.ERG=$$ERG
VAO.$$TGA=§§_01
GVT.$$TGA|$$TYP
DBP. Vartype is $$TYP.
CAL.$$TYA=($$TYP=2)+($$TYP=4)+($$TYP=5)
IVV.$$TYP=1
  $$RES=$$PAA,$$PAB
EIF.
IVV.$$TYA=1
  CAL.$$RES=$$PAA+$$PAB
EIF.
VAI.$$ERG=$$RES
SAV.Restore
RET.

ENR.
'============================================================
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

Note that currently Variables will normally loose their Variable-Type (and get Type 1 String) if they are assigned to another Variable. You can avoid this by using the VAO.-Command:

```
' This will preserve the Variable Type
VAO.$$ABC=$$EFG
```

The overall maximum length of variables depend on your available RAM and your actual Memory fragmentation

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.3 IEV. / NEV. - If exists Variable

MiniRobotLanguage (MRL)

IEV. / NEV.Conditional Statement

If exists Variable

Intention

Conditional Statement.
Test if a specified Variable exists.A Variable exists after it was defined anywhere in a Script. It does not matter if the content of the variable is zero, or if the variable is empty. Once it was defined, it exists.
A Variable can not be deleted, it will exist as long, as the Script runs.
Let's see an example:

```
' Here we define a Variable
VAR.$$ERT=15.05
IEV.$$ERT
  DBP.Variable exists
ELS.
  DBP.Variable NOT exists
EIF.
' Now we trry with a Variable that was not defined before.
IEV.$$ERX
  DBP.Variable exists
ELS.
  DBP.Variable NOT exists
EIF.
ENR.
```

As expected, the result is:

[00:07:24] Variable exists
[00:07:25] Variable NOT exists

You can only test for one Variable per command.
`NEV.` is the negative Form of `IEV.`

Syntax

IEV.P1 ELS. … EIF.
NEV.P1 ELS. … EIF.

Parameter Explanation

*© 2023 Theo Gottwald * https://www.smart-package.com*

P1 **-** Variable to test for. The variable can not be inside another
variable, always directly the given variable is tested.

IXX.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'**********************************
' IEV./EIF.-Sample
'**********************************
' Here we define a Variable
VAR.$$ERT=15.05
VAR.$$RTT=A
IEV.$$ERT
  DBP.Variable exists
ELS.
  DBP.Variable NOT exists
EIF.
' Now we trry with a Variable that was not defined before.
IEV.$$ERX
  DBP.Variable exists
ELS.
  DBP.Variable NOT exists
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- ELS. - ELSE 2150
- EIF. - END IF 2148

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.4 Inline Data Commands

What is "Inline Data"?
Inline-Data is Data that represents a picture, Password or Login Credentials, Text, or any other collection of Bytes.
"Inline" means that it is not an external file, but it is part of the Code.

You can now add Inline Data to any SPR-Script. The easiest way is to just enter the Inline-Data as Dezimal Numbers.
If you have ever used "BASIC" Language you may remind yourself to "READ and DATA" Commands.

With the SPR there is no separate READ and DATA Commands.
We only need the VDA. Command that starts the Block with Data, and then we need the VDE.-Command that ends the Block with Data.

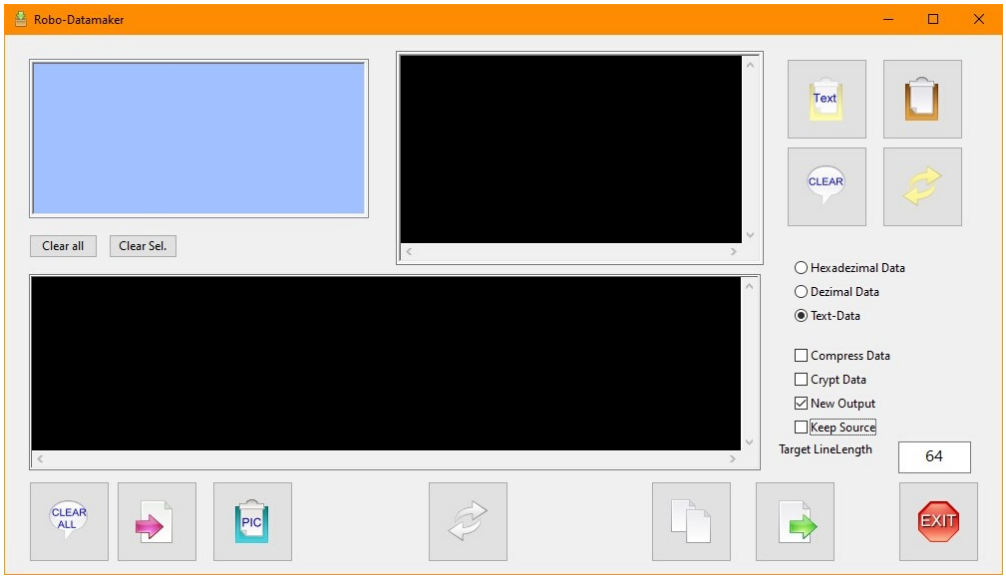
Technically when you jump to the Line with the Data, the Data is been put into the Variable after the VDA.- Command.

So the esiest way to use Inline Data is, to just enter it.

1. Just write your Data in dezimal numbers separated by ",".
This way you can manually enter data.

```
VDA.$$DTA|$$FIN=
64,65,255,0,21
VDE.
```

For all 3 further Options you will need to use the Datamaker.exe - Utility, which is in your SPR-Package included.



You can simply drag and drop files on to the Datamaker-Tool, to have them converted into Inline-Code.

The important Point is that the VDA.-Commands supports 4 Data-Types and additionally 2 Options.
You can simply set them in the Datamaker-Tool, the rest is done automatically.

© 2023 Theo Gottwald * https://www.smart-package.com

So lets take a look on the Datamaker-Tool.

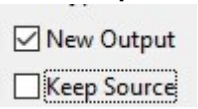
The Datamaker-Tool can get your Data from 3 Sources:
1. You can drag any file(s) into the Datamaker, they will then appear in the blue List-Box.
2. You can take Textual Data from your Clipboard and insert it into the upper black Box, using the PASTE-Button on the right side,.
3. You can insert a Picture from your Clipboard directly into the Datamaker. Use the blue "Paste Picture" Button for that.
I can recommend the "Greenshot" Freeware Utility to generate these pictures from Screenshots.
Using Greenshot you can directly put Screenshots into the Clipboard and make them to Inline-Data with just one Click.
Without saving it as a file.

Encryption and Compression

You can then process the Data according to the Checkmarks/Radiobuttons on the right side.
Therefore you can **encrypt** the Data (Good to hide Passwords). You can **compress** the Data (fast compression).
You do not need to worry on how to get the original data back, this is all done under the Hood from VDA.

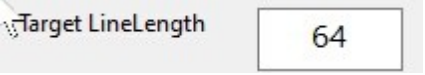
Further all Data is checksummed from the datamaker tool. The checksum is included in the data, so simple changes in the data will prevent the VDA. from unpacking the data. This will help you to keep your data safe.
Here is a hint: The encryption is always the same, with the same system, so it only to prevent people from reading the data visually.
Anybody who can let the script run, will of course get the original data out, by simply using VDA.

Further Options



You can further select if you want the processed files to be deleted from the blue List, or not ("**Keep Source**").
This will also influence the behavior of the Source Textbox.
In case you have dropped multiple files into the List, then you can convert them all in one strike.
For this you need to remove the "**New Output" Checkmark**, as this will generally clean the Output-Textbox before adding any new Code.

The Line-Lentgh for the Inline-Data



You can further decide the Line-Lenght of the Inline-Code. For this simply change the number in the white Edit-Box.
Default is generally 64, but you can make larger Lines. I did not test where the limit is.
Internally the Preprocessor will anyway connect all Inline-Data lines and store them in one place, so this may not have significant impact on the Runtime-behavior. You can

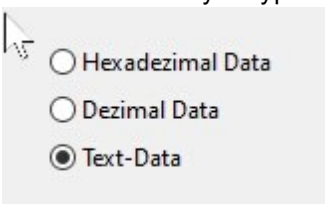
© 2023 Theo Gottwald * https://www.smart-package.com

change that for cosmetical reasons anyway. I suggest not to make lines longer then 255 Bytes because of the SPR-Editor (its something you can try).

The Middle grey Button with the Symbol is the "Convert to Inline Data" Button.
All items in the blue List will be processed according to the checkmarks when you click on it.

With Pictures from the Clipboard this is different. Because we can store the binary data easily, the pictures are immediately processed when you click on the "Paste Pic" Button.

There are currently 3 Types of Inline-Data supported. These are:



Generally prefer the "Text-Data" for most compact and fast processing.

Supported Data-Formats:

General comments:
As you can see, Dezimal Data starts with the Sequence "`?D". Then there is a `0,1` which means the Data is compressed but not encrypted. All this is done automatically if you use the Datamaker-Tool. The Data that has been created using the Datamaker-Tool also includes the Original Filename of the File - which will be in $$FIN - and a checksum to make sure that your data was not changed.
The chacksum is automatically calculated when the data is extracted. If the Data was changed, the VDA. will not extract the data and set the Timeout Flag..

1. Manually entered Dezimal Data: These are dezimal numbers between 0 and 255 separated by a ",".
This way you can manually enter data. `$$FIN` will be empty because this data can never contain a filename.

```
VDA.$$DTA|$$FIN=
64,65,255,0,21
VDE.
```

2. Dezimal Data: These are dezimal numbers between 0 and 255 separated by a ",". Each number represents 1 Byte of Data.
As you can see, Dezimal Data starts with the Sequence "`?D".

Looks like this:

```
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  386 Bytes in  7 Lines.
```

 © 2023 Theo Gottwald * https://www.smart-package.com

```
VDA.$$DTA|$$FIN=
?D0,1,110,0,0,0,74,81,236,212,59,216,196,99,142,123,50,231,210,17,63,
92,23,69,204,17,252,33,210,155,44,235,50,194,135,6,208,22,59,220,
54,112,63,212,166,150,66,0,107,254,33,79,44,112,112,48,228,80,57,
66,0,107,123,27,22,136,95,124,168,167,215,151,155,168,210,137,135,
92,42,62,194,173,226,6,148,10,42,36,169,15,60,40,235,88,44,96,229,
181,175,53,99,103,6,66,152,10,13,86,58,70,1
VDE.
```

3. Headezimal data. These are Hexadezimal numbers, where each uses 2 Digits therefore there is no Separation between them.
Hexadezimal Data starts with the Sequence "`?H".

```
' F:\00_MR\PB_MiniRobots-Editoren\SPRE 02\Source\Sample Scripts\Tests\File.txt
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  232 Bytes in  5 Lines.
VDA.$$DTA|$$FIN=
?H00016E0000004A51ECD43BD8C4638E7B32E7D2113F5C1745CC11FC21D29B2CEB32
C28706D0163BDC36703FD4A69642006BFE214F2C707030E4503942006B7B1B1688
5F7CA8A7D7979BA8D289875C2A3EC2ADE206940A2A24A90F3C28EB582C60E5B5AF
3563670642980A0D563A4601
VDE.
```

4. Text-Coded data.
This Format starts with "?M". This Format uses least Space because it uses all Textual Charactes to store the Information.
Please prefer this datatype for larger Data-Amounts, as it unpacks very efficiently and uses least space.

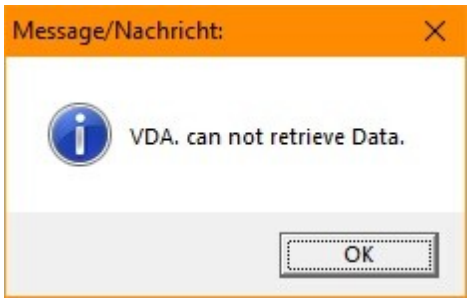
```
' F:\00_MR\PB_MiniRobots-Editoren\SPRE 02\Source\Sample Scripts\Tests\File.txt
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  154 Bytes in  3 Lines.
VDA.$$DTA|$$FIN=
?MAAFuAAAASlHs1DvYxGOOezLn0hE/XBdFzBH8IdKbLOsywocG0BY73DZwP9SmlkIA
a/4hTyxwcDDkUDlCAGt7GxaIX3yop9eXm6jSiYdcKj7CreIGlAoqJKkPPCjrWCxg
5bWvNWNnBkKYCg1WOkYB
VDE.
```

If Data that was created using the Datamaker-Tool was changed and therefore the **included checksum does not match**, you will get in $$DTA the Text "VDA. can not retrieve Data.".

```
' Sample that leads to the Error Message "VDA. can not retrieve Data."
' This data is invalid therefore it will not be extracted.
GSB.Laa
MBX.$$DTA
ENR.
'-------------------------------------------------------------
:Laa
' F:\00_MR\PB_MiniRobots-Editoren\SPRE 02\Source\Sample Scripts\Tests\File.txt
' $$DTA = Data
' $$FIN = Filename of Sourcefile
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Length:  154 Bytes in  3 Lines.
VDA.$$DTA|$$FIN=
?MAAFuAAAASlHs1DvYxGOOezLn0hEdw/XBdFzBH8IdKbLOsywocG0BY73DZwP9SmlkIA
a/4hTyxwcDDkUDlCAGt7GxaIX3yop9eXm6jSiYdcKj7CreIGlAoqJKkPPCjrWCxg
5bWvNWNnBkKYCg1WOkYB
VDE.
RET.
```



© 2023 Theo Gottwald * https://www.smart-package.com

3.81.4.1 Inline-Data Sample

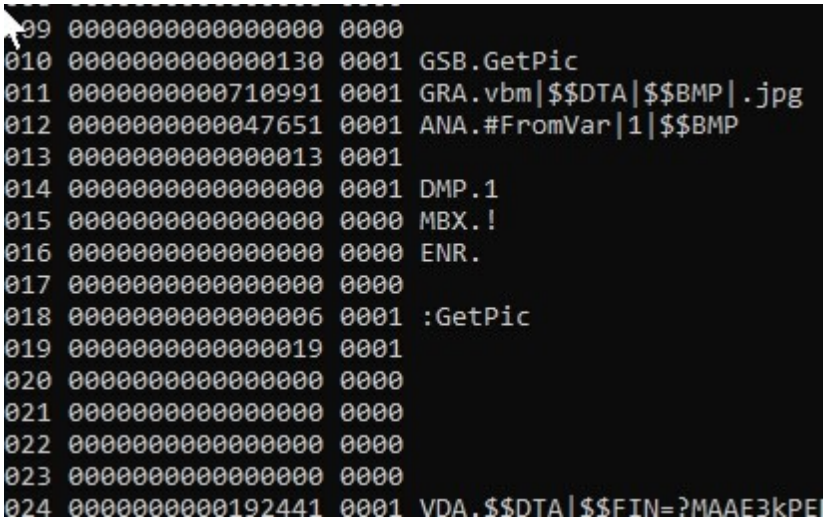
MiniRobotLanguage (MRL)

Inline-Data Sample and Speeddump

If exists Variable

Intention

Inline Data is not really limitted in Size unless it stays in the 32-Bit Memory limit.
Also there is no Speed-Problem.
Below is the Speed-Dump of 245kb of Inline-Data (see below). It just uses here 1/83th of a Second to get extracted.
Using Inline-Data, you can add all Sorts and supported Picture Formats directly to your Script.



Using Inline-Pictures and ANA-#FromVar

You can useall sorts of Inline-Pictures, no matter which Format they have.
BUT, as ANA.#FromVar will only accept pictures in a very defined Bitmap-Format, it is necessary to convert all other formats to this Format first.
Use the `GRA.vbm|$$DTA|$$BMP|.jpg` command to get the Inline pictures into the right Format.

It may also be necessary to use this command on ".bmp"-Inline Data, "Files", depending on how the files are internally organized.

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this case, just convert from "-bmp" to "bmp".

```
GSB.GetPic
GRA.vbm|$$DTA|$$BMP|.bmp
ANA.#FromVar|1|$$BMP
ANA.Show|1!
ENR
```

Let's see an example:

```
'#EXE:?path\
'#SPI:ForceWrite

GSB.GetPic
GRA.vbm|$$DTA|$$BMP|.jpg
ANA.#FromVar|1|$$BMP
ANA.Show|1!
DMP.6
MBX.!
ENR.
'============================================================

:GetPic

'------------------------------------------------------------
' .., Tests\NFT_Testpics\P2.jpg
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  245138 Bytes in  1477 Lines.
VDA.$$DTA|$$FIN=
  ?MAAE3kPED/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQECAgMCAgICAgQDAwID
  CAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAj/wAARCANsAt8DASIAAhEBAx
  CQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2
  lJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09f
  NOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiI
  /90ABAAF/9oADAMBAAIRAxEAPwD7l/4huhuAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  /ksP/kQ/sTDfy/i/8z+Xv+Xv8AMd/AMd/986Hzo/wCIbj/wCIbj/wCIbj
```

```
39aP+Ik53/z/AP8AyWH/AMiH9kYf+X8X/mf/1vuX/iG2/wCry/8AzHf/AN86P+Ibb/q8v/zHf/3zr+
751/ULk/3f1oyf7v60f8RJzv/n//AOSw/wDkQ/sjD/y/i/8AM/l6/wCIbb/q8v8A8x3/APfOj/iG2/
Nt/1eX/5jv8A++df1C5P939aMn+7+tH/ABEnO/8An/8A+Sw/+RD+yMP/AC/i/wDM/l6/4htv+ry//M
y/8AzHf/AN86P+Ibb/q8v/zHf/3zr+oXJ/u/rRk/3f1r+Uf+Ik53/wA//wDyWH/yJ9x/ZGH/AJfxf+
M/l6/wCIbb/q8v8A8x3/APfOj/iG2/6vL/8AMd//AHzr+oXJ/u/rRk/3f1o/4iTnf/P/AP8AJYf/AC
+RD+yMP/AC/i/wDM/l6/4htv+ry//Md//fOj/iG2/wCry/8AzHf/AN86/qFyf7v60ZP939aP+Ik53/
Ik53/wA//wDyWH/yJ9x/ZGH/AJfxf+YlFFFfDHrhRRQAUUUUAFFFFAH/9H+2miiiv4YP08KKKACi
opgfJ5GKdkeooAWijIPQ5ooAKKKAP/U/tpooor+GD9PCiiigAoopcDGMGgB1FReZ/s/rR5n+z+tI
1vhocxlVbtZHnXxS+JHhz4ReCde8f+L3uovD1hhrqSGIu0UZIXeqDlhuPPvnHTna+HHxC8EfFfwhpH
9rfxJ+y348h1fU7i4vfgzqrxP4lsHkYxWbHrqEK8gOoIMgUfMi5wSBXvfVYQUY1NObY/YfD7wbxPFC
/SiWNkYnP/16/jLF5a4rmP0SLdmpKzWhUQOC245HapKUgjrSAg9K8Q0CiiigAooooAKKKAP/9f+2u
ooAbRRRTNwooooAKKKAP//S/tpooor+GD9PCiiigAooooAKKKACiiigD//0/7aaKKK/hg/Twoooo
cv2hdYu/25f2af2Y/BOpXtvbRyXHiLxa8DhFeFbWX7NayPu3YZlMjDnjy+MMK7cNhufU6srwFbGTko
AZNdK2Ef/9X+2miojJ025FIHYkDNfwtzo/TyaiiiqAKKKiL87QWB9MUAOdcjgc1CQR1GKmG8n5QwNC
haKZhwRzn1p2xgqtkmrcWtw5kLVWpTKvIyc1EOelY1BpocvUVcgUmUnHHFVArDBxWjbHkE9eld+WK9
9xcSEdCqoSQT2r+IzRB52l25eMISnm4Bzw3I5r+TeIIKFKl8/wBD/VX9nYufH5k7XXLD7/esfuv/AN
CX4oeFfiX4XuNviLw5qSXjKODfWe4faoAPWSFpUXPAbae1f2k+CPF2g/ELwT4Z8ceGb+PUdC1eyg1C
Gwe9fCYmm4SaZ/Lilcs0U1SSOadWAwooooA//9D+2uiiiv4XP0gKKKACiiigAoooAKKKAP//R/t
CiiigAoooAKKKACiiigD//1P7aaKKK/hg/TwooooAKKKACiiigAoooA//9X+2miiiv4YP08TsP
Nfy5/s2fFe58df8ABQXwJ8ZdZ1GRT4o8YanJH5iB3jtZrOWO1hD8YVIYrdRgfeUnvz+5n/BRf4h3vw
KpTjR6M/tL6Kvhb/AGrw7nmZSjdypypw9eW7t6tI/tYcDaOBWfMR8y4qfS7201jTdP1Gynhu7S5t0u
gHBBqQOT6V/CsXfQ/UBjKV64pVByCBTkV5SABWilv8o+X5sVtDCuT0IlUSKB3A5LcewqzFGWznk/St
aa38XfEbwV4bvIx80F5qcMcwPYCItuP4CvosHwdVlryuxjLGQf2js47F93VvwqU6exJDBq8VuP2kvA
nwU2c9XHJbM//9f+4VdPYcDOKf8AYHP97H0r46u/2kvjqvlRw/s4R2EhTcyX3iGCNoj0KnbG3OSv5l
X7VV9FE9x8H/AIX2ztMflfxXOP3Py7X/AOPXqfn49hzzWxB47/aUhtnuL74NeAp3JHlR2/i8guMkHJ
2KffftN6Boe8eK/hx8Z/DiIFEkp8NT3USszYAD24kyOuWAwOK+fx3hziqdnyfkdEccrantD2m3gx4P
9Mnw30457V81jeDK8d4P7jeOMp33KclqQuRHj61F5BA+YBRXRSRoQPLdZOccMM5/OoGtgwRl2kHpgf
VH4J/Za+O/iNZVtprfwterGx/wCejI6g7uo5cdK/jqsUXTtNsLaNjGEhVTz0HYZ+hFf0v/8ABX/4ip
j8dRRT8ox/zbORvdVuNO1WTVIQ8N2C7ojnBzg4zjtX9OX/BGz4z/APCa2/hz4ZaBqHmaZ4fiXUtY
NiNryIG4t2x3JCXC/wDAwOa5+GcW4VvZt6S0P2j6bnh2s54Jq4unG9XCvnT68q+JfM/rjuIsBiMjIr
CiiigAooooAKKKAP//S/tpooor+GD9PCiiigAooooAKKKAP//R/tpooor+GD9PCiiigAooooAKKKAP
ooA//9f+2miiiv4YP08TsPCiiigAooooAKKKAP//S/tpooor+GD9PCiiigAooooAKKKAP//R/tp
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

nPIimOcRfRgNwb+9le4x1gzg7sL/AEr7A5yG5uZLV490bNEcfN/d+tXe2D0BB/EVDuVvlY8HqcZIH4
dSVDRsFPlzbWKiTIYD+Fl3A+w/aJCVAwQVDfnQ0hYDOKzlBSVi6dblkpLp23P5Mv2+P+CPNtbR6r48
gOhG4Mc5xyuMYOOv+oNd6Ppmow3dlqFrDfafeKY7u2mQPEwwQcrzuDrgHPIIzyMY/DX/AIKD/wDBLr
RQ+ndjshlSyLiqcquCjaMajd5QW3vae8vxXmfx+eGviMJ3hsNcV4bknYsw+6fqO1esLcBsujo6FTgk
kx/q8/zHXkV/JWNw1bDVXRrxcZR3TP8AaOljMJmuFjj8umqlKrFNNbWevy2/E//T/pK/4Is+IZPEHT
1/L2at/VY37H9BeMuX/VOLsyw/SNep/6UzBbov1NNpzdF+pptfnUtz4mnsFFFFIsKKKKACiiigD/1F
CiiigAooooAKKKACiiigD//1/7aaKKK/hg/TwooooAKKKKACiiigAooooA//9D+2miiiv4YP08KKF
z//R9q/4KFRtpH7Xf7VsHn3rO2u2swEpzgPptoeP9n5uB6CvzWvJym0hua/Vf/grXpEnh79s/wCOTI
LuEA5JryPxR42gsv9GtnjaZyU5UNnPX5T169OPqOtVPGvjl4LdrXTwAxOGc9T7V5h4C8FeNPjD8SPF
JX/T57H0H+y34R/aN+NPx/8ACXgL9mXVPHmi/FC7lEialot3NBJplosgMk8s0eCI0cjduOGZl79f7w
gsE0/T9X+PeuW0UninXFTLXLjkW8bHlbWP5widGJZjlixr7EtbabxHcNPLPIumjIYnhpiTztHQDGB0
JJJz1OffNdFFaSKoG95COrMACfyptjbRWkSxW0flQA4VQOFGOnv9evrnrWjX6RKCZ/KtNaFZ3SCJXc
ziMBCckZxXkPw1nm1+68feMbhvMiudUGl2R6g29sg3sp95ZHU/8AXMeldL8Stfi8P+HL29eZYWSJ2k
2rI4lN3GuwIAGc15vour3B8beKdAcMqxxwXcWRkbGDjr6bo8E+9eiEc7ueK8D1HXDp/x88N6T9rhjC
AOK2j/En4f6DrWnSupl89Wjk4eFklaN4iDzuRotjZ53KT0Ir0/UN39n3mZlT92cueijFflL8HviK3q
aXl7oN9Dp06WupbDJbTMOIZl+ZGPtuAz6jI70ngvxGnizQdO1aP/AFrrtljxhopV+WVSPZw4HtiqPh
aWmZJuLtuf/T/tdj4UZIFScf3hWZExkxKH3BwGzjAPHYDjH04PWryfdFfXcl9zz6dR2KGuab/bGlXc
eli+HvxV8M+J0je28MeKUXRNQ+Y+XHqMas9tOFHG5keWBmPU7B1WiN+p0nvSKwAycim3Vpbaha3Fjc
qFttkimwT9qgY4R+OAQfkI9R7134cEkY9snivOfGukXlzpw13S0T/hJtNLXVqdxjScY+eKTH3lf5Bq
ykchgeoI4x71DG+SeAD1yO9OCFgVchlPr6UcmmhpTquDvHc//9T+hP8A4KE/8E5fAH7Wnw5Hh6O3sP
B0YEMHXKuMEYHFf6d86C5szbzKLhywJLDO4g5LexJ+bjoelfjt/wVd/4Jr2P7Z3w3i8f/Dqy0ix/aC
9E15J7/eYv8Awb1RvD+wmZ5YpY3Pi7Vw7MR8+JAoIGOBhRxX7f6j1HHPNfkr/wAEPPh7q/gD9gbwXp
5oTq1GmtmubRnPNyq8cZNRk46g1O67eMDOagkI4r8wkrOx8RRm7DqKKKR0BRRRQB/9X+2miiiv4YP0
CiiigAooooA//9D+2miiiv4YP08KKKACiiigAooooAKKAP//R/tpooor+GD9PCiiigAooooAKKKKAP
X/YvxS+CHxGgtn8rWtEvNCupgvyiS1mjnhDMOhKXNzjJ6K1fzTePvFw0uL7PbnEr5PynJAr+1r/gtv
iFOP2kf7i/s9+J6eacFRy9ztPCylG3k3zL7/ANNCjc3bq8l5LcT7B8xYNtKAc5B7H3r+zP8A4Ibf8P
ML74EeBLiK/1BZ1zb6tqCsHhtDkEMgKbpB0K/KfvV/aLqrTfETX00u38qw8B6f8AupRGPLM5XgQxAf
N+RgFQD92NcAe56cZJ9TjxCFCIqoo2gAAYHoMDp7U2K3t4h5NtEsUCKEVVAACjgAAdh2qyIcnG39a/
uqxR6DYSDgxT3ki2yuPZTMX/AOAGsakuU6oPQ//T/sg+D13d+IPD2pePrpisXiDVLm+sSwAY6YjCK1
Odxb0R8UftofEG38JfD3WZHnXfLF9gt04PmzTYjCgdyNxJA6CvrPTNOXSNH0nQoflhsrSC3UdtqRr0
vk0o5jCriJ4WD/htR+drs9HOsrlhcNQnPeonL5aJfqcbBry32r6zp8DCSK0McZx/CzIGIP51sKpkGT
P/JWvrYsTyeTXyz4/QH9qn9kazVC5k1PX5GIH3VGg33H1yU/AEV04Kooqd+kWRCV3qfRF+d0UqgDJU
37Qmh39xM1xcat4W1WIjafn8i+glHPqRdnFfO8d874crYyj/ABKPJOPqt19x+reDKpT4joYWtG8Kqc
4Nv4fFNlcQE74o4Q32hkI4ZVhMrhT8rGMV+d3w5uH+H3xV8QeEQLq5UeFfGSvf2NuScR6xGpFxtXoqz
d/F3C1TIM5nhJq8Iy0v1i9Uy7+zL8Z7X4ueC7O6LRWeswq8F1aBsrZyRqMjP/PI5jdO5jdeuCa+qA5
pFtq0RCM+VlTduMcoPzDPfkHmvMynGRxlFzjpKLcZeTX+ZtxXw/UwOIUlrSnFSg/5k+nyO2jkIOR0S
S3PlPavdnm/wV+JOm/FX4b+HOlCeKO/g82WGfIlt7jJWaFweQY5kmjI6ApXrrfIFOM5r4g+DOrxe
8rDgqeh5+mR+Jr50fWB8L/i3Hol08UPg7xQsk2nyPId1tqSDfLB04WRFaVW6B1kB+8K+hsV5l8XPBM
IfiJ4D0PxFaVW6B1kB+8K+hsV5l8XPBMIfiJ4D0PxFeQyWeqsphvrdh81vdoSksOOuUZHGDXsySuQpO3PtX3FWk07HHGsmaiEnC/rUzEDO4jbI
bIfuwXDlY5D/AASNG5+VmNfU8oMysVZJU7MDkGuJ1rRNN8S6FregazAl3pN5bSW9zE3IkjZGDA+vHP
nZM9VurcqWI4BqhycZGDXQ3aybtvAA7EDcPr71hNHtYkjBr+WcywbpP3j6nCTUkFFMQ8Yp9eQdh//9
TwoooAKKKKACiiigAooooA//9H/2miiiv4YP08KKKACiiigAooooAKKKKAP/0v7aaKKK/hg/TwooooA
ooooAKKKKAP/0/7aaKKK/hg/TwooooAooooAKKKKAP/U/tpooor+GD9PCiiigAooooAKKKKAP/V/tp
ooor+GD9PCiiigAooooAKKKKAP/W/tpooor+GD9PCiiigAooooAKKKKAP/X/tpooor+GD9PCiiigAo
ooooAKKKKAP/Y/tpooor+GD9PCiiigAooooAKKKKAP/Z/tpooor+GD9PCiiigAooooAKKKKAP/a/tp
ooor+GD9PCiiigAooooAKKKKAP/b/tpooor+GD9PCiiigAooooAKKKKAP/c/tpooor+GD9PCiiigAo
ooooAKKKKAP/d/tpooor+GD9PCiiigAooooAKKKKAP/e/tpooor+GD9PCiiigAooooAKKKKAP/f/tp
ooor+GD9PCiiigAooooAKKKKAP/g/tpooor+GD9PCiiigAooooAKKKKAP/h/tpooor+GD9PCiiigAo
ooooAKKKKAP/i/tpooor+GD9PCiiigAooooAKKKKAP/j/tpooor+GD9PCiiigAooooAKKKKAP/k/tp
ooor+GD9PCiiigAooooAKKKKAP/l/tpooor+GD9PCiiigAooooAKKKKAP/m/tpooor+GD9PCiiigAo
ooooAKKKKAP/n/tpooor+GD9PCiiigAooooAKKKKAP/o/tpooor+GD9PCiiigAooooAKKKKAP/p/tp
ooor+GD9PCiiigAooooAKKKKAP/q/tpooor+GD9PCiiigAooooAKKKKAP/r/tpooor+GD9PCiiigAo
ooooAKKKKAP/s/tpooor+GD9PCiiigAooooAKKKKAP/t/tpooor+GD9PCiiigAooooAKKKKAP/u/tp
ooor+GD9PCiiigAooooAKKKKAP/v/tpooor+GD9PCiiigAooooAKKKKAP/w/tpooor+GD9PCiiigAo
ooooAKKKKAP/x/tpooor+GD9PCiiigAooooAKKKKAP/y/tpooor+GD9PCiiigAooooAKKKKAP/z/tp
ooor+GD9PCiiigAooooAKKKKAP/0/tpooor+GD9PCiiigAooooAKKKKAP/1/tpooor+GD9PCiiigAo
ooooAKKKKAP/2/tpooor+GD9PCiiigAooooAKKKKAP/3/tpooor+GD9PCiiigAooooAKKKKAP/4/tp
ooor+GD9PCiiigAooooAKKKKAP/5/tpooor+GD9PCiiigAooooAKKKKAP/6/tpooor+GD9PCiiigAo
ooooAKKKKAP/7/tpooor+GD9PCiiigAooooAKKKKAP/8/tpooor+GD9PCiiigAooooAKKKKAP/9/tp
ooor+GD9PCiiigAooooAKKKKAP/+/tpooor+GD9PCiiigAooooAKKKKAP//2Q==

*© 2023 Theo Gottwald * https://www.smart-package.com*

PMEk+Qc9RxXXeEtWttT0exvbd0khliWRMNk4IB/rXZRkqtPm7aBTm0ehq2cMMirUbnPbpWPFKGUDgf
nODG/wDer0tDwx7Cuv021tdc0XVNA1KKC5spEJKSDKuuMMrexyBj3NeXj8MpQals9zrpVLPQmSSHUI
AC0gYdq9mvIiFYKS3PU96/kzj7h6dCvJR1V9PQ+3yzEXSP/X/tmTGWxwKfUTblZsLShmJAYYFfww1y
TwoooAKKKKACiiigAooooA//9L+2miiiv4YP08KKKKACiiigAooooAKKKKAP//T/tpooor+GD9PCi
ScbgKkasoJyGFaFjEmRkgDrzVGFSHIbnNb1rCiquMqPavcybDXqp2uedi521OU+JvjGDwB8PvFHilk
gYG4ij9pLSr74vePfhL+zrod1Jb2N80viPxZPE5X7HpEBKIsnH3ppWKJg5/dyH+E4+jvEF7a6fZadc
BdXkfdIMcrGMEk/gaz4beK1gWNUMiAdM8tW/4RuzOfE/iQTpMkbNYW5UcMykBz/32cf8Ar7evG1O6C
4jY3eq2VuAzlQczLkZ+gPHfivr8LT2Rgbl3eR6V4WtbSQB3dFi246AYH8lNfP37OOkK/xg/ak15pDN
4V6mLp8lBjPpySdW82T5Y1ALHJwFHPr9K/Fr9pXWz41+NHw20tybq1s5NR8VuzfMFaNRZW6ndkgn7V
RRD5nbfOMqPmUZYgAE18TxzjfqnDeJnF+9Ncq+/Y/SfB/KI4viCg5/DB81+2hp+MjJZeH9SvreOaV
WAeHdGcM2M85I47V+mXwU16bxN8IvhnrMvn+dcaBYSOZDuYyCMK2T9UPHrn0r8i+j1VccPUoPdO7R
J/2p7ZpdJ0GcgCQtLHGxOMNsyv48Gvrn4deJU8efCjw34wZy1/Np0dveKDkrdRMFk3ehyCa/qDN6De
eW4nnX3JUQ4AORg+tfR0jyMqNITNgbAo6jAGB+uPqK+WfgTrEVzL8bPGUkiWkV74tv2BcBd0Nqsdsj
eFTo0H7OfiKVRBcX3ja0kChcOY1YooIPc+Zu/wCBeua1vA1rqnx1+LfirxsLVoU1zWGh09zIW26PbK
yZhRqvaMos9P8W+PPB2lLHZa3ZeMbVpV80KNBu5VRfu43RxsBzG3X1r8wf2rPib8IfDVp4U+J1t420
O67p1tqun3ttLb3EMkYYTRupV1xjncrMPxr+EsnxFXL8fRx1HWVOV1566r7j+q6eeUquEqYOsvdqJr
MmQUljRmUjn7uWX2r4b8L/CjwFqNnqvgvxt4I8La34i8PX93oEtzLpkaTyi3Obdg6gOheLyWyCByet
eCvFH9j5hWwk7qD0aPOf2SPHEnw/8R+IvgTq88q2+nhZ9Kkmfm702V2+zsmcZaFlkgYehQ9WFfsH4c
+gnwO+OfxVXRLa78Q/C2PxTpU1uswvPC2qR3BcDgu1tOY3PBU4UknIxmvS4Ez9Zvk0ZVP41H3Z+bXX
6XY6nrfgDwdrl/8ACD4q2IhbdJoNjfOlhqIJwzPbw+SQejIzKTg5H9CWgfHD4YeMrpvD1rrraT4rIT
Moz3B/XsljRnCUamzOKjG7tJbH7yeBNS8K/EjwDYePfCOr2HiXw+XiBKIvbR0DLNu/iyDk
DI6K+8BcgBHHYZ/okg1CVbyzvLuRpxtSNmbqVIxz+dfI4nLq2HquM5XXQwxuHVOUXHZjfEutaNpfhe
Uz/gtt+0hP8AsrfsA/EzxNYTE+K9ds28H6PFG+2S4ubl9o2Z5LBS5GM9K8Y/YW/Z+i/Y++Df7LXhF9
ps5ureK4A+U/4VsROBitpQs7HBRcjo9HKzXUlnIiEyoVVvQ13fhqNxZ3DjAngmEe7H3x6V5lZ3CW9
lJvOQH0uT6CvdrmArv3KVIPI9K4z49eF7j4o/BzxbpekK7eJI7ZNQ0soSCt9bsJocY9XQD8am+HXjP
VjuFXBxjmv5gx9Bxqs+uh/eHUUn3lJGaeI3zkdMetc5s2tiPoPagx5IJzmnlTgnjAxn25oAWRWyQAv
lr+GD9PCiioWfPQkUm7ATUU1PuinUwCiiigAooooA//S/tpooor+GD9PCiiigAooooAKKKKACiiigI
q6EOcMCDS+WxOBn6mr+rTfQjnKAqxkdjmrS27YySDTfszDngE1pDA1bbB7RdSmUJJORUeK1Vt22AkI
I7GhJAWPygioFbHoRUsR+bb0HWrpcsnZmdSSRo28OSSSDn9Kta1rujeFvD+seIdeulstGsbSW8uZcH
v13w/wAl9tiVFK54ePrqMGhPgC17N4U8U/GnxOvl+KPHF+2+KPHF8AeCyH9aMGhPgC17N4U8U/Gnx
ur2Xw2i1W+T7j4o/BzxbpekK7eJI7ZNQ0soSCt9bsJocY9XQD8am+HXjPWf2Ph/wAeCPC2t+IvD1
ySHig1Xw2i1W+T7j4o/BzxbpekK7eJI7ZNQ0soSCt9bsJocY9XQD8am+HXjPWf2Ph/wAeCPC2t+IvD1
ysSHJG7HPXrXrnje+/wCEp/BSX9kTwDYePfCOr2HiXw+XiBKIvbR0DLNu/iyDk
L/CDwxdW3hyzMFnpOiXmpzzsHkuLydUkcyzOSWeU+SCWYkk+1cv+yd4Sm0P4YeA7HxBqVnpHiXw
Wf2Q9Xnm+FniHvbSpSpPc+GvFF9JAAfcyWsqpeWoIOCAIrraMcfI3PTHa/Efwc2p+D1v4YeA7HxBqVnpHiXw
YvmqUpa90f/1/6n/wBrCBP+FcWGBq9Ftu8Yx1Yxywn3C4r3P9goy3Hhj4w
wXMXiz486S0snlGTSL1F44Z7eSIn64hA/Ov3SvjVi+D1TktaUkn+h8pFypZs6sNpLX1P/9D+sf4+fDbxH
3iXw+XiBKIvbR0DLNu/iyDk
7IX2jwX9tc2lxGhikRlI+or8ifj74Rk+GvxS8ZeBpLhrqTwzrl7pLTsMGQwTSFCT+hFfvo+Mf+Cnf
QV+TvxFvk0L9v/SZN7jS9Y0FNLvUHSItDcSxnPTduhCgYwM59q/ULwdrP9t+GNG1E2+0SSZCfmHH
oS0bv+Cucb/wVn/ZT8Z/tJ/sz6Z4Y+HVnHqfiXRfEFrrtnpMsqxfbo0hkikRlI+or8ifj74Rk+GvxS8ZeBpLhrqTwzrl7pLTsMGQwTSFCT+hFfvo+Mf+Cnf
oJBlHV1yUIPZircj27ZFfz3fCSe7/wCCOv8wCt8AwVL1f4CazePpPP7D3x5kl1nwpdXIdbPwx4jLkvBvJK
icxwI4JwARDcEkg4EiEda/a74/w2ul/D3D5xkl1nwpdXIdbPwx4jLkvBvJKgHd3rsLK9gHxGtuPrXyp9goy3Hhj4w
owhI91ZUx34FctSHsYwpdtX6kZjWaaodEkZfg7UGlsGjPzIjZUentXbI2ecdMV89fA7xZH4m8J+Hr8
XkKgHd3rsLK9aPQdSto3EREiFgerg5GBX2GOpczjY8nC1HG6Z0Xg69Zbv7JNlo5hxzynHb9a8A/Z+s
svFHhaTT5M8CS5sLH9qX4qPbSpSpPc+GvFF9JAAfcyWsqpeWoIOCAIrraMcfI3PTHa/Efwc2p+D1v4YeA7HxBqVnpHiXw
HP5jqWUpa90f/1/6n/wBrCBP+FcWGBq9Ftu8Yx1Yxywn3C4r3P9goy3Hhj4w
MvCfxY+vSpSpPc+GvFF9JAAfcyWsqpeWoIOCAIrraMcfI3PTHa/Efwc2p+D1v4YeA7HxBqVnpHiXw
HP5jqWUpa90f/1/6n/wBrCBP+FcWGBq9Ftu8Yx1Yxywn3C4r3P9goy3Hhj4w
jcaDYtJA8oOMfaW2xsM4+4Svua+hwGA5ldHpZHw9mebVfYZVhp15rdQi3b17H6dw22QCTx9KnNkoz0

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
+BUbmU+H/DVtvU4PyKbz7QAOeGxk9Pp9RSqYGlFc81c/b+HPoi8e5lZfVPZL+/JK3ra+vkf1lrbRDP
1XVPFetXDyLI8l9r17cl2HQt5kzZxk4z610SzPBxWkWz9BofQP4lemIxVOL6rV/of//W/uNL2jA+XN
yv4D+NXxr8FPsCRC08U3c0MY45FtcNJCThQMlDiv5oWZYKejTR/opmf0Gs8w8f3WMpyl0TuvxP67pk
fBb9qnQLjVPhpr4fUrUrHqelXP7u+01yMqssOQQvcSAlX96yr5bQrQtSep/PniF4P8R8LWebYd8n8C
sgmq5ZgyjJ71bjPyjJBNYYCjz1EkKtDqbdsFijllkcRxIN7M33MAHG72zjpXy38P9Vt9Sl+KfxruJT
5XzJ8Y9etfh34U8KfC3wjbyahc2dra6NYxFhumnby4IsgfxO5B465LdRX9ZeEfD6mvrbWi1+7f8AI+
PuhVAz7fdFeKeN/CFl4M8F/AT4QWscM0dx4ntrqaTduNxNCs15MzMeTmSMnn1rrvivr48P8AgnVron
tuL39kP/AIOJvg38UZbddN+G/wAbPDEvhe8uskRyXyIwTzBnBcSQQD2VuOpz+2/7DtuY5Pjt4tukuC
tnznhdxr2a8IrH1IW0VkdWBleo4Pqj9QP+CgH7b3h79hL9lfxx8UdUgbU/GN80ejeDtKjOZ9X1y5BW
x8ofB74iH/gtP/wVQ+GPxBspbjVf2QPgToFnrzWhY/YtQ8SXMSui7fuvIJCByM7IG/vV+2/xjeS9/a
zBGtYXAyCdreY6k7FyoO7lfcv2EtL03wt4l+PHg/SdKtdD0Z4tEu7G2gwkcaRwSQsI4xwijyosY4xt
+1fkPE+dS/1qoYZ/Cqb+9s/cclyP2nClWp150/kuhc/boupbjw98ToYmfzZrCHTFIXlDcvHbkgd8fa
FADc96+e8U8dDFVcNTi7qCa++3+R53BVGWFy+uk7OTj+Fz//0P7ZLbR1vNNa2ljDxPGVIPQjFfnx8X
IT6eo/Tr2r8bx+Vz5KWIo/HBn7LkmeU6VaeGrfBV0fqzyX4j3Vn4p+E+p+IfDk9tc6NqOmQ6jZyliy
tNJawXureG95J8zSHkZ2txn+GCSTAxzskixxX5j/ALSvinWvgf4h8B/G/wAINpt14rla58LXFnd8x3
28Vz+70qKTzLuOJhgq80yLBvXJVYmbPFfqbonh3TtB0jTtG0i0isNNtoooreKL5RCqDChQOgAJH0JH
vOMRKbsvhXQ5FgprZG/5GCBgZ+tfM37U1iH8GeD73IBtvE+kzZJHa7j9eO9fT0MkMiRyLIuGAYfSvH
3cu0zyQpI/HVioyamuTsgllCFyqk7V6v/sge/wCdfkWLp2m4+Z6lOtJKx//R/t0+Jfgiz8eeBPGXg6
+EPhj8OLjx94ltot8sENyWeM4xiYRqY4cjccSOG4yFavlH9mbQ/2to/jj+0NqF5oPwd8ExO+lg2Vy1
tntZJkOGh3DiQHsysFYe4r0YTftB6YwuLzRPhj4pJ6JbX9zZyv8ARXRo/wCVcv4j+L9hY2txpnxO8E
uYriAyFjBeQM8MsefUyIxHsw9a+OPi94ZTWPGOteG3gy954Fv7My8uRK74BA9dw3Y9q+v9DtLDw38V
8640bymcpvW0fXSSZ73DHCkf7VhVp6Rd/wAYtH0j+zN4jGs23h3V0uJGbVtCt7vO3aSNqnp/wLp2zX
Zve6Q+0Y+WMzxgf7vyKcdOBjpXz/AP8ABWH9nZ/21k+C37Ml5PqC+GZvEFvqepokzKkiB8KxwONnJ6
vkSsuAfJaCNApI+8kg6jJ/onupBDGtzkM0Z3jng85r+Zb9pjwIn/AASE/b7+FH7X/wAMbG40T9kD4l
IqpV9tH4WefnCVStKvT1i9j/0v6XfgRIfD2teL/CRuP9G0jxXrGmxDGD5P2x54QR2AiuIwPYCvuGIc
bGu/y428VZDNuU7j7jsaqody4bBq6mCD0JzXgVNzpjuyxExCoCcgcVwPx0c2V38F/HYbUBFpPiOwEz
j0cLNxqKSPoyZFZHV88dxXMznfliYyCflCHn159sYq/4c1u18SaBofiGzcCwv7O3vYJCAQY5YhIrY
KvUwqv7KPO15H3+Xe0xF3BXSP2lMKs27+E9q04ABgD2oW0O8yg/K2SQeq9APbs2RVuKH5h0r8rw2Xu
C+38VYe9f2cf8FGNbt9K/ZT8bWMkz239rXmmaPlTgkzXsI9+MBh07mv4qPjnejxLbfF9rmMFbhr5wD
5DGzRfIe8yB4+Mcqg9K/u9mgBUkgdOtf5G3h7xH4h8LeI/Dvi7wxf3Gi+J9KvrXVLC8hXJtLyCZZop
9pUV7kt/XQ8DOcOo1eZKyPqn7OoOQBX5f/8ABYLU5tG/Yk8XRxTiAXuv6DYuCAVkR9RgBU57cdRyMV
/spftb+Nv2QPidqJ0/4QeM9cafRZJ5Pl0PWpQuwLngRXC7AQMDeA3UsT/X28afI20ruUHaeq8dOpr/
n+0H4Kjg03W5X2rNrdpny7fUXVQB5kgH7zaAvmbiAqsqj9++kF4bwU5ZvhYK2vPZbbWZ7fEmBdOop0
gAooooAKKKKAP//V/tpooor+GD9pAOeGxk9Pp9RSqYGlFc81c/b+HPoi8e5lZfVPZL+/JK3ra+vkf1lrbRDP
1XVPFetXDyLI8l9r17cl2HQt5kzZxk4z610SzPBxWkWz9BofQP4lemIxVOL6rV/of//W/uNL2jA+XN
yv4D+NXxr8FPsCRC08U3c0MY45FtcNJCThQMlDiv5oWZYKejTR/opmf0Gs8w8f3WMpyl0TuvxP67pk
fBb9qnQLjVPhpr4fUrUrHqelXP7u+01yMqssOQQvcSAlX96yr5bQrQtSep/PniF4P8R8LWebYd8n8C
sgmq5ZgyjJ71bjPyjJBNYYCjz1EkKtDqbdsFijllkcRxIN7M33MAHG72zjpXy38P9Vt9Sl+KfxruJT
5XzJ8Y9etfh34U8KfC3wjbyahc2dra6NYxFhumnby4IsgfxO5B465LdRX9ZeEfD6mvrbWi1+7f8AI+
PuhVAz7fdFeKeN/CFl4M8F/AT4QWscM0dx4ntrqaTduNxNCs15MzMeTmSMnn1rrvivr48P8AgnVron
tuL39kP/AIOJvg38UZbddN+G/wAbPDEvhe8uskRyXyIwTzBnBcSQQD2VuOpz+2/7DtuY5Pjt4tukuC
tnznhdxr2a8IrH1IW0VkdWBleo4Pqj9QP+CgH7b3h79hL9lfxx8UdUgbU/GN80ejeDtKjOZ9X1y5BW
x8ofB74iH/gtP/wVQ+GPxBspbjVf2QPgToFnrzWhY/YtQ8SXMSui7fuvIJCByM7IG/vV+2/xjeS9/a
zBGtYXAyCdreY6k7FyoO7lfcv2EtL03wt4l+PHg/SdKtdD0Z4tEu7G2gwkcaRwSQsI4xwijyosY4xt
+1fkPE+dS/1qoYZ/Cqb+9s/cclyP2nClWp150/kuhc/boupbjw98ToYmfzZrCHTFIXlDcvHbkgd8fa
FADc96+e8U8dDFVcNTi7qCa++3+R53BVGWFy+uk7OTj+Fz//0P7ZLbR1vNNa2ljDxPGVIPQjFfnx8X
IT6eo/Tr2r8bx+Vz5KWIo/HBn7LkmeU6VaeGrfBV0fqzyX4j3Vn4p+E+p+IfDk9tc6NqOmQ6jZyliy
tNJawXureG95J8zSHkZ2txn+GCSTAxzskixxX5j/ALSvinWvgf4h8B/G/wAINpt14rla58LXFnd8x3
28Vz+70qKTzLuOJhgq80yLBvXJVYmbPFfqbonh3TtB0jTtG0i0isNNtoooreKL5RCqDChQOgAJH0JH
vOMRKbsvhXQ5FgprZG/5GCBgZ+tfM37U1iH8GeD73IBtvE+kzZJHa7j9eO9fT0MkMiRyLIuGAYfSvH
3cu0zyQpI/HVioyamuTsgllCFyqk7V6v/sge/wCdfkWLp2m4+Z6lOtJKx//R/t0+Jfgiz8eeBPGXg6
+EPhj8OLjx94ltot8sENyWeM4xiYRqY4cjccSOG4yFavlH9mbQ/2to/jj+0NqF5oPwd8ExO+lg2Vy1
tntZJkOGh3DiQHsysFYe4r0YTftB6YwuLzRPhj4pJ6JbX9zZyv8ARXRo/wCVcv4j+L9hY2txpnxO8E
uYriAyFjBeQM8MsefUyIxHsw9a+OPi94ZTWPGOteG3gy954Fv7My8uRK74BA9dw3Y9q+v9DtLDw38V
8640bymcpvW0fXSSZ73DHCkf7VhVp6Rd/wAYtH0j+zN4jGs23h3V0uJGbVtCt7vO3aSNqnp/wLp2zX
Zve6Q+0Y+WMzxgf7vyKcdOBjpXz/AP8ABWH9nZ/21k+C37Ml5PqC+GZvEFvqepokzKkiB8KxwONnJ6
vkSsuAfJaCNApI+8kg6jJ/onupBDGtzkM0Z3jng85r+Zb9pjwIn/AASE/b7+FH7X/wAMbG40T9kD4l
IqpV9tH4WefnCVStKvT1i9j/0v6XfgRIfD2teL/CRuP9G0jxXrGmxDGD5P2x54QR2AiuIwPYCvuGIc
bGu/y428VZDNuU7j7jsaqody4bBq6mCD0JzXgVNzpjuyxExCoCcgcVwPx0c2V38F/HYbUBFpPiOwEz
j0cLNxqKSPoyZFZHV88dxXMznfliYyCflCHn159sYq/4c1u18SaBofiGzcCwv7O3vYJCAQY5YhIrY
KvUwqv7KPO15H3+Xe0xF3BXSP2lMKs27+E9q04ABgD2oW0O8yg/K2SQeq9APbs2RVuKH5h0r8rw2Xu
C+38VYe9f2cf8FGNbt9K/ZT8bWMkz239rXmmaPlTgkzXsI9+MBh07mv4qPjnejxLbfF9rmMFbhr5wD
5DGzRfIe8yB4+Mcqg9K/u9mgBUkgdOtf5G3h7xH4h8LeI/Dvi7wxf3Gi+J9KvrXVLC8hXJtLyCZZop
9pUV7kt/XQ8DOcOo1eZKyPqn7OoOQBX5f/8ABYLU5tG/Yk8XRxTiAXuv6DYuCAVkR9RgBU57cdRyMV
/spftb+Nv2QPidqJ0/4QeM9cafRZJ5Pl0PWpQuwLngRXC7AQMDeA3UsT/X28afI20ruUHaeq8dOpr/
n+0H4Kjg03W5X2rNrdpny7fUXVQB5kgH7zaAvmbiAqsqj9++kF4bwU5ZvhYK2vPZbbWZ7fEmBdOop0
g8Tsb02ml/tQfhjnQK8xXWbQ/mn4vsmEwy/Ph9Rs8Ov/Ff4Ff8FAfgP8AtJ/tJ/tJ/tJ/tJ/tJ/tJ
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

YJ/bJ8TfD7xd+yZ8WPD1p4nguJIdU8FX8Q02/FvdcOy2bgeYvmCMl4mZeTk8mvpXxXbf2j8ZviLqQ
Tw/4L9eLPAmtr8IP26Z77xT4ce7Cjx/Fbk39m+FhjOoxj5pYwojDOoWReSQQQa/l3xM8J8bh8JVhh
aOFxC2M7XbAz9dqmvz2/Zi8eeHvHlv4+8SeEvEGieJfClz4jhutPv7G7SeC8iktYX3xspIwc9OvIzq
tUfAb4lfATxzYQXOga9pr2nmFMtaz8mOVT1V1ZVcMP8AaH8VfnB/wR/+PfjAeDfHv7EHxyvGT42/B6
+h3OF+0yqBjMDeX82MfL1zux+i4aq3H2a2PMjUSpun0PcfFM0ekftY+JhN5iyXmio6hlwp8q47HvzF
u6WZJoWRWVl7gqOfoex7jpX1GcUubDxkmr9up50aFT42nZ9Tv2IKoAejAn8jUltMBPOhPIQMP5VBGp
rU06QxalZt2Eg3VyVI8ysdsXZ3Pnr4TfEBPht+xZpHjjxTHHaXHg3wrexX8cm2NidKSaBg6DhW/0La
sms/E6bw6EhcktaaleR3s6Arwpa3uZgQcDGe/FfyhftNeMLbxhF8V7O3uHvJYb25kVjlEQhcKqO2FC
IGNvdgbJ4mGchlkWQYPI49a+hygDYz37V/Kt/wAGz37TUWpaF8d/2VNe1Njf2EyeNPD8Uzf8uMrCG6
7cXfiq3LYCfMkMMsxyTyAPLzkegPav5Nfiz8JfHvgH4RfCP43eMLOW28G/EeLUpvDkBgdZIrWHgSyt
f8F1vg5FefsYfDfV/CGn2lvo3gnxPp0aRW8QX7Np0kUlpsUDAWMZhUD2Ar+gPCLj6lluDweXxd51ZS
fCllT5lwTX7Uf8G6/wAa/G3wN+NHxx/YE+M+g+JvAviycSeItL0fWLd4Lm1vYQsd7AUcZ5jaOdSp2s
xq1LwbpUHxZ8PO/9neILaNYb4wSRSRyW8si8yQsJSTGxIGAVwa08XeLZUJ18nxsb0Ze9FvT7vX9Cs4
/FX9l34i+DPA2jS+IvFswtJLKzR9hmkW4jbqeMDBJzxgGv5p4GzanQzTD1pu0Yzi9eiueVRajUhN9C
5aoNa8YQaBDZ+Pvhhqd+xGrT2y/ZtSs7S6c7tzS27vHHJ0YoV/u19Tf8FJP2Arz4Y/sEfAHUdC0xb/
HjjdhnO5mIyDmv7E474peZ5HPH4N3hByjOPdXdv8z6jMsydem5U1p18z/9X+0rwN470z4jeC9A8cak
r+PeI2p1uaCsj9HoWexUJA6nFRh8kZwBSyfw01kAUEZr5WVSzsdBLuX1FLkHoc1VqaPvVRlcZJRRRV
ZyxOcDOK2LeXL/xYzWBFJsCrtBGa0bWcE88V6WXY5QkkzCcVK6Z//9D+rT/goF+yRpv7XPwK1Pw/YN
e9f34wuGCKcgZBOD1wc4+nXNfzCf8Fiv2Wl+FHxN0f8AaQ8I6dbW/wAPvGVymn6+sQ2Lpmt4byp2zw
WK4hjuI2BDI4yGGOhrH+0eXx1fvjtUqSHk8/TNfnmGk7XP8AYurVUrOJ84fESf4jfCm+tfEXguf+3F
3VpLFJHKrDhoyDkYrzn4Ifskj47fFbxT8APCllYad8Wr7TrnXPB99K4jttVe3XzZdLu+Ao8yIkxTYJ
/wCNdzFqFmI/MNzD5XALFwAp54PPt/I9Oayf2BfCH7JOm/FXUv2Xf22vhbrHhG7k1GTSdL8R22q3el
CVJ9SRxX19HhimtXI/k3jz6ZGEyWf1fEYKcqjjzRaa5Jp7Si77HyV/wR1+BPimx1b4h/tIeJdPutM8
AAAAAAAwMdqoTSF2XgjB6143EmIhpRi/hR/mNxnxZiM+zavm2IVpVW3bt2X3H//R/t3UnO0r8xqtqn
PXB9q/kvhnCe2xtKj/NKK/FH3WJmo03Jn5VfBbwvffHnxJ8DfBOoOREad4m1l/GvibzQcwaRDMb5w4
fWEelfq8AHjZRjH5V/S3itn1OvmiwUF+7pKMUuzsrnz2XUGoOr6nxVdSmL9pb9orVrm5dLSw8K+GLU
SQ30Y/EMD69e1fC3xttgPgh8Q9ORZPLOjSQmNcE4ypxkkAfexzX7t4cYb/hPhNbWX4JHzGLqv2r8z7
yv8AtBfFvU7j4geKbmcYmc3e5raFgRuVVjYMBgY3Yrhf+ClukXv7WPxf/YW/4J+6a6z+CdW1c+N/iN
O/4LK/CbWLn4IfD79rn4fW0jfFb4J+K7LxzaGKItNd6ZGxS9tlIIOGjdZCOn7o+tatx400/4gfET4F
/jiv59P2SYtc8ER/DH4GeKJ57zWvAHiTxV4NjkYfM9jLClzZEnJP+rhI64woqKUr0ZW3SPWyWPtZxX
qAAWKjmvj7w1p5m/aH+KEvmI0i6FoKJuGBy16SM5455x+FfK3/BT7/gp78L+vhvXwTpev2z6hpHgOK/a+jsrxIgEBhiBa9u3AjYuFMasnDEfPX86X7av/B13+0F4nGueBf2MP
IJBhtYiWVY9oOFXCkAdcg12Xws/ZyvNTRNd8fFtH0aQfJZjaJJIwwx5kgGUJBJPBJ56dR/W3AXg+6l
XBB6V+8f/BOz/gvD+3D4POh+A9R8U2nxHW3kit4dO8RRSol5GCxYLKgYLKSygAlQRwGr8c0Pwq8Eo1
RtpCvlWydy5yBX2/iT4N4XBYD2zrKFVbLTX/AIJ8vwR4sU8XjvqtSg3TfXa36n9w/wCyL+2JpP7RHh
K/ID9gH/AIQH4x/CXwR8XBR8XvgNda5vU8XjvqtSg3TfXa36n9w/wCyL+2JpP7RHhHQNnGr8c0Pwq8Eo1
TIk8AGSIAwLBfm8psEbl6c9omif8LQ8LrZa7JHa+LNLk2jULD79tOFxHPBIDwkiDdyCpyUYcjH6UNp
yviiVKkqclsf/9P+sPR/EOo6Nc/8I948tlsNUVvLt9RCkW+pgdCpP+rl9UPA52k5BHpqqkkA+UhiMr
qvyqvldVR5+bU/XqHE9CpaMtLnIfBCS58NN8QPDVpBvS31qa9soBIUVoLn98Cf8AZ85p0+ik+lZPwo
zDcw7GDg7kYNkjAJce+6vrqwdooI4mId8ZZum4+v417WV07tKa0PmeIKnsat8O7X3scRN4F0HQtNma
GAMs3Sv9cH42a5J4b+EfxQ8SQvsmsPD2oXin0Mdu7D+Rr/I7/ZK/ZQh/ahn+NfxO8V6fquoaPBNeSW
ype1J91kIz+ldvpPxV09JYRPe31g4PLEkBR65Gf5V9X/ALT/AOvwgvgvxAD42+Cx5BHpqqkkA+UhiMr
jq7H4Xl2KyvH0ViIL3Xtv03P6vP2I/8AggPH+2H+wN4c/8A42+NKt458aaPJqXhjSIrGOWwtYfMuYfMu
Qr+zl/wWF8ZfsEitX+NP7OvxA8Q6Ne/8I948tlsNUVvLt9RCkW+pgdCpP+rl9UPA52k5BHpqqkkA+UhiMr
9fG4Pe6pLkvo7N7W9O5+ueHeEyyrXj7CdopPms3pa1j/1PyG/wCCCkf7Yf7A3hz/wDjb40q3jnxpo8mpeGNIisY5bC1h8y5h8y5h8y5

H9Tt1e4tpSq3GlXIGJbW4QMdssbgqQCR0ILAhj7jKhZSWzjrg9a/it/Y5/aF+Jv/AASQ/a38ZfCn9c
6hldT6EEEe3NeJ4qcCPLcUq1B89Corxl011t8kKvTdKfKtnsyq0Tl8Ag01s428cVcYbX3Zyx4qoyY3
P08KKKKACiiigAooooAKQgEYNLRQB//R/tlKADOTU0YCZx0xTafEMAiv4Xstz9NlG5o28pGc5rgvjV
nEYVFGWxzU6tSjONam7Si0010a2P4Nviv8LPFvwM+J/jn4QeOLa4g8TaBfNbSO/W9tjgwXSjGdssZD
7lpxbvOlxZgv5TrcL5DxyqPnRlfBQg5BU8gqQT0Fc+d5TGlU9rT+F/gf7h/Rg8esNxfklOGImvrlFc
3I5TIJAr8/fBPhTx18T/ABdoXw8+GHhTV/HXjvULlYbLT7CMs8mG2vJIeRFCvUzNhVGT1AFf2RfsEf
lKSniqytZNNRj15l0v08z8o/+C3P7BVlr2lXn7Zfwo8Ol/EGm24TxzY2UBLXtgpCrqKoODLbqzK4wQ
aDqtpDe6RfwNa3ULKGWeJgQ0bA8FSM5B4IyCDX8Un7Qfws8c/wDBOL9tJJfhvHeaVb6NqyeLPBxjdp
WfM2R8uWrzv4L/Frwr8cPhV4A+LHgy9S88Pa9p0Wow8kvFuUbo3zgqysHQqRnKmvRnBAO4V/JGcU5e
PyyYUgAGvza/4Kl+G9R8cfs9+Afh5pcvkaj4i+J3hXQIn/iUXN2YX28H5tjvz2654r2eAaip5zh5y2
6RabRZWdvFbwYGAkaqFVQOwAAxXRQMeADjmvQr59LFY+ri6jupScl6X/AMjF0lTpKKPgj4+2FlD49+
EHxvttPkXzbjxJ8NtUihB/57abfQToPrtv5m/wCAmvzg+JTyXNpFoSqlxNf6ZraJblVd5Suk3f8AD1
c8FiSAAAB+srSEzr2BOPwr5a/ZS8LaV4H+G3hrwzpbyXMEOmWMsk7Hc07zQiZnLehaU4xwBgAAV9Ry
Lu3ic46nbIqk+gX0Ir9VtRuRF4j0yLLASpIo477GP9K+Bv2q44LLwi2vTNLHHo/ibStUUqegW6VWzr
+3r+1V4z/aw/aI+IPjTxRc6hrsst+bLSYLl3d4iDtaXZu2q77FAjAKxpsjXPlhj/AG3/APBUnxVY+P
h1FreaJZIb7U3YtCJBjDiPDSsOp2AEckV+f8FYnDYHEYrFVbOa1S6n7NxrkdfE4PDJO1J3uyvZ/8Ey
SONBjaijH4kgc1/pN/8ABUf9lyfx/wD8E2P2m7N7BGGlN5rmnaE+tQb0Ljdb/vZi2CAf3QmGSccjPU1/l
O/j16y1PXdL0eyvo7nVby6jtUy6qiO7BVLOxCqAWGSTxX+g//wAEwv8AgtR/wAEzfgP+wd8Ov2Yf
ihqPhbxZ4Z8Ow6Xq+kXXgnxBcRXMsbMWkjuLaxkhkBJ5yrAgnIr5Ar+Ff/AIKJ/8AJ1Hwt/wBiP
ok/9Ja+/v2Ov+CVP/BMb4Z/sHfDr9sb9vb4ieJ/Dvhzxb4Z0/WtL0vWfGGnWsMMt1bxyxxRxRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR

yv4zh7pHHTAycd//AK1fAv7TAOs6DD4C0qT/AEjXLy20ONgCNsFxIFnduDgLbG4bnpjmvu5cSYurC
XuGpJJa6OLXKFkeJPk+6cOoyPb09sV5ro1q/iTx/r/j5IXg0K5kjXRosllh0i2Bjt0UkDhyTLnocg
/wBnH5V+fs/iYRfHjQ7J2kX7TetAAG5IWBs4+uD+dfojrF39m8EWCOsSsukwlgx5G7YOf+/Z/I1+Hl
0zS9V1Caa4bbD83lRKvuX3KoABJY4AJ4Phnwu+JGvfGP4rab4r1i0l0jw/ZWElh4Y05yRKkDYL3dyp
yaOaNnlWJVG0ttBznr1weTj69K1y3gKjDETz7EL33Gy7f8A+7q8YtZXDLaTslq/U/9f+y/QoWh06y
YBGKtBFw3yjpVSMYmcYxyavqM5HtXH5m/Y/Ir/grVehNC+EXhWQwLF4qUaNKvnqjtbR6jbXkwwSCyC
GWw022yB2qO/WvM/+CD2qBvBX7Yvhx5sXFn8Vr6fyieVSaCNgSPQ7K4uLMNy8I1Ksd5TX3XZ+s5NSU
x7/5mtrPmZ//0P3U/wCC8Xw8m1/9lj4Y/Fa0g3T+C/HFhc3EoHMFpeK9m5z1x5kloceoHpX5U/8ABP
9kI85a8tUF1bjHcebbx/lX8X3hT4px/DP9tb/gmh+0Gt0bbTE1nS7W8uFTbttpphZSqWPYxagw6969
a/iTNqLimvNniUDM/jT8K0IQCU+tUvLcFTgEe1X4VPB6jPpXj4aD2OirNGvAcqcAHI59x6Gv40f+Cn
Pbap4g8e/Ge5htktmVt1vDtsy5dsDAVJeuRgcd6/rz6N8XSnia3/AC6jTlzeTfc9LIZq9R9Ej+vn4V
Ogr+W+JKsKlScodzx6Cs9T//0f7X6mj71DU0fev4ThufqDJKKKK1EFFFFABRRRQAUUUUAf/S/tpooo
qB2qfA6YFQJ94VYr+FKZ+oMntm+ZAOmK2IOSvTrWDbvhkOOOlb0JzjFfTZXWlGaRyYnRn8hf/BTj4o
Y3dg7WjuT1ywt0fnr5ma/Nz/AILQ+AYZvG3wR8d2+GnvtG1nRHJQfLJC9tdRkt1PW5UKeOTjqa7X/g
g3/d7P7kfuHPFIxDJgHHc/4V454u0+O98VaXBFBG89zo+oQQM4wI3JgKE49ypyfTmvZ55QAxGelcZr
bs2BknGa8/CStJNFyhY//9X+3vVtFsPFOga34a1VWfTdQtJrK4AH/LN0ZT+PzmvxB8J6jq+m+K/2U2
iJNe0iSOVwoUbg8tnDEM85a2I6ucfG+DWdSm62XVHpUhK3q0elxBh7TjVS2PuXwTZjRPj/8AtJ+Hxo
t9G8W+EtU0lZQ/M06NHeWwI6kFGnPPAA4rf8W6zDpN54B1m7n+zWw8SacjOTziZzDg+372v1HKK/tN
NRu4ZAVIdlIPXrXzVYvHpXirxv4cl8uOOdvtsC/76bWA/FWP4ivRlFtaHtZLXcJNI/m5/bC+Eut33T
PpfiKwDNnTdSiYxyIAwDBNysBuHGMdjXxf8At3Ws/hn41fCr4lJFYolzpFxpEsrIWdnsryK+iUZyB+
6y8MovH5ZUoL4on9EejfEjXPD7obS5Z7Yc+U4yPy7Guzk+P2majpdxpuv+HWuYZ42inRcNGykEEFWP
jVyt6adWf//W+qv2nP2Cvjr+zZ+1Fb/tS/8ABIbUZvhBq/iWSX/hMfCupXay+H9RlYrt22RUJHHjzP
nTT0x8yBhkdR0rqtOsQ0YVAeK/nvMc9xWY1liMZFOaVr9T+wqXD2WYLD2w1Kzve/XoSeBPCnh7wboO
Hk7NnxvEWYxjaMdCaysFYLuBHI4xWsumL1yc/St21slGAw5681sLZjHOfwr6anhtD8xrZm4vc4/+yk
xpSzppn/1/7TL3SWCOSuR9K4LUNLcAtsO7txX0ndaVG4bKEVxeoaGV3eWM+xr8exWSaXR+05TxElLU
fjX5eftG+IdT8TeP9Y8D6E7xtYaVLNfXkXP2SGQFZFj6EPJGphByCBK596/Tr4h6mfDViI7K1n1bWT
vCbd22rLtdn6Nw24YunUf2LO/wAkd34QSEeE9GvSRltNtkjwuwKpiU4C5OAOg5rV0iJrzXtNgYeZH5
eU/hDjXE3xslHa5+pvxR8Rva6DqkILokFlHFGQ2MgLjn3BY1+CPwc1oeNv279W1Pl7fTjdwQSKTtjW
SnGQ0ssUr8dSB5jj6YrThXA8mEqVO6aPkeZupzs//9D9BviPq0vjf9qj42627STW+ny2Hh+NicqPs9
X0t8HtQfR/jv4AuopTCPt6IefvZ7fjjFf1vjcP8A8JsqXTlR8LCv+8sfvFqFpdve3kq200iNO4BUZ5
vUfhPXbtVjv59ilEJLAfXmnR7iUQDLNgAAfpVVLxNThsr1JBJ5kK84xnjFdBor29tFezTLvmRlljJ6
GRlMVn4o0LRvGujor7llQSSw3GPUqWhBP+0PUV+k3/BQv4c6J8ZPhz4Jf8Tr4h6mfDViI7K1n1bWT
ZHHkgKa+t/sf+0eGauAh8erXm1qfqPDOI9tgquEvre6XXQ/s2mTzA3ALcj8q5eaNstuBHXtWT8LPir
D1nzuD3R/9H+2V4T5y4znHpWjbqwYZzmtB7YhgT1FSRRb3wM4r+PsPlM1rHU/Qp1L7jLjTYtQ07UbC
RcMD7Gv4k/8Agr78MJvD/wAWP2DKWX2eK7utL8cWSooCHzIoS7qPXdFL+Z9K/sr6OVRe3r4GeinF
CDvxQb4of8E2Pg1FdTQyal4duL/wzMoJJjFvcOIwfcxlD+VfsFLArYIOMmvwHjnht0cdVo20U5dPPT
r+Uv/got/wAF95vHUeqfDD9lC81jwz4OmWSG81eOXy9Q1QcqVilVibeHjnyysrZI3Jjn0uDvCjHZnU
sFU/Kn/BFD9mXUPi94/1D9un4l6TcR6FoVq/hz4cW1xG8cchIMd3qKqy4bO9Y43Hzf60knPH45/8E
JstA8KaRaQ2WmWluoWO2hiULGoUccKMZ71+rceZxguF8m/sTL5p16mlSS6npzr08PTeHo7vdn//S/t
/tpooor+GD9PCiiigAooooKKKKACiiigD//1P7waKKKAAKKKKACiiigD//0f+8GiigAooooKKKKACiiigD
WwI6uiP4+rF39m8EWCOsSsukwlgx5G7YOf+/Z/I1+Hl0zS9V1Caa4bbD83lRKvuX3KoABJY4AJ4Phnwu
0zS9V1Caa4bbD83lRKvuX3KoABJY4AJ4Phnwu+JGvfGP4rab4r1i0l0jw/ZWElh4Y05yRKkDYL3dyp
yaOaNnlWJVG0ttBznr1weTj69K1y3gKjDETz7EL33Gy7f8A+7q8YtZXDLaTslq/U/9f+y/QoWh06y
YBGKtBFw3yjpVSMYmcYxyavqM5HtXH5m/Y/Ir/grVehNC+EXhWQwLF4qUaNKvnqjtbR6jbXkwwSCyC
GWw022yB2qO/WvM/+CD2qBvBX7Yvhx5sXFn8Vr6fyieVSaCNgSPQ7K4uLMNy8I1Ksd5TX3XZ+s5NSU
x7/5mtrPmZ//0P3U/wCC8Xw8m1/9lj4Y/Fa0g3T+C/HFhc3EoHMFpeK9m5z1x5kloceoHpX5U/8ABP
9kI85a8tUF1bjHcebbx/lX8X3hT4px/DP9tb/gmh+0Gt0bbTE1nS7W8uFTbttpphZSqWPYxagw6969
a/iTNqLimvNniUDM/jT8K0IQCU+tUvLcFTgEe1X4VPB6jPpXj4aD2OirNGvAcqcAHI59x6Gv40f+Cn
Pbap4g8e/Ge5htktmVt1vDtsy5dsDAVJeuRgcd6/rz6N8XSnia3/AC6jTlzeTfc9LIZq9R9Ej+vn4V
Ogr+W+JKsKlScodzx6Cs9T//0f7X6mj71DU0fev4ThufqDJKKKK1EFFFFABRRRQAUUUUAf/S/tAooo
AKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooo
oA

6nsK6O2QrGVxjJz9KwvE9qwsftUcfmTQusqgeoORn8hXsYxP2Dg9rHnQj73MfulBqQvbTTrtQBvtoh
H9a/BMTh1SqtLa579CV0Y3wz1x9X8J6U8kizXEBktp8tyGQ7elemBy4RhxXzb8K74aR4r8U+GJ2O2V
+0N+yE3xnutK+Inw81y28F/GvS7fyrW9ljY2WtWe4O1lqESY3xcMUk6xFjj5CyN+snizQG8U+FvEPh
pv8AYP8Ajx488K/DrU/EPwf8WzSnVbvwTraPdeHtdiZ13tbhtsbsW8vbcW5VyjYLHaVH6zfC7/gutp
cTNDJa6gikDZMpDIJVZoWweRJntX4hfED/gin8VPH/gax+I/7Hn7RbvYSIynwp40cyf2ddK5We2W9j
jl0iROM4/wBJVFJ56KT0r2nSv22P2O9VhhvNN/al/Z/urV8hWPi+wTdgkHG6UdwRX8UvxO/YE/4K8f
75Ga+Ij4OcLyd6GM5V2PWp5ZhpawqaH95nir/gpT+w94Kv5NLv8A9ozwBrmqqdptdCmk1eRW7BjaLl
BpGgfslftEeHbaRNqIPDI0aJQe4e4aMqfyr6M8J/8E8P28f2SNAg+PH7TXglfCXgOTUtP0GWG48QwX
/CG3+Edn8RdS1LxD/bmlz3erC3gsHa1jgmTYgZy26BG4x1PpXJfGT/g4d/4KOfEafUtP8IeMPht8G9
xxHCMxVyctknY30r+sD4Gf8ABup/wT4+Faabe+NtD8cfG7XIJFkJ8QakYrV8DB/0W3CKVbk4Ymr42x
tFEUA4Nf0h/sI/8ABuDq9/qGjePv269Zhs9BQx3H/CD6JeHdeY5MV9eJtYJ6pEdx/vjOa/rN+HXwo
VGHeyv8Agee87c4OFJWTP//T/tb8HeBPBvwz8K6P4C8A+GtG8H+CtNt0ttO0/T7WO3gs4x0CRxqFHk
p6Damj71DU0feiG5sySiiitRBRRRQB//1P7aaKKK/hg/TwoooAKKKKACiiigAooooA//9X+2miiv
qNKUtETKSSuxsKckHFblsnyjGKoRxbpFwTXQW9qcnsQPyr6vh/Bu75keXiMSmfin/wAFrtSitvhP+z
BwX8XLbw3dfs7eA4r+GLUXtNc1TygwGAy21ujE/Saf8AKvmb/glr+3R8D/2NP2P/ABbf2mieIvip8a
V/cf/9f+6iQqEV8qDkAFhkZ7Z9s4r4r+P/7e37MX7NVrdr8QPiAl/wCJ1DeT4f0WD7fql3IOiCGIkF
80sjl5JXPJZmPJOSeWwfrX8yY7jLD4d3o+80f6e+E/7PLN8w5cZxLiFh6b+xHWfzvovxP1z+Pn/BYf
21zNGrFdyoW4QZ5CqFHBzgk55O48QWOGQPkHj2Nevfs//B/Xf2jvjT8OPgroEs1tJruorDfXCDLWer
/aHvZvj/APGjQIJ/g9pt4V8O6XOv7rxHeI/z3UyED/RopYyi4yJHVzyqjd/UdBaW9tFDHawR2qAABU
MPitmPGeeVs3zOXMpP3Y9IR6JIW40y31H7P5ibpYpllhIO0owPDKe1fih+yJ43XQPiZ4g8KaveRyak
VaPEisUuG+1Tu8G7r5c1u80ROMjep7V/TPgbhvbxxDeui/M/Ac8avofpB+xL4tfwJ8XPih8DtYeU2M
LiJhbzxkdCxCwqRjP701+zV14g0jxn4d8M+L9ImS5tdRt0k4OSrMoYqfQjI/Kv0PiLL50sRSxCXu1F
4K27KO48r7Pknsp5r91PjT4Z+0WFvr1uoNxbDy5GH8Ufb8q/Ffx5pi+B/wBtTwvryiSHSPGfhm40+T
o+8rsDnOK/cD4I+J/DPxZ+G9XgcFjZ3I+Wa2dhwGRwy89ce9fjVqpltbmSOUknC
TUIs/fOz09ACdoX8K6SCCNeAPm615j8MPir4D+MfhwRWi+Tw3cxeYk8eVeJuP3ckZ5jYZ5F
6Gocn1NGT6mkUf/R/uWi+Gvw/tPGuo/Em38E+Fk8e3UEdvPrH2KP7ZLEmdkZmxuCjJ4BxUX/AAl2uS
0Uc4yfeuhzVilSXYc85HXOKoTT/K7Y6daimmREZ9ykD3r5A/am/bQ+AX7JnhVvE/xf8AHWm6Pqmo
/BUrwL4K8aAHWm6G8gP2F
/BUrwL4K8aab8B/gNZ2vx0/aA1S8Gl2unW12sOm6bOeSb69GVj2rhiqgnAPTpXyN8V/2hf2uv2/NTu
cmKLMtxqV5IlrDPczHLyzeUkg3MeobpnFfFZZm1HNMxWX0Hdvf06n9C4Dw5q5NllTM8erSppNLz6H
/wDEsNvy6Acf628VD0x5Zx0FavxJ+J8vhbTm1eHWPiZ4g8KaveRyak503Uvs+oS
cnoB2rvtP8XWnw78NvEty6x6Ety6xOlmbcMTjjazsfoOCTxwMdxXntnE8kqgD347Af8A668D/aJSXOov4
g4Uk9BHEkSgducV0H7H3h1LnwreeO763aPU/EF3cazct14nmaSJAfQRtH+Zrmfim11B4G1/T9LhOLr
EUqMgZSpH6VuNbqDg5GecU0womc5wQVGB3IwP1xVVPeVjz07H11+zd4jF54B0iBiVu7KR7WQHsVbA/
ukerhaml2eP+LLiTwj8RLbxPErgQeW8qw6MD0IrwH4saZ
bOSS2f7RCczJ8yg9CRj/AAFeOeMNGTQtenNnn+y7gi5gwMABskg/iTxXsCS5RyV7A1mavpcPiPSk0y
Daa7AoO3R9VcBIr/ACR8sM5UQyAcLLsc8O5rooI7mwlktbqCS2mQlTnoR04PuKu3Vlpes2F3ouu2EC
vhZ4k1bwlqJ+D/xD1OE7lOCNrnw5qzkBfEOklm8rceCbuFNqTjjJ2SDhyF+iJPljLFgx7+59a/mTON
9zX2OEYNnJIPpVTxHpFr4h8O67oNzLPHa6lZzWcrwNtkWORCpVTxHpFr4h8O67oNzLPHa6lZzWcrwNt
U8H2+qWSzLrPgHxqnBg+nKwN+D3xEuvBPiWDxBpnxM8QeFNXvI5NKuZ9Ju59JvJ9JvJ9JvJ9JvJ9
vX4XfES68E+JYPEGmfEzxB4U1e8jk0q5n0m7n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n
XDTU+zyvM41ZOOh6RHfokjeajtk9vbn+YFfvv/wQX+H+ua/8Ufi/8Q7rwT4lg8QaZ8TPEHhTV7yO
K6/0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n0m8n
K+k4LpR+vxcnsn+R/KX08atWHWjJS1TlDm/wtq9/Jn9P0pKRuAQDmqSOMkEjrVycsBIu5Q2e1UkQ
QVZrZJGxnuA5/Gv17/bL1268Jfsd/tJ+KtPuPs+qaD8Ltc1Gxm2hvLuLUbSxuJ4JNrAxyeZHIy5B
R0dkiAim0q/2291aqO3kXbWcuM8LIpx8lfqZ+yv8Qv8AQ5SIBxy0bFHH
jtnurd0IZbiNjFeWrj+8Ss6Edsj0r+o88ypV8MlbWOp4NDY/brUNPg1S0u7G4UGCVdrAj1GRivx3/k
H2NeL/HnwFp3xO+G9XgcFjZ3I+Wa2dhwGRwy89ce9fjVqpltbmSOUknC
xloa6VexPa7zDICw9q/WaeFVRep3YbEWauJ4D8VeMfhwRWi+Tw3cxeYk8eVeJuP3ckZ5jYZ5F
rUs9deGM2+SynnkZH4jjP6V8LnXhzBPxF8OeBPBvwz8K6P4C8A+GtG8H+CtNt0ttO0/T7WO3gs4x0CRxqFHk
i87ieeMVa8D+Vg2qyvKlvovi7nsVYk9dPvmuGtBxH72yuJ4FXbzuMsqgAdyK/Cv2ZH4jjP6V8LnXhzBPxF8OeBPBvwz8K6P4C8A+GtG8H+CtNt0ttO0/T7WO3gs4x0CRxqFHk
lTBGxJ7GRI1J4yxiPPIA6V7von/BVD466Ddlfih+x9qN/piEF73wf4ji1B8Y+8Lecrpt5/h5Febi+F
IFzn5iccHmvrnwV+0x+z58Ddl+x/4Ddl+x/4Ddl+x/4

```
AIKg/Ev4jXs/w9/Yb8EWHiKGRntbr4jeIEdNE075trPZxrhr112naAQhOMkjg/D/AMOv2X9PsvFB+I
VQwHQcZr8Sz7irFY2dpuy6I/rzhXgfLcmpL6tBOT6tamUbCZ+GjUD0BGBX5u+DdT+2ftqfto6/tSG3
BxyXlYAL659a/XPo3ZcqnEKqSV2os/P/AB6zRw4fqNdbL8Wf/9P6V8SXT+NvHPh/wy8UNxp1lMl/qy
ldSGaY7R8+pSERRRKBwFDOigdMKfU190eBPDdroXh/Q/DNpG4NnaQ2vzE5bbGASeepYH6kj1r+wsJH
43s0kpI+8WUfwiuq+P3i4eJtbX4K6BeJKnk/adfKkq0Gm+ZsaNXX5kkm+dMgj5BKeoBGlo9gsVtbQx
8GXviLSrDxHeQyz2VlcXCJNdxowV2jU43hWIBx0r5s8GabDqPiKTxDOkpEssVrbZPAgjYkMB/tOz/q
KWKxtLC1p8kaklFy7X0Tflfc+lNauhpcAun2mAnDN/zy9z6L6/hQsqzwpImHVgCCOh9wa/MD4Uftfz
uWRJFyHhIO7eCQB3wK8DhPjXCZzS9phJarRx6prdM+o8SfB/OeE8ZLC5vSfeMl8Mo9GpbO/ke32F/N
uvuNNi08XBkkgysZPXbnp+H9a6eJcvc1e2p+XUq/Z3P/1P7DvGlgdY8NXixzCOWIxyxH12nP9TXzlP
xX6ph8QqSlQqI+RhUe57H4f1ew1/SrHWdOuEubG4TejL0I7L+BzWxhA5Yngdq+a/BV3c/DzUn0S+d4
earuQcY5rBtnQefeJPBb+M9CTRf7TudF16zlF7oGqRHEuk36/dbI+9E3CPH3Vm9eOu+FvxFvvFkGs+
e9mRvJ1BDybK9CHMkDEZx1VtrryvzfJcSZCsZSaWku56eDxnK7SZ76YkVmEUhaEAkODwV9c+lWvKJS
SvwXF5TLD1XQmndO3/Dd9j6qnUTjzXP/1Zv+CnPwqbw/8d/28/hnHbGGC5aLxfYR4ACrMiXQKjuN0s
dP8AHnw91DR5g+ApmsHIABAyWK38XGegHpVD/g2L+J4tPB37X37Nt/KEvdG8U2fi+yjdyWeG8t/scw
+NtfwHm9FxR5VCSexUJySaSkAKgA9aWvBOsKKKKACpo+9Q1NH3q4bgz/1v7aaKKK/hg/TwooooAKKP
aj2Y709I2cruzxV0qUpA2luN5kYADmrscW7GdpNW4bUsy7sYq6IFjCkbsAEscj5QOpxnJ4B+vSvocP
/a8/4Km+EfhfdaP8Lv2f7e0+JXxKjMkGoaspL6PoEg/gklBBubgdPJiyFbh2U/Kf56PiX8Stb8R6rc
nzS/w/5vQ++v2n/+Clnxr+PktxoHwyuNZ+CHwvkVlxaT7dW1CPlCJbmP5YAcfcjJYYwWr8nvGXxP8D
EE7Xeq391f3LHLNI5JJr4fNM3rYyXNzWj0R/sP4Y+CfD/B2D+r5ZRSn1m1eT9X+mx//R/Kf4oftb6/
AFbVsa3C0divJJnG7k1nF8MeMirbfeqhJ96uyFNM+ax83LcQn5SRwvpXqHwu+IDeD9TeO9eaTSLg4k
kdTkMPUV9WfsQftSt+yL+1D8OPjc6zSaBCJdH8SJHnfdaNO8RnUKPveW0cU49DEP71fl5qXxZ03TNQ
ZLi37tWLi/J9GvRn+n74Z8WeH/G3hrRvGHhbUrLXvDmp2kN9YXtvIrR3MEih0kDDjawYEH6+ldCsQI
nxp+FPx28E6L4++EHjvw9488I38Sy2l7YXKSrg9pMH5H4bKNhlIwQGyK/T8FiqWKScd2f8+/i/4N5v
14l0mVYwsN4zIAuAys27P1z3r668a+d4t/aW8E6ZBYzy6T4b0C9v5pXI8lbu5lFvEAepcxxytjHCse
4msbvWvg18fvGNvfXEn/AAq/xSketwq43Jp93vEVzImOg3tC8i8t++D/AN6vvvW7NoJpkcHHavnX4u
63M97oYaRVWz1ElmmtOT0kIlmjUZ+bzB6Cvu+5VVu/s84Elldp5TDIZS2CDyMggqfxr8SdCvPtel6N
AONmWjPzI2PzbifJnh66q0tYy1fkOK6I+DPjF4Nn+GXxz1nRbtrqPw74gs/t1i6ldjSw4EwUYHzCOE
Pw1oljpl9r2uabp+nahDHLYvMWDXiuoZfKRVZpMggjaCTkYBPFfW5DxBSdJKs0mvM2WHnbmSPmLWNJ
ArUlLAeSfF00MuO+N9mFJ69DiuPG+J2RUqrpVq6Ul2PosPw/jPYqs4PlZ//T+42lYLlCVqS1v542Lc
LvtGvYUzIBsHmPCFzkgEEjHfHGf6kw/EuWYr+BWTv5noPLq1J+9Gx9S6fq7OkSyYBPp3rs9OuBJmMk
4a+h2YarUTs2eu2qLc+U+0LcBBGr90XPRfQd8V+cHwh/Z2+Fnxf+OX7UeqfEHwJpXx/8Ag+9Pf+Ka
J7gpbAhuhzjtX8m+P8XQyuEoaT5vwP6T8B6i+vVJPbl26Hsmn+GtM0qK103TNLsbTTbeJLa1ghhSOC
fHloNlXP7MmjJ3JxnNdar20a58yBBj15/WsHUtdtIZBb2iXmq3jfcht4mcn/AHmAKIOOrso9666WBP
zm3to3mmXkhynlhc4IV+lfr3FoWpa7NHeeJxGltG/nR2UIEkMZ7eacjzXGMgAbBnPzdR+dXhfR9Ps
rIdZu4/uoiQIRGGx1AkeMjPGFx1Ga+nvH/jTTfhd4N1DxBeKbrU5tllY2jDM19dO4WOFO5kZnQex+Y
biBo5BAVKtdNGwyhlBYKDyI2wfvmv9FYyaSR/n3VTVRnlus+HPEPhWU/GeWePV/Ggl3eJFhkJ/tHTX
a5P4fGbwT4s1D4VX0yHTnWTUPDkm7AlswcyWyg/xQuy4HHySIRnDFWd9J3VmfRvhWzWDVdAtdkQiV0
2+Dvxk8Y+D9OihsvDc7/ANs6LHGgVIre4ZvMgVR0WOQMo/2dtcf4G+JOs/Dm4j0ry73Wvh3cu4vLCN
dzxiv8+fEDMMVwfxfUxWCfLCo1LlWzT3P96fAjKcq8UPCuhgs/ip1KadJTfxRcdpXWuiaP/V+yfBHx
X1yOoIH4h+BPGE/gXV92oX0j/Da/bbexEMzaJOxBN4igHdATgSxjlQ3mDoa+sdH8W+KPhjqwMJ1
/wDifY/b/wAJeNZvD2q2d5GcxMwVxkEMpPQ19j2txFqcOl69p5jOUG8j+JP4yfdfdTX45fCb4v6f
rGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb
rGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb
TW5oqjTuzUeUZGB+tLHcNDJHcJuWRTlWB5U+oI6VlQzhxjO761oIHGAwxWDguhTxF9j/1v7F/iF4Dv
dDdNaybCyOjdJoWAykiEgjIbaQQG7xG4IcI4OQQeQa434h+D4PHcmleJNP1e68F/FLS4mTS9ftBnzP
zQWd/C8Dknqg8xLXIHXaK/F7/giJ4tT4U/AbwL4Ha4Mv9n+HLeJ5Cu3eUjuDoWAykiEgjIbaQQG7xG4IcI
Hf2VPjDLK7W1x4h0Se4fA2vb3KfZH24PKhnkzuxjI464+14GwEq/CeLy6uv4V7easv8Agn67k81iMu
tRLQzMgxqaipxBHBGKbXwcY2PSuFFFFUB//X/tfqaPvUNTR96/hOG5+rdDqH+tsGSeepYH6kj1r+wsJH
1dSGaY7R8+pSERRRKBwFDOigdMKfU190eBPDdroXh/Q/DNpG4NnaQ2vzE5bbGASeepYH6kj1r+wsJH
43s0kpI+8WUfwiuq+P3i4eJtbX4K6BeJKnk/adfKkq0Gm+ZsaNXX5kkm+dMgj5BKeoBGlo9gsVtbQx
8GXviLSrDxHeQyz2VlcXCJNdxowV2jU43hWIBx0r5s8GabDqPiKTxDOkpEssVrbZPAgjYkMB/tOz/q
KWKxtLC1p8kaklFy7X0Tflfc+lNauhpcAun2mAnDN/zy9z6L6/hQsqzwpImHVgCCOh9wa/MD4UftfPz
uWRJFyHhIO7eCQB3wK8DhPjXCZzS9phJarRx6prdM+o8SfB/OeE8ZLC5vSfeMl8Mo9GpbO/ke32F/N
uvuNNi08XBkkgysZPXbnp+H9a6eJcvc1e2p+XUq/Z3P/1P7DvGlgdY8NXixzCOWIxyxH12nP9TXzlP
xX6ph8QqSlQqI+RhUe57H4f1ew1/SrHWdOuEubG4TejL0I7L+BzWxhA5Yngdq+a/BV3c/DzUn0S+d4
earuQcY5rBtnQefeJPBb+M9CTRf7TudF16zlF7oGqRHEuk36/dbI+9E3CPH3Vm9eOu+FvxFvvFkGs+
e9mRvJ1BDybK9CHMkDEZx1VtrryvzfJcSZCsZSaWku56eDxnK7SZ76YkVmEUhaEAkODwV9c+lWvKJS
SvwXF5TLD1XQmndO3/Dd9j6qnUTjzXP/1Zv+CnPwqbw/8d/28/hnHbGGC5aLxfYR4ACrMiXQKjuN0s
dP8AHnw91DR5g+ApmsHIABAyWK38XGegHpVD/g2L+J4tPB37X37Nt/KEvdG8U2fi+yjdyWeG8t/scw
+NtfwHm9FxR5VCSexUJySaSkAKgA9aWvBOsKKKKACpo+9Q1NH3q4bgz/1v7aaKKK/hg/TwooooAKKP
aj2Y709I2cruzxV0qUpA2luN5kYADmrscW7GdpNW4bUsy7sYq6IFjCkbsAEscj5QOpxnJ4B+vSvocP
/a8/4Km+EfhfdaP8Lv2f7e0+JXxKjMkGoaspL6PoEg/gklBBubgdPJiyFbh2U/Kf56PiX8Stb8R6rc
nzS/w/5vQ++v2n/+Clnxr+PktxoHwyuNZ+CHwvkVlxaT7dW1CPlCJbmP5YAcfcjJYYwWr8nvGXxP8D
EE7Xeq391f3LHLNI5JJr4fNM3rYyXNzWj0R/sP4Y+CfD/B2D+r5ZRSn1m1eT9X+mx//R/Kf4oftb6/
AFbVsa3C0divJJnG7k1nF8MeMirbfeqhJ96uyFNM+ax83LcQn5SRwvpXqHwu+IDeD9TeO9eaTSLg4k
kdTkMPUV9WfsQftSt+yL+1D8OPjc6zSaBCJdH8SJHnfdaNO8RnUKPveW0cU49DEP71fl5qXxZ03TNQ
ZLi37tWLi/J9GvRn+n74Z8WeH/G3hrRvGHhbUrLXvDmp2kN9YXtvIrR3MEih0kDDjawYEH6+ldCsQI
nxp+FPx28E6L4++EHjvw9488I38Sy2l7YXKSrg9pMH5H4bKNhlIwQGyK/T8FiqWKScd2f8+/i/4N5v
14l0mVYwsN4zIAuAys27P1z3r668a+d4t/aW8E6ZBYzy6T4b0C9v5pXI8lbu5lFvEAepcxxytjHCse
4msbvWvg18fvGNvfXEn/AAq/xSketwq43Jp93vEVzImOg3tC8i8t++D/AN6vvvW7NoJpkcHHavnX4u
63M97oYaRVWz1ElmmtOT0kIlmjUZ+bzB6Cvu+5VVu/s84Elldp5TDIZS2CDyMggqfxr8SdCvPtel6N
AONmWjPzI2PzbifJnh66q0tYy1fkOK6I+DPjF4Nn+GXxz1nRbtrqPw74gs/t1i6ldjSw4EwUYHzCOE
Pw1oljpl9r2uabp+nahDHLYvMWDXiuoZfKRVZpMggjaCTkYBPFfW5DxBSdJKs0mvM2WHnbmSPmLWNJ
ArUlLAeSfF00MuO+N9mFJ69DiuPG+J2RUqrpVq6Ul2PosPw/jPYqs4PlZ//T+42lYLlCVqS1v542Lc
LvtGvYUzIBsHmPCFzkgEEjHfHGf6kw/EuWYr+BWTv5noPLq1J+9Gx9S6fq7OkSyYBPp3rs9OuBJmMk
4a+h2YarUTs2eu2qLc+U+0LcBBGr90XPRfQd8V+cHwh/Z2+Fnxf+OX7UeqfEHwJpXx/8Ag+9Pf+Ka
J7gpbAhuhzjtX8m+P8XQyuEoaT5vwP6T8B6i+vVJPbl26Hsmn+GtM0qK103TNLsbTTbeJLa1ghhSOC
fHloNlXP7MmjJ3JxnNdar20a58yBBj15/WsHUtdtIZBb2iXmq3jfcht4mcn/AHmAKIOOrso9666WBP
zm3to3mmXkhynlhc4IV+lfr3FoWpa7NHeeJxGltG/nR2UIEkMZ7eacjzXGMgAbBnPzdR+dXhfR9Ps
rIdZu4/uoiQIRGGx1AkeMjPGFx1Ga+nvH/jTTfhd4N1DxBeKbrU5tllY2jDM19dO4WOFO5kZnQex+Y
biBo5BAVKtdNGwyhlBYKDyI2wfvmv9FYyaSR/n3VTVRnlus+HPEPhWU/GeWePV/Ggl3eJFhkJ/tHTX
a5P4fGbwT4s1D4VX0yHTnWTUPDkm7AlswcyWyg/xQuy4HHySIRnDFWd9J3VmfRvhWzWDVdAtdkQiV0
2+Dvxk8Y+D9OihsvDc7/ANs6LHGgVIre4ZvMgVR0WOQMo/2dtcf4G+JOs/Dm4j0ry73Wvh3cu4vLCN
dzxiv8+fEDMMVwfxfUxWCfLCo1LlWzT3P96fAjKcq8UPCuhgs/ip1KadJTfxRcdpXWuiaP/V+yfBHx
X1yOoIH4h+BPGE/gXV92oX0j/Da/bbexEMzaJOxBN4igHdATgSxjlQ3mDoa+sdH8W+KPhjqwMJ1
/wDifY/b/wAJeNZvD2q2d5GcxMwVxkEMpPQ19j2txFqcOl69p5jOUG8j+JP4yfdfdTX45fCb4v6f
rGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb
TW5oqjTuzUeUZGB+tLHcNDJHcJuWRTlWB5U+oI6VlQzhxjO761oIHGAwxWDguhTxF9j/1v7F/iF4Dv
dDdNaybCyOjdJoWAykiEgjIbaQQG7xG4IcI4OQQeQa434h+D4PHcmleJNP1e68F/FLS4mTS9ftBnzP
zQWd/C8Dknqg8xLXIHXaK/F7/giJ4tT4U/AbwL4Ha4Mv9n+HLeJ5Cu3eUjuDoWAykiEgjIbaQQG7xG4IcI
Hf2VPjDLK7W1x4h0Se4fA2vb3KfZH24PKhnkzuxjI464+14GwEq/CeLy6uv4V7easv8Agn67k81iMu
tRLQzMgxqaipxBHBGKbXwcY2PSuFFFFUB//X/tfqaPvUNTR96/hOG5+rdDqH+tsGSeepYH6kj1r+wsJ
1dSGaY7R8+pSERRRKBwFDOigdMKfU190eBPDdroXh/Q/DNpG4NnaQ2vzE5bbGASeepYH6kj1r+wsJH
43s0kpI+8WUfwiuq+P3i4eJtbX4K6BeJKnk/adfKkq0Gm+ZsaNXX5kkm+dMgj5BKeoBGlo9gsVtbQx
8GXviLSrDxHeQyz2VlcXCJNdxowV2jU43hWIBx0r5s8GabDqPiKTxDOkpEssVrbZPAgjYkMB/tOz/q
KWKxtLC1p8kaklFy7X0Tflfc+lNauhpcAun2mAnDN/zy9z6L6/hQsqzwpImHVgCCOh9wa/MD4UftfP
uWRJFyHhIO7eCQB3wK8DhPjXCZzS9phJarRx6prdM+o8SfB/OeE8ZLC5vSfeMl8Mo9GpbO/ke32F/N
uvuNNi08XBkkgysZPXbnp+H9a6eJcvc1e2p+XUq/Z3P/1P7DvGlgdY8NXixzCOWIxyxH12nP9TXzlP
xX6ph8QqSlQqI+RhUe57H4f1ew1/SrHWdOuEubG4TejL0I7L+BzWxhA5Yngdq+a/BV3c/DzUn0S+d4
earuQcY5rBtnQefeJPBb+M9CTRf7TudF16zlF7oGqRHEuk36/dbI+9E3CPH3Vm9eOu+FvxFvvFkGs+
e9mRvJ1BDybK9CHMkDEZx1VtrryvzfJcSZCsZSaWku56eDxnK7SZ76YkVmEUhaEAkODwV9c+lWvKJS
SvwXF5TLD1XQmndO3/Dd9j6qnUTjzXP/1Zv+CnPwqbw/8d/28/hnHbGGC5aLxfYR4ACrMiXQKjuN0s
dP8AHnw91DR5g+ApmsHIABAyWK38XGegHpVD/g2L+J4tPB37X37Nt/KEvdG8U2fi+yjdyWeG8t/scw
+NtfwHm9FxR5VCSexUJySaSkAKgA9aWvBOsKKKKACpo+9Q1NH3q4bgz/1v7aaKKK/hg/TwooooAKKP
aj2Y709I2cruzxV0qUpA2luN5kYADmrscW7GdpNW4bUsy7sYq6IFjCkbsAEscj5QOpxnJ4B+vSvocP
/a8/4Km+EfhfdaP8Lv2f7e0+JXxKjMkGoaspL6PoEg/gklBBubgdPJiyFbh2U/Kf56PiX8Stb8R6rc
nzS/w/5vQ++v2n/+Clnxr+PktxoHwyuNZ+CHwvkVlxaT7dW1CPlCJbmP5YAcfcjJYYwWr8nvGXxP8D
EE7Xeq391f3LHLNI5JJr4fNM3rYyXNzWj0R/sP4Y+CfD/B2D+r5ZRSn1m1eT9X+mx//R/Kf4oftb6/
AFbVsa3C0divJJnG7k1nF8MeMirbfeqhJ96uyFNM+ax83LcQn5SRwvpXqHwu+IDeD9TeO9eaTSLg4k
kdTkMPUV9WfsQftSt+yL+1D8OPjc6zSaBCJdH8SJHnfdaNO8RnUKPveW0cU49DEP71fl5qXxZ03TNQ
ZLi37tWLi/J9GvRn+n74Z8WeH/G3hrRvGHhbUrLXvDmp2kN9YXtvIrR3MEih0kDDjawYEH6+ldCsQI
nxp+FPx28E6L4++EHjvw9488I38Sy2l7YXKSrg9pMH5H4bKNhlIwQGyK/T8FiqWKScd2f8+/i/4N5v
14l0mVYwsN4zIAuAys27P1z3r668a+d4t/aW8E6ZBYzy6T4b0C9v5pXI8lbu5lFvEAepcxxytjHCse
4msbvWvg18fvGNvfXEn/AKq/xSketwq43Jp93vEVzImOg3tC8i8t++D/AN6vvvW7NoJpkcHHavnX4u
63M97oYaRVWz1ElmmtOT0kIlmjUZ+bzB6Cvu+5VVu/s84Elldp5TDIZS2CDyMggqfxr8SdCvPtel6N
AONmWjPzI2PzbifJnh66q0tYy1fkOK6I+DPjF4Nn+GXxz1nRbtrqPw74gs/t1i6ldjSw4EwUYHzCOE
Pw1oljpl9r2uabp+nahDHLYvMWDXiuoZfKRVZpMggjaCTkYBPFfW5DxBSdJKs0mvM2WHnbmSPmLWNJ
ArUlLAeSfF00MuO+N9mFJ69DiuPG+J2RUqrpVq6Ul2PosPw/jPYqs4PlZ//T+42lYLlCVqS1v542Lc
LvtGvYUzIBsHmPCFzkgEEjHfHGf6kw/EuWYr+BWTv5noPLq1J+9Gx9S6fq7OkSyYBPp3rs9OuBJmMk
4a+h2YarUTs2eu2qLc+U+0LcBBGr90XPRfQd8V+cHwh/Z2+Fnxf+OX7UeqfEHwJpXx/8Ag+9Pf+Ka
J7gpbAhuhzjtX8m+P8XQyuEoaT5vwP6T8B6i+vVJPbl26Hsmn+GtM0qK103TNLsbTTbeJLa1ghhSOC
fHloNlXP7MmjJ3JxnNdar20a58yBBj15/WsHUtdtIZBb2iXmq3jfcht4mcn/AHmAKIOOrso9666WBP
zm3to3mmXkhynlhc4IV+lfr3FoWpa7NHeeJxGltG/nR2UIEkMZ7eacjzXGMgAbBnPzdR+dXhfR9Ps
rIdZu4/uoiQIRGGx1AkeMjPGFx1Ga+nvH/jTTfhd4N1DxBeKbrU5tllY2jDM19dO4WOFO5kZnQex+Y
biBo5BAVKtdNGwyhlBYKDyI2wfvmv9FYyaSR/n3VTVRnlus+HPEPhWU/GeWePV/Ggl3eJFhkJ/tHTX
a5P4fGbwT4s1D4VX0yHTnWTUPDkm7AlswcyWyg/xQuy4HHySIRnDFWd9J3VmfRvhWzWDVdAtdkQiV0
2+Dvxk8Y+D9OihsvDc7/ANs6LHGgVIre4ZvMgVR0WOQMo/2dtcf4G+JOs/Dm4j0ry73Wvh3cu4vLCN
dzxiv8+fEDMMVwfxfUxWCfLCo1LlWzT3P96fAjKcq8UPCuhgs/ip1KadJTfxRcdpXWuiaP/V+yfBHx
X1yOoIH4h+BPGE/gXV92oX0j/Da/bbexEMzaJOxBN4igHdATgSxjlQ3mDoa+sdH8W+KPhjqwMJ1
/wDifY/b/wAJeNZvD2q2d5GcxMwVxkEMpPQ19j2txFqcOl69p5jOUG8j+JP4yfdfdTX45fCb4v6f
rGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb4v6frGmU8Jj+JP4yfdfdTX45fCb
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

yzrNlnL20rN8vmpv3xscAuMNgM1fnn8Pf+Cl37Lvj2JNP8R6zrHw9vpFBe216zKxKfa4jEkb855+Xq
y5jvDfO8urKjmOGnSd7XcXb77WP1w8N+LNC8R6fpHjjwXqMeseFLpTDNIgO4EfKyuh+6yklXDYKkEr
pDbw2HiHTbYEz3Jijx9utF3EM6hQDESTMgGSWGG9l8aeF/D/AMb/AIf27eGNZ083yyR6l4a1iHLjT
EUnHb14FdtpOmKQiRRmPnIEYxz6/WuC+Fd9/wk1hPZ31kNL8RafM1pqlk6sXtp16jdk5Q7tysfvqVP
DXxA8OWuuhogYLsLsvLN15R4Z1w6EEeoGOOc19P2tp5aIoGferb2ZkUrsLA8kV9nl+GqQ96Dafqfm
t3p9l4f8d32rWrKl1axeHr0zQMRn50KAA8Yxuxznd0qn/wAHCP7Hn9o+EtD/AGs/B2jST3+hINP8U
4O9Tsyy4bNfptHxEzXDYJRw7UnHR3ucODwuE+sxhidKb1utzivHNn8dNd8KeIP8AhC/B998PbZIWEW
oHjPwn8TPhAuqnxzpi+OfB0YDrr2l25M0SKx2reWYAKsqjPnQ7lzklB0o8M3+keJNJs9U8P6lp2qaV
6O3swx8sqSc966bTvDFxqEvlpagt2JFfBU8I5u0NWfW18fGkk6jsj//V/oMbSrW6Ku9lbTNgdVzxWr
HyLdeFZ0Uuu4P1Y45I9MmvyC8YatBF8Vvjz8N7G9urbxneeJYreBltijNFcWVs5ljJ4cRjzmYhsoSc
C6qqjG7fX6p4RY2nkecRx2K92nZqTfTt958rxbU/t3KpYWlrJdvwON8Warp/j+0sPCfhhlPwl0lVir
AGe0YgeZbsSCMAMmdjAFeXxwPJIXJJzX+hmVcQYfHUFiMM7wfU/jfNcgrYStKjXVpIs6dCFKHn2qp4
W6Gf8O/GFp4u0LT/ABDbwm0vUlMF5au4DWlzGdskR4J4Zep67hivreyIkiR2wucHk4xX566vEPhZ4+
3NZ1ZRSd/wCl1fyMJ4ZyknTV5dPn+vY//9b7X/4KYatY6d8F/B9vM8CX03ii0eFWbfny0kkfbgdVUE
YAhD/q4x/d3MeWGPx0+L/jZPFPiVraB3fT7QmGJexPdvrnI/Cvwv6QXE+FzTPm8I7wppRv6N/wCZ/v
XwS8bSf8Sq4Lt4R1BnwIwIyZNPZ853A4eMHggOvGFB/mc0PXtQ8OazZa3pV1La30Dh1cHHcZz/eB6k
4PxA4aqZdyKOKguanK2ql0s90u5+jnibwD4p8Bax/wAJD4f1G68P6uQFj1S1UeXdIpYqk0eCrr85yr
Rp+u2SNgRXSkktjHypIpSVeuASO1c54++Cdle210Layj1TTZBg2bg5iwOCjYJB4HPVegr/VrL8xw+P
rCcXcSzQypcQ4+R0P3h6/T6V/Jx4Q+LHxK/Z/wBRSO5N/wDEX4crN/pMbqXv7JMHoy8HGFIfyzwcPl
q4J79qzTbTQqyx7pE/ug5Y/Squh+JtG8S2hutG1CC/gHBKN90+jDsfrW4CAVYH5vUV+fVYuDtNWPSh
j1eQI+M8HvQVzwTkVzOk69a3kSr5oLYwBmt8OSAQxxWFSg1uiTyb41fB7wz8ZfBGueFvEEMqXstncw
AuVAxiv9NUAEYbD8cAjqewr+BD/gpD4Isfhb/wAFJfiH4cNu8GnN8QbLXrZXA2MLxYbnAHoHnb8TX0
cMhU4ByB7bcwsA3Br+bf/gmb+1P4Y+DHxr8W/D74iajqlr4e8a6Va3VrqUrFrPTb2zkaMiZukQeO5j
tKcg1tXFsGxwxFZ00DjnaBX4Xi8J7Lc9ijXTP//Q/tfop4QkZyKChAySK/hLkZ+oDKmj71DU0fenDc
jvTCGcgBTmpNvmkBTVq1hLkAqSAa6KNO+pn7XXQlt4AEGRvOMYrTSD93uLbEzjOB/UfSkgt0wSQjDC
wtmef5lDLMqpupUm1ZJX36u2yXc9l/bY/b48LfszWF18P/A0ej+Ovjzc2guINMkctbaTE+QlzqDJ9y
gm41yw0rVLzxn8S7+Z77UtTvLlrid7mTBkluZGbLTHA+Q4C8KoVQAPy11fVtT17U73WtYvrrUNSuHM
f3UK+iDj8+przkDA2hgwI596sVC/3j6V8zRaTP7Cp0VCKjHZH//T/AAU1uFNP2t6ZpjkbTX8KNprQ/
TfWehpJqaok+qsCLdTysJI5dh39vQ1iFl9ailMZ2+v8AKuxK55FWvymZcvJcXcs8pZ5ZGLMx7k9amf
ox0HGeKueW54KnNV2jY5ypxWlPcmpTd7MgbBdCpUPnAY9q6x7t/Dfh9ILVvLvdQO5wpIaO3BGAccgs
kbgfia1bb4geLYScazcyLjBEmGBH4iuUMbk8YApDE5HAxmrhQjPRq5yTx9aGqqHtHhz4pa1c32mado
USuP9aTX8lX/AATW/ZvuP2lf2p/h74QaeSz06K9huLi4Uj9yikuzE9RhI5GBHII4r/Qrh0a3hktLDT
9Guro7o4hHGTlQRjjtxVyD4Z+M/COoXXir7oYdbC7cTavoF6zi01HC7fMt3Ug291gkFuY3UnepYI6
Nzo2pKYBrdmCXe3mRvlmRSxeOeMNsYE52lkb7C+FfjzQvHlpPHb+dpHiayxFqWj3eY7qwfjG+MscqS
T9Rtb9zdWagnEYlfc0kXzEmKQsh/u8DH3tLCJ6yPx/HZ1CpseyizIVSEJP8AKp4ocKcjHGOtYmk2Xj
pV/aS209vIOJo3jZXT/gSMy/RjX+eR4S8Z/EX/gkWkV+wTUTe7iv9Tg8OeHdYeR7hP8WPE/b+Fuk3erSrL4h01n0bVCS
BMG4C4rkozdOsk/hlo32/wCHPby/FTrYWUY29pB3jf8AH8D+nzwj4Y8caFd2XiDwlN4Y8caFd2XiDwlrFnD
6fF34J+MP2NvHWxnL0o+Jfhv9ufwG8Ud8D+xJ8Ufh/8AtyeAHijvgf2JPij8P/8AuTwA8Ud8D+xJ8Ufh
ujeIYWa40bWLaJGlsLhQOCjDEsRP+siclZFyD2x8CXdl4q8IeKbncwR8QbSz0vxbIG+yy2bOtnrC7QW
V2r31l5qzwyRuYp7OVWDJLBKuGjcMAdykHAI6Eg/S+G3idiBwGPA+HzfD6wiR3Ec0KyxMrKwyC
ODJGQmSudua/E74sftkfHH7sQT6b4z+6Ez+6Ez+6Ez+6Ez+6Ez+6Ez+6Ez+6Ez+6Ez+6Ez+6Ez+6
F1x5dxjKQq3PDkE4J6ivxz8d/tLfFMll4KsVNvb6dZkq11Gr4ia+mHNxKIxEpUBY
Gw37ql+L9T/cXwZ+h3wrwgo5jmEvrmIWqb5eTX+WN238/I9H+J/xJstA0qTw9pEsf9szfu5gj5NtH
2R0BcnHzY5z16V9Ffs3/ABY5/I9uGj8M3xSG+z0jOsUUkAsOhH94
+ZT9mv4jL8QPAM/grwiNtH/hJ9Z0jwpYeSqQsUUkAsOhH94
7wLY+J4XMITTb0KQHSMbZTjAMi/xn3r4y8a/CH4k/wDh3Hb/CH4k/wDh3Hb/CH4k/wDh3Hb/CH4k/wDh
z+GpPtnw3+NZCNzNzvuS9gUjdcW0jD/SLUFkG9OV3KCF4r9WfAP7R2h6vJHpPxC8Ozf8I9rGkeFL
txefDT9pTQYLi4tEC23iWC3GL+BQN1zPalf3RUlN5QEIMnCoS1f1VnPD1HEe+9GfDwn1R/T6t9BPHH
r/CH7TfgvVCfBl+SNomJMM7eqOBjPt25zX53mfDlehVUqOpUcQr2Z6VfWOpaNcLPZlJbUnc
p8Q6NdGz1QqIzKxLRzqp3hLhARuUMAwP3g3II5B8upF1Fyvc6IyvqJ24SiUHaYyMgevv271/Gn/wcW
GQpIDAHGPxA/4OOfh5H49/ZK+Gf7Ovh/4ieIPhz4g+IPiDxBqOm+IPDOoyaZqtnZ6Xp0l5Ct1bqyyRNJ
98P/AAT8AUPhfPrB/qu4Ud8D+xJ8Ufh/8AtyeAHijvgf2JPij8P/8AuTwA8Ud8D+xJ8Ufh/8AtyeAHi
jmZuEhlJ3Fj8kchdztByPg8flFKo5YbER0u/+HPq4Sw94g8g8N+NdAHhvxroA8N+NdAHhvxroA8N
rldsVw4zkMAkpKltpy1fVXgbx94W+Hvg/wAM+Ffhb8SfEHw58QfEHxB4g1HTfEHhnUZNM1WzsvEO
rXlhcwrdW6sskTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0
kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE
0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJE0kTSRNJ
E

n50wAnoDXi2Z0iVYT7oqvUyEYx3qobgSUUUVqI//0v7aaKKK/hg/TwooooAKKKKACiiigAooooA//9
6n8KqwRMrk45/u+v4jgfjivyc/4KI/t8XHwjDfAb4KXcMnxgvofN1jVsLJF4TsOG3lP47yQcRJ91B+
iG2lRl8LhgP3MOciS8Ktk4yIQVL4Yqp/k8+P37Qtt4Nt77wX4P1GbVvEsjE31/JIZGhkk+aRzI2S8z
Ek8n8Sckk8/hXxvEGfvEScKTtA/3l+jh9G/L+CsvjKpFTxkvjn+i7JFS6nlvLqa8uppZ7qRi8kjsSX
zIEPRsgD368+nTNfwTGEpbI/62quJpw+N29dCD7oGe1VJHABGDXX+HvBHjnxc8S+F/BXizxFvO1RYa
5I7BhxVST5S2Dzmrskaw+ajSqXVygwDyQSD1wRgqR07VnsxJcg5FZSoyV01Zo9CviISpqVNpp9U7p/
bGKN8BPg34h8QaM03kvrd+psNKtxnBZruYBW28kqgZuOATxX0WW5bWrL3EfA8Y8c5Tk9L6zmdeNKC/
HD6P4NthbQxZx8rXk+53I5GVjT2r9rfgf/wSD/4J8fAS3sF8I/s7+ENf1i2C/wDEx8Rx/wBrXEmP7z
/wBmFWPYnHXCk9Aa7u4/Zl/aUQafLL+z58dba2upBFavL4P1JPtT7SxWIGHMhCgsQoOBg9xn/Ul8W6
xeFNPYhhYQE5BY4HmyDmRwf4VUD7TIPDali5ONRux/PGb/tAs5lO2FwdNR7Nvb5JH+VFrvhLxP4Xu
M3NruGFeCfG9GG4ncCMbe9f5f/7QPw4s/hF8cvjH8K9O1RdasfDXijUtChuQeZ0t7h4gzehIVc9edw
Wm6tI4O4bX4Kj73+GPoan0vQ9b8Q30el+HtG1PXL9yFSK1j8x3JPRVHJNfA4WjOu7UU5PyP6izjH0N
7BttfRK/8EkbqW3Vb344xS6hMywBYdCPlF2YKqgmbn5ivb1r9Uy3wYz6rQeKnS9nBK93ZO1r7X7H85
f1Z+HdDWeSKWCNUixuV853Z5BB9CDn8a+Yf2Kf2f7D9n/wDZ2+EPwj0x2e10XRreBpHX55Zim6V25F
UHORSWlryDj/AOtXQw26AKRgdq+0wGGUdT8JxeZOatIS3hG3GcCtBQsYpEQJgDkUrAkcV6uz0PBqSu
2wSGB7MqntX1msgLbc/NUGpWqXdlc2rqpV4yDke1c1enzU2vR/cdGDxEqVVPof/X/Jn9nj4l+Pv+CX
MGV43QMrqR1DK6HI7V/B3/wcLfs/weGP2wND8V6VoSww+LPCq6jI0Me2OS/tZxBKzc4LGI2wOOcE8c
rvD4p8vw6Neh+4ur6VqOi3Yv9LjDXCsBJCW2rKvpivPfiV8K/DHx80m01rRr1fB/xW06MR2GrCJdww
1E+WUdU138z84LLWdUsdc1Twf4v0mbwl460+UR3GnyuWEyn7txBIQPMgbGfMIAXo2D1+rPAHxivNKs
b+G3j4WY8eaQEd5LcbYdVtHbbFdwKSWCuVZSv8LqRnBFfD4rLq2Wt1cPL3ZH69lWd4bPaEcNjbKtH8
Z09jAurC3lvmvWhQzjhWPLIO4B7fQYHA4rk9WVizBAwwTu5zkV6G8PnFQnDHkA4y305x+tcFr91pmj
gml4F13Wte+LHwHGh+CPiAbSUSaTfWaz6RqDEM7FIh81tMSqASR9Nx45xX6lz+LfBMUcc8njHw15Lt
dZxT3tKyenqfwkftHax8NPFp8Oa14O0d/h/4ujuZtN1/Q0wts0yq2Jodp2gM6OrcKxyuQckj5MlWdC
Xx78L6P4x8D+BNe+E17qVlHeW914bf8A0Z0ddyu0DFkbIKsNrDg9RX6fxxhFUlTxlODSnFXVlp9x9P
8HIzGM6cpt9Qt4xyPNtpOCfXY7V+RXiLwZ4p8Gaxc+H/ABd4e1vwzrMLMktrfWrwyqQcEbHAJx7V+k
0bwP8Q9H1JpxFYTt9luQTgGNjjn6HB/Cv6Ff2RPGtjoXxW1jw1dagY9O8T6XHLEMDbJfWwLA/wDXQw
lnEmGxC0jOXJL/C92fnX0jPD+PE3h9mWAqLmqU4OpDycddPlof/R/XIANyp4q3FExBIOBXnHw78W/w
wLmx1y2uEuNNv7R/LurK4XLRyRycDcrDucEEqcgkH0NYFVegBqRocptZcqc5Hrx/9eolazT6nRGpbU
lpHiC2P2fV9G1G0aG/06fO0xz2rDzM5IwyhkYMGR3Qh6Nd0GLXUtyXurXU7adLuyvbRhHdWVypJili
xiMln7RxdTDN631cfPvY76GF+sPljZSPMPDPiHXvBEit4O1u98P24y5ijctAfXdGTjn2NfRHh39o/X
drOuHO1DJPDPCY25K8ng84BwpBOep6enIFe/GlgsfRVeg48rV7o5m50JezqrU+zdTuvhv8SNKtJNK1
DGGbbja/McuMIQwKV9Bf23p1gZbzVXtbOyhU+fNcMFRR1OSewAyc4471+VH7X37Wdx8RYbbwn4UuJv
TbPw1oWvX2raleQabeaDHEhuNRVivkhFBIMo37kGcnd5f3gVH6T2Gp6ZraLbSMYYpU8qe1uoWDxo+N
2sGja/40sml164spgkNzdSWcEoMijGLjfI6scdSc7iMn9hzuMMRh4YyK1f8Awx9fi8RdxutJaH07+x
SZdUv7JG+1W7ubHxHbhDtgv4cBhIoLIshCzQs2CD9xv89TSfi7rswjgjvNO1GydgFbcGRYsfvFLg8N
9pTdmf0Q/Cz4rad8TbC+sbmxufC3jnTTHBrOg3QP2jTpXBKkYGJYHCs6Tp8jKDyCCB6XPbMpPylkzj
Twv8TbGM/aLJsNb6pEOl1p0x5ntjnJGA8TFkfGBu/BePPD+NKTr0VdPodeU5q5PlZ6LcI4JCqQc1RA
Co2Z186P/9P+2Qy5PGMVICDyKr7eamj6Gv4VjK5+oadB9FFFWIKKKKACiiigAooooA//1P7aaKKK/hg/TwooooA//9
bji1XXyrQaFpCTLHcazfFfkgjz0GeWboqgk9gfteHcsbTqT0iupz4elXxeJp4PCxc6k2opLu9j5+/w
WaSV7m7kEs91PJxJcTEABh1OMDcQqjaM17X+0V8dNcudW8c/GL4ta9/afjjWJ/OupI8+SpxiKzt0PP
2PNjqmspPeKevKr9F5bnKaheXeqX93qN9PJcXVw2+d2OTI24Nknr1Gf8jFcZHUZqwURVL7CcdeCf0r
w/g5YvOK8aUI73ev3dfkf//V/BEEEZI4OQMeuP8APpX0N8Bv2Tf2kf2m9Ti0z4I/B/4eeDrqzW3
jOa/oE8IeDfBH4uE8Ad+x98PWPhXwV4W0dwDh4btIRBa2OnWiQQW8Y6KiIAFHsK/krLeEKFKN8U7v8D/AGV8Vf
GzyP3V+CX/BH3/gH4Q+Ddpb/tCfBr4eeDrqzW3jOa/oE8IeDfBH4uE8Ad+x98PWPhXwV4W0dwDh4btIRBa2OnWiQQW8Y6KiIAFHsK/krLeEKFKN8U7v8D/AGV8Vf
NoisbWO3QY/2UAHTjpW89hYXKNDdW0U1uwIdWQMGHoR3zU3mbs549K+K/wDgoN+1D/wDgoN+1D/wDgoN+1D

IsbR3z1rQtUyG759a5iGXcw5NdNazKVHUCs4ttnXVpuMbWP/0P7zljBYYHNOK5BBOMikVu6mo5ckEr
RtqI1PSL1JdSs45CseqWLlRPbMewdNw9AcHrnP8ASV/wcW61c+J1+EvwhsWaJbRbzx1dvGoZvNgEdp
Hw5/aa+D/gr41fCzWbXXPCOt2a3UTxNzav0kt5V5KSxuHjZTzlCcYIr1q6sYpQyum4fSv4X/APgnN+
wXrFql5p2oWUokiuYWHBBHfsQcEEEEDGK83H5e97aHnJunG5taXp0VmWMZIUsGK+hr5l/aP+D39v6r
YqGKpuzi/wAD8zo9BvAeCrKTwQvWuksfBmq3gCfZ5FjPO/kYH6V9iv8AD7RrckQ20CKSTwuDW5pXhr
/i659q0b9nH4MaOipbfDrwo7BCjNPZRzO+RyWZwSc9+ea+hIrSG1iUgYVRjnt/nFY02o7iyxQYI6n1
6/ma17smRSzEgkdPSsOS3IbjmqrWtoiqFKTd3UZ/EJ/wXe8Eaf4f/b58VSabYx2Nn4n8FaVqrrHEUS
gWmlfEr9mzx1OnkPd2OraE0xJIJRre4iQ56D5rnpg9favqP/AINzvij/AMJH+zp8aPgnd3sEt54T8W
tRaNd6T8ZvhV4Y8UXAVhBfiHyb22YgjMdxGVdTyeh/A195TWssigFSfwrhtc0yZdxRTnHXFfmmOoyr
DKMMsPwOktTbTSQTpJDMrFWRxhkYcEEe3vg+wr/UhvrK6NwgcvbzECNTkgKM1/OF/wAFvP8Agk3pMf
0041Woxn1Xa/k9r9z//S/mtiUKQBwOa+0f2Z72PWNM8VeBbqZWa5tnEeGI++nAP03Nn8K+Ndi7uhU9
/wDh/wDDLx3DIZri80q1ublQCgkk8sJIvJOFLKwHpX6H+HdYsNe06PU7GYPE3GO4+voevFfk1+zJY3
l5BHBZ/jMNDaFSUfudj7JU7wAR2qwf8AlnVHT54LuCO5gkWWB0DKwPBFXlBP3h06V6kpXPz6cOg8hT
Htg4I9wR0PvXLVw0akHTqK8WrNPqmYOUoNSi9T9MfhJ8RvAH7Yvw38Q+Cfir4I0hfFdqWs9e0WdzcQ
+9AC+ZXAeFPEev8AgXxbpHjbwlPFZ69YNhYEKRQapbsQrWNwSGCxOuMPjMcqRsAwBRv2p+HnjXwr8X
o/bH+Luoa5FH4b8NX9pF4As2gj1eWBg51m8niaaG2DA4MEUSLK7Z+ZpYk5G+vyA+K3jqK2tLuyjkjN
T4y/GeKzvI7ZdStRrV/ciFDvBFsScbzjjZ7dT0GM19bjuIamY2xEtE1oux7WXZfHDU+SPUzpfClt8X
CuSuQoABxwAAOAMfnZpfj3wn4MubJDrUFm6DzEG/97MxHzu2TguWLcjA9q+iPBPjLxp4/Yt4O8H+JP
RnpX9LH/AAb5/EHT9K8BfHe31bTpwP7fsAbpGBLAWm3BXGeA/r3r8Bdf+C/xw1rwvf2+o3fwv0eK7t
jd3ZFSp1K6hXlZHlZ9CcsO/Zo/s4sdX03UYEvbG7iurduhHUfWsvxT4X0bxnp9nYatdXOn6lZuZtH1
/iEDRNYl+WMMMQ3LZx+6c/TO0/MPWvbzfIld8i5onwWCxUoO6Z7N8P8A4h6w2tN8MfiskGmfEq3iDx
rSLtT8lxbSMr4ILZwwZTk5BXIpfAvxB1ux1K0+F3xWfTLb4kJAXtb20XyLTxPbJgG7toyX8sjgyRbi
PA8k3GPQ++w8+bUz2UqcYzUtMQAF8gg9qfXiuny6HoqNgopMj1FLSAKKKKACiiigD//V/tpooor+Gl
xV4b8DeFtY8YeLUstC8MafBJdX17cS+XFawIpd3ZjwAFVjnt25r+Q79rv8Aaq179qX4p33xR1iGfw
g0zcOD5aeXNMOcZiU/MSK/mj/ar+LL6dbw+CtJuXhYv4Q8Ewz3UP2qfUdS1S6dprvxDqEO
1jfBP4FfGH9pHx1a/Dj4a+HLzxDrV5IzJFtGIwH93xnJJBOckcbs85r3M8zSNO
1jfBP4FfGH9pHx1a/Dj4GfD/AF34keMXCyyQWSZSwhYkC4mmyscMYIA3yEDnA3Hiv0+/4Jz/APBGH4
B7qcnzLzUZgMGa5nYlpX98nGcAAcV5WS8KN2rYjRdj9c8fPp0ZXkUamUcNxWJxFrOf2Iej6v7j//1v
eF/DlhbfZrGw06zSG3s48jCxxooEYwMbRgVsS3ACnJBGPzrKmuyC2ACD1wetfy7Wzenhb06CsftnG/
CcVZgDsVJbHrVWGEs4xlT6Vv2ttGo8x2UKO7HingMDVrVVYwq1kh8EfB3EMPSv5V/wDgr3J8Yf8Ago
df8AEhg82y8LK65VBIQY5751yy24J2KoeXYpRX+cNF8DeFfhNYv4Q8Ewz3UP2qfUdS1S6dprvxDqE0
4G/sAfsw+H/E3xK+N3jjwZ4Y1Dw/4Z1DxDqV5IzJFtGIwH93xnJJBOckcbs85r3M8zSNO
7toCUOhaTIGLRwI6SLdXaFTJISi4VSB+0Wh6PonhTRNK8NeGdH03w/4Z0DxDqV5IzJFtGIwH93xnJJ
mleDvB2i2Wg+0WRNK8NeGdH03w/4Z0DxDqV5IzJFtGIwH93xnJJBOckcbs85r3M8zSNO
Pwloeo+JPEeowaPodpA9xcXMzBUijX7zHuccDA5ORjNeK+C9I8T/ABR8/ABR12P4ufFHSXFHSLrwr8PLOzCvh
GryRt5YIVj1C+j4HngFvLhbJiB3N+8J2+/axffawphZvJXjaDx9cVHqmptfyoEKxwKPlRTwK5zVtXs
8DWksqpEYLFHIUajdsSkStkHCbnBZuyB/UV6R448V6n8Q52tvOaw8IB2zCOGvCDxn0XvXnt+II7aO3
N8cUPCqgI6sSfSPEd1DDa3t5cOYIosuZHzuiA7nPPHQeldTcznCdoJOcV8ofG7xv5NtN4as5F8yRi1
Hd83QY9fav6+jBrY8FSSjocr4i1OPR9Mv9TkVmNujSoAxUlwCRyPev6Mv2QPhb/wqqPxd4Z0DxDqV5IzJFtGIwH93xnJJ
ZLZY8KW3EkqcjJA/+tX8w+OGZSrYynhYv3YLXzb/AMj9K4coezwvtNnI6GwhGCQPr710UZC7VxnpV
ksxYsQOTz2pNT1HLFwSi9APWvnz42fVtVikD6u8LR2MYYYuLhvljj98uVH457457457V4mOzOnB6
2K9hkMckYYfexsAz9K+4tOufQK6EV+YZdi41Fzxeh+p59lE8JXlh6qtKJ20EhPBBIpl3IH
kP8AwW2fuJHxcK44x64zX6Td1DDa3t5cOYIosuZHzuiA7nPPHQeldTcznCdoJOcV8ofG7xv5NtN4as5F8yRi1
JkQTq5jXccK44x64zX6Tg6UY0IR2sjqhC54x8T/hNg2K0nWIlWdohKjR4XKsp27WG0gkoeCfXgr
ilv5RjG77o9FBOcV8BX+oyEzkQJOcV8BX+oyEzkQJOcV8BX+oyEzkQJOcV8BX+oyxal/aFpuWSJxIjKvORzkEHoOCCD0wc4rKvBOLiY1ad1Y/0zdE17S/Er
btLDw5eTSjZoOoyHK2UzFiVt52yIGzhJMx5+ZFr+ne4uW81gi7snA4NfnuYYSrRnZrTuYUsMr2Z//9
tCKVn6jtW1OrfcToqLtEgmgUjhe1URbjfzwfpWzkeoqlO5DEquRXPVitzem3c/mA/wCDlfwRHrX7OA/
wI+tfuD/mHBc1p/hNq/hNZ/hNZ/hNZ

```
e7eIJ0yPOh3EnaeQTlSDg15/BHidPB1Vgsy96nHr19fM+dxmQxn71PQ//9X+l34f/G/XvCSxaR43nn
bwPrz+BPiloS+G/EpQm0YMr2WtQjrPaSjIOf4o2IkXHRlAeuv8J6x4l+G91FfeHWmuPDYc+fpzElAN
8iNsEMP+Wb5X0r0m6hXZlcMpPBHANfPou9E+OnhfSL/wzrh0PxxpT/adD1RsCXTLj+OKYfxwOPldG4
JGVzjOOc1Q3Fhjv/ADrqbq12AgrhvrXPSRkTYx/9av5+x+BlCTTPrqGIutSqpKnIqwDkA0OmVBFAAU
9JX7V8Q9XlOi+GoAhcC/dSfNkABxHEgaVj/sgdWAP19Pd29hbT3t1MkVvGpd2Y4CKOpJ9K/kn/a7+N
BUffrSeiUY62fr0XU+J9butb1Sea00Kx1Xxr4xv55TZ2dojTahrWpTOzFlUZZpZJGYsc4B3ZOBmv2F
NGlWfjXxl9l8b/Hi5tlF9q0nzQaUrjcbXT42H7uJdxUyH95IwLEhWVF/SM3JChD97npwB/n9Tk966s
uBnksTVF3yW3MWPpX8hZpm85K3MffUqN7SbuwaZi/UqelV2lkDEtkrnFKJPakBzkBc5NfL+3ct3c64
OvEnjfxXf/Bz4TMLa6gj/wCKo8TZIg8OQMv+pgOMy3zqSQoBWJSHY5wja3xG8a+J9T1y1+Enwt8q38
nPX1I2/0xwFwdD3a9aOlro+QzHH22OBurPw18I/hzaeD/BNrJaacN0du8rl5bmRz++nlc/MWdsHceW
4f8ADjLf/FjxZqUPhvwdpqAvKJ5pkhe7ZVBKxx+YrFiApJRBya/VL4O/DLw/8HPhz4f+HXhmFks7NA
bpa2lrAmyKGNVAVVUcAAAdAO/arDkSHJqzcyAyEDrVTBGAoU5OBz1NfzBm2JqYjESnPVtn2qpRirsX
Pbpx/e6V4T4d07WvjRrUfjz4g6Zd6J4DsLs3HhTw/KxQ35VcLqGow+rZJitySsa4dgzkBPtuEeEpYz
n02KF1eS4JPHTb6ZrFeS4mYS3ErzSkcsxyT/ACri/FPinTvDtkby+m2tg+XF/FI/9wD1r9/yvJIUFC
r61feINRbWNcyzg5trfPyQL24PU+/auUv75mdweMnOAeBX6LkuV8j5prU8rE1rbFTUbxnXA2InPyrw
RwBX5/6lf3OrXVzf3RLSyuZG9jXovxQ8bzeJ9amtba5aTRLfC24HSVs5Ln8zj2rifD3hzVvE939l0c
H6X4F065muZo7mVVJdz1Y47DtXyt8V/FL6Housx2l2sOopam4upMbzaW6bgzBf8Anq/Kxju2PQ1vi8
5PucAf0r4a/4J7/BvUPgr+y/8KvCWs2iWfiaawGs60pO4/bro+fIGYgEuC4U5/u+9ffUCqSuB1r+Hu
TzWheTsSOWFcR4g1CCCGTdIFIGTzXm4jEqO572W4Rykoo//S/s18S69DYxSyySIEyAoJ6mvzS+KniW
eMn5sjpXy94a0W/8LeDtI0jWrlL3xROW1DV7hQP397LhpHJ785UHqVRelfyrxRjJttQelz+0uAuG40
047fSv52f2h4tT8O2/w6+Mvh5jFrXgnxJZ6sjKQN1gzLFdK5xwCjyMcegr9//CevW+raLp2pWwZrS4
RMQo0+3P+jW8gH3hJcFXKnjNuucA8/vL8TPipofw28Maz4i1rUbews7O1lu55ZXVUgijUs7uSRgKAS
pmqSxokQdQo746V31n4s1W2VVjuywAwMnOK8bUlFO1ufaporySIZLsa+/PIp1W0ej674lv8AUo2iuZ
+0b/AIJeftswftBeBH+E3jzXZNQ+J2gWcM1lcXUubjxPpBBCXQzgvLGyNDNwMMFbneDX8RKHjqc19X
3N37VGyNtPFeGfs7fH/wn+0Z8KPCvxS8KrLbRX0O29spCpl0y8Q7JraQAnDxyBlx3GD0Ne8bgVJ59F
eZWIUUj+c7/g4bV7r9m7wDpvJku/HehxAZwGxI749/8AV1/MZefDix8WaQthfQrJG26MIQcuG4IGPX
191wphUsNL+82zbO6yn7JPpGx//U/oS/4I6/tVXPx5/ZS8MeBfGuppdfFrwEv/CJ6yWbL38VsTDbXh
ldicYz/UZHN5sAkVgy4ZuPQA/wA8V8FnuXvC1XHoz9Poxi0U9RYSEs2FQDBJPHr/AENcVqNnDcxu4c
KBgAHrXxeJafu7n02AhVotVNj+Rj/g488NSJffsq+ILaHULh418RWriNQUSPbaSFz6BSFz6bh6iv5k
PwOi+Gvxu8WeA9TWQ6Lp0sN7p7liTdWUi7omZiOqcow7sjc9CfL4j4LqrKY53HWHM4NLo7q35n+jP0
w1zC3OOpU8f1r4c/4J8T3Gi/DPwRBcvLbTasl3qLxAnDebcSyJn/AIAyn6Gv0svrKLVrK50+VM+eh
x/nXlosprC7uLGYfPFK0fPsa140ZPmQlXxjIr13OzsfJzipI/9X9tyoUkjBXqD6jGaibJbIHBrz74
qhwSscZc498CuskByw6Gua8WaIniTwx4i0CX5o7+xns29xIhU/zNc2YVXChJx+IxoJOaT7n8knxXvt
R3JDD+Pyxkn0OVb6kt6DP5T6tCwu7iKQYYMQwNfy9mDbxM79z9Ppw5Ukj3n9kHRPDl/4g+I3iC7gtp
Yo7jRdQVhMTceWIcMQoJYbclWUYJHIPpXr178W/GXiWOb/hHLDT444HZGluLgzBgF3Hbs4b2yeufSv
8z7G31XxWWuPEWpanqVzHcSK0JuGSFWDdBEm0dD3zXpNkltYEJp6C1UcZXGR75xj8wa8vEUJU6vsp
rX81n/BAL4Ua38Bv2Y/DXwL8HXmoXGleHLOSyjvNSkjeZ3lkZ2Ro75xj8wa8vEUJU6vsp7
f4t/Z81K00nxpeyeJvh1cziDSvFSQhHtz0WDU1ACRMcLtmB2OcghWwD+oBuOQGLHHaqerWNl4j0u+0
2flnAGW3juOepzwRk5OB69D40vtc1Cy8daBB/ZvxU0y0Ed1p8nyQeILINua3kI580Au0LYO1yQTtZs
sjaArhwR97ptOOAO9f0nSxWBzrCucHfyPl6LqUZcs1sffvhvxNo/j3w1pPijwPcNeaRdxCWNihRgc4
4pIVkAYEZHb/AD9Otfx/4h8KTwFaWj5X1PrMsxkGrX1ObnRgpwPl7UxEbYOKtXSOBjOTimLnaufSvx
dfArTvhP8NotTT4/8AhDwpPAVpaPlfU+sywkGrX1ObnRgpwPl7UxEbYOKtXSOBjOTimLnaufSvx
iHR57mWKHdIcskcKIuA0m/6CMvJVVVEGBgeg6D8Mn8/pX1rzqFOnGFNXZ6FPiXGU8A8roPlpTlefeW
E7se1OVgcLtwKYykE4zivOqzctju9RYwWGSCtTIp2g7aihMgx8o61qWtuXxuBHNdGEw3NY5a0uUsWc
aLYWsl5czucBI0Xex9zheAOT0HWvLPht4d1G+u9W+N3jDTUCWSBl5xO7fv
kv7Tvxx8Hfs8fTH8T/ABzCWnLvt4OaUCWSBl5xO7fvAfC2nDJjkIxJg5OCS3Tn3OOw6
AfC0s8zrD5fH/l5JL5dfwP/R+rP2DLvxT+2/wAFN/DHxW+2Idx+maDZXvimOxyz6NaWwMFhaqWyA
Uduf6NWBI25AJr8czrH1cQueo7s/oDxkhh6GezwOF/h4dKmvPl6/fcwpYzveQAtwTgdTx0H5VmeIde
PAJLHGAoySSAASQK+dtB0rWvjVreifEDvjwl4Ut/CXhS38JeFLfwl4Ut/CXhS38JeFLfwl4Ut/CXh
J3Abfo5/5I6gE5z98f8XfuTvb8SbgBt981981He3md2GyK5m4vVDMGZR/WvqQ+
e1cR4R+FEk8h1nxNcNd6RdxCWNihRgc4
MgXEPzNZeHYXUMjXQUhtxUgqgw7jB4B3V6kG9jaKUjZ+J/xQv7nWV0rTobXUPFd2PNstPe42RwxAgN
wymrahLcy614vvpUudT1e62G5vpCDx/sRrkARj5U6gE5z98f8ErfCVx47OXe8f8XfuTvb8SbgBt98
zmumgA+XgYrLtxnaxGW6DPYVrw/wmv5ChOLVo9L80mDxWpNNtUrkf/AF65K+
fC34eeNPiVqEfhr4beGtR8Q6pIpaOx0u1aeTaOpIUcAdyTgDuRX0Z8Ov+CPP7Y/jqOO+8Y2Gi+DbNsE
qzXAvLpR/1zjyoP/AAMV9Ff8Ezf+CZ/hf9jPwzH4y8YRQ6p8QtUtwLy+wGjsIyMm3tsjjPRpOrnnAwo
fC34eeNPiVqEfhr4beGtR8Q6pIpaOx0u1aeTaOpIUcAdyTgDuRX0Z8Ov+CPP7Y/jqOO+8Y2Gi+DbNsE
qzXAvLpR/1zjyoP/AMV9Ff8Ezf+CZ/hf9jPwzH4y8YRQ6p8QtUtwLy+wGjsIyMm3tsjjPRpOrnnAwo
qjjNrmvTR4JN9OWZ1LD7wTeTxLbQ8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K
qjjNrmvTR4JN9OWZ1LD7wTeTxLbQ8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K8K
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
41+G/gX8IfGvxD8bXt1D4Y0a1E7RxDc88xwkcMaZG6WRtqgd8knpmvS4XzCtRrKlTV3LReV+pweIWF
H3T2OBj/AD713HxS+KHjH4y/Ejxj8V/H13JL4w1y9a8uYlkJSxQn5LWPH/LOJQI8dDhsj5iK4NkGAS
yB1OKVXC55B/Gq7PxzgUzevrXpH1VPY0BOoHXFSRXrwyRyxPskU5BBxtPt/P8KzNy+tJ5ijHNAm4x1
pqv8L/iFYx+J+PM8P6n/AKLqcfAJP2d8M6Zyu9cgkHtzX8CfgTW7bT9VFvrMRu/Dd2psdUtuSLq0lU
QscIMkn5fP6GEUeeq7SZ34WftXaB/WusTNtcHcGyVI9Px/pmpkgk3Lwa/nJ8N/tof8FH/wBgS1gsf2
LBqUK4+ZlhJ/exg5/eRll9SCcV808vT1pPmRc8ZKm+WUbH14InC9M4qjPhMiQkAEZyRyT0A/+vXIfC
gXXvBXwuDfYNa8d3C+S0cbsA6G7I22+UBPlwbp++UGTVLKqm8nYUcY3sj//V/Uv/AILO/G/4X654g+
/wAGT+Ote8ffEfVvCI1LxBqkz+RaTNPqDL5dvCPmyfsMxZ5WZ3JBOMmvjLzov4SFXsPQV1ZDRjDDKJ
8TfDzv4X8SMpyJb22AUXDZ5/fRmKXjAyzelfxZmaNACGfcP7hw2fb3r9dv2F/wBrTU/hV4i8R6bNdT
G+Divg3RiRkPeLiW+nX3LG3iJ5yIiOM8ffcIwhBz179q+KP2O/hpD8N/AFppivHLcuWvbyYA7rq7mY
OgyM/e6V/Nz+3DoYs9X+HniPTre2bXdQW48Mwjdt8+SXLwqeDk+Yz8+jiv6cv+CzVmbj9i28163iMs
OaZRispqK8ZSjJ373W33Hx9TjXFcNZpg87wj96HtF96s/lqj12Hw/D8MdQ+H9rZKiWNnY6XCpRdikP
A4z7g1/W8KcYQjTjskkvRH8t47ETr1JVKju5Nt/N3/M//9b72+IWnHTPFLuAvkXS+cMDjPQ/qDXNxy
j8TX2Bo19BrOmWN/bM0omj3jJ+Ykdc+4IINfH9pbyMwWNwXHI46V7d8L9eFvqGp+HZDKFmgN9ayM3E
tTS9D+cf9rTwfLoeq6BrTrCbS/GpabP5aYXzLe9mUADqMpKvP/66/DPxhoL2nirV7FEZI0uHGSOg4C
nA7qMV/NGfUuTGVF5n6zhqd4RZ8wS6WfLl0y1svtMssYdy6/LHGeA7Zz8pJIC9Wx2Ga9q/Z/0DRze6
xW8KTOrrBcOYM5AEYBD4HsAG9697grHKljoWe+h7XD+JhDEpdz//1/5x7S0XR/FfinTJVMQjulfYec
nGCFw+PrgtX7y/8ABCH9mb/hJfiF45/ah8RaMJtL8P27+H/DTyrw2pTDbdSrxwUhZUVgRgytX61xPl
m8a+DtL1S+ZxrcZNhqKP96K6gLRygj3dWPbgr6ivj/SNSmsLu1vIWKzRSB1I9Qa+h9BvrXw78XGjhi
G6ZJMOGPzDcuOcggHOfxrn5ME734PSv5Ex0pU6jPv47nRIyzpJHKfNikjaN1bGACpHA6HOec5r4P+I
AZ/LrnI54r7jgvj+tl1VVIS0Stbuc2MwUakXZan5zLH4e+Inha4g8yO/8O6hFjdAxVk3DcpUnlGBUE
CFzKZ/Eegw4YRSuCz39qv99gS8sHSQgSDD5Lec2HjHTPCvinwJ8cdB1SLUPCTomn67NFcloJdJl2m0
M43KcgjnuKzXjCsOua/jXMsPyykmtj9DoVbpNbELHcc4xTacwAOBTa8I7AooooAKKKKACmszKMqzKf
1/umrlrubJlengseMnmm1PCvzfMOMVjB62Buxat4x8oIB7mt2CHglYy7Y4wcc1nWyqFBOAD7Vg/Evx
MCVju9UY+bbWGVPJjUR3DJk53wA/fwPZ9euGnujCDuWM4PP8WACB7Ag/r7Unwe8Bj4c+CNM0C+Npf6
MHi6/wDA/gqW70JoLnxnqNxDo/h60llMYutTnby4FLL8wQMS7EdFRskCv5uP+Cm2m6f4f+LSfDy2ul
6DzXHNfy6ft2+I7nxj8Yv2stetbh5pZtcudMtT52dqQpDAFUHkYMTDIz1r5XxAzWTm8JH4dPzP6n+i
8Oan4r8U3n2DSrRVYsEaR53P3Y4o1BaSRiAAigkkgDrXiHw5vfCv7P8A+z58IvDN5bJH4dPzP6n+Fp
RWpp6v7j8H4mzl4nG1sS73nKXXzv+pz/AIY8MeL/AIva7pPxL+NGkSeH9KsSeH9KsSeH9KsSeH9KsR
z3k5uJ2GMnjaPTHav3LKss5IqnTjaB8XXfM7yLniLxTeauZ2B4iU9Px715xf3yKjMpGanv8AUI
/ojvr+OJS7N5a+pGa821nUtY1llsdHhe0tG4e6YDK/QGuxmUF134fBBxXK+INd0nQ4PtGoTtCSx2pH
IrOyjUElzI2Fx1J57Dpya8q+J3xl0TQLiLT/ABLdxp/hSeTbeXcp/hSeTbeXcp/hSeTbeXcp/hSeTbeX
K1Dxle2pE98mTgafDJgmEkMBckBDg7dxB25nhzwx4f8ADFkbDR9PhjiZ2llkf97JdSsdzSSyHJkcsS
Z492yLGed5CDHfdX7zf8EyPgfffAf9jr4o/tJ/Dr4Z+HfiJ4Z+IXhn4heGfiF4Z+IXhn4heGfiF4Z
4r+YvGeXwVfHf9jr4O6B4htJrPx5q9xqfhSeTbeXcp/hSeTbeXcp/hSeTbeXcp/hSeTbeXcp/hSeTbeX
unnt6V+dHxs5fiz8fiyPgfffAf9jr4O6B4htJrPx5q9xqfhSeTbeXcp/hSeTbeXcp/hSeTbeXcp/hSeT
xyTqSP6o4ZymnSUYQ36nVarOJbmeQuWXcenGckfpx0rIUKu5lBAxTZ/MMjHnFV7h2WJNoJ5AwPvf8P
qAyks4P8UKjzEjJyrfM4JUjP0z/wVg/b8HhKy139lz4L699o8TXUJTxZq1rKAulwMDvsEcZxPIMll5
WO4iVigYuxwHPGTn1JOeeeua3PMSRA6OkkR6EHOa+CNP1vVLJm8u7kkQ84c5rvtL+JOpaesSuWCjrt
2O1mY4cZ7cVR3t61lDVraYMyXcLL/vUv9o2Y4e6iU/71I6oTstTfhbK8kZzTW6LWSmp2KKG+1xsR7l
P8TQmwY46gZr+T7VfHunW0REE4OQeQ3T3+lfa3/BOj9HvF2ieHNJ8Ra34i1vxFrfiLW/EWt+Itb8Ra34
i1vxFrfiLW/EWt+Itb8Ra34i1vxFrfiLW/EWt+Itb8Ra34i1vxFrfiLW/EWt+Itb8Ra34i1vxFrfiLW/E
39jT4ifFDQbD9hj4V6h4m/a7l15NQs/EPw6IkeCK3KpLIpJL3cDRRxncXmyNp9n0Dw1/w
tbfxDdQRRatqc+ZNQ1NgB/x8XLje6jsmFVTwFFfmWDx1PBNwg/fT+Vz7TM8ApRjOqtJbH//W4T4p/0
emfsteIvB9p4dDOaYSRA6OkkR6EHOa+CNP1vVLJm8u7kkQ84c5rvtL+JOpaesSuWCjrt
nRvY/S6eF5NII/GX/gV4dDOaYSRA6OkkR6EHOa+CNP1vVLJm8u7kkQ84c5rvtL+JOpaesSuWCjrt
5YVxhpEkD/oP8a/RMBD2dJQWqPBxOItL3tz7b1DxPaxRvCrKxIAXI6HPH617J8Cor/4k+HvF
IdT066tsq24MVTzAB6f6tjWlSkqn7uWzLwGJcasakd0z+2H9lPx9p3xW+Bvw/4k+HvF
0qcNkz9gEzJcbe5VPX49Glymqlc//1+PBvw/4k+HvF
3g20o7AA7Rur4d8eaQraz4msrGGMQSzym2G0beWOMDtg/wBK+myer7PFU6nZn02Dny14z7M91/ah8I
ZfmupCR15ZivphR2r+L/AOMvwL+Jnhn4heGfiF4Z+IXhn4heGfiF4Z+IXhn4heGfiF4Z+IXhn4heGfiF
HutZ8G6rFrEKqxDfZSxSVeMceXLLweMzrLTbh3tomJaRk59BtP5EL+Zr2n4R/8FXP2ovgZ4Z+IXhn4heGfiF4Z+IXhn4heGfiF4Z
WuR+C+r3N54Ch0L4ZfGjSJPD+lWJPD+lWJPD+lW
```

```
4ZaT8Mp9W1mPTEX4Da+8lrrVg0TsmgXlzIQ8sG75YrWRmbcgCrHK7MMbmr7nhlBVuvPHWq2oaZp+q6
8tbvxd4Vu38NajKkodroQ/6i5f8A25bdreRvVnb3r3KZ2O4+9fn78Dotb+CX7Ufif4J63cazqljrWq
6Clr8yPVCiiigAooooA//9L+2miiiv4YP08KKKKACmv9006igCvtbuDV+JSDgfSq5BIOKv20T5XHJP
8LeHtZ8Q6uypYWNrJdTgtj5UUsR9eK4f4FaBPoXw40XVNWtzD4l8RXD+ItXD8kTTsZNpJ7qjRxj/AC
pdz+Cqa9CV3klmmk++x3H6mvC/iVZv4o8ReF/B0kDXelXt3Et5EDjfAn76Qe+4RhCPR69uhyUIurN2
7r/if4keZHaWcckzFG1VjKwQAfII1L5Y4PQlcZr+uj4z/G7SvCl5afDzwzp83jf4t6zFt0vRLF9siP
w/Wx+JeLrq0ZO/yP2zw08TKHDWHxbablWhyfe1qe1+Bvhbftq7/Ev4ozS3niZkaOy08yBrXRoQ3yqc
rWqqTk/NlC+uNkjrlfUn1riLu92tIdw57mr+pXrGVgHBrhdRujkruGcHNfZYTCNKzPExFeyKt7e5D5
FHrjv9Kxta8S2OnQT3QvLSOGNGLzzSqkURAySzMR8oA7V8ZeK/2i59fupdL+FWm23xA1HeB/aV0Ht9
8U+PLi+uPh0b/wAN6O5kSTxVqtoPt9433Sun2EmVhjZTxPL8y8ERvw1c9L4ev9Xv4vEPj/XLjxr4jj
Ot57jV78hr3Urt/MvLxsY/eykktwB6KOyryKq3muaRbX1jpElwk2rTsTFaxyfvZUGQxx1VQQMueF/C
TXDEjdk8AKCBtjXaqDAAPU3zo0nKMXyl6MttDTunmt8z7RgAnkgCui0fw9rWuzCLSdNnusYDMBhUz0
x7HsK5cZWhCm51GlFLdmuGj7SrGnHqze/wCCVnwon1L9oD9pX48a1AL+LS5LbwRosqwkDKATXbRu+A
gD6V/D2eY2WJxlSvJ3uz9VrytJxWy0+4/9X+7mIYOPaom6GpogRyeaic7DzXwJ9bF6mHqt0trbTSuw
jACk56cV8FxBj3Ul7GPU/VuCcscF7eR8Aft7nVfiH4Fu/wBnTw9rM2gXfiaxnm1K+ik2ta2UQG1SBy
frt0I9DSVQfs+jwnEBbgEiVt82PSRR/DXlvwe1zTvhx+1B4o8LfbLOw0/wCImlvq1lblgvm6tYbEuA
UH0b9nbRL74PfC3WYdT+PGowhJXttrL4UtnJAnn52+cwD+XH1zl2woXd9J/t7/th6b+yP8ErnxLaxW
g8KpCjI6fD+GnAjx03i8Qv3a2ueh4r+IksrX1bCu9WS+45nWbzUNR1G71LULu5v7y5ke4mmmctJLIz
EkVM/U0IeSK9I9WGxejkYZOBnpV2G7lDfKSv0OKylJBAzxmrUbBTk0yjaXUbhOk0w/7aGpV1e7HKzS
d/a+LZbWzu54L65sZYLaWJiGjuFQyIykEHO+NDwQfeuE8r/Z/Wtrw3qJ8P8AiTQtcVcm1uopj9Awz+
MpGCXxX9Mvwj/aa+C37SXw10/wCIPwW+IPhIPh/xt4enQGR7WceZanptmhbEkTjoVcD8a/J7/gma+h/FP
D/sqazZqb/UPDd/qRj0K4QA/LDIxKrubH7i4SaJtwVRHxX5NiauFq1qlFpxqXsux+41cPN+znKXNTS
RAhYQqCeCXKKMn8SfGH/AAW8/ak+PukfD79mG1fwZ8DvGuqahFoviLxvbTtbWxDgIZZ2cFrCIFtzeU
YkkepgcnlhKsVjPefQ8urUjLmlHZI/j31zxLrnjfWdV8Y+KLp73xFqtxJqN7KwAMk0rF2OB05Y1jCA
sKFlJjP6PXzQqY5CmvT/AIONJH8WPhe8RCyL4g09lOOh+0JWlH+Ii6FW00vM/uB/4Io+KrzVvAP7Wr
sXUj2Z+5U8O6dON+qR//0P7F/sou7Wa2lTfE6MjA91PUV/NR8FdEuLew8K6BcwT2sulXuoaMbdodrF
eaOn0kj9Q49pc+Bfkd/eWokgnjYEbo2H6GvN/g/D5OiahEcAi5Ix9AK9gljG1vlGNp6fSuB+HtibWz
UYZ9x/8AXqcRPS56mX7WItOikQRiIbpUGQO5r+aD/goKsGrftd/GieKUSR2b6VYSBSMQpDYQnYc/7
t6lfG7lMW+CxDCNEHkwNvO3CABpN7YA4XOB+VcbY2nKn7CO59jgsN1Ph630jVtXhP/CLeHbrXZ1JSC
4Z37Ii+p9K+IvGuqaz4rvpb2cCLdxHCDkwoOig9N3qe/avyV2Wh7tOi7aH//0f5hLO+uNG1nTNZsJP
APWpuh2Muq6pY6ncWvzxOrRCNW3xRHleQQwzgnk4HzDFevDY+lpxsrH6jeJNVj0LSby6nfY20qoPcn
aldXR1HVp15ml+7kjKhQACqqSPfg+tddKdnF9mTKaTufqFqnh/QdW/ZL1WHwzbSJoGnXFnqNhDPL5k
7VrO3gvLHwxLZWJgbHnQQh/KL8H5wqhM9hgnpX7o/wDBvh8TW1f4KfGz4VXRAufD/iSG9hXOT5F3Dn
/ikgDM4ZXUjPY+wNfNNjrZg+MXiHw5K2xZ9JivhzyZI5tuPf5X/Svojwyxkuaw2E4IG4Z/qK+Ox0E4
dC/8PEyD/WBoxcIMKMDBjx17e9fQF2hA+XoMKfqABx6DGOK/i7xMwCp15Ox+t5FiL0ox7H//0v7Z3E
DndO6Ph/9tC9074d23wW+PT20klz4Q8ZWcMjohZ3069ZrS69zhZYpMHqYRX2iTFPDFJayRzQMisjK2
5ltVYsMAllgRumfmr9ExmF9rk1PES1fM1f7jzMPUtiOTyuezqSAAwwf50+kfBZcfeFLX57Vikz32FP
1P7LPjyketeHfCvw789oZPE+v2eihUbBe33NNc5HTb5MDj8a91upFkmVI4/JtolEUaKflAUYGBXhWs
ZlVY0Lkk4HHvXyL4l8ceM4fiAmg/DrRo9U8aT2bFLi+jYafo8TuQ1xdSL8xVdnESEPIflBHUfVevx3
ETU/3mt+IL8K1xfueqIoAWG3UEokEeFRQMEkFj0Op38sgQsEQj7qr0Wt3UrppAxZzn0zXFXc3yhjj1
4Leprh7+8G1sMB2z6Gp9X1eCOOR5JMbQSeQD9B718s/En486H4O1U+GbGyvfE3juRC6aJZBXnijPO
L8UPFkLiJ47aVRYWMoOCtxdEGNSOuz534PyjrXjHji88Z/ExLi3+IPiIaL4VnlVj4d0G7aNLiIZ/d3
zGp6X4r8daida+NevR+JYYy32Hw1ZxmHRtP4+9KhPm3sh6ZmbYBnai5rrXms4LeK2sY0jRIQqpGoAA
iP4yeLG8AfDmSNXt9J015LSS/XuTMNtxOp6YQInPWuic1DQ6JQjFankWq/FWy/t6bwr4L0rVPiN4z+j
9DPCfgT4XDfRU8OfwfpOhWEAVFDWyws2By2yPaTn1LHJ7cmteZvNXZwvzFshVHXHZQAOnp9c10u
0Dbz/SpwjLx1rGU0jkdWTd2QpGVIxjFcVo/w9l+Of7Svwi+Fps+OPEOqeIbxQ+sXkaxXMsYZ2iTFPD
Q/K1/UDc5Etrp+zFjAR1QJBtyoxhic9K/KfFTPlQwP1em7yn08j7DhPBSnN4ia0W3qf/1v7k4IxH1O
/sqLqHoZPl8sTWjTR4J4y1h9X1m4kaTzFQlV9ua/MX9u/wAbXutx/D39mMGh
Yuw0SykeISrnos04mm4GGQp1rxPDvJ5ZtmiVrqOrP3HOsVDLcslKO8VZep1uqyw6NpNnptmwj+yQLC
wjMeM5wPSvxb/br/AGrEGevCmpfHD4y2vQJqlFi0AwdOhTVKkrRR25xmtXC
y/enLFk9aZZWSOQH5kK/jU8ferKoQOS350vlj3p2Ahoqbyx6mjyx6miwH//Q/mfppAIIwCOhHtU/lj
wqYwYmgJYnJCgDPSvS/hh8fAf2mfpwA+OPEOqeIbxQ+sXkaxXMsYZ2iTFPDFJayRzQMisjK2Kl1C
Av7Vd5qvgXxXEpep/Cb9Hqzg5HZsnux/O+NJyopSwkFz97I/W+Eqca
o/7OOk/HXxx8Sfg/p2mrruj3p2Ahoqbyx6miwH//Q/mfppAIIwCOhHtU/lj
pSNSPZq+T4ExWKq4turdxSe/fQ9bjuphKWXRqYZcs5O1urWt3+R8vpCuwEBVGM9KsfZ+ASTT/MPoKl
q3jS4Kxr4f8ADus62pIyFeGykMZI/wB8+r/VDlyyyv6HVkKlKlRROoxwqmXmtxHXkpwa/MP9u/xta63H
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
fqN4b8Qi6s4FkffKBj6V/NePzd18yryvePM7H9nZlky+p0ZW2iez2785xzX5CeJ7OLRv21/2s9OS3i
Xh64GMHODX6x4UY9080hKL1Px/ifAurgqkVvZ/gYV1GqeZEFYAKQKwfD9gbW3+bIVndse5Oa7jUbYe
Xi3xSgMfiGB2RgjWyDOPdh9D9M0sRqtdjpwt+ay3PwN/b28P65oPx8ufE3iDUhD4X1vSbCHRpLhtkM
1nSvht4T8Bw/Ej4h3biW00iSIFLeP7purl2BWG3QnlzySoC7jxX2T4J/YZ/4UX4lk+JfxB1S1+IPxX
+JbeLl8UXGk+K9E1vwne237yHT9RtJLaa2QgkSNHIATnBG48Z4B6V54+lNdfIVZwflxnG6v7fv8Agc
H+teJwtxKs0pOqo8rXQ/VuKeGFllWMYy5oyvb5W/zPELr4dz6lGt7qQSO1jIhit1IVrm4YrtHH8KbW
saasfIzfvWWr8j4o/aF+Hz+Am8I62sDQ6ZrUkjWwU/Kdj7Gx7bu1eDqvmYiAxnCgnt2FfoV+35aWur
48xtivuI7kqJFx0O6v0V/wCCPHxRh+BH/BRj4gfB29uQmjeMrXUPDgDcBb20lkurU4/2kS6j/wCBgV
1wTX77wfN1sp5Fumfb4fDfWsmnSlqf2v31zHD+01ogdJHM+h3iBgv3NpgPPpyT9a+y/CEPnsrEL5R+
5/PHLKnPle5//9P+vTxVfXWlaJ4a1W2+zhNP8S6TcyebLsASSZbZ9p7DE4JA64xX1Hcq7QKC5diAdx
tzo4WqttpnM3MZIbkVlMMMRWzdE/MPpWSwzIR71/KeYL94fdU9xqpk1ej6ZBJ571WEe0DnOaspkRtj
iPWdRERcu22W9lbcScdeuMd/evhH9urxBP4h+J3wm+GWnyrPKJbeRoUb5jcXeoQQQKcjABWO5JzyVV
T5cBjjpmm+tfjc3qz6aGx//U/tpooor+GD9PCiiigAooooAKKKKACo5Pu1Hub1NKCSQrZIqY1FcZ/9
I1jnuMg45ybvPHTAzXvaSBwMnmvn34XNbT/Ef9omeC4kmlHiSytZAy4MZXSrVgo9sN1/2jXushESM2
atYoajcFsoMDI61xeo3IiDRnnjk+lbOo3SA43cgVwms3yIkpDc4r6HD0jCuznNYvQ0eFIXsMnrXz38
yEHLYzlVGT8e+IbeGDWLnxLqF22teJZI2K3szkmFOQfKQDEIGOQgORjLE19jlmEagqk1p+J5NaXNdI
zLklpbhixaRic5eQnPGfStzRk+Ifxl8R/wDCL/ArwNqnxZ8RLP5FxeQTi20fTADjN5qb5SFRz+7jWW
edh8uqt3ktD8qPBnhz4kfGPVb+y+D/w+1fxvZWsbSap4mnkSy8PaIkYJdrvUJDhmAUAxwLNIufmVRy
J4dItjExhd1FrpcRjxsVwTJIF58vauMSvXHeC/h9B8GdKNk9nZnx/cIrX10OulxEZEEQ4w/I3P8Aw8
a9c1LW9S1i8lutSu5buRjwGYlIx/dReir7AAVglnLSMxbe53Nk5JPue596lMoiUyPt2Lyc8V9rKKer
xJyBgKVz9OGJycD5eKSKH5c/ezxisTx94z074b+D9V8X6pG18LcAWtkmTLqNy3yRW0ajlpJZJIkUe7
uBm6EE/ubptpHawwWccMUEEKhI0jGFiAzgL7DNfNH7IHwGb4HfDWa78UQWtx8XfFU/8AwkHjDUFUFT
k8YFQyMQxJI218lSpu12aLeyM7Ub9bS2nuJSoRFJPv7V8e+NtbfVtUnwxZc+vA9q9k+JHiPyLWazhf
5471FPDlnNCwMtlbvzcTjjKhIlds99vtXjunaVZeGrnwB8OfDIit9EsLMWkVsF27LWGMRrGc8AqCgJ
A7KVM1wTMQNpJztWEHOCN2K/prwF4Y+r4KOLcbObPlvFriFTqLBReiWq8zktfxa6tqdtu+WOZkTIxk
2o2V3YXMcckE0bxMGGQVZSpB/Amv6RxeFhKjKm1dWeh+J4Cs3JPqf//Q/Fz496EGtrDXbWEFFCyF9w
r38ZSUK0oroz0pPUTzn9qf5ygfORVeopf4a5GKK1L5UMSRg1mOMMRV1G+QZbBxULAEngGpSLgQKMkA
Ln8Kl8r2WlCEYxipm09yXLsfSX7MHjq/+HnxN07XNMvJodSgjW/sthOftNpIt3GBjqWMBUDB5POM5r
9aik8lrW6jl3AZ4DAYPsc4PXg+nI/rG/4JVfsE/Bv4ufDrwP+1J8cNW1f44+P7SP+zNI0vW5mu9N8M
+s+E/hk+li40b7Lo9xDZ+OJFlVftEhVQsm1RI3mH90jRqAu50avzrSMRxLGI0hUAAKqgfTgd6/oW/w
asI8qnq0fOcaU508d7KpK7gQGpVRcjjvVjy0/uilwPQV9PCPQ+Wi7n//0v5sI1G9cDB6Cvpr9kzwDe
NeW7zaX4V8Nyx2rtwFu7xhEqj0cQxzZ9pK4+KsasNgK1e+qWh+icC5XLF5tQpr4b6n9ZUUVpbeVFZp
Py+Loun2PrXTbsMEAbOa+I/2ydIitPjd+yV42hVHnuLnW/DtzJ5IdhBNZCYBmPOzdDyOe3pX1F4b1h
95g6naBbm4yBgE9qw5LYKScZNd9ex+ajSDAZgSDjOD9K8/1zV9P0KKF9RufLaRvLjjRDLLcPjO2OFf
DiaPQ/ClhEseqeKLxMWsDGQhobNMkXc+M5IxHFgb3JYRn6j+EX7I/iX4jx2Xib4yQal4M8GyoJbTwz
DYTbufT5RknsZKpV3P/T/tF+A37PHw3+AXhqXSfBukFNUu9smqajcnzLzVJQMK1xKRltoJCoPlQHac
41+ONNtzAbTwtrG5xnIY2MwUgjkHcVr/AD2YLh/sMMRB+WMLj04r/S7/AG5vg+Pj/wDdss/G/4TW7LH
KpUirR5raelz6LE4+tiMFSdTVxbu/Wx6P4OtU+wWyAIXK5G5goJ643HgfU9Kj8Q/FTwL4Q4mey1XV2N
rMPuGuJ/4KT/AA28MeEv2l/i56K8F6Vp2h6Lp8OkCxtrOLbFFGNNtwADgE5wSQcfwnku1fS8X5hVy3
ecI00o3KSAwMgDDOQQQcHgfHlrbvPdWkLAliEUgfQZ/rVTSbeSbxVc6fHGYLWCF76bgAJK3ygHH8RD
K1mVLrStV0/Uol80AsIblGZSMdGUMp+tfSH/AAUx8Ir+4fxKXgrXUybXUdBjkLKOJPLd1AHv+8j/AV8
hZ0zt4OanhqkOx//1PXP+CPHxbPxi8O/BqQ+Lv7Lg8Ka5qOqeIrSDTZ1xpfqfhDw1qQXa1xp
55Jr+heK6KhFJHlcV4NUcyqRSsjP+Me2b4S/E1Vd0mHh7UXQrw2RbuePfivqLwtfpqfhDw1qQXa1xp
veLlJexp/M9jhJXnLyJbn+Lv0NZsnUsOpNat10/CqSRbvmIGM5r+RcZTvVPvoysNGWKirGHijZiMj+
V3Nq337/AHXuc2PxKp0nN7I+Hfhre237QH7XOnfEBru4k0OfVr3XdJiAJRtL06NrO0dj2Esssss4J5J
PH/CME5/yK+18xsfT/tH6nh3+7oRUV6rc4smoPkc+5//1f7bJVG7PfNMobcWwwPrRX8NSP05bBRRRU
klmYgCNQMljnjHFQWsKEfMwwBuOew5/wrxzxQ+IP+PfsxQ+IPIPFfy4vbPRxSfT/tH6nh3+7oRUV6rc4smoPkc+5//1f7bJVG7PfNMobcWwwPrRX8NSP05bBRRRU
jYjIwPfrTdSvZYkWMOjs+Q3GNoAAGAOB0rmXuWTe4PyqMnjoK/rDA4T2UYx6JWPlZQTm33J7ifYzgF
A+UNwDsXj65r6DB0VKSXc6YRsaGqat5Qcq6u3T6GvnXxl4hvbyK5s7Gaa1gdmW4mRwrOP7kWASDkjI
8LryJgl39mQeJr08bWhWUOllGw4LurTBQNvlklq9bFYzDZek8ZJRFKk5bH//1/2U+Jfxu8A+EPEPh
u6eAf+EPEvh7xHqOqeIrSDTZ1xpfqfhDw1qQXa1xp/2s3/HxLJ95i7
/d/EvwVnw7gcHmVKp7WliFo0rJS3tfW+nX5Hpfhfwj4J8Y+Ffh/4Y8Q+Ffh/4Y8Q+Ffh/4Y8Q
```

t+WZwQCQBnGK8Q+IviFAxtYnIXJBx3NebmmIVGk3c9vJcG6tZRaueP8Ai3VvtDyqzM/zE49+P/r18N
P8d9G02OS3m8B+EmGpXAVNyy3qtItsgLZBG9nuDnOPJt+xOfguHcFPMsyp4aKveSv6H9Bxq08BgZYi
pHySbee9u1siA3ykfZoYTjj1zXmPxT1W+s/C/iJ9LUzapPbfY7CNThmnlIijTqM/PIqjpxgda+7fif
+M2nyTaRYapHEgMJZJec4Vhjg/VfTvXza7Dy5B6jFfcOv6RHrmhavpU7iRpIC6t1y4ywH5gV8OXETx
xTqG7b9nmP2iEDjbu6g+vNfvL+3Z4ZjTQvhz8SlEka6bqcuj3zAjAtr0Rqhf/ZWeG2H1dvw/GX40aU
GT6moKlDKehBqTSMu47J9TU8TfP8xOKgFLQZV5dD/9L+bWJ1HerG8e9ZUBOWGeKs5Pqa9E7aexeDA9
PC+q6hqtxdySZWC1a2WZigOCFCxSH/gJFfygSEFG3FMHA+YZBGecjuK+jfAf7Svin4e/s7/Gv9n3TP
8TfjJrOYbvxDqs1+kBOTaQEgQwn/AHI1jX6qa8pjVQMcVG8jzM8spHmMeeAM/gOMVJGVVcZxX0eEwy
a38UdUsjban438Q3WpQEoAfsUB+zxc56EpIRiv5VdE8Mav4y1rw/4P0GLz9c1q8ttLsUzjzJp5FjQP
+eCGUe0xcsS18On6no7jdggljTDvJHGPwqcqCRxTSicgDJ25wOuK/nCjDmipH9TTp3Om8L6pc2VwLc
tbeICS4vJ2+5FAhzl2PRQOT7ZI9R+H37OXin446fo3iP9pTSdLh0BSJrDwTZyNLYREf6ufUSf+Pmfa
ISzY5ZB8w/Sn4I/sw+Dfhfc2HibVlk8a/EsRnztbvoVXyWYcpZwjKwRAEgKMtg8u3U+veC/Den6HbW
SgcDFODBQ2Sea+OjK7PclN3K7ADoMCqt25EZHfrVp3UMMntWfcMFG4HjpUTnY2oq7uzK1OAXMGwqCS
K7tbGxt4pJ5p5nCRwxopZmZjwAApOT6H0r8hfipoWl/tKeKPA3xy8R6IZvBNrr2kab4BsLiIq15cT3
EnX9W1HU73xvrt9HZ6lZXdugfQ/sj3CJbB4ztfb5m58BRu5AwRX4o/8ABWPUbLwt+118Q2FpBd6xf2
KP2N/wBp3xv+0z45+L+g+CtU+IPgu/htrbS5tLHmvp1lBCirbPATvVhIbiQsCd/mjptFf0PxJwtRzj
sYhZ3EeXbuB1r6d8MfsZftUeM777H4e+Bfj6SVJBFJJeWLWaQ5OMmSYKABznBzX6Z/AX/gix458RXc
s//V/Dv4eaH4h8aeKdL8KeDdF1XxH4ovWEFpZ6fC008rE/wxpgnaASTkYXJJAHP6TftfaT4mvv8Agn
ETsqiZy0plaU79yncSBjpX4pePPDll+0h8PPGvwk8Pa4pR9I0630edHEsFxPahMDcOqko/zA9T2r+I
AE0Pg3458I/FD4W+C/iF4B17TvEXhDVrCC80+7tvuTwOpII+hXYQed6yf3RX+X1pfwj8czeN9Z8Dwa
5UlIefa3/PQMAu4g8nHsLwi1umfN+IeXcl6s7czZ+kHxwmNt8GfigwR5idCuovLQEtJvTZtGOhO4V9
qBj+lfy14vO8KcFufM8Jxacn3OWn5f2xTUQ5ByAM9K0DaGQqSFpRbpGCZSsSgE5Y449vWv5kp5dUqV
eANHmiKjT7aWX5J5xz++mJkupG/hjihj52MT9ofFT4hzfG5NZTQriS3+AehTuL6+j5/4THUEJ/0aPg
XD2W1M6xytUatC+jUmfK4zFrGVfq8Nl1P//W/tK+Gfw70j4VfDzwZ8PPD7ltI0fTLewjycvMyIA8rr
NW0CiiikIKKKKACiiigD/9f+2miiiv4YP08YMscA4FatrC33fmJwO1UIIyzg7sE4retUIOcjpXq5LQ
M1v5oHkWxOAcRRbE4xyG9ad8VIbHxv4w+FnwgkuysF9qI1/WLdWwJtMsWSQxv/svO1qCP4lVxXo+va
22dsUY/vcbi3RV+Y8V1HifxDpnhbSr/W9YuorLS7aBriaUsAERR8xP4fn9cZ+edBtrjxFe2fxO8V2X
1oD0QE/NKfmdwSSAFROF8WfEqVddi+HPgLTLvxT8QJlV5LCBjGLCBiSJ7l8ExRksGI+9IM9Qc10Fmf
yf7Tf6rcHlprm5Yl5XJA5JwBwAAAB8xxhxzhcqXLFpzf4HRCjc//0P63Pg9+yxa6LrFt8Tvi3LZ+OP
cvJvQ+lWE+0fzz/8FRvgRN8Ofjto/wAY9Gs1j8AeNFNrqzjaBa65CkarIw2/8t4IyM5zvgyOTXX/AP
b/a9HuWAP2PUoj5kEw9MOq5+pr+U/TtQ8W+AfEdhrWkzTaH8S/C+TwIDfTsI9fTLewjycvMyIA8rr
bv4m/HwJ5V342v7m60uUKd6aU0ghs8g9c20MbY9JCK+/tM/aQ8O/GT9ifXPj74QY2UV94RvpJLWX/X
csNWVpwbTXmjxScKjD5VUdOFA/l2zn8Ki+8MjpXQ+J9Kk0m6lWYGOIt8lc2jZG3Ff1LCfMlJH5piG0
Br1n9nu2srv9pP4WSXh2T29hrlxbIeRK32eKMkfQSNn0z71w/irT5NB+N37QnhGeCC0W18WNeW+MhW
z0m68Q+KtReHSNA0iBgJdW1GXCxxDOcRjJaSTBVY1cnI6x/ETX9J8M+EtW8RazOLTT7UJPIzLtzubl
9PQ8zh/LoVf3lX4V+J7j+wN+yRP8B9A8R/A8R+If2gvDi9Ta6naysssEn90ZHkt9Tj
bv4m/HwJ5V342v7m60uUKd6aU0ghs8g9c20MbY9JCK+/tM/aQ8O/GT9ifXPj74QY2UV94RvpJLWX/X
csNWVpwbTXmjxScKjD5VUdOFA/l2zn8Ki+8MjpXQ+J9Kk0m6lWYGOIt8lc2jZG3Ff1LCfMlJH5piG0
Br1n9nu2srv9pP4WSXh2T29hrlxbIeRK32eKMkfQSNn0z71w/irT5NB+N37QnhGeCC0W18WNeW+MhW
z0m68Q+KtReHSNA0iBgJdW1GXCxxDOcRjJaSTBVY1cnI6x/ETX9J8M+EtW8RazOLTT7UJPIzLtzubl
9PQ8zh/LoVf3lX4V+J7j+wN+yRP8B9A8R/Ff4tX9aTuQc7utDybAc8U8Z74pjqXUrkda9BRsdivdII5Q/1pvaTOaWOLYScg0kkscUcjyyLFCq
0+H3j2xaKS01nRLLU4mU7lKywI/B9Pmr+aT/AIJnfA+fsS/4CNaJb6t8QpHvYo5lIeLSRG0Nc
AcyhP6xQXxRs/v0/Q+6Bjkk/lXO+JtZR21vE0kkrYJIUdF+6gLk4GQDwxB6MdCOP
DTsN3zOowhIP6cQ2awxx29spSBT8uSSfxJ5NeV/BzRv7C8HaVZDMY8vc3+0SSf617TCo2naMHqa/bu
h90DNQyMOvNVpLkbRyRz61Ue775+nNdcXY44QZYk5Q5GR7Vi3d8io6lgqjPJPHQE8/j/AJzRc3ewON
2xpqEXOT0Wp02u6g37ROu33h+0kkm+BemXyrqcwyI/E99BKp+yq4xuso5FQS9RK2YySu5WyPinZL4h
j2KgXpgD5c9SpIPJzXlHhO0TxJ+094+ixrX2uJwSwGCqC
D+or3P8Abd0KK1+MP7NXjGF4Fmns/EugS/OdzCSG0u1Uj0/4lzH/AIFXjBgwAwOBnHufQAep9K/Z/D
PcDgH2zW5b2wjbnbnHIz0+tf0/Uqtrlvof JzbvqW44/Mt5rRywV4nhGCK/LH
4e8JwbjqHiDi9Xs/sHgWxyfMoI5GemRyM5GCK/L
4e8JwbjqHiDxZ6vYvV4nhiDsYfMoI5GemRyM5GCK/L
kLo9paIYfncmRj9VxiveWPmRrhQr9BngZ9z2r4fiXGzr125PbsPN80r4uSq1Xdr7jynXYI9b+L3wV
I5bJWGeoI9q90+I/xM8D/Cvw5c4fHBLGs/Hbw
z6p4htJjFc+LY8c2+nyKcxWshyjXQ5kGVjIDbxD4gsPGP7Rd0i/FC3l8KfCNi3k+DDIjXWpZBxJqrc
l7Glsz//1/6ivjh4Wl1vSPh7+zj8NLa10HU9bLW4FpEFGh6NbqguLhF/5Z7Q8SKccu4UV9t+DfBfh
YDrI1I7Bj1tO1Uj0/4lzH/AIFXjBgwAwOBnHufQAep9K/Z/DPcDgH2zW5b2wjbnbnHIz0+tf0/Uqtrlvof

```
oAnhU7Ux1NdHp6Bd8jMeeAK522yTGAvSt+W4ttK0661K/mS2sraGS5mkbgJGg3MT7YU19nwphHUrRh
/htqus6fLputeJYJfFWpQufnjnvne52Mc9UW4EeO2welS/E7xtYfD7wL4j8X3DSn7JDiGM4zNO5EcM
Mu0K9mI0/B2lX37Qd5P/AGXNrWhfBu1uWhudTT91J4olTcHhtWHzJaoxANxkb2LrH8q725b4TfCDVf
xjp2zXynH3HkcvbwlBe9bc9ihC5//9H+4TR9G0Twhoum+HvDek2Oi6HZxLBbWttGEiijGSAFHA6nnv
Rkk+tfIRqOU3dnbGhc6WC4Xch3soHXHavwZ/4Kg/s6L4O8c237RHhiwaDwx4huIbPxNFCoEVpqYTbF
xSmvZydk9D6ngHjnFcNZvRzXDq/I7Nd4t6o/ln/ZY/aF/wCFTWPxt+CXiG9hh+H/AI20u6vLM3TNss
iHE1xqdoxaLUI2AOuabI7C31CLGNr7QBKP4JUbsyk/uv+yp8UY/if8B/gN8S3vG1DxBpVqmg62znDC
p5xVVzuHfH0r41KGMoZVKbuRnvX6lm1triI2sq7rYptcEfeVuK+G/iR4Ck0+7vo7e2Jkhd3iC/xJ1x
+GvF6RW32PxR4eFhcEDk39nIzgnHXMM5Xr/yx9BXiHgzXJfC3xX+DXiJneC2svEttFcfNtKRzh7dsr
e7KNn/wfmeLkmCdfCey7M+K/hX8D5f2wPj7HoOqx+b8A/AtxHdeIt65j1zWyuYtPGPvLCreZKM4G6N
/wCWk0kjO7N3Le1eyoqbsY5r4HibOJ4vESq30Z9fy+6qcPhiNWMHPf60PGoB7VK3y428VE7jYcnmvj
554i1Uyy3EhfOeFHXHvxXjfjbxZpvgzwp4q8TavdRWenabZT3dzI53eVGqsc4H3jgEjHU8Dk13F2ZT
T98lXhgsP7Rr4T//0/08+Fehan4f8H3/AIg8UW1xD408S3s3iDWPMbLQzzbSsH0ijEUYA4AXjrVDVT
6j6t/cfd/wDwS00S5v8A4h/tTeOJXZrG3bw/4XgAHCtHby30v4n+0YQf90DtX0z+1ppBPxg+BusxxX
l3yQNY+NNOtZHA3KILrfbNu/2Q00Zr+fsXn0qfGM6q78vyscM6P+wu5+Y2p2rabqV7p80RglhkaMo3
bbcgnJRm7Z6AZNfzS/t9+E9N+Hv7cX2+B0SPxb4Vju2VQcSzQSbS5baAT98Y6jiv6idVs/PhYllDIQ
yNvHAu7ncRzNJj/d3HH44xVRyCeDmretCJdTnWD/AFQdsd8fOx/r+VUa/niD0PrWf//U/mYoooOcHH
tS7B71CDkZqzQOTsro//1f5u2UKM81Ew3Y5qWToKjFekaKbuRMNpxnNIMnsamKg9c0oQ4O0UjohO7I
uk+EdBO1mGU+3XbBUYk9GCJJjHciu7JsN9YxVOmu9/uCrsfvfpPg2y8K6LpPhHw/bRaVoml28enWkP
H8P6VB8Lb+Lwv8e9MY+ZFZeIdDudNVY8FpL2BvPjQZ6sUM3Xsp717v0hcnWM4eq1La0rS9FomfongO
88MPSaqRi+6P7JzWv/ss5eTP/9b+2Lw9GLezigAQKoA49MV2sCKy4J7Vx2jjEbD3NdnbAleBX5XlMV
dwGMdR3/AEB9ecV8hftD/tFN4Da1+HPw+Fnr/wAYtRt/Ot4GBkg0a16G+vNpUiIHIRMhpHAUcEsrwu
M1SztLC3lWws4JZ7iWRy811cPw0s0pJMjsxdizcsTk+leReEvCNl4Tsb4F7nWtcvbhrvVNRujvutXu
J6rLKsSMCUhO7AHRUGM8+wHP0Brzb9mrTxe+DvEXxKnBjuPFur3GtQ7xtc2e4Q2xYHkBookYf9dB6I
IoVVA9AFA/CvyTj/ADSUaX1Wm9XudGU4dqfPLof/1/6sf28xHD4d/Z81RLMXU1v8RbaORkj5hjm02/
SOVP3iwLdvpj681+l+DNfmyfll9mTX4I5M41r8xxFjc2t/4r8Q2MYIvbFLaKRVZsRh4/MxtA2ZKupJ
0q10eLOCbMTUZPKtbhmwDtrwXwPpNz40+PkU8kRl03w3pjFQDuV726O3OOxWBH6/89K9c8WX8FtDMs
CGmfY9Pidx5khXC84/WtfxTrum+FfC/iHxFrNxHDpmnWc13cyMDjy0UsWx1UfLjHLHsDkCrlhZyR6f
4eWHh34VeFhq3jHW7q58SeJPEt6DFpelajeHzpk35DXk0avHGEj+XEaqzggqPQfCXwuttO1Q+L/GGu
9DJ6OHn7TlvJ9S3i5yShfQ//0P7JoQBIjSK7hcMEXjfjORnovY+nBHUisPwmh8f+NtUjjS1vvA3h5j
2hyPojwn4U0fwH4T0kzR6baRhFaVi0kzdTJIxJLSMSWLE8kk968/xC4lhhqDowfvNE5PlHParY
7HBIKKKKgCdPuin0xPuin1tDYR//0f7aaKKK/hg/TwooooArykjdgkVNDjByM1DMD83FSwdD+NZ63F
8PxZJByI7xJyNpyMYgbPrX7T5nMZvlSPSvENpb6ebHTIB5cNpbRQKgGAMKBgD04xXx78
jFqrfEr4myHzL7XdbljSRZAwFlaf6PF5foN0btgZySfev6ezfNVgMu9unZ9DyMNh7u5//9L+4XQdA
y7ssVbjtzVOPB3dCac+cjNRqSOhr4Oddyep7ECcgHrUQAL9BU1V2+8azlK2pZOrlHAFbtpMWkRWHBw
htAY8K+09BX4FfsFftLW3wE+Kur/AAy+IVrN4d+F3jC7ayuDqQaF/DGuKSnl3EbfccuDE2eNwTqK/r
cxjH7Rwzn1WhWhjMNH3lur7rr9x+2+E/FGCx2ErcE8Qu2DxGtOb/AOXVXo/JPqf/0/7CtKuJ0t2hu
2dzMThpyhVY2flwq8787v3UQt/zzYA8jgjAyMZ9+QPxr7PJOJaGLorG4SV4318n2OrxO8M8z4WzeeU
lXmmahrmkQaZ4I0O9Yh/DOh4DSTlOn2m8l/fMx5EawIfuHPj/wAZ9Gb9pj4t+H/2SdGtpLrwQscOv/
JMM6FBxk7uRosAMqeV4pgwOgxT3GMDnNV5DjPOOK/PJzbZ7dNaDZ5gg7n6VktOHV8Z4qS5mRQAcGuD
E8c15rq96baOSQDc4ycdO+P5ivy3OszlVk5X07H7zwlw/GlFJfEz/9T+lL4peP8ARPh4veE1R+Lv
P9417Hc3Np8XtaGoGKPU/Auj3xe3cx5j1TUIuFljcn544m5DfdLx8Z25qXxBL5q3ESbQpwM/jXheDf
IbYKJHikA98DOM/hX9cYaHJTT81+Z+A4mpzS26n7B/8ABPDSw0z03dCac+cjNRqSOhr4Oddyep7ECcg
5CZH619A3EZljkUcDGDx94Hgj9Xgj7h4h7qbPXnRbo8i6n53/FLTtO+Jvwt0n4eR3eP8ARPh4veE1R+Lv
r45+KvhN/BNeTUMCbbZCXgLdHQjd+mDn/d96/sbgbNVWiqUtNEz8wxdJxny9T5R+NvxQ4j4M/DiTxz
v1m/bS+NmpftpftA/EVvA/EVvA/gqqU4Mz8wxdJxny9T5R+NvxQ4j4M/DiTxzv1m/bS+NmpftpftA/
o+AeBxWw8LbDk2tRWiqUtNEz8wxdJxny9T5R+NvxQ4j4M/DiTxzv1m/bS+NmpftpftA/EVvA/gqqU4
m0ZGAIJBNUri8CxEbWY9MA4PJx159f8A9XWsHUdWhijZFlG5vkyG4U5x1HfNfn78UP2rNX8QeJNC
p8MxzeGvDOnXEVxZ2Ekf2eS+kkjKSXDqcsVKkqgOAMnJ5wPzr/4KJ/tJ/tJ/tJ/tJ/tJ/tJ/tJ/tJ
V3kdlNczw28EbRVA5Y9qEbWY9MA4PJx159f8A9XWsHUdWhijZFlG5vkyG4U5x1HfNfn78UP2rNX8QeJNC
QlGYlu5AQZ5AP4YQHkYnqFC5FefxVnM5ylKL0ij510lKR//Q/sf8AhyLwtsOTa1FaKpS00TPzDF0nGfL
E116yBemWPopx+VfjEWe+2xL0dle59HQp2VkfI37drCH9n67EVv+Iv+Cd/7Sf7Sf7Sf7Sf7Sf7Sf
j+AvAvhXwL4V8C+FfAvhXwL4V8C+FfAvhXwL4V8C+FfAvhXwL4V8C+FfAvhXwL4V8C+FfAvhXwL4V
j+AvhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPhWPh
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

+6NtsBt6jBrhtWvIkjkZ3kjYLlCkoRhjJJ3HgYwM+2a/QMklKWFpze7Sf36nl1MPY8Z+Ilrf+KrvRv
JdwwNPqIjTRyUwq2SspRwp6K7EybT1QIp4zX3p4MtI4LKFYkfbtAG484x3963zqv+6sjznozb1C+i0
CMqPx549q4vw00cP2GIZXGCfc1x5dg/3XtSHNNH/0f6/dFiBgHpgVoeKPE2heA/DF/4s8RC6fToDHF
J8WdU0i403wTpfmHwXZ3KGOS7R0KPqzRt/q2mTIhGAyxE42+YQPpOL89WEpttani4LAym0z0D4SeBc
getYct1ujxj5u1fx9xTxA69WU2z9PwGGSikhryliWP4VRY7j0pWJOTyB0xTB2r8zrVXJuR7MIWQlFF
2eUPH2nb9xAH+puMYz159K9ZgyW2k46V4f8AtLmSLR/g9rUIB+wfEDQZG+bAVZLjyGJJ4xtmP44r9g
R/D3w7MCtzb6RarOzLhpJ2jV5WbvkyM5Pua8K/bBkdvhzqmhoXZtf8Q+H9EPy5LQ3mq2kUi45zmN5F
diUjIxSH7x+lG4/3f1pMnJyMV56NoJ3Fqu33jViq7feNRPY1QmadAxySxJpvXNOiRlB3YrSjOyuNWs
+Sj/AIKf/sD337NXiy8+KngLRriT9nbXbotstiSPCOpSNuWOTH+qtZG3+U4+WORljyhaM17D+xx/wV
4h8G6tZyWeoWlzGHinhcBWRlPUEMfTHUEEc/gx+xt/wSt0PwZ+21488T6xrh8e/Az4d3UV14VkljDn
bP2I/Y++Btz8JPh/feIPFVkI/iv4tuTr3iq43l3N04JSDceqQqwjUcDANfXSsy5LKBS2yssUayHzHA
PSw0btJbmJrmsR6fA5kYM/YZ6V80eIddnvrmcGUiHtzW14z8SpPcTxK+6Qnnnoc15Tc3i4ZRvZzyMD
k2wjl8ZaxE2P7Mt5V3RWcTDpdTrzj7yROHxl4xWj8aPit4i1jxXp/wAB/gzc2dz8VdShFze3pHmw+E
ODKmOqRxdZe4uh/VnEOfrLsPKMX77WhRTSdF8N6DZeHfD+n2uiaVZ2qWltb2yeWixKABGoAAC8Kemf
17WmssYC2l3cMQvRUgcn9dtdXeWPzNw1d98F9GmvPEviNoI2kng8O6vKi46t9kkC/qRXsZpjXHCza0
RguQucjHJx65/r+Yz/DuZSlKvOT7s9qg7x1PjjX7WTwZ+09czbs+HvGOipJFHkKq6lYZ8xs9M+TOjH
/AnhrwHqWmTftDQazbazoZlUSx+HbVQRc3N4uC3lvA8sSxDa8jyqqnCsy/gt8VvHkGh6L4m8c+MfEt
fgTwtBZaXqGpRfZ7WGBQosLQDaxJ6neI9oB5YKzHlOfxpWJrjxxPKkfmiDTQoZuDEzy9h7iEc+1ez/
Y1Htjj5gfy61zt/F5YzjBJr1GOyEoClX249Kxtb8PP5IliVjtOcVyzV0cM9jzHHzbs06r93bPCOQRz
KhwOR0pQhBB4roESUoBPSkp6daAE2N6U9RgU+k4/Td+Hc/hx78ik03ojSlufoH/wSo+Cr/G39uzwBc
/EcsenCRANul2KCFGVsA4eR53wcjasZHORX7N6xDtLMoBwDjPev13hXCezwyctGy6qP/0P2P8FTNe+
JUDxeLdQcAjdtb8xX9NwfK7oU3d2R4n+238Ktf8AF/7PXhr4ofDlHt/jD8Oru38Y+FjESjTNCo+02v
uhA3IR74r5O+EenwfCjxL40+BQBs9F0cRar4dHJX+yrh32xJx83lSJJH3ONnrX8nfSP4QcqdPOMOve
rS7abXtSuV8P+H4Mm81K6/d21mo6tK54XHHvyAMmq2sWutfHyy/s7xFZ6v4X+DOQy6dLIyXvihC+d9
xtdY+Iemvtn1MJ8sttocvVUzlftoAUbT5Odu8dd4D+H3hX4beGtM8JeCtDtPD2gWq7Y4YuWJPLPJIf
9z+asyzetX0vaHbsYzIQzHbzXs3ws8JpfXI8Q3cLpbxEoiN/G/8AeFcX4a8MT+JdSSyVXggJ+eQDlQ
M/hgb34l+M7r4vObiw8BWkUmj+DbdmZfPtd2JtTKjgidkzET1hCN0kOMTxRdn4n+J7z4OaHPMtgojk
rWp14LCttMszzEALEiIi8AKMAD0A9KqJI2xic8d/T86qG5YsxyR9asRy7fmXOQeK/mipmMqld8u7Pr
f8Tm+QzOwyPmaKzhgBx903RHXOPRP2z9a0u28X/s92mtkvp+nf8ACQeKrjOClutvZraqcZzydRIGRj
R8njY/vHY8y8Jy3D+FfDzSpN5ktstw+csWlkJkfB53fM5P4+9c/qFm/jvX7XwZFNOugqFn1p4hy0Bw
nn6V+/YaCpU4010VjzK3mdVp1qj3EUUUEcaKdqoi4VfZR2X0HYV6r4t8c6X8MfC7alfENesggsoFPz
rDsdoZy3YKR3GcK2AniqyX2UeJX3sf/S/djSNRuvEGsar4h1GaS9nluHBl2llJBO7B9Fb5R7Zr3fwH
QeBfh7B8Qkm8IeDbuS6+F0ctxY+J/FEeVOtyoSrabpzEblhGWEl0p2nBjTcS5T9r4x4qw2VYS0rKyP
ToAOMD2qrpmn6R4a0fTNB0eytdO0mzt47S0t4YwiW8SIFVEUcBQAAAKy7q6kcup3NX8O8Z8dTxFeTc
0/7X6KKK/hA/UCdPuin0xPuin1tDYQUUUVQBRRRQAx+lLEM7/pTHJyR2pI5Ch4GalzsxPY//1P7fIC
ay2lxG4BEkboVYHPGMEjHvX8o8J432WMpyk9E0fbYyl7jSPmX47ahJq9x+z/ACaclpq1nefEHw88hZ
/bNfpa5jQffye3GM/hX7X4u1ualQcXpKDPJyyLauZ0/O8msVf9cP89q1ZpTlgMYNZSj98P8APav5cc
qOAufzroLaMgjaDxnisyC3YYPUe1dHA8VvD5szpGg+8SeinIJ/z0684wfreG8JOU9EeTi8RfQ8j+M/
l1C7dmeWeUjq7u5b26emPLfg1Fb/ABa8eLgSnw3HBJoZzcMd31C
jeGKXDfd2g13XjTxDHp9pPI00agA8Hqa+PdU1ebUruW7llJQliqk4z+Pb6/oeh+M4lztU06VPfufqf
sflGSRjS/aJ+PEPwi0NFeNvi/rcBv8v49Bv2J9f+GPgC/Bv2J9f+GPgC/Bv2J9f+GPgC/Bv2J9f+GPgC
OA+BnwatgV4c1JJ9WuPFnj/AFa5/tHxJ9oNxrN8QQzvjgRqAsaRjCoi4HJJr0nVZ2kkWRwu49cc
ScpO9zHny+AU5rHntlIJKkgVS8S+OvAg6eGeoNxrN8QQzvjgRqAsaRjCoi4HJJr0nVZ2kkWRwu49cc
iFqdupjmM2iX0Xlu23O5AMcZ9e2a/JnxR+2r4vAMxrN8QQzvjgRqAsaRjCoi4HJJr0nVZ2kkWRwu49cc
AHwrvNavbWwhh0K1S4mncIsJRdjbixGMFCMnGcV8sftZf8FB/hr8FNNuNJ8Ow6l4s1ZoWjuLqzkjtrOJ
bv0Hwz8QPjl1OzfSNIuIdIrRBEgUcD5RwxwAMnJ/IAfgy4MprFOtVlzRvexo3yI//17Pxx/
OyJCOQSer4964/VjNqN1LdXBZ5G+9jn885z+NeM+JtSinvDr1YNDC21z/t9x+Br2INxioR0iuh7NT
0YP51faySSMq67kNbkdoh6hePapvs6/3VqWZHiniDw1cQSNLGvmQtyCB93615veW7wSFWQZr6tms4p
ZDHGaSnkrtwDk9TTcV2cjOr2bGh8OBir0MmQDjtVHaM7uc1aiKjjNHIw9mzWjuAxC7T0qyDkZrKRtr
4RL8bf24vg94Ge/sNQ8Q2+s+LNWaFo7i6s5I7azjW78h8M/ED45dTs30jSLiHSK0QRIFHA+UcMcADJ
/IAfgy4MprFOtVlzRvexo3yI//17Pxx/
wfpX9CYOHJFRR6J5N8OfMX9p+5sSR6l4363KdhUYRL4rAC7Q0aN+S1yfwqhy
Ik4/Csbxj8O9Rg9I8XRG52f2N4jjtpLaz1a3RftNijlSRGT2JSMkHI+X3Nd3FBGFhWJcKq4ArWRN6ltoy
4M5Ygk4xxj3CKJflRAAi/dVQBt9MccD1Ax+FSQxbui8YxWnBbFTkjLV5uXZZQwkOXDQUfQ7MdjquIc
+tVmOMdGF2zKa0P/0v7RfBvhHw3pcNzHb2Nn5cMaxyXEcSxyXEcSxyXEcS

iNP8FWDNnasdoslxcjHvPfup558pfTn+xeCKrxmPo0af8KhBfefJYqktZGt4L0V9TvJbiRP3Syeby
yw789AOvYn0xX9Bqd3qfOYhq2p//0/6NPEJuvE07PrUktj4ftvnjs1A3sRyXkXIyAM8DJ6V4r4bfSF
bSA3zIhyCk9wA0SNnKgyP/CM/e3wi/Zd+FfwZSw1LT9Mn8Q+KoYBbpqmqP5stvGMEJbxjEdvGMABI
lTLv8pd0GYz9029lY6NaWthYWtta2kEYjhhiXYkKAYCqo4AAAFWZblItxTbuK7cn0yD/AErGmuTJK
ksTS009TXlOTZvDcSiiig0P/1P7X6KKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7Yn
+sxnUYdE1PUND+IJk+0YRbrSr+1N2qhiFT/R40kPIyFY8HOf00dllBdSjRnlSGyCOxB7183/ALUXwq
3saCOeJs85WWORT7g1+ncQcQ/wBoZVR/mprlZ4WFw9q9uh7E+Cx2+lZ27Eoz/nipQxd/mzUeBnPevy
tSTukfodbE22KcUPbH/1q0o7IthS2CeelYHjDxr4H+GHhrVfGfxE8VeGfBPhCyj8281LV7tLW1tUH8
c8W7bn7XRWLoG25dirbSRxnBx9a8Q+Ll3H4x1bRPgbpGozWWoayjz65NC5Eun6IuVuRleVlmG63Vhq
+JX7St/p2s/tG+Kbe3vPEEkFtHFHpCBAYdNhVeFSAMVIHBfc3ev0bJOEZ4f360bWPHq1HJaH1/pOk2
mZUUcnP0rl9U1COyhkuJpEWNVLHPYCtK8nyoGSDmvmD4teN5lLaFYO/myKQ5Xkheh45znnivDzPMoY
StGtifO1GVRkIBkFUJwhJI+8oG4nazP2i/2hvAH7N/w51Hxv48nkuScWum6ZaYkutaum4WCGP5vvHH
Lc9Id/I/rivi8NlmHtUXupad2z97v+CcHwr8KePrG/8A2u/HHifTPH/xo1JnsGhkTC+B1UEPYR27H9
JdXufietvBrdktlqWk6PqM0drrcUZJSO8hiYefsG4qZOEXco4NeJeI/2wNE060i034beFYIbKNcQyX
WppNLYzuP3+ba1PzRyPukG6VSpYjoy4+H/H37S2u6mkq/ED48eLPEhVsnT/C2zSrUMOzyW+2VsdOZ/
ubrUfCnw/wBEi1GR8tqOpk3d25PVmmk3OWPclznvmvNPEfx0+IGsiRbvxVd2VuRzFasIV/HYBn8a+U
rmvJb3xDcEsThc8da52bUZJCDI4PXHNckk2zldWXU//Q/JrUtaS4LkP8u4MFA4yPauIvkNxJuJLDJP
KvnqN2WzkikcyXLIu9+7tnJNW/EfiKXxhrz6w+1rCJDDY4H/ACyPG7/gQ/nVF18tSxIwKzdTUjkHRI
su3HOabU8yk42gVH5bYJ4qWzIjppjQjlcipghyM4qbywQOlSB//R/DTV/BtlqaySwp5FyQTx0Y+9eP
xmv/AA6oFxHmYxEdQOWI/DNeB2LO8Sv0z2Pb0/TB/GgDWpyAZP51GhJzk5p+SOhxQBaVnDDnHNS72z
H7TXx01CxDm4m0vwfps+3cyxQlru5x/dBea1B9dtfzOT8BpCNyqDuz09gfY4P5Cv7W/+CO3w3Pw1/Y
IP3ioPlBwPpXlWrybrh9zFVC5OTxX65g7W1JnKxhfAmE3Hxx+MGoPDI8UOhaNYLO2RvfzLmRh+IZCf
czy4jPfYK7ZSSR4cYWk33OxtIiV56itO3iyGVgDmq0MRycEjPNbtlbuynbCXIII+v+GMmvMrzuzuoQ
H//T/sE8GeFPtskOqX+6O2RgUGQCcHrzxj68V2/xA8deGPhp4O1fxl4qvV0zR7ONpi4UtI5AYhIx1e
QOPnt43RHRxw8iK4O1FB+pz/ADyME62IfunbGLk9BfgT8PvEnjHxTP8AtFfGfTY7fxnNHJaeE9Fk5T
pXxEqrlo9PkfSYTDJWuMlkJBBJJqFdoGVwD05odvkAOTxVUEhsAnHpX5TXxEpy1PZhT0LIZicACtPT
vy7+Avhm+fQrLVdaa4udQnU3920333u53aWV3/2izkfRRX1J+21rlreeFPh78F7e5v11nxhq6m5jgC
rfy0G7Arzv4h+KbbwXpNtfLp0+u65d3UWn6PpUD7ZtZv5M+Vax54y2CWbnYisxwASPTNTubDSdO1LV
QfMkR28MzY/S+J+KKOXYaVSb96Ssv8x0KDqSPXv2e/g0nwo8MS3WszWeq/EXWWN/4lv4VO2e5KhRDD
Nkhz3DyMBnk1XYksQTu96rbjnOcGkJJ5JzX5pPEykrSPcjSSJGGGbHem0yisJO5Xsx9NP3j9BRSUhx
OzuBYWRfKQ7Sa+ePB+k3Xw+8XeM9MsEafQGv5NWuLfc29IZ1LiaEEYZlcTBo0yWHzdVAb6DChkCnpX
78etRR29sd5t0ADHcxH8Z96sCNgNpXC1xexlORppsf/X/tmjBk+UdPWrUVoSy45PerVpBnAAz04raF
Uonl8O+EbO4Ad1wVNzeSYb7NagkAyOp3kbI1diQOP/4KDf8ABVz4K/sgeAfEHjTxZ4Z0/wAQ+Ibm
/doq4RYlAAXjcvJP6Zw9lOBpyUsXWjG3Rs+tyTwy4jzen7bL8DVqx7xhJr77Huv7WX7YPx2/bM8Zw+
zo1KhnRhuDAF16Hv14NetfCX4LeUeQHMPjR8SwJrA4qgb2ac4kkY1Sa3OOpalhjbcDz+NN
vGH5ZCeQHMZOCK/tfVj5YyAW6MQNoJ+lfPn7L6XP/gb0/j9lz4IeA/gb4Ou/HnizwzqfiHxDc3+
kMbHTyRaWkXnlTzSiRnkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkk
kMaE5Pc9hXi4nEwjFym9j0cNhZSfJDc//9D+2D4h+L4PD2lgHDXMpKwgHvjk/SvzL/aI/aC8Cfs9
dEijX+87YH/6q/ij/aW/av8AjH+2D4h+L4PD2lgHDXMpKwgHvjk/SvzL/aI/aC8Cfs9fIDc
yiecqo3ygFd2QMDGfJF8TqzNLNMk7MSSVfJ3dSSPTmvHkllU9Wf60jTSN/eQ+tfuWAy6nhoKFFWsfi
KypvEFycHzGHpXE+dcf3j+dOzKcEk/lVKVxSqNm9Lrc8oJMrA4qgb2ac4kkY1Sa3OOpalhjbcDz+NN
9uobS2XlpJGAVfqTXgPjvxyniL33Hg1yes67rniG8S61e4VoQMx20
f4M8KXPiK/RJUkh06PInfBwB6L33V9BX8Vr4f8N35tY44IILdygxwflOB+FYfEytviPLfDrpq/j7Sc
AIB6igADpSF0JJBpQQelIxP/0vxvyPUUtQv941IASAB1xXonoxVydVVhkntRsX1NCI2OlP2N6Uzd0S
SYDmCcfdYe+TjHcE0WI5EfH9nIvkIm4MBkbgMAnucduc1cyK5SO4u7d5La+tmsr+J3hniwf3ciEq3C
Oae4by12t/CCW+VgcZ64EjHB4PA64r/Qc/ZZ8QfDPWP2bvg3D8JfDVgMKI4
IdEDM5h6g31uD0uE+XIXiRAwOCEz7mSY2FCp7+zMz+0fVJdpckDG6vL9Yuobd7m4uWC26RlpDn+EDJ
EH1XzM+2K/W8DOMo88HoZ1dj3L9lTRr+3+AXgy8v7K403VdeFz4knhlbLxG9uJLoK5/vbZUGO2Mdq+
dqV5pvh27bSpVj1SdTZ2Z3dZnwAxA5wnU+vTvXWrDBGjS3TIkSKz7iceXhWy/pgDdn615bb6u3jDXv
YokiLElvmA5LN35zk19Q6Lp1rpdqttaRKiDhj3ZvesDwx4ei0SwUyJGly3zSMGyEGeASeg/rVT4g+I
PD4as41n8V3dqxSeKGRMR2EMgYFHdT87qwYRDHBfI+hdNstP0DTrHR9KsrGw0y0iW3t7eGNUSGNBtV
V5uEH7i6Hu4Kgnqxlzchywf5ax/OUlzkD0pZmDkhWwfSs9c7ivpxX4fjMVzH0tCmoqxdyuTll/OoV/
8PeHoAwVpr+cbUYE8bY0WeZv9mI+1fp/BeV/WK9OEVdt/h1PHxtVa3P/1f6frC6T4s/Hj4k8kfFN7iP
WUr89y4/1svT77M2T0JL8kcGuPktdW/aL8Rar8LfBz3Rr4vh4kfFN7iPWUr89y4/1svT77M2T0JL8kc
dqV5pvh27bSpVj1SdTZ2Z3dZnwAxA5wnU+vTvXWrDBGjS3TIkSKz7iceXhWy/pgDdn615bb6u3jDXv
YokiLElvmA5LN35zk19Q6Lp1rpdqttaRKiDhj3ZvesDwx4ei0SwUyJGly3zSMGyEGeASeg/rVT4g+I
fSQ00IqKKK4DcKKKKAP/1v7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK
8pd0GYz9029lY6NaWthYWtta2kEYjhhiXYkKAYCqo4AAAFWZblItxTbuK7cn0yD/AErGmuTJK
lTLv8pd0GYz9029lY6NaWthYWtta2kEYjhhiXYkKAYCqo4AAAFWZblItxTbuK7cn0yD/AErGmuTJK
ksTS009TXlOTZvDcSiiig0P/1P7X6KKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X
7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9b
Q2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/
UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFU
AUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAn
T7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/
1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7
op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK
/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EF
FFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAoo
ooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUU
UAf/1f7X7HKK/hA/UAooooAnT7op9MT7op9bQ2EFFFFUAUUUUAf/1f7X7HKK/hA/UAooooAnT7op
9MT7op9bQ2EFFFFUAUUUUAf/1f7X7H

OCCOtfzlftm/DP4M+HPiF4sj+G2kWKeGWuQ0H2e5lWJBty6hVccbt4JHXA54FfpGE8OsZWo8tOfs3
8L6yxi03UCjBtqxyfK2PQDjpXzr4igtIH1JNNt7eCI3DROsUYQF0BXOR8xwSepOcV5bfXEVrb3Eryv
GDR9bubbTLbThG8jQ+XKQ0zEN8yk/MFxxjHev6v/APg3S/Yl8S2vhvX/ANtr4w6N9j1PVk/szwTbTh
oa/07PAfhHw98PPB/h/wL4S06DSPDml2cdhY2kQCx28CKqoiqOgCqP1PeujhDEYpYf6vVklSjotOvU
6JdPb3mu2CzHOV8wFl/CvcxmNpQjeUrH8XZbgp1ZezpJyfkj/9H+5e+v7extZZ7mRYY1G4sx6AV+f3
xYf8FXP29p/2jviBP+z98OtaaL4FeHrxm1aa0ldD4n1CMlSsvQNbQtnC8h2PPQY/C54mWZ4mOCw/wv
c2067o3jZiMHHH+RxXj9vfgMuIo4F6BF+6v09q0V1MLj5gv/AAKv23KcppYGl9Xo7Lqfk/E3ENXH4
/CuWbxk6t89vaEewpP8AhMiT8trAD9aRB1PkYIyP0rSjiQoMK4x7cV5Ze+ONQDusMFnGR3K5rmNQ1
ZYS86/ekkdpGb8XJ/SqEgLDC/KPQV6p3nR61468Q6yiRrOmmWWMNFGMl/YtXFgtjYRnJyxz94/SrDZ
reK72ay01Da2sZCXOpKAwtc84TIwX6c8gZ5xyD9P6HoemeHtOh0zS4Db20YOBuy2SclixJy5JOW71z
heI5df1OCzV91pa5WMA5G4nk+/XFENwcLqx6h+zvAt947hQBWa20XUrxw3bFuQGHuCwH417zq2Egds
4yiNTnFRRDAIqrqMqwwZbhicCvRPQJHvF3HgVchlEkSuCM9OPrj+tcV5zA7hlua39Im84zDAwJFP6g
WtwB/wAsLgYKuT/EGAKnPfBryyC5BPyFQM8c54r728Y+FrHxloN74a1HzBaXCfMy/eTtlPQjg5/xr4
1r0j05JXOmjvHjPBBHpmlFyzsSX2jvz2rmftv+c0x78KkrMdqqm7PXoc4/HkUJkHS6U4uddvpkcOlv
3cD9h/8Agm1+2DP8Btb034R/EXVr24+DmuXGLSWUjHh3U5GOHR26W0q/KyDIjkIfjzMV/Qx4e0xvGH
/aQ03xNH45+DvxI8QNqvxnRbZ9I1K8nBl1rQreDy47cMfmM1sTLv5+cOGAxnH6BwzxBaHsJrU5sTzK
AOQsOcyMMcAFh7eMzSnSg6k3ZJHFRg5M/9X+oqx0PUPjb40f4Z6O1xYeC7Hy7zxZqa8CWNWDJpcLf8
hfQEe5dWM15eXeHudRumIaW4ncYzI5ViTjAJGBhQK84+GeLvxn8ctZWfz5bjxdLCzZyqRwQW8AA9cl
bVPit4vRodf1wh9OsnXB0TSusFsclv3xXDysD80jHHCrWF4qZviP8SNK+FkKi48PaOsWt+IiFyMsf9
9PWn3G3cuDu579aqvkIT3zX8t57mnO3Y+kw+F5SuzAsSKiB2NnrmpCMEio34Kmvm4ybZ6NPR6EoBLY
HhMz6L4cDP8AuprttyXt6vBySMW6N6LJj7xr0X9oTx9rc0MPwQ+HeszaR498R2sgu9Tgwz+GtJAxPe
1yyhluH/ALVxuitonvbufRY+TnUcYnSltb+NGtX/AMPfhzqkulaBY3KReKPEUB2fZ1wd9lZsQQ1y3P
aaTo1rGI4IYlC7B3yRgsScEltzEgkkkk1qSzAAnjrXyXHfiC8wbhHSmtgwmBktx1xcliw3Gsu5V2BP
ooAKKKACiiigAooooA//9D+2ZPuin0xPuin1/C0Nj9PCiiiqAY7EYwcVHvb1qYgHqKTYvpUu/QBoJ
H17+tfLHxu/bl/Za/Z98M+Itc8efGf4f/b7CF3TRbHVoLnUdQkUHbBBbRs0jyvt4XGcmv0nIeDZ1Gp
+CtXwze72a94Qt9OgKhlFprUdwyt1ILFFQ9uQ35V/MZ8Wfj/APFb45fGn4ofFPwp8PeJbd3XiXWf7T
mi+ayv62PtZNpH9RXx5/4KzeG9c+Gmt6D8B9G1OP4l3ttJDBdX09p9n08NuXcrpM4kmzyE7AZPpX8x
g+/8H6xoF34K+0+MXnDxeIVvJ0urOM4HlpHnaP8AVn59pB3YIr38qyuhhE/YppszjKzudBrN7Df29
KOBrue62IIzI/wDrG7ZYdQffv1FeD/GrV5vDthfTx4KXQNwVPCPNDEV59MqqZx1wT716yqSk9XsbTr
tYVweo/drn9c1+pH/BKb4JfDL4j/ALYHwo0jxpqX9m+GtS8Y6bZ6tqH2G4uOHpRTblLeX91eZ9Cfs6aVpvibxpJ4k8VuNI

*© 2023 Theo Gottwald * https://www.smart-package.com*

W8pUx3C4LNzngY+tT6jqF94iuI4kk8+aJjHJcSqXSAeoGQC/oh+pPavwh+H2YZtjnjMZ+7i3sf6aL6
Q+Sa+R1Hlg/MWFVdNsbOwtUgtBIsfBJYks59Wz3q+QhBBJr9pyDh+hl1H2GH2P8AMHxf8XMy4zzmec
HqfavSPVukBIHJOBSvcIsqwlWNy+fLRRuaVh0wq5b8hg963NE8Ma14sdBplsbLSWGft80e5CM/ejUk
g4/hHzdjive9I0jTNFsk0/SrC00+xQ/JFEmAPc+p96tIFA+U5FWE6VRAjrk7icVDJKiJ1yOmRVnYZC
pESIV9B64969y+LvjuNjNomnzKYlO2QqeCOwH05/Ovj3Urr7ZPLIzNtzgUij/9X+YSSQSOWB609RiN
+A2/rWKsnmvJJgjJzXa+IJIdN0C20yJQrSFBgdQFDc/rXDQ9PlGOlbnPLYyPEV6NP0y4uD94jYvuTX
VtNb8L6gSyxwaghkwcbhIrQjP4zL+VNGkD6RhPAxkDNXM1Sgc+UFVdoCKT7A0+g66Lsj/9b8dmfb2z
NEWSBG2MAo38J+vWvgG507xH4H1lvCfiy2ni2hvstzImzz0HTd/dYY4HO72r9cNH1WTTbhS4860YgS
L2NRvVcn5XQDov8AfHJ7gV5HIEl2vAwmj5wynOa55fGcU4WI8nkZJFe7fA3xUmmeJI9JupCkF0Cobs
16R71KXQkkjjlha2nWN7U8SI4LKUP3htAJOQSMAEnOK/W79lj/g4A/an/Zf0rQvhb8WvC+kfH3wJoc
HjsBSxEOSogqV3HZH+gt+xt/wWQ/Yj/bMt9O0nw18QB8LPig52SeEPF7R2WoebjpBIrPb3Knn5opSe
79elfqn+yn/wU9/aS/Zz8V+EtSvtfuvi74M0zUNOuzoviC8eaRorQz7UgvW/eQnbczKAVkT95gqQNw
51hnilIGY7eZiEugCQN8Z7jjORX6RTQF0L4XAPPzbtp9M8V/PGf5JOLamn9x6+GrJJI5CSPDg7cioz
45r+GLNH6SkypGseY0iUIgG0DGMe1TY2sW5LZzn3poRFIYSHP0p55ySQaVad+oOn1Gj5RgEirIGO5N
oAKKKACiiigD//S/tfooor+ED9QCiiigCdPuin0xPuin1tDYQUUUVQBU0JwfxFQ0+M7Wz7iqiruwp
c+s6P/wkOha5oJvLrTkvrOezeeFtskKyRlS6nsQCSD/Kv4mP2l/+CTf7YP7OUXiHULTwlH8ZPA8RZT
PeeHhq1wsVxFZzHzvKeKEFQYnlkjTJH7tUzkivmbxt4DnksdU0+28X+FJYHUiOI2cxlR8HDb1Ycgnq
fOq6X4tms44bjwv4etbhRGZG0y/nsvNYEE42lgDkZHHpVR9T+LGmLbxf2bHrsCyZT7XcoZAmOQZhjc
EJrK8gulH3thyU/3h2/GtGQDdu2gOBjd1z+HQfhVF1SRkdkBcZ28D5c9ccZrJqxxOLW45bpRuVN+7C
tFnkkNkhfXgmvrvWtetNItnuTIOclQeDX5kabq2p+H/EGka3pcypLbSrIUbpIB1AP8J+le8eIfjx4W
EHjyYsfKqj/a6t6sKpv4ktdYu3vnuVnKtmEsCRB7L259azJfFNnJL/Z1hHeXt0zMsYRcBxxk7mwCM5
B0z+dbdpoOmWtxJeXP2jUL09Z5yGbHYbeAB9MEdjXSat23OYS3vtclSXfLpmnHJJUATyA8bM87F9eN
T1NRSMRtVeWPQHr+VRZnG3c//9X+a1pmMu1OfWmPcFfMds+UvJODxzj+tZ/9oCe8/szTbSfVdXJx5F
y4IXg9snjoa9i8M/CmztD9q8VXEev35IdLcRbbWA9sxk5kPcMSOvTtXqtlpemaTaW1hptha2NpHghl
p91qNzHY2EMlxcP91UGSR611Gi+Gr7xJdrBaoEg/ilK5VRXv2heGNL8L2brbBpJ2AaaV/vSP657Adh
zHLH12n+dfn74q8Tz6rdSyNO77iSTnrSuaTTirI//9b+arX9YkubqV0LFCcjJNcoTzk1LLIznlt1Q1
/uglUwPTbn9KEM63xoy/2sIlbd5caqR6GubjYKpJyeKn1S4a61G9nYg5kPPtVOR/LgkcYYqu/H0Gak
uJVLbo8SLjrujdZP/ZBXavL2xXKeJ7VbjTZSc9Tz6ZGDTRpA/9f8WNDvhqGm6ddoy4mQvweNpAI/nW
4gvdDulktt0kbEeZGx4ce/v6GudUjA5HSpI2G7INBcZWPpCL+wfF1i8ckcFzE42vHKucH0YdxXxr8Y
/9D+ULUdB1TQb240zWrG40rUI8s8MwwW5wdh/jHPJHPc4rPZVCk4r9sPiH8H/B3xC0uWz1bRrK8iYP
pfDd9N/olwcxZ/hfsPxzX1rX5l2q39ldQ3NpM0dzGwZAOq//AF/SvuD4X+PoPFumLZXk6Lq8PyvuP+
/wAaj6H9K07KdlCknoMV3PjfSBq2krcRruvrWQTQ+jD+Jce4rzCwu45MyLgocY56DpWM4O5geiaJrW
dEU4yI7lsLcwgDAjkHmL1Ejfdr+e2zl2MCAa6GB1lVhJGsmRnc3OD6behH1rmxeCpYhfvYJm6mlsf/
T4w/FH4EeN9D+I3wf8d+J/h54006dZ7a90q7aEkhixiljz5c0LHAaGUNGwyCvOa/r5/4J9f8F4PAHx
IT6ZrpLe4s9Vggu9LuYdQsJlWSCeBhJHMjBSGVlyMFWDD1HIyKqyWxOSCGHYjO0/Ruh/CvxPN8llSl
ntVNoD//0v7aaKarbs8Yp1fwwfp5XbqabTm6mm1jPcYUUUVIBRRRQAUUUUAf/9P+1+iiiv4QP1+H2YZtjnjMZ+7i3sf6aL6
uDrtlG8Me27dnOO1c7FIVYAgrXQ2DpL99sDBP0r7rhjEvmaTPIxeHtFze2nr5H8Pf7Zfw58EePP2uf
2f8A2+/h7qHw8XWJZVXdFPZRW0gUDggTWMvPUljX5s/GxTZ+ETfIgWYny+eQ
/Lye+QOBXm+peK7e1fy7vTNWhIYjzI4TLGR6/L8//jp+tem+MWdr+VQgA3scCvL57SMsxVCCTk7SRk
Q30wT+uKwNV0S01CGWKZUcsuFL5Ow9iDyRXUaF8MrTxP4Zj17TdKhutUjkazvkRdkiyoeSNrc9m7cP
38OzYXC+dNrmnMHL7TcEgt3BBzjp61nyeD7H7UJNSvLjU92D5chXYo9CqkZ/GvYtfc+ihSs7nByanY
8JPOI9xpctipx7HkFpqTaldQafpbF5yCWYrJGIUGPmOQMNn+DrxXc6R4es9LYTqjXGoOv764f5nmbJ
ZOaJO4VVdWK0vf6VWBwwq3KF55PTFUZp7e1jlnnnihjQZJZgMfXPb3GTnjHeoscLw8uh/9b+btVLHA
ok4VQMAV61j03TZy/hjwX4f8JWclro9jHC7cvKAN7n3P4VvNGQN20D2A4H09KnQyMcY71ZaHKDglvS
S+G/CEeBdXaB24IyOles21nFBEscMaoOpPrV04nTSovqVtLsLTSrWO0gijhAH8Ax+dedfEfxzaeE9
2i/u7aEHKwRA8AD1PXNOU0tjWpVUdEf/1/51PFXiq61u9utQu3DysfugYCjsBXm0shlZ2JyanurrzF
+CIkhRIra2tFtR2wNuOK+FPhB4fTxB8QNDR1aZbMPcsB0LEbUH15Y/hX6O/ErZpHgG3tkxHcXE6Dnr
KtrQJbHk3hn93ZRO5+ZsE/UgGu1jkDrkkZrhNAkUwW0ZyP3YJ+pFdV5u1fk6CszI/9D+eUyKDgmszU
JTHa3lq0RPZmRtyjHrgyfhX16wdJWRgUGcjPcdQaDRO5WuEzk596rJIUBGAavSqGGD3qkiBkc9xVR2
ruFa1ldvayiWNxx1HrXIQzZbHANa8UpA6549K9O9tGeue8eHfFBeNInkBTuD2rsNS0vTfEOmTQSw2S
ubJMlrLB4liH8ar3XqoyRnBA+X9IuNT8LXcGp2s5twCDG6nIYZ/9m9Oo71+2lw1rqsJE0ayBuo9cYx
Fdw6bxjr7V8N6HPfeFNdt9U0u4jltyzRFgMEKPvJ9eefSvtHStTh1KytbuJ96SIGGO3tTOec7lmS3F
q3tg+tBB/9L+e6xnAwMjcDg89TXYWMisDzjtXmOnXcUsjFMhNwxk9q7/AE7azYJBB5FemeudGUGDgc
vxRBrXhSSQxjRdcnIsrh2ZeLW5ckWEr/AHQB+4djkqp5P9h37O/7Uvwg/ab8PXGqfDPxVb6pJbxm
S10XxxBrXhSSQxjRdcnIsrh2ZeLW5ckWEr/AHQB+4djkqp5P9h37O/7Uvwg/ab8PXGqfDPxVb6pJbxm
S10XxxBrXhSSQxjRdcnIsrh2ZeLW5ckWEr/AHQB+4djkqp5P9h37O/7Uvwg/ab8PXGqfDPxVb6pJbxm
lWN26Mobkdj0r8MzvgaVGo4zPXoVtNTnGiKjO3Iqp5LkEZ/Wupltgc4XHP8Ad/8Ar1EbBscKNx7cf
a4OdpGD19ageDn5hzXmzwc18SOinWvuVG6mip5EVRls571A+1Suw5z1rgdKSZutdhKKKKxAKKKAP

```
Y0hIAYZJzVmCTIXcD+NexluL9nJSRFbVcp+Nf/BZL4KXms6B8Lf2hdD0+8updBmk0HWlgVSfsN0ymK
a9t5clJY2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2
HfJZRiv0W+DX/Bu1+3V8Rby1uPiM/wAJvgf4akG+Sa/1d9Rvl44VbS2Tyyemc3C4J/ir+7TT2stPtY
3gnVNJ1z43/Ff4hfGm9hkNwdOt4Y9I02YDaPLeKMyTsoz1Mwzn7tfNX/AAWx/wCCfPwh/Zh8O/Ar4x
iwW81Ox0J9dsPk3GO5sj9pRgvf8A1bDHvX5JwnjKVPERdJKzaT0P1vizxAz7O58+a4qdVdm/dXotl8
SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9Etmkjht7YgcBFAqDW/Eunaajy3Nzb2wwQC7AV
N3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNl
/9f+ZJL3XJEhikt4ZtRkOYrW2feZCf4ZGwCijrvAGemK9w8FfCl5GTXvGMiz3jbTDaBSIocEfNg+n9
daFhZ2OMAVYhtpAxy2BSWmpyyoohjjVTgqM1oRQFtpOAPY1ZS33EFVFbVpp80hRY4yWJxUKbvZERon
yvIRuYg8Ivct7DvU0zzzOLW0t57u6d1jiji5aRieNo9ff0rx/wDaZ8e23wd8Jw+C9OvI1+JGrxE30s
dvO0jSFncnLMT1rJPBNd1SV2etUnced2WYkZNNpBS1iaURp61XLucoqK8Y7jqT6VYNVpJJoYnktYlu
4WJA/TivQf2Q/hnHo3gt3bEltp+mh2dufMfgAk+5wa8W+Ouom8+IWs2ivuFskduvcZA3H+dMdWHLG5
PyGK6SNm2YJzXPaURnGQBit1v4fpXonbOdh0rFQMHmnIxcDJ5qm5OSD0p0U3AyMCgmNZM5LxBNFpes
01d0l0ii6C4ztZBsIz9P5UzWnU0OqIB61nzfICF4BFaAbPbFQTgbW4HSmjZO5mh3xyw/OoWcZOTk1D
gXBDChgemW1wpA2nBrctbiZCjq4OD+Brz2wvNsmSwIxjrXV2U6MQdw4oA9W0zXG3LFJ8i5/Kulle3u
sacZZRhiBw4B7gVxfwu1OB4E095F8m4iF1ZNuyPLA+YE+uRnA/HpX3zPFBdQLHcKrIAFIxn5e/HfjK
8TRMv98OOfwCq34laisr/wDtK0t7xVCBwdwzyjA4INXUKMOdpPPX/PtXpnrnxUl83hnWL/QNado7m2
vdCvY20pi1qBtawlbbGQOhRuTGce7Bu60AfacSg20THkmkMYIxk1ynhbx1oevQxafJI+l6sCFNnc/K
o/aK8b/CZP7JW5PiHwhKCsukXv7y365JQdUPfsOmOc1+j3g3/gu9+1n8FfAi+BfDl3YeI/Ccbg2l3f
tfli1a/+IHxEkt3xKJdJ1WC2hY9NoXymTbkH6555r6H+HH/Bcz4s6VJHHe/GH4oWgdQXbW/D+la5FC
drdyBwB0NCy2ja3Kreh7ccbU6n9ufwg/4Llan4gureHXtC+DnxGsWjA/4l+rT+H9SZh3NteLJCxOOi
aeh9RXsngb44fFT4fyK3hjxzq9vaA82kzCW2kHo8T5B7enSvPxvC+BrRs4Wl5HVTx93aR/p4afcaTm
AftmmysUZTxlomifrhlzX9Lf7PP/AAVK8HeJdN0yx/aE07Q/AN1MQkXiXR5XuNDvcqTuZnBktOBkiU
yMq8bqSrKRyCDgimXNtszgZPSvx3OMr9lK1jspVmcyQQeaQmrskeGbIwPpVR1w5IPHpXys6DjuelFp
tGVGfpS0ULR3E1csRTHawII/Cml3XCj7vvTIyOakreOInHYXIf/X/tzSfaMBQfxqQXBbJCjHpVFgNx
yd/wUZ+Mdt8Dv2GP2l/iTLf/ANmXVr4Tvbazk83y2N1PG0MQRuzbnGMc1+rcH0ZSlFbt6nkYyOp/m0
dsSMZmHGcegA788kE47c1MmnyYJ2nB9q/qDDNezin2OGOxVluLu+fdf3N1fyE7t80rMf1pVtWONq4C
kV1s93baalsmx7q5uH8q1t41LPdSdfLUeuOcccZPTmvXfBnwzlgurXxJ4yKz34w9tZg7oLMnJ+b/AJ
NQkaxDAIHQ7e30rRkLuS+CGPBHqPU+p96Zg+hr3YI+pI5oxIcqSTUH2Z/wDIrRijZgTgitCOIBQDxT
q7oegRwRozAMcZY+tdeu2JtijaoHGKKbsyuQlaJYowMBR2wKzJ5wkbOWVE6deR+HXuOasy3aZKsT0J
58hteu0MHh6wkGXZtpKSbPcszD2Z89Vr8PfG3jTW/HXijWPFHiG6a/1e9nM80rHOcn7vtjOOK9v/AC
k0lefPc5CmEPtRsb0q03So6g6qK92xWIJ7Gut+Hvh668TeP/AAxo8cca24kN1I+7ByMKuePU5/Cua5
ivNVVLhsgZWEY2L684Svyk8dXz6p4x8V6luR45LyQqV7FSVA/INX7E/GrV7WyOtW1u0Vvp+l2jwQsc
ut+HZHBKYIOK9E9A57SnAYHFdJvB68Vx+lyK6p1ziuoTp+NBnVWg9uSSKYSrYyTSlwDjmqwkUttB5p
nPKP90ZHPDgfnSOilsfd2taZJpOsX2nsDlGOPoelY86ttPHavT/EsUes6F4d8XwsCk8CQ3C90lRAr2
gM9apPQCrO5Vyy9KkttUiJCurj8az7m5jQYOc1itMc/KSPepA9Vsr+EAHDnI4wa7HSbxRu7H0JrwrT
nnHSnXctxp17pmq2HktPBKrhJUDJKM/MjA9mXcvtnNcTp98YSFLGuoec3MDKrldynB9K9M9c8r8e+0
1tfxgsUXuAyqx78Bh/Fmvjjw9qTajp8MkoZL1CYLpGOTHMnysCfwJoAqeMvD6+KPC+paKdizOhaFsc
wB6h1f8A7+UAeWRyW1yUS7SK4iIwyEZx6nI5BPqMV6LonizX9LSC1TVYdW0tSFFveLvIGf4ZseYBjp
qx6EfWvluznMiSWzAvbsfmV8MH/DpXV6VcX2mPE2lX72kIBBhK5jZSOQMYZB04VgDxkHAw7gf//V/E
O3c9coiEA55P41xOveDBeztquiytpuqjqAoKXEZONkvr9eo69BivRRHHkAEE56cip/LUD5RtNJoykk
C8KXd/YahceH9RV3ZVLRsepI/hH4ZpEnfQRFcDpX1r+zx+0pq/wc1aDR9YFzrPgGck3FkHIaAbSGdC
deFbi5aOB2YqHmsJDk2sy8kx8xueT5ZJc/0EfBP45/D39oDwgPFfgHUZ50QCO9sJ1Ed5pc3Be3uIBz
8gj8TRJGmo6axIs/EtgpOYpxztkAZlinX54zIw5R3U+HxFwpQxtJygrSPoqSpzo8yvzH/9b+4G4tic
fyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+1+iiiv4aP
bs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw
+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWpyC
+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx
7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+
350wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+
BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+
4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS
9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo
2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04
SuPooooKCiiigAooooA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtf
jjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODo
Y2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/E
unaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwH
U/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooooA
/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHng
jkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl
8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7AVN
3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNlyP
xNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+1+iiiv4aPbs1U
CNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw+
HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWp
yC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8J
FZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hq
ttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO
/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FW
RHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC
6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCX
P4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350
wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+B
dJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+
4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+
XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oH
d3uo2WmhjF39vawDruYBvwHU/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6Gv
iD04SuPooooKCiiigAooooA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/
xMtfjjb5ksLdCNz2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea
+ixODoY2dx45+a4qdVdm/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgc
BFAqDW/Eunaajy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39
vawDruYBvwHU/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooK
CiiigAooooA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdC
Nz2yCDjkHngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a
4qdVdm/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaa
jy3Nzb2wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwH
U/hXIT+L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooo
oA/9f+1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjk
HngjkHkV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm
/dXotl8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2
wwQC7AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+
L45zJ/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+
1+iiiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkH
kV5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl
8SHVvoa/d5LY+LWpyC+EvDML/AC6Lp4brnyxT7+XS9EtmkjhtYgcBFAqDW/Eunaajy3Nzb2wwQC7
AVN3fa9feb9u1aWKM8JFZqIVx7sCXP4kD/AGaP7oHd3uo2WmhjF39vawDruYBvwHU/hXIT+L45z
J/YNlyPxNkzw05wl0Z93hqttGY20+350wEHpUp6moE6GviD04SuPooooKCiiigAooooA/9f+1+i
iiv4aPbs1UCNz61/Nb/wAHO/x1n8N+BdJ/Z+0HUfL1DxJ/xMtfjjb5ksLdCNz2yCDjkHngjkHkV
5D4C/ZL/Zw+HlnBb+H/hL4FWRHZxNdWK3E+4sWLGSXcxOSea+ixODoY2dx45+a4qdVdm/dXotl8
```

avAPlOAUPdXUgg/pj8aYzznxn8NNM8UFby08nStaBw0yrhJx/wBNAP5j9a8Yn+Gni/T5ZIzp/wBrUH
XPN7fTtixq6hCvPTvVZvDaxXQv8AS7ufRLwHJNvgJKf9tD8rfXGfevTzDHlVMSuvrjpUM1lE3zJGAf
ygDGOtc5N4TghmN9o1xc6HqBOS8LEJKe++P7rZ+maQNXO3zskCqoPrzjj6jpWbPpkMl5balGi/b4T8
Kr9D0I/u+1W0+8KqA7GG0LznNWVJ4Peg0jsf/9f8dlyrrICVkHRh1X6V+8n7GPxdl+JfwttLDVLkTa
Gj6vA4tQleex/X1/wS5vNTjtP2k/CMdp/wAUtaeKLHVbJsEiK4urFWnjXJIGCiyYHTzcV+pt1gnpgY
TkpSclszGPeoU6VM3tUKdK/GZbnsUth9FFFSaBRRRQB//9D+1+iiiv4QP1AKKKKAJ0+6KfTE+6KfW0
MJ90V/IT/wAHPvwZvLLx/wDsw/tDW0NxLZ6hp+oeDdRkjjzFbtC63dr5jdnfzr5R2wo71/XfFyq44C
yhtoIwGx6A5x/KnLHgEAcVf1vQ9d8Ia/r3hPxRpF74e8T6Xez6ZqVhcLslsbqFzHLC47MrAgjtwf4s
ruvEPjCALHpnhWw06yXcA99c/MF/hxFGDhfbdmu9+CHhmyuPA15rsNuJtVuLqeCeaRQHQo/yx5Hbbq
5JnQap4btQB/yy01jn8Xc1i3Xjn4gwGeWDxRDAh27IzZQttx1/h7/WvWtT8JahkNJYl+OpjOcflXmn
5pPBd+AAVSawkjGP9opJ/KoJ/Ct87FXt1Ynjkc//AK/T3r6W/ZK/YZ+L37ZPxj0X4UfCjR5L2+laKT
1D8ULP4V/Bn4IeCfH+qu6Nc3DmWCz0aEttaW9kcSBU7AjBz2bPH76z/8Eff2tLGG2TWPhT8HPFbLAJ
1P2fE7BgFOF7EV9tl3HGOppe2akz/MHjnN8nxWNbyTD+xoLa8m3Lz12v2R/FtrP/AATF/aL0S8nbU/
cq5YYJx1rw/4/wDxF+GHwT+DnxH+lnxZGnW/w+8O6TcanqTzQrKGjRCdiqerv8qqvcsB3r6PC+IdT
L0NY49zSAeZPIAD35Y56ceor+aH43/t5/Gb4nfHTx38bvCOpWfwq07UtSuLnTNA0COO1s9LtGkYxwn
hmDjBqSvy8yutOp3P7bP7Q+tfHvx5dT3DyW2lIzQWdvuz9nh9W9Wxzn3xXxRBCqJtUfKAFA9gMVqXc
QsmW3AZNTL1FJWcjkr7lSQLyD16Vmvv3L/dzz+PArTmOGbHHFZF1cGBJGwcBC2fpz/SoORXuf//U/D
umra+INQwYtN0/7VOSOFkcFif/HyPwFfUfi7Wpbm91TVrl1WW4meVMcABjkYH0Ir1kfZYb3I3Pkn9c
7qFz9N3Ne/3TElsk4614R8Z1RvCzlhu+dT/48KRzHmGlncgDDLqNp+uK31wFAFYel7QoI6lST+ZrbP
ugAMeYoxvGf91TX6OeGfE8XjbwBpl/aSpcNDCoZgc5jIyp/752n8a+AfEunNNardKpd0IGPVTwf0yf
zTxZCLW4VgT84zTeoONzgLqRjnpWFNLjJxjitO7LDPzGuenZgcZO3FOwyKeRCC4JzWZKUxnvTpZfml
mm3jboz9xieRXmyuCSMGp7Z3huIpocblOcHoaRk5pux9hWFyzmEIhZz0A9Kn1fxPoegWsNxrd/b6dP
7OPskQ6LxwSOtfK8T8VUMupXb9/t5H9E+CngFmPFddte5RW8v8u7sfRdv8Try5vpLe28L3+oaaXO2T
Pf3HpX4zLxWx1OsnOPuM/vCP0E+FcRgPZ0q0412tJc11fzVv1NC3uYftk0CyJJHKiTxMDkOT6H6c1p
zP8AOPxf8Jcw4PzaWV47Xdxl/Mr6P5n/1/xuqBy2/GBip6Y4GCe9emeuMDEcA0xhv+9zS0UAFFFFAF
pkdM+lW8VDbYMTMq/JuIBHTg4qevRPRTZKn3a+2/+Cc37IviD9Tt9rTwH8JoLS9tfhvpckXiTx5qQN
/glD+wvZfsN/s16TpHiaCzu/jd4lZNa8XXSJl4rpl+SzSTJzFAh8odATvfHzV4PEWcxwdBt7s66W5+
YSjoiqzZJPemAAdKWiviz0krBRRRQB//0f7X6KKK/hA/UAooooAKKKKAJ0+6KfTE+6KfW0NhBRRRVA
ZPjo0pI469G5/9P9BP8Agu1/wSin+IkfiT9eXwGeuaxhiGxdRSNMEzxRlVkYc
tmfs3+BdQuPAOoNu8d6Hotk7/wBk3btfOpQwxZKW7k5l2jEZO/BBavgeDeLfrEFQrfEtj6utQeqtdH
X4URTLLGZIZbedHHmBgwHmg5GRjPzEMcY6Zx0r7h/Zi/aX/4RP8As34Y/EiXkGdbfTNTJI/szBAS
vAL/AMOm483G+FXblsldoPU5HI719wfFKziWVobZA4VkcEDAG4Ejnp07duOmRWR+z5+zV8Sv2rPi34
pr2Xnfy9TyP9lz9jv4o/ti/GHS/hL8GHS/hL8GHS/qxY+dxlcDcR/eX+xbp4Om2oQ
4u9c1y5Ae+169PDTTy4ye4VOFRCABnNfTl3ePuYAivv3GhlycfiqW/E/xl8fPH3MONMdyxbp4Om2oQ
OTP23J9RuvD37C/w91ESWkKQeJfHM0JYbH+9ZWJIxlihE7gZAHlZ+9X9Ov7Uv7QHhn9lz9n74o/ti/GHS/hL8GHS/hL8GHS/
RdsfkxpF2jRe2Cf5/wANGvUprD0U3Ukf199HPhPL5Y2rxLxBJQweE1V/tzfwxSe79D5t1KRIp1soA2
xq+kXmXE85YTCv2WDT0it5L+8/0WhYiiGwfJ+lI0ZU8LgVdjBCgGmMjNkdq++o2irH81opjnpzShMA
5wSo5xzU+j6fLrGt6FpEYlkF1cpGyhScoDvbP4IR+NZVwwdsjBwSf0Nex/AvQZNa+IFtR+AvQZNa
bPwh4Ot0rDt6O0drAFXk
pr2Xnfy9TyP9lz9jv4o/ti/GHS/hL8GHS/hL8GHS/qpWRR+z5+zV8Sv2rPi34
4u9c1y5Ae+169PDTTy4ye4VOFRCABnNfN

```
/nNrlvE0ypCiB8P1xXT5b+7+tcD4lnSW6KoCSoxXsH1JzqMSXJBZtp4Hf/IzX3T+x94XW7uZdWWRXi
dRIEyoUKPp1ryLU7o+VNIPeuj8Q6ut7fTYBDZxjsBXBatcBbK6TdtO0kVrTjdHZUi7JHzT8Q71JdZX
AEyFjzWBjKN0f//X/l90hcZPq2a6hPuisDTVyqkcY5rcDEDGK9ROx7nsmS1IEIIPFESeYM8irGwepp
iWJ16hhX6A/Cnx9Z+N/DdleNMn26PMV0oGfLYdc46cYP418KajAqmRlWuObUr7w5dwa3przhLedJrc
7vUdPtmBu7y3tBjOZHA/Tqfwr7G/Yq+CHwu+Pvx3+Evhnx3f65q/gzWftMqRHUpkF0VtZZEjLK4OCU
aUk1fZ7H8cPhTSPG3xGvZdN+F3w38e/ErUV25j0jTJJV3E/d8zATP4496/Qv4Nf8EmP2zPisbPUvG9
vz3NvGjHTvDBx5L9WfV8P+BeXULfXZc7Wun5H5U/Ab/gkj+zV8IZYdY8cLrHxz8SgA+Zru2OyVwQQy
Z+w5fkmCw1NQwtJRXpqadsWwN5BbHJq3VWDJGT6Vopbu4Bwa8f2fY6px5XYzJQ6zRsiGU7gQo5JOeC
8OfZfsEWm/FPXkiRmBLLKY5s4Hb95X2/BCarTv2PxzxthzZTGS15ZJ/fdH//0f7Cv2jf2Uvgz+0/4e
J0inxzkfu2H3TnKj+2UFQvIAJ7k1zfi3wp4b8c6BqfhbxVpVhreg3kLQ3NvcRCRZUIOQwbjH8ua/nr
S4Wzu0Zhjc4DxkA/L0ySa/pf/bE/YMtv2UfFcfjfwTNc3/wm1e5aK1jdSX0S4PKwO2TvjIJ2OcHjaQ
48N63a+INGsdUtpEkWZNzqpz5b/xKfoa1JQQePu18o/B3xfH4c1J9J1J5Bps+cP2ikPc+xr62dN2C0
LLK2IlXLZIHH41r/AA+8GeNPHXiPSPCfgfw14g8a+PNWuSllpmlwmae9fICKAv3QAfmYkKueSMGvtT
Yfhp4YjvfiDNaRxan4r1DbNqep4zn98RmOPJz5SgKueK+ZzniWhg4XerLjBs/Lz/AIJqf8EStH+Ck/
wM0+S5VQyIoUDpj8z+pJ/Gse6mBBG7HvX4DxRxdXxDeulz1MNh7jrm43fOGOayZZzuyxPegkt0PH1c
p9fwtDY/TwoooqgCiiigAooooAKKKKAP/9X+2miiiv4YP08KjcnnBxipKjcY+YdaT2AZuI7mpNzgZC
YzUV1P/W/tgJycDk1YhNN8tc5xQ3yj5eK/hiEbM/UElaxpQgh8cZ/wD1V+OX/BwH8QtU+H3/AAS8+N
k0+FVYmV7K4S5ZRgH5mjhlQZwMt27/qfAEl9Zipdzx8ZA/zWrePChAoCjAGB2rURQowKr2rRSxpLDj
ey42PrJEblVVnwOBmvLb+UzXMrgbRk16deEx2zsBznFeV3WTJMMkHJAI7VnKpZaGXkVLiNpE+zxbHr
zCltECc5jDSbmA6qpEbEdwuO+a+6tM8ClA11rssvizVnI3XN2BIYSBwIFPywoARtVAMDuSTj5Lifi2
WftJfEzwr4c8Uad4Q0Dw/p2q2cV7BFqerLDPHDJGHTegQkOQwyvVTkHkZr5H/ZV+Bs/xk/aC+GXhK7
vFLBOc6srXbbsr+SX+Z/MZ4s/wCCZ37XVusuo2/hjwXqqrlcWuuIXA7YSRU3NnAIyK+Fvin8Kfil8K
V8P6jPJbXtslxE+22lYZjcFWwVBwe/NcGS+MuNjUX1uCadj4DNfBXBTp2wc3GXT9D//Q/CFpXKSMAh
jJ2rIM8HPyv4luhc+IL2ZXMgaQnPrVZbmVPFUlWpbM/SuLOEcbkuNlgcfHlqL8uj9NDU0kERLnjj+t
PerYUA5AqS0f/9H+ZLVrfy2uMAEZrh7uCPzJWfPzYBB6AAEfrmvVNZgBV3ABGa81vIj5jcDrXpH0i2
c+9f5zGheIZvC2p2ms2SRLNAxYblHyqQQ4GQeCFZD7N61/ez+xD8Rx8X/wBkr4D/ABBM8t3cal4dtP
A7ySnFTeQv8Ac/WvxfnR+lmWAwIOOaUKd3pWl5SICc4A/WvTPAvwr13xvcRyW8BttJV/3lwwwMf7Oe
SAs5Prk9Pwr58+Kfh1NI1Z2hjKox6AcV9BjOFqmFpe0mfnmW+IVLH4r2NLQ//9L+hqO3HmANlVJC8r
n1PRdQiJJ8vabAR4Tn1jOce1fyTwjNqtJd0f2v4sNyymaXRx/M/UG4YrtxUEMu0ryCR2pb4ldxxgVk
wL4U0HxZplzofjTwl4zOm61Zy4HkySWzqr5A5jkDxyxuPvLInGQa/wBEmAhvvdAcjHY+1fz1f8HAHT
Mor60MTFq3aWlnb5an8B9lBJEzEBTyDz3wc19KfDnxmr2kWj61MwuAcRTMc7h2Brw0QqGHlbTGUDhq
frQQf//T/F2f74+lQ1eeNHOdp6UmxOyivVUkeuUxwcsDivXvgH8DvHv7S/xl8CfAv4Ma9qPgzWf6Pi/
2FVAxYjGR/eJ/wAEu/8Agm94R/YR+Gl1qmuR6P4i/aD8RQxP4m1mBQyW6p8wsrNjysEZwcjBc7mI5X
aVU8D3OPy6VDdXIxsQhE7ADAA9h27dPQelYU87H72Cc1/NOe557ZynFu71PXwuFJZ55AW2rkVmtLI6
KKKAP//V/tfooor+ED9QJ0+6KfTE+6KfW0NhBRRRVAFFFFABRRRQB//W/tpooor+GD9QJ0+6KfTE+6
d1khkVl4dGUEg54IIXGDw2cHpXNo7xyAEE9614Z5Ao/lX02S4x0ZXTOTE0ebQ/g+/wCCzH/BJDxj+y
0TX9N03WNHvImguLe7iEsM0bLtKOjAhlI7FSK/mC/b8/4NyPANj5Ao/wCCzH/BJDxj+y0TX9N03WNHv
4z+FevQymJDqUSS2l6vJD2l7Cz29wmP40bJHUKQRXjltEQgfBIIyPcetfolLGKprBpry1M4t3P/Q/r
FxsHPrXJG3BudvRC2CPaueaRhK1rnpv7OPhWPxl8fdEtJYYJTpfh291GJHQMDKzxxhsHuN4IPbAr9S
RB4X8G2nh/S5Z44pFuXVTI57gDgZ7jG4577fY1/OHixCbzCMraWR/qj9DbH4R8Lyhdc0JtS7q+qPdv
LSQG3v8AVpJ9eusjDNJdSGUAjqNqNHGM9kWvpIqmwBogJlPyMM5Pscf1r8exUnUaXY8HijN3jMfUry
2lj+8xHnaOoHmE/hX8Q4ekudW7o/1Cwk06kY+aP5PP2j/B8Wk/BlvEF5AftWn3NnNG5/gkkZY3/76V
c5r+rPDPC1aOXOVbTmeh+E/S/wAfh6vFCp0veqU4QjJp9dXb7mj0LSvuD6VvxqpGSM1zGjd/pXWw5+
1FXomJPJ7ivSPpUzP1WFnTC96861GAx5O05Jr1m6UNE2RmvN9XUsxxQD3ONkiG5XAA569P1r+u3/A/AI
/subw9pt7wPlMiXUyDn2WRx+B9K/O/FHBwq5RNveLVj9O8IcZGhnMV3TP6jkjBIH86mjtZpW2wW8sz
KvEDC5YuXm5qj2Se3r2PBfhx+z5d6ibbWvFjfZrIYkS0ztZiD/GRyB2x3r7D0nQbDTbO1srKzht7SI
RCGxlsV9WG0YjJVc+leQ/FbTxLpZO0fKCcV+McQ4Dmw0j9m4SzL2ePg4s+BTEQGKFN4BKkrkBuxP40
h5LLBL4OtZDKXUiR1nfgY54B5yfpX5pw5DlxTiftniHiFPKavnb8z9PL8FQSTkYrAkJ3Jg10l6QQPp
XNbMM/c7faurJ82lRas9jPE00z/OT/4Ks/8ABOHxF+2H4Ks/jp8Ddfs/Dv7QHgqz+xaZqGpSeXZ6
YHkHOK/m5/aU/wCCEf8AwUb+Hvi+/wBb8Q/sqeNPiJ4Ze4Z9P1/wPbDWLe4tyxKSSWsTNcRMBgMr
geo9e8Af8G3Xxg8Q+Ibe3+Ln7Q/gnwvpLMPtF54Y0m41y4RT1IjkkFsjY9CyjPqK/Ur/gnz/wAEZ/2
Sf2Ffh/Z6Z4Y8E6P4s8ZLbqNX8beJ9Pg1DVLy4xlyjyq0VsGPCxRbVAHBDEEn9Ao/wCCzH/BJDxj
+y0TX9N03WNHvImguLe7iEsM0bLtKOjAhlI7FSK/mC/b8/4NyPANj5Ao/wCCzH/BJDxj+y0TX9N03
WNHv4z+FevQymJDqUSS2l6vJD2l7Cz29wmP40bJHUKQRXjltEQgfBIIyPcetfolLGKprBpry1M4t3P/
Q/rFxsHPrXJG3BudvRC2CPaueaRhK1rnpv7OPhWPxl8fdEtJYYJTpfh291GJHQMDKzxxhsHuN4IPbAr
9SRB4X8G2nh/S5Z44pFuXVTI57gDgZ7jG4577fY1/OHixCbzCMraWR/qj9DbH4R8Lyhdc0JtS7q+qPdv
LSQG3v8AVpJ9eusjDNJdSGUAjqNqNHGM9kWvpIqmwBogJlPyMM5Pscf1r8exUnUaXY8HijN3jMfUry
2lj+8xHnaOoHmE/hX8Q4ekudW7o/1Cwk06kY+aP5PP2j/B8Wk/BlvEF5AftWn3NnNG5/gkkZY3/76V
c5r+rPDPC1aOXOVbTmeh+E/S/wAfh6vFCp0veqU4QjJp9dXb7mj0LSvuD6VvxqpGSM1zGjd/pXWw5+
1FXomJPJ7ivSPpUzP1WFnTC96861GAx5O05Jr1m6UNE2RmvN9XUsxxQD3ONkiG5XAA569P1r+u3/A/
```

```
   dv8AHn/g3N/YX+KBvdQ+FV78Sf2dNbYFoxoOoi909nxxvs70ShV74iePOe1fzv8A7Vv/AARK/aK/Z
   kfhjfgS3DMAfmFZgtQWJI6+9fb/xC/4J+/tfeAZpf7V+CfifVYIwWabSNl+mB3HlEk/lXzBrfgLxr
   DntyB+Vfvn+xp8Zrf4x+A28E63O//Cd6XbLbyksub2zICR3CIDkkM3luc/eIP8QNfgfZ3FsHZf7Rs
   h+reBvHONyHM+VXdGr7slZ9tH6o//9L+hXwh4L1FdD0XSdF0q7uILa3S3jWOMkbU+Qdv9n1P1r2DS
   rM/8WsV7edGlFJKT1/pn52+N/hBeeA9DXVtT1GK4keYRokYOBnnHSvgb9sfX4PCP7Lvxx8TzeH9T8
   ZwffFfN5zhKeExkYJe6tfXY+94EznEYvL3WnK9RN/wDAP84H43/F7xF8afiP4j8c+LLq0k1meYKI4
   kf8AsAeLUzd/s+eH9HlY7y2kX13Y4PoAsrcfp7V+vZf4tYKMIwlTaUex+d8QeFePxmIliZVuactW3
   gHuB4X/aM8V2J2nyxf6Fbz9iQDskjGeMZx+Fe7Q8VMrna7ab8j4zEeEuaQdo2kfzVe1RqrbskYr9s
   K1jBONH16C4Zx6qhIP4da+pwnE+EqxUovRn55jsDUw1WVCsrSjuf/9P+c9VbA44odTtNfY3jD/gn/
   Iex61Yi69eOK9jtf2af2lL5DNbfs3/H6VM9vB2oZ/wDRVddYfsbfte3xIsv2Wf2hblgQDjwleKPbl
   g+hr6Q8Ff8EXf2v/AB8UfxLF4D+GdmGSTfqGqC5nRWOD+6gRs4znbuHTrXl4/ibBYa3tqiR9BlfC+
   lz9x5Xllm8sdFZc9azP2R/8AgjR8BvgJ4h0bxZ4vvdQ+OXxO3K0L6jarFpllcZGJLaxy3zZxjznk2
   U92/NdDvdA8LaVoFtDp2l20FpapwBGuAa62O2woHAb1qzDbg4+UjFXBAAvY16GFw0acVGKsj8xrYy
   xBtvM0q5CqCdueK8/NqKlhpLqehkle2KifnnrUA+3TIpILPtyOCMnGRnuOo9689/ZouLe1/bu8ZWM
   9zqlqcAf9ts5+lfj+Vw5caz+jc/iqmT1E+kW/uP1x1JVWXC9dtcjOD5hPbH9a6W6yQWbriuen7/jX
   nZ2CYBbNfxVQzKpDqfo7wyOoivEAOdpFK98vYqPSuT3y/wC1SiVgCDvbPtXes/l5mKwiZ0kt/I4PI
   Cp6cNi+VH//W/toHUnGKWiiv4YP08KQ9DS0h6GgCtRRRXOMKKKKACiiigD//1/7X6KKK/hA/UAoooo
   Eobn6lHcKsL90VXqwv3RW7KnsOooopGQUUUUAFFFFAH/0v7XnTLsc96eDj60N940lfw6j9VWxaVtuQ
   wi37O37QDySJZ+DPHlm2ruMKqaVfI1lOG45UGeB+ehXNfeNqx254zgV4D+2J4Y0rxp+yf+0FomuRPI
   YeIAWjjvbPnKRu3nRLnqNj54xngYruP2YfFOreOf2bvgH4v8QSR3Gu6j4R0m6vJVBHnSPaIzMQSep5
   yT69eoB7V2Hw7/Yb8O+EL1U07TfCvhPRdyGW30jT0tX/tIXPEhQLnr1PTqBkA1+i1r901bjQZPWvyKh
   OteysMo6I+MnWc3eR5z4r8J2PivRdQ0LU499pOpU8ZI9xXyU37Mnia3uJYrTVbCWxz+7LqQVHuBmvv
   1a4upF44j2ZP5mv0BSNduea+Z/jWi/ZpOMda+WzPhvBUaTnGGvq/8z6/h3jXMcTiVGpU93tZHxdNa4
   wXviD9u2ZIUt5B8W5YWI537NPtFz0Bz2x7V+tyxJEjMrbSRyfvfjhsjPvX5Jf8Erc7v25JCxYt8ZNS
   wKoNM+eoqMzP7V8/U4jqrZs+ejQjY3QlqwCrDFDjuFyR9M5xV1Y4wpARVHoO5rlfOfpkVqW8rkqueP
   aXouo6vYS4kgubdTJHLGQCjhs42kEHPvX6j6gq3FrPaSgPBcKIJl/wCeiMVUg/gxr57/AGfCknwq8J
   Pk+rf71fXVvtUgKMDoPYVkW3PX1rUg+8PrX1GRZVTwlD2VPZnicRZ1WzKs69d6v8D//V/vCgbKEY5c
   hM9HKJWxMT4L8RwP9vm/hIyQfQ1478O5rW3/AG7fgeW3fgeq8hi86X4feRraNScFQt1pr49+FFe6KAFvZGA
   I4wAF6Y7VzU33W+h/nXj8R/xH6n4ThvgR//W/tjtgKMDoPYVkW3PX1rUg+8PrX1GRZVTwlD2VPZnicRZ1WzKs69d6v8D//V/vCgbKEY5c
   FWP/1/7Yk+8KnqBPvCp6/hWGx+oMKKKKsQUh6GlpD0NAFaiiiucYUUUUAf/Q/tfooor+EA/UAoooo
VDE.
RET.
```

Remarks

-

Limitations:

-

See also:

- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- ANA.#FromVar 1406

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.4.2 VDA. / VDE. - Inline Data

MiniRobotLanguage (MRL)

VDA. Inline Data Start
VDE. Inline Data End

Variables with Inline Data

Intention

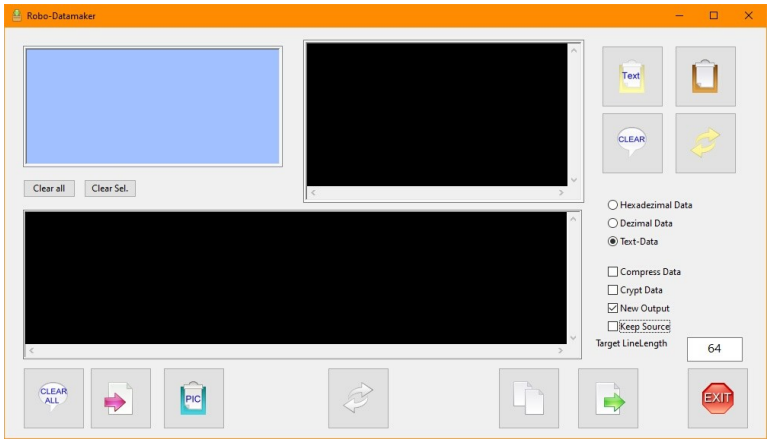
This command can be used to include Binary Data into your Script.
It is very useful to also include pictures or other data directly in the Script.

There are 4 Options to use this commands.

1. Just write your Data in dezimal numbers separated by ",".
This way you can manually enter data.

```
VDA.$$DTA|$$FIN=
64,65,255,0,21
VDE.
```

For all 3 further Options you will need to use the Datamaker.exe - Utility, which is in your SPR-Package included.



You can simply drag and drop files on to the Datamaker-Tool, to have them converted into Inline-Code.

There is a dedicated chapter about how to use the Datamaker-Tool 3483.

The important Point is that the VDA.-Commands supports 4 Data-Types and additionally 2 Options.
You can simply set them in the Datamaker-Tool, the rest is done automatically.

Supported Data-Formats:

*© 2023 Theo Gottwald * https://www.smart-package.com*

General comments:
As you can see, Dezimal Data starts with the Sequence "`?D". Then there is a 0,1 which means the Data is compressed but not encrypted. All this is done automatically if you use the Datamaker-Tool. The Data that has been created using the Datamaker-Tool also includes the Original Filename of the File - which will be in $$FIN - and a checksum to make sure that your data was not changed.
The chacksum is automatically calculated when the data is extracted. If the Data was changed, the VDA. will not extract the data and set the Timeout Flag..

1. Manually entered Dezimal Data: These are dezimal numbers between 0 and 255 separated by a ",".
This way you can manually enter data. $$FIN will be empty because this data can never contain a filename.

```
VDA.$$DTA|$$FIN=
64,65,255,0,21
VDE.
```

2. Dezimal Data: These are dezimal numbers between 0 and 255 separated by a ",". Each number represents 1 Byte of Data.
As you can see, Dezimal Data starts with the Sequence "`?D".

Looks like this:

```
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  386 Bytes in  7 Lines.
VDA.$$DTA|$$FIN=
?D0,1,110,0,0,0,74,81,236,212,59,216,196,99,142,123,50,231,210,17,63,
92,23,69,204,17,252,33,210,155,44,235,50,194,135,6,208,22,59,220,
54,112,63,212,166,150,66,0,107,254,33,79,44,112,112,48,228,80,57,
66,0,107,123,27,22,136,95,124,168,167,215,151,155,168,210,137,135,
92,42,62,194,173,226,6,148,10,42,36,169,15,60,40,235,88,44,96,229,
181,175,53,99,103,6,66,152,10,13,86,58,70,1
VDE.
```

3. Headezimal data. These are Hexadezimal numbers, where each uses 2 Digits therefore there is no Separation between them.
Hexadezimal Data starts with the Sequence "`?H".

```
' F:\00_MR\PB_MiniRobots-Editoren\SPRE 02\Source\Sample Scripts\Tests\File.txt
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  232 Bytes in  5 Lines.
VDA.$$DTA|$$FIN=
?H00016E0000004A51ECD43BD8C4638E7B32E7D2113F5C1745CC11FC21D29B2CEB32
C28706D0163BDC36703FD4A69642006BFE214F2C707030E4503942006B7B1B1688
5F7CA8A7D7979BA8D289875C2A3EC2ADE206940A2A24A90F3C28EB582C60E5B5AF
3563670642980A0D563A4601
VDE.
```

4. Text-Coded data.
This Format starts with "?M". This Format uses least Space because it uses all Textual Charactes to store the Information.

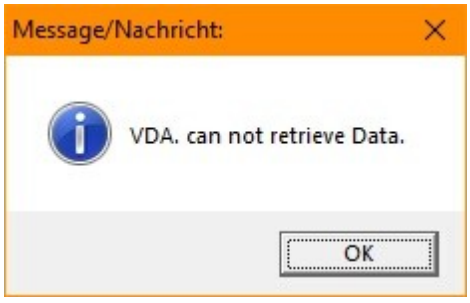
© 2023 Theo Gottwald * https://www.smart-package.com

Please prefer this datatype for larger Data-Amounts, as it unpacks very efficiently and uses least space.

```
' F:\00_MR\PB_MiniRobots-Editoren\SPRE 02\Source\Sample Scripts\Tests\File.txt
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  154 Bytes in  3 Lines.
VDA.$$DTA|$$FIN=
?MAAFuAAAASlHs1DvYxGOOezLn0hE/XBdFzBH8IdKbLOsywocG0BY73DZwP9SmlkIA
a/4hTyxwcDDkUDlCAGt7GxaIX3yop9eXm6jSiYdcKj7CreIGlAoqJKkPPCjrWCxg
5bWvNWNnBkKYCg1WOkYB
VDE.
```

If Data that was created using the Datamaker-Tool was changed and therefore the **included checksum does not match**, you will get in `$$DTA` the Text "`VDA. can not retrieve Data.`".

```
' Sample that leads to the Error Message "VDA. can not retrieve Data."
' This data is invalid therefore it will not be extracted.
GSB.Laa
MBX.$$DTA
ENR.
'------------------------------------------------------------
:Laa
' F:\00_MR\PB_MiniRobots-Editoren\SPRE 02\Source\Sample Scripts\Tests\File.txt
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  154 Bytes in  3 Lines.
VDA.$$DTA|$$FIN=
?MAAFuAAAASlHs1DvYxGOOezLn0hEdw/XBdFzBH8IdKbLOsywocG0BY73DZwP9SmlkIA
a/4hTyxwcDDkUDlCAGt7GxaIX3yop9eXm6jSiYdcKj7CreIGlAoqJKkPPCjrWCxg
5bWvNWNnBkKYCg1WOkYB
VDE.
RET.
```



Syntax

VDA.P1|P2=
<Data>
VDE.

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 – ***Variable.*** This Variable will be filled with the Binary Data.

P2 – ***Variable.*** This Variable will be filled with the Original Filename or a "-" Sign if there is no filename. It may also be empty for manual entered data.

Speed in Ticks:

The speed of the decoding from the Inlined Data heavily depends on the used Format. For our speed-Test we used Lines with 64 Characters. As you may have seen, the line-lenght can be adjusted in the Data-Maker-Tool.
For 2695 Lines of Hexadezimal Data the robot needed 524.000 Ticks to extract the data , that is roughly 194 Ticks per line of Hexadezimal Data. For the same data compressed an in textual format, the robot needed just 6725 Ticks.
Therefore Textual Data is the preferred way to include large sets of Data. Its more efficient in terms of Space and its optimized for Speed. In Decimal coding, we get 4765 Lines that equal about 325kb of Data is extracted from Decimal coding in 496219 Ticks, that is a 1/32 th of a second which equals 100 Ticks per line.
Therefore if using Textual Data within the 32-bit Data-Range, the Robot should extract the data generally below 1 Second in Real-Time.

Example

Sample-Script using VDA. and VDE. to store and display a small picture.

```
GSB.Laa
ANA.#FromVar|1|$$DTA
ANA.Show|1#!
ENR.
'============================================================
:Laa
'------------------------------------------------------------
' BITMAP from Clipboard(.bmp).
' $$DTA = Data
' $$FIN = Filename of Sourcefile
' Length:  1392 Bytes in  22 Lines.
VDA.$$DTA|$$FIN=
?MAAGuBQAASlG0yTbYYlYMQvkOznPrYa3D1MGg62CLOJf5wNvQQ99VP4L6gQQ5AQXC
OpZoDz0mKBeeLz0riH9lGqZQA79lN4b4AtrGQBuz43T5ifxZuUGlkhKiUgJlmuYw
k3eBRZ9a0WwN+wjqxthPIlWS4rJSTa7stW/t3kQBBcK6Vgi/hQq+XAXCutbI35Vy
cnr2e4dGgPuje8jH5vNDpNH9JmQJeUncSSENg3+Fx97EhJVfEiLVUgKi6nkAOwA7
Q6tQi8CGFKHNY+9du0AbM6NUiUHgy+AzrJivUop+P6VXFqAW7M3beMK6Vgg/Rapu
dHEj5VTggYgu0pyzMxTHsdsVxDLgZZmrHywVUnJyvaZYhz4U7MHlb7IdttBLoBZs
DXvIFVaAekJ5xx+vZ5VydaSQNX0onmz9e0hVNoD7d3Y2RgQKWaHdmbz4CQArkK5g
+wgnFmufWo78ANv4v2o5oOvoz9XxAKuGe4DygnB5IQX3DmfZabFFJ9Hszdv43VVm
gFOfF99Jkd2W7D2ryF9VkuKy0g3O3E2jjw0D/LXUnIcBFGQxZfqCag4kGRm0PSsI
P8VqjuRJZwmBzcvjMop+P+VntliHAcTZ0f2mpGkpkXifWg48BZt2E9JshcfetNEY
b7LaxgNLoFM84EsgExxwA4RhtYqW4LPs8I/ortxFYspOnG2zB8lhLROSscWS3odl
J4Z4//o2WxeebI3+2EcepJSt09RBYAuA4/SHmVkoq1MvjZv209TBoOvgSxinkQw9
g8R1Zca9td60QSUPcvJ9Brjf+gHE2dF95kQZ8Q0+ePfmyJKWYDladdJBeP96djtn
JqhXvp91T5rmmNOaAdtG+LdGiP8lOhYorw3OXA2Df4VHHqQUbbNEmYAdm0Bzr7HY
134/5ef2uFdZECOMfWAKJygTY3fda7BbqG/t3sRBJVJy8joW61+6XgTkxWcurJ0v
EnAxD3HNHiifWg68oKtQCwCgosnR/am0VCQRRZfeDw3zjGU3yWgPevYCRyZW8SeE
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
40IJyHk5qeBvTSOX3bOKGfH9Czvn5shHphMzLBWffgK6GfSvi92+HGja2gTL4/QH
GYnBmMngSijd5ZyzM5R5MAXVubRZKYFFYspOnO3zZ2m01wn5iXzZaTEFh4EEOYFF
4koerO064EsI7ums2X/ooAYwOM5/slWxtwtI7QJtGqbQs2y4PyoWoBZsjbvo+l4E
oVVajvy9a2ufWg484Mvgs2y4P6pWAOZEmbFt7sDb0MOf9Q/6NtvXfnz1KubIx+bz
w+SxbW6Ae6Drq39KhvgCWoZgC7vn9jiX+UAb8LMHyWEtk/fB5W+yHbbQy2DCPBVG
OwE=
VDE.
RET.
```

Remarks

-

Limitations:

Binary, octal and hexadezimal numbers can not be negative and they can not be floating point numbers.

See also:

- STR.TEXT TO MIME 3215
- STR.MIME TO TEXT 3198
- ANA.#FromVar 1406
- Inline-Data and Inline-Pictures 3483

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.5 IVB. / NVB. - If Variable Between

MiniRobotLanguage (MRL)

IVB. Conditional Statement

If Variable Between

Intention

Conditional Statement.
Test if a specified numeric variable or point-variable is between the two given limits P2 and P3. Let's see an example:

```
' Here we define the value that is tested
' and the minimum and maximum value
VAR.$$ERT=15.05
VAR.$$MIN=15.06
VAR.$$MAX=16.001
' Here comes the test
IVB.$$ERT|$$MIN|$$MAX
  DBP.We're between
ELS.
  DBP. We're outside
EIF.
```

As expected, the result is:

```
[13]  We're outside
```

If we change the line:

```
VAR.$$MIN=15.04
```

we get the result like this:

```
[11] We're between
```

Now lets see how it works with point-variables. Imagine you want to make sure that coordinates stay in a specified range. Its always an x and an y - coordinate.

```
VAP.$$ERT=55,30
VAP.$$MIN=0,0
VAP.$$MAX=45,40
IVB.$$ERT|$$MIN|$$MAX
  DBP.We're between
ELS.
  DBP. We're outside
EIF.
```

As the x-value is 55 and therefore exceeds the given range of 45, we're outside this time.

© 2023 Theo Gottwald * https://www.smart-package.com

What happens is, that the instruction will test that the x and the y value are in their range that is given with the two range-points.

In case of numeric variables, P2 and P3 can be any numeric variables with about 80 bits of precision.

In case of point-variables, P2 and P3 must be Points also

Syntax

IVB.P1|P2|P3 ELS. … EIF.

Parameter Explanation

P1 **-** Variable or Point Variable with value(s) to test

P2 **-** minimum value

P3 **-** maximum value(s)

IXX.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'***********************************
' IVB./EVB.-Sample
'***********************************
'-----------------------------------------
: $$LLV=0
: $$ULV=1
: $$TVA=.5
IVB.$$TVA,$$LLV,$$ULV
 $$ERG=1
ELS.
 $$ERG=0
EIF.
MBX.$$ERG
'-----------------------------------------
: $$LLV=0
: $$ULV=1
: $$TVA=.5
NVB.$$TVA,$$LLV,$$ULV
 $$ERG=0
ELS.
 $$ERG=1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
'----------------------------------------
: §§LLV=-5,-4
: §§ULV=5,6
: §§TVA=0,0
NVB.§§TVA,§§LLV,§§ULV
 §§ERG=0
ELS.
 §§ERG=1
EIF.
MBX.§§ERG
'----------------------------------------
: §§LLV=-5,-99
: §§ULV=5,0
: §§TVA=0,1
IVB.§§TVA,§§LLV,§§ULV
 §§ERG=0
ELS.
 §§ERG=1
EIF.
MBX.§§ERG
END.

' Using Point Variables
VAP.§§ERT=55,30
VAP.§§MIN=0,0
VAP.§§MAX=55,10
IVB.§§ERT|§§MIN|§§MAX
  DBP.We're between
ELS.
  DBP. We're outside
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- ELS. - ELSE 2150
- EIF. - END IF 2148

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.6 IVC. / NVC. - If-Variable-Contains

MiniRobotLanguage (MRL)

IVC./NVC. Conditional Statement

If Variable Contains

Intention

Conditional Statement. Test if a specified text or binary string is contained in a given variable/text. The first parameter must be a variable.

There are several special operators and the 2 options "**e** - equalcase" and "**b** - binary".
Let us start with a simple example. We want to compile a script and make it react on a commandline parameter.

```
' Get the commandline
VAR.$$CMD=$cmdexe$
IVC.$$CMD=/u
  MBX.We'll uninstall it!
EIF.
ENR.
```

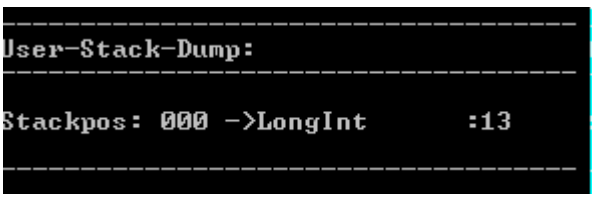
Now in case that we want to react on /u as well as on /U, we would need to specify the "e" option, that prevents the command from being case-sensitive. We would just change this line to get the desired effect:

```
IVC.$$CMD=/u|e
```

Now if you look for some special parameter in the command-line, you also want to know where it has been found. The position in the string, where the "/u" was. You get it from IVC. on the TOS.

```
VAR.$$CMD=/c:file.txt /U:mix.log
IVC.$$CMD=/u|e
  STS.DUMP
  MBX.!
EIF.
ENR.
```

The result will look like this, because the /u starts at character 13:



 © 2023 Theo Gottwald * https://www.smart-package.com

There is more. Assume that you want to know if a text contains ANY of a list of special characters. Going from here you would need to check for each single character. But help is here! You can use the **<** (**ANY OF**) Operator.

```
VAR.$$NAM=Rolf Müller
IVC.$$NAM<üäö
  DBP. german Umlaut found at $$000!
EIF.
ENR.
```

But do not intermix it with the **>** "**ONLY ANY OF**" Operator. This one will make sure that the given string contains ONLY characters that are are contained in the text/string given.

```
VAR.$$NAM=Rolfi
IVC.$$NAM>folR
  DBP. Its Ok!
ELS.
  DBP. Non matchin character found at $$000!
EIF.
ENR.
```

Which is not the case here. Therefore we get:

```
[12]  Non matchin character found at 5!
```

There is one more option to know, that is the "binary" option. When you want to find a string in a binary file, you must prevent the robot from searching for variables and special folders etc. in the binary code. The variable with the binary code must be resolved - and that is all.
This is done using the **b - flag**. Binary data is often read from files, maybe to patch them.
`IVC.` can help you on this. Here is an example for the usage of the b-option.

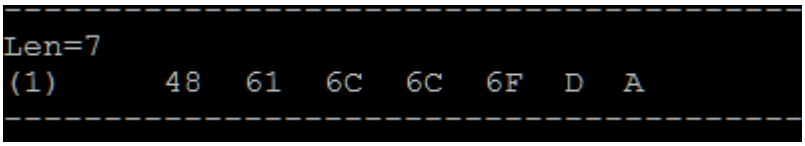
```
CFF.$$FIL|$$BIN
IVC.$$BIN=$$PAT
   ' do the patch
  STS.DUMP
  MBX.!
EIF.
ENR.
```

Hint: If you think that the evaluation is not correct, use

```
PRT.$$VAR|h
MBX.!
```

to check for invisible characters and Spaces. You will get a ASC-Dump like this, which shows all characters inside the variable with their Hex-ASC-Code.:

*© 2023 Theo Gottwald * https://www.smart-package.com*



The negative Form `NVC.` is also available.

Syntax

```
IVC.P1P2[|P3]… ELS. … EIF.
NVC.P1P2[|P3]… ELS. … EIF.
```

Parameter Explanation

P1 **-** `Variable[Operator]`

```
=  "contains" - the whole string must be contained in
variable.
    TOS is position in variable.

!  "contains not" - just the opposite of "=".TOS is position
    in variable.

>   contains "only any of". Must contain only each
    of the given characters. TOS is first non matching
character.

<   "contain any". Is true if any of the given characters is
    containes in variable. TOS is position of first match.

e   equalcase. The compare etc. will be done after all
characters
    have been converted to ANSI-Lower-case.

b   binary. In this case the Variables are only resolved once,
    specialfolders etc. are not resolved.
```

P2 **-** `Variable, contains the text/string with the characters to look for.`

P3 **-** ***(optional)*** `can be` **`b`** `- binary or` **`e`** `- equalcase.`

`IVC.`'s can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

© 2023 Theo Gottwald * https://www.smart-package.com

Speed in Ticks:

This command uses typically around 130 to 250 Ticks heavily depending on Parameters.

Example

```
'**********************************
' GFS.-Test
'**********************************
'DBM.2
'JNF.Move-Console.txt

: $$TXT=12345678Abcde
: §§NUM=456

' This is a "Contains"
IVC.$$TXT=§§NUM
 PRT. §§NUM is a part of $$TXT
ELS.
 PRT. §§NUM is NOT part of $$TXT
EIF.
' Stack contains place where it was found.
DMP.4

' This is a "NOT Contains"
IVC.$$TXT!§§NUM
 PRT. §§NUM is not a part of $$TXT
ELS.
 PRT.§§NUM is a part of $$TXT
EIF.
' Stack contains place where it was found.
DMP.4


' This is a "Contains ANY"
IVC.$$TXT<152
 PRT. $$TXT contains any of: 125
ELS.
 PRT. $$TXT contains none of: 125
EIF.
' Stack contains place where it was found.
DMP.4


' This is a "NOT Contains ANY"
IVC.$$TXT>5a6
 PRT. $$TXT contains all of: 5a6
ELS.
 PRT. $$TXT contains not all of of: 5a6
EIF.
' Stack contains the position of the first non-matching
character.
DMP.4

DMP.
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- 1.5. Features and Hints 80
- IWS. / NWS. - If Window State 3724
- IVC. / NVC. - If-Variable-Contains 3524
- IVV. / NVV. - If Variable Value 3534
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.81.7 IVP./ NVP. - If-Variable-Pattern

MiniRobotLanguage (MRL)

IVP. Conditional Statement

If Variable Pattern

Intention

Conditional Statement.
Test if a specified string evaluates true with a given 3.2 Standard-Search Pattern 341
.

`NVP.` is the negative Form of this statement.

Syntax

```
IVP.P1=P2 ... ELS. ... EIF.
```

Parameter Explanation

P1 - `Variable = Variable/Pattern`
`must be a Variable or Stack, Length=5`
P2 - `Pattern to compare with`

Important: Only **P2** can be a Pattern!

P1 must always be a variable.
P2 can be variable or Text and must be the Pattern.

IVP.'s can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'*********************************
' IVP.-Test
'*********************************
'DBM.2
'JNF.Move-Console.txt
: §§TXG=Outlook 2003 - Posteingang
: §§TXI=Outlook 2003 - Inbox
: §§PAT={Outlook&AND:{Posteingang&OR:Inbox}}
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' First we test against the german version
IVP.§§TXG=§§PAT
 PRT.Pattern german Ok.
ELS.
 PRT. Pattern does not fit german.
EIF.

' Next we test against the intl. version
IVP.§§TXI=§§PAT
 PRT.Pattern intl. Ok.
ELS.
 PRT. Pattern does not fit intl. .
EIF.

DMP.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- IWC. / NWC. - If Wildcard Pattern 3537
- 1.5. Features and Hints 80
- IWS. / NWS. - If Window State 3724
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- IVV. / NVV. - If Variable Value 3534

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.8 IVS. / NVS. - If-Variable-String

MiniRobotLanguage (MRL)

IVS. Conditional Statement

If Variable String

Intention

Conditional Statement.
Test if a specified string is equal, not equal, or lexically higher or lower then another.

If **=** or **!** are specified as operator, its a simple string-compare if the strings/texts are equal or not.

If **>** or **<** are specified as operator, the command will make a lexical compare of the two strings. However if you want to sort items, take a look on the SRT. - Sort Items 3147 -command.

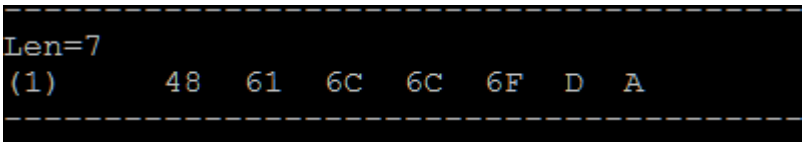
Please use `IVS.` for Strings instead of `IVV.` - which is only for Values (numeric).

You can use `NVS.` as the negative Form of the command.

Hint: If you think that the evaluation is not correct, use

```
PRT.$$VAR|h
MBX.!
```

to check for invisible characters and Spaces. You will get a ASC-Dump like this, which shows all characters inside the variable with their Hex-ASC-Code.:



Syntax

IVS.P1(op)P2|P3... ELS. ... EIF.

Parameter Explanation

P1 - `Variable`

 *© 2023 Theo Gottwald * https://www.smart-package.com*

(op) - Operator: =,>,<,! (! is NOT)

P2 **-** Variable

P3 **-** ***(optional)*** can be b - binary or e - equal-case.

e - equalcase.
The compare will be done after all characters have been converted to ANSI-Lower-case.

b - binary. In this case the Variables are only resolved once,
specialfolders etc. are ignored.

Example

```
'**********************************
' IVS.-Sample
'**********************************
: §§STA=Alles
: §§STB=Allex
: §§STC=alles
: §§TXT=comes alphabetically first
IVS.§§STA!§§STB
 PRT. Strings are not equal!
 IVS.§§STA>§§STB
   PRT."§§STA" §§TXT
 ELS.
   PRT."§§STB" §§TXT
 EIF.
EIF.

IVS.§§STA=§§STC|e
 PRT. In this case we are here.
EIF.
IVS.§§STA=allEs|e
 PRT. In this case we are here, too.
EIF.

:over
MBX.!
END.


'*******************************************
' IVS.--Sample 2
' This script will mix anybody with anybody
' unless June and Peter
'*******************************************
'DBM.2
: $$PAT=Ralf Rolf Peter Franz Jacob
: $$TAR=Monika Sabine Pauline June Sabine

FEM.:Lev1|$$VAR|$$TAR
MBX.See Debug-Window in Editor
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
END.

'-------------------------------------------------------------
:Lev1
'DBP.-Level 1 ($$VAR)

FEM.:Lev2|$$BAR|$$PAT
RET.
'-------------------------------------------------------------
:Lev2
'DBP.--Level 2 ($$BAR)

DBP.Now is $$VAR + $$BAR
IVS.$$VAR=June
  IVS.$$BAR=Peter
    EFX.
  EIF.
EIF.
RET.
```

Remarks

`IVS`. and other IXX. can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or `FEX`. (Enumerations) to unlimited Depth.

Limitations:

-

See also:

- 1.5. Features and Hints 80
- IWS. / NWS. - If Window State 3724
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- IVV. / NVV. - If Variable Value 3534

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.9 IVV. / NVV. - If Variable Value

MiniRobotLanguage (MRL)

IVV. Conditional Statement

If Variable Value

Intention

Test a condition and execute a block of program statements only if the condition is met.

There are two forms of `IVV.` available. First the quick-decision form, which is always like this:

```
IVV.(variable)(sign)(Number or variable)
```

It looks like this:

```
IVV.$$MAA=1
     ' Do something if result is NOT zero
ELS.
     ' Do this if result is zero.
EIF.
```

And secondly there is the mathematical `IVV.` It is formed like this:

```
IVV.(Any Formula like in CAL.)
```

The brackets (...) are needed so `IVV.` knows that you want the mathematical `IVV.` The result is tested against zero to decide whether the condition is true (if it is non-zero) or not true (when it is zero).

You can formulas with all operators from `CAL.`, if you include them into (...).
In that case the condition is true, when the result is NOT zero.

```
' This will make an OR'ed decision
IVV.($$MAA<9)+($$MAB>0)
' Do something if result is NOT zero
ELS.
' Do this if result is zero.
EIF.
```

You can also use a Syntax with delimiters "|". In this case you can use formulas on both sides as you wish.

```
'
IVV.($$MAA*5)|>|(8*1+0.5+$$XYZ)
```

`NVV.` is the negative version of `IVV.`

© 2023 Theo Gottwald * https://www.smart-package.com

Important:
Use `IVV.` only for numeric comparisons! For string comparisons use `IVS.` or `IVC.`

Syntax

```
IVV.P1|P2|P3
IVV.P1P2P3
   ...
ELS.
    ...
EIF.
```

Speed in Ticks:
This command uses typically around 60 to 120 Ticks, depending on the Parameters.

Parameter Explanation

P1 **-** Variable
P2- Condition
  ! (not equal),
  = (equal),
  > (bigger then),
  < (smaller then)
P3 **-** Any

When using Syntax 2, Codnition (P2) can be given as a Variable.

IVV's can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Speed in Ticks:
This command uses typically around 70 to 150 Ticks heavily depending on the given Parameters.

Example

```
: AAB=2
IVV.$$AAB!3
 IVV.$$AAA>1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  JNF.MySub.txt|$$AAA|2
 ELS.
  JNF.YourSub.txt
 EVV.
ELS.
: AAC=Hallo
: $$AAD=Nur
  IVV.$$AAC>$$AAD
  JNF.MySub.txt|$$AAA|2
 ELS.
  JNF.YourSub.txt
 EVV.
EVV.
```

Remarks

Use `IVV.` only for numeric comparisons! For string comparisons use `IVS.` or `IVC.`

Limitations:

-

See also:

- JIV. - Jump-If-Variable 2177
- IVS. / NVS. - If-Variable-String 3531
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- 1.5. Features and Hints 80

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.10 IWC. / NWC. - If Wildcard Pattern

IWC. / NWC. - If-Wildcard-Pattern Compare Previous 3534 Top 41 Next 3539

MiniRobotLanguage (MRL)

IWC. Conditional Statement

If WildCard Pattern Compare

Intention

Conditional Statement.
Test if a specified string evaluates true with a given Wildcard Pattern.
Wildcard-Patterns use the following Patterns:

? - matches any single character

***** - matches none or more of any characters

- matches any single numeric character

```
$$TXA=*Peter
$$TXB=Hallo Peter

' Here comes the magic
IWC.$$TXA|$$TXB
  DBV.Yes, we match the Pattern.
EIF.
```

NWC. is the negative Form of this statement.

Syntax

IWC.P1|P2 … ELS. … EIF.

Parameter Explanation

P1 **–** Variable to compare with

P2 **–** Variable with Pattern

Important: Only **P1** can be a Pattern.

IWC.'s can be nested to unlimited Depth, and they can also enclose Sub-Programm Calls or FEX. (Enumerations) to unlimited Depth.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'***********************************
' IWC.-Test
'***********************************
$$TXA=*Peter
$$TXB=Hallo Peter

IWC.$$TXA|$$TXB
  DBV.Yes, we match.
EIF.
END.
```

Remarks

-

Limitations:

-

See also:

- 1.6.1. Program Flow Control 121
- 1.5. Features and Hints 80
- IVP./ NVP. - If-Variable-Pattern 3529
- IVC. / NVC. - If-Variable-Contains 3524
- ELS. - ELSE 2150
- EIF. - END IF 2148
- CAL. - mathematical CALculation 2315
- IVV. / NVV. - If Variable Value 3534

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.11 SAV. - Save/Restore Variable

MiniRobotLanguage (MRL)

SAV. Command

Save / Restore Variables (Encapsulation of Variables)

Intention

This command can be used in Subroutines to safely use Variables, that may also be in use somewhere else. For this the Command will "Save" the Variable Content and give you a fresh and Clear Variable. You can think of making Variable "Subroutine local". **Think of Encapsulation of code.**

`SAV.` is like brackets in Math. (3+(4+5)) There is always a fitting Closing Bracket to an opening bracket.

In the same way **you need to call a `SAV.Restore` for each `SAV.Save`**. If you do not call a `SAV.Restore` for each `SAV.Save` there is a danger that the internal Stack which holds all the Variables may run over with time..

In Short: You can make a
`SAV.Save ( Sub-Script) SAV.Restore`
like a Encapsulation-Shell around your Subprograms.
This will make sure that all used (and specified) Variables will be the same value and state that they had before the Subprogram was called.

```
SAVe - Variables (DO whatever)  SAV.Restore Variables
```

`SAV.` will:
1. Save all specified Variables, whether they are currently defined or not.
2. Variables that are currently not defined (not in use) will be SAV.ed and Cleared, but deleted at the SAV.Restore.
   Therefore `SAV.` will create Variables that are not in use at that time and clear them.
3. Restore all SAVed Variables deleting the Values that may have been in these Variables.
   When there are Variables that have not been defined before, these will be deleted at the Restore.
4. Has no depth limit. You can go as many levels deep as you want.
5. `SAV.` uses a LIFO (LAST IN FIRST OUT)-System. Calling `SAV.Restore` will therefore restore all Variables that have been saved with the last `SAV.Save` statement. And the next call will restore the Variables from the `SAV.Save` before this one.

Lets see an example:

```
:Test
' $$TXT will be saved and cleared
SAV.Save|$$TXT
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GVT.$$PAA|$$RES
  VAR.$$TXT=$$PAA-$$PAB->Type is:$$RES
  DBP.$$TXT
' $$TXT will be restored and get the original Value and Type
SAV.Restore
RET.
```

In this Subrprogramm we use the Variable `$$TXT`. At the time of writing, we do not know if we will use this Variable in a Script that may call this Subprogram later.
Now we just make `$$TXT` "local" to this Subprogramm using the `SAV.`-Command..

`SAV.` **has unlimited Levels**. Therefore you can call Subprogrammes that are protected with `SAV.` in a recursive way.
Or call a Subprogram that uses such Variables, and calls another that uses such variables.
`SAV.` accepts up to 70 Variables, if you need more just use more successive `SAV.`-Statements.

Just keep in mind that `SAV.` uses a LIFO (LAST IN FIRST OUT)-System.

```
VAN.$$PAA=5.56
VAN.$$PAB=7
' Level 1
SAV.Save|$$PAA|$$PAB|$$PAC
 MBX.After SAV. the Variables are Cleared:$$PAA-$$PAB
 VAR.$$PAA=Hallo
 VAR.$$PAB=Peter
 ' Level 2
 SAV.Save|$$PAA|$$PAB|$$PAC
   VAR.$$PAA=Inside
   VAR.$$PAB=SPR
   GSB.Test
' Restore Level 1
   SAV.Restore
 GSB.Test
 ' Restore Original Values
SAV.Restore
GSB.Test
END.
'---------------------
:Test
SAV.Save|$$TXT|$$TAX
GVT.$$PAA|$$RES
VAR.$$TXT=$$PAA-$$PAB->Type is:$$RES
DBP.$$TXT
SAV.Restore
RET.
ENR.


'--------------------------------
' This is a Subprogram that will change the Format of a date from
' 20181204 -> 2018-12-04
' Called like this:
' GSB.DateFormat|$$RES
:DateFormat
SAV.Save|$$YEA|$$MON|$$DAY|$$INP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$INP=$$_01
GSS.1|4|$$INP|$$YEA
FMT.$$YEA|0000
GSS.5|6|$$INP|$$MON
FMT.$$MON|00
GSS.7|8|$$INP|$$DAY
FMT.$$DAY|00
VAR.$$RES=$$YEA-$$MON-$$DAY
SAV.Restore
RET.
```

Syntax

SAV.s|P1[|P2..|Px]
SAV.Save|P1[|P2..|Px]

Parameter Explanation

P1 **-** Variable-name of Source and Target variable

P2...Px **-** ***(optional)*** any Number of Variables to preserve.

SAV.r
SAV.Restore

Parameter Explanation

No Parameters are supported in Restore-Mode.
The Command will always restore all Variables that have been saved with the ***LAST*** *matching Save-Command.*

Example

```
'*********************************
' SAV.-Sample
'*********************************
VAN.$$PAA=5.56
VAN.$$PAB=7
SAV.Save|$$PAA|$$PAB|$$PAC
MBX.After SAV. the Variables are Cleared:$$PAA-$$PAB
VAR.$$PAA=Hallo
VAR.$$PAB=Peter
SAV.Save|$$PAA|$$PAB|$$PAC
VAR.$$PAA=Inside
VAR.$$PAB=SPR
GSB.Test
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SAV.Restore
GSB.Test
SAV.Restore
GSB.Test

END.
:Test
SAV.Save|$$TXT
GVT.$$PAA|$$RES
VAR.$$TXT=$$PAA-$$PAB->Type is:$$RES
DBP.$$TXT
SAV.Restore
RET.
ENR.
ENR.
'*********************************
```

Remarks

`SAV.` can be nested to any depth.

`SAV.` is like brackets in Math. (3+(4+5)) There is always a fitting Closing Bracket to an opening bracket.

In the same way **you need to call a `SAV.Restore` for each `SAV.Save`**. If you do not call a `SAV.Restore` for each `SAV.Save` there is a danger that the internal Stack which holds all the Variables may run over with time..

Limitations:

Note that currently Variables will normally loose their Variable-Type (and get Type 1 String) if they are assigned to another Variable. You can avoid this by using the `VAO.`-Command:

```
' This will preserve the Variable Type
VAO.$$ABC=$$EFG
```

The overall maximum length of variables depend on your available RAM and your actual Memory fragmentation

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.12 VAB. - Variable Binary

MiniRobotLanguage (MRL)

VAB. Command

Variable binary

Intention

This command can be used to assign a Value to a Variable.
It will not resolve any variable-names that are given on the value side.
It will not resolve any specialfolders or systemvariables on the value side.

It will just take the right side and put it into the variable name on the left "as it is".
Therefore its binary compatible.

If there is a variable-name on the right side, it will not be resolved with `VAB.`

```
VAB.$$NAM=Rainer
VAB.$$TXT=$$NAM
DBP.$$TXT
```

If you want the value-side to go through a one-time variable-resolution instead, use `VAO.`

As `VAB.` does not go through any variable-resolution process, its possibly the fastest way to clear a variable.

```
' This will clear a variable
VAB.$$TXT=
```

Syntax

VAB.P1=P2

Parameter Explanation

P1 - Variable-name

P2 - Value that will be assigned to the variable. If P2 is omitted,
the variable will be set to empty.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*********************************
' VAB.
'*********************************
'DBM.2
'This will clear the variable:
VAB.$$ABC=

' This will assign the variable $$ABC the
' string:
: $$AAA=Hans
: $$AAB=Peter
' The following specialfolder will not
' be resolved using VAB.
VAB.$$ABC=?path\$$AAA$$AAB.txt
MBX.$$ABC
END.
'*********************************
```

Remarks

Its important to understand that the VAB.-Command will not resolve Special-Folders and Special Variables, while it will resolve any other local or global variable names.

Limitations:

-

See also:

- VAB. - Variable Binary 3544
- VBA. - Variable Binary Append 3569
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.13 VAC. - Variable reCt

MiniRobotLanguage (MRL)

VAC. Command

Variable rectangular

Intention

This command can be used to assign Rectangular Coordinates to a Variable.
It is possible to use calculations if they are enclosed in brackets.

Variables of type RECT are currently not used in the engine, but may be used in future versions. If you use a RECT variable, the variable will actually most often just be converted into a text-variable. You can assign an zero RECT by just leaving the data-part empty:

```
VAC.$$REC=
DBP.$$REC
END.
' Will assign 0,0,0,0 as RECT to $$REC
```

Syntax

VAC.P1=P2,P3,P4,P5

Parameter Explanation

Rectangular Coordinates (**Rect**) are defined as (P2 to P5):

P1 - Variable-name

P2 - Value: left X coordinate

P3 - Value: top Y coordinate

P4 - Value: right X coordinate

P5 - Value: bottom Y coordinate

All coordinates (**P2** - **P5**) can contain variables, system-variables and formulas in (..)

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*********************************
' VAC.
'*********************************
VAR.§§IVA=1
VAP.$$MYP=(100+§§IVA*2),(-13+(2*§§IVA))
VAC.§§KOR=(100+§§IVA*2),(-13+(2*§§IVA)),$$MYP
DBP.§§KOR
END.
'*********************************
```

Remarks

-

Limitations:

-

See also:

- VAB. - Variable Binary 3544
- VBA. - Variable Binary Append 3569
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.14 VAF. - Variable Filename

MiniRobotLanguage (MRL)

VAF. Command

VAriable with Filename

Intention

This command can be used to assign a Filename to a variable.

```
'*********************************
' VAF.
'*********************************
VAF.$$FIL=Hallo
DBP.$$FIL
END.
'*********************************
```

Important: If the variable does not contain any path, `?path\` is automatically added to the variable content, to make it a valid path.

The `VAF.`- command makes a special folder, system-variables and variable-resolution. It will add `?path\` if there is no valid path given on the value-side. It will not resolve formulas.

Syntax

VAF.P1=[P2]

Parameter Explanation

P1 - `Variable-name`

P2 - ***(optional)*** `will be assigned to the variable. If P2 is omitted,`
`the variable will be set to empty.`

Speed in Ticks:

This command uses typically around 108 Ticks.

Example

```
'This will clear the variable:
VAF.$$ABC=
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' and systemvariables:
VAF.$$ABC=?path\$$AAA.txt
VAF.$$ABC=Hallo
```

imitations:

-

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.15 VAI. - Variable Indirekt

MiniRobotLanguage (MRL)

VAI. Command

VAriable Indirekt

Intention

This command can be used to assign a Value to a Variable which Name is stored inside another Variable.
The purpose of this command is to be able, to tell a Subprogramm that is called with `GSB.` or `JNF.` etc.
where to put the result of the Operation. Also it can be used to emulate Arrays of Variables from 0 to 99.
For example $$H00 to $$H99.

Here is an example.

```
' Variable gets the Name of another Variable.
' Always use VAB. for that!
VAB.$$OUT=$$INA
' Now here comes VAI.
VAI.$$OUT=1232
' Look what we got.
MBX.$$INA


' This Sample shows what its really good for.
VAB.$$INB=$$INC

GSB.CommandX|$$INB
MBX.$$INC

GSB.CommandX|$$IND
MBX.$$IND

ENR.
'============================================================

:CommandX
VAO.$$TXT=$$_01
DBP.$$TXT
VAI.$$TXT=123
RET.
```

Here is another example that shows how you can write to Variables `$$H01` to `$$H99` in a loop using `VAI`.

```
FOR.$$LOP|1|99
  VAN.$$CNT=$$LOP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    FMT.$$CNT|00
    VAR.$$HLT=$$H$$CNT
    VAI.$$HLT=$$LOP
  NEX.
```

Syntax

VAI.P1=P2

Parameter Explanation

P1 – Variable-name of a Variable that contains the name of another Variable.

P2 – Value that will be assigned to the variable that is inside the P1-Variable. If P2 is omitted,
the variable will be set to empty.

Example

```
'*********************************
' VAI.
'*********************************
  ' This Sample shows what its really good for.
  VAB.$$INB=$$INC
  GSB.CommandX|$$INB
  MBX.$$INC

  ' $$IND was not defined before
  GSB.CommandX|$$IND
  MBX.$$IND

  ENR.
  '============================================================

  :CommandX
  ' Important to use VAO. here!
  VAO.$$TXT=$$_01
  DBP.$$TXT
  ' Here comes VAI.
  VAI.$$TXT=123
  RET.
```

Remarks

Use `VAI.` together with `VAB.` and `VAO.`

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 2.1 : - Definiton-Command 361
- VBA. - Variable Binary Append 3569
- OPT. - Optional Settings 2406
- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- CAL. - mathematical CALculation 2315

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.16 VAN. - Variable Numeric

MiniRobotLanguage (MRL)

VAN. Command

Variable numeric

Intention

This command can be used to assign a numeric Value or a formula in ( ) to a Variable.
Specialfolders 345 are not resolved. Systemvariables 334 are resolved.

```
'This will set the variable to zero:
VAN.$$ABC=0
```

You can assign **Hexadezimal**, **octal** or **binary, floating point** and **negative** numbers to a variable using `VAN`.

```
' Hexadezimal assignement
VAN.$$ABC=&h90

' Binary assignement
VAN.$$ABC=&B0100101101

' Octal assignement
VAN.$$ABC=&O4574514
```

You can assign the result of a **formula** to a variable using `VAN`.

```
' This will assign the result to the variable
VAN.$$ABC=(4*5.1+(8*8))

' Negative numbers and floating point numbers
VAN.$$ABC=-25.004*

' use System-variables
VAN.$$XPO=#iw#*2
```

Syntax

VAN . P1=P2

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 **-** Variable-name

P2 **-** Value/formula in (..) that will be assigned to the variable.
If P2 is omitted, the variable will be set to empty.

Speed in Ticks:
This command uses typically around 90 Ticks.

Example

```
'*********************************
' VAN.
'*********************************
'This will set the variable to zero:
VAN.$$ABC=0

' This will assign the result to the variable
VAN.$$ABC=(4*5.1+(8*8))
VAN.$$ABC=-25.004
' Hexadezimal assignement
VAN.$$ABC=&h90
' Binary assignement
VAN.$$ABC=&B0100101101
' Octal assignement
VAN.$$ABC=&O4574514

END.
'*********************************
```

Remarks

-

Limitations:

Binary, octal and hexadezimal numbers can not be negative and they can not be floating point numbers.

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564

© 2023 Theo Gottwald * https://www.smart-package.com

- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.17 VAO. - Variable Once / Variable Clear

MiniRobotLanguage (MRL)

VAO. Command

Variable Once / Variable Clear

Intention

This command can be used to assign a Value to a Variable and iut can be used to completely delete a Variable (Variable Clear).

It will resolve only 1 variable that can be given on the value side, that is the first variable name on the value side.

The VAO.-command does **not** makes a special folder and system-variables resolution.

The VAO.-command can be used to assign Variables to other Variables, copying the content and preserving the Variable-Type.
For this on the Target-Side **P2** must be just a Variable.
Here are two examples for this.

```
   ' Copy Variable Content from one Variable to another
VAO.$$TGA=§§_01
VAO.$$TGA=§§ABC
```

To delete a Variable from Memory you can type:

```
VAO.$$TGA
```

or

```
VAO.$$TGA=
```

Both will do exactly the same. Unlike with other Commands, this will not only clear the content of a variable, but delete the variable completely from memory. This is sometimes necessary. For example when you want to use the variable to get a result from a MACRO-Function.
When using the second form like this:

```
VAO.$$TGA=
```

Make sure that there are no spaces on the right side of the = sign.
Otherwise the Variable will just be assigned those spaces.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

VAO.P1[=P2]

Parameter Explanation

P1 - Variable-name

P2 - Value that will be assigned to the variable. If **P2** is omitted, the variable will be set to empty. If **P2** is a Variable, then VAO. will just copy the content of **P1** into **P2**, this will preserve the original Type (Numeric, string etc.) of the **P2** - Variable.

Example

```
'*********************************
' VAO.
'*********************************
'DBM.2

'This will clear the variable:
VAO.$$ABC=

' This will assign the variable $$ABC the
' string:
' ?path\Hans$$AAB
: $$AAA=Hans
: $$AAB=Peter
VAO.$$ABC=?path\$$AAA$$AAB.txt

:over
MBX.Halt to read output

END.

'*********************************
' This example shows that VAO. can preserve
' the Variable-Type of an Variable
'*********************************
VAN.$$PAA=5.56
VAN.$$PAB=7
GSB.Add|$$PAA|$$PAB
MBX.Result is: $$RES

VAR.$$PAA=Max
VAR.$$PAB=Müller
GSB.Add|$$PAA|$$PAB
MBX.Result is: $$RES

END.
:Add
VAO.$$TGA=§$_01
GVT.$$TGA|§$TYP
DBP. Vartype is §$TYP.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
CAL.§§TYA=(§§TYP=2)+(§§TYP=4)+(§§TYP=5)
IVV.§§TYP=1
  $$RES=$$PAA,$$PAB
EIF.
IVV.§§TYA=1
  CAL.$$RES=$$PAA+$$PAB
EIF.
RET.
ENR.

'*********************************
' This example shows that VAO. and VAI.
' to get the result of a Macro-Function
'*********************************
VAN.$$PAA=5.56
VAN.$$PAB=7
VAO.$REA
%Add $$PAA|$$PAB|$$REA
MBX.Result is: $$REA
VAR.$$PAA=Max
VAR.$$PAB=Müller
VAO.$$REA
%Add $$PAA|$$PAB|$$REA
MBX.Result is: $$REA
END.
'------------------------------------------------------------
: %Add 3
GSB.Add|§§§01|§§§02|§§§03
END%

:Add
SAV.Save|$$ERG|$$TGA|$$TYP|$$PAA|$$PAB
VAR.$$PAA=§§_01
VAR.$$PAB=§§_02
VAO.$$ERG=§§_03
DBP.ERG=$$ERG
VAO.$$TGA=§§_01
GVT.$$TGA|$$TYP
DBP. Vartype is $$TYP.
CAL.$$TYA=($$TYP=2)+($$TYP=4)+($$TYP=5)
IVV.$$TYP=1
  $$RES=$$PAA,$$PAB
EIF.
IVV.$$TYA=1
  CAL.$$RES=$$PAA+$$PAB
EIF.
VAI.$$ERG=$$RES
SAV.Restore
RET.

ENR.
```

Remarks

*© 2023 Theo Gottwald * https://www.smart-package.com*

Its important to understand that the VAO.-Command will not resolve Special-Folders and Special Variables, while it will resolve any other local or global variable names, but only the first one in the line.

Limitations:

-

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.18 VAP. - Variable Point

VAP. - Variable-Point Previous 3556 Top 41 Next 3562

MiniRobotLanguage (MRL)

VAP. Command

VAriable Point

Intention

This command can be used to assign Point-Coordinates to a Variable-name.

It is possible to use calculations if they are enclosed in brackets.

Point Coordinate-variables can be used in most Mouse-Actions,
like `MPO., MMV., MLC.` and many more etc.

If you want to assign a zero coordinate, just leave the data part empty:

```
VAP.$$POI=
DBP.$$POI
END.
' Will assigne 0,0 to the variable $$POI
```

Syntax

VAP.[P1]=[P2,P3]

Parameter Explanation

P1 - Variable-name

P2 - X- Coordninate

P3 - Y- Coordninate

Speed in Ticks:

This command uses typically between 90 to 200 Ticks.

Example

```
'****************************************
'  VAP. - Sample
'****************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAP.$$MYP=12,13
MMV.$$MYP
' We make 4 mousemoves
FOR.§§IVA|1|200|50
VAP.§§KOR=(100+§§IVA*2),(-13+(2*§§IVA))
PAU.1
' relative mousemoves
MMV.$$MYP
MMV.r|§§KOR
NEX.
END.
```

Remarks

-

Limitations:

-

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.19 VAR. - Variable Set Value/Clear

MiniRobotLanguage (MRL)

VAR. Command

VARiable set value or clear

Intention

This command can be used to assign a Value to a Variable.

There are three forms, please look at the examples.
The VAR.-command makes a complete variable-resolution, this includes any special-folder or system-variables on the value-side.

Therefore it may be a bit less efficient then the VAB. command, which does not make a variable resolution.

There is an "Implicit VAR."-Command, from the preprocessor, if you just omit the VAR. and write:

```
$$AAA=12
$$BBB=99
```

In this case, the Preprocessor will just add the "VAR." in front of the Line for you. Therefore at runtime the Line would look like this:

```
VAR.$$AAA=12
VAR.$$BBB=99
```

To get a complete resolution of the target side, including formulas, you can use the 2.1 : - Definiton-Command 361.

Syntax

VAR.[P1]=[P2]

Parameter Explanation

P1 - Variable-name

P2 - Value that will be assigned to the variable. If P2 is omitted,
the variable will be set to empty.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'This will clear the variable:
VAR.$$ABC=

' This will assign the variable a value or string and replace
all special-folders
' and systemvariables:
VAR.$$ABC=Hallo

' This will assign a Value without Specialfolder-Processing
 : $$ABC=Hallo
```

Remarks

-

Limitations:

-

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.20 VAS. - Variable with String

VAS. - Variable with String Previous 3562 Top 41 Next 3566

MiniRobotLanguage (MRL)

VAS. Command

VAriable with String

Intention

This command can be used to assign a String/text value to a variable.

The VAS.-command makes a special folder and system-variables and variable-resolution.
Unlike VAF. it will not add anything to the string to make it a valid path.

Syntax

VAS.P1=[P2]

Parameter Explanation

P1 - Variable-name

P2 - ***(optional)*** will be assigned to the variable. If P2 is omitted,
the variable will be set to empty.

Example

```
'This will clear the variable:
VAS.$$ABC=

' This will assign the variable string $$AAA
' no matter, if $$AAA is a valid Variable or not.
' it would replace all special-folders
' and systemvariables:
VAS.$$ABC=?path\$$AAA.txt
VAS.$$ABC=Hallo
```

imitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.21 VAV. - Variables with Variables

MiniRobotLanguage (MRL)

VAV. Command

VAriables with Variables

Intention

This command can be used to assign a Value to a Variable.

It will resolve variables that are with a given variable up to any Level.

The normal `VAR.` command does not touch already resolved variable bytes, to be binary compatible to some degree, and to avoid dead-locks.

`VAV.` will also resolve special folders and variables to `MaxVarRes` which is 255 by default. `MaxVarRes` is set using the `OPT.`-command). Lets see an example:

```
'*********************************
' VAR. / VAV. comparison
'*********************************
VAB.$$TXA=Theo
VAB.$$TXB=$$TXA
VAB.$$TXC=$$TXB

VAV.$$TXD=$$TXC
DBV.$$TXD
' VAV. returns "Theo"

VAR.$$TXD=$$TXC
DBV.$$TXD
' VAR. returns "$$TXB"

END.
'*********************************
```

As you can see, `VAR.` does not resolve to any depth, while `VAV.` will.

If there are variables that will resolve to variables which resolve to variables in a circular way, `VAV.` may resolve again and again until `MaxDepth` is reached. See this example:

```
'*********************************
' VAR. / VAV.
'*********************************
'
' We set maximum resolution depth for variables
' to 255 (which is the default value)
OPT.MaxVarRes|255
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAB.$$TXA=Theo $$TXB
VAB.$$TXB=$$TXA

VAV.$$TXD=$$TXB
DBV.$$TXD
' VAV. returns "Theo Theo Theo Theo Theo Theo Theo Theo" etc.

VAR.$$TXD=$$TXB
DBV.$$TXD
' VAR. returns "$$TXB"

END.
'*********************************
```

Please see the difference in the example below.

Syntax

VAV.[P1]=[P2]

Parameter Explanation

P1 - Variable-name

P2 - Value that will be assigned to the variable. If P2 is omitted,
the variable will be set to empty.

Example

```
'*********************************
' VAV.
'*********************************
VAB.$$TXA=Theo
VAB.$$TXB=$$TXA
VAB.$$TXC=$$TXB
VAV.$$TXD=$$TXC
DBV.$$TXD
END.
VAR.$$ABC=->$$AAA
MBX.$$ABC
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Its important to understand that the VAV.-Command should not be used for binary content which is for example loaded using CFF. Use VAB. or VAO or even VAR. instead.

Limitations:

-

See also:

- 2.1 : - Definiton-Command 361
- VBA. - Variable Binary Append 3569
- OPT. - Optional Settings 2406
- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- CAL. - mathematical CALculation 2315

© 2023 Theo Gottwald * https://www.smart-package.com

3.81.22 VBA. - Variable Binary Append

MiniRobotLanguage (MRL)

VBA. Command

Variable binary

Intention

This command can be used to append up to 21 variables into variable **P1**.
Normally you would think of doing an variable-append like this:

```
VAV.$$ALL=$$T01$$T02$$T03 .... etc.
```

This is going to work, as long as these variables do not contain any binary data that could resolve to special folders, variable or system-variables.

If that is the case, `VAV.` and such commands would try to resolve these binary patterns and possibly do unwanted changes to your data.

This is where `VBA.` enters the scene. `VBA.` will just resolve each parameter one time!
That is called "binary compatible variable resolution". BEcause the variable will be resolved, but the binary data is not going to be changed.

Then `VBA.` will append all those strings into **P1**.

If **P1** was not empty, the content of P1 will will also be part of the new string.
If you do not want this, clear P1 before calling this command.

There are few thing to watch out for:

1. **P1** will also be appended
2. You need to specify a variable for each parameter. It must be a variable.
   Additional characters which are not part of a variable name will have the command run
   into an (timeout) error.
3. You can specify up to 21 parameters, but you need to specify at least two parameters.

How do i clear a variable P1 before using VBA. ?

As `VAB.` does not go through any variable-resolution process, its possibly the fastest way to clear a variable.

```
' This will clear a variable
' Be sure that you do not have a SPACE behind the = sign.
VAB.$$TXT=
```


© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

VBA.P1|P2[|P3] ...[|PX]

Parameter Explanation

P1 **-** Variable-name of Source and Target variable

P2 **-** Variable which will be resolved and then appended to P1

P3 ... Px **-** ***(optional)*** more variables that will be resolved (binary comaptible) and then appended to **P1**.

Example

```
'**********************************
' VBA.
'**********************************
: $$T01=Hallo
: $$T02=$sp$
: $$T03=Rolf
VBA.$$T01|$$T02|$$T03
DBV.$$T01
ENR.
'**********************************
```

Remarks

-

Limitations:

The overall maximum length of variables depend on your available RAM and your actual Memory fragmentation. String memory needs to be in one piece and can not be fragmented. As a raw assumption, 500 MB maximum size for strings should always be possible. If you stay below you should be on the safe side. The theoretical maximum is about ~2 GB.

See also:

- VAB. - Variable Binary 3544
- VAC. - Variable reCt 3546
- VAN. - Variable Numeric 3553

© 2023 Theo Gottwald * https://www.smart-package.com

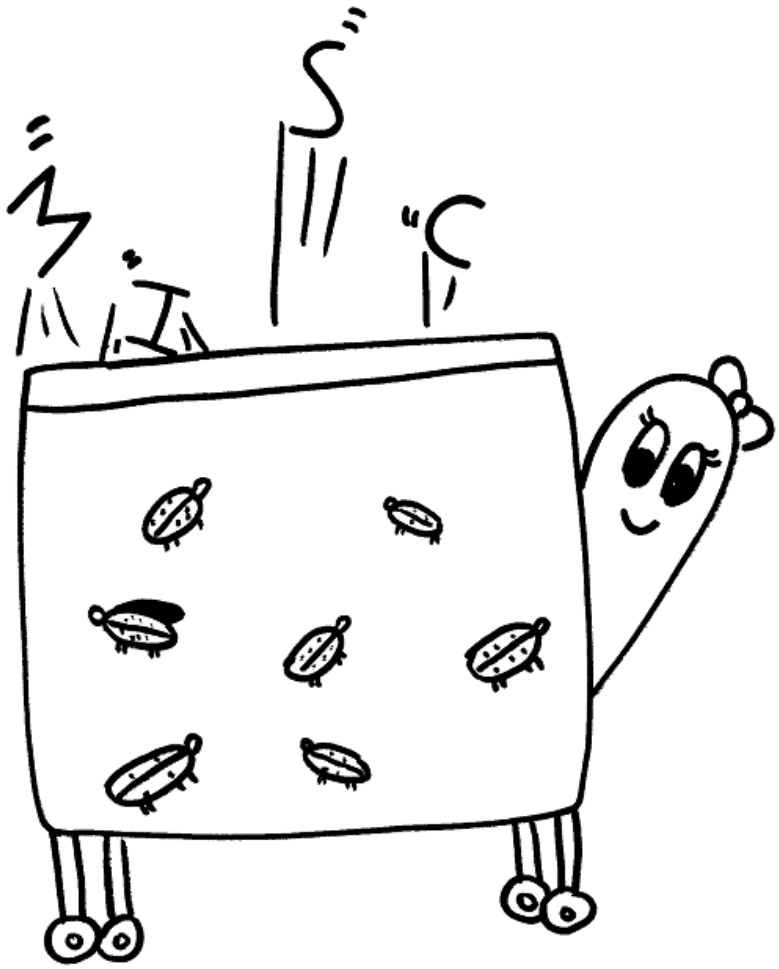
- VAO. - Variable Once 3556
- VAP: - Variable Point 3560
- VAR. - Variable Set Value/Clear 3562
- VAS. - Variable with String 3564
- VAV. - Variables with Variables 3566
- VBA. - Variable Binary Append 3569
- CAL. - mathematical CALculation 2315
- 2.1 : - Definiton-Command 361

© 2023 Theo Gottwald * https://www.smart-package.com

3.82 VB-Script

3.82.1 !Smarty's VB-Script Support

Smarty's VB-Script-Support



In Wikibooks there is an Introduction to VBS:
https://www.de.wikibooks.org/wiki/Visual_Basic_Script_(VBS):_Einf%C3%BChrung

See also here:
WSH-Reference

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
An A-Z Index of Windows VBScript commands

   Abs(number)      Absolute (positive) value of number
  .AppActivate      Activate running command
  .Application      BrowseForFolder/Open
   Array(el1,el2,el3)   Add values to an Array variable
   Arguments        Command line arguments
   Asc(String)      Return ASCII code for string
   AscW(String)     Return Unicode code for string
  .BrowseForFolder Prompt the user to select a folder
C
   Call subroutine (arguments) or  subroutine arguments
   CBool(expression)  Convert expression to Boolean (True/False) •
   CByte(expression)  Convert expression to Byte (0-255) •
   CCur(expression)   Convert expression to Currency (numeric) •
   CDate(expression)  Convert expression to Date •
   CDbl(expression)   Convert expression to Double (up to 1.79x10308) •
   Chr(ChrCode)       Return the string character for ChrCode (ASCII code)
   ChrB(ChrCode)      Return the string character for ChrCode (Byte code)
   ChrW(ChrCode)      Return the string character for ChrCode (Unicode/DBCS)
   CInt(expression)   Convert expression to Integer •
   CLng(expression)   Convert expression to long •
   CSng(expression)   Convert expression to single •
   CStr(expression)   Convert expression to a string •
  .CreateObject     Create an automation object / run an external command
  .CreateShortcut   Create Shortcut
   cscript          Run a VBScript .vbs file
  .CurrentDirectory Retrieve or change the current directory
D
   Date()          The current system date
   DateAdd         Add a time interval to a Date
   DateDiff        Return the time interval between two dates
   DatePart        Return part of a given date
   DateSerial      Return a Date from a numeric Year, Month and Day
   DateValue       Return a Date from a string expression
   Day(date)       Return the day component of date (1-31)
   Dim             Declare a new variable or array variable
   Do..Loop        Repeat a block of statements
   Drives
   Drive Map      .MapNetworkDrive - .Enum - .Remove
E
  .Echo            Echo text to screen, also StdOut.Write /.Popup
   End             End a procedure or code block
   Environment Variables  - Expand - .Delete/Remove
   Escape(String) Return only ASCII characters from a Unicode string.
   Eval(expr)      Evaluate an expression
  .Exec            Run a command
   Execute         Execute one or more  statements
   Exp(n)          Return e (base of natural logs) raised to a power n.
   Exit            Exit a block of code immediately
F
   FileSystemObject Work with Drives, Folders and Files
   Filter          Produce an array by filtering an existing array
   Fix(number)     Return the integer portion of a number
   For...Next      Repeat a block of statements a given number of times
   For Each...     Loop through the items in a collection or array
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   FormatCurrency Format a number with a currency symbol
   FormatNumber   Format a number
   FormatPercent  Format a number with a % symbol
   FormatDateTime Format a Date/Time value
   Function       Define a function procedure
G
   GetLocale()    Return the Regional LocaleID
  .GetObject      Get an Automation object
H
   Hex(number)    Return the hex value of number
   Hour(time)     Return the hour component of time
I
   If..Then       Conditionally execute a block of statements
   InputBox       Prompt for user input
   InStr          Find one string within another
   InStrRev       Find one string within another, starting from the end
   Int(number)    Return the integer portion of a number
   IsArray(varname)      Is varname an array?
   IsDate(expression)    Is expression a DateTime?
   IsEmpty(expression)   Is expression initialised?
   IsNull(expression)    Is expression NULL?
   IsNumeric(expression) Is expression a Numeric?
   IsObject(expression)  Is expression an object?
   Join           Combine the contents of an array into a single variable.
L
   LBound             Return the smallest subscript for an array.
   LCase(String)      Return String in lower case
   Left(String,len)   Return the leftmost len characters of string
   LeftB(String,len)  Return the leftmost len bytes of string
   Len(String)    Return length of string in characters
   LenB(String)   Return length of string in Bytes
   Log(number)    Return natural log of number
  .LogEvent       Log an item in the Event log
   LTrim(String)  Remove any leading spaces from a string expression
M
  .MapNetworkDrive Drive Map
   Mid            Return a mid-section from a string
   Minute(time)   Return the minute component of time
   Month(date)    Return the month component of date
   MonthName      Convert a month number to a descriptive Month
   MsgBox         Display a dialogue box message
N
  .Network        Access network resources
   Now            Return the current Date and Time
O
   On Error       Error handling
   Option Explicit   Force all variables to be defined
P
  .AddPrinterConn / .AddWindowsPrConn Add Printer connection
  .EnumPrinterConnections   List Printer connections
  .SetDefaultPrinter        Set default printer
  .RemovePrinterConnection  Remove printer connection
   Private VarName          Declare a local variable/array variable
   Public VarName           Declare a public variable/array variable
Q
  .Quit            Quit
R
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
   Randomize(number)   Initialise the random number generator
  .ReadLine        Accept user text input
   ReDim           Resize a dynamic array
   RegExp          Regular expression search object
  .RegDelete       Registry, delete
  .RegRead         Registry, read
  .RegWrite        Registry, write
   REM             Add a Comment
   Replace         Find and replace text
   RGB(Red,Green,Blue) Return a system color code
   Right(String,len)   Return the rightmost len characters of string
   RightB(String,len)  Return the rightmost len bytes of string
   Rnd             Return a random number
   Round           Round a number to n decimal places
   RTrim(String)   Remove any trailing spaces from a string expression
  .Run a command
S
   Second(time)    Return the second component of time
   Select...Case   Conditionally execute a block of statements
  .SendKeys
   Set variable = object   Assign an object reference
   SetLocale(LocaleID)     Set the Regional LocaleID
   Sgn(number)             The sign of a number
  .Shell + Shell.Application objects/methods
  .ShellExecute    Run a script/application in the Windows Shell
   Shortcut create .CreateShortcut
  .Sleep
   Space           Return a string consisting of spaces.
  .SpecialFolders Path to Desktop,Start menu, Programs...
   Split           Parse a string of delimited values into an array
   Sqr(number)     Square root
   StrComp         Compare two strings
   String          Create a string with a single character repeated
   StrReverse      Reverse a string
   Sub             Declare a sub procedure
T
   Time()          The current system time
   Timer()         The number of seconds since midnight
   TimeSerial      Construct a Time value from Hours, Minutes and seconds
   TimeValue       Convert a string to a Time value
   Trim(String)    Remove leading and trailing spaces from string
   TypeName        Return variable type (variant subtype).
U
   UBound          Return the largest subscript for an array dimension
   UCase(String)   Uppercase String
   UnEscape(Str)   Return Unicode characters from an escaped ASCII string
W
   Weekday         Return the day of the week (1-7)
   WeekdayName     Return the named day of the week
   While...Wend    Conditionally repeat a block of statements
   With            Assign multiple properties of an object
   Year(date)      Return the year component of date (1-12)
   ' Remark        Add a Comment

 All VBScript variables are variants (general Datatype for Numbers, Strings and
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Sample-Usage:
-------------

 Dim a
 Set a = CreateObject("WScript.Shell")
 a.SendKeys "{Enter}"


 Key               Code
 ==========================
 BACKSPACE ----- {BACKSPACE}, {BS}, or {BKSP}
 BREAK --------- {BREAK}
 CAPS LOCK ----- {CAPSLOCK}
 DEL or DELETE - {DELETE} or {DEL}
 DOWN ARROW ---- {DOWN}
 END ----------- {END}
 ENTER --------- {ENTER}or ~
 ESC ----------- {ESC}
 CTRL ---------- ^
 ALT ----------- %
 HELP ---------- {HELP}
 HOME ---------- {HOME}
 INS or INSERT - {INSERT} or {INS}
 LEFT ARROW ---- {LEFT}
 NUM LOCK ------ {NUMLOCK}
 PAGE DOWN ----- {PGDN}
 PAGE UP ------- {PGUP}
 PRINT SCREEN -- {PRTSC}
 RIGHT ARROW --- {RIGHT}
 SCROLL LOCK --- {SCROLLLOCK}
 TAB ----------- {TAB}
 UP ARROW ------ {UP}
 F1 ------------ {F1}
 F2 ------------ {F2}
 F3 ------------ {F3}
 F4 ------------ {F4}
 F5 ------------ {F5}
 F6 ------------ {F6}
 F7 ------------ {F7}
 F8 ------------ {F8}
 F9 ------------ {F9}
 F10 ----------- {F10}
 F11 ----------- {F11}
 F12 ----------- {F12}
 F13 ----------- {F13}
 F14 ----------- {F14}
 F15 ----------- {F15}
 F16 ----------- {F16}
```

Sample-MACRO:

```
'Sample:
STW.ct|ConsoleWindowClass|Eingabeaufforderung
MLI.
VAR.$$NAM=DIR{ENTER}
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
%VB_Sendkeys $$NAM
ENR.
'-------------------------------------
: %VB_Sendkeys
' Use TOS
#IF PARAMS=0
GSB.VB_Sendkeys|$tos$
#EIF
#IF PARAMS=1
GSB.VB_Sendkeys|$$$01
#EIF
END%
'-------------------------------------
' P1 - Zu sendender Text
:VB_Sendkeys
SAV.Save|$$NAM
IVV.$$_00|=|1
  $$NAM=$$_01
ELS.
  $$NAM=Hallo Welt!{ENTER}
EIF.
' Auflösung da VBS. nur vorsichtig, binärkompatibel auflöst.
VAR.$$NAM=$$NAM

VBS.VB|Test|$$NAM
Option Explicit
  Function Test(x)
  Dim a
  Set a = CreateObject("WScript.Shell")
  a.SendKeys x(0)
  End Function
VBE.
SAV.Restore
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.82.2 VBE. - VB-Script End

MiniRobotLanguage (MRL)

VBE. Command

Visual Basic Script End

Intention

This command is used to end a block with VB-Script. It is always used paired with `VBS.`
The statements between a `VBS.` and a `VBE.` are ignored by the SPR-Preprocessor. They are also ignored by the "Pretty-Printer" in the Editor.

`VBS.` and `VBE.` enable you, to include a VB-Script or Java-Script Code inline into the SPR-Script. You can use it to check for System-Objects (See Sample 2 below), or you can use it to call WMI-Functions.

You can also transfer Variable-Content into `VB-Script`-Variables (see example 1 below). And use them in your VB-Script.

And with `VBE.` you can get a single result value from the VB-Script back into your SPR-Script (see example 2 below).

Syntax

```
VBS.P1|P2[|P3...|Px]
...
VBE.P1
```

Parameter Explanation for VBE

P1 **-** `(out) - will hold to the result of the VB-Script Function that was called before.`

Example

```
'*******************************************
' VBS./VBE. - Sample
' You will get two Messageboxes.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
' One from inside VB-Script, one from MR.
'*******************************************
DBP.This is a MiniRobots-Script.
' Now starts the VB-Script-Part.
' IMPORTANT:
' The MSGBOX from VB is not TopMost and may be
' hidden behind the EDITOR!
' Keep Editor on the Right side,
' and Keep center of the screen free.
VBS.VB|Multiply|123|124
  Function Multiply(vPrms)
  Dim result
  result = vPrms(0) * vPrms(1)
  msgbox vPrms(0) * vPrms(1),,"from Inside VB-Script"
  Multiply = result
  End Function
VBE.$$RET
MBX.Result: $$RET
END.


'*******************************************
' VBS./VBE. - Sample 2
' Test if CDO is available
'*******************************************
DBP.Hallo
VBS.VB|Multiply|123|124
  Function Multiply(vPrms)
  Dim c
  Dim msg
  On Error Resume Next
  set c = CreateObject("MAPI.Session")
  If Err <> 0 then
  msg = "CDO ist auf diesem System nicht installiert."
  else
  msg = "CDO-Version " & c.Version & " steht zur Verfügung."
  end if
  msgbox msg
  Multiply = msg
  End Function
VBE.$$RET
MBX.Result: $$RET|from SPR
END.
```

Remarks

The VBS.-/JS.-Scripts will not be written to the harddisk prior to execution.
Therefore they are permanently crypted together
with the SPR-Script using the `'#CRY:`-Option.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

It is possible to start the VB-Scripts from inside `PRR./PRE`.
There are no Limitations about Multithreading here.

See also:

- VBS. - VB-Script Start

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.82.3 VBS. - VB-Script Start

MiniRobotLanguage (MRL)

VBS. Command

Visual Basic Script Start

Intention

This command is used to start a block with VB-Script. It is always used paired with VBE.

The statements between a VBS. and a VBE. are ignored by the SPR-Preprocessor. They are also ignored by the "Pretty-Printer" in the Editor.

VBS. and VBE. enable you, to include a VB-Script or Java-Script Code inline into the SPR-Script. You can use it to check for System-Objects (See Sample 2 below), or you can use it to call WMI-Functions.

You can also transfer Variable-Content into VB-Script-Variables (see example 1 below). And use them in your VB-Script.

And with VBE. you can get a single result value from the VB-Script back into your SPR-Script (see example 2 below).

Syntax

```
VBS.P1|P2[|P3...|Px]
...
VBE.P1
```

Parameter Explanation for VBS

P1 - (in) - VB or JS - this way you can choose between VB-Script
and JScript.

P2 - (in) - Name of the first function to start, without (). The function must have one single parameter which is an ARRAY of Variant.

P3-Px - (in) - these are the Parameters to transfer into the VB-Script. They will be transfered into the VARIANT-ARRAY.
You can use up to 20 Parameters.

© 2023 Theo Gottwald * https://www.smart-package.com

P4 **-** Value*

P5 **-** (out) Variable for Success-flag 1/0.*

Parameter Explanation for VBE

P1 **-** (out) - will be set to the result of the VB-Script Function called before.

Example

```
'*******************************************
' VBS./VBE. - Sample
' You will get two Messageboxes.
' One from inside VB-Script, one from MR.
'*******************************************
DBP.This is a MiniRobots-Script.
' Now starts the VB-Script-Part.
' IMPORTANT:
' The MSGBOX from VB is not TopMost and may be
' hidden behind the EDITOR!
' Keep Editor on the Right side,
' and Keep center of the screen free.
VBS.VB|Multiply|123|124
  Function Multiply(vPrms)
  Dim result
  result = vPrms(0) * vPrms(1)
  msgbox vPrms(0) * vPrms(1),,"from Inside VB-Script"
  Multiply = result
  End Function
VBE.$$RET
MBX.Result: $$RET
END.


'*******************************************
' VBS./VBE. - Sample 2
' Test if CDO is available
'*******************************************
DBP.Hallo
VBS.VB|Multiply|123|124
  Function Multiply(vPrms)
  Dim c
  Dim msg
  On Error Resume Next
  set c = CreateObject("MAPI.Session")
  If Err <> 0 then
  msg = "CDO ist auf diesem System nicht installiert."
  else
  msg = "CDO-Version " & c.Version & " steht zur Verfügung."
  end if
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  msgbox msg
  Multiply = msg
  End Function
VBE.$$RET
MBX.Result: $$RET|from SPR
END.
```

Remarks

The VBS.-/JS.-Scripts will not be written to the harddisk prior to execution.
Therefore they are permanently crypted together
with the SPR-Script using the `'#CRY:`-Option.

Limitations:

It is possible to start the VB-Scripts from inside `PRR./PRE`.
There are no Limitations about Multithreading here.

See also:

- VBE. - VB-Script End

© 2023 Theo Gottwald * https://www.smart-package.com

3.83 Window Operations

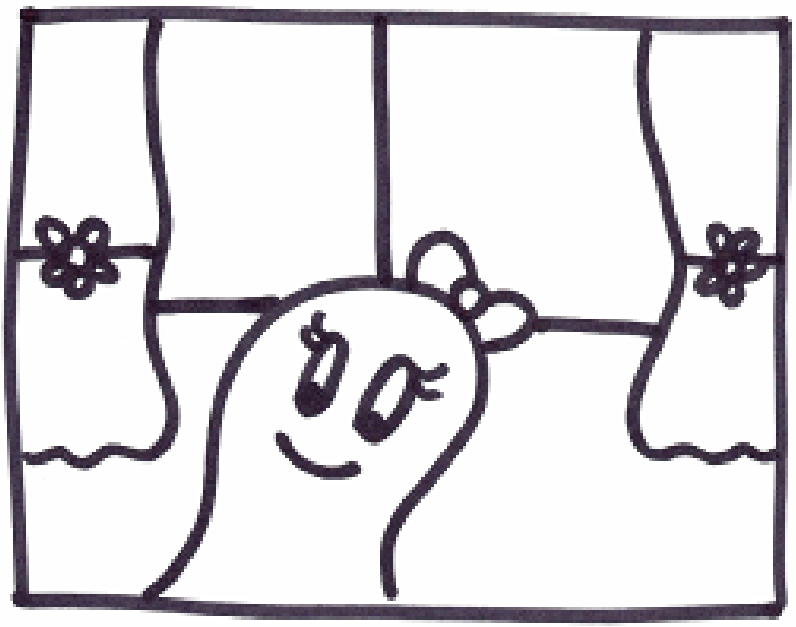
3.83.1 ! Smarty's Window Operations

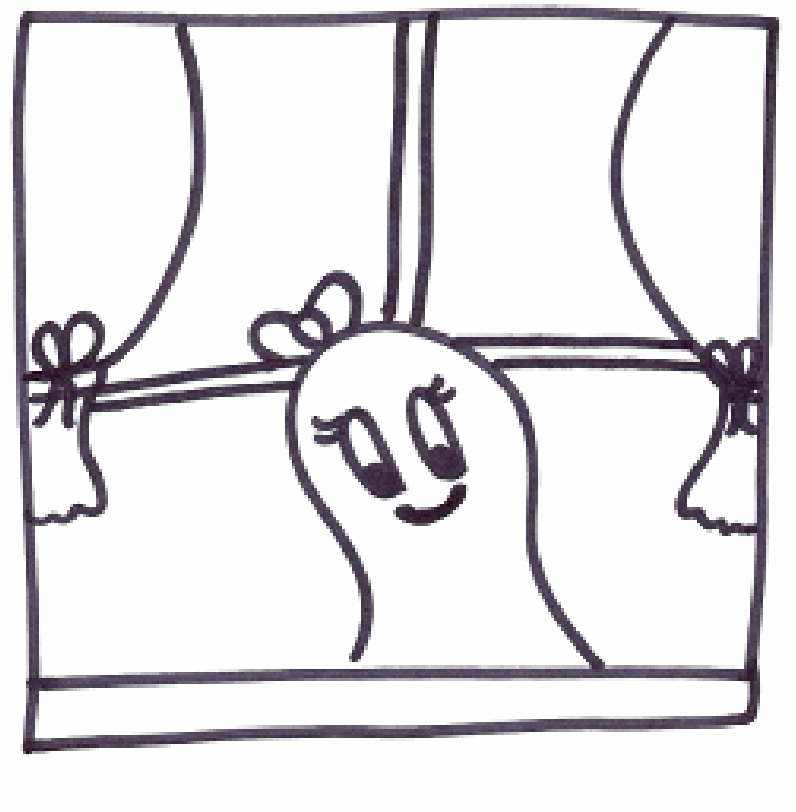
Smarty's Window Operations

Previous 3569 Top 41 Next 3586

Smarty's Window Operations

© 2023 Theo Gottwald * https://www.smart-package.com





© 2023 Theo Gottwald * https://www.smart-package.com

3.83.2 CAW. - Close-All-Windows

MiniRobotLanguage (MRL)

CAW. Command

Close all Windows

Intention

This command will close all windows that meet the given specification.

There are two modes, in which you can operate this command.
In Mode 1, you give just two parameters, the window-text, and the window-class.
Then all such windows will be **asked to close**.

In Mode 2, you use 3 parameters.The third parameters is the runtime in seconds.
In this mode, `CAW.` will be like a "Close New Windows" and close only Windows that appear NEW while it is running. You can also run it as a background robot using `PRR.` and `PRE.` See example 2.

Unlike most other window-commands, `CAW.` will only take Window-Text and Window-Class as parameters. `CAW.` has a "**deadlock-protection**" against windows which can not be closed.

Once it runs against such a window that can not be closed, it will stop execution.
Such windows are for example windows which have dialogs open or hang. In such cases you should use `CPR.` or `KPN.` to get rid of the hanging program.

Syntax

CAW.[P1][|P2][|P3]

Parameter Explanation

P1 - Windows-Class

P2 - Window-Text

P3 - Timeout in seconds if window can not be closed

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' CAW.-Demo
' In this Mode, all such windows will be closed.
'**********************************
'DBM.2
CAW.ieframe|private
: §§NUM=20
GSB.openem
PAU.10
CAW.ieframe
END.
MBX.!
END.

:openem
FOR.§§LOP|1|§§NUM
  EXS.iexplore|www.fa2.de
  PAU.1
NEX.
RET.
ENR.

'**********************************
' CAW.-Demo 2
' In this Mode only NEW appearing windows will be closed.
'**********************************
'DBM.2
CAW.ieframe|private
: §§NUM=2
GSB.openem
MBX. This IE will stay open after the CAW.$crlf$ because its
third parameter.
PAU.3
' Run CAW. as Parallel-Robot for 20 Seconds
PRR.
  PRT.Parallelrobot starting.
  CAW.ieframe|private|20
  PRT.Parallelrobot ending.
  END.
PRE.
: §§NUM=30
GSB.openem
MBX."Parallerobot CAW will close all NEW IE's until you click
me."
END.

:openem
FOR.§§LOP|1|§§NUM
  EXS.iexplore|www.fa2.de
  PAU.0.5
NEX.
RET.
```

Remarks

*© 2023 Theo Gottwald * https://www.smart-package.com*

`CAW`. is generally better then `KPN`. or `CPR`. because it will close the applications through the surface.

Limitations:

-

See also:

- CLW. - Close-Window 3592
- CTR. - Close-ThRead 2884
- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.3 CFW. - Copy-From-Window

CFW. - Copy from Window Previous 3586 Top 41 Next 3592

MiniRobotLanguage (MRL)

CFW. Command

Copy from Window

Intention

Reads the Text from the actual localized Window. If this is a Top-Window you get the Window-Title. If this is a child-window you may get something useful or not. It is faster then `GAI`. If `CFW`. fails, try `GAI`.

`CFW`. has four Modes of operation:

```
' No parameter
CFW.
```

Will copy the text from the actual window to the TOS.

```
CFW.$$TXT
```

Will just read the text from the actual window. In this Mode, CRLF, as well as leading and trailing spaces are filtered. Also all leading and trailing ASCII Characters below 33 are fitlered. This will not happen if you add the "u"-flag. Like this:

```
CFW.$$TXT|u
```

Will read the text "unfiltered".

```
CFW.|u
```

Will read the text "unfiltered" and put it on the TOS.

Syntax

CFW.[P1]

Parameter Explanation

`CFW`. works an most common controls.It does not work in any browser, use `GAI`. then.

P1 – Variable for Text which is read from the window. If omitted TOS is used.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'************************************
' CFW.-Sample
' Please run TestApp from SAmples Folder
'************************************

STS.CLEAR
STW.ct|PBWindowClass:0|TestApp
CFW.
STS.DUMP.
MBX. Got window title on TOS

STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Button|Button 2
CFW.$$TXT
PRT.Got Window-Text into Variable: $$TXT
' Clear Variable
VAR.$$TXT=

STW.ct|PBWindowClass:0|TestApp
SCW.nc|1|SysIPAddress32
CFW.$$TXT|u
PRT.Got Text from IP-Control: $$TXT

STS.CLEAR
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Edit
CFW.|u
STS.DUMP
MBX.Got Text from Edit-Control (unfiltered) on TOS.

STS.CLEAR
STW.ct|PBWindowClass:0|TestApp
SCW.nct|1|Edit
CFW.$$TXT
PRT.Got Text from Edit-Control, filtered:$crlf$+$$TXT
MBX.Got Text from Edit-Control filtered on Into VAR.

ENR.
'------------------------------------------------------------
```

Remarks

Uses the Windows-Engine.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- GAI. - Get-Acc-Information 591

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.4 CLW. - Close-Window

CLW. - Close Window **Previous** 3589 **Top** 41 **Next** 3619

MiniRobotLanguage (MRL)

CLW. Command

Close Window

Intention

This command will close the last localized window. If a Window-Handle is specified as parameter, this window will be closed. It will close **Top-windows or child-windows** via direct-engine 231.

Syntax

CLW.[P1]

Parameter Explanation

P1 – First Parameter is a Variable, local or Global. If omitted the actual window is been close.

Example

```
'************************************
' Example 1
'************************************
' Using CLW. without parameter
DBM.2
STW.t|TestApp
' The next line will close the Window.
CLW.
DMP.
END.

'************************************
' Example 2
'************************************
' Using CLW. using a window-handle as parameter
DBM.2
STW.t|TestApp
HTV.$$HWN
' You can work with other windows here
' We vlosethe window by its handle
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
CLW.$$HWN
DMP.
END.
```

Remarks

Limitations:

This command has no "compatible engine" equivalent.
It will always directly close the window. It does not need a mouse or a desktop.

See also:

- CTR. - Close-ThRead 2884
- CAW. - Close-All-Windows 3586
- CPR. - Close-PRocess 2882
- KPN. - Kill Process by Name 2904

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5 DTI.-Desktop & Mouse Commands

Geben Sie hier den Text ein.

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.1 DTI.Count Desktops

MiniRobotLanguage (MRL)

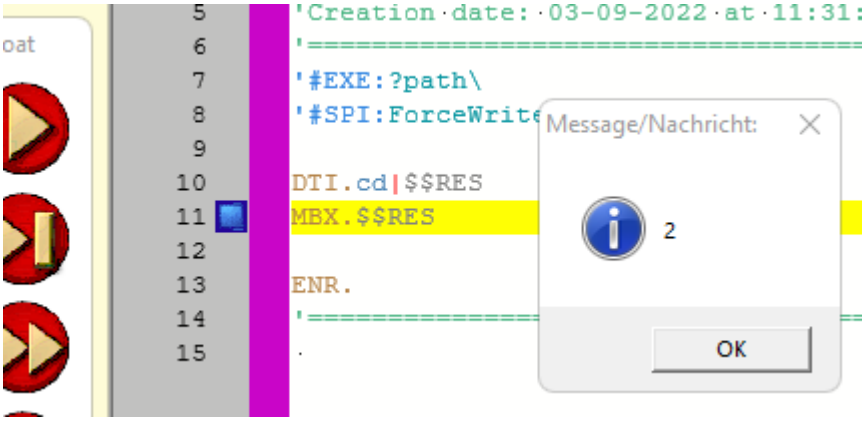
DTI.Count Desktops

Count Desktop-Screens (Monitors)

Intention

This command will find our how many "Monitors" (Desktops) are connected to your Computer.

```
DTI.cd|$$RES
MBX.$$RES
ENR.
```



Syntax

DTI.Count Desktops|[P1]

Parameter Explanation

P1 – (optional) Variable, local or Global. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd
DMP.6
MBX.!
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.2 DTI.DCoord

MiniRobotLanguage (MRL)

DTI.DCoord

Get Desktop-Screens Coordinates (Monitors)

Intention

This command will find our on which Coordinates "Monitors" (Desktops) starts (upper left corner).
If you have "Dual Screen", the result will look like that, because Desktop 2 is to the right of Desktop 1.

```
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.DC|$$NUA|$$HAA|$$WAA
  DBP.Desktop Positon of upper, left corner of Monitor $$NUA is at: $$HAA,$$WAA
NEX.
ENR.
```

```
[15:49:59] Position on 1 is  at: 0,0
[15:50:00] Position on 2 is  at: 4096,0
```

Syntax

DTI.DCoord|P1[|P2][|P3]

Parameter Explanation

P1 – Number of Desktop/Screen as given using `DTI.Count Desktops`.
P2 – (optional) Variable, will return the X-Coordinate of the upper left corner of this Desktop. If omitted the Result will be placed on TOS.
P3 – (optional) Variable, will return the Y-Coordinate of the upper left corner of this Desktop If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.DC|$$NUA|$$HAA|$$WAA
  DBP.Desktop Positon of upper, left corner of Monitor $$NUA is at: $$HAA,$$WAA
NEX.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.3 DTI.Depth

MiniRobotLanguage (MRL)

DTI.Depth

Count Desktop-Screens (Monitors)

Intention

This command will find our the Screen-Depth (how many Bits per Pixel) for each connected "Monitors" (Desktops).

```
' Get number of Desktops
DTI.cd|$$NUM
' Loop over all Desktops and get Depth
FOR.$$NUA|1|$$NUM
  DTI.dd|$$NUA|$$DEP
  DBP. Monitor $$NUA has a Depth of: $$DEP
NEX.
ENR.
```

```
[15:25:37] Monitor 1 has a Depth of: 32
[15:25:37] Monitor 2 has a Depth of: 32
```

Syntax

DTI.Depth|[P1]

Parameter Explanation

P1 – (optional) Variable, local or Global. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dd|$$NUA|$$DEP
  DBP. Monitor $$NUA has a Depth of: $$DEP
  DTI.df|$$NUA|$$FRE
  DBP. Monitor $$NUA has a Frequency of: $$FRE
NEX.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.4 DTI.Frequency

MiniRobotLanguage (MRL)

DTI.Frequency

Get Screen Refresh Frequency per Monitor

Intention

This command will find our how many "Monitors" (Desktops) are connected to your Computer.

```
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.df|$$NUA|$$FRE
  DBP. Monitor $$NUA has a Frequency of: $$FRE
NEX.
ENR.
```

Syntax

DTI.Frequency|P1[|P2]

Parameter Explanation

P1 – Desktop/Monitor number 1-based
P2 – (optional) Variable for the result, local or Global. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dd|$$NUA|$$DEP
  DBP. Monitor $$NUA has a Depth of: $$DEP
  DTI.df|$$NUA|$$FRE
  DBP. Monitor $$NUA has a Frequency of: $$FRE
NEX.
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.5 DTI.Height

MiniRobotLanguage (MRL)

DTI.Height

Get physical Height in Pixel of Desktop-Screens (Monitors)

Intention

This command will get the physical Height of each connected Monitor.

```
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DBP. Monitor $$NUA has a Height of: $$HEI
  DTI.dw|$$NUA|$$WID
  DBP. Monitor $$NUA has a Width of: $$WID
  DTI.sz|$$NUA|$$HAA|$$WAA
  DBP. Monitor $$NUA has a Size of: $$HAA,$$WAA
NEX.
ENR.
```

```
[15:36:56] Monitor 1 has a Height of: $$HEI
[15:36:56] Monitor 1 has a Width of: 4096
[15:36:56] Monitor 1 has a Size of: 2160,4096
[15:36:56] Monitor 2 has a Height of: $$HEI
[15:36:56] Monitor 2 has a Width of: 3840
[15:36:56] Monitor 2 has a Size of: 2160,3840
```

Syntax

DTI.Height[|P1]

Parameter Explanation

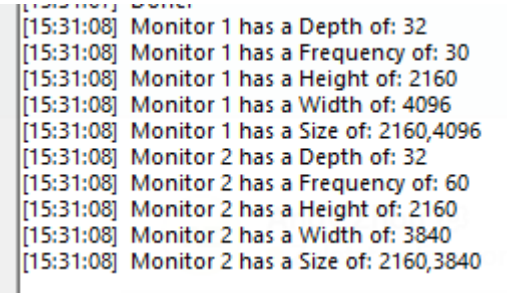
P1 – (optional) Variable, local or Global. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dd|$$NUA|$$DEP
  DBP. Monitor $$NUA has a Depth of: $$DEP
  DTI.df|$$NUA|$$FRE
  DBP. Monitor $$NUA has a Frequency of: $$FRE
  DTI.dh|$$NUA|$$HEI
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    DBP. Monitor $$NUA has a Height of: $$HEI
    DTI.dw|$$NUA|$$WID
    DBP. Monitor $$NUA has a Width of: $$WID
    DTI.sz|$$NUA|$$HAA|$$WAA
    DBP. Monitor $$NUA has a Size of: $$HAA,$$WAA
  NEX.
  ENR.
```



Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.6 DTI.MousePos

MiniRobotLanguage (MRL)

DTI.MousePos

Get X and Y Position of the Mouse Pointer on the total Desktop (all Monitors)

Intention

This command will return the X, and Y-Position of the Mousepointer in Screen-Coordinates.

```
DOL.1
  DTI.mp|$$XPO|$$YPO
  PRT.Position on $$MYH is  at: $$XPO,$$YPO
  PAU.0.1
OOP.
```

Syntax

DTI.MousePos[|P1][|P2]

Parameter Explanation

P1 – (optional) Variable for X-Result. If omitted the Result will be placed on TOS.
P2 – (optional) Variable for Y-Result. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DOL.1
  DTI.mp|$$XPO|$$YPO
  PRT.Position on $$MYH is  at: $$XPO,$$YPO
  PAU.0.1
OOP.
```

Remarks

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.7 DTI.Name

MiniRobotLanguage (MRL)

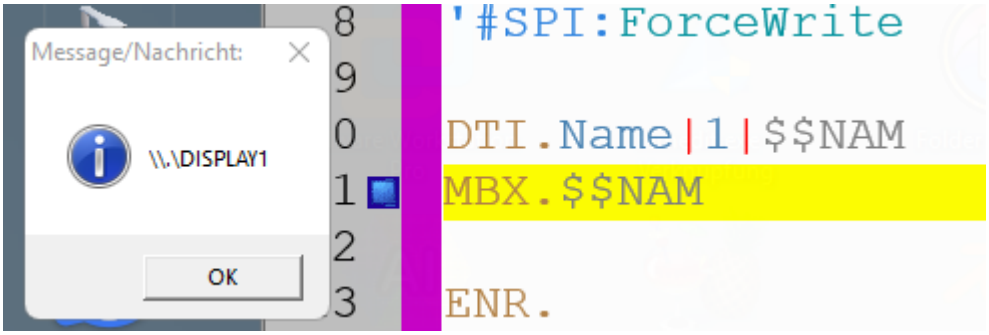
DTI.Name

Get Internal Windows-Name of Desktop-Screens (Monitors)

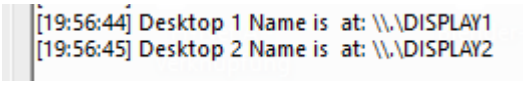
Intention

This command will the "Name" of the Display. Each Display "Monitor" has another Name.
The Display-Monitors are numbered starting with 1. Therefore "1" is always the Main Monitor.

```
DTI.cd|$$RES
MBX.$$RES
ENR.
```




```
' Example showing all connected Monitors
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dn|$$NUA|$$NAM
  DBP.Desktop $$NUA Name is  at: $$NAM
NEX.
ENR.
```



Syntax

DTI.Name|P1[|P2]

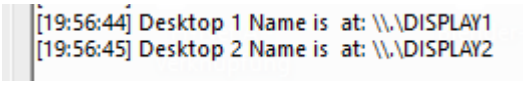
Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P1 – 1-based Number of the Monitor to return the name.
P2 – (optional) Variable,for result. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dn|$$NUA|$$NAM
  DBP.Desktop $$NUA Name is  at: $$NAM
NEX.
ENR.
```



Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.8 DTI.ScaleFactor

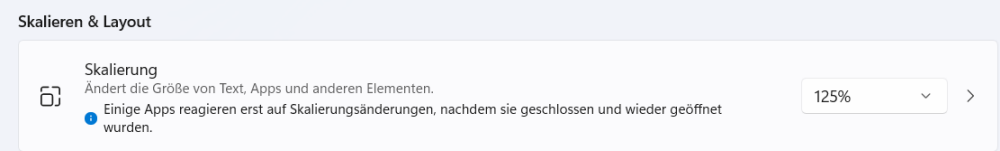
MiniRobotLanguage (MRL)

DTI.ScaleFactor

Return DPI-Scaling Factor for the primary Screen (Monitor)

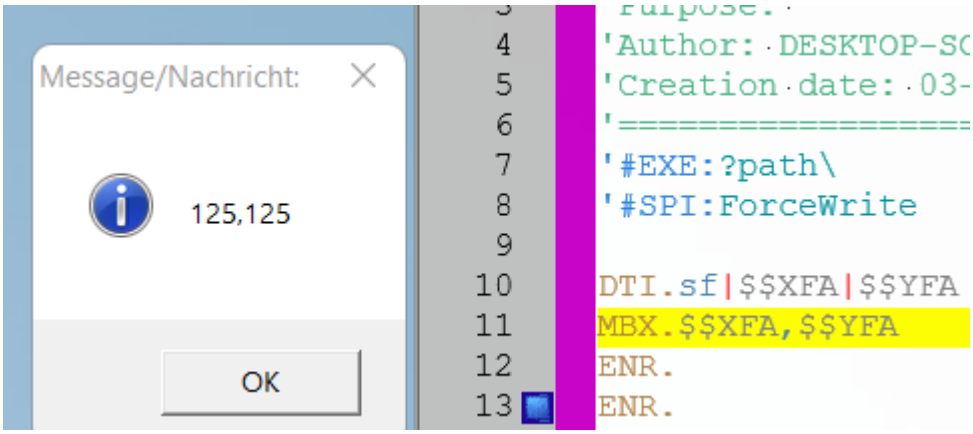
Intention

This command will give you the Scale-Factor of the primary Monitor. You will get the X- and the Y-Scale.
This Command is for the primary Monitor-Screen. You can not get the Factor for other connected Monitors if the are different.



.

```
DTI.sf|$$XFA|$$YFA
MBX.$$XFA,$$YFA
ENR.
```



Syntax

DTI.ScaleFactor[|P1][|P2]

Parameter Explanation

P1 – (optional) Variable, X-Result, If omitted the Result will be placed on TOS.
P2 – (optional) Variable, Y-Result, If omitted the Result will be placed on TOS.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'************************************
' Example 1
'************************************
DTI.sf|$$XFA|$$YFA
MBX.$$XFA,$$YFA
END.
```

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.9 DTI.ScalePos

MiniRobotLanguage (MRL)

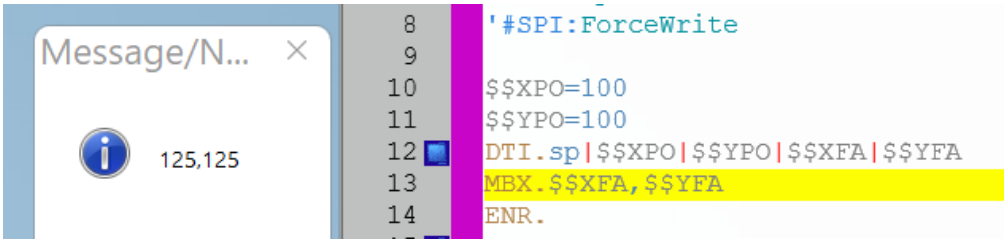
DTI.ScalePos

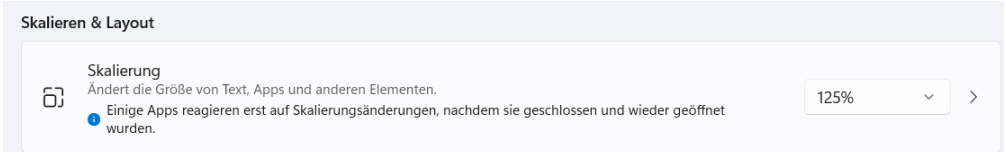
Calculate the real Position of a Pixel on the Screen using the Scale-Factor of the primary Screen (Monitor)

Intention

This command will return the scaled Pixel-Position using the Scale-Factor of the primary Screen.
If this is set to - for example 125% - then a Pixel that was originally at 100,100 will move to 125,125.

```
$$XPO=100
$$YPO=100
DTI.sp|$$XPO|$$YPO|$$XFA|$$YFA
MBX.$$XFA,$$YFA
ENR.
```





Syntax

DTI.ScalePos|P1|P2[|P3][|P4]

Parameter Explanation

P1 – Input-Position X,numeric or variable.
P2 – Input-Position Y,numeric or variable
P3 – (optional) Variable, X-Result, If omitted the Result will be placed on TOS.
P4 – (optional) Variable, Y-Result, If omitted the Result will be placed on TOS.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'************************************
' Example 1
'************************************
$$XPO=100
$$YPO=100
DTI.sp|$$XPO|$$YPO|$$XFA|$$YFA
MBX.$$XFA,$$YFA
ENR.
```

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.10 DTI.Size

MiniRobotLanguage (MRL)

DTI.Size

Get pysical Size of Desktop-Screens (Monitors)

Intention

This command will return the physical size in pixel of all connected "Monitors" (Desktops).

```
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DBP. Monitor $$NUA has a Height of: $$HEI
  DTI.dw|$$NUA|$$WID
  DBP. Monitor $$NUA has a Width of: $$WID
  DTI.sz|$$NUA|$$HAA|$$WAA
  DBP. Monitor $$NUA has a Size of: $$HAA,$$WAA
NEX.
ENR.
```

```
[15:36:56] Monitor 1 has a Height of: $$HEI
[15:36:56] Monitor 1 has a Width of: 4096
[15:36:56] Monitor 1 has a Size of: 2160,4096
[15:36:56] Monitor 2 has a Height of: $$HEI
[15:36:56] Monitor 2 has a Width of: 3840
[15:36:56] Monitor 2 has a Size of: 2160,3840
```

Syntax

DTI.Size[|P1][|P2]

Parameter Explanation

P1 – (optional) X-Size Result, Variable, local or Global. If omitted the Result will be placed on TOS.
P2 – (optional) Y-Size Result, Variable, local or Global. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dd|$$NUA|$$DEP
  DBP. Monitor $$NUA has a Depth of: $$DEP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    DTI.df|$$NUA|$$FRE
    DBP. Monitor $$NUA has a Frequency of: $$FRE
    DTI.dh|$$NUA|$$HEI
    DBP. Monitor $$NUA has a Height of: $$HEI
    DTI.dw|$$NUA|$$WID
    DBP. Monitor $$NUA has a Width of: $$WID
    DTI.sz|$$NUA|$$HAA|$$WAA
    DBP. Monitor $$NUA has a Size of: $$HAA,$$WAA
  NEX.
  ENR.
```

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.11 DTI.UnScalePos

MiniRobotLanguage (MRL)

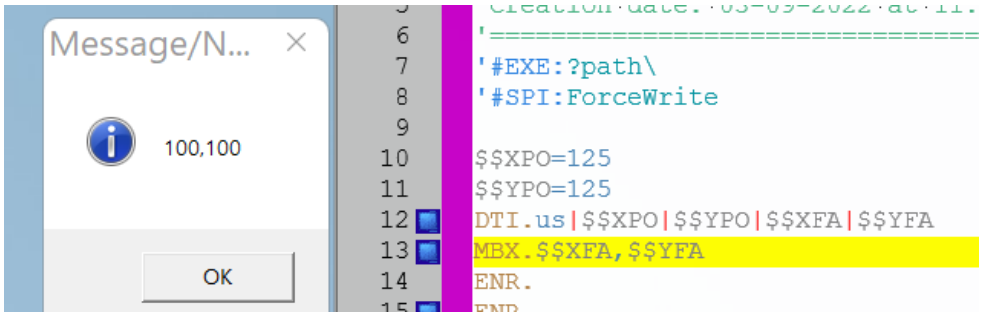
DTI.UnScalePos

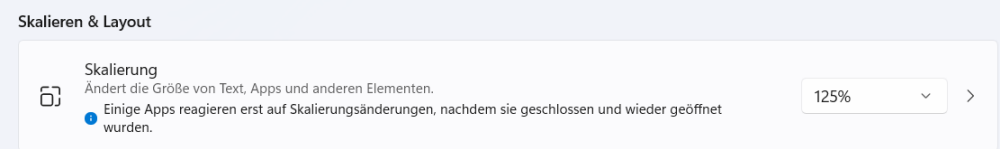
Count Desktop-Screens (Monitors)

Intention

This command will find our how many "Monitors" (Desktops) are connected to your Computer.

```
$$XPO=125
$$YPO=125
DTI.us|$$XPO|$$YPO|$$XFA|$$YFA
MBX.$$XFA,$$YFA
ENR.
```





Syntax

DTI.UnScalePos|P1|P2[|P3][|P4]

Parameter Explanation

P1 – Input-Position X,numeric or variable.
P2 – Input-Position Y,numeric or variable
P3 – (optional) Variable, X-Result, If omitted the Result will be placed on TOS.
P4 – (optional) Variable, Y-Result, If omitted the Result will be placed on TOS.

Example

```
'*************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Example 1
'************************************
$$XPO=125
$$YPO=125
DTI.us|$$XPO|$$YPO|$$XFA|$$YFA
MBX.$$XFA,$$YFA
ENR.
```

Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.5.12 DTI.Width

MiniRobotLanguage (MRL)

DTI.Width

Get physical Width in Pixel of Desktop-Screens (Monitors)

Intention

This command will get the physical Width of each connected Monitor.

```
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DBP. Monitor $$NUA has a Height of: $$HEI
  DTI.dw|$$NUA|$$WID
  DBP. Monitor $$NUA has a Width of: $$WID
  DTI.sz|$$NUA|$$HAA|$$WAA
  DBP. Monitor $$NUA has a Size of: $$HAA,$$WAA
NEX.
ENR.
```

```
[15:36:56] Monitor 1 has a Height of: $$HEI
[15:36:56] Monitor 1 has a Width of: 4096
[15:36:56] Monitor 1 has a Size of: 2160,4096
[15:36:56] Monitor 2 has a Height of: $$HEI
[15:36:56] Monitor 2 has a Width of: 3840
[15:36:56] Monitor 2 has a Size of: 2160,3840
```

Syntax

DTI.Width[|P1]

Parameter Explanation

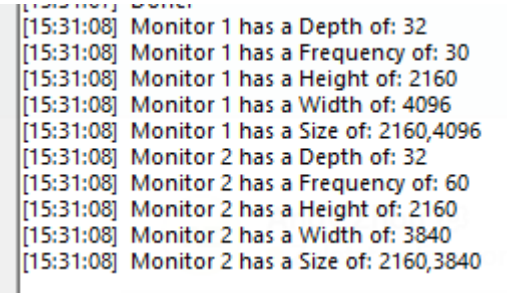
P1 – (optional) Variable, local or Global. If omitted the Result will be placed on TOS.

Example

```
'************************************
' Example 1
'************************************
DTI.cd|$$NUM
FOR.$$NUA|1|$$NUM
  DTI.dd|$$NUA|$$DEP
  DBP. Monitor $$NUA has a Depth of: $$DEP
  DTI.df|$$NUA|$$FRE
  DBP. Monitor $$NUA has a Frequency of: $$FRE
  DTI.dh|$$NUA|$$HEI
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  DBP. Monitor $$NUA has a Height of: $$HEI
  DTI.dw|$$NUA|$$WID
  DBP. Monitor $$NUA has a Width of: $$WID
  DTI.sz|$$NUA|$$HAA|$$WAA
  DBP. Monitor $$NUA has a Size of: $$HAA,$$WAA
NEX.
ENR.
```



Remarks

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.6 GAR. - Get actual Rectangle

MiniRobotLanguage (MRL)

GAR. Command

Get actual RECT

Intention

The GAR-Command can be used to get the actual localized rectangle of any window or object in screen coordinates. It may also return the type of localized object and the actual localized window-handle.

All windows or accessible objects or graphical objects that are localized have a rectangle outline. This command will return their rectangle coordinates, in this format:

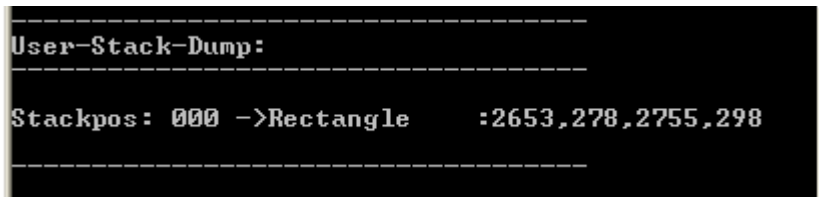
```
left,top,right,bottom
```

as screen coordinates.

The simplest mode of using GAR. is just writing

```
GAR.
```

then it may leave coordinates like these on the Stack:



Here is a example that uses P1 instead of the stack for the return-RECT coordinates:

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
GAR.$$REC
DBP.$$REC
ENR.
' Result is: 2653,278,2755,298
```

You can also find out, if the last localized Object was a window or something else:

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
SAO.r|42
GAR.$$REC|$$TYP
DBP.$$TYP
ENR.
' Result is "1" - accessible Object
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
SAO.r|42
GAR.$$REC|$$TYP
DBP.$$TYP
ENR.
' Result is "0" - window type object
```

In the following example we will use the command to retrieve the actual located window handle - if any.

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}
GAR.$$REC|$$TYP|$$HWN
DBP.$$REC-$$TYP-$$HWN

SAO.r|42
GAR.$$REC|$$TYP|$$HWN
DBP.$$REC-$$TYP-$$HWN
ENR.
```

Result is:

```
[09:41:05] [16] 2653,278,2755,298-0-265108
[09:41:06] [20] 2655,280,2753,296-1-0
```

Syntax

GAR.[P1][|P2][|P3]

Parameter Explanation

P1 - (optional) Variable that will get the Rectangle coordinates, if omitted, Stack is used.

P2 - (optional) Variable that will receive the type of the last localized Object.

These types are:

```
0 - Window-Object
1 - accessible (web/office) Object
2 - graphical object
```

P3 - (optional) Variable that receives the handle of the actual localized window, if any.
If the last localized object is not of type "window", this variable will be set to zero.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
STW.ct|MyClassName|Sample Application
SCW.nct|1|Edit|{&NOTEXT:}

GAR.$$REC|$$TYP|$$HWN
DBP.$$REC-$$TYP-$$HWN

SAO.r|42
GAR.$$REC|$$TYP|$$HWN
DBP.$$REC-$$TYP-$$HWN

ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.8. Selecting the Application (Topwindow) 150
- 1.9. Locating Child-Windows 162
- ! SCW - Search-Child-Window 3738
- !. STW. - Description 3814
- CST. - Coordinate System Transform 2331
- GAR. - Get actual Rectangle 3619
- MAP. - Move Active Point 2483
- GRM. - Get Rect Middle-Point 2337
- QFR. - Quadrant from RECT 1130

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.7 GCT. - Get-Childs-TopWindows

MiniRobotLanguage (MRL)

GCT. Command

Get Childs Topwindow

Intention

The GCT-Command can be used to return the Top-Window of the currently located Window, whether it is a child or a Top-Window.

If you have multiple equal Top-Windows, and you need to locate a "Non-unique" button in just one of them, GCT. and SCW. will help you to do the trick.

First use the SCW.-command 3752 with the "T"-Flag to search directly a "unique child-window" (static, button etc.) in the proper application-window. Next use GCT. to get the Top-Window of it. From that point you can locate another control in that window, using SCW..

You can use SAO to search a Desktop-Element and get the Top-Window using GCT.

GCT. will also set the Backjump-Line 3764 to its line number. Therefore all following SCW. will rollback to GCT.

Syntax

GCT.[P1][|P2]

Parameter Explanation

P1 - Variable with a Window-Handle
P2 - Variable P2 is been loaded with the Handle of the Top-Window from P1.

If P1 is given, the Top-Window of P1 is been calculated and located.

If P1 and P2 is given, Variable P2 is been loaded with the Handle of the Top-Window from P1. The Top-Window is NOT located in this case.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.t|Internet Explorer
SCW.c|msctls_
'Locate the Top-Window of the actual selected Window..
GCT.
' Mouse moves just as if we would not have been doing the SCW.
MMV.
```

Remarks

-

Limitations:

-

See also:

- ! SCW - Search-Child-Window 3738
- !. STW. - Description 3814

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.8 GFW. - Get Foreground Window

MiniRobotLanguage (MRL)

GFW. Command

Get Foreground Window

Intention

The `GFW.` Command retrieves a handle to the foreground window (that is the window with which the user is currently working).
The handle of the window is returned and the window is been immediately located.
If the handle can not be returned, for whatever reason, the command returns a zero.

Usage is as simple as:

```
' Top Enabled Window
GFW.$$REA
DBP.$$REA
```

Syntax

GFW.[P1]

Parameter Explanation

P1 **-** ***(optional)*** variable that will returns the window handle, if omitted TOS is use.
If the window is found, it is localized.

Example

```
GFW.$$REA
DBP.$$REA
```

Remarks

-

Limitations:

*© 2023 Theo Gottwald * https://www.smart-package.com*

-

See also:

- ! SCW - Search-Child-Window 3738
- !. STW. - Description 3814
- GSW. - Get Special Window 3630
- GCT. - Get-Childs-TopWindows 3622

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.9 GRW. - Get Related Window

MiniRobotLanguage (MRL)

GRW. Command

Get Related Window

Intention

The `GRW.` Command can be used to return the handle of windows, that are in a specified relation to the given input window. That is for example, a parent, an owner or a child-windows.

The handle is returned and the window is been immediately located.

If the handle can not be returned, for whatever reason, the command returns a zero.

Usage is as simple as:

```
' Locate the taskbar window above to the Start Button (XP
only)
GSW.sb|$$RET
GRW.pr|$$RET|$$REA
DBP.$$REA
```

What is the "Z-Order"?
The z-order of a window indicates the window's position in a stack of overlapping windows. This window stack is oriented along an imaginary axis, the z-axis, extending outward from the screen. The window at the top of the z-order overlaps all other windows. The window at the bottom of the z-order is overlapped by all other windows.
The system maintains the z-order in a single list. It adds windows to the z-order based on whether they are topmost windows, top-level windows, or child windows. A topmost window overlaps all other non-topmost windows, regardless of whether it is the active or foreground window. A topmost window has the TOPMOST style. All topmost windows appear in the z-order before any non-topmost windows. A child window is grouped with its parent in z-order.

For more Infos see: MSDN Windows

Syntax

GRW.[P1][|P2][|P3]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 - (optional) specifies the window that shall be returned, out of the following:

"parent","pr"
Retrieves a handle to the specified window's parent or owner. To retrieve a handle to a specified ancestor, use the Ancestor commands ("pa","ra","oa").

"child","ch"
The retrieved handle identifies the child window at the top of the Z order, if the specified window is a parent window; otherwise, the command returns zero. The function examines only child windows of the specified Input-window. It does not examine descendant windows.

"first","gf"
The retrieved handle identifies a window of the same type that is highest in the Z order. If the specified window is a topmost window, the handle identifies a topmost window. If the specified window is a top-level window, the handle identifies a top-level window. If the specified window is a child window, the handle identifies a sibling window.

"last","gl"
The retrieved handle identifies a window of the same type that is lowest in the Z order. If the specified window is a topmost window, the handle identifies a topmost window. If the specified window is a top-level window, the handle identifies a top-level window. If the specified window is a child window, the handle identifies a sibling window.

"next","gn"
The retrieved handle identifies a window below the specified window in the Z order. If the specified window is a topmost window, the handle identifies a topmost window. If the specified window is a top-level window, the handle identifies a top-level window. If the specified window is a child window, the handle identifies a sibling window.

"previouse","gp"
The retrieved handle identifies a window above the specified window in the Z order. If the specified window is a topmost window, the handle identifies a topmost window. If the specified window is a top-level window, the handle identifies a top-level window. If the specified window is a child window, the handle identifies a sibling window.

"owner","ow"
The retrieved handle identifies the specified window's owner window, if any.

"parent ancestor","pa"
Retrieves the parent window. This does not include the owner-window.

"root ancestor","ra"
Retrieves the root window by walking the chain of parent windows.

© 2023 Theo Gottwald * https://www.smart-package.com

"owner ancestor","oa"
Retrieves the owned root window by walking the chain of parent and owner windows returned, using "parent" command above.

"popup","po"
May retrieve the handle of an owned PopUp-Window to that window, if any.

"toplevel","tl"
Retrieves the window's top-level parent or owner window.

"topparent","tp"
Retrieves the window's top-level parent window.

"formhandle","fh"
Finds the handle of the top-level window or MDI child window that is the ancestor of the
specified window handle. The reference handle is the handle of any control on the form.

"childtop","ct"
Examines the Z order of the child windows associated with the specified parent window
and retrieves a handle to the child window at the top of the Z order.

P2 - (optional) P2 can be the window handle of the input window.
If its omitted, the actual window is used.
If there is no actual window, the function will return zero.

P3 - (optional) Variable P3 is been loaded with the Handle of the
Window if it is found. If the window is not found a zero is
returned. If P2 is omitted, the result is been placed on TOS.

If the window is found, it is localized.

Example

```
' Click the volume icon in the System-Notification Area.
GSW.sb|$$RET
AGR.43|Volume
MLC.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! SCW - Search-Child-Window 3738
- !. STW. - Description 3814
- GSW. - Get Special Window 3630
- GCT. - Get-Childs-TopWindows 3622

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.10 GSW. - Get Special Window

MiniRobotLanguage (MRL)

GSW. Command

Get Special Window

Intention

The `GSW.` Command can be used to return the handle of special windows, like the desktop, the Traybar or the Quick-Launch.

The handle is returned and the window is been immediately located.

If the handle can not be returned, because - for example - there is no explorer loaded, the command returns a zero.

Using this command is safer then locating these windows in another way, because it uses internal methods that are mostly safer for international or newer windows versions.

To be safe, test your script on the OS that it finally should run on.

Usage is as simple as:

```
' Click the volume icon in the System-Notification Area.
GSW.sb|$$RET
AGR.43|Volume
MLC.
```

Syntax

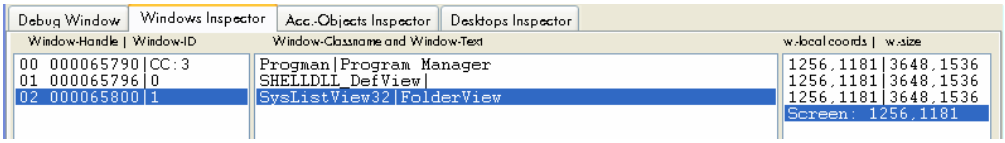
GSW.[P1][|P2]

Parameter Explanation

P1 - (optional) `specifies the window that shall be returned, out of the following:`

`"shell desktop","sd"` – locate the shell-desktop-Top-Window. In the example below it has the handle `65790`. On a dual-screen system, this window has the X-Size over both screens, and the Y-Size of the larger of both screens.

© 2023 Theo Gottwald * https://www.smart-package.com

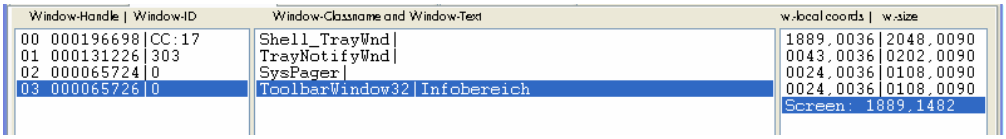


"desktop","dt" - locates the actual Input Desktop. On a dual-screen system it covers
only the main desktop with the taskbar and the start-menu. Its not visible in the windows
-analyzer when you move the mouse over the desktop.

"sc","shell child" - locate the shell-desktop-Top-Window and then the next child-window. In the example above it has the handle 65796 and the window-class "SHELLDLL_DefView".

"dtl","desktop listview" - locate the shell-desktop-Top-Window and then the desktop-listview. In the example above it has the handle 65800 and the window-class "syslistview32".

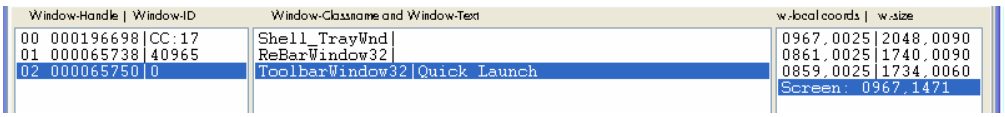
"tw","tray" - locate the Shell-Tray-Top-Window. In the example below it has the handle 196698 and the window-class "Shell_TrayWnd". On the primary desktop
this window covers the complete taskbar.



"tb","traybar" - locate the Desktop Traybar-User Notification Area-Toolbar-Window. This is the window that contains the Traybar Notification-Icons. In the example
above it has the handle 65726 and the window-class "toolbarwindow32".

"sb","start button" - locate the Shell-Tray-Top-Window and then the START-Button.. This option may currently only work under Windows XP.

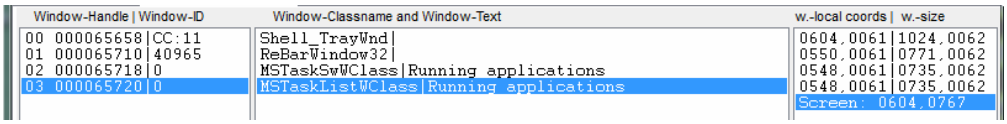
"ql","quick launch" - locate the Shell-Tray-Top-Window and then the Quicklaunch Area. This option may only work under Windows XP.



"ab","application bar"- locate the Shell-Tray-Top-Window and then the application
run-area Taskbar). In the picture below that is the window wit hthe handle 65718 and

© 2023 Theo Gottwald * https://www.smart-package.com

the class **"MSTaskSwWClass**.". The window on Level 3 seems to be a toolbar on XP
and a "MSTaskListWClass" under Windows 7.



P2 **-** ***(optional)*** Variable P2 is been loaded with the Handle of the
Window if it is found. If the window is not found a zero is
returned. If P2 is omitted, the result is been placed on TOS.

If the window is found, it is localized.

Example

```
' Click the volume icon in the System-Notification Area.
GSW.sb|$$RET
AGR.43|Volume
MLC.
```

Remarks

Tested with Windows XP and Windows 7. There may be changes needed for Windows 8 and higher.

Limitations:

-

See also:

- ! SCW - Search-Child-Window 3738
- !. STW. - Description 3814
- GRW. - Get Related Window 3626
- GCT. - Get-Childs-TopWindows 3622

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.11 GWP. - Get Window Position/Size

GWP. - Get Window Position and Size Previous 3630 Top 41 Next 3636

MiniRobotLanguage (MRL)

GWP. Command

Get Window Position and Size

Intention

The GWP.-Command can be used to get the position or the size of the actual selected Top- or Child-Window.

The window position is been calculated from the top-left corner of the window or child-window. To change window-position and size, use SWP. - Set Window Position and Size 3884

Syntax

GWP.[P1][,P2][|P3][,P4][|T]

Parameter Explanation

P1 - Left-Position
P2 - Top-Position
P3 - X-Size
P4 - Y-Size
T - if used, the Top_window of the actual selected Window is taken, no matter
if a Top- or a child-Window is the actual located window.

You can omit any of the Parameters.
In that case the results will be placed on the stack in the following Order:

```
TOS:  §§000 - Window-Centerpoint, X - Coordinate
      §§001 - Window-Centerpoint, Y - Coordinate

      §§002 - Window- Upper, left Corner, X - Coordinate
      §§003 - Window- Upper, left Corner, Y - Coordinate

      §§004 - Window-Size, X
      §§005 - Window-Size, Y

      §§006 - Window- Lower, right Corner, X - Coordinate
      §§007 - Window- Lower, right Corner, Y - Coordinate
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*******************************************
' GWP. Sample 2
'*******************************************
'
STW.ct|PBWindowClass:0|BCS - Testform
'SCW.t|Next
SWP.0,0|640,400
GWP.$$XPO,$$YPO|$$XSZ,$$YSZ
STS.DUMP
MBX.$$XPO,$$YPO-$$XSZ,$$YSZ
ENR.

'*******************************************
' GWP. Sample 2
'*******************************************
STW.c|ieframe
' The window is moved to position 0,0|640,400
' and resized to 640x400
SWP.0,0|640,400
GWP.$$XPO,$$YPO|$$XSZ,$$YSZ

PRT. The location and size of the located IE-Window is:
PRT. $$XPO,$$YPO - $$XSZ,$$YSZ
END.

' More Examples
' The window is resized to 500x500 pixel
GWP.|$$XSZ,$$YSZ

' The window is resized to 200x200 and vertically moved to
y=300
GWP.,$$YPO|$$XSZ,$$YSZ

' The window is only moved, not resized.
GWP.$$XPO,$$YPO
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- **3.1 Systemvariables** 334
- SWP. - Set Window Position and Size 3884
- SWS. - Set Window State 3889

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.12 HTV. - Handle-to-Variable

MiniRobotLanguage (MRL)

HTV. Command

Handle to Variable

Intention

This command returns the Window-Handle of previous localized windows into the given variable or on TOS 369.

Examples for usage:

```
' place handle of last localized window on TOS
HTV.

' place handle of last localized window in $$HWN
HTV.$$HWN

' place handle of last localized top-window in $$HWN
HTV.$$HWN|t

' place handle of last localized child-window in $$HWN
HTV.$$HWN|c
```

Syntax

HTV.[P1][|P2]

Parameter Explanation

P1 – First Parameter is a Variable, local or Global. If invalid, or **-**, Stack is used.
P2 – Second Parameter (Flag) is optional, it can be
a - Handle from last localized Window,Top- or Childwindow.
c - Handle from last localized Childwindow.
t - Handle from last localized Topwindow.

Example

```
'*****************************************************
' EXAMPLE : HTV
'*****************************************************
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
STW.ct|#32770|Sample App
 ' This Line puts the Handle on TOS
HTV.
' This Line puts the Handle into the given Variable
HTV. $$HWN
' This Line puts the last localized Childwindow into Stack
HTV.-|c
```

Remarks

-

Limitations:

-

See also:

- !. STW. - Description 3814
- ! SCW - Search-Child-Window 3738
- 1.8. Selecting the Application (Topwindow) 150
- 1.9. Locating Child-Windows 162
- EXH. - Execute-Handle 890
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.13 HWI. - Hide Window

MiniRobotLanguage (MRL)

HWI. Command

Hide Window or unhide Window using Transparency

Intention

You want to hide your Setup-Program from the user while installation?
`HWI.` can make windows invisible while you can still click all of their buttons!
`HWI.` can also make Windows transparent, that you can see through these windows.

This command can make Windows transparent or invisible.
It can also reverse that effect.
It can also make Windows from any applications that are already transparent, invisible or opaque.

Examples for usage:

```
' make last localized Topwindow get invisible
HWI.0

' make last localized Topwindow get opaque
HWI.255

' make last localized Topwindow half transparent
HWI.128

' make TopWindow with Handle $$HWN half transparent
HWI.125|$$HWN
```

This command can also be used to make windows appear softly from invisibility ("blend-in effect").

Syntax

HWI.[P1][|P2]

Parameter Explanation

P1 – numerical Parameter 0-255.
0 - Window is completely invisible
255 - Window is opaque (completely visible)
All values between make the window more or less transparent.

*© 2023 Theo Gottwald * https://www.smart-package.com*

P2 – ***(Optional)*** If given it is a Handle from a Top-Window.
If omitted, the last localized Top-Window is used.
This command works only with Top-Windows, if you want to make
Child-Windows, Buttons etc. invisible, use the SWS.-command instead.

Example

```
'******************************************************
' EXAMPLE : HWI
'******************************************************
STW.ct|#32770|Sample App
' Blend a window "in" slowly
FOR.$$CNT|0|255
 HWI.$$CNT
PAU.0.1
NEXT
```

Remarks

-

Limitations:

Requires Windows XP or higher.
This command works only with Top-Windows, if you want to make
Child-Windows, Buttons etc. invisible, use the SWS.-command instead.

See also:

- !. STW. - Description 3814
- 1.8. Selecting the Application (Topwindow) 150
- SWS. - Set Window State 3889

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.14 ICW. - If-Child-Window

3.83.14.1 ! ICW. - If-Child-Window

MiniRobotLanguage (MRL)

ICW. Command

Seach Child Window

Intention

This command localizes a specified Child-Window (Control, Button etc.) within a previous localized Top-Window (Application-Window).

To understand this command, be sure to read the chapters:
1.8. Selecting the Application (Topwindow) 150
and
1.9. Locating Child-Windows 162
first.

It will then sets the Mouse-Target Point into the window-center.

`ICW.` has a lot of features, which are controlled using the 1.4. The Prototype Parameter System 73 .Be sure you have read and understand that chapter before trying to understand `ICW.`

Unless using the "T"-Option, the `ICW.`-command needs a Top-Window being localized first. Therefore please read the section 1.8. Selecting the Application (Topwindow) 150 carefully.

The Command has a "Rollback" Feature, when the Child-Window is not found in few Seconds. It jumps back to the `STW.` (Search Top-Window) and tries to get a new Top-Window, which possibly has the wanted button or control.
This will continue, until its time has run out.
If you use the `DBM.1` -command, you can see these "Rollback-Jumps" in the Debugging-Console. See the example below.

```
DBM.1
STW.Act|PBWindowClass:0|TestApp
' This button/childwindow does not exist
ICW.nctb|2|Button|Button 23|.5
MLE.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
[10:13:11] STW.Act|PBWindowClass:0|TestApp
[10:13:11] SCW.nctb|2|Button|Button 23|.5
[10:13:12] STW.Act|PBWindowClass:0|TestApp
[10:13:12] SCW.nctb|2|Button|Button 23|.5
[10:13:12] STW.Act|PBWindowClass:0|TestApp
[10:13:12] SCW.nctb|2|Button|Button 23|.5
[10:13:13] STW.Act|PBWindowClass:0|TestApp
[10:13:13] SCW.nctb|2|Button|Button 23|.5
```

You can control this Feature using the "b" - Backjump-Time Flag 3764.

Syntax

ICW.[Prototypes|][Parameters|]

Parameter Explanation

Parameter

```
 a - (np) "all" - search all windos, includes
     invisible and disabled windows.
 b - (p) "backjump"-flag, time in seconds, 0 disables the jump
 c - (p)  window-class
 d - (p) window-depth
 e - (np) enabled, window must be enabled
 f - (np) disabled, window must be disabled
 g - (p) Goto on found
 h - (p) handle. Immediately locate the window with given
handle.
     All other prototypes will be ignored.
 i - (p) window-id
 j - (p) WaitForInputIdle (Value)
 k - (np) WaitForInputIdle (no Value), use value
      from last Call/default-value
 n - (p) Fensternumber in der Ebene ("WR->SWC")
 o - (p) on not found goto (Label)
 p - (p) Process-ID (PID 137)
 q - (p) Thread-ID (TID)
 r - (p) result number ("WR-SPC-Count")
 s - (np) internal reversing at child windows search
 t - (p) text
 u - (np) invisible, window must be invisible
 v - (np) visible, window must be visible
 w - (p) total waiting-time in Seconds, fractions (0.5)
     are possible, zero is possible.

 E - (np) Add Window to Exclude List after localization
 I - (p) p = handle. Locate window and remove window
     from Exclude list.
 K - (n) Immediately clear Exclude-List before specified
search
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Q - (n) use previous Prototypes
S - (n) Search only next Level (SWC-Option)
T - (n) Search desktopwide, ignore localized Top-Window
V - (p/Label) Gosub on not found
W - (p/Label) Gosub on  found
X - (n) Run without Error on not found
```

Explanation:
(np) -> only prototype, no additional parameter required
(p) -> prototype requires additional parameter
(NUM/VAR) - window handle/variable
(p/Label) - Label-Parameter required, can also be a variable that contains the label

P2 **-** Parameters, separated by | according to the prototypes

Example

```
STW.Act|PBWindowClass:0|TestApp
ICW.nctE|2|Button|Button 2
' Coordinates relative to left, upper corner of button
MPO.o|42,22
MLE.
EIF.
```

Remarks

When using `ICW.` to check if a child-window is available or not, it makes sense that you use the "b"-flag with a value of "0" to disable the backjump.

Limitations:

-

See also:

- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- 1.5. Features and Hints 150
- 1.9. Locating Child-Windows 162
- 1.8. Selecting the Application (Topwindow) 150
- 1.7. Application and PID 137
- 3.1 Systemvariables 334
- !. STW. - Description 3814

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.15 IEW. / NEW. - If Exist Window

MiniRobotLanguage (MRL)

IEW. Conditional Statement

If Exist Window

Intention

Conditional Statement. Test if one or more (up to 20) a specified window handle(s) is valid or not. A window handle is valid until the corresponding window has been closed (destroyed).

`IEW.` tells you, if the actual window is still available or has been closed.
If no parameter is given, the actual window is been tested. If you give parameter, these must be window handles, or the letters "**c**" or "**t**".
These letters stand for "child" and "Top". If given they will be replaced with the last window of that kind that has been localized.

c - last localized child-window

t - last localized Top-Window

Please note, that if the actual window is a child-window (e.g. a button), you need to specify "**t**" as parameter,if you want to test the actual top-window. Some examples:

```
'Get first window handle
STW.ct|CabinetWClass|F:\00_RPGM
HTV.$$HAA
'Get second window handle
STW.ct|Chrome_WidgetWin_0|meebo.com
HTV.$$HAB
'Test both handles
IEW.$$HAA|$$HAB
  DBP.All windows are alive
ELS.
  DBP.At least one of the windows is history
EIF.
ENR.
```

Other modes of operation are:

```
' Check actual located window
IEW.

' check last located child window
IEW.c

' check actual Top-window
IEW.t
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' check window with handle in $$MYH
IEW.$$MYH
```

NEW. is the negative version "Not Exist Window".

Syntax

IEW.P1[..|Px] ... ELS. ... EIF.

Parameter Explanation

P1 **- numeric or "c" or "t".** Can be Variable that contains a window handle. If omitted actual window is taken. You can also use these letters:

- **c** - last localized child-window
- **t** - last localized Top-Window

If you add multiple handles, they are AND'ed with IEW. If ANY of the given handles is invalid, the instruction will fail.

If you give multiple handles with NEW. (negative form of the command), the opposite will happen. If all of the given handles are valid, the instruction will fail.

IEW.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Speed in Ticks:

This command uses typically around 290 Ticks.

Example

```
'*********************************
' IEW./NEW.-Sample
'*********************************
STW.t|Sample
MMV.
IEW.
 MBX.Ok
ELS.
 MBX.Diabled!
EIF.
END.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- 1.8. Selecting the Application (Topwindow) 150
- HTV. Handle-to-Variable 3636
- !. STW. - Description 3814
- ! SCW - Search-Child-Window 3738
- "h" - Flag - SCW. 3779

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16 ITW. - If Top Window

3.83.16.1 ! ITW. / NTW. -If Top-Window

MiniRobotLanguage (MRL)

ITW. Conditional Statement

If Top Window

Intention

This conditional statement works like the `STW.`-command, and also has all options (prototypes) from the `STW.`-command.
Unlike `STW.`, `ITW.` does not wait for a window to appear by default.
If the window is not there it will choose the `ELS.`-case.

If you want `ITW.` to wait for a window, you can specify the "w"-Flag and the waiting time manually.

It forms conditional statement to make structured scripts more readable.

`ITW.` localizes a specified Top-Window and sets the actual point into its center.

All following `SCW`-commands will be within this window.

`ITW.` has a default window-treatment 3814 to ensure that any further action in that window will be effective. This default treatment will do following:
- normalize minimized windows (like the "D"-Flag
- windows that are outside the desktop-work area will be moved to 0,0 (like "G"-Flag)
- The window will be moved in front of all other (non-TopMost 3838)-windows.
- The window will receive the input focus. (like the "J"-Flag)

This default window-treatment can be completely deactivated using the "A" - Flag - ITW. 3651.

This makes sense, for example when working with Menu's or other windows which disappear when the default treatment is being applied.

Alternatively the default window treatment can be influenced using other flags, like "B" or "G".

If the window should be brought into work area, without other treatment, use the "B"-Flag. "G", will just prevent the repositioning of the window and let the other treatment-steps happen.

All window search operations change the Timeout System Variables `#tio#` and `#til#`. Using these you can find out if a window was found or not, even if you have used the"X" Flag (X - run without Error).

*© 2023 Theo Gottwald * https://www.smart-package.com*

Hint: Please note that the Flags have a priority from left to right (highest Priority). If you add other flags after the "A" there may be a treatment. Because the more right the flags, the higher their priority. If you place the "A" last inside the prototypes, all involved treatment will be disabled.

See two examples.
In this first case the "A" Flag has priority over the "C" Flag, because its right of it.

```
ITW.tCA|explorer
  MMV.
EIF.
```

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
ITW.tAC|explorer
MMV.
EIF.
```

Some Flags deactivate the default treatment or parts of it. See description below.

Hint: If you omit all parameters like this:
`ITW.`
the command will automatically restore the parameters from the previous call. This makes it easy to always locate the same Top-Window.Note that this also applies to `STW.` as both use the same parameters-store.

Syntax

ITW.P1[|P2]...[|Px]

Parameter Explanation

P1 – Prototype
P2 .. **PX** - Parameters according to prototypes

```
A - (np) Switch off default window treatment completely
B - (np) disables following parts of "default treatment":
     NO Restore from Minimized, NO Set Input-Focus,
     NO Bring to Front
C - (np) If the window i maximized, it will be restored to
    normal size
D - (np) If the window is minimized,  will be restored to its
    previous size, before it was minimized,
E - (np) Put Localize window, then add window to
    Search Exclude List.
F - (np) Move window to center of primary desktop
G - (np) disable default window-treatment-"repositioning if
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
     outside"
H - (np) a window that is outside the primary desktop working
area
    will be repositioned. Default is that the window can be
also
    on a second desktop.
I - (NUM/VAR) Remove window with the given window-handle from
the
    search-exclude-list.
J - (np) Bring window to Front. Does not switch off default
window
    treatment.
K - Immediately clear Exclude-List before specified search
M - (np) Set window to TOPMOST. Such windows will be on Top
of all
    other Non-TOPMOST windows. DANGER - even Messageboxes may
be
    hidden behind a TOPMOST window. Use this with caution!
N - (np) remove the TOPMOST Property from a window.
Q - use previous Prototypes (only prototypes)
V - (p/Label) Gosub Line/Label if the window was NOT found
W - (p/Label) Gosub Line/Label if the window was found
X - (np) run without error if window was not found. Timeout
Flag
    #tio# and #til# will anyway be set and can be used.

a - (np) changes internal settings that "all" windows will be
    found. This includes invisible and disabled windows.
b - (np) "back" - moves window to the back, behind other
windows.
c - (p) "class", window-class parameter
e - (np) enabled, the window must be enabled or it will
ignored.
f - (np) disabled, the window must be disabled or it will
ignored.
g - (p/Label) Goto Line/Label if the window was found
h - (p) localize the window directly using a given window
handle.
    Note, if "h" is given, other search conditions will
    be ignored.
j - (p) WaitForInputIdle, you can specify the WII timeout
k - (np) WaitForInputIdle - no parameter, use the default
    WII timeout.
o - (Label/Variable) on not found goto (Label)
p - (p) process-ID, search only windows from process with
this
    given PID
q - (p) ThreadID,, search only windows from process with this
    given TID
r - (p) No. of result to choose (Result-Number for Top-
Windows)
t - (p) text, window-text (often the window caption)
u - (np) invisible, search only invisible windows
v - (np) visible, search only visible windows
w - (p) total waiting-time in Seconds, 0.5 (FP-numbers) is
    possible.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Explanation:
(np) -> only prototype, no additional parameter required
(p) -> prototype requires additional parameter
(NUM/VAR) - window handle/variable
(p/Label) - Label-Parameter required, can also be a variable that contains the label

P2 - Px - Data according to the prototypes, separated by |

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
' We use ITW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
'
NTW.wt|0|BCS - Testform
  EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.

' Now we can use the Application and press Buttons
' No STW, - needed, as ITW. locates the window already
SCW.nctb|1|Button|Next|0
' Compatible Button Click
MLC.

'-------------------------------------------------------------
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- ! STW. -Search Top-Window 3814
- 1.5. Features and Hints 150
- 1.9. Locating Child-Windows 162
- 1.7. Application and PID 137
- 3.1 Systemvariables 334

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2 "A" - Flag - ITW.

"A" - Flag "No windows treatment" Previous 3646 Top 41 Next 3653

MiniRobotLanguage (MRL)

"A" - Flag "No windows treatment"

If Top Window

Intention

ITW. has a default window-treatment to ensure that any further action in that window will be effective. This default treatment will do following:
- normalize minimized windows (like the "D"-Flag
- windows that are outside the desktop-workarea will be moved to 0,0 (like "G"-Flag)
- The window will be moved in front of all other (non-TopMost)-windows.
- The window will receive the input focus. (like the "J"-Flag)

This default window-treatment can be completely deactivated using the "A"-Flag.

This makes sense, for example when working with Menu's or volatile windows, which disappear when the default treatment is being applied.

Usage:

```
ITW.cA|#32768
 MMV.
EIF.
```

Please note that the Flags are internally analyzed in the order in which they are given. If you add other flags after the "A" there may be a treatment.
If you place the "A" last inside the prototypes, all such treatment will be disabled.

In this case the "A" Flag has priority over the "C" Flag, because its right of it.

```
ITW.tCA|explorer
 MMV.
EIF.
```

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
STW.tAC|explorer
MMV.
```

Example

```
'#EXE:?pfad\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'#SPI:ForceWrite
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
' If its available
ITW.cA|#32767
 MMV.
EIF.
ENR.
```

Remarks

Alternatively the default window treatment can be influenced using other flags, like "B" or "G".

If the window should be brought into work area, without other treatment, use the "B"-Flag. "G", will just prevent the repositioning of the window and let the other treatment-steps happen.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3 "B" - Flag - ITW.

"B" - Flag "partial treatment" Previous 3651 Top 41 Next 3655

MiniRobotLanguage (MRL)

"B" - Flag "partial treatment"

If Top Window

Intention

ITW. has a default window-treatment to ensure that any further action in that window will be effective. This default treatment will be reduced, when the "B" - Flag is set.

When the "B" Flag is in effect, windows that are outside the desktop-workarea will be moved to the upper, left corner of the desktop - that is position 0,0. That is the only treatment that will be done.

Also this can be disabled with the "G"-Flag. Means, that the flags "G" and "B" together are just like the "A" flag,

Example:

```
ITW.Bc|WindowClass
 MMV.
EIF.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
ITW.cB|#32767
 ' Locate the button
 SCW.t|Ok
 ' Click the button
 MLE.
EIF.
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.4 "C" - Flag - ITW.

"C" - Flag "restore if maximized" Previous 3653 Top 41 Next 3656

MiniRobotLanguage (MRL)

"C" - Flag "restore if maximized"

If Top Window

Intention

If the window is maximized, this flag will bring the window back to its normal size.
You can maximize, minimize and restore windows using the SWS. - Set Window State 3889 command.

It does not matter, if the window was minimzed before it was maximized.
The window will be normal size after this flag.

Example

```
ITW.ct|PBWindowClass:0|TestApp
' Maximize Window
SWS.z
' Wait 1 Second
PAU.1
EIF.
' Window will be restore to normal size
STW.ACct|PBWindowClass:0|TestApp
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.5 "D" - Flag - ITW.

"D" - Flag "restore if minimzed" Previous 3655 Top 41 Next 3657

MiniRobotLanguage (MRL)

"D" - Flag "restore if minimized"

If Top Window

Intention

If the window is minimized, this flag will bring the window back to its normal size.
You can maximize, minimize and restore windows using the SWS. - Set Window State 3889 command.

It does not matter, if the window was maximized before it was minimized.
The window will be normal size after this flag.

Example

```
ITW.Act|PBWindowClass:0|TestApp
 SWS.z
 SWS.m
 PAU.1
EIF.
STW.ADct|PBWindowClass:0|TestApp
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.6 "E" - Flag - ITW.

"E" - Exclude window" Previous 3656 Top 41 Next 3659

MiniRobotLanguage (MRL)

"E" - Flag "Exclude window"

If Top Window

Intention

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you get the window out the "Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

Example

```
' Use Exclude List

' Add window to exclude list
ITW.tE|TestAP
MMV.
VAR.$$HWN=#hwn#

' Remove window from exclude List
ITW.I|$$HWN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' If you comment the ITW. above the window can not be found
here:
ITW.t|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List availabe. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "I" - Flag - ITW. 3665
- "K" - Flag - ITW. 3669

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.7 "F" - Flag - ITW.

"F" - Flag "center window on desktop" Previous 3657 Top 41 Next 3661

MiniRobotLanguage (MRL)

"F" - Flag "Center on Desktop"

Move Window to center of primary desktop

Intention

If the "F"-Flag is given, the window will be move to the center of the primary desktop. The "F"-Flag will override the default windows treatment, which would move the window to 0,0 if its outside of the desktop.
The "F"-Flag will always move the window to the center, no matter where the window was before.

```
ITW.Fc|WindowClass
 MMV.
EIF.
```

Example

```
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
ITW.cF|Myclass
' Move Mouse to Center of window
' in center of desktop
 MMV.
EIF.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.8 "G" - Flag - ITW.

MiniRobotLanguage (MRL)

"G" - Flag "don't move window"

don't move the Top Window

Intention

This flag prevents the default window treatment from moving the window around. Without this Flag, a Top-Window that is outside the working area, will be move to position 0,0, that is the upper left corner of the desktop.

Having the "G"-Flag set, the window will just not be moved, no matter if its outside the working area or not.

```
ITW.cG|WindowClass
 MMV.
EIF.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window if its available
ITW.cG|#32768
...
EIF.
ENR.
```

Remarks

Please note that other prototypes may influence the power of the "G"-Flag, especially, if the are right of the "G"-Flag which means they have a higher priority.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.9 "H" - Flag - ITW.

"H" - Flag "stay in primary desktop" Previous 3661 Top 41 Next 3665

MiniRobotLanguage (MRL)

"H" - Flag "primary desktop"

If Top Window

Intention

This flag will move a window to position 0,0, that is the upper left corner, if the window is full or partial outside the primary desktop workarea.

It will override the default window treatment.
While by default also a Dual-Screen Desktop will be accepted as workarea, this will not be the case here.

The "H"-Flag makes sure the window is complete on the primary desktop.

```
ITW.cH|WindowClass
 MMV.
EIF.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
ITW.tH|word
 MMV.
EIF.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.1("I" - Flag - ITW.

MiniRobotLanguage (MRL)

"I" - Flag "remove from Exclude List"

If Top Window

Intention

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with `ITW.` and not with `STW.` It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
ITW.cE|WindowClass
 MMV.
ELS.
'...
EIF.
```

And how do you get the window out the "Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
ITW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

There is another way, how to search windows which are currently excluded from search?
You can also use the "K"-Flag.

```
' clear whole exclude List
ITW.Kc|WindowClass
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list.

© 2023 Theo Gottwald * https://www.smart-package.com

After this all previously "exclude listed" windows can be found again.

Example

```
' Use Exclude List
' Add window to exclude list
ITW.tE|TestApp
 VAR.$$HWN=#hwn#
EIF.
' Remove window from exclude List
ITW.I|$$HWN
'...
EIF.
' If you comment the ITW. above the window can not be found
here:
ITW.t|TestApp
'...
EIF.
ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag. To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using `SCW`. there is also Exclude-List available. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "E" - Flag - ITW. 3657
- "K" - Flag - ITW. 3669

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.1."J" - Flag - ITW.

MiniRobotLanguage (MRL)

"J" - Flag "window to front"

If Top Window

Intention

The window comes to front already with the default window treatment, without any additional flags. But you can disable that, and select yourself exactly what you want. Like this:

```
ITW.AJc|WindowClass
 MMV.
EIF.
```

The "A" Flag will switch of all default window treatment. Then comes the "J"-Flag and tells the robot to bring the window to front. Be careful with the operator priority. If you place the prototypes like this:

```
ITW.JAc|WindowClass
 MMV.
EIF.
```

then the "A"-Flag will switch of the "J"-Flag because it is more right and therefore has a higher priority.

The "J"-Flag itself will not disable the default window treatment. Therefore something like:

```
ITW.Jc|WindowClass
 MMV.
EIF.
```

```
is possibly a waste of letters, as the default behavior of
ITW. is already to bring the window to front.

Note that the Flag "B" does not switch off the "J"-Flag,
therefore when using "B", the sequence of the prototypes "B"
and "J" should not be a issue.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ITW.AJc|#32768
 ' Locate the button
 MMV.
 MBX. Compare with this:
EIF.
ITW.Ac|#32768
 ' Locate the button
 MMV.
EIF.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.1 "K" - Flag - ITW.

"K" - Flag "Clear Exclude List and search" Previous 3667 Top 41 Next 3671

MiniRobotLanguage (MRL)

"K" - Flag "Clear Exclude List"

Clear Exclude List and search

Intention

Immediately clear Exclude-List before specified search.

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with `ITW.` and not with `STW.` It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
ITW.cE|WindowClass
 MMV.
EIF.
```

And how do you search windows which are currently excluded from search?
This is what the "K"-Flag is good for.

```
' clear whole exclude List
ITW.Kc|WindowClass
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list.

After this all previously "exclude listed" windows can be found again.

Example

```
' Use Exclude List

' Add window to exclude list
ITW.Et|TestApp
 MMV.
 VAR.$$HWN=#hwn#
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Remove all windows from exclude List
' and search

' If you remove the "K" below, the window can not be found!
ITW.Kt|TestApp
 MBX.Hello
EIF.
ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using `SCW`. there is also Exclude-List available. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Currently you can not check which windows are exclude-listed. There is also no way of showing the content of an exclude-list.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "I" - Flag - ITW. 3665
- "E" - Flag - ITW. 3657

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.1 "M" - Flag - ITW.

"M" - Flag "TopMost" Previous 3669 Top 41 Next 3673

MiniRobotLanguage (MRL)

"M" - Flag "TopMost"

If Top Window

Intention

In windows, there is a property of Top-Windows that will make them stay always in front of all other windows.

This property is called "TopMost". A window that is "TopMost" is always on top of all other windows, that are not TopMost.

Only another TopMost window can be in front of an TopMost window.

For example the MR-Editor can be set to TopMost.

Now you may think "this sounds good!". Be warned. This is an advanced feature for advanced users.

Programs have Messageboxes and also open up other windows. In case they have been made TopMost, you may not be able to see those Messages or PopUp windows. This may lead for you to think that the program hangs.

Why don't you see those messages?
Because they will not be TopMost and therefore they will appear behind the TopMost window!

Therefore use TopMost style only when really needed and when you know what you are doing.

Using the Flag "M", the window will be made TopMost immediately after it was localized.

```
ITW.cM|WindowClass
 MMV.
EIF.
```

Use the "N"-Flag to remove the TopMost-property from a TopMost window.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
ITW.Mc|#32768
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$HWN=#hwn#

' Remove the TopMost Property
ITW.hN|$$HWN
 '
ENR.
```

Remarks

-

Limitations:

-

See also:

- "N" - Flag - ITW. 3673

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.14"N" - Flag - ITW.

MiniRobotLanguage (MRL)

"N" - Flag "No TopMost"

No more TopMost

Intention

In windows, there is a property of Top-Windows that will make them stay always in front of all other windows.

This property is called "TopMost".

A window that is "TopMost" is always on top of all other windows, that are not TopMost.

Only another TopMost window can be in front of an TopMost window.

For example the MR-Editor can be set to TopMost.

Now you may think "this sounds good!". Be warned. This is an advanced feature for advanced users.

Programs have Messageboxes and also open up other windows. In case they have been made TopMost, you may not be able to see those Messages or PopUp windows. This may lead for you to think that the program hangs.

Why don't you see those messages?
Because they will not be TopMost and therefore they will appear behind the TopMost window!

Therefore use TopMost style only when really needed and when you know what you are doing.

Using the Flag "M", the window will be made TopMost immediately after it was localized.

```
ITW.cM|WindowClass
 MMV.
EIF.
```

Use the "N"-Flag to remove the TopMost-property from a TopMost window.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
ITW.Mc|#32768
 VAR.$$HWN=#hwn#
EIF.
' Remove the TopMost Property
ITW.hN|$$HWN
'...
EIF.

ENR.
```

Remarks

-

Limitations:

-

See also:

- "M" - Flag - ITW. 3671

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.1 "Q" - Flag - ITW.

"Q" - Flag "Repeat Prototype" Previous 3673 Top 41 Next 3677

MiniRobotLanguage (MRL)

"Q" - Flag "Repeat Prototype"

If Top Window

Intention

This flag will just repeat the Prototype that was last used with `ITW.` or `STW.` or any other Top-Window search instruction.

```
ITW.c|WindowClass
 MMV.
EIF.
' this will do the same
ITW.Q|WindowClass
 MMV.
EIF.
```

Note that there is another way of repeating the last ITW. or ITW.. That is, to just leave the whole parameter part empty.

```
ITW.c|WindowClass
 MMV.
EIF.
' this will do the same
ITW.
 MMV.
EIF.
```

While "Q" will only retype the prototypes internally. Leaving the parameters away, will retype the whole parameters, including the prototypes, using those from last time the command was called.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
ITW.c|WindowClass
 MMV.
EIF.
' this will do the same
ITW.Q|WindowClass
 MMV.
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

`ITW.` and `ITW.` share the same parameter store. Therefore if you have localized the window with `ITW.` then you can also use these parameters for `ITW.` and vice versa.

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.1("V" - Flag - ITW.

"V" - Flag "GoSub if not found. " Previous 3675 Top 41 Next 3679

MiniRobotLanguage (MRL)

"V" - Flag "GoSub if not found"

Gosub if not found

Intention

If the window you are looking for is not found within the specified waiting time, several things can happen. Normally you will get an error-message.

But that does not necessarily be what you want. Maybe you want just to see if the window is there and if not start the program in a Subroutine.

```
' "V" - Flag - on not found Gosub
ITW.ctVw|PBWindowClass:0|BCS|Sub|1
 ' Move Mouse to the Install Button
 SCW.ct|butt|Install
 MMV.
EIF.
ENR.
'-------------------------------------------------------------
' This Subprogram will be called if the window is not found
:Sub
EXE.?exepfad\Sample Apps\BSC-Testform.exe
ITW.ct|PBWindowClass:0|BCS
 '
EIF.
RET.
```

The script above will try to locate the "BCS - Testform" and if its not yet started, it will be started and localized.

Example

```
ITW.ctVw|PBWindowClass:0|TestAppx|Sub|1
 MBX.Ready
EIF.
ENR.
'-------------------------------------------------------------
:Sub
MBX.Error - Program not found
RET.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

You can not mix the "V" and the "W"-Flag. If both are found, the "W"-Flag takes priority

See also:

- "g" - Flag - ITW. 3693
- "o" - Flag - ITW. 3701
- "w" - Flag - ITW. 3716
- "X" - Flag - ITW. 3681

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.17 "W" - Flag - ITW.

MiniRobotLanguage (MRL)

"W" - Flag "GoSub if found"

If Top Window

Intention

If the window you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to first call a Subprogram in that case and then continue with the script. This is done using the "W"-Flag.

```
ITW.ctW|PBWindowClass:0|BCS|Sub
 MMV.
EIF.
ENR.
:Sub
MBX.!
RET.
```

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Example

```
'
' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

' If found close that window.
ITW.ctw|PBWindowClass:0|BCS|1
 CAW.PBWindowClass:0|BCS
 ' Wait until its gone
 MTW.
EIF.

' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

ITW.ct|PBWindowClass:0|BCS
 SCW.ct|butt|Install
 MMV.
EIF.
ENR.
'------------------------------------------------------------
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Limitations:

-

See also:

- "g" - Flag - ITW. 3693
- "o" - Flag - ITW. 3701
- "V" - Flag - ITW. 3677
- "X" - Flag - ITW. 3681

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.16.18 "X" - Flag - ITW.

"X" - Flag "run without error" Previous 3679 Top 41 Next 3683

MiniRobotLanguage (MRL)

"X" - Flag "run without error"

If Top Window

Intention

If the window you are looking for is not found within the specified waiting time, several things can happen. Normally you will just get an error-message. If you do not want to get an error, but just have the script continue running, use the "X"-Flag.

You can use the `#tio#` 3.1 Systemvariables 334 to check if the window was found or not. Also you can use the "V","W","g" and "o" Flags to react inside the `ITW`.

For example in such a script:

```
ITW.cXg|WindowClass|found
 MMV.
EIF.
:found
MPO.
```

This script will invert the normal behavior of `ITW`. It will continue running in case of the window is not found. And in case the window is found, it will jump to the Label "found".

Example

```
'-------------------------------------------------------
' "X"-Flag Sample 1
' Run only if its not yet running ...
'-------------------------------------------------------
'
' Run application only if the window is not found
'
' t - text (caption of the window)
'
NTW.tw|BCS
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.
MBX.Ready
ENR.


'-------------------------------------------------------
' "X"-Flag Sample 2
'-------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not
' yet running.
'
' We use ITW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window
'  we search for
'
ITW.wt|0|BCS - Testform
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.

' Now we can use the Application and press Buttons
ITW.wt|0|BCS - Testform
 SCW.nctb|1|Button|Next|0
 ' Compatible Button Click
 MLC.
EIF.

'------------------------------------------------------------
ENR.
```

Remarks

-

Limitations:

-

See also:

- "g" - Flag - ITW. 3693
- "o" - Flag - ITW. 3701
- "w" - Flag - ITW. 3716
- "V" - Flag - ITW. 3677

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.19 "a" - Flag - ITW.

"a" - Flag "all windows" Previous 3681 Top 41 Next 3685

MiniRobotLanguage (MRL)

"a" - Flag "all windows"

If Top Window

Intention

In most cases, we want to act on visible and enabled window only.
Therefore the !ITW. command ignores by default windows, if they are disabled or invisible.

The "a" flag changes this behavior. Using the "a"-Flag, the ITW. command will locate "all windows". This includes Top-Window which are invisible or disabled, or both.
Using the Flag "a" ALL Top windows will be found if they meet the search criteria.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
ITW.at|BCS - Testform
  EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.

' Next we make the thing invisible
SWS.i
' Wait a second
PAU.1
' Locate it using "all"
ITW.wt|0|BCS - Testform
 ' Locate the Button using "all"
 SCW.anctb|1|Button|Next|0
 ' CLick the button, while its invisible
 MLE.
 ' Get the Top-Handle
 GCT.
 ' Make it visible again
 SWS.v
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

If you definitely know that the window is invisible, you can use the "u" - Flag - ITW. 3712 which will only search for invisible windows and ignore visible windows.

The same applies for disabled windows, and the "f" - Flag - ITW. 3691 .

Limitations:

-

See also:

- "e" - Flag - ITW. 3689
- "f" - Flag - ITW. 3691
- "u" - Flag - ITW. 3712
- "v" - Flag - ITW. 3714

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2("b" - Flag - ITW.

MiniRobotLanguage (MRL)

"b" - Flag "window to back"

window to back

Intention

In most cases we want the window to come in front of other windows. so we can use it. For this we do not need any `ITW.`-flag, as this is the default behavior of the `ITW.`-command.

In some cases, we want just the opposite. We want a window to go behind all other windows. We sent a "window to back". For this we use the "b"-Flag.

It will locate the window "on Top" and sent it to back. See the example below.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

' Open 5 Terstforms and position them nicely
FOR.$$CNT|1|5
  EXE.?exepfad\Sample Apps\BSC-Testform.exe
  ITW.p|#pid#
   SWP.(100+55*$$CNT),(200+55*$$CNT)
  EIF.
NEX.

FOR.$$CNT|1|50
'Move Testform "On top" to back
  ITW.ctb|PBWindowClass:0|BCS - Testform
  ' Make small rest of 0.1 seconds so we can see it
  PAU.0.1
  EIF.
NEX.

' Close all Testforms
CAW.PBWindowClass:0|BCS - Testform
ENR.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2."c" - Flag - ITW.

"c" - Flag "class" Previous 3685 Top 41 Next 3689

MiniRobotLanguage (MRL)

"c" - Flag "class"

If Top Window

Intention

Window-Class" value. This is the "window class" of the window we are looking for.

This flag needs a parameter. It supports 3.2 Standard-Search Pattern 341 and therefore its enough to give a unique part of the value. By default the value is not case sensitive.

```
ITW.c|WindowClass
 MMV.
EIF.
```

Example

```
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
ITW.c|#32768
 ' Locate the button
 SCW.t|Ok
 ' Click the button
 MLE.
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- 3.2 Standard-Search Pattern 341
- "t" - Flag - ITW. 3709

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.27 "e" - Flag - ITW.

"e" - Flag "enabled windows" Previous 3687 Top 41 Next 3691

MiniRobotLanguage (MRL)

"e" - Flag "enabled windows"

If Top Window

Intention

To ignore disabled windows is one of the default settings of the `ITW.` command. Therefore you will not need to specify the "e"-Flag often.

It can however make sense, to override the default behavior by using the "f"-flag which will make `ITW.` to ignore enabled windows and just look for disabled windows.

If you want to catch both, the "a"-flag is what you need.
However the "a"->Flag will automatically also enable the search for invisible windows. If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "e" - flag together with the "a" to search only windows that are enabled. No matter if they are visible or not.

Lets take a look, on "what will work and what will not work".
This will search only for enabled windows.

```
ITW.ce|WindowClass
 MMV.
EIF.
```

Since this is a default behavior of `ITW.`, we have wasted the "e" at this place. Its not needed.

Now we would like to search for windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
ITW.cef|WindowClass
 MMV.
EIF.
```

The robot understands that we want enabled and disabled windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
ITW.cav|WindowClass
 MMV.
EIF.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the invisible windows. We do this by adding the "v"-Flag after the "a". In this case the "v"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
' Start an Application only, if that Application is not yet
running.
'
ITW.wt|BCS - Testform
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.

' Next we disable the whole window
SWS.d
' Wait half of a second
PAU.0.5
' Try to locate it using "e"
' Will not work
ITW.wte|0|BCS - Testform
'---
ELS.
 MBX.We are here.
EIF.

ENR.
```

Remarks

-

Limitations:

-

See also:

- "a" - Flag - ITW. 3683
- "f" - Flag - ITW. 3691
- "u" - Flag - ITW. 3712
- "v" - Flag - ITW. 3714

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.23 "f" - Flag - ITW.

"f" - Flag "disabled windows" Previous 3689 Top 41 Next 3693

MiniRobotLanguage (MRL)

"f" - Flag "disabled windows"

If Top Window

Intention

To ignore disabled windows is one of the default settings of the `ITW.` command. Therefore if you want to search for a window that has been disabled, you will need the "f"-flag.

The "f"-Flag will override the default behavior. It will make `STW.` to ignore enabled windows and just look for disabled windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for invisible windows. If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "f" - flag together with the "a" to search only windows that are disabled. No matter if they are visible or not.

Lets take a look, on this. This will search only for disabled windows.

```
ITW.cf|WindowClass
 MMV.
EIF.
```

Now we would like to search for windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
ITW.cef|WindowClass
 MMV.
EIF.
```

The robot understands that we want enabled and disabled windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
ITW.cav|WindowClass
 MMV.
EIF.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the invisible windows. We do this by adding the "v"-Flag after the "a". In this case the "v"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

–

Remarks

-

Limitations:

-

See also:

- "e" - Flag - ITW. 3689
- "a" - Flag - ITW. 3683
- "u" - Flag - ITW. 3712
- "v" - Flag - ITW. 3714

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2 "g" - Flag - ITW.

"g" - Flag "goto if found" Previous 3691 Top 41 Next 3695

MiniRobotLanguage (MRL)

"g" - Flag "Goto if found"

Goto if found

Intention

If the window you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to jump to a Label in that case.
And you can use the "X"-Flag to tell the robot to just continue with the script in case of error - if the window is not found.
This would change the behavior of the command, but sometimes this can be a good solution. Take this example:

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - goto Label "GoOn" if the window was found
' X - continue the script as normal, without error,
'     if the window is not found
ITW.tgXw|BCS|GoOn|0
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.
:GoOn
MBX.Ready
ENR.
```

In case you use the "g"-Flag, together with the "W" -Flag ("on Found Gosub"), the "W"-Flag will be ignored. The "g"-Flag has a generally higher priority.

Example

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - gotoLabel "Sub" if thewindow was found
' X - continue the script as normal, without error,
'     if the window is not found
ITW.tgXw|BCS|GoOn|0
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
EIF.
:GoOn
MBX.Ready
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- "W" - Flag - ITW. 3679
- "o" - Flag - ITW. 3701
- "V" - Flag - ITW. 3677
- "X" - Flag - ITW. 3681

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2 "h" - Flag - ITW.

"h" - Flag "handle" Previous 3693 Top 41 Next 3697

MiniRobotLanguage (MRL)

"h" - Flag "handle"

If Top Window by handle

Intention

This flag is an exception. In fact it must not search the window, it will just localize the window "by handle". Of course this is a very efficient solution. Therefore if you have already localized your window, keep the handle in a variable and just feed it back to `ITW.` when you need that window again.

```
' Localize by handle
ITW.cta|ieframe|Update
 ' We grab the handle in a variable
 HTV.$$MYW
EIF.

' Later we can always get definitely this window,
' without further parameters
ITW.h|$$MYW
 MMV.
EIF.
```

```
' Some more examples
ITW.h|52367
 MMV.
EIF.

ITW.h|#hnw#
 MMV.
EIF.
```

Example

```
'
' Localize by handle
ITW.cta|ieframe|Update
' We grab the handle in a variable
: $$MYW=#hwn#

' Later we can always get definitely this window,
' without further parameters
ITW.h|$$MYW
 MMV.
EIF.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Using HTV.
STW.t|BCS - Testform
HTV.$$HWN

ITW.h|$$HWN
  DBP.here
EIF.

ENR.
```

Remarks

-

Limitations:

-

See also:

- "E" - Flag - ITW. 3657

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2("j" - Flag - ITW.

"j" - Flag "WII. by number" Previous 3695 Top 41 Next 3699

MiniRobotLanguage (MRL)

"j" - Flag "WII. by number"

Wait for Input Idle and tell how long

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time must be given as a numeric value or variable.

Note the value you specify will replace the default internal value for "WII. without parameter", used for the "k" - Flag - STW. 3865 . But also for WII. if no parameters are given.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions are allowed. For example:

```
' Will wait no longer then a half second
ITW.cj|WindowClass|0.5
 MMV.
EIF.
```

Technically the "j"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement. Just that it is integrated and does not need a separate line of code.

Example

```
' j - wait 5 Seconds maximum
ITW.cj|WindowClass|5
 MMV.
EIF.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

- WII. - Wait-for-Input-Idle 2909
- "k" - Flag - ITW. 3699

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.27 "k" - Flag - ITW.

MiniRobotLanguage (MRL)

"k" - Flag "WII. default"

If Top Window

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time is taken from an internal value that is been set with the last `WII.` statement or with the "j" - Flag - ITW. 3697.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions of a second are allowed.

```
ITW.cj|WindowClass|5
 MMV.
EIF.

' Will do the same without repeating the value
ITW.ck|WindowClass
 MMV.
EIF.
```

Technically the "k"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement without a parameter. Just that it is integrated and does not need a separate line of code.

Example

```
' Demonstrate the "j" and the "k" Flag
ITW.cj|WindowClass|5
 MMV.
EIF.
' Will do the same without repeating the value
ITW.ck|WindowClass
 MMV.
EIF.
ENR.
```

 © 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- "j" - Flag - ITW. 3697
- WII. - Wait-for-Input-Idle 2909

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2 "o" - Flag - ITW.

MiniRobotLanguage (MRL)

"o" - Flag "on not found Goto"

If Top Window

Intention

If the window you are looking for is not found within the specified waiting time, normally you will end up with an error-message.

Maybe you want just to jump at another place in the script the, and start a program. Or react in another way. That is where the "o" Flag enters the game. The script just continues. The window is not found and we tell the robot to jump to a Label in that case.

Take this example:

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
ITW.toXw|BCS|GoOn|0
 GTO.weit
EIF.

:GoOn
 EXE.?exepfad\Sample Apps\BSC-Testform.exe

:weit

MBX.Ready
ENR.
```

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Example

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
ITW.toX|BCS|GoOn
 GTO.weit
EIF.

:GoOn
EXE.?exepfad\Sample Apps\BSC-Testform.exe

:weit

MBX.Ready
ENR.
```

Remarks

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Limitations:

-

See also:

- "W" - Flag - ITW. 3679
- "g" - Flag - ITW. 3693
- "V" - Flag - ITW. 3677
- "X" - Flag - ITW. 3681

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.2 "p" - Flag - ITW.

MiniRobotLanguage (MRL)

"p" - Flag "Process ID"

If Top Window

Intention

You can localize a window safely, if you have the PID (Process ID) of the application behind that window.

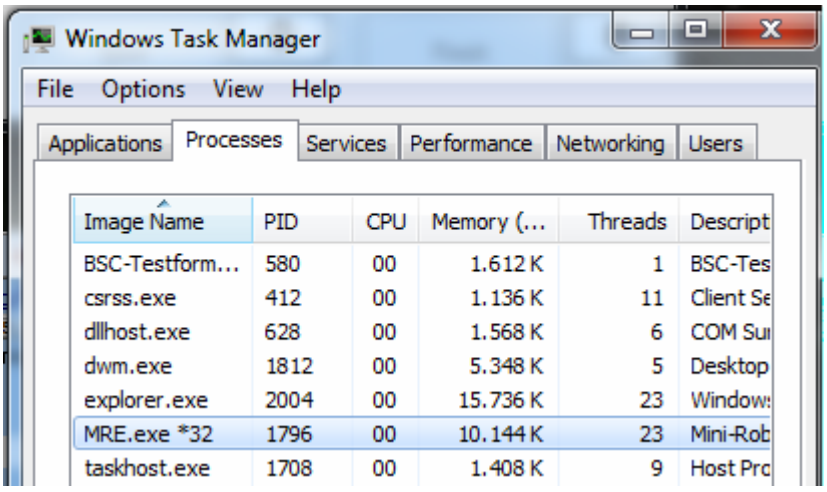
Be sure that you have read and understood the chapter 1.7. Application and PID 137.

Hint: If you have localized a window, you also have the PID of the process behind that window in `#pid#`. You can save it for later use in a variable. Like this:

```
ITW.c|pbwind
: §§PID=#pid#
EIF.

or use:

ITW.c|pbwind
 HTV.§§PID
EIF.
```

In both cases, you get the PID of that process into the local variable §§PID 366.

You can see the PID of a process in the task-manager::



The `PID` value is a normal search parameter and can be used alone or together with other search parameters.

```
' Search a window from process with ID 3435
' and the given window class.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
ITW.ch|pbwind|580
 MMV.
EIF.
```

Example

```
' Locate window only by PID
ITW.p|580
 MMV.
EIF.

ENR.
```

Remarks

-

Limitations:

-

See also:

- 1.7. Application and PID 137
- GDF. - Get-Data-From 2889
- 1.6.1. Program Flow Control 121
- GPI. - Get Process Information 2893
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3("q" - Flag - ITW.

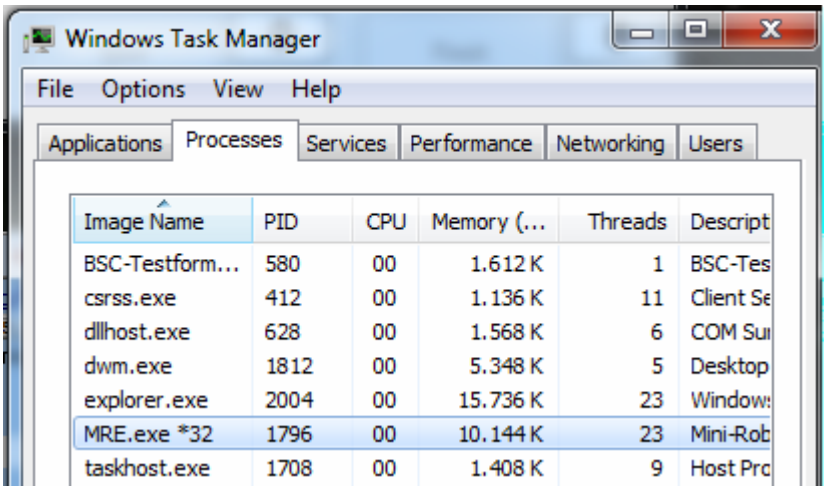
MiniRobotLanguage (MRL)

"q" - Flag "Thread ID"

If Top Window

Intention

This is an advanced parameter. The chances are that you will never need it.
Processes have a number of Threads. See this picture here:



On the left side you can see the name of the process. Right of it the PID of the process. Later you see "Threads".
In a very simplified form, a Thread is a subprogram of that process, that runs an own small program. Generally the more threads a program has, the better it will scale up on multiple CPU-Core's.
While all Threads of a process belong to that process and therefore have the same PID, they have a different Thread-ID. In case you may need to separate between windows of the same process, which is "Multi-threaded", the "q"-flag is for you.

```
ITW.cq|WindowClass|109872
 MMV.
EIF.
```

Hint: If you have localized a window, you also have the TID of the process behind that window in `#tid#`. You can save it for later use in a variable. Like this:

```
ITW.c|pbwind
 : $$TID=#tid#
EIF.
```

This way, you get the TID of that Thread into the local variable §§PID 366.

© 2023 Theo Gottwald * https://www.smart-package.com

If you don't "understand a word", or don't have very special cases, just ignore this flag.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe
' Locate the window
ITW.p|#pid#
' Show the Thread-ID
MBX.#tid#
EIF.

ENR.
```

Remarks

Limitations:

See also:

- 1.7. Application and PID 137
- "p" - Flag - ITW. 3703

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3."r" - Flag - ITW.

"r" - Flag "result number" Previous 3705 Top 41 Next 3709

MiniRobotLanguage (MRL)

"r" - Flag "result number"

If Top Window

Intention

There may be multiple results in a window-search with `ITW.`. Normally you get the first result that fits to your search parameters.
Using the "r"-Flag you can define which result you want, Nr.1, Nr.2 or Nr. 20. Just add the "r"-Flag and the result number.

If omitted, always result Nr.1 is returned.

Note: If there are not enough results, your search will fail!
Therefore you can also use this feature to count windows.

```
'We'll get the 3rd result with that window class
ITW.cr|WindowClass|3
 MMV.
EIF.
```

Example

```
'We'll get the 4th result
' with that window class
'
ITW.cr|WindowClass|4
 MMV.
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3 "t" - Flag - ITW.

"t" - Flag "text" Previous 3707 Top 41 Next 3712

MiniRobotLanguage (MRL)

"t" - Flag "text"

If Top Window

Intention

This flag is one of the most important search parameters. It is the window-text. Most often you can even see the text of a top-window without using the windows-Inspector.
Here is the simplest case how you can grab a window.

```
ITW.t|TestApp
MMV.
```

The text parameter is a string-value and therefore supports 3.2 Standard-Search Pattern 341.
Let us try some of these.

As always with Search patterns, it is enough to give a unique part of the value. By default the text is not case sensitive. You can change that behavior using the search patterns. Take the sample-script below and replace the line:

```
' Locate the window
ITW.t|Testf
```

with the following lines. This will help you to understand the use of Patterns for text search.

```
' Lets start with this
ITW.t|{&DOCAPS:Testf}
```

This will force the text to be case-sensitive! Try this:

```
' This will not find the window
' because of case-sensitivity!
ITW.t|{&DOCAPS:testf}
```

It does not find the window because the &DOCAPS: switches case-sensitivity on.
Now how about this:

```
ITW.t|{&EXACT:bcs - Testform}
```

It does find the window. But then whats the "&EXACT:" good for?
Try this:

```
ITW.t|{&EXACT:bcs - Testf}
```


© 2023 Theo Gottwald * https://www.smart-package.com

It does not find the window because the &EXACT: switches off the default behavior that a part of the text is enough. CAn it be combined with the &DOCAPS: ?

We try. This should work:
```
ITW.t|{&EXACT:&DOCAPS:BCS - Testform}
```

and this should not work:
```
ITW.t|{&EXACT:&DOCAPS:bCS - Testform}
```

Yes, everything work as expected.

Do you know the old times, when you had simple Wildcards as patterns? The"*****" and the "**?**" ?

You can have them back if you want. Try this:

```
ITW.t|{&WILDC:b?? - Testform}
```

Yes, it works. And how about:

```
ITW.t|{&WILDC:b?? - Tes*orm}
or
ITW.t|{&WILDC:b?? - *s*or?}
```

It works.
We kept them for you.
You can still use even the old DOS-Patterns if you really want.
But there is much more if you use the new patterns.

How about a fuzzylogic and a OR'ed combination of two patterns?

```
ITW.t|{&WILDC:b?? }&OR:{BCS-tesform&FUZZY:6}
```

and you can not only combine two patterns, you can combine as many patterns using &OR: as you like! If you want to support many languages, this is the way to go.

No matter if there is a "YES" on that button or a "Qui" or whatever. So long you know it, or at least a bit of it, you can locate it using Search patterns.

Important Note:
Never forget the `EIF.` when using `ITW.`. Of course you can also use `ELS.` We have left this away in the examples above, to explain things shorter.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Locate the window
ITW.t|Testf
 ' Close the window
 CAW.PBWindowClass:0|BCS - Testform
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- 3.2 Standard-Search Pattern 341
- "c" - Flag - ITW. 3687

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3 "u" - Flag - ITW.

"u" - Flag "invisible" Previous 3709 Top 41 Next 3714

MiniRobotLanguage (MRL)

"u" - Flag "invisible"

If Top Window

Intention

To ignore invisible windows is one of the default settings of the `ITW.` command. Therefore if you want to search for a window that has is invisible, you will need the "u"-flag.

The "u"-Flag will override the default behavior. It will make `ITW.` to ignore visible windows and look for invisible windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "u" - flag together with the "a" to search only windows that are invisible. No matter if they are enabled or not.

Lets take a look, on this. This will search only for invisible windows.

```
ITW.cu|WindowClass
 MMV.
EIF.
```

Now we would like to search for windows which are visible, as well as for invisible windows. We can do that in two ways.The first way is like this:

```
ITW.cuv|WindowClass
 MMV.
EIF.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
ITW.cae|WindowClass
 MMV.
EIF.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
CAW.PBWindowClass:0|BCS - Testform
'
' Start an Application only,
' if that Application is not yet running.
'
ITW.wtUX|0|BCS - Testform|found
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
EIF.
ITW.p|#pid#
 ' Next we make the window invisible
 SWS.i
 ' Wait half of a second
 PAU.2.5
EIF.
' We locate it even though its invisible
ITW.wtu|1|BCS - Testform
 ' and make it visible again
 SWS.v
EIF.
ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - ITW. 3689
- "f" - Flag - ITW. 3691
- "a" - Flag - ITW. 3683
- "v" - Flag - ITW. 3714
- 1.8. Selecting the Application (Topwindow) 150

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3 "v" - Flag - ITW.

"v" - Flag "visible" Previous 3712 Top 41 Next 3716

MiniRobotLanguage (MRL)

"v" - Flag "visible"

If Top Window

Intention

To ignore invisible windows is one of the default settings of the `ITW.` command. Therefore you will not need to specify the "v"-Flag often.

It can however make sense, to override the default behavior by using the "u"-flag which will make `ITW.` to visible windows and just look for invisible windows.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also switch on the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "v" - flag together with the "a" to search only windows that are visible. No matter if they are enabled or not.

Lets take a look, on how to do this. This will search only for visible windows.

```
ITW.cv|WindowClass
 MMV.
EIF.
```

Since this is a default behavior of `ITW.`, we have wasted the "v" at this place. Its not needed.

Now we would like to search for windows which are visible, as well for invisibble windows. We can do that in two ways.The first way is like this:

```
ITW.cuv|WindowClass
 MMV.
EIF.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
ITW.cae|WindowClass
 MMV.
EIF.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
CAW.PBWindowClass:0|BCS - Testform
'
' Start an Application only,
' if that Application is not yet running.
'
ITW.wtUX|0|BCS - Testform|found
 EXE.?exepfad\Sample Apps\BSC-Testform.exe
 :found
EIF.
ITW.p|#pid#
 ' Next we make the window invisible
 SWS.i
 ' Wait half of a second
 PAU.2.5
EIF.
' We locate it even though its invisible
ITW.wtu|1|BCS - Testform
 ' and make it visible again
 SWS.v
EIF.
ENR.
```

Remarks

-

Limitations:

-

See also:

- "e" - Flag - ITW. 3689
- "f" - Flag - ITW. 3691
- "a" - Flag - ITW. 3683
- "u" - Flag - ITW. 3712
- 1.8. Selecting the Application (Topwindow) 150

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.16.3 "w" - Flag - ITW.

"w" - Flag "wait time" Previous 3714 Top 41 Next 3718

MiniRobotLanguage (MRL)

"w" - Flag "wait time"

If Top Window

Intention

How long do we wait if a window will appear or not?
Let's try it.

```
: $$TIM=#dtime#
ITW.tX|nix
EIF.
 : $$TIM=#dsince#
 MBX.$$TIM
EIF.
ENR.
```

The answer is that `ITW.` will not wait and immediately continue.
Unless you specify a waiting time.

It's currently about 15 seconds. When you read this it can be another value.
In fact this is the default time that we wait for a window before we run into an error-message or do something else (like in this script).

Now, its very important to understand that we do not generally wait 15 seconds, and then look if the window has appeared.
What we do is, we try ... try ... and try and check permanently, if the window has appeared. If the window appears any time, the script will immediately continue.

If the time is over, we do something else.

Now 15 seconds is enough for some cases, in many cases the window will appear after 45 seconds or any other time. Now this is where the "w"-flag enters the game.

Using the "w"-flag we can tell the robot how long he should wait. It can be 0 seconds like this:

```
ITW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
EIF.
```

or 360 seconds like here.

```
ITW.wtu|360|BCS - Testform
```

and there is no real limitation.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

-

Remarks

-

Limitations:

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.17 IWE. / NWE. - If Window Enabled

MiniRobotLanguage (MRL)

IWE. Conditional Statement

If Windows Enabled

Intention

Conditional Statement.

Test if one or more specified window(s) are enabled or not. If no parameter is given, the actual window is been tested.

You can enable or disable child- or top-windows using the SWS. - Set Window State 3889 .-command.

`IWE.` tells you, if the actual window is enabled or has been disabled. If no parameter is given, the actual window is been tested. If you give parameter, these must be window handles, or the letters "**c**" or "**t**".

These letters stand for "child" and "Top". If given they will be replaced with the last window of that kind that has been localized.

c - last localized child-window

t - last localized Top-Window

Please note, that if the actual window is a child-window (e.g. a button), you need to specify "**t**" as parameter,if you want to test the actual top-window. Some examples:

```
'Get first window handle
STW.ct|CabinetWClass|c:
HTV.$$HAA
'Get second window handle
STW.ct|Chrome_WidgetWin_0|meebo.com
HTV.$$HAB
'Test both handles
IWE.$$HAA|$$HAB
  DBP.All windows are enabled
ELS.
  DBP.At least one of the windows is disabled
EIF.
ENR.
```

Other modes of operation are:

```
' Check actual located window
IWE.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' check last located child window
IWE.c

' check actual Top-window
IWE.t

' check window with handle in $$MYH
IWE.$$MYH
```

NWE. is the negative version "Not Window Enabled".

Syntax

IWE.[P1][|P2]... ELS. ... EIF.

Parameter Explanation

P1 - numeric/variable or "c" or "t". If omitted actual window is taken.

P2 - optional parameters which are also tested for enabled. If one of them is disabled the test fails (implicit=AND). Can be up to 20 additional parameters.

IWE.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'*********************************
' IWE./NWE.-Sample
'*********************************
STW.t|Sample
MMV.
IWE.
 MBX.Ok
ELS.
 MBX.Diabled!
EIF.
END.
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- SWS. - Set Window State 3889
- IWS. / NWS. - If Window State 3724
- IWI. / NWI. - If Window Iconic 3721
- IWV. / NWV. - If Window Visible 3727
- IWZ. / NWZ. - If Window Zoomed 3730

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.18 IWI. / NWI. - If Window Iconic

MiniRobotLanguage (MRL)

IWI. Conditional Statement

If Window Iconized

Intention

Conditional Statement. Test if a specified window is iconic (minimized) or not.
If no parameter is given, the actual window is been tested.

Test if one or more specified window(s) are iconic or not. If no parameter is given, the actual window is been tested.

You can iconize and normalize child- or top-windows using the SWS. - Set Window State 3889 .-command.

If you give parameter, these must be window handles, or the letters "**c**" or "**t**". These letters stand for "child" and "Top". If given they will be replaced with the last window of that kind that has been localized.

c - last localized child-window

t - last localized Top-Window

Please note, that if the actual window is a child-window (e.g. a button), you need to specify "**t**" as parameter,if you want to test the actual top-window. Some examples:

```
'Get first window handle
STW.ct|CabinetWClass|c:
HTV.$$HAA
'Get second window handle
STW.ct|Chrome_WidgetWin_0|meebo.com
HTV.$$HAB
'Test both handles
IWI.$$HAA|$$HAB
  DBP.All windows are iconic
ELS.
  DBP.At least one of the windows is not iconic
EIF.
ENR.
```

Other modes of operation are:

```
' Check actual located window
IWI.

' check last located child window
IWI.c
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' check actual Top-window
IWI.t

' check window with handle in $$MYH
IWI.$$MYH
```

NWI. is the negative version "Not Window Iconic".

Syntax

IWI.[P1][|Px]… ELS. … EIF.

Parameter Explanation

P1 - numeric/variable or "c" or "t". If omitted actual window is taken.

P2 - optional parameters which are also tested for enabled. If one of them is disabled the test fails (implicit=AND). Can be up to 20 additional parameters.

IWI.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'*********************************
' IWI.-Sample
'*********************************
STW.t|Sample
MMV.
IWI.
 MBX.Ok
ELS.
 MBX.Diabled!
EIF.
END.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- SWS. - Set Window State 3889
- IWS. / NWS. - If Window State 3724
- IWE. / NWE. - If Window Enabled 3718
- IWV. / NWV. - If Window Visible 3727
- IWZ. / NWZ. - If Window Zoomed 3730

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.19 IWS. / NWS. - If Window State

MiniRobotLanguage (MRL)

IWS. Conditional Statement

If Window State

Intention

Conditional Statement.

What is a "windows state"?

- windows can be
- enabled, disabled
- minimized, normal
- maximized
- visible or invisible

Test these windows states using IWS.

Using `IWS.` you can test the last localized or any specified child- or top-window for its states.

As you can specify multiple states, you can completely check out a window in one run.

`IWS.` is the counterpart to `SWS.` Using SWS. - Set Window State 3889 you can set the states of a window.

If no parameter is given, the actual window is been tested.

`NWS.` is the negative version "Not Window State".

Syntax

```
IWS.[P1][|P2]... ELS. ... EIF.
```

Parameter Explanation

P1 - prototype

```
  Use the following letters for P1:

' e - enabled
' d - disabled
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' v - visible
' i - invisble
' m - minimzed
' n - normal (not minimized)
' z - zoomed (maximized)
' u - unzoomed
' t - use actual Top-Window instead of actual window
```

If no parameters are specified, "**evn**" is taken to the actual window.
Multiple states can be given. In that case they are connected with AND. All given states MUST be set as given to let the condition evaluate to true.

P2 - (optional) parameter which can be a window-handle.
When given, this window is tested.

Example

```
'**********************************
' IWS./NWS.-Sample
'**********************************
STW.t|Sample
MMV.
IWS.ev
 MBX.Ok
ELS.
 MBX.No
EIF.

STW.t|Sample
HTV.$$HWN
STW.t|anotherwindows
MMV.
' Note that the first window is going to be tested.
IWS.ev|$$HWN
 MBX.Ok
ELS.
 MBX.Diabled!
EIF.
END.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- SWS. - Set Window State 3889
- IWE. / NWE. - If Window Enabled 3718
- IWI. / NWI. - If Window Iconized 3721
- IWV. / NWV. - If Window Visible 3727
- IWZ. / NWZ. - If Window Zoomed 3730

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.20 IWV. / NWV. - If Window Visible

MiniRobotLanguage (MRL)

IWV. Conditional Statement

If Window Visible

Intention

Conditional Statement.

Test if one or more specified window(s) are visible or not. If no parameter is given, the actual window is been tested.

You can make windows (includes child-windows) visible or invisible using the SWS. - Set Window State 3889 .-command.

`IWV.` tells you, if the actual window is visible or has been made invisible. If no parameter is given, the actual window is been tested. If you give parameter, these must be window handles, or the letters "**c**" or "**t**".

These letters stand for "child" and "Top". If given they will be replaced with the last window of that kind that has been localized.

c - last localized child-window

t - last localized Top-Window

Please note, that if the actual window is a child-window (e.g. a button), you need to specify "**t**" as parameter,if you want to test the actual top-window. Some examples:

```
'Get first window handle
STW.ct|CabinetWClass|c:
HTV.$$HAA
'Get second window handle
STW.ct|Chrome_WidgetWin_0|meebo.com
HTV.$$HAB
'Test both handles
IWV.$$HAA|$$HAB
  DBP.All windows are visible
ELS.
  DBP.At least one of the windows is invisible
EIF.
ENR.
```

Other modes of operation are:

```
' Check actual located window
IWV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' check last located child window
IWV.c

' check actual Top-window
IWV.t

' check window with handle in $$MYH
IWV.$$MYH
```

NWV. is the negative version "Not Window Visible".

Syntax

IWV.[P1][|P2...|P20]… ELS. … EIF.

Parameter Explanation

P1 - numeric/variable or "c" or "t". If omitted actual window is taken.

P2 - optional parameters which are also tested for enabled. If one of them is disabled the test fails (implicit=AND). Can be up to 20 additional parameters.

IWV.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'*********************************
' IWV.-Sample
'*********************************
STW.t|Sample
MMV.
IWV.
 MBX.Ok
ELS.
 MBX.No
EIF.
END.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

-

See also:

- SWS. - Set Window State 3889
- IWS. / NWS. - If Window State 3724
- IWV. / NWE. - If Window Enabled 3718
- IWI. / NWI. - If Window Iconized 3721
- IWZ. / NWZ. - If Window Zoomed 3730

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.21 IWZ. / NWZ. - If Window Zoomed

IWZ. / NWZ. - If Window Zoomed (maximized) Previous 3727 Top 41 Next 3733

MiniRobotLanguage (MRL)

IWZ. Conditional Statement

If Window Zoomed

Intention

Conditional Statement.

Test if one or more specified window(s) are zoomed (maximized) or not. If no parameter is given, the actual window is been tested.

You can enable or disable child- or top-windows using the SWS. - Set Window State 3889 .-command.

`IWZ.` tells you, if the actual window is zoomed or not. If no parameter is given, the actual window is been tested. If you give parameter, these must be window handles, or the letters "**c**" or "**t**".

These letters stand for "child" and "Top". If given they will be replaced with the last window of that kind that has been localized.

c - last localized child-window

t - last localized Top-Window

Please note, that if the actual window is a child-window (e.g. a button), you need to specify "**t**" as parameter,if you want to test the actual top-window. Some examples:

```
'Get first window handle
STW.ct|CabinetWClass|c:
HTV.$$HAA
'Get second window handle
STW.ct|Chrome_WidgetWin_0|meebo.com
HTV.$$HAB
'Test both handles
IWZ.$$HAA|$$HAB
  DBP.All windows are enabled
ELS.
  DBP.At least one of the windows is disabled
EIF.
ENR.
```

Other modes of operation are:

```
' Check actual located window
IWZ.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' check last located child window
IWZ.c

' check actual Top-window
IWZ.t

' check window with handle in $$MYH
IWZ.$$MYH
```

NWZ. is the negative version "Not Window Zoomed".

Syntax

IWZ.[P1][|P2...|P20]… ELS. … EIF.

Parameter Explanation

P1 - numeric/variable or "c" or "t". If omitted actual window is taken.

P2 - optional parameters which are also tested for enabled. If one of them is disabled the test fails (implicit=AND). Can be up to 20 additional parameters.

IWZ.'s can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Example

```
'**********************************
' IWZ.-Sample
'**********************************
STW.t|Sample
MMV.
IWZ.
 MBX.Ok
ELS.
 MBX.No!
EIF.
END.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

-

See also:

- SWS. - Set Window State 3889
- IWS. / NWS. - If Window State 3724
- IWZ. / NWE. - If Window Enabled 3718
- IWI. / NWI. - If Window Iconized 3721
- IWV. / NWV. - If Window Visible 3727
- IWZ. / NWZ. - If Window Zoomed 3730

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.22 MAW. - Missing-Active-Window

MiniRobotLanguage (MRL)

MAW. Command

Missing Active Window

Intention

Stops script execution at the given Line until the last localized window disappears (is closed). The command has possible prototypes for advanced usage.

Syntax

MAW.[P1][|P2][|P3]

Parameter Explanation

P1 - optional: Prototypes

```
 h - handle, if specified, you need to supply the window-
handle
     of the window to wait for.

 t - timeout, if specified the command will only wait the
specified
     amount of time in seconds and then go on with the next
command.
     The timeout flag is set in that case, otherwise its
cleared.

 p - Top-Window from actual window has to be closed..

 a - continues if the window becomes invisible or disabled.

 v - continues if the window becomes invisible.

 e - continues if the window becomes disabled.
```

P2 - optional: Variable
P3 - optional: time to pause in seconds

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*********************************************
' This Example will locate a Window and wait till
' it is removed.
' Please note that you need to have such a window.
'*********************************************
DBM.2
STW.c|#32770
MAW.
PRT. Window was removed
DMP.
END.

'*********************************************
' This Example will locate a Window and wait till
' it is removed. It will use a timeout of 5 Seconds.
' Please note that you need to have such a window.
'*********************************************
DBM.2
STW.c|#32770
HTV.$$MYW
' Locating another window
STW.c|ieframe
' We use the handle therefore we wait
' until the #32770 window is been closed.
MAW.ht|$$MYW|5
PRT. Window was removed or timeout happened
DMP.
END.

'*********************************************
' This Example will wait till the IE 6 is ready
'*********************************************
STW.c|ieframe
SCW.cw|explorer_server|360
SCW.cdw|msctls_progress32|2|160
HTV.§§HTC
MAW.v
MBX.IE is ready.
END.
```

Remarks

-

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- CLW. - Close-Window 3592
- CAW. - Close-All-Windows 3586

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.23 SAW. - Set Active Window

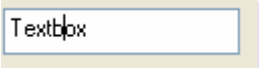
MiniRobotLanguage (MRL)

SAW. Command

Set Active Window

Intention

The SAW.-Command will give an edit control the input-focus. As a result you may see a blinking cursor inside.



The input focus can only be in one window/control at the same time.
If you type something on the Keyboard, the control with the input focus will reveive the typing. The same happens, if you simulate typing using SKP. Send-Key-Press 2289 and such commands. That's why SAW. can be used before you type into a textbox.

When using the Direct Engine 231 (commands for example: RST. and SDT.) the input-focus is not needed and therefore SAW. is not needed.

In most cases you will not give a parameter to SAW., but just locate the child window before using ! SCW - Search-Child-Window 3738 .

You can however add a variable with a window handle from a child-window.
The the robot tries to give the input-focus to that child-window.

SAW. will also try to bring the window to front, if it is behind other windows.
Usage is simple. Just write:

```
STW.Act|PBWindowClass:0|TestApp
SCW.cn|edit|2
SAW.
```

Syntax

SAW.[P1]

Parameter Explanation

P1 **-** ***(optional)*** VAR or formula in (..).
Must be a valid Window handle

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
STW.AgXctw|over|PBWindowClass:0|TestApp|0
EXE.?exepath\Sample Apps\TestApp_01.exe
:over
'Move pointer to Window-Area.
STW.Act|PBWindowClass:0|TestApp
SCW.cn|edit|2
SAW.
ENR.
```

Remarks

To move a Application window (Top-window) to front, use SFW. - Set Foreground WIndow3812. or ! STW. -Search Top-Window3814.

Limitations:

-

See also:

- SFW. - Set Foreground WIndow3812
- SWS. - Set Window State3889
- ! STW. -Search Top-Window3814
- ! SCW - Search-Child-Window3738
- WII. - Wait-for-Input-Idle2909
- SKP. Send-Key-Press2289
- SKT. - Send-Key-Text2293

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24 SCW. - Search-Child-Window

3.83.24.1 ! SCW. - Search-Child-Window

MiniRobotLanguage (MRL)

SCW. Command

Seach Child Window

Intention

This command localizes a specified Child-Window (Control, Button etc.) within a previous localized Top-Window (Application-Window).

To understand this command, be sure to read the chapters:
1.8. Selecting the Application (Topwindow) 150
and
1.9. Locating Child-Windows 162
first.

It will then sets the Mouse-Target Point into the window-center.

`SCW.` has a lot of features, which are controlled using the 1.4. The Prototype Parameter System 73 .Be sure you have read and understand that chapter before trying to understand `SCW.`

Unless using the "T"-Option, the `SCW.`-command needs a Top-Window being localized first. Therefore please read the section 1.8. Selecting the Application (Topwindow) 150 carefully.

The Command has a "Rollback" Feature, when the Child-Window is not found in few Seconds. It jumps back to the `STW.` (Search Top-Window) and tries to get a new Top-Window, which possibly has the wanted button or control.
This will continue, until its time has run out.
If you use the `DBM.1` -command, you can see these "Rollback-Jumps" in the Debugging-Console. See the example below.

```
DBM.1
STW.Act|PBWindowClass:0|TestApp
' This button/childwindow does not exist
SCW.nctb|2|Button|Button 23|.5
MLE.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
[10:13:11] STW.Act|PBWindowClass:0|TestApp
[10:13:11] SCW.nctb|2|Button|Button 23|.5
[10:13:12] STW.Act|PBWindowClass:0|TestApp
[10:13:12] SCW.nctb|2|Button|Button 23|.5
[10:13:12] STW.Act|PBWindowClass:0|TestApp
[10:13:12] SCW.nctb|2|Button|Button 23|.5
[10:13:13] STW.Act|PBWindowClass:0|TestApp
[10:13:13] SCW.nctb|2|Button|Button 23|.5
```

You can control this Feature using the "b" - Backjump-Time Flag 3764.

Syntax

SCW.[Prototypes|][Parameters|]

Parameter Explanation

Parameter

```
 a - (np) "all" - search all windos, includes
     invisible and disabled windows.
 b - (p) "backjump"-flag, time in seconds, 0 disables the jump
 c - (p)  window-class
 d - (p) window-depth
 e - (np) enabled, window must be enabled
 f - (np) disabled, window must be disabled
 g - (p) Goto on found
 h - (p) handle. Immediately locate the window with given
handle.
     All other prototypes will be ignored.
 i - (p) window-id
 j - (p) WaitForInputIdle (Value)
 k - (np) WaitForInputIdle (no Value), use value
      from last Call/default-value
 n - (p) Fensternumber in der Ebene ("WR->SWC")
 o - (p) on not found goto (Label)
 p - (p) Process-ID (PID 137)
 q - (p) Thread-ID (TID)
 r - (p) result number ("WR-SPC-Count")
 s - (np) internal reversing at child windows search
 t - (p) text
 u - (np) invisible, window must be invisible
 v - (np) visible, window must be visible
 w - (p) total waiting-time in Seconds, fractions (0.5)
     are possible, zero is possible.

 E - (np) Add Window to Exclude List after localization
 I - (p) p = handle. Locate window and remove window
     from Exclude list.
 K - (n) Immediately clear Exclude-List before specified
search
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Q - (n) use previous Prototypes
S - (n) Search only next Level (SWC-Option)
T - (n) Search desktopwide, ignore localized Top-Window
V - (p/Label) Gosub on not found
W - (p/Label) Gosub on  found
X - (n) Run without Error on not found
```

Explanation:
(np) -> only prototype, no additional parameter required
(p) -> prototype requires additional parameter
(NUM/VAR) - window handle/variable
(p/Label) - Label-Parameter required, can also be a variable that contains the label

P2 **-** Parameters, separated by | according to the prototypes

Example

```
STW.Act|PBWindowClass:0|TestApp
SCW.nctE|2|Button|Button 2
' Coordinates relative to left, upper corner of button
MPO.o|42,22
MLE.
```

Remarks

When using `SCW.` to check if a child-window is available or not, it makes sense that you use the "b"-flag with a value of "0" to disable the backjump.

Limitations:

-

See also:

- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- 1.5. Features and Hints 150
- 1.9. Locating Child-Windows 162
- 1.8. Selecting the Application (Topwindow) 150
- 1.7. Application and PID 137
- 3.1 Systemvariables 334
- !. STW. - Description 3814

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "E" - Flag - SCW.

MiniRobotLanguage (MRL)

"E" - Flag "Exclude window"

Search Child Window

Intention

Imagine you have several equal buttons or controls, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number), or the "n"-Flag "leveled window number".

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". In fact there is one such list for Top-windows and there is one such list for child-windows.

Once in that list, a window can not be found by STW. or SCW. in a window search.

Now if you add a window to this "Exclude List", it will not be found any more. Not with SCW. and not with ICW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.c|WindowClass
' Will add a button to the Exclude-List
SCW.cE|button
MMV.
```

And how do you remove a window from the "Search Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

Can you clear the whole list with one strike?

Yes, you can also use the "K"-Flag.

```
' clear whole exclude List
STW.c|WindowClass
SCW.K
```


© 2023 Theo Gottwald * https://www.smart-package.com

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list. Unlike the "I"-Flag, the "K"-Flag can be combined with any other window search.

After this all previously "exclude listed" windows can be found again.

Short Examples:

-

Example

-

Remarks

Windows which have been closed, and all their child-windows, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Top-Windows using STW. there is also Exclude-List available. The Exclude List for Top-Windows is a separate list and will therefore survive if you clear the exclude list for Child-Windows.

Currently you can not check which windows are exclude-listed. There is also no way of showing the content of an exclude-list.

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "I" - Flag - SCW. 3743
- "K" - Flag - SCW. 3745

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.3 "I" - Flag - SCW.

MiniRobotLanguage (MRL)

"I" - Flag "remove from Exclude List"

Search Child Window

Intention

Imagine you have several equal buttons or controls, which you want to handle one after the other.

Yes, there are features available to do that, for example the "r"-Flag (Window-number), or the "n"-Flag "leveled window number".

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". In fact there is one such list for Top-windows and there is one such list for child-windows.

Once in that list, a window can not be found by STW. or SCW. in a window search.

Now if you add a window to this "Exclude List", it will not be found any more. Not with SCW. and not with ICW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.c|WindowClass
' Will add a button to the Exclude-List
SCW.cE|button
MMV.
```

And how do you remove a window from the "Search Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

Can you clear the whole list with one strike?

Yes, you can also use the "K"-Flag.

© 2023 Theo Gottwald * https://www.smart-package.com

```
' clear whole exclude List
STW.c|WindowClass
SCW.K
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list. Unlike the "I"-Flag, the "K"-Flag can be combined with any other window search.

After this all previously "exclude listed" windows can be found again.

Short Examples:

-

Example

-

Remarks

Windows which have been closed, and all their child-windows, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Top-Windows using STW. there is also Exclude-List available. The Exclude List for Top-Windows is a separate list and will therefore survive if you clear the exclude list for Child-Windows.

Currently you can not check which windows are exclude-listed. There is also no way of showing the content of an exclude-list.

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "E" - Flag - SCW. 3741
- "K" - Flag - SCW. 3745

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.4 "K" - Flag - SCW.

"K" - Flag "Clear Exclude List and search" Previous 3743 Top 41 Next 3748

MiniRobotLanguage (MRL)

"K" - Flag "Clear Exclude List"

Search Child Window

Intention

Immediately clear Exclude-List before specified search.

Imagine you have several equal buttons or controls, which you want to handle one after the other.

Yes, there are features available to do that, for example the "r"-Flag (Window-number), or the "n"-Flag "leveled window number".

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". In fact there is one such list for Top-windows and there is one such list for child-windows.

Once in that list, a window can not be found by STW. or SCW. in a window search.

Now if you add a window to this "Exclude List", it will not be found any more. Not with SCW. and not with ICW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.c|WindowClass
' Will add a button to the Exclude-List
SCW.cE|button
MMV.
```

And how do you remove a window from the "Search Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

Can you clear the whole list with one strike?

Yes, you can also use the "K"-Flag.

© 2023 Theo Gottwald * https://www.smart-package.com

```
' clear whole exclude List
STW.c|WindowClass
SCW.K
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list. Unlike the "I"-Flag, the "K"-Flag can be combined with any other window search.

```
' clear whole exclude List
STW.c|WindowClass
SCW.Knct|1|Button|Button 2
```

After this all previously "exclude listed" windows can be found again.

Short Examples:

-

Example

-

Remarks

Windows which have been closed, and all their child-windows, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Top-Windows using STW. there is also Exclude-List available. The Exclude List for Top-Windows is a separate list and will therefore survive if you clear the exclude list for Child-Windows.

Currently you can not check which windows are exclude-listed. There is also no way of showing the content of an exclude-list.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "E" - Flag - SCW. 3741
- "I" - Flag - SCW. 3743

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.5 "Q" - Flag - SCW.

"Q" - Flag "Repeat Prototype" Previous 3745 Top 41 Next 3750

MiniRobotLanguage (MRL)

"Q" - Flag "Repeat Prototype"

Search Child Window

Intention

This flag will just repeat the Prototype that was last used with SCW. or ICW.

```
STW.c|WindowClass
SCW.ct|button|Next
MMV.
' this will do the same
STW.Q|WindowClass
SCW.Q|button|Next
MMV.
```

Note that there is another way of repeating the last SCW. or ICW.. That is, to just leave the whole parameter part empty.

```
STW.c|WindowClass
SCW.ct|button|Next
MMV.

' this will do the same
STW.
SCW.
MMV.
```

While "Q" will only retype the prototypes internally. Leaving the parameters away, will retype the whole parameters, including the prototypes, using those from last time the command was called.

Short Examples:

-

Example

-

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.6 "S" - Flag - SCW.

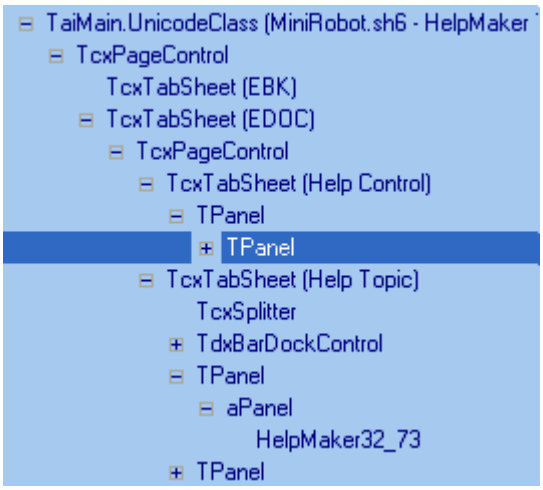
MiniRobotLanguage (MRL)

"S"-Flag "only next level"

Search Child Window

Intention

This is an advanced command. When locating child windows that are hidden in a child-window tree, with multiple branches, the recording programs will sometimes record the way "how to get there". Take a look at this tree:



The child-window is somewhere deep in a tree, we can not just locate it like this:

```
SCW.c|tpanel
```

because there are multiple child windows with that class. Also we have no valid child-window ID. And we can not really use the text ("t"-flag) because this child window does not offer a text for us. Here is a way to get there:

```
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.Sct|TcxPageControl|{&NOTEXT:}
SCW.Sct|TcxTabSheet|EDOC
SCW.Sct|TcxPageControl|{&NOTEXT:}
SCW.Sct|TcxTabSheet|Help Topic
SCW.Snct|2|TPanel|{&NOTEXT:}
MBX.#hwn#
ENR.
```

Using the "S"-Flag we can build a copy of the window tree in our script. This way, we can really locate the child window that we are looking for. Alternative you can also use the "n"-Flag which has a similar effect and allows numbering of the results.

© 2023 Theo Gottwald * https://www.smart-package.com

Short Examples:

–

Example

–

Remarks

-

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "n" - Flag - SCW. 3789
- "r" - Flag - SCW. 3797

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.7 "T" - Flag - SCW.

"T" - Flag "(total search) Search desktopwide" Previous 3750 Top 41 Next 3754

MiniRobotLanguage (MRL)

"T"-Flag "total, desktopwide"

Search Child Window

Intention

There are cases when you may not want to specify the Top-Window. In these cases you can directly search for a specific child-window using the "T"-flag.

The "T"-Flag searches "desktopwide" in all windows for the specified control, child-window.

Short Examples:

```
SCW.Tctr|button|cancel|1
MMV.
ENR.
```

Example

-

Remarks

-

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "n" - Flag - SCW. 3789
- "r" - Flag - SCW. 3797
- "p" - Flag - SCW. 3793

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.8 "V" - Flag - SCW.

"V" - Flag "GoSub if not found. " Previous 3752 Top 41 Next 3756

MiniRobotLanguage (MRL)

"V" - Flag "GoSub if not found"

Search Child Window

Intention

If the child-window (button, control etc.) is not found within the specified waiting time, several things can happen. Normally you will get an error-message and the script will end.

But that does not necessarily be what you want. Maybe you want just to see if the window is there and if not start a Subroutine to do something.

```
' "V" - Flag - on not found Gosub
STW.ctw|PBWindowClass:0|BCS|1
' This button is not found
' so we try anotehr one in the subprogram
SCW.ctVw|Button|InstaX|Sub|0
' Move Mouse to the Install Button
MMV.
ENR.
'-------------------------------------------------------------
' This Subprogram will be called if the window is not found
:Sub
' Try another button
SCW.ct|Button|Install
MLC.
RET.
```

The script above will try to locate the "BCS - Testform" and if its not yet started, it will be started and localized.

Short Examples:

```
' Code example for "V"-Flag (on not found GoSub)
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.wSctV|0|TPanelx|{&NOTEXT:}|notf
MBX.#hwn#
ENR.

:notf
MBX.did not find this one!
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

_

Remarks

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "g" - Flag - SCW. 3777
- "o" - Flag - SCW. 3791
- "w" - Flag - SCW. 3809

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.9 "W" - Flag - SCW.

"W" - Flag "GoSub if found" Previous 3754 Top 41 Next 3758

MiniRobotLanguage (MRL)

"W" - Flag "GoSub if found"

Search Child Window

Intention

If the child- you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to first call a Subprogram in that case and then continue with the script. This is done using the "W"-Flag.

```
STW.ct|PBWindowClass:0|BCS
SC.ctW|Button|Install|Sub
MMV.
ENR.
:Sub
MBX.!
RET.
```

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Short Examples:

```
'
' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

' If found close that window.
STW.ctWw|PBWindowClass:0|BCS|Sub|1

' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

STW.ct|PBWindowClass:0|BCS
SCW.ct|butt|Install
MMV.

ENR.
'------------------------------------------------------------
:Sub
CAW.PBWindowClass:0|BCS
' Wait until its gone
MTW.
RET.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

_

Remarks

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "o" - Flag - SCW. 3791
- "g" - Flag - SCW. 3777
- "V" - Flag - SCW. 3754

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.1("X" - Flag - SCW.

MiniRobotLanguage (MRL)

"X" - Flag "run without error"

Search Child Window

Intention

If the child-window you are looking for is not found within the specified waiting time, several things can happen. Normally you will just get an error-message. If you do not want to get an error, but just have the script continue running, use the "X"-Flag.

You can use the `#tio#` 3.1 Systemvariables 334 to check if the window was found or not. Also you can use the "V","W","g" and "o" Flags to react inside the STW.

For example in such a script:

```
STW.c|WindowClass
' This button does not exist
SCW.Xcw|ButtonX|0
' We'll just move the mouse inside the Top-Window
MMV.

ENR.
```

This script will invert the normal behavior of STW. It will continue running in case of the window is not found. And in case the window is found, it will jump to the Label "found".

Short Examples:

```
'-------------------------------------------------------
' "X"-Flag Sample 1
' Run only if its not yet running ...
'-------------------------------------------------------
'
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - gotoLabel "Sub" if thewindow was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'-------------------------------------------------------
' "X"-Flag Sample 2
'-------------------------------------------------------
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' g ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
'
STW.wtgX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Now we can use the Application and press Buttons
STW.wt|0|BCS - Testform
SCW.nctb|1|Button|Next|0
' Compatible Button Click
MLC.

'-------------------------------------------------------------
ENR.
```

Remarks

You can use the `#tio#` 3.1 Systemvariables 334 to check if the window was found or not. Also you can use the "V","W","g" and "o" Flags to react inside the SCW.

Limitations:

See also:

- 1. STW. - Decription 3814

© 2023 Theo Gottwald * https://www.smart-package.com

- SCW - Search-Child-Window 3738
- 3.1 Systemvariables 334

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.1 "a" - Flag - SCW.

"a" - Flag "all child-windows" Previous 3758 Top 41 Next 3764

MiniRobotLanguage (MRL)

"a" - Flag "all child-windows"

Search Child Window

Intention

In most cases, we want to act on visible and enabled child-window (buttons, controls ...) only. Therefore the SCW. command ignores by default child-windows, if they are disabled or invisible.

The "a" flag changes this behavior. Using the "a"-Flag, the SCW. command will locate "all child-windows". This includes child-windows which are invisible or disabled, or both.
Using the Flag "a" all child-windows will be found if they meet the search criteria.

Example

```
'*********************************************
' SCW-Sample 1
'*********************************************
DBM.1

' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.Act|PBWindowClass:0|TestApp
SCW.ct|Button|Button 1
HTV.$$MYC
' Make button invisible
SWS.i

' Show handle of the childwindow
DBP.$$MYC

' Locate Topwindow again with previous parameters
GCT.
' Locate Button even though its invisible
SCW.wcta|0|Button|Button 1

FOR.$$CNT|1|10
' Press that button 10 times using direct engine
  MLE.
NEX.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PAU..5

' Make the button visible again
SCW.h
SWS.v
ENR.

'*********************************************
' SCW-Sample 2
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

If you definitely know that the child-window is invisible, you can use the "u"-Flag. 3878 which will only search for invisible child- windows and ignore visible child-windows.

The same applies for disabled windows, and the "f"-Flag.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.1 "b" - Flag - SCW.

"b" - Flag "backjump" Previous 3761 Top 41 Next 3767

MiniRobotLanguage (MRL)

"b"-Flag "backjump"

Search Child Window

Intention

What is the "**backjump**"?
It's some kind of "**rollback**", that is done, when a child-window is not immediately found.
If a Child-window (for example a button) is not found, there is a slight danger, that we are inside the wrong Top-Window. Because the SCW. always searches inside the current Top-Window, this behavior could lead to problems. Even if a fitting window with our button would appear, we could not grab it, because we are only searching in a previous window that had the right title.
To prevent this problem and also check new windows if they appear, the SCW.-command may initiate a so called "Backjump" to the previous STW. A Backjump means that the SCW. jumps back to the STW. and this way looks for new windows that may potentially have the button we are looking for.

The "b"-flag controls the "Backjump" behavior. It describes the backjump-time in seconds. You can specify fractions of seconds like this "0.5".
And you can completely disable the backjump, if you set the "b"-flag to a value of zero.

Short Examples:

```
' We use just the button caption and the backjump-flag
SCW.tb|Cancel|0
SCW.bt|0|Cancel

' Here we set the backjump to 250 ms
' all three commands do technically the same
' we have just changed the order of the parameters
SCW.ctb|Button|Cancel|250
SCW.tbc|Cancel|250|Button
SCW.btc|250|Cancel|Button
```

Example

```
'MR Script-file:
'Version:
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'Purpose: STW.-Features Demo
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
' In the following script we'll make three sorts of backjumps.
' Fast, slow and no backjump.
' Run it from the Editor and follow the script.
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Close Testform if already there
CAW.PBWindowClass:0|BCS - Testform

' Wait 200 ms
PAU.0.2

' Start an Application only, if that Application
' is not yet running.
'
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' U ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Now we can use the Application and press Buttons
' we use the following flags:
' w ... 0 - set waiting time to zero
' t ... "BCS - Testform" - caption of the window we are
looking for
STW.wt|0|BCS - Testform
'
' We use the following flags in SCW.:
'
' "c" ... "Button" - window-class (its a button control)
' "t" ... "Cancel" - text on the button we search for
' "b" ... "backjump" is set to 120 ms
' "X" - Flag - no error if not found
' "w" - set waiting time to 5 seconds
SCW.ctbXw|Button|notthere|120|5
' Just move to the button, no click.

' We make another STW. or else the Backjump would go to the
STW. in Line 34
STW.wt|0|BCS - Testform
' We set the backjump now to 1 second (=1000 ms)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
SCW.ctbXw|Button|notthere|1000|5

' We make another STW. or else the Backjump would go to the
STW. in Line 34
STW.wt|0|BCS - Testform
' We disable the backjump now
SCW.ctbXw|Button|notthere|0|5

MBX. Ready
'-------------------------------------------------------------
ENR.
```

Remarks

You can specify rollback times smaller then a second. Use in that case floating point notation, like "0.25".

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.1 "c" - Flag - SCW.

"c" - Flag "class" Previous 3764 Top 41 Next 3769

MiniRobotLanguage (MRL)

"c"-Flag "class"

Search Child Window

Intention

Child-Window-Class" value. This is the "window class" of the child-window (button,control etc.) we are looking for. Some typical window classes for child-windows are:

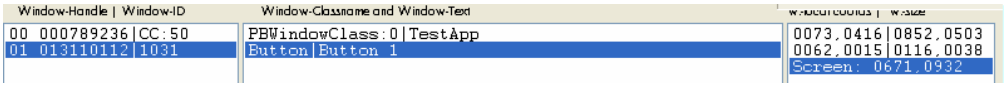
- button (button or checkmark)
- edit (editable text)
- static (static text)
- systabcontrol32 (Tab-Control)
- systreeview32 (Tree-Control)

But there are many more, see example below.

This flag needs a parameter. It uses the 1.4. The Prototype Parameter System 73.
As all text-parameters, the "class" parameter supports the 3.2 Standard-Search Pattern 341.
Therefore you do not always need to type the whole class string.
Its enough to give a unique part of the value. By default the value is not case sensitive.

How do you get the window-class?

If you drive with the mouse over the button or control, then you can see the class as well as other informations about the control in the "windows inspector". See example picture below. The Inspector is integrated in the MR-Editor. Possibly you need to switch the "windows inspector" on before he appears at the bottom of the Editor..



Use the class flag most often together with other flags, like the "t"-flag, to locate the Child-Window of your choice.

```
STW.c|WindowClass
SCW.ct|Button|Ok
MMV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Short Examples:

```
SCW.ct|button|Cancel|0
```

Example

```
' c - class
' n - number
' t - text
  STW.ct|Tai0470.UnicodeClass|Power Editor
  SCW.nct|1|TPanel|{&NOTEXT:}
  SCW.nct|1|TeePageControl|{&NOTEXT:}
  SCW.nct|1|TeeTabSheet|{&NOTEXT:}
  SCW.nct|2|TPanel|{&NOTEXT:}
  SCW.nct|1|TAdvGlowButton|Apply
  MLE.
```

Remarks

-

Limitations:

See also:

- SCW - Search-Child-Window 3738
- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- "t" - Flag - SCW. 3801

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.14 "d" - Flag - SCW.

MiniRobotLanguage (MRL)

"d"-Flag "depth"

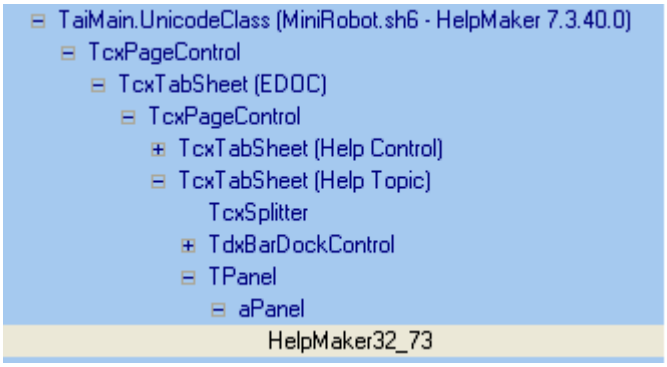
Search Child Window

Intention

This is the "depth" parameter. It needs a numeric value according to the 1.4. The Prototype Parameter System 73.

What is the "depth" of a child-window?

Child-windows are organized below a Top-Window because that's what makes the Child-windows. They belong to a Top-Window from an application. If that would be the whole story, all child-window would have a depth of 1.
Because that means that they are in the next level below the Top-Window. Now in reality a child window can also be below another child window. It can even for a "tree of child-windows". See the example in the picture below.



In this example, this would be some of the the window depths:

```
Top-Level: TaiMain.UnicodeClass | MiniRobot.sh6
1. TcxPageControl | {&NOTEXT:}
2. TcxTabSheet | EDOC
3. TcxPageControl | {&NOTEXT:}
4. TcxTabSheet | Help Topic
5. TPanel | {&NOTEXT:}
```

Therefore we could try to locate the **`TcxTabSheet|Help`** Topic like this:

```
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.nct|1|TcxPageControl|{&NOTEXT:}
SCW.nct|1|TcxTabSheet|EDOC
SCW.nct|1|TcxPageControl|{&NOTEXT:}
SCW.nct|1|TcxTabSheet|Help Topic
MBX.#hwn#
```


© 2023 Theo Gottwald * https://www.smart-package.com

but if we use the depth parameter, we could try this shorter version:

```
STW.ct|TaiMain.UnicodeClass|MiniRobot.sh6 - HelpMaker 7.3.40.0
SCW.dct|4|TcxTabSheet|Help Topic
MBX.#hwn#
```

which will also locate the right window.

In case there are more windows of the same window-class in the same depth, use the "n"-flag to have them numbered. Then select the right number to get the child window you want.

Short Examples:

-

Example

'

Remarks

-

Limitations:

The robot will search windows up to any depth.

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.1 "e" - Flag - SCW.

MiniRobotLanguage (MRL)

"e"-Flag "enabled"

Search Child Window

Intention

The picture shows an enabled button (Install) and a disabled button (Next).

To ignore disabled child-windows (button, controls etc.) is the default behavior of the SCW. command. Therefore you will not need to specify the "e"-Flag often.
It can however make sense, to override the default behavior by using the "f"-flag which will make SCW. to ignore enabled child-windows and just look for disabled child-windows. This could be an button that is disabled, for example.

If you want to catch both, enabled and disabled child-windows, the "a"-flag is what you need. However the "a"->Flag will automatically also enable the search for invisible child-windows.

If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "e" - flag together with the "a" to search only child-windows that are enabled. No matter if they are visible or not.

For example. This will search only for enabled child-windows.

```
STW.c|WindowClass
SCW.ect|button|ok
MMV.
```

Since this is a default behavior of `SCW.`, we have wasted the "e" at this place. Its not needed. Now we would like to search for child-windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
STW.c|WindowClass
SCW.efct|button|ok
MMV.
```

If we give all two prototypes, the robot understands that we want enabled and disabled child-windows. But there is another way to do it. We can do this:

```
STW.c|WindowClass
SCW.aect|button|ok
MMV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

The "a" Flag says "search all child-windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled child-windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want enabled child-windows.".

Short Examples:

-

Example

```
'*********************************************
' SCW-Sample
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

You can enable buttons if they are disabled. You do this, using the `SWS.e` -command. This way you can click buttons, you would normally not be able to click.

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "f" - Flag - SCW. 3774

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.1("f" - Flag - SCW.

MiniRobotLanguage (MRL)

"f"-Flag "disabled"

Search Child Window

Intention

The picture shows an enabled button (Install) and a disabled button (Next).

To ignore disabled child-windows (button, controls etc.) is the default behavior of the SCW. command. While the robot can work with disabled controls as well as with enabled controls.

What can you do with disabled controls (Buttons, etc.)?
You can do nearly everything with them, that you can do with enabled controls. But the most important thing is, that you can enable them, using the `SWS.e` - command.

It can however make sense, to override the default behavior by using the "f"-flag which will make SCW. to ignore enabled child-windows and just look for disabled child-windows. This could be an button that is disabled, for example.

If you want to catch both, enabled and disabled child-windows, the "a"-flag is what you need. However the "a"->Flag will automatically also enable the search for invisible child-windows.

If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "f" - flag together with the "a" to search only child-windows that are disabled. No matter if they are visible or not.

For example. This will search only for disabled child-windows.

```
STW.c|WindowClass
SCW.fct|button|ok
MMV.
```

Now we would like to search for child-windows which are enabled, as well for disabled. We can do that in two ways.The first way is like this:

```
STW.c|WindowClass
SCW.efct|button|ok
MMV.
```

If we give all two prototypes, the robot understands that we want enabled and disabled child-windows. But there is another way to do it. We can do this:

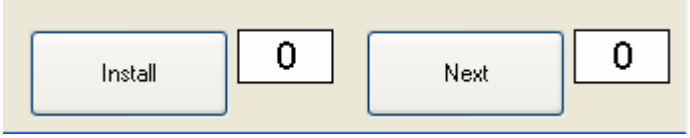
© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.c|WindowClass
SCW.afct|button|ok
MMV.
```

The "a" Flag says "search all child-windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the enabled child-windows. We do this by adding the "f"-Flag after the "a". In this case the "f"-Flag will override the "a"-Flag. And tell the robot "we just want disabled child-windows.".

Short Examples:

```
' You can try yourself to enable an disabled button
' Use this script and the "BCS-Testform"
' from the Sample Application
' use this example script
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nctf|1|Button|next
SWS.e
```



Example

```
'*********************************************
' SCW-Sample
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

You can disable buttons, immediately after they appear using the `SWS.d` -command.
This way you can prevent the user from clicking buttons, he should not click.

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "e" - Flag - SCW. 3771

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.17 "g" - Flag - SCW.

MiniRobotLanguage (MRL)

"g"-Flag "Goto on found"

Search Child Window

Intention

If the child-window (button/control .. etc.) you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to jump to a Label in that case.

If you also use the "X"-Flag to tell the robot to just continue with the script in case of an error - if the window is not found, then you can just reverse the normal behavior of the SCW.-command..

Sometimes this can be a perfect solution - depending on the process flow, that you want to implement.

Note that the 3.1 Systemvariables 334 #tio# and #til# will always flag if the child-window was found or not. This does not depend on the usage of the "o" or "g" or "X"-Flags. Therefore you can later still find out what happened.

Here is an example

```
SCW.Xtg|OK|found
MBX. The OK-Button was not found!
END.
:found
MBX.The OK-Button was found.
ENR.
```

In case you use the "g"-Flag, together with the "W" -Flag ("on Found Gosub"), the "W"-Flag will be ignored. The "g"-Flag has a generally higher priority.

Short Examples:

```
SCW.Xtg|OK|found
MBX. The OK-Button was not found!
END.
:found
MBX.The OK-Button was found.
ENR.
```

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'*********************************************
' SCW-Sample
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.tog|button 1|ero|found
' Script will nver get here!
END.

:found
GCT.
SCW.to|button 1|ero
SWS.vd

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

If you specify the "g"-Flag "on found Goto" together with the "W"-Flag (on found GoSub), then the "g"-Flag takes priority and the "W"-Flag will be ignored.

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "o" - Flag - SCW. 3791
- "w" - Flag - SCW. 3809
- "V" - Flag - SCW. 3754

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.18 "h" - Flag - SCW.

"h" - Flag "handle" Previous 3777 Top 41 Next 3782

MiniRobotLanguage (MRL)

"h"-Flag "handle"

Search Child Window

Intention

This flag is an exception. In fact it must not search the window, it will just localize the window "by handle". Of course this is a very efficient solution. Therefore if you have already localized your child-window before, keep the handle. You can keep the handle for example in a variable. Or on the stack.And then just feed it back to SCW. when you need that window again, using the "h"-flag.
As the "h"-flag does not search anything but immediately localizes the window, you should not combine it with other search flags, like "p","c","t" or these. It will just take priority and ignore these. You can combine the "h"-flag with the flags like "o","g",V" and "W". For example:

```
SCW.hg|123|notfound
```

Is completely Ok.

When using the "h"-flag with a valid child-window handle, there is no need to first use STW. to get a Top-Window. You can immediately locate a child-window. If you then need the Top-Window from that child window, you can use the GCT.-command.

```
' Localize by handle
STW.t|TestApp
' We grab the handle in a variable
SCW.t|button 1
HTV.$$MYC

' Later we can always get definitely this window,
' without further parameters
' No need to first localize a Top-Window!
SCW.h|$$MYC
MMV.
'
' Work with another window
STW.h|52367
MMV.

' jump immediately back to this child-window without
' the need to localize a Top-window first.
SCW.h|$$MYC
MMV.

ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Short Examples:

```
STW.ct|PBWindowClass:0|BCS - Testform
SCW.nct|1|Button|Install
HTV.$$MYC

SCW.t|invisi
MMV.

SCW.hg|123|notfound
ENR.

:notfound
MBX.Notfound!
ENR.
```

Example

```
'*********************************************
' SCW-Sample
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
' Keep the child-window handle in $$MYC
: $$MYC=#hwn#

' Get the Top-Window
GCT.
' Press another button
SCW.to|button 2|ero
MLE.

' Immediately locate the button via "h" again and press it.
SCW.h|$$MYC|ero
MLE.

ENR.

:ero
MBX.Window not found!
ENR.
```

Remarks

Locating child-windows using the "H"-Flag is not a windows-search, but a direct localization.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.19 "i" - Flag - SCW.

MiniRobotLanguage (MRL)

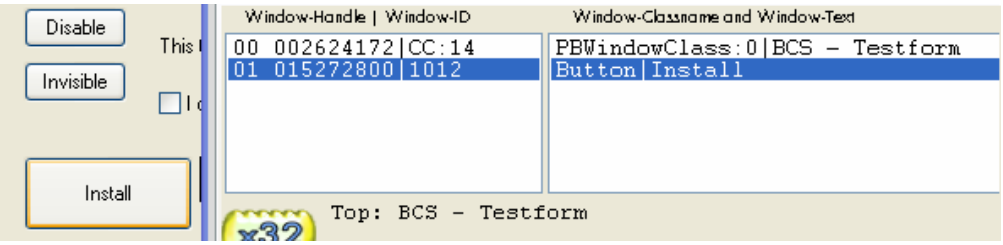
"i"-Flag "child-window ID"

Search Child Window

Intention

The child-window ID is traditionally a internal number, that the programmer assigns a window control at design-time. Therefore it can be used to locate a window in many cases. But be careful, not all child-windows ID's are really consistent.

You can see the window ID in the windows-Inspector.



In this case the button "Install" has a ID of 1012. Such an ID looks good and the chances are good that you can use it for the localization of the button. Therefore it must be permanent, means, that if you start the program many times, the ID must be always the same.

This is not always the case. Watch out if:

- the child-window ID is higher then 32000
- the child-window ID is zero or -1
- the child-window has a class name that starts with "t" like "tnewcheckmark" or "tnewlistbox".

In all these cases, the chances are that the id is not the same between multiple starts of the application, and therefore unusable for a safe window localisation.

How do you localize a window using the ID?

```
' First get the Top-Window:
STW.t|testap
' then get the child-window
SCW.i|1012
' now you can click it!
MLC.
```

Normally there should not be multiple child-windows with the same ID in the same dialog. Therefore if you have a valid child-window ID, you do not need to specify a "c"-class or "t"-text to locate the window. The ID is enough.
Unless you want. A "t"- with text can improve readability of your script.

© 2023 Theo Gottwald * https://www.smart-package.com

While it should not be needed, the "i" is a normal search flag and can be combined with all other search flags .

Short Examples:

-

Example

```
'*********************************************
' SCW-Sample
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

Normally there should not be multiple child-windows with the same ID in the same dialog. Therefore if you have a valid child-window ID, you do not need to specify a "c"-class or "t"-text to locate the window. The ID is enough.

While it should not be needed, the "i" is a normal search flag and can be combined with all other search flags, including "c" or "t".

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2("j" - Flag - SCW.

MiniRobotLanguage (MRL)

"j"-Flag "WII. by number"

Search Child Window

Intention

"Wait for Input Idle" halts script execution until the localized child-window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target child-window (-process) is busy. The maximum waiting time must be given as a numeric value or variable.

If you have specified the `WII. command or the` "j"-Flag in the STW.-statement, then there is no need to specify it again in the SCW., as internally the WII. works based on the process behind the window. And the process is the same for the Top- and the child-window.

Note the value you specify will replace the default internal value for "WII. without parameter", used for the "k" - Flag - STW. 3865 . But also for WII. if no parameters are given.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions are allowed. For example:

```
' WII. wait no longer then a half second
STW.c|WindowClass
SCW.tj|OK|0.5
MMV.

' Will do the same as "k" will repeat
' the value from "j" internally.
STW.c|WindowClass
SCW.tk|OK
MMV.
```

Technically the "j"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement. Just that it is integrated and does not need a separate line of code.

Short Examples:

© 2023 Theo Gottwald * https://www.smart-package.com

_

Example

_

Remarks

-

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341
- "k" - Flag - SCW. 3787

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2."k" - Flag - SCW.

MiniRobotLanguage (MRL)

"k"-Flag "WII. default"

Search Child Window

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time is taken from an internal value that is been set with the last WII. statement or with the "j" - Flag - STW. 3863.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions are allowed. For example:

```
' WII. wait no longer then a half second
STW.c|WindowClass
SCW.tj|OK|0.5
MMV.

' Will do the same as "k" will repeat
' the value from "j" internally.
STW.c|WindowClass
SCW.tk|OK
MMV.
```

Technically the "k"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement without a parameter. Just that it is integrated and does not need a separate line of code.

Short Examples:

-

Example

© 2023 Theo Gottwald * https://www.smart-package.com

_

Remarks

-

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 3.2 Standard-Search Pattern 341

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "n" - Flag - SCW.

MiniRobotLanguage (MRL)

"n"-Flag "Count on level"

Search Child Window

Intention

The "n"-flag needs a numeric parameter. This parameter specifies a number of a window that is exactly one level below the window that is the actual window. No matter if its a child- or a top-window.

There may be multiple results in a window-search with `SCW.`. Normally you get the first result that fits to your search parameters.

Using the "n"-Flag you can define which result you want, Nr.1, Nr.2 or Nr. 20. Just add the "n"-Flag and the result number that you are looking for.

In child-window search, the "r"-flag searches all child-windows below the Top-window.
That differs from the "n"-flag, which only looks at the child-windows that are directly below the actual located Top- or child-window.

If "r" (and "n") is omitted, always result Nr.1 from the rest of search parameters is returned.

Note: If there are not enough results, your search will fail!
Therefore you can also use this feature to count child-windows.

```
STW.c|WindowClass
' give me the third button on the next level
SCW.cn|button|3
MMV.
```

Short Examples:

-

Example

-

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "r" - Flag - SCW. 3797
- "S" - Flag - SCW. 3750

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "o" - Flag - SCW.

"o" - Flag "on not found Goto" Previous 3789 Top 41 Next 3793

MiniRobotLanguage (MRL)

"o"-Flag "on not found Goto"

Search Child Window

Intention

If the child-window (button, control etc.) is not found within the specified waiting time, normally you will end up with an error-message and the script will stop.

Maybe you want something else.
Using the "o"-Flag you can avoid the error and instead jump to another place in the script.
Using the "o"-Flag, the script just continues in that case. The window is not found and we tell the robot to jump to a Label in that case.

Take this example:

```
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
STW.t|BCS
SCW.toXw|Next|GoOn|0
GTO.weit

:GoOn
 MBS: The Next Button was not found!

:weit

MBX.Ready
ENR.
```

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Short Examples:

```
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
STW.t|BCS
SCW.toXw|Next|GoOn|0
GTO.weit

:GoOn
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 MBS: The Next Button was not found!

:weit

MBX.Ready
ENR.
```

Example

-

Remarks

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "g" - Flag - SCW. 3777
- "w" - Flag - SCW. 3809
- "V" - Flag - SCW. 3754

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "p" - Flag - SCW.

MiniRobotLanguage (MRL)

"p" - Flag "Process ID"

Search Child Window

Intention

The "p"-flag ist mostly used with Top-Windows. Within a Top-window all child-windows have the same PID. You can however use it in very special cases for child-windows also.

Especially if you use the "T"-Flag, which does search through all windows (desktopwide).

You can localize a window safely, if you have the PID (Process ID) of the application behind that window.

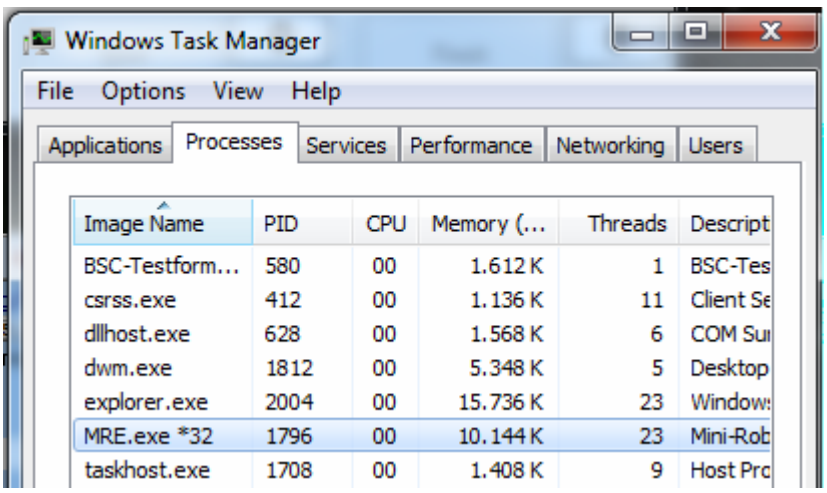
Be sure that you have read and understood the chapter 1.7. Application and PID 137.

Hint: If you have localized a window, you also have the PID of the process behind that window in `#pid#`. You can save it for later use in a variable. Like this:

```
STW.c|pbwind
: §§PID=#pid#

or use:

STW.c|pbwind
HTV.§§PID
```

In both cases, you get the PID of that process into the local variable §§PID 366.

You can see the PID of a process in the task-manager::



 © 2023 Theo Gottwald * https://www.smart-package.com

The PID value is a normal search parameter and can be used alone or together with other search parameters.

```
' Search a window from process with ID 3435
' and the given window class.
'
STW.ch|pbwind|580
MMV.
```

Short Examples:

```
' Locate window only by PID
STW.p|580
MMV.

ENR.

SCW.btc|250|Cancel|Button
```

Example

-

Remarks

-

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "T" - Flag - SCW. 3750

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "q" - Flag - SCW.

MiniRobotLanguage (MRL)

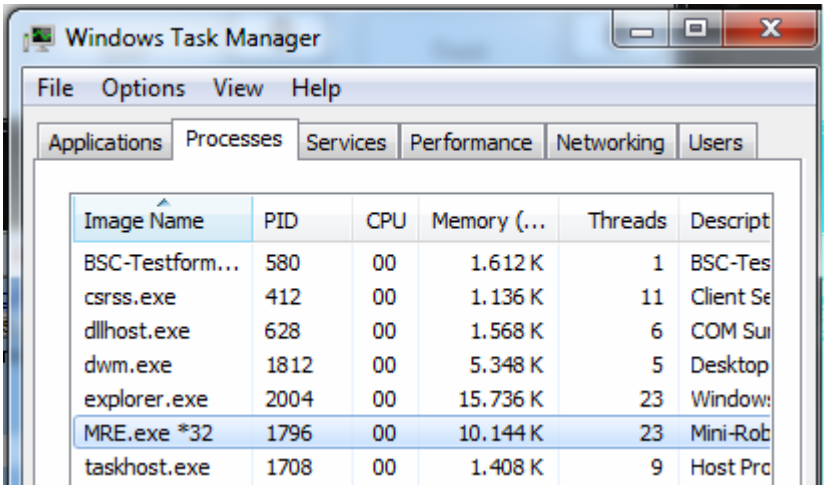
"q" - Flag "Thread ID"

Search Child Window

Intention

Top-Windows belong to a single Thread therefore this parameter is only for very special cases in SCW.

This is an advanced parameter. The chances are, that you will never need it. Processes have a number of Threads. See this picture here:



On the left side you can see the name of the process. Right of it the PID of the process. Later you see "Threads".
In a very simplified form, a Thread is a subprogram of that process, that runs an own small program. Generally the more threads a program has, the better it will scale up on multiple CPU-Core's.
While all Threads of a process belong to that process and therefore have the same PID, they have a different Thread-ID. In case you may need to separate between windows of the same process, which is "Multi-threaded", the "q"-flag is for you.

```
STW.cq|WindowClass|109872
MMV.
```

Hint: If you have localized a window, you also have the TID of the process behind that window in `#tid#`. You can save it for later use in a variable. Like this:

```
STW.c|pbwind
: §§TID=#tid#
```

This way, you get the TID of that Thread into the local variable §§PID 366.

© 2023 Theo Gottwald * https://www.smart-package.com

If you don't "understand a word", or don't have very special cases, just ignore this flag.

Short Examples:

_

Example

_

Remarks

-

Limitations:

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "p" - Flag - SCW. 3793

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2("r" - Flag - SCW.

MiniRobotLanguage (MRL)

"r"-Flag "result number"

Search Child Window

Intention

There may be multiple results in a window-search with `SCW.`. Normally you get the first result that fits to your search parameters.

Using the "r"-Flag you can define which result you want, Nr.1, Nr.2 or Nr. 20. Just add the "r"-Flag and the result number.

In child-window search, the "r"-flag searches all child-windows below the Top-window.

That differs from the "n"-flag, which only looks at the child-windows that are directly below the actual located Top- or child-window.

If "r" (and "n") is omitted, always result Nr.1 is returned.

Note: If there are not enough results, your search will fail!
Therefore you can also use this feature to count child-windows.

```
STW.c|WindowClass
' give me the third button
SCW.cr|button|3
MMV.
```

Short Examples:

-

Example

-

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "n" - Flag - SCW. 3789
- "S" - Flag - SCW. 3750

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.27 "s" - Flag - SCW.

"s" - Flag "reverse direction" Previous 3797 Top 41 Next 3801

MiniRobotLanguage (MRL)

"s"-Flag "reverse direction"

Search Child Window

Intention

The "s"-Flag will reverse the internal search direction of child-windows. Example:
If there are four child-windows on a level. You would specify these like that:

```
SCW.ctn|Button|Cancel|1
...
SCW.ctn|Button|Cancel|4
```

```
now after adding the "s"-Flag, the child-window Nr.1 will be
Nr.4, the Ne.3 will be Nr.2. They will just completely reverse
their search order.
This can be handy if you have a large number of child windows
and you want to "grab them from the other end".

This way you may be less dependent on changes within those
other child-windows.

Also this flag is needed for compatibility with some
WinRobots-Scripts, as WinRobots does use a reverse search
direction sometimes.
```

Short Examples:

-

Example

-

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "t" - Flag - SCW.

"t" - Flag "text" Previous 3799 Top 41 Next 3804

MiniRobotLanguage (MRL)

"t" - Flag "text"

Search Child Window

Intention

This flag is one of the most important search parameters. It is the window-text. Most often you can even see the text of a top-window without using the windows-Inspector.
Here is the simplest case how you can grab a button or child-window.

```
SCW.t|Yes
MMV.
```

The text parameter is a string-value and therefore supports 3.2 Standard-Search Pattern 341.
Let us try some of these.

As always with Search patterns, it is enough to give a unique part of the value. By default the text is not case sensitive. You can change that behavior using the search patterns. Take the sample-script below and replace the line:

```
' Locate the window
SCW.t|next
```

with the following lines. This will help you to understand the use of Patterns for text search.

```
' Lets start with this
SCW.t|{&DOCAPS:Next}
```

This will force the text to be case-sensitive! Try this:

```
' This will not find the window
' because of case-sensitivity!
STW.t|{&DOCAPS:next}
```

It does not find the button because the &DOCAPS: switches case-sensitivity on.
Now how about this:

```
SCW.nct|2|Button|{&EXACT:&next}
```

It does find the button. But then whats the "&EXACT:" good for?
Try this:

```
STW.t|{&EXACT:ext}
```


© 2023 Theo Gottwald * https://www.smart-package.com

It does not find the window because the &EXACT: switches off the default behavior that a part of the text is enough. Can it be combined with the `&DOCAPS:` ?

We'll try this with Top-Windows here, because its less to type. The patterns work in the same way with child-windows.

We try. This should work:
```
STW.t|{&EXACT:&DOCAPS:BCS - Testform}
```
and this should not work:
```
STW.t|{&EXACT:&DOCAPS:bCS - Testform}
```

Yes, everything work as expected.

Do you know the old times, when you had simple Wildcards as patterns? The"*" and the "?" ?

You can have them back if you want. Try this:

```
STW.t|{&WILDC:b?? - Testform}
```

Yes, it works. And how about:

```
STW.t|{&WILDC:b?? - Tes*orm}
```
or
```
STW.t|{&WILDC:b?? - *s*or?}
```

It works. We kept them for you.

You can still use even the old DOS-Patterns if you really want.
But there is much more if you use the new patterns.

How about a fuzzylogic and a OR'ed combination of two patterns?

```
STW.t|{&WILDC:b?? }&OR:{BCS-tesform&FUZZY:6}
```

and you can not only combine two patterns, you can combine as many patterns using &OR: as you like! If you want to support many languages, this is the way to go.

No matter if there is a "YES" on that button or a "Qui" or whatever. So long you know it, or at least a bit of it, you can locate it using Search patterns.

Short Examples:

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe

' Locate the window
STW.t|Testf

' Close the window
CAW.PBWindowClass:0|BCS - Testform
```


© 2023 Theo Gottwald * https://www.smart-package.com

ENR.

Example

-

Remarks

-

Limitations:

See also:

-
- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- "n" - Flag - SCW. 3789
- "r" - Flag - SCW. 3797
- "S" - Flag - SCW. 3750

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.2 "u" - Flag - SCW.

MiniRobotLanguage (MRL)

"u" - Flag "invisible"

Search Child Window

Intention

To ignore invisible windows is one of the default settings of the SCW. command. Therefore if you want to search for a child-window that has is invisible, you will need the "u"-flag.

The "u"-Flag will override the default behavior. It will make SCW. to ignore visible windows and look for invisible windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "u" - flag together with the "a" to search only windows that are invisible. No matter if they are enabled or not.

Lets take a look, on this. We'll show it with Top-Windows (STW.) here, but the system is completely equal for SCW. and ICW. This will search only for invisible windows.

```
STW.cu|WindowClass
MMV.
```

Now we would like to search for windows which are visible, as well as for invisible windows. We can do that in two ways.The first way is like this:

```
STW.cuv|WindowClass
MMV.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cae|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Short Examples:

-

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*******************************************
' SCW-Sample
'*******************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

-

Limitations:

-

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- 1.4. The Prototype Parameter System 73
- "V" - Flag - SCW. 3806

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.3("v" - Flag - SCW.

"V" - Flag "GoSub if not found. " Previous 3804 Top 41 Next 3809

MiniRobotLanguage (MRL)

"v" - Flag "visible"

Search Child Window

Intention

To ignore invisible windows is one of the default settings of the SCW. command. Therefore you will not need to specify the "v"-Flag often.

It can however make sense, to override the default behavior by using the "u"-flag which will make SCW. to visible windows and just look for invisible windows.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also switch on the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "v" - flag together with the "a" to search only windows that are visible. No matter if they are enabled or not.

Lets take a look, on how to do this. In this case we will look at it with `STW.`, but its equal for `SCW.` and `ICW.` This will search only for visible windows.

```
STW.cv|WindowClass
MMV.
```

Since this is a default behavior of `SCW.`, we have wasted the "v" at this place. Its not needed.

Now we would like to search for windows which are visible, as well for invisible windows. We can do that in two ways.The first way is like this:

```
STW.cuv|WindowClass
MMV.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cae|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Short Examples:

-

Example

```
'*********************************************
' SCW-Sample
'*********************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

-

Limitations:

See also:

- 1. STW. - Decription 3814

© 2023 Theo Gottwald * https://www.smart-package.com

- SCW - Search-Child-Window 3738
- 1.4. The Prototype Parameter System 73
- "u" - Flag - SCW. 3804

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.24.3."w" - Flag - SCW.

MiniRobotLanguage (MRL)

"w" - Flag "wait time"

Search Child Window

Intention

How long do we wait if a window will appear or not?
Let's try it.

```
: $$TIM=#dtime#
STW.tX|nix
: $$TIM=#dsince#
MBX.$$TIM
ENR.
```

This i what we get.

It's currently about 15 seconds. When you read this it can be another value.
In fact this is the default time that we wait for a window before we run into an error-message or do something else (like in this script).

Now, its very important to understand that we do not generally wait 15 seconds, and then look if the window has appeared.
What we do is, we try ... try ... and try and check permanently, if the window has appeared. If the window appears any time, the script will immediately continue.

If the time is over, we do something else.

Now 15 seconds is enough for some cases, in many cases the window will appear after 45 seconds or any other time. Now this is where the "w"-flag enters the game.

Using the "w"-flag we can tell the robot how long he should wait. It can be 0 seconds like this:

```
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
```

or 360 seconds like here.

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.wtu|360|BCS - Testform
```

and there is no real limitation.

Short Examples:

-

Example

```
'*******************************************
' SCW-Sample
'*******************************************
' Start TestApp only, if it's is not yet running.
'
STW.wtUXg|0|TestApp|found
EXE.?exepfad\Sample Apps\TestApp_01.exe
:found

STW.t|testapp

SCW.to|button 1|ero
SWS.i

GCT.
SCW.tuo|button 1|ero
SWS.vd

GCT.
SCW.tfo|button 1|ero
SWS.id

GCT.
SCW.tufo|button 1|ero

SWS.

ENR.
:ero
MBX.Window not found!
ENR.
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

- 1. STW. - Decription 3814
- SCW - Search-Child-Window 3738
- "g" - Flag - SCW. 3777
- "o" - Flag - SCW. 3791
- "V" - Flag - SCW. 3754

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.25 SFW. - Set Foreground WIndow

MiniRobotLanguage (MRL)

SFW. Command

Set Foreground window

Intention

The `SFW.`.-Command will bring the last localize Application-window (Top-Window) in front of all other non-TopMost windows.

Usage is simple:

```
' Normally STW. already includes a SFW.
' We have disabled this here, using the "A"-Flag
STW.Act|PBWindowClass:0|TestApp
SFW.
SCW.cn|edit|2
```

`STW.` already contains an implicit `SFW.` if the "A"-Flag is not given therefore you will most often not need to explicitly call this command.

Important note: Windows that are **TopMost** will always stay in front of all non TopMost windows. An example for a TopMost windows is the Script-Editor, if the Menu Entry "Display always on Top" is checked.

To make a Window "TopMost", use "M" - Flag - STW. 3838

Syntax

SFW.[P1]

Parameter Explanation

P1 - ***(optional)*** `VAR or formula in (..).`
`Must be a valid Window handle`

Example

```
STW.AgXctw|over|PBWindowClass:0|TestApp|0
EXE.?exepath\Sample Apps\TestApp_01.exe
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
:over

' Normally STW. already includes a SFW.
' We have disabled this here, using the "A"-Flag
STW.Act|PBWindowClass:0|TestApp
SCW.cn|edit|2
SFW.
ENR.
```

Remarks

Will set the timeout-flag if the operation was not successful.

Limitations:

-

See also:

- SAW. - Set Active Window 3736
- SWS. - Set Window State 3889
- ! STW. -Search Top-Window 3814
- ! SCW - Search-Child-Window 3738
- WII. - Wait-for-Input-Idle 2909
- SKP. Send-Key-Press 2289
- SKT. - Send-Key-Text 2293

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26 STW. - Search Top Window

3.83.26.1 ! STW. -Search Top-Window

MiniRobotLanguage (MRL)

STW. Command

Search Top Window

Intention

This command localizes a specified Top-Window and sets the actual point into its center.
All following SCW-commands will be within this window.
STW. has a default window-treatment 3814 to ensure that any further action in that window will be effective. This default treatment will do following:
- normalize minimized windows (like the "D"-Flag
- windows that are outside the desktop-work area will be moved to 0,0 (like "G"-Flag)
- The window will be moved in front of all other (non-TopMost 3838)-windows.
- The window will receive the input focus. (like the "J"-Flag)

This default window-treatment can be completely deactivated using the "A"-Flag 3819.

This makes sense, for example when working with Menu's or other windows which disappear when the default treatment is being applied.

Alternatively the default window treatment can be influenced using other flags, like "B" or "G".

If the window should be brought into work area, without other treatment, use the "B"-Flag. "G", will just prevent the repositioning of the window and let the other treatment-steps happen.

All window search operations change the Timeout System Variables `#tio#` and `#til#`. Using these you can find out if a window was found or not, even if you have used the"X" Flag (X - run without Error).

Hint: Please note that the Flags have a priority from left to right (highest Priority). If you add other flags after the "A" there may be a treatment. Because the more right the flags, the higher their priority. If you place the "A" last inside the prototypes, all involved treatment will be disabled.

See two examples.
In this first case the "A" Flag has priority over the "C" Flag, because its right of it.

```
STW.tCA|explorer
MMV.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
STW.tAC|explorer
MMV.
```

Some Flags deactivate the default treatment or parts of it. See description below.

Hint: If you omit all parameters like this:
`STW.`
the command will automatically restore the parameters from the previous call. This makes it easy to always locate the same Top-Window.Note that this also applies to `ITW.` as both use the same parameters-store.

Syntax

STW.P1[|P2]...[|Px]

Parameter Explanation

P1 – Prototype
P2 .. PX - Parameters according to prototypes

```
A - (np) Switch off default window treatment completely
B - (np) disables following parts of "default treatment":
     NO Restore from Minimized, NO Set Input-Focus,
     NO Bring to Front
C - (np) If the window i maximized, it will be restored to
    normal size
D - (np) If the window is minimized,  will be restored to its
    previous size, before it was minimized,
E - (np) Put Localize window, then add window to
    Search Exclude List.
F - (np) Move window to center of primary desktop
G - (np) disable default window-treatment-"repositioning if
     outside"
H - (np) a window that is outside the primary desktop working
area
    will be repositioned. Default is that the window can be
also
    on a second desktop.
I - (NUM/VAR) Remove window with the given window-handle from
the
    search-exclude-list.
J - (np) Bring window to Front. Does not switch off default
window
    treatment.
K - Immediately clear Exclude-List before specified search
M - (np) Set window to TOPMOST. Such windows will be on Top
of all
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
    other Non-TOPMOST windows. DANGER - even Messageboxes may
be
    hidden behind a TOPMOST window. Use this with caution!
N - (np) remove the TOPMOST Property from a window.
Q - use previous Prototypes (only prototypes)
V - (p/Label) Gosub Line/Label if the window was NOT found
W - (p/Label) Gosub Line/Label if the window was found
X - (np) run without error if window was not found. Timeout
Flag
    #tio# and #til# will anyway be set and can be used.

a - (np) changes internal settings that "all" windows will be
    found. This includes invisible and disabled windows.
b - (np) "back" - moves window to the back, behind other
windows.
c - (p) "class", window-class parameter
e - (np) enabled, the window must be enabled or it will
ignored.
f - (np) disabled, the window must be disabled or it will
ignored.
g - (p/Label) Goto Line/Label if the window was found
h - (p) localize the window directly using a given window
handle.
    Note, if "h" is given, other search conditions will
    be ignored.
j - (p) WaitForInputIdle, you can specify the WII timeout
k - (np) WaitForInputIdle - no parameter, use the default
    WII timeout.
o - (Label/Variable) on not found goto (Label)
p - (p) process-ID, search only windows from process with
this
    given PID
q - (p) ThreadID,, search only windows from process with this
    given TID
r - (p) No. of result to choose (Result-Number for Top-
Windows)
t - (p) text, window-text (often the window caption)
u - (np) invisible, search only invisible windows
v - (np) visible, search only visible windows
w - (p) total waiting-time in Seconds, 0.5 (FP-numbers) is
    possible.
```

Explanation:

(np) -> only prototype, no additional parameter required

(p) -> prototype requires additional parameter

(NUM/VAR) - window handle/variable

(p/Label) - Label-Parameter required, can also be a variable that contains the label

P2 - Px - Data according to the prototypes, separated by |

Example

```
'MR Script-file:
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'Version:
'Purpose: STW.-Features Demo
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' g ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
'
STW.wtgX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Now we can use the Application and press Buttons
STW.wt|0|BCS - Testform
SCW.nctb|1|Button|Next|0
' Compatible Button Click
MLC.

'------------------------------------------------------------
ENR.
```

Remarks

To select/locate a Top- Window, you will most often use `STW.` or in few cases `LAP.`

Limitations:

-

See also:

- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- 1.5. Features and Hints 150
- 1.9. Locating Child-Windows 162

© 2023 Theo Gottwald * https://www.smart-package.com

- 1.7. Application and PID 137
- 3.1 Systemvariables 334
- LAP. - Locate at Point 599

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2 "A" - Flag - STW.

"A" - Flag "No windows treatment" Previous 3814 Top 41 Next 3821

MiniRobotLanguage (MRL)

"A" - Flag "No windows treatment"

Search Top Window

Intention

STW. has a default window-treatment to ensure that any further action in that window will be effective. This default treatment will do following:
- normalize minimized windows (like the "D"-Flag
- windows that are outside the desktop-workarea will be moved to 0,0 (like "G"-Flag)
- The window will be moved in front of all other (non-TopMost)-windows.
- The window will receive the input focus. (like the "J"-Flag)

This default window-treatment can be completely deactivated using the "A"-Flag.

This makes sense, for example when working with Menu's or volatile windows, which disappear when the default treatment is being applied.

Usage:

```
STW.cA|#32768
MMV.
```

Please note that the Flags are internally analyzed in the order in which they are given. If you add other flags after the "A" there may be a treatment.
If you place the "A" last inside the prototypes, all such treatment will be disabled.

In this case the "A" Flag has priority over the "C" Flag, because its right of it.

```
STW.tCA|explorer
MMV.
```

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
STW.tAC|explorer
MMV.
```

Example

```
'MR Script-file:
'Version:
'Purpose: STW.-Features Demo
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
STW.cA|#32767
MMV.
ENR.
```

Remarks

Alternatively the default window treatment can be influenced using other flags, like "B" or "G".

If the window should be brought into work area, without other treatment, use the "B"-Flag. "G", will just prevent the repositioning of the window and let the other treatment-steps happen.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.3 "B" - Flag - STW.

"B" - Flag "partial treatment" Previous 3819 Top 41 Next 3823

MiniRobotLanguage (MRL)

"B" - Flag "partial treatment"

Search Top Window

Intention

STW. has a default window-treatment to ensure that any further action in that window will be effective. This default treatment will be reduced, when the "B" - Flag is set.

When the "B" Flag is in effect, windows that are outside the desktop-workarea will be moved to the upper, left corner of the desktop - that is position 0,0. Thats the only treatment that will be done.

Also this can be disabled with the "G"-Flag. Means, that the flags "G" and "B" together are just like the "A" flag,

Example:

```
STW.Bc|WindowClass
MMV.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
STW.cB|#32767
' Locate the button
SCW.t|Ok
' Click the button
MLE.
ENR.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.26.4 "C" - Flag - STW.

MiniRobotLanguage (MRL)

"C" - Flag "restore if maximized"

Search Top Window

Intention

If the window is maximized, this flag will bring the window back to its normal size.
You can maximize, minimize and restore windows using the SWS. - Set Window State 3889 command.

It doesn't matter, if the window was minimzed before it was maximized.
The window will be normal size after this flag.

Example

```
STW.ct|PBWindowClass:0|TestApp
' Maximize Window
SWS.z
' Wait 1 Second
PAU.1
' Window will be restore to normal size
STW.ACct|PBWindowClass:0|TestApp
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.5 "D" - Flag - STW.

"D" - Flag "restore if minimzed" Previous 3823 Top 41 Next 3825

MiniRobotLanguage (MRL)

"D" - Flag "restore if minimized"

Search Top Window

Intention

If the window is minimized, this flag will bring the window back to its normal size.
You can maximize, minimize and restore windows using the SWS. - Set Window State 3889 command.

It does not matter, if the window was maximized before it was minimized.
The window will be normal size after this flag.

Example

```
STW.Act|PBWindowClass:0|TestApp
SWS.z
SWS.m
PAU.1
STW.ADct|PBWindowClass:0|TestApp
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.6 "E" - Flag - STW.

"E" - Exclude window" Previous 3824 Top 41 Next 3827

MiniRobotLanguage (MRL)

"E" - Flag "Exclude window"

Search Top Window

Intention

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you get the window out the "Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

Example

```
' Use Exclude List

' Add window to exclude list
STW.tE|TestAP
MMV.
VAR.$$HWN=#hwn#

' Remove window from exclude List
STW.I|$$HWN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' If you comment the STW. above the window can not be found
here:
STW.t|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List availabe. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "I" - Flag - STW. 3832
- "K" - Flag - STW. 3836
- "h" - Flag - STW. 3861

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.7 "F" - Flag - STW.

"F" - Flag "center window on desktop" Previous 3825 Top 41 Next 3829

MiniRobotLanguage (MRL)

"F" - Flag "Center on Desktop"

Move Window to center of primary desktop

Intention

If the "F"-Flag is given, the window will be move to the center of the primary desktop. The "F"-Flag will override the default windows treatment, which would move the window to 0,0 if its outside of the desktop.
The "F"-Flag will always move the window to the center, no matter where the window was before.

```
STW.Fc|WindowClass
MMV.
```

Example

```
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.cF|Myclass
' Move Mouse to Center of window
' in center of desktop
MMV.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.8 "G" - Flag - STW.

MiniRobotLanguage (MRL)

"G" - Flag "don't move window"

don't move the Top Window

Intention

This flag prevents the default window treatment from moving the window around. Without this Flag, a Top-Window that is outside the working area, will be move to position 0,0, that is the upper left corner of the desktop.

Having the "G"-Flag set, the window will just not be moved, no matter if its outside the working area or not.

```
STW.cG|WindowClass
MMV.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.cG|#32768

ENR.
```

Remarks

Please note that other prototypes may influence the power of the "G"-Flag, especially, if the are right of the "G"-Flag which means they have a higher priority.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.9 "H" - Flag - STW.

"H" - Flag "stay in primary desktop" Previous 3829 Top 41 Next 3832

MiniRobotLanguage (MRL)

"H" - Flag "primary desktop"

Search Top Window

Intention

This flag will move a window to position 0,0, that is the upper left corner, if the window is full or partial outside the primary desktop workarea.

It will override the default window treatment.
While by default also a Dual-Screen Desktop will be accepted as workarea, this will not be the case here.

The "H"-Flag makes sure the window is complete on the primary desktop.

```
STW.cH|WindowClass
MMV.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
STW.tH|word
MMV.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1("I" - Flag - STW.

MiniRobotLanguage (MRL)

"I" - Flag "remove from Exclude List"

Search Top Window

Intention

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you get the window out the "Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

There is another way, how to search windows which are currently excluded from search?
You can also use the "K"-Flag.

```
' clear whole exclude List
STW.Kc|WindowClass
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list.

After this all previously "exclude listed" windows can be found again.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
' Use Exclude List
' Add window to exclude list
STW.tE|TestApp
VAR.$$HWN=#hwn#

' Remove window from exclude List
STW.I|$$HWN

' If you comment the STW. above the window can not be found
here:
STW.t|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag. To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List availabe. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "E" - Flag - STW. 3825
- "K" - Flag - STW. 3836

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1."J" - Flag - STW.

"J" - Flag "window to front" Previous 3832 Top 41 Next 3836

MiniRobotLanguage (MRL)

"J" - Flag "window to front"

Search Top Window

Intention

The window comes to front already with the default window treatment, without any additional flags. But you can disable that, and select yourself exactly what you want. Like this:

```
STW.AJc|WindowClass
MMV.
```

The "A" Flag will switch of all default window treatment. Then comes the "J"-Flag and tells the robot to bring the window to front. Be careful with the operator priority. If you place the prototypes like this:

```
STW.JAc|WindowClass
MMV.
```

then the "A"-Flag will switch of the "J"-Flag because it is more right and therefore has a higher priority.

The "J"-Flag itself will not disable the default window treatment. Therefore something like:

```
STW.Jc|WindowClass
MMV.
```

```
is possibly a waste of letters, as the default behavior of
STW. is already to bring the window to front.

Note that the Flag "B" does not switch off the "J"-Flag,
therefore when using "B", the sequence of the prototypes "B"
and "J" should not be a issue.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.AJc|#32768
' Locate the button
MMV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX. Compare with this:
STW.Ac|#32768
' Locate the button
MMV.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1 "K" - Flag - STW.

"K" - Flag "Clear Exclude List and search" Previous 3834 Top 41 Next 3838

MiniRobotLanguage (MRL)

"K" - Flag "Clear Exclude List"

Clear Exclude List and search

Intention

Immediately clear Exclude-List before specified search.

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you search windows which are currently excluded from search?
This is what the "K"-Flag is good for.

```
' clear whole exclude List
STW.Kc|WindowClass
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list.

After this all previously "exclude listed" windows can be found again.

Example

```
' Use Exclude List

' Add window to exclude list
STW.Et|TestApp
MMV.
VAR.$$HWN=#hwn#

' Remove all windows from exclude List
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' and search

' If you remove the "K" below, the window can not be found!
STW.Kt|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List available. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Currently you can not check which windows are exclude-listed. There is also no way of showing the content of an exclude-list.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "I" - Flag - STW. 3832
- "E" - Flag - STW. 3825

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1 "M" - Flag - STW.

"M" - Flag "TopMost" Previous 3836 Top 41 Next 3840

MiniRobotLanguage (MRL)

"M" - Flag "TopMost"

Search Top Window

Intention

In windows, there is a property of Top-Windows that will make them stay always in front of all other windows.

This property is called "TopMost". A window that is "TopMost" is always on top of all other windows, that are not TopMost.

Only another TopMost window can be in front of an TopMost window.

For example the MR-Editor can be set to TopMost.

Now you may think "this sounds good!". Be warned. This is an advanced feature for advanced users.

Programs have Messageboxes and also open up other windows. In case they have been made TopMost, you may not be able to see those Messages or PopUp windows. This may lead for you to think that the program hangs.

Why don't you see those messages?
Because they will not be TopMost and therefore they will appear behind the TopMost window!

Therefore use TopMost style only when really needed and when you know what you are doing.

Using the Flag "M", the window will be made TopMost immediately after it was localized.

```
STW.cM|WindowClass
MMV.
```

Use the "N"-Flag to remove the TopMost-property from a TopMost window.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.Mc|#32768
VAR.$$HWN=#hwn#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Remove the TopMost Property
STW.hN|$$HWN

ENR.
```

Remarks

Limitations:

See also:

"N" - Flag - STW. 3840

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.14"N" - Flag - STW.

MiniRobotLanguage (MRL)

"N" - Flag "No TopMost"

No more TopMost

Intention

In windows, there is a property of Top-Windows that will make them stay always in front of all other windows.

This property is called "TopMost".

A window that is "TopMost" is always on top of all other windows, that are not TopMost.

Only another TopMost window can be in front of an TopMost window.

For example the MR-Editor can be set to TopMost.

Now you may think "this sounds good!". Be warned. This is an advanced feature for advanced users.

Programs have Messageboxes and also open up other windows. In case they have been made TopMost, you may not be able to see those Messages or PopUp windows. This may lead for you to think that the program hangs.

Why don't you see those messages?
Because they will not be TopMost and therefore they will appear behind the TopMost window!

Therefore use TopMost style only when really needed and when you know what you are doing.

Using the Flag "M", the window will be made TopMost immediately after it was localized.

```
STW.cM|WindowClass
MMV.
```

Use the "N"-Flag to remove the TopMost-property from a TopMost window.

Example

```
'#EXE:?pfad\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.Mc|#32768
VAR.$$HWN=#hwn#

' Remove the TopMost Property
STW.hN|$$HWN

ENR.
```

Remarks

Limitations:

See also:

"M" - Flag - STW. 3838

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1!"Q" - Flag - STW.

MiniRobotLanguage (MRL)

"Q" - Flag "Repeat Prototype"

Search Top Window

Intention

This flag will just repeat the Prototype that was last used with STW. or ITW. or any otehr Top-Window search instruction.

```
STW.c|WindowClass
MMV.
' this will do the same
STW.Q|WindowClass
MMV.
```

Note that there is another way of repeating the last STW. or ITW.. That is, to just leave the whole parameter part empty.

```
STW.c|WindowClass
MMV.
' this will do the same
STW.
MMV.
```

While "Q" will only retype the prototypes internally. Leaving the parameters away, will retype the whole parameters, including the prototypes, using those from last time the command was called.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
STW.c|WindowClass
MMV.
' this will do the same
STW.Q|WindowClass
MMV.
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

`STW.` and `ITW.` share the same parameter store. Therefore if you have localized the window with ITW. then you can also use these parameters for STW. and vice versa.

Limitations:

-

See also:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.26.1("V" - Flag - STW.

"V" - Flag "GoSub if not found. " Previous 3842 Top 41 Next 3846

MiniRobotLanguage (MRL)

"V" - Flag "GoSub if not found"

Gosub if not found

Intention

If the window you are looking for is not found within the specified waiting time, several things can happen. Normally you will get an error-message.

But that does not necessarily be what you want. Maybe you want just to see if the window is there and if not start the program in a Subroutine.

```
' "V" - Flag - on not found Gosub
STW.ctVw|PBWindowClass:0|BCS|Sub|1
' Move Mouse to the Install Button
SCW.ct|butt|Install
MMV.
ENR.
'-------------------------------------------------------------
' This Subprogram will be called if the window is not found
:Sub
EXE.?exepfad\Sample Apps\BSC-Testform.exe
STW.ct|PBWindowClass:0|BCS
RET.
```

The script above will try to locate the "BCS - Testform" and if its not yet started, it will be started and localized.

Example

```
STW.ctVw|PBWindowClass:0|TestAppx|Sub|1
MBX.Ready
ENR.
'-------------------------------------------------------------
:Sub
MBX.Error - Program not found
RET.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

You can not mix the "V" and the "W"-Flag. If both are found, the "W"-Flag takes priority

See also:

- "g" - Flag - STW. 3859
- "o" - Flag - STW. 3867
- "W" - Flag - STW. 3846
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1 "W" - Flag - STW.

"W" - Flag "GoSub if found" Previous 3844 Top 41 Next 3848

MiniRobotLanguage (MRL)

"W" - Flag "GoSub if found"

Search Top Window

Intention

If the window you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to first call a Subprogram in that case and then continue with the script. This is done using the "W"-Flag.

```
STW.ctW|PBWindowClass:0|BCS|Sub
MMV.
ENR.
:Sub
MBX.!
RET.
```

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Example

```
'
' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

' If found close that window.
STW.ctWw|PBWindowClass:0|BCS|Sub|1

' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

STW.ct|PBWindowClass:0|BCS
SCW.ct|butt|Install
MMV.

ENR.
'-------------------------------------------------------------
:Sub
CAW.PBWindowClass:0|BCS
' Wait until its gone
MTW.
RET.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Limitations:

-

See also:

- "g" - Flag - STW. 3859
- "o" - Flag - STW. 3867
- "V" - Flag - STW. 3844
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.18 "X" - Flag - STW.

"X" - Flag "run without error" Previous 3846 Top 41 Next 3850

MiniRobotLanguage (MRL)

"X" - Flag "run without error"

Search Top Window

Intention

If the window you are looking for is not found within the specified waiting time, several things can happen. Normally you will just get an error-message. If you do not want to get an error, but just have the script continue running, use the "X"-Flag.

You can use the `#tio#` 3.1 Systemvariables 334 to check if the window was found or not. Also you can use the "V","W","g" and "o" Flags to react inside the STW.

For example in such a script:

```
STW.cXg|WindowClass|found
MMV.

:found
MPO.
```

This script will invert the normal behavior of STW. It will continue running in case of the window is not found. And in case the window is found, it will jump to the Label "found".

Example

```
'-------------------------------------------------------
' "X"-Flag Sample 1
' Run only if its not yet running ...
'-------------------------------------------------------
'
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - gotoLabel "Sub" if thewindow was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.


'-------------------------------------------------------
' "X"-Flag Sample 2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' g ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
'
STW.wtgX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Now we can use the Application and press Buttons
STW.wt|0|BCS - Testform
SCW.nctb|1|Button|Next|0
' Compatible Button Click
MLC.

'-------------------------------------------------------------
ENR.
```

Remarks

Limitations:

See also:

- "g" - Flag - STW. 3859
- "o" - Flag - STW. 3867
- "V" - Flag - STW. 3844
- "W" - Flag - STW. 3846

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.1 "a" - Flag - STW.

"a" - Flag "all windows" Previous 3848 Top 41 Next 3852

MiniRobotLanguage (MRL)

"a" - Flag "all windows"

Search Top Window

Intention

In most cases, we want to act on visible and enabled window only.
Therefore the STW. command ignores by default windows, if they are disabled or invisible.

The "a" flag changes this behavior. Using the "a"-Flag, the STW. command will locate "all windows". This includes Top-Window which are invisible or disabled, or both.
Using the Flag "a" ALL Top windows will be found if they meet the search criteria.

Example

```
'MR Script-file:
'Version:
'Purpose: STW.-Features Demo
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Next we make the thing invisible
SWS.i
' Wait a second
PAU.1
' Locate it using "all"
STW.wt|0|BCS - Testform
' Locate the Button using "all"
SCW.anctb|1|Button|Next|0

' CLick the button, while its invisible
MLE.
' Get the Top-Handle
GCT.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Make it visible again
SWS.v
ENR.
```

Remarks

If you definitely know that the window is invisible, you can use the "u" - Flag - STW. 3878 which will only search for invisible windows and ignore visible windows.

The same applies for disabled windows, and the "f" - Flag - STW. 3857

Limitations:

See also:

- "e" - Flag - STW. 3855
- "f" - Flag - STW. 3857
- "u" - Flag - STW. 3878
- "v" - Flag - STW. 3880

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2("b" - Flag - STW.

MiniRobotLanguage (MRL)

"b" - Flag "window to back"

window to back

Intention

In most cases we want the window to come in front of other windows. so we can use it. For this we do not need any STW.-flag, as this is the default behavior of the STW.-command.

In some cases, we want just the opposite. We want a window to go behind all other windows. We sent a "window to back". For this we use the "b"-Flag.

It will locate the window "on Top" and sent it to back. See the example below.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

' Open 5 Terstforms and position them nicely
FOR.$$CNT|1|5
  EXE.?exepfad\Sample Apps\BSC-Testform.exe
  STW.p|#pid#
  SWP.(100+55*$$CNT),(200+55*$$CNT)
NEX.

FOR.$$CNT|1|50
'Move Testform "On top" to back
  STW.ctb|PBWindowClass:0|BCS - Testform
  ' Make small rest of 0.1 seconds so we can see it
  PAU.0.1
NEX.

' Close all Testforms
CAW.PBWindowClass:0|BCS - Testform
ENR.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.26.2."c" - Flag - STW.

MiniRobotLanguage (MRL)

"c" - Flag "class"

Search Top Window

Intention

Window-Class" value. This is the "window class" of the window we are looking for.

This flag needs a parameter. It supports 3.2 Standard-Search Pattern 341 and therefore its enough to give a unique part of the value. By default the value is not case sensitive.

```
STW.c|WindowClass
MMV.
```

Example

```
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
STW.c|#32768
' Locate the button
SCW.t|Ok
' Click the button
MLE.
ENR.
```

Remarks

Limitations:

See also:

- 3.2 Standard-Search Pattern 341
- "t" - Flag - STW. 3875

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2 "e" - Flag - STW.

"e" - Flag "enabled windows" Previous 3854 Top 41 Next 3857

MiniRobotLanguage (MRL)

"e" - Flag "enabled windows"

Search Top Window

Intention

To ignore disabled windows is one of the default settings of the STW. command.
Therefore you will not need to specify the "e"-Flag often.
It can however make sense, to override the default behavior by using the "f"-flag which will make STW. to ignore enabled windows and just look for disabled windows.
If you want to catch both, the "a"-flag is what you need.
However the "a"->Flag will automatically also enable the search for invisible windows. If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "e" - flag together with the "a" to search only windows that are enabled. No matter if they are visible or not.

Lets take a look, on "what will work and what will not work".
This will search only for enabled windows.

```
STW.ce|WindowClass
MMV.
```

Since this is a default behavior of `STW.`, we have wasted the "e" at this place. Its not needed.

Now we would like to search for windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
STW.cef|WindowClass
MMV.
```

The robot understands that we want enabled and disabled windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cav|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the invisible windows. We do this by adding the "v"-Flag after the "a". In this case the "v"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Start an Application only, if that Application is not yet
running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Next we disable the whole window
SWS.d
' Wait half of a second
PAU.0.5
' Try to locate it using "e"
' Will not work, but give an error
STW.wte|0|BCS - Testform
ENR.
```

Remarks

-

Limitations:

-

See also:

- "f" - Flag - STW. 3857
- "u" - Flag - STW. 3878
- "v" - Flag - STW. 3880
- "a" - Flag - STW. 3850

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.23 "f" - Flag - STW.

MiniRobotLanguage (MRL)

"f" - Flag "disabled windows"

Search Top Window

Intention

To ignore disabled windows is one of the default settings of the STW. command. Therefore if you want to search for a window that has been disabled, you will need the "f"-flag.

The "f"-Flag will override the default behavior. It will make STW. to ignore enabled windows and just look for disabled windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for invisible windows. If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "f" - flag together with the "a" to search only windows that are disabled. No matter if they are visible or not.

Lets take a look, on this. This will search only for disabled windows.

```
STW.cf|WindowClass
MMV.
```

Now we would like to search for windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
STW.cef|WindowClass
MMV.
```

The robot understands that we want enabled and disabled windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cav|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the invisible windows. We do this by adding the "v"-Flag after the "a". In this case the "v"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Start an Application only, if that Application is not yet
running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Next we disable the whole window
SWS.d
' Wait half of a second
PAU.0.5
' Try to locate it using "e"
' Will work, we get the window
STW.wtd|0|BCS - Testform

ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - STW. 3855
- "u" - Flag - STW. 3878
- "v" - Flag - STW. 3880
- "a" - Flag - STW. 3850

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2 "g" - Flag - STW.

"g" - Flag "goto if found" Previous 3857 Top 41 Next 3861

MiniRobotLanguage (MRL)

"g" - Flag "Goto if found"

Goto if found

Intention

If the window you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to jump to a Label in that case.
And you can use the "X"-Flag to tell the robot to just continue with the script in case of error - if the window is not found.
This would change the behavior of the command, but sometimes this can be a good solution. Take this example:

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - goto Label "GoOn" if the window was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.
```

In case you use the "g"-Flag, together with the "W" -Flag ("on Found Gosub"), the "W"-Flag will be ignored. The "g"-Flag has a generally higher priority.

Example

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - gotoLabel "Sub" if thewindow was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- "o" - Flag - STW. 3867
- "V" - Flag - STW. 3844
- "W" - Flag - STW. 3846
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2!"h" - Flag - STW.

"h" - Flag "handle" Previous 3859 Top 41 Next 3863

MiniRobotLanguage (MRL)

"h" - Flag "handle"

Search Top Window by handle

Intention

This flag is an exception. In fact it must not search the window, it will just localize the window "by handle". Of course this is a very efficient solution. Therefore if you have already localized your window, keep the handle in a variable and just feed it back to STW. when you need that window again.

```
' Localize by handle
STW.cta|ieframe|Update
' We grab the handle in a variable
HTV.$$MYW

' Later we can always get definitely this window,
' without further parameters
STW.h|$$MYW
MMV.

' Some more examples
STW.h|52367
MMV.

STW.h|#hnw#
MMV.
```

Example

```
'
' Localize by handle
STW.cta|ieframe|Update
' We grab the handle in a variable
: $$MYW=#hwn#

' Later we can always get definitely this window,
' without further parameters
STW.h|$$MYW
MMV.

ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

See also:

"E" - Flag - STW. 3825

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2("j" - Flag - STW.

"j" - Flag "WII. by number" Previous 3861 Top 41 Next 3865

MiniRobotLanguage (MRL)

"j" - Flag "WII. by number"

Wait for Input Idle and tell how long

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time must be given as a numeric value or variable.

Note the value you specify will replace the default internal value for "WII. without parameter", used for the "k" - Flag - STW. 3865 . But also for WII. if no parameters are given.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions are allowed. For example:

```
' WIll wait no longer then a half second
STW.cj|WindowClass|0.5
MMV.
```

Technically the "j"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement. Just that it is integrated and does not need a separate line of code.

Example

```
' j - wait 5 Seconds maximum
STW.cj|WindowClass|5
MMV.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

- WII. - Wait-for-Input-Idle 2909
- "k" - Flag - STW. 3865

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.27 "k" - Flag - STW.

MiniRobotLanguage (MRL)

"k" - Flag "WII. default"

Search Top Window

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time is taken from an internal value that is been set with the last WII. statement or with the "j" - Flag - STW. 3863.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions of a second are allowed.

```
STW.cj|WindowClass|5
MMV.
' Will do the same without repeating the value
STW.ck|WindowClass
MMV.
```

Technically the "k"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement without a parameter. Just that it is integrated and does not need a separate line of code.

Example

```
' Demonstrate the "j" and the "k" Flag
STW.cj|WindowClass|5
MMV.
' Will do the same without repeating the value
STW.ck|WindowClass
MMV.
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- "j" - Flag - STW. 3863
- WII. - Wait-for-Input-Idle 2909

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2 "o" - Flag - STW.

"o" - Flag "on not found Goto" Previous 3865 Top 41 Next 3869

MiniRobotLanguage (MRL)

"o" - Flag "on not found Goto"

Search Top Window

Intention

If the window you are looking for is not found within the specified waiting time, normally you will end up with an error-message.

Maybe you want just to jump at another place in the script the, and start a program. Or react in another way. That is where the "o" Flag enters the game. The script just continues. The window is not found and we tell the robot to jump to a Label in that case.

Take this example:

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
STW.toXw|BCS|GoOn|0
GTO.weit

:GoOn
 EXE.?exepfad\Sample Apps\BSC-Testform.exe

:weit

MBX.Ready
ENR.
```

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Example

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
' w - set waiting time to 0 seconds
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
STW.toXw|BCS|GoOn|0
GTO.weit

:GoOn
 EXE.?exepfad\Sample Apps\BSC-Testform.exe

:weit

MBX.Ready
ENR.
```

Remarks

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Limitations:

-

See also:

- "g" - Flag - STW. 3859
- "V" - Flag - STW. 3844
- "W" - Flag - STW. 3846
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.2 "p" - Flag - STW.

MiniRobotLanguage (MRL)

"p" - Flag "Process ID"

Search Top Window

Intention

You can localize a window safely, if you have the PID (Process ID) of the application behind that window.

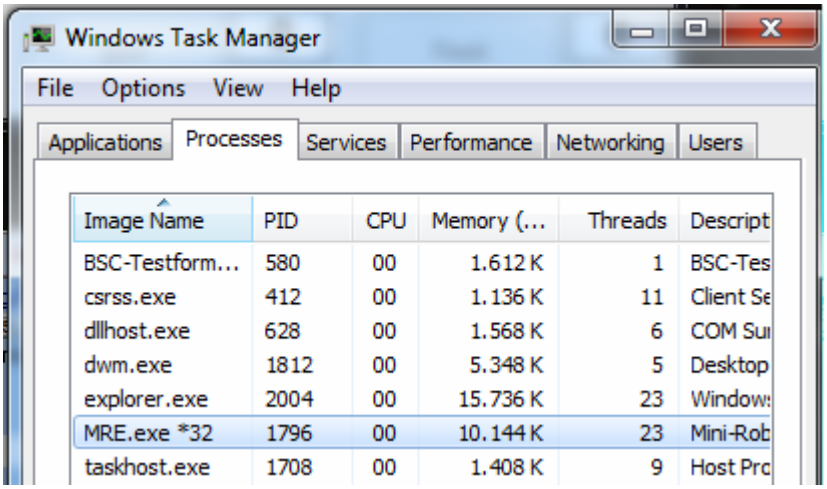
Be sure that you have read and understood the chapter 1.7. Application and PID 137.

Hint: If you have localized a window, you also have the PID of the process behind that window in `#pid#`. You can save it for later use in a variable. Like this:

```
STW.c|pbwind
: §§PID=#pid#

or use:

STW.c|pbwind
HTV.§§PID
```

In both cases, you get the PID of that process into the local variable §§PID 366.

You can see the PID of a process in the task-manager::



The PID value is a normal search parameter and can be used alone or together with other search parameters.

```
' Search a window from process with ID 3435
' and the given window class.
'
STW.ch|pbwind|580
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MMV.
```

Example

```
' Locate window only by PID
STW.p|580
MMV.

ENR.
```

Remarks

Limitations:

See also:

- 1.7. Application and PID 137
- GDF. - Get-Data-From 2889
- 1.6.1. Program Flow Control 121
- GPI. - Get Process Information 2893
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.3("q" - Flag - STW.

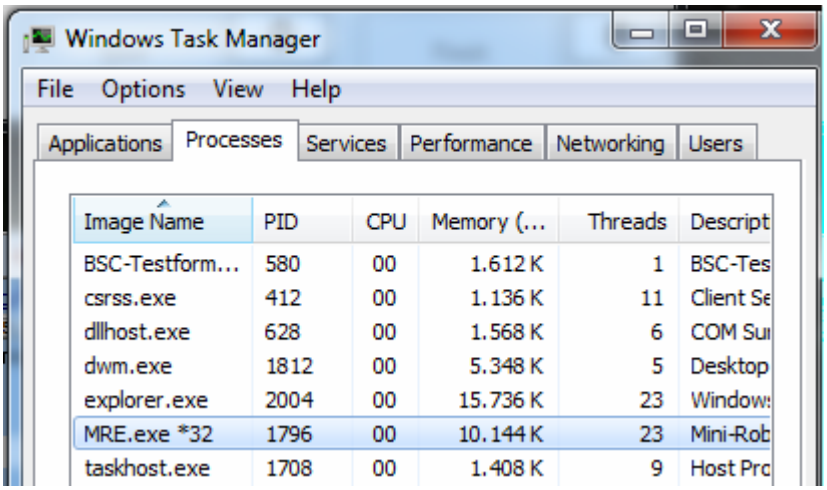
MiniRobotLanguage (MRL)

"q" - Flag "Thread ID"

Search Top Window

Intention

This is an advanced parameter. The chances are that you will never need it.
Processes have a number of Threads. See this picture here:



On the left side you can see the name of the process. Right of it the PID of the process. Later you see "Threads".
In a very simplified form, a Thread is a subprogram of that process, that runs an own small program. Generally the more threads a program has, the better it will scale up on multiple CPU-Core's.
While all Threads of a process belong to that process and therefore have the same PID, they have a different Thread-ID. In case you may need to separate between windows of the same process, which is "Multi-threaded", the "q"-flag is for you.

```
STW.cq|WindowClass|109872
MMV.
```

Hint: If you have localized a window, you also have the TID of the process behind that window in `#tid#`. You can save it for later use in a variable. Like this:

```
STW.c|pbwind
: §§TID=#tid#
```

This way, you get the TID of that Thread into the local variable §§PID 366.

If you don't "understand a word", or don't have very special cases, just ignore this flag.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe
' Locate the window
STW.p|#pid#
' Show the Thread-ID
MBX.#tid#
ENR..

ENR.
```

Remarks

Limitations:

See also:

- 1.7. Application and PID 137
- "p" - Flag - STW. 3869

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.3."r" - Flag - STW.

"r" - Flag "result number" Previous 3871 Top 41 Next 3875

MiniRobotLanguage (MRL)

"r" - Flag "result number"

Search Top Window

Intention

There may be multiple results in a window-search with `STW.`. Normally you get the first result that fits to your search parameters.
Using the "r"-Flag you can define which result you want, Nr.1, Nr.2 or Nr. 20. Just add the "r"-Flag and the result number.

If omitted, always result Nr.1 is returned.

Note: If there are not enough results, your search will fail!
Therefore you can also use this feature to count windows.

```
'We'll get the 3rd result with that window class
STW.cr|WindowClass|3
MMV.
```

Example

```
'We'll get the 4th result
' with that window class
'
STW.cr|WindowClass|4
MMV.
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.3 "t" - Flag - STW.

"t" - Flag "text" Previous 3873 Top 41 Next 3878

MiniRobotLanguage (MRL)

"t" - Flag "text"

Search Top Window

Intention

This flag is one of the most important search parameters. It is the window-text. Most often you can even see the text of a top-window without using the windows-Inspector.
Here is the simplest case how you can grab a window.

```
STW.t|TestApp
MMV.
```

The text parameter is a string-value and therefore supports 3.2 Standard-Search Pattern 341.
Let us try some of these.

As always with Search patterns, it is enough to give a unique part of the value. By default the text is not case sensitive. You can change that behavior using the search patterns. Take the sample-script below and replace the line:

```
' Locate the window
STW.t|Testf
```

with the following lines. This will help you to understand the use of Patterns for text search.

```
' Lets start with this
STW.t|{&DOCAPS:Testf}
```

This will force the text to be case-sensitive! Try this:

```
' This will not find the window
' because of case-sensitivity!
STW.t|{&DOCAPS:testf}
```

It does not find the window because the &DOCAPS: switches case-sensitivity on.
Now how about this:

```
STW.t|{&EXACT:bcs - Testform}
```

It does find the window. But then whats the "&EXACT:" good for?
Try this:

```
STW.t|{&EXACT:bcs - Testf}
```


© 2023 Theo Gottwald * https://www.smart-package.com

It does not find the window because the &EXACT: switches off the default behavior that a part of the text is enough. CAn it be combined with the &DOCAPS: ?

We try. This should work:
```
STW.t|{&EXACT:&DOCAPS:BCS - Testform}
```
and this should not work:
```
STW.t|{&EXACT:&DOCAPS:bCS - Testform}
```

Yes, everything work as expected.

Do you know the old times, when you had simple Wildcards as patterns? The"*****" and the "**?**" ?

You can have them back if you want. Try this:

```
STW.t|{&WILDC:b?? - Testform}
```

Yes, it works. And how about:

```
STW.t|{&WILDC:b?? - Tes*orm}
or
STW.t|{&WILDC:b?? - *s*or?}
```

It works.
We kept them for you.
You can still use even the old DOS-Patterns if you really want.
But there is much more if you use the new patterns.

How about a fuzzylogic and a OR'ed combination of two patterns?

```
STW.t|{&WILDC:b?? }&OR:{BCS-tesform&FUZZY:6}
```

and you can not only combine two patterns, you can combine as many patterns using &OR: as you like! If you want to support many languages, this is the way to go.

No matter if there is a "YES" on that button or a "Qui" or whatever. So long you know it, or at least a bit of it, you can locate it using Search patterns.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe

' Locate the window
STW.t|Testf

' Close the window
CAW.PBWindowClass:0|BCS - Testform
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Limitations:

See also:

- 3.2 Standard-Search Pattern 341
- "c" - Flag - STW. 3854

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.3 "u" - Flag - STW.

MiniRobotLanguage (MRL)

"u" - Flag "invisible"

Search Top Window

Intention

To ignore invisible windows is one of the default settings of the STW. command. Therefore if you want to search for a window that has is invisible, you will need the "u"-flag.

The "u"-Flag will override the default behavior. It will make STW. to ignore visible windows and look for invisible windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "u" - flag together with the "a" to search only windows that are invisible. No matter if they are enabled or not.

Lets take a look, on this. This will search only for invisible windows.

```
STW.cu|WindowClass
MMV.
```

Now we would like to search for windows which are visible, as well as for invisible windows. We can do that in two ways.The first way is like this:

```
STW.cuv|WindowClass
MMV.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cae|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
CAW.PBWindowClass:0|BCS - Testform
'
' Start an Application only,
' if that Application is not yet running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

STW.p|#pid#
' Next we make the window invisible
SWS.i
' Wait half of a second
PAU.2.5
' We locate it even though its invisible
STW.wtu|1|BCS - Testform
' and make it visible again
SWS.v
ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - STW. 3855
- "f" - Flag - STW. 3857
- "v" - Flag - STW. 3880
- "a" - Flag - STW. 3850
- 1.8. Selecting the Application (Topwindow) 150

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.34 "v" - Flag - STW.

"v" - Flag "visible" Previous 3878 Top 41 Next 3882

MiniRobotLanguage (MRL)

"v" - Flag "visible"

Search Top Window

Intention

To ignore invisible windows is one of the default settings of the STW. command. Therefore you will not need to specify the "v"-Flag often.

It can however make sense, to override the default behavior by using the "u"-flag which will make STW. to visible windows and just look for invisible windows.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also switch on the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "v" - flag together with the "a" to search only windows that are visible. No matter if they are enabled or not.

Lets take a look, on how to do this. This will search only for visible windows.

```
STW.cv|WindowClass
MMV.
```

Since this is a default behavior of `STW.`, we have wasted the "v" at this place. Its not needed.

Now we would like to search for windows which are visible, as well for invisibble windows. We can do that in two ways.The first way is like this:

```
STW.cuv|WindowClass
MMV.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cae|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
CAW.PBWindowClass:0|BCS - Testform
'
' Start an Application only,
' if that Application is not yet running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

STW.p|#pid#
' Next we make the window invisible
SWS.i
' Wait half of a second
PAU.2.5
' We locate it even though its invisible
STW.wtu|1|BCS - Testform
' and make it visible again
SWS.v
ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - STW. 3855
- "f" - Flag - STW. 3857
- "u" - Flag - STW. 3878
- "a" - Flag - STW. 3850
- 1.8. Selecting the Application (Topwindow) 150

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.26.3!"w" - Flag - STW.

MiniRobotLanguage (MRL)

"w" - Flag "wait time"

Search Top Window

Intention

How long do we wait if a window will appear or not?
Let's try it.

```
: $$TIM=#dtime#
STW.tX|nix
: $$TIM=#dsince#
MBX.$$TIM
ENR.
```

This i what we get.

It's currently about 15 seconds. When you read this it can be another value.
In fact this is the default time that we wait for a window before we run into an error-message or do something else (like in this script).

Now, its very important to understand that we do not generally wait 15 seconds, and then look if the window has appeared.
What we do is, we try ... try ... and try and check permanently, if the window has appeared. If the window appears any time, the script will immediately continue.

If the time is over, we do something else.

Now 15 seconds is enough for some cases, in many cases the window will appear after 45 seconds or any other time. Now this is where the "w"-flag enters the game.

Using the "w"-flag we can tell the robot how long he should wait. It can be 0 seconds like this:

```
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
```

or 360 seconds like here.

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.wtu|360|BCS - Testform
```

and there is no real limitation.

Example

Remarks

-

Limitations:

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.27 SWP. - Set Window Position and Size

MiniRobotLanguage (MRL)

SWP. Command

Set Window Position and Size

Intention

The `SWP.`-Command can be used to set the position or to change the size of the actual
selected Top- or Child-Window. You need to select/locate a child or Top-Window using the STW. or the `LAP.` command before you can use `SWP.`

The window position is been calculated from the top-left corner of the window or child-window.

The `SWP.` command has a lot of additional features. If you want to place the window just in the middle of the screen, without changing the size, you can just write:

```
' Place the actual localized window in the middle of the
screen
SWP.mid
```

You can still resize the window if you do this:

```
' Place the actual localized window in the middle of the
screen
' and resize it to 600x400 Pixel
SWP.mid|600,400
```

Using some more tricks, you can for example make the window to quarter of its current size while not changing the position of the window at all.

```
SWP.|hw,hw
```

Here are some more complete examples:

```
' locate / select the Window that shall be resized using STW.
STW.c|ieframe
' The window is moved to position 0,0|640,400
' and resized to 640x400
SWP.0,0|640,400

' The window is resized to 500x500 pixel
SWP.|500,500

' The window is resized to 100x200 and vertically moved to
y=300
SWP.,300|100,200
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
' The window is only moved, not resized.
SWP.100,200
```

Syntax

SWP.[P1][,P2][|P3][,P4][|T]

Parameter Explanation

P1 - Left-Position
P2 - Top - Position
P3 - X - Size
P4 - Y - Size

You can specify three special Options as [|T]:
T - use Top-Window and whole Desktop Area of primary Monitor
M - use Top-Window and whole Desktop Area of all Monitors
F - use Desktop without Traybar and Top-Window
C - use actual localized window and whole Desktop Area. If you do not specify any of these additional Options, the actual window is used, and the whole-Desktop Area.

If you want to place the window just in the middle of the screen, without changing the size, you can just write:

```
' Place the actual localized window in the middle of the
screen
SWP.mid
```

For the position parameters **P1** (Left-Position) you can specify these words:

- **"mid"**
- **"lb"** (left bound) equals to zero
- **"rb"** (right bound) moves the window to the right border of the primary desktop

or just the first letter of these. They will take the actual size and calculate the new size as expected.

For the position parameters P2 (Top-Position) you can specify these words:

- **"mid"**
- **"tb"** (top bound)
- **"bb"** (bottom bound)

or just the first letter of these. They will take the actual size and calculate the new size as expected.

For the size parameters P3 and P4),you can specify the following words, which are based on the desktop size. Note that these calculations are influenced by adding the special options "**|T**" or "**|F**".

© 2023 Theo Gottwald * https://www.smart-package.com

- "half" - half desktop size
- "full" - full desktop size
- "quarter" - quarter desktop size

or use these, which calculate the new size based on the current window size. Note that these options are influenced by adding special options "**|T**" or "**|C**" to the command.

- "halfwin","hw"
- "doublewin","dw"
- "quarterwin","qw"

They will take the actual size of the desktop and calculate the new size of the window as a part of the desktop size. It will use the size of the primary desktop, no matter if you have dual- or multi-screen.

You can omit any of the Parameters or just write "**x**".. In these cases, the parameters will be internally substituted with the current values. If you write for example:

```
' The window is horizontally moved to position 100, the
vertical position is not changed
SWP.100,x

' The window is moced into the center of the desktop
SWP.mid,mid

' The window is only resized to half of its size in both
directions
SWP. half,half

' The window is only moved, not resized. It's move to the left
desktop border.
SWP.lb,x
```

Example

```
'-------------------------------------------
' SWP.-Example 1
'-------------------------------------------
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe
' Locate the window
STW.p|#pid#
' Position it "around"
SWP.lb,tb
PAU..1
SWP.mid,mid
PAU..1
SWP.rb,bb
PAU..1
SWP.|(#iw#/2)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.

'-------------------------------------------
' SWP.-Example 2
'-------------------------------------------
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

' Open 5 Terstforms and position them nicely
FOR.$$CNT|1|5
  EXE.?exepfad\Sample Apps\BSC-Testform.exe
  STW.p|#pid#
  SWP.(100+55*$$CNT),(200+55*$$CNT)
NEX.

FOR.$$CNT|1|50
'Move Testform "On top" to back
  STW.ctb|PBWindowClass:0|BCS - Testform
  ' Make small rest so we can see it
  PAU..1
NEX.

' Close all Testforms
CAW.PBWindowClass:0|BCS - Testform
ENR.
```

Remarks

Additional to the features that SWP. delivers, you can always use these 3.1 Systemvariables 334 for size or position calculations.
They are based on the last located object (window or whatever).
As always, use mathematical brackets () to enclose calculations. 80

```
#ssx#,#ssy# -   Screen size X,Y
#cwh#   -       Debug Console Window-Handle
#wsx#,#wsy# -   Work-area-Size X,Y
#sbp#   -       Screen-bits per pixel (Colordepth)
#fex#   -       FEX - Elements left on the FEX-Stack
#fxd#   -       FEX - Nesting-Depth
#il#            -       actual Element, left side X
#it#            -       actual Element, upper side Y
#ir#            -       actual Element, right side X
#ib#            -       actual Element, bottom side y
#iw#            -       actual Element, width X
#ih#            -       actual Element, height Y
#icx#   -       actual Element, Centerpoint X
#icx#   -       actual Element, Centerpoint Y
```

Example for this:

```
' Will make the x-size of the window half of current x-size
SWP.|(#iw#/2)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Will move the window to its own centerpoint
SWP.(#icx#)
```

Limitations:

Not all Windows can be resized to exactly any size. This depends on the Internals of the target software as well as possibly on the used OS.
Using Windows 7, for example:

```
PRT.Output-Window. $crlf$ Close the Utility with this window!
' Console Fenster rechts anlehnen
STW.ctwAXo|consolewindowclass|system mess|2|Lab_nocon
SWP.0,0|600,400
PAU.15
:Lab_nocon
```

This window will be sizes to the nearest possible size, which is 605,395 in my testcase.

See also:

- "b" - Flag - STW. 3852
- SWS. - Set Window State 3889

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.28 SWS. - Set Window State

SWS. - Set Window State Previous 3884 Top 41 Next 3584

MiniRobotLanguage (MRL)

SWS. Command

Set Window State

Intention

Make a window invisible. Or enable or disable a button.
Minimze a window, or restore it, if it was minimized or maximized before.

These and more such things can be done using `SWS.`

SWS. changes the State of the actual Top- or Child-window. If you give multiple prototypes they will apply in the order from left to right.

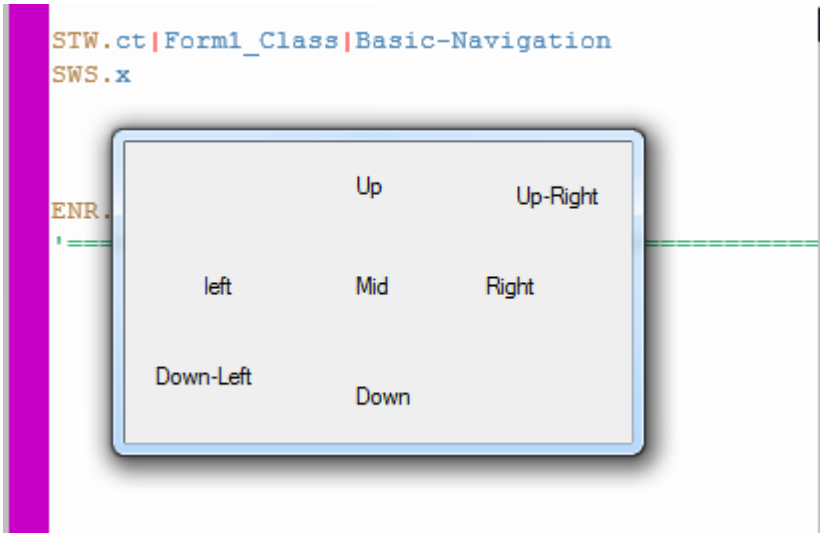
Example:
```
SWS.evna
```

Example, this will make a localized window invisible:
```
SWS.i
```

Example, this will make a localized window visible again:
```
SWS.v
```

Normally all of the SWS. Options can be changed back to their originasl state. Unlike **c** and **x**. These are **two special Options** that can remove the Window-Caption or give a Window a Caption. However these two Options should be used with care, as they can damage a running application-window in a way that you may need to restart the application.
Therefore test first if you can use them in your special case,.

```
SWS.x
```

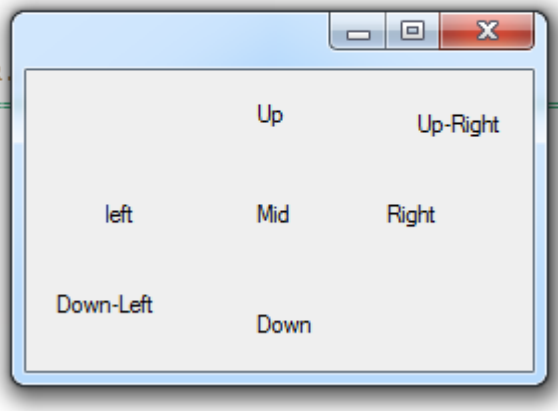


© 2023 Theo Gottwald * https://www.smart-package.com

This is how a window may look after you have removed the Caption using `SWS.x`.
This may not work with any window. Its for those cases, where you do not want the User to be able to move windows on the desktop around. Also there is no close Gadget anymore.

`SWS.c`

```
STW.ct|Form1_Class|Basic-Navigation
SWS.c

ENR
'=
```



Here the Caption is back, using `SWS.c`
Note that the window can not be totaly restored like it was before.
While there is a caption, the window may still have a different behavior when being resized.
Here is the window as it looked before.
As you can see it looks a bit different.

Syntax

SWS.[P1][|P2]

© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** Prototype(s)

```
' a - activates the window after setting the state.
' e - enabled
' d - disabled
' v - visible
' i - invisble
' m - minimzed
' n - normal (not minimized)
' z - zoomed (maximized)
' u - unzoomed
' t - use actual Top-Window instead of actual window
' c - add Caption to Window (c<>x)
' x - remove caption from Window (c<>x)
```

Multiple states can be given. In that case they are connected with AND.

If no parameters are specified, "evn" is the default parameter.

P2 **-** (NUM/VAR) optional parameters, window-handle. When given, this window is used instead of the actual localized window.

Example

```
'**********************************
' SWS.-Sample 1
'**********************************
STW.t|Sample
MMV.
SWS.evna

STW.t|Sample
HTV.$$HWN

STW.t|anotherwindows
MMV.
' Note that the first window is going to be tested.
SWS.eva|$$HWN
END.


'**********************************
' SWS.-Sample 2
'**********************************

'#EXE:?pfad\
'#SPI:ForceWrite
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Start an Application only, if that Application is not yet
running.
'
STW.wtgX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
STW.wt|0|BCS - Testform
' Nextwe make the thing invisible
SWS.i
' Wait 5 seconds
PAU.5
' Locate it using "all"
STW.wta|0|BCS - Testform
' Locate the Button using "all"
SCW.anctb|1|Button|Next|0

' Click the button, while its invisible!
MLE.
' Get the Top-Handle
GCT.
' Make it visible again
SWS.
ENR.
```

Remarks

You can disable buttons, immediately after they appear using the `SWS.d` -command. This way you can prevent the user from clicking buttons, he should not click.

You can check (child-)window-states with IWS. / NWS. - If Window State 3724.

Limitations:

-

See also:

- IWS. / NWS. - If Window State 3724
- IWE. / NWE. - If Window Enabled 3718
- IWI. / NWI. - If Window Iconized 3721
- IWV. / NWV. - If Window Visible 3727
- IWZ. / NWZ. - If Window Zoomed 3730

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.29 WCW. - Wait for new Child-Window

3.83.29.1 ! WCW. - Wait for new Child-Window

MiniRobotLanguage (MRL)

WCW. Command

Wait for new Child Window

Intention

This command localizes a specified Child-Window (Control, Button etc.) within a previous localized Top-Window (Application-Window).

Unlike SCW. the WCW. Command will only recognize new Childwindows or Child-Windows that change after the Command was called.

To understand this command, be sure to read the chapters:
1.8. Selecting the Application (Topwindow) 150
and
1.9. Locating Child-Windows 162
first.

It will then sets the Mouse-Target Point into the window-center.

WCW. has a lot of features, which are controlled using the 1.4. The Prototype Parameter System 73 .Be sure you have read and understand that chapter before trying to understand WCW.

Unless using the "T"-Option, the WCW.-command needs a Top-Window being localized first. Therefore please read the section 1.8. Selecting the Application (Topwindow) 150 carefully.

The Command has a "Rollback" Feature, when the Child-Window is not found in few Seconds. It jumps back to the STW. (Search Top-Window) and tries to get a new Top-Window, which possibly has the wanted button or control.
This will continue, until its time has run out.
If you use the DBM.1 -command, you can see these "Rollback-Jumps" in the Debugging-Console. See the example below.

```
DBM.1
STW.Act|PBWindowClass:0|TestApp
' This button/childwindow does not exist
WCW.nctb|2|Button|Button 23|.5
MLE.
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
[10:13:11] STW.Act|PBWindowClass:0|TestApp
[10:13:11] SCW.nctb|2|Button|Button 23|.5
[10:13:12] STW.Act|PBWindowClass:0|TestApp
[10:13:12] SCW.nctb|2|Button|Button 23|.5
[10:13:12] STW.Act|PBWindowClass:0|TestApp
[10:13:12] SCW.nctb|2|Button|Button 23|.5
[10:13:13] STW.Act|PBWindowClass:0|TestApp
[10:13:13] SCW.nctb|2|Button|Button 23|.5
```

You can control this Feature using the "b" - Backjump-Time Flag 3764.

Syntax

WCW.[Prototypes|][Parameters|]

Parameter Explanation

Parameter

```
 a - (np) "all" - search all windos, includes
     invisible and disabled windows.
 b - (p) "backjump"-flag, time in seconds, 0 disables the jump
 c - (p)  window-class
 d - (p) window-depth
 e - (np) enabled, window must be enabled
 f - (np) disabled, window must be disabled
 g - (p) Goto on found
 h - (p) handle. Immediately locate the window with given
handle.
     All other prototypes will be ignored.
 i - (p) window-id
 j - (p) WaitForInputIdle (Value)
 k - (np) WaitForInputIdle (no Value), use value
      from last Call/default-value
 n - (p) Fensternumber in der Ebene ("WR->SWC")
 o - (p) on not found goto (Label)
 p - (p) Process-ID (PID 137)
 q - (p) Thread-ID (TID)
 r - (p) result number ("WR-SPC-Count")
 s - (np) internal reversing at child windows search
 t - (p) text
 u - (np) invisible, window must be invisible
 v - (np) visible, window must be visible
 w - (p) total waiting-time in Seconds, fractions (0.5)
     are possible, zero is possible.

 E - (np) Add Window to Exclude List after localization
 I - (p) p = handle. Locate window and remove window
     from Exclude list.
 K - (n) Immediately clear Exclude-List before specified
search
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Q - (n) use previous Prototypes
S - (n) Search only next Level (SWC-Option)
T - (n) Search desktopwide, ignore localized Top-Window
V - (p/Label) Gosub on not found
W - (p/Label) Gosub on  found
X - (n) Run without Error on not found
```

Explanation:
(np) -> only prototype, no additional parameter required
(p) -> prototype requires additional parameter
(NUM/VAR) - window handle/variable
(p/Label) - Label-Parameter required, can also be a variable that contains the label

P2 **-** Parameters, separated by | according to the prototypes

Example

```
STW.Act|PBWindowClass:0|TestApp
WCW.nctE|2|Button|Button 2
' Coordinates relative to left, upper corner of button
MPO.o|42,22
MLE.
```

Remarks

When using `WCW.` to check if a child-window is available or not, it makes sense that you use the "b"-flag with a value of "0" to disable the backjump.

Limitations:

-

See also:

- 1.4. The Prototype Parameter System 73
- 3.2 Standard-Search Pattern 341
- 1.5. Features and Hints 150
- 1.9. Locating Child-Windows 162
- 1.8. Selecting the Application (Topwindow) 150
- 1.7. Application and PID 137
- 3.1 Systemvariables 334
- !. STW. - Description 3814

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30 WTW. - Wait new Top Window

3.83.30.1 ! WTW. -Wait New Top-Window

MiniRobotLanguage (MRL)

WTW. Command

Wait for New Top Window

Intention

This command localizes a specified Top-Window and sets the actual point into its center.
Unlike STW., the WTW.-Commandwill ignore Windows that are already open!
The WTW.-Command will wait for a Window that appears after the Command has been called.

All following `SCW`-commands will be within this window.

`WTW`. has a default window-treatment 3814 to ensure that any further action in that window will be effective. This default treatment will do following:
- normalize minimized windows (like the "D"-Flag
- windows that are outside the desktop-work area will be moved to 0,0 (like "G"-Flag)
- The window will be moved in front of all other (non-TopMost 3838)-windows.
- The window will receive the input focus. (like the "J"-Flag)

This default window-treatment can be completely deactivated using the "A"-Flag 3819.

This makes sense, for example when working with Menu's or other windows which disappear when the default treatment is being applied.

Alternatively the default window treatment can be influenced using other flags, like "B" or "G".

If the window should be brought into work area, without other treatment, use the "B"-Flag. "G", will just prevent the repositioning of the window and let the other treatment-steps happen.

All window search operations change the Timeout System Variables `#tio#` and `#til#`. Using these you can find out if a window was found or not, even if you have used the"X" Flag (X - run without Error).

Hint: Please note that the Flags have a priority from left to right (highest Priority). If you add other flags after the "A" there may be a treatment. Because the more right the flags, the higher their priority. If you place the "A" last inside the prototypes, all involved treatment will be disabled.

See two examples.

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this first case the "A" Flag has priority over the "C" Flag, because its right of it.

```
WTW.tCA|explorer
MMV.
```

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
WTW.tAC|explorer
MMV.
```

Some Flags deactivate the default treatment or parts of it. See description below.

Hint: If you omit all parameters like this:
`WTW.`
the command will automatically restore the parameters from the previous call. This makes it easy to always locate the same Top-Window.Note that this also applies to `ITW.` as both use the same parameters-store.

Syntax

WTW.P1[|P2]...[|Px]

Parameter Explanation

P1 – Prototype
P2 .. **PX** - Parameters according to prototypes

```
A - (np) Switch off default window treatment completely
B - (np) disables following parts of "default treatment":
     NO Restore from Minimized, NO Set Input-Focus,
     NO Bring to Front
C - (np) If the window i maximized, it will be restored to
    normal size
D - (np) If the window is minimized,  will be restored to its
    previous size, before it was minimized,
E - (np) Put Localize window, then add window to
    Search Exclude List.
F - (np) Move window to center of primary desktop
G - (np) disable default window-treatment-"repositioning if
     outside"
H - (np) a window that is outside the primary desktop working
area
    will be repositioned. Default is that the window can be
also
    on a second desktop.
I - (NUM/VAR) Remove window with the given window-handle from
the
    search-exclude-list.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
J - (np) Bring window to Front. Does not switch off default
window
    treatment.
K - Immediately clear Exclude-List before specified search
M - (np) Set window to TOPMOST. Such windows will be on Top
of all
    other Non-TOPMOST windows. DANGER - even Messageboxes may
be
    hidden behind a TOPMOST window. Use this with caution!
N - (np) remove the TOPMOST Property from a window.
Q - use previous Prototypes (only prototypes)
V - (p/Label) Gosub Line/Label if the window was NOT found
W - (p/Label) Gosub Line/Label if the window was found
X - (np) run without error if window was not found. Timeout
Flag
    #tio# and #til# will anyway be set and can be used.

a - (np) changes internal settings that "all" windows will be
    found. This includes invisible and disabled windows.
b - (np) "back" - moves window to the back, behind other
windows.
c - (p) "class", window-class parameter
e - (np) enabled, the window must be enabled or it will
ignored.
f - (np) disabled, the window must be disabled or it will
ignored.
g - (p/Label) Goto Line/Label if the window was found
h - (p) localize the window directly using a given window
handle.
    Note, if "h" is given, other search conditions will
    be ignored.
j - (p) WaitForInputIdle, you can specify the WII timeout
k - (np) WaitForInputIdle - no parameter, use the default
    WII timeout.
o - (Label/Variable) on not found goto (Label)
p - (p) process-ID, search only windows from process with
this
    given PID
q - (p) ThreadID,, search only windows from process with this
    given TID
r - (p) No. of result to choose (Result-Number for Top-
Windows)
t - (p) text, window-text (often the window caption)
u - (np) invisible, search only invisible windows
v - (np) visible, search only visible windows
w - (p) total waiting-time in Seconds, 0.5 (FP-numbers) is
    possible.
```

Explanation:

(np) -> only prototype, no additional parameter required

(p) -> prototype requires additional parameter

(NUM/VAR) - window handle/variable

(p/Label) - Label-Parameter required, can also be a variable that contains the label

P2 - Px - Data according to the prototypes, separated by |

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'MR Script-file:
'Version:
'Purpose: STW.-Features Demo
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' g ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
'
STW.wtgX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Now we can use the Application and press Buttons
STW.wt|0|BCS - Testform
SCW.nctb|1|Button|Next|0
' Compatible Button Click
MLC.

'------------------------------------------------------------
ENR.
```

Remarks

To select/locate a Top- Window, you will most often use `STW.` or in few cases `LAP.`

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- **1.4. The Prototype Parameter System** 73
- **3.2 Standard-Search Pattern** 341
- 1.5. Features and Hints 150
- 1.9. Locating Child-Windows 162
- 1.7. Application and PID 137
- 3.1 Systemvariables 334
- LAP. - Locate at Point 599

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2 "a" - Flag - WTW.

"a" - Flag "all windows" Previous 3848 Top 41 Next 3852

MiniRobotLanguage (MRL)

"a" - Flag "all windows"

Search Top Window

Intention

In most cases, we want to act on visible and enabled window only.
Therefore the STW. command ignores by default windows, if they are disabled or invisible.

The "a" flag changes this behavior. Using the "a"-Flag, the STW. command will locate "all windows". This includes Top-Window which are invisible or disabled, or both.
Using the Flag "a" ALL Top windows will be found if they meet the search criteria.

Example

```
'MR Script-file:
'Version:
'Purpose: STW.-Features Demo
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Next we make the thing invisible
SWS.i
' Wait a second
PAU.1
' Locate it using "all"
STW.wt|0|BCS - Testform
' Locate the Button using "all"
SCW.anctb|1|Button|Next|0

' CLick the button, while its invisible
MLE.
' Get the Top-Handle
GCT.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Make it visible again
SWS.v
ENR.
```

Remarks

If you definitely know that the window is invisible, you can use the "u" - Flag - STW. 3878 which will only search for invisible windows and ignore visible windows.

The same applies for disabled windows, and the "f" - Flag - STW. 3857

Limitations:

See also:

- "e" - Flag - STW. 3855
- "f" - Flag - STW. 3857
- "u" - Flag - STW. 3878
- "v" - Flag - STW. 3880

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.3 "A" - Flag - WTW.

"A" - Flag "No windows treatment" Previous 3814 Top 41 Next 3821

MiniRobotLanguage (MRL)

"A" - Flag "No windows treatment"

Search Top Window

Intention

STW. has a default window-treatment to ensure that any further action in that window will be effective. This default treatment will do following:
- normalize minimized windows (like the "D"-Flag
- windows that are outside the desktop-workarea will be moved to 0,0 (like "G"-Flag)
- The window will be moved in front of all other (non-TopMost)-windows.
- The window will receive the input focus. (like the "J"-Flag)

This default window-treatment can be completely deactivated using the "A"-Flag.

This makes sense, for example when working with Menu's or volatile windows, which disappear when the default treatment is being applied.

Usage:

```
STW.cA|#32768
MMV.
```

Please note that the Flags are internally analyzed in the order in which they are given. If you add other flags after the "A" there may be a treatment.
If you place the "A" last inside the prototypes, all such treatment will be disabled.

In this case the "A" Flag has priority over the "C" Flag, because its right of it.

```
STW.tCA|explorer
MMV.
```

In this case the "C" Flag is right of the "A" Flag and therefore has priority over the "A" Flag.

```
STW.tAC|explorer
MMV.
```

Example

```
'MR Script-file:
'Version:
'Purpose: STW.-Features Demo
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'Author: Theo Gottwald
'Creation date: 09.12.11
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
STW.cA|#32767
MMV.
ENR.
```

Remarks

Alternatively the default window treatment can be influenced using other flags, like "B" or "G".

If the window should be brought into work area, without other treatment, use the "B"-Flag. "G", will just prevent the repositioning of the window and let the other treatment-steps happen.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.4 "b" - Flag - WTW.

"b" - Flag "window to back" Previous 3850 Top 41 Next 3854

MiniRobotLanguage (MRL)

"b" - Flag "window to back"

window to back

Intention

In most cases we want the window to come in front of other windows. so we can use it. For this we do not need any STW.-flag, as this is the default behavior of the STW.-command.

In some cases, we want just the opposite. We want a window to go behind all other windows. We sent a "window to back". For this we use the "b"-Flag.

It will locate the window "on Top" and sent it to back. See the example below.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

' Open 5 Terstforms and position them nicely
FOR.$$CNT|1|5
  EXE.?exepfad\Sample Apps\BSC-Testform.exe
  STW.p|#pid#
  SWP.(100+55*$$CNT),(200+55*$$CNT)
NEX.

FOR.$$CNT|1|50
'Move Testform "On top" to back
  STW.ctb|PBWindowClass:0|BCS - Testform
  ' Make small rest of 0.1 seconds so we can see it
  PAU.0.1
NEX.

' Close all Testforms
CAW.PBWindowClass:0|BCS - Testform
ENR.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.30.5 "B" - Flag - WTW.

"B" - Flag "partial treatment" Previous 3819 Top 41 Next 3823

MiniRobotLanguage (MRL)

"B" - Flag "partial treatment"

Search Top Window

Intention

STW. has a default window-treatment to ensure that any further action in that window will be effective. This default treatment will be reduced, when the "B" - Flag is set.

When the "B" Flag is in effect, windows that are outside the desktop-workarea will be moved to the upper, left corner of the desktop - that is position 0,0. Thats the only treatment that will be done.

Also this can be disabled with the "G"-Flag. Means, that the flags "G" and "B" together are just like the "A" flag,

Example:

```
STW.Bc|WindowClass
MMV.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
STW.cB|#32767
' Locate the button
SCW.t|Ok
' Click the button
MLE.
ENR.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.30.6 "C" - Flag - WTW.

MiniRobotLanguage (MRL)

"C" - Flag "restore if maximized"

Search Top Window

Intention

If the window is maximized, this flag will bring the window back to its normal size.
You can maximize, minimize and restore windows using the SWS. - Set Window State 3889 command.

It doesn't matter, if the window was minimzed before it was maximized.
The window will be normal size after this flag.

Example

```
STW.ct|PBWindowClass:0|TestApp
' Maximize Window
SWS.z
' Wait 1 Second
PAU.1
' Window will be restore to normal size
STW.ACct|PBWindowClass:0|TestApp
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.7 "c" - Flag - WTW.

"c" - Flag "class" Previous 3852 Top 41 Next 3855

MiniRobotLanguage (MRL)

"c" - Flag "class"

Search Top Window

Intention

Window-Class" value. This is the "window class" of the window we are looking for.

This flag needs a parameter. It supports 3.2 Standard-Search Pattern 341 and therefore its enough to give a unique part of the value. By default the value is not case sensitive.

```
STW.c|WindowClass
MMV.
```

Example

```
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
STW.c|#32768
' Locate the button
SCW.t|Ok
' Click the button
MLE.
ENR.
```

Remarks

Limitations:

See also:

- 3.2 Standard-Search Pattern 341
- "t" - Flag - STW. 3875

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.8 "D" - Flag - WTW.

"D" - Flag "restore if minimzed" Previous 3823 Top 41 Next 3825

MiniRobotLanguage (MRL)

"D" - Flag "restore if minimized"

Search Top Window

Intention

If the window is minimized, this flag will bring the window back to its normal size.
You can maximize, minimize and restore windows using the SWS. - Set Window State 3889 command.

It does not matter, if the window was maximized before it was minimized.
The window will be normal size after this flag.

Example

```
STW.Act|PBWindowClass:0|TestApp
SWS.z
SWS.m
PAU.1
STW.ADct|PBWindowClass:0|TestApp
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.9 "e" - Flag - WTW.

MiniRobotLanguage (MRL)

"e" - Flag "enabled windows"

Search Top Window

Intention

To ignore disabled windows is one of the default settings of the STW. command.
Therefore you will not need to specify the "e"-Flag often.
It can however make sense, to override the default behavior by using the "f"-flag which will make STW. to ignore enabled windows and just look for disabled windows.
If you want to catch both, the "a"-flag is what you need.
However the "a"->Flag will automatically also enable the search for invisible windows. If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "e" - flag together with the "a" to search only windows that are enabled. No matter if they are visible or not.

Lets take a look, on "what will work and what will not work".
This will search only for enabled windows.

```
STW.ce|WindowClass
MMV.
```

Since this is a default behavior of `STW.`, we have wasted the "e" at this place. Its not needed.

Now we would like to search for windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
STW.cef|WindowClass
MMV.
```

The robot understands that we want enabled and disabled windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cav|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the invisible windows. We do this by adding the "v"-Flag after the "a". In this case the "v"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Start an Application only, if that Application is not yet
running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Next we disable the whole window
SWS.d
' Wait half of a second
PAU.0.5
' Try to locate it using "e"
' Will not work, but give an error
STW.wte|0|BCS - Testform
ENR.
```

Remarks

-

Limitations:

-

See also:

- "f" - Flag - STW. 3857
- "u" - Flag - STW. 3878
- "v" - Flag - STW. 3880
- "a" - Flag - STW. 3850

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.1("E" - Flag - WTW.

MiniRobotLanguage (MRL)

"E" - Flag "Exclude window"

Search Top Window

Intention

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you get the window out the "Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

Example

```
' Use Exclude List

' Add window to exclude list
STW.tE|TestAP
MMV.
VAR.$$HWN=#hwn#

' Remove window from exclude List
STW.I|$$HWN
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' If you comment the STW. above the window can not be found
here:
STW.t|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List availabe. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "I" - Flag - STW. 3832
- "K" - Flag - STW. 3836
- "h" - Flag - STW. 3861

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.1."F" - Flag - WTW.

"F" - Flag "center window on desktop" Previous 3825 Top 41 Next 3829

MiniRobotLanguage (MRL)

"F" - Flag "Center on Desktop"

Move Window to center of primary desktop

Intention

If the "F"-Flag is given, the window will be move to the center of the primary desktop. The "F"-Flag will override the default windows treatment, which would move the window to 0,0 if its outside of the desktop.
The "F"-Flag will always move the window to the center, no matter where the window was before.

```
STW.Fc|WindowClass
MMV.
```

Example

```
'
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.cF|Myclass
' Move Mouse to Center of window
' in center of desktop
MMV.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.17 "f" - Flag - WTW.

"f" - Flag "disabled windows" Previous 3855 Top 41 Next 3859

MiniRobotLanguage (MRL)

"f" - Flag "disabled windows"

Search Top Window

Intention

To ignore disabled windows is one of the default settings of the STW. command. Therefore if you want to search for a window that has been disabled, you will need the "f"-flag.

The "f"-Flag will override the default behavior. It will make STW. to ignore enabled windows and just look for disabled windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for invisible windows. If you do not want that, just add a prototype "v" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "f" - flag together with the "a" to search only windows that are disabled. No matter if they are visible or not.

Lets take a look, on this. This will search only for disabled windows.

```
STW.cf|WindowClass
MMV.
```

Now we would like to search for windows which are enabled, as well for disabled windows. We can do that in two ways.The first way is like this:

```
STW.cef|WindowClass
MMV.
```

The robot understands that we want enabled and disabled windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cav|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the invisible windows. We do this by adding the "v"-Flag after the "a". In this case the "v"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Start an Application only, if that Application is not yet
running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Next we disable the whole window
SWS.d
' Wait half of a second
PAU.0.5
' Try to locate it using "e"
' Will work, we get the window
STW.wtd|0|BCS - Testform

ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - STW. 3855
- "u" - Flag - STW. 3878
- "v" - Flag - STW. 3880
- "a" - Flag - STW. 3850

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.13 "g" - Flag - WTW.

MiniRobotLanguage (MRL)

"g" - Flag "Goto if found"

Goto if found

Intention

If the window you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to jump to a Label in that case.
And you can use the "X"-Flag to tell the robot to just continue with the script in case of error - if the window is not found.
This would change the behavior of the command, but sometimes this can be a good solution. Take this example:

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - goto Label "GoOn" if the window was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.
```

In case you use the "g"-Flag, together with the "W" -Flag ("on Found Gosub"), the "W"-Flag will be ignored. The "g"-Flag has a generally higher priority.

Example

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - gotoLabel "Sub" if thewindow was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- "o" - Flag - STW. 3867
- "V" - Flag - STW. 3844
- "W" - Flag - STW. 3846
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.14"G" - Flag - WTW.

"G" - Flag "don't move window" Previous 3827 Top 41 Next 3830

MiniRobotLanguage (MRL)

"G" - Flag "don't move window"

don't move the Top Window

Intention

This flag prevents the default window treatment from moving the window around. Without this Flag, a Top-Window that is outside the working area, will be move to position 0,0, that is the upper left corner of the desktop.

Having the "G"-Flag set, the window will just not be moved, no matter if its outside the working area or not.

```
STW.cG|WindowClass
MMV.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.cG|#32768

ENR.
```

Remarks

Please note that other prototypes may influence the power of the "G"-Flag, especially, if the are right of the "G"-Flag which means they have a higher priority.

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.1!"H" - Flag - WTW.

"H" - Flag "stay in primary desktop" Previous 3829 Top 41 Next 3832

MiniRobotLanguage (MRL)

"H" - Flag "primary desktop"

Search Top Window

Intention

This flag will move a window to position 0,0, that is the upper left corner, if the window is full or partial outside the primary desktop workarea.

It will override the default window treatment.
While by default also a Dual-Screen Desktop will be accepted as workarea, this will not be the case here.

The "H"-Flag makes sure the window is complete on the primary desktop.

```
STW.cH|WindowClass
MMV.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Click on the OK Button in a messagebox
' Locate the Application (Top-)Window
STW.tH|word
MMV.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.1("h" - Flag - WTW.

MiniRobotLanguage (MRL)

"h" - Flag "handle"

Search Top Window by handle

Intention

This flag is an exception. In fact it must not search the window, it will just localize the window "by handle". Of course this is a very efficient solution. Therefore if you have already localized your window, keep the handle in a variable and just feed it back to STW. when you need that window again.

```
' Localize by handle
STW.cta|ieframe|Update
' We grab the handle in a variable
HTV.$$MYW

' Later we can always get definitely this window,
' without further parameters
STW.h|$$MYW
MMV.

' Some more examples
STW.h|52367
MMV.

STW.h|#hnw#
MMV.
```

Example

```
'
' Localize by handle
STW.cta|ieframe|Update
' We grab the handle in a variable
: $$MYW=#hwn#

' Later we can always get definitely this window,
' without further parameters
STW.h|$$MYW
MMV.

ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

See also:

"E" - Flag - STW. 3825

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.17 "I" - Flag - WTW.

"I" - Flag "remove from Exclude List" Previous 3830 Top 41 Next 3834

MiniRobotLanguage (MRL)

"I" - Flag "remove from Exclude List"

Search Top Window

Intention

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you get the window out the "Exclude List"?
This is what the "I"-Flag is good for.

```
' Remove window from exclude List
STW.I|$$HWN
```

The "I"-Flag however need a window handle as parameter and will not work in any other constellation. Therefore keep the handles of those excluded windows, if you want to have them back later.

There is another way, how to search windows which are currently excluded from search?
You can also use the "K"-Flag.

```
' clear whole exclude List
STW.Kc|WindowClass
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list.

After this all previously "exclude listed" windows can be found again.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
' Use Exclude List
' Add window to exclude list
STW.tE|TestApp
VAR.$$HWN=#hwn#

' Remove window from exclude List
STW.I|$$HWN

' If you comment the STW. above the window can not be found
here:
STW.t|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag. To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List availabe. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "E" - Flag - STW. 3825
- "K" - Flag - STW. 3836

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.18 "J" - Flag - WTW.

MiniRobotLanguage (MRL)

"J" - Flag "window to front"

Search Top Window

Intention

The window comes to front already with the default window treatment, without any additional flags. But you can disable that, and select yourself exactly what you want. Like this:

```
STW.AJc|WindowClass
MMV.
```

The "A" Flag will switch of all default window treatment. Then comes the "J"-Flag and tells the robot to bring the window to front. Be careful with the operator priority. If you place the prototypes like this:

```
STW.JAc|WindowClass
MMV.
```

then the "A"-Flag will switch of the "J"-Flag because it is more right and therefore has a higher priority.

The "J"-Flag itself will not disable the default window treatment. Therefore something like:

```
STW.Jc|WindowClass
MMV.
```

```
is possibly a waste of letters, as the default behavior of
STW. is already to bring the window to front.

Note that the Flag "B" does not switch off the "J"-Flag,
therefore when using "B", the sequence of the prototypes "B"
and "J" should not be a issue.
```

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.AJc|#32768
' Locate the button
MMV.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX. Compare with this:
STW.Ac|#32768
' Locate the button
MMV.
ENR.
```

Remarks

Limitations:

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.19 "j" - Flag - WTW.

MiniRobotLanguage (MRL)

"j" - Flag "WII. by number"

Wait for Input Idle and tell how long

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time must be given as a numeric value or variable.

Note the value you specify will replace the default internal value for "WII. without parameter", used for the "k" - Flag - STW. 3865 . But also for WII. if no parameters are given.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions are allowed. For example:

```
' WIll wait no longer then a half second
STW.cj|WindowClass|0.5
MMV.
```

Technically the "j"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement. Just that it is integrated and does not need a separate line of code.

Example

```
' j - wait 5 Seconds maximum
STW.cj|WindowClass|5
MMV.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

- WII. - Wait-for-Input-Idle 2909
- "k" - Flag - STW. 3865

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2("k" - Flag - WTW.

MiniRobotLanguage (MRL)

"k" - Flag "WII. default"

Search Top Window

Intention

"Wait for Input Idle" halts script execution until the localized window is ready to process our Input.

This Command can be seen as a dynamic Pause-Statement, it waits only if our target window is busy. The maximum waiting time is taken from an internal value that is been set with the last WII. statement or with the "j" - Flag - STW. 3863.

Unlike a fixed pause statement which would always wait for a fixed time. `WII.` is the better solution.

If the process is a console application or does not have a message queue, "Wait For Input Idle" returns immediately.

The given time is in seconds. Fractions of a second are allowed.

```
STW.cj|WindowClass|5
MMV.
' Will do the same without repeating the value
STW.ck|WindowClass
MMV.
```

Technically the "k"-Flag does the same like the WII. - Wait-for-Input-Idle 2909 statement without a parameter. Just that it is integrated and does not need a separate line of code.

Example

```
' Demonstrate the "j" and the "k" Flag
STW.cj|WindowClass|5
MMV.
' Will do the same without repeating the value
STW.ck|WindowClass
MMV.
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- "j" - Flag - STW. 3863
- WII. - Wait-for-Input-Idle 2909

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2."K" - Flag - WTW.

"K" - Flag "Clear Exclude List and search" Previous 3834 Top 41 Next 3838

MiniRobotLanguage (MRL)

"K" - Flag "Clear Exclude List"

Clear Exclude List and search

Intention

Immediately clear Exclude-List before specified search.

Imagine you have several equal windows, which you want to handle one after the other.
Yes, there are features available to do that, for example the "r"-Flag (Window-number). Or your could use FEW.

But how about excluding those windows which you have already handled from further window searches?

To understand this you need to know that there is a Window Search "Exclude-List". Now if you add a window to this "Exclude List", it will not be found any more. Not with STW. and not with ITW. It will be there but it will be ignored.

How do you add a window to the "Exclude List"?

```
STW.cE|WindowClass
MMV.
```

And how do you search windows which are currently excluded from search?
This is what the "K"-Flag is good for.

```
' clear whole exclude List
STW.Kc|WindowClass
```

The "K"-Flag will immediately, even before the search starts, clear the exclude list. No matter if a window is found or not, using the prototype "K" will clear the exclude list.

After this all previously "exclude listed" windows can be found again.

Example

```
' Use Exclude List

' Add window to exclude list
STW.Et|TestApp
MMV.
VAR.$$HWN=#hwn#

' Remove all windows from exclude List
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' and search

' If you remove the "K" below, the window can not be found!
STW.Kt|TestApp

ENR.
```

Remarks

Windows which have been closed, will automatically be removed from the exclude list. There is no need to remove them manually from the exclude list.
If anyway you want to remove a window from the exclude list, you can use the "I"-Flag.
To completely clear the Exclude list, use the "K"-Flag.

When locating Child-Windows using SCW. there is also Exclude-List available. The Exclude List for Child-Windows is a separate list and will therefore survive if you clear the exclude list for Top-Windows.

Currently you can not check which windows are exclude-listed. There is also no way of showing the content of an exclude-list.

Limitations:

There are no limitations about the number of windows you may add to the exclude list.

See also:

- "I" - Flag - STW. 3832
- "E" - Flag - STW. 3825

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2 "M" - Flag - WTW.

MiniRobotLanguage (MRL)

"M" - Flag "TopMost"

Search Top Window

Intention

In windows, there is a property of Top-Windows that will make them stay always in front of all other windows.

This property is called "TopMost". A window that is "TopMost" is always on top of all other windows, that are not TopMost.

Only another TopMost window can be in front of an TopMost window.

For example the MR-Editor can be set to TopMost.

Now you may think "this sounds good!". Be warned. This is an advanced feature for advanced users.

Programs have Messageboxes and also open up other windows. In case they have been made TopMost, you may not be able to see those Messages or PopUp windows. This may lead for you to think that the program hangs.

Why don't you see those messages?
Because they will not be TopMost and therefore they will appear behind the TopMost window!

Therefore use TopMost style only when really needed and when you know what you are doing.

Using the Flag "M", the window will be made TopMost immediately after it was localized.

```
STW.cM|WindowClass
MMV.
```

Use the "N"-Flag to remove the TopMost-property from a TopMost window.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.Mc|#32768
VAR.$$HWN=#hwn#
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Remove the TopMost Property
STW.hN|$$HWN

ENR.
```

Remarks

Limitations:

See also:

"N" - Flag - STW. 3840

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2 "N" - Flag - WTW.

MiniRobotLanguage (MRL)

"N" - Flag "No TopMost"

No more TopMost

Intention

In windows, there is a property of Top-Windows that will make them stay always in front of all other windows.

This property is called "TopMost".

A window that is "TopMost" is always on top of all other windows, that are not TopMost.

Only another TopMost window can be in front of an TopMost window.

For example the MR-Editor can be set to TopMost.

Now you may think "this sounds good!". Be warned. This is an advanced feature for advanced users.

Programs have Messageboxes and also open up other windows. In case they have been made TopMost, you may not be able to see those Messages or PopUp windows. This may lead for you to think that the program hangs.

Why don't you see those messages?
Because they will not be TopMost and therefore they will appear behind the TopMost window!

Therefore use TopMost style only when really needed and when you know what you are doing.

Using the Flag "M", the window will be made TopMost immediately after it was localized.

```
STW.cM|WindowClass
MMV.
```

Use the "N"-Flag to remove the TopMost-property from a TopMost window.

Example

```
'#EXE:?pfad\
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'#SPI:ForceWrite
'
' Locate the Application (Top-)Window
STW.Mc|#32768
VAR.$$HWN=#hwn#

' Remove the TopMost Property
STW.hN|$$HWN

ENR.
```

Remarks

Limitations:

See also:

"M" - Flag - STW.

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2 "o" - Flag - WTW.

MiniRobotLanguage (MRL)

"o" - Flag "on not found Goto"

Search Top Window

Intention

If the window you are looking for is not found within the specified waiting time, normally you will end up with an error-message.

Maybe you want just to jump at another place in the script the, and start a program. Or react in another way. That is where the "o" Flag enters the game. The script just continues. The window is not found and we tell the robot to jump to a Label in that case.

Take this example:

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
STW.toXw|BCS|GoOn|0
GTO.weit

:GoOn
 EXE.?exepfad\Sample Apps\BSC-Testform.exe

:weit

MBX.Ready
ENR.
```

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Example

```
' Run application only if the window is not found
'
' t - text (caption of the window)
' o - goto Label "GoOn" if the window was NOT found
' X - continue the script as normal, without error,
'     if the window is not found
' w - set waiting time to 0 seconds
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
STW.toXw|BCS|GoOn|0
GTO.weit

:GoOn
 EXE.?exepfad\Sample Apps\BSC-Testform.exe

:weit

MBX.Ready
ENR.
```

Remarks

In case you use the "o-Flag, together with the "V" -Flag ("on not Found Gosub"), the "V"-Flag will be ignored. The "o"-Flag has a generally higher priority.

Limitations:

-

See also:

- "g" - Flag - STW. 3859
- "V" - Flag - STW. 3844
- "W" - Flag - STW. 3846
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2 "p" - Flag - WTW.

MiniRobotLanguage (MRL)

"p" - Flag "Process ID"

Search Top Window

Intention

You can localize a window safely, if you have the PID (Process ID) of the application behind that window.

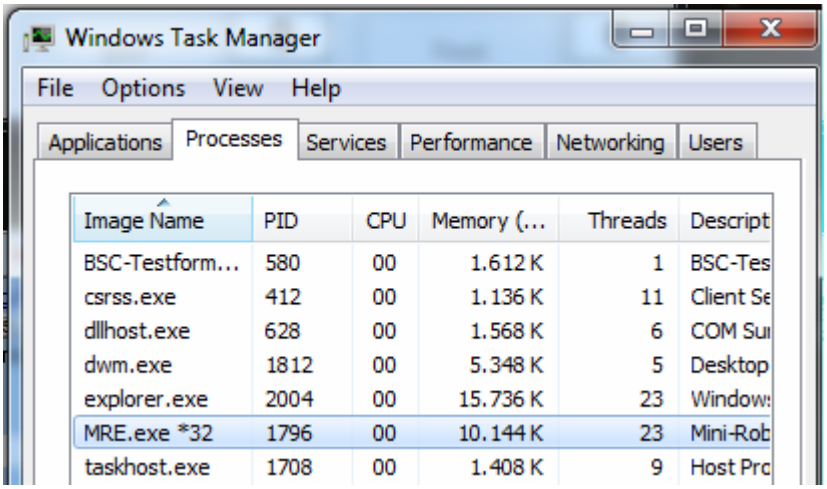
Be sure that you have read and understood the chapter 1.7. Application and PID 137.

Hint: If you have localized a window, you also have the PID of the process behind that window in `#pid#`. You can save it for later use in a variable. Like this:

```
STW.c|pbwind
: §§PID=#pid#

or use:

STW.c|pbwind
HTV.§§PID
```

In both cases, you get the PID of that process into the local variable §§PID 366.

You can see the PID of a process in the task-manager::



The PID value is a normal search parameter and can be used alone or together with other search parameters.

```
' Search a window from process with ID 3435
' and the given window class.
'
STW.ch|pbwind|580
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MMV.
```

Example

```
' Locate window only by PID
STW.p|580
MMV.

ENR.
```

Remarks

Limitations:

See also:

- 1.7. Application and PID 137
- GDF. - Get-Data-From 2889
- 1.6.1. Program Flow Control 121
- GPI. - Get Process Information 2893
- IEP. / NEP. - If-Exists-Process 2900
- FEP. - For-Each-Process 932
- WPR. - Wait-Process-Running 2911
- WPT. - Wait-Process-Termination 2913
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.2("q" - Flag - WTW.

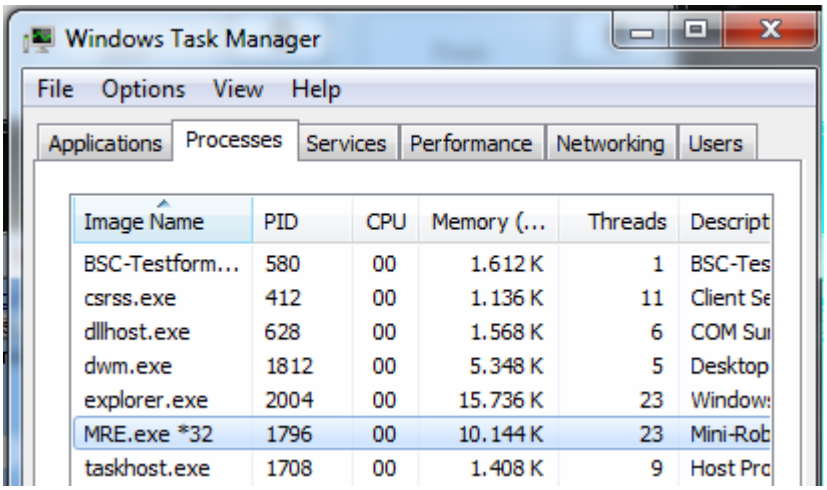
MiniRobotLanguage (MRL)

"q" - Flag "Thread ID"

Search Top Window

Intention

This is an advanced parameter. The chances are that you will never need it.
Processes have a number of Threads. See this picture here:



On the left side you can see the name of the process. Right of it the PID of the process. Later you see "Threads".
In a very simplified form, a Thread is a subprogram of that process, that runs an own small program. Generally the more threads a program has, the better it will scale up on multiple CPU-Core's.
While all Threads of a process belong to that process and therefore have the same PID, they have a different Thread-ID. In case you may need to separate between windows of the same process, which is "Multi-threaded", the "q"-flag is for you.

```
STW.cq|WindowClass|109872
MMV.
```

Hint: If you have localized a window, you also have the TID of the process behind that window in `#tid#`. You can save it for later use in a variable. Like this:

```
STW.c|pbwind
: §§TID=#tid#
```

This way, you get the TID of that Thread into the local variable §§PID 366.

If you don't "understand a word", or don't have very special cases, just ignore this flag.

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe
' Locate the window
STW.p|#pid#
' Show the Thread-ID
MBX.#tid#
ENR..

ENR.
```

Remarks

Limitations:

See also:

- 1.7. Application and PID 137
- "p" - Flag - STW. 3869

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.27 "Q" - Flag - WTW.

MiniRobotLanguage (MRL)

"Q" - Flag "Repeat Prototype"

Search Top Window

Intention

This flag will just repeat the Prototype that was last used with STW. or ITW. or any otehr Top-Window search instruction.

```
STW.c|WindowClass
MMV.
' this will do the same
STW.Q|WindowClass
MMV.
```

Note that there is another way of repeating the last STW. or ITW.. That is, to just leave the whole parameter part empty.

```
STW.c|WindowClass
MMV.
' this will do the same
STW.
MMV.
```

While "Q" will only retype the prototypes internally. Leaving the parameters away, will retype the whole parameters, including the prototypes, using those from last time the command was called.

Example

```
'#EXE:?pfad\
'#SPI:ForceWrite
'
STW.c|WindowClass
MMV.
' this will do the same
STW.Q|WindowClass
MMV.
ENR.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

`STW.` and `ITW.` share the same parameter store. Therefore if you have localized the window with ITW. then you can also use these parameters for STW. and vice versa.

Limitations:

-

See also:

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.83.30.2 "r" - Flag - WTW.

MiniRobotLanguage (MRL)

"r" - Flag "result number"

Search Top Window

Intention

There may be multiple results in a window-search with `STW.`. Normally you get the first result that fits to your search parameters.
Using the "r"-Flag you can define which result you want, Nr.1, Nr.2 or Nr. 20. Just add the "r"-Flag and the result number.

If omitted, always result Nr.1 is returned.

Note: If there are not enough results, your search will fail!
Therefore you can also use this feature to count windows.

```
'We'll get the 3rd result with that window class
STW.cr|WindowClass|3
MMV.
```

Example

```
'We'll get the 4th result
' with that window class
'
STW.cr|WindowClass|4
MMV.
ENR.
```

Remarks

-

Limitations:

-

See also:

-

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.29 "t" - Flag - WTW.

MiniRobotLanguage (MRL)

"t" - Flag "text"

Search Top Window

Intention

This flag is one of the most important search parameters. It is the window-text. Most often you can even see the text of a top-window without using the windows-Inspector.
Here is the simplest case how you can grab a window.

```
STW.t|TestApp
MMV.
```

The text parameter is a string-value and therefore supports 3.2 Standard-Search Pattern 341.
Let us try some of these.

As always with Search patterns, it is enough to give a unique part of the value. By default the text is not case sensitive. You can change that behavior using the search patterns. Take the sample-script below and replace the line:

```
' Locate the window
STW.t|Testf
```

with the following lines. This will help you to understand the use of Patterns for text search.

```
' Lets start with this
STW.t|{&DOCAPS:Testf}
```

This will force the text to be case-sensitive! Try this:

```
' This will not find the window
' because of case-sensitivity!
STW.t|{&DOCAPS:testf}
```

It does not find the window because the &DOCAPS: switches case-sensitivity on.
Now how about this:

```
STW.t|{&EXACT:bcs - Testform}
```

It does find the window. But then whats the "&EXACT:" good for?
Try this:

```
STW.t|{&EXACT:bcs - Testf}
```


© 2023 Theo Gottwald * https://www.smart-package.com

It does not find the window because the &EXACT: switches off the default behavior that a part of the text is enough. CAn it be combined with the &DOCAPS: ?

We try. This should work:
```
STW.t|{&EXACT:&DOCAPS:BCS - Testform}
```
and this should not work:
```
STW.t|{&EXACT:&DOCAPS:bCS - Testform}
```

Yes, everything work as expected.

Do you know the old times, when you had simple Wildcards as patterns? The"*****" and the "**?**" ?

You can have them back if you want. Try this:

```
STW.t|{&WILDC:b?? - Testform}
```

Yes, it works. And how about:

```
STW.t|{&WILDC:b?? - Tes*orm}
or
STW.t|{&WILDC:b?? - *s*or?}
```

It works.
We kept them for you.
You can still use even the old DOS-Patterns if you really want.
But there is much more if you use the new patterns.

How about a fuzzylogic and a OR'ed combination of two patterns?

```
STW.t|{&WILDC:b?? }&OR:{BCS-tesform&FUZZY:6}
```

and you can not only combine two patterns, you can combine as many patterns using &OR: as you like! If you want to support many languages, this is the way to go.

No matter if there is a "YES" on that button or a "Qui" or whatever. So long you know it, or at least a bit of it, you can locate it using Search patterns.

Example

```
' Close all such windows if there are some open
CAW.PBWindowClass:0|BCS - Testform

EXE.?exepfad\Sample Apps\BSC-Testform.exe

' Locate the window
STW.t|Testf

' Close the window
CAW.PBWindowClass:0|BCS - Testform
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

Limitations:

See also:

- 3.2 Standard-Search Pattern 341
- "c" - Flag - STW. 3854

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.3("u" - Flag - WTW.

MiniRobotLanguage (MRL)

"u" - Flag "invisible"

Search Top Window

Intention

To ignore invisible windows is one of the default settings of the STW. command. Therefore if you want to search for a window that has is invisible, you will need the "u"-flag.

The "u"-Flag will override the default behavior. It will make STW. to ignore visible windows and look for invisible windows, only.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also enable the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "u" - flag together with the "a" to search only windows that are invisible. No matter if they are enabled or not.

Lets take a look, on this. This will search only for invisible windows.

```
STW.cu|WindowClass
MMV.
```

Now we would like to search for windows which are visible, as well as for invisible windows. We can do that in two ways.The first way is like this:

```
STW.cuv|WindowClass
MMV.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cae|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
CAW.PBWindowClass:0|BCS - Testform
'
' Start an Application only,
' if that Application is not yet running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

STW.p|#pid#
' Next we make the window invisible
SWS.i
' Wait half of a second
PAU.2.5
' We locate it even though its invisible
STW.wtu|1|BCS - Testform
' and make it visible again
SWS.v
ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - STW. 3855
- "f" - Flag - STW. 3857
- "v" - Flag - STW. 3880
- "a" - Flag - STW. 3850
- 1.8. Selecting the Application (Topwindow) 150

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.3."V" - Flag - WTW.

"V" - Flag "GoSub if not found. " Previous 3842 Top 41 Next 3846

MiniRobotLanguage (MRL)

"V" - Flag "GoSub if not found"

Gosub if not found

Intention

If the window you are looking for is not found within the specified waiting time, several things can happen. Normally you will get an error-message.

But that does not necessarily be what you want. Maybe you want just to see if the window is there and if not start the program in a Subroutine.

```
' "V" - Flag - on not found Gosub
STW.ctVw|PBWindowClass:0|BCS|Sub|1
' Move Mouse to the Install Button
SCW.ct|butt|Install
MMV.
ENR.
'-------------------------------------------------------------
' This Subprogram will be called if the window is not found
:Sub
EXE.?exepfad\Sample Apps\BSC-Testform.exe
STW.ct|PBWindowClass:0|BCS
RET.
```

The script above will try to locate the "BCS - Testform" and if its not yet started, it will be started and localized.

Example

```
STW.ctVw|PBWindowClass:0|TestAppx|Sub|1
MBX.Ready
ENR.
'-------------------------------------------------------------
:Sub
MBX.Error - Program not found
RET.
```

Remarks

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

You can not mix the "V" and the "W"-Flag. If both are found, the "W"-Flag takes priority

See also:

- "g" - Flag - STW. 3859
- "o" - Flag - STW. 3867
- "W" - Flag - STW. 3846
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.3 "v" - Flag - WTW.

MiniRobotLanguage (MRL)

"v" - Flag "visible"

Search Top Window

Intention

To ignore invisible windows is one of the default settings of the STW. command. Therefore you will not need to specify the "v"-Flag often.

It can however make sense, to override the default behavior by using the "u"-flag which will make STW. to visible windows and just look for invisible windows.

If you want to catch both, the "a"-flag is what you need.

However the "a"->Flag will automatically also switch on the search for disabled windows. If you do not want that, just add a prototype "e" after the "a". As the prototype on the right has a higher priority, you get exactly what you want this way.

In the same way, you can use the "v" - flag together with the "a" to search only windows that are visible. No matter if they are enabled or not.

Lets take a look, on how to do this. This will search only for visible windows.

```
STW.cv|WindowClass
MMV.
```

Since this is a default behavior of `STW.`, we have wasted the "v" at this place. Its not needed.

Now we would like to search for windows which are visible, as well for invisibble windows. We can do that in two ways.The first way is like this:

```
STW.cuv|WindowClass
MMV.
```

The robot understands that we want visible and invisible windows. If we just give all two prototypes. But there is another way to do it. We can do this:

```
STW.cae|WindowClass
MMV.
```

The "a" Flag says "search all windows", enabled, disabled, visible and invisible. Now we just need to tell the robot that we do not want to get the disabled windows. We do this by adding the "e"-Flag after the "a". In this case the "e"-Flag will override the "a"-Flag. And tell the robot "we just want visible windows.".

© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
CAW.PBWindowClass:0|BCS - Testform
'
' Start an Application only,
' if that Application is not yet running.
'
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

STW.p|#pid#
' Next we make the window invisible
SWS.i
' Wait half of a second
PAU.2.5
' We locate it even though its invisible
STW.wtu|1|BCS - Testform
' and make it visible again
SWS.v
ENR.
```

Remarks

Limitations:

See also:

- "e" - Flag - STW. 3855
- "f" - Flag - STW. 3857
- "u" - Flag - STW. 3878
- "a" - Flag - STW. 3850
- 1.8. Selecting the Application (Topwindow) 150

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.3 "W" - Flag - WTW.

"W" - Flag "GoSub if found" Previous 3844 Top 41 Next 3848

MiniRobotLanguage (MRL)

"W" - Flag "GoSub if found"

Search Top Window

Intention

If the window you are looking for is found within the specified waiting time, normally the script just continues. You can however ask the robot to first call a Subprogram in that case and then continue with the script. This is done using the "W"-Flag.

```
STW.ctW|PBWindowClass:0|BCS|Sub
MMV.
ENR.
:Sub
MBX.!
RET.
```

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Example

```
'
' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

' If found close that window.
STW.ctWw|PBWindowClass:0|BCS|Sub|1

' And then open a new instance
EXE.?exepfad\Sample Apps\BSC-Testform.exe

STW.ct|PBWindowClass:0|BCS
SCW.ct|butt|Install
MMV.

ENR.
'-------------------------------------------------------------
:Sub
CAW.PBWindowClass:0|BCS
' Wait until its gone
MTW.
RET.
```

Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

The "W"-Flag can be used together with the "V"-Flag, while only one of these will be called. In case you use the "g"-Flag "on found Goto", the "W"-Flag will be ignored.

Limitations:

-

See also:

- "g" - Flag - STW. 3859
- "o" - Flag - STW. 3867
- "V" - Flag - STW. 3844
- "X" - Flag - STW. 3848

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.34 "w" - Flag - WTW.

MiniRobotLanguage (MRL)

"w" - Flag "wait time"

Search Top Window

Intention

How long do we wait if a window will appear or not?
Let's try it.

```
: $$TIM=#dtime#
STW.tX|nix
: $$TIM=#dsince#
MBX.$$TIM
ENR.
```

This i what we get.

It's currently about 15 seconds. When you read this it can be another value.
In fact this is the default time that we wait for a window before we run into an error-message or do something else (like in this script).

Now, its very important to understand that we do not generally wait 15 seconds, and then look if the window has appeared.
What we do is, we try ... try ... and try and check permanently, if the window has appeared. If the window appears any time, the script will immediately continue.

If the time is over, we do something else.

Now 15 seconds is enough for some cases, in many cases the window will appear after 45 seconds or any other time. Now this is where the "w"-flag enters the game.

Using the "w"-flag we can tell the robot how long he should wait. It can be 0 seconds like this:

```
STW.wtUX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found
```

or 360 seconds like here.

© 2023 Theo Gottwald * https://www.smart-package.com

```
STW.wtu|360|BCS - Testform
```

and there is no real limitation.

Example

Remarks

-

Limitations:

The maximum number for the waiting time is the equivalent for 23100 days. In short: you can not ask the robot to wait longer for a window then ~63 years. Anything below will be just fine. Note that negative waiting times will be rounded to zero.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

3.83.30.3!"X" - Flag - WTW.

"X" - Flag "run without error" Previous 3846 Top 41 Next 3850

MiniRobotLanguage (MRL)

"X" - Flag "run without error"

Search Top Window

Intention

If the window you are looking for is not found within the specified waiting time, several things can happen. Normally you will just get an error-message. If you do not want to get an error, but just have the script continue running, use the "X"-Flag.

You can use the `#tio#` 3.1 Systemvariables 334 to check if the window was found or not. Also you can use the "V","W","g" and "o" Flags to react inside the STW.

For example in such a script:

```
STW.cXg|WindowClass|found
MMV.

:found
MPO.
```

This script will invert the normal behavior of STW. It will continue running in case of the window is not found. And in case the window is found, it will jump to the Label "found".

Example

```
'-------------------------------------------------------
' "X"-Flag Sample 1
' Run only if its not yet running ...
'-------------------------------------------------------
'
' Run application only if the window is not found
'
' t - text (caption of the window)
' g - gotoLabel "Sub" if thewindow was found
' X - continue the script as normal, without error,
'     if the window is not found
STW.tgXw|BCS|GoOn|0
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:GoOn
MBX.Ready
ENR.


'-------------------------------------------------------
' "X"-Flag Sample 2
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------
'#EXE:?pfad\
'#SPI:ForceWrite
'
' Start an Application only, if that Application is not yet
running.
'
' We use STW here to test if the Application is already
running
' in that case we jump over the EXE-command and contine.
' We use the following flags:
' w ... 0 - Set waiting time to 0 seconds
' t ... "BCS - Testform" - this is the caption text of the
window we search for
' g ... "found" - jump to Label ":found" if the window already
exists
' X ... continue without errormessage if the window was not
found in time
'
STW.wtgX|0|BCS - Testform|found
EXE.?exepfad\Sample Apps\BSC-Testform.exe
:found

' Now we can use the Application and press Buttons
STW.wt|0|BCS - Testform
SCW.nctb|1|Button|Next|0
' Compatible Button Click
MLC.

'-------------------------------------------------------------
ENR.
```

Remarks

Limitations:

See also:

- "g" - Flag - STW. 3859
- "o" - Flag - STW. 3867
- "V" - Flag - STW. 3844
- "W" - Flag - STW. 3846

© 2023 Theo Gottwald * https://www.smart-package.com

3.84 XML-Parser

3.84.1 ! Sample-XML Data

MiniRobotLanguage (MRL)

Sample XML-Data

XML-Data that is used for the following Help Pages

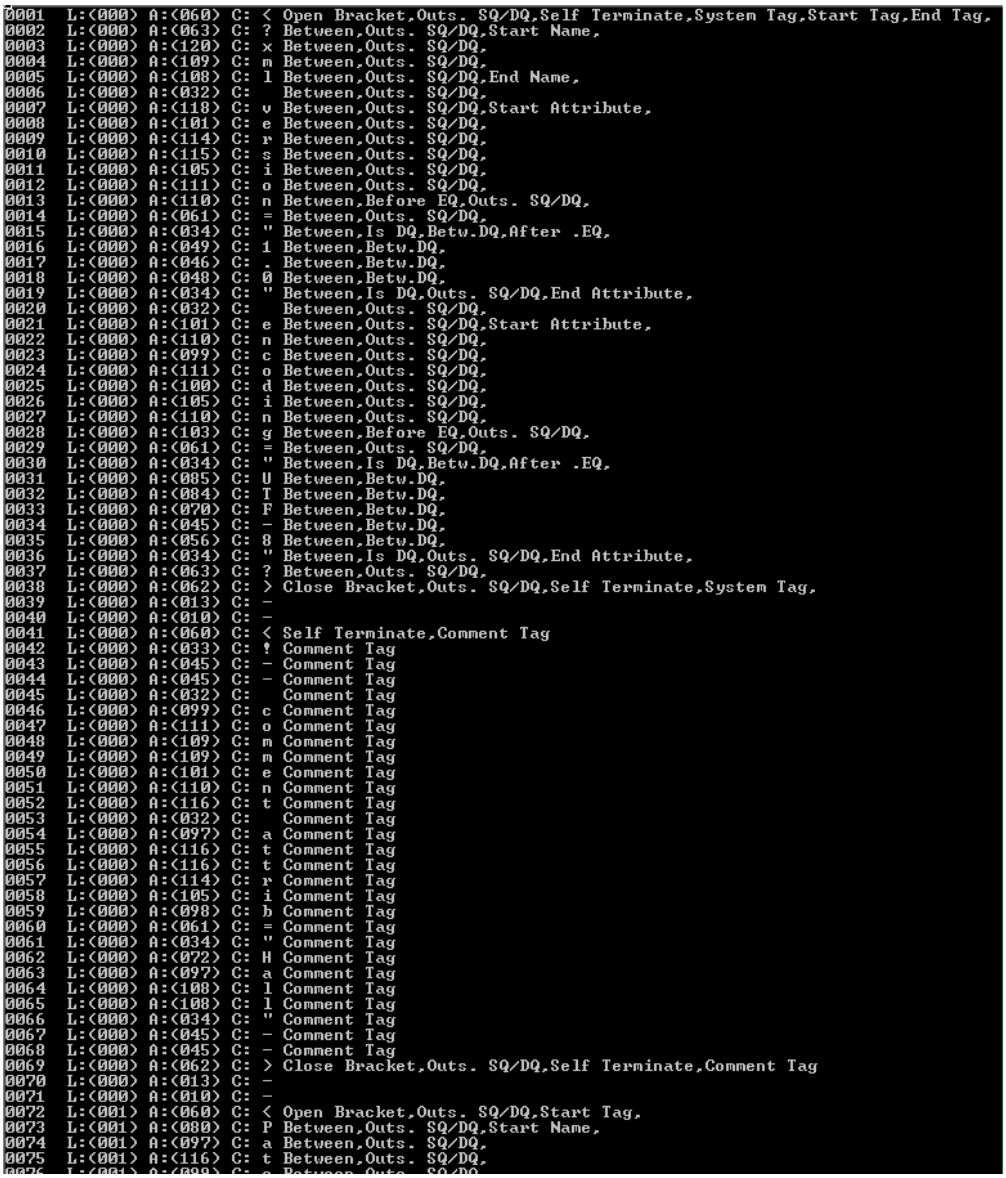
Intention

We will use this Sample XML-Data for all the commands on the following help pages.
Below you will find the sample-Data as well as a XML-Flags Dump.
This shows you how the Commands and queries work.

XML-Flags-Dump

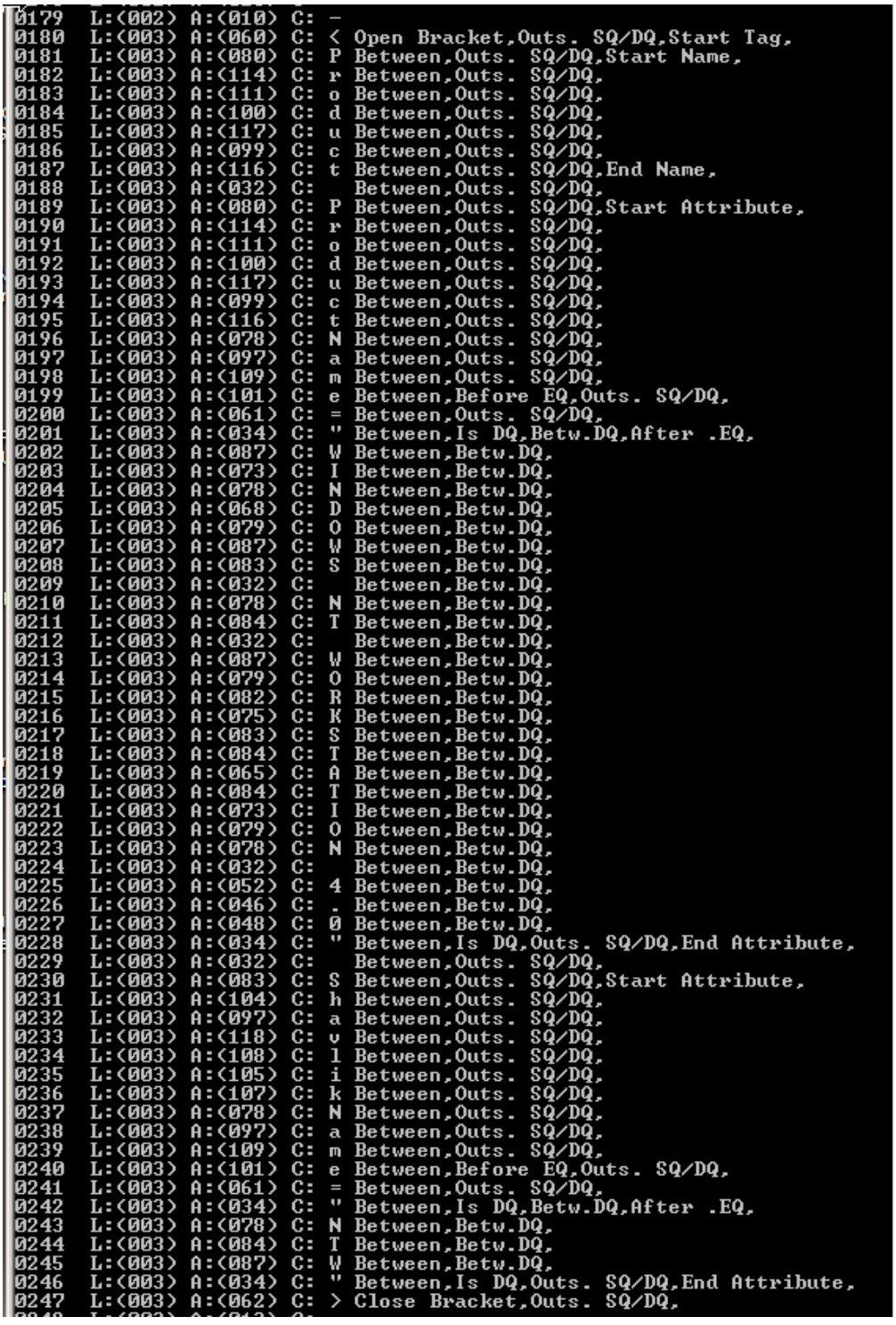
This is an XML-Flags Dump of the used XML-Data. You can see the Flags that is set for any of the Bytes in the XML-Data.

*© 2023 Theo Gottwald * https://www.smart-package.com*

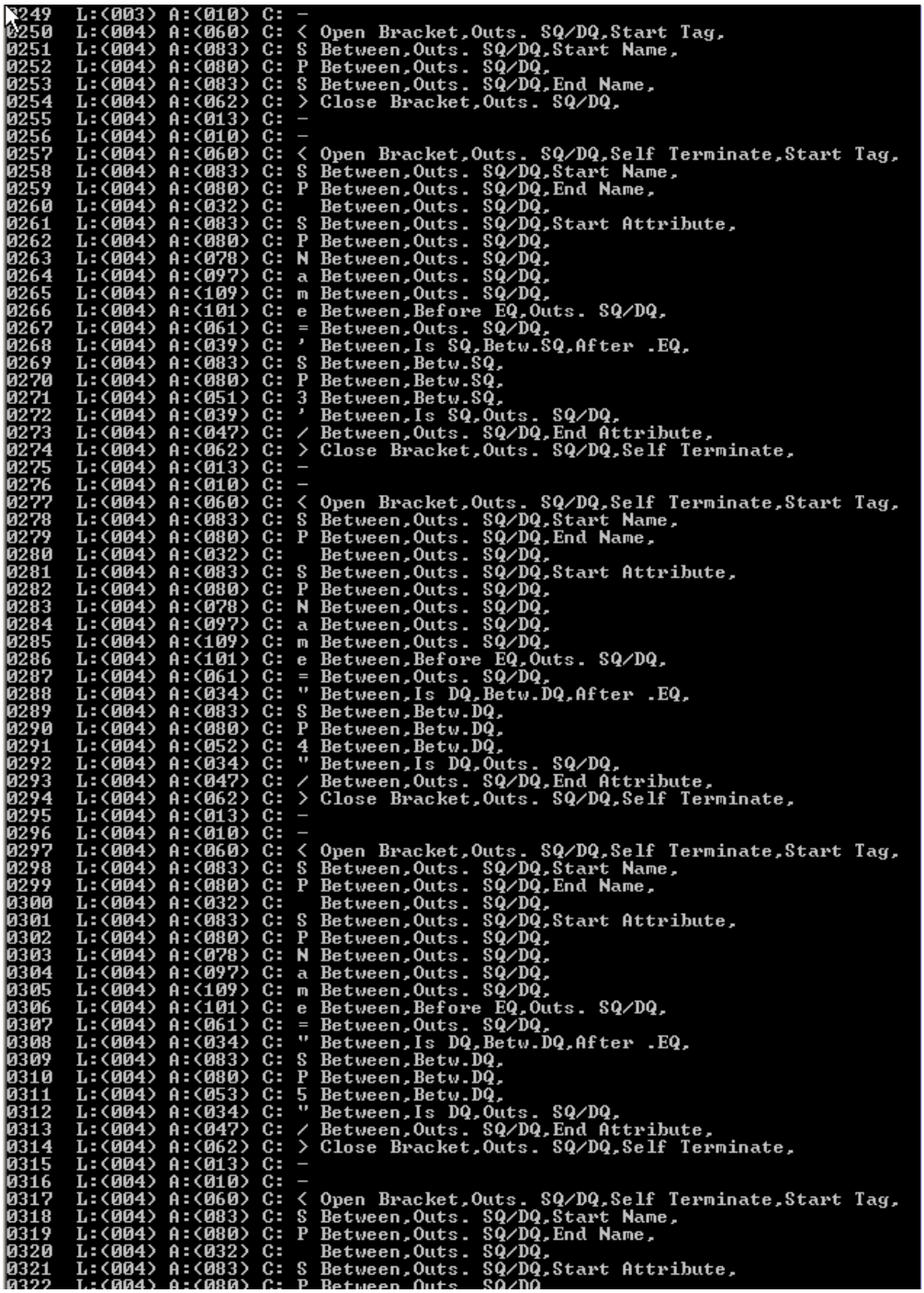


Here is another one:

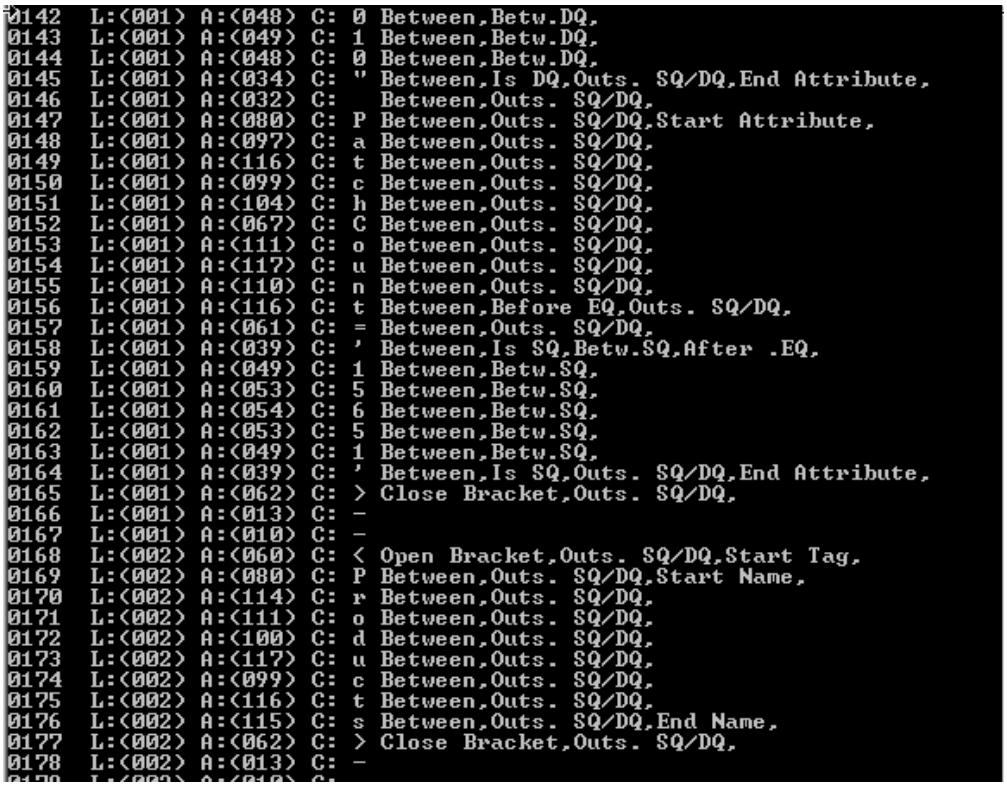
© 2023 Theo Gottwald * https://www.smart-package.com



© 2023 Theo Gottwald * https://www.smart-package.com



*© 2023 Theo Gottwald * https://www.smart-package.com*



In this above Dump you can see a part that is delimited by Single Quotes (SQ).
Below you can see the used XML-Data.

Example-XML-Data

```
'***********************************
' XML-Sample Data
' Used to demonstrate the XML-Features
' Mote that this XML-File has been changed
' for demonstration purposes
' and has no authentic technical information
' about Updates.
'***********************************

<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount='15651'>
<Products>
<Product ProductName="WINDOWS NT WORKSTATION 4.0"
ShavlikName="NTW">
<SPS>
<SP SPName='SP3'/>
<SP SPName="SP4"/>
<SP SPName="SP5"/>
<SP SPName="SP6"/>
<SP SPName="SP6A"/>
</SPS>
</Product>
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
<Patch PatchName="SQLServer2008-KB2716433-x86.exe"
SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
<SURL LGROUP="ENSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe"/>
<SURL LGROUP="FRSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe"/>
<SURL LGROUP="DESTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe"/>
<SURL LGROUP="CHHKSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-
FC1C-4A71-9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-
x86.exe"/>
</SURLS>
<AffectedProductSPs>
<AffectedProductSP ShavlikName="SQL2K8AS" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008STD" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008EXPRESSAS" SPName="SP2"
BulletinID="MS12-070"/>
</AffectedProductSPs>
</DL>
</Patch>
</Products>
</PatchInformation>
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973

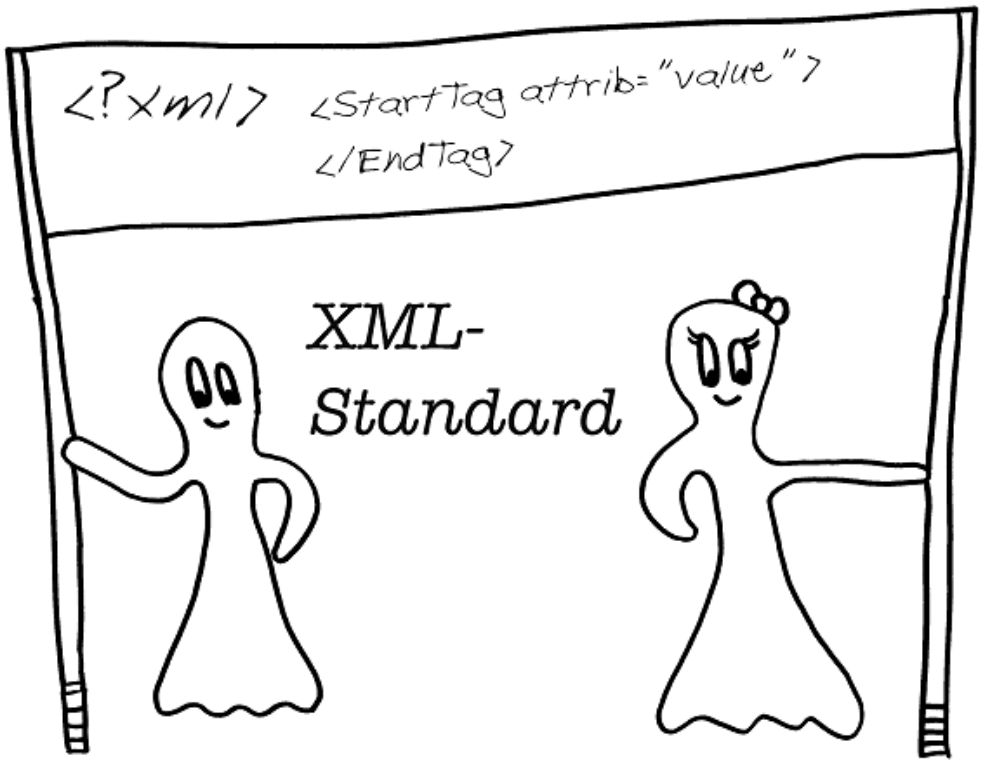
© 2023 Theo Gottwald * https://www.smart-package.com

- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.2 ! Smarty's XML-Features

Smarty's XML-Commands



What is XML good for?

XML stands for Extensible Markup Language. It was developed at the World Wide Web Consortium, by a group of people who wanted to improve on HTML and SGML. It was defined as a formal specification in February 1998.

Possibly you have XML Data files as output from other software.
Possibly you want to generate or analyze XML Data from whatever sources.

The question is "how can you (re-)act up on these data files?"

The answer is now implemented in the XML.-Commands.
These commands make it easy for you to go through these XML-Files.

What can you do with these XML-Commands?

- Parse the XML-File and search for Tags, Attributes or data in these files.
- Generate new XML Files and copy parts of the old files to these new files.

Here are some more Infos about XML.

A simple XML-Data could look like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  <to>Peter</to>
  <from>Ralph</from>
  <heading>Reminder</heading>
  <body>Don't forget making your skript!</body>
</note>
```

Clean and UnClean your Data

As you can see, we have data that is enclosed and described using "tags".
We have "Start-Tags" like `<note>`
and we have "End-Tags" like `</note>`
between these we have data that can itself be XML or other data.
Note that there are some characters that should not be in this data, because they are reserved for tags. These characters are:

& - should be replace by &
< - should be replace with <
> - should be replace with >
" - should be replace with "
' - should be replace with &spos;

For this purpose the `XML.` command has the "`Clean`"-Option.
It will prepare your data to be valid as XML-Data.
And if you want your data back to the original state, just use the "`unclean`" option.

Here is an example:

```
'**********************************
' XML.-Clean/UnClean Sample
'**********************************
'
$$XMF=My current <XML\> is &new
' Now we are going to parse it
XML.Clean|$$XMF|$$XMC
XML.UnClean|$$XMC|$$XMU
PRT.Original:    $$XMF
PRT.After CLean: $$XMC
PRT.--------------
PRT.UnCLeaned:   $$XMU
MBX.!
ENR.
```

Self terminated Tags

© 2023 Theo Gottwald * https://www.smart-package.com

Some XML-Tags do not have a closing tag because they are "Self-Terminated".
They look like this:

```
' This is "self terminated" tag.
<Name/>
```

The first tag, also known as "Root-Tag" is a system tag, because it starts with "<?".
System Tags are always self terminated. They do not have an explicit close tag.

```
' This is a system tag, its also "self terminated".
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Also comment tags are self terminated:

```
<!-- this is a comment tag -->
```

If you want to learn more about XML, see the www-resource below.

Here are a few important notes:

1. XML is not HTML, it has stricter format rules

2. XML Tags are Case Sensitive

3. All XML Elements Must Have a Closing Tag unless they are "self terminated".

4. You must correctly nest, your tag. This is *not* allowed:
   <B><I>Something</B></I>

5. It *must* be nested correctly, like this:
   <B><I><Something</I></B>

6. XML Documents Should Have a Root Element
   The built in parser however will take the first < he finds as the root-element.

7. Currently the XML-Parser is not an XML-Validator. It will instead try to do its best
   to understand what YOU want. Instead of complaining if you have not hold
   on all the rules of XML. You may however find error-messages in the process if you do not
   obey the rules. If problems appear, the XML-Commands will always set the "Timeout-Flag"
   and will store the error. You can get the stored errors description using:

```
XML.load|$$MYF
' Test the timeout-flag for errors
' and jump to a LAbel if there were problems
JIT.Lab_Error

Lab_Error:
' Get the Error-Message in to a variable
XML.get error|$$ERT
DBP.$$ERT
END.
```

Possibly you will get an additional error number that is easier to react up on using:

© 2023 Theo Gottwald * https://www.smart-package.com

```
XML.get error|$$ERT|$$ERN
```

8. White space and carriage returns are preserved

9. Attribute values of tags must be enclosed in either single or double quotes.
If you use a single quote, then double quotes are allowed in the value.
Vice versa is also true.

10. Non-tag data is expected to be free of <, >, and &. While the parser may still parse it properly,
this goes for attribute values as well. You should replace these like described above using the
supplied "clean" and "unclean" options..

Using the `CVF.` (Convert File) -Command, you can convert XML-Files to better readable Text-Files. These files can be displayed in any text-editor. For this most often, just replace all "><" with ">$crlf<".

Generally you could also work with XML files by just using the available String-Commands.
Using the XML-Commands makes things much easier.

In the following tutorials we will show you how it works, and why its much easier this way.

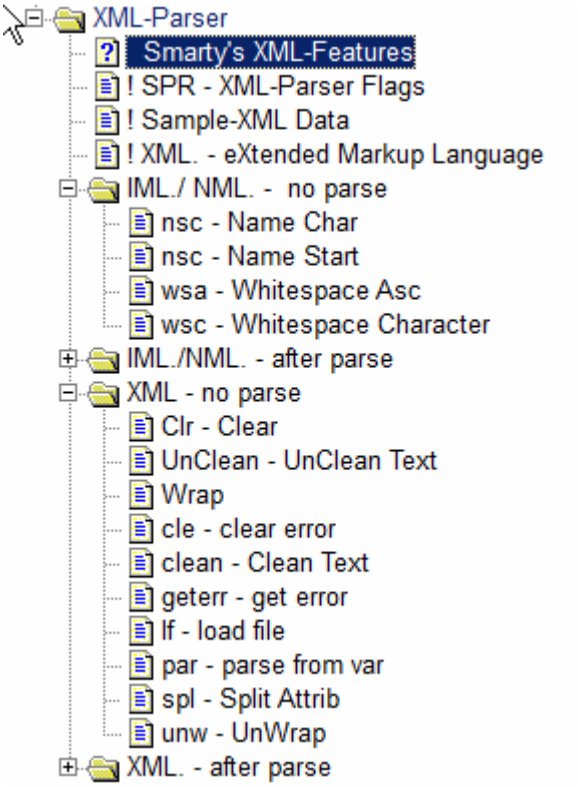
How does it work generally?

There are few commands that can always be used, even if no XML-File has been parsed. These are:

- Parse - parse a file that is inside a variable
- Load - load a file from file-system
- Clear - clear / delete the previous loaded XML File and all Flags.
- Clean Text - used to prepare a text as XML-Data
- Unclean Text - undo the "clean" command
- Get Error - retrieve the XML-Error - Status
- Clear Error - clear the XML-Error Status and all error flags and variables

also there are some `IML.`-commands that can be used on any text.
Currently these are (see under "no parse"):

© 2023 Theo Gottwald * https://www.smart-package.com



All the other `XML/IML.`-Commands require an XML-File or XML-Data to be loaded first.

At the process of loading/parsing, the XML-Parser Engine adds internal tags to any of the loaded characters. Loading an XML.-File is as simple as:

```
$$MYF=C:\filename.xml
XML.load|$$MYF
```

The loading command will automatically call the parsing command.
If you have already loaded the XML-file into a variable.you can use the command:

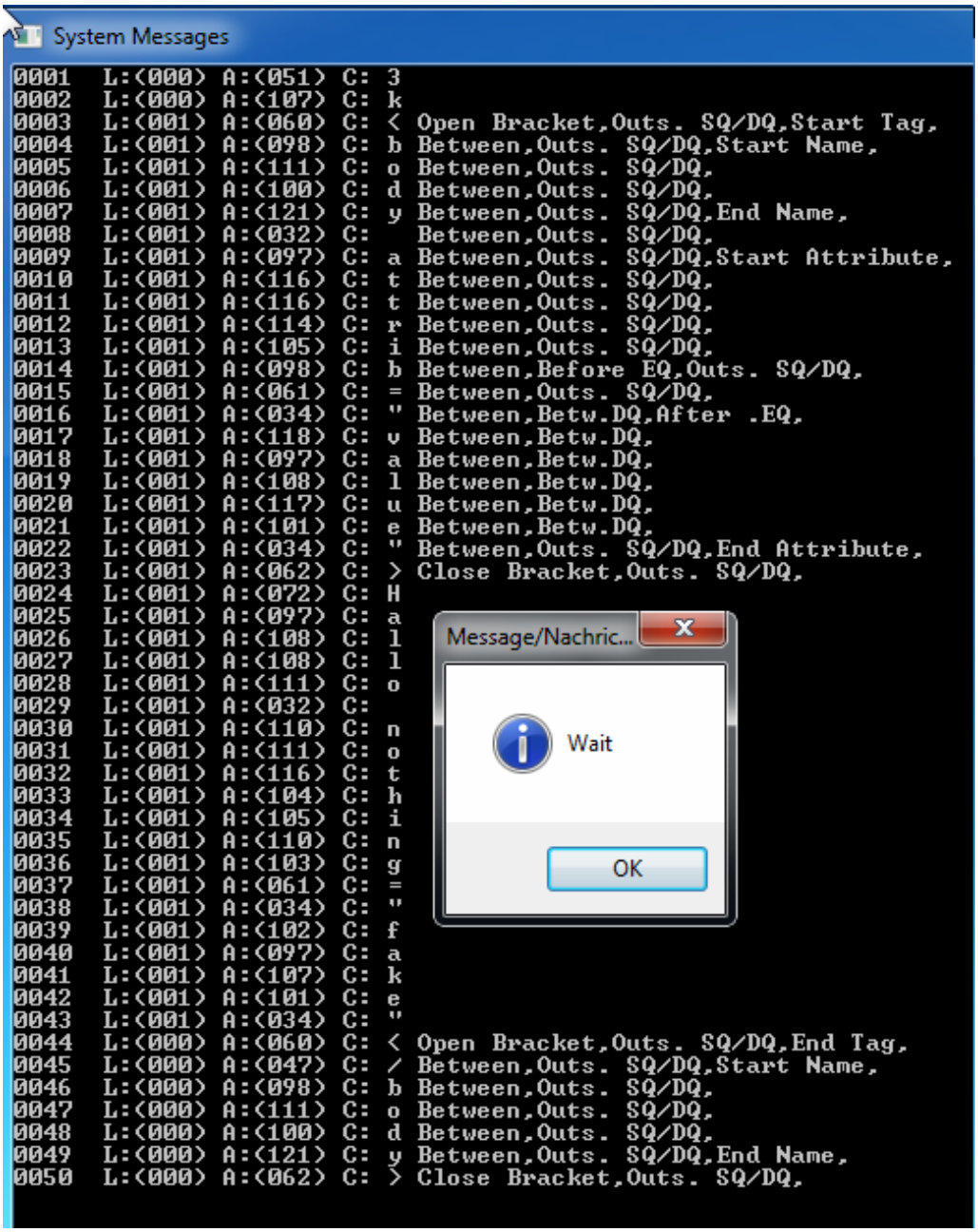
```
$$XML=3k<body attrib="value">Hallo nothing="fake"</body>
XML.parse|$$XML
```

After this, the file is in the internal XML Memory and it has been tagged with "Flags" in a way that you can much easier work with the file. You can read more details about these "Flags" below.
Using these Flags you can later more easily navigate inside the XML-Document.
You can see this using the

```
XML.dump
```

Command. Lets see an example, using the built in "dump" option.

© 2023 Theo Gottwald * https://www.smart-package.com




```
' This is a XML-"One-Liner"
$$XML=3k<body attrib="value">Hallo nothing="fake"</body>
' First we need to parse it
XML.parse|$$XML
' Here we check if there was an error during the parsing
JIT.Lab_Err
' Now we display the internal tables
XML.dump
' This is just to wait until want to close the script
MBX.Wait
ENR.

' This is the part that would be called if there would be an
error
' during the parsing of the XML
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

Above is the result output of this script.

In the above picture we see on the very left the "address" of each byte of XML-Code,

It starts basically with the first character on the left as 1 and counts up until the length of the file.

Note that several XML commands will never go beyond the root-address that is the first "<" in the file. In our case the root-address is 3. Which is the address of the very first start tag.

None of the XML commands will go above the end address of the last byte in the XML-File.
Doing so would lead to a `GPF (Windows General Protection Fault)` and is therefore prevented.

The second number from the left shows the "Level" "**`L:`**" of the character.
Each Start Tag increases the level by 1, while each end tag decreases the level by one.

Then comes "**`A:`**" the ASC-value of the letter(code) and the "**`C:`**" Character.
Note that some .characters are replaced by a "-" if they are not printable.

After that, on the right you will see a comma separated list with flags that are being assigned to each byte in the XML-code. You can work with these flags using the commands:

```
XML.search inside down|$$FLG
XML.search inside up|$$FLG
```

These two commands search "Inside" that is between the "<" and the ">" brackets.

If you want to search for anythingthat is outside (including the "<"">" themselves, you need to use these commands.

```
XML.search all down|$$FLG
XML.search all up|$$FLG
```

To search something you simply define `$$FLG` as one of the SPR XML-Flags below.
For example:

```
$$FLG=open bracket
XML.search all down|$$FLG
```

will search a "open bracket" in the "down" direction. Note that if the current position is on a "open bracket", then this one will be found!
Therefore you may want to use the

```
XML.increment position
```

command before you start search. The position that is found, is placed on the TOS (Top of Stack) unless you specify a return variable. Like this:

© 2023 Theo Gottwald * https://www.smart-package.com

```
XML.increment position
$$FLG=open bracket
XML.search all down|$$FLG|$$POS
```

But even if you do not use a return variable, all searches remember the "actual position" internally. Like you locate a window, also the XML-Parser has always a "actual position".
Immediately after parsing the XML-File, the actual position is set to the "Root-Position". That is the first "<" in the file.

Now if you search something then it can be found or it can not be found. If nothing is found, then the actual position is not assigned!
The search command will return a zero in this case.
If you want to find out the actual position, you can use the

```
XML.get position|$$POS
```

command. In the same way you can set the actual position to another place in the XML-File using the

```
$$POS=79
XML.set position|$$POS
```

commands. This way you can jump up and down in the XML file like you desire.

Note that this is just the "basics". There are multiple command that make it more easy for you to find specified places or data inside the XML-File.

To understand the internals of the XML-Parser better you can read the Chapter:

SPR - XML-Parser Flags 3981

At this place we will not explain more details about XML, as you can get this information easily in the World Wide Web. For example here:

1. XML-Basics
2. XML-Schools
3. XML-Validator

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.3 ! SPR - XML-Parser Flags

Smarty's XML-Parser Flags

During the Parsing, the XML-Parser will assign Flags to each byte of the XML-Code.
These flags are the base of nearly all builtin XML-Commands.

You can see these flags using the

```
XML.Dump
```

Command. There is an example Output in the picture below.

The SPR-XML-Flags

Here is a complete list with all **SPR-XML Flags** and their corresponding numbers.

```
"open bracket"    =   1
"close bracket"   =   2
"between"         =   4
"is sq"           =   8
"between sq"      =  16     = &H10
"is dq"           =  32     = &H20
"between dq"      =  64     = &H40
"before eq"       = 128     = &H80
"after eq"        = 256     = &H100
"outside sdq"     = 512     = &H200
"self terminate"  = 1024    = &H400
"system tag"      = 2048    = &H800
"start name"      = 4096    = &H1000
"end name"        = 8192    = &H2000
"start attribute" = 16384   = &H4000
"end attribute"   = 32768   = &H8000
"start tag"       = 65536   = &H10000
"end tag"         = 131072  = &H20000
"comment"         = 262144  = &H40000
```

Note that you can specify whether the flag name, or the flag number, or the hexadecimal flag number. You can also combine Flags using the "+" Operator.

How to use this flags:

You can use these flags in several ways.

Assume we are at the Root-Adress and we are looking for the first "**=**" Sign.
This is also called "EQ" from "Equal Sign".
The other Shortcuts are "DQ" for "Double Quote" and "SQ" for "Single Quote".

*© 2023 Theo Gottwald * https://www.smart-package.com*

In this file, the first real "=" is at position 14. is at position 14 as you can see in the Dump below.
We can use the "Before EQ" Flag to get the position..

like this:

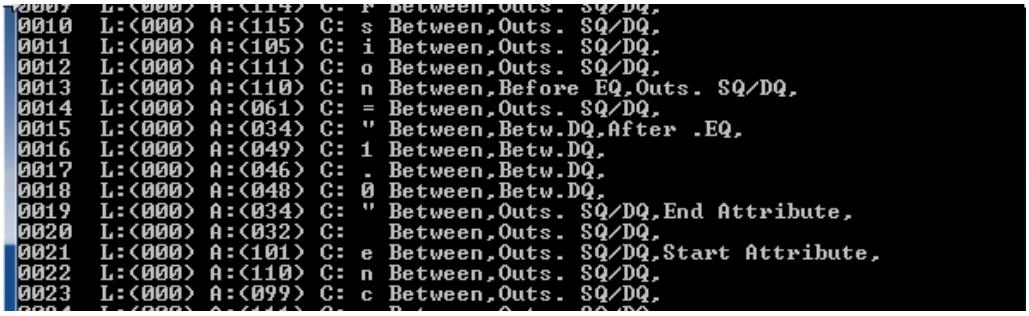
1. You can specify the flag number in a search command,

```
XML.search all down|128|$$RES
```

We have specified the "128" here as its the number for this flag.

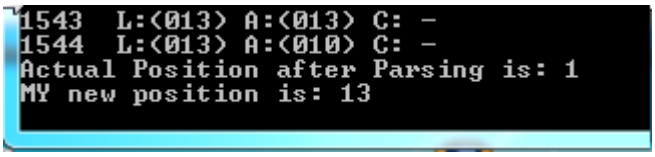
Example:
512 is the flag number for the "outside sdq" Flag.
128 is the flag number for the "before eq"

You can find these numbers above where the flags are.



2. You can specify a hexadezimal flag number as well:

```
XML.search all down|&H80|$$RES
PRT.MY new position is: $$RES
```



3. You can add multiple numbers using the "+" sign to search for combinations of flags:

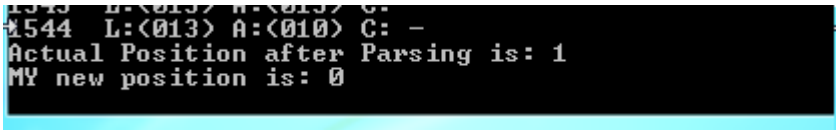
```
XML.search all down|&H80+&H200|$$RES
PRT.MY new position is: $$RES
```

In this case, ALL of the given flags must be available, otherwise the result will be zero.
Like in this example:

```
XML.search all down|&H80+12|$$RES
PRT.MY new position is: $$RES
```

We have an AND-Operation, if you use multiple flags,
As you can see, you can intermix hexadezimal and other numbers.

© 2023 Theo Gottwald * https://www.smart-package.com



4. You can specify the flag-name for better readablity.

```
PRT.Actual Position after Parsing is: $$POS
XML.search all down|"before eq"|$$RES
PRT.MY new position is: $$RES
```

It doesn't matter, if you specify the DQ's or not. You can also write:

```
XML.search all down|before eq|$$RES
```

which will do all the same. You can as well add Flags this way, using:

```
XML.search all down|"before eq","between"|$$RES
```

or you can do it like this:

```
XML.search all down|before eq+between|$$RES
```

theoretically the instructions run a bit slower if you use this text-form then if you use the numbers.
If you are not going to work on hundred thousands of Tags, this should not be of any interest to you.

You can directly access these flags using the command:

```
XML.gef flags|$$BYT|$$FLG
PRT.Byte Nr.$$BYT contains these flags: $$FLG
```

Note that these flags are cumulated to one number.
To work on these flags you can also use these commands:

```
' This will return the Flags as text like with Dump.
XML.gef flags as text|$$BYT|$$FLT
```

More features:

You can also retrieve the Level of a XML-Address like this:

```
XML.get level|234|$$LEV
PRT. The Level (234) is: $$LEV
```

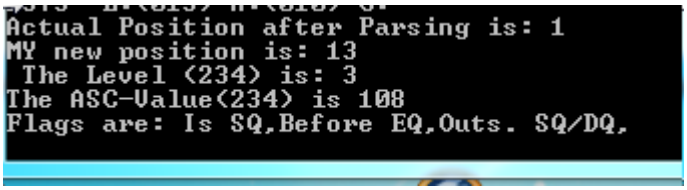
And of course you can retrieve the ASC-Value of any Address:

```
XML.get asc|234|$$ASC
PRT.The ASC-Value(234) is $$ASC
```

If you have a flag-value and you want to convert it to the appropriate Flags as text, you can use:

```
XML.flags to text|$$FLG|$$FTX
PRT.Flags are: $$FTX
```


© 2023 Theo Gottwald * https://www.smart-package.com



Here is a explanation of these flags:

1. This Flag is assigned to any real open bracket "<" from any used tag.
"Open Bracket" = 1

2. This Flag is assigned to any real close bracket ">" from any used tag.
"Close Bracket" = 2

4. This Flag is assigned to any character that is inside a tag. Inside a tag means, between a
open bracket ">" and a close bracket ">".
"Between" = 4

8. This Flag is assigned to any SQ (Single Quote) that encloses an attribute.
This flag is not assigned to other SQ's for example in an data-part or inside an DQ-delimited
value of an attribute.
"Is SQ" = 8

16.. This Flag is assigned to any character that is between SQ as an attribute value.
Between SQ" = 16

32. This Flag is assigned to any real DQ (Double Quote) that encloses an attribute.
This flag is not assigned to other DQ's for example in an data-part or inside an SQ-delimited
"is dq" = 32

64.. This Flag is assigned to any character that is between DQ as an attribute value.
"Between DQ" = 64

128. This Flag is assigned to the last letter (non whitespace character) that is before an
EQ Sign that is interpreted to be part of an attribute.
"before eq" = 128

256. This Flag is assigned to the first SQ or DQ , that is after an
EQ Sign that is interpreted to be part of an attribute.
"after eq" = 256

512. This Flag is assigned to to characters inside tags only. If they are not between SQ or DQ.
This means that they are not part of the value of an attribute, if this flag is set.
"outside sdq" = 512

© 2023 Theo Gottwald * https://www.smart-package.com

1024. This Flag is assigned to the open- and to the close-bracket of a self terminated tag.
These are comments, system-tags or dedicated "self terminated" tags.
See example XML-Dump below.
`"self terminate" = 1024`

2048 This Flag is assigned to the open- and to the close-bracket of a system-tag.
See example XML-Dump below.
`"system tag" = 2048`

4096. This Flag is assigned to the first letter of the tag name.
See example XML-Dump below.
`"start name"=4096`

8192. This Flag is assigned to the last letter of the tag name.
`"end name" = 8192`

16384. This Flag is assigned to the first letter of any found attribute.
See example XML-Dump below.
`"start attribute" = 16384`

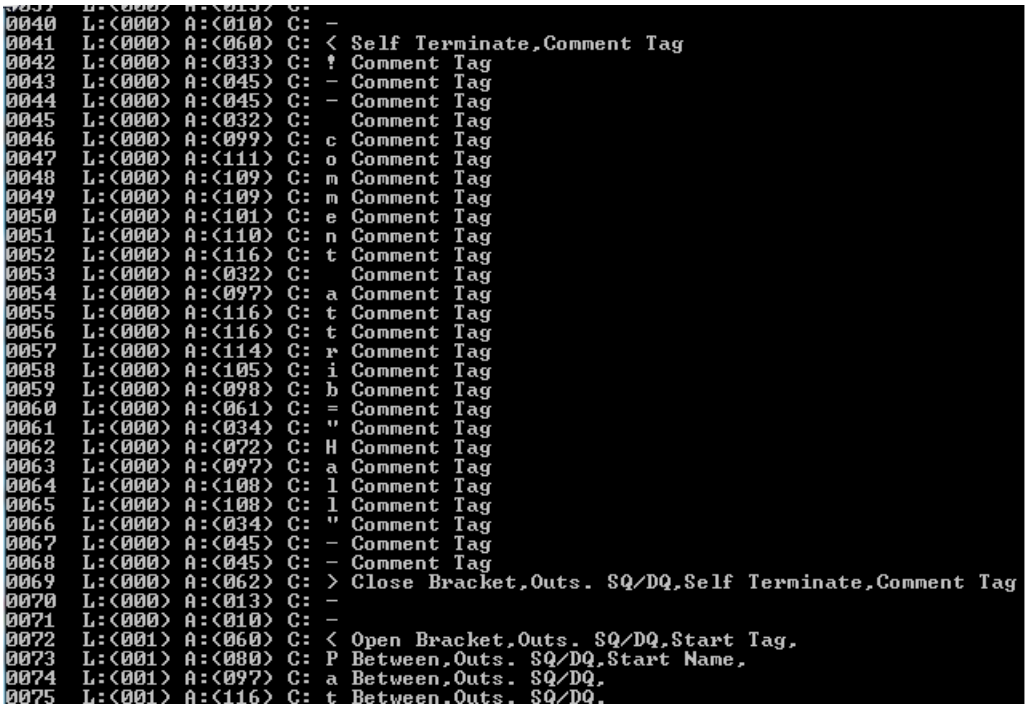
32768. This Flag is assigned to the last letter of any found attribute.
Regularly this is a SQ or a DQ-Sign.
See example XML-Dump below.
`"end attribute" = 32768`

65536. This Flag is assigned to the first "<" that starts a new Start-Tag. Its the opposite of an End-Tag-Flag (below). See example XML-Dumps below.
`"start tag" = 65536`

131072. This Flag is assigned to the first "<" that starts a new End-Tag. Its the opposite of an Start-Tag-Flag (above). See example XML-Dumps below.
`"end tag" = 131072`

262144. This Flag is assigned to any character that is inside a comment-tag.
There is an picture below that shows the Dump of an comment tag.
`"comment" = 262144`

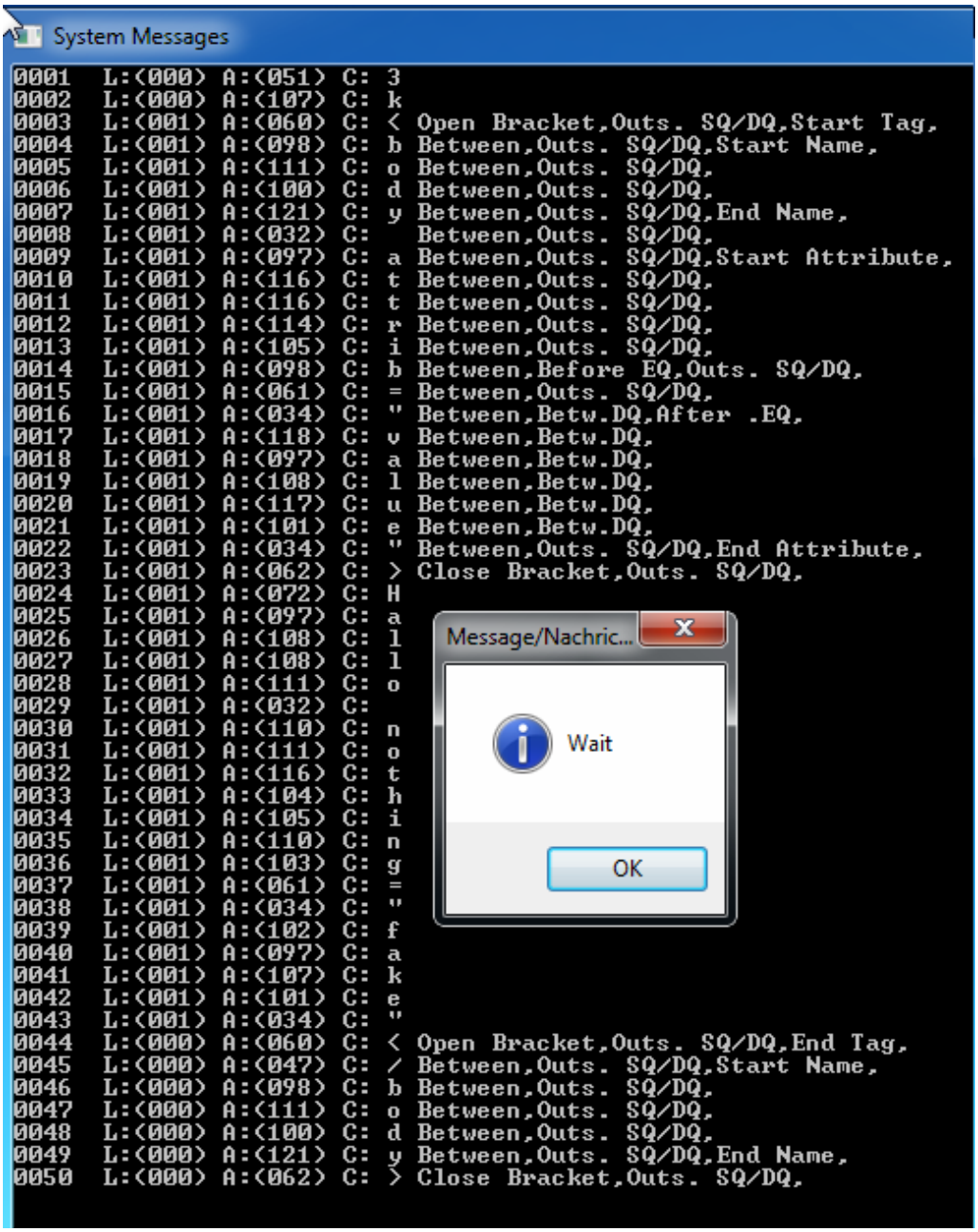
© 2023 Theo Gottwald * https://www.smart-package.com



Below is another XML-Dump, where you can see the flags.

```
' This is a XML-"One-Liner"
$$XML=3k<body attrib="value">Hallo nothing="fake"</body>
' First we need to parse it
XML.parse|$$XML
' Here we check if there was an error during the parsing
JIT.Lab_Err
' Now we display the internal tables
XML.dump
' This is just to wait until want to close the script
MBX.Wait
ENR.

' This is the part that would be called if there would be an
error
' during the parsing of the XML
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

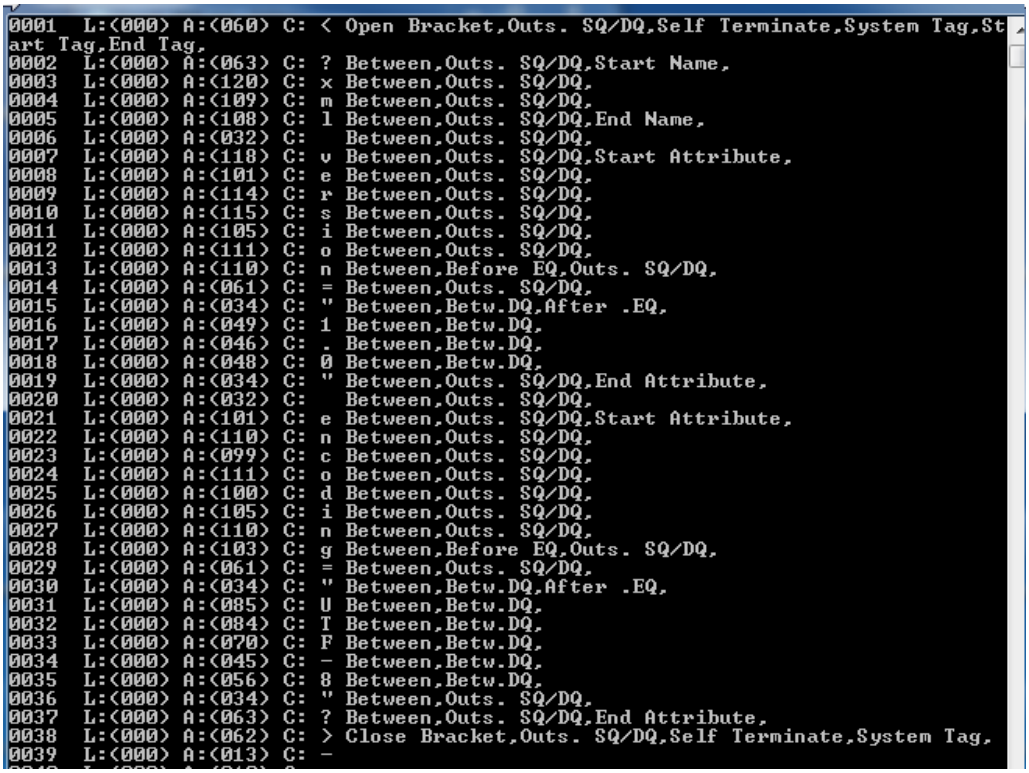

© 2023 Theo Gottwald * https://www.smart-package.com



Above is the result output of this script.

Below is another XML-Dump showing several SPR XML-Flags.

© 2023 Theo Gottwald * https://www.smart-package.com



See also:

- afz - Any Flags Zero 4016
- afs - Any Flags Set 4010
- stt - System Tag 4091
- stn - Start Name 4087
- stm - Self Terminate 4083
- stg - Start Tag 4079
- sta - Start Attribute 4075
- osd - Outside SDQ 4071
- obt - Open Bracket 4067
- isq - Is SQ 4063
- idq - Is DQ 4059
- etg - End Tag 4055
- edn - End Name 4051
- eat - End Attribute 4047
- cmt - Comment 4042
- cbt - Close Bracket 4038
- btw - Between 4034
- bsq - Between SQ 4030
- beq - Before EQ 4026
- bdq - Between DQ 4022

© 2023 Theo Gottwald * https://www.smart-package.com

- aeq - After EQ 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.4 ! XML. - eXtended Markup Language

MiniRobotLanguage (MRL)

XML. Command

Load, Parse and Analyze XML-Data

Intention

This command is a universal command to load, parse and analyze XML-Files and XML-Data.
Depending on the used Subcommand, more then 50 commands are available to handle your XML-Files.

There are 2 general sort of Sub-commands.

1. Commands that can be used anytime.

2. Commands that require an XML-File to be loaded or parsed before they can be used.
   Most commands are in this group.
   We can split these second sort of commands in further three groups:

a) Complex commands like:
```
XML.search name and attribute|$$NAM|$$ATN
```

b) Low level commands. These are all commands that work directly with the XML-Flags, like :
```
XML.search all up|$$FLG
```

c) Simple commands like:
```
XML.get asc|$$NUM|$$RES
```

Debugging XML-Scripts

You can use the commands:

```
XML.Print
XML.Dump
```

to get more or less detailed Print-Outs of your XML-Data. There is an example Script below in the examples section.

Internal notes

© 2023 Theo Gottwald * https://www.smart-package.com

Additionally there are few commands that enable you to change the loaded XML-Data.
Here is an important rule:

Changing the internal XML.-Data after Parsing is supported, but any change in Size would invalidate all previous XML-Parsing from that changed byte down to the end of the XML-Data. Therefore a "ReParsing" Option is available and will sometimes be automatically called.
This is due to the 1:1 Synchronity of the **XML-Flags** with the **XML-Data** Bytes that is eventually getting lost.

Even if you change large portions of XML-Data without changing the size, you may need to re-parse the XML-Data to have valid XML-Flags.

Note: If you do not change the XML-Data Size, the re-parsing is not done automatically!
You can make a manual "ReParse" of the changed XML-Data using the Subcommand:

```
XML.ReParse
```

Here is an example for an automatic ReParse. If you use (example):

```
XML.set xml part|6|6|sy
```

the Subcommand will automatically "re-parse" the XML-Data because the size of the XML-Data was changed during the command execution.

On very large XML-Files this may take always a few seconds, On small files the delay should not be noticed. Also problems while parsing the XML-Data will be reported just as if you had been called the "Parse" Subcommand.

You can make all changes in a way that the size of the data is not changed and the XML-Flags stay valid. This will prevent an internal Re-Parsing. You could for example change

```
<namex >
```

to

```
<namey >
```

using

```
XML.set chr|5|y
```

this will not change the size of the XML-Data and therefore the synchronity between the XML-Data and the flags will stay intact.

Here is another example that show where the problem is. If you change:

```
<namex >
```

to

```
<namesy >
```


© 2023 Theo Gottwald * https://www.smart-package.com

using

```
$$XML=<namex >
XML.parse|$$XML
XML.Print

XML.set xml part|6|6|sy
XML.Print
MBX.!
```

then downward from the "s", all XML-Flag Data would loose their "Flags" and therefore the whole Data is being Re-Parsed".

if you plan to make a completely changed XML-Data version of a very large XML-File, i would recommend to just extract portions of the original data, and rebuild it using variables.

Syntax

XML.P1[|P2..Px]

Parameter Explanation

P1 - (optional) Subcommand, one of the following:

See Subfolders in Index

P2 - Px (depend on used Subcommand P1)

Example

```
'-----------------------------
' Demonstrate XML.-Debug Commands
'-----------------------------
$$XMF=?pfad\Small_XML.txt
XML.load|$$XMF
JIT.Lab_Err
XML.Print
MBX.This is an XML-Printout
XML.Print|$$TXT
DBP.$$TXT
MBX.It can be directed to the Editor-Debug Window.
XML.Dump|20|100
MBX.This is a partial XML-Dump.
XML.Dump
MBX.This is a complete XML-Dump.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
' Error-Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
END.


'-----------------------------
' Demonstrate several Low Level
' XML._Subcommands
'-----------------------------
$$XMF=?pfad\Small_XML.txt
XML.load|$$XMF
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
' We use a Low Level Search here.
XML.search all down|"before eq","between"|$$RES
PRT.MY new position is: $$RES
PRT.------------------------
' We evaluate the Level at position 234
XML.get level|234|$$LEV
PRT. The Level (234) is: $$LEV
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|234|238
' We get a single ASC Byte (Text) out of the XML-Data
XML.get asc|234|$$ASC
PRT.The ASC-Value(234) is $$ASC
PRT.------------------------
' We get the Flag-Data of this byte as text
XML.gef flags as text|234
PRT.The Flags at Pos.234 is $$000
PRT.------------------------
' Here is an example on how to calculate with Flag-Values
CAL.$$FLG=128+512+8
XML.flags to text|$$FLG|$$FTX
PRT.Flags are: $$FTX
PRT.------------------------
MBX.!
ENR.
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.5 IML./ NML. - no parse

3.84.5.1 nsc - Name Char

MiniRobotLanguage (MRL)

IML.Name Char
IML.nsc

Check if XML-"Name"-Data complies with Standards

Intention

This command is used to verify if a single character, or any character in P1 is a valid XML-Character to define a XML-Name.

As this is a String Command, no parsing is required before using this command.

Here is an usage example:

```
$$XMF=SML<_Name
NML.ncc|$$XMF
  PRT.$$XMF is not a valid XML-Name.
  PRT.It contains an invalid character at position $$000.
EIF.
```

Syntax

IML.Name Char|P2[|P3]

Parameter Explanation

P2 – Variable with XML-Data. XML-Data can contain $cr, $lf or not.

P3 – ***(optional)*** Variable can contain a Position, then only the character at this position is been tested. If omitted or zero, all characters are tested. Only in this case, and if a character is found that does not apply to the Standards for a "XML name letter" then the position of the character is been placed on TOS (Top of Stack).

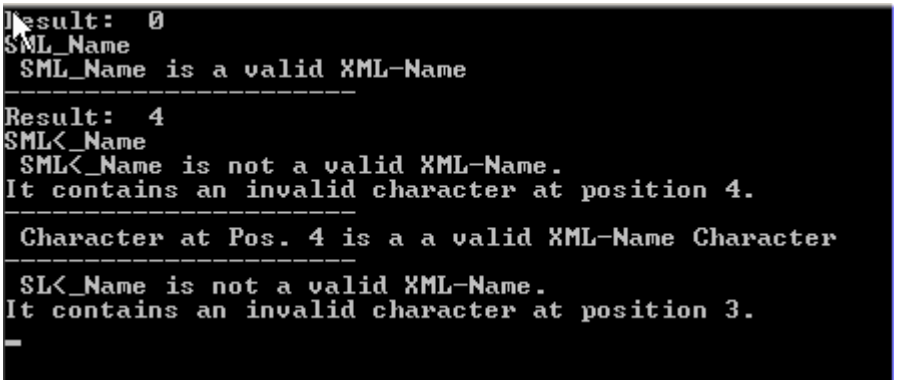
Example

```
'*********************************
' IML./NML. - Sample Script
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
' Valid Name
$$XMF=SML_Name
IML.ncc|$$XMF
  PRT. $$XMF is a valid XML-Name
EIF.
PRT.----------------------
' Not a valid Name-Char at Position 4
$$XMF=SML<_Name
NML.ncc|$$XMF
  PRT. $$XMF is not a valid XML-Name.$crlf$It contains an
invalid character at position $$000.
EIF.
PRT.----------------------
$$XMF=SML_Name
$$POS=4
IML.ncc|$$XMF|$$POS
  PRT. Character at Pos. $$POS is a a valid XML-Name Character
EIF.
PRT.----------------------
' Not a valid Name-Char at Position 3
$$POS=3
$$XMF=SL<_Name
NML.ncc|$$XMF|$$POS
  PRT. $$XMF is not a valid XML-Name.$crlf$It contains an
invalid character at position $$POS.
EIF.
MBX.!
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **nsc - Name Start** 3998

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.5.2 nsc - Name Start

MiniRobotLanguage (MRL)

IML.Name Start
IML.nsc

Verify if a String contains only valid XML Name-Start characters

Intention

This command is used to verify if a single character, or any character in P1 is a valid XML-Character to define a XML-Name Start Letter.

As this is a String Command, no parsing is required before using this command.

Here is an usage example:

```
$$XMF=SML<_Name
NML.nsc|$$XMF
  PRT.$$XMF is not a valid XML-Name Start letter.
  PRT.It contains an invalid character at position $$000.
EIF.
```

Syntax

IML.Name Start|P2[|P3]

Parameter Explanation

P2 – Variable with XML-Data. XML-Data can contain $cr, $lf or not.

P3 – ***(optional)*** Variable can contain a Position, then only the character at this position is been tested. If omitted or zero, all characters are tested. Only in this case, and if a character is found that does not apply to the Standards for a "XML name start letter" then the position of the character is been placed on TOS (Top of Stack).

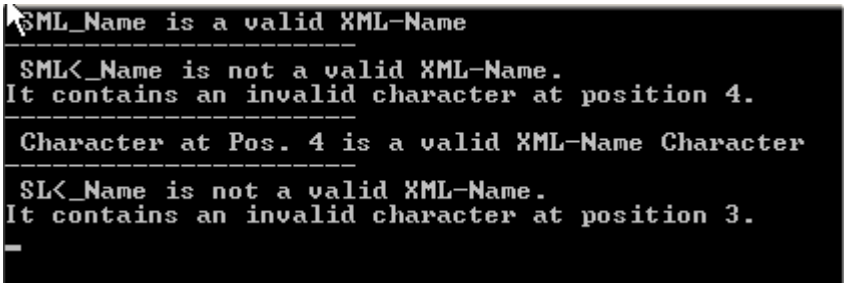
Example

```
'*********************************
' Valid Name Start Character(s)
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$XMF=SML_Name
IML.nsc|$$XMF
  PRT. $$XMF is a valid XML-Name
EIF.
PRT.----------------------
' Not a valid Name-Char at Position 4
$$XMF=SML<_Name
NML.nsc|$$XMF
  PRT. $$XMF is not a valid XML-Name.$crlf$It contains an
invalid character at position $$000.
EIF.
PRT.----------------------
$$XMF=SML_Name
$$POS=4
IML.nsc|$$XMF|$$POS
  PRT. Character at Pos. $$POS is a valid XML-Name Character
EIF.
PRT.----------------------
' Not a valid Name-Char at Position 3
$$POS=3
$$XMF=SL<_Name
NML.nsc|$$XMF|$$POS
  PRT. $$XMF is not a valid XML-Name.$crlf$It contains an
invalid character at position $$POS.
EIF.
MBX.!
ENR.
```

This is the result of thios sample Script:



.

Remarks

As this is a String Command, no parsing is required before using this command.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.5.3 wsa - Whitespace Asc

MiniRobotLanguage (MRL)

IML.Whitespace Asc
IML.wsa

Verify if a numeric number evaluates to a ASCII Code that is a valid XML Whitespace character

Intention

This command is used to check ASC-Values whether they are valid XML-Whitespaces or not.

As this is a String Command, no parsing is required before using this command.

Usage is simple:

```
$$XMF=64
IML.wsa|$$XMF
  PRT.Yes $$XMF is a valid XML-Whitespace character-code!
ELS.
  PRT.No, $$XMF is not a valid XML-Whitespace character-code!
EIF.
```

Syntax

IML.Whitespace Asc|P2

Parameter Explanation

P2 – Variable with an numeric ASC value (0-255).

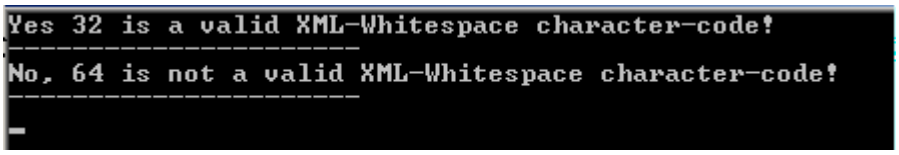
Example

```
'**********************************
' Valid Whitespace Character-ASC Numbers
'**********************************
'
$$XMF=32
GSB.test
$$XMF=64
GSB.test
MBX.!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ENR.
:test
IML.wsa|$$XMF
  PRT.Yes $$XMF is a valid XML-Whitespace character-code!
ELS.
  PRT.No, $$XMF is not a valid XML-Whitespace character-code!
EIF.
PRT.----------------------
RET.
```

This is the result of the sample script:



Remarks

As this is a String Command, no parsing is required before using this command.

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **wsc - Whitespace Character** 4003

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.5.4 wsc - Whitespace Character

MiniRobotLanguage (MRL)

IML./NML.Whitespace Character
IML./NML.wsc

Verify if a String contains only valid XML Whitespace characters

Intention

This command is used to verify if a single character, or any character in P1 is a valid XML-Whitespace Character.

As this is a String Command, no parsing is required before using this command.

Here is an usage example:

```
$$XMF=  $crlf$SL-Name
NML.wsc|$$XMF
  PRT.This is not a valid XML-Whitespace.$crlf$It contains an
invalid character at position $$000.
EIF.
```

Syntax

IML.Whitespace Character|P2[|P3]

Parameter Explanation

P2 – Variable with XML-Data. XML-Data can contain $cr, $lf or not.

P3 – ***(optional)*** Variable can contain a Position, then only the character at this position is been tested. If omitted or zero, all characters are tested. Only in this case, and if a character is found that does not apply to the Standards for a "XML name start letter" then the position of the character is been placed on TOS (Top of Stack).

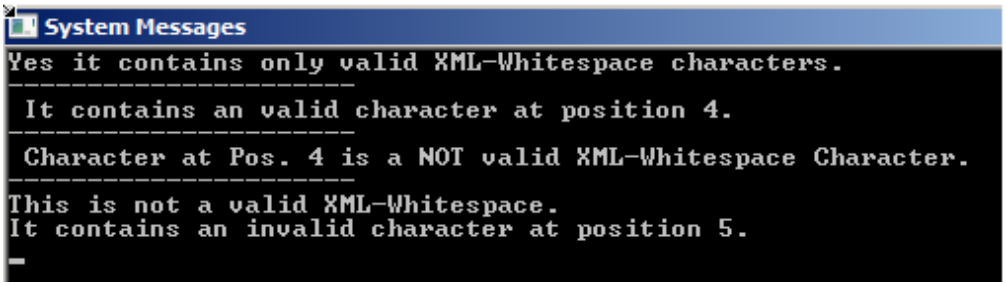
Example

```
'*********************************
' Valid Whitespace Characters
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
'
$$XMF=$sp$$crlf$
IML.wsc|$$XMF
  PRT.Yes it contains only valid XML-Whitespace characters.
EIF.
PRT.----------------------
' Is a valid Name-Char at Position 4
$$XMF=SML _Name
IML.wsc|$$XMF|4
  PRT. It contains an valid character at position 4.
EIF.
PRT.----------------------
$$XMF=    _Name
$$POS=4
NML.wsc|$$XMF|$$POS
  PRT. Character at Pos. $$POS is a NOT valid XML-Whitespace
Character.
EIF.
PRT.----------------------
' Not a valid Name-Char at Position 3
$$POS=3
$$XMF=  $crlf$SL-Name
NML.wsc|$$XMF
  PRT.This is not a valid XML-Whitespace.$crlf$It contains an
invalid character at position $$000.
EIF.
MBX.!
ENR.
```

This is the result of the sample script:



Remarks

As this is a String Command, no parsing is required before using this command.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **wsa - Whitespace Asc** 4001

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6 IML./NML. - after parse

3.84.6.1 aeq - After EQ

MiniRobotLanguage (MRL)

IML.After EQ
IML.aeq
IML.After EQZ
IML.aeqz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and "after an Equal-Sign" "=" or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.aeq
  PRT. Pos. 30 is after an Equal Sign.
EIF.

' With One parameter IML. will check just the given Byte
IML.aeq|30
  PRT. Pos. 30 is after an Equal Sign.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.aeq|20|40
  PRT. Pos. $$000 is after an Equal Sign.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.aeqz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

As always NML. is the negative Form of IML.

Syntax

IML.After EQ[|P2][|P3]
IML.After EQZ[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

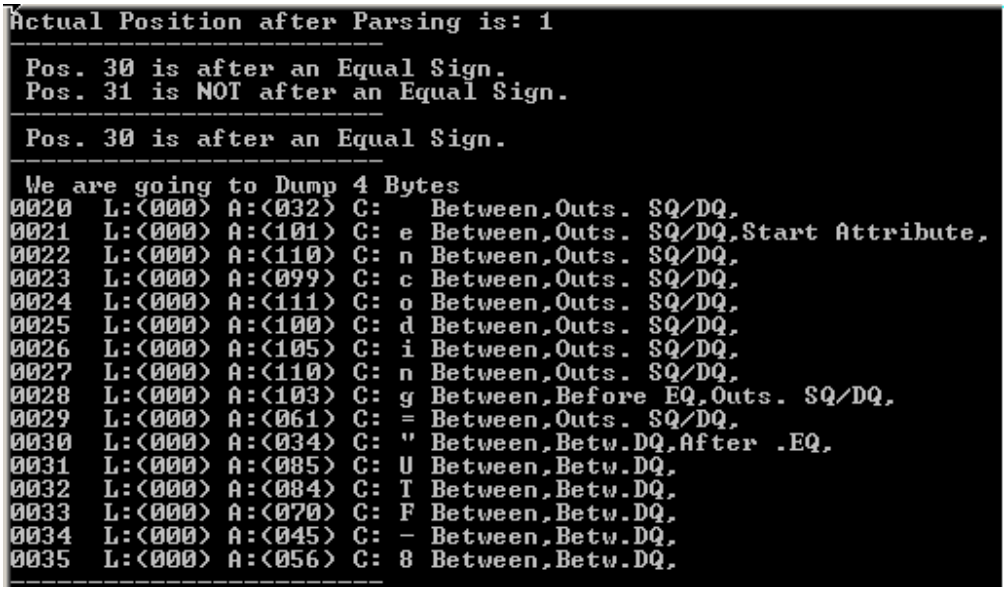
Example

```
'**********************************
' IML.Sample
'**********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
IML.aeq|30
  PRT. Pos. 30 is after an Equal Sign.
EIF.
IML.aeq|31
  PRT. Pos. 31 is after an Equal Sign.
ELS.
  PRT. Pos. 31 is NOT after an Equal Sign.
EIF.
PRT.------------------------
IML.aeq|20|40
  PRT. Pos. $$000 is after an Equal Sign.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.aeqz|30|40
  PRT. Pos. $$000 is the first byte after that.
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|20|35
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.2 afs - Any Flags Set

MiniRobotLanguage (MRL)

IML.Any Flags Set
IML.afs

Find out if the specified XML-Flag is set or Not set.

Intention

This command is used to analyze XML-Flags. Using this command, you can check more complex conditions by using multiple XML-Flags at once.
These XML-Flags will be AND'ed not OR'ed.

Any result will only be found, if ALL of the given XML-Flags are set at that target byte.

For example:

```
' This will search for a combination of the two flags
' starting at byte 1 until byte 500
IML.afs|Close Bracket+self terminate|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.

' Instead of the + you can also use the ,
IML.afs|Close Bracket,self terminate|1|500

' You can use the numbers directly
IML.afs|2+1024|1|500
```

or

```
IML.afs|2,1024|1|500
```

or

```
' This version is a few microseconds faster :-)
' if you are working on hundred thousands of tags in few
seconds
' use this one
IML.afs|1026|1|500
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
XML.Set Pos|30
IML.afs|8,2,2048
  PRT. Pos. 30 is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.afs|(1024+8+6+2)|30
  PRT. Pos. 30 is flagged.
EIF.

' With three parameters, IML. will start searching for a Byte
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.afs|Close Bracket+self terminate|20|40
  PRT. Pos. $$000 is after an Equal Sign.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Any Flags Set|P2[|P3][|P4]

Parameter Explanation

P2 - Variable with XML-Data. XML-Data Flags, see below.

P3 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P4 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

XML-Data Flags, to be used as P2:

- You can combine as many of the following flags as you want
- You can use the numeric value of the flag, or the better readable description

1. This Flag is assigned to any real open bracket "**<**" from any used tag.
   `"Open Bracket" = 1`

2. This Flag is assigned to any real close bracket "**>**" from any used tag.
   `"Close Bracket" = 2`

4. This Flag is assigned to any character that is inside a tag. Inside a tag means, between a

© 2023 Theo Gottwald * https://www.smart-package.com

open bracket ">" and a close bracket ">".
"Between" = 4

8. This Flag is assigned to any SQ (Single Quote) that encloses an attribute.
This flag is not assigned to other SQ's for example in an data-part or inside an DQ-delimited value of an attribute.
"Is SQ" = 8

16.. This Flag is assigned to any character that is between SQ as an attribute value.
Between SQ" = 16

32. This Flag is assigned to any real DQ (Double Quote) that encloses an attribute.
This flag is not assigned to other DQ's for example in an data-part or inside an SQ-delimited
"is dq" = 32

64.. This Flag is assigned to any character that is between DQ as an attribute value.
"Between DQ" = 64

128. This Flag is assigned to the last letter (non whitespace character) that is before an EQ Sign that is interpreted to be part of an attribute.
"before eq" = 128

256. This Flag is assigned to the first SQ or DQ , that is after an EQ Sign that is interpreted to be part of an attribute.
"after eq" = 256

512. This Flag is assigned to to characters inside tags only. If they are not between SQ or DQ. This means that they are not part of the value of an attribute, if this flag is set.
"outside sdq" = 512

1024. This Flag is assigned to the open- and to the close-bracket of a self terminated tag.
These are comments, system-tags or dedicated "self terminated" tags.
See example XML-Dump below.
"self terminate" = 1024

2048 This Flag is assigned to the open- and to the close-bracket of a system-tag.
See example XML-Dump below.
"system tag" = 2048

4096. This Flag is assigned to the first letter of the tag name.
See example XML-Dump below.
"start name"=4096

8192. This Flag is assigned to the last letter of the tag name.
"end name" = 8192

16384. This Flag is assigned to the first letter of any found attribute.
See example XML-Dump below.
"start attribute" = 16384

32768. This Flag is assigned to the last letter of any found attribute.

© 2023 Theo Gottwald * https://www.smart-package.com

Regularly this is a SQ or a DQ-Sign.
See example XML-Dump below.
"end attribute" = 32768

65536. This Flag is assigned to the first "<" that starts a new Start-Tag. Its the opposite of an End-Tag-Flag (below). See example XML-Dumps below.
"start tag" = 65536

131072. This Flag is assigned to the first "<" that starts a new End-Tag. Its the opposite of an Start-Tag-Flag (above). See example XML-Dumps below.
"end tag" = 131072

262144. This Flag is assigned to any character that is inside a comment-tag.
There is an picture below that shows the Dump of an comment tag.
"comment" = 262144

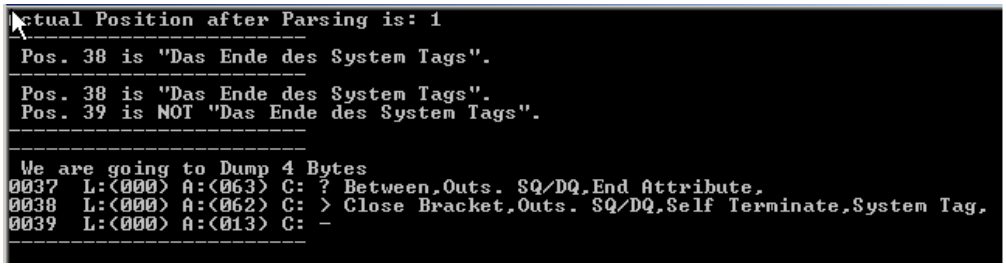
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=afs
$$TXT=Das Ende des System Tags
$$XMF=?pfad\Small_XML.txt
$$FLG=Close Bracket+self terminate
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|$$FLG|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$FLG|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|$$FLG|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Now we are scanning for "Bit not set"
IML.$$PROz|$$FLG|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|$$PLU
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990

© 2023 Theo Gottwald * https://www.smart-package.com

- par - parse from var 4117
- lf - load file 4113
- stt - System Tag 4091
- stn - Start Name 4087
- stm - Self Terminate 4083
- stg - Start Tag 4079
- sta - Start Attribute 4075
- osd - Outside SDQ 4071
- obt - Open Bracket 4067
- isq - Is SQ 4063
- idq - Is DQ 4059
- etg - End Tag 4055
- edn - End Name 4051
- eat - End Attribute 4047
- cmt - Comment 4042
- cbt - Close Bracket 4038
- btw - Between 4034
- bsq - Between SQ 4030
- beq - Before EQ 4026
- bdq - Between DQ 4022
- afz - Any Flags Zero 4016
- afs - Any Flags Set 4010
- aeq - After EQ 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.3 afz - Any Flags Zero

MiniRobotLanguage (MRL)

IML.Any Flags Zero
IML.afz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can check more complex conditions by using multiple XML-Flags at once.
These XML-Flags will be AND'ed not OR'ed.

Any result will only be found, if ALL of the given XML-Flags are NOT set (are zero) at that target byte.

For example:

```
' This will search for a combination of the two flags
' starting at byte 1 until byte 500
IML.afz|Close Bracket+self terminate|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.

' Instead of the + you can also use the ,
IML.afz|Close Bracket,self terminate|1|500

' You can use the numbers directly
IML.afz|2+1024|1|500
```

or

```
IML.afz|2,1024|1|500
```

or

```
' This version is a few microseconds faster :-)
' if you are working on hundred thousands of tags in few
seconds
' use this one
IML.afz|1026|1|500
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.afz|8,2,2048
  PRT. Pos. 30 is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.afz|(1024+8+6+2)|30
  PRT. Pos. 30 is flagged.
EIF.

' With three parameters, IML. will start searching for a Byte
with
' the specified Flag NOT set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.afz|Close Bracket+self terminate+8|20|40
  PRT. Pos. $$000 is after an Equal Sign.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Any Flags Zero|P2[|P3][|P4]

Parameter Explanation

P2 – Variable with XML-Data. XML-Data Flags, see below..

P3 – ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P4 – ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

XML-Data Flags, to be used as P2:

- You can combine as many of the following flags as you want
- You can use the numeric value of the flag, or the better readable description

1. This Flag is assigned to any real open bracket "<" from any used tag.
   `"Open Bracket" = 1`

2. This Flag is assigned to any real close bracket ">" from any used tag.
   `"Close Bracket" = 2`

4. This Flag is assigned to any character that is inside a tag. Inside a tag means, between a
   open bracket ">" and a close bracket ">".

© 2023 Theo Gottwald * https://www.smart-package.com

```
"Between"  =  4
```

8. This Flag is assigned to any SQ (Single Quote) that encloses an attribute.
This flag is not assigned to other SQ's for example in an data-part or inside an DQ-delimited value of an attribute.

```
"Is SQ"  =  8
```

16.. This Flag is assigned to any character that is between SQ as an attribute value.

```
Between SQ"  =  16
```

32. This Flag is assigned to any real DQ (Double Quote) that encloses an attribute.
This flag is not assigned to other DQ's for example in an data-part or inside an SQ-delimited

```
"is dq"  =  32
```

64.. This Flag is assigned to any character that is between DQ as an attribute value.

```
"Between DQ"  =  64
```

128. This Flag is assigned to the last letter (non whitespace character) that is before an EQ Sign that is interpreted to be part of an attribute.

```
"before eq"  =  128
```

256. This Flag is assigned to the first SQ or DQ , that is after an EQ Sign that is interpreted to be part of an attribute.

```
"after eq"  =  256
```

512. This Flag is assigned to to characters inside tags only. If they are not between SQ or DQ. This means that they are not part of the value of an attribute, if this flag is set.

```
"outside sdq"  =  512
```

1024. This Flag is assigned to the open- and to the close-bracket of a self terminated tag.
These are comments, system-tags or dedicated "self terminated" tags.
See example XML-Dump below.

```
"self terminate"  =  1024
```

2048 This Flag is assigned to the open- and to the close-bracket of a system-tag.
See example XML-Dump below.

```
"system tag"  =  2048
```

4096. This Flag is assigned to the first letter of the tag name.
See example XML-Dump below.

```
"start name"=4096
```

8192. This Flag is assigned to the last letter of the tag name.

```
"end name"  =  8192
```

16384. This Flag is assigned to the first letter of any found attribute.
See example XML-Dump below.

```
"start attribute"  =  16384
```

32768. This Flag is assigned to the last letter of any found attribute.
Regularly this is a SQ or a DQ-Sign.
See example XML-Dump below.

```
"end attribute"  =  32768
```


© 2023 Theo Gottwald * https://www.smart-package.com

65536. This Flag is assigned to the first "<" that starts a new Start-Tag. Its the opposite of an End-Tag-Flag (below). See example XML-Dumps below.
"start tag" = 65536

131072. This Flag is assigned to the first "<" that starts a new End-Tag. Its the opposite of an Start-Tag-Flag (above). See example XML-Dumps below.
"end tag" = 131072

262144. This Flag is assigned to any character that is inside a comment-tag. There is an picture below that shows the Dump of an comment tag.
"comment" = 262144

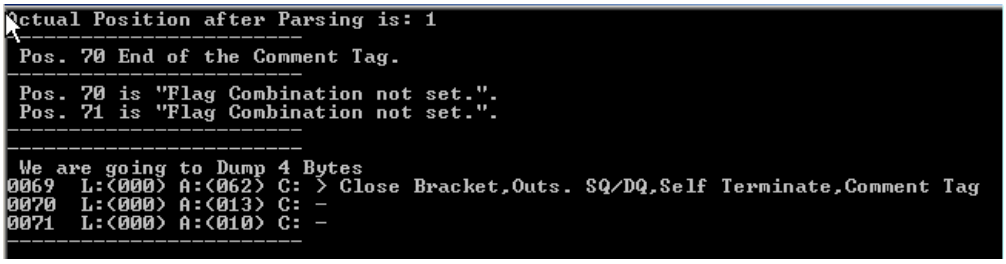
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=afz
$$TXT=Flag Combination not set.
$$XMF=?pfad\Small_XML.txt
$$FLG=Close Bracket+comment
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.cmt|1|500
  $$STR=$$000
EIF.
IML.$$PRO|$$FLG|$$STR|500
  $$ERG=$$000
  PRT. Pos. $$ERG End of the Comment Tag.
EIF.
PRT.------------------------
IML.$$PRO|$$FLG|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|$$FLG|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PROz|$$FLG|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|$$PLU
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

- afz - Any Flags Zero 4016
- afs - Any Flags Set 4010
- stt - System Tag 4091
- stn - Start Name 4087
- stm - Self Terminate 4083
- stg - Start Tag 4079
- sta - Start Attribute 4075
- osd - Outside SDQ 4071
- obt - Open Bracket 4067
- isq - Is SQ 4063
- idq - Is DQ 4059
- etg - End Tag 4055
- edn - End Name 4051
- eat - End Attribute 4047
- cmt - Comment 4042
- cbt - Close Bracket 4038
- btw - Between 4034
- bsq - Between SQ 4030
- beq - Before EQ 4026
- bdq - Between DQ 4022
- aeq - After EQ 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.4 bdq - Between DQ

MiniRobotLanguage (MRL)

IML.Between DQ
IML.bdq
IML.Between DQZ
IML.bdqz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute and is "between Double Quotes" or not.
You can also scan for the first byte that is outside of these double quotes (DQ).
DQ inside an attribute value that is delimited using SQ are not real DQ and are therefore not flagged.

```
' Here is an example:
' These DQ's will not be flagged
' also the "=" after "Liste" will not be flagged
<Hallo first='Liste="1234'"ok'>

' These DQ's will be flagged
<Hallo first="Liste='1234'ok">
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.bdq
  PRT. Pos. 30 is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.bdq|30
  PRT. Pos. 30 is between DQ's
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.bdq|20|40
  PRT. Pos. $$000 is flagged.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z". This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.bdqz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Between DQ[|P2][|P3]
IML.bdq[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

Example

```
'**********************************
$$PRO=Between DQ
$$TXT=Between Double Quotes
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|30
  PRT. Pos. 30 is "$$TXT".
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PRO|21
  PRT. Pos. 21 is "$$TXT".
ELS.
  PRT. Pos. 21 is NOT "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|20|40
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|20|($$RES+10)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
------------------------
 Pos. 30 is "Between Double Quotes".
 Pos. 21 is NOT "Between Double Quotes".
------------------------
 Pos. 30 is "Between Double Quotes".
------------------------
 Pos. 36 is the first byte after "Between Double Quotes".
------------------------
 We are going to Dump 4 Bytes
0020  L:(000) A:(032) C:   Between,Outs. SQ/DQ,
0021  L:(000) A:(101) C: e Between,Outs. SQ/DQ,Start Attribute,
0022  L:(000) A:(110) C: n Between,Outs. SQ/DQ,
0023  L:(000) A:(099) C: c Between,Outs. SQ/DQ,
0024  L:(000) A:(111) C: o Between,Outs. SQ/DQ,
0025  L:(000) A:(100) C: d Between,Outs. SQ/DQ,
0026  L:(000) A:(105) C: i Between,Outs. SQ/DQ,
0027  L:(000) A:(110) C: n Between,Outs. SQ/DQ,
0028  L:(000) A:(103) C: g Between,Before EQ,Outs. SQ/DQ,
0029  L:(000) A:(061) C: = Between,Outs. SQ/DQ,
0030  L:(000) A:(034) C: " Between,Betw.DQ,After .EQ,
0031  L:(000) A:(085) C: U Between,Betw.DQ,
0032  L:(000) A:(084) C: T Between,Betw.DQ,
0033  L:(000) A:(070) C: F Between,Betw.DQ,
0034  L:(000) A:(045) C: - Between,Betw.DQ,
0035  L:(000) A:(056) C: 8 Between,Betw.DQ,
0036  L:(000) A:(034) C: " Between,Outs. SQ/DQ,
0037  L:(000) A:(063) C: ? Between,Outs. SQ/DQ,End Attribute,
0038  L:(000) A:(062) C: > Close Bracket,Outs. SQ/DQ,Self Terminate,System Tag,
0039  L:(000) A:(013) C: -
0040  L:(000) A:(010) C: -
0041  L:(000) A:(060) C: < Self Terminate,Comment Tag
0042  L:(000) A:(033) C: ! Comment Tag
0043  L:(000) A:(045) C: - Comment Tag
0044  L:(000) A:(045) C: - Comment Tag
0045  L:(000) A:(032) C:   Comment Tag
0046  L:(000) A:(099) C: c Comment Tag
------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **If - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.5 beq - Before EQ

MiniRobotLanguage (MRL)

IML.Before EQ
IML.beq
IML.Before EQZ
IML.beqz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and "before an Equal-Sign" "=" or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.beq
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.beq|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.beq|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.beqz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

IML.Before EQ[|P2][|P3]
IML.beq[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

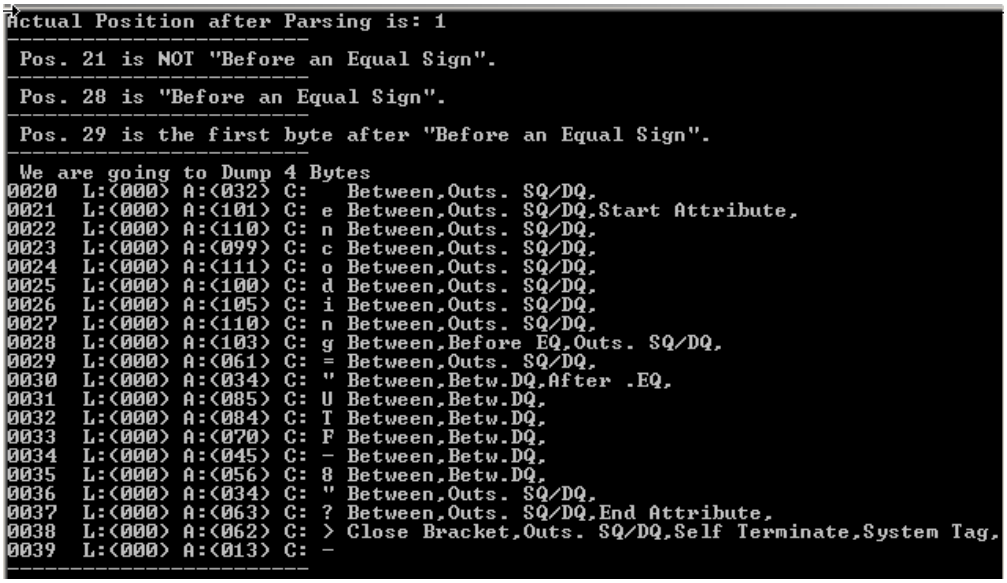
Example

```
'**********************************
' IML:beq - Sample
'**********************************
'Prototype für IML.
$$PRO=Before EQ
$$TXT=Before an Equal Sign
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|30
  PRT. Pos. 30 is "$$TXT".
EIF.
IML.$$PRO|21
  PRT. Pos. 21 is "$$TXT".
ELS.
  PRT. Pos. 21 is NOT "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|20|40
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|20|($$RES+10)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.6 bsq - Between SQ

MiniRobotLanguage (MRL)

IML.Between SQ
IML.bsq
IML.Between SQZ
IML.bsqz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and "between real Single Quotes" "=" or not.
Single Quotes inside of an Attribute Value that is delimited with Double Quotes are not "real" Single Quotes, and are not flagged.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.bsq
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.bsq|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.bsq|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.bsqz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Between SQ[|P2][|P3]
IML.bsq[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

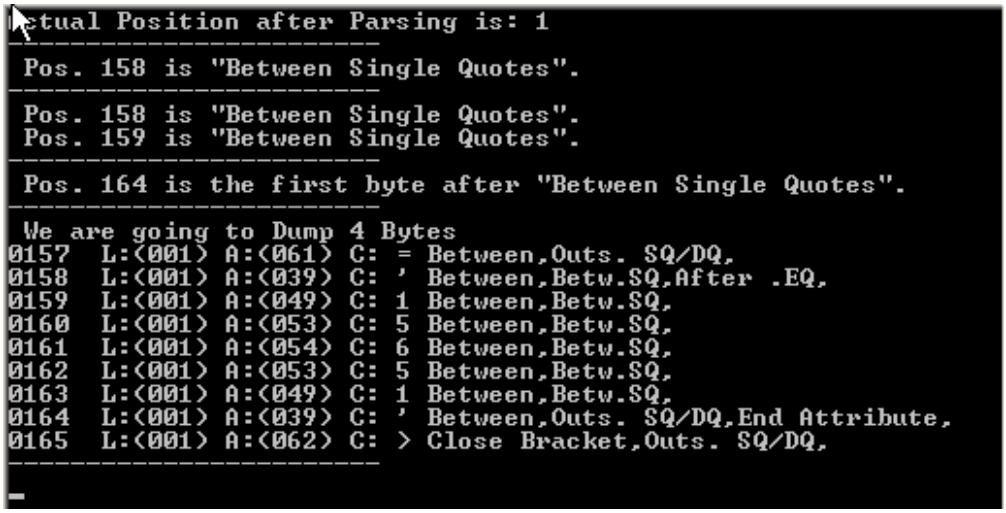
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Between SQ
$$TXT=Between Single Quotes
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113

© 2023 Theo Gottwald * https://www.smart-package.com

Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.7 btw - Between

MiniRobotLanguage (MRL)

IML.Between
IML.btw
IML.BetweenZ
IML.btwz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is inside a Tag that is between < > or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.btw
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.btw|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.btw|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.btwz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IML.Between[|P2][|P3]
IML.btw[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

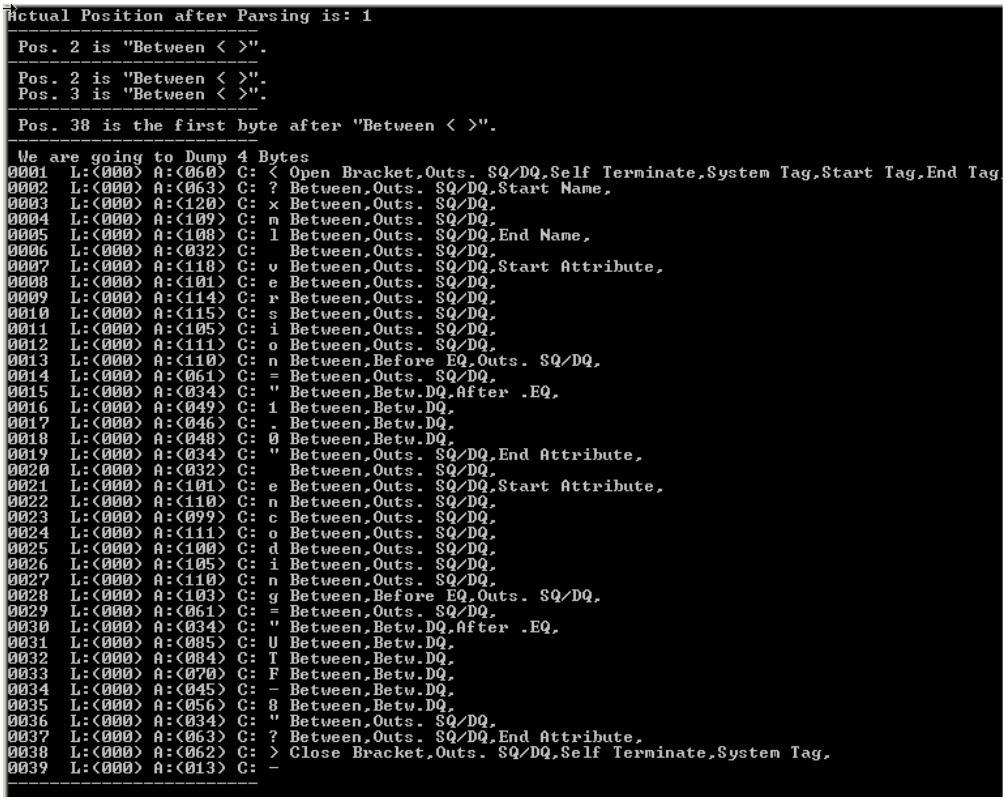
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Between
$$TXT=Between < >
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the

© 2023 Theo Gottwald * https://www.smart-package.com

par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- XML. - eXtended Markup Language 3990
- par - parse from var 4117
- If - load file 4113
- stt - System Tag 4091
- stn - Start Name 4087
- stm - Self Terminate 4083
- stg - Start Tag 4079
- sta - Start Attribute 4075
- osd - Outside SDQ 4071
- obt - Open Bracket 4067
- isq - Is SQ 4063
- idq - Is DQ 4059
- etg - End Tag 4055
- edn - End Name 4051
- eat - End Attribute 4047
- cmt - Comment 4042
- cbt - Close Bracket 4038
- btw - Between 4034
- bsq - Between SQ 4030
- beq - Before EQ 4026
- bdq - Between DQ 4022
- afz - Any Flags Zero 4016
- afs - Any Flags Set 4010
- aeq - After EQ 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.8 cbt - Close Bracket

MiniRobotLanguage (MRL)

IML.Close Bracket
IML.cbt
IML.Close BracketZ
IML.cbtz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is flagged as a real > or not.
Close brackets (while not really conform with standards) inside attribute values are ignored and would not be flagged as a close bracket. Also close brackets are only valid after an < open bracket first. Duplicate > are not flagged.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.cbt
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.cbt|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.cbt|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.cbtz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

As always `NML.` is the negative Form of `IML.`

Syntax

IML.close bracket[|P2][|P3]
IML.cbt[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

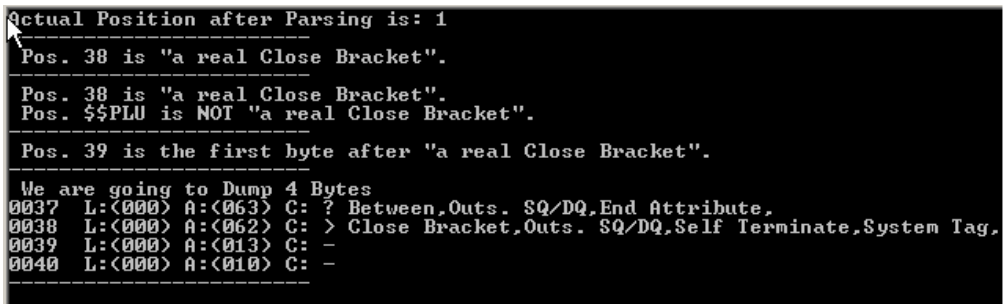
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Close Bracket
$$TXT=a real Close Bracket
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.9 cmt - Comment

MiniRobotLanguage (MRL)

IML.Comment
IML.cmt
IML.CommentZ
IML.cmtz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is flagged as a comment or not.

In XML comments look like this:

```
<!-- comment attrib="Hall"-->
```

All characters inside this comment Tag are flagged as a comment.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.cmt
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.cmt|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.cmt|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.cmtz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Comment[|P2][|P3]
IML.cmt[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

Example

```
'*********************************
'
'*********************************
'Prototype für IML.
$$PRO=Comment
$$TXT=a Character inside a comment
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
------------------------
 Pos. 41 is "a Character inside a comment".
------------------------
 Pos. 41 is "a Character inside a comment".
 Pos. 42 is "a Character inside a comment".
------------------------
 Pos. 70 is the first byte after "a Character inside a comment".
------------------------
 We are going to Dump 4 Bytes
0040  L:(000) A:(010) C: -
0041  L:(000) A:(060) C: < Self Terminate,Comment Tag
0042  L:(000) A:(033) C: ! Comment Tag
0043  L:(000) A:(045) C: - Comment Tag
0044  L:(000) A:(045) C: - Comment Tag
0045  L:(000) A:(032) C:   Comment Tag
0046  L:(000) A:(099) C: c Comment Tag
0047  L:(000) A:(111) C: o Comment Tag
0048  L:(000) A:(109) C: m Comment Tag
0049  L:(000) A:(109) C: m Comment Tag
0050  L:(000) A:(101) C: e Comment Tag
0051  L:(000) A:(110) C: n Comment Tag
0052  L:(000) A:(116) C: t Comment Tag
0053  L:(000) A:(032) C:   Comment Tag
0054  L:(000) A:(097) C: a Comment Tag
0055  L:(000) A:(116) C: t Comment Tag
0056  L:(000) A:(116) C: t Comment Tag
0057  L:(000) A:(114) C: r Comment Tag
0058  L:(000) A:(105) C: i Comment Tag
0059  L:(000) A:(098) C: b Comment Tag
0060  L:(000) A:(061) C: = Comment Tag
0061  L:(000) A:(034) C: " Comment Tag
0062  L:(000) A:(072) C: H Comment Tag
0063  L:(000) A:(097) C: a Comment Tag
0064  L:(000) A:(108) C: l Comment Tag
0065  L:(000) A:(108) C: l Comment Tag
0066  L:(000) A:(034) C: " Comment Tag
0067  L:(000) A:(045) C: - Comment Tag
0068  L:(000) A:(045) C: - Comment Tag
0069  L:(000) A:(062) C: > Close Bracket,Outs. SQ/DQ,Self Terminate,Comment Tag
0070  L:(000) A:(013) C: -
0071  L:(000) A:(010) C: -
------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **If - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.10 eat - End Attribute

MiniRobotLanguage (MRL)

IML.End Attribute
IML.eat
IML.End AttributeZ
IML.eatz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and in fact the "End of an attribute" or not.
This flag is normally given to DQ (Double Quote) or SQ (Single Quote) Signs at the end of an attribute value.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.eat
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.eat|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.eat|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.eatz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

As always `NML.` is the negative Form of `IML.`

Syntax

IML.End Attribute[|P2][|P3]
IML.eat[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

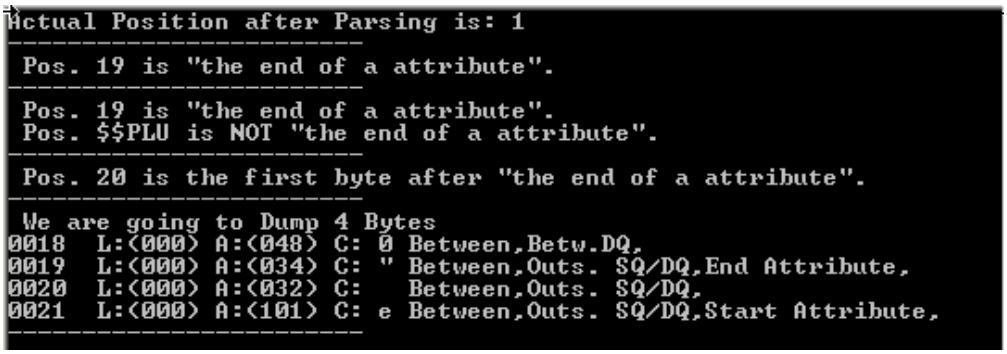
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=End Attribute
$$TXT=the end of a attribute
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.11 edn - End Name

MiniRobotLanguage (MRL)

IML.End Name
IML.edn
IML.End NameZ
IML.ednz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is the last byte of the Tag-Name or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.edn
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.edn|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.edn|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.ednz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IML.End Name[|P2][|P3]
IML.edn[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

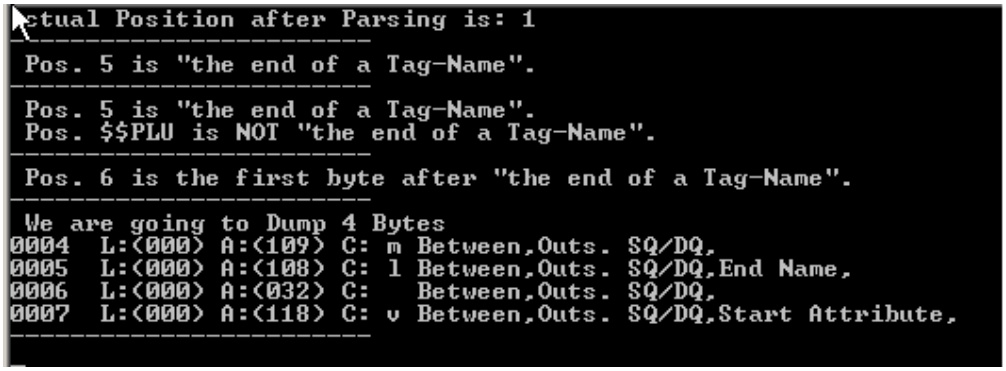
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=End Name
$$TXT=the end of a Tag-Name
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.12 etg - End Tag

MiniRobotLanguage (MRL)

IML.End Tag
IML.etg
IML.End TagZ
IML.etgz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is the start of a real End-Tag or not.

Example:

```
<Hallo> is a Start-Tag then
</Hallo> is the End-Tag of this.
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.etg
  PRT. Pos. 30 is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.etg|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.etg|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.etgz|30|40
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  PRT. Pos. $$000 is flagged.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.End Tag[|P2][|P3]
IML.etg[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

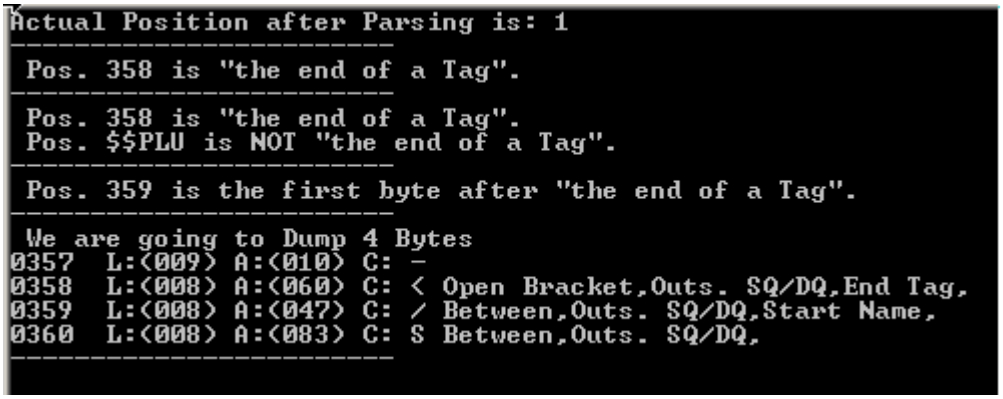
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=End Tag
$$TXT=the end of a Tag
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
' Wir starten hier von 30 da System-Tags (Pos.1)
' Self-Termianting sind, dabei ist das Start-Tag auch das End
Tag,
PRT.------------------------
IML.$$PRO|30|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
PRT.-----------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.-----------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.-----------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.-----------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113

© 2023 Theo Gottwald * https://www.smart-package.com

Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.13 idq - Is DQ

MiniRobotLanguage (MRL)

IML.Is DQ
IML.idq
IML.Is DQZ
IML.idqz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and a real DQ (Double Quote) or not.
A DQ that is enclosed in a attribute Value inside SQ (Single Quotes) is not a real DQ and is not flagged.

```
' Here is an example:
' These DQ's will not be flagged
<Hallo first='Liste="1234'"ok'>

' These DQ's will be flagged
<Hallo first="Liste='1234'ok">
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.idq
  PRT. Pos. 30 is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.idq|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.idq|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.idqz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Is DQ[|P2][|P3]
IML.idq[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

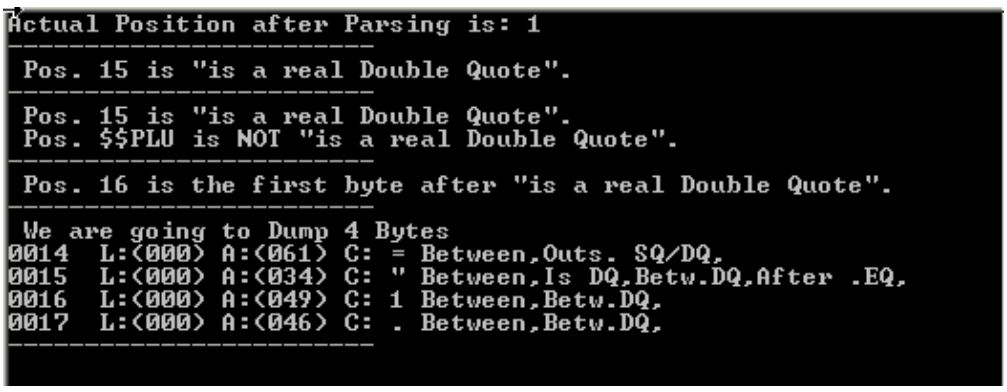
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Is DQ
$$TXT=is a real Double Quote
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.-----------------------
IML.$$PRO|10|500
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the

© 2023 Theo Gottwald * https://www.smart-package.com

If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **If - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.14 isq - Is SQ

MiniRobotLanguage (MRL)

IML.Is SQ
IML.isq
IML.Is SQZ
IML.isqz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and "after an Equal-Sign" "=" or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.isq
  PRT. Pos. 30 is is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.isq|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.isq|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.isqz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IML.Is SQ[|P2][|P3]
IML.isq[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

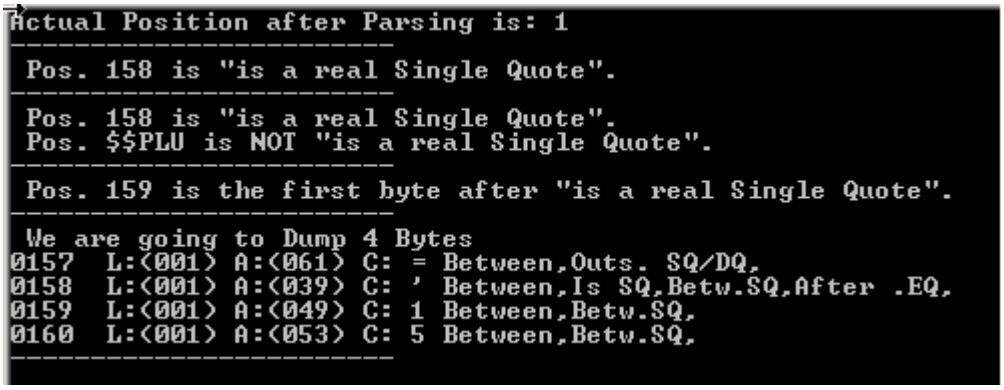
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Is SQ
$$TXT=is a real Single Quote
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command. In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.6.15 obt - Open Bracket

MiniRobotLanguage (MRL)

IML.Open Bracket
IML.obt
IML.Open BracketZ
IML.obtz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is flagged as a real < or not.
Open brackets (while not really conform with standards) inside attribute values are ignored and would not be flagged as a open bracket. Also open brackets are not valid after an < open bracket before. Duplicate < are not flagged.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.obt
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.obt|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.obt|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.obtz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Open Bracket[|P2][|P3]
IML.obt[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

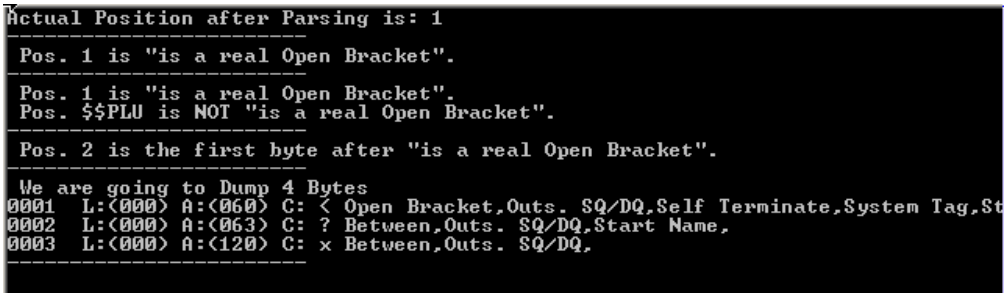
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=is a real Open Bracket
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- XML. - eXtended Markup Language 3990
- par - parse from var 4117
- lf - load file 4113
- stt - System Tag 4091
- stn - Start Name 4087
- stm - Self Terminate 4083
- stg - Start Tag 4079
- sta - Start Attribute 4075
- osd - Outside SDQ 4071
- obt - Open Bracket 4067
- isq - Is SQ 4063
- idq - Is DQ 4059
- etg - End Tag 4055
- edn - End Name 4051
- eat - End Attribute 4047
- cmt - Comment 4042
- cbt - Close Bracket 4038
- btw - Between 4034
- bsq - Between SQ 4030
- beq - Before EQ 4026
- bdq - Between DQ 4022
- afz - Any Flags Zero 4016
- afs - Any Flags Set 4010
- aeq - After EQ 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.16 osd - Outside SDQ

IML./NML.Outside SDQ Previous 4067 Top 41 Next 4075

MiniRobotLanguage (MRL)

IML.Outside SDQ
IML.osd
IML.Outside SDQZ
IML.osdz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is outside of any Single Quotes"or Double Quotes (SQ/DQ).

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.osd
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.osd|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.osd|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.osdz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IML.Outside SDQ[|P2][|P3]
IML.osd[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

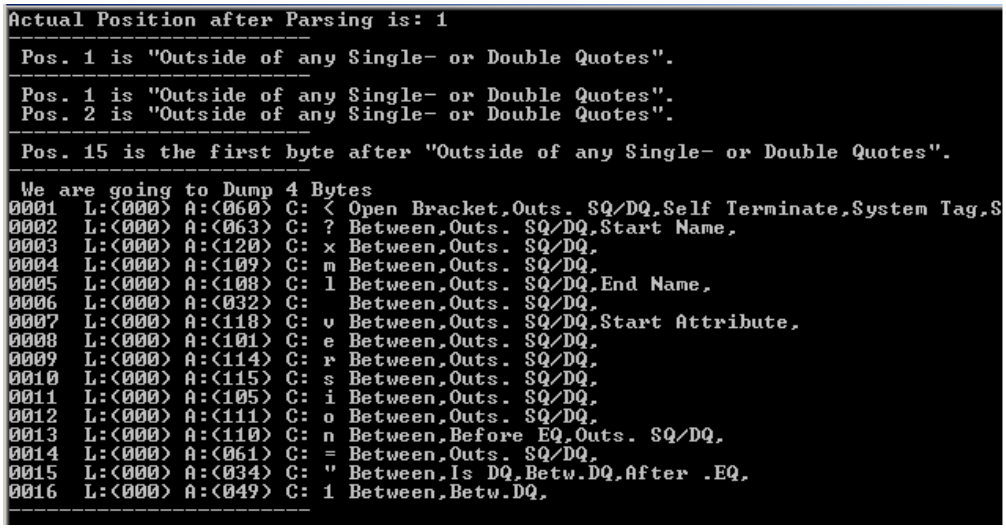
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Outside SDQ
$$TXT=Outside of any Single- or Double Quotes
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.17 sta - Start Attribute

MiniRobotLanguage (MRL)

IML.Start Attribute
IML.sta
IML.Start AttributeZ
IML.staz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is part of an attribute, and in fact the "Start of an attribute" or not.
This flag is normally given to the first Non-whitespace character inside a tag, after the name and an additional whitespace character.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.sta
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.sta|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.sta|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.staz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

As always NML. is the negative Form of IML.

Syntax

IML.Start Attribute[|P2][|P3]
IML.sta[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

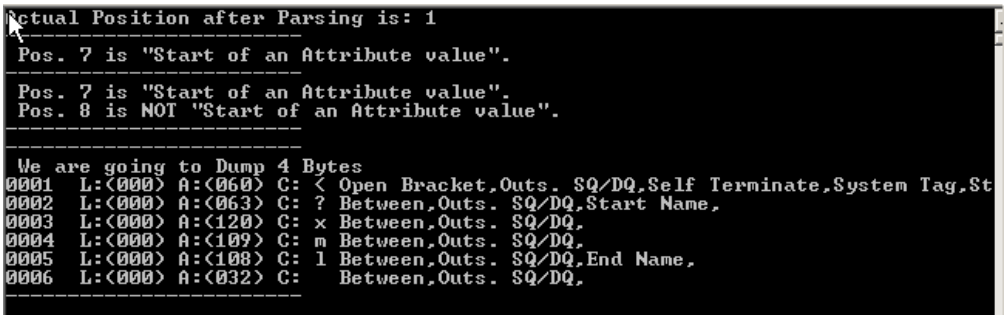
Example

```
'**********************************
'
'**********************************
$$PRO=Start Attribute
$$TXT=Start of an Attribute value
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.6.18 stg - Start Tag

MiniRobotLanguage (MRL)

IML.Start Tag
IML.stg
IML.Start TagZ
IML.stgz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is the start of a real Start-Tag or not.

Example:

```
<Hallo> is a Start-Tag then
</Hallo> is the End-Tag of this.
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.stg
  PRT. Pos. 30 is flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.stg|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.stg|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.stgz|30|40
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.Start Tag[|P2][|P3]
IML.stg[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

Example

```
'*********************************
'
'*********************************
'Prototype für IML.
$$PRO=Start Tag
$$TXT=Start of an Tag
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.6.19 stm - Self Terminate

MiniRobotLanguage (MRL)

IML.Self Terminate
IML.stm
IML.Self TerminateZ
IML.stmz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is marked as a part of a "self terminated tag" (could for example be a system tag) or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.stm
  PRT. Pos. 30 is flagged
EIF.

' With One parameter IML. will check just the given Byte
IML.stm|30
  PRT. Pos. 30 is flagged
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.stm|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.stmz|30|40
  PRT. Pos. $$000 is not flagged.
EIF.
```

As always `NML.` is the negative Form of `IML.`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IML.Self Terminate[|P2][|P3]
IML.stm[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

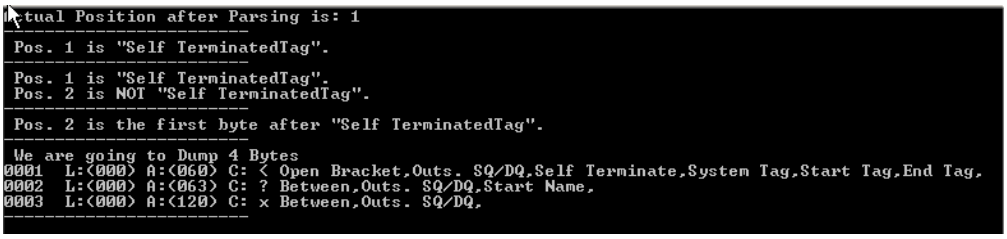
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Self Terminate
$$TXT=Self TerminatedTag
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
```

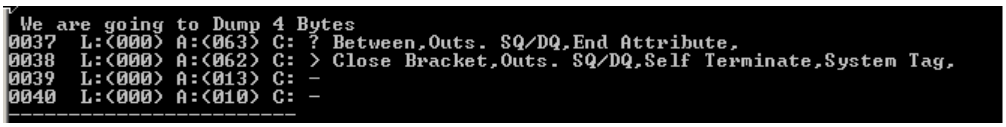

© 2023 Theo Gottwald * https://www.smart-package.com

```
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



In the second Dump, you can see that also the End-> is flagged as "self-terminate".



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.6.20 stn - Start Name

MiniRobotLanguage (MRL)

IML.Start Name
IML.stn
IML.Start NameZ
IML.stnz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is the First byte of the Tag-Name or not.

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|30
IML.stn
  PRT. Pos. 30 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.stn|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.stn|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.stnz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
EIF.
```

As always `NML.` is the negative Form of `IML.`

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

IML.Start Name[|P2][|P3]
IML.stn[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

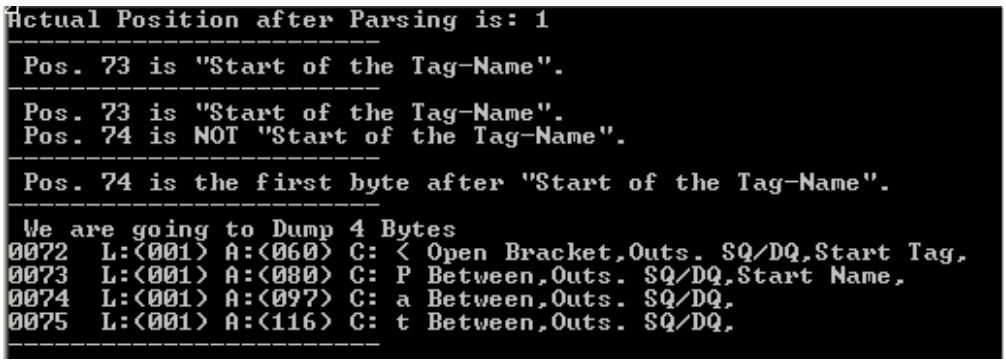
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Start Name
$$TXT=Start of the Tag-Name
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|10|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

This command needs the XML-Data to be parsed first, using the
par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **lf - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.6.21 stt - System Tag

MiniRobotLanguage (MRL)

IML.System Tag
IML.stt
IML.System TagZ
IML.sttz

Find out if the specified XML-Flag is set.

Intention

This command is used to analyze XML-Flags. Using this command, you can find out if the actual XML-Data-Byte is the < or the > of a System Tag.

System Tags look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
```

To understand this better, see the XML-Sample-Data and XML-Flags-Dump here:
! Sample-XML Data 3966

There are 6 ways of using this command, depending on the number of Parameters.

```
' With no Parameter, IML. will use the current position
XML.Set Pos|1
IML.stt
  PRT. Pos. 1 flagged.
EIF.

' With One parameter IML. will check just the given Byte
IML.btw|30
  PRT. Pos. 30 is flagged.
EIF.

' With two parameters, IML: will start searching for a Bate
with
' the specified Flag set, and if one is found, place that
' Bytes Number on the TOS (Top of Stack).
IML.stt|20|40
  PRT. Pos. $$000 is flagged.
EIF.
```

Additionally you can suffixe in all three cases above the subcommand with a "z".
This will change the internal search to "***Bit not set***" (Z= Bit=Zero).

```
' Now we are scanning for "Bit not set"
IML.sttz|30|40
  PRT. Pos. $$000 is the first byte after that Flag is set.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
```

As always `NML.` is the negative Form of `IML.`

Syntax

IML.System Tag[|P2][|P3]
IML.stt[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** ***Variable*** with the Position, where to test the XML-Flag.
If **P3** is specified, then this is the Start-Byte to scan at.
If the value in P3<P2 then the Scan is done in reverse order.

P3 - ***(optional)*** ***Variable*** with the ending position for the scan.
If **P3** is specified, then P3 is the End-Byte to scan.
If the value in P3<P2 then the Scan is done in reverse order.

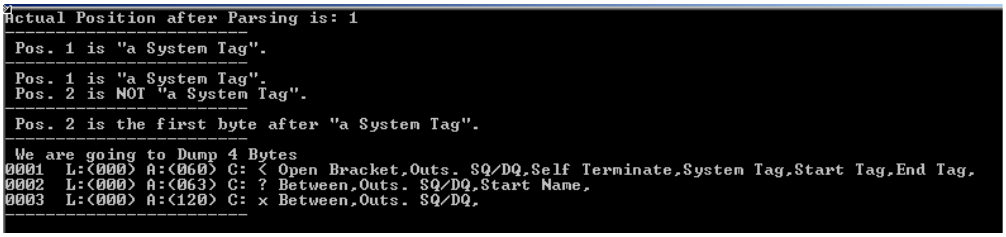
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=System Tag
$$TXT=a System Tag
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
```

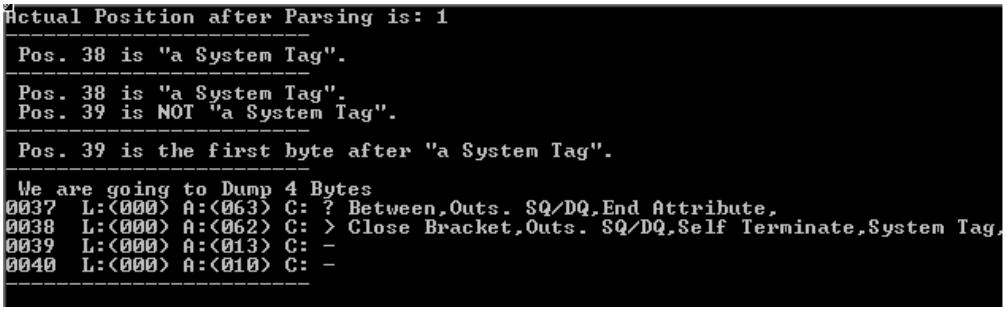

© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here you can see that also the > is marked as system Tag. This way you can recognize the end of a system tag more easily.



Remarks

This command needs the XML-Data to be parsed first, using the

© 2023 Theo Gottwald * https://www.smart-package.com

par - parse from var 4117
or the
If - load file 4113
Command.
In this process the XML-Parser builds a Table with flags that is analyzed using this commands. You can see a XML-Dump of such XML-Data Flags here:
! Sample-XML Data 3966

Limitations:

-

See also:

- !! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **If - load file** 4113
- **stt - System Tag** 4091
- **stn - Start Name** 4087
- **stm - Self Terminate** 4083
- **stg - Start Tag** 4079
- **sta - Start Attribute** 4075
- **osd - Outside SDQ** 4071
- **obt - Open Bracket** 4067
- **isq - Is SQ** 4063
- **idq - Is DQ** 4059
- **etg - End Tag** 4055
- **edn - End Name** 4051
- **eat - End Attribute** 4047
- **cmt - Comment** 4042
- **cbt - Close Bracket** 4038
- **btw - Between** 4034
- **bsq - Between SQ** 4030
- **beq - Before EQ** 4026
- **bdq - Between DQ** 4022
- **afz - Any Flags Zero** 4016
- **afs - Any Flags Set** 4010
- **aeq - After EQ** 4006

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7 XML - no parse

3.84.7.1 Clr - Clear

MiniRobotLanguage (MRL)

XML.Clear
XML.clr

Clear all internal XML-Data and "undo" XML.-Parsing

Intention

This command is used to clear all internal XML-Data from a previously parsed XML-file and return the memory to the system.
During the loading/parsing of XML-Data, the SPR collects internal data-tables and flags to speed up the later commands.

```
XML.Clear
```

Deletes all collected Data and puts the SPR into the state right before parsing any XML-Data from new. It will also delete all Error-Messages, therefore you do not need to call

```
XML.Clear Error
```

explicitly after this command. If you load or parse a new XML-File, this command is transparently executed, therefore you do not need to call it before parsing an new XML-File.

In most cases, you do not need to call this command as its automatically called.

Even if the script ends, the robot will automatically free all memory automatically. Therefore this command is not really needed in most cases.

Syntax

XML.Clear

Parameter Explanation

-

Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'**********************************
' XML.Clear Sample
'**********************************
$$XMF=?pfad\Small_XML.txt
' We load the XML-Data into a Vaariable
CFF.$$XMF|$$XML
' Now we are going to parse it
XML.Parse|$$XML
'XML.Dump
' We check if there are errors
JIT.Lab_Err

' Now we clear the parsing.
XML.Clear

' We try an XML.-Command that will of course run into an
error.
XML.get pos|$$POS
' We jump to the error-handler
JIT.Lab_Err
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
MBX.!
'-------------------------------------------------------------
ENR.
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
[11:35:15] [33]
XML.-Error: Before you can use XML-Commands, you need to load or parse an valid XML-File.
[11:35:15] MRE: Robot ended: XML_Clear.mrt
```

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990

© 2023 Theo Gottwald * https://www.smart-package.com

- par - parse from var 4117
- If - load file 4113
- cle - clear error 4106

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.2 UnClean - UnClean Text

MiniRobotLanguage (MRL)

XML.UnClean Text
XML.UnClean

Prepare Textual Data for inclusion in XML-Documents

Intention

This There are special characters that are not valid inside XML-Documents, because they have a special meaning. You find them below.

In XML-Files, we have data that is enclosed and described using "tags".
We have "Start-Tags" like `<note>`
and we have "End-Tags" like `</note>`
between these we have data that can itself be XML or other data.

Because of this, there are some characters that should not be used in the XML-data, because they are reserved for tags. These characters are:

& - should be replace by &
< - should be replace with <
> - should be replace with >
" - should be replace with "
' - should be replace with &spos;

For this purpose the `XML.` command has the "`Clean`"-Option.

It will prepare your data to be valid as XML-Data.

And if you want your data back to the original state, just use the "`unclean`" option.

Here is an example:

```
' Prepare Data for use in XML-Files
$$XMD=>&XMLData>&<<>
XML.clean|$$XMD|$$XMO
DBP. the ready for use XML-Data is in $$XMO
$$XMD=
XML.unclean|$$XMO|$$XMD
DBP.$$XMD contains the original data
END.
```

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

XML.UnClean Text|P2[|P3]

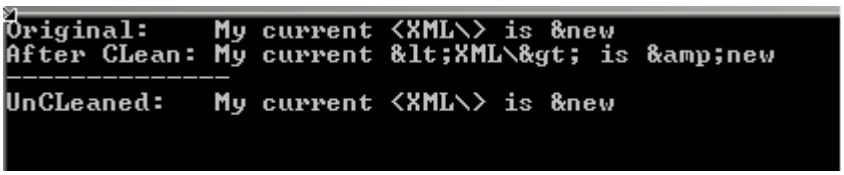
Parameter Explanation

P2 – Variable with XML-Data that is going to be cleaned.
P3 – Variable to return the result. If omitted the result is placed on TOS.

Example

```
'*********************************
' XML.-Clean/UnClean Sample
'*********************************
'
$$XMF=My current <XML\> is &new
' Now we are going to parse it
XML.Clean|$$XMF|$$XMC
XML.UnClean|$$XMC|$$XMU
PRT.Original:    $$XMF
PRT.After CLean: $$XMC
PRT.--------------
PRT.UnCLeaned:   $$XMU
MBX.!
ENR.
```

This is the result of the sample script:



Remarks

Unlike other commands, XML.Clean and XML.UnCLean can be used without the need to first parse or load an XML-File. They are basically String/Text Replace-Operations.

Limitations:

-

See also:

- ! Smarty's XML-Features 3973

© 2023 Theo Gottwald * https://www.smart-package.com

- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **clean - Clean Text** 4108

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.7.3 Wrap

MiniRobotLanguage (MRL)

XML.Wrap

Wrap a text into <>

Intention

This command is used to generate an XML-Tag.

As Input you give it any Text, as the result you will get an XML-Tag.
Wrap is intelligent, look what it really does.

```
' We do a simple Wrap
$$XMF=Hallo
XML.Wrap|$$XMF
PRT.1. Wrapped: $$XMF
PRT.------------------------
' We do a more complex WRAP
$$XMF=<Hallo
XML.Wrap|$$XMF
PRT.2. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the space left of the
<)
$$XMF= <Hallo
XML.Wrap|$$XMF
PRT.3. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the spaces left of
the <)
$$XMF=   <<Hallo
XML.Wrap|$$XMF
PRT.4. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$<Hallo>$sp$
XML.Wrap|$$XMF
PRT.5. Wrapped: $$XMF
PRT.------------------------
' We do a maximum complex WRAP (Notice the space right of the
>)
VAR.$$XMF=$sp$   <<<Hallo>>>    $sp$
XML.Wrap|$$XMF
PRT.6. Wrapped: $$XMF
PRT.------------------------
```

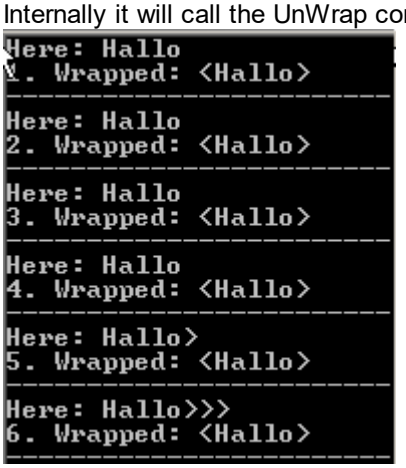
As you can see in the Output below, the Wrap command is intelligent in what its doing.
It will not "just place <> around".

© 2023 Theo Gottwald * https://www.smart-package.com

It will first check

- does the text need to be trimmed?
- are there already brackets around?
- Whitespaces inside the Tag are preserved
- Whitespaces before the < or
- after the > are trimmed
- it will trim all XML Whitespaces and Tabs, these are ASCII:
  `&H20,&H9,&HD,&HA`

Internally it will call the UnWrap command first and then Wrap the text.



Syntax

XML.Wrap|P2[|P3]

Parameter Explanation

P2 – Variable with XML-Data that is to be Wrapped into <>.
If P3 is omitted, the result it stored into this variable.

P3 – ***(Optional)*** Variable to return the result.

Example

```
'***********************************
' XML.Warp/Unwrap Sample
'***********************************
' We do a simple Wrap
$$XMF=Hallo
XML.Wrap|$$XMF
PRT.1. Wrapped: $$XMF
PRT.------------------------
' We do a more complex WRAP
$$XMF=<Hallo
XML.Wrap|$$XMF
PRT.2. Wrapped: $$XMF
PRT.------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

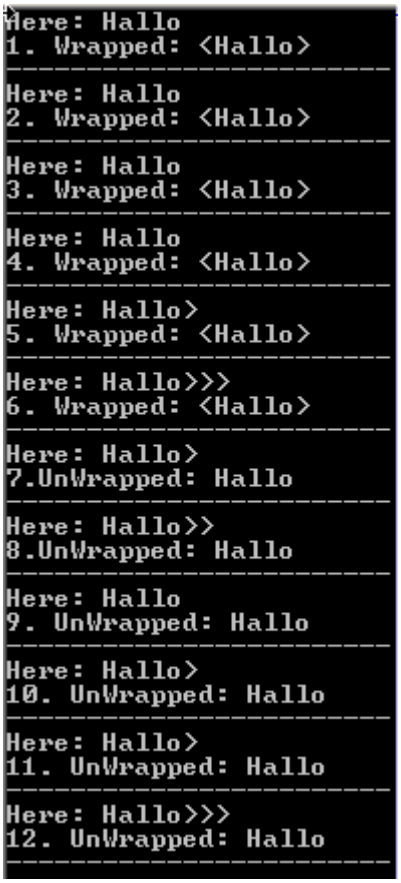
```
' We do a even more complex WRAP (Notice the space left of the
<)
$$XMF= <Hallo
XML.Wrap|$$XMF
PRT.3. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the spaces left of
the <)
$$XMF=   <<Hallo
XML.Wrap|$$XMF
PRT.4. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$<Hallo>$sp$
XML.Wrap|$$XMF
PRT.5. Wrapped: $$XMF
PRT.------------------------
' We do a maximum complex WRAP (Notice the space right of the
>)
VAR.$$XMF=$sp$   <<<Hallo>>>    $sp$
XML.Wrap|$$XMF
PRT.6. Wrapped: $$XMF
PRT.------------------------
' We do a simple UnWrap
$$XMF=Hallo>
XML.UnWrap|$$XMF
PRT.7.UnWrapped: $$XMF
PRT.------------------------
' We do a more complex UnWRAP
$$XMF=<Hallo>>
XML.UnWrap|$$XMF
PRT.8.UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the space left of
the <)
$$XMF= <Hallo
XML.UnWrap|$$XMF
PRT.9. UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the spaces left of
the <)
$$XMF=   <<Hallo>
XML.UnWrap|$$XMF
PRT.10. UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$<Hallo>$sp$
XML.UnWrap|$$XMF
PRT.11. UnWrapped: $$XMF
PRT.------------------------
' We do a maximum complex UnWRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$   <<<Hallo>>>    $sp$
XML.UnWrap|$$XMF
PRT.12. UnWrapped: $$XMF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.------------------------
MBX.!
ENR.
```

below is the result of the sample script:

You can see that the Wrap and the UnWrap commands are smart and deliver you exactly what you need.



Remarks

Wrap does not automatically "Clean" the Text, if needed use the "`XML.Clean`" Command first.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **unw - UnWrap** 4123
- **clean - Clean Text** 4108
- **UnClean - UnClean Text** 4098

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.4 cle - clear error

MiniRobotLanguage (MRL)

XML.Clear Error
XML.cle

Clear internal Error flags

Intention

This command is used to clear all internal XML.-Error-Flags.

Usage is:

```
XML.Clear Error
```

Typically it can be used after the Error has been handled.

When loading or parsing a new XML-File, all Error Flags are automatically cleared, therefore no need to use this command for initilization.

Syntax

XML.Clear Error

Parameter Explanation

-

Example

```
''''''''''''''''''''''''''''''''''''
' XML.Clear Error-Sample
'
''''''''''''''''''''''''''''''''''''
$$XMF=?pfad\Small_XMLx.txt
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
:enx
MBX.!
ENR.
'------------------------------------------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' Error Handler
:Lab_Err
' Get Error
XML.get error|$$ERT
PRT.$$ERT
' Clear Error
XML.cle
' Error is cleared
XML.get error|$$ERT
PRT. Error cleared! $$ERT
GTO.enx
ENR.
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **geterr - get error** 4111
- **Clr - Clear** 4095
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.5 clean - Clean Text

MiniRobotLanguage (MRL)

XML.Clean Text
XML.Clean

Prepare Textual Data for inclusion in XML-Documents

Intention

This There are special characters that are not valid inside XML-Documents, because they have a special meaning. You find them below.

In XML-Files, we have data that is enclosed and described using "tags".
We have "Start-Tags" like `<note>`
and we have "End-Tags" like `</note>`
between these we have data that can itself be XML or other data.

Because of this, there are some characters that should not be used in the XML-data, because they are reserved for tags. These characters are:

& - should be replace by &
< - should be replace with <
> - should be replace with >
" - should be replace with "
' - should be replace with &spos;

For this purpose the `XML.` command has the "`Clean`"-Option.

It will prepare your data to be valid as XML-Data.

And if you want your data back to the original state, just use the "`unclean`" option.

Here is an example:

```
' Prepare Data for use in XML-Files
$$XMD=>&XMLData>&<<>
XML.clean|$$XMD|$$XMO
DBP. the ready for use XML-Data is in $$XMO
$$XMD=
XML.unclean|$$XMO|$$XMD
DBP.$$XMD contains the original data
END.
```

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

XML.Clean Text|P2[|P3]

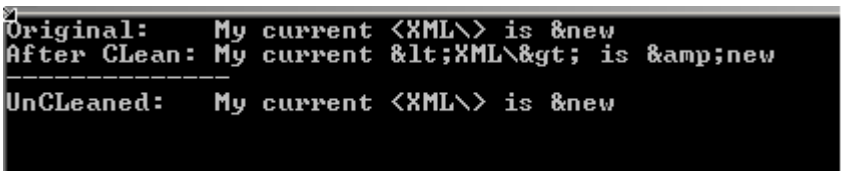
Parameter Explanation

P2 – Variable with XML-Data that is going to be cleaned.
P3 – Variable to return the result. If omitted the result is placed on TOS.

Example

```
'*********************************
' XML.-Clean/UnClean Sample
'*********************************
'
$$XMF=My current <XML\> is &new
' Now we are going to parse it
XML.Clean|$$XMF|$$XMC
XML.UnClean|$$XMC|$$XMU
PRT.Original:    $$XMF
PRT.After CLean: $$XMC
PRT.-------------
PRT.UnCLeaned:   $$XMU
MBX.!
ENR.
```

This is the result of the sample script:



Remarks

Unlike other commands, XML.Clean and XML.UnCLean can be used without the need to first parse or load an XML-File. They are basically String/Text Replace-Operations.

Limitations:

-

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **UnClean - UnClean Text** 4098

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.6 geterr - get error

MiniRobotLanguage (MRL)

XML.Get Error
XML.geterr

Read internal Error flags to get informations about parsing Errors

Intention

This command is used to read all internal XML.-Error-Flags.
As the XML-Parser does not display Error-Messages, you need to use `JIT.` and this commands to find out about possible parsing problems.

It does only read Errors that happen during XML.-Parsing.
Usage is:

```
XML.Get Error|$$ERT
```

Typically it can be used during Error handling.

When loading or parsing a new XML-File, all Error Flags are automatically cleared.

Below is a complete example Script that shows Error handling.

Syntax

XML.Get Error|P2[|P3]

Parameter Explanation

P2 – Variable to return the Error-Messages that happened during the execution of an XML.-Command. Messages are stacked and collected until you use `XML.Clear Error.`

P3 – Variable to return a Error-Number. Currently not fully implemented.

Example

```
''''''''''''''''''''''''''''''''''''
' XML.Get Error/Clear Error-Sample
'
''''''''''''''''''''''''''''''''''''
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$XMF=?pfad\Small_XMLx.txt
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
:enx
MBX.!
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
' Get Error
XML.get error|$$ERT
PRT.$$ERT
' Clear Error
XML.cle
' Error is cleared
XML.get error|$$ERT
PRT. Error cleared! $$ERT
GTO.enx
ENR.
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **cle - clear error** 4106

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.7 If - load file

MiniRobotLanguage (MRL)

XML.Load File
XML.Load
XML.lf

Get XML-Data from a file

Intention

This command is used to Load XML-Data from a file. During execution of this command, the XML-Parser will "Interpret" the XML-Data and decide

- Where Tags Start and End
- where Attributes start and end
- where names of tags and attributes start and end

It will then internally create a internal set of flags that is a 1:1 description of the interpreted XML-Data. Based on these flags all the complex XML.-Subcommands do their work.

Therefore most of the XML.-Subcommands require a XML-File to be loaded or Parsed prior they can set to work.
Note that the Load-Command will transparently call the Parse command, after having first load the XML-File into internal Memory.

Usage is very simple:

```
$$XMF=?pfad\Small_XML.txt
' Now we are going to load and parse it
XML.Load File|$$XMF
```

We can find out if we had errors during parsing using the TImeout-Flag.
Or we can add a third Parameter that will be zero if there are any errors.

```
XML.Load File|$$XMF|$$RES
```

If the Parsing was successfiul, `$$RES` will contain a value for `XML-Root`,
that is the Byte-Adress of the first "<" in the XML-Data. It will always be larger then zero.

If the result `$$RES` is zero, then the parsing was not successful. In this case you may find an Parsing Error on the TOS (To of Stack).

Always make sure that the file you are going to load, really exists.

You can use the
IEF. / NEF. - If - exist - File 1050
Query for this.

© 2023 Theo Gottwald * https://www.smart-package.com

```
IEF.$$FIL
$$XMF=?pfad\Small_XML.txt
  ' Now we are going to load and parse it
  XML.Load File|$$FIL
ELS.
  PRT.File $$FIL not found.
EIF.
```

Syntax

XML.Load|P2[|P3]

Parameter Explanation

P2 - Variable with XML-Data. XML-Data can contain $cr, $lf or not.
P3 - Variable to return the result. The result is 0 in case of parsing errors. Otherwise the result is >0 and contains the value XML_Root.

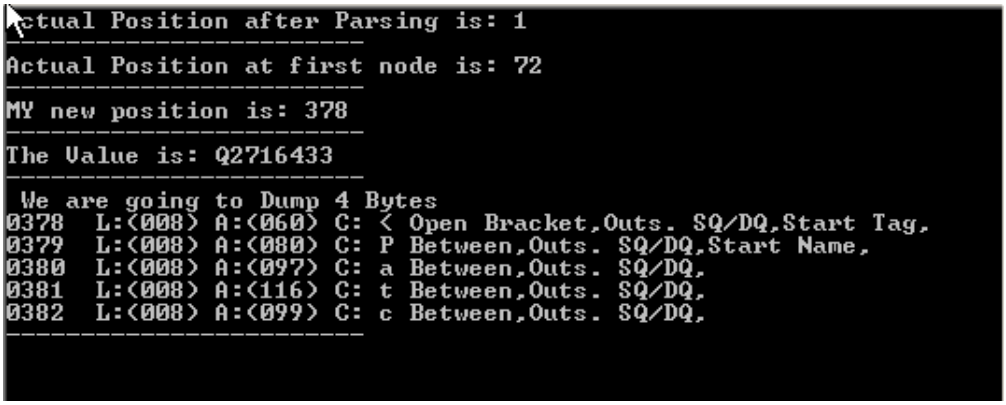
Example

```
'**********************************
' XML.Load-Sample
' We are looking for the Tag below, extracting an attribute-value
' "SQNumber".
' <Patch
'  PatchName="SQLServer2008-KB2716433-x86.exe"
SQNumber="Q2716433">
'**********************************
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
XML.goto first node
XML.get pos|$$POS
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.Actual Position at first node is: $$POS
PRT.-----------------------
' We use a Hi-Level Search here.
XML.search name and attribute|{&EXACT:Patch}|SQNumber
$$REA=$$000
PRT.MY new position is: $$REA
PRT.-----------------------
XML.attrib value by name|SQNumber|$$RES
PRT.The Value of the attribute is: $$RES
PRT.-----------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|$$REA|($$REA+4)
PRT.-----------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
<Product ProductName="WINDOWS NT 4.0" SName="NTW">
<SPS>
<SP SPName="SP3"/>
<SP SPName="SP4"/>
<SP SPName="SP5"/>
<SP SPName="SP6"/>
<SP SPName="SP6A"/>
</SPS>
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe"
SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
<SURLS>
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.8 par - parse from var

MiniRobotLanguage (MRL)

XML.Parse from Var
XML.Parse
XML.Par

Get XML-Data that is already inside a Variable

Intention

This command is used to Load XML-Data from a Variable. During execution of this command, the XML-Parser will "Interpret" the XML-Data and decide

- Where Tags Start and End
- where Attributes start and end
- where names of tags and attributes start and end

It will then internally create a internal set of flags that is a 1:1 description of the interpreted XML-Data. Based on these flags all the complex XML.-Subcommands do their work.

Therefore most of the XML.-Subcommands require a XML-File to be loaded or Parsed prior they can set to work.
Note that the Load-Command will transparently call the Parse command, after having first load the XML-File into internal Memory.

Usage is very simple:

```
$$XMF=?pfad\Small_XML.txt
' We load the XML-Data into a Vaariable
CFF.$$XMF|$$XML
' Now we are going to parse it
XML.Parse|$$XML
```

We can find out if we had errors during parsing using the TImeout-Flag.
Or we can add a third Parameter that will be zero if there are any errors.

```
XML.Parse|$$XML|$$RES
```

If the Parsing was successful, `$$RES` will contain a value for `XML-Root`,
that is the Byte-Adress of the first "<" in the XML-Data. It will always be larger then zero.

If the result `$$RES` is zero, then the parsing was nut successful. In this case you may find an Parsing Error on the TOS (To of Stack).

© 2023 Theo Gottwald * https://www.smart-package.com

Syntax

XML.Parse|P2[|P3]

P2 - Variable with XML-Data. XML-Data can contain $cr, $lf or not.
P3 - Variable to return the result. The result is 0 in case of parsing errors. Otherwise the result is >0 and contains the value XML_Root.

Parameter Explanation

-

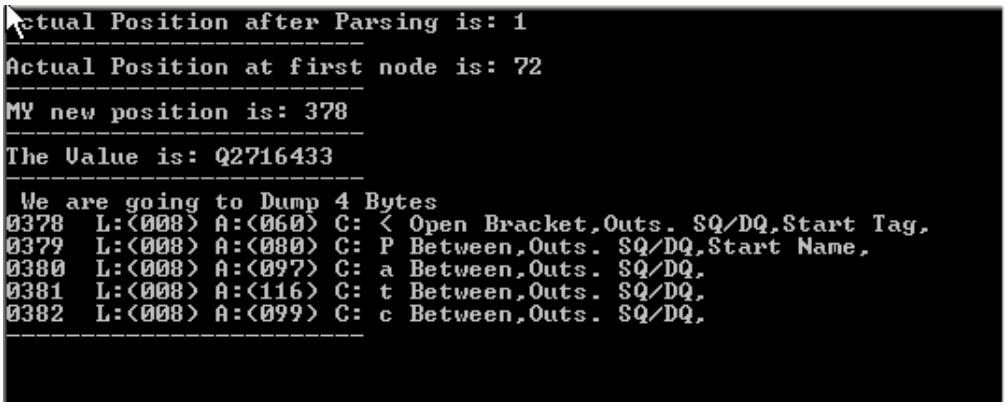
Example

```
'***********************************
' XML.Parse.-Sample
' We are looking for the Tag below, extracting an attribute-value
' "SQNumber".
' <Patch
'  PatchName="SQLServer2008-KB2716433-x86.exe" SQNumber="Q2716433">
'***********************************
$$XMF=?pfad\Small_XML.txt
' We load the XML-Data into a Vaariable
CFF.$$XMF|$$XML
' Now we are going to parse it
XML.Parse|$$XML
'XML.Dump
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
XML.goto first node
XML.get pos|$$POS
PRT.Actual Position at first node is: $$POS
PRT.------------------------
' We use a Hi-Level Search here.
XML.search name and attribute|{&EXACT:Patch}|SQNumber
$$REA=$$000
PRT.MY new position is: $$REA
PRT.------------------------
XML.attrib value by name|SQNumber|$$RES
PRT.The Value of the attribute is: $$RES
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|$$REA|($$REA+4)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.------------------------
MBX.!
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
<Product ProductName="WINDOWS NT 4.0" SName="NTW">
<SPS>
<SP SPName="SP3"/>
<SP SPName="SP4"/>
<SP SPName="SP5"/>
<SP SPName="SP6"/>
<SP SPName="SP6A"/>
</SPS>
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe"
SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **If - load file** 4113

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.9 spl - Split Attrib

MiniRobotLanguage (MRL)

XML.Split Attrib
XML.spl

Split a String (Attribute) into "Name" and Value"

Intention

This command is used to easily split an attribute String into its components, name and value. This is a pure String-Operation and does not need a previous parsing.

```
'Usage is simple:
$$XMF=encoding="UTF-8?"
' We are going to split this in Name and Value
XML.Split Attrib|$$XMF|$$NAM
DBP.$$NAM--$$000
'-------------------------------------------------------------
XML.Split Attrib|$$XMF|$$NAM|$$VAL
DBP.$$NAM--$$VAL
ENR.
```

Syntax

XML.spl|P2[|P3][|P4]

Parameter Explanation

P2 – Variable with XML-Data. XML-Data can contain $cr, $lf or not.

P3 – Variable to return the result. The result is 0 in case

P4 – Variable to return the result value. If omitted this value is placed on TOS (Top of Stack).

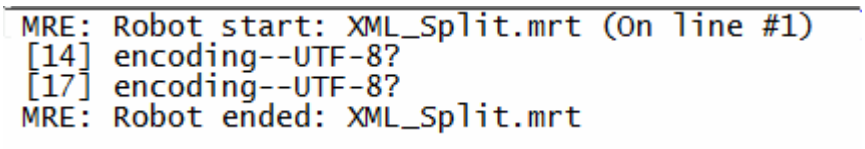
Example

```
'****************************
'  XML.-Split Attrib - Sample
'****************************
$$XMF=encoding="UTF-8?"
' We are going to split this in Name and Value
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
XML.Split Attrib|$$XMF|$$NAM
DBP.$$NAM--$$000
'-------------------------------------------------------------
XML.Split Attrib|$$XMF|$$NAM|$$VAL
DBP.$$NAM--$$VAL
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.7.10 unw - UnWrap

MiniRobotLanguage (MRL)

XML.UnWrap

UnWrap a tag / remove the <>

Intention

This command is used to extract the inside from an XML-Tag.
In short this command will remove the < > from the tag.

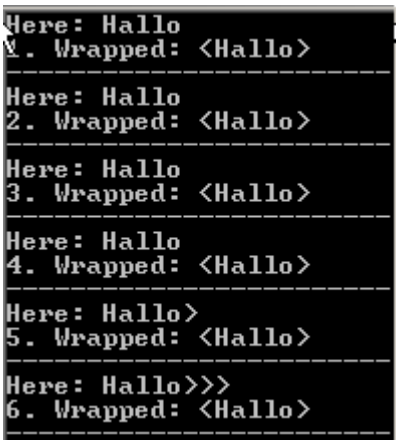
As Input you give it any Tag, as the result you will get an Text.
UnWrap is intelligent, look what it really does.

```
' We do a simple UnWrap
$$XMF=Hallo>
XML.UnWrap|$$XMF
PRT.7.UnWrapped: $$XMF
PRT.------------------------
' We do a more complex UnWRAP
$$XMF=<Hallo>>
XML.UnWrap|$$XMF
PRT.8.UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the space left of
the <)
$$XMF= <Hallo
XML.UnWrap|$$XMF
PRT.9. UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the spaces left of
the <)
$$XMF=   <<Hallo>
XML.UnWrap|$$XMF
PRT.10. UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$<Hallo>$sp$
XML.UnWrap|$$XMF
PRT.11. UnWrapped: $$XMF
PRT.------------------------
' We do a maximum complex UnWRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$   <<<Hallo>>>    $sp$
XML.UnWrap|$$XMF
PRT.12. UnWrapped: $$XMF
PRT.------------------------
MBX.!
ENR.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

As you can see in the Output below, the UnWrap command is intelligent in what its doing.
It will not "just remove any <>.
It will first check

- does the text need to be trimmed?
- are there brackets around?
- Whitespaces inside the Tag are preserved
- Whitespaces before the < or
- after the > are trimmed
- it will trim all XML Whitespaces and Tabs, these are ASCII:
  `&H20,&H9,&HD,&HA`



Syntax

XML.UnWrap|P2[|P3]

Parameter Explanation

P2 – Variable with XML-Data that is to be UnWrapped from the <>.
If P3 is omitted, the result it stored into this variable.
P3 – ***(Optional)*** Variable to return the result.

Example

```
'***********************************
' XML.Warp/Unwrap Sample
'***********************************
' We do a simple Wrap
$$XMF=Hallo
XML.Wrap|$$XMF
PRT.1. Wrapped: $$XMF
PRT.------------------------
' We do a more complex WRAP
$$XMF=<Hallo
```


© 2023 Theo Gottwald * https://www.smart-package.com

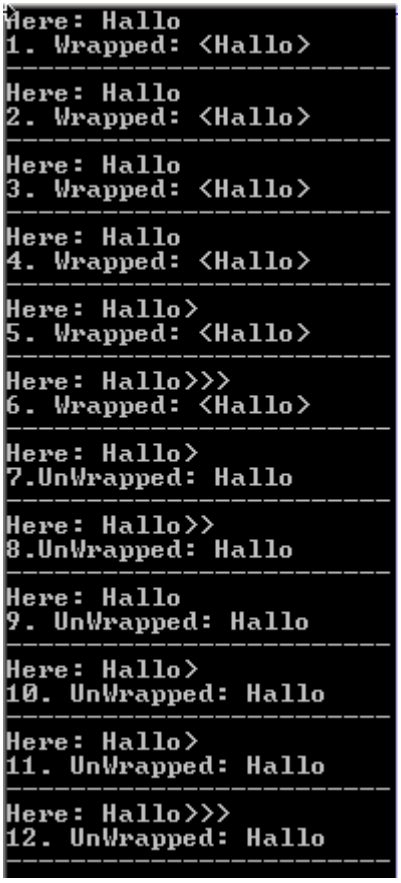
```
XML.Wrap|$$XMF
PRT.2. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the space left of the
<)
$$XMF= <Hallo
XML.Wrap|$$XMF
PRT.3. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the spaces left of
the <)
$$XMF=   <<Hallo
XML.Wrap|$$XMF
PRT.4. Wrapped: $$XMF
PRT.------------------------
' We do a even more complex WRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$<Hallo>$sp$
XML.Wrap|$$XMF
PRT.5. Wrapped: $$XMF
PRT.------------------------
' We do a maximum complex WRAP (Notice the space right of the
>)
VAR.$$XMF=$sp$   <<<Hallo>>>    $sp$
XML.Wrap|$$XMF
PRT.6. Wrapped: $$XMF
PRT.------------------------
' We do a simple UnWrap
$$XMF=Hallo>
XML.UnWrap|$$XMF
PRT.7.UnWrapped: $$XMF
PRT.------------------------
' We do a more complex UnWRAP
$$XMF=<Hallo>>
XML.UnWrap|$$XMF
PRT.8.UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the space left of
the <)
$$XMF= <Hallo
XML.UnWrap|$$XMF
PRT.9. UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the spaces left of
the <)
$$XMF=   <<Hallo>
XML.UnWrap|$$XMF
PRT.10. UnWrapped: $$XMF
PRT.------------------------
' We do a even more complex UnWRAP (Notice the space right of
the >)
VAR.$$XMF=$sp$<Hallo>$sp$
XML.UnWrap|$$XMF
PRT.11. UnWrapped: $$XMF
PRT.------------------------
' We do a maximum complex UnWRAP (Notice the space right of
the >)
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
VAR.$$XMF=$sp$    <<<Hallo>>>     $sp$
XML.UnWrap|$$XMF
PRT.12. UnWrapped: $$XMF
PRT.------------------------
MBX.!
ENR.
```

below is the result of the sample script:

You can see that the Wrap and the UnWrap commands are smart and deliver you exactly what you need.



Remarks

Wrap does not automatically "`UnClean`" the Text, if needed use the "`XML.UnClean`" Command first.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **Wrap** 4101
- **clean - Clean Text** 4108
- **UnClean - UnClean Text** 4098

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8 XML. - after parse

3.84.8.1 abm - Attrib by Number

XML.Attrib by Number **Previous** 4123 **Top** 41 **Next** 4131

MiniRobotLanguage (MRL)

XML.Attrib by Number
XML.abm

Get complete attributes from an XML-Tag "by number"

Intention

This command is used to retrieve an attribute from an XML-Data Tag.

```
<Product ProductName="STATION 10" Name="Value">
```

This Tag has two attributes:

Number 1:
`ProductName="STATION 10"`

and Number 2:

```
Name="Value"
```

You could retrieve then using

```
XML.abm|1|$$RES
```

to get Attribute Nr.1 into `$$RES`

or

```
XML.abm|2|$$RES
```

to get Attribute Nr.2 into `$$RES`

To Split the attributes you can use the

```
XML.spl|$$ERG|$$NAM|$$VAL
PRT.It can be splitted into: $$NAM and $$VAL
```

command.

After this command, the actual position is on the attributes End.
To get back to the start of the attribute, use:

```
XML.siu|start attribute
```

This will set the actual position to the start of the attribute.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

XML.Attrib by Number|P2[|P3]

Parameter Explanation

P2 – Variable with the number of the attribute that you want.

P3 – ***(optional)*** Variable to return the result. The result is "" in case no such attribute was found. If P3 was omitted the result is placed on TOS.

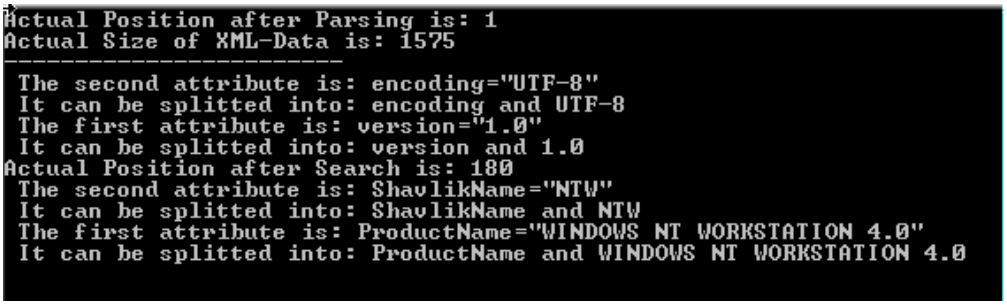
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of XML-Data is: $$LEN
' Hier gehts los
PRT.------------------------
XML.abm|2|$$ERG
PRT. The second attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.abm|1|$$ERG
PRT. The first attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.sdn|{&EXACT:Product}
XML.get pos|$$POS
PRT.Actual Position after Search is: $$POS
XML.abm|2|$$ERG
PRT. The second attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.abm|1|$$ERG
PRT. The first attribute is: $$ERG
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here is a XML-Dump of the used XML-Data:

! Sample-XML Data 3966

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **abn - Attrib by Name** 4131
- **abv - Get Attrib by Value** 4135
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.2 abn - Attrib by Name

MiniRobotLanguage (MRL)

XML.Attrib by Name
XML.abn

Get complete attributes from an XML-Tag "by name"

Intention

This command is used to retrieve an attribute from an XML-Data Tag.

```
<Product ProductName="STATION 10" Name="Value">
```

This Tag has two attributes:

Number 1:
`ProductName="STATION 10"`

and Number 2:

```
Name="Value"
```

You could retrieve then using

```
XML.abm|productname|$$RES
```

to get Attribute Nr.1 into `$$RES`

or

```
XML.abm|&EXACT:Name|$$RES
```

to get Attribute Nr.2 into `$$RES`

As you can use "Parts of a Name" to specify the name of the attribute, simply writing "Name" would possibly bring you the first attribute instead of the second.

You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want. For example, you can use:

```
XML.abm|{Pro&NDTHEN:Name}&OR:PName|$$RES
```

to specify a Name of an attribute. THis way you can also specify alternative names, for exaple if you have to analyze different languages.

To Split the attributes you can use the

```
XML.spl|$$ERG|$$NAM|$$VAL
PRT.It can be splitted into: $$NAM and $$VAL
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

command.

After this command, the actual position is on the attributes End.
To get back to the start of the attribute, use:

```
XML.siu|start attribute
```

This will set the actual position to the start of the attribute.

Syntax

XML.Attrib by Name|P2[|P3][|P4]

Parameter Explanation

P2 - Variable with the name of the attribute that should be retrieved.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want.

P3 - ***(optional)*** Variable to return the result. The result is "" in case no such attribute was found. If P3 was omitted the result is placed on TOS.

P4 - ***(optional)*** Count. In case there are multiple variables with the same name, you
can specify the number of the attribute that you want. If omitted "1" is the default number.

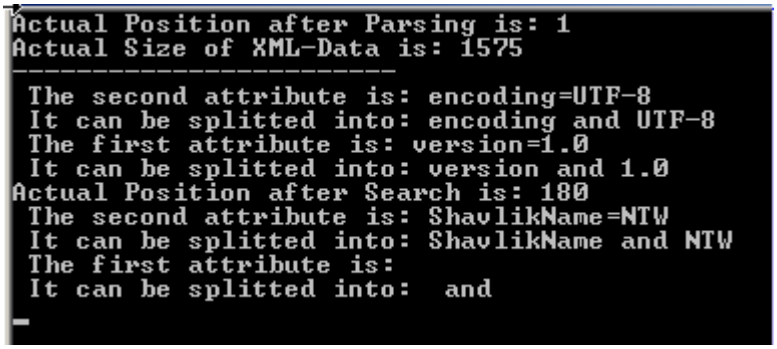
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
```

 © 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.Actual Size of XML-Data is: $$LEN
' Hier gehts los
PRT.------------------------
XML.abn|ncod&ANDTHEN:g|$$ERG
PRT. The second attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.abn|ver|$$ERG
PRT. The first attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.sdn|{&EXACT:Product}
XML.get pos|$$POS
PRT.Actual Position after Search is: $$POS
XML.abn|lik|$$ERG
PRT. The second attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.abm|uctn|$$ERG
PRT. The first attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **abm - Attrib by Number** 4128
- **abv - Get Attrib by Value** 4135
- **avbn - Attrib Value by Name** 4142

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.3 abv - Get Attrib by Value

MiniRobotLanguage (MRL)

XML.Get Attrib by Value
XML.abv

Get attribute value by specifying the attribute name

Intention

This command is used to retrieve an attribute from an XML-Data Tag.

```
<Product ProductName="STATION 10" Name="Value">
```

This Tag has two attributes:

Number 1:
`ProductName="STATION 10"`

and Number 2:

```
Name="Value"
```

You could retrieve then using

```
XML.abv|statio|$$RES
```

to get Attribute Nr.1 into `$$RES`

or

```
XML.abv|valu|$$RES
```

to get Attribute Nr.2 into `$$RES`

You can use "Parts of a Value" to specify the Value of the attribute that you want.

You can and need to use 3.2 Standard-Search Pattern 341 to specify the value of the attribute that you want. For example, you can use:

```
XML.abv|Va&AND:e|$$RES
```

to specify a Value of an attribute. This way you can also specify alternative Values, for exaple if you have to analyze different languages.

To Split the attributes you can use the

```
XML.spl|$$ERG|$$NAM|$$VAL
PRT.It can be splitted into: $$NAM and $$VAL
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

command.

After this command, the actual position is on the attributes End.
To get back to the start of the attribute, use:

```
XML.siu|start attribute
```

This will set the actual position to the start of the attribute.

Syntax

XML.Get abv|P2[|P3][|P4]

Parameter Explanation

P2 - Variable with the Value of the attribute that should be retrieved.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want.

P3 - ***(optional)*** Variable to return the result. The result is "" in case no such attribute was found. If P3 was omitted the result is placed on TOS.

P4 - ***(optional)*** Count. In case there are multiple variables with the same value, you
can specify the number of the attribute that you want. If omitted "1" is the default number.

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of XML-Data is: $$LEN
' Hier gehts los
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.------------------------
XML.abv|utf|$$ERG
PRT. The second attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.abv|1.0|$$ERG
PRT. The first attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.sdn|{&EXACT:Product}
XML.get pos|$$POS
PRT.Actual Position after Search is: $$POS
XML.abv|NTW|$$ERG
PRT. The second attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
XML.abv|work&AND:4|$$ERG
PRT. The first attribute is: $$ERG
XML.spl|$$ERG|$$NAM|$$VAL
PRT. It can be splitted into: $$NAM and $$VAL
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
Actual Size of XML-Data is: 1575
------------------------
 The second attribute is: encoding=UTF-8
 It can be splitted into: encoding and UTF-8
 The first attribute is: version=1.0
 It can be splitted into: version and 1.0
Actual Position after Search is: 180
 The second attribute is: ShavlikName=NTW
 It can be splitted into: ShavlikName and NTW
 The first attribute is: ProductName=WINDOWS NT WORKSTATION 4.0
 It can be splitted into: ProductName and WINDOWS NT WORKSTATION 4.0
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **abm - Attrib by Number** 4128
- **abn - Attrib by Name** 4131

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.4 acnt - Get Attrib Count

MiniRobotLanguage (MRL)

XML.Get Attrib Count
XML.acnt

Get number of attributes

Intention

This command is used to retrive the number of attributes of the Tag that is at the actual position. The actual position is the position that you can set using any search command or

```
XML.set pos|$$POS
```

Usage is simple:

```
XML.acnt|$$ERG
PRT. Here we have $$ERG attributes
```

Syntax

XML.Get Attrib Count[|P2]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result. If omitted, the result is been placed on TOS.

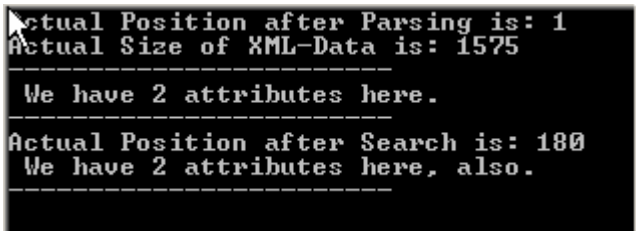
Example

```
'*********************************
'
'*********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of XML-Data is: $$LEN
' Hier gehts los
PRT.------------------------
XML.acnt|$$ERG
PRT. We have $$ERG attributes here.
PRT.------------------------
XML.sdn|{&EXACT:Product}
XML.get pos|$$POS
PRT.Actual Position after Search is: $$POS
XML.acnt|$$ERG
PRT. We have $$ERG attributes here, also.
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981

© 2023 Theo Gottwald * https://www.smart-package.com

- XML. - eXtended Markup Language 3990
- par - parse from var 4117
- abn - Attrib by Name 4131
- abv - Get Attrib by Value 4135
- abm - Attrib by Number 4128
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.5 avbn - Attrib Value by Name

MiniRobotLanguage (MRL)

XML.Attrib Value by Name
XML.avbn

Get an attributes value by specifying the attributes name

Intention

This command is used to retrieve an attributes value from an XML-Data Tag.
For this you have to set the current position to the Tag that has these attributes, that you want. Then you can use this command.

Here is an example for an XML-Tag with two attributes.

```
<Product ProductName="STATION 10" Name="Value">
```

This Tag has two attributes:

Number 1:
`ProductName="STATION 10"`

and Number 2:
`Name="Value"`

You could retrieve these attributes values using

```
XML.avbn|productname|$$RES
```

to get the value of Attribute Nr.1 into `$$RES`

or

```
XML.avbn|&EXACT:Name|$$RES
```

to get the value of Attribute Nr.2 into `$$RES`

As you can use "Parts of a Name" to specify the name of the attribute, simply writing "Name" would possibly bring you the value of the first attribute instead of the second.

You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want. For example, you can use:

```
XML.avbn|{Pro&NDTHEN:Name}&OR:PName|$$RES
```

to specify a Name of an attribute. This way you can also specify alternative names, for example if you have to analyze different languages.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Syntax

XML.avbn|P2[|P3][|P4]

Parameter Explanation

P2 - Variable with the name of the attribute that should be retrieved.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want.

P3 - ***(optional)*** Variable to return the result. The result is "" in case no such attribute was found. If P3 was omitted the result is placed on TOS.

P4 - ***(optional)*** Count. In case there are multiple variables with the same name, you
can specify the number of the attribute that you want. If omitted "1" is the default number.

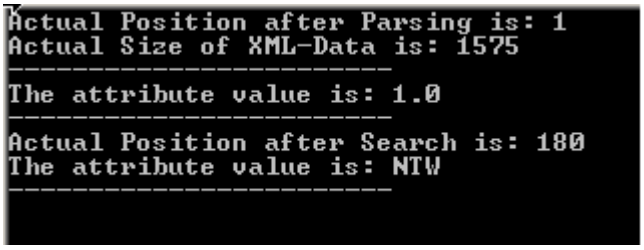
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of XML-Data is: $$LEN
'Here we start
PRT.------------------------
XML.avbn|ver|$$ERG
PRT.The attribute value is: $$ERG
PRT.------------------------
XML.sdn|{&EXACT:Product}
XML.get pos|$$POS
PRT.Actual Position after Search is: $$POS
XML.avbn|Sha|$$ERG
PRT.The attribute value is: $$ERG
PRT.------------------------
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
<Product ProductName="WINDOWS NT 4.0" SName="NTW">
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **abn - Attrib by Name** 4131
- **abv - Get Attrib by Value** 4135

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.6 dec pos - Decrement Position

MiniRobotLanguage (MRL)

XML.Decrement Position
XML.dec pos

Decrement current position Pointer

Intention

This command is used to change the internal "actual position Pointer". It will decrement this pointer by 1. If **P2** is given, the actual position will be decremented by the number in P2.

Immediately after parsing, the actual position is at the Root of the Document.
After any successful search, the actual position is at the start tag of the found tag.

To find out the actual position, you can use the gpo - Get Position 4182 command.
To set the position to another value you can use the spo - Set Position 4273 command.

Syntax

XML.Decrement Position[|P2]

Parameter Explanation

P2 **-** ***(optional)*** Variable or numeric parameter, that is decremented from current position. If omitted, 1 is used.

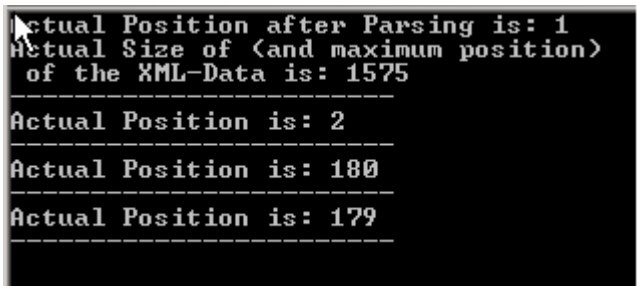
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of (and maximum position)
PRT. of the XML-Data is: $$LEN
'Here we start
PRT.------------------------
XML.inc pos
GSB. pos
PRT.------------------------
XML.sdn|{&EXACT:Product}
GSB.pos
PRT.------------------------
XML.dec pos
GSB. pos
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
:pos
XML.get pos|$$POS
PRT.Actual Position is: $$POS
RET.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

You can not Increment the actual position to values that are outside of the current loaded XML-File. The command will prevent this internally. The valid values range from 1 to XML-Length.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gpo - Get Position** 4182
- **gpr - Get Position for Previous** 4185
- **spo - Set Position** 4273

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.7 dmp - Dump

MiniRobotLanguage (MRL)

XML.Dump
XML.dmp

Show the actual loaded XML-Data

Intention

This command is used to show the actual loaded XML-Data together with the XML-Data-Flags. You can see a sample of such a XML-Data Dump here:

```
0297  L:(004) A:(060) C: < Open Bracket,Outs. SQ/DQ,Self Terminate,Start Tag,
0298  L:(004) A:(083) C: S Between,Outs. SQ/DQ,Start Name,
0299  L:(004) A:(080) C: P Between,Outs. SQ/DQ,End Name,
0300  L:(004) A:(032) C:   Between,Outs. SQ/DQ,
0301  L:(004) A:(083) C: S Between,Outs. SQ/DQ,Start Attribute,
0302  L:(004) A:(080) C: P Between,Outs. SQ/DQ,
0303  L:(004) A:(078) C: N Between,Outs. SQ/DQ,
0304  L:(004) A:(097) C: a Between,Outs. SQ/DQ,
0305  L:(004) A:(109) C: m Between,Outs. SQ/DQ,
0306  L:(004) A:(101) C: e Between,Before EQ,Outs. SQ/DQ,
0307  L:(004) A:(061) C: = Between,Outs. SQ/DQ,
0308  L:(004) A:(034) C: " Between,Is DQ,Betw.DQ,After .EQ,
0309  L:(004) A:(083) C: S Between,Betw.DQ,
0310  L:(004) A:(080) C: P Between,Betw.DQ,
0311  L:(004) A:(053) C: 5 Between,Betw.DQ,
0312  L:(004) A:(034) C: " Between,Is DQ,Outs. SQ/DQ,
0313  L:(004) A:(047) C: / Between,Outs. SQ/DQ,End Attribute,
0314  L:(004) A:(062) C: > Close Bracket,Outs. SQ/DQ,Self Terminate,
0315  L:(004) A:(013) C: -
```

At the very left side you can see the sequential **Byte-number** (Address) of the Data Bytes.
These start at 1 and go up to the Length of your XML-Data in Bytes.
Then you can see the "**Level**" of the current Data-Byte.
Next you can see the **ASCII-Value** of the Data Byte itself.
Next, after the C: you can see the **Character String** of that Data-Byte.
Finally there are the **XML-Flags** of each Data Byte.

Syntax

XML.Dump[|P2][|P3]

Parameter Explanation

P2 - (optional) numeric Parameter - Start-Byte

P3 - (optional) numeric Parameter - End-Byte

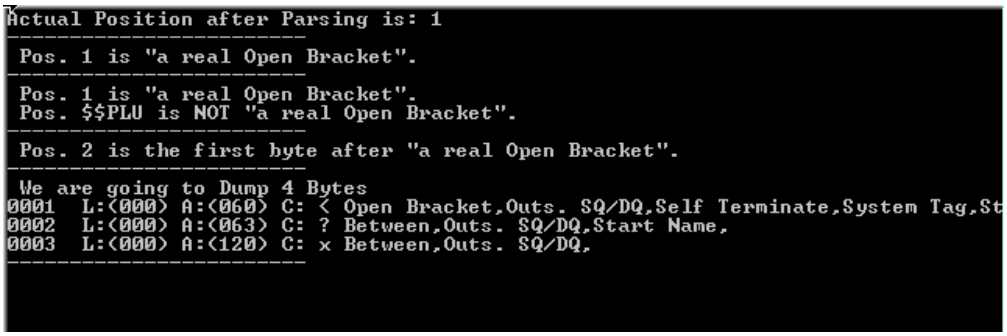
© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
IML.$$PRO|1|500
  $$ERG=$$000
  PRT. Pos. $$ERG is "$$TXT".
EIF.
PRT.------------------------
IML.$$PRO|$$ERG
  PRT. Pos. $$ERG is "$$TXT".
EIF.
IML.$$PRO|($$ERG+1)
  CAL.$$PLU=$$ERG+1
  PRT. Pos. $$PLU is "$$TXT".
ELS.
  PRT. Pos. $$PLU is NOT "$$TXT".
EIF.
PRT.------------------------
' Now we are scanning for "Bit not set"
IML.$$PROz|$$ERG|($$ERG+50)
  $$RES=$$000
  PRT. Pos. $$RES is the first byte after "$$TXT".
EIF.
PRT.------------------------
PRT. We are going to Dump 4 Bytes
XML.Dump|($$ERG-1)|($$RES+1)
PRT.------------------------
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **glev - Get Level** 4179
- **prt - Print** 4226

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.8 dncb - Down Close Bracket

MiniRobotLanguage (MRL)

XML.Down Close Bracket
XML.dncb

Move position to close bracket in down direction

Intention

This command is used to move the actual position to the next > Closing bracket in down direction.

Usage:

```
' The new position will be in $$ERG
' if nothing found $$ERG=0
XML.dncb|$$ERG

' One Parameter will also work.
XML.dncb
XML.get pos|$$POS
PRT.Actual Position is: $$POS
```

Syntax

XML.Down Close Bracket[|P2]

Parameter Explanation

P2 **-** ***(optional)*** Variable to return the result, that is the new position or zero, if nothing was found. . The result is 0 in case nothing was found.
If something was found, the actual position will be set to this new position.
If P2 is omitted the result is been placed on TOS.

Example

```
'*********************************
'
'*********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.dncb|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
------------------------
Actual Position is: 38
------------------------
 We are going to Dump these Bytes.
------------------------
0033  L:(000) A:(070) C: F Between,Betw.DQ,
0034  L:(000) A:(045) C: - Between,Betw.DQ,
0035  L:(000) A:(056) C: 8 Between,Betw.DQ,
0036  L:(000) A:(034) C: " Between,Is DQ,Outs. SQ/DQ,End Attribute,
0037  L:(000) A:(063) C: ? Between,Outs. SQ/DQ,
0038  L:(000) A:(062) C: > Close Bracket,Outs. SQ/DQ,Self Terminate,System Tag,
0039  L:(000) A:(013) C: -
0040  L:(000) A:(010) C: -
0041  L:(000) A:(060) C: < Self Terminate,Comment Tag
0042  L:(000) A:(033) C: ! Comment Tag
0043  L:(000) A:(045) C: - Comment Tag
------------------------
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **upob - Up Open Bracket** 4301
- **upcb - Up Close Bracket** 4298
- **dnob - Down Open Bracket** 4155
-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.9 dnob - Down Open Bracket

MiniRobotLanguage (MRL)

XML.Down Open Bracket
XML.dnob

Move position to open bracket in down direction

Intention

This command is used to move the actual position to the next < Open bracket in down direction.

Usage:

```
' The new position will be in $$ERG
' if nothing found $$ERG=0
XML.dnob|$$ERG

' One Parameter will also work.
XML.dnob
XML.get pos|$$POS
PRT.Actual Position is: $$POS
```

Syntax

XML.Down Open Bracket[|P2]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result, that is the new position or zero, if nothing was found. . The result is 0 in case nothing was found.
If something was found, the actual position will be set to this new position.
If P2 is omitted the result is been placed on TOS.

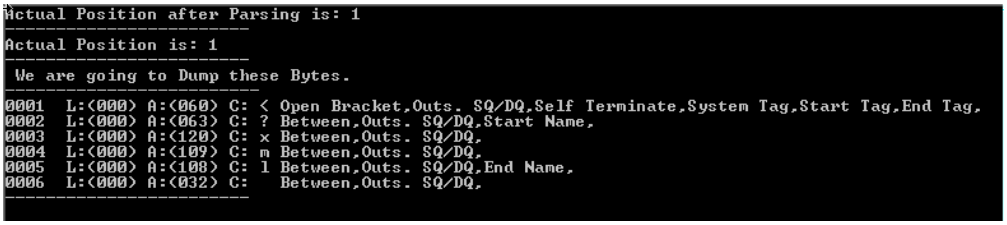
Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.dnob|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **upob - Up Open Bracket** 4301
- **upcb - Up Close Bracket** 4298
- **dncb - Down Close Bracket** 4152

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.10 ftxt - Flags to Text

MiniRobotLanguage (MRL)

XML.Flags to Text
XML.ftxt

Convert numeric XML-Data Flags to textual readable flags

Intention

This command is used to convert a numeric expression into the textual equivalent of the XML-Data-Flags.

Example:

```
XML.ftxt|2|$$ERG
PRT.$$ERG
```

```
XML.ftxt|127|$$ERG
PRT.$$ERG
```

If you use

```
XML.ftxt
PRT.$$ERG
```

Then the command will use the flags from the actual position as input.
Therefore this operation will need a XML-File to be parsed or loaded first.

Syntax

XML.Flags to Text[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** numeric parameter that is converted to the equivalent flags. If omitted, the current position's flags are being used.

P3 - ***(optional)*** Variable for the result. If omitted TOS is used..

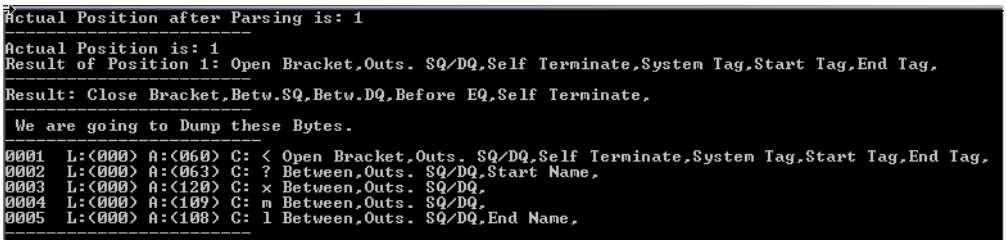
*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.dnob|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.ftxt
PRT.Result of Position $$POS: $$000
PRT.------------------------
XML.ftxt|1234|$$ERG
PRT.Result: $$ERG
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------

MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.11 gbl - Get Block

MiniRobotLanguage (MRL)

XML.Get Block
XML.gbl

Get a complete XML-Data Block

Intention

This command is used to retrieve the start- and end-position of the XML-Data-Block that we are in. They will be placed in **P2** and **P3**.

You can extract the block using the `XML.gxp`-command.
Here is an example:

```
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
' Now we get the data from the block into $$BLO
XML.gxp|$$BSA|$$BSE|$$BLO
PRT.------------------------
PRT. This is the Block:
PRT.$$BLO
```

Syntax

XML.Get Block[|P2][|P3]

Parameter Explanation

P2 - (optional) Variable to return the start-byte of the block.
P3 - (optional) Variable to return the end-byte of the block.

The results are 0 in case of parsing errors. Otherwise the results of both parameters are
>0 and contain the start- and end-byte number, of that block where we are currently in.
If omitted any of the values are being placed on TOS (Top of Stack).

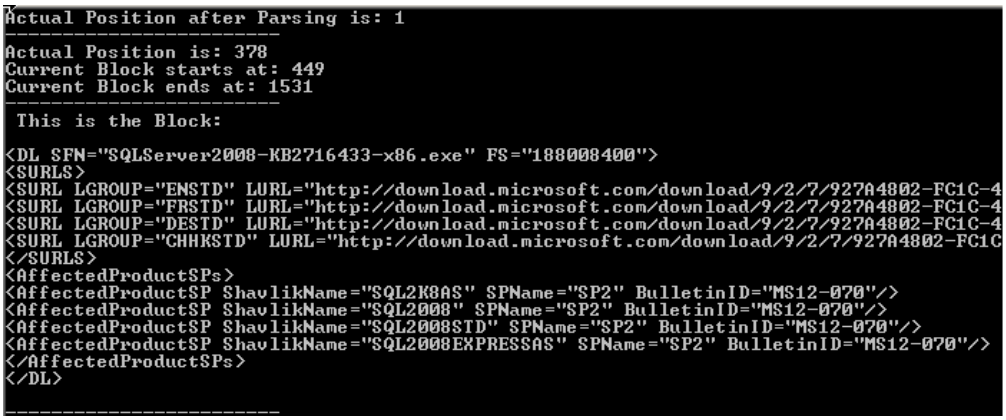
© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
' We search fo a Tag with the Name "Patch"
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
XML.gxp|$$BSA|$$BSE|$$BLO
PRT.------------------------
PRT. This is the Block:
PRT.$$BLO
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Here is the important part of the used XML-Data:

```
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe" SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
<SURL LGROUP="ENSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
<SURL LGROUP="FRSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
<SURL LGROUP="DESTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
<SURL LGROUP="CHHKSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
</SURLS>
<AffectedProductSPs>
<AffectedProductSP ShavlikName="SQL2K8AS" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008" SPName="SP2" BulletinID="MS12-
070"/>
<AffectedProductSP ShavlikName="SQL2008STD" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008EXPRESSAS" SPName="SP2"
BulletinID="MS12-070"/>
</AffectedProductSPs>
</DL>
</Patch>
</Products>
```

Remarks

-

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gse - Get Start End Tags** 4194
- **gxp - Get XML Part** 4209
- **sxp - Set XML Part** 4293
- **sdn - Search Down Tag Name** 4247

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.12 gend - Goto End Tag

MiniRobotLanguage (MRL)

XML.Goto End Tag
XML.GEnd

Goto nearest End-Tag

Intention

This command will set actual position to the **nearest `</xxx >` `End`**-Tag. It will jump over Tags that are self terminated.

The found End-Tag is just the "<" open bracket od the nearest End-Tag that is nearest to the current position.

Usage:

```
XML.GEnd|$$END

or

' We give the starting Position, that should be corrected to
' the start-position of the nearest End-Tag.
VAR.$$STA=199
XML.GEnd|$$END|$$STA
```

Hint:
If you need **the End-Tag of your current Tag**, use
`XML.gbl|$$STA|$$END`
instead. You will have the End-Tag Position in `$$END`.

Syntax

XML.Goto End Tag[|P2][|P3]

Parameter Explanation

P2 **-** ***(optional)*** Variable to return the result. The result is 0 in case of parsing errors.
Otherwise the result is >0 and contains the position of the End-Tag of our current positions start-tag.

*© 2023 Theo Gottwald * https://www.smart-package.com*

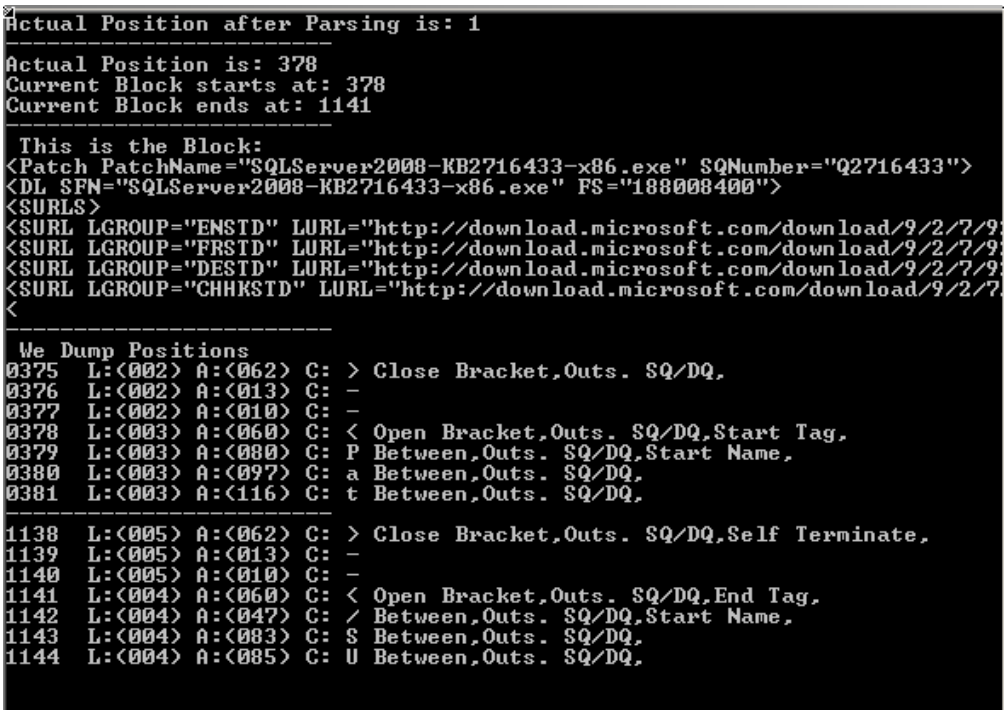
P3 **-** ***(optional)*** Numeric value that is taken as the start-tag position.
If omitted, current position is used.

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gend|$$BSE
PRT.Current Block starts at: $$POS
PRT.Current Block ends at: $$BSE
XML.gxp|$$POS|$$BSE|$$BLO
PRT.------------------------
PRT. This is the Block:
PRT.$$BLO
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
XML.Dump|($$BSE-3)|($$BSE+3)

MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gend - Goto End Tag** 4165
- **gst - Goto Start Tag** 4197
- **gse - Get Start End Tags** 4194
- **gbl - Get Block** 4161

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.13 gfl - Get Flags

MiniRobotLanguage (MRL)

XML.Get Flags
XML.gfl

Get XML-Flags

Intention

This command is used to retrieve the numerical XML-Flag Data for any Byte of the loaded XML-Code.

Usage is simple:

```
' Result will be on TOS (Top of Stack)
XML.gfl
' We take it drom TOS into Variable $$ERG
$$ERG=$$000
' We display the numeric value
PRT.Flags are: $$ERG
' This converts the number into a textual representation
XML.Ftxt|$$ERG|$$TXT
' We display the text
PRT. As Text: $$TXT
' Now we esamine the location $$POS+1
' Result comes into $$ERG
XML.gfl|($$POS+1)|$$ERG
' We convert and display its
XML.Ftxt|$$ERG|$$TXT
PRT. Next Byte as Text: $$TXT
```

Syntax

XML.Get Flags|P2[|P3]

Parameter Explanation

P2 - ***(optional)*** numeric parameter that is a byte address of the XML-Data. If omitted, the current position's flags are being used.

P3 - ***(optional)*** Variable for the result. If omitted TOS is used..

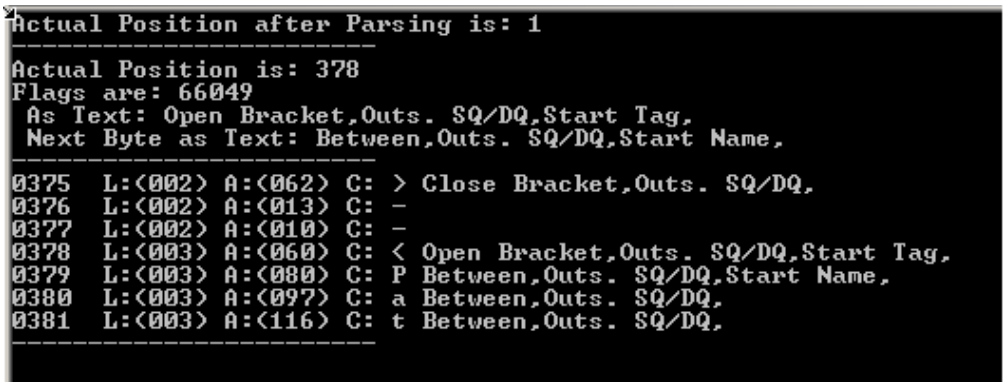
*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gfl
$$ERG=$$000
PRT.Flags are: $$ERG
XML.Ftxt|$$ERG|$$TXT
PRT. As Text: $$TXT
XML.gfl|($$POS+1)|$$ERG
XML.Ftxt|$$ERG|$$TXT
PRT. Next Byte as Text: $$TXT
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.14 gfn - Get First Node

MiniRobotLanguage (MRL)

XML.Get First Node
XML.gfn

Get Position of first node in XML-Data

Intention

This command is used to move actual position to the fist "<" first node in the XML-Data.
If your XML-Data looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1."'>
<Products>
```

then the black marked < is the actual position after a call to "First node".
All System- and comment-Tags will be ignored.

It will return the found position as a result in **P2 (if P2 is specified)** or leave the address on the TOS.

Syntax

XML.Get First Node[|P2]

Parameter Explanation

P2 - (optional) Variable to return the result. The result is 0 in case of parsing errors.
Otherwise the result is >0 and contains the byte-address of the first node..

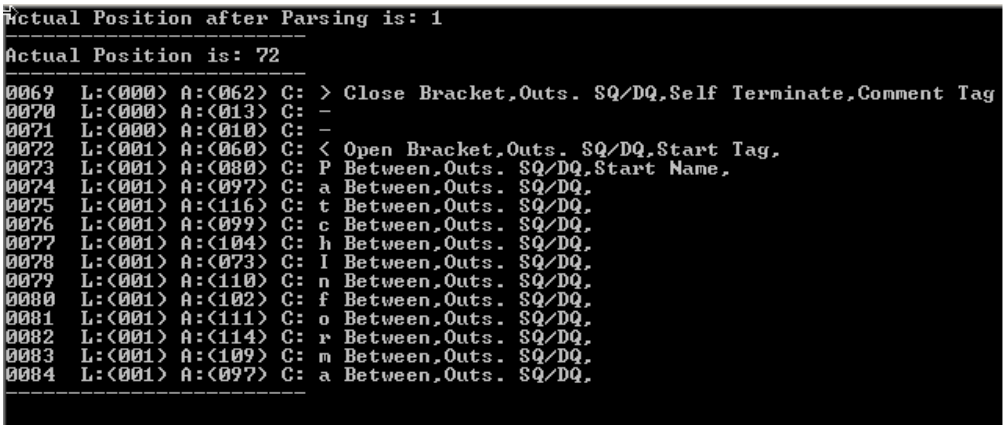
Example

```
'*********************************
'
'*********************************
'Prototype für IML.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gfn
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+12)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
```


© 2023 Theo Gottwald * https://www.smart-package.com

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gtr - Goto Root** 4206
- **gro - Get Root** 4188
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.15 gftxt - Get Flags as Text

MiniRobotLanguage (MRL)

XML.Get Flags as Text
XML.gftxt

Get XML-Flags as text

Intention

This command will retrieve the XML-Flags from a specified Byte-Address (**P2**) of the XML-Data. The result will be converted to readable text.

```
Usage is simple:

XML.gftxt
$$TXT=$$000
PRT.Actual Flags are: $$TXT
PRT.------------------------
XML.gftxt|($$POS+1)|$$TXT
PRT.Actual Flags are: $$TXT
PRT.------------------------
```

```
------------------------
Actual Flags are: Open Bracket,Outs. SQ/DQ,Self Terminate,System Tag,Start
------------------------
Actual Flags are: Between,Outs. SQ/DQ,Start Name,
------------------------
0001  L:(000) A:(060) C: < Open Bracket,Outs. SQ/DQ,Self Terminate,System
```

Syntax

XML.Get Flags as Text[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** *numeric Byte address of the XML-Data Byte from which you want the flags.*

P3 - ***(optional)*** *Variable for the textual result. If omitted, teh result is been placed on TOS.*

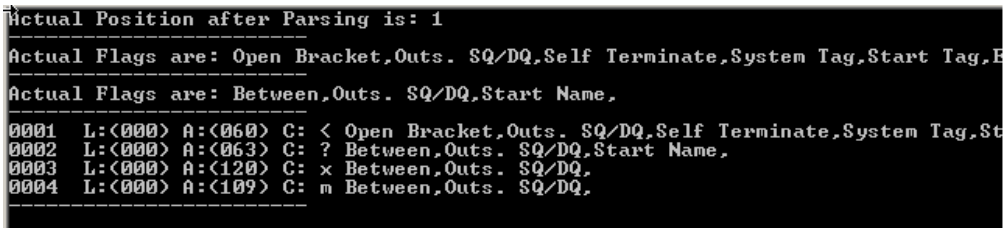
Example

```
'************************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
'*********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gftxt
$$TXT=$$000
PRT.Actual Flags are: $$TXT
PRT.------------------------
XML.gftxt|($$POS+1)|$$ERG
PRT.Actual Flags are: $$ERG
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.16 gle - Get Len

MiniRobotLanguage (MRL)

XML.Get Len
XML.gle

Get Length of XML-Data

Intention

This command is useful to retrieve the total length of the parsed XML-Data. XML-Data can be addressed from 1 to this number.

Usage is simple:

```
XML.get len|$$LEN
PRT.Actual Size of XML-Data is: $$LEN

or

XML.get len
$$LEN=$$000
PRT.Actual Size of XML-Data is: $$LEN
```

Syntax

XML.Get Len[|P2]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result.
The result is >0 and contains the total length of the parsed XML-Data in Bytes.
If omitted the result is placed on TOS (Top of Stack).

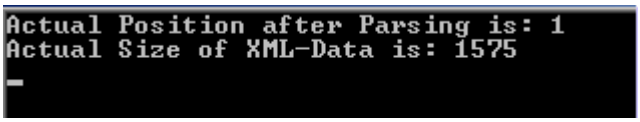
Example

```
'*********************************
'
'*********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of XML-Data is: $$LEN
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gpo - Get Position** 4182
- **spo - Set Position** 4273
- **dmp - Dump** 4149
- **gro - Get Root** 4188

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.17 glev - Get Level

MiniRobotLanguage (MRL)

XML.Get Level
XML.glev

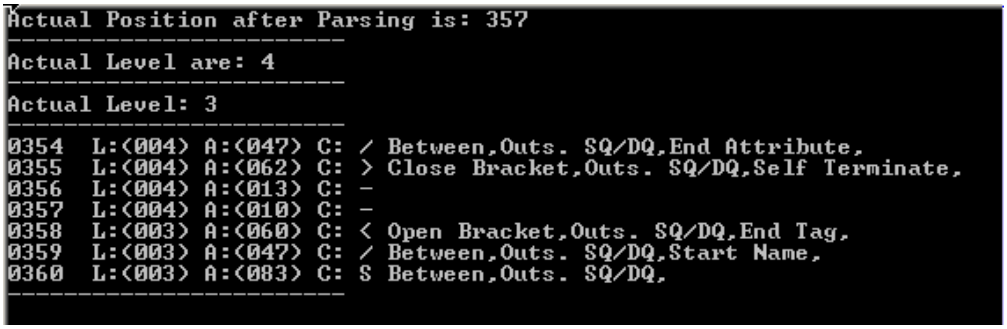
Get XML-Nesting Level

Intention

This command is used to retrieve the nesting level of any position in the XML-File. Note that Self-Terminating Tags do not increase or decrease the nesting level.

```
' Example for nesting Level.
<Level 0>
  <Level 1>
     <Level 2>

     <Level 2>
  <Level 1>
<Level 0>
```

You can see the nesting Level in the XML-Data Dump, behind the "L:" :



Syntax

XML.Get Level[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** *numeric Byte address of the XML-Data Byte from which you want the nesting Level.*

P3 - ***(optional)*** *Variable for the textual result.*

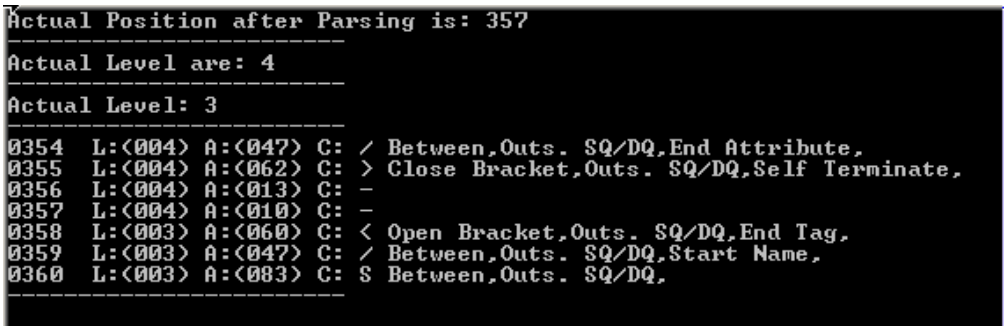
 © 2023 Theo Gottwald * https://www.smart-package.com

If omitted, teh result is been placed on TOS.

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
XML.Set Pos|357
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gLev
$$TXT=$$000
PRT.Actual Level are: $$TXT
PRT.------------------------
XML.gLev|($$POS+1)|$$ERG
PRT.Actual Level: $$ERG
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gpo - Get Position** 4182
- **dmp - Dump** 4149
- **gro - Get Root** 4188
- **prt - Print** 4226

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.18 gpo - Get Position

MiniRobotLanguage (MRL)

XML.Get Position
XML.gpo

Get or range check actual Position in XML-Data

Intention

This command is used to retrieve the actual position inside the XML-Data.
The actual position is automatically set to XML_Root (see gro - Get Root 4188) after Parsing.
It is changed by any successful search command to the search result position.
You can directly set the actual position using spo - Set Position 4273.

The XML Position or "actual position" is a number between 1 and XML-Lenght (see gle - Get Len 4177). Its simply a Byte Number in the XML-Data.

If **P3** is specified, the command will correct the numeric parameter in **P3** and make sure that it is in a valid positional range for the actual XML-Data.

The only difference between
`XML.gpo|$$RE|$$POS`
and
`XML.gpr|$$RE|$$POS`
is, that if **P3**< XML-Root then "`gpr`" will return XML-Length as **P2,** while "`gpo`" will return XML_Root as **P2**.

Syntax

XML.Get Position[|P2][|P3]

Parameter Explanation

P2 – ***(optional)*** *Variable for the numeric result of the actual XML-Position, or (if **P3** is specified), for the corrected value that was in **P3**..*
*If omitted, the result is been placed on **TOS***

P3 – ***(optional)*** *numeric Byte address of the XML-Data Byte that should be taken as position.*
*If **P3** is given, the number in **P3** and the value is been verified against the real size of the actual XML-Data. If **P3** is within the actual XML-Data size,*

 *© 2023 Theo Gottwald * https://www.smart-package.com*

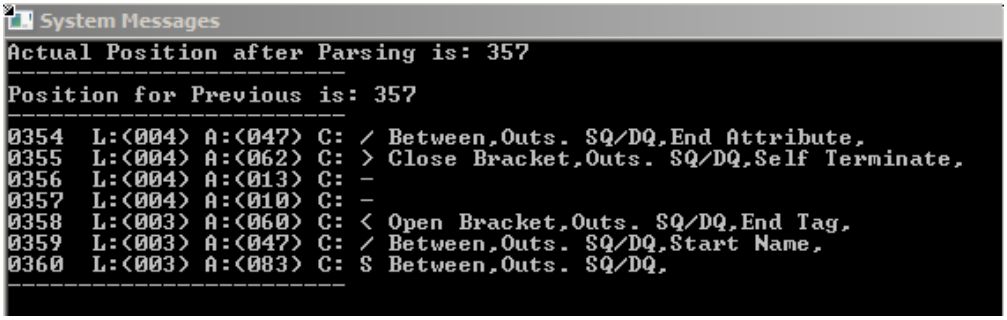
it is returned unchanged. If P3 is not a valid XML-Address because its lower then
XML-Root or higher then XML-Length then its changed to the highest or lowest allowed value. The result is always in ***P2****. If P3 is given, the result is the "corrected"*
value in P3 and not necessarily the actual XML-Position.
.

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
XML.Set Pos|357
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gpr
PRT.Position for Previous is: $$000
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **spo - Set Position** 4273
- **dmp - Dump** 4149
- **gro - Get Root** 4188

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.19 gpr - Get Position for Previous

MiniRobotLanguage (MRL)

XML.Get Position for Previous
XML.gpr

Get or range check actual Position in XML-Data

Intention

This command is used to retrieve the actual position inside the XML-Data.
The actual position is automatically set to XML_Root (see gro - Get Root 4188) after Parsing.
It is changed by any successful search command to the search result position.
You can directly set the actual position using spo - Set Position 4273.

The XML Position or "actual position" is a number between 1 and XML-Lenght (see gle - Get Len 4177). Its simply a Byte Number in the XML-Data.

If **P3** is specified, the command will correct the numeric parameter in **P3** and make sure that it is in a valid positional range for the actual XML-Data.

The only difference between
```
XML.gpr|$$RE|$$POS
```
and
```
XML.gpo|$$RE|$$POS
```
is, that if P3< XML-Root then "gpr" will return XML-Length as **P2,** while **"gpo"** will return XML_Root as **P2**.

Syntax

XML.gpr[|P2][|P3]

Parameter Explanation

P2 – ***(optional)*** *Variable for the numeric result of the actual XML-Position, or (if **P3** is specified), for the corrected value that was in **P3**..*
*If omitted, the result is been placed on **TOS***

P3 – ***(optional)*** *numeric Byte address of the XML-Data Byte that should be taken as position.*
*If **P3** is given, the number in **P3** and the value is been verified against the real size of the actual XML-Data. If **P3** is within the actual XML-Data size,*

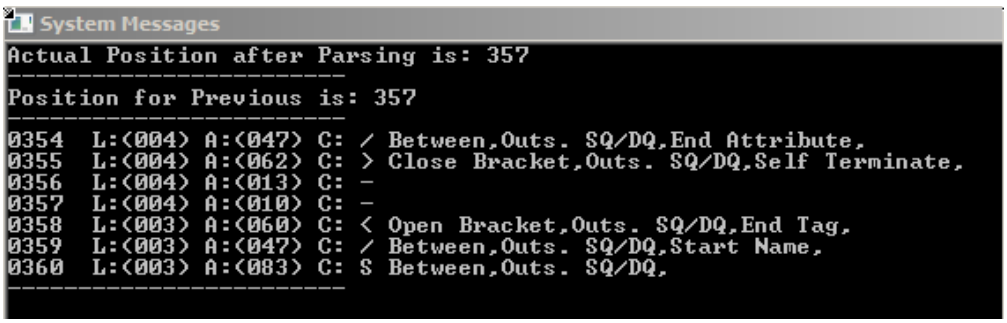
© 2023 Theo Gottwald * https://www.smart-package.com

it is returned unchanged. If P3 is not a valid XML-Address because its lower then
*XML-Root or higher then XML-Length then its changed to the highest or lowest allowed value. The result is always in **P2**. If P3 is given, the result is the "corrected"*
value in P3 and not necessarily the actual XML-Position.

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
XML.Set Pos|357
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gpr
PRT.Position for Previous is: $$000
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gpo - Get Position** 4182
- **spo - Set Position** 4273
- **dmp - Dump** 4149
- **gro - Get Root** 4188

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.20 gro - Get Root

MiniRobotLanguage (MRL)

XML.Get Root
XML.gro

Get Root Position in XML-Data

Intention

This command is used to retrieve the position of the first open bracket in the XML-Data, which is called "XML-Root".

If for example, your XML-Code starts like this:

```
3w<?xml version="1.0" encoding="UTF-8"?>
```

then XML-Root would be at Position 3. If your XML looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Then XML-Root will be at position 1.
After Parsing, the actual Position will always be XML-Root.

This command will not change the "actual position". It will just return the value of XML-Root whether in **P2** or on TOS.

Syntax

XML.Get Root[|P2]

Parameter Explanation

P2 ***- (optional)*** Variable to return the result. The result is 0 in case of parsing errors. Otherwise the result is >0 and contains the value XML_Root.

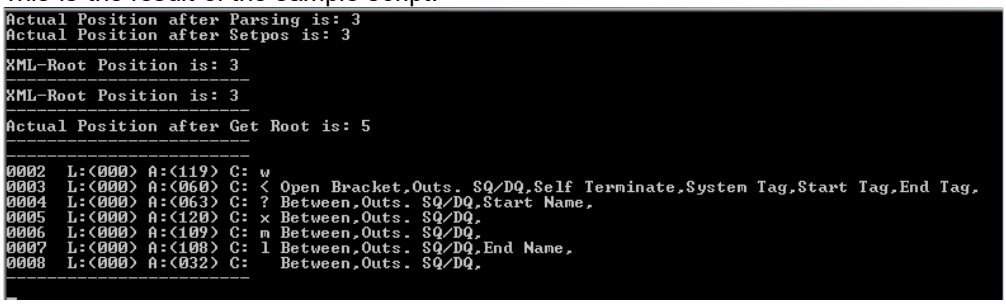
Example

```
'*********************************
'
'*********************************
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'
$$XMF=3w<?xml version="1.0" encoding="UTF-8"?>
' Now we are going to parse it
XML.Parse|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
XML.Set Pos|5
PRT.Actual Position after Setpos is: $$POS
PRT.------------------------
XML.gro
PRT.XML-Root Position is: $$000
PRT.------------------------
XML.gro|$$POS
PRT.XML-Root Position is: $$POS
PRT.------------------------
XML.get pos|$$POS
PRT.Actual Position after Get Root is: $$POS
PRT.------------------------
XML.gtr
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gtr - Goto Root** 4206
- **gfn - Get First Node** 4171
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.21 gsc - Get Asc

MiniRobotLanguage (MRL)

XML.Get Asc
XML.gsc

Get XML-Data from a file

Intention

This command is used to retrieve the ASCII-Value of any byte position of the XML-Data.
Usage:

```
$$POS=5
XML.gsc|$$POS|$$ASC
PRT.ASCII at $$POS is: $$ASC
```

Syntax

XML.Get Asc[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** *numeric Byte address of the XML-Data Byte that should be taken as position. If omitted, actual position is taken. If **P3** is given, the number in **P3** is used instead of the actual XML-Position.*

P3 - ***(optional)*** *Variable for the numeric result that is the ASCII-value of the Character at the actual position. If P3 is specified the ASCII value at that position is returned instead. If omitted, the result is been placed on **TOS***

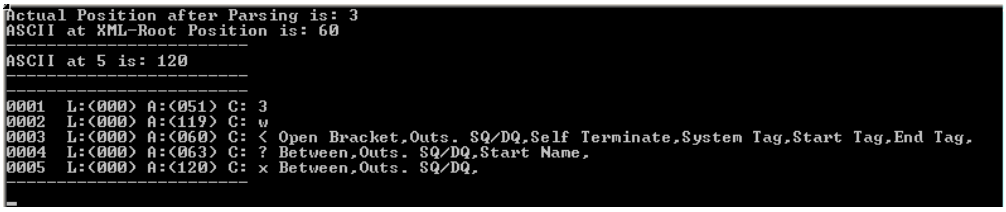
Example

```
'*********************************
'
'*********************************
'
$$XMF=3w<?xml version="1.0" encoding="UTF-8"?>
' Now we are going to parse it
XML.Parse|$$XMF
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
XML.gsc
PRT.ASCII at XML-Root Position is: $$000
PRT.------------------------
$$POS=5
XML.gsc|$$POS|$$ASC
PRT.ASCII at $$POS is: $$ASC
PRT.------------------------
XML.gtr
PRT.------------------------
XML.Dump|1|5
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990

© 2023 Theo Gottwald * https://www.smart-package.com

- par - parse from var 4117
- scr - Set Chr 4242

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.22 gse - Get Start End Tags

MiniRobotLanguage (MRL)

XML.Get Start End Tags
XML.gse

Get Stat-/End-Tag position of actual Position's Tag

Intention

This command is used to get the Start- and End-Position of the XML-Data Block at the actual position. Thhs command will return the Start- and End-Tag that corresponds together.

Example:

```
<Tag_A>
     <Tag_B>
       Content
     </Tag_B>
</Tag_A>
```

If we are in Tag_A then we will get the position of the bold marked Characters of the XML-Data. This command is very useful if you want to retrieve a complete part of the XML-Data.

See also gbl - Get Block 4161 to retrieve directly the content inside the Start- and End-Tags..

Syntax

XML.Get Start End Tags[|P2][|P3]

Parameter Explanation

P2 **-** ***(optional)*** Variable to return the start-byte of the Tag.
P3 **-** ***(optional)*** Variable to return the end-byte of the Tag.

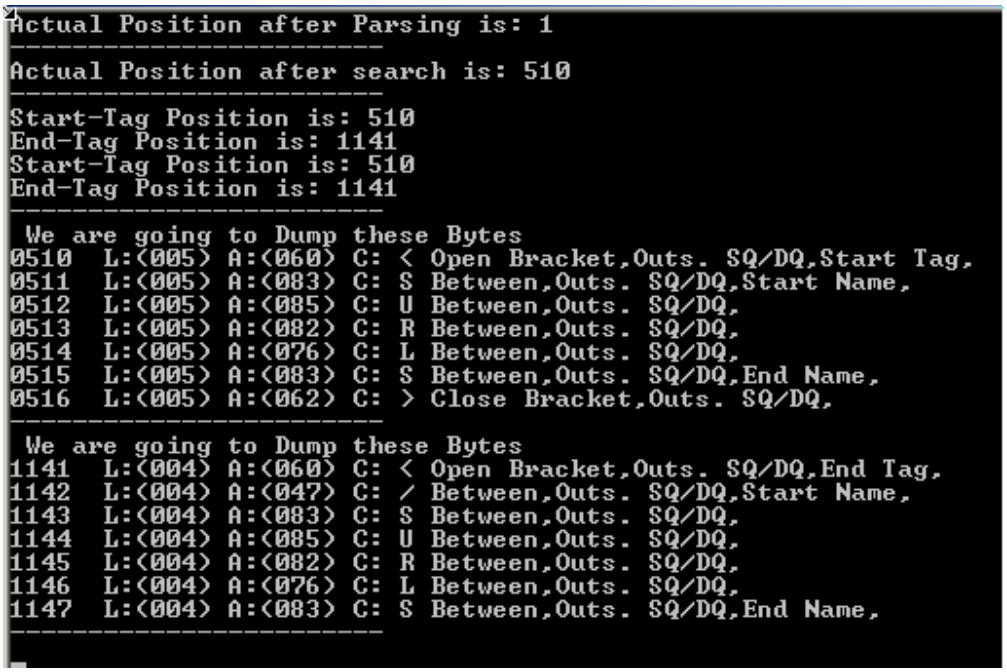
Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'*********************************
'
'*********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
XML.goto first node
' We use a Hi-Level Search here.
XML.sdn|SURLS
XML.get pos|$$POS
PRT.Actual Position after search is: $$POS
PRT.------------------------
XML.gse|$$STA|$$END
PRT.Start-Tag Position is: $$STA
PRT.End-Tag Position is: $$END
PRT.Start-Tag Position is: $$STA
PRT.End-Tag Position is: $$END
PRT.------------------------
PRT. We are going to Dump these Bytes
XML.Dump|$$STA-1|($$STA+6)
PRT.------------------------
PRT. We are going to Dump these Bytes
XML.Dump|$$END-1|($$END+6)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gbl - Get Bloc** 4161
- **gxp - Get XML Part** 4209
- **sxp - Set XML Part** 4293
- **sdn - Search Down Tag Name** 4247
- **mst - My Start Tag** 4223

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.23 gst - Goto Start Tag

MiniRobotLanguage (MRL)

XML.Goto Start Tag
XML.gst

Move actual Position to nearest upward Start-Tag

Intention

This command is used to change the actual position to the nearest Start-Tag in upward direction.If the actual position is already on a Start-Tag, the position will not be changed.

```
XML.gst
```

or

```
XML.gst|$$POS
```

In the second sample, the variable $$POS will return the new position or the unchanged position, if we are already on a start-Tag.

If you specify P3 like this;

```
XML.gst|$$POS|$$NPO
```

then **P3** will be used, instead of the actual position.

Syntax

XML.Goto Start Tag[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that is on a Start-Tag.
*If omitted, the result is been placed on **TOS***

P3 - ***(optional)*** *numeric Byte address of the XML-Data Byte that should be taken as position. If P3 is given, the result in **P2** is the "corrected" value from P3 on teh nearest Start-Tag in upward direction (previous)..*

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gse|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
XML.gst|$$BAA|$$BSA
PRT. Using gst on $$BSA -> $$BAA
XML.gst|$$BAB|$$BSE
PRT. Using gst on $$BSE -> $$BAB
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)

MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.24 gtfn - Goto First Node

MiniRobotLanguage (MRL)

XML.Goto First Node
XML.gtfn

Move actual Position to the first node in the XML-Data

Intention

This command is used to move actual position to the fist "<" first node in the XML-Data.
If your XML-Data looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1."'>
<Products>
```

then the black marked < is the actual position after a call to "First node".
All System- and comment-Tags will be ignored.

This is similar like `XML.gfn` but will not return a result and not place anything on the stack.

Syntax

XML.Goto First Node

Parameter Explanation

-

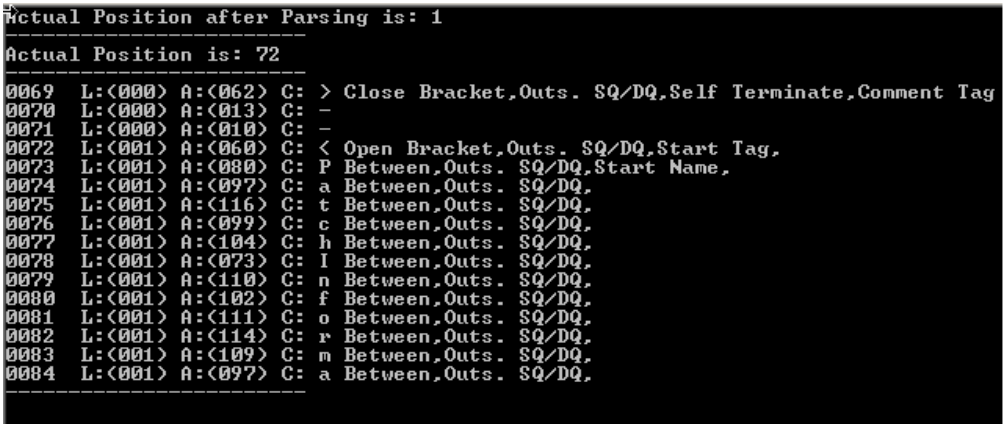
Example

```
'*********************************
'
'*********************************
'
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gtfn
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+12)
PRT.------------------------
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gtr - Goto Root** 4206
- **gro - Get Root** 4188
- **gfn - Get First Node** 4171
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.25 gtn - Get this Name

MiniRobotLanguage (MRL)

XML.Get this Name
XML.gtn

Get the name of the Tag at the actual position

Intention

This command is used to retrieve the Tag's name a t the actual position.

Here are two examples for Tag-Names:

```
' The name of this Tag is "Patch"
<Patch PatchName="SQLServer2">

' The name of this Tag is SP
<SP SPName='SP3'/>
```

Usage is simpe:

```
XML.gtn
PRT.Name is: $$000
```

or

```
XML.gtn|$$BAB|$$POS
PRT. The Tag name at position $$POS is $$BAB.
```

Syntax

XML.Get this Name[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result. The result is the name of the Tag or "" if no name could be retrieved.
*If **P2** is omitted, the result is been placed on **TOS***

P3 - ***(optional)*** *numeric Byte address of the XML-Data Byte that should be taken as position. If **P3** is omitted, the actual position is taken instead.*

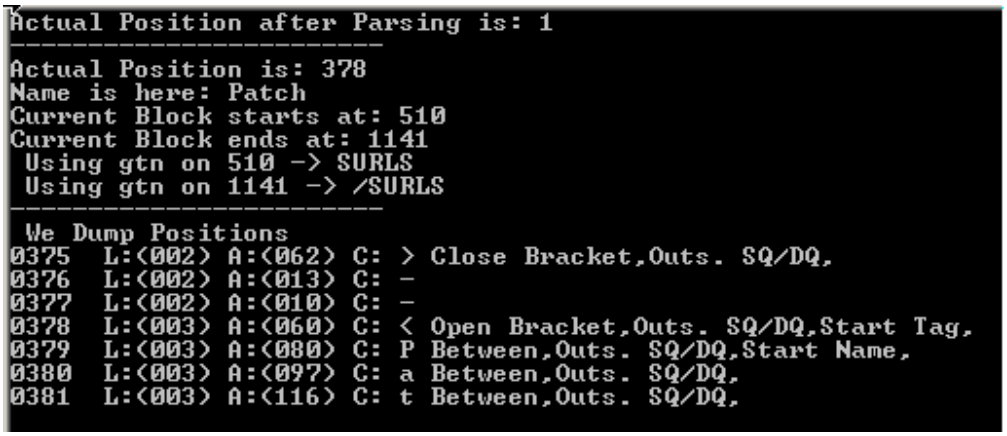
© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gtn
PRT.Name is here: $$000
XML.gse|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
XML.gtn|$$BAA|$$BSA
PRT. Using gtn on $$BSA -> $$BAA
XML.gtn|$$BAB|$$BSE
PRT. Using gtn on $$BSE -> $$BAB
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$POS-3)|($$POS+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.26 gtr - Goto Root

MiniRobotLanguage (MRL)

XML.Goto Root
XML.gtr

Goto Root Position in XML-Data

Intention

This command is used to position the internal "actual position" onto the first open bracket in the XML-Data, which is called "XML-Root".

If for example, your XML-Code starts like this:

```
3w<?xml version="1.0" encoding="UTF-8"?>
```

then XML-Root would be at Position 3. If your XML looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Then XML-Root will be at position 1.

After Parsing, the actual Position will always be XML-Root.
Therefore you only need to call this function if you want to go back to a defined position.

This command will set the actual position and additionally return the value of XML-Root whether in **P2** or on TOS.

Syntax

XML.Goto Root[|P2]

Parameter Explanation

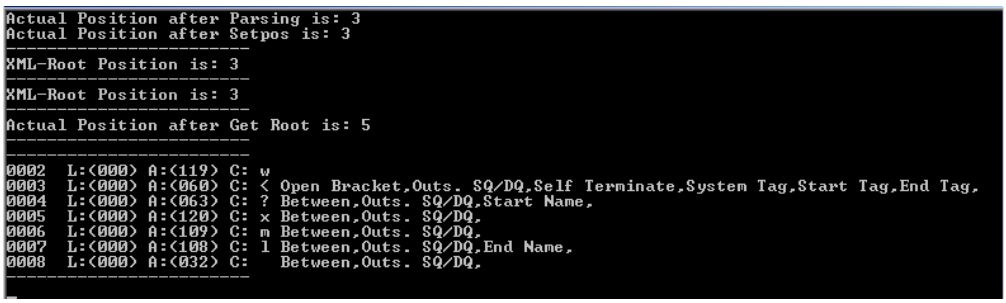
P2 - (optional) Variable to return the result. The result is 0 in case of parsing errors. Otherwise the result is >0 and contains the value XML_Root.

Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
'
'**********************************
'
$$XMF=3w<?xml version="1.0" encoding="UTF-8"?>
' Now we are going to parse it
XML.Parse|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
XML.Set Pos|5
PRT.Actual Position after Setpos is: $$POS
PRT.------------------------
XML.gro
PRT.XML-Root Position is: $$000
PRT.------------------------
XML.gro|$$POS
PRT.XML-Root Position is: $$POS
PRT.------------------------
XML.get pos|$$POS
PRT.Actual Position after Get Root is: $$POS
PRT.------------------------
XML.gtr
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

© 2023 Theo Gottwald * https://www.smart-package.com

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gro - Get Root** 4188
- **gfn - Get First Node** 4171
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.27 gxp - Get XML Part

MiniRobotLanguage (MRL)

XML.Get XML Part
XML.gxp

Get a part of the XML-Data into a variable

Intention

This command is used to retrieve a specified part or all of the XML-Data into a variable or onto TOS.

You can specify the start- and end-position of the XML-Data-Block that we want. They will be placed in **P2** and **P3**. Here is an example:

```
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
' Now we get the data from the block into $$BLO
XML.gxp|$$BSA|$$BSE|$$BLO
PRT.------------------------
PRT. This is the Block:
PRT.$$BLO
```

To get all of the XML-Data into a variable, just use **P4**, and leave **P2** and **P3** empty or fill them with a zero. Having all XML-Data in a variable, enables you to use all SPR String-Commands on the XML-Data, to search or replace things.
After that you can just "`XML.Parse`" that variable and work on the changed XML-Data.

Syntax

XML.Get XML Part|P2[|P3][|P4]

Parameter Explanation

P2 - ***(optional)*** Variable to specify the start-byte of the block.
If omitted or zero, XML-Root will be used.

P3 - ***(optional)*** Variable to specify the end-byte of the block.
If omitted or zero, XML-Length will be used.

© 2023 Theo Gottwald * https://www.smart-package.com

P4 **-** ***(optional)*** Variable to return the result. The result is "" (empty) in case of parsing errors. Otherwise the result is the XML-Data-Part..

To get all of the XML-Data into a variable, just use **P4**, and leave **P2** and **P3** empty or fill them with a zero. Having all XML-Data in a variable, enables you to use all SPR String-Commands on the XML-Data, to search or replace things.
After that you can just "`XML.Parse`" that variable and work on the changed XML-Data.

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
' We search fo a Tag with the Name "Patch"
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
XML.gxp|$$BSA|$$BSE|$$BLO
PRT.------------------------
PRT. This is the Block:
PRT.$$BLO
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

This is the result of the sample script:

Here is the important part of the used XML-Data:

```
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe" SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
<SURL LGROUP="ENSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
<SURL LGROUP="FRSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
<SURL LGROUP="DESTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
<SURL LGROUP="CHHKSTD"
LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A71-
9A2D-2DCEAC11B266/KB2716433/SQLServer2008-KB2716433-x86.exe"/>
</SURLS>
<AffectedProductSPs>
<AffectedProductSP ShavlikName="SQL2K8AS" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008" SPName="SP2" BulletinID="MS12-
070"/>
<AffectedProductSP ShavlikName="SQL2008STD" SPName="SP2"
BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008EXPRESSAS" SPName="SP2"
BulletinID="MS12-070"/>
</AffectedProductSPs>
</DL>
</Patch>
</Products>
```

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **sxp - Set XML Part** 4293
- **sdn - Search Down Tag Name** 4247
- **gbl - Get Block** 4161

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.28 inc pos - Increment Position

MiniRobotLanguage (MRL)

XML.Increment Position
XML.inc pos

Get XML-Data from a file

Intention

This command is used to change the internal "actual position Pointer". It will increment this pointer by 1. If **P2** is given, the actual position will be incremented by the number in P2.

Immediately after parsing, the actual position is at the Root of the Document.
After any successful search, the actual position is at the start tag of the found tag.

To find out the actual position, you can use the gpo - Get Position 4182 command.
To set the position to another value you can use the spo - Set Position 4273 command.

Syntax

XML.Increment Position[|P2]

Parameter Explanation

P2 **-** ***(optional)*** Variable or numeric parameter, that is incremented from current position. If omitted, 1 is used.

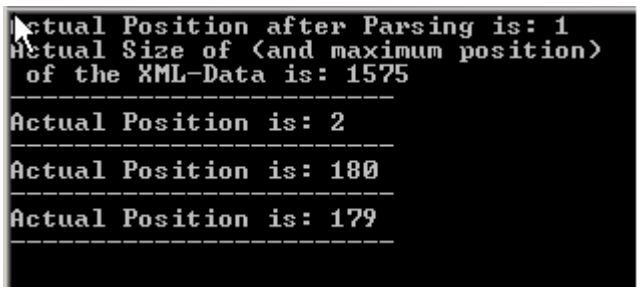
Example

```
'*********************************
'
'*********************************
'Prototype für IML.
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
XML.get len|$$LEN
PRT.Actual Size of (and maximum position)
PRT. of the XML-Data is: $$LEN
'Here we start
PRT.------------------------
XML.inc pos
GSB. pos
PRT.------------------------
XML.sdn|{&EXACT:Product}
GSB.pos
PRT.------------------------
XML.dec pos
GSB. pos
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
:pos
XML.get pos|$$POS
PRT.Actual Position is: $$POS
RET.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

You can not Increment the actual position to values that are outside of the current loaded XML-File. The command will prevent this internally. The valid values range from 1 to XML-Length.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gpo - Get Position** 4182
- **gpr - Get Position for Previous** 4185
- **spo - Set Position** 4273
- **dec pos - Decrement Position** 4146

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.29 ixp - Insert XML-Part

MiniRobotLanguage (MRL)

XML.Insert XML Part
XML.ixp

Insert Text at a specified place into the XML-Data (and reparse if needed)

Intention

This command is used to insert Text or Data parts into the XML-Data.

If you do change the total length of the XML-Data then the Parser will automatically re-parse the data and set all flags from new.
If you do not change the total length of the XML-Data then its up to you, if you want a Re-Parse or not.
See the command: **rpa - Reparse** 4232 .

If you want to remove or replace parts of the XML-Data, you can either use

- ssc - Set Asc 4280
- scr - Set Chr 4242
- sxp - Set XML Part 4293

Or transfer the complete XML-Data into a Variable, then use the normal "String Commands, and then Parse the Variable again using the XML-Parse command.

To get all of the XML-Data into a variable, use
`XML.gxp|||$$ALL`
just use **P4**, and leave **P2** and **P3** empty or fill them with a zero.

Having all XML-Data in a variable, enables you to use all SPR String-Commands on the XML-Data, to search or replace things.

After that you can just "`XML.Parse`" that variable and work on the changed XML-Data.

Syntax

XML.Insert XML Part|P2|P3|P4

Parameter Explanation

 *© 2023 Theo Gottwald * https://www.smart-package.com*

P2 – **Text** that is inserted at Place P3 - or if P3 is omitted, at the actual position into
the XML-Data.

P3 – ***(optional) numeric position*** where the text shall be inserted into the XML-Data. If zero, the actual position is taken.
If negative, it will count position from the end of the XML-Data backwards.

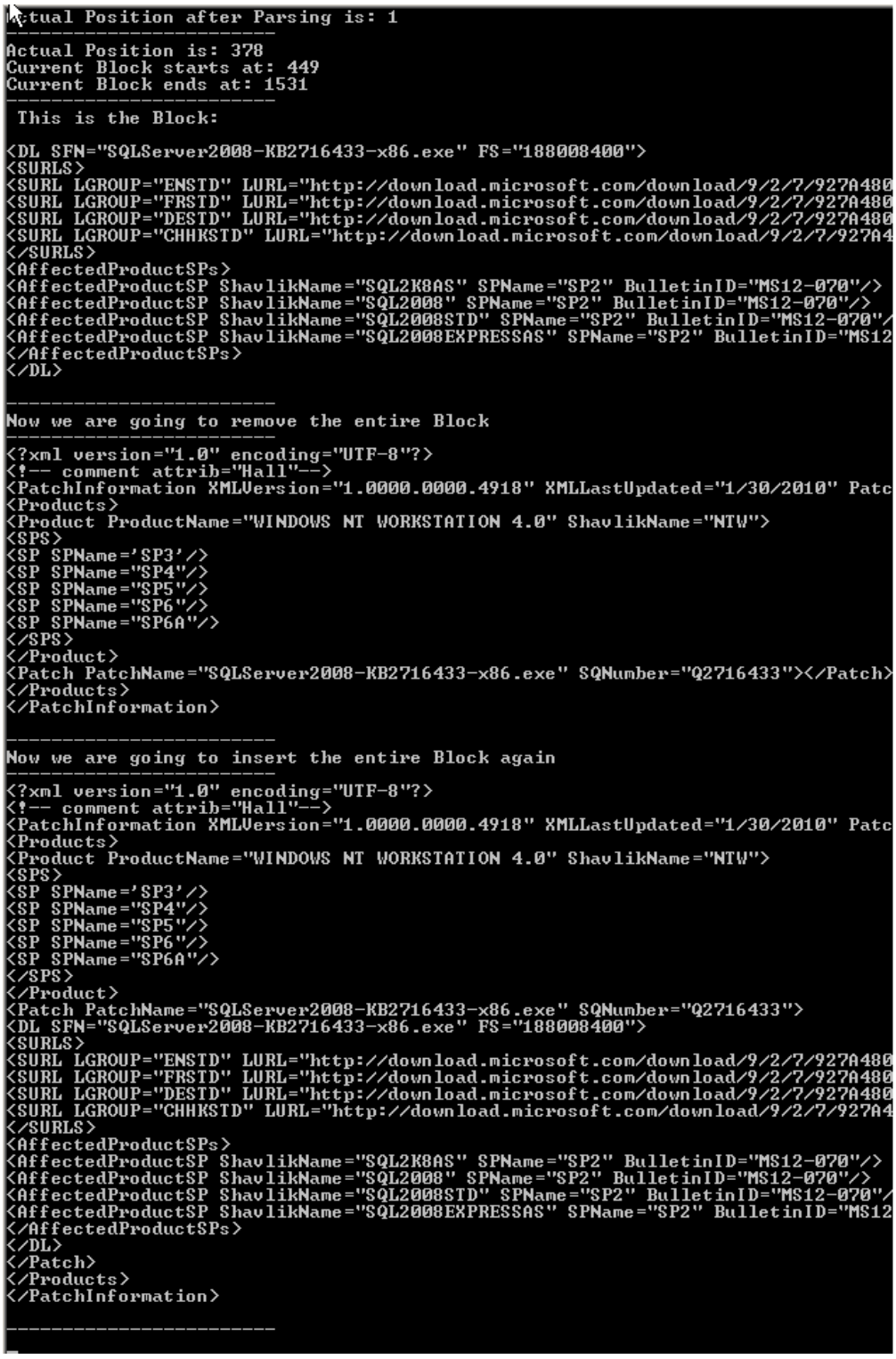
Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
XML.gxp|$$BSA|$$BSE|$$BLO
PRT.------------------------
PRT. This is the Block:
PRT.$$BLO
PRT.------------------------
PRT.Now we are going to remove the entire Block
PRT.------------------------
XML.sxp|$$BSA|$$BSE|
XML.Print
PRT.------------------------
PRT.Now we are going to insert the entire Block again
PRT.------------------------
XML.ixp|$$BLO|$$BSA
XML.Print
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gxp - Get XML Part** 4209
- **rpa - Reparse** 4232
- ssc - Set Asc 4280
- scr - Set Chr 4242
- sxp - Set XML Part 4293
- **sdn - Search Down Tag Name** 4247
- **gbl - Get Block** 4161
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.30 met - My End Tag

MiniRobotLanguage (MRL)

XML.My End Tag
XML.met

Get the End-Tag from the Tag at actual position

Intention

This command is used to set actual position to the End-Tag of the actual Start-Tag.
This command will return the End-Tag that corresponds with the actual Start-Tag.

Example:

```
<Tag_A>
     <Tag_B>
       Content
     </Tag_B>
</Tag_A>
```

If we are in <Tag_A then we will get the position of the bold marked Character
`</Tag_A>` from the End-Tag .

See also gbl - Get Block 4161 to retrieve directly the content inside the Start- and End-Tags..
See also gse - Get Start End Tags 4194 if you want both Tags with one command.

Syntax

XML.My End Tag[|P2]

Parameter Explanation

P2 - (optional) Variable to return the result. If nothing is found, the variable will be 0.

If omitted, the result will also be placed on TOS. The actual position will be changed, if something was found.

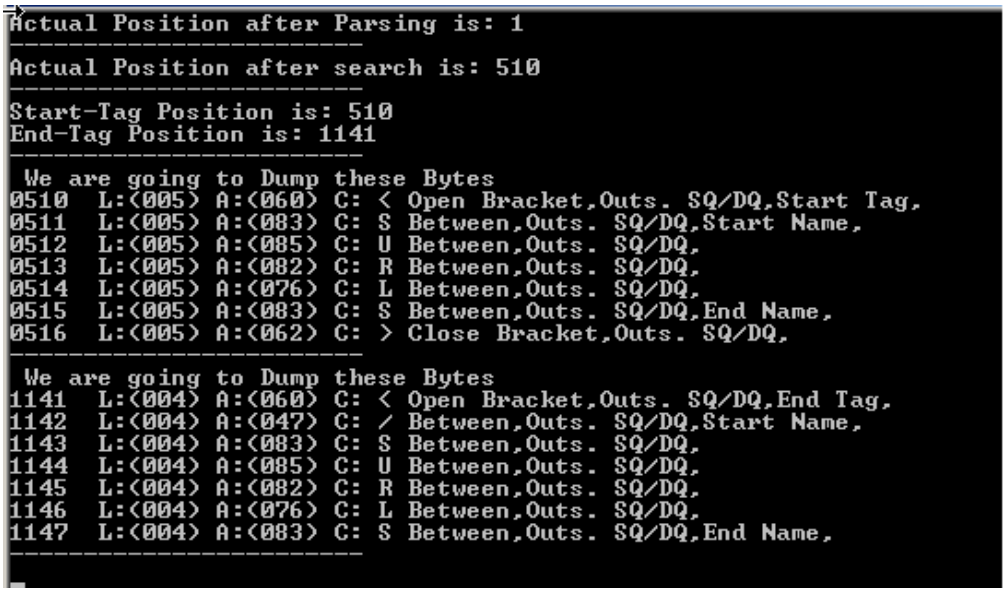
Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
XML.goto first node
' We use a Hi-Level Search here.
XML.sdn|SURLS
XML.get pos|$$POS
PRT.Actual Position after search is: $$POS
PRT.------------------------
XML.mst|$$STA
XML.met|$$END
PRT.Start-Tag Position is: $$STA
PRT.End-Tag Position is: $$END
PRT.------------------------
PRT. We are going to Dump these Bytes
XML.Dump|$$STA-1|($$STA+6)
PRT.------------------------
PRT. We are going to Dump these Bytes
XML.Dump|$$END-1|($$END+6)
PRT.------------------------
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **mst - My Start Tag** 4223
- gse - Get Start End Tags 4194
- **gbl - Get Bloc** 4161
- **gxp - Get XML Part** 4209
- **sxp - Set XML Part** 4293
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.31 mst - My Start Tag

MiniRobotLanguage (MRL)

XML.My Start Tag
XML.mst

Get the Start-Tag from the Tag at actual position

Intention

This command is used to set actual position to the Start-Tag of the actual End-Tag. This command will return the Start-Tag that corresponds with the actual located End-Tag.

Example:

```
<Tag_A>
     <Tag_B>
       Content
     </Tag_B>
</Tag_A>
```

If we are in `</Tag_A>` then we will get the position of the bold marked Character `<Tag_A>` from the Start-Tag .

See also gbl - Get Block 4161 to retrieve directly the content inside the Start- and End-Tags..
See also gse - Get Start End Tags 4194 if you want both Tags with one command.

Syntax

XML.My Start Tag[|P2]

Parameter Explanation

P2 - (optional) Variable to return the result. If nothing is found, the variable will be 0.

If omitted, the result will also be placed on TOS. The actual position will be changed, if something was found.

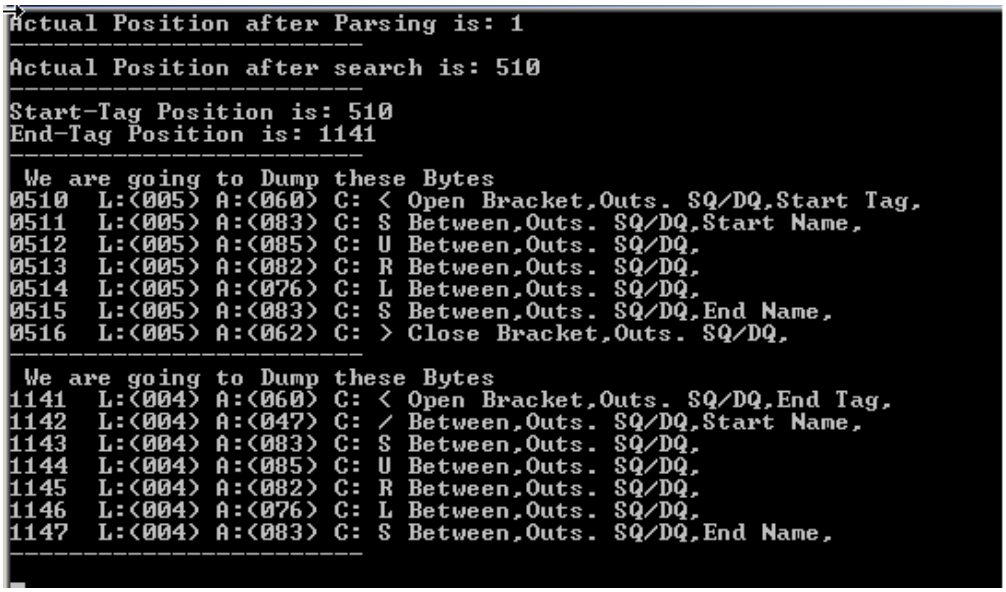
Example

© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
PRT.------------------------
XML.goto first node
' We use a Hi-Level Search here.
XML.sdn|SURLS
XML.get pos|$$POS
PRT.Actual Position after search is: $$POS
PRT.------------------------
XML.mst|$$STA
XML.met|$$END
PRT.Start-Tag Position is: $$STA
PRT.End-Tag Position is: $$END
PRT.------------------------
PRT. We are going to Dump these Bytes
XML.Dump|$$STA-1|($$STA+6)
PRT.------------------------
PRT. We are going to Dump these Bytes
XML.Dump|$$END-1|($$END+6)
PRT.------------------------
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **met - My End Tag** 4220
- **gse - Get Start End Tags** 4194
- **gbl - Get Bloc** 4161
- **gxp - Get XML Part** 4209
- **sxp - Set XML Part** 4293
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.32 prt - Print

MiniRobotLanguage (MRL)

XML.Print
XML.prt

Pretty Print the whole XML-Data into a Variable or a debug console

Intention

This command will print the whole XML-Data into a Debug- Console window, or if a variable is given, into the variable.
It will try to produce XML-Text that is loadable into a text-editor program, by inserting
Linefeeds between the Tags.

```
XML.Print|$$END
$$FIL=?desktop\PrintOut.txt
DEL.$$FIL
CTF.$$FIL|$$END
```

if you omit the variable, the Printout will go into a debug console window.

```
XML.Print
```

Syntax

XML.Print[|P2]

Parameter Explanation

P2 **-** Variable to return the XML-Listing.

Example

```
'*********************************
'
'*********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.Print
PRT.------------------------
XML.Print|$$END
$$FIL=?desktop\PrintOut.txt
DEL.$$FIL
CTF.$$FIL|$$END
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918" XMLLastUpdated="1/30/2010" PatchCount='1
<Products>
<Product ProductName="WINDOWS NT WORKSTATION 4.0" ShavlikName="NTW">
<SPS>
<SP SPName='SP3'/>
<SP SPName="SP4"/>
<SP SPName="SP5"/>
<SP SPName="SP6"/>
<SP SPName="SP6A"/>
</SPS>
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe" SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
<SURL LGROUP="ENSTD" LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A
<SURL LGROUP="FRSTD" LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A
<SURL LGROUP="DESTD" LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-4A
<SURL LGROUP="CHHKSTD" LURL="http://download.microsoft.com/download/9/2/7/927A4802-FC1C-
</SURLS>
<AffectedProductSPs>
<AffectedProductSP ShavlikName="SQL2K8AS" SPName="SP2" BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008" SPName="SP2" BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008STD" SPName="SP2" BulletinID="MS12-070"/>
<AffectedProductSP ShavlikName="SQL2008EXPRESSAS" SPName="SP2" BulletinID="MS12-070"/>
</AffectedProductSPs>
</DL>
</Patch>
</Products>
</PatchInformation>

------------------------
```

Additionally you will find a file caller "Printout.txt" on your desktop.

Remarks

-

Limitations:

© 2023 Theo Gottwald * https://www.smart-package.com

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **dmp - Dump** 4149

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.33 ptse - Position to Start or End Tag

MiniRobotLanguage (MRL)

XML.Position to Start or End Tag
XML.ptse

Get or change current position to nearest Start- or End-Tag

Intention

This command is used to locate the nearest Start- or End-Tag to the actual position.
It will change the actual position, and it will return the found Tag-Position,
if you specify **P2**.
If you specify **P3**, then **P3** will be taken instead of the actual position.

If the actual position is already on a Start-Tag or End-Tag, the position will not be changed.

```
XML.ptse
```

or

```
XML.ptse|$$POS
```

In the second sample, the variable $$POS will return the new position or the unchanged position, if we are already on a start-Tag.

If you specify P3 like this;

```
XML.ptse|$$POS|$$NPO
```

then **P3** will be used, instead of the actual position.

Syntax

XML.ptse[|P2][|P3]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that is on a Start-Tag.

*© 2023 Theo Gottwald * https://www.smart-package.com*

*If omitted, the result is been placed on **TOS***

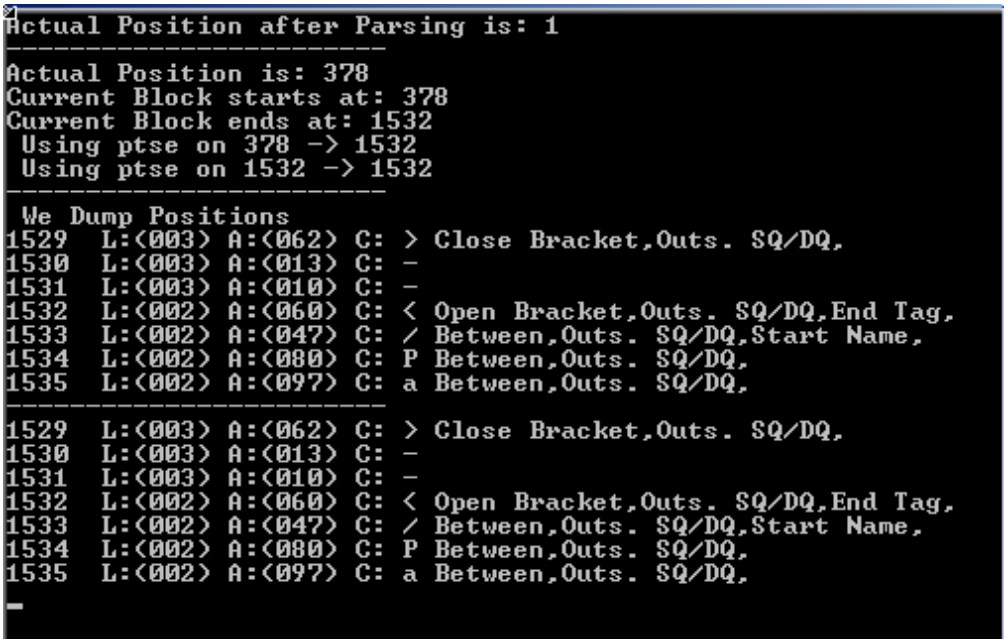
P3 *- **(optional)** numeric Byte address of the XML-Data Byte that should be taken as position. If P3 is given, the result in **P2** is the "corrected" value from P3 on teh nearest Start-Tag in upward direction (previous)..*

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gse|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
XML.ptse|$$BAA|$$BSA
PRT. Using ptse on $$BSA -> $$BAA
XML.ptse|$$BAB|$$BSE
PRT. Using ptse on $$BSE -> $$BAB

PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$BAA-3)|($$BAA+3)
PRT.------------------------
XML.Dump|($$BAB-3)|($$BAB+3)

MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gend - Goto End Tag** 4165
- **gst - Goto Start Tag** 4197
- **gse - Get Start End Tags** 4194
- **gbl - Get Block** 4161

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.34 rpa - Reparse

MiniRobotLanguage (MRL)

XML.Reparse
XML.rpa

Parse changed XML-Data and re-generate XML-Flags

Intention

This command is used to Parse already loaded or parsed XML-Data again.
This is only needed in very few cases, after you have changed the XML-Data using any of the commands that change the XML-Data, like:

- ssc - Set Asc 4280
- scr - Set Chr 4242
- sxp - Set XML Part 4293

If you use sxp - Set XML Part 4293 and change the total length of the XML-Data then the reparse will be done automatically.

I other cases, when you just change a few characters, you may want to make the reparse manually, because in these cases its not done automatically (because it takes time on large XML-files).

Syntax

XML.Reparse

Parameter Explanation

-

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:SURLS
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
PRT.------------------------
PRT.This is the block before change:
PRT.------------------------
XML.Print
PRT.------------------------
PRT.Now we use Set-XML-Part to change the block
$$TXT=!!!!!!!! New Block-Text !!!!!!!!!!
XML.sxp|$$BSA|$$BSE|$$TXT
PRT.------------------------
XML.Print
PRT.------------------------
PRT. In this case the Reparse was done internally.
PRT.------------------------
PRT.Now we will make a character change in the XML-Data
XML.scr|$$BSA|>
XML.ReParse
PRT.------------------------
XML.Print
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$BSA-3)|($$BSA+3)
PRT.------------------------
XML.Dump|($$BSB-3)|($$BSB+3)

MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- ssc - Set Asc 4280
- scr - Set Chr 4242
- sxp - Set XML Part 4293

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.35 sad - Search All Down

MiniRobotLanguage (MRL)

XML.Search All Down
XML.sad

Search XML-Flags in down direction

Intention

This command is used to search specified XML-Data Flags, starting from the actual position, in downward direction. You can use all these flags (See Link):

! SPR - XML-Parser Flags 3981

Usage:

```
XML.sad|start tag+open bracket+self terminate
XML.get pos|$$POA
PRT.Actual Position is: $$POA
```

`XML.sad` will always start from the actual position.

Syntax

XML.Search All Down|P2[|P3]

Parameter Explanation

P2 – *numeric or textual Flag-Description. For Details see here:*
. ! SPR - XML-Parser Flags 3981

P3 – ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that matched all of the specified flags.
If omitted, the result is been placed on ***TOS***

Example

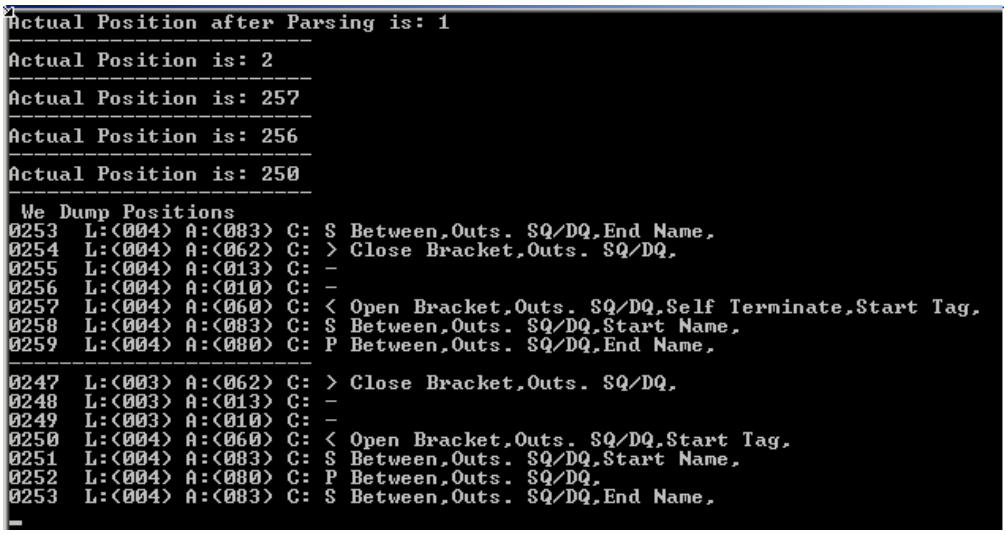
```
'*********************************
'
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.inc pos
XML.get pos|$$POA
PRT.Actual Position is: $$POA
PRT.------------------------
XML.sad|start tag+open bracket+self terminate
XML.get pos|$$POA
PRT.Actual Position is: $$POA
PRT.------------------------
XML.dec pos
XML.get pos|$$POA
PRT.Actual Position is: $$POA
PRT.------------------------
XML.sau|start tag
XML.get pos|$$POB
PRT.Actual Position is: $$POB
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$POA-3)|($$POA+3)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **siu - Search Inside Up** 4254
- **sid - Search Inside Down** 4250
- **sau - Search All Up** 4239
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.36 sau - Search All Up

MiniRobotLanguage (MRL)

XML.Search All Up
XML.sau

Search XML-Flags in upward direction

Intention

This command is used to search specified XML-Data Flags, starting from the actual position, in upward direction. You can use all these flags (See Link):

! SPR - XML-Parser Flags 3981

Usage:

```
XML.sau|start tag+open bracket+self terminate
XML.get pos|$$POA
PRT.Actual Position is: $$POA
```

`XML.sau` will always start from the actual position.

Syntax

XML.Search All Up|P2[|P3]

Parameter Explanation

P2 – *numeric or textual Flag-Description. For Details see here:*
. ! SPR - XML-Parser Flags 3981

P3 – ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that matched all of the specified flags.
If omitted, the result is been placed on ***TOS***

Example

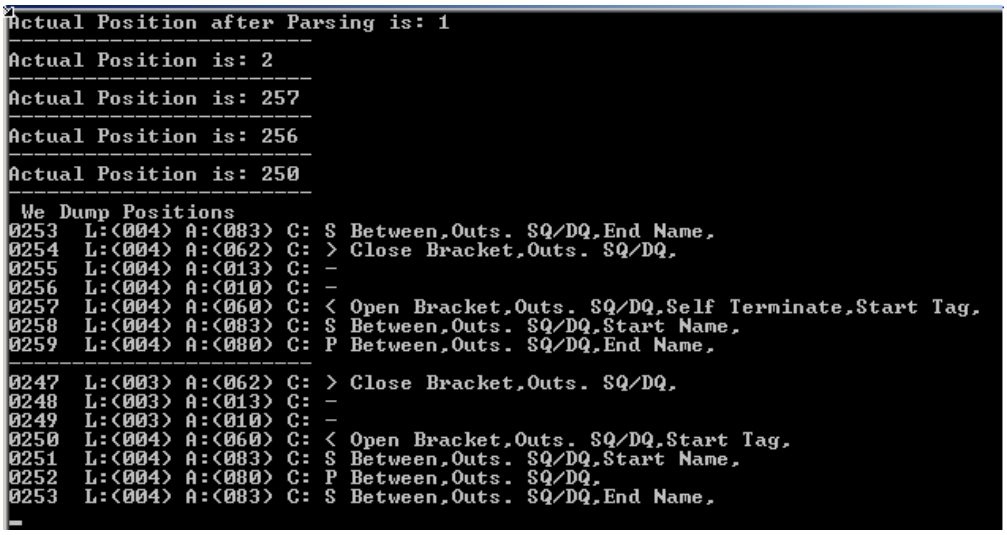
```
'*********************************
'
'*********************************
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.inc pos
XML.get pos|$$POA
PRT.Actual Position is: $$POA
PRT.------------------------
XML.sad|start tag+open bracket+self terminate
XML.get pos|$$POA
PRT.Actual Position is: $$POA
PRT.------------------------
XML.dec pos
XML.get pos|$$POA
PRT.Actual Position is: $$POA
PRT.------------------------
XML.sau|start tag
XML.get pos|$$POB
PRT.Actual Position is: $$POB
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$POA-3)|($$POA+3)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **siu - Search Inside Up** 4254
- **sid - Search Inside Down** 4250
- **sad - Search All Down** 4236

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.37 scr - Set Chr

MiniRobotLanguage (MRL)

XML.Set Chr
XML.scr

Change a Byte in the XML-Data to a specified Character

Intention

This command is used to change a Single Character inside the XML-Data to a specified other character.
You may need a Reparse, if this change will influence the XML-Structure.
Reparse is done using: rpa - Reparse 4232

```
' These lines are doing the same thing:
' as 62=ASCII of ">"
XML.scr|$$BSA|>
XML.ssc|$$BSA|62
```

Hints:
To specify an ASCII Value instead of a CHaracter, use:
ssc - Set Asc 4280

To replace whole words or Data-Blocks, use
sxp - Set XML Part 4293

Syntax

XML.Set Chr|P2[|P3]

Parameter Explanation

P2 - ***(optional)*** *numeric Byte address of the XML-Data Byte that should be taken as position. If zero, the actual position is taken.*

P3 - ***(optional)*** *Variable* with just one character that is been placed inside the XML-Data at the position which is specified in ***P2.***

Example

```
'*********************************
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'
'*********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:SURLS
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
PRT.------------------------
PRT.This is the block before change:
PRT.------------------------
XML.Print
PRT.------------------------
PRT.Now we use Set-XML-Part to change the block
$$TXT=!!!!!!!! New Block-Text !!!!!!!!!!
XML.sxp|$$BSA|$$BSE|$$TXT
PRT.------------------------
XML.Print
PRT.------------------------
PRT. In this case the Reparse was done internally.
PRT.------------------------
PRT.Now we will make a character change in the XML-Data
XML.scr|$$BSA|>
XML.ReParse
PRT.------------------------
XML.Print
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$BSA-3)|($$BSA+3)
PRT.------------------------
XML.Dump|($$BSB-3)|($$BSB+3)

MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gsc - Get Asc** 4191
- **rpa - Reparse** 4232
- **ssc - Set Asc** 4280
- **sxp - Set XML Part** 4293
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.38 sdn - Search Down Tag Name

MiniRobotLanguage (MRL)

XML.Search Down Tag Name
XML.sdn

Get XML-Data from a file

Intention

This command is used to search a XML-Tag with a specified Tag-Name in downward direction.

It is important that you understand that you can use "Parts of a Name" to specify the name of the Tag you search for.
This may have unwanted side effects that you have to be aware of.
Take this example.

```
<PName>
     <Name>
       Nametext
     </Name>
</PName>
```

Searching like this:

```
XML.sdn|Name
```

will bring up the next such Tag, that can be the Tag

```
</PName>
```

because the name contains the ward "Name".

To specify exactly what you are looking for, you have to use the
3.2 Standard-Search Pattern 341

In this case you would write
`XML.sdn|&EXACT:Name`
to really get what you want.

In the same way you could specify multiple valid words and connect them with `&OR:`
`XML.sdn|{&EXACT:Name}&OR:{&EXACTnombre}`
and use this feature to build international scripts.

Syntax

© 2023 Theo Gottwald * https://www.smart-package.com

XML.Search Down Tag Name|P2[|P3]

Parameter Explanation

P2 – Variable with Tag-Name to search for, use: 3.2 Standard-Search Pattern 341

P3 – Variable to return the result. The result is 0 in case nothing was found.
Otherwise the result is >0 and contains the position of the found Open-Tag "<"..

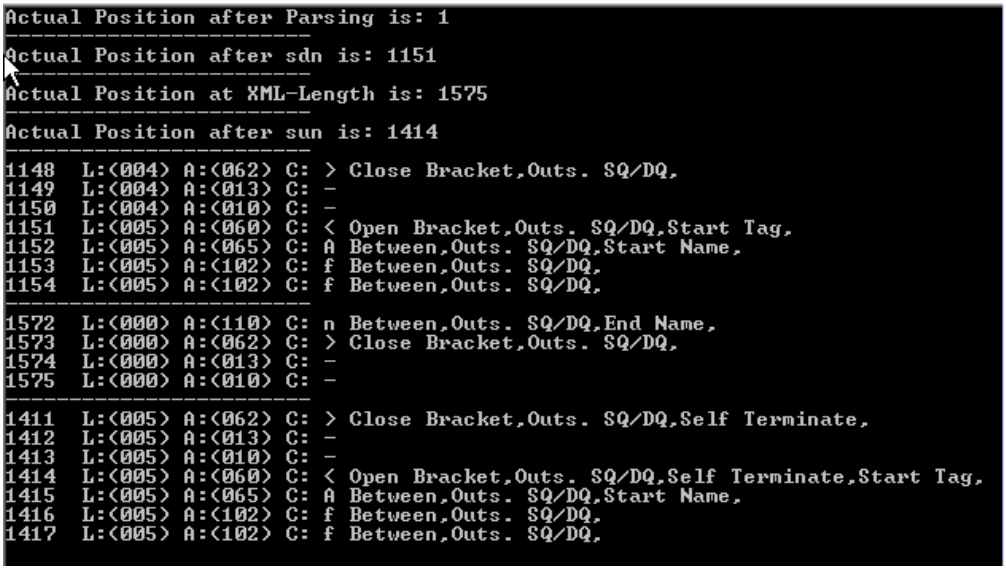
Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|Affec&ANDTHEN:Prod
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.set pos|-1
XML.get pos|$$POB
PRT.Actual Position at XML-Length is: $$POB
PRT.------------------------
XML.sun|Affec&ANDTHEN:Prod
XML.get pos|$$POC
PRT.Actual Position after sun is: $$POC
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+3)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
PRT.------------------------
XML.Dump|($$POC-3)|($$POC+3)
MBX.!
:enx
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- 3.2 Standard-Search Pattern 341
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.39 sid - Search Inside Down

MiniRobotLanguage (MRL)

XML.Search Inside Down
XML.sid

Search XML-Flags in down direction - inside the current tag

Intention

This command is used to search specified XML-Data Flags, starting from the actual position, in downward direction. It will just search inside the current tag.
Any valid closing bracket ">" will end the search.

You can use all these flags, also combinations of these flags (See Link):

! SPR - XML-Parser Flags 3981

Usage:

```
XML.sid|start attribute
XML.get pos|$$POA
PRT.Actual Position is: $$POA
```

`XML.sid` will always start from the actual position.

Syntax

XML.Search Inside Down|P2[|P3]

Parameter Explanation

P2 – *numeric or textual Flag-Description. For Details see here:*
. ! SPR - XML-Parser Flags 3981

P3 – ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that matched all of the specified flags.
*If omitted, the result is been placed on **TOS.***

Example

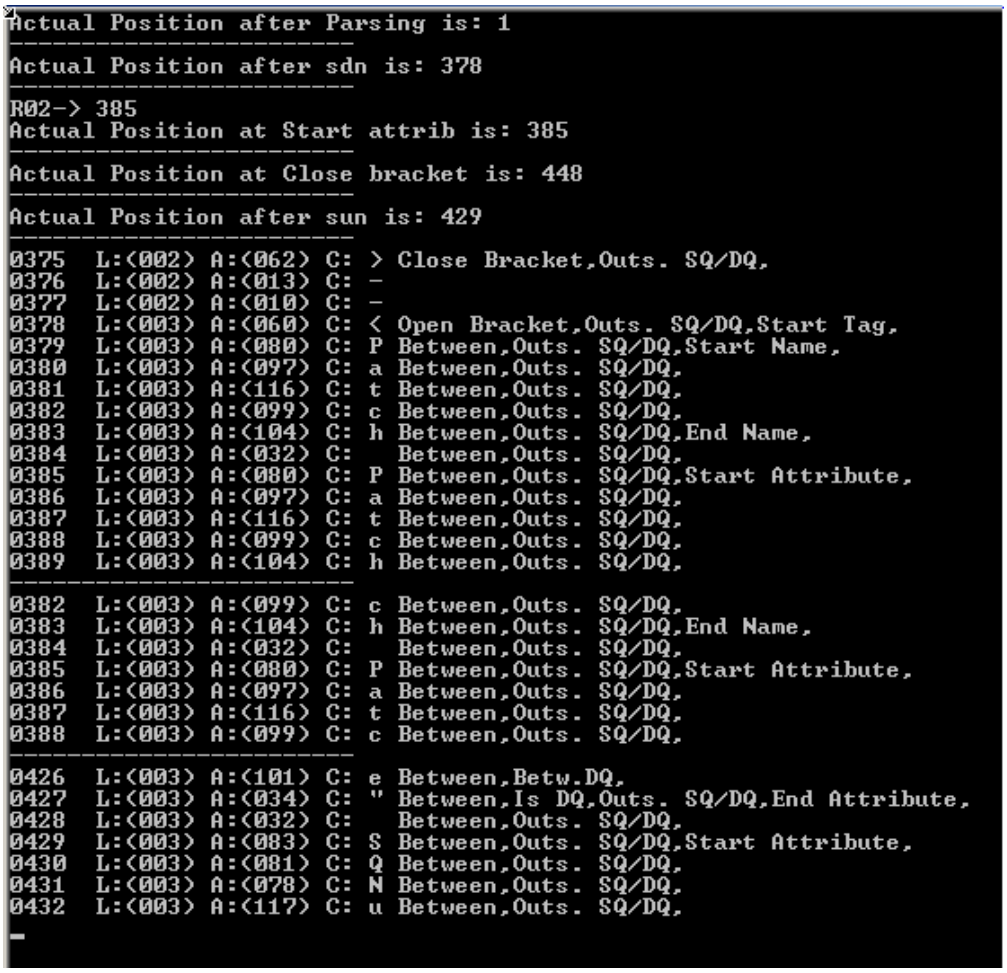
```
'*********************************
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'
'*********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.inc pos
XML.sid|start attribute
XML.get pos|$$POB
PRT.Actual Position at Start attrib is: $$POB
PRT.------------------------
' We go to the Close bracket >
XML.dncb
XML.get pos|$$POS
PRT.Actual Position at Close bracket is: $$POS
PRT.------------------------
XML.dec pos
XML.siu|start attribute
XML.get pos|$$POC
PRT.Actual Position after sun is: $$POC
PRT.------------------------
XML.Dump|($$POA-3)|($$POB+4)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
PRT.------------------------
XML.Dump|($$POC-3)|($$POC+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- 3.2 Standard-Search Pattern 341
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **siu - Search Inside Up** 4254
- **sau - Search All Up** 4239
- **sad - Search All Down** 4236

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.40 siu - Search Inside Up

MiniRobotLanguage (MRL)

XML.Search Inside Up
XML.siu

Search XML-Flags in upward direction - inside the current tag

Intention

This command is used to search specified XML-Data Flags, starting from the actual position, in upward direction. It will just search inside the current tag.
Any valid opening bracket "<" will end the search.

You can use all these flags, also combinations of these flags (See Link):

! SPR - XML-Parser Flags 3981

Usage:

```
XML.siu|start attribute
XML.get pos|$$POA
PRT.Actual Position is: $$POA
```

`XML.siu` will always start from the actual position.

Syntax

XML.Search Inside Up|P2[|P3]

Parameter Explanation

P2 – *numeric or textual Flag-Description. For Details see here:*
. ! SPR - XML-Parser Flags 3981

P3 – ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that matched all of the specified flags.
If omitted, the result is been placed on ***TOS***

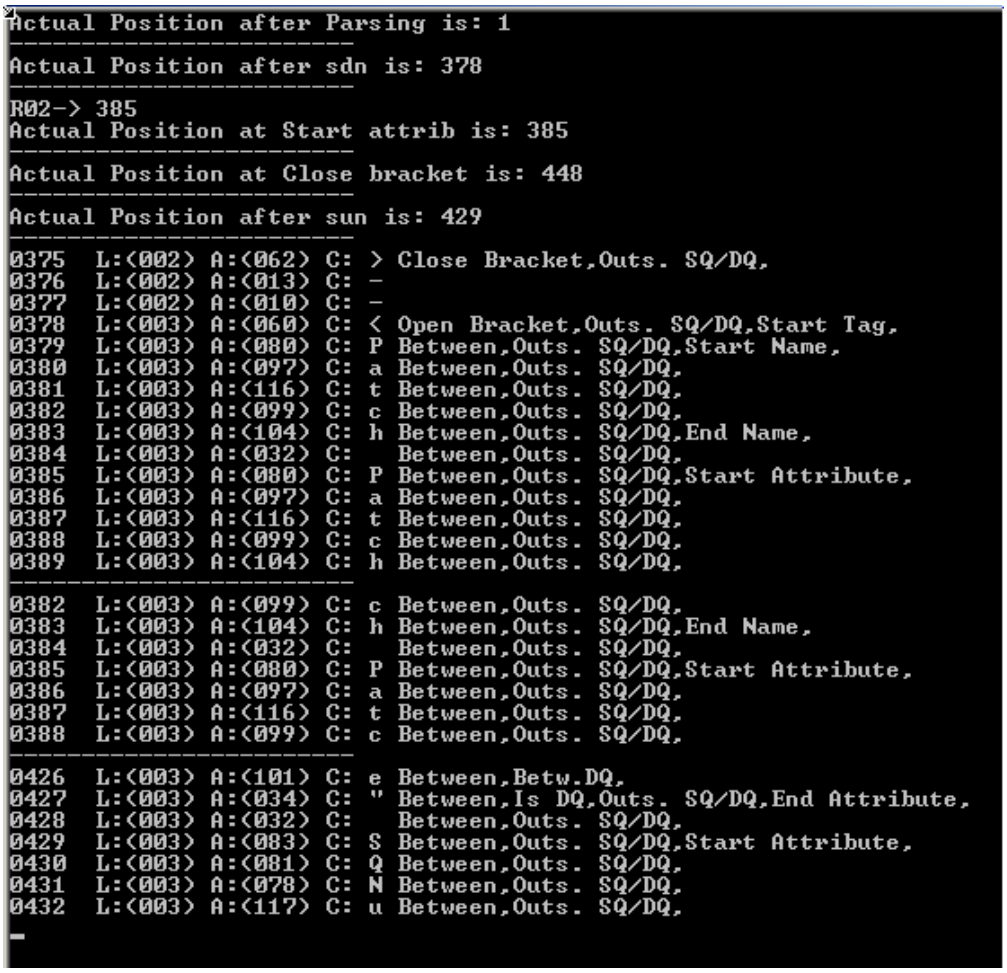
Example

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:Patch
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.inc pos
XML.sid|start attribute
XML.get pos|$$POB
PRT.Actual Position at Start attrib is: $$POB
PRT.------------------------
' We go to the Close bracket >
XML.dncb
XML.get pos|$$POS
PRT.Actual Position at Close bracket is: $$POS
PRT.------------------------
XML.dec pos
XML.siu|start attribute
XML.get pos|$$POC
PRT.Actual Position after sun is: $$POC
PRT.------------------------
XML.Dump|($$POA-3)|($$POB+4)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
PRT.------------------------
XML.Dump|($$POC-3)|($$POC+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **sid - Search Inside Down** 4250
- **sau - Search All Up** 4239
- **sad - Search All Down** 4236

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.41 sna - Search Name and Attribute

MiniRobotLanguage (MRL)

XML.Search Name and Attribute XML.sna

Search specified Tags with specified attributes

Intention

This command is used to search a defined Place in the XML-Data. To specify this place, you can give the Tag-Name and the name of an attribute inside of the Tag.

To search the following Tag:

```
<Patch PatchName="SQLServer" SQNumber="Q2716433">
```

you could use this line:

```
XML.sna|Patch|SQNum
```

For this command you need to provide the name as well as the attribute-Name.
If you want to search only "by Name" you can use the

- sun - Search Up Tag Name 4290
- sdn - Search Down Tag Name 4247

commands.

This command will search from the current position downwards.

If you have several fitting Tags or attributes with the same name, you can specify a count in **P3** and **P4**.

```
XML.sna|Patch|SQNum|2|3
```

In this case, the 2nd Tag with a name and attribute that matches the specification is a valid result.
Only the 3rd attribute that matches the attribute specification will be valid in this case.

It is important to understand that the first counter - in this case **P2** will count the total match of both - name and attribute . while the second counter **P3** only defines the count of the attribute in the actual examined tag.

Example:

```
<Patch PatchName="SQLServer" SQNumber="Q2716433"
SANumber="Q2716433" SCNumber="Q2716433">
```

Assume that you do not know the exact name of the third attribute of the tag you are looking for. You could write:

© 2023 Theo Gottwald * https://www.smart-package.com

```
XML.sna|Patch|S&ANDTHEN:Num|1|3
```

Note the use of 3.2 Standard-Search Pattern 341 that are perfect for these cases. Using the Pattern, any attribute Name that starts with a "S" and then has following the word "Num" will be fine.

Syntax

XML.sna|P2|P3[|P4][|P5]

Parameter Explanation

P2 – Variable with the name of the Tag that should be searched for.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the Tag-Name.

P3 – Variable with the name of the attribute that should be retrieved.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want.

P3 – ***(optional)*** Count. In case there are multiple Tags and attributes with names that fit the description, you can specify the number of the Tag Name that you want.If omitted "1" is the default number.
P3 does not only count "valid Tag Names" but valid results that have a fitting Tag name as well as a fitting attribute name.

P4 – ***(optional)*** Count. In case there are multiple attributes with the same name, you
can specify the number of the attribute that you want. If omitted "1" is the default number.

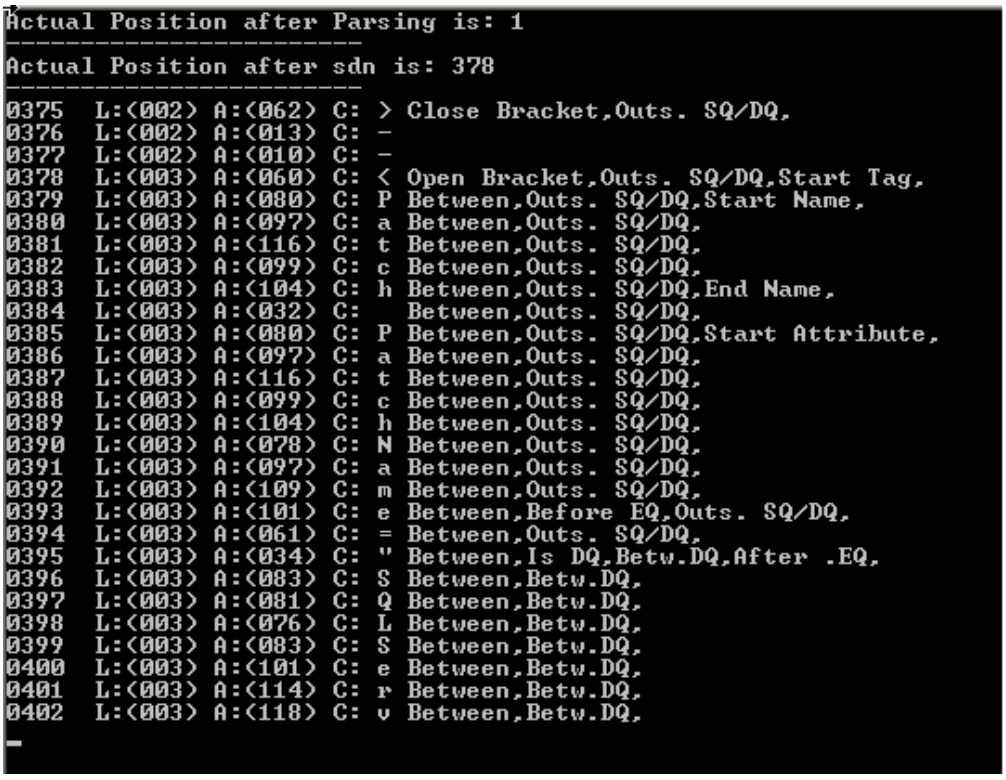
Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.Actual Position after Parsing is: $$POS
' Here we search ...
PRT.------------------------
XML.sna|Patch|SQNumb
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+24)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



Remarks

Currently there is no search in upward direction like `XML.sna`.

Limitations:

-

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- sun - Search Up Tag Name 4290
- sdn - Search Down Tag Name 4247
-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.42 snaup - Search Name and Attribute Up

XML.Search Name and Attribute upwards ***Previous*** 4258 ***Top*** 41 ***Next*** 4267

MiniRobotLanguage (MRL)

XML.Search Name and Attribute up
XML.snaup

Search specified Tags with specified attributes

Intention

This command is used to search a defined Place in the XML-Data. To specify this place, you can give the Tag-Name and the name of an attribute inside of the Tag.

To search the following Tag:

```
<Patch PatchName="SQLServer" SQNumber="Q2716433">
```

you could use this line:

```
XML.snaup|Patch|SQNum
```

For this command you need to provide the name as well as the attribute-Name.
If you want to search only "by Name" you can use the

- sun - Search Up Tag Name 4290
- sdn - Search Down Tag Name 4247

commands.

This command will search from the current position upwards.
To search downwards, use: sna - Search Name and Attribute 4258

If you have several fitting Tags or attributes with the same name, you can specify a count in **P3** and **P4**.

```
XML.sna|Patch|SQNum|2|3
```

In this case, the 2nd Tag with a name and attribute that matches the specification is a valid result.
Only the 3rd attribute that matches the attribute specification will be valid in this case.

It is important to understand that the first counter - in this case **P2** will count the total match of both - name and attribute . while the second counter **P3** only defines the count of the attribute in the actual examined tag.

Example:

```
<Patch PatchName="SQLServer" SQNumber="Q2716433"
SANumber="Q2716433" SCNumber="Q2716433">
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

Assume that you do not know the exact name of the third attribute of the tag you are looking for. You could write:

```
XML.snaup|Patch|S&ANDTHEN:Num|1|3
```

Note the use of 3.2 Standard-Search Pattern 341 that are perfect for these cases. Using the Pattern, any attribute Name that starts with a "S" and then has following the word "Num" will be fine.

Syntax

XML.sna|P2|P3[|P4][|P5]

Parameter Explanation

P2 – Variable with the name of the Tag that should be searched for.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the Tag-Name.

P3 – Variable with the name of the attribute that should be retrieved.
You can and need to use 3.2 Standard-Search Pattern 341 to specify the name of the attribute that you want.

P3 – ***(optional)*** Count. In case there are multiple Tags and attributes with names that fit the description, you can specify the number of the Tag Name that you want.If omitted "1" is the default number.
P3 does not only count "valid Tag Names" but valid results that have a fitting Tag name as well as a fitting attribute name.

P4 – ***(optional)*** Count. In case there are multiple attributes with the same name, you
can specify the number of the attribute that you want. If omitted "1" is the default number.

Example

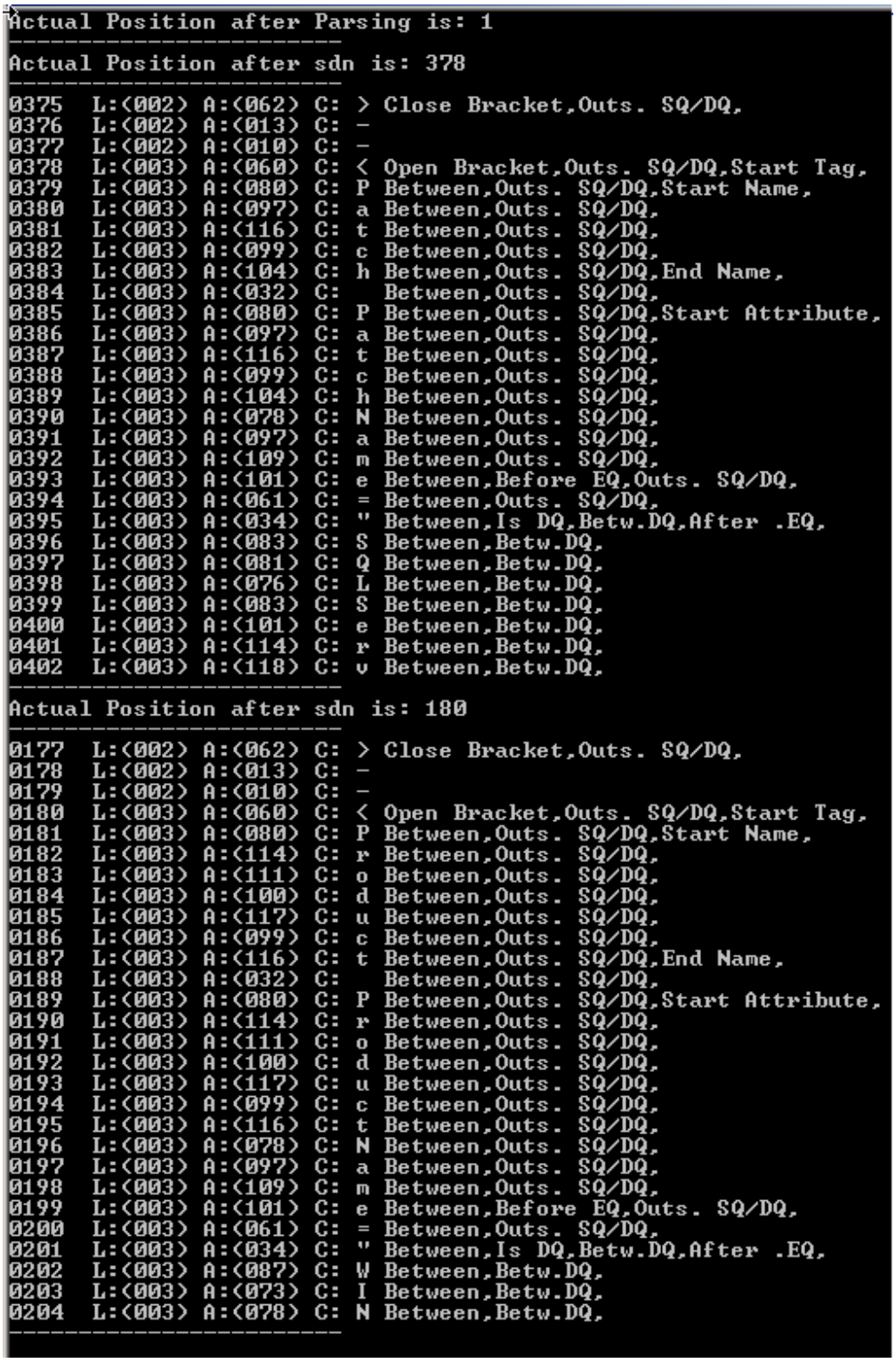
```
'**********************************
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.Actual Position after Parsing is: $$POS
' Here we search ...
PRT.------------------------
XML.sna|Patch|SQNum
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+24)
PRT.------------------------
XML.snaup|Prod|likn
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+24)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

*© 2023 Theo Gottwald * https://www.smart-package.com*

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- sna - Search Name and Attribute 4258
- sun - Search Up Tag Name 4290
- sdn - Search Down Tag Name 4247
-

*© 2023 Theo Gottwald * https://www.smart-package.com*

3.84.8.43 spa - Split Attrib at

MiniRobotLanguage (MRL)

XML.Split Attrib at
XML.spa

Split attribute at actual position into name and value

Intention

This command is used to split an attribute at the current position into name and value.
For this command you need to have the actual position inside of an attribute name.

For this you can use the

- abm - Attrib by Number 4128
- abn - Attrib by Name 4131
- abv - Get Attrib by Value 4135

After these commands, the actual position is on the attributes End.
To get back to the start of the attribute, use:

```
XML.siu|start attribute
```

This will set the actual position to the start of the attribute.Then you can use

```
XML.spa
```

Unlike `XML.spl` you do not give a text string as parameter, but just two variables that will be filled with the name and value of the attribute at the actual position.

Technically the command will look for a "start attribute"-Flag upwards first.
If one is found, that attribute will be taken. If none is found, the command will search an attribute in downward direction. All of these must be inside the Tag at the actual position. This command will not search into other tags, outside of the Tag at the actual position.

Syntax

XML.Split Attrib at[|P2][|P3]

Parameter Explanation

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - ***(optional)*** Variable to return the name of the attribute. If omitted the result is placed on TOS.

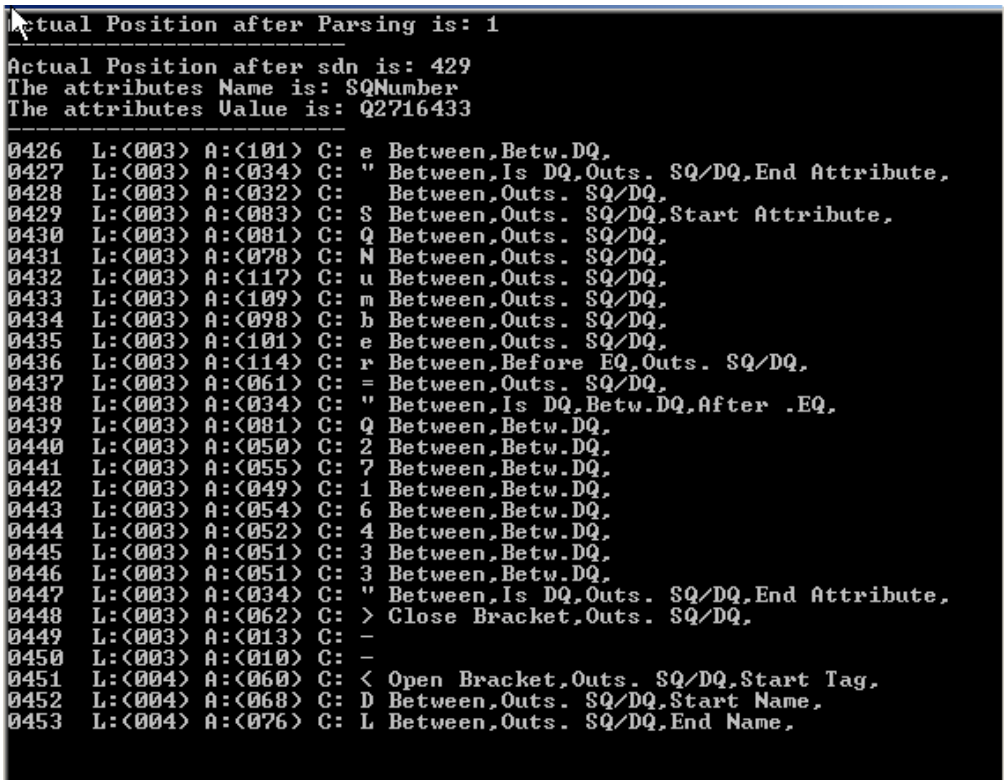
P3 - ***(optional)*** Variable to return the value of the attribute. If omitted this value is placed on TOS (Top of Stack).

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Here we search ...
PRT.------------------------
XML.sna|Patch|SQNum
XML.abn|SQNum
XML.siu|start attribute
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
XML.spa|$$NAM|$$VAL
PRT. The attributes Name is: $$NAM
PRT. The attributes Value is: $$VAL

PRT.------------------------
XML.Dump|($$POA-3)|($$POA+24)
MBX.!
:enx
ENR.
'------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **spl - Split Attrib** 4121
- **sid - Search Inside Down** 4250
- abm - Attrib by Number 4128
- abn - Attrib by Name 4131
- abv - Get Attrib by Value 4135

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.44 spdn - Search Pattern Down

MiniRobotLanguage (MRL)

XML.Search Pattern Down
XML.spdn

Search a textual pattern in downward direction

Intention

This command is a textual search that will search "brute force" through the tags for any matching textual pattern.
As usual, you can use 3.2 Standard-Search Pattern 341 that are perfect for these cases.

Technically this command will start at the next "Start Tag" in downward direction, and then scan inside of "Open Bracket" to "Close Bracket" for the wanted Textual Pattern..

This way you can search any combination of attribute and values or other Tag combinations, like multiple attributes or values.

You can use spup - Search Pattern Up 4276 for the same type of search in reverse direction.

In case the search was successful, the actual position will be changed to the Start-Tag (Open Bracket) of the Tag that contains the textual pattern.

Syntax

XML.spdn|P2[|P3][|P4]

Parameter Explanation

P2 **-** ***Pattern*** to search for. Use 3.2 Standard-Search Pattern 341 or a part of the wanted text.
P3 **-** ***(optional)*** Result, numeric position
P4 **-** ***(optional)*** Count, in case there are multiple fitting matches per Tag.

Example

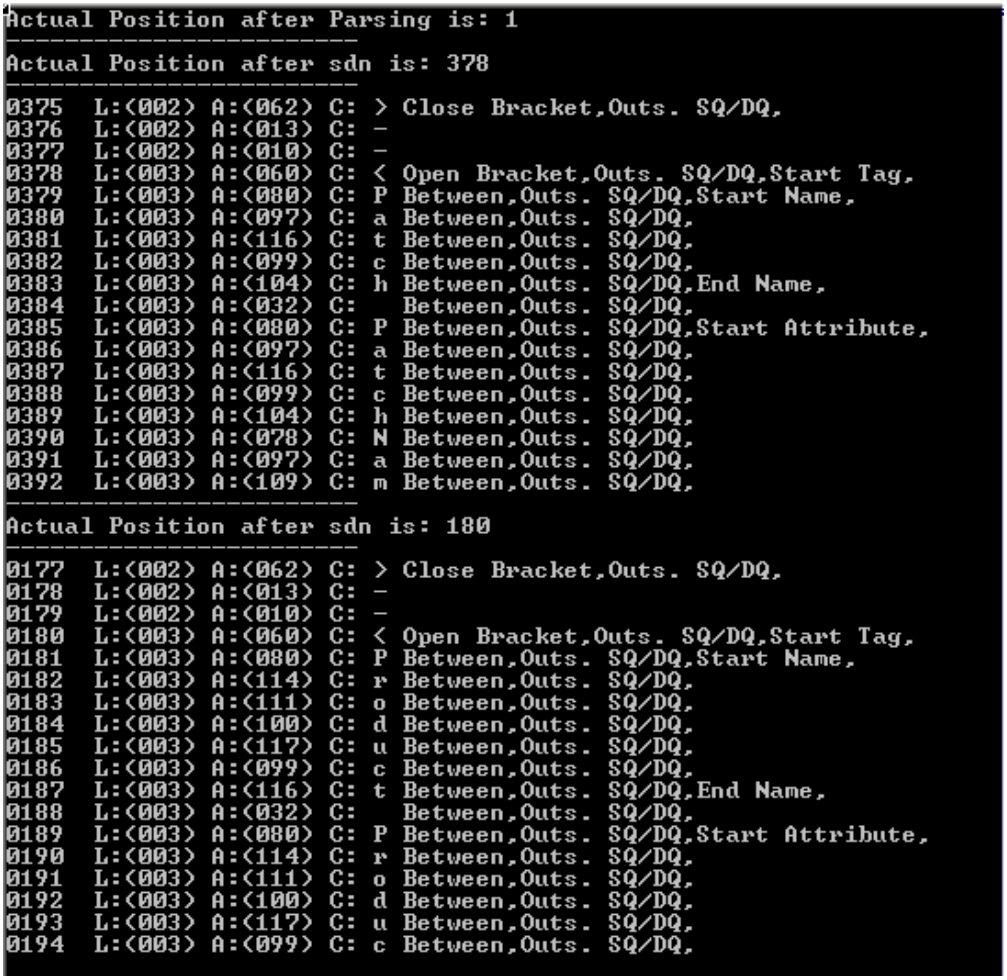
```
'*************************************
'
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Here we search ...
PRT.------------------------
XML.spdn|Sqnumbe
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
GSB.dump
PRT.------------------------
XML.spup|likna
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
GSB.dump
MBX.!
:enx
ENR.
'------------------------------------------------------------
:dump
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+14)
RET.
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- spup - Search Pattern Up 4276
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.45 spo - Set Position

MiniRobotLanguage (MRL)

XML.Set Position
XML.spo

Manually set the actual position (pointer to actual position)

Intention

This command is used to set the actual position to y byte-position in the XML-Data, that starts at (Byte Nr.) 1 and goes up to Byte Nr. "XML-Length".

You can specify a negative position, in this case, the bytes are counted reverse, starting at the end of the XML-Data, that is XML-Length.

```
-1 = Byte "XML-Lenght",
-2 = "XML-Lenght"-1
-3 = "XML-Lenght"-2
```
etc.

To find out where the actual position is, you can use:
gpo - Get Position 4182

Usage:

```
XML.Set Pos|357
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
```

Syntax

XML.Set Position[|P2]

Parameter Explanation

P2 **-** ***(optional)*** numeric parameter that contains a valid position inside the XML-Data.
If omitted or zero, the actual position is used.
If the given number is larger then XML-Length, the number is set to XML-Length.

*© 2023 Theo Gottwald * https://www.smart-package.com*

You can specify a negative position, in this case, the bytes are counted reverse, starting at the end of the XML-Data, that is XML-Length.

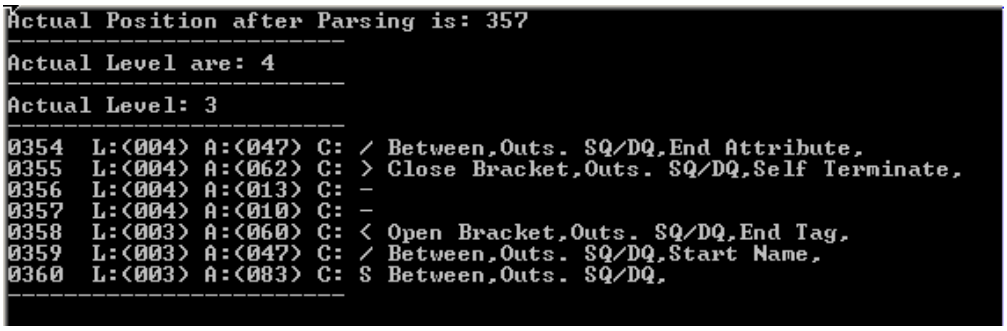
```
-1 = Byte "XML-Lenght",
-2 = "XML-Lenght"-1
-3 = "XML-Lenght"-2
```
etc.

Example

```
'*********************************
'
'*********************************
'
$$XMF=?pfad\Small_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
XML.Set Pos|357
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.gLev
$$TXT=$$000
PRT.Actual Level are: $$TXT
PRT.------------------------
XML.gLev|($$POS+1)|$$ERG
PRT.Actual Level: $$ERG
PRT.------------------------
XML.Dump|($$POS-3)|($$POS+3)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gpo - Get Position** 4182
- **gtr - Goto Root** 4206
- **gfn - Get First Node** 4171

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.46 spup - Search Pattern Up

MiniRobotLanguage (MRL)

XML.Search Pattern Up
XML.spup

Search a textual pattern in upward direction

Intention

This command is a textual search that will search "brute force" through the tags for any matching textual pattern.

As usual, you can use 3.2 Standard-Search Pattern 341 that are perfect for these cases.

Technically this command will start at the next "Start Tag" or "End Tag" in upward direction, And then scan inside of "Open Bracket" to "Close Bracket" for the wanted Textual Pattern..

This way you can search any combination of attribute and values or other Tag combinations, like multiple attributes or values.

You can use spdn - Search Pattern Down 4270 to have the same search for a downward search.

In case the search was successful, the actual position will be changed to the Start-Tag (Open Bracket) of the Tag that contains the textual pattern.

Syntax

XML.spup|P2[|P3][|P4]

Parameter Explanation

P2 - ***Pattern*** to search for. Use 3.2 Standard-Search Pattern 341 or a part of the wanted text.
P3 - ***(optional)*** Result, numeric position
P4 - ***(optional)*** Count, in case there are multiple fitting matches per Tag.

Example

```
'*************************************
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Here we search ...
PRT.------------------------
XML.spdn|Sqnumbe
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
GSB.dump
PRT.------------------------
XML.spup|likna
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
GSB.dump
MBX.!
:enx
ENR.
'------------------------------------------------------------
:dump
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+14)
RET.
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Here we search ...
PRT.------------------------
XML.spdn|Sqnumbe
XML.get pos|$$POA
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.Actual Position after sdn is: $$POA
GSB.dump
PRT.------------------------
XML.spup|likna
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
GSB.dump
MBX.!
:enx
ENR.
'-------------------------------------------------------------
:dump
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+14)
RET.
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
------------------------
Actual Position after sdn is: 378
------------------------
0375  L:(002) A:(062) C: > Close Bracket,Outs. SQ/DQ,
0376  L:(002) A:(013) C: -
0377  L:(002) A:(010) C: -
0378  L:(003) A:(060) C: < Open Bracket,Outs. SQ/DQ,Start Tag,
0379  L:(003) A:(080) C: P Between,Outs. SQ/DQ,Start Name,
0380  L:(003) A:(097) C: a Between,Outs. SQ/DQ,
0381  L:(003) A:(116) C: t Between,Outs. SQ/DQ,
0382  L:(003) A:(099) C: c Between,Outs. SQ/DQ,
0383  L:(003) A:(104) C: h Between,Outs. SQ/DQ,End Name,
0384  L:(003) A:(032) C:   Between,Outs. SQ/DQ,
0385  L:(003) A:(080) C: P Between,Outs. SQ/DQ,Start Attribute,
0386  L:(003) A:(097) C: a Between,Outs. SQ/DQ,
0387  L:(003) A:(116) C: t Between,Outs. SQ/DQ,
0388  L:(003) A:(099) C: c Between,Outs. SQ/DQ,
0389  L:(003) A:(104) C: h Between,Outs. SQ/DQ,
0390  L:(003) A:(078) C: N Between,Outs. SQ/DQ,
0391  L:(003) A:(097) C: a Between,Outs. SQ/DQ,
0392  L:(003) A:(109) C: m Between,Outs. SQ/DQ,
------------------------
Actual Position after sdn is: 180
------------------------
0177  L:(002) A:(062) C: > Close Bracket,Outs. SQ/DQ,
0178  L:(002) A:(013) C: -
0179  L:(002) A:(010) C: -
0180  L:(003) A:(060) C: < Open Bracket,Outs. SQ/DQ,Start Tag,
0181  L:(003) A:(080) C: P Between,Outs. SQ/DQ,Start Name,
0182  L:(003) A:(114) C: r Between,Outs. SQ/DQ,
0183  L:(003) A:(111) C: o Between,Outs. SQ/DQ,
0184  L:(003) A:(100) C: d Between,Outs. SQ/DQ,
0185  L:(003) A:(117) C: u Between,Outs. SQ/DQ,
0186  L:(003) A:(099) C: c Between,Outs. SQ/DQ,
0187  L:(003) A:(116) C: t Between,Outs. SQ/DQ,End Name,
0188  L:(003) A:(032) C:   Between,Outs. SQ/DQ,
0189  L:(003) A:(080) C: P Between,Outs. SQ/DQ,Start Attribute,
0190  L:(003) A:(114) C: r Between,Outs. SQ/DQ,
0191  L:(003) A:(111) C: o Between,Outs. SQ/DQ,
0192  L:(003) A:(100) C: d Between,Outs. SQ/DQ,
0193  L:(003) A:(117) C: u Between,Outs. SQ/DQ,
0194  L:(003) A:(099) C: c Between,Outs. SQ/DQ,
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- spdn - Search Pattern Down 4270
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.47 ssc - Set Asc

MiniRobotLanguage (MRL)

XML.Set Asc
XML.ssc

Change a Byte in the XML-Data to a specified ASCII-value

Intention

This command is used to change a Single Character inside the XML-Data to a specified other character. Unlike with

you can specify the character here, using the ASCII-value.

```
' These lines are doing the same thing:
' as 62=ASCII of ">"
XML.scr|$$BSA|>
XML.ssc|$$BSA|62
```

You may need a Reparse, if this change will influence the XML-Structure.
Reparse is done using: rpa - Reparse 4232

Hints:
To specify an Character instead of an ASCII Value, use:
scr - Set Chr 4242

To replace whole words or Data-Blocks, use
sxp - Set XML Part 4293

Syntax

XML.Set Asc|P2|P3

Parameter Explanation

P2 – numeric parameter that contains a valid position inside the XML-Data.
If omitted or zero, the actual position is used.
If the given number is larger then XML-Length, the number is set to XML-Length.
You can specify a negative position, in this case, the bytes are counted reverse, starting at the end of the XML-Data, that is XML-Length.

© 2023 Theo Gottwald * https://www.smart-package.com

```
-1 = Byte "XML-Lenght",
-2 = "XML-Lenght"-1
-3 = "XML-Lenght"-2
```
etc.

P3 – Single Character (can be in a variable also) that is been placed inside the XML-Data
at the specified position.

Example

```
'*********************************
'
'*********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:SURLS
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
PRT.------------------------
PRT.This is the block before change:
PRT.------------------------
XML.Print
PRT.------------------------
PRT.Now we use Set-XML-Part to change the block
$$TXT=!!!!!!!! New Block-Text !!!!!!!!!!
XML.sxp|$$BSA|$$BSE|$$TXT
PRT.------------------------
XML.Print
PRT.------------------------
PRT. In this case the Reparse was done internally.
PRT.------------------------
PRT.Now we will make a character change in the XML-Data
XML.scr|$$BSA|>
' This line is doing the same again! (62=ASCII of ">" )
XML.ssc|$$BSA|62
XML.ReParse
PRT.------------------------
XML.Print
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$BSA-3)|($$BSA+3)
PRT.------------------------
XML.Dump|($$BSB-3)|($$BSB+3)
MBX.!
:enx
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **rpa - Reparse** 4232
- ssc - Set Asc 4280
- scr - Set Chr 4242
- sxp - Set XML Part 4293

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.48 std - Start Tag Down

MiniRobotLanguage (MRL)

XML.Start Tag Down
XML.std

Move the actual position to the next Start Tag in downward direction

Intention

This command will move the actual position to the next Start Tag in downward direction.

Syntax

XML.Start Tag Down[|P2]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that is on a start-tag.
If nothing was found the result is zero, and the actual position is not changed.
If omitted, the result is been placed on ***TOS***

Example

```
'*********************************

ENR.
```

This is the result of the sample script:

Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
<Product ProductName="WINDOWS NT 4.0" SName="NTW">
<SPS>
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
<SP SPName="SP3"/>
<SP SPName="SP4"/>
<SP SPName="SP5"/>
<SP SPName="SP6"/>
<SP SPName="SP6A"/>
</SPS>
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe"
SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.49 stu - Start Tag Up

MiniRobotLanguage (MRL)

XML.Start Tag Up
XML.stu

Move the actual position to the next Start Tag in upward direction

Intention

This command will move the actual position to the next Start Tag in upward direction.

Syntax

XML.Start Tag Up[|P2]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result. The result is >0 and contains the actual position that is on a start-tag.
If nothing was found the result is zero, and the actual position is not changed.
If omitted, the result is been placed on ***TOS***

Example

```
'*********************************
'
'*********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
PRT.------------------------
XML.sdn|Affec&ANDTHEN:Prod
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.set pos|-1
XML.get pos|$$POB
PRT.Actual Position at XML-Length is: $$POB
PRT.------------------------
XML.sun|Affec&ANDTHEN:Prod
XML.get pos|$$POC
PRT.Actual Position after sun is: $$POC
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+3)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
PRT.------------------------
XML.Dump|($$POC-3)|($$POC+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
------------------------
Actual Position after sdn is: 1151
------------------------
Actual Position at XML-Length is: 1575
------------------------
Actual Position after sun is: 1414
------------------------
1148  L:(004) A:(062) C: > Close Bracket,Outs. SQ/DQ,
1149  L:(004) A:(013) C: -
1150  L:(004) A:(010) C: -
1151  L:(005) A:(060) C: < Open Bracket,Outs. SQ/DQ,Start Tag,
1152  L:(005) A:(065) C: A Between,Outs. SQ/DQ,Start Name,
1153  L:(005) A:(102) C: f Between,Outs. SQ/DQ,
1154  L:(005) A:(102) C: f Between,Outs. SQ/DQ,
------------------------
1572  L:(000) A:(110) C: n Between,Outs. SQ/DQ,End Name,
1573  L:(000) A:(062) C: > Close Bracket,Outs. SQ/DQ,
1574  L:(000) A:(013) C: -
1575  L:(000) A:(010) C: -
------------------------
1411  L:(005) A:(062) C: > Close Bracket,Outs. SQ/DQ,Self Terminate,
1412  L:(005) A:(013) C: -
1413  L:(005) A:(010) C: -
1414  L:(005) A:(060) C: < Open Bracket,Outs. SQ/DQ,Self Terminate,Start Tag,
1415  L:(005) A:(065) C: A Between,Outs. SQ/DQ,Start Name,
1416  L:(005) A:(102) C: f Between,Outs. SQ/DQ,
1417  L:(005) A:(102) C: f Between,Outs. SQ/DQ,
```

Remarks

-

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.50 sun - Search Up Tag Name

MiniRobotLanguage (MRL)

XML.Search Up Tag Name
XML.sun

Get XML-Data from a file

Intention

This command is used to search a XML-Tag with a specified Tag-Name in upward direction.
It is important that you understand that you can use "Parts of a Name" to specify the name of the Tag you search for.
This may have unwanted side effects that you have to be aware of.
Take this example.

```
<PName>
    <Name>
      Nametext
    </Name>
</PName>
```

Searching like this:

```
XML.sun|Name
```

will bring up the next such Tag, that can be the Tag

```
</PName>
```

because the name contains the ward "Name".

To specify exactly what you are looking for, you have to use the
3.2 Standard-Search Pattern 341

In this case you would write
`XML.sun|&EXACT:Name`
to really get what you want.

In the same way you could specify multiple valid words and connect them with `&OR:`
`XML.sun|{&EXACT:Name}&OR:{&EXACTnombre}`
and use this feature to build international scripts.

Syntax

*© 2023 Theo Gottwald * https://www.smart-package.com*

XML.Search Up Tag Name|P2[|P3]

Parameter Explanation

P2 – Variable with Tag-Name to search for, use: 3.2 Standard-Search Pattern 341.

P3 – Variable to return the result. The result is 0 in case nothing was found.
Otherwise the result is >0 and contains the position of the found Open-Tag "<".

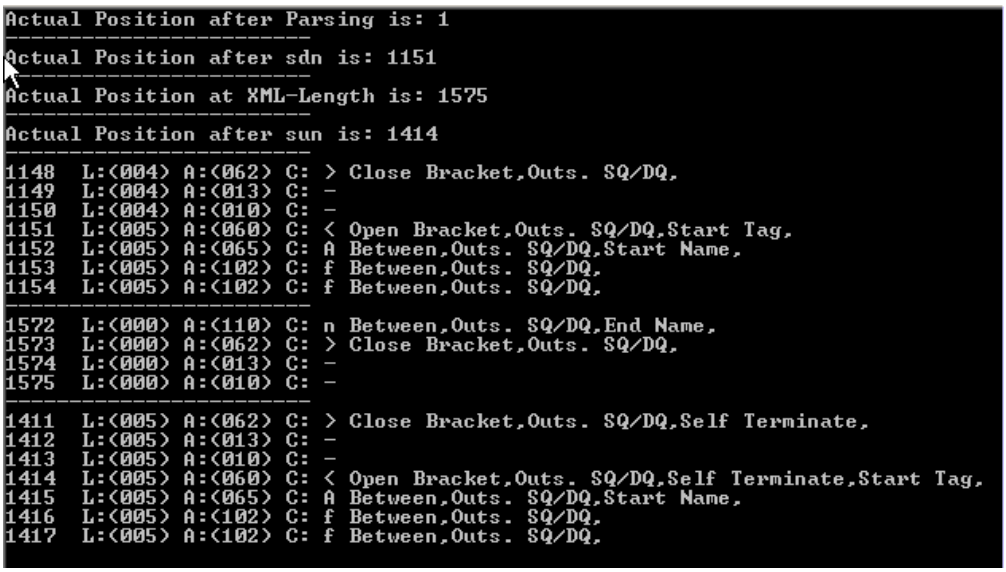
Example

```
'***********************************
'
'***********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|Affec&ANDTHEN:Prod
XML.get pos|$$POA
PRT.Actual Position after sdn is: $$POA
PRT.------------------------
XML.set pos|-1
XML.get pos|$$POB
PRT.Actual Position at XML-Length is: $$POB
PRT.------------------------
XML.sun|Affec&ANDTHEN:Prod
XML.get pos|$$POC
PRT.Actual Position after sun is: $$POC
PRT.------------------------
XML.Dump|($$POA-3)|($$POA+3)
PRT.------------------------
XML.Dump|($$POB-3)|($$POB+3)
PRT.------------------------
XML.Dump|($$POC-3)|($$POC+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
XML.get error|$$ERT
DBP.$$ERT
ENR.
ENR.
```

This is the result of the sample script:



Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.51 sxp - Set XML Part

MiniRobotLanguage (MRL)

XML.Set XML Part
XML.sxp

Remove/Replace Text at a specified place of the XML-Data (and reparse if needed)

Intention

This command is used to change parts of the XML-Data.
You can replace XML-Parts with other parts.

If you just want to remove a part of the XML-Data, then leave **P4** empty.

If you do change the total length of the XML-Data then the Parser will automatically re-parse the data and set all flags from new.
If you do not change the total length of the XML-Data then its up to you, if you want a Re-Parse or not.

See the command: rpa - Reparse 4232 .

Syntax

XML.Set XML Part|P2|P3|P4

Parameter Explanation

P2 – ***numeric Start-position*** of the text that is going to be replaced with the new text in ***P4***. ***If zero, XML-Root is taken.***
If negative, it will count position from the end of the XML-Data backwards.

P3 – ***numeric End-position*** of the text that is going to be replaced with the new text in ***P4***. ***If zero, XML-Length is used.***
If negative, it will count position from the end of the XML-Data backwards.
P3 must always be larger then P2.

P4 – **Text** that is inserted, replacing the text in the XML-Data that is between **P2** and **P3**.

*© 2023 Theo Gottwald * https://www.smart-package.com*

Example

```
'**********************************
'
'**********************************
'
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.sdn|&EXACT:SURLS
XML.get pos|$$POS
PRT.Actual Position is: $$POS
XML.gbl|$$BSA|$$BSE
PRT.Current Block starts at: $$BSA
PRT.Current Block ends at: $$BSE
PRT.------------------------
PRT.This is the block before change:
PRT.------------------------
XML.Print
PRT.------------------------
PRT.Now we use Set-XML-Part to change the block
$$TXT=!!!!!!!! New Block-Text !!!!!!!!!!
XML.sxp|$$BSA|$$BSE|$$TXT
PRT.------------------------
XML.Print
PRT.------------------------
PRT. In this case the Reparse was done internally.
PRT.------------------------
PRT.Now we will make a character change in the XML-Data
XML.scr|$$BSA|>
XML.ReParse
PRT.------------------------
XML.Print
PRT.------------------------
PRT. We Dump Positions
XML.Dump|($$BSA-3)|($$BSA+3)
PRT.------------------------
XML.Dump|($$BSB-3)|($$BSB+3)
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **gxp - Get XML Part** 4209
- **rpa - Reparse** 4232
- ssc - Set Asc 4280
- scr - Set Chr 4242
- ixp - Insert XML-Part 4216
- **sdn - Search Down Tag Name** 4247
- **gbl - Get Block** 4161
-

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.52 upcb - Up Close Bracket

MiniRobotLanguage (MRL)

XML.Up Close Bracket
XML.upcb

Move position to close bracket in upward direction

Intention

This command is used to move the actual position to the next > Close bracket in upward direction.

Usage:

```
' The new position will be in $$ERG
' if nothing found $$ERG=0
XML.upcb|$$ERG

' One Parameter will also work.
XML.upcb
XML.get pos|$$POS
PRT.Actual Position is: $$POS
```

Syntax

XML.Up Close Bracket[|P2]

Parameter Explanation

P2 - ***(optional)*** Variable to return the result, that is the new position or zero, if nothing was found. . The result is 0 in case nothing was found.
If something was found, the actual position will be set to this new position.
If P2 is omitted the result is been placed on TOS.

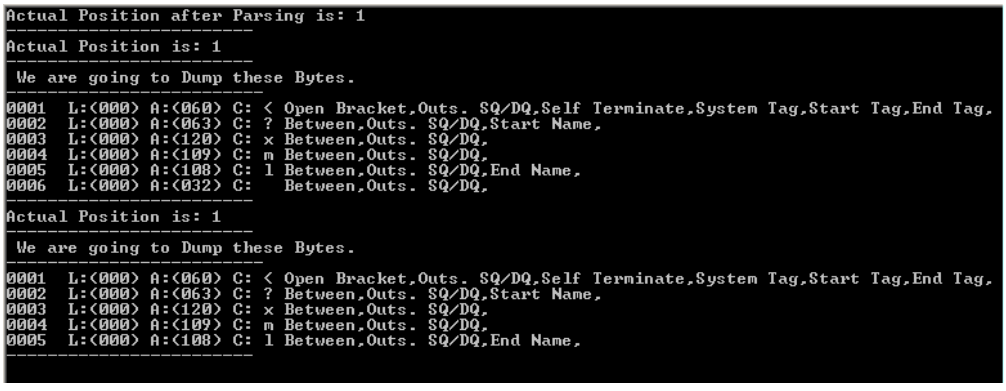
Example

```
'*********************************
'
'*********************************
'Prototype für IML.
$$PRO=Open Bracket
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$TXT=a real Open Bracket
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.dnob|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------
XML.upcb|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:



 © 2023 Theo Gottwald * https://www.smart-package.com

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **upob - Up Open Bracket** 4301
- **dnob - Down Open Bracket** 4155
- **dncb - Down Close Bracket** 4152

© 2023 Theo Gottwald * https://www.smart-package.com

3.84.8.53 upob - Up Open Bracket

MiniRobotLanguage (MRL)

XML.Up Open Bracket
XML.upob

Move position to close bracket in downward direction

Intention

This command is used to move the actual position to the next < Open bracket in upward direction.

Usage:

```
' The new position will be in $$ERG
' if nothing found $$ERG=0
XML.upob|$$ERG

' One Parameter will also work.
XML.upob
XML.get pos|$$POS
PRT.Actual Position is: $$POS
```

Syntax

XML.Up Open Bracket[|P2]

Parameter Explanation

P2 **-** ***(optional)*** Variable to return the result, that is the new position or zero, if nothing was found. . The result is 0 in case nothing was found.
If something was found, the actual position will be set to this new position.
If P2 is omitted the result is been placed on TOS.

Example

```
'**********************************
'
'**********************************
'Prototype für IML.
$$PRO=Open Bracket
$$TXT=a real Open Bracket
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

```
$$XMF=?pfad\SQ_XML.txt
NEF.$$XMF
  MBX.File $$XMF was not found!
  GTO.enx
EIF.
' Now we are going to parse it
XML.Load File|$$XMF
' We check if there are errors
JIT.Lab_Err
' We get the "actual position" - after parsing.
XML.get pos|$$POS
PRT.Actual Position after Parsing is: $$POS
' Hier gehts los
PRT.------------------------
XML.dnob|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.-------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------
XML.upob|$$ERG
XML.get pos|$$POS
PRT.Actual Position is: $$POS
PRT.------------------------
PRT. We are going to Dump these Bytes.
PRT.-------------------------
XML.Dump|($$ERG-5)|($$ERG+5)
PRT.------------------------
MBX.!
:enx
ENR.
'-------------------------------------------------------------
' Error Handler
:Lab_Err
XML.get error|$$ERT
DBP.$$ERT
ENR.
```

This is the result of the sample script:

```
Actual Position after Parsing is: 1
------------------------
Actual Position is: 1
------------------------
 We are going to Dump these Bytes.
------------------------
0001  L:(000) A:(060) C: < Open Bracket,Outs. SQ/DQ,Self Terminate,System Tag,Start Tag,End Tag,
0002  L:(000) A:(063) C: ? Between,Outs. SQ/DQ,Start Name,
0003  L:(000) A:(120) C: x Between,Outs. SQ/DQ,
0004  L:(000) A:(109) C: m Between,Outs. SQ/DQ,
0005  L:(000) A:(108) C: l Between,Outs. SQ/DQ,End Name,
0006  L:(000) A:(032) C:   Between,Outs. SQ/DQ,
------------------------
Actual Position is: 1
------------------------
 We are going to Dump these Bytes.
------------------------
0001  L:(000) A:(060) C: < Open Bracket,Outs. SQ/DQ,Self Terminate,System Tag,Start Tag,End Tag,
0002  L:(000) A:(063) C: ? Between,Outs. SQ/DQ,Start Name,
0003  L:(000) A:(120) C: x Between,Outs. SQ/DQ,
0004  L:(000) A:(109) C: m Between,Outs. SQ/DQ,
0005  L:(000) A:(108) C: l Between,Outs. SQ/DQ,End Name,
0006  L:(000) A:(032) C:   Between,Outs. SQ/DQ,
------------------------
```

Here is the important part of the used XML-Data that is analyzed here:

```
<?xml version="1.0" encoding="UTF-8"?>
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
<!-- comment attrib="Hall"-->
<PatchInformation XMLVersion="1.0000.0000.4918"
XMLLastUpdated="1/30/2010" PatchCount="1261">
<Products>
<Product ProductName="WINDOWS NT 4.0" SName="NTW">
<SPS>
<SP SPName="SP3"/>
<SP SPName="SP4"/>
<SP SPName="SP5"/>
<SP SPName="SP6"/>
<SP SPName="SP6A"/>
</SPS>
</Product>
<Patch PatchName="SQLServer2008-KB2716433-x86.exe"
SQNumber="Q2716433">
<DL SFN="SQLServer2008-KB2716433-x86.exe" FS="188008400">
<SURLS>
```

You can find the XML-Sample-Data file in the Installation folder in the Samples-Folder "XML".

Remarks

-

Limitations:

-

See also:

- ! Smarty's XML-Features 3973
- ! SPR - XML-Parser Flags 3981
- **XML. - eXtended Markup Language** 3990
- **par - parse from var** 4117
- **upcb - Up Close Bracket** 4298
- **dnob - Down Open Bracket** 4155
- **dncb - Down Close Bracket** 4152

© 2023 Theo Gottwald * https://www.smart-package.com

3.85 ZIP. - Handle Archives

3.85.1 ! Smarty's ZIP-Commands

Smarty's ZIP-Commands Previous 937 Top 41 Next 946

Smarty's ZIP Commands

Smarty can work with ZIP-Archives.
You can:

- extract Archives with Folders and files. (`ZIP.`)
- compress complete foiders into a single ZIP-File. (`ZIP.`)
- compress and uncompress Text-Data that is available in a variable. (`ZIP.`)
- Test if a specified **filename or foldername exists inside an Archive** or not. (`IFZ.`)

You can also apply these methods to groups of files, using `FEF`.

*© 2023 Theo Gottwald * https://www.smart-package.com*

If you see this Logo, then you can use "Wildcards" in the Filesystem-Commands.

These are the well-known "*" and "?" - Wildcards.
Using Wildcards in the Pathname is not recommended, use them only in the Filename or extension.
Note that the first file or folder that fits to the wildcard is been taken.
Use Wildcards with extreme caution, as this may lead to unexpected results if there are multiple fitting files.

Here is a code Sample that will do a simple ZIP.- Operation.

```
'
'SPR Script-file: ZIP:-Sample
'Purpose:
'Author: Theo Gottwald
'
'============================================================
'#EXE:?path\
'#SPI:ForceWrite
  $$FIL=?path\readme.txt
  CFF.$$FIL|$$TXT
  FOR.§§LOP|0|10
    ZIP.sdc|§§LOP
    DBP.Compression is §§LOP
    CFF.$$FIL|$$TXT
    LEN.$$TXT|$$LEA
    DBP.Original=$$LEA
    $$TIA=#dtime#
    ZIP.cv|$$TXT|$$TAR
    $$TIB=#dsince#
    CAL.$$TIT=$$TIB|r
    LEN.$$TAR|$$LEB
    DBP.Compressed=$$LEB / Time: $$TIT sec.
    ZIP.uv|$$TAR
    LEN.$$TAR|$$LEC
    DBP.UnCompressed=$$LEC
  NEX.
MBX.!
ENR.
'============================================================
```


© 2023 Theo Gottwald * https://www.smart-package.com

3.85.2 IFZ. / NFZ. - If File in ZIP

MiniRobotLanguage (MRL)

IFZ./NFZ. Conditional Statement

If Exist File/Folder in Zip-Archive

Intention

Conditional Statement. Test if a specified **filename or foldername exists inside an Archive** or not.
To specify an Folder-Name, suffix it with an "\.

Example:

```
$$FOL=Target\
```

You can specify additional Parameters to test if something is a file or a dir.

To search for a file that is inside an Subfolder, you have to specify the whole path of the file as it is inside the Archive.

Example:

```
$$FIB=Testing\readme2.txt
```

When you specify the filename / directory path, the robot will check if the last sign of the name/path is a \ (backslash). If the last sign is a backlash, then the command will check for a directory (NOT for a file!). If the last sign is NOT a backslash, the command will check for a file NOT for a directory.

```
' You can use a wildcard-sign from
' ? - single character
' * - one or more characters
' # - single digit
'
$$FIA=Document.*
$$FIB=?path\Target.zip

IFZ.$$FIA|$$FIB
  MBX. File $$FIB xistiert in ZIP-Archiv
ELS.
  MBX. File $$FIB xistiert NICHT in ZIP-Archiv
EIF.
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

You can use NFZ.(Not Exists File in ZIP) as the negative Form of the command.

Syntax

IFZ.P1|P2 ... ELS. ... EIF.

Parameter Explanation

P1 **-** Variable/filename/path that should be searched for.
Pathes must be ending with a \ (backslash)

```
' You can use a wildcard-sign from these
' ? - single character
' * - one or more characters
' # - single digit
```

P2 **-** The path to the Archive (ZIP-File) that should be examined.

Example

```
'*********************************************
' IFZ-Sample
'*********************************************
§§FIA=?path\Target.zip
§§FIB=*readme2.txt

IFZ.§§FIB|§§FIA
  MBX. File §§FIB xistiert in ZIP-Archiv
ELS.
  MBX. File §§FIB xistiert NICHT in ZIP-Archiv
EIF.

END.
```

Remarks

IFZ. and other IXX. can be nested to unlimited Depth, and they can also enclose Sub-Program Calls or FEX. (Enumerations) to unlimited Depth.

Limitations:

The use of wildcards should be limited to one wildcard-character.

See also:

© 2023 Theo Gottwald * https://www.smart-package.com

- ZIP. - Handle Archives 4309
- cf - compress file 4313
- zf - Zip Folder 4332
- uf - unzip file/folder 4326
- cv - compress var 4318
- uv - uncompress var 4329
- IFZ. / NFZ. - If File in ZIP 4306
- sdk - set default compression 4323
- IEF. / NEF. - If - exist - File 1050

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.3 ZIP. - Handle Archives

MiniRobotLanguage (MRL)

ZIP. Command

Compress or uncompress Files / Archives

Intention

The `ZIP.`-Command can be used to compress or uncompress files or folders to or from a ZIP-File. It can be used to extract files from a Standard ZIP-files that has been created using another ZIP-Tool.

Possible Uses:
You can add a single ZIP-FIle to your Executable and unzip it at RunTime.
Or download a ZIP-File from WEB and unpack it at Runtime.
You can have zipped Archives on a network share and load and install them on clients in your network.

`ZIP.` can be used to extract Archives with Folders and files.
`ZIP.` can be used to compress complete foiders into a single ZIP-File.
`ZIP.` can be used to compress and uncompress Text-Data that is available in a variable.

Limitations:
`ZIP.` can be used to compress complete folders into a single ZIP-File.
In this case, empty Sub-Folders are ignored and not included in the archive.

`ZIP.` can be used to compress and uncompress Text-Data that is available in a variable.
In this case the result is in a own format, and therefore not compatible with any unzip, other then the ZIP.-Command itself.

Note that you can split very large files using these commands before "Zipping" them:

- **FIL.Split File** 1001
- **FIL.Join File** 994 **and**
- **FIL.Join Delete** 990

Its recommended that you turn the builtin compression in the `FIL.Split File` command off, before using ZIP on these files. Otherwise the ZIP compression can not be effective because these files would already be compressed.

Syntax

ZIP.P1|P2[|P3][|P4][|P5]

*© 2023 Theo Gottwald * https://www.smart-package.com*

Change Settings etc.:

"sdc", "set default compression" - used to set the compression / speed ratio for all compress-operations. The Ratio is been set using a numeric value between 0 and 10.

```
0 -> use default compression
1 -> no compression
10 -> maximum compression
```

"gtfn","get zip filename" - used to create a filename with the ending ".zip" from another filename or foldername.

```
Listing.txt -> Listing.zip
```

Pack and Unpack Zipped Archives:

"zf","zip folder" - used to compress a complete folder with all subfolders into a single "zip"-Archive. The ZIP-File is compatible to any standard ZIP-Tool. Any empty Folders in teh Source folder will be ignored.

```
ZIP.zf|$$FOL[|...P5]
```

"cf","compress file" - used to create a file with the ending ".zip" from another filename.
Will automatically generate the Target Filename. If the Target "zip"-File exists, the File is added to that archive. Note that ZIP-Archives can hold multiple times the same filename, therefore Files inside the Archive are not overwritten if you add a file with the same name.
Unlike "**cftf**" will only take one Parameter:

```
ZIP.cf|$$FIL
ZIP.cf|C:\File.txt|c:\File.zip
```

"uf","unzip file" - uncompress / unzip a ZIP-File to a destination folder. Can be used to uncompress complete ZIP-Archives including all Subfolders. Generally the folder-structure of the ZIP-Archive is preserved, with the exception that empty folders, from inside the zip-file are ignored. Is the unpack command for both of the above commands "**zf**" and "**cf**".

```
ZIP.uf|$$FIL|?path\Extracted_Zip\
```

(Un-)Compress Data:

"cv","compress var" - used to compress the content of a Variable. The result van be uncompressed only using "**uv**". The result is not compatible with the standard "zip" Format.

"uv","uncompress var" - used to uncompress the content of a Variable. The source must have been compressed using "**cv**". The command is not compatible with the standard "zip" Format.

"**deb**","**debug exe**" - this is a special case that takes no parameter. Called from an compiled executable, it will ZIP the actual Runtime-Folder, and place it as a file named "Debug.zip" on the common desktop.

```
' Its a shortcut for this:
§§FIF=?path\
§§FIB=?cdesktop\Debug.zip
ZIP.zf|§§FIF|§§FIB|*.*|sho
```


© 2023 Theo Gottwald * https://www.smart-package.com

Parameter Explanation

P1 **-** Prototype, one from, the following:

P2 **-** P4 Parameter depending on Type of Operation

Example

```
'************************************************
' Sample: ZIP.
'************************************************
'
$$FIL=?path\readme.txt
CFF.$$FIL|$$TXT
FOR.§§LOP|0|10
  ZIP.sdc|§§LOP
  DBP.Compression is §§LOP
  CFF.$$FIL|$$TXT
  LEN.$$TXT|$$LEA
  DBP.Original=$$LEA
  $$TIA=#dtime#
  ZIP.cv|$$TXT|$$TAR
  $$TIB=#dsince#
  CAL.$$TIT=$$TIB|r
  LEN.$$TAR|$$LEB
  DBP.Compressed=$$LEB / Time: $$TIT sec.
  ZIP.uv|$$TAR
  LEN.$$TAR|$$LEC
  DBP.UnCompressed=$$LEC
NEX.
ENR.
```

Remarks

I get these3 values:

Original=94580480 Bytes

default Compression is 0
Compressed=590448 / Time: 1.37 sec.

Compression is 1
Compressed=94594916 / Time: 1.22 sec.

Compression is 2
Compressed=2764314 / Time: .85 sec.

Compression is 3
Compressed=2355309 / Time: .89 sec.

Compression is 4

© 2023 Theo Gottwald * https://www.smart-package.com

Compressed=2394226 / Time: .89 sec.

Compression is 5
Compressed=1258712 / Time: 1.39 sec.

Compression is 6
Compressed=622451 / Time: 1.38 sec.

Compression is 7
Compressed=590448 / Time: 1.34 sec.

Compression is 8
Compressed=560951 / Time: 1.4 sec

Compression is 9
Compressed=552086 / Time: 1.36 sec.

Compression is 10
Compressed=552086 / Time: 1.41 sec.

Limitations:

`ZIP.` can be used to compress complete foiders into a single `ZIP`-File.
In this case, empty Sub-Folders are ignored and not included in the archive.

`ZIP.` can be used to compress and uncompress Text-Data that is available in a variable.
In this case the result is not compatible with any unzip other then the `ZIP.`-Command here in the SPR.

See also:

- sdk - set default compression 4323
- IFZ. / NFZ. - If File in ZIP 4306
- cf - compress file 4313
- uf - unzip file/folder 4326
- uv - uncompress var 4329
- cv - compress var 4318
  - FIL.Split File 1001
  - FIL.Join File 994
- FIL.Join Delete 990

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.4 cf - compress file

MiniRobotLanguage (MRL)

ZIP.compress file

Compress the content of a file

Intention

The `ZIP.cf` - Command can be used to used to compress the content of a file.
This command will create a Standard ZIP-File that is compatible with the ZIP-Format.

There are generally two ways of usage.

```
' Give Source- and Target filename
ZIP.cf|$$TXT|$$TAR
```

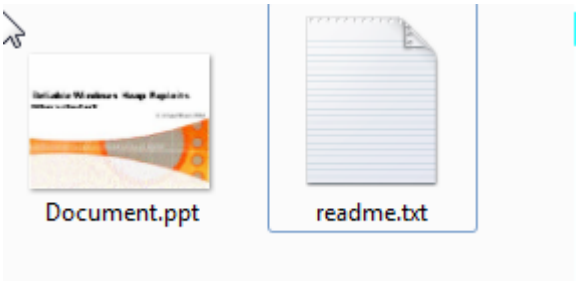
or

```
' ZIP the file and created a ZIP-Filename automatically
$$ZIP=?path\Myfile.txt
ZIP.cf|$$ZIP

' a ZIP-File wit the name "Myfile.zip" is been created in the
' ?path\ folder where the source file was.
```

You can find out about this constructed filename using the "**gtfn**" prototype.

```
$$FIA=?path\Test\Document.ppt
$$FIB=?path\Test\readme.txt
$$FID=?path\Test\Target.zip
```

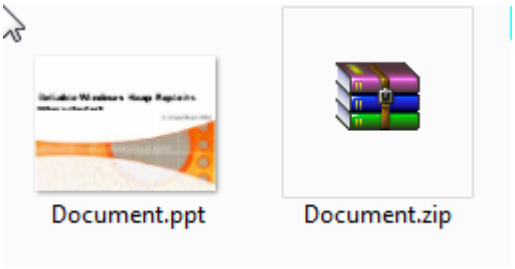
Here is a picture how it looks practically.



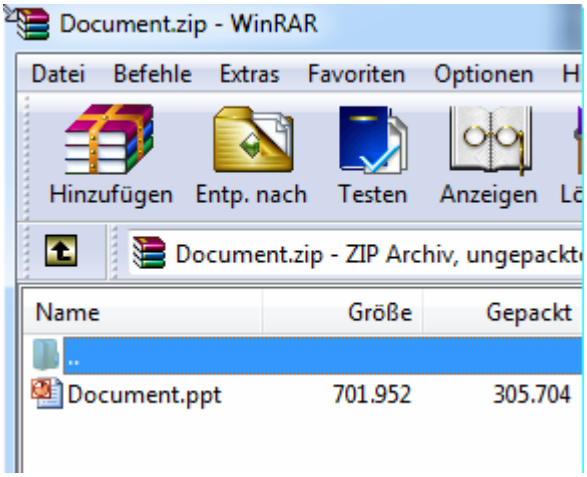

```
' Create a new "Document.zip" File
ZIP.cf|$$FIA
```

This is what we get:

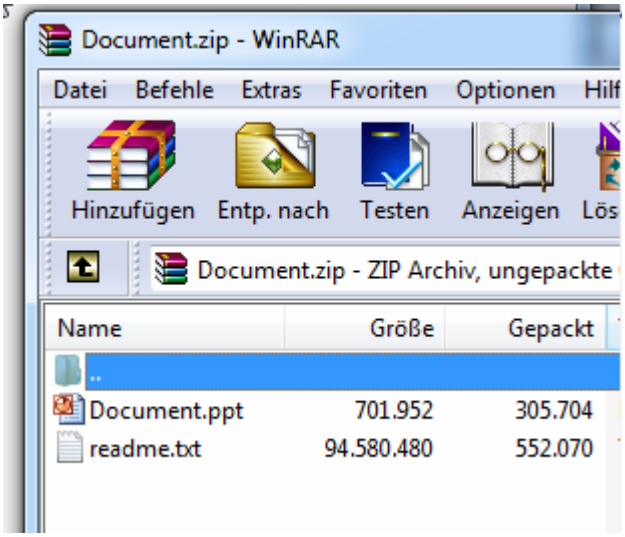
© 2023 Theo Gottwald * https://www.smart-package.com



Looking into the File we can see this*:



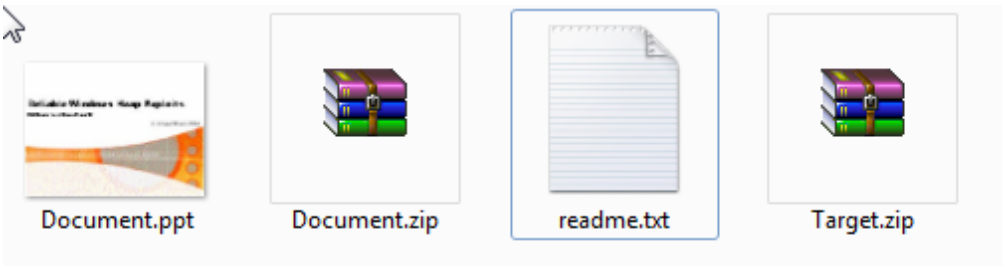

```
' Find out the filename of the new Archive
ZIP.gtfn|§§FIA|§§FIX

' §§FIX will be the filename of the new ZIP-Archive

' Now we add a second File into the ZIP-File
ZIP.cf|§§FIB|§§FIX
```



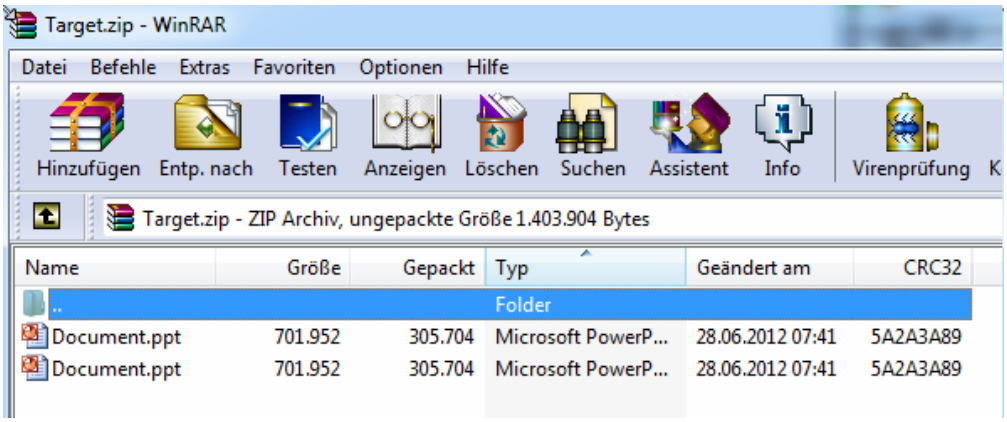
Now we are going to create a second ZIP-File using the **P3** Parameter.
This way we specify the Target-Filename.

© 2023 Theo Gottwald * https://www.smart-package.com

```
' Creat e new ZIP-File §§FID
ZIP.cf|§§FIA|§§FID

' Add a file to §§FID
ZIP.cf|§§FIA|§§FID
```



The picture also shows that you can have multiple time the same file into a single ZIP-Archive.
Which is a not-recommended feature.



The "**cf**","**zf**" and the "**uv**" command are compatible with standard ZIP-Format.
If the Target-Zip File already exists, the File is added to the ZIP-File.

* We have used "WinRar to look into these files. WinRar is an external Program and not Part of the SPR Distribution.For more Infos on WinRar see
http://winrar-rog.de/

Syntax

ZIP.cf|P2[|P3]

The "**cf**" and the "**uf**" command are compatible with standard ZIP-Format.

Parameter Explanation

P1 - "cf" or "compress file"

© 2023 Theo Gottwald * https://www.smart-package.com

P2 - Variable with the filename of the file that is to be compressed. If **P3** is missing, a ZIP-FIle is been created like this:

```
ZIP.cf|Myfile.txt
' a ZIP-File wit the name "Myfile.zip" is been created in the
' same folder where the source file was.
```

You can find out about this filename using the "gtfn" prototype.

P3 - Variable with the target file-name and path.

Example

```
'*********************************************
' Sample 1: ZIP.
'*********************************************
$$FIA=?path\Document.ppt
$$FIB=?path\readme.txt
$$FID=?path\Target.zip

' Create a new "Document.zip" File
ZIP.cf|$$FIA

' Find out the filename of the new Archive
ZIP.gtfn|$$FIA|$$FIX

' Add a second File into the ZIP-File
ZIP.cf|$$FIB|$$FIX

' Creat e new ZIP-File $$FID
ZIP.cf|$$FIA|$$FID

' Add a file to $$FID
ZIP.cf|$$FIA|$$FID

ENR.
```

Remarks

The "**cf**","**zf**" and the "**uf**" command are compatible with standard ZIP-Format.
If the Target-Zip File already exists, the File is added to the ZIP-File.

Limitations:

There is a size limitation that depends on the available 32-bit Memory in the WOW64.
Its somewhere between 1 GB and 2 GB.

© 2023 Theo Gottwald * https://www.smart-package.com

See also:

- ZIP. - Handle Archives 4309
- IFZ. / NFZ. - If File in ZIP 4306
- uf - unzip file/folder 4326
- uv - uncompress var 4329
- cv - compress var 4318
- sdk - set default compression 4323

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.5 cv - compress var

MiniRobotLanguage (MRL)

ZIP.compress var

Compress the content of a variable

Intention

The `ZIP.cv` - Command can be used to used to compress the content of a variable. There are generally two ways of usage.

```
' Give Source- and Target variable
ZIP.cv|$$TXT|$$TAR
```

or

```
' ZIP the Content of the given variable
ZIP.cv|$$TXT
```

The "**cv**" and the "**uv**" command are not compatible with standard ZIP-Format.

Syntax

ZIP.cv|P2[|P3]

The "cv" and the "uv" command are not compatible with standard ZIP-Format.

Parameter Explanation

P1 - "cv" or "compress var"

P2 - Variable with Content to be compressed. If P3 is missing, Result is placed in this variable.

P3 - Variable for the compressed result.

Example

```
'*********************************************
' Sample 1: ZIP.
'*********************************************
'
$$FIL=?path\readme.txt
CFF.$$FIL|$$TXT
FOR.$$LOP|0|10
  ZIP.sdc|$$LOP
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
  DBP.Compression is §§LOP
  CFF.$$FIL|$$TXT
  LEN.$$TXT|$$LEA
  DBP.Original=$$LEA
  $$TIA=#dtime#
  ZIP.cv|$$TXT|$$TAR
  $$TIB=#dsince#
  CAL.$$TIT=$$TIB|r
  LEN.$$TAR|$$LEB
  DBP.Compressed=$$LEB / Time: $$TIT sec.
  ZIP.uv|$$TAR
  LEN.$$TAR|$$LEC
  DBP.UnCompressed=$$LEC
NEX.
ENR.

'*********************************************
' Sample 2: ZIP.
'*********************************************
'
$$FIL=?path\readme.txt
CFF.$$FIL|$$TXT
LEN.$$TXT|$$LEA
DBP.Original=$$LEA
$$TIA=#dtime#
ZIP.cv|$$TXT|$$TAR
$$TIB=#dsince#
CAL.$$TIT=$$TIB|r
LEN.$$TAR|$$LEB
DBP.Compressed=$$LEB / Time: $$TIT
$$TIA=#dtime#
ZIP.cv|$$TXT
$$TIB=#dsince#
CAL.$$TIT=$$TIB|r
LEN.$$TXT|$$LEB
DBP.Compressed=$$LEB / Time: $$TIT

'*********************************************
' Sample 3: ZIP.
'*********************************************
'
$$FIL=?path\readme.txt
$$FIB=?path\readme_2.txt

: %LenCs 1
GMD.|§§§01|§§CSA
LEN.§§§01|§§LEA
DBP.Original=§§LEA * Checksum: §§CSA
END%

' Loading Textfile into Variable
CFF.$$FIL|$$TXT
%LenCs $$TXT

'Compressing COntent of Variable
$$TIA=#dtime#
ZIP.cv|$$TXT|$$TAR
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$TIB=#dsince#
CAL.$$TIT=$$TIB|r
%LenCs $$TAR

' Uncompressing content of variable
$$TIA=#dtime#
ZIP.uv|$$TAR
$$TIB=#dsince#
MBX.$$TAR|$$TAR
CAL.$$TIT=$$TIB|r
%LenCs $$TAR

' Textdatei unter anderem Namen herausschreiben
CTF.$$FIB|$$TAR
ENR.
```

Remarks

The "cv" and the "uv" command are not compatible with standard ZIP-Format.

Limitations:

There is a size limitation that depends on the available 32-bit Memory in the WOW64.
Its somewhere between 1 GB and 2 GB.

See also:

- ZIP. - Handle Archives 4309
- uv - uncompress var 4329
- sdk - set default compression 4323

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.6 gtfn - get zip filename

MiniRobotLanguage (MRL)

ZIP.get Zip Filename

Compute an filename ending in ".zip"

Intention

The `ZIP.gtfn` - Command can be used to compute the Target-Filename of an ZIP.-Operation.
You just call the command with the Source-Path of the File or Folder to be Zipped.
As a result you will get the Target Filename of the ZIP-File that will be created.

```
' Only give the Path of the folder to ZIP.
ZIP.gtfn|$$FIF|$$FIP
' Will display the Filename that is expected to be used by a
following
' ZIP.cf|$$FIP - Operation
MBX.$$FIP
```

In this case the Folder-Name will be taken to construct the Name of the ZIP-Archive.
This operation is a pure String-Operation, it does not make any changes on files.

The "**cf**","**zf**" and the "**uv**" command are compatible with standard ZIP-Format.

Syntax

ZIP.gtfn|P2[|P3]

Parameter Explanation

P1 - "gtfn" or "Get Zip Filename"

P2 - Variable with the path or the filneame of the file or folder that is to be compressed with a later command.

P3 - (optional) Variable with the target file-name and path.
If omitted the result is placed on TOS.

Example

```
'*********************************************
' Sample 1: ZIP.
'*********************************************
ZIP.gtfn|$$FIF|$$FIP
MBX.$$FIP
```

 *© 2023 Theo Gottwald * https://www.smart-package.com*

ENR.

Remarks

This operation is a pure String-Operation, it does not make any changes on files.

Limitations:

-

See also:

- ZIP. - Handle Archives 4309
- uf - unzip file/folder 4326
- uv - uncompress var 4329
- cv - compress var 4318
- sdk - set default compression 4323
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.7 sdk - set default compression

MiniRobotLanguage (MRL)

ZIP.set default compression

Set the default compression ratio for ZIP Operations

Intention

The `ZIP.sdc` - Command can be used to used to set the compression / speed ratio for all compress-operations. The Ratio is been set using a numeric value between 0 and 10.

```
0 -> use default compression
```

These are the compression ratios, the higher, the stronger the compression.

```
1 -> no compression
...
7 -> good compression
10 -> maximum compression
```

Here are some sample values to give you an idea what happens if you specify another value.

```
Original=94580480 Bytes

default Compression is 0
Compressed=590448 / Time: 1.37 sec.

Compression is 1 (No compression, just pack)
Compressed=94594916 / Time: 1.22 sec.

Compression is 2 (remove redundances)
Compressed=2764314 / Time: .85 sec.

Compression is 3
Compressed=2355309 / Time: .89 sec.

Compression is 4
Compressed=2394226 / Time: .89 sec.

Compression is 5
Compressed=1258712 / Time: 1.39 sec.

Compression is 6
Compressed=622451 / Time: 1.38 sec.

Compression is 7
Compressed=590448 / Time: 1.34 sec.

Compression is 8
Compressed=560951 / Time: 1.4 sec
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
Compression is 9
Compressed=552086 / Time: 1.36 sec.

Compression is 10 (maximum compression)
Compressed=552086 / Time: 1.41 sec.
```

Syntax

ZIP.sdc|P2

"sdc", "set default compression" - used to set the compression / speed ratio for all compress-operations. The Ratio is been set using a numeric value between 0 and 10.

```
0 -> use default compression
1 -> no compression
10 -> maximum compression
```

Parameter Explanation

P1 - "sdc" or "set default compression"

P2 - Value from 0 to 10 or numeric variable

Example

```
'***********************************************
' Sample: ZIP.
'***********************************************
'
$$FIL=?path\readme.txt
CFF.$$FIL|$$TXT
FOR.§§LOP|0|10
  ZIP.sdc|§§LOP
  DBP.Compression is §§LOP
  CFF.$$FIL|$$TXT
  LEN.$$TXT|$$LEA
  DBP.Original=$$LEA
  $$TIA=#dtime#
  ZIP.cv|$$TXT|$$TAR
  $$TIB=#dsince#
  CAL.$$TIT=$$TIB|r
  LEN.$$TAR|$$LEB
  DBP.Compressed=$$LEB / Time: $$TIT sec.
  ZIP.uv|$$TAR
  LEN.$$TAR|$$LEC
  DBP.UnCompressed=$$LEC
NEX.
ENR.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Remarks

On modern computers, you can most often just use the highest compression without a large time penalty.

Limitations:

–

See also:

- ZIP. - Handle Archives 4309
- cf - compress file 4313
- zf - Zip Folder 4332
- uf - unzip file/folder 4326
- cv - compress var 4318
- uv - uncompress var 4329
- IFZ. / NFZ. - If File in ZIP 4306

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.8 uf - unzip file/folder

MiniRobotLanguage (MRL)

ZIP.unzip file/folder

UnZip the content of a file or a complete Archive with Folders

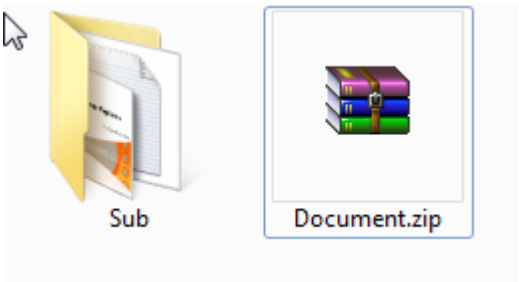
Intention

The `ZIP.uf` - Command can be used to unZip (unpack) the content of a Standard ZIP-file, that is compatible with the ZIP-Format.
These Archives can be created with the ZIP.-Command "**`zf`**" or "**`cf`**" or with any available other Standard ZIP-Compression Utility.

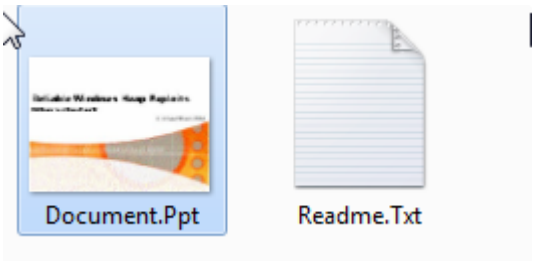
It can be used in two ways:

```
' This will extract the content of the Archive
' into ?path.
$$FID=?path\Test\Document.zip
ZIP.uf|$$FID

or

' Give Source-Filename and Target Folder
$$FID=?path\Test\Document.zip
$$FIP=?path\Test\Sub
ZIP.uf|$$FID|$$FIP
```

This is the result. Note that the Target Folder is been created if it does not exist.



Inside the "Sub" Folder we find the files, that have been unpacked from the Archive "`Document.zip`".



Important:

*© 2023 Theo Gottwald * https://www.smart-package.com*

If the Target-File does exist, the ZIP.-command **will try to delete any existing Target file**.
If this fails, the Target file will not be overwritten.
In case of such errors, the operation will continue without Error-Message.
Therefore its up to you to make sure that no (locked) target files exist before using the ZIP.-Command.

The "**cf**","**zf**" and the "**uv**" command are compatible with standard ZIP-Format.

Syntax

ZIP.uf|P2[|P3]

The "**cf**" and the "**uf**" command are compatible with standard ZIP-Format.

Parameter Explanation

P1 - "uf" or "unzip file"

P2 - Variable with the filename of the Source (Zip-)file that is to be unpacked.
If **P3** is missing, the Files will be extracted in the same path where **P2** is.:

P3 - (optional) Variable with the target file-path where the files from inside the Archive **P1**,
will be extracted. .

Example

```
'***********************************************
' Sample 1: ZIP.
'***********************************************
$$FIP=?path\Test\Sub
$$FID=?path\Test\Document.zip

' Unpack to ?path\
ZIP.uf|$$FID

' Unpack to $$FIP
ZIP.uf|$$FID|$$FIP

ENR.
```

Remarks

The "**cf**","**zf**" and the "**uf**" command are compatible with standard ZIP-Format.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

There is a size limitation that depends on the available 32-bit Memory in the WOW64.
Its somewhere between 1 GB and 2 GB.

See also:

- ZIP. - Handle Archives 4309
- IFZ. / NFZ. - If File in ZIP 4306
- cf - compress file 4313
- uf - unzip file/folder 4326
- uv - uncompress var 4329
- cv - compress var 4318
- sdk - set default compression 4323

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.9 uv - uncompress var

MiniRobotLanguage (MRL)

ZIP.uncompress var

Uncompress the content of a variable

Intention

The `ZIP.uv` - Command can be used to used to uncompress the content of a variable,
which has been compressed with the "**cv**" command before.
.
There are generally two ways of usage.

```
' Give Source- and Target variable
ZIP.uv|$$TXT|$$TAR
```

or

```
' ZIP the Content of the given variable
ZIP.uv|$$TXT
```

The "**cv**" and the "**uv**" command are not compatible with standard ZIP-Format.

Syntax

ZIP.uv|P2[|P3]

Parameter Explanation

P1 - "cv" or "compress var"

P2 - Variable with Content to be compressed. If P3 is missing, Result is placed in this variable.

P3 - Variable for the compressed result.

Example

```
'*********************************************
' Sample 1: ZIP.
'*********************************************
'
$$FIL=?path\readme.txt
CFF.$$FIL|$$TXT
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
FOR.§§LOP|0|10
  ZIP.sdc|§§LOP
  DBP.Compression is §§LOP
  CFF.$$FIL|$$TXT
  LEN.$$TXT|$$LEA
  DBP.Original=$$LEA
  $$TIA=#dtime#
  ZIP.cv|$$TXT|$$TAR
  $$TIB=#dsince#
  CAL.$$TIT=$$TIB|r
  LEN.$$TAR|$$LEB
  DBP.Compressed=$$LEB / Time: $$TIT sec.
  ZIP.uv|$$TAR
  LEN.$$TAR|$$LEC
  DBP.UnCompressed=$$LEC
NEX.
ENR.

'***********************************************
' Sample 2: ZIP.
'***********************************************
'
$$FIL=?path\readme.txt
CFF.$$FIL|$$TXT
LEN.$$TXT|$$LEA
DBP.Original=$$LEA
$$TIA=#dtime#
ZIP.cv|$$TXT|$$TAR
$$TIB=#dsince#
CAL.$$TIT=$$TIB|r
LEN.$$TAR|$$LEB
DBP.Compressed=$$LEB / Time: $$TIT
$$TIA=#dtime#
ZIP.cv|$$TXT
$$TIB=#dsince#
CAL.$$TIT=$$TIB|r
LEN.$$TXT|$$LEB
DBP.Compressed=$$LEB / Time: $$TIT

'***********************************************
' Sample 3: ZIP.
'***********************************************
'
$$FIL=?path\readme.txt
$$FIB=?path\readme_2.txt

: %LenCs 1
GMD.|§§§01|§§CSA
LEN.§§§01|§§LEA
DBP.Original=§§LEA * Checksum: §§CSA
END%

' Loading Textfile into Variable
CFF.$$FIL|$$TXT
%LenCs $$TXT

'Compressing COntent of Variable
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$TIA=#dtime#
ZIP.cv|$$TXT|$$TAR
$$TIB=#dsince#
CAL.$$TIT=$$TIB|r
%LenCs $$TAR

' Uncompressing content of variable
$$TIA=#dtime#
ZIP.uv|$$TAR
$$TIB=#dsince#
MBX.$$TAR|$$TAR
CAL.$$TIT=$$TIB|r
%LenCs $$TAR

' Textdatei unter anderem Namen herausschreiben
CTF.$$FIB|$$TAR
ENR.
```

Remarks

The "**cv**" and the "**uv**" command are not compatible with standard ZIP-Format.

Limitations:

There is a size limitation that depends on the available 32-bit Memory in the WOW64.
Its somewhere between 1 GB and 2 GB.

See also:

- ZIP. - Handle Archives 4309
- cv - compress var 4318
- sdk - set default compression 4323

© 2023 Theo Gottwald * https://www.smart-package.com

3.85.10 zf - Zip Folder

MiniRobotLanguage (MRL)

ZIP.Zip Folder

Compress the entire content of a folder

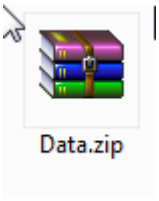
Intention

The `ZIP.zf` - Command can be used to used to compress the entire content of a folder and all Subfolders. Currently empty Folders and Subfolders are Ignored.
This command will create a Standard ZIP-File that is compatible with the ZIP-Format.

There are generally two ways of usage.

```
' Only give the Path of the folder to ZIP.
$$FIF=?path\Data\
ZIP.zf|$$FIF
```

In this case the Folder-Name will be taken to construct the Name of the ZIP-Archive. Also the Path of the Folder will the path where the Archive is been created. If the Archive does exist, files are added to the Archive. If the File is not a valid ZIP-File it will be deleted..



You can specify the name and path of the ZIP-Archive that is to be created.

```
$$FIF=?path\Data\
$$FIP=?path\Test\Sub\Hempel.zip
ZIP.zf|$$FIF|$$FIP
```

Or you can just specify the path where the ZIP-File is been created.

```
$$FIF=?path\MyData\
$$FIP=?path\
ZIP.zf|$$FIF|$$FIP
```

In this case the name of the Archive is been constructed from the name of the Source-Folder.

In case the Target Folder of the Archive that is to be created, does not exist, it will automatically be created.

If the Target-Zip File already exists, the Files are added to the existing ZIP-File.

*© 2023 Theo Gottwald * https://www.smart-package.com*

The "**cf**","**zf**" and the "**uv**" command are compatible with standard ZIP-Format.

Syntax

ZIP.zf|P2[|P3][|P4][|P5]

The "**zf**" and the "**uf**" command are compatible with standard ZIP-Format.

Parameter Explanation

P1 - "zf" or "Zip Folder"

P2 - Variable with the path of the folder that is to be compressed.

P3 - (optional) Variable with the target file-name and path.
If omitted the Target Zip-Name is constructed from the Source-Folder name,

P4 - (optional) Variable with a File-Pattern using Wildcards.
If omitted, "*** . ***" is used.

P5 - (optional) Flags, can be any of these:
s - also collect System-Files
z - do NOT recurse Subfolders
o - Sort Files Downwards
h - also collect hidden files
If omitted, Subfolders are recursed and collected. But none of the other options.

Example

```
'***********************************************
' Sample 1: ZIP.
'***********************************************
$$FIF=?path\MyData\
$$FIP=?path\
' Compress the MyData-Folder
' to a Zip-File "MyData.zip" in ?path\
ZIP.zf|$$FIF|$$FIP


$$FIF=?path\MyData\
$$FIP=?path\Test\MyZipData.zip
ZIP.zf|$$FIF|$$FIP

ENR.
```

Remarks

The "**cf**","**zf**" and the "**uf**" command are compatible with standard ZIP-Format.
If the Target-Zip File already exists, the Files are added to the existing ZIP-File.

© 2023 Theo Gottwald * https://www.smart-package.com

Limitations:

There is a size limitation that depends on the available 32-bit Memory in the WOW64.
Its somewhere between 1 GB and 2 GB.

See also:

- ZIP. - Handle Archives 4309
- IFZ. / NFZ. - If File in ZIP 4306
- uf - unzip file/folder 4326
- uv - uncompress var 4329
- cv - compress var 4318
- sdk - set default compression 4323
- 3.3 Specialfolders 345

© 2023 Theo Gottwald * https://www.smart-package.com

4. AddOn Components

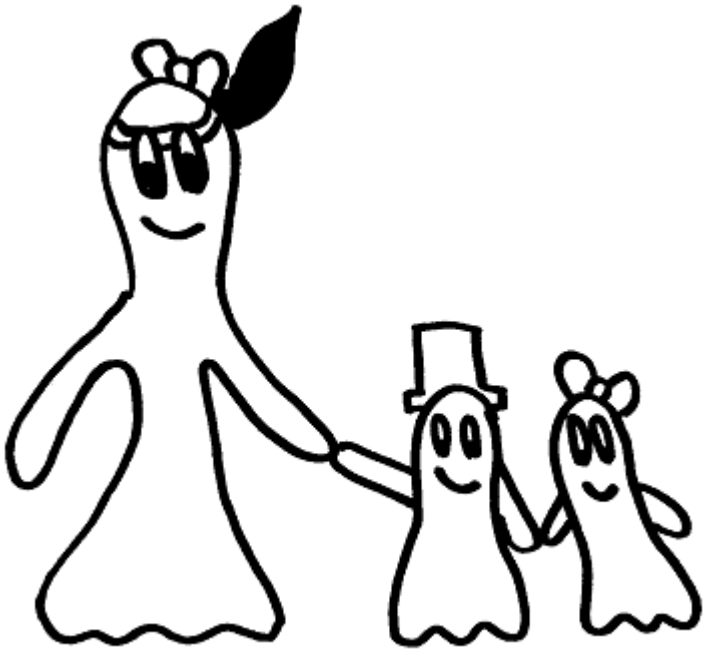
4 4. AddOn Components

4.86 MR Connect for use with SAP®

4.86.1 ! Smarty's Addon for use with SAP® (Connect to SAP®)

Smarty's Addon for use with SAP® (MCFS) ***Previous*** 4332 ***Top*** 41 ***Next*** 4342

Smarty's Addon's: MR Connect for use with SAP® (MCFS)



"MiniRobots Connect for use with SAP® " (In short: **MCFS**) is an AddOn which offers the possibility to connect the Smart Package Robot to SAP® GUI for Windows.

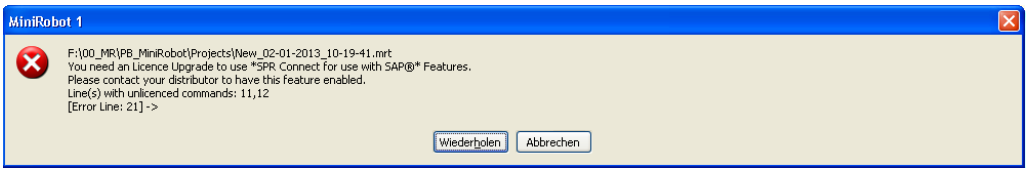
The communication between the SPR and the SAP® GUI for Windows is done via the SAP® GUI Scripting interface.

Therefore It is necessary to activate SAP® GUI Scripting, otherwise all commands from this AddOn would not work.

Important Hint:

This is an Third Party AddOn Component that is integrated into the SPR System as a AddOn.
It was developed by Stefan Schnell. Due to new developments from Stefan, the Add On will not need a separate licence any more.

*© 2023 Theo Gottwald * https://www.smart-package.com*



If a session is busy, it can't executes SAP® GUI Scripting. Each procedure inside "SPR Connect to SAP®" discovers the status of a session. If a session is busy, "Connect to SAP® " jumps over the command and will do nothing. In some cases you may get the timeout flag set.

Requirements

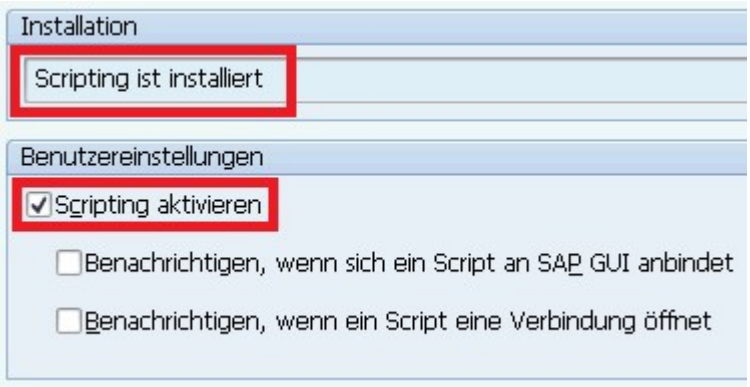
Operating system Windows 7 or higher.
Full standard installation of SAP GUI for Windows 7.30 or higher.

You need to enable GUI Scripting on Presentation Server, like this:

Choose the menu item Options... from the system menu of the SAP® Logon.

Choose the node Scripting.

© 2023 Theo Gottwald * https://www.smart-package.com



***Hint**: It is better to disable the notifications, otherwise you got a requester for each script execution.*

Sample-Script:

```
'_
Begin-------------------------------------------
------------------
'_
Variables---------------------------------------
-----------------
VAR.$$DEB
VAR.$$FAK
VAR.$$PRC
VAR.$$PID
VAR.$$VRC
VAR.$$SES
VAR.$$BUS
VAR.$$CMD
VAR.$$WND
VAR.$$CON
VAR.$$SEI

'_
Main--------------------------------------------
-------
'-Set DebugMode and FakeMode, only for
testing--------------
SSP.SetDebugMode|1
SSP.SetFakeMode|1
SSP.GetDebugMode|$$DEB
MBX.$$DEB
SSP.GetFakeMode|$$FAK
MBX.$$FAK

'-Shows about message
box---------------------------------
GSP.About
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
'-Pings an SAP
system---------------------------------------
GSP.Ping|000|BCUSER|minisap|DE|NSP|192.168.145.130|
0|$$PRC
IVV.$$PRC=1
MBX.Ping successful
ELS.
MBX.Ping not successful
EIF.

'-Checks if SAP GUI for Windows is
installed----------------
ISP.SAPGUIInstalled
'-Checks if SAP Logon is loaded and start it if
necessary---
NSP.SAPGUI Exists
GSP.SAPGUILaunch
$$PID=#pid#
MBX.PID of SAP Logon is $$PID
EIF.

'-Gets the SAP GUI Scripting
version------------------------
GSP.ScriptingVersion|$$VRC
MBX.SAP GUI Scripting Version $$VRC
'-Checks if a connection
exists-----------------------------
NSP.Connection Exists
MBX.Please open an SAP connection
EIF.

'-Gets the first free session of connection
0---------------
GSP.FirstFreeSession|0|$$SES
MBX.First free session is $$SES
CAL.$$SES=$$SES - 1

'-Checks if session is
busy---------------------------------
GSP.SessionBusy|0|$$SES|$$BUS
IVV.$$BUS=1
MBX.Session $$SES of connection 0 is busy
ELS.
MBX.Session $$SES of connection 0 is not busy
EIF.

'-Executes a small
script-------------------------------------
$$CMD=session.findById("wnd[0]/usr/txtRSYST-
MANDT").text = "000" $crlf$
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
$$CMD=$$CMDsession.findById("wnd[0]/usr/txtRSYST-
BNAME").text = "BCUSER" $crlf$
$$CMD=$$CMDsession.findById("wnd[0]/usr/pwdRSYST-
BCODE").text = "minisap" $crlf$
$$CMD=$$CMDsession.findById("wnd[0]/usr/txtRSYST-
LANGU").text = "DE" $crlf$
$$CMD=$$CMDsession.findById("wnd[0]/tbar[0]/btn[0]"
).press $crlf$
GSP.ExecSub|0|$$SES|$$CMD

'-Gets window which uses transaction
SE16------------------
GSP.WhichWindowHas|Transaction|SE16|$$WND
MBX.Handle of the window is $$WND

'-Gets the first connection from user
BCUSER----------------
GSP.WhichConnectionHas|User|BCUSER|$$WND
MBX.Handle of the window is $$WND

'-Gets the first session which runs report
SAPLWB_INITIAL_TOOL--
GSP.WhichSessionHas|Program|
SAPLWB_INITIAL_TOOL|$$WND
MBX.Handle of the window is $$WND

'-Gets the connection which contains the window
handle 4711-
GSP.WhichConnectionhWnd|4711|$$CON
MBX.Connection is $$CON

'-Gets the session with the window handle
4711-------------
GSP.WhichSessionhWnd|4711|$$SEI
MBX.Session is $$SEI
ELS.
MBX.SAP GUI for Windows is not installed
EIF.

'_
End-----------------------------------------------
-------
END.
```

Trademark/legal notice

SAP® and SAP® GUI Scripting are registered trademarks of SAP AG
Smart Package Robot and MiniRobot are properties of Theo Gottwald

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

4.86.2 GSP. - Get SAP® Data

GSP. - Get Data from SAP®-Application Previous 4336 Top 41 Next 4349

MiniRobotLanguage (MRL)

GSP. Command

Get Data from SAP®-Application

Intention

The `GSP.`-Command is the command to retrieve information from the SAP®-System via SPR-Skript. It has multiple options that makes automation easier then ever before.

Usage is simple:

```
GSP.(subcommand)|(one or more parameters)|(Variable with
result of operation)
```

Syntax

GSP.subcommand[|P1][|P2][|P3]

Parameter Explanation

These general Rules apply:

1. If you omit the Variable for the Result of the command, the result will be stored on the stack.

2. Session numbers are counted from 1 to 6 as usual.

3. If you omit the connection number, a 1 is used as default number.

4. If there was a problem a session number of zero may be returned and the timeout flag will be set. Otherwise the timeout flag will be cleared.

5. All Subcommands have a long name that is better readable, and a short abbreviation that saves you typing. You can choose which one you prefer.

The following subcommands are currently supported:

*© 2023 Theo Gottwald * https://www.smart-package.com*

"About","a."
Shows a messagebox with information about "Connect to SAP® ".
This command needs and accepts no parameter.
Example:

```
GSP.about
```

"FirstFreeSession","ffs."
Delivers the first session number of the given connection which is not busy.
Example:

```
' $$CON - Number of SAP® connection
' $$RES - Result of Operation
'
GSP.ffs.|$$CON|$$RES

GSP.FirstFreeSession|5|$$SES
MBX.First free session of connection 5 is $$SES
```

"ExecScript","ExecSub","exs."
Executes SAP® GUI Scripting code without a return value. Example:

```
' $$CON = Number of SAP® connection
' $$SES = Number of SAP® session
' $$CMD = SAP® GUI Script code
'
GSP.exs.|$$CON|$$SES|$$CMD

$$CMD=session.findById("wnd[0]/usr/txtRSYST-MANDT").text =
"000"
$$CMD=$$CMD$crlf$
$$CMD=$$CMDsession.findById("wnd[0]/usr/txtRSYST-BNAME").text =
$$CMD=$$CMD"BCUSER" $crlf$
$$CMD=$$CMDsession.findById("wnd[0]/usr/pwdRSYST-BCODE").text =
$$CMD=$$CMD "minisap" $crlf$
$$CMD=$$CMDsession.findById("wnd[0]/usr/txtRSYST-LANGU").text =
"DE"
$$CMD=$$CMD$crlf$
$$CMD=$$CMDsession.findById("wnd[0]/tbar[0]/btn[0]").press
$crlf$

GSP.ExecSub|0|0|$$CMD
```

"Ping","p."
Sends a ping to a SAP® system. You need to specify these arguments:
If the ping is successful the return value is 1, otherwise 0.
If an error occurs the return value is -1.
Example:

```
' $$SAC - SAPClient = Number OF the CLIENT (Mandant)
' $$USN - SAPUser = NAME OF the USER
' $$SAW - SAPPassWord = Password OF the USER
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' $$SAL - SAPLanguage = Language, e.g. DE OR EN.
' $$SID - SAPSystem = System ID (SID)
' $$SAH - SAPHostName = NAME OF the HOST (Application SERVER)
' $$SSN - SAPSystemNumber = Number OF the system (INSTANCE
number)
' $$RES - Variable for Result
'
GSP.Ping|$$SAC|$$USN|$$SAW|$$SAL|$$SOD|$$SAH|$$SSN|$$RES

GSP.Ping|000|BCUSER|minisap|DE|NSP|192.168.145.130|0|$$PRC
IVV.$$PRC=1
      MBX.Ping successful
ELS.
      MBX.Ping not successful
EIF.
```

"SapGuiLaunch","sl."

Starts the program saplogon.exe or saplgpad.exe, which are the base of the SAP® GUI for Windows®. If the start is successful the return value is the process ID (PID) of the program, otherwise 0. `#PID#` is set by this command to the PID or if an error happened to zero..
Example:

```
GSP.sl.
```

"ScriptingVersion","sv."

Delivers the SAP® GUI Scripting version. The SAP® GUI Scripting version as string, e.g. 7300.249. The first number is the major and the second number is the minor version.
Example:

```
' Version number will be on TOS
GSP.sv.
' Version number will be in Variable
GSP.sv|$$RES
```

"SessionBusy","sb."

Checks if the session is busy. If the specified SAP® session is busy the return value is 1, otherwise 0. Example:

```
' $$CON = Number of the SAP® connection
' $$SES = Number of the SAP® session
' $$RES - Variable for Result
'
GSP.sb.|$$CON|$$SES|$$RES
```

"WhichConnectionHas","wch."

Delivers the connection number of first connection with the given property.Number of the connection, otherwise -1. If the session is busy, this function returns -1.
Example:

```
'
' $$PRO = Property
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
' $$VAL = Value of Property
' $$RES - Variable for Result
'
GSP.wch.|$$PRO|$$VAL|$$RES
```

"WindowToConnection","wtc.","WhichConnectionHwnd"

Delivers the connection number of the given window handle, or in case of error -1.
Example:

```
'
' $$HWN = Handle of a Window
' $$RES - Variable for Result
'
GSP.wtc.|$$PRO|$$RES
```

"WhichSessionHas","wsh."

Delivers the session number of first session (1-6) with the given connection and property, otherwise 0. Example:

```
'
' $$PRO = Property
' $$VAL = Value of Property
' $$RES - Variable for Result
'
GSP.wtc.|$$PRO|$$VAL|$$RES
```

"WindowToSession","wts.","WhichSessionHwnd"

Delivers the session number of the given window handle, or in case of error -1.
If you omit $$HWN the actual window is used.
If the command can not evaluate a valid window handle (no localized window and no handle given
Example:

```
' $$VAL = Value of Property
' $$RES - Variable for Result
'
GSP.wts.|$$HWN|$$RES
' If you omit $$HWN the actual window is used.
GSP.wts.||$$RES
' If you omit $$RES the TOS will contain the result
GSP.wts.
```

"WhichWindowHas","wwh."

Delivers the handle of first session window with the given property.Handle of the session window, otherwise 0. Example:

```
'
' $$PRO = Property
' $$VAL = Value of Property
' $$RES - Variable for Result
'
GSP.wwh.|$$PRO|$$VAL|$$RES

GSP.WhichConnectionHas|User|BCUSER|$$WND
MBX.Handle of the window is $$WND
```


© 2023 Theo Gottwald * https://www.smart-package.com

"DebugMode","dm."
Delivers the actual debugging state of the MRCS-AddOn.
For internal use only. Example:

```
GSP.dm.|$$RET
```

"FakeMode","fm."
Delivers the actual FakeMode state of the MRCS-AddOn.
For internal use only. Example:

```
GSP.fm.|$$RET
```

P1..Px - (optional) depending on the subcommand, there may be no or multiple parameters needed.

Examples

```
GSP.WhichConnectionhWnd|4711|$$CON
MBX.Connection is $$CON
END.

$$WND = 4711
GSP.WhichConnectionhWnd|$$WND|$$CON
GSP.WhichSessionhWnd|$$WND|$$SES
MBX.Window with handle $$WND is connection $$CON and session
$$SES
END.

GSP.WhichSessionHas|Program|SAPLWB_INITIAL_TOOL|$$WND
MBX.Handle of the window is $$WND
END.

GSP.WhichSessionhWnd|4711|$$SES
MBX.Session is $$SES
END.

$$WND = 4711
GSP.WhichConnectionhWnd|$$WND|$$CON
GSP.WhichSessionhWnd|$$WND|$$SES
MBX.Window with handle $$WND is connection $$CON and session
$$SES
END.

GSP.WhichWindowHas|Transaction|SE16|$$WND
MBX.Handle of the window is $$WND
END.

GSP.SessionBusy|0|0|$$BUS
```

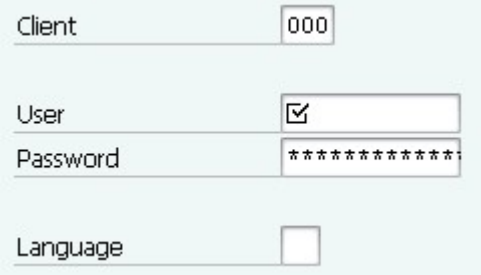

© 2023 Theo Gottwald * https://www.smart-package.com

```
IVV.$$BUS=1
    MBX.Session 0 of connection 0 is busy
ELS.
    MBX.Session 0 of connection 0 is not busy
EIF.
END.
```

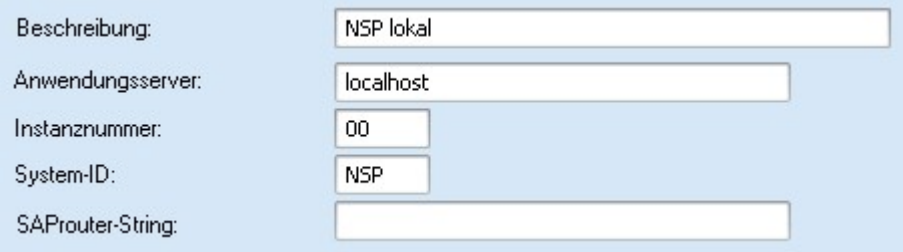
Remarks

Some of these Sub-commands will set the timeout-flag if the operation was not successful.

You get the arguments for the PING Subcommand as follows:



Also you get the information from the property item from the system of the SAP logon program.



Important hint:
Ping uses the ANSI version of the SAP® GUI ActiveX Controls.

Limitations:

If a session is busy, it can't executes SAP® GUI Scripting. Each procedure inside MCFS discovers the status of a session. If a session is busy, "Connect to SAP® " jumps over the command and will does nothing. In some cases you may get the timeout flag set.

See also:

- ! Smarty's Addon for use with SAP® (Connect to SAP®) 4336

© 2023 Theo Gottwald * https://www.smart-package.com

- ISP. / NSP. - If SAP® State 4349
- WSP. - Wait for SAP® State 4342
- SSP. - Set SAP® Data 4352

© 2023 Theo Gottwald * https://www.smart-package.com

4.86.3 ISP. / NSP. - If SAP® State

MiniRobotLanguage (MRL)

ISP. Conditional Statement

If SAP®- State

Intention

Conditional Statement.

Use this statement to test for several states around the SAP®- System.

```
ISP.connection exists
ISP.sapgui exists
ISP.SAPGUIInstalled
ISP.version>700.32
ISP.version=700.32
ISP.version!700.32
ISP.version=700.32
```

`NSP.` - is the negative form of this statement.

Syntax

ISP.Subcommand[|P1...|Px]

Parameter Explanation

These general Rules apply:

1. If you omit the Variable for the Result of the command, the result will be stored on the stack.

2. Session numbers are counted from 1 to 6 as usual.

3. If you omit the connection number, a 1 is used as default number.

4. If there was a problem a session number of zero may be returned and the timeout flag will be set. Otherwise the timeout flag will be cleared.

5. All Subcommands have a long name that is better readable, and a short abbreviation that saves you typing. You can choose which one you prefer.

The following subcommands are currently supported:

*© 2023 Theo Gottwald * https://www.smart-package.com*

"Connection","Connection Exists","ce."

Checks if a connection to an SAP® system exists.
Example:

```
ISP.ce.
      DBP.Connection exists!
ELS.
      DBP.No Connection exists!
EIF.

NSP.Connection Exists
MBX.Please open an SAP connection
EIF.
```

"Sapgui","Sapgui Exists","se."

Checks if SAP® Logon is loaded.Example:

```
NSP.SAPGUI Exists
      GSP.SAPGUILaunch
      $$PID=#pid#
      MBX.PID of SAP Logon is $$PID
ELS.
     DBP.SAPGUI is not loaded!
EIF.
```

"SapguiInstalled","si."

Checks if SAP® GUI for Windows® is installed.
Example:

```
ISP.SAPGUIInstalled
      MBX.SAP GUI is installed
ELS.
      MBX.SAP GUI is not installed
EIF.
```

"Version","Ve."

Test the SAP® GUI Scripting version against a given value. You can use the Operators:
= - Test for Equality (Version=)
! - Test for NOT Equal (Version NOT Equal to)
> - Test for Version higher then
< - Test for Version lower then

Example:

```
ISP.ve.>7300.248
     DBP.SAPGUI is loaded!
ELS.
     DBP.SAPGUI is not loaded!
EIF.
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'***********************************
' ISP.-Sample
'***********************************
'
ISP.
'...
ELS.
'...
EIF.
ENR.
```

Remarks

-

Limitations:

If a session is busy, it can't executes SAP® GUI Scripting. Each procedure inside **MCFS** discovers the status of a session. If a session is busy, "Connect to SAP® " jumps over the command and will does nothing. In some cases you may get the timeout flag set.

See also:

- ! Smarty's Addon for use with SAP® (Connect to SAP®) 4336
- GSP. - Get SAP® Data 4342
- WSP. - Wait for SAP® State 4342
- SSP. - Set SAP® Data 4352

© 2023 Theo Gottwald * https://www.smart-package.com

4.86.4 SSP. - Set SAP® Data

MiniRobotLanguage (MRL)

SSP. Command

Set Data from SAP®-Application

Intention

The `SSP.`-Command is the command to set or change settings in the MCFS via SPR-Skript. Its currently only for internal use.

Usage is simple:

```
SSP.(subcommand)|(one or more parameters)|(Variable with
result of operation)
```

Syntax

SSP.subcommand[|P1][|P2][|P3]

Parameter Explanation

These general Rules apply:

1. If you omit the Variable for the Result of the command, the result will be stored on the stack.

2. Session numbers are counted from 1 to 6 as usual.

3. If you omit the connection number, a 1 is used as default number.

4. If there was a problem a session number of zero may be returned and the timeout flag will be set. Otherwise the timeout flag will be cleared.

5. All Subcommands have a long name that is better readable, and a short abbreviation that saves you typing. You can choose which one you prefer.

The following subcommands are currently supported:

"DebugMode","dm."
Set the actual debugging state of the MCFS-AddOn.
For internal use only. Example:

*© 2023 Theo Gottwald * https://www.smart-package.com*

```
SSP.dm.|$$MOD
```

"FakeMode","fm."

Set the actual FakeMode state of the MCFS-AddOn.
For internal use only. Example:

```
SSP.fm.|$$MOD
```

P1..Px - (optional) depending on the subcommand, there may be no or multiple parameters needed.

Example

-

Remarks

-

Limitations:

If a session is busy, it can't executes SAP® GUI Scripting. Each procedure inside MCFS discovers the status of a session. If a session is busy, "Connect to SAP® " jumps over the command and will does nothing. In some cases you may get the timeout flag set.

See also:

- ! Smarty's Addon for use with SAP® (Connect to SAP®) 4336
- ISP. / NSP. - If SAP® State 4349
- WSP. - Wait for SAP® State 4342
- GSP. - Get SAP® Data 4342

© 2023 Theo Gottwald * https://www.smart-package.com

4.86.5 WSP. - Wait for SAP® State

MiniRobotLanguage (MRL)

WSP. Command

Wait For SAP® Event

Intention

This command is useful to have your script wait until certain conditions are met or not met.

Examples:

```
WSP.![Max. Wait]|connection exists
WSP.[Max. Wait]|sapgui exists
WSP.[Max. Wait]|version>700.32
WSP.[Max. Wait]|version=700.32
WSP.[Max. Wait]|version!700.32
```

Syntax

WSP.[Timeout in s]|Subcommand[P1]

Parameter Explanation

These general Rules apply:

1. If you omit the Variable for the Result of the command, the result will be stored on the stack.

2. Session numbers are counted from 1 to 6 as usual.

3. If you omit the connection number, a 1 is used as default number.

4. If there was a problem a session number of zero may be returned and the timeout flag will be set. Otherwise the timeout flag will be cleared.

5. All Subcommands have a long name that is better readable, and a short abbreviation that saves you typing. You can choose which one you prefer.

6. If the state does not change within a given time, the timeout-flag will be set.

*© 2023 Theo Gottwald * https://www.smart-package.com*

7. If you prefix the time-value with a "**!**" then the condition value is negated and we are waiting for just the opposite condition.

The following subcommands are currently supported:

"Connection","Connection Exists","ce."

Waits for a connection to an SAP® system to exist.
Example:

```
' Wait endless time
WSP.|ce.
' Wait 25 seconds
WSP.25|ce.
' Wait that connection breaks within 25 seconds
WSP.!25|ce.
```

"Sapgui","Sapgui Exists","se."

Waits for SAP® Logon to be loaded/unloaded.Example:

```
WSP.|se.
' Wait 25 seconds
WSP.25|se.
' Wait that it is unloaded within 25 seconds
WSP.!25|se.
```

"SapGuiInstalled","si."

Checks if SAP® GUI for Windows® is installed.
Example:

```
WSP.|si.
' Wait 25 seconds
WSP.25|si.
' Wait that it is uninstalled within 23 seconds
WSP.!(20+5)|si.
```

"Version","ve."

Wait for the SAP® GUI Scripting to have a specified version compared against a given value. You can use the Operators:
= - Test for Equality (Version=)
! - Test for NOT Equal (Version NOT Equal to)
> - Test for Version higher then
< - Test for Version lower then

Example:

```
WSP.ve.>7300.248
```


© 2023 Theo Gottwald * https://www.smart-package.com

Example

```
'*****************************************************
 WSP.!34|connection exists
 WSP.14|sapgui exists
 WSP.$$TIM|version>700.32
 WSP.$$SEC|version=700.32
 WSP.(($$AUT+4)*$$FAK)|version!(700.32+$$NUM)
```

Remarks

-

Limitations:

If a session is busy, it can't executes SAP® GUI Scripting. Each procedure inside **MCFS** discovers the status of a session. If a session is busy, "Connect to SAP® " jumps over the command and will does nothing. In some cases you may get the timeout flag set.

See also:

- ! Smarty's Addon for use with SAP® (Connect to SAP®) 4336
- GSP. - Get SAP® Data 4342
- ISP. / NSP. - If SAP® State 4349
- SSP. - Set SAP® Data 4352

© 2023 Theo Gottwald * https://www.smart-package.com

4.87 InstaRec - the Recorder

4.87.1 4.9.1 InstaRec - User-Templates

Customizing the InstaRec Outcome **Previous** 39 **Top** 41 **Next** 73

Technical Notes on "InstaRec"

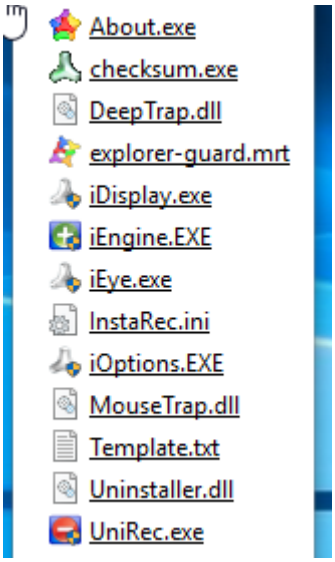
What you should know and don't see from outside

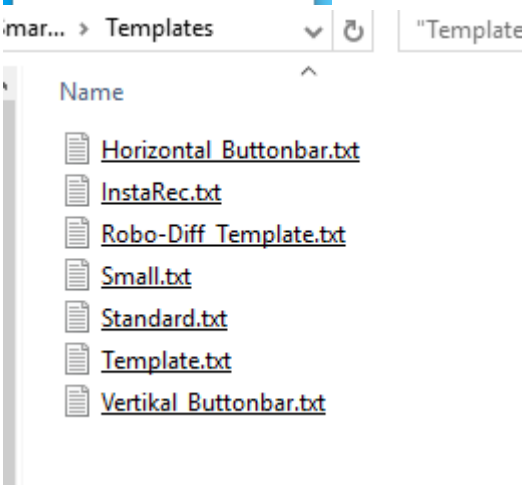
InstaRec was developed with the idea of "Just do your Installation,
InstaRec will do your Script".
We believe that its currently the best Scripting Recorder on the market.

This Chapter describes the more advanced topics:

Customizing your Scripts using "User-Templates".

*© 2023 Theo Gottwald * https://www.smart-package.com*





1. InstaRec-Installation folder "InstaRec.txt" (Template) with user-Template-File. (Contents may change)

2. General Template-folder with

InstaRec uses Templates

Using "Templates, you can easily customize the outcome of your Scripting-Recorder.
Generally the Template-Format is the same for "InstaRec" and for "SPR-Editor".

When creating a Scriptfile, InstaRec first tries to load a "Template-File".
The name of the Template-File is whether "`InstaRec.txt`" or "`Template.txt`", depending on where it is found.
The format of the file is always the same.

The path of this Template-file can be:
1. In the InstaRec-Installation folder for a file called "`Template.txt`"
2. In the "Templates-Folder" that is in the Main-Installation folder for a file called "`InstaRec.txt`"
3. If no Template is found, an internal predefined Template is used.

InstaRec will check these locations and take the first File that is found.

*© 2023 Theo Gottwald * https://www.smart-package.com*

The Template-file is the base for the later script. A Template may look like the one below.

Please note the first two lines. They are necessary!
If you omit these two lines, the first two lines from your template will be missing.

```
Default Template for InstaRec
.mrt
'SPR Script-file: <filename>
'Original Filename: <Fullfilename>
'Version: 1.00, saved at: [<path>]
'Purpose:
'Author: <user>
'Creation date: <date>,<time>
'============================================================
'#EXE:?pfad\
'#SPI:ForceWrite
'
'
```

InstaRec will delete the lower part of the Script, and insert its own generated Script-code instead.
Therefore you can omit the last lines that are normally in any Template.
In this example we have omitted these last lines already.

You can use the following Meta-Tags in these Template-Files:

<filename> -> "Myfile.mrt" or "Install_LuxRenderSetup.rem"
Note: ".rem" is an internal filetype which is later converted to ".mrt"

<Fullfilename> -> "C:\Users\UserA\Desktop\Projekt_04092016_065037\Install_Lux

<path> -> "C:\Users\UserA\Desktop\Projekt_04092016_065037\"

<user> -> "UserA" (Username)

<time> -> "04-09-2016" (mm-dd-yyyy)

<date> -> "06:50:54" (hh:mm:ss)

These Meta-Tags are replaced at Script creation time with the real filenames and pathes.
If you change the Template, the final Script will also change.

© 2023 Theo Gottwald * https://www.smart-package.com

4.87.2 4.9.2 InstaRec - AutoPatchSkript.exe

Using your own "PlugIns" with InstaRec Previous 39 Top 41 Next 73

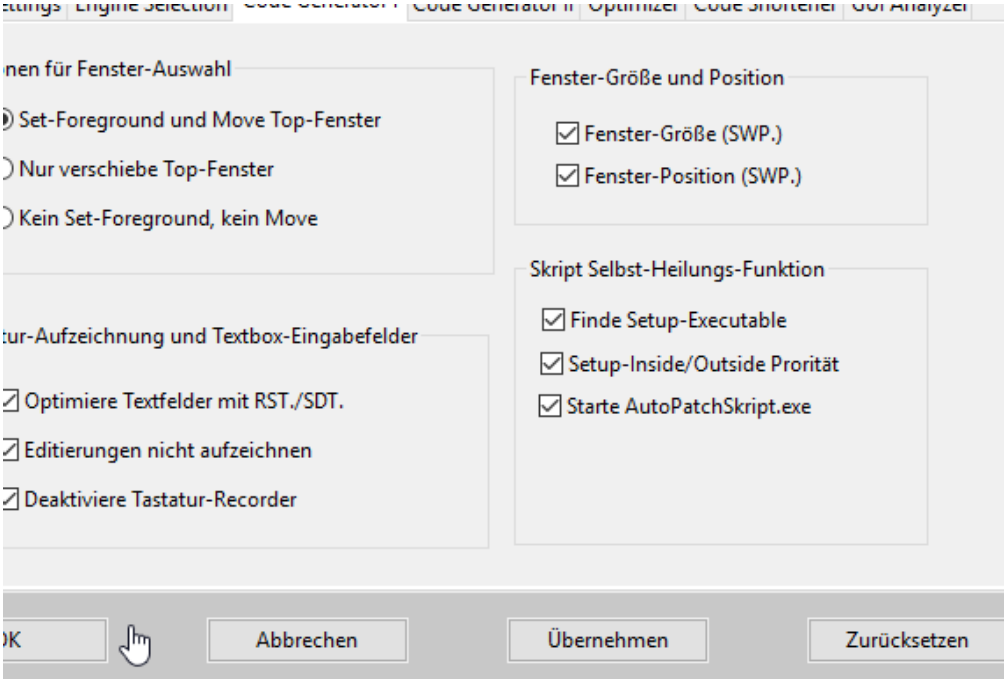
Using the "AutoPatchSkript.exe"

You can customize your final Skript before it is loaded into the editor.

InstaRec was developed with the idea of "Just do your Installation, InstaRec will do your Script".
We believe that its currently the best Scripting Recorder on the market.

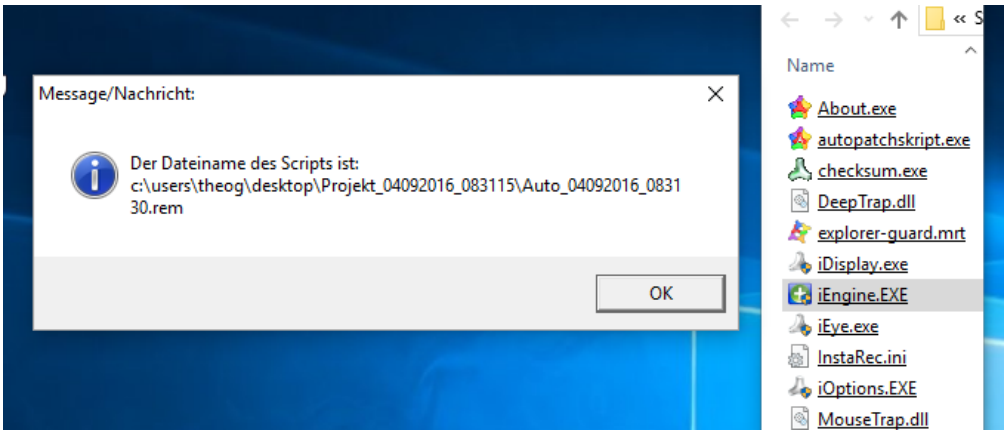
This Chapter describes the more advanced topics:

Customizing your Scripts using "AutoPatchSkript.exe".



1. iOptions Dialog with the "AutopatchSkript.exe" Option.
You need to check this to use the "AutoPatchSkript.exe".
(Contents may change)

*© 2023 Theo Gottwald * https://www.smart-package.com*



2. InstaRec Folder with the "AutopatchSkript.exe" File.
There is an "AutoPatchSkript.mrt" in the InstaRec-Folder as a Sample.
It can be compiled to the "AutoPatchSkript.exe" Executable.
Its just an starting point for your own PlugIns.
(Contents and Design may change in future releases)

InstaRec can use your own "PlugIn" to manipulate the Script after it was recorded.

Currently the way of the Skript is this:

1. You record the Script.
2. InstaRec loads the Template File
3. InstaRec inserts the Script into the Template
4. The Sciptfile is saved on Disk
5. If an "`AutoPatchSkript.exe`" file is available in the InstaRec-Installation Folder, InstaRec will start it with the Skript-Filename as Parameter
   and wait until the Executable has ended.
6. If available, your "AutoPatchSkript.exe" can now manipulate the Skriptfile. It must end however, because InstaRec will wait until you are ready.
7. Now InstaRec will start the ".rem" to ".mrt" Converter this will generate the final Script-File.
8. Next the SPR-Editor will load the Script-File.

*© 2023 Theo Gottwald * https://www.smart-package.com*

4.88 GPX-Korrektor

What is the GPX-Korrektor?

Did you know that you can add your hometown or other regions of the world to "Google Street View"?
Even if they are already in Street View, you can update the Data with newer versions.

Of course you will need a 360 Degree Camera like these:
Necessary 360 Degree Cameras.

https://amzn.to/3X59Uko
or this one
https://amzn.to/3lePl88
or this one:
https://amzn.to/3juGRcg

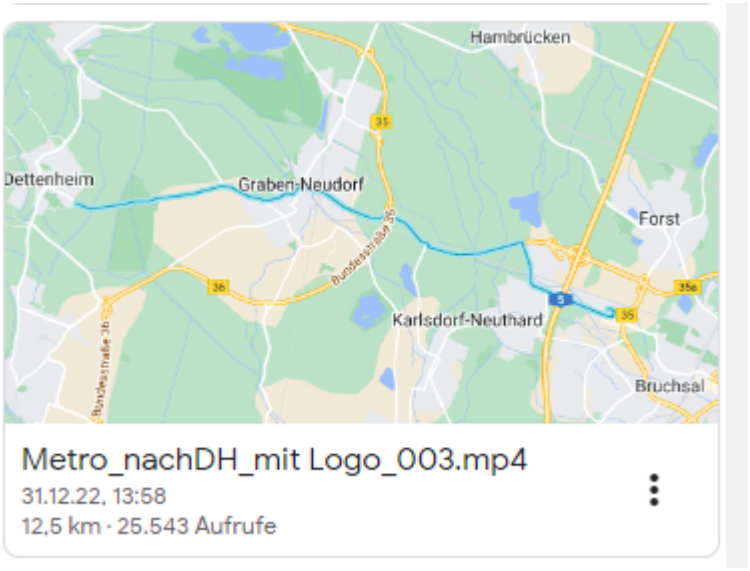
Next you will make a walk through the street or place that you want to add to/Update in Google Street View.
And you will record a 360 Degree Video, together with the necessary GPS-Data.
The GPS-Data will be stored in GPX-File.

For the first 2 Cameras it may be necessary to extract the GPS-Data using this free tool:
https://drive.google.com/drive/folders/1U7youYqqE4JTriF37Ac52h29tQGxM2p7
out of the Camera Video files.

Once you have the GPX-Files and you may have some ".mp4" Files that you have recorded.
Then you can go to:
https://streetviewstudio.maps.google.com/
and upload these files.

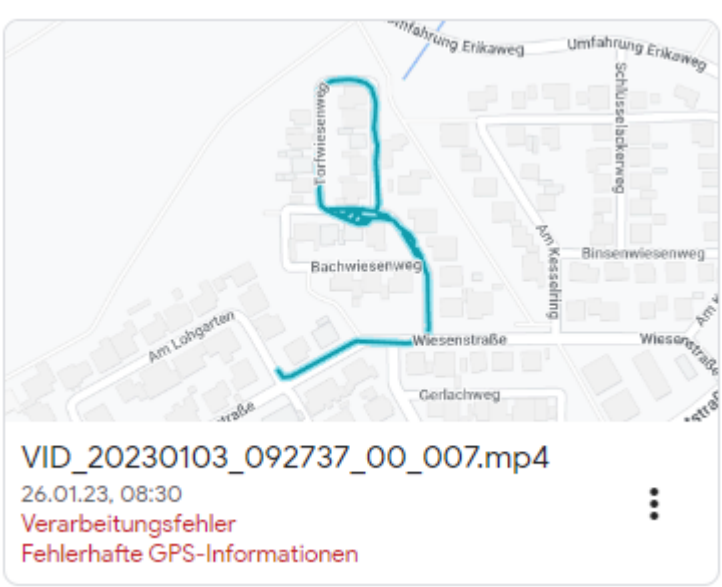
After some time (some days to 2 weeks) your Uploads will be visible in Google Street View.
You will also see, how many people have seen your Uploads.

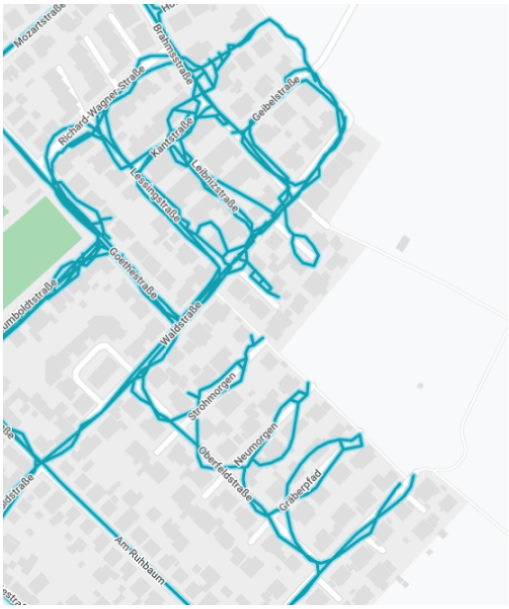
© 2023 Theo Gottwald * https://www.smart-package.com



Now in some cases, the GPS-Coordinates have not been good enough to add the material to Street View.
In such case, Google may "throw an error message" like this one. Or it may accept the file which may result in (Picture 2).

*© 2023 Theo Gottwald * https://www.smart-package.com*





Picture 1: Google is not satisfied with the GPS-Coordinates in the GPX-File.
Picture 2: Google took the unmodified Coordinates from the Camera/Smartphone and the result is not perfect.

This will show up in the Street-View, as the "Marks on the Floor" will be on the wrong places" and not "In a Line".

*© 2023 Theo Gottwald * https://www.smart-package.com*

Picture 3: The "Marks on the road" may be at the wrong place, if the uploaded Coordinates are not on the road.

This is where you can use the SPR or the GPX-Korrektor.

Also the SPR has now Scripting capabilities to work with XML-Files and Timestamps like "00:00:000", so if you can automate the process where possible.
If you can let an SPR-Script do "some corrections". This will be mostly the case if there are only corrections about the time-stamps.

In most cases the GPX-Extraktor is the tool of choice here, as you can move any Trackpoint
and you can also use the automatic corrections with the "Check and Repair" Button.

The GPX-Korrektor is an Add-On that can be used to interactively manipulate geograhical Coordiantes in a GPX-File.
The Program is optimized for usage with Google-Street View compatible Uploads.

So this program can help you to correct the GPS-Coordinates in a GPX-File before uploading the file to **Street View**.

The Program will need a BING-Maps Key to work. 4368

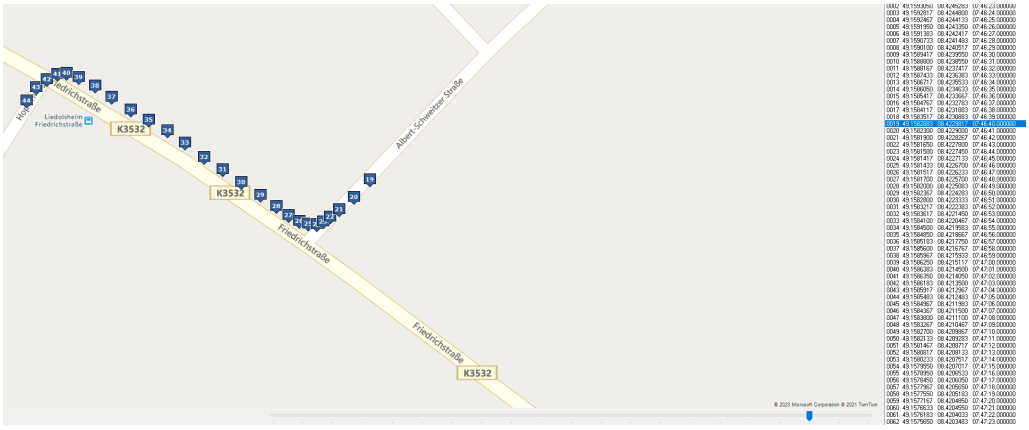
The Main-Window shows on the right side a list with numbered GPS-Coordinate.
The Selected Coordinate is the starting Coordinate unless other Settings.
When you start the program only 1 Coordinate is display and also only if a Point is selected.
You need to select on or more points to update the Map-Display.

*© 2023 Theo Gottwald * https://www.smart-package.com*

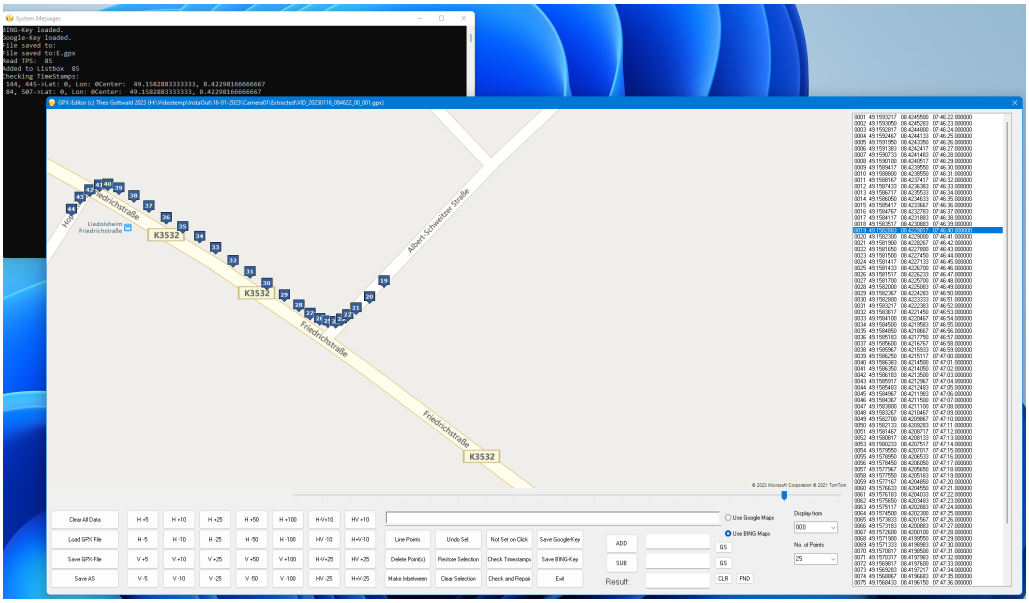
Even after changing Settings, it may be necessary to first select a Point to have the display updated.

You see on the Maps each GPX-Point as as Landmark with its number.

You have a **ZOOM-Switcher Slider** which you can use to **change the ZOOM** of the Map.



The following Picture shows the GPX-Korrektor Program with its 2 windows (Black Output Window, and Main Window).
Please do NOT Close the Output Window. Otherwise the Main Window will also be closed and all Data will be lost!



Here is my hometown "Dettenheim".
As you can see it is nearly completely added to Street View using the SPR and also did some Corrections, using the GPX Korrektor Utility.
In this process the Program was developed.

*© 2023 Theo Gottwald * https://www.smart-package.com*

© 2023 Theo Gottwald * https://www.smart-package.com

4.88.1 Eingabe des BING bzw. Goggle-API Keys

The Program will need a BING-Maps Key to work.

Here are the steps to get a Bing Maps key:

1. Create a Bing Maps account:
   Go to the Bing Maps website (https://www.bingmapsportal.com/) and sign up for a Bing Maps account.

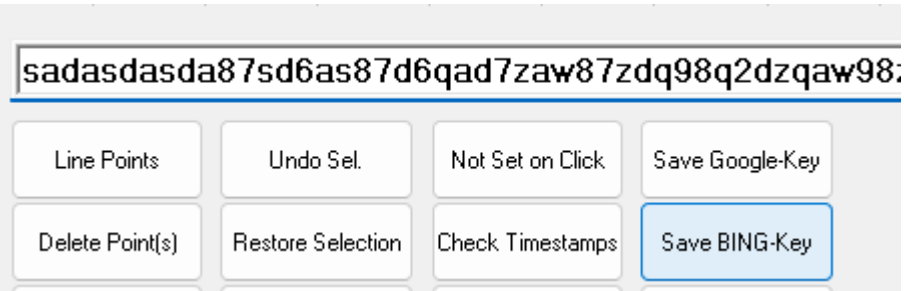
2. Get an API license:
   In the Bing Maps portal, go to the "**My Account**" section and select "**My keys**."
   Then click on "**Create a key**" to get a Bing Maps API license.

3. Choose the API:
   Select the Bing Maps Imagery API, which provides access to Bing Maps satellite and aerial imagery.

4. Add the Key to the Korrektor:
   Copy the API key generated in the portal into the large white Textbox and press "**Save BING Key**".

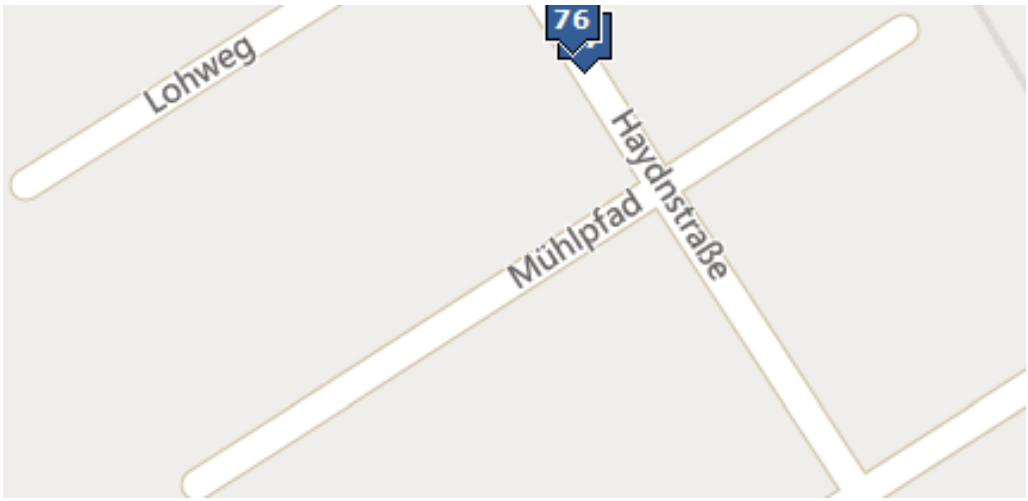
This is only done once, as the Key is been stored in a encrypted way in the Computer.

Note: You can also add an Google-Key but that is not of much use in the current version.



If you have added the Google-Maps API, you can check if the Map Information is similar to Google Maps or not - like in this case.
Currently the program will not display any Trackpoints on the Google-Maps.

© 2023 Theo Gottwald * https://www.smart-package.com



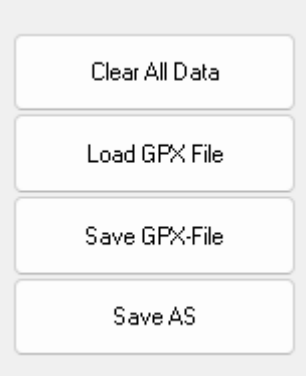
BING-Maps, Stand 01/2023

Google-Maps Stand 01/2023

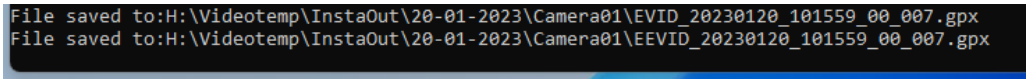
*© 2023 Theo Gottwald * https://www.smart-package.com*

4.88.2 FIle Load/Save-Buttons

These Buttons can be used to Load and Save the GPX-File.



After each File-Save, the File-Path and Filename is shown in the Console Window.



© 2023 Theo Gottwald * https://www.smart-package.com

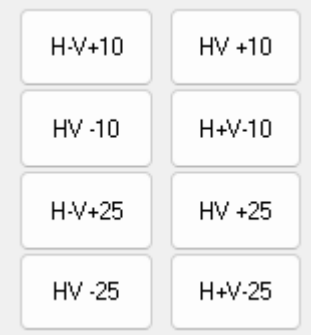
4.88.3 H-/V+- Buttons

These Buttons are used to MOVE the selected Trackpoint(s).

There are two sections:

1. The first Section will move the selected Trackpoint(s) only Left/Right or Up/Down:

2. The Second Section can move the selected TrackPoints Diagonal:



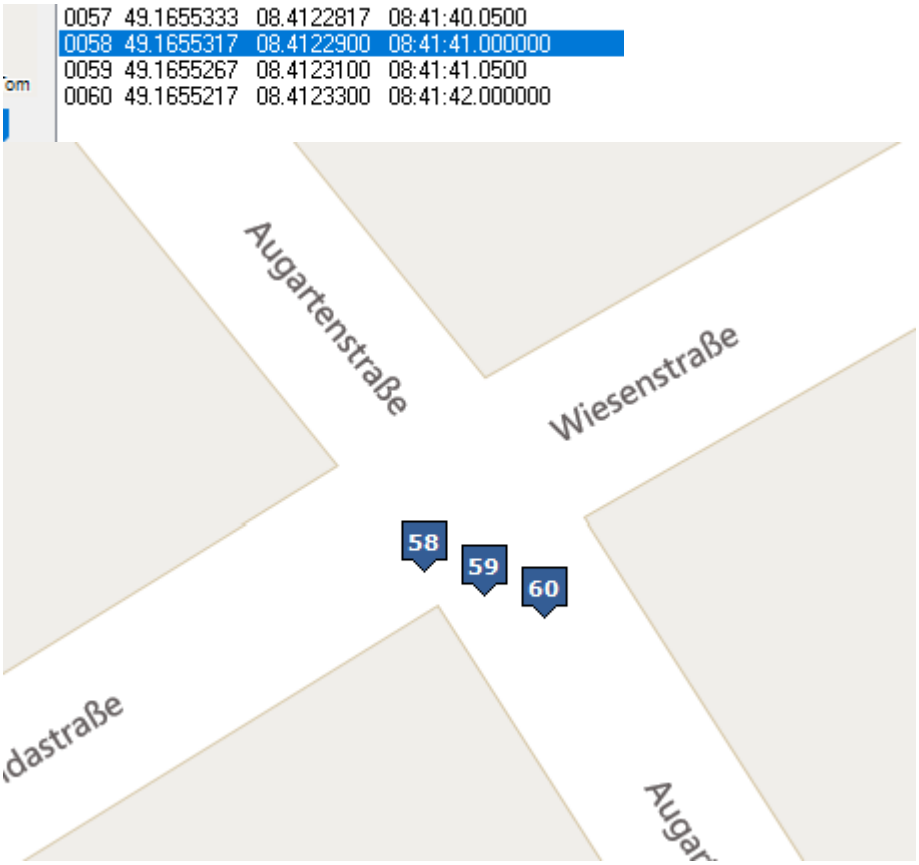
You can choose between 10 or 25 Movement Units.
The Amount of Movement is always the same, no matter, which ZOOM-Level is set.

Generally all selected Trackpoints will move by the same amount.

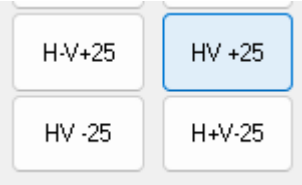
Also **V+** will move the Points "**Upwards**", while **V-** will move the Trackpoints "**Downwards**".
The **H-** will move to the **Left** Side, while **H+** will move the Trackpoints to the **Right** Side.

Lets take these Points. Trackpoint 58 is selected.

© 2023 Theo Gottwald * https://www.smart-package.com

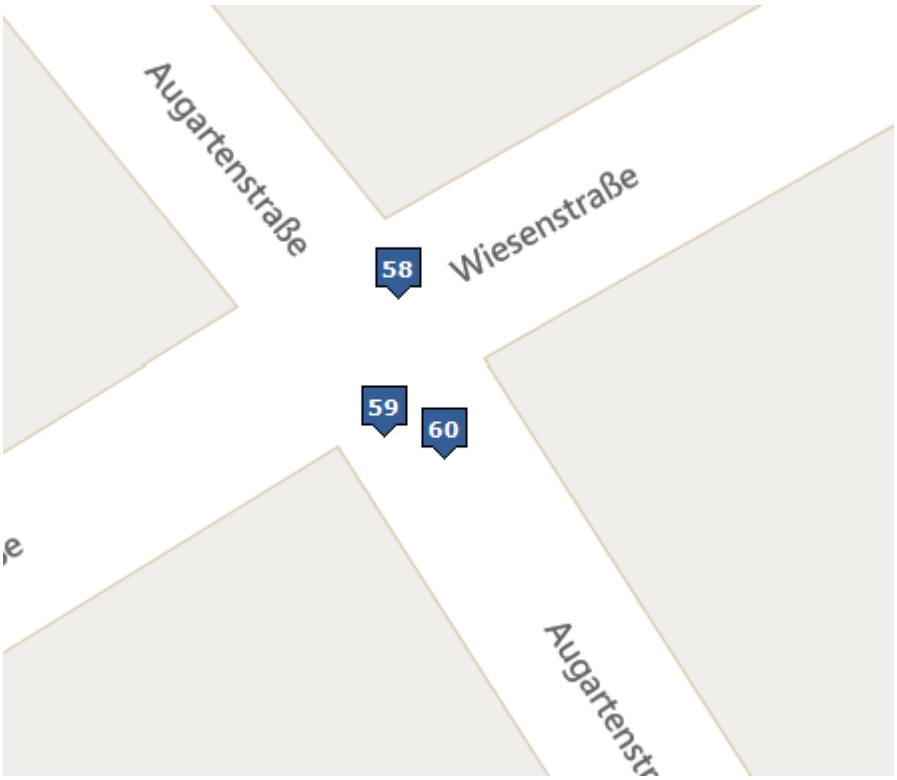


Now I press this Button:



Here is the result:

© 2023 Theo Gottwald * https://www.smart-package.com

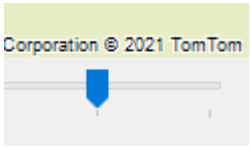


© 2023 Theo Gottwald * https://www.smart-package.com

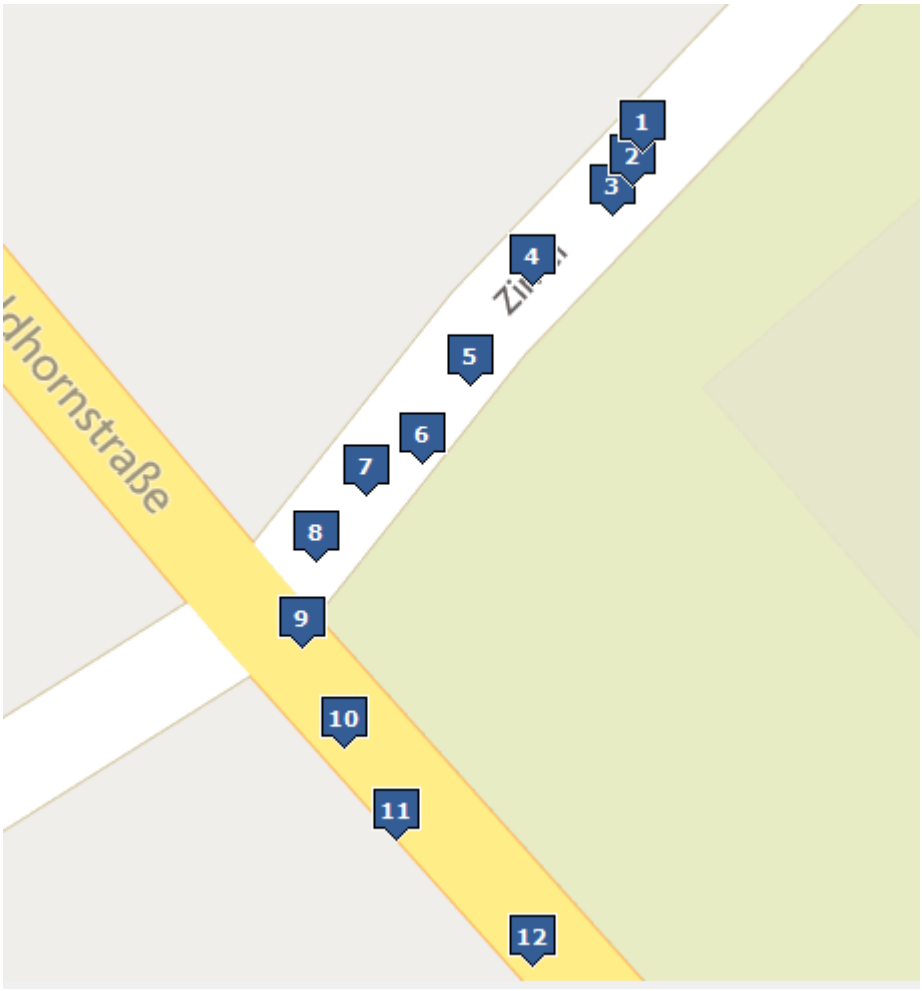
4.88.4 Line Points - Button

This Button is possibly the most important Button.
It can re-order Trackpoints on a L9ine between the two selected Points.

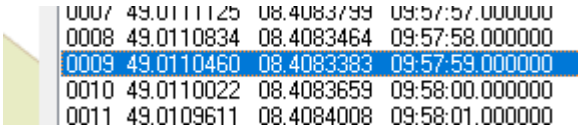
Let's see an Example.

We are in this Zoom-Level: 

Situation in the RAW GPX-File is this:

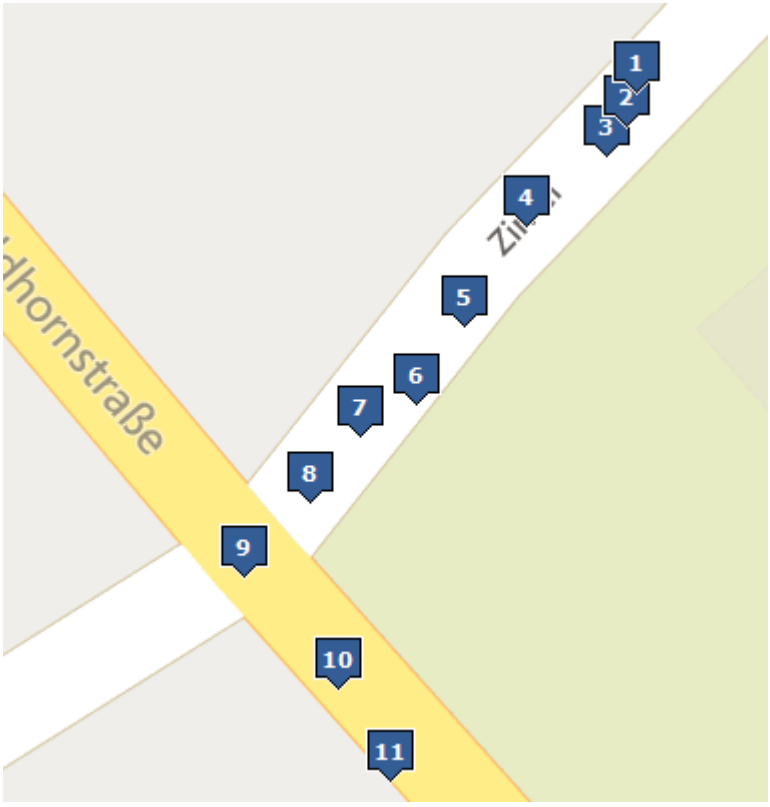


In the first step i will move Trackpoint 9 into the Center of the roads.
For this i will select TP 9 and then Move it using the H/V Buttons.



Now Situation looks like this:

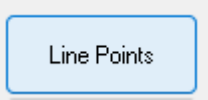
© 2023 Theo Gottwald * https://www.smart-package.com



Now I can use the "Line Points" Button, to move all Points between Trackpoint 1 and Trackpoint 9 on a Line.
For this i will select these two Trackpoints that mark the Start and End of the Line.
All Points between will be moved in a 90 Degree Angle towards this Line.
These Points are selected.

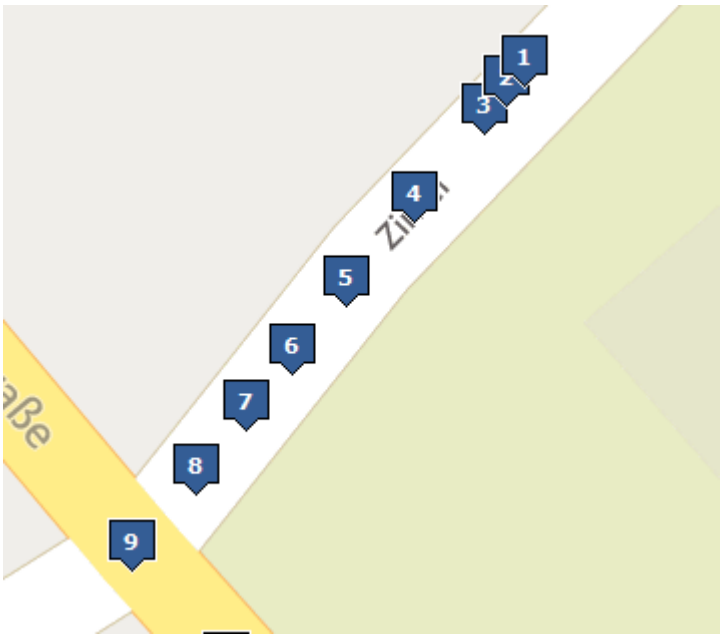
0001	49.0112642	08.4085655	09:57:51.000000
0002	49.0112494	08.4085585	09:57:52.000000
0003	49.0112355	08.4085457	09:57:53.000000
0004	49.0112048	08.4084915	09:57:54.000000
0005	49.0111611	08.4084510	09:57:55.000000
0006	49.0111271	08.4084178	09:57:56.000000
0007	49.0111125	08.4083799	09:57:57.000000
0008	49.0110834	08.4083464	09:57:58.000000
0009	49.0110510	08.4083033	09:57:59.000000

I press this Button:



Here we have the result. All Points between 1 and 9 are now on the Line.

© 2023 Theo Gottwald * https://www.smart-package.com



Here we have another example.

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

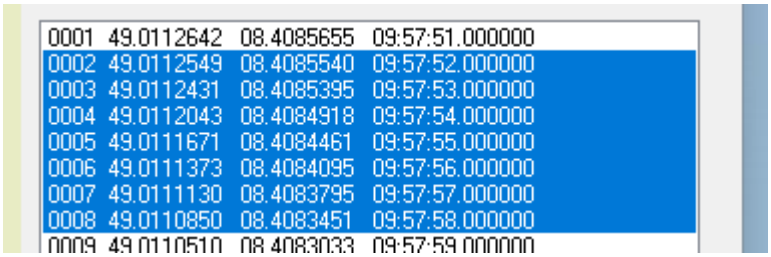
4.88.5 Make Inbetween / Delete Button

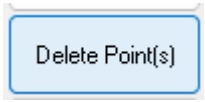
This Button will do 2 things:
1. It will set a new Trackpoint after each currently selected Point.
2. It will select all new created Points.

So what is it good for?

Lets take this situation:

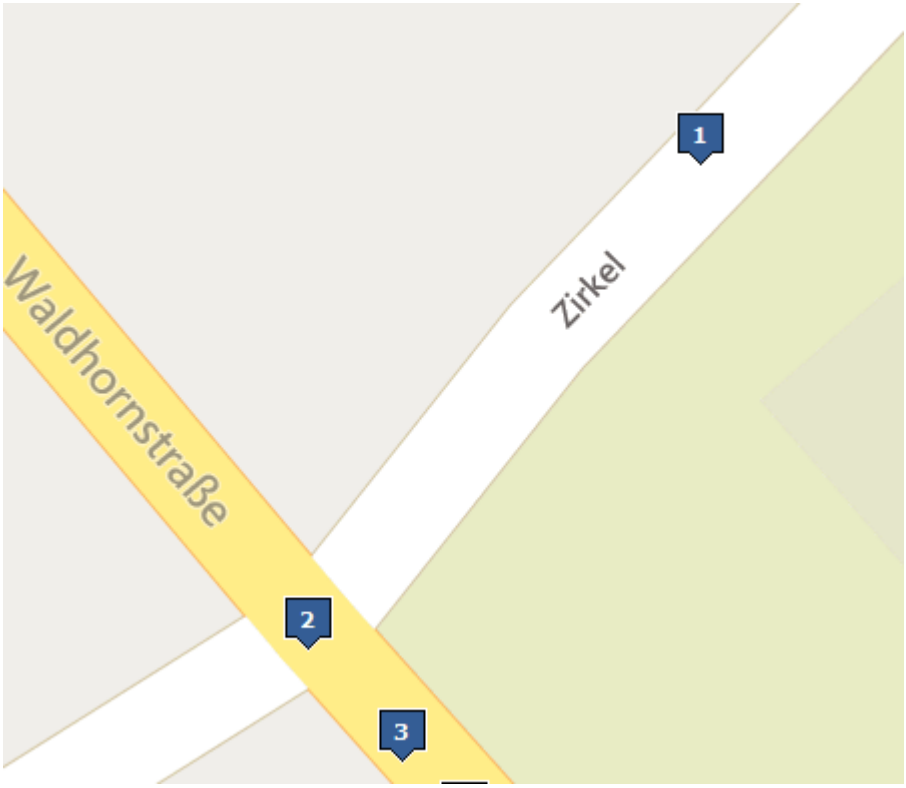
I am now going to delete the TrackPoints 2 to 8.



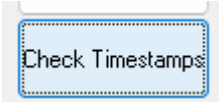


The situation looks like this:

© 2023 Theo Gottwald * https://www.smart-package.com



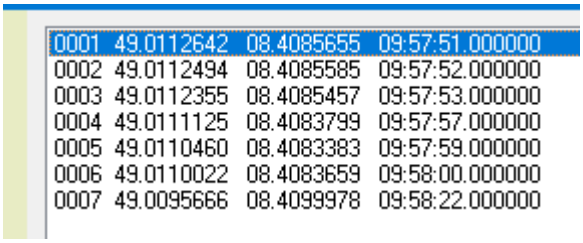
More important is that we have a time-problem because Street View will not accept a time distance larger then 5 Seconds.



And we have here a time distance of ...

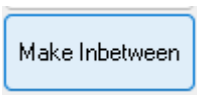
```
Checking TimeStamps:
Bad Time-Error between Stamps:  1/ 2=  8000
```

We can see it here also:



0001	49.0112642	08.4085655	09:57:51.000000
0002	49.0112494	08.4085585	09:57:52.000000
0003	49.0112355	08.4085457	09:57:53.000000
0004	49.0111125	08.4083799	09:57:57.000000
0005	49.0110460	08.4083383	09:57:59.000000
0006	49.0110022	08.4083659	09:58:00.000000
0007	49.0095666	08.4099978	09:58:22.000000

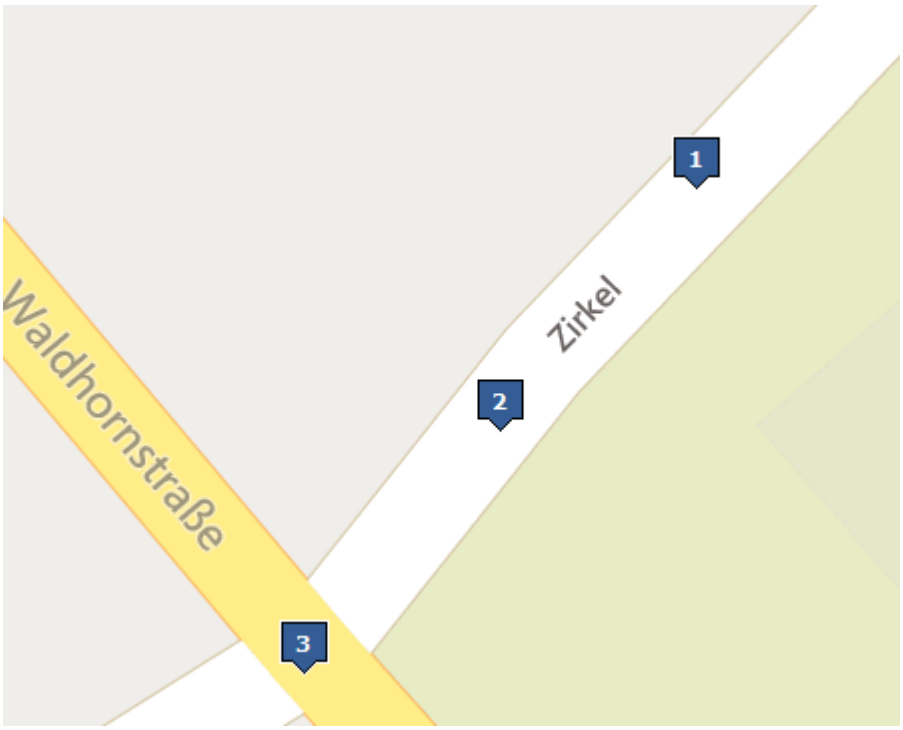
We can now select Trackpoint 1 - the point before the TP we want to insert - and press



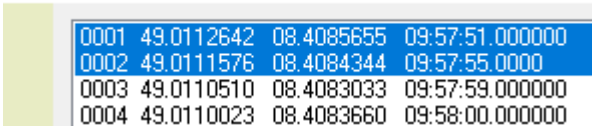
This will change the situation in 2 ways.

First we get one new Trackpoint here:

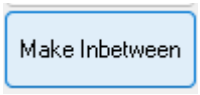
© 2023 Theo Gottwald * https://www.smart-package.com



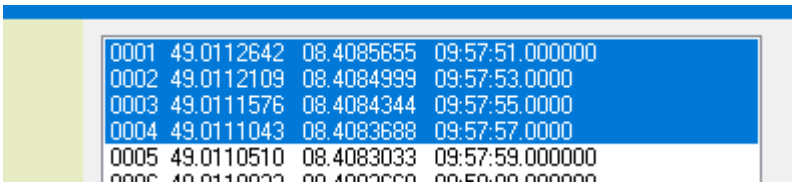
Second, the new TP is selected as well as the original TP.



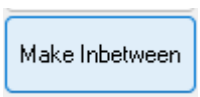
Now we can immediately press the Button again ...



Now we get this.

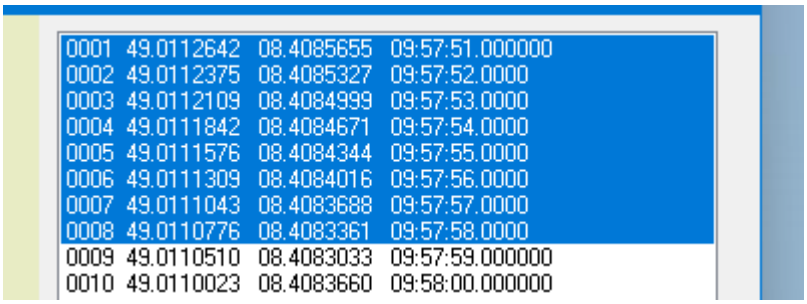


and we can immediately press the button once more.



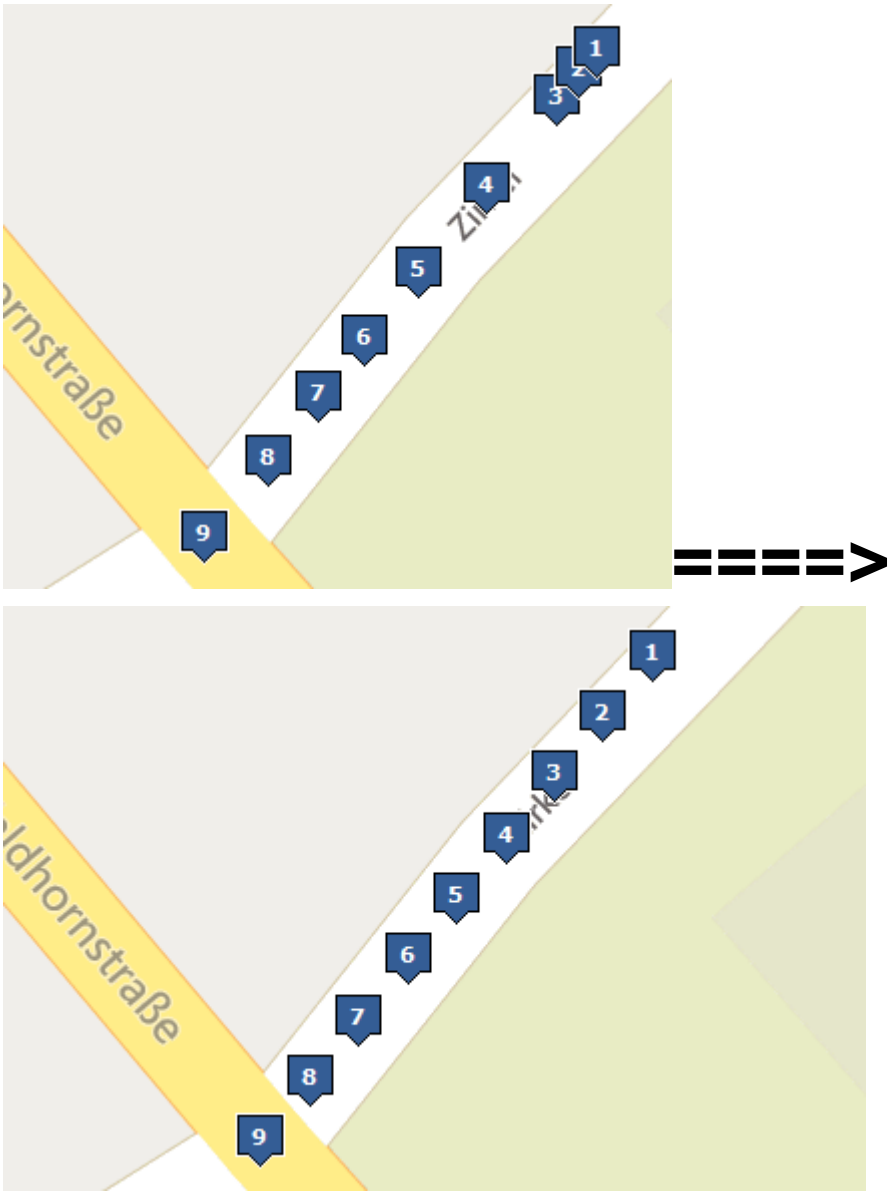
Now we get a Situation that looks like this:

© 2023 Theo Gottwald * https://www.smart-package.com



0001	49.0112642	08.4085655	09:57:51.000000
0002	49.0112375	08.4085327	09:57:52.0000
0003	49.0112109	08.4084999	09:57:53.0000
0004	49.0111842	08.4084671	09:57:54.0000
0005	49.0111576	08.4084344	09:57:55.0000
0006	49.0111309	08.4084016	09:57:56.0000
0007	49.0111043	08.4083688	09:57:57.0000
0008	49.0110776	08.4083361	09:57:58.0000
0009	49.0110510	08.4083033	09:57:59.000000
0010	49.0110023	08.4083660	09:58:00.000000

Where all (new) Points are perfectly distributed.
This should only be done in special cases because it may lead to pictures having wrong GPS Coordinates by some meters.
Therefore you should prefer the "Line Points" Button.

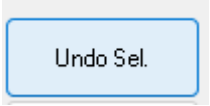


© 2023 Theo Gottwald * https://www.smart-package.com

© 2023 Theo Gottwald * https://www.smart-package.com

4.88.6 UNDO Selected - Button

During the Loading of the GPX File, the Program will make an internal Copy of the original file.
If you press this Button,

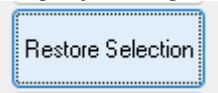


The program will replace actual values of ANY SELECTED TRACKPOINT with the original values that have been stored during the Loading of the file.
If you have added Trackpoints or deleted Trackpoints using this Button can lead to confusion.

*© 2023 Theo Gottwald * https://www.smart-package.com*

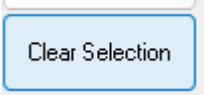
4.88.7 Restore/Clear Selection Buttons

During Operations, the Trackpoints-Listview may loose the Input-Focus and will not show properly which Trackpoints are selected.
Or it will show only one Trackpoint that is currently selected.
To get your original Selection back, press:



Then the Listview will display all currently selected Trackpoints.

If you want to clear any selection, use:

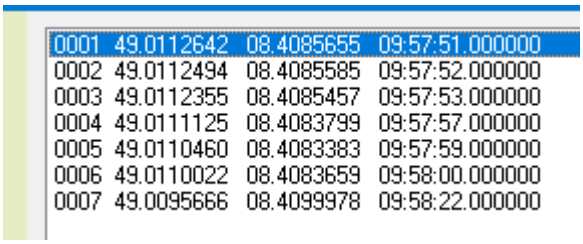


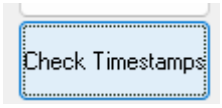
© 2023 Theo Gottwald * https://www.smart-package.com

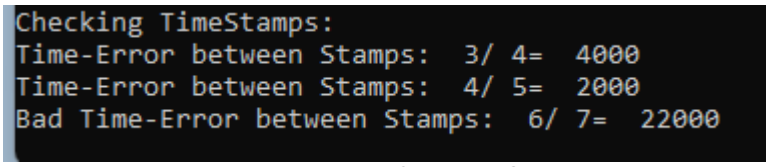
4.88.8 Check Timestamp Button

Generally Google-Street View will NOT accept GPX Files with Trackpoints taht are mor then 5 Seconds from each other.
This Function will do several Tests, if the Trackpoints are valid and the file is ready for Upload.

Here is an Example:







The results will be displayed in the Conso0le Output Windows.

The times that are displayed here are in ms. 4000 ms = 1 Second.

© 2023 Theo Gottwald * https://www.smart-package.com

4.88.9 Check and Repair Button

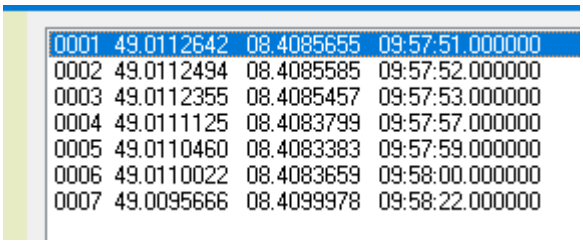
There are several Conditions, under which a GPX-File should not be uploaded because the results will not be as expected.

- Trackpoints must not be at the same time.
- And the time between Trackpoints must never exceed 5 Seconds.
- Trackpoints must be in the right Order -means a later trackpoint can not have an earlier time.

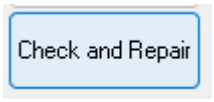
thee are several such conditions.
Using this button, all such Conditions are automatically checked and repaired using this Button.
The Button has some things in common with the "Check Timestamp Button",
but will automatically repair the Timestamps by adding Trackpoints and reordering if needed.

If there are Trackpoints added, these Trackpoints will always be positioned between current Trackpoints.

Here is an example.



0001	49.0112642	08.4085655	09:57:51.000000
0002	49.0112494	08.4085585	09:57:52.000000
0003	49.0112355	08.4085457	09:57:53.000000
0004	49.0111125	08.4083799	09:57:57.000000
0005	49.0110460	08.4083383	09:57:59.000000
0006	49.0110022	08.4083659	09:58:00.000000
0007	49.0095666	08.4099978	09:58:22.000000




```
Checking TimeStamps:
Checking for equal TimeStamps:
Added to Listbox  7
Inserting Trackpoints between:  3- 4
Added to Listbox  8
Added to Listbox  10
Inserting Trackpoints between:  7- 8
Added to Listbox  11
Inserting Trackpoints between:  10- 11
Added to Listbox  12
Added to Listbox  14
Added to Listbox  18
Added to Listbox  26
Added to Listbox  42
```

Resulting Time-Stamps after Repair:

© 2023 Theo Gottwald * https://www.smart-package.com

0001	49.0112642	08.4085655	09:57:51.000000
0002	49.0112494	08.4085585	09:57:52.000000
0003	49.0112355	08.4085457	09:57:53.000000
0004	49.0112047	08.4085043	09:57:54.0000
0005	49.0111740	08.4084628	09:57:55.0000
0006	49.0111432	08.4084214	09:57:56.0000
0007	49.0111125	08.4083799	09:57:57.000000
0008	49.0110792	08.4083591	09:57:58.0000
0009	49.0110460	08.4083383	09:57:59.000000
0010	49.0110022	08.4083659	09:58:00.000000
0011	49.0109574	08.4084169	09:58:00.0687
0012	49.0109125	08.4084679	09:58:01.0375
0013	49.0108677	08.4085189	09:58:02.0062
0014	49.0108228	08.4085699	09:58:02.0750
0015	49.0107779	08.4086209	09:58:03.0437
0016	49.0107331	08.4086719	09:58:04.0125
0017	49.0106882	08.4087229	09:58:04.0812
0018	49.0106433	08.4087739	09:58:05.0500
0019	49.0105985	08.4088249	09:58:06.0187
0020	49.0105536	08.4088759	09:58:06.0875
0021	49.0105087	08.4089269	09:58:07.0562
0022	49.0104639	08.4089779	09:58:08.0250
0023	49.0104190	08.4090289	09:58:08.0937
0024	49.0103742	08.4090799	09:58:09.0625
0025	49.0103293	08.4091309	09:58:10.0312
0026	49.0102844	08.4091818	09:58:11.0000
0027	49.0102396	08.4092328	09:58:11.0687
0028	49.0101947	08.4092838	09:58:12.0375
0029	49.0101498	08.4093348	09:58:13.0062
0030	49.0101050	08.4093858	09:58:13.0750
0031	49.0100601	08.4094368	09:58:14.0437
0032	49.0100153	08.4094878	09:58:15.0125
0033	49.0099704	08.4095388	09:58:15.0812
0034	49.0099255	08.4095898	09:58:16.0500
0035	49.0098807	08.4096408	09:58:17.0187
0036	49.0098358	08.4096918	09:58:17.0875
0037	49.0097909	08.4097428	09:58:18.0562
0038	49.0097461	08.4097938	09:58:19.0250
0039	49.0097012	08.4098448	09:58:19.0937
0040	49.0096564	08.4098958	09:58:20.0625
0041	49.0096115	08.4099468	09:58:21.0312
0042	49.0095666	08.4099978	09:58:22.000000

*© 2023 Theo Gottwald * https://www.smart-package.com*

4.88.10 ADD / SUB / GS / CLR / FND Buttons

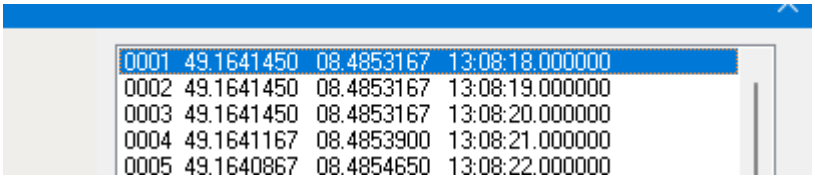
These Buttons will support you, if you CUT the Original Video at the start or End.
Using these features you can delete those Trackpoints that need be deleted if the Video is shorter.

Adapt a GPX-File to a Video where parts of the Start have been

Lets say you have a Video that is Originally starting from 13:08:18:0000
Now you need to delete 01:45 Seconds from the START.

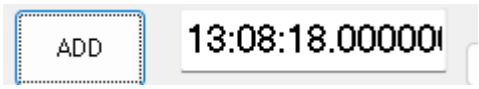
So you would do the following steps:

1. Select the Start-Timestamp (Nr.1)

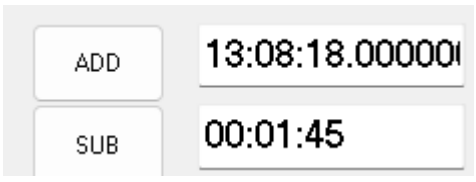


2. Click the "GS-Button to copy the Timestamp. (GS -> "Get Setting")

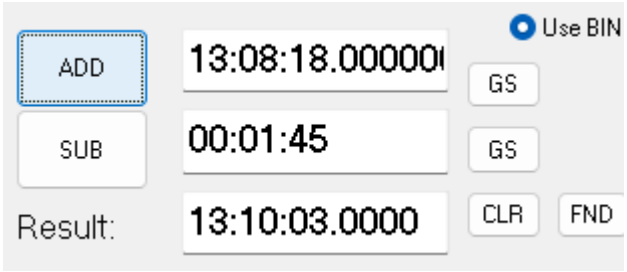
3. The result will be:



4. Now manually enter the Amount of time that you have cut of the start from the video, into the second field.



5. Now press the ADD-Button. The result will be:

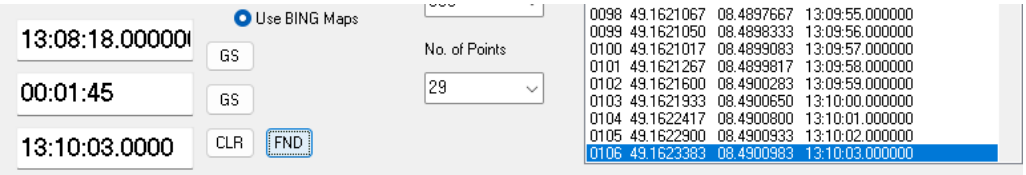


In the RESULT Field, you will see the new start of the GPX-File.

© 2023 Theo Gottwald * https://www.smart-package.com

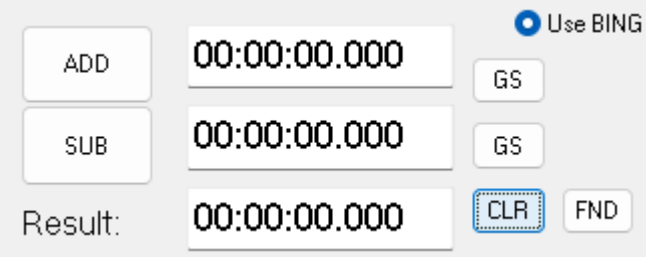
Now press the FND-Button to get the exact Trackpoint, that is the first Trackpoint in this video.
In the Listbox, the Trackpoint, that is the first Trackpoint in the video is now selected - in this Case it is Trackpoint Nr.16.

You will now delete all Trackpoints before this Trackpoint, then the GPX-File will fit o the Video.



Using the CLR-Button to clear all Fields

Next lets clear all field. Press

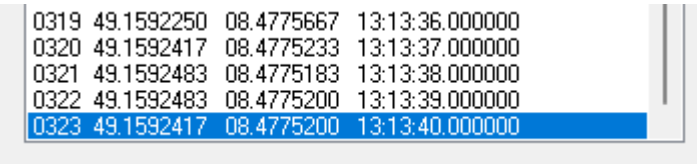


Adapt a GPX-File to a Video where parts of the End have been c

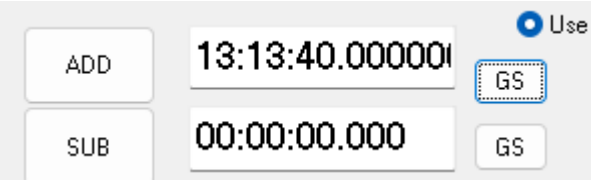
Lets say you have a Video that is Originally ending at 13:13:40:0000
Now you need to delete 01:45 Seconds from the END of the Video.

So you would do the following steps:

1. Select the END-Timestamp - the last Timestamp at the end of the Video.

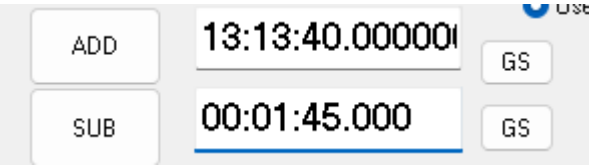


2. Click the "GS-Button to copy the Timestamp. (GS -> "Get Setting")

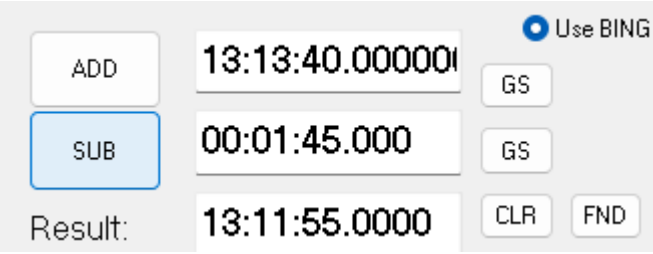


3. Enter the Time to "cut away" into Field 2

© 2023 Theo Gottwald * https://www.smart-package.com

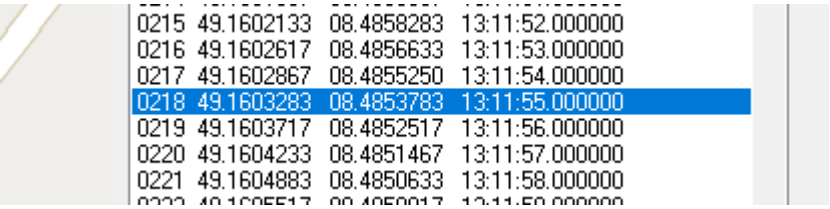


4. Press the SUB-Button to get this result:



In the RESULT Field, you will see the new END of the GPX-File.

Now press the FND-Button to get the exact Trackpoint, that is the last Trackpoint in this video.
In the Listbox, the Trackpoint, that is the last Trackpoint in the video is now selected - in this Case it is Trackpoint Nr.218.

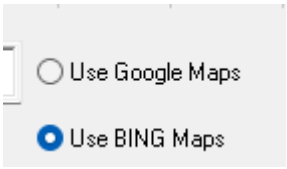


You will now delete all Trackpoints after this Trackpoint, then the GPX-File will fit to the Video.

© 2023 Theo Gottwald * https://www.smart-package.com

4.88.11 Choose between BING Maps and Google Maps

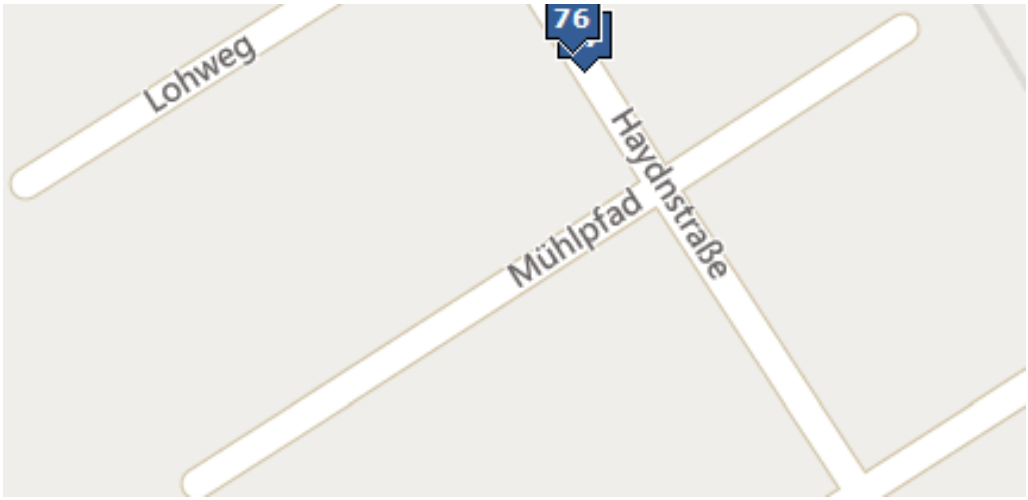
There is a option to choose between BING Maps API and Google Maps API.



Currently the Program only supports the BING Maps API, this is due to the fact that the resolution of the
Pictures we get from Google API is limited to 640x48 at the time of writing.
So just leave that on BING Maps.

You can add a Google API Key and the program will display a Google Map from that place. but currently without any GPX-Points.

If you have added the Google-Maps API, you can check if the Map Information is similar to Google Maps or not - like in this case.
Currently the program will not display any Trackpoints on the Google-Maps.



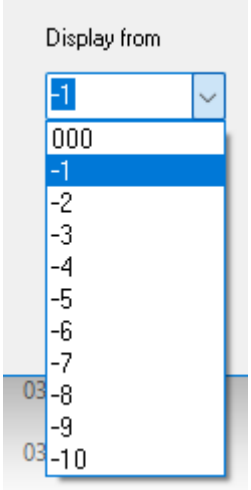
BING-Maps, Stand 01/2023

© 2023 Theo Gottwald * https://www.smart-package.com

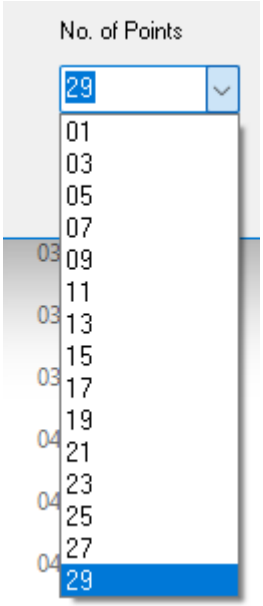
4.88.12 Display from / No. of Points

There are two Fields where you can select the number of Trackpoints, that are displayed, and you can decide in which Trackpoint to start.
Generally the display will always display 2 Trackpoint, the Selected Trackpoint and the next Trackpoint - if there is one.

To see which way the GPS-Data is going, it is recommended to set "**Number of Points**" on "29".
Most often there is no need to change the setting from "Display from".
If you work with the last selected Trackpoint, this setting comes in handy.



Choose the starting Trackpoint to display, always calculated from the first current selected Trackpoint.



Choose the number of Trackpoints that will be displayed, here a larger number is recommended.

© 2023 Theo Gottwald * https://www.smart-package.com

4.88.13 Additional Comments

Here are some additional Tipps on uploading to Google Street V

1. At the time of this writing, uploads are done at: Street View Studio

2. Stop Recording while turning the car. The Overlapping of Coordinates may lead to problems when uploading. All Coordinates should go in a clear direction.

3. Remove Trackpoints that are on the same location as other Trackponts. If needed make new Trackpoints just nearby.

4. Do not completely delete Trackpoints if not needed, because there is a dependence in the Timeline/Trackpoints of the GPX-File and the Pictures in the Video.

5. The "Marks on the road" will be at the **wrong place if the coordinates are not on the right place**. It looks like Google does not move these to the road at the time of writing.

Picture: The "Marks on the road" depend on the Coordinates that have been uploaded with the GPX-File.

© 2023 Theo Gottwald * https://www.smart-package.com

Last Changes & Updates

5 Last Changes & Updates

Last Changes and Updates

02.05.2023
Help-File:
Many new Illustrations and improvements in the Starting Guide.

Added the correct Flags for the FIndpattern Commands:

```
About the P5 - Parameter ("Debug Flag"):
The Flag-Parameter meanwhile can do a lot more then just help Debuging.

1 - Enable Debug-Output in Console Window
2 - Snap last used Window again if not found
4 - Use "Difference Map Search"
8 - Disable Filtering (if set)
16 - Use "Invert Pattern"
32 - Use "Complement Pattern"
64 - Use "Extended Search" (takes longer, makes more evaluations)
```

SPR-Editor:
Removed a nasty bug in the SPR-Editor that was there for quite a while.
It could lead to a unexpected crash of the Editor when loading a lot of Subscripts.
Now the bug has been removed.
Breakpoints in Subscripts have been repaired.

GPX-Korrektor
Improved "GPX-Korrektor.exe", it has now a "Overview" Button.
Improved "GPX-Korrektor.exe", it can do Satellite View in 2 Modes.

SPR:
Changes: TXT.glx / CPC.f
Lots of Tests.

13.03.2023
Added new SFD. / Set File Date - Command (untested).
Added new Component "GPX-Korrektor.exe" in an Subfolder.
There is an own Chapter for this new Tool in the Help-File.

Update to the XML-Command. Internal Searches start now from (Pos+1). This may affect old Scripts that use XML-Commands.

What is a Timestamp?
A Timestamp is a Time-Value in the Format: `HH:MM:SS` or
`HH:MM:SS.mmm`
Using these new `TCA.` and more new Time-Stamp Commands you can work easily with Timestamps.

```
TCA.Add|P1|P2[|P3]
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TCA.Dif|P1|P2[|P3]
TCA.Div|P1|P2[|P3]
TAC.Between|P1|P2[|P3]
TCA.Split|P1|P2|P3
TCA.Make|[P1][|P2][|P3]
TCA.Ovl|P1|P2[|P3]
```

also new:

```
IAT. / NAT.  If After Timestamp
IBT. / NBT.  If Before Timestamp
```

Using these new commands, TImestamps like "HH:MM:SS" can easily be **Added, Subtracted (Time-Difference),**
Divided, Split into Parts, Constructed and you can get the Timestamp between two Timestamps.

28.11.2022

New Command: GetGradientColor 1325 will calculate the Gradient color between two Colors with given Steps.

Error removed with

```
ANA.Color_Diff_MIN|$$C01|$$E05|$$ERA
```

Added Section with Graphical Subprograms 406 (Macros).
These Subprograms can be used for Picture Processing.

24.09.2022

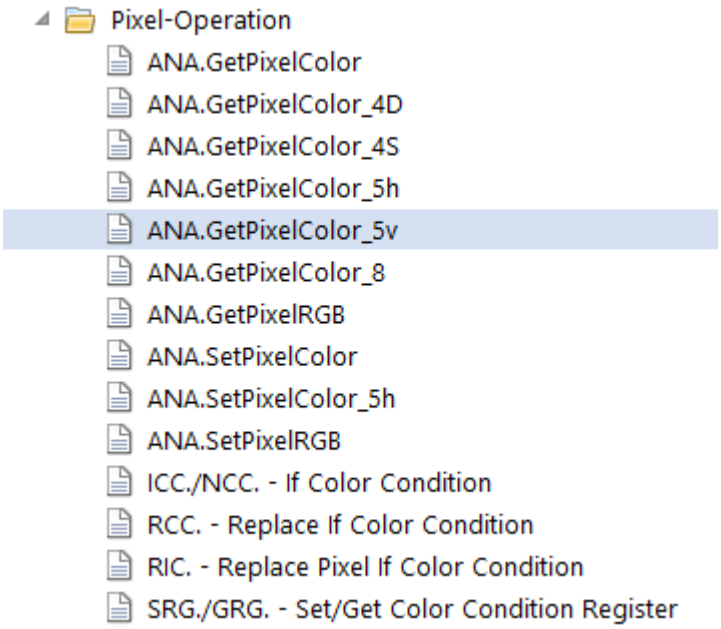
ANA.ReplaceColorT und ReplaceColorD have been improved.
Normally these Commands will replace a given Color with another color if the color fits the Color specification **P2 (+Tolerance)**.
Now there is the Option to prefix **P2** with an ! which make the Command switch to "Negative Mode", means it will replace all Colors that do NOT fit the specification in **P2**.

27.08.2022

Bugfix with DTV.
During the enhancement of DTV. to understand US-formatted Date's, there has been a little Bug which has now been removed.

New Chapter "Cleaning Scanned Writings" in the help-File.
You can use the SPR to clean-up bad scannings 1209.
For this there are new Commands

© 2023 Theo Gottwald * https://www.smart-package.com



These commands read and write the Colors of up to 8 Pixels in one run and therefore can speed up execution times of such Scripts.

03.08.2022

Added new commands:

```
DIC.Set_Multiple|$$RET|$$DEL|$$IND|$$VAL
DIC.Set_Value|$$REF|$$DEL|$$INX|$$VAL
DIC.Format_Numeric|$$LIN|$$DEL|$$IND|$$FMT|$$RES
```

These commands can be used to quickly change the Value/Data of one or more Cells of a csv-File or other delimited Data.

Changed:

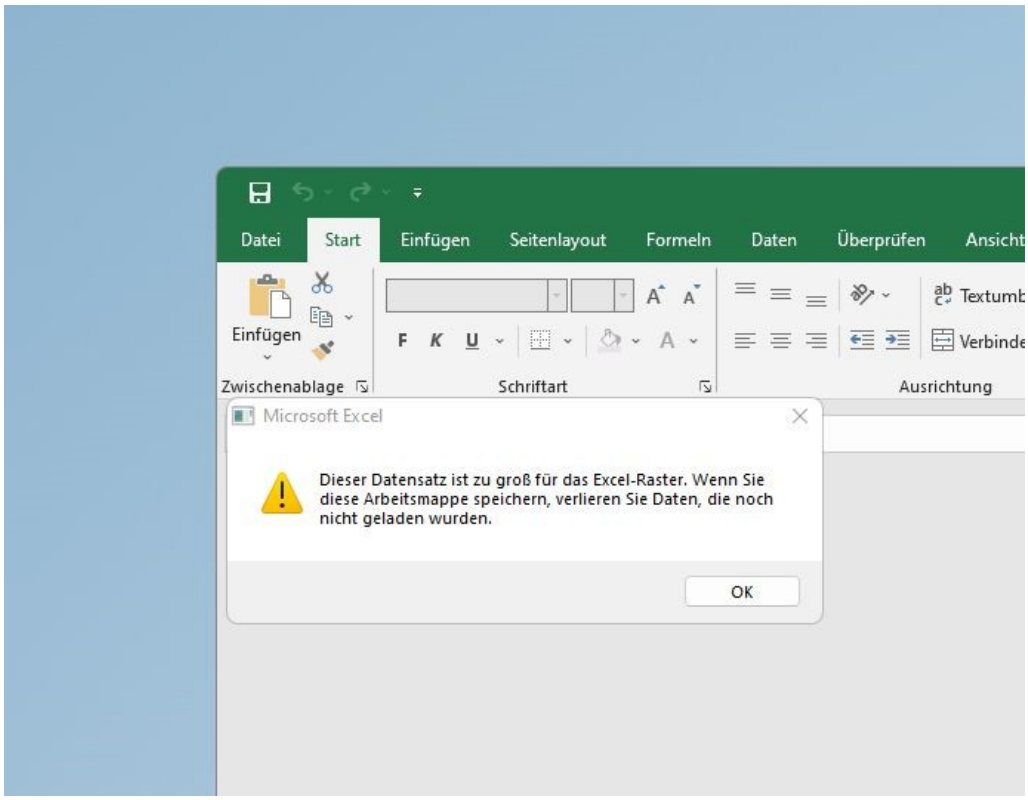
```
VDF. - Verfiy Date Format
```

Using this command, you can verify if a Variable contains a valid date. The command will now also convert a US-formatted Date like "2022/12/31" into a European type Date, that can be used with the other Date-Commands like `DTV.` and others. This was useful because in csv-Files there are often US-formatted Dates.

23.05.2022

Some smaller Corrections. Tests show that the SPR can handle csv-Files that are larger then Excel can handle.
A csv with more then 1.000.000 Lines can be processed using the SPR in less then 15 Seconds, while Excel will not accept it anymore.

© 2023 Theo Gottwald * https://www.smart-package.com



22.05.2022

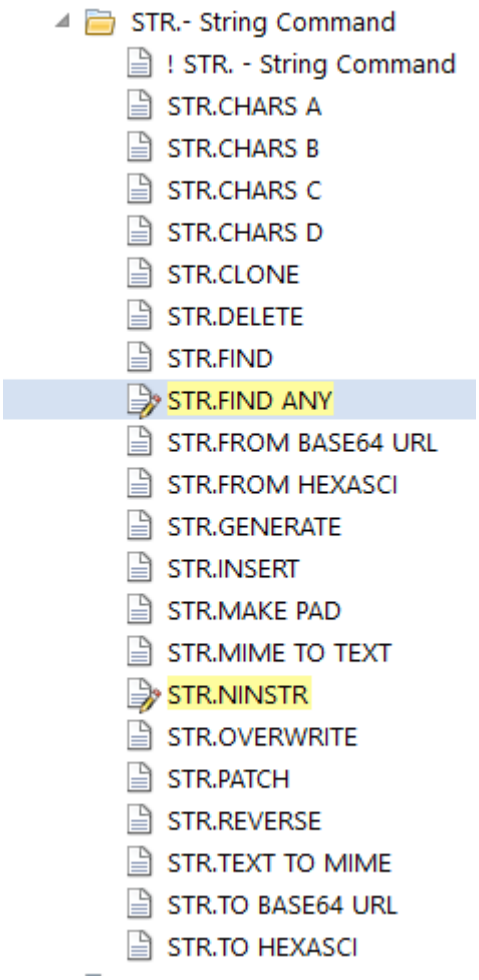
Added **many more commands to the TXT.-Command**, and also also the commands have been sorted by Type.
Some commands that are time intensive have been updated for Speed.
For example the TXT.Sort-Commands have been tested with a File with 354000 Lines and do their sort within a second or two.
Therefore the TXT.-Commands are now an own Category, not more in the String-Category.

`PRT.` Command improved. Can now print much larger Texts in one run.
Help for TXT.glp corrected.

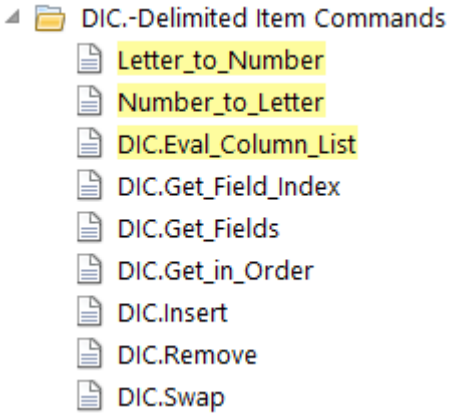
Excel-Function "`Excel_LetterToNumber`" and "`Excel_NumberToLetter`" have also been added to the DIC.-Command.
These have been changed internally and replaced by native programmed versions that will NOT call the Excel API.
This way these functions are much faster and do not rely on an Installation of Excel anymore.

New STR.-String Commands (Sort of Mulit-Instring):

© 2023 Theo Gottwald * https://www.smart-package.com



New DIC.-Commands



Many new commands for TXT.
These will complete the TXT.-Command-Set in a way that it can also be used as a virtual , multidimensional Array with Sorting capability.
It can be used as a Que, or as a Stack as well.

```
TXT.itas   and  TXT.insert_text_at_start
TXT.ilas   and  TXT.insert_line_at_start

TXT.itae   and  TXT.insert_text_at_end
TXT.ilae   and  TXT.insert_line_at_end
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
 TXT.itap   and  TXT.insert_text_at_pos
 TXT.ilap   and  TXT.insert_line_at_pos

 TXT.itbl   and  TXT.insert_text_before_line
 TXT.ilbl   and  TXT.insert_line_before_line

 TXT.ital   and  TXT.insert_text_after_line
 TXT.ilal   and  TXT.insert_line_after_line

TXT.sbcl   and  TXT.sort_by_column_letter
TXT.sbci   and  TXT.sort_by_column_integer
TXT.sbcf   and  TXT.sort_by_column_float

TXT.fic   and  TXT.find_first_in_column
TXT.flc   and  TXT.find_last_in_column
TXT.gcb   and  TXT.get_column_block
```

and much more.

30.04.2022

New Command:

`TXT.set_text_lines|$$LIA|$$LIB|$$REP`

Lines `$LIA` (from Begin of the line) bis `$$LIB` (until end of this line) will be removed/replaced with the content of `$$REP`

Change: `TXT.frv` is now `TXT.fov` as it was documented.

Sample:

```
:Lab_Remove_Cols
$$FIK=?exeloc\Test.txt
$$LIA="Q","M","K","I","H",F"
TXT.toc|$$BAK
TXT.Load_pure_Text|$$PAA|$$FIK
TXT.Delimited_Items_Remove|;|$$LIA
TXT.Get_pure_Text|$$PAA
TXT.fov|$$BAK
TXT.Get_Line_Count|$$NUL
RET.
```

24.04.2022

These commands are useful for Excel- and ***CSV-Files***. You can remove Columns, Insert Columns, or Swap Columns.
In real life many stuff is in Excel-Files, and often we have to work with these.
It was possible using SBD. before but now it becomes really easy.

Added:

```
TXT.crp / TXT.cr_to_crlf
TXT.lfp / TXT.lf_to_crlf
TXT.crr / TXT.crlf_to_cr
TXT.lfr / TXT.crlf_to_lf
TXT.gtls /TXT.Get_Text_Lines
TXT.gtb / TXT.Get_Text_Block
TXT.gdi / TXT.get_delimited_item_from_line
```


© 2023 Theo Gottwald * https://www.smart-package.com

```
TXT.gda / TXT.get_any_delimited_item_from_line
TXT.gdit /TXT.get_delimited_item_from_text
TXT.gdat /TXT.get_any_delimited_item_from_text
```

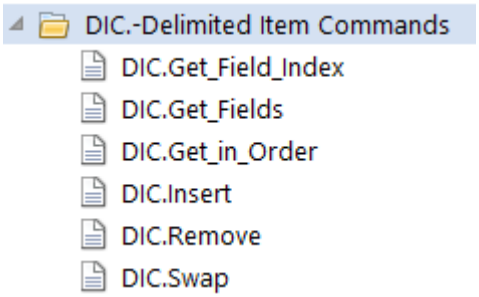
These 3 are also new. They will process not just a Line but the complete TXT.-Buffer.

```
TXT.Delimited_Items_Insert
TXT.Delimited_Items_Remove
TXT.Delimited_Items_Swap
```

More new Command:

```
DIC. - Delitimed Item Command

DIC.Swap|$$DEL|$$LIN|$$INA|$$INB|$$RES
DIC.Insert|$$DEL|$$LIN|$$IND|$$RES
DIC.remove|$$DEL|$$LIN|$$IND|$$RES
DIC.Get_in_Order|$$DEL|$$LIN|$$IND|$$RES
DIC.Get_Fields|$$DEL|$$LIN|$$IND|$$RES
```

These commands can be found here3078.



24.04.2022
Many Changes and Bug Fixes with the Pattern-Search.
There were Bugs that could make the Patternsearch crash if used extensively (WFP. ?).
These Bugs have been removed.

Also Pattern-Search is now faster and has more Options like X-Ray and Extended Search.

Even searching for 5 Patterns at the same time will be in a second with an actual CPU.
Its amazing fast.
Internally the Engine can already search 6 Patterns at the same time, but we did not make a command for that at the time of this writing.

Added lots of new Pattern-Sizes1862 that are not yet supported from the Pattern-Recorder (HE-Recorder).
Looks like a completely new Pattern-Recorder would be fine.

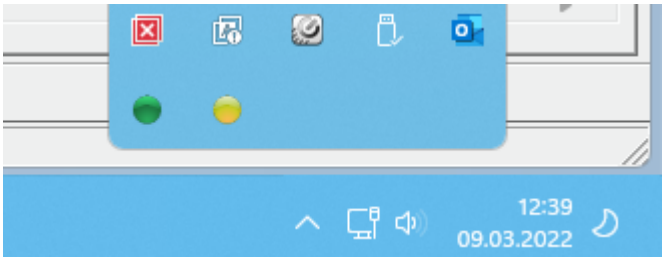
There are no more "Bad Patterns". If its a Pattern, you can use it. All Patterns are now fine.

Searching for more then 1 Pattern will now generally use IR 0 for the Target Bitmap, and IR 1 ... IR (number of Pattern to search for) for the Patterns. This is a general rule for a "Multi-Pattern Search Commands".
Therefore these "Multi-Pattern Search Commands" commands will overwrite IR 0 and the following IR's.

© 2023 Theo Gottwald * https://www.smart-package.com

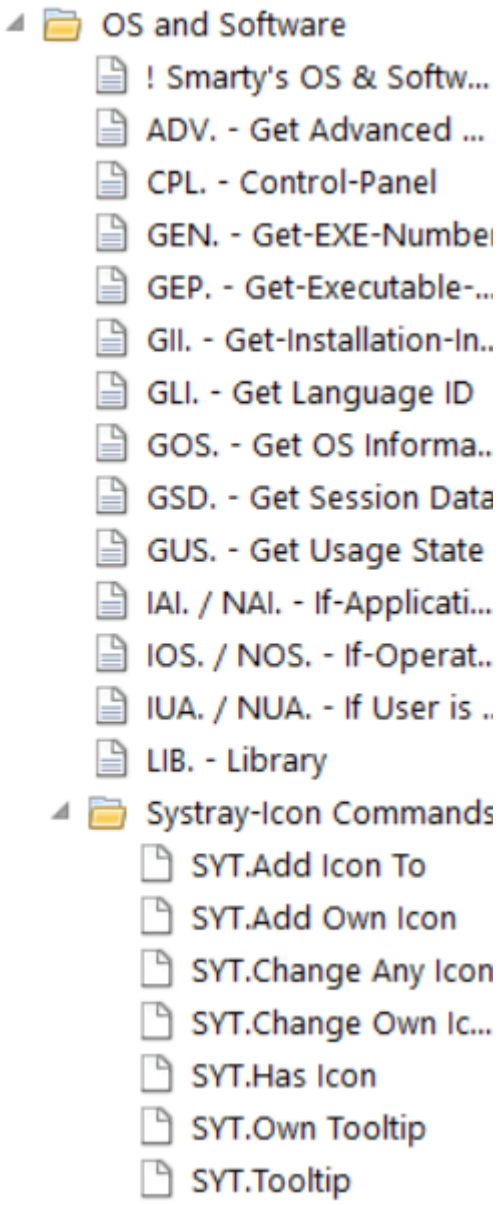
If you use these IR's save them somewhere (ANA.ToVar) or move them into higher IR's (ANA.Move)
that are unused - for example IR 10 and following.

09.03.2022
New Command `SYT.` (Systray-Icon) mostly completed.
You can now add an Systray-Icon, also a Tool-Tipp is supported, but not yet any menu.
You can also change the Icon (make it blink).



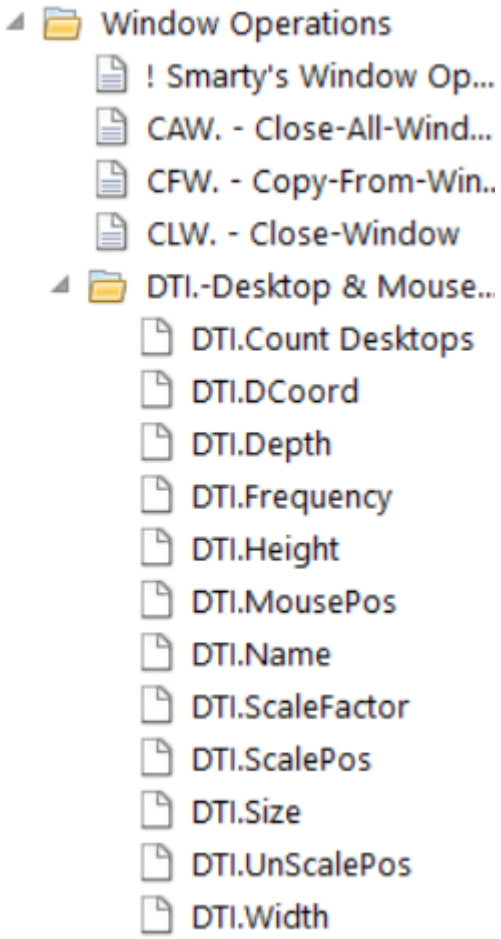
The new SYT.-Command can be found under "OS & Software".

© 2023 Theo Gottwald * https://www.smart-package.com



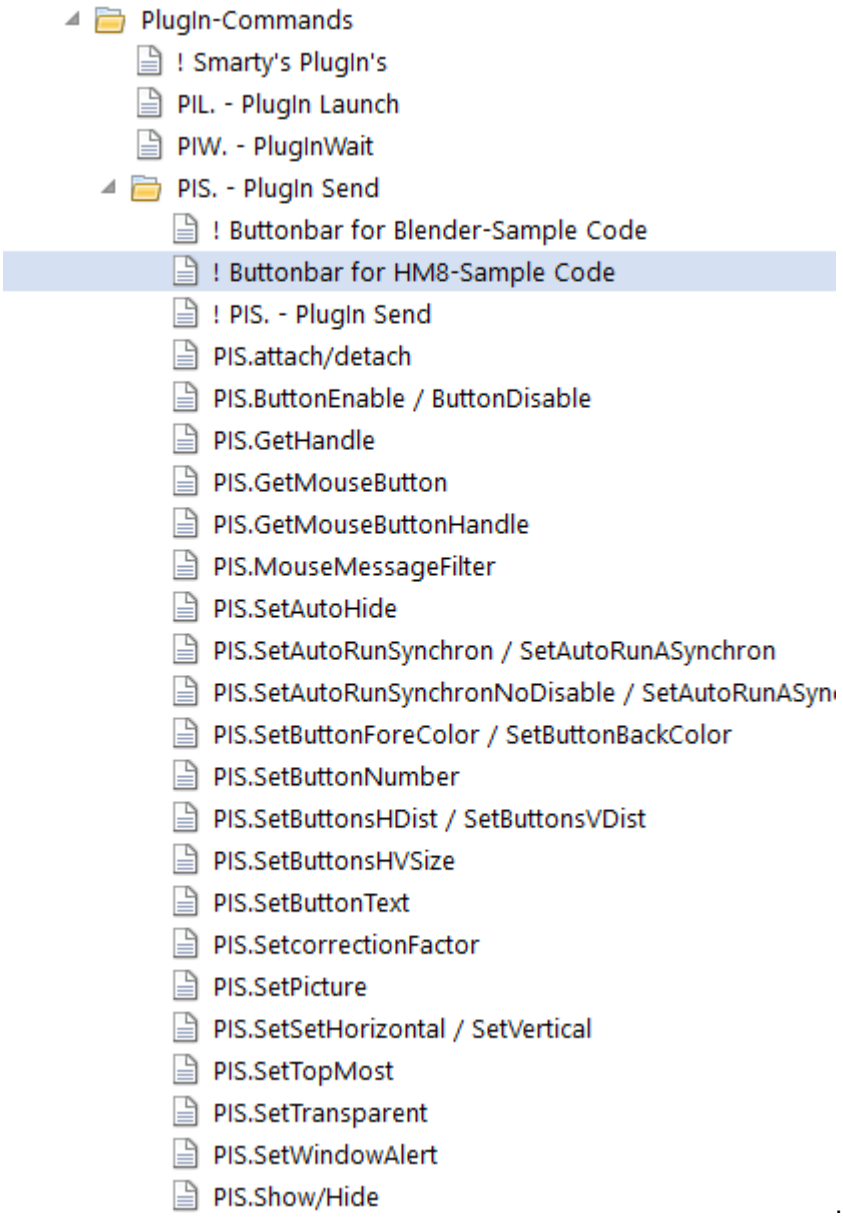
02.03.2022
New Command `DTI.` (Desktop-Information) completed.
This command will give all needed Information on number of visible Desktops, coordinates, DPI-Scaling etc.

*© 2023 Theo Gottwald * https://www.smart-package.com*



14.02.2022
In this Update i have revisited the "Plugin_01.exe" the Buttonbar.
The help was incomplete and is now completely documetned:

*© 2023 Theo Gottwald * https://www.smart-package.com*



New: `PIS.GetMouseButtonHandle`
New: `PIS.attach` - lots of more Options on how to attach the Buttonbar
Bei den PIS.Befehlen gab es Änderungen, bitte die Hilfe beachten, wenn etwas nicht geht.

```
PIS.inv -> PIS.Hid
PIS.vis -> PIS.Sho
PIS.SetVertikal -> PIS.SetVertical
PIS.SwitchTopMost->PIS.SetTopMost
```

Improved:
Help for PIS. has been updated and improved. Many new help-Pages.

Efficiency of the running Butonbar is improved (needs less CPU).
Behavior when Target WIndow was minimized has been improved.

`PIS.SetButtonBackColor`
now works properly and can permanently change the Button Color.

© 2023 Theo Gottwald * https://www.smart-package.com

Buttons in the Buttobar (Plugin_01) do not have any default Text by now.
Also you can now remove any Text using:
`PIS.$$REA|sBt|1|`
for Example if you want to display a Picture, using
`PIS.$$REA|SetPicture|1|$$FIX|32|5`
this command has been Improved.
It can now Scale the pictures to any size, if **P4** and **P5** are specified. This way you can use any pictures for the Buttonbar, just specify the size they should have on the Button.

New feature:
If Size or distance is specified with "zero" then the previous value is preserved.
For example:

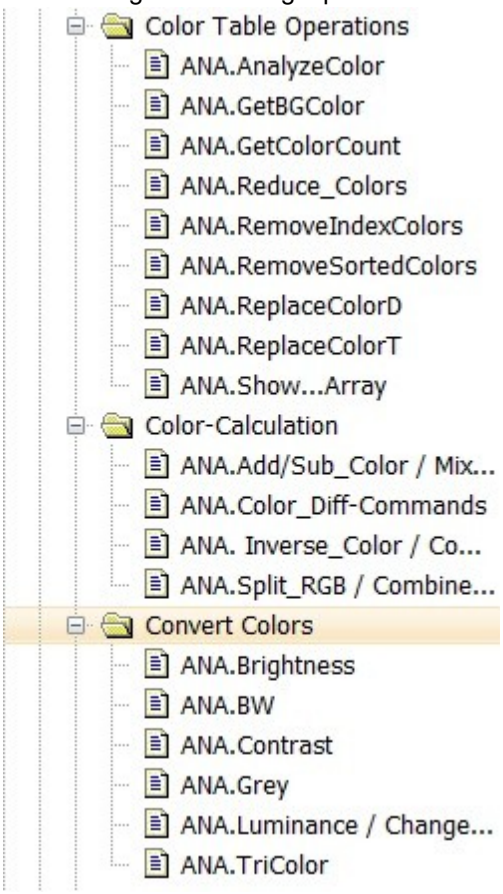
```
PIS.$$REA|SetButtonsHVSize|270|0
```

Here the Y-Size of the Buttons is not changed.

11.02.2022
1. Lots of Changes in the Help-File.
2. Undercover Graphics (UCV.) moved to Image Processing Topics.
3. Helpfile is now available as E_Book (PDF-Format) in the "E-Book-Helpfile" Folder.

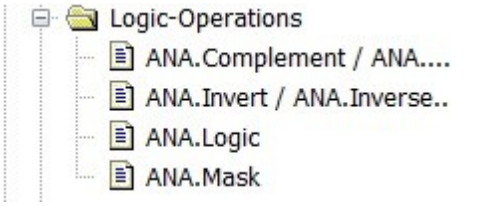
27.01.2022
Added: ANA.ColorDiffMap-Command
This commandcan be used to generate Outlines, it will find borders of an Object.

20.01.2022
Lots of Image Processing Options have been added:

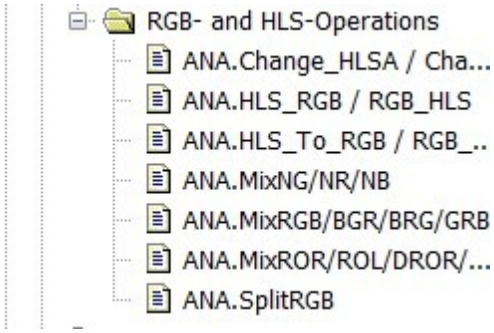


Also there were changes and addition with the Logic Operations that can be used to combine Images (IR's).

© 2023 Theo Gottwald * https://www.smart-package.com



And there is a bunch of RGB/ HSL-Operations



The SPR can now be used to snaphot, convert, resize, cut and process pictures in Brightness, Color and Contrast.
I have tested Image resolutions up to.10200x6800 Pixels*, which is a good resolution even for commercial pictures..
Using the SPR for commercial Pictures? Yes, is possible. Here is an example.

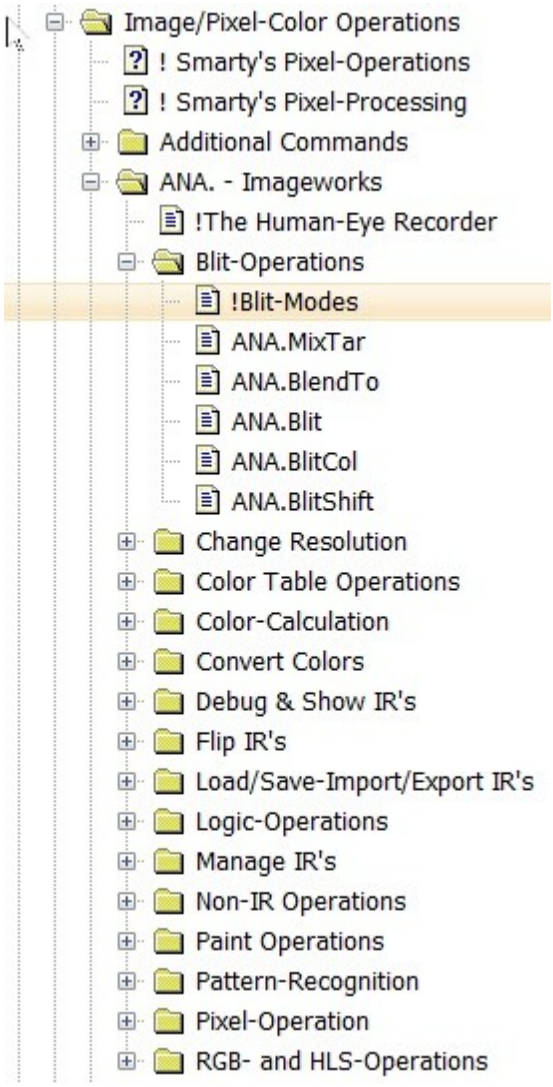
I have set this Picture for Sale on Dreamstime, see: "Color Wheel Background".

You can do all sorts of automated Image Processing using the SPR.

More examples:
- Remove or Add Artefacts or Watermarks
- Clean Up Pictures
- Change Color and Brightness
- Change Resolution
- Cut Borders of pictures
- a very strong side of the SPR is the **ability to Combine Pictures**
You can combine up to at least 20 Pictues in any logic way.
Using very fast commands like ANA-Logic, or use a Pixel-based Script using any of the other commands.
If the pictures do not originally have the same resolution, use ANA.EqualRes to stretch one of the IR's to the Target resolution.

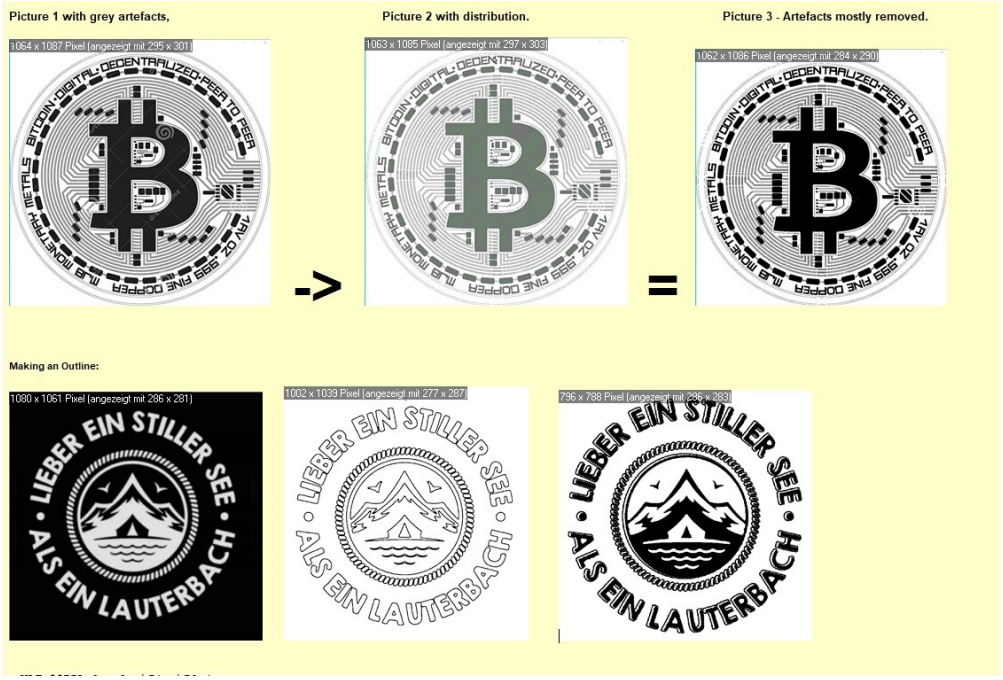
Here is an overview over the current Help-File Sections that are mostly new..

© 2023 Theo Gottwald * https://www.smart-package.com



The following examples are from ANA.Logic and ANA.Mask.

© 2023 Theo Gottwald * https://www.smart-package.com



For more details please see the command descriptions.

27.11.2021
Improved:
- GFP. and SFP. got new Option "LH", "LastHits" which returns the number of Pixel-Hits from the last find.

```
WTP.$$PAA|$$PAB|30,40|60
```


*© 2023 Theo Gottwald * https://www.smart-package.com*

```
GFP.LH|$$PAT
MBX.Pixelhits: $$PAT
```

- GFP. and SFP. got new Option "FA", "FlagA" which will set the internal "Flag A" to a value that is currently 0,1,2 or 3.
This will influence the behavior of the Fuzzy Logic during the Pattern-Search. The higher the number, the more Pixels will be evaluated, and therefore Patterns will be recognized.

Fixed
```
GFP.LastPattern|$$PAT
```
Now you can find out which of multiple Patterns was found for example with
```
WTP.$$PAA|$$PAB|30,40|60
GFP.LastPattern|$$PAT
MBX.$$PAT
```

26.11.2021
- Fixed: Patterncommands LTP. / WTP. etc. with Multiple Patterns when Patternsize was not 8x8
- Fixed: DMP.-Command since last change Numbers were not recognized

17.11.2021 Added:
- ANA.CutRect
- ANA.FromClipboard
- ANA.ToClipboard and ANA.ToClipboardRect
These commands place pictures in the WindowsClipboard.

########################
Release Date for Free and Visual Version
These Versions do not have all upcoming Updates.
########################

16.11.2021 Compiled Free/Sample Version
Therefore all Improvements after this date are not added to the Free-Version, as it will only be updated once a year.

07.11.2021
Improved Color-Analyzer in SPRE and HE-Recorder.

06.10.2021
Added new Option:
Inline-Data using VDA. and VDE.
New Option for Pattern-Search: "ID:" for direct use of Inline Data.
New Option for Pattern-Search: "IR:" for direct use of Image Register.

Changed:
SCS. now for String-Data and use SCN. now for numerical cases.

New:
CFP.-Command for SCS. Multiple-Choice
ESL. - ESL. - End SELECT Loop

Changed/Repaired:
NFC., NWC., NDC.,NLC. have been changed/repaired.
SPR-Editor: Problems with closing the Editor when using "auto-loaded files" solved.

Added SPRE:
Many new Commands and

© 2023 Theo Gottwald * https://www.smart-package.com

VDA.-VDE. to the SPR-Editor. It can be folded.

SPRE also:
Floaterbar got several new Buttons:
D - Datamaker / Q - Quadrant-Analyzer / H - HE-Recorder
U1, U2,U3 - for User-Scripts
Floaterbar is now resizable.
Improved Editor-Recording should work now.

SPR: New Commands:
IFP. - If FInd Pattern .. EIF.
NFP. - If Not FInd Pattern .. EIF.
ITP. - If Two Pattern
NTP. - Not If Two Pattern
LFP. - Locate FindPattern
LTP. - Locate Two Pattern
CFP. - CASE FindPattern
JFP. - Jump FindPattern
JTP. - Jump If any of Two Pattern
JTN. - Jump If none of Two Pattern was found
SFP. - Set FindPattern Parameter
GFP. - Get FindPattern Parameter
WFP. - WFP. - Wait For Pattern
WTP. - Wait For ANY of Two Patterns
JNP. - JNP. - Jump NOT FindPattern

Added several Multithreaded versions of Office- and Browser commands.
AVF. - similar Command like AMS. for Browser and Office
AFF. added to AFT.
AGF. - added to AGR.

New Mousecommand:
MDL. - Mouse Drag Left - Drag Items with the Mouse
MRM. - Mouse Move Right Click - Faster Mousemove and Right Click

Several changes around Screen DPI Settings.
Robot should now generally record and replay in 100% Mode.

New Keyboard-Commands:
Faster:
KBI. - Keyboard Input-Buffer Commands, all options from KBC. but typing is faster because characters are directly placed in the Input buffer.
SDH. - Send Key Hex-Code,easily access all hidden characters of a font using the "Alt-NumPad" Trick
SDA. - Send Key ANSI, Faster Typing using new Technology yet still compatible Engine
SDU. - Send Key Unicode, Faster Typing using new Technology yet still compatible Engine

Improved KBC.H-Paste-Over supports now Parameter to Paste over

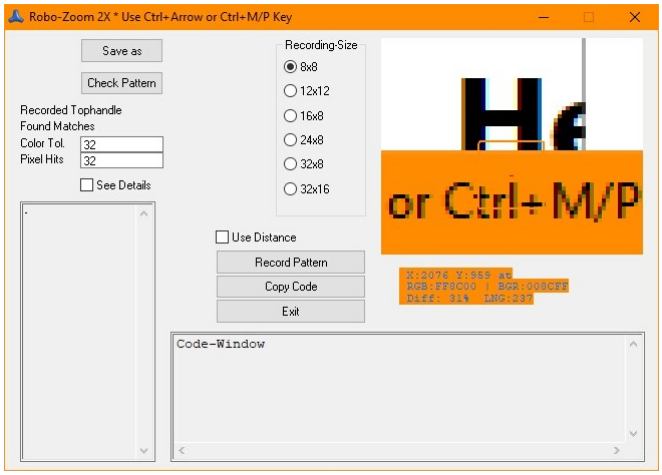
```
KBC.h-paste over|$$TXT
```

New Conversion Commands:
ATI. - Convert ANSI to UNICODE
UTA. - Convert UNICODE to ANSI

© 2023 Theo Gottwald * https://www.smart-package.com

14.072021
MAP. - Command changed so its compatible with MMV.
ANA.Snapfind1 Command is not the Simplest way to locate a Pattern in a window.
GAI. - did not work properly - now should work.
GFS. - neu codiert noch nicht alles getestet.

30.06.2021
Added "Human-Eye-Recorder"- Using this Tool you can make Snapshots of Patterns for use with the Find-Pattern Commands.
There is also a new Option "Use Distance" that will prevent "Mouse Over Effects" becauseyou Record while the Mouse is "far away".
The new Tool is currently lo0cated in the Main Folder.



There are multiple changes and enhancements with the ANA.-Command.

31.05.2021
Changes:
Changed to BSC: and BCS. Now these commands do have more Clicking-Options.
Additions in the Helpfile to these Commands.

ANA.-Command 1136 offers a lot of Options to recognize GUI Elements that could not have been recognized until now. In fact, ANA. con now find anything.

Added: `?pattern\` as equivalent to "?path\Pattern\". This is the default Path for Search Patterns, if no other path is given. It is used for example with the ANA.Load Command.
Added: ANA.findpclick
This command is complete package that will do all the work when you want to click whatever
on WEB-Sites. You just give the Pattern, and you say the details (how often to clock, how to click etc.).
And let it run.

Known Problems:

Robo-Diff:
Using Robo-Diff will only work on minimalistic Systems, with just Windows and the Application to watch.

© 2023 Theo Gottwald * https://www.smart-package.com

Due to the 32 bit-Architecture of Robo-Diff, the program can currently not handle the large Data Amount of heavily loaded 64 bit Systems.
This could only be changed if the Program would save its data while processing, and this would make the whole process much slower.
So there is currently the advice to use Robo-Diff only on minimalistic Systems with just Windows and the program to record.

Editor:
- Running a Script that uses a Macro-Library. Under some circumstances the Editor will switch into a library Macro File and show this file instead of the Script file that is executed. In such case close the Editor and load the Script file new.

© 2023 Theo Gottwald * https://www.smart-package.com

- ! -

! Smarty's Preprocessor 246

- # -

'#aft: 310
#aos 273
#aos: 273
'#AOS: 273
#cbw# 334
#CIL 332
#cil: 332
'#CIL: 332
'#CRY: 302
#cwh# 334
#date# 334
#datetime# 334
#dsince#- 334
#dtime# 334
#EIF 262
#exc 295
#exc: 295
'#EXC: 295
#EXE 277
#exe: 277
'#EXE: 277
#exemode# 334
#EXV 280
#exv: 280
'#EXV: 280
#fex# 334
#FromVar 1406
#fxd# 334
'#hid: 283
#hwn# 334
#ib# 334
#ICF: 316
'#ICF: 316
#ICO: 313
'#ICO: 313
#ICS: 318
'#ICS: 318
#icx# 334
#IF 262
#IF PARAMS 262
#IF RECURS 262
#ih# 334
#il# 334
#INC 248
'#INC 290
'#inc: 290
#ir# 334
#it# 334
#iw# 334
#ixp#
#iyp# 334
#LIB: 298
#linenum# 334
#MAN 321
#man: 321
'#MAN: 321
#mau: 325
'#MAU: 325
#mpx# 334
#mpy# 334
#mutex1# 334
#mutex2# 334
#OEND 251
#ONCE 251
#opx# 334
#opy# 334
#pid# 334
#RAU 271
'#RAU 271
#RAU: 271
'#RAU: 271
#sbp# 334
#since# 334
#spi 287
#spi: 287
'#SPI: 287
#ssx#
#ssy# 334
#stack# 334
#tid# 334
#til# 334
#time# 334
#timer# 334
#tio# 334
#tos# 334
#VER 328
#ver: 328
'#VER: 328
#wsx#
#wsy# 334

- $ -

$$ 366
bel 334

© 2023 Theo Gottwald * https://www.smart-package.com

bs 334
$cmdexe$ 334
$cmdl$ 334
cpb 334
cr 334
$crlf$ 334
$date$ 334
dq 334
$dtos$ 334
eof 334
esc 334
$exename$ 334
ff 334
lf 334
now 334
nul 334
$Script$ 334
sp 334
sq 334
$startdate$ 334
$startscript$ 334
$starttime$ 334
tab 334
$time$ 334
tos 334
$utime$ 334
vt 334

- & -

&AND: 341
&ANDTHEN: 341
&DOCAPS: 341
&EXACT: 341
&FUZZY: 341
&H 352
&H0 352
&NOT: 341
&NOTEXT: 341
&OR: 341
&TOLERANT: 341
&WILDC: 341
&XOR: 341

- : -

: 361
:on_error 364

- ? -

?auto 345
?auto\ 345
?cauto 345
?cauto\ 345
?cdesktop 345
?cdesktop\ 345
?cstarmen 345
?cstarmen\ 345
?cstartprog 345
?cstartprog\ 345
?curdir 345
?curdir\ 345
?desktop 345
?desktop\ 345
?exeloc 345
?exeloc\ 345
?exepath 345
?exepath\ 345
?exepfad 345
?exepfad\ 345
?fonts 345
?fonts\ 345
?mainpath 345
?mainpath\ 345
?path 345
?path\ 345
?pfad 345
?pfad\ 345
?progs 345
?progs\ 345
?progs64 345
?progs64\ 345
?sendto 345
?sendto\ 345
?startmen 345
?startmen\ 345
?startprog 345
?startprog\ 345
?sysfolder 345
?sysfolder(xx) 345
?temp 345
?temp\ 345
?wi 345
?wi\ 345
?ws 345
?ws\ 345
?ws64 345
?ws64\ 345

*© 2023 Theo Gottwald * https://www.smart-package.com*

- § -

§§ 366
§§§ 253
§§§99 253

- 1 -

16to32 446
16to8 449

- 3 -

32to16 452

- 8 -

8to16 455

- A -

ach 3043
ach. 3043
Addon 4336
adv 2572
adv. 2572
AFF. 558
aft 558
AFT. 558
After EQ 4006
AGF. 564
AGR. 564
alb 3045
alb. 3045
ams 570
ams. 570
ANA. HLS_RGB 1741
ANA.#FromVar 1406
ANA.Add_2_Colors 1307
ANA.Add_3_Colors 1307
ANA.Add_4_Colors 1307
ANA.Add2 1307
ANA.Add3 1307
ANA.Add4 1307
ANA.AnalyzeColor 1225
ana.blit 1172
ANA.BlitCol 1175
ANA.BlitShift 1179
ANA.Brightness 1336
ana.bw 1342
ANA.cco 1328
ANA.cdb 1315
ANA.cdg 1315
ANA.cdmax 1315
ANA.cdmin 1315
ANA.cdr 1315
ANA.cdrgb 1315
ANA.Change_HLSA 1734
ANA.Change_HLSM 1734
ANA.Color_Diff_B 1315
ANA.Color_Diff_G 1315
ANA.Color_Diff_Max 1315
ANA.Color_Diff_Min 1315
ANA.Color_Diff_R 1315
ANA.Color_Diff_RGB 1315
ANA.ColorDiffMapX 1429
ANA.ColorDiffMapXY 1429
ANA.ColorDiffMapY 1429
ANA.Combine_RGB 1328
ANA.Compare 1474
ANA.Complement 1435
ANA.Complementary_Color 1298, 1435
ANA.Contrast 1352
ana.copy 1477
ANA.CutRect 1186
ANA.dbo 1488
ANA.dli 1493
ANA.DownScale 1530
ANA.DrawBox 1488
ANA.DrawKLine 1493
ANA.drawline 1493
ana.equalres 1189
ANA.Fill 1499
ANA.FindClick 1560
ana.findpattern 1537
ana.findpattern2 1546
ANA.FindPattern4 1550
ana.findpclick 1560
ANA.FromClipboard 1410
ANA.Get_Data 1564
ANA.GetBGColor 1239
ANA.GetColorCount 1242
ANA.GetPixelColor 1619
ANA.GetPixelRGB 1655
ana.getres 1192
ANA.GetResult 1567
ANA.GetResultsPack 1571
ANA.GetXYFromBytePos 1195
ANA.GetXYFromPixelPos 1198
ANA.gpc 1619
ANA.gprgb 1655

*© 2023 Theo Gottwald * https://www.smart-package.com*

ANA.Grab 1201
ANA.gres 1567
ana.grey 1358
ANA.grpi 1571
ANA.gxyb 1195
ANA.gxyp 1198
ana.hflip 1395
ANA.hgrad 1503
ANA.HGradient 1503
ANA.hls_to_rgb 1745
ana.hvflip 1398
ANA.Inverse_Color 1442
ana.invert 1442
ANA.Load 1412
ana.logic 1449
ANA.Luminance 1366
ANA.Mask 1455
ANA.Mix_2_Colors 1307
ANA.Mix_3_Colors 1307
ANA.Mix_4_Colors 1307
ANA.Mix2 1307
ANA.Mix3 1307
ANA.Mix4 1307
ANA.MixBGR 1756
ANA.MixBRG 1756
ANA.MixDROL 1765
ANA.MixDROR 1765
ANA.MixGRB 1756
ANA.MixNB 1750
ANA.MixNG 1750
ANA.MixNR 1750
ANA.MixRGB 1756
ANA.MixROL 1765
ANA.MixROR 1765
ANA.Move 1479
ANA.New 1481
ANA.PatternFillHwn 1511
ANA.PrintAT 1516
ANA.Reduce_Colors 1246
ANA.RemoveIndexColors 1254
ANA.RemoveSortedColors 1259
ANA.ReplacecolorD 1268
ANA.ReplaceColorT 1275
ana.resizeto 1204
ANA.RGB_HLS 1741
ANA.Save 1415
ANA.sco 1328
ANA.Set_Data 1564
ANA.SetPixelColor 1665
ANA.SetPixelRGB 1691
ANA.Show 1387
ANA.showhits 1390
ANA.ShowHots 1390
ANA.ShowSimArray 1284
ANA.ShowSortArray 1284
ANA.Snap 1418
ANA.snapfind1 1581
ANA.spc 1665
ANA.Split 1780
ANA.Split_RGB 1328
ana.SplitRGB 1780
ANA.sprgb 1691
ANA.Sub 1307
ANA.Swap 1484
ANA.ToClipboard 1421
ana.tovar 1424
ana.tricolor 1377
ana.vflip 1401
ANA.vgrad 1521
ANA.VGradient 1521
ANA_BlendTo 1168
ANA_MixTar 1162
ANA_PatternFill 1508
AND 458
andthen 341
Any Flags Set 4010
Any Flags Zero 4016
append 3569
Application 137
area 156
asc 3047
asc. 3047
atf 946
atf. 946
attach 2755
Attrib 4121
Attrib by Name 4131
Attrib by Number 4128
Attrib Value by Name 4142
ATU. 3237
AVF. 575

- B -

background 1921
bch 3049
bch. 3049
bcs 2431
bcs. 2431
Before EQ 4026
Between 4034
Between DQ 4022
Between SQ 4030

*© 2023 Theo Gottwald * https://www.smart-package.com*

bgr 1921
binary 3569
bit 441
BIT. 441
BIT.16to32 446
BIT.16to8 449
BIT.32to16 452
BIT.8to16 455
BIT.AND 458
BIT.Clear 461
BIT.GET 464
BIT.LShift 467
BIT.NOT 470
BIT.OR 473
BIT.RotateLeft 476
BIT.RotateRight 479
BIT.RShift 482
BIT.SET 485
BIT.Show 488
BIT.Store 491
BIT.ToBin 494
BIT.Toggle 498
BIT.ToVal 501
BIT.XOR 505
blb 3051
blb. 3051
blend 1971
blender 2710
blend-in 1974
blend-out 1977
blendto 1168
blit 1156, 1172
BlitCol 1175
Blit-Modes 1156
BlitShift 1179
box 1925
Browser and Office Commands 554
bsc 2435
bsc. 2435
btc 635
btc. 635
ButtonDisable 2763
bw 1342

- C -

cae 2996
cae. 2996
cal 2315
cal. 2315
can 2990
can. 2990
cap 1103
cap. 1103
cas 2987
cas. 2987
CASE 2982
caw 3586
caw. 3586
cax 2320
cax. 2320
cbr 582
cbr. 582
ccd 1109
ccd. 1109
ccv 1117
ccv. 1117
cdd 638
cdd. 638
cff 948
cff. 948
cfi 951
cfi. 951
CFP. 2993
cfu 2067
cfu. 2067
cfw 3589
cfw. 3589
chd 955
chd. 955
choice 2982
choose 2982
chr 3062
chr. 3062
cle 4106
clean 4108
Clean Text 4108
Clear 461, 3745, 4095
clear error 4106
client 156
CLOSE 516
close bracket 4038
clp 2009
clp. 2009
clw 3592
clw. 3592
cnb 672
cnb. 672
cns 2070
cns. 2070
color 1088
Commands 2008
Comment 4042

*© 2023 Theo Gottwald * https://www.smart-package.com*

Compare 1474
compiler 60
compiling 60
coordinate 356
cop 957
cop. 957
copy 1082, 1477
cov 1929
cover 1929
CPC. 2425
cpl 2574
cpl. 2574
cpr 2882
cpr. 2882
csc 960
csc. 960
csp 3064
csp. 3064
cst 2331
cst. 2331
ctf 964
ctf. 964
cti 966
cti. 966
ctr 2884
ctr. 2884
CutRect 1186
cvf 969
cvf. 969

- D -

dat 689
dat. 689
Data-Block Commands 671
date 688
dbc 2863
dbc. 2863
dbm 765
dbm. 765
dbo 767, 1488
dbo. 767
dbp 769
dbp. 769
dbv 771
dbv. 771
dbw 1933
dc 1937
dcp. 971
DDD. 2012
Debugging Commands 763
dec pos 4146
decimal 352
Decrement Position 4146
definition 361
del 974
del. 974
delete 1082
desktop bw 1933
desktop copy 1937
desktopwide 3752
detach 2755
DFD. 2015
DIC. 3086
did 2916
did. 2916
dip 773
dip. 773
dit 2018
dit. 2018
dlf 2074
dlf. 2074
dli 1493
dlv 2076
dlv. 2076
dmp 775
dmp. 775
dnob 4155
docaps 341
dol 2299
dol. 2299
Down Close Bracket 4152
Down Open Bracket 4155
dpi 1123
dpi. 1123
draw 1962
draw to 1962
drawbox 1488
drawline 1493
drs 427
drs. 427
dsd. 2021
dtc 693
dtc. 693
dtv 746
dtv. 746
Dump 4149

- E -

efx 922
efx. 922

*© 2023 Theo Gottwald * https://www.smart-package.com*

eif 2148
'EIF- 253
eif. 2148
'EIF+ 253
eli 1941
ellipse 1941
els 2150
els. 2150
Encryption 812
end 2362
End Attribute 4047
End Name 4051
End Tag 4055
END% 253
end. 2362
enr 2169
enr. 2169
equalres 1189
error 364
ESC. 2999
esl. 3002
eve 851
eve. 851
evg 854, 857
evg. 854, 857
exa 872
exa. 872
exact 341
exb 876
exb. 876
exc 879
exc. 879
Excel|CLOSE 516
Excel|Exist 518
Excel|FORMULA GET 520
Excel|FORMULA SET 522
Excel|GET_CELL_TEXT_BY_COORD 524
Excel|GET_CELL_TEXT_BY_RANGE 527
Excel|LASTCOLUMN 530
Excel|LASTROW 530
Excel|LETTER_TO_NUMBER 532
Excel|NUMBER_TO_LETTER 534
Excel|OPEN 536
Excel|PAGE_UPDOWN 538
Excel|SAVE 540
Excel|SAVEAS 542
Excel|SCROLL_LEFTRIGHT 545
Excel|SCROLL_UPDOWN 547
Excel|SENDKEY 549
Excel|WRITE 552
Exclude 3741, 3743, 3745
exd 881
exd. 881
exe 883
exe. 883
Execution Commands 867
exf 885
exf. 885
exh 890
exh. 890
EXIST 518
exm 892
exm. 892
exo 895
exo. 895
exr 902
exr. 902
exs 905
exs. 905
exu 910
exu. 910
exv 912
EXV. 912
exx 916
exx. 916

- F -

fef 924
fef. 924
fel 927
fel. 927
fem 929
fem. 929
fep 932
fep. 932
fer 935
fer. 935
fes 2550
fes. 2550
fev 937
fev. 937
fgr 1945
fif 977
fif. 977
fil.cmp 987
fil.Compare 987
fil.ffn 1009
fil.find free number 1009
fil.generate numbered files 1006
fil.generate numbered name 1012
fil.gnf 1006
fil.gnn 1012

© 2023 Theo Gottwald * https://www.smart-package.com

fil.join delete 990
fil.join delete. 990
fil.join file 994
fil.join file. 994
fil.len 998
fil.Length 998
fil.mov 1015
fil.move numbered file 1015
fil.split file 1001
fil.split file. 1001
fil.swap files 1018
fil.swp 1018
file 1085
File-Commands 940, 4304
fill 1499
findpattern 1537
findpattern2 1546
FindPattern4 1550
FindPClick 1560
Firefox 401
Flags 3981
Flags to Text 4158
fmt 3107
fmt. 3107
font 1987, 1990
font set 1990
for 2302
for. 2302
foreground 1945
FORMULA GET 520
formula set 522
found 3756
fpa2 1546
fpa4 1550
fpc 1020
fpc. 1020
fpi 2886
fpi. 2886
FromClipboard 1410
FromVar 1406
ftp 2078, 2080, 2088
ftp download 2080
ftp. 2078
ftp.ftp download 2080
FTP.tdl 2085
FTP.Test Download 2085
FTP.test with password 2088
ftp.tpw 2088
ftxt 4158
fuzzy 341
fzc 2334
fzc. 2334

- G -

GAC\ 3752
GAI 591
GAI. 591
gar 3619
gar. 3619
gbc 640
gbc. 640
gbe 675
gbe. 675
gbn 677
gbn. 677
gci 642
gci. 642
gcn 2091
gcn. 2091
gct 3622
gct. 3622
gdf 2889
gdf. 2889
gec 825
gec. 825
gek 2240
gek. 2240
gen 2580
gen. 2580
gend 4165
gep 2582
gep. 2582
gerterr 4111
ges 3067
ges. 3067
Get Asc 4191
Get Attrib by Value 4135
Get Attrib Count 4139
Get Block 4161
GET CELL TEXT BY COORD 524
GET CELL TEXT BY RANGE 527
Get Error 4111
Get First Node 4171
Get Flags 4168
Get Flags as Text 4174
Get Len 4177
Get Level 4179
Get Position 4182
Get Position for Previous 4185
Get Root 4188
Get Start End Tags 4194
Get this Name 4203

© 2023 Theo Gottwald * https://www.smart-package.com

Get XML Part 4209
GET_CELL_TEXT_BY_COORD 524
GET_CELL_TEXT_BY_RANGE 527
getres 1192
getxyfrombytepos 1195
getxyfrompixelpos 1198
gfb 680
gfb. 680
gfd 1023
gfd. 1023
gfi 1026
gfi. 1026
gfl 1031
gfl. 1031
GFP. 1869
gfs 3122
gfs. 3122
gft 1037
gft. 1037
gftxt 4174
gfv 1040
gfv. 1040
gfw 3624
gfw. 3624
gii 2584
gii. 2584
gik 2122
gik. 2122
gip 2094
gip. 2094
gks 2243
gks. 2243
glas 1949
glc 1043
glc. 1043
glev 4179
gli 2587
gli. 2587
global 366
gmd 831
gmd. 831
gmp 2439
gmp. 2439
gns 2098
gns. 2098
gos 2591
gos. 2591
gosub 361, 3754, 3756
GOTO 361
Goto End Tag 4165
Goto First Node 4200
Goto Root 4206
Goto Start Tag 4197
gpc 1801
gpc. 1801
gpi 2893
gpi. 2893
gpu 2898
gpu. 2898
GRA.con 1792
GRA.convert 1792
GRA.get 1784
GRA.getresolution 1784
GRA.sca 1787
GRA.scale 1787
GRA.vbm 1797
GRA.vtobmp 1797
Grab 1201
gradient 1953
grd 1953
gre 3128
gre. 3128
grey 1358
GRG. 1725
gri 778
gri. 778
grm 2337
grm. 2337
grr 3131
grr. 3131
grs 430
grs. 430
grw 3626
grw. 3626
gsb 2218
GSB. 2218
gsc 3134
gsc. 3134
gsd 2595
gsd. 2595
gsi 1046
gsi. 1046
gsp 4342
gsp. 4342
gss 3137
gss. 3137
gsw 3630
gsw. 3630
gto 2189
GTO. 2189
gud 2101
gud. 2101
gun 2103
gun. 2103

*© 2023 Theo Gottwald * https://www.smart-package.com*

gup 2024
gup. 2024
gus 2598
gus. 2598
gut 2028
gut. 2028
gvl 2105
gvl. 2105
GVT. 3477
gwp 3633
gwp. 3633
gxyb 1195
gxyp 1198

- H -

hdc 2664
HDC. 2664
HE 1140
Help and Manual 2733
hexadecimal 352
hflip 1395
Hide 2860
hit. 835
hit.. 835
htv 3636
htv. 3636
Human-Eye-Recorder 1140
hvflip 1398
hwi 3638
hwi. 3638

- I -

iai 2601
iai. 2601
ibr 595
ibr. 595
ICC. 1700
ICD. 2666
Icon 2931
ICW. 3640
ID 386
idc 709
idc. 709
idp 1126
idp. 1126
ief 1050
ief. 1050
iek 2246
iek. 2246
iem 2402
iem. 2402
iep 2900
iep. 2900
ier 2918
ier. 2918
iev. 3481
iew 3643
iew. 3643
IF 262
ifc 2340
ifc. 2340
ifd 2152
ifd. 2152
ifl 2155
ifl. 2155
IFP. 1873
ifv 2161
ifv. 2161
ifz 4306
ifz. 4306
igs 859
igs. 859
iik 2125
iik. 2125
iks 2249
iks. 2249
ilc. 1805
ili 2163
ili. 2163
image 1958
IML.aeq 4006
IML.afs 4010
IML.After EQ 4006
IML.afz 4016
IML.Any Flags Set 4010
IML.Any Flags Zero 4016
IML.Before EQ 4026
IML.beq 4026
IML.Between 4034
IML.Between DQ 4022
IML.Between SQ 4030
IML.bsq 4022, 4030
IML.btw 4034
IML.cbt 4038
IML.close bracket 4038
IML.cmt 4042
IML.Comment 4042
IML.eat 4047
IML.edn 4051
IML.End Attribute 4047
IML.End Name 4051

*© 2023 Theo Gottwald * https://www.smart-package.com*

IML.End Tag 4055
IML.etg 4055
IML.idq 4059
IML.Is DQ 4059
IML.Is SQ 4063
IML.isq 4063
IML.Name Char 3995
IML.Name Start 3998
IML.nsc 3995
IML.obt 4067
IML.Open Bracket 4067
IML.osd 4071
IML.Outside SDQ 4071
IML.Self Terminate 4083
IML.sta 4075
IML.Start Attribute 4075
IML.Start Name 4087
IML.Start Tag 4079
IML.stg 4079
IML.stm 4083
IML.stn 4087
IML.stt 4091
IML.System Tag 4091
IML.vsc 3998
IML.Whitespace Asc 4001
IML.whitespace character 4003
IML.wsa 4001
IML.wsc 4003
imo 2503
imo. 2503
inc pos 4213
include 248
Increment Position 4213
Indentation Checker 3247
Inline 3488
Input and 2008
Insert XML Part 4216
InterKOM 2116
Internet and Network 2066
invert 1442
ios 2607
ios. 2607
ipc 1816
ipc. 1816
IPD. 2669
IRA. 1824
irc 1828
irc. 1828
irs 434
irs. 434
Is DQ 4059
Is SQ 4063
isc 2552
isc. 2552
isi 2557
isi. 2557
isp 4349
isp. 4349
ito 2404
ito. 2404
itp. 1877
itw 3646
itw. 3646
iua 2612
iua. 2612
iva 1053
iva. 1053
ivb 3521
ivb. 3521
ivc 3524
ivc. 3524
IVN. 2343
ivp 3529
ivp. 3529
ivs 3531
ivs. 3531
ivv 3534
ivv. 3534
iwc 3537
iwc. 3537
iwe 3718
iwe. 3718
iwi 3721
iwi. 3721
iws 3724
iws. 3724
iwv 3727
iwv. 3727
iwz 3730
iwz. 3730
IXX-Commands 2146

- J -

jfp. 1881
jis 2171
jis. 2171
jit 2174
jit. 2174
jiv 2177
jiv. 2177
jiz 2180
jiz. 2180

*© 2023 Theo Gottwald * https://www.smart-package.com*

jle 2183
jle. 2183
jme 2186
jme. 2186
jmp 2189
jmp. 2189
jnf 2191
jnf. 2191
jnj 2194
jnj. 2194
jnp. 1881
jnr 2197
jnr. 2197
jns 2200
jns. 2200
jnt 2203
jnt. 2203
jnz 2206
jnz. 2206
jor 2209
jor. 2209
jrr 2212
jrr. 2212
jrs 2215
jrs. 2215
jsr 2218
jsr. 2218
JTN. 1885
JTP. 1885
JXX-Commands 2168

- K -

kbc. 2252
KBI. 2252
keycode 2235
keycode. 2235
kpn 2904
kpn. 2904
kyd 2256
kyd. 2256
kyu 2260
kyu. 2260

- L -

label 361
Language 386
Language-ID 386
lap. 599
lap.. 599
LASTCOLUMN 530
LASTROW 530
lbe 649
lbe. 649
lbo 2506
lbo. 2506
lbr 601
lbr. 601
ldi 1958
len 3140
len. 3140
LETTER_TO_NUMBER 532
lf 4113, 4121
lff 1056
lff. 1056
lfn 1058
lfn. 1058
lfp. 1890
LFT. 1060
lib 2614
lib. 2614
licence 238
lin 1962
line 1962
List 2235, 3743, 3745
load 1412, 1958, 4113, 4121
load file 4113, 4121
load image 1958
local 366
lof. 1063
Logic 1449
logoc 1449
Loops 2298
LShift 467
LTP. 1894
lvi 651
lvi. 651

- M -

macro 253, 361
Mail 789
MailConfig.dat 789
map 2483
map. 2483
mat 1967
mathematical 2314
matte 1967
maw 3733
maw. 3733
mbt 2031

*© 2023 Theo Gottwald * https://www.smart-package.com*

mbt. 2031
mbx 2036
mbx. 2036
mdc 2442
mdc. 2442
MDE 2487
MDE. 2487
mdf 1066
mdf. 1066
mdi 2445
mdi. 2445
mdm 2490
mdm. 2490
MISC-Operations 2361
mixtar 1162
mkd 1068
mkd. 1068
mlc 2448
mlc. 2448
mld 2451
mld. 2451
mle 2492
mle. 2492
mli 2454
mli. 2454
mlm 2457
mlm. 2457
mlu 2463
mlu. 2463
mmv 2466
mmv. 2466
mnd 2108
mnd. 2108
Modes 1156
mos 2509
mos. 2509
Mouse-Operations 2412
MOV. 1070
move 1082, 1479
MP3 2522
mpa 2523
mpa. 2523
mpb 2526
mpb. 2526
mpc 2530
mpc. 2530
MPO 2469
MPO. 2469
mps 2536
mps. 2536
mrc 2472
mrc. 2472
mre 2495
mre. 2495
mri 2476
mri. 2476
mrk 2907
mrk. 2907
MRM. 2460
msb 2040
msb. 2040
Multimedia 2522
mum 2498
mum. 2498
mwc 2514
mwc. 2514
mwe 2500
mwe. 2500
mwr 2480
mwr. 2480
mwt 2516
mwt. 2516
My End Tag 4220
My Start Tag 4223

- N -

nai 2601
nai. 2601
Name Char 3995
Name Start 3998
nav 606
nav. 606
nb 1971
nbi 1974
nbo 1977
nbr 595
nbr. 595
NCC. 1700
NCD. 2666
ndc 709
ndc. 709
ndp 1126
ndp. 1126
nef 1050
nef. 1050
nek 2246
nek. 2246
nem 2402
nem. 2402
nep 2900
nep. 2900
ner 2918

*© 2023 Theo Gottwald * https://www.smart-package.com*

ner. 2918
nev. 3481
new 1481, 3643
new. 3643
nex 2308
nex. 2308
Next Level 3750
nfc 2340
nfc. 2340
nfd 2152
nfd. 2152
nfl 2155
nfl. 2155
NFP. 1873
nfv 2161
nfv. 2161
nfz 4306
nfz. 4306
ngs 859
ngs. 859
nik 2125
nik. 2125
nks 2249
nks. 2249
nlc 1805
nlc. 1805
nli 2163
nli. 2163
NML.aeq 4006
NML.afs 4010
NML.After EQ 4006
NML.afz 4016
NML.Any Flags Set 4010
NML.Any Flags Zero 4016
NML.bdq 4022
NML.Before EQ 4026
NML.beq 4026
NML.Between 4034
NML.Between DQ 4022
NML.Between SQ 4030
NML.bsq 4030
NML.btw 4034
NML.cbt 4038
NML.close bracket 4038
NML.cmt 4042
NML.Comment 4042
NML.eat 4047
NML.edn 4051
NML.End Attribute 4047
NML.End Name 4051
NML.End Tag 4055
NML.etg 4055
NML.idq 4059
NML.Is DQ 4059
NML.Is SQ 4063
NML.isq 4063
NML.Name Char 3995
NML.Name Start 3998
NML.nsc 3995, 3998
NML.obt 4067
NML.Open Bracket 4067
NML.osd 4071
NML.Outside SDQ 4071
NML.Self Terminate 4083
NML.sta 4075
NML.Start Attribute 4075
NML.Start Name 4087
NML.Start Tag 4079
NML.stg 4079
NML.stm 4083
NML.stn 4087
NML.stt 4091
NML.System Tag 4091
NML.Whitespace Asc 4001
NML.whitespace character 4003
NML.wsa 4001
NML.wsc 4003
nmo 2503
nmo. 2503
no 1971, 1974, 1977
no bend 1971
no bend-in 1974
no bend-out 1977
no error 3758
no transparnecy 1980
non-client 156
nos 2607
nos. 2607
NOT 470
not found 3754
notext 341
npc 1816
npc. 1816
NPD. 2669
nra 1824
nra. 1824
nrc 1828
nrc. 1828
nrs 434
nrs. 434
nsi 2557
nsi. 2557
nsp 4349
nsp. 4349

© 2023 Theo Gottwald * https://www.smart-package.com

nto 2404
nto. 2404
ntp. 1877
ntr 1980
ntw 3646
ntw. 3646
nua 2612
nua. 2612
number_to_letter 534
nva 1053
nva. 1053
nvb 3521
nvb. 3521
nvc 3524
nvc. 3524
nvp 3529
nvp. 3529
nvs 3531
nvs. 3531
nvv 3534
nvv. 3534
nwc 3537
nwc. 3537
nwe 3718
nwe. 3718
nwi 3721
nwi. 3721
nws 3724
nws. 3724
nwv 3727
nwv. 3727
nwz 3730
nwz. 3730

- O -

octal 352
OEND 251
OFC.excel 508
on 364
on_error 364
ONCE 251
only 3750
oop 2312
oop. 2312
open 536
Open Bracket 4067
opt 2406
opt. 2406
OR 473
OS 2570
OUT. 2043
Output 2008
Outside SDQ 4071

- P -

paf 1508
pafh 1511
PAGE_UPDOWN 538
par 4117
Parallel Robot Operations 421, 2632
PARAMS 262
parse 4117
Parse from Var 4117
Parser 3973, 3981
pattern 341
patternfill 1508
patternfillhwn 1511
pau 783
pau. 783
pdc. 2658
PDC.cde 2677
PDC.cdn 2674
PDC.cld 2671
PDC.Close Desktop 2671
PDC.Create Desktop no Explorer 2674
pdc.Create Desktop no Explorer 2674
pdc.Create Desktop with Explorer 2677
pdc.gd 2682
PDC.gdn 2679
PDC.Get current Desktop Name 2679
PDC.goto default 2682
PDC.ruda 2684
PDC.rudb 2684
PDC.Run on Desktop WFI 2684
PDC.Run on Desktop WFO 2684
PDC.Set User 2688
pdc.su 2688
PDC.swd 2691
PDC.Switch to Desktop 2691
pid 137, 2881
pil 2702
pil. 2702
pis 2749
pis. 2749
PIS.ButtonEnable 2763
PIS.GetHandle 2766
PIS.GetMouseButton 2777
PIS.GetMouseButtonHandle 2787
PIS.MouseMessageFilter 2797
PIS.sbt 2818

*© 2023 Theo Gottwald * https://www.smart-package.com*

PIS.SetAutoHide 2800
PIS.SetAutoRunSynchron 2802
PIS.SetAutoRunSynchronNoDisable 2804
PIS.SetButtonBackColor 2807
PIS.SetButtonForeColor 2807
PIS.SetButtonNumber 2810
PIS.SetButtonsHDist 2812
PIS.SetButtonsHVSize 2815
PIS.SetButtonsVDist 2812
PIS.SetcorrectionFactor 2823
PIS.SetPicture 2830
PIS.SetTopMost 2836
PIS.SetTransparent 2847
PIS.SetWindowAlert 2850
PIS.Show 2860
piw 2706
piw. 2706
pixel 1088
Play 2522
plot 1995
Plugin 2693
Plugin01 2693
PNS. 3022
Point-Convert-Lib 412
pop 3007
pop. 3007
Position to Start or End Tag 4229
pov 3010
pov. 3010
Powershell 2866
pre 2638
pre. 2638
pri 1984
Print 1984, 4226
printat 1516
process 2881
prr 2642
prr. 2642
prs 2647
prs. 2647
prt 2045
prt. 2045
prv 2651
prv. 2651
pss. 3026
psw. 2874
pus 3013
pus. 3013
puv 3019
puv. 3019
PVS. - Push Variables on Stack/Que 3015
pwe. 2877
PWS. 2874

- Q -

QFR. 1130

- R -

RAU 271
rbo 2519
rbo. 2519
RCC. 1706
rdi 2921
rdi. 2921
rdm 2112
rdm. 2112
rdt 2923
rdt. 2923
rect 356
rectangle 356
RECURS 262
registry 2915
remove 3743
ren 1074
ren. 1074
rename 1082
Reparse 4232
Repeat Prototype 3748
rer 2925
rer. 2925
RES. 2936
RES.Change Icon 2938
RES.Change Manifest 2941
RES.Change Version 2944
RES.CM 2941
RES.EA 2950
RES.EI 2955
RES.EM 2958
RES.EV 2964
RES.Extract All 2950
RES.Extract Icon 2955
res.Extract Manifest 2958
RES.Extract Version 2964
RES.Icons Path 2978
RES.ip 2978
RES.Last Icon 2978
res.Last Manifest 2969
res.Last Version 2973
RES.LI 2978
RES.LM 2969
RES.LV 2973

© 2023 Theo Gottwald * https://www.smart-package.com

res.re 2948
RES.Refresh Explorer 2948
reset 1987
reset font 1987
resizeto 1204
Resourcen 2931
Resources 2931
ret 2224
ret. 2224
rew 2928
rew. 2928
rgb_to_hls 1745
RIC. 1719
rmd 1077
rmd. 1077
rnd 2348
rnd. 2348
RotateLeft 476
RotateRight 479
rpl 3143
rpl. 3143
rra 1836
rra. 1836
rrc 2227
rrc. 2227
rrs 2230
rrs. 2230
rsf 1987, 1990
RShift 482
rst 2264
rst. 2264
rts 2233
rts. 2233
Run as user 271
run without error 3758

- S -

s1 349
S10 349
S11 349
S12 349
s13 349
S14 349
S15 349
S16 349
S17 349
S18 349
S19 349
S2 349
S20 349
S21 349
S22 349
S23 349
s24 349
s25 349
s26 349
s27 349
s28 349
s29 349
S3 349
S30 349
S4 349
S5 349
S6 349
S7 349
s8 349
s9 349
SAC\ 3752
Sample XML 3966
sao 625
sao. 625
SAV. 3539
SAV.r 3539
sav.restore 3539
SAV.Save 3539
Save 540, 1415
SAVEAS 542
saw 3736
saw. 3736
say 2048
say. 2048
sbd 3070
sbd. 3070
sbe 682
sbe. 682
sbm 3073
sbm. 3073
SBT. 2055
sci 657
sci. 657
scintilla 238
SCROLL_LEFTRIGHT 545
SCROLL_UPDOWN 547
scs 2983
scs. 2983
scw 3738
scw. 3738
SDA. 2267
SDH. 2271
SDK. 2275
sdt 2278
sdt. 2278

© 2023 Theo Gottwald * https://www.smart-package.com

SDU. 2267
Search 341, 3750, 3752
Search All Down 4236
Search All Up 4239
Search Down Tag Name 4247
Search Inside Down 4250
Search Inside Up 4254
Search Name and Attribute 4258
Search Name and Attribute up 4262
Search Pattern Down 4270
Search Pattern Up 4276
Search Up Tag Name 4290
sec 2129
sec. 2129
sek 2282
sek. 2282
Self Terminate 4083
sendkey 549
ser 2559
ser. 2559
service 2544
set 1990
Set Asc 4280
Set Chr 4242
set font 1990
set pixel 1995
Set Position 4273
Set XML Part 4293
SetAutoRunASynchron 2802
SetAutoRunASynchronNoDisable 2804
SetHorizontal 2833
SetVertical 2833
SFP. 1899
sfw 3812
sfw. 3812
sgs 861
sgs. 861
sha 840
sha. 840
shf 1082
shf. 1082
SHFshf 1082
Show 488, 1387, 2860
sht. 2061
sib 685
sib. 685
sid 2114
sid. 2114
sik 2131
sik. 2131
SKI. 2285
skp 2289
skp. 2289
skt 2293
skt. 2293
sma 797
sma. 797
Smarty 763, 940, 2008, 2146, 4304
Smarty' 867
Smarty Script language 393
Smartys 812
Smarty's Compiler 267
SmartysFEX-Commands 920
smb 799
smb. 799
smh 660
smh. 660
smp 803
smp. 803
snap 1418
Software 2570
spc 1843
spc. 1843
specialfolders 345
spi 1995
Split Attrib at 4267
SplitRGB 1780
SRG. 1725
srt 3147
srt. 3147
srv 2134
srv. 2134
ssp. 4352
ssp.. 4352
Stack Commands 3005
Start Attribute 4075
Start Name 4087
Start Tag 4079
Start Tag Down 4285
Start Tag Up 4287
stf 2410
stf. 2410
stj 3030
stj. 3030
Store 491
str 3150
str. 3150
str.chars a 3155
str.chars b 3159
str.chars c 3163
str.chars d 3167
str.clone 3171
str.clone. 3171
str.delete 3174

*© 2023 Theo Gottwald * https://www.smart-package.com*

str.delete. 3174
str.find 3176
str.find. 3176
str.from base64 url 3182
str.from base64 url. 3182
str.from hexasci 3185
str.from hexasci. 3185
str.generate 3188
str.generate. 3188
str.insert 3192
str.insert. 3192
str.make pad 3195
str.mime to text 3198
str.mime to text. 3198
str.overwrite 3205
str.overwrite. 3205
str.patch 3208
str.patch. 3208
str.reverse 3212
str.reverse. 3212
str.text to mime 3215
str.text to mime. 3215
str.to base64 url 3219
str.to base64 url. 3219
str.to hexasci 3222
str.to hexasci. 3222
String Operations 3042
sts 3038
sts. 3038
stv 3040
stv. 3040
stw 3814
stw. 3814
svl 2539
svl. 2539
swap 1484
swp 3884
swp. 3884
sws 3889
sws. 3889
system 334
System Tag 4091
systemcolors 349
system-colors 349
systemvariable 334
SYT. 2618

- T -

tdl 2085
Test Download 2085
test with password 2088
Testing_Macros 415
Text-File Operations 3252
There 334
time 688
ToBin 494
ToClipboard 1421
Toggle 498
tolerant 341
ToVal 501
tovar 1424
tpw 2088
tra 2001
transparency 2001
tricolor 1377
ttv 751
ttv. 751
tvf 665
tvf. 665
tvi 668
tvi. 668
twm 2136
twm. 2136
TXT.Clear 3416
TXT.clr 3416
TXT.Convert_to_ANSI 3258
TXT.convert_to_unicode 3261
TXT.cta 3258
TXT.ctu 3261
TXT.Find_in_Text 3334
TXT.Find_Line_with 3331
TXT.fit 3334
TXT.flw 3331
TXT.fov 3449
TXT.From_Var 3449
TXT.gal 3351
TXT.gbl 3353
TXT.gel 3355
TXT.get_after_line 3351
TXT.get_before_line 3353
TXT.Get_Filename 3470
TXT.Get_from_LOF 3452
TXT.Get_Lenght 3472
TXT.Get_Line 3355
TXT.Get_Line_Count 3357
TXT.get_line_lefttrim 3359
TXT.Get_Line_Position 3361
TXT.Get_Line_trimmed 3363
TXT.Get_pure_Text 3455
TXT.get_to_end 3369
TXT.Get_To_Start 3371
TXT.gfl 3452

*© 2023 Theo Gottwald * https://www.smart-package.com*

TXT.gfn 3470
TXT.glc 3357
TXT.gll 3359
TXT.glp 3361
TXT.glt 3363
TXT.gpt 3455
TXT.gte 3369
TXT.gtl 3472
TXT.gts 3371
TXT.Line_Start_End_by_Position 3373
TXT.Linenumber_from_Position 3376
TXT.lnp 3376
TXT.Load Unicode_File 3420
TXT.Load_pure_Text 3457
TXT.Load_Textfile 3418
TXT.lpt 3457
TXT.lsp 3373
TXT.ltf 3418
TXT.luf 3420
TXT.rep 3338
TXT.replace 3338
TXT.Replace_in_Line 3341
TXT.replace_in_text 3344
TXT.ril 3341
TXT.rit 3344
TXT.sau 3422
TXT.sav 3424
TXT.save 3424
TXT.save_as_unicode 3422
TXT.Save_File 3424
TXT.Set_Line 3411
TXT.Set_Line_to 3411
TXT.slt 3411
TXT.To_LOF 3459
TXT.To_Var 3462
TXT.To_Var_and_Clear 3464
TXT.toc 3464
TXT.tol 3459
TXT.tov 3462
TXT.tra 3275
TXT.translate_chars 3275

- U -

ucv 1914
ucv. 1914
UCV.bgr 1921
UCV.box 1925
UCV.cov 1929
UCV.dbw 1933
UCV.dc 1937
UCV.draw to 1962
UCV.drawto 1962
UCV.eli 1941
UCV.fgr 1945
UCV.font set 1990
UCV.glas 1949
UCV.grd 1953
UCV.ldi 1958
UCV.line 1962
UCV.mat 1967
UCV.matte 1967
UCV.nb 1971
UCV.nbi 1974
UCV.nbo 1977
UCV.no blend 1971
UCV.no blend-in 1974
UCV.no blend-out 1977
UCV.no transparnecy 1980
UCV.ntr 1980
UCV.plot 1995
UCV.pri 1984
UCV.print 1984
UCV.reset font 1987
UCV.rsf 1987, 1990
UCV.set font 1990
UCV.set pixel 1995
UCV.spi 1995
UCV.tra 2001
UCV.transparency 2001
UCV.wri 2005
UCV.write 2005
Unclean 4098
UnClean Text 4098
UNI. 2364
unwrap 4123
Up Close Bracket 4298
Up Open Bracket 4301
url 3234
url. 3234
User-Macros 399
usm 786
usm. 786
UTA. 3239

- V -

vab 3544
vab. 3544
vac 3546
vac. 3546
vaf 3548

*© 2023 Theo Gottwald * https://www.smart-package.com*

vaf. 3548
VAI. 3550
van 3553
van. 3553
vao 3556
vao. 3556
vap 3560
vap. 3560
var 3562
var. 3562
variable 334, 361, 3569
Variable Commands 3474
variables 366
vas 3564
vas. 3564
vav 3566
vav. 3566
vba 3569
vba. 3569
vbe. 3578
vbs. 3581
VB-Script Support 3572
vbt 3225
vbt. 3225
VDA. 3516
vdc 2350
vdc. 2350
VDE. 3516
vdf 739
vdf. 739
VErsion 2931
vflip 1401
vic 2352
vic. 2352
vlt 3228
vlt. 3228
vrd 3076
vrd. 3076
vrt 3231
vrt. 3231
vsl 3113
vsl. 3113
vsm 3116
vsm. 3116
vsr 3119
vsr. 3119
vtb 2354
vtb. 2354
vtd 756
vtd. 756
vtf 742
vtf. 742
vth 2356
vth. 2356
vtl 3053
vtl. 3053
vtm 3056
vtm. 3056
vtn 2358
vtn. 2358
vtt 760
vtt. 760
vtu 3059
vtu. 3059

- W -

wait 1085
WAV 2522
wbr 630
wbr. 630
WCW. 3893
wff 1085
wff. 1085
wfm 2140
wfm. 2140
WFP. 1903
wfv 2655
wfv. 2655
Whitespace Asc 4001
wii 2909
wii. 2909
wik 2142
wik. 2142
wildc 341
windows operation 1911, 3584
wlc 1846
wlc. 1846
wnk 2296
wnk. 2296
wpr 2911
wpr. 2911
wpt 2913
wpt. 2913
wra 1851
wra. 1851
wrap 4101
wrc 1856
wrc. 1856
wri 2005
write 552, 2005
wrs 436
wrs. 436

© 2023 Theo Gottwald * https://www.smart-package.com

wsc 2566
wsc. 2566
wsp 4354
wsp. 4354
WTP. 1907
WTW. 3896

- X -

xml 3990
xml. 3990
XML.abm 4128
XML.abn 4131
XML.abv 4135
XML.acnt 4139
XML.Attrib by Name 4131
XML.Attrib by Number 4128
XML.Attrib Value by Name 4142
XML.avbn 4142
XML.cle 4106
XML.Clean 4108
XML.Clean Text 4108
XML.Clear 4095
XML.clear error 4106
XML.clr 4095
XML.dec pos 4146
XML.dncb 4152
XML.dnob 4155
XML.Down Close Bracket 4152
XML.Down Open Bracket 4155
XML.Dump 4149
XML.Flags to Text 4158
XML.ftxt 4158
XML.Get Asc 4191
XML.Get Attrib by Value 4135
XML.Get Attrib Count 4139
XML.Get Block 4161
XML.Get Error 4111
XML.Get First Node 4171
XML.Get Flags 4168
XML.Get Flags as Text 4174
XML.Get Len 4177
XML.Get Level 4179
XML.Get Position 4182
XML.Get Position for Previous 4185
XML.Get Root 4188
XML.Get Start End Tags 4194
XML.Get this Name 4203
XML.Get XML Part 4209
XML.geterr 4111
XML.gfl 4168
XML.gfn 4171
XML.gftxt 4174
XML.gle 4177
XML.glev 4179
XML.Goto First Node 4200
XML.Goto Root 4206
XML.Goto Start Tag 4197
XML.gpo 4182
XML.gpr 4185
XML.gro 4188
XML.gsc 4191
XML.gse 4194
XML.gst 4197
XML.gtfn 4200
XML.gtn 4203
XML.gtr 4206
XML.gxp 4209
XML.inc pos 4213
XML.Increment Position 4213
XML.Insert XML Part 4216
XML.ixp 4216
XML.lf 4113
XML.Load 4113
XML.Load File 4113
XML.met 4220
XML.mst 4223
XML.My End Tag 4220
XML.My Start Tag 4223
XML.par 4117
XML.Parse 4117
XML.Parse from Var 4117
XML.Position to Start or End Tag 4229
XML.Print 4226
XML.prt 4226
XML.ptse 4229
XML.Reparse 4232
XML.rpa 4232
XML.sad 4236
XML.sau 4239
XML.scr 4242
XML.sdn 4247
XML.Search All Down 4236
XML.Search All Up 4239
XML.Search Down Tag Name 4247
XML.Search Inside Down 4250
XML.Search Inside Up 4254
XML.Search Name and Attribute 4258
XML.Search Name and Attribute up 4262
XML.Search Pattern Down 4270
XML.Search Pattern Up 4276
XML.Search Up Tag Name 4290
XML.Set Asc 4280

*© 2023 Theo Gottwald * https://www.smart-package.com*

XML.Set Chr 4242
XML.Set Position 4273
XML.Set XML Part 4293
XML.sid 4250
XML.siu 4254
XML.sna 4258
XML.snaup 4262
XML.spa 4267
XML.spdn 4270
XML.spl 4121
XML.Split Attrib 4121
XML.Split Attrib at 4267
XML.spo 4273
XML.spup 4276
XML.ssc 4280
XML.Start Tag Down 4285
XML.Start Tag Up 4287
XML.std 4285
XML.stu 4287
XML.sun 4290
XML.sxp 4293
XML.UnClean 4098
XML.UnClean Text 4098
XML.UnWrap 4123
XML.Up Close Bracket 4298
XML.Up Open Bracket 4301
XML.upcb 4298
XML.upob 4301
XML.Wrap 4101
XML:Decrement Position 4146
XML:gbl 4161
XML:GEnd 4165
XML:Goto End Tag 4165
XML-Commands 3973, 3981
XML-Parser 3973, 3981
xor 341, 505, 843
xor. 843

- Z -

zip 4309
zip. 4309
ZIP.cf 4313
ZIP.compress 4318
ZIP.compress file 4313
ZIP.compress var 4318
ZIP.cv 4318
ZIP.get 4321
ZIP.get zip 4321
ZIP.get zip filename 4321
ZIP.sdc 4323
ZIP.set 4323
ZIP.set default 4323
ZIP.set default compression 4323
ZIP.uf 4326
ZIP.uncompress 4329
ZIP.uncompress var 4329
ZIP.unzip 4326
ZIP.unzip file 4326
ZIP.uv 4329
ZIP.zf 4332
ZIP.zip 4332
ZIP.zip folder 4332

*© 2023 Theo Gottwald * https://www.smart-package.com*