Automise Manual

Table of Contents

0

Part I Automise 1

1 Automise Overview 1

2 Version History 1

3 Automise License 1

Part II Getting Started 5

1 Automise Basic Concepts 5

2 Automise IDE Modes 6

3 Upgrading Projects From Previous Versions 6

Part III Automise IDE Reference 8

1 Welcome Page 8

2 IDE Design Mode 9

Working with Actions 10
Properties tab 12
Action Lists 13
Actions Tab 14
Project Tree 16
Run Tab 18
Quick Help 20
Script Editor 20
Keyboard Shortcuts 21
Search & Replace 22
Error Handling 23

3 IDE Debug Mode 24

Watches 25

4 History & Statistics 25

5 IDE Running Mode 26

Running a Project 27
Run Log 28
Validation 31

6 IDE Features 32

Options Dialog 32
Design Time Options 33
Runtime Options 35
Logging Options 36
Validation Options 39
Welcome Page Options 40
Automatic Action Usage Updates 40
Check for Updates 42
Messages 42
Package Manager 42
Tray Icon 43
Keyboard Shortcuts 45

7 Automise Projects 46

Project Files & Other Files 46

Project Global Script 47
Project Edit Password 49
Project Summary 50
Project Information 51

Part IV Scripting 52

1 Scripting in Automise 52
2 Script Editor Options 53
3 Script Debugging Options 55
4 Global Script functions 56
Format DateTime Formatting Options 58
MessageBox Constants 59
5 Action Script Events 60
Action Properties and Methods 61
Execute Condition 63
6 Accessing the Options settings via scripting 64
7 Accessing TStrings based properties 65

Part V Variables 66

1 Variables Overview 66
2 Adding Variables 68
3 Using Variables 71
4 Variable Evaluation and Recursion 72
5 Escaping Variable References 72
6 Variable Types 72
Project Variables 72
User Variables 73
Environment Variables 73
Application Variables 73
7 Action List Parameters 74
8 Variable Options 75
9 Variable Format Strings 78
10 DateTime Format Strings 80

Part VI Actions Reference 81

1 Custom Actions 83
2 Common Action Properties 84
Timing Properties 87
Logging Properties 88
Action Output Monitors 89
Run Action as User 92
Setting Processor affinity & process priority 94
Error Dialog Monitor 95
3 Edit Field 96
4 Archiving 97
7Zip 97
Create Archive 97
Delete from Archive 101
Extract Archive 102
List Archive 103

Test Archive 104
Update Archive 105
Create Zip File 106
Extract Zip File Action 110
SecureZIP 111
SecureZIP Add Files To Archive 112
SecureZIP Extract Archive 116
SecureZIP List Archive Contents 117
SecureZIP Test Archive Integrity 119
SecureZIP Common Pages 119
SecureZIP Security Page 120
SecureZIP Filters Page 121
SecureZIP Exclude List Page 123
WinRAR Action 123
5 CD/DVD Burner Actions 126
Burn CD/DVD Action 127
Burn ISO Action 130
Check Ready Action 130
Create ISO Action 131
Erase CD/DVD RW 132
6 Database 133
SQL Server Actions 133
DTSRun Action 133
Execute SQL Action 134
SQL Server Backup Database 137
SQL Server Best Practices Analyzer 138
SQL Server Check Catalogue 140
SQL Server Check Database 141
SQL Server Options 141
SQL Server Rebuild Indexes 142
SQL Server Remove Unused Space 143
SQL Server Restore Database 144
SQL Server Set Access Mode 146
SQL Server Update DB Statistics 147
ADO Dataset Iterator 147
ADO Execute SQL 149
ADO Execute Stored Procedure 150
7 Automise IDE 152
Save Automise Project Action 152
Export Log Action 153
Set Estimated Progress Total Action 155
Set Run Status Message Action 156
8 FileSets 158
Accessing FileSets 159
FileSet Define 160
FileSet Patterns 162
FileSet Filters 164
FileSet Copy/Move Action 165
FileSet Delete Files 167
FileSet Log Files 167
FileSet Refresh 168
FileSet to Variable 169
9 Files & Directories 170
Calculate File CRC32 170
Calculate File MD5 171
Check if File Exists Action 172

Concatenate Files Action 174
Copy File(s) Action 174
Copy/Move File List 176
Create Directory Action 178
Create Text File 179
Delete Directory Action 180
Delete File(s) Action 182
Extract Version Info 183
File Comparison Tools 184
Araxis Merge Compare Actions 184
Beyond Compare 186
Beyond Compare Action 186
Run Beyond Compare Script 187
File Compare Action 188
File Dependency Action 189
Find File(s) 192
Get File Size 194
Get File/Folder Date 195
Move Directory Action 197
Move File(s) Action 198
Read Text File 200
Rename File or Directory Action 201
RoboCopy 202
Robocopy 203
Robocopy Mirror 204
Robocopy Move 204
Run Robocopy Job 205
Robocopy Common Pages 208
Set File Attributes Action 215
Text Replace Action 216
Touch File(s) 217
Write Text File 219
XCopy 220
10 Flow Control 225
Action Group 225
Case Action 227
Delay Action 227
Else Action 228
Exit Action List Action 229
For Loop Action 229
If .. Then Action 231
If Prev Action Failed Action 233
Include Automise Project 234
Raise Exception 235
Run Action List Action 236
Running actions in Parallel 237
Stop Run Action 240
Switch Action 241
Try/Catch/Finally/End Actions 241
Waiting For Events (Wait For actions) 244
Wait For Command 246
Wait For File 247
Wait For Ini File 249
Wait For Process 251
Wait For Registry 252
Wait For Remote Computer 254
Wait For Script Result 256
Wait For Variable 258

Wait Until ... 259
While Loop Action ... 261
11 GUI Automation ... 262
Overview ... 262
Target Property Page ... 263
Automatic Configuration ... 263
Manual Configuration ... 264
Actions ... 268
Perform Mouse Click ... 268
Perform Mouse Move ... 270
Send Keyboard Input ... 271
Set Scrollbar Position ... 274
Wait For Window or Control ... 275
12 Ini Files & Registry ... 279
Export Registry Key ... 279
Read Ini File ... 279
Read/Set/Delete Registry Value ... 280
Write Ini File ... 284
13 Installers ... 285
Windows Installer ... 285
MSI Generic ... 285
MSI Merge ... 286
MSI Transform ... 287
MSI Update Properties ... 288
MSI Validation ... 289
Windows Installer - Install ... 290
Windows Installer - Reinstall ... 290
Windows Installer - Uninstall ... 291
14 Interactive ... 292
Ask Question Action ... 292
Beep Action ... 294
Choose One Action ... 294
Input Box Action ... 296
Message Box Action ... 297
Multi Question ... 298
Prompt for File or Directory ... 300
Prompt for Password Action ... 301
Prompt for Variables Action ... 303
Enhanced Prompt for Variables Action ... 305
15 Internet ... 309
Convert HTML to MHT File ... 309
FTP Actions ... 310
FTP Mirror ... 326
FTP Upload ... 333
FTP Client (deprecated) ... 338
HTTP Get Action ... 342
NNTP News Post Action ... 343
Send Email (SMTP) ... 346
Send ICQ Message Action ... 352
Send MSN Message Action ... 353
Telnet Client Action ... 356
Web Service Action ... 357
Web Service Iterator Action ... 359
WGet Download ... 361
SSH ... 363
SSH Open Connection ... 363

SSH Close Connection 364
SSH Execute Commands 365
SSH Batch Execute 366
SFTP 367
SFTP Connect 368
SFTP Disconnect 370
SFTP Check if Connected 370
SFTP Create Directory 371
SFTP Upload FileSet 371
SFTP Upload File 372
SFTP Upload Files 373
SFTP Dow nload File 375
SFTP Dow nload Files 376
SFTP Remove File 377
SFTP Remove Files 378
SFTP Remove Directory 379
SFTP Rename File 379
SFTP List Files 380
SFTP Query Available Space 382
SFTP Query Home Directory 383
SFTP Copy Remote File 384
FTPS Actions 385
FTPS Change Directory 386
FTPS Change Up Directory 387
FTPS Check If Connected 387
FTPS Connect 388
FTPS Create Directory 392
FTPS Delete File 393
FTPS Disconnect 394
FTPS Dow nload File 395
FTPS Dow nload Files 396
FTPS Get File Date 398
FTPS Get File Size 399
FTPS Get Remote Checksum 400
FTPS Get Remote Directory 401
FTPS List Directory 401
FTPS No-op 404
FTPS Remove Directory 405
FTPS Rename File 405
FTPS Send Command 406
FTPS Upload Files 408
EC2 Actions 409
EC2 Get Instance Properties 409
EC2 Change Instance States 410
EC2 Run Instances 411
EC2 Terminate Instance 413
EC2 Get Image Properties 414
EC2 Create Image 416
EC2 Deregister Image 417
EC2 Get Volume Properties 418
EC2 Attach Volume 420
EC2 Detach Volume 421
EC2 Create Volume 422
EC2 Delete Volume 423
EC2 Get Snapshot Properties 424
EC2 Create Snapshot 425
EC2 Delete Snapshot 426
EC2 Get Availability Zones 427

16 Internet Information Services (IIS) 429

IIS 5 429

Server Selection 429
Website Selection 431
Backup IIS Metabase Action 432
Create Virtual Directory Action 433
Create Web Site Action 436
Delete Virtual Directory Action 438
Delete Web Site Action 438
Restore IIS Metabase Action 439
Start / Stop / Pause Website Actions 440
Start / Stop / Restart WWW Service Actions 443
Unload IIS Application Action 446

IIS 6 447

Server Selection 448
Website Selection 449
Backup IIS Metabase Action 450
Create Application Pool 452
Create Virtual Directory Action 453
Create Web Site Action 455
Delete Application Pool 457
Delete Virtual Directory Action 459
Delete Web Site Action 459
Restore IIS Metabase Action 460
Start / Stop / Pause Website Actions 461
Start / Stop / Restart WWW Service Action 464
Unload IIS Application Action 467

IIS 6 FTP 468

Server Selection 468
FTP Site Selection 470
Create FTP Directory Action 471
Create FTP Site Action 474
Delete FTP Directory Action 478
Delete FTP Site Action 478
Start / Stop / Pause FTP Site Actions 479
Start / Stop / Restart FTP Service Actions 482

IIS 7 485

Server Selection 486
Website Selection 488
Create Website Action 489
Delete Website Action 490
Start / Stop / Restart Website 491
Create Application 492
Delete Application 493
Create Virtual Directory 494
Delete Virtual Directory 495
Create Application Pool 496
Delete Application Pool 497
Start / Stop / Recycle Application Pool 498

17 Iterators 499

File Contents Iterator 501
File/Fileset Iterator 502
Folder Iterator 503
INI File Iterator 504
List Iterator 505

18 Microsoft Message Queuing (MSMQ) 507

MSMQ Clear Queue 508

MSMQ Create Queue ... 509
MSMQ Delete Queue ... 510
MSMQ Peek Message ... 511
MSMQ Receive Message ... 512
MSMQ Send Message ... 513
MSMQ Wait for Message ... 514
19 Misc Actions ... 515
Comment Action ... 515
Generate Random Number ... 516
Get DateTime Action ... 517
Image File Manipulation ... 519
Image Manipulation ... 519
Lossless JPEG Transform ... 523
Mutex/Lock File Action ... 526
Run Script Action ... 527
Sort Text List Action ... 528
Spell Checking ... 529
Text Find / Replace Action ... 532
Simple Maths Action ... 537
20 .NET Actions ... 537
.Net Framework Tools ... 537
Precompile ASP.NET 2.0 Application ... 538
Register Assembly in COM [REGASM] ... 541
Run ASPNET_REGIIS.EXE ... 541
Serviced Components Installation Tool [REGSVCS] ... 542
.Net SDK Tools ... 544
GAC Download Cache [GACUTIL] ... 544
GAC Install [GACUTIL] ... 545
GAC Uninstall [GACUTIL] ... 546
.NET Framework Options ... 547
21 Network ... 548
Check If Host Exists ... 548
22 PDF Files ... 550
Convert PDF ... 550
Convert to PDF ... 550
PDF Insert Text ... 551
PDF Merge Files ... 553
23 Stack and Queue Actions ... 553
Stack/Queue Define ... 554
Stack/Queue Clear ... 555
Stack/Queue Get Count ... 555
Stack/Queue Insert Item ... 556
Stack/Queue Is Empty ... 557
Stack/Queue Iterator ... 558
Stack/Queue Log Items ... 560
Stack/Queue Peek Item ... 561
Stack/Queue Pop Item ... 562
Stack/Queue Push Item ... 563
24 String Manipulation ... 564
CSV Field Iterator ... 564
Path Manipulation ... 565
String Add Breaks ... 567
String Casing ... 569
String Concatenation ... 570
String Encryption ... 571
String Padding ... 573

String Quoting ... 574
String Pos ... 575
String Replace ... 576
String Reverse ... 578
String SubString ... 579
String Trimming ... 580
25 Variables ... 581
Append to Variable Action ... 581
Define Variable Action ... 582
If Variable Defined Action ... 584
Load Variables From Ini ... 585
Log Variable Values ... 588
Replace Variables ... 589
Reset Variable to Default ... 590
Save Variables To Ini ... 591
Set Variable Action ... 593
26 Virtualization Systems ... 595
Microsoft Hyper V ... 595
Hyper V Server Page ... 596
Hyper V Check VM State ... 598
Hyper V Start VM ... 598
Hyper V Stop VM ... 599
Hyper V Reset VM ... 600
Hyper V Pause VM ... 601
Hyper V Suspend VM ... 602
Hyper V Create VM Snapshot ... 603
Hyper V Apply VM Snapshot ... 604
Hyper V Delete VM Snapshot ... 605
Hyper V List Virtual Machines ... 606
Microsoft Virtual Server ... 607
Virtual Server Check VM Status ... 608
Virtual Server List VMs ... 610
Virtual Server Pause VM ... 611
Virtual Server Reset VM ... 612
Virtual Server Save VM State ... 613
Virtual Server Start VM ... 614
Virtual Server Turn Off VM ... 615
VMWare Server ... 616
VMWare Server List VMs ... 617
VMWare Server Start / Stop / Reset / Suspend VM ... 618
VMWare Workstation ... 620
Check Virtual Machine Status ... 620
Create Snapshot ... 621
Reset Virtual Machine ... 622
Revert To Snapshot ... 622
Start Virtual Machine ... 623
Stop Virtual Machine ... 624
Suspend Virtual Machine ... 625
VMWare ... 626
List Virtual Machine ... 628
Change Virtual Machine State ... 629
Snapshot ... 630
Execute Guest Command ... 631
27 Windows OS ... 633
Activate Windows ... 633
Active Directory Administration ... 634
Active Directory Add User(s) To Group ... 634

Active Directory Create Group 636
Active Directory Create Object 637
Active Directory Create User 639
Active Directory Edit User Properties 640
Active Directory Enumerate Group 643
Active Directory Get Info 644
Active Directory Move Object(s) 647
Active Directory Remove User(s) From Group 648
Active Directory Rename 650
Close Process Action 651
COM+ Administration 653
COM+ Delete 653
COM+ Edit Application 654
COM+ Register 658
COM+ Shutdown Application 660
COM+ Start Application 661
Control Service Action 662
Create Shortcut 663
Event Log 665
Event Log - Backup 666
Event Log - Clear 667
Event Log - Search 668
Execute Powershell Script 670
Execute Program Action 672
File & Folder Security 675
Change File Permissions 675
Change Folder Permissions 676
Change Ownership 678
Generate New GUID 679
If COM Class Registered 680
Logical and Network Drives 682
Backup 682
Chkdsk Action 684
Defrag Disk 686
Empty Recycle Bin 687
Format Drive 688
Get Available Drives 689
Get Disk Free Space 689
Get Logical Disk Information 690
Is Recycle Bin Empty 692
Map Network Drive Action 693
Subst Drive Action 694
Unmap Network Drive Action 695
Net Send Message Action 696
Networking Configuration and Diagnostics 697
Configure IP Security 697
Configure Network 698
Find Network Adapters 699
Name Server Lookup 701
Rasdial VPN / Remote Network Connection 702
Trace Route 704
NTFS Junction Points 704
Clean Old Mount Point Entries 705
Disable Automatic Mounting 705
Enable Automatic Mounting 705
Get Mount Points Volume Names 705
Get Mounted Volume Name 706
Mount Volume 707

Permanent Remove Mount Point ... 708
Remove Mount Point ... 709
Printers ... 710
Manage Print Jobs ... 711
Manage Printer ... 712
Register DLL/OCX Action ... 713
Remote Reboot Action ... 714
Remote Shutdown Action ... 715
Run DOS Command Action ... 716
Shell Execute ... 718
Sysinternals PsTools Suite ... 718
PsExec (Execute Remote Process) ... 719
PsFile (List/Close Remote Files) ... 723
PsGetSID (Security ID Info) ... 724
PsInfo (View System Information) ... 725
PsKill (Kill Process) ... 726
PsList (List Processes) ... 727
PsLoggedOn (Logged On Users) ... 728
PsLogList (List event logs) ... 729
PsService (Service Manager) ... 733
PsShutdown (Shutdown Computer) ... 734
PsSuspend (Suspend/Resume Processes) ... 736
System Information ... 737
System Restore ... 738
System Restore Create Restore Point ... 739
System Restore Disable ... 741
System Restore Enable ... 741
System Restore Get Configuration ... 742
System Restore Get Last Restore Status ... 744
System Restore List Restore Points ... 744
System Restore perform Restore ... 746
System Restore Set Configuration ... 747
WBAdmin ... 748
WBAdmin Delete Catalog ... 748
WBAdmin Delete System State Backup ... 749
WBAdmin Disable Backup ... 751
WBAdmin Display Current Backup Settings ... 751
WBAdmin Enable Backup ... 752
WBAdmin Get Disks ... 755
WBAdmin Get Items ... 757
WBAdmin Get Versions ... 758
WBAdmin Restore Catalog ... 759
WBAdmin Start Backup ... 760
WBAdmin Start Recovery ... 763
WBAdmin Start System Recovery ... 765
WBAdmin Start System State Backup ... 767
Window Exists Action ... 767
Windows Page File ... 768
Page File List Performance ... 769
Page File List Properties ... 771
Page File List Use ... 772
Windows Scheduler ... 773
Add Scheduled Task ... 773
Delete Scheduled Task ... 776
End Running Scheduled Task ... 777
Run Scheduled Task ... 778
Scheduling Remote Tasks ... 779
Windows Shares ... 780

Create Share 781
Remove Share 782
Find Share 783
Windows Updates 785
Is Hot Fix Installed 785
List Hot Fixes 786
Update Windows 787
WMI Actions 788
WMI Chkdsk 788
WMI Defrag 790
WMI Format Drive 791
WMI Kill Process Action 793
WMI Process Info Action 794
WMI Process Iterator 796
WMI Run Process Action 797
WMI Schedule Chkdsk 798
28 XML Actions 799
Create XML Node 801
Delete XML Element/Attribute 804
Delete XML Nodes 805
Edit XML File Action 806
Merge XML Action 807
Extract XML Fragment Action 808
Read XML Value to Variable 809
Transform XML 811
Validate XML File 812
XML Document Define 813
XML Node Exists Action 814
XML Node Iterator 816
XML Parser Options 817
XML Save Document 820
Part VII Automating Automise 820
1 Automise IDE 821
Scheduling projects 821
IDE Command Line Interface 825
Exit Codes 826
2 Command Line version 827
Exit Codes 829
Part VIII Tips and Tricks 829
1 Two Environments 829
2 Reusing Logic 832
3 A Configurable project 837
4 Analysing Output 842
5 Documenting your Project 849
6 Counting Errors with Try/Catch Blocks 852
7 Using Output Monitors 854
Part IX Reference 857
1 Regular Expression Reference 857
2 INI Files 863

Part X Support

Part X Support 864

1 Known Problems ... 864

2 Automise Support ... 864

Index 866

1 Automise

1.1 Automise Overview

Automate Windows with Automise

What is Automise™?

Automise is a powerful general purpose automation tool that allows anyone to automate just about anything! Once a project is created, it's easy for anyone to run the project (single click or keystroke) or it can be scheduled to run automatically. Automise saves time, often running tasks in seconds that would take minutes or hours if done manually.

Throw out those complicated Batch Files!

Many people use DOS batch files to automate certain tasks. These batch files are typically difficult to maintain, have poor error handling, and little or no error logging. Automise is easy to use, so the automation task doesn't become the domain of one key person!

How does Automise Work?

Automise uses a combination of built in functionality and the ability to automate other programs/executables. Most programs support some sort of automation interface, for example a command line interface, or a COM interface. Automise leverages these interfaces into a consistent and easy to use GUI application. Automise defines Actions, where each action provides an interface to some third party tool or to an internal function. Actions are chained together to create the automation process. Automise also supports Active Scripting, each action exposes events which can be coded in VBScript or JavaScript.

1.2 Version History

Version history is available at: http://www.automise.com/automise-version-history.aspx

1.3 Automise License

Please Read the License Agreement for Automise before using this software.

Automise End User Licence Agreement

VSoft Technologies Pty Ltd End User License Agreement

This software end user license agreement (“EULA”) is a legal agreement between you (either an individual or a single entity) and VSoft Technologies Pty Ltd (VSoft). Read this document carefully before completing the installation process and using the Software. It provides a license to use the Software and contains warranty information and liability disclaimers. By installing, copying, or otherwise using the Software, you are confirming your acceptance of this EULA and agreeing to become bound by the terms and conditions of this license agreement. If you do not agree with the terms of this EULA, do not install or use the Software.

This EULA covers the following software products: “Automise Lite”, “Automise Professional”, “ATCMD” and “Automise ActionStudio”, herein referred to as the Software.

1. Copyright and Ownership

This Software is owned by VSoft Technologies Pty Ltd and is protected by copyright laws and international copyright treaties. Therefore, you must treat the Software like any other copyrighted material except that you may make one copy of the Software solely for backup or archival purposes.

2. Definitions

Virtual Machine – a single instance of an operating system running in a "virtualized" computer.

Site – all of the licensing organisation’s locations within a 10km (6 mile) radius.

3. Grant of License

The licenses granted in this section are subject to the terms and conditions set forth in this EULA. Upon acceptance of this Agreement, VSoft grants you a non-exclusive, non-transferable, limited license to install and use the Software. You may install and use the Software as permitted by the license type purchased as described below in License Types. The license type purchased is specified in the payment receipt and is also displayed in the About dialog in the Software.

4. Editions

4.1 Professional: Professional edition licenses provide the complete functionality of the Software including all actions, the Integrated Development Environment (IDE) and ActionStudio. The Professional edition is available in Named, Concurrent and Enterprise licenses and is subject to the restrictions of these licenses.

4.2 Lite: Lite edition offers a reduced feature set of the Software’s complete functionality. The Lite edition is only available in a Named user license as it is intended to be for individual use only.

4.3 ATCMD Run-time: ATCMD run-time licensing facilitates large scale distribution of Automise’s full functionality. The ATCMD Run-time edition allows Automise Lite and Automise Professional projects to be scheduled and triggered to run on remote computers; however, these licenses do not provide the full Automise IDE for creating projects. ATCMD Run-time edition is provided under the ATCMD Run-Time license described below.

5. License Types

5.1 Named User License: Under the terms of a Named User License, one named user is permitted to use the software. A Named User is a specific individual designated by you to use the Software. A non-human operated device is counted as a Named User if the device utilizes the Software. A Named User License may not be used concurrently on different computers. The Named user may install a reasonable number of copies of the Software, for example on a Desktop and Laptop computer, provided that the specified Named User is the only individual that uses the Software.

5.2 Concurrent User License: Under the terms of a Concurrent User License, you may distribute and install internally within one Site in your organisation up to 10 copies of the Software for each Concurrent User License provided the number of users accessing or using the Software simultaneously does not exceed the Concurrent User Licenses purchased for such use. Only Automise Professional is available for in a Concurrent license.

5.3 Enterprise License: is provided on a case-by-case basis to provide a combination of licenses suitable to an organisation's needs. Enterprise licenses may include Automise Professional Named licenses, Automise Professional Concurrent licenses and ATCMD run-time only licenses.

5.4 ATCMD Run-Time License: provides access to a run-time version of the command-line version of the Software. Under the terms of an ATCMD Run-Time License, one entity is permitted to use the software. An entity is a specific individual designated by you to use the Software. A non-human operated device is counted as an entity if the device utilizes the Software. A single ATCMD Run-Time License may not be used concurrently on different computers.

5.5 Trial License: Under the terms of a Trial License you may use the Software for Evaluation purposes only. You may use the Software to test whether it meets your requirements. The Software is equipped with a mechanism that prevents the usage of the Software after a certain period of time has elapsed. You agree that you will delete the Software from all computers systems to which you have installed it when this date has been reached, or purchase a license to allow you to continue using the Software.

5.6 Not For Resale License (NFR): Under the terms of a Not For Resale License the Software may be installed and used by the licensed user only for the purpose of reviewing or evaluating the Software. Not For Resale Licenses may not be used for commercial, professional, or for-profit purposes.

6. License Restrictions

The following license restrictions apply, except to the extent where contrary to applicable law.

6.1 The Software product is licensed, not sold. You are not obtaining title to the Software or any copyrights. You may not sublicense, rent, lease, convey, modify, translate, convert to another programming language, decompile, reverse engineer, or disassemble the Software for any purpose.

6.2 You may not transfer your rights or license to another person or entity without written permission from VSoft.

6.3 VSoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA and you must destroy all copies of the Software.

6.4 The Software License key file may not be distributed, modified or decrypted.

7. License Enforcement and Software Updates

7.1 The Software may utilise your internal network connection to prevent intentional or unintentional violation of your license agreement by communicating with other running instances of the Software.

7.2 The Software may utilise your Internet connection for the purposes of determining if a new version of the Software is available. No personal details or license information is transmitted during this process.

8. Disclaimer of Warranty

THIS SOFTWARE IS PROVIDED TO YOU "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED INCLUDING BUT NOT LIMITED TO THE APPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE. YOU ASSUME THE ENTIRE RISK AS TO THE ACCURACY AND THE USE OF THE SOFTWARE AND ALL OTHER RISK ARISING OUT OF THE USE OR PERFORMANCE OF THIS SOFTWARE AND DOCUMENTATION. VSOFT DOES NOT WARRANT THAT THE SOFTWARE IS ERROR-FREE OR WILL OPERATE WITHOUT INTERRUPTION. THE SOFTWARE IS NOT DESIGNED, INTENDED OR LICENSED FOR USE IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE CONTROLS, INCLUDING WITHOUT LIMITATION, THE DESIGN, CONSTRUCTION, MAINTENANCE OR OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, AND LIFE SUPPORT OR WEAPONS SYSTEMS. VSOFT SPECIFICALLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH PURPOSE.

9. Limitation of Liability

VSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES WHATSOEVER ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF VSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL VSOFT BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO DAMAGES OR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS, EVEN IF VSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES/JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY. VSOFT’S TOTAL LIABILITY TO YOU FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER WILL BE LIMITED TO THE AMOUNT PAID BY YOU FOR THE SOFTWARE THAT CAUSED SUCH DAMAGE.

10. Marketing

You agree to be identified as a customer of VSoft and agree that VSoft may refer to you by name, trade name and trademark, if applicable, and may briefly describe your business in VSoft’s marketing materials and web site. You hereby grant VSoft a license to use your name and any of your trade names and trademarks solely in connection with the rights granted to VSoft pursuant to this marketing section.

11. General

This Agreement is governed by the laws of the Australian Capital Territory, Australia. If any provision of this Agreement is invalid or unenforceable, that provision shall be enforced to the maximum extent permissible, to give effect to the parties' intention, and the other provisions shall continue in full force and effect. You acknowledge that this license constitutes the entire agreement between the parties, and supersedes any proposal, representation or prior agreement, oral or written, and any other communications. If this Software was acquired outside Australia, then local laws may apply.

Should you have any questions concerning this EULA, or if you desire to contact VSoft for any reason, please contact us directly:
Postal: PO Box 126, Erindale Centre, ACT 2903
Telephone: +61 2 6282 7488
Facsimile: +61 2 6282 7588
Email: salesinfo@vsoft-tech.com.au

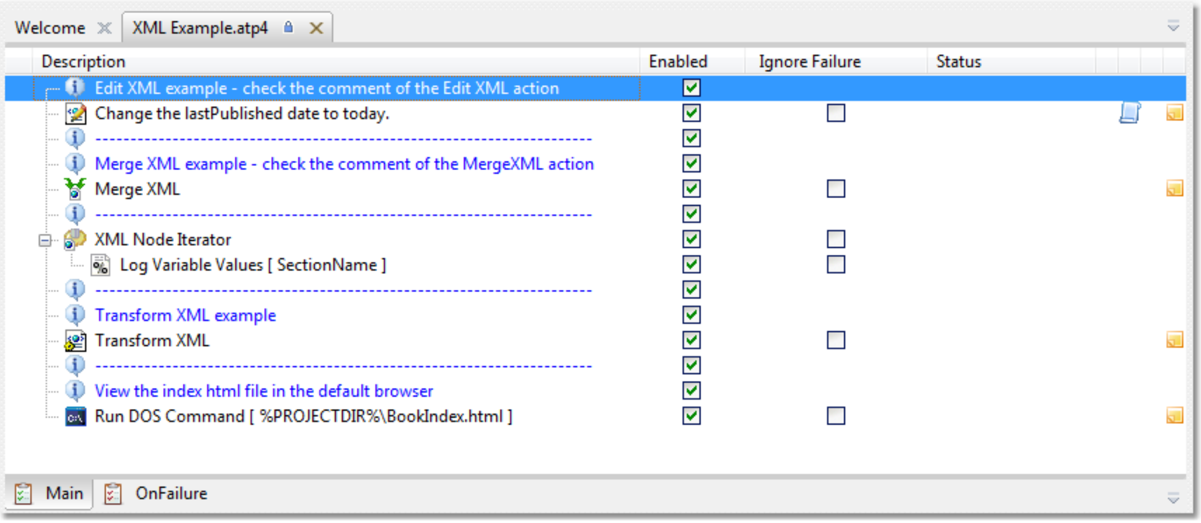
2 Getting Started

2.1 Automise Basic Concepts

- Use the Automise IDE to design your project using Actions (see below)
- Test and Debug your project in the IDE
  - Run your project in the IDE or using ATCMD
  - Distribute your project/s to other people to use with Automise Runner.

Actions and Action Lists

Automise projects have one or more Action Lists, which in turn contain Actions to perform the project. Each action represents one step in the process, by performing a task such as copying files, creating directories, etc.



Description	Enabled	Ignore Failure	Status
Edit XML example - check the comment of the Edit XML action	☑		
Change the lastPublished date to today.	☑	☐	
--	☑		
Merge XML example - check the comment of the MergeXML action	☑		
Merge XML	☑	☐	
--	☑		
XML Node Iterator	☑	☐	
Log Variable Values [SectionName]	☑	☐	
--	☑		
Transform XML example	☑		
Transform XML	☑	☐	
--	☑		
View the index html file in the default browser	☑		
Run DOS Command [%PROJECTDIR%\BookIndex.html]	☑	☐	

Main and OnFailure

By default each Automise project has a Main Action List and an OnFailure Action List. The project starts at the first enabled action in the Main action list, if an error occurs then it can be either handled by Try Catch Actions, or the project will switch to the OnFailure Action List (assuming it is not empty) and continue from there. You can think of the OnFailure Action List as a global error handler, it is from here you can perform cleanup tasks when a project fails, such as deleting temporary or intermediate files etc.

User Action Lists

In addition to the default Action Lists, Automise also allows you to define custom Action Lists for each project, which can then be run using the Run Action List action.

See Also

Working with the Action Types Panel | Action Lists

2.2 Automise IDE Modes

The Automise IDE supports three main modes of operation:

- Design mode - create and edit your project
- Debug mode - use breakpoints, step through you project and watch variables
- Running mode - displays status, statistics, and progress

The way these modes are achieved is by various tabs in the Automise user interface. The **Design & Debug** tab is used for the design and debug modes. Here, you can work on your project, adding actions, editing script and debugging variables. The **Run** tab is selected automatically when running a project. It gives statistics about the currently running project.

2.3 Upgrading Projects From Previous Versions

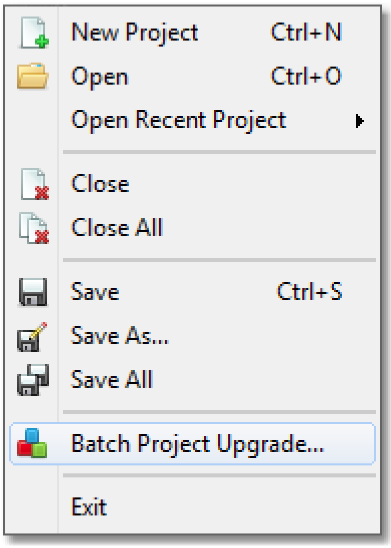
Upgrading One Project File

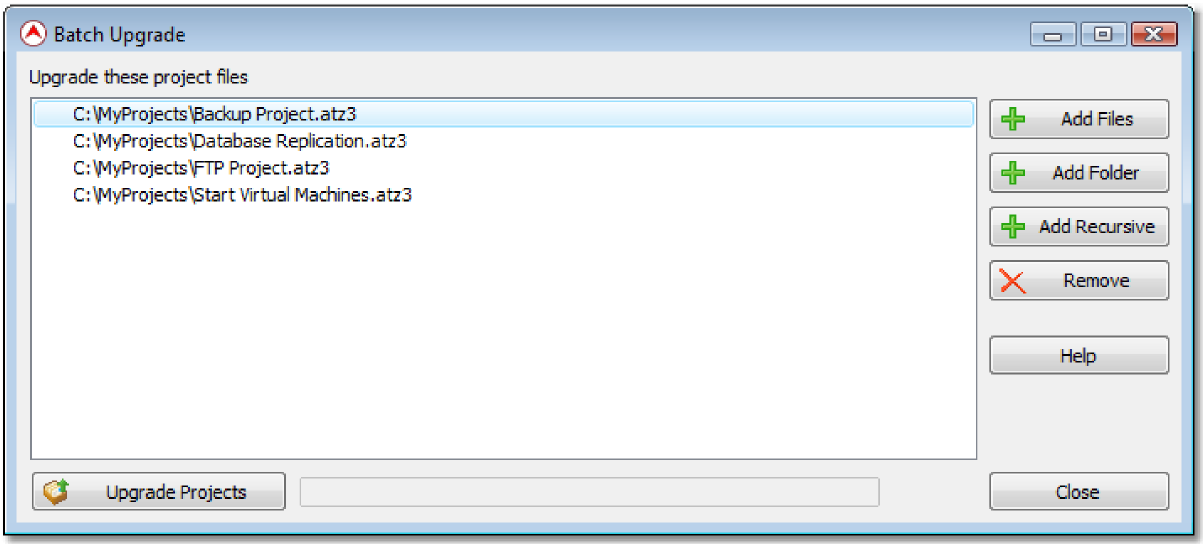
Automise can load any project file from any previous version of Automise. When you save the project file, it is automatically converted to the latest version.

Upgrading Several Project Files

To convert a number of project files at once, use the **Batch Project Upgrade** as follows.

Select **Batch Project Upgrade...** from the **File** menu.





C:\MyProjects\Backup Project.atz3
C:\MyProjects\Database Replication.atz3
C:\MyProjects\FTP Project.atz3
C:\MyProjects\Start Virtual Machines.atz3

Add Files: Select one or more project files to be upgraded.
Add Folder: Select a folder which will be scanned for old project files. Each file found is added to the list of project files to be upgraded.
Add Recursive: Select a folder which will be scanned for old project files (including any subfolders). Each file found is added to the list of project files to be upgraded.
Remove: Removes the selected project file(s) from the list of project files to be upgraded.
Upgrade Projects: Starts the upgrade process.

The Batch Upgrade processes the project files one at a time, saving them to new files. If a project has already been upgraded then the file will be skipped. If there is an error upgrading the project then you will be alerted via a dialog box and instructed to perform the upgrade manually (ie. open the file in the IDE and save it). Please note that the upgrade process will not modify or delete any of your old project files (including their auxiliary files).

After the upgrading process is complete, you can double click on the name of a project file in the list to see whether it was converted. For example, if a project file has already been upgraded and saved in the same directory, it will be skipped in the upgrade process.

When upgrading from old Automise projects, project files are converted as follows.

Previous file	**Saved as**
.atp3	.atp4
.atz3	.atz4
.atv, .atw, .atd	.atpinf
.atl2, .lck	(not converted)

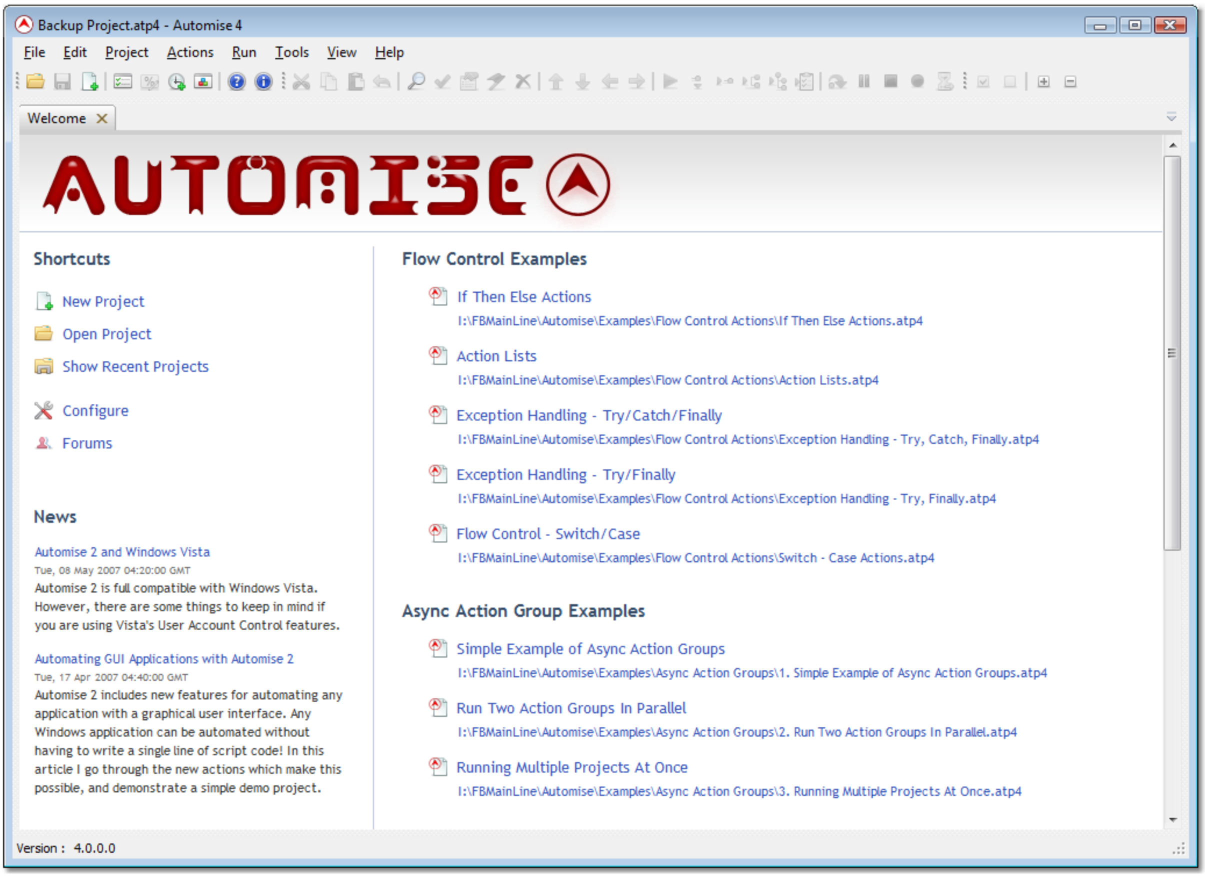
See also

Project Files

3 Automise IDE Reference

3.1 Welcome Page

The Welcome Page is a kind of portal within Automise, providing access to recent projects, news about current releases, tips of the day and other useful links. To show the Welcome Page, click the **Welcome** tab in the top left of the main window.



Shortcuts

- New Project
- Open Project
- Show Recent Projects
- Configure
- Forums

News

Automise 2 and Windows Vista
Tue, 08 May 2007 04:20:00 GMT
Automise 2 is full compatible with Windows Vista. However, there are some things to keep in mind if you are using Vista's User Account Control features.

Automating GUI Applications with Automise 2
Tue, 17 Apr 2007 04:40:00 GMT
Automise 2 includes new features for automating any application with a graphical user interface. Any Windows application can be automated without having to write a single line of script code! In this article I go through the new actions which make this possible, and demonstrate a simple demo project.

Flow Control Examples

If Then Else Actions
I:\FBMainLine\Automise\Examples\Flow Control Actions\If Then Else Actions.atp4

Action Lists
I:\FBMainLine\Automise\Examples\Flow Control Actions\Action Lists.atp4

Exception Handling - Try/Catch/Finally
I:\FBMainLine\Automise\Examples\Flow Control Actions\Exception Handling - Try, Catch, Finally.atp4

Exception Handling - Try/Finally
I:\FBMainLine\Automise\Examples\Flow Control Actions\Exception Handling - Try, Finally.atp4

Flow Control - Switch/Case
I:\FBMainLine\Automise\Examples\Flow Control Actions\Switch - Case Actions.atp4

Async Action Group Examples

Simple Example of Async Action Groups
I:\FBMainLine\Automise\Examples\Async Action Groups\1. Simple Example of Async Action Groups.atp4

Run Two Action Groups In Parallel
I:\FBMainLine\Automise\Examples\Async Action Groups\2. Run Two Action Groups In Parallel.atp4

Running Multiple Projects At Once
I:\FBMainLine\Automise\Examples\Async Action Groups\3. Running Multiple Projects At Once.atp4

Version : 4.0.0.0

Getting Started

The Getting Started section contains useful links:

- Create a new project
- Open an existing project
- Getting started tutorial - opens a tutorial recommended for first-time users.
- Open help file
- Automise Support Forums - takes you to the online support forums where you can get help from VSoft support staff and other users. The forums are opened in your default web browser.

Tip Of The Day

A new tip of the day loads automatically each time you open Automise.

Learn More About

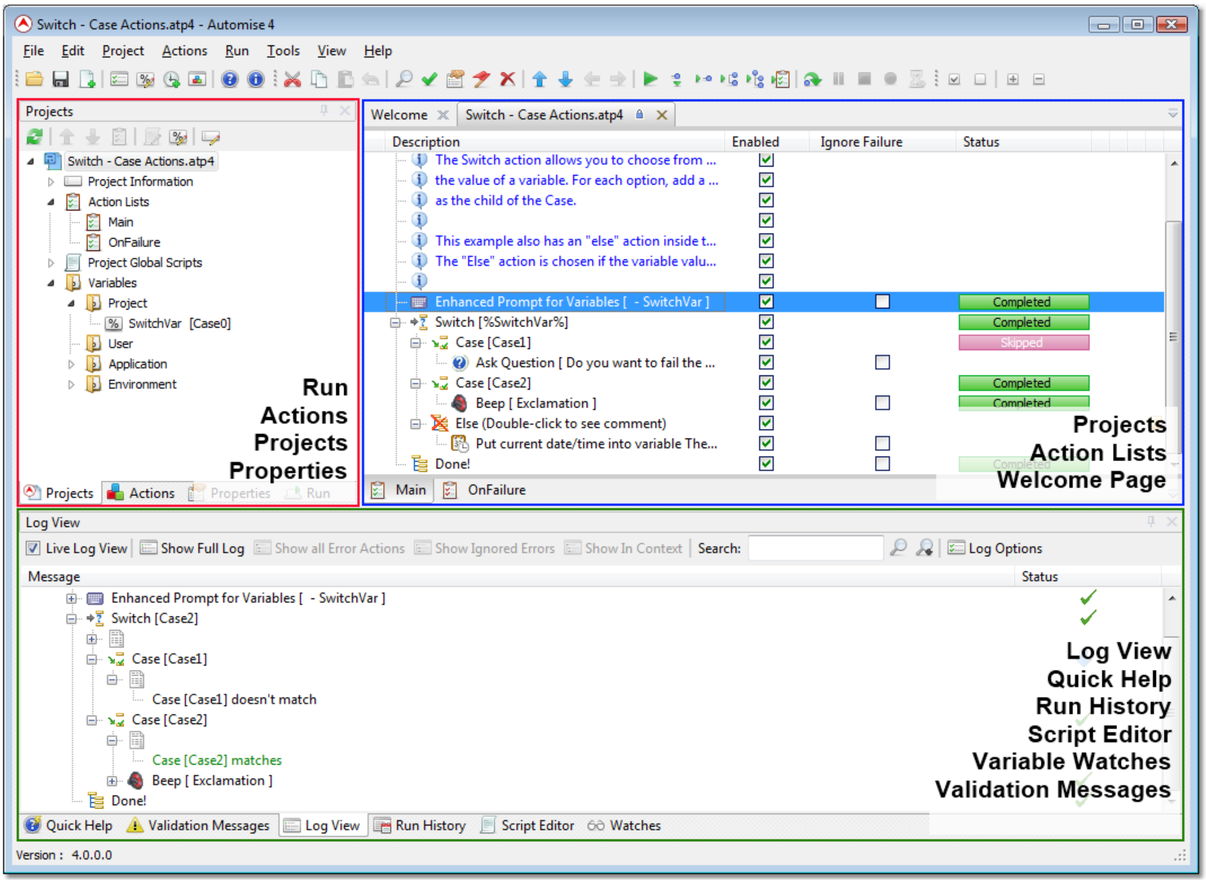
These links point to help on the most important concepts you need to know about. If you're new to Automise, we suggest having a read.

Newsfeed

The Newsfeed displays an RSS feed of recent events of interest. There are several feeds available, such as news, blogs, and latest articles. You can also choose any RSS feed you like, by clicking on the "Customise" link as well as disabling the newsfeed completely.

3.2 IDE Design Mode

Automise's Design mode consists of three main sections:



Description	Enabled	Ignore Failure	Status
The Switch action allows you to choose from ...	☑		
the value of a variable. For each option, add a ...	☑		
as the child of the Case.	☑		
	☑		
This example also has an "else" action inside t...	☑		
The "Else" action is chosen if the variable valu...	☑		
	☑		
Enhanced Prompt for Variables [- SwitchVar]	☑	□	Completed
Switch [%SwitchVar%]	☑		Completed
Case [Case1]	☑		Skipped
Ask Question [Do you want to fail the ...	☑	□	
Case [Case2]	☑		Completed
Beep [Exclamation]	☑	□	Completed
Else (Double-click to see comment)	☑		
Put current date/time into variable The...	☑	□	
Done!	☑	□	Completed

Message	Status
Enhanced Prompt for Variables [- SwitchVar]	✓
Switch [Case2]	✓
Case [Case1]	
Case [Case1] doesn't match	
Case [Case2]	
Case [Case2] matches	
Beep [Exclamation]	
Done!	

Project/Action Types/Action Inspector

This section contains four tabs:

- Project: Contains a tree of the action lists, global script and variables in your project.
- Actions: Lets you add new actions to your project.
- Properties: Lets you edit any property of any action, as an alternative to using the properties dialog.
- Run Tab: Shows progress and statistics while projects are running.

Action Lists

This is where you create and edit the actions of your project. There are two built-in action lists, Main and OnFailure. You can also add extra actions lists. To add an action to an Action list, just click on the Action Type that you want to add, it will be inserted after the currently selected action in the current action list, or you can drag and drop the action on the list where you would like it. This is covered in more detail later in Working with the Action Types Panel.

Log/Script Editor

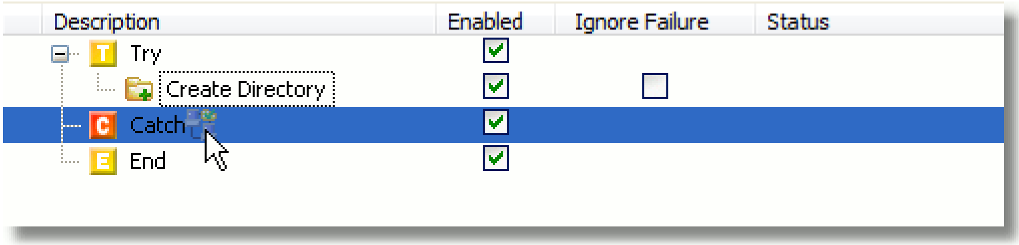
There are several tabs in the bottom section, but the most relevant ones during design of your project are the Script Editor and the Quick Help.

3.2.1 Working with Actions

Adding Actions to an Action List

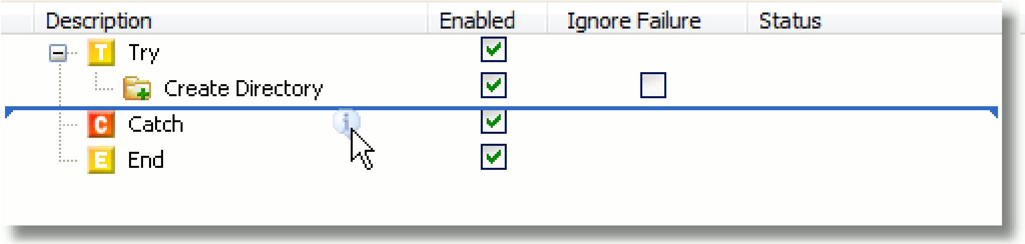
You can add actions to an Action List using two methods. Simply clicking on the name of an Action Type in the Actions Tab will add the Action to the Action List after the currently selected Action in the Action List. You can also use Drag and Drop to add actions to the Action list. Using Drag and Drop enables you to place the new action with more precision.

The Action List Tree view provides guide lines to indicate where the action will be dropped. In this example the action will be dropped as a child to the highlighted action, because the mouse is over the icon or name of the action.



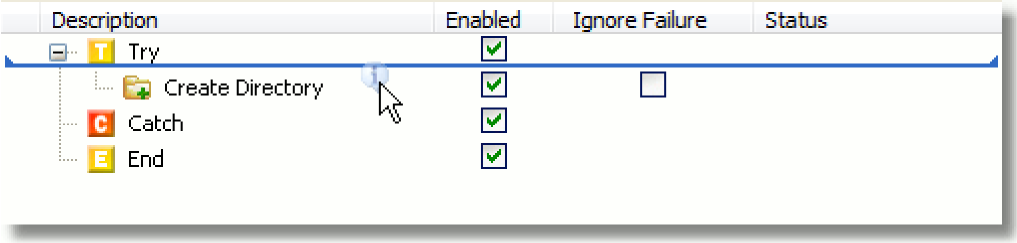
Description	Enabled	Ignore Failure	Status
Try	☑		
Create Directory	☑	□	
Catch	☑		
End	☑		

In this example the Action will be dropped Before the action under the guide line, because the guide line indicators point down.



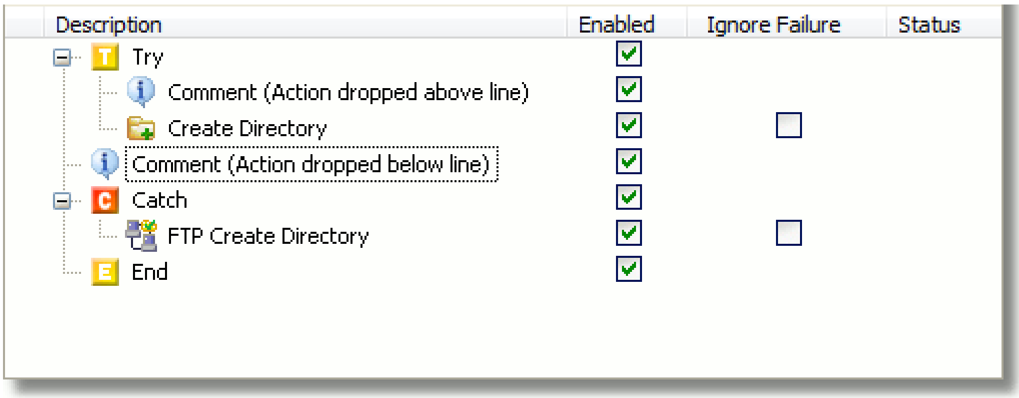
Description	Enabled	Ignore Failure	Status
Try	☑		
Create Directory	☑	□	
Catch	☑		
End	☑		

In this example the Action will be dropped After the action above the line, because the guide line indicators point up.



Description	Enabled	Ignore Failure	Status
Try	☑		
Create Directory	☑	□	
Catch	☑		
End	☑		

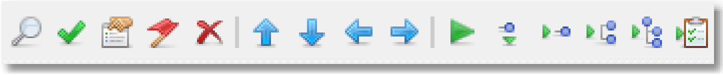
After the above Drag and Drop operations our action list looks like this :



Description	Enabled	Ignore Failure	Status
Try	☑		
Comment (Action dropped above line)	☑		
Create Directory	☑	☐	
Comment (Action dropped below line)	☑		
Catch	☑		
FTP Create Directory	☑	☐	
End	☑		

Moving Actions

Actions can be moved using Drag and Drop, or using the Arrow buttons on the Actions Toolbar.



You can also use the Ctrl+Arrow keys to move actions up and down or to indent/outdent actions.

Any Action (apart from Comment actions) can be a parent to other actions. When an Action has child Actions, it executes first, and then if it succeeds then the child actions execute.

Selecting Actions

The usual Windows selection rules apply, using the control and shift keys to select multiple actions. You can only select multiple actions at the same level in the tree.

Copying Actions

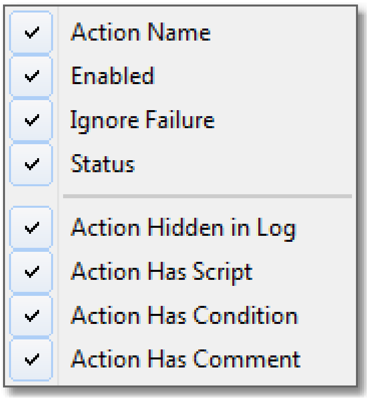
Actions can be copied and pasted using the clipboard, in which case the actions are pasted after the currently selected Action. You can also use Drag and Drop with the Control Key down to copy the dragged Actions.

Deleting Actions

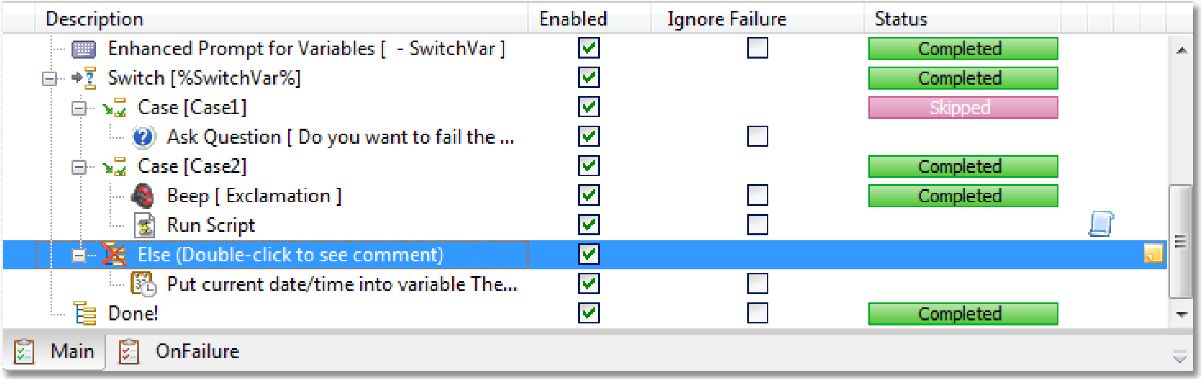
To delete an Action, select it and press the delete key, or use the Delete button on the Actions Toolbar.

Action List Columns

You can select which columns are shown in the action list view by right clicking on any column header.



Columns:



Description	Enabled	Ignore Failure	Status
Enhanced Prompt for Variables [- SwitchVar]	☑	☐	Completed
Switch [%SwitchVar%]	☑		Completed
Case [Case1]	☑		Skipped
Ask Question [Do you want to fail the ...	☑	☐	
Case [Case2]	☑		Completed
Beep [Exclamation]	☑	☐	Completed
Run Script	☑	☐	
Else (Double-click to see comment)	☑		
Put current date/time into variable The...	☑	☐	
Done!	☑	☐	Completed

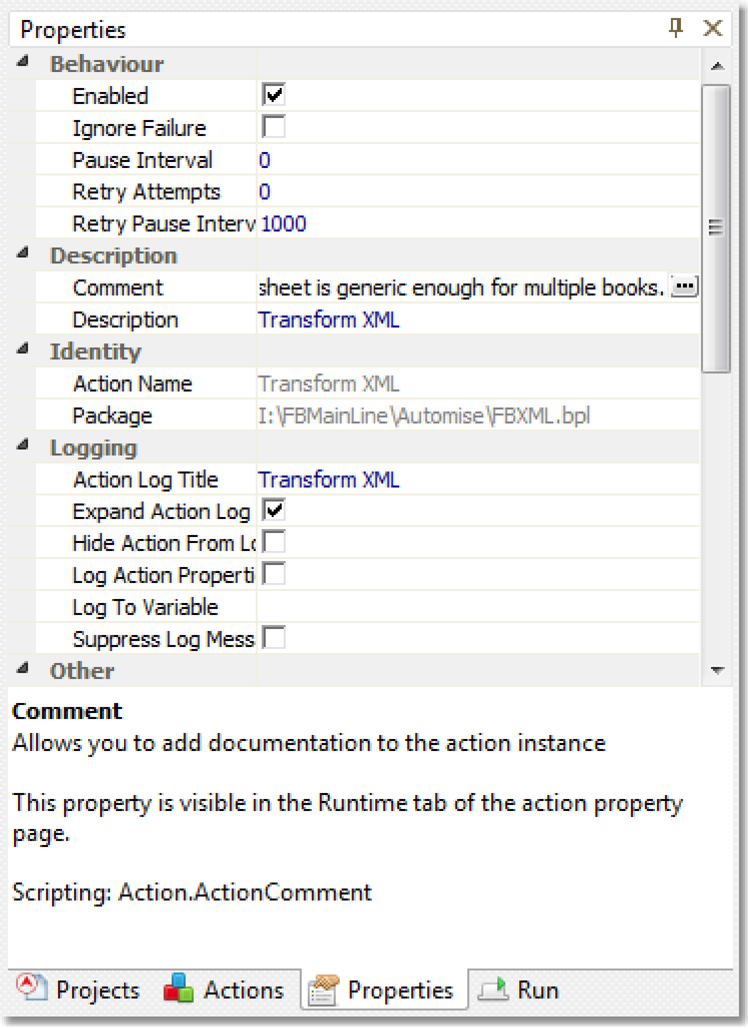
- Action: shows the underlying name of the action, such as "Action Group" or "List Iterator". Useful if you frequently rename actions.
- Enabled: shows a checkbox allowing you to quickly enable or disable actions.
- Ignore Failure: shows a checkbox that specifies whether execution continues even if this action fails. See Ignore Failure.
- Status: shows a coloured bar indicating statuses such as "Completed" or "Error".
- Action Hidden in Log: Shows an icon if the "Hide action from log" action property is set.
- Action Has Script: Shows an icon if there is a script event attached to the action.
- Action Has Condition: Shows an icon if there is an execute condition defined on the action.
- Action Has Comment: Shows an icon if the action has text in the comment field.

3.2.2 Properties tab

The Properties Tab provides an easy way to view and change some properties of a selected action or multiple selected actions. When an action is selected in the Action List, then the Properties Tab updates with the properties of that action. Almost every property displayed can be edited, except for the Action Name and Package properties. You can also edit properties which are not visible the edit window for a given action, such as "Expand Action Log Title", which controls whether variable references in action titles are expanded in the log.

Every action property can also be accessed through scripting. The context-sensitive description at the bottom shows the name

Some actions provide a context sensitive description at the bottom of the Action Inspector.



Properties

Behaviour

Enabled	☑
Ignore Failure	☐
Pause Interval	0
Retry Attempts	0
Retry Pause Interv	1000

Description

Comment	sheet is generic enough for multiple books.
Description	Transform XML

Identity

Action Name	Transform XML
Package	I:\FBMainLine\Automise\FBXML.bpl

Logging

Action Log Title	Transform XML
Expand Action Log	☑
Hide Action From Lo	☐
Log Action Properti	☐
Log To Variable	
Suppress Log Mess	☐

Other

Comment
Allows you to add documentation to the action instance

This property is visible in the Runtime tab of the action property page.

Scripting: Action.ActionComment

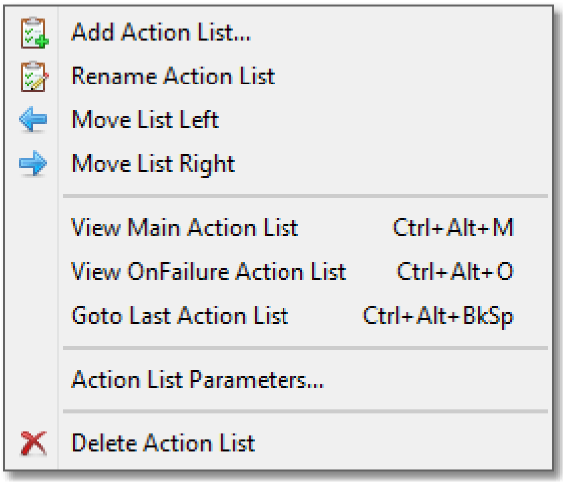
3.2.3 Action Lists

By Default, a new Automise project has two Action Lists, Main and OnFailure. These default Action Lists cannot be deleted or moved. When a project starts, the first enabled action in the Main Action List is executed. If any action fails, execution switches to the OnFailure Action List, if it contains any actions. The OnFailure Action List allows you to perform cleanup tasks when the project fails for any reason.

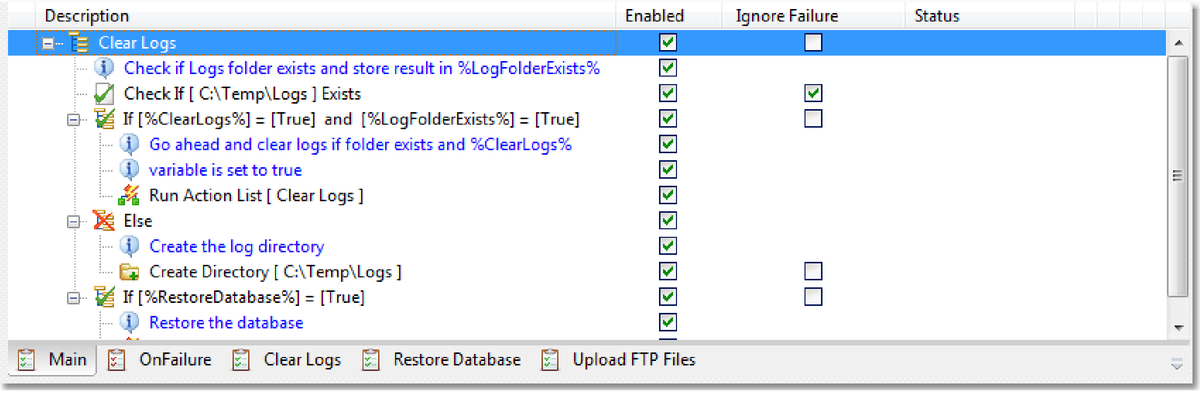
(Tip : You can drag actions from one action list to another by dragging them over the action list tab, then dropping them on the other action list.)

Adding Action Lists

You can Add, Delete, Rename or re-order extra Action Lists from the Project menu, or by right clicking on the Action List headings:


Add Action List...
Rename Action List
Move List Left
Move List Right
View Main Action List Ctrl+Alt+M
View OnFailure Action List Ctrl+Alt+O
Goto Last Action List Ctrl+Alt+BkSp
Action List Parameters...
Delete Action List

To run a specific action lists, add a "Run Action List" action to the calling action list, then set the ActionList property of that action.



Description	Enabled	Ignore Failure	Status
Clear Logs	☑	□	
Check if Logs folder exists and store result in %LogFolderExists%	☑		
Check If [C:\Temp\Logs] Exists	☑	☑	
If [%ClearLogs%] = [True] and [%LogFolderExists%] = [True]	☑	□	
Go ahead and clear logs if folder exists and %ClearLogs%	☑		
variable is set to true	☑		
Run Action List [Clear Logs]	☑		
Else	☑		
Create the log directory	☑		
Create Directory [C:\Temp\Logs]	☑	□	
If [%RestoreDatabase%] = [True]	☑	□	
Restore the database	☑		

In the above screenshot, the Run Action List action calls the "Clear Logs" action list if certain conditions are met.

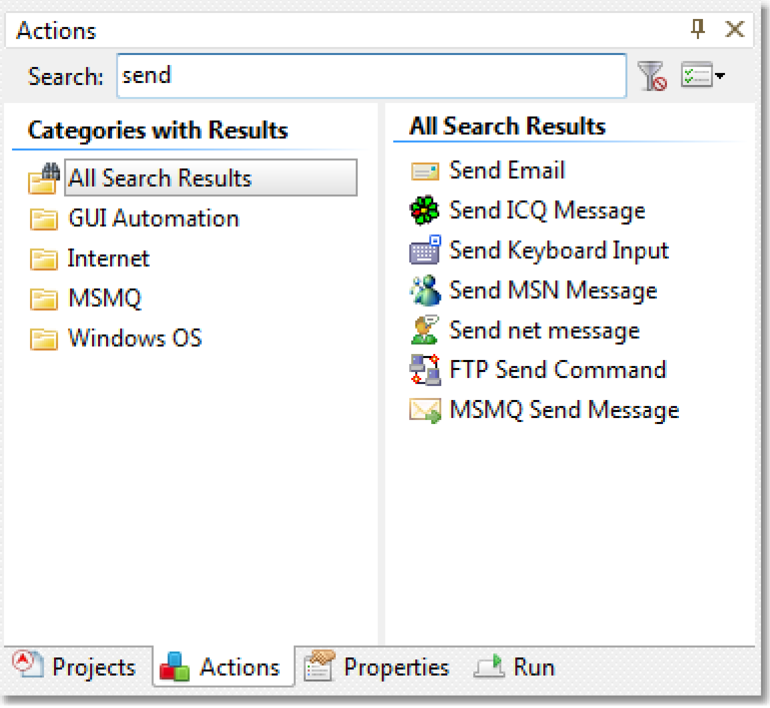
Action Lists can be treated like subroutines - you can call them as often as you require. Through the use of Action List Parameters and Automise variables, Action Lists can be used like functions or method calls.

See Also

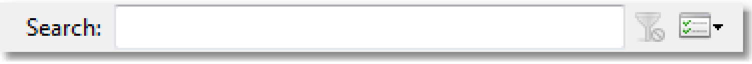
Automise Variables | Action List Parameters

3.2.4 Actions Tab

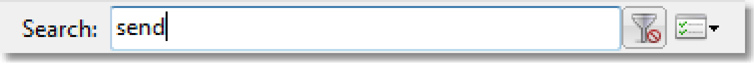
The Actions Tab is used to search for actions to add to your project. To add a new action to your project, simply click on it.



The available actions are grouped in categories. The categories are based either on functionality, or in some cases on the third party product they support, for example "Source Safe" or "CVS". You can search for Action Types by name using the Search edit box at the top of the Actions Tab. You can use the keyboard shortcut Ctrl+I to set the focus to the filter edit control:



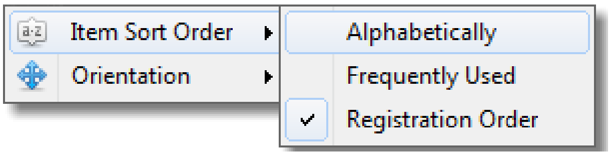
To clear the search filter click on the "Clear Search Field" button or press Escape :



Note: Only actions that are in installed packages are displayed. If you can't find an action you need, try adding packages in the Package Manager.

Options:

There are two display options for the Actions Tab: Item Sort Order and Orientation.



Item Sort Order controls the order the actions are displayed in.

- **Alphabetically**
- **Frequently Used** - more frequently used actions are placed first
- **Registration Order** - the order chosen by the designer of the actions

Orientation controls the display of the categories and actions lists:

- **Left to right**: Two columns
- **Top to bottom:** One column, with the actions below categories.

Note: the orientation display also applies to the Options screen.

Tip:

To search for an action and add it without using the mouse:
1. Press Ctrl+I to focus the action filter
2. Type a few letters of the name of the action
3. Press the Down arrow key until you reach the action.
4. Press Enter to create the new action.

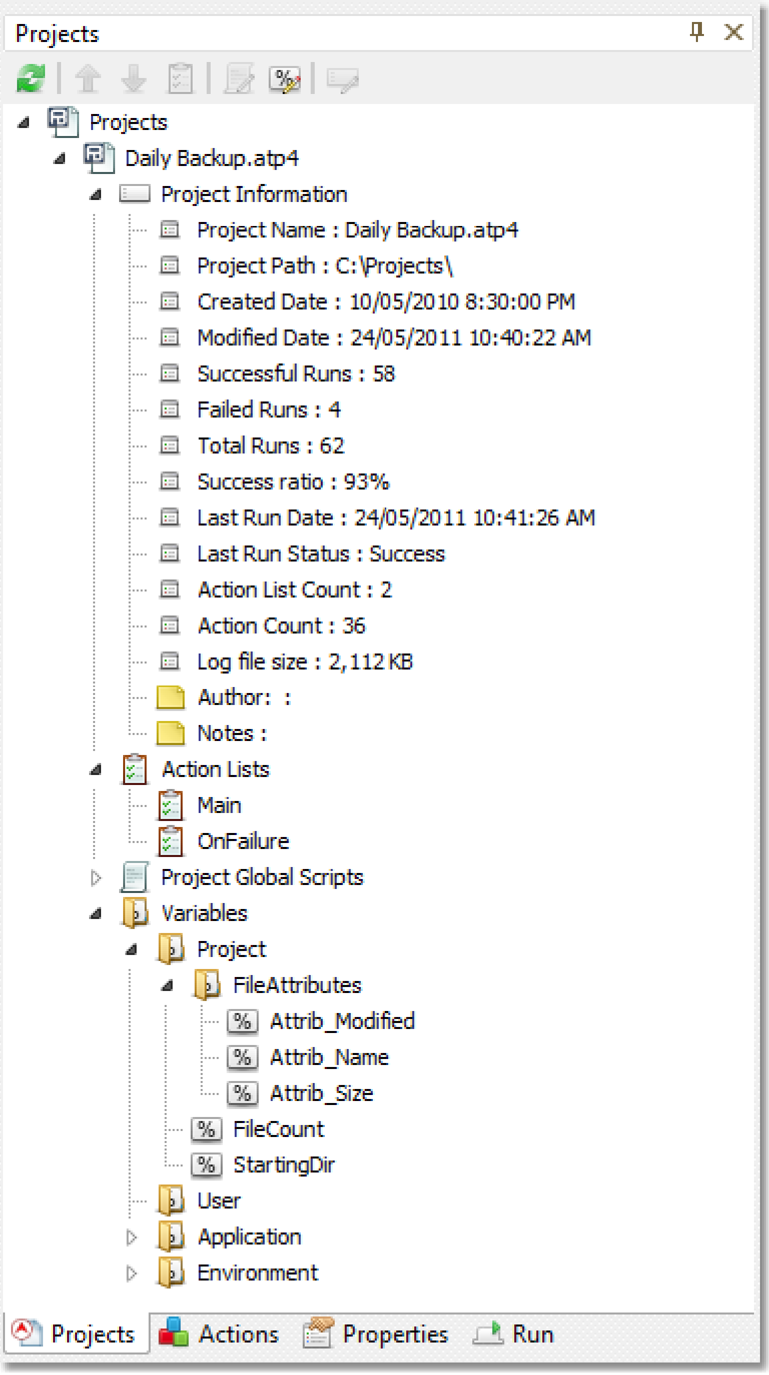
See Also

Action Lists

3.2.5 Project Tree

The project tree displays the following information:

- Project file information, such as name, location, author, notes.
- All the action lists in your project
- Nodes for the Project Global Script
- All the project, user, system and environment variables for your project



- Projects
  - Daily Backup.atp4
    - Project Information
      - Project Name : Daily Backup.atp4
      - Project Path : C:\Projects\
      - Created Date : 10/05/2010 8:30:00 PM
      - Modified Date : 24/05/2011 10:40:22 AM
      - Successful Runs : 58
      - Failed Runs : 4
      - Total Runs : 62
      - Success ratio : 93%
      - Last Run Date : 24/05/2011 10:41:26 AM
      - Last Run Status : Success
      - Action List Count : 2
      - Action Count : 36
      - Log file size : 2,112 KB
      - Author: :
      - Notes :
    - Action Lists
      - Main
      - OnFailure
    - Project Global Scripts
    - Variables
      - Project
        - FileAttributes
          - Attrib_Modified
          - Attrib_Name
          - Attrib_Size
        - FileCount
        - StartingDir
      - User
      - Application
      - Environment

You can do the following things in the Project Tree:

Action Lists

Add a new action list by right clicking any action list and choosing "Add Action List..."

To reorder action lists:
- Press Ctrl+Up or Ctrl+Down on an action list.
- Right click an action list and select "Move Action List Up/Down".
- Select an action list and click up the up or down button on the project toolbar.

Project Global Scripts

To open a Project Global Script:

- Double click the node (VBScript, JavaScript, or PowerShell)
- Click the toolbar button
- Right click a global script node and select "Edit Global Scripts"

Variables

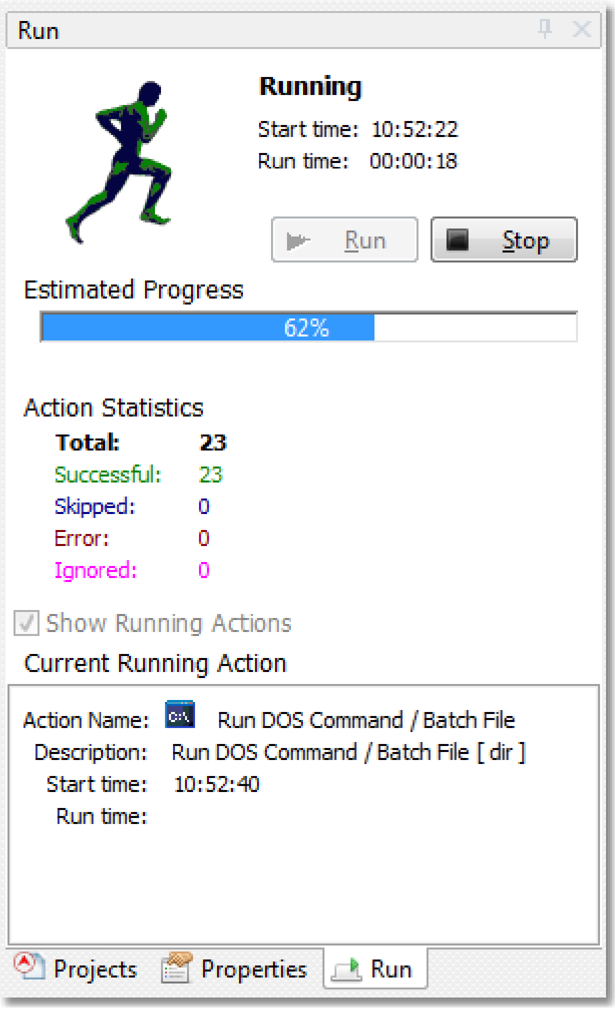
To add a variable, right click Variables and select "Add Variable"

To edit a variable, right click it and select "Edit Variable", or double click it.

To open the Edit Variables dialogue, select "Edit Variables" from any variable's context menu, or click the Toolbar icon.

3.2.6 Run Tab

The Run Tab shows the status of the currently running project, with a progress bar. The bar increments by one every time an action completes while your project runs, and the total is dynamically calculated. This is what the progress bar looks like during a run:



Running

Start time: 10:52:22
Run time: 00:00:18

Estimated Progress

62%

Action Statistics

Total:	**23**
Successful:	23
Skipped:	0
Error:	0
Ignored:	0

☑ Show Running Actions

Current Running Action

Action Name: Run DOS Command / Batch File
Description: Run DOS Command / Batch File [dir]
Start time: 10:52:40
Run time:

Estimated Progress

The calculation of progress is only an estimate - these are the steps Automise makes to estimate the run progress:

1. When a project first starts, the progress is set to zero and the progress bar total is calculated.
2. First the log archive is queried to find the last successful run. If there was a last successful run, then the amount of actions that executed in that run is used to set the total.
3. If there was no last successful run, then Automise calculates how many actions are in your project and uses that as the total.

For some projects, the above methods to calculate the progress may not be adequate. For example, a project may have two modes (eg. Full and Quick Backup). The two different modes may have very different progress totals, and using the last successful run action count method will only work some of the time. You can override the estimated total using a either the script method "SetEstimatedProgressTotal", or by using the Set Estimated Progress Total Action.

To display a custom status message below the status bar, use the Set Run Status Message action.

Running Statistics

This section displays the following information:

- A "running man" animation to indicate the status of the run. When the project completes this changes to either a tick or cross depending on the result.
- Buttons to Run or Stop the run
- Estimated progress of the entire run.
- Action Statistics (total, successful actions, skipped, actions in error, errors ignored)
- List of recent actions (showing result of action, the run time and action description)

Current Running Actions

This section shows the current ActionList as well as a list of all the currently running actions (it is possible to have more than one action running simultaneously using the ASync Action Group)

Setting the estimated progress using script

The SetEstimatedProgressTotal script method takes a single integer parameter which allows the progress total to be set to any value at any time.

Using the same example above - your project has Full and Quick Backup modes. You know (because you've run the Full mode enough times) that full mode will run 147 actions, and that the Quick Backup runs only 38 actions. Your project will have some logic at the start which determines if Full or Quick Backup is required and using that same logic it can call the script function to set the estimated progress total (in Javascript):

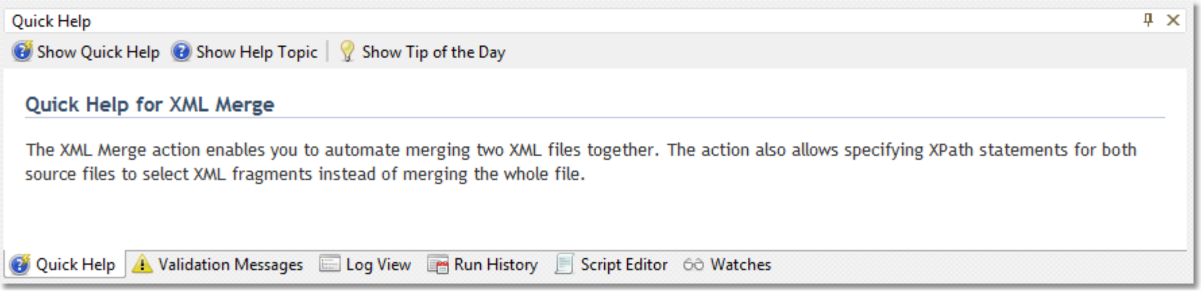
```
if (FullBackup) {
  SetEstimatedProgressTotal(147)
} else {
  SetEstimatedProgressTotal(38)
}
```

This gives you complete control over the estimated progress, and it can be set at any point in your project.

3.2.7 Quick Help

The Quick Help tab at the bottom of the Design & Debug view provides summary help information for an action. To view the quick help for an action:

- Select an action in an action list, and:
  - press Alt F1, or
  - right-click and select Quick Help, or
  - click the "Show Quick Help for ..." button on the quick help tab
- Right-Click on an action in Action Types and select "Show Quick Help"



Quick Help for XML Merge

The XML Merge action enables you to automate merging two XML files together. The action also allows specifying XPath statements for both source files to select XML fragments instead of merging the whole file.

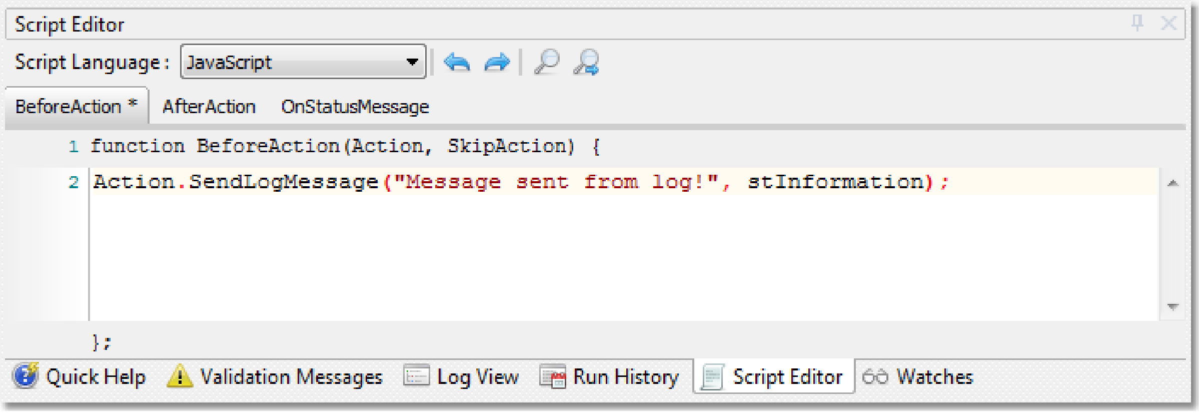
You can also see the quick help for an action on the General tab of the action's edit window.

The Quick Help also displays information about new versions of Automise, either when a new version is detected automatically, or running Check For Updates manually.

3.2.8 Script Editor

The script editor is where you can write VBScript, JavaScript or PowerShell in response to any events fired at a particular action.

Changing the highlighted action in the action list will change the script editor to show the available events and the script code for the selected event of the current action.



Script Language: JavaScript

```
function BeforeAction(Action, SkipAction) {
Action.SendLogMessage("Message sent from log!", stInformation);
};
```

For more information on scripting see Scripting in Automise.

3.2.9 Keyboard Shortcuts

Use the following keyboard shortcuts to quickly navigate your way around Automise.

	Embarcadero style	**Microsoft style**
Run	F9	F5
Step	F8	F10
Stop run	Escape	Shift+F5
Run from action	Ctrl+F9	Alt+F5
Run selected actions	Alt+F9	Ctrl+Alt+F5
Run selected actions and children	Ctrl+Alt+F9	Shift+Alt+F5
Run current action list	Shift+Ctrl+F9	Shift+Ctrl+F5
Continue	Shift+F9	Ctrl+F5
Enable/disable action	F4	F11
Toggle breakpoint	F5	F9
Clear all breakpoints	Shift+Ctrl+F5	Ctrl+F9
Action properties	F11 / Enter	F12 / Enter
Select Properties Tab	F12	F4

To switch between Embarcadero and Microsoft style key mappings, use the Design Time Options page.

Other design and debugging shortcuts include:

Action	**Shortcut**
Indent action	Ctrl+Right
Outdent action	Ctrl+Left
Move action up	Ctrl+Up
Move action down	Ctrl+Down
Delete action	Del
Collapse node	Left / Numpad minus (-)
Expand node	Right / Numpad plus (+)
Collapse all	Numpad slash (/)
Expand all	Numpad asterisk (*)
Edit action title	F2 / Any unassigned key
Filter log by action	F7
Quick help on action	Alt+F1

See Working with Actions for more information.

Inside actions:

Action help	F1
Edit Field	F2
Add variable	F3

Navigational shortcuts:

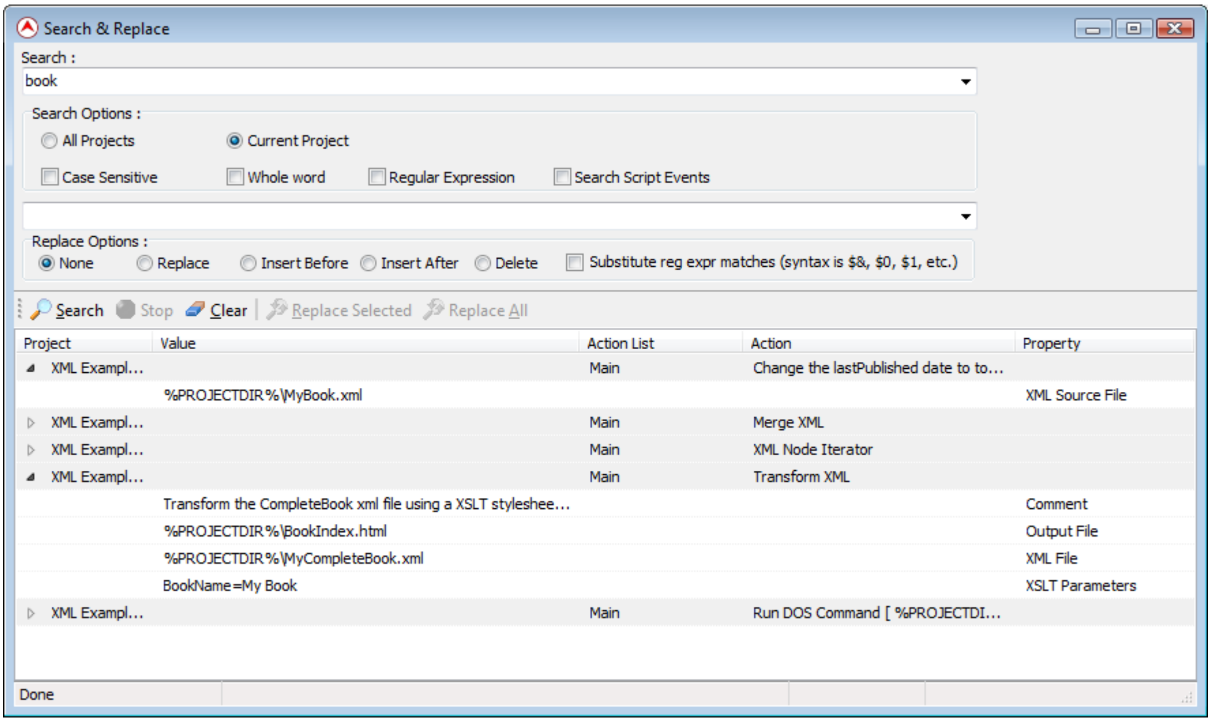
Select History & Statistics tab	Ctrl+H
Select Run Log	Alt+F1
Select Actions tab	Ctrl+I
Edit variables	Shift+F2
Find and replace	Ctrl+F
Next search result	F3
Previous search result	Shift+F3

Project shortcuts

New project	Ctrl+N
Open project	Ctrl+O
Save project	Ctrl+S

3.2.10 Search & Replace

The Search & Replace window is a powerful way of finding actions and also changing text in a number of actions at once, regardless of which property the text occurs in. Launch the Search & Replace window from the Edit menu, or with the keyboard shortcut Ctrl+F.



Search & Replace

Search : book

Search Options : ○ All Projects ● Current Project

□ Case Sensitive □ Whole word □ Regular Expression □ Search Script Events

Replace Options : ● None ○ Replace ○ Insert Before ○ Insert After ○ Delete □ Substitute reg expr matches (syntax is $&, $0, $1, etc.)

Project	Value	Action List	Action	Property
XML Exampl...		Main	Change the lastPublished date to to...	
	%PROJECTDIR%\MyBook.xml			XML Source File
XML Exampl...		Main	Merge XML	
XML Exampl...		Main	XML Node Iterator	
XML Exampl...		Main	Transform XML	
	Transform the CompleteBook xml file using a XSLT styleshee...			Comment
	%PROJECTDIR%\BookIndex.html			Output File
	%PROJECTDIR%\MyCompleteBook.xml			XML File
	BookName=My Book			XSLT Parameters
XML Exampl...		Main	Run DOS Command [%PROJECTDI...	

Done

Searching for text

Search: Enter the search term to be searched for.

Case sensitive: If selected, *BOOK* will not match *book*.

Whole word: The search string must not appear as part of another word: *book* will not match *mybook*.

Regular expression: If selected, the search term becomes a regular expression. For example, *b.*k* would match *book*.

Search Script Events: If selected, the text of script events on actions is also searched. This is important if renaming a variable, for instance.

Press <Enter> or click the **Search** button to search for your specified term. For each match, the action list, action, property and value are shown. Note that a single action usually matches several times, because the string is contained in the title, the log title, and so forth.

Double-click on a match to highlight the action in the Design & Debug window. You can then use the keyboard shortcuts F3 and Shift+F3 to move forward and backward through matches.

Stop: Aborts the search, if it is taking too long.

Clear: Removes search results.

Replace

To replace matches, first perform a search as above.

Replace: The string that search matches will be replaced with.

Replace options:
- **None:** No replacements can be made. Functions as a sort of "safety catch".
- **Replace:** Replacements will be made as normal.
- **Insert before:** Replacement text is added before the search text.
- **Insert after:** Replacement text is added after the search text.
- **Delete:** Search term is replaced with nothing.
- **Substitute regular expression matches:** If using the "regular expression" search option, use this option to allow subexpressions to be substituted.

After performing a search and setting the replace options, click one of the replace buttons:
- **Replace Selected:** Select the matches you wish to replace first. The replacements are made, and the matches removed from the list.
- **Replace all:** All matches are replaced at once.

Example of using regular expression substitution:
- Search for "rm (.*).txt" with Regular Expression flag on.
- Replace with "del $1.xml" with Substitute Regular Expression Matches flag on.
- A match such as "delete file.txt" will be replaced with "del file.xml"

3.2.11 Error Handling

Errors can be handled at three levels: individual actions, groups of actions, or project level.

Individual actions

Set the "Ignore Failure" flag on an action, and follow it with a "If Prev Action Failed"

action to handle an anticipated error in a single action.
Each action can also retry if it fails, for more information on this see Timing Properties.

Groups of actions

For localized error handling, use the Try, Catch, Finally and End actions in the Flow Control category. The valid combinations are Try-Catch-End, Try-Finally-End or Try-Catch-Finally-End. To use these actions you must place the actions you want to run as child actions of the try, catch or finally.

Localised error handling works as follows:

1. Child actions of the TRY are executed
2. If any child action of the TRY fails, then execution skips to the Catch or Finally section
3. If the CATCH section exists, and an action in the TRY section failed, then any child actions are executed
4. If the FINALLY section exists, then child actions are always executed.
5. The END action signifies the end of the exception handling block

Project level error handling

Every project contains an OnFailure action list which is executed if any action in the Main action list fails and it isn't handled by a Try-Catch block.

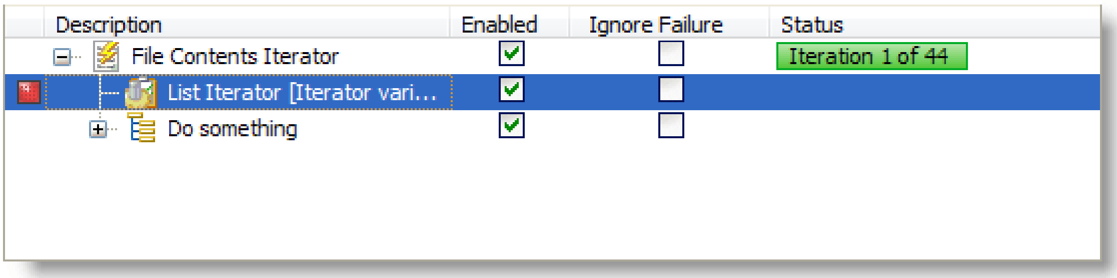
By default, if an action fails it will abort running the Main action list, and run any actions in the OnFailure action list. If you want to ignore an error for a particular action and continue processing, then set the Ignore Error property of the action.

3.3 IDE Debug Mode

After you have created your project, you may want to debug it. Automise allows you to step one action at a time through your project, to use breakpoints to pause the run at particular points, and to view and edit the values of variables when the run is paused. You can also make use of the live logging view of the log when debugging.

How to Add and Remove Breakpoints

- Right click on an action and choose Breakpoint.
- Press F5 (or F9, if using "Microsoft Style" key bindings)
- Choose Breakpoint from the action menu.
- Click in the gutter to the left of the action



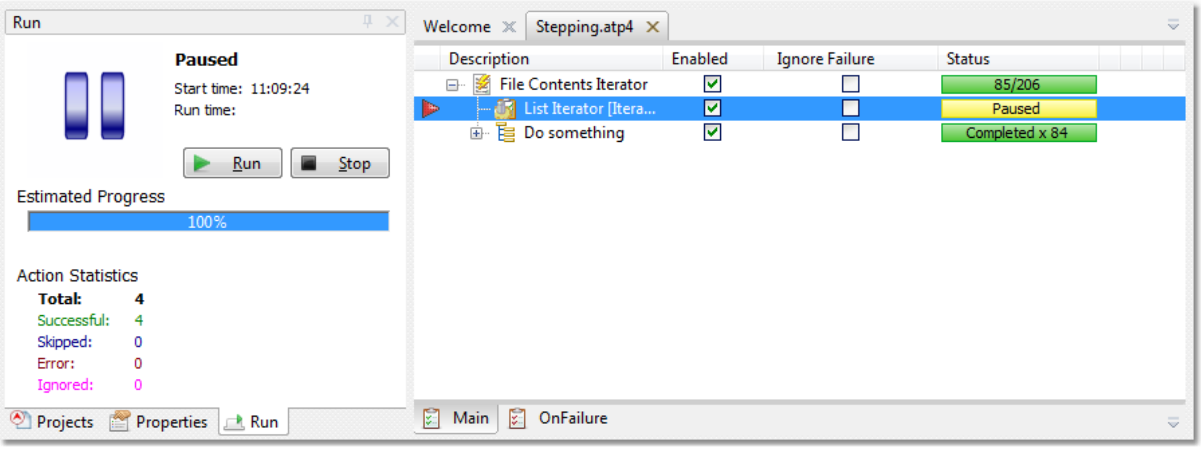
Description	Enabled	Ignore Failure	Status
File Contents Iterator	☑	□	Iteration 1 of 44
List Iterator [Iterator vari...	☑	□	
Do something	☑	□	

Notice that the List Iterator action has a breakpoint. The run will pause when it reaches this List Iterator action.

How to step through your project

The following screen shot shows a paused project. To step through:

- Click the toolbar icon; or
- Press the F8 (or F10 with the Microsoft style bindings) key; or
- Choose Step from the Run menu.



Paused
Start time: 11:09:24
Run time:

Estimated Progress
100%

Action Statistics

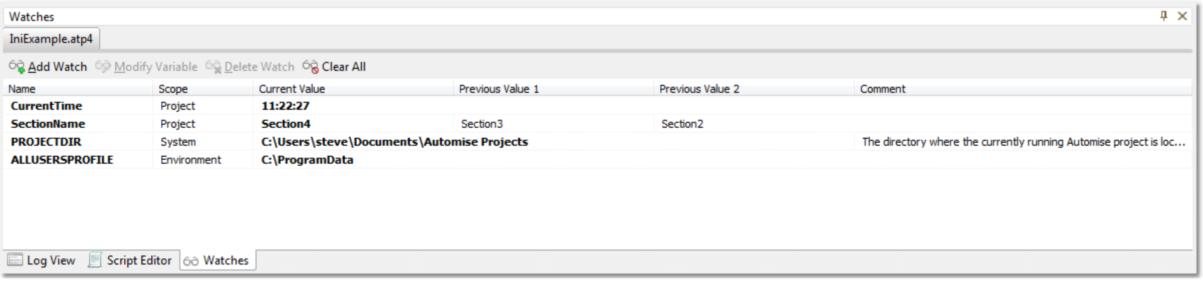
Total:	4
Successful:	4
Skipped:	0
Error:	0
Ignored:	0

Description	Enabled	Ignore Failure	Status
File Contents Iterator	☑	□	85/206
List Iterator [Itera...	☑	□	Paused
Do something	☑	□	Completed x 84

See Watches and Live Logging for more information on debugging projects at runtime.

3.3.1 Watches

Watches let you watch and modify variable values during a run. They assist you in debugging, so that you can step over actions and see the current and previous values for the specified variables.



Name	Scope	Current Value	Previous Value 1	Previous Value 2	Comment
CurrentTime	Project	**11:22:27**			
SectionName	Project	**Section4**	Section3	Section2	
PROJECTDIR	System	**C:\Users\steve\Documents\Automise Projects**			The directory where the currently running Automise project is loc...
ALLUSERSPROFILE	Environment	**C:\ProgramData**			

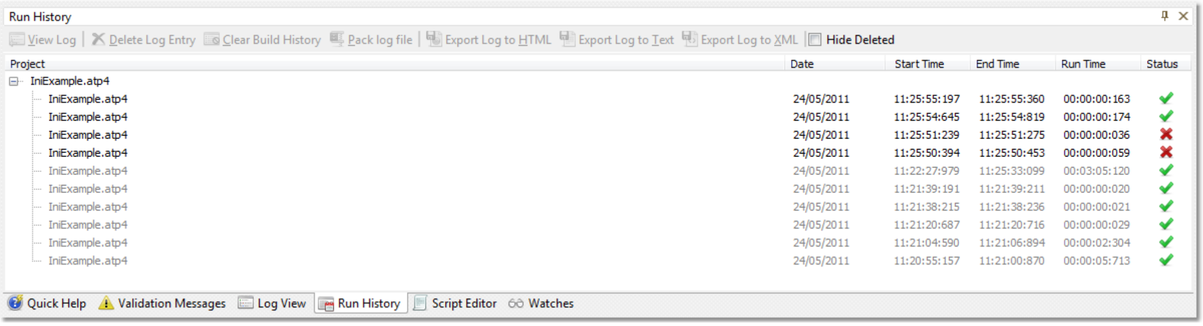
To add a new watch, click on the Add Watch button or drag a variable from the Project tab.

To modify a watch variable, double click its entry or click the Modify Variable button. You can change the current variable "on the fly", even while the run is paused.

3.4 History & Statistics

The Run History Tab displays a summary of the previous runs of the current project.

You can load the details of a previous run into the Log tab to see the details of that run.



Project	Date	Start Time	End Time	Run Time	Status
IniExample.atp4					
IniExample.atp4	24/05/2011	11:25:55:197	11:25:55:360	00:00:00:163	✓
IniExample.atp4	24/05/2011	11:25:54:645	11:25:54:819	00:00:00:174	✓
IniExample.atp4	24/05/2011	11:25:51:239	11:25:51:275	00:00:00:036	✗
IniExample.atp4	24/05/2011	11:25:50:394	11:25:50:453	00:00:00:059	✗
IniExample.atp4	24/05/2011	11:22:27:979	11:25:33:099	00:03:05:120	✓
IniExample.atp4	24/05/2011	11:21:39:191	11:21:39:211	00:00:00:020	✓
IniExample.atp4	24/05/2011	11:21:38:215	11:21:38:236	00:00:00:021	✓
IniExample.atp4	24/05/2011	11:21:20:687	11:21:20:716	00:00:00:029	✓
IniExample.atp4	24/05/2011	11:21:04:590	11:21:06:894	00:00:02:304	✓
IniExample.atp4	24/05/2011	11:20:55:157	11:21:00:870	00:00:05:713	✓

The History keeps a record for every run of the project, but only a finite number of detailed logs are kept. The maximum number of detailed logs can be changed in the Options screen. When the detailed log has been deleted (either automatically or manually), the entry for that log will be grayed out and you can no longer view the details of the log.

View Log

Loads the log in the Run Log tab of the Design and Debug screen.

Delete Log Entry

Deletes the log permanently.

Clear History

Deletes all stored logs permanently.

Pack log file

Compresses the stored log file to save space

Export Log

Exports the selected log to HTML, Text, or XML

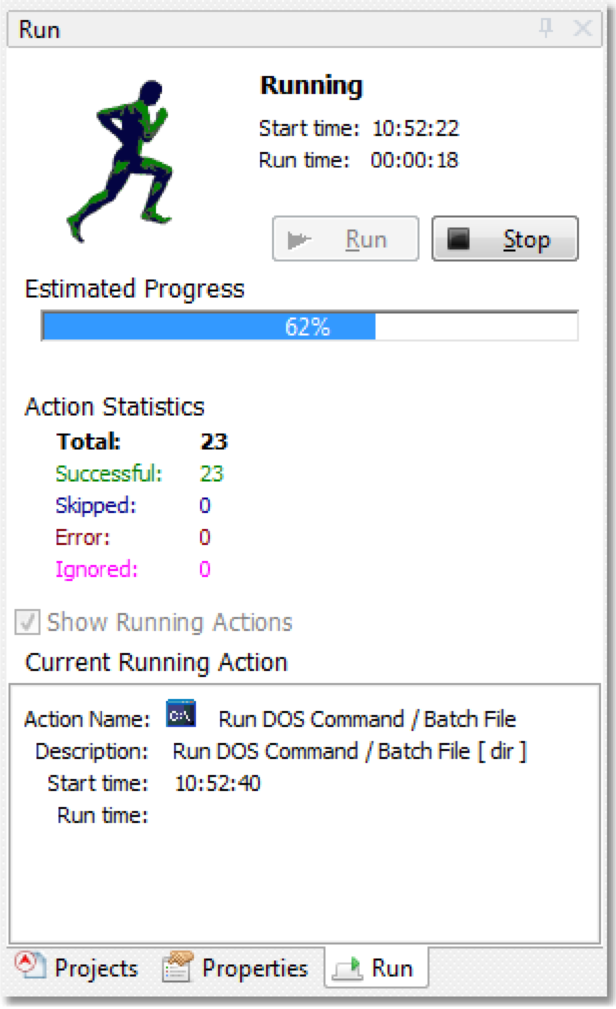
Hide Deleted

If unchecked, deleted logs are displayed in grey. Otherwise, they are hidden.

See also: Logging Options

3.5 IDE Running Mode

When Running, the Run Tab is shown to provide information about the currently executing action and the progress of the project.



Running

Start time: 10:52:22

Run time: 00:00:18

Estimated Progress

62%

Action Statistics

Total:	**23**
Successful:	23
Skipped:	0
Error:	0
Ignored:	0

☑ Show Running Actions

Current Running Action

Action Name: Run DOS Command / Batch File

Description: Run DOS Command / Batch File [dir]

Start time: 10:52:40

Run time:

To see the logs from previous runs, use the History & Statistics tab.

3.5.1 Running a Project

When you have created your process (and maybe debugged it), you can then start the run in a number of ways:

- **Run**: starts on the first action in the Main action list and runs until an unhandled error or all actions in Main are executed

- **Continue from selected** action without resetting the log. This option is only enabled when the previous run ended in error. Restarting the project will append to the existing log instead of starting a new log entry.

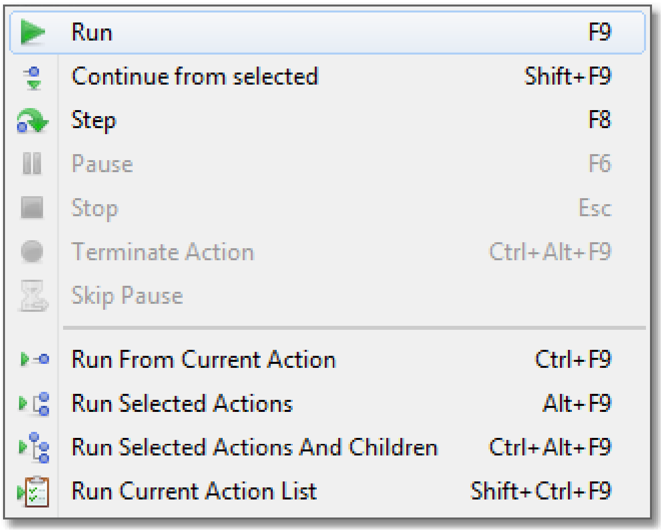
- **Run from Current Action** - the project starts from the currently selected action and runs to the end of the current action list.

- **Run Selected Actions** - runs just the currently selected actions.

- **Run Selected Actions and Children** - runs the currently selected actions and any child actions.

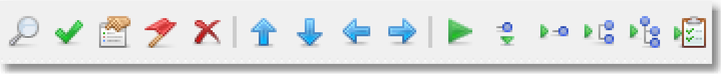
The Run commands can be accessed in the following ways:

1. The Run menu:


Run F9
Continue from selected Shift+F9
Step F8
Pause F6
Stop Esc
Terminate Action Ctrl+Alt+F9
Skip Pause
Run From Current Action Ctrl+F9
Run Selected Actions Alt+F9
Run Selected Actions And Children Ctrl+Alt+F9
Run Current Action List Shift+Ctrl+F9

2. Shortcut keys (eg. F9 for Run)

3. Toolbar



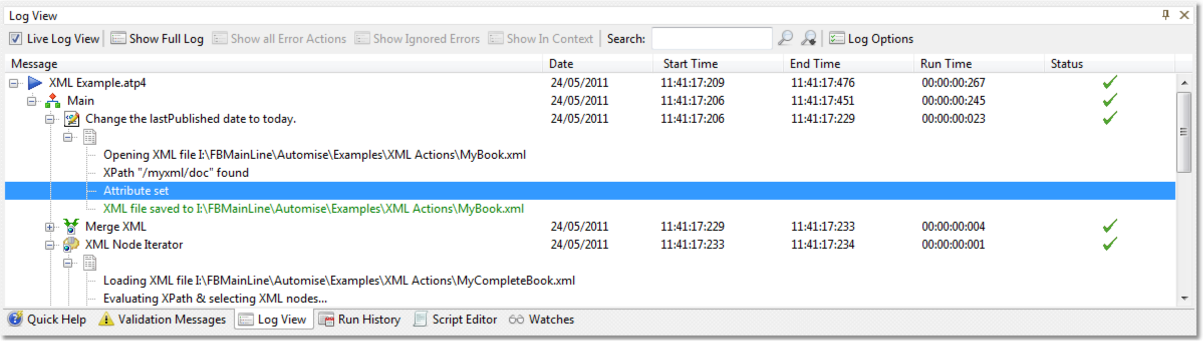
4. Right-click context menu of an action.

In addition to these run commands, you may want to step, pause or stop the project using the following commands:

- **Step** - If the run is paused, it will run the next action and pause. If the run hasn't started, it will run the first action in Main and then pause.
- **Pause** - The project will pause after the current action completes.
- **Stop** - The project will stop after the current action completes. Some actions will stop prematurely if Stop is pressed, but most will not.
- **Terminate** - Automise will attempt to terminate the currently running action, and immediately stop the project. Terminate is a more abrupt version of Stop. Some actions don't support being terminated.
- **Continue** - The project will continue running if it's in the paused state

3.5.2 Run Log

The Run Log tab is a tree which contains nodes representing the actions that have been executed, with any output from these actions.



Message	Date	Start Time	End Time	Run Time	Status
XML Example.atp4	24/05/2011	11:41:17:209	11:41:17:476	00:00:00:267	✓
Main	24/05/2011	11:41:17:206	11:41:17:451	00:00:00:245	✓
Change the lastPublished date to today.	24/05/2011	11:41:17:206	11:41:17:229	00:00:00:023	✓
Opening XML file I:\FBMainLine\Automise\Examples\XML Actions\MyBook.xml					
XPath "/myxml/doc" found					
Attribute set					
XML file saved to I:\FBMainLine\Automise\Examples\XML Actions\MyBook.xml					
Merge XML	24/05/2011	11:41:17:229	11:41:17:233	00:00:00:004	✓
XML Node Iterator	24/05/2011	11:41:17:233	11:41:17:234	00:00:00:001	✓
Loading XML file I:\FBMainLine\Automise\Examples\XML Actions\MyCompleteBook.xml					
Evaluating XPath & selecting XML nodes...					

Status

The status Column displays an icon representing the current status of an action :

◷ Running: The action is currently running

⬇ Skipped: The Action was skipped, either because the Condition was not met, or because the SkipAction parameter in the BeforeAction event handler was set to True

✓ Completed: The Action completed successfully

✗ Error: The action failed. The run will now terminate after running the OnFailure action list, unless there is a surrounding try/catch block.

▬ Error Ignored - An Error occurred, but the Ignore Failure property was set to True.

You can copy the text from the log text nodes by selecting the node and pressing Ctrl+C. To control the text generated for a node, you can set the "Action Log Title" property in the Properties Tab.

Clicking on an item in the log will select the corresponding action in the Automise IDE. To show the logs for a particular action, right-click and choose "Filter Log By Action".

Live Log View

When a project is executing, you can display the log in real time. However, this can be very CPU intensive so for better performance, you can switch live logging off.

- Live logging off: the log is updated as normal, but only displayed once the project stops.
- Live logging on: the log is displayed with every change.

Live Logging can be switched on and off while the project is running.

If Live Logging is switched on while a project is running, then the full project hierarchy (as seen above) is not available. Only actions which have run since live logging was enabled are shown, and the tree hierarchy does not always represent the hierarchical structure of the project. To see the complete hierarchy, it is necessary to wait until the project has completed.

You can set a default setting for Live Logging on the Logging Options page.

Show Full Log

Displays the complete log, if you had selected one of the filtering options below.

Show all Error Actions

Click to Show all Error Actions button to show only the actions which failed with an Error status (✖). This includes actions which failed as part of a Try/Catch block.

Show Ignored Errors

Click to show only errors which were ignored (▬) due to the Ignore Failure property.

Show In Context

When selecting a node using "Show all Error Actions" or "Show Ignored Errors", click "Show In Context" to show the full log, focused around the error.

Search

Shows only actions containing a certain string. Lines of output containing that string are highlighted.

Search options

- Search action names and messages: (default) Shows actions where either the title of the action or the output produced contains the string
- Search action names: Shows only actions where the title of the action contains the string.
- Search messages: Shows only actions where the string was found in the output produced by the action

Press Enter or click the search button to apply the search filter. Click the "Show Full Log" button to clear it.

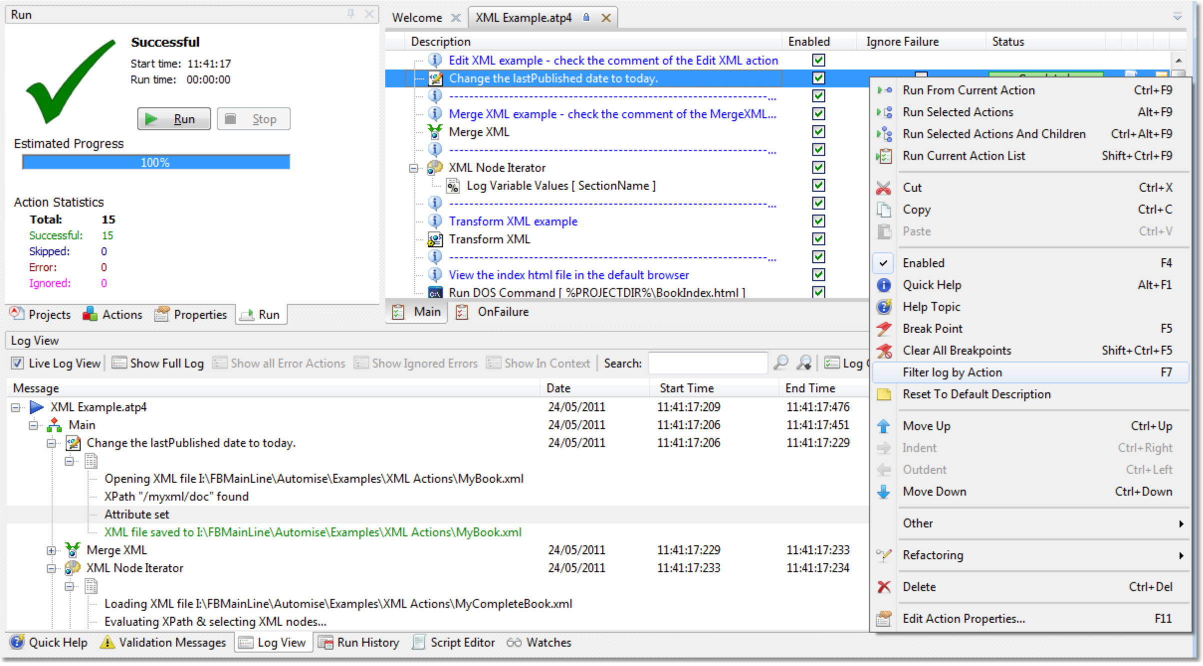
Note: searching action names or messages is equivalent to adding "action:" or "message" to the front of the search query.

Log Options

Brings up the Logging section of the Options Dialog where you can set options such as which types of actions appear in the log, and whether hierarchical action messages are shown flat or not.

Filter Log by Action

To only see log steps corresponding to a particular action, right click the action and select **Filter Log By Action**. If the action occurred multiple times in the run, multiple entries will be shown in the log. To unfilter the log, click **Show Full Log**.



Action Statistics

Total:	15
Successful:	15
Skipped:	0
Error:	0
Ignored:	0

Description	Enabled	Ignore Failure	Status
Edit XML example - check the comment of the Edit XML action	☑		
Change the lastPublished date to today.	☑		
---...	☑		
Merge XML example - check the comment of the MergeXML...	☑		
Merge XML	☑		
---...	☑		
XML Node Iterator	☑		
Log Variable Values [SectionName]	☑		
---...	☑		
Transform XML example	☑		
Transform XML	☑		
---...	☑		
View the index html file in the default browser	☑		
Run DOS Command [%PROJECTDIR%\BookIndex.html]	☑		

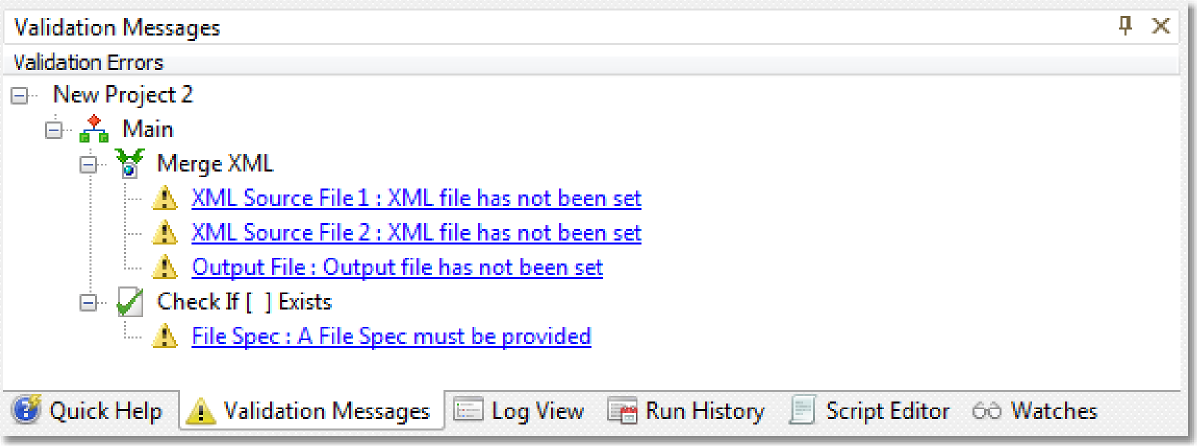
Message	Date	Start Time	End Time
XML Example.atp4	24/05/2011	11:41:17:209	11:41:17:476
Main	24/05/2011	11:41:17:206	11:41:17:451
Change the lastPublished date to today.	24/05/2011	11:41:17:206	11:41:17:229
Opening XML file I:\FBMainLine\Automise\Examples\XML Actions\MyBook.xml			
XPath "/myxml/doc" found			
Attribute set			
XML file saved to I:\FBMainLine\Automise\Examples\XML Actions\MyBook.xml			
Merge XML	24/05/2011	11:41:17:229	11:41:17:233
XML Node Iterator	24/05/2011	11:41:17:233	11:41:17:234
Loading XML file I:\FBMainLine\Automise\Examples\XML Actions\MyCompleteBook.xml			
Evaluating XPath & selecting XML nodes...			

3.5.3 Validation

Validation is an optional step that checks that all actions in the project appear to have been set up properly. The validation tab then appears, displaying any validation errors in the current project or the action just edited.

Validation can be triggered in the following ways:

- When a project starts (see Tools->Options->General Options->Variables and Validation.)
- After an action has been edited with the property dialog.
- Manually (Project->Validate Project menu.)
- When ATCMD is started with the /C option set.



Validation Errors

- New Project 2
  - Main
    - Merge XML
      - XML Source File 1 : XML file has not been set
      - XML Source File 2 : XML file has not been set
      - Output File : Output file has not been set
    - Check If [ ] Exists
      - File Spec : A File Spec must be provided

Clicking on a validation error opens the corresponding action in your project for editing.

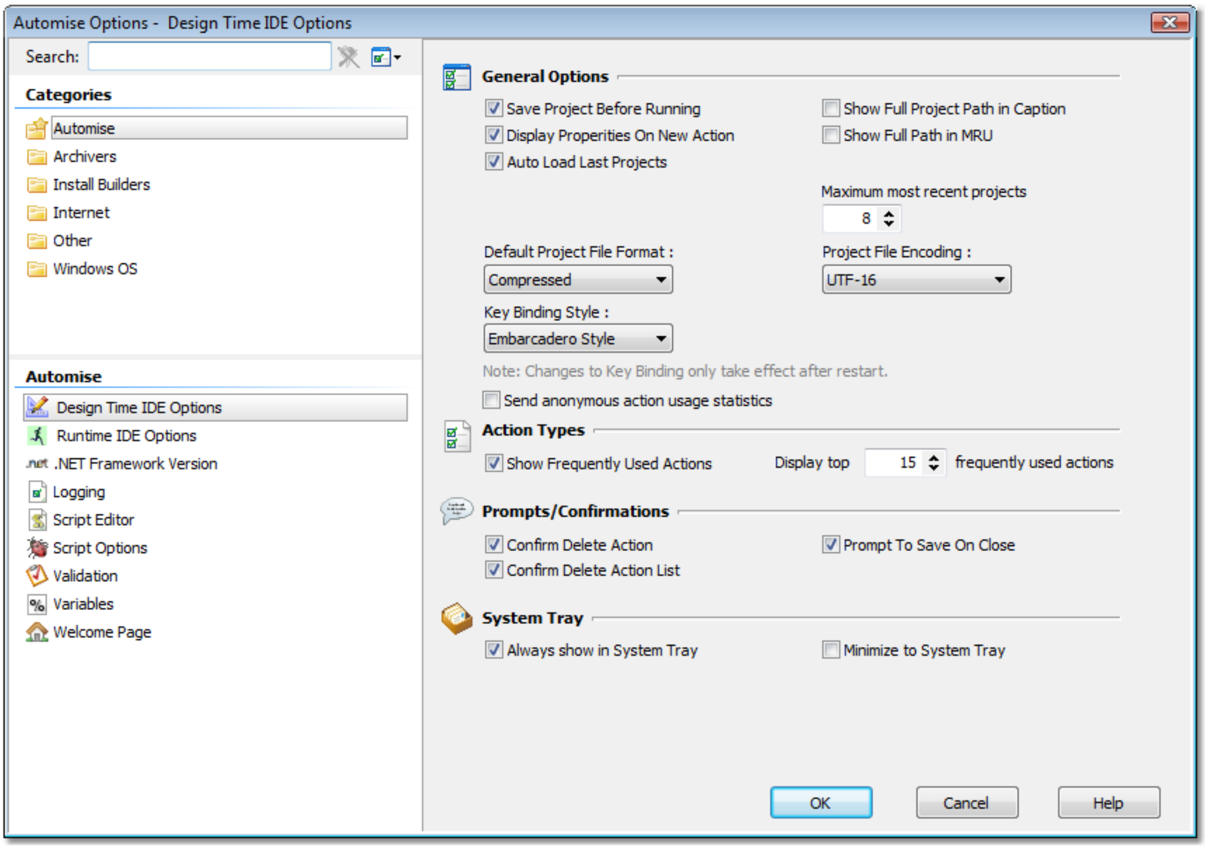
Validation does not prevent the project from being saved, but will prevent the project from being run. You can turn off pre-run validation in the Validation Options page.

3.6 IDE Features

3.6.1 Options Dialog

The options dialog allows you to set general Automise options and preferences, as well as options for some action types.

Options are grouped into categories on the left. You can search for options using the search box, and alter the orientation and sort order using the option button. See Actions Tab.



General Options

☑ Save Project Before Running
☑ Display Properities On New Action
☑ Auto Load Last Projects
☐ Show Full Project Path in Caption
☐ Show Full Path in MRU

Maximum most recent projects: 8

Default Project File Format : Compressed

Project File Encoding : UTF-16

Key Binding Style : Embarcadero Style

Note: Changes to Key Binding only take effect after restart.

☐ Send anonymous action usage statistics

Action Types

☑ Show Frequently Used Actions   Display top 15 frequently used actions

Prompts/Confirmations

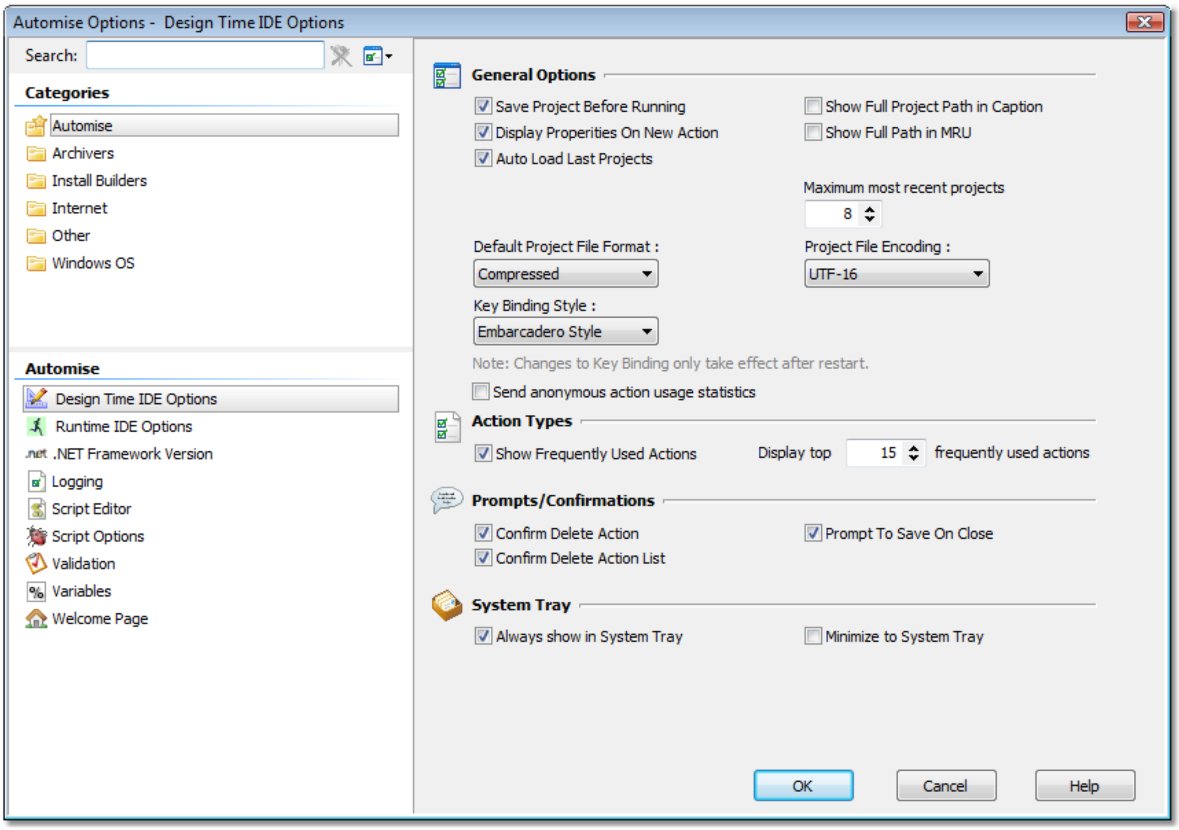
☑ Confirm Delete Action
☑ Confirm Delete Action List
☑ Prompt To Save On Close

System Tray

☑ Always show in System Tray
☐ Minimize to System Tray

See also:
Design Time Options
Runtime Options
Validation Options
Variables Options
Logging Options
Scripting Debugger Options
Scripting Editor Options

3.6.1.1 Design Time Options



Automise Options - Design Time IDE Options

Search:

Categories

- Automise
- Archivers
- Install Builders
- Internet
- Other
- Windows OS

Automise

- Design Time IDE Options
- Runtime IDE Options
- .NET Framework Version
- Logging
- Script Editor
- Script Options
- Validation
- Variables
- Welcome Page

General Options

☑ Save Project Before Running
☑ Display Properities On New Action
☑ Auto Load Last Projects
□ Show Full Project Path in Caption
□ Show Full Path in MRU

Maximum most recent projects: 8

Default Project File Format : Compressed

Project File Encoding : UTF-16

Key Binding Style : Embarcadero Style

Note: Changes to Key Binding only take effect after restart.

□ Send anonymous action usage statistics

Action Types

☑ Show Frequently Used Actions Display top 15 frequently used actions

Prompts/Confirmations

☑ Confirm Delete Action
☑ Confirm Delete Action List
☑ Prompt To Save On Close

System Tray

☑ Always show in System Tray
□ Minimize to System Tray

General Options

Save Project Before Running

If this option is selected, the currently open project will be saved before it is run. If the project has not yet been saved, or cannot be saved in its current location, a "Save As..." dialog will be opened.

Display Properties On New Action

If this option is selected, the Action Property pages dialog will open automatically whenever a new action is added to the project.

Auto Load Last Project

If this option is selected, the last loaded project will open automatically whenever Automise starts up. If it is not selected, a new (empty) project will be opened.

Show Full Project Path in Caption

If this option is set, the Automise IDE's window title will be the full path to the project file instead of just the file name.

Show Full Path in MRU

Similar to the previous option, if this option is set then the File -> Recent menu will show full project file paths instead of just file names.

Maximum most recent projects

You can choose the maximum number of projects to be shown under the File -> Recent menu.

Key Binding Style

With "Microsoft Style" key bindings, F5 runs the project, F9 toggles breakpoints, and F10 steps. With "Embarcadero Style" key bindings, F9 runs, F5 toggles breakpoints, F8 steps, and so forth. When changing this option, you must restart Automise for the change to take effect.

Default Project File Format

If "compressed", projects are saved in the compressed .atz4 format. Otherwise they are saved in the uncompressed .atp4 format by default.

Project File Encoding

Selects the encoding that Automise will use when saving the project file.

Send anonymous action usage statistics

See Automatic Action Usage Updates.

Action Types

Show Frequently Used Actions

If this option is selected, then the "Frequently Used" action type category will appear in the action types list. You can also set how many actions to show on this list.

Prompts/Confirmations

Confirm Delete Action / Action List

Select whether you wish to see a confirmation dialog before deleting actions/ action lists.

Prompt to Save on Close / New Project

Select whether you wish to see a "Save Project First?" confirmation dialog before closing an open project, or starting a new project. Note that disabling these options can easily result in loss of data.

Show Tip of the Day on Startup

Enable this option to see a "Tip of the day" on the Welcome Tab when you start Automise each day.

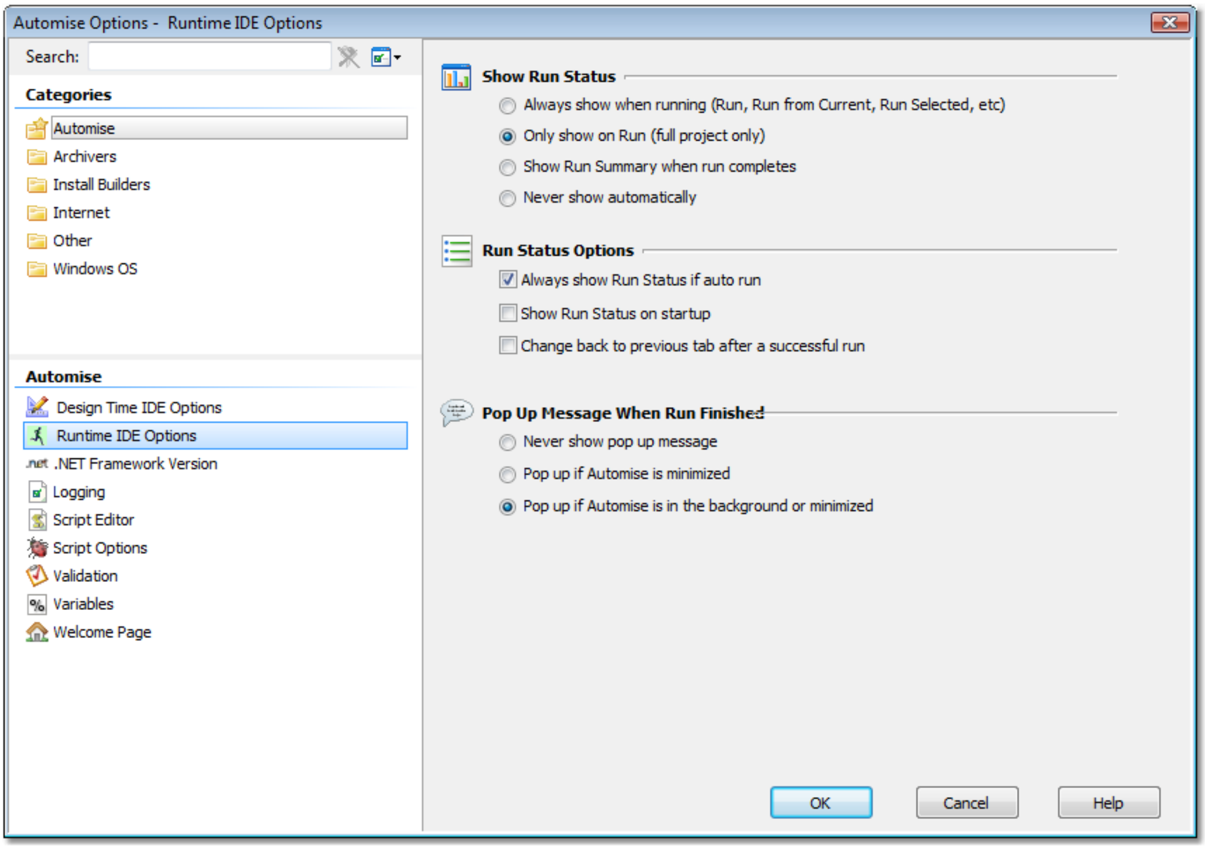
System Tray

Automise can show an icon in the system tray. You can choose whether to not show the system tray icon, or to only show it when Automise is minimized. Double-clicking on the system tray icon will restore the Automise IDE as the current focused application.

Minimize to System Tray

Causes Automise to be minimize to the system tray rather than the taskbar when minimized.

3.6.1.2 Runtime Options



Automise Options - Runtime IDE Options

Search:

Categories

- Automise
- Archivers
- Install Builders
- Internet
- Other
- Windows OS

Automise

- Design Time IDE Options
- Runtime IDE Options
- .NET Framework Version
- Logging
- Script Editor
- Script Options
- Validation
- Variables
- Welcome Page

Show Run Status

○ Always show when running (Run, Run from Current, Run Selected, etc)
● Only show on Run (full project only)
○ Show Run Summary when run completes
○ Never show automatically

Run Status Options

☑ Always show Run Status if auto run
□ Show Run Status on startup
□ Change back to previous tab after a successful run

Pop Up Message When Run Finished

○ Never show pop up message
○ Pop up if Automise is minimized
● Pop up if Automise is in the background or minimized

Show Run Status

Automatically switches to the Run Tab in certain situations:

- Anytime a project is running
- Only when a project is run from scratch ("Run" menu item)
- When the project completes
- Never

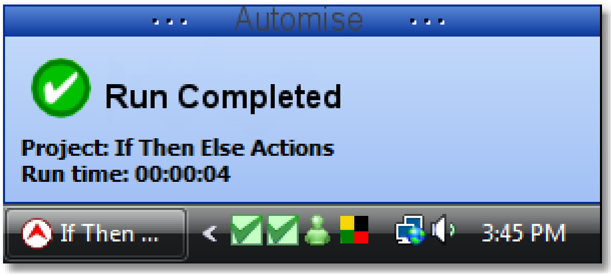
Run Status Options

- **Always show Run Status if auto Run**: if running from the command line, shows the Run Tab.
- **Show Run Status on startup**: automatically chooses the Run Tab when Automise is first run.

- **Change back to previous tab after a successful Run:** if the Run Tab is automatically chosen for a project which completes successfully, returns to the previously selected tab.

Pop Up Message When Run Ends

Automise can pop up a message notifier above the system tray, whenever a project stops running:



Run Completed

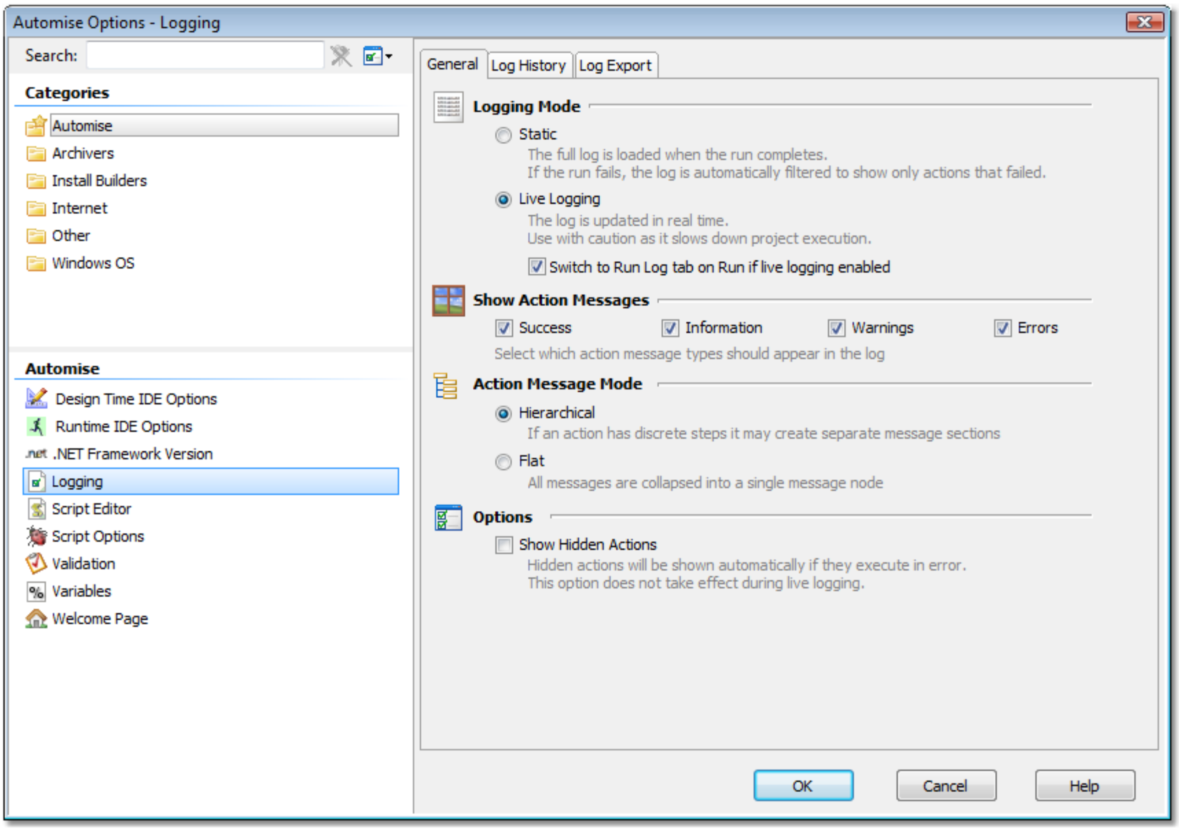
Project: If Then Else Actions
Run time: 00:00:04

The message disappears when you click on it, or on the Automise IDE window.

You can choose to popup this message only if the Automise IDE is minimized, or whenever Automise is either minimized or in the background (ie not the current focused application.)

3.6.1.3 Logging Options

General Options



Logging Mode

○ Static
The full log is loaded when the run completes.
If the run fails, the log is automatically filtered to show only actions that failed.

● Live Logging
The log is updated in real time.
Use with caution as it slows down project execution.

☑ Switch to Run Log tab on Run if live logging enabled

Show Action Messages

☑ Success ☑ Information ☑ Warnings ☑ Errors

Select which action message types should appear in the log

Action Message Mode

● Hierarchical
If an action has discrete steps it may create separate message sections

○ Flat
All messages are collapsed into a single message node

Options

□ Show Hidden Actions
Hidden actions will be shown automatically if they execute in error.
This option does not take effect during live logging.

Overview

Each project has its own log file that is automatically created if it doesn't exist when the project is created or opened. The Log File is a database which stores detailed information of previous projects and also a summary record of all past projects for the particular project. It is important to decide how many detailed logs you want to store in the log file, as this makes a significant difference to the size of the log file on disk if you have a large project. There are three pages of logging options.

Logging Mode

By default, **Static** logging is used. This means that when a project is run, all events are written to the project's log file, but they are only displayed on the Run Log tab when the project stops.

Alternatively, **Live logging** means that as each event occurs, the Run Log tab is updated. This is useful for debugging, but is much slower, especially when running large numbers of quick actions.

- Switch to Run Log tab on Run if live logging enabled: Automatically switches to the Run Log tab when performing a run, if using live logging. This is a general convenience for you.

Show Action Messages

These options filter the amount of information displayed in the log. All information that the action generates is always stored in the log. Each action is responsible for tagging a status message with either Success, Information, Warning or Error. It's possible that some actions may not correctly tag the message, especially actions that rely on a 3rd party executables as it can be difficult to detect if certain command line output are errors or informational for example.

- Success: Action messages with status of Success are displayed in the log. Success messages are displayed in green.
- Information: Action messages with status of Information are displayed in the log. Informational messages are displayed in black.
- Warnings: Action messages with status of Waning are displayed in the log. Warning messages are displayed in orange.
- Errors: Action messages with status of Error are displayed in the log. Error messages are displayed in red. You should normally leave this enabled.

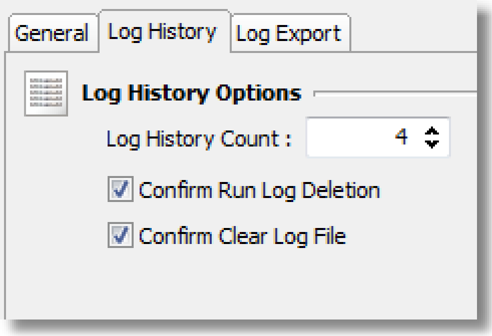
Action Message Mode

- Hierarchical: Text from actions can be shown in a tree structure, representing different kinds of information such as output, text created by the action itself etc.
- Flat: All text from an action is displayed in a single flat list.

Options

Show Hidden Actions: Displays "hidden" actions in the log if they execute in error. See Action logging properties for information on hidden actions.

Log History Options



Log History Options

Log History Count : 4

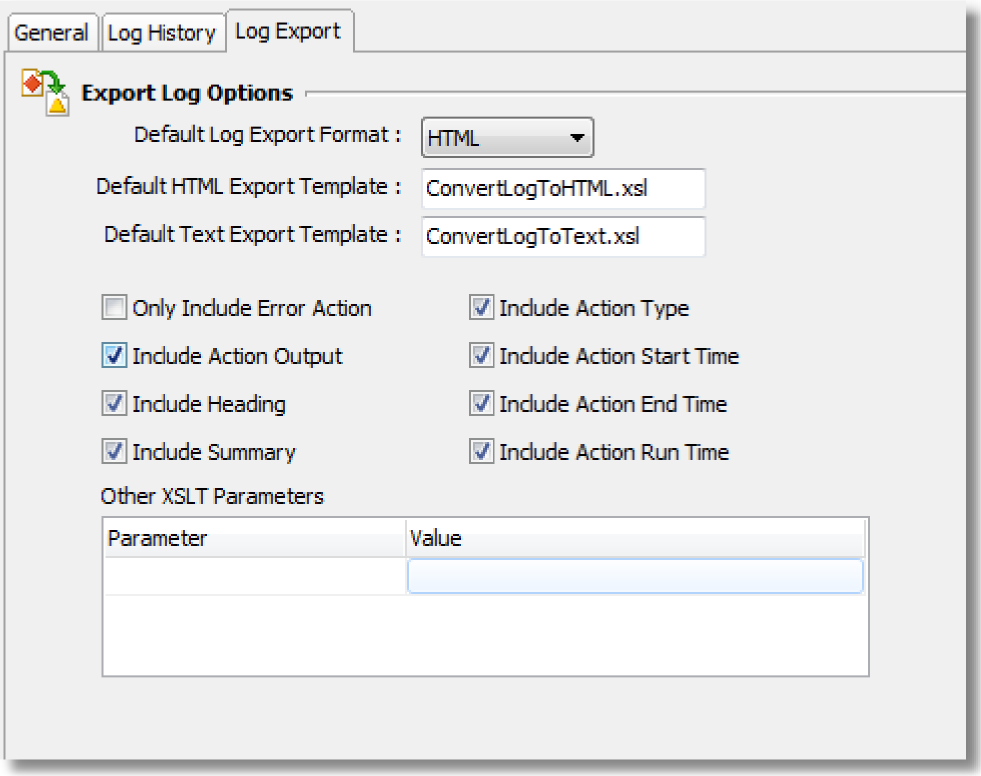
☑ Confirm Run Log Deletion

☑ Confirm Clear Log File

- Log History Count: The maximum number of detailed logs to store in the log file. Increasing this number gives you more history, but a larger log file.
- Confirm Run Log Deletion: A confirmation dialog is shown if a log is manually deleted in the Run History.
- Confirm Clear Log File: A confirmation dialog is shown if the Clear History option is pressed in the Run History.

Export Log Options

These options control how logs are exported from the History & Statistics tab. They are equivalent to the options on the Export Log Action.



Export Log Options

Default Log Export Format : HTML

Default HTML Export Template : ConvertLogToHTML.xsl

Default Text Export Template : ConvertLogToText.xsl

□ Only Include Error Action

☑ Include Action Output

☑ Include Heading

☑ Include Summary

☑ Include Action Type

☑ Include Action Start Time

☑ Include Action End Time

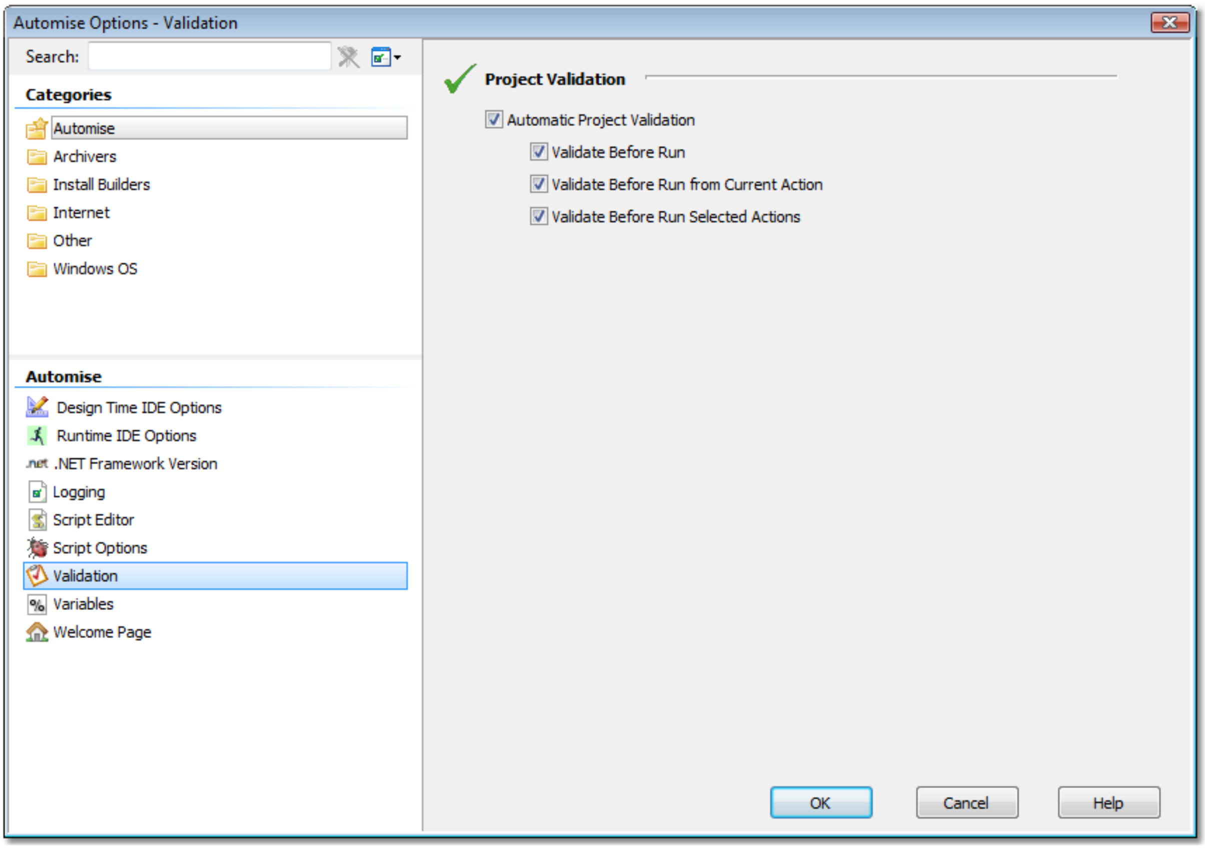
☑ Include Action Run Time

Other XSLT Parameters

Parameter	Value

- Default Log Export Format: choose the format for exporting the log: text, xml or html
- Templates: The template to use. Some templates are located in: <Install Directory>\Stylesheet
- Options: The level of detail to include in the exported log.
- Other XSLT Parameters: if you are using your own template stylesheet, then you can provide extra XSLT parameters to control the transformation.

3.6.1.4 Validation Options



Project Validation

☑ Automatic Project Validation

☑ Validate Before Run

☑ Validate Before Run from Current Action

☑ Validate Before Run Selected Actions

When **Automatic Project Validation** is set, Automise automatically performs some simple validation when running a project, avoiding failed runs caused by simple, detectable errors.

Three options are available:

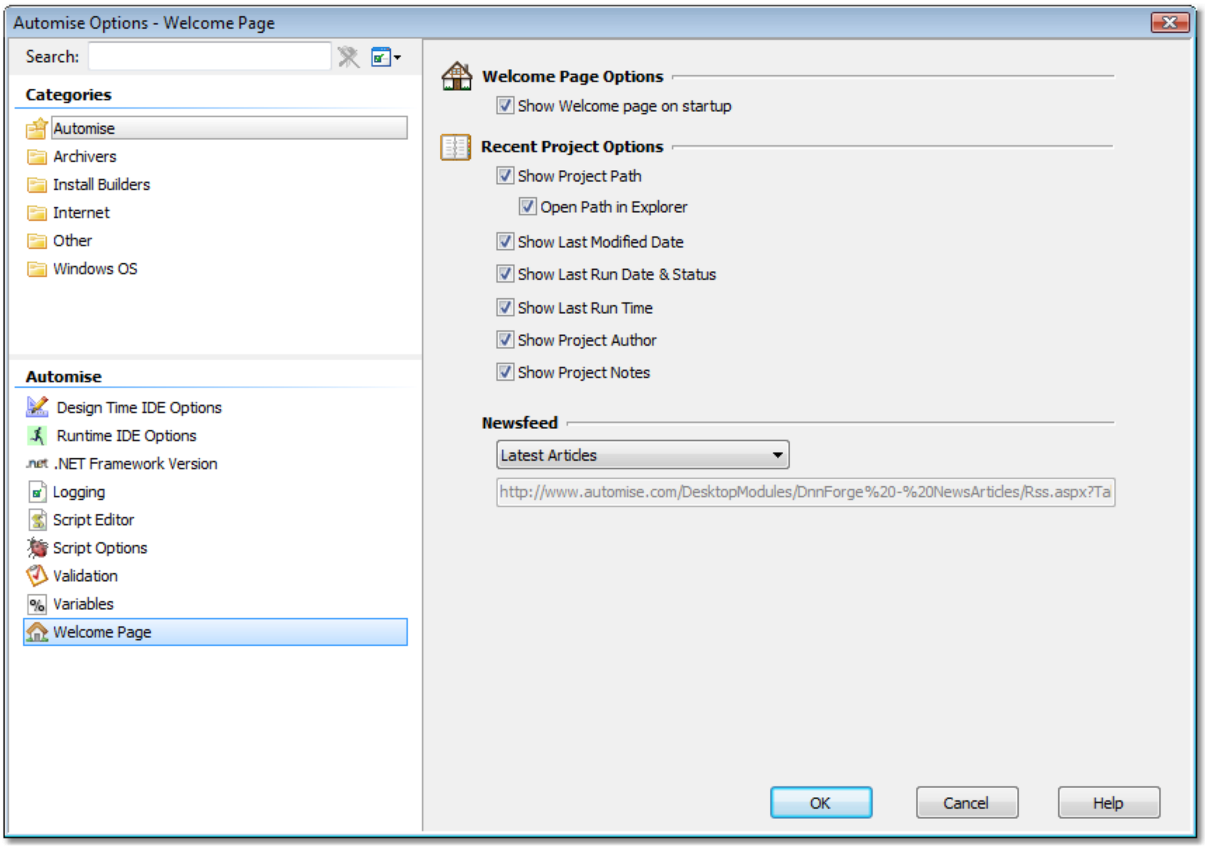
- Validate Before Run
- Validate Before Run from Current Action
- Validate Before Run Selected Actions

See Running a Project for more information on the difference between these types of runs.

Normally, validation should be turned on. You may wish to selectively turn it off if you need to run part of a project while another part is incomplete.

For more information about what errors validation can detect, see Project Validation.

3.6.1.5 Welcome Page Options



Welcome Page Options

☑ Show Welcome page on startup

Recent Project Options

☑ Show Project Path

☑ Open Path in Explorer

☑ Show Last Modified Date

☑ Show Last Run Date & Status

☑ Show Last Run Time

☑ Show Project Author

☑ Show Project Notes

Newsfeed

Latest Articles

http://www.automise.com/DesktopModules/DnnForge%20-%20NewsArticles/Rss.aspx?Ta

The Welcome Page options control which elements are shown on the Welcome Page.

Welcome Page Options

Sets whether the Welcome tab is selected when Automise is launched. If not selected, the Design & Debug tab is selected instead.

Recent Project Options

Controls which is information is shown in the **Recent Projects** pane of the Welcome Page. Turning off some options allows more recent projects to be listed without having to scroll.

Newsfeed

Controls which RSS feed is displayed. You can choose "custom" to add any RSS feed at all, or to turn off the newsfeed.

3.6.2 Automatic Action Usage Updates

What Are Automatic Usage Updates?

In order to continue making Automise better, we're interested to know what actions users are working with the most.

In order to gather this information, we ask that you please allow Automise to send monthly updates of which actions have been added to projects. This information is all totally anonymous. Only the following data is sent, in plain text:

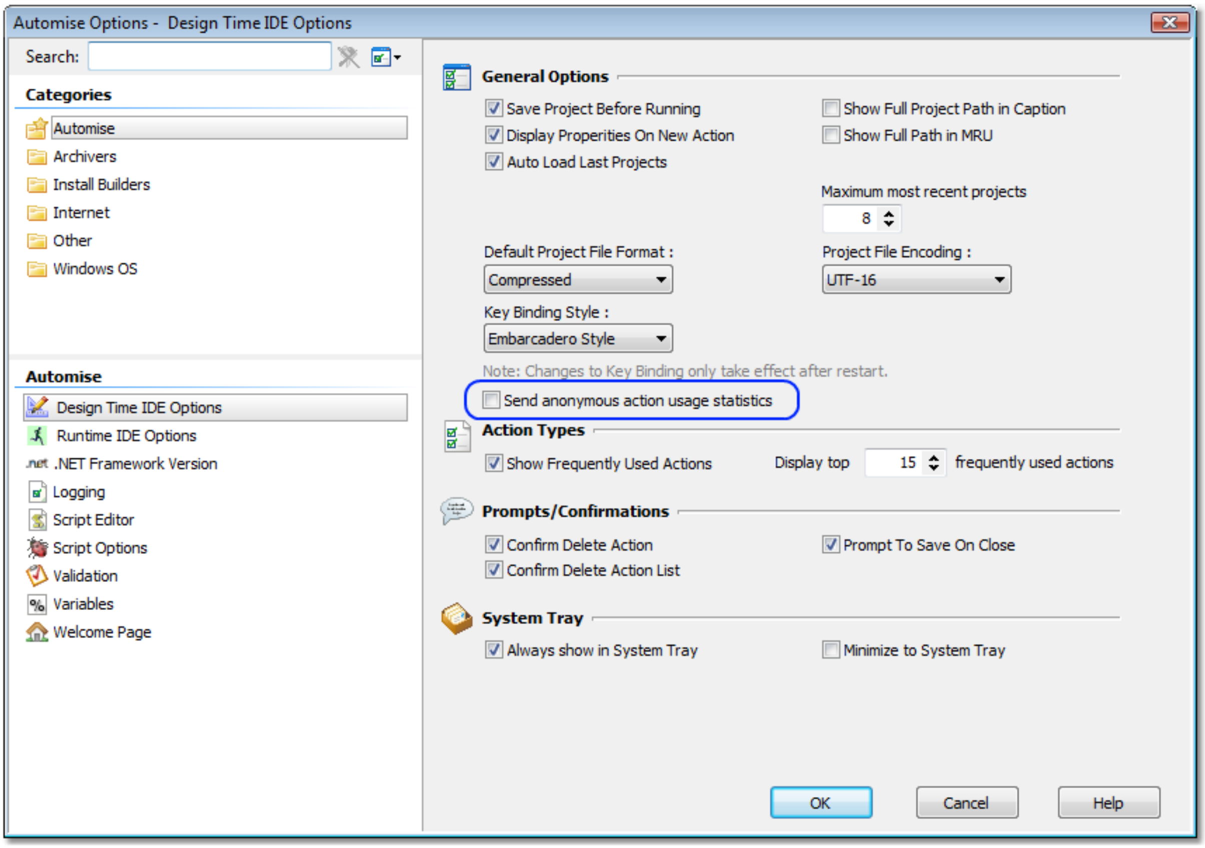
- The version of Automise that you are using.
- The names (and total numbers) of actions which were added to projects.

No identifying information of any kind is recorded, not even your IP address. None of your action settings are submitted, only action names (as they are displayed in the Action Types frame.)

If you have a registered copy of Automise, then you will be prompted and asked for permission to start submitting this information. No further dialogs will be displayed, unless an error occurs (see below.)

Enabling and Disabling Automatic Updates

Automatic update submission is disabled by default. If you want to enable or disable update submission at any time, you can go to Tools -> Options and click on "Automise" and then "General IDE Options":



Automise Options - Design Time IDE Options

Search:

Categories

Automise
Archivers
Install Builders
Internet
Other
Windows OS

Automise

Design Time IDE Options
Runtime IDE Options
.NET Framework Version
Logging
Script Editor
Script Options
Validation
Variables
Welcome Page

General Options

☑ Save Project Before Running
☑ Display Properities On New Action
☑ Auto Load Last Projects
□ Show Full Project Path in Caption
□ Show Full Path in MRU

Maximum most recent projects: 8

Default Project File Format : Compressed

Project File Encoding : UTF-16

Key Binding Style : Embarcadero Style

Note: Changes to Key Binding only take effect after restart.

□ Send anonymous action usage statistics

Action Types

☑ Show Frequently Used Actions   Display top 15 frequently used actions

Prompts/Confirmations

☑ Confirm Delete Action
☑ Confirm Delete Action List
☑ Prompt To Save On Close

System Tray

☑ Always show in System Tray
□ Minimize to System Tray

How Updates are Submitted

Updates are submitted by using HTTP to connect to our web site. If a proxy server is configured in Internet Explorer, then this proxy information will be used for the connection (proxy authentication is not supported.)

If an error occurs, a dialog will be displayed. Update submission will then be re-attempted the next time Automise is launched. If you are consistently receiving errors about failed submissions, then you may need to consider disabling the automatic usage updates feature.

3.6.3 Check for Updates

Once a week Automise prompts you to check for any updates by querying the Automise website. No personal user information is sent during this process.

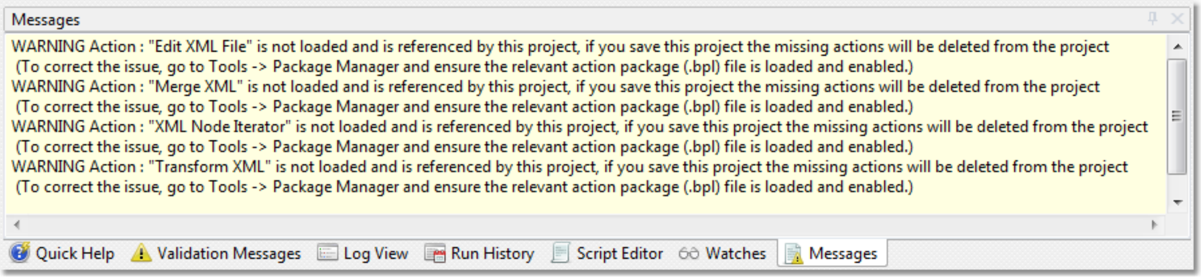
If an update is available, the Quick Help tab will display the details of the update and provide a link to any updated files and further instructions.

You can manually check for updates by choosing the "Check for Updates" menu item in the Help menu.

3.6.4 Messages

The Messages tab only appears in order to display important information requiring your attention.

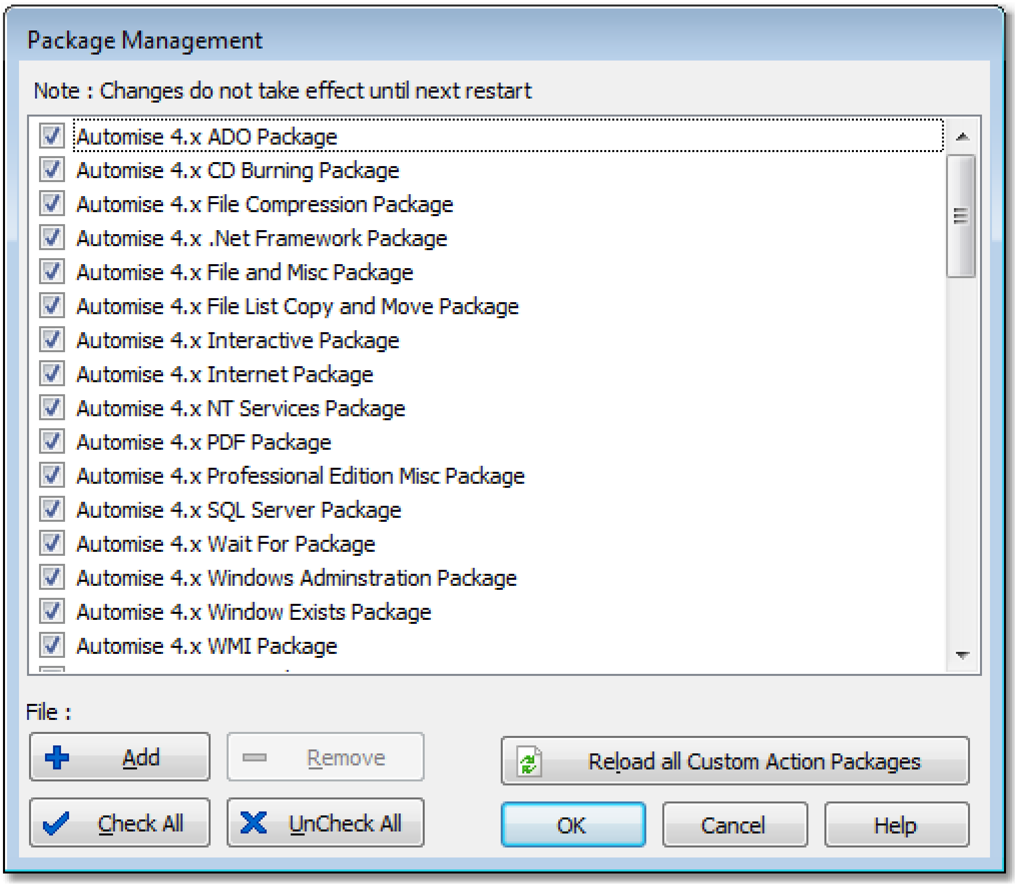
For example, if a project is loaded and it contains an action which is has not been loaded in the IDE (because the package has been disabled or removed), then it shows an error message such as the one below:



```
WARNING Action : "Edit XML File" is not loaded and is referenced by this project, if you save this project the missing actions will be deleted from the project
 (To correct the issue, go to Tools -> Package Manager and ensure the relevant action package (.bpl) file is loaded and enabled.)
WARNING Action : "Merge XML" is not loaded and is referenced by this project, if you save this project the missing actions will be deleted from the project
 (To correct the issue, go to Tools -> Package Manager and ensure the relevant action package (.bpl) file is loaded and enabled.)
WARNING Action : "XML Node Iterator" is not loaded and is referenced by this project, if you save this project the missing actions will be deleted from the project
 (To correct the issue, go to Tools -> Package Manager and ensure the relevant action package (.bpl) file is loaded and enabled.)
WARNING Action : "Transform XML" is not loaded and is referenced by this project, if you save this project the missing actions will be deleted from the project
 (To correct the issue, go to Tools -> Package Manager and ensure the relevant action package (.bpl) file is loaded and enabled.)
```

3.6.5 Package Manager

The Automise Package Manager allows you to choose which action packages are loaded at a given time.



Package Management

Note : Changes do not take effect until next restart

- ☑ Automise 4.x ADO Package
- ☑ Automise 4.x CD Burning Package
- ☑ Automise 4.x File Compression Package
- ☑ Automise 4.x .Net Framework Package
- ☑ Automise 4.x File and Misc Package
- ☑ Automise 4.x File List Copy and Move Package
- ☑ Automise 4.x Interactive Package
- ☑ Automise 4.x Internet Package
- ☑ Automise 4.x NT Services Package
- ☑ Automise 4.x PDF Package
- ☑ Automise 4.x Professional Edition Misc Package
- ☑ Automise 4.x SQL Server Package
- ☑ Automise 4.x Wait For Package
- ☑ Automise 4.x Windows Adminstration Package
- ☑ Automise 4.x Window Exists Package
- ☑ Automise 4.x WMI Package

File :

At startup, only selected packages are loaded. Unchecking unwanted packages helps to tidy up the Action Types panel as well as accelerating Automise startup. Packages can also be removed entirely by clicking the Remove button.

Note that changes to packages only take effect after Automise has been restarted.

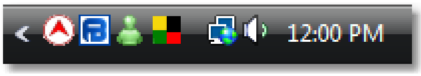
Automise action packages come in three kinds:

- - Internal Automise BPL packages.
-
- - Internal Automise custom action packages (such as the IIS 5 custom action package shown above.)
-
- - Custom Automise action packages created with ActionStudio or downloaded from the community downloads web page.

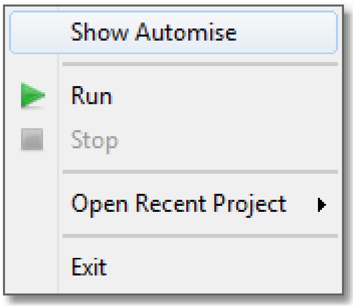
(Note that the Automise package manager will automatically add new custom action packages from the ActionDefs folder, although custom action packages can be added from any location.)

3.6.6 Tray Icon

Automise can optionally show in the System Tray:



The Automise Tray Icon provides the following menu on Right-Click:



Show Automise [<project>] - This will restore Automise if it is minimised. The current open project name is shown in brackets.

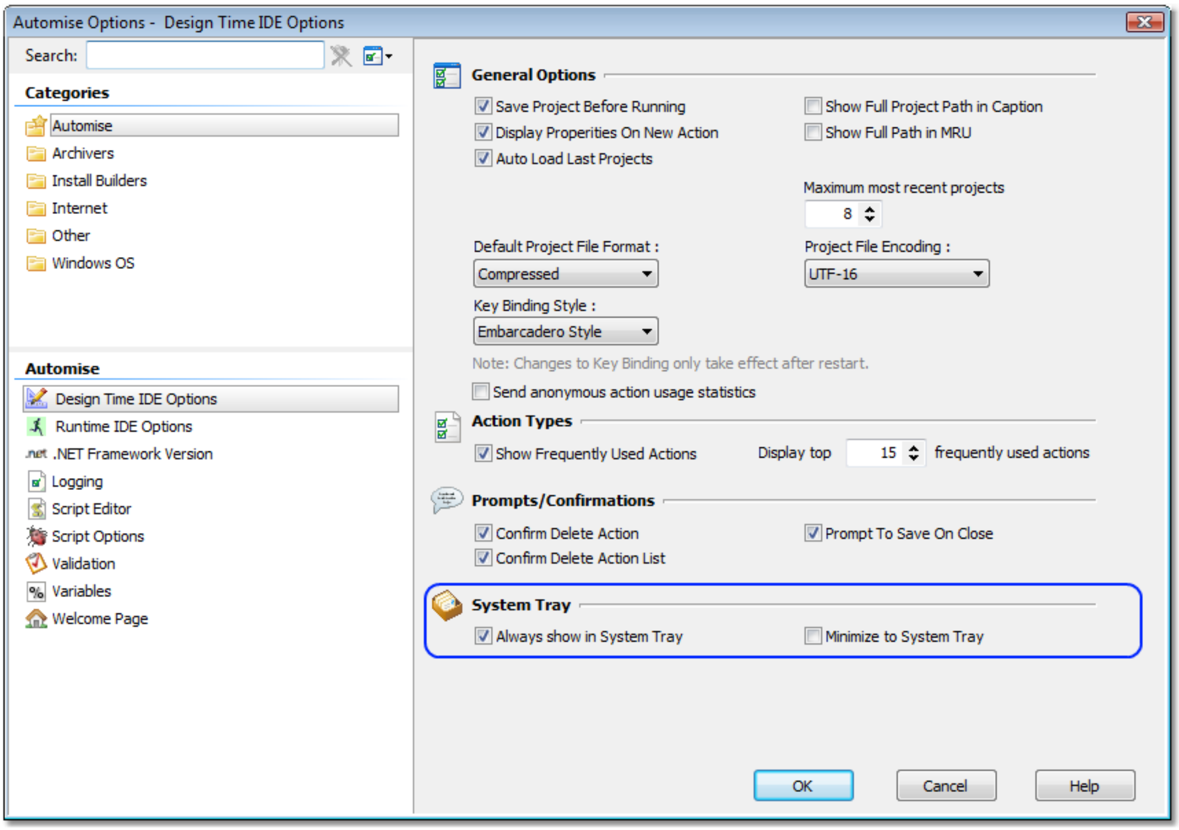
Run <project> - This will start the project.

Stop - Stops the project

Open Recent Project - This allows you to open a project from the MRU (Most Recently Used) list of Automise projects.

Exit - Closes Automise

To change the System Tray settings:



General Options

☑ Save Project Before Running
☑ Display Properities On New Action
☑ Auto Load Last Projects
☐ Show Full Project Path in Caption
☐ Show Full Path in MRU

Maximum most recent projects: 8

Default Project File Format : Compressed

Project File Encoding : UTF-16

Key Binding Style : Embarcadero Style

Note: Changes to Key Binding only take effect after restart.

☐ Send anonymous action usage statistics

Action Types

☑ Show Frequently Used Actions Display top 15 frequently used actions

Prompts/Confirmations

☑ Confirm Delete Action
☑ Confirm Delete Action List
☑ Prompt To Save On Close

System Tray

☑ Always show in System Tray
☐ Minimize to System Tray

System Tray:

Always show in System Tray - The tray icon will show at all times, regardless of the windows state (Minimised, Maximised, etc)

Minimize to System Tray - When minimised, Automise will not show on the task bar or the task manager, but will show on the system tray. To restore it, double click the tray icon.

The Tray Icon will also display the state of the project:

Ⓐ - Automise is currently idle

🏃 - project is currently running

✓ - project completed successfully

✖ - project completed with an error

⚠ - project failed validation

⏸ - project is paused

3.6.7 Keyboard Shortcuts

Use the following keyboard shortcuts to quickly navigate your way around Automise.

	Embarcadero style	**Microsoft style**
Run	F9	F5
Step	F8	F10
Stop run	Escape	Shift+F5
Run from action	Ctrl+F9	Alt+F5
Run selected actions	Alt+F9	Ctrl+Alt+F5
Run selected actions and children	Ctrl+Alt+F9	Shift+Alt+F5
Run current action list	Shift+Ctrl+F9	Shift+Ctrl+F5
Continue	Shift+F9	Ctrl+F5
Enable/disable action	F4	F11
Toggle breakpoint	F5	F9
Clear all breakpoints	Shift+Ctrl+F5	Ctrl+F9
Action properties	F11 / Enter	F12 / Enter
Select Properties Tab	F12	F4

To switch between Embarcadero and Microsoft style key mappings, use the Design Time Options page.

Other design and debugging shortcuts include:

Action	**Shortcut**
Indent action	Ctrl+Right
Outdent action	Ctrl+Left

Move action up	Ctrl+Up
Move action down	Ctrl+Down
Delete action	Del
Collapse node	Left / Numpad minus (-)
Expand node	Right / Numpad plus (+)
Collapse all	Numpad slash (/)
Expand all	Numpad asterisk (*)
Edit action title	F2 / Any unassigned key
Filter log by action	F7
Quick help on action	Alt+F1

See Working with Actions for more information.

Inside actions:

Action help	F1
Edit Field	F2
Add variable	F3

Navigational shortcuts:

Select History & Statistics tab	Ctrl+H
Select Run Log	Alt+F1
Select Actions tab	Ctrl+I
Edit variables	Shift+F2
Find and replace	Ctrl+F
Next search result	F3
Previous search result	Shift+F3

Project shortcuts

New project	Ctrl+N
Open project	Ctrl+O
Save project	Ctrl+S

3.7 Automise Projects

3.7.1 Project Files & Other Files

Automise can use one of two different file formats for the project file, compressed and uncompressed.

<project>.atz4

Compressed Automise project file. This is the default format for a project file. You can use any standard zip tool to decompress it (you'll probably need to change extension for most zip tools to recognise the format though). It is recommended that you use the compressed project file format for large projects.

<project>.atp4

Uncompressed Automise project file. Project files are standard XML format and so are quite verbose. If you are adding project files to your version control system, it is recommended that you use this format so that incremental changes can be recorded.

Auxiliary Project Files

Automise sometimes creates one or more of the following files alongside the project file. Files are always created in the same directory.

<project>.atpinf

The project information file contains other information and properties relevant to the project, such as:

- Persistent variable variables
- Watches on variables
- Meta-data, such as last build date
- Node State (if the action is expanded or collapsed in the Action List tree)
- Is Breakpoint (if a breakpoint is set on this action)
- Author and project notes set on the Project Information screen.

This file can be version controlled, but must not be read only when a build is running. The format of the file is XML.

Note: This information was stored in files such as .fbv, .fbw and .fbd in previous versions. These extensions are no longer used.

<project>.log4

The log4 file is the log archive file. It records the logs of any previous projects, up to the "Log History Count" option (default is 4). The log file must be writeable in order to open the project file in Automise.

It is recommended that this file is not added to version control, because it is a binary database file. If you do put this file under version control, make sure it is not read only when the project is opened or run.

<project>.lck

The lock file is used to indicate when the project is currently opened in Automise. Other instances of Automise cannot open the same project at the same time. The lock file is deleted when a project is closed, or Automise closes. If Automise terminates unexpectedly the lock file for the opened project may not be cleaned up automatically; in this instance it's safe to delete the lock file manually.

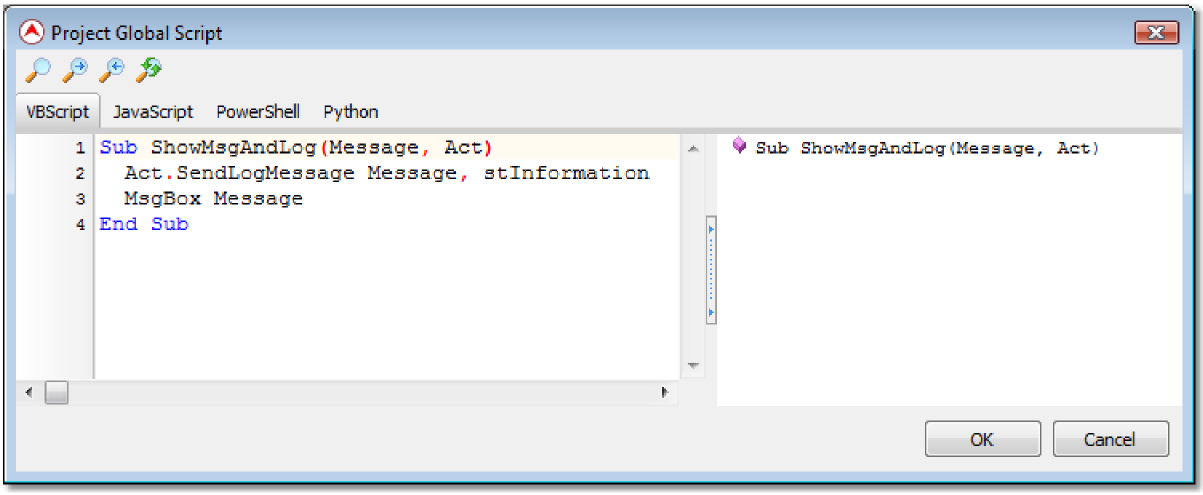
See also

Upgrading Projects

3.7.2 Project Global Script

Automise allows you to have global VBScript, JavaScript and PowerShell code stored in your project file. This script code is available to all actions in the project, in the action event script code.

The Project Global Script is accessible from the Project menu. Here is an example of some global VBScript:



```
Sub ShowMsgAndLog(Message, Act)
  Act.SendLogMessage Message, stInformation
  MsgBox Message
End Sub
```

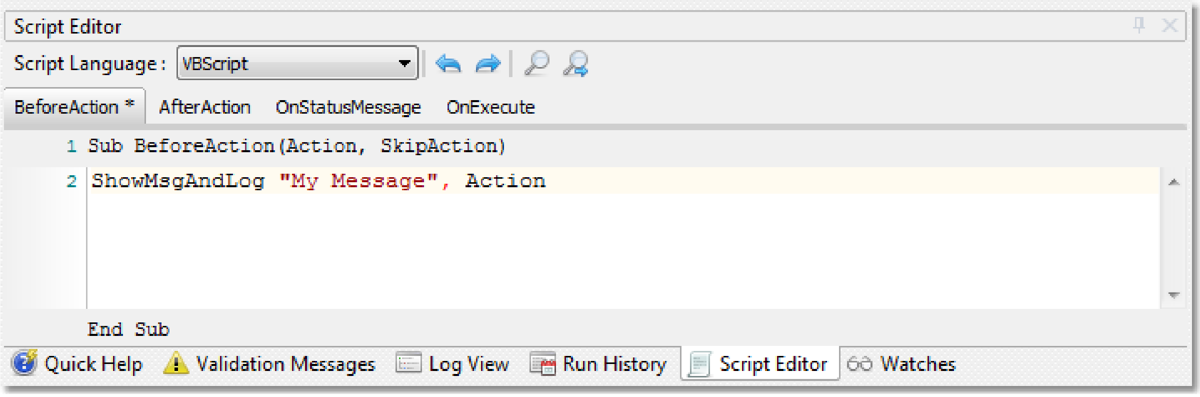
This dialog supports syntax highlighting, code completion, parameter hints and also will display subs, functions and constants in the right hand side section to make it easy to see an overview of your code at a glance as well as easy navigation within the script (double-click an item and it scrolls the code to the right spot).

From left to right, the four buttons on the search toolbar are:

- Find (Ctrl+F)
- Find Next (F3)
- Find Previous (Ctrl+F3)
- Replace (Ctrl+H)

Using the Project Global Script

Once you have defined a variable, sub or function in the Project Global Script, you can use it in script events. It is available in all scripting like any other variable, sub or function.



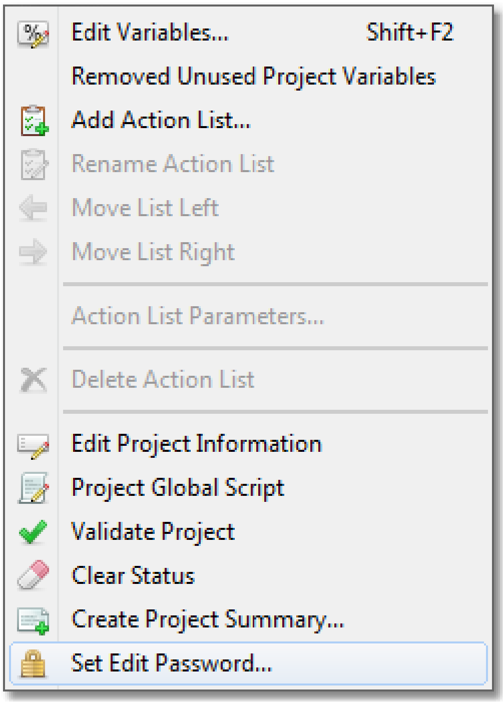
Script Language : VBScript

```
Sub BeforeAction(Action, SkipAction)
ShowMsgAndLog "My Message", Action
End Sub
```

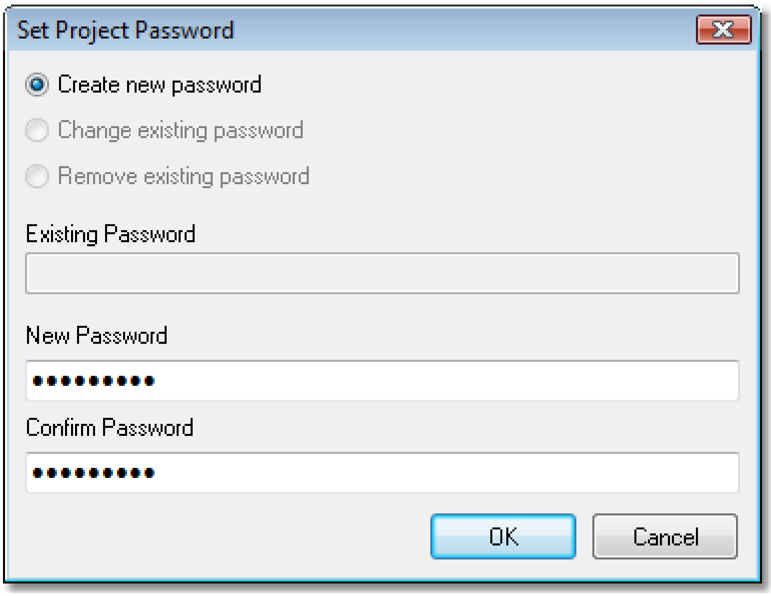
3.7.3 Project Edit Password

The Edit Password for a project allows the owner of the project to set a password which is required to view and edit the steps of the project in design mode. When you open a project is opened that has an edit password, it will open on the Build Summary screen - clicking the Design tab will prompt for the password. The project can be run without the password.

To set an edit password, choose "Set Edit Password" from the Project menu:



This will bring up the Project Edit Password dialog. Type in the password required to edit the project. To set no password for the project, simply leave the new password fields blank.


Set Project Password

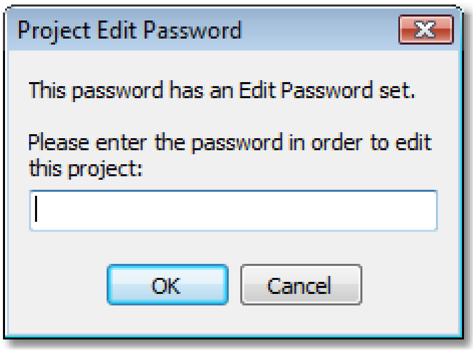
● Create new password
○ Change existing password
○ Remove existing password

Existing Password

New Password: •••••••••

Confirm Password: •••••••••

Now that the project password is set, if you click on the Design tab you get the following prompt:



Project Edit Password

This password has an Edit Password set.

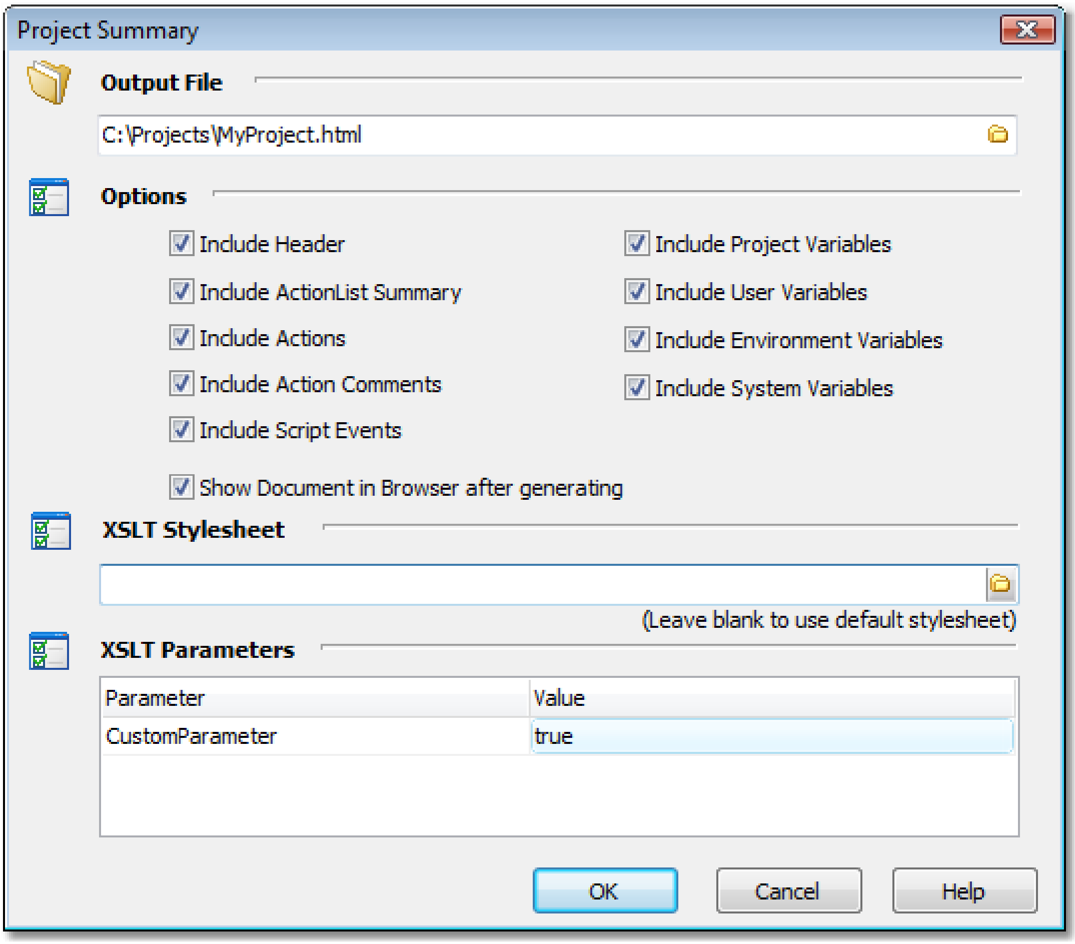
Please enter the password in order to edit this project:

One of the limitations with project edit passwords is that the project file must be saved as a compressed project file (.atz4) - as this prevents the encrypted password in the project file being removed easily.

3.7.4 Project Summary

The project summary produces an HTML Document with an overview of the project.

You can choose which information is output to the summary document.



Project Summary

Output File

C:\Projects\MyProject.html

Options

- ☑ Include Header
- ☑ Include ActionList Summary
- ☑ Include Actions
- ☑ Include Action Comments
- ☑ Include Script Events
- ☑ Include Project Variables
- ☑ Include User Variables
- ☑ Include Environment Variables
- ☑ Include System Variables

☑ Show Document in Browser after generating

XSLT Stylesheet

(Leave blank to use default stylesheet)

XSLT Parameters

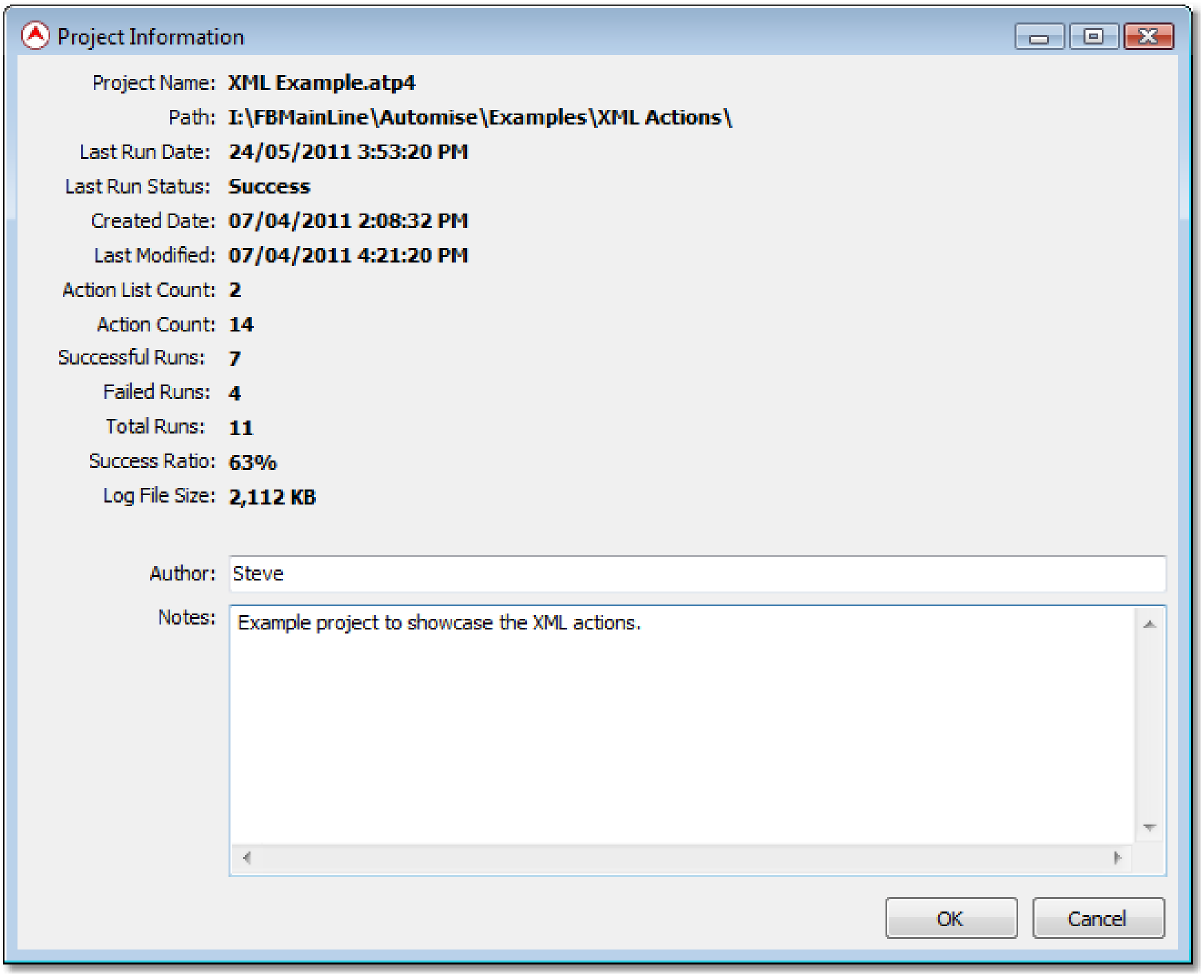
Parameter	Value
CustomParameter	true

The XSLT Stylesheet option allows you to use your own custom stylesheet to control the format of the output. If this field is left blank the default stylesheet (<install directory>\Stylesheets\CreateProjectSummaryFromProject.xsl) will be used.

The XSLT Parameters allow you to pass custom parameters to your stylesheet.

3.7.5 Project Information

The Project Information window displays properties of your project, such as the location and last build date, and lets you set author details and free text notes.



Project Information

Project Name: XML Example.atp4
Path: I:\FBMainLine\Automise\Examples\XML Actions\
Last Run Date: 24/05/2011 3:53:20 PM
Last Run Status: Success
Created Date: 07/04/2011 2:08:32 PM
Last Modified: 07/04/2011 4:21:20 PM
Action List Count: 2
Action Count: 14
Successful Runs: 7
Failed Runs: 4
Total Runs: 11
Success Ratio: 63%
Log File Size: 2,112 KB

Author: Steve
Notes: Example project to showcase the XML actions.

The **Author** and **Notes** fields can contain any information you like. They are shown by default on the Welcome Tab.

To view the Project Information window:

- Double-click the **Project Information** node in the Project Tab.
- Select **Project Information** in the **Project** menu.

Note: The project information for a project called "project1" is written to a file called "project1.fbpinf" in the same directory. This file must always be writable.

4 Scripting

4.1 Scripting in Automise

Automise supports VBScript and JavaScript languages (via Active Scripting) and also Python and PowerShell scripting (via the PowerShell runtime).

PowerShell

To use PowerShell in Automise you need to have PowerShell 1.x installed on your machine. The download at the time of writing is: http://www.microsoft.com/windowsserver2003/technologies/management/powershell/download.mspx

Python

Python support is included with Automise, using Microsoft's IronPython 2.0.

VBScript and JavaScript

To use VBScript and JavaScript you need to have Active Scripting installed on your machine (this should be installed by default). If you don't have Active Scripting support installed, or you want to upgrade Active Scripting on your machine, the download for Windows Script 5.6 is: http://www.microsoft.com/downloads/details.aspx?FamilyId=C717D943-7E4B-4622-86EB-95A22B832CAA&displaylang=en

Debugging

Active scripting based languages can be debugged using the Active Script Debugger, provided the language vendor supports debugging. To enable debugging you will need the Active Script Debugger installed, or Visual Studio.NET (which overrides the script debugger). PowerShell and Python script debugging is not yet possible.

What can I do in the Scripts

VBScript and JavaScript scripts can make use of almost all the functionality of Active Scripting, including external COM/ActiveX calls. PowerShell scripts can make use of all the PowerShell functionality. Python scripts can use IronPython and .NET Framework built-in functionality, as well as Python standard libraries if these are installed.

Automise Script Events are Triggered Before and After an Action Executes. See Action Script Events for more information on these events.

There is also a Run Script action.

Using Automise Variables in Scripts

Automise variables are available in script events as global variables. You can reference them just as you would any other identifier. In PowerShell, you must access Automise variables with the following syntax:
$FBVariables.GetVariable(<variablename>) and $FBVariables.SetVariable(<variablename>)

Using Action List Parameters in Scripts

Action List Parameters are available in script events as global variables (same as Automise variables.)

Including External Scripts

You can include external script files in your VBScript and JavaScript action scripts, by inserting a comment with "USEUNIT scriptfilename" You cannot include external script files with

PowerShell.

Example - VBScript

'USEUNIT c:\Automise\scripts\iis_stuff.vbs

Example - JavaScript

//USEUNIT c:\Automise\scripts\iis_stuff.js

Note that the external script file must be written in the same script language as the event handler script. Automise provides sample scripts to do things such as restart IIS, shut down COM+ components, etc.

You can also use Automise variables in the path, for example : 'USEUNIT "%SCRIPTPATH%\Test.vbs"
Note that this is the only time you would use the %variable% syntax for Automise variables in the script editor, as the useunit line is preprocessed before the script is run.

Custom Action Studio Script Actions

Entire action types can be written in Action Studio for inclusion in Automise.

See Also

Related documentation and VBScript/JavaScript languages descriptions can also be found on the Microsoft Active Scripting site:

http://msdn.microsoft.com/scripting/

PowerShell documentation can be found at the Windows PowerShell site:

http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx

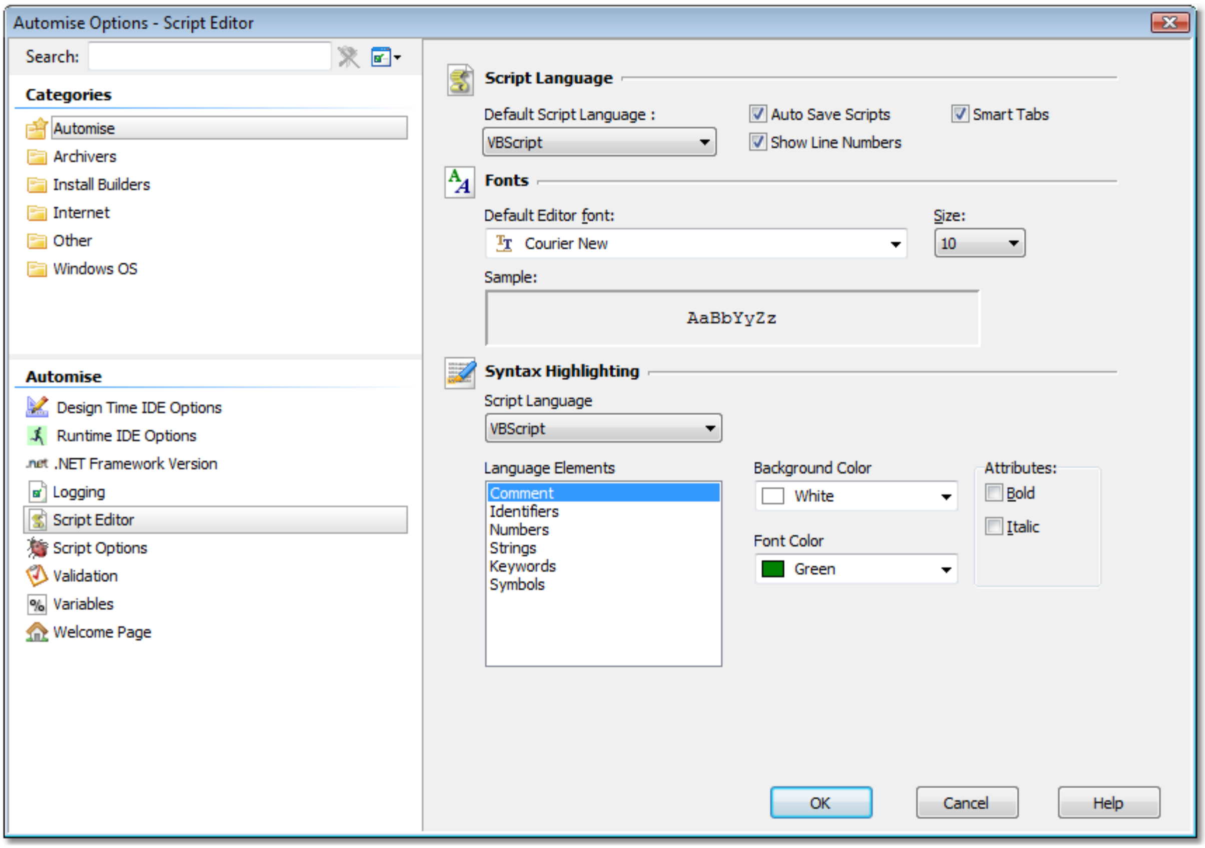
Information about IronPython can be found at the IronPython home page on Codeplex:

http://www.codeplex.com/IronPython

Global Script Functions | Action Script Events | Accessing TStrings based Properties

4.2 Script Editor Options

To edit the Automise Scripting Options, go to Tools menu -> Options -> Scripting.



Default Script Language : VBScript

☑ Auto Save Scripts

☑ Smart Tabs

☑ Show Line Numbers

Default Editor font: Courier New

Size: 10

Sample: AaBbYyZz

Script Language: VBScript

Background Color: White

Font Color: 🟩 Green

Attributes:

□ Bold

□ Italic

The Script Editor Options allow you to customize the script editor in Automise.

Script Language

Default Script Language - choose either VBScript, JavaScript or PowerShell as the default scripting language.

Auto Save Scripts - action script events will automatically be saved if you move to a different action

Smart Tabs - enable smart tabs if you want to have tab width values depending on upper lines

Show Line Numbers - show the line numbers column in the script editor (makes it easier to find script errors as they are reported by the line number)

Fonts

Default Editor Font - choose the font to display your script code in

Size - size of the font in the script editor

Syntax Highlighting

Script Language - choose the language so you can change the colours and attributes of the various language elements

Language Element - the various language elements of the chosen language which can be changed

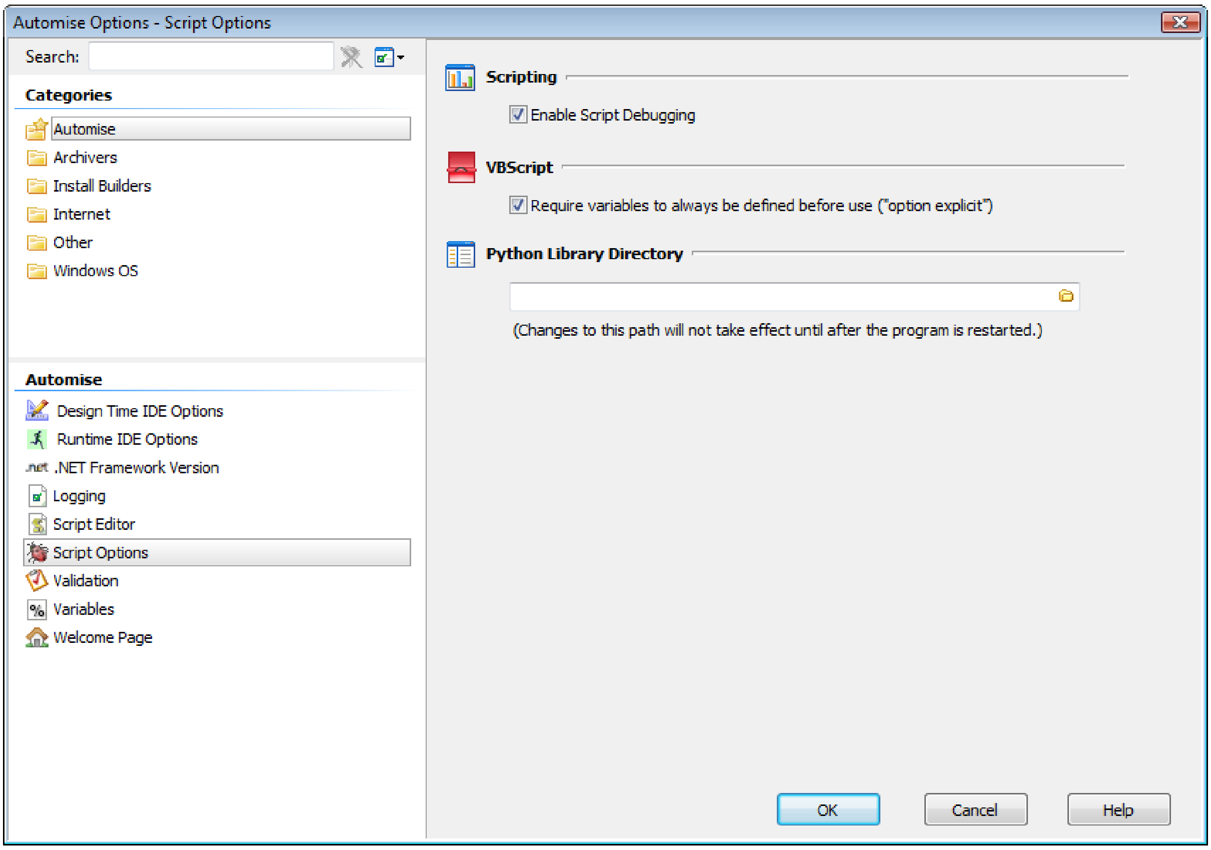
Background Color - choose the background colour of the selected language element

Font Color - choose the font colour of the selected language element

Attributes - select Bold and/or Italic for the selected language element

4.3 Script Debugging Options

To edit the Automise Scripting Debug Options, go to Tools menu -> Options -> Scripting and then choose the Script Debugging tab.



☑ Enable Script Debugging

☑ Require variables to always be defined before use ("option explicit")

(Changes to this path will not take effect until after the program is restarted.)

The Script Debugger Options allow you to customize the script debugger in Automise.

Enable Script Debugging - Active scripting based languages can be debugged using the Active Script Debugger, provided the language vendor supports debugging. To enable debugging you will need the Active Script Debugger installed, or Visual Studio.NET (which overrides the script debugger). If this option is turned off, then a dialog will appear containing the error if an error occurs when executing script code.

VBScript, Option Explicit - turning on "Option Explicit" forces all variables that are used in your VBScript code to be declared before they are used (eg. dim MyValue)

4.4 Global Script functions

Global Scripting Functions

Apart from standard VBScript/JavaScript/PowerShell functions, Automise exposes the following global functions :

procedure SaveProject;
Saves the current Automise project.

function ExtractFilePath(value : string) : string;
Extracts the path (minus the filename) from a fully qualified filename.

function ExtractFileName(value : string) : string;
Extracts the FileName (minus the path) from a fully qualified filename.

function ExtractFileDrive(value : string) : string;
Extracts the filename drive letter

function ExtractFileExt(value : string) : string;
Extracts the file extension including the period

function ExpandUNCFileName(FileName : String) : String;
Expands any pathname to either the fully qualified UNC pathname (ie \\server\share\folder\file.txt) or (if the file is a local file)
the fully qualified local pathname (ie C:\folder\file.txt.)

procedure FBSetCaption(value : string);
Sets the titlebar caption for Automise
FBSetCaption does nothing when Automise is run from the scheduler.

function GetClipBoardText : string;
Gets the text currently on the clipboard
GetClipBoardText does nothing when Automise is run from the scheduler.

procedure CopyToClipBoard(const value : string)
Copies the string to the clipboard
CopyToClipBoard does nothing when Automise is run from the scheduler.

function FBFormatDateTime(format : string; value : DateTime) : string;
Formats the specified DateTime as a string
See Format DateTime Formatting Options

function StrToDate(value : string) : DateTime;
Converts a string into a Date

function StrToDateTime(value : string) : DateTime;
Converts a string into a DateTime

function ChangeFileExt(filename : string; newext : string) : string;
Changes the file extension of filename to the specified new extension.
eg. ChangeFileExt("c:\temp\test.txt", ".doc") = "c:\temp\test.doc"

function IncludeTrailingPathDelimiter(value : string) : string;
Appends a trailing path delimiter to the specified directory if required.

function ExcludeTrailingPathDelimiter(value : string) : string;

Removes a trailing path delimiter from the specifed directory if it exists.

function ExpandFileName(value : string) : string;
Expands the short filename and path to the fulll filename/path

function FileExists(value : string) : boolean;
Returns true if the specified file exists

function GetCurrentDir : string;
Returns the current working directory

function SetCurrentDir(value : string) : boolean;
Set the current working directory

function ExpandRelativePath(filepath : string; relativeto : string) : string;
Returns the full path and filename of the file specified with the relative path.
eg. ExpandRelativePath("..\..\Source", "myfile.txt") = "c:\Dev\Source\myfile.txt"

function NewGUIDString : string;
Creates a new GUID (Globally Unique Identifier)

function ExtractMajorVer(value : string) : string;
Extracts the Major Version value from a version string
eg. ExtractMajorVer("3.0.23.1") = "3"

function ExtractMinorVer(value : string) : string;
Extracts the Minor Version value from a version string
eg. ExtractMajorVer("3.0.23.1") = "0"

function ExtractReleaseVer(value : string) : string;
Extracts the Release Version value from a version string
eg. ExtractMajorVer("3.0.23.1") = "23"

function ExtractBuildVer(value : string) : string;
Extracts the Build Version value from a version string
eg. ExtractMajorVer("3.0.23.1") = "1"

function MessageBox(text : string; title : string; style : integer) : integer;
Displays a message box to the user.
See MessageBox Constants

procedure alert(text : string);
Displays an alert dialog to the user.

function EncryptString(value : String) : String;
function DecryptString(value : String) : String;

Two functions to scramble and descramble passwords, etc. (ie for storage in project files.)
Uses blowfish with a hardcoded key. Not to be considered secure.

function RunProcess...

Deprecated. Allows the user to launch an external process and obtain a result code when it exits.
Not recommended (use ActionStudio to create an Execute Program Action instead.)

procedure SetEstimatedProgressTotal(Value : Integer);

Use to set the estimated total number of actions (for the Build Summary estimated progress bar.)
See Estimated Progress for an example script.

4.4.1 Format DateTime Formatting Options

The valid format specifiers for the **FBFormatDateTime** script function are

Specifier	Displays
c	Displays the date using the format given by the ShortDateFormat global variable, followed by the time using the format given by the LongTimeFormat global variable. The time is not displayed if the date-time value indicates midnight precisely.
d	Displays the day as a number without a leading zero (1-31).
dd	Displays the day as a number with a leading zero (01-31).
ddd	Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd	Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
ddddd	Displays the date using the format given by the ShortDateFormat global variable.
dddddd	Displays the date using the format given by the LongDateFormat global variable.
e	Displays the year in the current period/era as a number without a leading zero (Japanese, Korean and Taiwanese locales only).
ee	Displays the year in the current period/era as a number with a leading zero (Japanese, Korean and Taiwanese locales only).
g	Displays the period/era as an abbreviation (Japanese and Taiwanese locales only).
gg	Displays the period/era as a full name. (Japanese and Taiwanese locales only).
m	Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	Displays the month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNames global variable.
mmmm	Displays the month as a full name (January-December) using the strings given by the LongMonthNames global variable.
yy	Displays the year as a two-digit number (00-99).
yyyy	Displays the year as a four-digit number (0000-9999).
h	Displays the hour without a leading zero (0-23).
hh	Displays the hour with a leading zero (00-23).
n	Displays the minute without a leading zero (0-59).
nn	Displays the minute with a leading zero (00-59).
s	Displays the second without a leading zero (0-59).
ss	Displays the second with a leading zero (00-59).
z	Displays the millisecond without a leading zero (0-999).
zzz	Displays the millisecond with a leading zero (000-999).
t	Displays the time using the format given by the ShortTimeFormat global variable.
tt	Displays the time using the format given by the LongTimeFormat global

	variable.
am/pm	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm	Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.
/	Displays the date separator character given by the DateSeparator global variable.
:	Displays the time separator character given by the TimeSeparator global variable.
'xx'/"xx"	Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Example usages of the FBFormatDateTime function (in VBScript)

```
dim s
s = FBFormatDateTime("ddmmyyyy",Now)  ' returns current date in this format 01012003
s = FBFormatDateTime("c",Now)            ' returns the current date in the system short date format, eg. 01/12/2003
```

4.4.2 MessageBox Constants

function MessageBox(text : string; title : string; style : integer) : integer;
Displays a message box to the user.

The valid values for the style are:

- mbOK
- mbOKCANCEL
- mbABORTRETRYIGNORE
- mbYESNOCANCEL
- mbRETRYCANCEL
- mbICONHAND
- mbICONQUESTION
- mbICONEXCLAMATION
- mbICONASTERISK
- mbDEFBUTTON1
- mbDEFBUTTON2
- mbDEFBUTTON3
- mbDEFBUTTON4

The MessageBox function return value corresponds to the button which the user pressed:

- mrOK
- mrCANCEL
- mrABORT
- mrRETRY
- mrIGNORE
- mrYES
- mrNO
- mrCLOSE
- mrHELP

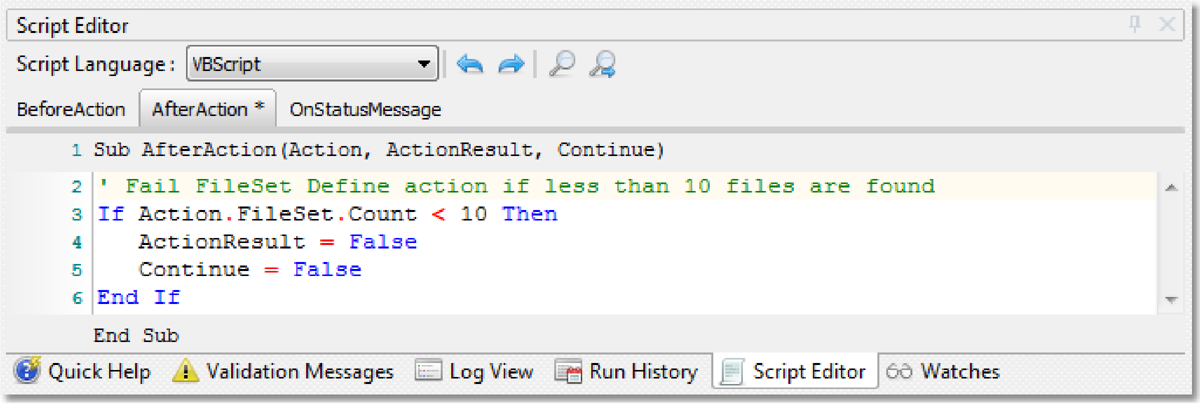
eg.

```
if MessageBox("Do you want to cancel the build", "Cancel Build", mbOKCANCEL) = mrOK then
  // build needs to stop
end if
```

4.5 Action Script Events

Each Action in Automise has **BeforeAction**, **AfterAction,** and **OnStatusMessage** script events. Some Actions (such as Iterators) define more script events.

The action Script Events are shown at the bottom of the Automise IDE in the Script Editor tab, see screenshot.



Script Language : VBScript

```
Sub AfterAction(Action, ActionResult, Continue)
' Fail FileSet Define action if less than 10 files are found
If Action.FileSet.Count < 10 Then
    ActionResult = False
    Continue = False
End If
End Sub
```

The following script events are common to all actions:

BeforeAction Event
Description: called before the action is executed.
Parameters:

- **Action** parameter allows access to the action properties and methods (see Action Properties and Methods).
- **SkipAction** parameter allows the script to return true for the action to be skipped during a project.

AfterAction Event
Description: called after the action has executed.
Parameters:

- **Action** parameter allows access to the action properties and methods (see Action Properties and Methods).
- **ActionResult** parameter - indicates if the action succeeded or failed. This may also be set to override the action status.
- **Continue** parameter - return false to stop the build, return true to continue the build ignoring the ActionResult

OnStatusMessage Event
Description: called whenever the action generates a log message
Parameters:

- **Action** parameter allows access to the action properties and methods (see Action Properties and Methods).
- **StatusMessage** - the status message object contains the information of the status message (Lines, MessageText, MessageTitle, and Progress)

Execute Condition

In addition, each action has an Execute Condition property which is a Boolean expression which should return true for the action to execute. See Action Reference for more information. The Condition property is displayed in the Property dialog for the action, in the Properties tab (Execute Condition section).

4.5.1 Action Properties and Methods

Action Properties and Methods

The Action object is passed into Script Events. The methods and properties available on the Action object are as follows:

Procedure/Methods

- procedure **SetLogTitle**(value As String)
- procedure **SendLogMessage**(MessageText As String)
- procedure **Echo**(MessageText As String)
- procedure **SendProgress**(Value As String, progress As Integer)

Functions

- function **ExpandExpression**(expr As String) As String
- function **ChildActions**(index As Integer) As OleVariant
- function **Parent** As OleVariant

Properties

- **ActionComment** As String
- **ActionLogTitle** As String
- **ActionName** As String
- **ChildActionCount** As Integer
- **Description** As String
- **IgnoreFailure** As Boolean
- **LogToVariable** As String
- **PauseInterval** As Cardinal
- **SuppressStatusMessages** As Boolean

Detailed information on the procedures, functions and properties:

procedure **SetLogTitle**(value As String)
Allows you to set the title of the Automise log.

procedure **SendLogMessage**(MessageText As String)
Sends a message to the output window. Note that this will also trigger the OnStatusMessage event (except when called from within the OnStatusMessge event!

procedure **Echo**(MessageText As String)
Same as SendLogMessage (see above).

procedure **SendProgress**(Value As String, progress As Integer)
Sends a progress message to the Run Status window. This enables the action to report it's progress as it's executing. Progress is a percentage and should therefore range between 0 and 100.

function **ExpandExpression**(expr As String) As String
This expands the string passed in by substituting Automise variables when %<variable>% is encountered.

function **ChildActions**(index As Integer) As OleVariant
Allows access to child actions, index is zero based. Use ChildActionCount property to get a

count of the child actions.

Example: Set 'Delete Read Only' option in all child Delete Files(s) actions... (in this example RemoveReadOnly is a Automise variable) add this code to the BeforeAction event handler of the parent action (an action group for example).

```
Dim child
Dim count
Dim i

count = Action.ChildActionCount - 1
For i = 0 To count
  Set child = Action.ChildActions(i)
  If Not (child Is Nothing) Then
     If child.ActionName = "Delete File(s)" Then
        child.DeleteReadOnly = RemoveReadOnly
     End If
  End If
Next
```

function **Parent** As OleVariant
Returns the Parent action of the current action. This will return null for root level actions.

property: **ActionComment** As String
Provides access to the action comment.

property: **ActionLogTitle** As String
Allows you to change the title of the action's entry in the output window. Note that this has no effect if set from the AfterAction event (since the log entry has already been made.)
eg.

```
 Action.LogTitle  = "Full Build - VB6"
 Action.SendLogMessage("File deleted")
```

property: **ActionName** As String
Provides access to the Action's name

property: **ChildActionCount** As Integer
Returns the number of Child Actions the action has.

property: **Description** As String
Provides access to the Action's description

property: **IgnoreFailure** As Boolean
Provides access to the Action's IgnoreFailure flag

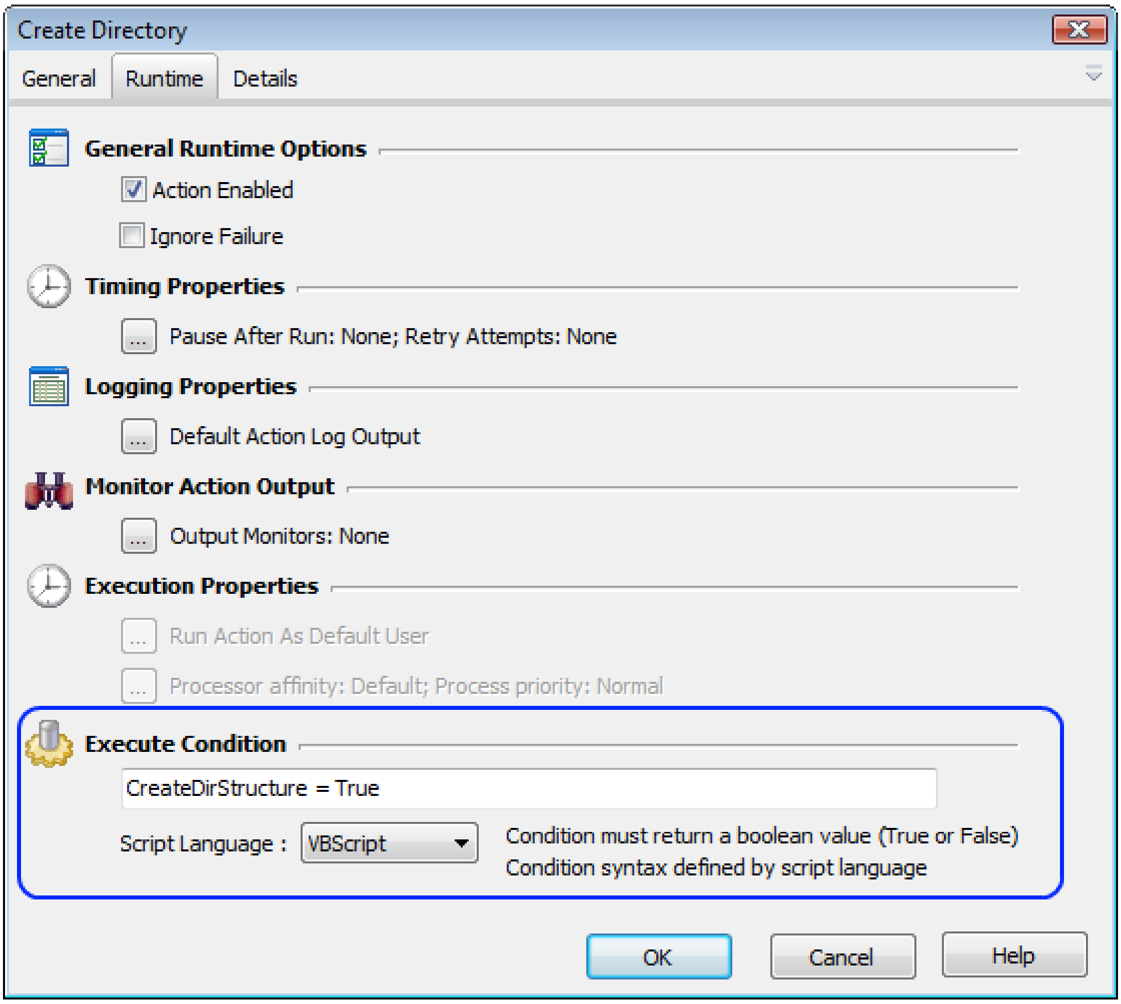
property: **LogToVariable** As String
Provides access to the Action's LogToVariable flag

property: **PauseInterval** As Cardinal
Provides access to the Action's PauseInterval property

property: **SuppressStatusMessages** As Boolean
Provides access to the Action's SuppressStatusMessages property

4.5.2 Execute Condition

Every action in Automise has an Execute Condition property:



☑ Action Enabled

☐ Ignore Failure

Pause After Run: None; Retry Attempts: None

Default Action Log Output

Output Monitors: None

Run Action As Default User

Processor affinity: Default; Process priority: Normal

CreateDirStructure = True

Script Language : VBScript

Condition must return a boolean value (True or False)
Condition syntax defined by script language

The Execute Condition field is evaluated before running the action. If the condition evaluates to True (or is empty) then the action will execute, otherwise the action will be skipped.

Automise Variables

Execute conditions are specified in a scripting language, so %'s are not used when referring to Automise variables. For example, if you have a Automise variable *CreateDirStructure* then you can simple specify *CreateDirStructure* and not %*CreateDirStructure%*. In the above example the previous actions should have set the *CreateDirStructure* variable to either true or false.

Here is another example of how to use the Execute Condition property:

(VBScript)
ClearLogs = "All"

(Javascript)

ClearLogs == "All"

(Powershell)
$FBVariables.GetVariable("ClearLogs") -eq "All"

where ClearLogs is an Automise variable.

4.6 Accessing the Options settings via scripting

Automise allows you to modify Automise options at runtime using scripting. This makes it possible to specify the path to a third party tool at runtime so that a Automise project can for example use a different version of a third party tool. Note that these modified settings are not saved and will be lost when you close Automise, unless you choose Tools -> Options and click OK before closing.

The options objects are accessible using the GetOptionsObject script function. This function takes the name of the options object as a parameter. The name is the same as appears in the Options dialog. For example to set the default Hyper V Host you could use the following code:

```
Dim hyperVOptions
Set hyperVOptions = GetOptionsObject("Hyper V Options")
hyperVOptions.Properties.PropertyAsString("DefaultHostname") =
"MyHyperVServer"
```

Listed below are the available properties on the IDE options objects.

Automise General Options

Name : General Options

```
   property PromptOnClose : boolean
   property PromptOnNew : boolean
   property DisplayPropDialog : boolean
   property SaveBeforeRun : Boolean
   property ShowTree : Boolean
   property AutoLoadLastProject : Boolean
   property PromptOnDeleteAction: boolean
   property AutoSizeListColumns : boolean
   property DisplayScriptErrors : Boolean
```

Automise Script Editor

Name : Script Editor

```
   property LineNumbers: boolean
   property DefaultScriptLanguage : String
   property AutoSaveScripts : boolean
```

Automise Logging Options

Name : Logging

```
   property LogHistoryCount : integer
```

```
property IncludeActionOutput : boolean
property OnlyIncludeErrorAction : boolean
property ConfirmDeleteRunLog : boolean
property ConfirmDeleteLogFile : boolean
```

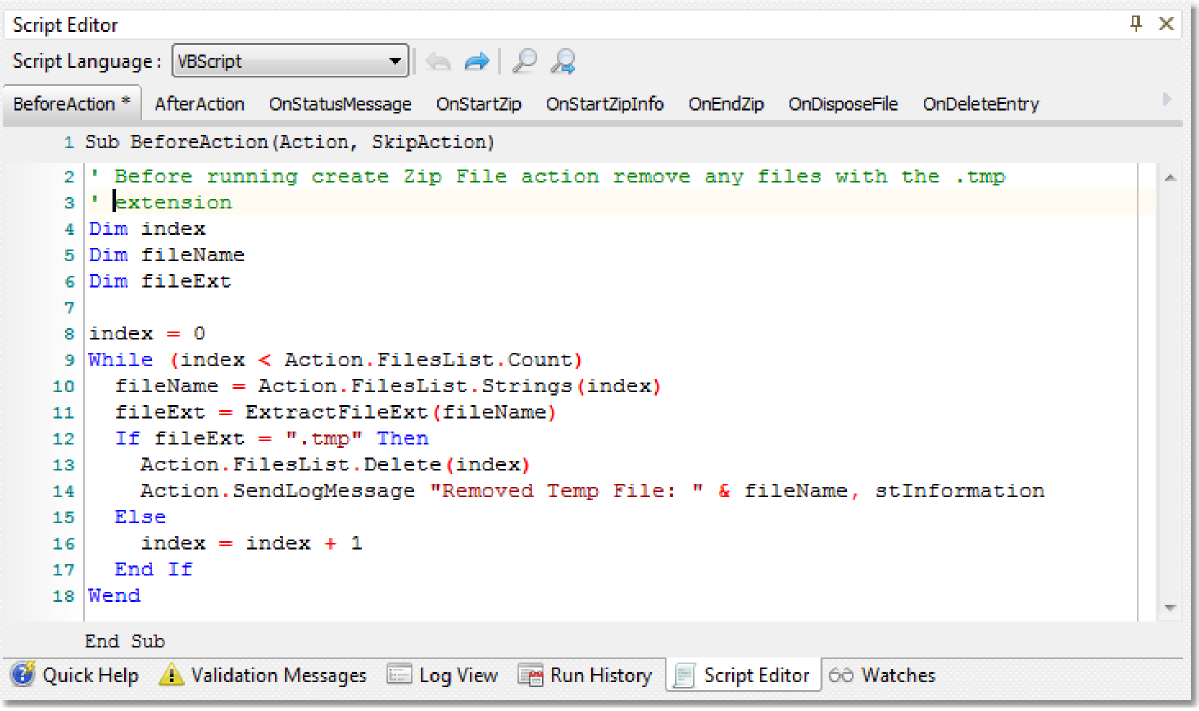
Properties of Option objects for other option objects are available through the script editor automatic code completion (like Intellisense).

4.7 Accessing TStrings based properties

You will note that some Action properties are of type TStrings. TStrings is a standard Delphi type that we have exposed to the scripting engine. It is basically a string collection. Below are the properties of TStrings that you will find useful:

procedure Clear;
function Add(const S: string): Integer;
procedure Delete(Index: Integer);
procedure Insert(Index: Integer; const S: string);
function IndexOf(const S: string): Integer;
property Strings[Index: Integer]: string; //index is zero based
property Count: Integer;
property Text: string; //returns the entries in the collection, each entry on a new line
property Names[Index: Integer]: string; // used to access the name of a string which is of the format <name>=<value>
property Values[Index: String]: string; // used to access the value of a string which is of the format <name>=<value>
property ValueFromIndex[Index: Integer] : string; // used to access the value of a string which is of the format <name>=<value>

Example:
The following code removes any file from the File list of the Create Zip file action that has a file extension of **.tmp* before the Zip file is created:



```
Sub BeforeAction(Action, SkipAction)
' Before running create Zip File action remove any files with the .tmp
' extension
Dim index
Dim fileName
Dim fileExt

index = 0
While (index < Action.FilesList.Count)
  fileName = Action.FilesList.Strings(index)
  fileExt = ExtractFileExt(fileName)
  If fileExt = ".tmp" Then
    Action.FilesList.Delete(index)
    Action.SendLogMessage "Removed Temp File: " & fileName, stInformation
  Else
    index = index + 1
  End If
Wend

End Sub
```

5 Variables

5.1 Variables Overview

Variables in Automise are the key to making your projects dynamic. Variables can be used in almost every text property of every Automise action, for example in fields that specify file paths, directories etc.

There are 5 types of variables available:

- Local variables are defined on Action Groups, and only available to child actions.

- Project variables are defined and managed by you, and are specific to this project.

- User variables are defined and managed by you, and are global across all projects, but are specific to one Windows user.
- Application variables are defined by Automise, to give you useful information about the context of the project.
- Environment variables simply provide access to Windows environment variables such as PATH and OS. The range of environment variables available depends on your installation of Windows and other software.

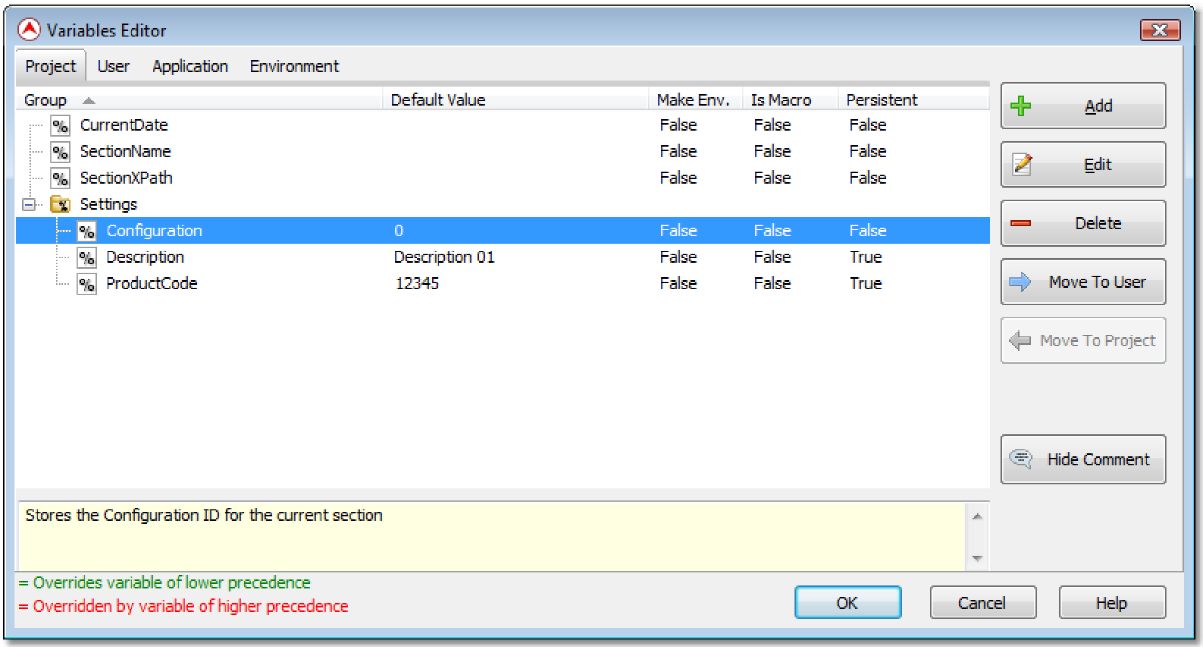
System and Environment variables cannot be edited in any way from Automise.

Variables can be modified at run time by using actions (such as the Set Variable Action), or from Active Scripting events. Some actions allow you to set a variable to reflect the action's output. Action output can also be logged to a variable.

Variables defined in Automise can be referenced in Automise VBScript and JavaScript in the same way as normal script variables. In PowerShell, variables can be referenced by using the following syntax: $FBVariables.GetVariable(<variablename>)/$FBVariables. SetVariable(<variablename>). Only Project and User variables can be modified.

Variables Editor

To create or edit existing variables, use the Variables Editor. You can launch the Variables Editor from the Project menu by selecting Edit Variables, with the keyboard short cut <Shift+F2>, or by double clicking on the Variables node of the Project Tree.



Group	Default Value	Make Env.	Is Macro	Persistent
CurrentDate		False	False	False
SectionName		False	False	False
SectionXPath		False	False	False
Settings				
Configuration	0	False	False	False
Description	Description 01	False	False	True
ProductCode	12345	False	False	True

Stores the Configuration ID for the current section

= Overrides variable of lower precedence

= Overridden by variable of higher precedence

In addition to the four categories of variables, there are three flags that can be set on each variable.

Make Env.

Project and User Variables can be made available as environment variables to applications that are executed by Automise. That is, if you create a variable MYVAR and set the "Make Env." flag, a command shell action could access it as %MYVAR%.

Is Macro

The Is Macro flag forces Automise to re-evaluate the variable by expanding the Default Value, whenever that variable is referenced during the build. This is particularly useful for variables that are used in scripting: whereas most actions automatically expand all variable references, regardless of the "Is Macro" flag, this does not happen in script events. If you set this flag, however, any contained variable references are automatically expanded.

For example, you can make a "Buildpath" macro variable that contains "%BuildHome%\%ProductName%\%BuildName%". Each time it is accessed, whether from script or an action, it will expand to the current value of those variables.

Macro Variables cannot be set during the project using the Set Variable action or any other means.

Persistent

The values of persistent variables are stored automatically between executions of the Automise project. Persistent variables are saved regardless of whether or not the project is saved.

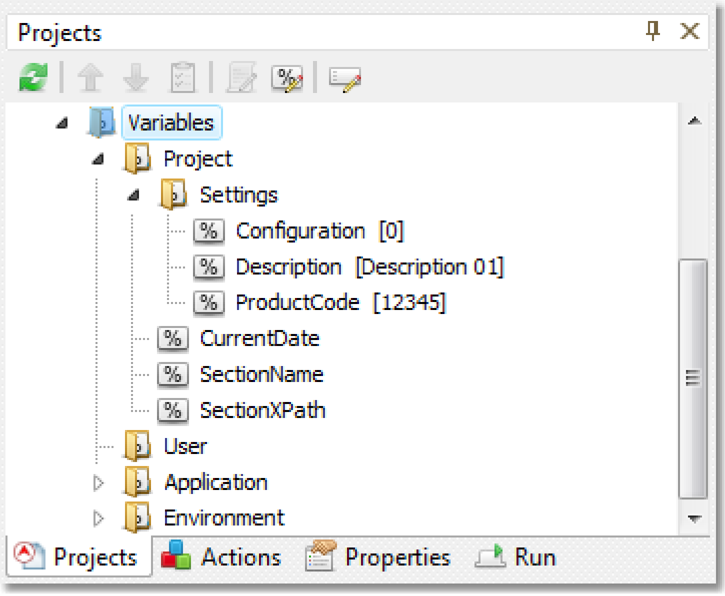
Persistent variable values are stored in .fbpinf files. Each fbpinf file has the same name as the Automise project (to persist variables, Automise needs write access to the directory containing the project and the ability to create or write to the .fbpinf file.)

Groups

Variables can be organised into variable groups to assist with managing your project. The grouping has no effect on variable scope or behaviour at runtime.

- To create a group, simply edit a variable and change its "Group" property to some new name.
- To remove a variable from a group, edit it and set the group to blank.
- Use dots to create nested groups: group1.group.

In the following example, variable "ObjName" is in the group "BuildVars.Debug".



See Also

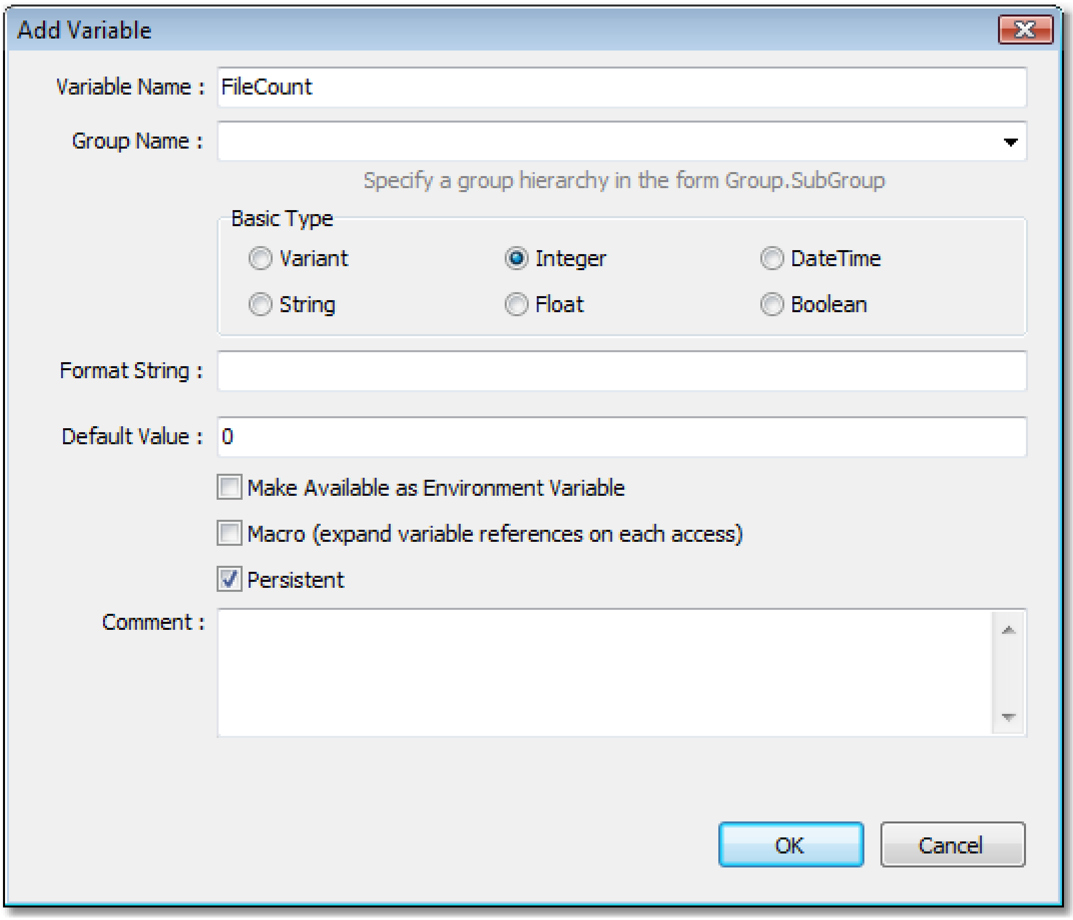
Using Variables
Project Variables
User Variables
Environment Variables
Application Variables
Action List Parameters
Escaping Variable References

5.2 Adding Variables

Variables can be added from one of the following places:

- The Variables Dialog: select **Edit Variables** from the **Tools** menu, or press <Shift+F2> in the main window.
- In the Project Tree, press <Ins> or right click and select **Add Variable**.
- In an action window, press F3 in any text field.

When adding or editing a variable, the following dialog is displayed:



Variable Name : FileCount

Group Name :

○ Variant ● Integer ○ DateTime
○ String ○ Float ○ Boolean

Format String :

Default Value : 0

□ Make Available as Environment Variable

□ Macro (expand variable references on each access)

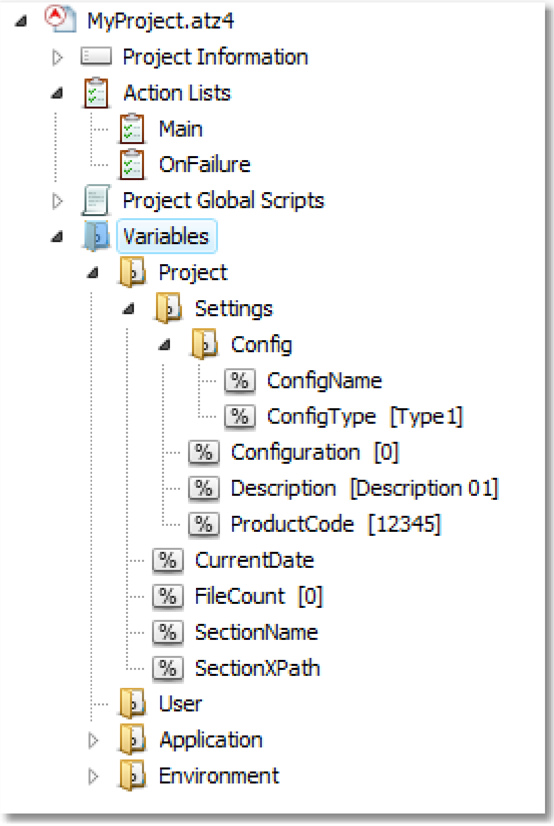
☑ Persistent

Comment :

Variable Name
The name by which you will refer to the variable in your project.
In an action you reference the variable as follows, eg. %BUILDNUM%
In a script you reference the variable by name, without the % signs.

Group Name

Variables can be grouped for display purposes, this makes it easier to find when working with projects that have a lot of variables. Child groups can be created by using dots in the group name, for example in the image below the group name for Var3 is Parent.Child



```
MyProject.atz4
    Project Information
    Action Lists
        Main
        OnFailure
    Project Global Scripts
    Variables
        Project
            Settings
                Config
                    ConfigName
                    ConfigType  [Type1]
                Configuration  [0]
                Description  [Description 01]
                ProductCode  [12345]
            CurrentDate
            FileCount  [0]
            SectionName
            SectionXPath
        User
        Application
        Environment
```

Basic Type

By default, variables in Automise are Variants. Variants can contain various types, which makes them very versatile. They can however be problematic on some occasions. For example if you wanted to store 03 in a variable, the Variant's type inference would deem that to be an integer, and store it as an integer. When you later retrieve that value, you would get 3, not three. Setting the Basic Type to string would resolve this issue, since the value would always be stored and and retrieved as a string.

Format String

When the basic type variable is set to other than Variant, you can specify a format string which will be used when ever the variable's value is evaluated in a text field in Automise. See these topics for valid format strings :

Variable Format Strings

DateTime Format Strings

Default Value
The variable will be set to this default value when a build is started. Once the variable is created, changing the default value does not change the current value. To reset the current value to the default value, use a Reset Variable to Default action.

Make Available as Environment Variable
Project and User Variables can be made available as environment variables to applications that are executed by Automise.

Macro
This will treat the variable as a "macro" or "function". It's value will be re-evaluated

whenever it is referenced and it cannot be set

Persistent
The variable value will be saved between Automise executions.

See the Variables Overview for details on the variable flags.

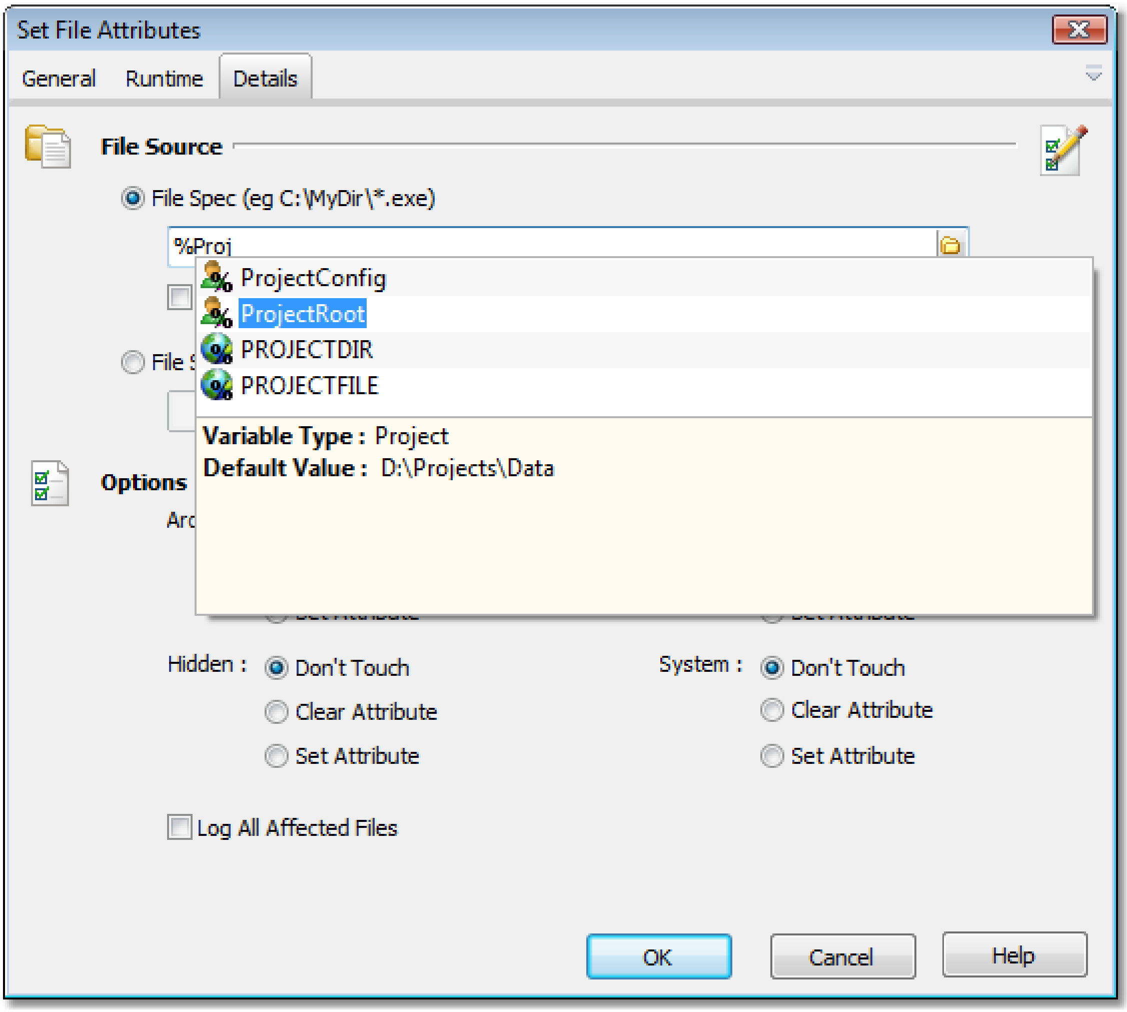
5.3 Using Variables

Variables can be used in most text properties for Actions. To use a Variable in a property, enclose the variable in percent symbols,

eg. : %OUTPUTDIR%

If you need to use a % symbol in a property, make sure you escape it with another %, eg. %%. See Escaping Variable Contents.

A common use for variables is to set file paths, for example to set the File Spec for Set File Attributes:



● File Spec (eg C:\MyDir\*.exe)

%Proj

- ProjectConfig
- ProjectRoot
- PROJECTDIR
- PROJECTFILE

Variable Type : Project
Default Value : D:\Projects\Data

Hidden : ● Don't Touch ○ Clear Attribute ○ Set Attribute

System : ● Don't Touch ○ Clear Attribute ○ Set Attribute

□ Log All Affected Files

Variables can also be used in Script Events.

See also: Variable Sense.

5.4 Variable Evaluation and Recursion

By default, when a Automise variable's value is referenced in a text field (using the %myvariable% syntax), Automise will expand the value of that variable recursively. So if the value of myvariable is %anothervariable% then the value of anothervariable will also be expanded. This will continue until no more variable references are found.

In most cases, the default expansion is exactly what we need, however in some cases you might require that the recursive variable expansion didn't happen. When referencing the variable in a text field, use a ! symbol in front of the name, e.g %!myvariable% - what this does is tell Automise to only expand the variable reference, but leave any variable references in the value of that variable unexpanded.

5.5 Escaping Variable References

You can reference a variable from an action by using percent signs, ie %USERNAME%. See Using Variables.

To insert a percent sign, use %% - ie "%%Hello" will expand to "%Hello".

Some other characters are also escaped:

All Escape Sequences

Character	Escape As
%	%%
#	##
$(	$$(

($(and # are deprecated ways of referring to variables and action list parameters. We recommend you use % wherever possible.)

Disabling Recursive Expansion

If you have a variable whose contents contain characters marked above, then you can use %!USERNAME% to tell Automise not to expand any variable references in the contents.

Example: Suppose you had a project variable MySQL, which contained the text "WHERE Connection LIKE 'G%'". If you reference the variable as %!MySQL%, then the contents will inserted verbatim without being scanned for variable references.

Reading Text Files

The Read Text File action has an "escape any variables" option which will automatically escape any variable references found when reading a file.

5.6 Variable Types

5.6.1 Project Variables

Project Variables are specific to a given project and are stored in the project's .atv file. This makes them available on any machine where the project is run.

5.6.2 User Variables

User Variables are global across all projects, but are specific to a single Windows user. This makes them useful for storing settings that all projects may need, such as paths to programs or machine-specific options. They are stored in an INI file in the following location:

\Documents and Settings\<username>\Application Data\Automise\ATUserVariables.ini

5.6.3 Environment Variables

Automise reads in the system's Environment variables when it starts. These can be used in the same manner as Project and User Variables. Setting an Environment variable can only be done during a run, either with the Set Variable Action or in the Action Script events . The values assigned to Environment variables by Automise during a run are not saved after a run. Automise sets them temporarily and then restores them to their original values at the end of the run.

Environment variables in Automise are also made available to applications that it executes.

Note: If you modify an Environment variable outside of Automise, Automise will not see this change until you restart Automise.

5.6.4 Application Variables

Application Variables are defined by Automise at startup. The actual variables available will depend on which software you have installed on your machine. Application Variables cannot be modified.

These Application Variables are available on all machines :

SYSDIR	The Windows System Directory, eg. 'D:\WINNT\System32'
WINDIR	The Windows Directory - eg. 'D:\WINNT'
COMPUTERNAME	The name of the computer you are running Automise on.
USERNAME	The Name of the currently logged on User
DOSCMD	The path to the Dos command interpreter. This is used internally by Automise.

The following Application Variables are created by Automise:

ISAUTORUN	True if the project is being run automatically from a command line parameter.
ISCONSOLE	True if the project is being run from the console under ATCMD.EXE.
ISINTERACTIVE	True if the project is being run under the Automise IDE.
ISINCLUDEDPROJECT	True if the current project is being run as a child project

	(via the Include Automise Project Action)
PROJECTFILE	The full path to the currently running Automise project file.
PROJECTDIR	The dir where the currently running Automise project lives.
AUTOMISEDIR	The directory in which Automise is installed.
IGNOREDERRORS	The number of actions in error during the build that were ignored.

5.7 Action List Parameters

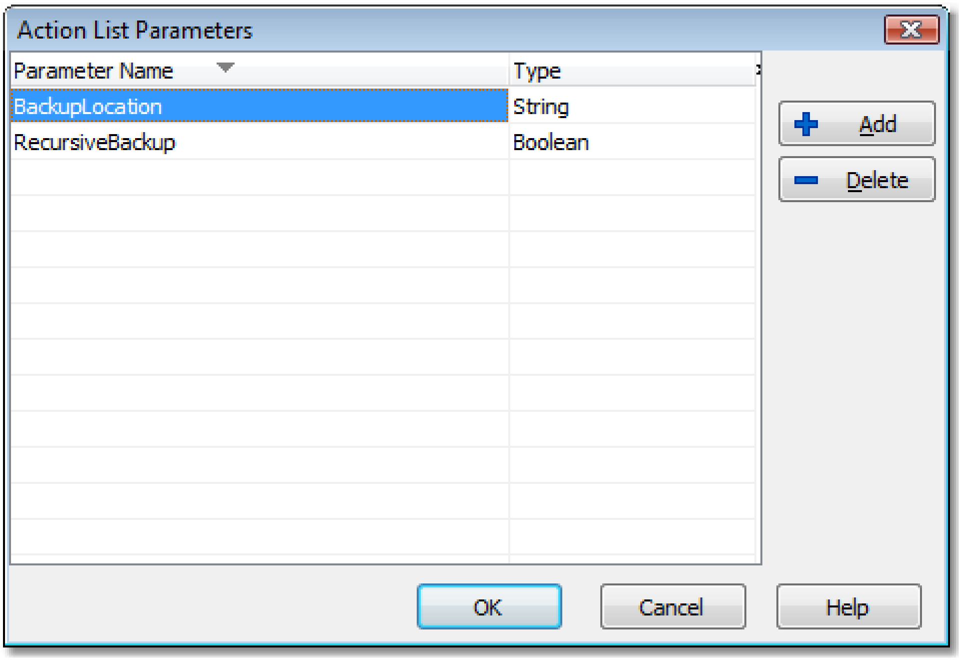
Action List Parameters allow you to make an action list behave as a function of specific values when you call it.

The scope of an action list parameter is confined to the actions within the Action List. Action List Parameters can be modified from script in the same manner as Automise variables.

Action List Parameters can be accessed anywhere that Automise variables can be used. Action List Parameters take precedence over Automisevariables of the same name. The syntax to access an action list parameter outside of script is the same as for a variable, ie %ParameterName%.

Action List Parameters are available on any action list except Main and OnFailure.

To view or modify action list parameters you can either:
1. Right-click on the action list tab and select "Action List Parameters...", or
2. Select the "Action List Parameters..." menu item from the Project menu.

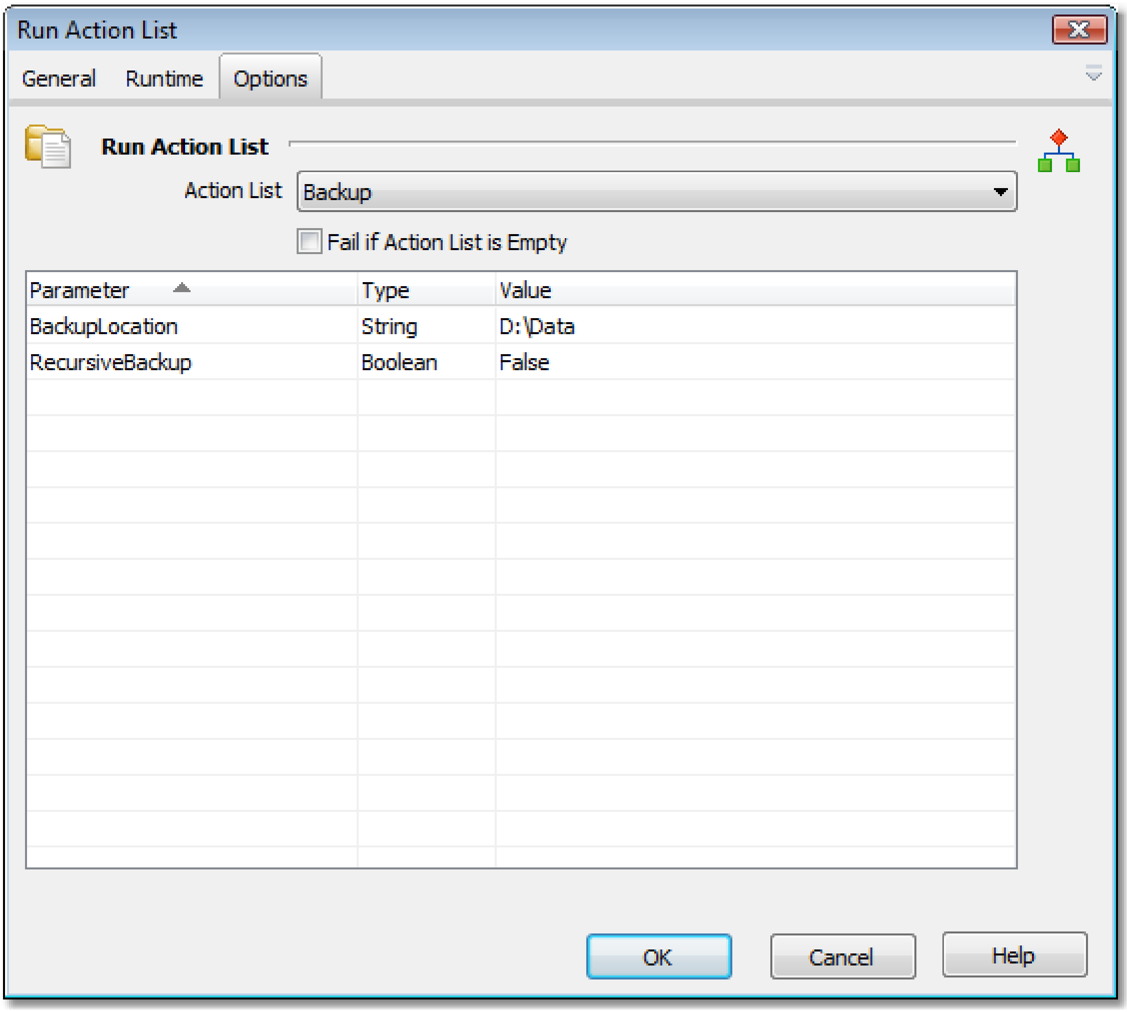


Action List Parameters

Parameter Name	Type
BackupLocation	String
RecursiveBackup	Boolean

In the above example, the variables %SOURCEPATH% and %BUILDNUM% are available to any actions in the action list with these parameters.

To run an action list, use a "Run Action List" action (available in the "Flow Control" action types category). The Run Action List action's property page allows you to specify the value of each action list parameter:



Action List: Backup

□ Fail if Action List is Empty

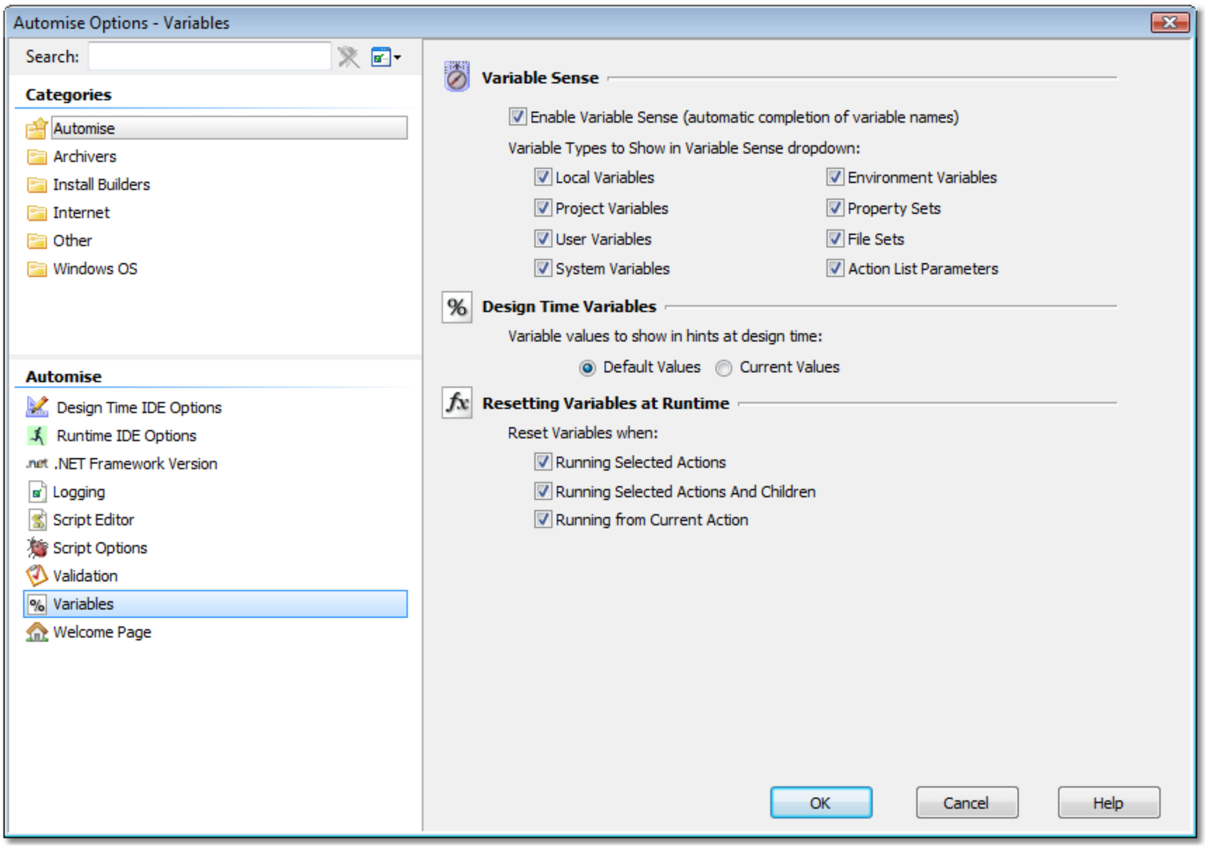
Parameter	Type	Value
BackupLocation	String	D:\Data
RecursiveBackup	Boolean	False

Note:

An alternative way to access Action List Parameters in fields of actions is by using the following syntax: #Parameter#
This syntax is deprecated. To escape this sequence, use an extra #, eg. ##Parameter#.

5.8 Variable Options

To edit Variable Options, go to Tools -> Options and click Variables in the General category.



Automise Options - Variables

Search:

Categories

- Automise
- Archivers
- Install Builders
- Internet
- Other
- Windows OS

Automise

- Design Time IDE Options
- Runtime IDE Options
- .NET Framework Version
- Logging
- Script Editor
- Script Options
- Validation
- Variables
- Welcome Page

Variable Sense

☑ Enable Variable Sense (automatic completion of variable names)

Variable Types to Show in Variable Sense dropdown:

☑ Local Variables ☑ Environment Variables
☑ Project Variables ☑ Property Sets
☑ User Variables ☑ File Sets
☑ System Variables ☑ Action List Parameters

Design Time Variables

Variable values to show in hints at design time:

● Default Values ○ Current Values

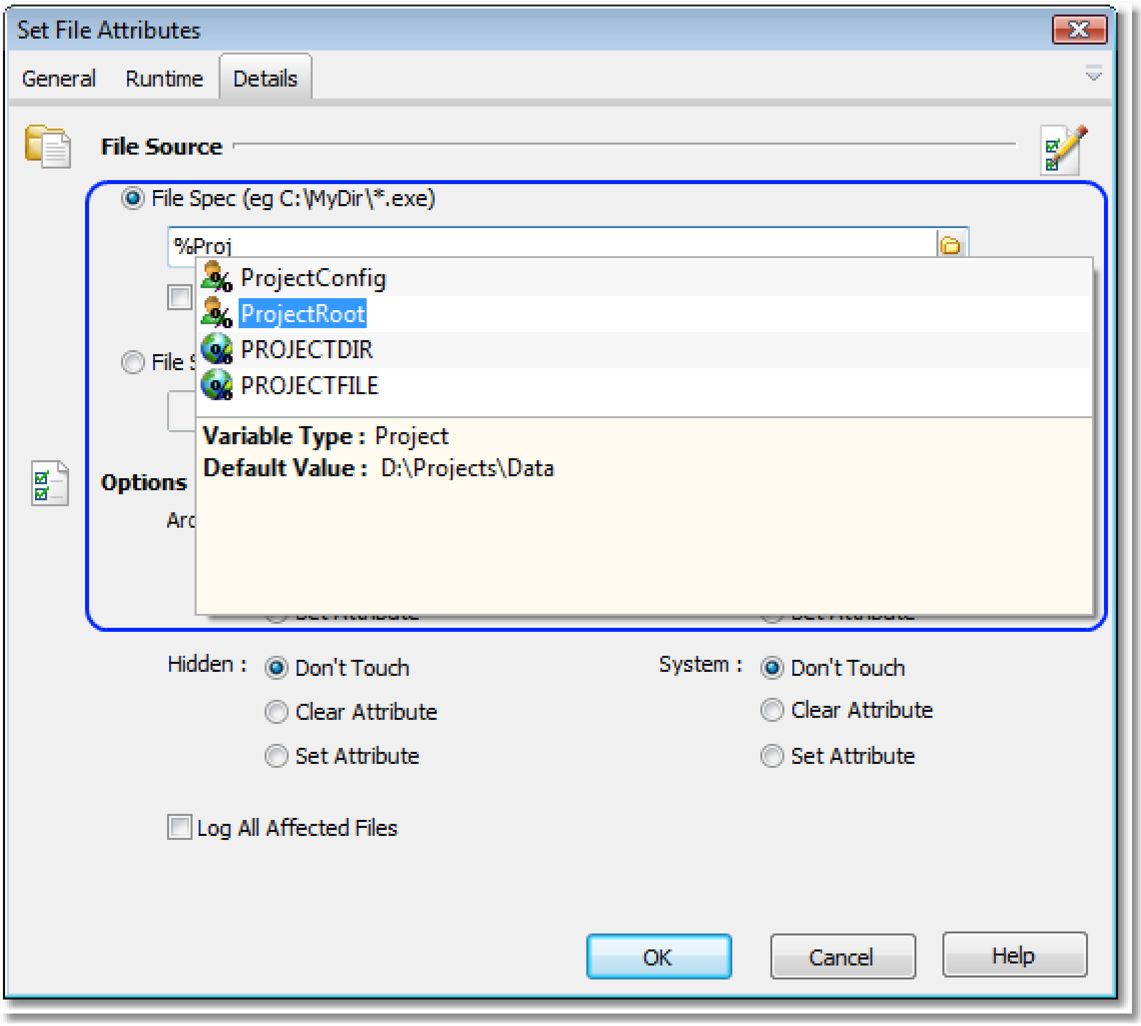
Resetting Variables at Runtime

Reset Variables when:

☑ Running Selected Actions
☑ Running Selected Actions And Children
☑ Running from Current Action

Variable Sense

Variable Sense allows you to see a drop-down list with possible completions for variable names:



Variable Type : Project
Default Value : D:\Projects\Data

Hidden : ● Don't Touch ○ Clear Attribute ○ Set Attribute

System : ● Don't Touch ○ Clear Attribute ○ Set Attribute

□ Log All Affected Files

The list opens automatically when you type % to begin entering a variable name. You can control this behaviour as follows.

"**Enable Variable Sense**"

Uncheck this checkbox if you don't want to see Variable Sense drop-downs when typing variable names.

"**Variable Types To Show...**"

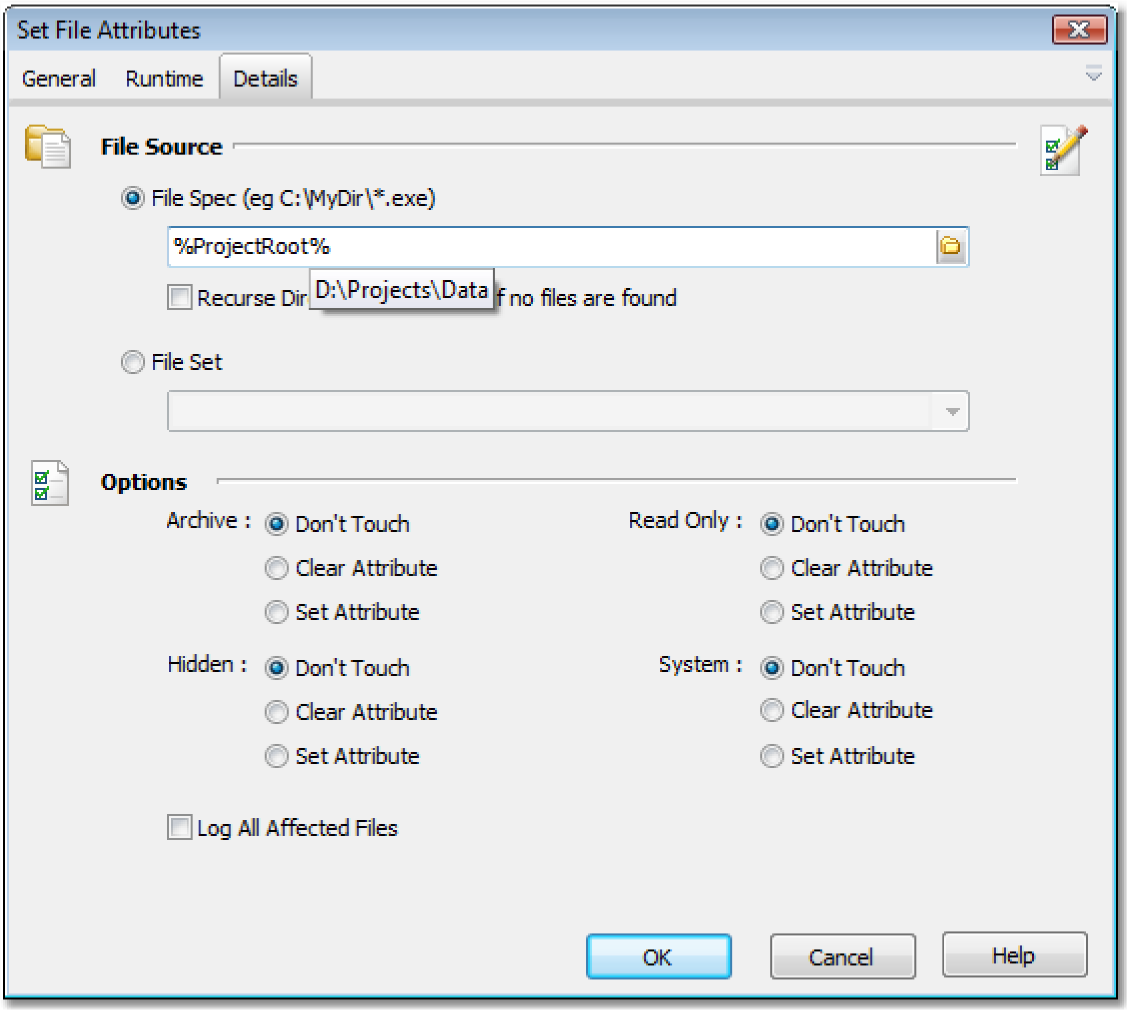
Select the Variable Types that you wish to show in the Variable Sense drop-down.

By default, all variable types are shown, however you can narrow this to only include certain variable types, and/or other items such as File Sets and Action List Parameters.

Design Time Variables

"Variable Values to show in hints at design time"

When you hover your mouse over an text field or memo box on an action's property page, you will see a hint field with all of the variable references in that field expanded:



Set File Attributes

General | Runtime | Details

File Source

● File Spec (eg C:\MyDir\*.exe)

%ProjectRoot%

□ Recurse Dir D:\Projects\Data f no files are found

○ File Set

Options

Archive : ● Don't Touch ○ Clear Attribute ○ Set Attribute

Read Only : ● Don't Touch ○ Clear Attribute ○ Set Attribute

Hidden : ● Don't Touch ○ Clear Attribute ○ Set Attribute

System : ● Don't Touch ○ Clear Attribute ○ Set Attribute

□ Log All Affected Files

You can choose to use either the Variables' Default Values, or their Current Values, during this expansion.

Resetting Variables at Runtime

Non-persistent variables are reset to their default values each time a project is run. By selecting these options, you can also choose whether they are reset when running part of a project.

5.9 Variable Format Strings

Format Strings

Note : For Date Format String see this topic : DateTime Format Strings

Format strings specify required formats to general-purpose formatting routines. Format strings passed to the string formatting routines contain two types of objects--literal characters and format specifiers. Literal characters are copied word for word to the resulting string. Format specifiers fetch arguments from the argument list and apply the formatting to them.
Format specifiers have the following form:
"%" [index ":"] ["-"] [width] ["." prec] type

A format specifier begins with a % character. After the percent sign come the following elements, in this order:

1. An optional argument zero-offset index specifier (that is, the first item has index 0), [index ":"].
2. An optional left justification indicator, ["-"].
3. An optional width specifier, [width].
4. An optional precision specifier, ["." prec].
5. The conversion type character, type.

The following table summarizes the possible values for type.

Value	Meaning
d	Decimal. The argument must be an integer value. The value is converted to a string of decimal digits. If the format string contains a precision specifier, it indicates that the resulting string must contain at least the specified number of digits; if the value has less digits, the resulting string is left-padded with zeros.
u	Unsigned decimal. Similar to "d", but no sign is output.
e	Scientific. The argument must be a floating-point value. The value is converted to a string of the form "-d.ddd...E+ddd". The resulting string starts with a minus sign if the number is negative. One digit always precedes the decimal point. The total number of digits in the resulting string (including the one before the decimal point) is given by the precision specifier in the format string; a default precision of 15 is assumed if no precision specifier is present. The "E" exponent character in the resulting string is always followed by a plus or minus sign and at least three digits.
f	Fixed. The argument must be a floating-point value. The value is converted to a string of the form "-ddd.ddd...". The resulting string starts with a minus sign if the number is negative. The number of digits after the decimal point is given by the precision specifier in the format string- a default of 2 decimal digits is assumed if no precision specifier is present.
g	General. The argument must be a floating-point value. The value is converted to the shortest possible decimal string using fixed or scientific format. The number of significant digits in the resulting string is given by the precision specifier in the format string; a default precision of 15 is assumed if no precision specifier is present. Trailing zeros are removed from the resulting string, and a decimal point appears only if necessary. The resulting string uses the fixed-point format if the number of digits to the left of the decimal point in the value is less than or equal to the specified precision, and if the value is greater than or equal to 0.00001. Otherwise the resulting string uses scientific format.
n	Number. The argument must be a floating-point value. The value is converted to a string of the form "-d,ddd,ddd.ddd...". The "n" format corresponds to the "f" format, except that the resulting string contains thousand separators.
s	String. The argument must be a string value. The string or character is inserted in place of the format specifier. The precision specifier, if present in the format string, specifies the maximum length of the resulting string. If the argument is a string that is longer than this maximum, the string is truncated.
x	Hexadecimal. The argument must be an integer value. The value is converted to a string of hexadecimal digits. If the format string contains a precision specifier, it indicates that the resulting string must contain at least the specified number of digits; if the value has fewer digits, the resulting string is left-padded with zeros.

Conversion characters may be specified in uppercase as well as in lowercase; both produce the same results.

For all floating-point formats, the actual characters used as decimal and thousand separators are obtained from the DecimalSeparator and ThousandSeparator which is set for the user at the operating system level.

Index, width, and precision specifiers can be specified directly, using a decimal digit string (for example "%10d"). Note that width is an integer value, while precision is an unsigned integer value. For example,

Format ('%8.2f', [123.456]);

A width specifier sets the minimum field width for a conversion. If the resulting string is shorter than the minimum field width, it is padded with blanks to increase the field width. The default is to right-justify the result by adding blanks in front of the value, but if the format specifier contains a left-justification indicator (an "-" en dash character preceding the width specifier), the result is left-justified by adding blanks after the value.

5.10 DateTime Format Strings

Date Time Format Strings

Specifier Displays
c Displays the date using the format given by the ShortDateFormat global variable, followed by the time using the format given by the LongTimeFormat global variable. The time is not displayed if the date-time value indicates midnight precisely.

d Displays the day as a number without a leading zero (1-31).
dd Displays the day as a number with a leading zero (01-31).
ddd Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
ddddd Displays the date using the format given by the ShortDateFormat global variable.
dddddd Displays the date using the format given by the LongDateFormat global variable.
e (Windows only) Displays the year in the current period/era as a number without a leading zero (Japanese, Korean and Taiwanese locales only).
ee (Windows only) Displays the year in the current period/era as a number with a leading zero (Japanese, Korean and Taiwanese locales only).
g (Windows only) Displays the period/era as an abbreviation (Japanese and Taiwanese locales only).
gg (Windows only) Displays the period/era as a full name. (Japanese and Taiwanese locales only).
m Displays the month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm Displays the month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNames global variable.
mmmm Displays the month as a full name (January-December) using the strings given by the LongMonthNames global variable.
yy Displays the year as a two-digit number (00-99).
yyyy Displays the year as a four-digit number (0000-9999).
h Displays the hour without a leading zero (0-23).
hh Displays the hour with a leading zero (00-23).
n Displays the minute without a leading zero (0-59).

nn Displays the minute with a leading zero (00-59).
s Displays the second without a leading zero (0-59).
ss Displays the second with a leading zero (00-59).
z Displays the millisecond without a leading zero (0-999).
zzz Displays the millisecond with a leading zero (000-999).
t Displays the time using the format given by the ShortTimeFormat global variable.
tt Displays the time using the format given by the LongTimeFormat global variable.
am/pm Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm Uses the 12-hour clock for the preceding h or hh specifier, and displays the contents of the TimeAMString global variable for any hour before noon, and the contents of the TimePMString global variable for any hour after noon.
/ Displays the date separator character given by the DateSeparator global variable.
: Displays the time separator character given by the TimeSeparator global variable.
'xx'/"xx" Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

6 Actions Reference

What are Actions?

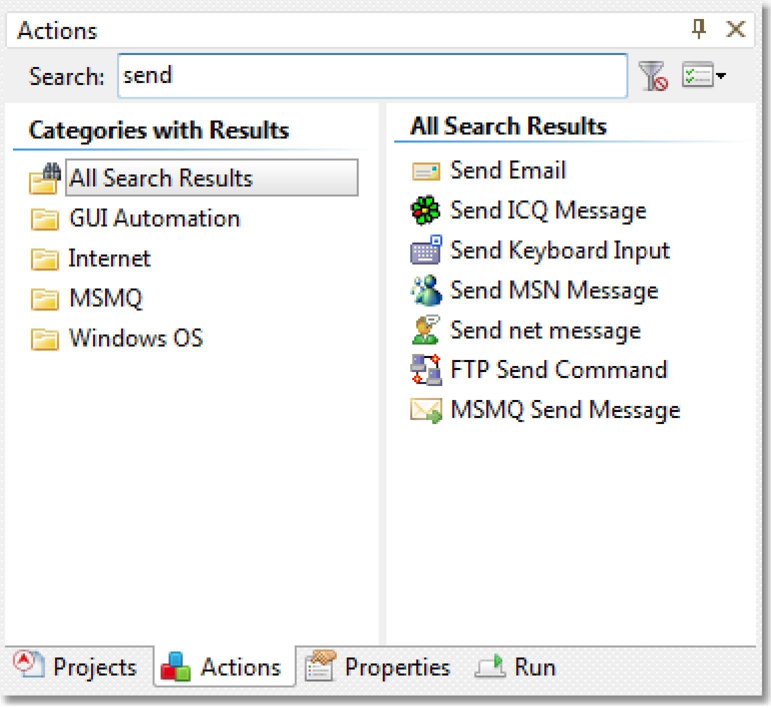
Actions are the cornerstone of Automise. Automise has built in actions to do things such as Copy Files, Move files, run programs etc. In addition, Automise has actions to interface with a number of third party products such as IIS, PsTools and 7-Zip.

Action Lists

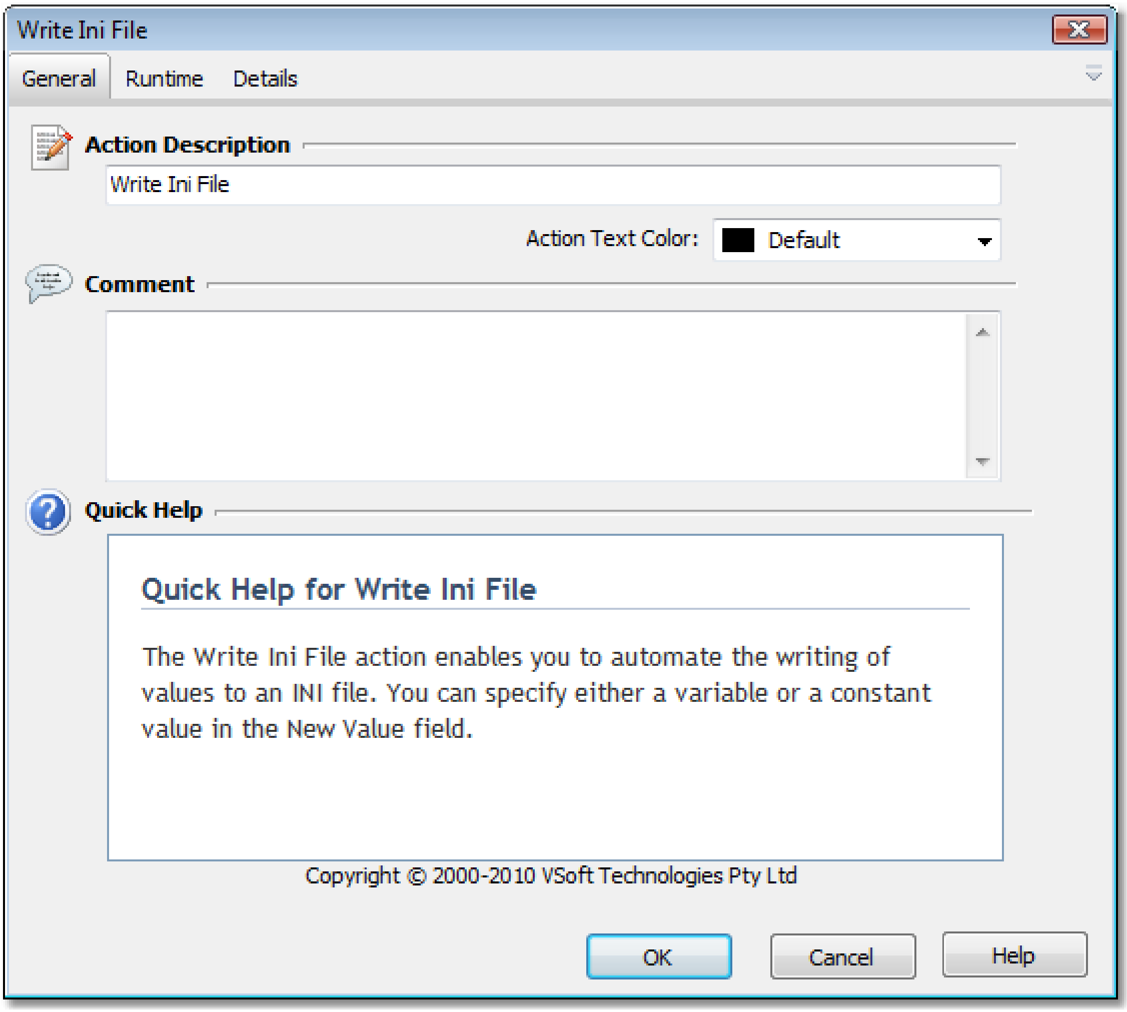
Automise has two default Action Lists that you can add actions to, the Main Action List and the OnFailure Action List. When you are creating your project, you will start off by adding actions to the Main Action list. See the Action Lists topic for more information.

Using Actions

To add an action to the action list, select an action from the Action Types frame (on the left hand side of the main window) by clicking on the action type you wish to add.



After clicking on the action type, by default the properties dialog will be displayed:



Write Ini File

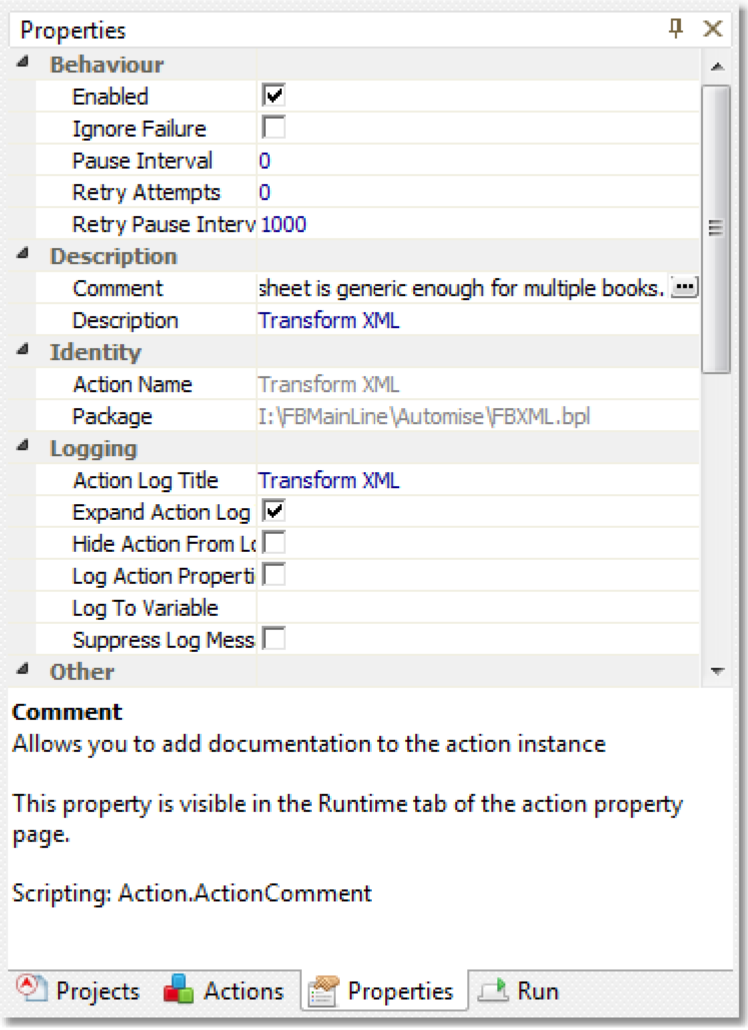
Quick Help for Write Ini File

The Write Ini File action enables you to automate the writing of values to an INI file. You can specify either a variable or a constant value in the New Value field.

Copyright © 2000-2010 VSoft Technologies Pty Ltd

All actions have a common set of properties, displayed on the Properties tab. The number of additional tabs in this dialog will depend on which action type you selected.

In addition to the properties window, you can also edit most of each action's properties in the Action Inspector tab.



Behaviour	
Enabled	☑
Ignore Failure	□
Pause Interval	0
Retry Attempts	0
Retry Pause Interv	1000
Description	
Comment	sheet is generic enough for multiple books.
Description	Transform XML
Identity	
Action Name	Transform XML
Package	I:\FBMainLine\Automise\FBXML.bpl
Logging	
Action Log Title	Transform XML
Expand Action Log	☑
Hide Action From Lo	□
Log Action Properti	□
Log To Variable	
Suppress Log Mess	□
Other	

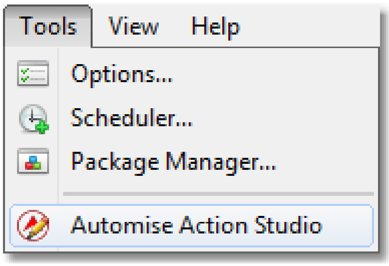
Comment
Allows you to add documentation to the action instance

This property is visible in the Runtime tab of the action property page.

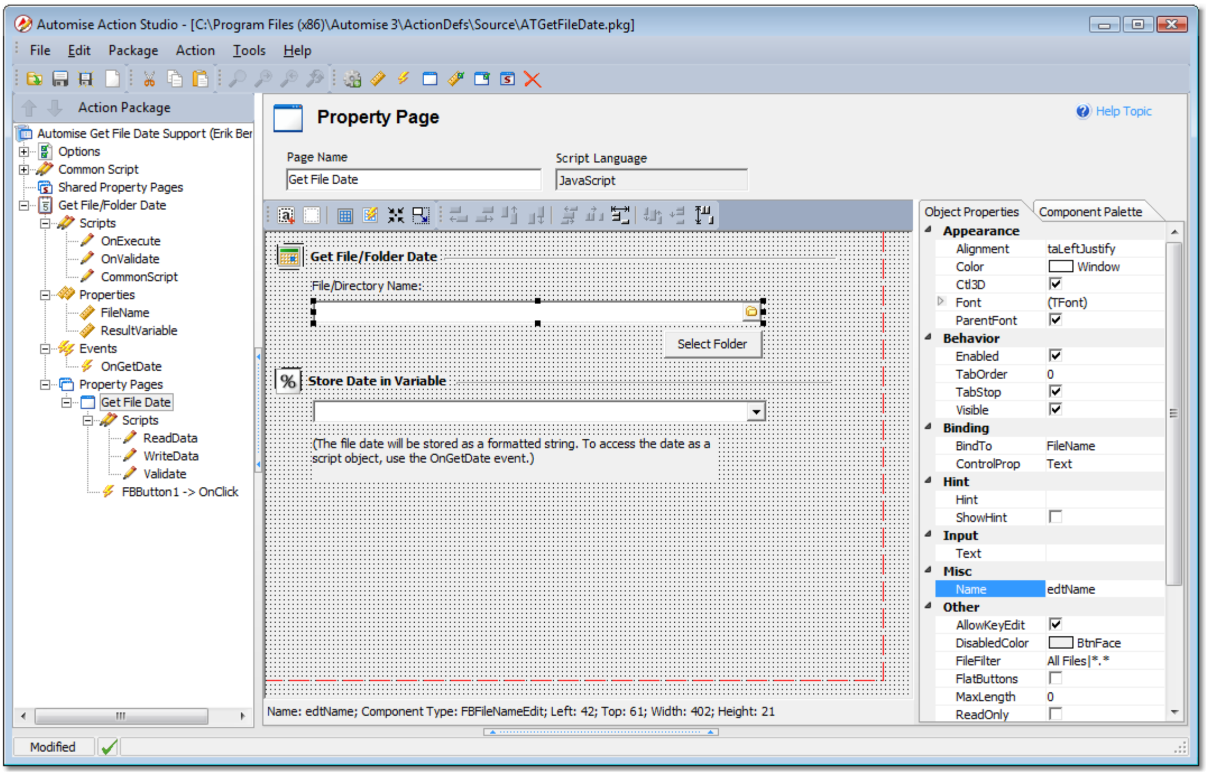
Scripting: Action.ActionComment

6.1 Custom Actions

Automise allows you to create and use your own actions. To create Custom Actions use ActionStudio, which is available from the Start menu, or from the Tools menu in Automise.



ActionStudio is a fully featured IDE that lets you easily create custom actions with graphic option page editor, script editor and more. You can create Actions using PowerShell, JavaScript, VBScript and any .NET language. For more information on creating custom actions, please see the interactive help inside ActionStudio, or the ActionStudioManual.pdf file which is installed in your Automise main directory.



Page Name: Get File Date

Script Language: JavaScript

(The file date will be stored as a formatted string. To access the date as a script object, use the OnGetDate event.)

Appearance	
Alignment	taLeftJustify
Color	□ Window
Ctl3D	☑
Font	(TFont)
ParentFont	☑
Behavior	
Enabled	☑
TabOrder	0
TabStop	☑
Visible	☑
Binding	
BindTo	FileName
ControlProp	Text
Hint	
Hint	
ShowHint	□
Input	
Text	
Misc	
Name	edtName
Other	
AllowKeyEdit	☑
DisabledColor	□ BtnFace
FileFilter	All Files|*.*
FlatButtons	□
MaxLength	0
ReadOnly	□

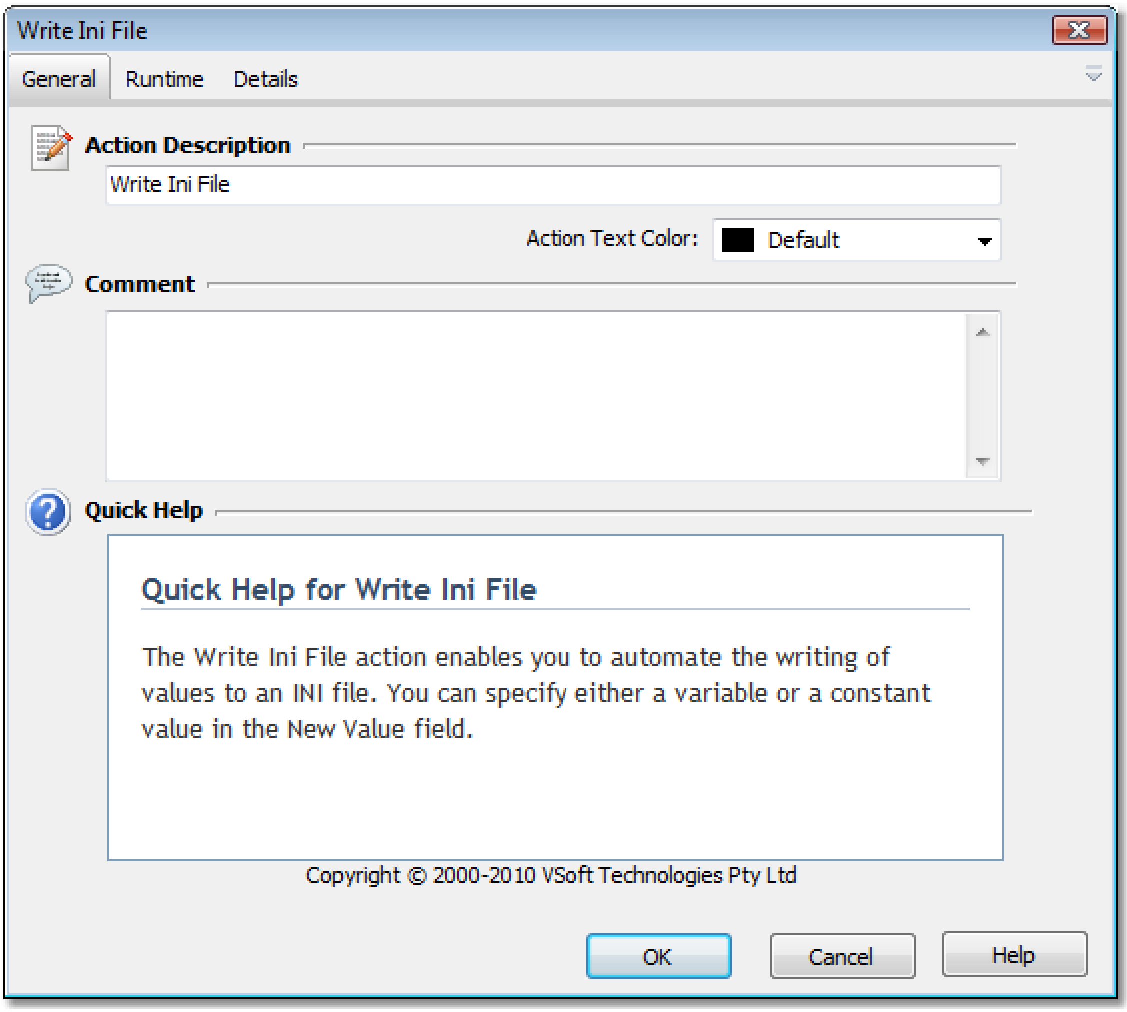
Name: edtName; Component Type: FBFileNameEdit; Left: 42; Top: 61; Width: 402; Height: 21

6.2 Common Action Properties

All Automise actions have two default property pages - General and Runtime.

General Property Page

This property page contains general settings, such as the Action Description, Quick Help, and Comment.



Action Description

Write Ini File

Action Text Color: ■ Default

Comment

Quick Help

Quick Help for Write Ini File

The Write Ini File action enables you to automate the writing of values to an INI file. You can specify either a variable or a constant value in the New Value field.

Copyright © 2000-2010 VSoft Technologies Pty Ltd

Action Description
Give the action a meaningful but short description. This will be displayed in the IDE and in the build log. Some actions will automatically update their default name with some contextual information (ie. Run Action List will append the name of the action list to run.)

Tip**:** Clearing an action's description will reset it to the default (automatic) description.

Text Color
The color given to the text of the action. Changing the color of the text can help you make a particular action stand out.

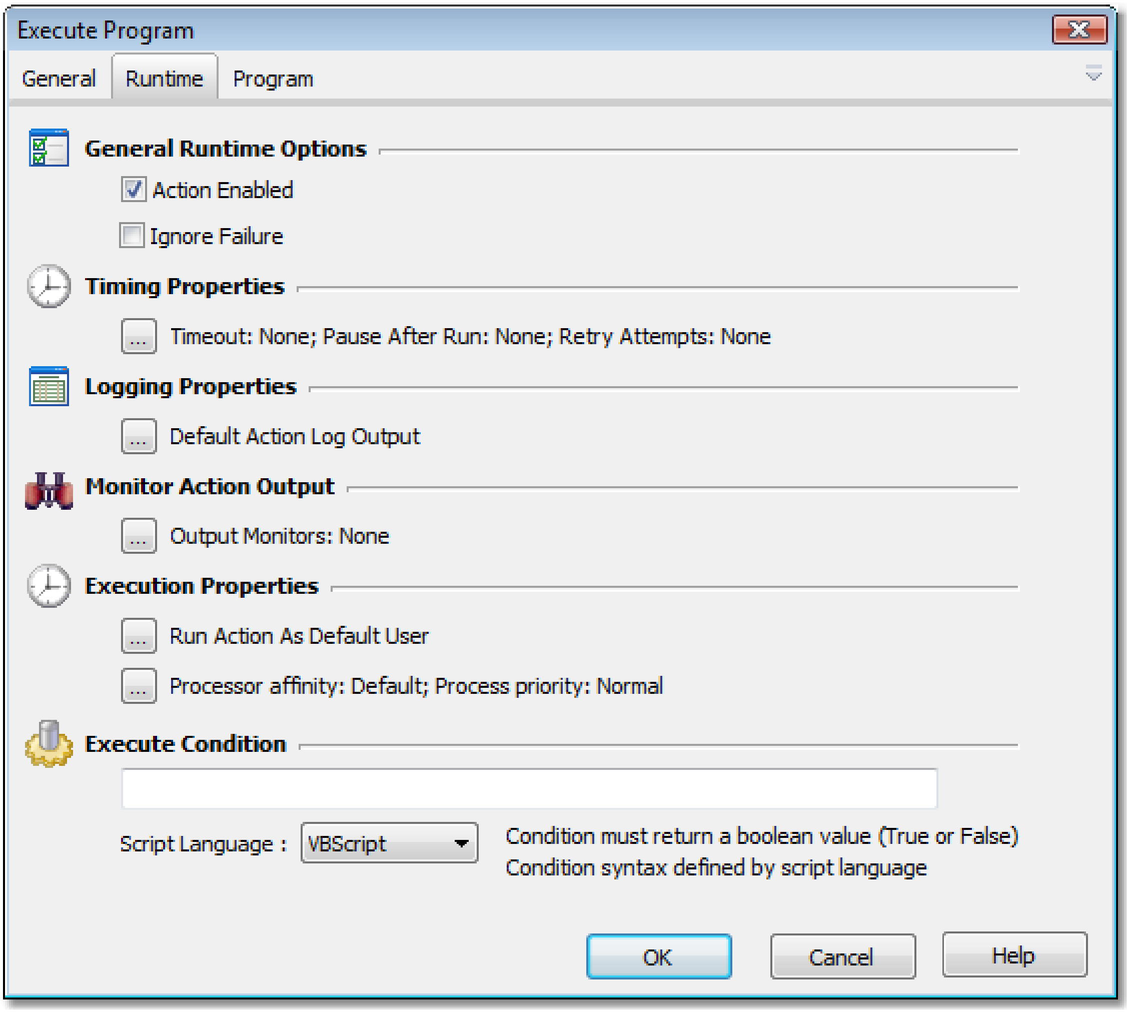
Comment
A place to store notes to yourself or your team members about the action. If the action has a comment, then an indicator icon will appear in the action list against the action.

Quick Help
If the action has a quick help file, then the quick help is displayed in this area. You should refer to the help file for more detailed help for the action.

Runtime Property Page

This property page contains settings which take effect when the action is executing.

Many of the properties on this page are edited by clicking the Edit buttons (marked ...) on the left-hand side of the page.



Execute Program

General Runtime Options

☑ Action Enabled

□ Ignore Failure

Timing Properties

[...] Timeout: None; Pause After Run: None; Retry Attempts: None

Logging Properties

[...] Default Action Log Output

Monitor Action Output

[...] Output Monitors: None

Execution Properties

[...] Run Action As Default User

[...] Processor affinity: Default; Process priority: Normal

Execute Condition

Script Language : VBScript

Condition must return a boolean value (True or False)
Condition syntax defined by script language

Action enabled
Disable an action to have it ignored when the project is running. Disabled actions are shown grayed out in italics.

Ignore failure
Even if this action fails, the run will continue. Some actions (such as Run Action List) cannot Ignore Failure. You can use the "If Prev Action Failed" action immediately after an action with Ignore Failure turned on to detect if the action did fail. For more sophisticated control of failed actions, use Try/Catch blocks.

Timing Properties

Timing properties let you set an action timeout length, pause after run length, and retry attempts. Click on the ellipsis (...) button to edit these settings.

Logging Properties

Logging Properties allow you to suppress log output, hide an action from the log, log to a variable, or record action properties in the log. Click on the ellipsis (...) button to edit these settings.

Monitor Action Output

Action Output Monitors perform monitoring on this action's output. Click on the ellipsis (...) button to edit action monitors.

Execution Properties

This section allows you to edit the user that the action runs as, and also the processor affinity and priority for the action. Click on the relative ellipsis (...) button to edit these options.

These options are only available for actions which run external executables.

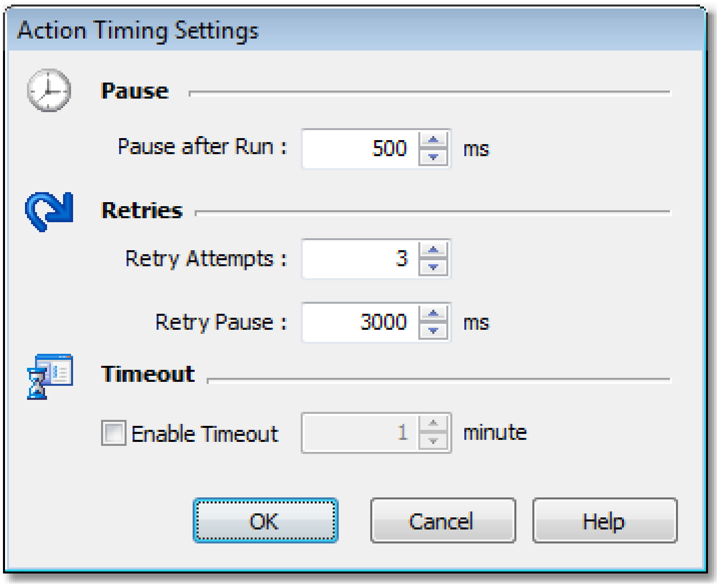
See the property pages for Run Action as User, and Setting Processor Affinity and Priority, for more information.

Execute Condition

Actions can be set to only execute if a certain condition holds. Enter the condition as a boolean expression in either VBScript, JavaScript or PowerShell. See Execute Condition for more details.

6.2.1 Timing Properties

This property dialog allows you to edit properties relating to an action's timeout, pause interval and retry settings. This dialog appears when you click the ellipsis (...) edit button on the Runtime property page.



Pause after Run : 500 ms

Retry Attempts : 3

Retry Pause : 3000 ms

☐ Enable Timeout 1 minute

Pause after Run

After the action completes, the project will pause for the specified period of time after this action completes before continuing on to the next action.

Retry Attempts

If the action fails, it will be retried this many times before failing outright.

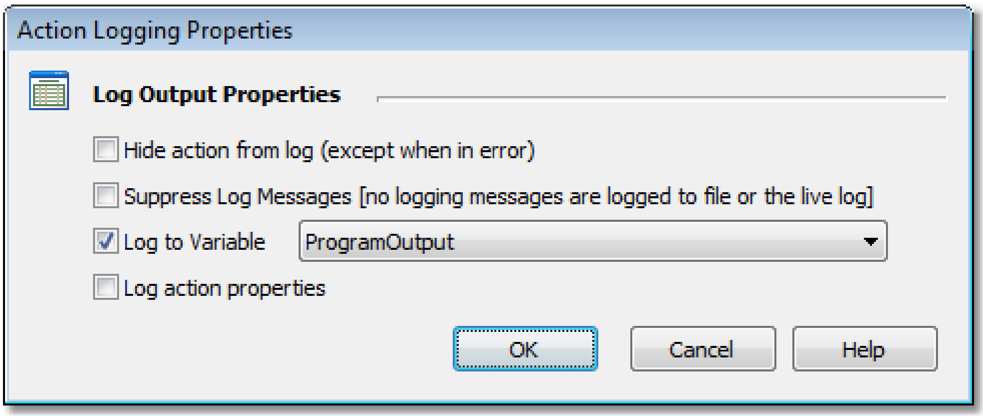
Set the Retry Pause to delay between each retry. Setting Retry on "Try" actions allows you to retry an entire group of actions as one. See the Try action topic for more details.

Enable Timeout

If this properties is enabled and the action runs for longer than the specified number of minutes, it will be aborted and will fail. This property is not available for all actions.

6.2.2 Logging Properties

This property dialog allows you to edit properties relating to an action's log output. This dialog appears when you click the ellipsis (...) edit button on the Runtime property page.



Action Logging Properties

Log Output Properties

□ Hide action from log (except when in error)

□ Suppress Log Messages [no logging messages are logged to file or the live log]

☑ Log to Variable ProgramOutput

□ Log action properties

Hide action from log

If this property is enabled, the action not appear at all in the Automise log file at all, unless it fails.

Suppress Log Messages

If this property is enabled, none of the messages output by the action will be logged in the log file or the log view. The action itself still appears, though (unless Hide action from log is also enabled.) Use this property with care as the log messages will never appear anywhere. Consider not enabling this option, but instead filtering the log view by status message type. This property can be useful where an action outputs a lot of output and will cause performance or space problems with the log file and build engine.

Log to Variable

If this property is enabled, the action's log output will be written to the Automise variable specified. This option is not available is "Suppress Log Messages" is enabled.

Log action properties

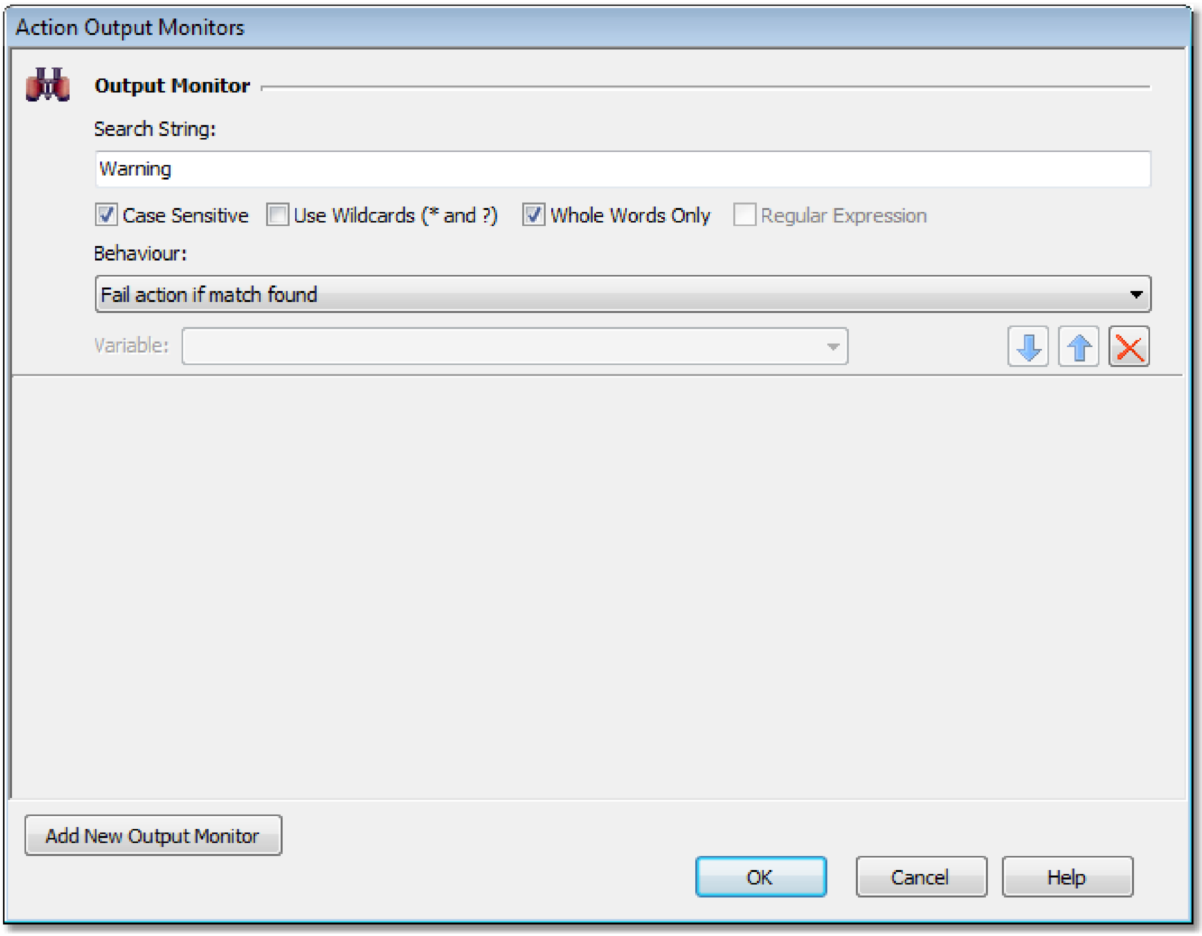
If this property is enabled, then all of the action's properties will be written to the log before the action begins running. The properties will be written in a special log message group called "Action Properties". This is useful for debugging. This option is not available if "Suppress Log Messages" is enabled.

6.2.3 Action Output Monitors

Action Output Monitors allow you to automatically scan an action's output for certain content, and perform various actions if that content is found. For example, you can have an action fail if text like "warning" is found - even though normally that action would succeed.

Only output created while the action is running is monitored - output from the BeforeAction and AfterAction script events is not monitored.

To edit Action Output Monitors, go to the Runtime property page and click on the ellipsis (...) edit button for Output Monitors.



Search String: Warning

☑ Case Sensitive ☐ Use Wildcards (* and ?) ☑ Whole Words Only ☐ Regular Expression

Behaviour: Fail action if match found

Variable:

Click the "Add New Output Monitor" button to add an output monitor to the action. There is no limit on the number of monitors an action can have. All output monitors operate independently of each other.

Click "Remove Monitor" to remove a monitor from the action.

Each monitor has a group of settings:

Search String

This is the string to search for in the action output. In the example above, the monitor will match any line that contains the string "Warning :".

Note that matches cannot be made over multiple lines of output. If you need to match across multiple lines, it is recommended that you log the action output to a variable and use the Text Find/Replace action.

Case Sensitive

If this box is checked, output matching will be case sensitive.

Use Wildcards

If this box is checked, the search string can contain the wildcard characters "*" and "?". * will match any number of characters (including none), whereas ? will match any single character.

If using wildcards, the characters * and ? can be escaped as ** and ??, respectively.

Whole Words Only

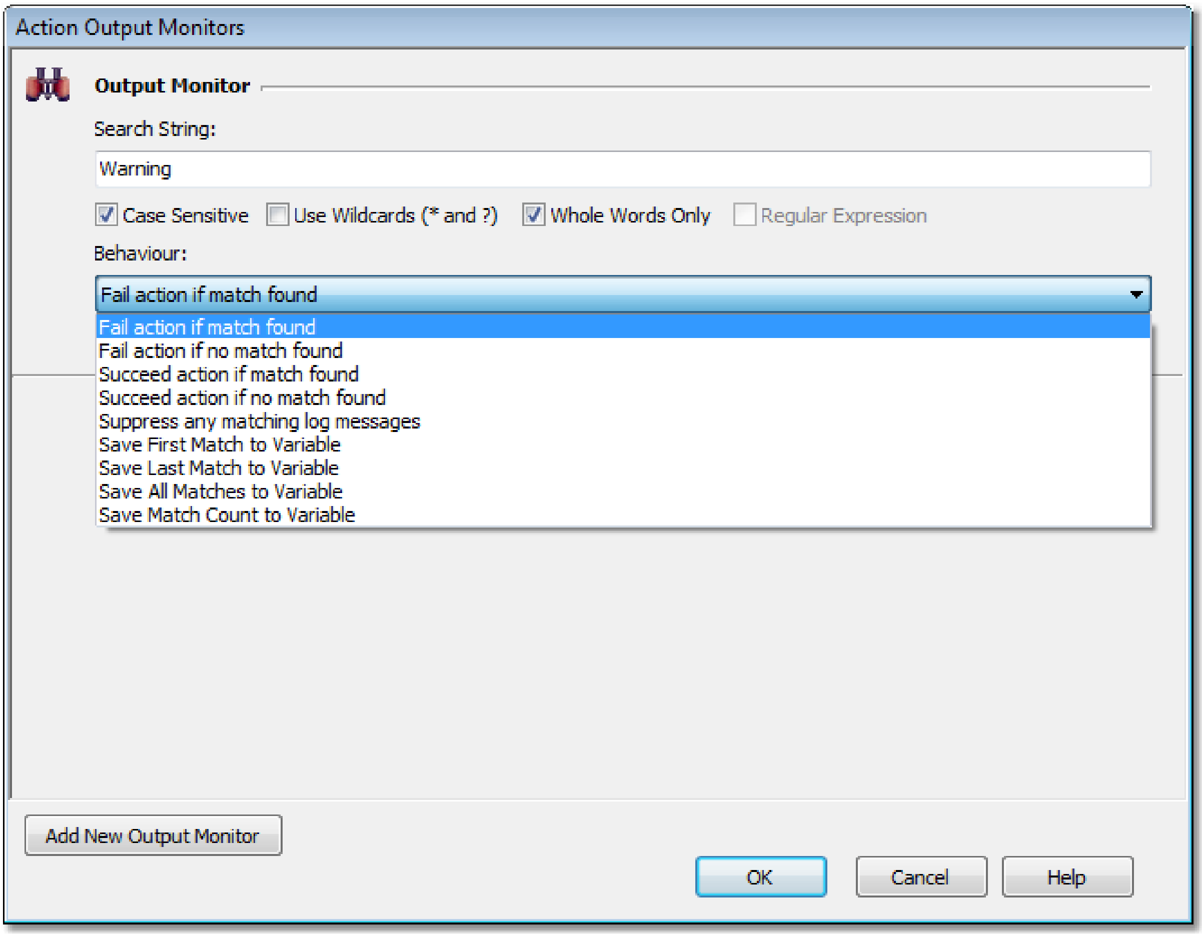
If this box is checked, the search string must match a whole word - not part of a word.

Regular Expression

If this box is checked, the search string is treated as a regular expression. This checkbox cannot be used along with "Use Wildcards" or "Whole Words Only".

Behaviour

The Behaviour dropdown menu allows you to choose what happens if this monitor matches some output from the action.



Action Output Monitors

Output Monitor

Search String:

Warning

☑ Case Sensitive □ Use Wildcards (* and ?) ☑ Whole Words Only □ Regular Expression

Behaviour:

Fail action if match found

Fail action if match found
Fail action if no match found
Succeed action if match found
Succeed action if no match found
Suppress any matching log messages
Save First Match to Variable
Save Last Match to Variable
Save All Matches to Variable
Save Match Count to Variable
Add New Output Monitor
OK
Cancel
Help

Fail action if match found

If the search string is matched then the action will always fail, regardless of the action result status. Any matched lines will be highlighted as "Error" output.(*)

Fail action if no match found

The action will always fail (regardless of result status), unless at least one match for the search string is found.(*)

Succeed action if match found

If the search string is matched then the action will always succeed, regardless of the action result status. Any matched lines will be highlighted as "Success" output.(*)

Succeed action if no match found

The action will always succeed (regardless of result status), unless at least one match for the search string is found.(*)

Suppress any matching log messages

Any line of output which matches will be suppressed from the action's log output.

Save First / Last Match to Variable

(For these behaviours, a dropdown appears allowing you to choose a variable name.)

The line which matches the search string will be written to a variable when the action completes. In the case of multiple matches, "First Match" means that only the first match will be written, "Last Match" means only the last match.

If there are no matching lines, the variable is not written to.

Save All Matches To Variable

(For this behaviour, a dropdown appears allowing you to choose a variable name.)

Any string which matches the search string will be written to a variable. In the case of multiple matches, each match will be written on a new line. If there are no matches, the variable will be set to an empty string.

Save Match Count To Variable

(For this behaviour, a dropdown appears allowing you to choose a variable name.)

The total number of matching lines will be written to a variable.

(*) Note: Using Multiple Monitors with Fail/Succeed

If there are multiple monitors which force the action to both fail and succeed, and both find matches in the action is output, then the monitors are applied in the order they are shown in the dialog box, ie if a "Succeed" monitor is triggered followed by a "Fail" monitor, the action will fail. The action log output will show the status of both monitors, in order.

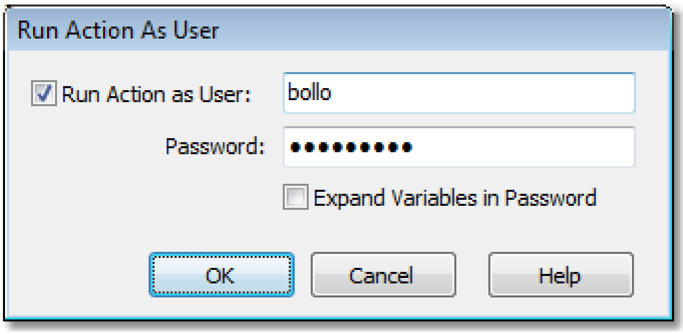
Note that the AfterAction script event has the opportunity to futher modify the action result.

Note: Monitors Are Applied Live

Monitoring is applied to the output immediately as it is received from the action, before the OnStatusMessage script event. However, results (like setting variables) are not applied until the action finishes running. If you use variable references in the search string, be aware that the variable values may change as the action runs.

6.2.4 Run Action as User

This property dialog allows you to choose run an action as another user. This option is only available for actions which run an external process. The dialog appears when you click the ellipsis (...) edit button on the Runtime property page.



Run Action As User

☑ Run Action as User: bollo

Password: •••••••••

□ Expand Variables in Password

Run Action As User

By setting the Username and Password, Automise will attempt to run this particular action under a different user account. There are many security requirements that need to be met before Windows will allow this, the minimum requirements are detailed below. You may need to ask your System Administrator if you still cannot get past the security errors.

The following security privileges must be granted for a process to be run under a different users login:

Windows 2000:
- Act as part of the operating system
- Create a token object
- Replace a process-level token
- Increase quotas

Windows XP and Server 2003
- Act as part of the operating system
- Create a token object
- Replace a process-level token
- Adjust memory quotas for a process

Note that changing security privileges may compromise system security policies. Please ensure that your system is secure before and after modifying these settings.

One way of testing that the security requirements are met and that your username and password are correct is to use the Execute Program action to run the WhoAmI.exe program which is usually located in C:\Windows\System32\WhoAmI.exe

When this is run under a different user, the log will contain the username that the action successfully ran under.

Troubleshooting:

Invalid username or password

If the current user is not a domain account, and the impersonation account is set to a domain account then authentication fails.

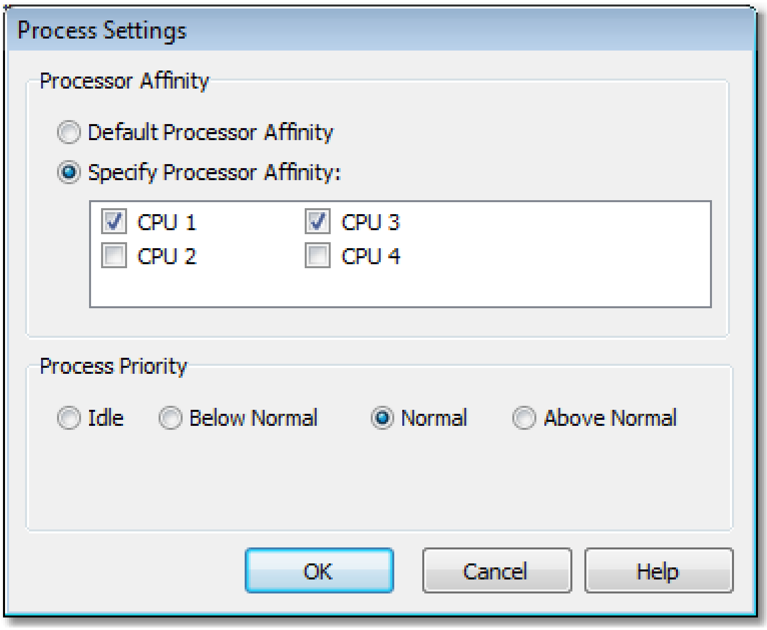
The application failed to initialize properly (0xc0000142). Click on OK to terminate the application.

This message box appears if the user who you are impersonating has incorrect privileges.

Try making the user part of the administration group to solve this problem.

6.2.5 Setting Processor affinity & process priority

This property dialog allows you to choose processor affinity and process priority settings. This option is only available for actions which run an external process. The dialog appears when you click the ellipsis (...) edit button on the Runtime property page.



Process Settings

Processor Affinity

○ Default Processor Affinity

● Specify Processor Affinity:

☑ CPU 1 ☑ CPU 3

□ CPU 2 □ CPU 4

Process Priority

○ Idle ○ Below Normal ● Normal ○ Above Normal

Processor Affinity

This allows you to choose which processors a process is allowed to be scheduled on. This option is useful if your computer has more than one physical processor, and you wish to control which processors can execute which jobs.

"Default Processor Affinity" means that the child process will be run with the same processor affinity as Automise. Unless you are running Automise with a special processor affinity, this is the same as selecting all processors.

Note that some programs may set their own processor affinity when executing. Automise will not change the processor affinity of these programs.

Process Priority

This allows you to choose which Priority Class a process is run in. This allows you to set some processes to have higher priorities than other processes, and can be particularly useful when using Async Action Groups to run actions in parallel.

Note that some programs (for instance, programs designed to run in the background) may lower their own priority class when they are executing. Automise will not change the priority of these programs.

Warning: Setting "Above Normal" process priority may cause Automise to become unresponsive when running, depending on the specific program's resource needs. Use this option with care.

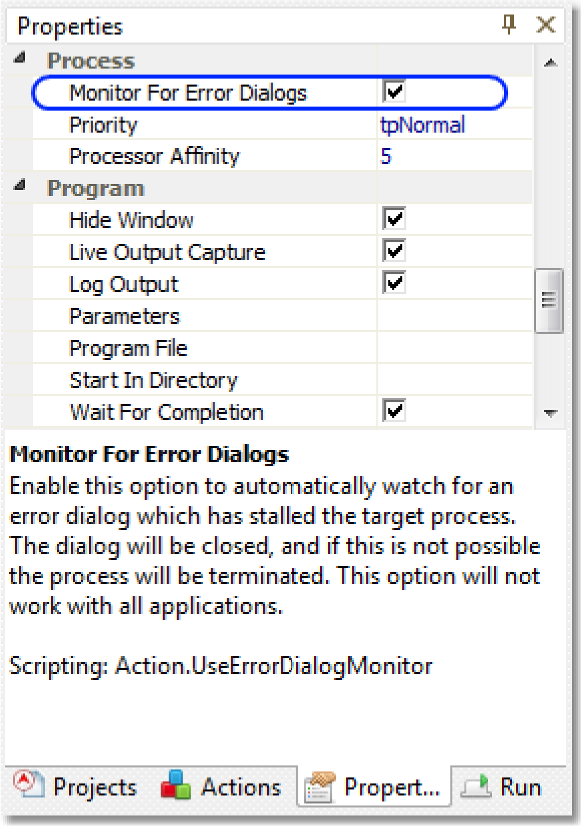
6.2.6 Error Dialog Monitor

Note: The Error Dialog Monitor is a feature for advanced users.

Sometimes when Automise launches an external program, that program may display an error dialog box. This can pose problems when a project is running unattended.

The ideal solution to this problem is to ask the author of the external program to rewrite the application so it does not display error dialogs during automatic execution. However, this is not always possible. In such situations, the Error Dialog Monitor feature may be of use.

The Error Dialog Monitor does not have any configuration options. To enable it, select the action which launches the problematic application, and select the Properties tab. Look for the section "Process" and the checkbox marked "Monitor For Error Dialogs". This property only appears for actions which launch external processes.



Process	
Monitor For Error Dialogs	☑
Priority	tpNormal
Processor Affinity	5
Program	
Hide Window	☑
Live Output Capture	☑
Log Output	☑
Parameters	
Program File	
Start In Directory	
Wait For Completion	☑

Monitor For Error Dialogs
Enable this option to automatically watch for an error dialog which has stalled the target process. The dialog will be closed, and if this is not possible the process will be terminated. This option will not work with all applications.

Scripting: Action.UseErrorDialogMonitor

When this option is available and the process is running, the monitor will watch for the following conditions:

- Process has displayed a visible, dialog-style, window for at least 5 seconds
- Process has been idle, not using any CPU time or performing any I/O, for at least 5 seconds(*)

If both of these conditions are found to be true, Automise will attempt to close the dialog. If the dialog cannot be closed, the process will be terminated.

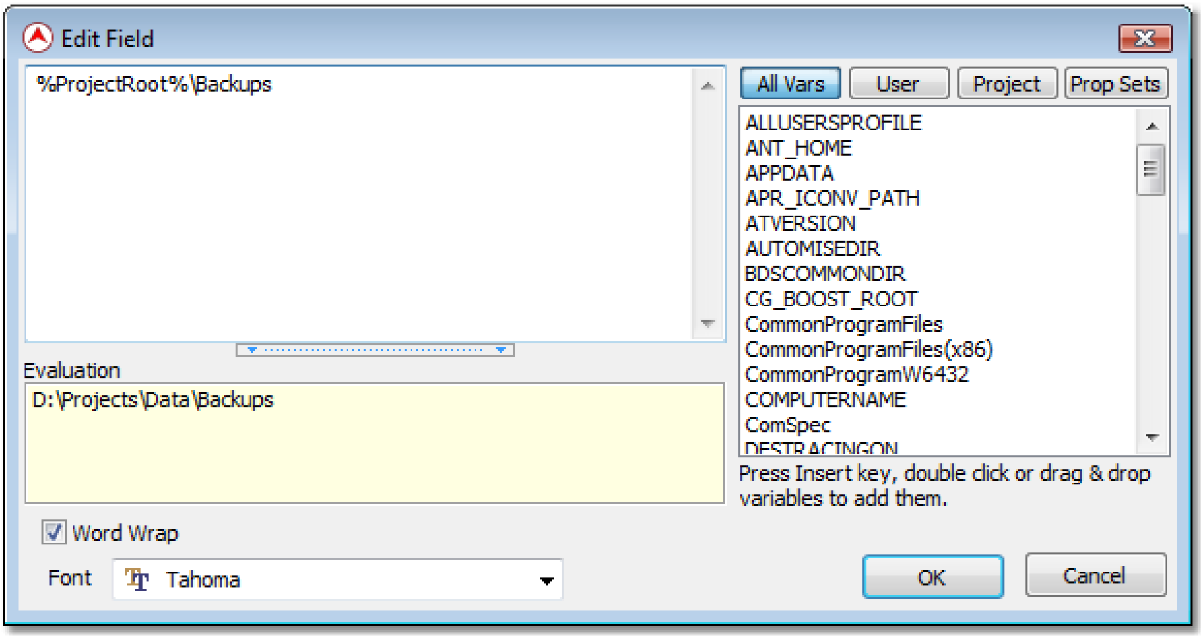
Automise will also attempt to record any text which was visible in the dialog before it was closed. This approach works for some dialogs and not for others, depending on the target application.

If the dialog was successfully closed, a message will be shown in the log but the action will not automatically fail. If the process needs to be terminated, the action will always fail.

(*) This feature is not foolproof and will only work with some applications. Many applications perform background processing which will fool Automise into thinking the process is still active, even though it may have stopped with an error dialog. Also, dialog detection cannot detect hung or deadlocked applications. In all of these cases, it is recommended you use the Timeout feature to time out execution.

6.3 Edit Field

You can use the Edit Field window to edit any text field in any action. Press <F2> to invoke it. It is particularly useful for multiline text, text with variables or other long values.



Edit Field

%ProjectRoot%\Backups

Evaluation

D:\Projects\Data\Backups

ALLUSERSPROFILE
ANT_HOME
APPDATA
APR_ICONV_PATH
ATVERSION
AUTOMISEDIR
BDSCOMMONDIR
CG_BOOST_ROOT
CommonProgramFiles
CommonProgramFiles(x86)
CommonProgramW6432
COMPUTERNAME
ComSpec
DESTRACINGON

Press Insert key, double click or drag & drop variables to add them.

☑ Word Wrap

Font: Tahoma

In the main editing area (top left) you can freely edit the value of the field. Press <Enter> to enter carriage return/linefeed characters.

The **Evaluation** area below shows the current evaluation of the text, by substituting in the current values of all variables.

The **Word Wrap** checkbox changes the display so that long lines of text are not automatically wrapped. It does not modify the value of the field, but may be useful if you are working with preformatted text.

The **Font** selection box lets you choose a different font to display the text with. It has no effect on the value of the field. It may be useful when dealing with text with foreign characters. If you are working with preformatted text, you may wish to choose a fixed width font such as Courier New.

Adding a variable

To add a variable reference to the current value, type % and the first few letters of its name. Once the correct variable is highlighted in the list on the right, press <Ins> to complete the reference. You can also drag and drop variables or double click them.

6.4 Archiving

6.4.1 7Zip

The 7Zip actions enable you to automate archive operations.

The archive operations available are:

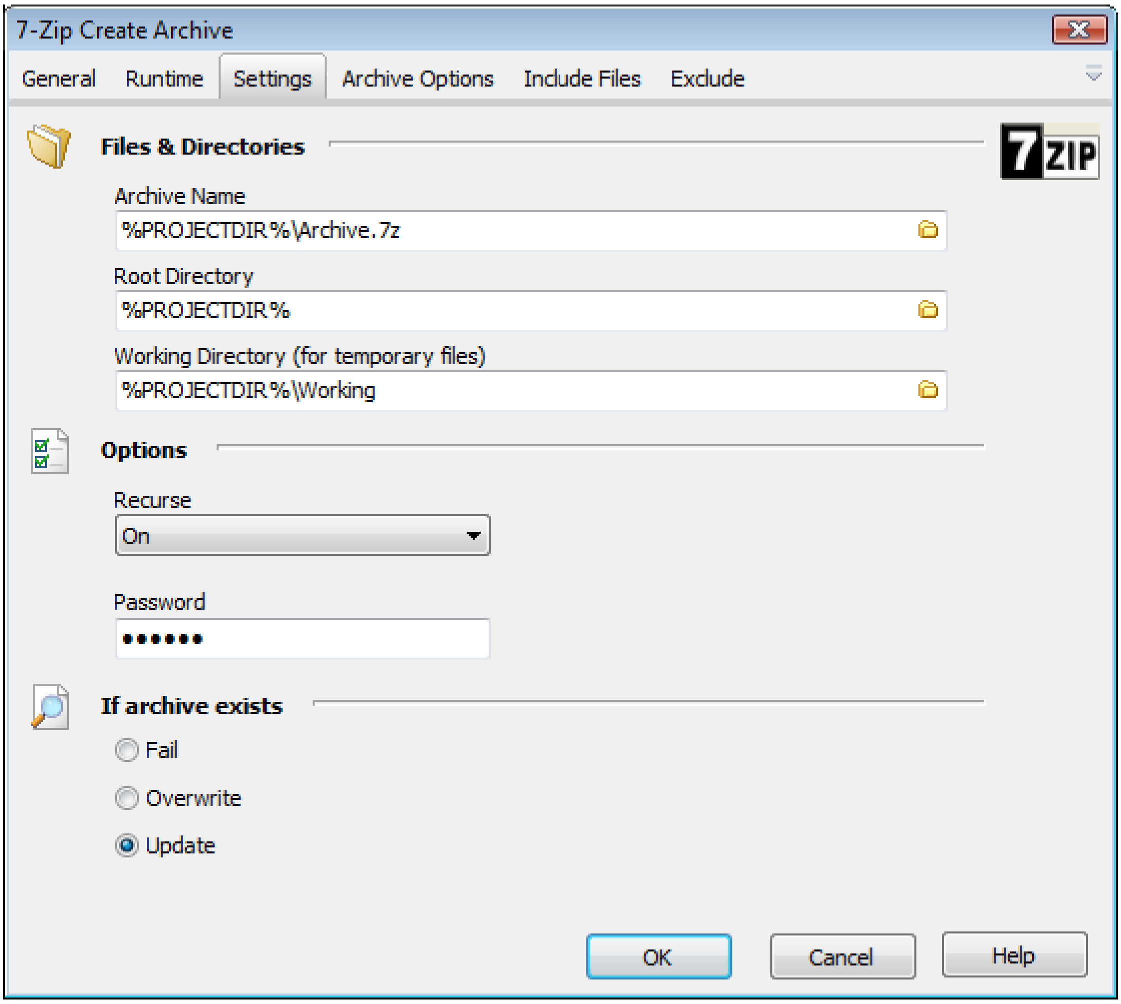
Create Archive (supports Zip, 7z, GZip, BZip2, TAR)
Test Archive (supports Zip, 7z, GZip, BZip2, TAR)
List files in Archive (supports Zip, 7z, GZip, BZip2, TAR, RAR, ARJ, CAB, CPIO, RPM, DEB, SPLIT)
Extract Archive (supports Zip, 7z, GZip, BZip2, TAR, RAR, ARJ, CAB, CPIO, RPM, DEB, SPLIT)
Update Archive (supports Zip, 7z, GZip, BZip2, TAR)
Delete file from Archive (supports Zip, 7z, GZip, BZip2, TAR)

These actions require the 7Zip tool, available at http://www.7-zip.org/

6.4.1.1 Create Archive

The Create Archive actions allows you to create archives using any of the following formats:
Zip, 7z, GZip, BZip2, TAR

More Info on the 7Zip based actions



7-Zip Create Archive

Files & Directories

Archive Name: %PROJECTDIR%\Archive.7z

Root Directory: %PROJECTDIR%

Working Directory (for temporary files): %PROJECTDIR%\Working

Options

Recurse: On

Password: ●●●●●●

If archive exists

○ Fail

○ Overwrite

● Update

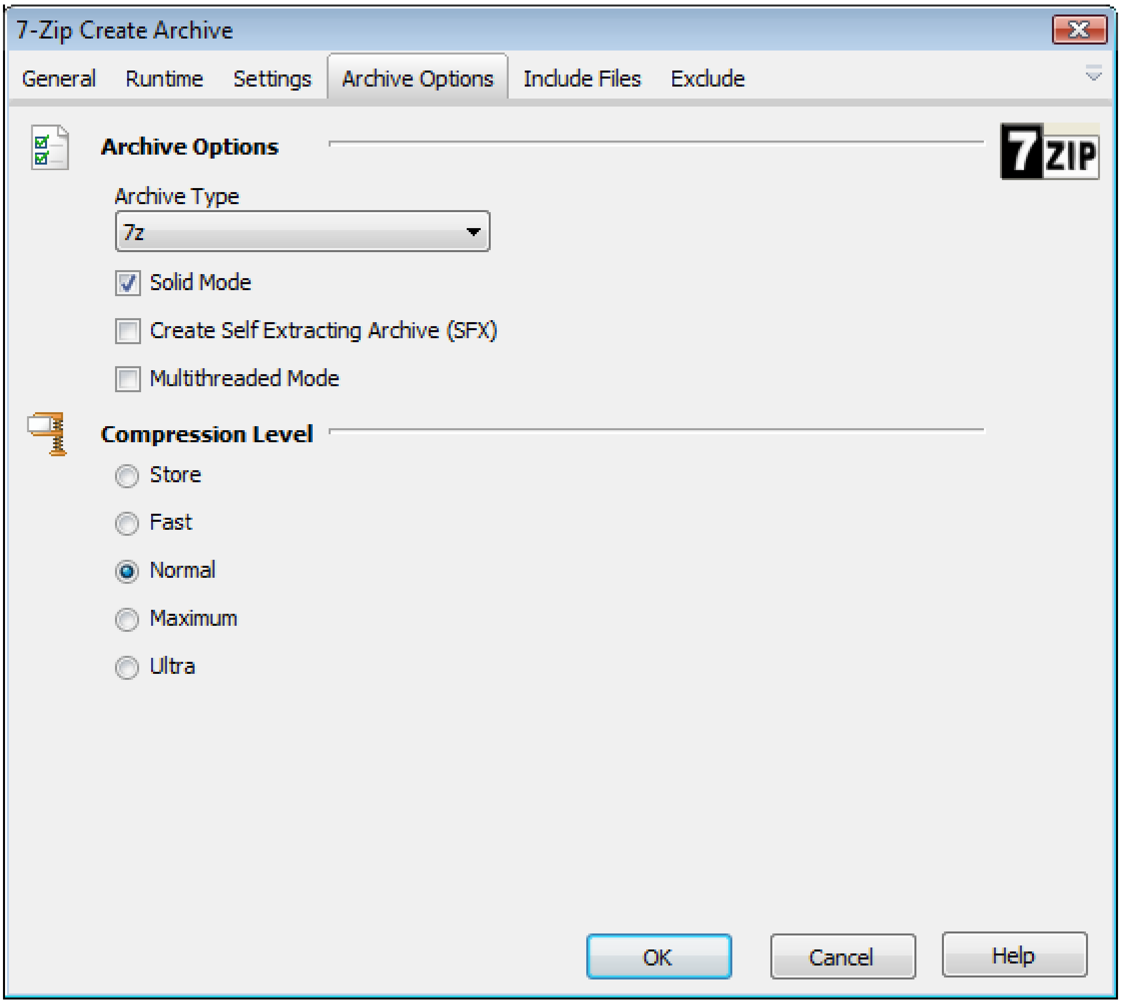
Archive Name - specify the name of the new archive. The file extension should match the type of archive you are creating (eg. ZIP)

Root Directory - you can optionally set this to a directory so that the included files can be relative to this directory

Working Directory - any temporary files will be placed in this directory

Recurse - specify how it should deal with sub-folders

Password - specify a password to protect the archive. You'll need to supply this password to decrypt the archive



Archive Options

Archive Type: 7z

☑ Solid Mode

□ Create Self Extracting Archive (SFX)

□ Multithreaded Mode

Compression Level

○ Store

○ Fast

● Normal

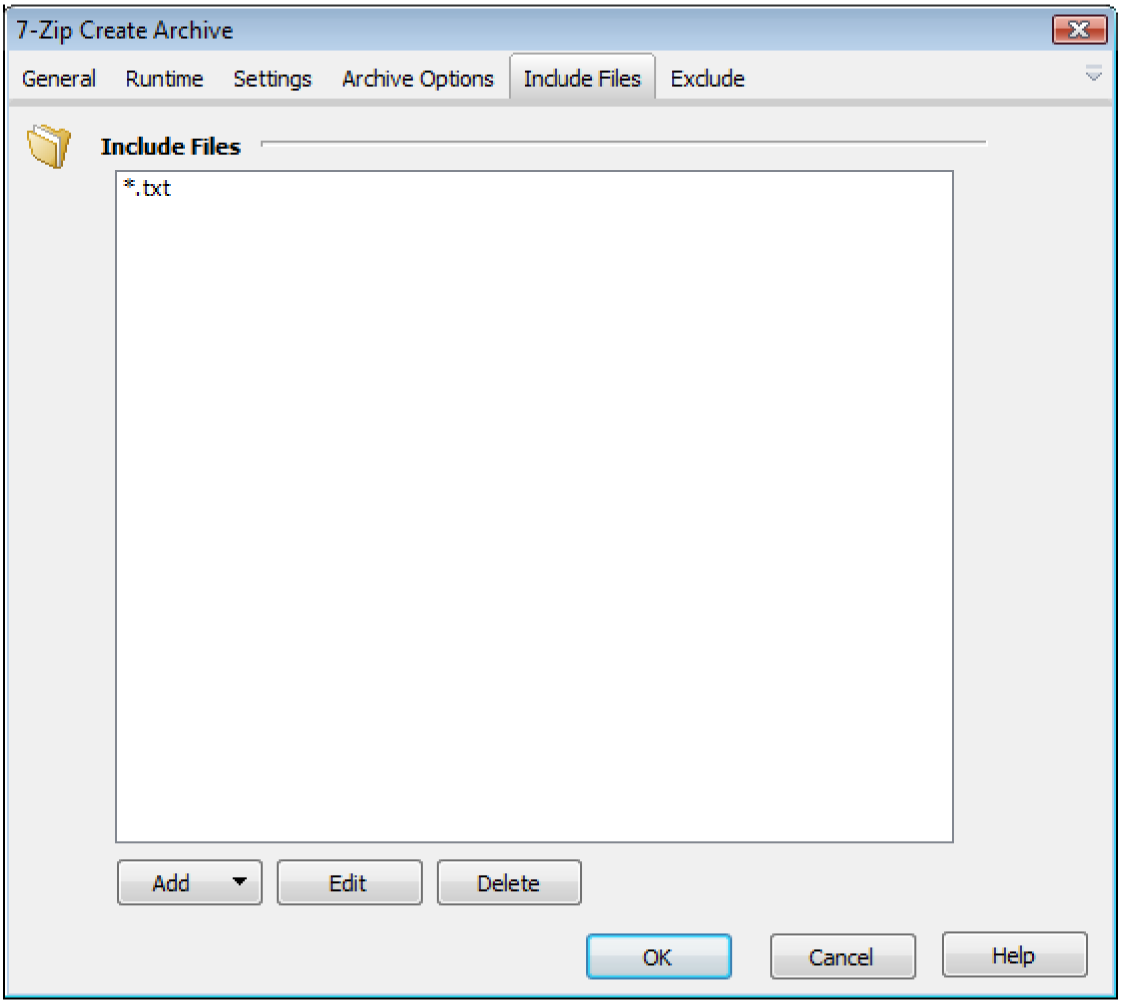
○ Maximum

○ Ultra

Archive Type - specify the archive type you want to create. It should match the file extension of the archive name.

Solid Mode, SFX, and Multithreaded are modes available if you choose the 7zip format

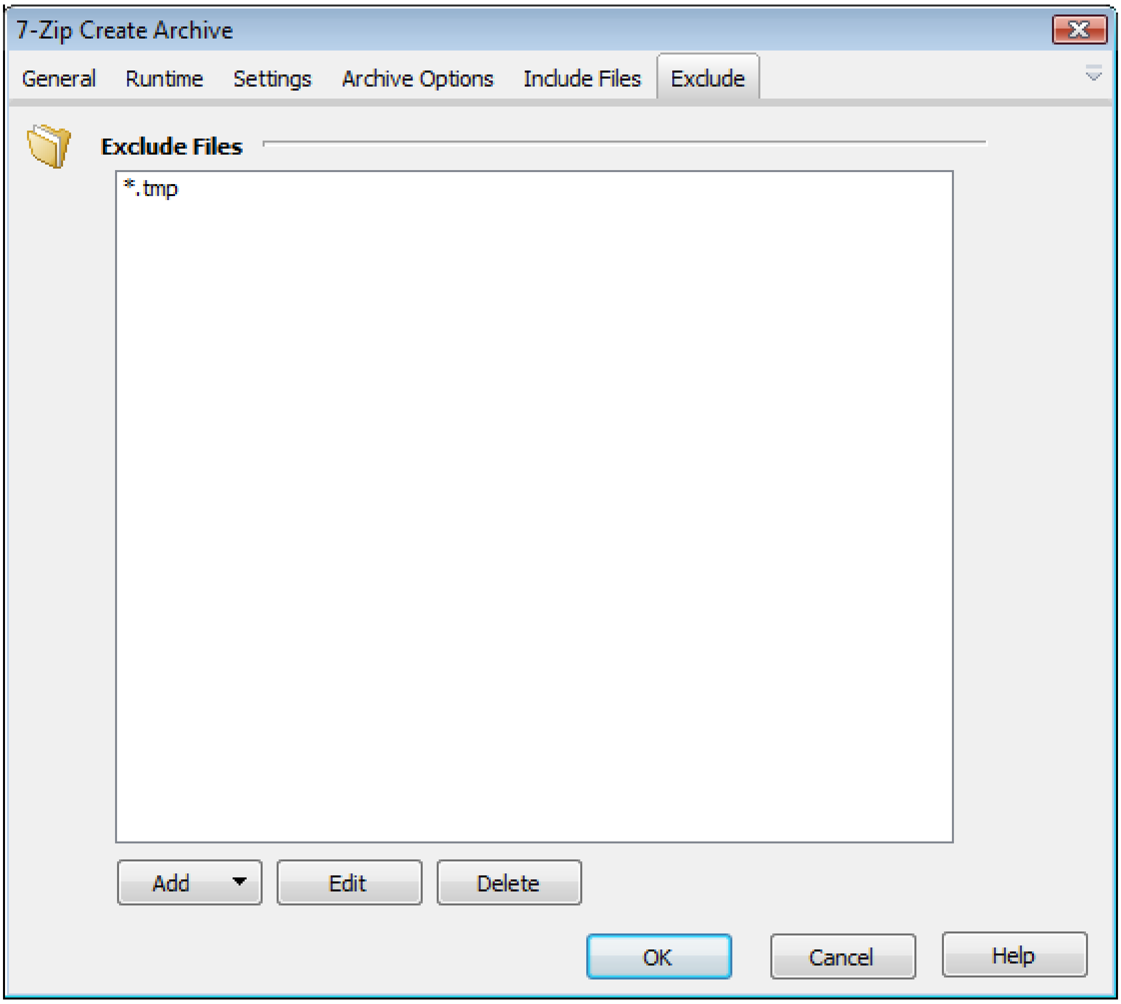
Compression Level - specify which compression level you require. Higher compression levels require more CPU and memory. Some compression formats only support a subset of the available compression levels.



*.txt

Specify the files to include in the new archive.
If you don't specify a fully qualified filename, then the working directory must be set on the Settings tab.

The add button allows you to add a file, folder, or Other. Other is typically used to enter a wildcard filespec, such as *.txt

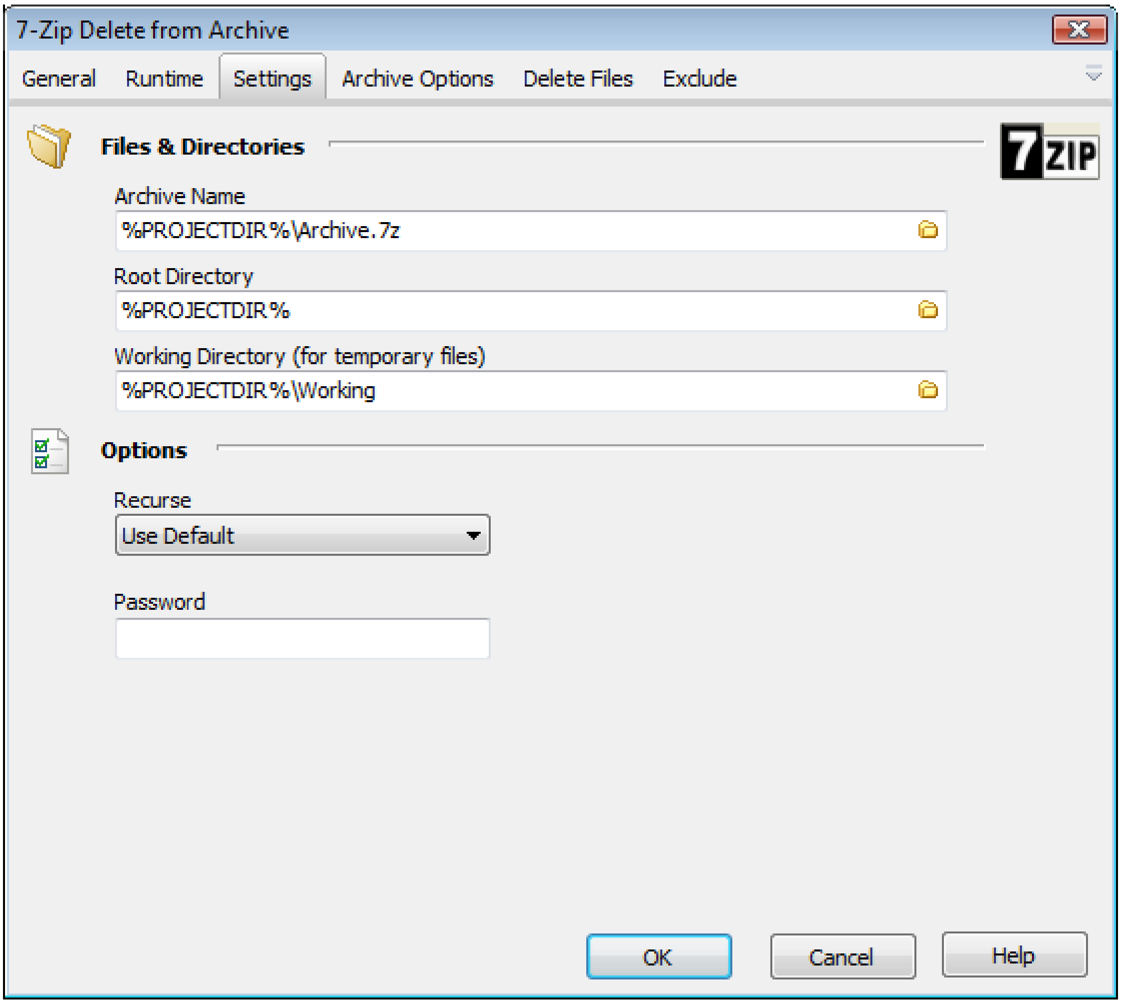


*.tmp

Specify any files or filespecs to exclude from the archive.

6.4.1.2 Delete from Archive

The Delete from Archive actions allows you to delete files within an archive in any of the following formats:
Zip, 7z, GZip, BZip2, TAR



7-Zip Delete from Archive

Files & Directories

Archive Name: %PROJECTDIR%\Archive.7z

Root Directory: %PROJECTDIR%

Working Directory (for temporary files): %PROJECTDIR%\Working

Options

Recurse: Use Default

Password:

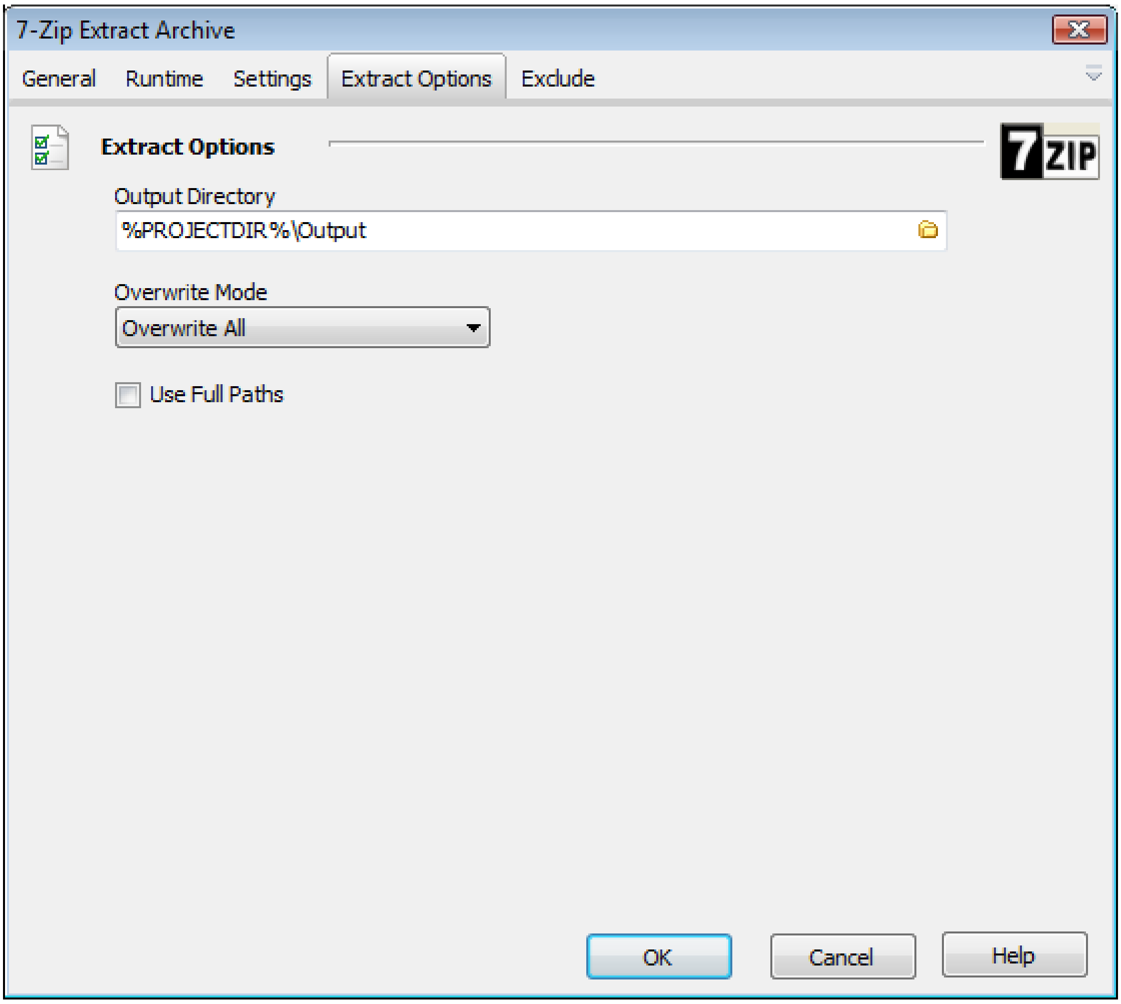
More Info on the 7Zip based actions

For more detailed descriptions of the options, see the Create Archive action.

6.4.1.3 Extract Archive

The Extract Archive actions allows you to extract the files from an archive in any of the following formats:
Zip, 7z, GZip, BZip2, TAR, RAR, ARJ, CAB, CPIO, RPM, DEB, SPLIT

More Info on the 7Zip based actions



7-Zip Extract Archive

Output Directory: %PROJECTDIR%\Output

Overwrite Mode: Overwrite All

□ Use Full Paths

Output Directory - specify the directory where the extracted files will be saved to

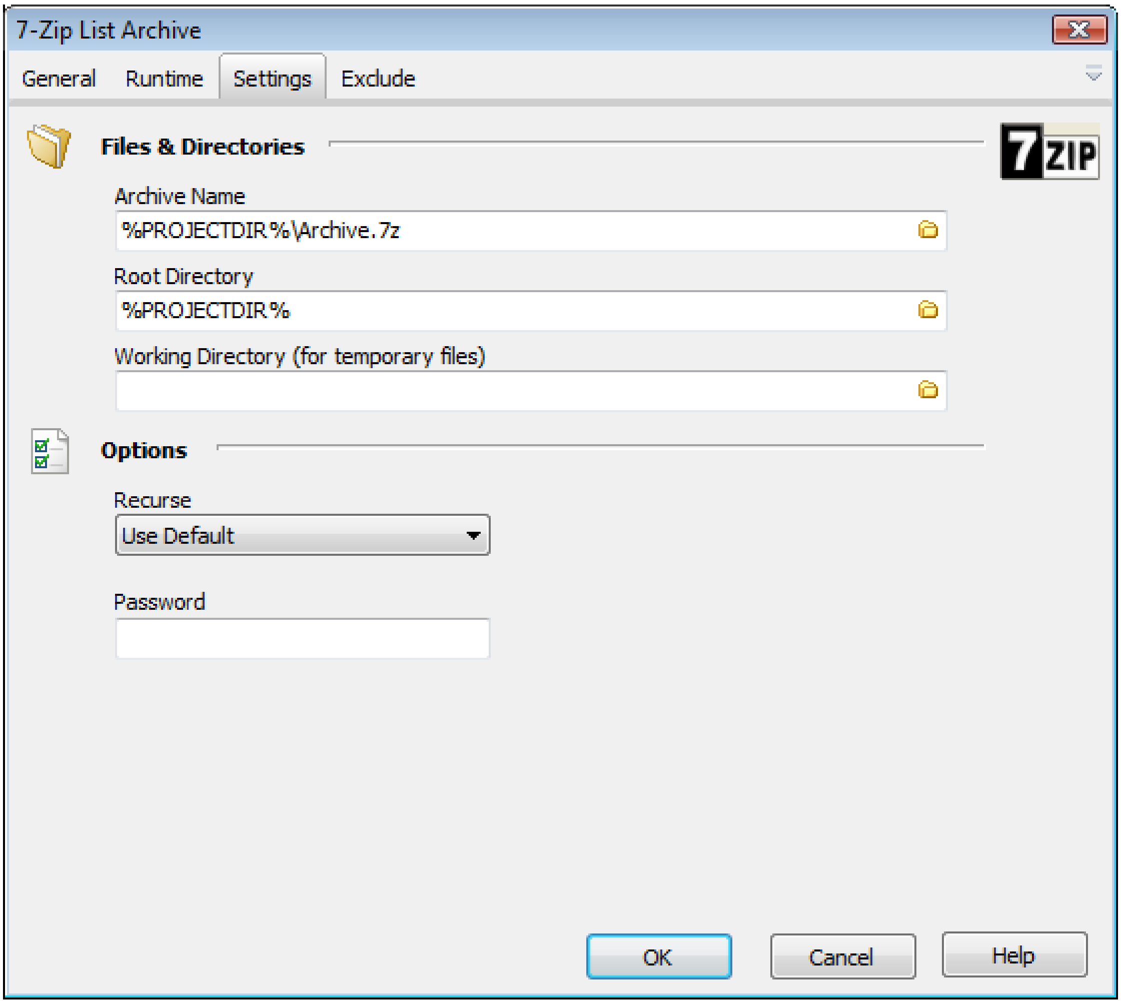
Overwrite Mode - specify how existing files with the same name will be dealt with

Use Full Paths - Files will be extracted with their paths as they are in the archive file. eg. if a file in the archive is in a directory called "dir" then the files in that directory will be extracted to <outputdirectory>\dir\<files>

For more detailed descriptions of the other options, see the Create Archive action.

6.4.1.4 List Archive

The List Archive actions allows you to list the files stored within an archive in any of the following formats:
Zip, 7z, GZip, BZip2, TAR



Archive Name: %PROJECTDIR%\Archive.7z

Root Directory: %PROJECTDIR%

Working Directory (for temporary files):

Recurse: Use Default

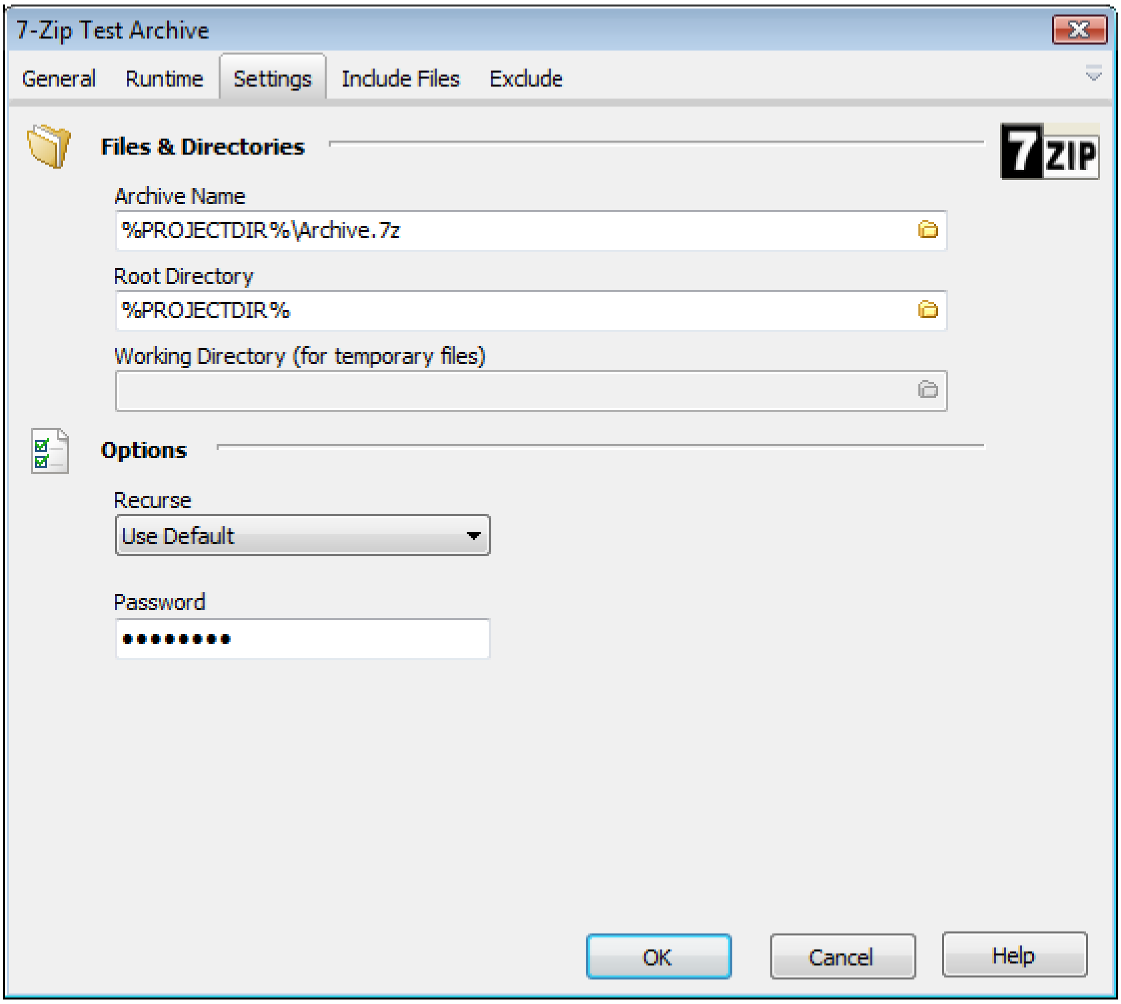
Password:

More Info on the 7Zip based actions

For more detailed descriptions of the options, see the Create Archive action.

6.4.1.5 Test Archive

The Test Archive actions allows you to test the integrity of an archive in any of the following formats:
Zip, 7z, GZip, BZip2, TAR



Archive Name: %PROJECTDIR%\Archive.7z

Root Directory: %PROJECTDIR%

Working Directory (for temporary files):

Recurse: Use Default

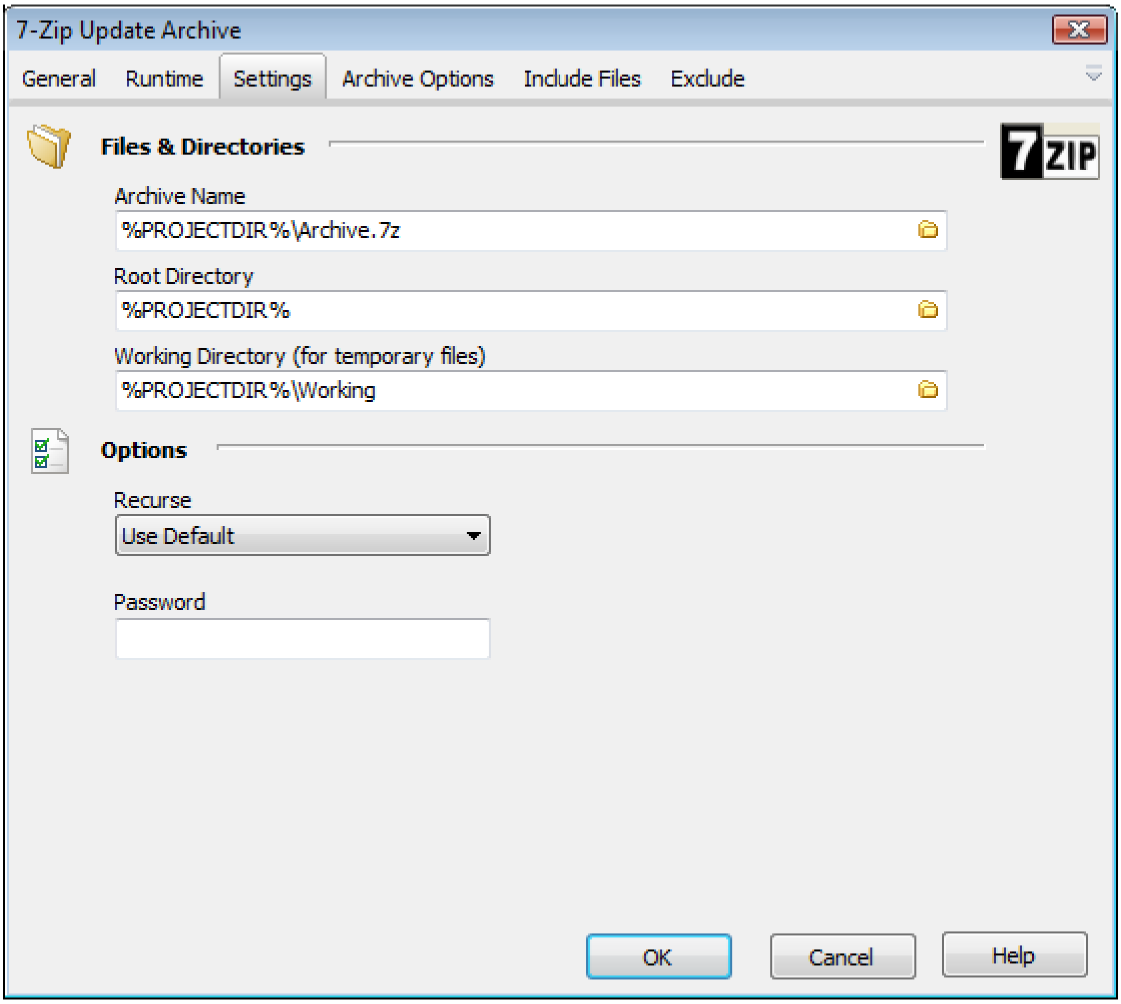
Password: ●●●●●●●●●

More Info on the 7Zip based actions

For more detailed descriptions of the options, see the Create Archive action.

6.4.1.6 Update Archive

The Update Archive actions allows you to update (or freshen) the files within an existing archive in any of the following formats:
Zip, 7z, GZip, BZip2, TAR



Archive Name: %PROJECTDIR%\Archive.7z

Root Directory: %PROJECTDIR%

Working Directory (for temporary files): %PROJECTDIR%\Working

Recurse: Use Default

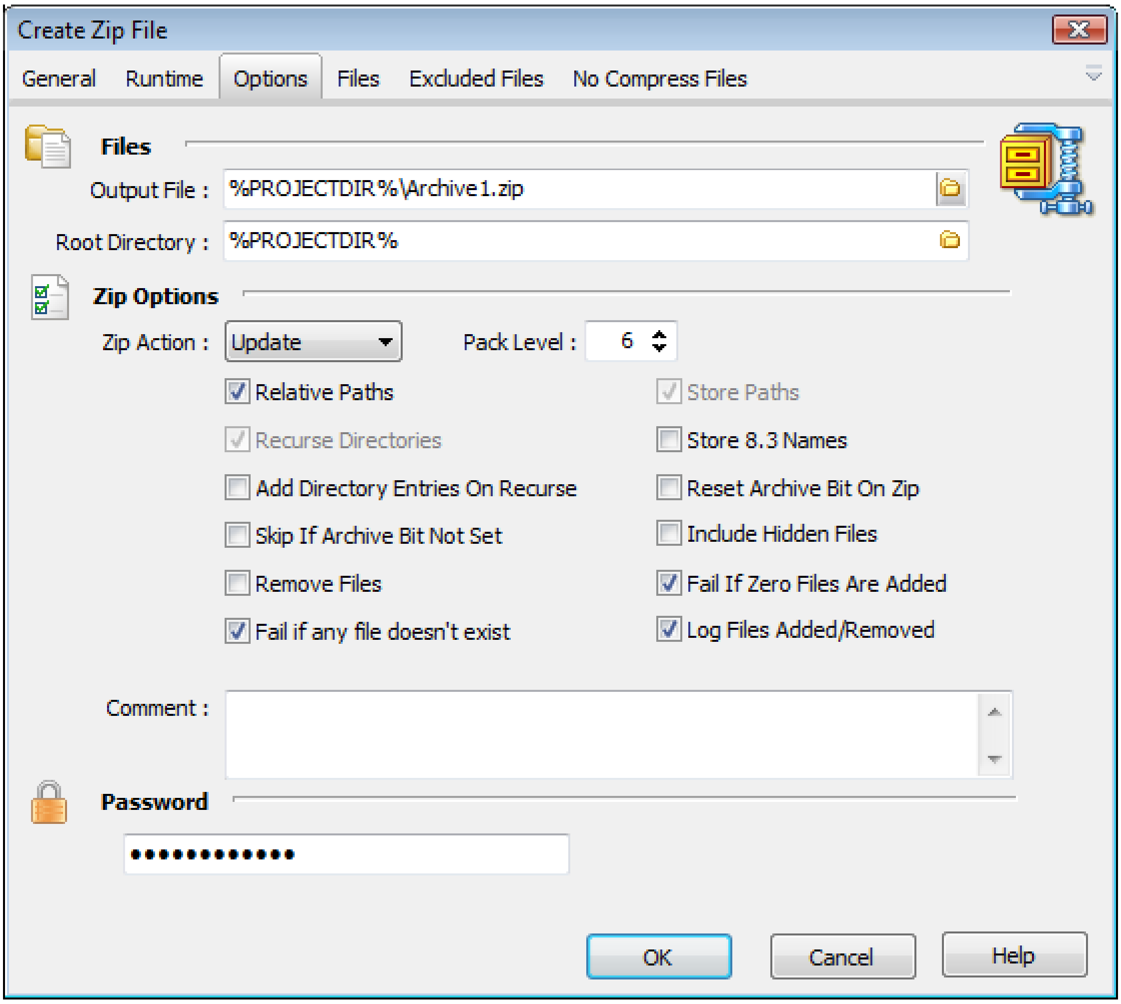
Password:

More Info on the 7Zip based actions

For more detailed descriptions of the options, see the Create Archive action.

6.4.2 Create Zip File

This action provides the ability to create Zip files.



Files

Output File : %PROJECTDIR%\Archive1.zip

Root Directory : %PROJECTDIR%

Zip Options

Zip Action : Update

Pack Level : 6

☑ Relative Paths

☑ Store Paths

☑ Recurse Directories

□ Store 8.3 Names

□ Add Directory Entries On Recurse

□ Reset Archive Bit On Zip

□ Skip If Archive Bit Not Set

□ Include Hidden Files

□ Remove Files

☑ Fail If Zero Files Are Added

☑ Fail if any file doesn't exist

☑ Log Files Added/Removed

Comment :

Password

●●●●●●●●●●●●

Output File : Output FileName

Zip Action : The ZipAction property determines whether files will be replaced in a zip or not. If ZipAction is set to zaUpdate then a zip entry will only be replaced if the disk file is newer than the zip entry. If ZipAction is set to zaReplace, then the zip entry will be replaced by the disk file regardless of the file dates. A ZipAction of zaFreshen is the same as zaUpdate, except that filenames that do not match any entries already in the zip file will be ignored and not added to the zip.

Pack Level : The PackLevel property determines how hard the compression algorithm will try to compress files. This property can be given a value from 0 through 9. A value of 0 is no compression at all (STORED) which is useful for adding things like other zip files (which will compress very little if any) to an archive. A value of 1 will compress the fastest, but the compression ratio will be the lowest. A value of 9 will compress the slowest, but the compression ratio will be the highest. You may specify particular files that you do not want to try to compress by adding their filespecs or wildcards to the NoCompressList.

Relative Paths : The Relative Paths property should be set to True if you wish to save path information but only wish to save path information relative to a specified directory. The only path information that is saved is for subdirectories below the specified directory. This is similar to the Relative Path option of PKZip for Windows. Whenever you set Relative Paths property to True, the Recurse Property and the Store Paths Property are automatically set to True also. Likewise, if the Store Paths Property is set to False, then the Relative Paths Property is automatically set to False also.

Store Paths : The StorePaths property, if set to True, will cause path information to be stored with the zip entry. If the StoreVolumes Property is set to True then the entire path will be stored. If the StoreVolumes Property is False, then only the path information will be stored. If StorePaths is False, then only the filename itself is stored. Note that if the RelativePaths Property is set to True, then this StorePaths property will automatically be set to True also. Likewise, if the StorePaths property is ever set to False, then both the RelativePaths and the StoreVolumes Property will automatically be set to False.

Recurse Directories : The Recurse property determines whether subdirectories will be recursed to look for files to be compressed when zipping with a wildcard mask. Set to true if you wish subdirectories to be traversed. If this value is set to True, and a wildcard mask is specified in the FilesList without any path information, then the value of the RootDir Property will determine which directory zipping will start in. Results may be unexpected or even bad if you do not supply path information either in the FilesList or the RootDir Property.

Store 8.3 Names : When set to True, this will force any long file and pathnames to be stored in DOS 8.3 format. This is useful if you plan to unzip the files onto a WIN3.X system where long filenames are not valid.

Add Directory Entries On Recurse : If Add Directory Entries On Recurse is True, then when you do a recursive search through subdirectories (Recurse = True) a separate entry will be made in the archive for each directory. This will allow even empty directories to be restored. If Add Dir Entries On Recurse is False, path information will still be stored with each file that is compressed, but a separate entry will not be inserted for the directories.

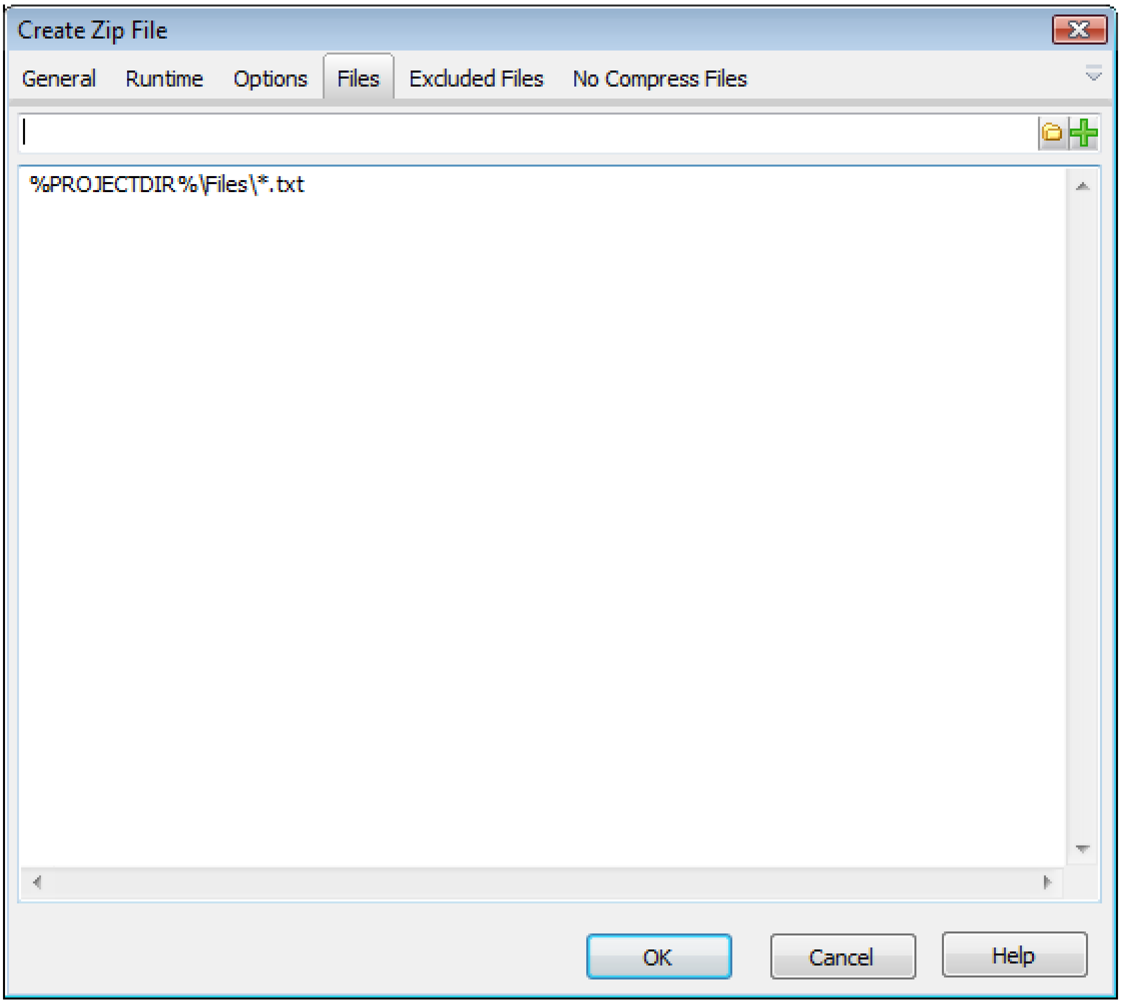
Root directory : The RootDir property determines where zipping will start for any wildcard entries or filenames in the FilesList Property that do not already include path information. Essentially, the value of RootDir will be prepended to anything in the FilesList that does not have any path information when zipping. Also, when storing relative path information using the RelativePaths Property, you must use this RootDir property to specify the directory from where path information will begin being saved.

Skip if Archive bit Not Set : Setting this to True will cause files that do not have their Archive Bit set (turned on) to be skipped during zip operations. Therefore, while this is set to True, only files with their Archive attribute turned on will be zipped.

Reset Archive bit On zip : Setting this to True will cause the Archive Bit for each file to be reset (turned off) after being zipped.

Remove Files : The Remove Files property, if set to True, will cause the original disk files that were added to the zip file to be deleted from the disk, in effect, moving the files into the zip file. USE THIS OPTION WITH EXTREME CAUTION. If an exception occurs during processing, files will not be Deleted.

Fail If Zero Files are Added : Fail If Zero Files are added to zip file. This will cause the run to stop unless the ignore failure property is set.



%PROJECTDIR%\Files\*.txt

The Files Section allows you to specify the files that will be added to the zip file. You can use Wildcards (*, ?) and Automise variables when specifying the files.

The Excluded Files section allows you specify file that should not be included in the resulting zip file. You can use Wildcards (*, ?) and Automise variables when specifying the files.

The No Compress Files allows you to specify files that should be added to the zip file but should not be compressed. This is useful for adding other zip files. You can use Wildcards (*, ?) and Automise variables when specifying the files.

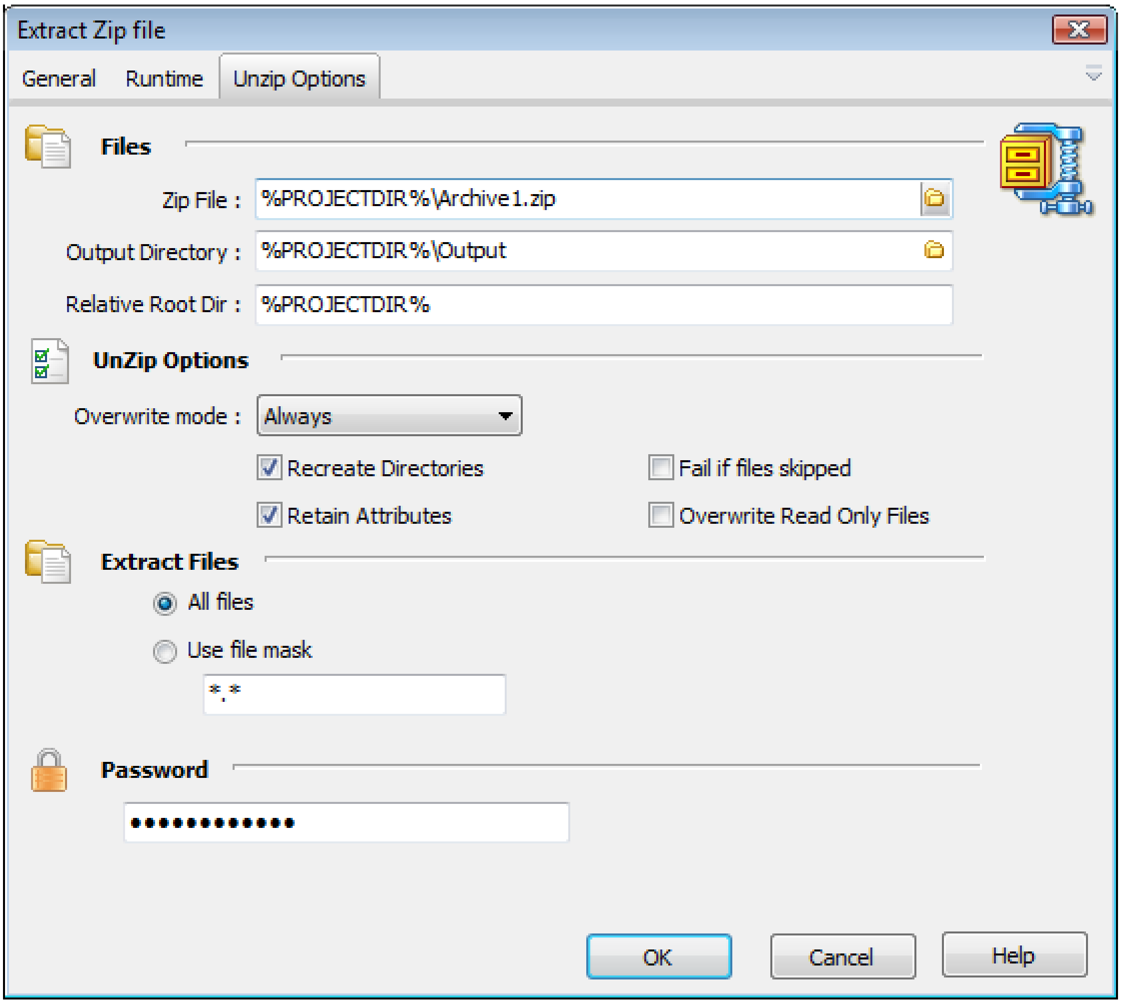
Scripting Info

The Action properties available are :

property OutputFileName : WideString
property PackLevel: Integer // 0 - 9
property Recurse: WordBool
property Dispose: WordBool
property StorePaths: WordBool
property RelativePaths: WordBool
property Store83Names: WordBool
property SkipIfArchiveBitNotSet : WordBool
property ResetArchiveBitOnZip: WordBool
property AddDirEntriesOnRecurse: WordBool
property FailIfZeroFiles : WordBool
property RootDir : WideString

6.4.3 Extract Zip File Action

This action allows you to extract files from a zip archive file. You can choose which files to extract using file masks or just extract all files.



Zip File : %PROJECTDIR%\Archive1.zip

Output Directory : %PROJECTDIR%\Output

Relative Root Dir : %PROJECTDIR%

Overwrite mode : Always

☑ Recreate Directories

☐ Fail if files skipped

☑ Retain Attributes

☐ Overwrite Read Only Files

● All files

○ Use file mask

.

●●●●●●●●●●●●

Zip File - the location of the zip file to extract

Output Directory - the directory where the extracted files will be copied to

Relative Root Dir - this allows you to extract files from an archive that was not created using the relative paths option and still treat it as though it was. In this case you should set Relative Root Dir to the value that you want stripped from the internal path information for each file that is unzipped. For instance, if you had an archive containing the following information:

Filename	*Path Info*
file1.txt	backups\subdir1\
file2.txt	backups\subdir1\subdir2\
file3.txt	backups\subdir1\subdir3\

and you wished to extract these files to an Output Directory of c:\restore\backups and retain the directory structure, normally you would end up with something like the following being extracted:
c:\restore\backups\backups\subdir1\file1.txt

c:\restore\backups\backups\subdir1\subdir2\file2.txt
c:\restore\backups\backups\subdir1\subdir3\file3.txt

when what you really wanted was:
c:\restore\backups\subdir1\file1.txt
c:\restore\backups\subdir1\subdir2\file2.txt
c:\restore\backups\subdir1\subdir3\file3.txt

You can produce the desired results simply by setting Relative Root Dir to "backups\subdir1\".

Overwrite Mode - Set this property to determine what should be done if a file is about to be extracted and a file of the same name already exists in the destination directory. The possible modes are: Always, Never, If Newer, If Older.

Always The file will be overwritten.
Never The file will not be extracted if it would overwrite a file.
ifNewer The file will only overwrite the existing one if the archived file is newer than the existing one.
ifOlder The file will only overwrite the existing one if the archived file is older than the existing one.

Recreate Directories - Set this property to True if you want to use directory information in the zipfile when extracting files. The directories will be created relative to the Destination Directory. If this property is False, all files will be extracted to the Destination Directory, which could possibly result in files of the same name overwriting each other if the Overwrite Mode property is set to Always.

Retain Attributes - This property should be set to true if you wish unzipped files to retain the attributes that they had when they were originally zipped.

Extract Files - Choose whether to extract all files or only files that match a file mask.

Password - If uncompressing an encrypted password, enter the password here.

6.4.4 SecureZIP

SecureZIP

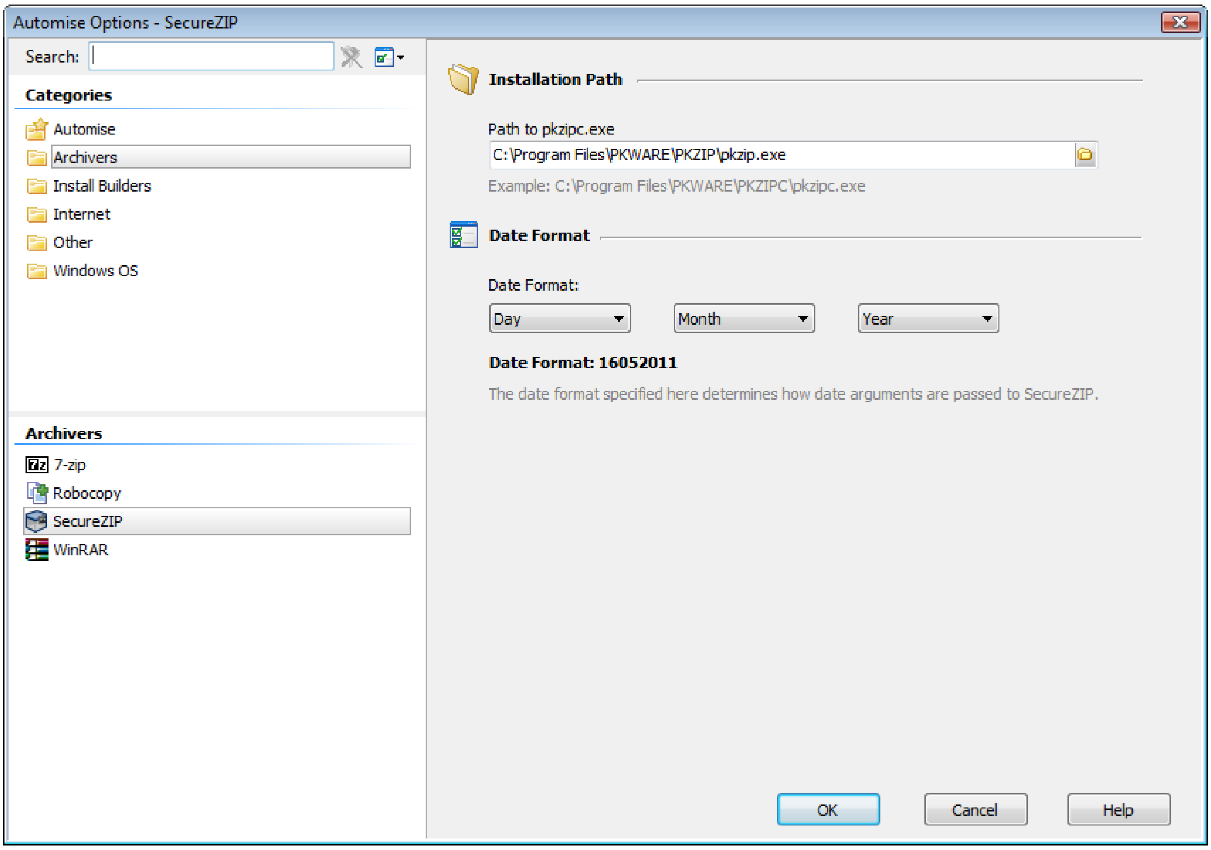
The SecureZIP actions allow you to automate archive operations using SecureZIP Command Line Interface (CLI).

SecureZIP CLI is available from www.pkware.com

The following SecureZIP actions are available:
- Add Files To Archive
- Extract Archive
- List Archive Contents
- Test Archive Integrity

SecureZIP Options

Before using the SecureZIP actions you need to go to the SecureZIP options page via Tools -> Options.



Automise Options - SecureZIP

Search:

Categories

- Automise
- Archivers
- Install Builders
- Internet
- Other
- Windows OS

Archivers

- 7-zip
- Robocopy
- SecureZIP
- WinRAR

Installation Path

Path to pkzipc.exe: C:\Program Files\PKWARE\PKZIP\pkzip.exe

Example: C:\Program Files\PKWARE\PKZIPC\pkzipc.exe

Date Format

Date Format: Day Month Year

Date Format: 16052011

The date format specified here determines how date arguments are passed to SecureZIP.

Installation Path
From the options page you need to specify the location of the executable (pkzipc.exe).

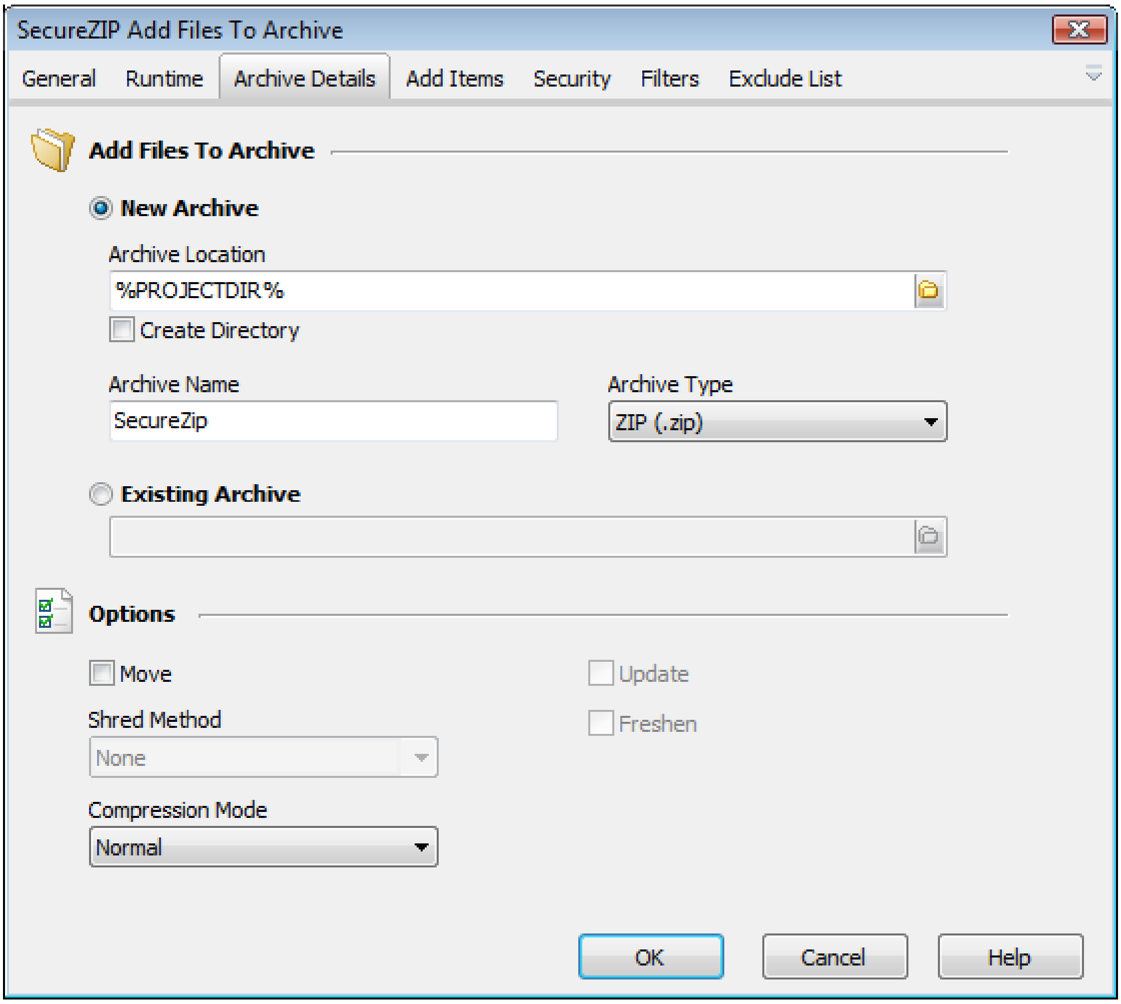
Date Format
You also need to set the date format that SecureZIP is expecting when using date filters, this will vary depending on your Regional settings. Generally the format expected will be along the lines of *ddmmyyyy* or *mmddyyy*. Use the controls to set the date format so that the current date is displayed below the controls in the format specified.

6.4.4.1 SecureZIP Add Files To Archive

SecureZIP Add Files To Archive

The SecureZIP Add Files To Archive action allows you to create a new archive or update an existing archive.

On the *Archive Details* page specify whether the archive to add files to is new archive or an existing archive.



Add Files To Archive

● New Archive

Archive Location: %PROJECTDIR%

□ Create Directory

Archive Name: SecureZip

Archive Type: ZIP (.zip)

○ Existing Archive

Options

□ Move

□ Update

Shred Method: None

□ Freshen

Compression Mode: Normal

If creating a **New Archive** you need to specify the following details:

- **Archive Location** - The destination directory for the new archive. If the directory you have provided does not exist, enable the Create Directory option, which will create the specified directory before creating the archive.
- **Archive Name** - The name of your archive.
- **Archive Type** - This determines the type of archive to create.

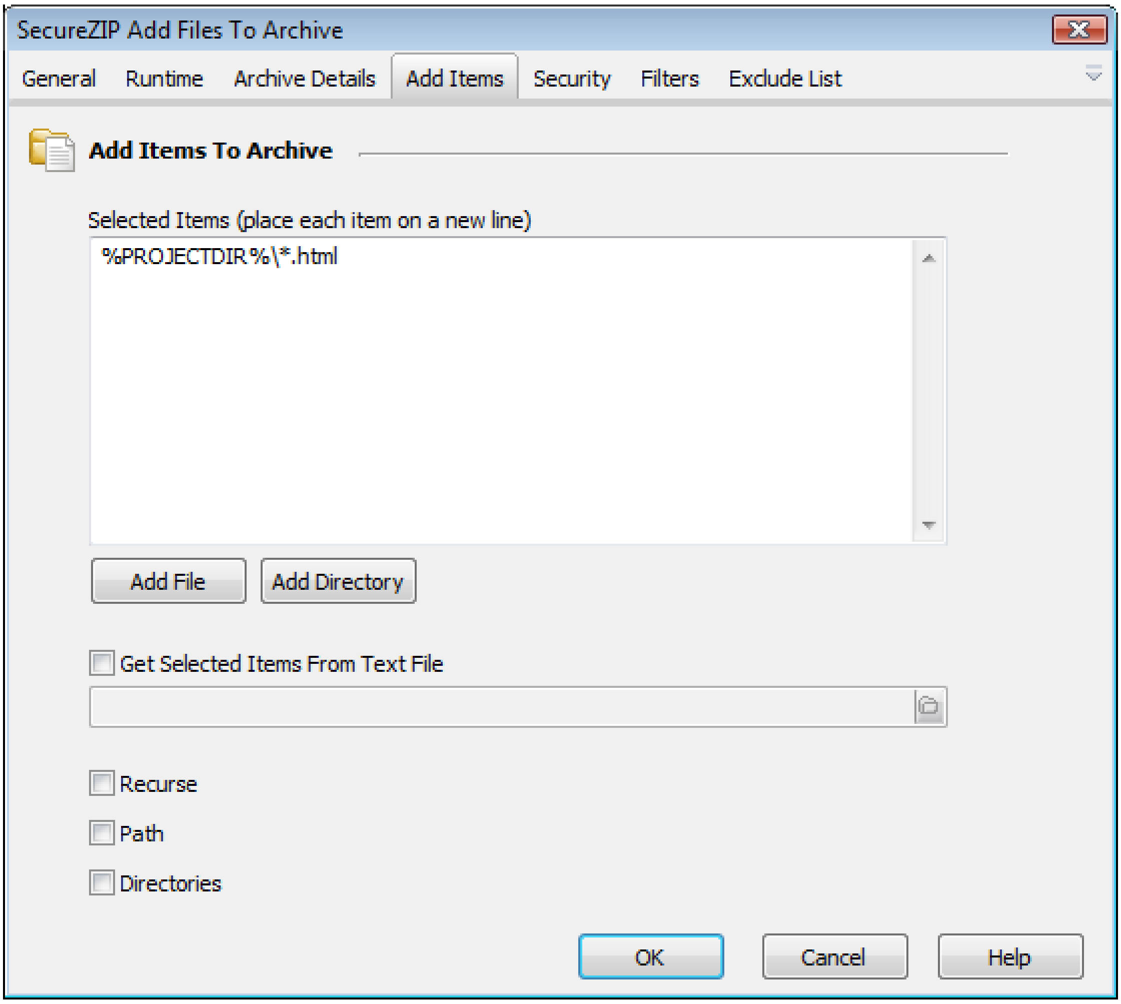
If adding files to an existing archive select the **Existing Archive** option and specify the location of the archive to update.

The following options are also available from the *Archive Details* page:

- **Move** - Files that are added to the archive are deleted from the original location once they have been added successfully.
- **Shred Method** (only available when using the *Move* option) - To prevent recovery of a file that has been deleted as a result of the *Move* option, the *Shred* option overwrites the files data to prevent recovery of the file. The following sub-options are available:
    - **None** - No shredding will occur, file will only be deleted on move operation.

    - **Random** - Overwrites the files once with random data.
    - **Dod5220** - Overwrites the files three times.
    - **NSA** - Overwrites the files seven times.
- **Compression Mode** - Allows you to determine the balance between speed and amount of compression. The following modes are available:
    - **Store** - Provides no compression, simply stores files within an archive.
    - **Speed** - Provides fast performance but lowest compression.
    - **Fast** - Provides second fastest performance with low compression.
    - **Normal** - Provides the best balance between compression and speed. This is the default option.
    - **Maximum** - Provides the maximum level of compression with the slowest performance.
- **Update** (only available when updating an Existing Archive) - This option allows you to update an existing archive by adding only new or modified files.
- **Freshen** (only available when updating an Existing Archive) - This option allows you to freshen an existing archive by re-compressing files modified files.

The *Add Items* page allows you to specify files and directories to add to the archive. If you are using the *Update* or *Freshen* options to modify an existing archive it is not necessary to specify items here.



SecureZIP Add Files To Archive

Add Items To Archive

Selected Items (place each item on a new line)

```
%PROJECTDIR%\*.html
```

□ Get Selected Items From Text File

□ Recurse

□ Path

□ Directories

As an alternative to listing all the items in the **Selected Items** field you can create a text file that contains the list of items to add and use the **Get Selected Items From Text File** option to specify the items to add.

The following options are available from the *Add Items* page:

- **Recurse** - Allows you to store all the files that exist in the subdirectories of a specified directory (by default subdirectories of a specified directory will be ignored).
- **Path** - Stores paths for files included in the archive, which allows you to recreate a directory structure on extraction. The path option only saves files that are referenced, subdirectories will be ignored.
- **Directories** - The directories option combines the *Recurse* and the *Path* options. By selecting this option, the path information for each file will be saved and each subdirectory (and the files within) will be added to the archive.

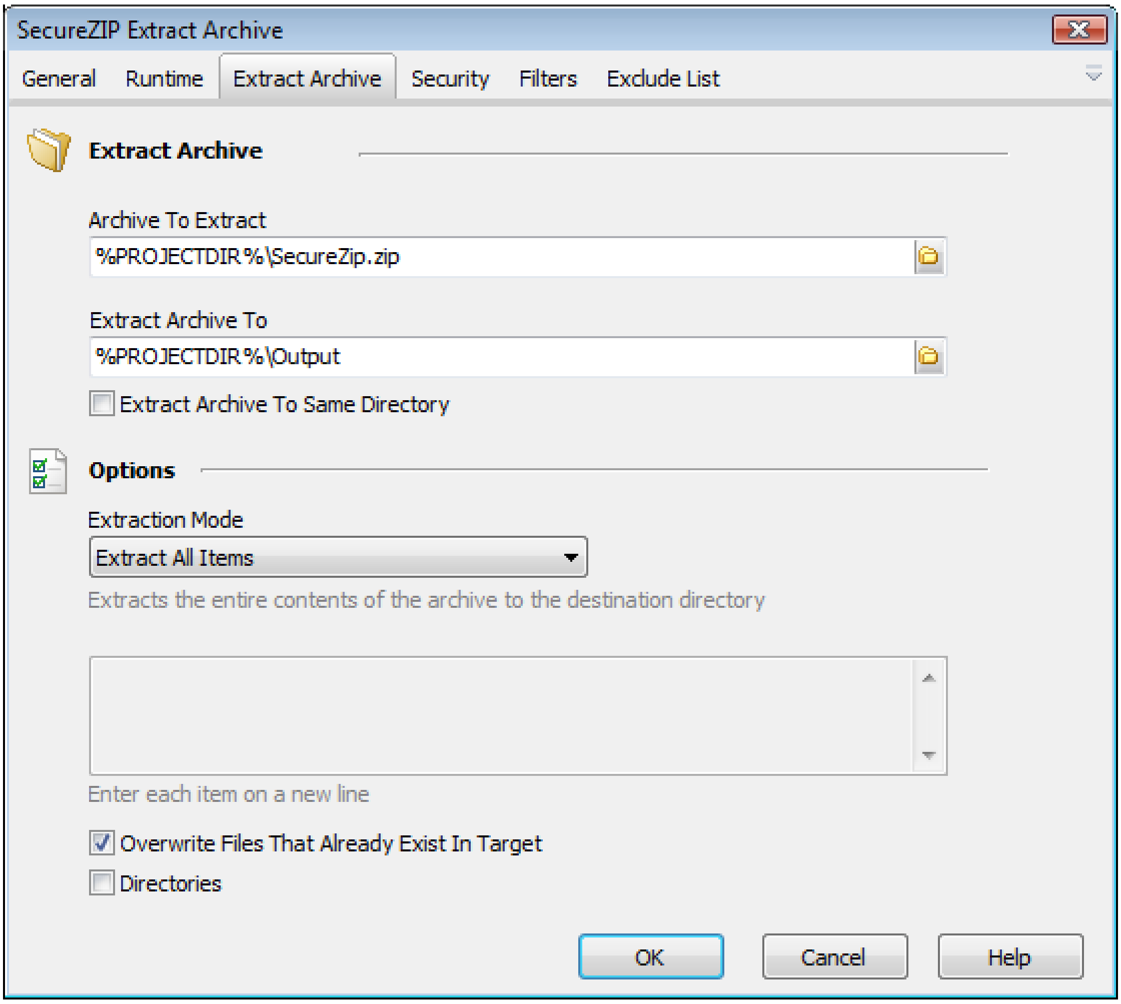
Additional Information

- To secure an archive see the help for the SecureZIP Security Page.
- To apply filters see the the help for the SecureZIP Filters Page.
- To use an exlcusion list see the help for the SecureZIP Exclude List Page.

6.4.4.2 SecureZIP Extract Archive

SecureZIP Extract Archive

The SecureZIP Extract Archive action allows you to extract the contents of an archive to specified location.



Extract Archive

Archive To Extract: %PROJECTDIR%\SecureZip.zip

Extract Archive To: %PROJECTDIR%\Output

□ Extract Archive To Same Directory

Options

Extraction Mode: Extract All Items

Extracts the entire contents of the archive to the destination directory

Enter each item on a new line

☑ Overwrite Files That Already Exist In Target

□ Directories

On the *Extract Archive* page, specify the archive to extract. Specify the directory to output the contents of the archive to, alternatively you can select the **Extract Archive To Same Directory** option which will output the contents of the archive into the directory where the archive itself resides.

The following options are also available from the *Extract Archive* page:

- **Extraction Mode**
  - **Extract All Items** - Extracts the entire contents of the archive to the destination specified.
  - **Extract Selected Items** - Extracts only specified items to the destination.
  - **Update** - Extracts only files that are not already in the directory or are newer versions of files that are already there.
  - **Freshen** - Extracts only files that are newer versions of files that already exist in the target directory.
- **Overwrite Files That Already Exist In Target** (only available when using *Extract All Items* or *Extract Specified Items* mode) - Specifies that any files to be extracted which already exist in the target directory will be overwritten. If this option is not enabled files that already exist in the target directory will not be extracted.
- **Directories** - Extracts any directory tree structure saved with files.

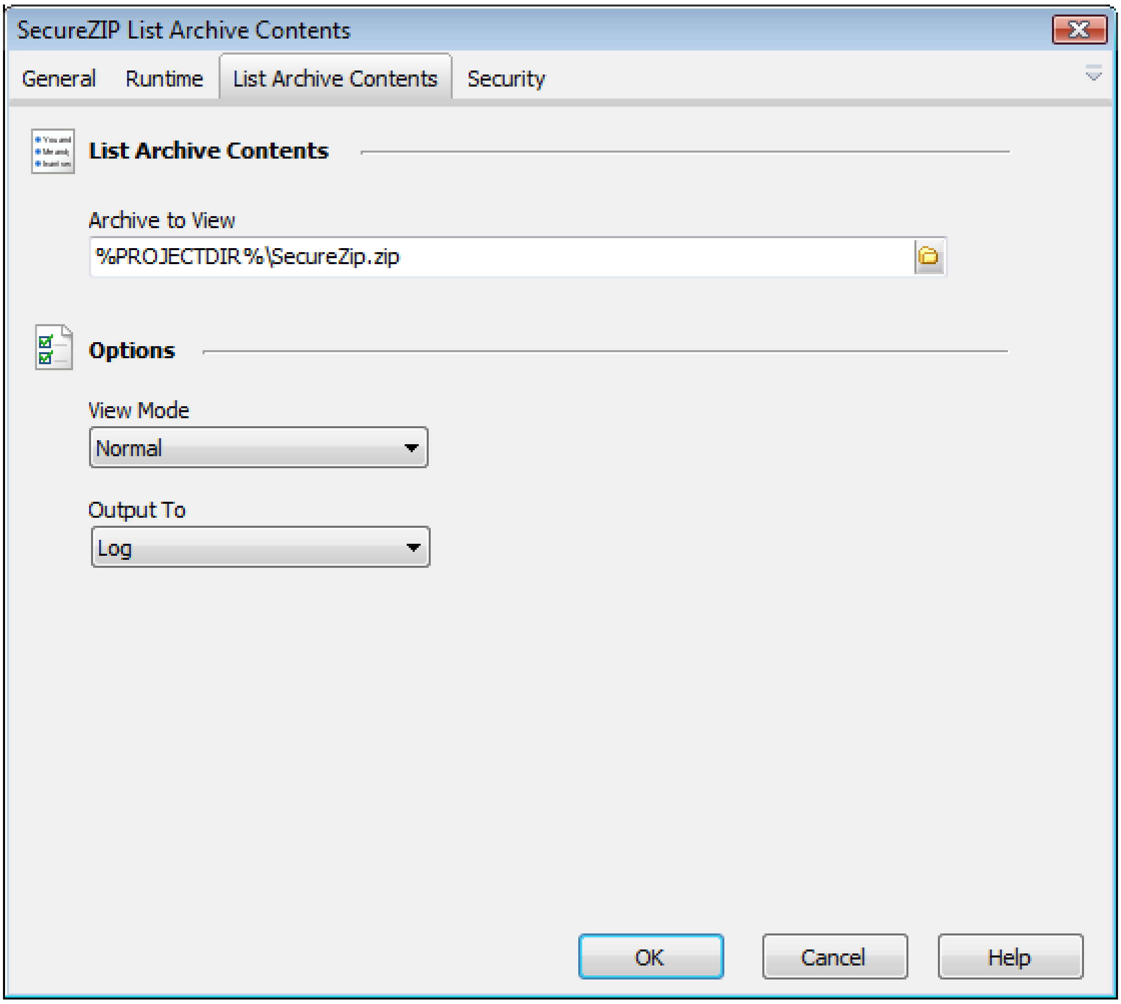
Additional Information

- To access a secure archive see the help for the SecureZIP Security Page.
- To apply filters see the the help for the SecureZIP Filters Page.
- To use an exlcusion list see the help for the SecureZIP Exclude List Page.

6.4.4.3 SecureZIP List Archive Contents

SecureZIP List Archive Contents

The SecureZIP List Archive Contents action allows you to display the contents of an archive.



List Archive Contents

Archive to View: %PROJECTDIR%\SecureZip.zip

Options

View Mode: Normal

Output To: Log

To view the contents of an archive, specify the archive in the field provided on the *List Archive Contents* page.

Set the following options:

- **View Mode**
  - **Brief** - Displays very basic information about the contents of the archive.
  - **Normal** - The default output.
  - **Details** - Displays detailed information about each file within the archive.
- **Output To**
  - **Log** - Write the archive contents to the build log only.
  - **File** - Write the archive contents to a specified file only.
  - **Log and File** - Write the archive contents to both the build log and a specified file.

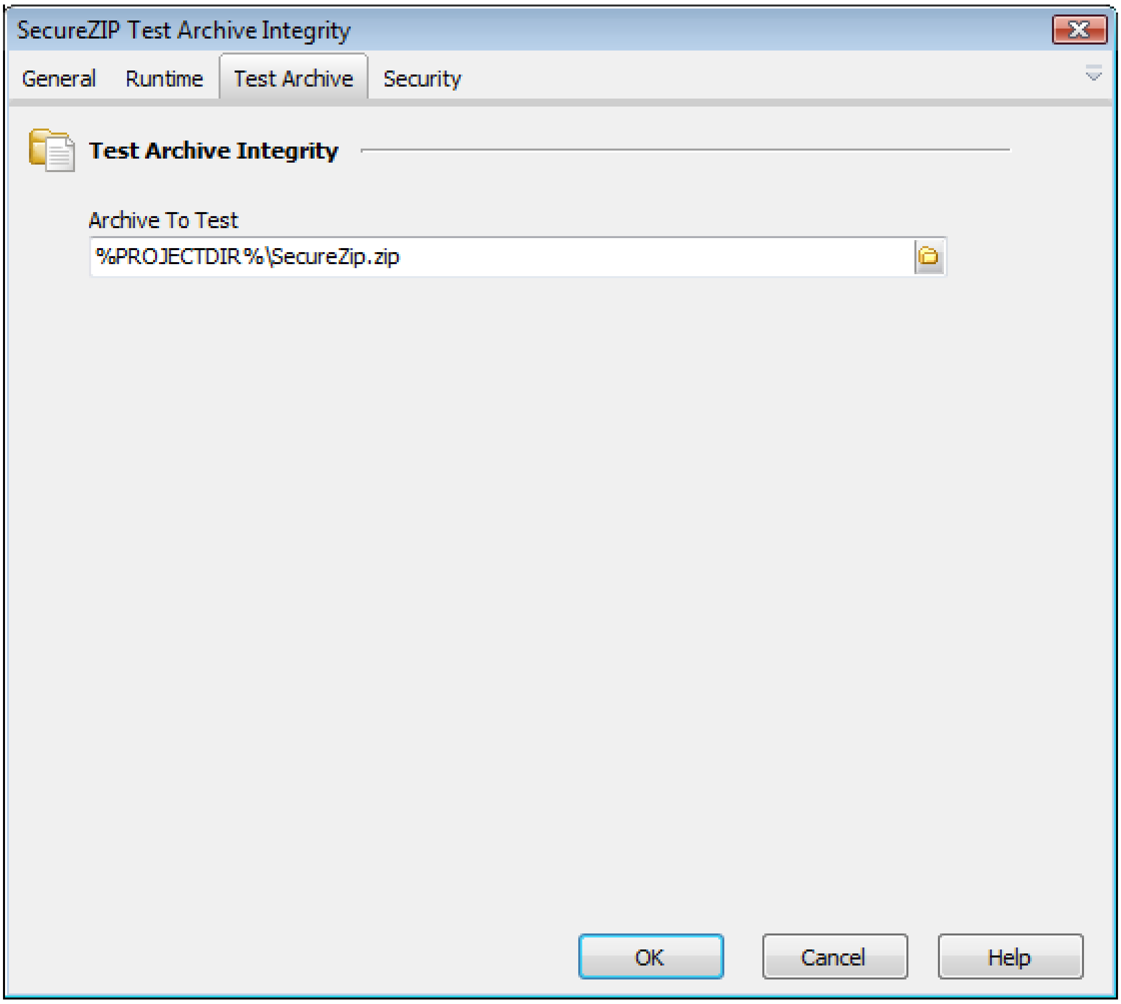
Additional Information

- To access a secure archive see the help for the SecureZIP Security Page.

6.4.4.4 SecureZIP Test Archive Integrity

SecureZIP Test Archive Integrity

The SecureZIP Test Archive Integrity action tests an archive to ensure that there is no damage that could prevent extraction from occurring.



Archive To Test: %PROJECTDIR%\SecureZip.zip

Simply specify the archive that you wish to test. from the *Test Archive* page.

SecureZIP tests the archive by extracting the contents without writing the items to disk.

Additional Information

- To access a secure archive see the help for the SecureZIP Security Page.

6.4.4.5 SecureZIP Common Pages

The SecureZIP actions contain the following shared pages:

- SecureZIP Security Page
- SecureZIP Filters Page
- SecureZIP Exclude List Page

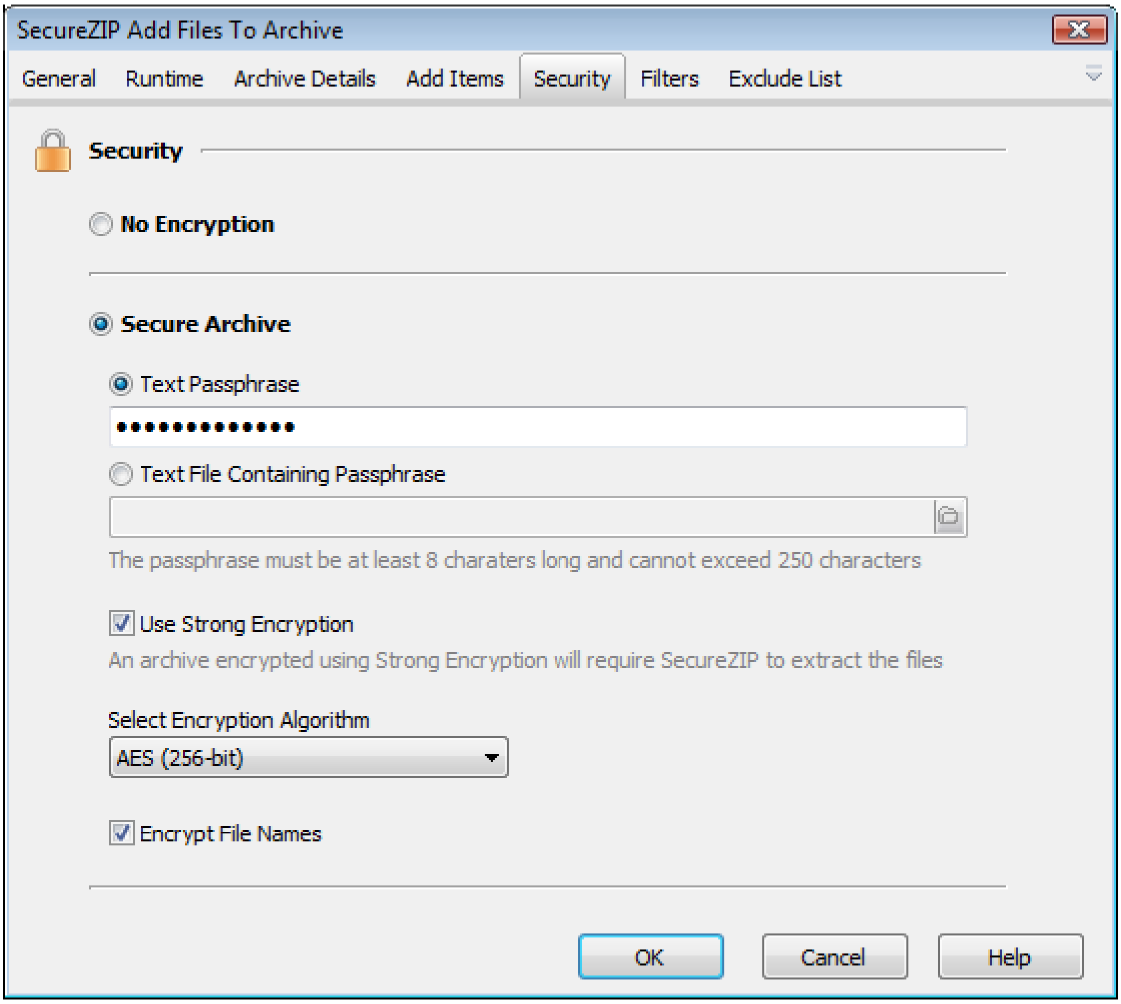
6.4.4.5.1 SecureZIP Security Page

The *Security* Page allows you to work with secure archives.

Securing an Archive

When creating an archive the *Security* page offers two core options:

- **No Encryption** - The contents of the archive being created will be accessible to everyone.
- **Secure Archive** - Depending on the security options specified the contents of the archive will be protected and only be accessible to those who can provide the correct credentials.



○ No Encryption

● Secure Archive

● Text Passphrase

•••••••••••••

○ Text File Containing Passphrase

The passphrase must be at least 8 charaters long and cannot exceed 250 characters

☑ Use Strong Encryption

An archive encrypted using Strong Encryption will require SecureZIP to extract the files

Select Encryption Algorithm

AES (256-bit)

☑ Encrypt File Names

To create a secure archive select the **Secure Archive** option and use the following security options provided:

- **Text Passphrase** - Provide a passphrase in the field provided.
- **Text File Passphrase** - Provide the location of a text file that contains a passphrase.
- **Use Strong Encryption** - Select this option to allow strong encryption to be used.

The contents of an archive that was created using strong encryption cannot be accessed without SecureZIP and the passphrase that was used to encrypt the archive.

- **Encryption Algorithm** - The strong encryption algorithm to use:
  - AES 256-bit
  - AES 192-bit
  - AES 128-bit
  - 3DES 168-bit
- Encrypt File Names - Enabling this option prevents an un-authorised user from even opening the archive to view the file names. Attempting to open an archive that was created using this option via Windows Explorer results in an error message indicating that the archive is invalid.

Note: The value for passphrase must be between at least 8 characters and not exceed 250 characters in length.

Accessing a Secure Archive

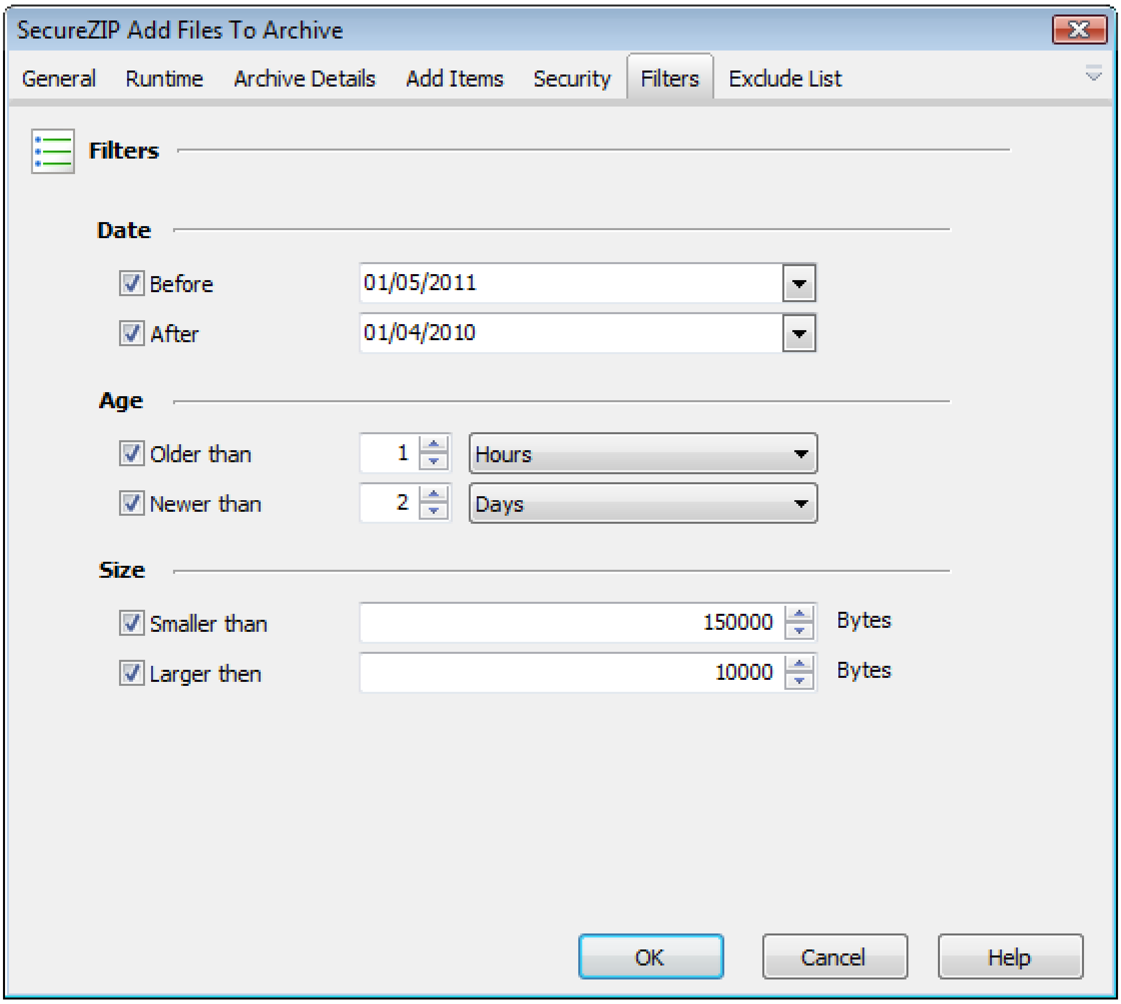
To access a secure archive you must provide the passphrase that was used to secure the archive.

On the *Security* Page specify the passphrase that was used to secure the archive. This can be done either of the two methods:

- **Text Passphrase** - Provide a passphrase in the field provided.
- **Text File Passphrase** - Provide the location of a text file that contains a passphrase.

6.4.4.5.2 SecureZIP Filters Page

The SecureZIP *Filters* page can be used to specify the criteria that items have to meet before they are selected when creating and extracting archives.



Filters

Date

☑ Before 01/05/2011

☑ After 01/04/2010

Age

☑ Older than 1 Hours

☑ Newer than 2 Days

Size

☑ Smaller than 150000 Bytes

☑ Larger then 10000 Bytes

The *Filters* page allows you to specify the following criteria:

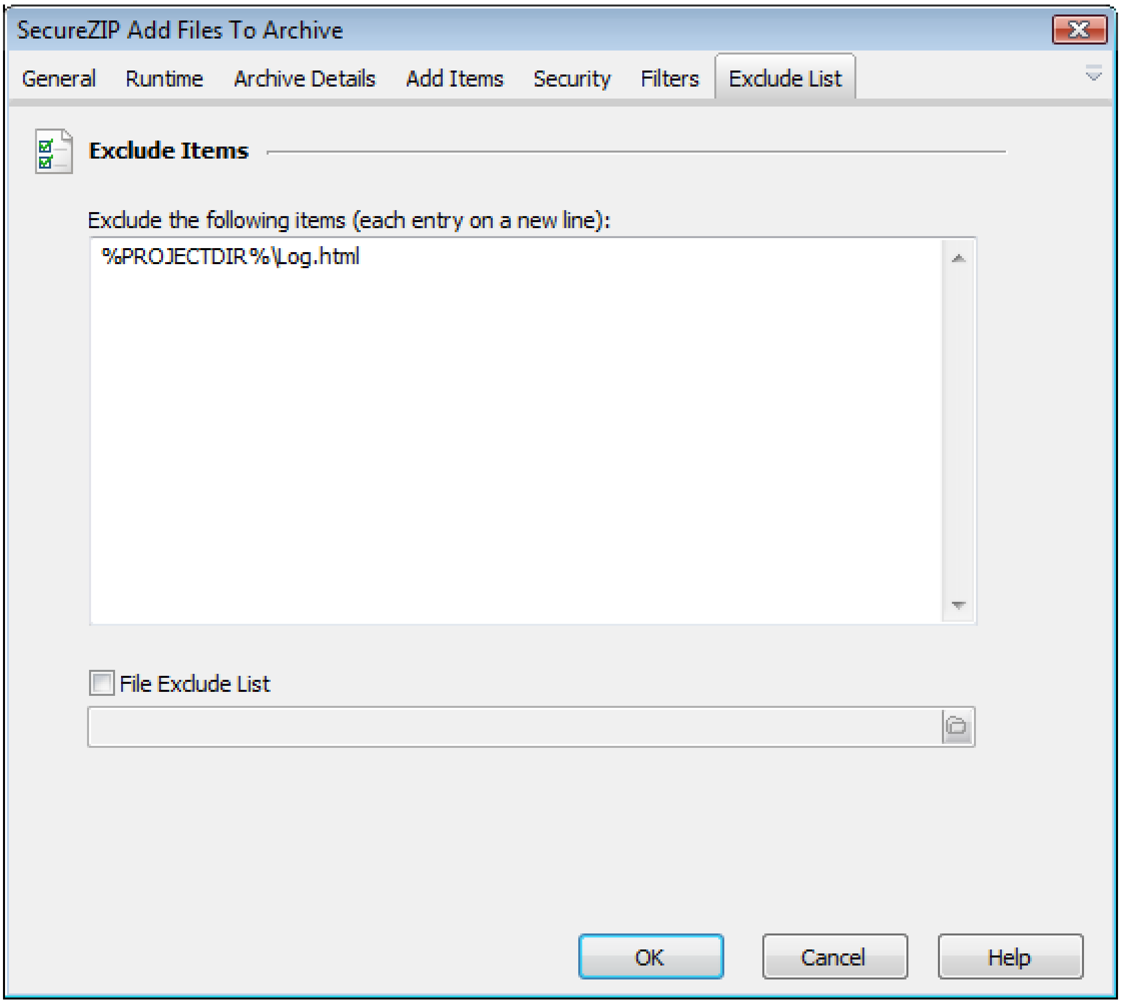
- **Date** - Select files based on the date that they were last modified.
  - **Before** - Files that were modified before the specified date.
  - **After** - Files that were modified after the specified date.
- **Age** - Select files based on their age. Specify a value and a unit.
  - **Older than** - Only select files that are older than the period specified.
  - **Newer than** - Only select files that are newer than the period specified.
- **Size** - Select files based on their size.
  - **Smaller than** - Only select items smaller than the specified number of bytes.
  - **Larger than** - Only select items larger than the specified number of bytes.

Note that if the filters are set in a way that no files meet the criteria the action will fail.

Before using Date filters make sure you have specified the expected date format on the options page. For more information on specifying the date format click here.

6.4.4.5.3 SecureZIP Exclude List Page

The SecureZIP *Exclude List* page allows you specify a list of items that will be ignored when creating or extracting an archive.



Exclude the following items (each entry on a new line):

```
%PROJECTDIR%\Log.html
```

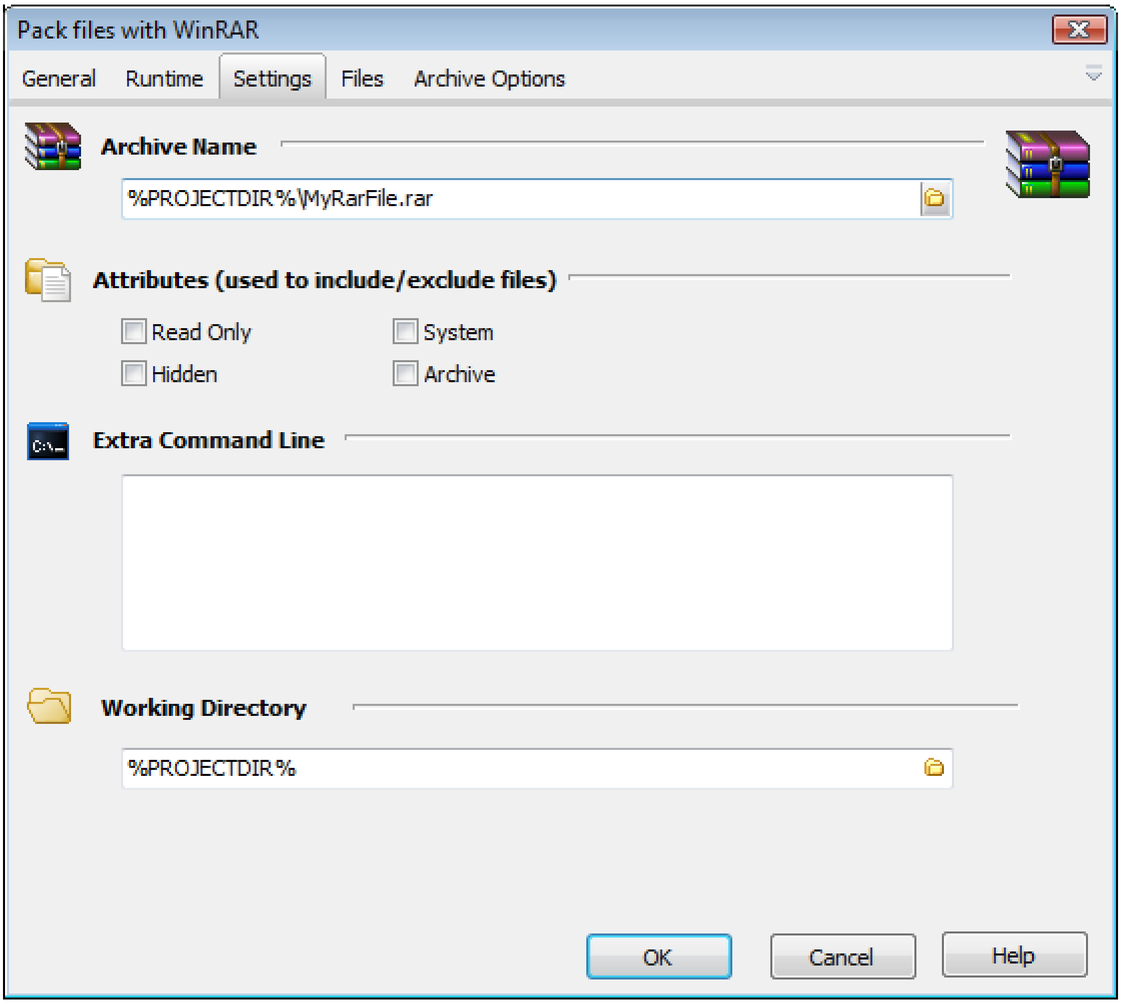
□ File Exclude List

On the *Exclude List* page enter the names of the files that you wish to exclude from the process and when the action is run these files will be ignored.

The *Exclude List* page also provides an option to specify a **File Exclude List** which allows you to specify a text file that contains the exclude list.

6.4.5 WinRAR Action

The WinRAR action enables you create RAR archives using WinRAR.



Archive Name: %PROJECTDIR%\MyRarFile.rar

Attributes (used to include/exclude files)

□ Read Only □ System
□ Hidden □ Archive

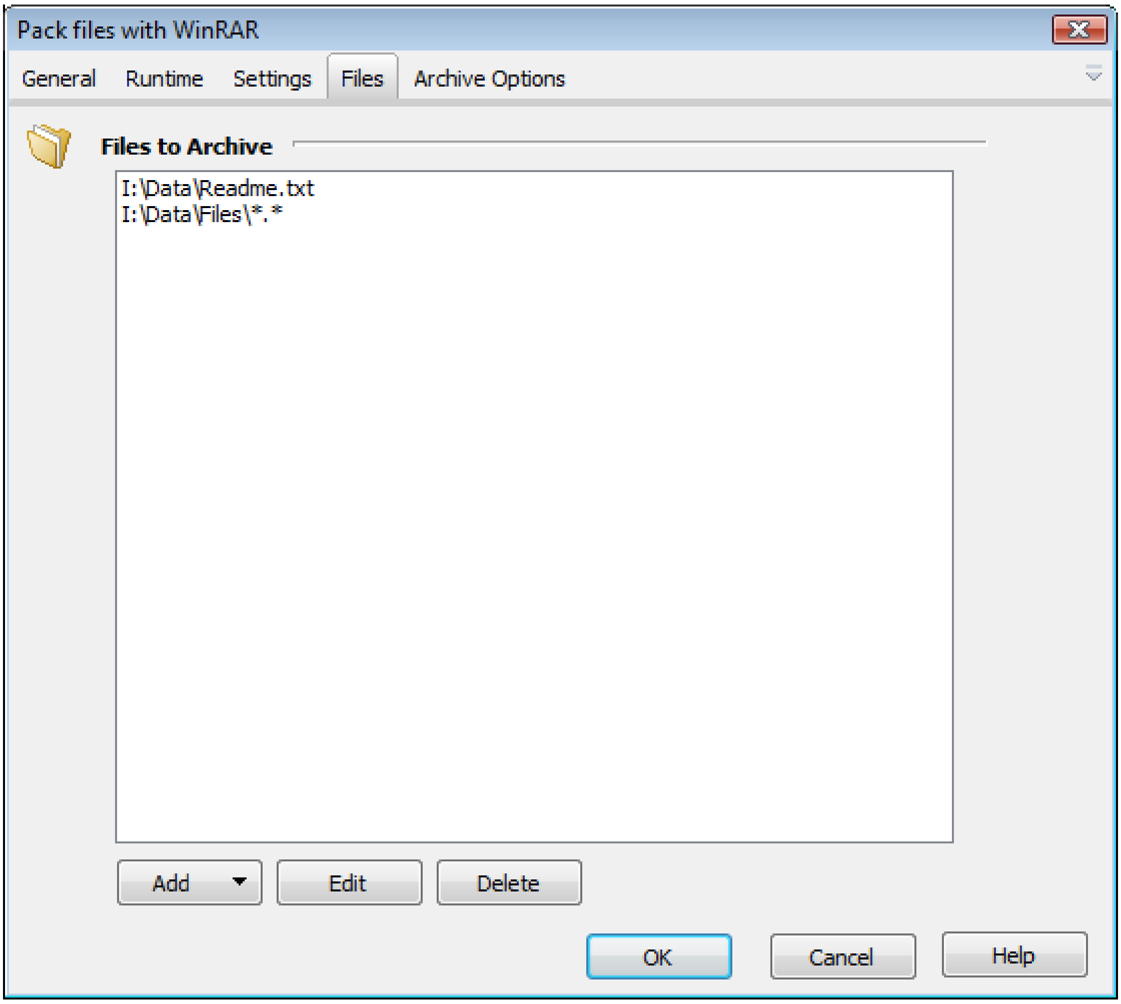
Extra Command Line

Working Directory: %PROJECTDIR%

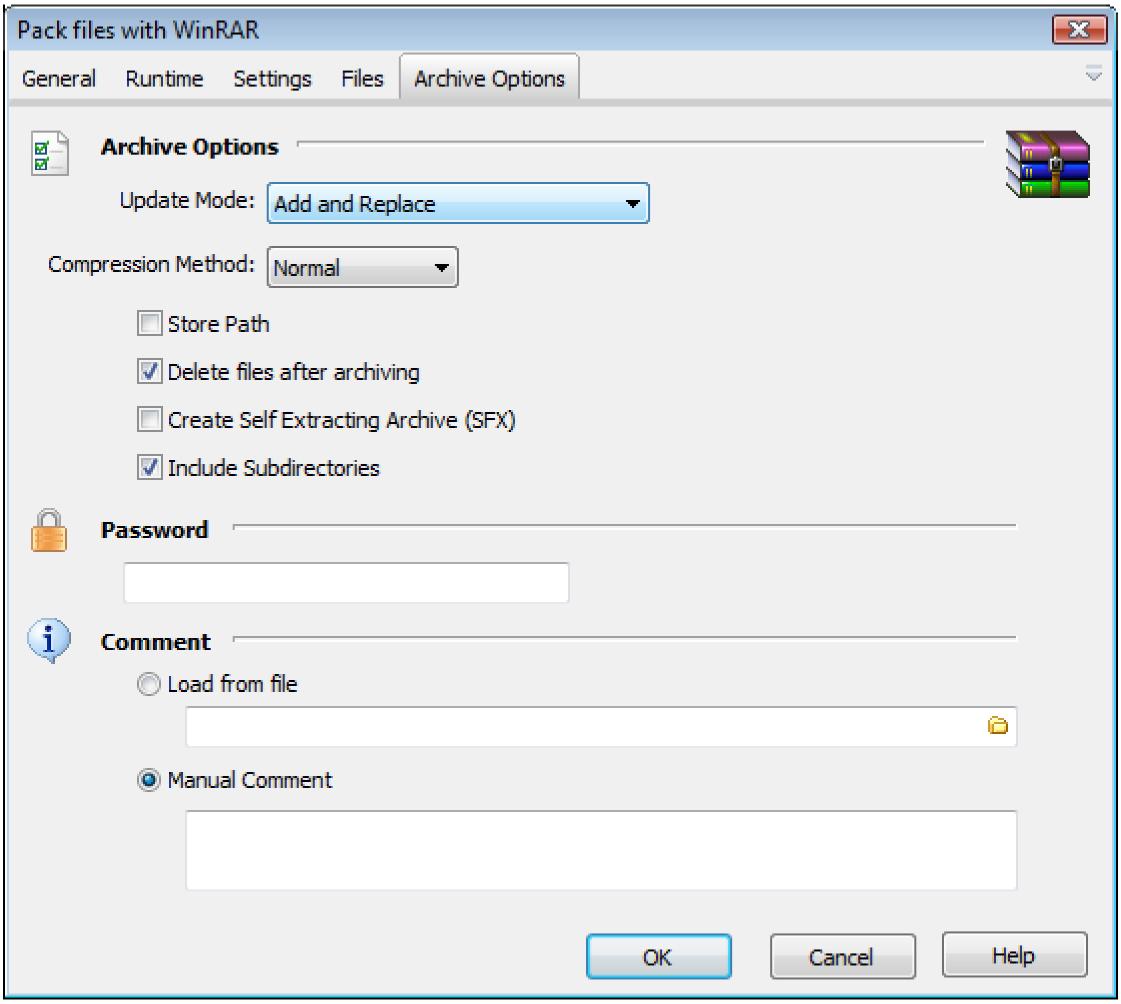
Archive Name - specify the name of the archive to create

Attributes - In the above screenshot, only files with the Read Only attribute, and not system files will be added to the archive.

Extra Command Line - If there are any WinRAR options which the FB action doesn't surface, then you can manually specify them



```
I:\Data\Readme.txt
I:\Data\Files\*.*
```



Pack files with WinRAR

Archive Options

Update Mode: Add and Replace

Compression Method: Normal

□ Store Path

☑ Delete files after archiving

□ Create Self Extracting Archive (SFX)

☑ Include Subdirectories

Password

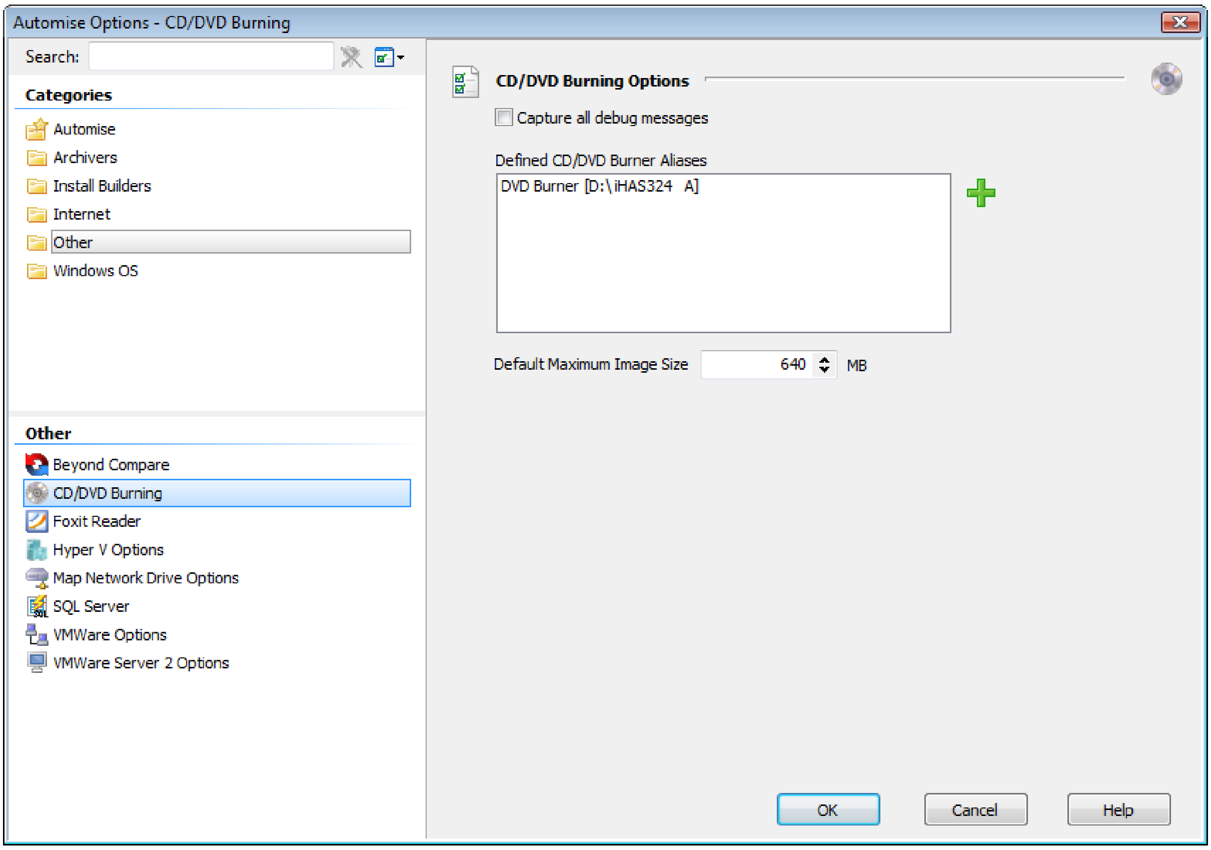
Comment

○ Load from file

● Manual Comment

6.5 CD/DVD Burner Actions

In order to use the CD/DVD Burner actions, you need to define Aliases for the burner(s) available on your machine. Aliases are what you use to determine which burner an action will use. This allows you to define the same alias for different burners on other machines, enabling the project to run even though other machines have different burners.



Automise Options - CD/DVD Burning

Search:

Categories

- Automise
- Archivers
- Install Builders
- Internet
- Other
- Windows OS

Other

- Beyond Compare
- CD/DVD Burning
- Foxit Reader
- Hyper V Options
- Map Network Drive Options
- SQL Server
- VMWare Options
- VMWare Server 2 Options

CD/DVD Burning Options

□ Capture all debug messages

Defined CD/DVD Burner Aliases

DVD Burner [D:\iHAS324 A]

Default Maximum Image Size 640 MB

Automise uses the Windows IMAPI 2.0 api for CD/DVD burning. This must be installed for these actions to work correctly :

Windows XP SP3:
http://support.microsoft.com/kb/932716/en-us

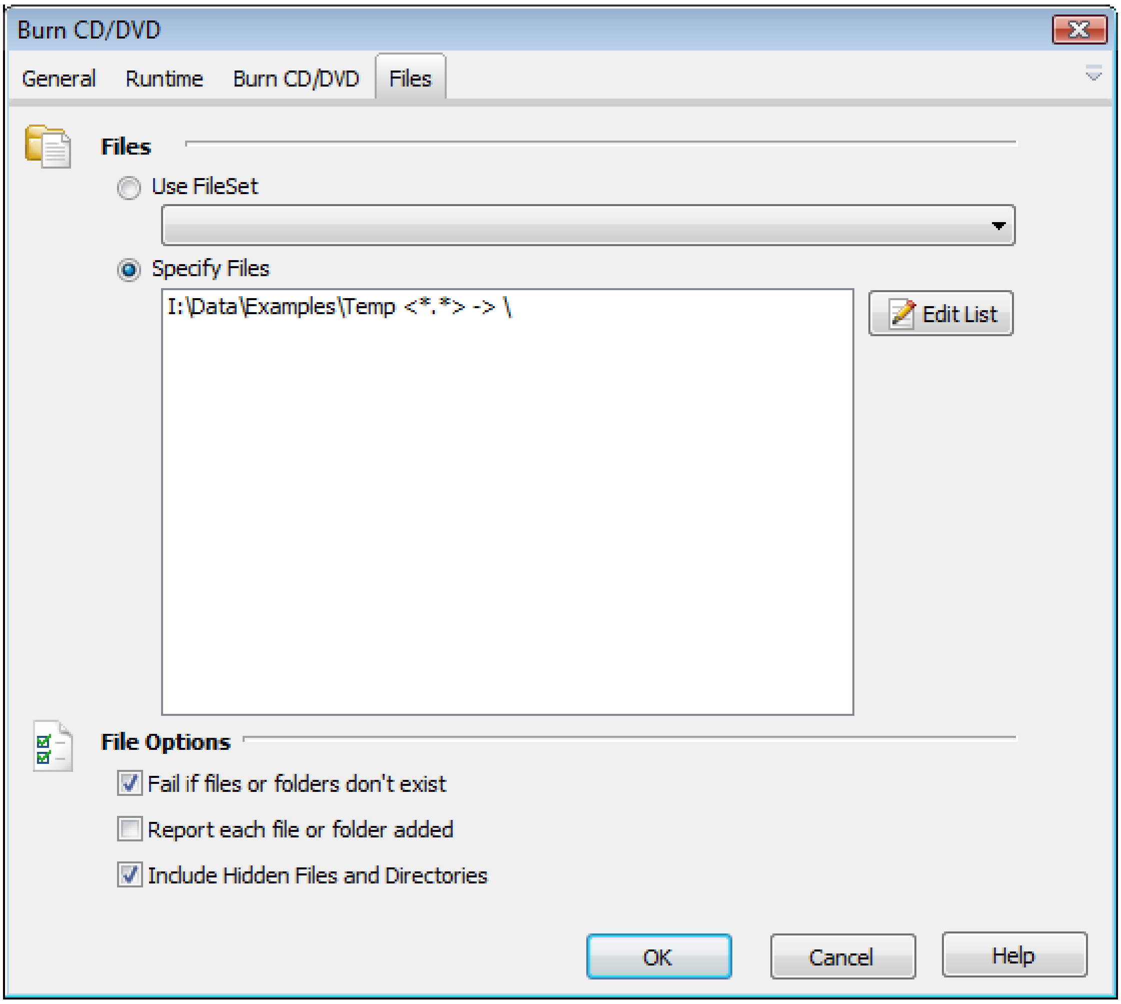
Windows Server 2003 :
http://support.microsoft.com/kb/932716/en-us

Windows XP SP3
Windows Server 2003 SP1 and above
Windows Vista RTM, SP1
Windows Server 2008

Windows Feature Pack for Storage 1.0 Download - Adds bluray and burn verification support, recommended.

6.5.1 Burn CD/DVD Action

This action supports burning files direct to CDR/CDRW/DVDR & DVDRW Media (depending on your burner hardware). The files burned directly to the drive, ie an image is not created first. Note that to use this action you need to define an alias for your burner first (see here for more info).



Files

○ Use FileSet

● Specify Files

I:\Data\Examples\Temp <*.*> -> \

File Options

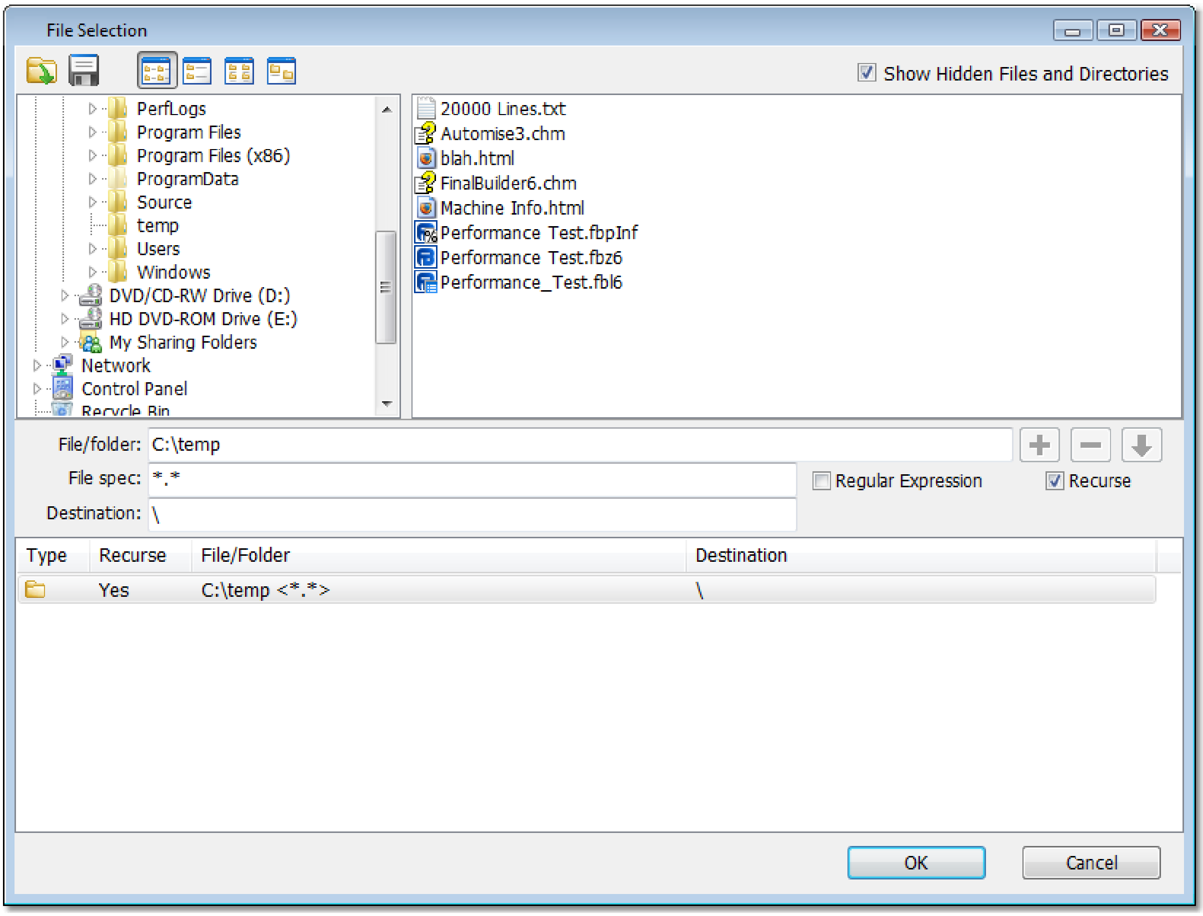
☑ Fail if files or folders don't exist

□ Report each file or folder added

☑ Include Hidden Files and Directories

If you are having problems seeing the files on your DVD or CD after burning, then you might have to enable the "Eject after Write" option so that the operating system refreshes the contents of the media.

The File Selection editor allows you to select files & folders and use wildcards or regular expressions to add files to the selection.



☑ Show Hidden Files and Directories

File/folder: C:\temp

File spec: *.*

Destination: \

☐ Regular Expression

☑ Recurse

Type	Recurse	File/Folder	Destination
🗀	Yes	C:\temp <*.*>	\

In the File spec you can enter a normal DOS type file spec using wild cards * and ?

You can also specify a regular expression, but you need to select the "Regular Expression" checkbox. If you are specifying *.* you should not select regular expression as it is an invalid regular expression.

Other Options:

"Joliet File System"

Enables the Joliet extensions to the ISO9660 file system. This allows you to have file names longer than 8.3 characters. In almost all instances, you'll want to leave this option enabled.

"Close Disc (Finalize)"

Enable this option if you don't want to write any further data to the disc.

"Test Write"

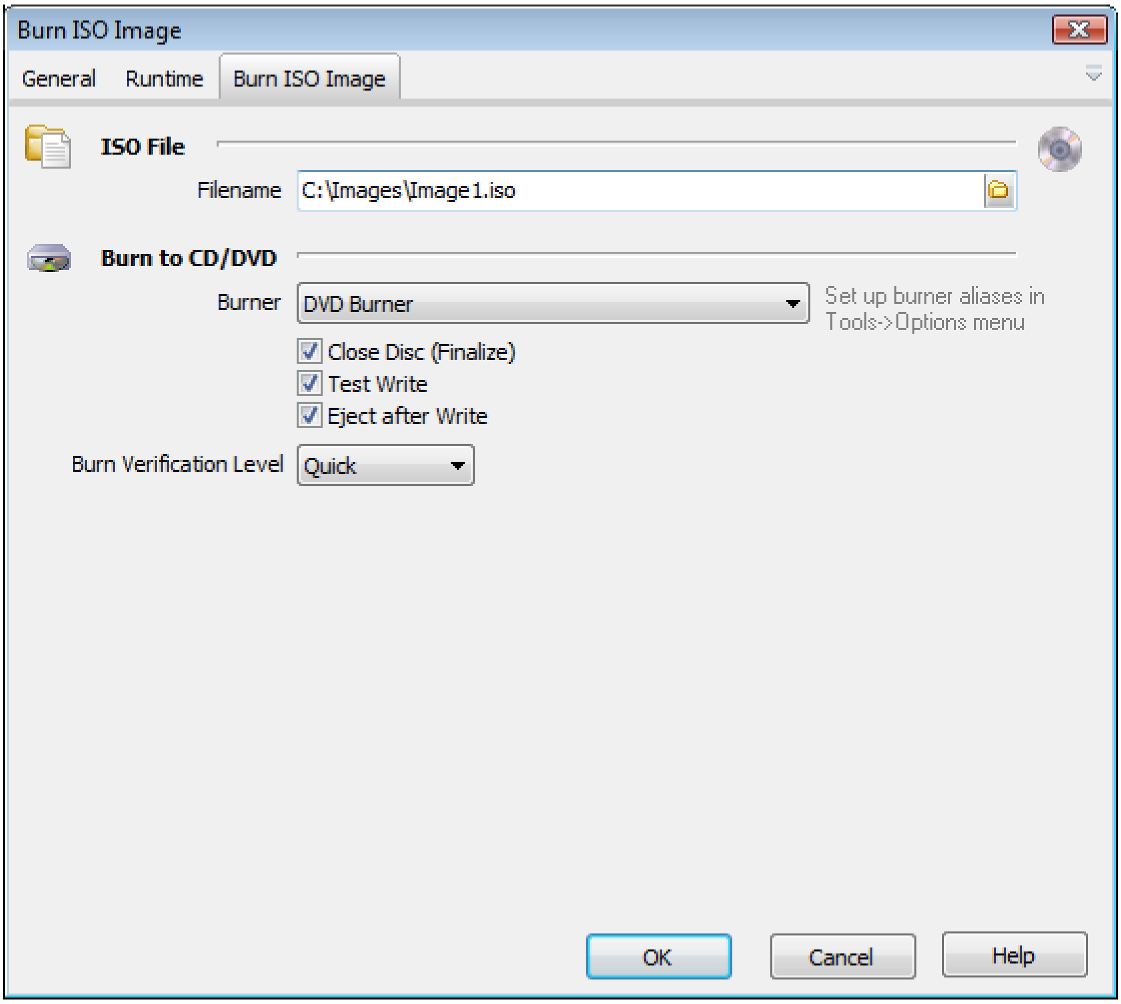
Enable this option if you don't want to actually write any data to the disc.

"Eject after write"

Enable this option to eject the disc when writing completes. If you are having problems seeing the files on your DVD or CD after burning, then you might have to enable this option.

6.5.2 Burn ISO Action

This action supports burning an ISO file generated by Automise(or any other tool that generates valid ISO Files) to CDR/CDRW/DVDR & DVDRW Media (depending on your burner hardware). Note that to use this action you need to define an alias for your burner first (see here for more info).



ISO File

Filename C:\Images\Image1.iso

Burn to CD/DVD

Burner DVD Burner

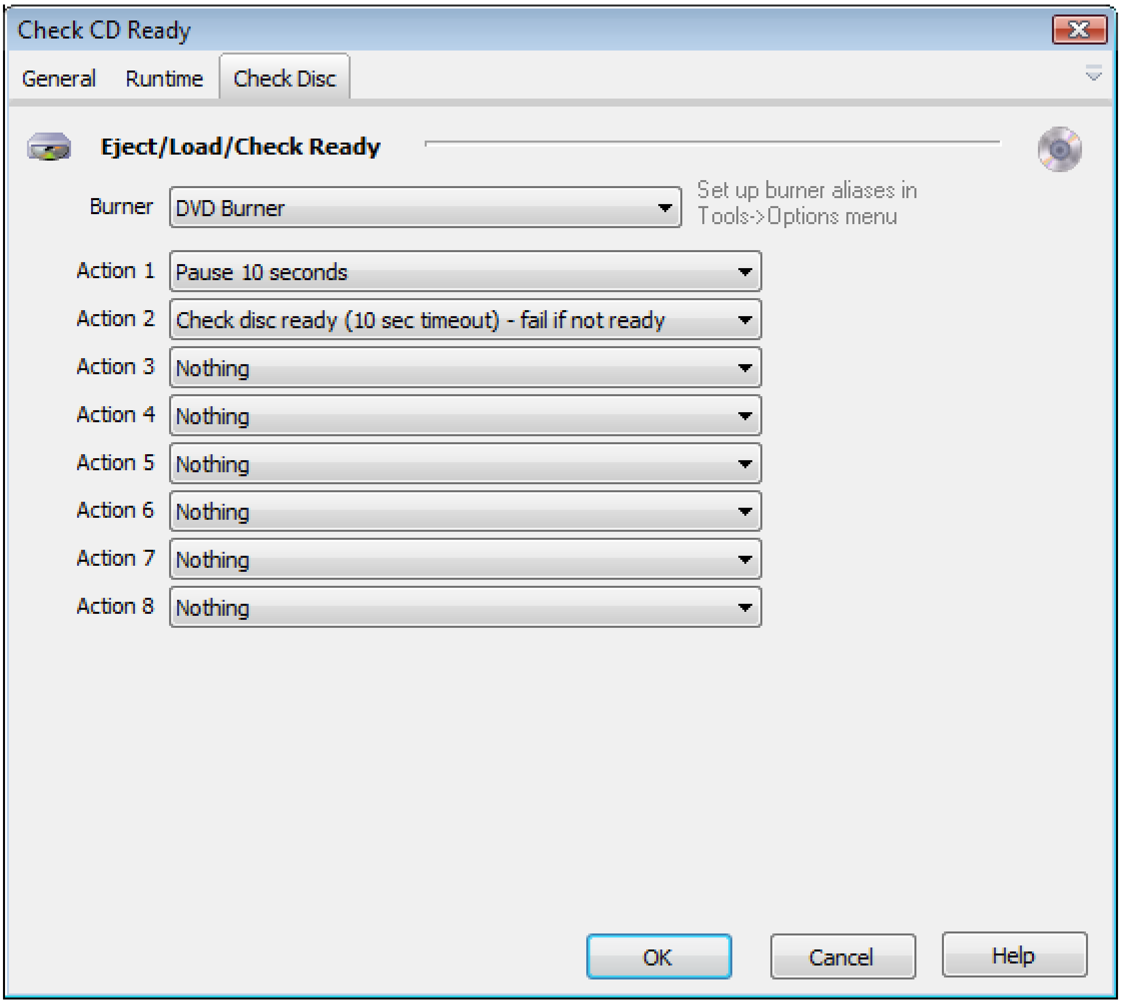
Set up burner aliases in Tools->Options menu

☑ Close Disc (Finalize)
☑ Test Write
☑ Eject after Write

Burn Verification Level Quick

6.5.3 Check Ready Action

This action allows you to automate to some degree the process of loading a CD into the burner, and making sure the burner is ready to burn before attempting to burn to CD. An example which shows the use of this action is installed in the Automise\Examples\CDBurner directory. Note that to use this action you need to define an alias for your burner first (see here for more info).



Burner: DVD Burner

Set up burner aliases in Tools->Options menu

Action 1: Pause 10 seconds

Action 2: Check disc ready (10 sec timeout) - fail if not ready

Action 3: Nothing

Action 4: Nothing

Action 5: Nothing

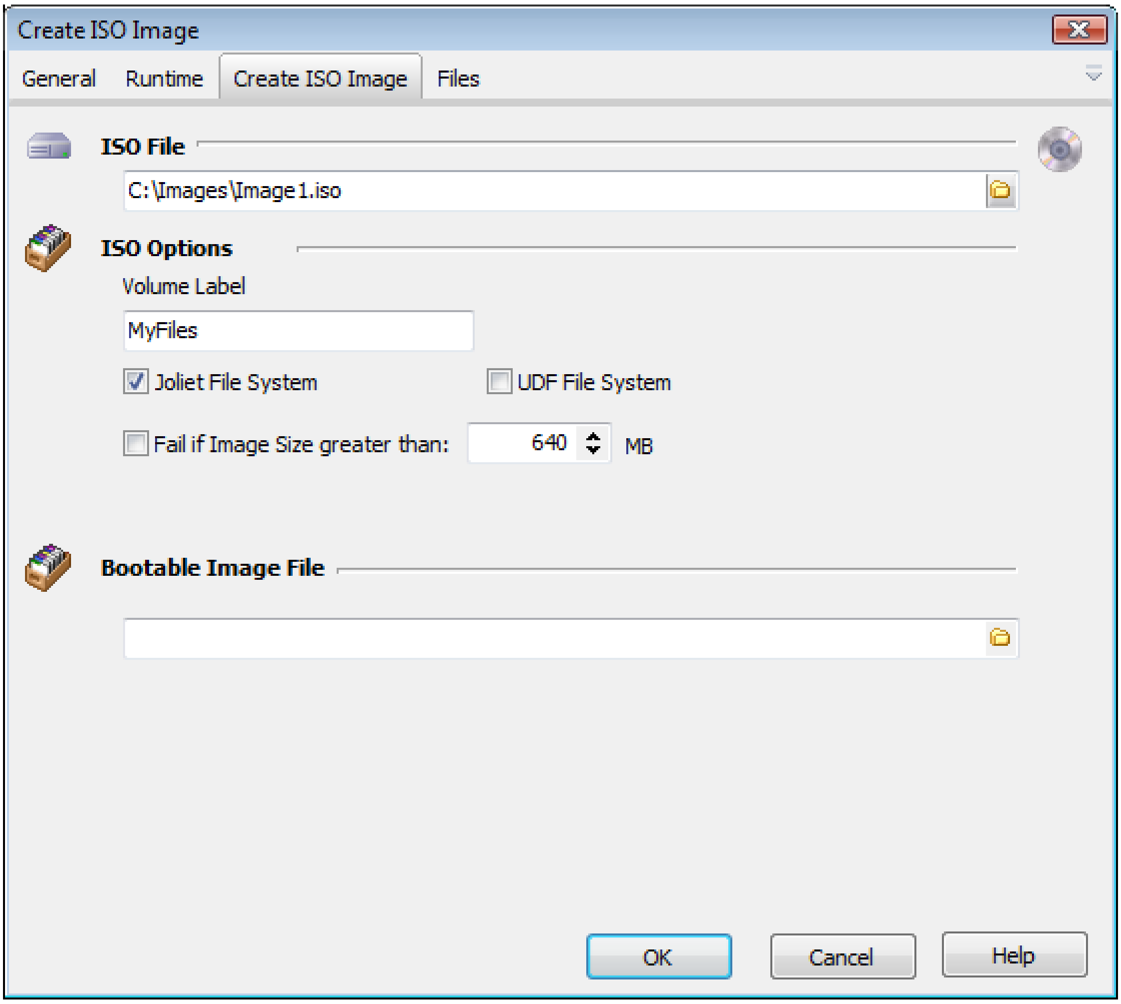
Action 6: Nothing

Action 7: Nothing

Action 8: Nothing

6.5.4 Create ISO Action

This action allows you to create an ISO CD/DVD Image File. The File Selection is exactly the same as in the Burn CD Action. Note that Automise does not impose any limits on the size of the resulting ISO File.



C:\Images\Image1.iso

Volume Label: MyFiles

☑ Joliet File System

□ UDF File System

□ Fail if Image Size greater than: 640 MB

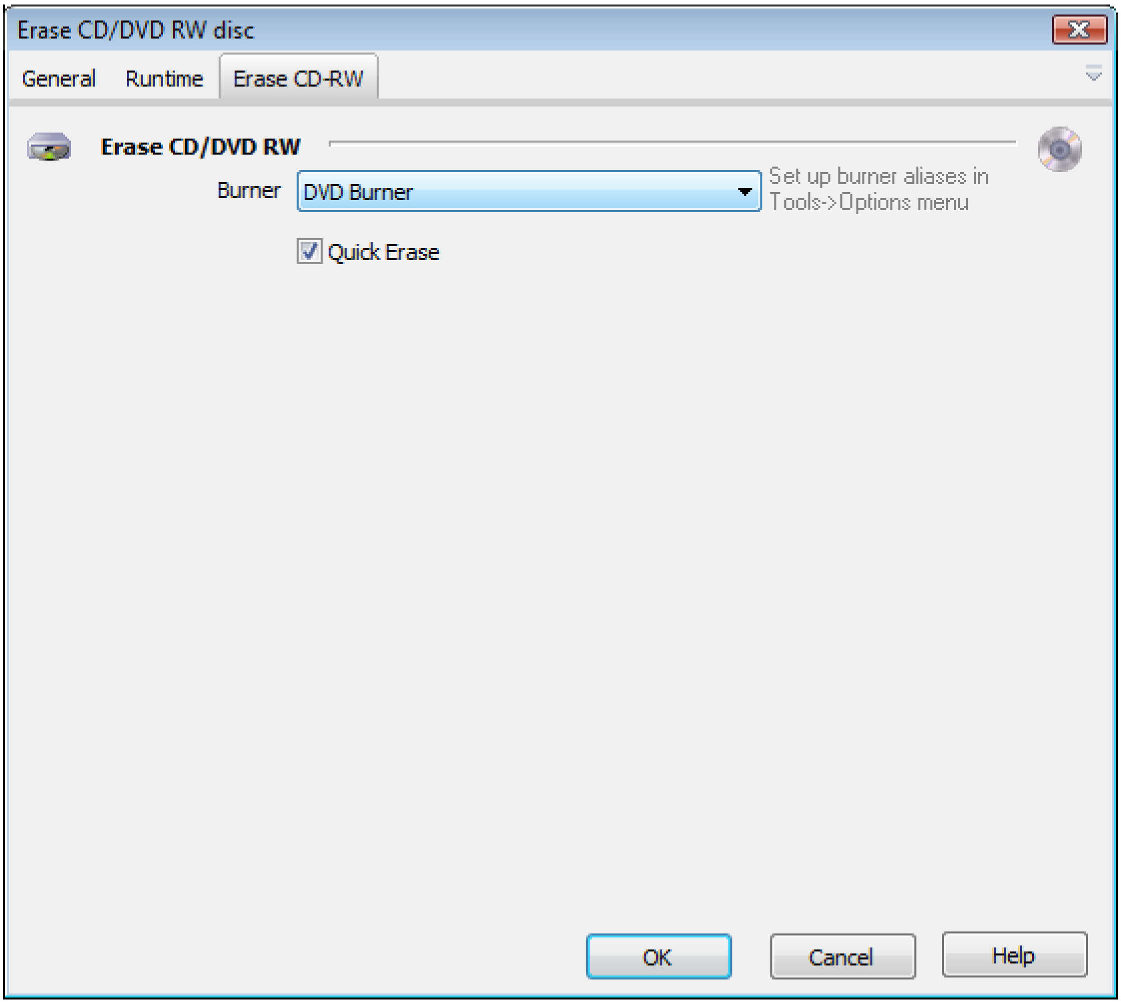
"Joliet File System"

Enables the Joliet extensions to the ISO9660 file system. This allows you to have file names longer than 8.3 characters. In almost all instances, you'll want to leave this option enabled.

6.5.5 Erase CD/DVD RW

This action allows you to Erase rewritable CD/DVD media (if your burner supports this). Note that to use this action you need to define an alias for your burner first (see here for more info). If your burner does not support rewritable media then it will not appear in the list of burner aliases.

Quick Erase is recommended for most media and is substantially faster (full erase can take a long time on a DVDRW!).



Burner: DVD Burner

Set up burner aliases in Tools->Options menu

☑ Quick Erase

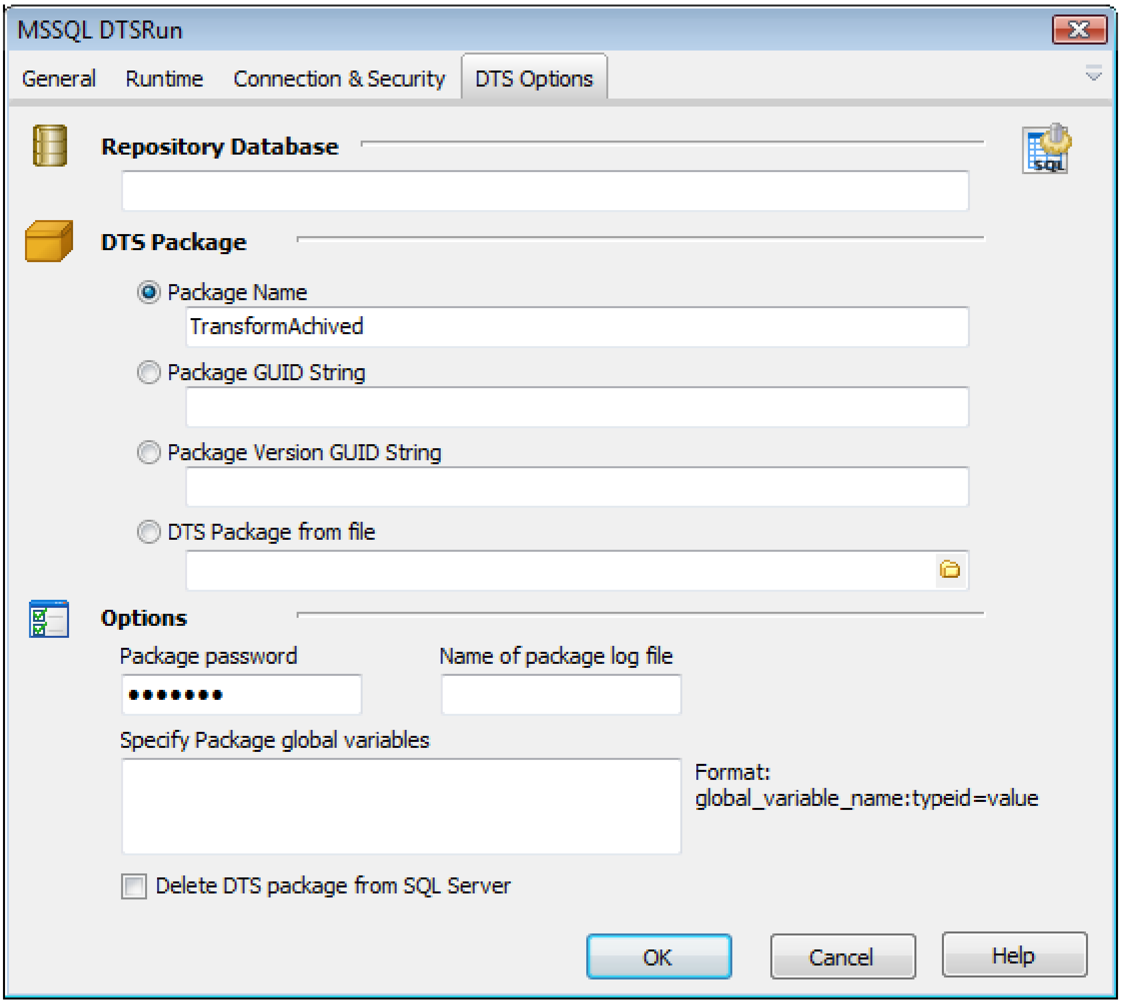
6.6 Database

6.6.1 SQL Server Actions

6.6.1.1 DTSRun Action

[Automise Professional Edition]

The dtsrun utility executes a package created using Data Transformation Services (DTS). The DTS package can be stored in the Microsoft® SQL Server™ msdb database, a COM-structured storage file, or SQL Server Meta Data Services.



MSSQL DTSRun

Repository Database

DTS Package

● Package Name

TransformAchived

○ Package GUID String

○ Package Version GUID String

○ DTS Package from file

Options

Package password: •••••••

Name of package log file:

Specify Package global variables

Format:
global_variable_name:typeid=value

□ Delete DTS package from SQL Server

For information on the options for this see:

http://msdn.microsoft.com/library/en-us/coprompt/cp_dtsrun_95kp.asp

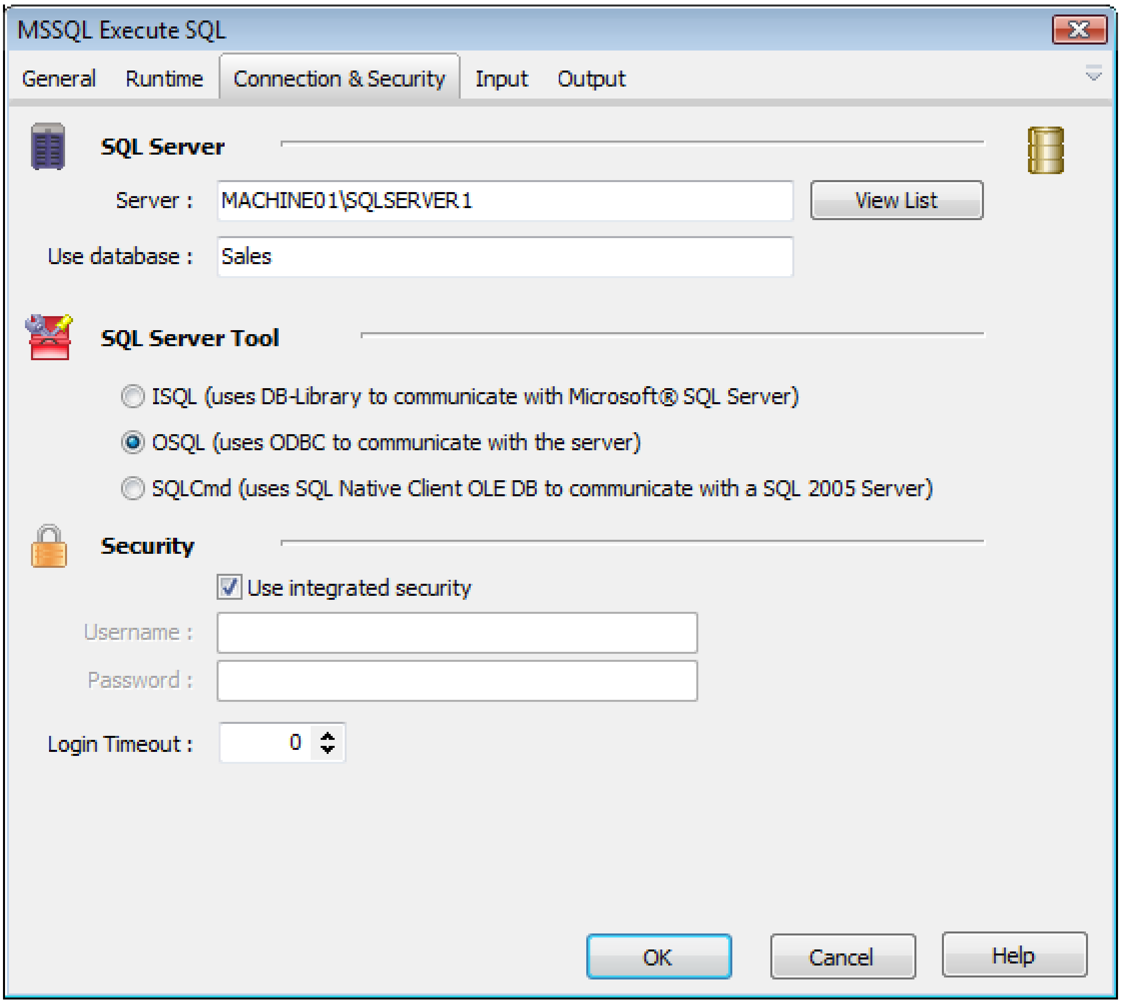
To set the location of DTSRun see SQL Server Options

6.6.1.2 Execute SQL Action

[Automise Professional Edition]

Execute an SQL statement against Microsoft SQL Server and capture the result set.

Connection & Security



Server : MACHINE01\SQLSERVER1

Use database : Sales

○ ISQL (uses DB-Library to communicate with Microsoft® SQL Server)

● OSQL (uses ODBC to communicate with the server)

○ SQLCmd (uses SQL Native Client OLE DB to communicate with a SQL 2005 Server)

☑ Use integrated security

Username :

Password :

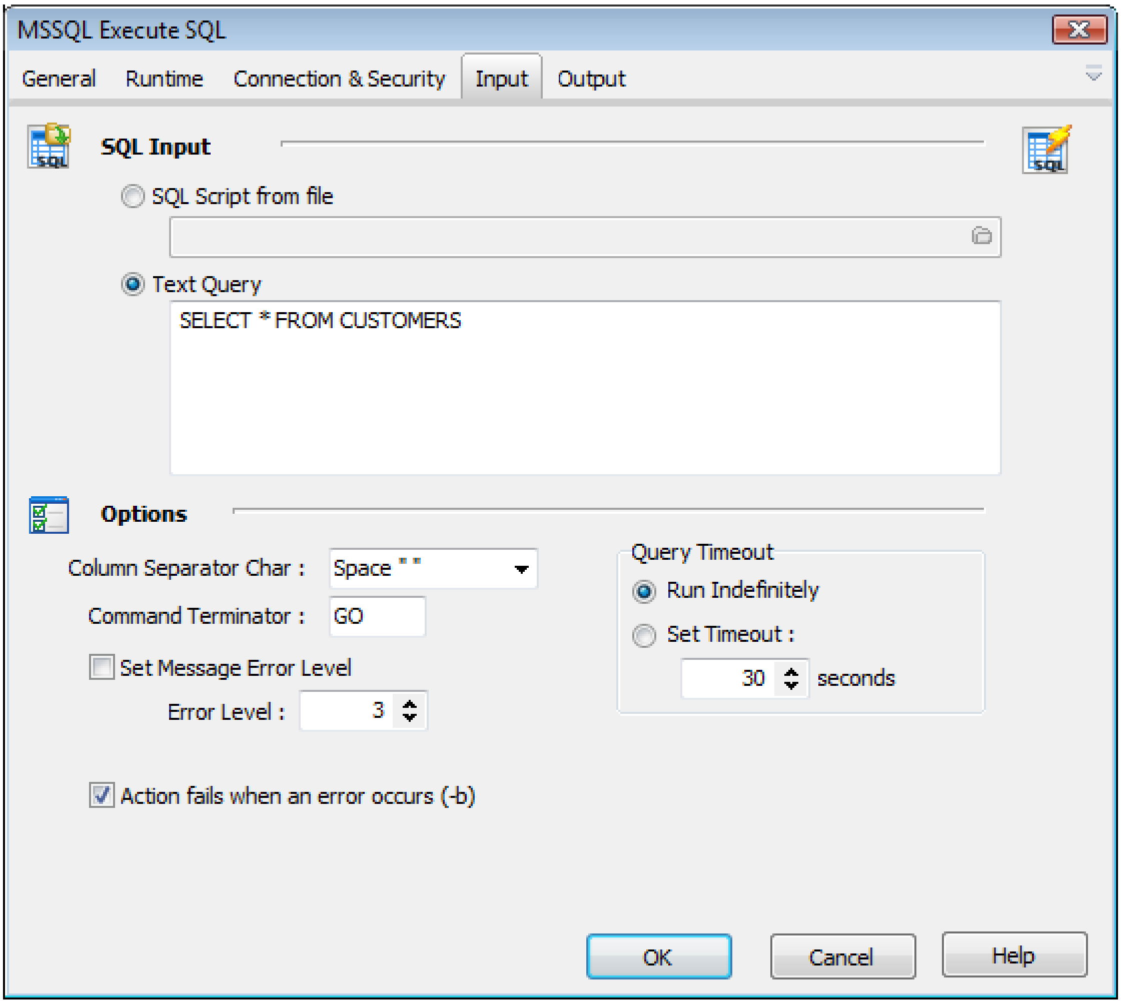
Login Timeout : 0

Server - Specify the MS SQL Server to use. Clicking "View List" will attempt to locate any SQL Servers on the network.
Use Database - Specify a database to run the SQL against.

SQL Server Tool - choose the method (isql, osql or sqlcmd) to use to connect with.

Security - If not using integrated (ie. Windows) security, you should specify the username and password to use.

Input



SQL Input

○ SQL Script from file

● Text Query

SELECT * FROM CUSTOMERS

Options

Column Separator Char : Space " "

Command Terminator : GO

□ Set Message Error Level

Error Level : 3

Query Timeout

● Run Indefinitely

○ Set Timeout :

30 seconds

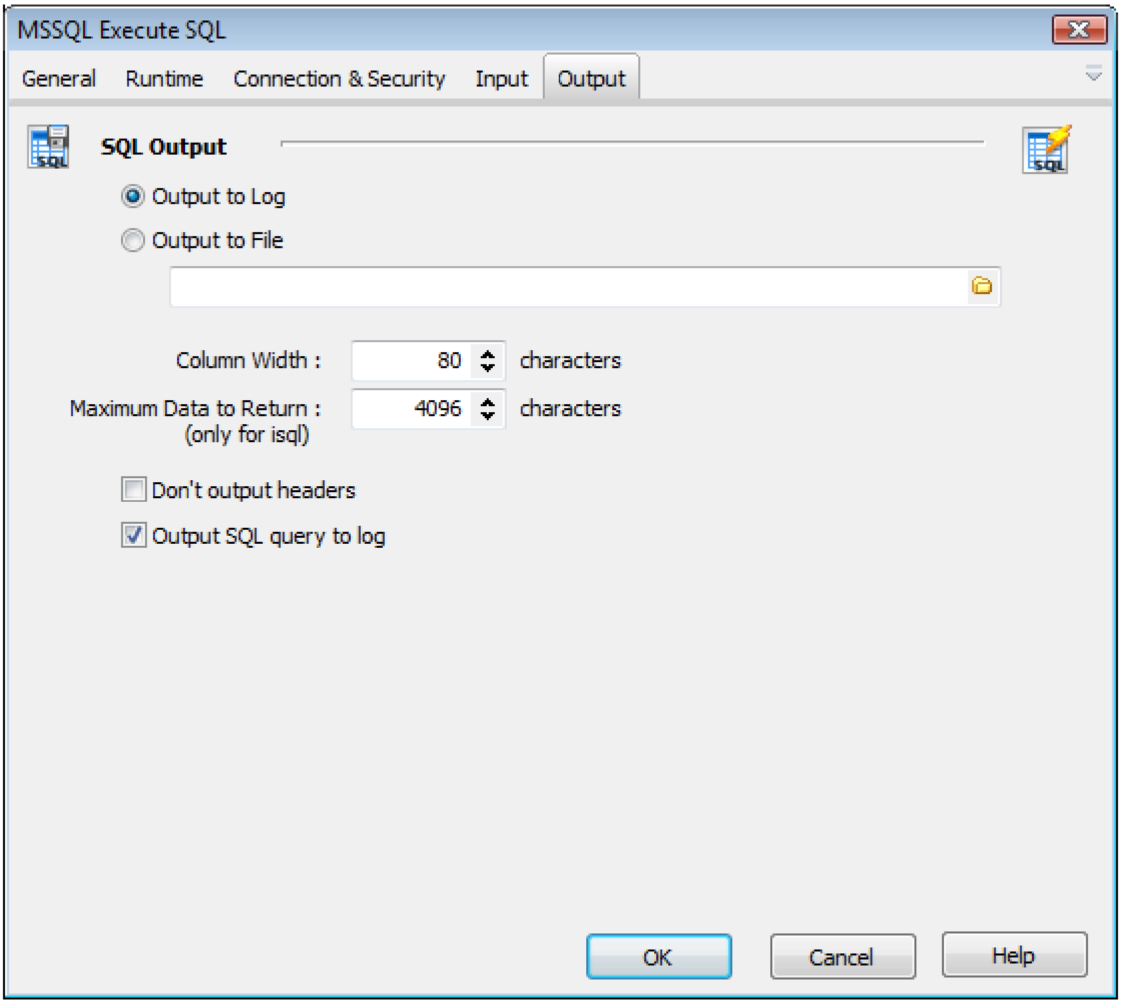
☑ Action fails when an error occurs (-b)

SQL Input - SQL Script from file - specify a file which contains an SQL Script.
- Text Query - Specify the SQL Query to execute.

For information on the other options see:

isql : http://msdn.microsoft.com/library/en-us/coprompt/cp_isql_8r39.asp
osql : http://msdn.microsoft.com/library/en-us/coprompt/cp_osql_1wxl.asp
sqlcmd : http://msdn2.microsoft.com/en-us/library/ms162773.aspx

Output


MSSQL Execute SQL

SQL Output

● Output to Log
○ Output to File

Column Width : 80 characters
Maximum Data to Return : (only for isql) 4096 characters

□ Don't output headers
☑ Output SQL query to log

The query results can either be output to a file or to the Automise Log.
For information on the other options see:

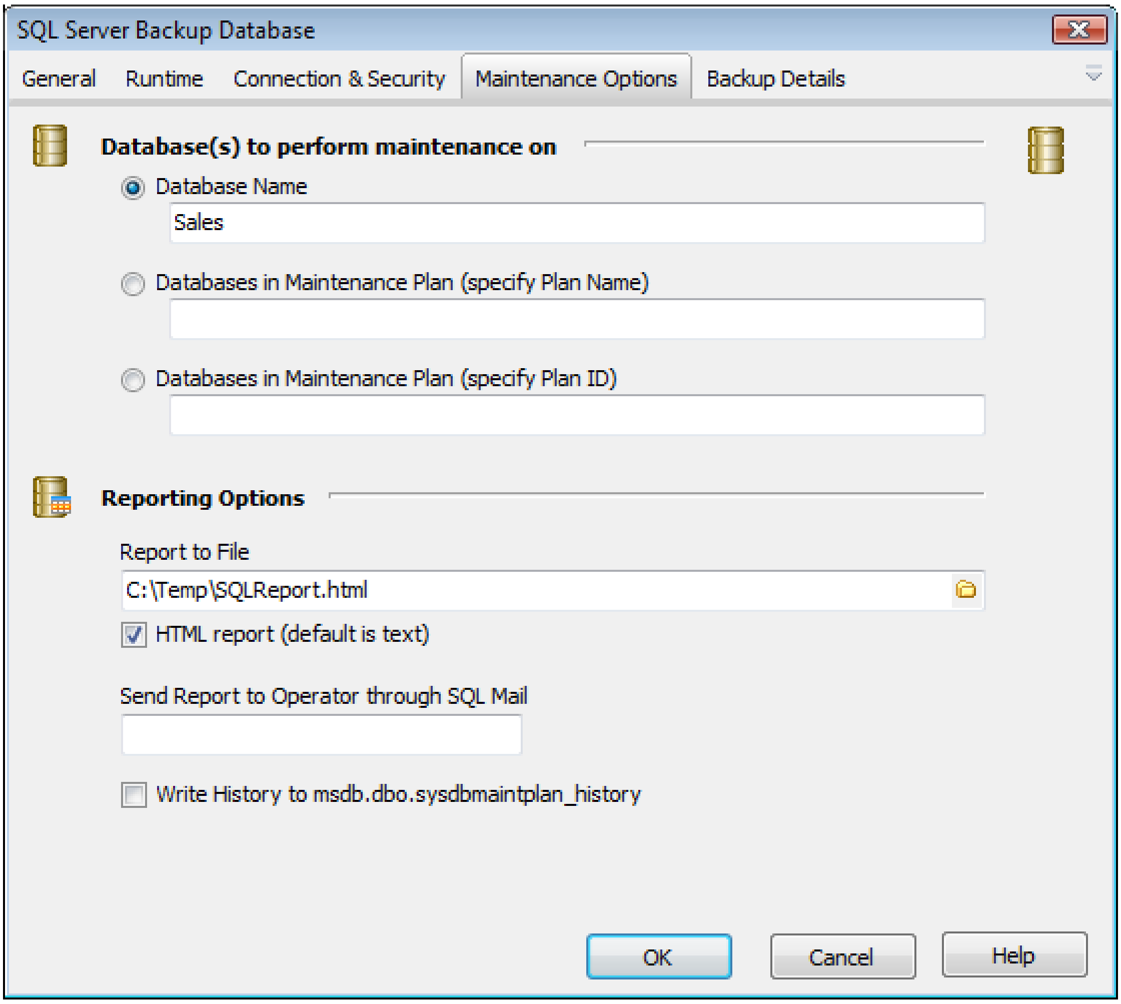
isql : http://msdn.microsoft.com/library/en-us/coprompt/cp_isql_8r39.asp
osql : http://msdn.microsoft.com/library/en-us/coprompt/cp_osql_1wxl.asp
sqlcmd : http://msdn2.microsoft.com/en-us/library/ms162773.aspx

To set the location of ISQL, OSQL and SQLCmd see SQL Server Options

6.6.1.3 SQL Server Backup Database

[Automise Professional Edition]

Performs a backup operation on the entire database or only the transaction log.
The file name for a database backup is generated automatically as follows:
dbname_db_yyyyMMddhhmm.BAK



Database(s) to perform maintenance on

● Database Name: Sales

○ Databases in Maintenance Plan (specify Plan Name)

○ Databases in Maintenance Plan (specify Plan ID)

Reporting Options

Report to File: C:\Temp\SQLReport.html

☑ HTML report (default is text)

Send Report to Operator through SQL Mail

☐ Write History to msdb.dbo.sysdbmaintplan_history

For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

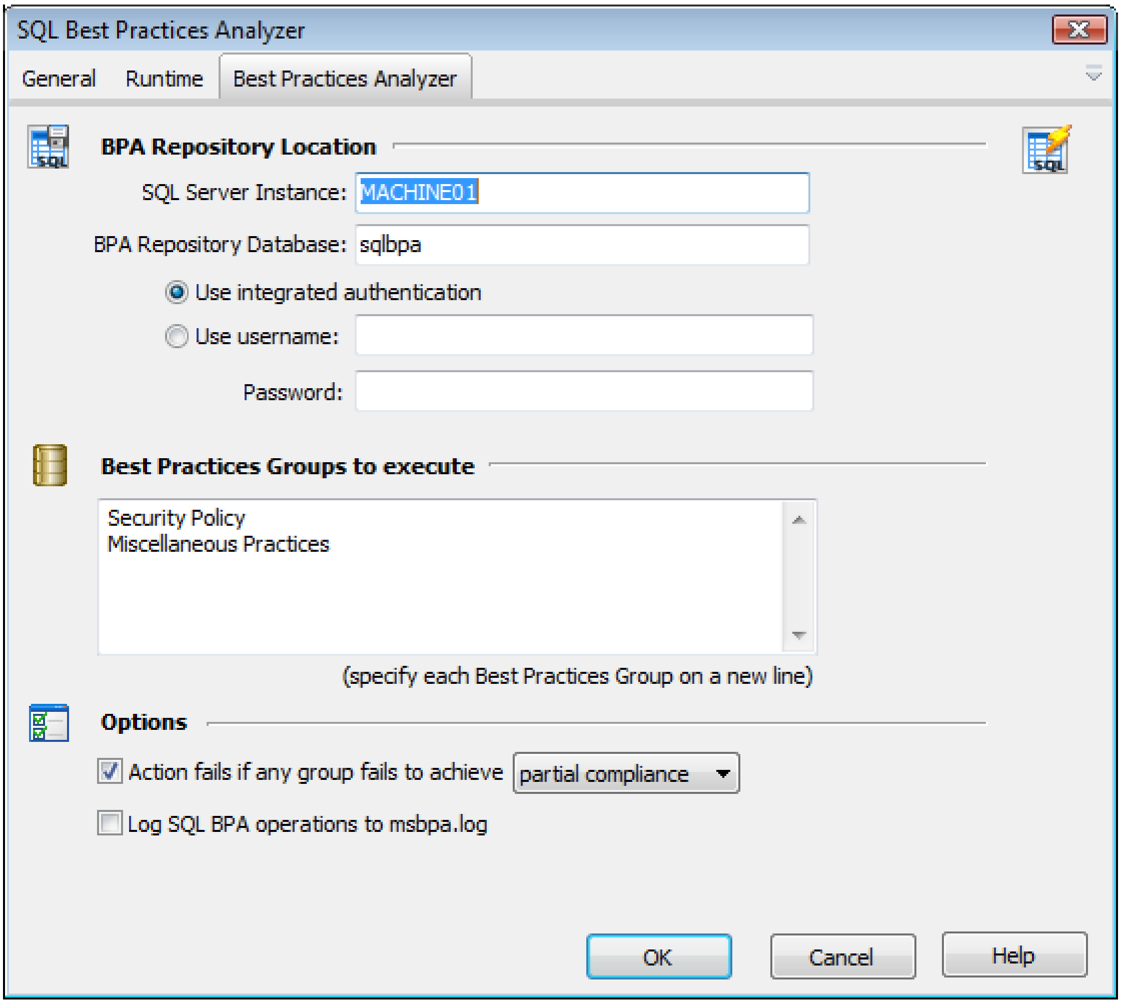
6.6.1.4 SQL Server Best Practices Analyzer

[Automise Professional Edition]

The Microsoft SQL Server Best Practices Analyzer is a free tool, provided by Microsoft, which lets you test for the implementation of many common SQL Best Practices.

The SQL Server Best Practices Analyzer Action allows you to verify those Best Practices from within Automise.

Specify the SQL Server Best Practices Analyzer installation directory from the SQL Server tab in the Options.



SQL Server Instance: MACHINE01

BPA Repository Database: sqlbpa

● Use integrated authentication

○ Use username:

Password:

Security Policy
Miscellaneous Practices

(specify each Best Practices Group on a new line)

☑ Action fails if any group fails to achieve partial compliance

□ Log SQL BPA operations to msbpa.log

BPA Repository Location

The BPA Repository is a database which contains your Best Practices Groups (created with the Microsoft SQL Best Practices Analyzer application.) Specify the SQL Server Instance, database name, and authentication options for the Repository database.

Best Practices Groups to execute

Specify the Best Practices Groups you wish to check against. Again, these are defined inside the Microsoft SQL Best Practices Analyzer application. If you are specifying more than one Best Practices Group, specify each on a new line. Note that at present you cannot execute Best Practice Groups associated with SQL Servers which require SQL authentication (Windows authentication only.) This is a current limitation of the Microsoft command line tool.

Action fails if any groups fails to...

The action can be set to fail if any group fails to pass compliance or (optionally) partial compliance.

Log SQL BPA Operations to msbpa.log

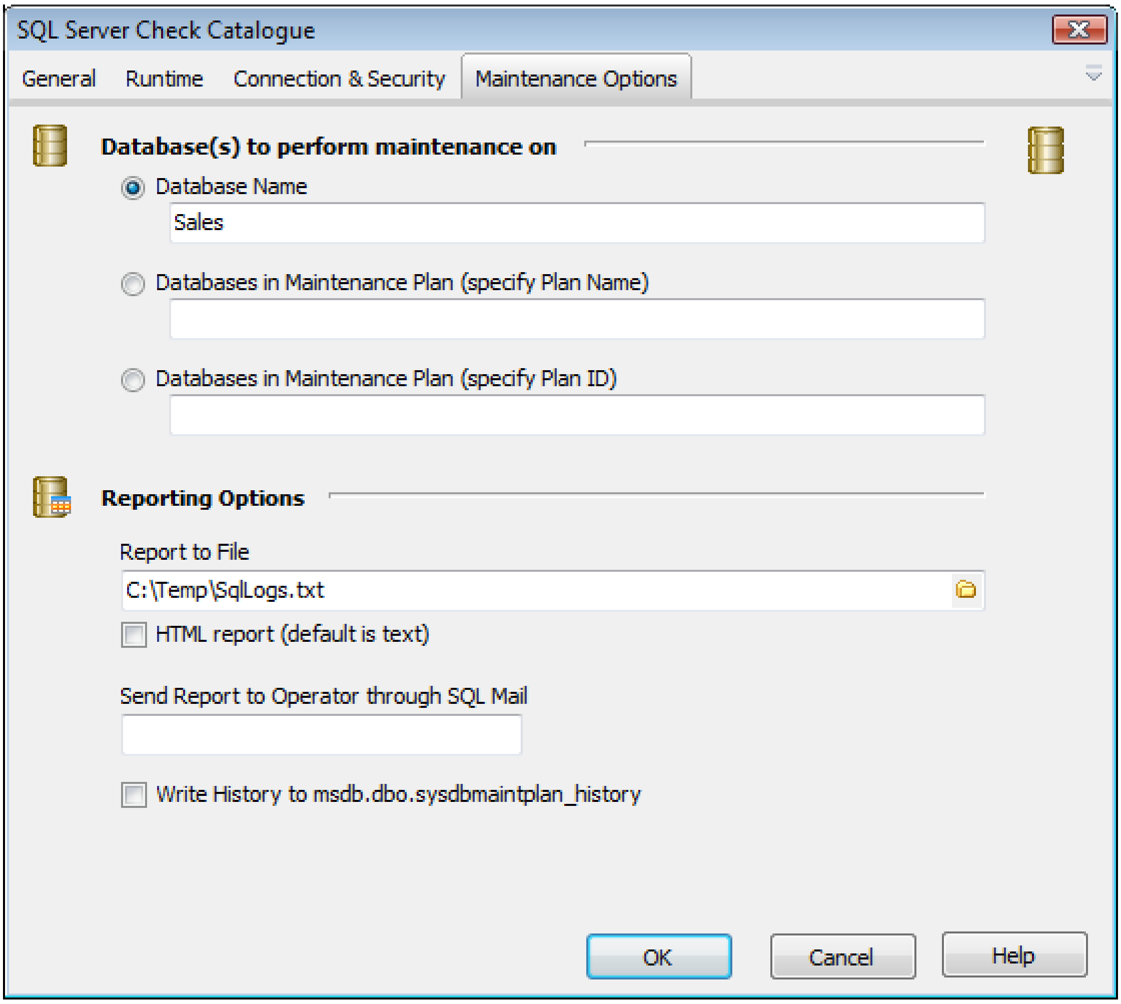
If this option is checked, the analyzer will log its behaviour to msbpa.log inside the 'log' directory of the Best Practices Analyzer directory. Note that each Best Practices Group also generates a log file in this directory, regardless of whether or not this option is checked.

To set the location of the Best Practises Analyser installation directory see SQL Server Options

6.6.1.5 SQL Server Check Catalogue

[Automise Professional Edition]

The Check Catalogue action checks for consistency in and between system tables in the specified database.
This action checks that every data type in syscolumns has a matching entry in systypes and that every table and view in sysobjects has at least one column in syscolumns.



SQL Server Check Catalogue

Tabs: General | Runtime | Connection & Security | Maintenance Options

Database(s) to perform maintenance on

● Database Name: Sales

○ Databases in Maintenance Plan (specify Plan Name):

○ Databases in Maintenance Plan (specify Plan ID):

Reporting Options

Report to File: C:\Temp\SqlLogs.txt

□ HTML report (default is text)

Send Report to Operator through SQL Mail:

□ Write History to msdb.dbo.sysdbmaintplan_history

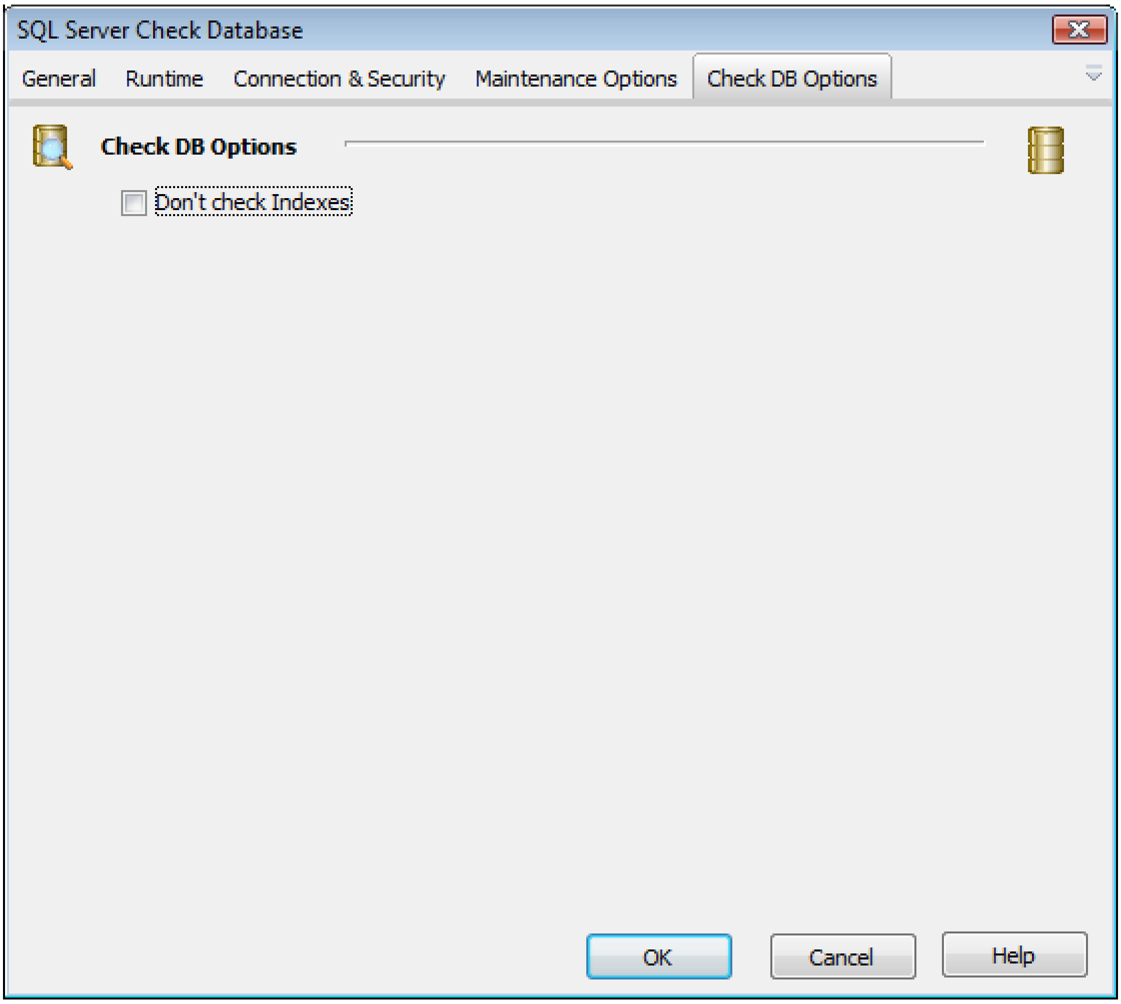
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.6.1.6 SQL Server Check Database

[Automise Professional Edition]

The Check Database action will Check allocation and structural integrity of Database objects.
The "Don't Check Indexes" option specifies that nonclustered indexes for nonsystem tables should not be checked. This decreases the overall execution time because it does not check nonclustered indexes for user-defined tables. "Don't Check Indexes" has no effect on system tables, because system table indexes are always checked.



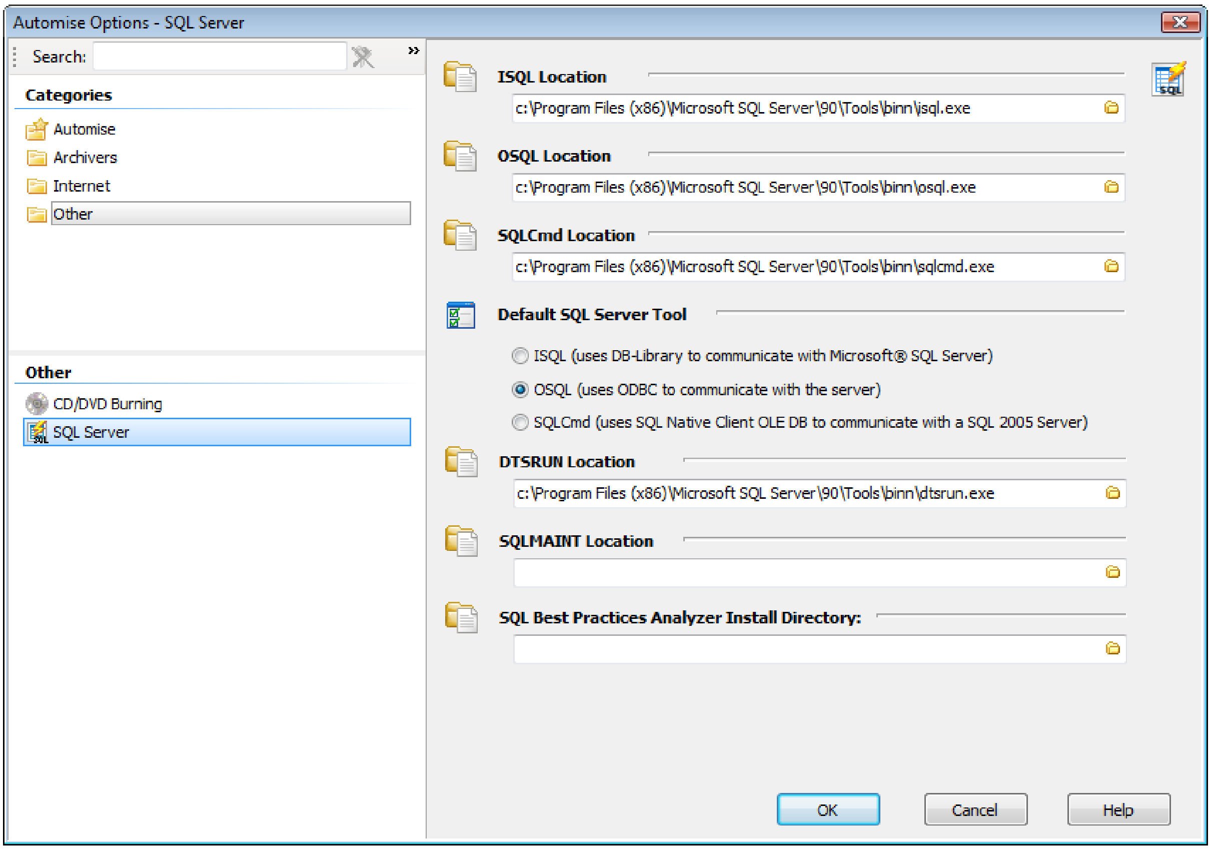
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.6.1.7 SQL Server Options

[Automise Professional Edition]

The SQL Server Options allow you to set the various paths to the SQL Server tools.
Access the options page via Tools Menu | Options | Other Category | SQL Server



Automise Options - SQL Server

Search:

Categories

- Automise
- Archivers
- Internet
- Other

Other

- CD/DVD Burning
- SQL Server

ISQL Location: c:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\isql.exe

OSQL Location: c:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\osql.exe

SQLCmd Location: c:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\sqlcmd.exe

Default SQL Server Tool

○ ISQL (uses DB-Library to communicate with Microsoft® SQL Server)

● OSQL (uses ODBC to communicate with the server)

○ SQLCmd (uses SQL Native Client OLE DB to communicate with a SQL 2005 Server)

DTSRUN Location: c:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\dtsrun.exe

SQLMAINT Location:

SQL Best Practices Analyzer Install Directory:

ISQL Location - the location of isql.exe

OSQL Location - the location of osql.exe

SQLCmd Location - the location of sqlcmd.exe

Default SQL Server Tool - choose the default SQL Server tool for the MSSQL Execute SQL action.

DTSRUN Location - the location of dtsrun.exe

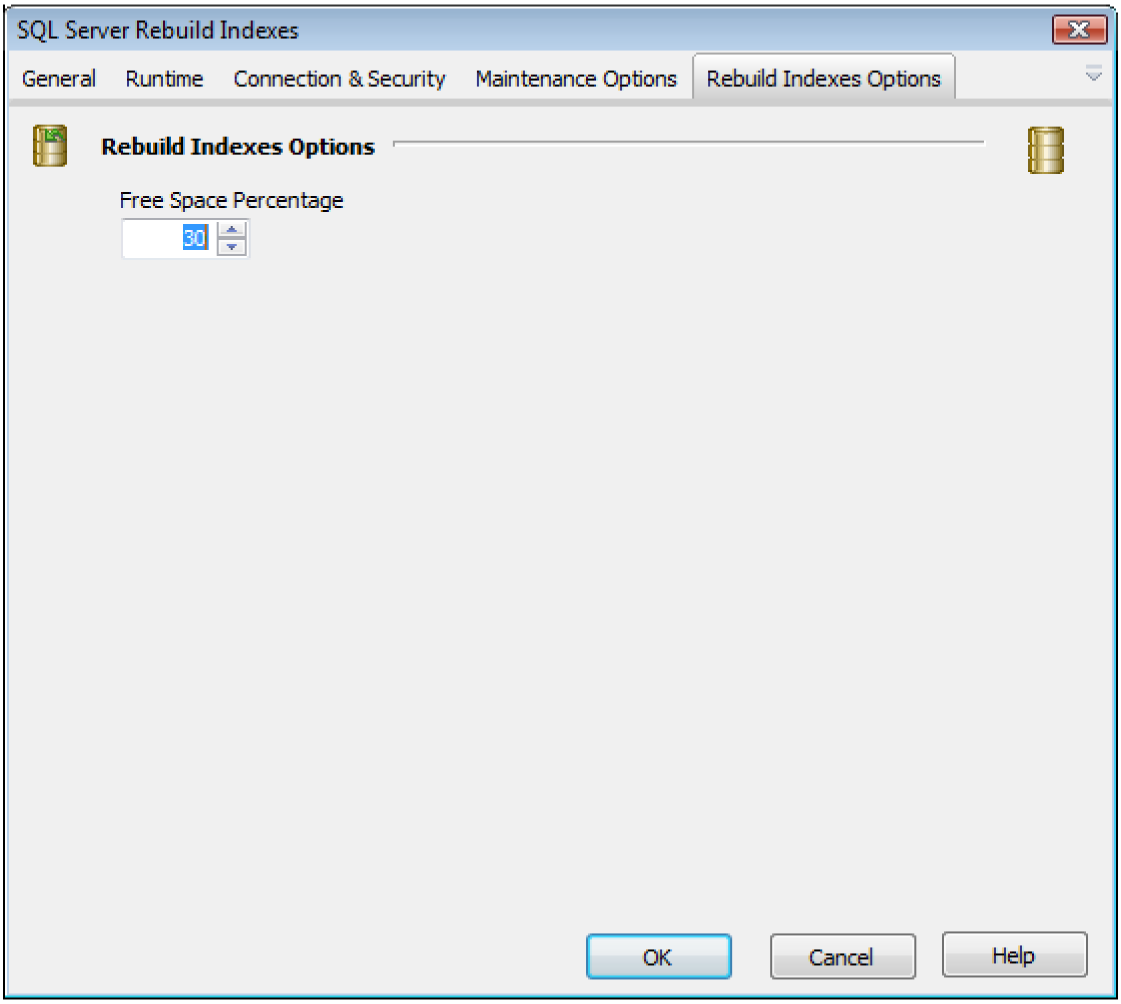
SQLMAINT Location - the location of sqlmaint.exe

SQL Best Practices Analyzer Install Directory - the installation directory of the SQL Best Practices Analyser

6.6.1.8 SQL Server Rebuild Indexes

[Automise Professional Edition]

The SQL Server Rebuild Indexes action specifies that indexes on tables in the target database should be rebuilt by using the "free space" percent value as the inverse of the fill factor. For example, if free space percentage is 30, then the fill factor used is 70. If a free space percentage value of 100 is specified, then the indexes are rebuilt with the original fill factor value.



Rebuild Indexes Options

Free Space Percentage: 30

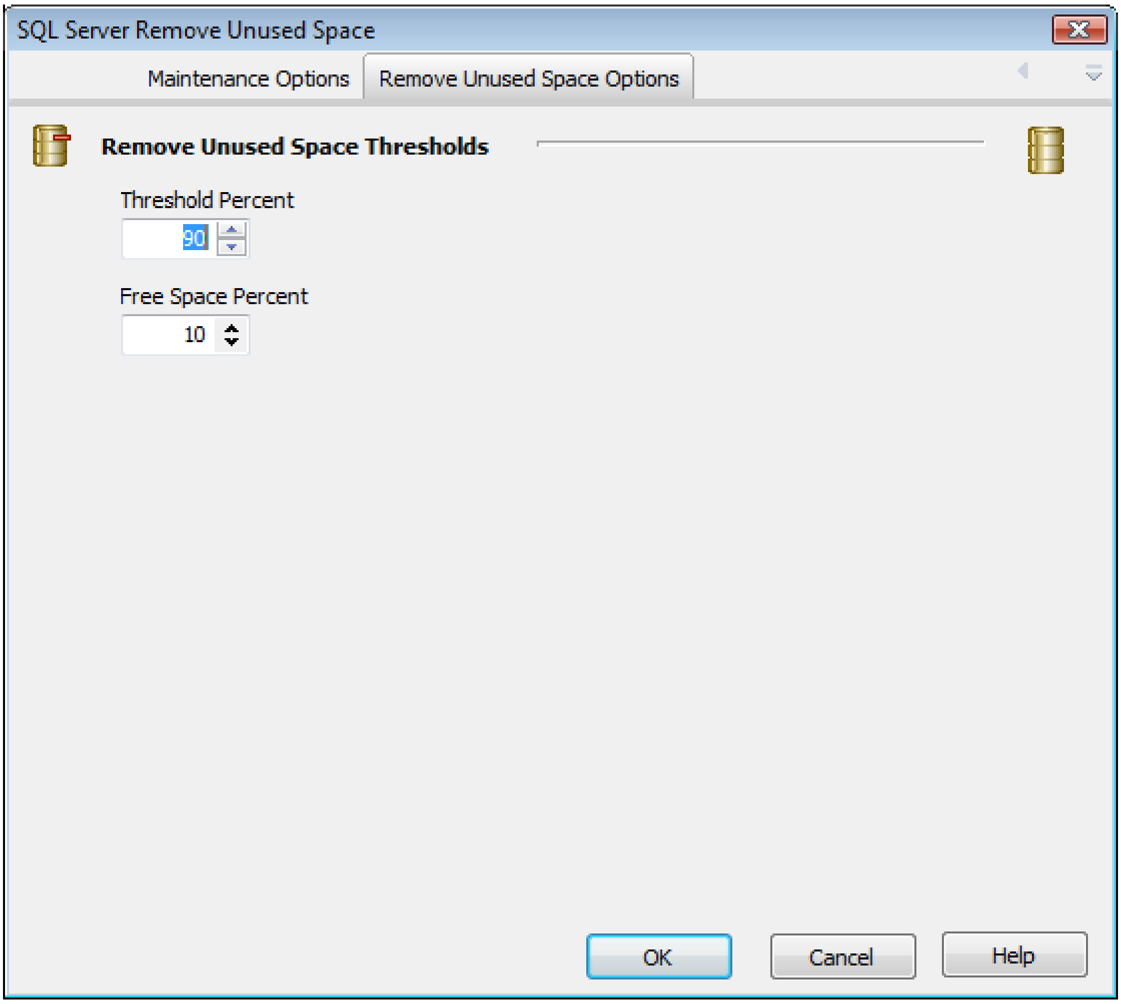
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.6.1.9 SQL Server Remove Unused Space

[Automise Professional Edition]

This action will ask SQL Server to remove unused space from the specified database. This option is only useful for databases that are defined to grow automatically. Threshold_percent specifies in megabytes the size that the database must reach before sqlmaint attempts to remove unused data space. If the database is smaller than the threshold_percent, no action is taken. Free_percent specifies how much unused space must remain in the database, specified as a percentage of the final size of the database. For example, if a 200-MB database contains 100 MB of data, specifying 10 for free_percent results in the final database size being 110 MB. Note that a database will not be expanded if it is smaller than free_percent plus the amount of data in the database. For example, if a 108-MB database has 100 MB of data, specifying 10 for free_percent will not expand the database to 110 MB; it will remain at 108 MB.



SQL Server Remove Unused Space
Maintenance Options
Remove Unused Space Options
Remove Unused Space Thresholds
OK
Cancel
Help

Threshold Percent: 90

Free Space Percent: 10

Specify the threshold percent and free space percent (see above).

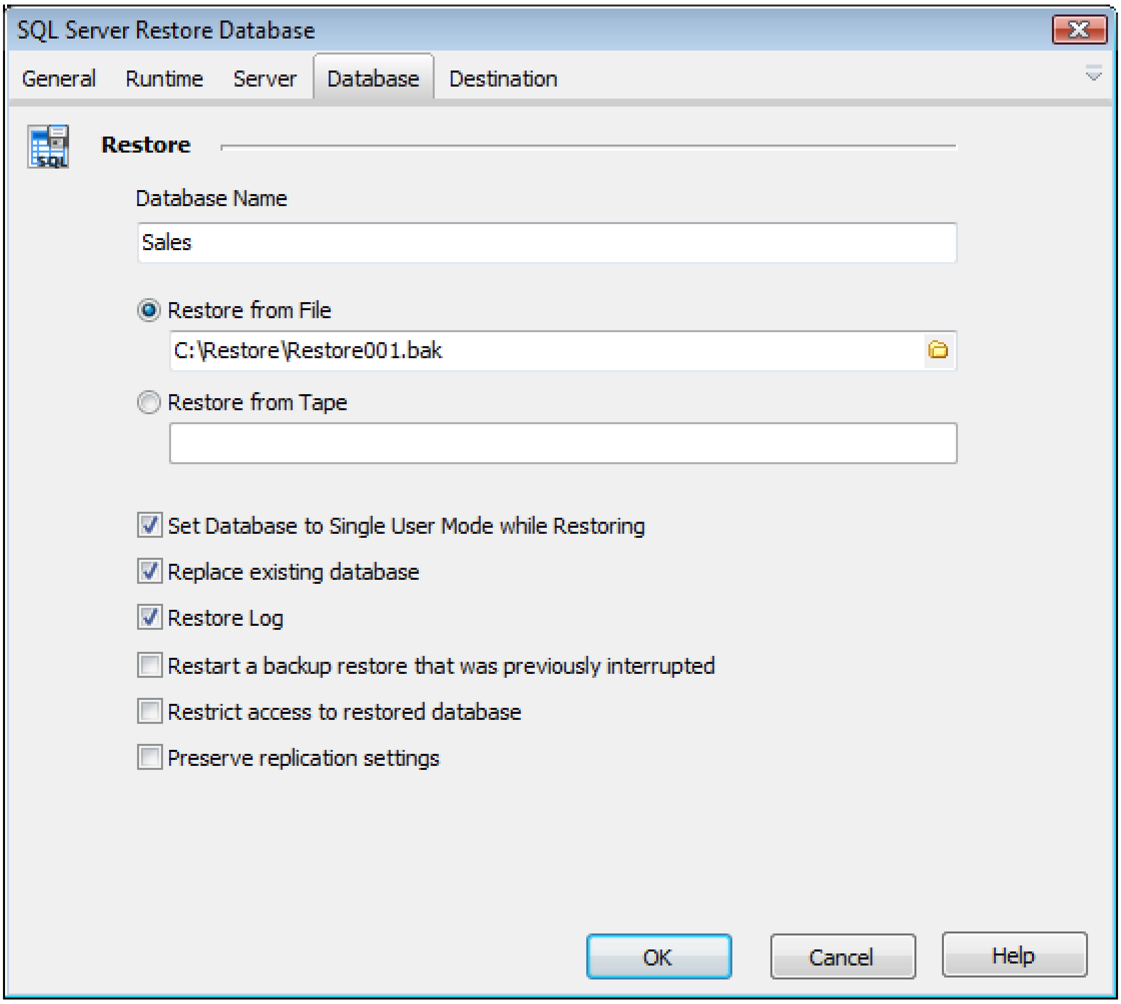
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.6.1.10 SQL Server Restore Database

[Automise Professional Edition]

The SQL Server Restore Database action enables you to restore a SQL database from a disk or tape drive.



SQL Server Restore Database

Restore

Database Name

Sales

● Restore from File

C:\Restore\Restore001.bak

○ Restore from Tape

☑ Set Database to Single User Mode while Restoring

☑ Replace existing database

☑ Restore Log

□ Restart a backup restore that was previously interrupted

□ Restrict access to restored database

□ Preserve replication settings

Database Name
The name of the database to restore to.

Restore From File
The path of the backup file to be restored. The path must be accessible by SQL Server.

Restore From Tape
The name of the tape drive to restore the database from, i.e. '\\.\TAPE0'.

Set Database to Single User Mode while Restoring
The database will be set to single user mode before the restore procedure starts, then changed back to multi-user once completed.

Replace Existing Database
Forces SQL Server to create the specified database and related files even if a database with an existing name already exists.

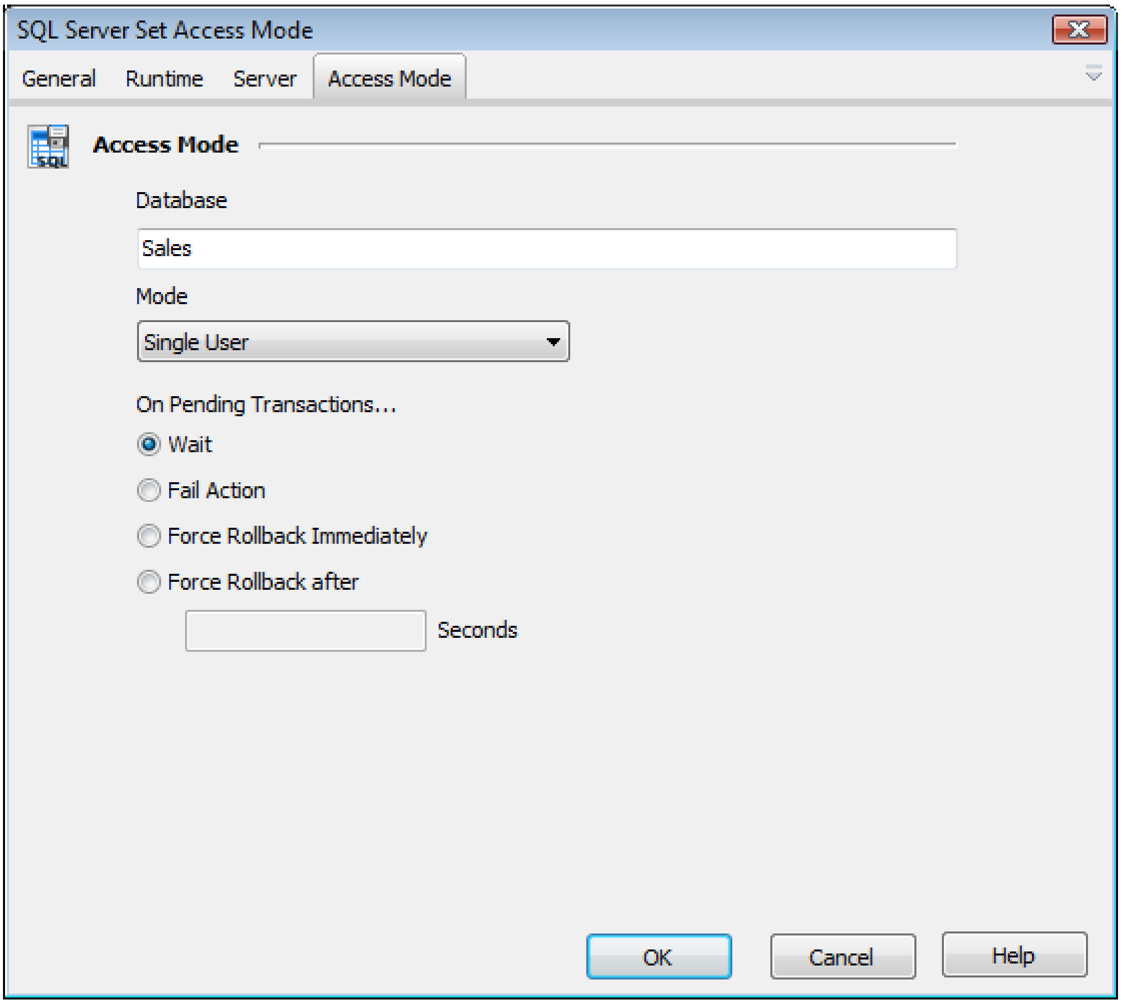
Restart a backup restore that was previously interrupted
If a previous restore attempt was interrupted, SQL Server will restore from the point where the last attempt was interrupted.

Restrict access to restored database
Restricts access to the newly restored database to members of the *db_owner, dbcreator* and *sysadmin* roles.

6.6.1.11 SQL Server Set Access Mode

[Automise Professional Edition]

The SQL Server Set Access Mode action enables you to change the access mode of a database.



Access Mode

Database: Sales

Mode: Single User

On Pending Transactions...

● Wait

○ Fail Action

○ Force Rollback Immediately

○ Force Rollback after

Seconds

Database
The database which to change the access mode.

Mode
The mode to set the database to.

Single User - Single user mode allows only a single user to be connected to the database any one time.
Restricted User - Restricted user mode allows only members of db_owner, dbcreator and sysadmin to be connected to the database, but does not limit the number of connections.
Multi User - Multi user mode allows all users with the appropriate permissions to connect to the database.

On Pending Transactions...
Determines how to react to pending transactions when the mode is being set to Single or

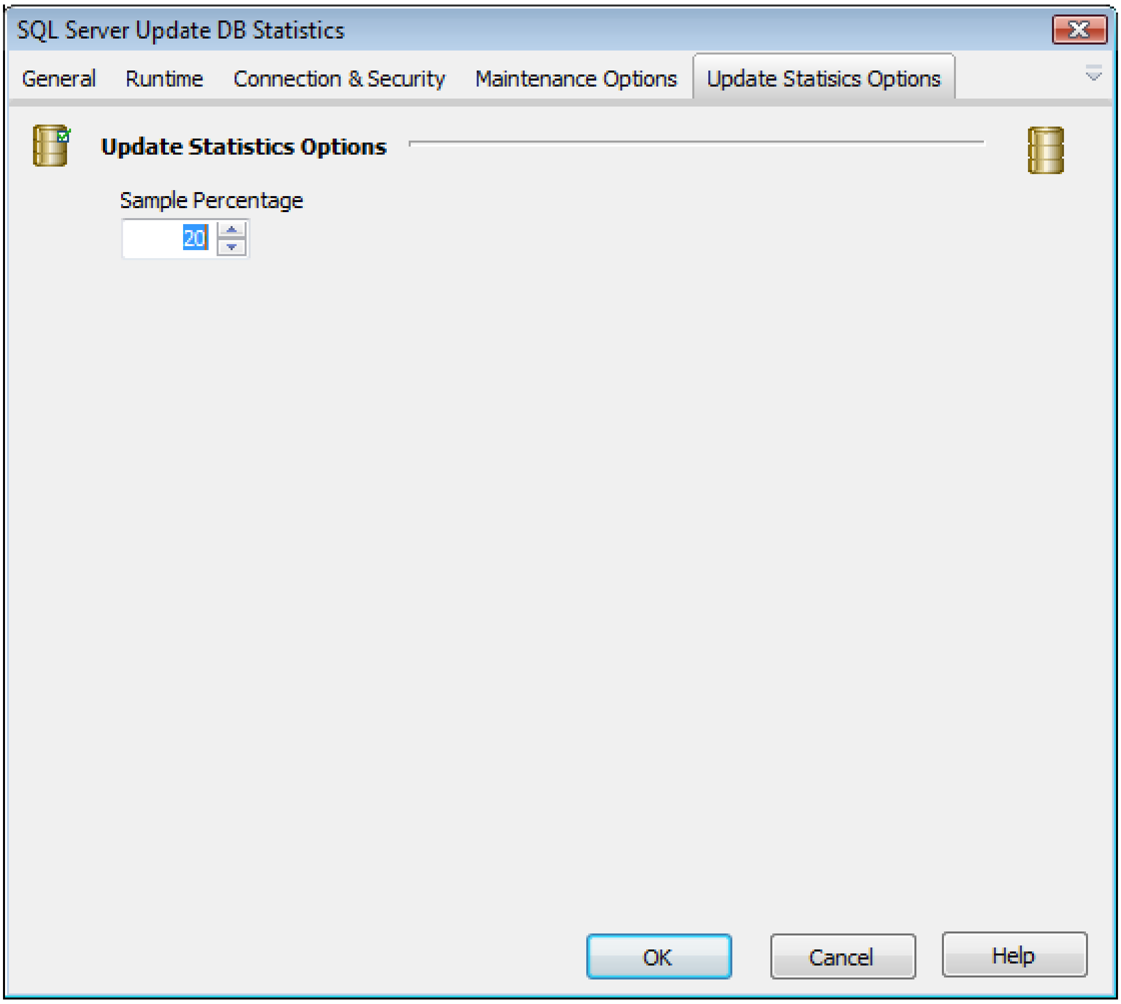
Restricted.

Wait - The action will wait indefinitely until all transactions have been committed or rolled back.
Fail Action - The action will fail if any transactions are pending.
Force Rollback Immediately - Any transactions that are pending will be rolled back immediately.
Force Rollback after - Any transactions that are pending will be rolled back after the specified number of seconds.

6.6.1.12 SQL Server Update DB Statistics

[Automise Professional Edition]

The Update DB Statistics action updates information about the distribution of key values for one or more statistics groups (collections) in the specified database.



Sample Percentage: 20

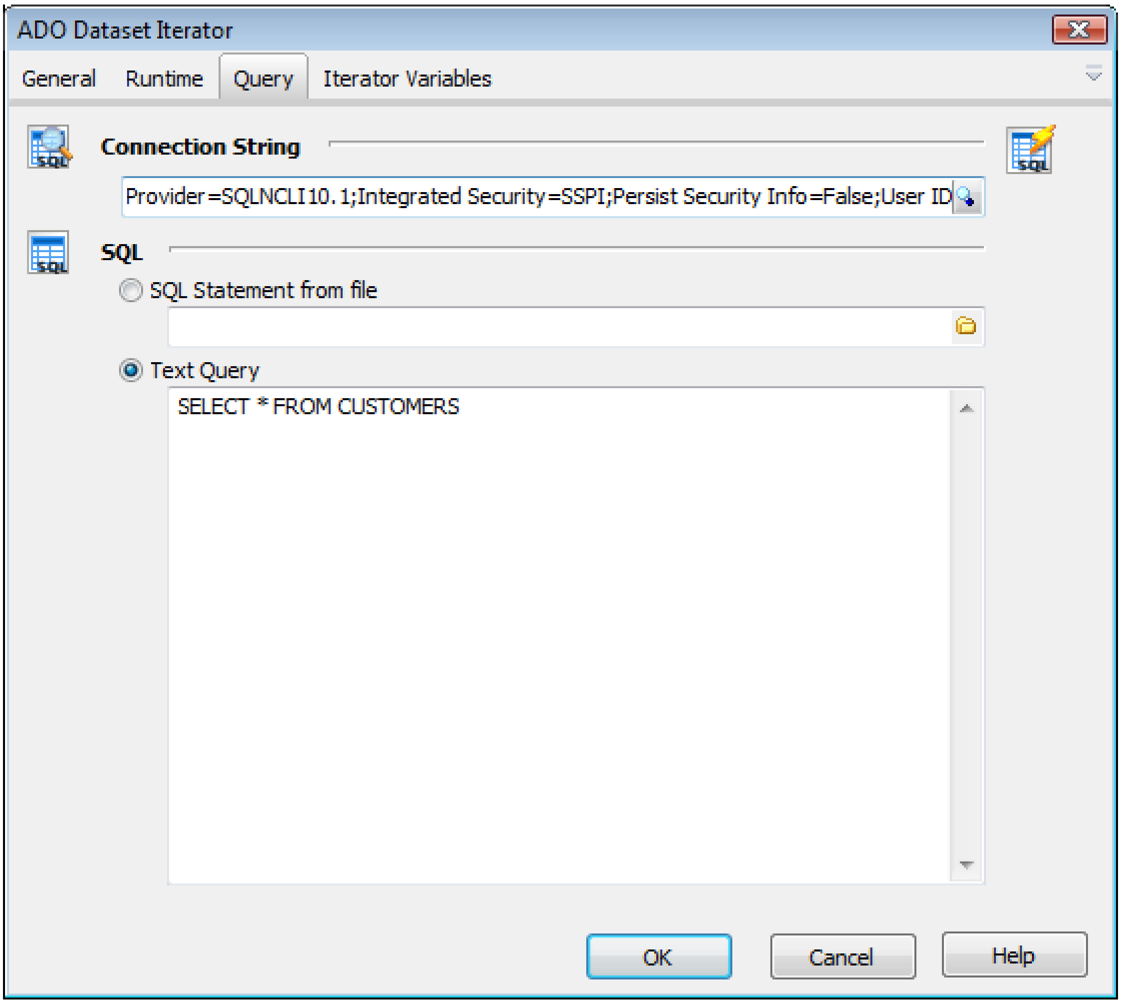
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.6.2 ADO Dataset Iterator

The ADO Dataset Iterator action allows you to perform a group of actions for each row returned from a SQL query on a database using ADO.

For more information on Iterator style actions, see Iterators.



Provider=SQLNCLI10.1;Integrated Security=SSPI;Persist Security Info=False;User ID

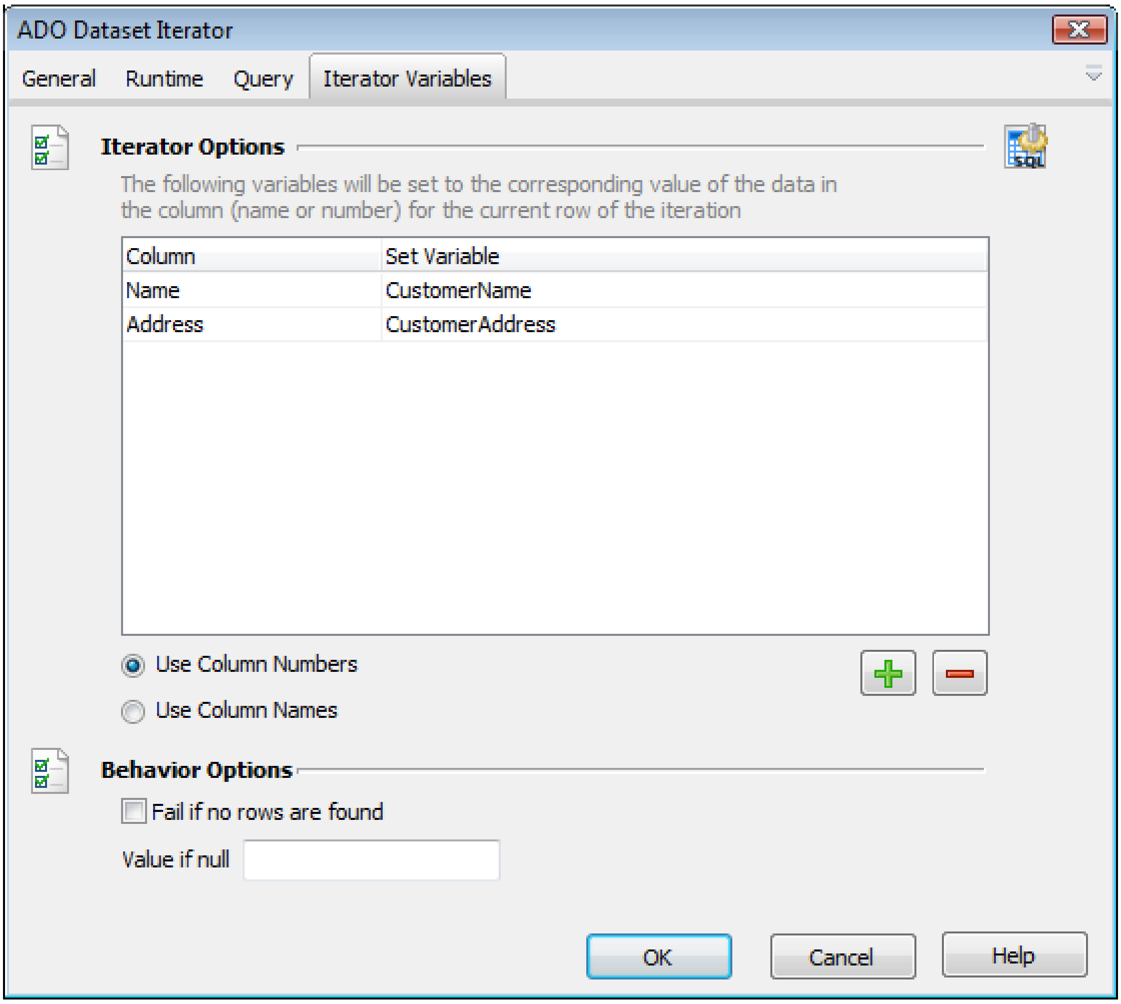
○ SQL Statement from file

● Text Query

```
SELECT * FROM CUSTOMERS
```

Connection String - specify a connection string to your ADO data source. You can use the built in connection string builder to create and test your connection string.

SQL - either specify a file containing SQL, or specify the SQL statement in the text field



Iterator Options

The following variables will be set to the corresponding value of the data in the column (name or number) for the current row of the iteration

Column	Set Variable
Name	CustomerName
Address	CustomerAddress

● Use Column Numbers

○ Use Column Names

Behavior Options

□ Fail if no rows are found

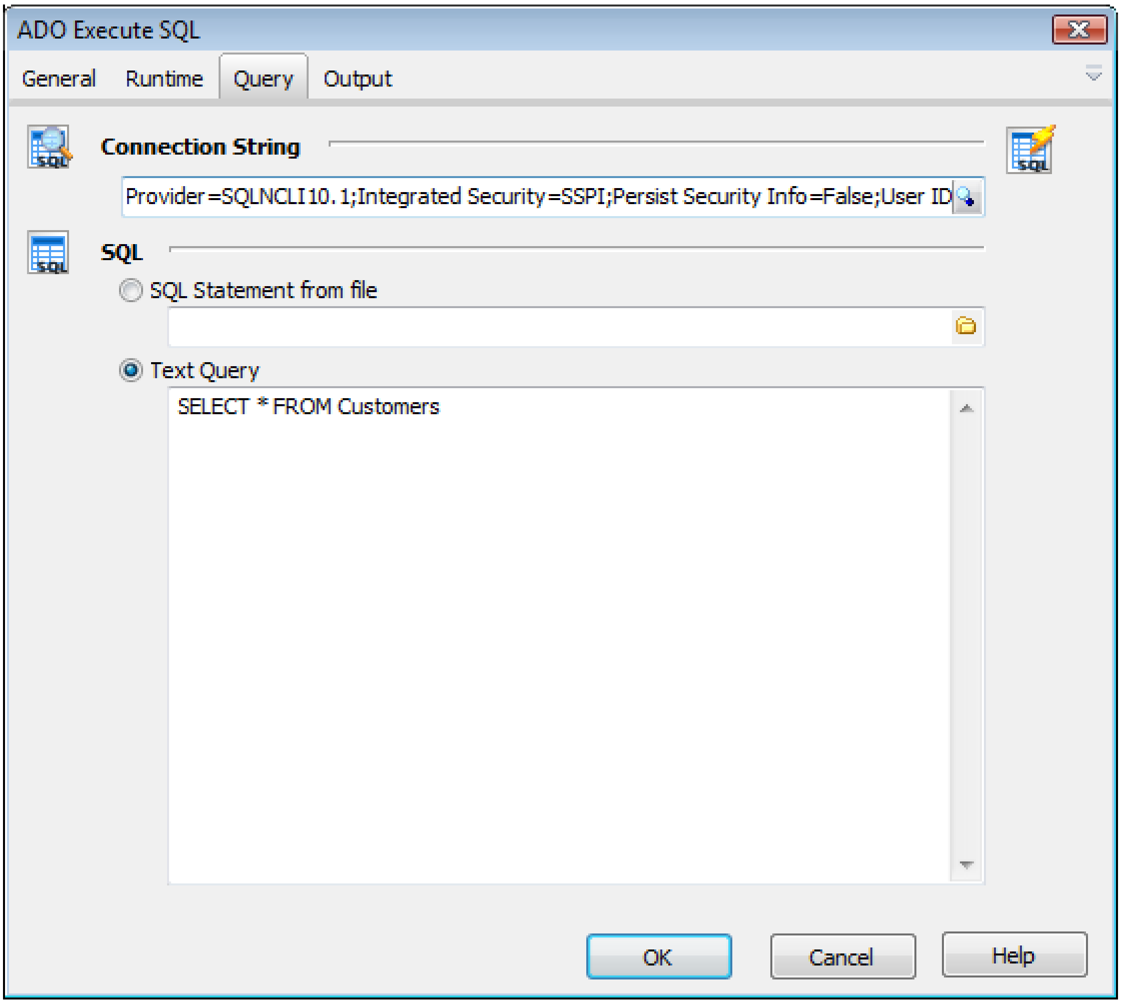
Value if null

Use the grid to set up which column values you want to put in variables for each iteration of the action. The above example shows that the Capability column and Version column values will be placed into the CapabilityName and VersionNum Automise variables. Use the Plus and Minus buttons to add and remove rows in the grid.

Use Column Number / Use Column Names - the Action will either use the column index to find the value, or the column name.

6.6.3 ADO Execute SQL

The ADO Query action enables you to execute a SQL statement against an ADO database connection



Connection String

Provider=SQLNCLI10.1;Integrated Security=SSPI;Persist Security Info=False;User ID

SQL

○ SQL Statement from file

● Text Query

SELECT * FROM Customers

Connection String - specify a connection string to your ADO data source. You can use the built in connection string builder to create and test your connection string.

SQL - either specify a file containing SQL, or specify the SQL statement in the text field

SQL Output - Choose if you want the results output to the FB log, or to and XML file using the ADO XML file format.

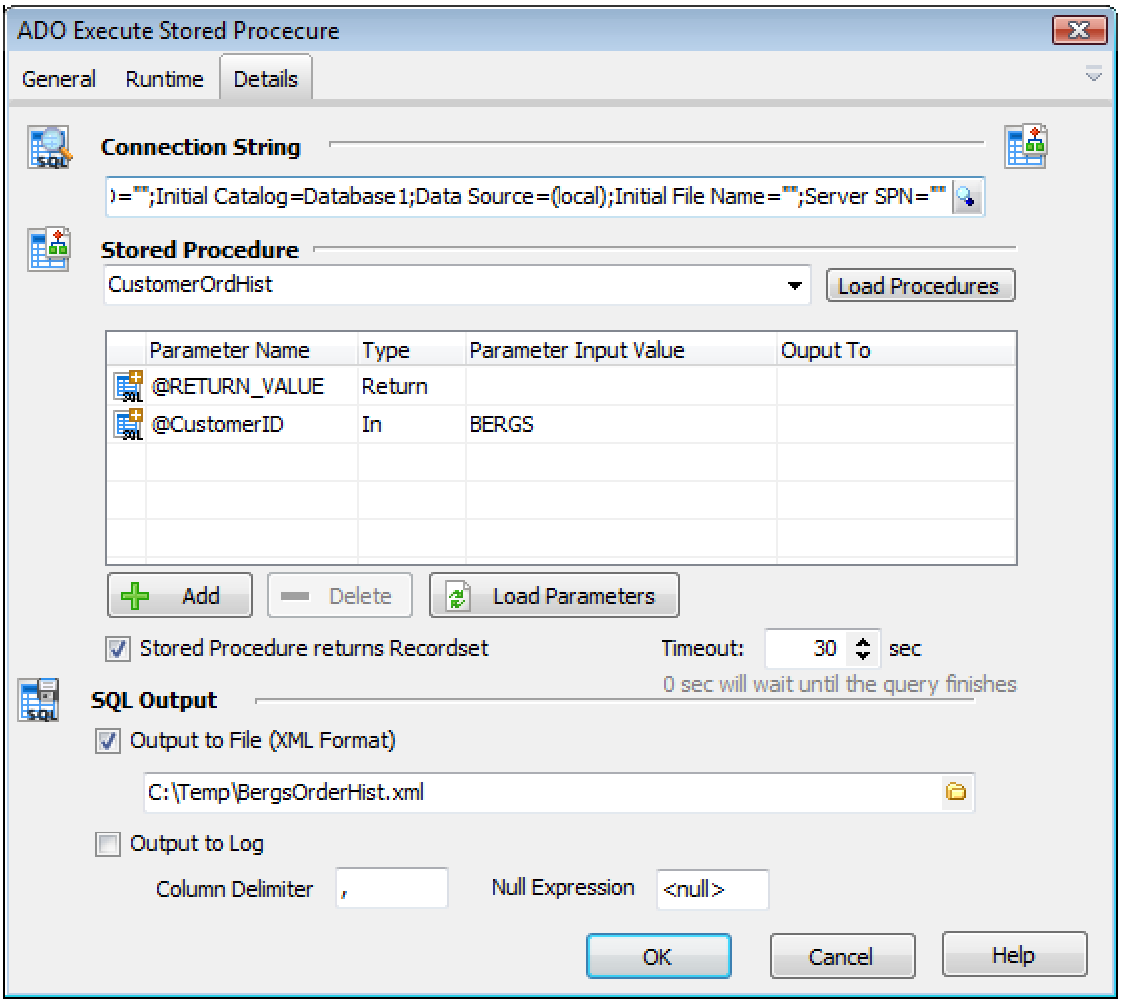
6.6.4 ADO Execute Stored Procedure

The ADO Stored Procedure action enables you to automate the execution of stored procedures using ADO.

The action property pages helps you build your connection string, displays a list of stored procedures in the target database, and will retrieve the parameters for the selected stored procedure. The value of each input parameter can be set, and the out values and return values can be saved to a variable.

It is important to set "Stored Procedure Returns Recordset" correctly, as this controls how the action internally calls the stored procedure.

NOTE: This action is only available in the Professional Edition of Automise



Connection String

)="";Initial Catalog=Database1;Data Source=(local);Initial File Name="";Server SPN=""

Stored Procedure

CustomerOrdHist

Parameter Name	Type	Parameter Input Value	Ouput To
@RETURN_VALUE	Return		
@CustomerID	In	BERGS	

☑ Stored Procedure returns Recordset

Timeout: 30 sec

0 sec will wait until the query finishes

SQL Output

☑ Output to File (XML Format)

C:\Temp\BergsOrderHist.xml

□ Output to Log

Column Delimiter ,

Null Expression <null>

Connection String - specify a connection string to your ADO data source. You can use the built in connection string builder to create and test your connection string.

Stored Procedure - once the connection string has been specified, you can click the "Refresh" button and a list of available stored procedures will be listed. Once you select a stored procedure, the parameters will be listed. If the stored procedure is yet to be defined then you can manually enter the stored procedure name.

Parameters - The Name and Type of the parameters should retrieved automatically, if not then you need set them to the correct values. For any IN or INOUT parameters you should set a value (either a hardcoded value, or use an FB Variable). For any Return, OUT, or INOUT parameters you can optionally set an FB Variable for the value to be saved to when the stored procedure executes.

Use the **Add** and **Delete** buttons to manually define the parameters if the stored procedure is yet to be defined.

Stored Procedure returns Recordset - It is very important that you set the "Stored Procedure returns Recordset" option correctly, as this determines how the stored procedure is called internally.

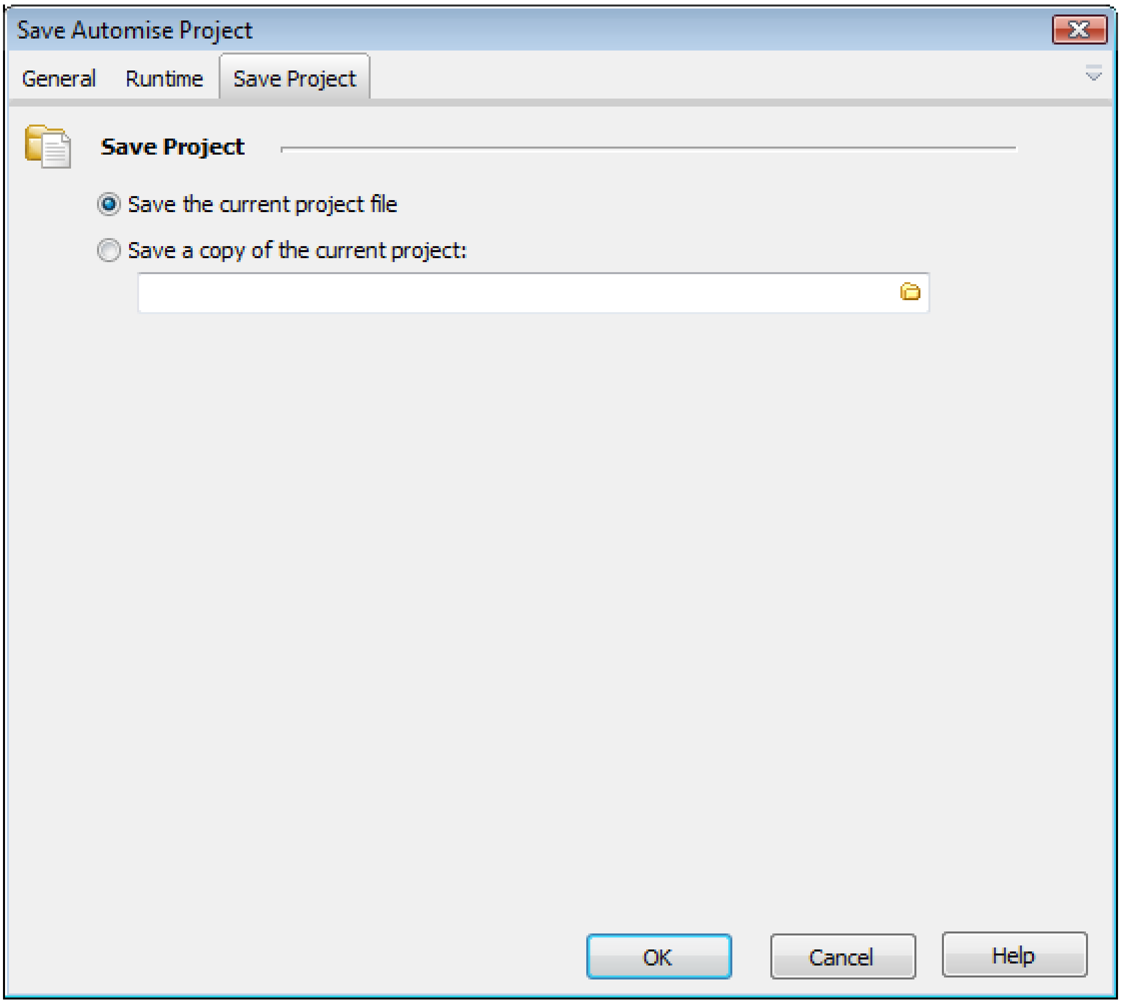
SQL Output - You can output a result recordset to the FB log, and/or to and XML file using the ADO XML file format.

6.7 Automise IDE

6.7.1 Save Automise Project Action

The **Save Automise Project** Action allows you to save the current Automise project from inside your project.

Reasons you may want to do this include saving changes to action properties, and saving the status of persistent variables.



Save Project

● Save the current project file

○ Save a copy of the current project:

Save the current project file

Saves the current project to the current project file. The equivalent of choosing File -> Save in the Automise IDE.

Save a copy of the current project

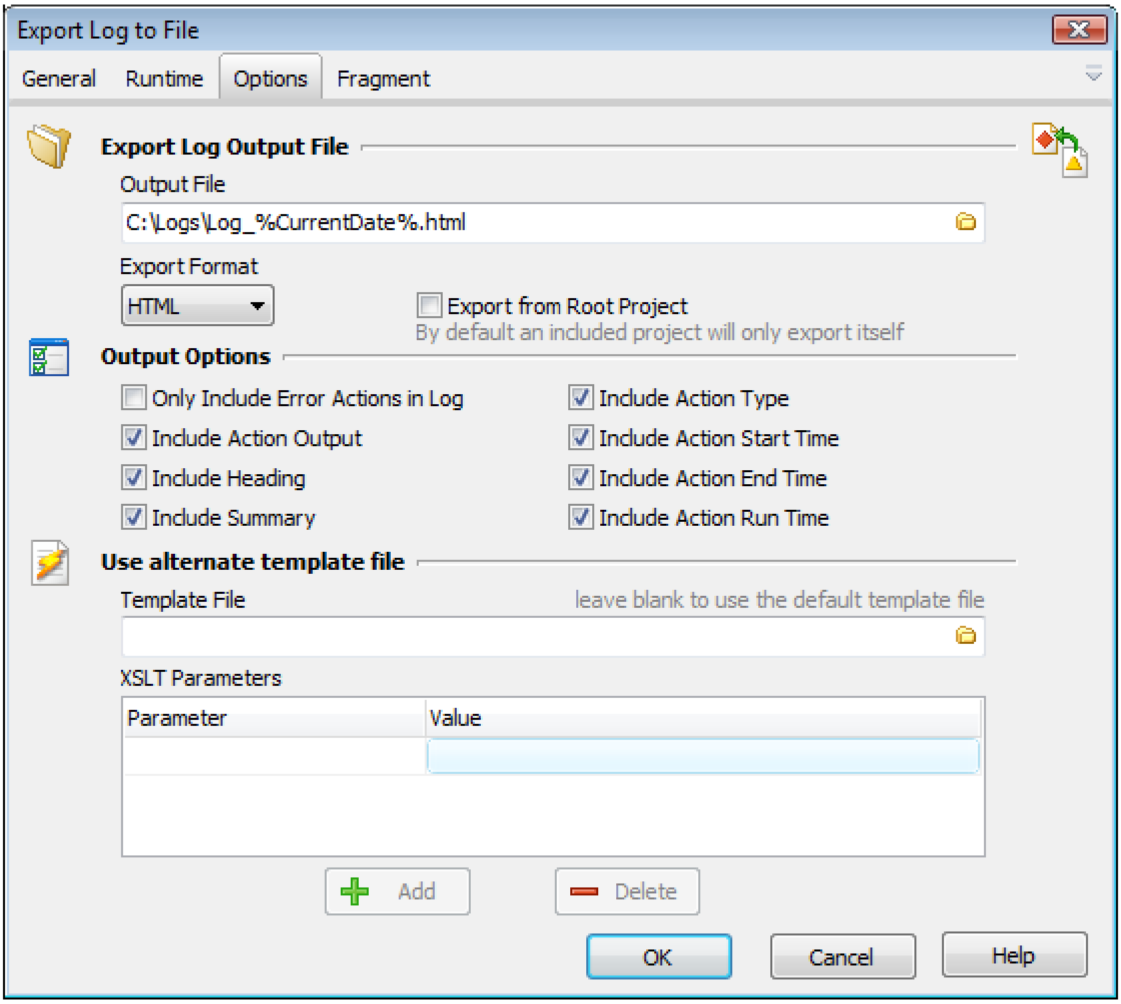
A copy of the current project will be written to the file you specify.

6.7.2 Export Log Action

The **Export Log Action** exports a Automise run to a text based log file.

This enables the log to be archived, sent as an attachment via email, further processed, etc.

You can choose to export the current log, in Text, HTML or XML formats.



Export Log Output File

Output File: C:\Logs\Log_%CurrentDate%.html

Export Format: HTML

□ Export from Root Project
By default an included project will only export itself

Output Options

□ Only Include Error Actions in Log
☑ Include Action Output
☑ Include Heading
☑ Include Summary
☑ Include Action Type
☑ Include Action Start Time
☑ Include Action End Time
☑ Include Action Run Time

Use alternate template file

Template File — leave blank to use the default template file

XSLT Parameters

Parameter	Value

Output File - specify the filename that the log will be written to

Export Format - choose between Text, HTML, or XML. The default template files are as follows:
- HTML - Automise Install Directory\stylesheets\ConvertLogToHTML.xsl
- XML - Automise Install Directory\stylesheets\ConvertLogToXML.xsl
- Text - Automise Install Directory\stylesheets\ConvertLogToText.xsl

Only Include Error Actions in Log - only actions that failed are included in the log

Include Action Output - check this option to include the action messages in the log

Include Heading - check this option to include a heading in the output (not applicable for XML format): Automise Logfile for <project name>

Include Summary - check this option to include a summary in the output (not applicable for XML format).
The summary contains the following information:
- Project File
- Log Title
- Date
- Start Time
- End Time
- Run Time

Include Action Type - check this option to include the action name in the output (not applicable for XML format).

Include Action Start Time - check this option to include the action start time in the output (not applicable for XML format).

Include Action End Time - check this option to include the action start time in the output (not applicable for XML format).

Include Action Run Time - check this option to include the action run time in the output (not applicable for XML format).

Use alternate template file

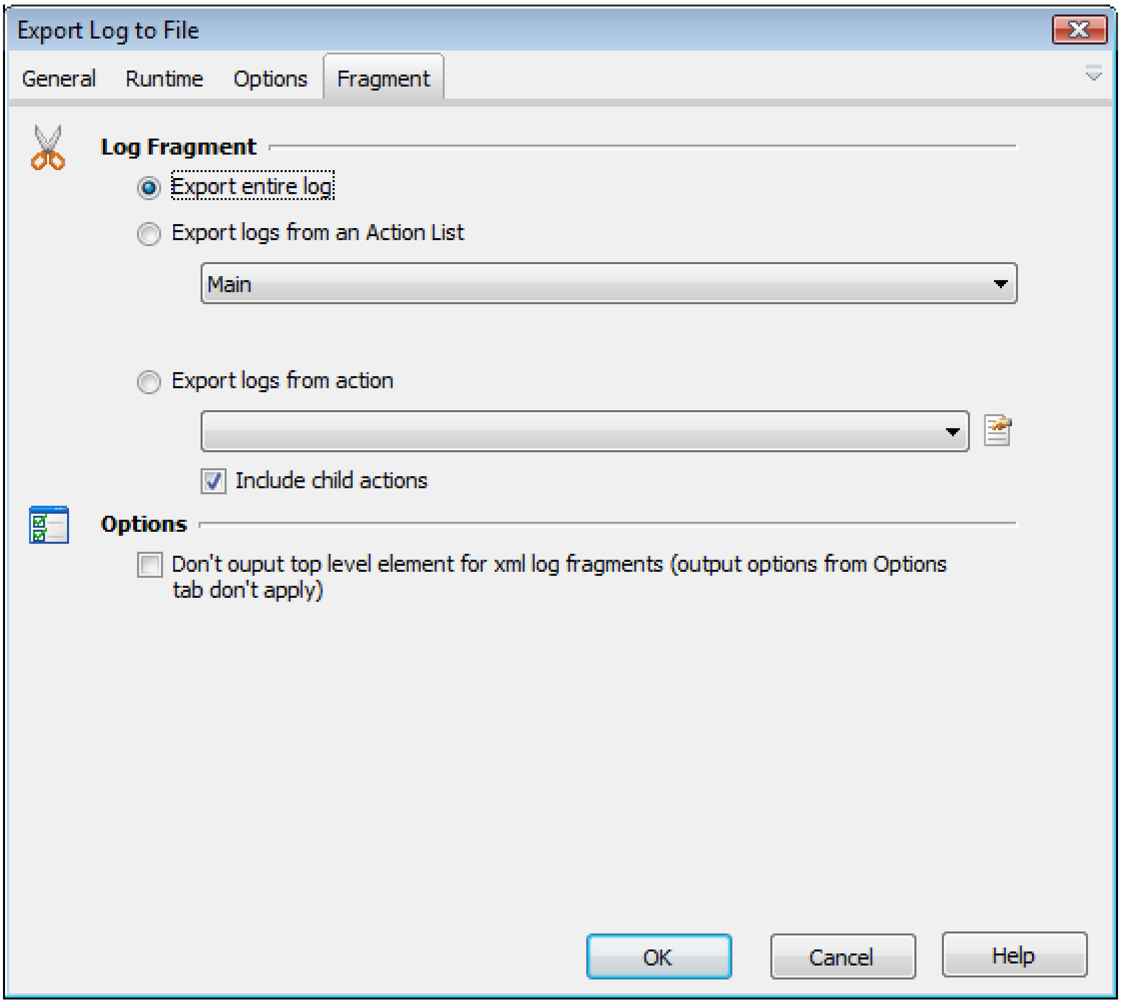
Template File - You can also specify a different XSL Stylesheet template file to alter the output format of the log file. When you leave Template File blank, Automise will choose the appropriate stylesheet depending on the Export Format chosen (see Export Format above).

The "XSLT Parameters" allow you to pass variables to your XSL Stylesheet.

Note that the default options for the log can be set in the Automise options dialog.

Export Fragment

You may wish to only export a section of the log, eg. the log of a particular Action List, or even just the log of a certain action.



Log Fragment

● Export entire log

○ Export logs from an Action List

Main

○ Export logs from action

☑ Include child actions

Options

□ Don't ouput top level element for xml log fragments (output options from Options tab don't apply)

Export entire log - selecting this will export the entire log (up to when this action runs).

Export logs from an Action List - choose this option to export the logs of a particular Action List. Select the action list to export.

Export logs from action - choose this option to export the logs of a particular Action, and then choose the action to export. Actions are listed by description in alphabetical order. If you have two actions with the same name, then they will both be listed (and if you need to choose one, then we recommend you rename them so you can tell them apart).

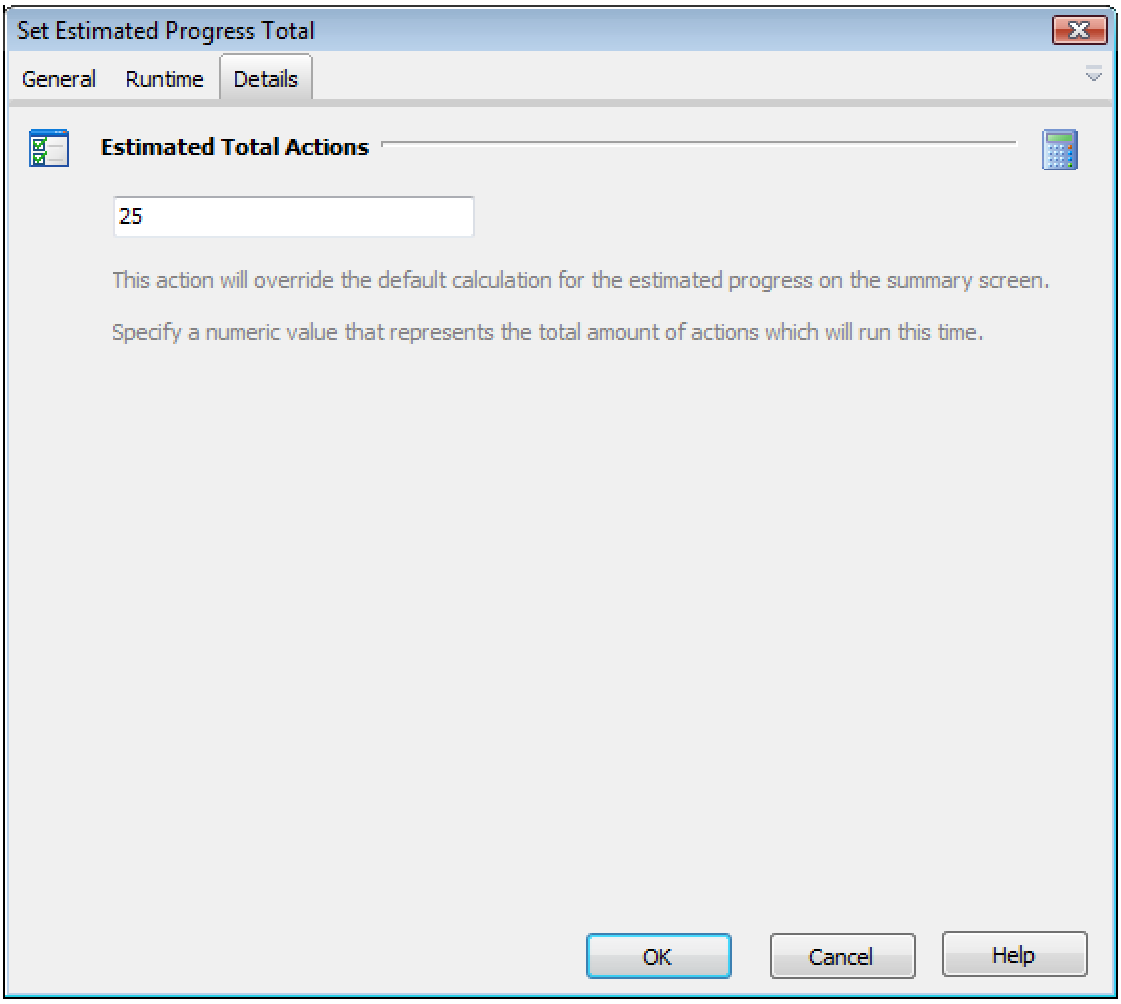
Include child actions - choose this option to include in the logs any child actions of the action chosen. This option will not export the logs of any Action List run as a child action.

Note: for both exporting of a particular Action List and Actions, if that action or action list is run more than once, then all occurrences will will be exported to the log.

6.7.3 Set Estimated Progress Total Action

The Set Estimated Progress Total action allows you to override the total action for the estimated progress status bar. For more information on the estimated progress calculation and display, see EstimatedProgress.

The action takes a single positive numeric value, which overrides the total used for the estimated progress status bar. The field accepts variables, but the variable value must be a positive numeric value.



Estimated Total Actions

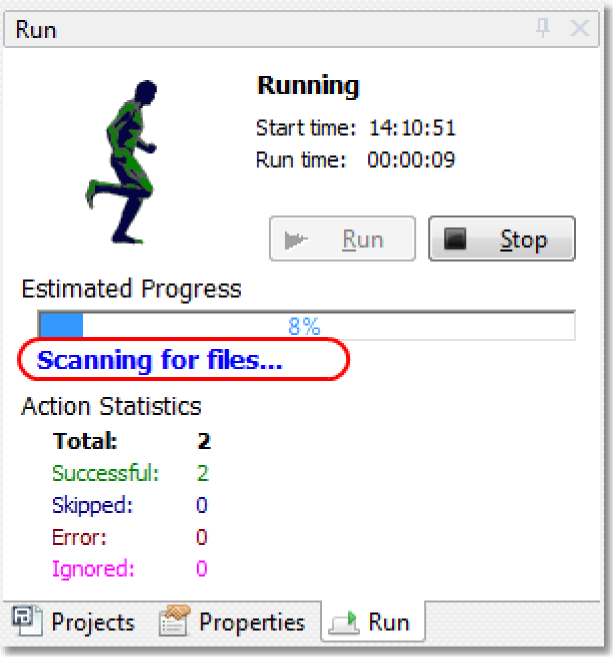
25

This action will override the default calculation for the estimated progress on the summary screen.

Specify a numeric value that represents the total amount of actions which will run this time.

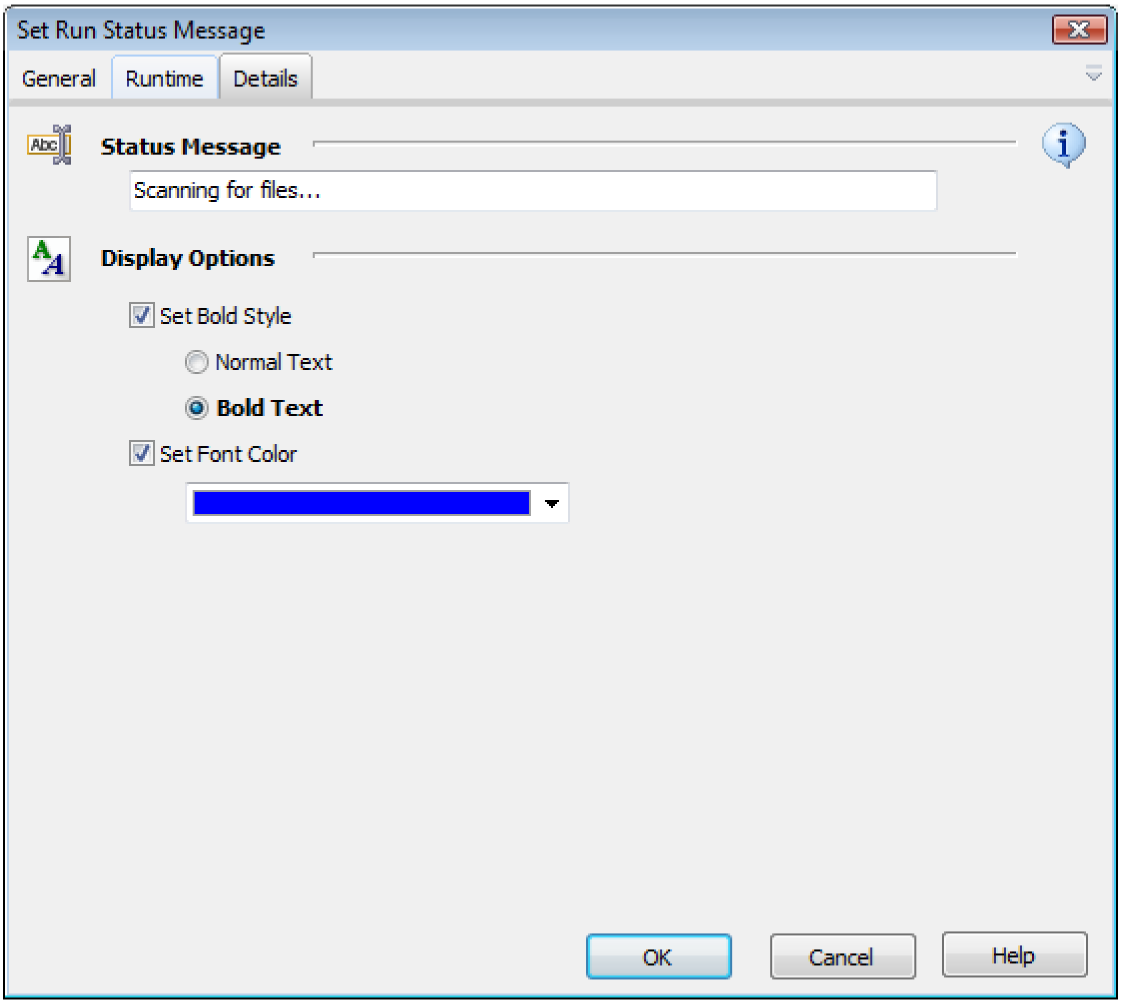
6.7.4 Set Run Status Message Action

The Set Run Status Message action allows you to add a custom message to the project Status screen. This message can be anything you like - so for example it could be "Running Installer" or "Uploading". It's a message that will give you an indication as to what the current progress of the project is.



The above screen shot shows the custom status message "Scanning for files".

The options for Set Run Status Message action are:



Status Message

Scanning for files...

Display Options

☑ Set Bold Style

○ Normal Text

● Bold Text

☑ Set Font Color

Status Message: the message you want to appear below the progress bar. Variable references are expanded, eg. "We're now running project %CurrentProject%".

Set Bold Style: overrides the current bold style.

Normal Text: sets the text to non-bold.

Bold Text: sets the text to bold.

Set Font Color: sets the color of the text (and then set the colour using the colour combo box).

When a new project is started, the status message will be cleared and the font and style set back to default (ie. Normal and Black).

6.8 FileSets

How FileSets work

FileSets allow you to specify a set of patterns to select files, exclusion patterns to exclude certain files, and filters to further limit the files selected. The FileSet can then be used to generate a list of files, which can then be further used in other actions which can either natively use a FileSet object, or can take a list of files (there are many output formats available).

To define a FileSet use the FileSet Define action. This is where you set up the include and exclude patterns, the filters, the root directory, and the default output separator and quoting style. A FileSet must have at least one Include pattern, but does not have a maximum. A FileSet can also have any number of Exclude patterns and filters.

When a FileSet is processed, Automise scans the file system for each include pattern using the include patterns parameters for the search. Each file and directory found is then tested against the exclude patterns, and finally each file is tested against each filter before being added to the list of files in the FileSet. This list of files is then available during the rest of the project. See Accessing FileSets to see how other actions in the project can use the list of files selected.

6.8.1 Accessing FileSets

There are a number of ways to use the list of files found by a FileSet:

1. Use the FileSet To Variable action to output the list of files to a Automise variable in the chosen format
2. Use the FileSet directly in script code to iterate the fileset or generate a string containing the files in the fileset, eg. MyFileSet.Count
3. Use the FileSet directly in fields of Automise actions using the variable syntax, eg. %MyFileSet.File1%

Methods of a FileSet

The methods of a FileSet can be accessed in both script and via the variable syntax in the fields of actions using the following syntax:

- **Count** - returns the amount (integer) of files in the FileList
- **IsEmpty** - returns true or false (boolean) depending if any files are in the FileSet
- **File<index>** - returns the filename (string) specified by the index. The index specified is zero based (ie. first file in fileset is index 0) and must be less than the total amount of files in the FileSet.
- **AllFiles** - returns all the files (string) in the FileSet

If accessing the FileSet using the variable syntax, the following parameters are also available:

:quote=[double|single]

Specifying the quote type overrides the default quote type as specified in the FileSet Define action. Can be appended to both the File and AllFiles methods

:sep=[crlf|cr|lf|comma|space|tab|colon|semicolon]

Specifying the separator type overrides the default separator character as specified in the FileSet Define action. Can be appended to the AllFiles method.

Examples:

To retrieve the count of files in the fileset:
Variable Syntax: %MyFileSet.Count%

Script: MyFileSet.Count

To retrieve file 10 in the FileSet:
Variable Syntax: %MyFileSet.File10%
Script: MyFileSet.File10

To retrieve file 5 in the FileSet overriding the default quoting to use double quotes:
Variable Syntax: %MyFileSet.File5:quote=double%

```
Script:    MyFileSet.OutputQuoting = qtDouble
           MyFileSet.File5
```

To retrieve file 5 in the FileSet overriding the default separator:
Variable Syntax: %MyFileSet.File5:sep=comma%

```
Script:    MyFileSet.OutputSeparator = spComma
           MyFileSet.File5
```

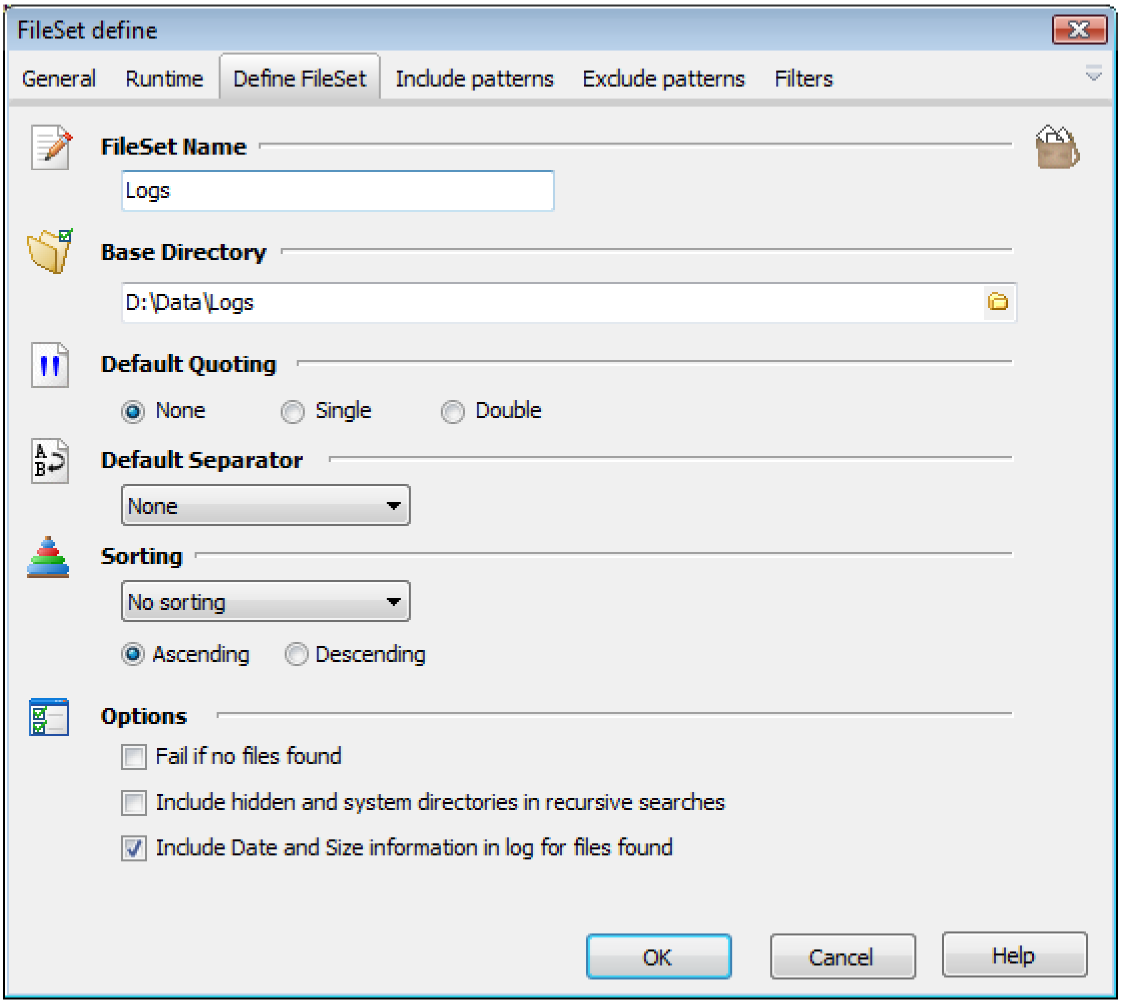
To retrieve all files in the FileSet:
Variable Syntax: %MyFileSet.AllFiles%
Script: MyFileSet.AllFiles

To retrieve all files in the FileSet overriding the default separator and quote type:
Variable Syntax: %MyFileSet.File5:quote=single:sep=comma%

```
Script:    MyFileSet.OutputQuoting = qtSingle
           MyFileSet.OutputSeparator = spComma
           MyFileSet.AllFiles
```

6.8.2 FileSet Define

The FileSet Define action is used to create a new FileSet.



FileSet Name: Logs

Base Directory: D:\Data\Logs

Default Quoting: ● None ○ Single ○ Double

Default Separator: None

Sorting: No sorting

● Ascending ○ Descending

Options

□ Fail if no files found

□ Include hidden and system directories in recursive searches

☑ Include Date and Size information in log for files found

The FileSet Define action has the following properties:

FileSet Name - the name the FileSet will be referred to in other actions, script, and action fields

Base Directory - the root directory where the search for files will be based

Default Quoting - the default quoting style

Default Separator - the default separator style used when outputting more than one file

Sorting - choose how the files found will be sorted:
- No sorting - the files in the FileSet are returned in the order they were found (fastest)
- File name (natural) - sort order similar to how Windows Explorer sorts files (not case sensitive and numbers are treated as such)
- File name (ASCII) - basic ASCII sorting of file names
- File size - sorted according to the size of the files
- File date - sorted according to last modified date/time

Options

Fail if no files found - the action will fail if no files are added to the FileSet
Include hidden and system directories - by default these directories aren't searched in recursive searches
Include Date and Size information - when the action runs it reports all files found,

this option includes the file size and modified date in the output

After setting up the options for the FileSet you need to add Include Patterns, Exclude Patters and Filters:

Include Patterns - a list of patterns which will be used to find files
Exclude Patterns - a list of patterns used to exclude files
Filters - a list of filters used to refine the selection of files based on different criteria

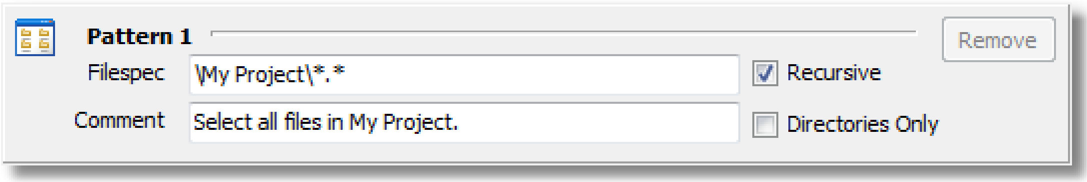
For more information on Patterns, see FileSet Patterns
For more information on Filters, see FileSet Filters

The following actions natively support FileSets:

- Burn CD/DVD
- Create ISO
- FTP Upload
- File Iterator
- Image Manipulation
- Lossless JPEG Transform
- Spell Checking
- Get File Size
- Check File Exists Action

6.8.2.1 FileSet Patterns

FileSet Patterns are used to specify which files are included and excluded from the FileSet.



Pattern 1
Filespec: \My Project*.*
☑ Recursive
Comment: Select all files in My Project.
□ Directories Only

The Filespec determines which files are added to the FileSet but files can be excluded by using exclusion patterns and filters.

The filespec can specify the following:

- Simple filespec, eg. *.cs
- Filespec including path from root directory, eg. \My project*.*
- Filespec including path occurring anywhere, eg. My project*.*

NOTE: If you are specifying all files in a certain directory, then the *.* filespec is optional, but the directory name must end with a slash. Eg. \DIR1\ is equivalent to \DIR1*.*

Simple filespec
This uses standard Windows type wildcards to specify files. The asterisk "*" specifies any characters and the question mark "?" specifies any single character.

Examples:

- All files: *.*
- All C# source files: *.cs
- Any file that has three characters and an extension of txt: ???.txt

Filespec including path from root

If the filespec begins with a backslash then that sub directory must be directly under the base directory of the filespec (specified on the Define Fileset property page). If you have recursion enabled and a directory by the same name is found elsewhere in the file system but not as a direct subdirectory of the root, then the directory will not be searched. This can be thought of as simply changing the root directory for this pattern.

Example:
Using two patterns, only choose files from the directories <root>\DCU\ and <root>\BAK\ even though the root directory may have many more subdirectories:

- Filespec 1: \DCU*.*
- Filespec 2: \BAK*.*

Filespec including path occurring anywhere
If the filespec contains a directory, but does not begin with a slash and recursive is specified then the directory will be selected anywhere it is found as a subdirectory from the root. For example, consider the following file structure:

```
<root dir>
  +--- Projects
        +--- Project 1
                +--- Output
        +--- Project 2
                +--- Output
```

If you want to select all *.exe files from either of the Output directories, specify that the pattern uses recursion and use the following filespec:

- Output*.exe

Exclusion patterns
Exclusion patterns function in a very similar way to include patterns. Exclusion patterns are used to test the directory or file found by the include pattern to see if the file or directory should be included or discarded. Exclusion patterns follow the same rules as include patterns as far as the formatting of the filespec goes. Exclude patterns can be used to specify the following:

- a filespec (eg. *.exe)
- a directory from the root, eg. \Out\
- a directory anywhere, eg. Out\
-

Specifying a directory may also include a filespec.

Some examples:

All files, except executables
Include Pattern: *.*
Exclude Pattern: *.exe

All files (recursive), except any directory with the name "Output"
Include Pattern: *.*
Exclude Pattern: Output\

All files with extension .cs (recursive), except not in the Output directory as a subdirectory of the root
Include Pattern: *.cs
Exclude Pattern: \Output\

6.8.2.2 FileSet Filters

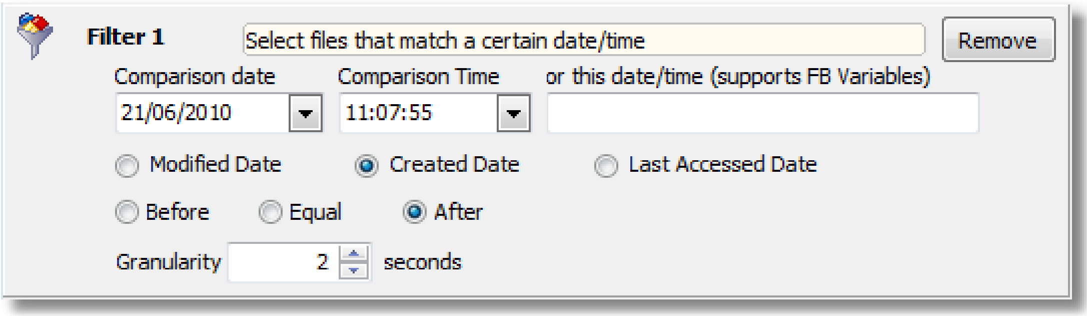
Filters are used to further refine the files chosen by properties other than the file name or directory. For example, you may want to only include files of a certain size, containing particular text, matching particular attributes, or based on date and time (or a combination of all of these). When a pattern finds a file it then passes this file onto each of the filters that have been defined, and only if the file passes through each filter does the file get added to the FileSet.

The filters available are:

- Date/Time
- Contents
- Attributes
- Size

Date/Time Filter

The Date/Time Filter allows you to filter the FileSet files based on their modified date, accessed date, or creation date.



Filter 1 — Select files that match a certain date/time

Comparison date: 21/06/2010
Comparison Time: 11:07:55
or this date/time (supports FB Variables):

○ Modified Date ● Created Date ○ Last Accessed Date

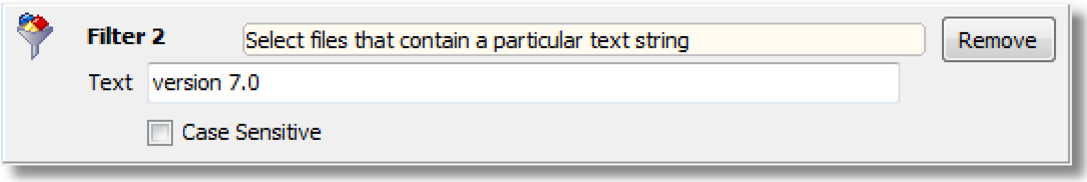
○ Before ○ Equal ● After

Granularity 2 seconds

This example shows that only files created after 4/2/2006 18:02:19 will be added to the FileSet. The granularity allows slight differences in the time you specify and the time retrieved from the Windows file system - a granularity of 2-5 seconds is recommended.

Contents Filter

The Contents filter allows you to filter the FileSet files depending if their contents contain certain text. This filter needs to open each file and scan for the text, so the performance of creating the FileSet may be poor, especially if it contains large files.



Filter 2 — Select files that contain a particular text string

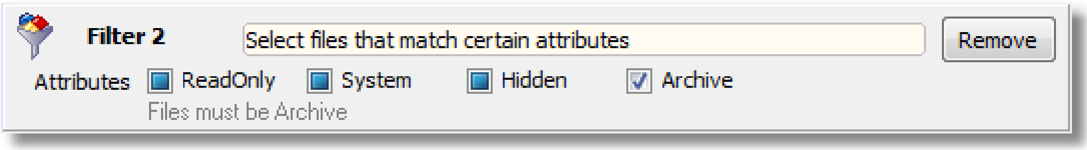
Text: version 7.0

□ Case Sensitive

This example shows that only files which contain the text "version 4.0" will be added to the FileSet.

Attributes Filter

The Attributes filter allows you to filter the FileSet files depending on the file system attributes of the files.



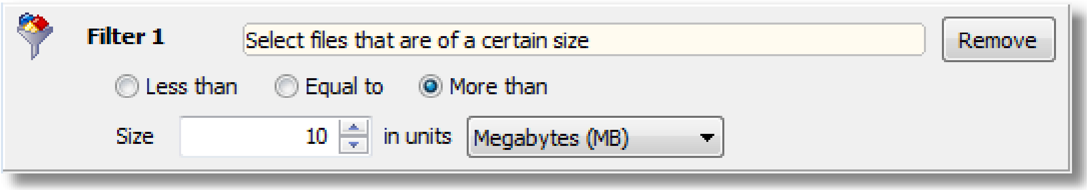
Filter 2 Select files that match certain attributes

Attributes ■ ReadOnly ■ System ■ Hidden ☑ Archive

Files must be Archive

This example shows that files must have the Archive bit set.

Size Filter
The Size filter allows you to filter the FileSet files depending on the size of the file specified in bytes, MB, KB or GB.



Filter 1 Select files that are of a certain size

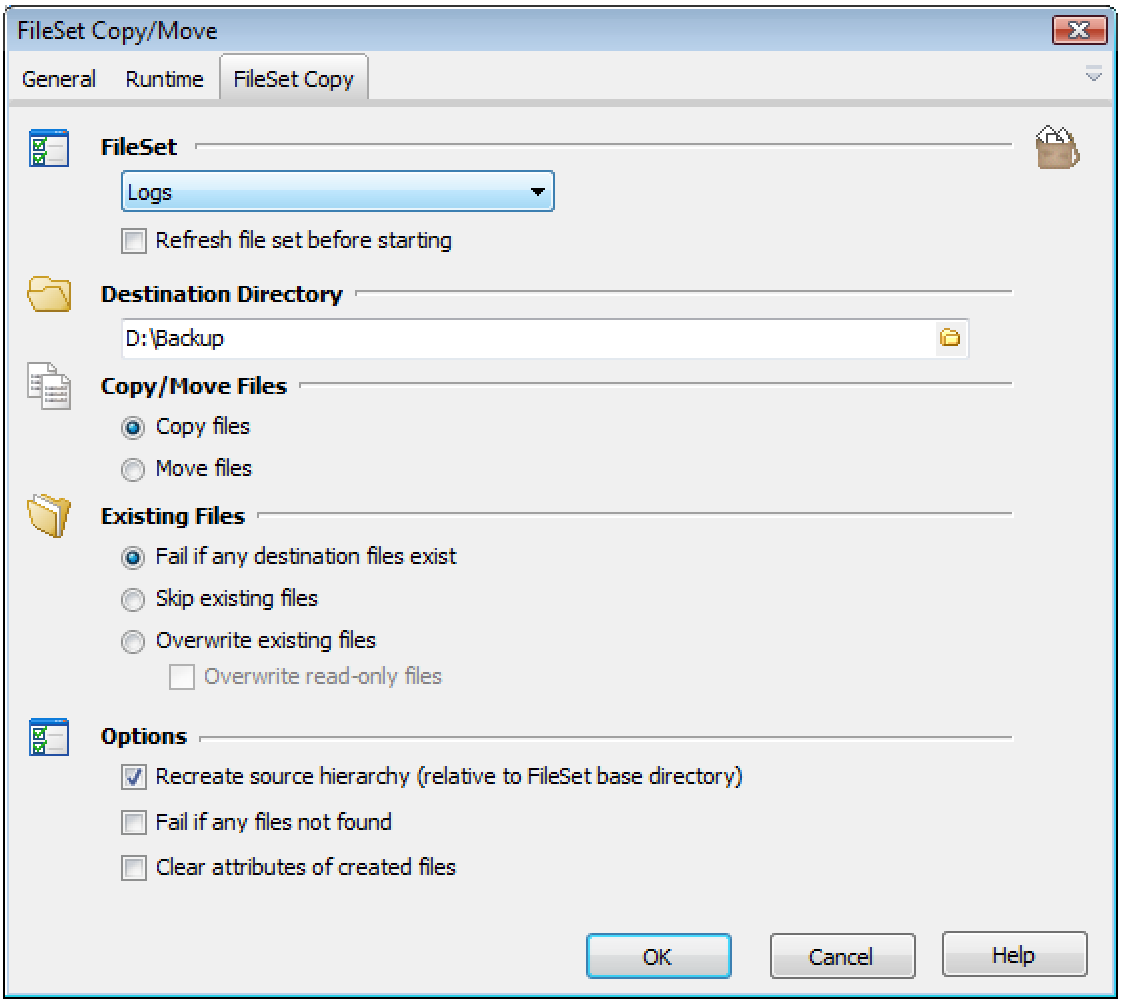
○ Less than ○ Equal to ● More than

Size 10 in units Megabytes (MB)

This example shows that only files bigger than 10 MB will be added to the fileset.

6.8.3 FileSet Copy/Move Action

The Copy/Move FileSet action allows you to copy or move all the files in a FileSet to a new location.



FileSet Copy/Move

FileSet

Logs

□ Refresh file set before starting

Destination Directory

D:\Backup

Copy/Move Files

● Copy files

○ Move files

Existing Files

● Fail if any destination files exist

○ Skip existing files

○ Overwrite existing files

□ Overwrite read-only files

Options

☑ Recreate source hierarchy (relative to FileSet base directory)

□ Fail if any files not found

□ Clear attributes of created files

FileSet - specify the name of the FileSet to use

Refresh FileSet before starting - if this option is set the FileSet will be refreshed before this action runs. This means it will be recreated by using the patterns/filters from the FileSet Define action. Use this option when it's possible that the files may have changed between when the FileSet was created to the point where you want to copy or move the files in the FileSet.

Destination Directory - the new location to move or copy the files to

Copy/Move Files - select the mode of the action

Existing Files - select what you would like the action to do if files of the same name are found in the destination directory

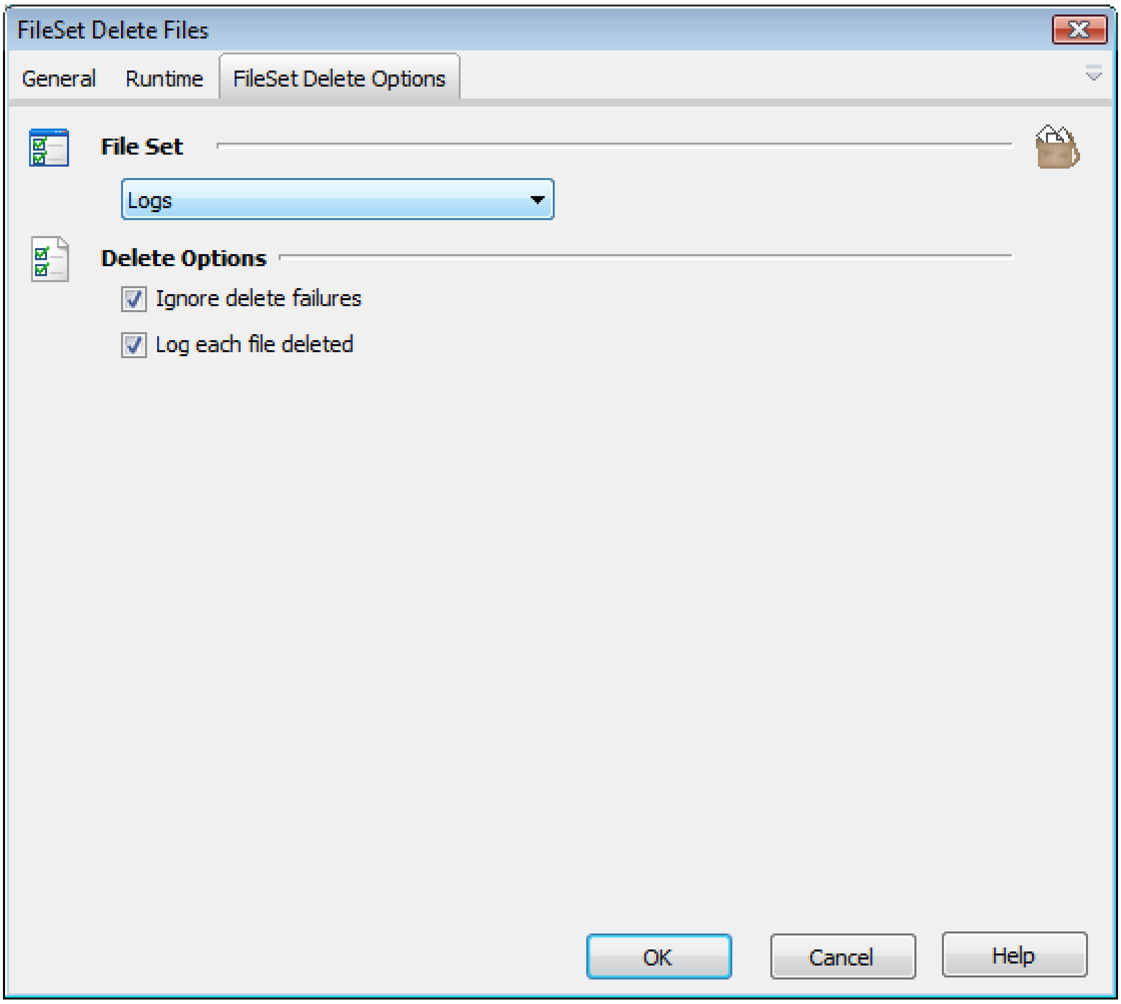
Recreate source hierarchy - if this option is not set, then all the files in the FileSet (regardless of their path) will be placed in the destination directory.

Fail if any files not found - the action will fail if any of the files in the FileSet are not found when performing the copy or move

Clear attributes of created files - this option removes the following flags from the newly created files: ReadOnly, System, Hidden, Archive

6.8.4 FileSet Delete Files

The FileSet Delete Files action will delete all the files in the specified FileSet.



File Set

Logs

Delete Options

☑ Ignore delete failures

☑ Log each file deleted

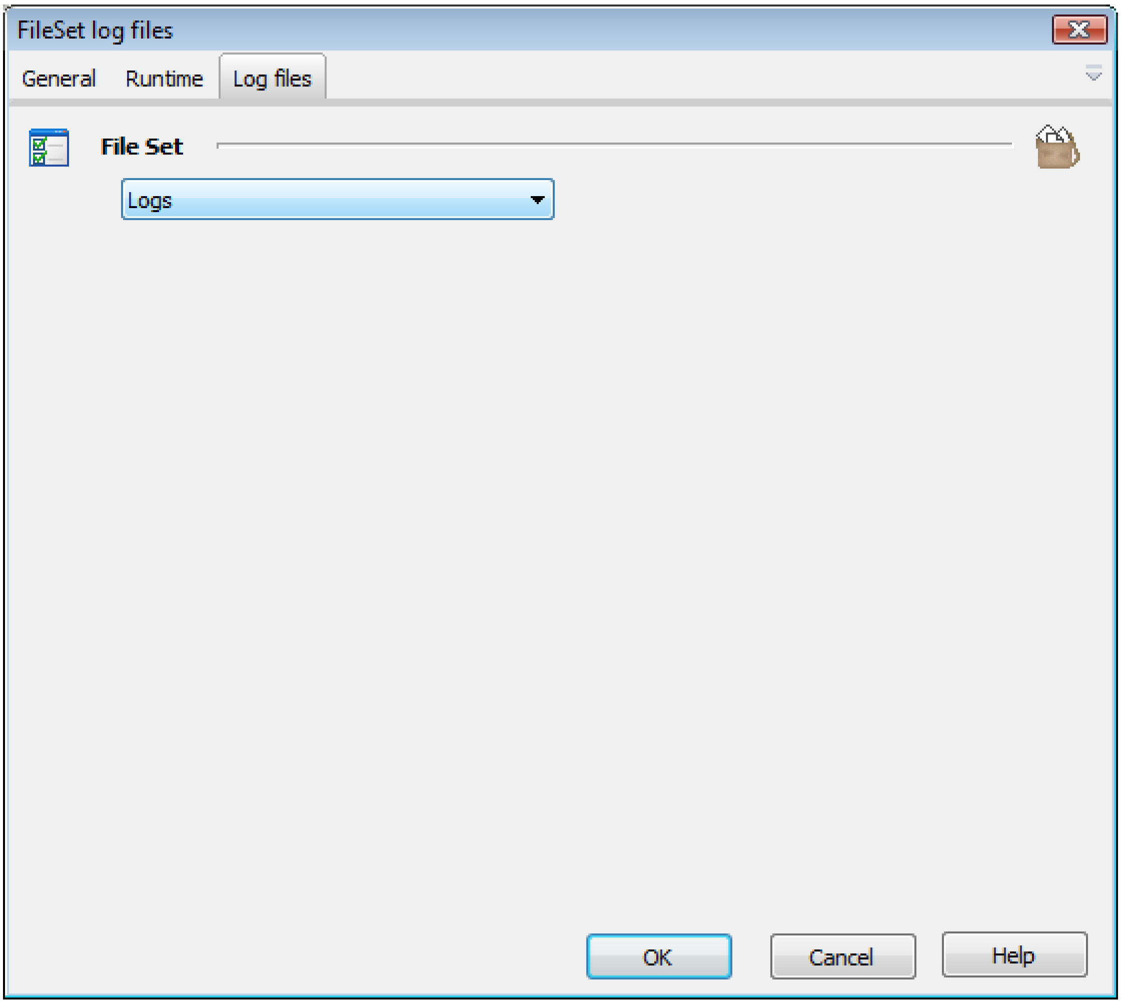
FileSet - specify the FileSet to use.

Ignore delete failures - do not fail the action or stop deleting files if a file fails to be deleted

Log each file deleted - write the filename to the log for each file as it's being deleted

6.8.5 FileSet Log Files

This action simply outputs to the log all the files in the selected FileSet. Use this action to keep a record of what files were in the FileSet in the project log.

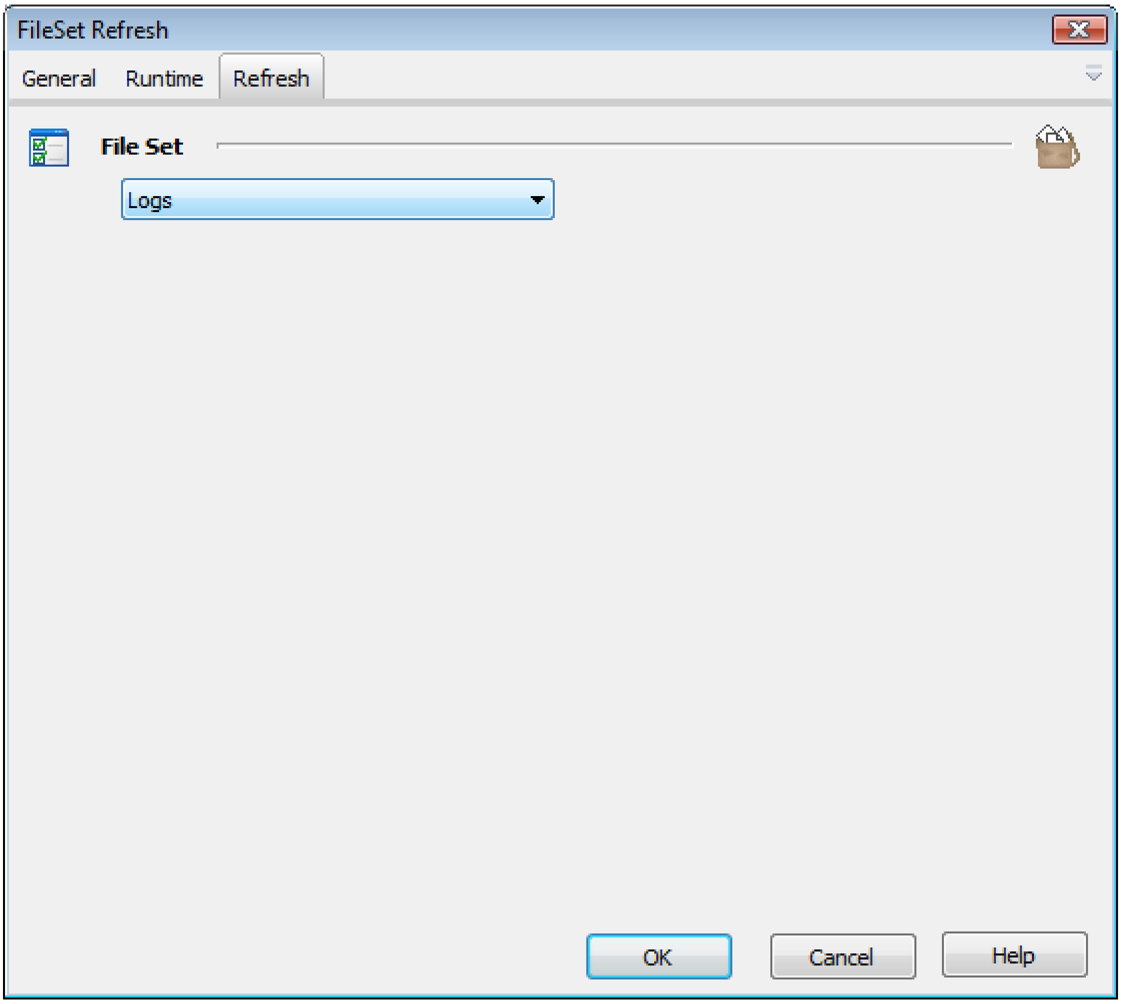


File Set

Logs

6.8.6 FileSet Refresh

The FileSet Refresh action simply re-runs the process to find files specified in the FileSet. When the FileSet Define action runs it processes the FileSet at that time, but at a later point in the project it may be necessary to Refresh the FileSet as files may have changed (eg. more files added, files deleted, attributes changed, etc.)



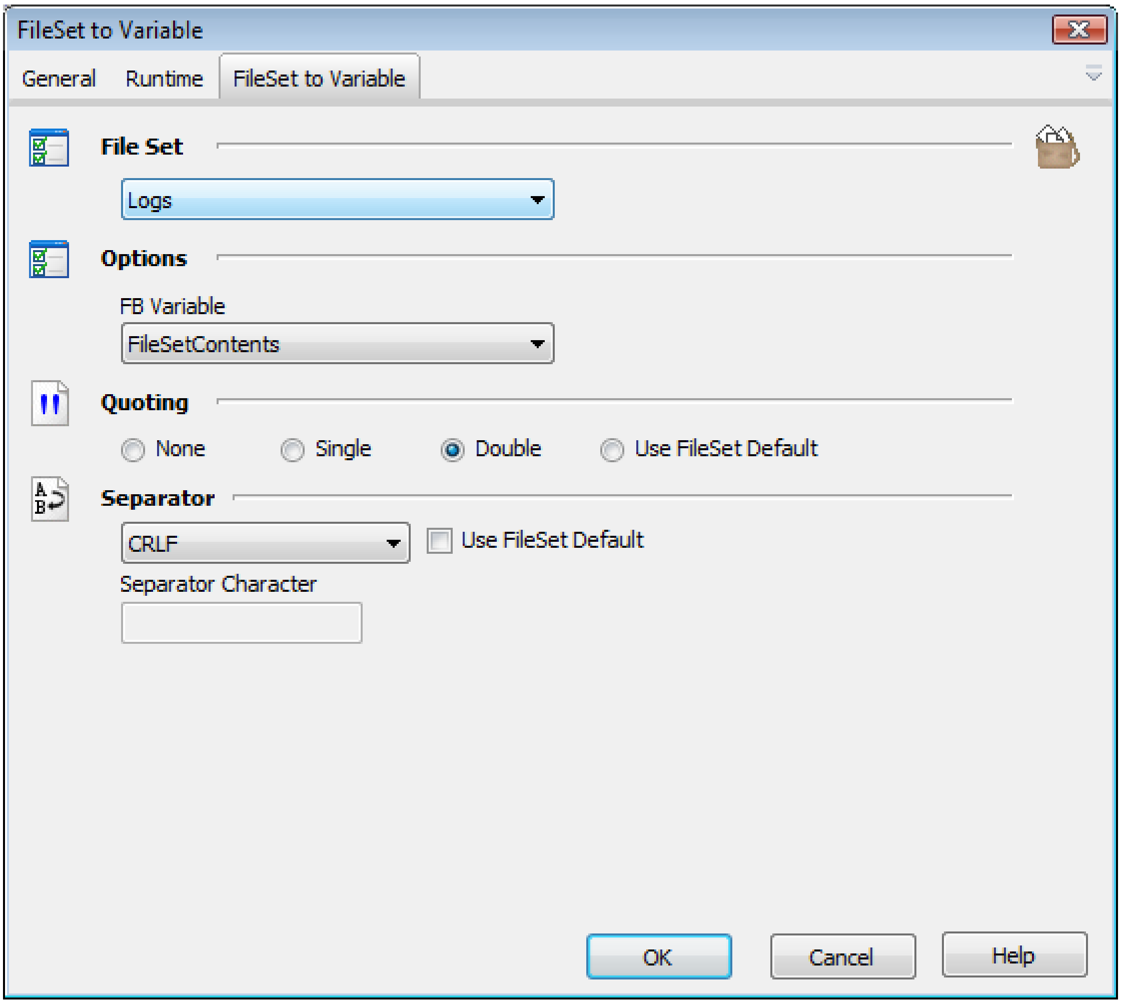
File Set

Logs

6.8.7 FileSet to Variable

The FileSet to Variable action allows you to create a listing of the files specified by the FileSet in a Automise variable. The properties of the action are:

- **File Set** - specify the FileSet to use
- **FB Variable** - specify the variable which will be used to output the file listing
- **Quoting** - specify the quoting style to use, either none, double quotes, single quotes or use the FileSet default (from the FileSet Define action)
- **Separator** - specify the separator to use if multiple files are in the FileSet. If the specific separator you want to use is not listed, then use the Other and fill in the Separator Character box with the character(s) you want to use. You may also specify to use the default separator defined for the FileSet



File Set
Logs

Options
FB Variable
FileSetContents

Quoting
○ None ○ Single ● Double ○ Use FileSet Default

Separator
CRLF □ Use FileSet Default
Separator Character

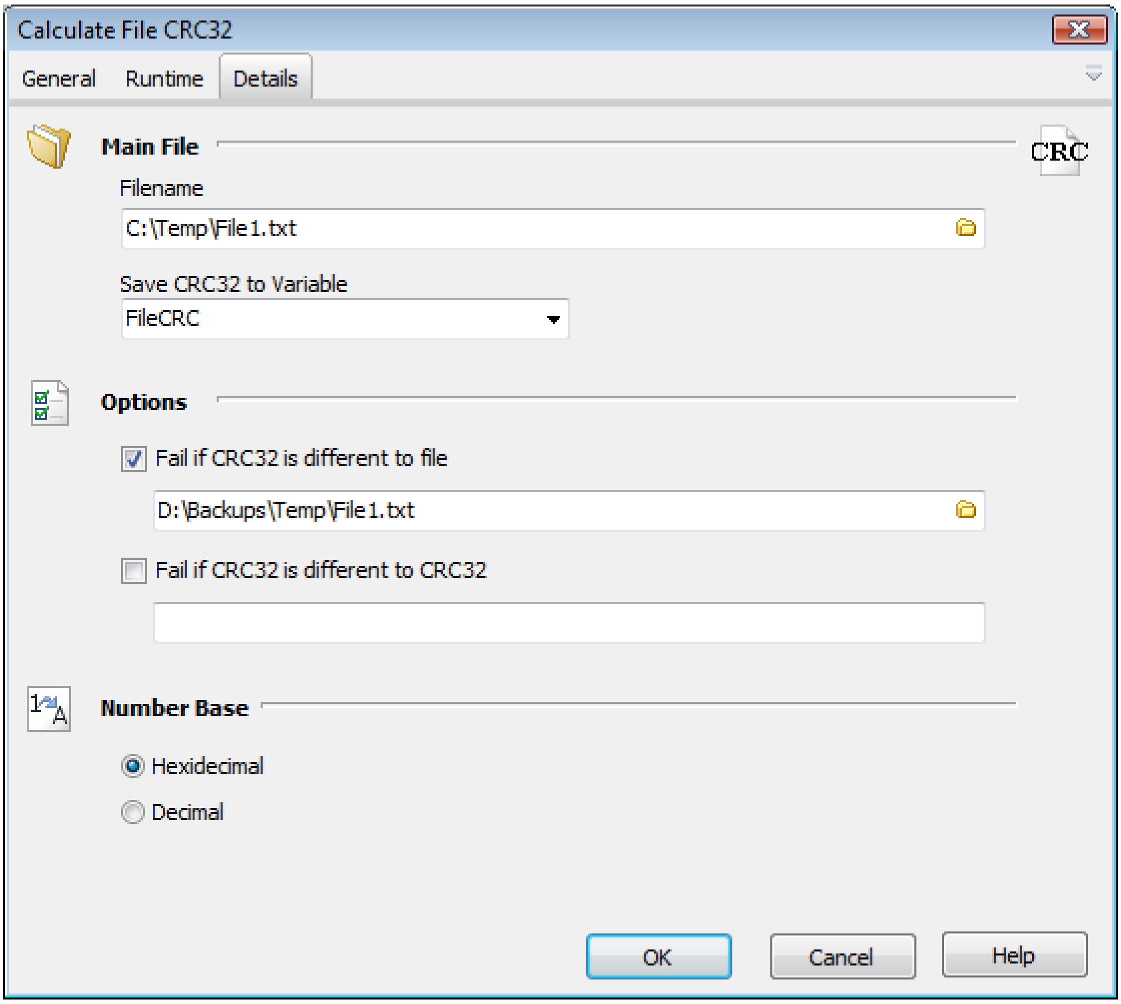
6.9 Files & Directories

6.9.1 Calculate File CRC32

The File CRC32 action enables you to automate the calculation of a file's CRC32. The action can also fail if the CRC differs from another file or a specified CRC32.

What is CRC32?

The CRC is acronym for "Cyclic Redundancy Code" and 32 represent the length of checksum in bits. The "CRC" term is reserved for algorithms that are based on the "polynomial" division idea. The idea to compute the checksum is equal for all CRC algorithms: Take the data as a VERY long binary number and divide it by a constant divisor. If you do this with integer values you get a rest; this rest is the CRC checksum (for example 7 / 3 = 2 + rest 1 => 1 is the checksum of 8. CRC is a family of algorithms and CRC32 is one certain member of this family (other members are CRC16, XMODEM,...); CRC32 produces a checksum with a length of 32 Bit (= 4Byte).



Main File

Filename: C:\Temp\File1.txt

Save CRC32 to Variable: FileCRC

Options

☑ Fail if CRC32 is different to file

D:\Backups\Temp\File1.txt

□ Fail if CRC32 is different to CRC32

Number Base

● Hexidecimal

○ Decimal

Filename - specify the file which you want to calculate the CRC32 for.

Variable - specify the Automise variable to save the calculated file CRC32 to.

Fail if CRC32 is different to file - this calculates the CRC32 of the specified file and compares it against the CRC32 of the main file and fails the action if the two differ.

Fail if CRC32 is different to CRC32 - specify a CRC32 to compare to the calculated CRC32 of the main file

Number Base - choose whether the CRC32 should be displayed/compared in hexadecimal or decimal format.

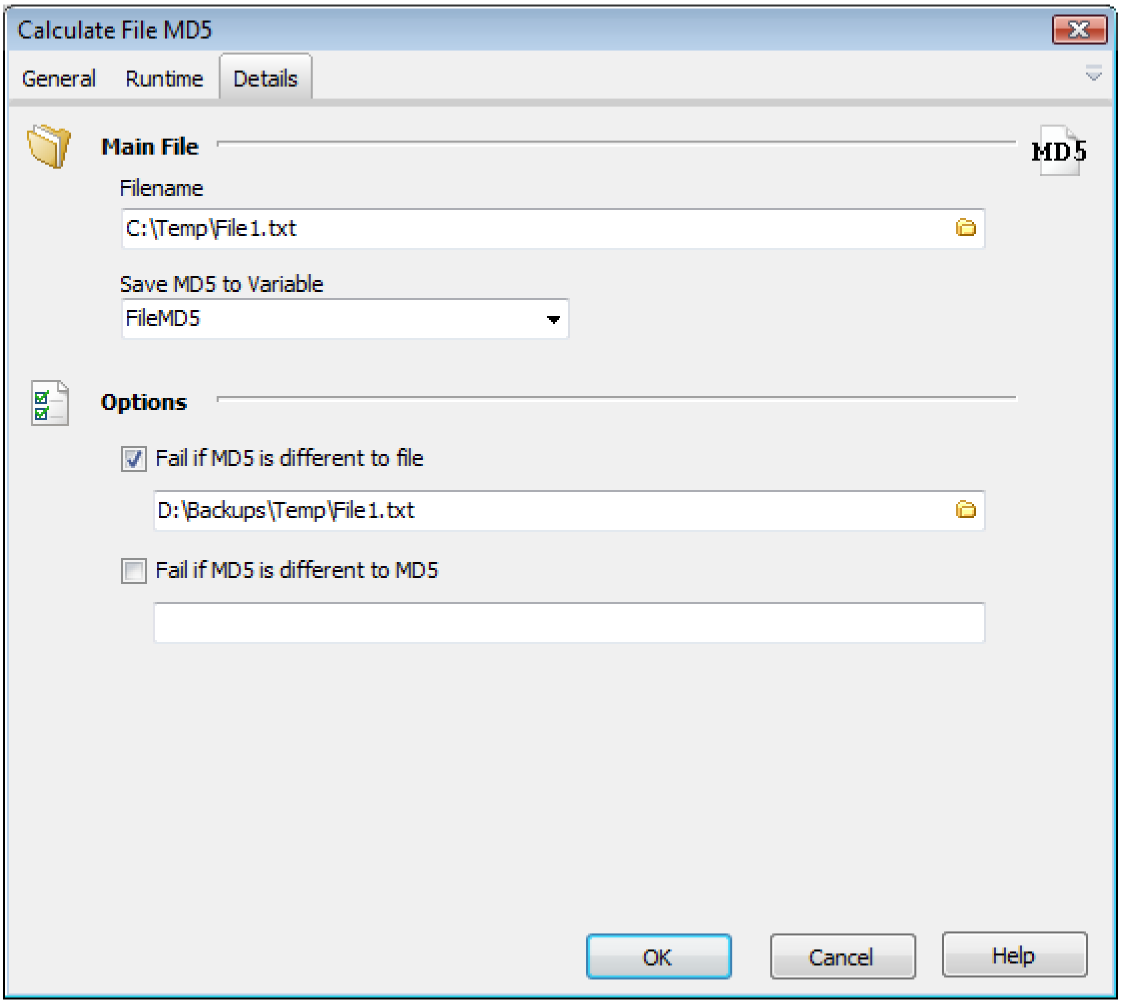
6.9.2 Calculate File MD5

The File MD5 action enables you to automate the calculation of a file's MD5. The action can also fail if the MD5 differs from another file or a specified MD5.

What is MD5?

The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key crypto system such as RSA.

In essence, MD5 is a way to verify data integrity, and is much more reliable than checksum and many other commonly used methods.



Calculate File MD5

Main File

Filename: C:\Temp\File1.txt

Save MD5 to Variable: FileMD5

Options

☑ Fail if MD5 is different to file

D:\Backups\Temp\File1.txt

□ Fail if MD5 is different to MD5

Filename - specify the file which you want to calculate the MD5 for.

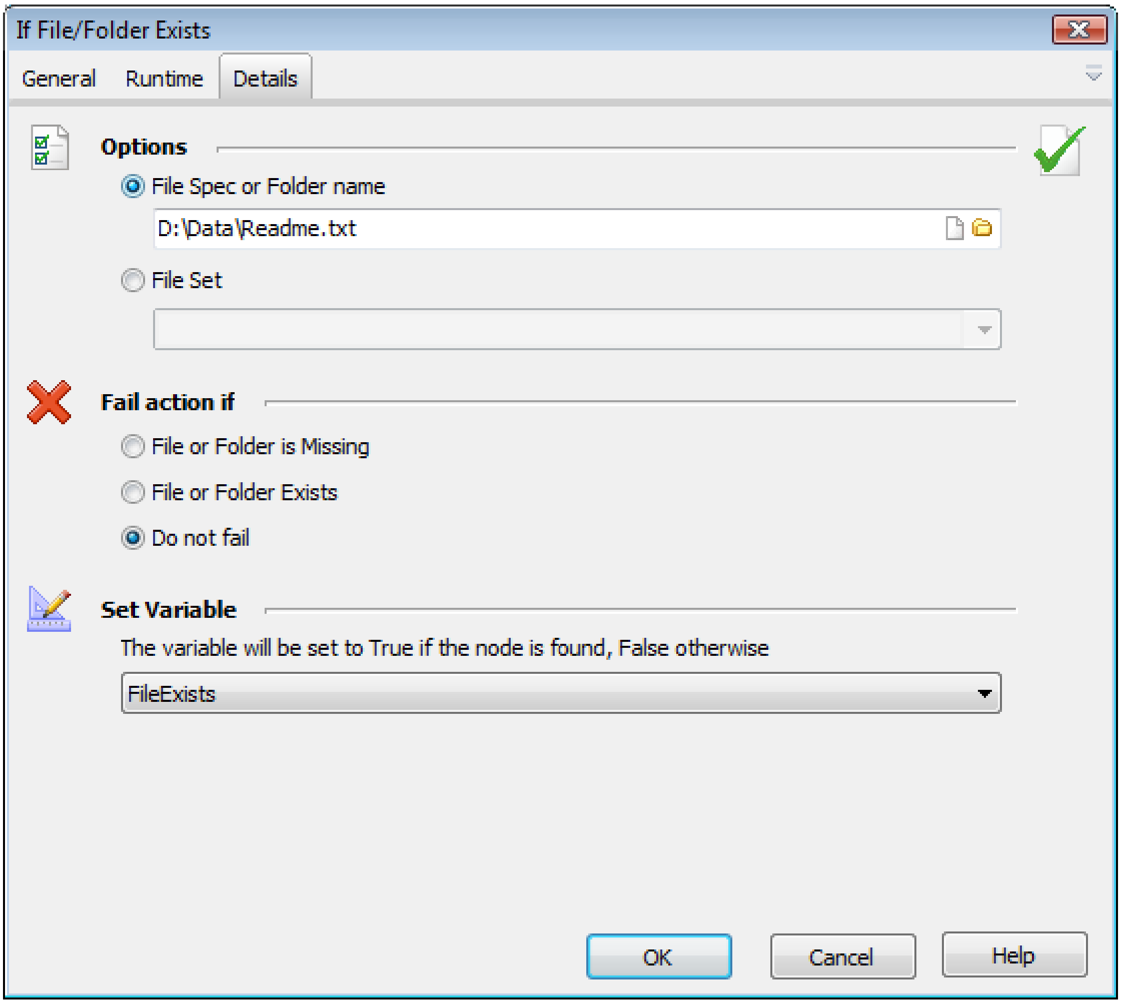
Variable - specify the Automise variable to save the calculated file MD5 to.

Fail if MD5 is different to file - this calculates the MD5 of the specified file and compares it against the MD5 of the main file and fails the action if the two differ.

Fail if MD5 is different to MD5 - specify a MD5 to compare to the calculated MD5 of the main file

6.9.3 Check if File Exists Action

This action allows you to check for the existence of a specific file or folder.



Options

● File Spec or Folder name

D:\Data\Readme.txt

○ File Set

Fail action if

○ File or Folder is Missing

○ File or Folder Exists

● Do not fail

Set Variable

The variable will be set to True if the node is found, False otherwise

FileExists

You can choose to fail the action if the file or folder exists, or if the file or folder is missing, or not to fail at all. You can also choose a variable to set. Its value will be set to "True" if the file or folder exists, or "False" otherwise.

When using the **Do not fail** option, you should specify a variable to hold the result of the file check.

Examples of usage:

A critical file must be present:
1.Specify the name of the file.
2.On the **Details** tab, choose **Fail Action If - File or Folder is Missing**.
3.Do not specify a variable.
The build will immediately abort if the required file is missing.

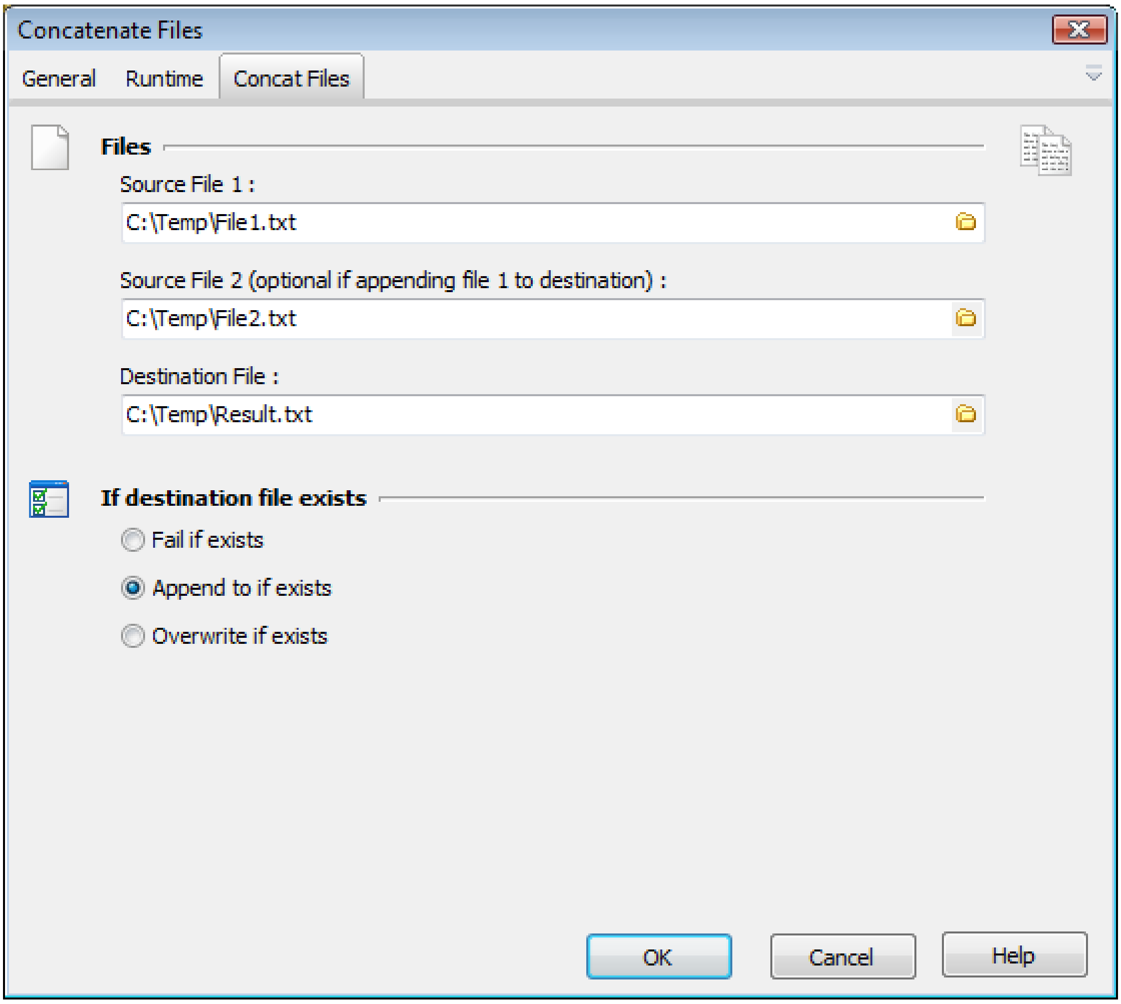
The user should be warned if any files are present in a directory.
1.Specify a path and a file mask, such as c:\projects\Output\*.*
2.On the **Details** tab, choose **Fail Action If - File or Folder is Missing**.
3.On the **Runtime** tab, select **Fail Action If - Do not fail**.
4.Create a (nested) sub-node with the required warning behaviour.
The build will skip the sub-node if the file is not present, but will keep running.

Complex logic involving the presence of a file:
1.Specify the name of the file.
2.Choose "Do not Fail"
3.Specify the name of a variable which will indicate the presence of the file.
4.In a later "If...Then" step, use the name of the variable.

"True" indicates that the file exists. "False" indicates that the file does not exist.

6.9.4 Concatenate Files Action

This action concatenates two text files into a single new file, or merges/appends a file into the destination file.



Source File 1 : C:\Temp\File1.txt

Source File 2 (optional if appending file 1 to destination) : C:\Temp\File2.txt

Destination File : C:\Temp\Result.txt

If destination file exists

○ Fail if exists

● Append to if exists

○ Overwrite if exists

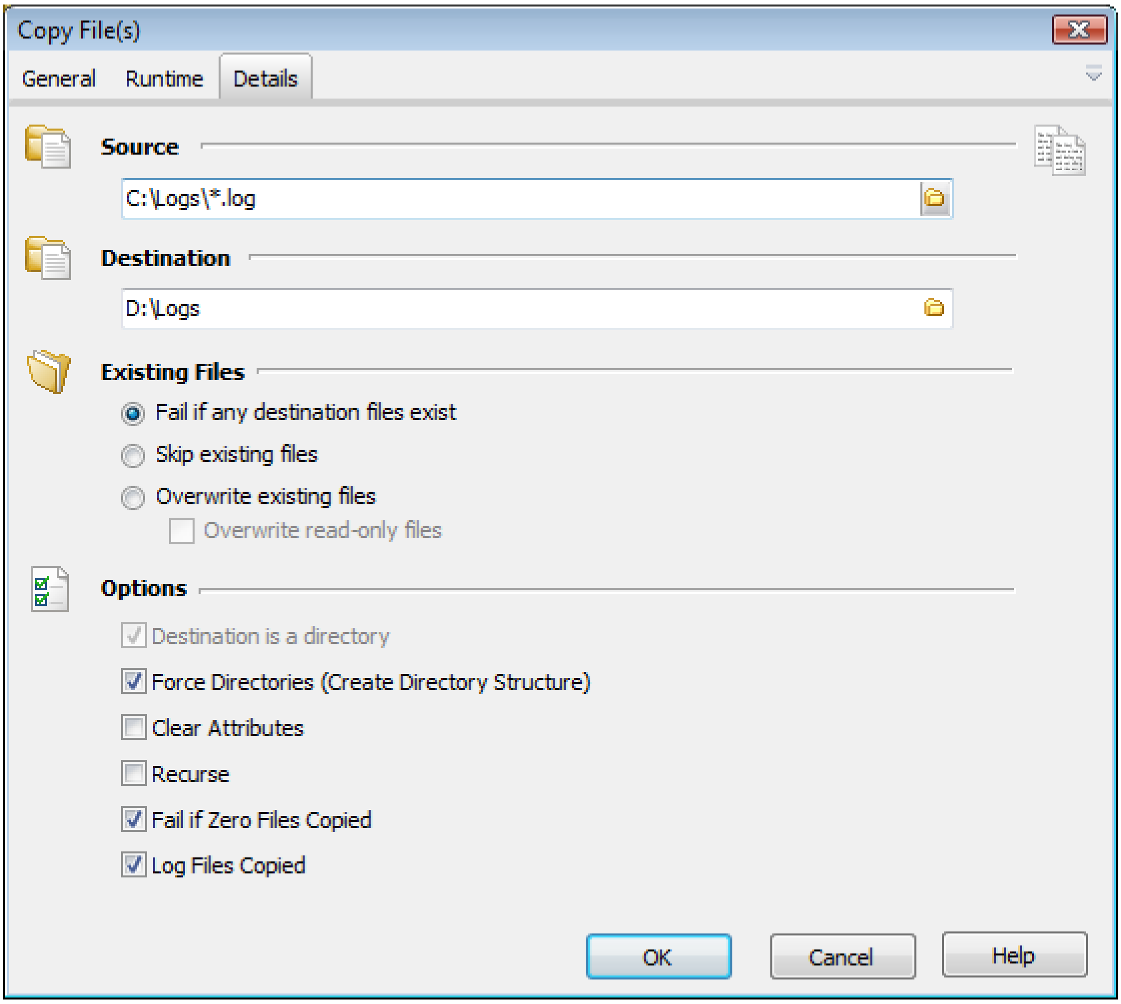
Note that Source Files 1 & 2 cannot be the same file, and they cannot be the same as the destination file. Source 2 may be blank if appending to an existing file.

6.9.5 Copy File(s) Action

The Copy File(s) action will copy a File or a set of files that match a file spec to the destination file or directory. You can use wildcards in the source setting to copy more than one file (the "Destination is a directory" property will be set automatically). To copy files from subdirectories as well, check the Recurse property. If the target directories do not already exist, check the Force Directories property to make Automise create the directories as needed.

When copying a single file (not using wildcards), then you can use the Destination field to rename the file at the same time as copying it to another location.

The Copy/Move FileSet action provides more sophisticated support for choosing files to copy.



Source: C:\Logs\*.log

Destination: D:\Logs

Existing Files

● Fail if any destination files exist

○ Skip existing files

○ Overwrite existing files

□ Overwrite read-only files

Options

☑ Destination is a directory

☑ Force Directories (Create Directory Structure)

□ Clear Attributes

□ Recurse

☑ Fail if Zero Files Copied

☑ Log Files Copied

Existing Files

If a file already exists at the destination, select how you would like the action to behave - fail, skip or overwrite.

Destination is a directory - Set this option to indicate that the destination is a directory name, not a file name.

Force Directories - If this option is set and the destination directory, or any parent directories, do not exist, then they will be created.

Clear Attributes - If this option is set, the destination file will be created with all attributes (Archive, Read Only, Hidden, System) cleared.

Recurse - Set this option to search for file spec matches in subdirectories of the source directory.

Fail if Zero Files Copied - If this option is set and no files are copied, the action will fail.

Log Files Copied - Each file copied will be logged. If this option is not set, only skipped files or files with errors are logged.

Scripting Info

The Action properties available are :

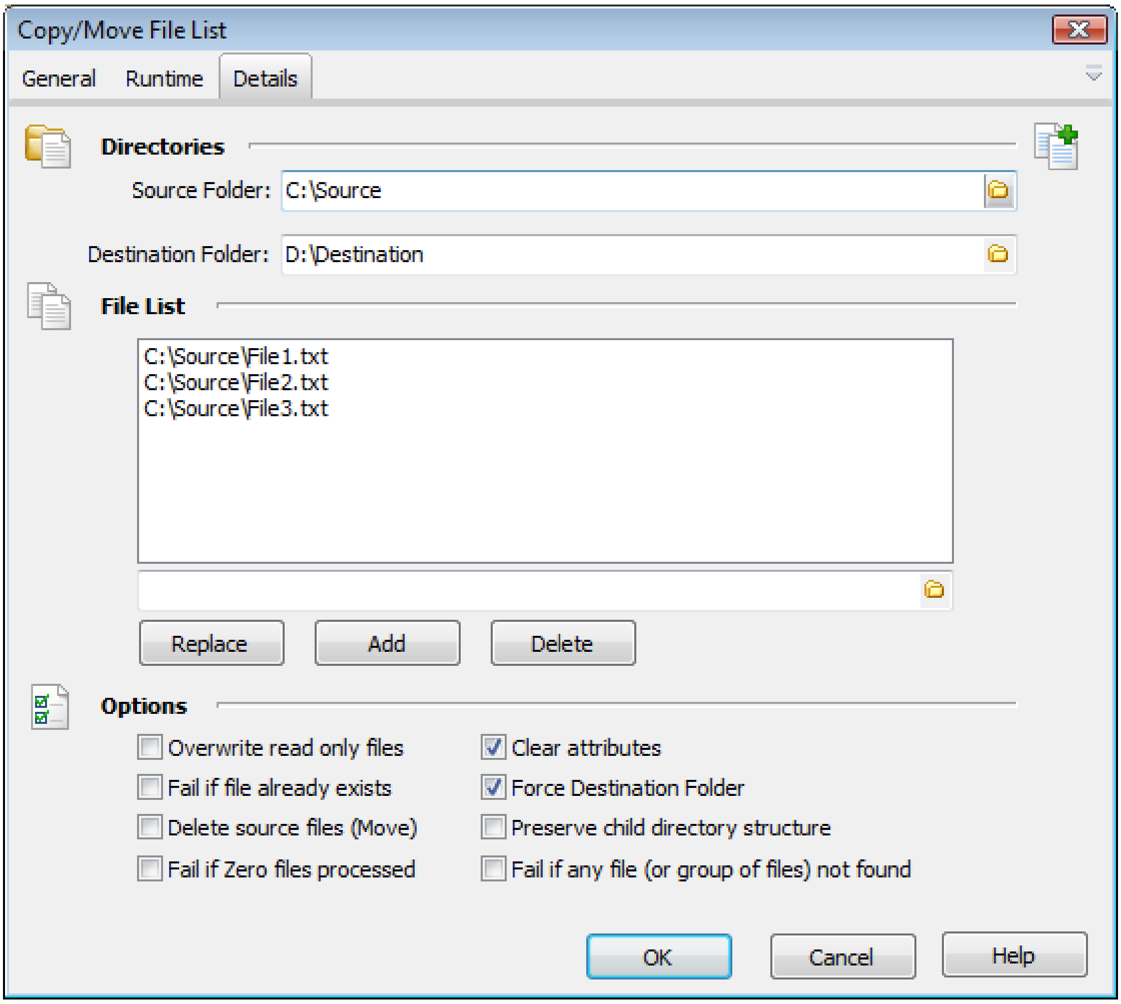
property FileSpec :WideString; // the file specification for the files to copy. You may use Wildcards.
property FilesAffected : integer ; // read only, the number of files Copied
property Target :WideString; // The target file or directory
property Force : WordBool; //Use Force Directories to create a directory and all parent directories that do not already exist.
property TargetIsDir : WordBool;// Specifies that the Target Is a Directory
property FailIfExists : WordBool;// Fail if the target file Exists
property Overwrite : WordBool; // Overwrite existing files
property OverwriteReadOnly : WordBool;//Overwrite Read Only Files
property Recurse : WordBool;// Recurse subdirectories, only valid if the target is a directory
property ClearAttributes : WordBool // clear the copied file's attributes.
property FailIfZeroFiles : WordBool;
property LogFilesAffected : WordBool;

6.9.6 Copy/Move File List

The Copy/Move File List Action allows you to copy move a list of files from one directory to another.

WARNING: This action is deprecated and may be removed in a future version of Automise. Use the Copy/Move FileSet action, instead.

This action was written by Jim Gunkel from Nevrona Designs. It makes it possible to provide several different file specs for the files that should be moved/copied. Jim has kindly made this action available to all Automise users. The source for this action is installed as an example of creating custom Automise Actions.



Directories

Source Folder: C:\Source

Destination Folder: D:\Destination

File List

C:\Source\File1.txt
C:\Source\File2.txt
C:\Source\File3.txt

Options

□ Overwrite read only files
□ Fail if file already exists
□ Delete source files (Move)
□ Fail if Zero files processed
☑ Clear attributes
☑ Force Destination Folder
□ Preserve child directory structure
□ Fail if any file (or group of files) not found

Source Folder

The folder where the files will be copied/moved from

Destination Folder

The Folder where the files will be copied/moved to.

Overwrite Read Only Files

Overwrite existing read only Files.

Clear Attributes

Clear the file attributes (Read Only, Hidden, System) during the copy/move.

Fail if file already exists

Fail if a file to be copied/moved already exists in the destination directory

Delete Source Files (Move)

Move the files (default is copy)

Force Destination Folder

When set, the Destination Folder will be created if it does not exist

Preserve child directory structure

When set, files from subdirectories inside the source directory will be copied into the appropriate sub-folders inside the destination folder. Files located outside the source folder are be copied directly into the destination folder.

In the example above, if "Preserve child directory structure" was set, then the file C:\temp\testing\bugreport.txt would be copied to C:\other\testing\bugreport.txt but the file C:\debug.txt would be copied to C:\other\debug.txt.

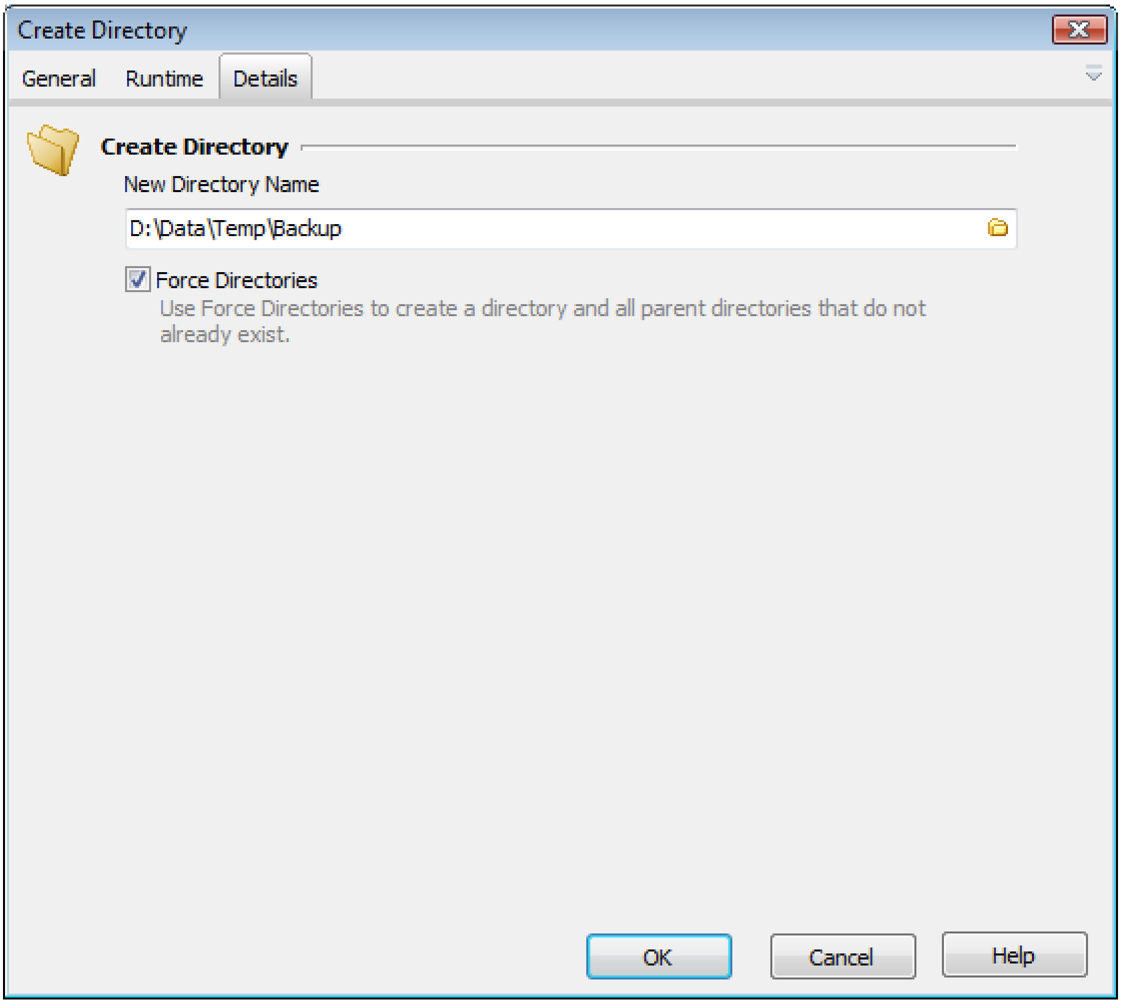
File List

The list of files to be moved/copied from the Source Folder. This can include wildcards. Paths can be local (relative to the source directory) or absolute.

(Note that wildcards do not recurse subdirectories.)

6.9.7 Create Directory Action

This action will create the specified directory, if the specified directory already exists then the action will not do anything.



Create Directory

New Directory Name: D:\Data\Temp\Backup

☑ Force Directories

Use Force Directories to create a directory and all parent directories that do not already exist.

Force Directories

Select Force Directories to create a directory and all parent directories that do not already exist. (If this option is not set, and you try to create a directory whose parent directory does not exist, then the action will fail.)

Scripting Info

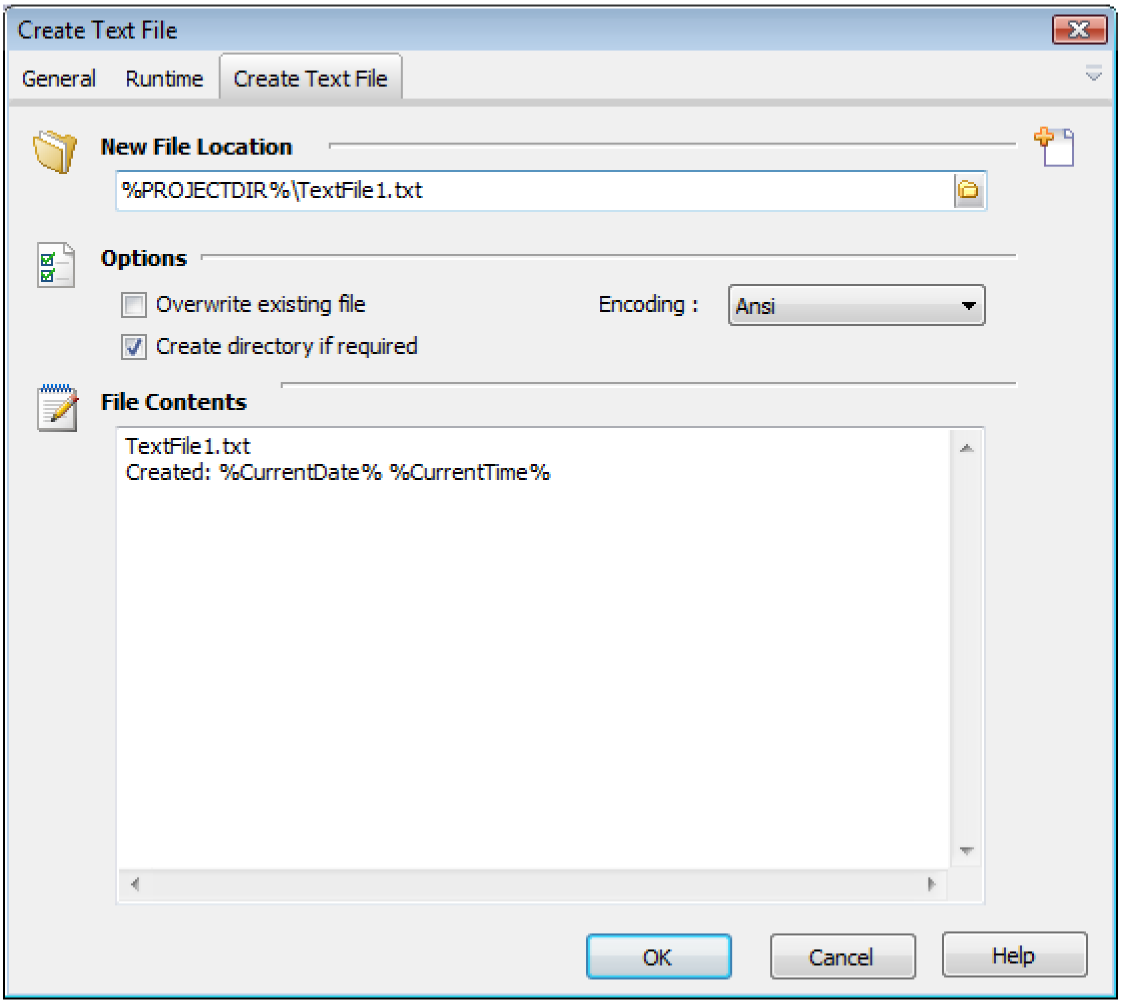
The Action properties available are :

property FileOrDirectory : WideString;// The Directory to Create
property Force : WordBool; // Force the creation of directories.

6.9.8 Create Text File

The Create Text File action enables you to create a new file containing text.

To add text to an existing text file, use the Write Text File action.



New File Location: %PROJECTDIR%\TextFile1.txt

□ Overwrite existing file

Encoding : Ansi

☑ Create directory if required

File Contents:

```
TextFile1.txt
Created: %CurrentDate% %CurrentTime%
```

New File Location
Path to new file to create.

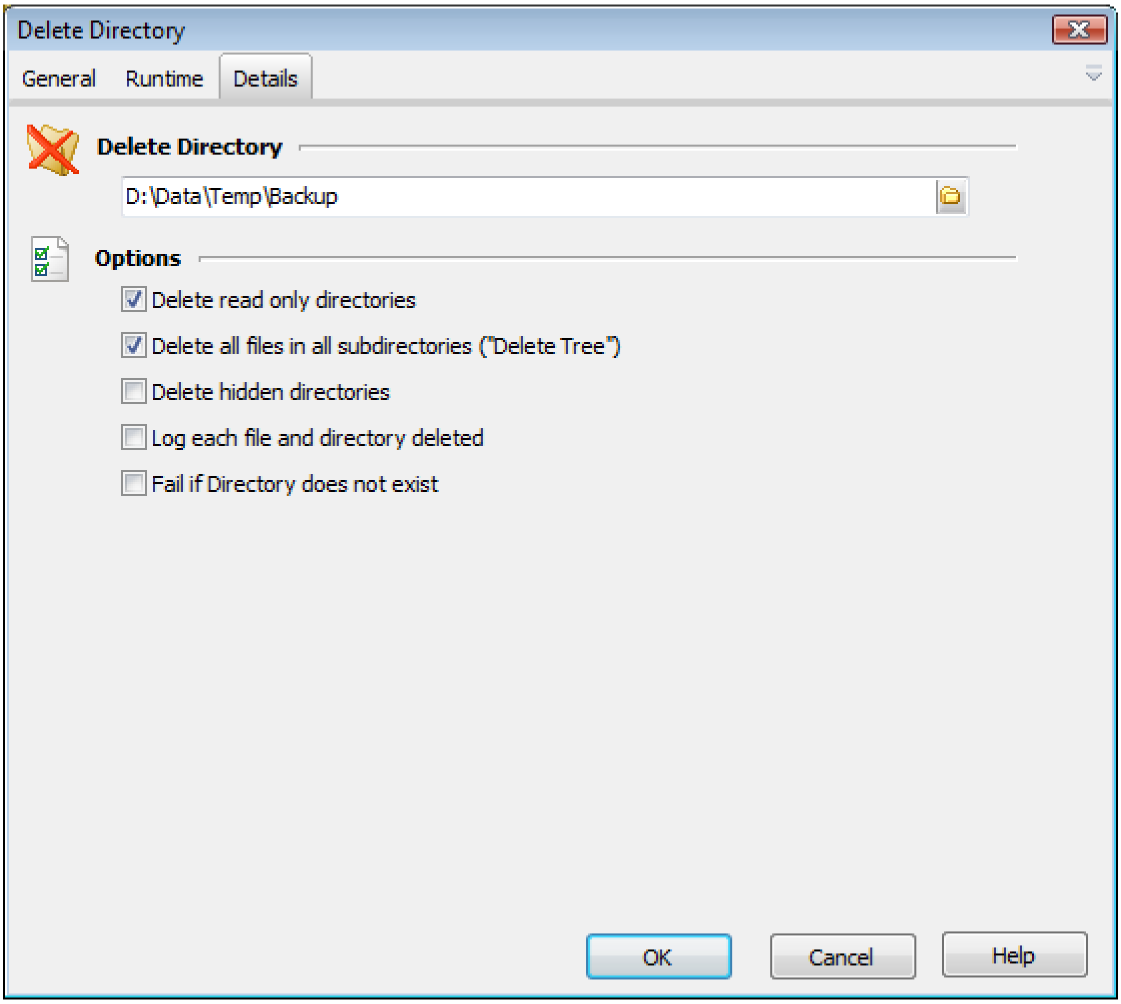
Overwrite Existing File
If this is not set, and the file already exists, the action will fail.

Create Directory if required
If any directory doesn't exist in the new file location, then it will be automatically created.

File Contents
Specify the text to be written to the file. You can use Automise variables in the File Contents field to customise the file based on the build.

6.9.9 Delete Directory Action

This action will delete the specified directory. If Delete Tree is set then the action will delete any files in the target directory, if not then the action will fail if the directory is not empty.



Delete Directory

General Runtime Details

Delete Directory

D:\Data\Temp\Backup

Options

☑ Delete read only directories

☑ Delete all files in all subdirectories ("Delete Tree")

☐ Delete hidden directories

☐ Log each file and directory deleted

☐ Fail if Directory does not exist

Delete Read Only / Hidden Directories

If these options are set, directories will be deleted even if they have their "Read Only" or "Hidden" attributes set. Otherwise, attempting to delete such directories will result in an error.

Delete Tree

If Delete Tree is set, the action will recursively delete all files and subdirectories of the chosen directory.

Log each deleted file and directory

If this option is set, the pathname of each file and directory will be logged as it is deleted.

Scripting Info

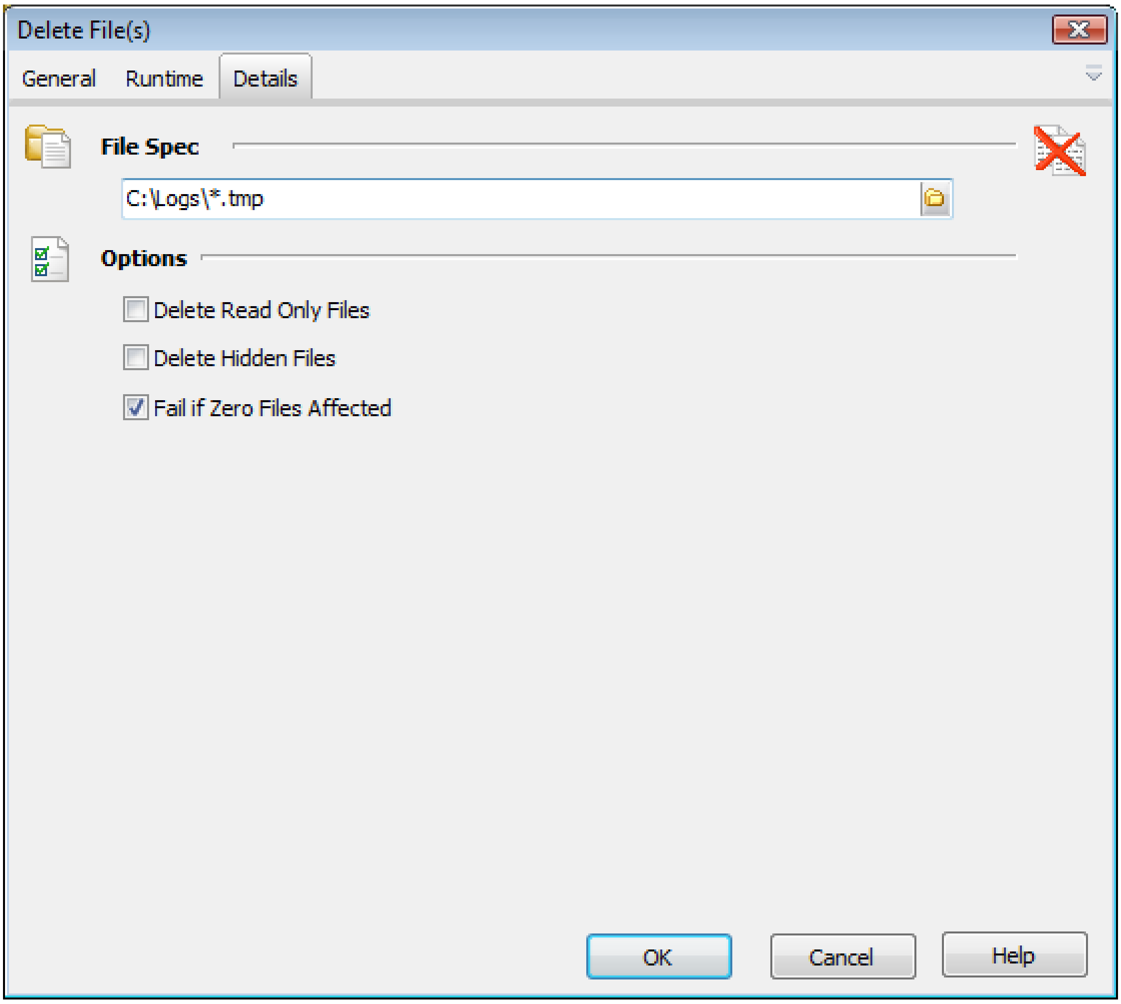
The Action properties available are :

property FileOrDirectory : WideString;// The Directory to Delete
property DeleteReadOnly : WordBool;
property DeleteHidden : WordBool;
property DeleteTree : WordBool;

6.9.10 Delete File(s) Action

The Delete files action deletes one or more files.

Note that this action does not provide an option to recurse through directories. For this you need the Delete Directory Action.



C:\Logs\*.tmp

□ Delete Read Only Files

□ Delete Hidden Files

☑ Fail if Zero Files Affected

FileSpec

Specify the file(s) to delete. You can use wildcards (? or *) in the file name.

Delete Read Only Files - Normally read only files will not be deleted. Check this option to remove them.

Delete Hidden Files - Similar to Delete Read Only files.

Fail If Zero Files Affected - If this option is checked and no files are found, the action will fail.

Scripting Info

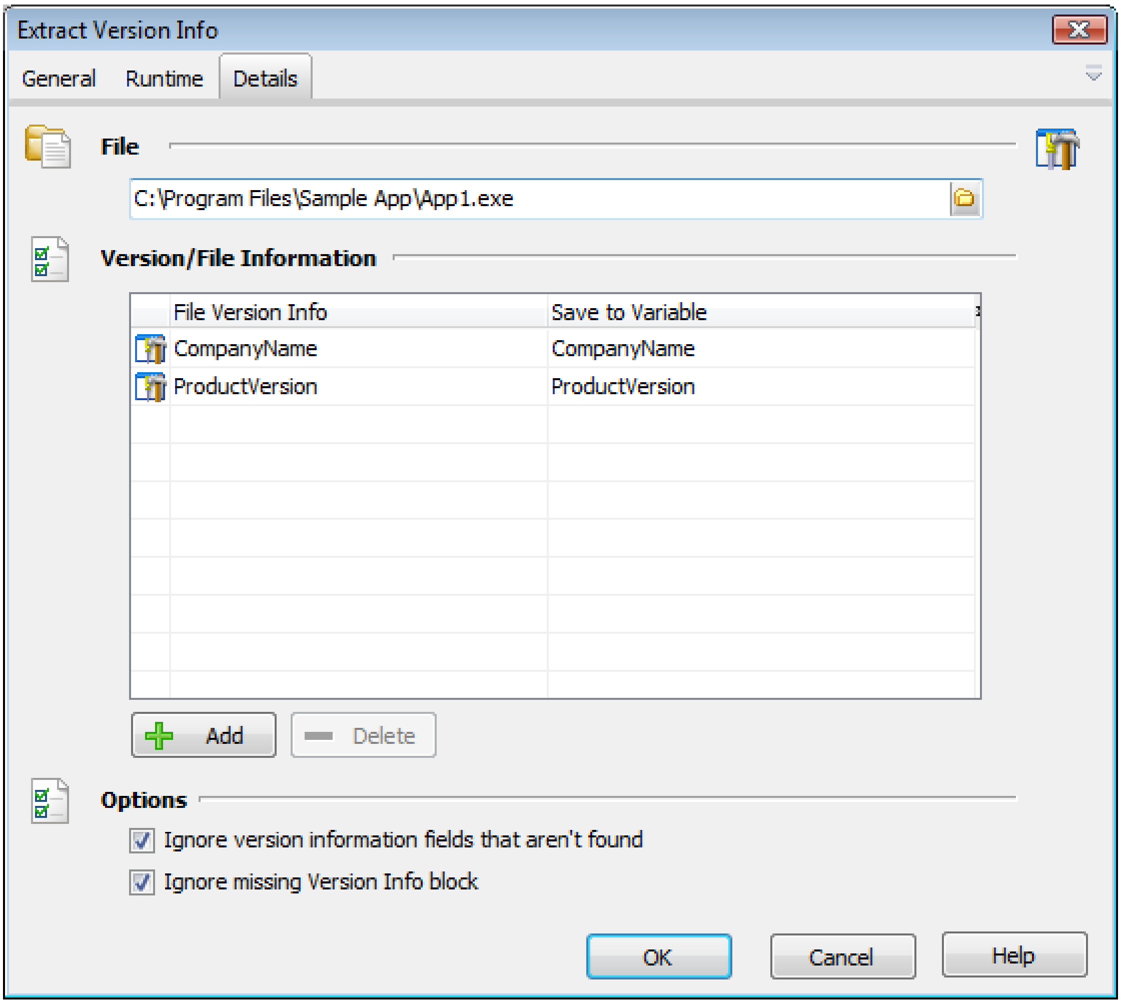
The Action properties available are :

property FileSpec :WideString; // the file specification for the files to delete. You may use Wildcards.
property FilesAffected : integer ; // read only, the number of files deleted
property FailIfNoFile : WordBool; // fail if no files affected.
property DeleteReadOnly : WordBool; // delete read only files.
property DeleteHidden : WordBool; //delete hidden files.
property LogFilesDeleted : Boolean; //logs each file that gets deleted

6.9.11 Extract Version Info

The Extract Version Information action enables you to extract the Win32 Version and File Information stored in the executable.

Extracting file information works with both normal Win32 files as well as .Net assemblies.



C:\Program Files\Sample App\App1.exe

File Version Info	Save to Variable
CompanyName	CompanyName
ProductVersion	ProductVersion

☑ Ignore version information fields that aren't found
☑ Ignore missing Version Info block

File
Specify a Win32 or .Net EXE, OCX, or DLL file (may work with other types too.)

File Version Info
Specify the field you want to extract. You may either select from the predefined types, or type in your own value.

For most names, the "File Version Info" name reflects the name of a string in the file resources. The exceptions are "Product Version" and "File Version", which return the full version numbers.

Save To Variable
Specify the Automise variable to save the value to.

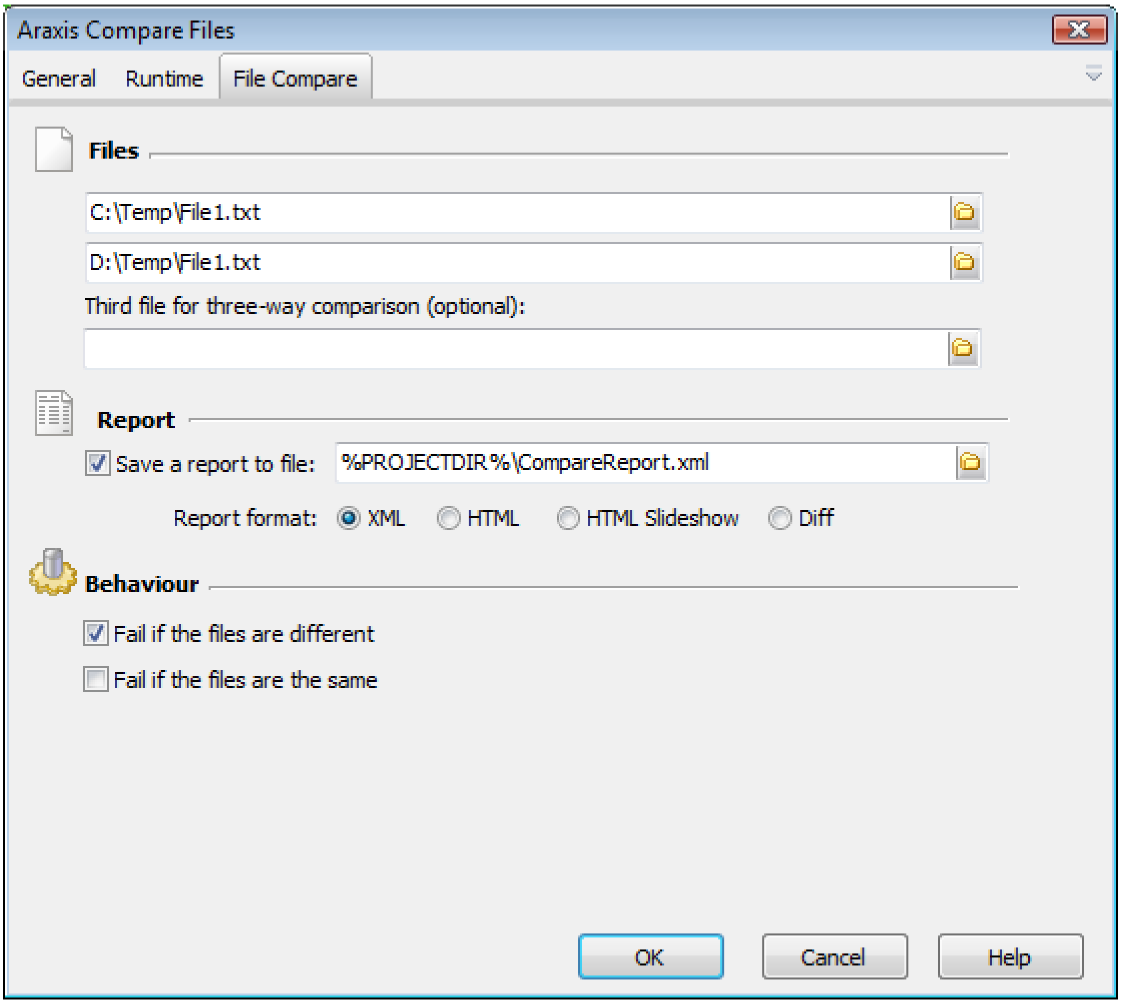
Ignore Version Information fields that aren't found
Set this to ignore any fields that are not found. The variables will be set to a blank string.

6.9.12 File Comparison Tools

6.9.12.1 Araxis Merge Compare Actions

Automise supports the "File Compare" and "Folder Compare" functionality of Araxis Merge, which is a third-party product produced by Araxis Software.

Both the Araxis File Compare & Araxis Folder Compare actions use the preferences set inside the Araxis Merge application to determine comparison options.



C:\Temp\File1.txt
D:\Temp\File1.txt
Third file for three-way comparison (optional):

☑ Save a report to file: %PROJECTDIR%\CompareReport.xml
Report format: ● XML ○ HTML ○ HTML Slideshow ○ Diff

☑ Fail if the files are different
☐ Fail if the files are the same

Files

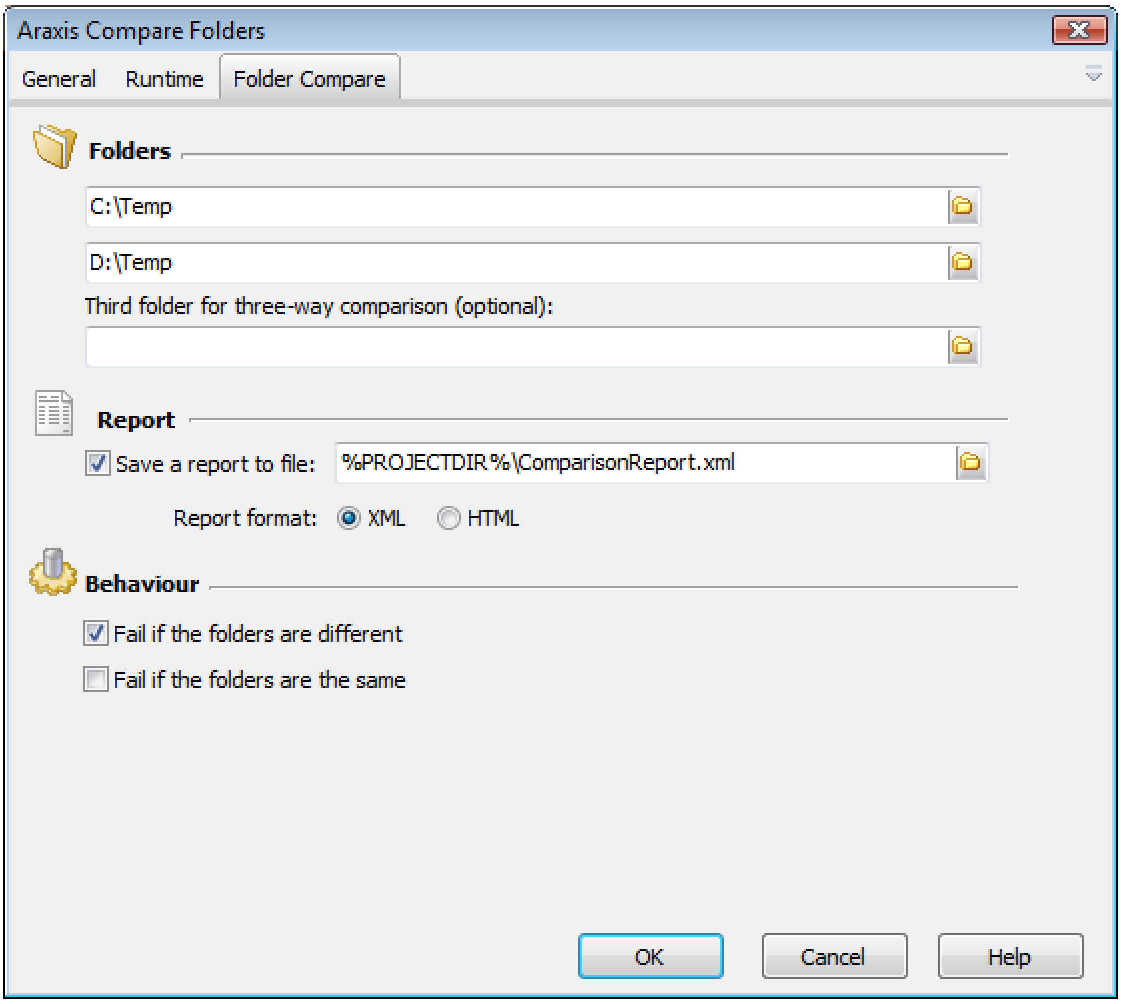
Araxis Merge can compare either two or three files, generating a two- or three-way comparison report.

Report

Araxis Merge can optionally save a comparison report in XML, HTML or Diff format. See the Araxis Merge documentation for details on each format.

Behavior

The action can be set to fail if the files are different, fail if they are the same, or fail only on errors (if neither box is checked.)



C:\Temp

D:\Temp

Third folder for three-way comparison (optional):

☑ Save a report to file: %PROJECTDIR%\ComparisonReport.xml

Report format: ● XML ○ HTML

☑ Fail if the folders are different

□ Fail if the folders are the same

NB: Comparing folders with large contents can take a long time to complete.

Folders

Specify two or three folders to compare.

Report

Similarly to the Araxis Compare Files action, Araxis Compare Folders can save a comparison report. Available report formats are HTML or XML.

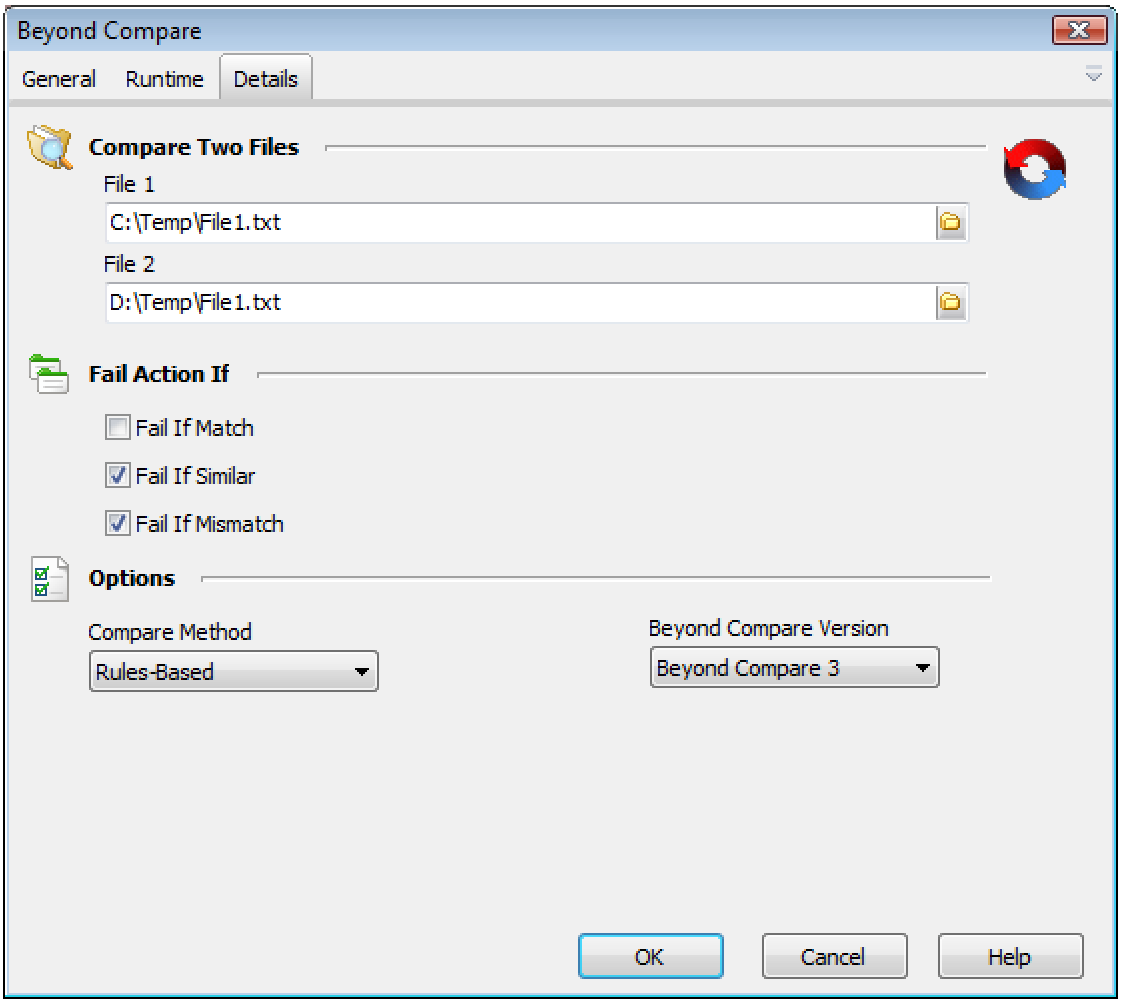
Behaviour

The action can be set to fail if the folders are different, fail if they are the same, or fail only on errors (if neither box is checked.)

6.9.12.2 Beyond Compare

6.9.12.2.1 Beyond Compare Action

The Beyond Compare uses the QuickCompare mode of Scooter Software's Beyond Compare to compare the contents of two files.



File 1: C:\Temp\File1.txt

File 2: D:\Temp\File1.txt

☐ Fail If Match

☑ Fail If Similar

☑ Fail If Mismatch

Compare Method: Rules-Based

Beyond Compare Version: Beyond Compare 3

Compare Two Files

Specify the path to each file.

Fail Action If

The action can be set to fail on any combination of matching, similar and mismatched files.

Compare Method

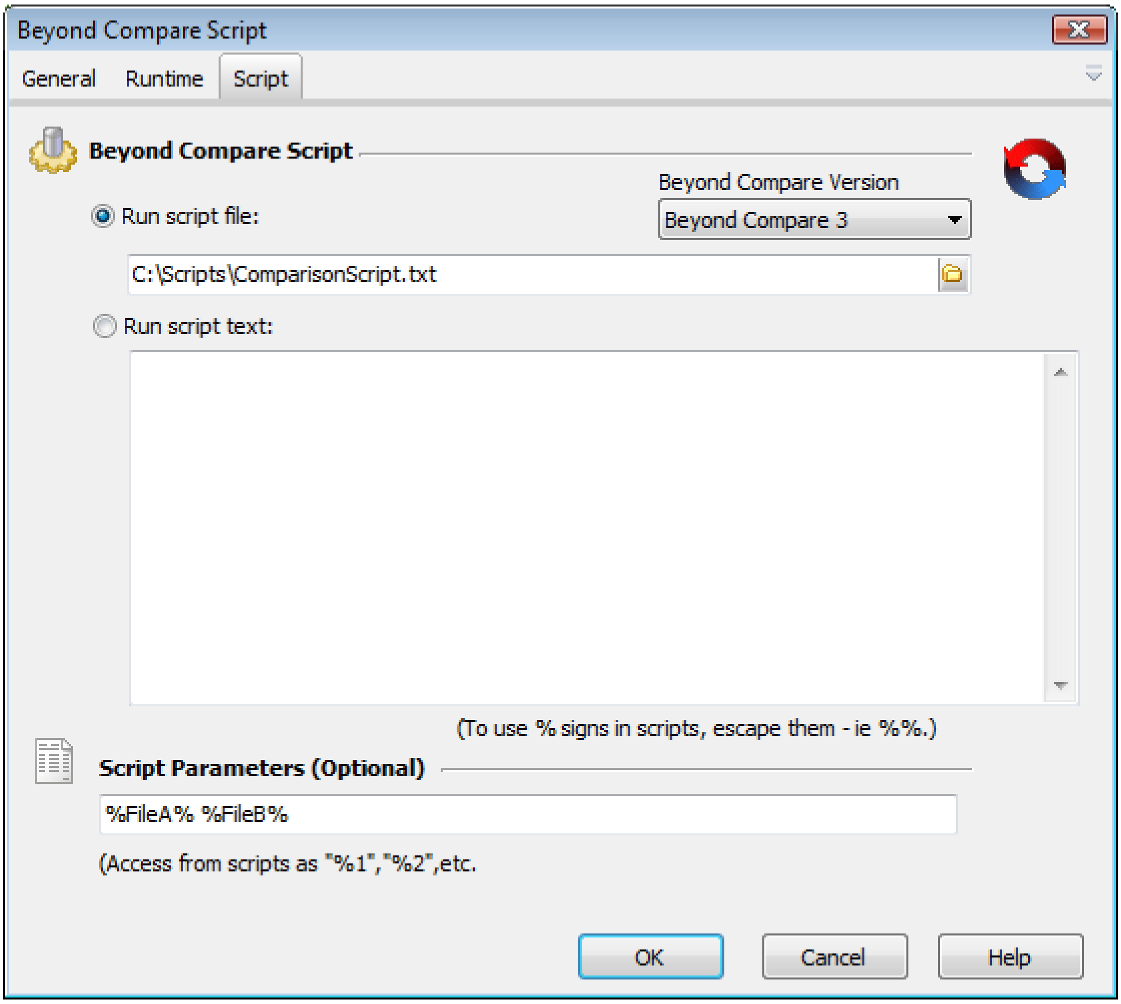
Comparisons can be made based on rules-based, size, CRC or full binary comparison.

See the Beyond Compare help file for the details of each comparison method.

6.9.12.2.2 Run Beyond Compare Script

The Run Beyond Compare Script action runs a script which has been created with the Beyond Compare scripting language. Among other things, this scripting language allows you to automate Folder Compare behaviour.

See the Beyond Compare Help for sample scripts and a Beyond Compare scripting reference.



Beyond Compare Version: Beyond Compare 3

● Run script file:

C:\Scripts\ComparisonScript.txt

○ Run script text:

(To use % signs in scripts, escape them - ie %%.)

Script Parameters (Optional)

%FileA% %FileB%

(Access from scripts as "%1","%2",etc.

Run script file

To run a script which has been saved in a file, enter the path of the file here.

Run Script Text

Alternatively, you can enter a script in the dialog shown. The script will be written out to a temporary file at runtime.

Note that you can use Automise variables in the script by using the % syntax. If you want to use % signs in the script itself, escape them as %%.

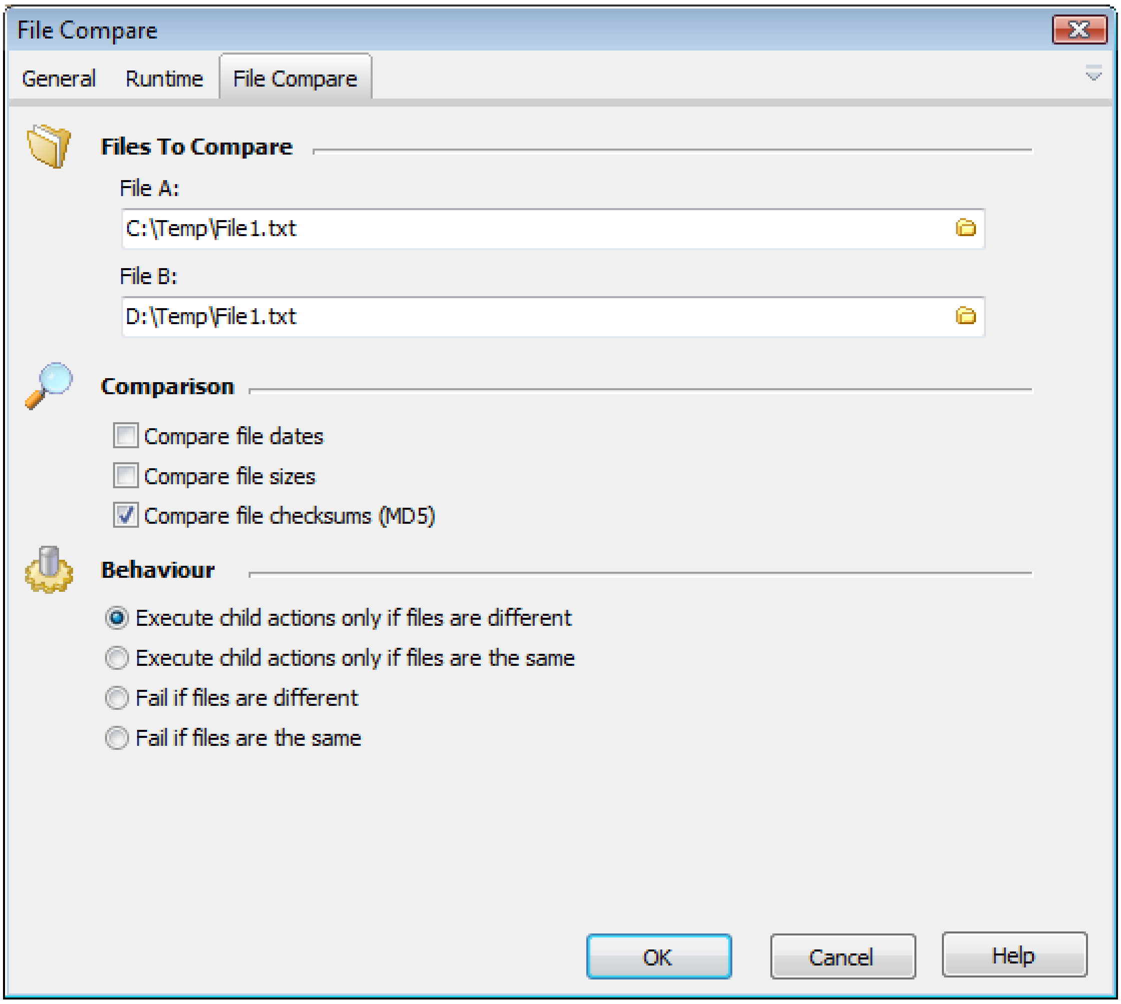
Script Parameters (Optional)

Enter extra command-line parameters for the BeyondCompare script. These are accessed within scripts as %1, %2, etc. (note that if using the "Run Script Text" option, these parameters will need to be double-escaped, ie %%1, %%2.)

6.9.12.3 File Compare Action

The File Compare action allows you to compare two files and use the result to influence the flow of your project.

For more advanced comparison options, you can use the (third party) Beyond Compare actions.



Files To Compare

File A:
C:\Temp\File1.txt

File B:
D:\Temp\File1.txt

Comparison

□ Compare file dates
□ Compare file sizes
☑ Compare file checksums (MD5)

Behaviour

● Execute child actions only if files are different
○ Execute child actions only if files are the same
○ Fail if files are different
○ Fail if files are the same

Files To Compare

Specify the file names of the files to compare.

Tip: Use File Iterators to compare multiple files.

Comparison

Select one or more methods by which to compare files. "Compare file checksums" is the most accurate.

"Compare file dates"

If this option is set, the action will compare file modification dates to determine if the files were modified at different times. Note that dates are considered different if they are more than a millisecond apart.

"Compare file sizes"

If this option is set, the action will compare file sizes to determine if the files are different. This option is much faster than calculating a full checksum, but also less accurate (it does not check file contents.)

"Compare file checksums (MD5)"

If this option is set, the action will generate an MD5 checksum to determine if the files are different. Although MD5 checksum collisions can occur, this option more or less guarantees whether or not the file contents are identical.

Behaviour

Choose how the result of the comparison will influence the flow of the project. You can choose to have the action only execute its child actions if the files are different/the same, or alternatively have the action fail outright if files are different/the same.

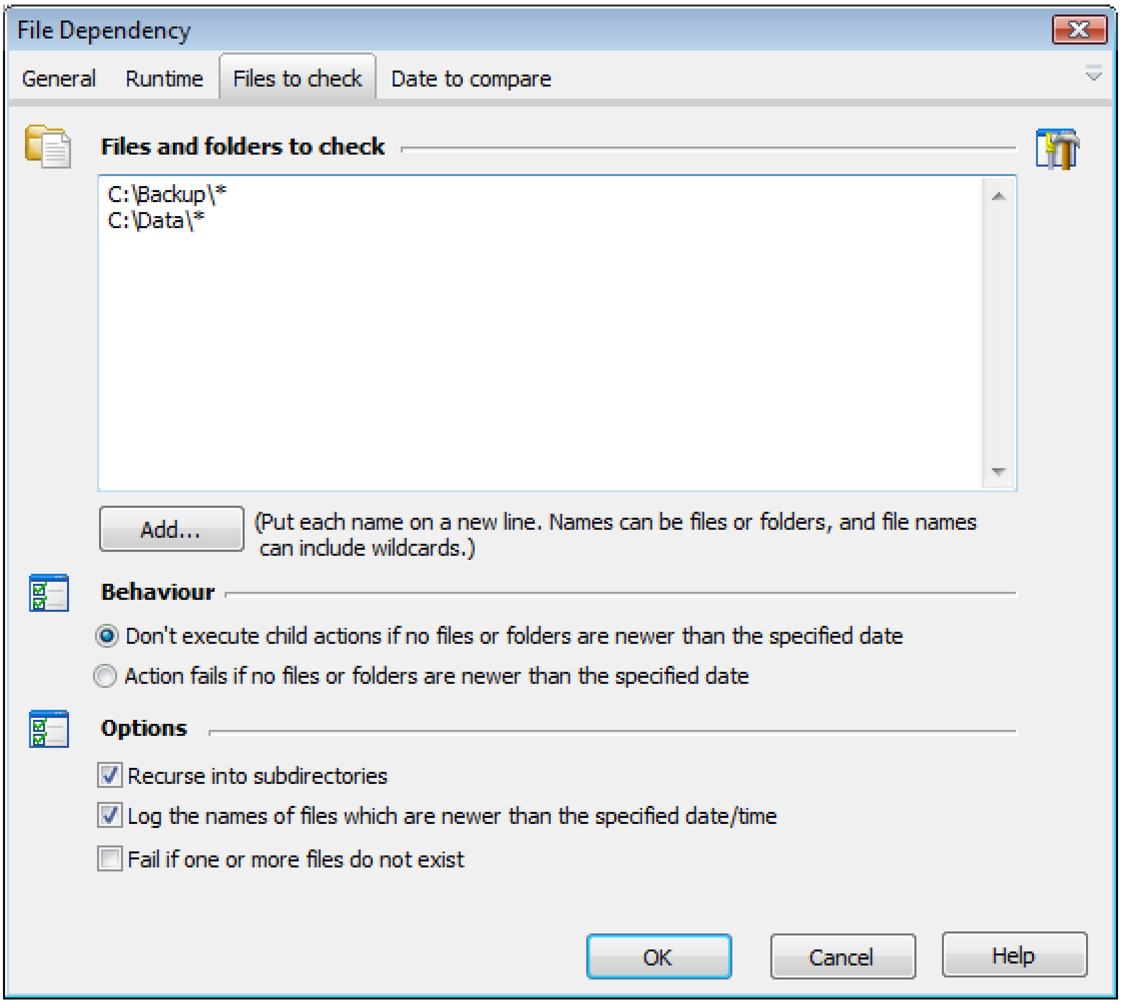
6.9.13 File Dependency Action

[Automise Professional Edition]

The file dependency action allows you to control the flow of your build based on whether or not a file (or set of files) has changed relative to a time stamp.

For example, you can choose to run a set of compiler actions only if the source files are newer than the compiled application.

In order to use a file dependency action, you have to choose
(a) the files and folders to check (the "source files") and
(b) the date timestamp to compare them to.



Files and folders to check

```
C:\Backup\*
C:\Data\*
```

(Put each name on a new line. Names can be files or folders, and file names can include wildcards.)

Behaviour

● Don't execute child actions if no files or folders are newer than the specified date

○ Action fails if no files or folders are newer than the specified date

Options

☑ Recurse into subdirectories

☑ Log the names of files which are newer than the specified date/time

□ Fail if one or more files do not exist

Files and Folders to Check

These are the files which will be compared against the time stamp (the source files, in the example above.) The action only executes its children if at least one of the files in this list is newer than the time stamp.

Files can be specified in the following ways:

Format	**Example**
Full Pathname	D:\ComponentA\MainClass.cs
Wildcard Filespec	D:\ComponentA\*.cs
Directory Pathname	D:\ComponentA\

Behaviour

If you choose "Don't execute child actions...", the child actions of the File Dependency actions will only be executed if one or more files is newer than the timestamp. Otherwise, Automise will skip to the next sibling action of the File Dependency.

If you choose "Action fails..." then the action will fail outright if none of the files is newer than the timestamp. This can lead to the build failing completely, or you can use a Try... Catch block to catch the error.

Options

"Recurse into subdirectories"

If you choose this option, then Automise will expand subdirectories of directory & wildcard pathnames that you specify. In the example from the screenshot, "Recurse into subdirectories" would return all ".cs" files from all subdirectories of the "ComponentA" directory.

"Recurse into subdirectories" does not recursively search for plain pathname file names.

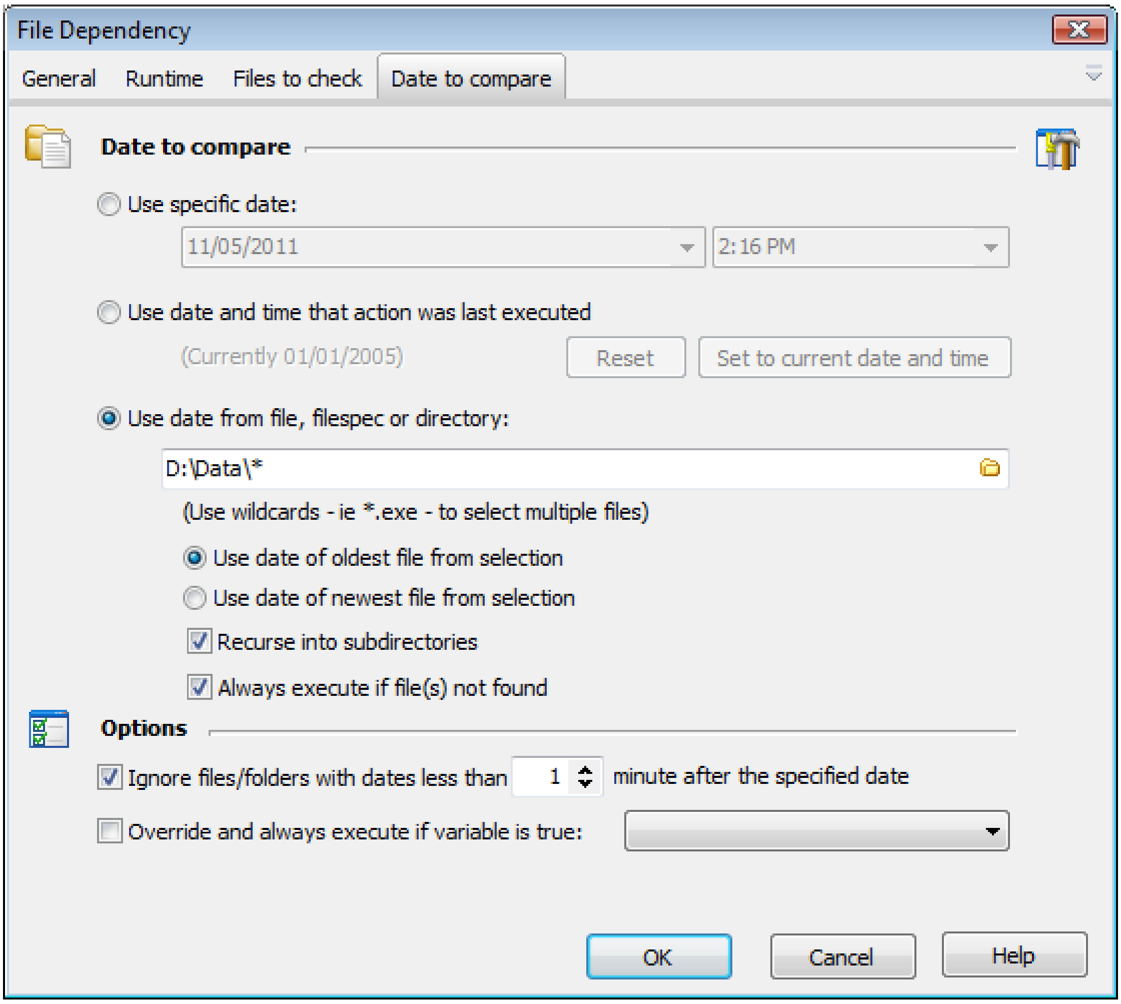
"Log the names of files which are newer than the specified date/time"

If you choose this option, then Automise will log the names of files which are newer than the timestamp.

"Fail if one or more files does not exist"

This option will cause the action to fail if any of the full pathnames specified do not exist.

The action will not fail if a wildcard filespec fails to match any files, or if a directory is empty. However, the action will fail (regardless of how this option is set) if none of the specified files can be found.



File Dependency

General | Runtime | Files to check | Date to compare

Date to compare

○ Use specific date: 11/05/2011 2:16 PM

○ Use date and time that action was last executed

(Currently 01/01/2005)

○ Use date from file, filespec or directory:

D:\Data\*

(Use wildcards - ie *.exe - to select multiple files)

● Use date of oldest file from selection

○ Use date of newest file from selection

☑ Recurse into subdirectories

☑ Always execute if file(s) not found

Options

☑ Ignore files/folders with dates less than 1 minute after the specified date

□ Override and always execute if variable is true:

Date to Compare

Specify the kind of timestamp you wish to use. This is the date of the "built files".

Use Specific Date

Enter a specific (static) date and time combination.

Use date and time that action was last executed

Automise keeps track of the last time that each File Dependency action has run.

Click the "Reset" button to set the last run time back to January 1, 2005.

Click the "Set to current date and time" to bring the last run time value up to the present moment.

Use date from file, filespec, or directory

As indicated, you can use a file (or group of files) to determine the timestamp.

Enter the full pathname to a file, a wildcard pathname (as in the screenshot), or a directory name. You can choose to have either the oldest or the newest file used for the timestamp.

Options

"Ignore files/folders with dates less than..."

If you select this option then the timestamp will be brought forward the specified number of minutes to account for "jitter" in the timestamps of object files, etc.

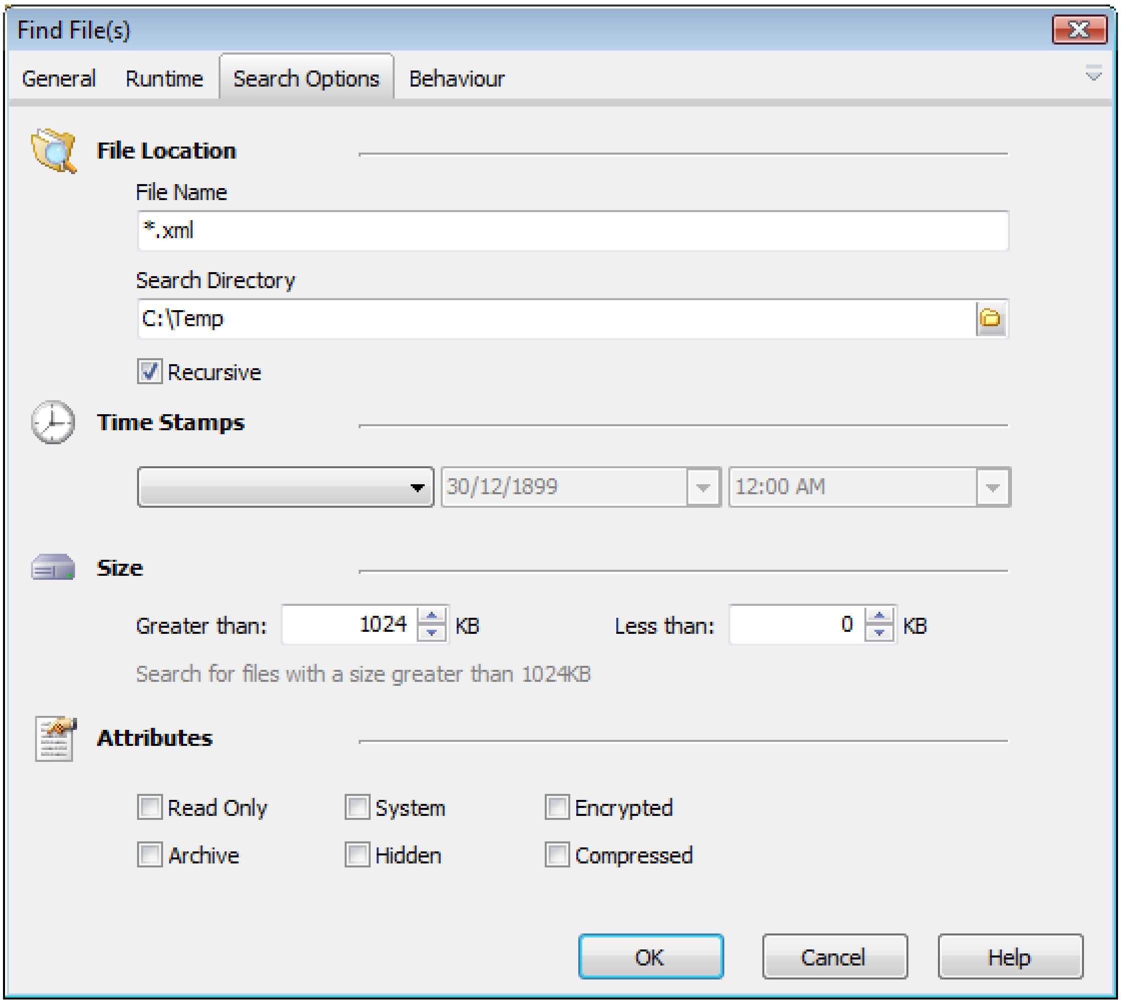
"Override and always execute if variable is true..."

If this option is selected, then the chosen Automise variable (which needs to be a boolean variable) is evaluated before file dependencies are calculated. If the variable evaluates to true then the dependency evaluation is skipped and the child actions are run automatically.

(This option is useful if you want to sometimes force your build to go ahead, regardless of timestamps.)

6.9.14 Find File(s)

The Find File(s) action enables you to search for files matching a specified search criteria.



Find File(s)

File Location

File Name: *.xml

Search Directory: C:\Temp

☑ Recursive

Time Stamps

30/12/1899 12:00 AM

Size

Greater than: 1024 KB Less than: 0 KB

Search for files with a size greater than 1024KB

Attributes

□ Read Only □ System □ Encrypted
□ Archive □ Hidden □ Compressed

File Name
The file spec used during the search. The value can contain the * or ? wildcard operators.

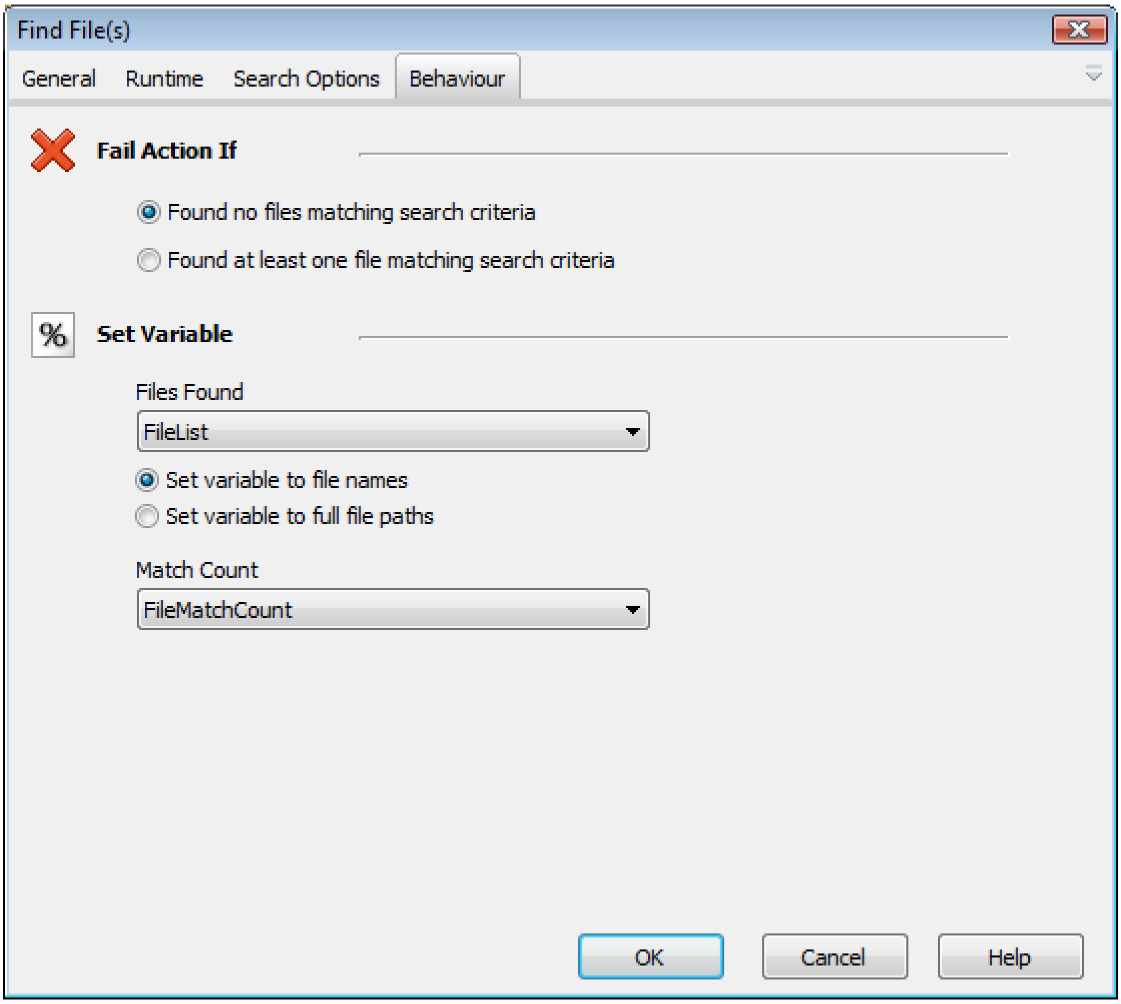
Search Directory
The directory to search in to find the required files. The * or ? wildcard operators can be used to recursively search a directory tree, but doing so may slow down the search dramatically.

Time Stamps
Enables searching of files matching the specified time stamps.

Size
Allows files to be found matching a certain size.

Attributes
Allows file to be found based on the attributes set.

Fail Action If
Alters the behaviour of the action regarding the success or failure when files are found.



Find File(s)

Fail Action If

● Found no files matching search criteria

○ Found at least one file matching search criteria

Set Variable

Files Found

FileList

● Set variable to file names

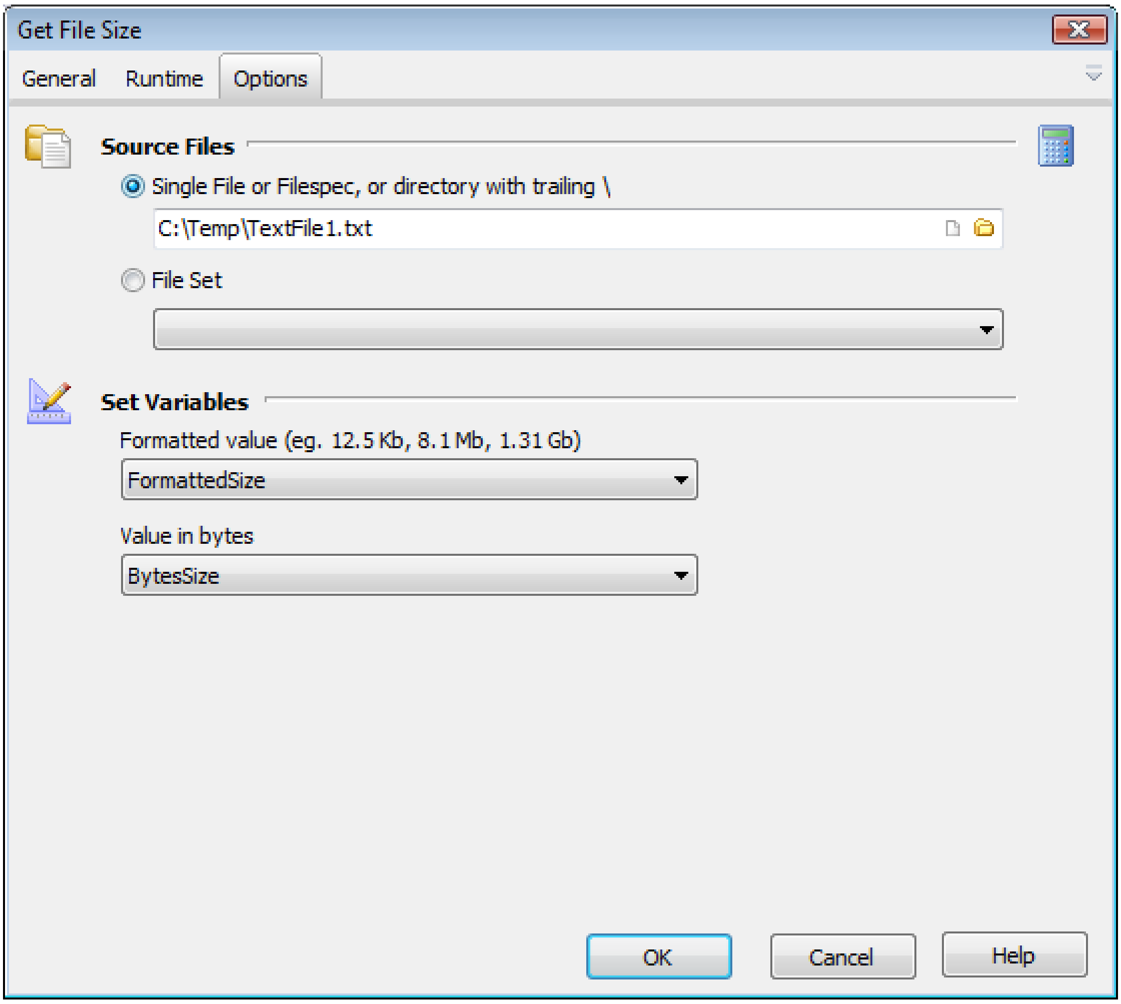
○ Set variable to full file paths

Match Count

FileMatchCount

6.9.15 Get File Size

The Get File Size action gets the size of one or more files, or the size of an entire directory into a variable.



Source Files

● Single File or Filespec, or directory with trailing \

C:\Temp\TextFile1.txt

○ File Set

Set Variables

Formatted value (eg. 12.5 Kb, 8.1 Mb, 1.31 Gb)

FormattedSize

Value in bytes

BytesSize

Source Files

Single File or Filespec, or directory - you can specify the path to a single file (eg. c:\temp\myfile.txt), or use a file specification with wildcards (eg. c:\Images\*.png). If you want to get the size of an entire directory, specify the directory name with a trailing backslash \

File Set - Specify the name of the FileSet

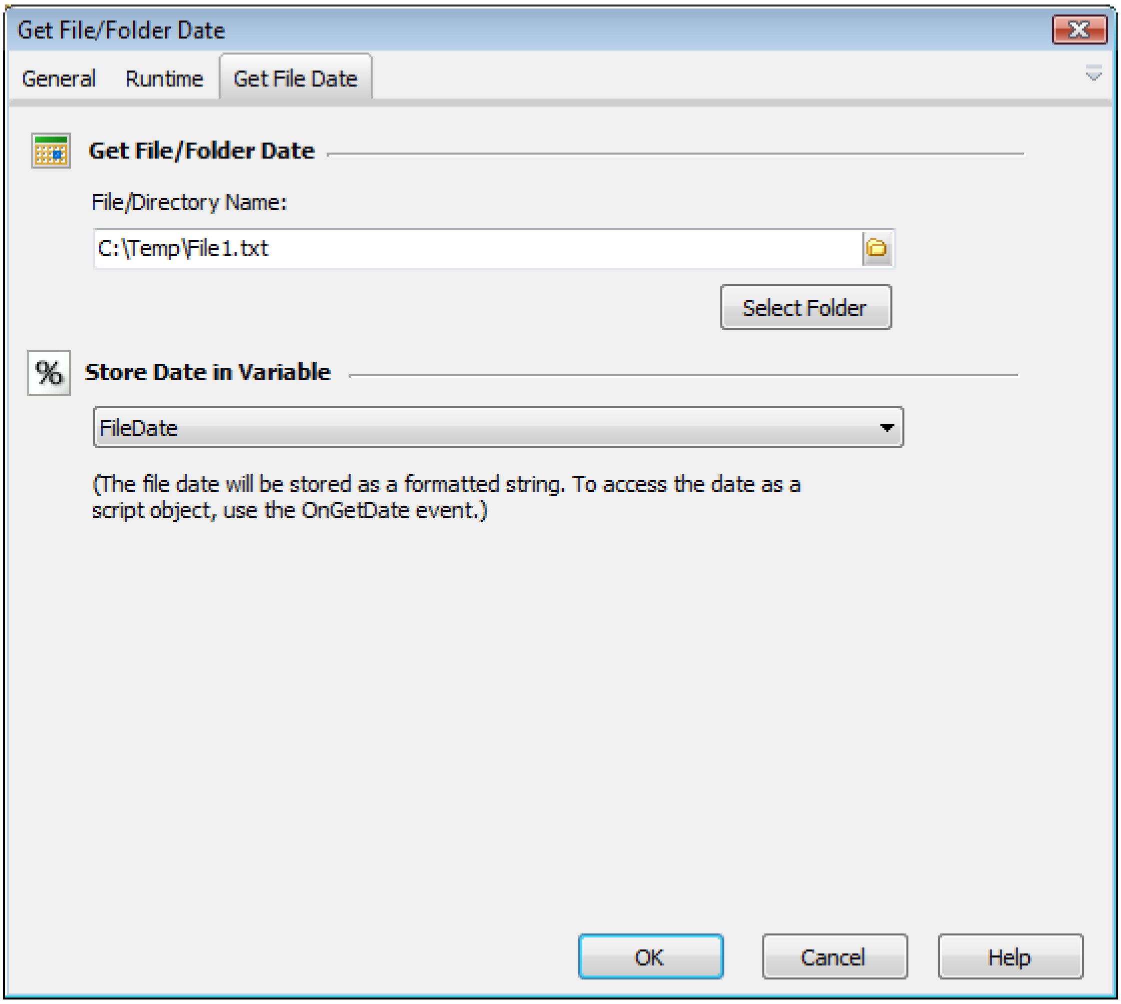
Set Variables

Formatted value - the total size will be formatted in human readable form, such as 12.5 Kb, 8.1 Mb, or 1.31 Gb

Value in bytes - the total size will be saved to the selected variable in bytes

6.9.16 Get File/Folder Date

This action will get the Modified Date of the specified file or folder and put it in a variable.



File/Directory Name:

C:\Temp\File1.txt

FileDate

(The file date will be stored as a formatted string. To access the date as a script object, use the OnGetDate event.)

File/Directory Name

Specify the file or folder name to retrieve the modification date. The action will fail if the file does not exist.

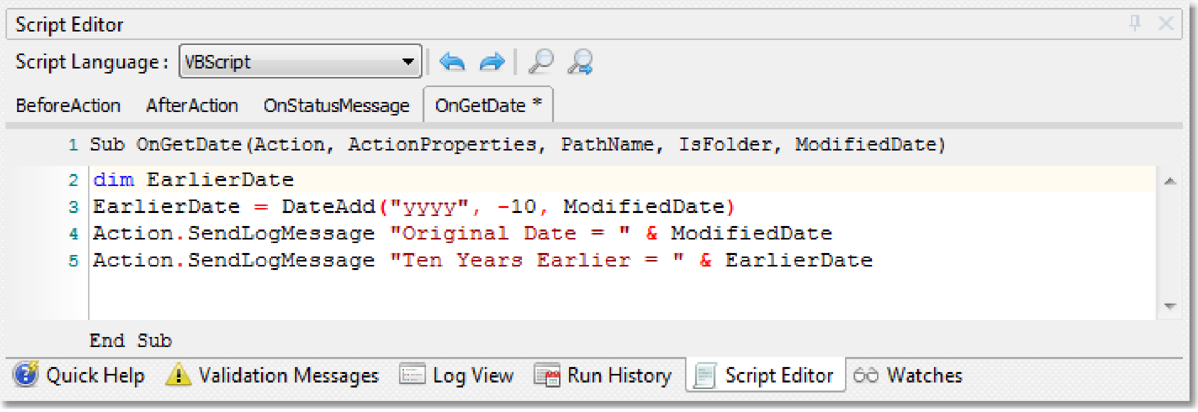
Store Date in Variable

Optionally, choose a variable to store the date in. The date will be stored as a formatted date/time string.

Modification Date As Object

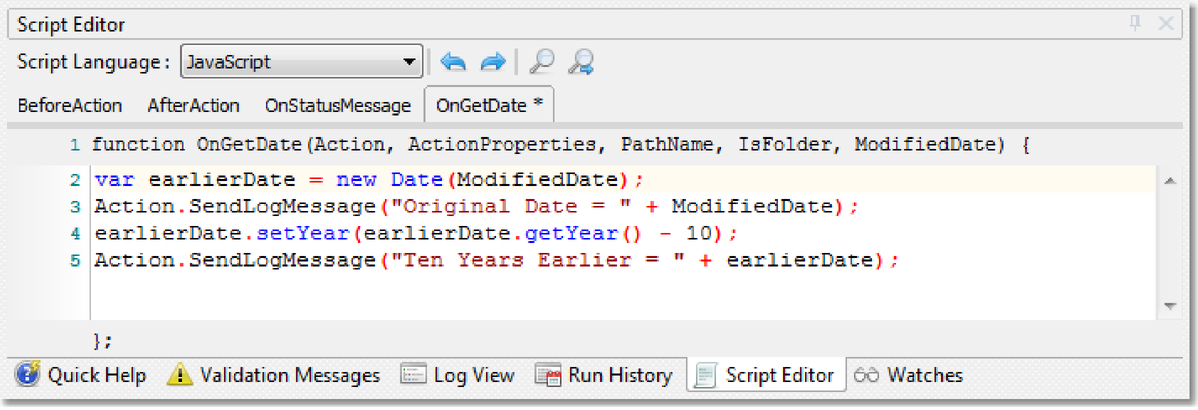
If you wish to access the date as an object, you can use the OnGetDate script event.

For VBScript, you can use Date functions directly to operate on the ModifiedDate parameter:



```
Sub OnGetDate(Action, ActionProperties, PathName, IsFolder, ModifiedDate)
dim EarlierDate
EarlierDate = DateAdd("yyyy", -10, ModifiedDate)
Action.SendLogMessage "Original Date = " & ModifiedDate
Action.SendLogMessage "Ten Years Earlier = " & EarlierDate

End Sub
```

For JavaScript, you need to create a new JavaScript Date object to wrap the ModifiedDate parameter:

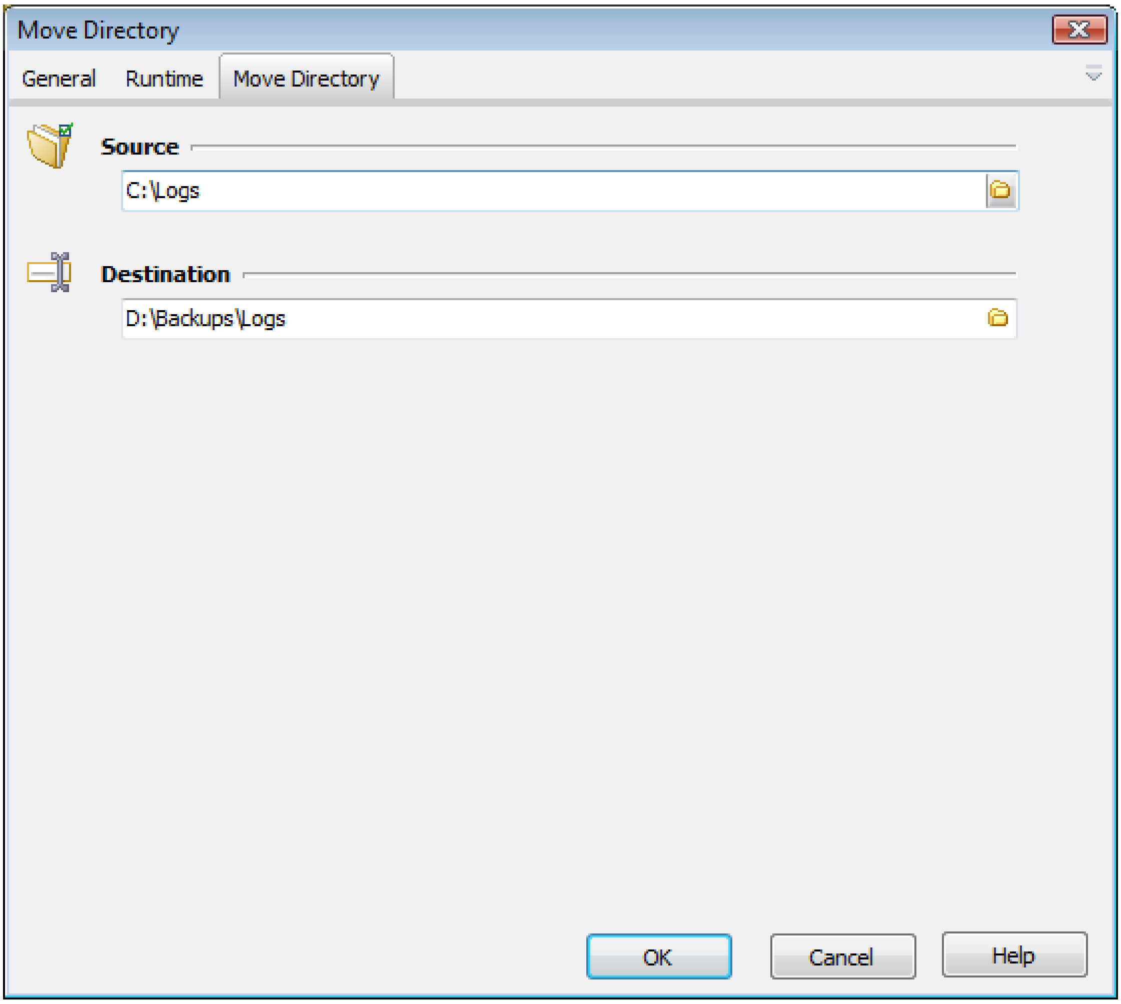


```
function OnGetDate(Action, ActionProperties, PathName, IsFolder, ModifiedDate) {
var earlierDate = new Date(ModifiedDate);
Action.SendLogMessage("Original Date = " + ModifiedDate);
earlierDate.setYear(earlierDate.getYear() - 10);
Action.SendLogMessage("Ten Years Earlier = " + earlierDate);

};
```

6.9.17 Move Directory Action

This action will move a folder and its contents to a new name.

Note that the **Destination** is the new complete name for the folder, so to move c:\test to c:\temp\test, specify c:\temp\test as the destination directory.



Source: C:\Logs

Destination: D:\Backups\Logs

Scripting Info

The Action properties available are:

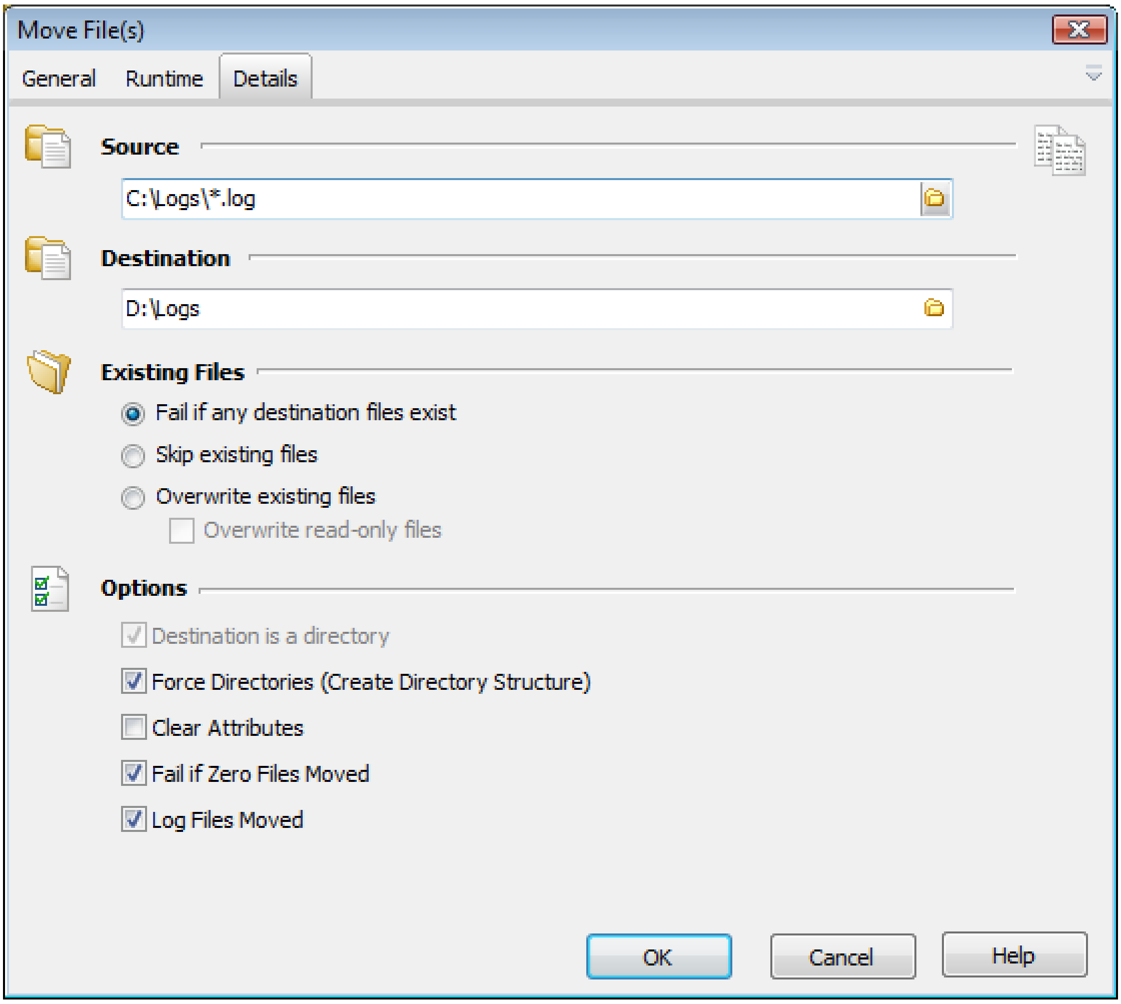
property Directory : String; // the directory you wish to move.
property DestinationDirectory : String; // the new path for the directory.

6.9.18 Move File(s) Action

The Move File(s) action will move a File or a set of files that match a file spec to the destination file or directory. You can use wildcards in the source setting to move more than one file (the "Destination is a directory" property will be set automatically). If the target directories do not already exist, check the Force Directories property to make Automise create the directories as needed.

If any file fails to move, then the action will fail.

The Copy/Move FileSet action provides more sophisticated support for choosing files to move.



Move File(s)

General | Runtime | Details

Source

C:\Logs\*.log

Destination

D:\Logs

Existing Files

● Fail if any destination files exist
○ Skip existing files
○ Overwrite existing files
□ Overwrite read-only files

Options

☑ Destination is a directory
☑ Force Directories (Create Directory Structure)
□ Clear Attributes
☑ Fail if Zero Files Moved
☑ Log Files Moved

Existing Files

If a file already exists at the destination, select how you would like the action to behave - fail, skip or overwrite.

Destination is a directory - Set this option to indicate that the destination is a directory name, not a file name.

Force Directories - If this option is set and the destination directory, or any parent directories, do not exist, then they will be created.

Clear Attributes - If this option is set, the destination file will be created with all attributes (Archive, Read Only, Hidden, System) cleared.

Fail if Zero Files Moved - If this option is set and no files are moved, the action will fail.

Log Files Moved - Each file moved will be logged. If this option is not set, only skipped files or files with errors are logged.

Scripting Info

The Action properties available are :

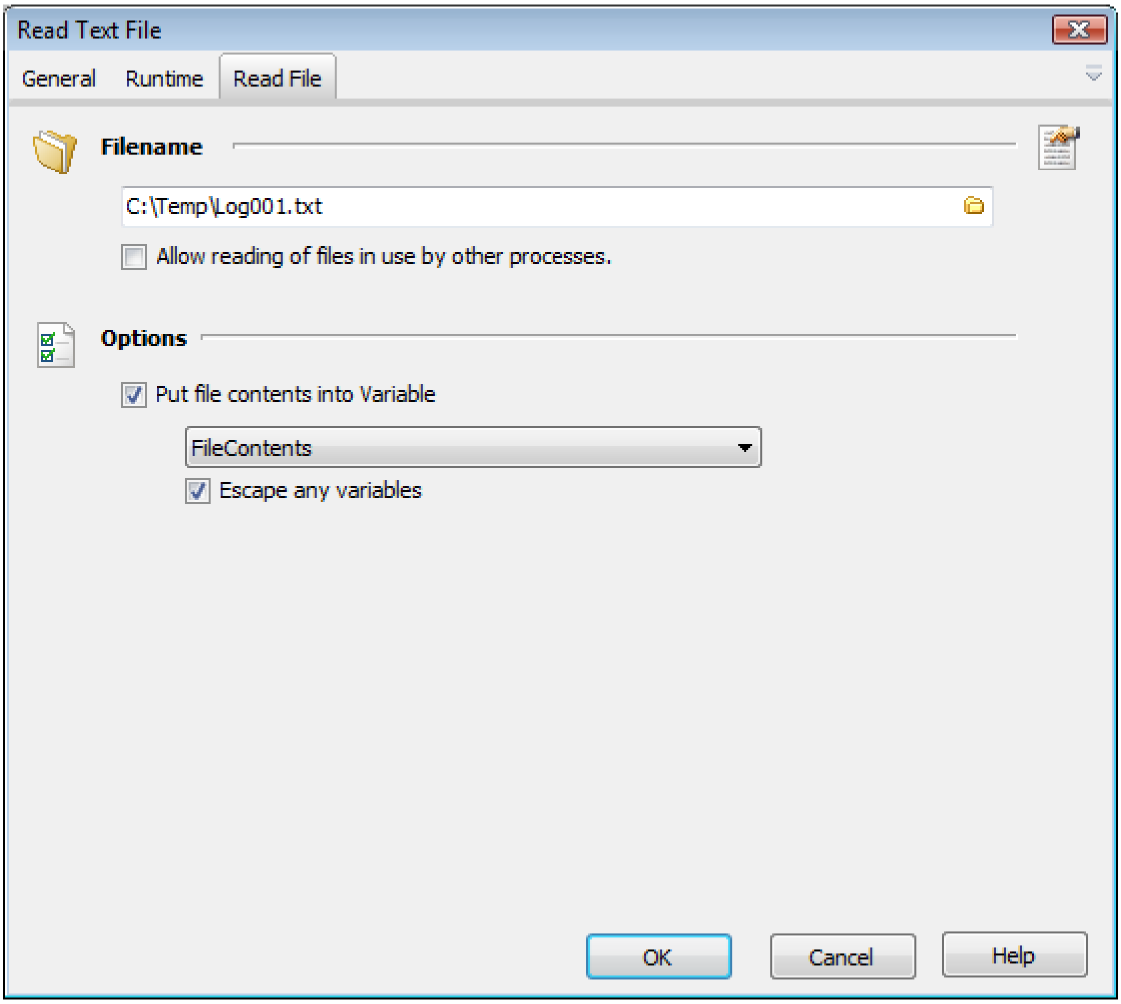
property FileSpec :WideString; // the file specification for the files to move. You

may use Wildcards.
property FilesAffected : integer ; // read only, the number of files Moved
property Target :WideString; // The target file or directory
property Force : WordBool; //Use Force Directories to create a directory and all parent directories that do not already exist.
property TargetIsDir : WordBool;// Specifies that the Target Is a Directory
property FailIfExists : WordBool;// Fail if the target file Exists
property Overwrite : WordBool; // Overwrite existing files
property OverwriteReadOnly : WordBool;//Overwrite Read Only Files
property ClearAttributes : WordBool // clear the moved file's attributes.
property FailIfZeroFiles : WordBool;
property LogFilesAffected : WordBool;

6.9.19 Read Text File

The Read Text File action enables you to automate the reading of a text file.

You can read the text file into a Automise variable, and also process each line in the OnReadLine event.



Filename

C:\Temp\Log001.txt

□ Allow reading of files in use by other processes.

Options

☑ Put file contents into Variable

FileContents

☑ Escape any variables

Filename

Specify the name of the text file to read.

Put file contents into variable

Specify this option (and a variable) so that you can examine and manipulate the contents of the file in subsequent actions.

When the action executes, the "**OnReadLine**" event is called for each line of the text file which makes it easy to process the file line-by-line in script.

Escape any variables

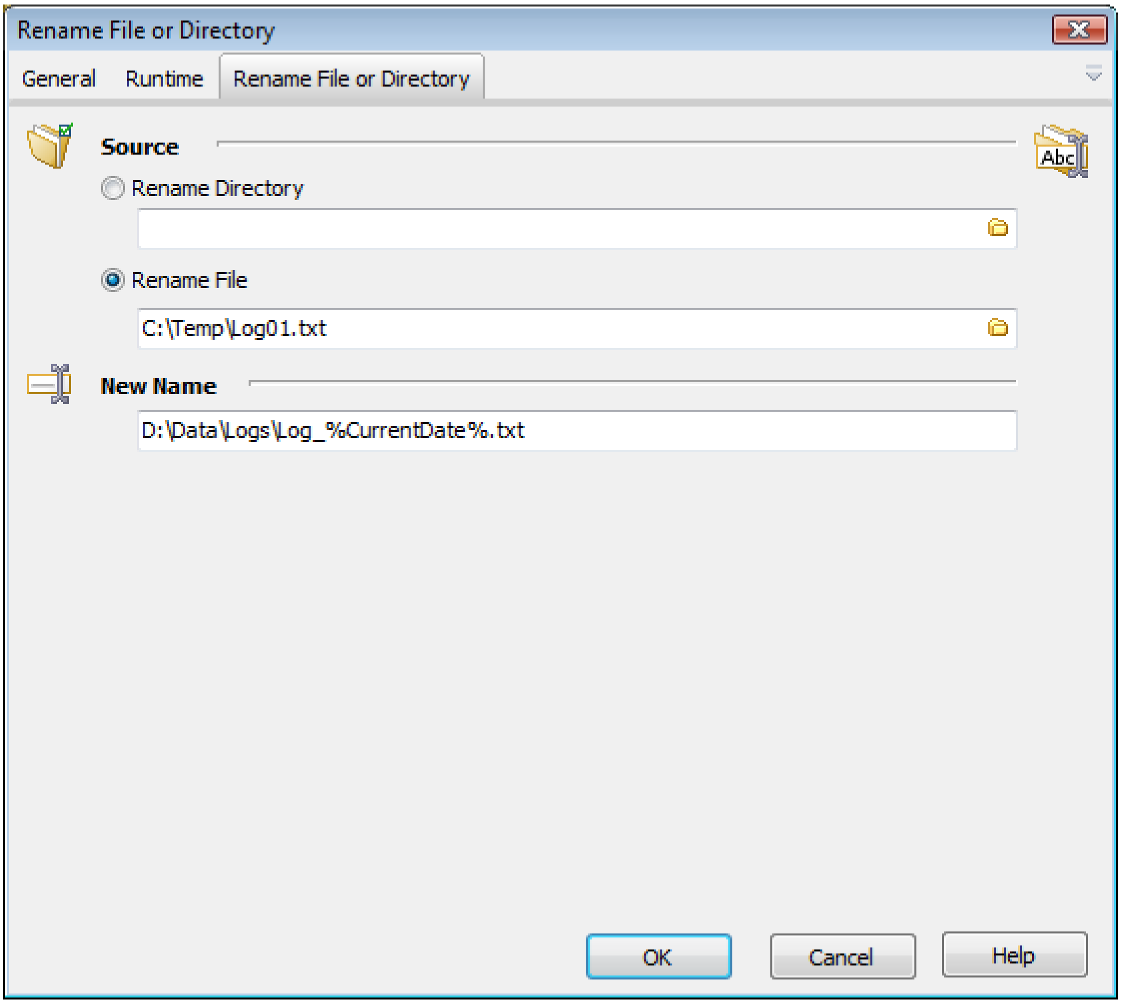
Specify this option to escape any variable references in the source file.

For example, if the source contains % signs then they will be replaced as %%. This means that you can safely use an action which expands variable references in the variable contents, without errors.

Also see Escaping Variable References

6.9.20 Rename File or Directory Action

This action allows you to rename a File or a Directory.



○ Rename Directory

● Rename File

C:\Temp\Log01.txt

New Name

D:\Data\Logs\Log_%CurrentDate%.txt

Rename Directory - select this to rename a directory and enter the existing directory to rename

Rename File - select this to rename a file and enter the existing filename to rename

New Name - the new name of the file or directory (do not specify the path)

6.9.21 RoboCopy

The RoboCopy action enables you to automate file and directory copying using RoboCopy. Robocopy is a very powerful tool for copying, moving, and synchronising directories and files with error recovery. It is included in the Windows Resource Kit (downloadable from Microsoft for your specific Windows version.)

The RoboCopy actions are:
Robocopy - perform file and directory copy operations
Robocopy Job - run Robocopy Jobs
Robocopy Move - perform file and directory move operations
Robocopy Mirror - synchronise source directory to destination directory

For more information please see Robocopy at Microsoft Technet.

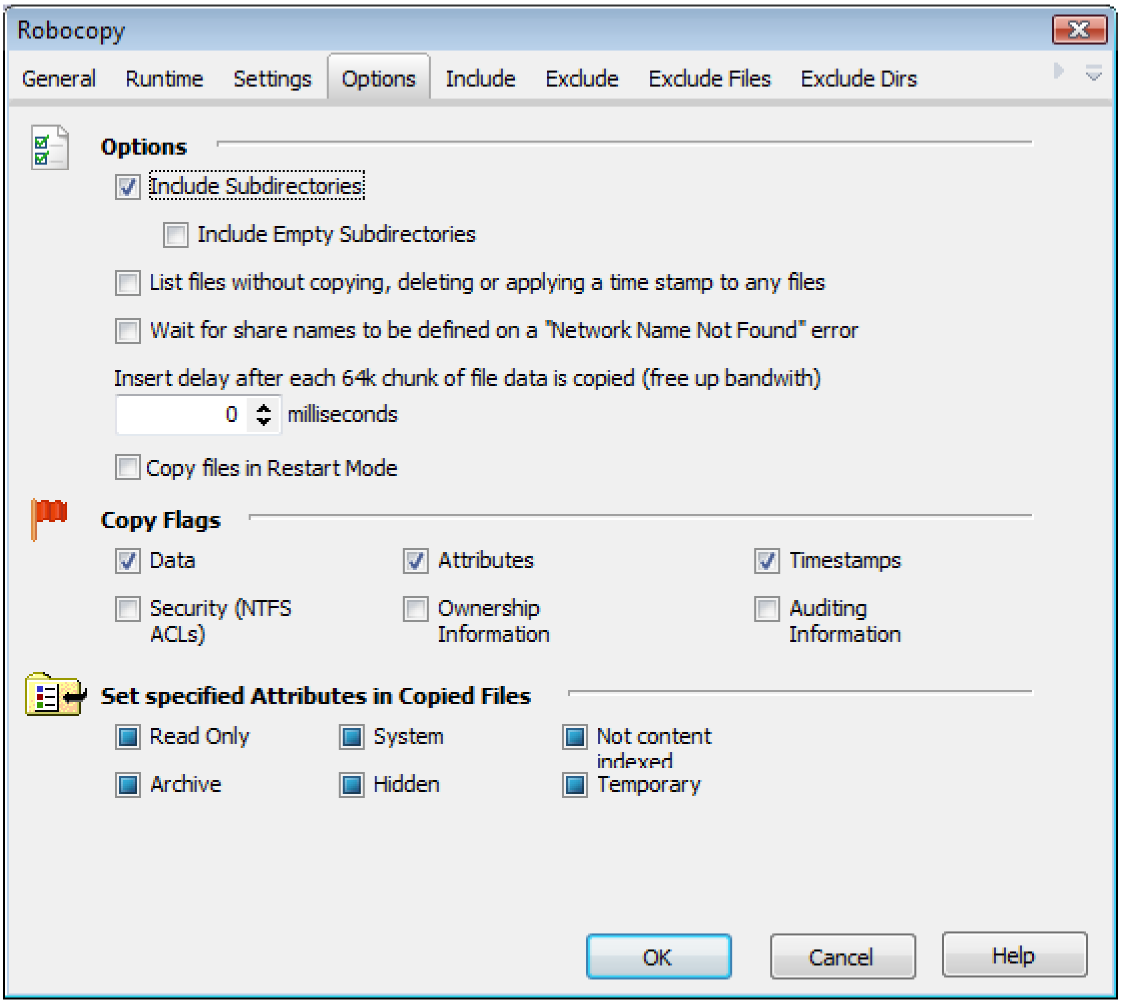
6.9.21.1 Robocopy

The Robocopy action allows you to perform advanced file and directory copy operations using Microsoft's Robocopy.

See the Robocopy Common Pages section for help on the following pages:

- Robocopy Settings
- Robocopy Include
- Robocopy Exclude
- Robocopy Exclude Files
- Robocopy Exclude Dirs
- Robocopy Logging

Options Page



Options

☑ Include Subdirectories

☐ Include Empty Subdirectories

☐ List files without copying, deleting or applying a time stamp to any files

☐ Wait for share names to be defined on a "Network Name Not Found" error

Insert delay after each 64k chunk of file data is copied (free up bandwith)

0 milliseconds

☐ Copy files in Restart Mode

Copy Flags

☑ Data ☑ Attributes ☑ Timestamps

☐ Security (NTFS ACLs) ☐ Ownership Information ☐ Auditing Information

Set specified Attributes in Copied Files

▣ Read Only ▣ System ▣ Not content indexed

▣ Archive ▣ Hidden ▣ Temporary

Options

Include Subdirectories - Perform a recursive copy

Include Empty Subdirectories - By default empty directories are not copied as part of the copy operation, enabling this option will include empty directories.

List files without copying, deleting or applying a time stamp to any files -

Enabling this option provides a preview of the operation by listing all the files and directories that will be copied, without actually performing the copy.

Wait for share names to be defined on a "Network Name Not Found" error - This option will allow the copy operation to retry if a network share required for the operation is not found.

Insert delay after each 64k chunk of file data is copied (free up bandwidth) - This option can be used to free up bandwidth when copying over a slow network connection by inserting a delay between sending data. Use the control provided to specify the delay (in milliseconds).

Copy files in Restart Mode - Enabling this option allows Robocopy to resume copying from the point of failure rather than starting from scratch, should the copy fail.

Copy Flags - Use the Copy Flags options to set which attributes of the files should be included in the copy information.

Set specified Attributes in Copied Files - Set the attributes of the copied files, by default attributes are copied from source.

6.9.21.2 Robocopy Mirror

The Robocopy Mirror action allows you to perform file and directory mirroring using Microsoft's Robocopy.

This action allows you to mirror the contents of one directory to another. This action should be used with caution as the mirror operation will delete files from the destination that no longer exist in the source directory.

See the Robocopy Common Pages section for help on the following pages:
- Robocopy Settings
- Robocopy Include
- Robocopy Exclude
- Robocopy Exclude Files
- Robocopy Exclude Dirs
- Robocopy Logging

6.9.21.3 Robocopy Move

The Robocopy Move action allows you to move files and directories using Microsoft's Robocopy.

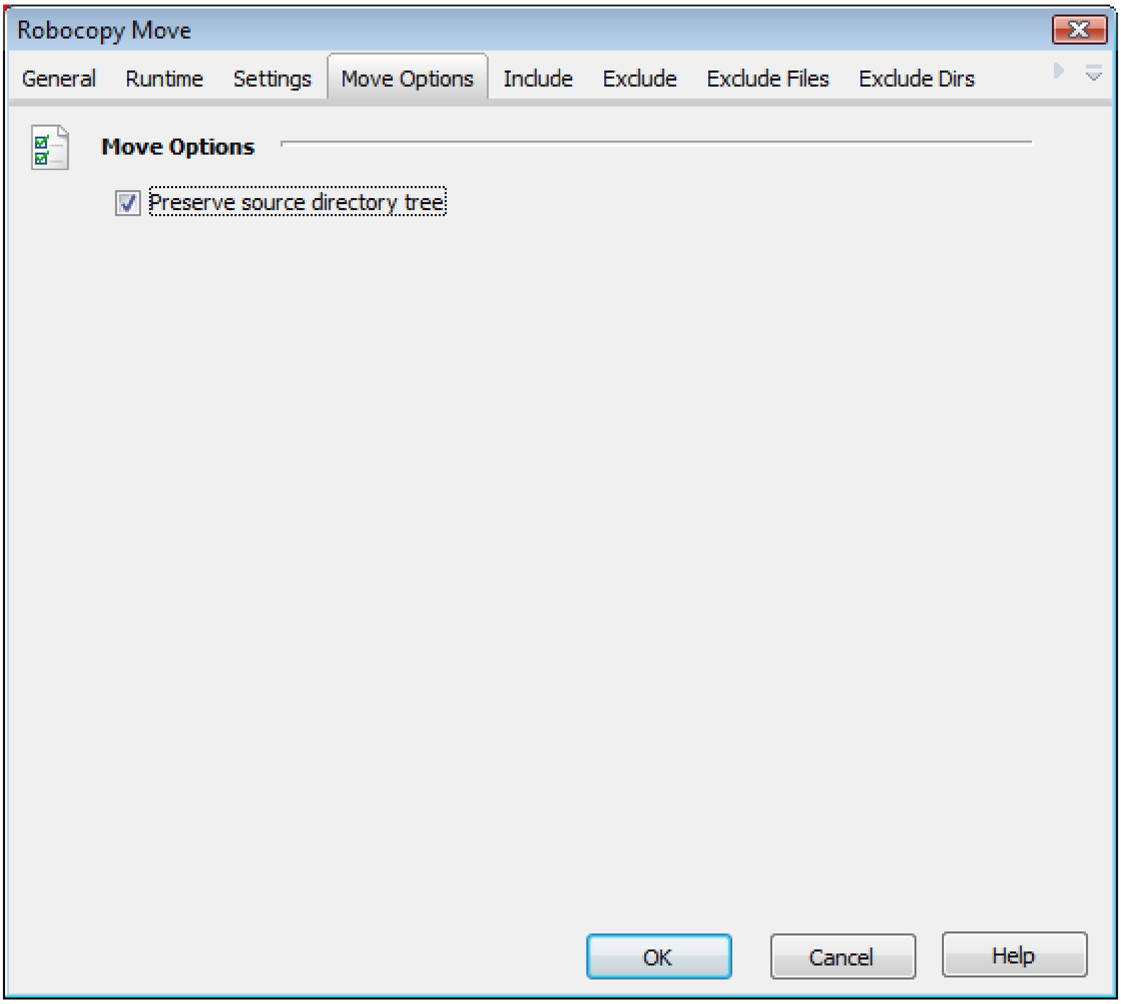
This action allows you to move the contents of one directory to another.

See the Robocopy Common Pages section for help on the following pages:
- Robocopy Settings
- Robocopy Include
- Robocopy Exclude
- Robocopy Exclude Files

- Robocopy Exclude Dirs
- Robocopy Logging

Move Options Page

Preserve source directory tree - Enabling this option moves both files and directories.



☑ Preserve source directory tree

6.9.21.4 Run Robocopy Job

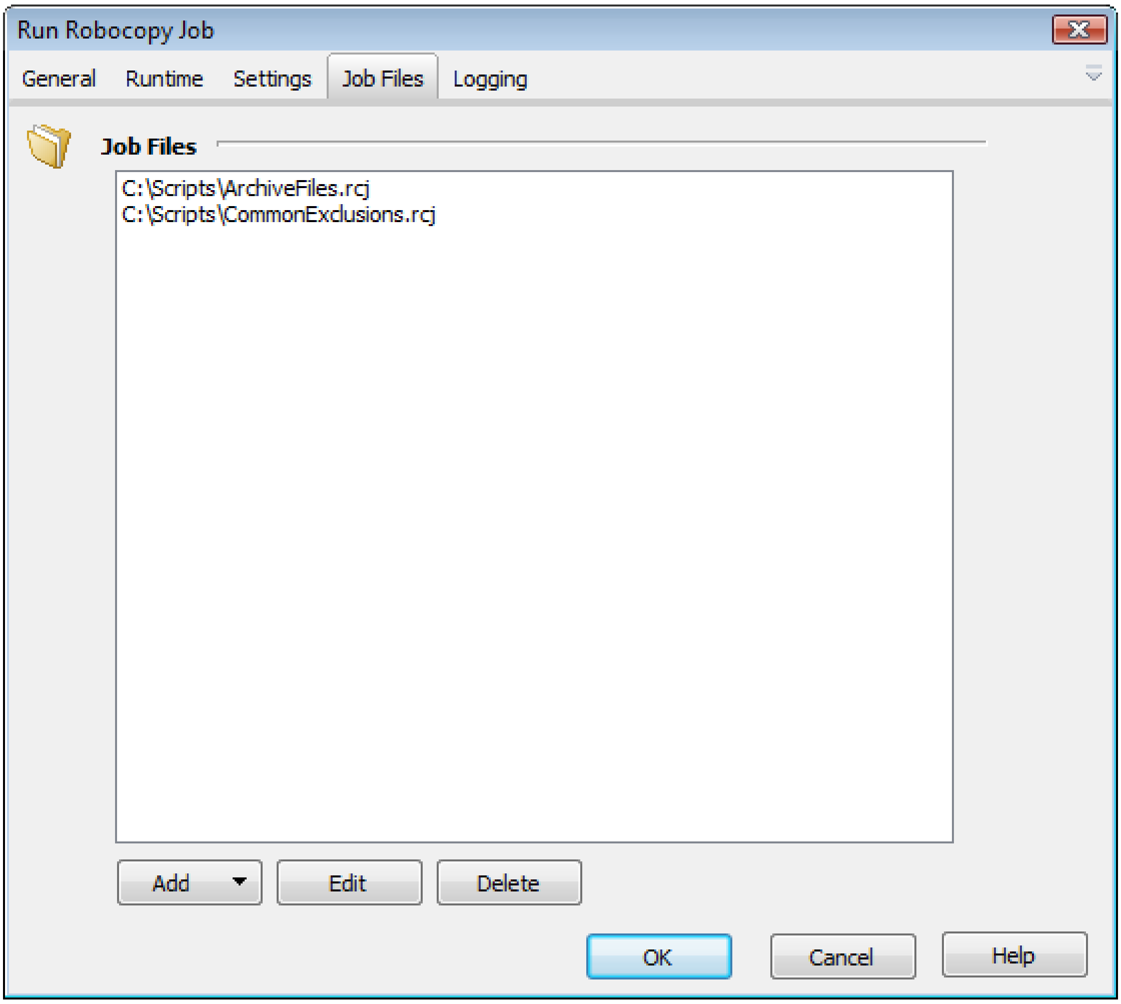
The Run Robocopy Job action allows you to pass in Robocopy Job Files (.rcj) that contain the parameters for the Robocopy operation.

See the Robocopy Common Pages section for help on the following pages:

- Robocopy Settings
- Robocopy Logging

Job Files

The Job Files page allows you to list the job files that you want to execute.



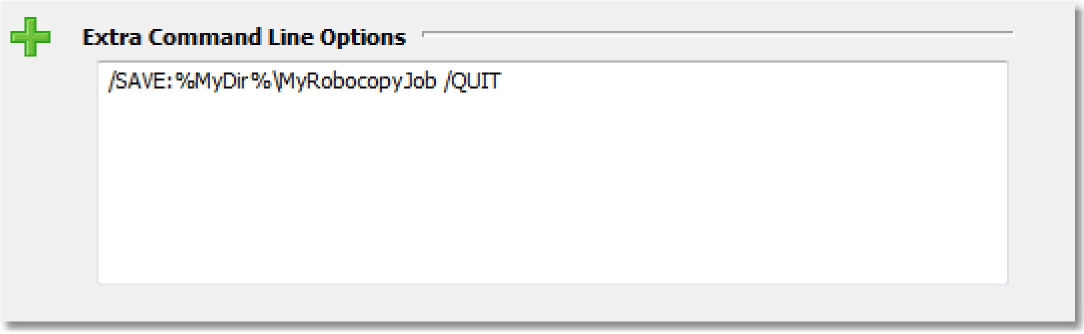
C:\Scripts\ArchiveFiles.rcj
C:\Scripts\CommonExclusions.rcj

About Robocopy Jobs

You can create a Robocopy Job file by using the */SAVE:<filename>* switch when running a Robocopy command. When using the save switch to create the job file only the switches that preceed the save switch will be saved to the job file (any switches that occur after the save switch will be omitted from the job file).

If you want to create a job file without actually running the Robocopy operation place the */QUIT* switch directly after the save switch. This will cause the operation to abort directly after creating the job file.

This can be done from Automise by using the desired action and adding the /SAVE:<filename> and /QUIT switches in the extra command line arguments field.



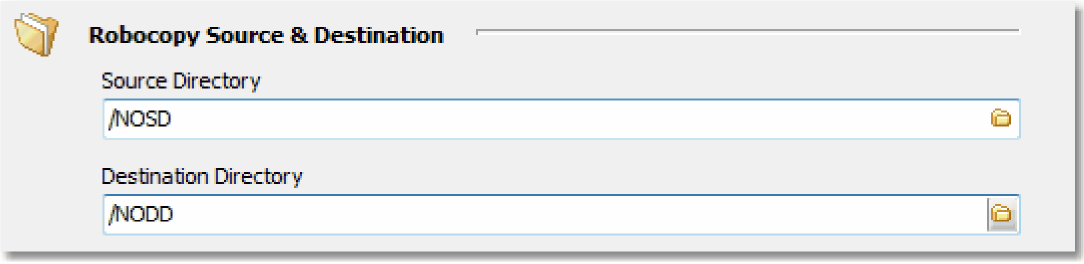
/SAVE:%MyDir%\MyRobocopyJob /QUIT

Creating a Robocopy job file from a Automise action.

When using Robocopy Job files the commands stored in the files will be executed as a single operation (not an operation per job file). The intention here is that you may use job files to store parameters that are common to a number of Robocopy operations that you intend to perform. For instance if you use the same set of exclusion rules each time you perform a Robocopy oepration you may want to store these rules in a job file and then use this job each time you use Robocopy.

You need to take care not to specify the source or destination directory more than once. If your job file passes the */SD* switch (which is the source directory switch, the same goes for the */DD* destination directory switch) can only be specified once in the set of jobs. If your set of job files contains either the */SD* or the */DD* switch then you cannot pass in the source directory or destination directory via the action.

If you want to use the directories specified in your job file then you would need to specify the following for the Source and Destination respectively, */NOSD* and */NODD.* These arguments stand for No Source Directory and No Destination Directory (which instruct Robocopy to use the values from within the job file). If you want to pass your source and directory in via the action then you need to remove the source directory and destination directory switches from the script (or at least comment them out).



Source Directory
/NOSD

Destination Directory
/NODD

Running a Robocopy job where the source and destination are specified in the file requires the source and destination to be specified as /NOSD and /NODD respectively.

This also allows you to create a job file that uses a common source for instance (where the source directory is specified in the job file using the */SD* switch) where you pass the */NOSD* switch as the source directory in the action and then specify the destination directory.

Sample Job File

The text below is a sample of what a a Robocopy Job folder. A comment is started by a double semi-colon (::) and can be placed after a switch. As you can see each command line switch is placed on a new line.

```
:: Sample Robocopy Job File
```

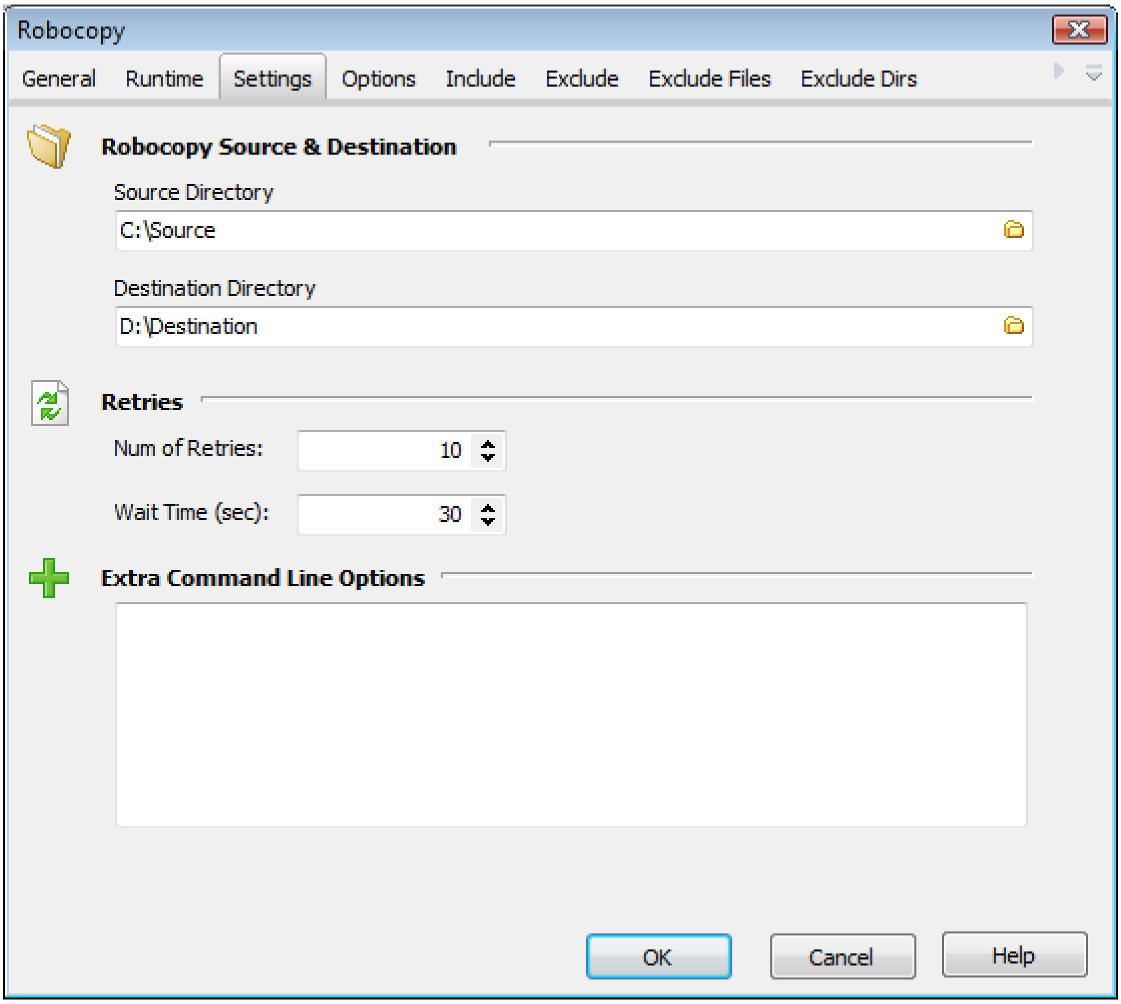
```
::
:: The next line specifies the source directory for the operation
   /SD:C:\My Files\

:: The next line is the destination directory for the operation
   /DD:D:\My Backups\

:: Copy Options
   /S       :: This switch will copy subdirectories but not empty sub
directories
   /Z       :: This switch will perform the operation in restartable mode

:: Excluded Files
   /XF
      Readme.txt  :: Any file called 'Readme.txt' will not be included in
the operation
      *.tmp       :: Any file with the extension .tmp will be excluded
```

6.9.21.5 Robocopy Common Pages

The following pages are shared by the Robocopy actions.

Settings Page

The settings page allows you to set the basic details of the Robocopy operation including the source and destination for the operation.



Robocopy

Robocopy Source & Destination

Source Directory: C:\Source

Destination Directory: D:\Destination

Retries

Num of Retries: 10

Wait Time (sec): 30

Extra Command Line Options

Robocopy Source & Destination

Source Directory - This is the directory that contains the files that you want to copy. Do not specify a file or file mask here.

Destination Directory - This is the directory that will be the destination for the copied files.

Retries

Num of Retries - This allows you to specify the number of times to retry failed copied attempts.

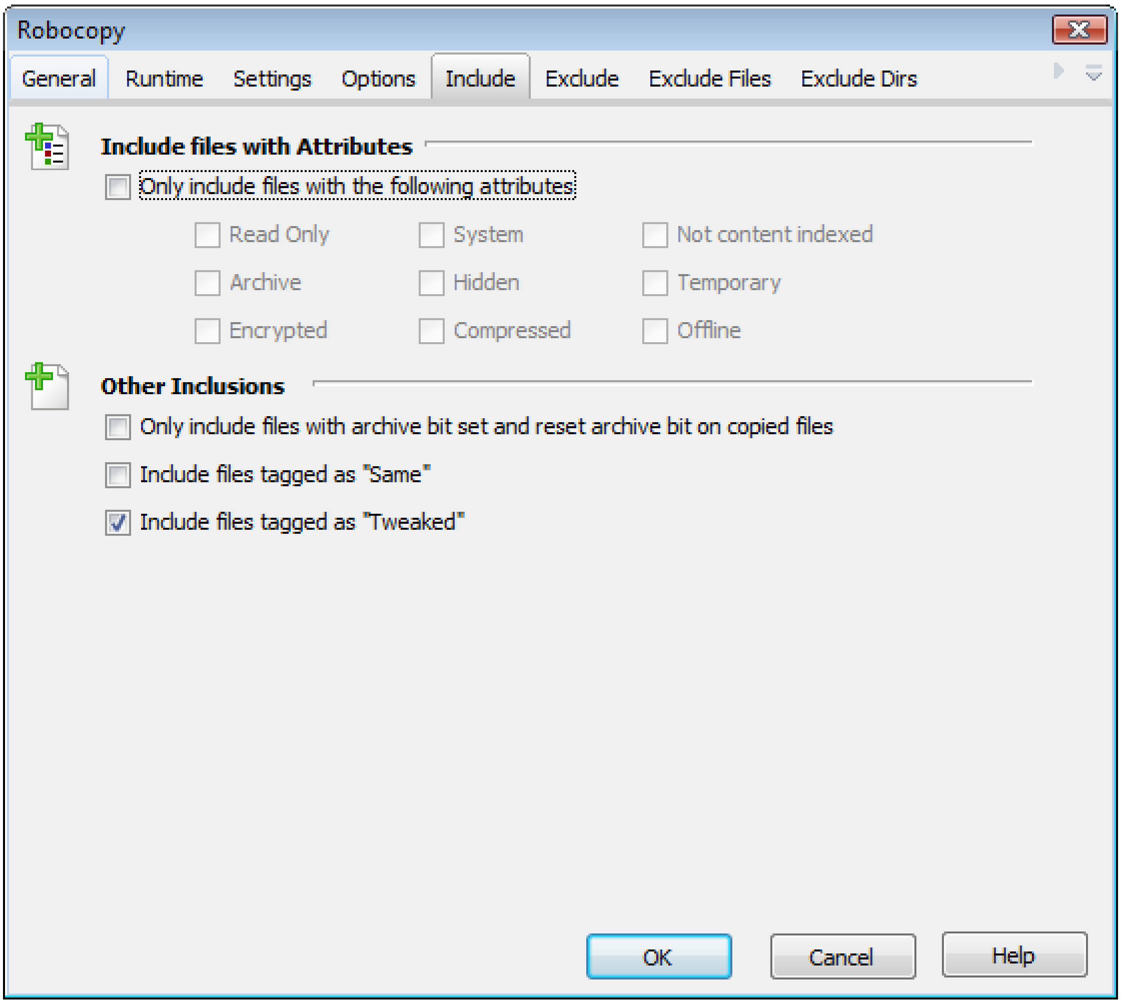
Wait Time (sec) - This allows you to specify the number of seconds between retries of failed copy attempts.

Extra Command Line Options

Extra Command Line Options - Specify any additional command line arguments to pass to Robocopy.

Include Page

The Include page allows you to specify the conditions which files must meet to be included in the Robocopy operation.



Include files with Attributes

□ Only include files with the following attributes

□ Read Only □ System □ Not content indexed

□ Archive □ Hidden □ Temporary

□ Encrypted □ Compressed □ Offline

Other Inclusions

□ Only include files with archive bit set and reset archive bit on copied files

□ Include files tagged as "Same"

☑ Include files tagged as "Tweaked"

Include files with Attributes

Only Include files with the following attributes - Selectively copy files based on the attributes of the source files.

Other Inclusions

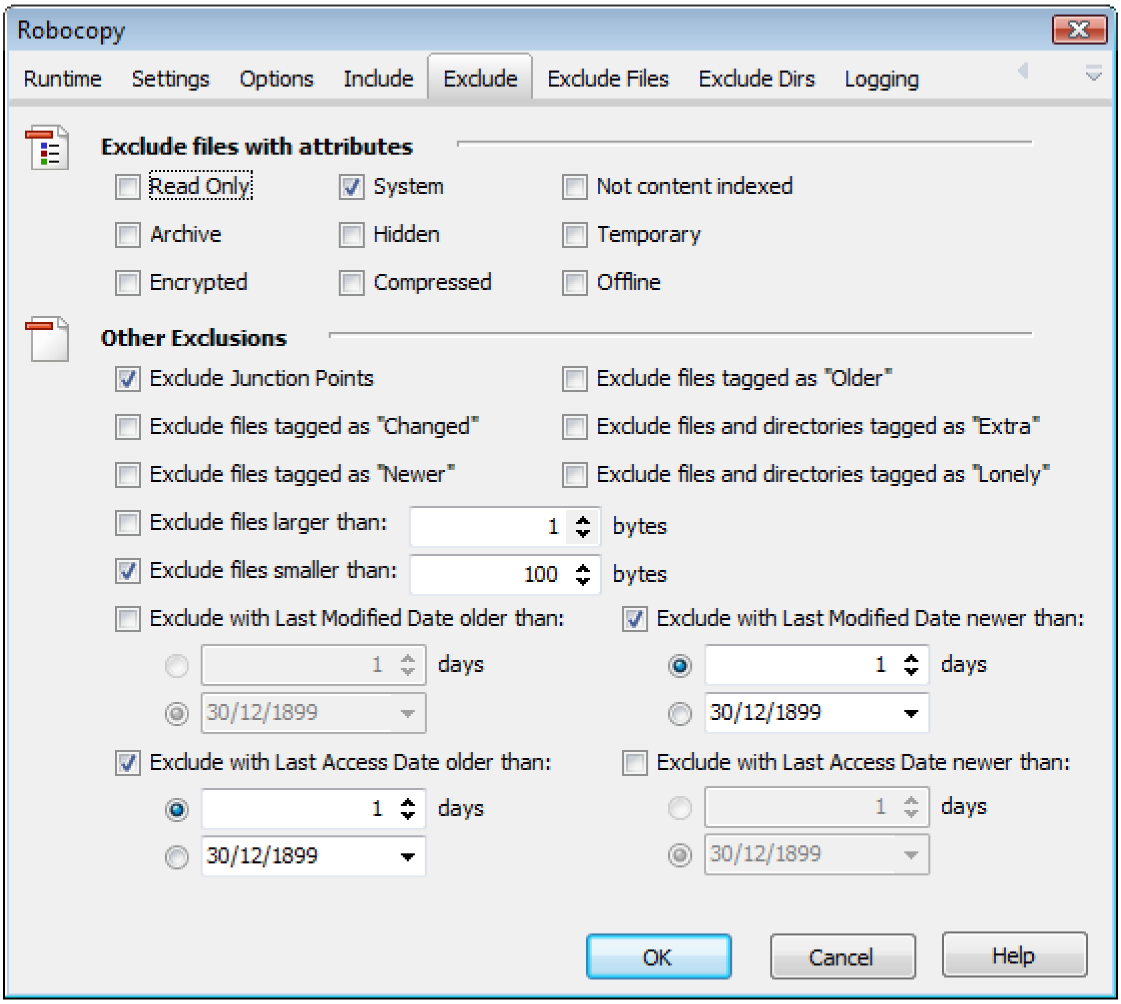
Only include files with archive bit set and reset archive bit on copied files - Only source files that have the archive bit set will be copied. Once the copy has completed the archive bit will be reset on these files.

Include files tagged as "Same" - Files are deemed to be same when the file exists in both the source and the destination and are identical in contents, size, time stamp and attributes.

Include files tagged as "Tweaked" - Files are deemed to be tweaked when the file exists in both the source and the destination, containing identical in contents, size, time stamp but differing attributes.

Exclude Page

The Exclude page allows you to specify conditions to exlcude files from the Robocopy operation.



Exclude files with attributes

□ Read Only ☑ System □ Not content indexed
□ Archive □ Hidden □ Temporary
□ Encrypted □ Compressed □ Offline

Other Exclusions

☑ Exclude Junction Points □ Exclude files tagged as "Older"
□ Exclude files tagged as "Changed" □ Exclude files and directories tagged as "Extra"
□ Exclude files tagged as "Newer" □ Exclude files and directories tagged as "Lonely"
□ Exclude files larger than: 1 bytes
☑ Exclude files smaller than: 100 bytes
□ Exclude with Last Modified Date older than: ○ 1 days ● 30/12/1899
☑ Exclude with Last Modified Date newer than: ● 1 days ○ 30/12/1899
☑ Exclude with Last Access Date older than: ● 1 days ○ 30/12/1899
□ Exclude with Last Access Date newer than: ○ 1 days ● 30/12/1899

Exclude files with attributes - Exclude files from the Robocopy operation which have the selected attributes.

Other exclusions

Exclude Junction Points - Self explanatory.

Exclude files tagged as "Older" - Exclude files that exist in both the source and destination where the source file is older than the destination file.

Exclude files tagged as "Changed" - Exclude files that exist in both the source and destination where the timestamps match but the file sizes are different.

Exclude files tagged as "Extra" - Exclude files that exist in the destination but do not exist in the source directory.

Exclude files tagged as "Newer" - Exclude files that exist in both the source and destination where the source file is newer than the destination file.

Exclude files tagged as "Lonely" - Exclude files that exist in the source but not in the destination.

Exclude files larger than *n* bytes - Self explanatory.

Exclude files smaller than *n* bytes - Self explanatory.

Exclude with Last Modified Date older than - Self explanatory.

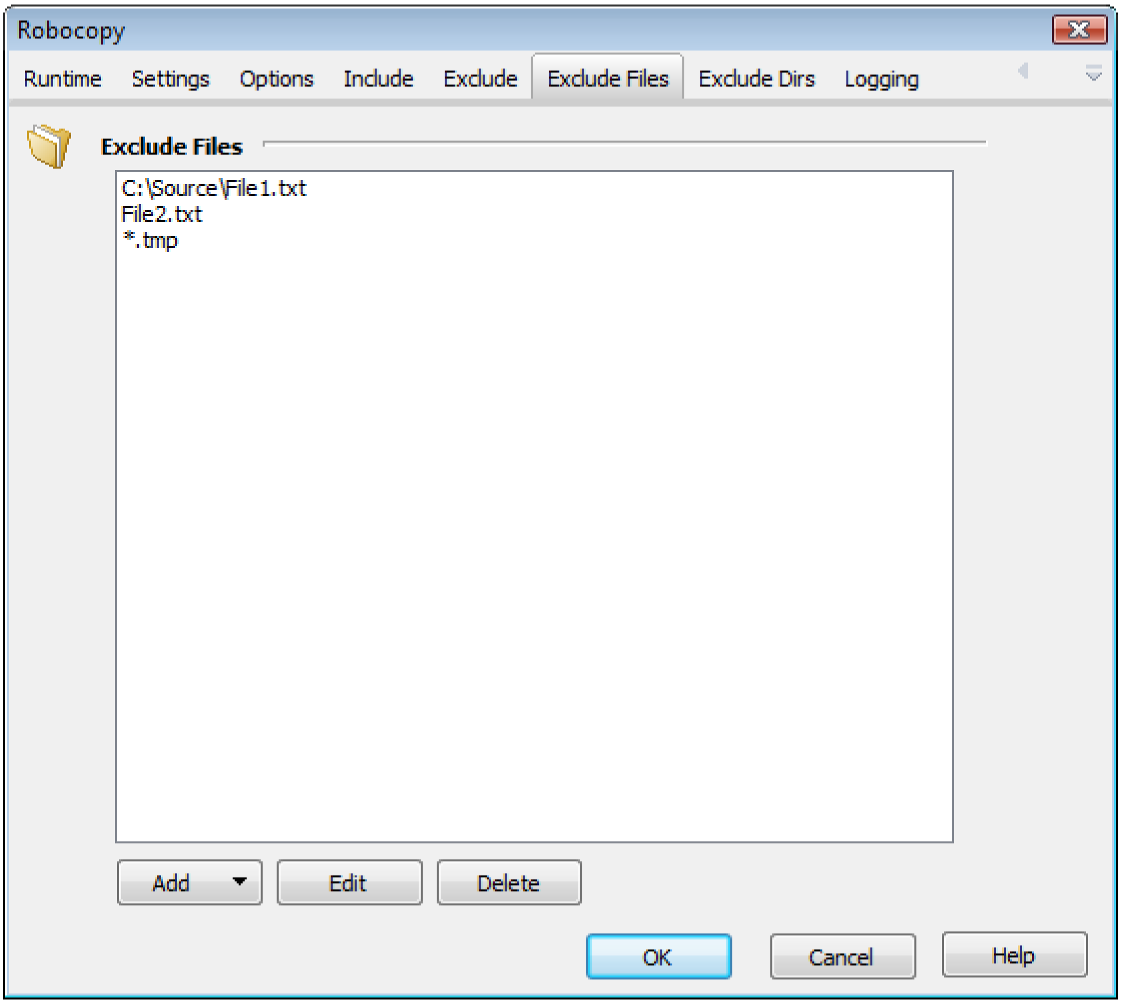
Exclude with Last Modified Date newer than - Self explanatory.

Exclude with Last Access Date older than - Self explanatory.

Exclude with Last Access Date newer than - Self explanatory.

Exclude Files Page

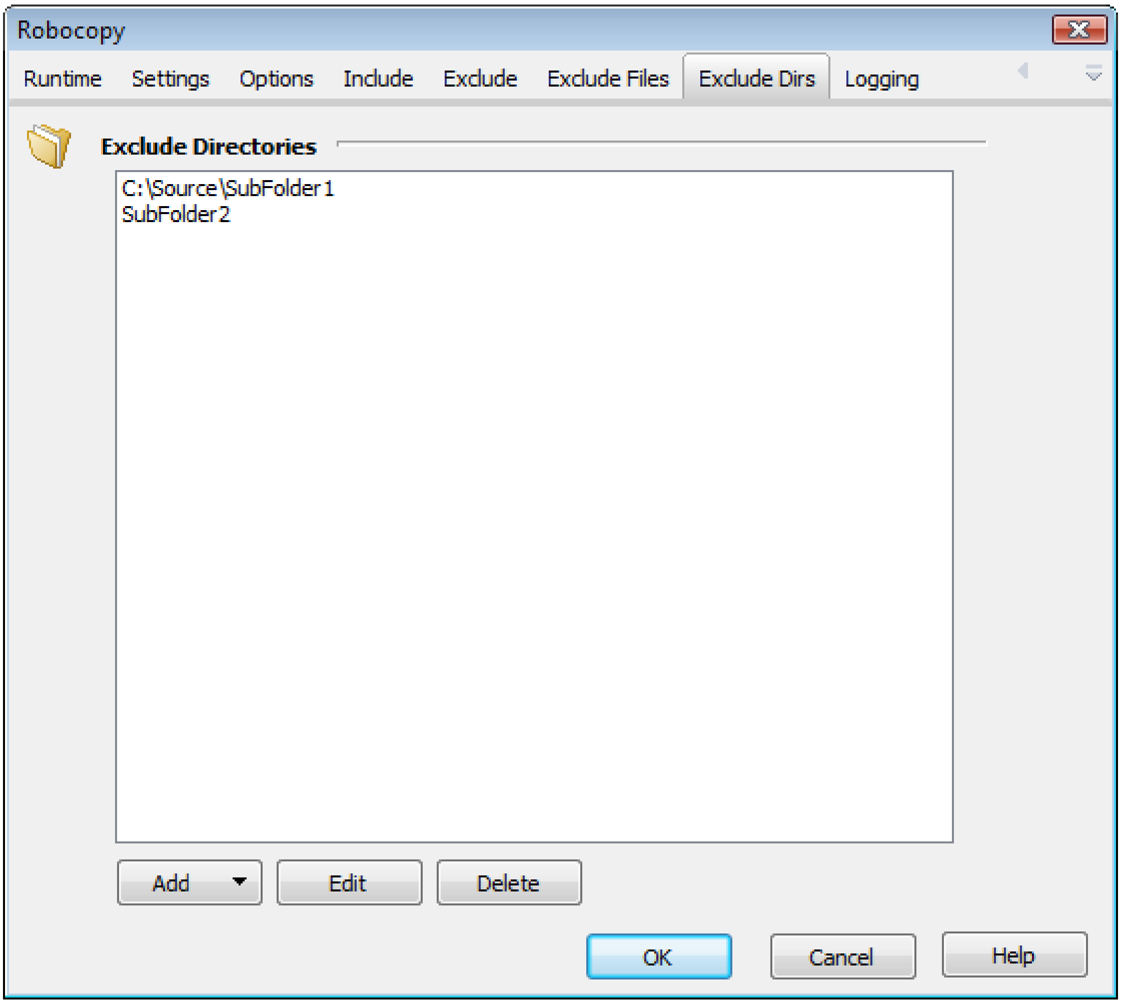
The Exclude Files page allows you to specify file names and file masks to exclude files from the Robocopy operation.



C:\Source\File1.txt
File2.txt
*.tmp

Exclude Dirs Page

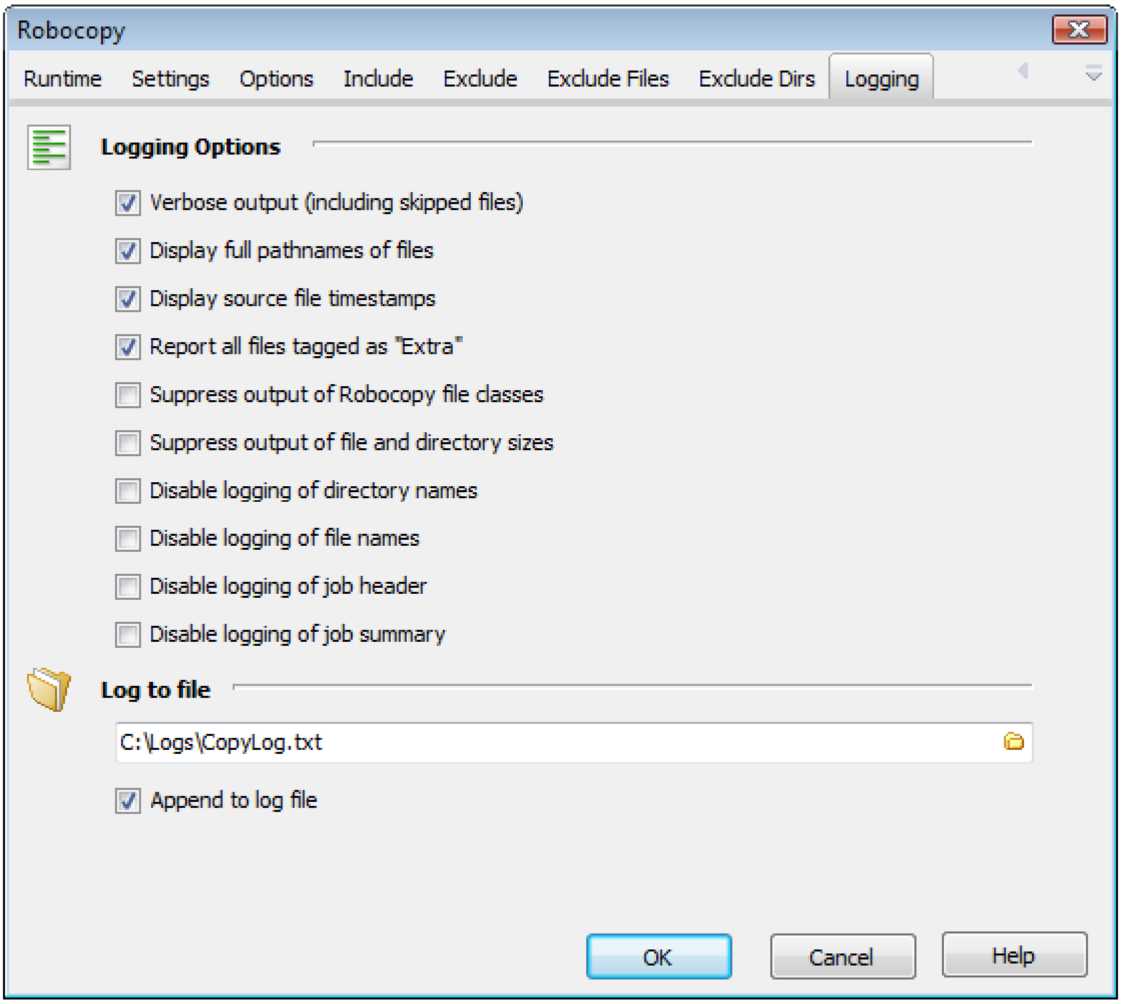
The Exclude Dirs page allows you to specify directories to be excluded from the Robocopy operation.



C:\Source\SubFolder1
SubFolder2

Logging Page

The logging page provides options to customise the output from Robocopy.



Logging Options

☑ Verbose output (including skipped files)

☑ Display full pathnames of files

☑ Display source file timestamps

☑ Report all files tagged as "Extra"

□ Suppress output of Robocopy file classes

□ Suppress output of file and directory sizes

□ Disable logging of directory names

□ Disable logging of file names

□ Disable logging of job header

□ Disable logging of job summary

Log to file

C:\Logs\CopyLog.txt

☑ Append to log file

Logging Options

Verbose Output - Increase the verbosity of the log output, this will include the logging of skipped files.

Display full pathnames of files - Self explanatory.

Display source file timestamps - Self explanatory.

Report all files tagged as "Extra" - Enable reporting of extra files. Extra files are files which exist the destination directory but do not exist in the source directory.

Suppress output of Robocopy file classes - Disable reporting of Robocopy file classes. Robocopy file classes are *Lonely, Tweaked, Same, Changed, Newer, Older, Extra* and *Mismatched*.

Suppress output of file and directory sizes - Self explanatory.

Disable logging of directory names - Self explanatory.

Disable logging of file names - Self explanatory.

Disable logging of job header - Self explanatory.

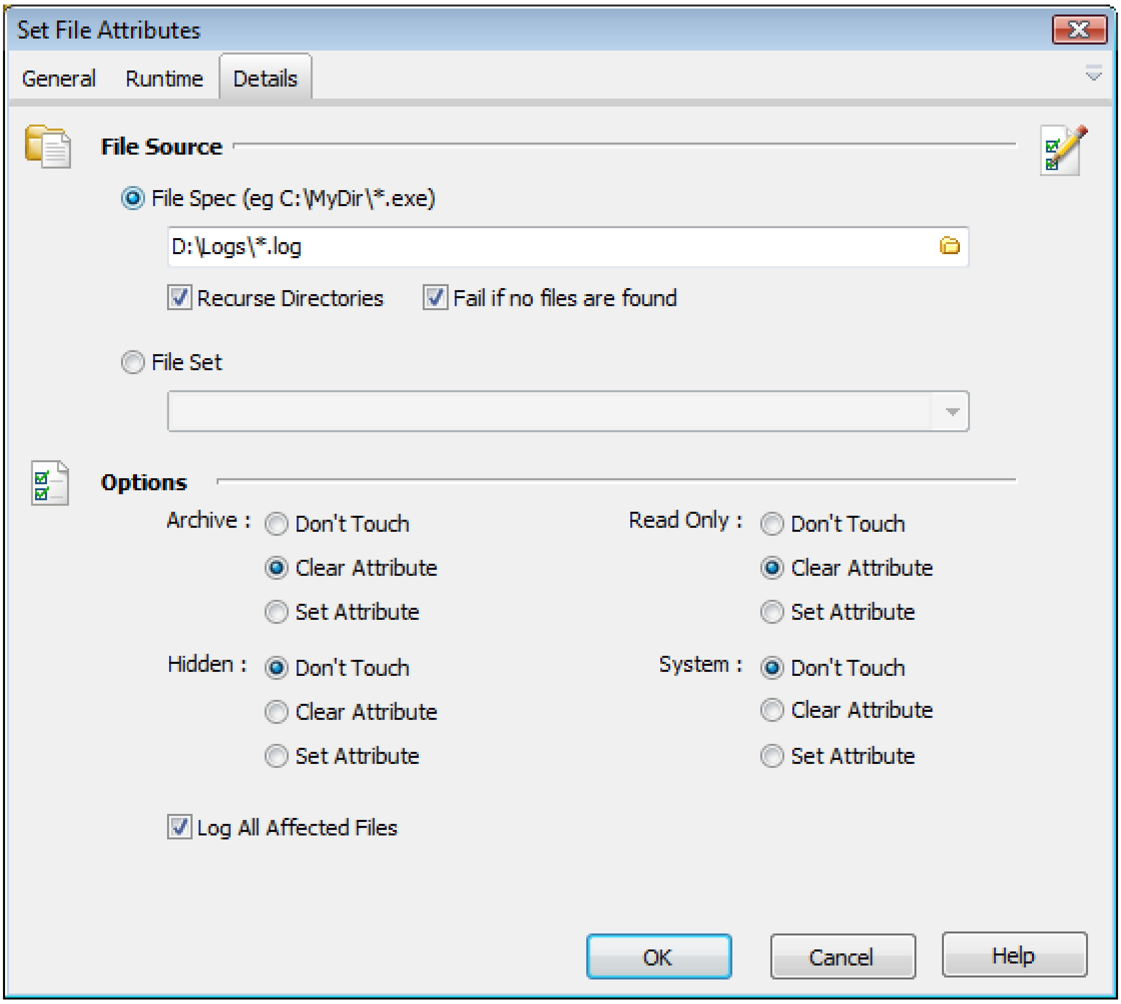
Disable logging of job summary - Self explanatory.

Log To File - Provide a file path to log Robocopy output to a file as well as a the Automise log.

Append to log file - Append to the file specified rather than overwriting it each time the action is run.

6.9.22 Set File Attributes Action

This action enables you to set a file or a set of file's attributes.



File Source

● File Spec (eg C:\MyDir\*.exe)

D:\Logs\*.log

☑ Recurse Directories ☑ Fail if no files are found

○ File Set

Options

Archive : ○ Don't Touch ● Clear Attribute ○ Set Attribute

Read Only : ○ Don't Touch ● Clear Attribute ○ Set Attribute

Hidden : ● Don't Touch ○ Clear Attribute ○ Set Attribute

System : ● Don't Touch ○ Clear Attribute ○ Set Attribute

☑ Log All Affected Files

Scripting Info

The Action properties available are :

property FileSpec : WideString ;// The file specification of the files to process. Wildcards (*, ?) may be used
property Recurse : WordBool;// Recurse Sub Directories

property LogAllFiles : WordBool;//Log All Files processed to the Automise Output Tree.

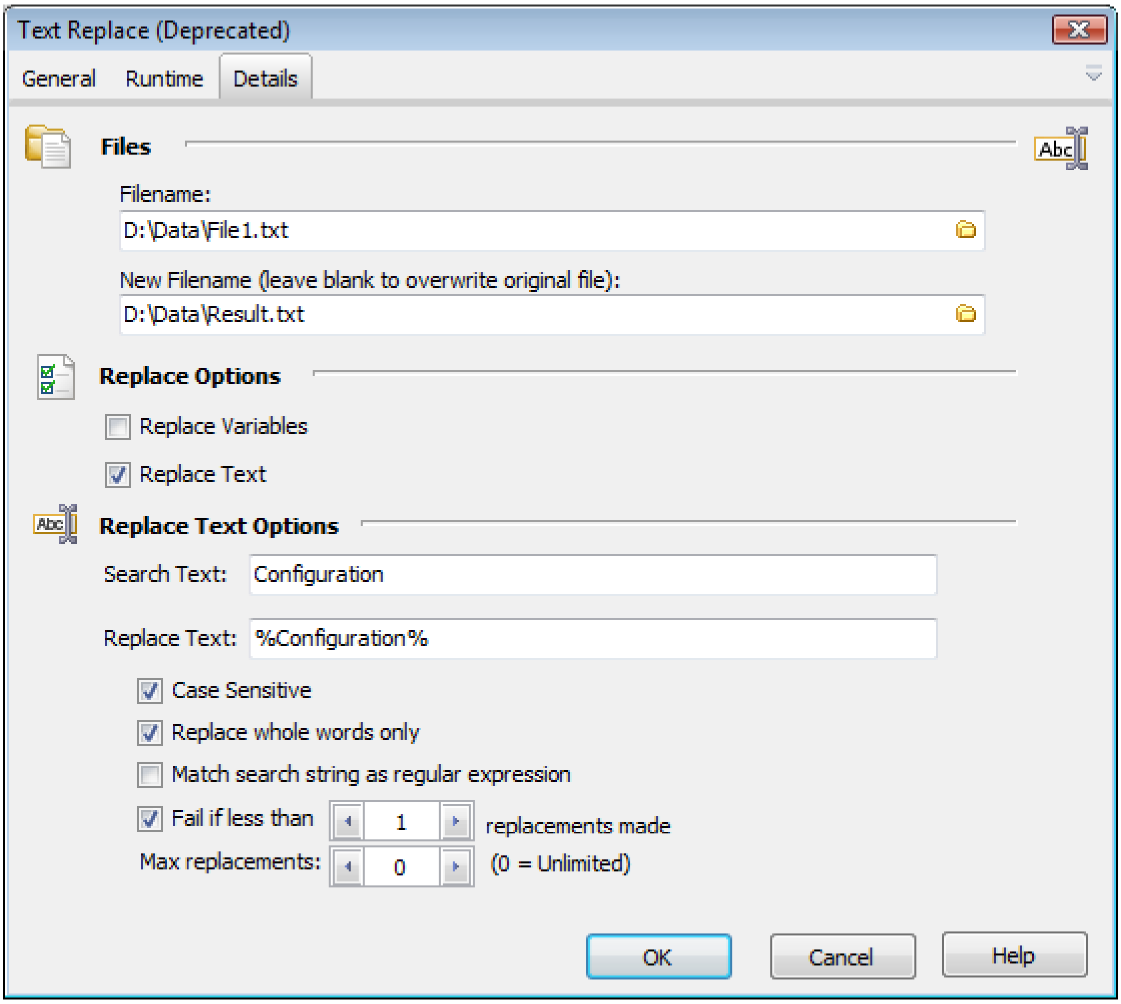
6.9.23 Text Replace Action

NOTE: This action has been deprecated and replaced with the Text Find / Replace Action and the Replace Variables action. Please use one of these actions where possible.

The text replace action allows you to replace strings and Automise variables in a file with specified values.

(Note that this will only work on Text files, do not attempt to use it with binary files.)

If you want to only search for content, use the Text Find Action.



Filename: D:\Data\File1.txt

New Filename (leave blank to overwrite original file): D:\Data\Result.txt

Replace Options

☐ Replace Variables

☑ Replace Text

Replace Text Options

Search Text: Configuration

Replace Text: %Configuration%

☑ Case Sensitive

☑ Replace whole words only

☐ Match search string as regular expression

☑ Fail if less than 1 replacements made

Max replacements: 0 (0 = Unlimited)

Files

Specify a source and a destination file. If you want the replacements to be written to the same file as the source, leave the 'New Filename' blank.

Replace Automise Variables

If this box is checked, Automise variables will be replaced with their values. Mark variables you want replaced with percent signs, ie %VarName%

Note that the Text Replacement action does not discriminate between FB Variables and regular percent signs, so either escape "real" percent signs as "%%" (these can even then be replaced with % using Replace Text), or use a different notation and replace specific strings instead.

Match Search String As Regular Expression

If this box is checked, the search string will be treated as a Regular Expression (see the Regular Expression reference.) Note that the Case Sensitive and Whole Words Only options do not apply if this option is selected (they will be greyed out.)

Fail if less than N replacements made

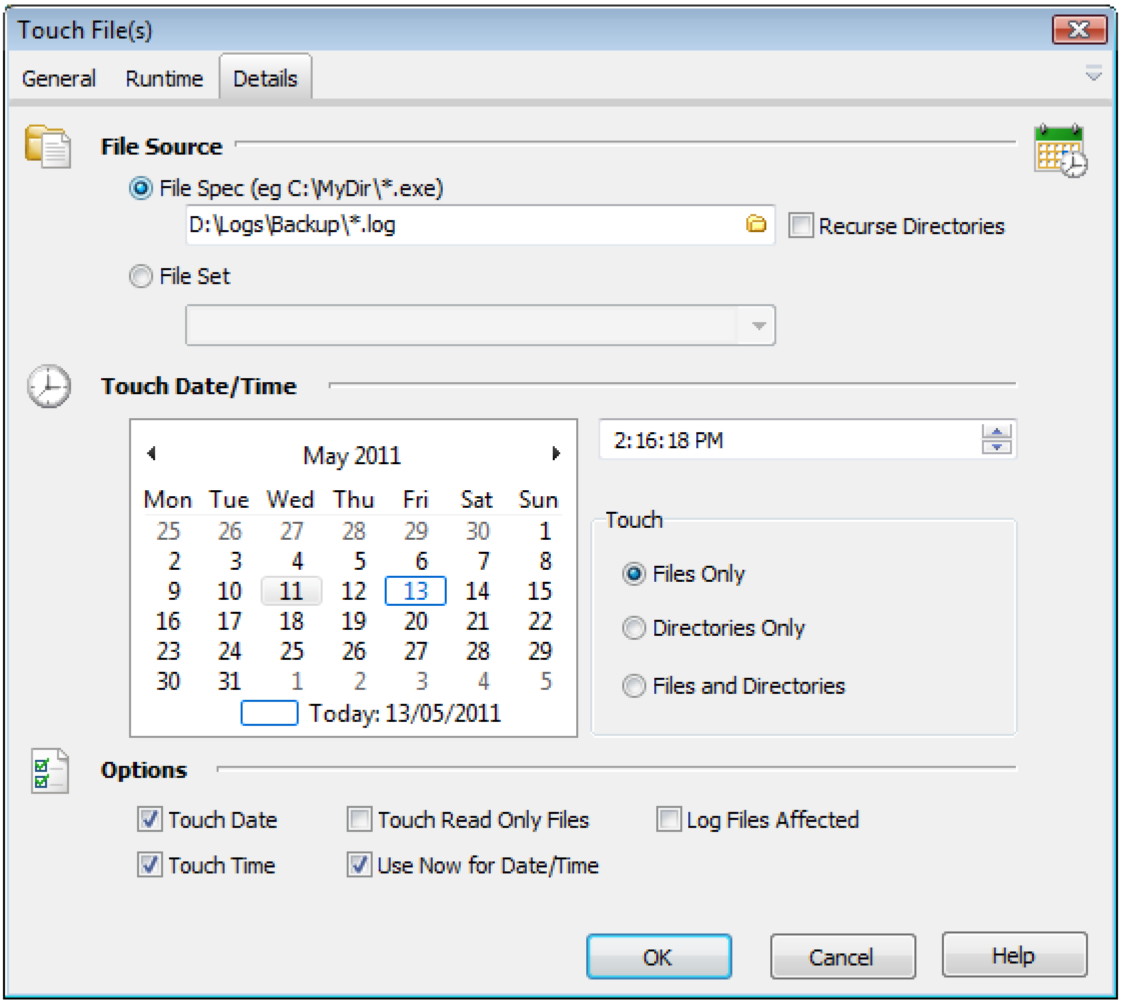
The action can be set to fail if an insufficient amount of replacements are made. This is useful to check that there isn't a syntax error or other mistake in the text file.

Max Replacements: N

The action will replace only the first N matches it finds. Set to zero to replace all matches.

6.9.24 Touch File(s)

This Action type provides the same functionality as the Touch command line utility provided with many C++ tools. It allows you to set the last modified File Date and/or Time for a file or a set of files matching a filespec or contained in a fileset.



Touch File(s)

General Runtime Details

File Source

● File Spec (eg C:\MyDir\*.exe)

D:\Logs\Backup\*.log

□ Recurse Directories

○ File Set

Touch Date/Time

May 2011

Mon	Tue	Wed	Thu	Fri	Sat	Sun
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Today: 13/05/2011

2:16:18 PM

Touch

● Files Only

○ Directories Only

○ Files and Directories

Options

☑ Touch Date

□ Touch Read Only Files

□ Log Files Affected

☑ Touch Time

☑ Use Now for Date/Time

File Spec - the name of a file or a filespec (that contains wildcards) that the touch command will affect

Recurse Directories - touch files in subdirectories that match the file spec

File Set - the name of a FileSet (created with the File Set Define action) that the touch command will affect

Touch Date/Time - specify the date and/or time on the calendar and timepicker that will be applied to the files

Touch Date - if selected, then the date of the file will be touched

Touch Time - if selected, then the time of the file will be touched (at least one of Date or Time must be selected)

Touch Read Only Files - if not checked, then read only files will be ignored

Use Now for Date/Time - the date/time will be the current date time when the action executes. If you want to specify a date/time based on a variable, then you can use the DateTime property of the action in script (see below)

Log Files Affected - log all files that were touched

Scripting Info

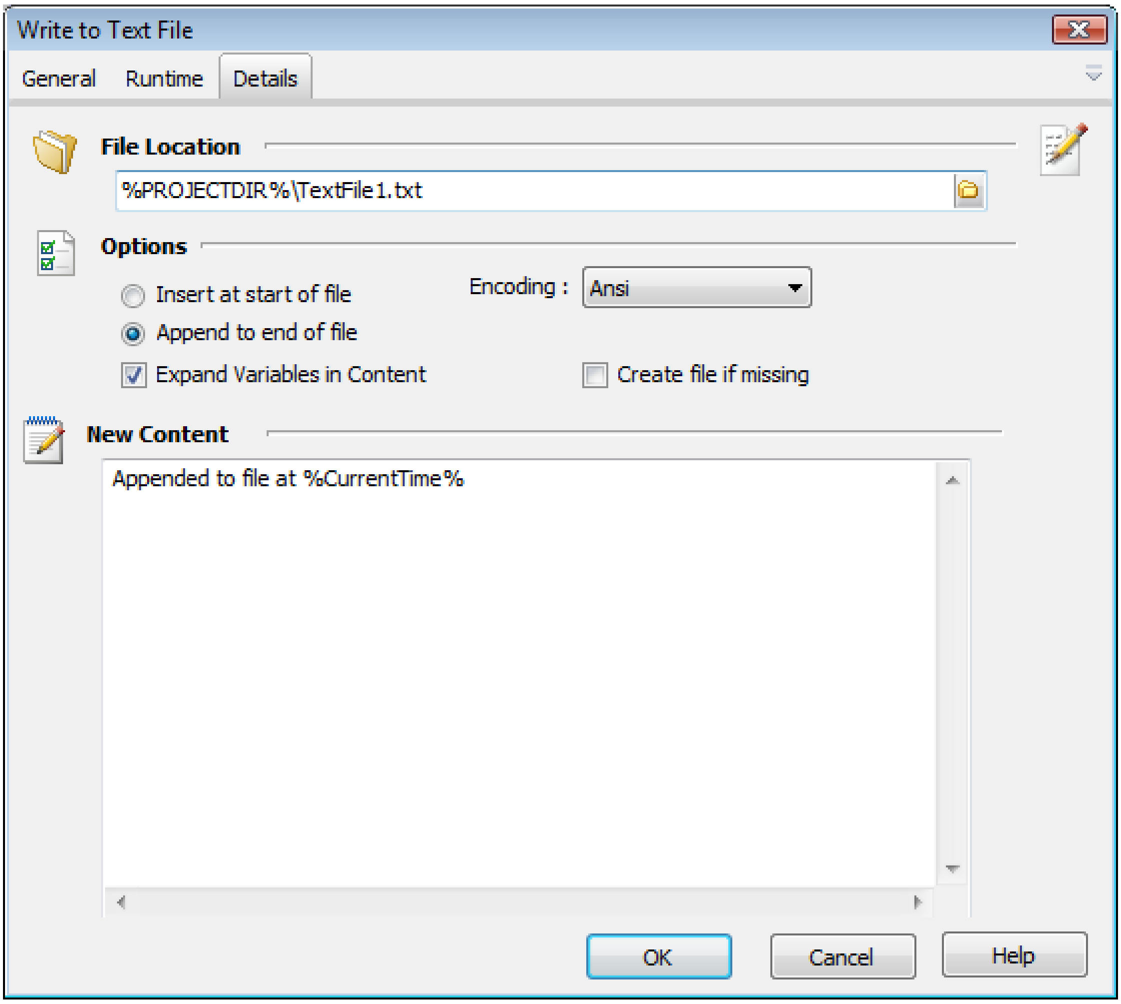
The Action properties available are :

property DateTime: TDateTime; // Delphi DateTime
property FileSpec: WideString; // files to touch, can include wildcards
property TouchDate: WordBool; // set the date
property TouchTime: WordBool;// set the time
property TouchReadOnly: WordBool;// by default, read only files will not be touched.
property Recurse: WordBool; // recurse directories

6.9.25 Write Text File

The Write to Text File action enables you to automate inserting and appending text to an existing file.

You can use Automise variables in the File Contents field to customise the content. You can either append to the end of the file, or insert text at the start of the file.



File Location: %PROJECTDIR%\TextFile1.txt

○ Insert at start of file
● Append to end of file
☑ Expand Variables in Content

Encoding : Ansi

□ Create file if missing

New Content: Appended to file at %CurrentTime%

File Location - specify the filename to write to (the example shows a variable which will contain a filename during execution - provided by a file iterator action)

Insert location - Specify to append to end of file, or insert at beginning

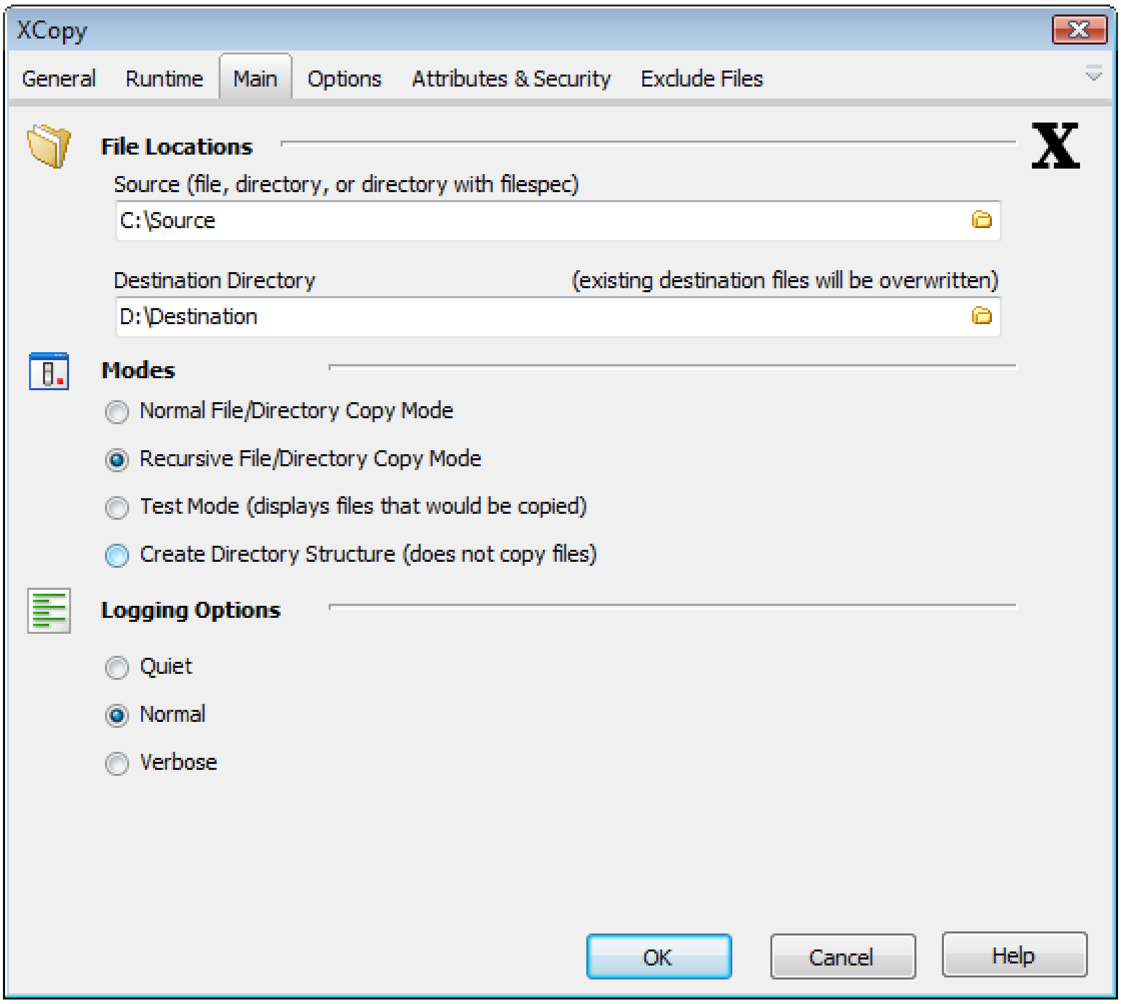
Create file if missing - the action will normally fail if the file doesn't already exist.

New Content - specify the content to write to the text file

6.9.26 XCopy

The XCopy action enables you to automate file and directory copying using XCopy. XCopy is a powerful utility included in Windows which is faster and more capable than standard file and copy tools.

Main Page



Source (file, directory, or directory with filespec): C:\Source

Destination Directory (existing destination files will be overwritten): D:\Destination

Modes
- ○ Normal File/Directory Copy Mode
- ● Recursive File/Directory Copy Mode
- ○ Test Mode (displays files that would be copied)
- ○ Create Directory Structure (does not copy files)

Logging Options
- ○ Quiet
- ● Normal
- ○ Verbose

File Locations

Source - Specify the folder or file or file mask that you want to copy.

Destination Directory - Specify the destination directory for the copy operation.

Modes

Normal File/Directory Copy Mode - Only copy the files in the directory specified, do not include sub directories.

Recursive File/Directory Copy Mode - Copy files in directory and sub-directories. Empty directories will also be copied.

Test Mode - Log the results of the operation without actually copying any files. Allows you to preview the result without actually copying any files.

Create Directory Structure - Only copy the directory structure, do not copy the files. Empty directories will also be copied.

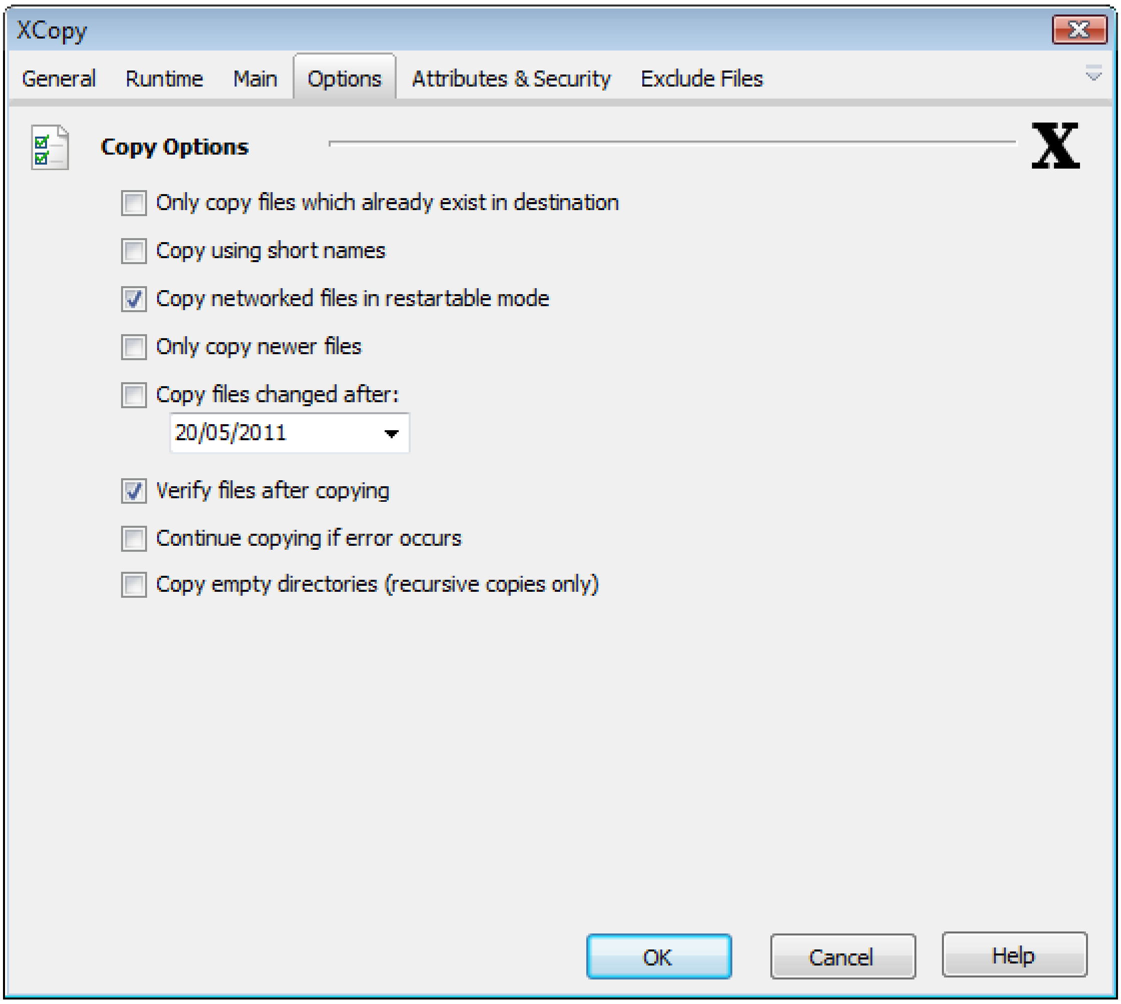
Logging Options

Quiet - Suppress all the output messages from XCopy.

Normal -Display the normal output from XCopy.

Verbose -Display source and destination paths for each item in the operation.

Options Page



☐ Only copy files which already exist in destination
☐ Copy using short names
☑ Copy networked files in restartable mode
☐ Only copy newer files
☐ Copy files changed after:
20/05/2011
☑ Verify files after copying
☐ Continue copying if error occurs
☐ Copy empty directories (recursive copies only)

Copy Options

Only copy files which already exist in destination - Files that exist in the source directory and do not exist in the destination directory will not be copied.

Copy using short names - Enable when the destination requires that files are named in short format (8.3 characters).

Copy networked files in restartable mode - Files are copied in restartable mode so that the copy operation can be resumed from point of failure if the network connection fails.

Copy only newer files - Enable this option to only copy files that exist in both the source and the destination where the source file is newer than the destination file.

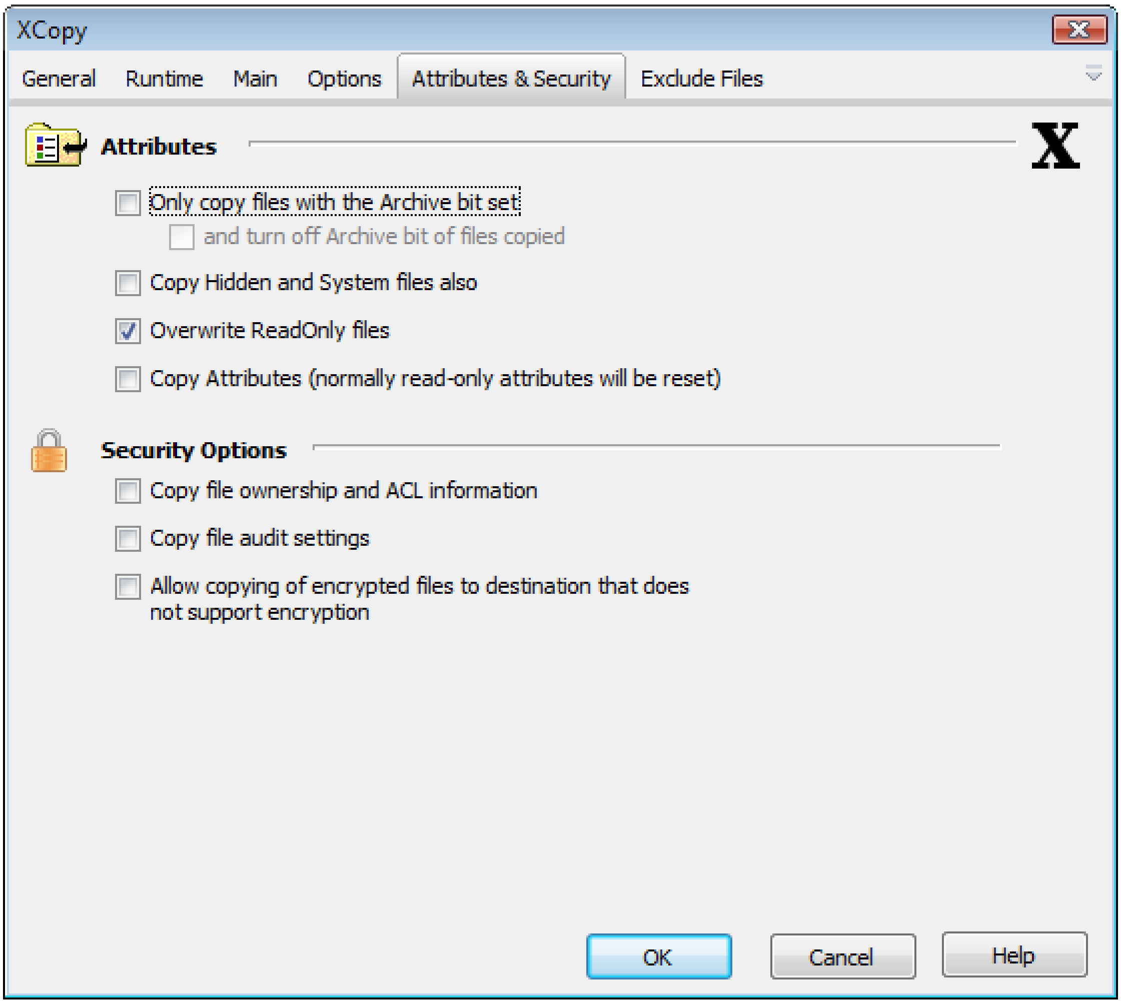
Copy files that are changed after - Enable this option to only copy files that exist in both the source and the destination where the source files have changed after the date specified.

Verify files after copying - After copying each file verify that the destination file is identical to the source file.

Continue copying if error occurs - If a particular file fails to copy, ignore the error and continue with the next file.

Copy empty directories (recursive copies only) - When performing a recursive copy, enable this option to include empty directories.

Attributes & Security Page



□ Only copy files with the Archive bit set

□ and turn off Archive bit of files copied

□ Copy Hidden and System files also

☑ Overwrite ReadOnly files

□ Copy Attributes (normally read-only attributes will be reset)

□ Copy file ownership and ACL information

□ Copy file audit settings

□ Allow copying of encrypted files to destination that does not support encryption

Attributes

Only copy files with the Archive bit set - Files that do not have the archive bit set are not copied.

Turn off Archive bit of files copied - Once files are copied, the archive bit is turned off.

Copy Hidden and System files also - System and Hidden files are included in the copy operation.

Overwrite Read Only files - Allow existing read-only files to be overwritten.

Copy Attributes (normally read-only attributes will be reset) - Retains the file attributes from the source file. By default XCopy will remove the read-only attribute from any files copied.

Security Options

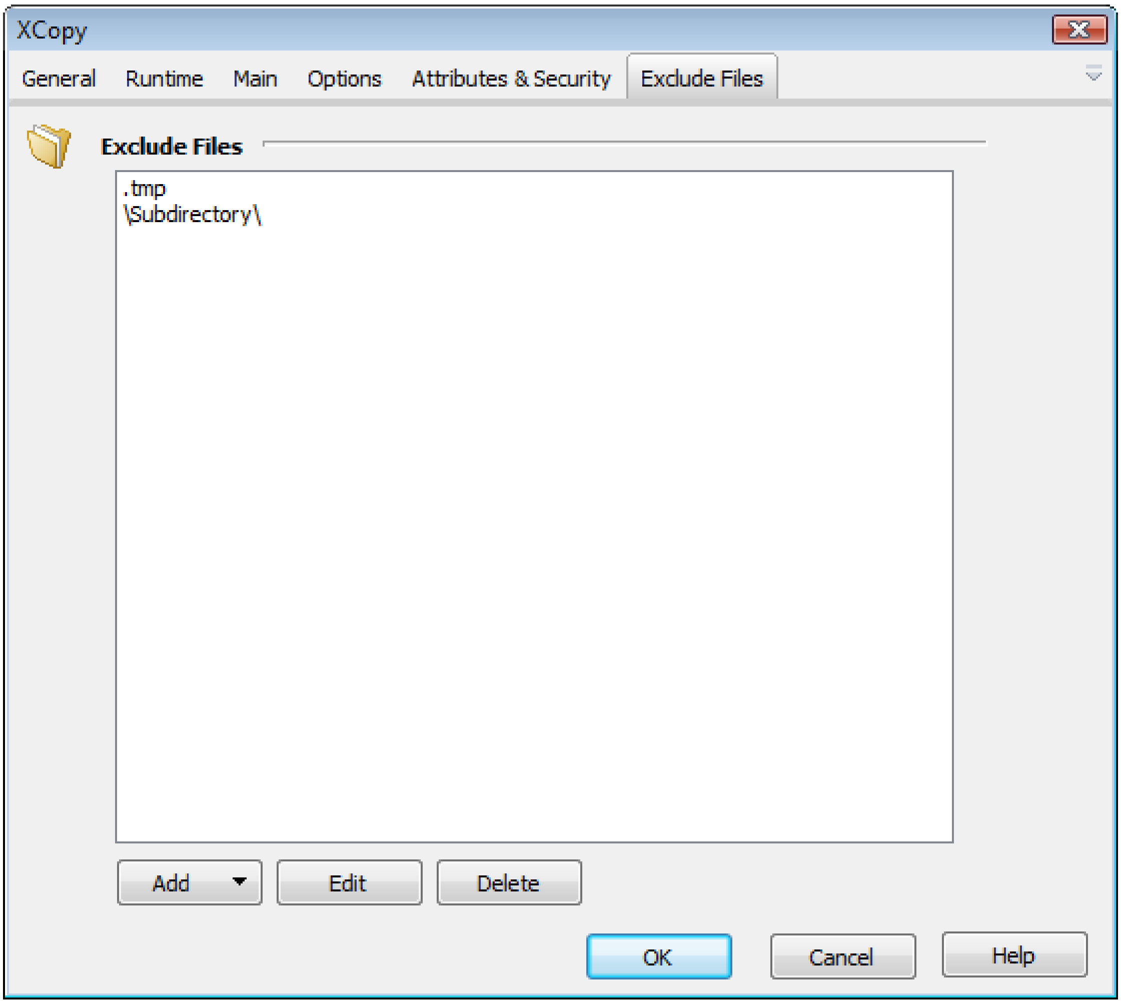
Copy file ownership and ACL information - Retains the file ownership and access control list information in the copy.

Copy file audit settings - Copies the file audit settings as well as the system access control list information.

Allow copying of encrypted files to destination that does not support encryption - Decrypt files if destination does not support encryption.

Exclude Files Page

Use the Exclude Files page to specify files that should not be included in the copy operation. Note that XCopy does not support the use of wildcard characters in the exclusion list, it only allows substring matches (i.e. *.txt* to match a any text files).



.tmp
\Subdirectory\

For more information on XCopy see: http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/xcopy.mspx

6.10 Flow Control

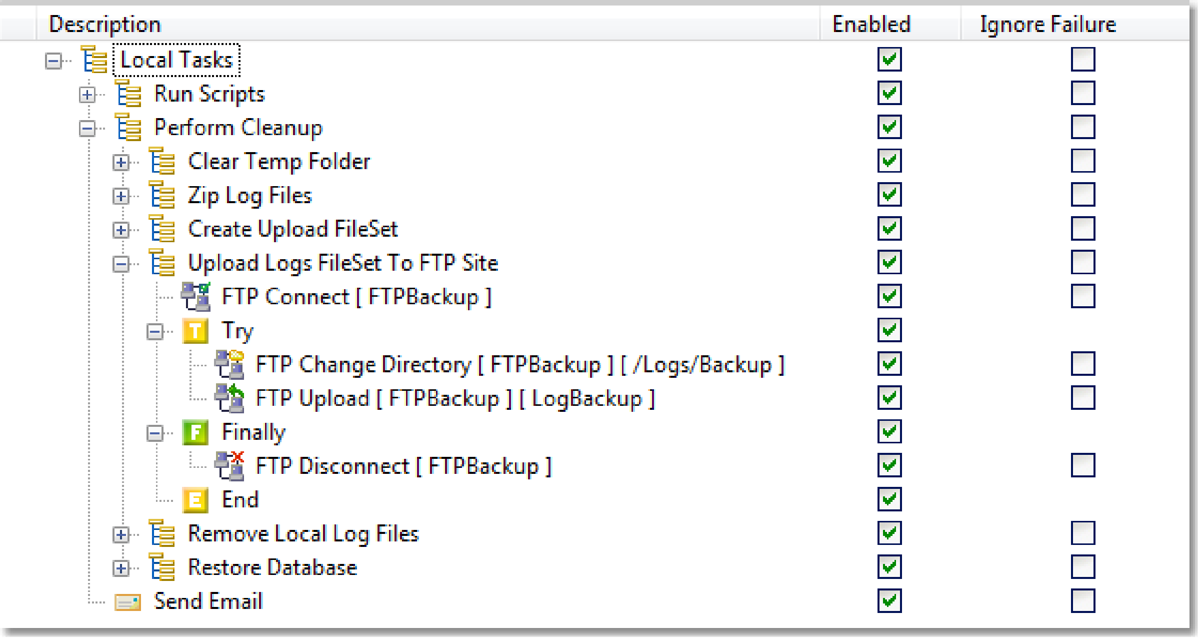
Automise includes many actions which let you control which actions are executed in your project. The Flow Control category in the Actions Pane includes the following flow control actions:

- Action Group - a "do nothing" action to help you structure your project with parent/child relationships between actions
- ASync Action Group - Runs all child actions simultaneously
- While Loop - Runs it's child actions until the Condition property returns False
- Include Project - Run a separate Automise project
- Run Action List - Run another Action List (just like calling a sub-procedure)
- Stop Run - Aborts the running of the current project
- Try/Catch/Finally/End - Catch and deal with errors. See Error Handling
- If .. Then - Add conditionals to your project
- If Prev Action Failed - Run a special set of actions if the previous action failed
- Switch/Case - Run certain actions based on a condition
- Delay - Add a delay to your project
- For Loop - Run a set of child actions for a specified number of times
- "Wait for" actions - suspends the project run until an event happens

Also see the Interactive category for prompting the user for particular values, as well as the Variables section.

6.10.1 Action Group

The Action Group action type has no specific functionality, but is extremely useful for structuring your project.



Description	Enabled	Ignore Failure
Local Tasks	☑	☐
Run Scripts	☑	☐
Perform Cleanup	☑	☐
Clear Temp Folder	☑	☐
Zip Log Files	☑	☐
Create Upload FileSet	☑	☐
Upload Logs FileSet To FTP Site	☑	☐
FTP Connect [FTPBackup]	☑	☐
Try	☑	
FTP Change Directory [FTPBackup] [/Logs/Backup]	☑	☐
FTP Upload [FTPBackup] [LogBackup]	☑	☐
Finally	☑	
FTP Disconnect [FTPBackup]	☑	☐
End	☑	
Remove Local Log Files	☑	☐
Restore Database	☑	☐
Send Email	☑	☐

There are a couple of ways to add child actions to an Action Group (or any other action for that matter):

1. Add an action from Action Types by using drag and drop - drop the action on top the action you want to be the parent

2. Select an action and indent it under the parent action by using either Ctrl-Right Arrow, or by using the menu (Actions | Indent) or toolbar buttons

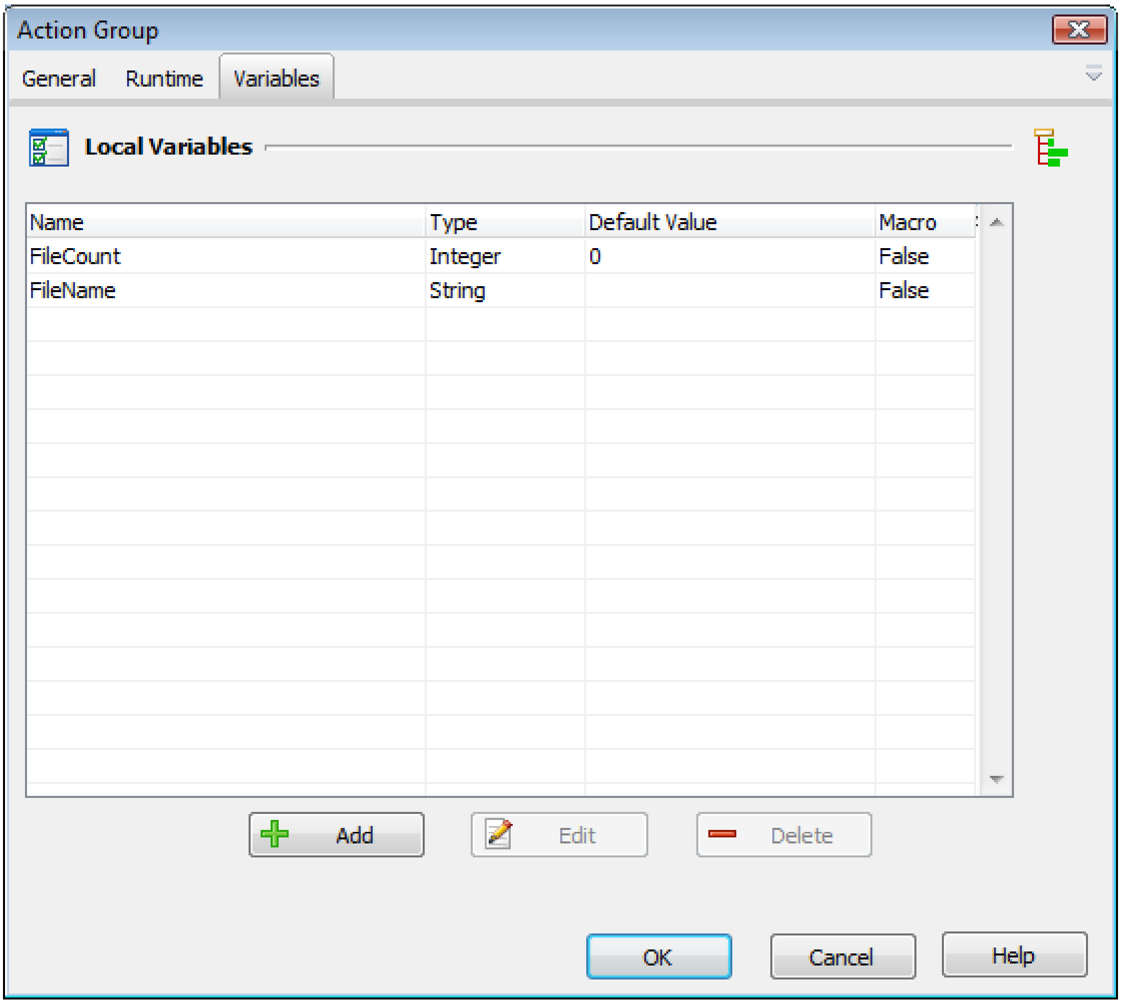
Notes:
The Action Group's BeforeAction, AfterAction, and OnStatusMessage script events execute the same as for any other action type.

There is also an Async Action Group, which runs its child actions in parallel.

Any action can be used as a parent action for other actions - you don't have to use Action Group

Local Variables

Action Groups allow you to declare Local variables are only available to child actions of the groups. Since action groups can be nested, you can also override variables in child groups, and actions in child groups can also reference actions in the parent groups. Local variables are also very useful when running actions in parallel.



Name	Type	Default Value	Macro
FileCount	Integer	0	False
FileName	String		False

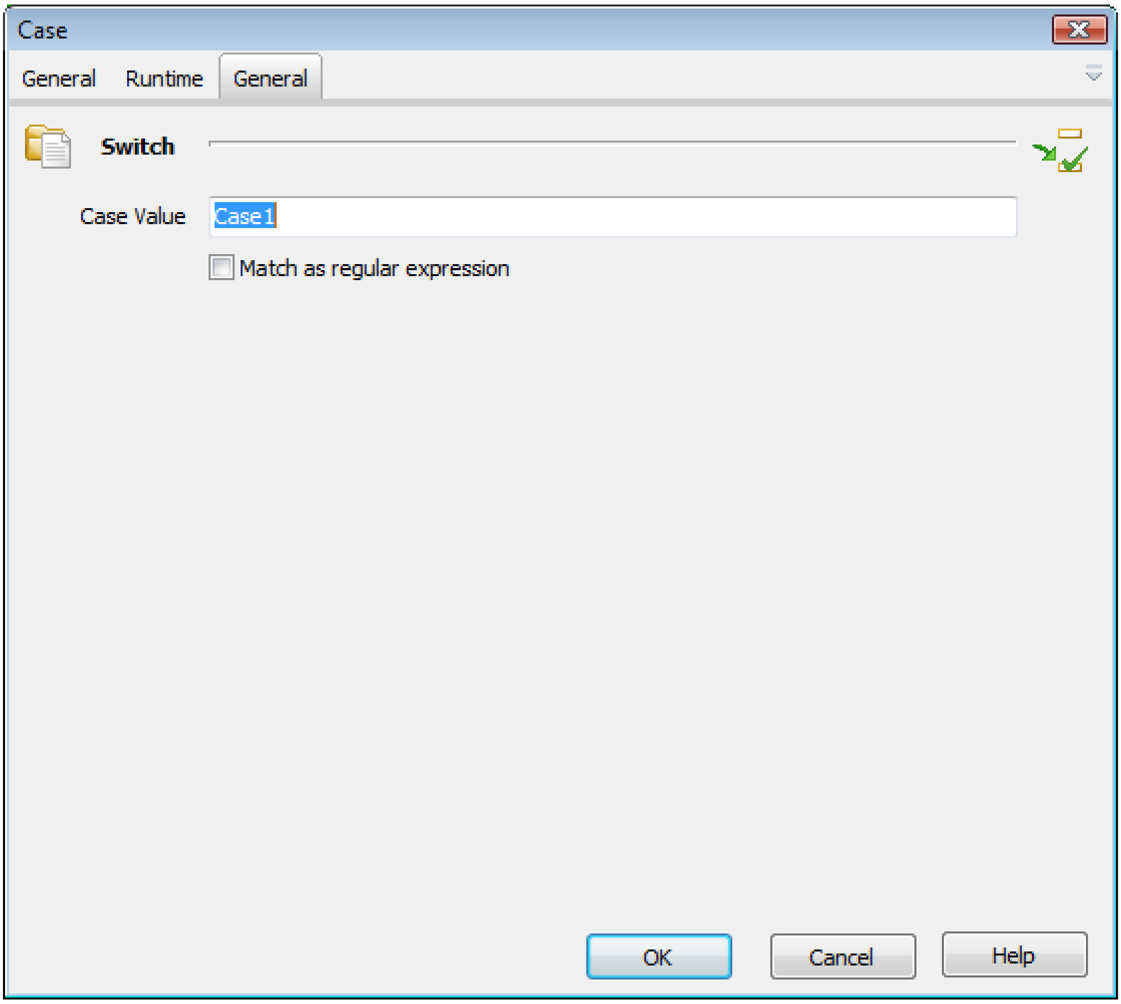
See also

Action Lists | Async Action Group | Include Automise Project Action

6.10.2 Case Action

The Case Action is used as the child of a Switch action to choose a particular case value.

See the Switch Action for details and an example.



Case Value: Case1

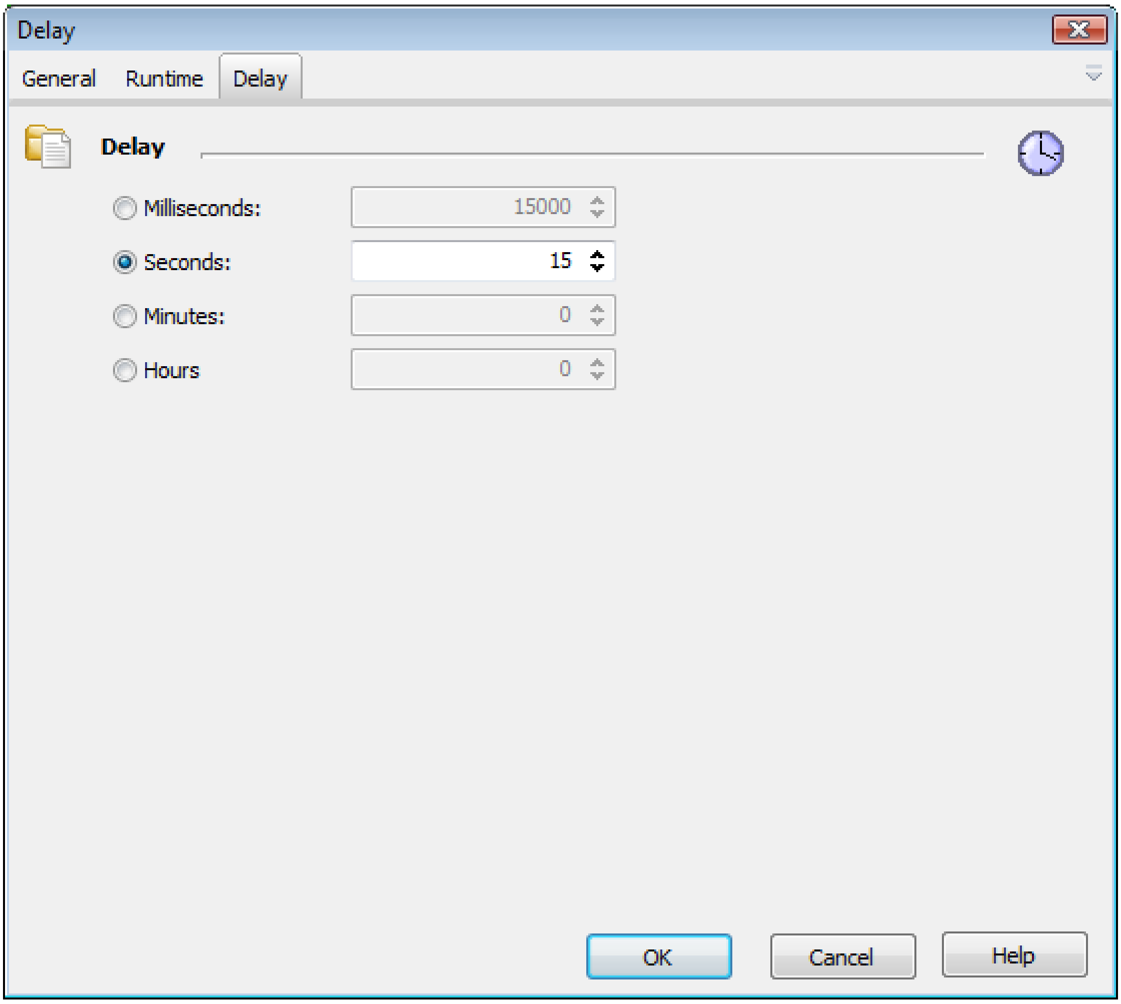
☐ Match as regular expression

See Also

Switch Action

6.10.3 Delay Action

This Action will pause for the time specified in the Delay property (in milliseconds).



○ Milliseconds: 15000

● Seconds: 15

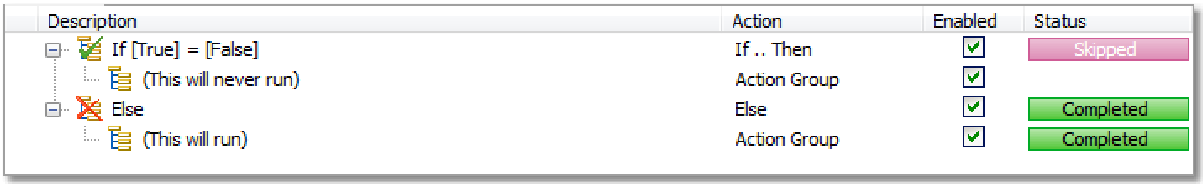
○ Minutes: 0

○ Hours 0

6.10.4 Else Action

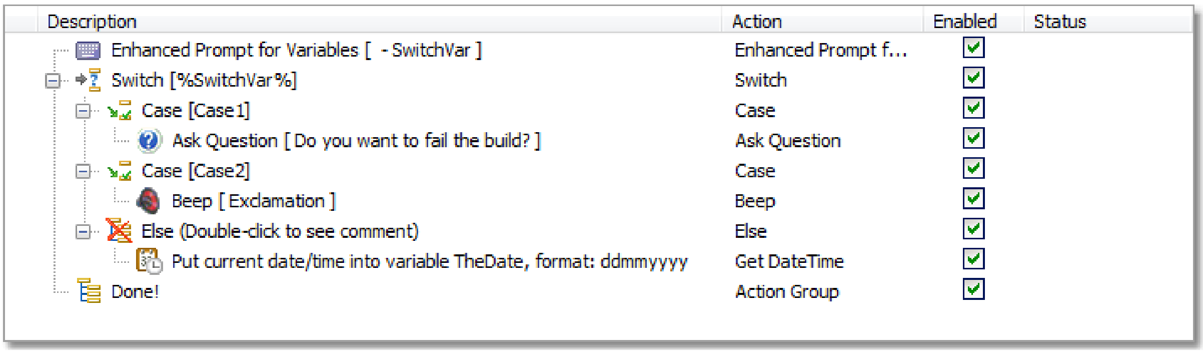
The else action can be used with the "If..Then" action, the "If Prev Action Failed" action, the Switch/Case Action, or the If COM Class Registered action:

For if actions:



Description	Action	Enabled	Status
If [True] = [False]	If .. Then	☑	Skipped
(This will never run)	Action Group	☑	
Else	Else	☑	Completed
(This will run)	Action Group	☑	Completed

For switch actions:



Description	Action	Enabled	Status
Enhanced Prompt for Variables [- SwitchVar]	Enhanced Prompt f...	☑	
Switch [%SwitchVar%]	Switch	☑	
Case [Case1]	Case	☑	
Ask Question [Do you want to fail the build?]	Ask Question	☑	
Case [Case2]	Case	☑	
Beep [Exclamation]	Beep	☑	
Else (Double-click to see comment)	Else	☑	
Put current date/time into variable TheDate, format: ddmmyyyy	Get DateTime	☑	
Done!	Action Group	☑	

When used with the Switch Action the Else action should be the last child action of the Switch Action (an error will be reported if not).

See Also

If .. Then Action | If Prev Action Failed Action | Switch Action | Case Action

6.10.5 Exit Action List Action

The "Exit Action List" action causes the current action list to terminate and control flow returns to the calling action list. The action will not cause an error condition.

Note that all Finally sections will run first before the action list is exited.

See Also

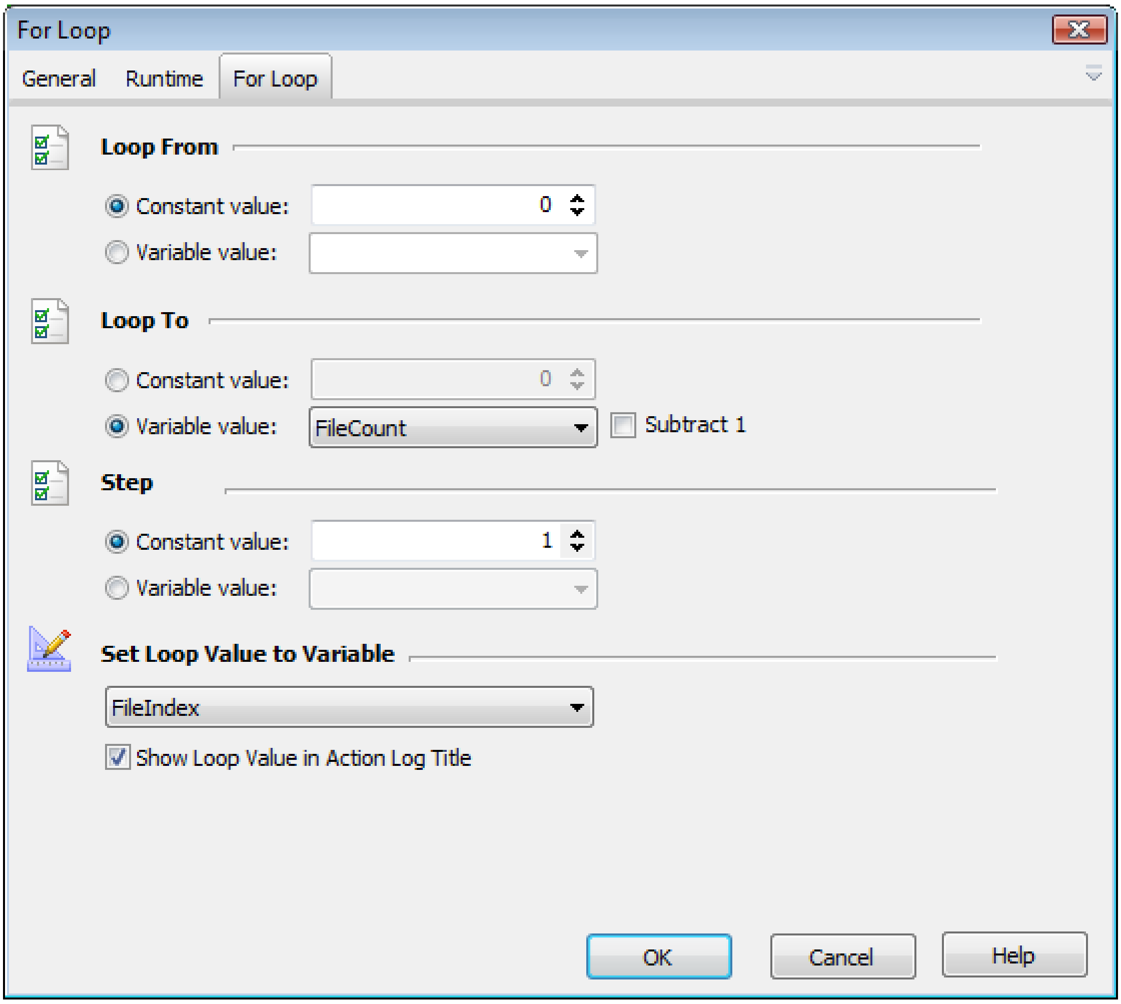
Stop Run Action

6.10.6 For Loop Action

The for loop action allows you to iterate a variable across a range of values. The For Loop action is actually an iterator, so all information from the general section on iterators applies.

The following screenshot is the equivalent of

For AAA = 0 To Project_Count Step 1 (BASIC)
or
for(AAA = 0; AAA <= Project_Count; AAA++) (C/Java)



Loop From

● Constant value: 0

○ Variable value:

Loop To

○ Constant value: 0

● Variable value: FileCount

□ Subtract 1

Step

● Constant value: 1

○ Variable value:

Set Loop Value to Variable

FileIndex

☑ Show Loop Value in Action Log Title

Loop From

The initial value for the loop value. The value can be either a constant or a Automise variable.

Loop To

The final ('to') value for the loop value. The loop will repeat until the loop value is greater than the 'to' value (for Step greater than zero) or less than the 'to' value (for Step less than zero.)

The 'to' value can be either a constant or a Automise variable. If using a variable, then checking the Subtract 1 checkbox will loop to 1 less than the variable value.

Loop Step

The loop value will be incremented by this amount after each iteration. The value is incremented *before* it is evaluated against the 'to' value.

The Step value can be positive or negative (or zero for an infinite loop), and can be either a constant or a Automise variable.

Set Loop Value to Variable

Specify a Automise variable to use for the loop value. If no variable is specified, the loop value will be stored internally.

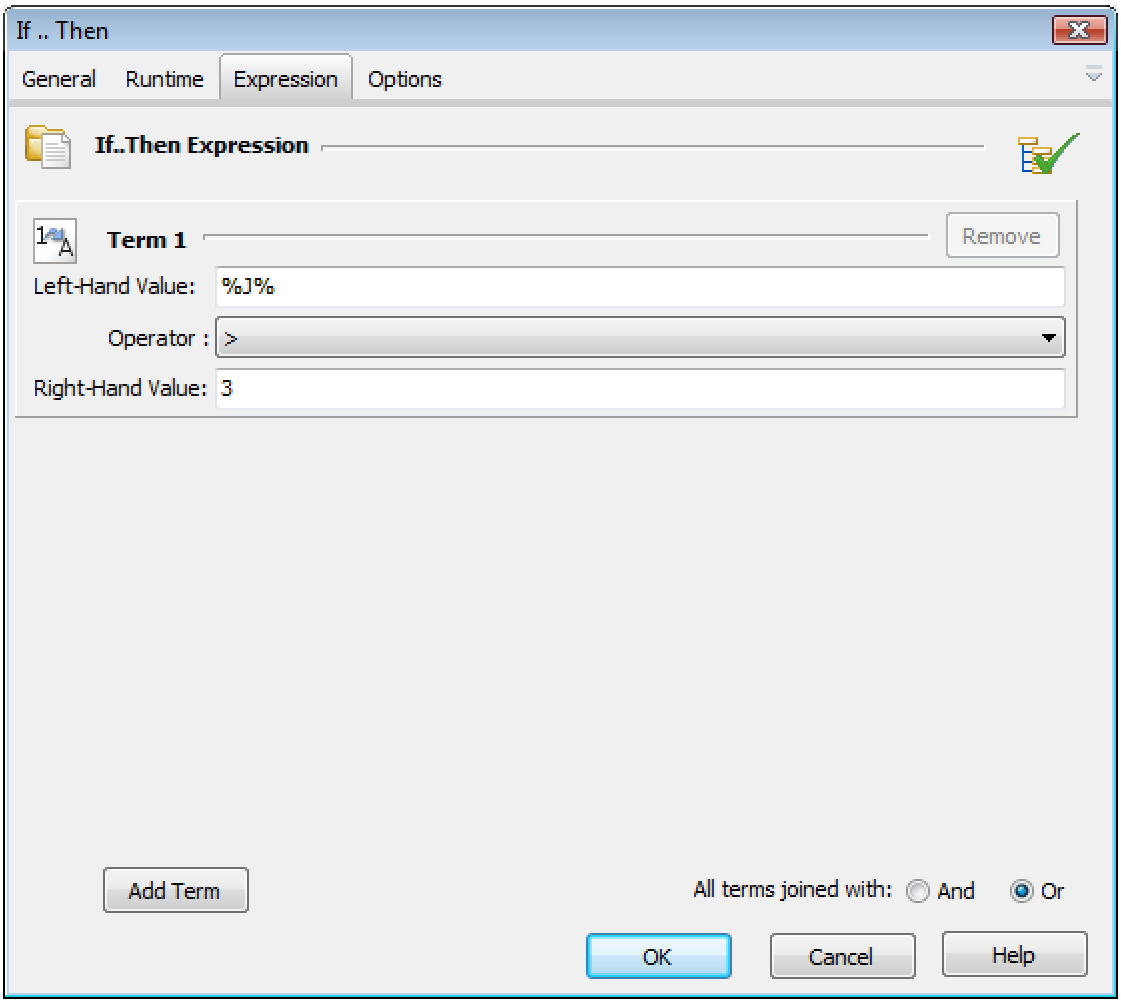
TIP: If you use Automise variables for any or all of 'to', 'step' and 'loop value' values, then it is possible to change the values (either from actions or scripts) while the loop is running.

6.10.7 If .. Then Action

This action uses a boolean expression to determine if its child actions should run or not.

The Help and Manual action and the Copy Files(s) action will only be run if the expression:
%Build_Help% = True and %MajorVersion% > 2
The "Do the next thing" action will be run regardless of whether the If..Then action runs it's child actions or not.

If..Then Properties Dialog



If .. Then

Left-Hand Value: %J%

Operator : >

Right-Hand Value: 3

All terms joined with: ○ And ● Or

Each term is comprised of a left hand side, a right hand size, and a comparison operator. Use %VariableName% syntax to compare the values of variables. String, integer or floating point values can all be compared.

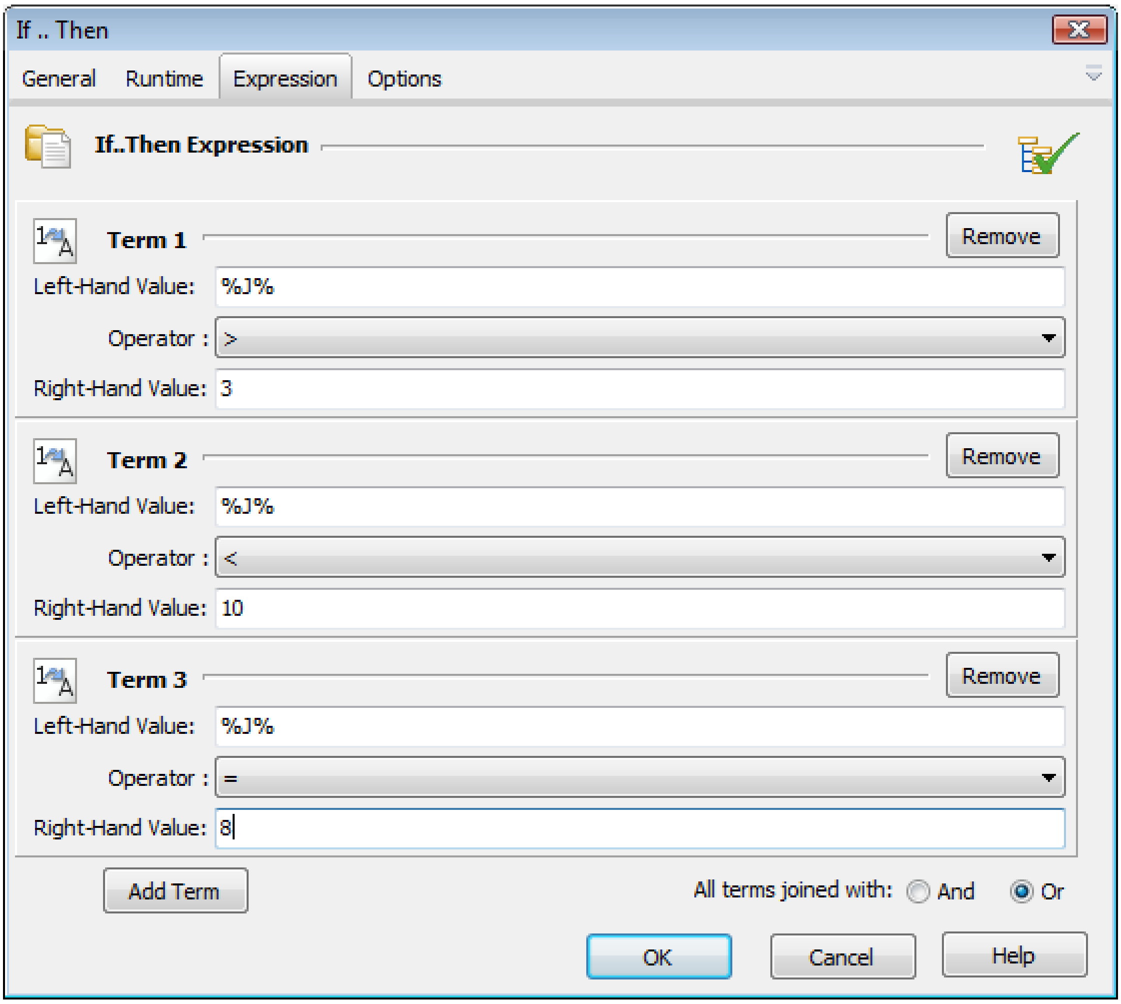
Each expression can have an unlimited number of terms. Click on the "Add Term" button to add a new term. Terms can be joined together with boolean "And" or "Or" directives.

The available boolean operators are:

- Equal =
- Not Equal <>
- Less Than <
- Less Than or Equal <=
- Greater Than >
- Greater Than or Equal >=
- Matches Regular Expression
- Is Substring Of
- Is Not Substring Of
- Is Member of Comma-Separated List
- Is Not Member of Comma-Separated List

Example

The following If... Then action will execute its children when variable J is equal to 4, 5, 6, 7 or 9:



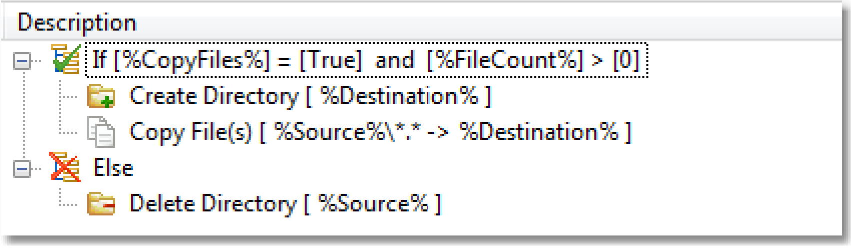
Term 1
Left-Hand Value: %J%
Operator : >
Right-Hand Value: 3

Term 2
Left-Hand Value: %J%
Operator : <
Right-Hand Value: 10

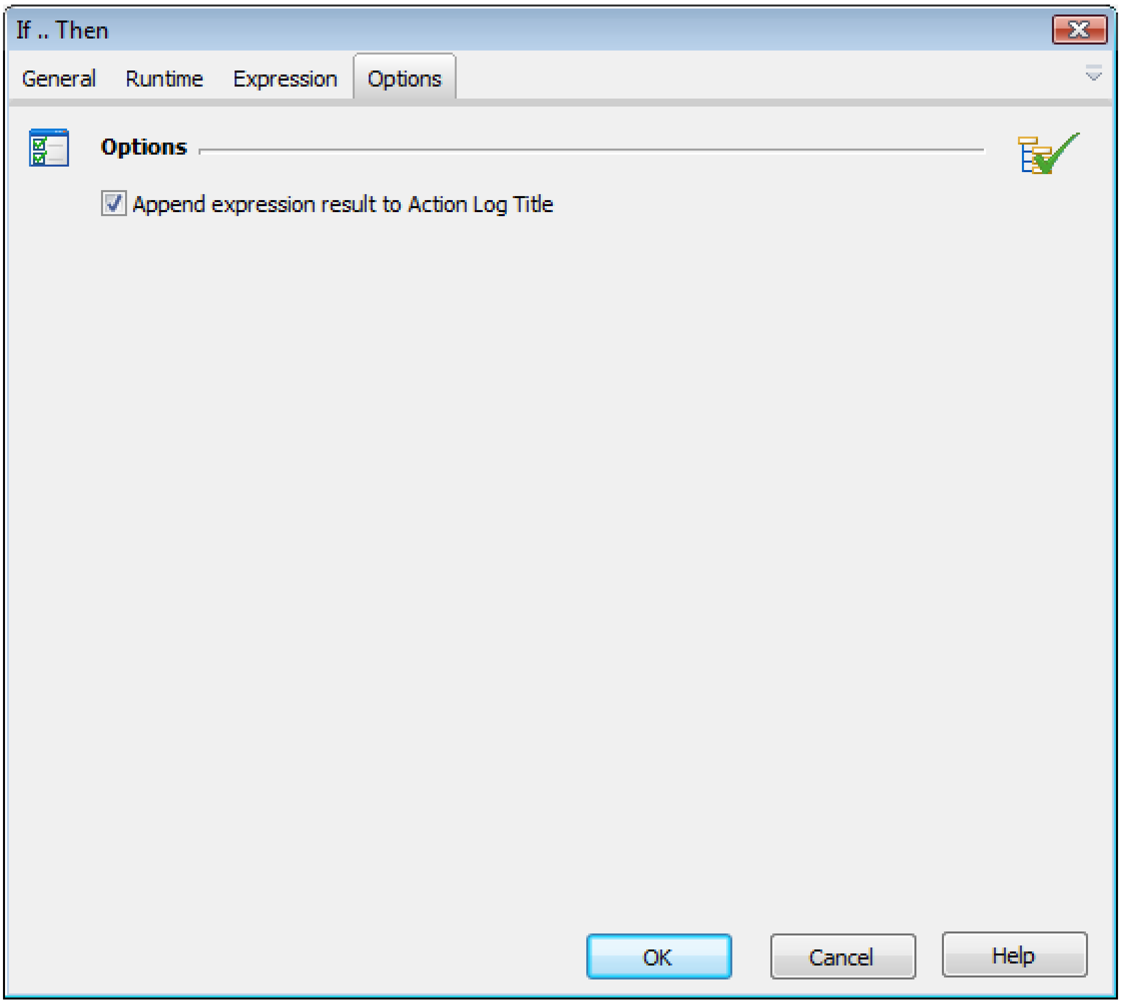
Term 3
Left-Hand Value: %J%
Operator : =
Right-Hand Value: 8

All terms joined with: ○ And ● Or

You can also use If..Then with the Else action, to provide an alternative set of actions to run. See the Else Action for details.



```
Description
⊟ If [%CopyFiles%] = [True]  and  [%FileCount%] > [0]
    Create Directory [ %Destination% ]
    Copy File(s) [ %Source%\*.* -> %Destination% ]
⊟ Else
    Delete Directory [ %Source% ]
```



☑ Append expression result to Action Log Title

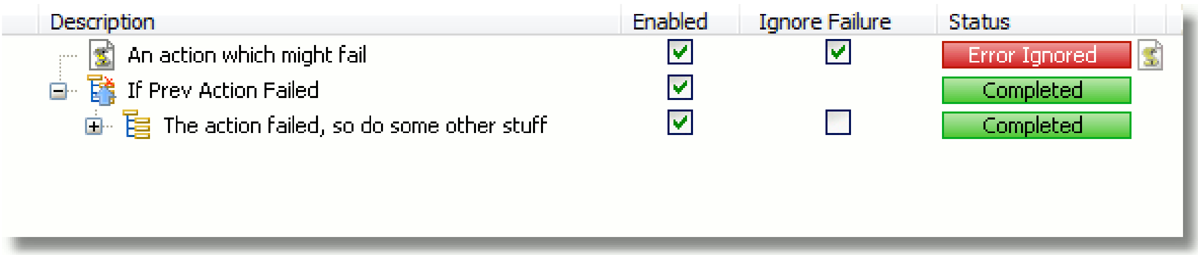
The "Append expression result to Action Log Title" will show the result (True or False) as part of the action's log title when it runs. The result is appended to the normal log title.

See also

If Prev Action Failed Action

6.10.8 If Prev Action Failed Action

This action will execute it's child actions only if the previous sibling action failed. For this to happen the previous sibling action needs Ignore Failure enabled:



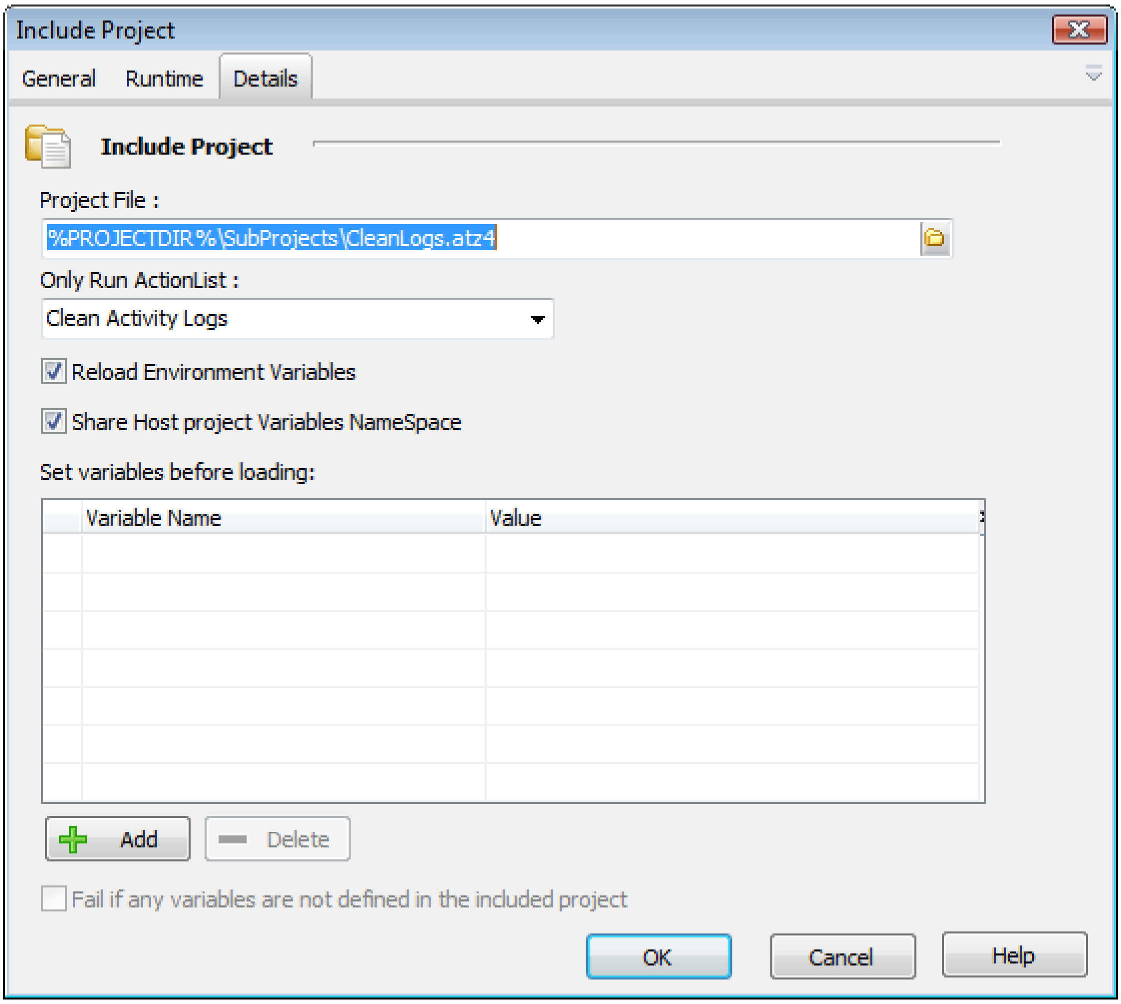
Description	Enabled	Ignore Failure	Status
An action which might fail	☑	☑	Error Ignored
If Prev Action Failed	☑		Completed
The action failed, so do some other stuff	☑	☐	Completed

You can also use execute a set of actions if the previous action succeeds (see the Else Action.)

For more sophisticated error handling and control, you may wish to look at the Try and Catch actions.

6.10.9 Include Automise Project

This action type allows you to include other Automise projects in your project. This allows you to modularise your project by including subprojects, as an alternative to using many Action Lists.



Include Project

Project File : %PROJECTDIR%\SubProjects\CleanLogs.atz4

Only Run ActionList : Clean Activity Logs

☑ Reload Environment Variables

☑ Share Host project Variables NameSpace

Set variables before loading:

Variable Name	Value

☐ Fail if any variables are not defined in the included project

Project File

Specify the project file to include.

Only Run ActionList

Normally, the Main Action List of the included project will be run. This combo box allows you to specify a different action list to run.

Note that Action List Parameters cannot be set.

Reload Environment Variables

The "Reload Environment Variables" forces Automise to reload the environment variables before executing the included project, environment variables are usually only loaded at startup.

Share Host Variables Namespace

If the "Share Host project Variables Namespace" option is checked, then the host project and the Included project will share one variable namespace.

This option should be used with caution. When the included project is executed, any project variables of the included project are loaded at that time. If there are variable name clashes then the existing project variable of the host project will be kept. This can lead to unexpected results. When this option is not enabled, the host and the included projects each have their own variable namespace.

Set Project Variables in Included Project

If the variables namespace is not shared, it is possible to define individual variables to set in the included project. These can be used similarly to Action List Parameters. Specify the name of any variables in the included project, and the values to set them to. Note that if the values reference any variables, these are variables from the parent project namespace, not the included project namespace.

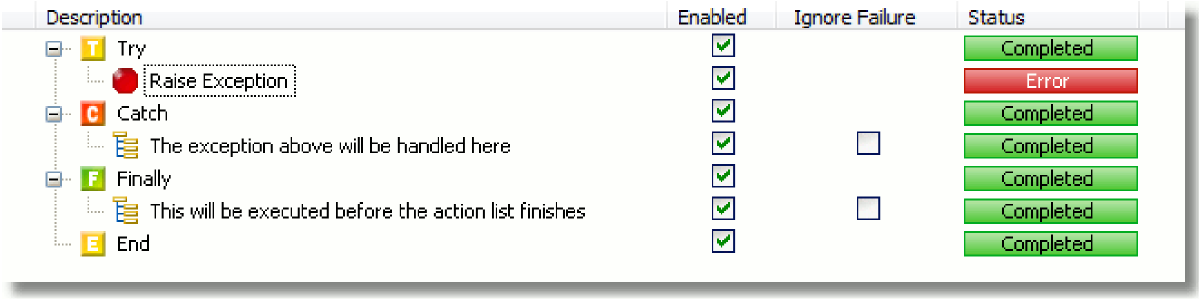
(If you wish to set variables in a shared host namespace, you can use Set Variables actions prior to the Included Project.)

Fail if Any Variables Do Not Exist in the Included Project

If this option is set, the action will fail if any variables to be set do not exist in the included project. Otherwise, a warning will be shown but the project will execute.

6.10.10 Raise Exception

The raise exception action will cause an error condition to be set when it is executed. The flow of the project will then change as if an error had occurred.



Description	Enabled	Ignore Failure	Status
T Try	☑		Completed
Raise Exception	☑		Error
C Catch	☑		Completed
The exception above will be handled here	☑	☐	Completed
F Finally	☑		Completed
This will be executed before the action list finishes	☑	☐	Completed
E End	☑		Completed

The flow will depend on if the action is enclosed in any Try blocks, and if the error isn't handled by an EXCEPT block then the flow will switch to the OnFailure action list.

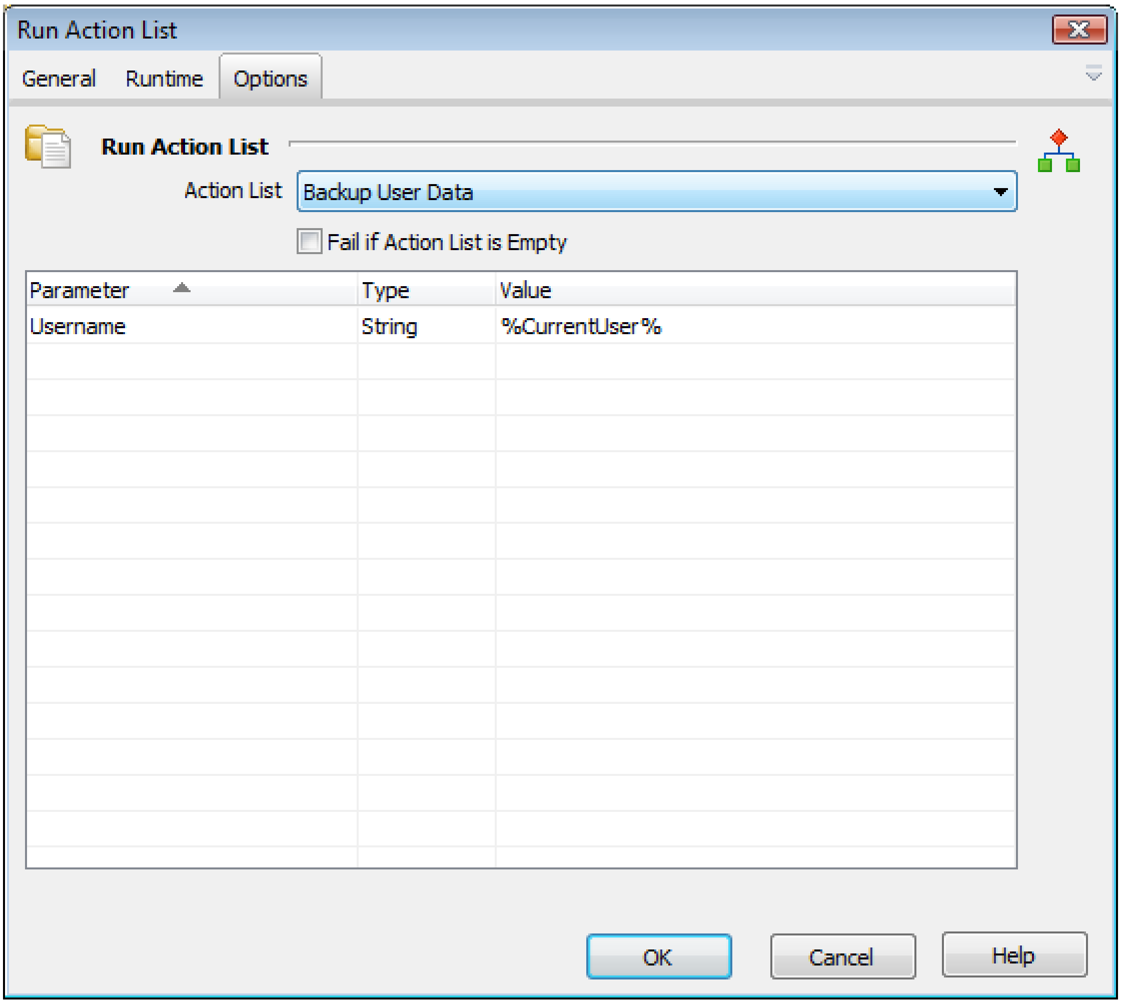
In the example above, the project stopped with status "Succeeded" because the exception was handled in the Try/Catch block.

6.10.11 Run Action List Action

This action enables you to run a different Action Lists. Running another action list is like calling a sub procedure - after the actions in the action list run, then the project continues on to the next action after the Run Action List action.

You can add new Actions Lists from the Project Menu.

NOTE: you should take care to avoid circular references between action lists - Automise does not check for recursion, so Action Lists may be used recursively, but doing so may cause your project to infinitely recurse.



Run Action List

Action List: Backup User Data

□ Fail if Action List is Empty

Parameter	Type	Value
Username	String	%CurrentUser%

See Also

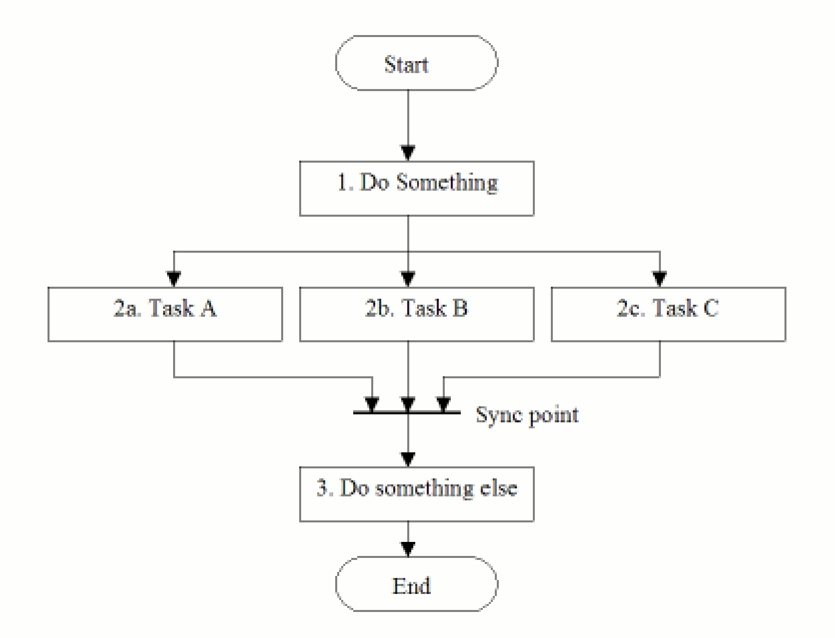
Action Lists
Action List Parameters

6.10.12 Running actions in Parallel

[Automise Professional Edition]

Motivation

Automise allows you to structure your project so that multiple actions can be run simultaneously. A graphical representation of multiple actions running in parallel is represented by this flow diagram:



After Task 1 finishes, tasks 2a, 2b and 2c all run simultaneously. Only when 2a, 2b and 2c all finish (the sync point), can task 3 can run.

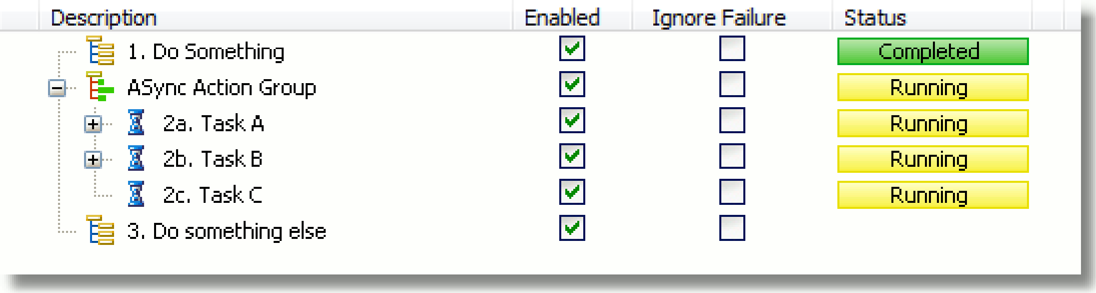
Async Action Groups

To run actions in parallel use the "ASync Action Group" action (in the Flow Control category).

ASync Action Group

This action looks like a normal Action Group action, except that each of its immediate children are run simultaneously. When all the parallel child actions have completed, the ASync Action group finishes and control continues sequentially (this is the Sync Point in the above diagram.)

The flow diagram above can be represented in Automise as follows:

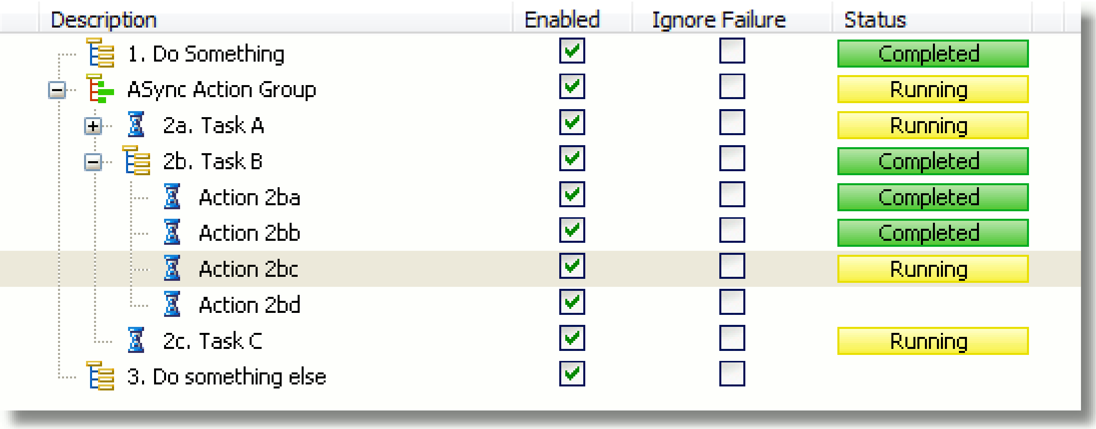


Description	Enabled	Ignore Failure	Status
1. Do Something	☑	□	Completed
ASync Action Group	☑	□	Running
2a. Task A	☑	□	Running
2b. Task B	☑	□	Running
2c. Task C	☑	□	Running
3. Do something else	☑	□	

In the screenshot above, Task 1 has already completed and now all 3 of the parallel tasks are running at once. When all the parallel tasks have completed, Task 3 will run.

Grouping actions under Async

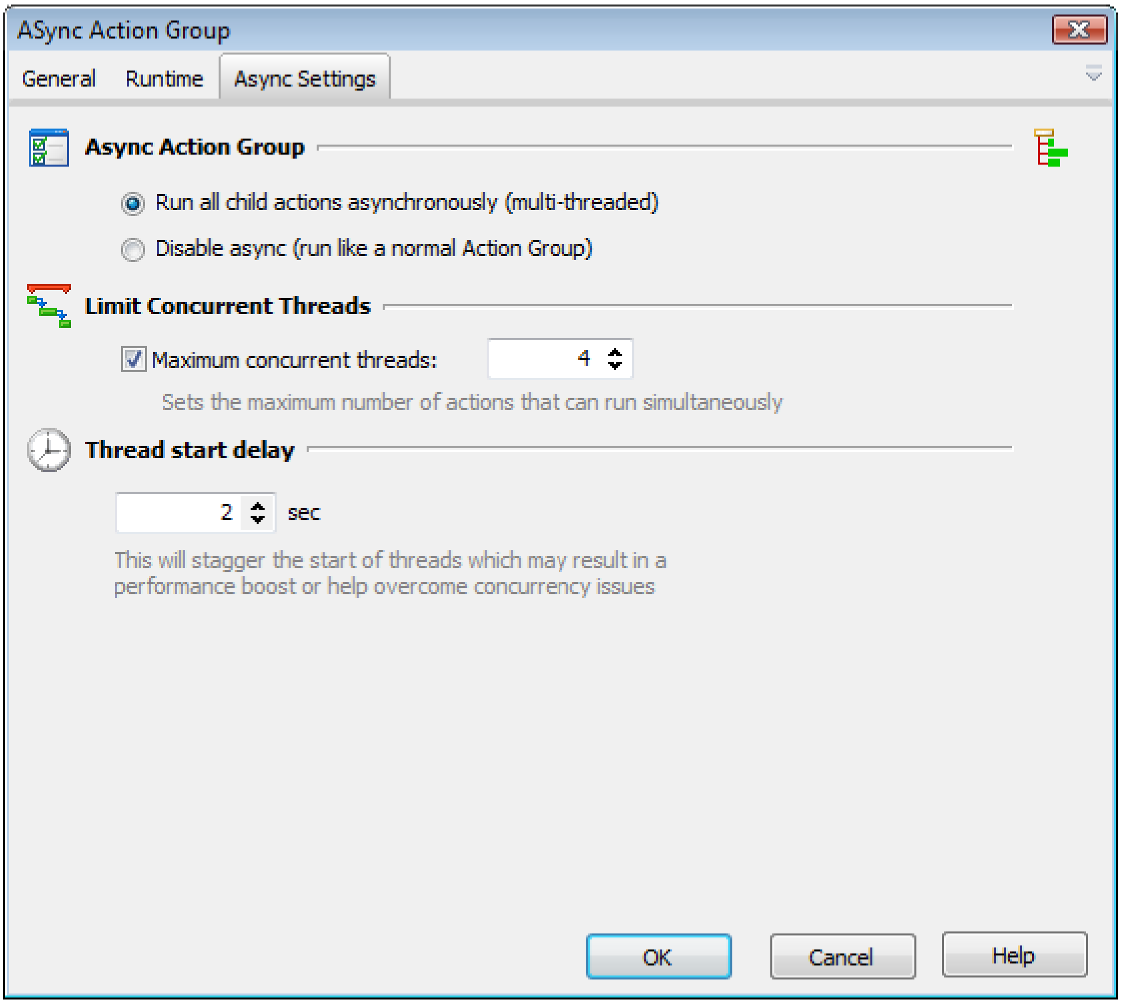
If actions under an ASync group have child actions, then these actions will run sequentially in the same thread as the parent (ie not asynchronously) and follow the normal processing rules:



Description	Enabled	Ignore Failure	Status
1. Do Something	☑	□	Completed
ASync Action Group	☑	□	Running
2a. Task A	☑	□	Running
2b. Task B	☑	□	Completed
Action 2ba	☑	□	Completed
Action 2bb	☑	□	Completed
Action 2bc	☑	□	Running
Action 2bd	☑	□	
2c. Task C	☑	□	Running
3. Do something else	☑	□	

(In the example above, task 2b has 4 sequential child actions. The first and second have completed, and the third is currently running.) Tasks 2a and 2c are running in parallel.

Async Action Group Properties



Async Action Group

● Run all child actions asynchronously (multi-threaded)

○ Disable async (run like a normal Action Group)

Limit Concurrent Threads

☑ Maximum concurrent threads: 4

Sets the maximum number of actions that can run simultaneously

Thread start delay

2 sec

This will stagger the start of threads which may result in a performance boost or help overcome concurrency issues

You can choose to Disable Async to have the Async Action Group behave like a normal (sequential) action group. This may be useful for debugging purposes, or for running the project on hardware which may not perform faster with multiple actions running simultaneously.

Setting a thread start delay allows you to "stagger" the start of the threads, which may give a performance boost or help overcome concurrency issues. Starting from the second child of the Async Action Group, each child will be started the specified number of seconds after the previous async child.

If you would like to limit the number of simultaneously running actions, then set the maximum concurrent threads. It must be at least 2, otherwise the Async group will behave like a normal Action Group.

Getting the Most from Async Action Groups

The purpose of ASync Action Group is to allow your project to run multiple actions simultaneously to reduce the total running time. This is most beneficial under one or more of the following conditions:

- More than one CPU (or more than one CPU core) is available.
- "IO bound" actions can be run together with "CPU bound" actions.
- Slow network-bound actions, or other non-CPU intensive actions, can be run together with CPU intensive actions.

It is entirely possible to slow the project down by running certain actions together, for example if you have two IO bound actions then running them together will may result in an overall reduction in performance. Similarly, it is not recommended that the number of CPU bound actions running in parallel exceeds more than one greater than the number of available CPU cores (and, in some cases, less.)

You can even run ASync Action groups inside ASync Action groups. Again, it is recommended that caution is exercised when doing this.

Limitations

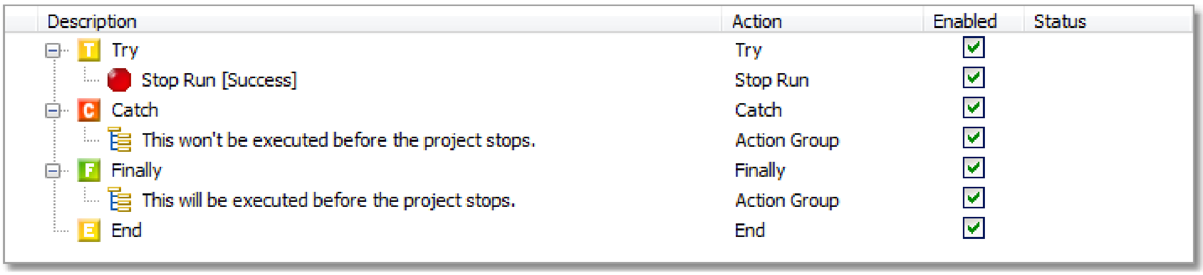
- You cannot have any interactive actions which require user input (eg. Prompt for Variables) under Async Action Groups.

Things to be aware of for actions running in parallel

- Reading and writing the same Automise variables (if running the same action list multiple times asynchronously, use Action List Parameters.)
- Reading and writing the same files.
- Reading and writing the same registry values.
- Some compilers (such as the Delphi compiler) and some other tools may lock certain files and a second instance may not succeed.
-

6.10.13 Stop Run Action

The "Stop Run" action will cause the project to stop running.



Description	Action	Enabled	Status
Try	Try	☑	
Stop Run [Success]	Stop Run	☑	
Catch	Catch	☑	
This won't be executed before the project stops.	Action Group	☑	
Finally	Finally	☑	
This will be executed before the project stops.	Action Group	☑	
End	End	☑	

The **Stop Run Result** property determines if the project stops with success or failure. If **Stop Run Result** is set to **Failure,** then the OnFailure List will execute.

Note that if the Stop Action is inside a Try/Finally block the Finally block will still execute.

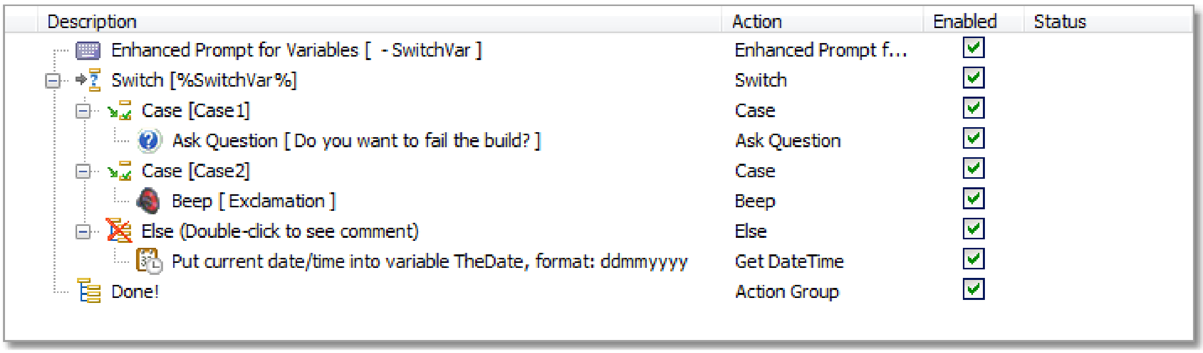
See Also

Exit Action List Action

6.10.14 Switch Action

This action (along with the Case & Else actions) provides a simple selector based on a simple case insensitive string comparison. The Case actions need to be child actions of the Switch action. When a case value matches the switch value the child actions of that Case action will be executed.

Below is an example using Switch, Case, and Else.



Description	Action	Enabled	Status
Enhanced Prompt for Variables [- SwitchVar]	Enhanced Prompt f...	☑	
Switch [%SwitchVar%]	Switch	☑	
Case [Case1]	Case	☑	
Ask Question [Do you want to fail the build?]	Ask Question	☑	
Case [Case2]	Case	☑	
Beep [Exclamation]	Beep	☑	
Else (Double-click to see comment)	Else	☑	
Put current date/time into variable TheDate, format: ddmmyyyy	Get DateTime	☑	
Done!	Action Group	☑	

See Also

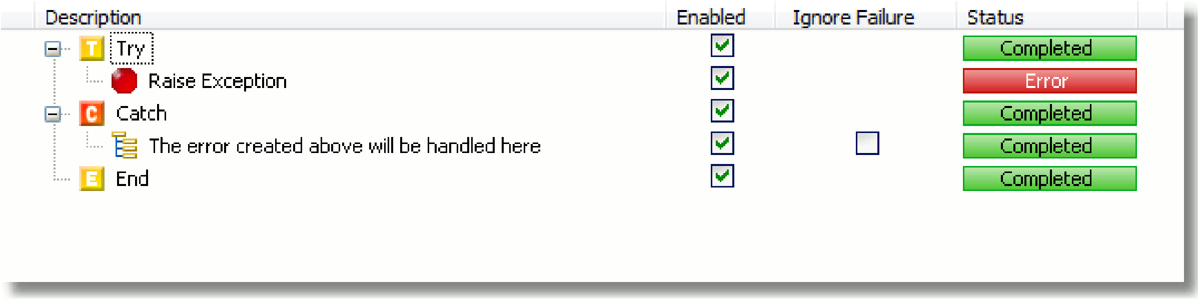
Else Action

6.10.15 Try/Catch/Finally/End Actions

The Try action, along with the Catch, Finally and End actions provide structured exception handling. They allow you to create localised error handling and resource protection, just as you do in programming languages such as C++, C#, Delphi etc.

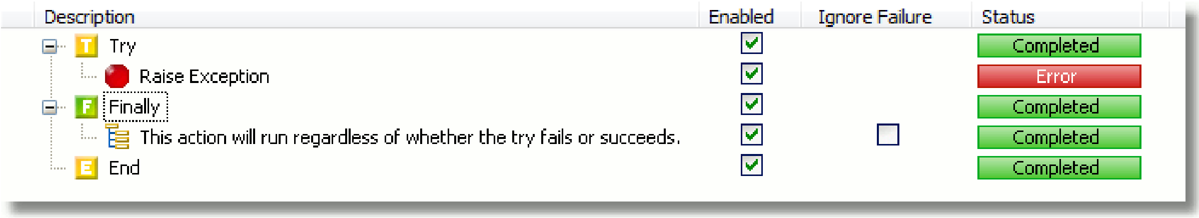
For each Try action, there should be a matching End Action at the same level. The Try must also have either a Catch Action or a Finally Action as it's next sibling. The actions can be used in the following combinations :

1. Try ... Catch block



Description	Enabled	Ignore Failure	Status
Try	☑		Completed
Raise Exception	☑		Error
Catch	☑		Completed
The error created above will be handled here	☑	□	Completed
End	☑		Completed

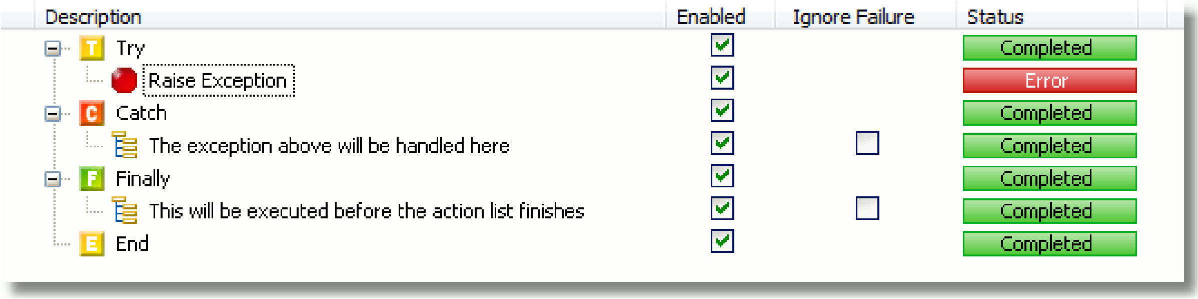
2. Try ... Finally block



Description	Enabled	Ignore Failure	Status
Try	☑		Completed
Raise Exception	☑		Error
Finally	☑		Completed
This action will run regardless of whether the try fails or succeeds.	☑	□	Completed
End	☑		Completed

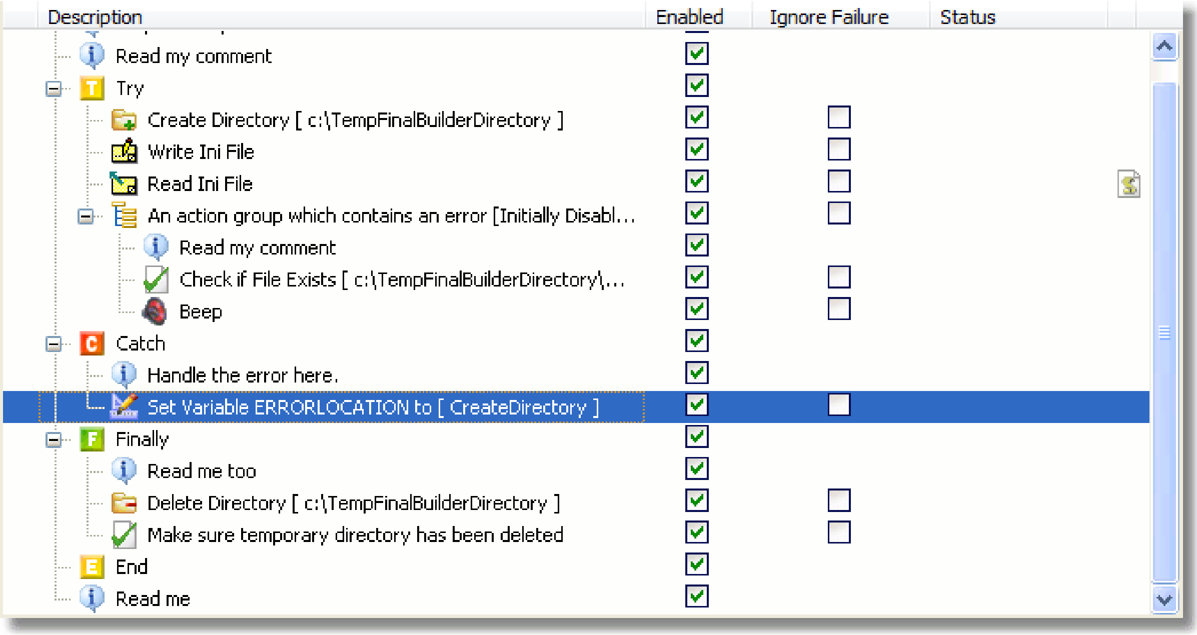
A Finally Action will almost always execute it's child actions, the exception being if there is a structural error with the try or catch (ie. a missing end, or finally before catch etc).

3. Try ... Catch ... Finally block



Description	Enabled	Ignore Failure	Status
Try	☑		Completed
Raise Exception	☑		Error
Catch	☑		Completed
The exception above will be handled here	☑	□	Completed
Finally	☑		Completed
This will be executed before the action list finishes	☑	□	Completed
End	☑		Completed

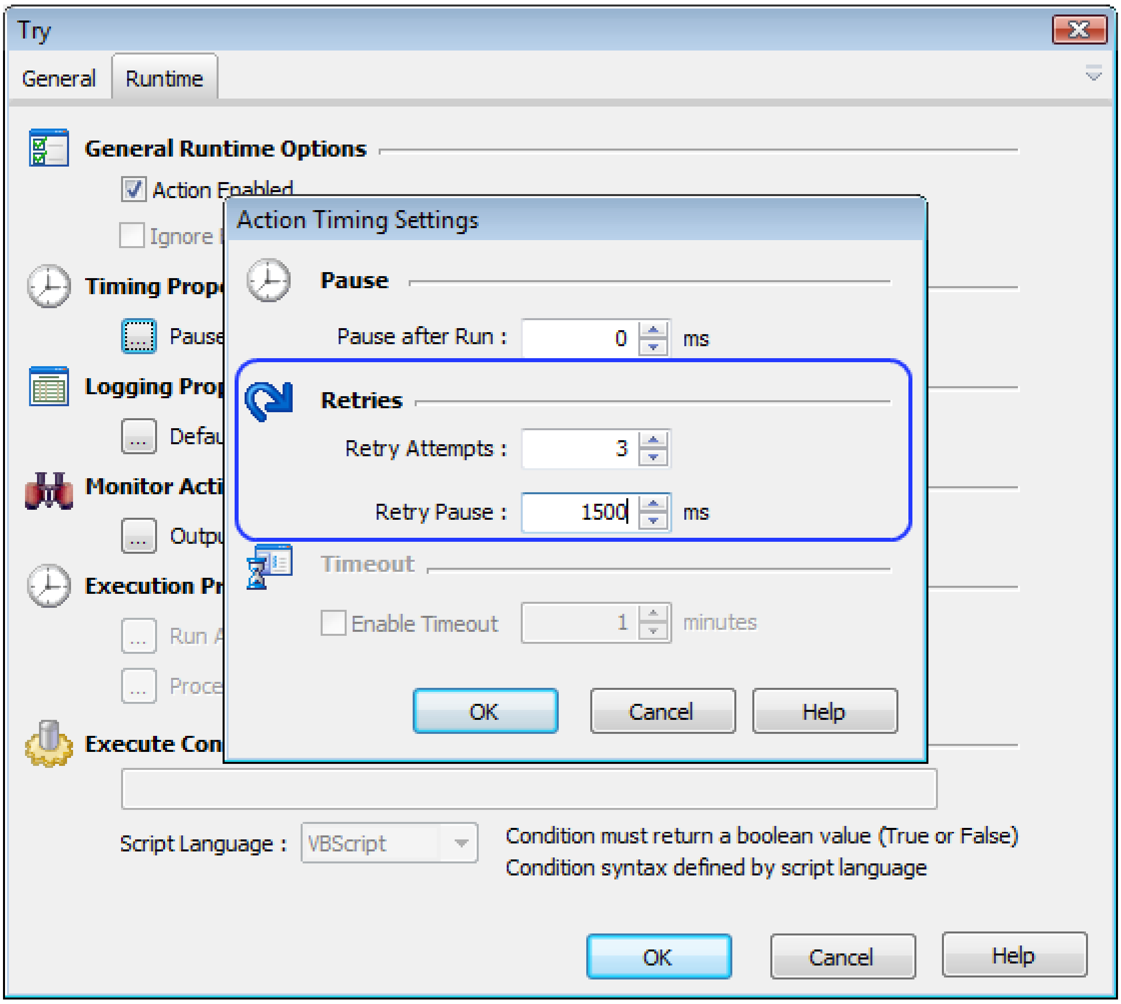
A more complete example:



Description	Enabled	Ignore Failure	Status
Read my comment	☑		
Try	☑		
Create Directory [c:\TempFinalBuilderDirectory]	☑	□	
Write Ini File	☑	□	
Read Ini File	☑	□	
An action group which contains an error [Initially Disabl...	☑	□	
Read my comment	☑		
Check if File Exists [c:\TempFinalBuilderDirectory\...	☑	□	
Beep	☑	□	
Catch	☑		
Handle the error here.	☑		
Set Variable ERRORLOCATION to [CreateDirectory]	☑	□	
Finally	☑		
Read me too	☑		
Delete Directory [c:\TempFinalBuilderDirectory]	☑	□	
Make sure temporary directory has been deleted	☑	□	
End	☑		
Read me	☑		

Retrying Try Actions

All actions have the option to retry multiple times before failing outright. Try action retries work differently to normal action retries. If you set a number of Retry Attempts on a Try action, the entire try block (ie all of the child actions) will be retried if one of the actions from that block fails.



☑ Action Enabled

□ Ignore

Pause after Run : 0 ms

Retry Attempts : 3

Retry Pause : 1500 ms

□ Enable Timeout 1 minutes

Script Language : VBScript

Condition must return a boolean value (True or False)
Condition syntax defined by script language

The Catch (or Finally) section of the Try/Catch/Finally will not be run until all the retries have failed.

Warning: Retrying Try Actions may give unpredictable behaviour when used inside Async Action Groups. Specifically, retries will be unpredictable if multiple action lists are running concurrently. If you are running Try blocks inside concurrent action lists (ie the same Try block in the same action list being run multiple times), then the Try block will be retried an unpredictable number of times. See Known Problems for details.

6.10.16 Waiting For Events (Wait For actions)

The "Wait For..." actions allow you to pause your project until a particular condition is met.

Alternatives to "Wait For..."

This can be accomplished by using other actions, for instance the While Loop action, to repeat a series of actions until something happens. However, "Wait For..." actions provide a more convenient package, and have the advantage of only logging data to the Automise log when something actually happens.

Polling Type

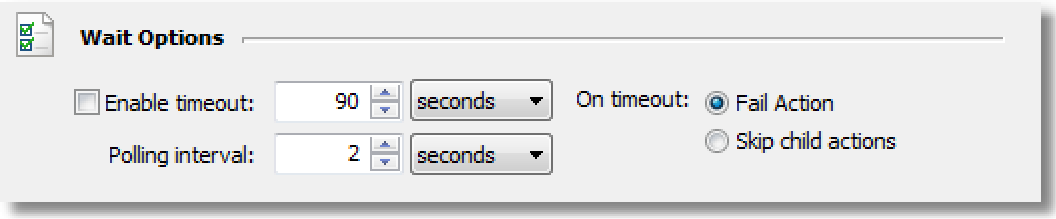
Note that the "Wait For..." action is a "busy-waiting" or "polling" wait - a particular wait condition is checked every so many seconds to see if it has changed.

Concurrency Limitations with Wait For Actions

It is up to the user to ensure proper synchronisation between a "Wait For" action and the rest of the project/system. Even though a certain condition may be true when a "Wait For..." action has unblocked and continued, there is no guarantee that the condition is necessarily still true when the next action runs. Another action (or program) may change the state of the system in-between the two actions running, and it is up to the user to ensure this cannot happen.

Wait Options

All Wait For events have the following options in their property pages:



Wait Options

□ Enable timeout: 90 seconds

Polling interval: 2 seconds

On timeout: ● Fail Action ○ Skip child actions

Enable Timeout

The action will time out if it has been running for this length of time without succeeding. Timeouts can be set in seconds, minutes, or hours. The action can be set to Fail outright or merely skip its child actions if it times out.

On Timeout, the action's OnTimedOut Script Event will be executed. By changing the "Fail" parameter inside the OnTimedOut event, a script can choose to fail or skip child actions on a case-by-case basis.

Polling Interval

The action will pause for this long in between each poll of its given wait condition. The polling interval can be set in seconds, minutes, or hours. Regardless of polling interval, the action polls immediately when it is first run.

If the polling interval is set to zero, the action will poll as quickly as it can without pausing.

(Pausing between polling intervals will be interrupted in the case that the action times out or the user stops the project run.)

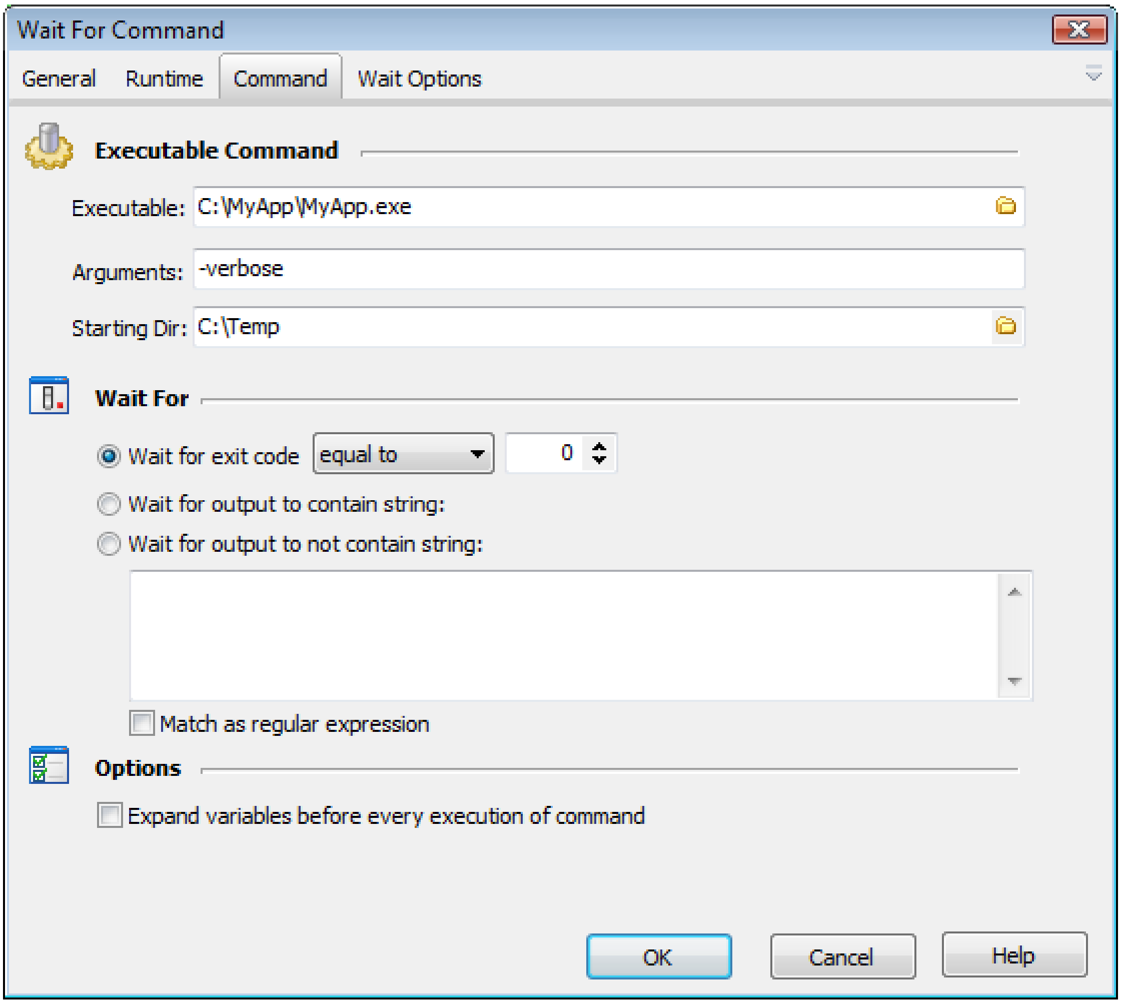
Timing Limitations

Note that the Polling Interval is not a polling frequency - it does not guarantee that the action will poll every X seconds, it only guarantees that the action will pause for X seconds in between each poll. In reality, the action will poll every X+(Length of last polling event) seconds, and in some actions (ie Wait For Remote Computer) the polling event itself may take several seconds.

6.10.16.1 Wait For Command

The Wait For Command action allows you to execute an external process and wait until it exhibits certain behaviour, or returns a certain value. For an overview of "Wait For..." events, including specifics of the Wait Options, see the Waiting For Events topic.

(Note that because some executables may use a large amount of system resources, you may wish to set the polling interval on the Wait Options to a high value.)



Executable: C:\MyApp\MyApp.exe

Arguments: -verbose

Starting Dir: C:\Temp

● Wait for exit code equal to 0

○ Wait for output to contain string:

○ Wait for output to not contain string:

□ Match as regular expression

□ Expand variables before every execution of command

Executable Command

Specify the details of the executable to run. You can specify a starting directory, and any command line arguments you wish to pass to the executable.

Note that even if the Wait For action times out, the executable program will not be terminated if it is hung. If a hung process is a likely possibility, it is recommended to use Wait For Process (with timeout) and a WMI Kill Process action in an Async Action Group, alongside the Wait For Command.

"**Wait for exit code...**"

You can wait for a single exit code value, or a range of exit codes. Normally, a non-zero exit code indicates an error. However, many applications do not return meaningful exit codes. See the documentation for the executable you want to run to find out if it returns exit code values.

"**Wait for output to contain string:**"
&
"**Wait for output to not contain string:**"

These two options will search for a substring in the output from the executable and continue if it is found (or not found.) Both StdOut and StdErr are captured.

Check the "**Match as regular expression**" box to match the substring as a regular expression.

"**Expand variables before every execution of command**"

If this option is set, variable names (ie %VariableName%) in the action parameters will be expanded every time the target command is run. Otherwise, they are only expanded once (when the action starts waiting.)

Setting this option allows you to change the parameters of the wait call while it is running. However, it should not be taken lightly. In particular, note that variable changes are not atomic. This means that while one variable value may have been updated, another may not yet have been updated when the executable is run. This can lead to unspecified behaviour. If you really want to change the parameters while running, a safer bet is to choose this option, but change the actual action properties from the OnCommandOutput script event - see below. This guarantees that all the arguments are changed at once.

Script Events

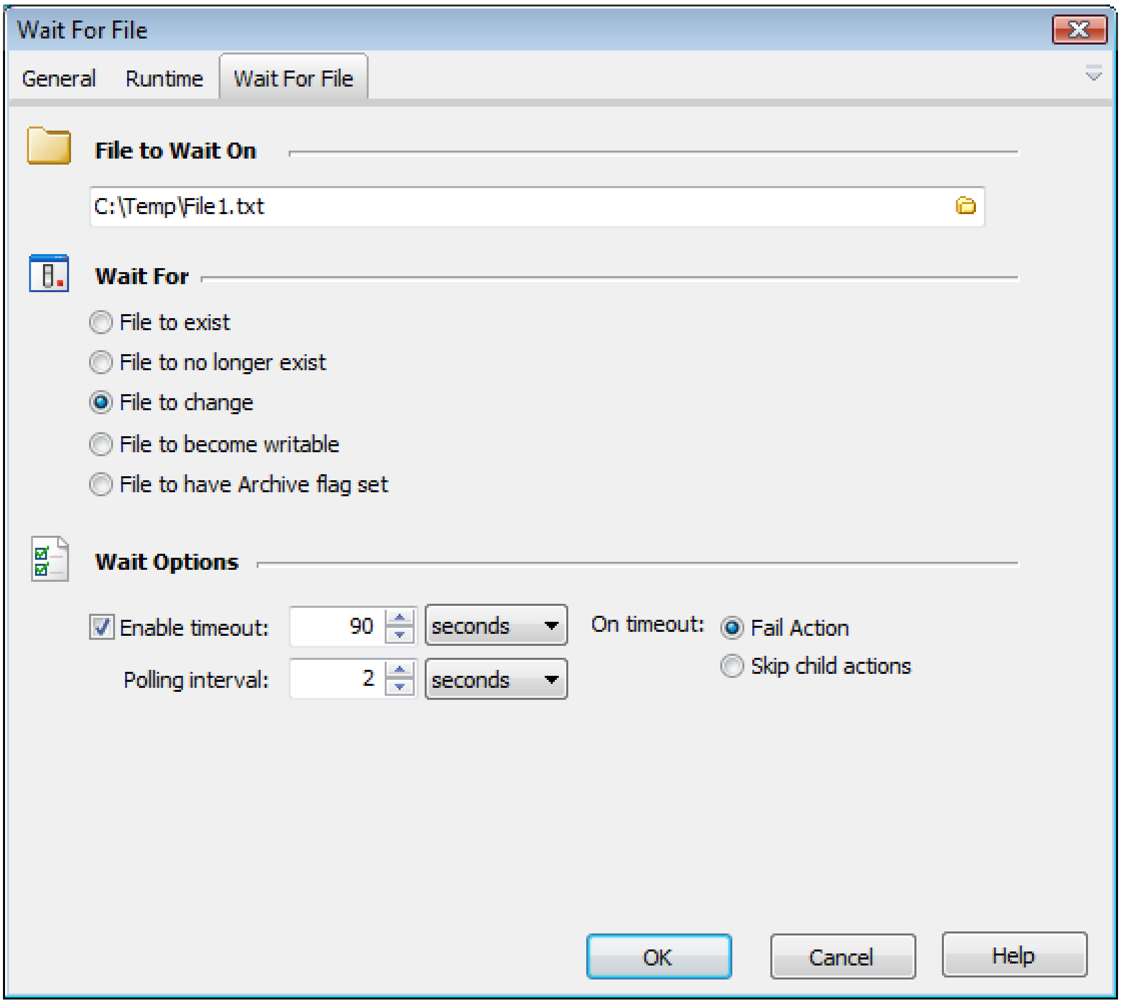
In addition to the normal Wait script events, Wait For Command provides an OnCommandOutput event. This script event allows you to do more advanced parsing of the command output (available in the Output parameter) or the exit code (available in the Exit Code parameter.) Set the Continue parameter to True in order to stop waiting and continue the project run.

(Tip: If you wish to depend solely upon the OnCommandOutput event for control of waiting, set the action to wait for an improbable exit code - ie -1.)

6.10.16.2 Wait For File

The Wait For File event pauses the project run until something happens to a file.

For an overview of "Wait For..." events, including specifics of the Wait Options, see this topic.



File to Wait On

C:\Temp\File1.txt

Wait For

○ File to exist
○ File to no longer exist
● File to change
○ File to become writable
○ File to have Archive flag set

Wait Options

☑ Enable timeout: 90 seconds
Polling interval: 2 seconds
On timeout: ● Fail Action ○ Skip child actions

You can choose to wait for one of the following conditions:

File to exist

The project run will continue when the given file has been created.

File to no longer exist

The project run will continue when the given file has been deleted or moved.

File to change

The project run will continue when either the given file's length or modification date has changed, relative to the length and modification date which were observed when the action started.

File to become writable

The project run will continue when the file has its Read Only flag cleared, and is not locked for write access.

In the example screenshot above, if Package.rar is being created when the action runs then the run will pause until the archiving process is finished and the file has been closed.

(Note that even though a file may be writable at the precise moment this action succeeds, it is up to you to ensure that another process does not lock it again before your action can open it.)

File to have Archive flag set

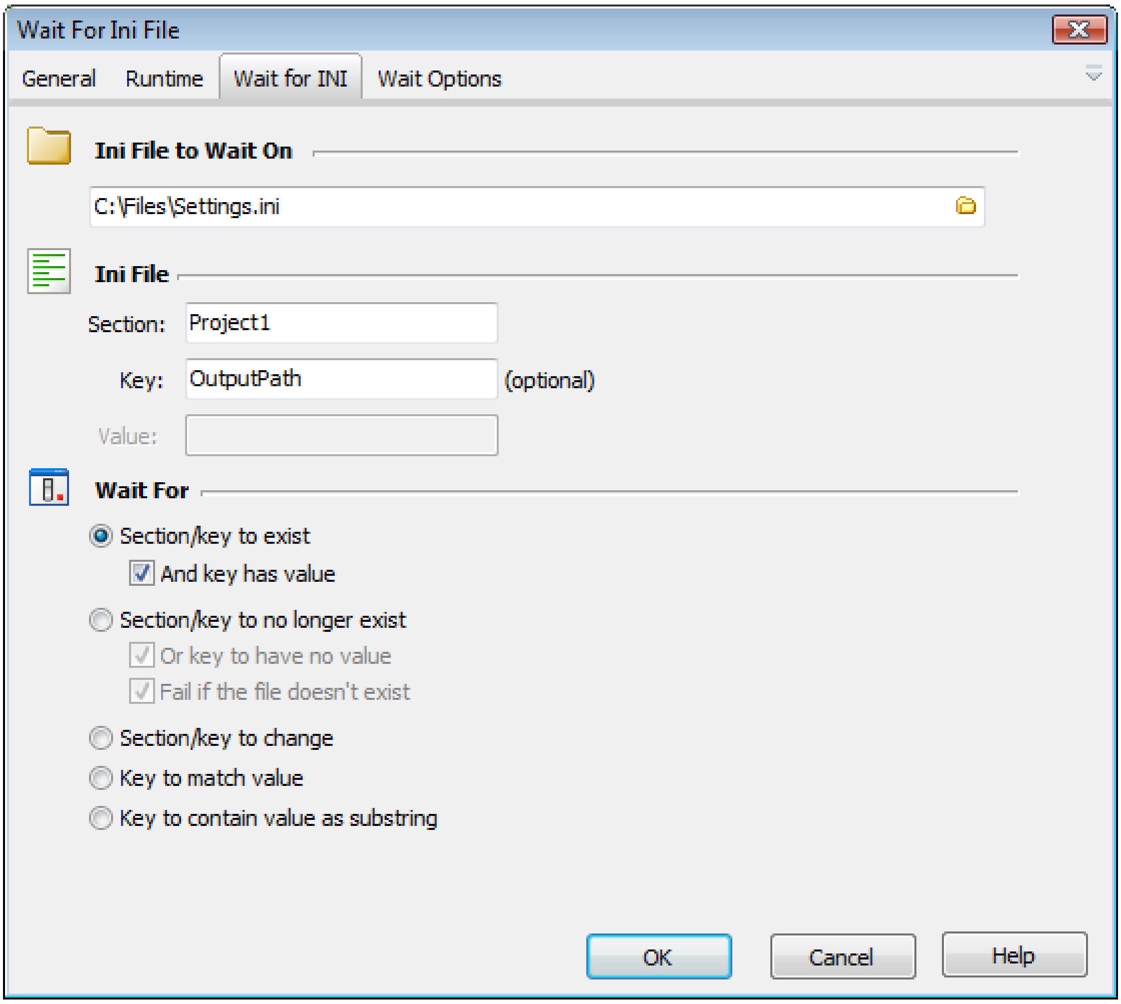
The project run will continue when the file has its Archive (A) bit set.

Wait Options

See the Waiting For Events topic.

6.10.16.3 Wait For Ini File

The Wait For Ini File action allows you to wait until a section or entry in an Ini File matches a specific condition. For an overview of "Wait For..." events, including specifics of the Wait Options, see this topic.



Wait For Ini File

Ini File to Wait On

C:\Files\Settings.ini

Ini File

Section: Project1

Key: OutputPath (optional)

Value:

Wait For

● Section/key to exist

☑ And key has value

○ Section/key to no longer exist

☑ Or key to have no value

☑ Fail if the file doesn't exist

○ Section/key to change

○ Key to match value

○ Key to contain value as substring

Ini File to Wait On

Specify the name of the Ini file to monitor for changes.

Ini File

Specify the section and (optionally) the key name to monitor for changes. If you are waiting for a certain value (or substring), specify it here too.

Wait For

"**Section/key to exist**"

Use this option to wait for either a section or a key (under a section) to appear in the Ini file. If you specify a key name, you can also check whether or not they key needs to take a value. Otherwise, 'KeyName=' will be considered as an existing key.

"**Section/key to no longer exist**"

Use this option to wait until a section or key no longer exists. If you specify a key name, check the "Or key to have no value" box if you want a blank value (ie 'KeyName=') to be a sufficient condition for continuing.

You can choose to have the action fail if the file doesn't exist, otherwise a file not found error will be considered as passing (if the file doesn't exist, the section or key doesn't exist either!)

"**Section/key to change**"

Use this option to wait until a section or a key changes.

If you specify a key, then the action will continue when the key is erased, created, or its value is changed.

If you only specify a section, the action will continue when a key is added or removed from the section, or when the value of any key in the section changes.

"**Key to match value**"

You need to specify a key name for this option.

The action will continue when the value for the key perfectly matches the specified value.

"**Key to contain value as substring**"

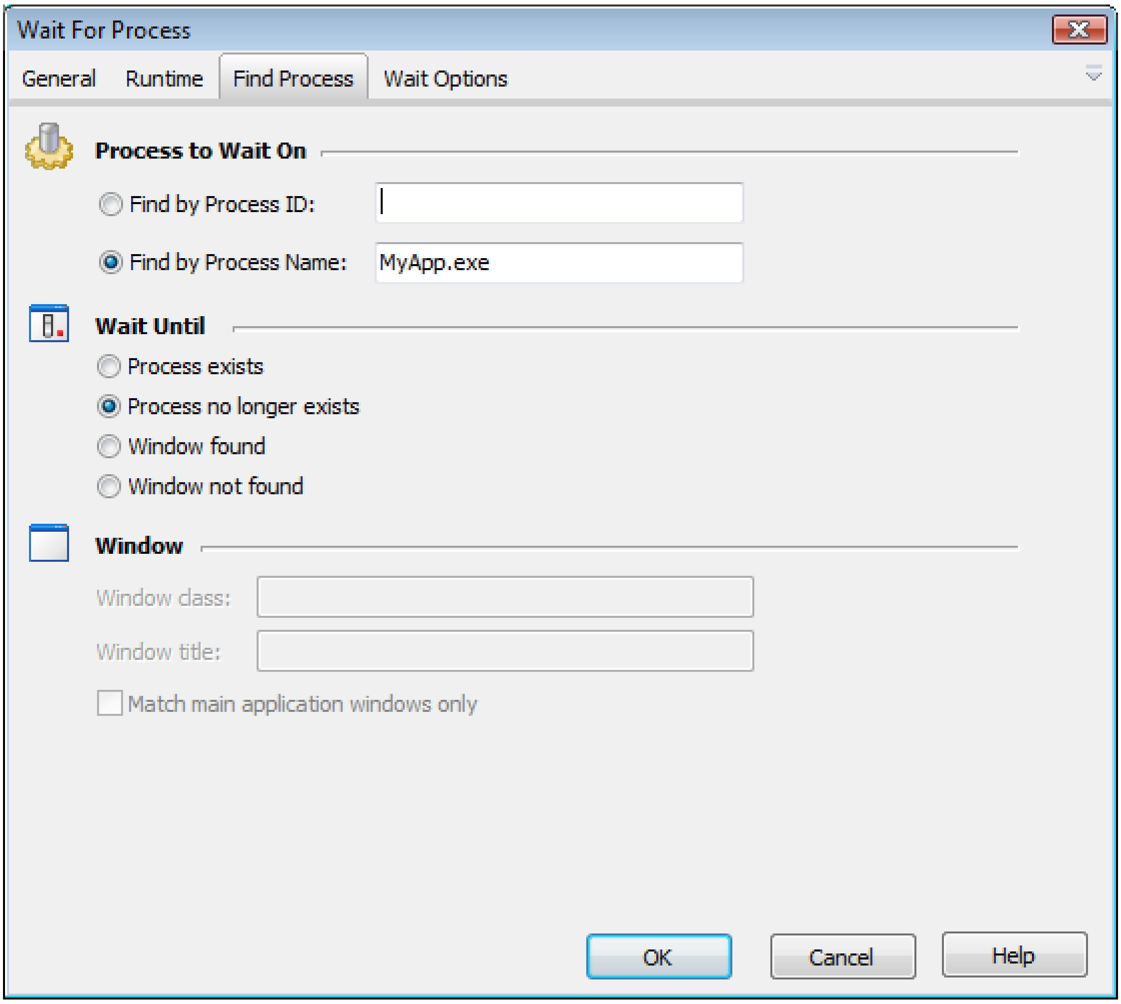
You need to specify a key name and a value to wait for.

The action will continue when the value for the key contains the substring you specify in the "Value" field.

6.10.16.4 Wait For Process

The Wait for Process action waits for a given process to start or finish, or alternatively for it to open or close a window.

For an overview of "Wait For..." events, including specifics of the Wait Options, see this topic.



Process to Wait On

○ Find by Process ID:

● Find by Process Name: MyApp.exe

Wait Until

○ Process exists

● Process no longer exists

○ Window found

○ Window not found

Window

Window class:

Window title:

□ Match main application windows only

Process to Wait On

The process can be specified by Process ID or name.

Wait Until

The wait action will block until one of the following happens:

Process exists - The process is running in the system.

Process no longer exists - The process has terminated (or, in the case of Process Name, no processes of the given name are running.)

Window found - A window matching the "Window" description has been found.

Window not found - No window matching the "Window" description was found.

Window

Windows can be specified by Window class, Window title, or both. (Window Classes can be determined using a Windows "spy" utility like Embarcadero's WinSight32 or Microsoft's Spy++.)

If both Window Class and Window Title are left blank then the Wait for Process action will match any window belonging to the relevant process.

Alternatively, if the process name and ID are left blank then the Wait for Process action will match windows that belong to any process.

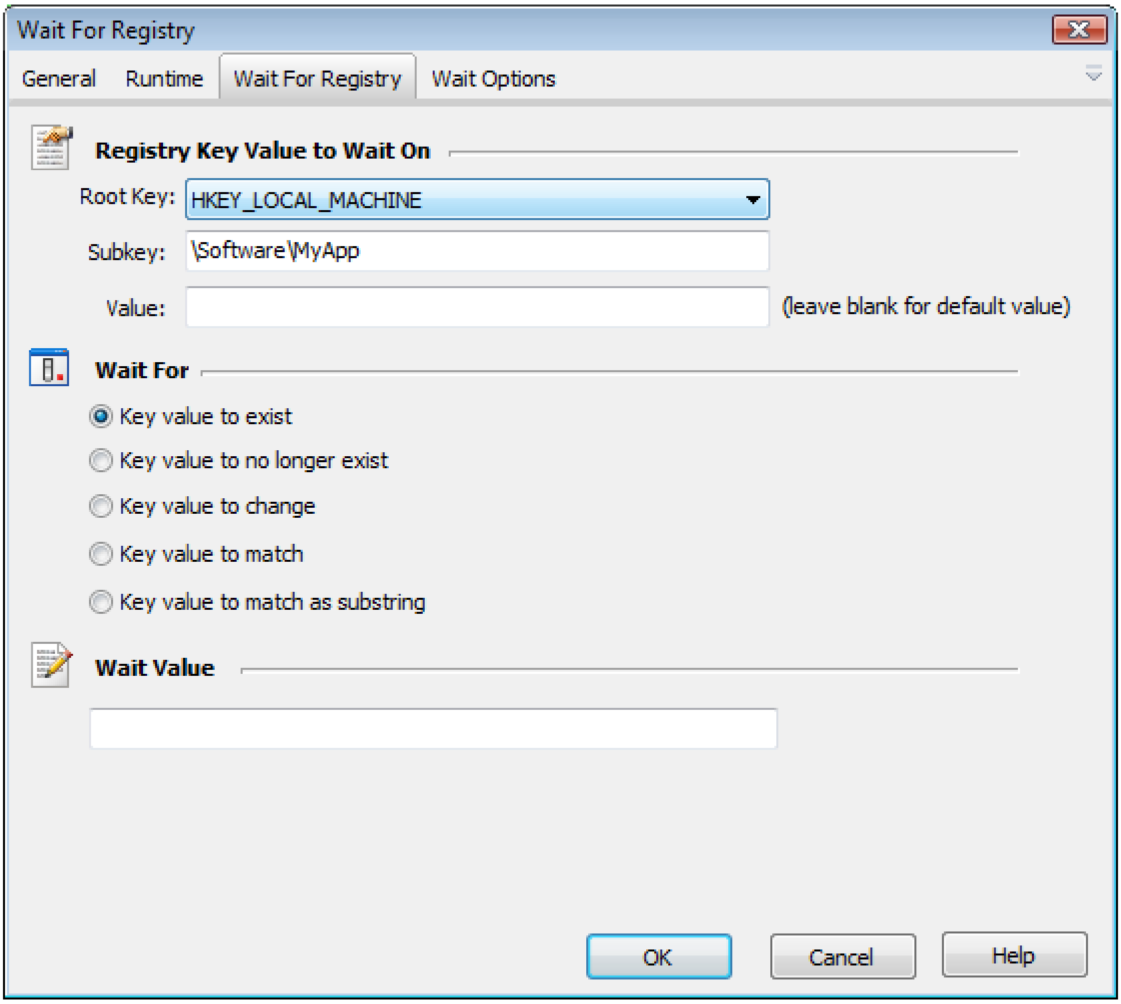
Check the *"Main application windows only"* box to only match main (ie parent) application windows. If the box is unchecked, all child windows will be searched.

Wait Options

See the Waiting For Events topic.

6.10.16.5 Wait For Registry

The Wait For Registry action allows you to wait until a change occurs in the Windows registry. For an overview of "Wait For..." events, including specifics of the Wait Options, see this topic.



Registry Key Value to Wait On

Root Key: HKEY_LOCAL_MACHINE

Subkey: \Software\MyApp

Value: (leave blank for default value)

Wait For

● Key value to exist

○ Key value to no longer exist

○ Key value to change

○ Key value to match

○ Key value to match as substring

Wait Value

Registry Key Value to Wait On

Specify a root key value, a subkey, and (optionally) a key value name. Leave the Value field blank to choose the (Default) value for the key.

Wait For

"**Key value to exist**"

The action will pause until the specified value (or key) exists.

"**Key value to no longer exist**"

The action will pause until the specified value (or key) no longer exists. A blank string value does not qualify as non-existence.

"**Key value to change**"

The action will pause until the key value has changed from the value read when the action begins waiting.

"**Key value to match**"

The action will pause until the key value until the key value matches the specified "Wait

Value".

"**Key value to match as substring**"

The action will pause until the key value contains the specified "Wait Value" as a substring.

Wait Value

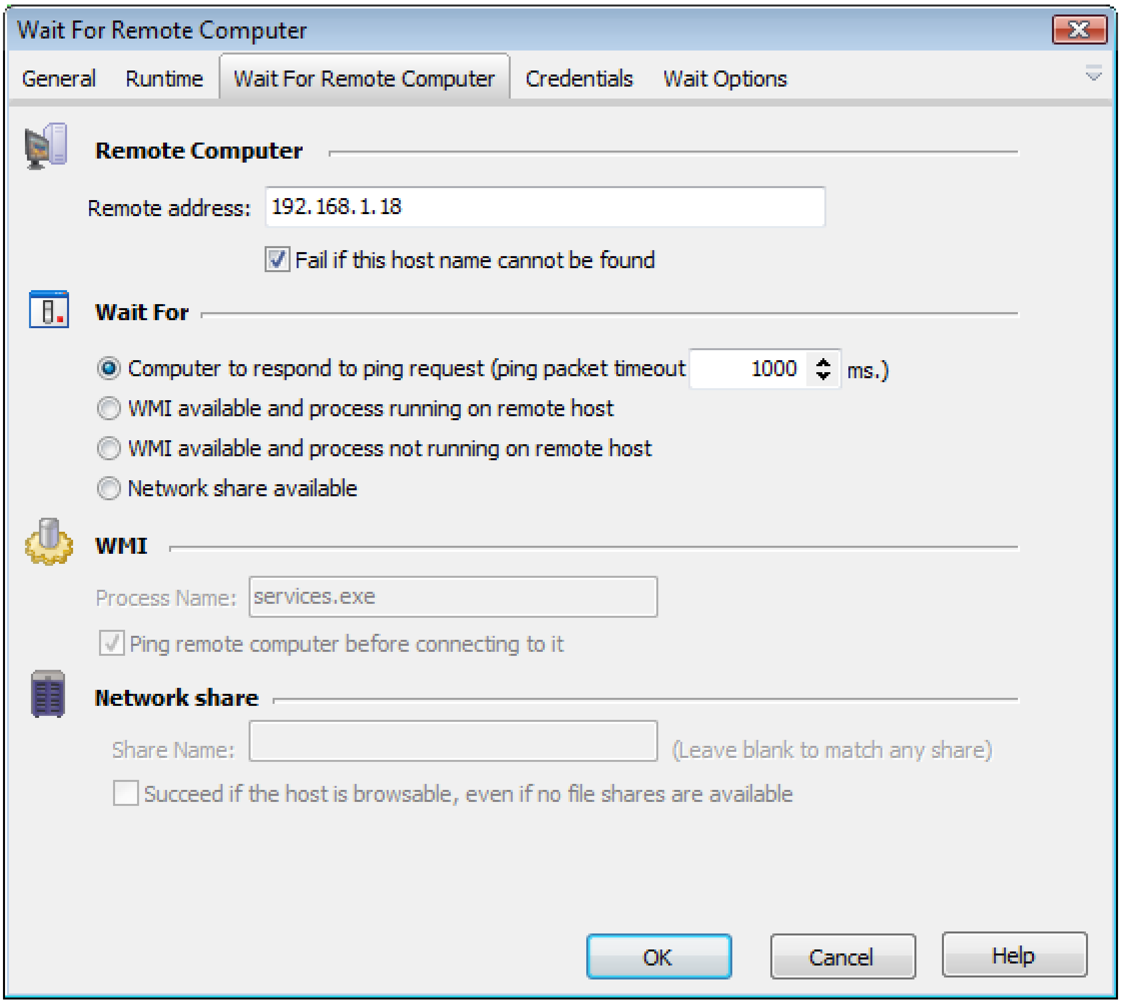
Specify the value to wait for, in the case of the last two Wait For options.

String values are treated literally. DWORD values are treated as their integer values. Binary data blocks are extracted as byte strings. No other data types are supported.

6.10.16.6 Wait For Remote Computer

The Wait for Remote Computer action waits for something to happen to a remote computer.

For an overview of "Wait For..." events, including specifics of the Wait Options, see the Waiting For Events topic.



Wait For Remote Computer

Remote Computer

Remote address: 192.168.1.18

☑ Fail if this host name cannot be found

Wait For

● Computer to respond to ping request (ping packet timeout 1000 ms.)
○ WMI available and process running on remote host
○ WMI available and process not running on remote host
○ Network share available

WMI

Process Name: services.exe

☑ Ping remote computer before connecting to it

Network share

Share Name: (Leave blank to match any share)

☐ Succeed if the host is browsable, even if no file shares are available

Remote Computer

Enter the IP address or hostname or a remote computer.

Check the "*Fail if this host name cannot be found*" box if you want the action to fail if the host name cannot be resolved.

Wait For

Computer to respond to ping request

At each polling interval, Automise will send a single ping packet to the remote computer (or any other remote network device.) If the remote computer responds, the action will continue.

Set the *ping packet timeout* to the maximum length to wait for a single valid reply.

Important: Note that responding to pings may not be a good indication that a computer is fully functional. While a Windows machine responds to pings quite early in the startup process, a given service may not come online until some time later. Using a Control Service action, or the *WMI available and process running* option, may give a more accurate picture of remote computer's status. Alternatively, use a Delay action to pause after the Wait action finishes.

WMI available and process running on remote host

At each polling interval, Automise will attempt to connect to WMI Services on the remote computer and verify if a given process is running.

Optionally, you can set the action to ping the remote computer before connecting via WMI. This can be a good idea, as WMI requests may take a long time (up to a minute) to time out if the remote computer is inaccessible, and during this time the action cannot time out or be aborted.

(Checking if a process like *services.exe* is running may be a more accurate way to determine whether a Windows computer is online.)

WMI available and process not running on remote host

Similar to the above option, the action will wait until the named process is *not* running. Note that if the computer shuts down, this condition will time out without passing (WMI must be available.) Similarly, you can set the action to ping the remote host before attempting a WMI connection.

Network Share Available

The action will succeed when a given shared folder (or any share) is available on the remote computer.

Leave the Share Name blank to match any share.

If matching any share, choose the *Succeed if network browseable, even if no shares available* option if you are only waiting for Windows networking to come online, and do not care whether or not any resources are shared.

Wait Options

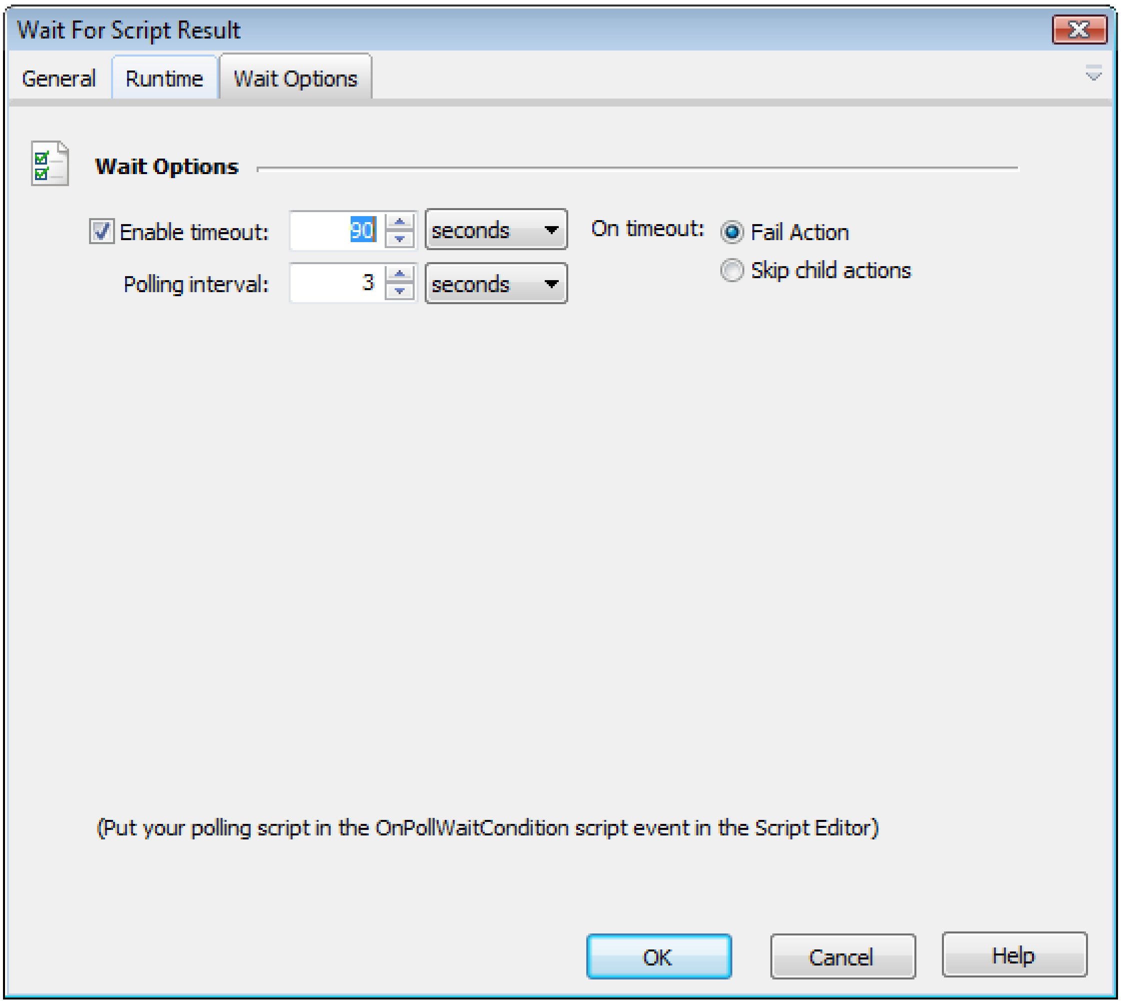
See this topic.

6.10.16.7 Wait For Script Result

The Wait for Script Result action is the most generic "Wait For..." action. After each polling interval, the script event "OnPollWaitCondition" is executed. Set the parameter "KeepWaiting" to false to stop waiting.

Wait Options

See the Waiting For Events topic for more information on Wait For actions.



Wait For Script Result

General | Runtime | Wait Options

Wait Options

☑ Enable timeout: 90 seconds

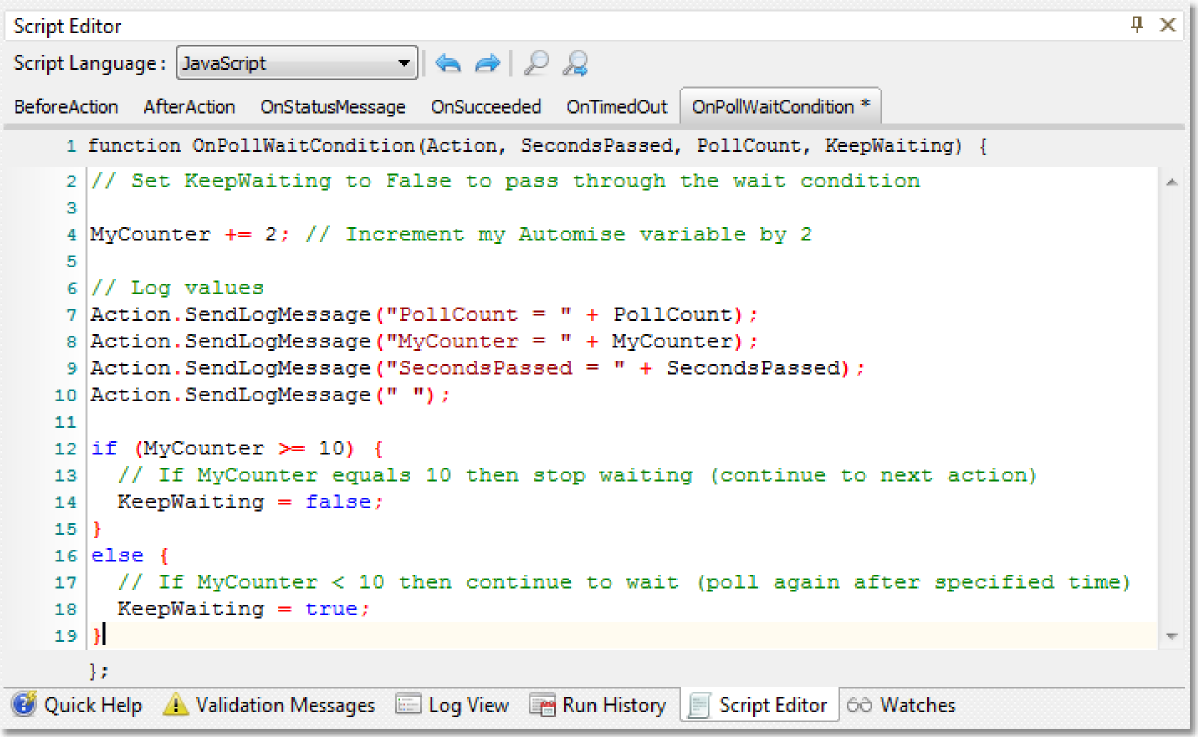
Polling interval: 3 seconds

On timeout: ● Fail Action ○ Skip child actions

(Put your polling script in the OnPollWaitCondition script event in the Script Editor)

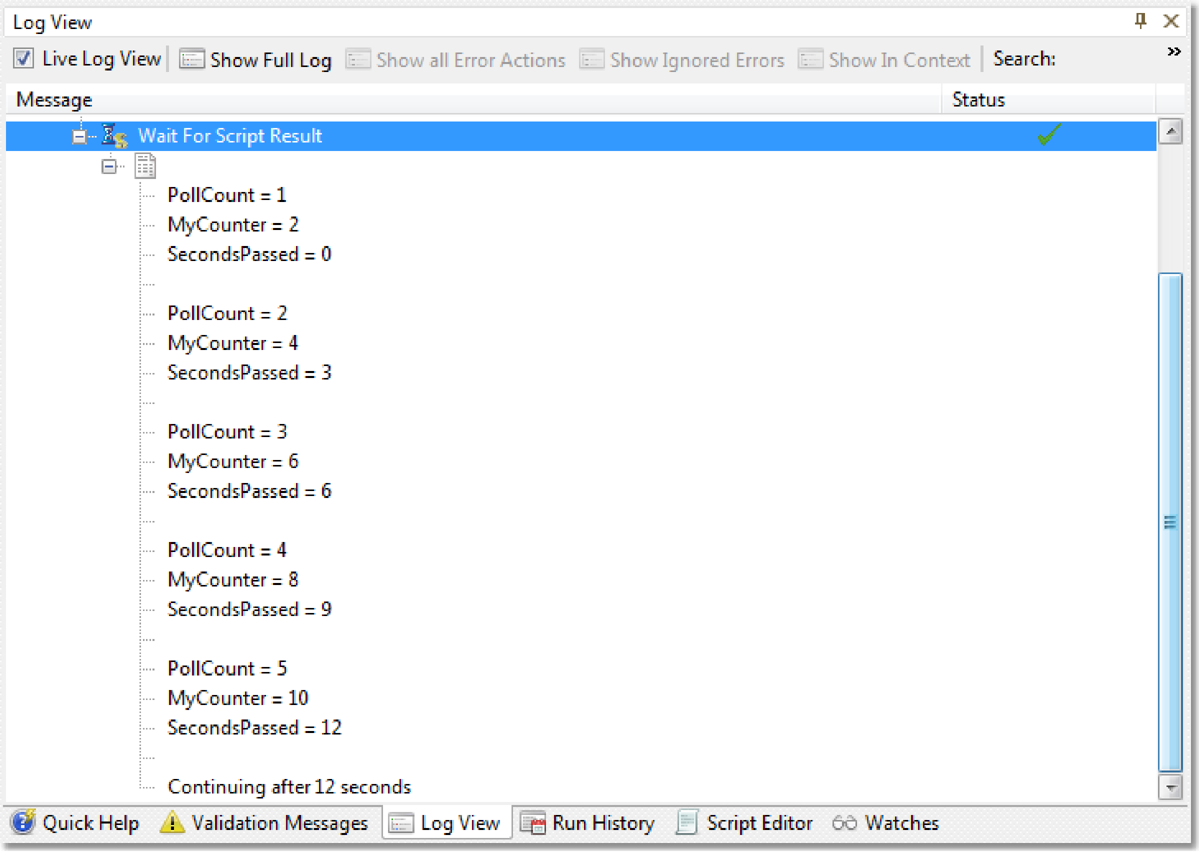
In this example the Automise variable *MyCounter* is incremented by 2 and then evaluated to see if the value is greater or equal to 10. If the value is less than 10 then the action will poll again after the specified period (set via the Wait Options tab of the action).

If the value is greater or equal to 10 then the action will complete successfully and the next action in the action list will be executed.



```
function OnPollWaitCondition(Action, SecondsPassed, PollCount, KeepWaiting) {
// Set KeepWaiting to False to pass through the wait condition

MyCounter += 2; // Increment my Automise variable by 2

// Log values
Action.SendLogMessage("PollCount = " + PollCount);
Action.SendLogMessage("MyCounter = " + MyCounter);
Action.SendLogMessage("SecondsPassed = " + SecondsPassed);
Action.SendLogMessage(" ");

if (MyCounter >= 10) {
  // If MyCounter equals 10 then stop waiting (continue to next action)
  KeepWaiting = false;
}
else {
  // If MyCounter < 10 then continue to wait (poll again after specified time)
  KeepWaiting = true;
}
};
```

The output would look as follows (assuming variable *MyCounter* was set to zero before the Wait For Script Result action commenced):



```
Wait For Script Result                                   ✓
    PollCount = 1
    MyCounter = 2
    SecondsPassed = 0

    PollCount = 2
    MyCounter = 4
    SecondsPassed = 3

    PollCount = 3
    MyCounter = 6
    SecondsPassed = 6

    PollCount = 4
    MyCounter = 8
    SecondsPassed = 9

    PollCount = 5
    MyCounter = 10
    SecondsPassed = 12

    Continuing after 12 seconds
```

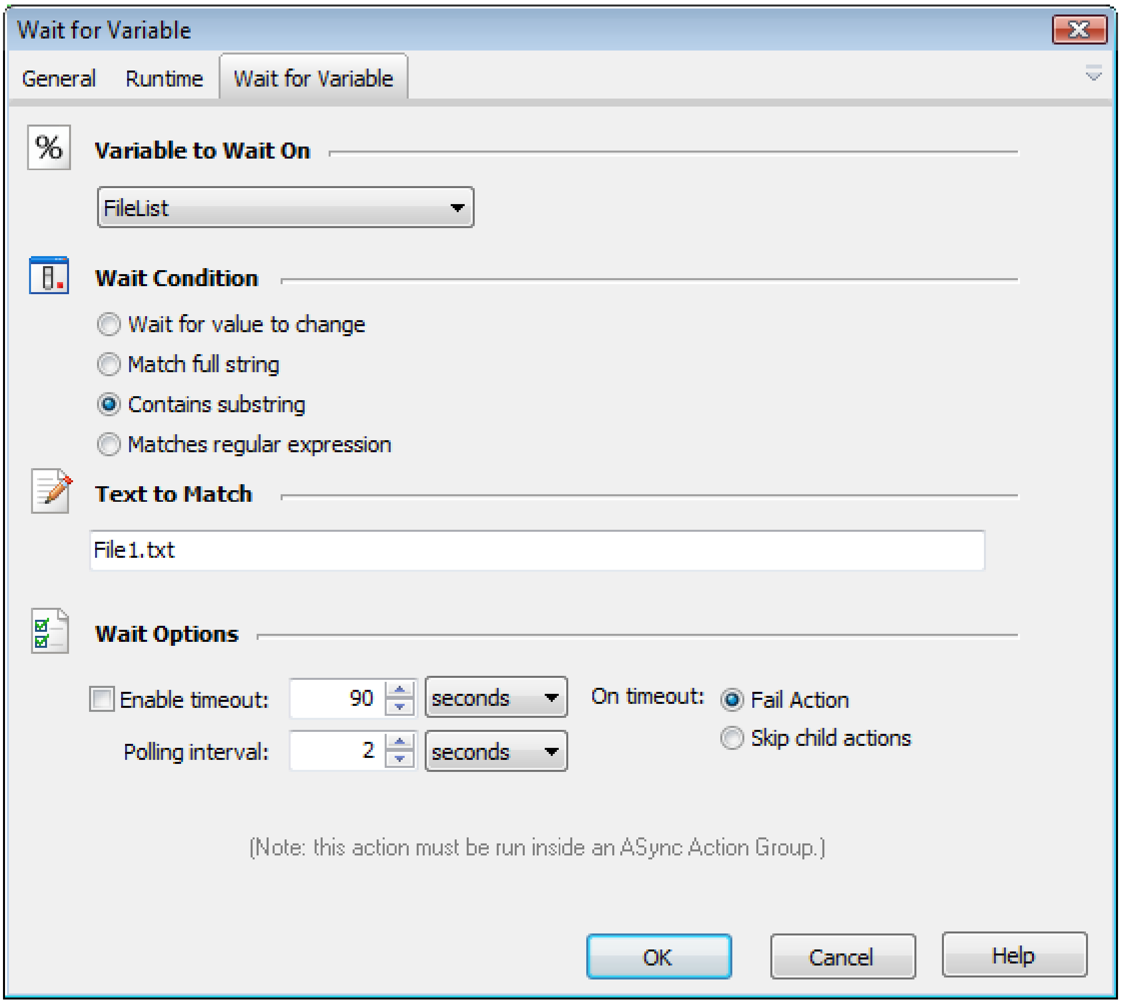
6.10.16.8 Wait For Variable

[Automise Professional Edition]

Wait for Variable pauses the project run until a given Automise variable has changed.

For an overview of "Wait For..." events, including specifics of the Wait Options, see this topic.

Note that this action can **only** be used from inside an ASync Action Group. Otherwise, the variable value will never change and the action can never unblock (the action will fail outright if it is run from outside an ASync Action Group.)



Wait for Variable

Variable to Wait On: FileList

Wait Condition:
- ○ Wait for value to change
- ○ Match full string
- ● Contains substring
- ○ Matches regular expression

Text to Match: File1.txt

Wait Options:
- □ Enable timeout: 90 seconds
- Polling interval: 2 seconds
- On timeout: ● Fail Action ○ Skip child actions

(Note: this action must be run inside an ASync Action Group.)

You can wait for the following conditions:

Wait For Value to Change

The action will unblock as soon as the value of the variable changes, relative to its value when the action first runs.

Match full string

The action will unblock as soon as the variable's value matches exactly the text in the

"Text to Match" edit field.

Contains substring

The action will unblock as soon as the variable's value contains the text in the "Text to Match" edit field.

Matches regular expression

The action will unblock as soon as the variable's value matches the regular expression specified in the "Text to Match" edit field.

Text to Match

Enter the text to match against the condition. If the variable is a number, it is safe to enter numeric digits here. If the variable is boolean, use "True" and "False."

Wait Options

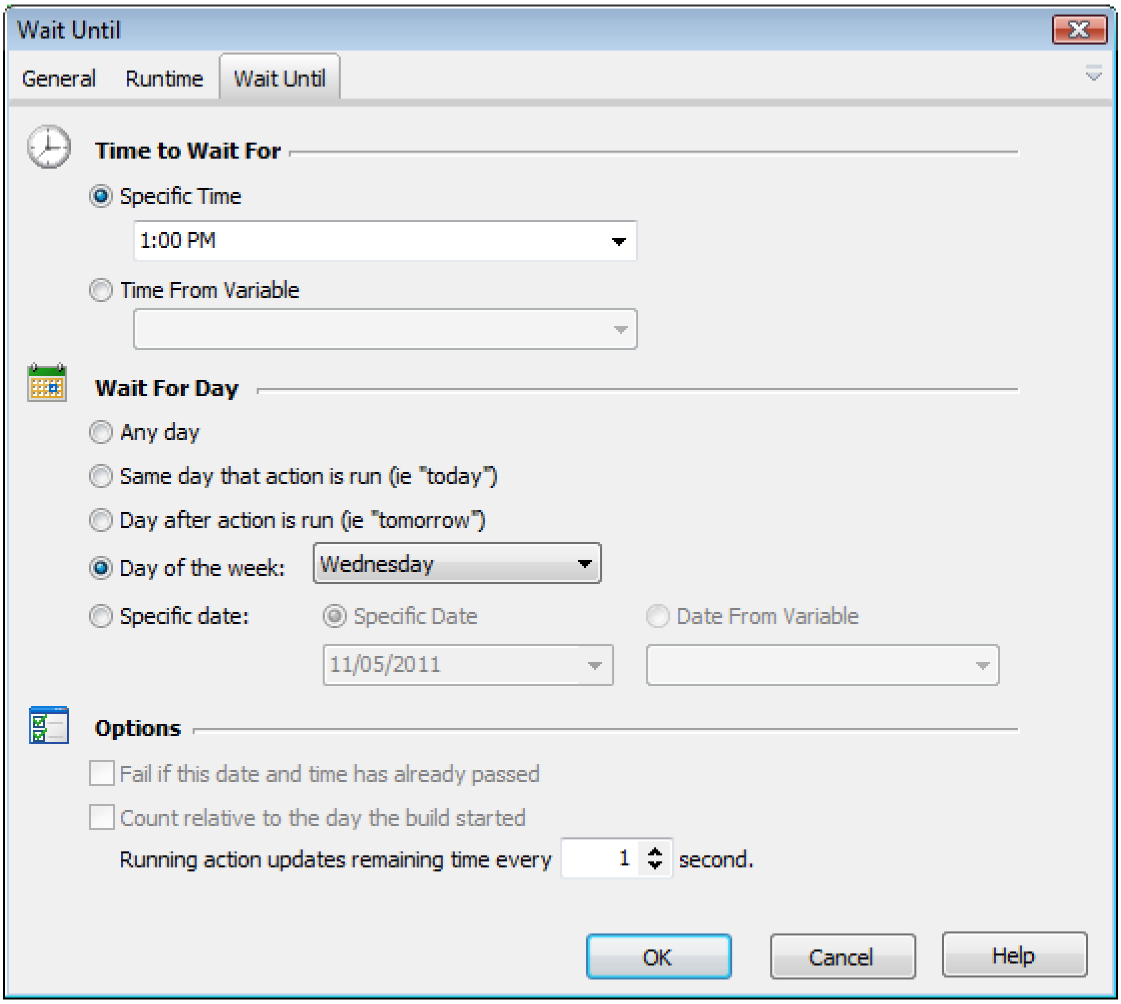
See the Waiting For Events topic.

6.10.16.9 Wait Until

The Wait Until action enables you to wait for a specific amount of time before continuing the project run.

(Unlike other Wait For actions, Wait Until is not a polling wait and cannot time out.)

If you want to wait for a specific length of time (instead of until a specific time), use the Delay action.



Time to Wait For

● Specific Time

1:00 PM

○ Time From Variable

Wait For Day

○ Any day

○ Same day that action is run (ie "today")

○ Day after action is run (ie "tomorrow")

● Day of the week: Wednesday

○ Specific date: ● Specific Date ○ Date From Variable

11/05/2011

Options

□ Fail if this date and time has already passed

□ Count relative to the day the build started

Running action updates remaining time every 1 second.

Time to Wait For

Specify the time you wish to wait for by typing it into the field, or click the drop down button to see an analog clock face that you can (optionally) use to set the time. Use the left and right mouse buttons to set the hour and minute hands, respectively, then click the "Set" button to set the new time.

Wait For Day...

... Any day
The action will go off at the next occurrence of the given time (ie today or tomorrow.) If you set the time to be 5pm and the action starts at 3pm, then it will wait for two hours. If the action starts at 6pm, it will wait 23 hours until 5pm the next day.

... Same day that action is run
The action will run the same day that it is started. If you set the time to be 5pm and the action starts at 3pm, it will wait for two hours. If the action starts at 6pm, it will run immediately.

... Day after action is run
The action will run at the specified time, the day after the action is run.

... Day of the week
The action will wait for the specified day of the week.

... Specific date
The action will wait for a specific date before running.

Fail if this date and time has already passed

If the date/time combination has already passed when the action is run, it will normally continue immediately.

Check this option to have the action fail outright if the date and time have already passed.

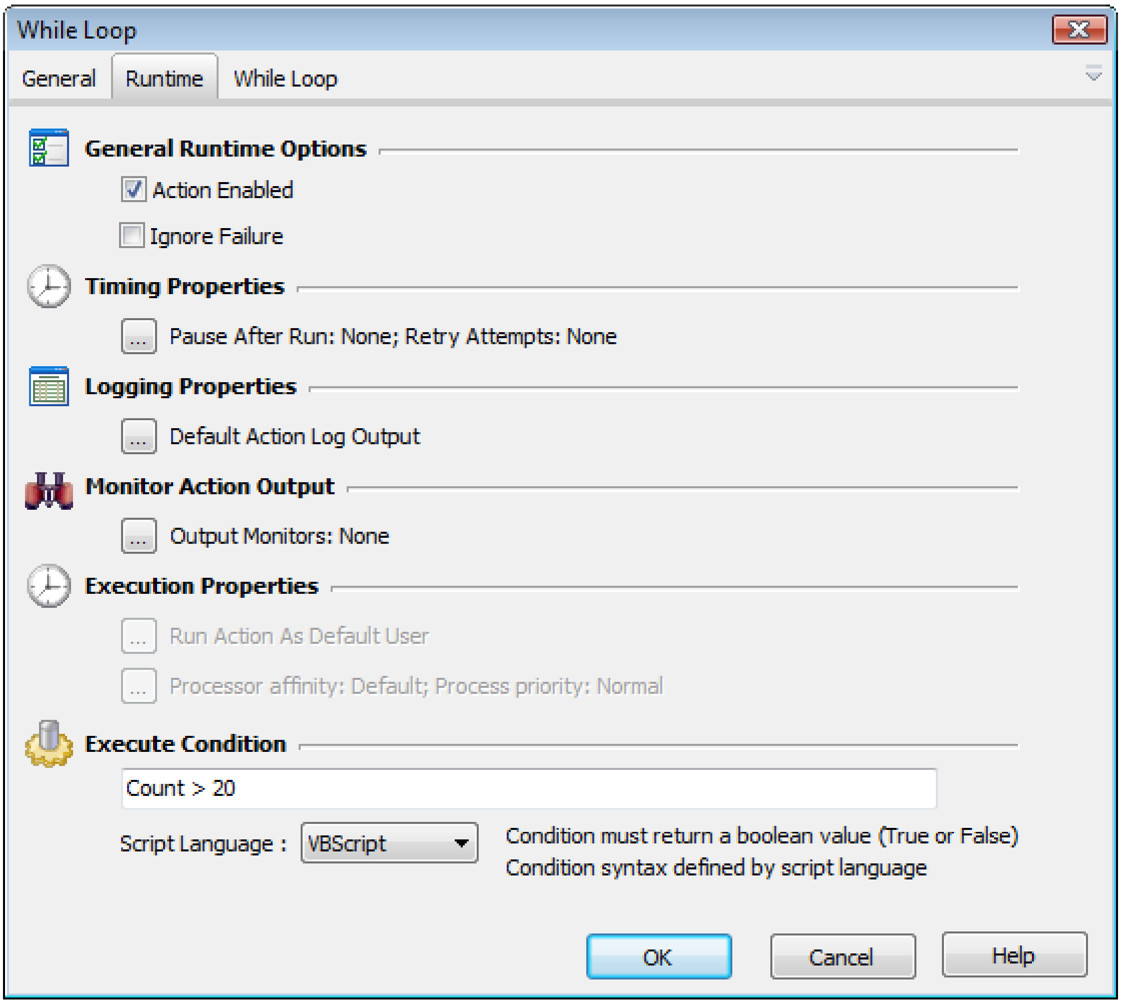
Running action updates remaining time every...

While the action is running, the *Current Running Actions* section of the Run Tab is updated with a progress bar and the amount of time left until execute resumes. Use this option to determine how often that information is updated.

To disable updates entirely, set the update interval to zero seconds.

6.10.17 While Loop Action

The While Loop action will run a set of actions (the child actions of the While Loop) while it's Condition (specified on the Runtime tab) evaluates to True. The While action has no specific functionality other it's looping capability (specifically it tells the stepping engine which action to run next, overriding the default behaviour).



General Runtime Options

☑ Action Enabled

□ Ignore Failure

Timing Properties

... Pause After Run: None; Retry Attempts: None

Logging Properties

... Default Action Log Output

Monitor Action Output

... Output Monitors: None

Execution Properties

... Run Action As Default User

... Processor affinity: Default; Process priority: Normal

Execute Condition

Count > 20

Script Language : VBScript

Condition must return a boolean value (True or False)
Condition syntax defined by script language

NOTE: Note that it will execute its script events (like any other action) first before executing it's child actions. What this means is that the While Loop Action will execute it's AfterAction Event Before any of the Child Actions are executed. The reason for this is that the stepping engine does not have any knowledge of which type of action it is executing and does not treat the while loop action any differently.

While a While Loop is an iterator, it does not share any of the common iterator script events and properties.

6.11 GUI Automation

6.11.1 Overview

The GUI Automation actions allow you to **automate any Windows GUI application** by performing simple operations such as clicking, sending keyboard input, dragging scrollbars, and waiting for dialogs and windows to appear.

See these topics to learn about the actions which can be used for GUI Automation:

- "Wait for Window or Control" Action
- "Perform Mouse Click" Action
- "Perform Mouse Move" Action
- "Send Keyboard Input" Action

- "Set Scrollbar Position" Action

See these topics to learn about the methods used to select controls to automate:

- Control Selection : the Target Property Page (Automatic Configuration)
- Advanced: Manual Configuration

Scripting

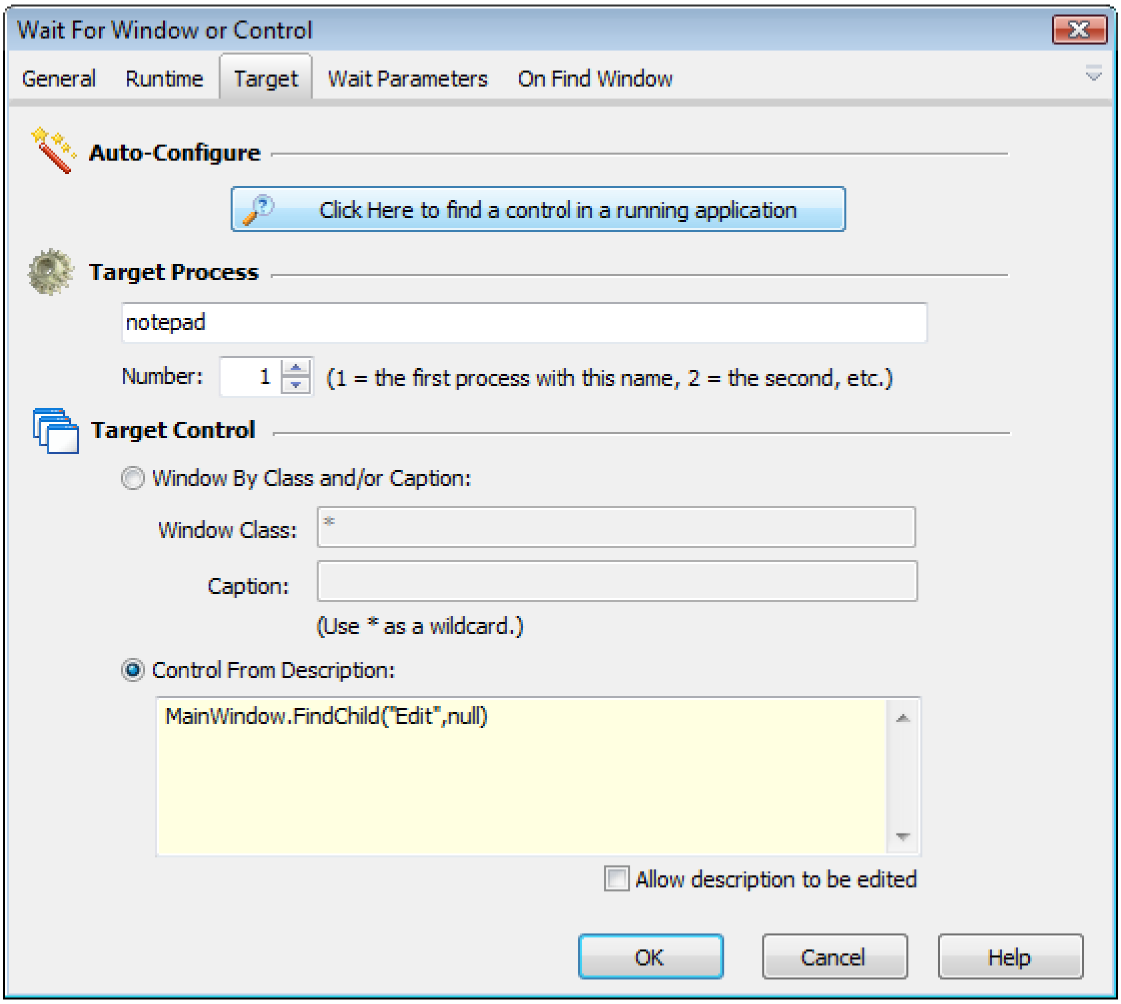
The actions use an object model called the "GUI Scripting Model" to perform these actions. Complete documentation for the Scripting Model will be made available in a future version of Automise.

6.11.2 Target Property Page

6.11.2.1 Automatic Configuration

All of the GUI Automation Actions use the "Target" property page to describe the window or control that the action targets.

Here is an example of an automatically configured Target Property Page:



Wait For Window or Control

Target Process: notepad

Number: 1 (1 = the first process with this name, 2 = the second, etc.)

○ Window By Class and/or Caption:

Window Class: *

Caption:

(Use * as a wildcard.)

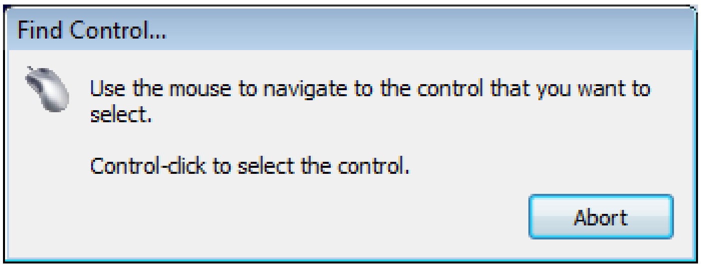
● Control From Description:

MainWindow.FindChild("Edit",null)

□ Allow description to be edited

Auto-Configure

This is the recommended way to configure this property page. When you click on the "Click Here..." button, Automise is hidden and the following dialog pops up:



Find Control...

Use the mouse to navigate to the control that you want to select.

Control-click to select the control.

While you search, Automise will place a red highlight rectangle around the control which is currently under the mouse pointer.

Navigate to a point over the control that you want to choose for the action, and Control-Click to select it. All of the other fields on the property page will be populated with a description of the control. **In most cases, this is all you need to create a reproducible description of the target control.**

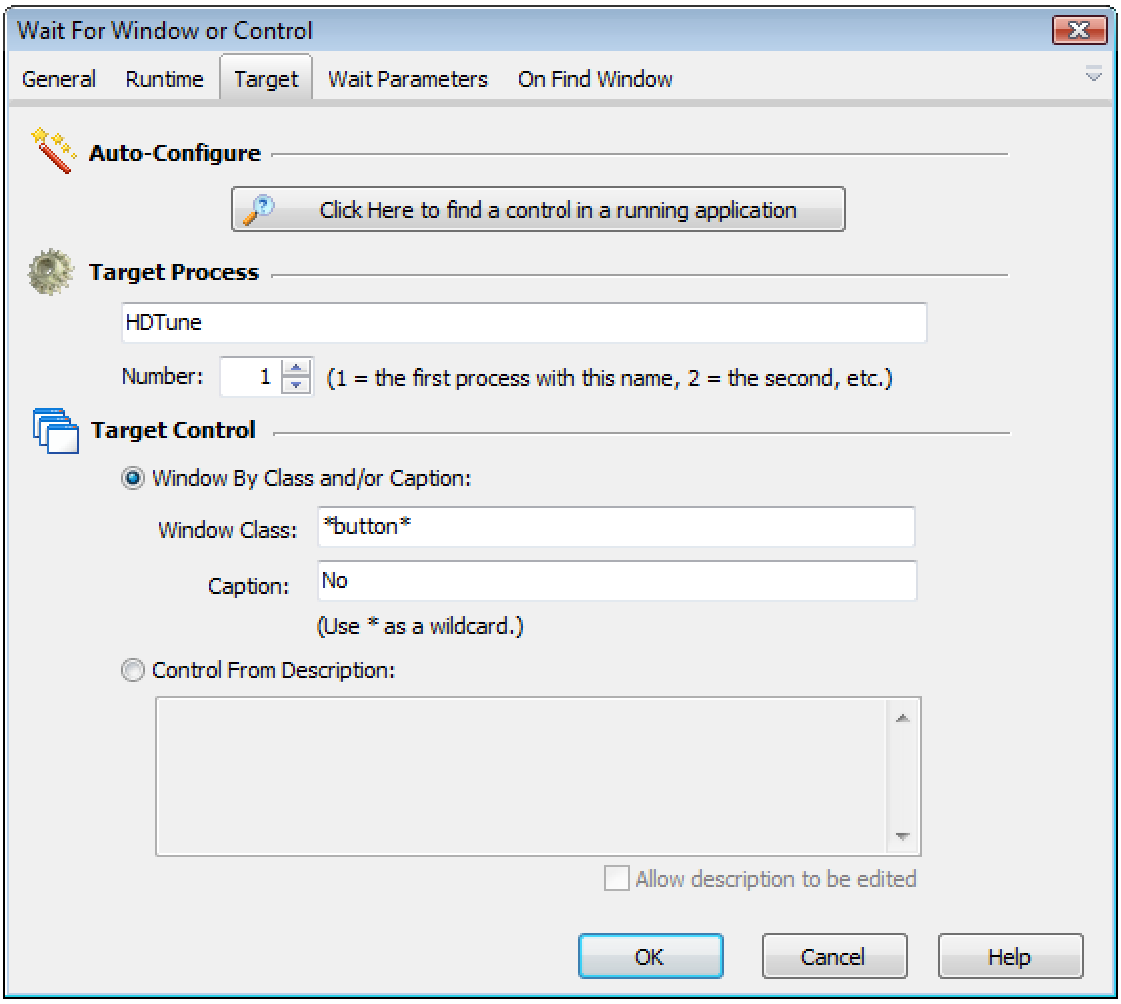
However, you can fine tune the selection by editing each field on the property page. This is described in the Manual Configuration topic.

6.11.2.2 Manual Configuration

All of the GUI Automation Actions use the "Target" property page to describe the window or control that the action targets.

In some circumstances, Automatic Configuration may not be able to select a control for the GUI Automation action. In this case, you can manually edit the fields on the Target property page.

Note: It is recommended that you use Automatic Configuration wherever possible.



HDTune

Number: 1 (1 = the first process with this name, 2 = the second, etc.)

● Window By Class and/or Caption:

Window Class: *button*

Caption: No

(Use * as a wildcard.)

○ Control From Description:

□ Allow description to be edited

Target Process

The name of the target process which owns the control.

Process names are the same as shown in the Windows Task Manager "Processes" tab, but without the ".exe" extension.

Process Number

Process Number is used to identify multiple running processes with the same name. Number 1 is the first running process, 2 is the second, etc. The numbers are ordered by the start times of the processes - processes which started running first will have lower numbers.

Target Control

Controls can be found in one of two ways : By class/caption, or By description.

Window by Class and/or Caption

This allows you to select a window based on two parameters:

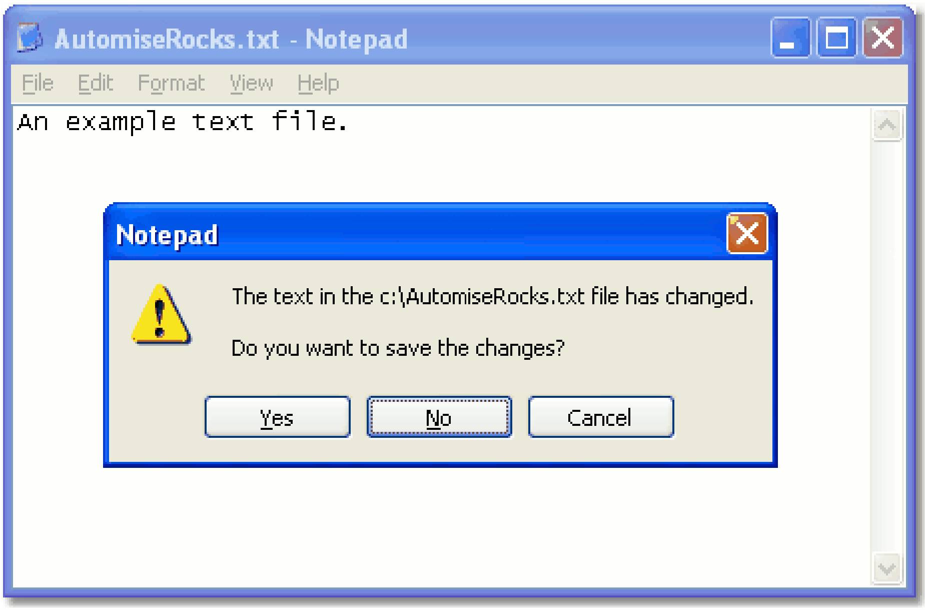
- The Window Caption. This is normally the text displayed in the window.
- The Window Class. This is normally set by the developer of the application.

Tip: In Microsoft Windows, the definition of a "window" includes traditional windows like modal dialogs and document windows, but also includes many user controls like text fields, buttons, and toolbars.

The target window can exist anywhere in the target process. Window classes and captions are not case sensitive. "*" is a wildcard which matches any string.

The following example will return the "No" button in this Notepad modal dialog:

Here is the modal dialog:

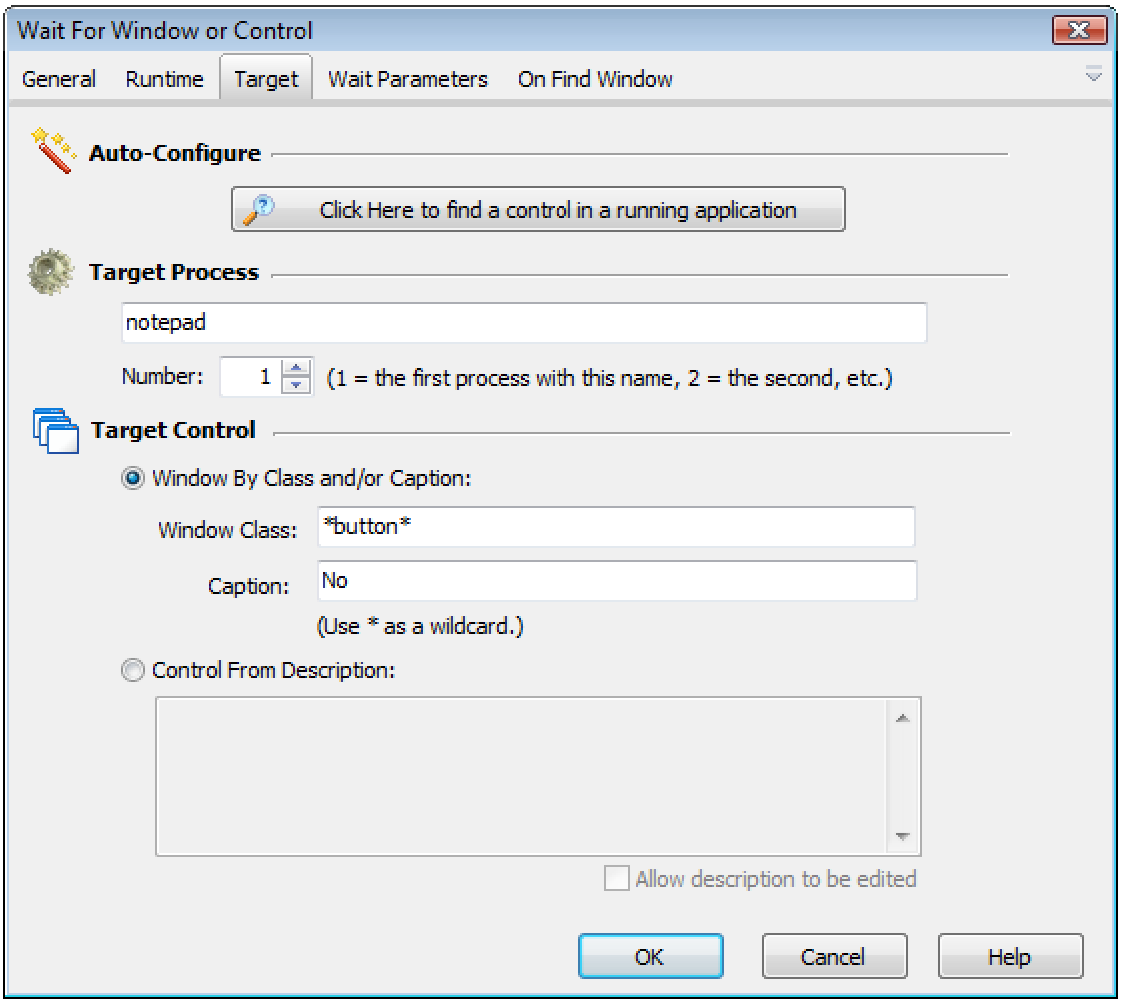


An example text file.

The text in the c:\AutomiseRocks.txt file has changed.

Do you want to save the changes?

Here is the property page:



notepad

Number: 1 (1 = the first process with this name, 2 = the second, etc.)

● Window By Class and/or Caption:

Window Class: *button*

Caption: No

(Use * as a wildcard.)

○ Control From Description:

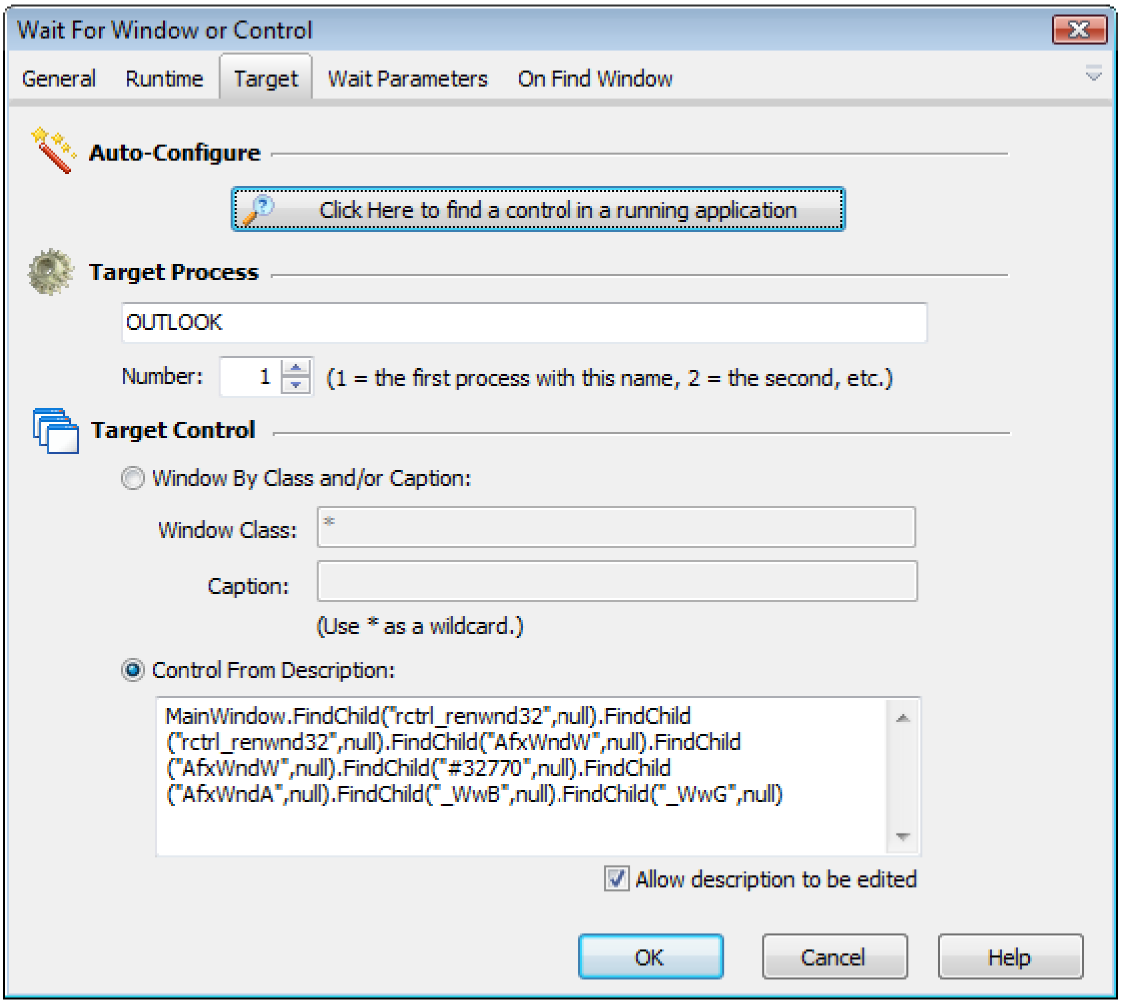
□ Allow description to be edited

Control from Description

The Control From Description option uses a description string to explain to Automise how to find the control. The Description is designed to stay valid even if the window is moved around, or if the window is closed and reopened at some time in the future.

When you use the Automatic Configuration mode, Automise generates a Description and populates the Description field with it.

The example at the top of this help topic shows a simple description - a Modal Dialog of the EFD Software "HDTune" application. The example below shows a much more complicated description - for a button inside the Seapine "Surround SCM Client" application:



Wait For Window or Control

OUTLOOK

Number: 1 (1 = the first process with this name, 2 = the second, etc.)

○ Window By Class and/or Caption:

Window Class: *

Caption:

(Use * as a wildcard.)

● Control From Description:

```
MainWindow.FindChild("rctrl_renwnd32",null).FindChild
("rctrl_renwnd32",null).FindChild("AfxWndW",null).FindChild
("AfxWndW",null).FindChild("#32770",null).FindChild
("AfxWndA",null).FindChild("_WwB",null).FindChild("_WwG",null)
```

☑ Allow description to be edited

Descriptions are specified using the structure of the Automise GUI Scripting Model. The full API of the GUI Scripting Model will be made available in the future.

Until then, it is recommended that you do not edit the Description strings - just use the Automatic Configuration mode to generate them. Nevertheless, if you want to edit or copy/paste descriptions between actions, then you can do so by checking the "Allow description to be edited" checkbox.

6.11.3 Actions

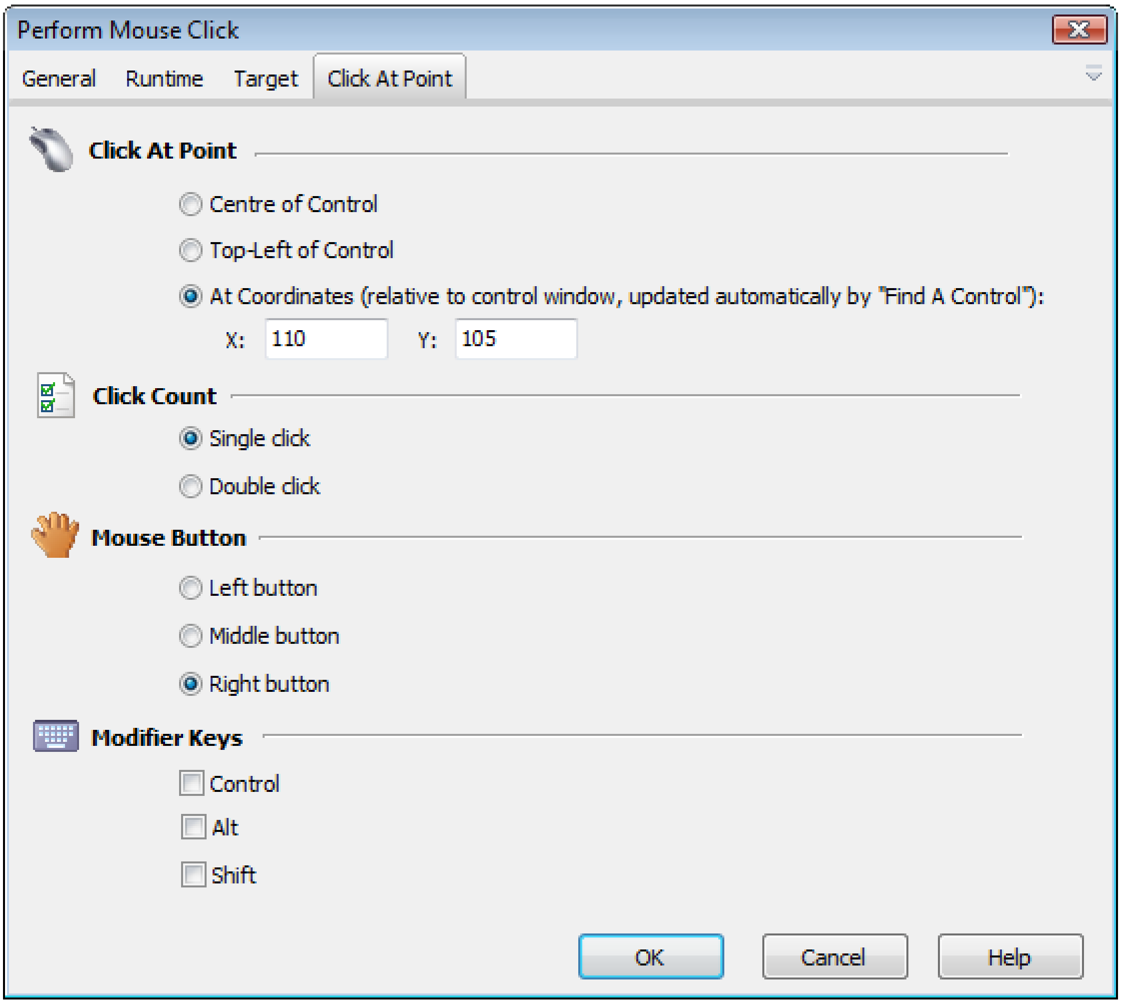
6.11.3.1 Perform Mouse Click

The Perform Mouse Click action clicks on a target control.

Target Property Page

Use the Target Property Page to select the window or control to click on.

Click At Point Property Page



Click At Point

○ Centre of Control

○ Top-Left of Control

● At Coordinates (relative to control window, updated automatically by "Find A Control"):

X: 110 Y: 105

Click Count

● Single click

○ Double click

Mouse Button

○ Left button

○ Middle button

● Right button

Modifier Keys

□ Control

□ Alt

□ Shift

Click At Point

This option will be automatically filled in if you use the Automatic Configuration function of the Target property page.

"**Centre of Control**"

The action will click at the central point of the control. This is best for controls like buttons, where it doesn't matter where the click is.

"**Top-Left of Control**"

The action will click at the top-left corner of the control.

"At Coordinates"

The action will click at the specified point. The coordinates are given as client coordinates of control's window.

"At Coordinates" can be important for some controls because Automise doesn't always know about every control in a window - sometimes, all it can see is a group of controls or an entire window. Normally, the Automatic Configuration function can decide whether or not a click should be accompanied by coordinates.

Click Count

Select whether you would like the action to perform a single or a double-click.

Mouse Button

Select which button the action should click with.

Modifier Keys

Enable the Control, Alt, or Shift keys during the click.

Notes

If necessary, the action will restore minimized windows, and/or move a window to the front of the screen, before clicking on a child control. However, the action will fail if is not on screen after these operations are completed. This means that sometimes you may need to use more than one click to achieve a goal. For example, in order to click a button on a tabbed sheet, you should add two clicks - one to click on the correct tab, and the second to click the button on the sheet.

6.11.3.2 Perform Mouse Move

Use the Perform Mouse Move action to move the mouse to a specific point over a control. This action can be used if you need to hover the mouse over a target control in order for something to happen.

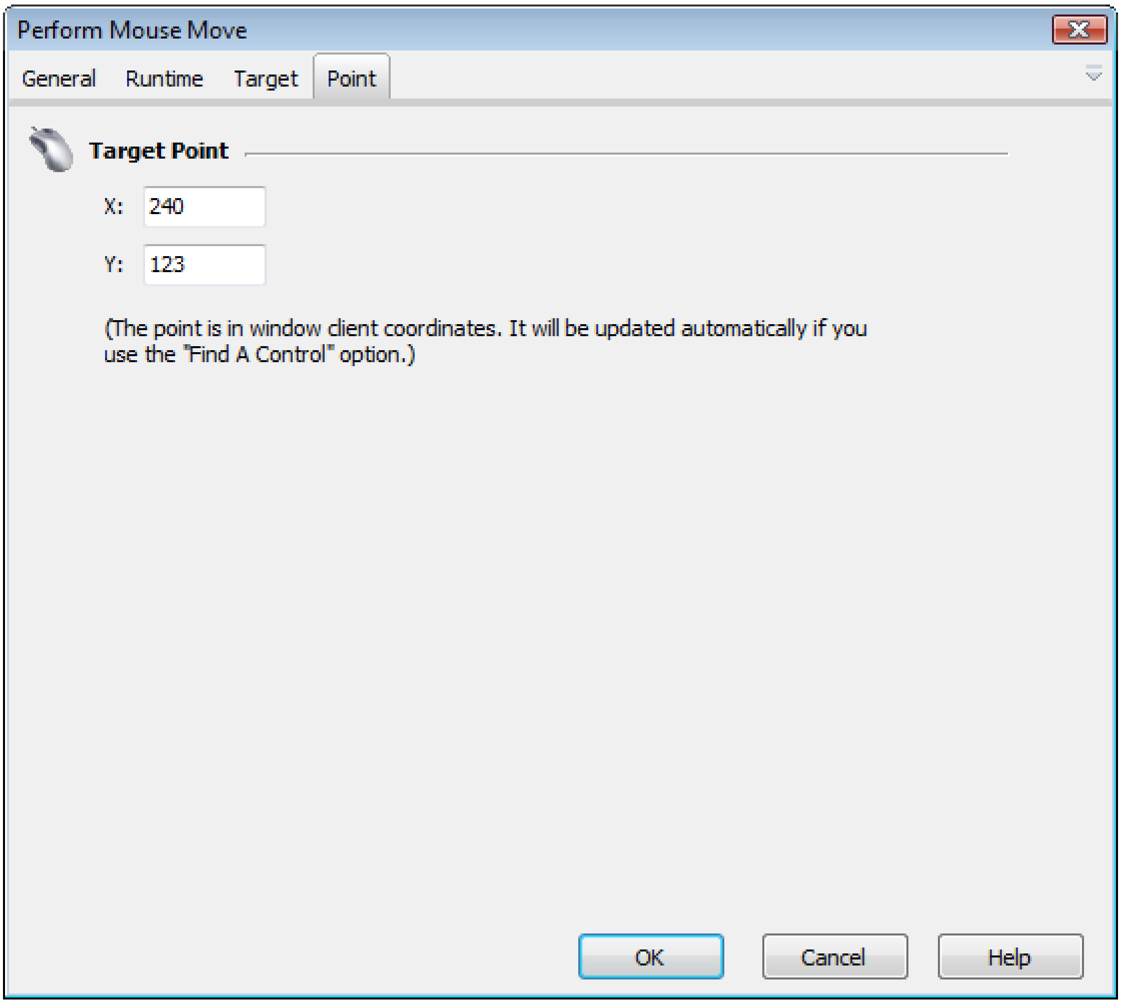
*Note: It is **not** necessary to use Perform Mouse Move each time you use Perform Mouse Click.*

Target Property Page

Use the Target Property Page to select the window or control for the mouse to hover over.

Note that sometimes it may be necessary to work around the automatically detected control, if the control pops up at the target point *after* the mouse has moved there.

Point Property Page



Target Point

X: 240

Y: 123

(The point is in window client coordinates. It will be updated automatically if you use the "Find A Control" option.)

Choose the target point for the mouse to move over. The point is in client coordinates of the control window.

Using the "Automatic Configuration" option on the Target page will automatically fill these fields in with the mouse position.

Notes

To hover the mouse for a specific amount of time, use the "Pause after Run" option on the Runtime property page.

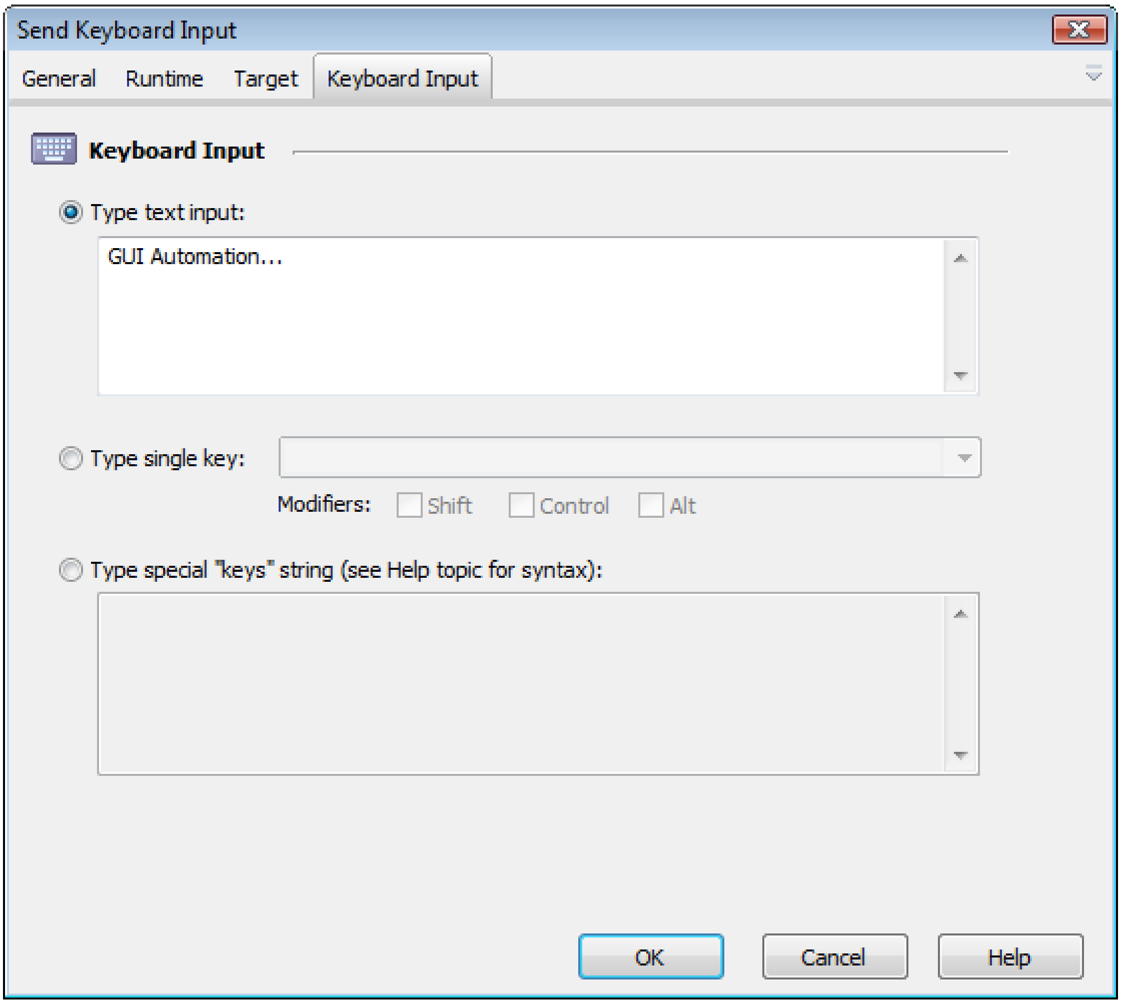
6.11.3.3 Send Keyboard Input

The Send Keyboard Input action "types" text into a window or control.

Target Property Page

Use the Target Property Page to select the window or control to type into. This window/ control will be focused before the keyboard input is sent to it.

Keyboard Input Property Page



● Type text input:

GUI Automation...

○ Type single key:

Modifiers: □ Shift □ Control □ Alt

○ Type special "keys" string (see Help topic for syntax):

Type text input

Select this option to type a string of normal characters (ie no special characters) to the target control. For example, this can be used to enter data into a text field.

You can also enter newline characters here, which will be sent as presses of the Enter key.

Type special key

Select this option to type a single key on the keyboard. This can be used to type a non-printing character such as a backspace, tab, escape, etc. or to perform a keyboard shortcut such as Control-A.

Select modifiers (Shift, Control, Alt) to simulate holding a key down while the special key is pressed.

Type special "keys" string

Select this option to send a string of special keys to the target control. The format of this string is the exactly the same as for the Visual Basic **SendKeys** Sub, or the .NET **System.Windows.Forms.SendKeys** class.

To specify multiple keys, type them in sequence.

For "printable" characters, you can just enter the character (ie for the 'A' key, enter "A".)

The exceptions to this rule are the characters +, ^, %, ~, (,), { and }. To type any of these characters literally, surround them in curly braces, ie {+}, {^}, {%%}, {|}, {(}, {)}, {{} or {}}.

For special keys, you can use the following aliases:

Character	**Alias**
Backspace	{BACKSPACE}, {BS} or {BKSP}
Break	{BREAK}
Caps Lock	{CAPSLOCK}
Delete ("Del")	{DELETE} or {DEL}
End	{END}
Enter ("Return")	{ENTER} or ~
Escape ("Esc")	{ESC}
Help	{HELP}
Home	{HOME}
Insert ("Ins")	{INSERT} or {INS}
Num Lock	{NUMLOCK}
Page Down	{PGDN}
Page Up	{PGUP}
Scroll Lock	{SCROLLOCK}
Tab	{TAB}
F1 through F16	{F1} ... {F16}
Keypad +	{ADD}
Keypad -	{SUBTRACT}
Keypad /	{DIVIDE}
Keypad *	{MULTIPLY}
Arrow Keys (Up, Down, Left, Right)	{UP}, {DOWN}, {LEFT}, {RIGHT}

To apply modifiers (Shift, Alt, and Control) to a key press, precede the key name with one of the following modifier characters:

Modifier Key	**Modifier Character**
Control	^
Shift	+
Alt	%%

(**Note:** "Alt" is actually a single % sign, but it is escaped as %% because a single % sign indicates a variable.)

For example, to specify Control-A, enter "^A".

To apply a modifier key to multiple key presses at once, enclose those keys in parentheses. For example, to specify "Control-A, then Control-Home, then Control-Z", enter "^(A{HOME}Z)".

6.11.3.4 Set Scrollbar Position

The Set Scrollbar Position action allows you to move a scrollbar to a desired position.

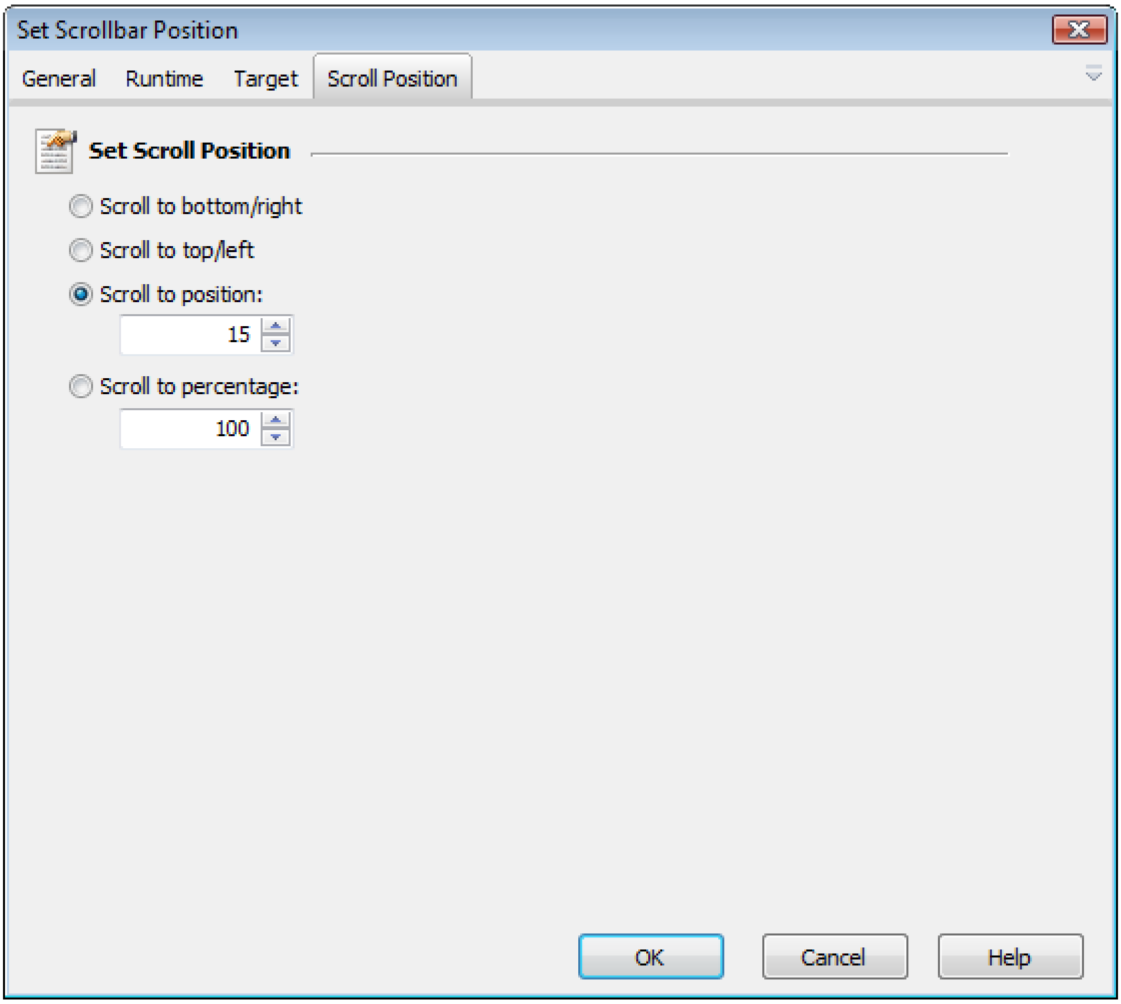
Any scrollbar which implements the standard Windows scrollbar behaviour can be moved (note that some non-standard scrollbars do not support this functionality.)

Target Property Page

Use the Target Property Page to select the scrollbar to move. If you use the automatic configuration mode, then this will ensure that the control you select supports scrolling.

Hint: To select the scrollbar of a window, you will need to control-click on the scrollbar, not the window itself.

Tip: If the scroll bar you want to move doesn't support standard scrolling, you may still be able to automate it using the Perform Mouse Click action. Alternatively, ask the software's developer to add support for standard Windows scroll functionality!



○ Scroll to bottom/right

○ Scroll to top/left

● Scroll to position: 15

○ Scroll to percentage: 100

Scroll to bottom/right

Select this option to scroll a vertical scrollbar to the very bottom, or a horizontal scrollbar to the very right.

Scroll to top/left

Select this option to scroll a vertical scrollbar to the very top, or a horizontal scrollbar to the very left.

Scroll to position

Select this option to set the scroll position to an explicit value. Note that different scroll bars can have different value ranges, so it's not recommended that you use this option unless you know a lot about the scroll bar in question. For some scroll bars, certain values may be inaccessible, so the action will set the bar as close as it can.

Scroll to percentage

Select this option to set the scrollbar to a certain percentage of its available range. 0% is the topmost (or leftmost) position of the bar, and 100% is the bottommost (or rightmost) position.

Notes

This action cannot move a disabled or invisible scroll bar.

6.11.3.5 Wait For Window or Control

The Wait For Window or Control action allows you to pause a running project until a window or control matching certain criteria has appeared on the screen.

Some uses for this tool include:

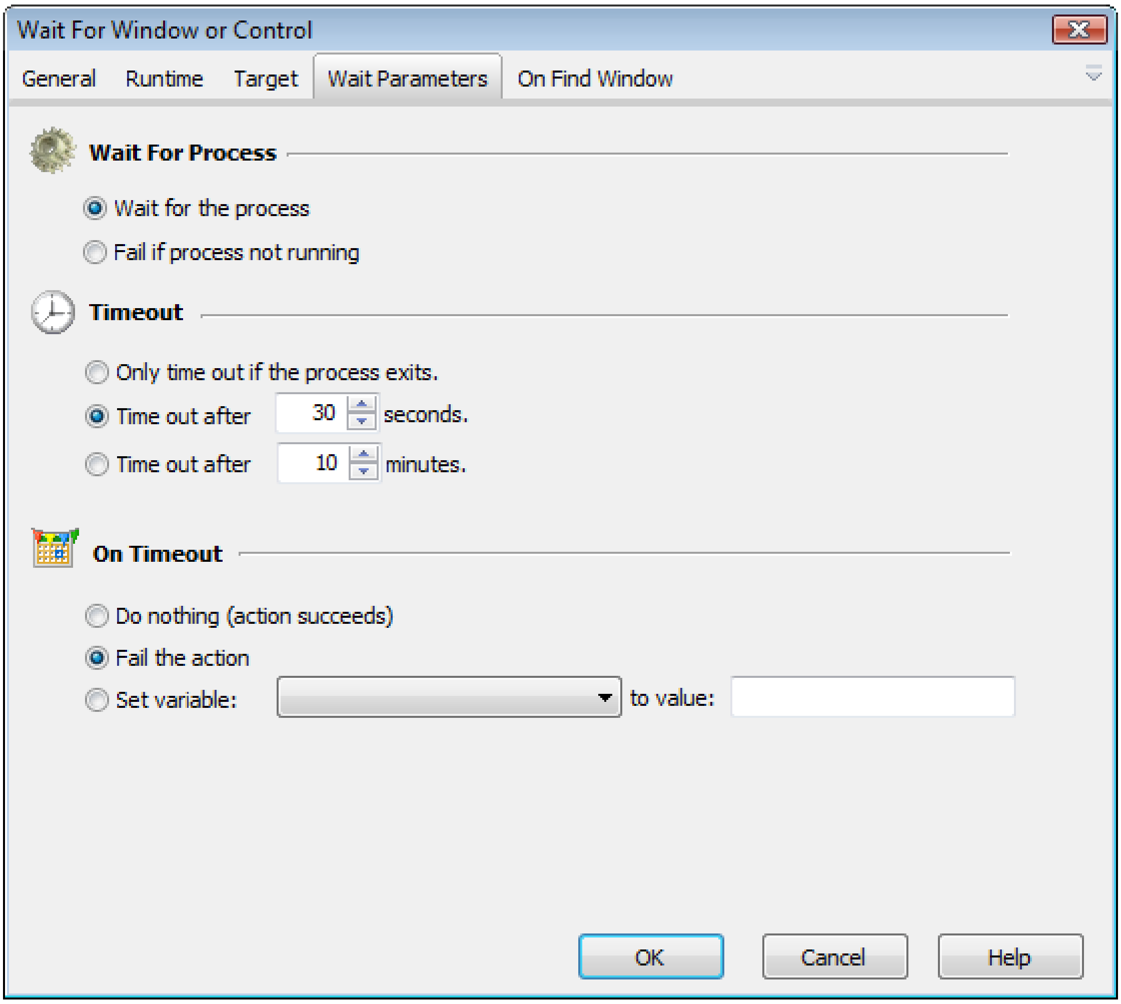
- Monitoring a running application in case it pops up an Error or Warning dialog.
- Waiting for a process to finish starting up.
- Waiting for a process to exit after an operation completes.

Target Property Page

Use the Target Property Page to select the window or control to wait for. This is easiest if you can automatically configure the action by "showing" it a copy of the object you want to wait for.

WARNING: Sometimes, Automise will automatically configure a window to have a simple description such as "ModalDialog" or "MainWindow". These descriptions can be quite ambiguous. It is suggested that you configure the action to wait for a specific button or other control inside the target window, or that you reconfigure the action to match on the window class and/or caption.

Wait Parameters Property Page



Wait For Process

● Wait for the process

○ Fail if process not running

Timeout

○ Only time out if the process exits.

● Time out after 30 seconds.

○ Time out after 10 minutes.

On Timeout

○ Do nothing (action succeeds)

● Fail the action

○ Set variable: [ ] to value: [ ]

Wait For Process

"Wait for the process"

If this option is selected and the target process is not running when the action is started, then the action will wait until the process starts running (and then wait for the window/ control.)

"Fail if process not running"

If this option is selected and the target process is not running when the action is started, then the action will fail straight away.

Timeout

"Only time out if the process exits"

This option disables timeouts for the Wait action, except for when the target process exits.

"Time out after x seconds/minutes"

This option waits for a maximum length of time, and then "times out." You can set the timeout behaviour under "On Timeout":

On Timeout

"**Do Nothing**"

If this option is selected, the action will time out and succeed.

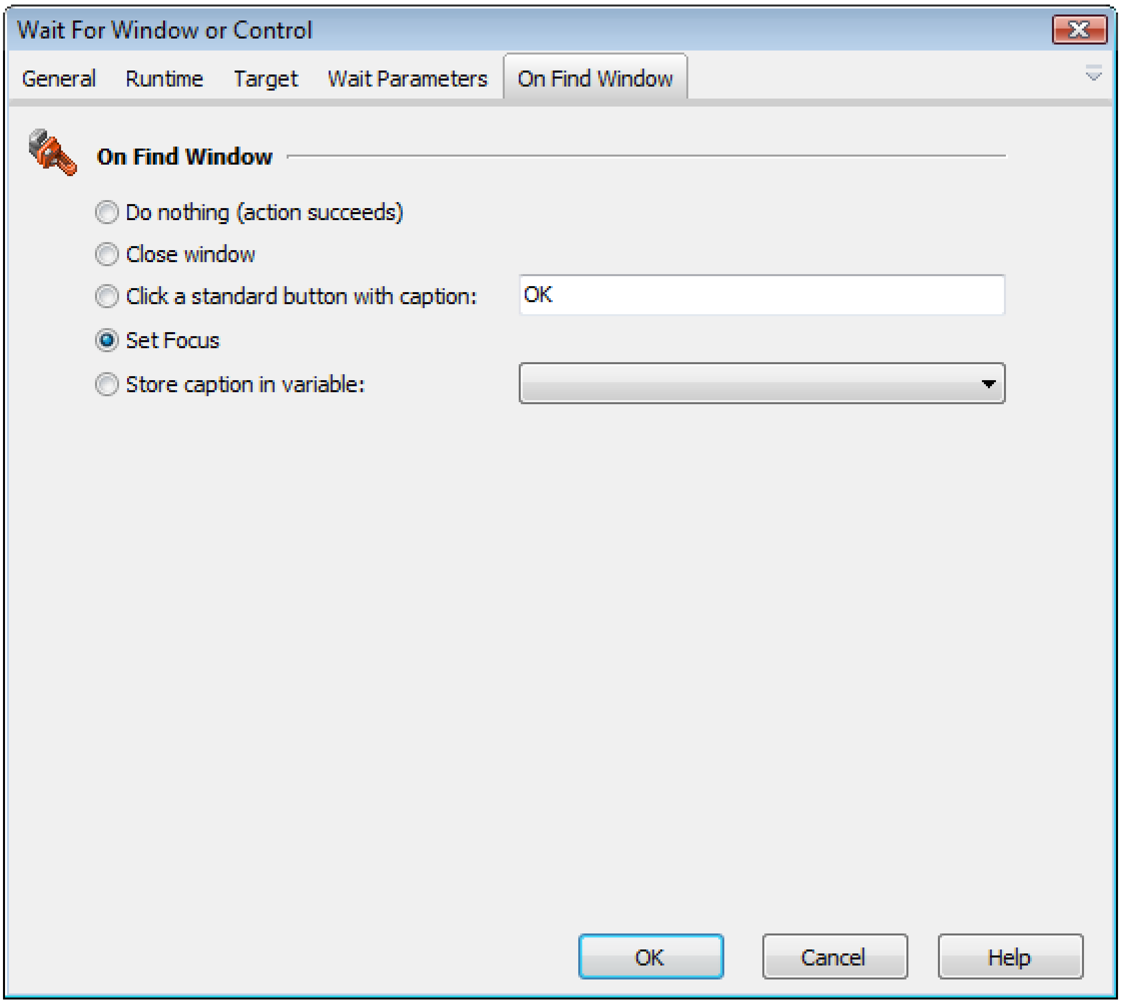
"**Fail the action**"

If this option is selected, then the action will fail if it times out.

"**Set Variable**"

If this option is selected, then the specified variable will be set to the specified value if the action times out. If the action does not time out, the variable will not be changed.

On Find Window Property Page



Wait For Window or Control

On Find Window

○ Do nothing (action succeeds)
○ Close window
○ Click a standard button with caption: OK
● Set Focus
○ Store caption in variable:

This property page allows you to choose what to do if the window or control is found.

"Do nothing"

The action will succeed without doing anything to the window/control.

"Close window"

The action will close the window which contains the target control (or the window itself, if the target is a window.) This is the equivalent of clicking on the "Close" box of the relevant window.

"Click a standard button..."

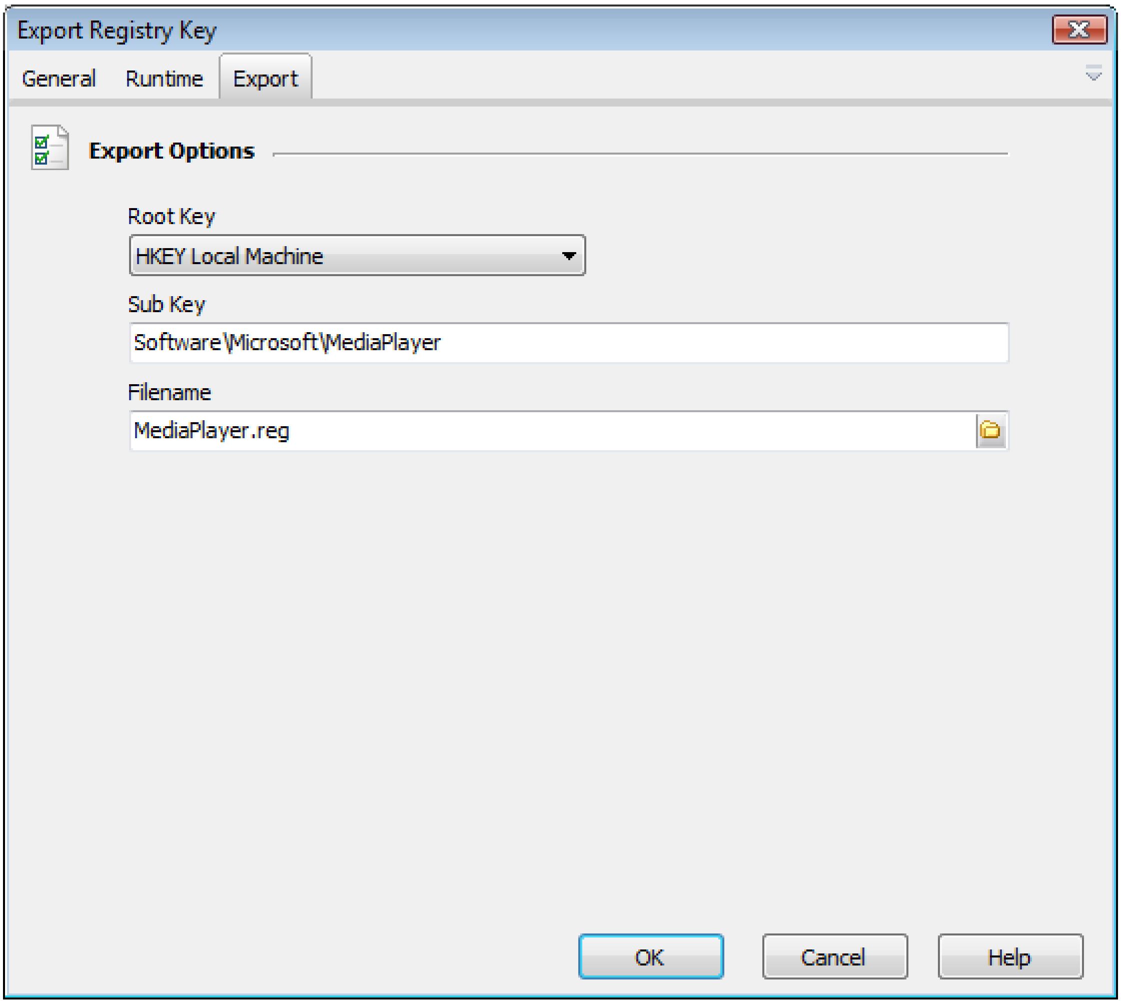
If the target control is a window or a modal dialog which contains "standard" Windows buttons, then this option will click on the button with the specified name. This is useful for modal warning or error dialogs which may need to be automatically dismissed.

Hint: If you need to perform more sophisticated clicking, add a Perform Mouse Click action to the project and place it as a child/sibling of the Wait action.

6.12 Ini Files & Registry

6.12.1 Export Registry Key

The Export Registry Key action enables you to export a registry key as registration (.reg) file.



Root Key
HKEY Local Machine

Sub Key
Software\Microsoft\MediaPlayer

Filename
MediaPlayer.reg

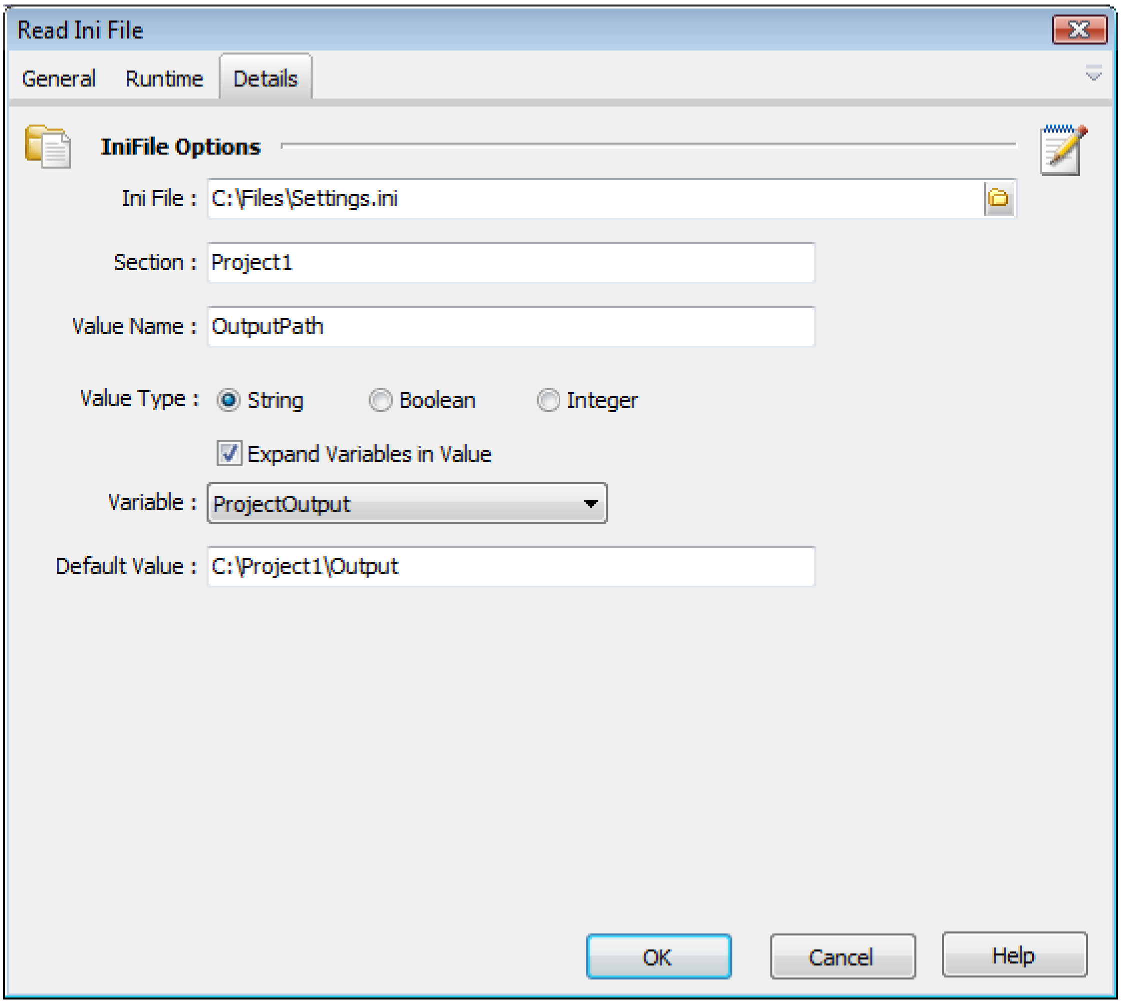
Root Key
The root key of path that is to be exported. Possible keys are 'HKEY Local Machine', 'HKEY Current User', 'HKEY Classes Root', 'HKEY Users' and 'HKEY Current Config'.

Sub Key
The remaining path to the key that is to be exported. For example 'Software\Microsoft\Windows'.

Filename
The registration file that the key will be exported to.

6.12.2 Read Ini File

This action allows you to read values from an ini file into Automise variables.



Ini File : C:\Files\Settings.ini

Section : Project1

Value Name : OutputPath

Value Type : ● String ○ Boolean ○ Integer

☑ Expand Variables in Value

Variable : ProjectOutput

Default Value : C:\Project1\Output

Ini File : The Fully qualified path to the ini file.
Section : The ini File section where the value name is found.
Value Name : The name of the value to read
Variable : The name of the Automise Variable to read the value into.
Default Value : The default value to use if the value name is not found in the ini file.

Scripting Info

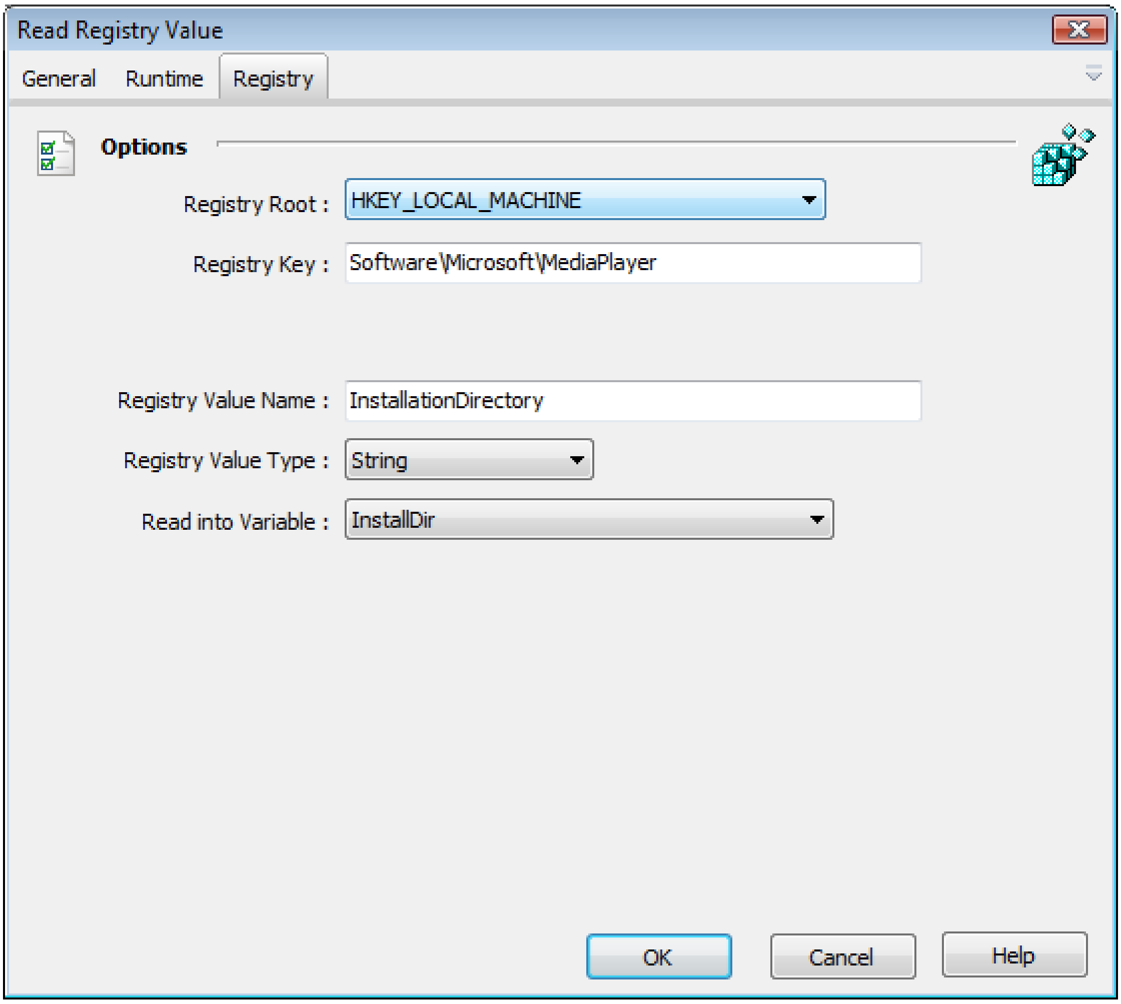
The Action properties available are :

property IniFile : WideString
property Section : WideString
property ValueName : WideString
property VariableName : WideString
property DefaultValue : WideString

See also INI file format information

6.12.3 Read/Set/Delete Registry Value

These actions allow you to Read a Registry Value into a Automise variable, Set a Registry Value or Delete a Registry Key/Value.



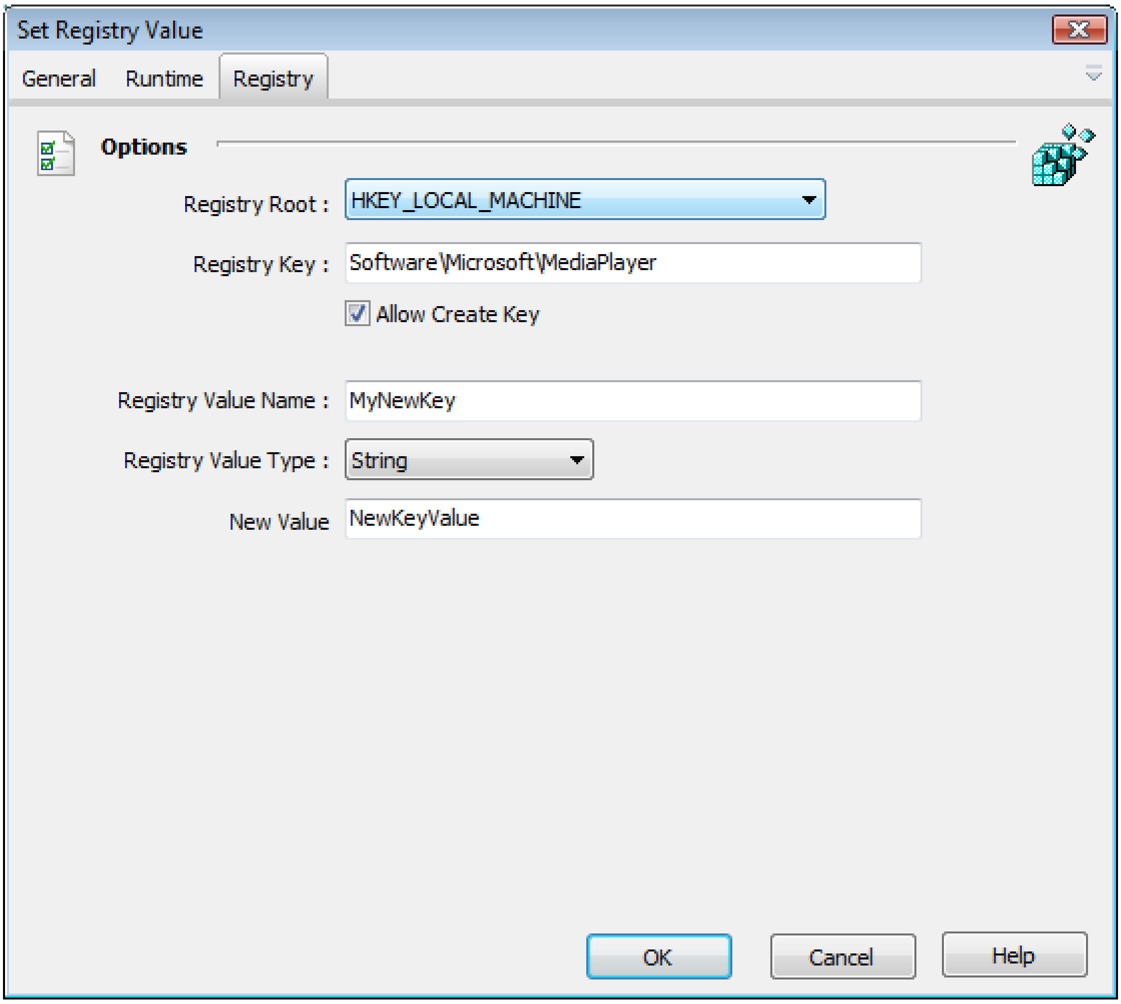
Registry Root : HKEY_LOCAL_MACHINE

Registry Key : Software\Microsoft\MediaPlayer

Registry Value Name : InstallationDirectory

Registry Value Type : String

Read into Variable : InstallDir



Registry Root : HKEY_LOCAL_MACHINE

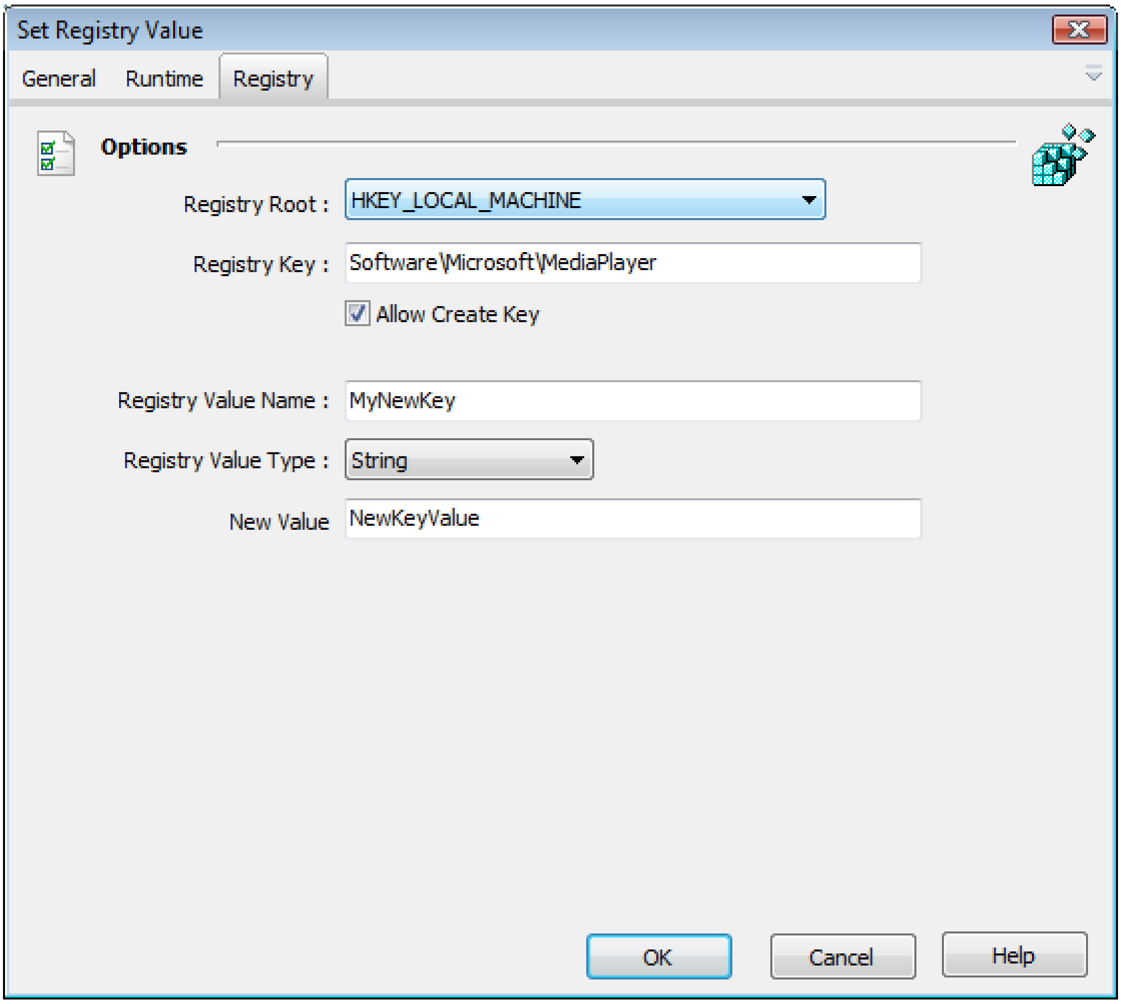
Registry Key : Software\Microsoft\MediaPlayer

☑ Allow Create Key

Registry Value Name : MyNewKey

Registry Value Type : String

New Value NewKeyValue



Registry Root : HKEY_LOCAL_MACHINE

Registry Key : Software\Microsoft\MediaPlayer

☑ Allow Create Key

Registry Value Name : MyNewKey

Registry Value Type : String

New Value NewKeyValue

Enable the "Delete Key" option to delete a key instead of a value. To delete a key, it must contain no subkeys (although it may contain values.)

Scripting Info

The Action properties available for all 3 actions are :

```
property RegRoot      : HKEY;  valid values are :
                          HKEY_CLASSES_ROOT,HKEY_CLASSES_ROOT,
                          HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE,
                          HKEY_USERS,HKEY_CURRENT_CONFIG
property RegKey       : String;
property RegValType   : TRegValueType - valid values are rtInteger and rtString
property RegValueName : String;
```

For Read Value :

```
property FBVariable : string;
```

For Set Value:

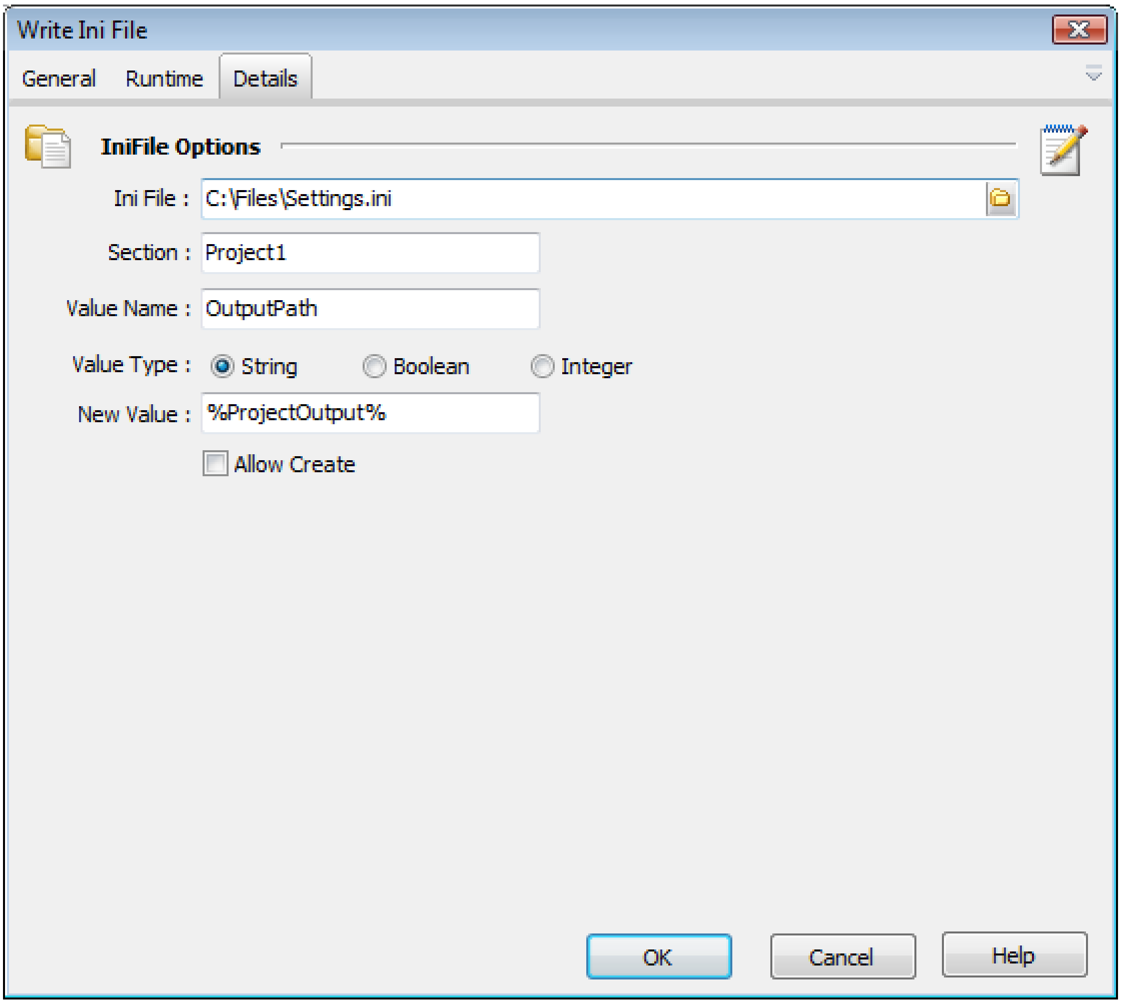
```
property AllowCreateKey;
```

For Delete Value/Key:

```
property DeleteKey;
```

6.12.4 Write Ini File

This action allows you to write values to an ini file.



Ini File : C:\Files\Settings.ini

Section : Project1

Value Name : OutputPath

Value Type : ● String ○ Boolean ○ Integer

New Value : %ProjectOutput%

□ Allow Create

Ini File : The Fully qualified path to the ini file.
Section : The ini File section where the value name is found.
Value Name : The name of the value to write
New Value : The value to write to the ini file. You can use Automise variables here.
Allow Create : Allow the ini file to be created if it does not already exist.

Scripting Info

The Action properties available are :

property IniFile : WideString
property Section : WideString
property ValueName : WideString
property NewValue : WideString
property AllowCreate : WordBool

See also INI file format information

6.13 Installers

6.13.1 Windows Installer

Windows Installer actions enables the manipulation of Windows Installer databases, allowing for customization of pre-built installation packages.

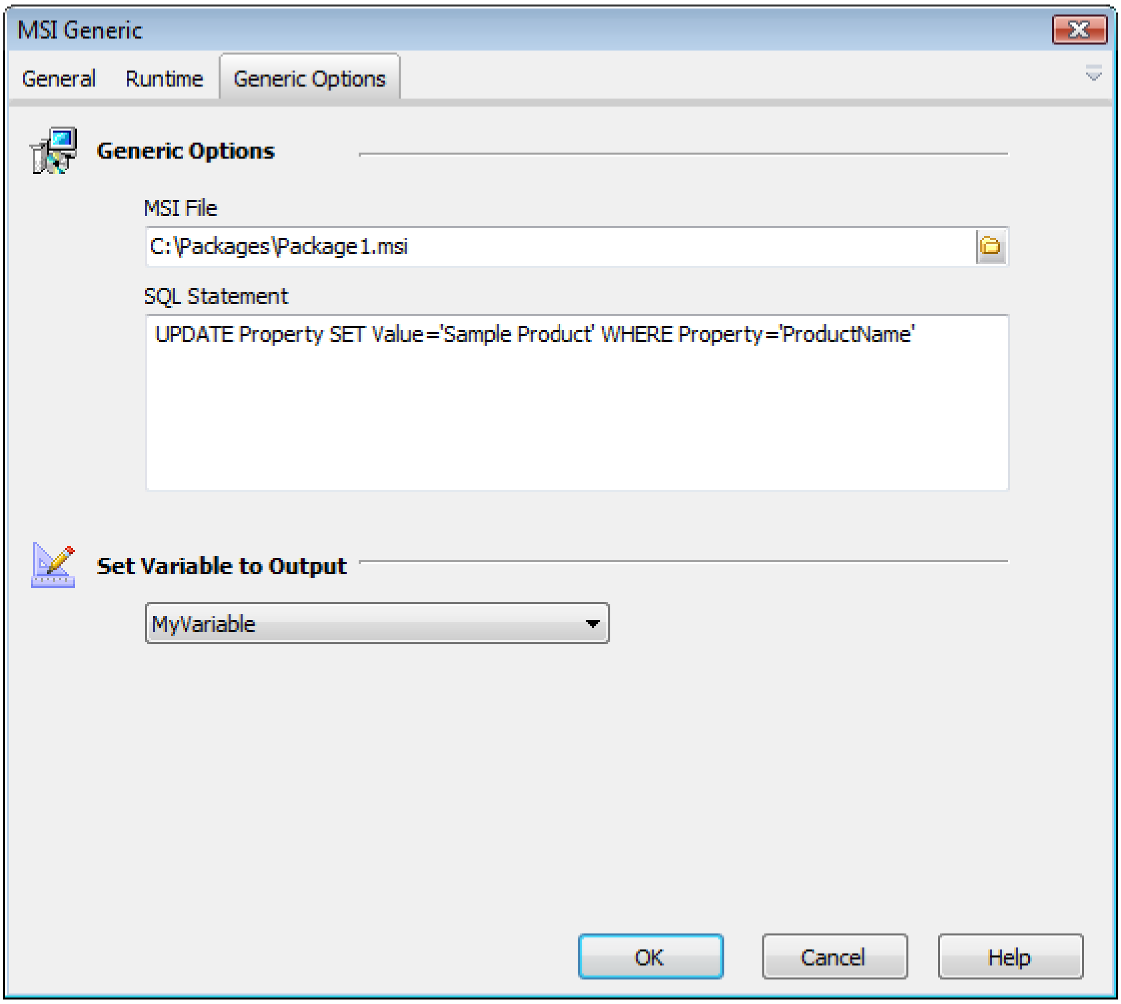
More information regarding Windows Installer can be found at the Microsoft Windows Installer Start Page.

Global Options
The location of MsiVal2 can be specified in the global options dialog: Tools - Options - Install Builders - MSI Sdk.

6.13.1.1 MSI Generic

[Automise Professional Edition]

The MSI Generic action enables you to modify an existing MSI database file using standard SQL statements against the internal tables.



MSI File
C:\Packages\Package1.msi

SQL Statement
UPDATE Property SET Value='Sample Product' WHERE Property='ProductName'

MyVariable

MSI File
The location of the MSI file to manipulate using the SQL Statement.

SQL Statement
The statement to be executed against the MSI database.

Example SQL Statements

Change the Product Name to 'Sample Product':

UPDATE Property SET Value='Sample Product' WHERE Property='ProductName'

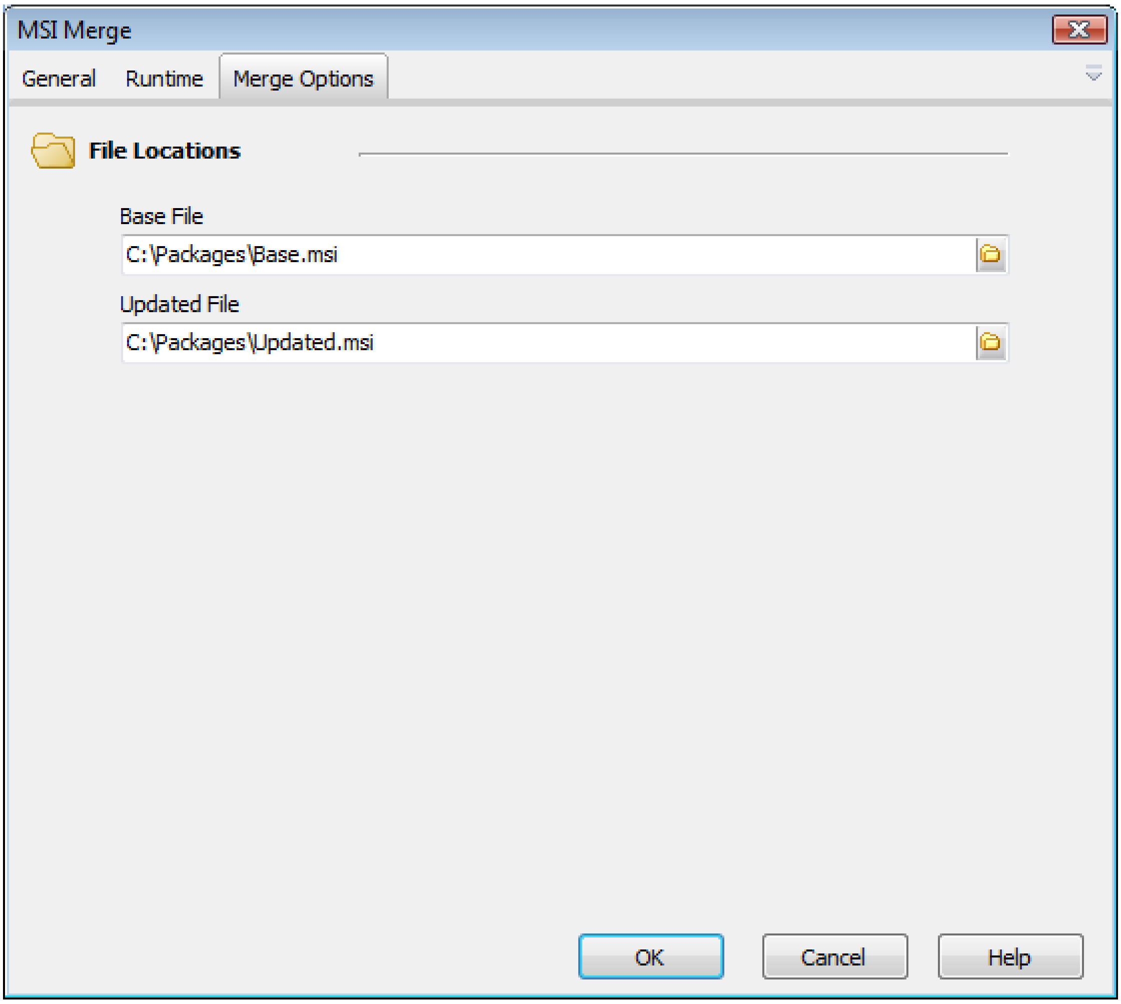
Change the Dialog Font to 'Courier':

UPDATE TextStyle SET FaceName='Courier', Size='10' WHERE TextStyle='DlgFont8'

6.13.1.2 MSI Merge

[Automise Professional Edition]

The MSI Merge action enables you to merge two MSI databases into one.



Base File: C:\Packages\Base.msi

Updated File: C:\Packages\Updated.msi

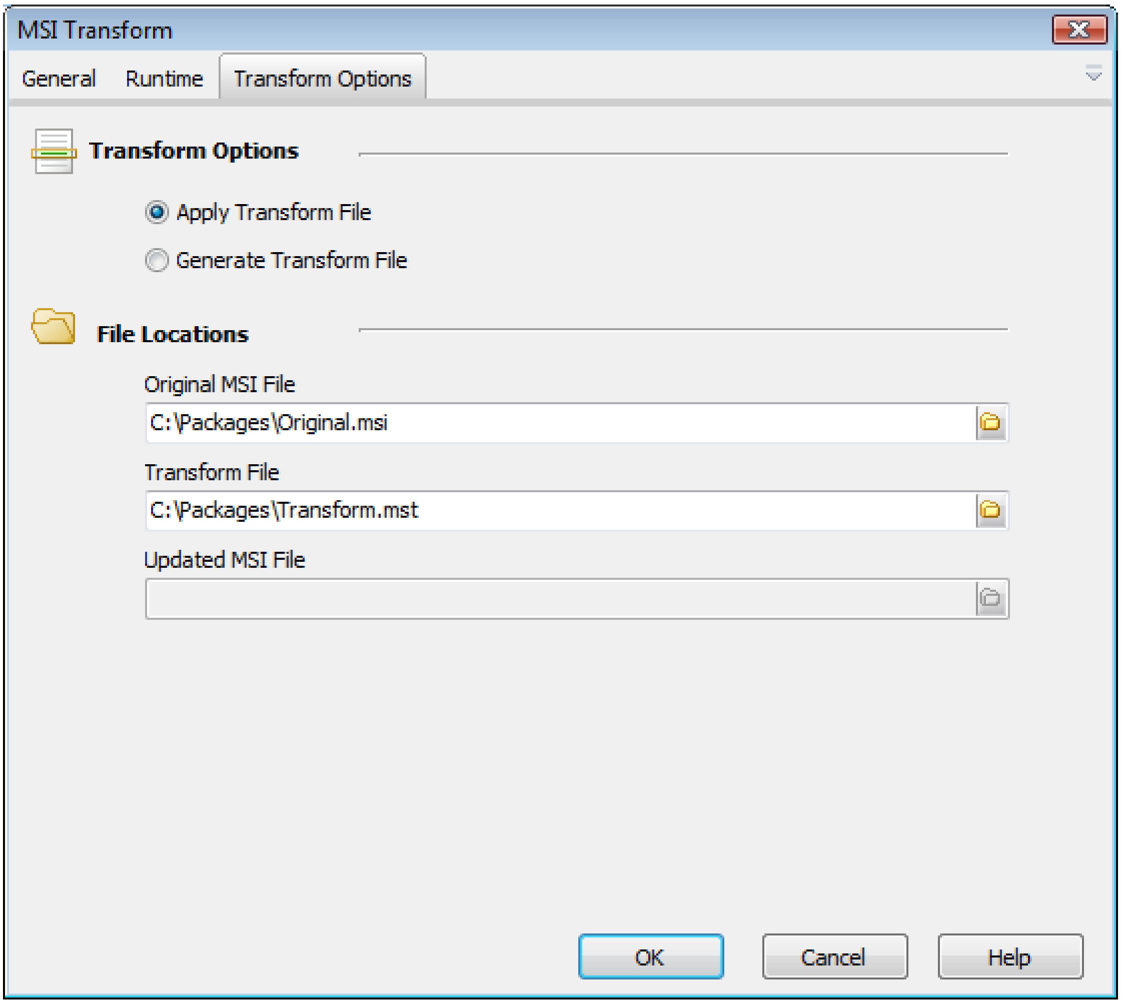
Base File
The location of the MSI file that will be merged.

Updated File
The location of the MSI file that contains the changes that should be merged into the Base file.

6.13.1.3 MSI Transform

[Automise Professional Edition]

The MSI Transform action enables you to generate a MSI transform files, or apply transforms to existing MSI databases.



● Apply Transform File
○ Generate Transform File

Original MSI File: C:\Packages\Original.msi

Transform File: C:\Packages\Transform.mst

Updated MSI File:

Original MSI File
An MSI file that has not contain the changes of the transform file or the updated MSI file.

Transform File
The location of the transform file, which will be created when generating a transform.

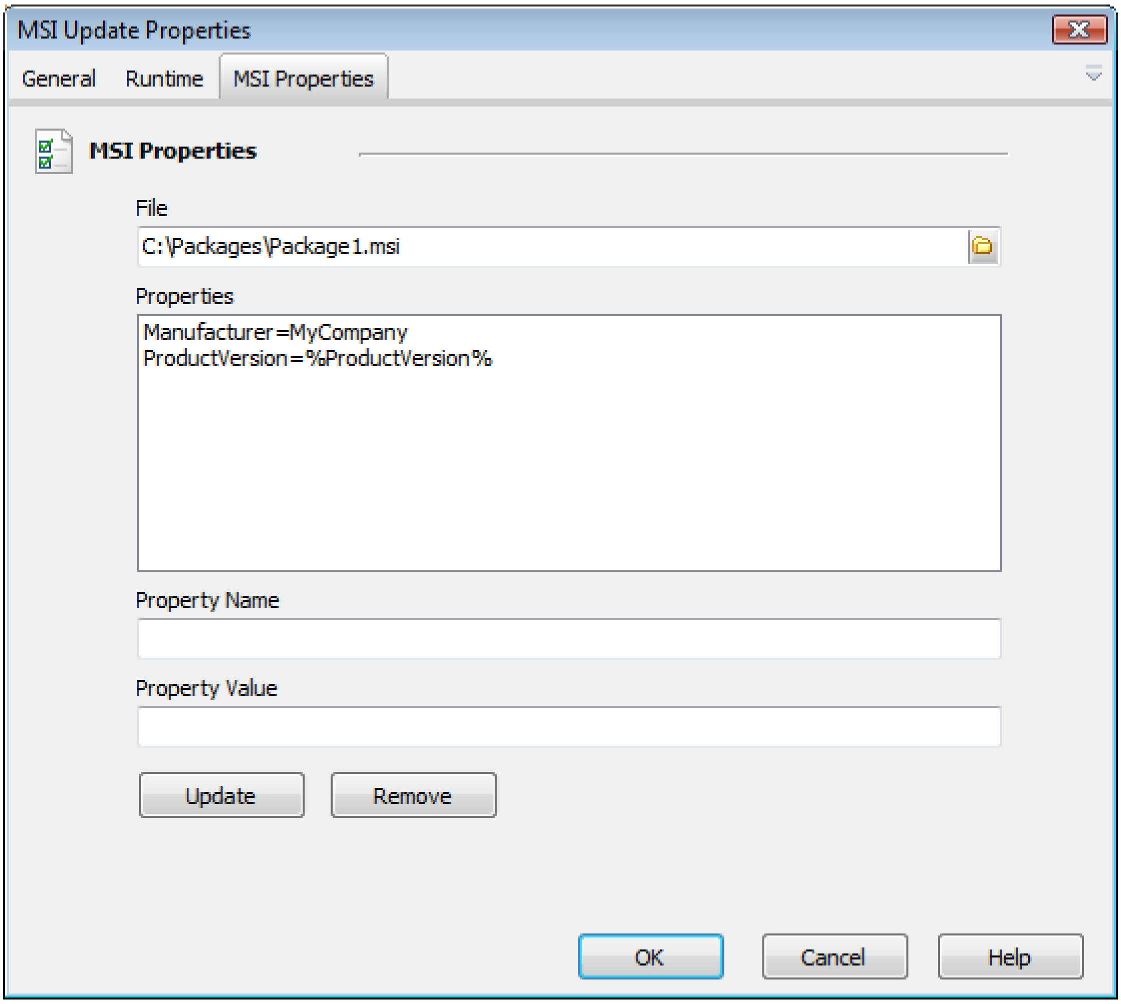
Updated MSI File
The MSI file which contains the changes to the original file, which will be used when generating a transform file.

6.13.1.4 MSI Update Properties

[Automise Professional Edition]

The MSI Update Properties action enables you to alter existing properties within an MSI Database.

Note: This action will not add or remove properties from the MSI database, it will only update existing properties.



File: C:\Packages\Package1.msi

Properties:

```
Manufacturer=MyCompany
ProductVersion=%ProductVersion%
```

Property Name:

Property Value:

File
The location of the MSI file to be updated.

Properties
The list of properties contained in the MSI database. Once the MSI file has been specified, a list of all the properties will be available from here.

Property Name
The name of the property to update. If the name cannot be found in the Properties list when Update is clicked, it will be added.

Property Value
The value of the selected property.

Update/Remove
The update button will save the changes made to the Property to the Properties list. The remove button will remove the selected Property from the Properties list.

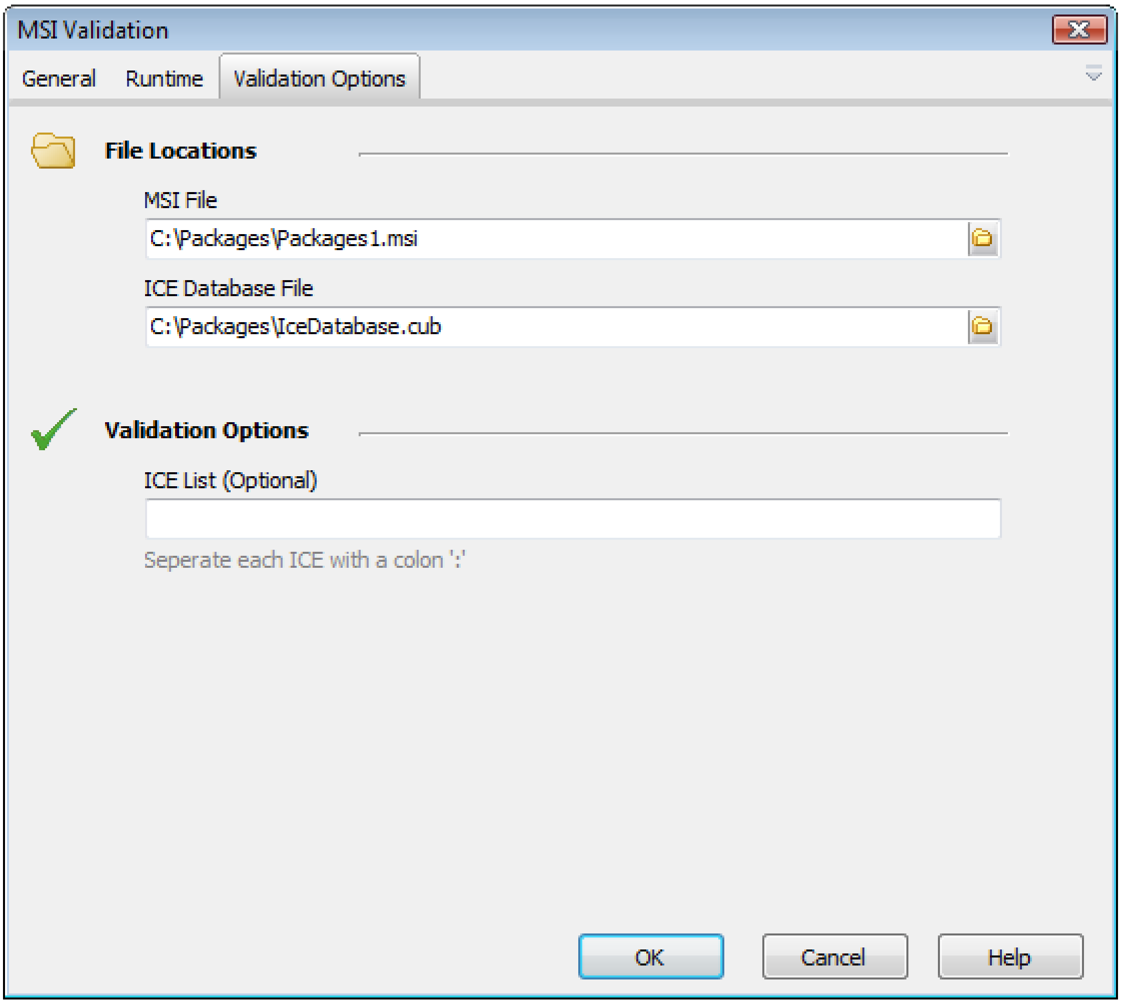
Neither of these buttons will actually remove items from or insert items into the database, they are simply there to allow alteration of the properties list.

6.13.1.5 MSI Validation

[Automise Professional Edition]

The MSI Validation action enables you to perform Internal Consistency Evaluation on an existing MSI database.

This action uses MsiVal2 executable, which is part of the Platform SDK Components for Windows Installer Developers.



MSI File: C:\Packages\Packages1.msi
ICE Database File: C:\Packages\IceDatabase.cub

ICE List (Optional):
Seperate each ICE with a colon ':'

MSI File
The MSI file to perform the validation on.

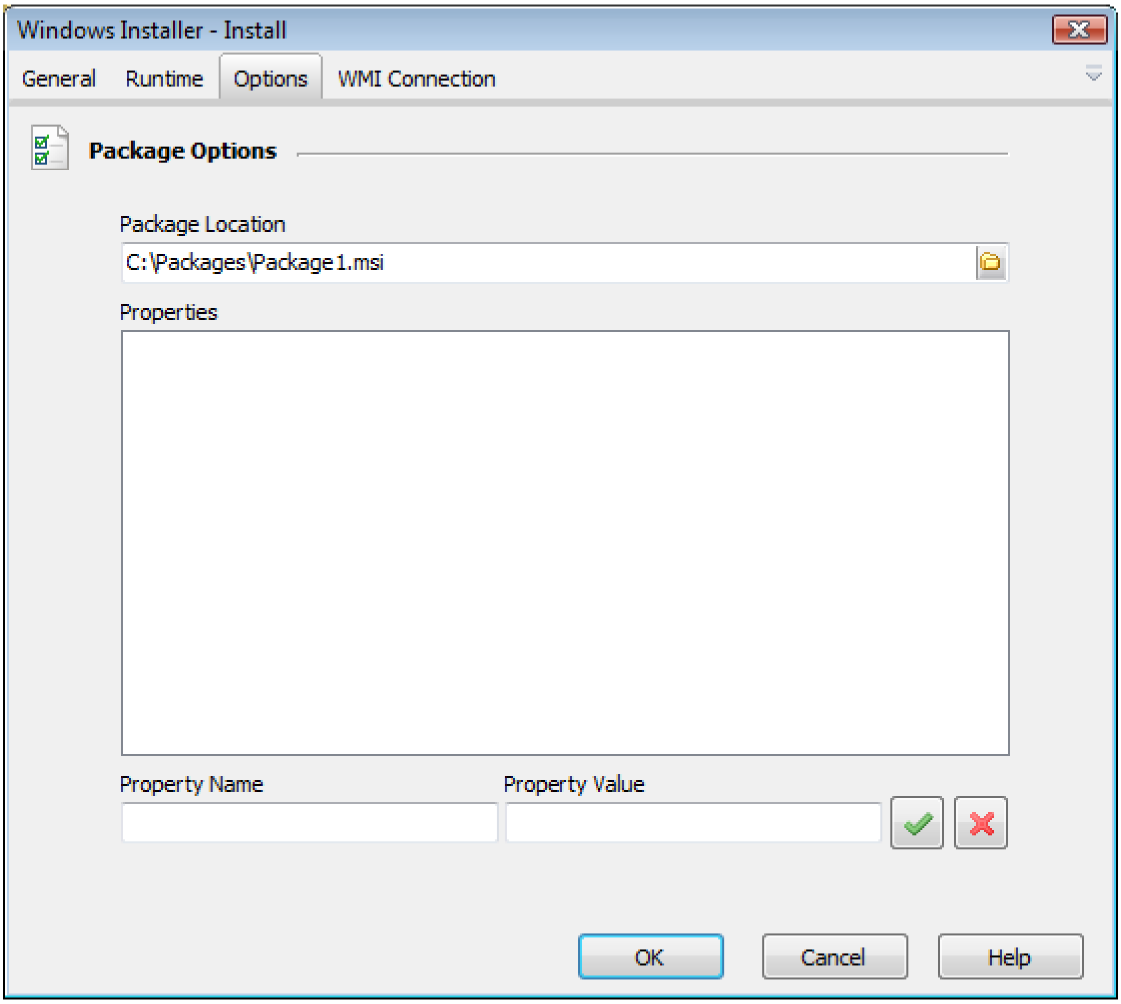
ICE Database File
The location of the ICE Database (.Cub) file to use. The MsiVal2 directory contains several .Cub files which can be used.

ICE List

A colon delimited list of the required ICE Modules to be used when performing the validation.

6.13.1.6 Windows Installer - Install

The Windows Installer - Install action enables a windows installer package to be installed on a local or remote computer.



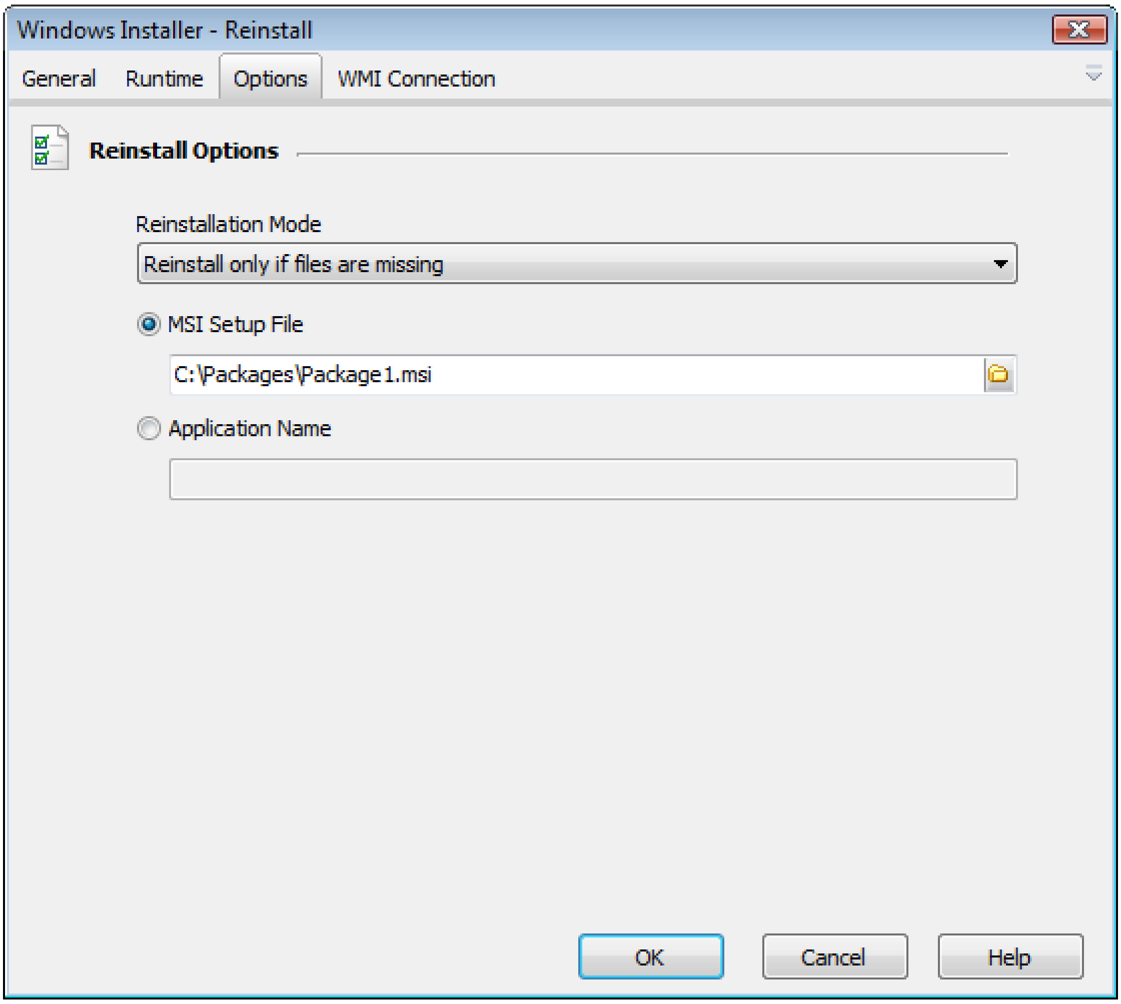
Package Location: C:\Packages\Package1.msi

Package Location
The location of the package to be installed. The package must be accessible by the computer on which it is being installed.

Properties
The properties to be passed into the installer.

6.13.1.7 Windows Installer - Reinstall

The Windows Installer - Reinstall action enables the reinstallation of an application installed by a windows installer package.



Reinstallation Mode: Reinstall only if files are missing

● MSI Setup File

C:\Packages\Package1.msi

○ Application Name

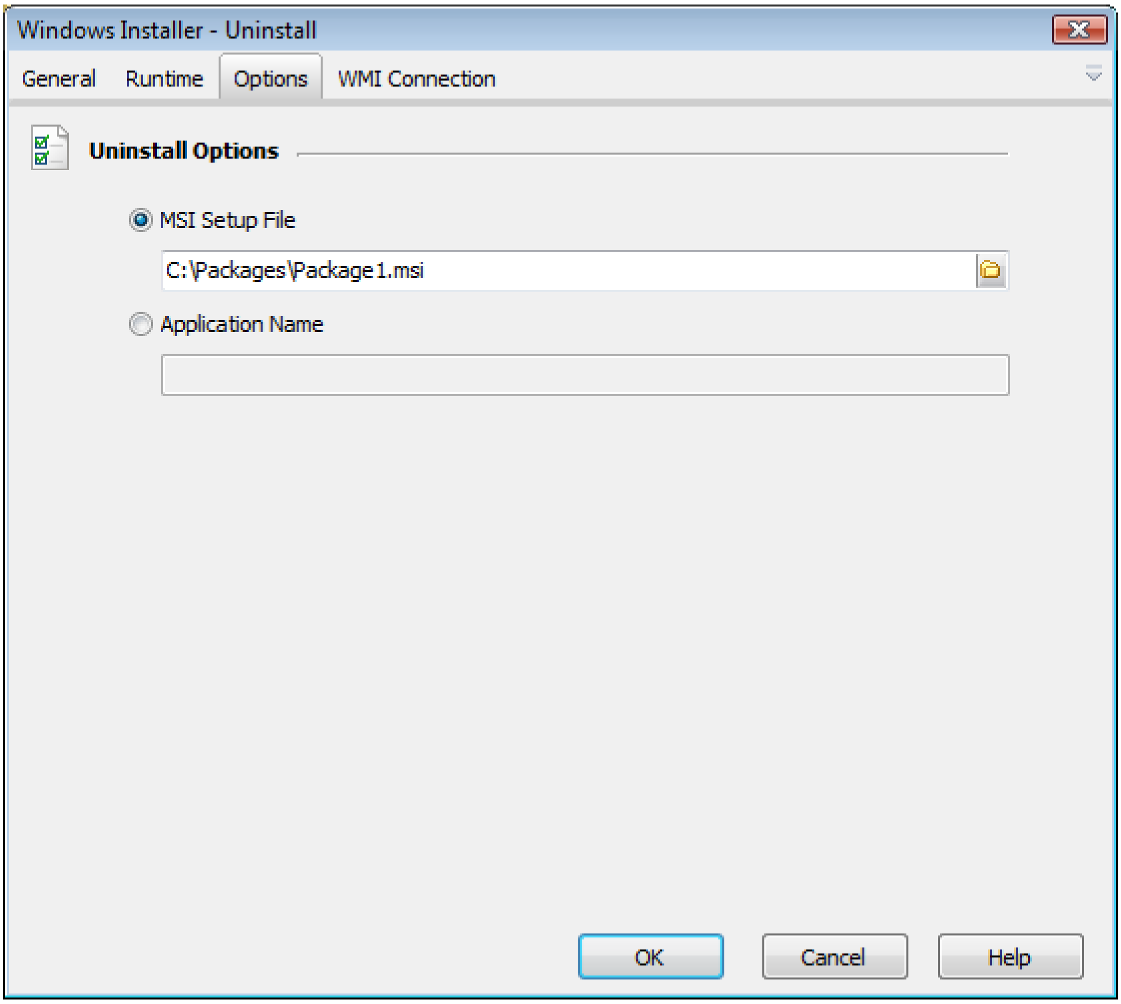
Reinstallation Mode
The type of reinstallation to be performed.

MSI Setup File
The setup file used to install the application. The setup file does not need to be accessible by the computer which the application is being uninstalled on, as it is only used to get the products identification number.

Application Name
The name of the application to uninstall. If more then one application is found with the same name, only one of the applications will be uninstalled.

6.13.1.8 Windows Installer - Uninstall

The Windows Installer - Uninstall action allows a application installed by a windows installer package to be uninstalled.



● MSI Setup File

C:\Packages\Package1.msi

○ Application Name

MSI Setup File
The setup file used to install the application. The setup file does not need to be accessible by the computer which the application is being uninstalled on, as it is only used to get the products identification number.

Application Name
The name of the application to uninstall. If more then one application is found with the same name, only one of the applications will be uninstalled.

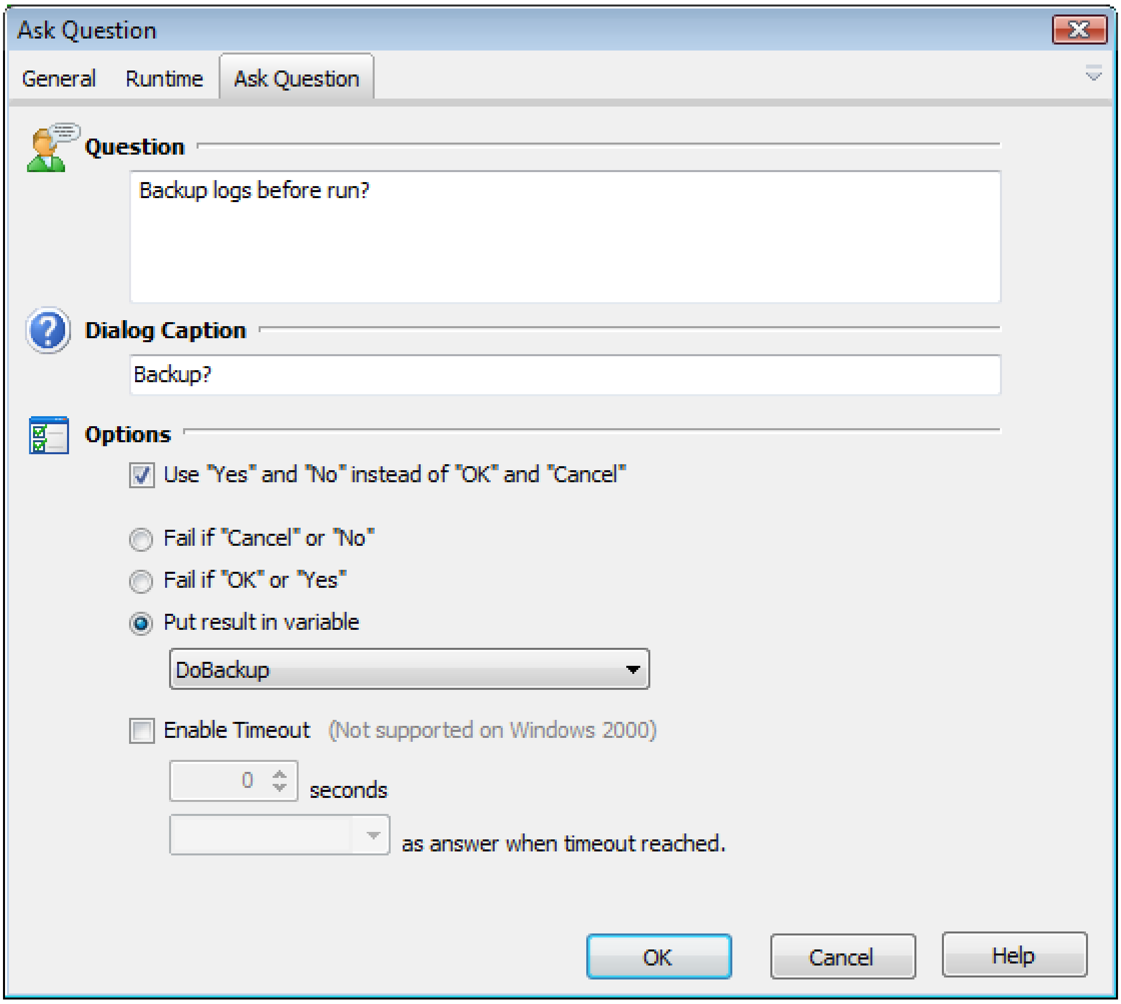
6.14 Interactive

Interactive actions can only be run in interactive mode, ie. when the build is being run manually and not from the command line or a schedule.

6.14.1 Ask Question Action

This action allows you ask the person running the project a yes/no type question. You can optionally store the result in a Automise variable. For more information on variables, including creating your own project and user variables, see here.

Note: This action should not be used unattended as it will cause them to hang waiting for user input. By default, interactive actions are not available from the Automise console. Interactive actions cannot be used inside Async Action Groups.



Ask Question

Question

Backup logs before run?

Dialog Caption

Backup?

Options

☑ Use "Yes" and "No" instead of "OK" and "Cancel"

○ Fail if "Cancel" or "No"

○ Fail if "OK" or "Yes"

● Put result in variable

DoBackup

□ Enable Timeout (Not supported on Windows 2000)

0 seconds

as answer when timeout reached.

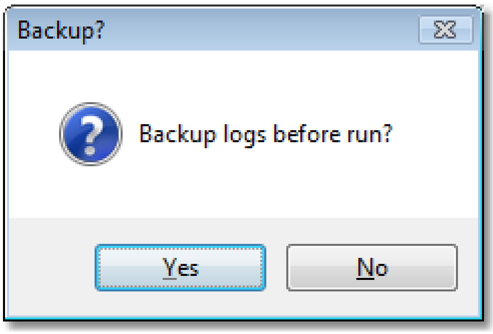
Question
This is the question that will be displayed in the dialog box. It may contain variables which will be expanded at runtime.

Dialog Caption
The text to use for the dialog caption.

Options
The dialog has two buttons, either OK and Cancel, or Yes and No - choose which style you want to use.

The behaviour of the action can be customised to fail on Cancel/No (the default), to fail if the user chooses OK/Yes, or not to fail at all and instead save the result to a variable. If you choose to store the result in a Automise variable, then OK / Yes will set the variable to True, and Cancel / No will set the variable to False

Below is an example of what the dialog will look like at runtime:

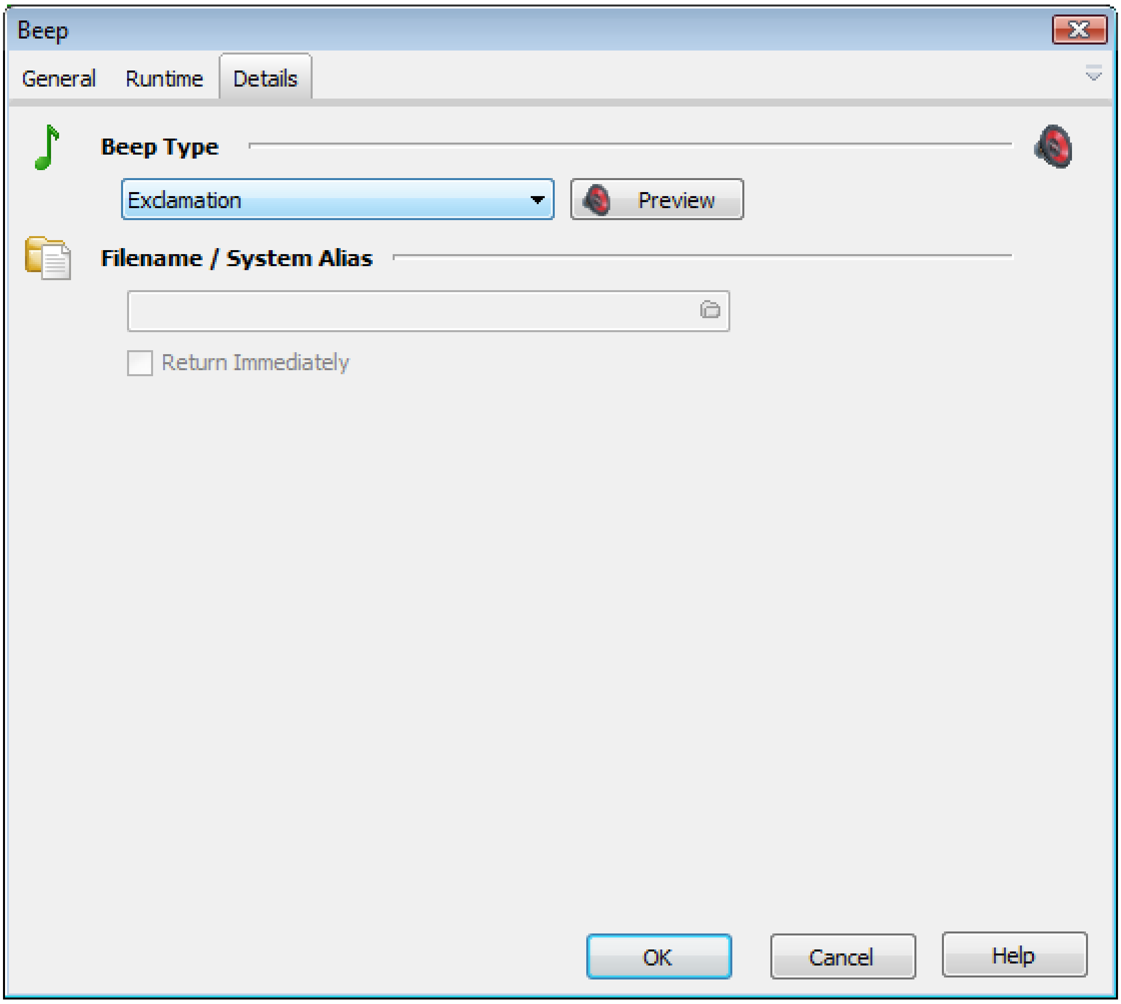


Backup?

Backup logs before run?

6.14.2 Beep Action

The Beep Action will play a sound, either one of the pre defined system sounds, or a Wave File, or a System Sound Alias. Automise creates two System Aliases, Build Complete and Build Error. You can change the sounds assigned to these aliases in the windows control panel.



Beep

Beep Type

Exclamation

Filename / System Alias

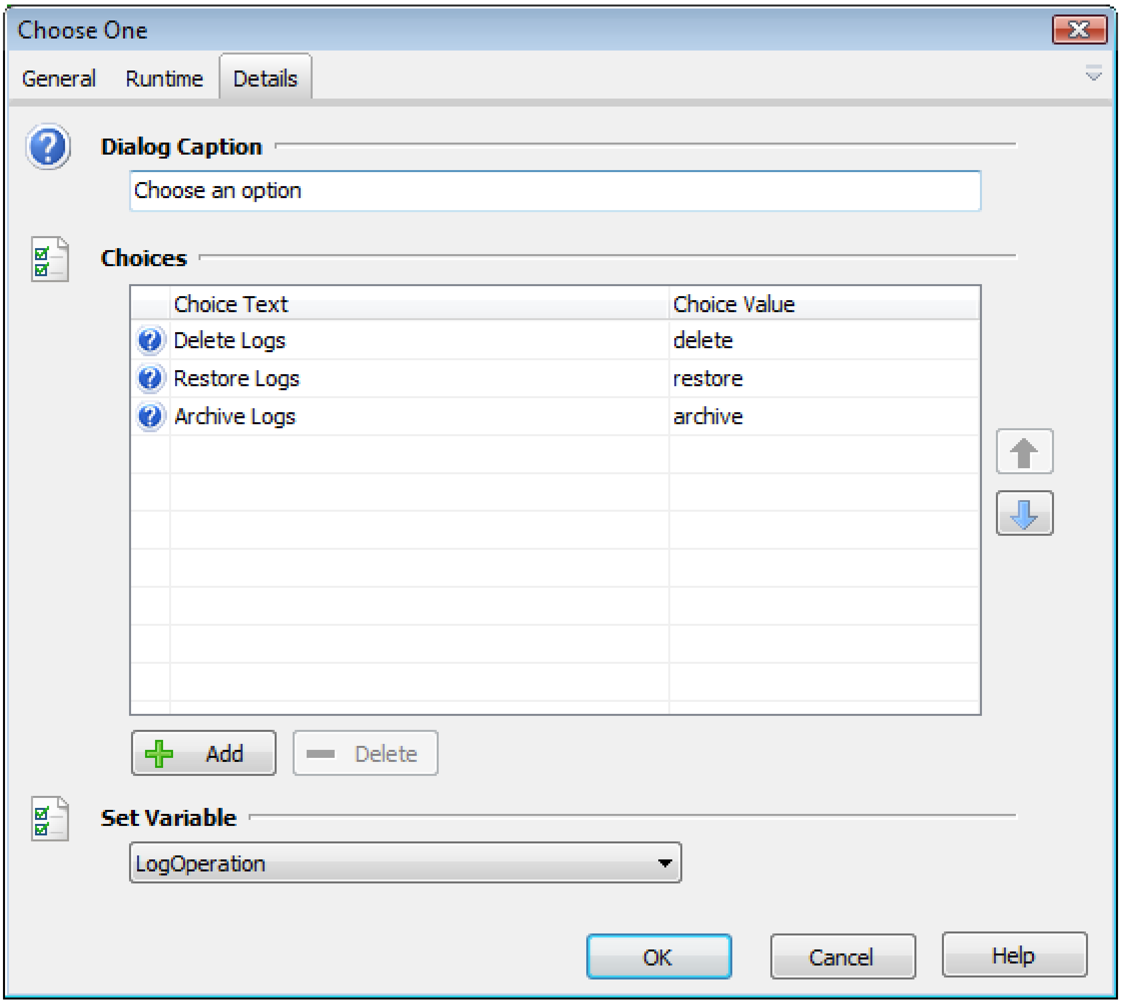
□ Return Immediately

6.14.3 Choose One Action

The Choose One action allows you to prompt the user to make a single choice given one or more options. The choices are presented as Radio Buttons.

For more information on variables, including creating your own project and user variables, see here.

Note: By default, interactive actions are not available from the Automise console. Interactive actions cannot be used inside Async Action Groups.



Choose One

General | Runtime | Details

Dialog Caption: Choose an option

Choices

Choice Text	Choice Value
Delete Logs	delete
Restore Logs	restore
Archive Logs	archive

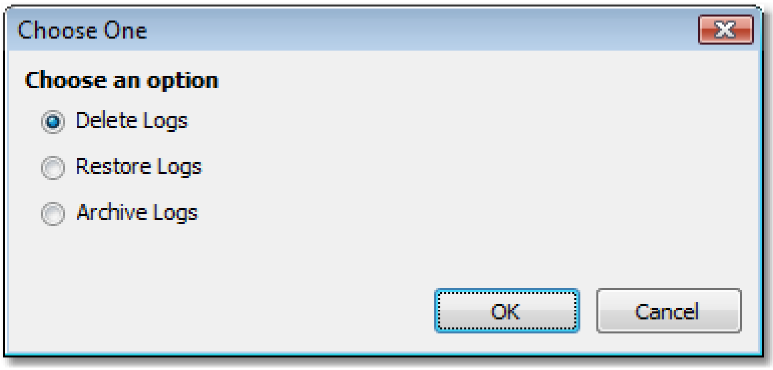
Set Variable: LogOperation

Dialog Caption
The text to display above the choices

Choices
Add one or more choices with their corresponding value. The variable will be set to the value of the chosen choice.

Set Variable
Select the variable to set to the chosen choice value.

When the action runs, the user will be prompted by the following dialog:



Choose an option

● Delete Logs

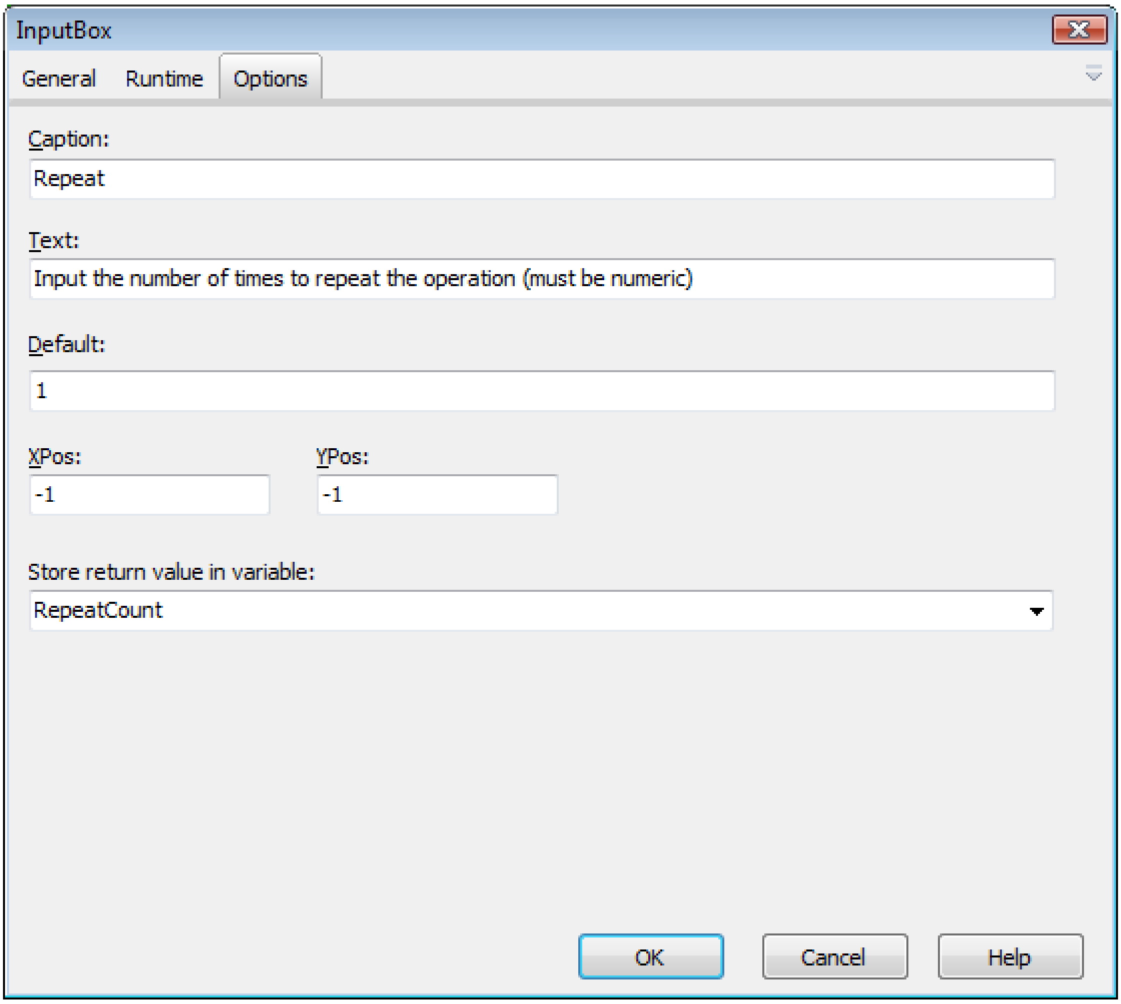
○ Restore Logs

○ Archive Logs

The first option is always chosen as default. Pressing OK will set the variable, whereas Cancel will fail the action.

6.14.4 Input Box Action

This action displays a dialog which allows the user to enter some text. The action allows a default value and a text prompt. The value entered by the user can be saved to a variable. For more information on variables, including creating your own project and user variables, see here.



Caption: Repeat

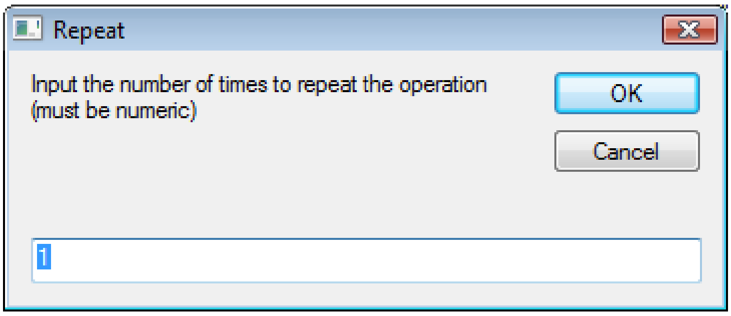
Text: Input the number of times to repeat the operation (must be numeric)

Default: 1

XPos: -1

YPos: -1

Store return value in variable: RepeatCount



Repeat

Input the number of times to repeat the operation (must be numeric)

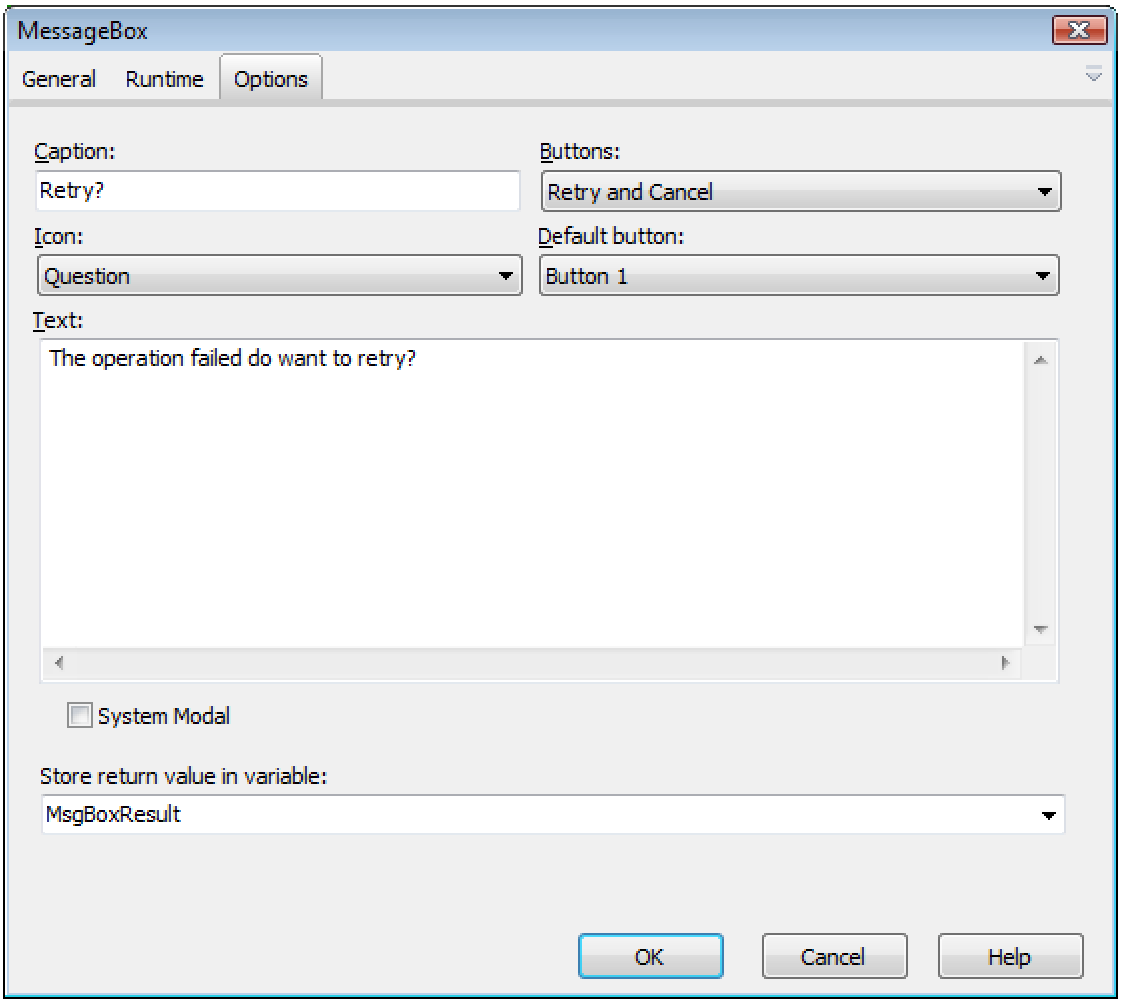
1

Note: This action should not be used in unattended projects as it will cause them to hang waiting for user input. By default, interactive actions are not available from Automise console. Interactive actions cannot be used inside Async Action Groups.

This action was donated by Peter Thornqvist.

6.14.5 Message Box Action

This action displays a standard Windows message box. The message box can contain the following icons: error, warning, exclamation, information, or none. There is also the choice of caption, text, and which buttons to display, as well as setting the default button and saving the return value (from the button clicked) to a variable. For more information on variables, including creating your own project and user variables, see here.



MessageBox

Caption: Retry?

Buttons: Retry and Cancel

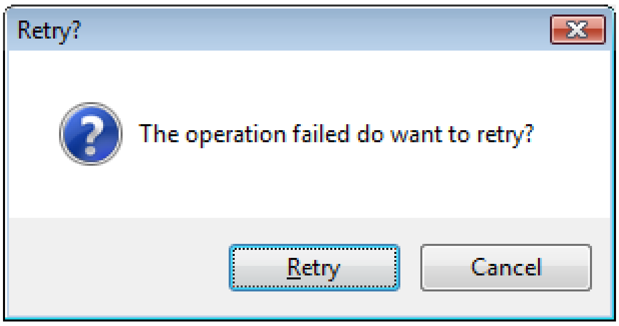
Icon: Question

Default button: Button 1

Text: The operation failed do want to retry?

☐ System Modal

Store return value in variable: MsgBoxResult



Retry?

The operation failed do want to retry?

Note: This action should not be used in unattended projects as it will cause them to hang waiting for user input. By default, interactive actions are not available from Automise console. Interactive actions cannot be used inside Async Action Groups.

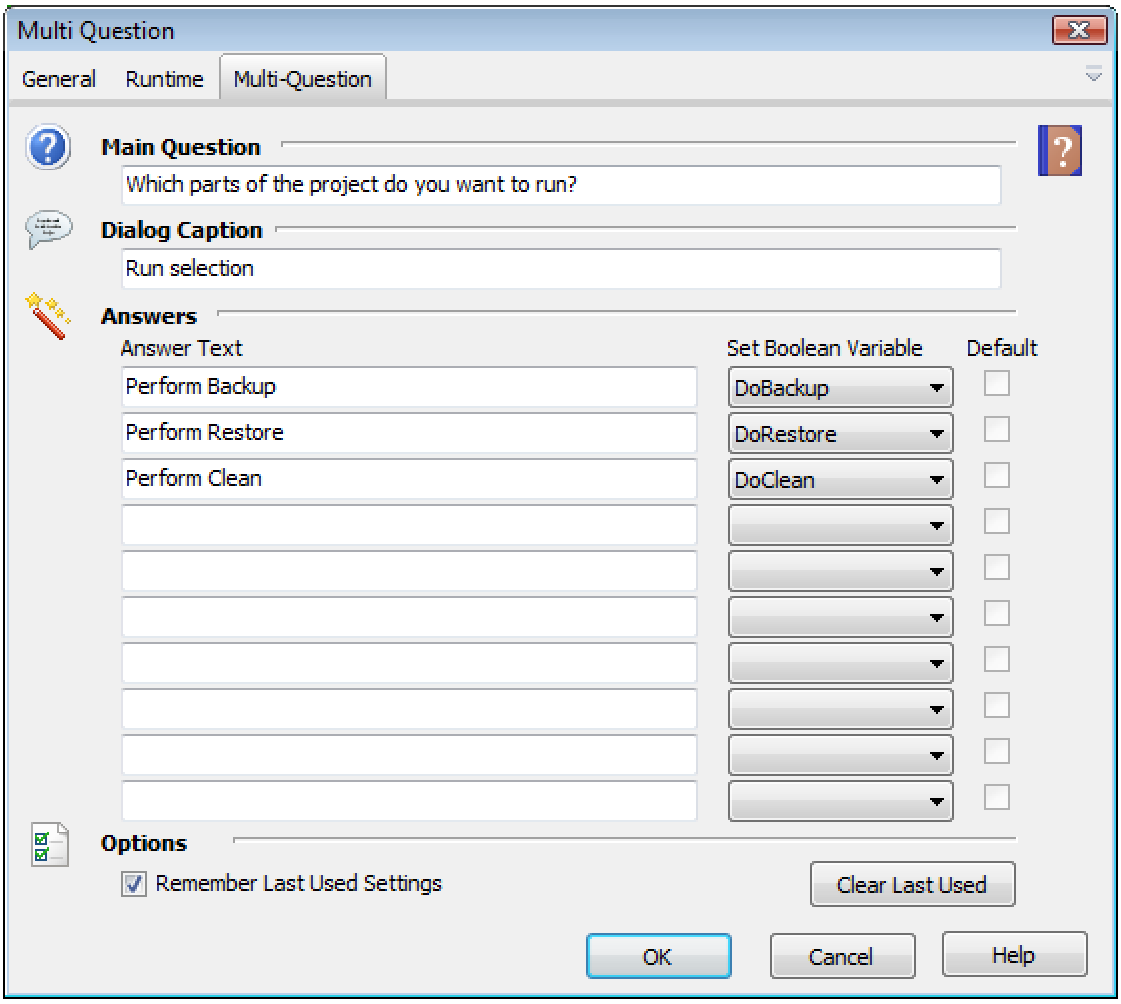
This action was donated by Peter Thornqvist.

6.14.6 Multi Question

The MultiQuestion action enables you to interact with the user during the execution of your project. Each answer can be either set to True or False which is then saved in Automise variables which can then control the flow of your project.

For more information on variables, including creating your own project and user variables, see here.

Note: This action should not be used unattended as it will cause them to hang waiting for user input. By default, interactive actions are not available from the Automise console. Interactive actions cannot be used inside Async Action Groups.



Main Question

Which parts of the project do you want to run?

Dialog Caption

Run selection

Answers

Answer Text	Set Boolean Variable	Default
Perform Backup	DoBackup	□
Perform Restore	DoRestore	□
Perform Clean	DoClean	□
		□
		□
		□
		□
		□
		□
		□

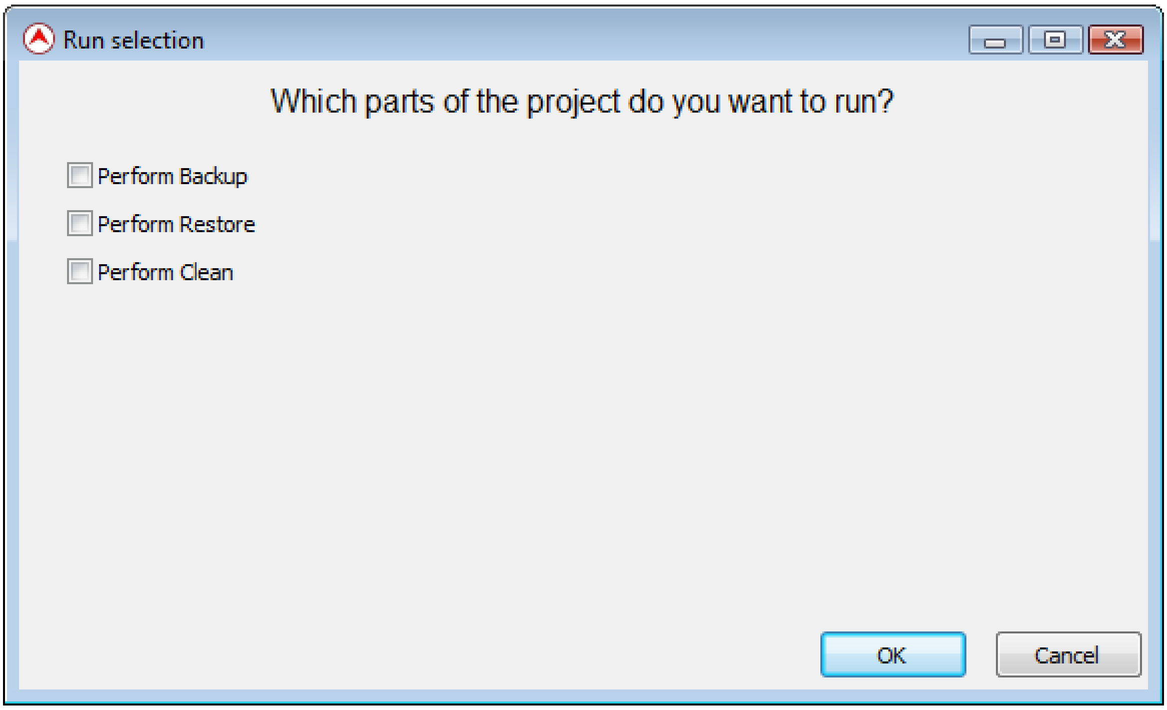
Options

☑ Remember Last Used Settings

For Example, the main question could be "What parts of the project do you want to run today". Answer 1: "Source", Answer 2: "Run Install", Answer 3: "Deploy". The boolean answers can then be used in the Condition property of other actions to control whether they are executed or not.

If you specify **Remember Last Used Settings** then the default value when the action runs will be set to the previously selected values when the action was last run.

This is what the action looks like at run time:



Run selection

Which parts of the project do you want to run?

□ Perform Backup

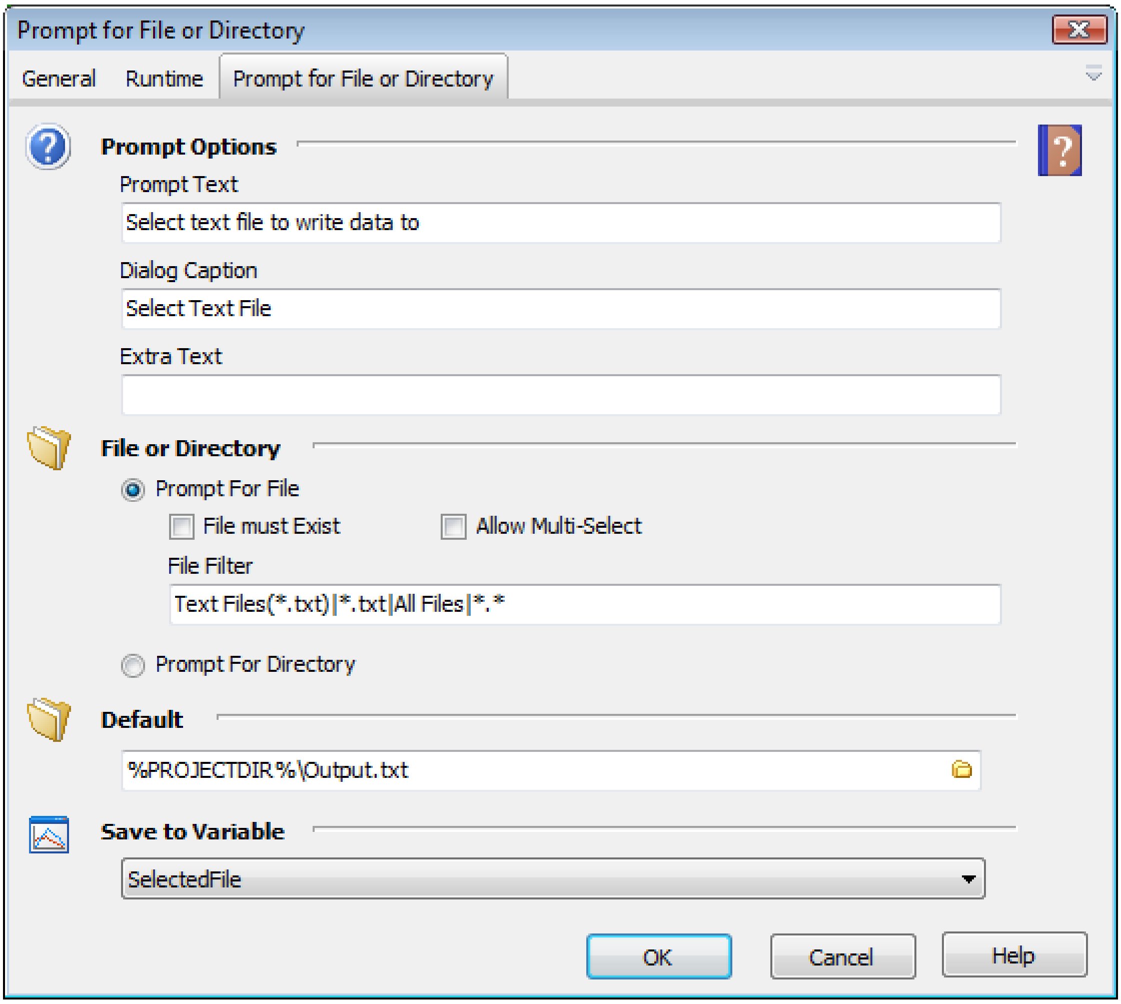
□ Perform Restore

□ Perform Clean

6.14.7 Prompt for File or Directory

The Prompt for File or Directory action enables you to ask the user to specify a file or directory during your project. The file or directory specified is saved in a variable so that it can be used in subsequent actions. For more information on variables, including creating your own project and user variables, see here.

Note: This action should not be used in unattended projects as it will cause them to hang waiting for user input. By default, interactive actions are not available from Automise console. Interactive actions cannot be used inside Async Action Groups.



Prompt for File or Directory

Prompt Options

Prompt Text: Select text file to write data to

Dialog Caption: Select Text File

Extra Text:

File or Directory

● Prompt For File

□ File must Exist □ Allow Multi-Select

File Filter: Text Files(*.txt)|*.txt|All Files|*.*

○ Prompt For Directory

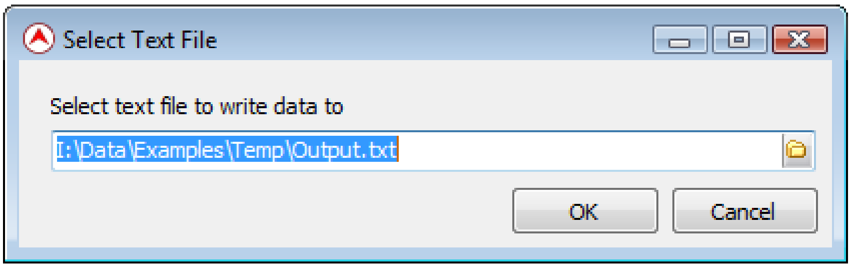
Default

%PROJECTDIR%\Output.txt

Save to Variable

SelectedFile

When the action is executed it looks something like this:



Select Text File

Select text file to write data to

I:\Data\Examples\Temp\Output.txt

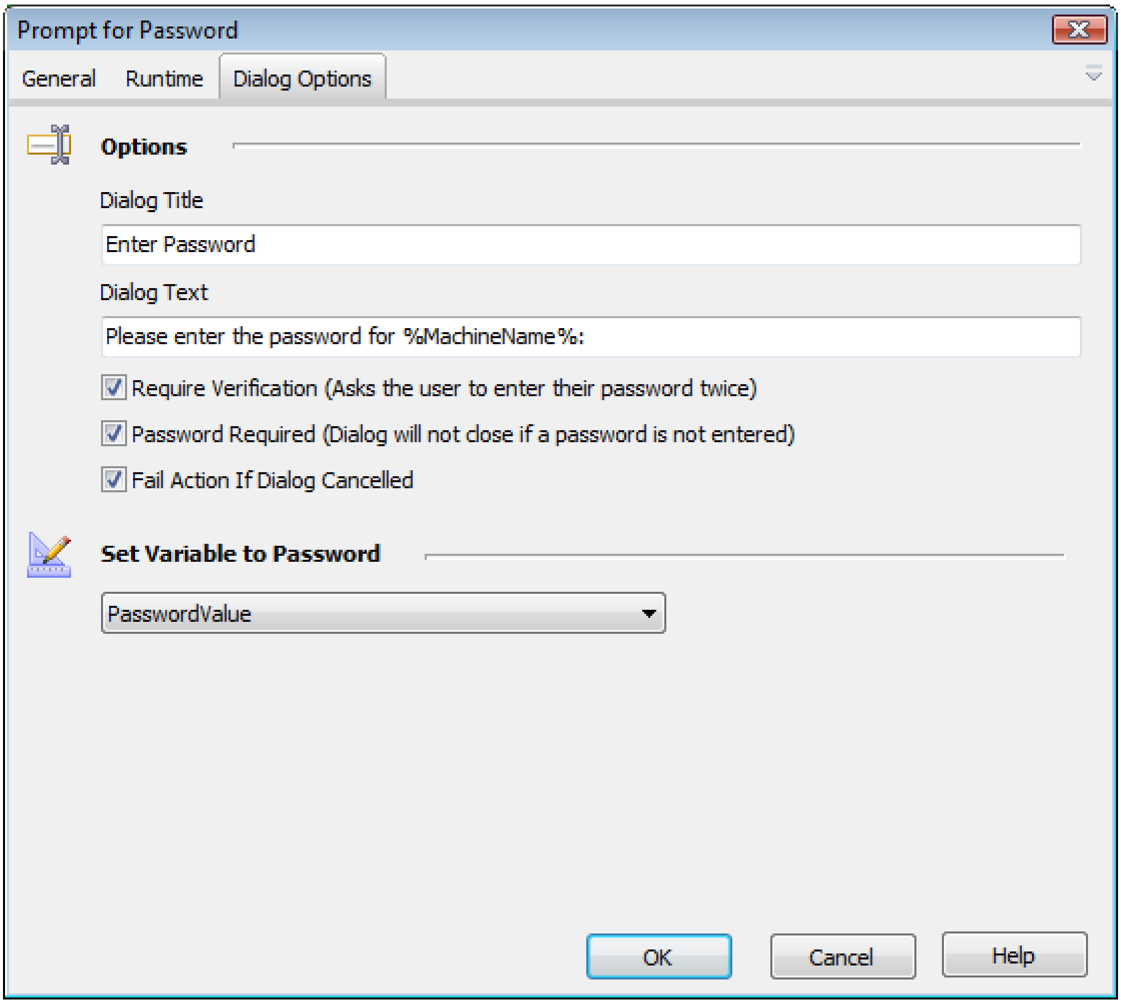
6.14.8 Prompt for Password Action

This action allows you to prompt the user to enter a password at run time.

For more information on variables, including creating your own project and user variables, see here.

Note: By default, interactive actions are not available from the Automise console. Interactive actions cannot be used inside Async Action Groups.

Prompt for Password Options



Prompt for Password

Options

Dialog Title

Enter Password

Dialog Text

Please enter the password for %MachineName%:

☑ Require Verification (Asks the user to enter their password twice)

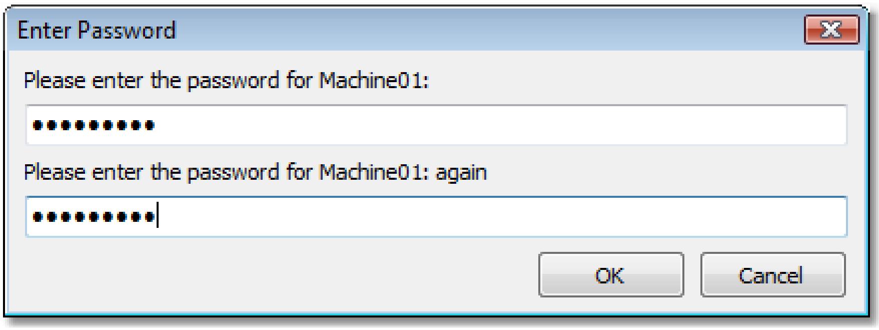
☑ Password Required (Dialog will not close if a password is not entered)

☑ Fail Action If Dialog Cancelled

Set Variable to Password

PasswordValue

Prompt for Password Dialog



Enter Password

Please enter the password for Machine01:

••••••••••

Please enter the password for Machine01: again

••••••••••

Dialog Title
Specify the title shown in the password prompt.

Dialog Text
Specify the text given to the user in the prompt. This value will be used to automatically generate the verification text.

Require Verification
Normally the dialog will prompt the user to enter a password in once, this can be changed so that the password must be entered twice by setting this option.

Password Required
Enabling this option forces the user to either enter a password or cancel the dialog.

Fail Action If Dialog Cancelled
By default this action will fail if the user cancels or closes the dialog. Changing this option will allow the build to continue even if the dialog was cancelled.

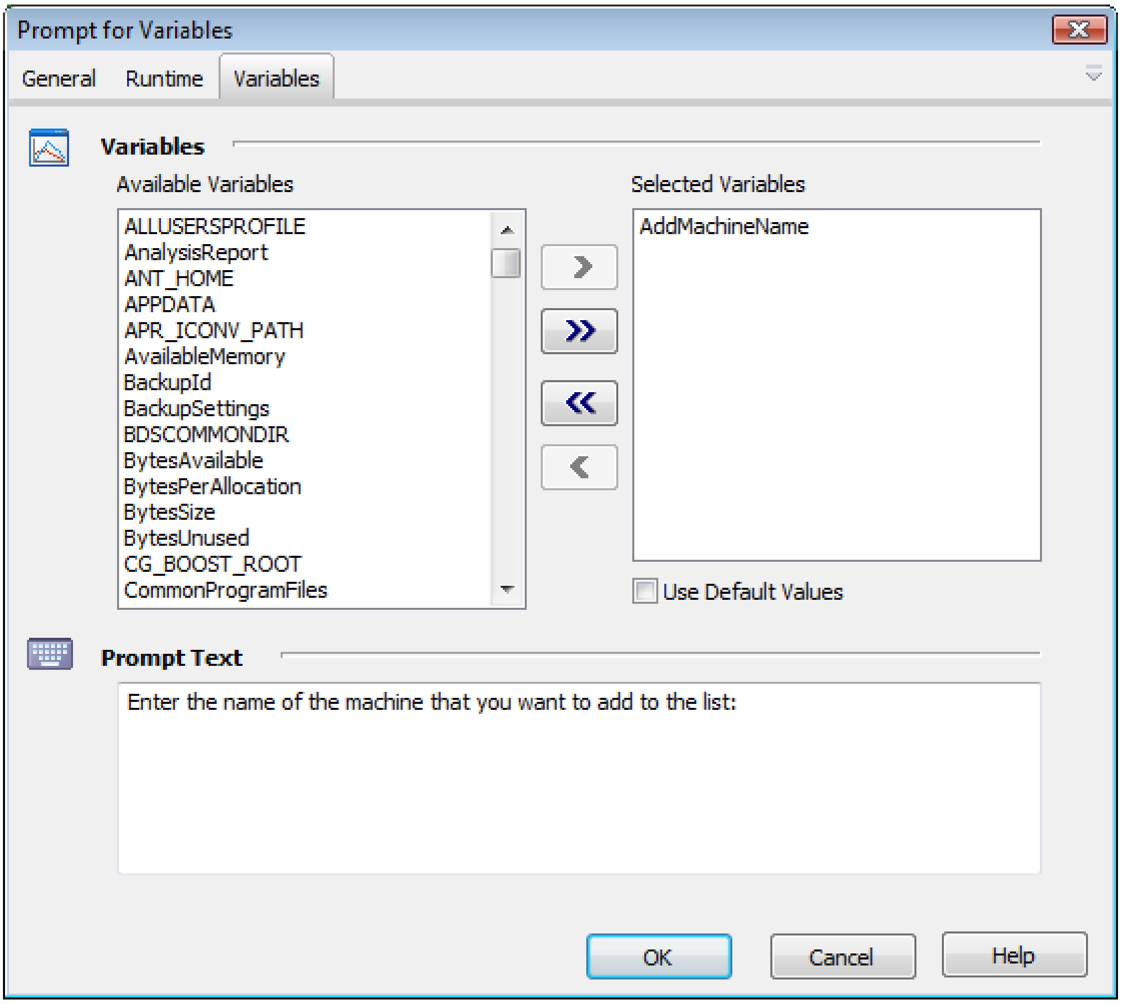
Set Variable to Password
Select the variable for the password to be saved back into. This variable may then be subsequently used in other actions that require the password.

6.14.9 Prompt for Variables Action

This action allows you to prompt the user for variable values at run time. For more information on variables, including creating your own project and user variables, see here.

Note: This action should not be used unattended, as it will cause them to hang waiting for user input. By default, interactive actions are not available from the Automise console. Interactive actions cannot be used inside Async Action Groups.

For more sophisticated prompts requirements, use the Prompt For Variables (Enhanced) action.



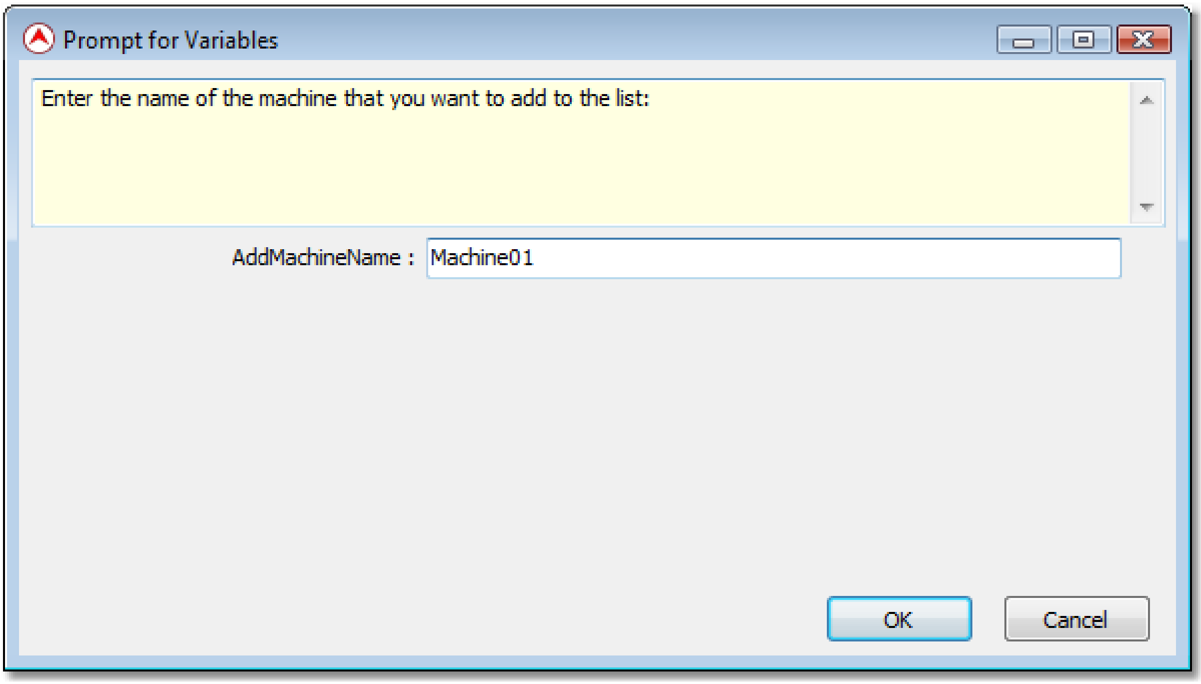
Enter the name of the machine that you want to add to the list:

Available Variables & Selected Variables - Move Variables from the Available Variables side to the Selected Variables side to prompt for them when the action runs

Use Default Values - fill out the variable value at runtime with the default value of the variable. If this option is not selected, the variable value will be blank.

Prompt Text - the text that will be placed at the top of the dialog

At runtime the above configuration would cause this form to be displayed when the action executes.



Enter the name of the machine that you want to add to the list:

AddMachineName : Machine01

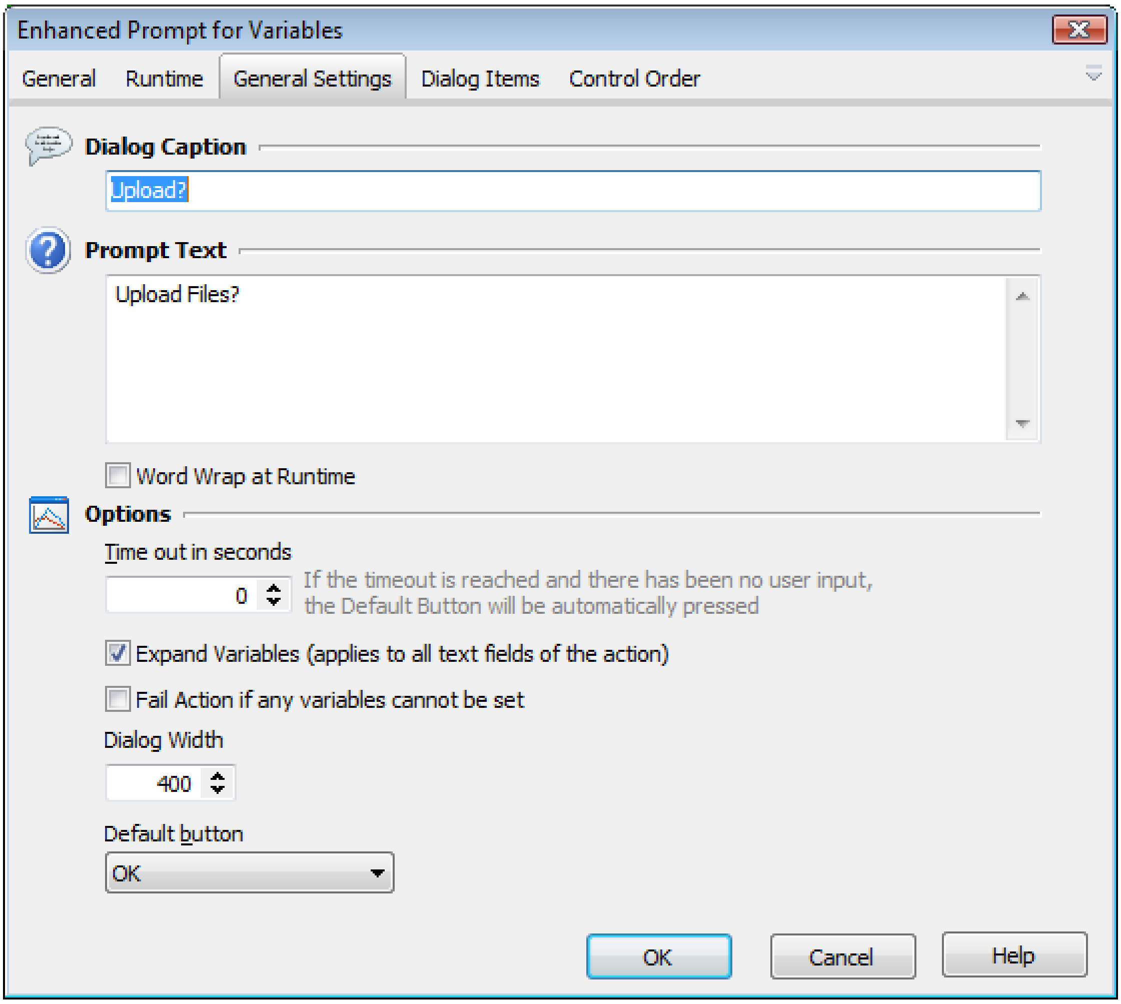
6.14.10 Enhanced Prompt for Variables Action

This action (kindly donated by Peter Thörnqvist) is an enhanced version of the Prompt for Variables action that allows you to specify the control types to be used at runtime in the prompt form. The action also has more features and allows greater customisation.

For more information on variables, including creating your own project and user variables, see here.

Note: By default, interactive actions are not available from the Automise console. Interactive actions cannot be used inside Async Action Groups.

The general settings lets you customise the dialog appearance.



Enhanced Prompt for Variables

Dialog Caption

Upload?

Prompt Text

Upload Files?

□ Word Wrap at Runtime

Options

Time out in seconds

0

If the timeout is reached and there has been no user input, the Default Button will be automatically pressed

☑ Expand Variables (applies to all text fields of the action)

□ Fail Action if any variables cannot be set

Dialog Width

400

Default button

OK

Dialog Caption - the text to appear in the dialog header.

Prompt Text - the text to appear at the top of the dialog (optional).

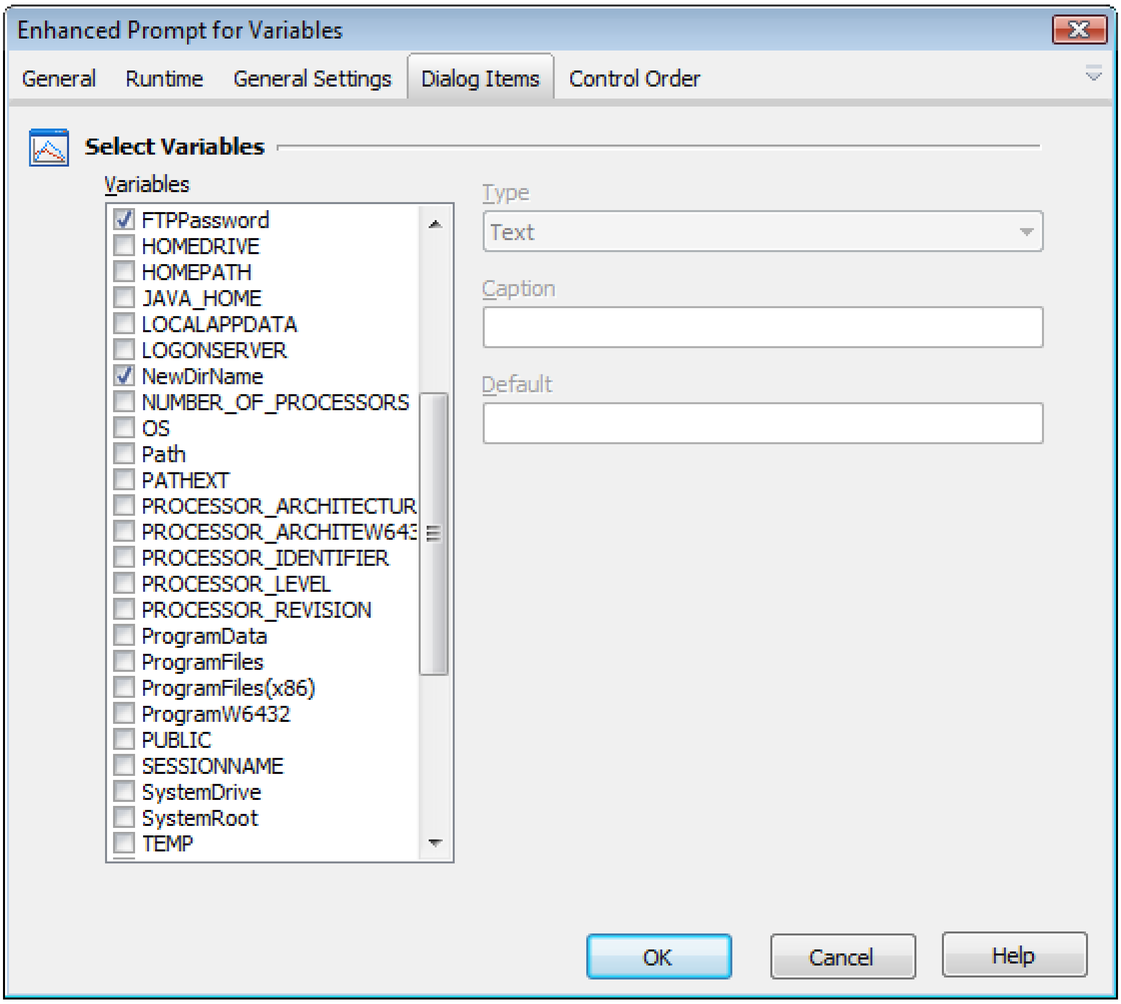
Word Wrap at Runtime - set this to automatically wrap the prompt text according to the width of the dialog.

Time Out In Seconds - The dialog will automatically close if the user hasn't interacted with the dialog for the specified time

Expand Variables - check this to replace variables with their value in the dialog

Dialog Width - set the width of the dialog at runtime. You may want to adjust this to a larger width to accommodate large pathnames or other options that require a lot of horizontal space. Be aware that the size of the dialog isn't too big for your user's monitors.

Default Button - This used for the timeout to choose a suitable action when the timeout is reached



Variables

☑ FTPPassword
☐ HOMEDRIVE
☐ HOMEPATH
☐ JAVA_HOME
☐ LOCALAPPDATA
☐ LOGONSERVER
☑ NewDirName
☐ NUMBER_OF_PROCESSORS
☐ OS
☐ Path
☐ PATHEXT
☐ PROCESSOR_ARCHITECTUR
☐ PROCESSOR_ARCHITEW643
☐ PROCESSOR_IDENTIFIER
☐ PROCESSOR_LEVEL
☐ PROCESSOR_REVISION
☐ ProgramData
☐ ProgramFiles
☐ ProgramFiles(x86)
☐ ProgramW6432
☐ PUBLIC
☐ SESSIONNAME
☐ SystemDrive
☐ SystemRoot
☐ TEMP

Type: Text

Caption:

Default:

Variables - this list allows you to select which variables should be added to the dialog at runtime. Selecting a variable that is checked allows you to configuring the settings for that variable.

Type - choose the type of the variable (this determines the control used at runtime).

- Text - a single line text field
- Memo - a 6 line text field
- Integer - a text field that only allows numbers to be entered
- Float - a text field that only allows floating point values to be entered
- Boolean - a checkbox
- Date - a date chooser control
- Time - a time chooser control
- Sorted List - a combobox control (drop down list) that contains the sorted items entered in Values (one item per line)
- Unsorted List - a combobox control (drop down list) that contains the items entered in Values (one item per line)
- Select File - a text field that also allows browsing to a file
- Select Folder - a text field that also allows browsing to a folder
- Check List - a list of checkboxes with the values as entered in Values. The variable will be set to the text of the checkbox chosen.
- Text (Password) - a single line text field that displays it's value as * so that others can't read the password entered

Caption - the name of the control as it will appear on the dialog at runtime

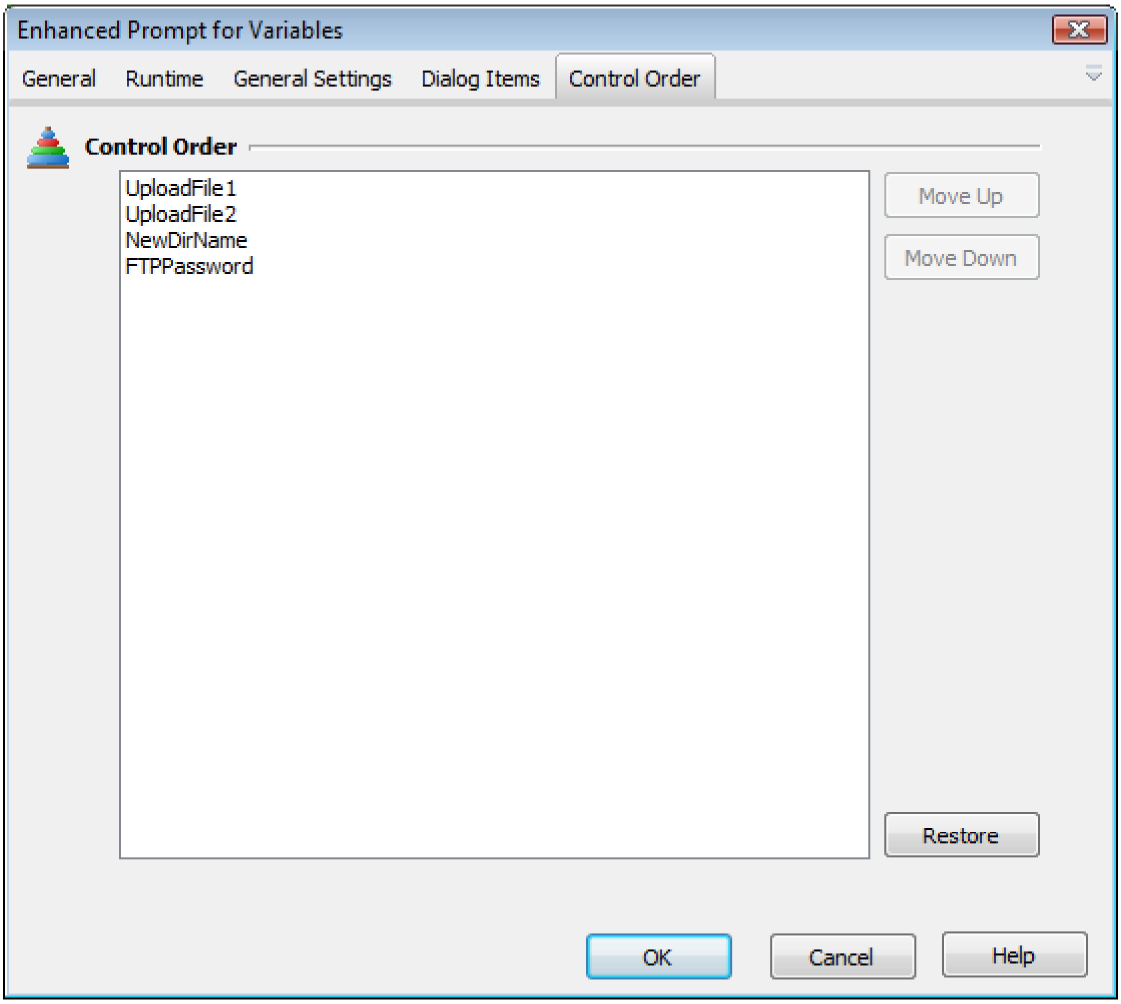
Default - the default value of the control

Values can have initial defaults (which can be variable values or fixed values.)

For "Check List" type prompts, specify default values as a comma-delimited list of values to enable at runtime.

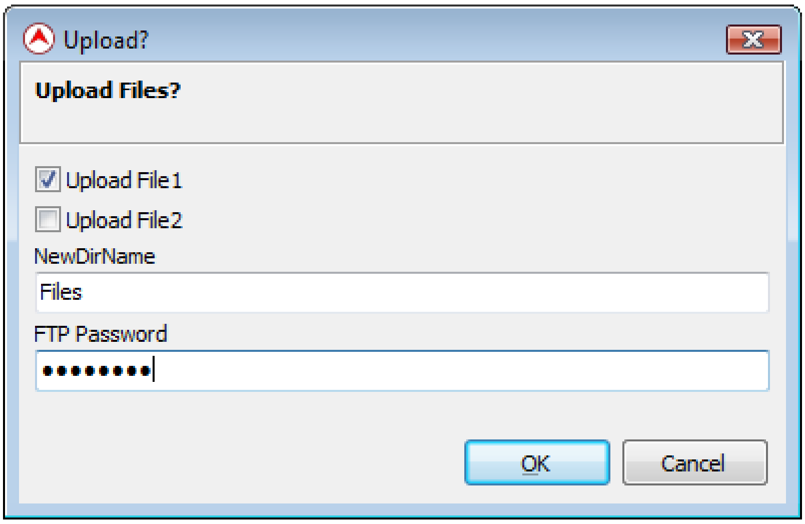
Control Order

This tab allows you to set the order of the various controls as they will appear on the runtime dialog



UploadFile1
UploadFile2
NewDirName
FTPPassword

This is the prompt form at Runtime, showing 4 variables. Note that clicking on the Cancel button would cause the action to fail and stop the run. Also notice that a timeout has been set and is currently at 23 seconds.



Upload Files?

☑ Upload File1

□ Upload File2

NewDirName

Files

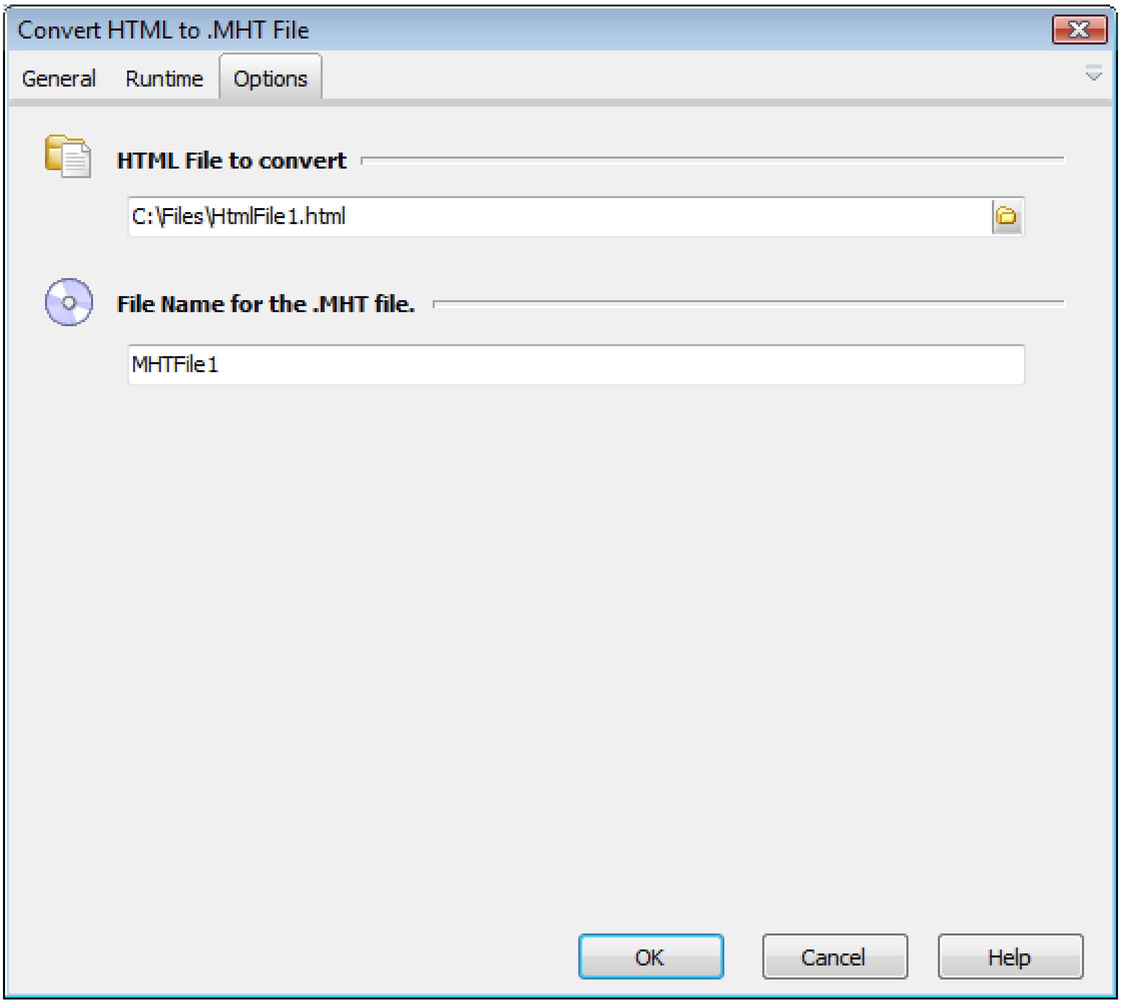
FTP Password

●●●●●●●●●

6.15 Internet

6.15.1 Convert HTML to MHT File

This Action will take an HTML file and convert it to a .MHT file. An .MHT file is a Multipurpose Internet Mail Extension HTML (MHTML) which is basically an archived Web Page. All relative links in the Web page are remapped and the embedded content is included in the .MHT file. The absolute references or hyperlinks on the web page remain unchanged and the .MHT file can then be viewed using Internet Explorer.



HTML File to convert

C:\Files\HtmlFile1.html

File Name for the .MHT file.

MHTFile1

The options for the action are:

HTML File to Convert - the name of the HTML file on disk to convert to MHT

File Name for the .MHT file - the destination MHT file will be written to this file name.

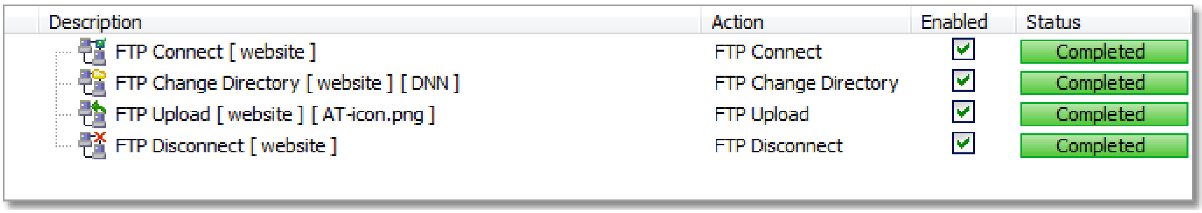
[This action was kindly donated by Robert Kozak]

6.15.2 FTP Actions

The FTP Actions allow you to add various FTP commands to your project.

Use the FTP Connection to define a connection, and then specify this connection in the other FTP actions. Your project should include an FTP Disconnect action to close the FTP connection.

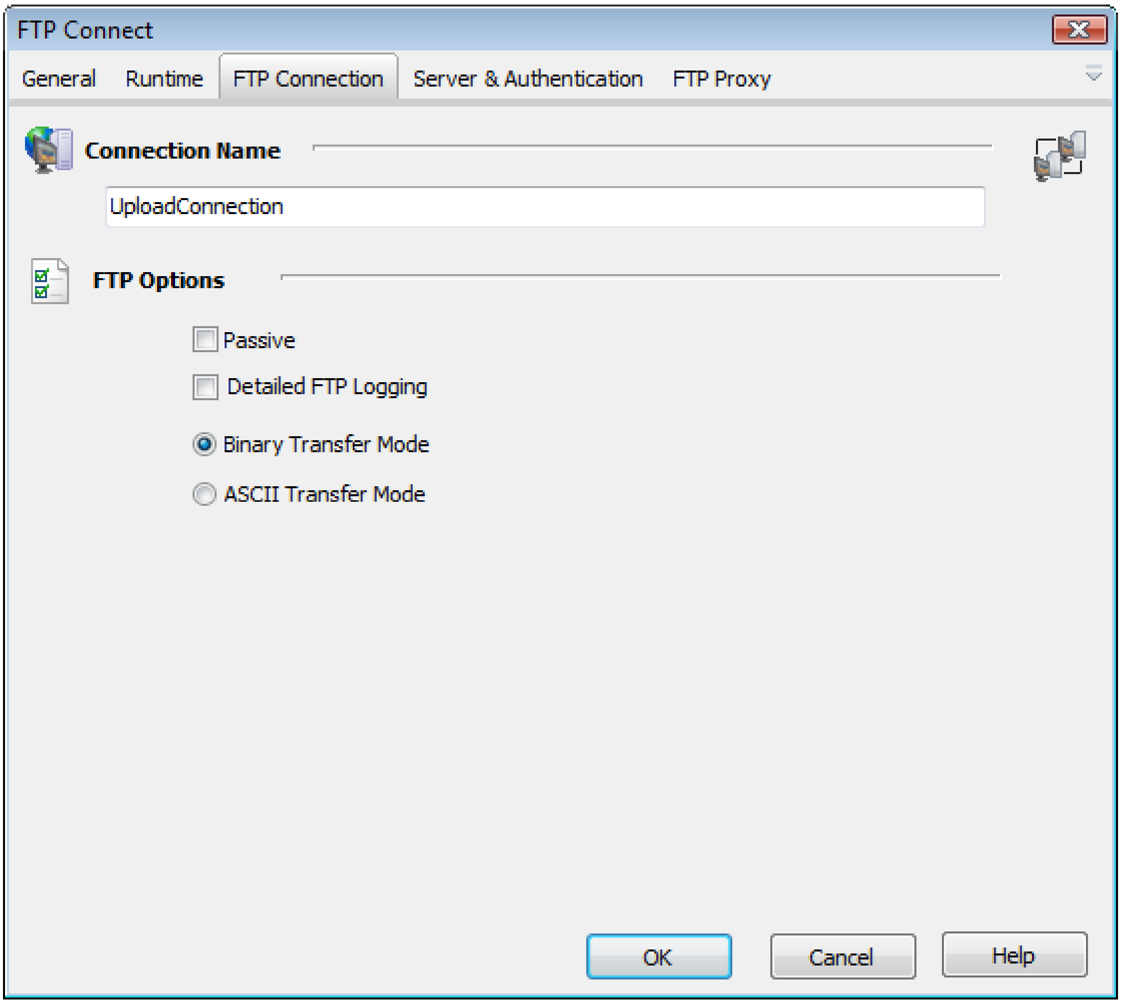
Example:



Description	Action	Enabled	Status
FTP Connect [website]	FTP Connect	☑	Completed
FTP Change Directory [website] [DNN]	FTP Change Directory	☑	Completed
FTP Upload [website] [AT-icon.png]	FTP Upload	☑	Completed
FTP Disconnect [website]	FTP Disconnect	☑	Completed

FTP Connect

This action is used to define the connection name (for use in the other FTP actions), and will also connect to the FTP server during your build using the supplied credentials.



Connection Name

UploadConnection

FTP Options

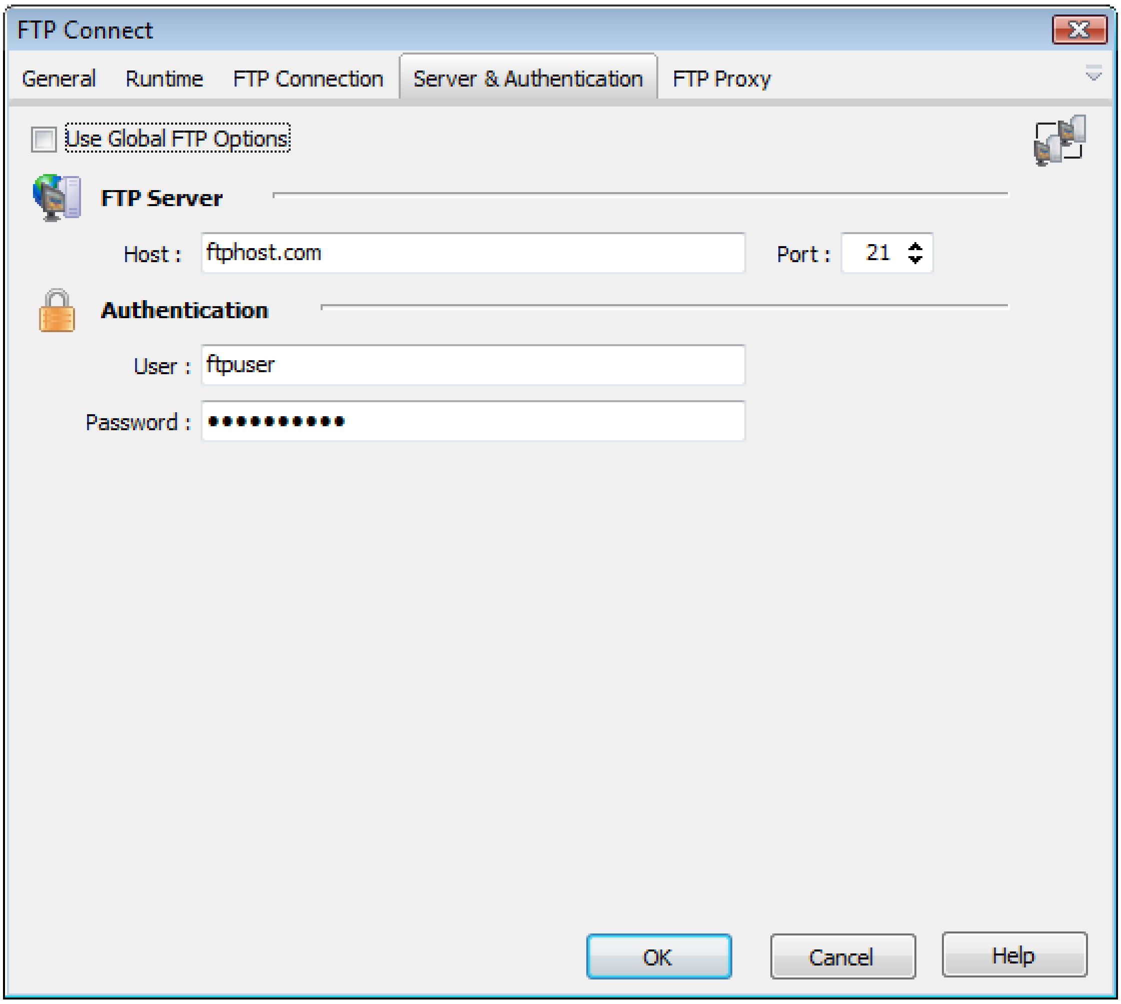
□ Passive

□ Detailed FTP Logging

● Binary Transfer Mode

○ ASCII Transfer Mode

- Connection Name - specify a unique name for the connection. This is used in the other FTP actions to select which connection to use.
- Passive - Instructs the FTP client and server to use passive mode transfers, this is needed when going through some firewalls
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Binary/ASCII Transfer - Sets the transfer mode to either binary or text.
- FTP Server - specify the internet address of the FTP server to connect to. You can either use an IP or hostname.
- Port - specify the port to use for FTP. The standard port number is 21.
- Authentication - if the FTP connection requires authentication, then enter the username and password.
- FTP Proxy - specify the proxy settings if using a proxy server.



FTP Connect

□ Use Global FTP Options

FTP Server

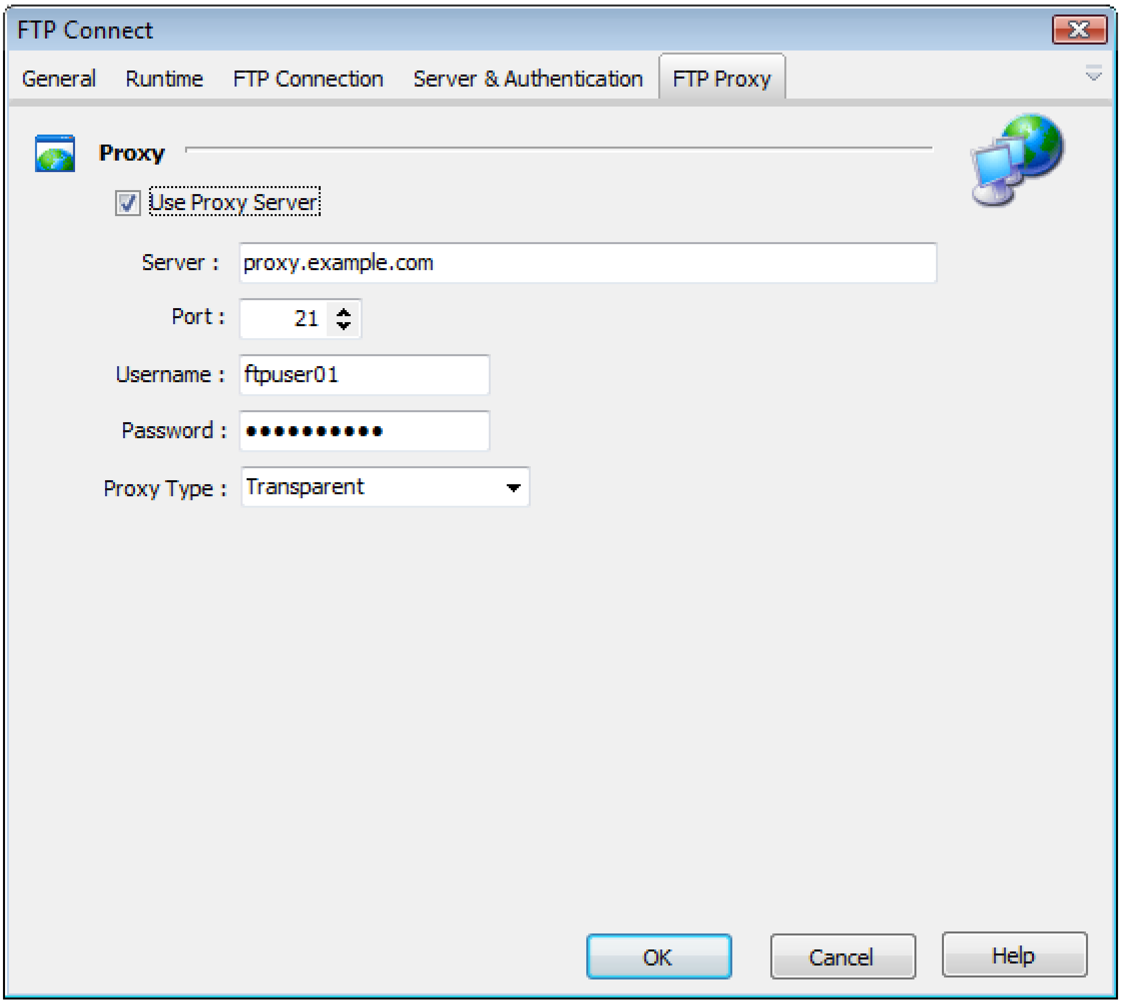
Host : ftphost.com

Port : 21

Authentication

User : ftpuser

Password : ●●●●●●●●●●●



☑ Use Proxy Server

Server : proxy.example.com

Port : 21

Username : ftpuser01

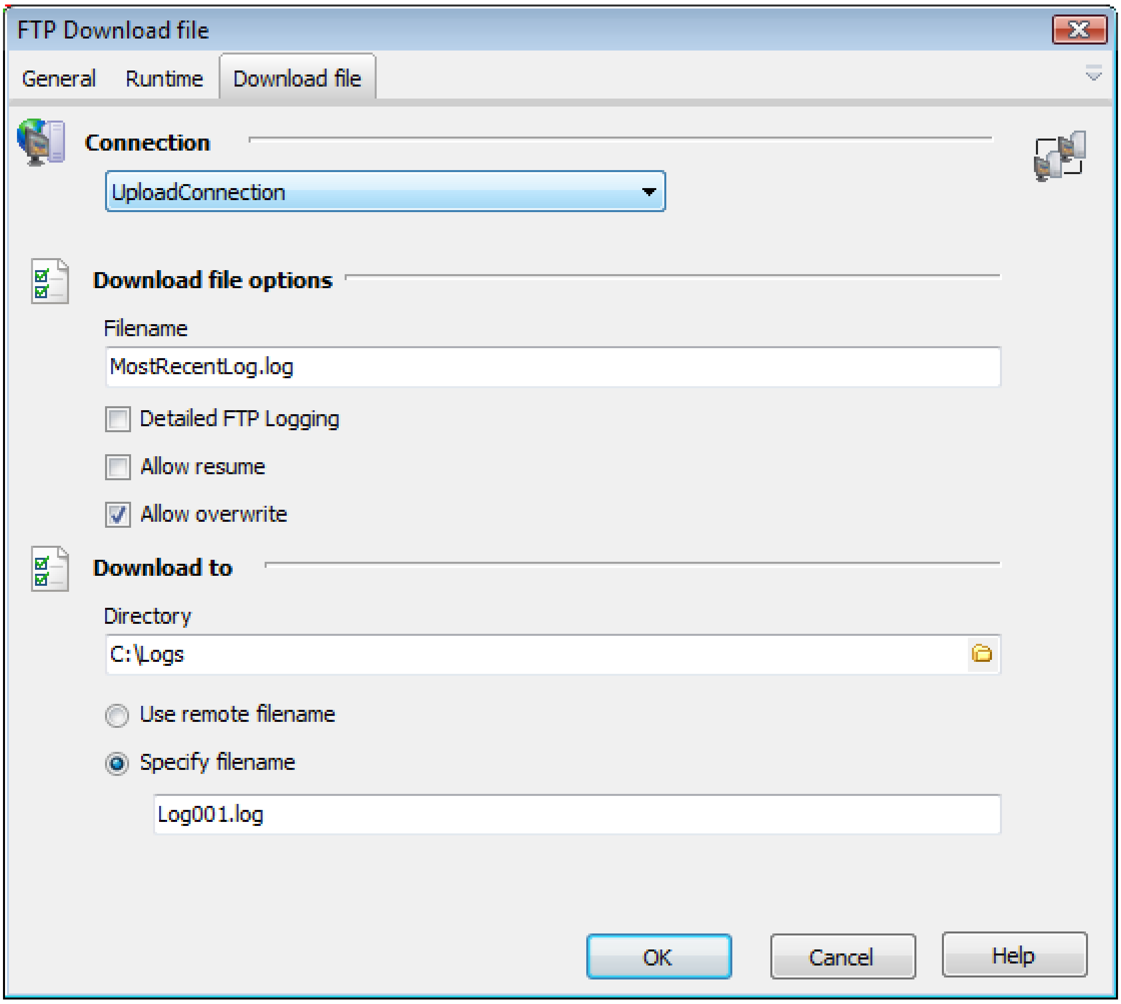
Password : ●●●●●●●●●●

Proxy Type : Transparent

FTP Download File

This action is used to download a specified file from an FTP server to the local file system. Before running this action, you may want to use a FTP Change Directory action to change the remote directory.

- Connection Name - select the connection (see FTP Connect action)
- Filename - specify the name of the remote file to download
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Allow resume - if the file exists locally and is incomplete, then the download will attempt to resume the build
- Allow overwrite - if the file exists locally the action will not fail, but will delete the local file before downloading
- Download to directory - specify the local directory to save the file to
- Use remote/specified file - specify to use the remote filename for the local copy, or to use a different filename



Connection

UploadConnection

Download file options

Filename

MostRecentLog.log

□ Detailed FTP Logging

□ Allow resume

☑ Allow overwrite

Download to

Directory

C:\Logs

○ Use remote filename

● Specify filename

Log001.log

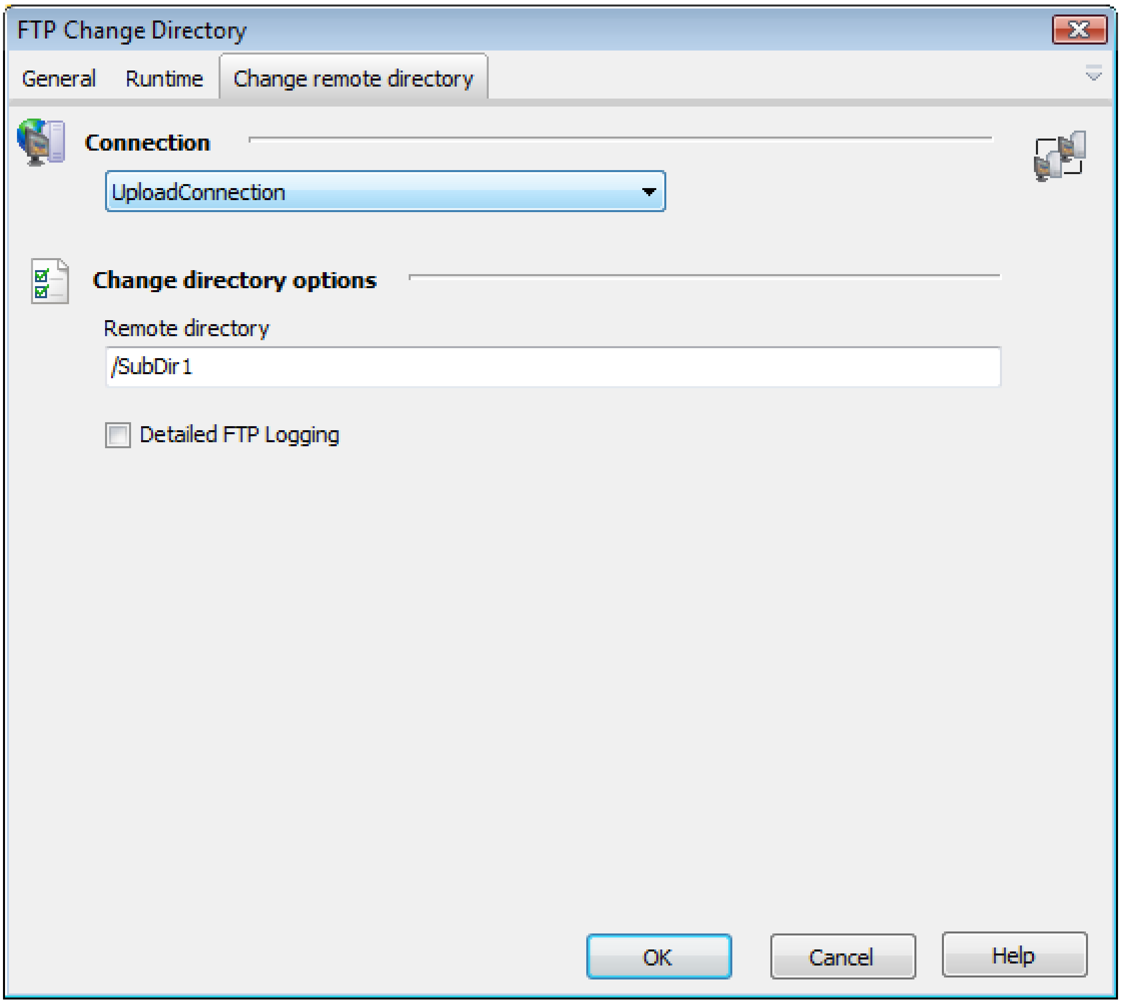
FTP Upload

See the FTP Upload topic.

FTP Change Directory

This action is used to change the remote directory on the FTP server. You can use a single directory name, eg. "wwwroot" or specify a directory hierarchy, eg. "wwwroot\Automise\downloads"

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Remote Directory - specify the name of the remote directory or directory hierarchy to change to



Connection

UploadConnection

Change directory options

Remote directory

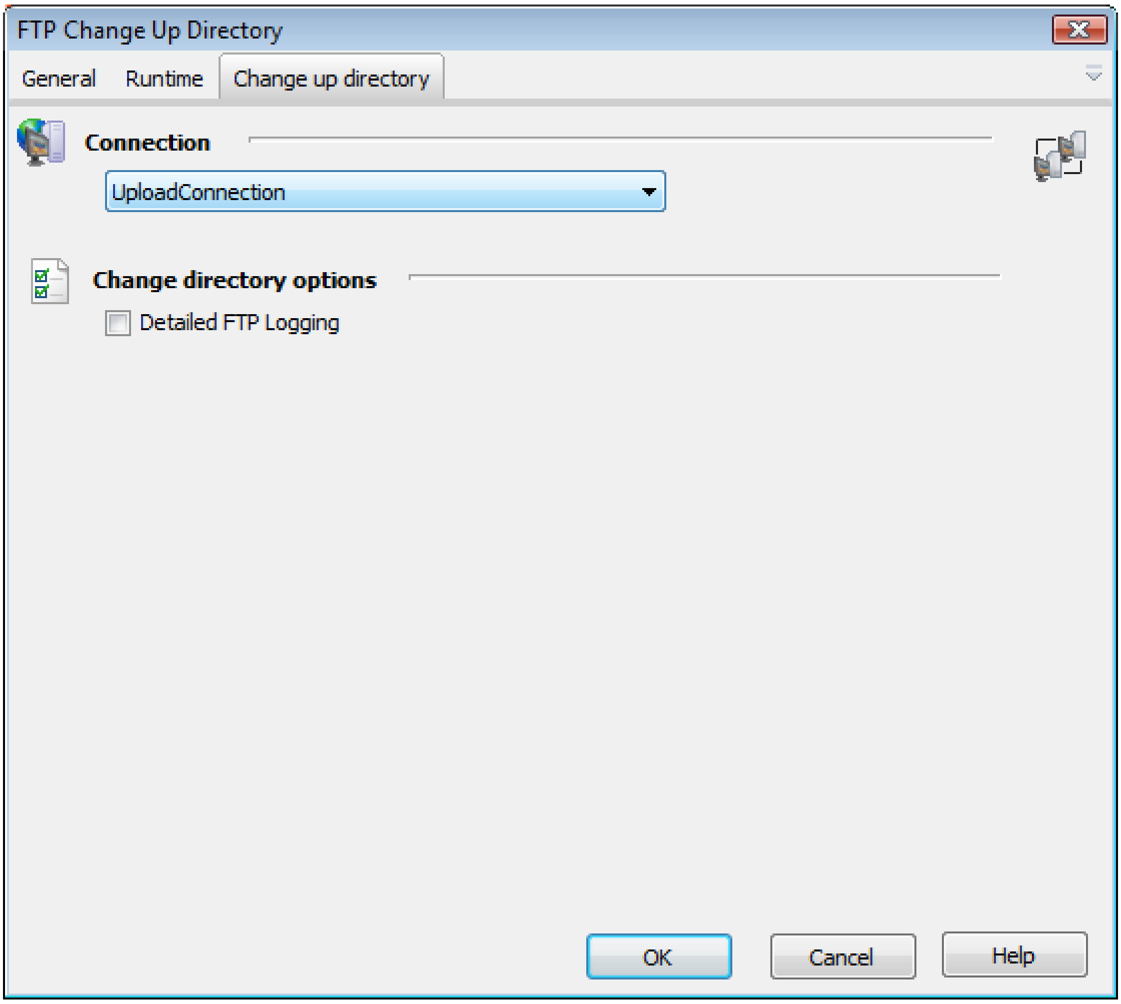
/SubDir1

□ Detailed FTP Logging

FTP Change Up Directory

This action is used to change up a directory on the FTP server. It's the equivalent of doing "cd .." in the DOS file system.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log



Connection

UploadConnection

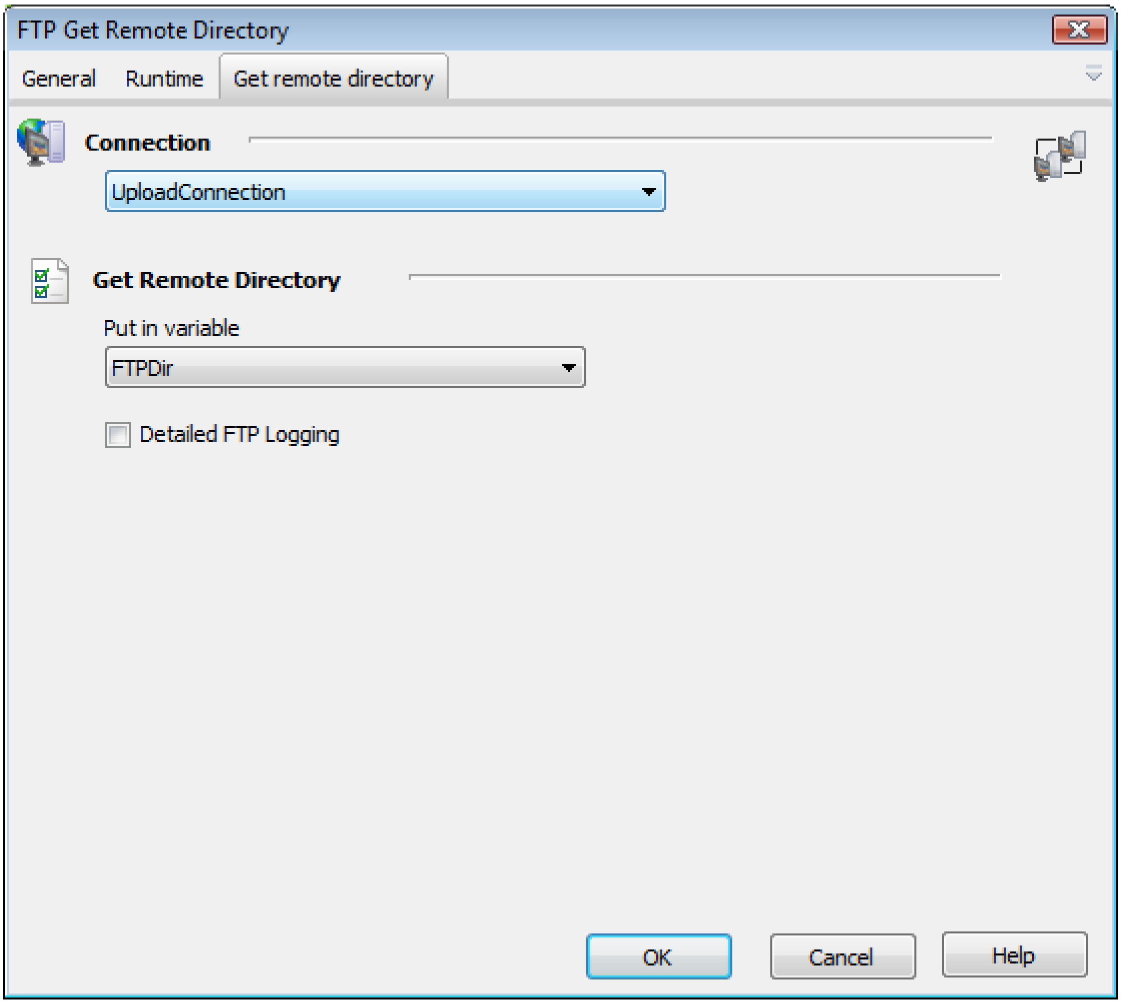
Change directory options

□ Detailed FTP Logging

FTP Get Remote Directory

This action is used to get the current directory on the FTP server. It's the equivalent of doing a "PWD" in the DOS file system. You may wish to use this action to simply log a record of the remote directory, or you can capture the remote directory into a Automise variable for use in other actions.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Put in Automise variable - specify the variable to hold the remote directory.



FTP Get Remote Directory

Connection

UploadConnection

Get Remote Directory

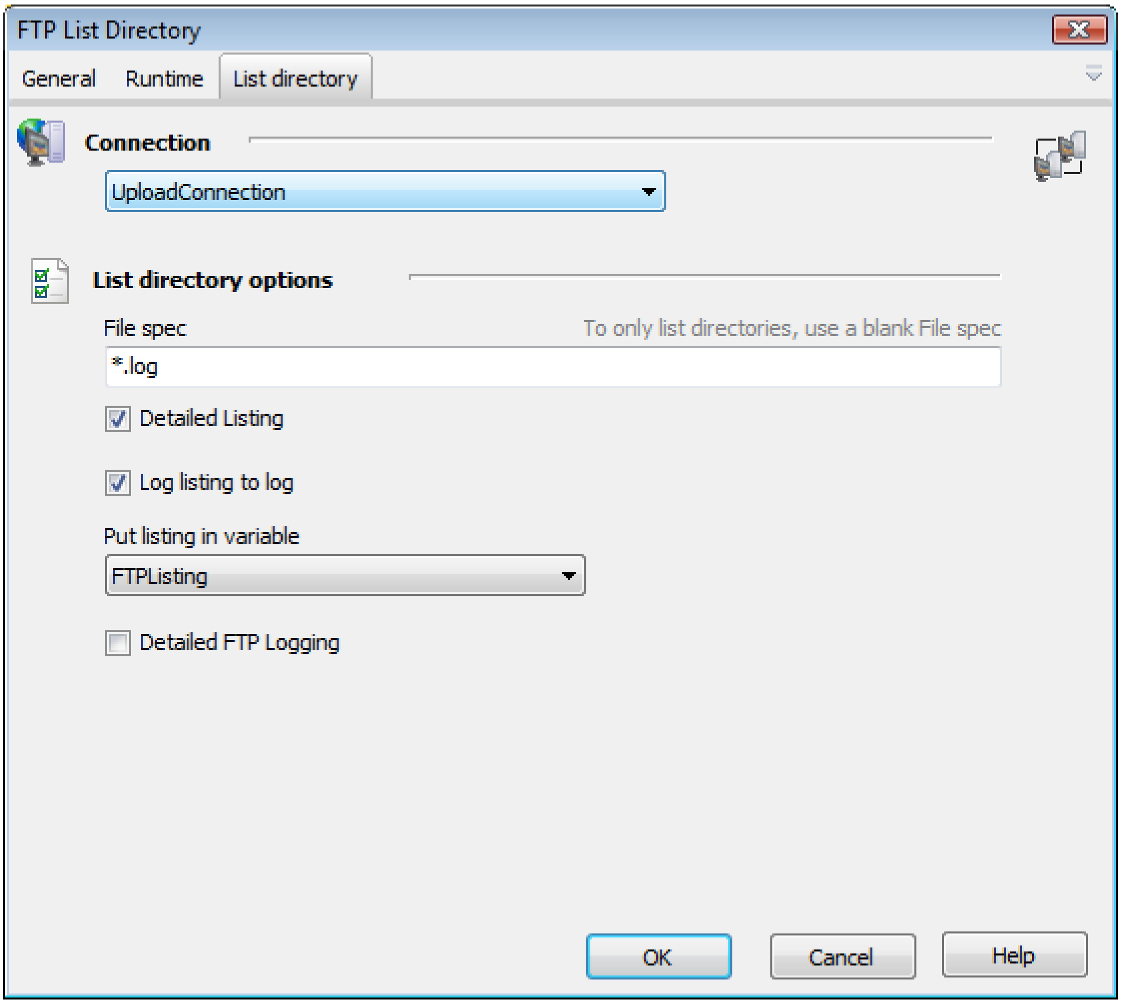
Put in variable: FTPDir

□ Detailed FTP Logging

FTP List Directory

This action is used to get a list of the files in the current remote directory of the FTP server. You could use the information gathered from this action to specify files to download using the FTP Download File action in conjunction with a List Iterator action.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- File Spec - specify the file specification to search for files, eg. *.bin
- Detailed Listing - other information such as the file size is also retrieved (the extra information is dependent on the FTP server)
- Log listing to FB Log - if this is checked, then the listing will be sent to the Automise log.
- Put listing in FB variable - if a variable is specified, then the listing of files is saved to the FB variable



FTP List Directory

Connection

UploadConnection

List directory options

File spec

To only list directories, use a blank File spec

*.log

☑ Detailed Listing

☑ Log listing to log

Put listing in variable

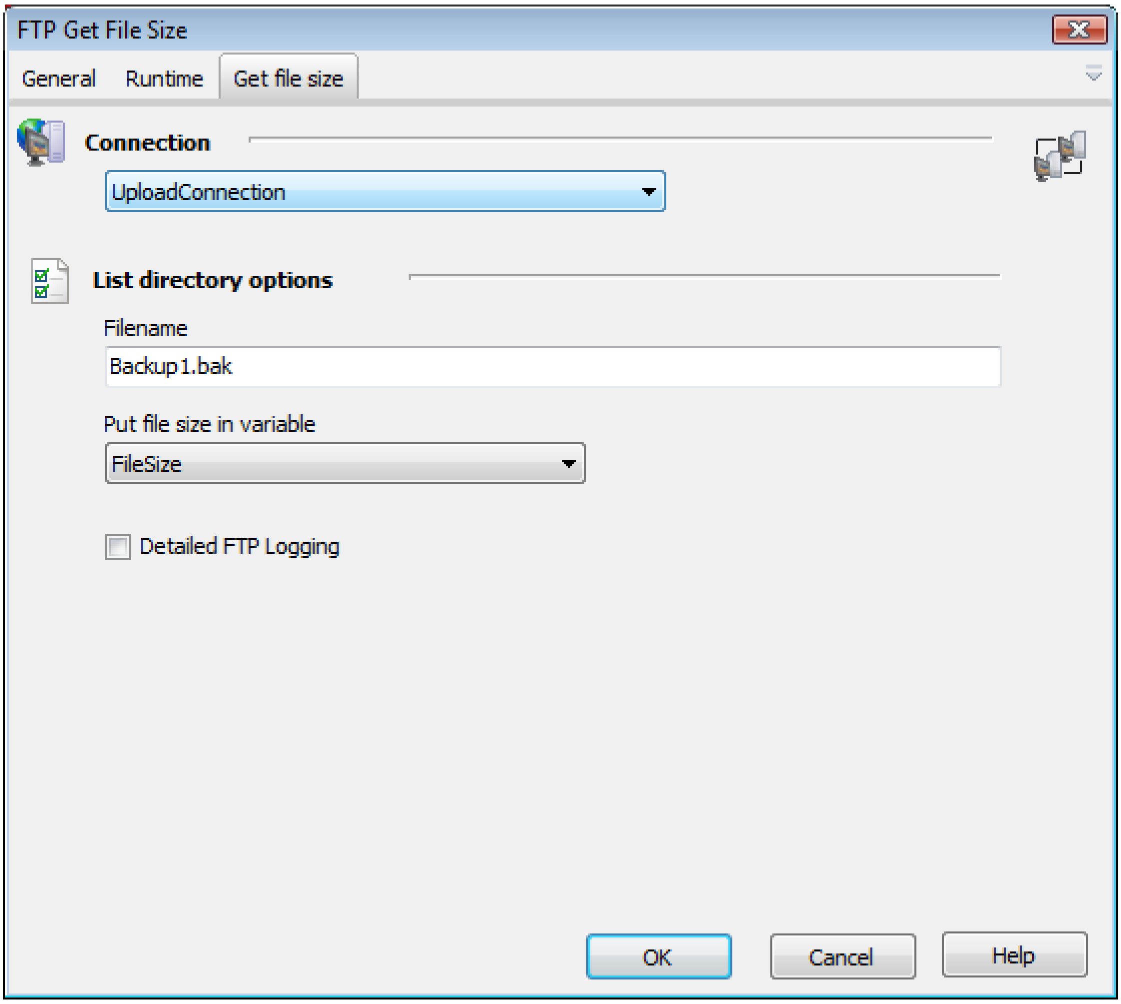
FTPListing

□ Detailed FTP Logging

FTP Get File Size

This action is used to retrieve the size of the specified file on the FTP server.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Filename - Specify the name of the file on the server to query.



Connection

UploadConnection

List directory options

Filename

Backup1.bak

Put file size in variable

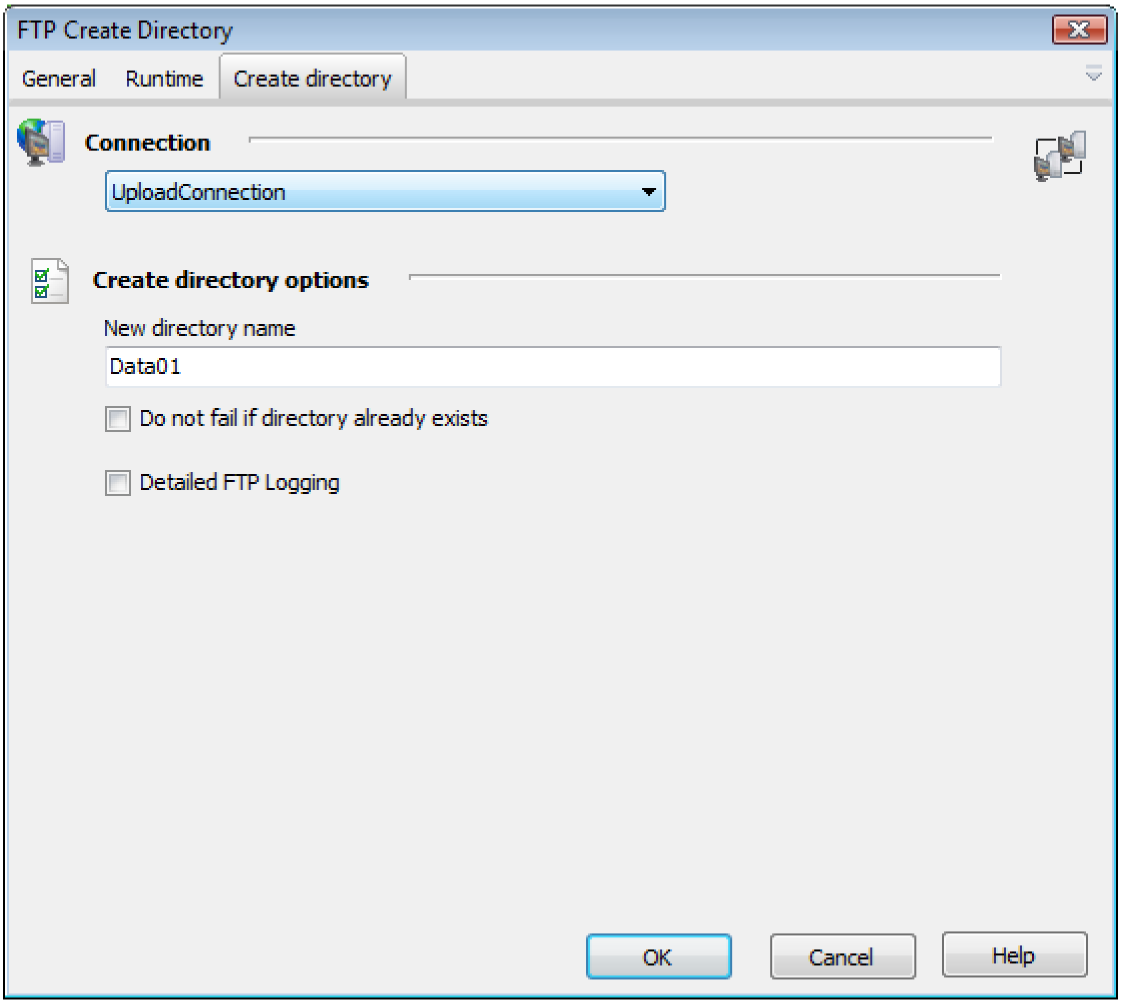
FileSize

□ Detailed FTP Logging

FTP Create Directory

This action is used to create a directory on the FTP server.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- New directory name - the name to give the new directory.
- Do not fail if directory already exists - with this checked the action will not fail if the directory already exists on the server



UploadConnection

New directory name: Data01

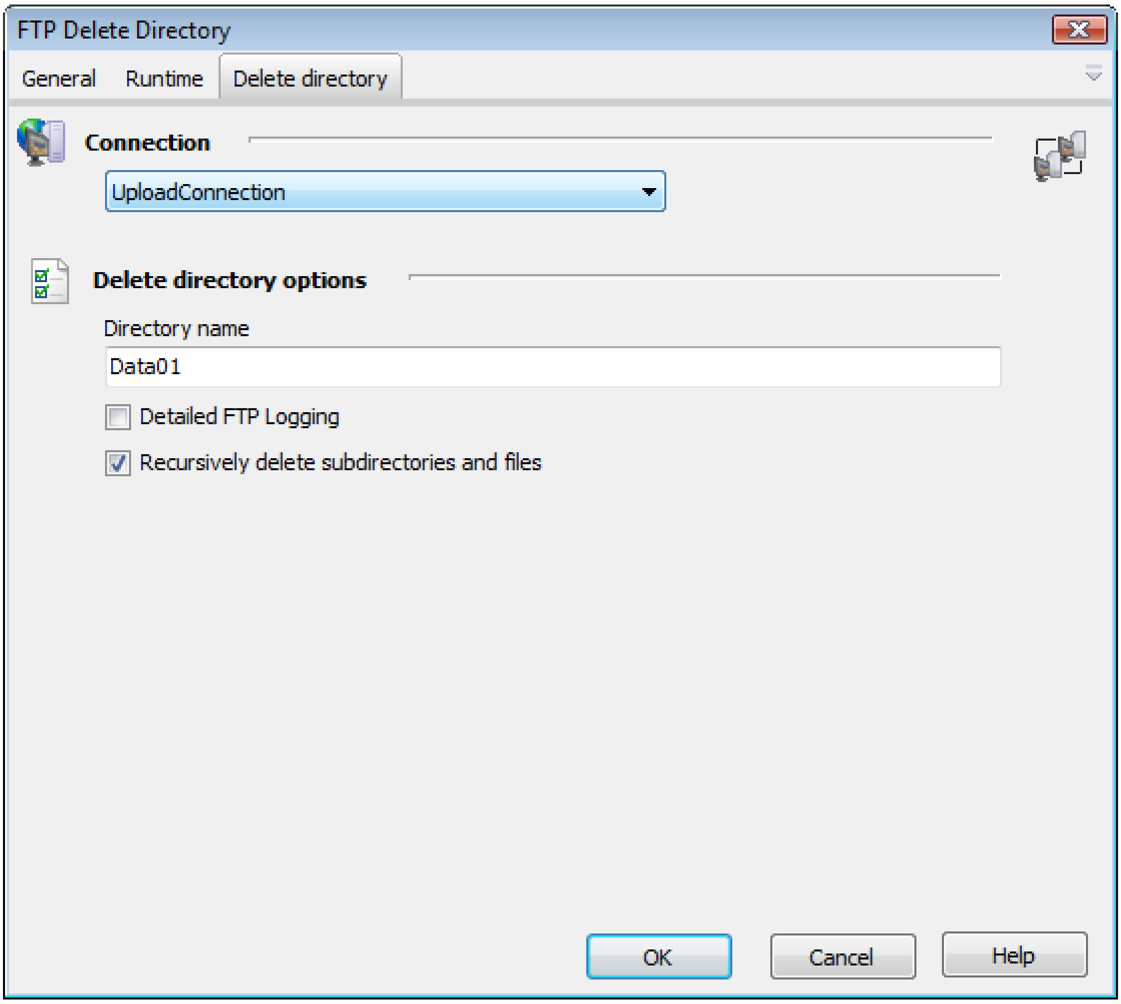
□ Do not fail if directory already exists

□ Detailed FTP Logging

FTP Delete Directory

This action is used to delete a directory on the FTP server.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Directory name - the name of the remote directory to delete.



FTP Delete Directory

Connection

UploadConnection

Delete directory options

Directory name: Data01

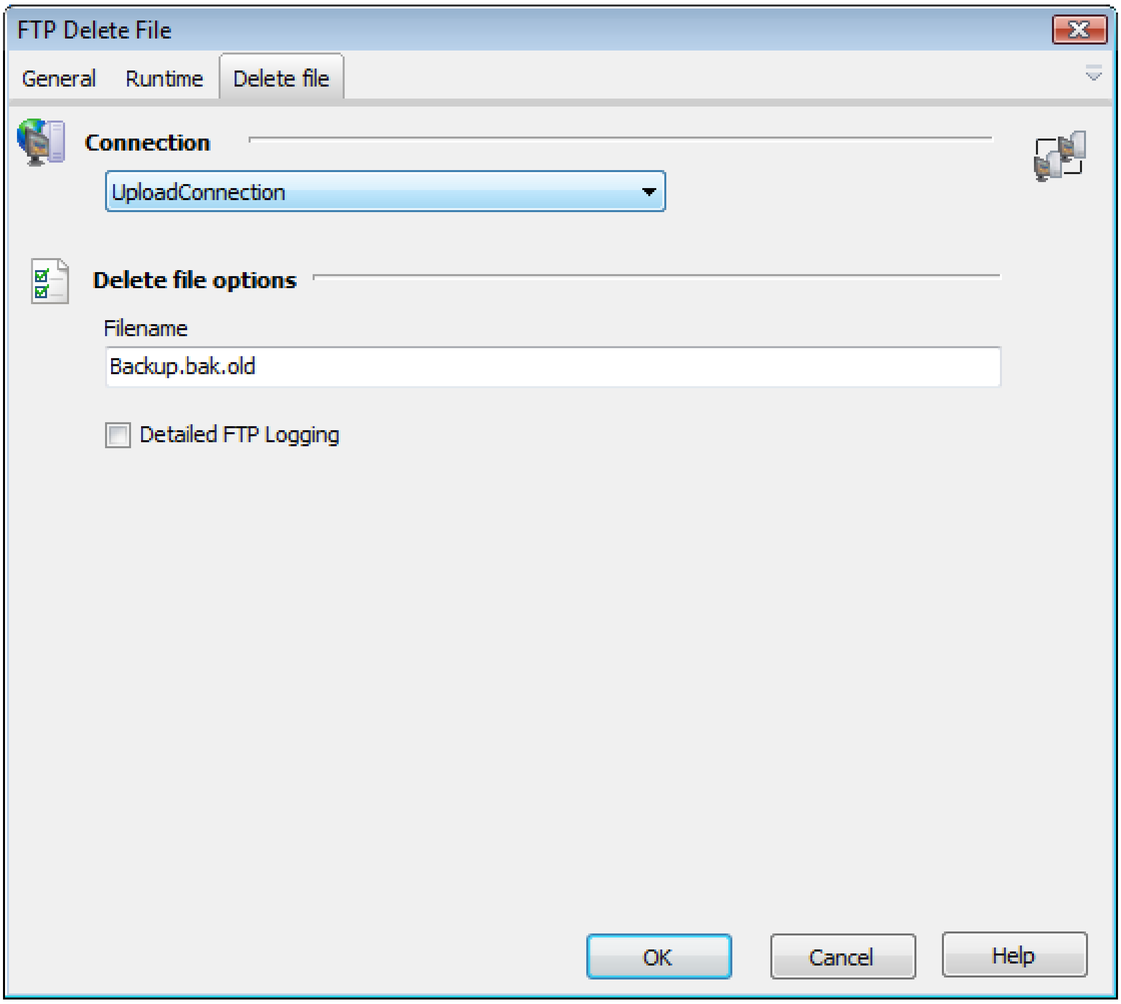
□ Detailed FTP Logging

☑ Recursively delete subdirectories and files

FTP Delete File

This action is used to delete a file on the FTP server.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Filename - the name of the remote file to delete.



Connection

UploadConnection

Delete file options

Filename

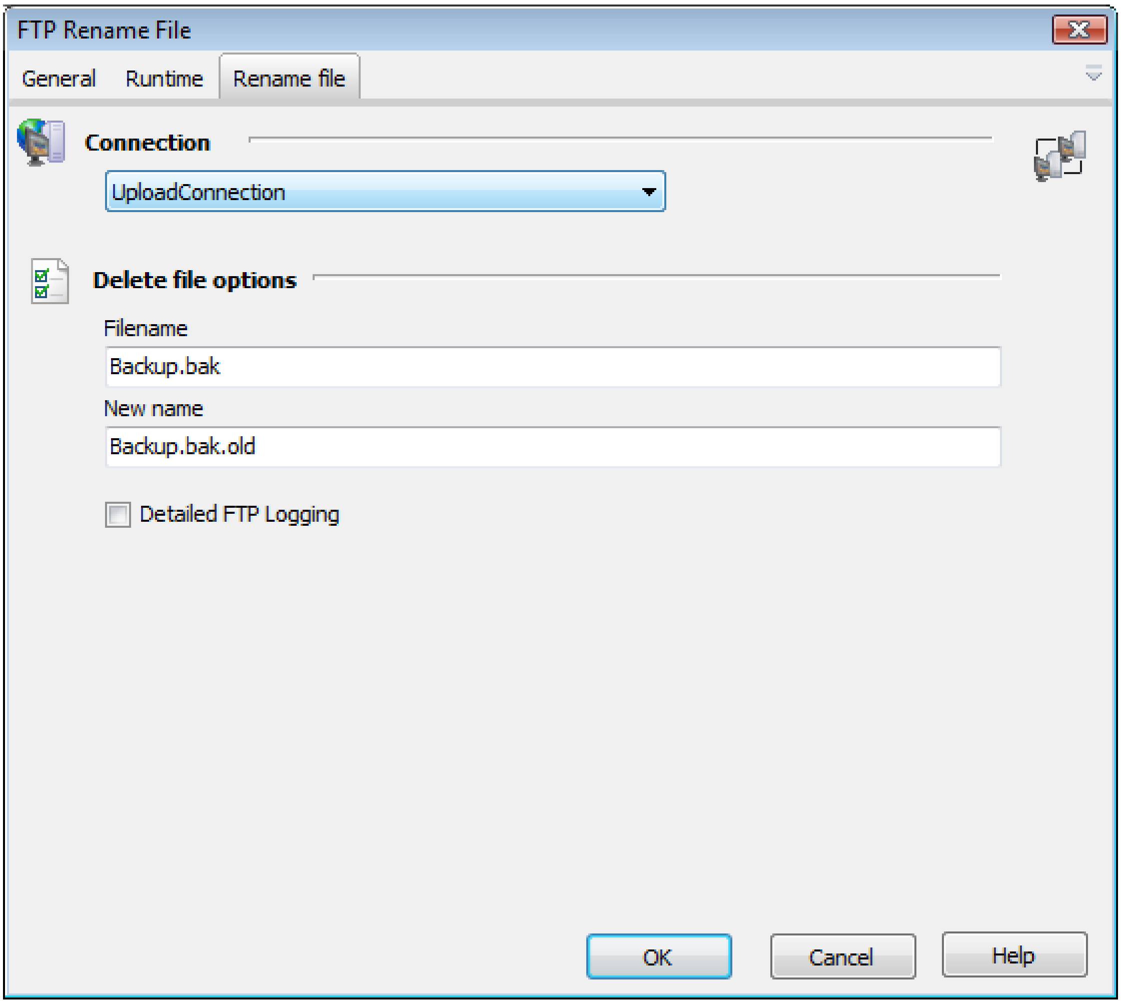
Backup.bak.old

□ Detailed FTP Logging

FTP Rename File

This action is used to rename a file on the FTP server.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log
- Filename - the name of the file to rename
- New name - the new name to give the specified file.



Connection

UploadConnection

Delete file options

Filename

Backup.bak

New name

Backup.bak.old

□ Detailed FTP Logging

FTP Wait For File

This action is similar to the other Wait For actions. See this topic for details on set the polling frequency and timeout. Note that the polling frequency for the FTP action must always be less than the server inactivity timeout, or the server will disconnect the FTP connection in between checking for the file.

The wait for file action waits for a file on the remote server. The file must be in the current directory (use the FTP Change Directory command to set the directory.)

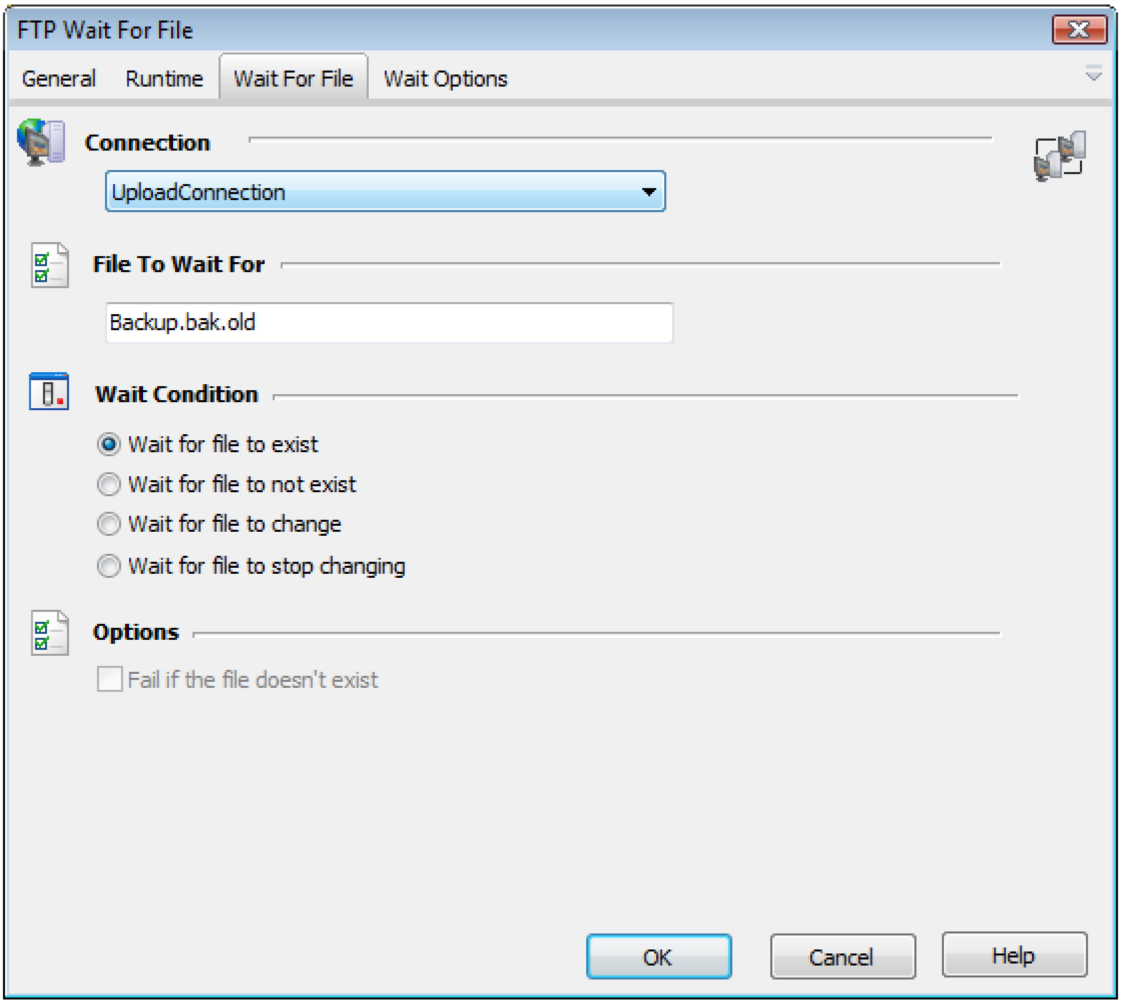
You can wait for a file to:

- Exist.
- Not exist.
- Change (this is when the file's size or date is different to the size/date when the action started.)
- Stop Changing (this is when a file exists, and its size and date do not change in between two polling events. This is useful for telling when a file upload has finished.)

For Wait For Exist/Not Exist, you can use file wildcards (ie *.*) to check for groups of files.

In the case of Wait For Change/Stop Changing, you can set the action to fail if the file does not exist. Otherwise, the action will wait until the file exists and - in the case of

Stop Changing - has stopped resizing.



Connection

UploadConnection

File To Wait For

Backup.bak.old

Wait Condition

● Wait for file to exist

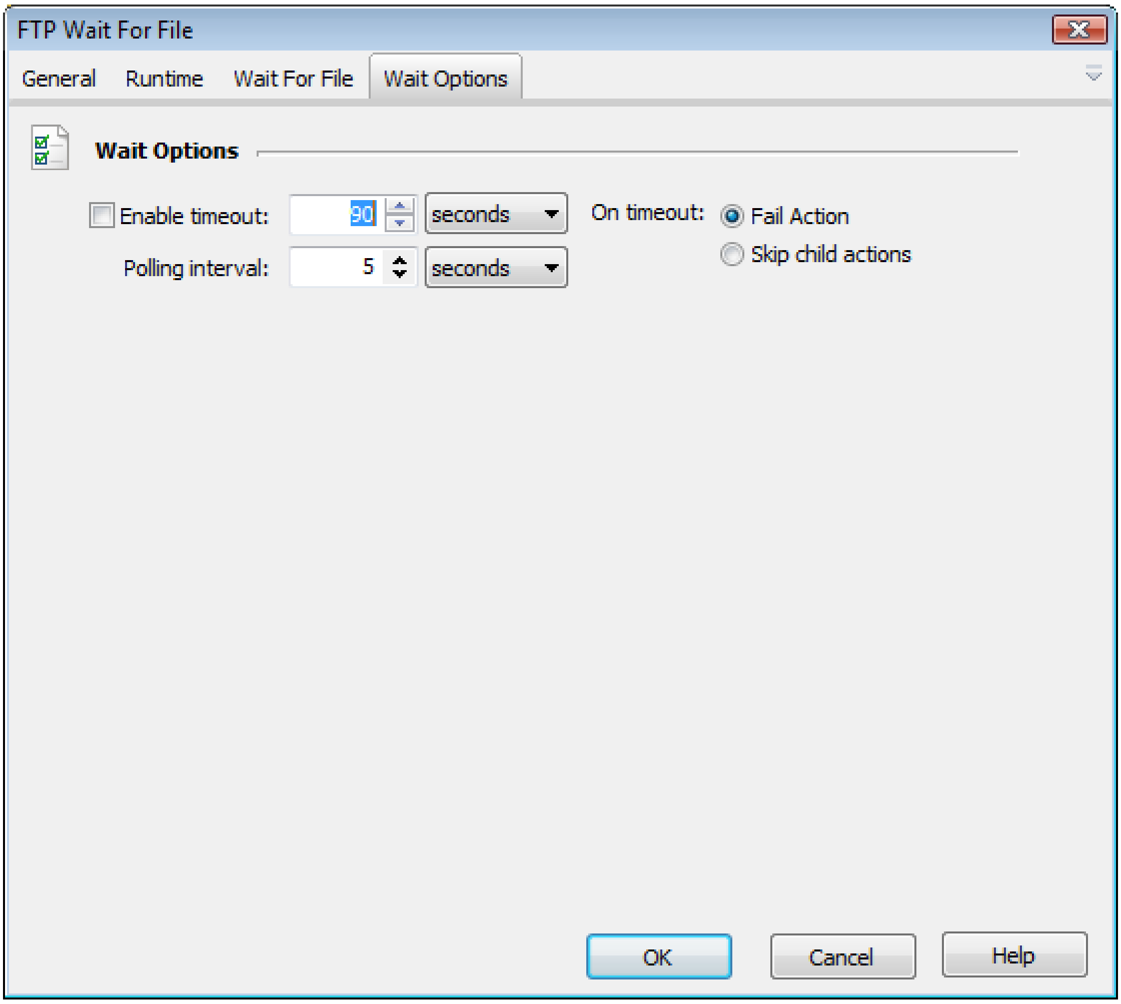
○ Wait for file to not exist

○ Wait for file to change

○ Wait for file to stop changing

Options

□ Fail if the file doesn't exist



□ Enable timeout: 90 seconds

Polling interval: 5 seconds

On timeout: ● Fail Action ○ Skip child actions

FTP No-op

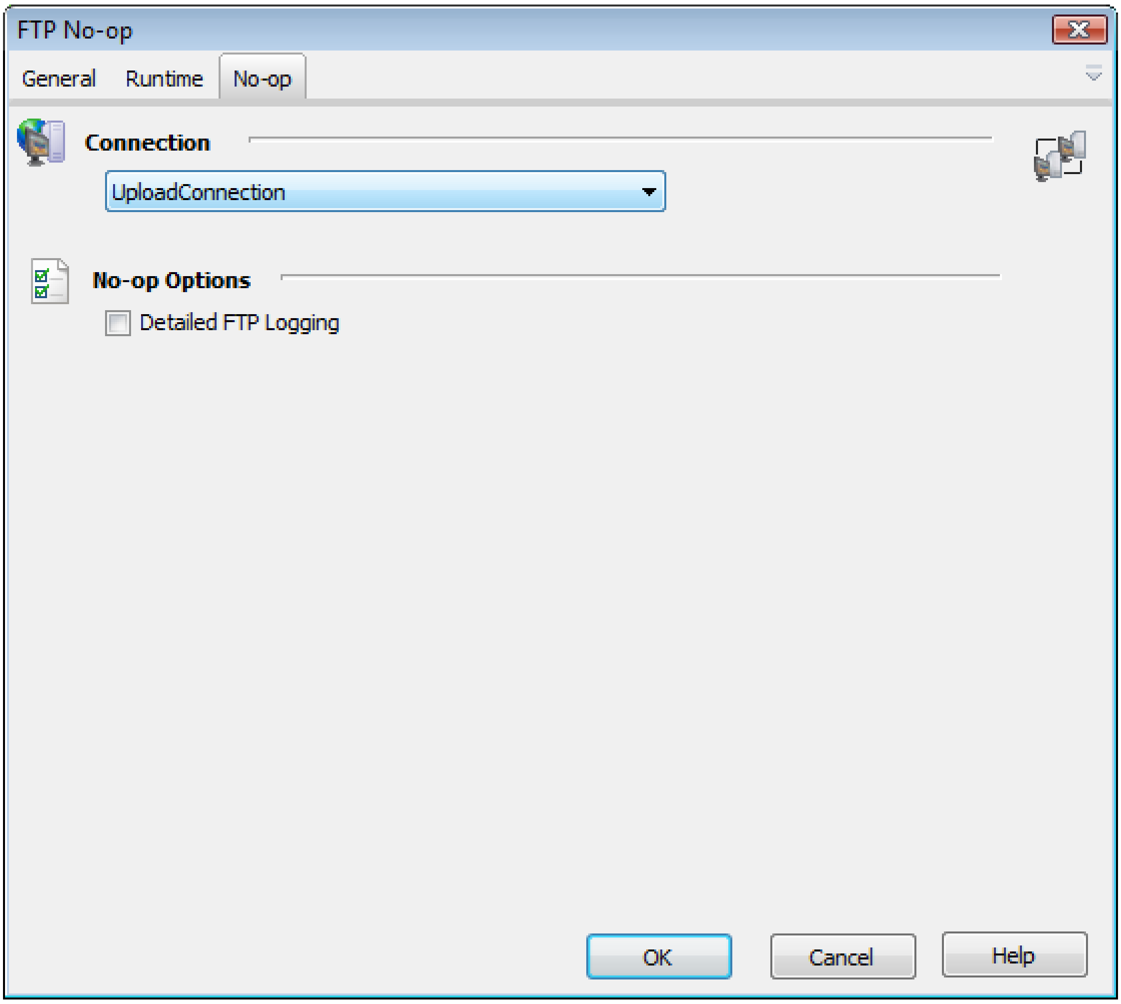
This action is used to keep the connection alive to the FTP server. If you connect to the FTP server at the start of the project, you may want to use the FTP No-op at certain points during your build so that the connection is maintained. Many FTP servers will close the connection after a certain amount of time with no activity.

- Connection Name - select the connection (see FTP Connect action)
- Detailed FTP Logging - Outputs the raw FTP commands and log messages to the log

FTP Disconnect

This action is used to disconnect the specified FTP connection. This action should be called after all FTP operations have completed so that the connection is not left open when the build ends.

- Connection Name - select the connection (see FTP Connect action)



6.15.2.1 FTP Mirror

FTP Mirror Action

The FTP Mirror action supports two operations:

- Mirroring files and directories from the local file system to an FTP Server

- Mirroring files and directories from an FTP Server to the local file system

This action also allows you to specify whether files that exist in both source and destination should be overwritten in the destination based on the following rules:

- File CRC32 hash does not match (only available where FTP Server supports CRC Hashing)

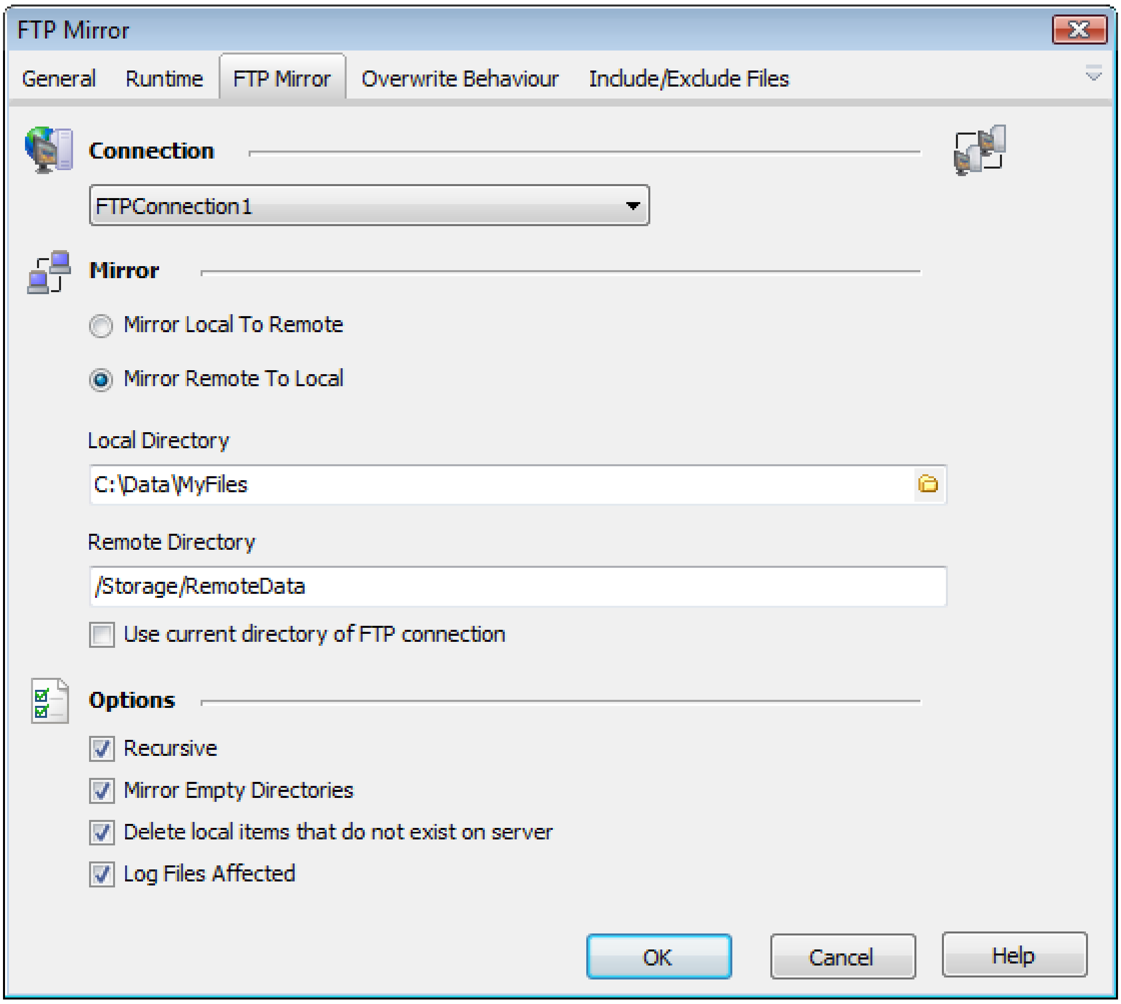
- File size of source and destination file does not match

- Modified date of source file is newer than destination file

Specify which files should be included in the mirror operation based on file extension. Files can be excluded by name, file mask or regular expression matching.

FTP Mirror

The FTP Mirror tab provides the basic settings for the FTP Mirror operation that are required before you can run the action.



FTP Mirror

Connection

FTPConnection1

Mirror

○ Mirror Local To Remote

● Mirror Remote To Local

Local Directory: C:\Data\MyFiles

Remote Directory: /Storage/RemoteData

□ Use current directory of FTP connection

Options

☑ Recursive

☑ Mirror Empty Directories

☑ Delete local items that do not exist on server

☑ Log Files Affected

Connection - The FTP Mirror action requires you to create an FTP connection using the FTP Connect action. This determines which FTP connection the action should use.

Mirror Local To Remote - Choose this option if you want to mirror your local files to your FTP Server.

Mirror Remote To Local - Choose this option if you want to mirror your remote files to your local file system.

Local Directory - Provide the local directory that will be used as the base directory for local files. This directory itself will not be created on the FTP Server if mirroring local to remote, the files within this directory will be mirrored to remote directory specified.

Remote Directory - Provide the remote directory that will be used as the base directory for remote files. This directory itself will not be created locally when mirroring remote to local, the files within this directory will be mirrored to the local directory specified.

Use current directory of FTP connection - Rather than specify a remote directory, you can enable this option which will use the current directory for the specified FTP connection. You can use the FTP Change Directory action to set the current directory for the FTP connection.

Recursive - Enabling this options allows the mirror action to include sub-folders and the files within to be included in the mirror operation, otherwise only the files in the base directories will be mirrored.

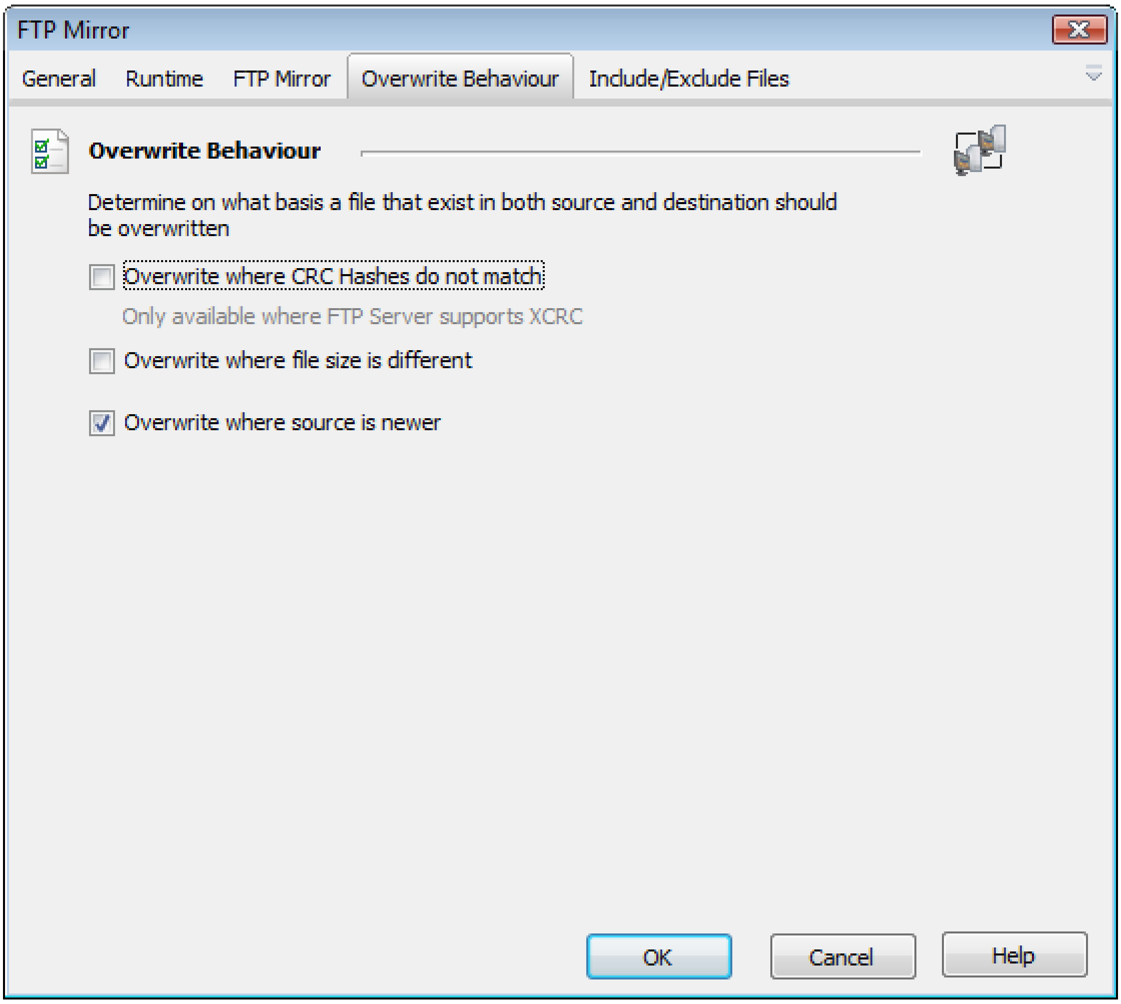
Mirror Empty Directories - Enabling this option allows directories that are empty on the source end to be recreated in the destination, by default empty directories are not re-created.

Delete items on server that do not exist locally/Delete local items that do not exist on server - Enable this option if you wish to create an exact mirror where files and directories that exist at your destination that do not exist at your source will be deleted. **Use this option with care** as it can permanently remove data. For instance mirroring an empty source directory to a non-empty destination with this option enabled would delete all the files and directories in the destination directory.

Log Files Affected - Enable this option to write a list of mirrored files to the log.

Overwrite Behaviour

The Overwrite Behaviour tab provides options to specify whether files that exist in both the source and destination should be overwritten. Without enabling any of these options, existing files will never be overwritten.



Determine on what basis a file that exist in both source and destination should be overwritten

☐ Overwrite where CRC Hashes do not match

Only available where FTP Server supports XCRC

☐ Overwrite where file size is different

☑ Overwrite where source is newer

Overwrite where CRC Hashes do not match - Enable this option to compare file CRC32 hashes when the file exists in both the source and the destination. If the hash values do not match then the destination file will be replaced with the source file.

Note: This option will only work where the FTP Server being used supports CRC checks for files.

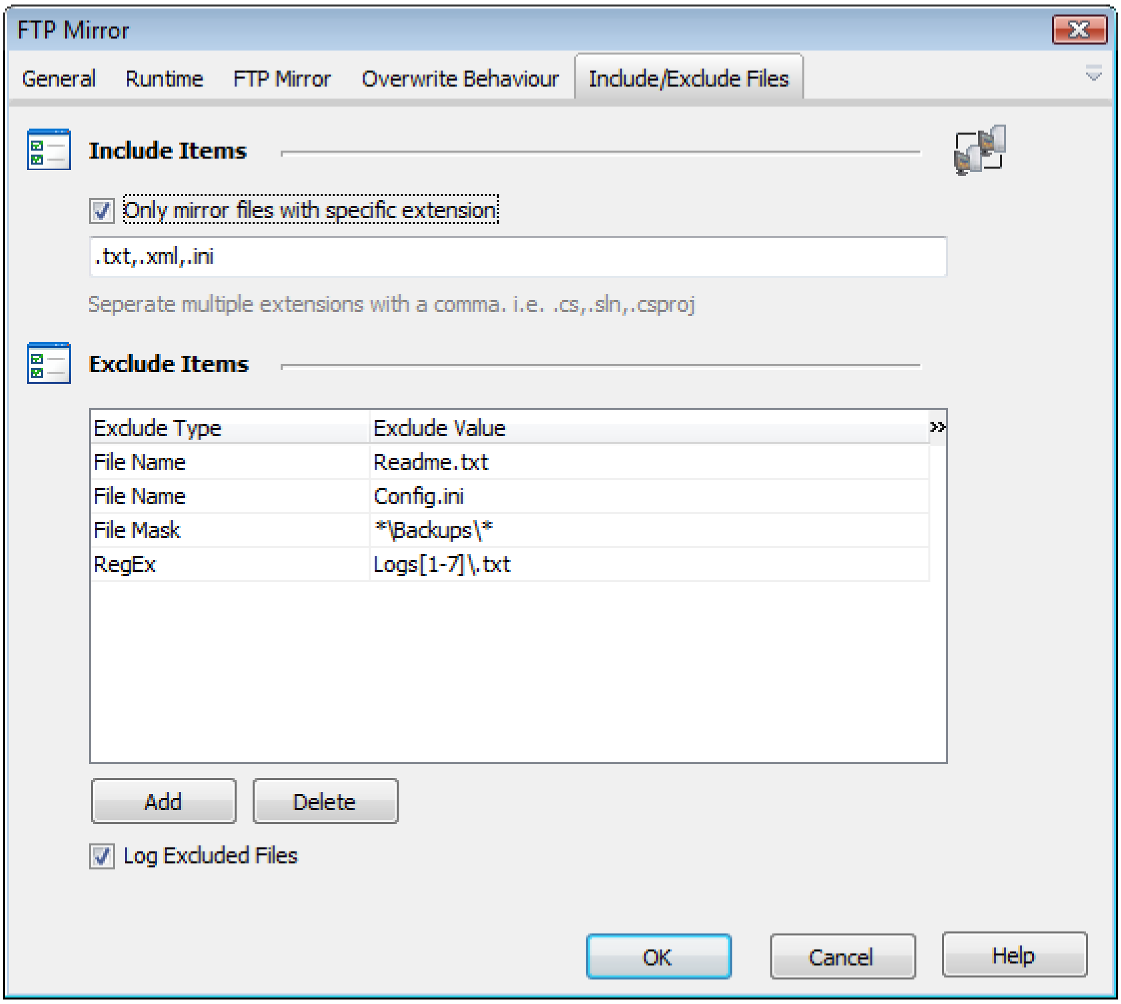
Overwrite where file size is different - Enable this option to compare the file size when the file exists in both the source and the destination. If the size values do not match then the destination file will be replaced with the source file.

Overwrite where source is newer - Enable this option to compare the file date when the file exists in both the source and the destination. If the source file is newer then the destination file will be replaced with the source file.

Note: This action automatically calculates the time difference between the FTP Server and the local machine and compensates for this when comparing the file dates.

Include/Exclude Files

The Include/Exclude Files tab allows you to determine which files are included and excluded for the mirror operation.



☑ Only mirror files with specific extension

.txt,.xml,.ini

Seperate multiple extensions with a comma. i.e. .cs,.sln,.csproj

Exclude Type	Exclude Value
File Name	Readme.txt
File Name	Config.ini
File Mask	*\Backups*
RegEx	Logs[1-7]\.txt

☑ Log Excluded Files

Include Items

Enabling the **Only mirror files with specific extension** option allows you to select which files are included/excluded for the mirror operation. This allows you to pass a comma-delimited list of file extensions. This list can only contain file extensions, not file names or wild cards.

The following Include list is valid:

.txt,.xml,.ini

The following Include list is invalid:

MyFile.txt,.xml,\Settings*.ini*

When using this option only files that match the specified extension(s) will be considered, this means that files with this extension will be mirrored.

Exclude Items

The Exclude Items section allows you to specify items to be excluded from the mirror operation. You can use as many exclusion rules as required. Files can be excluded on the following basis:

File Name - Exclude any files that match the file name specified. This value cannot include any wildcard characters, the value must be the file name and extension of the file that you want to exclude.

Example: *Readme.txt* - this will exclude any file named Readme.txt

File Mask - Exclude any files that match the file mask specified. This value can contain wildcard characters (i.e. * and ?) to exclude files and directories.

Example: *\\Backups\* - this will exclude all files and directories that are within the Backups directory.

Regex - Exclude any files that match the regular expression specified.

Example: *Logs[1-7]\.txt* - this would exclude the any file named Logs where the number between *Logs* and *.txt* is in the range 1-7.

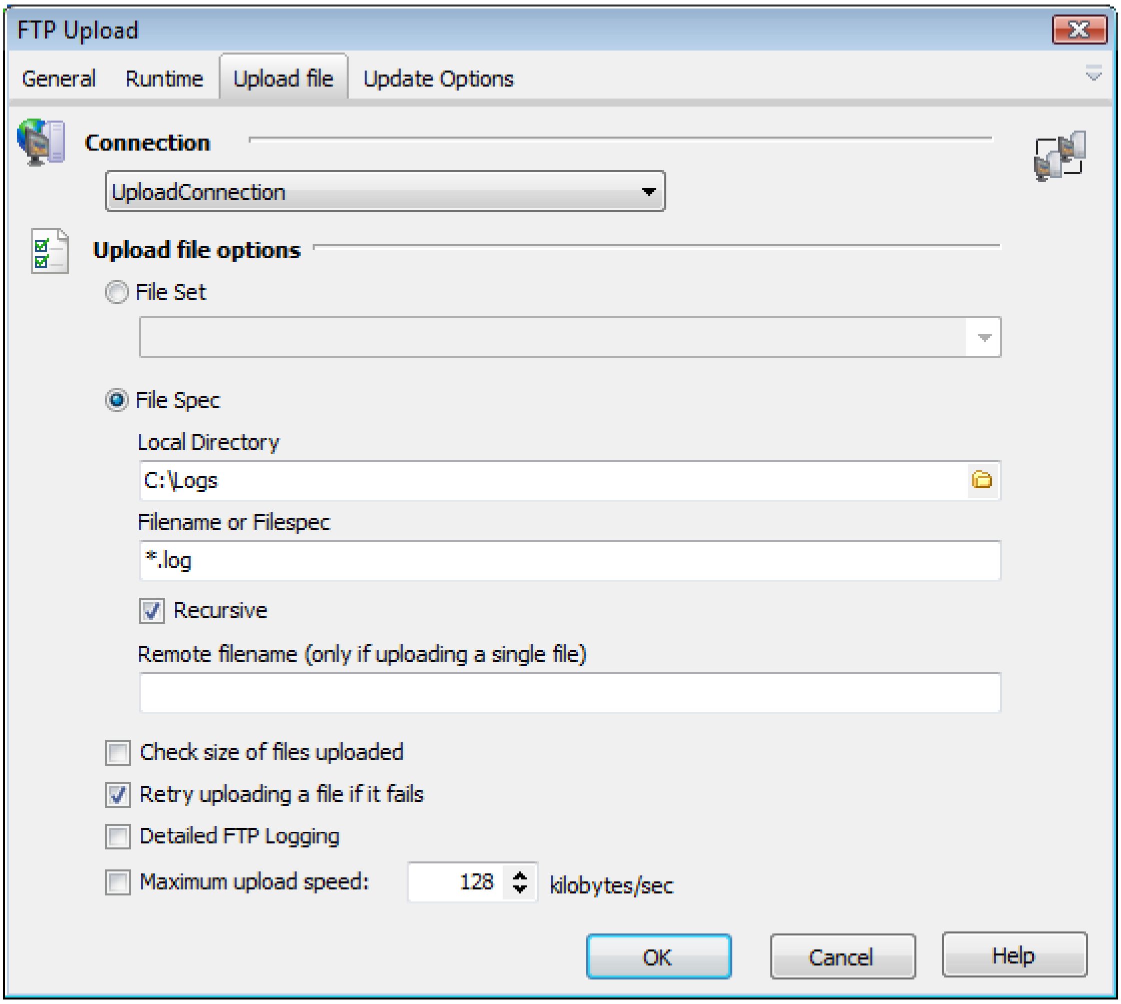
See the Regular Expression reference for more on using regular expressions.

Log Excluded Files - Enable this option to log the list of files that have been excluded based on the exclude list filters.

Note that the Exclude list overrides the Include list, so if you set your include filter to only include text files (.txt) and you add an exclude filter to exclude any files called 'Readme.txt' then all files called 'Readme.txt' will be excluded from the mirror operation.

6.15.2.2 FTP Upload

This action is used to upload one or more files using the chosen connection. Before running this action, you may want to use a FTP Change Directory action to change the remote directory.



Connection

UploadConnection

Upload file options

○ File Set

● File Spec

Local Directory

C:\Logs

Filename or Filespec

*.log

☑ Recursive

Remote filename (only if uploading a single file)

□ Check size of files uploaded

☑ Retry uploading a file if it fails

□ Detailed FTP Logging

□ Maximum upload speed: 128 kilobytes/sec

Connection Name

Select the connection (see FTP Connect action.)

Local Directory

Specify the local directory where the files which you want to upload are located.

Filename or Filespec

If you are uploading a single file, then enter the filename here (without directory, as that is specified with Local Directory). If you want to upload multiple files, then enter a filespec, eg. *.zip, or MyProject.*

Recursive

If you specify a filespec, then setting the recursive option will find any matching files in any subdirectories of the Local Directory. The files will be placed into the correct subdirectories on the FTP server. Subdirectories on the server will be created if required.

Remote Filename

If you are uploading a single file, then you can change the name of the uploaded file on the server.

Check size of files uploaded

This option will attempt to check the size of the file uploaded against the size of the original file and fail if the size differs. This option doesn't work with all FTP servers.

Detailed FTP Logging

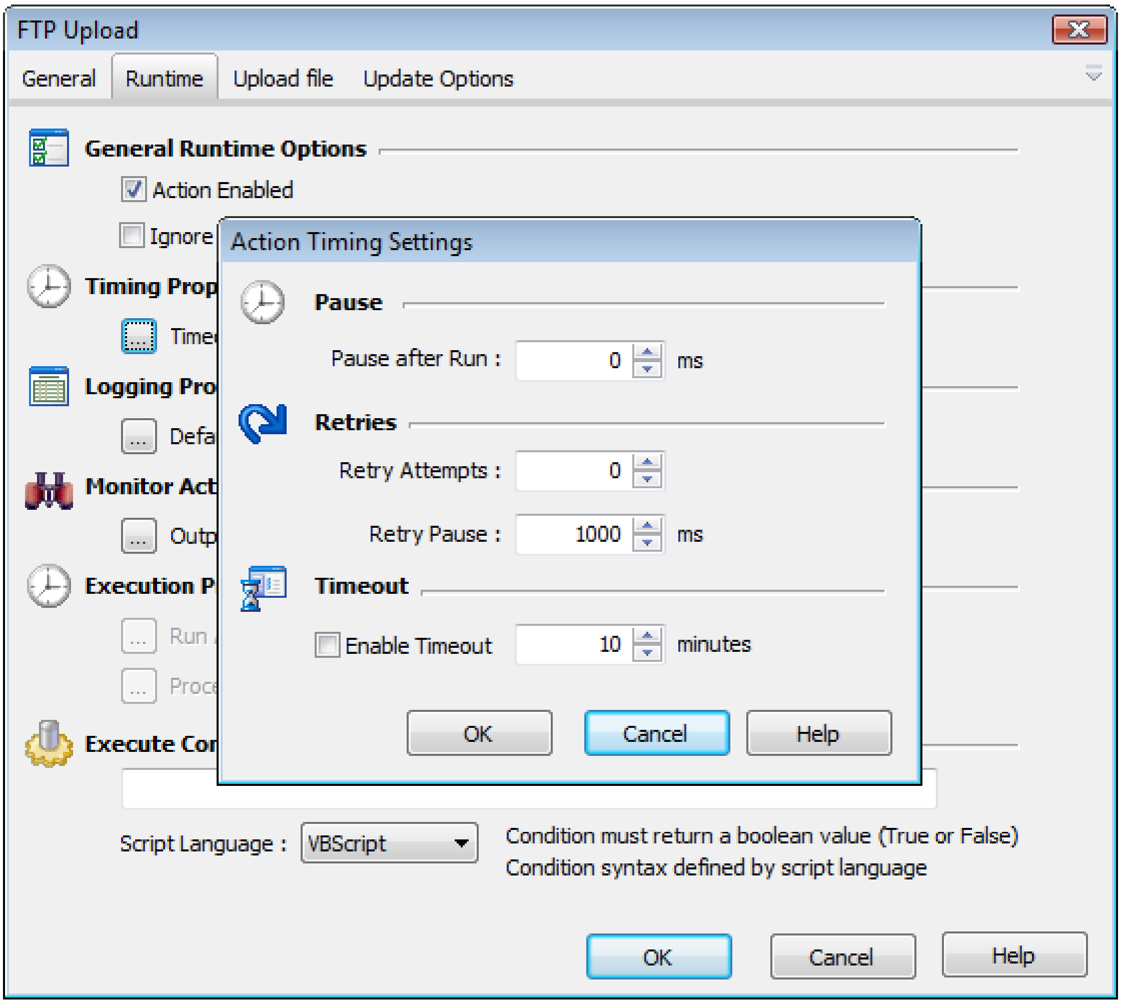
Outputs the raw FTP commands and log messages to the log.

Maximum Upload Speed

You can choose a maximum speed in (in kilobytes per second) to throttle the upload. This allows you to conserve bandwidth for other applications.

Upload Timeout

You can set a maximum time for the upload to complete in:



FTP Upload

General Runtime Options

☑ Action Enabled

□ Ignore

Action Timing Settings

Pause

Pause after Run : 0 ms

Retries

Retry Attempts : 0

Retry Pause : 1000 ms

Timeout

□ Enable Timeout 10 minutes

Script Language : VBScript

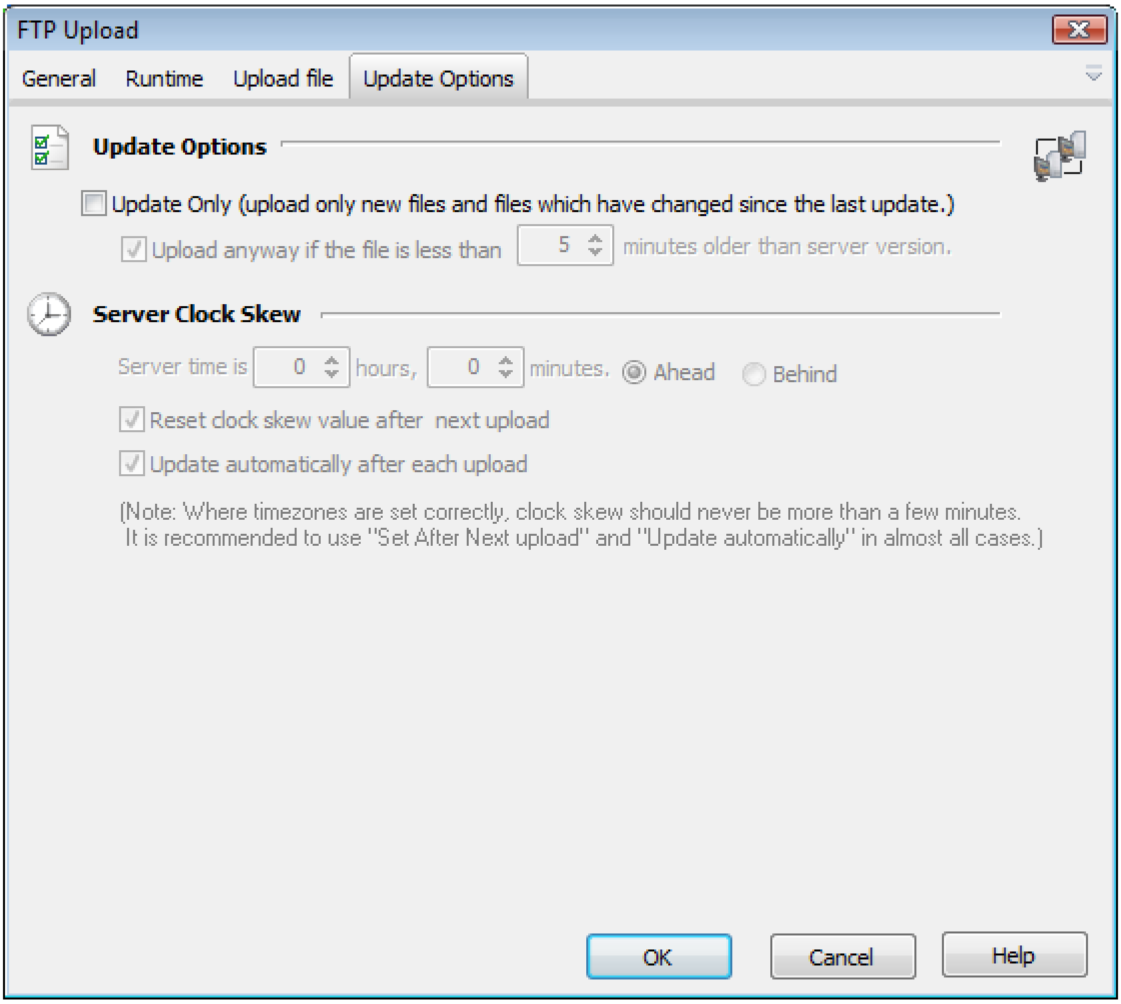
Condition must return a boolean value (True or False)

Condition syntax defined by script language

The upload will be aborted, and the action will fail, if the transfer has not completed within this timeframe.

Update Options

The update options allow you to conserve bandwidth and time by only uploading files which have changed since the last upload:



Update Options

□ Update Only (upload only new files and files which have changed since the last update.)

☑ Upload anyway if the file is less than 5 minutes older than server version.

Server Clock Skew

Server time is 0 hours, 0 minutes. ● Ahead ○ Behind

☑ Reset clock skew value after next upload

☑ Update automatically after each upload

(Note: Where timezones are set correctly, clock skew should never be more than a few minutes.
It is recommended to use "Set After Next upload" and "Update automatically" in almost all cases.)

Update Only

If you check this box, only files which differ from the versions on the server will be uploaded.

Local files will be uploaded if they:

- Do not exist on the server,
- Are a different size to the copy of the file on the server, or
- Are newer than the copy of the file on the server.

Note that not all FTP Servers support this option. The FTP Server must be able to return reliable time/date and file size information in order to use the Update Only feature.

... Upload anyway if the file is less than...

This option allows you to "feather" the comparison times to ensure false-negatives do not occur. It is recommended this option is enabled to account for clock drift over time. For extra reliability, set it to a high value (eg 120 minutes.)

Server Clock Skew

The Server Clock Skew options allow you to compensate for differences between the timestamps reported by the server and the local date/time (for example, the local computer may think the time is 1430 UTC, while the server may report 1436 UTC.)

In cases where timezones are set incorrectly, or the FTP server does not use UTC, clock skew may be several hours.

It is generally recommended that you do not edit the clock skew value manually, but rely on the **"Reset clock skew value after next upload"** and "**Update automatically after each upload**" options to set it automatically.

Reset clock skew value after next upload

If this option is checked, the next file that the action uploads will never be skipped, even if it has not changed. The timestamp of that file will be used to set the clock skew value. This only happens once (the option is turned off after the clock skew value has been set.)

It is recommended that this option is set the first time the action is used, and any time there is a possibility that the FTP Server or local computer's timezone information might have changed.

Update automatically after each upload

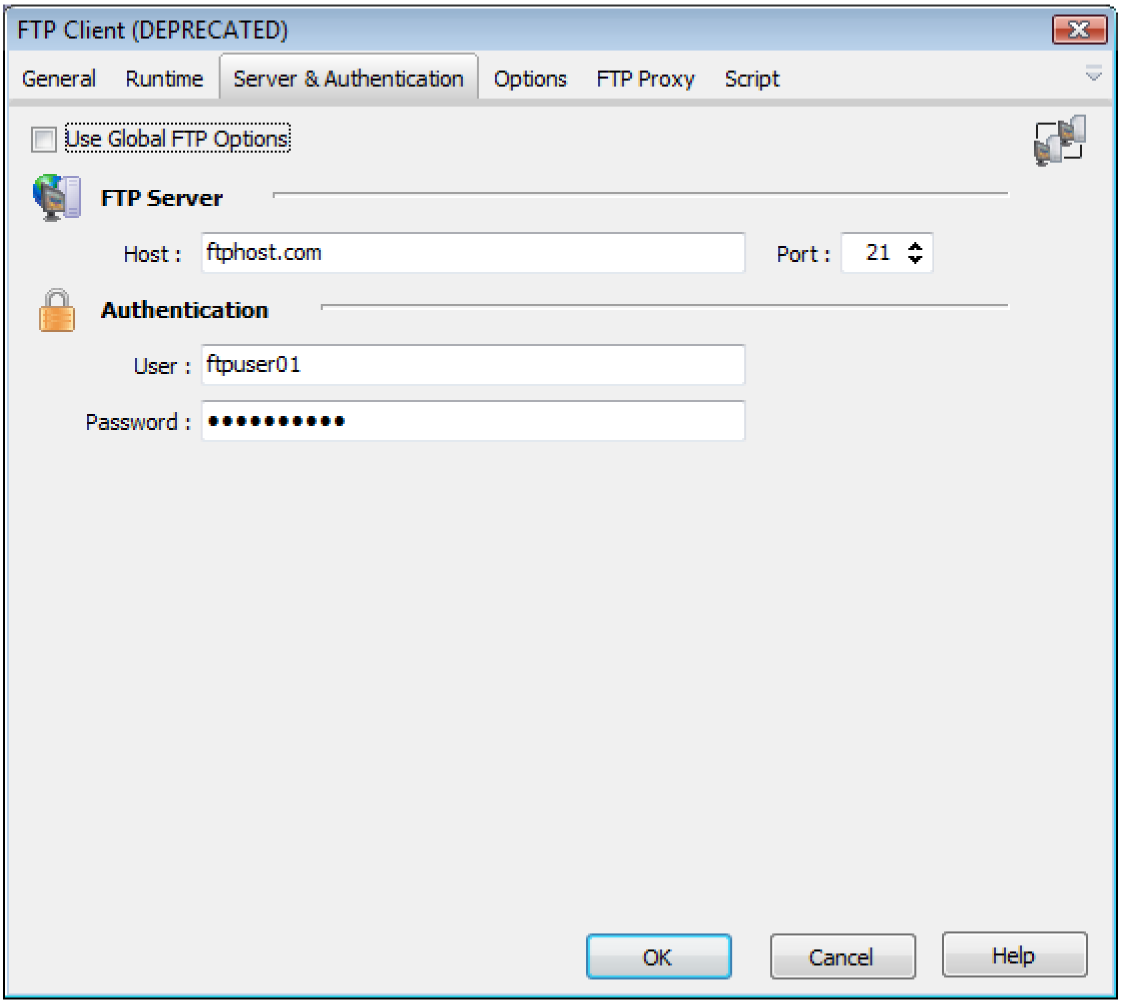
If this option is checked, then each time a file is updated its timestamp is used to reset the clock skew value if it has changed. This helps overcome "clock drift" over time.

6.15.3 FTP Client (deprecated)

This action has been deprecated, but is still available for backwards compatibility.

Please see the new set of FTP Actions.

This action provides a basic FTP client, which can be used to upload or download files to/from an FTP server.



□ Use Global FTP Options

FTP Server

Host : ftphost.com

Port : 21

Authentication

User : ftpuser01

Password : ●●●●●●●●●●●

Host : The host name or IP address of the ftp server

User : Your user id on the ftp server

Password : Your password on the ftp server

Port : The port on which the FTP server (or proxy if you are using one) is listening

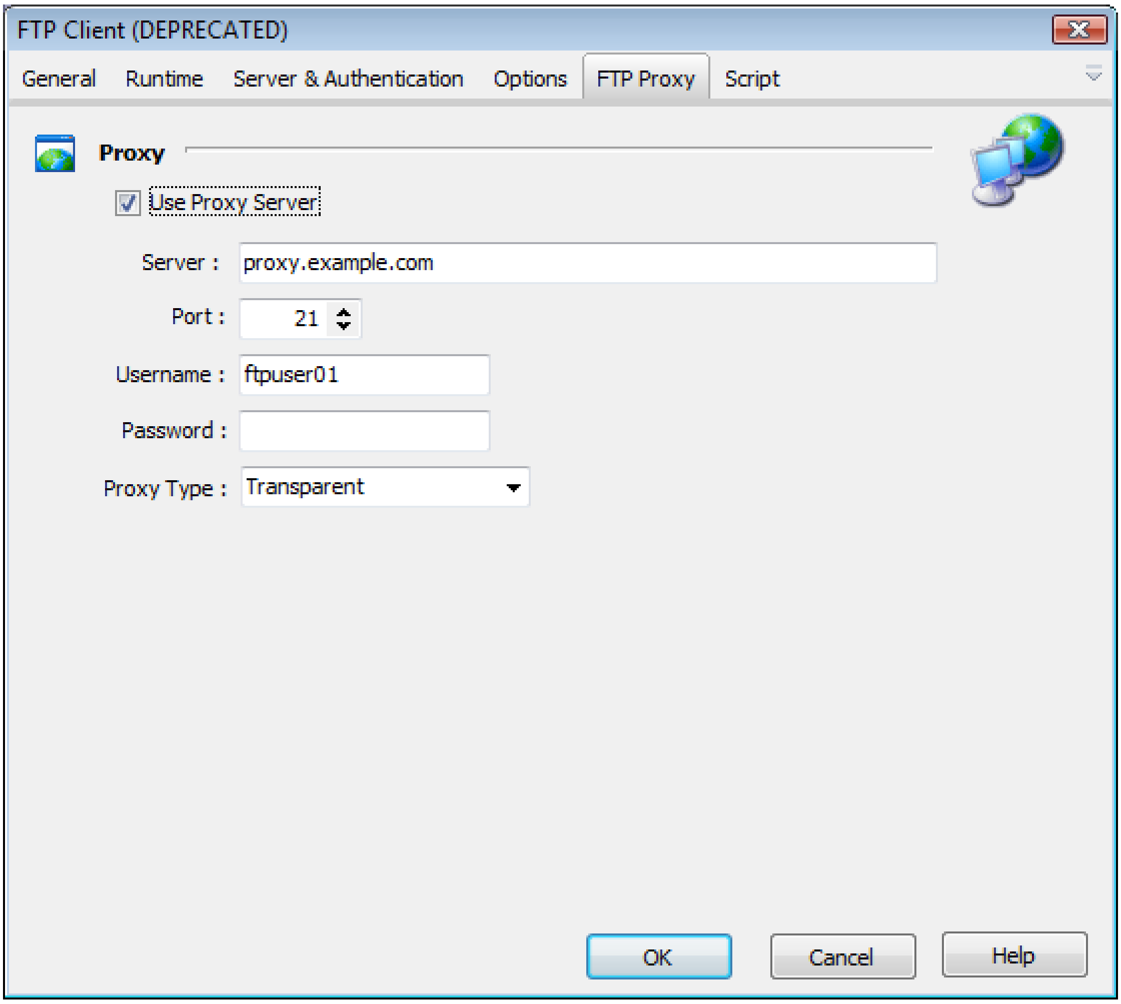
Passive : Instructs the FTP client and server to use passive mode transfers, this is needed when going through some firewalls

Detailed Logging : Enables Detailed logging.

Binary Transfer Mode : Sets the transfer mode to Binary (the default)

ASCII Transfer Mode : Sets the transfer mode to ASCII

Proxy Settings



☑ Use Proxy Server

Server : proxy.example.com

Port : 21

Username : ftpuser01

Password :

Proxy Type : Transparent

Server : The host name or ip address of the proxy

Port : The port number of the proxy

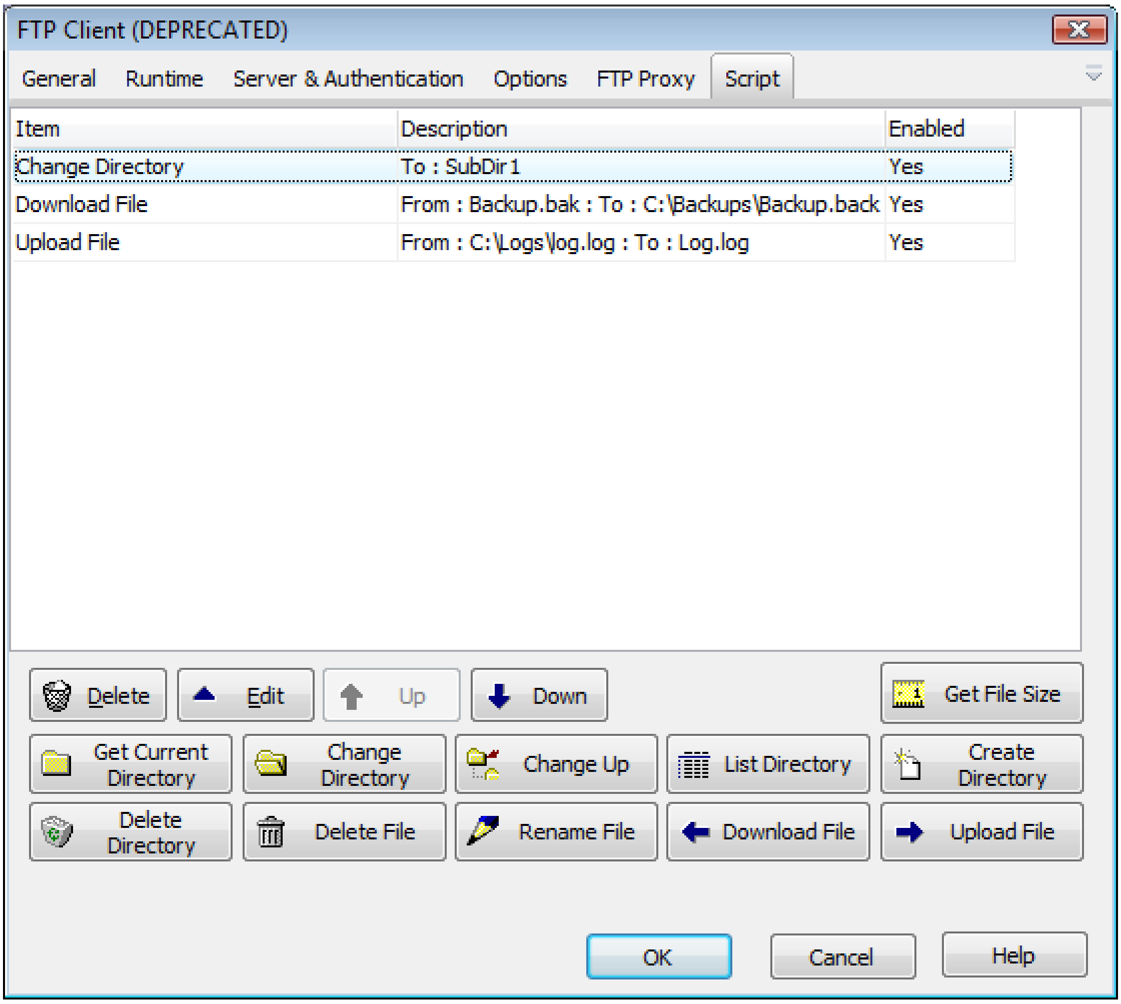
UserName : The User name for authentication

Password : The password for authentication

Proxy Type :

None - don't use a proxy
UserSite - Send command USER user@hostname
Site - Send command SITE (with logon)
Open - Send command OPEN
UserPass - USER user@firewalluser@hostname / PASS pass@firewallpass
Transparent - First use the USER and PASS command with the firewall username and password, and then with the target host username and password.
HttpProxyWithFtp - HTTP Proxy with FTP support. Will be supported in Indy 10
CustomProxy - use OnCustomFTPProxy to customize the proxy login

FTP Script



Item	Description	Enabled
Change Directory	To : SubDir1	Yes
Download File	From : Backup.bak : To : C:\Backups\Backup.back	Yes
Upload File	From : C:\Logs\log.log : To : Log.log	Yes

The FTP client works by adding FTP commands to the list. The available commands are :

Get Current Directory : This can be retrieved into a Variable

Change Directory : Change the remote directory, you can use Automise Variables with this command.

Change Up : Changes the remote directory to its parent directory, ie up one level

List Directory : Lists the remote directory into a variable.

Create Directory : Create a sub directory in the current remote directory.

Delete Directory : Deletes the specified remote directory.

Delete File : Deletes the specified remote File.

Rename File : Renames the specified Remote File.

Download File : Downloads the specified remote file to the specified local file.

Upload File : Uploads the specified local file to the specified remote file.

Scripting Info

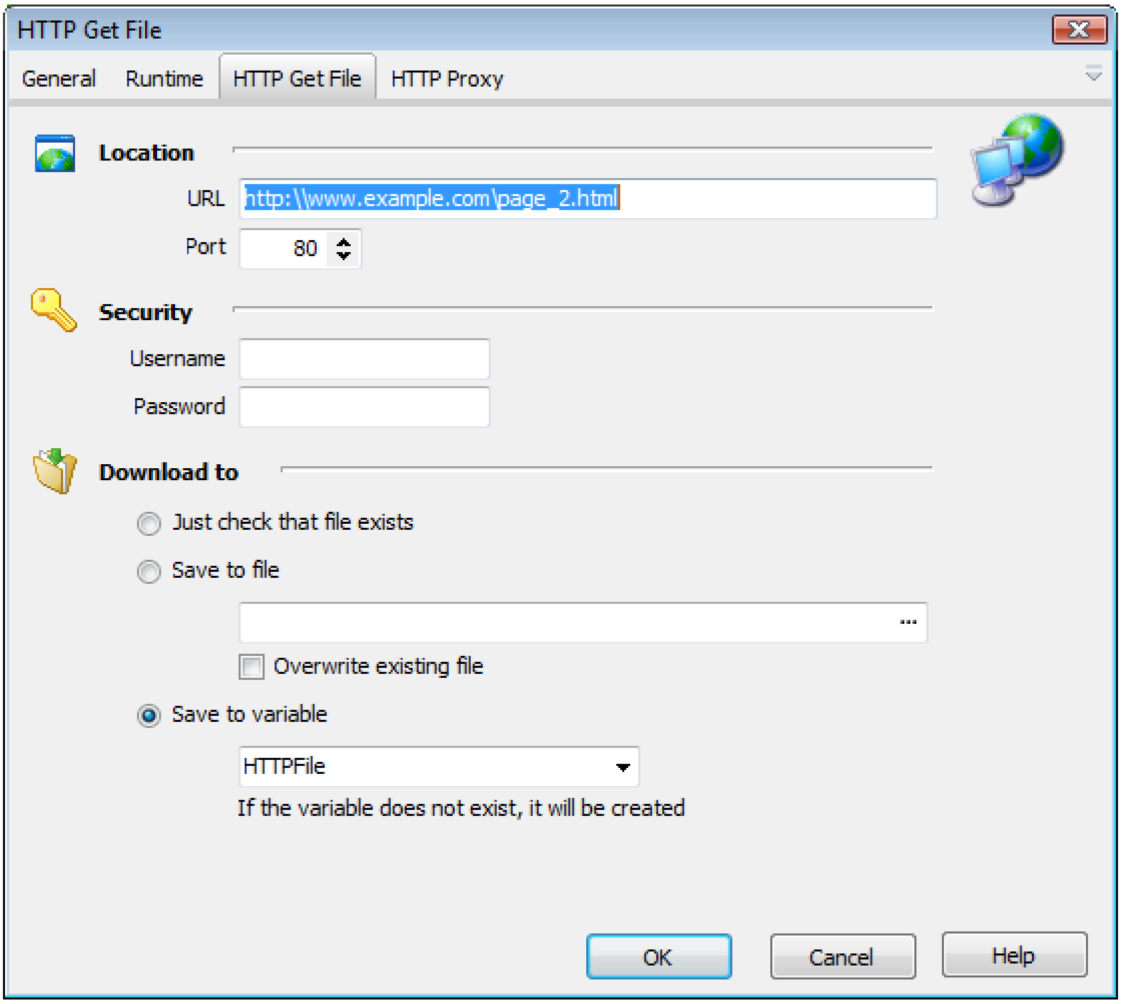
The Action properties available are :

property Host : WideString
property Port : integer
property UserID : WideString
property Password : WideString
property Passive : WordBool
property CurrentDir : WideString
property DetailedLogging : WordBool

This action uses the Open Source Indy Components, for more information see the Indy web site : http://www.nevrona.com/Indy

6.15.4 HTTP Get Action

This action enables you to download a file using the http protocol.



HTTP Get File

Location

URL: http:\\www.example.com\page_2.html

Port: 80

Security

Username:

Password:

Download to

○ Just check that file exists

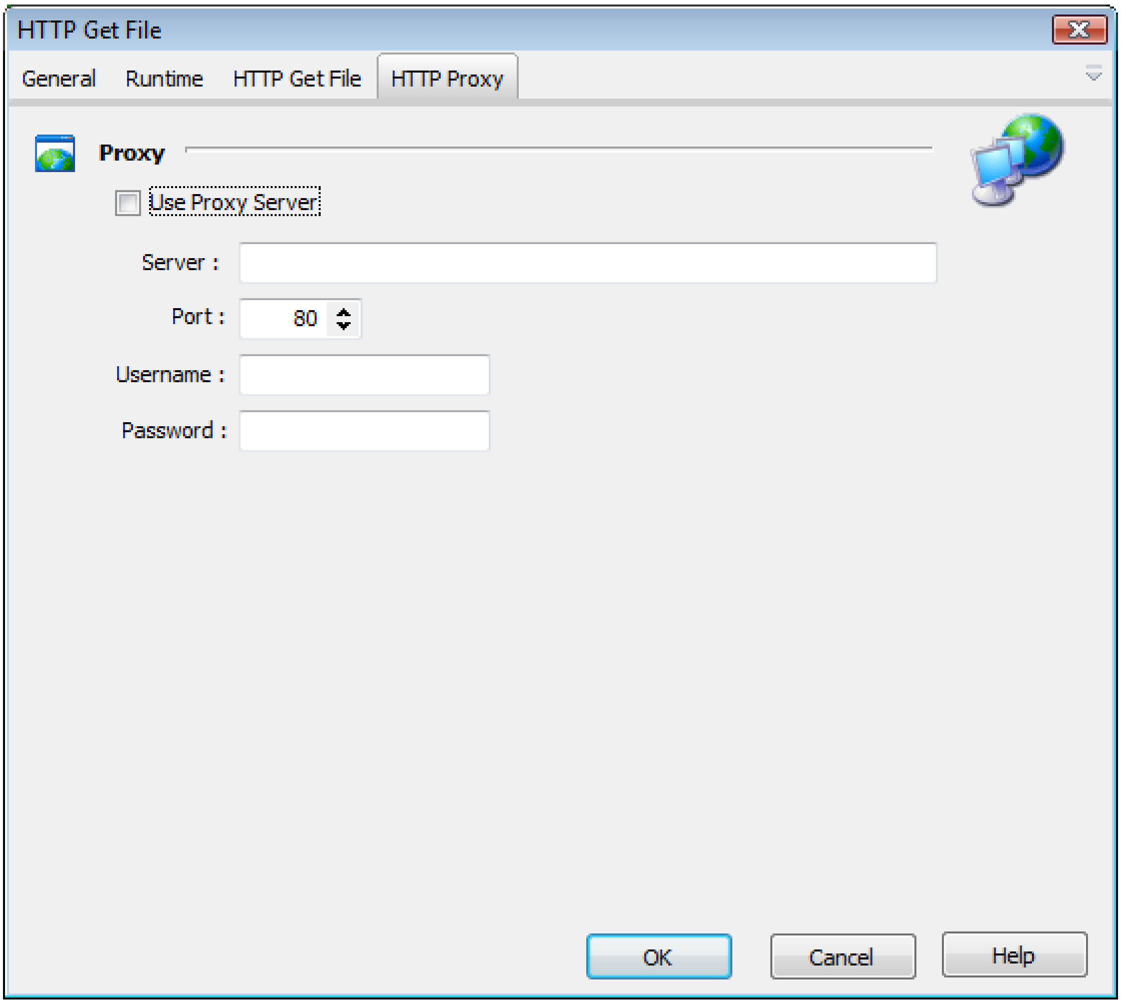
○ Save to file

□ Overwrite existing file

● Save to variable

HTTPFile

If the variable does not exist, it will be created



□ Use Proxy Server

Server :

Port : 80

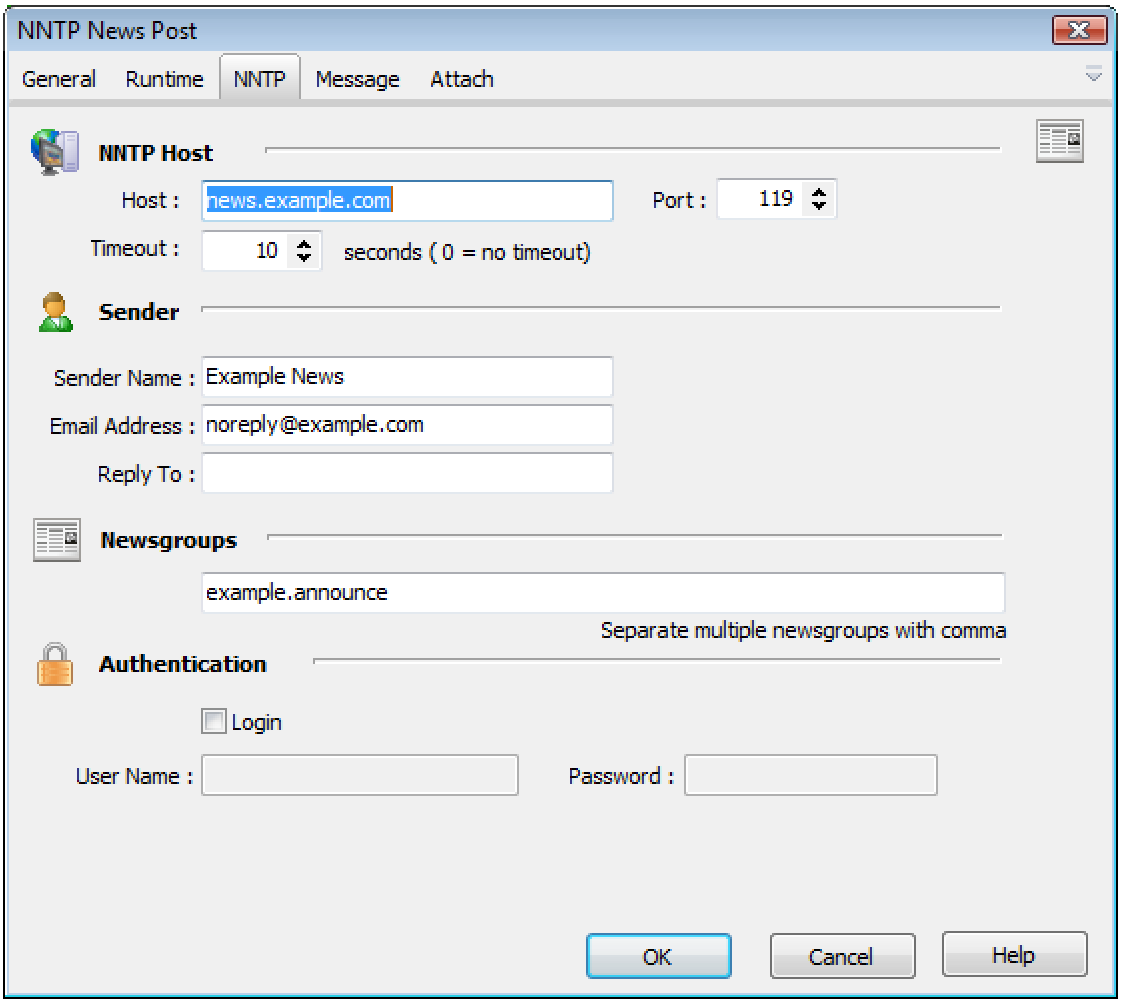
Username :

Password :

6.15.5 NNTP News Post Action

This action allows you to post a message on one or more news servers.

WARNING: Automating news posts to the outside world (ie not on an internal news server) is not only unwise, it can be construed as spam.



NNTP Host

Host : news.example.com

Port : 119

Timeout : 10 seconds (0 = no timeout)

Sender

Sender Name : Example News

Email Address : noreply@example.com

Reply To :

Newsgroups

example.announce

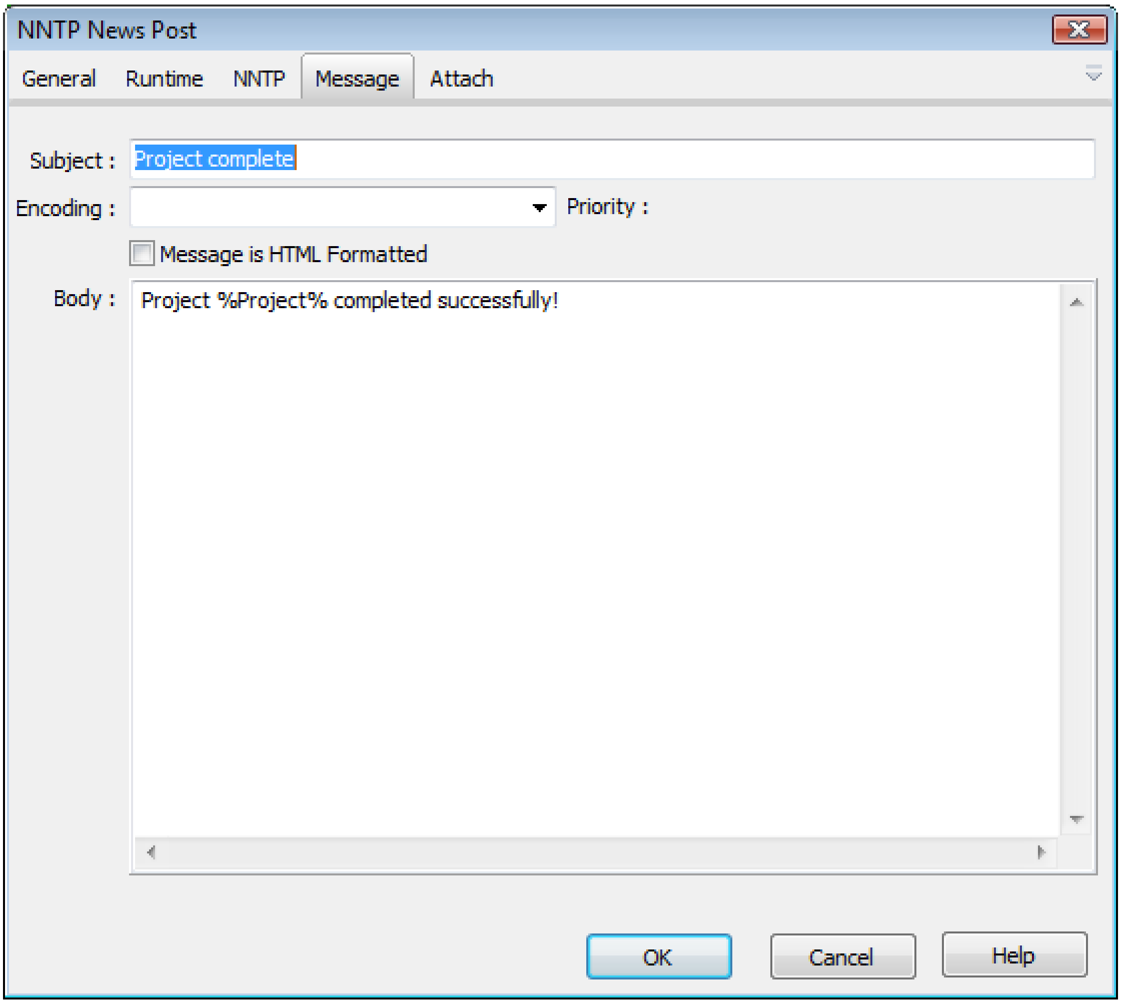
Separate multiple newsgroups with comma

Authentication

□ Login

User Name :

Password :



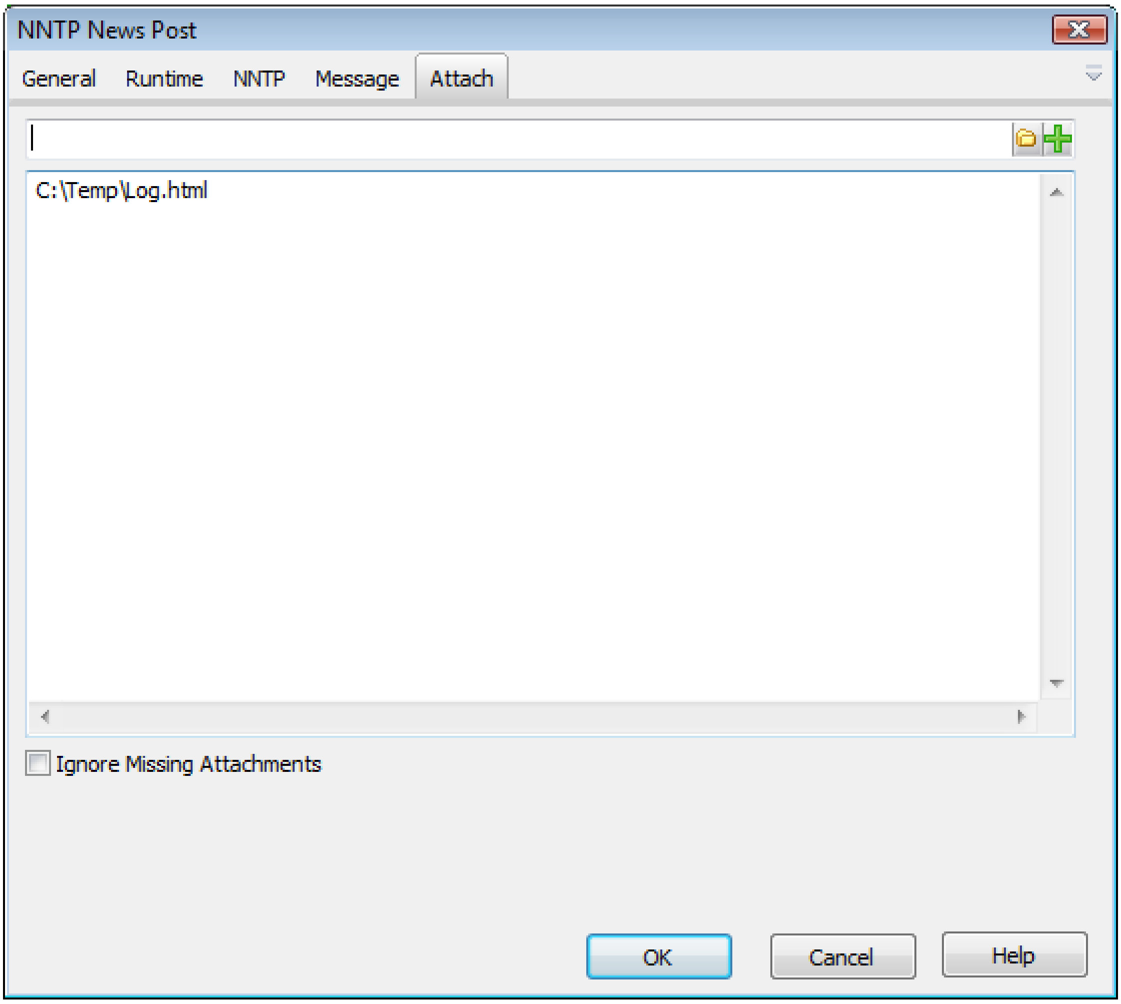
Subject : Project complete

Encoding :

Priority :

□ Message is HTML Formatted

Body : Project %Project% completed successfully!

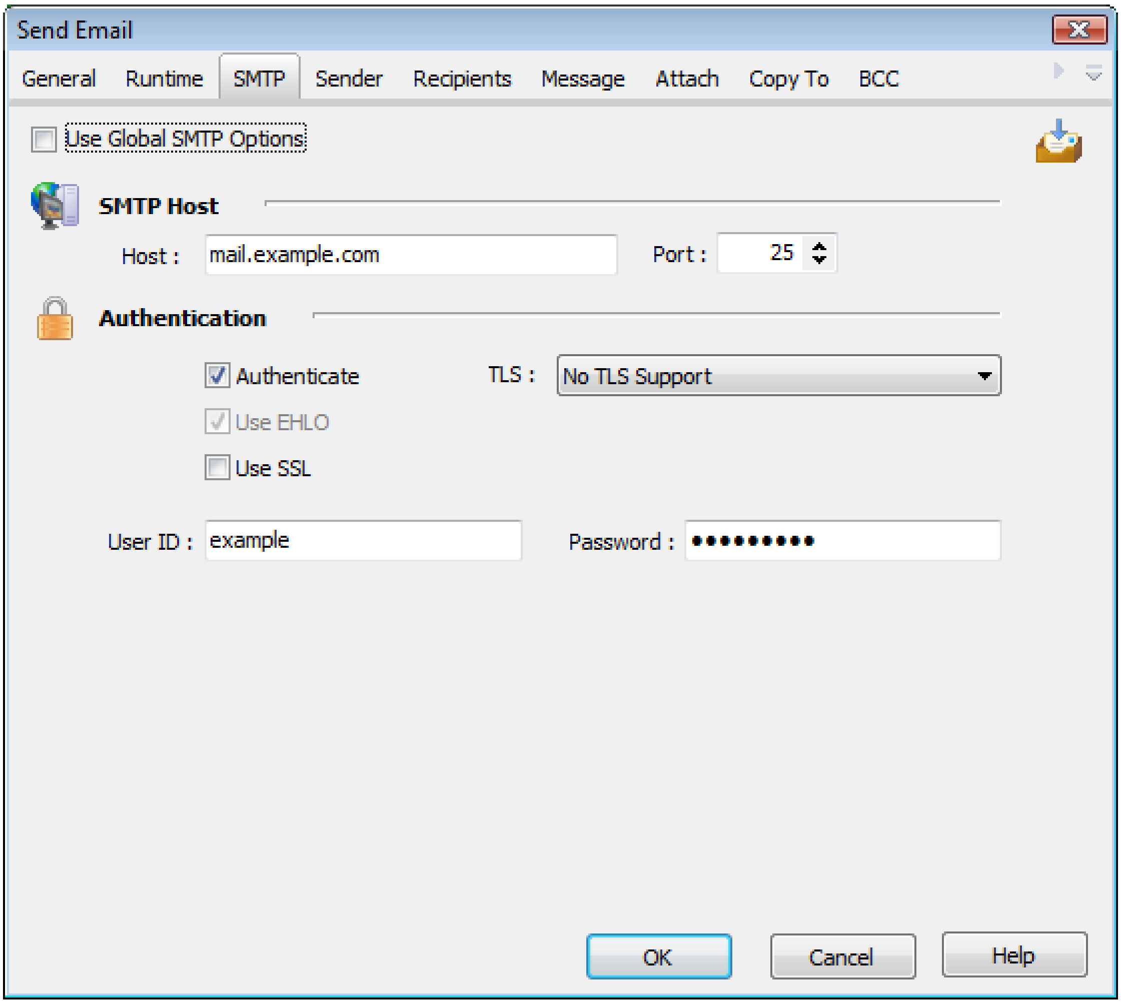


C:\Temp\Log.html

□ Ignore Missing Attachments

6.15.6 Send Email (SMTP)

This action allows your Automise projects to send email via an SMTP server.



□ Use Global SMTP Options

SMTP Host

Host : mail.example.com

Port : 25

Authentication

☑ Authenticate

TLS : No TLS Support

☑ Use EHLO

□ Use SSL

User ID : example

Password : ●●●●●●●●●●

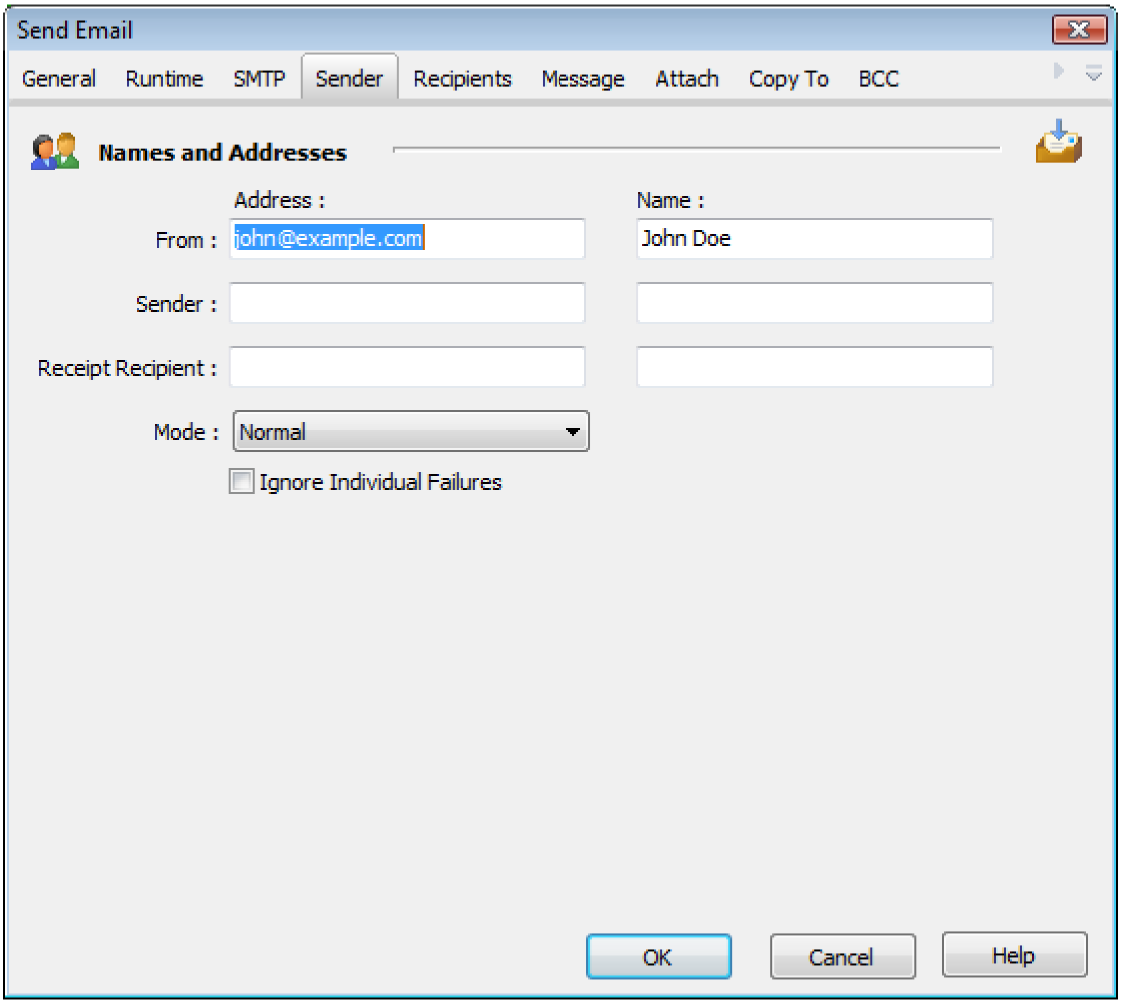
Host : The host name (or IP address) of the SMTP server

Port : The port on which to connect, defaults to 25

UserID / Password: If your smtp server requires authentication, provide your userid and password

Use EHLO: Enabling EHLO uses a slightly different protocol to communicate with the SMTP server. EHLO allows the SMTP server to report it's capabilities to the client (in this case Automise), and then the client can adjust it's protocol to support the reported capabilities.

It is generally a good idea to enable this option.



Address :
Name :

From : john@example.com John Doe

Sender :

Receipt Recipient :

Mode : Normal

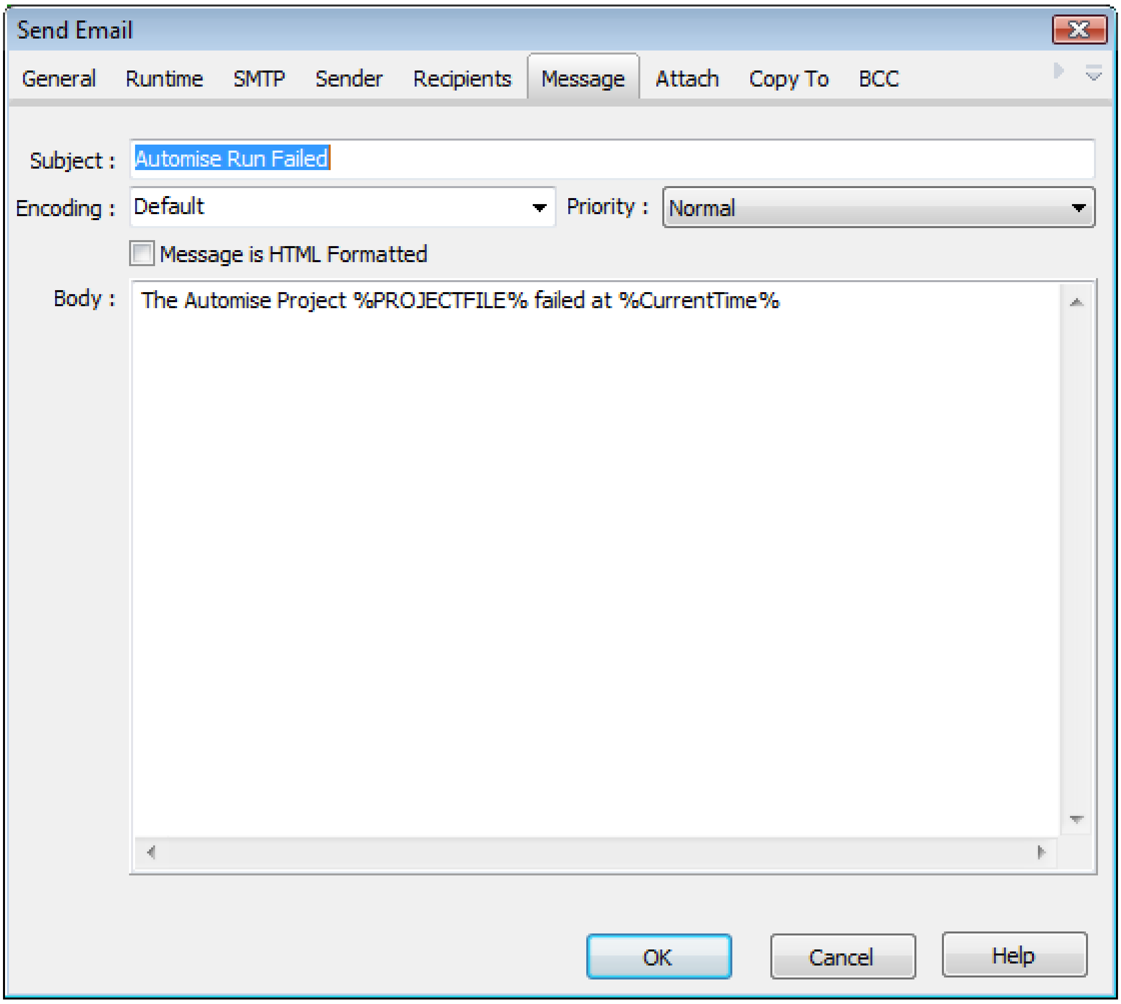
□ Ignore Individual Failures

From : This must be set to a valid email address

Sender : Set this when sending mail on behalf of someone else

Receipt Recipient : If you require a notice receipt then set the address for the receipt message to be sent to

Mode : When operating in normal Mode, the action sends a single email message. When operating in individual mode, it sends a message to each individual specified in the recipients property. Note that the copy to and blind copy to lists are not used in this mode.



Subject : Automise Run Failed

Encoding : Default

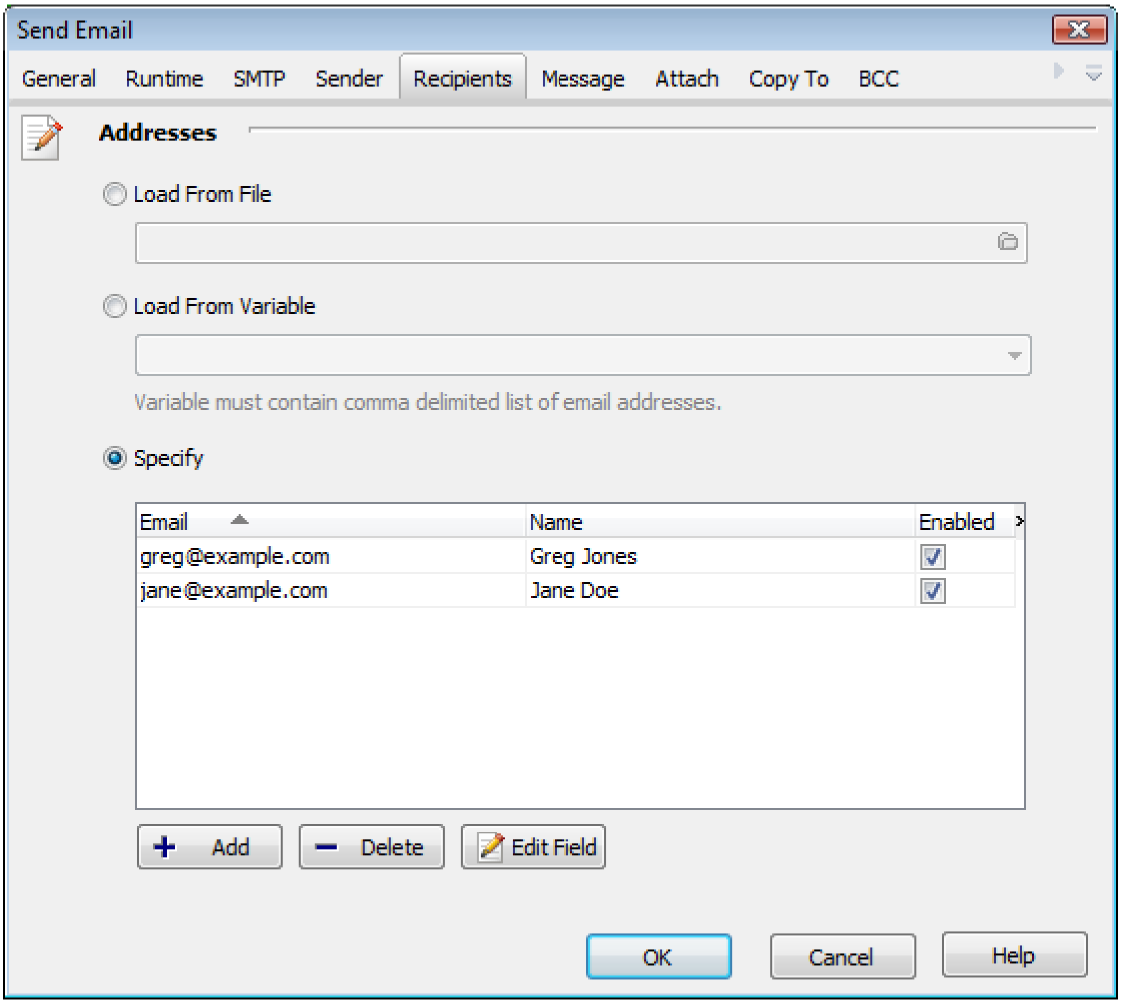
Priority : Normal

☐ Message is HTML Formatted

Body : The Automise Project %PROJECTFILE% failed at %CurrentTime%

Subject : The email message's subject

Body : The actual message.



Addresses

○ Load From File

○ Load From Variable

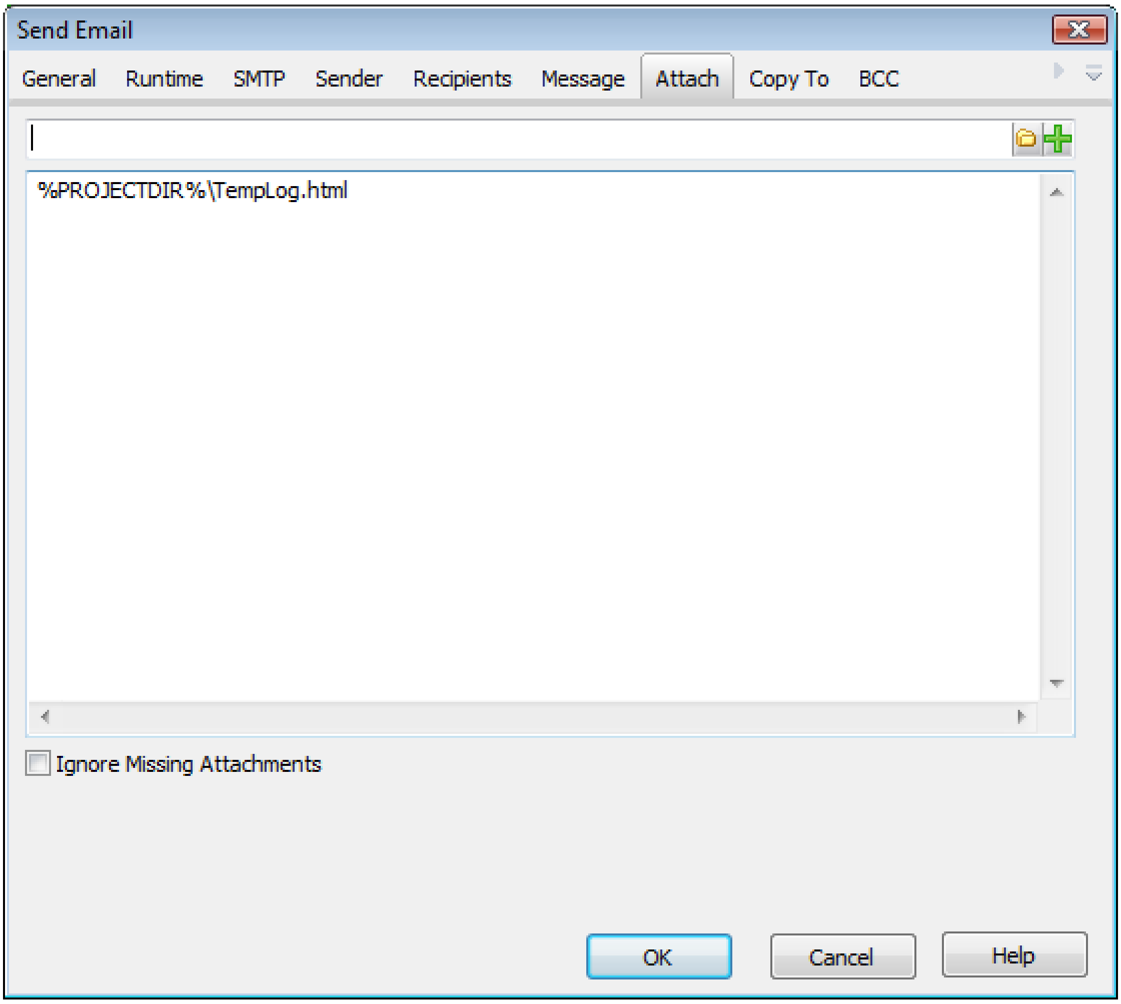
Variable must contain comma delimited list of email addresses.

● Specify

Email	Name	Enabled
greg@example.com	Greg Jones	☑
jane@example.com	Jane Doe	☑

The Recipients, Copy To, Blind Copy To and Reply To tabs all have the same grid that allows you to specify multiple email address/names.

Address can also be loaded from a text file. The format of the file is one recipient per line, with a comma separating the email address and name.



%PROJECTDIR%\TempLog.html

□ Ignore Missing Attachments

To attach a file, click on the ... button and select the file(s) using the dialog. Then click on the tick button to add them to the attachments list.

Scripting Info

The Action properties available are :

property Host : WideString;
property Port : integer;
property UserId : WideString;
property Password : WideString;
property Authenticate : WordBool;
property Subject : WideString;

property Recipients : IFBAddressList;
property CopyTo : IFBAddressList;
property BlindCopyTo : IFBAddressList;

property Sender : IFBEmailAddress;
property From : IFBEmailAddress;
property ReplyTo : IFBEmailAddress;
property ReceiptRecipient : IFBEmailAddress;
property Mode : integer ; // valid values are emNormal and emIndividual
property Body : WideString;

IFBAddressList interface (for Recipients, CopyTo and Blind Copy To)

function Add(const Name,Address : WideString) : IFBEmailAddress;
function Item(Index : integer) : IFBEmailAddress;
procedure Clear;
procedure RemoveItem(Index : integer);
property Count : integer;

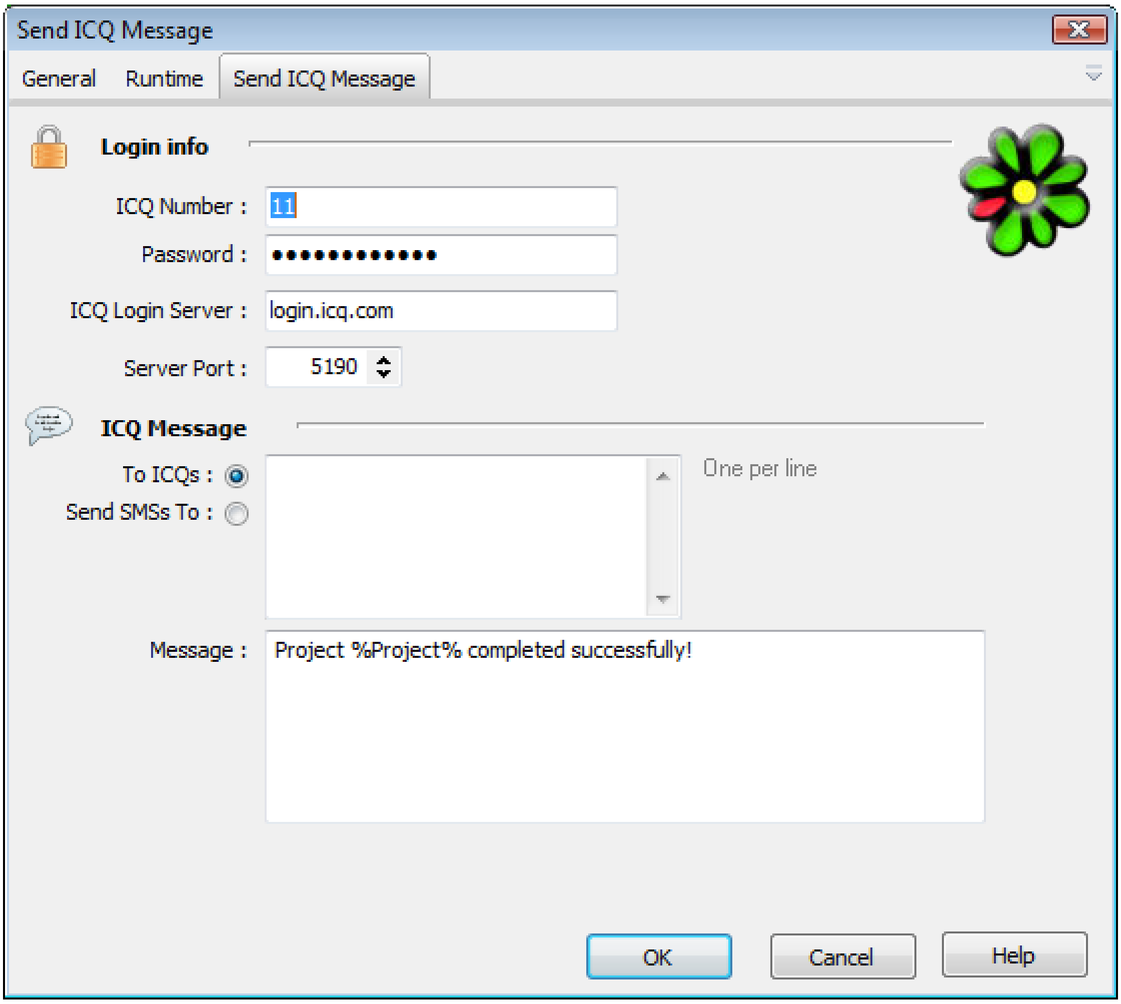
IFBEmailAddress interface (for Sender, From,ReplyTo and ReceiptRecipient)

property Name : WideString
property Address : WideString
property Text : WideString; //in the form "My Name" <myaddress@mycompany.com>

This action uses the Open Source Indy Components, for more information see the Indy web site : http://www.nevrona.com/Indy

6.15.7 Send ICQ Message Action

This action enables you to send ICQ messages.



ICQ Number : 11

Password : ●●●●●●●●●●●●●

ICQ Login Server : login.icq.com

Server Port : 5190

To ICQs : ●

Send SMSs To : ○

Message : Project %Project% completed successfully!

Login info

Specify an existing ICQ number, password, an ICQ server to log in to and a port to connect to.

ICQ Message

Messages can be sent to either ICQ numbers of SMS phone numbers (for SMS, the target phone number will need to be connected to an ICQ-supported network.) Put each new phone or ICQ number on a new line.

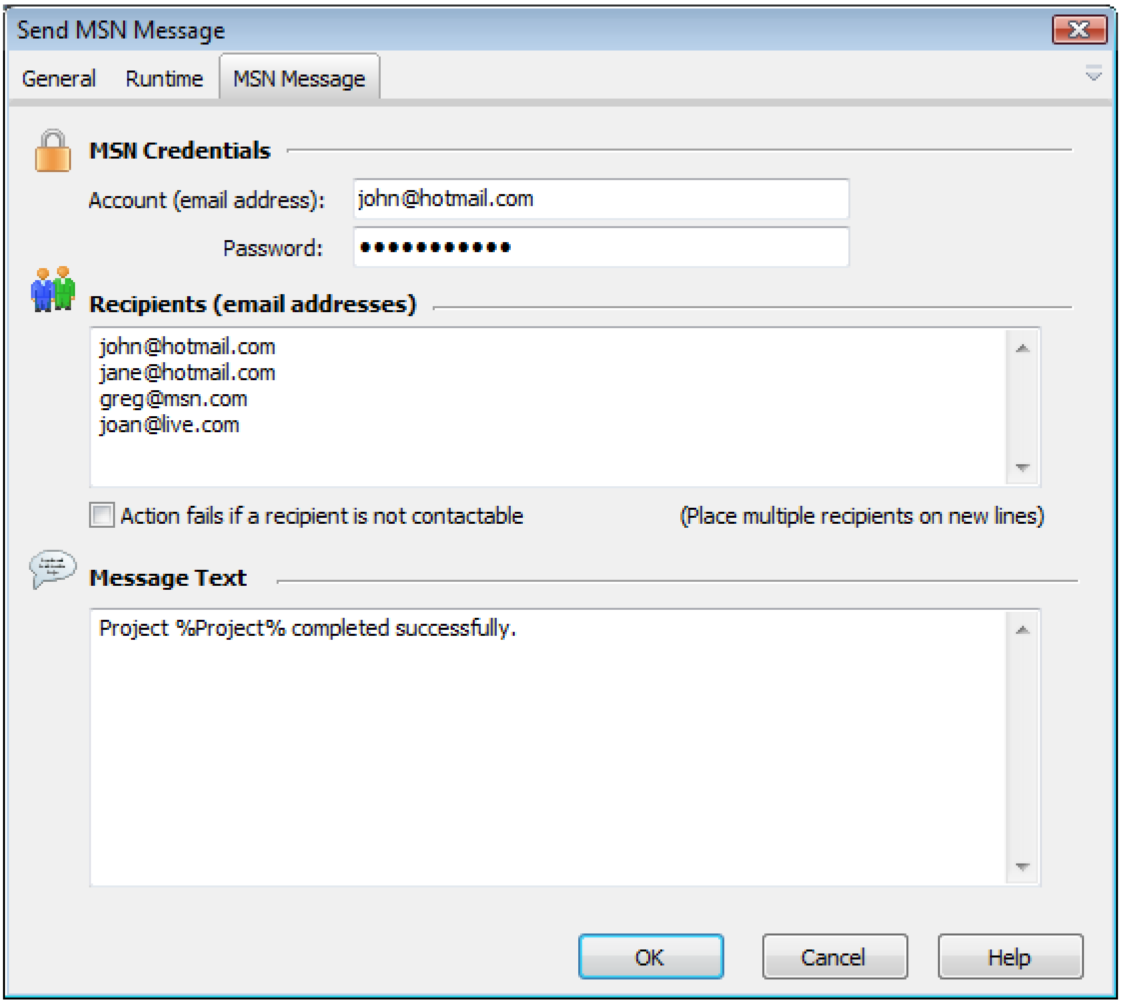
Message

The text of the ICQ message.

6.15.8 Send MSN Message Action

The Send MSN Message action enables you to send a message to one or more MSN Messenger users as part
of your project.

The MSN action uses the open source dotMSN library to access MSN Messenger (see below for copyright details.)



Account (email address): john@hotmail.com

Password: ●●●●●●●●●●●●

john@hotmail.com
jane@hotmail.com
greg@msn.com
joan@live.com

□ Action fails if a recipient is not contactable

(Place multiple recipients on new lines)

Project %Project% completed successfully.

MSN Credentials

Specify an account and a password for Microsoft's MSN. You can register for an MSN account at http://www.passport.net.

Note that the MSN action does not require Windows Messenger to be installed. Furthermore, MSN does not allow the same user to be logged in multiple times. It is suggested that, to avoid conflicts, you create a separate MSN account for Automise.

To set server and proxy details, go to the Options.

Recipients

Specify each recipient on a new line. Specify email addresses (used for MSN login) rather than screen names.

While the recipients do not need to be on the contact list for the login account, it is recommended that the login account is added to each of the recipients' contact lists.

(ie In the above example, while angus@Automise.com does not need vincent@Automise.com on his contact list, it is recommended that vincent@Automise.com adds angus@Automise.com to his contact list.)

Action fails if a recipient is not contactable

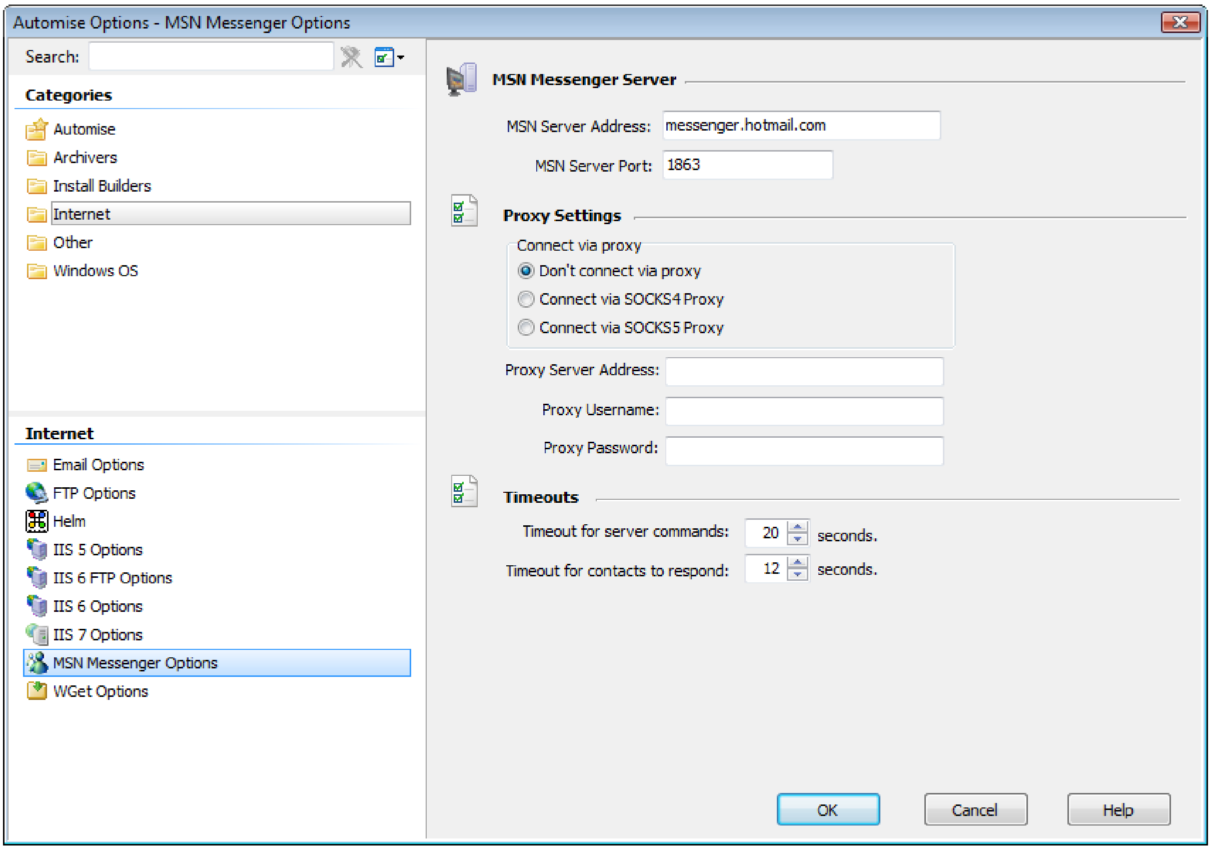
The action can be set to fail if one or more contacts could not be reached (either because they are offline or because that email address is not registered with MSN.)

Note that, unlike ICQ, MSN cannot send messages to clients who are offline (the message sending will fail.)

Message Text

This is the message that will be sent to each client.

Options



Automise Options - MSN Messenger Options

MSN Messenger Server

MSN Server Address: messenger.hotmail.com

MSN Server Port: 1863

Proxy Settings

Connect via proxy

● Don't connect via proxy

○ Connect via SOCKS4 Proxy

○ Connect via SOCKS5 Proxy

Proxy Server Address:

Proxy Username:

Proxy Password:

Timeouts

Timeout for server commands: 20 seconds.

Timeout for contacts to respond: 12 seconds.

To access the MSN Messenger Options, go to Tools -> Options and click on the 'Internet' tab.

MSN Messenger Server

Specify the server to log in to. The defaults are recommended for the standard MSN network.

Proxy Settings

If your build machine is firewalled, the action can connect via a SOCKS4 or SOCKS5 proxy.

Timeouts

Timeout for server commands

This is the timeout for the server to respond to commands. Because the server can sometimes be quite sluggish to respond (especially if you have lots of contacts), it is not recommended that this is set any lower than five seconds.

If your internet connection is slow or congested, you may need to set it higher.

Timeout for contacts to respond

This is the amount of time each contact will be allowed to respond to the message before being declared offline or unavailable.

You may need to increase this value if online users are failing to receive messages.

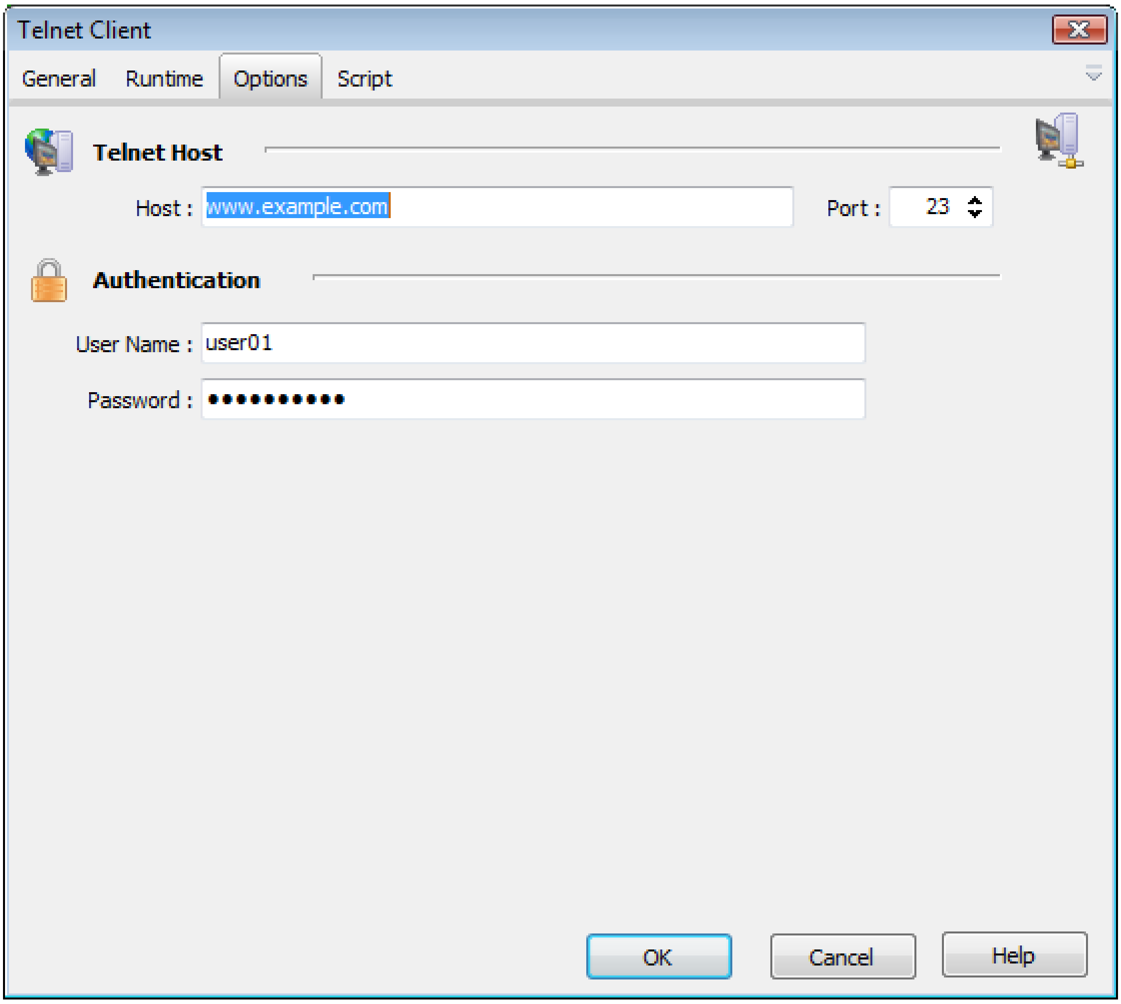
DotMSN Copyright

The Automise MSN Message action uses the open source DotMSN assembly to access MSN. The DotMSN binary assembly remains copyright as follows:

```
DotMSN Copyright (c) 2002-2005, Bas Geertsema, Xih Solutions
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.
    * Neither the names of Bas Geertsema or Xih Solutions nor the names of its
      contributors may be used to endorse or promote products derived
      from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.
```

6.15.9 Telnet Client Action

This action provides a simple scriptable Telnet client.



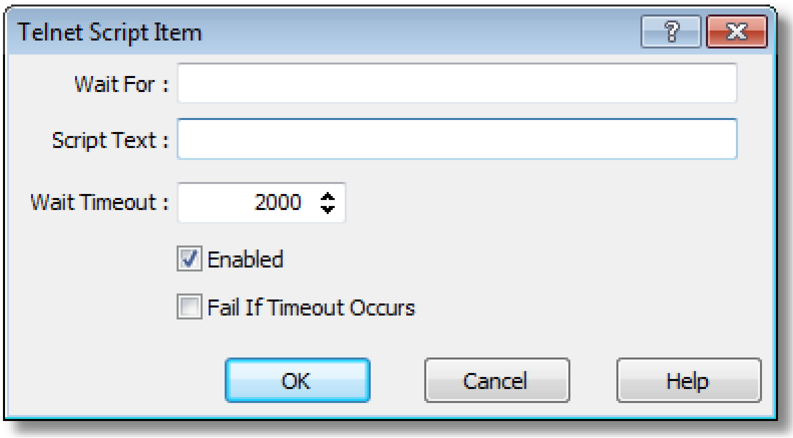
Host : www.example.com

Port : 23

User Name : user01

Password : ●●●●●●●●●●●

The script items allow you to wait for a string and (optionally) respond with a string.



Wait For :

Script Text :

Wait Timeout : 2000

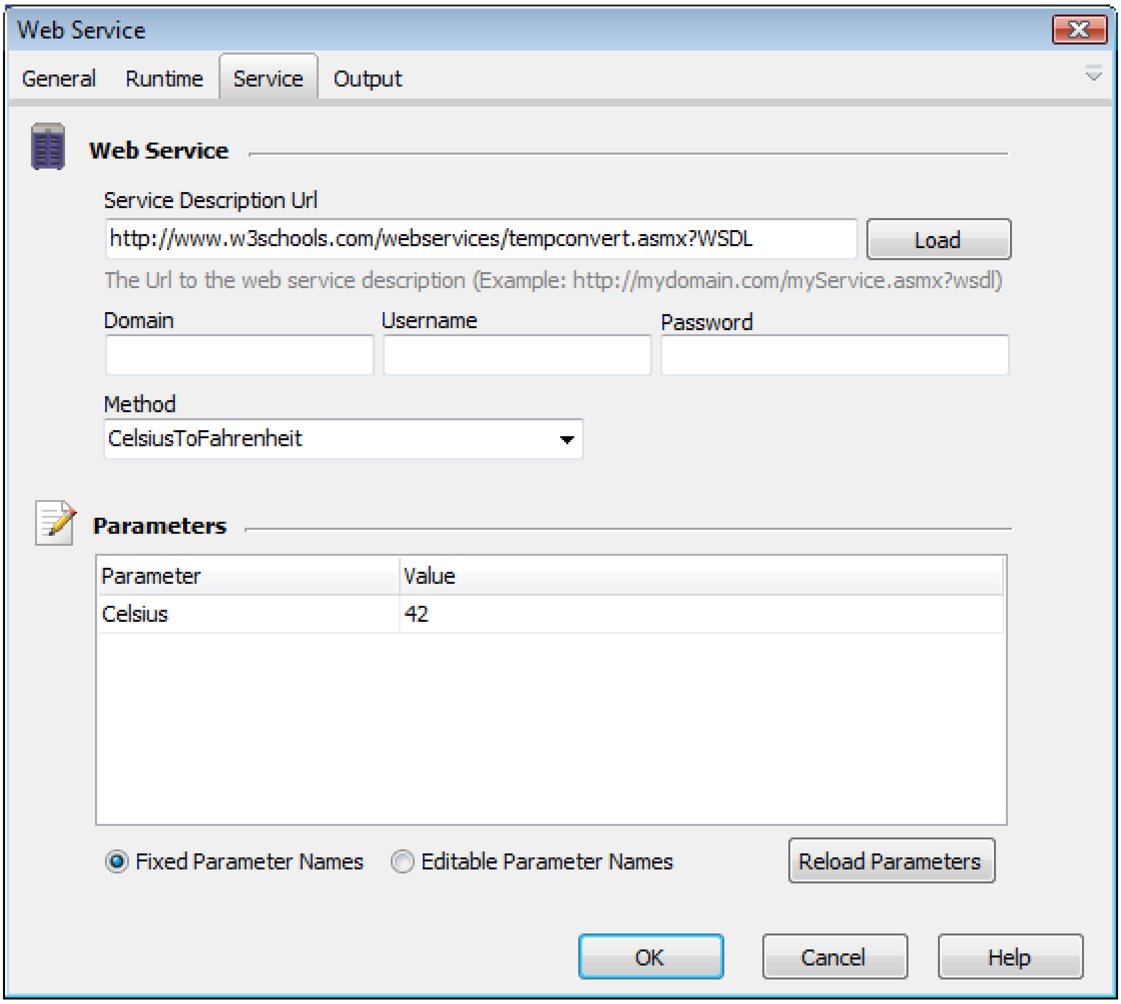
☑ Enabled

□ Fail If Timeout Occurs

To send the UserName set the ScriptText field to @USERNAME
To send the Password set the ScriptText field to @PASSWORD

6.15.10 Web Service Action

The Web Service Action allows you to call a method of a web service and read results into your running project.



Service Description Url: http://www.w3schools.com/webservices/tempconvert.asmx?WSDL

The Url to the web service description (Example: http://mydomain.com/myService.asmx?wsdl)

Domain:

Username:

Password:

Method: CelsiusToFahrenheit

Parameter	Value
Celsius	42

● Fixed Parameter Names ○ Editable Parameter Names

Service Description URL

Enter the URL for the web service description (WSDL) file. Once you have entered a valid URL, click the Load button to automatically populate the rest of the page with data from the Web Service.

In the example above, the WSDL is for a "Temperature Conversion" service.

Method

Enter the name of the method to call. You can call any method which takes only Primitive Types as parameters (complex type parameters are not supported.)

This dropdown is automatically populated when you click the Load button.

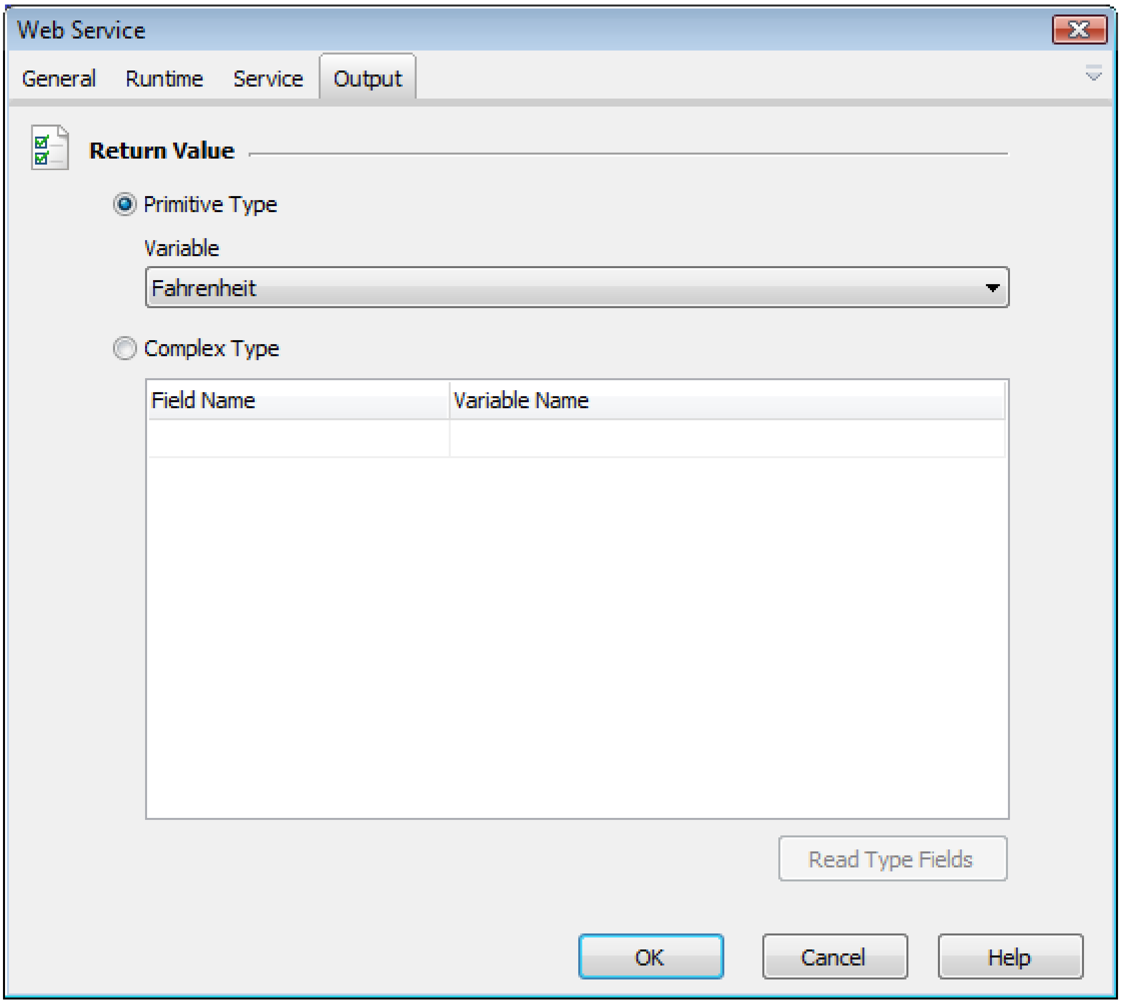
Parameters

Enter the values for method parameters. If you have clicked the Load button, this list is automatically populated when you choose a method name from the Method dropdown.

You can choose to show the list as **Fixed Parameter Names** (as shown above) or **Editable Parameter Names.** The latter displays parameters as a text field of

<Key>=<Value> pairs. This allows you to enter parameters when the WSDL is not available at design time.

The Reload Parameters button will clear the parameter values and reset the list.



● Primitive Type

Variable: Fahrenheit

○ Complex Type

The Output page allows you to set variable values based on the output of the method.

In the example above, the method returns a Primitive Type which will be assigned to the Celcius variable.

If the method returns an array of primitive types, they will be assigned to the variable as a list of strings - one per line. Alternatively, you can use the Web Service Iterator Action to iterate through them.

If the method returns a complex type, you can choose a variable name to set for each field of the complex type. Click on "Read Type Fields" to load the list of available field names from the WSDL file. You do not need to set a variable for every field, just the fields you need.

6.15.11 Web Service Iterator Action

The Web Service Iterator action is very similar to the Web Service action, but can be used when a web service returns an array of values. The action will iterate over each entry in the array.

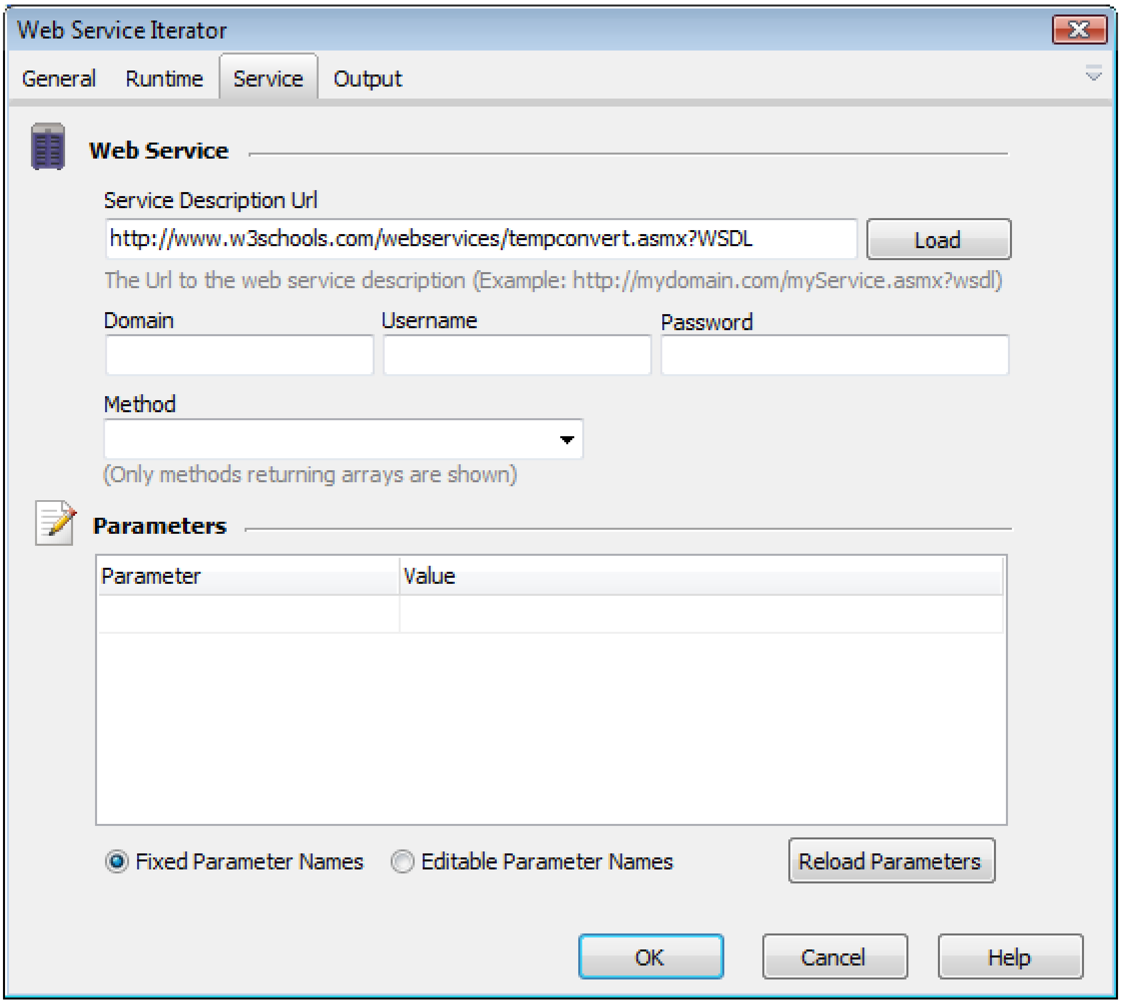
For more information about iterators, see the Iterators Overview Topic.

Service Page

Configuring the Service details for the action is the same as the Web Service action. See that topic for information.

The only difference is that the "Method" dropdown will only show methods that return arrays of values.

Output Page



Web Service Iterator

General Runtime Service Output

Web Service

Service Description Url

http://www.w3schools.com/webservices/tempconvert.asmx?WSDL

Load

The Url to the web service description (Example: http://mydomain.com/myService.asmx?wsdl)

Domain

Username

Password

Method

(Only methods returning arrays are shown)

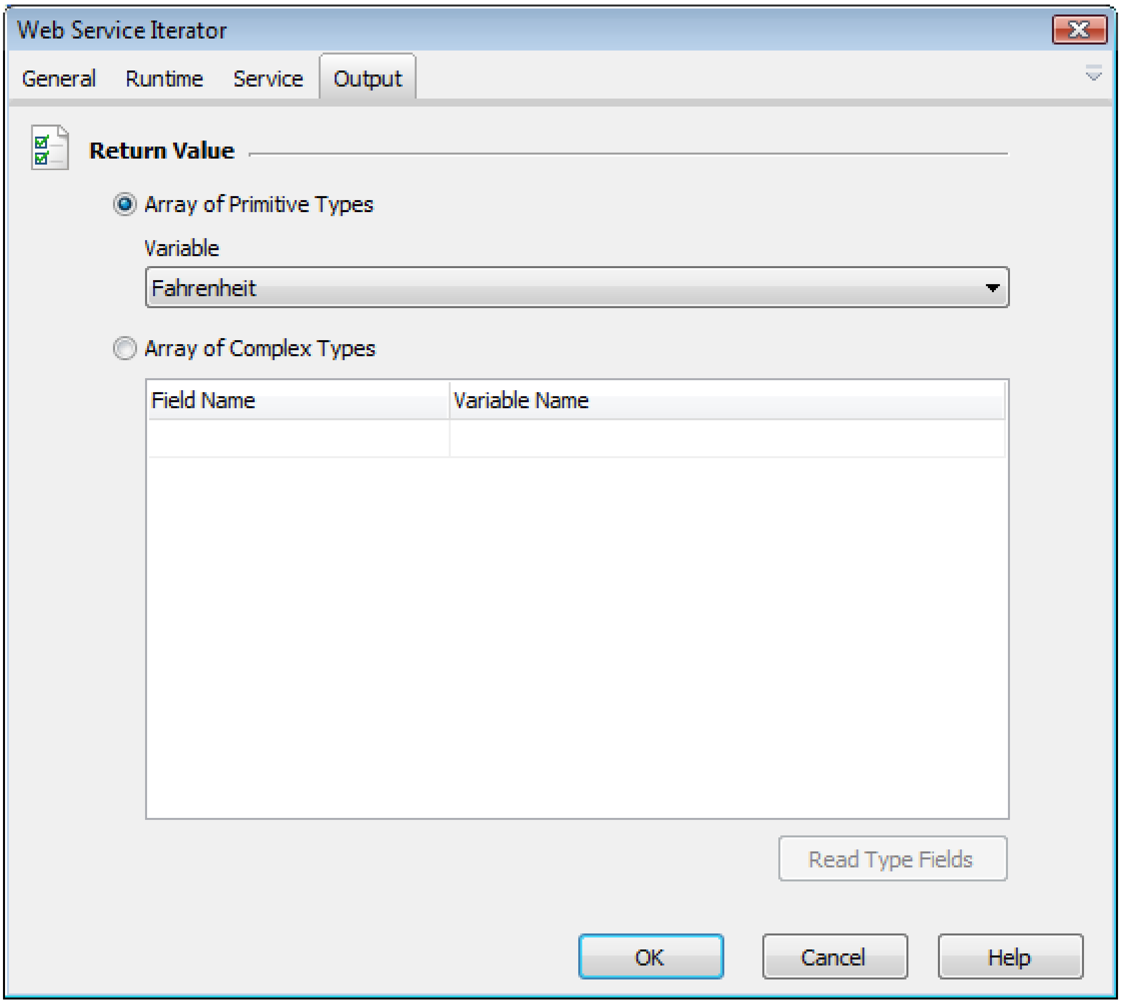
Parameters

Parameter	Value

● Fixed Parameter Names ○ Editable Parameter Names

Reload Parameters

OK Cancel Help



Web Service Iterator

General | Runtime | Service | Output

Return Value

● Array of Primitive Types

Variable: Fahrenheit

○ Array of Complex Types

Field Name	Variable Name

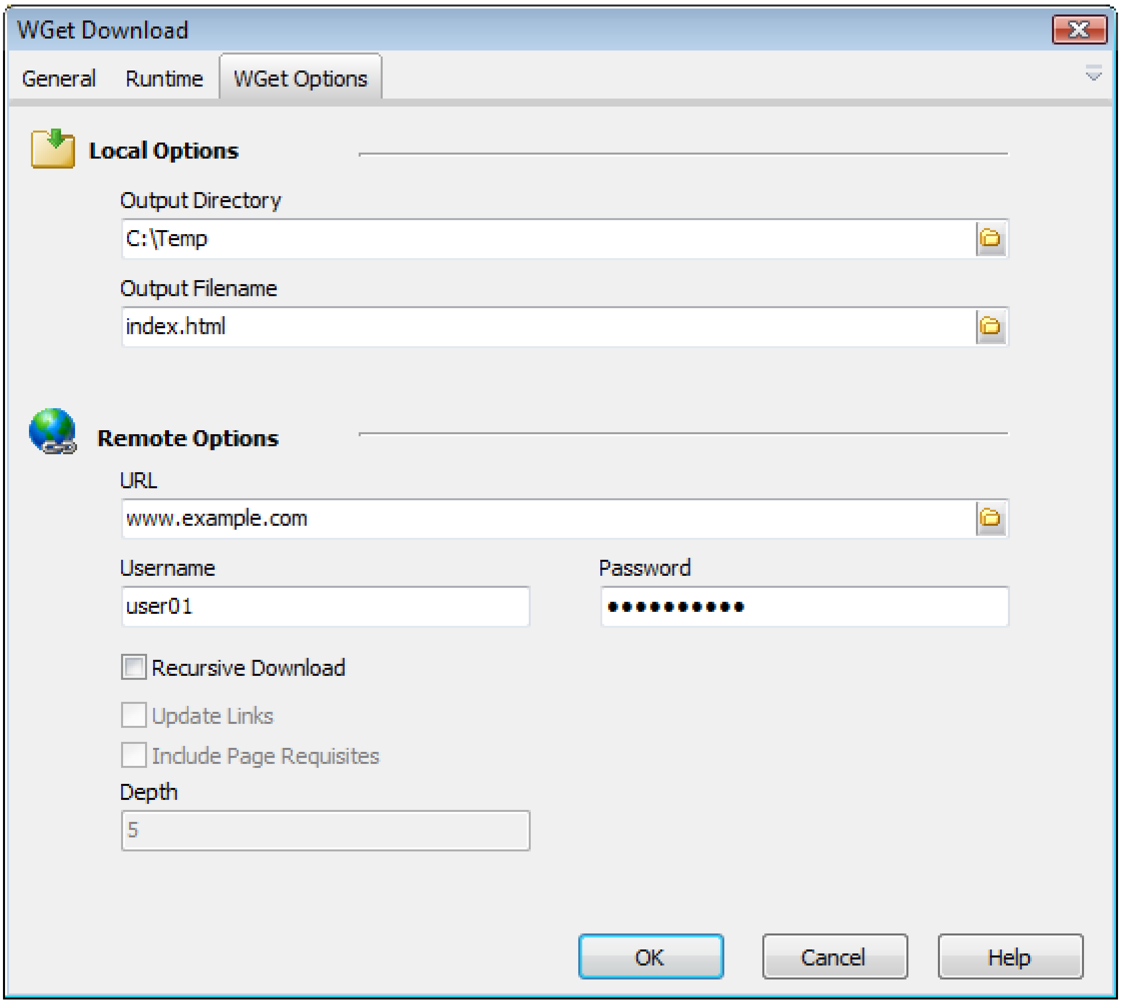
The Output page is similar to the output page for the Web Service Action.

If the method returns an array of Primitive types, then a variable is selected to iterate the values of the array.

If the method returns an array of complex types (as shown above), then the iterator will iterate over the values of the array and set each variable to the corresponding field of the complex type. Click on "Read Type Fields" to populate the list of available fields for the method. You do not need to specify a variable for each field, just the fields which you need to read.

6.15.12 WGet Download

The WGet Download action enables you to retrieve files using HTTP, HTTPS or FTP from a remote server.
The location of the WGet executable can be specified in the Tools - Options - Internet - WGet Options
For more information see http://www.gnu.org/software/wget/



Output Directory: C:\Temp

Output Filename: index.html

URL: www.example.com

Username: user01

Password: ●●●●●●●●●●

□ Recursive Download

□ Update Links

□ Include Page Requisites

Depth: 5

Output Directory
The base location that WGet will use when downloading files.

Output Filename
The Filename that the file will be saved to. If a full path is provided the Output Directory value will be ignored.

URL
The remote file to download

Username & Password
The authentication information to send to the remote server.

Recursive Download
Use this option to traverse parts of the Web (or a single http or ftp server), following links and directory structure.

Update Links
After the download is complete, convert the links in the document to make them suitable for local viewing.

Include Page Requisites
With this option enabled, WGet will download all the files that are necessary to properly display a given html page. This includes such things as images, sounds, and referenced style sheet.

Depth
The maximum depth to which the retrieval may descend into a remote site.

Scripting Info

The Action properties available are :

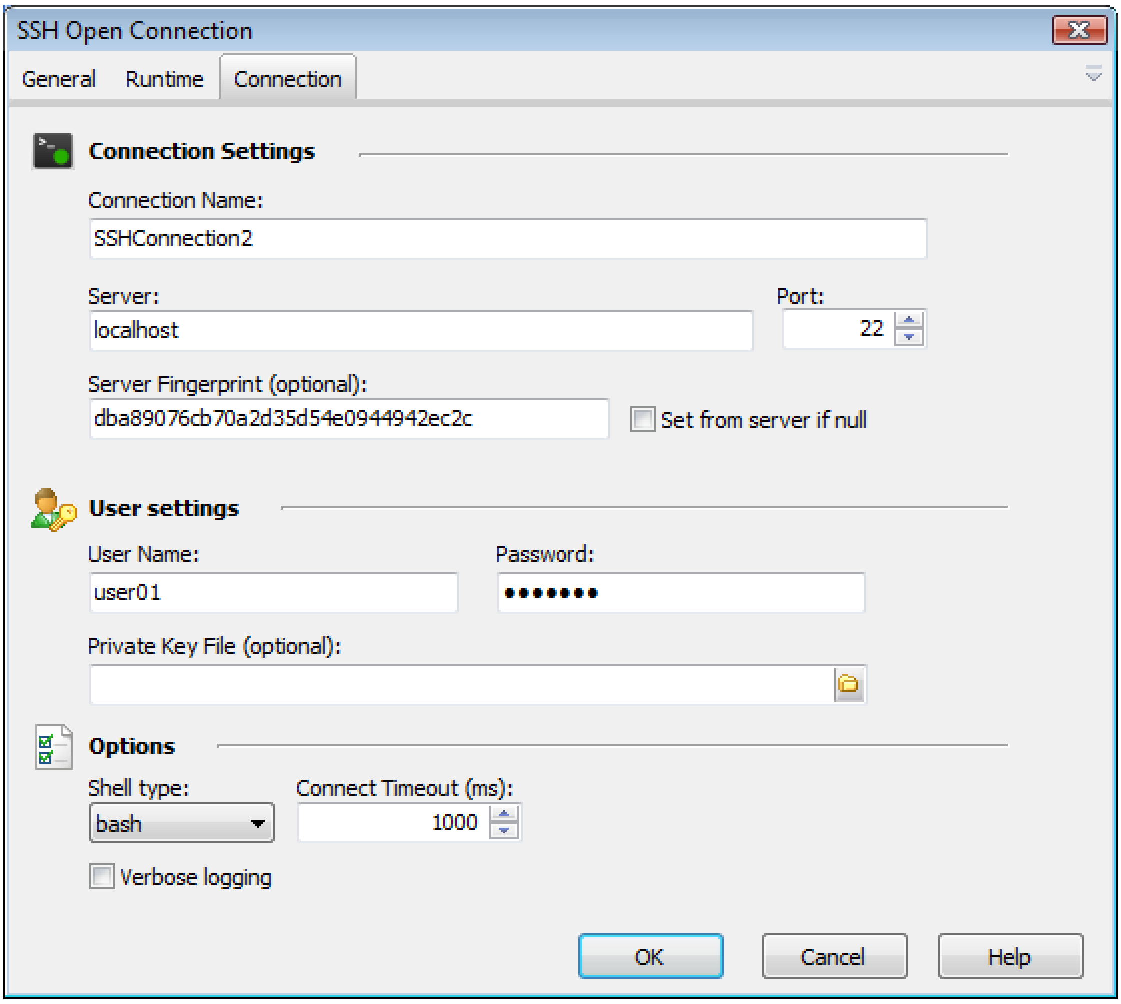
```
property  URL                    : String;
property  OutputDirectory        : String;
property  OutputFilename         : String;
property  Username               : String;
property  Password               : String;
property  Recursive              : Boolean;
property  RecursiveDepth         : String;
property  UpdateLinks            : Boolean;
property  IncludePageRequisites  : Boolean;
```

6.15.13 SSH

The SSH actions allow you to connect to an SSH server and execute arbitrary commands. There are two broad approaches: batch and open/execute/close.

6.15.13.1 SSH Open Connection

The SSH Open Connection action opens a connection to a SSH server. This connection can then be used by the SSH Execute Command and SSH Close Connection actions.



Connection Settings

Connection Name: SSHConnection2

Server: localhost

Port: 22

Server Fingerprint (optional): dba89076cb70a2d35d54e0944942ec2c

□ Set from server if null

User settings

User Name: user01

Password: •••••••

Private Key File (optional):

Options

Shell type: bash

Connect Timeout (ms): 1000

□ Verbose logging

Connection Name
A name for the connection.

Server
The IP address or URL for the server.

Port
The port that the SSH server is listening to

Server Fingerprint
This is the MD5 hash of the server's public RSA key. It is used to validate that the server you are connected to is the one that are are expecting to connect to.
To get the fingerprint, you can:

- On the SSH server, run `ssh-keygen -lf /etc/ssh/ssh_host_rsa_key.pub`
- Tick the "Set from server if null" box. The first time the action connects to the server, it will store the sever's fingerprint for you

User Name
The user name to authenticate with. This is required

Password
Either the password for the user **or** the password for the private key file, if one is specified

Private Key File
A PuTTY-formatted private key file. If password protected, the password for the file must be specified in the "Password" field. **NB** if a key file is specified it will be used. If no key file is specified, username/password will be used.

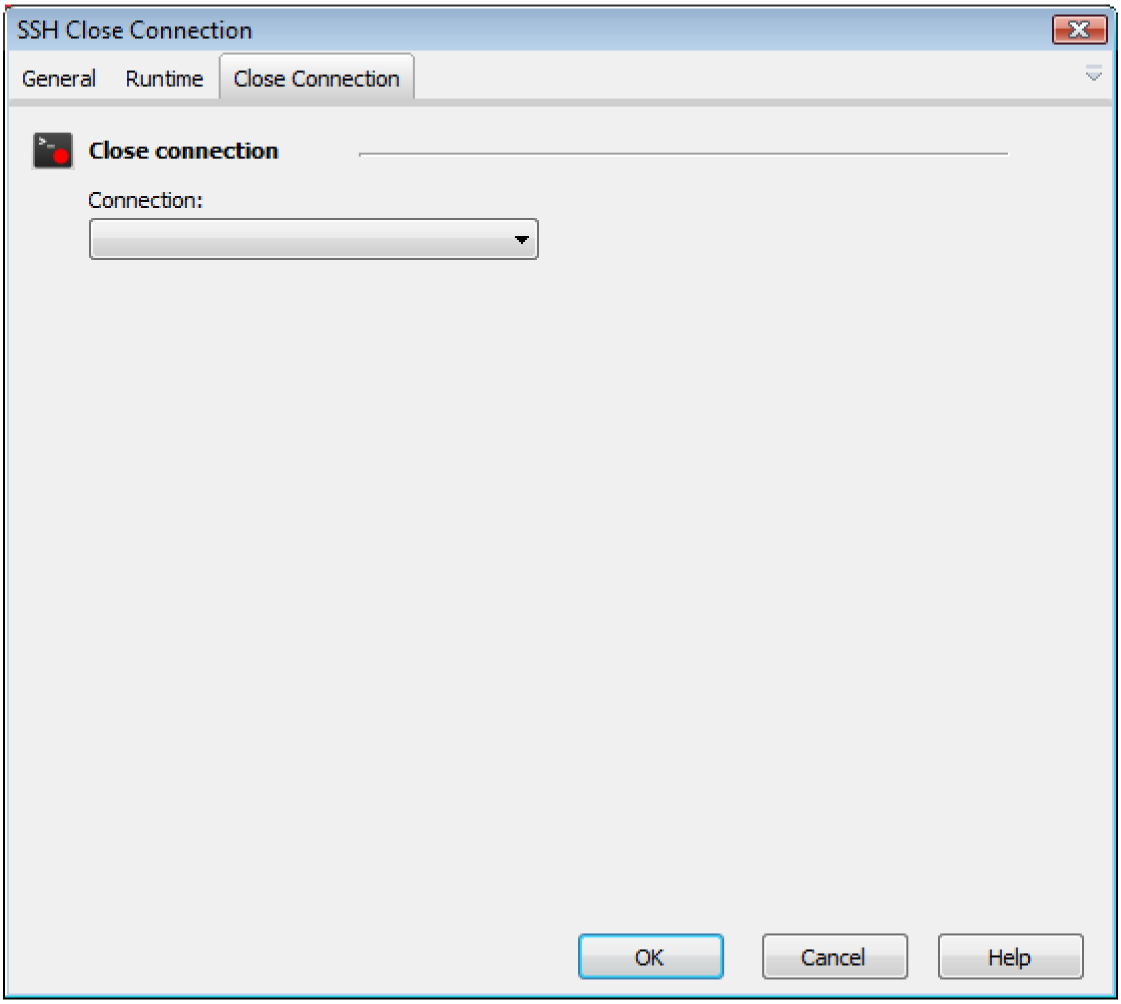
Shell Type
You must choose a shell type that is installed on the SSH server. This allows the interactive prompt to be removed from the server's responses.

Connect Timeout
The amount of time (in ms) that the action should wait for the server to connect. If set too low, the connection may fail, or the setting of the shell type may fail. If you are seeing "garbage" characters in the output from commands, try changing shell types and/or increasing the timeout.

Verbose logging
Enable verbose logging

6.15.13.2 SSH Close Connection

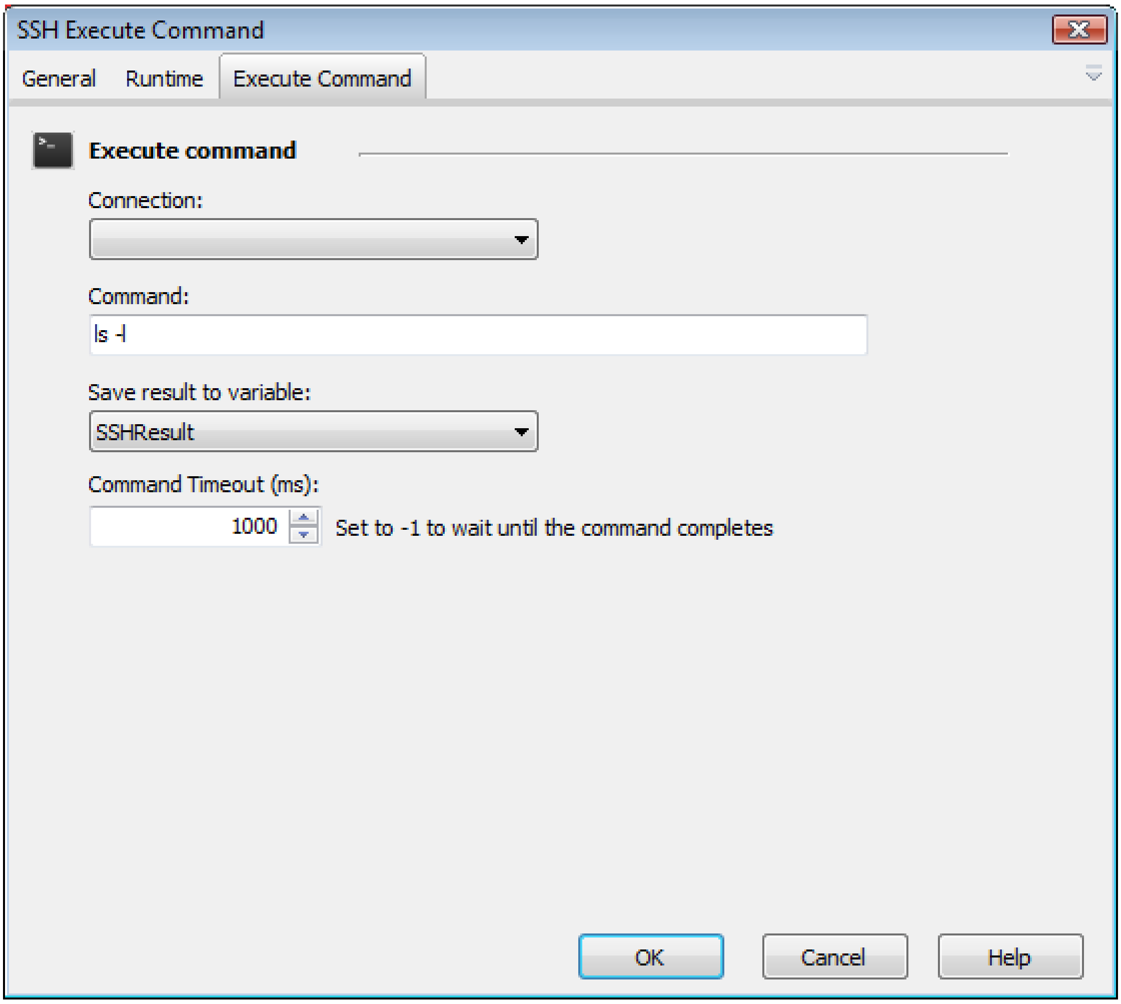
The SSH Close Connection action closes a connection opened by SSH Open Connection. If you do not use this action to close a connection, the connection will remain open until the FinalBuilder project is closed.



Connection
The connection to close. If the connection is already closed, this action will fail.

6.15.13.3 SSH Execute Commands

The SSH Execute Command action uses a connection opened with SSH Open Connection to execute a single command on the SSH server. The result of this command can optionally be stored in a variable.



Execute command

Connection:

Command: ls -l

Save result to variable: SSHResult

Command Timeout (ms): 1000 Set to -1 to wait until the command completes

Connection
The previously opened connection to use.

Command
The command to execute on the the SSH server.

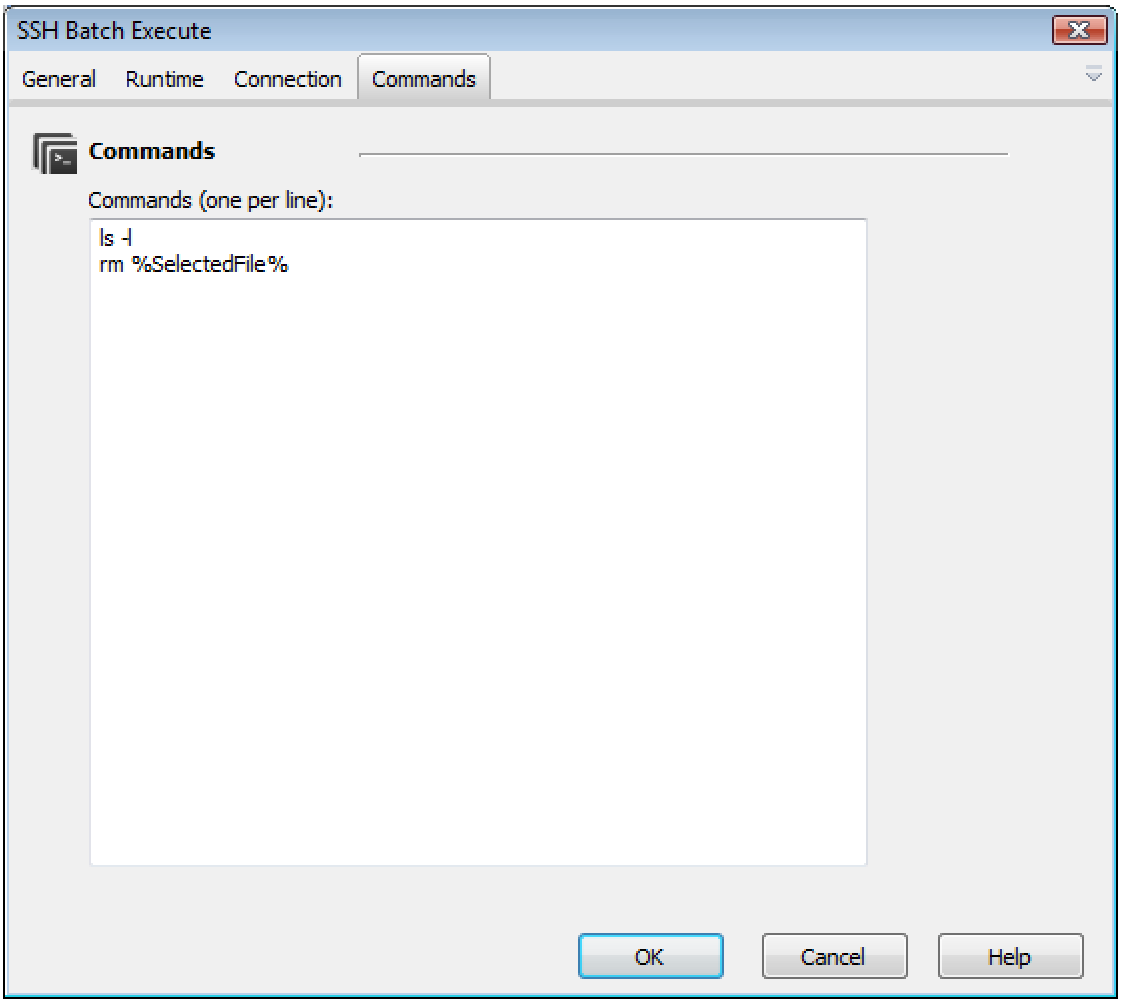
Save result to variable
Optionally save the output from the command to the specified variable.

Timeout
Specifies how long the action should wait for a response from the server. If you are receiving partial or no result when you expect one, try increasing the Timeout.

6.15.13.4 SSH Batch Execute

The SSH Batch Execute action opens a connection to an SSH server, performs multiple commands then closes the connection. This action is useful if you have multiple commands to run and do not care about the result of each. If you need to validate the results of individual commands, use SSH Open Connection, SSH Execute Command and SSH Close Connection.

See SSH Open Connection for how to configure the connection.



Commands (one per line):

```
ls -l
rm %SelectedFile%
```

Commands
A list of commands to be run on the SSH server, in order. Results from the commands are written to the build log. Commands are executed in the same session, so you can (for example) change directories etc.

6.15.14 SFTP

The SFTP actions in Automise provide client-side functionality for SFTP (Secure File Transfer Protocol).

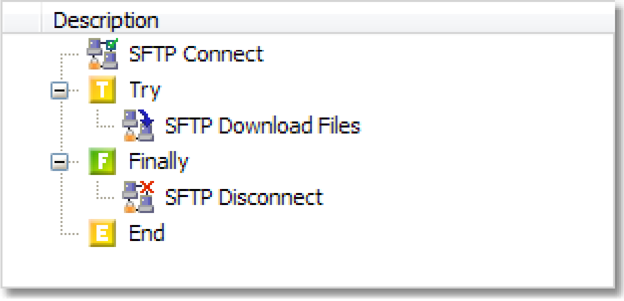
The actions implement SSH File Transfer Protocol. It works over secure SSH channel and has nothing common with FTP protocol. SFTP works as a subsystem of SSHv2 protocol, i.e. SFTP negotiation flow is the following:

- TCP connection is established
- Secure SSH connection is established
- SFTP handshake is performed over SSH channel

To add SFTP support to your project you should perform the following sequence of operations:

- Add a SFTP Connect action (specify address/port, authentication parameters etc.)
- Add TRY, FINALLY, and END actions.
- Under the TRY action, add SFTP actions for the tasks that you need to perform (eg. SFTP Download File, SFTP List Files, etc)
- Under the FINALLY action, add an SFTP Disconnect action

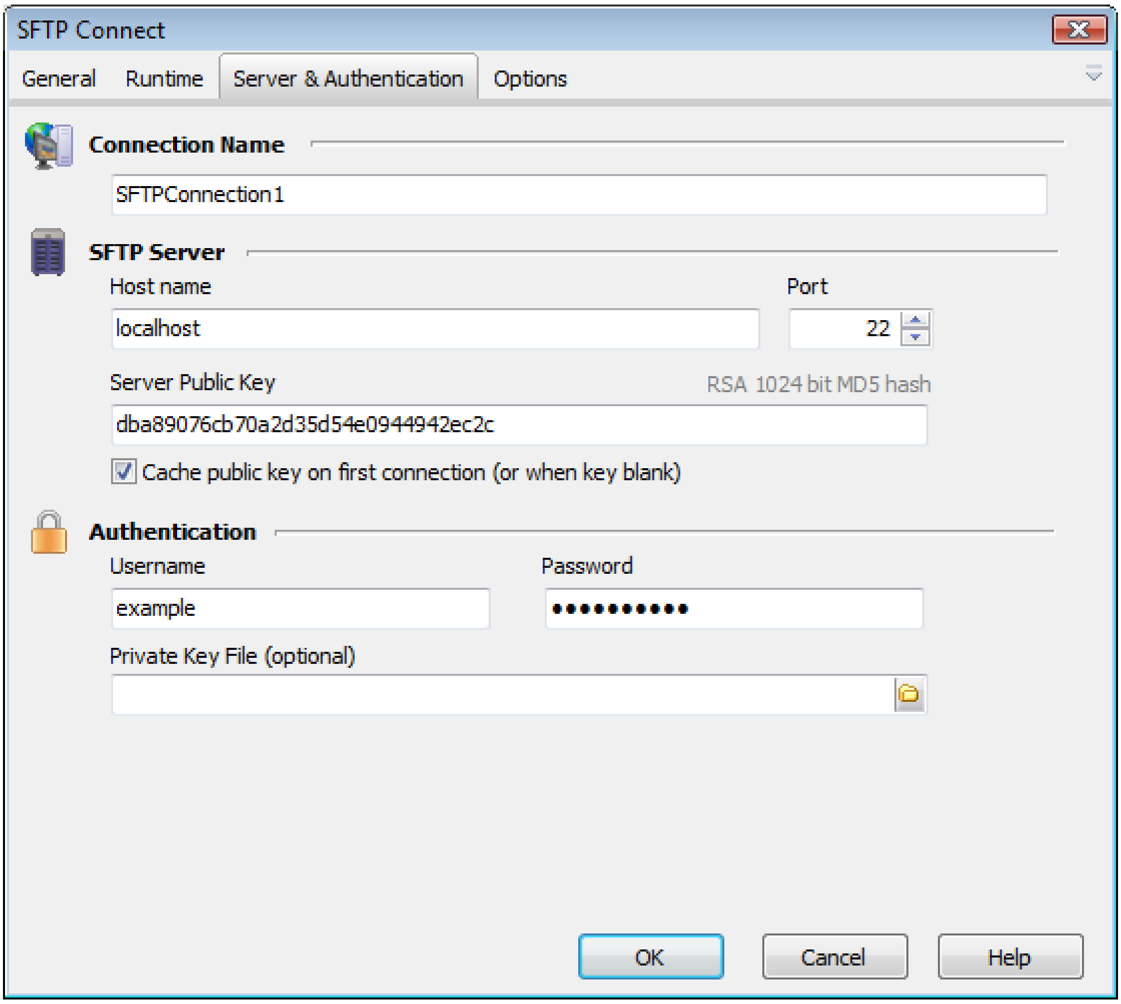
A simple example should look something like this:



Description
- SFTP Connect
- Try
  - SFTP Download Files
- Finally
  - SFTP Disconnect
- End

6.15.14.1 SFTP Connect

Use the SFTP Connect action to create a connection to an SFTP server which you can then use with other SFTP actions.



SFTP Connect

Connection Name

SFTPConnection1

SFTP Server

Host name: localhost

Port: 22

Server Public Key: dba89076cb70a2d35d54e0944942ec2c

RSA 1024 bit MD5 hash

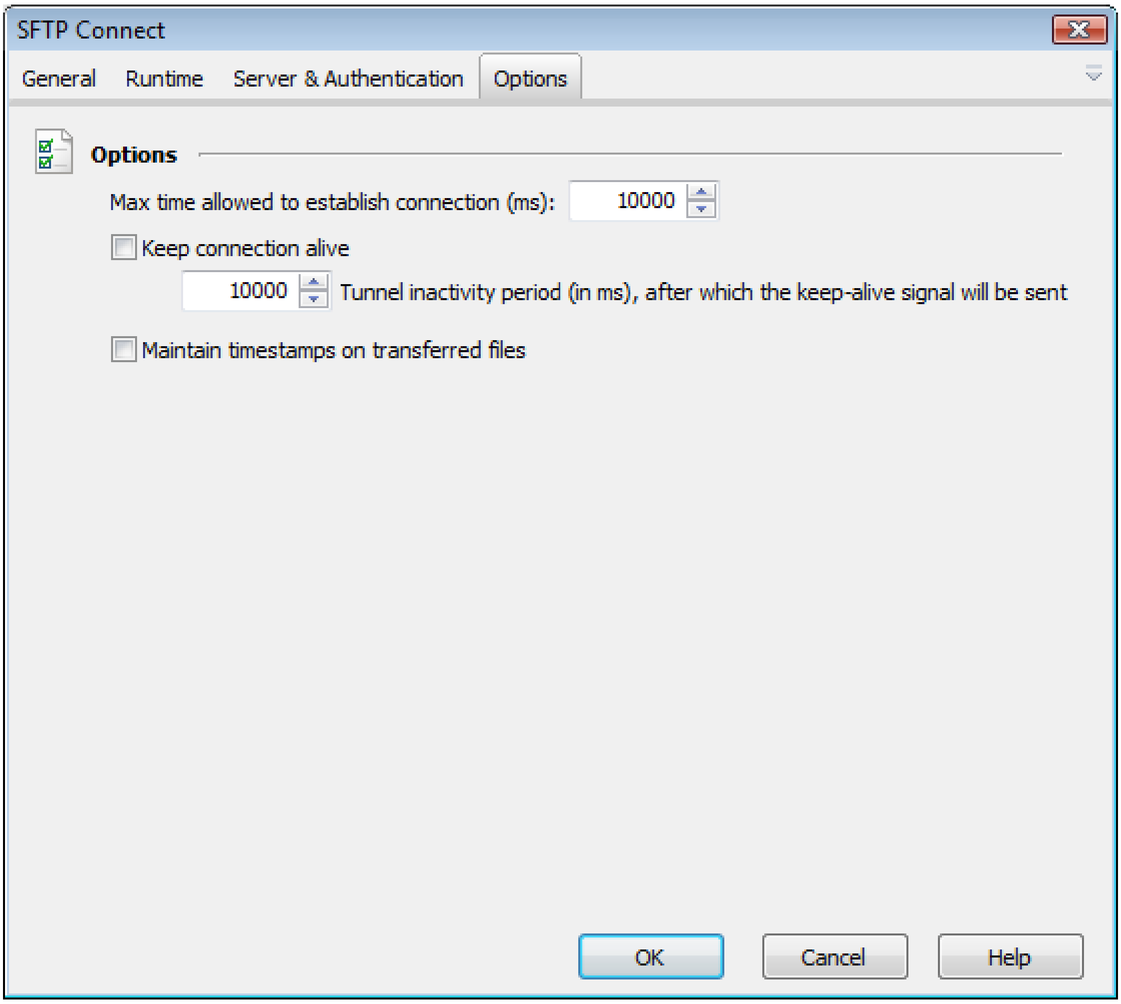
☑ Cache public key on first connection (or when key blank)

Authentication

Username: example

Password: ●●●●●●●●●●●

Private Key File (optional):



Options

Max time allowed to establish connection (ms): 10000

□ Keep connection alive

10000 Tunnel inactivity period (in ms), after which the keep-alive signal will be sent

□ Maintain timestamps on transferred files

Connection Name

Enter the name for the connection. This is the name you need to choose in the other SFTP actions that will use this connection.

SFTP Server

Host name

The hostname of the SFTP server to connect to.

Port

The port of the SFTP server to connect to. Usually port 22.

Server Public Key & Cache public key

This is the RSA 1024 bit MD5 hash of the server key. For SFTP, as well as using a secure transport, it validates the identity of the server against a known key. You may either enter this key, or by turning on the Cache checkbox the key will automatically be saved in this field if the field is blank (and then used on subsequent connections to validate the server).

Keep connection alive

Turning this option on will automatically keep the connection alive by sending data to the SFTP host server at intervals specified by the Tunnel inactivity period.

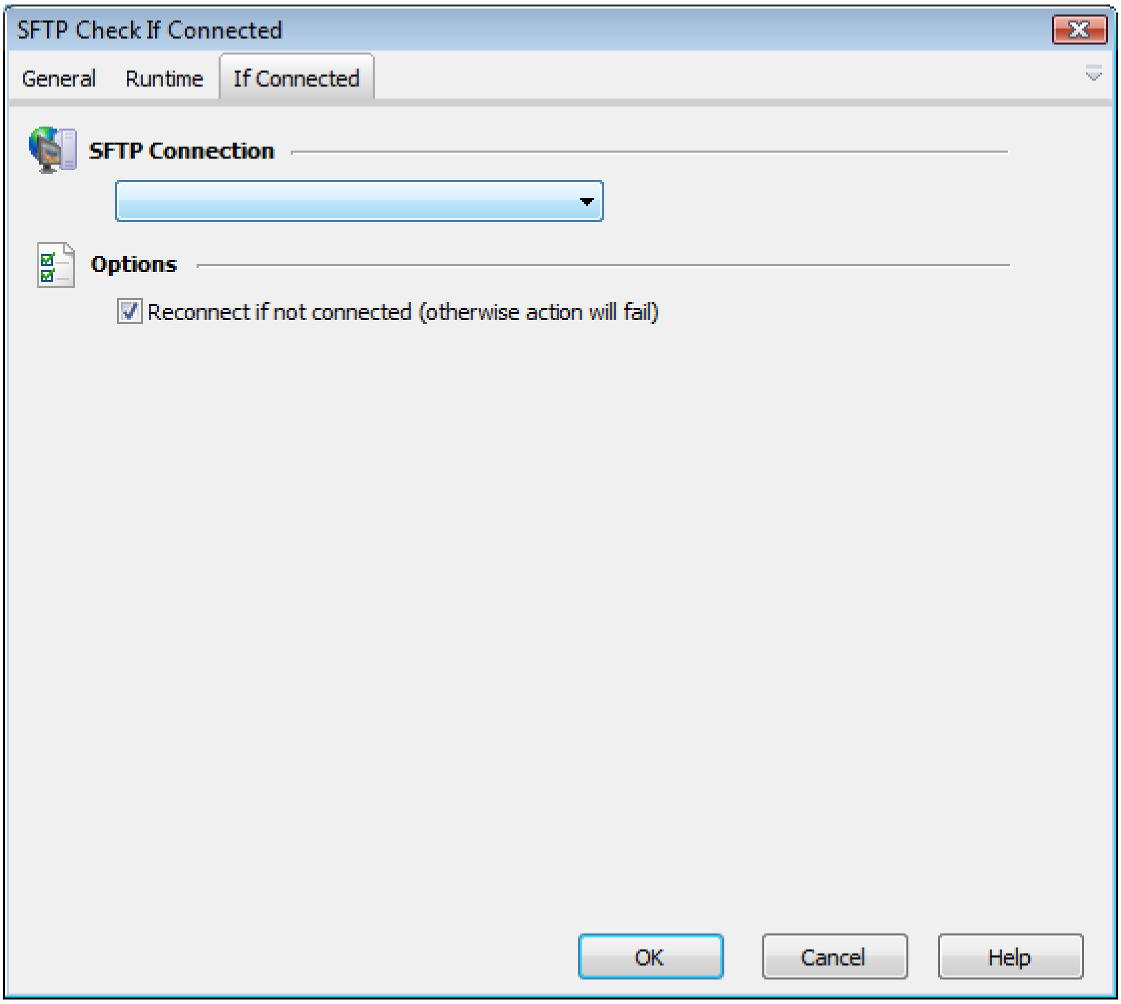
6.15.14.2 SFTP Disconnect

Simply specify the SFTP connection in the SFTP Disconnect action property page and when this action runs it will perform a disconnect from the SFTP server.

6.15.14.3 SFTP Check if Connected

Use the SFTP Check if Connected action to validate if the connection to the SFTP server is still valid.

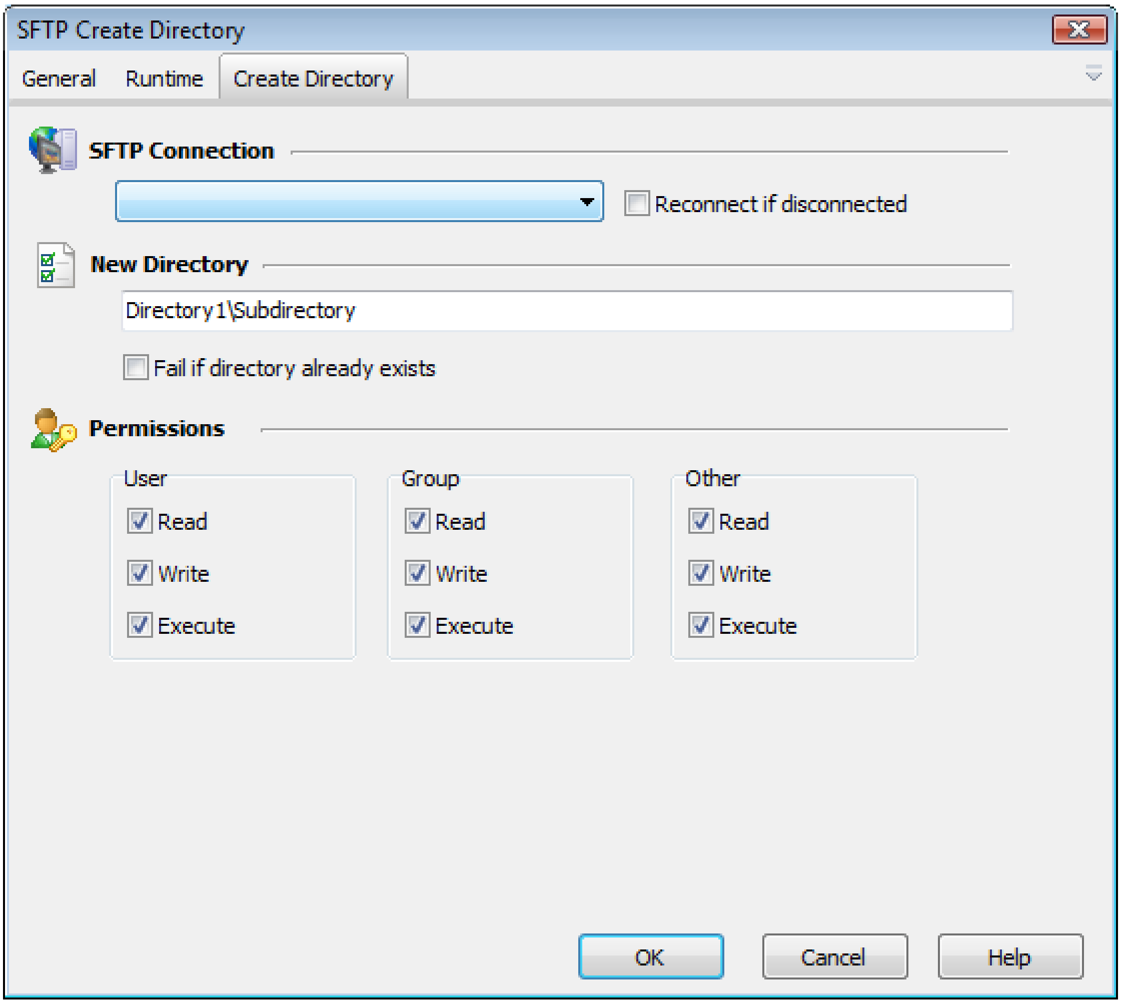
Optionally reconnect to the SFTP server (using the details from the SFTP Connect action) if the connection has been lost.



☑ Reconnect if not connected (otherwise action will fail)

6.15.14.4 SFTP Create Directory

Use the SFTP Create Directory action to create a directory on the remote server.



SFTP Connection

□ Reconnect if disconnected

New Directory

Directory1\Subdirectory

□ Fail if directory already exists

Permissions

User: ☑ Read ☑ Write ☑ Execute

Group: ☑ Read ☑ Write ☑ Execute

Other: ☑ Read ☑ Write ☑ Execute

New Directory

Specify the new directory name. Make sure the directory is a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does. Creating a directory tree (ie. more than one directory specified doesn't exist, is server dependent - if not supported by the server you will likely get a "permission denied" error).

Fail if directory already exists

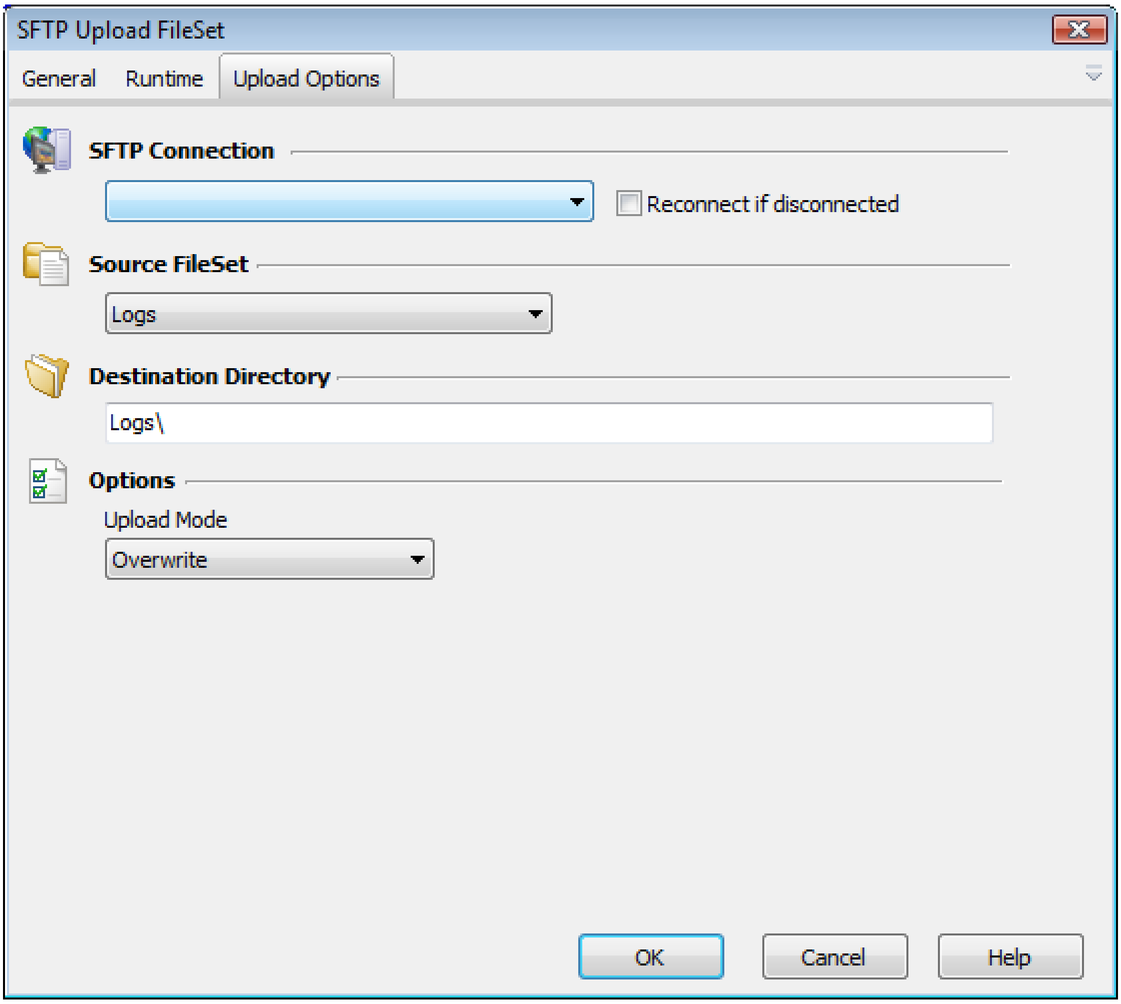
Optionally fail the action if the directory to create already exists on the server.

Permissions

Specify the permissions that the new directory should have.

6.15.14.5 SFTP Upload FileSet

Use the SFTP Upload FileSet action to upload all the files in an existing FileSet.



SFTP Upload FileSet

SFTP Connection

□ Reconnect if disconnected

Source FileSet: Logs

Destination Directory: Logs\

Options

Upload Mode: Overwrite

Source FileSet

Select the FileSet that contains the files that you want to transfer.

Destination Directory

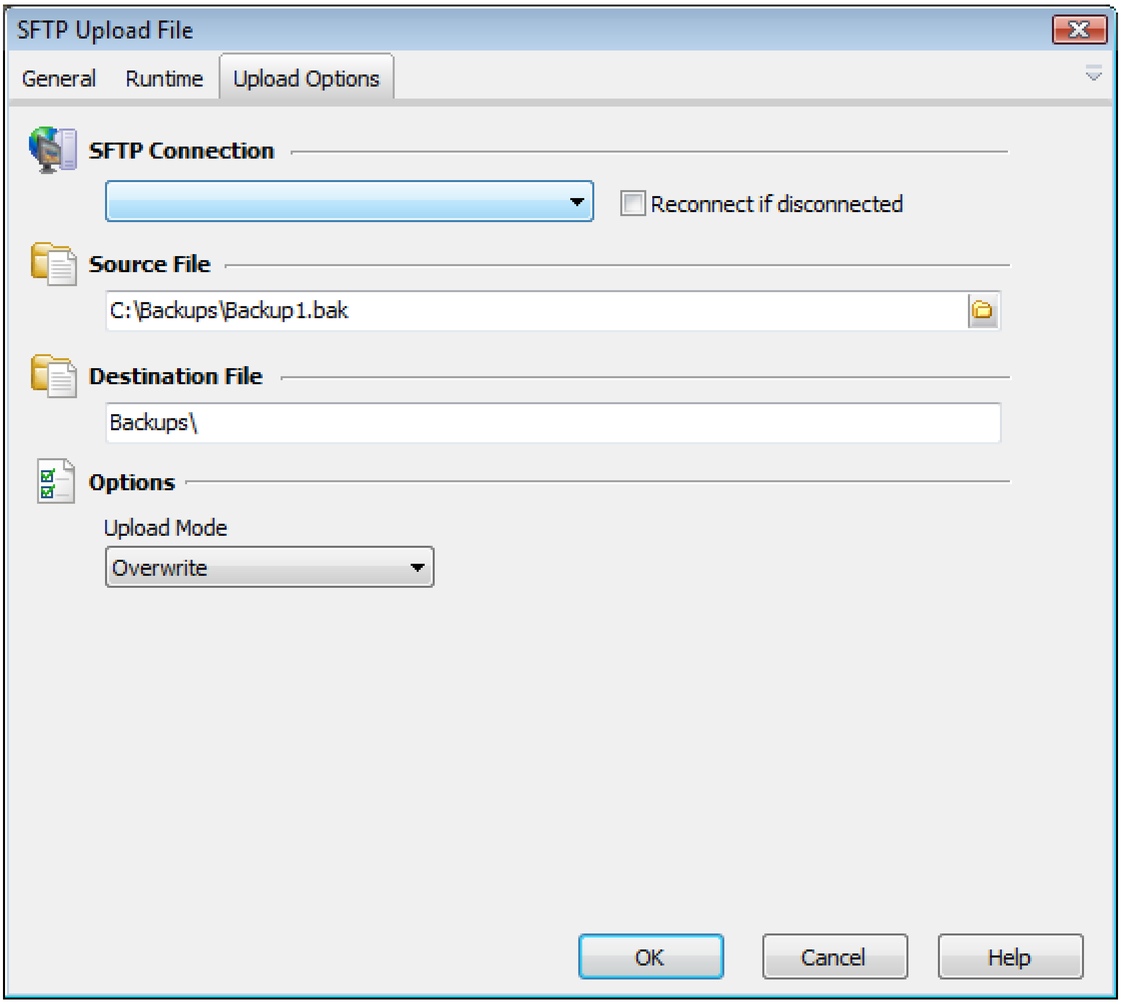
Specify the name of the directory on the SFTP server. You may specify a directory that doesn't exist and it will be created for you. Make sure the directory is a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Upload Mode

Select what should be done if a file already exists on the server with the same name. Select either: Overwrite, Skip, Append, or Resume.

6.15.14.6 SFTP Upload File

Use the SFTP Upload File action to upload a single file to the SFTP server.



Source File

Select the file that you want to transfer. This action only supports transferring a single file. To transfer multiple files use either SFTP Upload Files or SFTP Upload FileSet.

Destination File

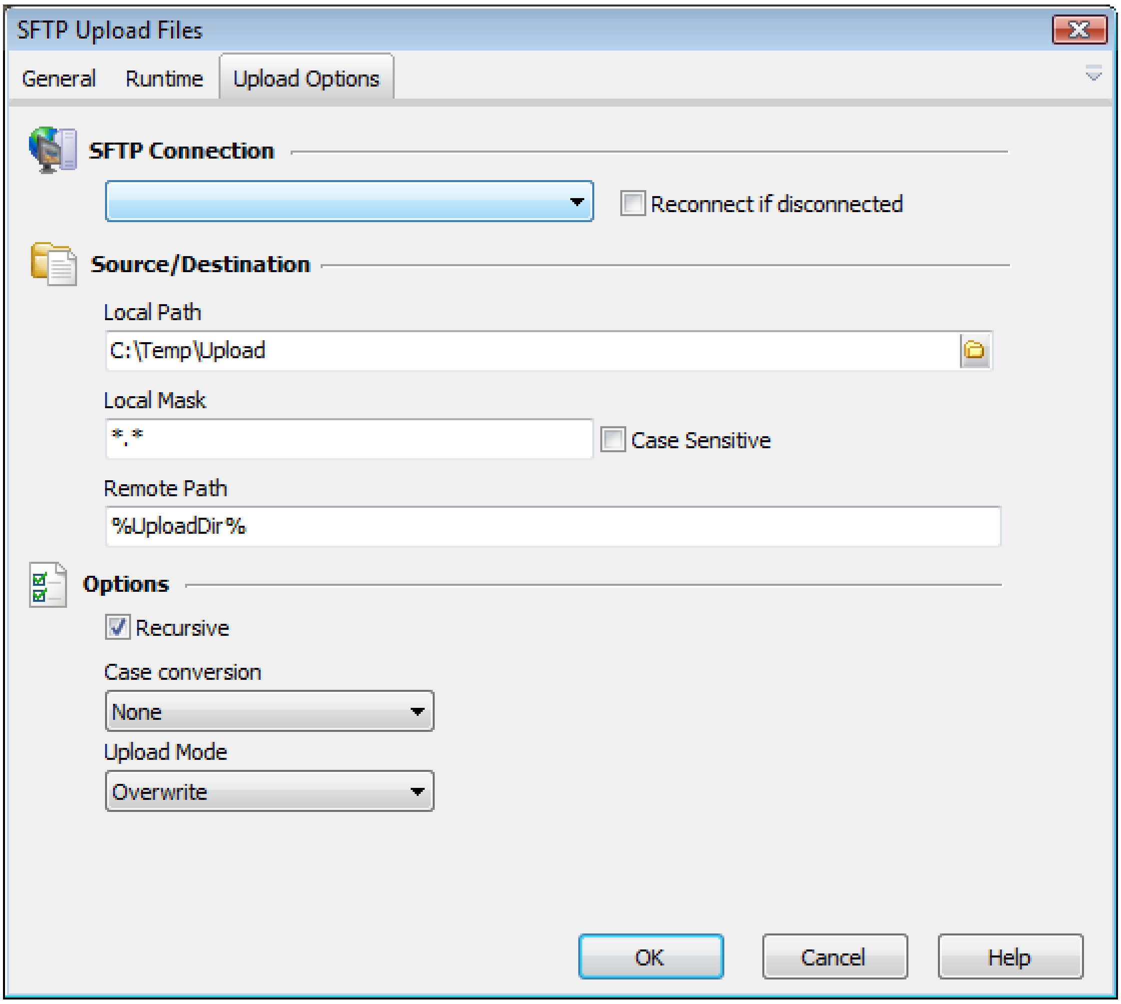
Specify the destination directory or destination directory including the filename on the SFTP server. You may specify a directory that doesn't exist and it will be created for you. Make sure the directory is a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does. If specifying a filename, then the file will be transferred as the new filename.

Upload Mode

Select what should be done if a file already exists on the server with the same name. Select either: Overwrite, Skip, Append, or Resume.

6.15.14.7 SFTP Upload Files

Use the SFTP Upload Files action to upload files that match a mask to the SFTP server. To transfer a single file it's easier to use the SFTP Upload File action. An alternative to the SFTP Upload Files action is the SFTP Upload FileSet action as it has more flexibility in choosing which files and the same set of files can be reused with other actions.



SFTP Connection

□ Reconnect if disconnected

Source/Destination

Local Path: C:\Temp\Upload

Local Mask: *.*

□ Case Sensitive

Remote Path: %UploadDir%

Options

☑ Recursive

Case conversion: None

Upload Mode: Overwrite

Source/Destination

Local Path - Specify the local path for the source files.
Local Mask - Specify the mask to test the local files against.
Case Sensitive - Turning this on makes the local mask case sensitive.

Remote Path - Specify the destination directory. You may specify a directory that doesn't exist and it will be created for you. Make sure the directory is a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Recursive

To recursively upload files from child directories turn this flag on.

Case Conversion

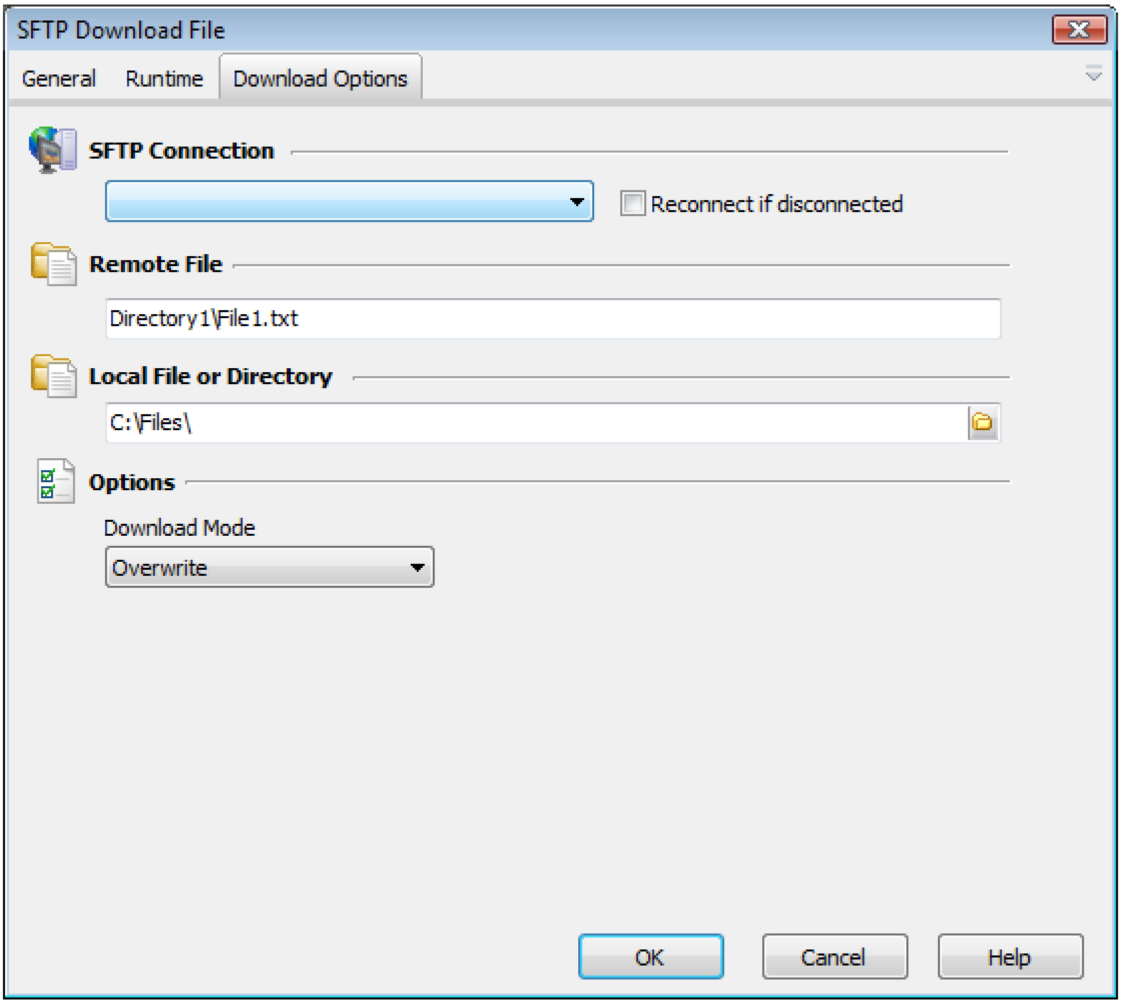
To convert the filenames to either lowercase or uppercase choose the corresponding option, otherwise choose "None" to leave the filenames as-is.

Upload Mode

Select what should be done if a file already exists on the server with the same name. Select either: Overwrite, Skip, Append, or Resume.

6.15.14.8 SFTP Download File

Use the SFTP Download File action to download a single file from the SFTP server.



SFTP Download File

General Runtime Download Options

SFTP Connection

□ Reconnect if disconnected

Remote File

Directory1\File1.txt

Local File or Directory

C:\Files\

Options

Download Mode

Overwrite

Remote File

Enter the name of the file on the remote server that you want to transfer. Make sure the full path is included (specified from the user's home directory) as the SFTP protocol doesn't have the concept of a "current directory" like FTP does. This action only supports transferring a single file. To transfer multiple files use SFTP Download Files.

Local File or Directory

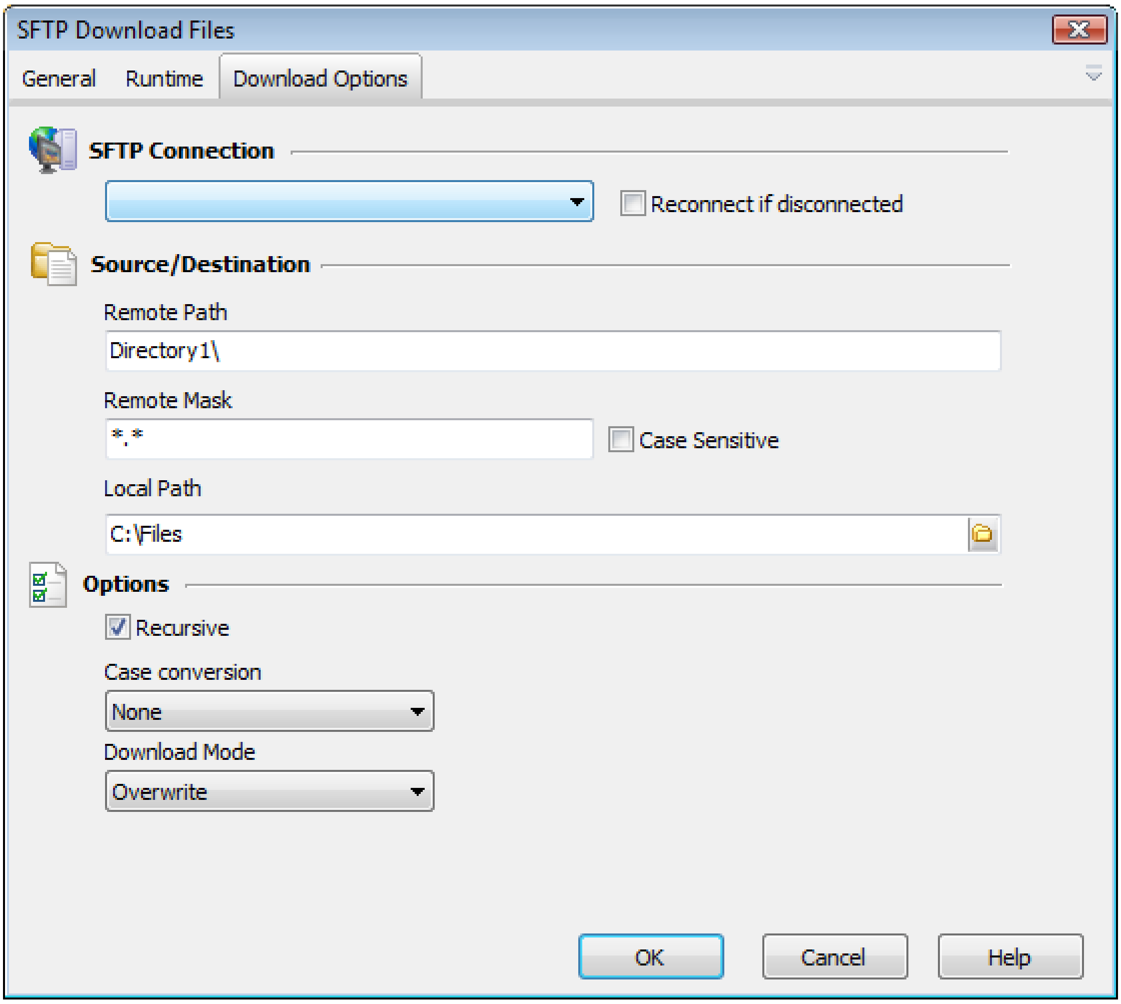
Specify either a local filename (including path) or a local directory. If specifying the local filename, then the file will be transferred and renamed to this new name.

Download Mode

Select what should be done if a file already exists on the local machine with the same name. Select either: Overwrite, Skip, Append, or Resume.

6.15.14.9 SFTP Download Files

Use the SFTP Download Files action to download multiple files SFTP server. To download a single file, it's easier to use the SFTP Download File action.



SFTP Download Files

□ Reconnect if disconnected

Remote Path: Directory1\

Remote Mask: *.*

□ Case Sensitive

Local Path: C:\Files

☑ Recursive

Case conversion: None

Download Mode: Overwrite

Remote Path

Specify the remote path on the SFTP Server. The path must be specified from the user's home directory as the SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Remote Mask

Specify a file mask to select the files on the remote server. Use the case sensitive option to only match files that are in the same case as the file mask.

Local Path

Specify a local directory where the files will be downloaded to.

Recursive

To recursively download files from child directories turn this flag on.

Case Conversion

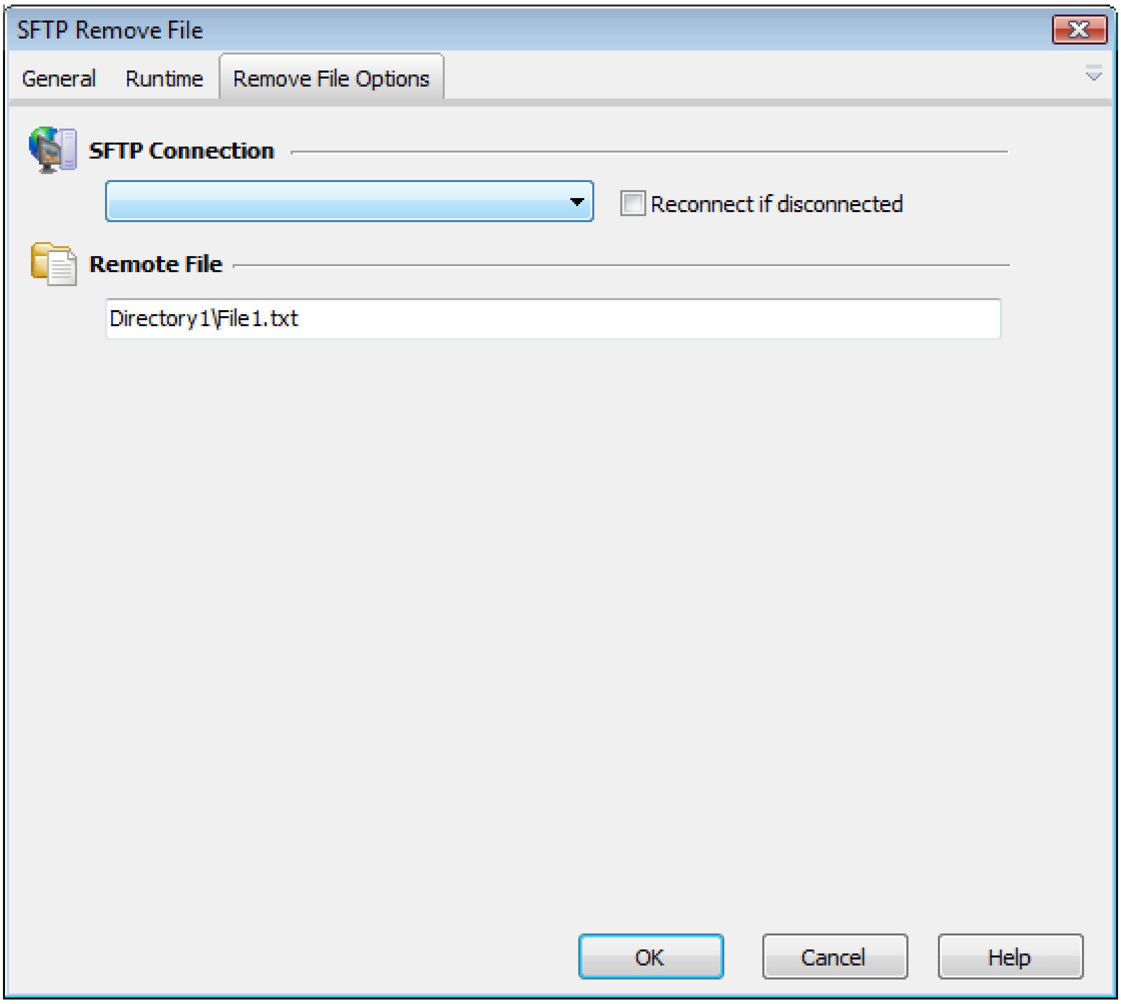
To convert the filenames to either lowercase or uppercase choose the corresponding option, otherwise choose "None" to leave the filenames as-is.

Download Mode

Select what should be done if a file already exists on the local machine with the same name. Select either: Overwrite, Skip, Append, or Resume.

6.15.14.10 SFTP Remove File

Use the SFTP Remove File action to delete a single file from the SFTP server. To delete multiple files use the SFTP Remove Files action. To remove an entire directory (or directory tree) use the SFTP Remove Directory action.



SFTP Remove File

General | Runtime | Remove File Options

SFTP Connection

□ Reconnect if disconnected

Remote File

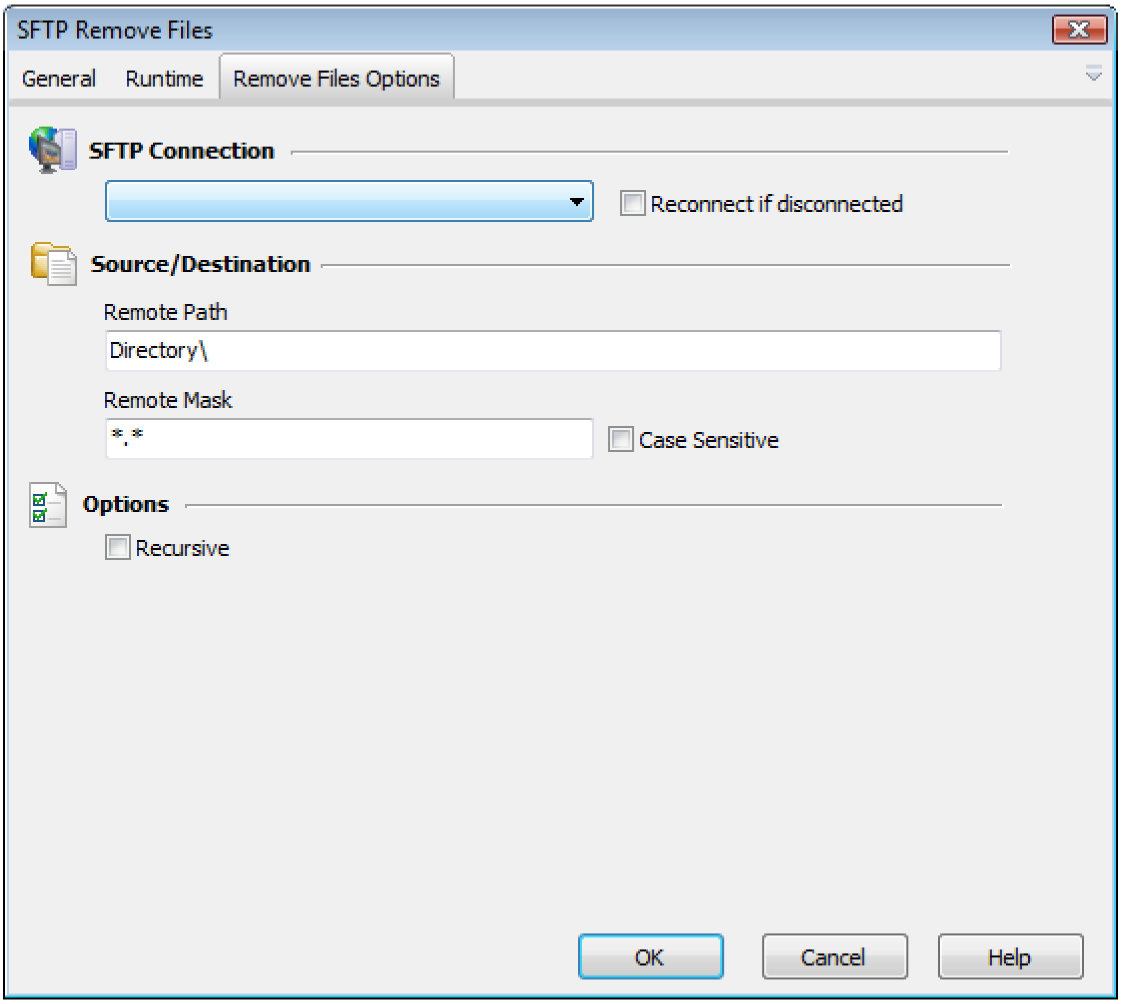
Directory1\File1.txt

Remote File

Enter the name of the file on the remote server that you want to transfer. Make sure the full path is included (specified from the user's home directory) as the SFTP protocol doesn't have the concept of a "current directory" like FTP does. This action only supports deleting a single file. To delete multiple files and directories use SFTP Remote Files.

6.15.14.1: SFTP Remove Files

Use the SFTP Remove Files action to delete a multiple files from the SFTP server. To delete a single file it's easier to use SFTP Remove File. To remove an entire directory (or directory tree) use the SFTP Remove Directory action.



□ Reconnect if disconnected

Source/Destination

Remote Path: Directory\

Remote Mask: *.*

□ Case Sensitive

Options

□ Recursive

Remote Path

Enter the path on the remote server. Make sure the full path is included (specified from the user's home directory) as the SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Remote Mask

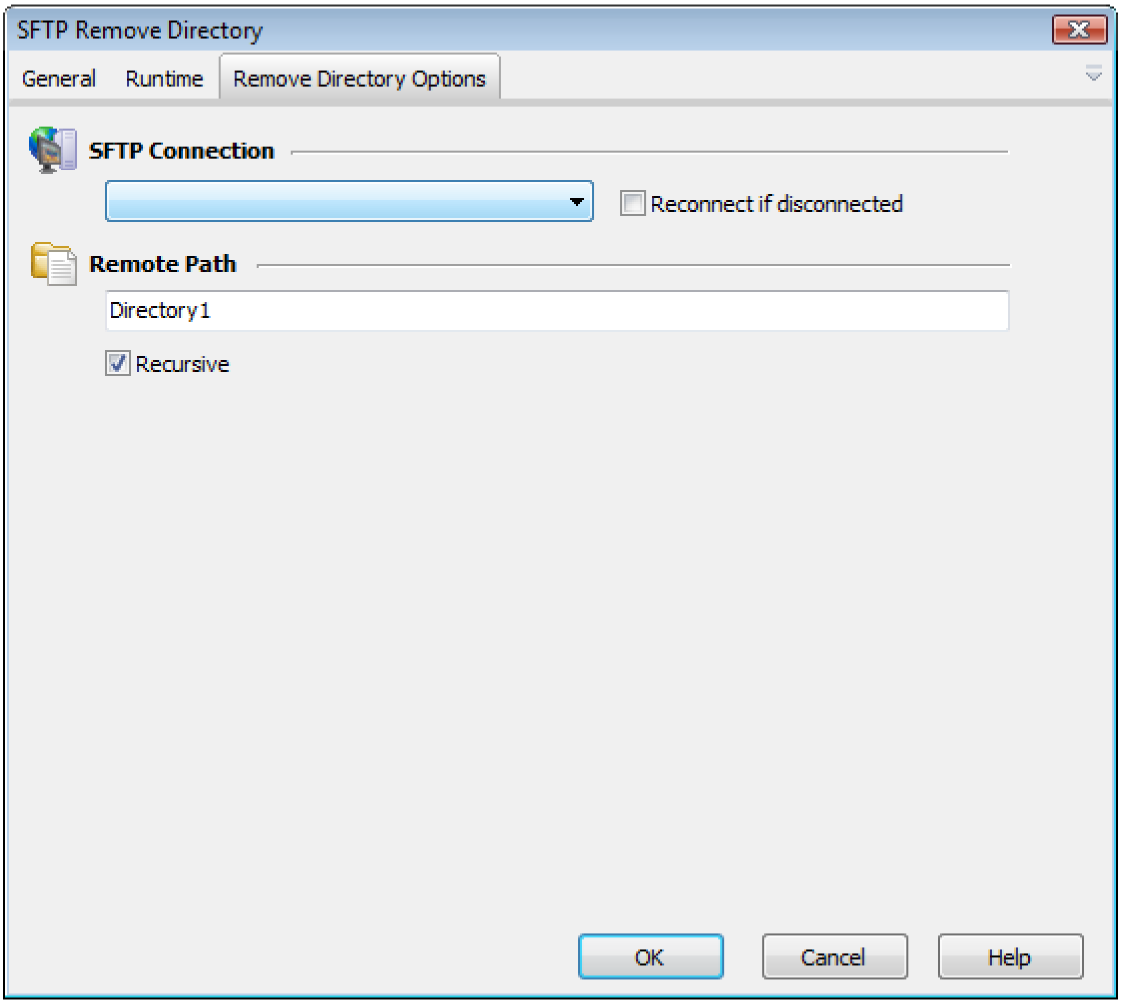
Enter the remote mask to match against the files and directories to delete. Case Sensitive specifies whether to treat filenames case-sensitively.

Recursive

Specify the recursive option to delete files in child directories also.

6.15.14.1:SFTP Remove Directory

Use the SFTP Remove Directories action to delete a directory from the SFTP server. To delete a single file use SFTP Remove File, or to delete multiple files use the SFTP Remove Files action.



SFTP Connection

☐ Reconnect if disconnected

Remote Path

Directory1

☑ Recursive

Remote Path

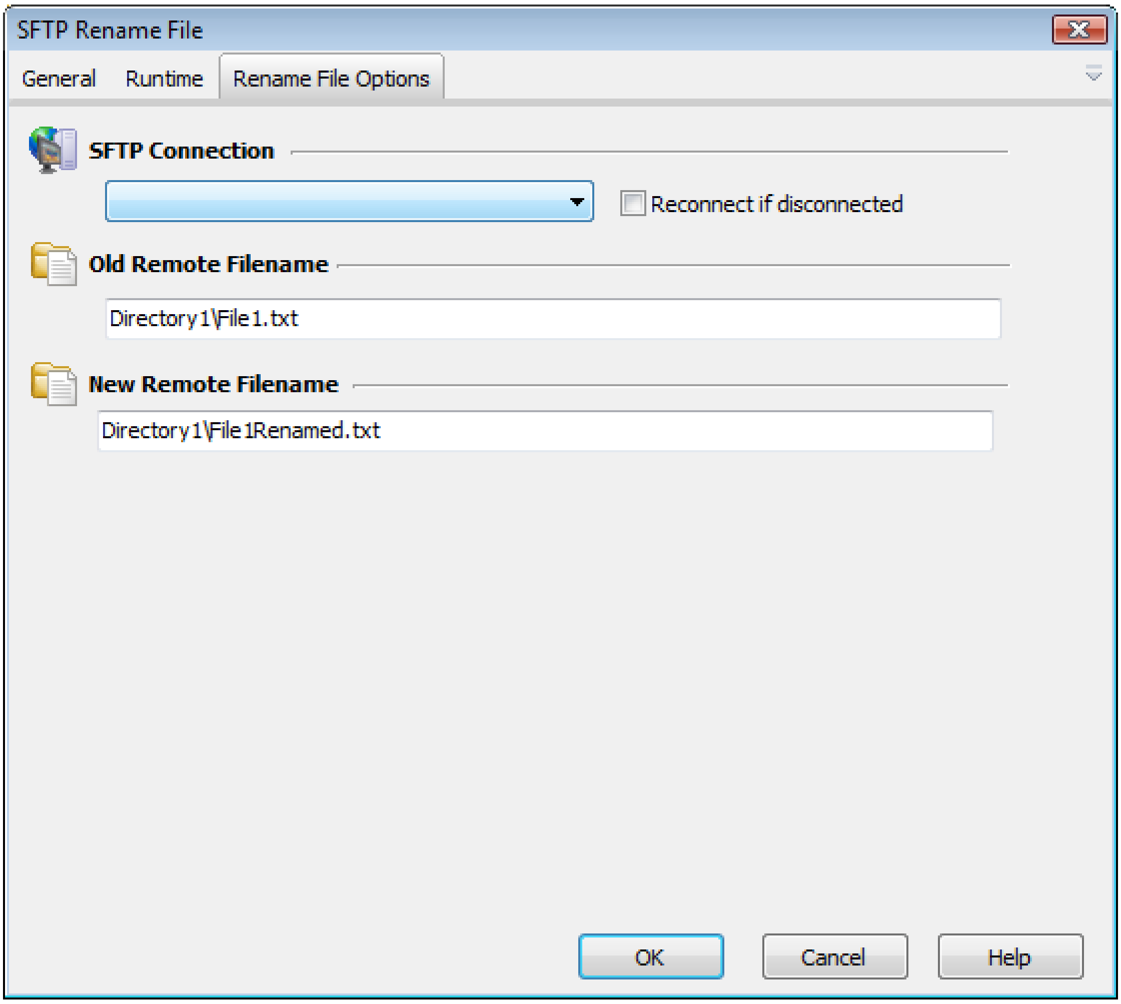
Enter the path to the directory to be removed on the remote server. Make sure the full path is included (specified from the user's home directory) as the SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Recursive

Specifies whether to remove directories recursively, i.e. including subdirectories

6.15.14.1:SFTP Rename File

Use the SFTP Rename File action to change the name of a file on the SFTP server.



□ Reconnect if disconnected

Old Remote Filename

Directory1\File1.txt

New Remote Filename

Directory1\File1Renamed.txt

Old Remote Filename

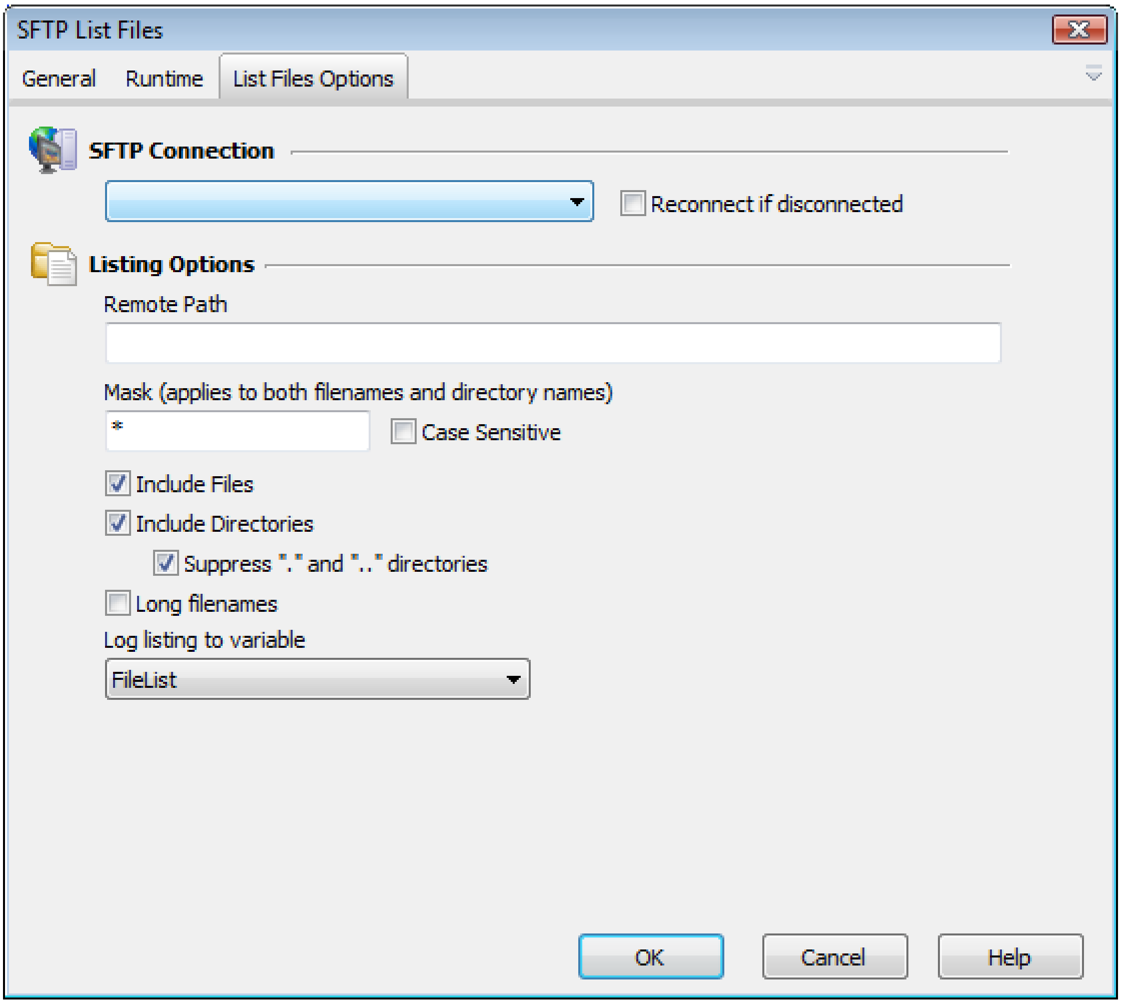
Path to existing file on the remote SFTP server.

New Remote Filename

Specifies the new name/location of the file on the remote SFTP server.

6.15.14.14 SFTP List Files

Use the SFTP List Files action to list the files and directories on the SFTP server.



SFTP Connection

☐ Reconnect if disconnected

Listing Options

Remote Path

Mask (applies to both filenames and directory names): *

☐ Case Sensitive

☑ Include Files

☑ Include Directories

☑ Suppress "." and ".." directories

☐ Long filenames

Log listing to variable: FileList

Remote path

Path to the directory to read on the remote SFTP server.

Mask

Specifies the mask (wildcards) to select the names. Only the names of the files and directories, that match the mask, will be returned. Note, that SFTP doesn't support name masks, so no matter what mask you specify, the whole directory will be read and then scanned for matches.

Case Sensitive

Specifies whether the mask is case-sensitive

Include Files

Specifies whether the names of the files and symlinks is returned

Include Directories

Specifies whether the names of the directories is returned

Suppress "." and ".." directories

The special directories "." and ".." will be suppressed from the listing.

Long filenames

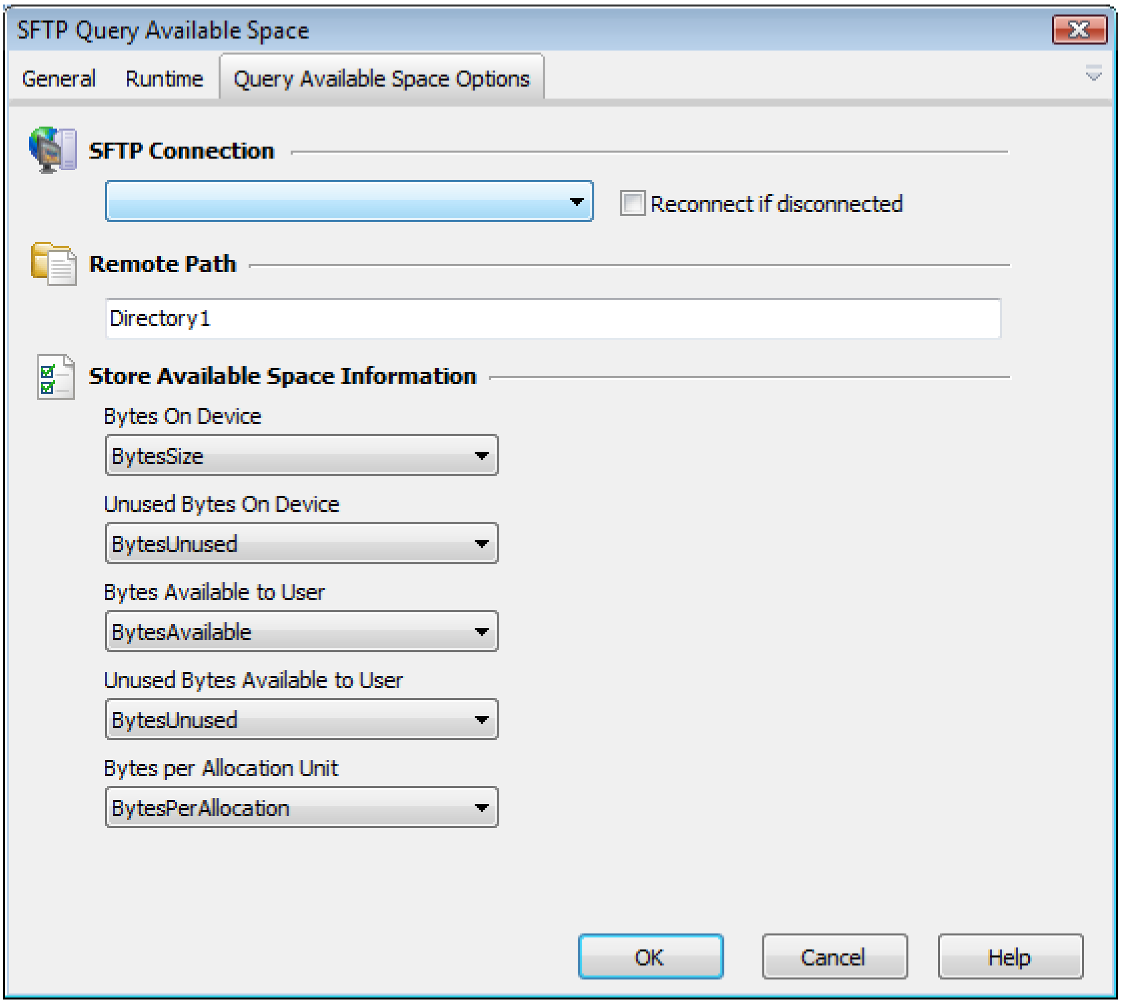
Specify to return a long listing. The long listing includes the permissions, size of file, ownership and other properties.

Log listing to variable

Specify a variable name to copy the listing into.

6.15.14.1!SFTP Query Available Space

Use the SFTP Query Available Space action to change the name of a file on the SFTP server.



SFTP Query Available Space

SFTP Connection

□ Reconnect if disconnected

Remote Path

Directory1

Store Available Space Information

Bytes On Device: BytesSize

Unused Bytes On Device: BytesUnused

Bytes Available to User: BytesAvailable

Unused Bytes Available to User: BytesUnused

Bytes per Allocation Unit: BytesPerAllocation

Remote path

Path to the desired directory

Bytes on Device

Specifies total available space on a device in bytes. Specify the variable to store the information in.

Unused Bytes on Device

Specifies unused space on a device in bytes. Specify the variable to store the information in.

Bytes available to user

Specifies total space available to the current user in bytes. Specify the variable to store the information in.

Unused Bytes available to User

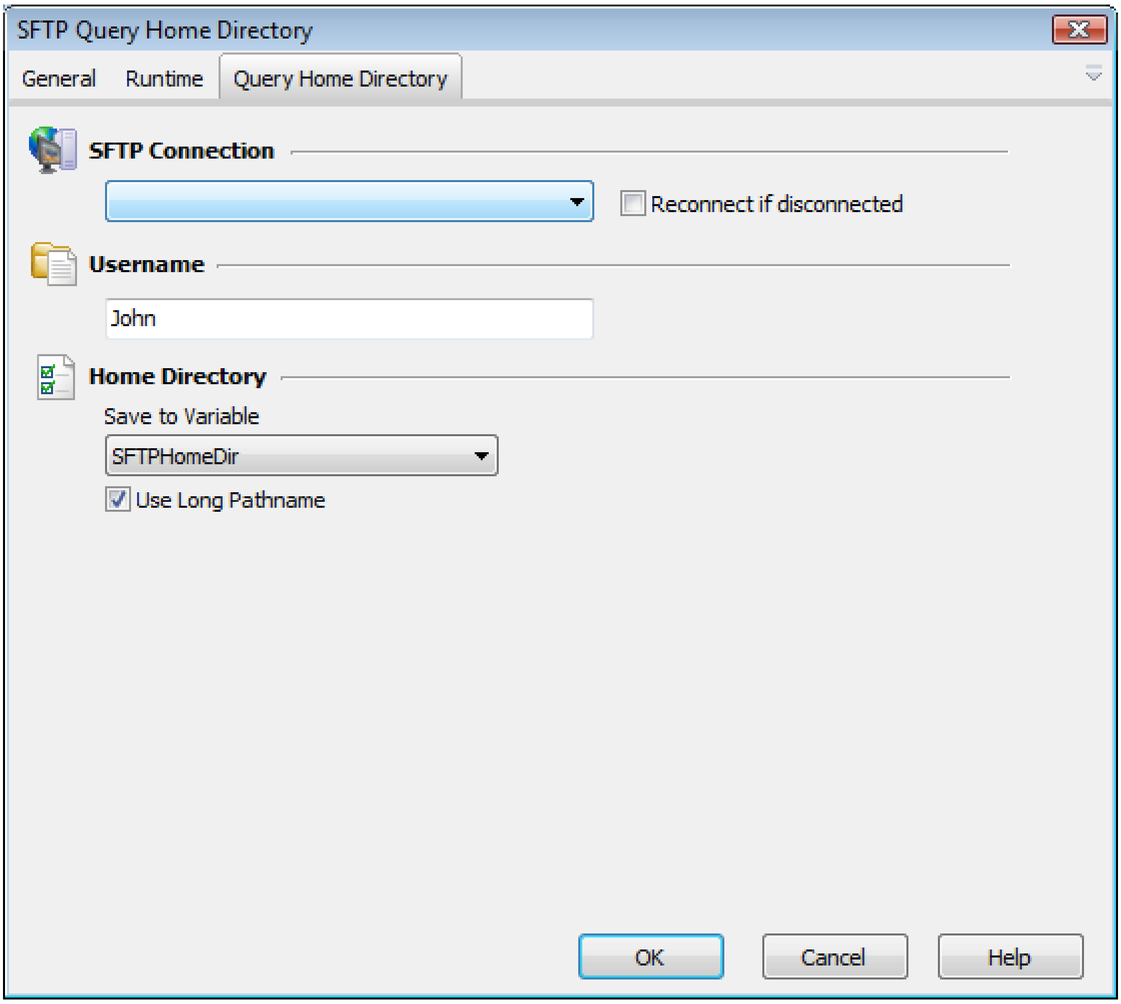
Specifies unused space available to the current user in bytes. Specify the variable to store the information in.

Bytes per Allocation Unit

Specifies number of bytes per allocation unit on the device. Specify the variable to store the information in.

6.15.14.10 SFTP Query Home Directory

Use the SFTP Query Home Directory action to request the path to the user's home directory.



SFTP Connection

☐ Reconnect if disconnected

Username

John

Home Directory

Save to Variable

SFTPHomeDir

☑ Use Long Pathname

Username

Specify the username for which the home directory is requested

Save to variable

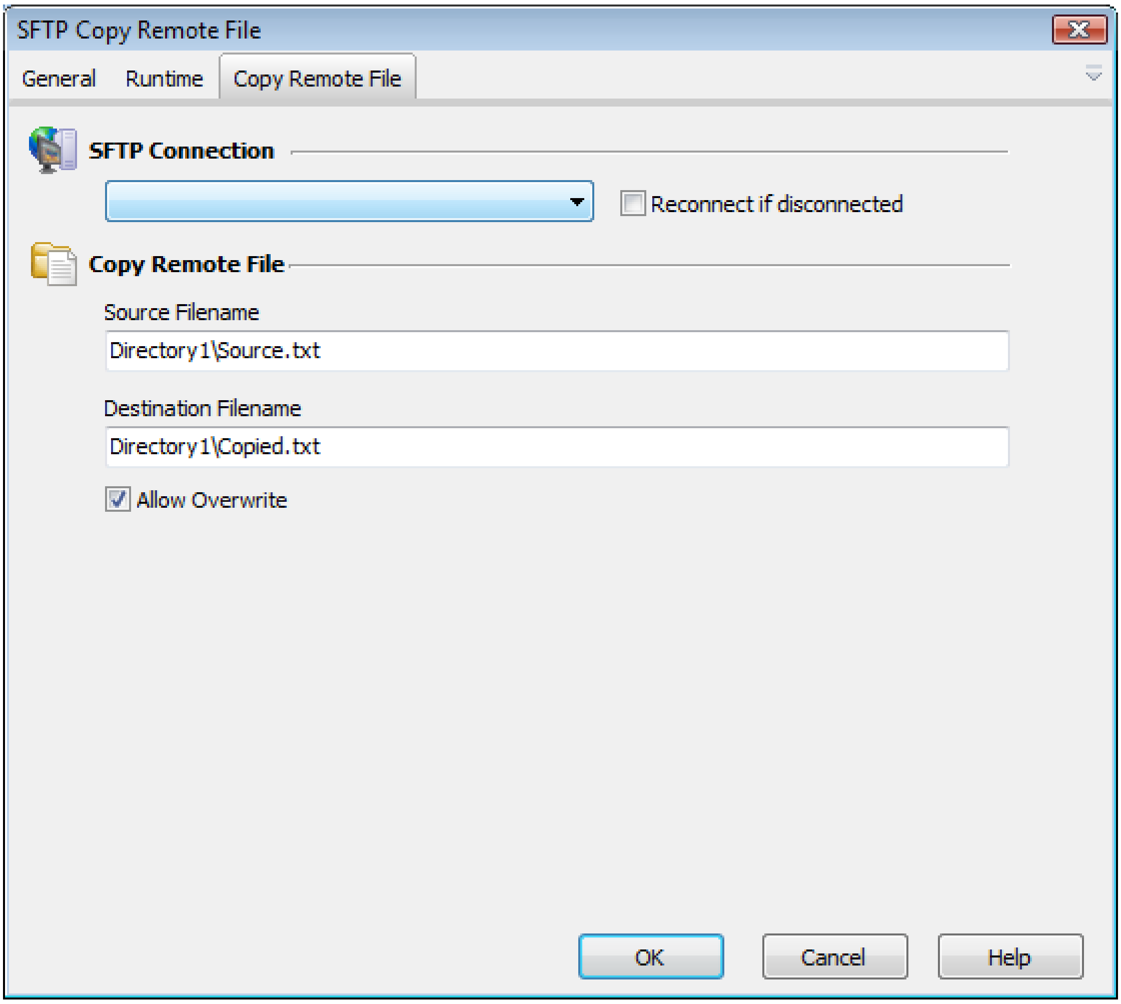
Specify a variable to save the home directory information into.

Use Long Pathname

Specify to return a long listing. The long listing includes the permissions, size of file, ownership and other properties.

6.15.14.1:SFTP Copy Remote File

Use the SFTP Copy Remote File action to copy a remote file into a new location.



□ Reconnect if disconnected

Source Filename: Directory1\Source.txt

Destination Filename: Directory1\Copied.txt

☑ Allow Overwrite

Source Filename

Specify the file to be copied on the remote SFTP server. Make sure the filename includes a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Destination Filename

Specify where the file should be copied to on the remote SFTP server. Make sure the filename includes a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Allow Overwrite

Specify whether to overwrite file with the same name if it already exists.

6.15.15 FTPS Actions

The FTPS (FTP over SSL) actions allow you to access your FTP server using SSL certificate authentication.

Use the FTPS Connect action to establish a connection to the FTPS server. Once a connection has been established, this connection can be used by the remaining FTPS actions to perform various operations.

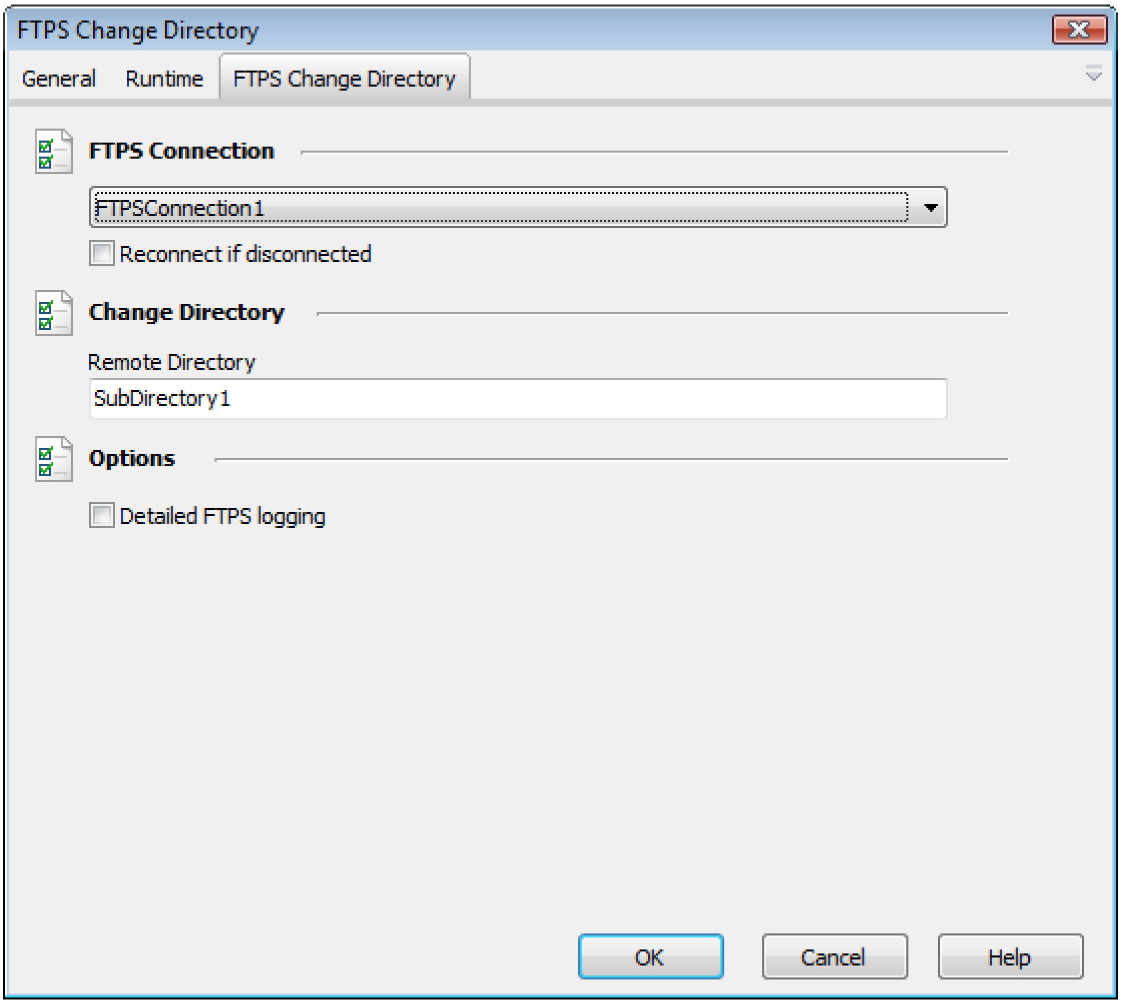
FTPS Options

The FTPS options allows you to specify global values that can be used, to save you having to enter the same details every time you want to establish a connection.

See the FTPS Connect action for more details about these fields.

6.15.15.1 FTPS Change Directory

The FTPS Change Directory action allows you to change the current directory of the FTPS server.



FTPS Change Directory

General | Runtime | FTPS Change Directory

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Change Directory

Remote Directory: SubDirectory1

Options

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

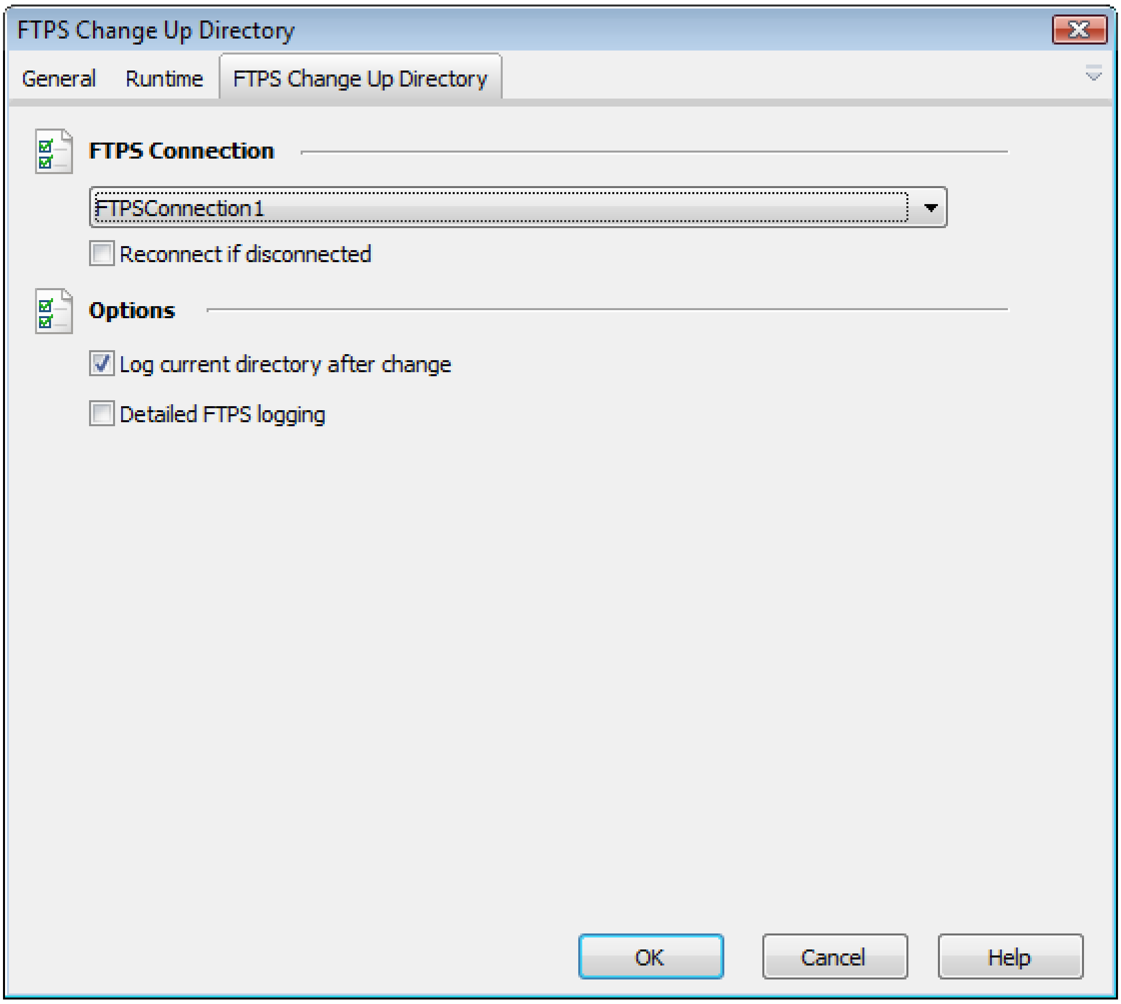
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote Directory - The name of the remote directory to change to. If the directory specified does not exist then the action will fail.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.2 FTPS Change Up Directory

The FTPS Change Up Directory action changes from the current directory to the parent of the current directory on the FTPS server.



FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Options

☑ Log current directory after change

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

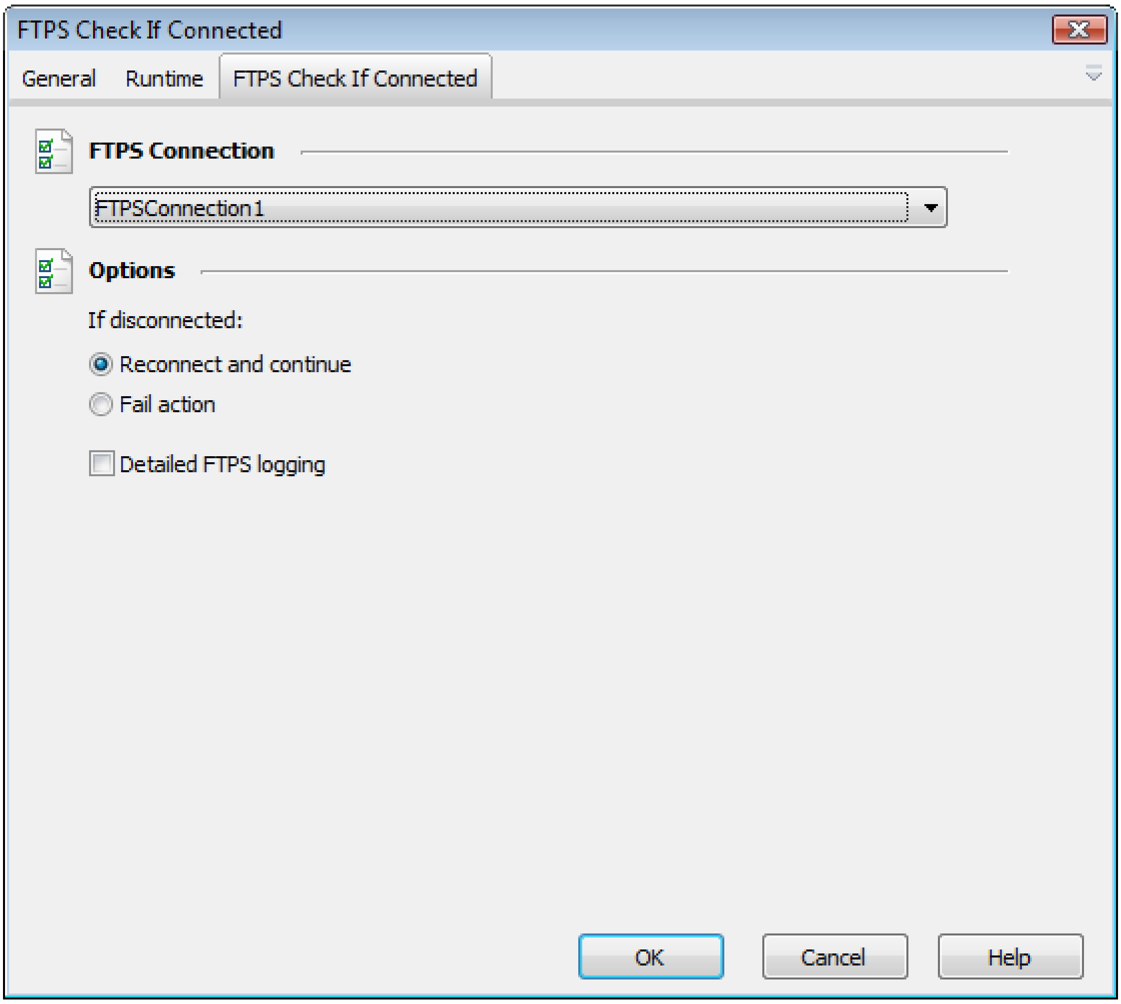
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Log current directory after change - Write the name of the current directory to the log once the change has completed.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.3 FTPS Check If Connected

The FTPS Check If Connected action allows you to check whether the connection to the FTPS server is open and available for use.



Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

If Disconnected - The options determine the behaviour of the action if the connection has been disconnected.

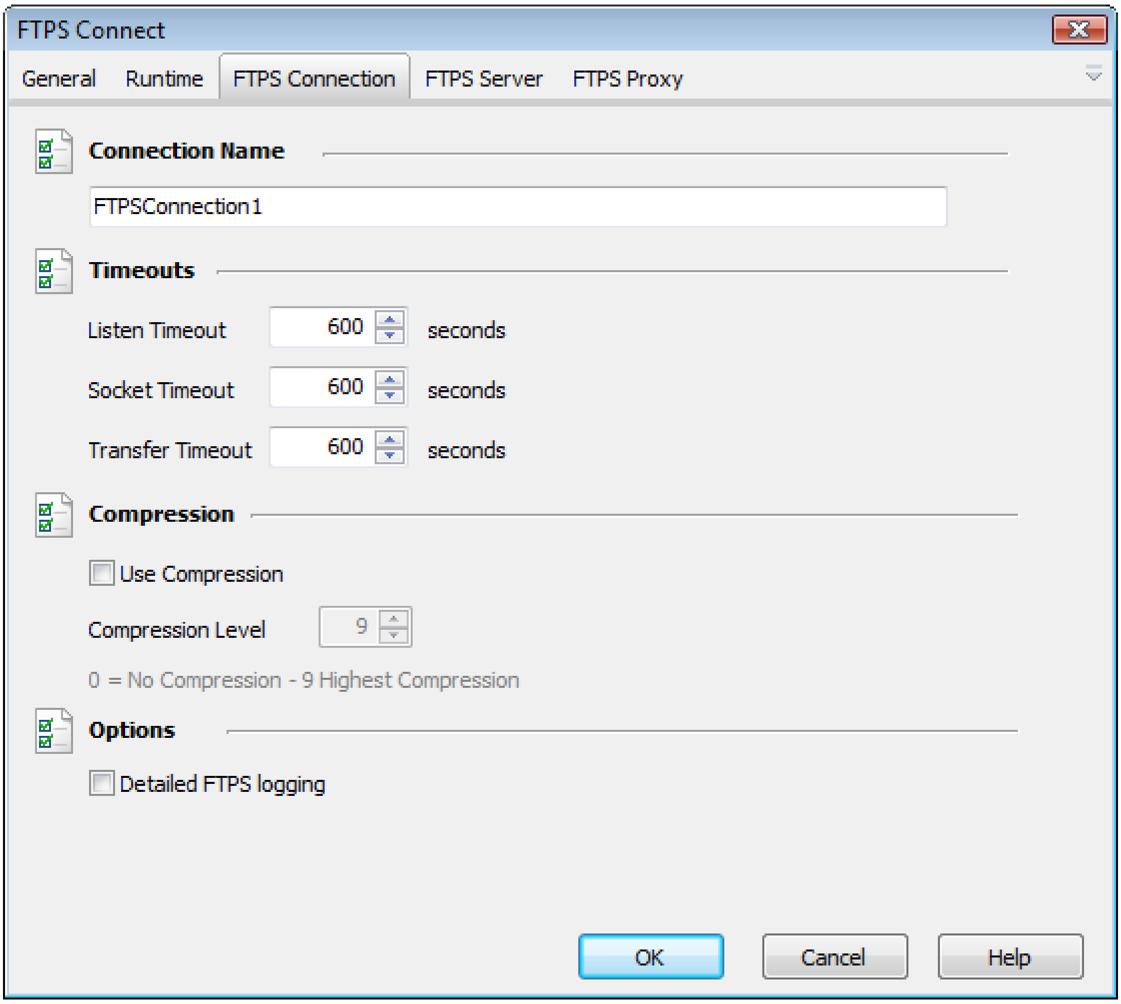
> **Reconnect and continue** - Attempt to reconnect and continue executing the subsequent actions. If reconnection fails then action will fail.
>
> **Fail action** - Action will fail immediately if connection has been disconnected.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.4 FTPS Connect

The FTPS Connect action is used to establish a connection to an FTPS server. Once successfully established, the connection can be used to perform other FTPS operations (by using the other FTPS actions).

FTPS Connection



FTPS Connect

Connection Name

FTPSConnection1

Timeouts

Listen Timeout 600 seconds

Socket Timeout 600 seconds

Transfer Timeout 600 seconds

Compression

□ Use Compression

Compression Level 9

0 = No Compression - 9 Highest Compression

Options

□ Detailed FTPS logging

Connection Name - The name for the new FTPS connection.

Timeouts

Listen Timeout - This is the maximum time the listening socket will be open in the active mode (in seconds).

Socket Timeout - This is the time period in which the client can establish a socket connection to the server (in seconds).

Transfer Timeout - This is time period which a client will wait for a data connection from the remote side to be accepted (in seconds).

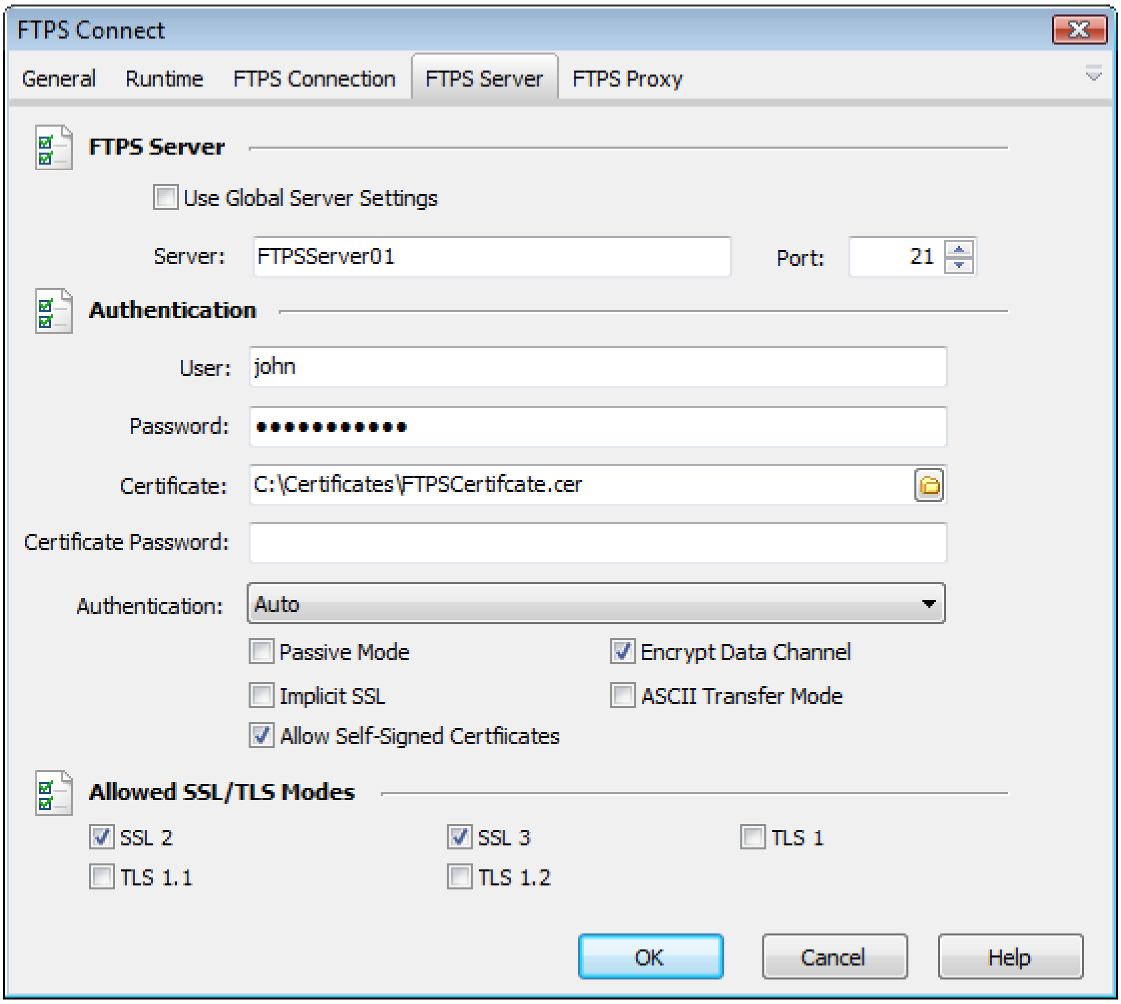
Compression

Use Compression - Enable the use of Mode Z compression (if supported by the FTPS server).

Compression Level - Indicates the compression level (where 0 is no compression and 9 is the highest level of compression).

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

FTPS Server



FTPS Server

□ Use Global Server Settings

Server: FTPSServer01 Port: 21

Authentication

User: john

Password: ●●●●●●●●●●●●

Certificate: C:\Certificates\FTPSCertifcate.cer

Certificate Password:

Authentication: Auto

□ Passive Mode ☑ Encrypt Data Channel

□ Implicit SSL □ ASCII Transfer Mode

☑ Allow Self-Signed Certficates

Allowed SSL/TLS Modes

☑ SSL 2 ☑ SSL 3 □ TLS 1

□ TLS 1.1 □ TLS 1.2

Use Global Server Settings - Enable this option to use the server details specified in the FTPS Options page.

Server - The host name or address of the FTPS server.

User - The username for the FTPS connection.

Password - The password for the FTPS user.

Certificate - The SSL certificate file to be used for FTPS authentication.

Certificate Password - (Optional) Only required where certificate contains an embedded private key protected by a passphrase (this is common in PFX files). If the certificate does not require a password, leave this field empty.

Authentication - Specified the authorisation command that should be sent to the server to request an explicit SSL session.

Auto - Attempt to specify the command used by the server automatically.

AUTH TLS - Use the Auth TLS command.

AUTH SSL - Use the Auth SSL command.

AUTH TLS-C - Use the Auth TLS-C command (clear data channel).

AUTH TLS-P - Use the Auth TLS-P command (protected data channel)

Passive Mode - Specifies whether to use active or passive mode for file transfers.

Encrypt Data Channel - Specifies whether or not data channel (used for file transfers) will be encrypted.

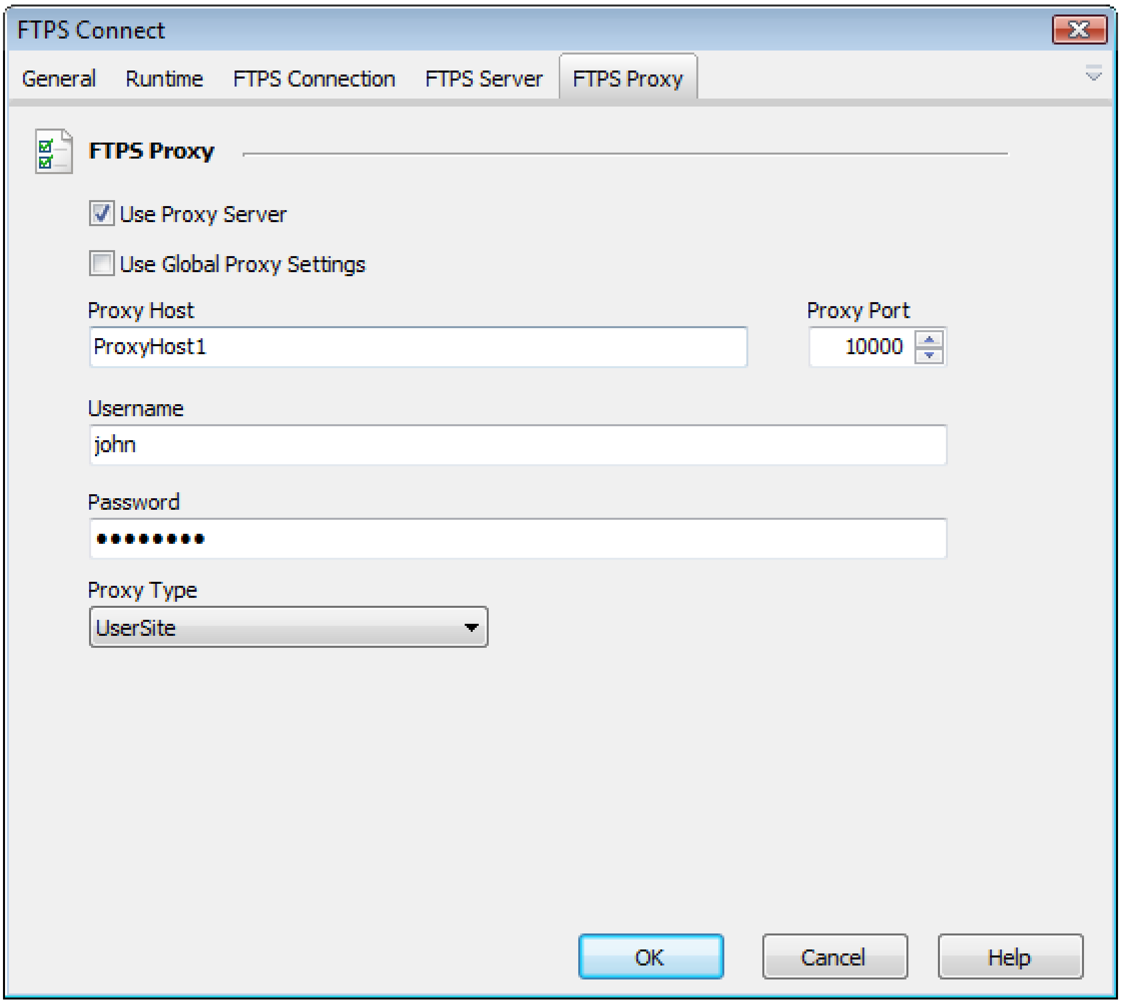
Implicit SSL - Specifies whether or not connection should be established in implicit or explicit mode.

ASCII Transfer Mode - Specifies whether connection should use Binary transfer mode or ASCII transfer mode.

Allow Self-Signed Certificates - Enable this option to allow connection using a self signed certificate.

Allowed SSL/TLS Modes - These options define which SSL/TLS versions are allowed to be used during the FTPS session.

FTPS Proxy



☑ Use Proxy Server

□ Use Global Proxy Settings

Proxy Host: ProxyHost1

Proxy Port: 10000

Username: john

Password: ••••••••

Proxy Type: UserSite

Use Proxy Server - Enable this option to use a proxy server when connecting to the FTPS server.

Use Global Proxy Settings - Enable this option to use the proxy settings defined in the in FTPS Options page.

Proxy Host - The host name or address of the proxy to use.

Proxy Port - The port number used to connect to the host.

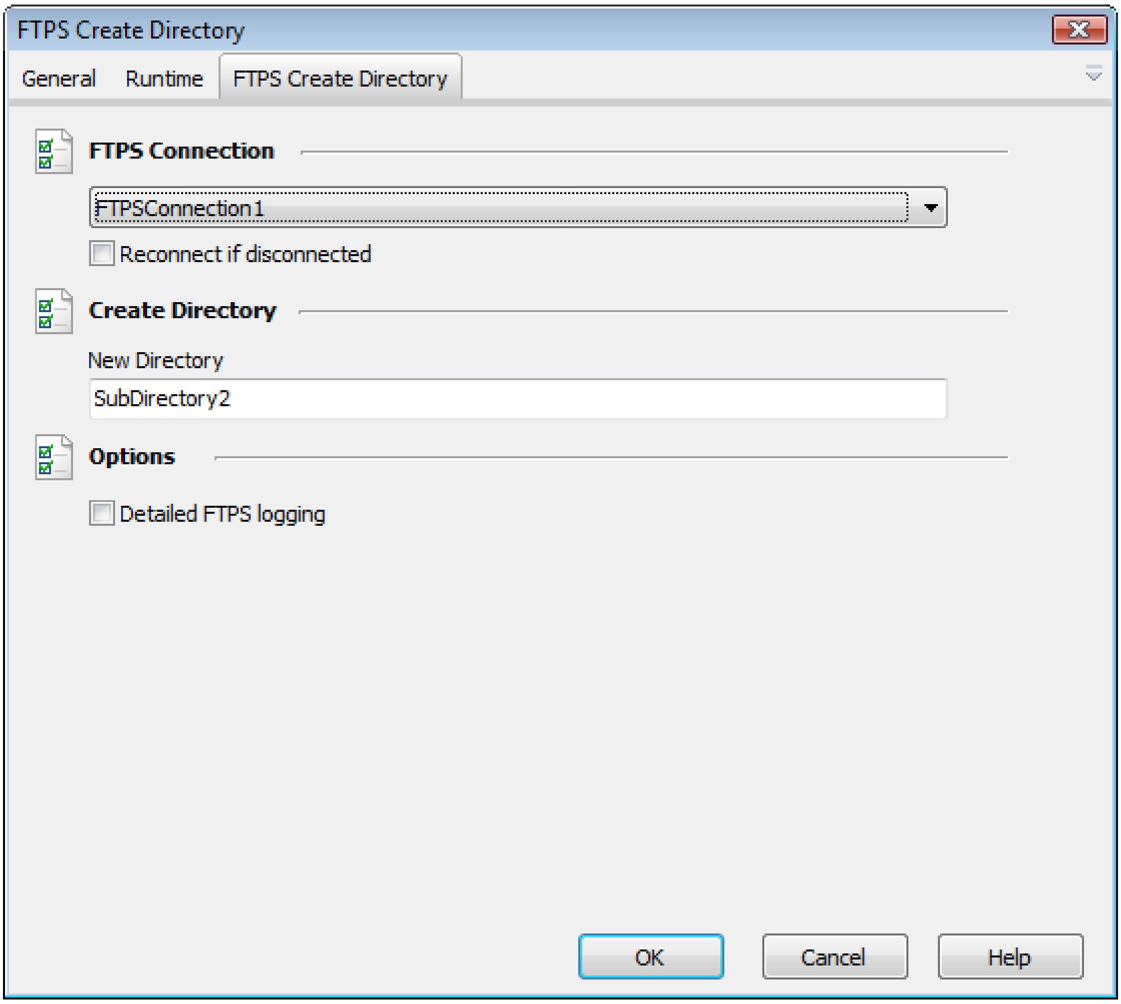
Username - The username for the proxy connection.

Password - The password for the proxy connection.

Proxy Type - The type of proxy server being used.

6.15.15.5 FTPS Create Directory

The FTPS Create Directory action creates a new directory on the FTPS Server.



FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Create Directory

New Directory

SubDirectory2

Options

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

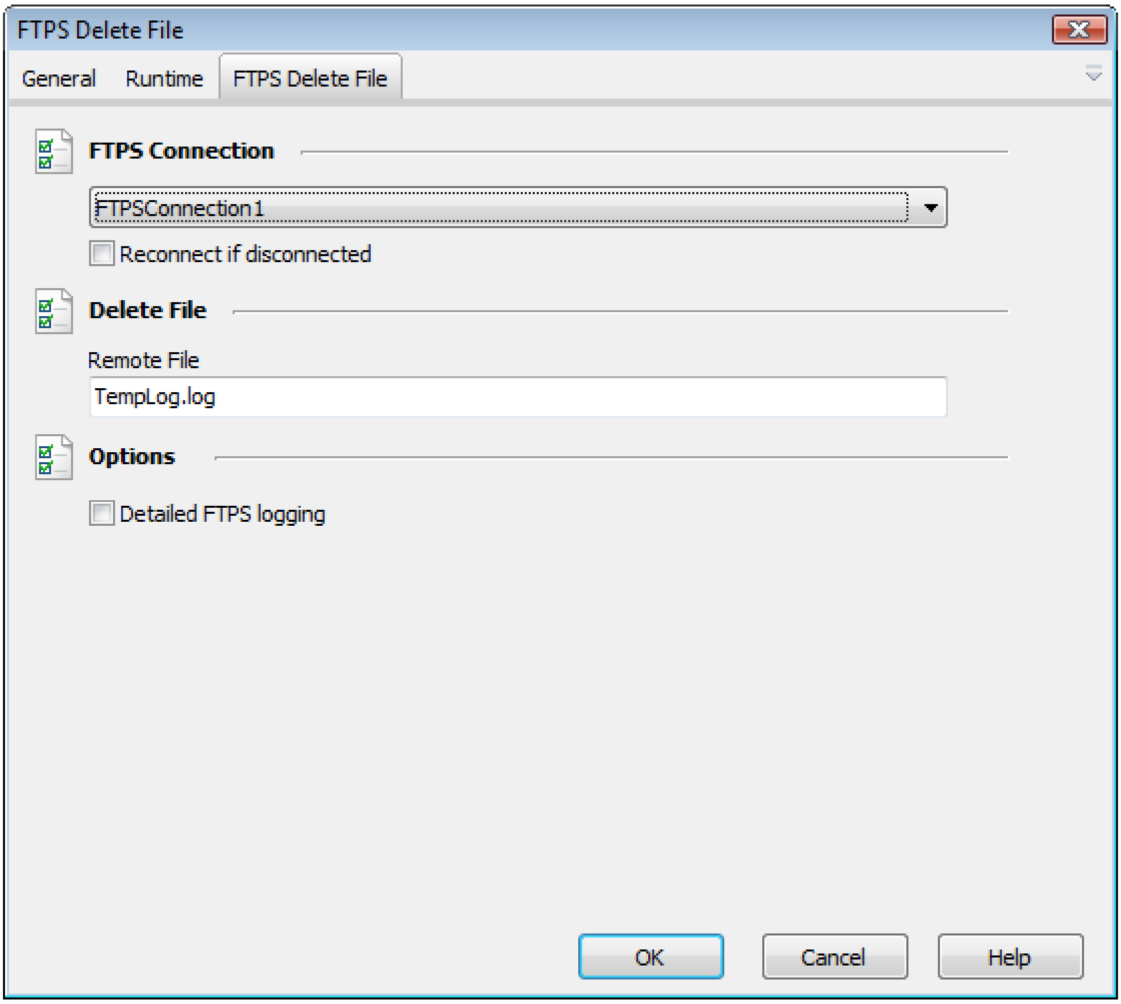
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

New Directory - The name of the new directory to be created.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.6 FTPS Delete File

The FTPS Delete File action allows you to delete a file from the FTPS server.


FTPS Delete File

General | Runtime | FTPS Delete File

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Delete File

Remote File

TempLog.log

Options

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

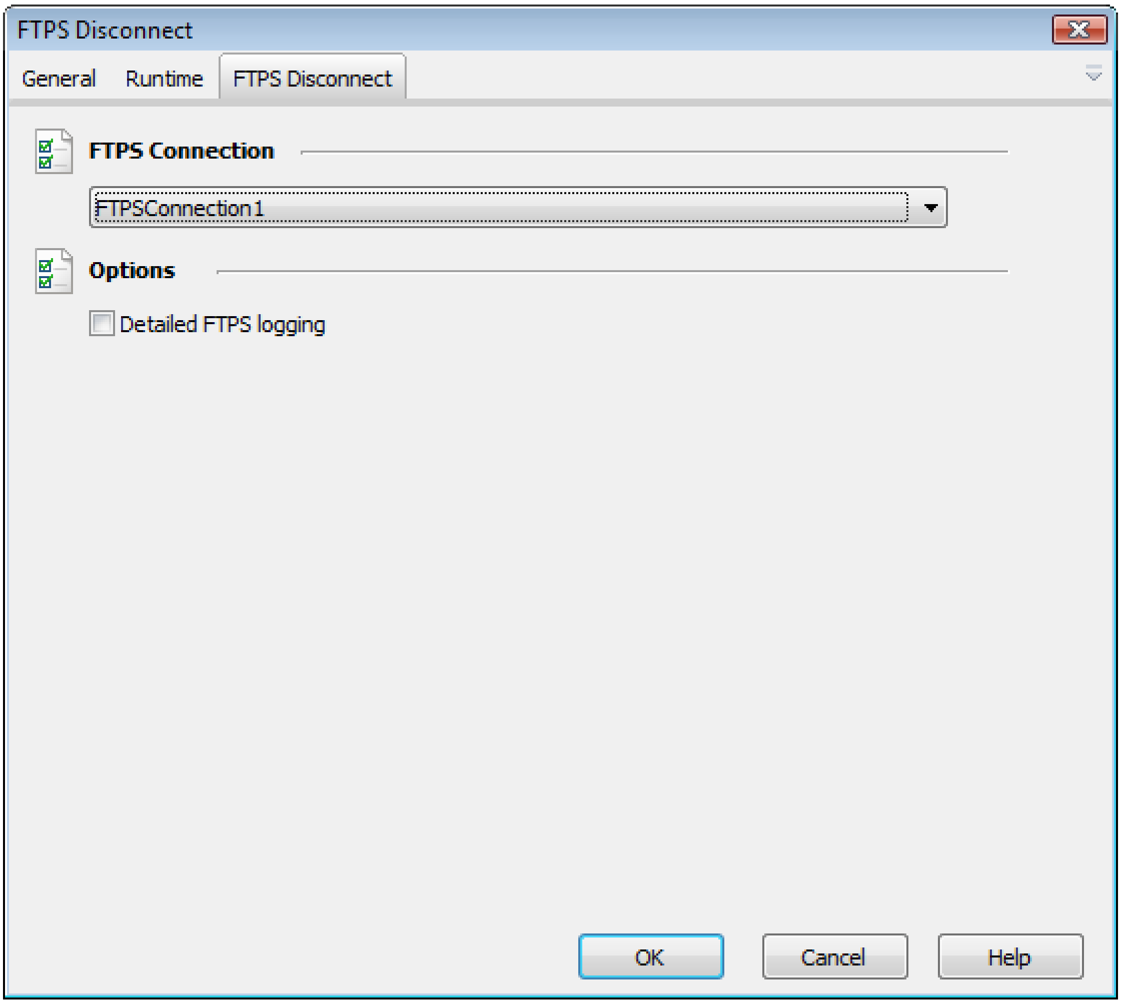
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote File - This is the name of the file to be deleted.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.7 FTPS Disconnect

The FTPS Disconnect action is used to end the specified FTPS session.



FTPS Connection

FTPSConnection1

Options

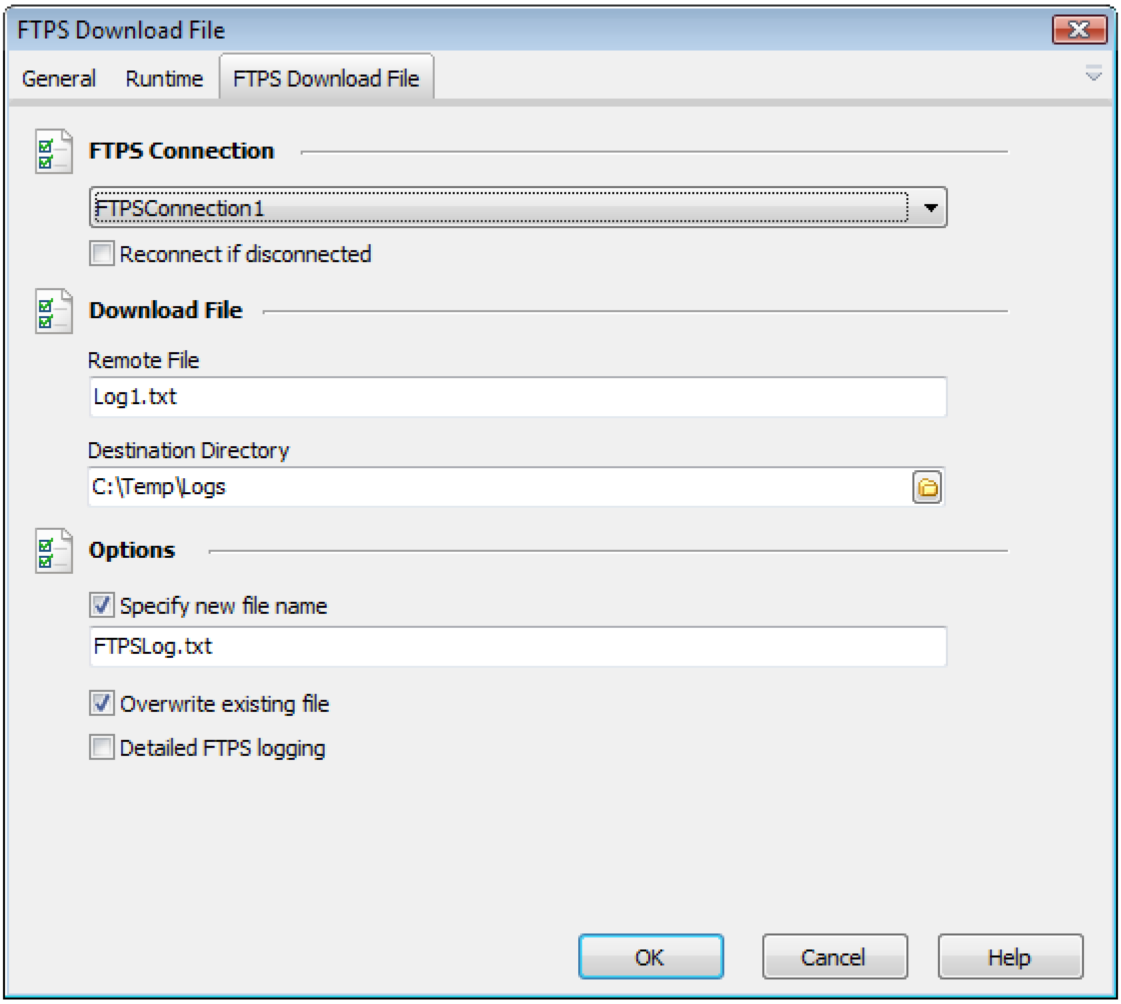
□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.8 FTPS Download File

The FTPS Download File action is used to download a single file from the FTPS server to the local filesystem.



FTPS Download File

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Download File

Remote File: Log1.txt

Destination Directory: C:\Temp\Logs

Options

☑ Specify new file name

FTPSLog.txt

☑ Overwrite existing file

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote File - The name of the remote file to download.

Destination Directory - The local directory to download the selected file to.

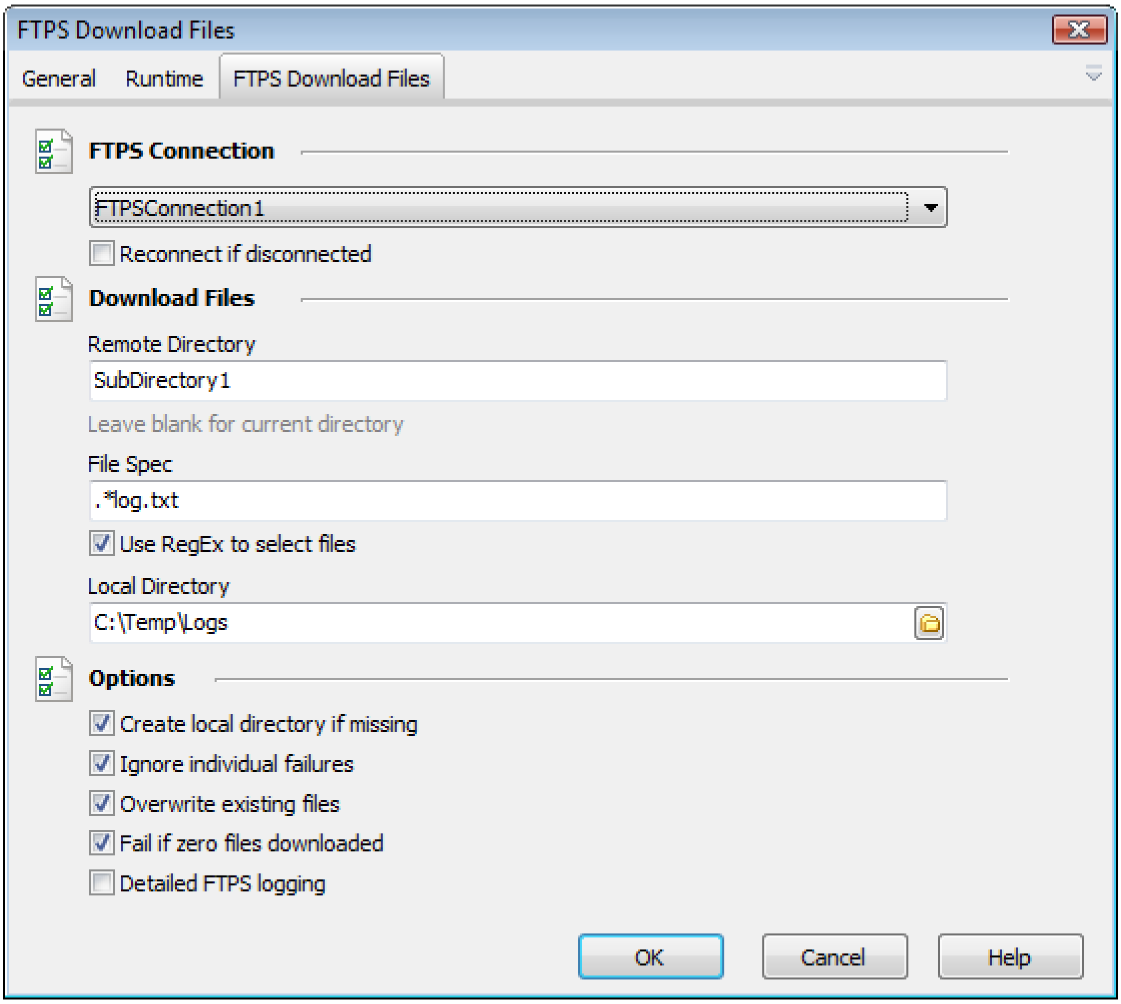
Specify a new file name - This option allows you to rename the file as part of the download operation.

Overwrite existing file - Enable this option to overwrite a file of the same name in the destination directory. If this option is not enabled and the file already exists in the local directory then the action will fail.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.9 FTPS Download Files

The FTPS Download Files action can be used to download multiple files from the server to the local file system.



FTPS Download Files

General | Runtime | FTPS Download Files

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Download Files

Remote Directory: SubDirectory1

Leave blank for current directory

File Spec: .*log.txt

☑ Use RegEx to select files

Local Directory: C:\Temp\Logs

Options

☑ Create local directory if missing

☑ Ignore individual failures

☑ Overwrite existing files

☑ Fail if zero files downloaded

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote Directory - The remote directory that contains the files to be downloaded.

File Spec - Specify a file mask to only download files in the remote directory that match the mask.

Use RegEx to select files - Enable this option to use a RegEx to select the files to be downloaded (the regular expression is specified via the File Spec field).

Local Directory - This is the local directory that the files will be downloaded to.

Create local directory if missing - If the local directory specified does not exist, create it before attempting to download files.

Ignore individual failures - This option allows a single file to download to fail without affecting the overall outcome of the action.

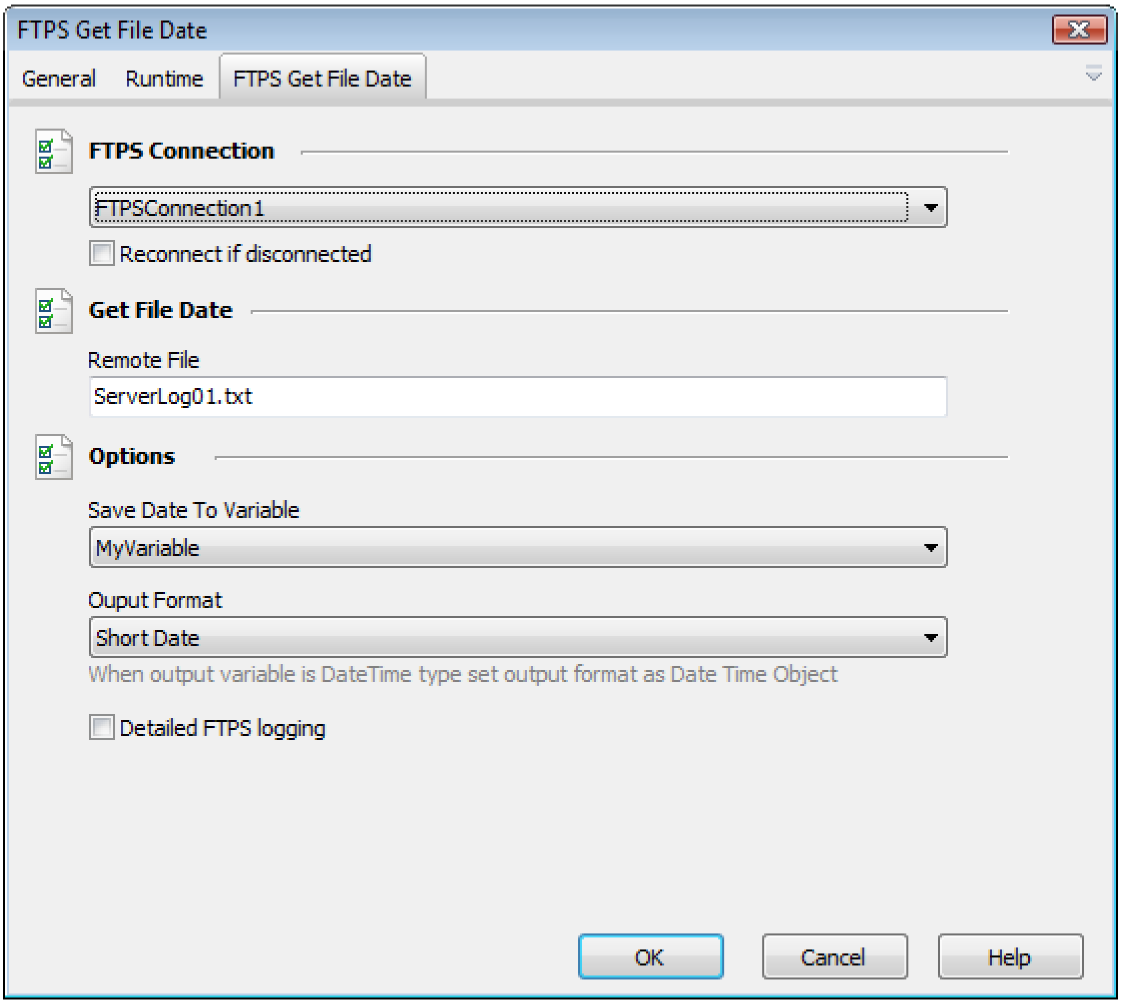
Overwrite existing files - If a file exists in the destination, overwrite it.

Fail if zero files downloaded - If no files are downloaded, fail the action.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.10 FTPS Get File Date

The FTPS Get File Date allows you to get the modified date of a file on the FTPS server.



FTPS Get File Date

FTPSConnection1

□ Reconnect if disconnected

Remote File: ServerLog01.txt

Save Date To Variable: MyVariable

Ouput Format: Short Date

When output variable is DateTime type set output format as Date Time Object

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote File - The name of the file to retrieve the date of.

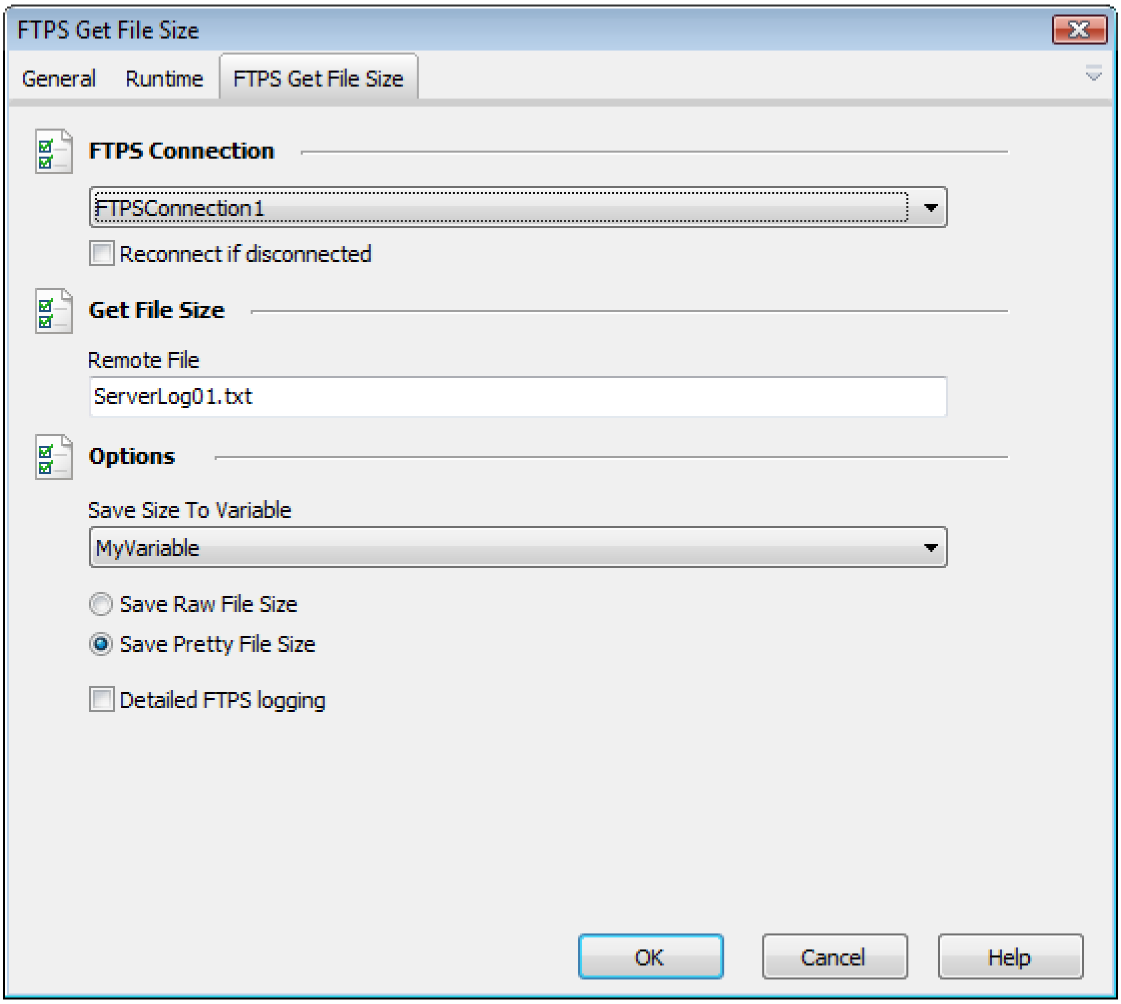
Save Date To Variable - Specify a variable to store the file date in.

Output Format - The date time format to output the file date as.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.1:FTPS Get File Size

The FTPS Get File Size allows you to retrieve the size of a file from the FTPS server.



FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Get File Size

Remote File: ServerLog01.txt

Options

Save Size To Variable: MyVariable

○ Save Raw File Size

● Save Pretty File Size

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote File - The name of the remote file to retrieve the size for.

Save Size To Variable - Specify a variable to save the file size to.

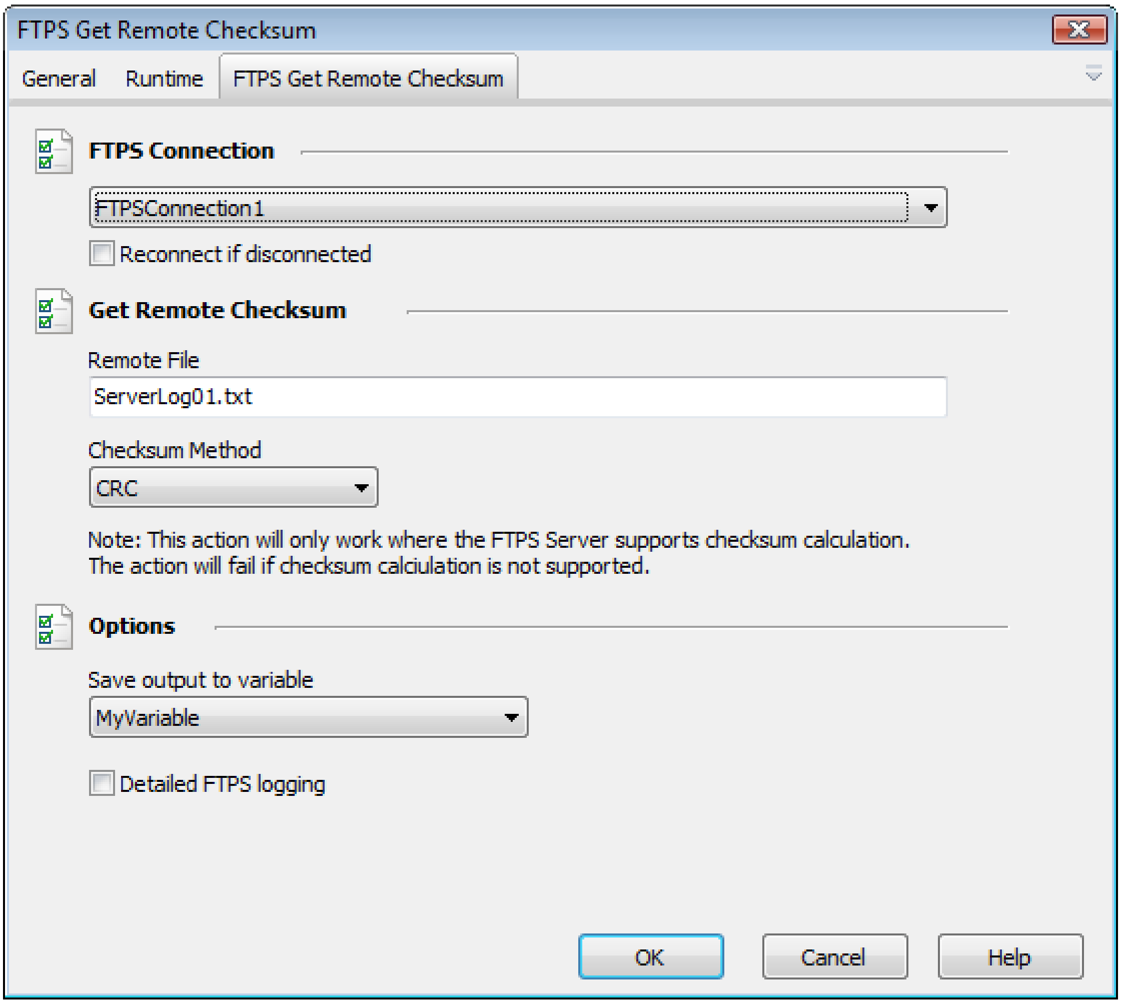
Save Raw File Size - Enable this option to save the file size in bytes.

Save Pretty File Size - Enable this option to save the file size in pretty formatted.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.11FTPS Get Remote Checksum

The FTPS Get Remote Checksum action allows you to retrieve the remote checksum for a selected file.



FTPS Get Remote Checksum

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Get Remote Checksum

Remote File: ServerLog01.txt

Checksum Method: CRC

Note: This action will only work where the FTPS Server supports checksum calculation.
The action will fail if checksum calciulation is not supported.

Options

Save output to variable: MyVariable

□ Detailed FTPS logging

This action will only work where the server supports remote checksums. If the server does not support the selected checksum method the action will fail.

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Remote File - The name of the file to get the remote checksum of.

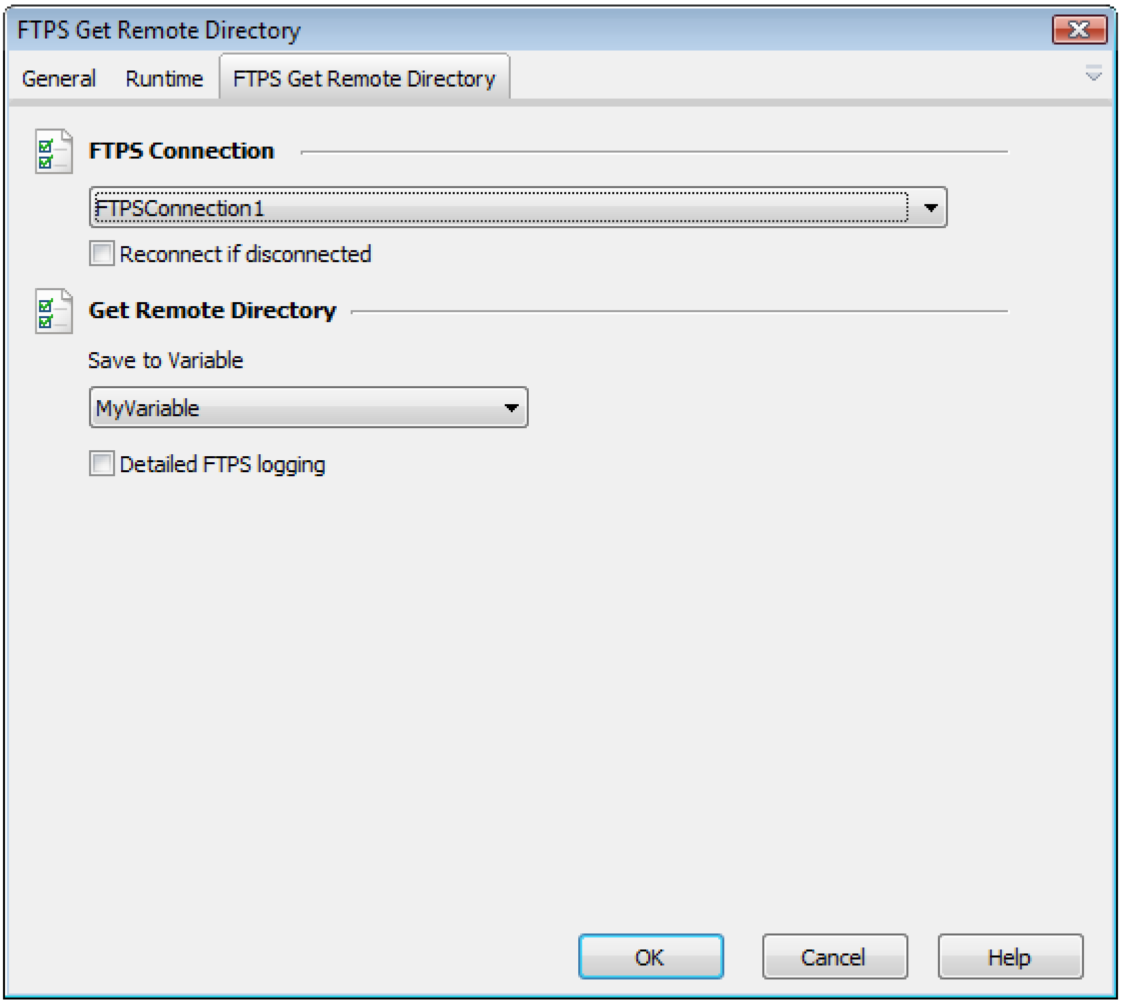
Checksum Method - This is the method used to specify the checksum method (method must be supported by the FTPS server).

Save output to variable - The name of the variable to save the checksum value to.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.13 FTPS Get Remote Directory

The FTPS Get Remote Directory action allows you to get the name of the current directory from the FTPS server.



FTPS Get Remote Directory

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Get Remote Directory

Save to Variable

MyVariable

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

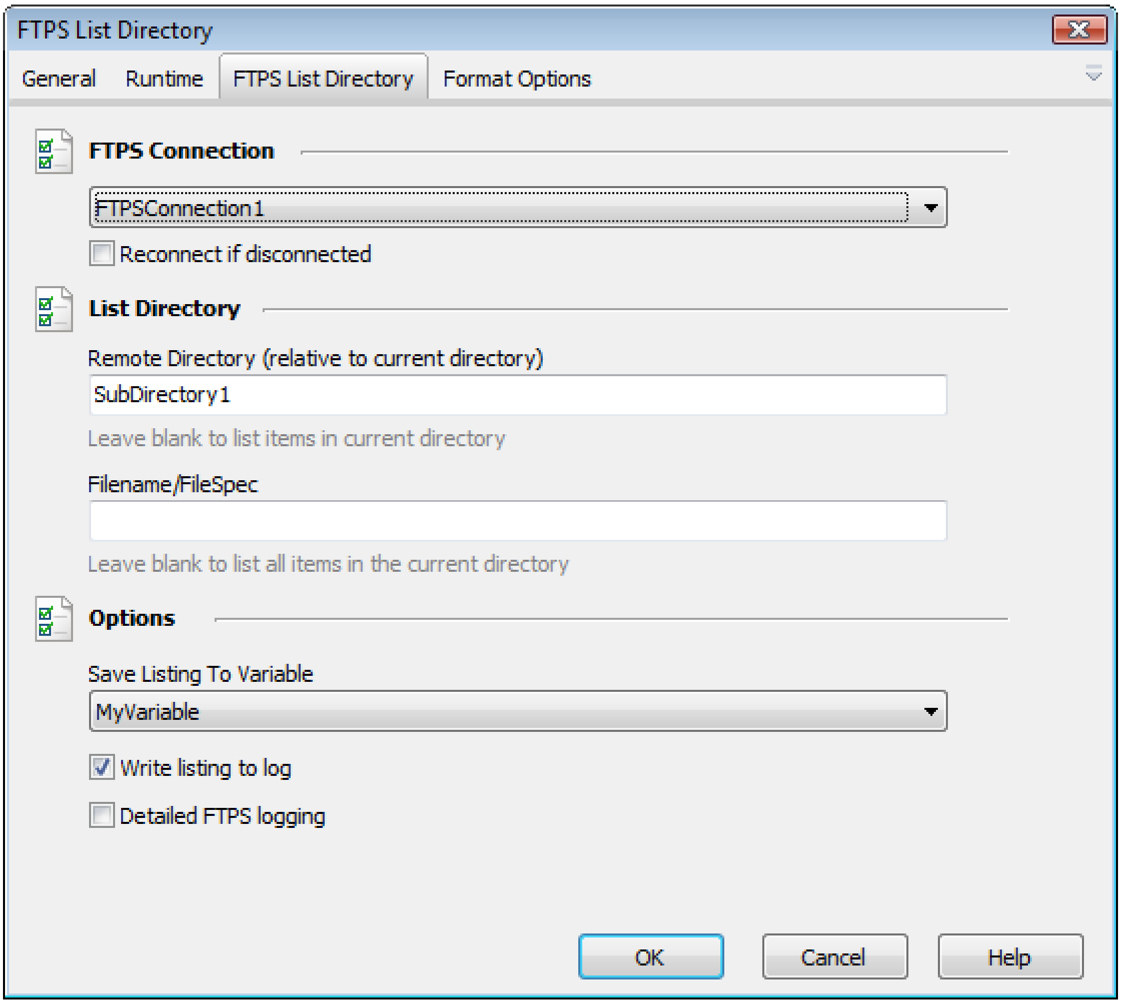
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Save to variable - Specify the name of a variable to save the directory name to.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.14 FTPS List Directory

The FTPS List Directory action lists the contents of the specified directory.



FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

List Directory

Remote Directory (relative to current directory)

SubDirectory1

Leave blank to list items in current directory

Filename/FileSpec

Leave blank to list all items in the current directory

Options

Save Listing To Variable

MyVariable

☑ Write listing to log

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

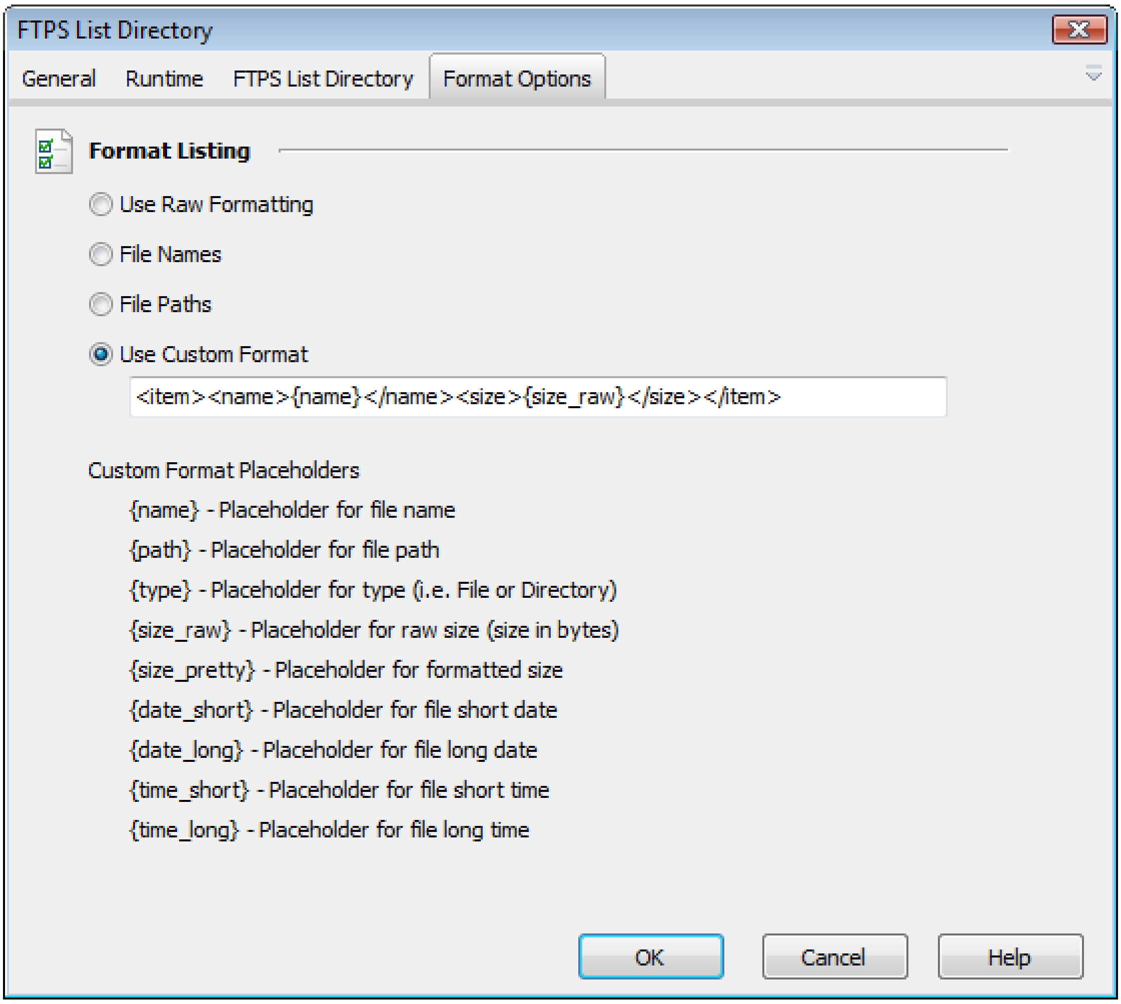
Remote Directory - The name of the remote directory to list the contents of (leave blank to list the current directory).

Filename/FileSpec - Specify a filename/file spec to only list specific items (leave blank to list items in the directory).

Save Listing To Variable - The name of the variable to save the listing to.

Write listing to log - Enable this option to write the listing to the log.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.



Format Listing

○ Use Raw Formatting

○ File Names

○ File Paths

● Use Custom Format

<item><name>{name}</name><size>{size_raw}</size></item>

Custom Format Placeholders

{name} - Placeholder for file name

{path} - Placeholder for file path

{type} - Placeholder for type (i.e. File or Directory)

{size_raw} - Placeholder for raw size (size in bytes)

{size_pretty} - Placeholder for formatted size

{date_short} - Placeholder for file short date

{date_long} - Placeholder for file long date

{time_short} - Placeholder for file short time

{time_long} - Placeholder for file long time

Format Options - Use the format options to specify the format that the listing should be listed in.

Use Raw Formatting - List the items in the format that the server returns the listing as.

File Names - List only the names of the items.

File Paths - List the paths of the items.

Use Custom Format - Use this option to specify your own format string using the placeholders specified.

The recognise placeholders are:

{name} - The name of the item.

{path} - The full path of the item.

{type} - The type of the item (i.e. File, Directory, Symbolic Link)

{size_raw} - Raw size in bytes (if the item is a directory this will be zero)

{size_pretty} - Pretty size (if the item is a directory this will be zero)

{date_short} - The modified date as a short date.

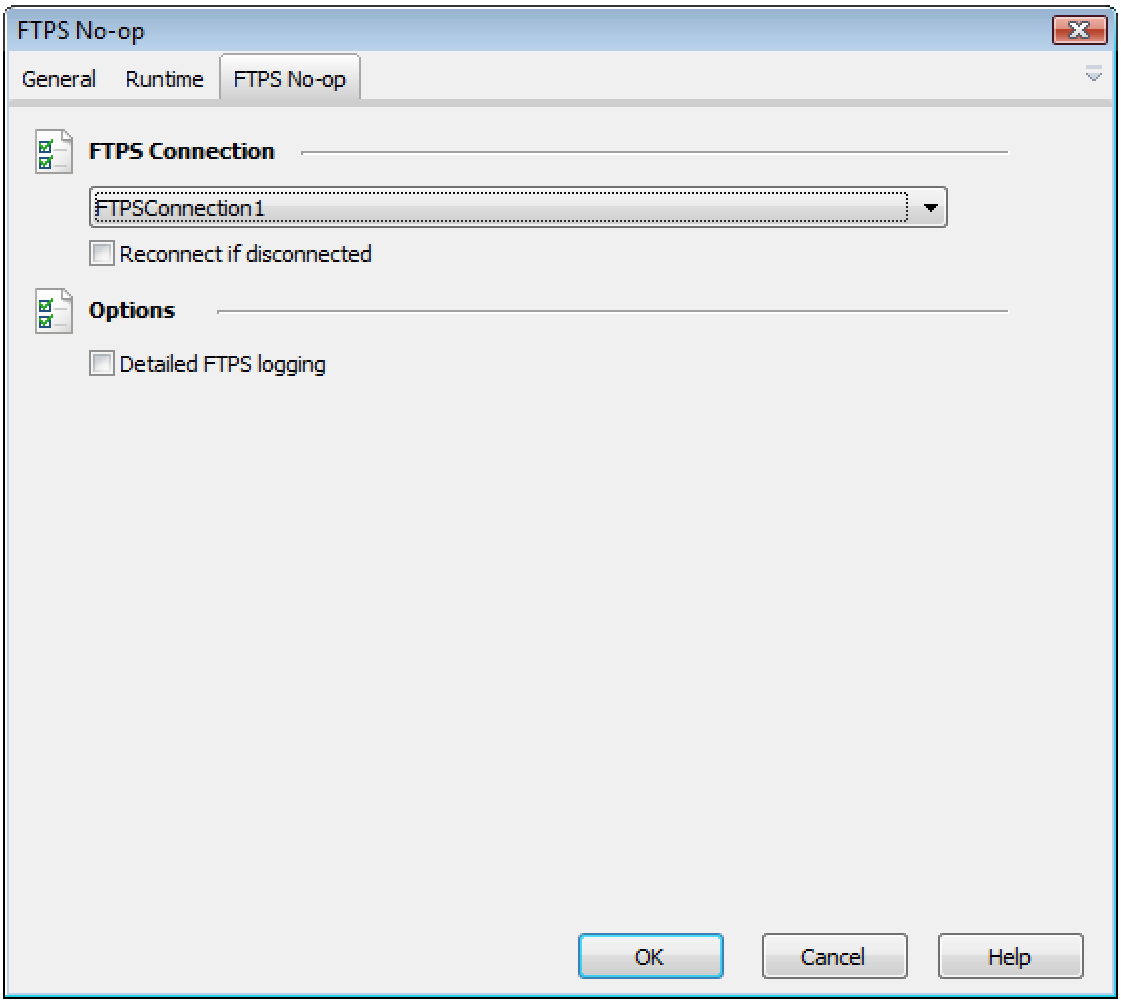
{date_long} - The modified date as a long date.

{time_short} - The modified time as a short time.

{time_long} - The modified time as a long time.

6.15.15.1!FTPS No-op

The FTPS No-op action sends a NOOP command to the FTPS server. This is used to send a dummy packet to keep the connection alive.



FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Options

□ Detailed FTPS logging

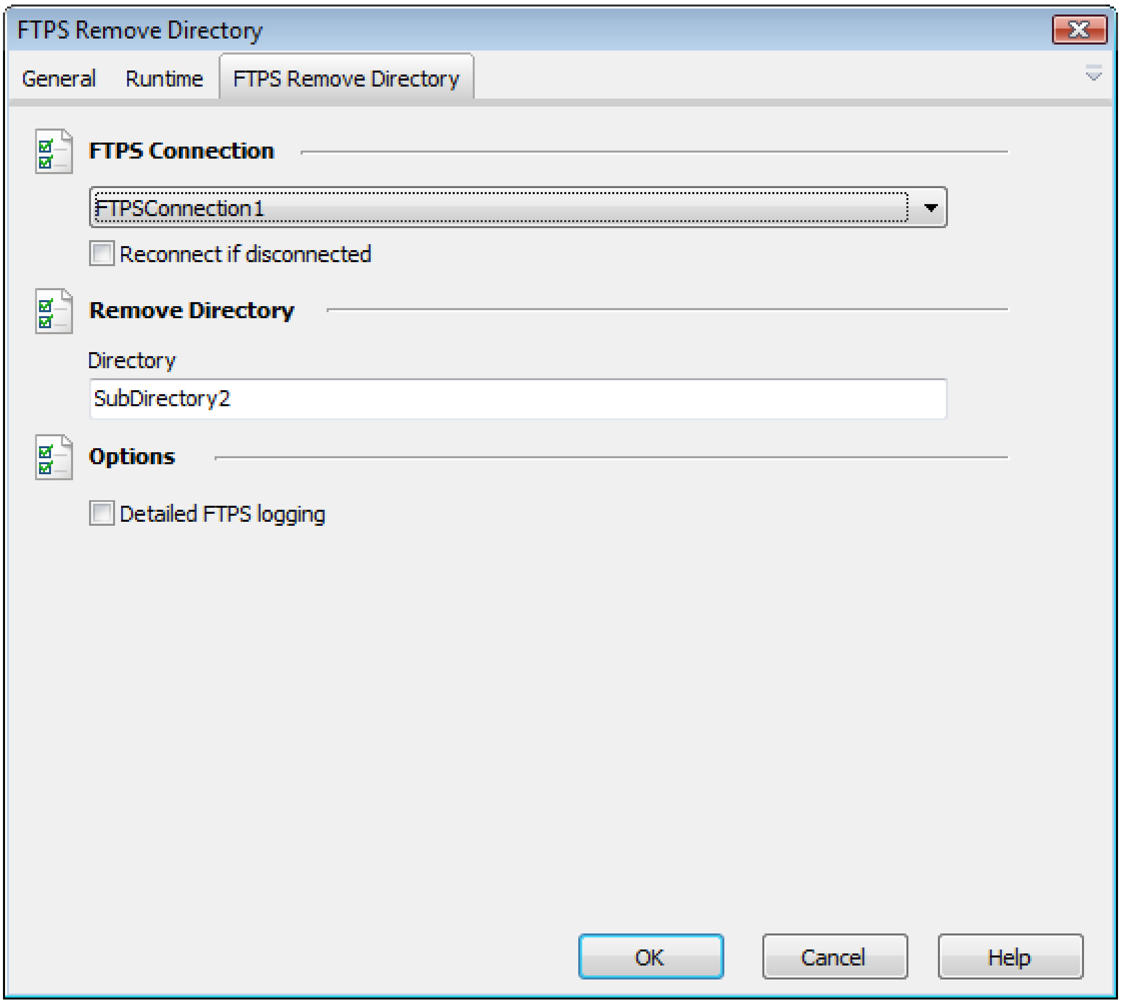
Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.10 FTPS Remove Directory

The FTPS Remove Directory action is used to remove a directory from the FTPS server.



FTPS Remove Directory

FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Remove Directory

Directory: SubDirectory2

Options

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

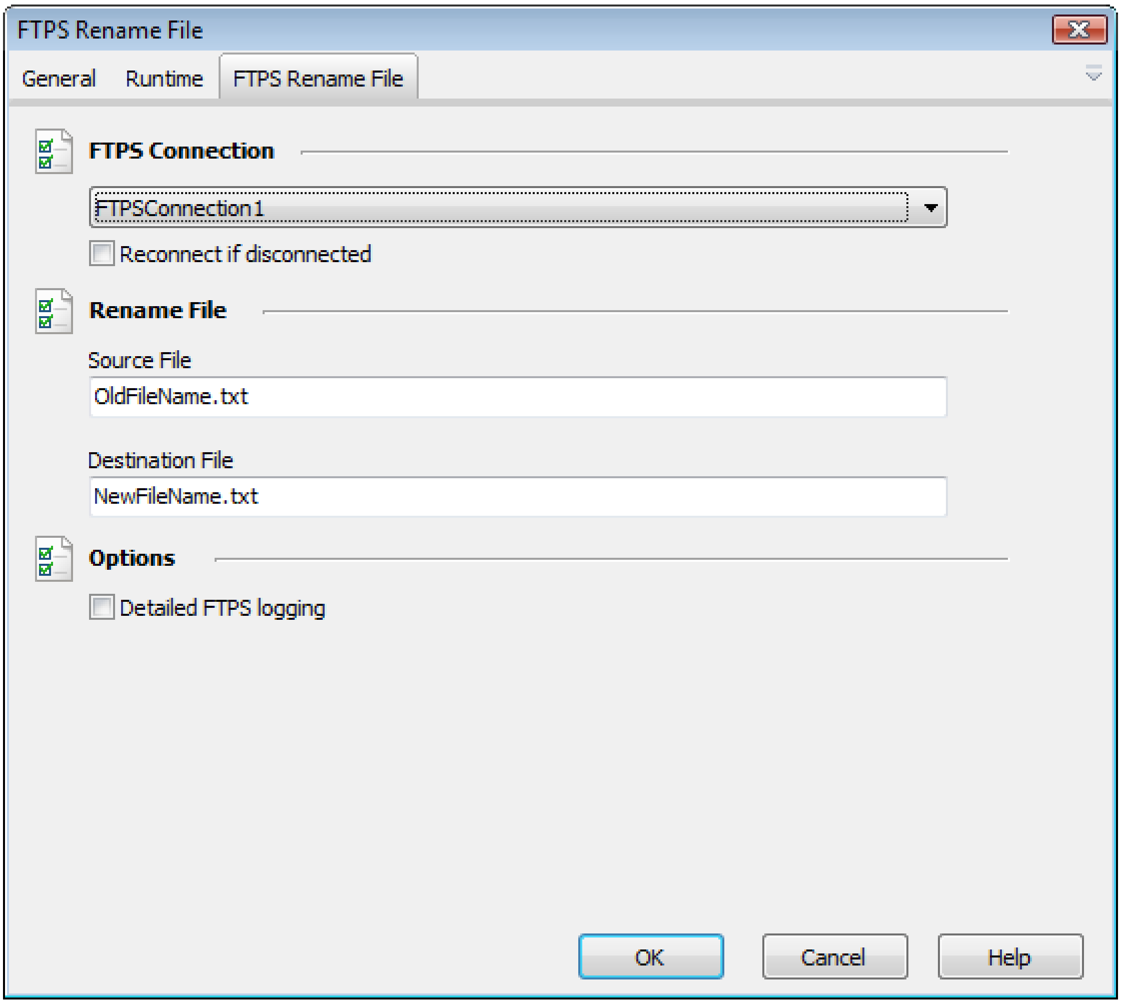
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Directory - Specify the name of the directory to be removed.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.11 FTPS Rename File

The FTPS Rename File action is used to rename a file on the FTPS server.



FTPSConnection1

□ Reconnect if disconnected

Source File: OldFileName.txt

Destination File: NewFileName.txt

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

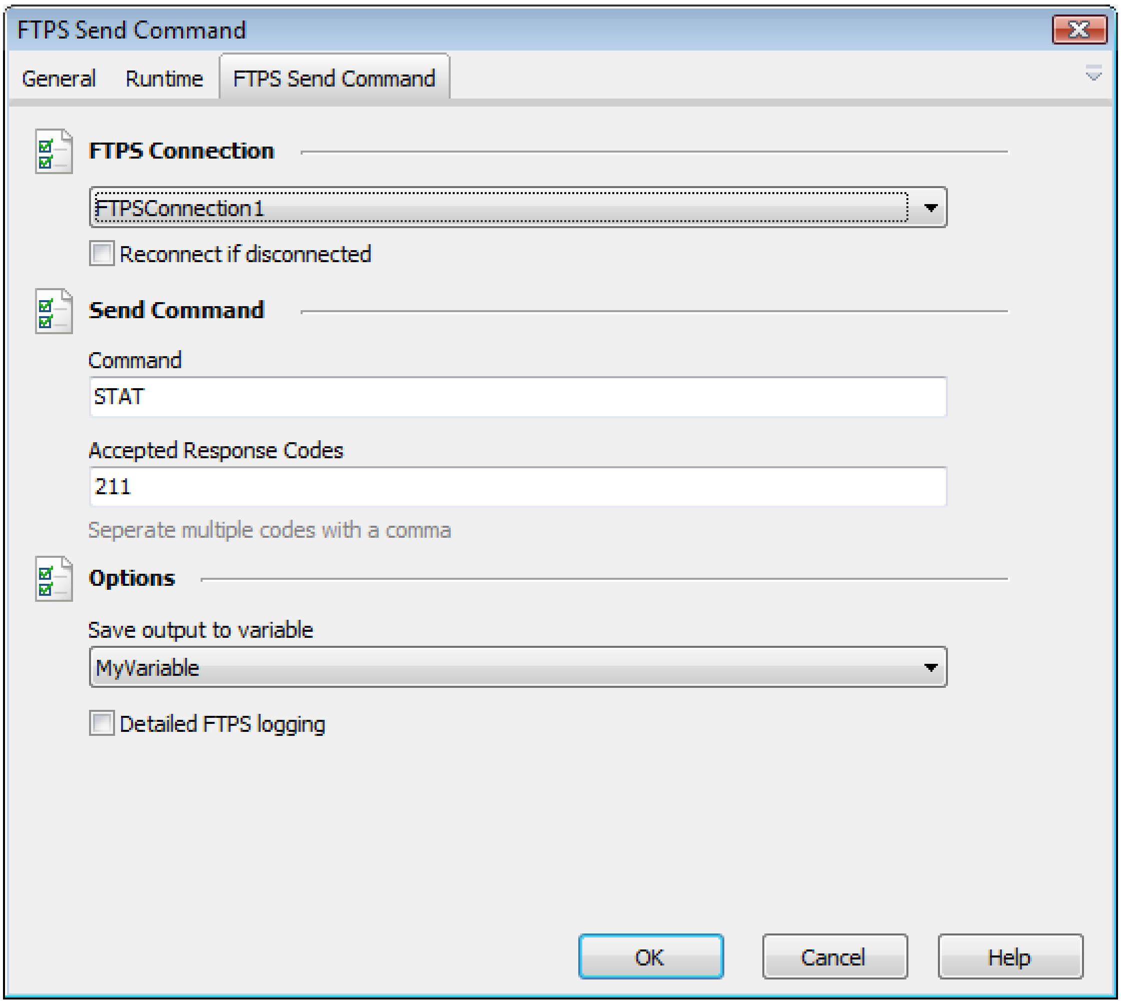
Source File - The name of the file to be renamed.

Destination File - The new name for the file.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.18FTPS Send Command

The FTPS Send Command action allows you to send a custom command to the FTPS server.



FTPS Connection

FTPSConnection1

□ Reconnect if disconnected

Send Command

Command: STAT

Accepted Response Codes: 211

Seperate multiple codes with a comma

Options

Save output to variable: MyVariable

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Command - This is the command to be sent to the FTPS Server.

Accepted Response Codes - When sending a command, at least one accepted response code must be provided. If the code returned by the server is not in the list of accepted responses then the action will fail.

Accepted response codes must be an numeric value. Multiple response codes can be passed be separating each response code with a comma.

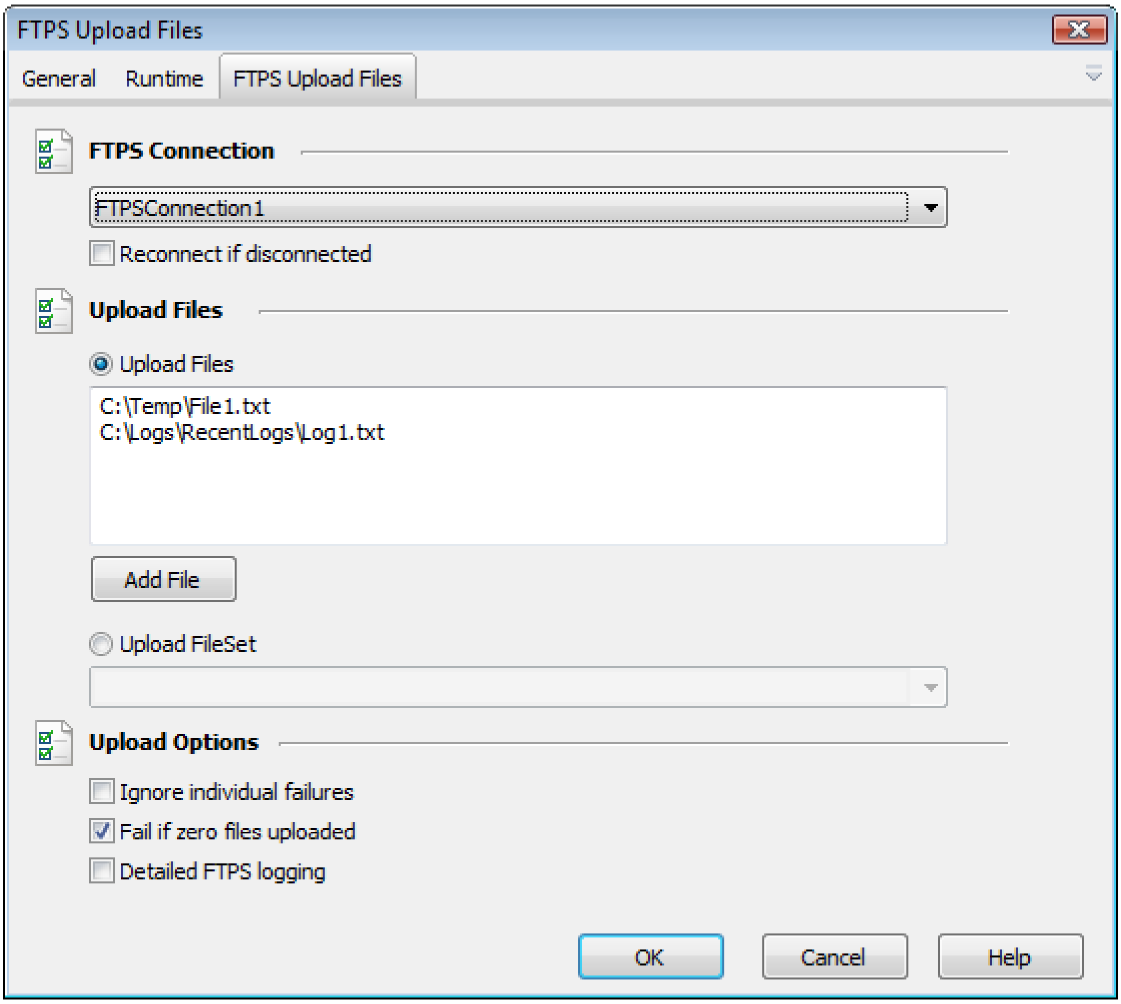
Using the detailed logging option will help you determine the accepted response code that you will need to use for this action. The accepted response code will vary depending on the command being sent.

Save output to variable - Specify the name of a variable to save the output of a command to.

Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.15.19 FTPS Upload Files

The FTPS Upload files action allows you to upload one or more files from the local file system to the FTPS server.



FTPSConnection1

□ Reconnect if disconnected

● Upload Files

```
C:\Temp\File1.txt
C:\Logs\RecentLogs\Log1.txt
```

○ Upload FileSet

□ Ignore individual failures

☑ Fail if zero files uploaded

□ Detailed FTPS logging

Connection Name - The name of the FTPS connection to be used for the operation.

Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

Upload Files - Use the upload files option to provide a list of individual files to be uploaded. Each file listed will be uploaded to the current FTPS directory.

Upload FileSet - Use the upload file set option to provide a file set to be uploaded. When uploading a file set the folder structure of the file set will be maintained, using the current FTPS directory as the root directory.

Ignore individual failures - Enable this option to allow a single file upload to fail without affecting the overall outcome of the action.

Fail if zero files uploaded - The action will fail if zero files were uploaded.

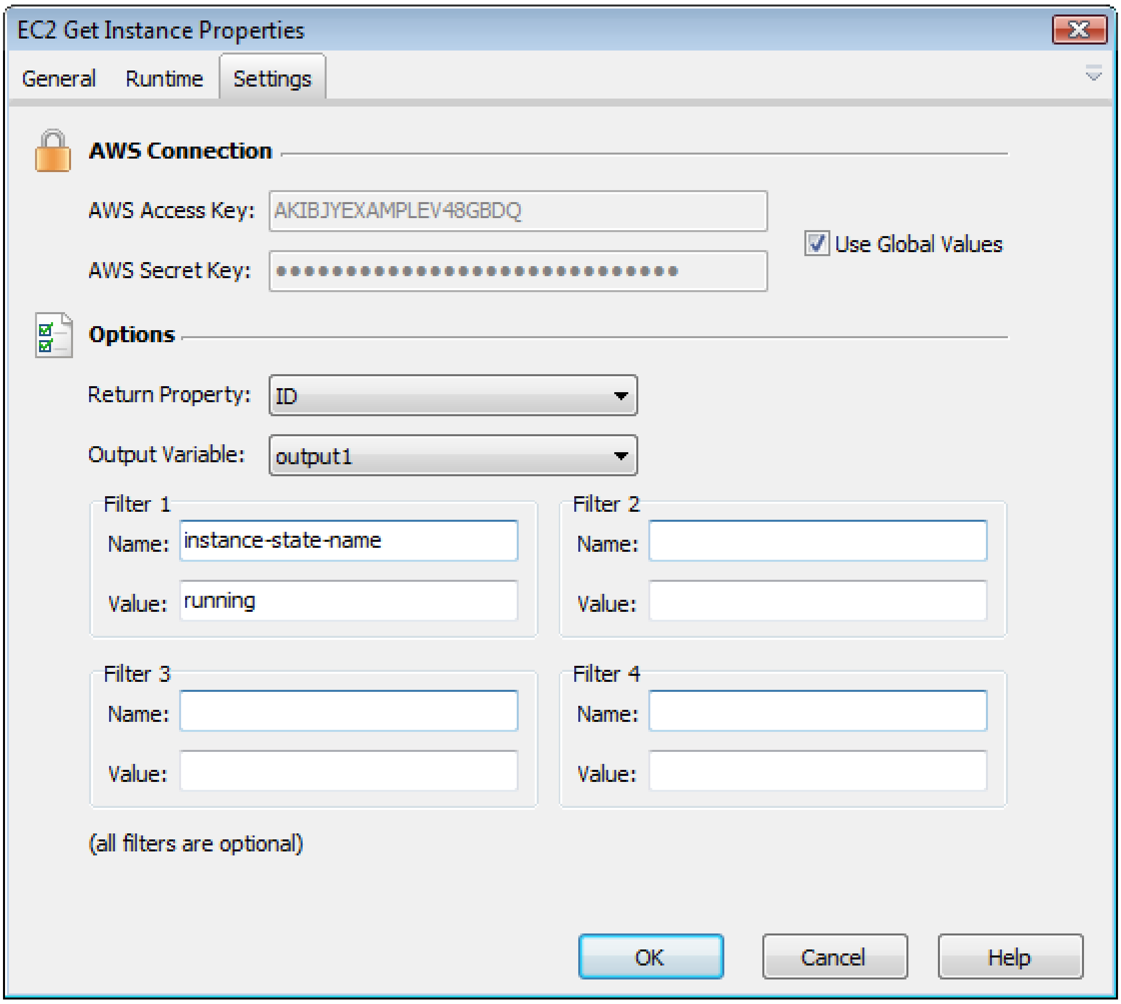
Detailed FTPS logging - Enable this option to log the messages that are sent between the client and the server. This may be useful when attempting to troubleshoot problems.

6.15.16 EC2 Actions

The EC2 actions allow you to manage your EC2 instances from your project.

6.15.16.1 EC2 Get Instance Properties

This action allows you to retrieve the selected property of instances available to you. You can filter which results to return using the optional filters.


EC2 Get Instance Properties

General Runtime Settings

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

☑ Use Global Values

Options

Return Property: ID

Output Variable: output1

Filter 1
Name: instance-state-name
Value: running

Filter 2
Name:
Value:

Filter 3
Name:
Value:

Filter 4
Name:
Value:

(all filters are optional)

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Return Property

The property of the instances to return. Choose between ID, Instance State, IP Address, or DNS Address. Note that for IP Address and DNS Address the instance must be running.

Output Variable

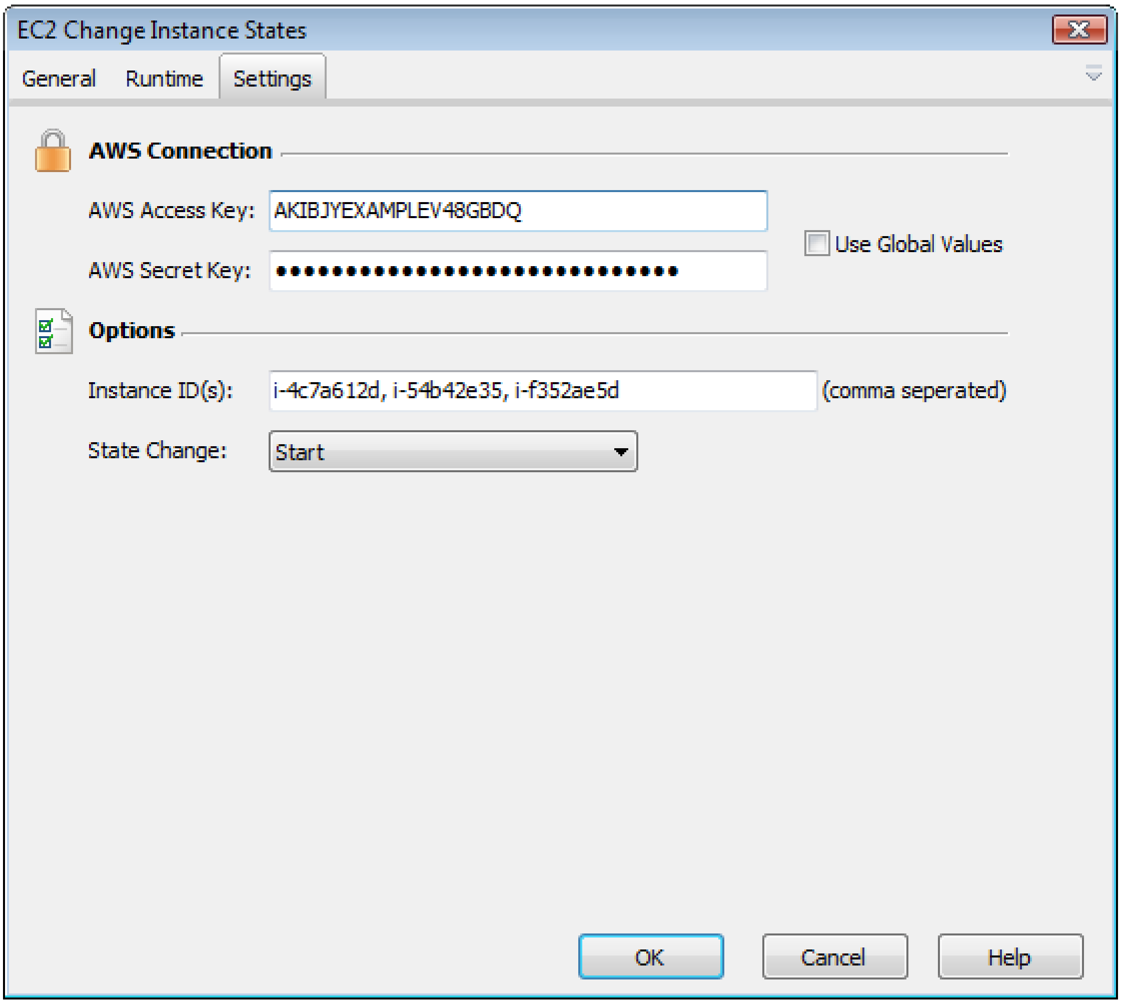
The variable to output the selected property to. If more than one instance matches the filters, a comma seperated list of properties will be returned.

Filters

The name and value of any filters on which instances will be returned. The filter name field will automatically complete with valid filter names. For a list of filters & valid values: http://docs.amazonwebservices.com/AWSEC2/2011-05-15/APIReferenceindex.html?ApiReference-query-DescribeInstances.html

6.15.16.2 EC2 Change Instance States

This action enables you to change the state of one or more instances.


EC2 Change Instance States

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Options

Instance ID(s): i-4c7a612d, i-54b42e35, i-f352ae5d (comma seperated)

State Change: Start

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Instance IDs

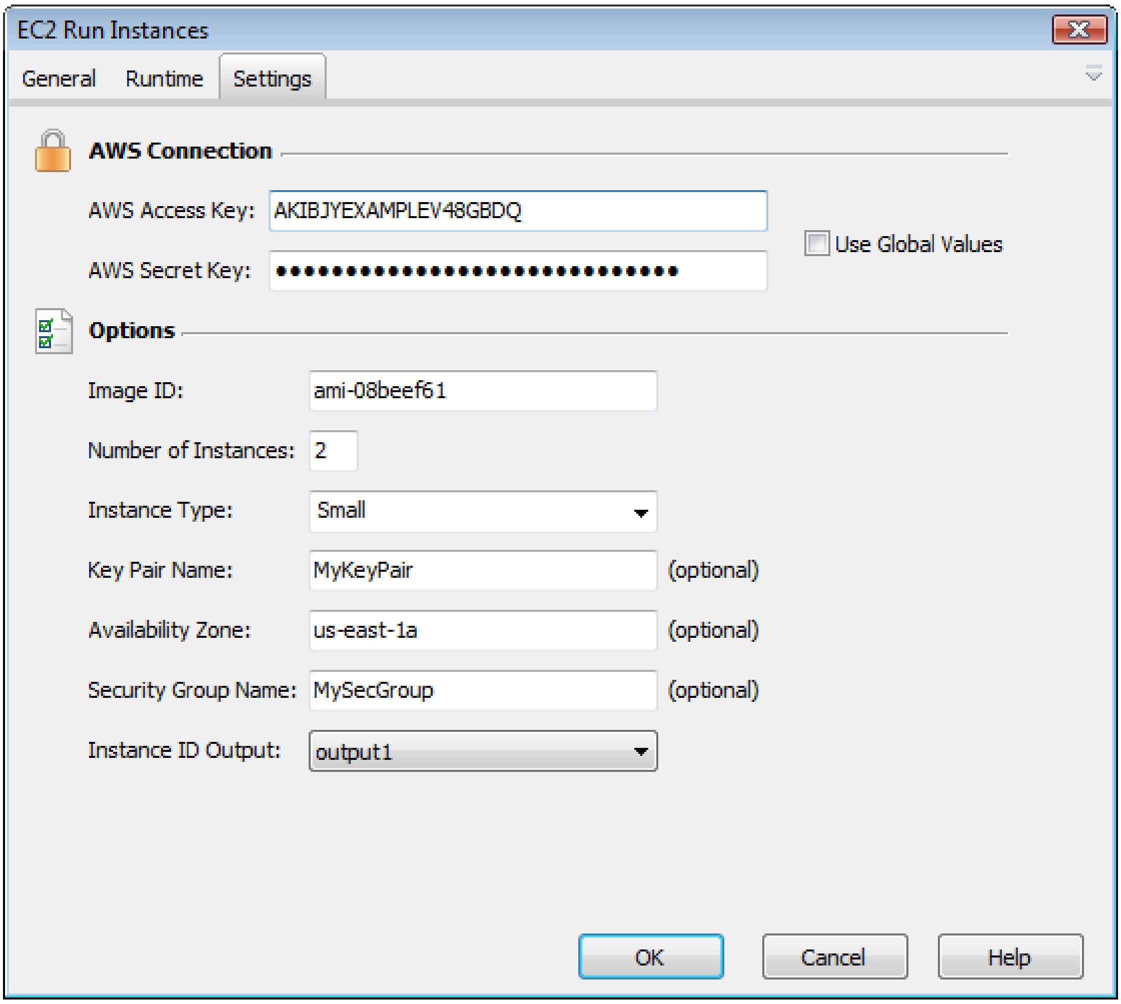
A comma seperated list of the IDs of the instances to change state.

State Change

The state change to apply to the specified instances. Choose between start, stop, and reboot.

6.15.16.3 EC2 Run Instances

This action enables you to run (create) one or more instances.


EC2 Run Instances

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Options

Image ID: ami-08beef61

Number of Instances: 2

Instance Type: Small

Key Pair Name: MyKeyPair (optional)

Availability Zone: us-east-1a (optional)

Security Group Name: MySecGroup (optional)

Instance ID Output: output1

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Image ID

The ID of the image to create the instance(s) from.

Number of Instances

The number of instances to create.

Instance Type

Type of instance to create. See http://aws.amazon.com/ec2/instance-types/ for information on the various instance types.

Key Pair Name

Optional. The name of a key pair that you have previously created via the AWS web interface. Needed if using a stock amazon AMI, and wish to access the instance via SSH.

Availability Zone

Optional. The name of the availability zone to create the new intstances in. If not specified, amazon will choose a zone for you. Use the Get Availability Zones action to get a list of available availability zones.

Security Group

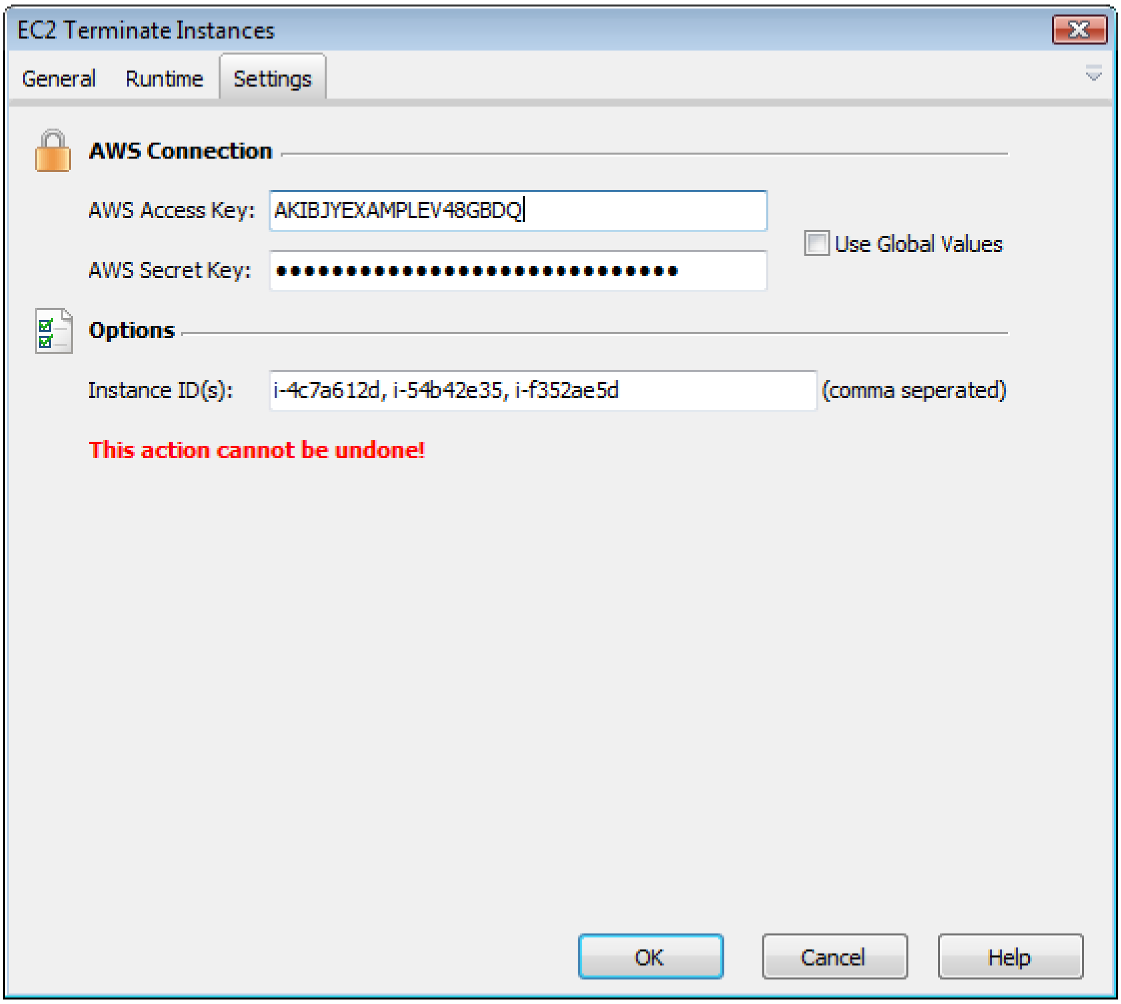
Optional. The name of a security group you have previously created via the AWS web interface. If not specifed, your default security group will be used.

Instance ID Output

The variable to return the ID(s) of the newly created instances to. Will return a comma seperated list if more than one instance was created.

6.15.16.4 EC2 Terminate Instance

This action enables you to terminate one or more instances. Note that this action is permanent and cannot be undone.



AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Instance ID(s): i-4c7a612d, i-54b42e35, i-f352ae5d (comma seperated)

This action cannot be undone!

AWS Connection

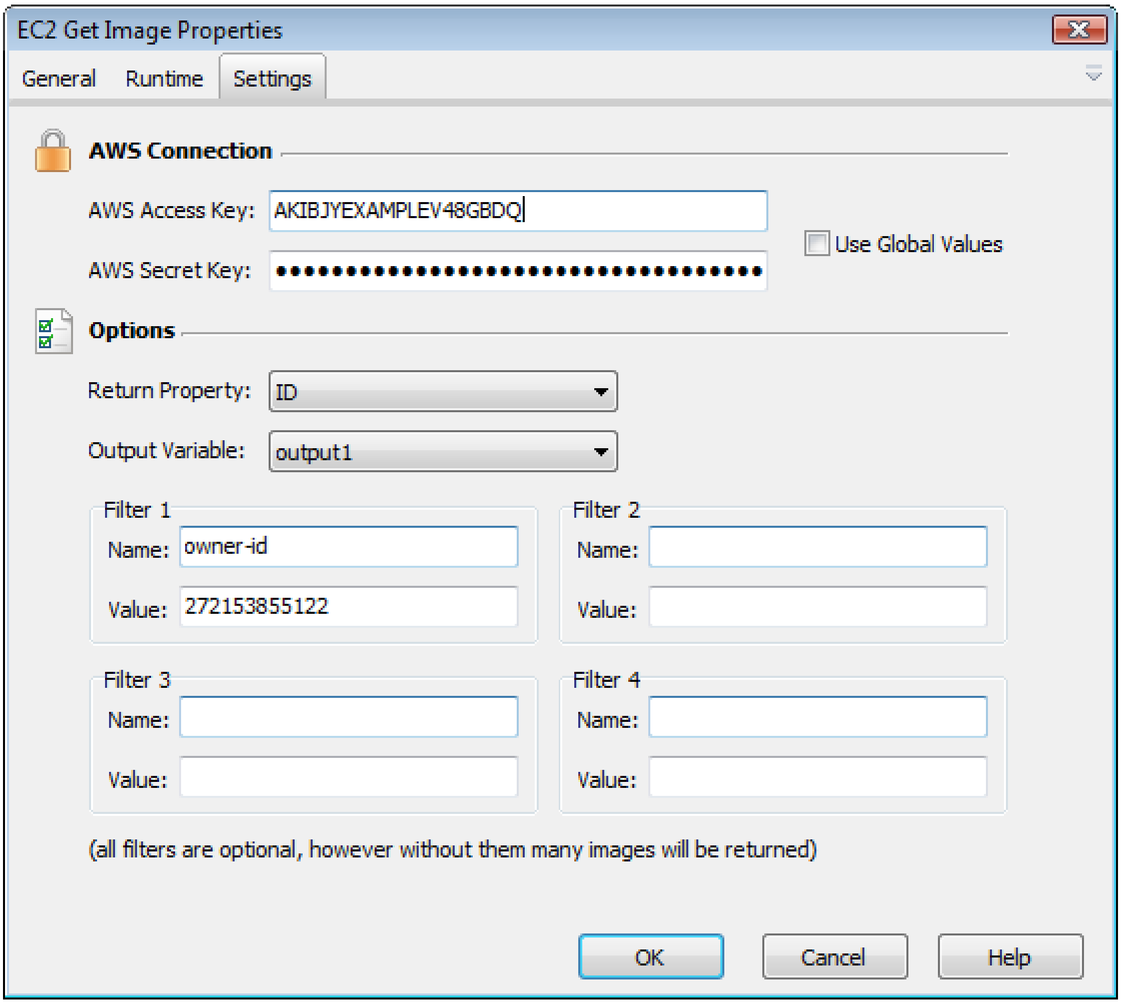
The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Instance IDs

A comma seperated list of the IDs of the instances to terminate.

6.15.16.5 EC2 Get Image Properties

This action allows you to retrieve the selected property of images available to you. You can filter which results to return using the optional filters.



EC2 Get Image Properties

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ●●

□ Use Global Values

Options

Return Property: ID

Output Variable: output1

Filter 1
Name: owner-id
Value: 272153855122

Filter 2
Name:
Value:

Filter 3
Name:
Value:

Filter 4
Name:
Value:

(all filters are optional, however without them many images will be returned)

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Return Property

The property of the images to return. Choose between ID, Owner ID, State, or Name.

Output Variable

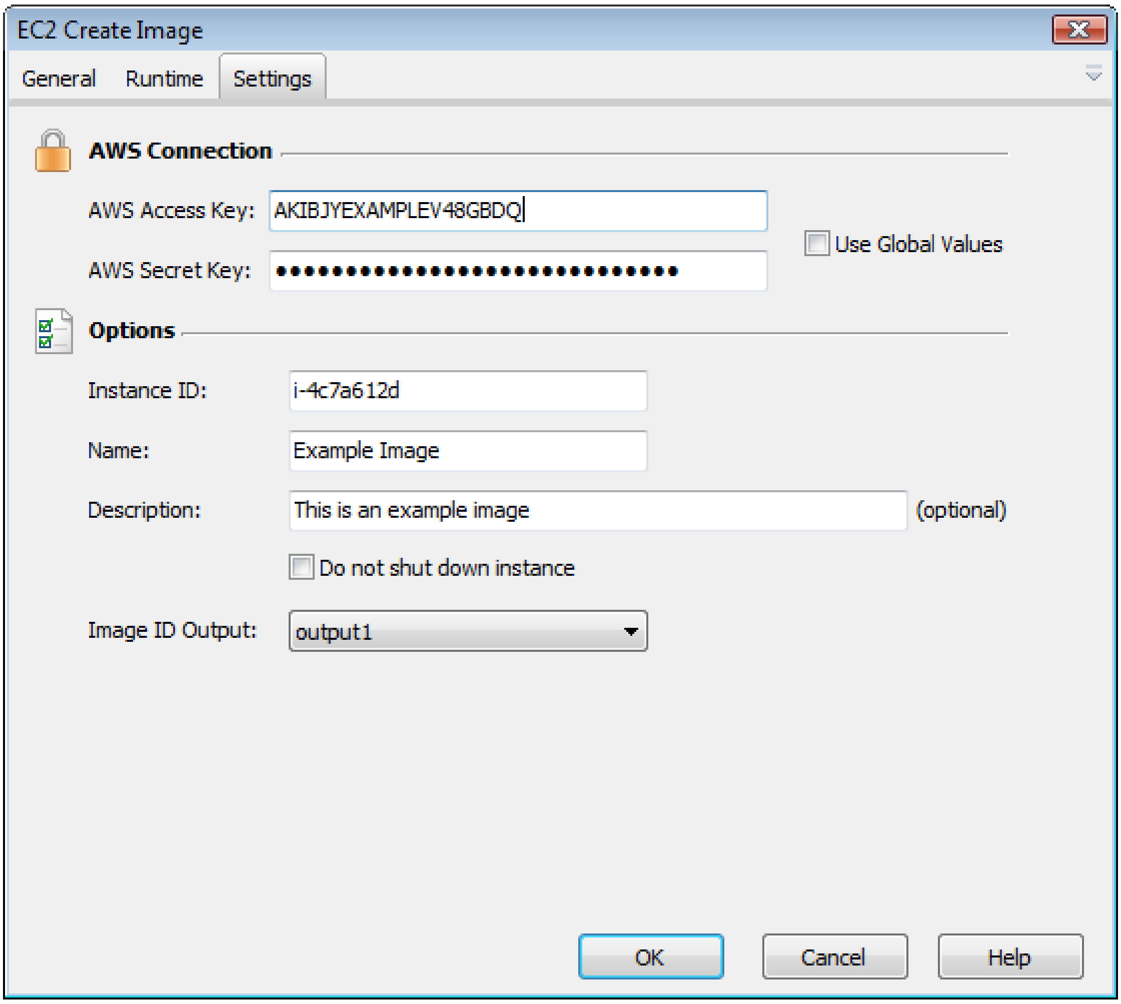
The variable to output the selected property to. If more than one image matches the filters, a comma seperated list of properties will be returned.

Filters

The name and value of any filters on which images will be returned. The filter name field will automatically complete with valid filter names. For a list of filters & valid values: http://docs.amazonwebservices.com/AWSEC2/2011-05-15/APIReferenceindex.html?ApiReference-query-DescribeImages.html

6.15.16.6 EC2 Create Image

This action enables you to create an image from an instance.



AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Options

Instance ID: i-4c7a612d

Name: Example Image

Description: This is an example image (optional)

□ Do not shut down instance

Image ID Output: output1

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Instance ID

The instance ID to create the image from.

Name

The name for the new image.

Description

The description for the new image.

Do not shut down instance

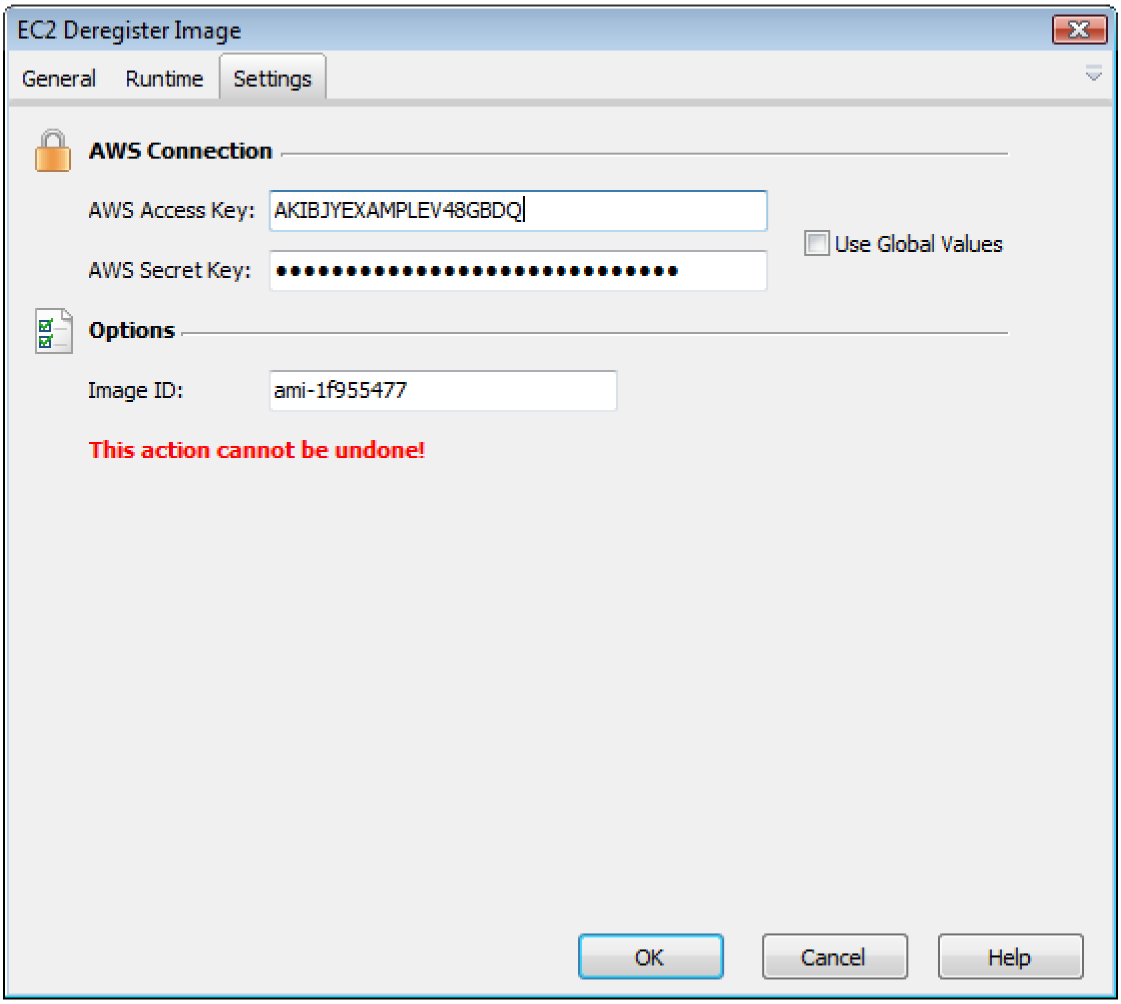
Checking this will create the image without stopping the instance first. This is not reccommended as it may cause I/O errors in the created image.

Image ID Output

The variable to return the ID of the newly created image to.

6.15.16.7 EC2 Deregister Image

This action enables you to deregister (delete) an image. Please note that this action is permanent and cannot be undone.



EC2 Deregister Image

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••

□ Use Global Values

Options

Image ID: ami-1f955477

This action cannot be undone!

AWS Connection

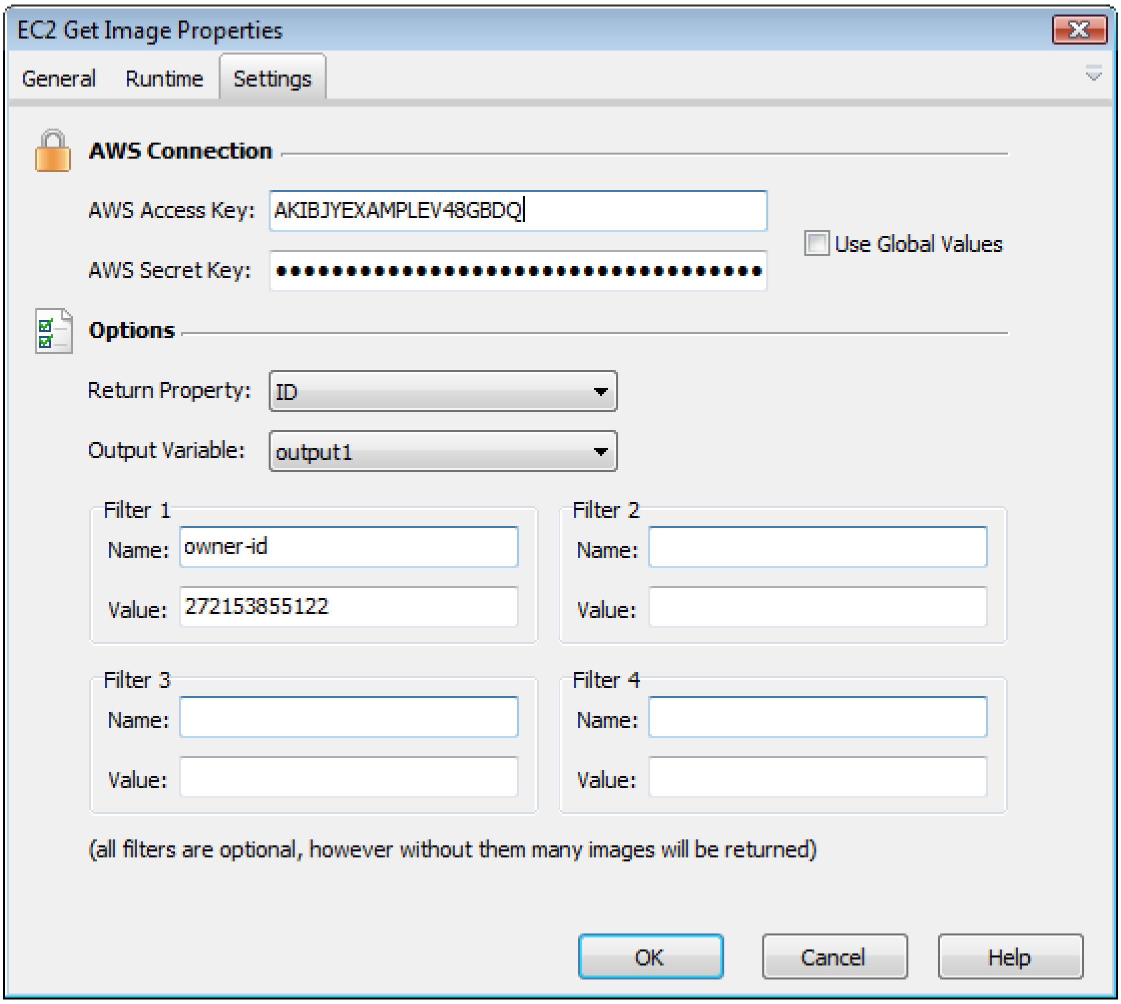
The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Image ID

The ID of the image you wish to deregister.

6.15.16.8 EC2 Get Volume Properties

This action allows you to retrieve the selected property of volumes available to you. You can filter which results to return using the optional filters.



EC2 Get Image Properties

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••

□ Use Global Values

Options

Return Property: ID

Output Variable: output1

Filter 1
Name: owner-id
Value: 272153855122

Filter 2
Name:
Value:

Filter 3
Name:
Value:

Filter 4
Name:
Value:

(all filters are optional, however without them many images will be returned)

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Return Property

The property of the volumes to return. Choose between ID, Status, Size, or Availability Zone.

Output Variable

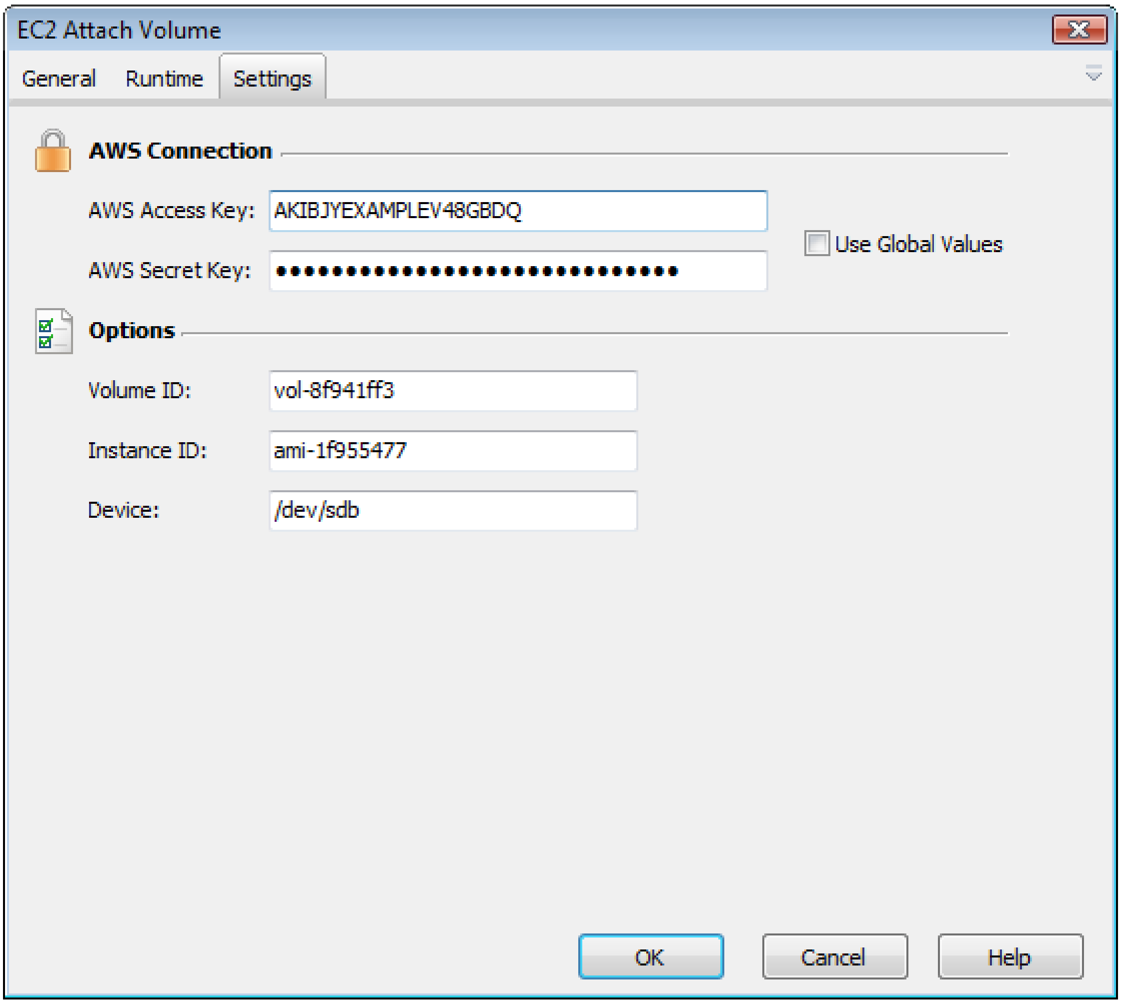
The variable to output the selected property to. If more than one volume matches the filters, a comma seperated list of properties will be returned.

Filters

The name and value of any filters on which volumes will be returned. The filter name field will automatically complete with valid filter names. For a list of filters & valid values: http://docs.amazonwebservices.com/AWSEC2/2011-05-15/APIReferenceindex.html?ApiReference-query-DescribeVolumes.html

6.15.16.9 EC2 Attach Volume

This action enables you to attach a volume to an instance.



AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Volume ID: vol-8f941ff3

Instance ID: ami-1f955477

Device: /dev/sdb

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Volume ID

The ID of the volume to attach.

Instance ID

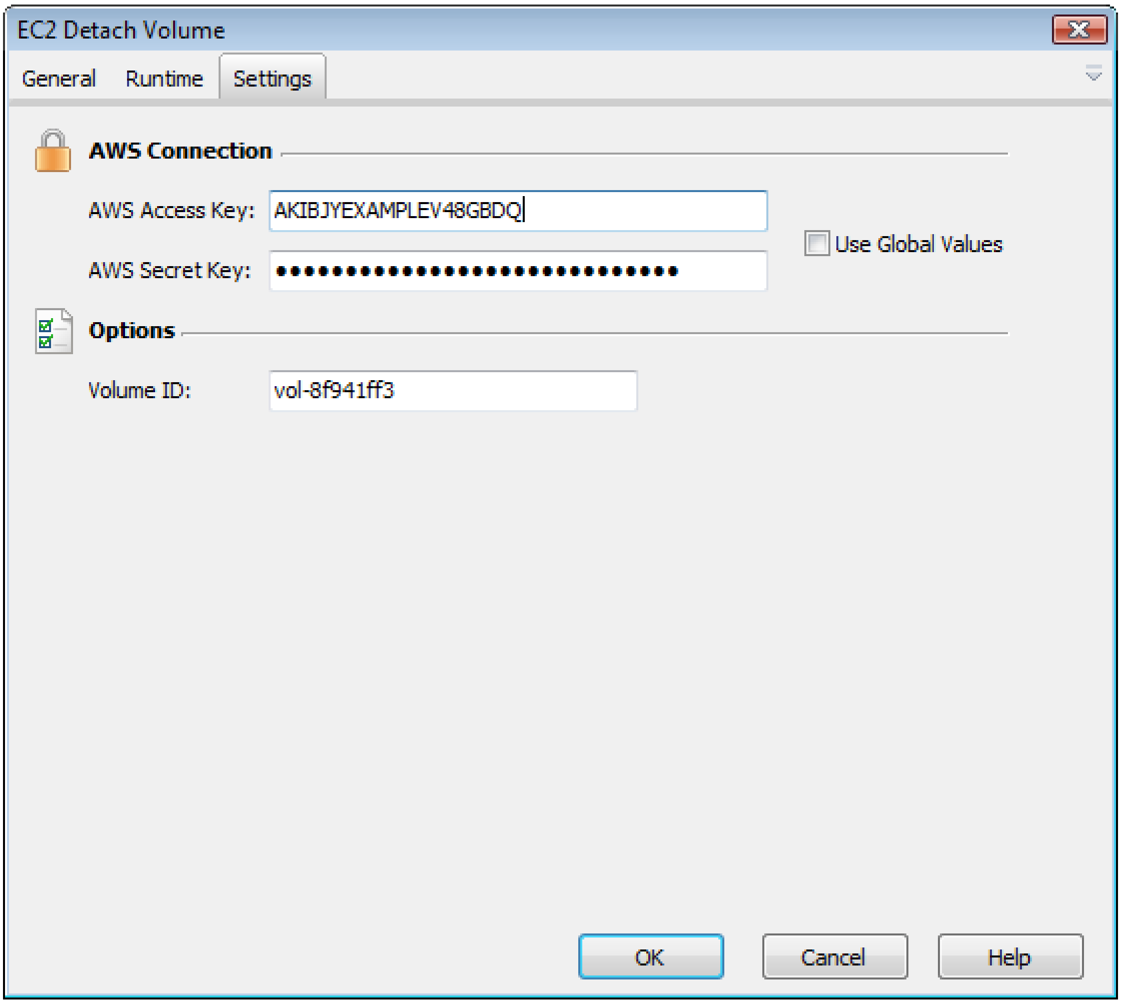
The ID of the instance to attach to.

Device

The device to expose the volume to the instance as. For example, /dev/sdh, or xvdh.

6.15.16.10 EC2 Detach Volume

This action enables you to detach a volume from any instances it is attached to.


EC2 Detach Volume

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Options

Volume ID: vol-8f941ff3

AWS Connection

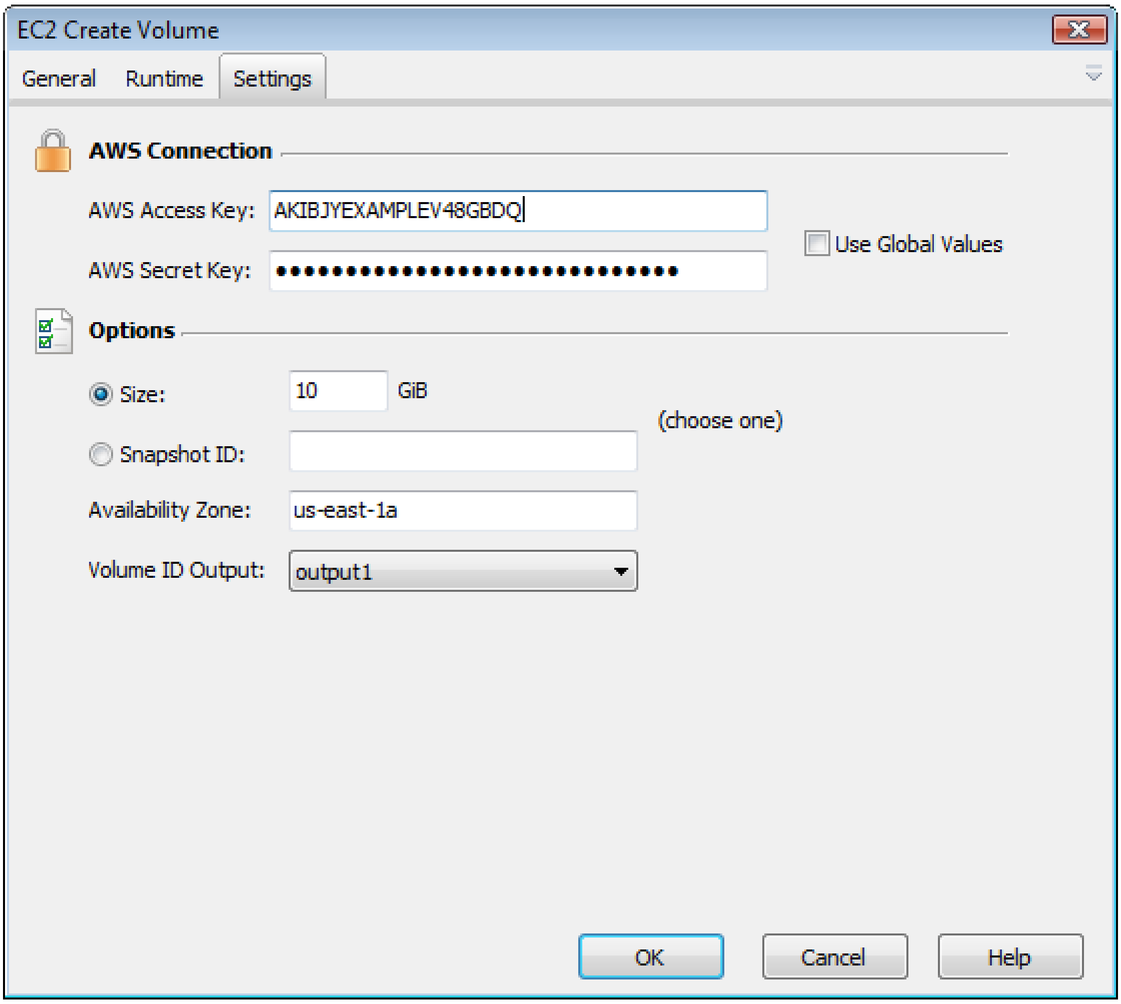
The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Volume ID

The ID of the volume that you wish to detach from all instances

6.15.16.1 EC2 Create Volume

This action enables you to create a volume, either as a blank volume of a specific size, or a volume restored from a snapshot.



AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••••

□ Use Global Values

● Size: 10 GiB

○ Snapshot ID:

Availability Zone: us-east-1a

Volume ID Output: output1

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Size

The size of the volume to create. Note: Use this OR a snapshot ID, not both.

Snapshot ID

The snapshot to create the volume from. The new volume will have the same size and contents as the original volume that the snapshot was made from. Note: Use this OR a size, not both.

Availability Zone

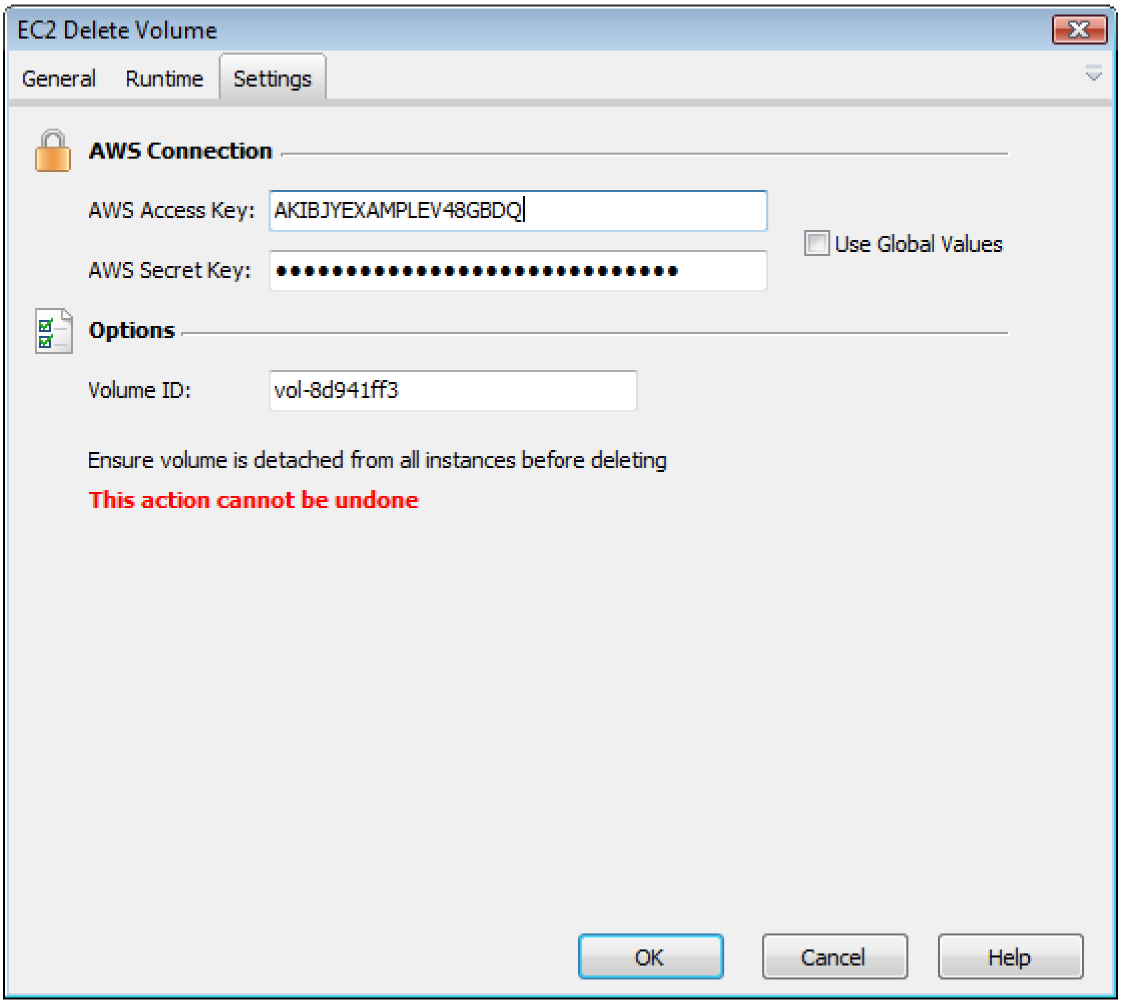
The availability zone for the volume to be created in. Use the Get Availability Zones action to get a list of availability zones that are available to you.

Volume ID Output

The variable to return the ID of the newly created volume to.

6.15.16.1:EC2 Delete Volume

This action enables you to delete a volume. Note that this action is permanent and cannot be undone.



EC2 Delete Volume

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Volume ID: vol-8d941ff3

Ensure volume is detached from all instances before deleting

This action cannot be undone

AWS Connection

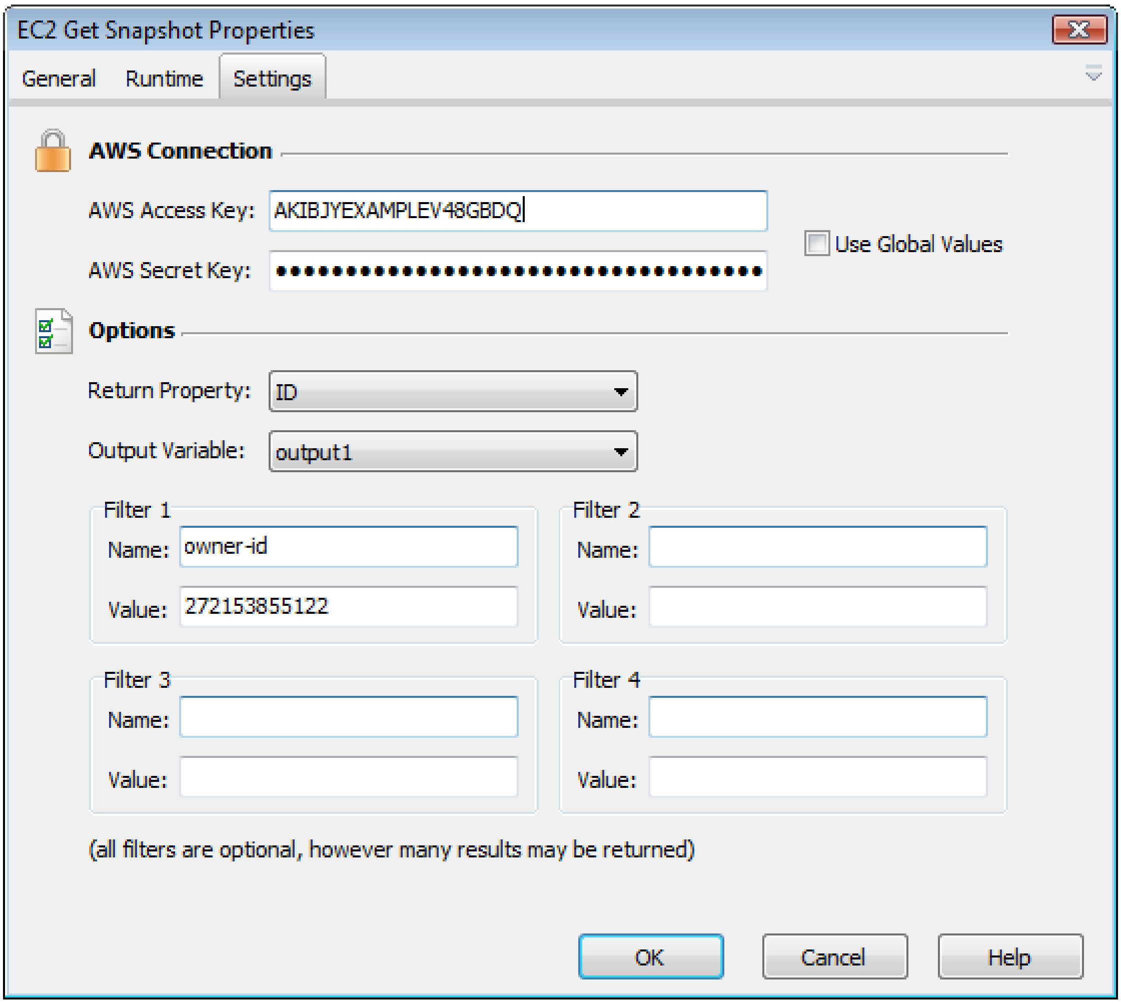
The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Volume ID

The ID of the volume that you wish to delete.

6.15.16.1: EC2 Get Snapshot Properties

This action allows you to retrieve the selected property of snapshots available to you. You can filter which results to return using the optional filters.



EC2 Get Snapshot Properties

General Runtime Settings

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ●●

□ Use Global Values

Options

Return Property: ID

Output Variable: output1

Filter 1
Name: owner-id
Value: 272153855122

Filter 2
Name:
Value:

Filter 3
Name:
Value:

Filter 4
Name:
Value:

(all filters are optional, however many results may be returned)

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Return Property

The property of the snapshots to return. Choose between ID, Volume ID, Status, Size, or Description.

Output Variable

The variable to output the selected property to. If more than one snapshot matches the filters, a comma seperated list of properties will be returned.

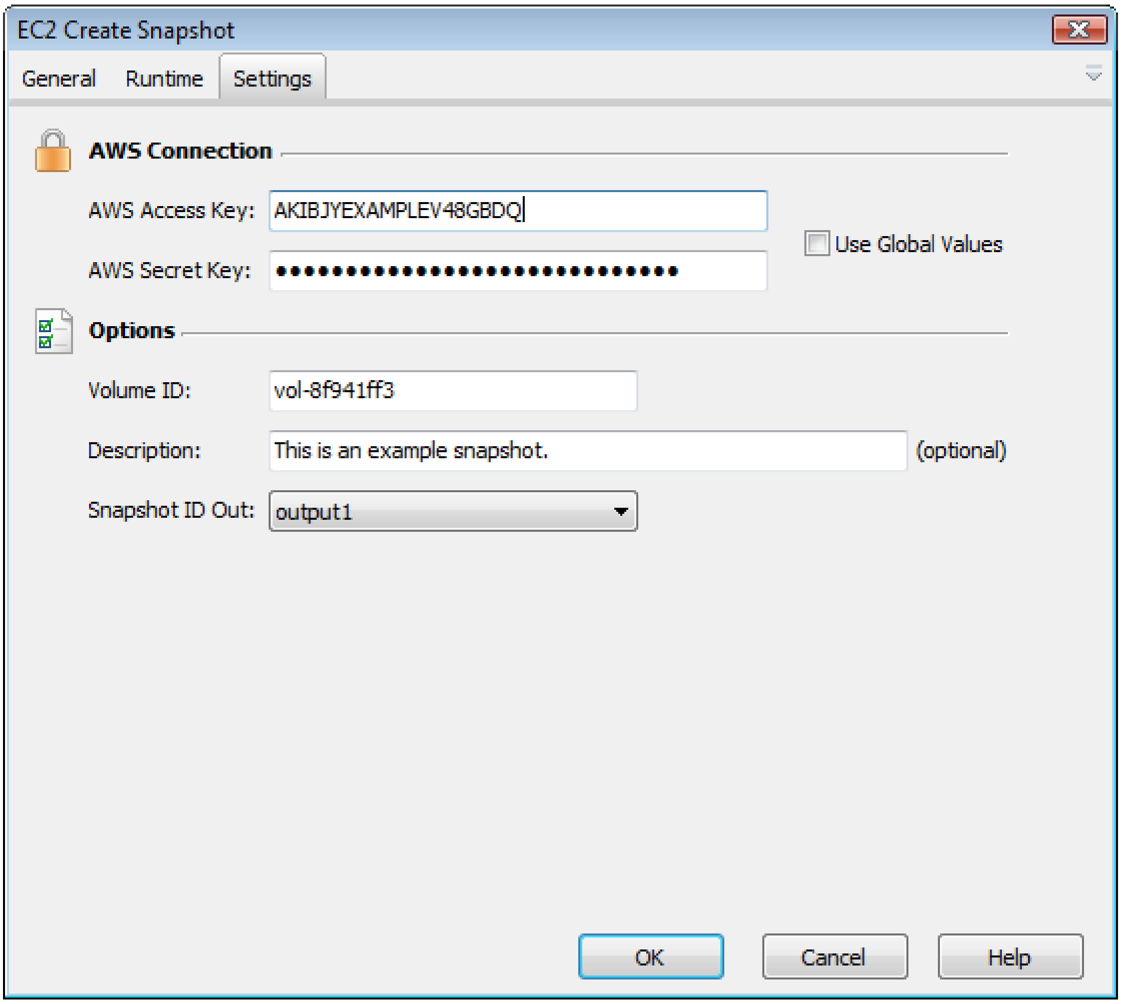
Filters

The name and value of any filters on which snapshots will be returned. The filter name field will automatically complete with valid filter names. For a list of filters & valid values: http://docs.amazonwebservices.com/AWSEC2/2011-05-15/APIReferenceindex.html?ApiReference-query-DescribeSnapshots.html

6.15.16.14 EC2 Create Snapshot

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.



AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Volume ID: vol-8f941ff3

Description: This is an example snapshot. (optional)

Snapshot ID Out: output1

Volume ID

The volume ID to create a snapshot of.

Description

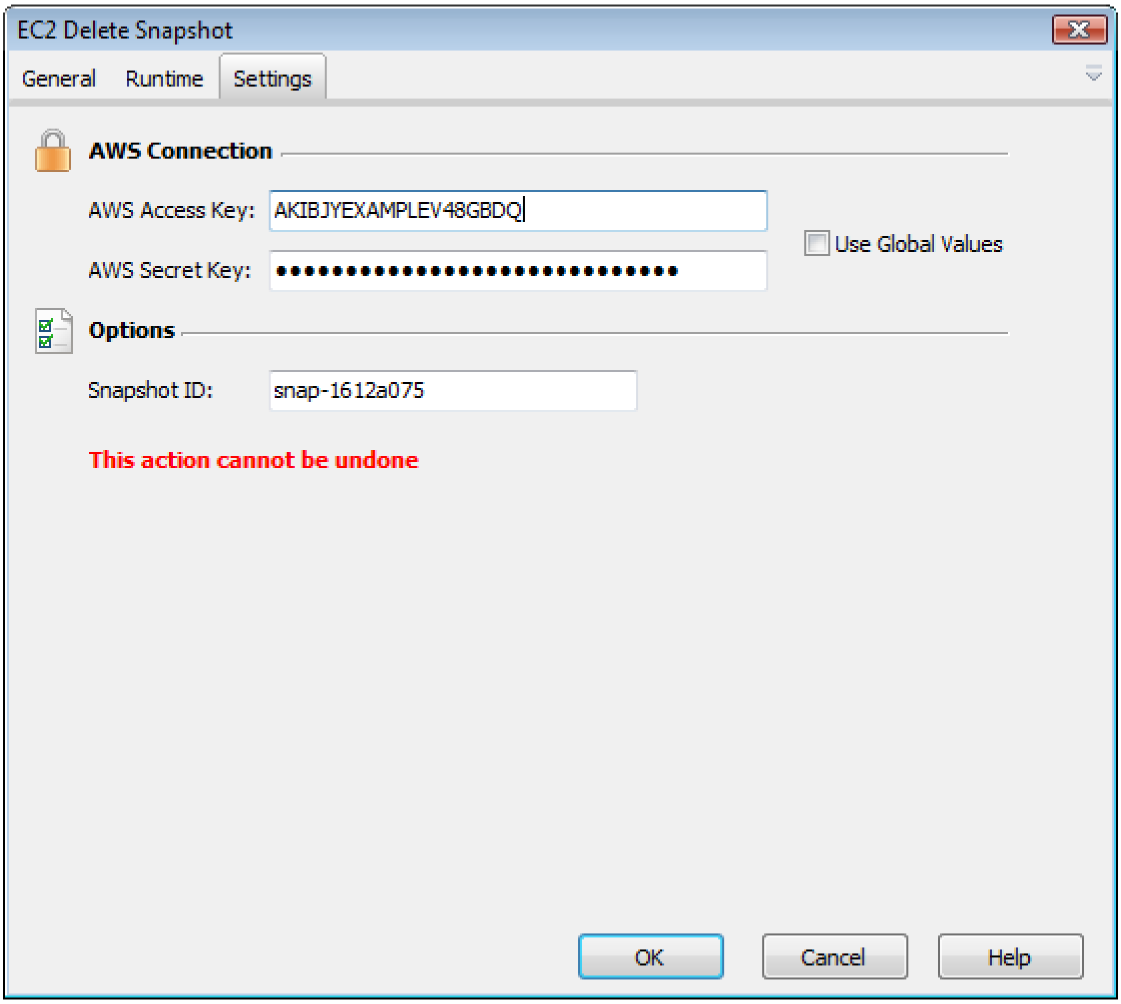
Optional. A description of the snapshot, visible in the AWS web interface, and available through the Get Snapshot Properties method.

Snapshot ID Output

The variable to return the ID of the newly created snapshot to.

6.15.16.1!EC2 Delete Snapshot

This action enables you to delete a snaphot. Note that this action is permanent and cannot be undone.



AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••••••••••••••••••••••••••••••

□ Use Global Values

Snapshot ID: snap-1612a075

This action cannot be undone

AWS Connection

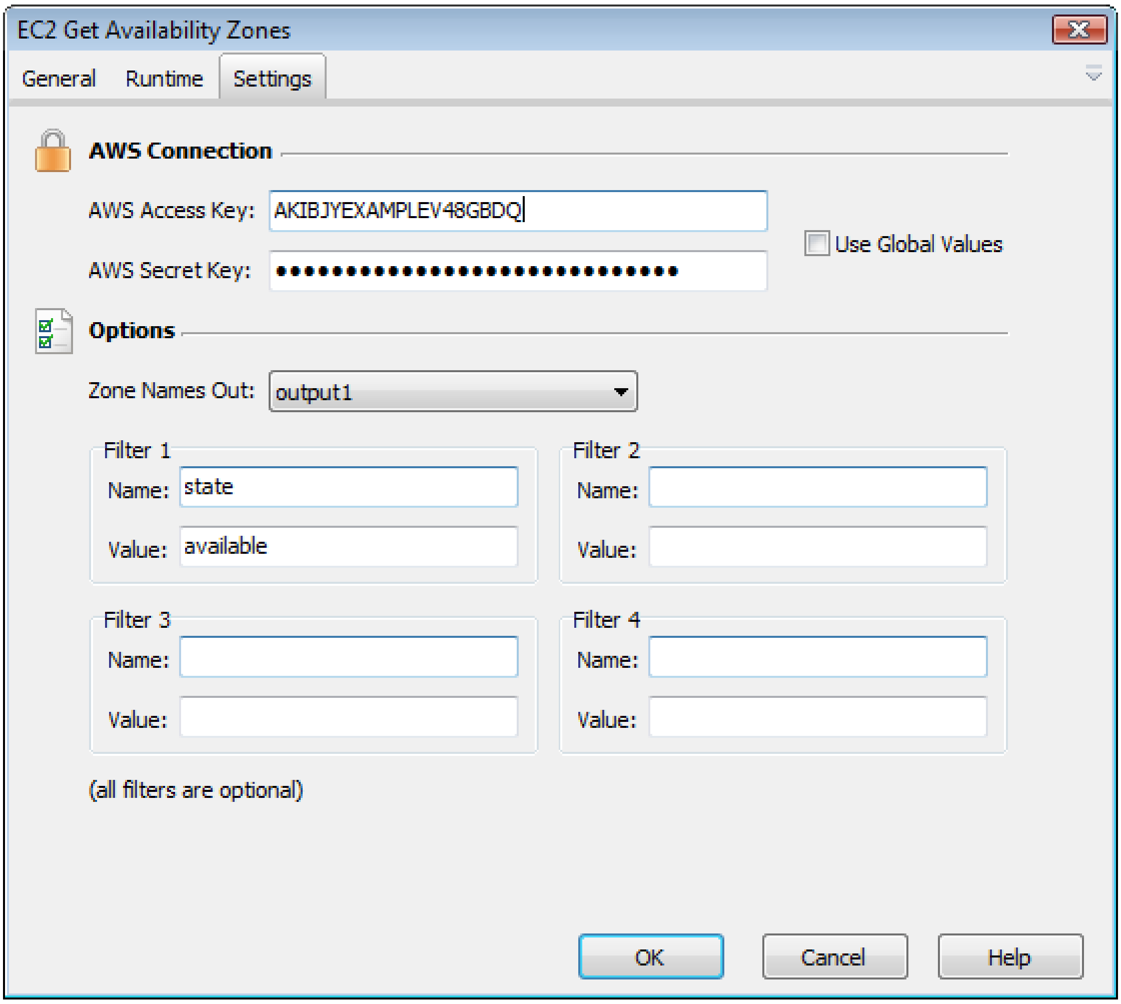
The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Snapshot ID

The snapshot ID to delete.

6.15.16.10 EC2 Get Availability Zones

This action allows you to retrieve the name of availability zones available to you. You can filter which results to return using the optional filters.



EC2 Get Availability Zones

General | Runtime | Settings

AWS Connection

AWS Access Key: AKIBJYEXAMPLEV48GBDQ

AWS Secret Key: ••

□ Use Global Values

Options

Zone Names Out: output1

Filter 1
Name: state
Value: available

Filter 2
Name:
Value:

Filter 3
Name:
Value:

Filter 4
Name:
Value:

(all filters are optional)

AWS Connection

The Access Key and Secret Key for your AWS account. If "Use Global Values" is checked, the values set in the options page (Tools > Options > Internet > EC2) will be used instead.

Output Variable

The variable to output the zone names to. If more than one zone matches the filters, a comma seperated list of zone names will be returned.

Filters

The name and value of any filters on which zones will be returned. The filter name field will automatically complete with valid filter names. For a list of filters & valid values: http://docs.amazonwebservices.com/AWSEC2/2011-05-15/APIReferenceindex.html?ApiReference-query-DescribeAvailabilityZones.html

6.16 Internet Information Services (IIS)

The IIS actions allow you to administer Microsoft IIS 5 and IIS 6 servers from your project. There are different sets of actions for IIS 5 and IIS 6.

The IIS 5 actions use ADSI (the Active Directory Service Interface) to connect to IIS. This means that only the currently logged in user's credentials can be used to connect to the server.

The IIS 6 actions use WMI (the Windows Management Instrumentation) interface to connect to IIS. This means that alternative credentials (ie username/password combinations) can be used to connect to IIS. In addition, there are some features which are only available in IIS 6. Generally, though, the two actions have identical feature sets.

Note that IIS 5 actions can still be used for connections to IIS 6, although IIS 6-only features will not be available.

6.16.1 IIS 5

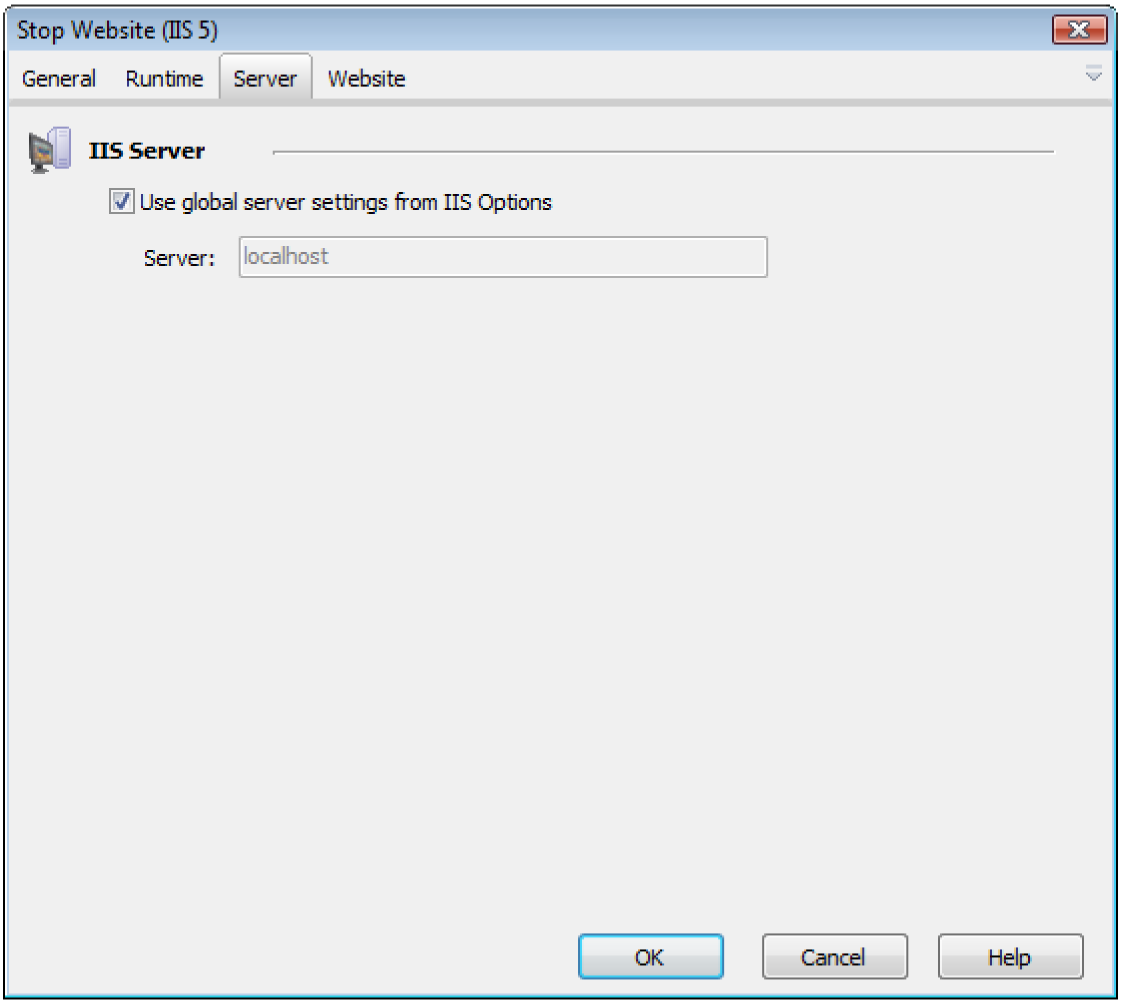
The IIS 5 actions use the ADSI interface to administer features of IIS.

These actions also work with IIS 6.0, but some IIS 6.0 administration features (such as user/password authentication) are not available.

6.16.1.1 Server Selection

For each IIS action, you can specify the IIS Server as either a global server (specified in the options) or a particular server name.

Note that the currently logged in user must have access privileges for the IIS server you wish to administer.



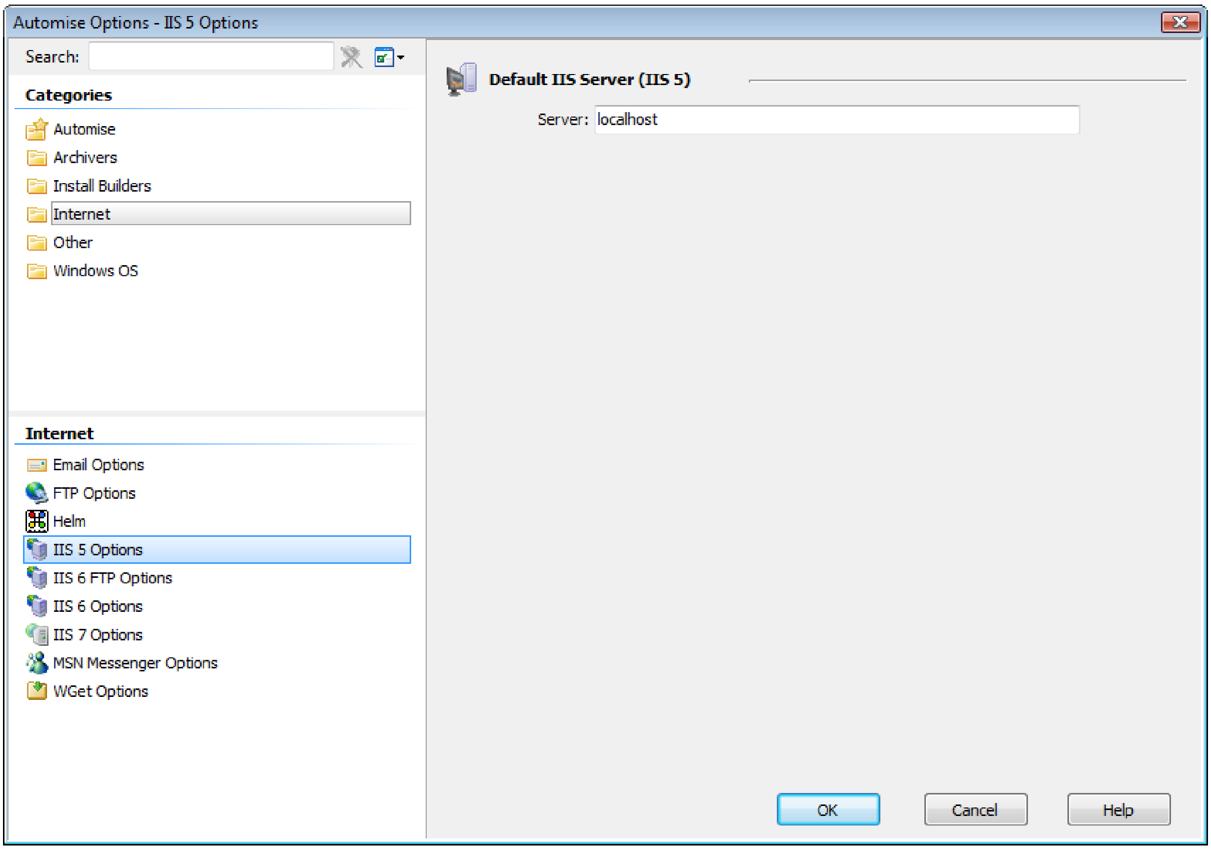
Stop Website (IIS 5)

IIS Server

☑ Use global server settings from IIS Options

Server: localhost

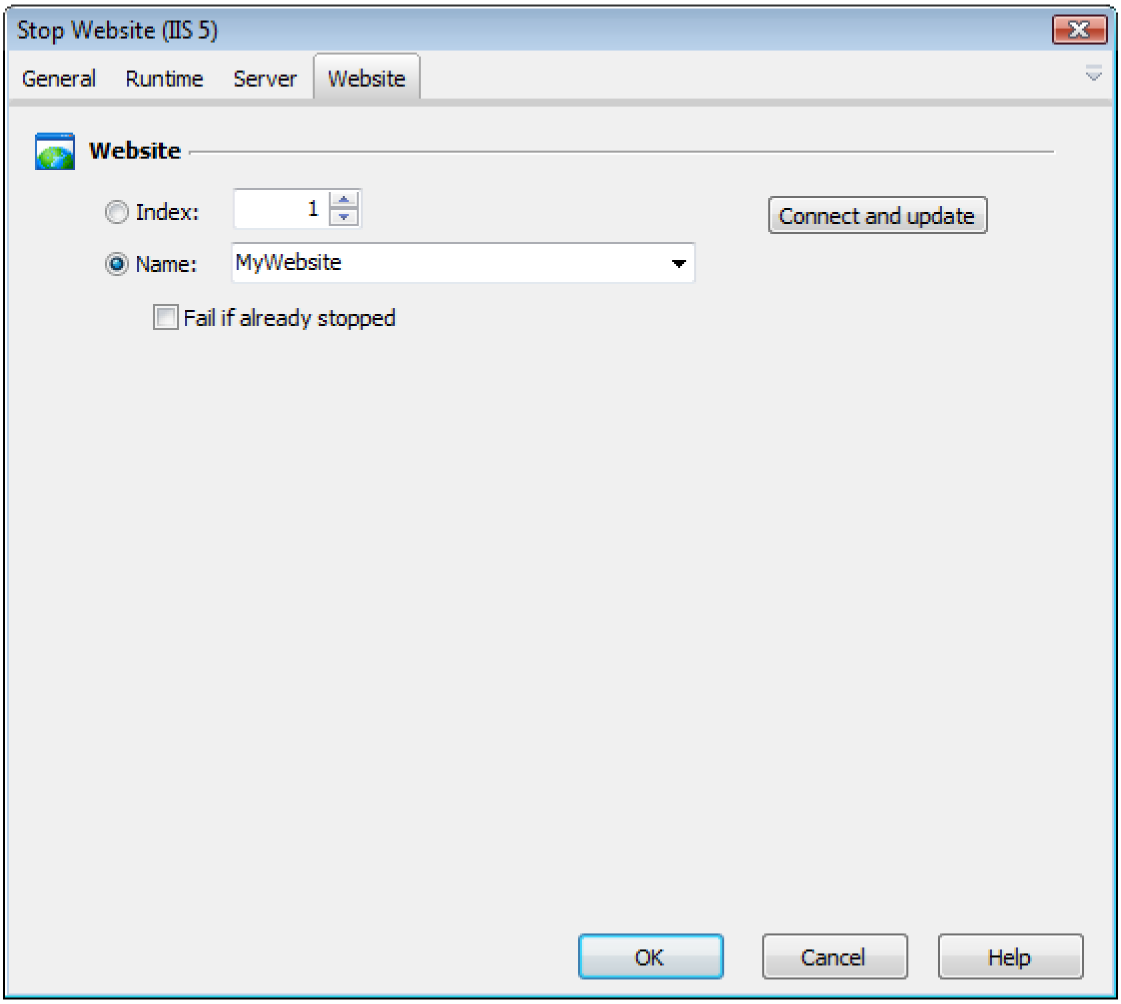
To select the global server name, choose Options from the Tools menu and then click on the IIS 5 Options tab under 'Internet'.



Server: localhost

6.16.1.2 Website Selection

You can specify IIS websites by index number (also called the Identifier, can be seen under the Web Sites option in the IIS Manager application) or by name (also called the Description.)



Stop Website (IIS 5)

Tabs: General, Runtime, Server, Website

Website

○ Index: 1

● Name: MyWebsite

□ Fail if already stopped

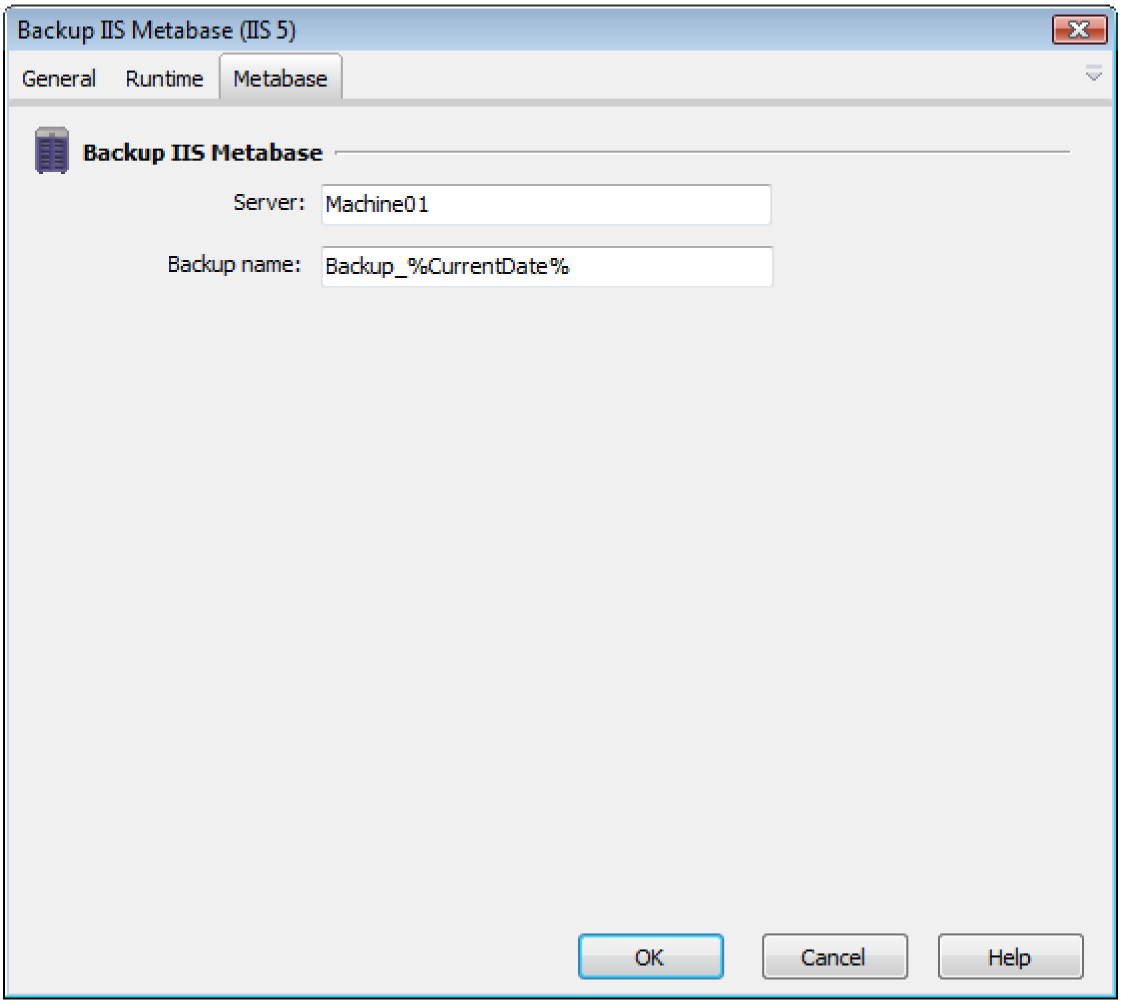
To connect to the server and update the list of available web site names, click "Connect and update." Alternatively, just type a name into the combo box.

The "**Fail if already...**" checkbox allows you to specify whether some actions will abort if a given action (ie stopping or starting a web site) has already been taken.

6.16.1.3 Backup IIS Metabase Action

The Backup IIS Metabase action creates an incremental backup of the IIS metabase configuration.

You can access existing metabase backups from inside IIS Manager, by right-clicking on the Computer name and choosing
"All Tasks->Backup/Restore Configuration".



Server: Machine01

Backup name: Backup_%CurrentDate%

Server

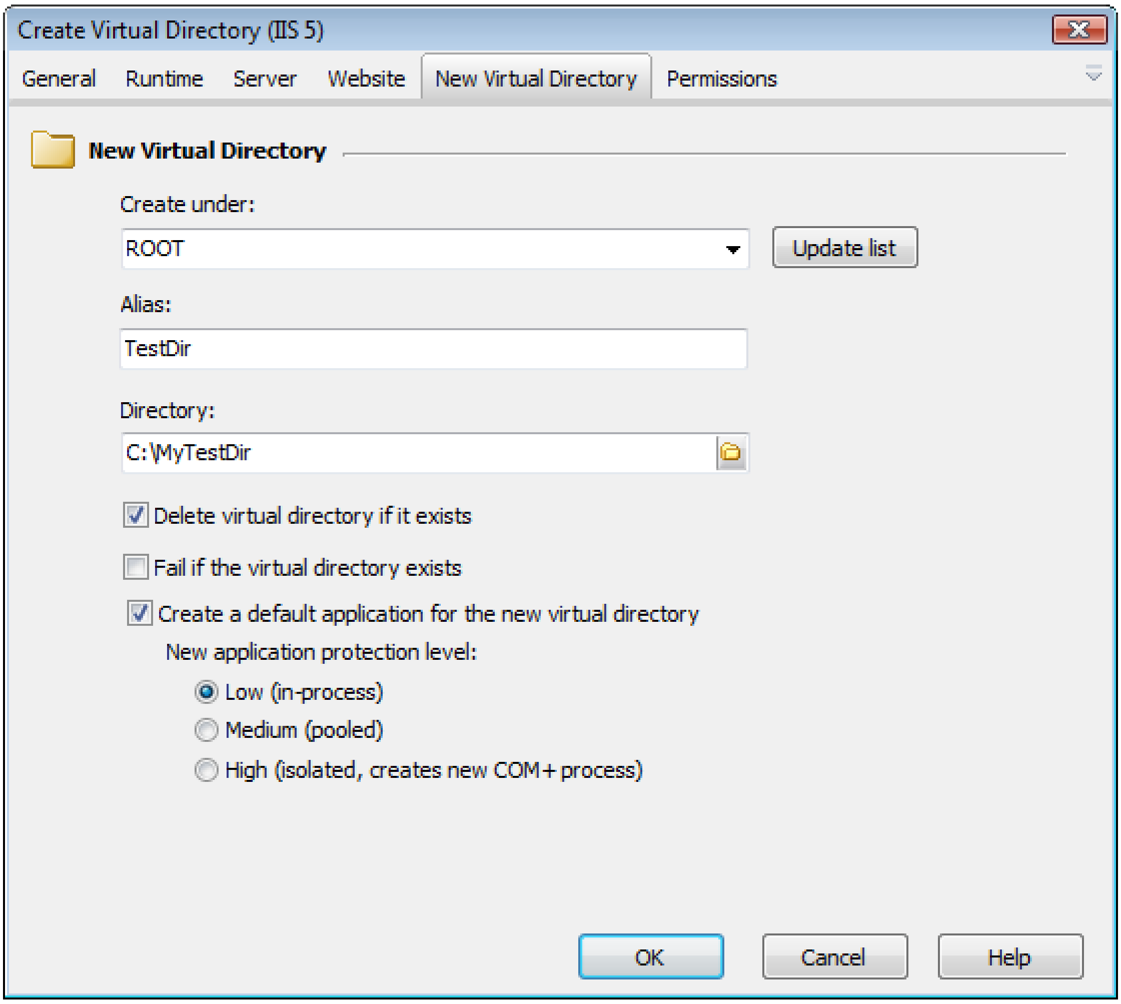
Specify the name of the server you wish to back up.

Backup name

Specify the name you wish to give the backup. Multiple backups with the same name will be given incremental backup version numbers (ie Automise IIS Backup 0, Automise IIS Backup 1, Automise IIS Backup 2...)

6.16.1.4 Create Virtual Directory Action

The Create Virtual Directory action allows you to add a new virtual directory to an existing IIS web site.



New Virtual Directory

Create under: ROOT

Alias: TestDir

Directory: C:\MyTestDir

☑ Delete virtual directory if it exists

☐ Fail if the virtual directory exists

☑ Create a default application for the new virtual directory

New application protection level:

● Low (in-process)

○ Medium (pooled)

○ High (isolated, creates new COM+ process)

Server & Website

Specify the Server and Website to use via the Server Selection & Website Selection tabs, respectively.

Create under

Specify an existing IIS Virtual Directory to be the parent of the new directory. Click the 'Update list' button to download a list
of Virtual Directories from the specified IIS Server.

Alias

Specify the name by which you would like the new directory to be known.

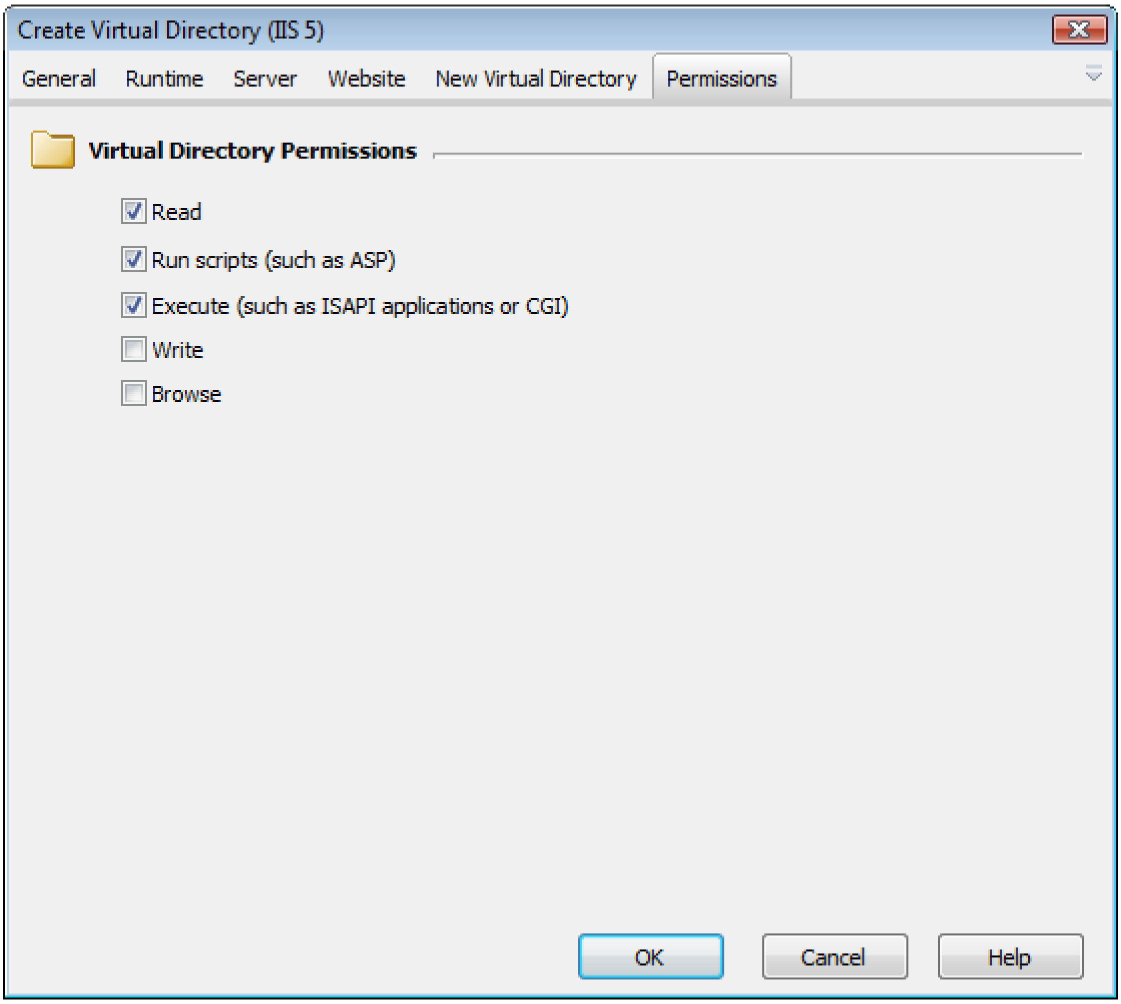
Directory

Specify the local directory for the contents of the new virtual directory.

Create a default application...

Check this box to create a default application in the new virtual directory. Check "create a Pooled application" to create a pooled application.

Permissions Tab



Create Virtual Directory (IIS 5)

Virtual Directory Permissions

☑ Read

☑ Run scripts (such as ASP)

☑ Execute (such as ISAPI applications or CGI)

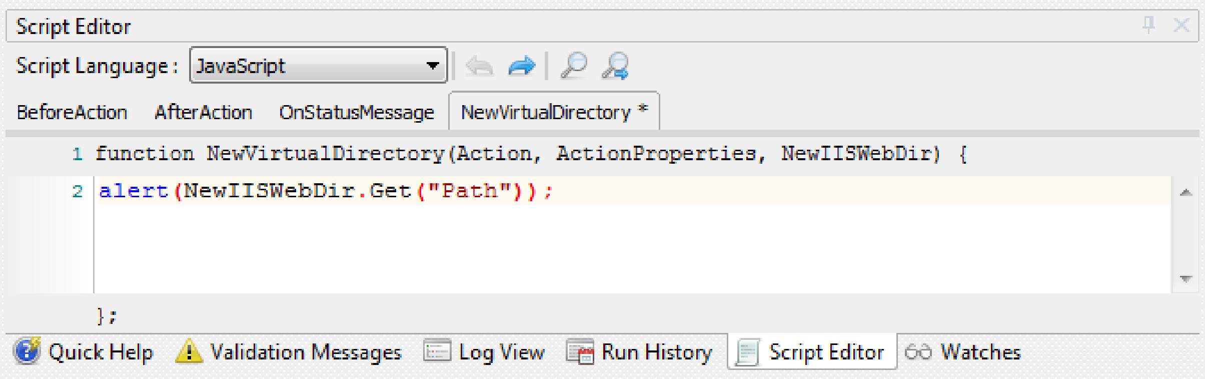
□ Write

□ Browse

Set the permissions you want IIS users to have on the new Virtual Directory.

Script Events

The Create Virtual Directory object has an extra script event, named NewVirtualDirectory, which is called once the new virtual directory has been set up. It allows you to set properties which may not be available in the action, or perform more complicated operations on the new directory.



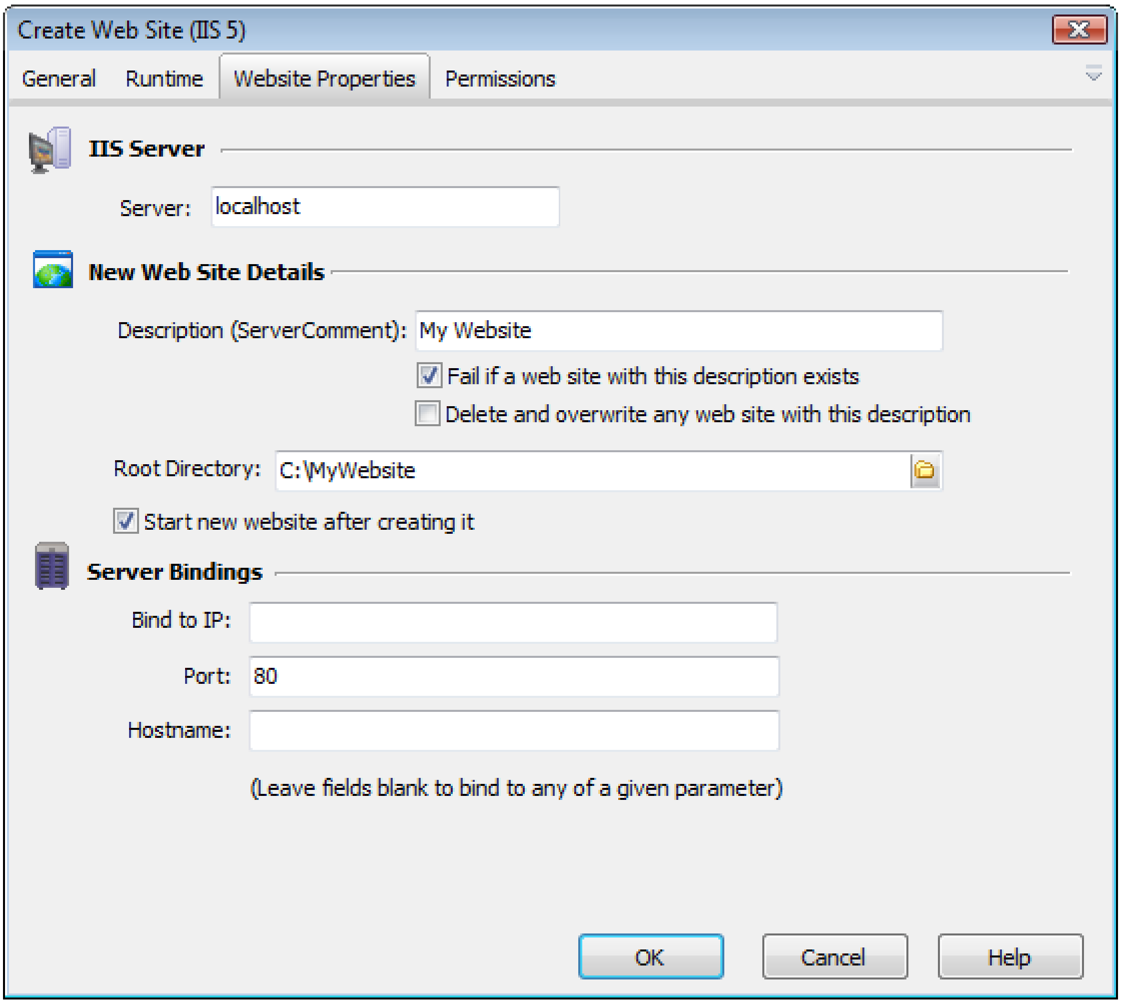
```
function NewVirtualDirectory(Action, ActionProperties, NewIISWebDir) {
alert(NewIISWebDir.Get("Path"));
};
```

The **NewIISWebDir parameter** is an ADSI container object is of type IIsWebVirtualDir (see the MSDN Documentation for details.) You will need to use the .Get() and .Put() methods to retrieve and set properties on the object. See the MSDN documentation for a full list of property names.

It is not necessary to call the .SetInfo() method to save any changes, this is called for you when the event exits.

6.16.1.5 Create Web Site Action

Use the IIS Create Web Site action to create a new web site. The Create Web Site action also creates a root virtual directory.



Server: localhost

Description (ServerComment): My Website

☑ Fail if a web site with this description exists

□ Delete and overwrite any web site with this description

Root Directory: C:\MyWebsite

☑ Start new website after creating it

Bind to IP:

Port: 80

Hostname:

(Leave fields blank to bind to any of a given parameter)

IIS Server

Choose the hostname of the IIS Server on which to create the web site.

New Web Site Details

Description - Description specifies a ServerComment property for the web site (in other words its name.) Web sites are identified either by this property or their index number.

Fail if a web site... - If this option is checked, the action will fail if another web site already has this description.

Delete and overwrite... - If this option is checked, a web site with this name will be overwritten and replaced with the new site. The new web site will have the same index number as the old site.

If neither *Fail* nor *Delete* is selected, a new web site will be created even if it has the same description as a pre-existing one.

Root Directory - Specify a directory local to the server on which to create the root virtual directory. Set the permissions for this directory on the Permissions tab.

Start new web site after creating it - If this option checked, the new web site will be switched to the Running state after creation.

Note that even if this fails (most likely because another running web site is bound to the same host/server/port), the new web site is still created (in the Stopped state), but the action will fail.

For fine-grained control over failure (ie for use in Try...Catch blocks), use a separate Start Web Site action.

Server Bindings

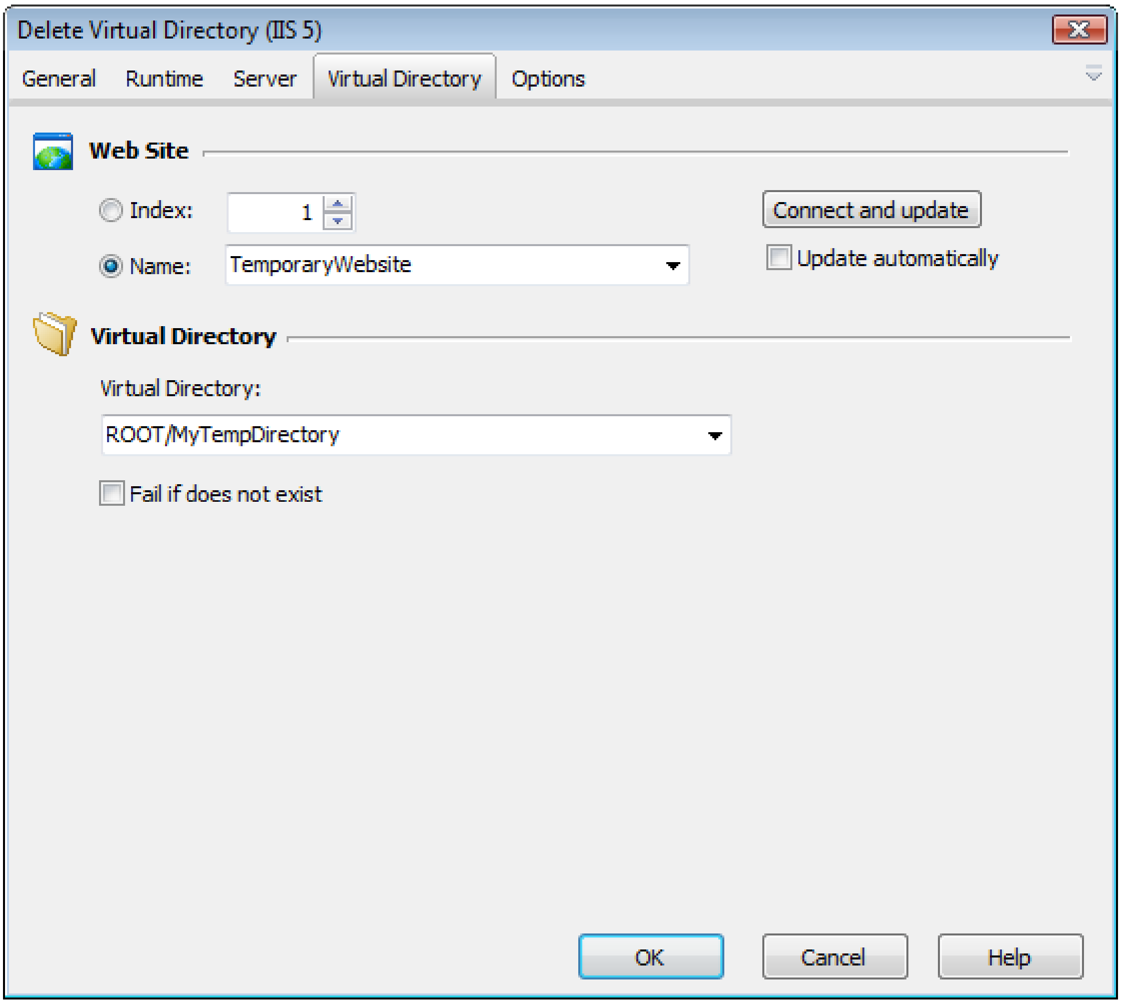
You can specify a specific IP, hostname, or Port combination to bind the new server to. Leave values blank to match all-inclusive wild cards.

Note that if any of these values overlap with an existing web site, the two cannot be run concurrently.

Permissions Tab

Set the permissions for the new web site's root Virtual Directory.

6.16.1.6 Delete Virtual Directory Action



○ Index: 1

● Name: TemporaryWebsite

□ Update automatically

Virtual Directory: ROOT/MyTempDirectory

□ Fail if does not exist

Web Site

Choose a web site name or index, as per the Website Selection tab. Check the "Update automatically" box to have changes in the web site selection automatically reload the virtual directory listing.

Virtual Directory

Choose the virtual directory to delete.

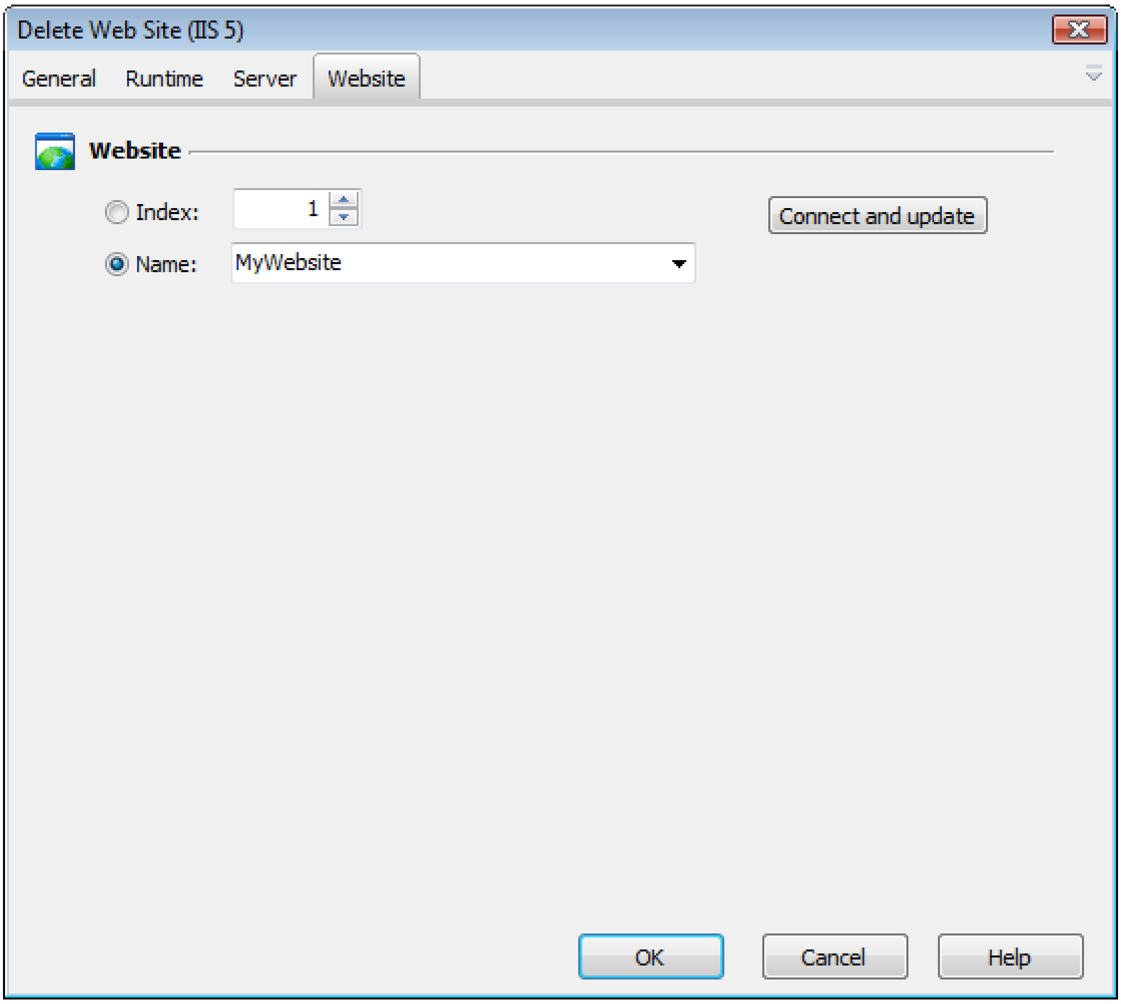
Fail if does not exist

Check this box if you want the action to fail should the specified virtual directory not exist.

6.16.1.7 Delete Web Site Action

The Delete Web Site action allows you remove a web site from the IIS metabase.

Use the Server Selection tab to choose the IIS server to delete the web site from, and the Website Selection tab to choose the web site to delete.



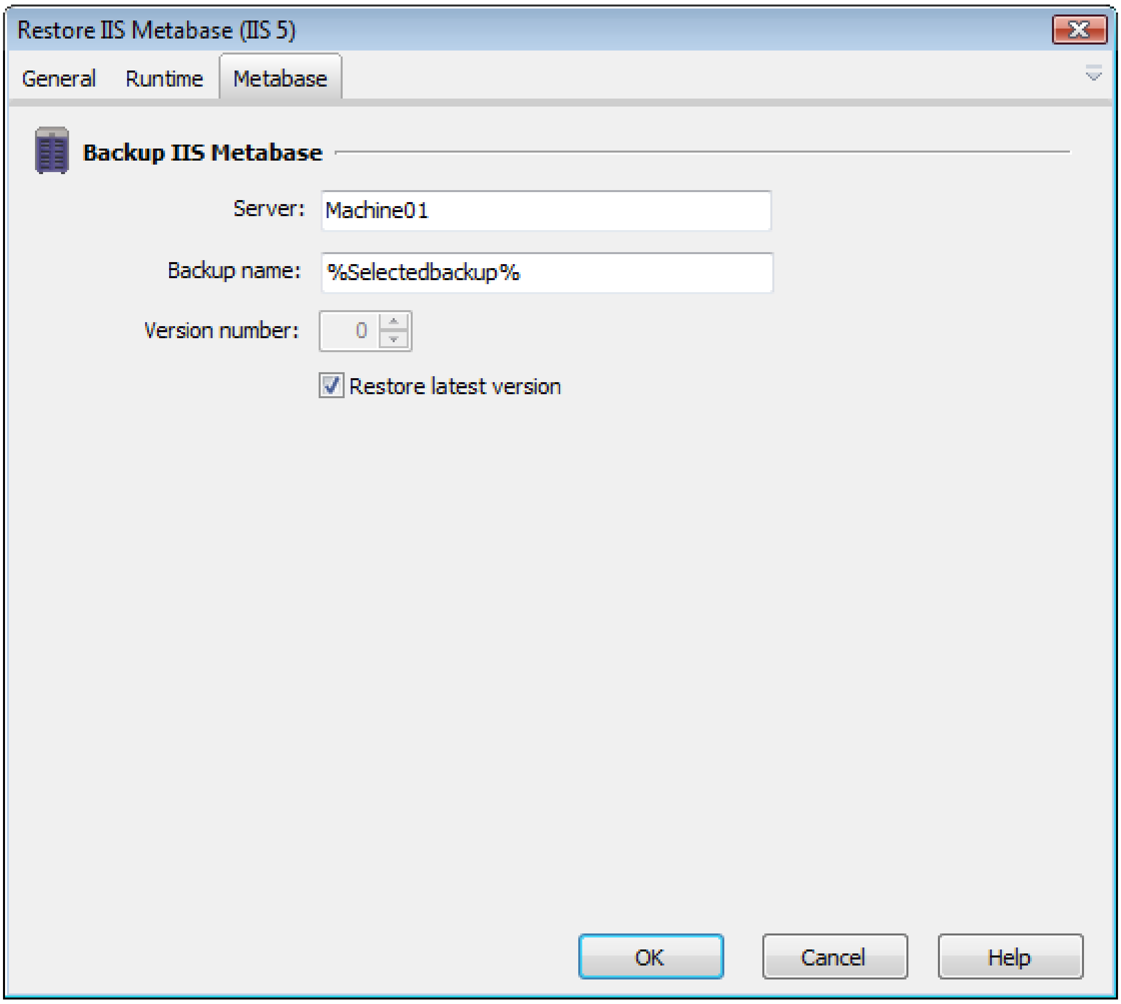
○ Index: 1

● Name: MyWebsite

6.16.1.8 Restore IIS Metabase Action

The Restore IIS Metabase action allows you restore a previously backed up IIS Metabase configuration.

You can access existing metabase backups from inside IIS Manager, by right-clicking on the Computer name and choosing
"All Tasks->Backup/Restore Configuration".



Server: Machine01

Backup name: %Selectedbackup%

Version number: 0

☑ Restore latest version

Server

Specify the name of the server with the metabase you wish to restore.

Backup name

Specify the name of the backup you wish to restore

Version Number

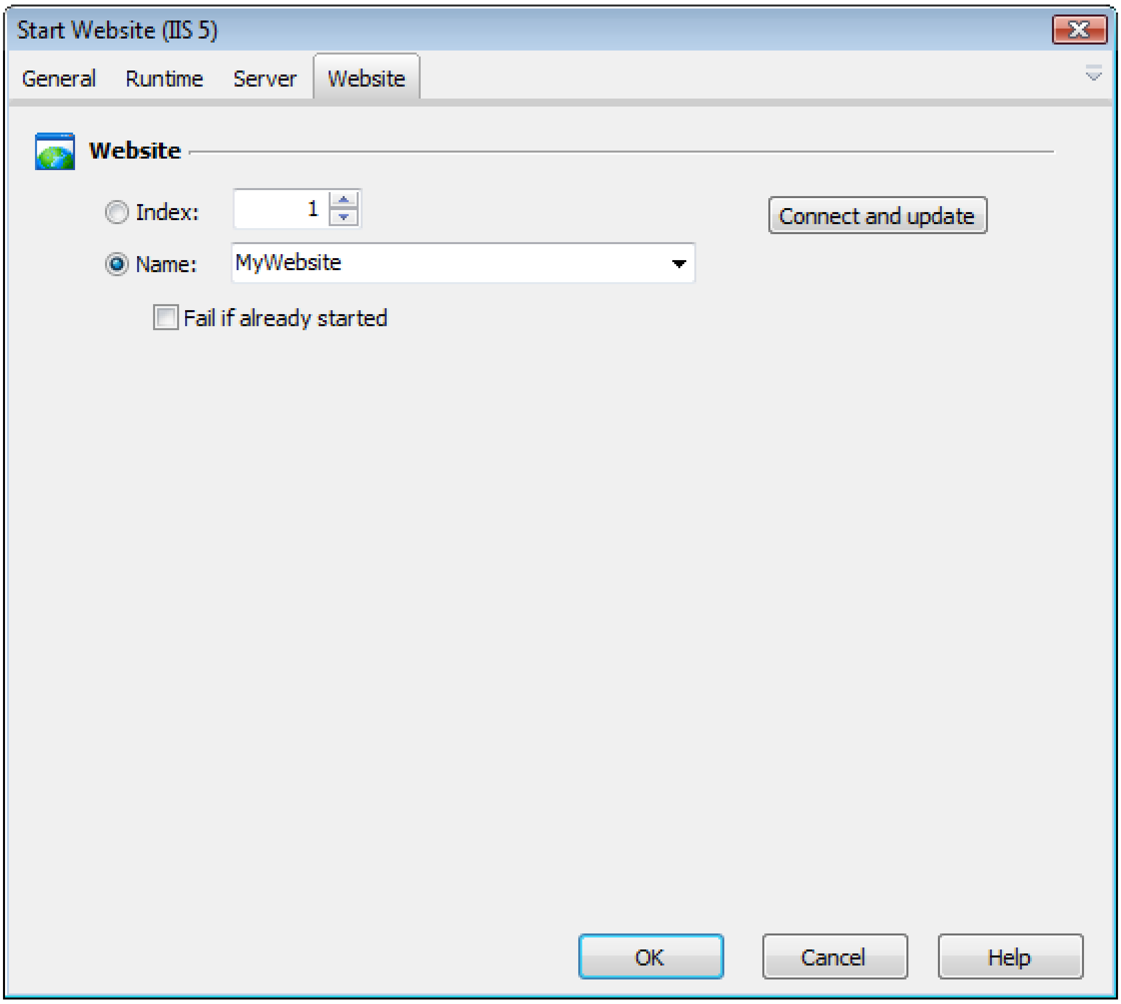
Specify a specific version number to restore, or choose "Restore latest version" to automatically find the latest backup and restore it.

6.16.1.9 Start / Stop / Pause Website Actions

The Start, Stop & Pause Website actions allow you to take individual IIS websites on- and off-line.

Use the Server Selection tab to choose the server you wish to administrate, then use the Website Selection tab to choose the web site you wish to modify.

The actions can be set not to fail if the selected web site is already in the desired state (running/stopped.)



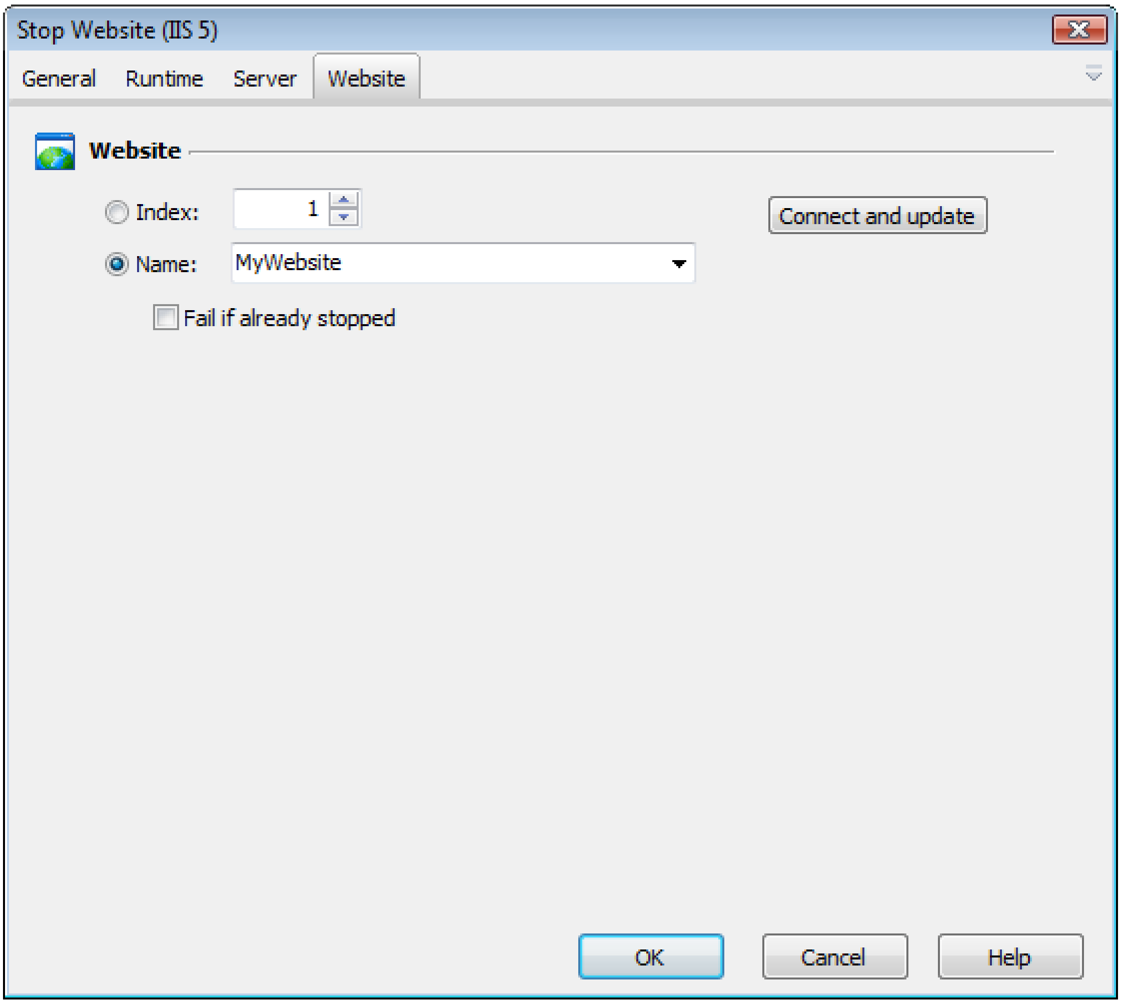
Start Website (IIS 5)

Website

○ Index: 1

● Name: MyWebsite

□ Fail if already started



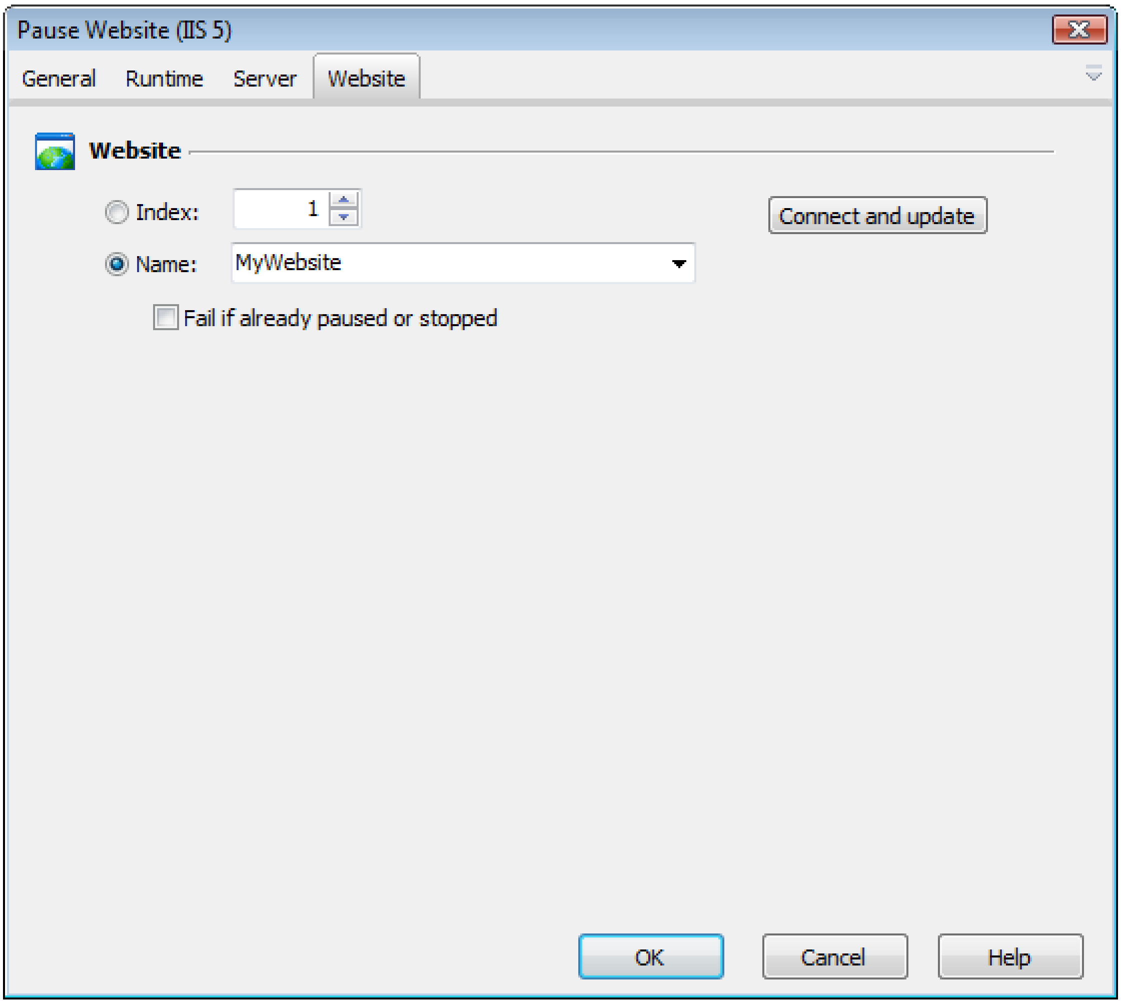
Stop Website (IIS 5)

Website

○ Index: 1

● Name: MyWebsite

□ Fail if already stopped



○ Index: 1

● Name: MyWebsite

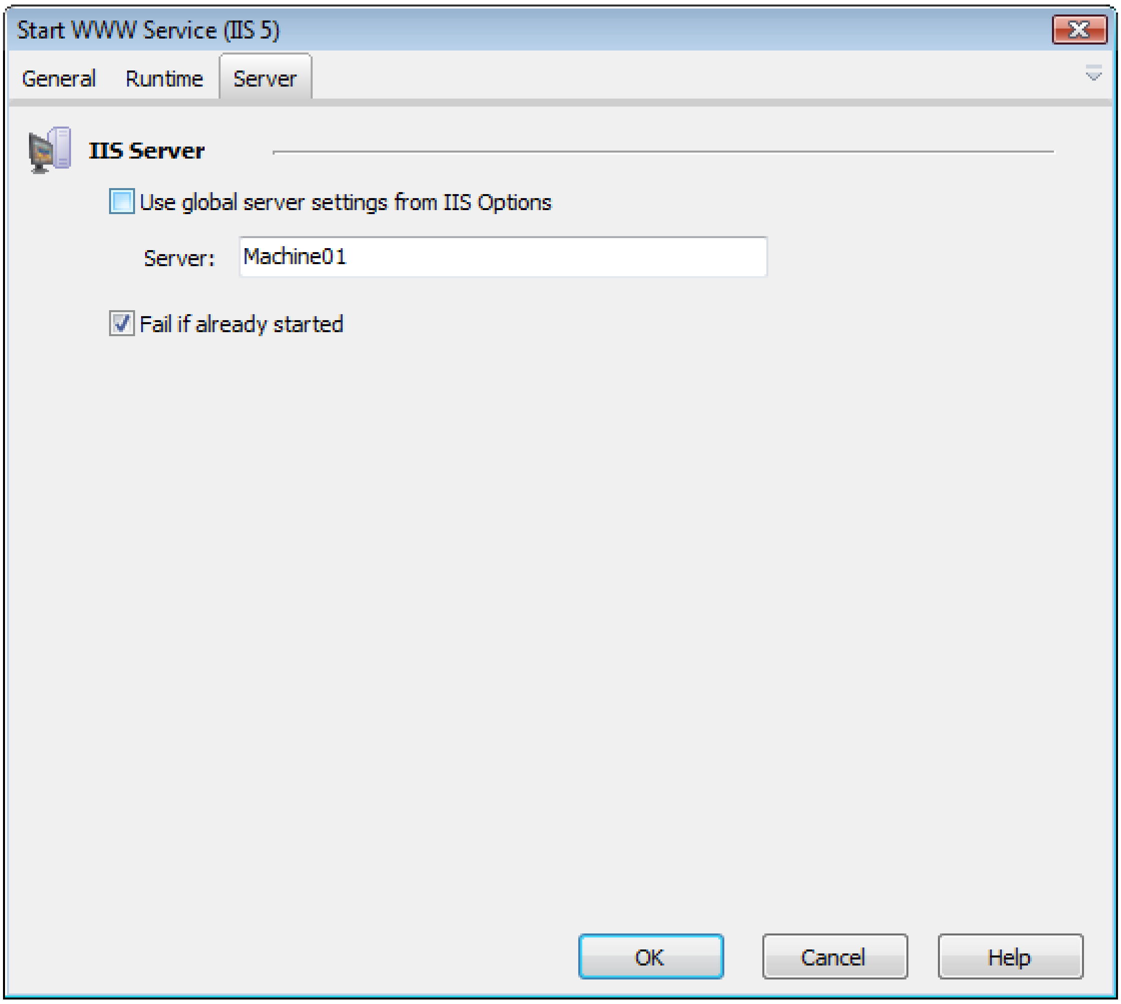
□ Fail if already paused or stopped

6.16.1.10 Start / Stop / Restart WWW Service Actions

The Start, Stop & Restart WWW Service actions use WMI to start and stop the IIS WWW service (W3SVC) on a server running IIS.

Use the Server Selection tab to choose the server you wish to administrate.

Each of the actions can be set not to fail if the service is already started/stopped (in the case of the Restart action, the service will be started even if it is initially stopped.)



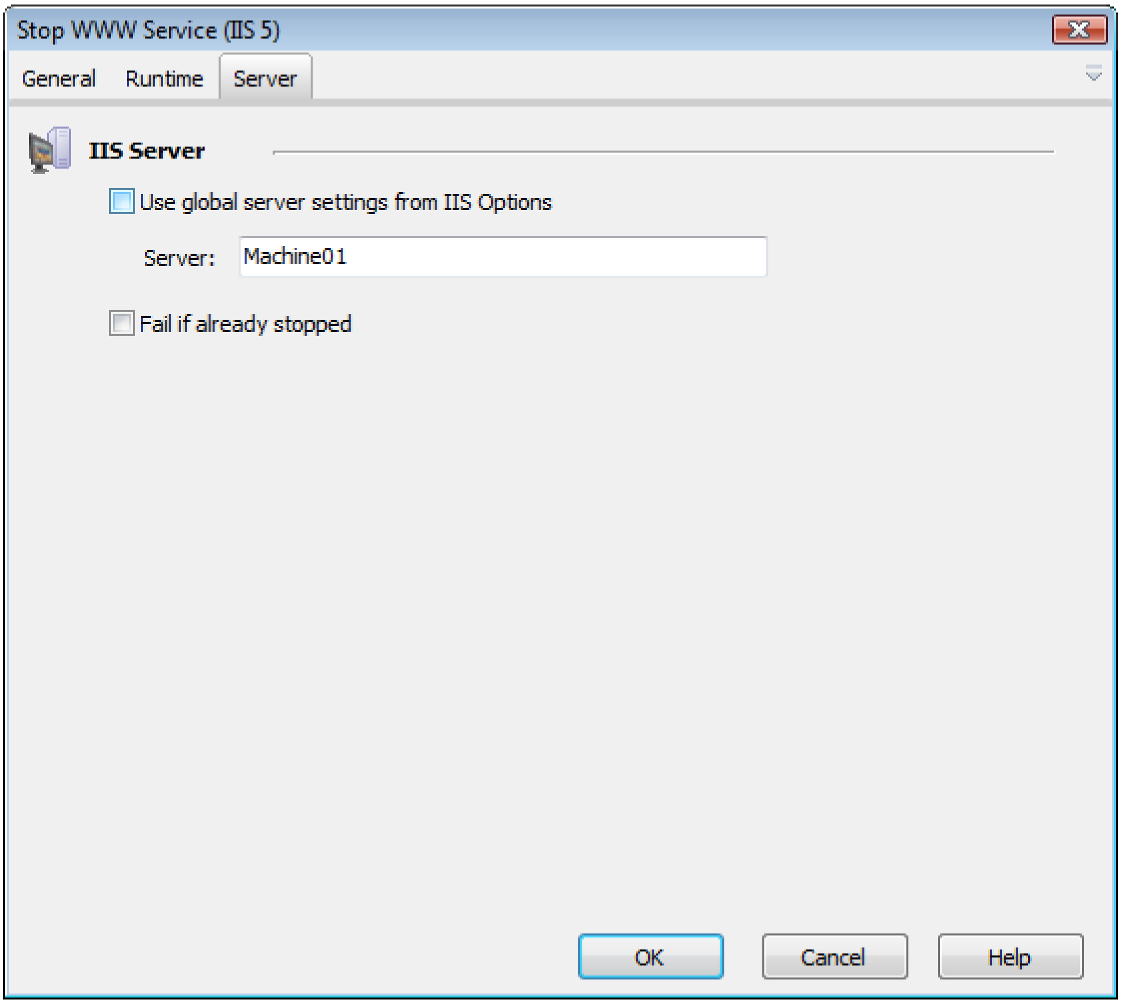
Start WWW Service (IIS 5)

IIS Server

□ Use global server settings from IIS Options

Server: Machine01

☑ Fail if already started



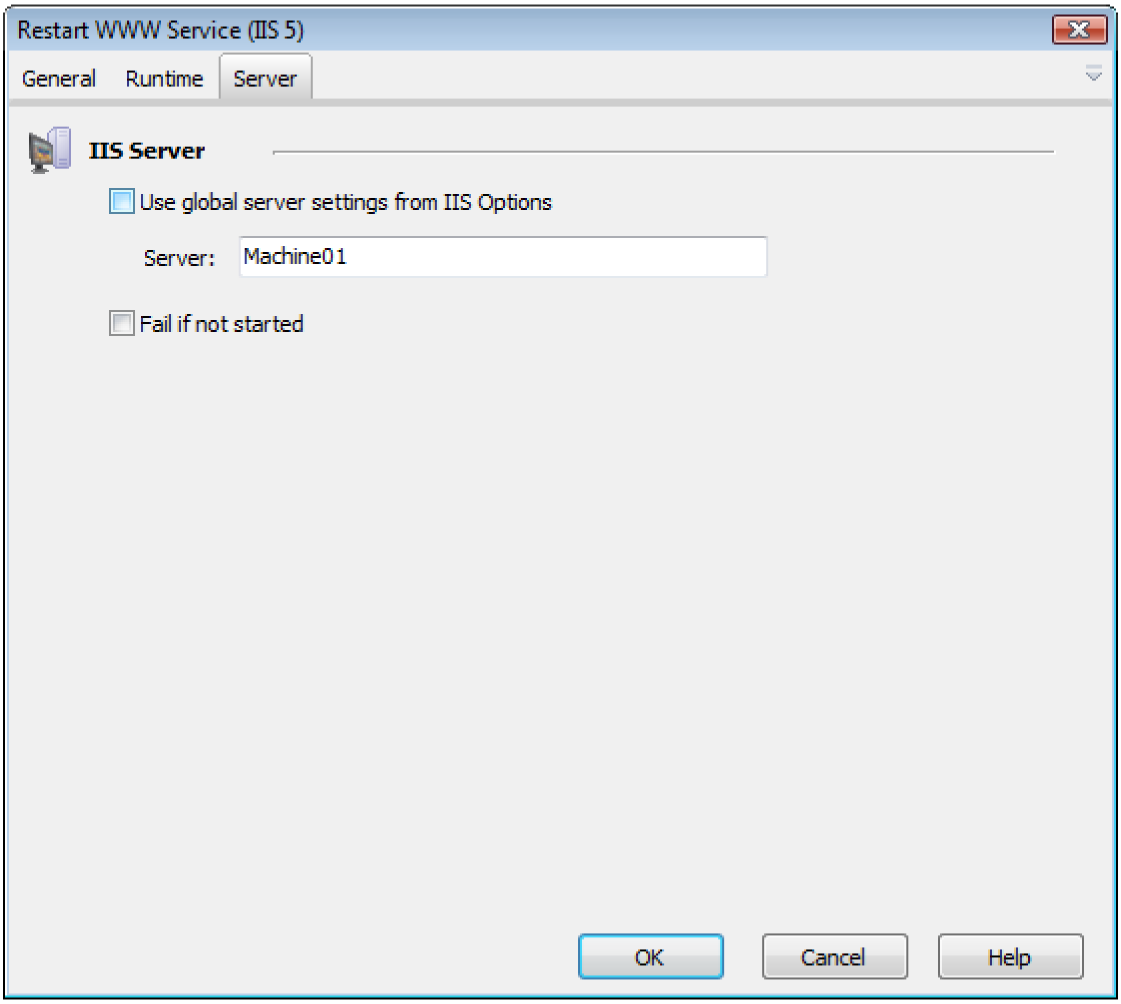
Stop WWW Service (IIS 5)

IIS Server

□ Use global server settings from IIS Options

Server: Machine01

□ Fail if already stopped



Restart WWW Service (IIS 5)

IIS Server

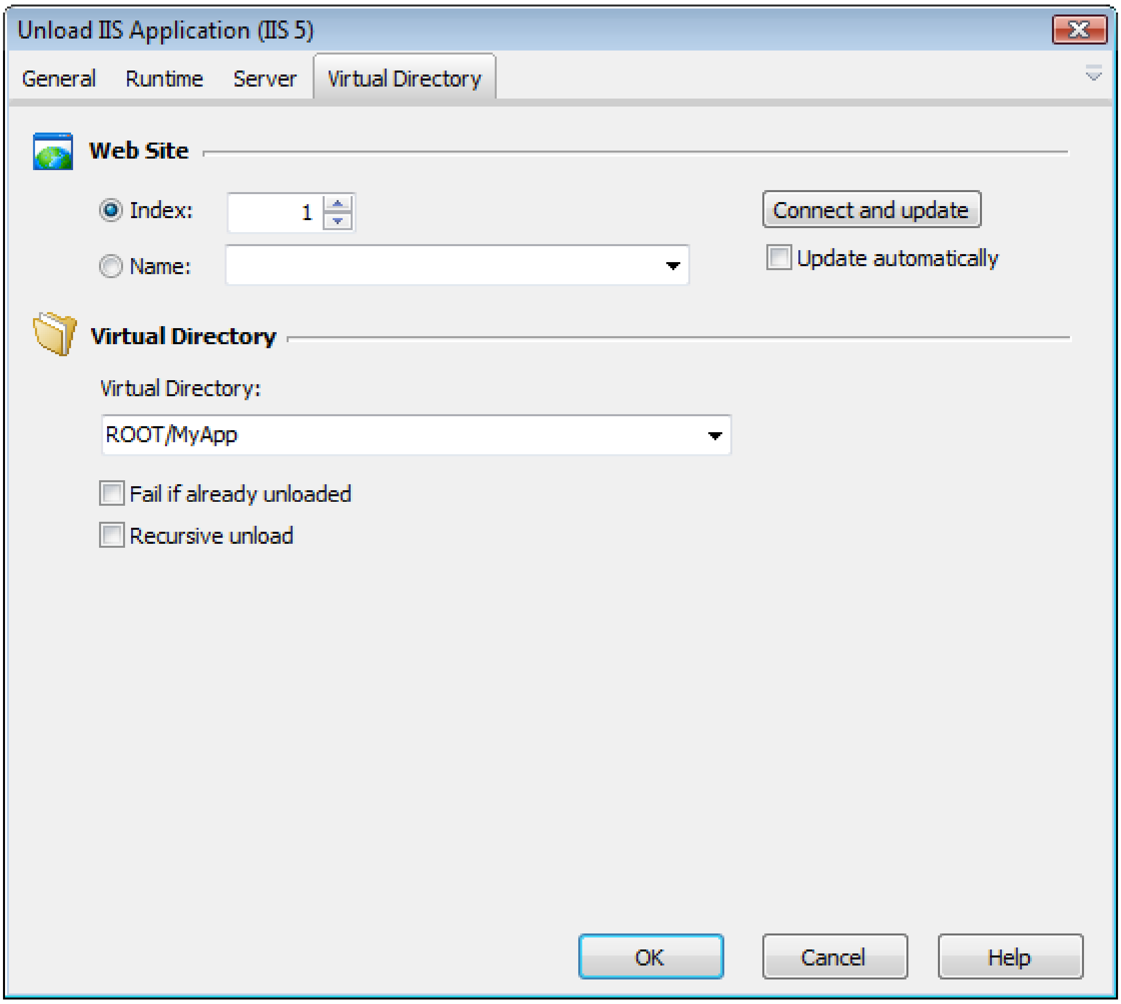
□ Use global server settings from IIS Options

Server: Machine01

□ Fail if not started

6.16.1.11 Unload IIS Application Action

The Unload IIS Application actions allows you unload a given IIS web application from memory.



● Index: 1

○ Name:

□ Update automatically

Virtual Directory: ROOT/MyApp

□ Fail if already unloaded

□ Recursive unload

Web Site

Select a web site, as per the Website Selection tab. Choose "update automatically" to have the virtual directory list automatically update as you change web sites.

Virtual Directory

Choose the virtual directory which hosts the application that you wish to unload. You can set the action not to fail if the application is already unloaded.

Select "Recursive Unload" to recursively unload applications in all child directories.

6.16.2 IIS 6

The IIS 6 actions use the WMI interface to administer features of IIS.

If you are using Windows 2000 or Windows XP (32-bit edition), then you may need to download the Windows Server 2003 Administration Tools pack from Microsoft (a free download) before you can administer an IIS 6 server.

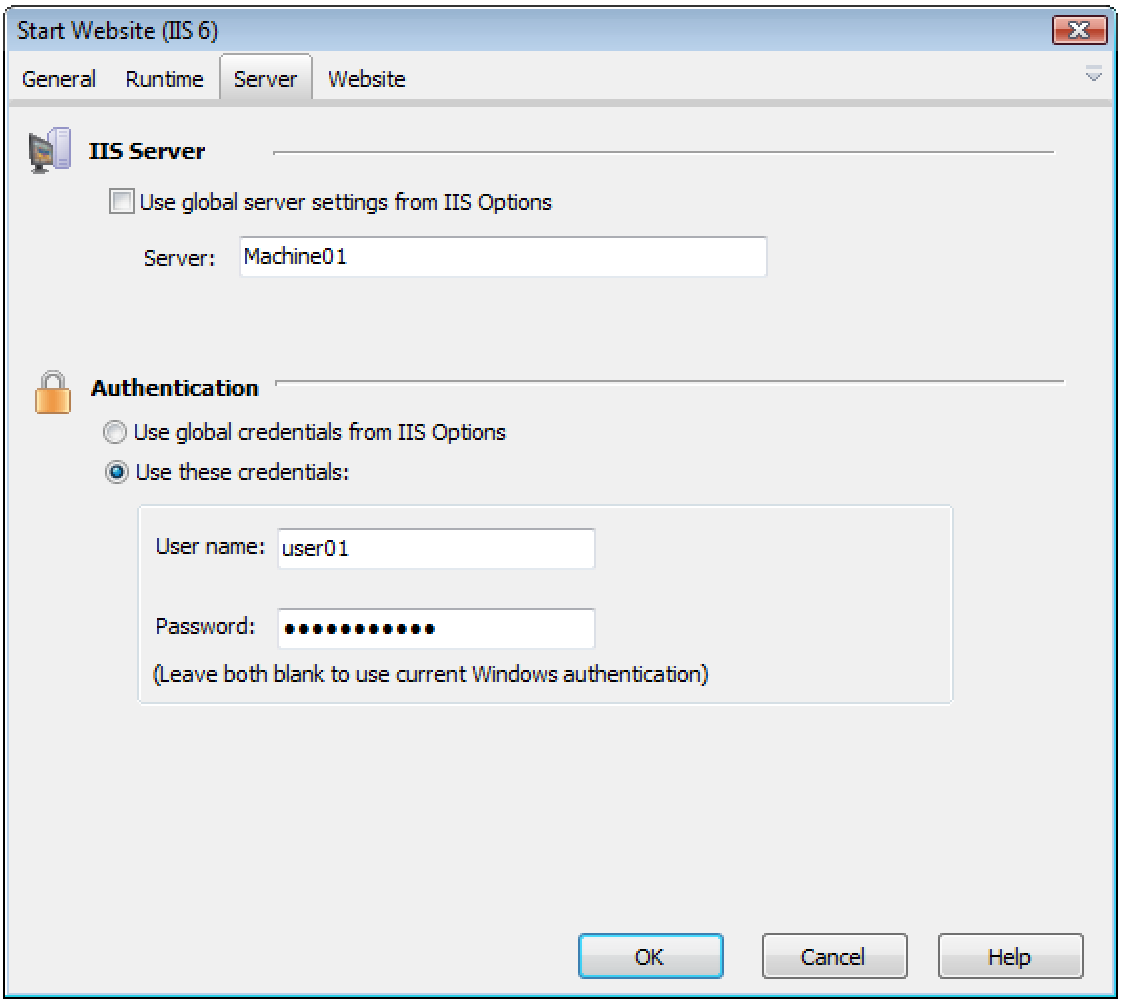
(The WMI interface is not available in IIS versions below 6.0.)

6.16.2.1 Server Selection

For each IIS action, you can specify the IIS Server as either a global server (specified in the options) or a particular server name. You can also specify credentials as either the credentials specified in the Options, or a particular set of credentials.

If the credentials are left blank, Windows authentication with the credentials of the currently logged-in user will be used.

The IIS 6 Actions use WMI to connect to the remote system, so you will need to ensure that WMI Remote Administration is available on the host computer. Here is an MSDN article about configuring Windows Firewall for WMI.



IIS Server

□ Use global server settings from IIS Options

Server: Machine01

Authentication

○ Use global credentials from IIS Options

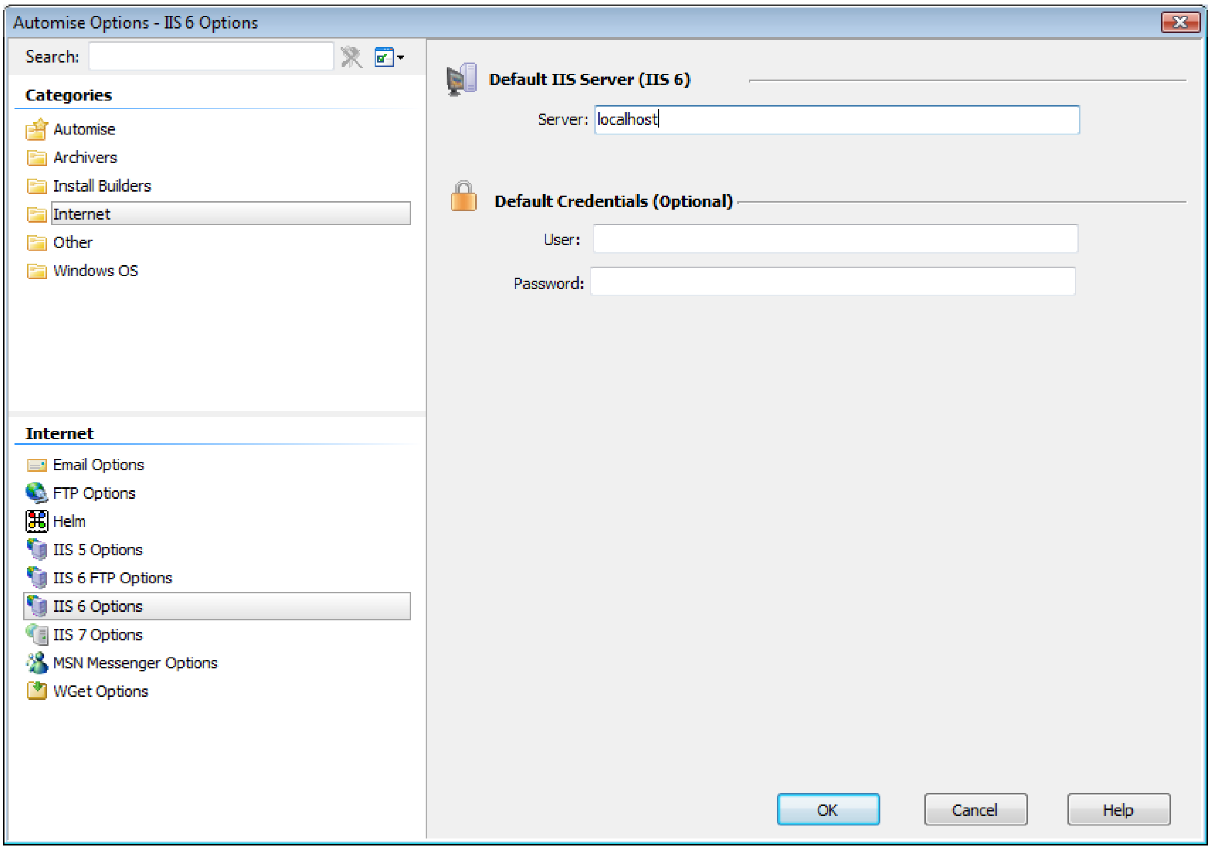
● Use these credentials:

User name: user01

Password: ●●●●●●●●●●●●

(Leave both blank to use current Windows authentication)

To select the global server name and credentials, choose Options from the Tools menu and then click on the IIS 6 Options tab under 'Internet':



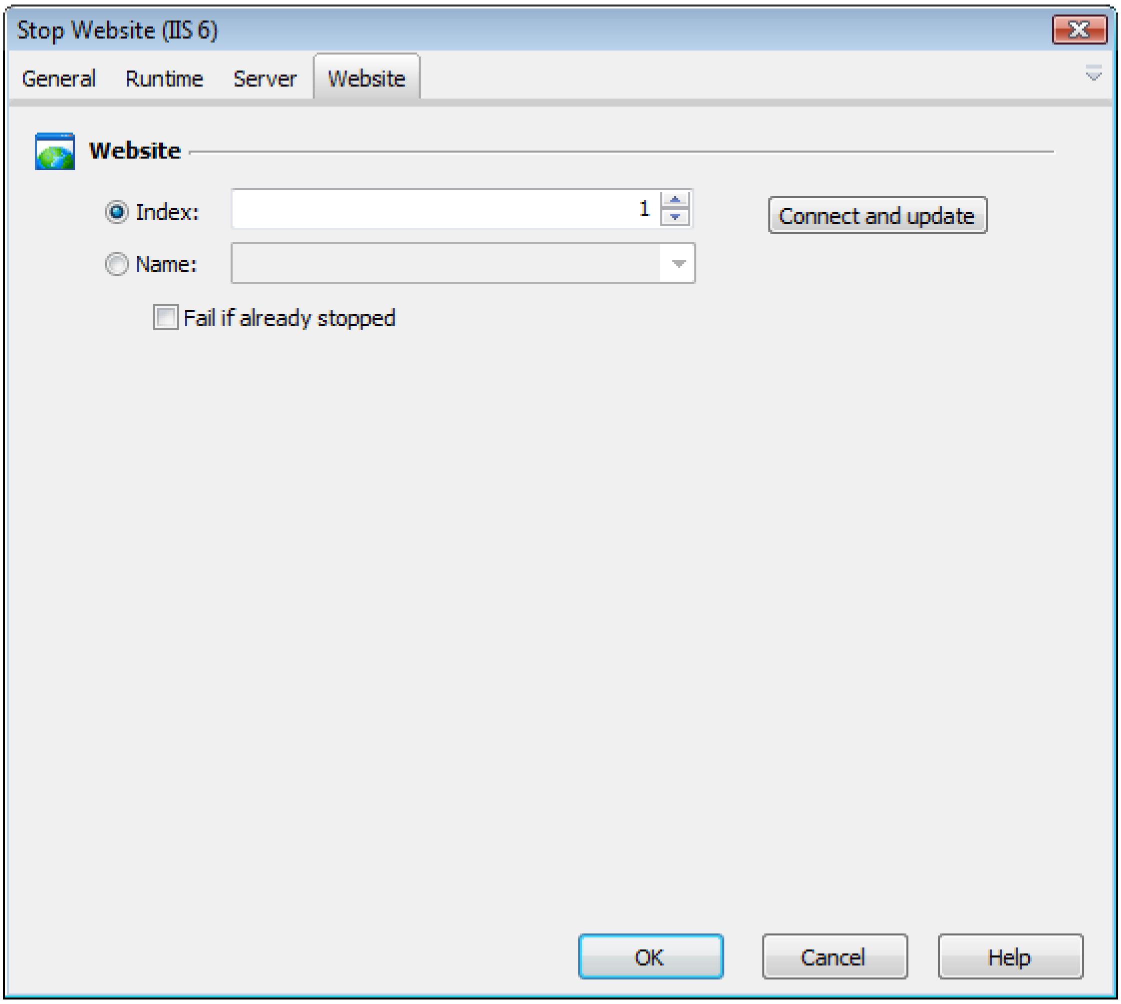
Server: localhost

User:

Password:

6.16.2.2 Website Selection

You can specify IIS websites by index number (also called the Identifier, can be seen under the Web Sites option in the IIS Manager application) or by name (also called the Description.)



Stop Website (IIS 6)

Website

● Index: 1

○ Name:

□ Fail if already stopped

To connect to the server and update the list of available web site names, click "Connect and update." Alternatively, just type a name into the combo box.

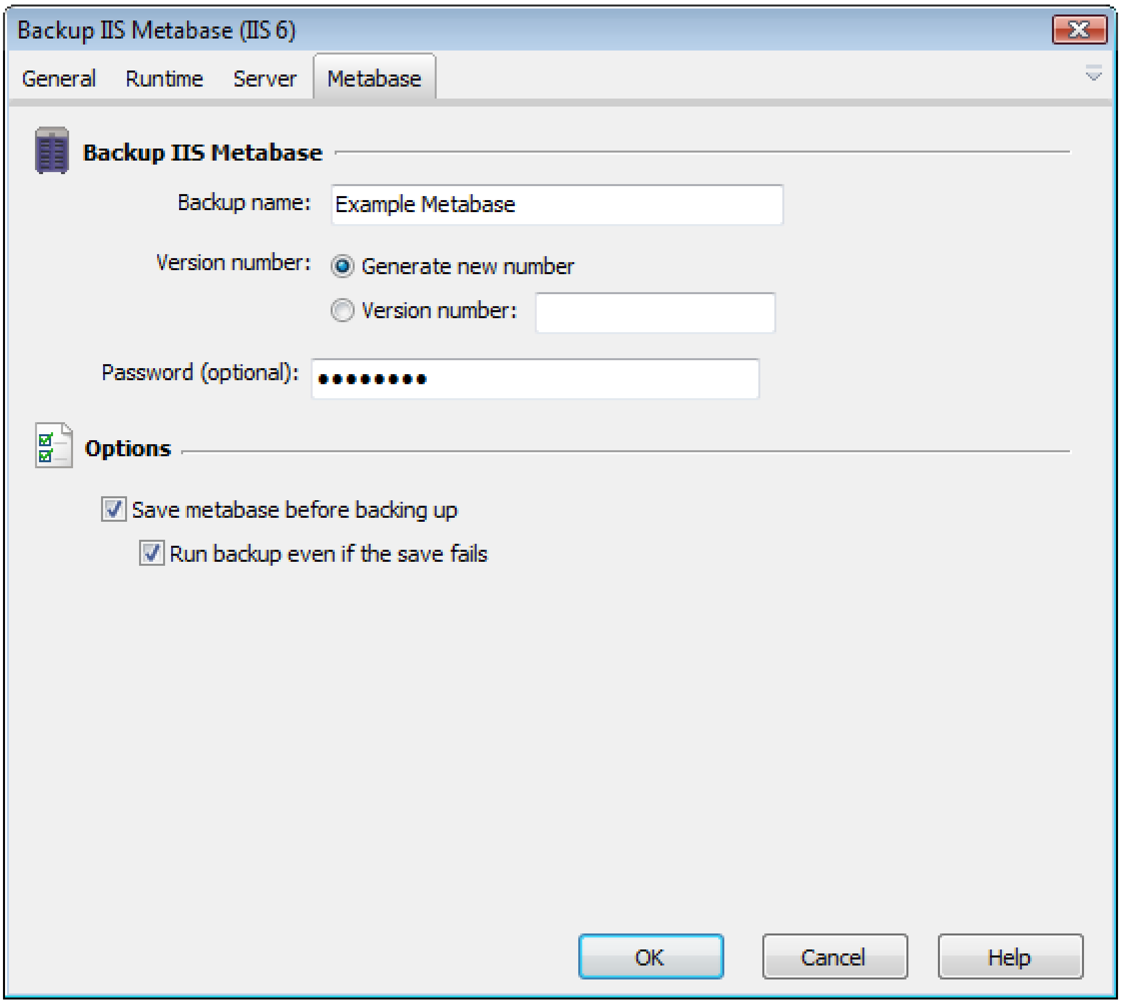
The "**Fail if already...**" checkbox allows you to specify whether some actions will abort if a given action (ie stopping or starting a web site) has already been taken.

6.16.2.3 Backup IIS Metabase Action

The Backup IIS Metabase action creates a backup of the entire IIS metabase.

You can access existing metabase backups from inside IIS Manager, by right-clicking on the Computer name and choosing
"All Tasks->Backup/Restore Configuration".

For help selecting Server and Credential properties (on the Server tab), see the Server Selection topic.



Backup IIS Metabase (IIS 6)

Backup IIS Metabase

Backup name: Example Metabase

Version number: ● Generate new number

○ Version number:

Password (optional): ●●●●●●●●●

Options

☑ Save metabase before backing up

☑ Run backup even if the save fails

Backup name

Specify the name you wish to give the backup.

Version number

Choose "Generate new number" to automatically generate an incremental backup number.

Choose "Version Number" to specify a version number. You can use a variable in this field, if you wish (for instance, to have the backup number correspond to the build number.)

If you try and create a backup with the same name & version number as a previous backup, an error will be raised.

Password (optional)

Optionally, IIS 6 metabase backups can be protected with a password. You will need the password in order to restore the backup.

Save metabase before backing up

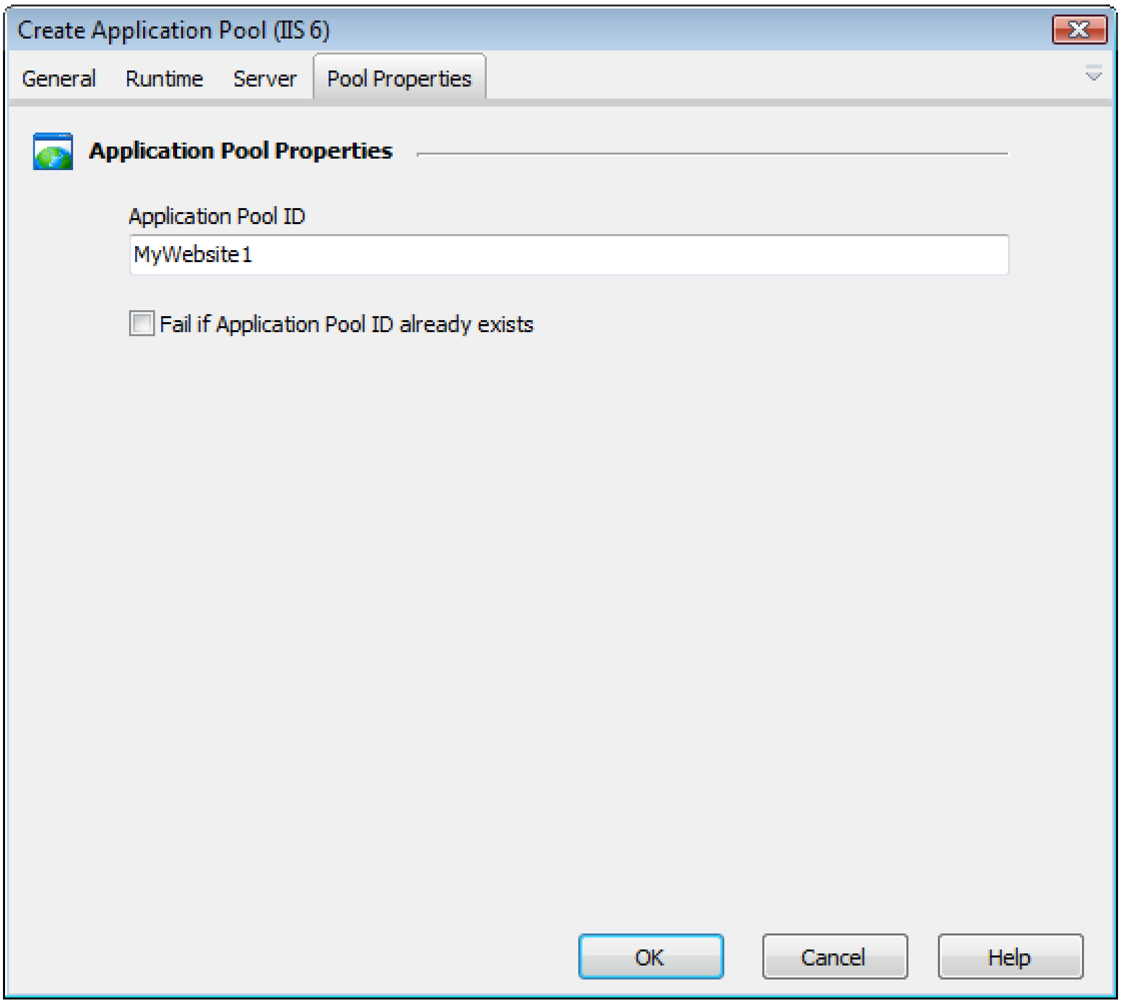
The contents of the metabase will be saved before the backup is undertaken.

Run backup even if the save fails

If this option is selected, the backup operation will be forced to continue even if the initial save fails. Ignored if not used in conjunction with "Save metabase before backing up."

6.16.2.4 Create Application Pool

The IIS Create Application Pool action allows you to create new Application Pools on an IIS Server.



Create Application Pool (IIS 6)

Application Pool Properties

Application Pool ID: MyWebsite1

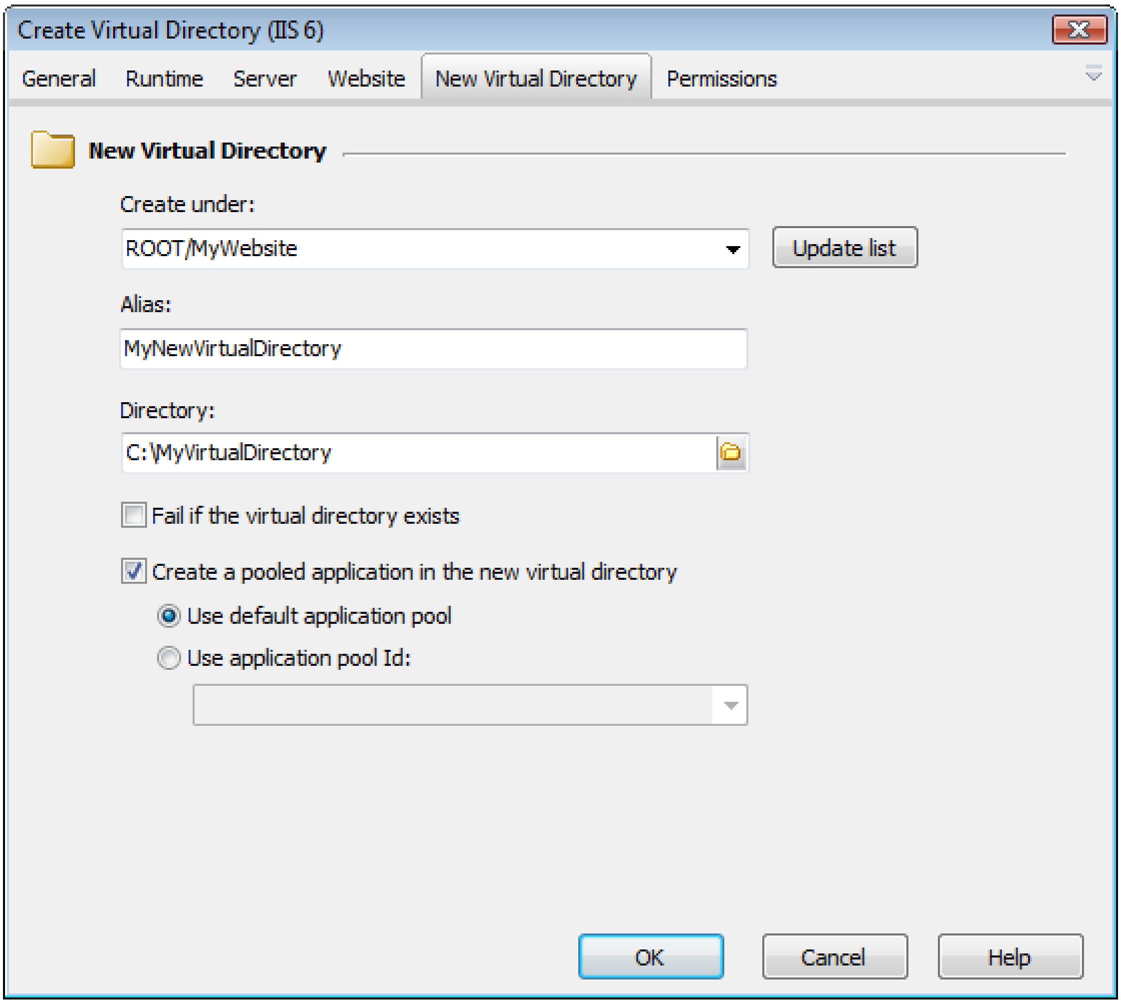
□ Fail if Application Pool ID already exists

Application Pool ID
The identifier to associate with the new Application Pool.

Fail if Application Pool ID already exists
Performs a check on the server to see whether an existing application pool has the same identifier.

6.16.2.5 Create Virtual Directory Action

The Create Virtual Directory action allows you to add a new virtual directory to an existing IIS web site.



Create under: ROOT/MyWebsite

Alias: MyNewVirtualDirectory

Directory: C:\MyVirtualDirectory

□ Fail if the virtual directory exists

☑ Create a pooled application in the new virtual directory

● Use default application pool

○ Use application pool Id:

Server & Website

Specify the Server and Website to use via the Server Selection & Website Selection tabs, respectively.

Create under

Specify an existing IIS Virtual Directory to be the parent of the new directory. Click the 'Update list' button to download a list
of Virtual Directories from the specified IIS Server.

Alias

Specify the name by which you would like the new directory to be known.

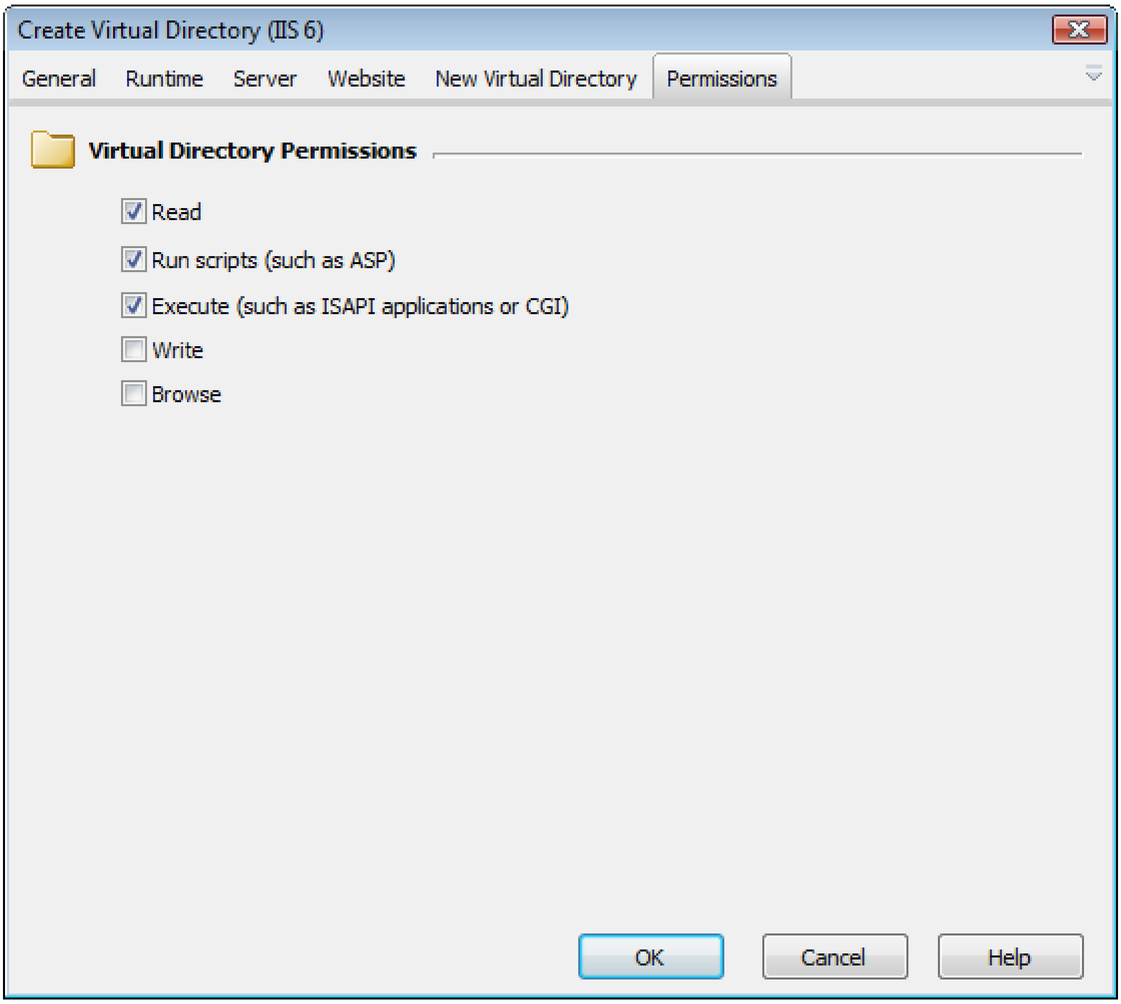
Directory

Specify the local directory for the contents of the new virtual directory.

Create a pooled application...

Check this box if you would like to create a pooled web application running in the new virtual directory.

Permissions Tab



Create Virtual Directory (IIS 6)

General | Runtime | Server | Website | New Virtual Directory | Permissions

Virtual Directory Permissions

☑ Read

☑ Run scripts (such as ASP)

☑ Execute (such as ISAPI applications or CGI)

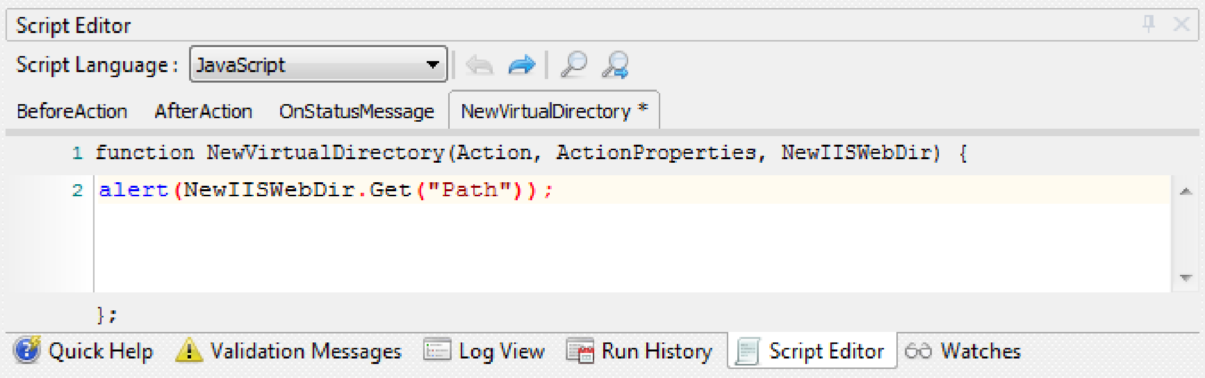
□ Write

□ Browse

Set the permissions you want IIS users to have on the new Virtual Directory.

Script Events

The Create Virtual Directory object has an extra script event, named NewVirtualDirectory, which is called once the new virtual directory has set up. It allows you to set properties which may not be available in the action, or perform more complicated operations on the new directory.



```
function NewVirtualDirectory(Action, ActionProperties, NewIISWebDir) {
alert(NewIISWebDir.Get("Path"));

};
```

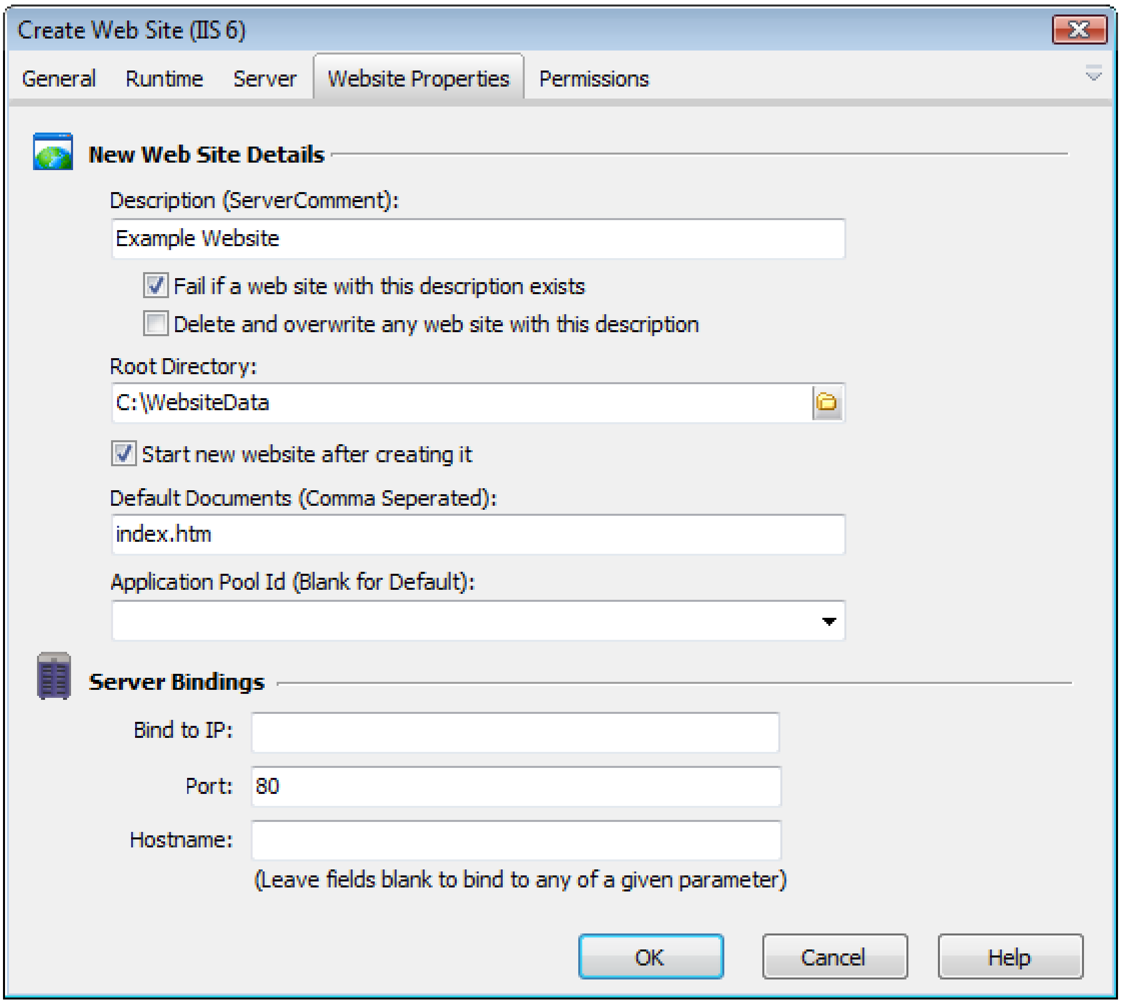
The **NewIISWebDir parameter** is an WMI settings object of type IIsWebVirtualDirSetting (see the MSDN Documentation.)

You can retrieve and set the virtual directory's properties directly using the syntax shown in the above example. See the MSDN Documentation for a full list of available properties. Unfortunately, code completion is not available on this object.

There is no need to call the .Put_() method to save any changes which are made (this is done automatically when the event completes.)

6.16.2.6 Create Web Site Action

Use the IIS Create Web Site action to create a new web site. The Create Web Site action also creates a root virtual directory.



New Web Site Details

Description (ServerComment): Example Website

☑ Fail if a web site with this description exists

□ Delete and overwrite any web site with this description

Root Directory: C:\WebsiteData

☑ Start new website after creating it

Default Documents (Comma Seperated): index.htm

Application Pool Id (Blank for Default):

Server Bindings

Bind to IP:

Port: 80

Hostname:

(Leave fields blank to bind to any of a given parameter)

Select the server (and connection credentials) under the Server tab (see the Server Selection topic for details.)

New Web Site Details

Description - Description specifies a ServerComment property for the web site (in other words its name.) Web sites are identified either by this property or their index number.

Fail if a web site... - If this option is checked, the action will fail if another web site already has this description.

Delete and overwrite... - If this option is checked, a web site with this name will be overwritten and replaced with the new site. The new web site will have the same index number as the old site.

If neither *Fail* nor *Delete* is selected, a new web site will be created even if it has the same description as a pre-existing one.

Root Directory - Specify a directory local to the server on which to create the root virtual directory. Set the permissions for this directory on the Permissions tab.

Start new web site after creating it - If this option is checked, the new web site will be switched to the Running state after creation.

Note that even if this fails (most likely because another running web site is bound to the same host/server/port), the new web site is still created (in the Stopped state), but the action will fail.

For fine-grained control over failure (ie for use in Try...Catch blocks), use a separate Start Web Site action.

Server Bindings

You can specify a specific IP, hostname, or Port combination to bind the new server to. Leave values blank to match all-inclusive wild cards.

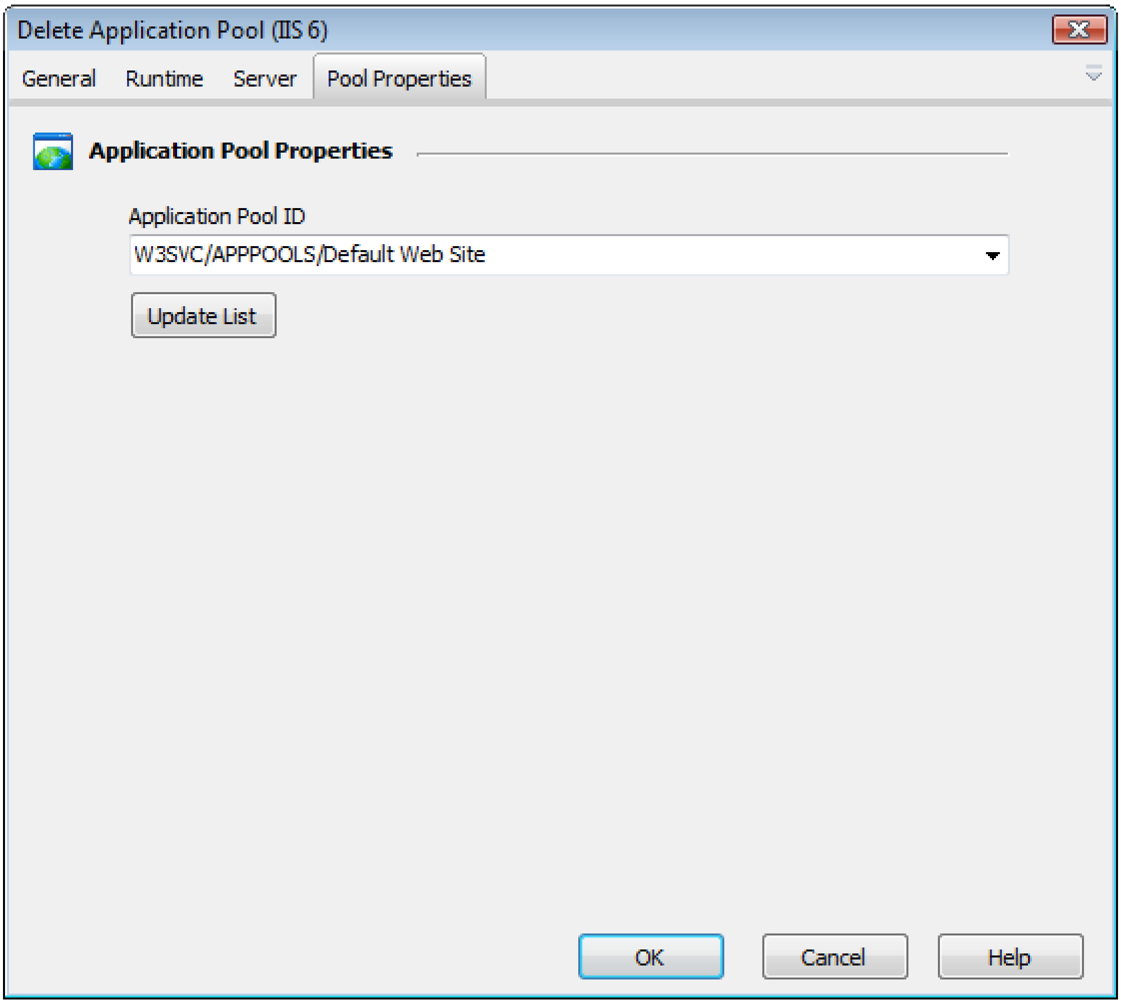
Note that if any of these values overlap with an existing web site, the two cannot be run concurrently.

Permissions Tab

Set the permissions for the new web site's root Virtual Directory.

6.16.2.7 Delete Application Pool

The IIS Delete Application Pool action enables you to remove existing application pools from an IIS server.

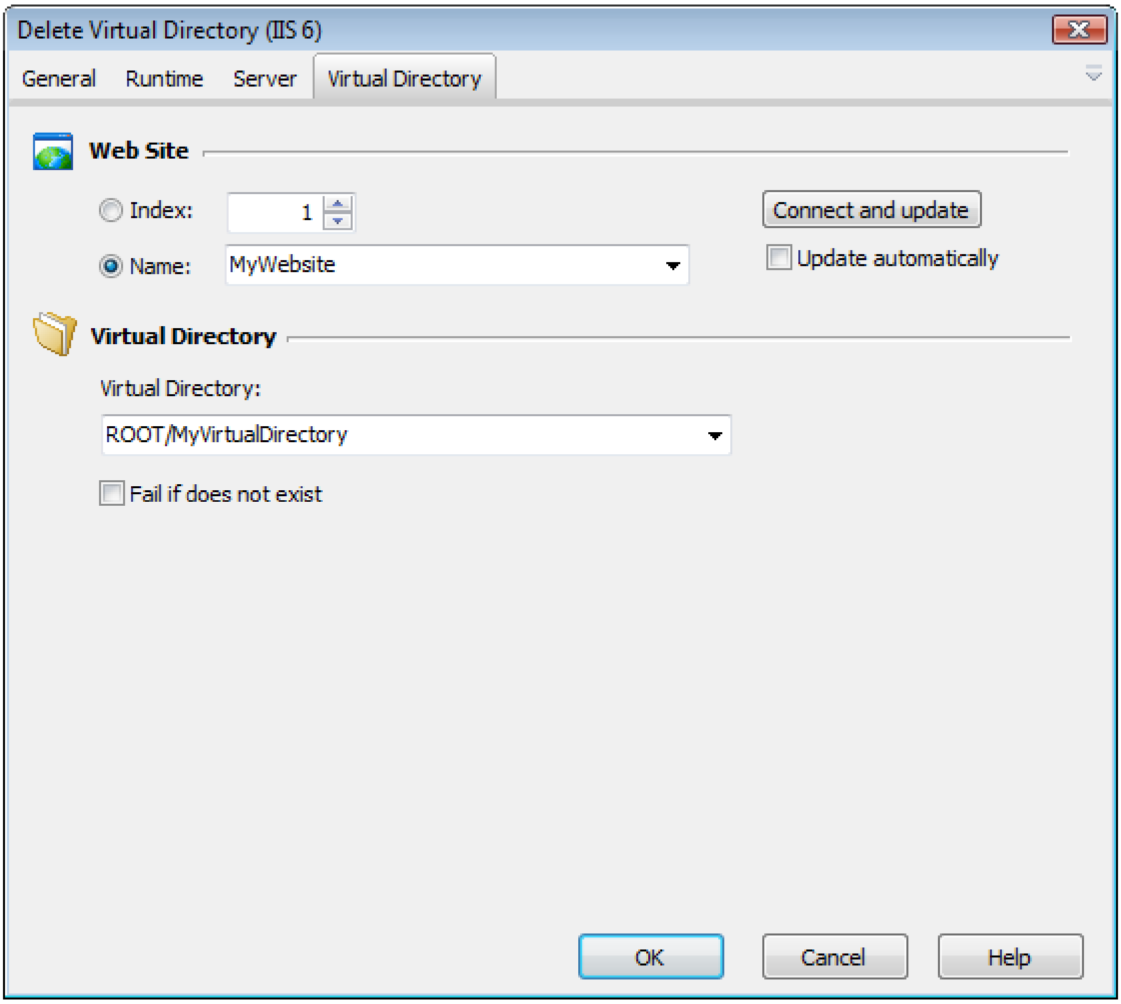

Delete Application Pool (IIS 6)

Application Pool Properties

Application Pool ID: W3SVC/APPPOOLS/Default Web Site

Application Pool ID
The identifier of the Application Pool to remove. The 'Update List' button will connect with the IIS Server and retrieve all existing Application Pool IDs, adding them to the drop down list.

6.16.2.8 Delete Virtual Directory Action



Delete Virtual Directory (IIS 6)

Web Site

○ Index: 1

● Name: MyWebsite

□ Update automatically

Virtual Directory

Virtual Directory: ROOT/MyVirtualDirectory

□ Fail if does not exist

Web Site

Choose a web site name or index (for details see the Website Selection tab.) Check the "Update automatically" box to have changes in the web site selection automatically reload the virtual directory listing.

Virtual Directory

Choose the virtual directory to delete. Click "Connect and update" to have a list of existing virtual directories loaded into the combo box.

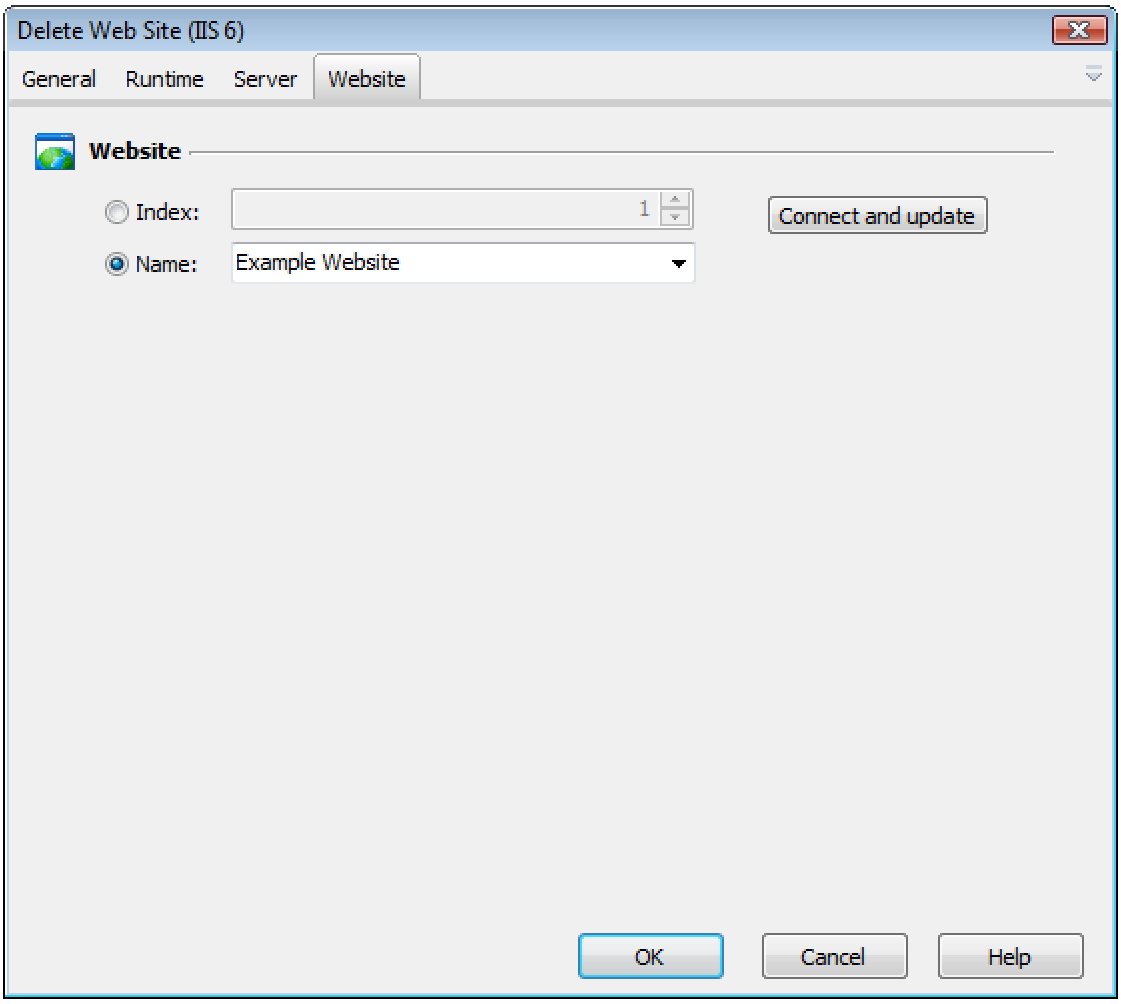
Fail if does not exist

Check this box if you want the action to fail should the specified virtual directory not exist.

6.16.2.9 Delete Web Site Action

The Delete Web Site action allows you remove a web site from the IIS metabase.

Use the Server Selection tab to choose the IIS server to delete the web site from & credentials for that server. Then use the Website Selection tab to choose the web site to delete.



○ Index: 1

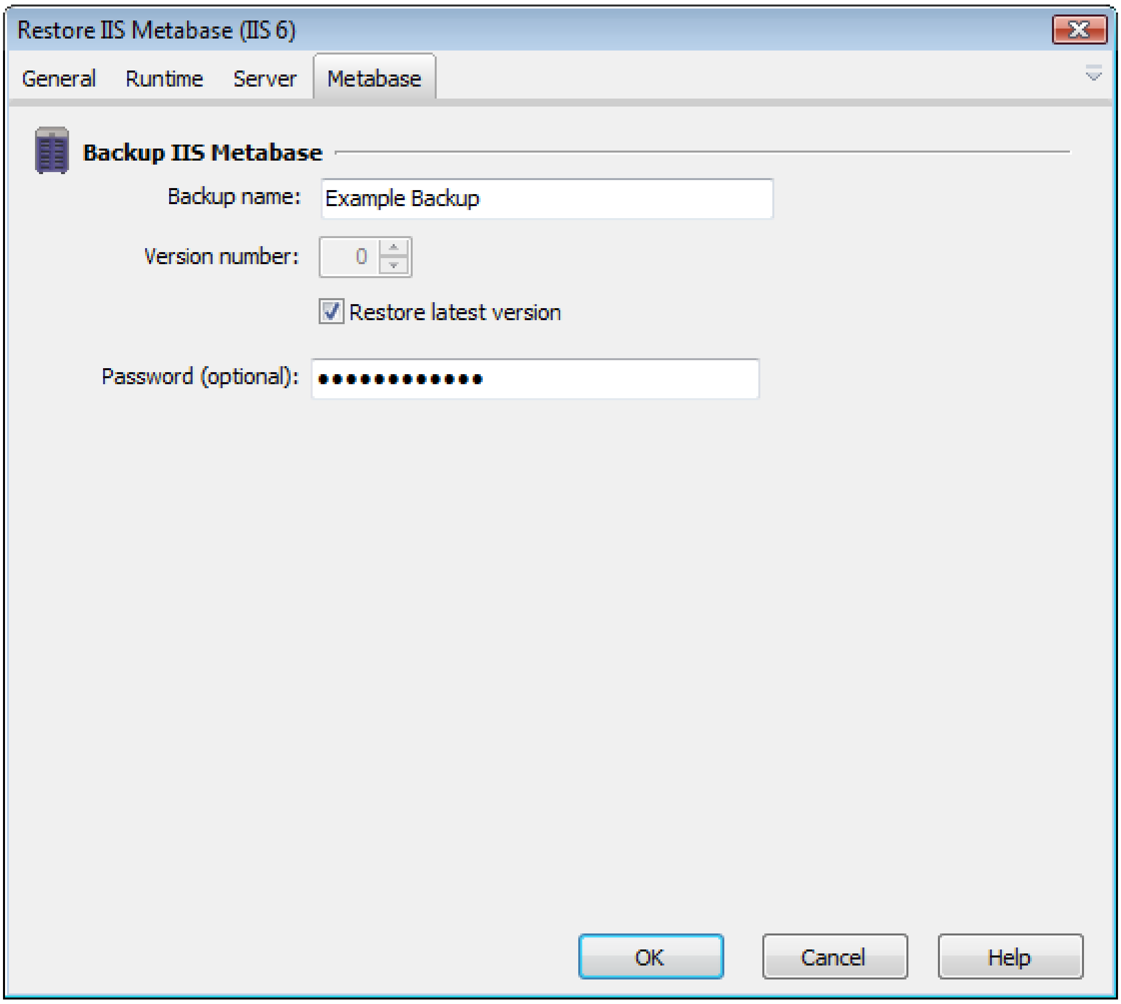
● Name: Example Website

6.16.2.10 Restore IIS Metabase Action

The Restore IIS Metabase action allows you restore a previously backed up IIS Metabase configuration.

You can access existing metabase backups from inside IIS Manager, by right-clicking on the Computer name and choosing
"All Tasks->Backup/Restore Configuration".

For help selecting Server and Credential properties (on the Server tab), see the Server Selection topic.



Backup IIS Metabase

Backup name: Example Backup

Version number: 0

☑ Restore latest version

Password (optional): ●●●●●●●●●●●●

Backup name

Specify the name of the backup you wish to restore

Version Number

Specify a specific version number to restore, or choose "Restore latest version" to automatically find the latest backup and restore it.

Password (optional)

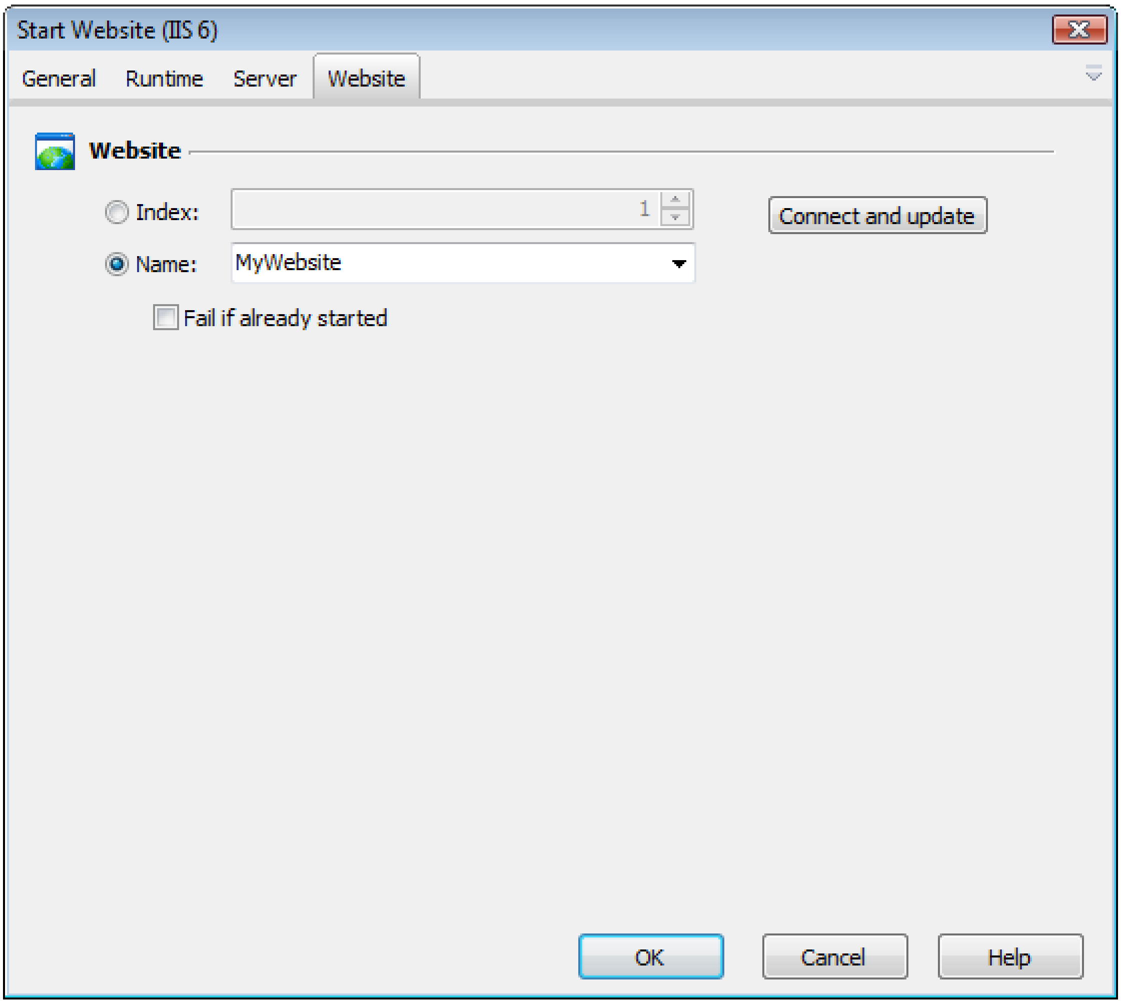
If the backup was created with a password, you will need to enter the password here in order to restore it.

6.16.2.11 Start / Stop / Pause Website Actions

The Start, Stop & Pause Website actions allow you to take individual IIS websites on- and off-line.

Use the Server Selection tab to choose the server you wish to administer, and credentials for the connection.
Then use the Website Selection tab to choose the web site you wish to modify.

The actions can be set not to fail if the selected web site is already in the desired state (running/stopped.)



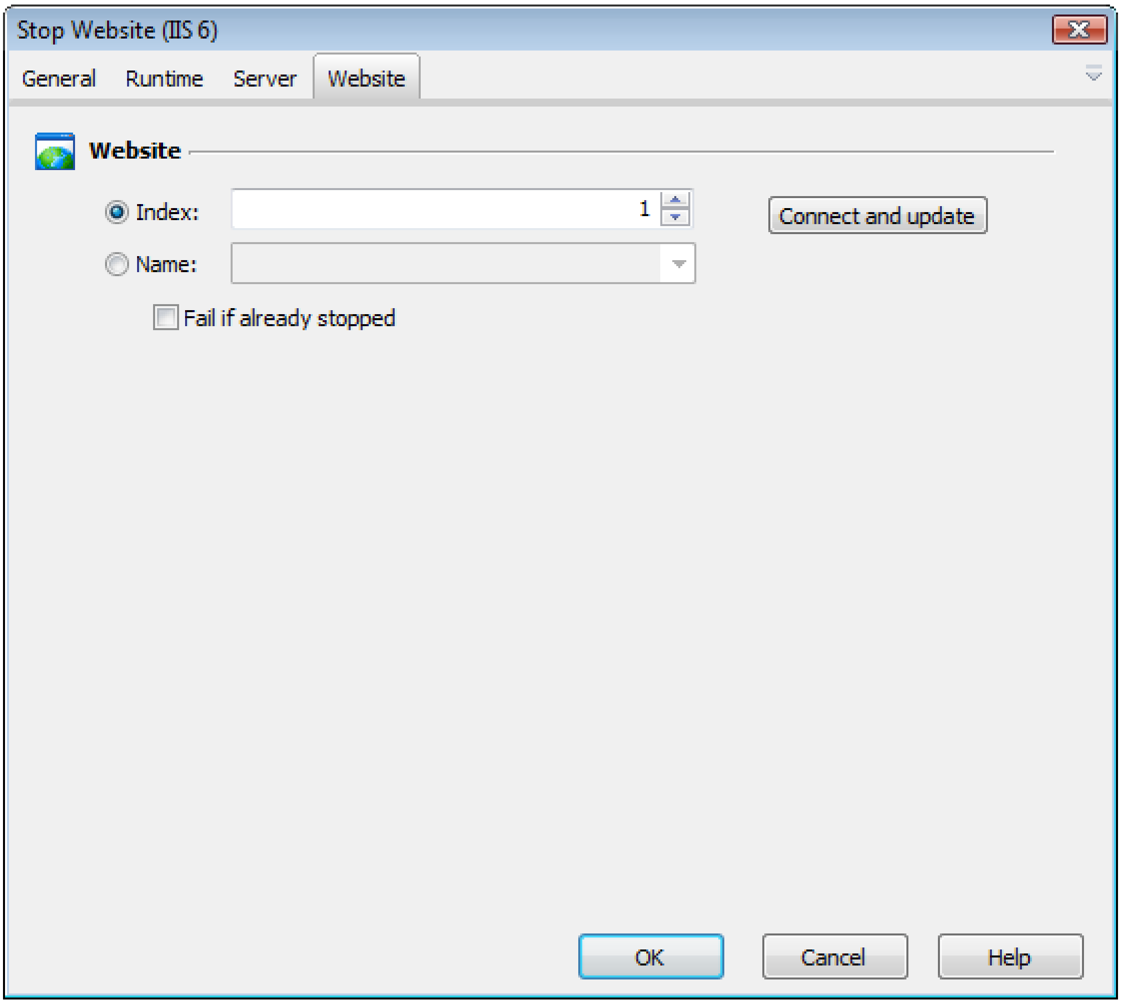
Start Website (IIS 6)

Website

○ Index: 1

● Name: MyWebsite

☐ Fail if already started



Stop Website (IIS 6)

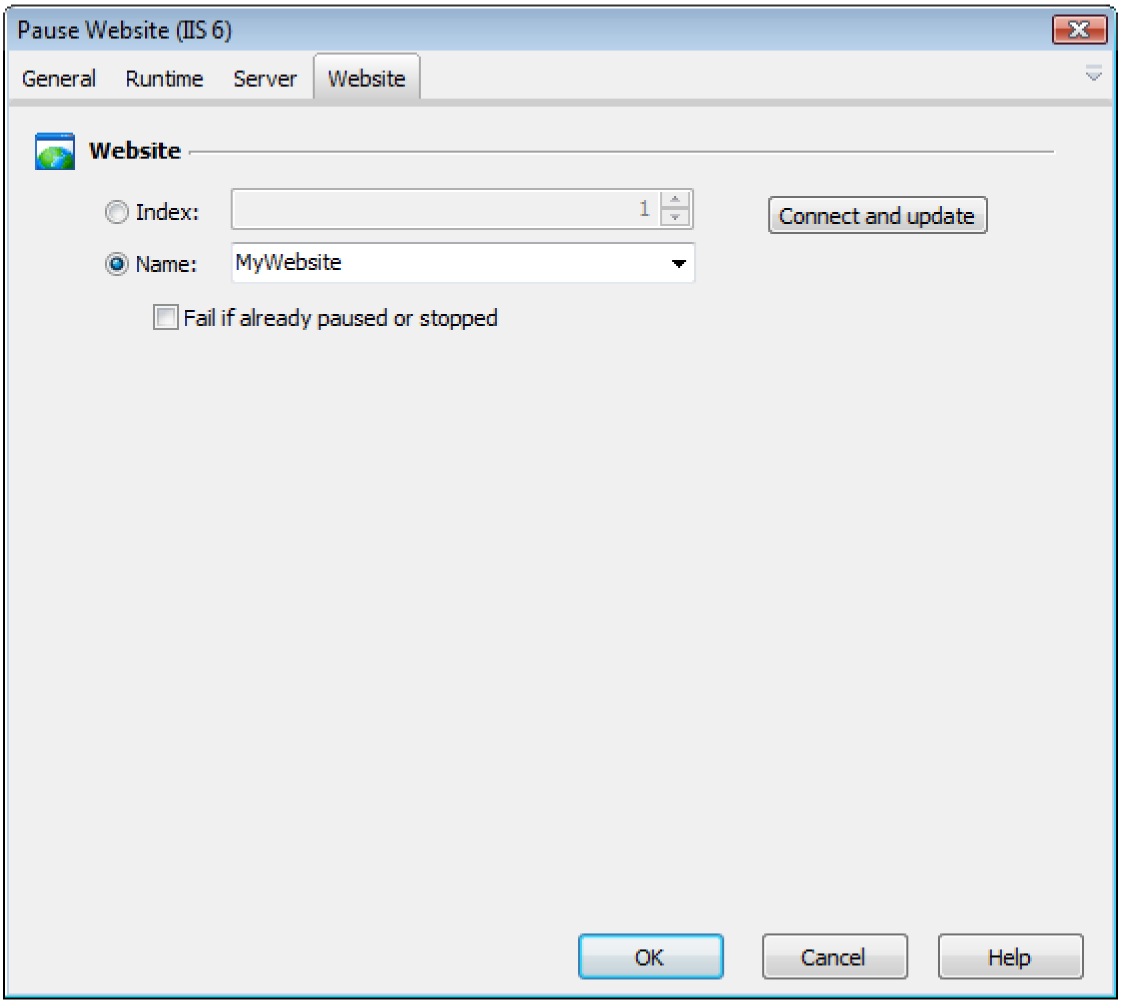
General | Runtime | Server | Website

Website

● Index: 1

○ Name:

□ Fail if already stopped



Pause Website (IIS 6)

○ Index: 1

● Name: MyWebsite

□ Fail if already paused or stopped

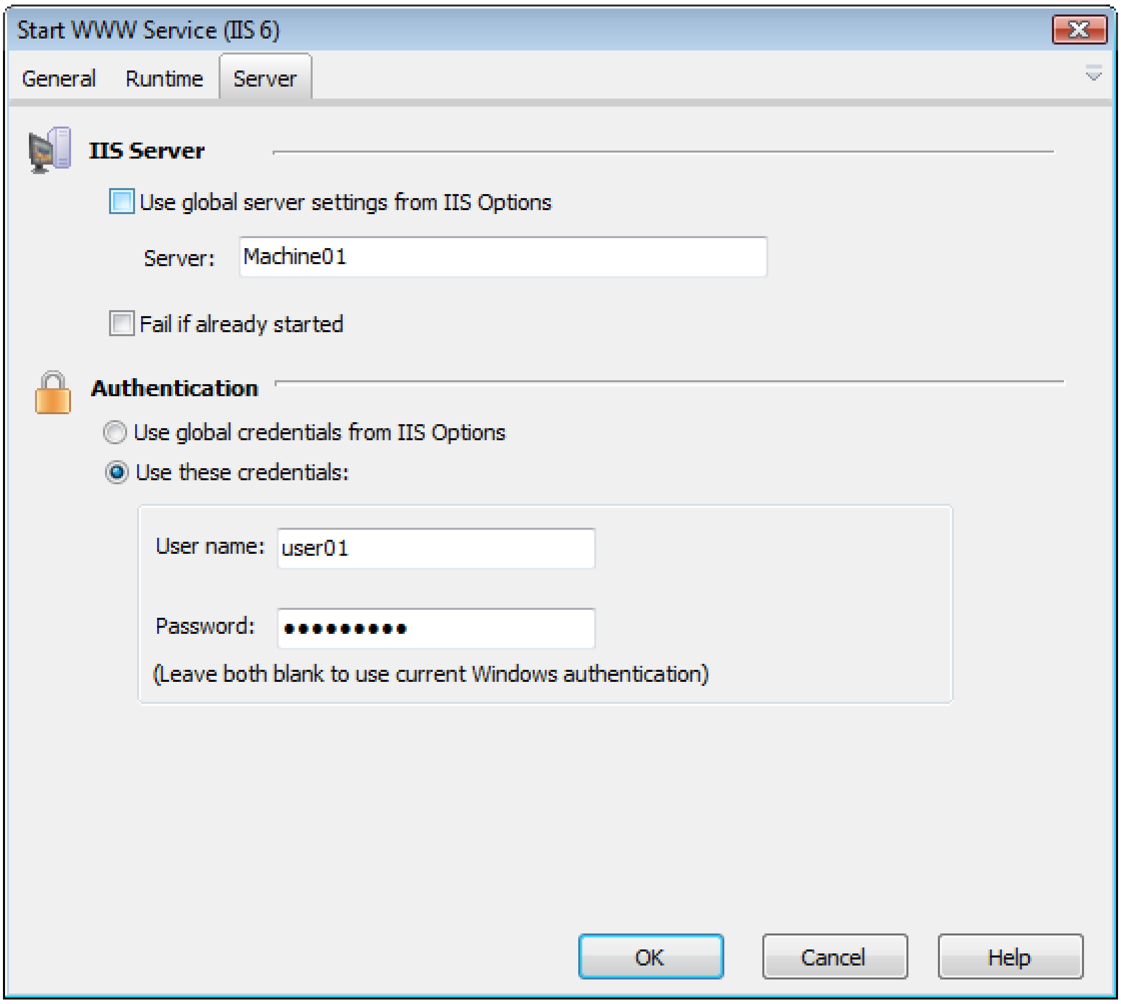
6.16.2.12 Start / Stop / Restart WWW Service Action

The Start, Stop & Restart WWW Service actions use WMI to start and stop the IIS WWW service (W3SVC) on a server running IIS.

Use the Server Selection tab to choose the server you wish to administer, and to specify credentials for connecting to the server.

Each of the actions can be set not to fail if the service is already started/stopped (in the case of the Restart action, the service will be started even if it is initially stopped.)

It is recommended that you use the Restart action instead of a Stop followed immediately by a Start action (the Restart action waits to ensure the service is fully stopped before restarting it.)



Start WWW Service (IIS 6)

IIS Server

☐ Use global server settings from IIS Options

Server: Machine01

☐ Fail if already started

Authentication

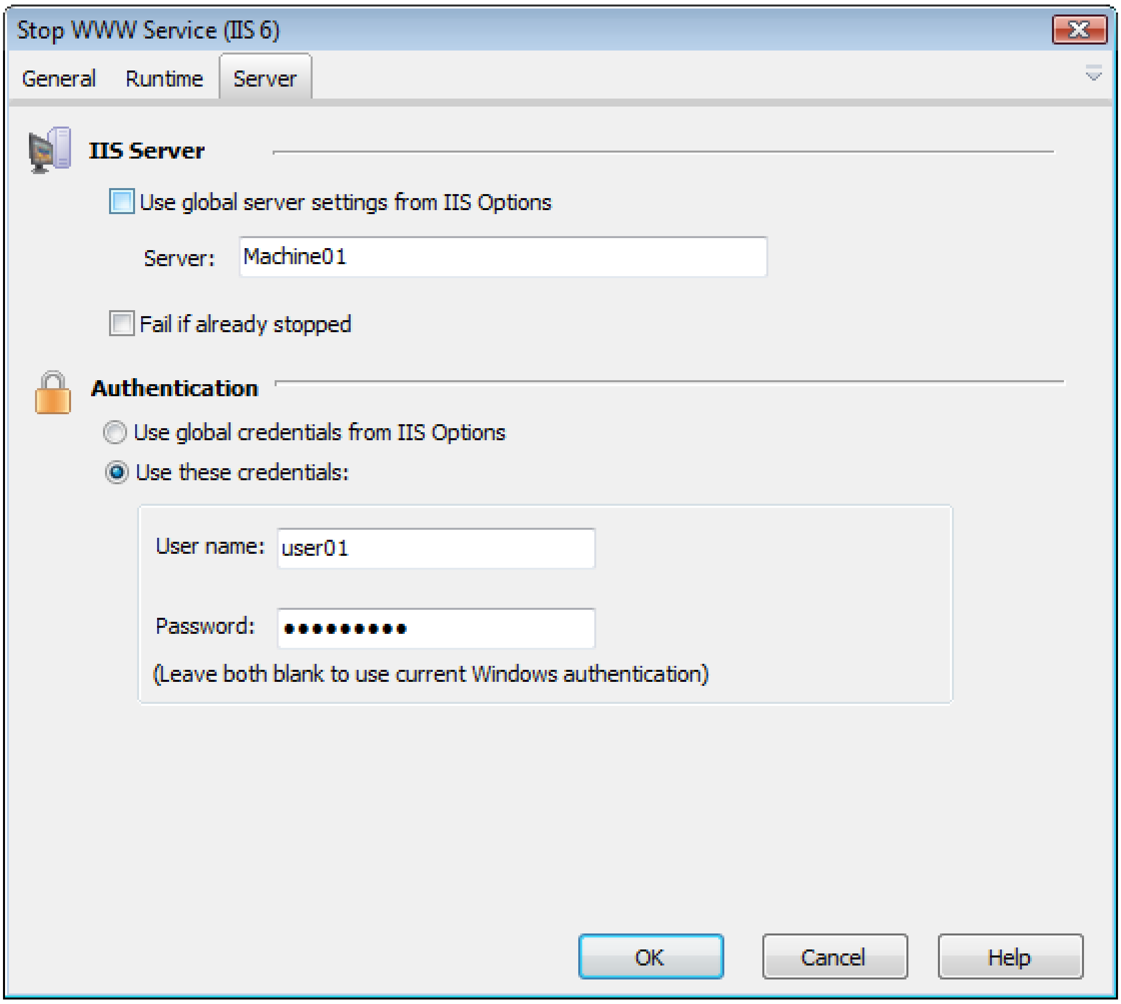
○ Use global credentials from IIS Options

● Use these credentials:

User name: user01

Password: ●●●●●●●●●●

(Leave both blank to use current Windows authentication)



IIS Server

☐ Use global server settings from IIS Options

Server: Machine01

☐ Fail if already stopped

Authentication

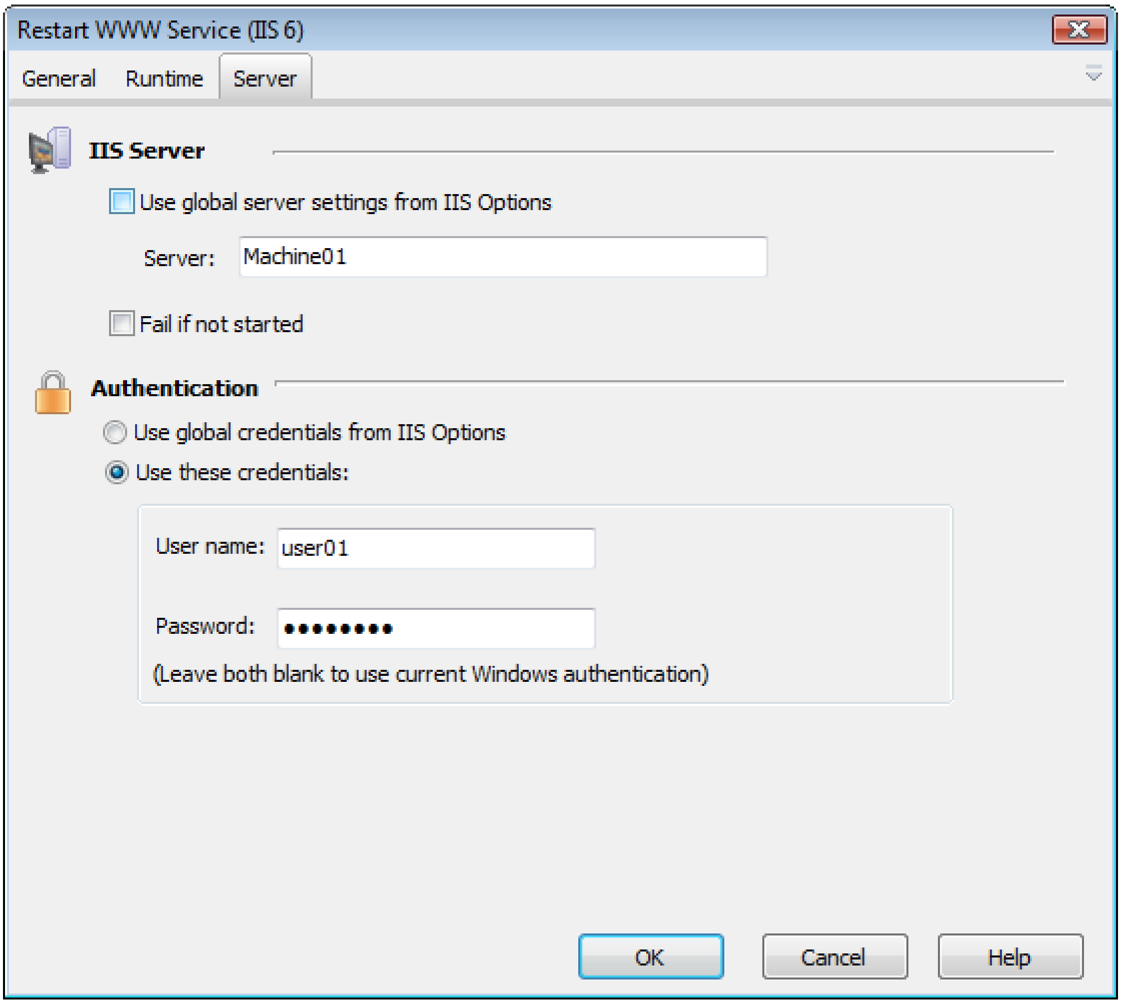
○ Use global credentials from IIS Options

● Use these credentials:

User name: user01

Password: ••••••••••

(Leave both blank to use current Windows authentication)



□ Use global server settings from IIS Options

Server: Machine01

□ Fail if not started

○ Use global credentials from IIS Options

● Use these credentials:

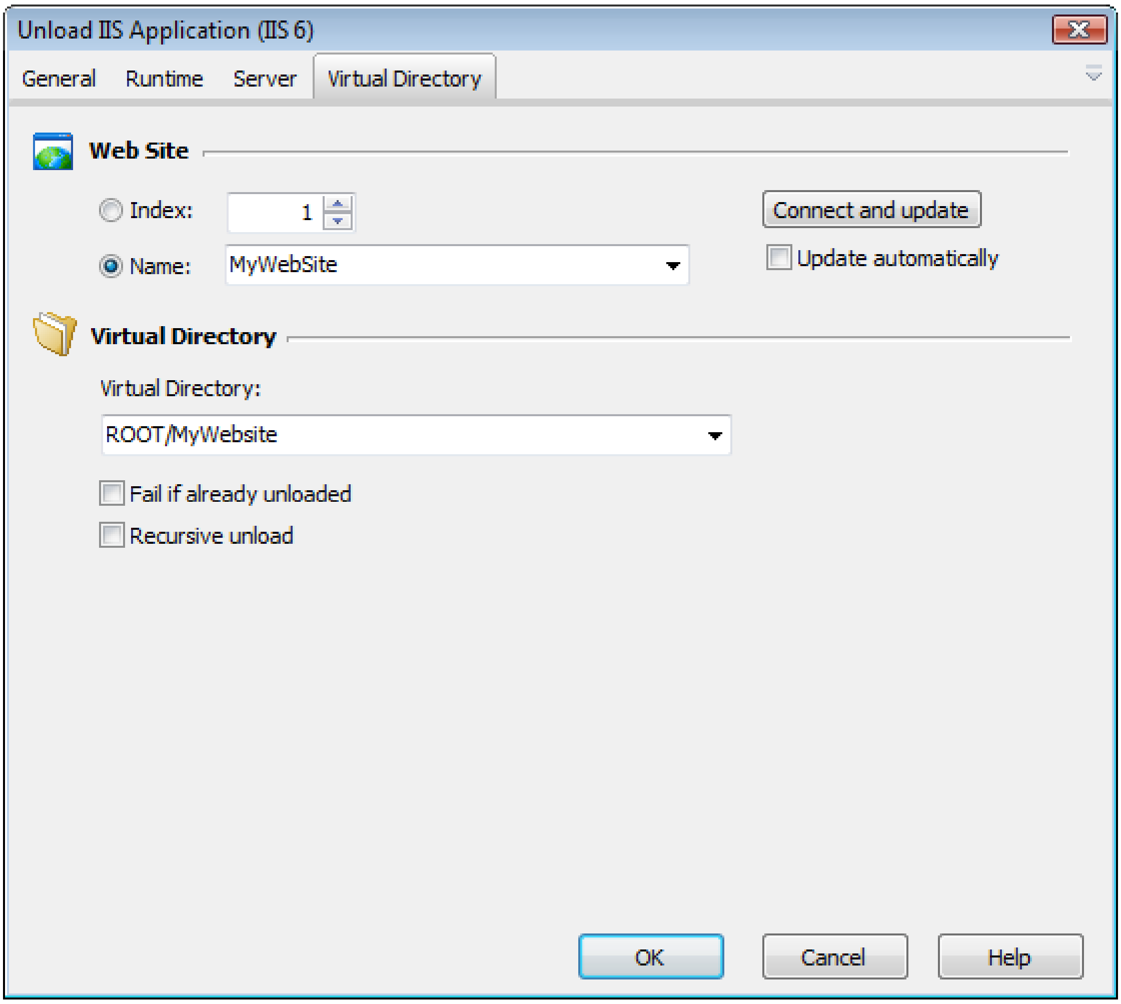
User name: user01

Password: ●●●●●●●●●

(Leave both blank to use current Windows authentication)

6.16.2.13 Unload IIS Application Action

The Unload IIS Application actions allows you unload a given IIS web application from memory.



Unload IIS Application (IIS 6)

Web Site

○ Index: 1

● Name: MyWebSite

□ Update automatically

Virtual Directory

Virtual Directory: ROOT/MyWebsite

□ Fail if already unloaded

□ Recursive unload

Web Site

Select a web site (for details see the Website Selection tab.) Choose "update automatically" to have the virtual directory list automatically update as you change web sites.

Virtual Directory

Choose the virtual directory which hosts the application that you wish to unload. You can set the action not to fail if the application is already unloaded.

Select "Recursive Unload" to recursively unload applications in all child directories.

6.16.3 IIS 6 FTP

The IIS 6 FTP actions use the WMI interface to administer the FTP server features of IIS 6.

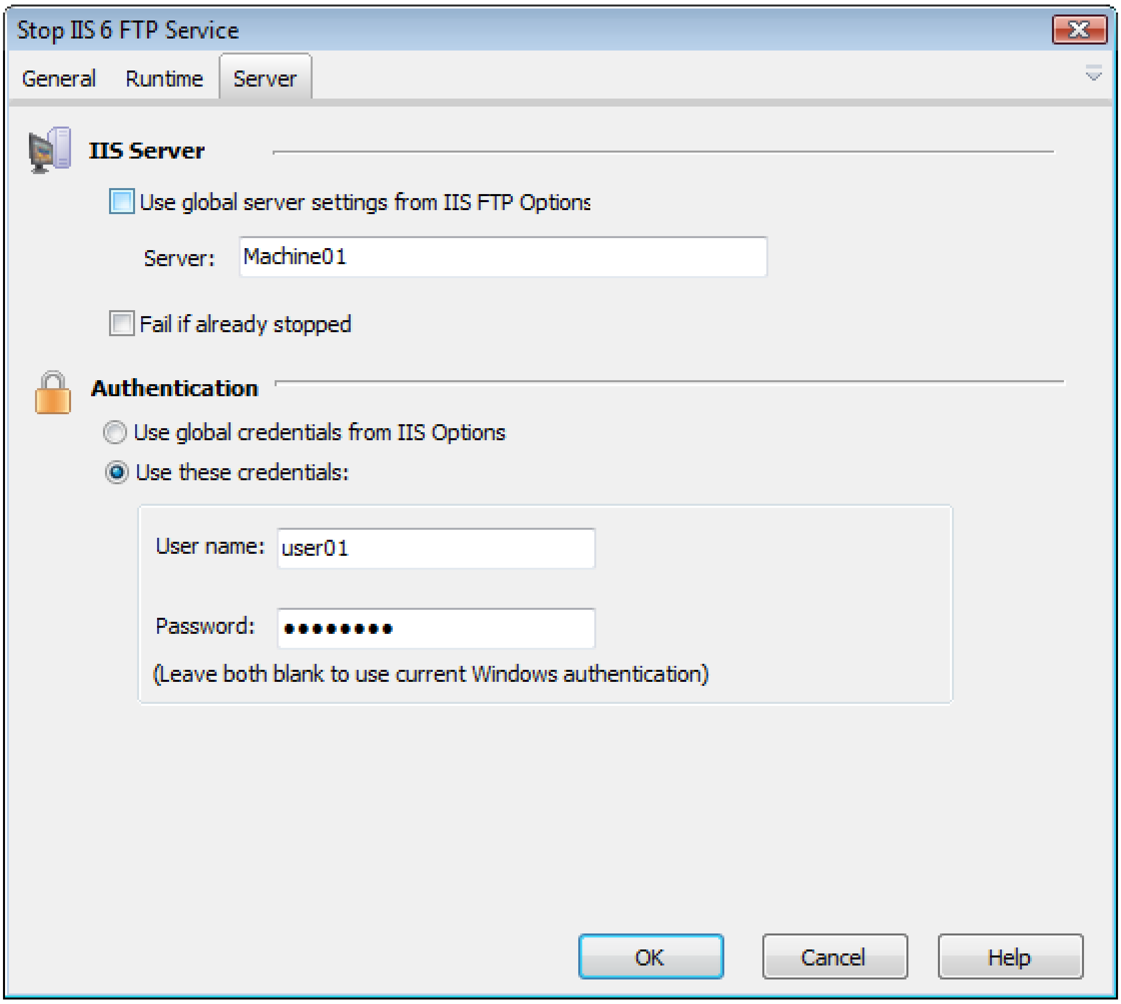
(The WMI interface is not available in IIS versions below 6.0.)

6.16.3.1 Server Selection

[Automise Professional Edition]

For each IIS FTP action, you can specify the IIS Server as either a global server (specified in the IIS 6 FTP Options) or a particular server name. You can also specify credentials as either the credentials specified in the Options, or a particular set of credentials.

If the credentials are left blank, Windows authentication with the credentials of the currently logged-in user will be used.



IIS Server

□ Use global server settings from IIS FTP Options

Server: Machine01

□ Fail if already stopped

Authentication

○ Use global credentials from IIS Options

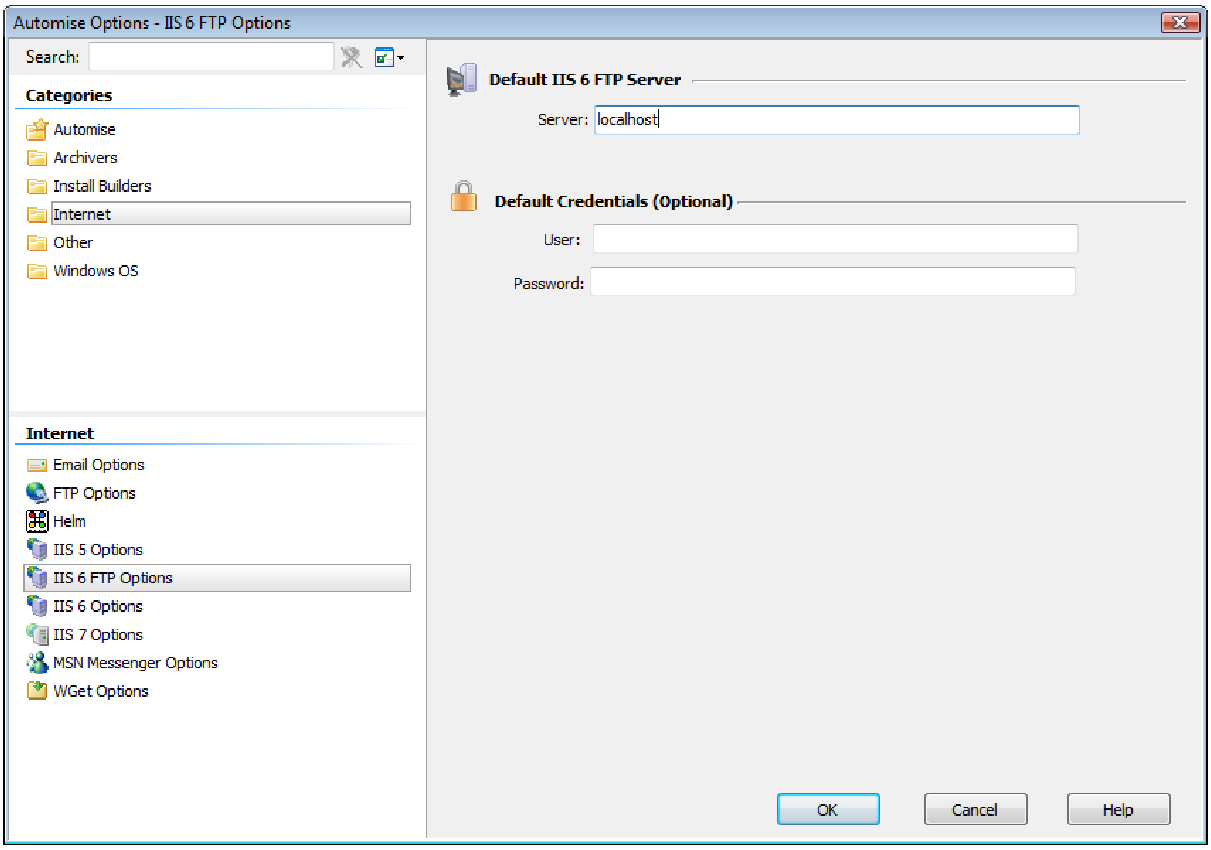
● Use these credentials:

User name: user01

Password: •••••••••

(Leave both blank to use current Windows authentication)

To select the global server name and credentials, choose Options from the Tools menu and then click on the IIS 6 Options tab under 'Internet':



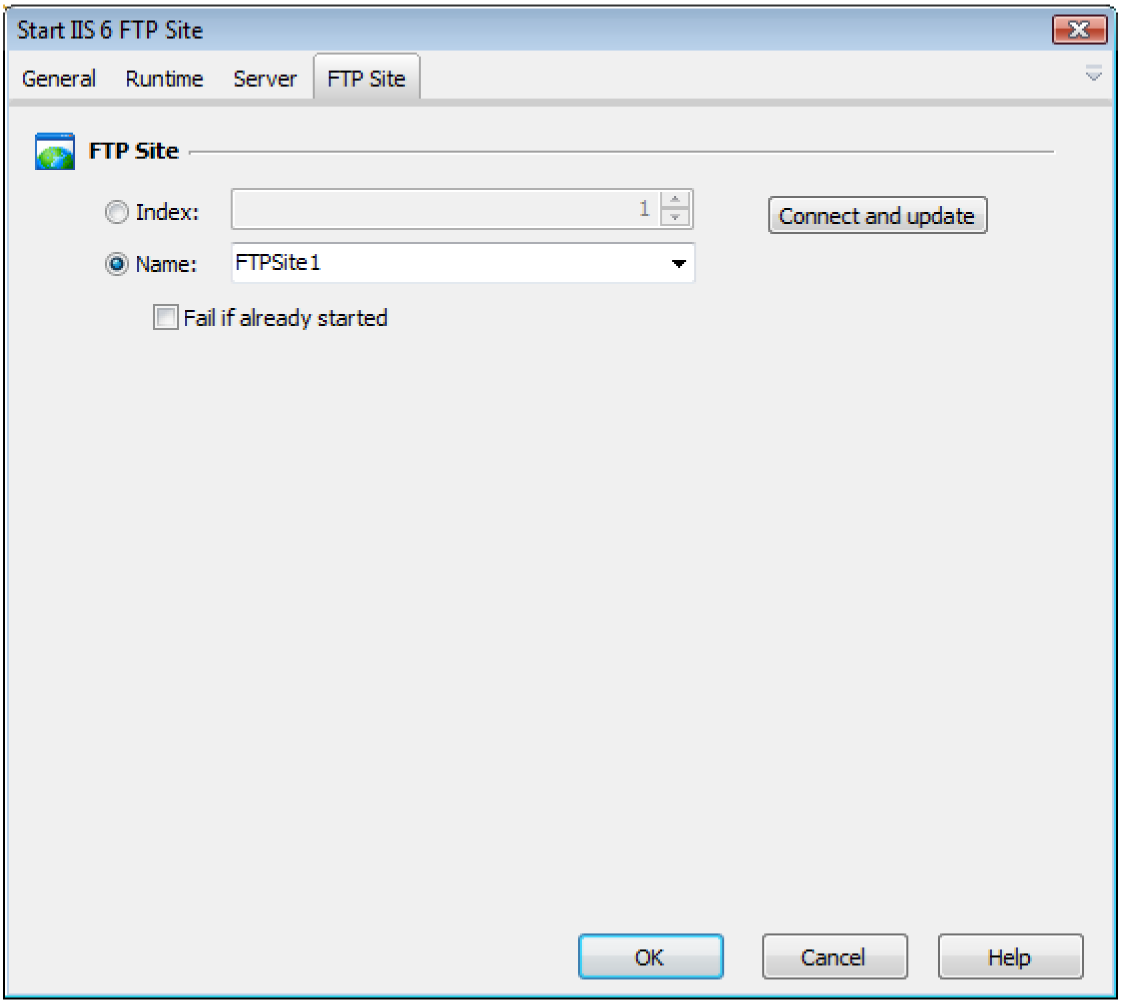
Server: localhost

6.16.3.2 FTP Site Selection

[Automise Professional Edition]

Each action which works on an FTP Site has an FTP Site property page, similar to the one shown below.

You can specify IIS 6 FTP sites by index number (also called the Identifier, which can be seen under the FTP Sites node in the IIS Manager application) or by name (a site's name is also called its Description.)

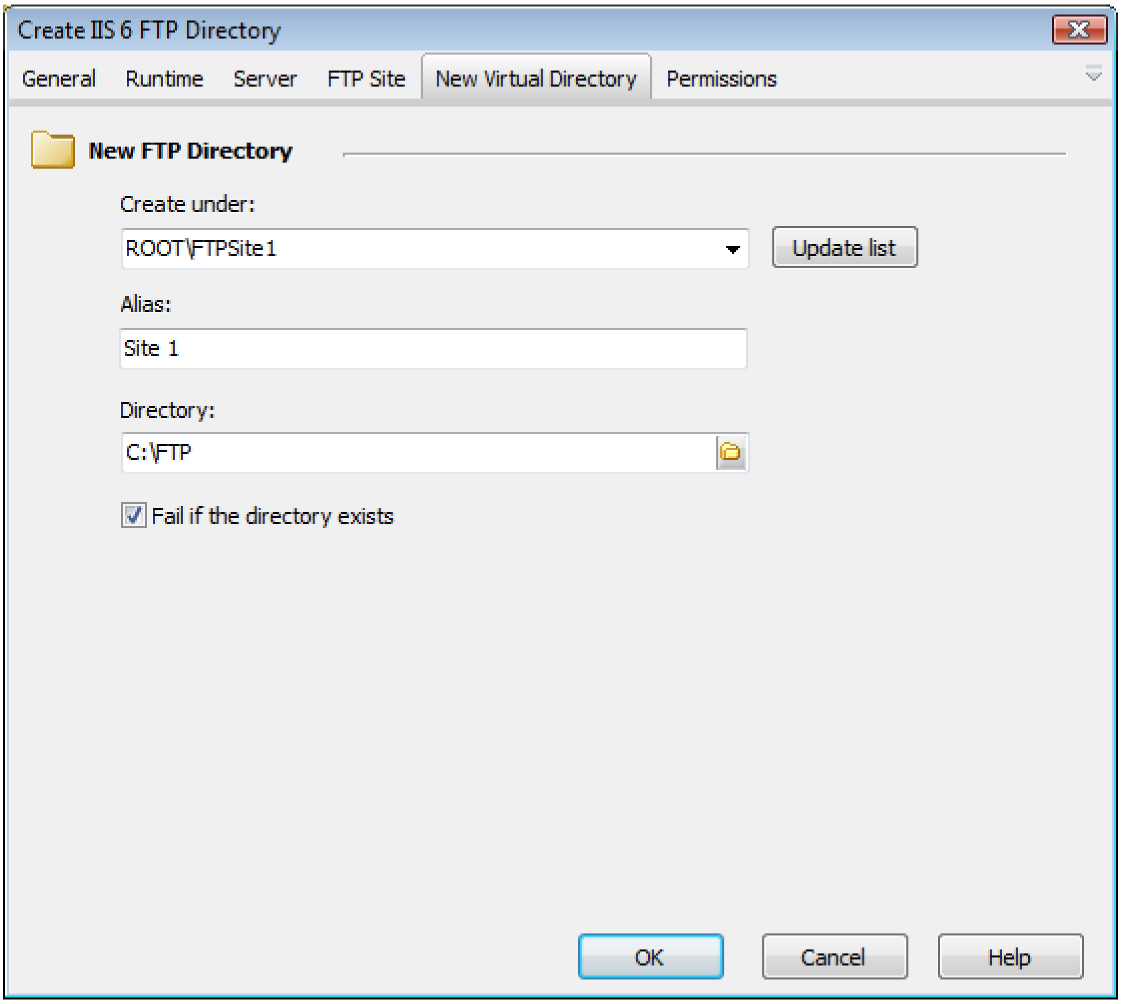


6.16.3.3 Create FTP Directory Action

[Automise Professional Edition]

Server & Website

Specify the Server and FTP site to use via the Server Selection & FTP Site Selection tabs, respectively.



Create under: ROOT\FTPSite1

Alias: Site 1

Directory: C:\FTP

☑ Fail if the directory exists

Create under

Specify an existing IIS FTP Directory to be the parent of the new directory. Click the 'Update list' button to download a list
of FTP Virtual Directories from the specified IIS Server.

Alias

Specify the name by which you would like the new directory to be known.

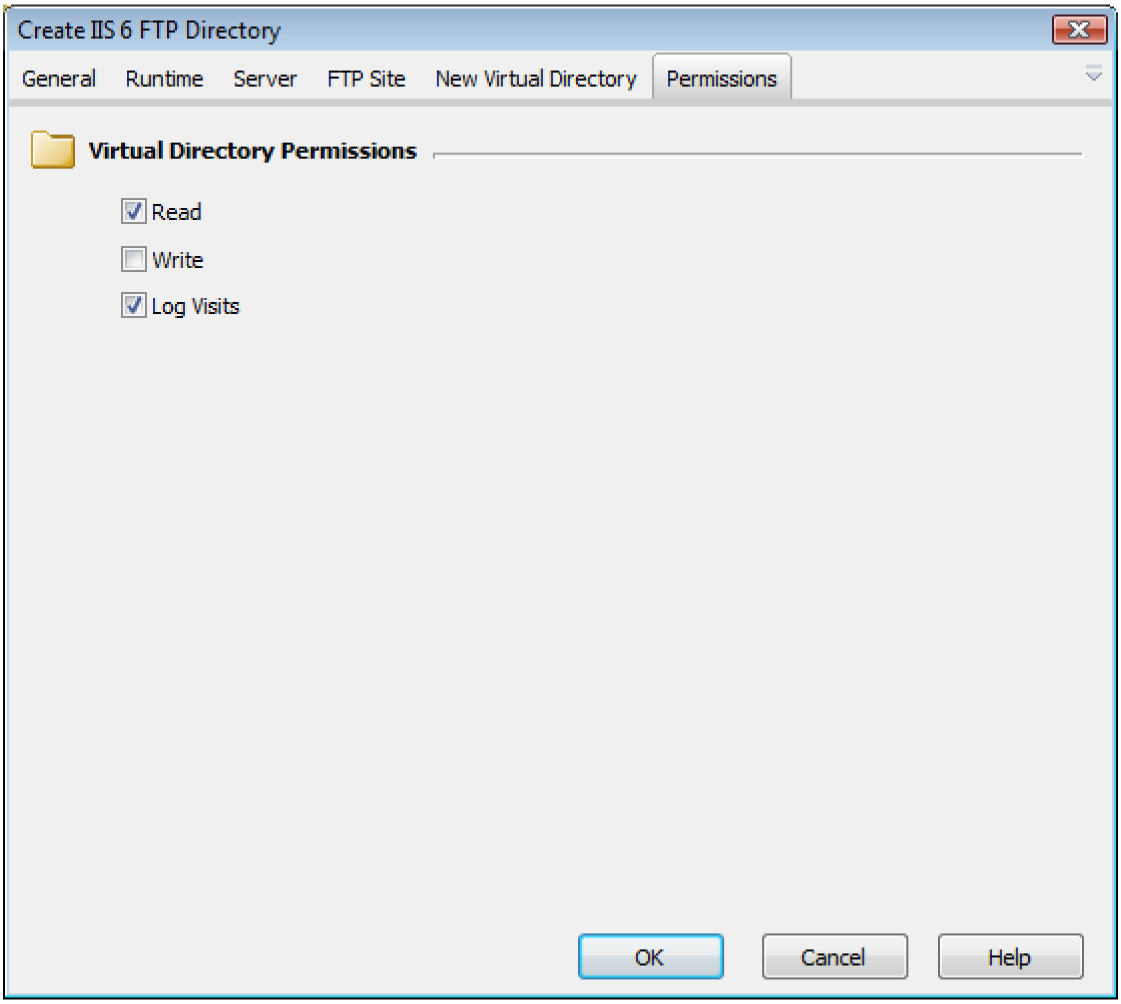
Directory

Specify the local directory for the contents of the new virtual directory.

Fail if the directory exists

If this option is enabled and the directory already exists, the action will fail. Otherwise, the action will edit the properties of the pre-existing virtual directory.

Permissions Tab



Virtual Directory Permissions

☑ Read

□ Write

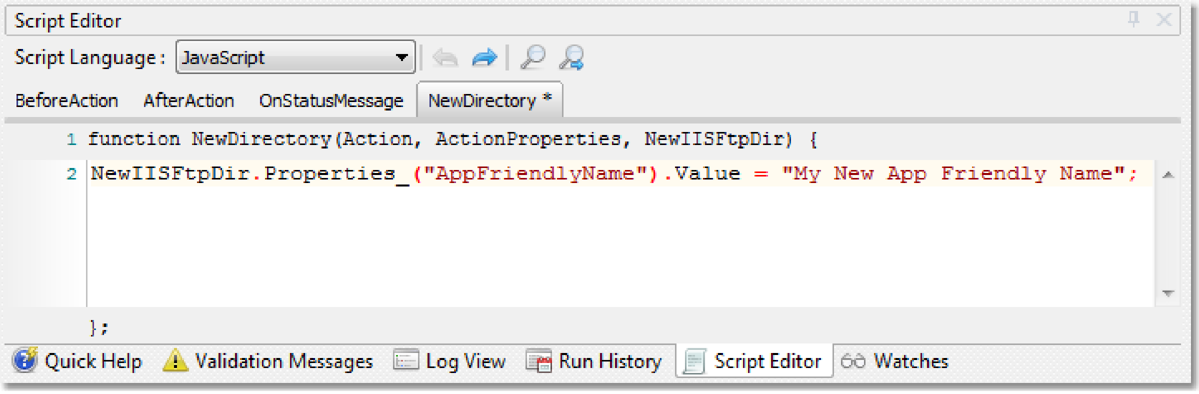
☑ Log Visits

Set the permissions you want FTP users to have on the new Directory.

Set "Log Visits" if you want accesses to the new directory to appear in the IIS Log.

Script Events

The Create FTP Directory Action has an extra script event, named NewDirectory, which is called once the new directory has been set up. It allows you to set properties which may not be available in the action, or perform more complicated operations on the new directory.



```
function NewDirectory(Action, ActionProperties, NewIISFtpDir) {
NewIISFtpDir.Properties_("AppFriendlyName").Value = "My New App Friendly Name";
};
```

The **NewIISFtpDir parameter** is an WMI settings object of type IIsFtpVirtualDirSetting (see the MSDN Documentation.)

You can retrieve and set the virtual directory's properties directly using the syntax shown in the above example. See the MSDN Documentation for a full list of available properties. Unfortunately, code completion is not available on this object.

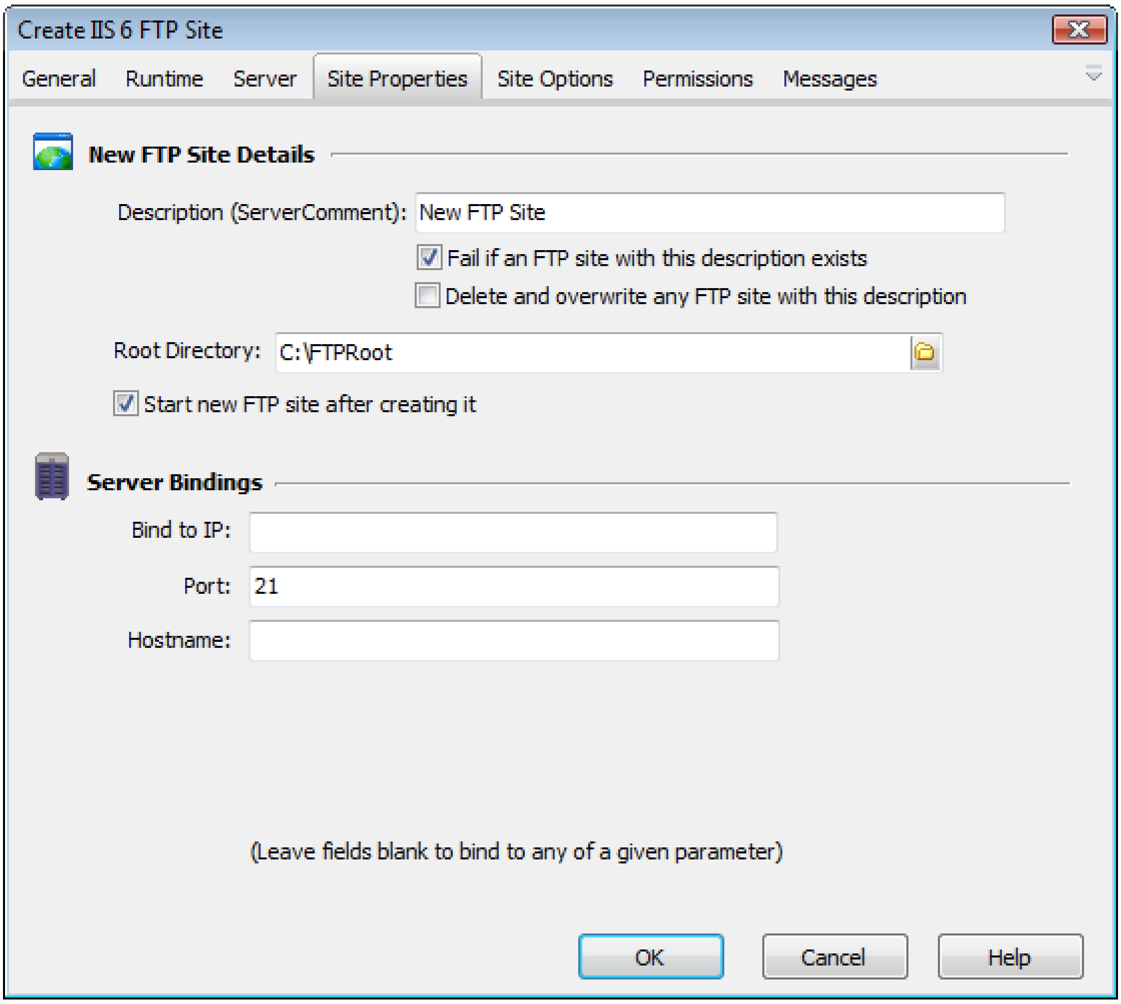
There is no need to call the .Put_() method to save any changes which are made (this is done automatically when the event completes.)

6.16.3.4 Create FTP Site Action

[Automise Professional Edition]

Use the Create FTP Site Action to create a new FTP site, complete with a root directory.

Specify the IIS Server to connect to under the Server tab.



New FTP Site Details

Description (ServerComment): New FTP Site

☑ Fail if an FTP site with this description exists

□ Delete and overwrite any FTP site with this description

Root Directory: C:\FTPRoot

☑ Start new FTP site after creating it

Server Bindings

Bind to IP:

Port: 21

Hostname:

(Leave fields blank to bind to any of a given parameter)

Description

Enter a description that the new FTP site will be known by.

"Fail if an FTP site with this description exists"

Check this box to fail if another FTP site already has the same name.

"Delete and overwrite any FTP site with this description"

Check this box to replace an existing FTP site with a matching name.

If neither of these options are selected and another FTP site exists with a matching name, the new site is created alongside it.

Root Directory

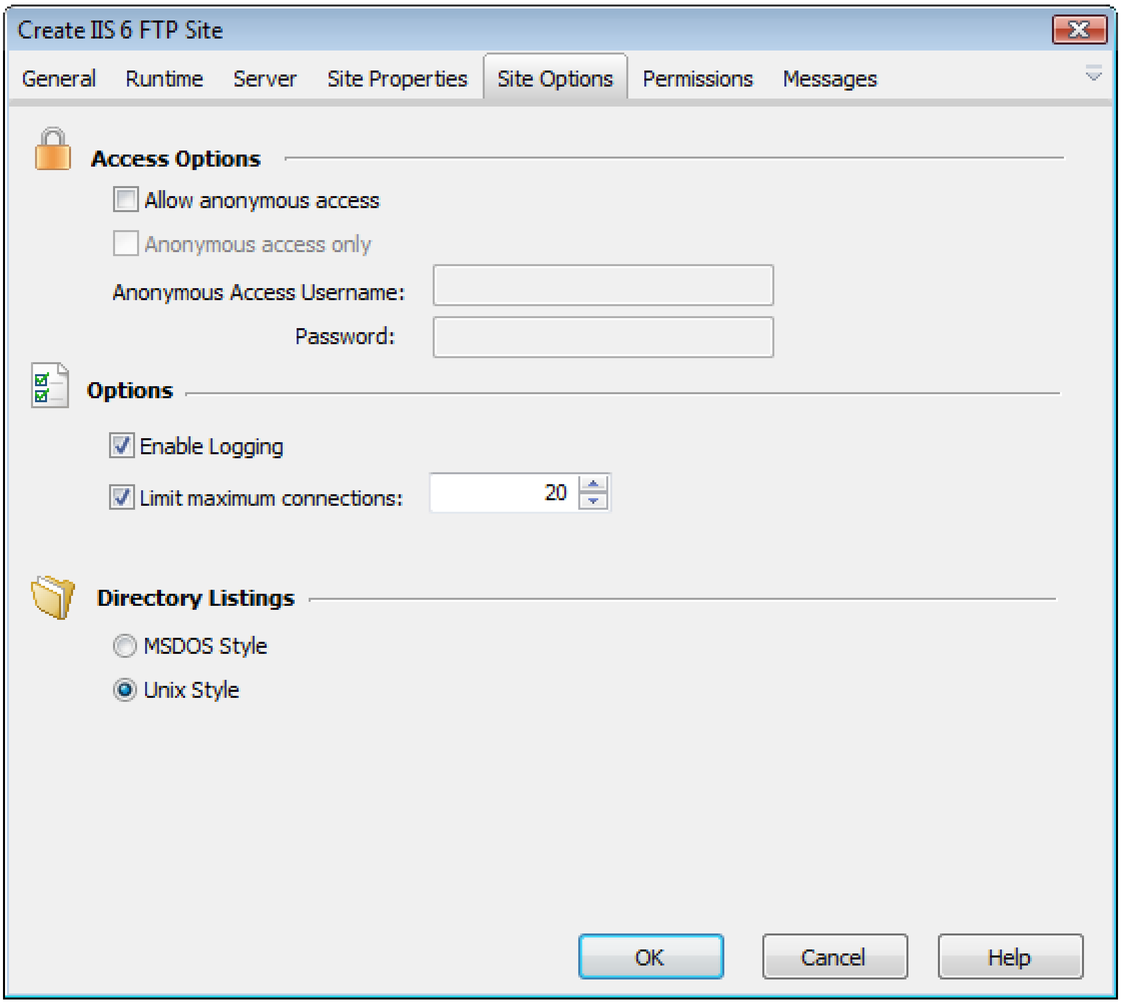
Specify the directory (on the IIS server) where you want files to be hosted for the root FTP directory.

Start new FTP site after creating it

Enable this option to start the new FTP site when the action is run. Otherwise, the site is created in the Stopped state.

Server Bindings

You can specify IP, Port and Hostname bindings for the new FTP site. Leave any field blank in order to bind to any of a given parameter.



□ Allow anonymous access

□ Anonymous access only

Anonymous Access Username:

Password:

☑ Enable Logging

☑ Limit maximum connections: 20

○ MSDOS Style

● Unix Style

Allow anonymous access

Enable this option to allow a user to log in as "anonymous" (ie anonymous FTP.)

Anonymous access only

If this option is enabled, the anonymous user will be the only user who is allowed to log in.

Anonymous Access Username and Password

Specify a Windows username and password on the server. When users log in as anonymous, they are in fact logging in as this user.

Enable Logging

Enable this option to log all access to the FTP site. Note that you can also disable logging individually for different FTP directories (See the Permissions tab.)

Limit Maximum Connections

Specify a number here in order to limit the number of concurrent connections to this FTP site.

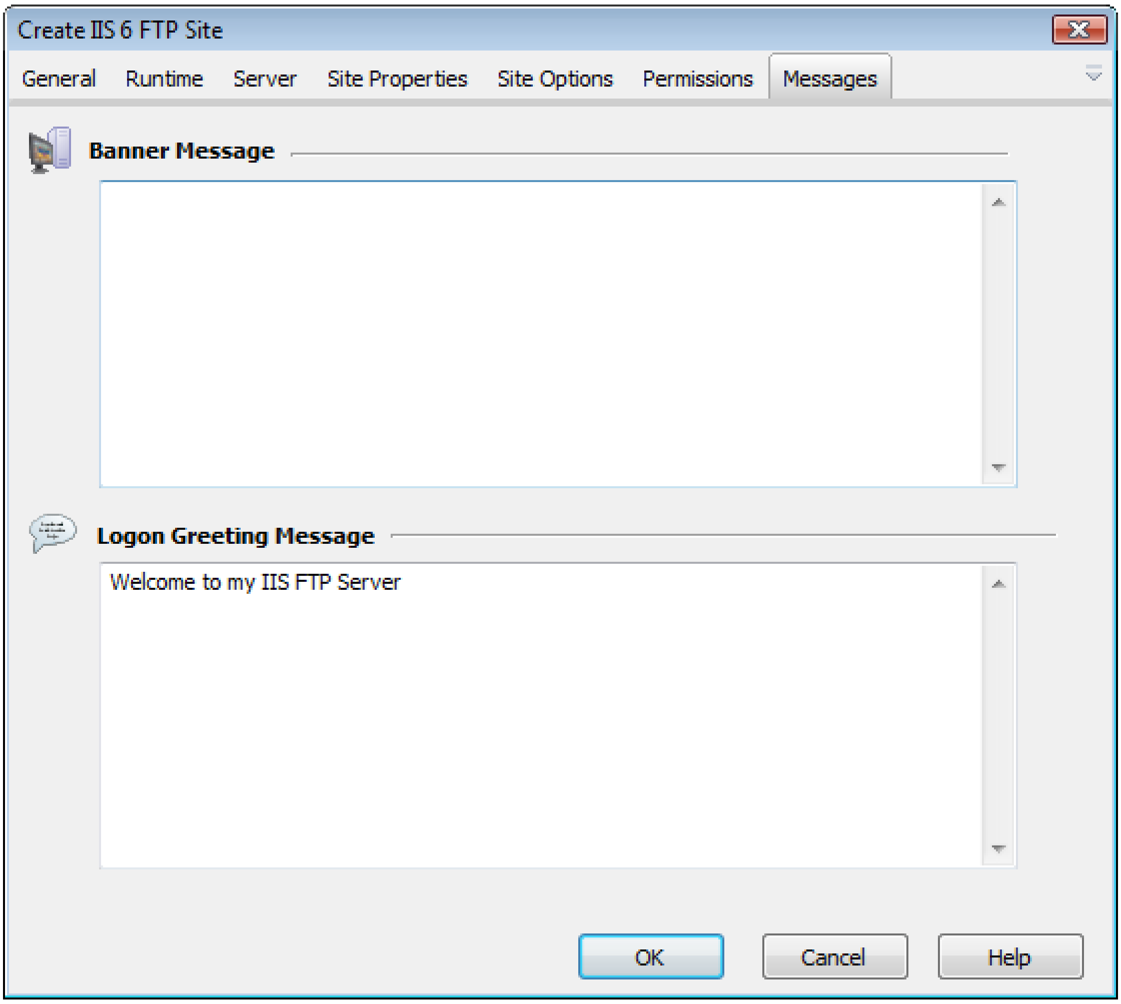
Directory Listings

The default directory listing format can be "MSDOS Style" or "Unix Style".

Permissions Tab

Determines the permissions for the root FTP directory. See the Permissions section of the Create FTP Directory Action for more details.

NB: There is an observed bug in IIS 6.0 whereby even though "Read" access is set on the root directory of a new FTP site, the Read checkbox appears unchecked when viewed in the IIS Management console. Despite the checkbox appearing unchecked, the root directory does support Read access.

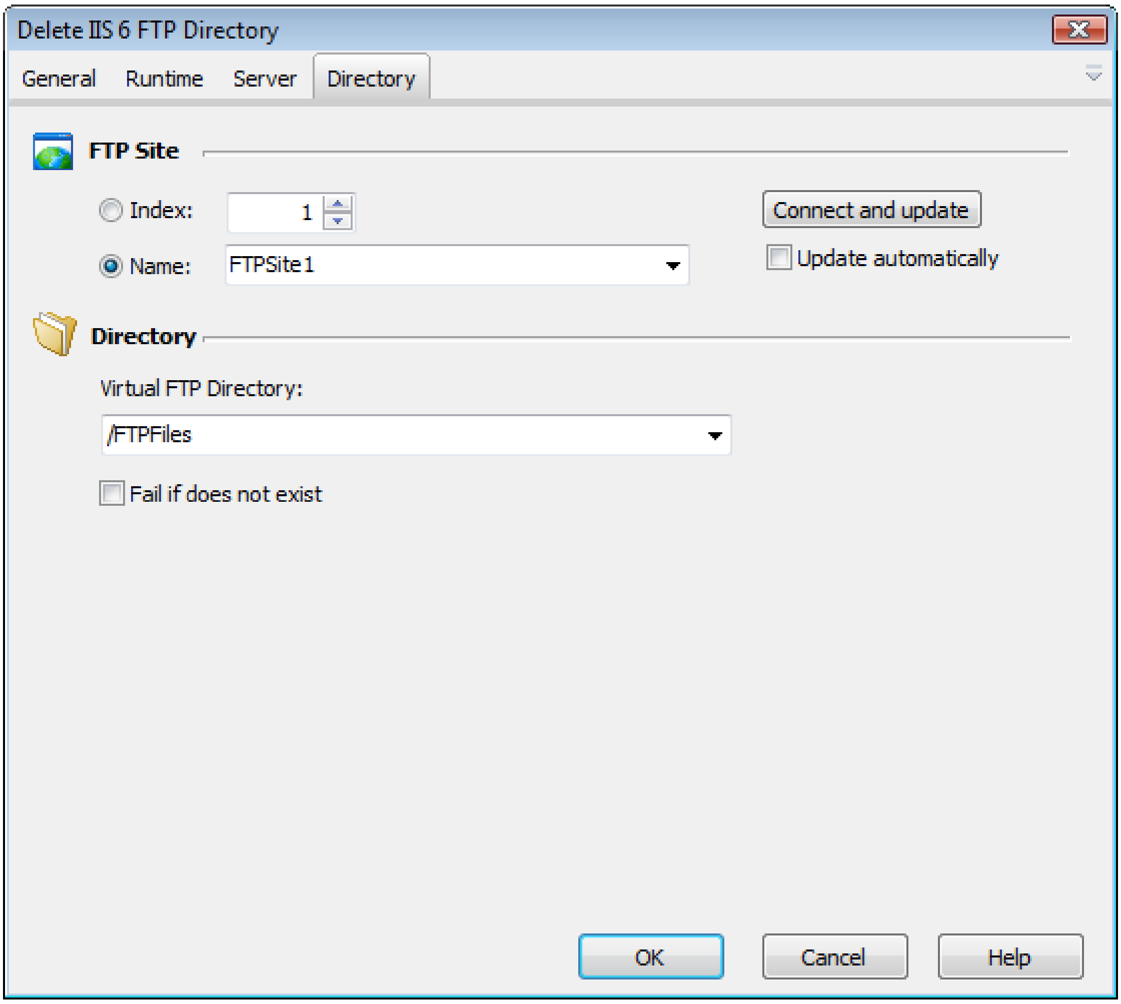


Welcome to my IIS FTP Server

The Banner Message and Logon Greeting Message are sent to the user when they first connect to site, and when they log in (respectively.)

6.16.3.5 Delete FTP Directory Action

[Automise Professional Edition]



FTP Site

○ Index: 1

● Name: FTPSite1

□ Update automatically

Directory

Virtual FTP Directory: /FTPFiles

□ Fail if does not exist

FTP Site

Choose an FTP site name or index (for details see the FTP site Selection tab.) Check the "Update automatically" box to have changes in the web site selection automatically reload the directory listings combo box.

Virtual FTP Directory

Choose the FTP directory to delete. Click "Connect and update" to have a list of existing virtual directories loaded into the combo box.

Fail if does not exist

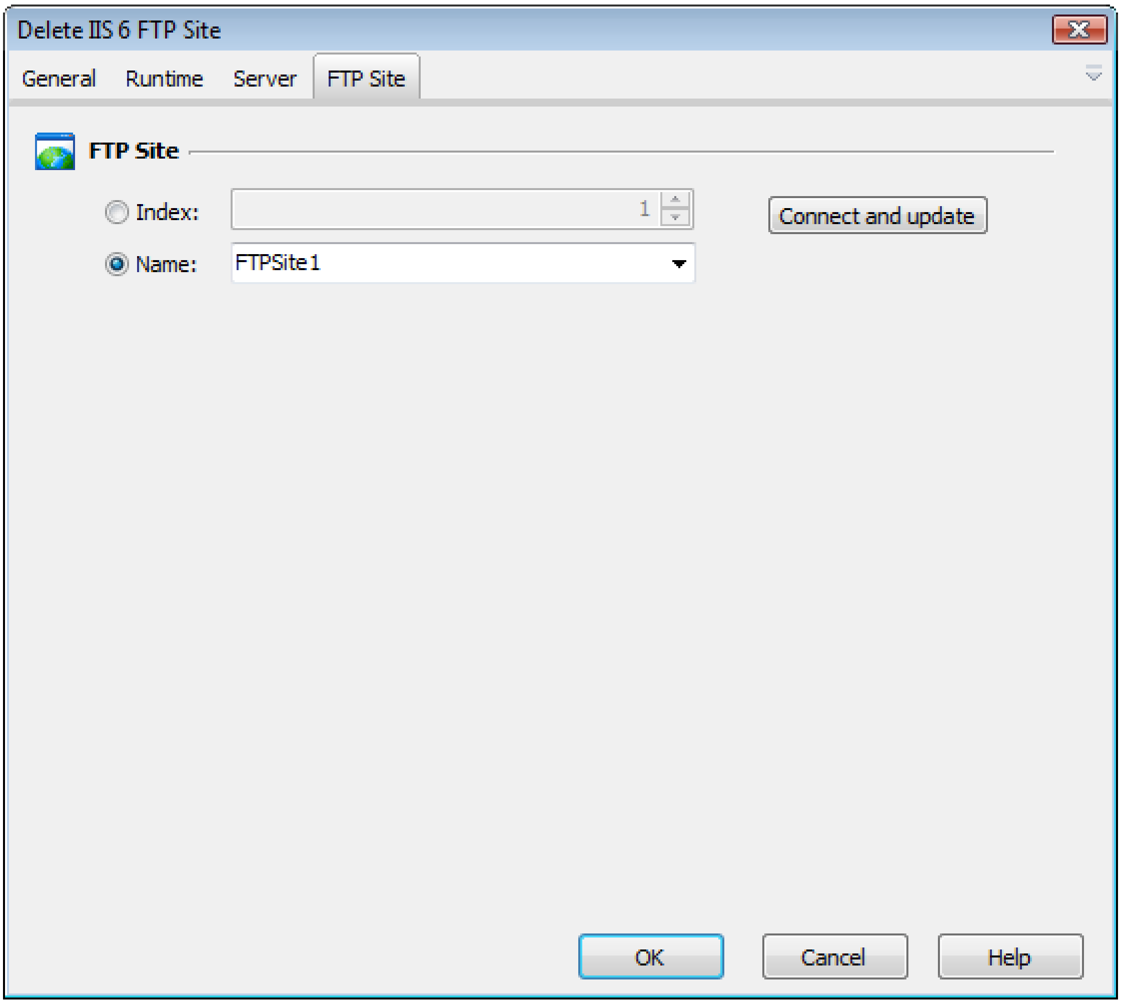
Check this box if you want the action to fail should the specified directory not exist.

6.16.3.6 Delete FTP Site Action

[Automise Professional Edition]

The Delete FTP Site Action allows you to remove an existing IIS FTP site.

Specify the Server to connect to, and the FTP site to delete, via the Server Selection & FTP Site Selection tabs, respectively.



○ Index: 1

● Name: FTPSite1

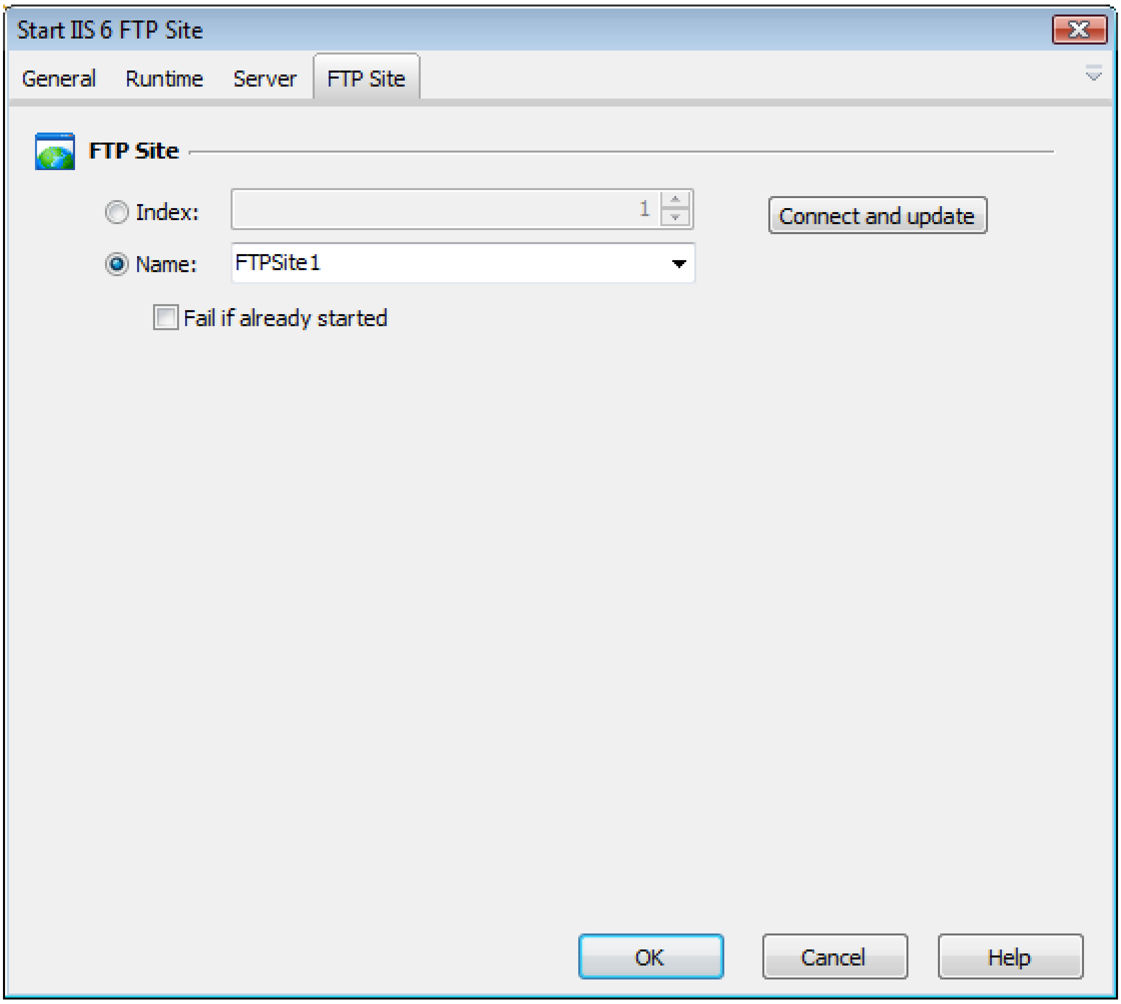
6.16.3.7 Start / Stop / Pause FTP Site Actions

[Automise Professional Edition]

The Start, Stop & Pause FTP site actions allow you to start and stop individual IIS 6 FTP Sites.

Use the Server Selection tab to choose the server you wish to administer, and credentials for the connection.
Then use the FTP Site Selection tab to choose the FTP site you wish to modify.

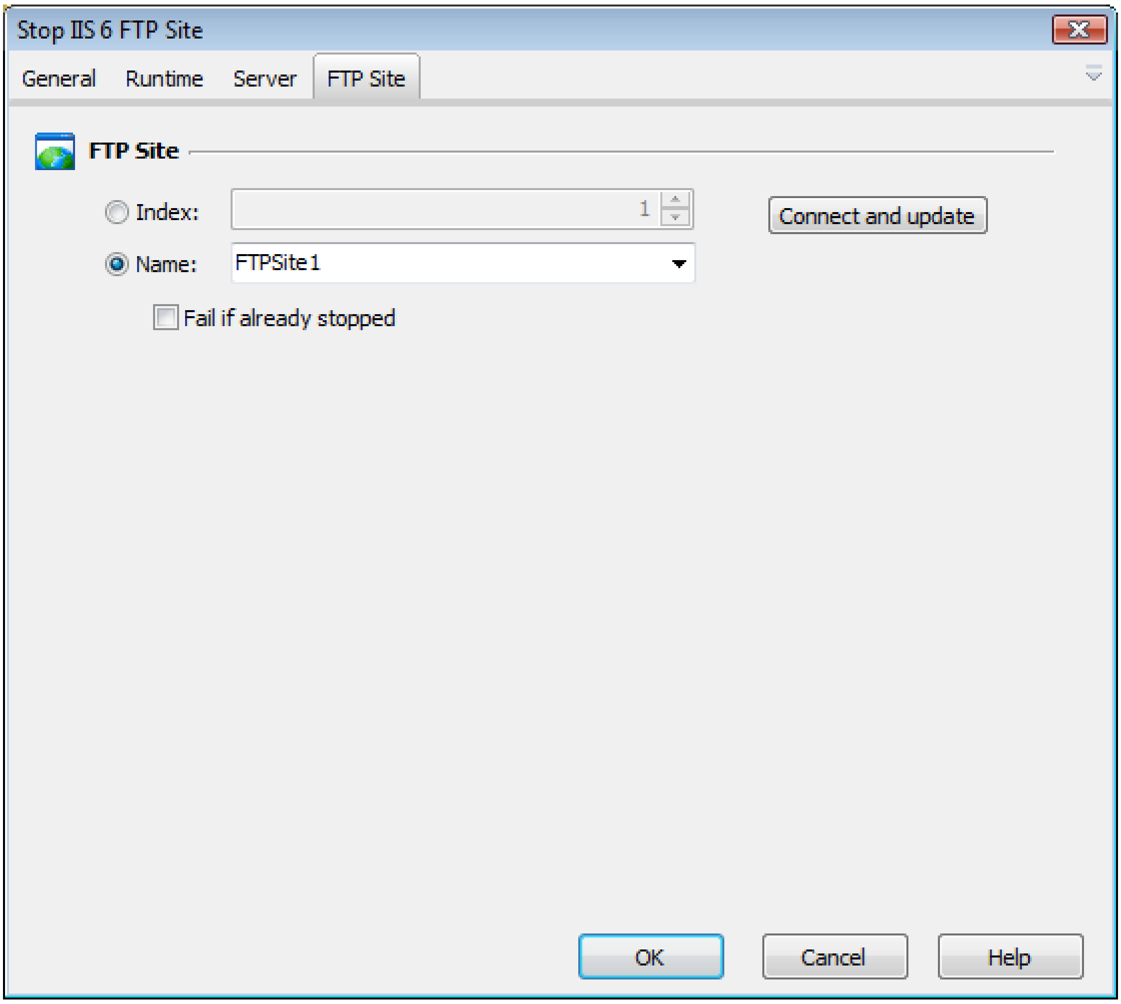
The actions can be set not to fail if the selected web site is already in the desired state (running/stopped.)



○ Index: 1

● Name: FTPSite1

□ Fail if already started



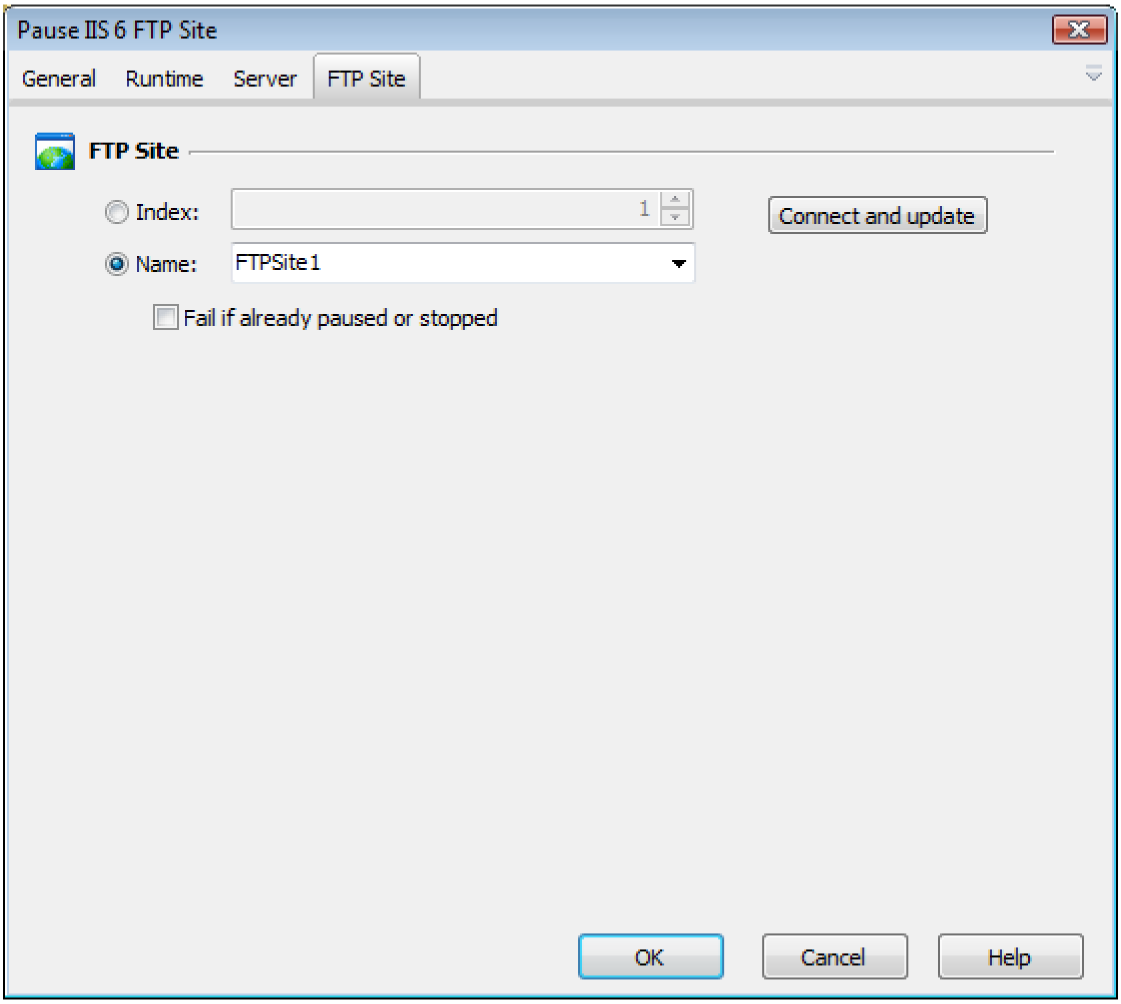
Stop IIS 6 FTP Site

FTP Site

○ Index: 1

● Name: FTPSite1

□ Fail if already stopped



FTP Site

○ Index: 1

● Name: FTPSite1

□ Fail if already paused or stopped

6.16.3.8 Start / Stop / Restart FTP Service Actions

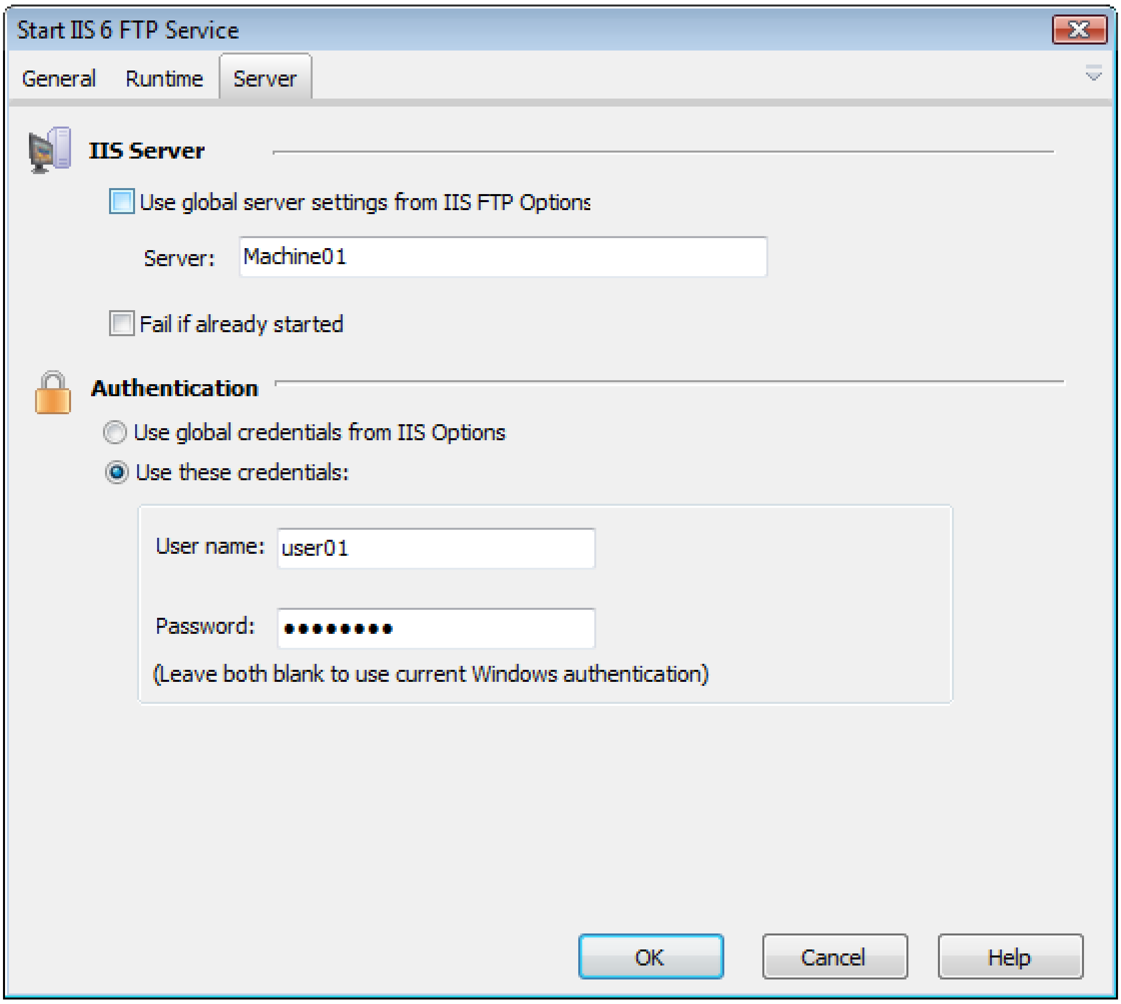
[Automise Professional Edition]

The Start, Stop & Restart FTP Service actions use WMI to start and stop the IIS FTP service (MSFTPSVC) on a server running IIS 6.

Use the Server Selection tab to choose the server you wish to administer, and to specify credentials for connecting to the server.

Each of the actions can be set not to fail if the service is already started/stopped (in the case of the Restart action, the service will be started even if it is initially stopped.)

It is recommended that you use the Restart action instead of a Stop followed immediately by a Start action (the Restart action waits to ensure the service is fully stopped before restarting it.)



□ Use global server settings from IIS FTP Options

Server: Machine01

□ Fail if already started

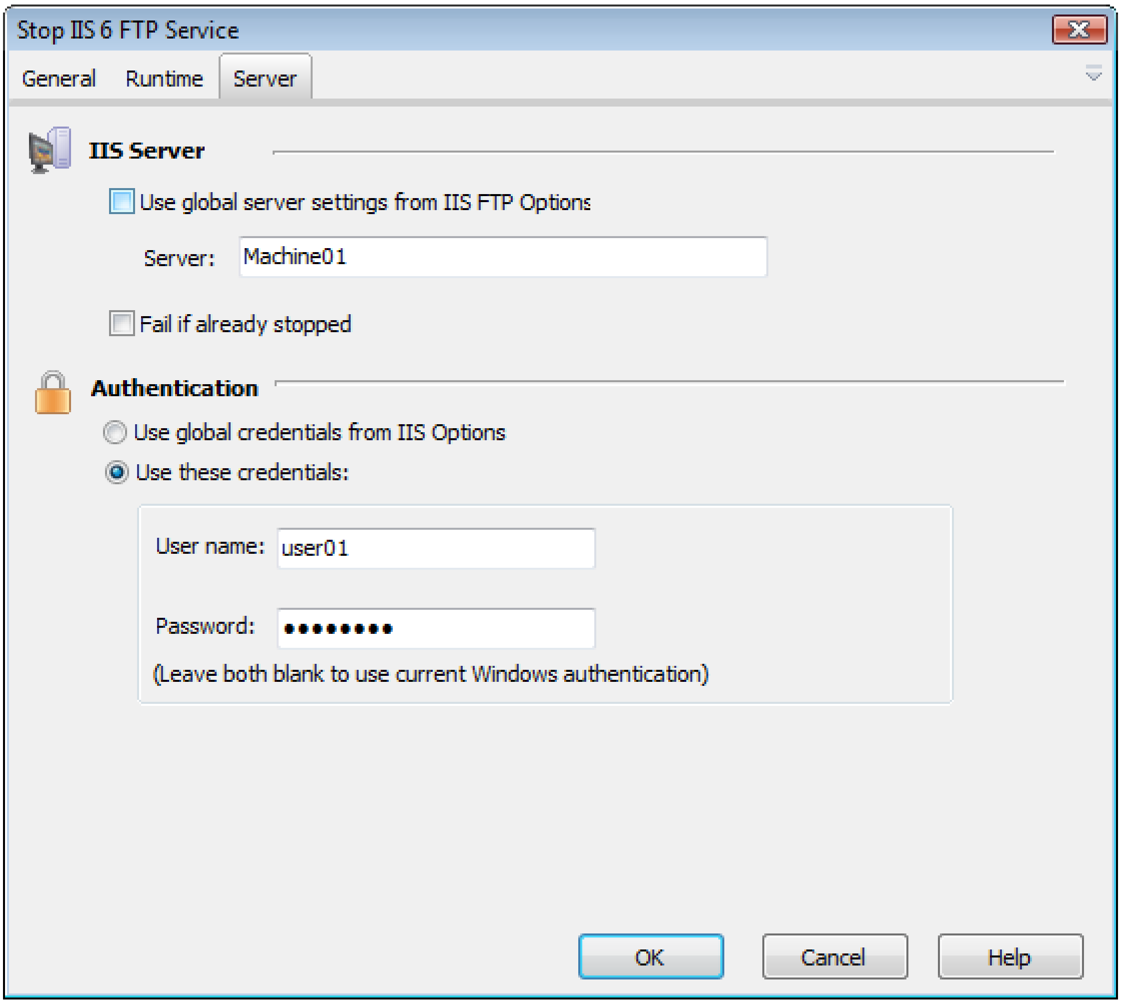
○ Use global credentials from IIS Options

● Use these credentials:

User name: user01

Password: ●●●●●●●●●

(Leave both blank to use current Windows authentication)



Stop IIS 6 FTP Service

IIS Server

□ Use global server settings from IIS FTP Options

Server: Machine01

□ Fail if already stopped

Authentication

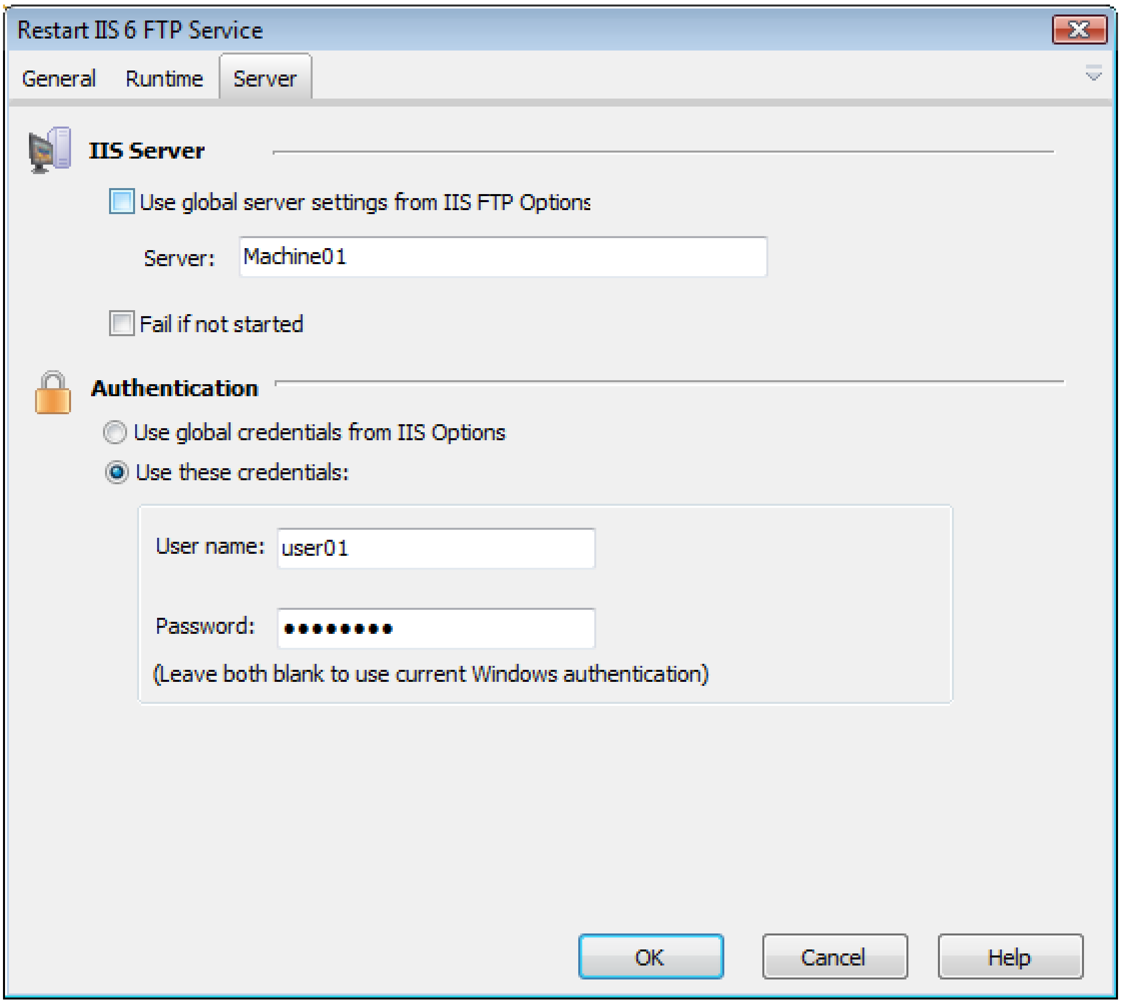
○ Use global credentials from IIS Options

● Use these credentials:

User name: user01

Password: ●●●●●●●●●

(Leave both blank to use current Windows authentication)



□ Use global server settings from IIS FTP Options

Server: Machine01

□ Fail if not started

○ Use global credentials from IIS Options

● Use these credentials:

User name: user01

Password: ●●●●●●●●●

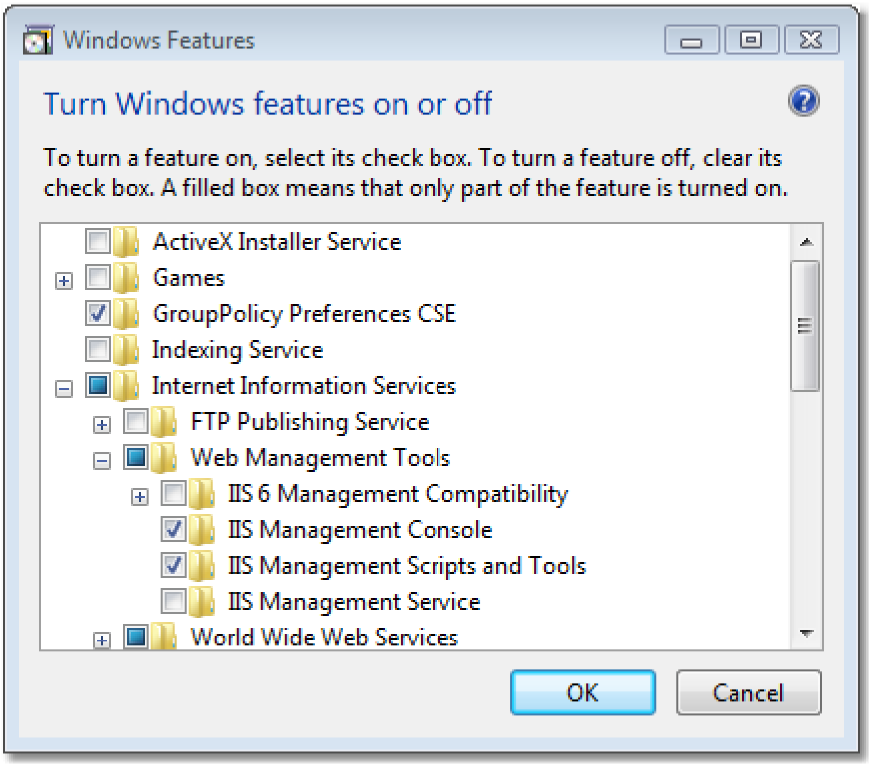
(Leave both blank to use current Windows authentication)

6.16.4 IIS 7

The IIS 7 actions allow you to connect to a local or remote server instance and manage web sites, virtual directories, applications and application pools.

Requirements

The IIS 7 actions require that the 'IIS Management Scripts and Tools' windows feature is installed on the machine hosting the IIS instance.



Windows Features

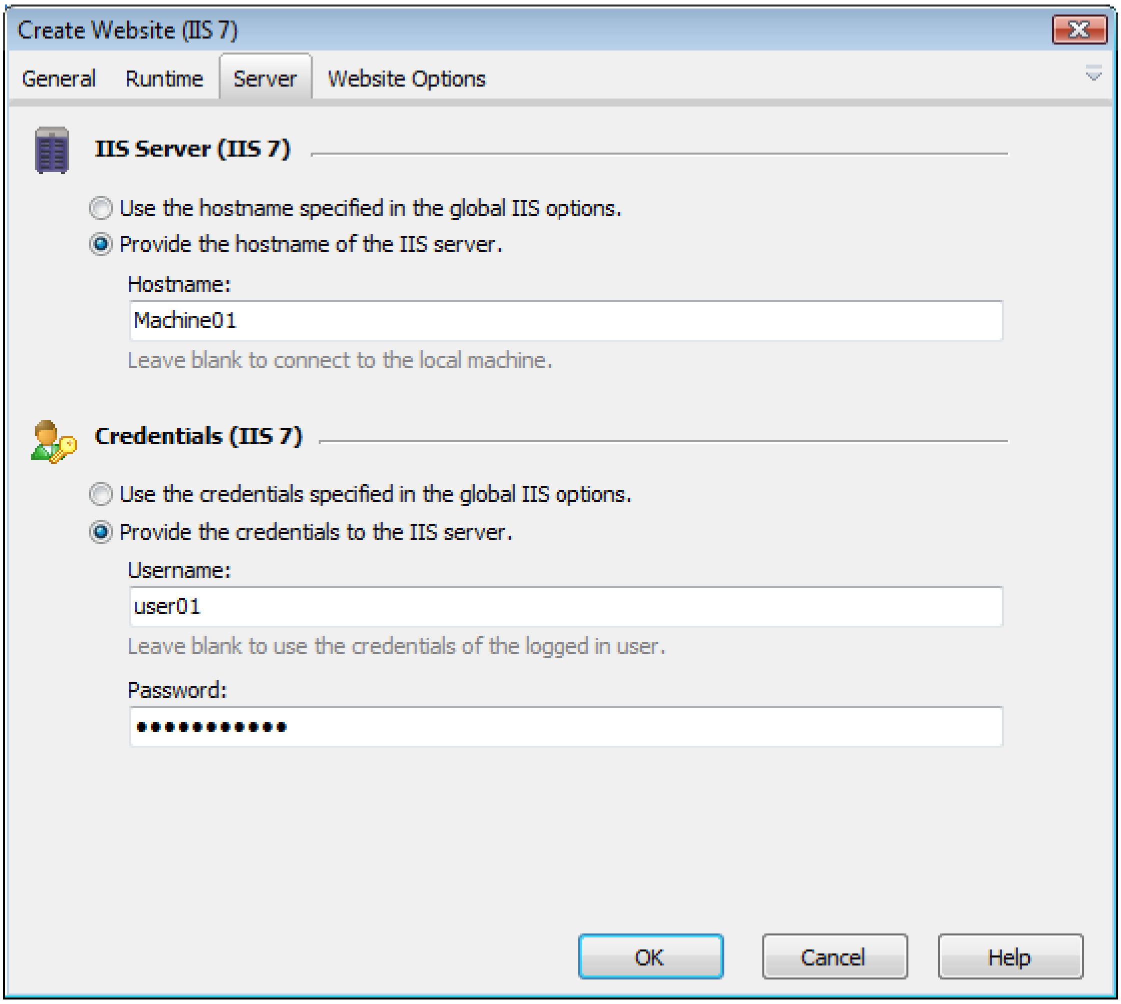
Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- □ ActiveX Installer Service
- ⊞ □ Games
- ☑ GroupPolicy Preferences CSE
- □ Indexing Service
- ⊟ ■ Internet Information Services
  - ⊞ □ FTP Publishing Service
  - ⊟ ■ Web Management Tools
    - ⊞ □ IIS 6 Management Compatibility
    - ☑ IIS Management Console
    - ☑ IIS Management Scripts and Tools
    - □ IIS Management Service
  - ⊞ ■ World Wide Web Services

6.16.4.1 Server Selection

Each IIS 7 action has a common server tab which is used to configure which machine to connect to and which credentials to provide.



Create Website (IIS 7)

IIS Server (IIS 7)

○ Use the hostname specified in the global IIS options.

● Provide the hostname of the IIS server.

Hostname: Machine01

Leave blank to connect to the local machine.

Credentials (IIS 7)

○ Use the credentials specified in the global IIS options.

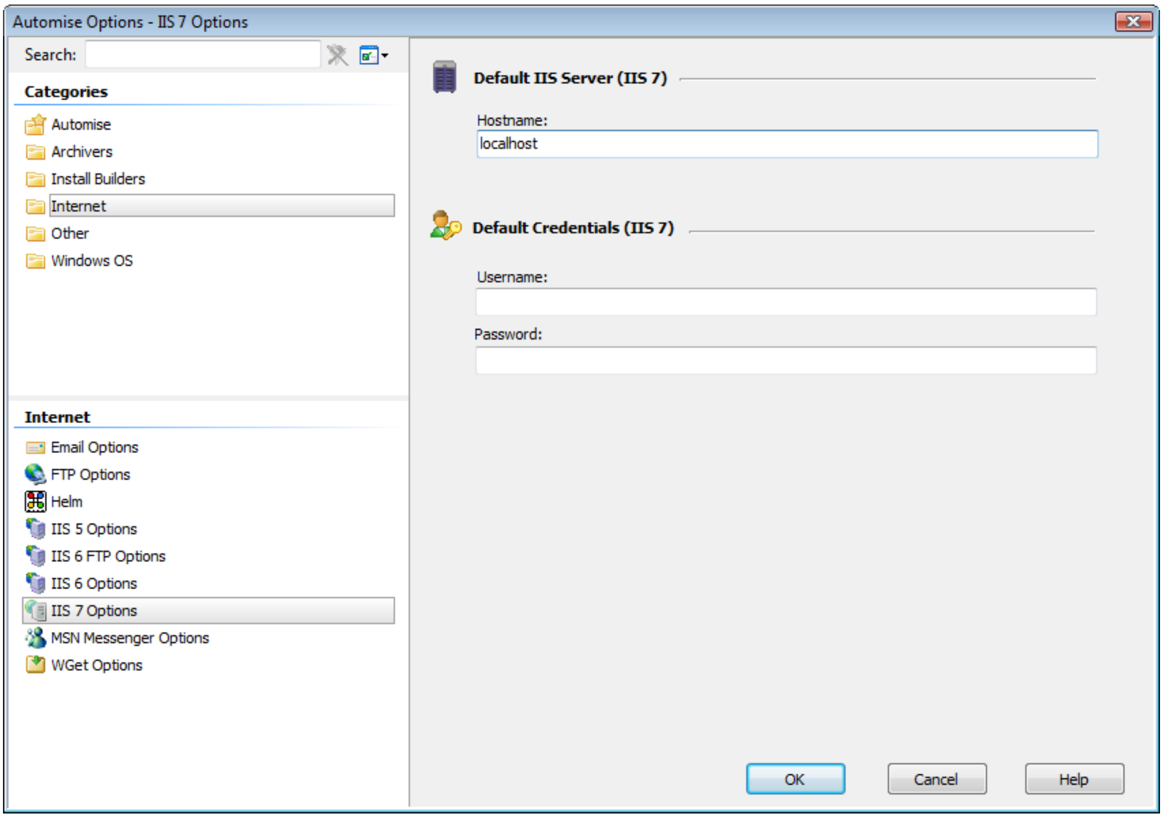
● Provide the credentials to the IIS server.

Username: user01

Leave blank to use the credentials of the logged in user.

Password: ●●●●●●●●●●●●

The IIS 7 actions can be configured to use the options specified in the global options or to have the settings provided on a per-action basis. To specify the global options go to the Automise options and navigate to the IIS 7 options page.



Automise Options - IIS 7 Options

Search:

Categories

- Automise
- Archivers
- Install Builders
- Internet
- Other
- Windows OS

Internet

- Email Options
- FTP Options
- Helm
- IIS 5 Options
- IIS 6 FTP Options
- IIS 6 Options
- IIS 7 Options
- MSN Messenger Options
- WGet Options

Default IIS Server (IIS 7)

Hostname: localhost

Default Credentials (IIS 7)

Username:

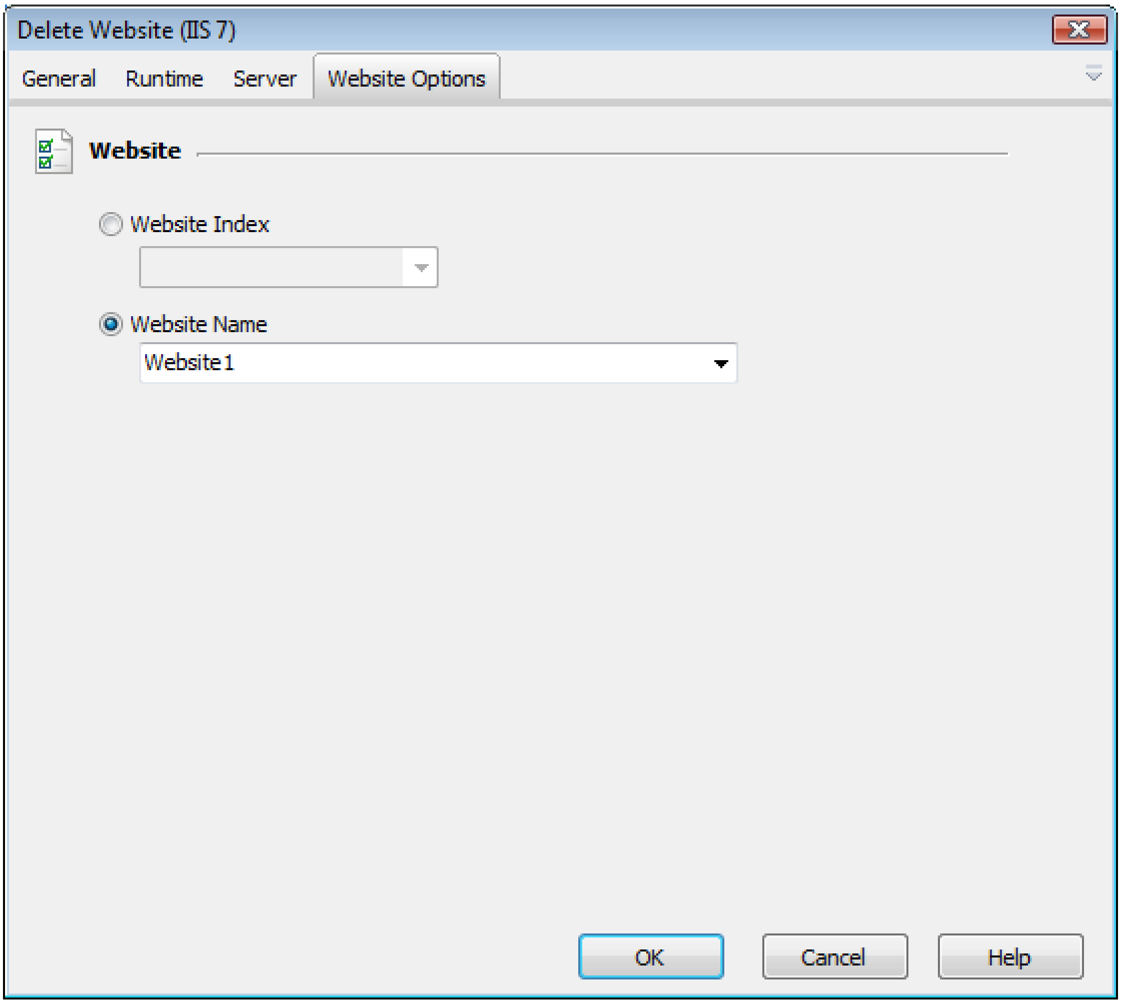
Password:

Hostname
The name of the machine to connect to, leave blank if you wish to connect to the local machine.

Username
The windows user account to use to connect to the machine hosting the IIS 7 server. Leave blank if you wish to use the currently logged in user.

6.16.4.2 Website Selection

Any IIS 7 action that deals with an existing website will have use a common website options tab. This tab allows you to select an existing web site either by ID or by the website's name.

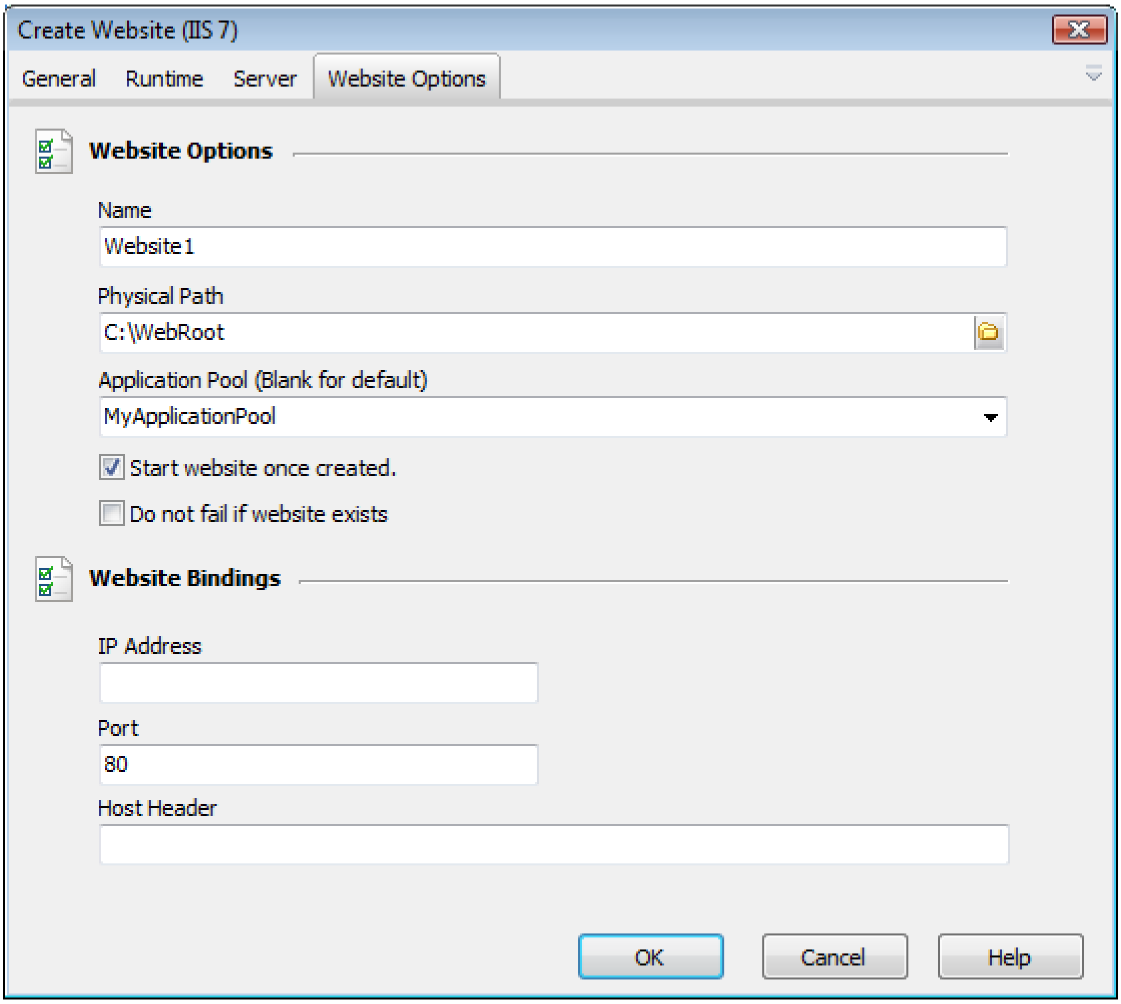


○ Website Index

● Website Name: Website1

6.16.4.3 Create Website Action

The Create Website (IIS 7) actions allows you to create a new website on an IIS 7 server.



Create Website (IIS 7)

Website Options

Name: Website1

Physical Path: C:\WebRoot

Application Pool (Blank for default): MyApplicationPool

☑ Start website once created.

☐ Do not fail if website exists

Website Bindings

IP Address:

Port: 80

Host Header:

Name
The name of the new web site, the name must be unique on the server.

Physical Path
A fully qualified path of a physical directory to which you want the web site mapped.

Start website once created
Determines whether the website will be started once it has been created.

IP Address
The IP Address to bind the new web site to, leave blank to bind to all unassigned IP addresses.

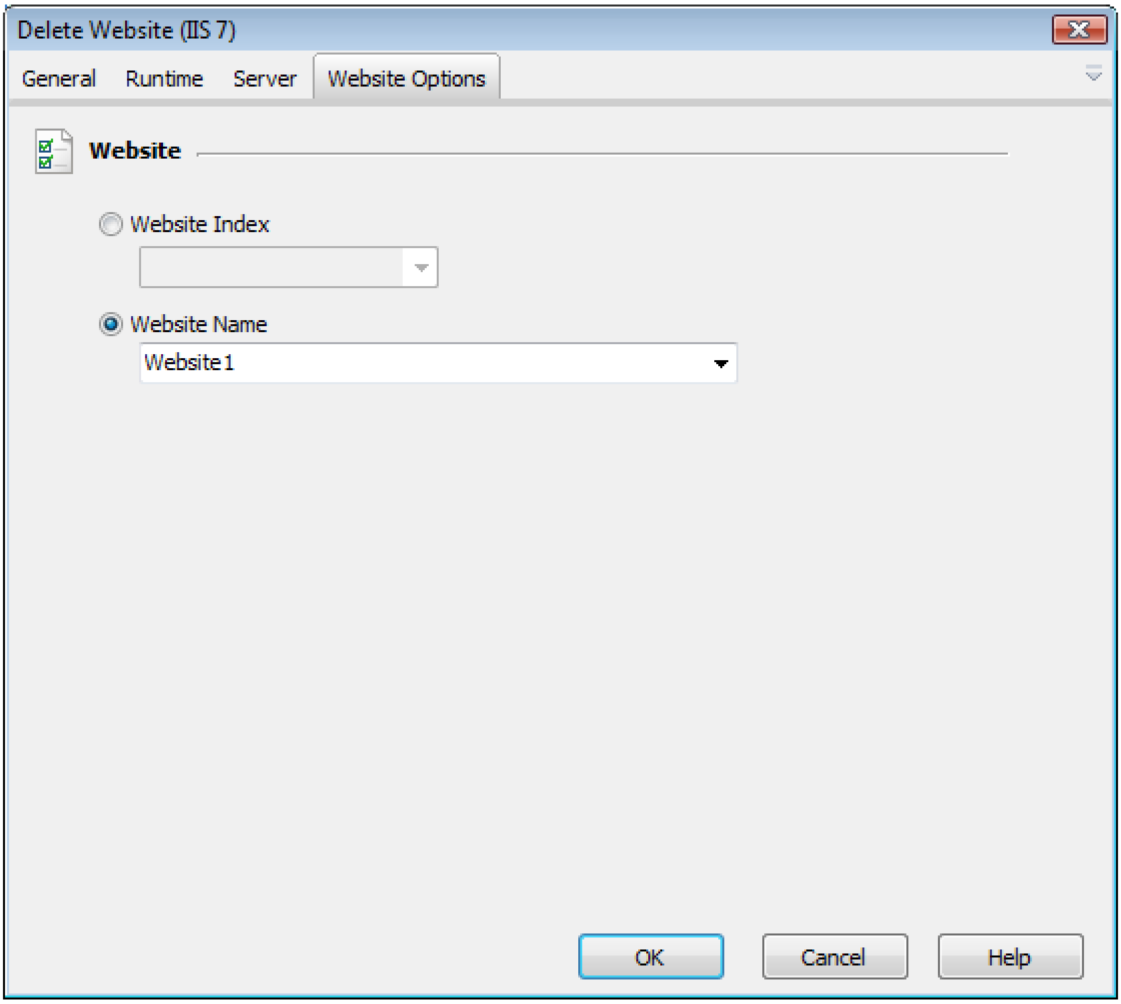
Port
The port which the new website should use. Default is port 80.

Host Header
The host header to bind the site to.

6.16.4.4 Delete Website Action

The Delete Website (IIS 7) action enables you to delete an existing website.

This action only requires that you specify the existing website to delete using the common website options tab.



Delete Website (IIS 7)

Website

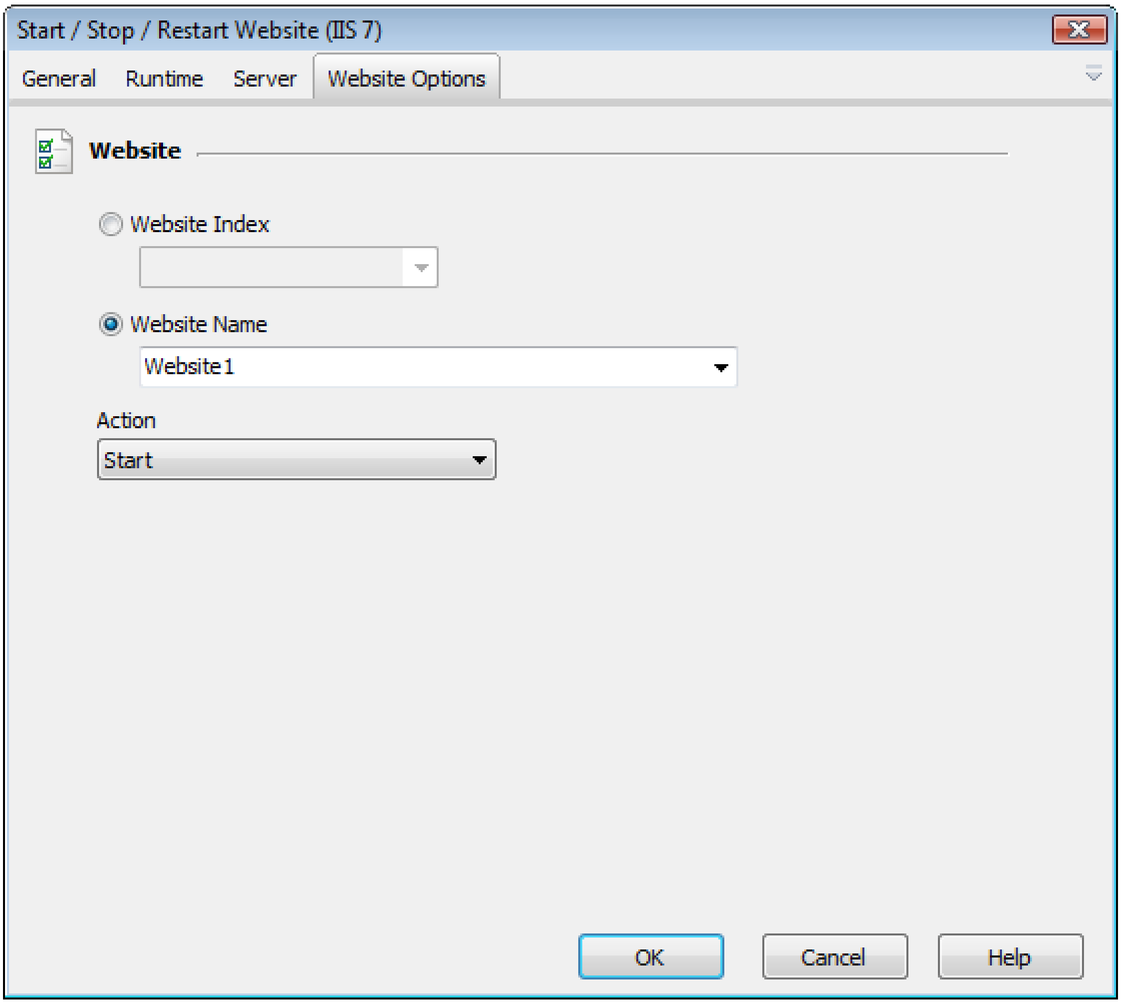
○ Website Index

● Website Name

Website1

6.16.4.5 Start / Stop / Restart Website

The Start / Stop / Restart Website (IIS 7) action enables you to change the current state of an existing website.



○ Website Index

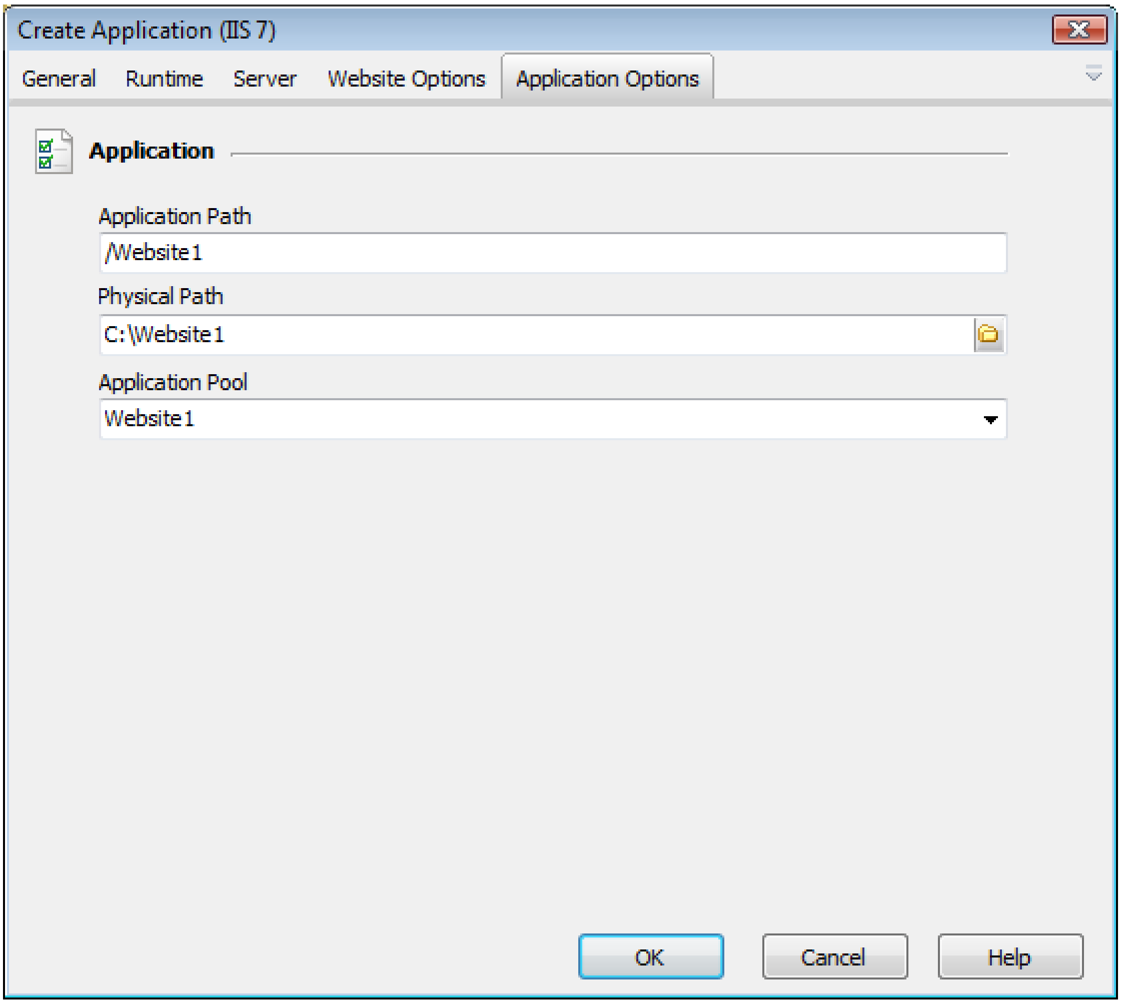
● Website Name: Website1

Action: Start

You can choose to start, stop or restart the web site by selected the relevant action in the drop down list.

6.16.4.6 Create Application

The Create Application (IIS 7) action enables you to create a new application under an existing website.



Application Path: /Website1

Physical Path: C:\Website1

Application Pool: Website1

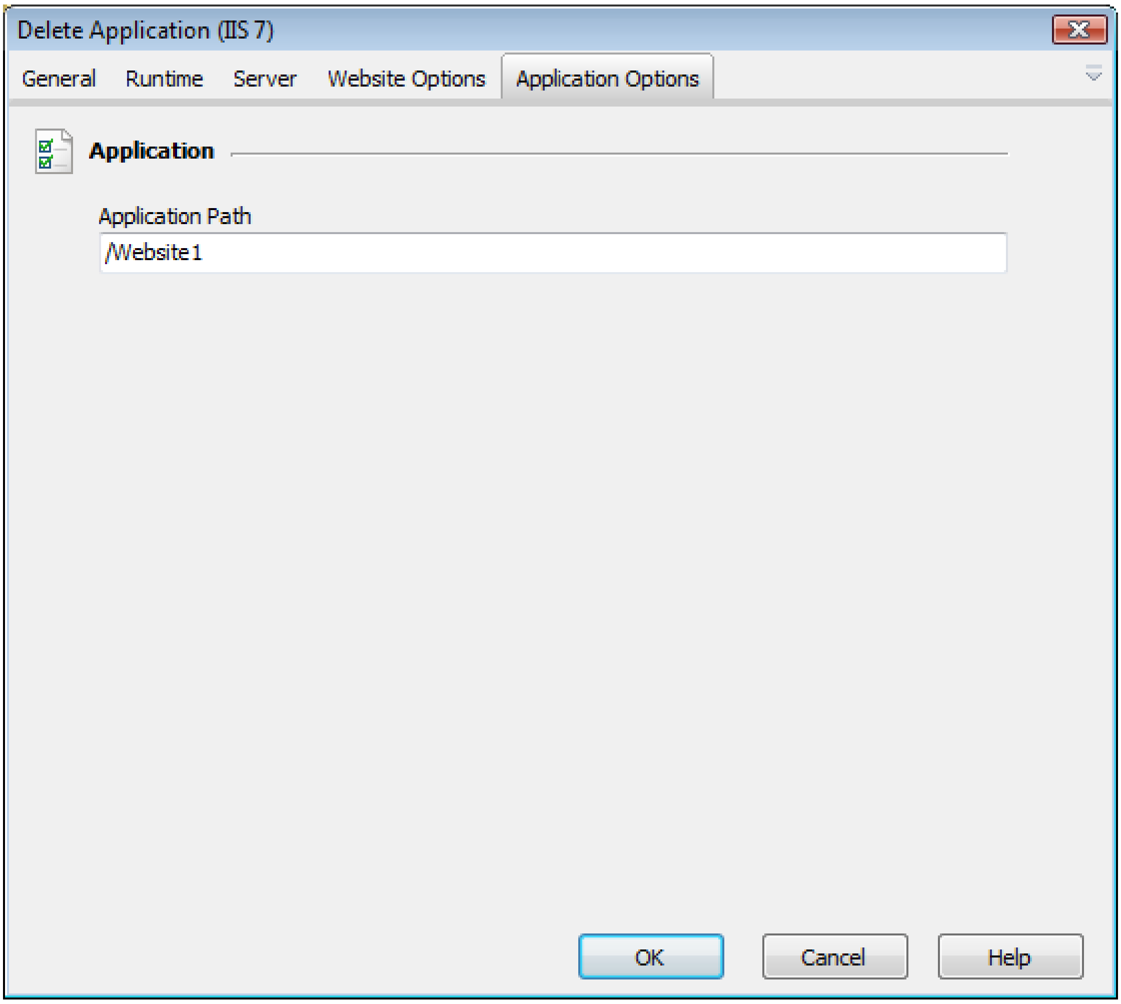
Application Path
The path of the new application.

Physical Path
An optional value that specifies the physical path for the application's root virtual directory.

Application Pool
The application pool which the new application will use, the default is the 'DefaultAppPool'.

6.16.4.7 Delete Application

The Delete Application (IIS 7) action enables you to delete an existing application on a website.

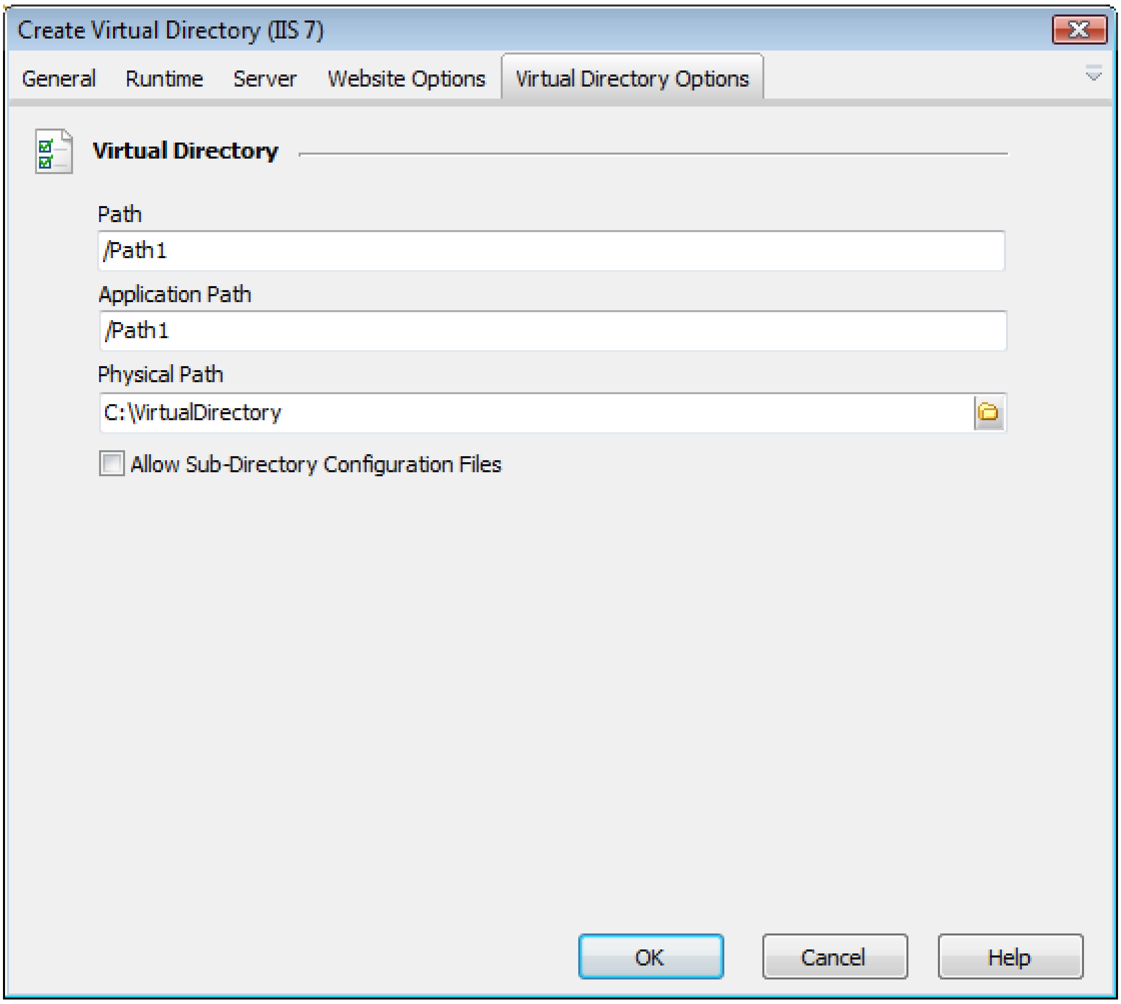


Application Path: /Website1

Application Path
The path to the application that you wish to delete.

6.16.4.8 Create Virtual Directory

The Create Virtual Directory (IIS 7) action enables you to create a new virtual directory on an existing website.



Create Virtual Directory (IIS 7)

Virtual Directory

Path: /Path1

Application Path: /Path1

Physical Path: C:\VirtualDirectory

□ Allow Sub-Directory Configuration Files

Path
The path of the virtual directory.

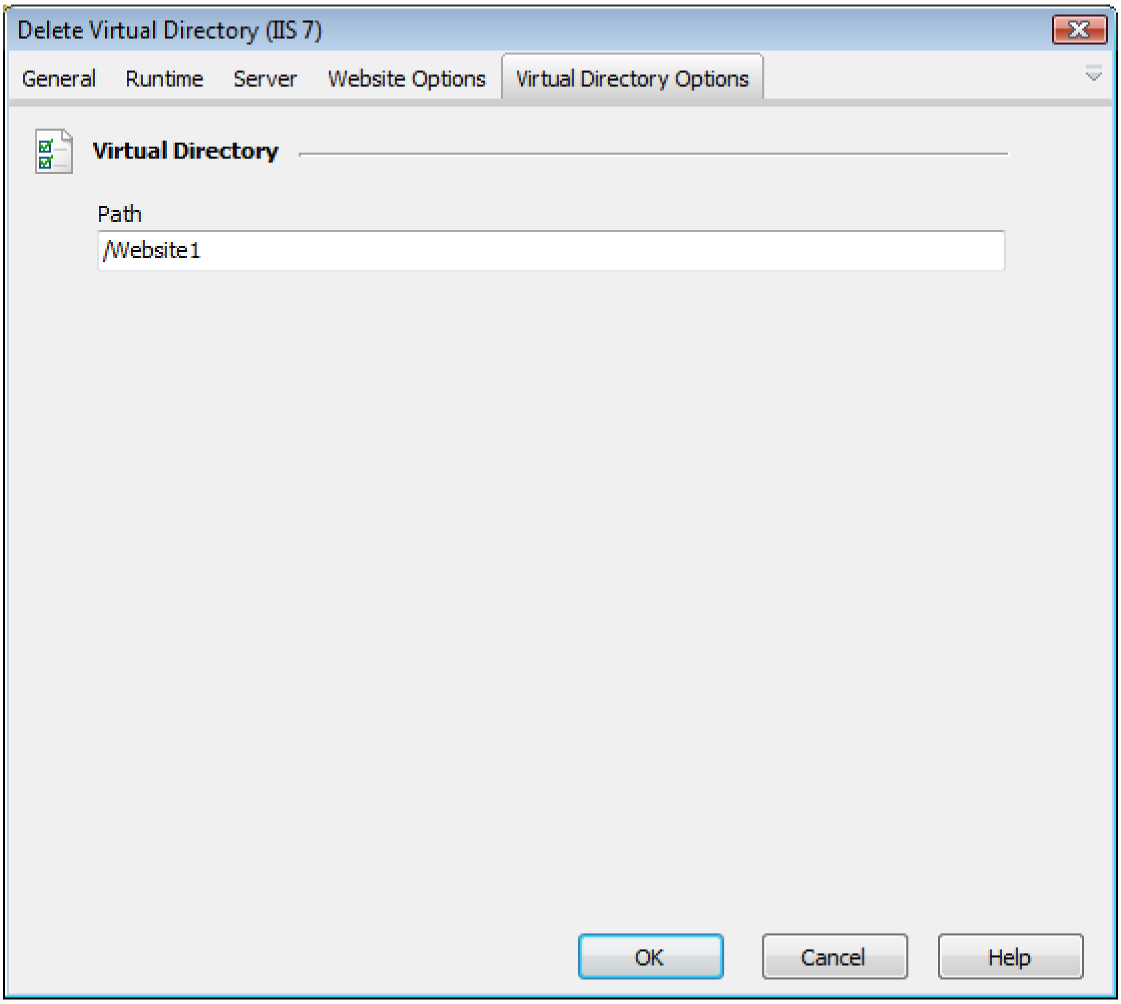
Application Path
The path of the application that the virtual directory will be created in. If the virtual directory is to be created in the root application then use '/'.

Physical Path
The physical path to the file directory where the content for the virtual directory is stored.

Allow Sub-Directory Configuration Files
Determines whether IIS will look for web.config in content directories lower then the current level.

6.16.4.9 Delete Virtual Directory

The Delete Virtual Directory (IIS 7) action enables you to delete an existing virtual directory from a website.

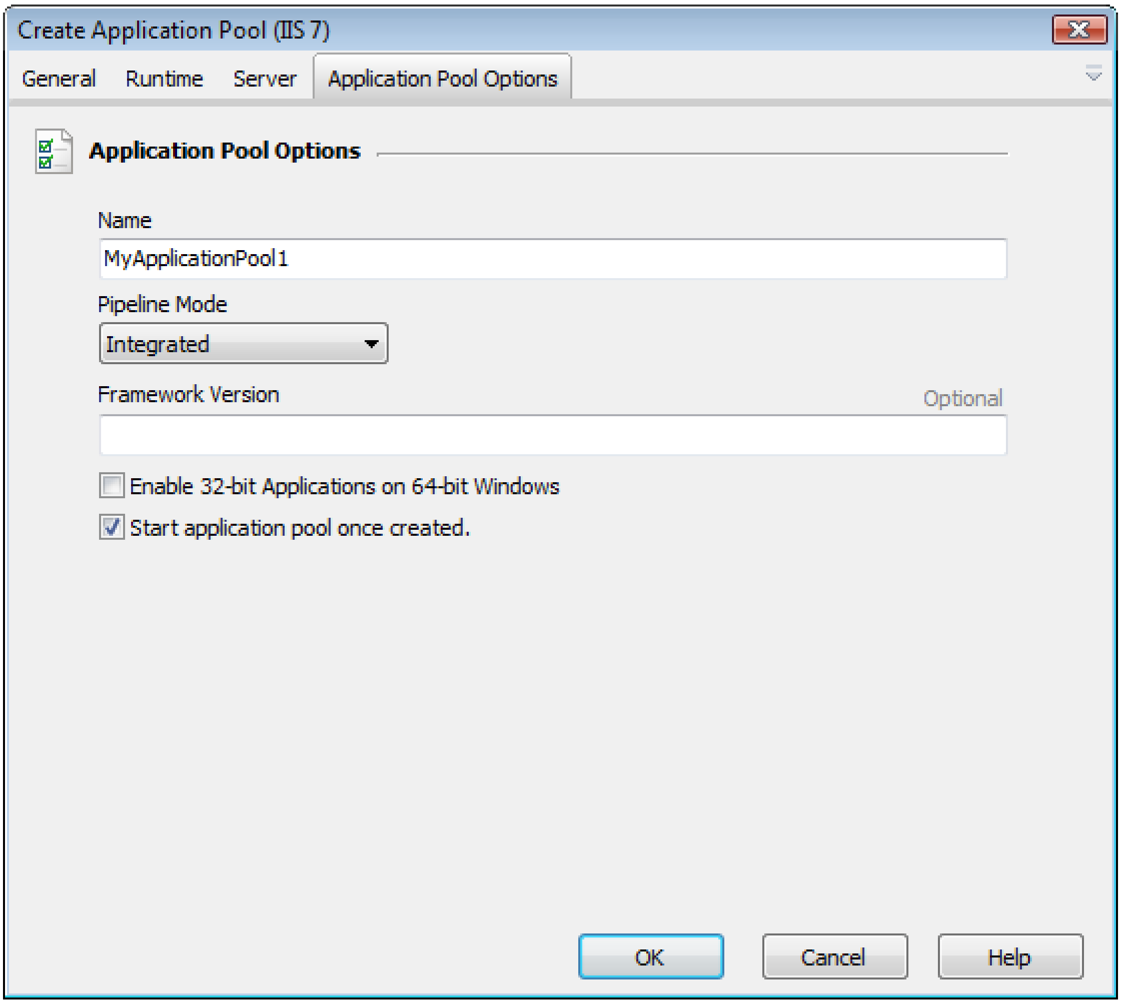


Path
/Website1

Path
The path of the virtual directory that you wish to delete.

6.16.4.10 Create Application Pool

The Create Application Pool (IIS 7) actions enables you to create a new application pool on an IIS server.


Create Application Pool (IIS 7)

Application Pool Options

Name: MyApplicationPool1

Pipeline Mode: Integrated

Framework Version (Optional):

□ Enable 32-bit Applications on 64-bit Windows

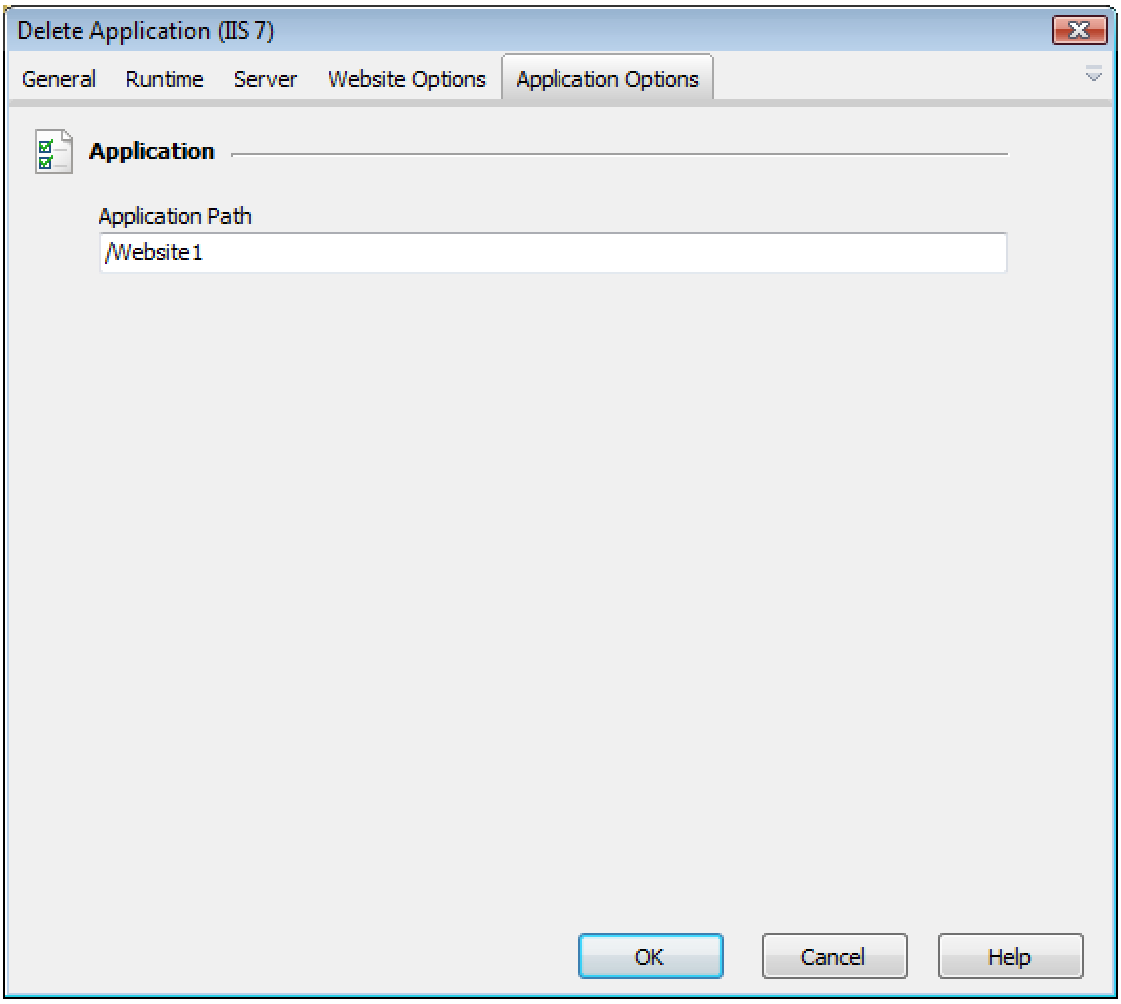
☑ Start application pool once created.

Name
The name that you wish to give the new application pool. This name must be unique on the server.

Start application pool once created
Determines whether the application pool will be started once it has been created.

6.16.4.11 Delete Application Pool

The Delete Application Pool (IIS 7) options enables you to delete an existing application pool on an IIS 7 server.

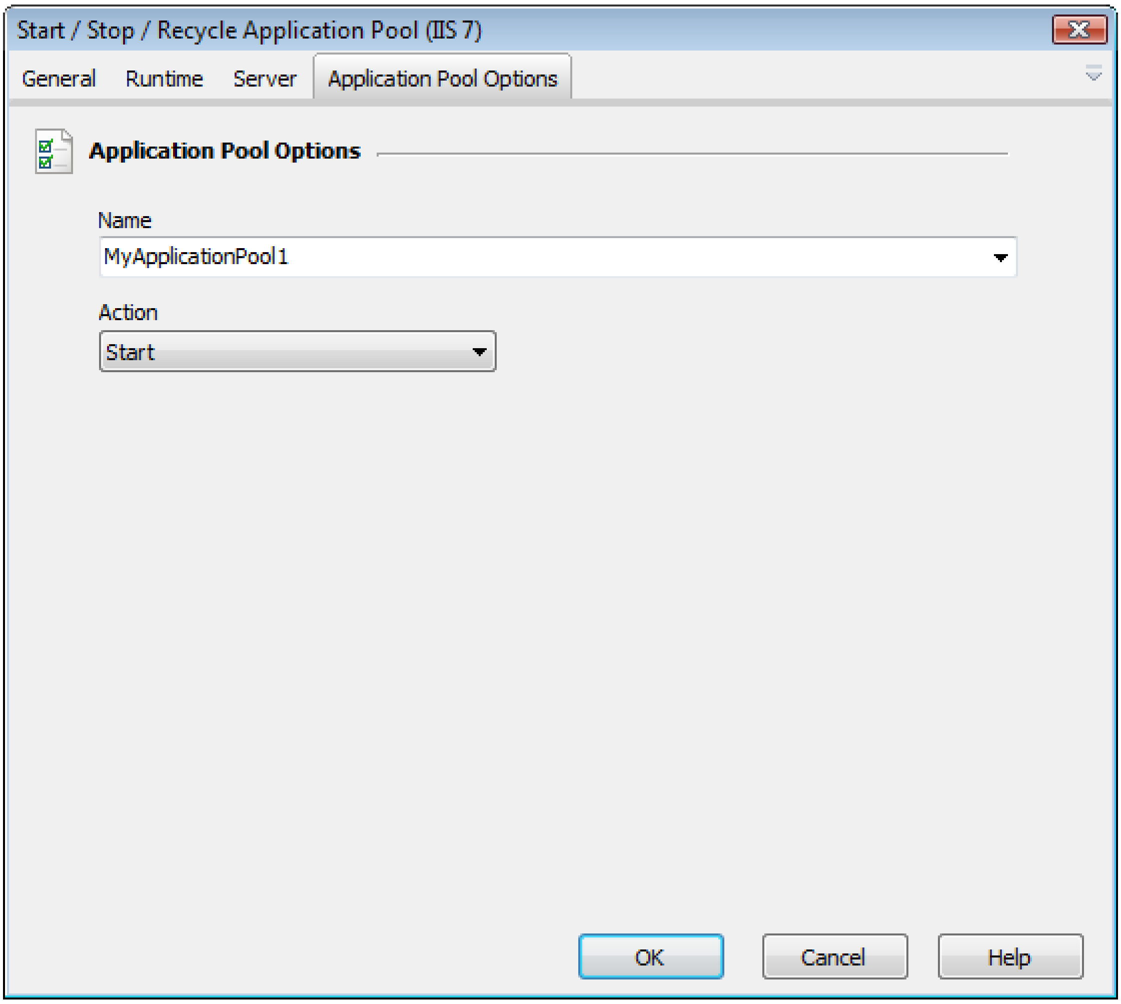


Application Path: /Website1

Name
The name of the existing application pool that is to be deleted.

6.16.4.12 Start / Stop / Recycle Application Pool

The Start / Stop / Recycle Application Pool enables you to start, stop or recycle an existing application pool on an IIS server.



Start / Stop / Recycle Application Pool (IIS 7)

Application Pool Options

Name: MyApplicationPool1

Action: Start

Name
The name of the application pool to start, stop or recycle.

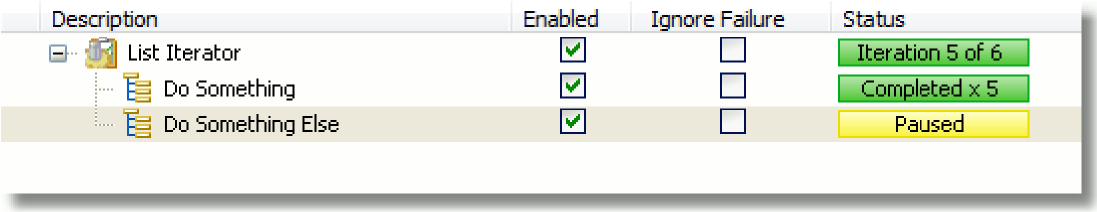
Action
The action to take when the action is executed, possible values are Start, Stop or Recycle.

6.17 Iterators

Iterator actions work by executing their child actions for each item that the iterator finds.

For example, a list iterator might have 3 items: "A, B, and C", and a variable to set : "ListItem".

When the iterator runs, it will run its child actions 3 times - the first time, the variable "ListItem" will have value "A". The second time, it will have value "B" and the last time, value "C".



Description	Enabled	Ignore Failure	Status
List Iterator	☑	□	Iteration 5 of 6
Do Something	☑	□	Completed x 5
Do Something Else	☑	□	Paused

(The list iterator in the above screenshot has been paused during the fifth of six iterations.)

Iterator Actions Summary

Action Name	**Summary**
File Iterator	Iterate through multiple files or a FileSet.
List Iterator	Iterate the elements of a list.
Folder Iterator	Iterate through multiple folders.
File Contents Iterator	Iterate the lines of a text file.
INI File Iterator	Iterate the sections or entries of an INI file.
WMI Process Iterator	Iterate running processes on a Windows computer.
XML Node Iterator	Iterate XML nodes matching an XPath.
ADO Dataset Iterator	Iterate for each row in a dataset returned from a SQL query.
CSV Field Iterator	Iterate through each row of a CSV or other character-delimited file.
Web Service Iterator	Iterate through an array of values returned from a Web Service.

There is also the While Loop action which, while not technically an iterator, can execute its children more than once.

Iterator Script Events

Iterators have the following script events:

Before Action & **After Action** (common to all actions)

These events fire before and after each iteration (including the last iteration.) It is not really recommended that you use these events for Iterator actions.

OnAfterEachIteration

This event fires after all the child actions have run for each iteration. In the example above, OnAfterEachIteration would execute following each invocation of the "Do something else" action group.

This is equivalent to having a Run Script Action as the last child of the iterator.

Note: The best way to run some script before each iteration begins is to add a Run Script Action as the first child of the iterator.

OnFirstRun

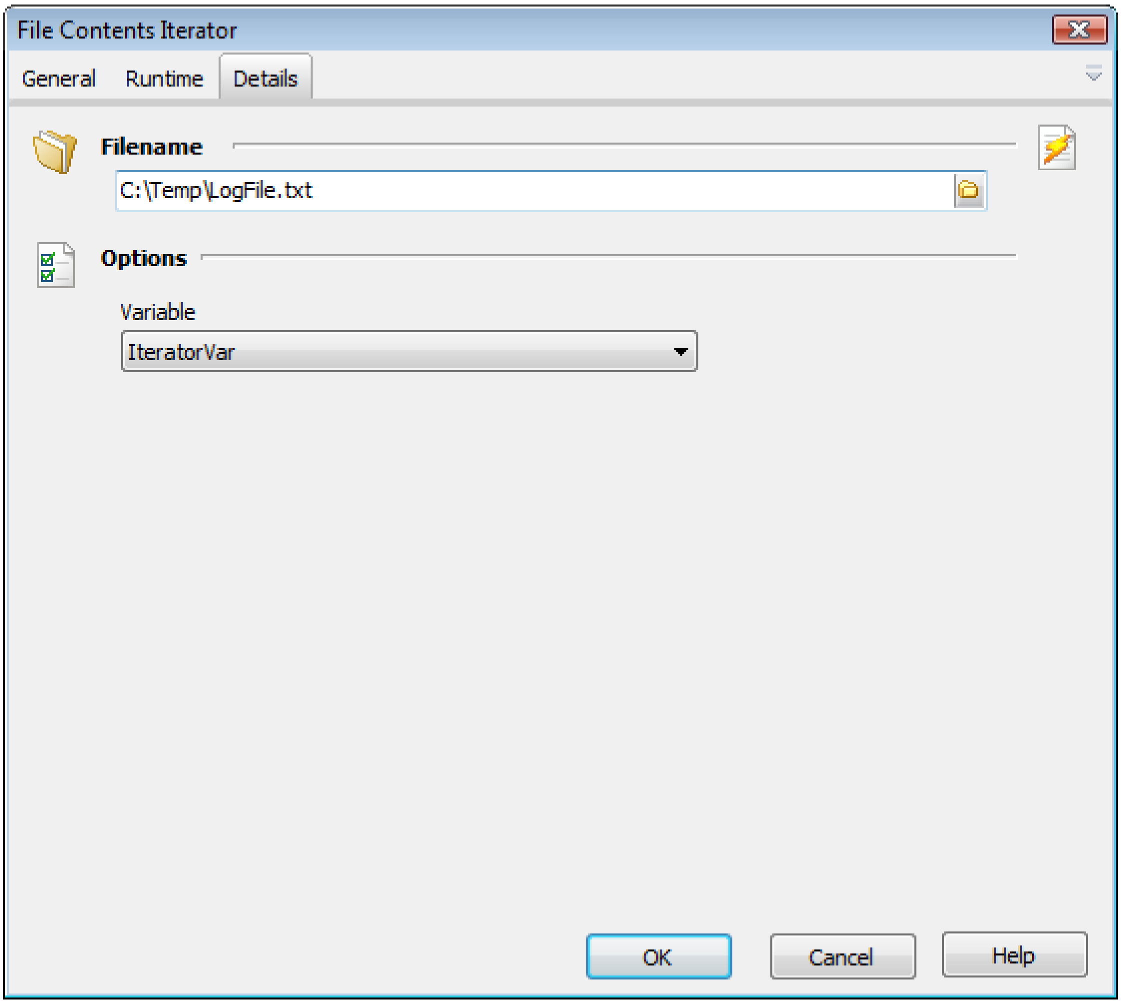
This script event fires the first time the action is run, before the variable has been set to its first value. If the iterator relies on some variables to generate the list of iterated values, you can set them from the OnFirstRun event.

This is equivalent to having a Run Script action placed immediately before the iterator action.

6.17.1 File Contents Iterator

The File Contents Iterator action enables you to iterate over each line of a text a file

For general information on iterators, see the Iterators overview topic. For an iterator which can read each field of a comma-delimited (or other character-delimited) file, see the CSV Field Iteratoraction.



Filename: C:\Temp\LogFile.txt

Variable: IteratorVar

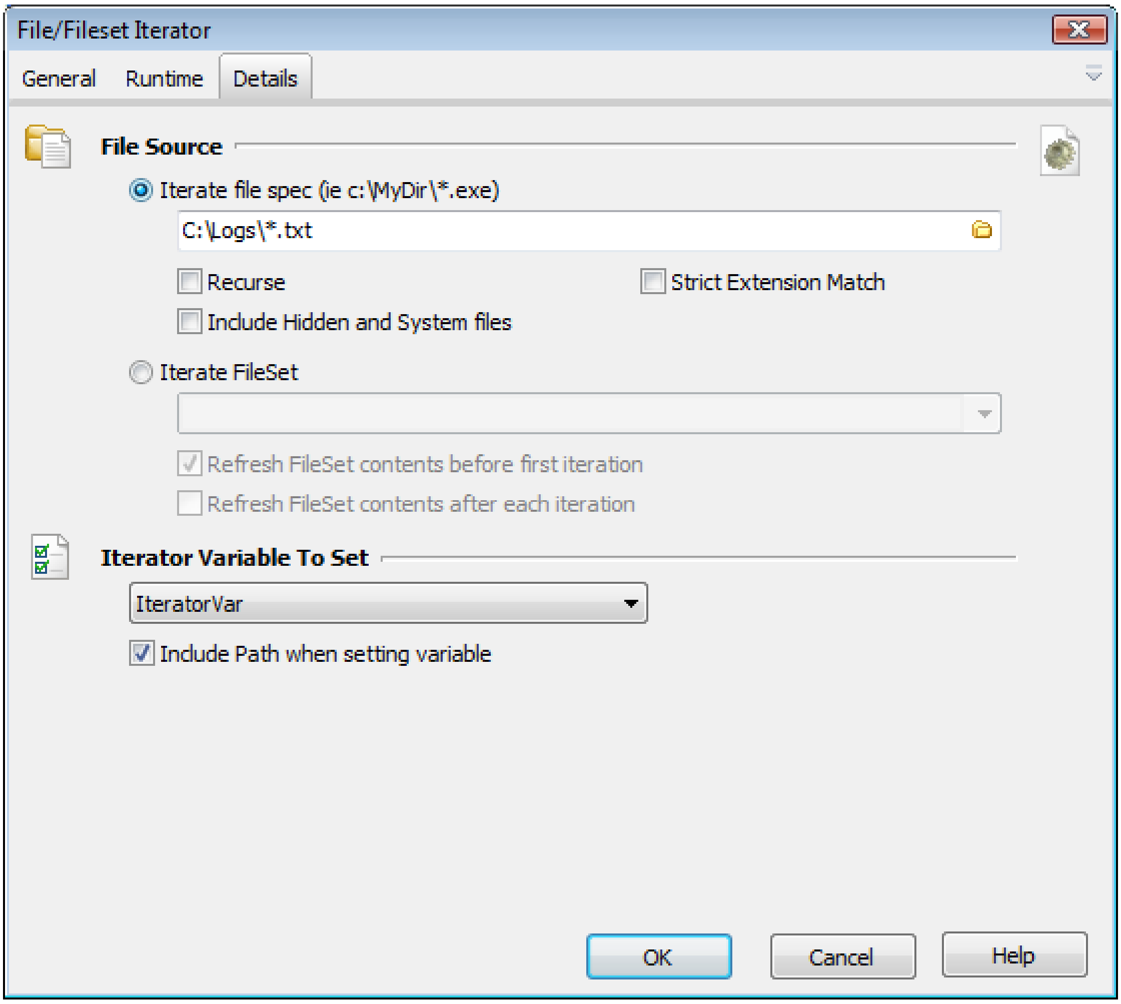
Filename - the name of the text file to read

Variable - the Automise variable to place the current line of the file in for each iteration

6.17.2 File/Fileset Iterator

The File Iterator Action allows you run a set of actions for each file in a list. The file list can be determined by a file spec (ie C:\MyDir\*.obj) or a FileSet (see the FileSets topic).

For general information about iterators, see the iterators overview.



● Iterate file spec (ie c:\MyDir\*.exe)

C:\Logs\*.txt

□ Recurse

□ Strict Extension Match

□ Include Hidden and System files

○ Iterate FileSet

☑ Refresh FileSet contents before first iteration

□ Refresh FileSet contents after each iteration

Iterator Variable To Set

IteratorVar

☑ Include Path when setting variable

File Source

Specify either a file spec (use wildcards like * to select multiple files) or a FileSet (previously defined with the FileSet Define action.) to iterate. Each file in the list will be iterated once.

These options are only available when iterating a filespec:

Recurse - Recurse into subdirectories looking for files which match the filespec.

Include Hidden and System files - Include hidden and system files in the search.

These options are only available when iterating a FileSet:

Refresh FileSet contents before first iteration - If this option is set, the contents of the FileSet will be refreshed before the iterator starts iterating. This is equivalent to placing a FileSet Refresh action as the immediate previous sibling of the File Iterator.

Refresh FileSet contents after each iteration - If this option is set, then the contents of the FileSet will be refreshed after each iteration completes. Any new files which appear when the FileSet is refreshed will be appended to the list of files to iterate. Any files which are no longer part of the refreshed FileSet will be removed from the list (provided they have not yet been iterated) and will not be iterated.

Warning: Refreshing the FileSet contents after each iteration may have unexpected results, if you are not extremely sure what is happening. Use with caution.

Variable To Set

The name of a Automise Project/User Variable. On each iteration, the variable value will be set to the path/name of each file.

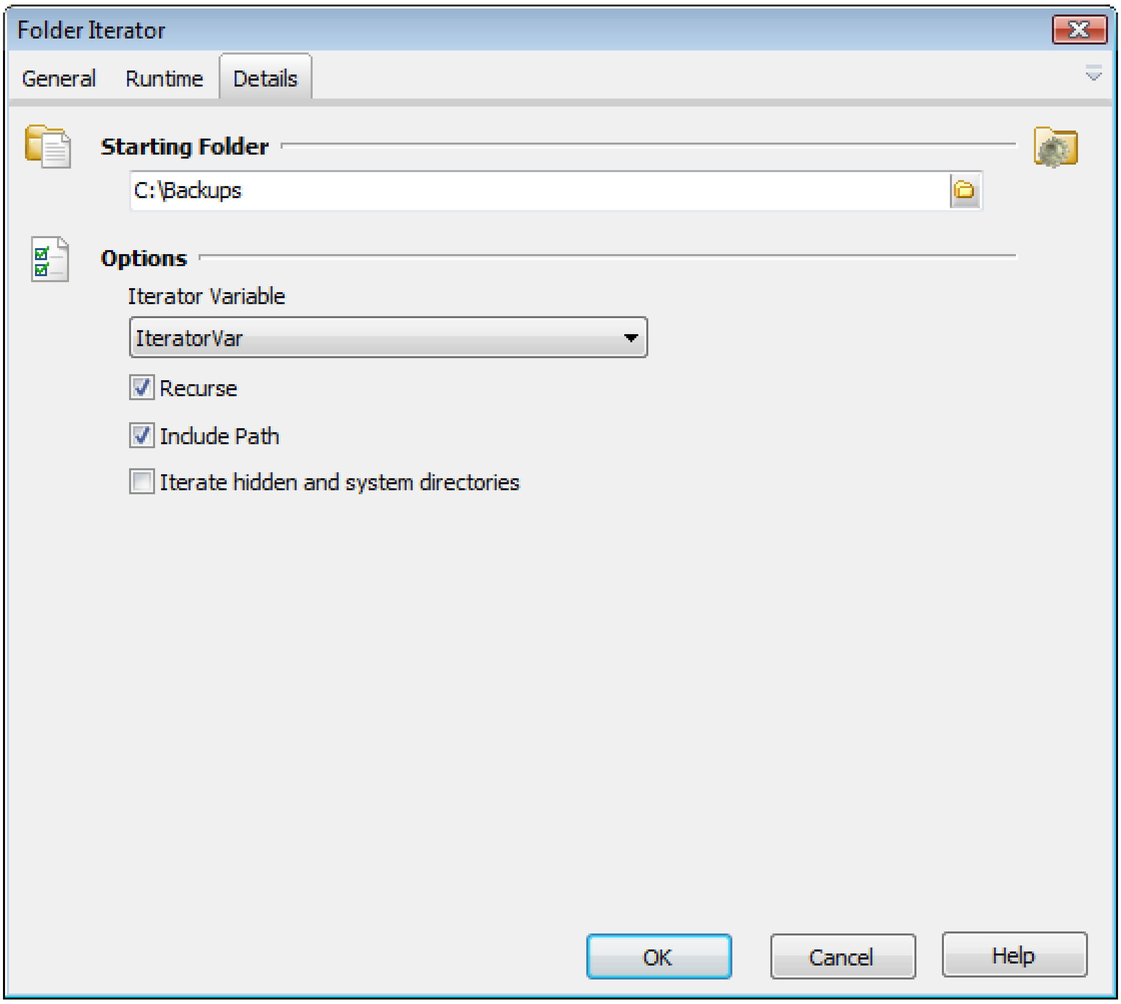
Include Path when setting variable

If this option is set, the variable will be the full path to each file. If it is not set, only the filename (without any directory info) is set.

6.17.3 Folder Iterator

The Folder Iterator action enables you to repeat a set of actions for one or more folders. Specify a starting folder (eg. c:\temp) and the action will iterate over each folder found inside the starting folder.

(For general information on iterators, see the Iterators overview.)



Folder Iterator

Starting Folder

C:\Backups

Options

Iterator Variable

IteratorVar

☑ Recurse

☑ Include Path

□ Iterate hidden and system directories

Starting Folder - Enter the root folder to begin the search from.

Variable - Specify the variable for the iterator to set.

Recurse - Select recurse to recurse into subfolders.

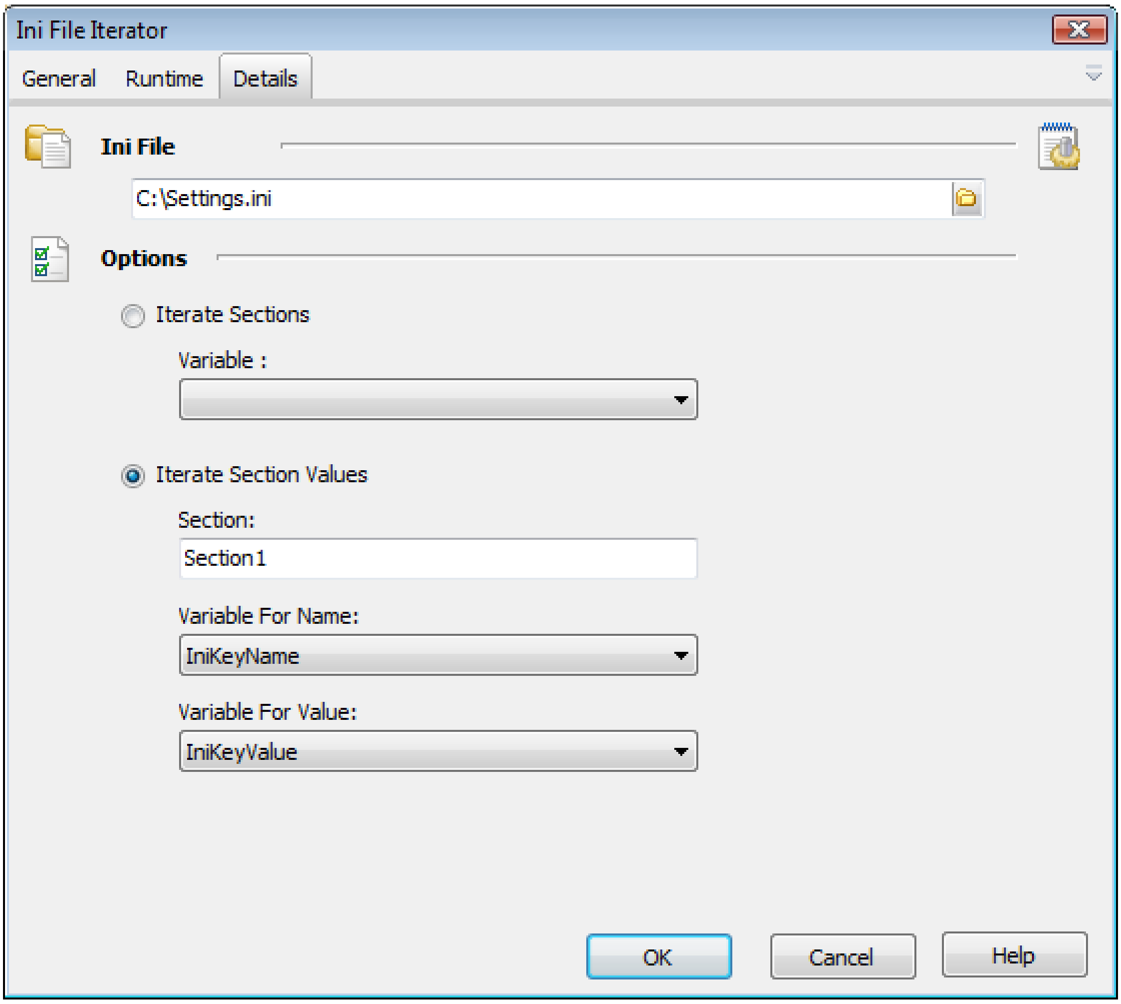
Include Path - If this property is set then the variable will be set to the complete path to the folder. Eg. "c:\temp\myfolder", instead of just the folder name ie "myfolder"

Iterate hidden and system directories - Set this option to also include directories marked with the hidden and/or system attributes.

6.17.4 INI File Iterator

The IniFile Iterator action enables you to repeat a set of steps for each section or each value in a section of an INI file.

The action can work in two modes: Section Iterator, or Section Values Iterator. In Section Iterator mode, the child actions will be called for each section that exists in the specified INI file. In the Section Values mode, the child actions will be called for each Name=Value within the specified section of the INI file.



Ini File

C:\Settings.ini

Options

○ Iterate Sections

Variable :

● Iterate Section Values

Section: Section1

Variable For Name: IniKeyName

Variable For Value: IniKeyValue

Ini File - Specify the INI file to read

Iterate Sections - Iterates for each section found in the INI file.

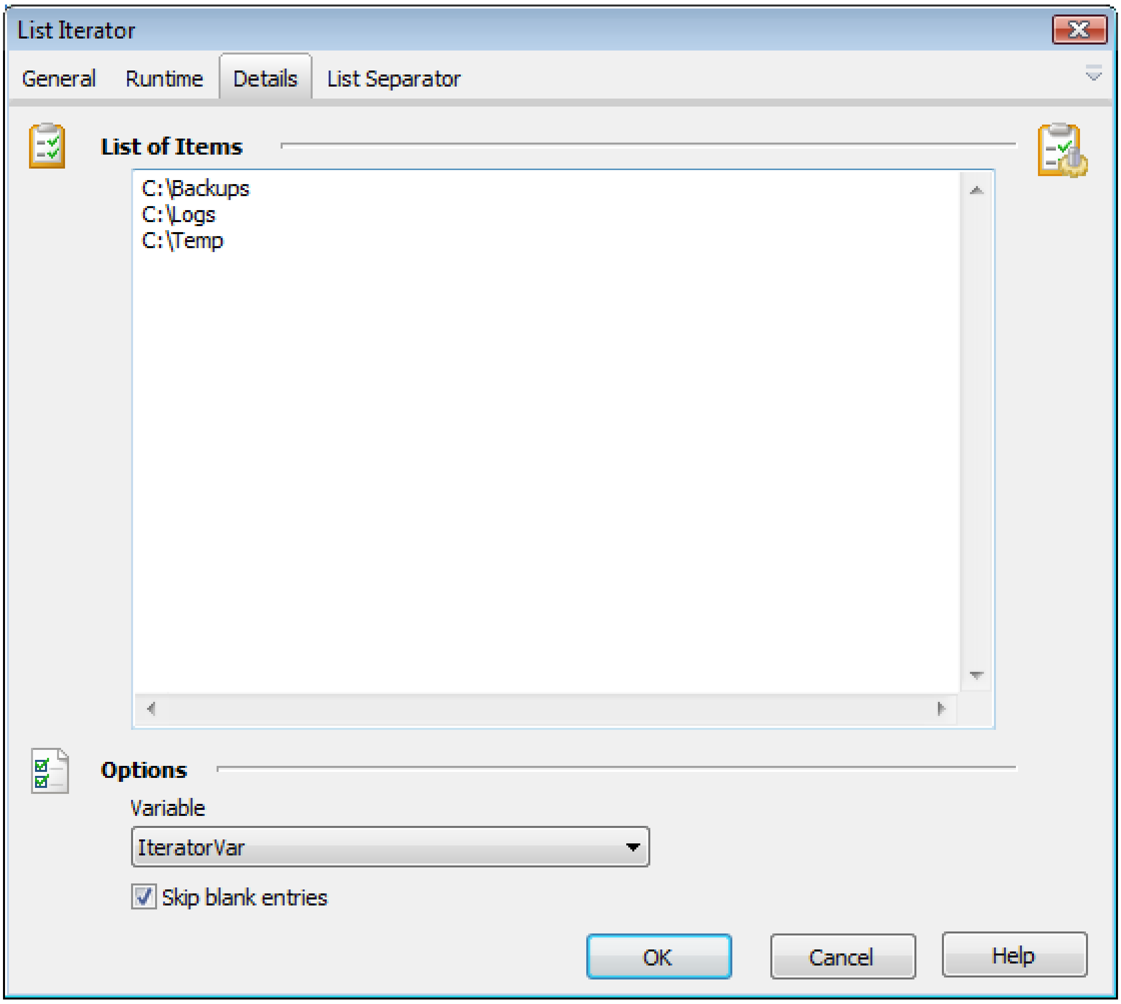
Iterate Section Values - You need to specify which section in the INI file, and then the action will iterate for each Name=Value pair found in the specified section

See also INI file format information

6.17.5 List Iterator

The List Iterator Action allows you to perform a set of actions for each item in a list.

For general information on iterators, see the Iterators overview topic.



List of Items

```
C:\Backups
C:\Logs
C:\Temp
```

Options

Variable: IteratorVar

☑ Skip blank entries

List of Items

This is the list of items to iterate through.

By default, each new line represents a value to iterate. In the above example, the iterator will set the variable User first to Fred (and run it's child actions), then to George (and run it's child actions) and then to Judy (and run it's child actions).

Automise Variables (ie %ListContents%) can be used as part (or all) of the list.

Variable

The name of the Automise Project/User Variable to set for each value.

Skip Blank Entries

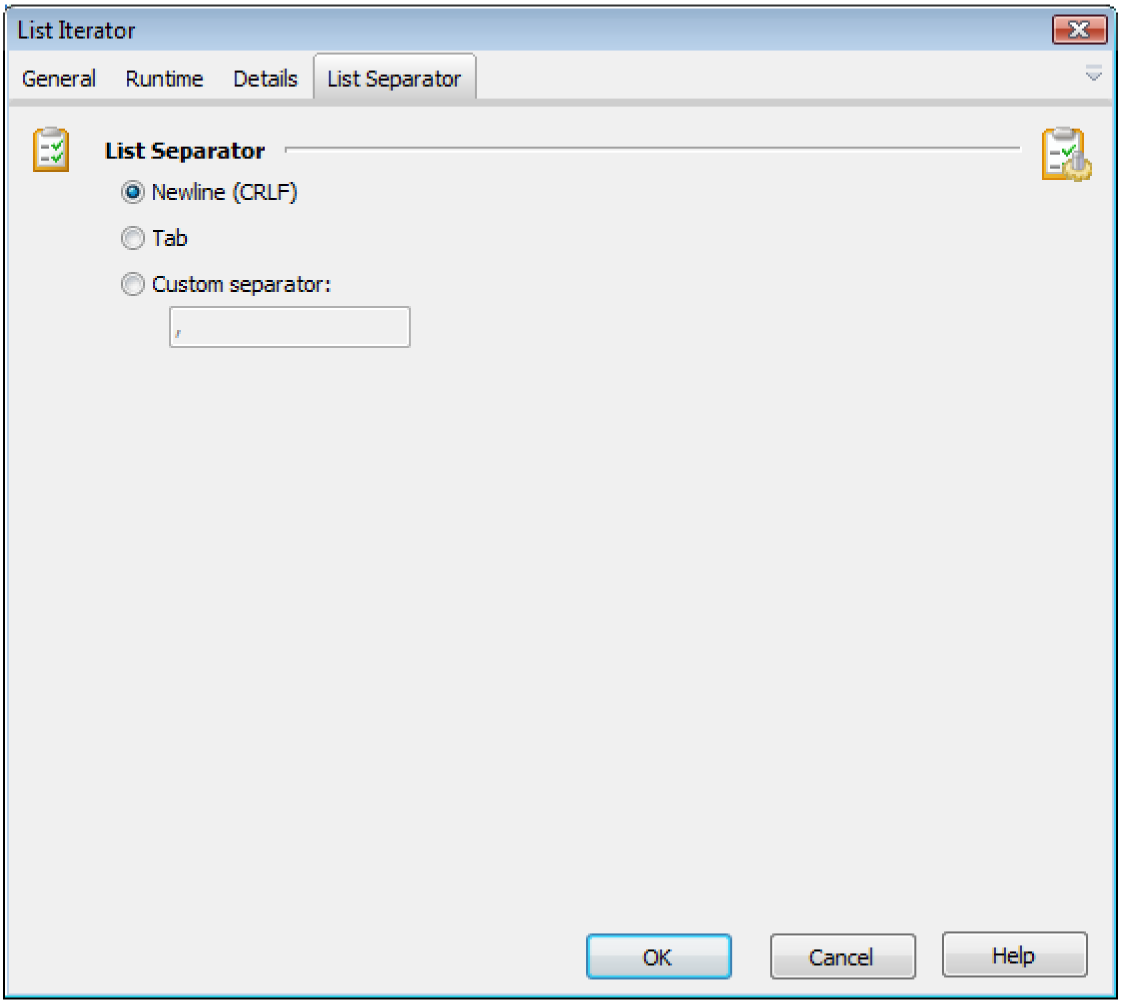
If this option is checked, empty lines will be skipped.

Scripting Info

You can set the list of items in script from the "**OnFirstRun**" event. Example script text:

```
Action.ListOfItems.clear
Action.ListOfItems.Add("Item 1")
Action.ListOfItems.Add("Item 2")
```

List Separator



List Separator

● Newline (CRLF)

○ Tab

○ Custom separator:

,

By default, each item in the list appears on a new line.

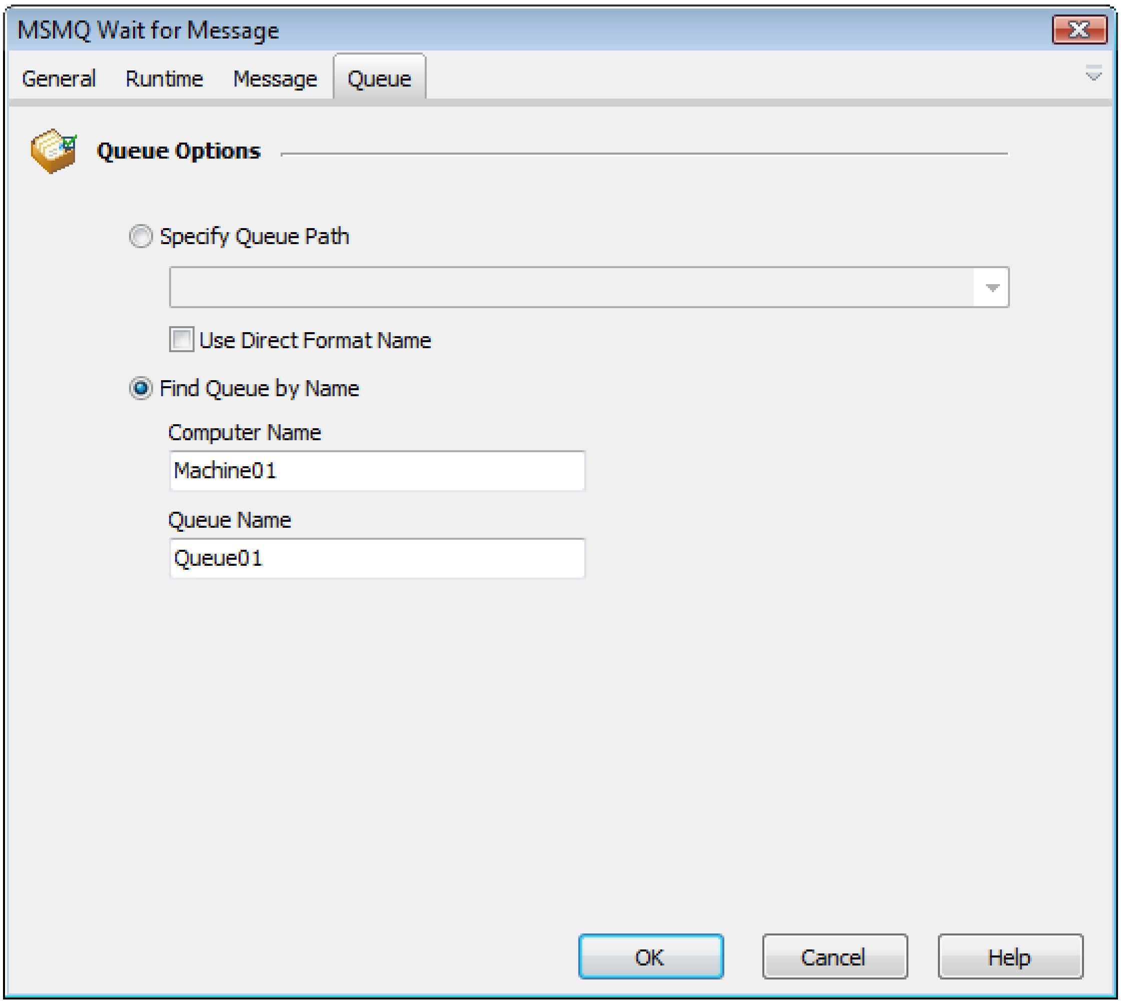
Optionally, you can change this so that items in the list are delimited by tabs, or by a custom separator character or string (ie comma-delimited.)

Note that if you use a list separator other than newline, scripting of the ListOfItems property will not work as expected.

6.18 Microsoft Message Queuing (MSMQ)

The Microsoft Message Queuing actions enables the administration of queues and the sending and receiving of messages.

The actions share a common options page which defines which message queue to use.



○ Specify Queue Path

☐ Use Direct Format Name

● Find Queue by Name

Computer Name: Machine01

Queue Name: Queue01

Specify Queue Path
This list contains all public and private queues found, if the required queue is not listed, it may be specified.

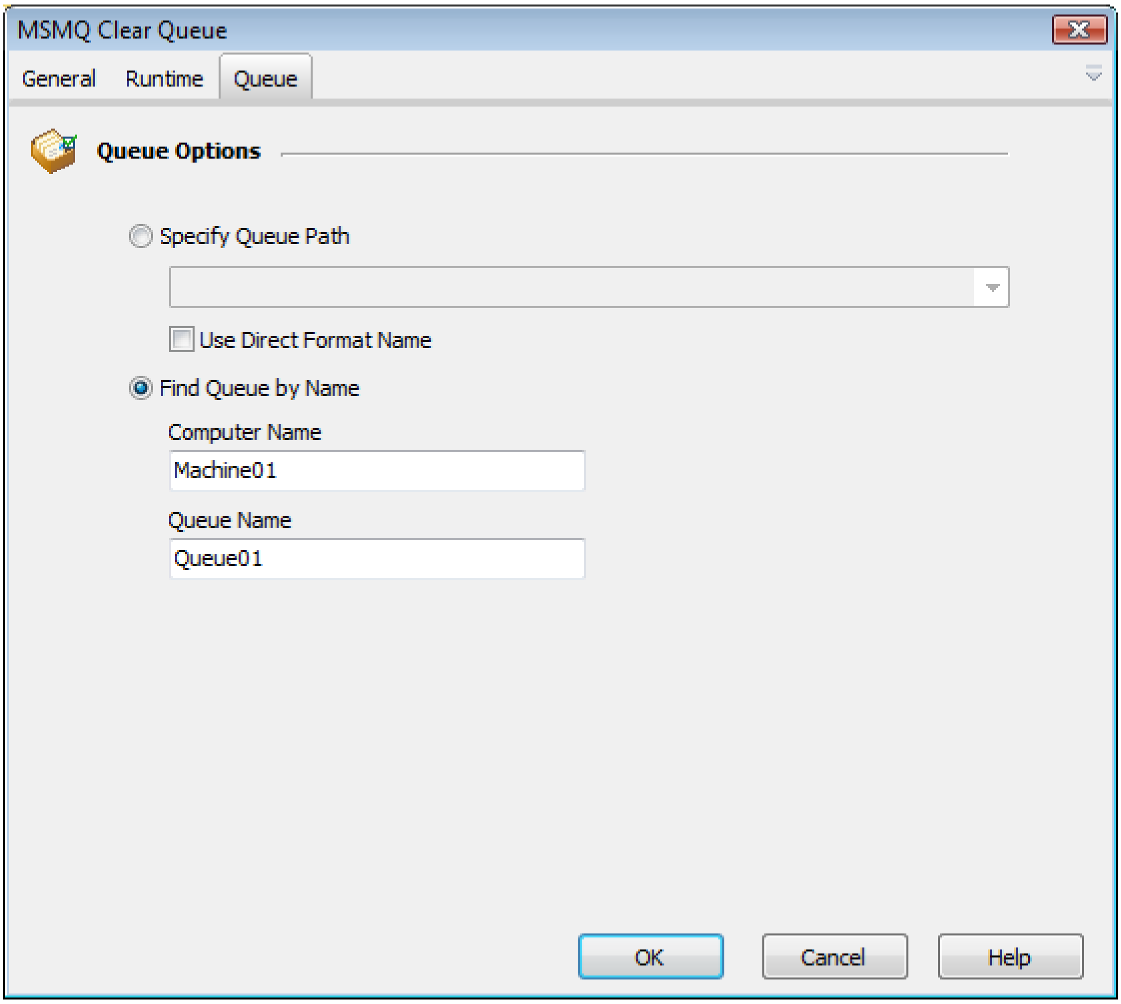
Use Direct Format Name
Used when referencing a queue that are not listed by the directory service, or when the directory service is unavailable. This is typically used when the queue is located on a computer which is not connected to a domain.

Find Queue by Name
Tries to locate the queue by searching the specified computer for the specified queue name.

6.18.1 MSMQ Clear Queue

The MSMQ Clear Queue action will remove all message currently queued in the specified queue.

This action only requires that the common options be specified.



○ Specify Queue Path

□ Use Direct Format Name

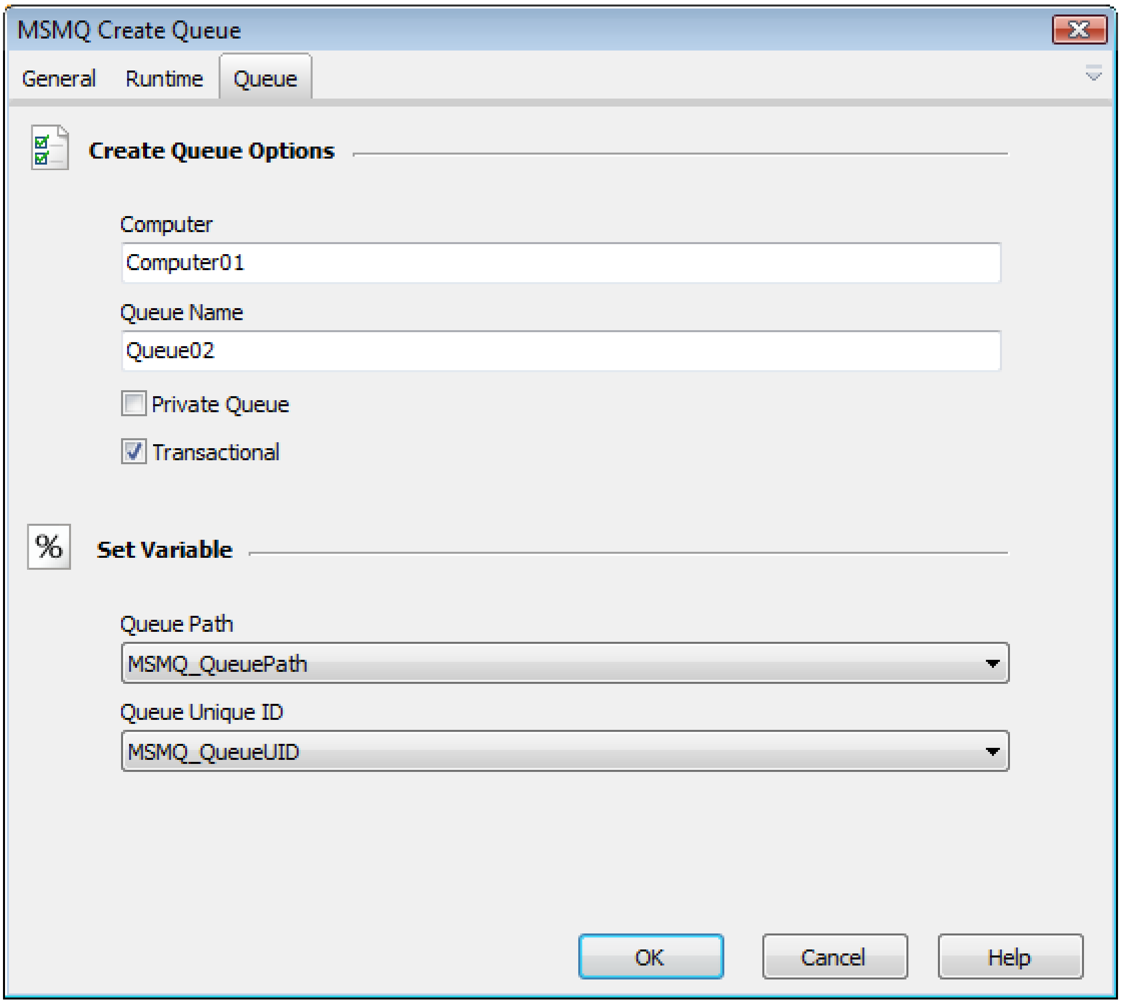
● Find Queue by Name

Computer Name: Machine01

Queue Name: Queue01

6.18.2 MSMQ Create Queue

The MSMQ Create Queue action enables the creation of public or private queues.



MSMQ Create Queue

Create Queue Options

Computer: Computer01

Queue Name: Queue02

□ Private Queue

☑ Transactional

Set Variable

Queue Path: MSMQ_QueuePath

Queue Unique ID: MSMQ_QueueUID

Computer
The name of the computer that the queue will be created on.

Queue Name
The name of the queue.

Private Queue
Forces the queue to be made private, which will prevent it being listed in the directory service.

Transactional
Enables the creation of transactional queues.

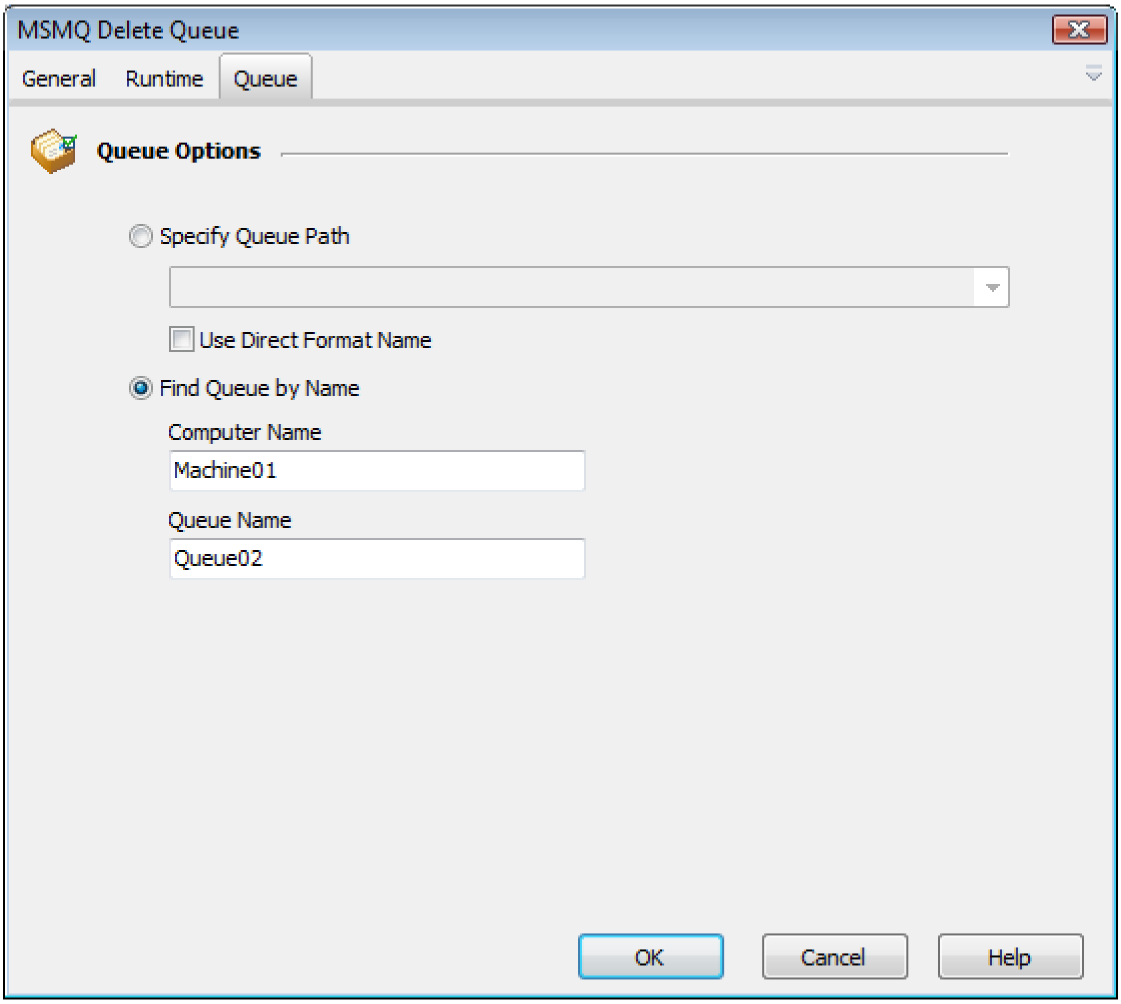
Set Variable
Queue Path
Sets the specified variable to the path of the new queue.

Queue Unique ID
Sets the specified variable to the unique ID of the new queue.

6.18.3 MSMQ Delete Queue

The MSMQ Delete Queue action allows the removal of message queues.

This action only requires that the common options be specified.



○ Specify Queue Path

□ Use Direct Format Name

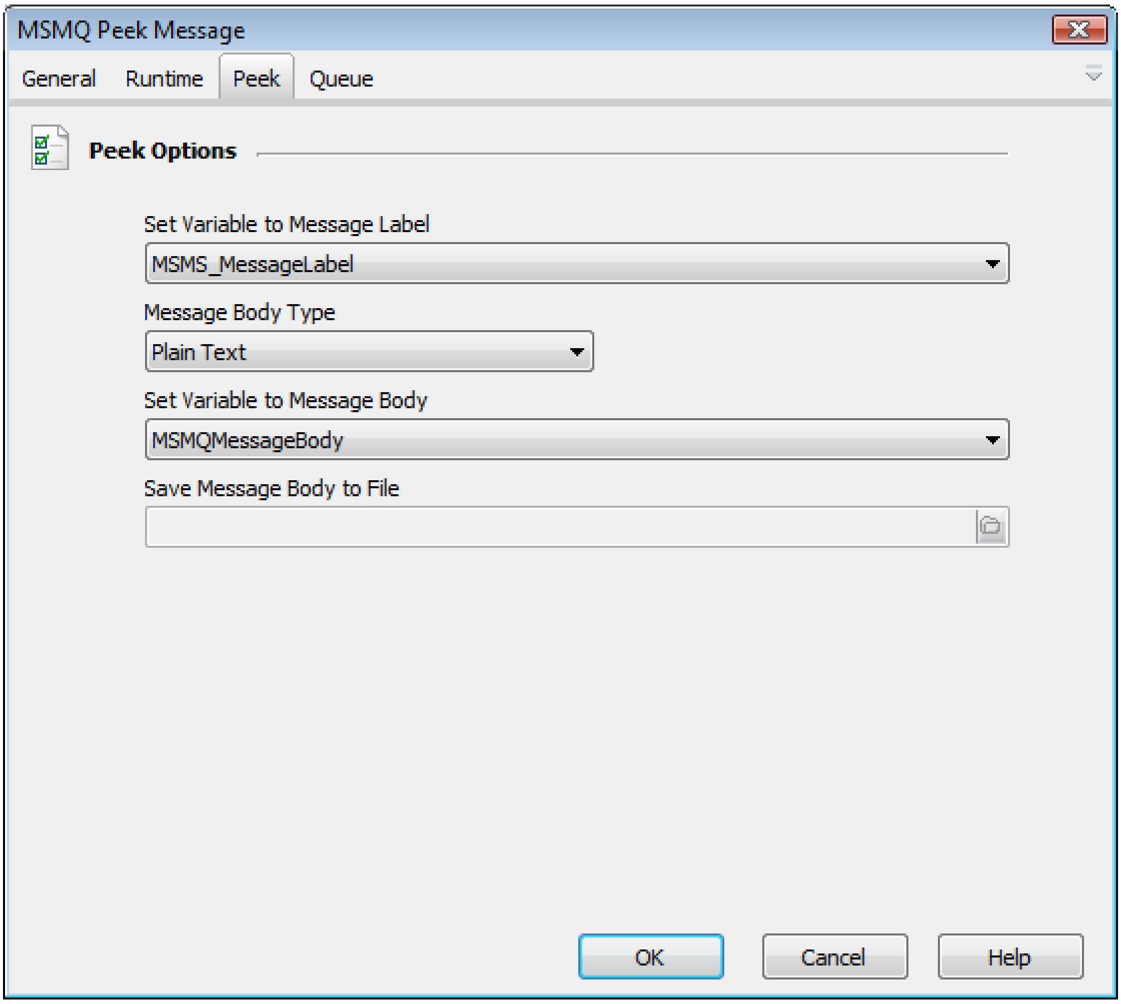
● Find Queue by Name

Computer Name: Machine01

Queue Name: Queue02

6.18.4 MSMQ Peek Message

The MSMQ Peek Message action will retrieve the top message from the queue, without removing it.



MSMQ Peek Message

Peek Options

Set Variable to Message Label: MSMS_MessageLabel

Message Body Type: Plain Text

Set Variable to Message Body: MSMQMessageBody

Save Message Body to File:

Set Variable to Message Label
Sets the specified variable to the message label.

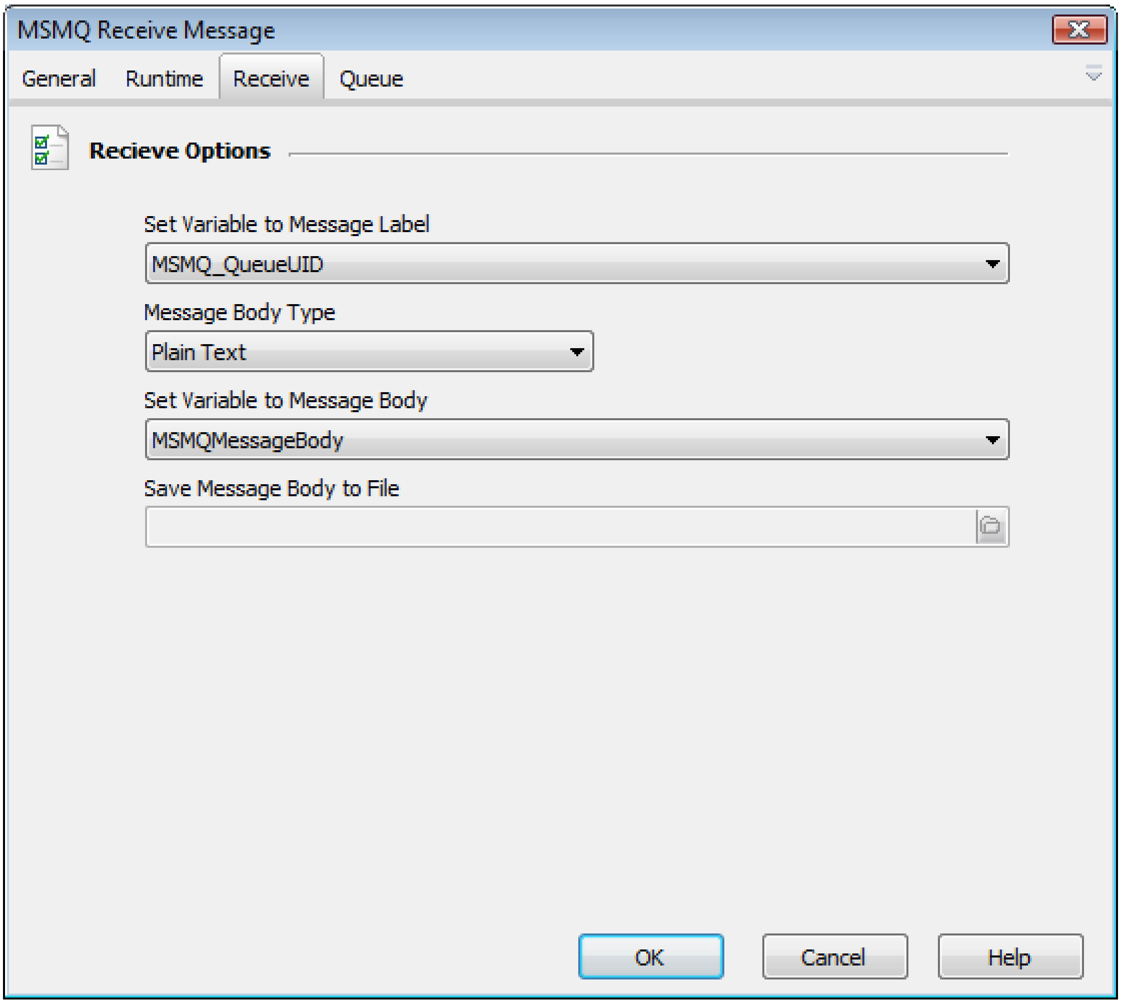
Message Body Type
Specifies the type of message that is expected.

Set Variable to Message Body
Sets the specified variable to the message body.

Save Message Body to File
Saves the binary file received in the message to the specified file.

6.18.5 MSMQ Receive Message

The MSMQ Retrieve Message action will retrieve the top message from queue. The message will be removed from the queue once it has been retrieved.



Set Variable to Message Label: MSMQ_QueueUID

Message Body Type: Plain Text

Set Variable to Message Body: MSMQMessageBody

Save Message Body to File:

Set Variable to Message Label
Sets the specified variable to the message label.

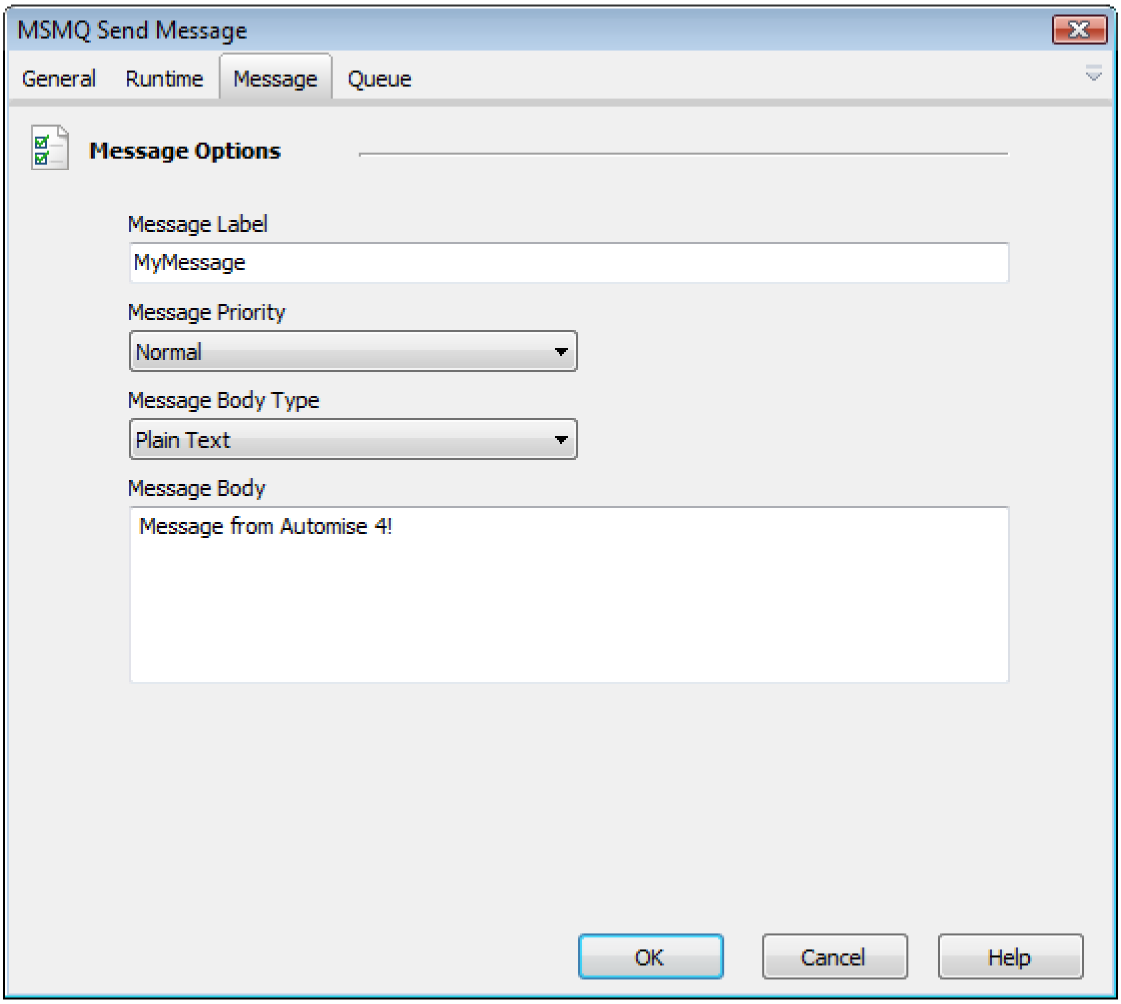
Message Body Type
Specifies the type of message that is expected.

Set Variable to Message Body
Sets the specified variable to the message body.

Save Message Body to File
Saves the binary file received in the message to the specified file.

6.18.6 MSMQ Send Message

The MSMQ Send Message action allows the sending of text or binary messages to a specified queue.



Message Label: MyMessage

Message Priority: Normal

Message Body Type: Plain Text

Message Body: Message from Automise 4!

Message Label
The label of the message to be sent. The label can be used to identify the message in the Wait for Message action.

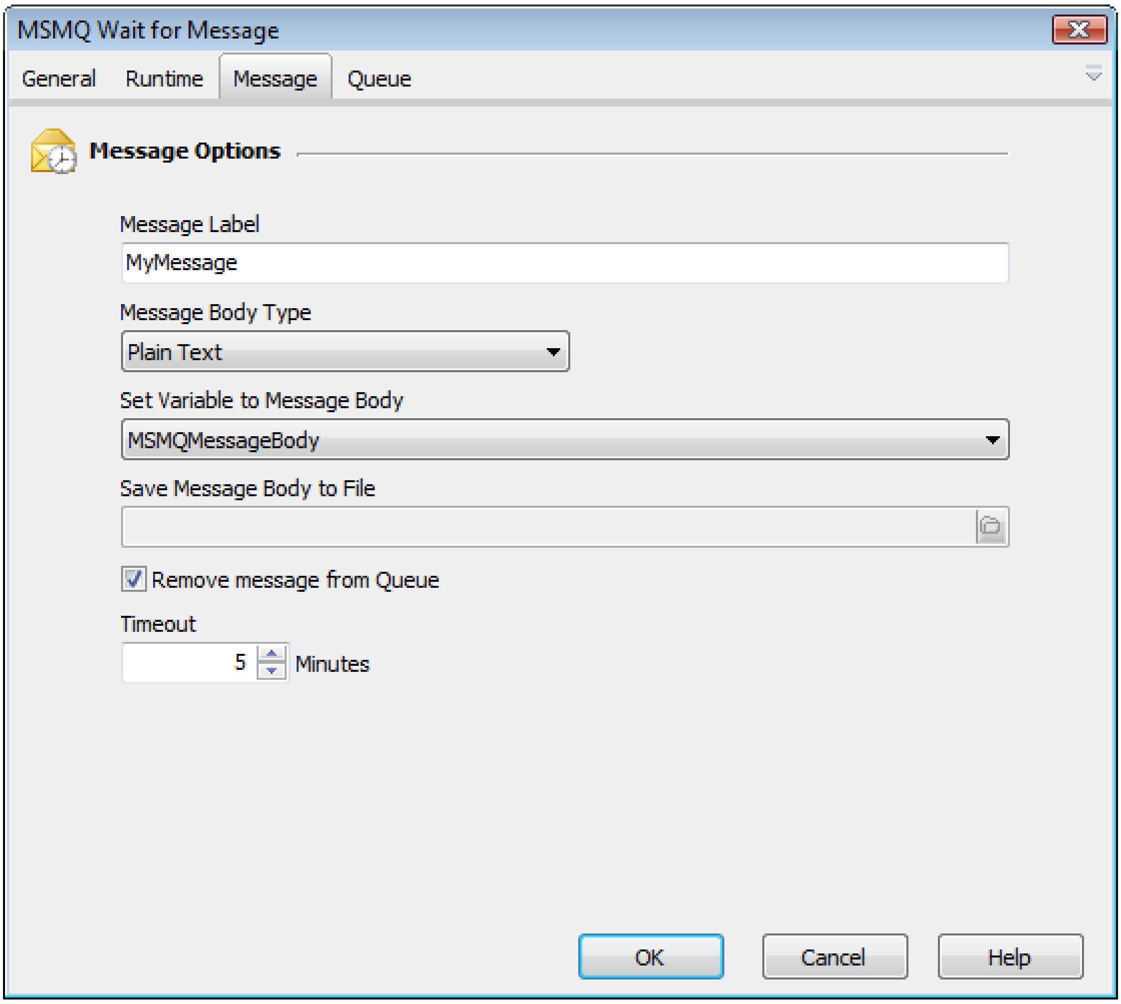
Message Priority
Enables the adjustment of the priority of the message. A higher priority message be closer to the top of the queue then a lower priority message.

Message Body Type
The type of message to be sent. Possible options are 'Plain Text' and 'Binary File'.

Message Body
If the message type is text, then the body of the message can be specified. If the message type is a binary file, the location of the file can be specified.

6.18.7 MSMQ Wait for Message

The MSMQ Wait for Message action will wait for a message to be sent to the specified queue matching the given label.



Message Label
MyMessage

Message Body Type
Plain Text

Set Variable to Message Body
MSMQMessageBody

Save Message Body to File

☑ Remove message from Queue

Timeout
5 Minutes

Message Label
The label of the message to wait for.

Message Body Type
The expected type that the message will be, possible options are 'Plain Text' or 'Binary File'.

Set Variable to Message Body
When the type is set to plain text, the body can be written to a variable for later use.

Save Message Body to File
When the type is set to binary file, the body can be saved to a file for later use.

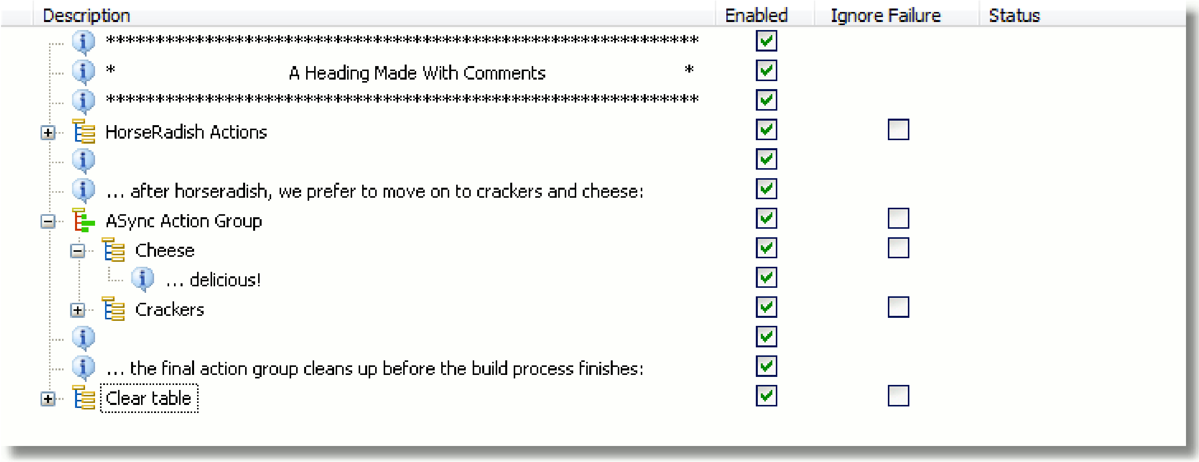
Remove Message from Queue
Once the message has been received the message will be removed from the queue.

Timeout
The length in minutes the action will wait for the message to be sent.

6.19 Misc Actions

6.19.1 Comment Action

Comment Actions provide a way to place descriptive text inside action lists.



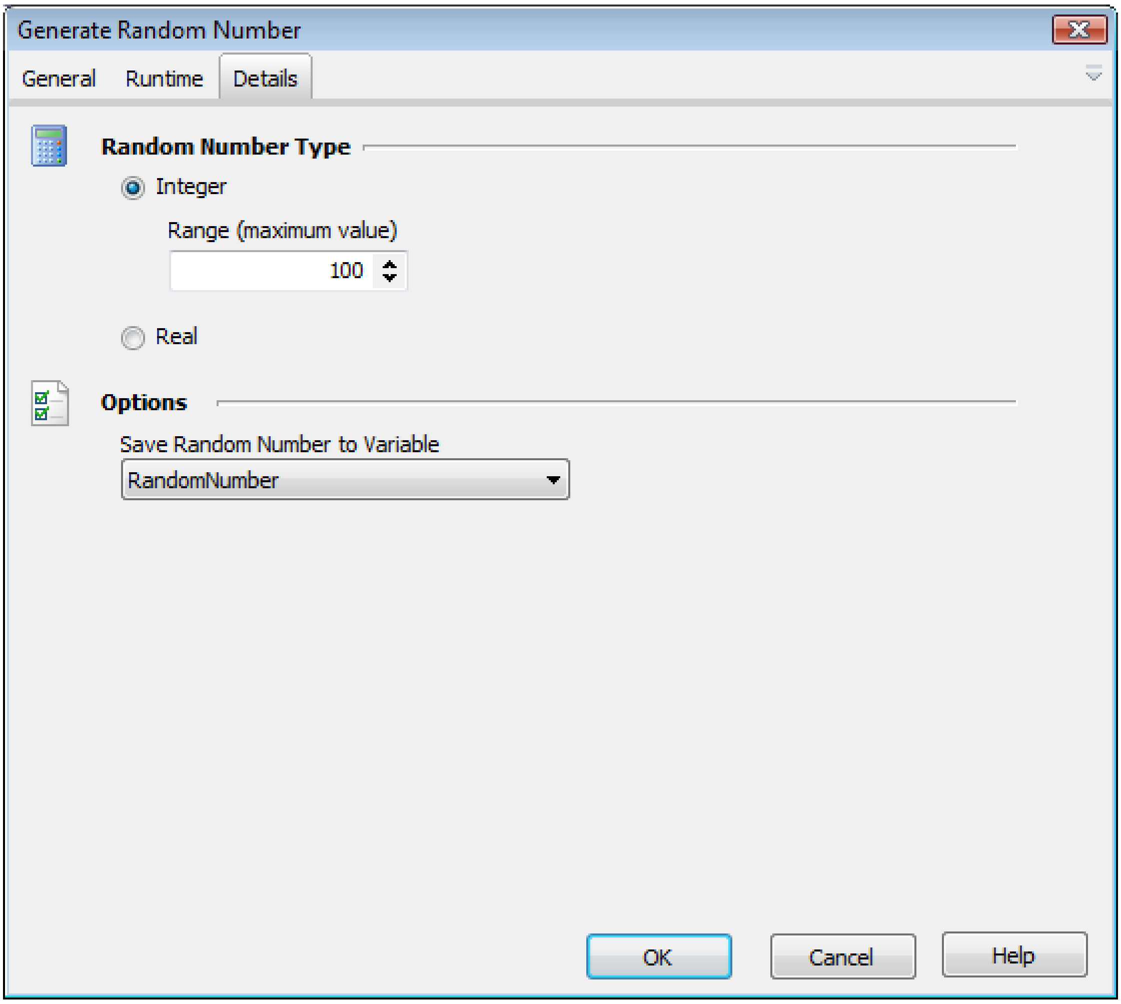
Description	Enabled	Ignore Failure	Status
**	☑		
* A Heading Made With Comments *	☑		
**	☑		
HorseRadish Actions	☑	□	
	☑		
... after horseradish, we prefer to move on to crackers and cheese:	☑		
ASync Action Group	☑	□	
Cheese	☑	□	
... delicious!	☑		
Crackers	☑	□	
	☑		
... the final action group cleans up before the build process finishes:	☑		
Clear table	☑	□	

Comments do not get executed at run time and do not have script events.

Action Groups can also be used to make your build easy to understand.

6.19.2 Generate Random Number

The Generate Random Number action allows you to generate a random number. The random number can either be a integer in the specified range, or a floating point number.



Random Number Type

● Integer

Range (maximum value)

100

○ Real

Options

Save Random Number to Variable

RandomNumber

Random Number Type

Integer/Range
This option will generate a whole number between 0 and the maximum specified.

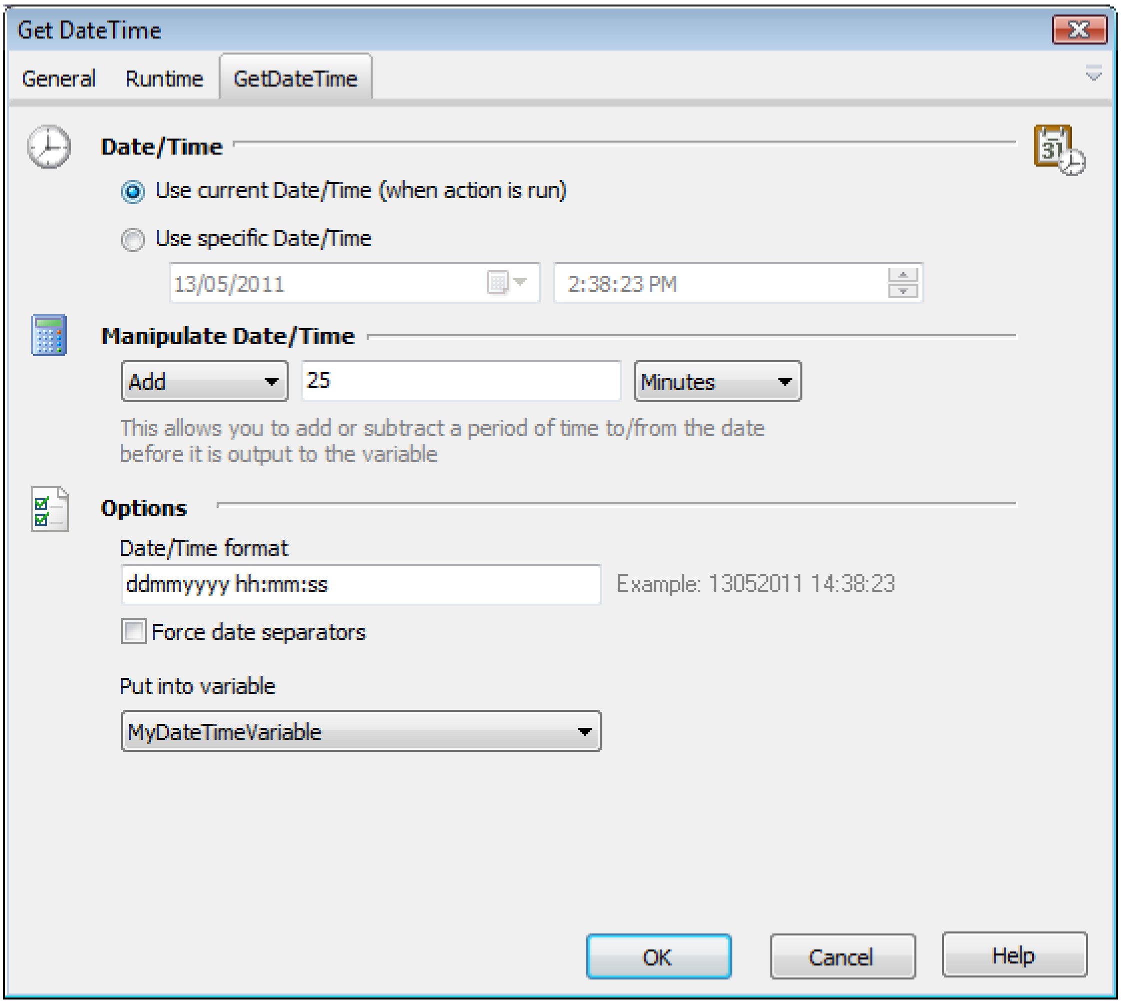
Real
This option generates a floating point number between 0 and 1

Options

Variable
Specify the variable to save the random number to

6.19.3 Get DateTime Action

This action formats a specific Date or Date/Time value into a Automise Variable.



● Use current Date/Time (when action is run)

○ Use specific Date/Time

13/05/2011 2:38:23 PM

Add 25 Minutes

This allows you to add or subtract a period of time to/from the date before it is output to the variable

Date/Time format

ddmmyyyy hh:mm:ss

Example: 13052011 14:38:23

□ Force date separators

Put into variable

MyDateTimeVariable

Date/Time - select if you want to use the current date/time or a specific date/time

Manipulate Date/Time - use this option to either add or subtract a time period to/from the date/time. You need to choose either "Add" or "Subtract", enter an amount (you may use variables in this field), and then the units: Seconds, Minutes, Hours, Days, Weeks, Months, or Years. For example, if you wanted to subtract 5 days from today's date, simply choose "Subtract", enter 5 in the middle field, and then select "Days". If the amount field isn't an integer (or the variable used doesn't return an integer), then the action will fail. Leave the value field blank to use the date/time as-is.

Date/Time format - enter the desired format for the date/time. See below for a detailed listing of all possibilities.

Force Date Separators - option forces the format string to use literal '/' and ':' instead of the separators specified in the Regional Settings Control Panel.

Put into variable - the date/time (after any manipulation) will be output to this variable in the desired format

Valid date format specifiers for the Format property : DateTime Format Strings

6.19.4 Image File Manipulation

6.19.4.1 Image Manipulation

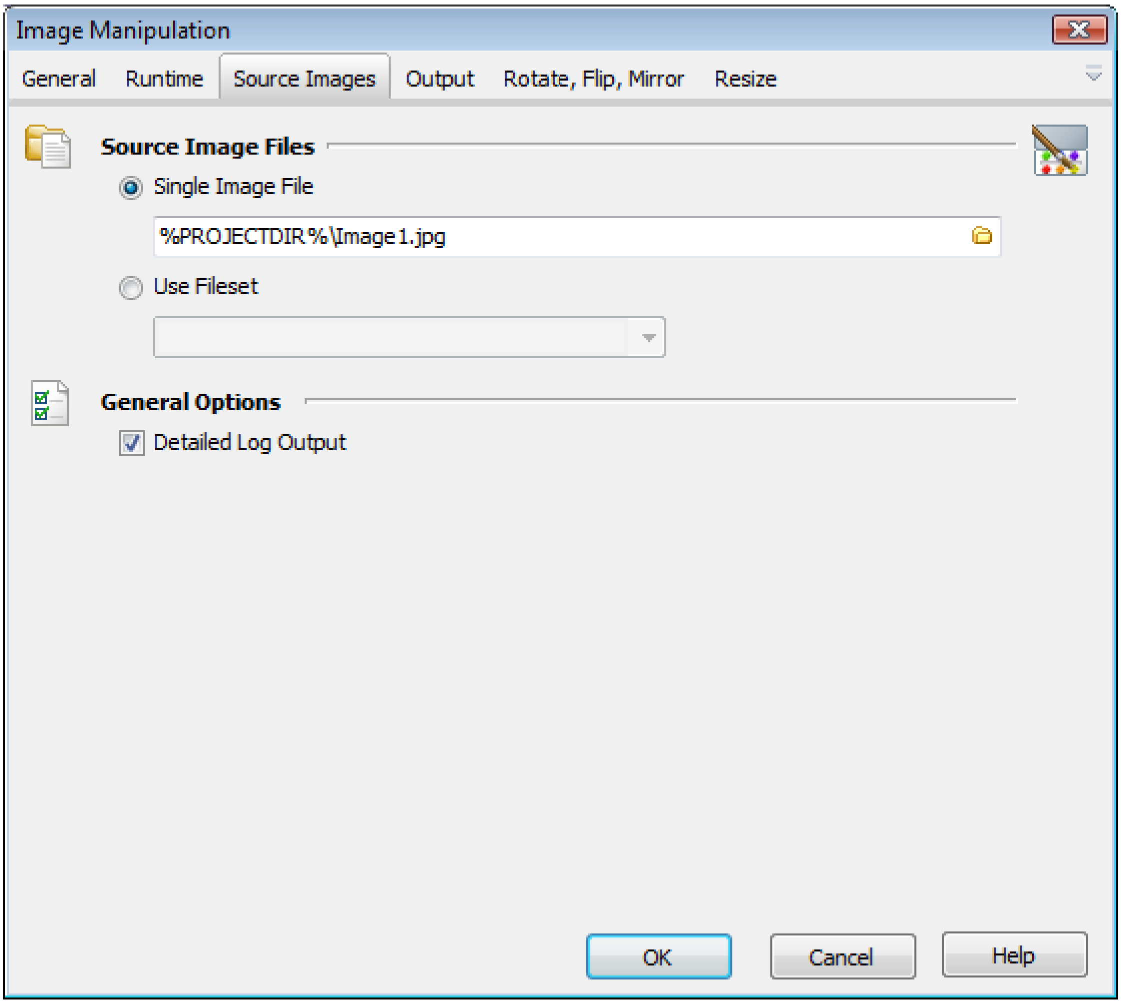
[Automise Professional Edition]

The Image Manipulation action can be used for performing the following operations on image files:

- Rotating
- Flipping
- Mirroring
- Resizing
- Converting to another format

The types of files supported are:

- JPEG (*.jpg)
- Graphics Interchange Format (*.gif)
- Tagged Image File Format (*.tiff)
- Portable Network Graphics (*.png)
- Bitmaps (*.bmp)



Source Image Files

● Single Image File

%PROJECTDIR%\Image1.jpg

○ Use Fileset

General Options

☑ Detailed Log Output

Source Images

The Image Manipulation action can either use a FileSet, or you can specify a single image file. If you want to operate on all files in a certain directory, then you need to create a FileSet using the FileSet Define action.

Detailed Log Output
This option will report more detail in the log. For example, if you are flipping, resizing and mirroring an image, then in the log you will see separate entries for each operation performed as well as the source and destination files.

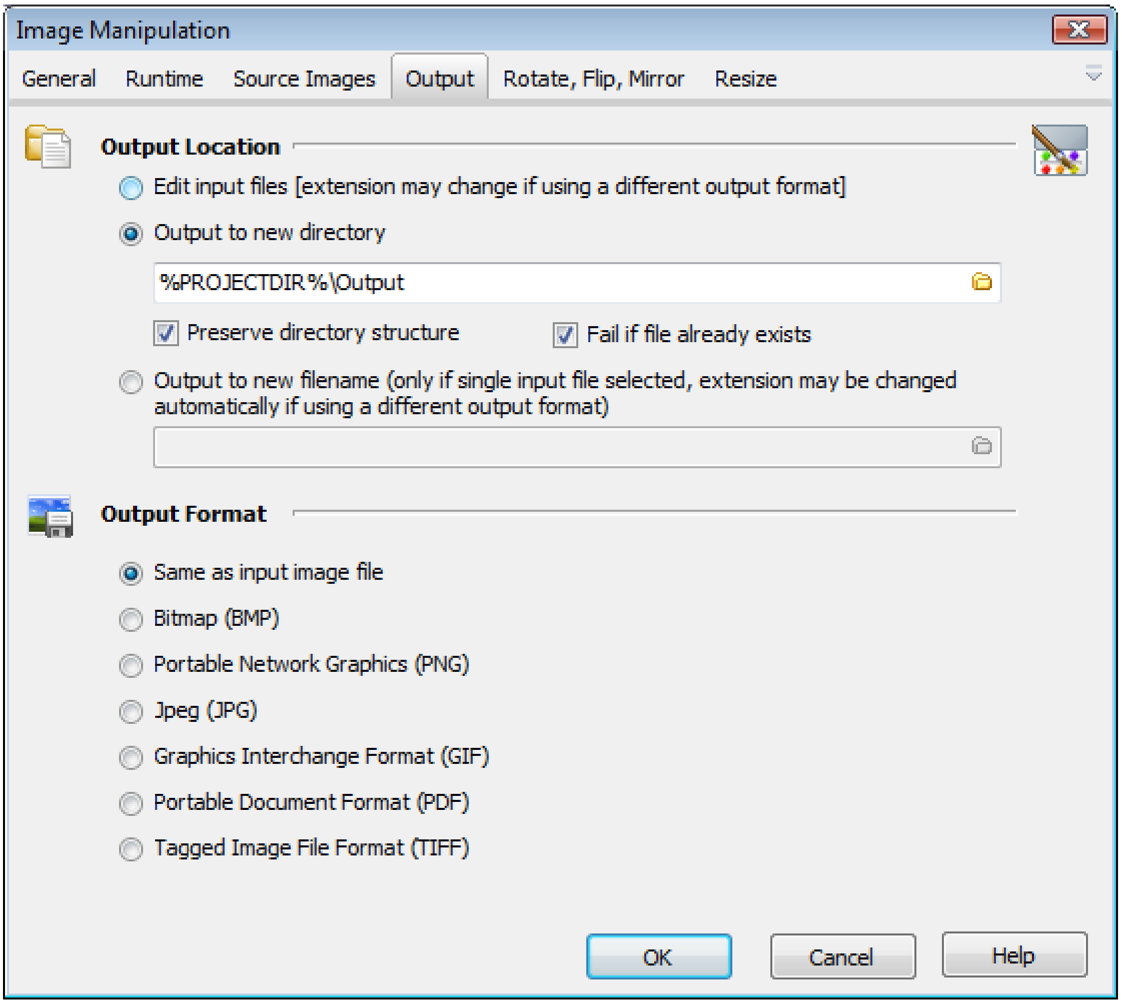


Image Manipulation

Output Location

○ Edit input files [extension may change if using a different output format]

● Output to new directory

%PROJECTDIR%\Output

☑ Preserve directory structure ☑ Fail if file already exists

○ Output to new filename (only if single input file selected, extension may be changed automatically if using a different output format)

Output Format

● Same as input image file

○ Bitmap (BMP)

○ Portable Network Graphics (PNG)

○ Jpeg (JPG)

○ Graphics Interchange Format (GIF)

○ Portable Document Format (PDF)

○ Tagged Image File Format (TIFF)

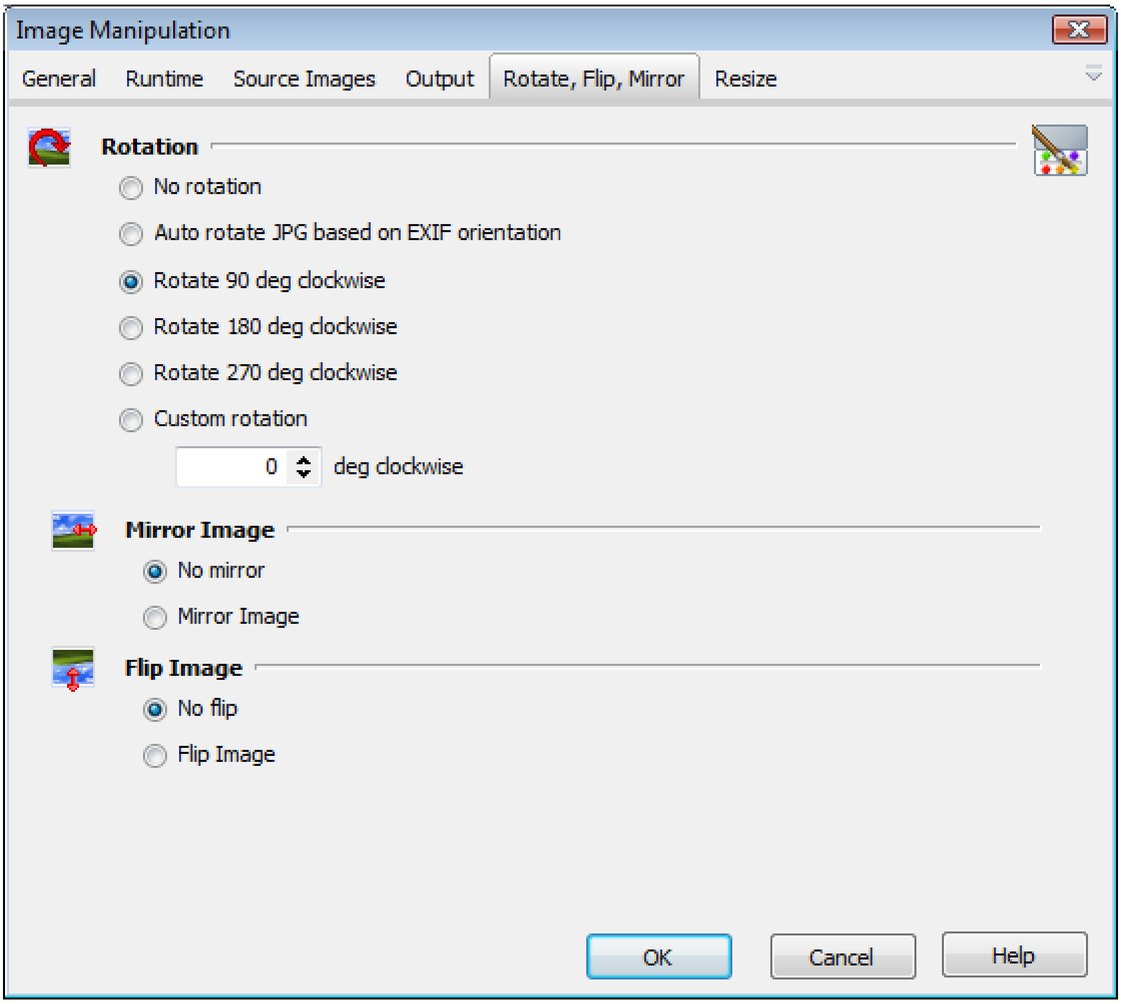
Output Location
There are three options for the output of the image once it has been manipulated in some way:

Edit Input Files : This option will operate directly on the source images and no backups will be created. Any operation performed on the images cannot be undone, so be careful! If you have chosen to save in a different output format, then this option will create a new file if the file type differs from the source file.

Output to New Directory : This option will leave your source files untouched, and will create new files in the chosen directory. The preserve output directory option can be useful when using a FileSet that includes multiple directories - the directory structure of the source files will be recreated under the new directory. Only the directories deeper than the Base Directory in the FileSet will be recreated. The Fail if File already exists option will abort the action if the output file already exists and files already processed will remain.

Output to New File Name : This option can only be used when the source image is a single file. This option allows you to rename the output file. The extension will be generated automatically depending on the output format chosen.

Output Format
The output format allows you to convert the images into a different format. The formats supported are: Bitmap, PNG, Jpeg, PDF, TIFF, and GIF. To use the format of the source image, use the "Same as input image file" option.



Rotation

○ No rotation

○ Auto rotate JPG based on EXIF orientation

● Rotate 90 deg clockwise

○ Rotate 180 deg clockwise

○ Rotate 270 deg clockwise

○ Custom rotation

0 deg clockwise

Mirror Image

● No mirror

○ Mirror Image

Flip Image

● No flip

○ Flip Image

Rotation
The rotation options can be used to perform a series of rotations to the source images.

No Rotation : No rotation operation is performed

Auto Rotation JPG based on EXIF orientation : Most modern digital cameras can record the orientation of the camera when the photo was taken, and they record this information in the JPEG file in the EXIF properties. Using this option an automatic rotation can be performed on the photo so that viewing the photo in any application will always display the photo with the correct orientation. The EXIF orientation flag is also changed so that no other automatic rotation will be performed (by this action or another application). For example, if the photo was taken in portrait, with the camera rotated 90 degrees clockwise, then the photo will automatically be rotated 90 degrees.

Rotate 90 deg clockwise : Perform a 90 degrees clockwise rotation

Rotate 180 deg clockwise : Perform a 180 degrees clockwise rotation

Rotate 270 deg clockwise : Perform a 270 degrees clockwise rotation

Custom rotation : Rotate the image between 0 and 360 degrees

Mirror Image

Mirroring the image will perform a horizontal flip

Flip Image

Flipping the image will perform a vertical flip

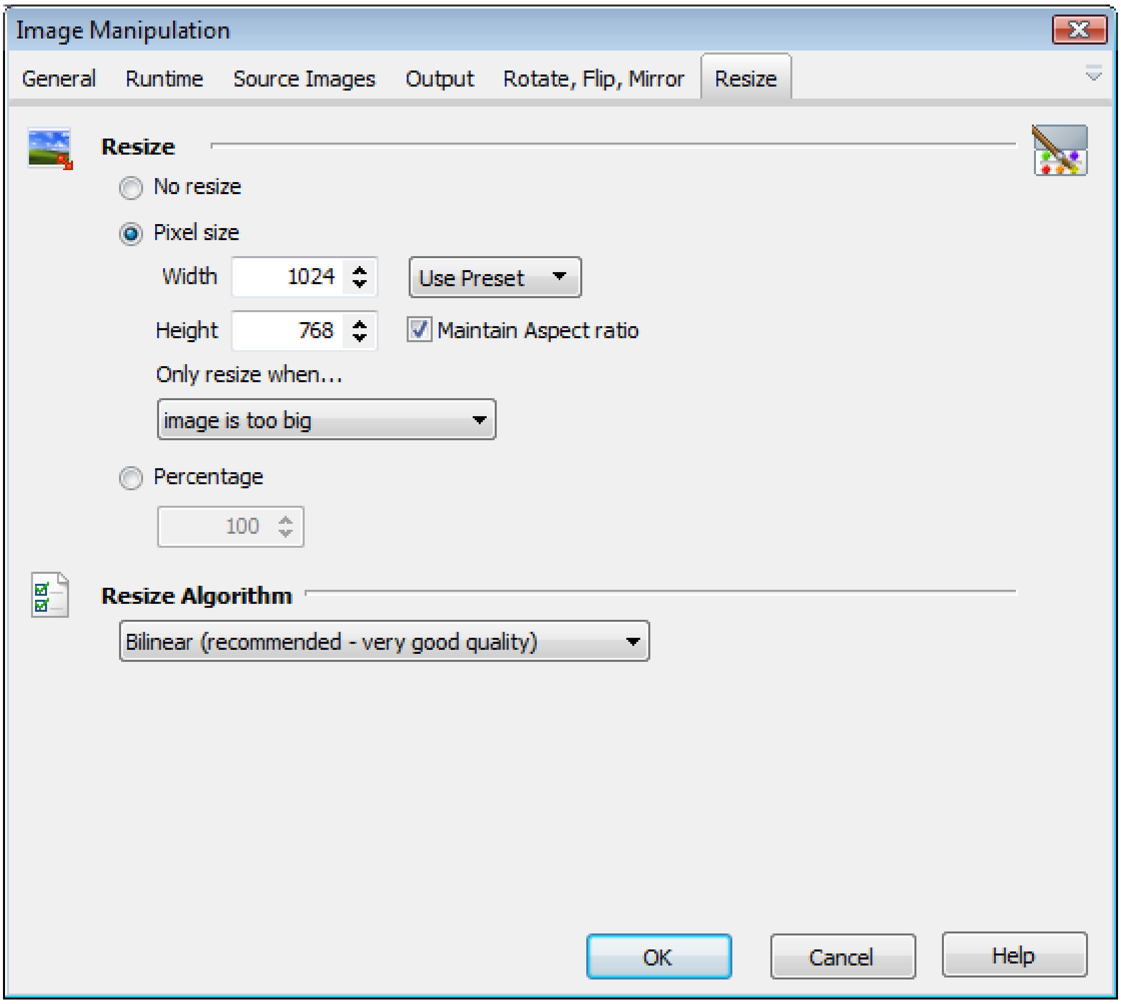


Image Manipulation

Resize

○ No resize

● Pixel size

Width 1024

Height 768

☑ Maintain Aspect ratio

Only resize when...

image is too big

○ Percentage

100

Resize Algorithm

Bilinear (recommended - very good quality)

Resize Image

There are two way to resize an image: by a percentage or to a fixed size. A percentage resize will always maintain the aspect ratio (the ratio between the height and width), whereas maintaining the aspect ratio by specifying a fixed size resize is optional.

Pixel Size : Specify the new width and height. If you want to maintain the aspect ratio then enter -1 in either the height or the width (The Maintain aspect ratio checkbox will force a -1 as the height when the action runs)

Percentage : Specify the new size of the image as a percentage of the original size. So to reduce the image to half the original size, specify 50 percent.

Resize Algorithm : There are many resize algorithms which can be used. The Bilinear algorithm gives very good results, but does require processing time.

6.19.4.2 Lossless JPEG Transform

[Automise Professional Edition]

The Lossless JPEG Transform action is used to perform certain transform operations on JPEG files without any loss of information.

Normally, simply opening and saving a JPEG file (even without performing any manipulation of the image) will results in the file being re-encoded which will result in loss of information as the JPEG algorithm is lossy - see http://en.wikipedia.org/wiki/JPEG and http://en.wikipedia.org/wiki/Lossy_compression.

The types of operations which can be performed by this action are as follows:

Auto Rotation JPG based on EXIF orientation : Most modern digital cameras can record the orientation of the camera when the photo was taken, and they record this information in the JPEG file in the EXIF properties. Using this option an automatic lossless rotation can be performed on the photo so that viewing the photo in any application will always display the photo with the correct orientation. The EXIF orientation flag is also changed so that no other automatic rotation will be performed (by this action or another application). For example, if the photo was taken in portrait, with the camera rotated 90 degrees clockwise, then the photo will automatically be rotated 90 degrees.

Rotate 90 deg clockwise : Perform a lossless 90 degrees clockwise rotation

Rotate 180 deg clockwise : Perform a lossless 180 degrees clockwise rotation

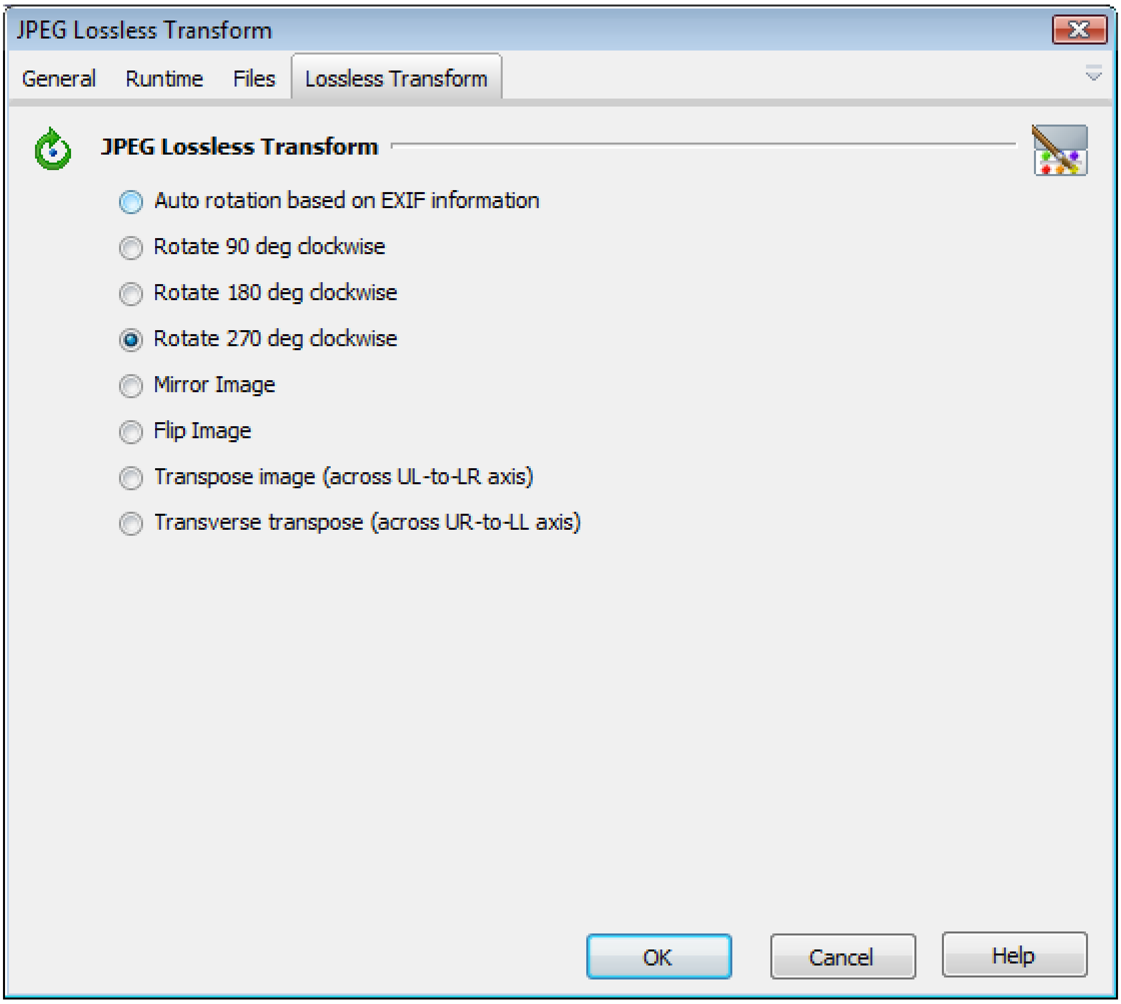
Rotate 270 deg clockwise : Perform a lossless 270 degrees clockwise rotation

Mirror Image : Perform a lossless horizontal flip

Flip Image : Perform a lossless vertical flip

Transpose Image : Mirror image across upper left to lower right axis. Same as horizontal flip and then 270 degree rotate

Transverse Transpose : Mirror image across upper right to lower left axis. Same as horizontal flip and then 90 degree rotate



JPEG Lossless Transform

JPEG Lossless Transform

○ Auto rotation based on EXIF information

○ Rotate 90 deg clockwise

○ Rotate 180 deg clockwise

● Rotate 270 deg clockwise

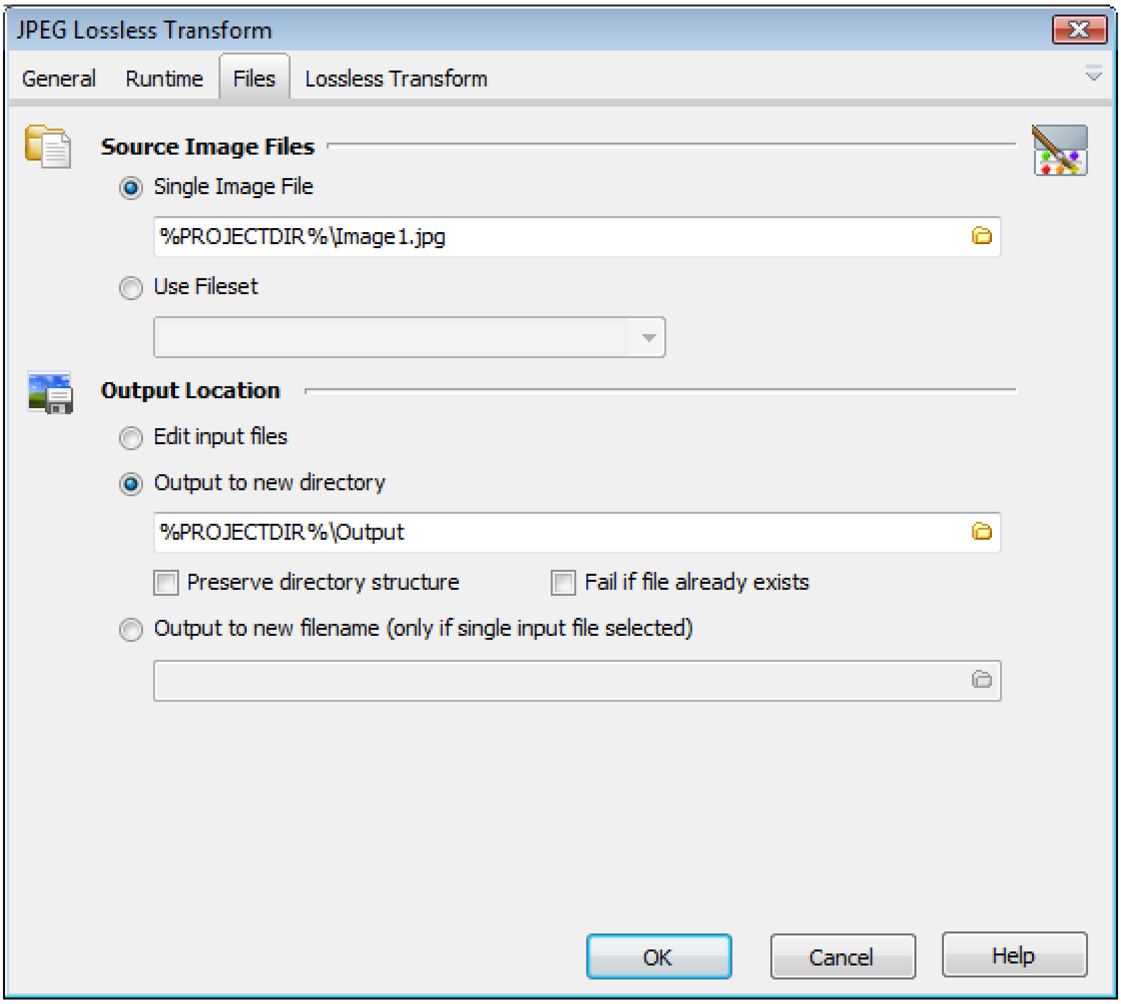
○ Mirror Image

○ Flip Image

○ Transpose image (across UL-to-LR axis)

○ Transverse transpose (across UR-to-LL axis)

Use the Files tab to specify the source images and the output location



JPEG Lossless Transform

Source Image Files

● Single Image File

%PROJECTDIR%\Image1.jpg

○ Use Fileset

Output Location

○ Edit input files

● Output to new directory

%PROJECTDIR%\Output

□ Preserve directory structure □ Fail if file already exists

○ Output to new filename (only if single input file selected)

Source Images

The Image Manipulation action can either use a FileSet, or you can specify a single image file. If you want to operate on all files in a certain directory, then you need to create a FileSet using the FileSet Define action.

Output Location

There are three options for the output of the image once it has been transformed in some way:

Edit Input Files : This option will operate directly on the source images and no backups will be created. Any operation performed on the images cannot be undone, so be careful!

Output to New Directory : This option will leave your source files untouched, and will create new files in the chosen directory. The preserve output directory option can be useful when using a FileSet that includes multiple directories - the directory structure of the source files will be recreated under the new directory. Only the directories deeper than the Base Directory in the FileSet will be recreated. The Fail if File already exists option will abort the action if the output file already exists and files already processed will remain.

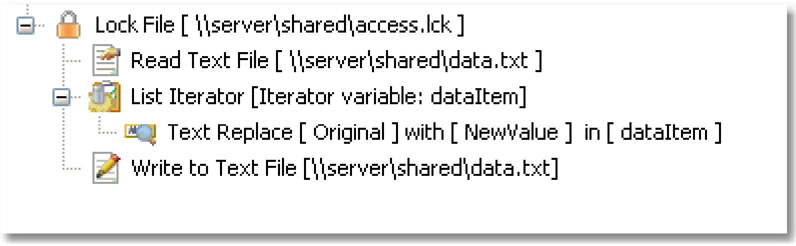
Output to New File Name : This option can only be used when the source image is a single file. This option allows you to rename the output file.

6.19.5 Mutex/Lock File Action

The Mutex/Lock File Action allows you to establish access to an exclusive resource. The resource can be either a lock file or a global Windows mutex object. The action can be used to control exclusive access within a Automise project, between multiple concurrent instances of Automise, or between Automise and a different application.

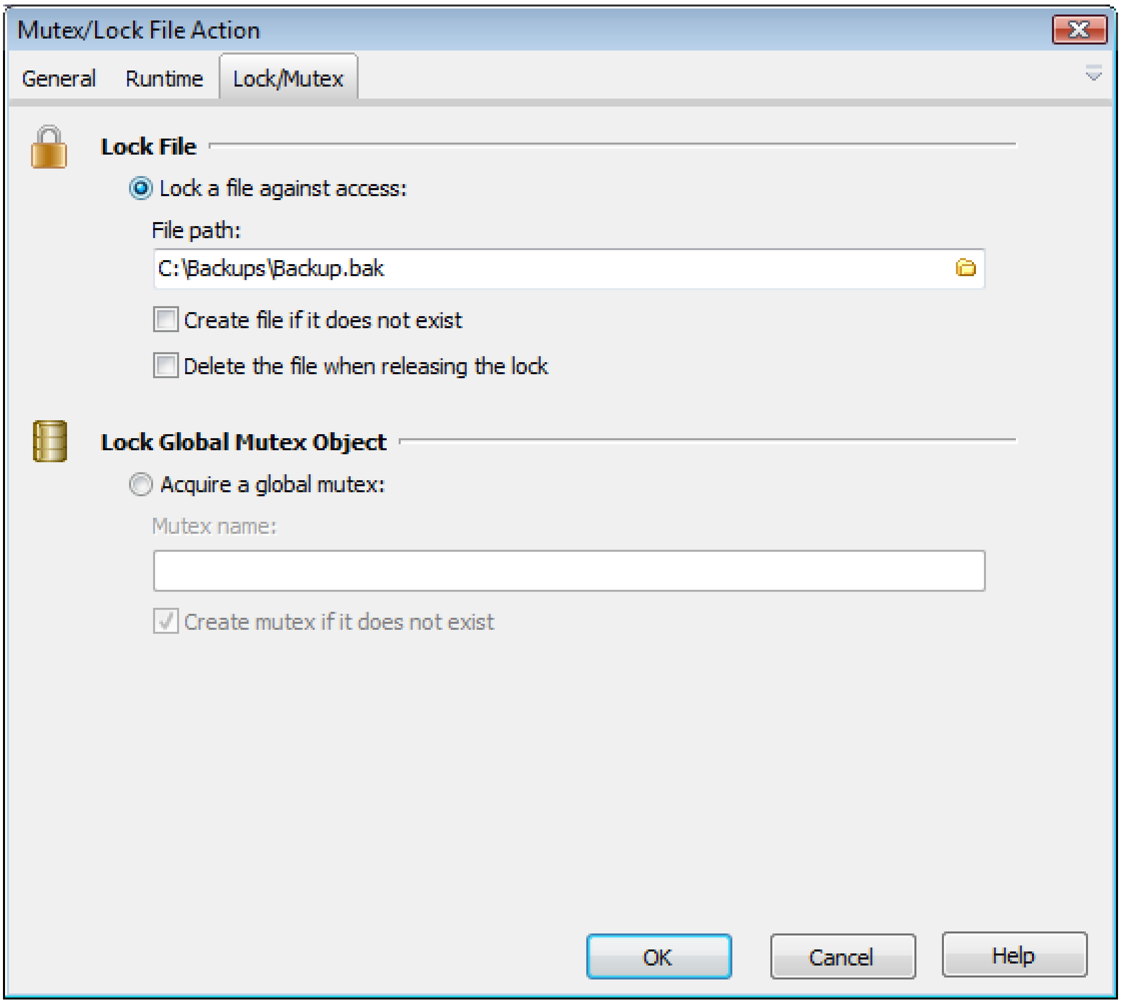
The exclusive access lasts until all child actions have finished running.

Example



```
Lock File [ \\server\shared\access.lck ]
    Read Text File [ \\server\shared\data.txt ]
    List Iterator [Iterator variable: dataItem]
        Text Replace [ Original ] with [ NewValue ]  in [ dataItem ]
    Write to Text File [\\server\shared\data.txt]
```

In the above example, exclusive access to the lock file "\\server\shared\access.lck" will be established, then all of the child actions will run. Once the last action, "Write To Text File", has completed, the file will be released.



Lock File

● Lock a file against access:

File path:

C:\Backups\Backup.bak

□ Create file if it does not exist

□ Delete the file when releasing the lock

Lock Global Mutex Object

○ Acquire a global mutex:

Mutex name:

☑ Create mutex if it does not exist

Lock File

If this option is set, the specified file path will be locked exclusively. The action will block until it obtains exclusive access to the file, or a timeout is reached. No access (read, write or delete) is allowed to the file until the lock action releases it.

Create file if it doesn't exist - If this option is not enabled, and the file to lock is not found at runtime, then the action will fail.

Delete the file when releasing the lock - Enable this option to delete the file when the exclusive lock is released.

Note that because the file is locked, child actions of the Mutex/Lock File Action will not be able to access the file. It is not recommended that you lock files which contain the data you intend to use. Instead, create a dedicated lock file.

Shared server files can be specified by using UNC paths, as shown above. Note that Automise will need full access to the network share.

Lock Global Mutex Object

Mutexes are a class of Windows operating system object. Automise allows you to acquire and release mutexes which belong to the system's Global namespace, which is shared between all other processes on the system.

Mutex name - Specify the name of the global mutex to acquire when the action runs (and release once all children are completed.) In Windows, the full name of the mutex will be "Global\<MutexName>".

Create mutex if it doesn't exist - Check this box in order to create the mutex if it is not found at runtime. You should only disable this option if you know for certain that the mutex has been created and retained by another process. This is because mutex objects are deleted as soon as no process retains a handle to them. The action does not keep a handle to the mutex object it uses, except for when the mutex is acquired.

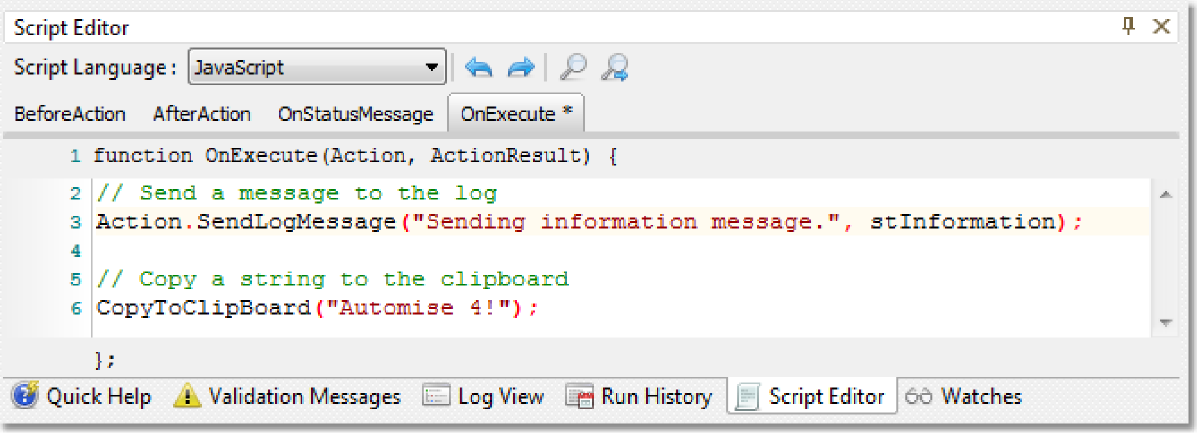
Timeout

You can set a timeout for the action under the "Runtime" -> "Timing Settings" tab. The action will fail if the timeout is reached and the mutex or file is still unavailable.

6.19.6 Run Script Action

The Run Script Action is a simple action that provides an extra script event, OnExecute, where you can place active script code:

(For more information on Automise scripting and action script events, see the Scripting in Automise overview topic.)

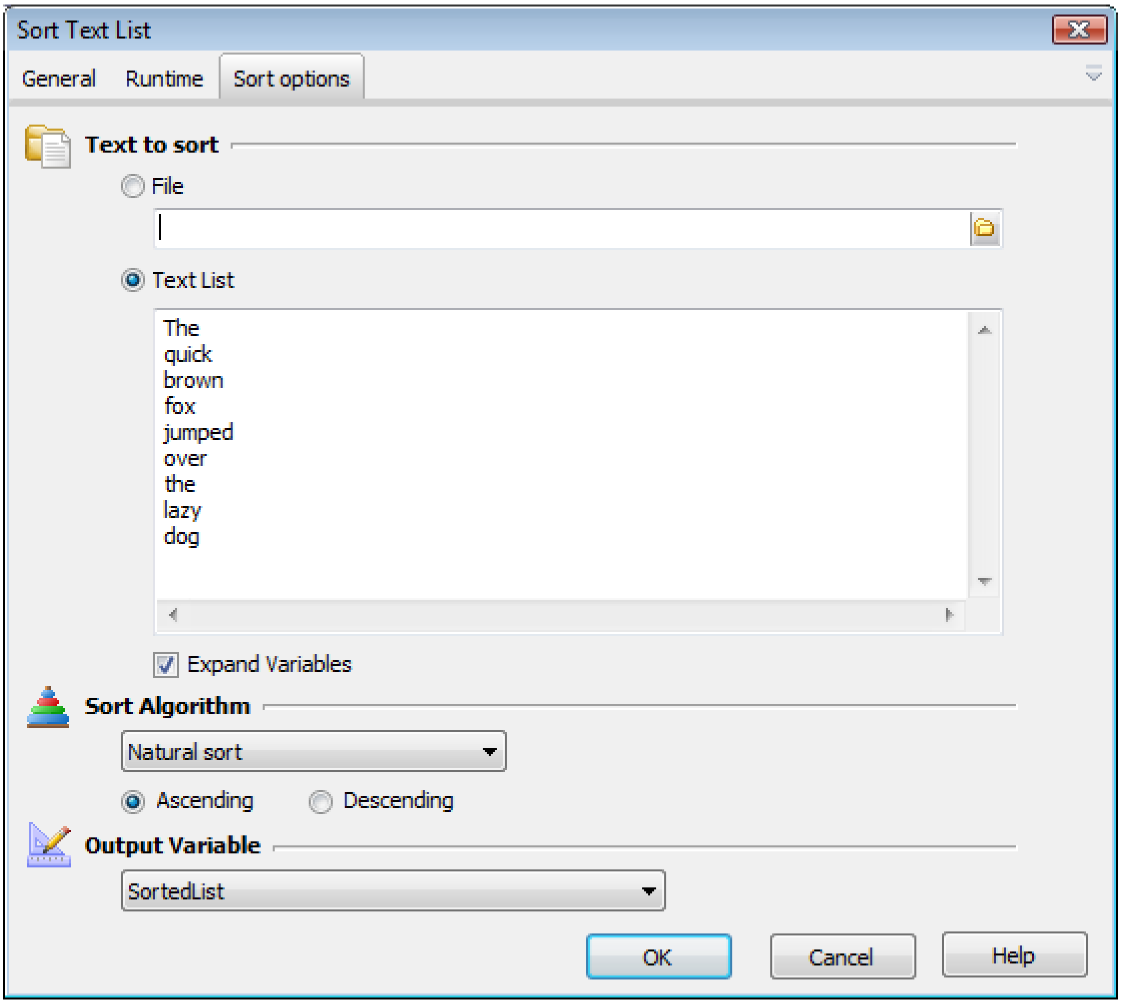


```
function OnExecute(Action, ActionResult) {
// Send a message to the log
Action.SendLogMessage("Sending information message.", stInformation);

// Copy a string to the clipboard
CopyToClipBoard("Automise 4!");
};
```

The OnExecute event has an extra boolean parameter, ActionResult, which determines the outcome of the action. ActionResult starts set to true. To fail the action, set ActionResult to false.

6.19.7 Sort Text List Action

The Sort Text List allows you to sort a list of items or the text in a file using various algorithms in either ascending or descending order.



○ File

● Text List

```
The
quick
brown
fox
jumped
over
the
lazy
dog
```

☑ Expand Variables

Sort Algorithm: Natural sort

● Ascending ○ Descending

Output Variable: SortedList

Text to sort - choose either a text file or enter a text list to sort

- **File** - select a text file to sort
- **Text List** - enter a list of text to sort. If the items you want to sort are in a variable, then enter the variable (eg. %MyItems%) and make sure the **Expand Variables** option is checked. To sort a list of variables, turn off the Expand Variables option.

Sort Algorithm - select the sorting algorithm to use to sort your list. Select to sort the list in ascending order or descending order.

- Natural Sort - sort order similar to how Windows Explorer sorts files (not case sensitive and numbers are treated as such)
- ASCII sort - basic ASCII sorting (not case sensitive)
- ASCII case sensitive sort - exactly that
- Random - items are randomly ordered, using the Fisher-Yates shuffle (see: http://en.wikipedia.org/wiki/Knuth_shuffle)

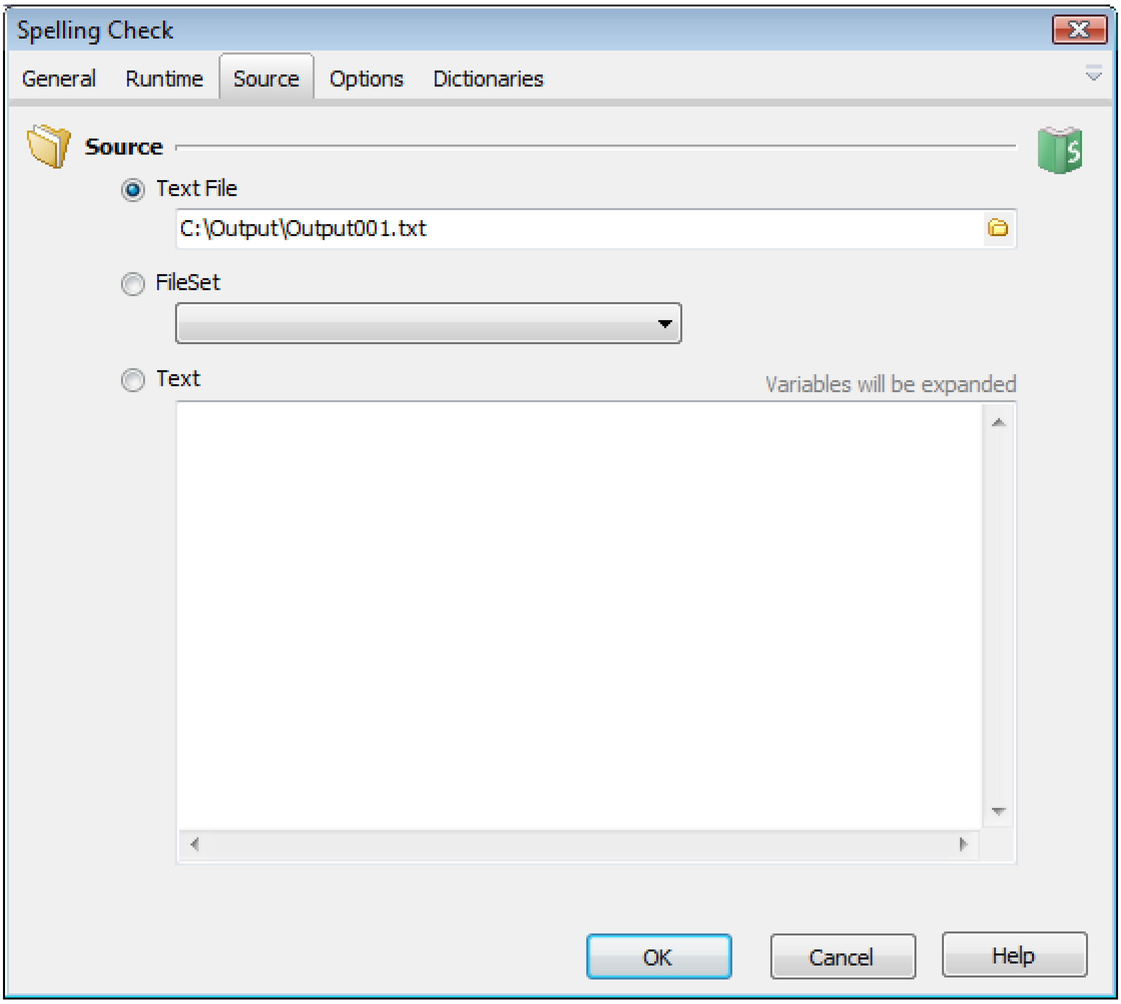
Output Variable

The sorted text list will be output to the selected variable. To write the items to a file, use the Write to Text File action and specify the variable name as the contents.

6.19.8 Spell Checking

[Automise Professional Edition]

The Spell Check action allows you to check a text file for any spelling errors (ie. unknown words).



● Text File

C:\Output\Output001.txt

○ FileSet

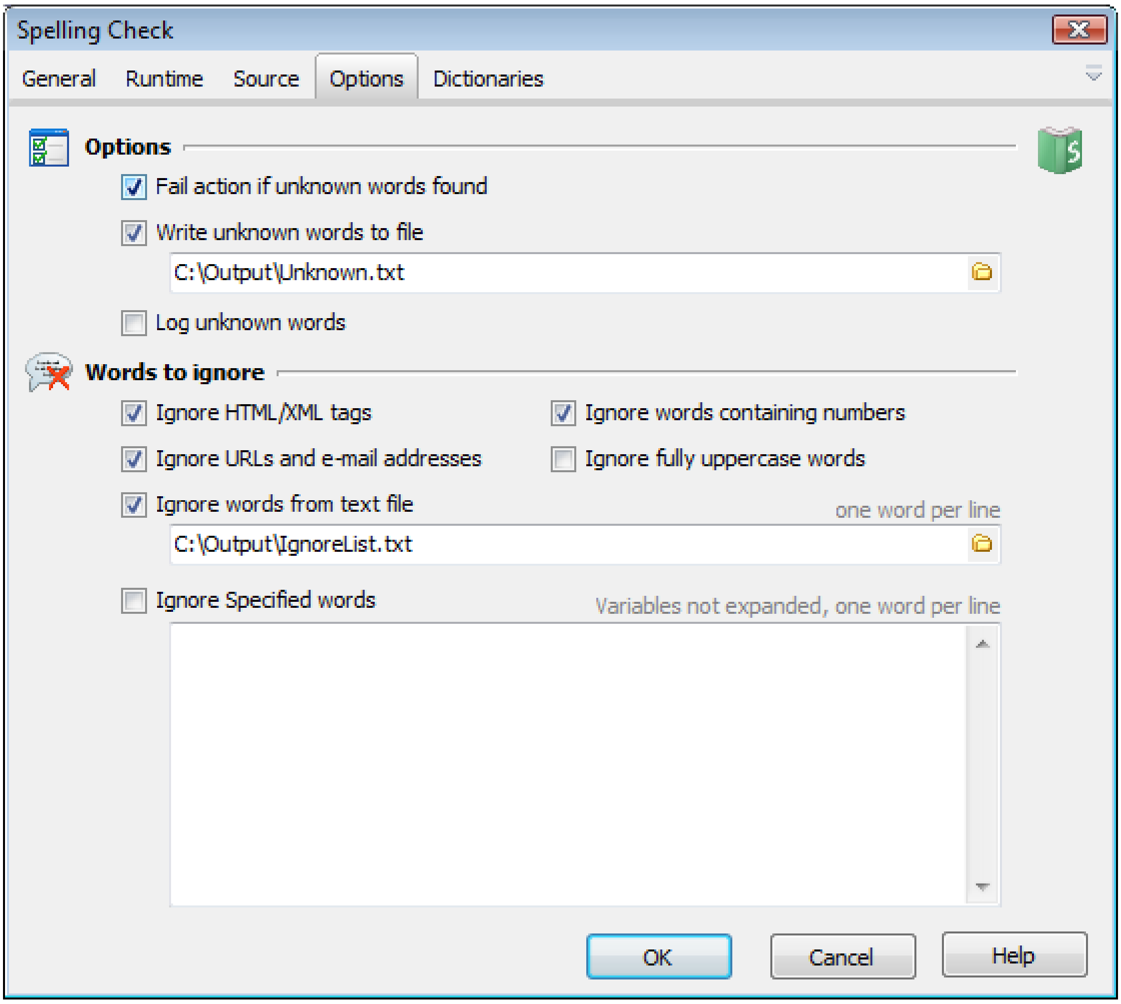
○ Text

Select the source to check spelling of:

Text File - specify a text file

FileSet - specify a FileSet to use. Make sure that only text files are included in the FileSet

Text - enter text to check. You may use variables here which will be expanded first.



Options

☑ Fail action if unknown words found

☑ Write unknown words to file

C:\Output\Unknown.txt

☐ Log unknown words

Words to ignore

☑ Ignore HTML/XML tags

☑ Ignore words containing numbers

☑ Ignore URLs and e-mail addresses

☐ Ignore fully uppercase words

☑ Ignore words from text file

one word per line

C:\Output\IgnoreList.txt

☐ Ignore Specified words

Variables not expanded, one word per line

Options

Fail if unknown words found - setting this will fail the action if one or more words are unknown

Write unknown words to file - Any unique unknown words found will be appended to the specified text file. The file will be created if it doesn't exists.

Log unknown words - each unique unknown word will be written to the log

Words to Ignore

Ignore HTML/XML tags - this will ignore words inside "<" ">" tags. eg. <speeling>errror</speeling> will only test the word "errror" and ignore "speeling"

Ignore words containing numbers - this will ignore words like "3ware"

Ignore URLs and e-mail addresses - this ignores words that start with an internet protocol like "http://" and also words that contain the @ symbol.

Ignore fully uppercase words - this will ignore words like "SHOUT"

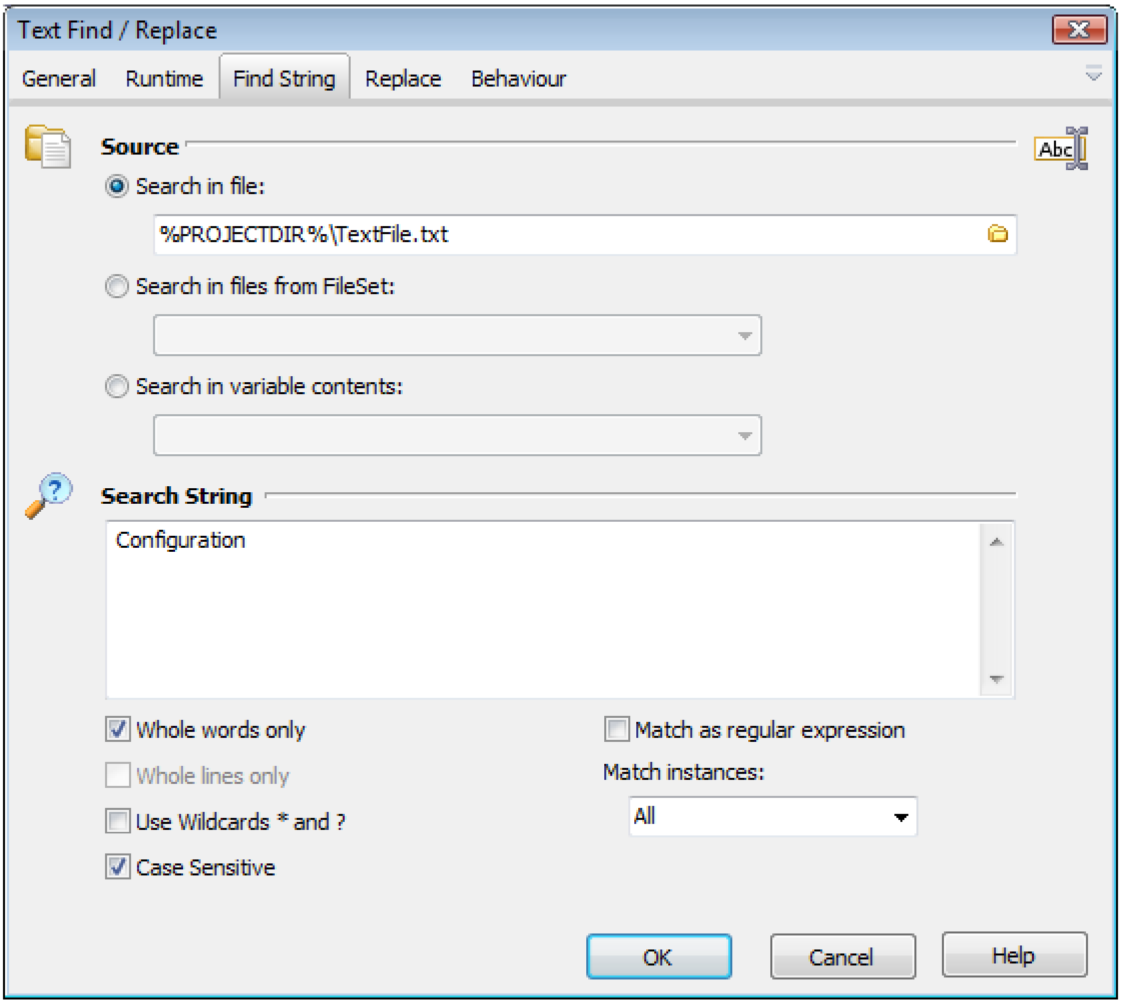
Ignore words from text file - each line of the file is interpreted as a word and will be ignored

Custom dictionary - you may specify other Addict dictionaries to use. See here for other dictionaries: http://www.addictivesoftware.com/dicts-extern.htm

Ignore Specified words - add other words to ignore, one line per word

6.19.9 Text Find / Replace Action

The Text Find / Replace action allows you search for or replace a string. You can search in a file, or in variable contents. The search string can be plaintext or a regular expression. Wildcards are also available.



Text Find / Replace

Source

● Search in file:

%PROJECTDIR%\TextFile.txt

○ Search in files from FileSet:

○ Search in variable contents:

Search String

Configuration

☑ Whole words only

□ Whole lines only

□ Use Wildcards * and ?

☑ Case Sensitive

□ Match as regular expression

Match instances: All

Source

The text to search can be sourced from either a file, the contents of a variable, or from each of the files in a FileSet.

When processing a FileSet, each file will be processed in turn (see below for more details, under the Behaviour tab.)

Search String

The string to search for can be plain text, plain text with wildcards, or a regular expression. It can optionally span multiple lines. A range of search options are available:

"Whole words Only"

If this option is checked, the search will only match whole words.

"Whole lines Only"

If this option is checked, the search will only match results which make up an entire line.

*"Wildcard * matches any substring"*

If this box is checked, the search string can contain the wildcard characters "*" and "?". * will match any number of characters (including none), whereas ? will match any single character.

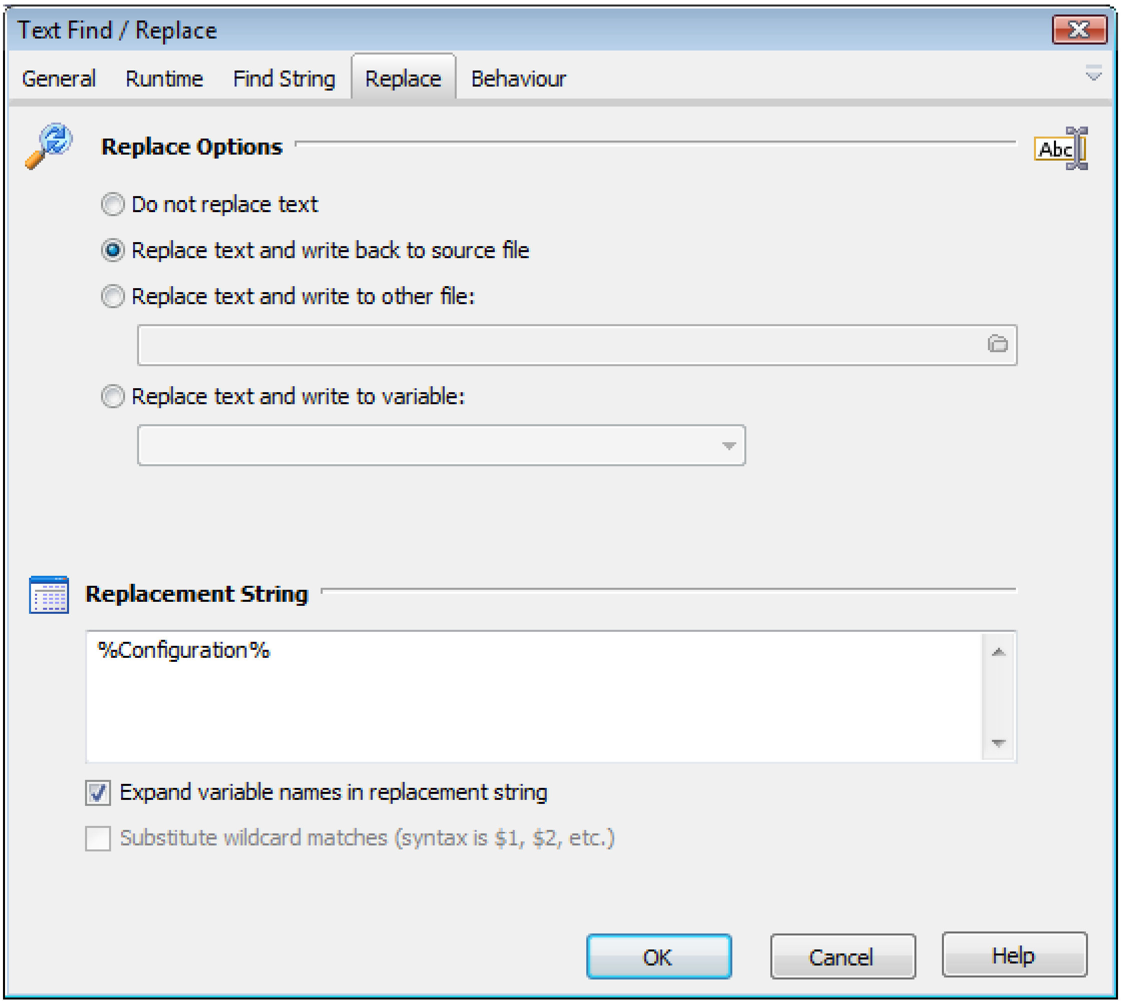
If using wildcards, the characters * and ? can be escaped as ** and ??, respectively.

"Case Sensitive"

If checked, the search will be case sensitive.

"Match as Regular Expression"

If checked, the search string will be treated as a regular expression. See the Regular Expression reference for details. If this option is set, the other search options listed above do not apply and will be grayed out.



Replace Options

○ Do not replace text

● Replace text and write back to source file

○ Replace text and write to other file:

○ Replace text and write to variable:

Replacement String

%Configuration%

☑ Expand variable names in replacement string

□ Substitute wildcard matches (syntax is $1, $2, etc.)

Replace Options

"Do not replace text"

If this option is set, then the action behaves as a "Text Find" action only. All other options on this property page will be disabled.

"Replace text and write back to source file/variable"

If this option is set, then the action will make relevant replacements and write the new text over the existing source file or variable.

"Replace text and write to (other) file"

If this option is set, the output text will be written to another file (as specified.)

Replacement String

Enter the string you wish to use as a replacement.

"Expand variable names in replacement string"

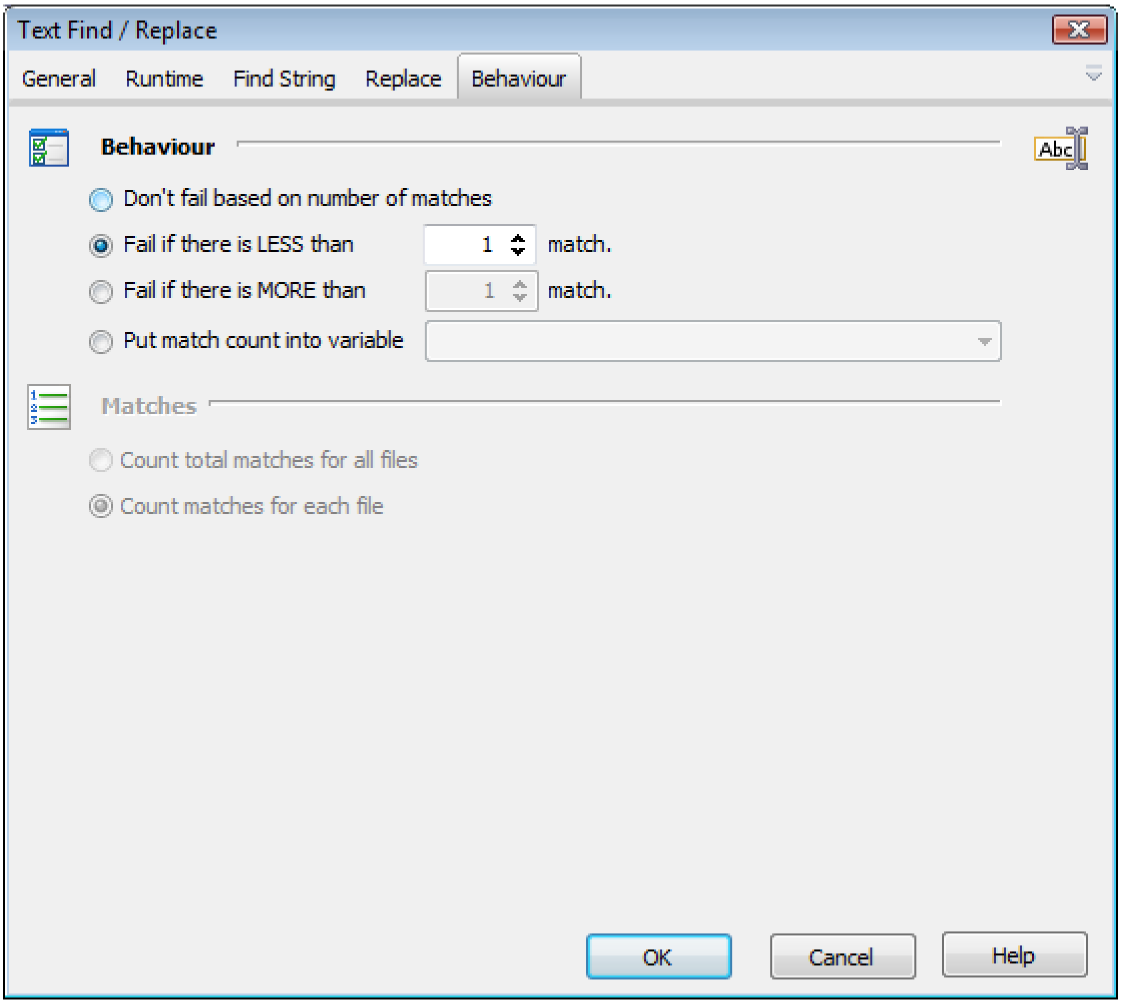
Enable this option to expand any variable reference (ie %VarName%) in the replacement string to their variable values, before making any replacements.

"Substitute wildcard matches (syntax is $1, $2, etc.)"

Enable this option to replace $1, $2, etc. with the text of the corresponding wildcard (*) match from the search string. *"Wildcard * matches any substring"* must be selected for this option to take effect. In the example above, $1 would be replaced with the value which matched the leading * from "*.dll" (based on the search string in the above screenshot.)

If *"Match as Regular Expression"* is enabled, then this option is renamed to become *"Substitute regular expression matches (syntax is $&, $0, $1, $2, etc.)"*. In this case, matching and replacement is performed as per the standard Regular Expression rules for sub-expression substitution.

Note:
If the replacement text contains numbers adjacent to the substitution marker, escape it like this: ${1}
For example, if you want to replace every "ver <n>" with "ver <n>5" in strings like "Release ver 3 is now out", you could use "ver (\d+)" as the regular expression to match, and replace it with "ver ${1}5".



Behaviour

○ Don't fail based on number of matches
● Fail if there is LESS than 1 match.
○ Fail if there is MORE than 1 match.
○ Put match count into variable

Matches

○ Count total matches for all files
● Count matches for each file

Behaviour

You can choose to have the action fail if there are less than or more than a specified number of matches found. Note that if replacement is enabled and the action fails because of too few or too many matches, then the replacement string is not written to the target file/variable. The only exception is when using FileSets and "*Count total matches...*" is enabled (see below.)

Alternatively, you can choose to write the total number of matches to a variable.

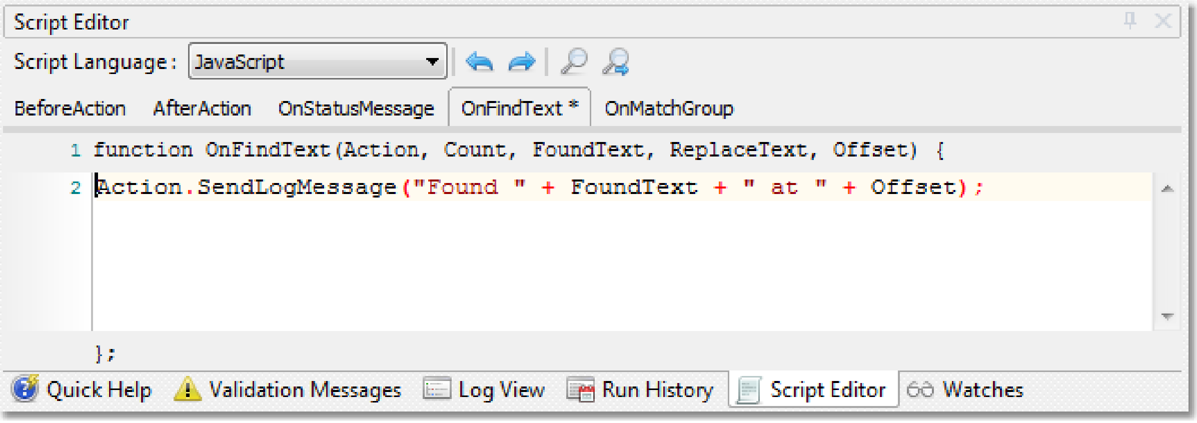
Matches

This option is only enabled if you are searching all files in a FileSet. It allows you to specify whether or not you wish to count the grand total of matches across all files, or count matches on a file by file basis.

When replacement is enabled and the action is set to "Count total matches for all files", the replacement text will be written to every file regardless of whether or not the action fails. When "Count matches for each file" is selected, the replacement text will only be written if the action succeeds for that file. If the action fails while processing a file, no other files in the FileSet will be processed (the action stops.)

Script Events

In the Script Editor tab, you can see the *OnFindText* scripting event which allows you to respond to individual matches.



```
function OnFindText(Action, Count, FoundText, ReplaceText, Offset) {
Action.SendLogMessage("Found " + FoundText + " at " + Offset);

};
```

The *OnFindText* event has the following parameters:

Action
An object representing the Text Find Action.

Count
How many matches have been found so far as this action runs (starts from one and increments with each match.)

FoundText
The exact string that was matched. For plain text searches, this will be the search string. For regular expressions, it will be the specific instance which was matched.

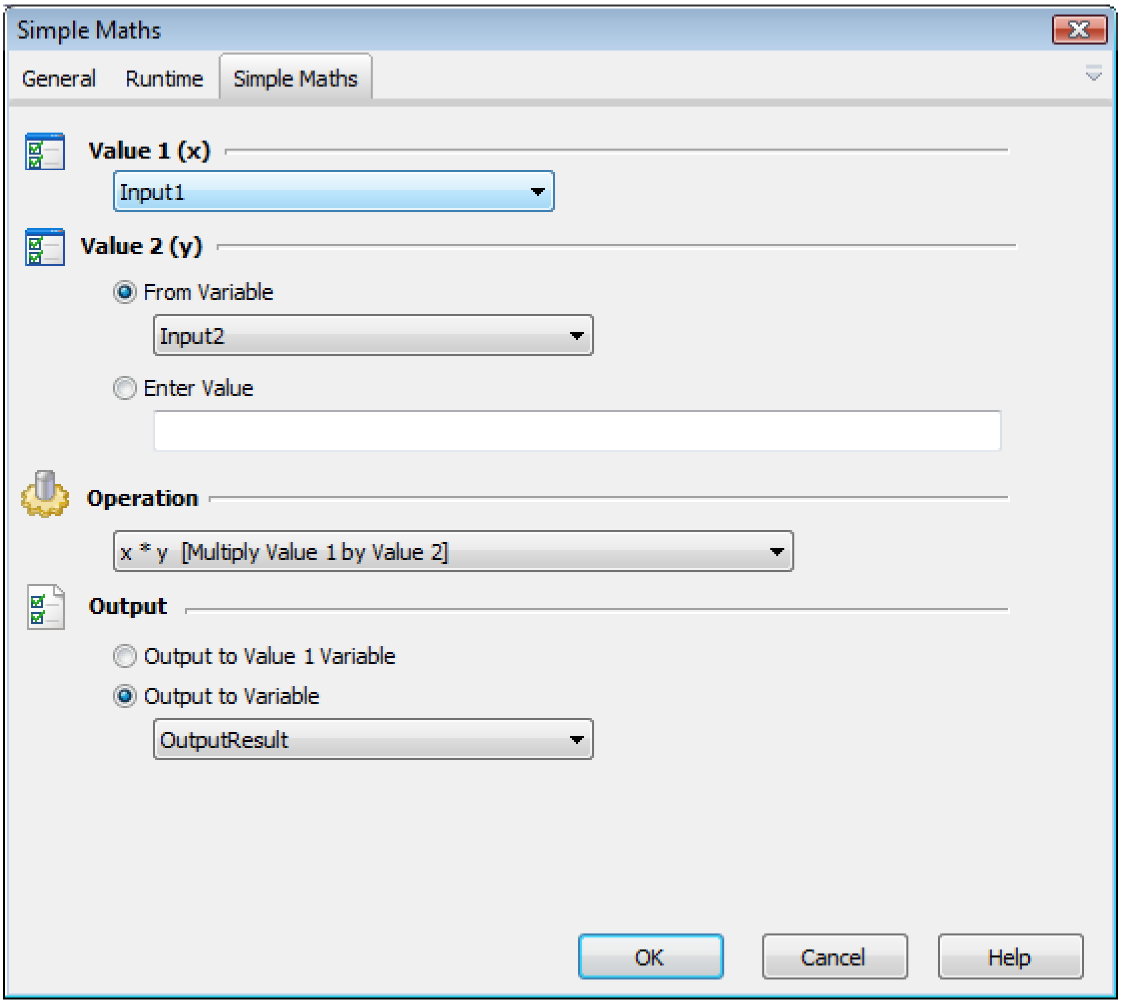
Offset
The offset in the source at which the text was found. *Offset* is zero-indexed.

ReplaceText

The string which will be used for replacements (if replace is enabled.) This is a pass-by-reference parameter, so changes made here will reflect the replacement string which is used by the action.

6.19.10 Simple Maths Action

Use the Simple Maths action to perform mathematical calculations.



Simple Maths

Value 1 (x)

Input1

Value 2 (y)

● From Variable

Input2

○ Enter Value

Operation

x * y [Multiply Value 1 by Value 2]

Output

○ Output to Value 1 Variable

● Output to Variable

OutputResult

All Simple Maths calculations require an x value (Value 1) which needs to be provided from a variable. Some calculations require a y value, which can be provided from a variable or you may enter the value manually (and this field supports variables).

Select the operation to perform and then select if you would like the result to be written back to the Value 1 variable or written to a different variable.

6.20 .NET Actions

6.20.1 .Net Framework Tools

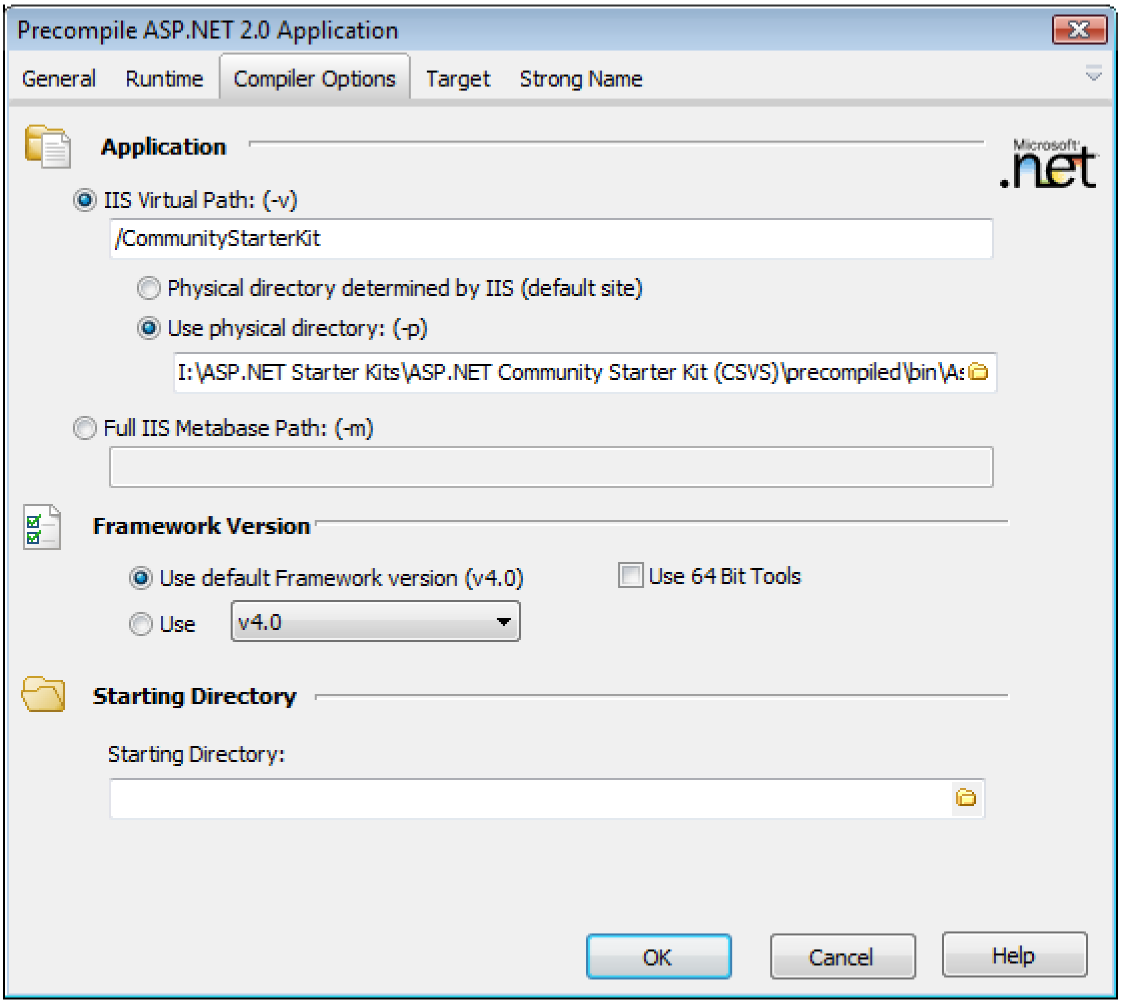
The .Net Framework tools are installed as part of the .Net framework (ie. if you have .Net installed, you have the framework tools)

6.20.1.1 Precompile ASP.NET 2.0 Application

The ASP.NET 2.0 compiler action allows you to precompile ASP.NET 2.0 applications before deployment.

Precompiling has two main advantages:

- The ASP application does not need to be compiled when first accessed.
- By deploying only compiled ASP applications, the ASPX source files do not need to be in the IIS virtual directories.



Application

● IIS Virtual Path: (-v)

/CommunityStarterKit

○ Physical directory determined by IIS (default site)

● Use physical directory: (-p)

I:\ASP.NET Starter Kits\ASP.NET Community Starter Kit (CSVS)\precompiled\bin\A:

○ Full IIS Metabase Path: (-m)

Framework Version

● Use default Framework version (v4.0) □ Use 64 Bit Tools

○ Use v4.0

Starting Directory

Starting Directory:

Application Path

There are three ways to specify the path to your ASP.NET Application:

IIS Virtual Path & Physical Directory Determined by IIS

The application will be located by using the IIS metabase and the default website. The physical (local) folder will be determined via the IIS metabase.

IIS Virtual Path & 'Use Physical Directory'

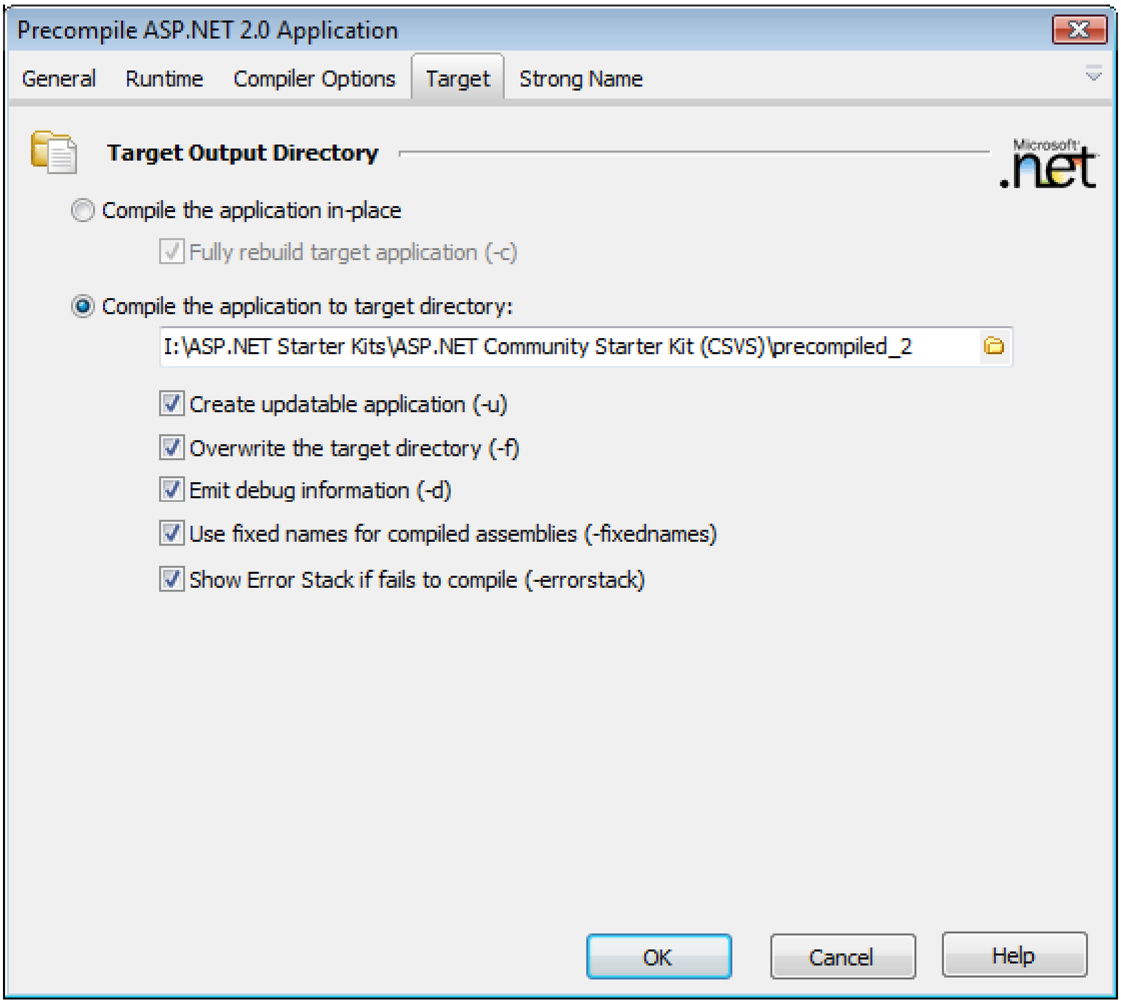
The application will be compiled from the specified physical (local) directory, but as if it was installed at the IIS virtual directory given by the path. This is useful if you want to compile your ASP.NET application from one (offline) directory, but deploy the assemblies in another directory.

Full IIS Metabase Path

The application will be located by using a full IIS metabase path. The physical (local) folder will be determined via the IIS metabase.

Framework Version

Specify the .NET framework version that you want to compile with. The ASP.NET 2.0 Compiler action requires .NET 2.0.



Precompile ASP.NET 2.0 Application

Target Output Directory

○ Compile the application in-place

☑ Fully rebuild target application (-c)

● Compile the application to target directory:

I:\ASP.NET Starter Kits\ASP.NET Community Starter Kit (CSVS)\precompiled_2

☑ Create updatable application (-u)

☑ Overwrite the target directory (-f)

☑ Emit debug information (-d)

☑ Use fixed names for compiled assemblies (-fixednames)

☑ Show Error Stack if fails to compile (-errorstack)

Target Output Directory

Compile the application in-place

The application will compile to the same directory as the source.

Fully rebuild target application : All sources will be rebuilt, not just those which have changed.

Compile the application to target directory

The application will compile to the specified target directory. Compiling to a target directory automatically implies "fully rebuild target application."

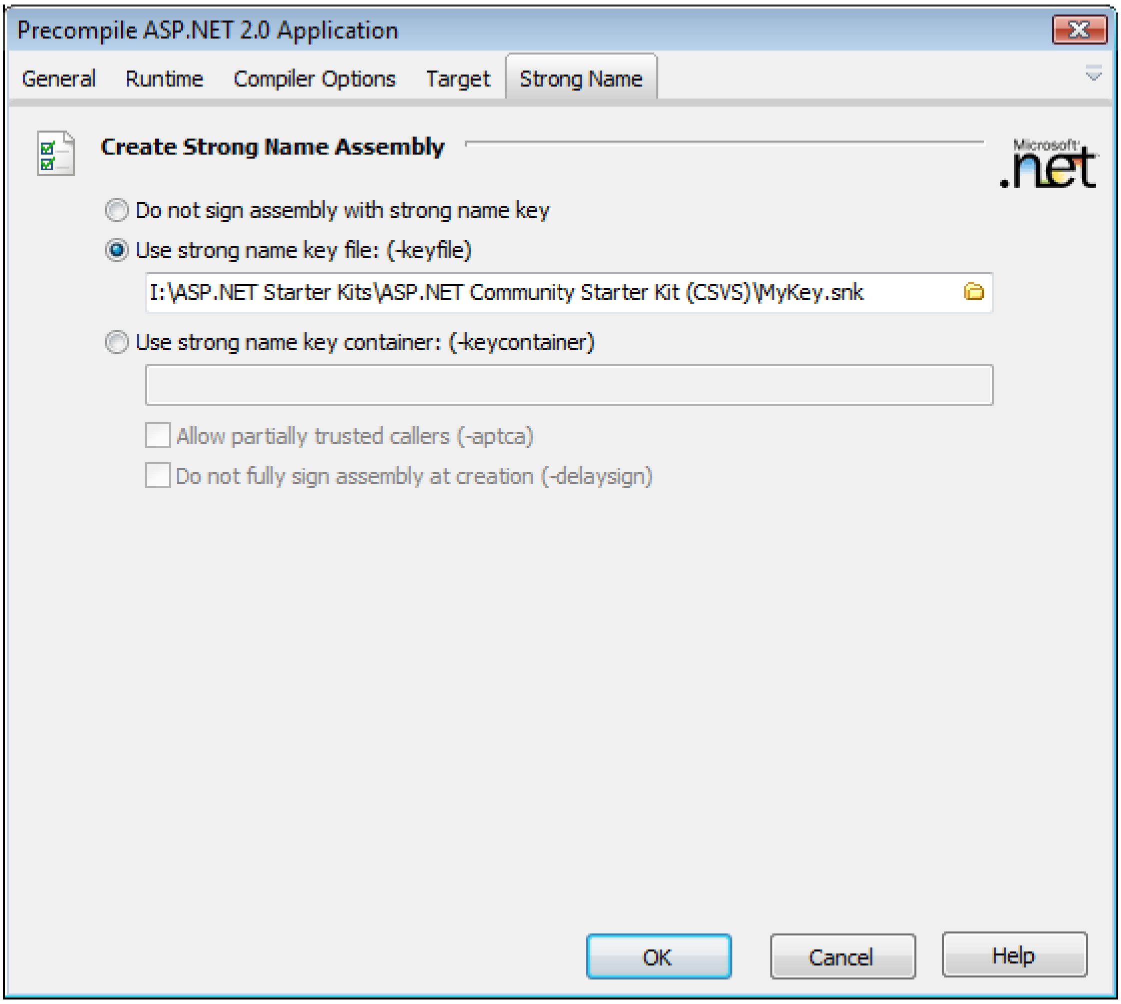
Create updatable application : The compiled application will be updatable.

Overwrite the target directory : If the target directory already exists, its contents will be overwritten.

Emit debug information : The compiler will emit debug information to the log during the compile process.

Use fixed names for compiled assemblies : The compiled assemblies will be given fixed names.

(If this option is not set, the names will be autogenerated.)



Precompile ASP.NET 2.0 Application

Create Strong Name Assembly

○ Do not sign assembly with strong name key

● Use strong name key file: (-keyfile)

I:\ASP.NET Starter Kits\ASP.NET Community Starter Kit (CSVS)\MyKey.snk

○ Use strong name key container: (-keycontainer)

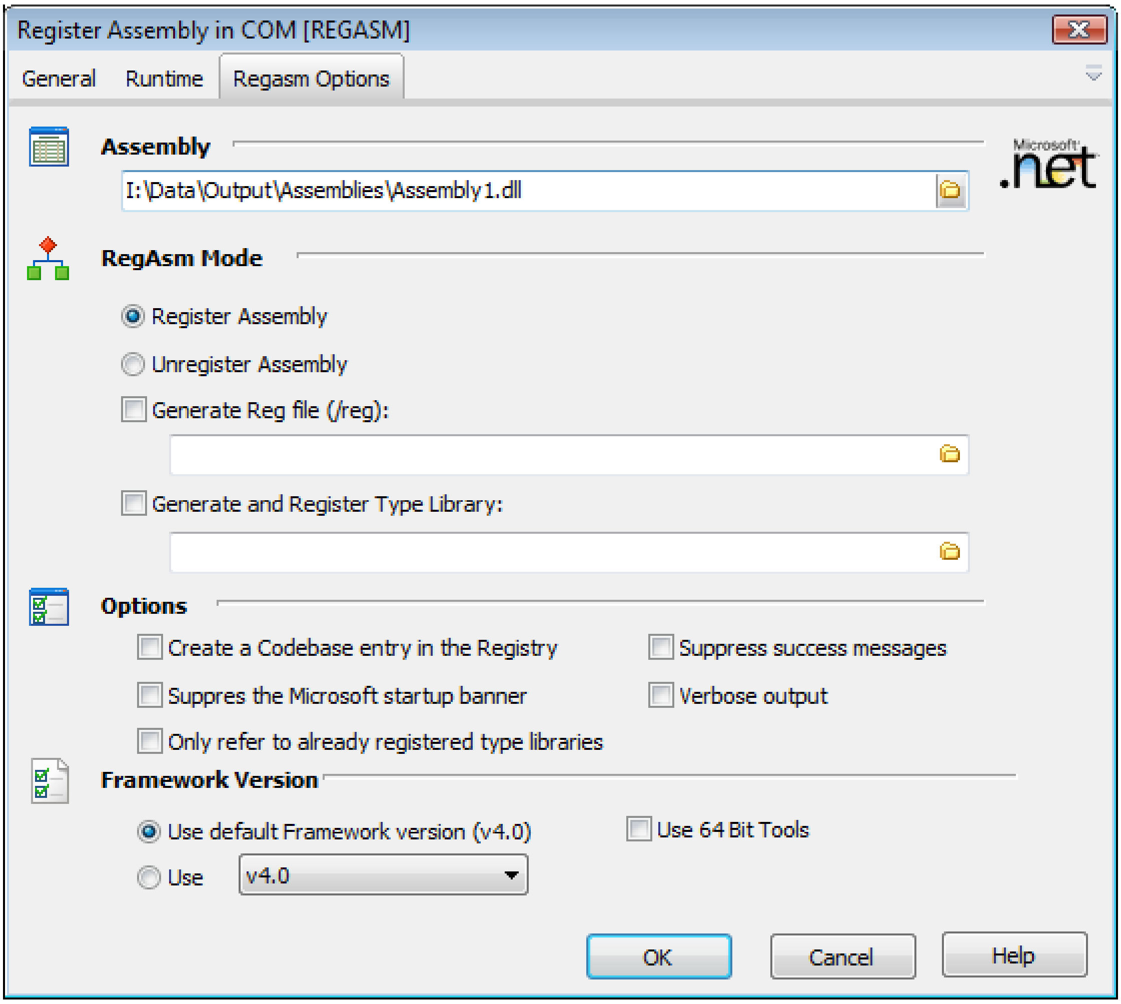
□ Allow partially trusted callers (-aptca)

□ Do not fully sign assembly at creation (-delaysign)

The compiled application assembly can be signed with a strong name from a key file or key container. If using a key container, the application can be set to allow partially trusted callers or to not fully sign the compiled assembly.

6.20.1.2 Register Assembly in COM [REGASM]

The Assembly Registration tool reads the metadata within an assembly and adds the necessary entries to the registry, which allows COM clients to create .NET Framework classes transparently. Once a class is registered, any COM client can use it as though the class were a COM class. The class is registered only once, when the assembly is installed. Instances of classes within the assembly cannot be created from COM until they are actually registered.



Register Assembly in COM [REGASM]

General | Runtime | Regasm Options

Assembly

I:\Data\Output\Assemblies\Assembly1.dll

RegAsm Mode

● Register Assembly

○ Unregister Assembly

□ Generate Reg file (/reg):

□ Generate and Register Type Library:

Options

□ Create a Codebase entry in the Registry

□ Suppress success messages

□ Suppres the Microsoft startup banner

□ Verbose output

□ Only refer to already registered type libraries

Framework Version

● Use default Framework version (v4.0)

□ Use 64 Bit Tools

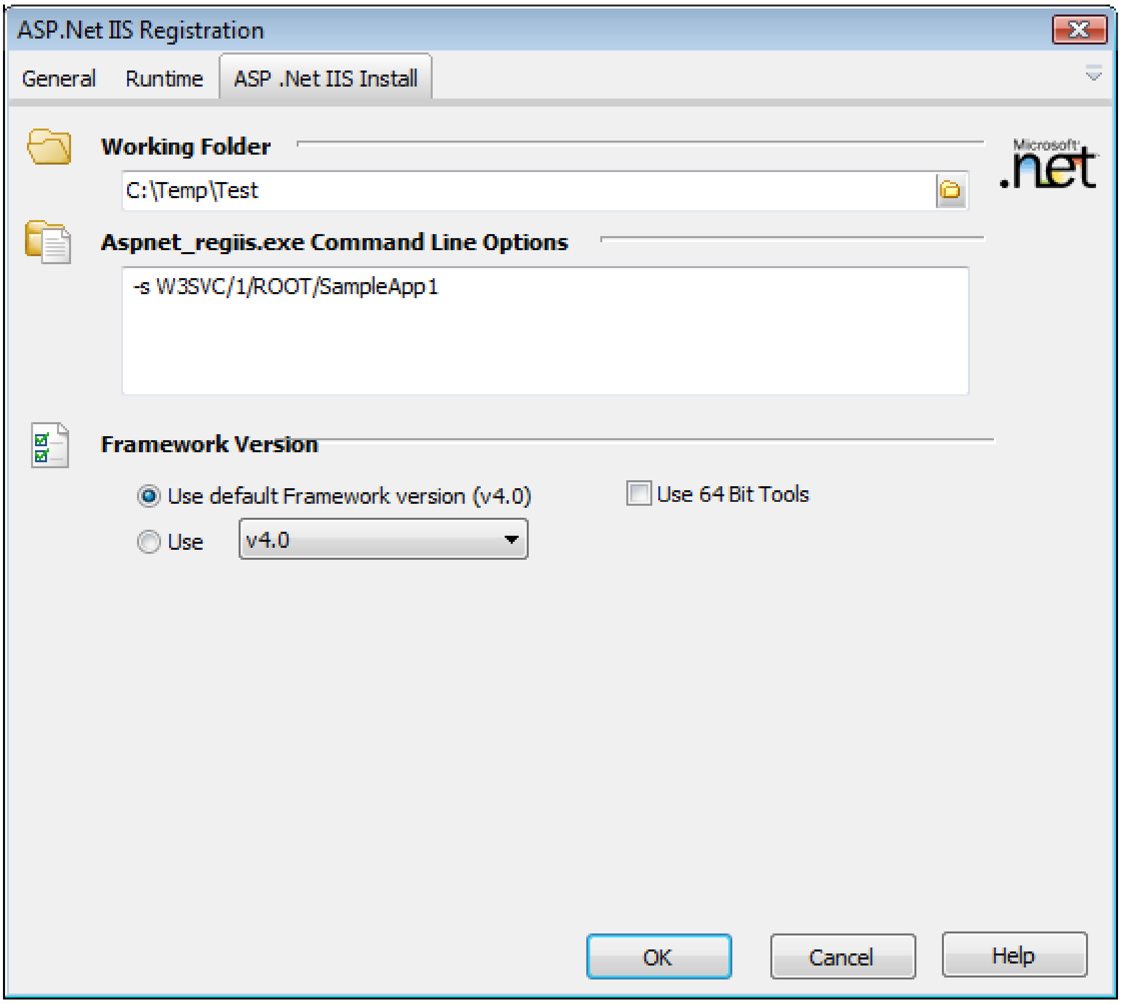
○ Use v4.0

For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfAssemblyRegistrationToolRegasmexe.asp

6.20.1.3 Run ASPNET_REGIIS.EXE

Allows an administrator or installation program to update the scriptmaps for an ASP.NET application to point to the ASP.NET ISAPI version associated with the tool. You can also use the tool to perform other ASP.NET configuration operations.



Working Folder

C:\Temp\Test

Aspnet_regiis.exe Command Line Options

-s W3SVC/1/ROOT/SampleApp1

Framework Version

● Use default Framework version (v4.0)

○ Use v4.0

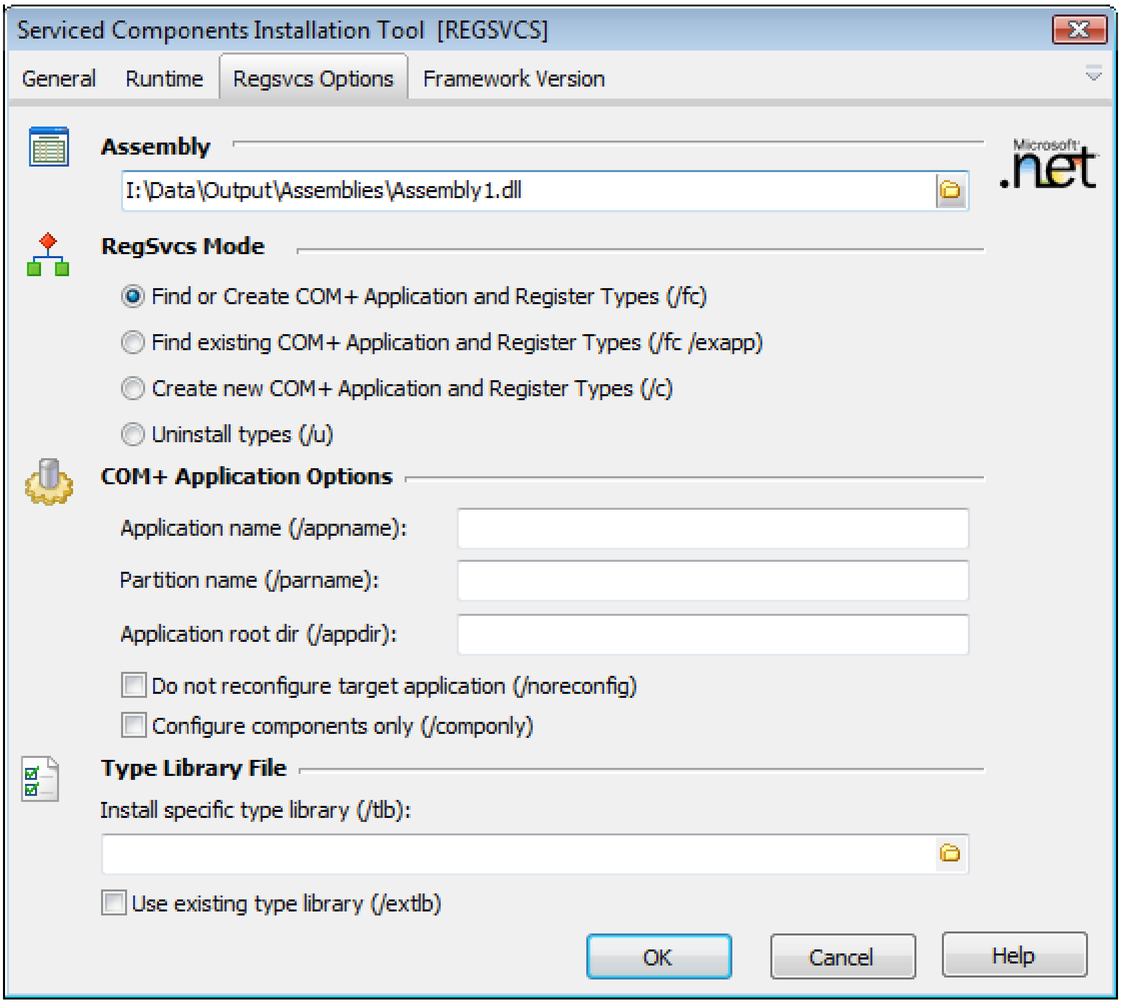
□ Use 64 Bit Tools

For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfASPNETIISRegistrationToolAspnet_regiisexe.asp

6.20.1.4 Serviced Components Installation Tool [REGSVCS]

The Serviced Components Installation Tool (regsvcs.exe) loads and registers serviced component classes from .NET assemblies into COM+ applications. You can also unregister assemblies.



Serviced Components Installation Tool [REGSVCS]

General | Runtime | Regsvcs Options | Framework Version

Assembly

I:\Data\Output\Assemblies\Assembly1.dll

RegSvcs Mode

● Find or Create COM+ Application and Register Types (/fc)

○ Find existing COM+ Application and Register Types (/fc /exapp)

○ Create new COM+ Application and Register Types (/c)

○ Uninstall types (/u)

COM+ Application Options

Application name (/appname):

Partition name (/parname):

Application root dir (/appdir):

□ Do not reconfigure target application (/noreconfig)

□ Configure components only (/componly)

Type Library File

Install specific type library (/tlb):

□ Use existing type library (/extlb)

Assembly

The name of the assembly to install/uninstall from.

RegSvcs Mode:

"Find or create COM+ Application and Register Type (/fc)"

If a COM+ Application matching the assembly (or specified name) is found, it will be used. Otherwise, a new application will be created.

"Find existing COM+ Application and Register Types (/fc /exapp)"

If a COM+ Application matching the assembly (or specified name) is found, it will be used. Otherwise, the action will fail.

"Create new COM+ Application and Register Types (/c)"

A new COM+ Application will be created. If a matching COM+ Application already exists, the action will fail.

"**Uninstall types (/u)**"

Remove a previously registered assembly.

COM+ Application Options:

Application Name (/appname)

Specify a name for the COM+ Application to register or unregister (or ID of an application to unregister.) If no name is specified, the assembly file name will be used for the application name.

Partition Name (/parname)

Specify the name or ID of a COM+ partition in which to locate the assembly.

Application root directory (/appdir)

Specify a root directory for the COM+ Application.

"**Do not reconfigure target application (/noreconfig)**"

Only applies when registering components. Default behaviour is to reconfigure.

"**Configure components only (/componly)**"

Only applies when registering components. If this option is set, only components (not methods or interfaces) will be reconfigured.

Type Library File

It is possible to specify a name (or path) to a specific type library file to generate/use.

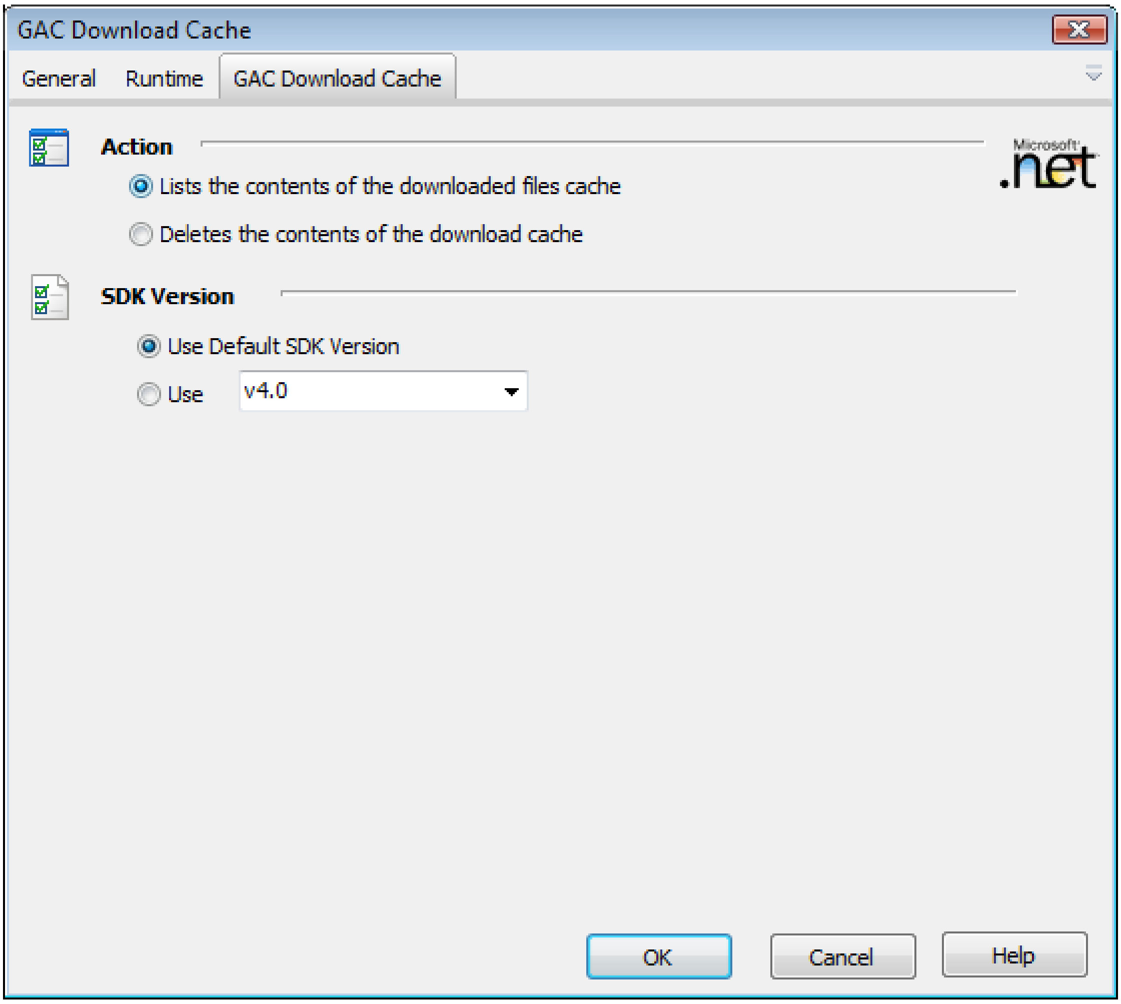
If "**Use existing type library**" is checked, the type library will not be regenerated.

6.20.2 .Net SDK Tools

The .Net SDK tools are installed as part of the .Net SDK, please check if you have the SDK installed, the standard place is C:\Program Files\Microsoft.NET\SDK

6.20.2.1 GAC Download Cache [GACUTIL]

Lists or deletes the contents of the downloaded files cache.



Action

● Lists the contents of the downloaded files cache

○ Deletes the contents of the download cache

SDK Version

● Use Default SDK Version

○ Use v4.0

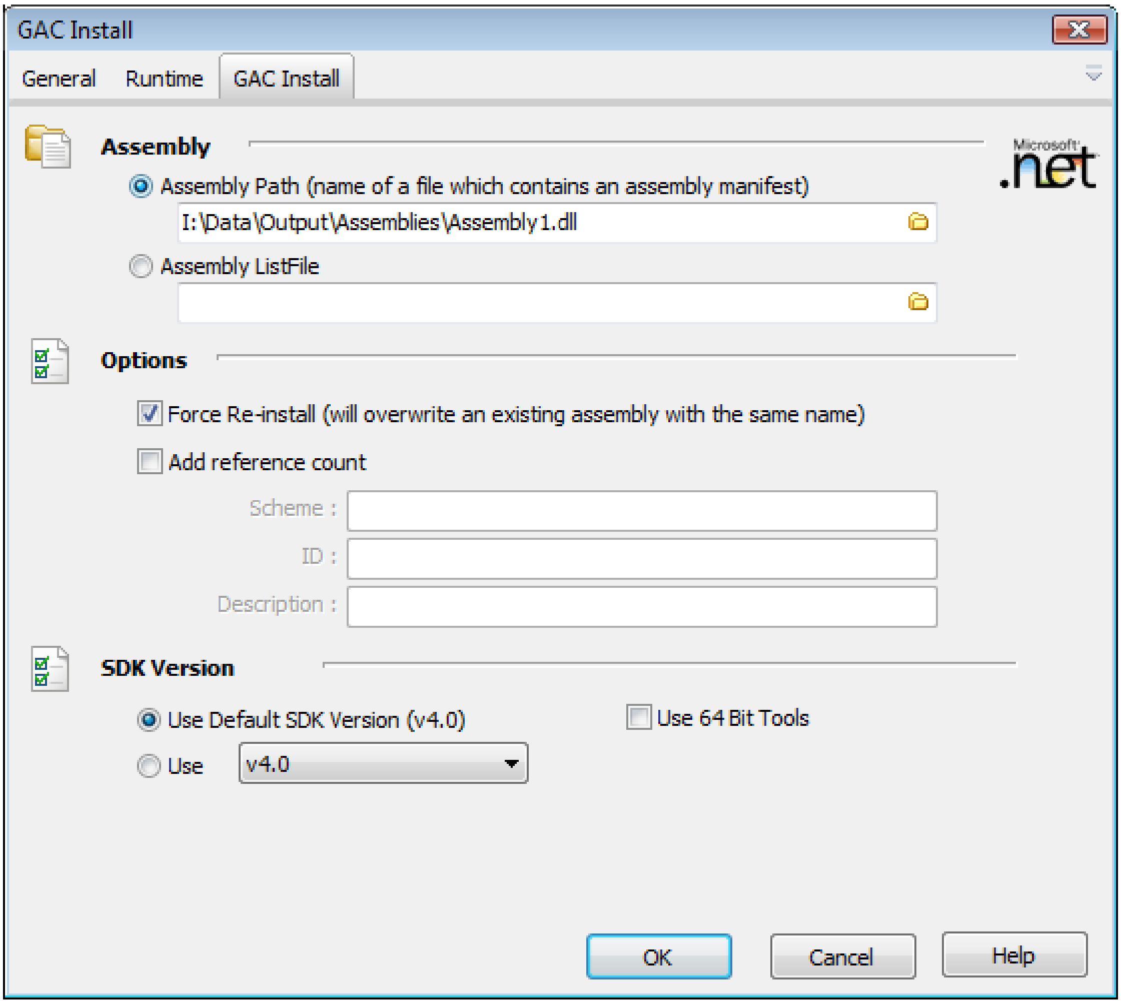
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutilexe.asp

6.20.2.2 GAC Install [GACUTIL]

Installs an assembly into the global assembly cache.

This action automates the following options:
/i
/if
/il
/ir



GAC Install

Assembly

● Assembly Path (name of a file which contains an assembly manifest)

I:\Data\Output\Assemblies\Assembly1.dll

○ Assembly ListFile

Options

☑ Force Re-install (will overwrite an existing assembly with the same name)

□ Add reference count

Scheme :

ID :

Description :

SDK Version

● Use Default SDK Version (v4.0)

□ Use 64 Bit Tools

○ Use v4.0

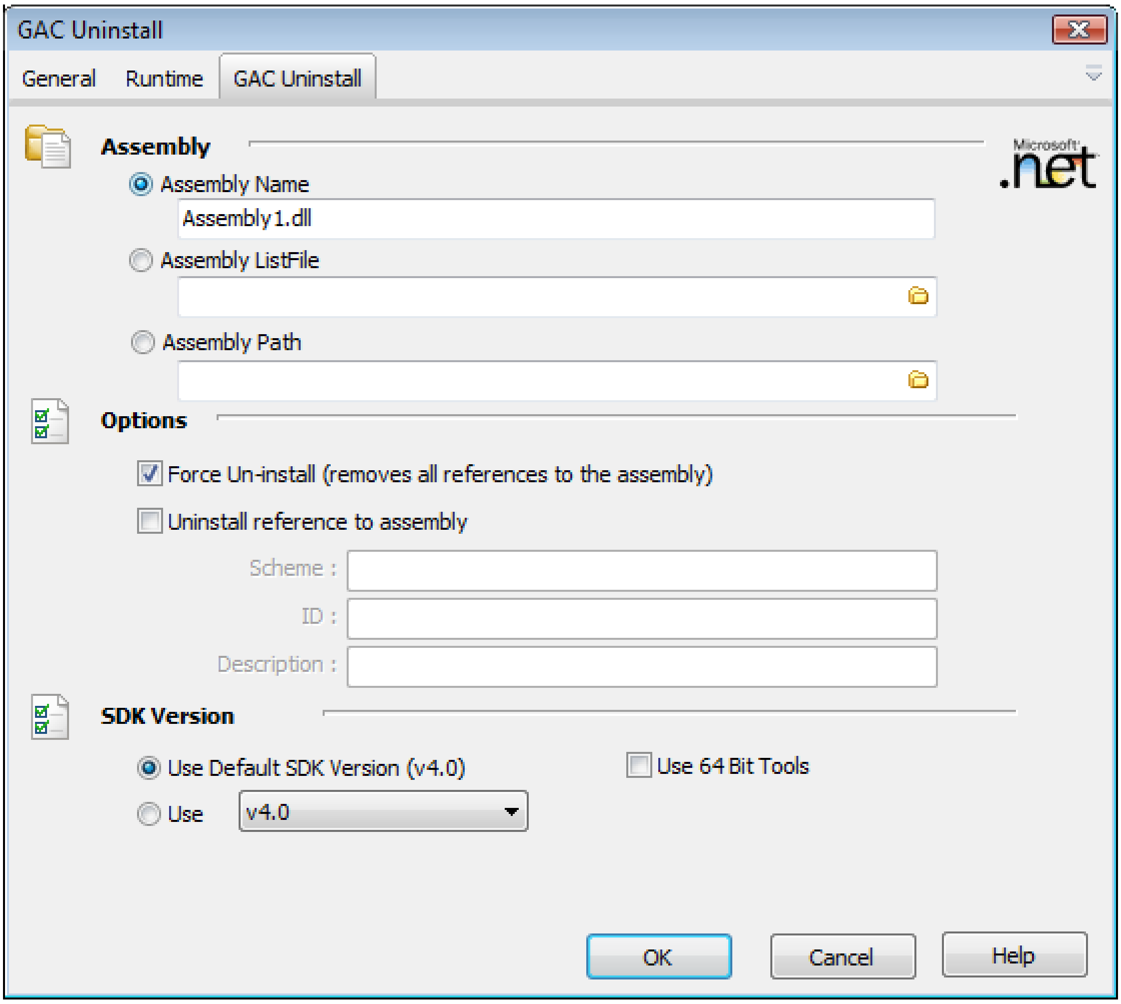
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutilexe.asp

6.20.2.3 GAC Uninstall [GACUTIL]

Uninstalls an assembly from the global assembly cache.

This action automates the following options:
/u
/uf
/ul
/ur



Assembly

● Assembly Name: Assembly1.dll

○ Assembly ListFile

○ Assembly Path

Options

☑ Force Un-install (removes all references to the assembly)

□ Uninstall reference to assembly

Scheme :

ID :

Description :

SDK Version

● Use Default SDK Version (v4.0)

□ Use 64 Bit Tools

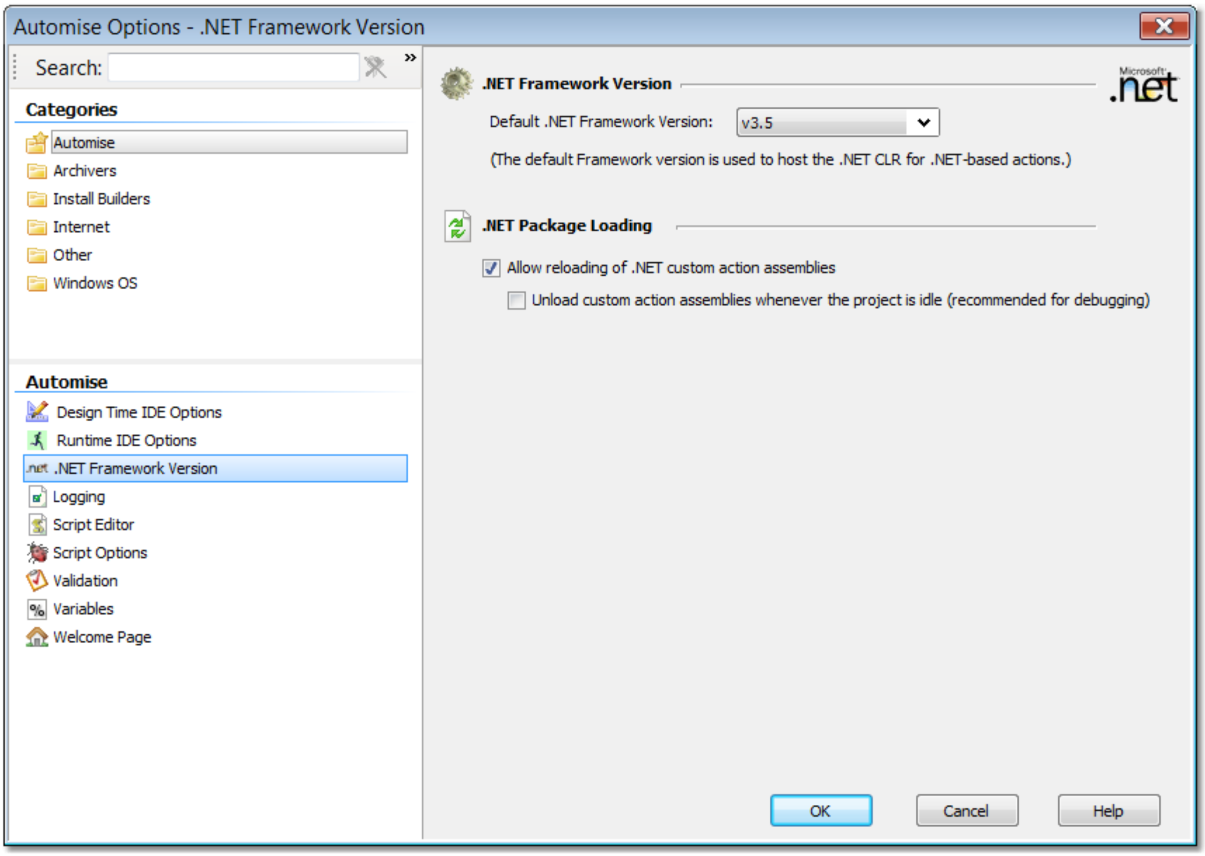
○ Use v4.0

For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutilexe.asp

6.20.3 .NET Framework Options

The .NET Framework Version options control settings used by .NET actions.



Automise Options - .NET Framework Version

.NET Framework Version

Default .NET Framework Version: v3.5

(The default Framework version is used to host the .NET CLR for .NET-based actions.)

.NET Package Loading

☑ Allow reloading of .NET custom action assemblies

□ Unload custom action assemblies whenever the project is idle (recommended for debugging)

Default .NET Framework Version

Specifies the default for new .NET actions added to your build. Most .NET actions let you then override that default.

Allow reloading of .NET custom action assemblies

Checking this option allows Automise to reload assemblies required by custom action implemented in .NET. Reloading will take place automatically when an action package is saved in ActionStudio, or from the Package Manager dialog.

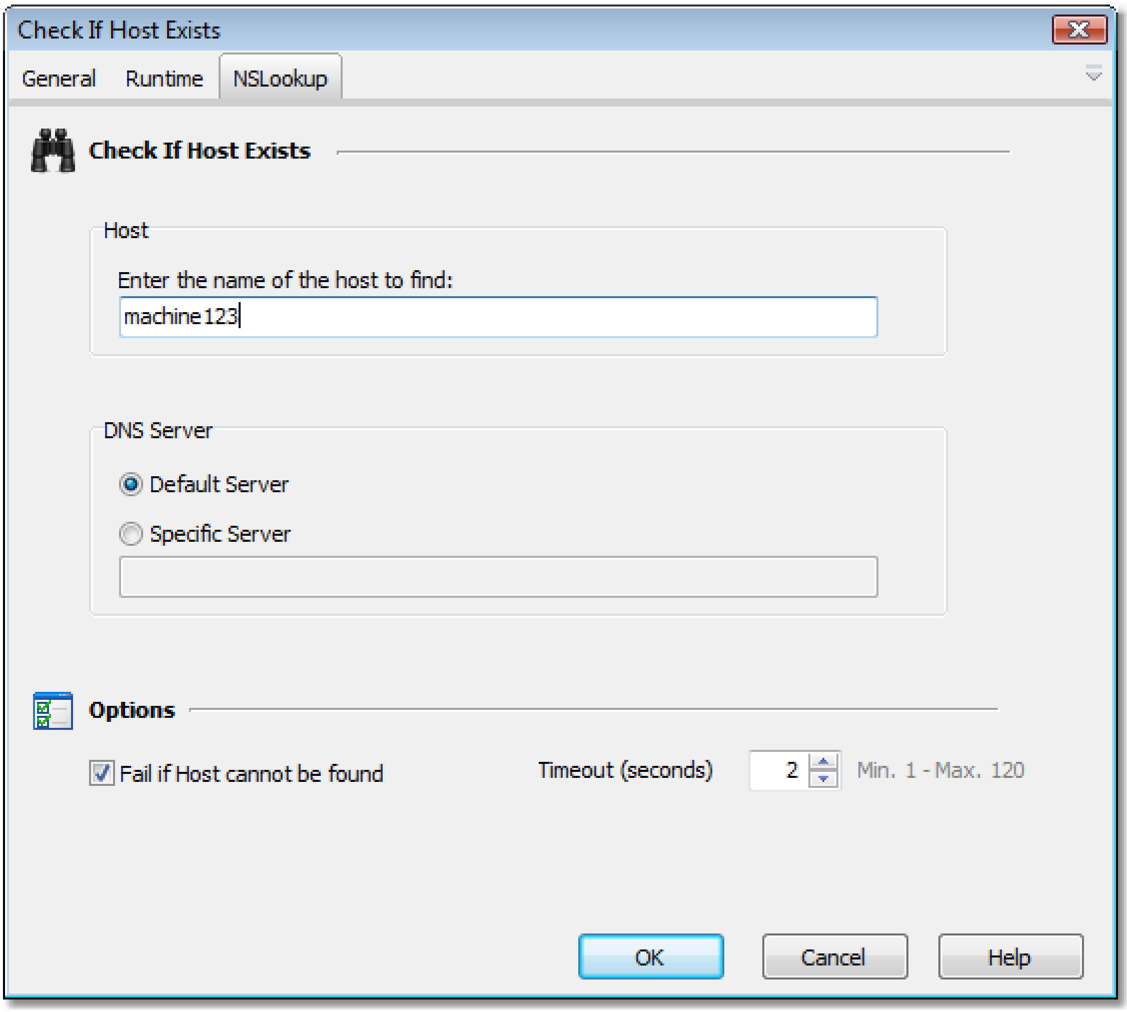
Unload custom action assemblies whenever the project is idle

Checking this option means that Automise will unload custom action assemblies when Automise is idle. For example, after a build have completed or when the action properties dialog is closed.

6.21 Network

6.21.1 Check If Host Exists

The Check If Host Exists action allows you to query a DNS server to determine if a host exists.



Check If Host Exists

Host

Enter the name of the host to find: machine123

DNS Server

● Default Server

○ Specific Server

Options

☑ Fail if Host cannot be found

Timeout (seconds) 2 Min. 1 - Max. 120

To determine if a host exists:

1. Specify the name of the host that you want to check.

2. Choose a DNS server to query:

   - Default Server - selecting this option will query the server which is defined as the current machine's DNS server.

   - Specific Server - specify the DNS server to query.

3. Set the options:

   - Fail if Host cannot be found- The action will fail if the host is not found. If not selected, the action will pass regardless (a warning will be raised if the host is not found).

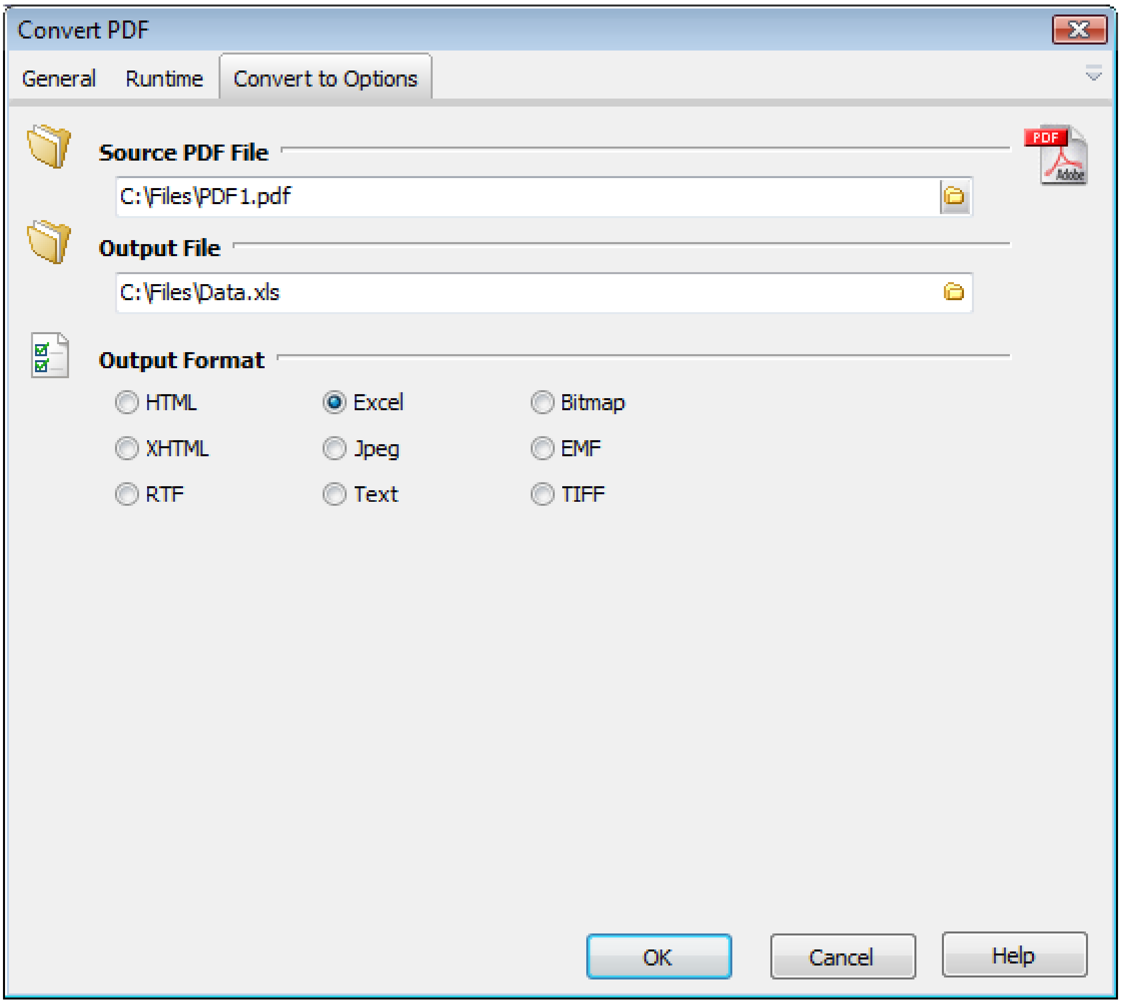
   - Timeout - Specify the timeout value at which point the query will be abandoned if the host has not been found.

6.22 PDF Files

6.22.1 Convert PDF

[Automise Professional Edition]

The Convert PDF action will convert a PDF file into one of the following formats: HTML, Excel, Bitmap, XHTML, Jpeg, EMF, RTF, Text or TIFF.



Source PDF File: C:\Files\PDF1.pdf

Output File: C:\Files\Data.xls

Output Format

○ HTML ● Excel ○ Bitmap
○ XHTML ○ Jpeg ○ EMF
○ RTF ○ Text ○ TIFF

Source PDF File - specify the source PDF file to convert

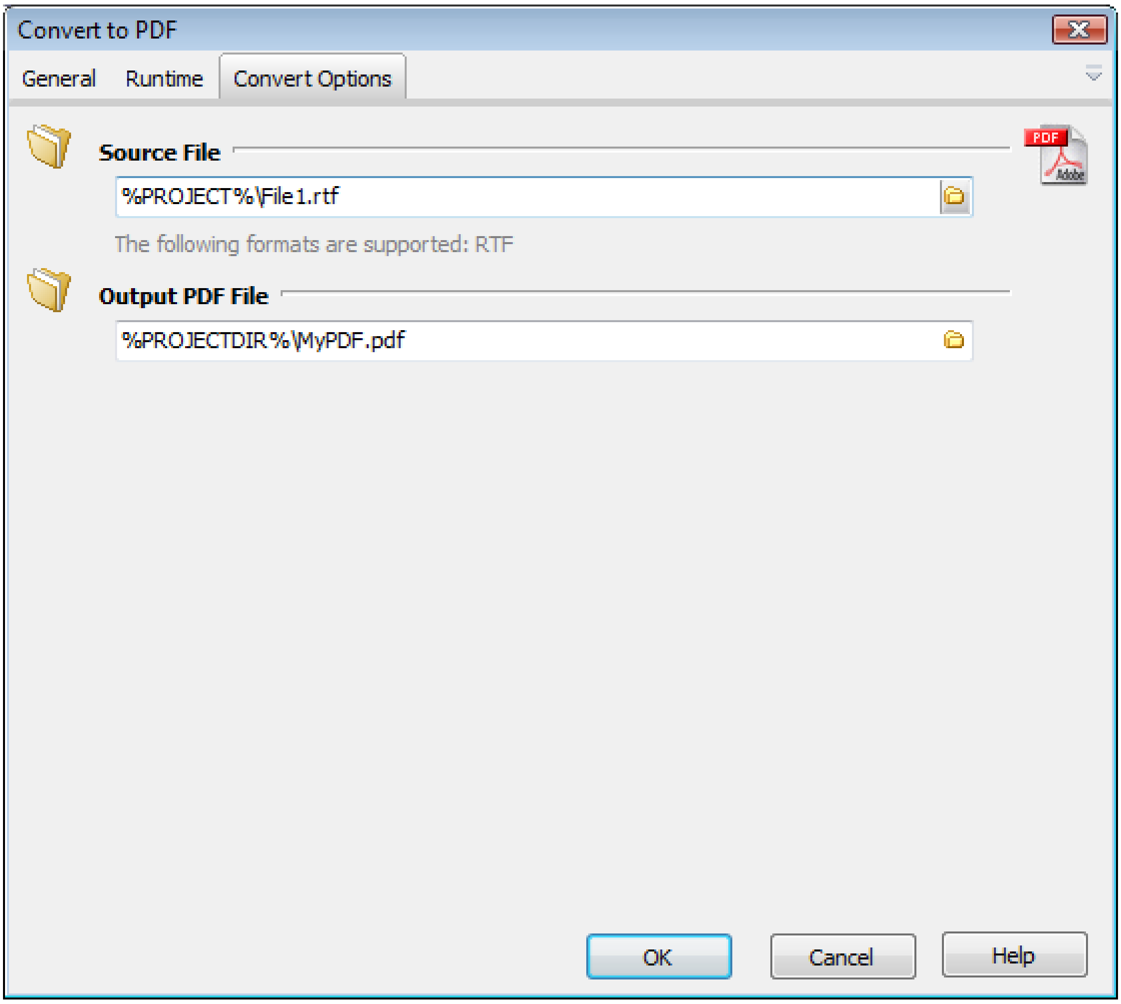
Output File - Specify the output filename

Output Format - specify the output format. Changing the format will automatically change the file extension of the output file. The valid formats to convert to are: HTML, Excel, Bitmap, XHTML, Jpeg, EMF, RTF, Text and TIFF.

6.22.2 Convert to PDF

[Automise Professional Edition]

The Convert to PDF action will convert an RTF (Rich Text Format) action into a PDF File.



Source File
%PROJECT%\File1.rtf
The following formats are supported: RTF

Output PDF File
%PROJECTDIR%\MyPDF.pdf

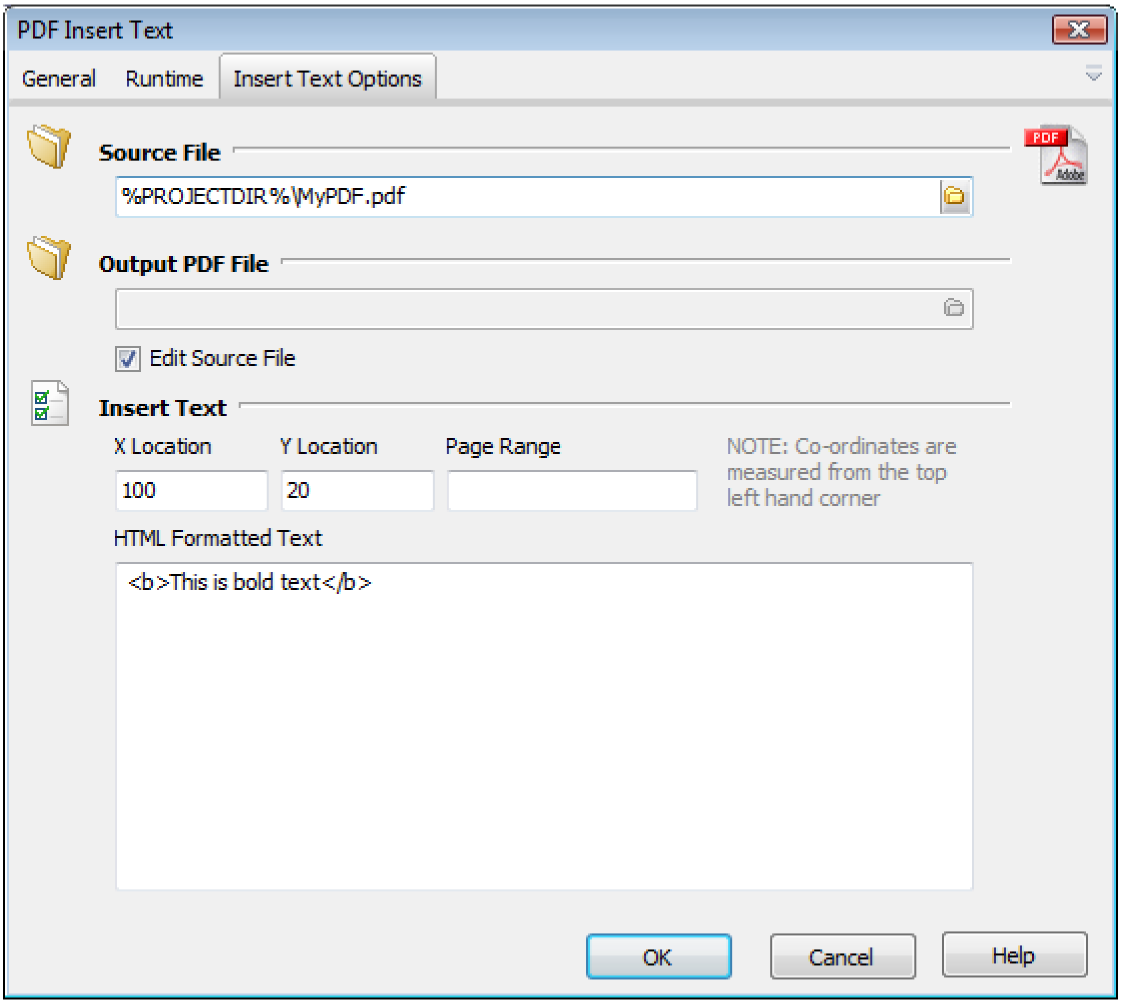
Source File - specify the RTF file to convert to PDF

Output PDF File - specify the PDF filename to write to

6.22.3 PDF Insert Text

[Automise Professional Edition]

The PDF Insert Text action will add HTML formatted text to a PDF file at the specified X, Y location.


PDF Insert Text

Source File

%PROJECTDIR%\MyPDF.pdf

Output PDF File

☑ Edit Source File

Insert Text

X Location	Y Location	Page Range
100	20	

NOTE: Co-ordinates are measured from the top left hand corner

HTML Formatted Text

```
<b>This is bold text</b>
```

Source File - specify the source PDF file

Output PDF File - specify the new PDF file if not editing the source file

Edit Source File - all changes made will be written back to the source file

X, Y Location - specify the location to insert the text. The location is the x, y co-ordinates measured from the top left hand corner of the page. The X, Y values must be a valid integer or floating point number.

Page Range - specify the page range to write the text to. The format is <page number> or <page start>-<page end>. If left blank, page 1 will be assumed. To specify all pages after the specified page, leave the <page end> blank, eg. 1- specifies all pages. Some examples are: "3" - just write to page 3. "2-45" - write to pages 2 to 45 inclusive. "5-" - write to all pages from 5 until the end of the document.

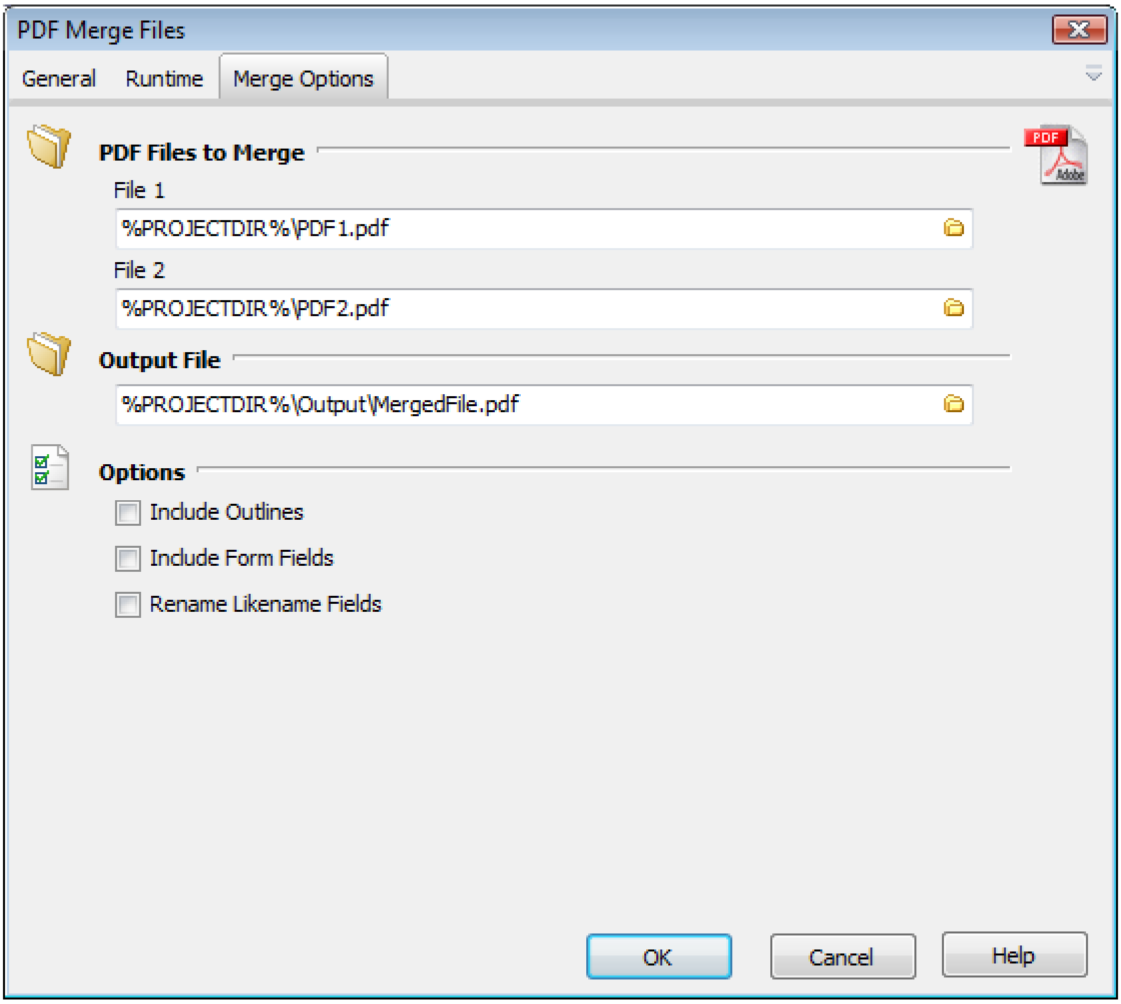
HTML Formatted Text - specify the text to write out formatted using HTML. The HTML tags supported are:

Bold	<b>....</b>
Italic	<i>....</i>
Underlined	<u>....</u>

6.22.4 PDF Merge Files

[Automise Professional Edition]

The PDF Merge Files action will merge two PDF files into a new single PDF File.



File 1
%PROJECTDIR%\PDF1.pdf

File 2
%PROJECTDIR%\PDF2.pdf

Output File
%PROJECTDIR%\Output\MergedFile.pdf

☐ Include Outlines
☐ Include Form Fields
☐ Rename Likename Fields

File 1, File 2 - The files to merge

Output File - the name of the new PDF file

Include Outlines - Include the outlines from the source files in the new PDF file

Include Form Fields -Include Form Fields from the source files in the new PDF file

Rename Likename Fields - Rename any field names which are the same in the two source files in the merged file.

6.23 Stack and Queue Actions

[Automise Professional Edition]

The Stack and Queue actions allow you to build up a list of items and then access the items in either a first in, first out (Queue) or first in, last out (Stack) manner.

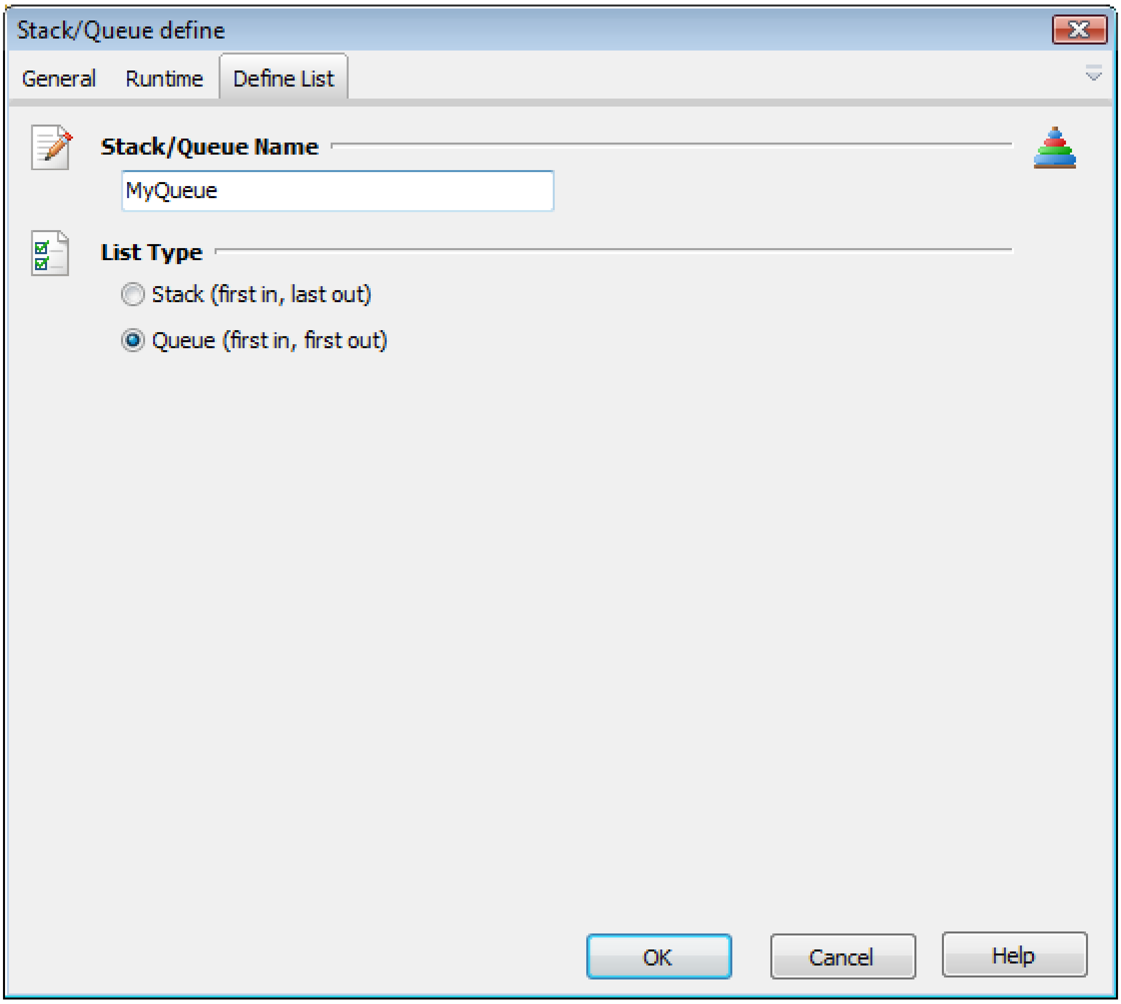
Firstly, use the Define action to create a new list, and then use the other actions to push

, pop, iterate, clear, etc. items on the list.

6.23.1 Stack/Queue Define

[Automise Professional Edition]

The Stack/Queue Define action is used for creating a new list and giving it a name.



Stack/Queue Name: MyQueue

List Type
○ Stack (first in, last out)
● Queue (first in, first out)

Stack/Queue Name - the name you want to give the list. The other Stack and Queue actions refer to the list by name.

List Type - Stack (first in, last out), or Queue (first in, first out). The type determines in what order the items in the list will be popped or iterated.

Stack/Queue examples
Say we have 3 items to put on the list, a, b, and c:

- Push a
- Push b
- Push c

If we pop the items off a Stack, we get:

- Pop c
- Pop b
- Pop a

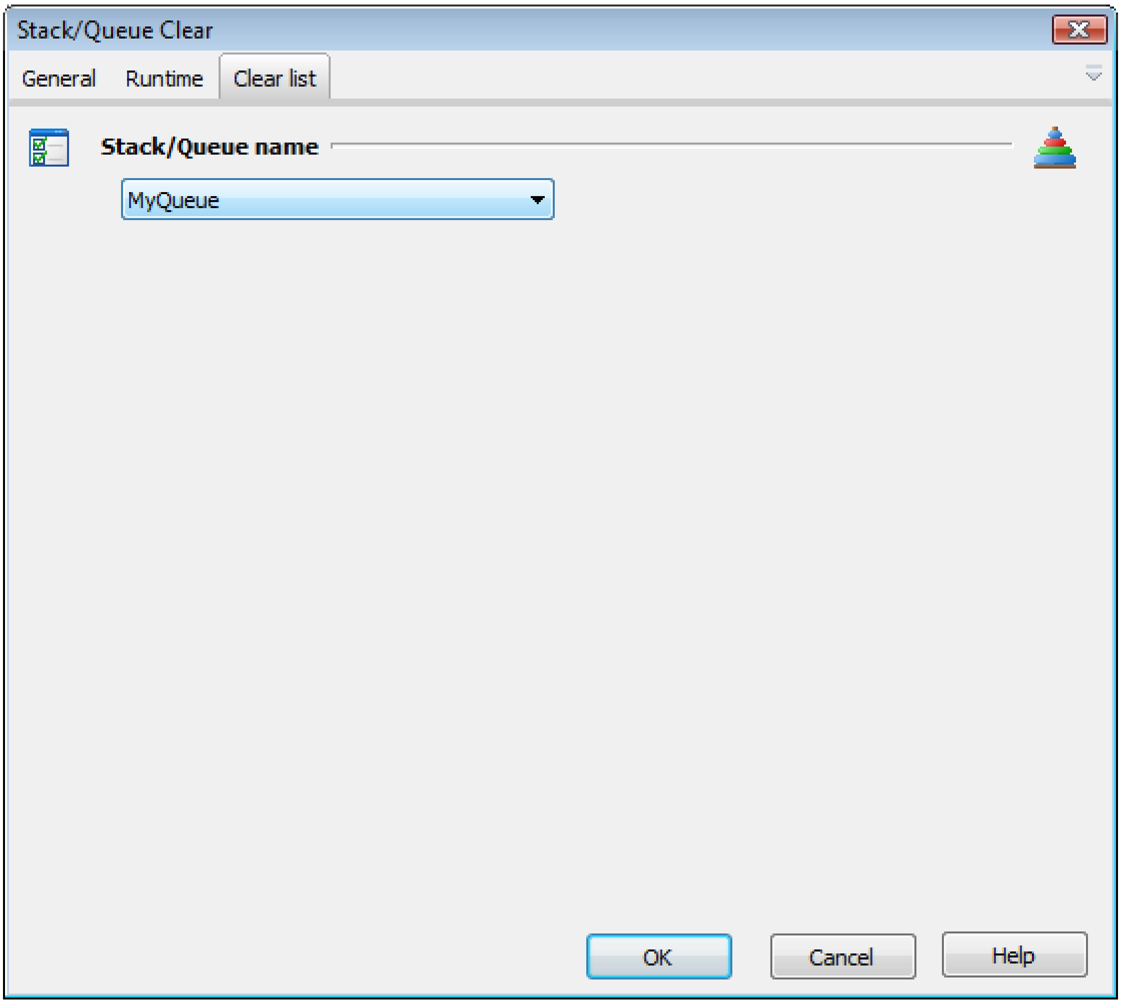
If we pop the items off a Queue, we get:

- Pop a
- Pop b
- Pop c

6.23.2 Stack/Queue Clear

[Automise Professional Edition]

The Stack/Queue Clear action is used to remove all values from the list.



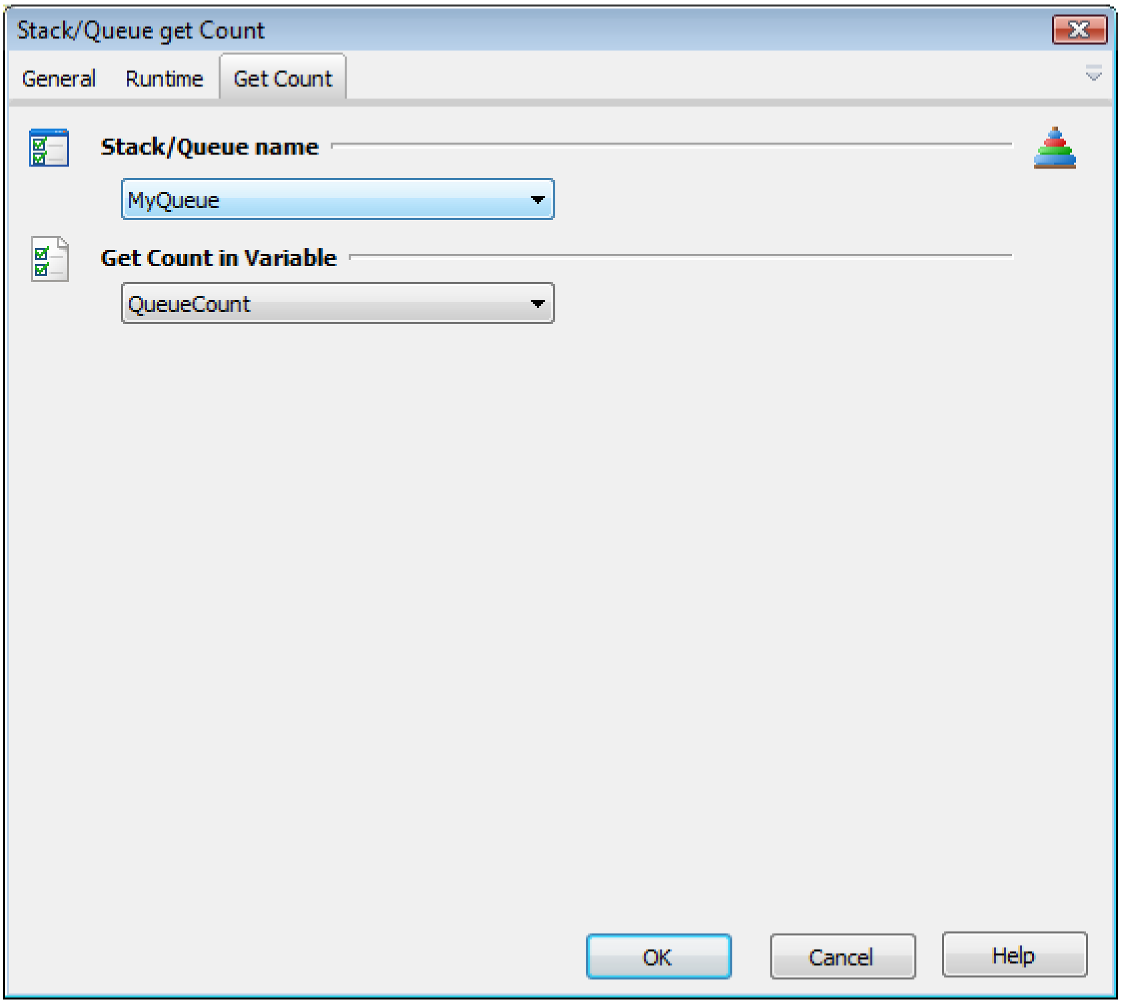
Stack/Queue name: MyQueue

Stack/Queue name - select the name of the list which you want to clear. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

6.23.3 Stack/Queue Get Count

[Automise Professional Edition]

The Stack/Queue Get Count action is used to get the amount of items that is currently on the list.



Stack/Queue name: MyQueue

Get Count in Variable: QueueCount

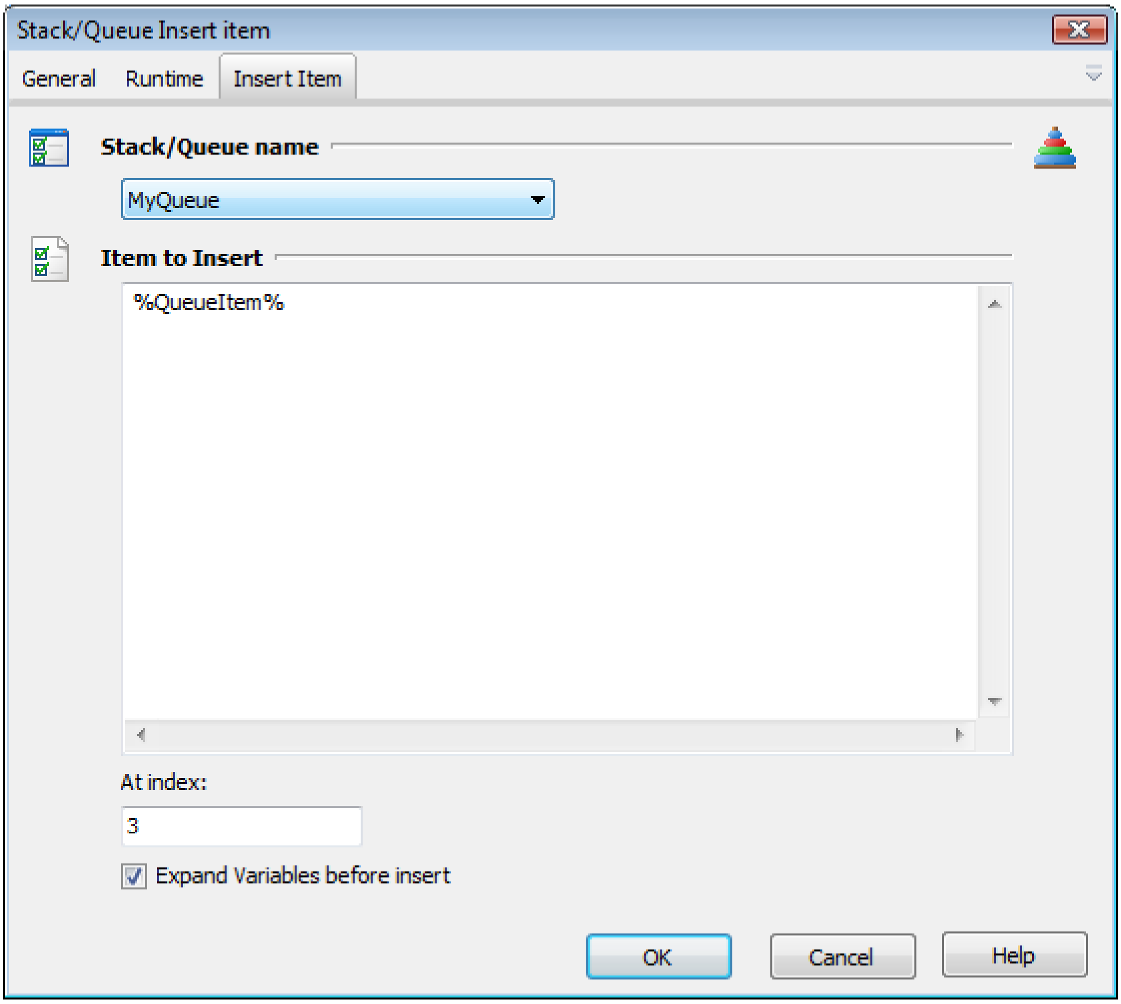
Stack/Queue name - select the name of the list which you want to get the count of items in. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Get Count in Variable - specify the variable to set the count to

6.23.4 Stack/Queue Insert Item

[Automise Professional Edition]

The Stack/Queue Insert Item action is used to push an item at a particular location in the list.



Stack/Queue name

MyQueue

Item to Insert

%QueueItem%

At index:

3

☑ Expand Variables before insert

Stack/Queue name - select the name of the list which you want to add new items to. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Item to Insert - the text of the item to insert into the list

At Index - the index to insert the item. Specifying an invalid index will cause the action to fail. The index is zero based, see NOTE below.

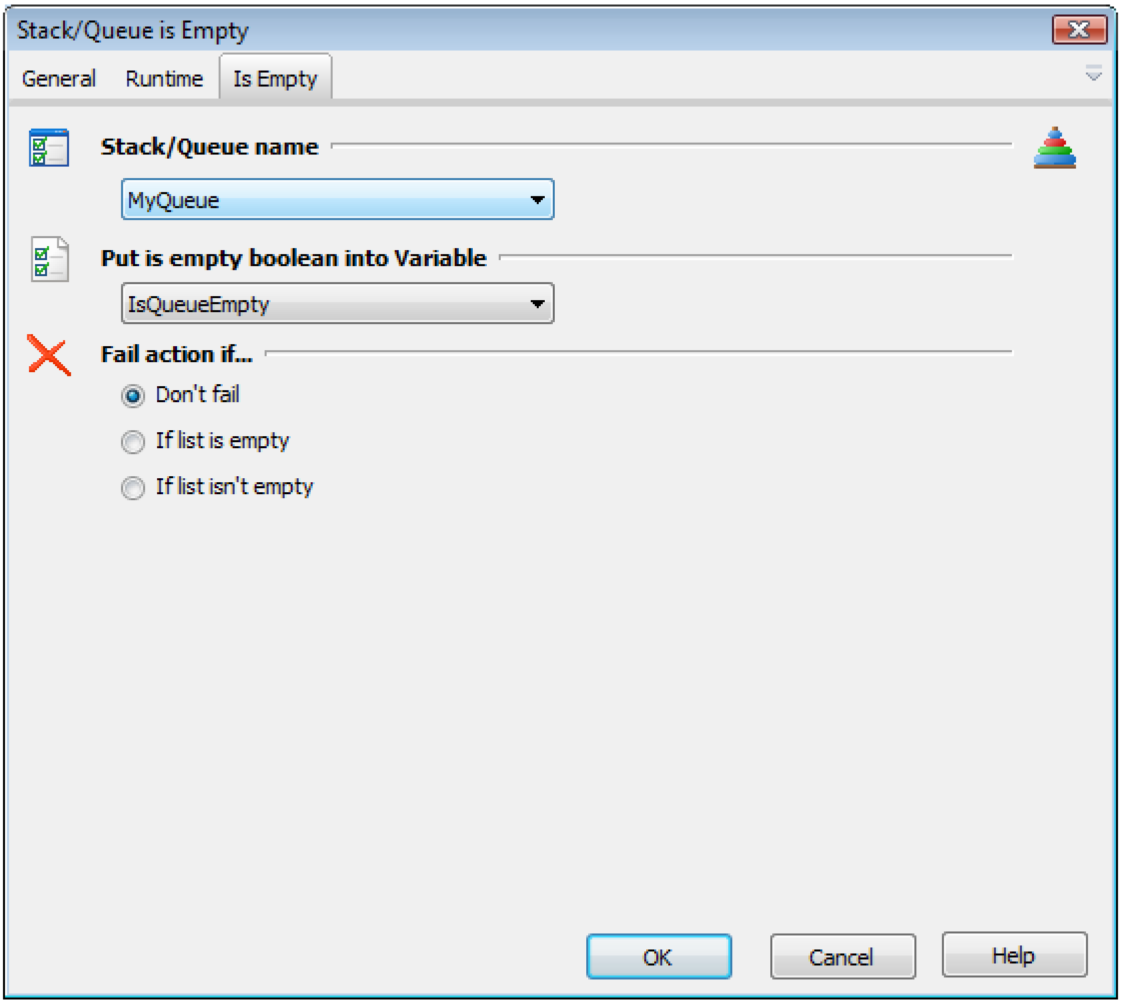
Expand variables before insert - if you specify variables as part of the text to push, eg. %MyVariable% then selecting this option will expand the variables before the item is pushed onto the list.

NOTE: Using this action requires you to understand the underlying implementation. Internally both queues and stacks are represented by a zero based array. Items are always popped from element 0. A stack pushes new items at element 0, whereas a queue adds new elements to the end of the array.

6.23.5 Stack/Queue Is Empty

[Automise Professional Edition]

The Stack/Queue Is Empty action is used to test if the list contains any items or not.



Stack/Queue is Empty

Stack/Queue name: MyQueue

Put is empty boolean into Variable: IsQueueEmpty

Fail action if...

- ● Don't fail
- ○ If list is empty
- ○ If list isn't empty

Stack/Queue name - select the name of the list which you want to test. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

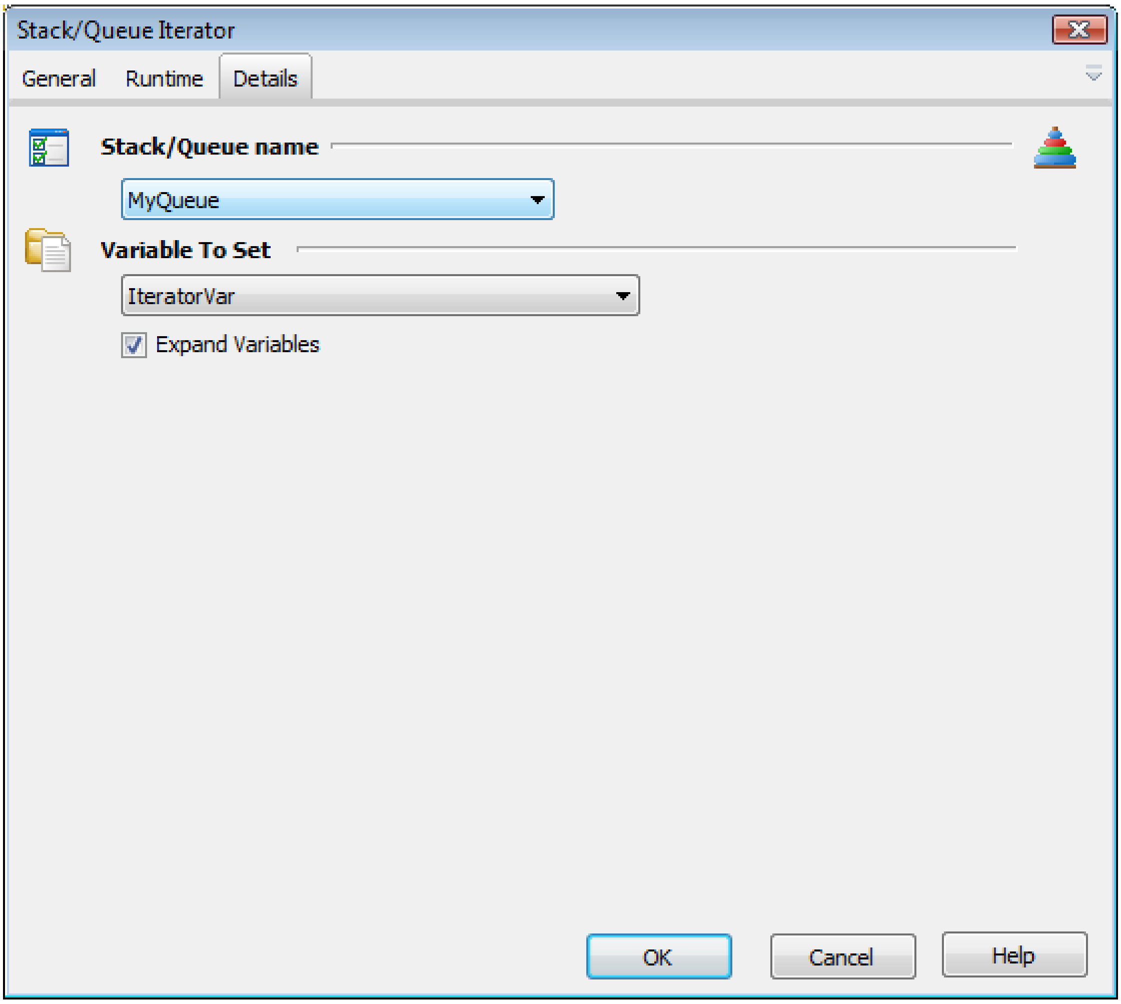
Put is empty boolean into Variable - specify a variable to set to a boolean indicating if the list is empty or not (True means list is empty).

Fail action if... - if you want the action to fail if the list is empty or if the list isn't empty, then set this property.

6.23.6 Stack/Queue Iterator

[Automise Professional Edition]

The Stack/Queue Iterator action is used to run a set of actions for each of the items in the list.



Stack/Queue Iterator

Stack/Queue name

MyQueue

Variable To Set

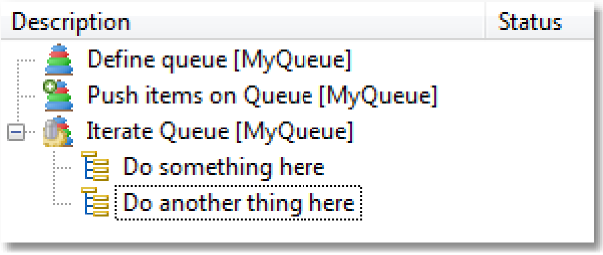
IteratorVar

☑ Expand Variables

Stack/Queue name - select the name of the list which you want to iterate. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Variable To Set - specify the variable to set to the current items for the iteration.

Below is an example of how to use the iterator.



Description | Status

Define queue [MyQueue]
Push items on Queue [MyQueue]
Iterate Queue [MyQueue]
Do something here
Do another thing here

After the queue is defined, and some items have been added to it, the iterator action will execute the "Do something" and "Do another thing here" actions for every item found in the list. For the child actions to get access to the current item from the iterator, they need to use the variable used in the iterator action.

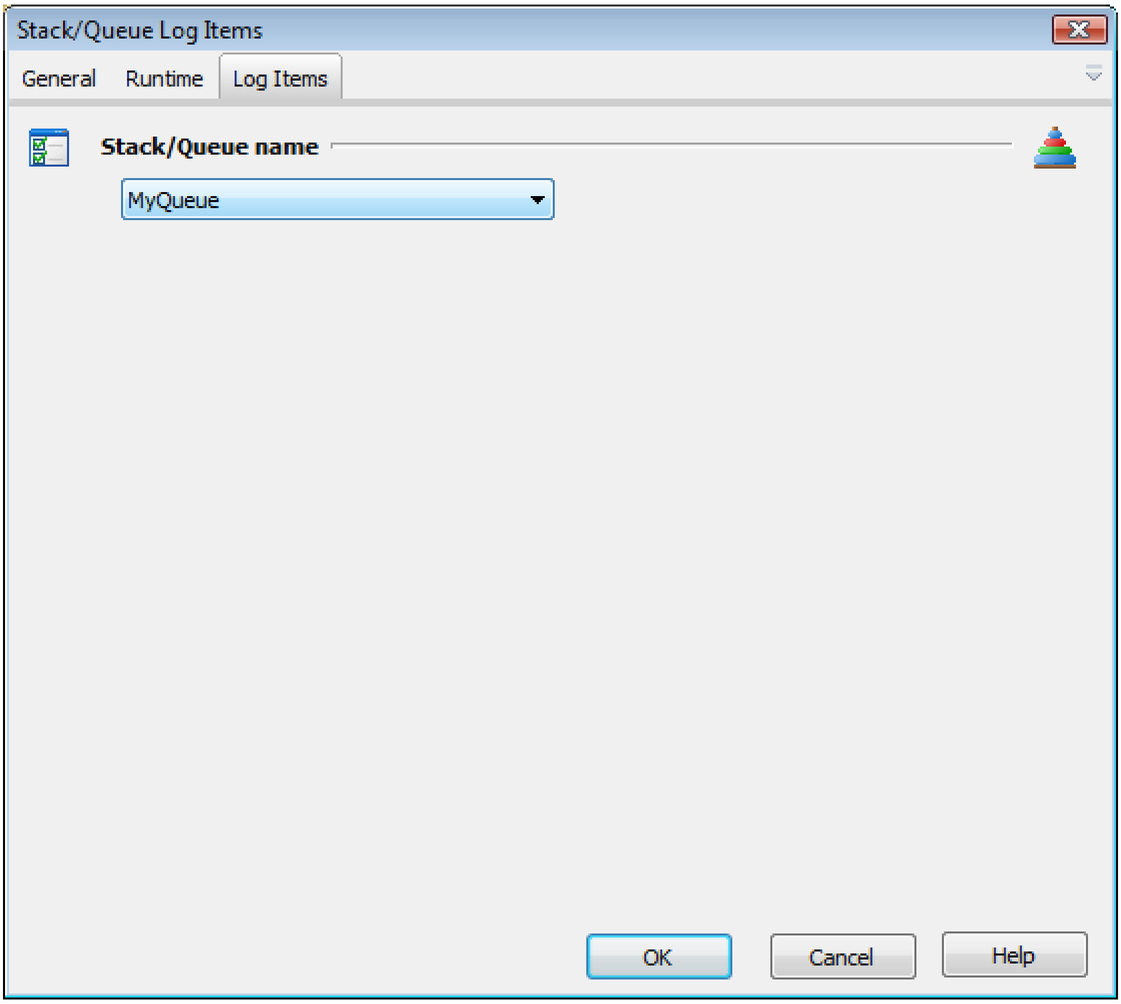
If the list includes the following values: "a" and "b". Then, the Iterator Variable will be set to "a" and then the "Do..." actions will be executed. And then for the next iteration, Iterator Variable will be set to "b" and then the "Do..." actions run again.

After the iterator action runs, the list will be empty.

6.23.7 Stack/Queue Log Items

[Automise Professional Edition]

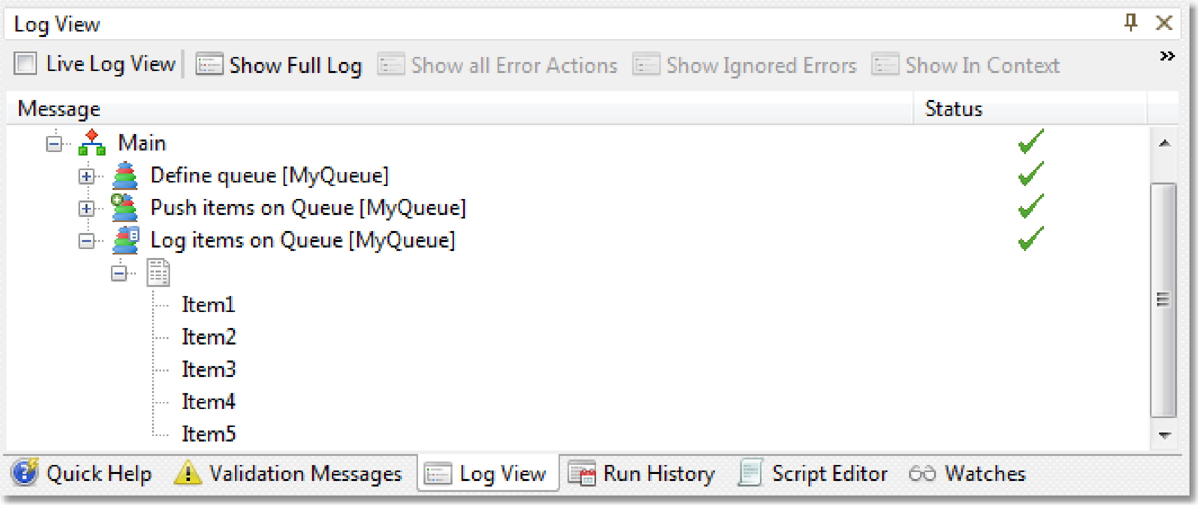
The Stack/Queue Log Items action is used to output all the items in the list into the log (the action does not modify the list in any way).



Stack/Queue name: MyQueue

Stack/Queue name - select the name of the list which you want to log the values of. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Below is an example of the log output of a queue that contains 3 items, "a", "b", and "c".



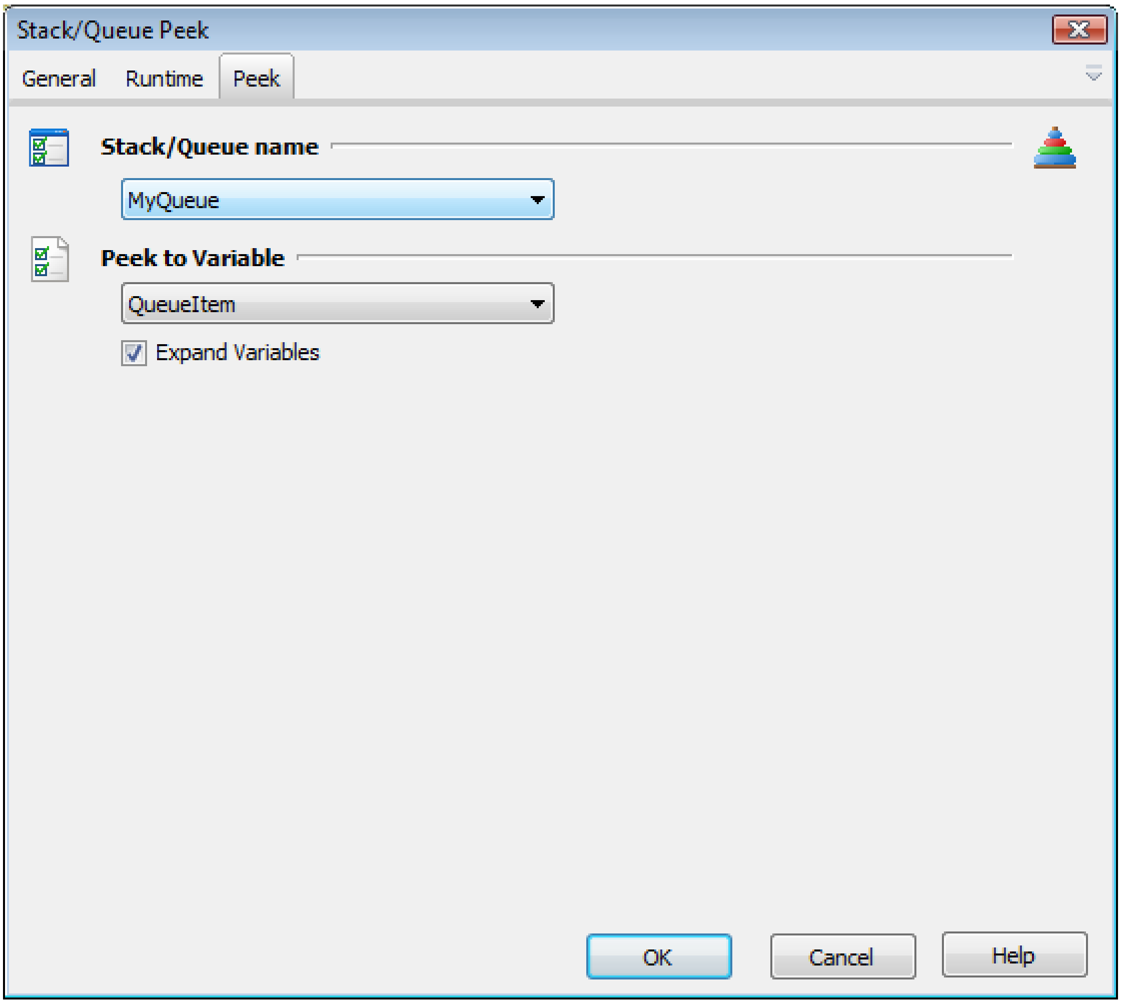
Log View

Message	Status
Main	✓
Define queue [MyQueue]	✓
Push items on Queue [MyQueue]	✓
Log items on Queue [MyQueue]	✓
Item1	
Item2	
Item3	
Item4	
Item5	

6.23.8 Stack/Queue Peek Item

[Automise Professional Edition]

The Stack/Queue Peek Item action is used to retrieve the next item from the list and set a variable to the value without removing the item from the list.



Stack/Queue Peek

Stack/Queue name: MyQueue

Peek to Variable: QueueItem

☑ Expand Variables

Stack/Queue name - select the name of the list which you want to peek. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Peek to Variable - select a Variable to set to the peeked value. The item will not be removed from the list.

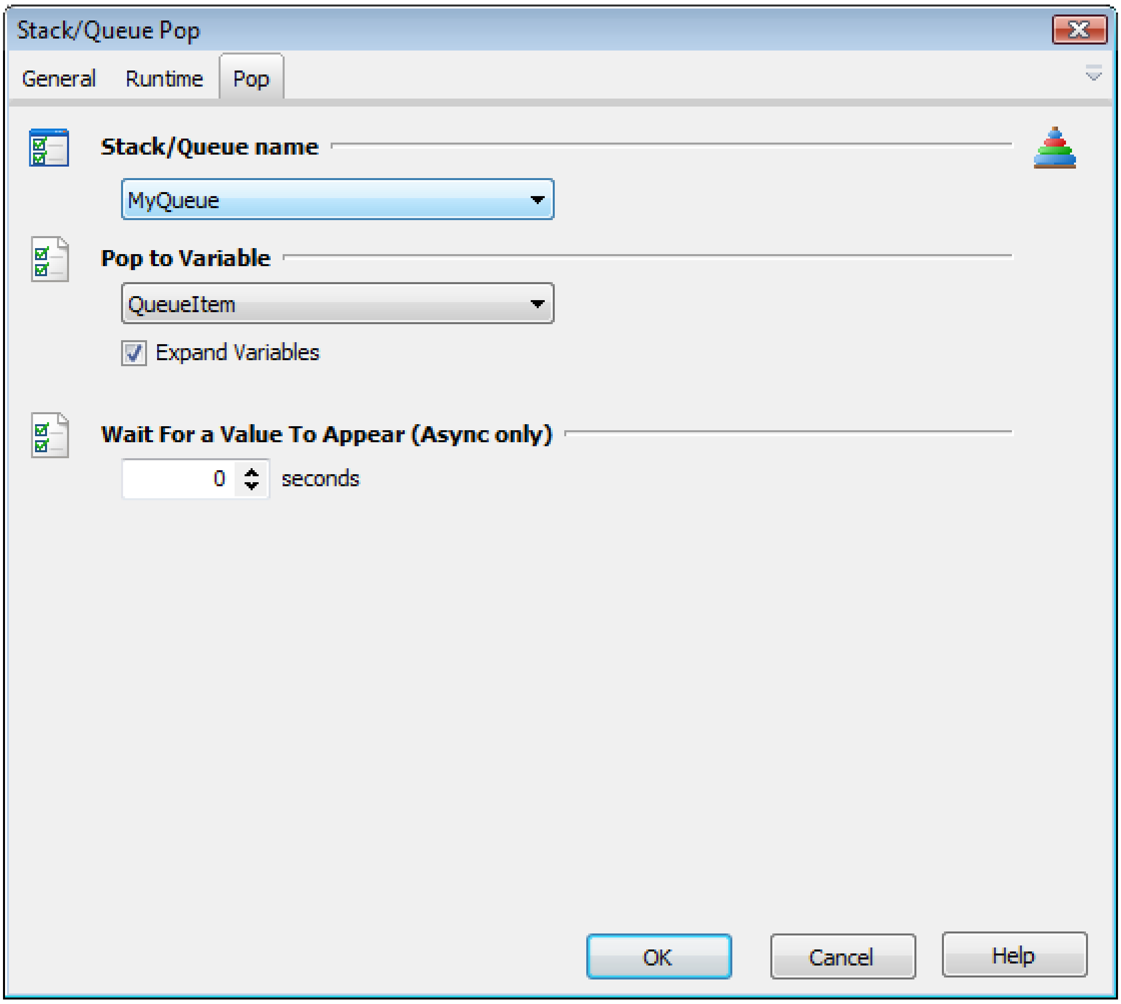
Expand variables - this will expand any variables in the item after it has been retrieved.

NOTE: If there are no more items to peek, then the action will fail.

6.23.9 Stack/Queue Pop Item

[Automise Professional Edition]

The Stack/Queue Pop Item action is used to remove the next item from the list and set a variable to the popped value.



Stack/Queue Pop

General Runtime Pop

Stack/Queue name: MyQueue

Pop to Variable: QueueItem

☑ Expand Variables

Wait For a Value To Appear (Async only): 0 seconds

Stack/Queue name - select the name of the list which you want to pop an item from. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Pop to Variable - select a Variable to set to the popped value.

Expand variables - this will expand any variables in the item after it has been popped.

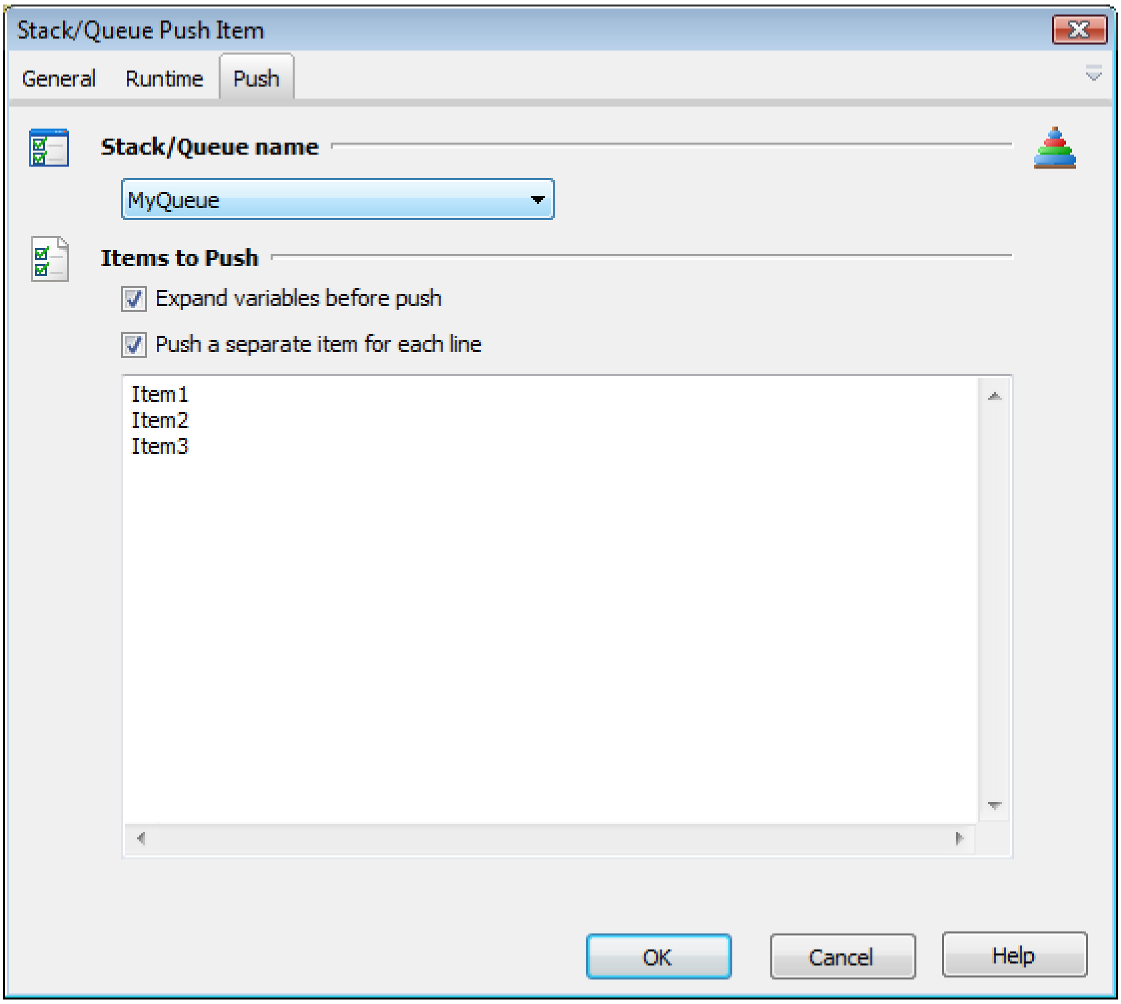
Wait For a Value to Appear - If there are no items to pop, then the action will wait for this amount of times for an item to appear in the queue. If no item appears within the specified time, the action will fail.

NOTE: If there are no more items to pop, then the action will fail.

6.23.10 Stack/Queue Push Item

[Automise Professional Edition]

The Stack/Queue Push Item action is used to push one or more items into the list.



Stack/Queue name: MyQueue

Items to Push

☑ Expand variables before push

☑ Push a separate item for each line

```
Item1
Item2
Item3
```

Stack/Queue name - select the name of the list which you want to add new items to. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Expand variables before push - if you specify variables as part of the text to push, eg. %MyVariable% then selecting this option will expand the variables before the item is pushed onto the list.

Push a separate item for each line - this option allows you to choose if you want a new item on the list for each line, or if you want multi-line items. In the above example, a separate item will be pushed on the queue for "a", "b", and "c". If this option was unchecked, a single item "a<crlf>b<crlf>c<crlf>" would be pushed onto the queue.

The Item - specify the item(s) to be pushed onto the list in the memo box

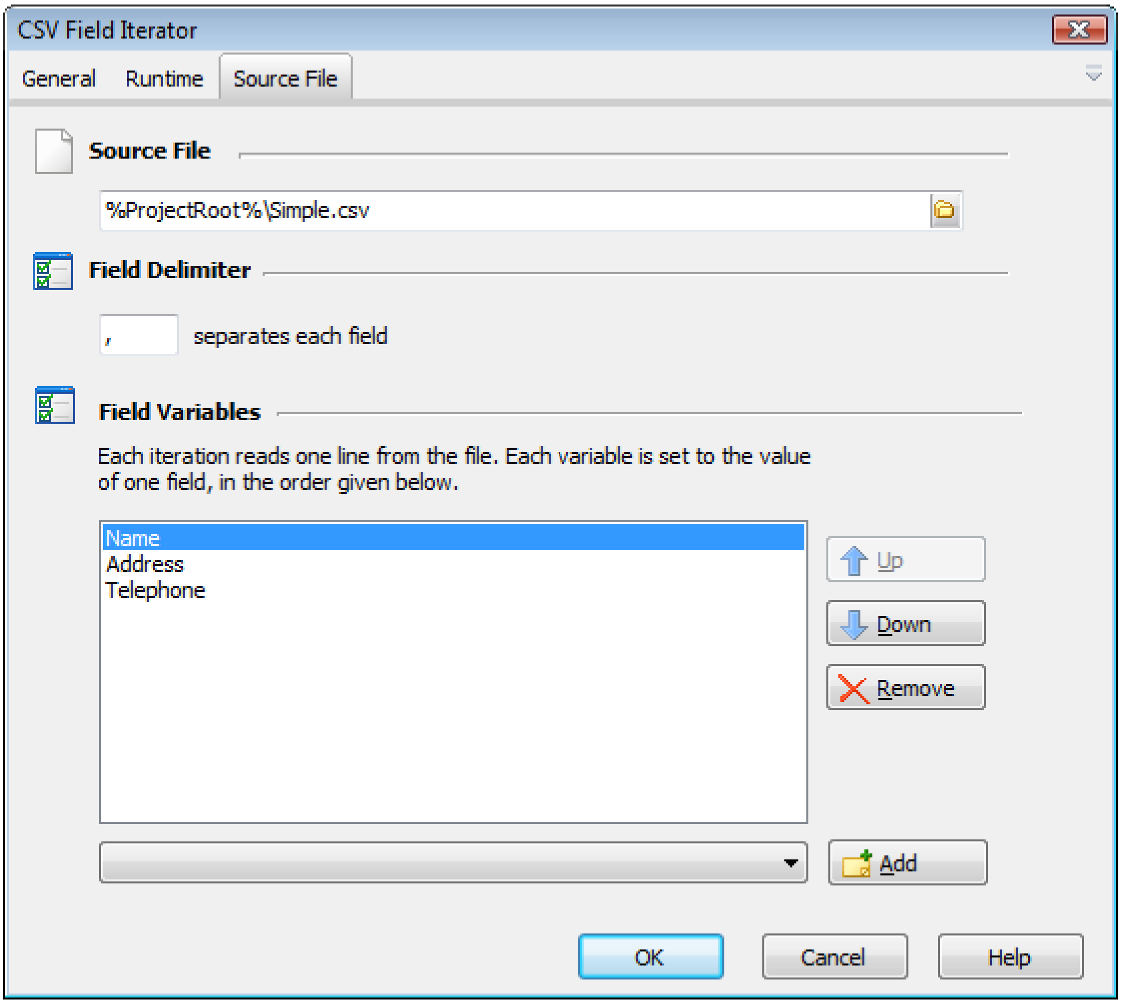
6.24 String Manipulation

6.24.1 CSV Field Iterator

The CSV Field Iterator action allows you to easily iterate through the rows of any file where each line is a list of character-delimited fields (ie CSV comma-delimited files, tab-delimited files, etc.)

As each row in the file is iterated, field variables are set to the value of each field in that row.

For more information about iterator actions, see the Iterators topic. For some other actions that iterate over file contents, you may want to see the List Iterator, File Contents Iterator and XML Iterator actions.



Source File

%ProjectRoot%\Simple.csv

Field Delimiter

, separates each field

Field Variables

Each iteration reads one line from the file. Each variable is set to the value of one field, in the order given below.

Name
Address
Telephone

Example

The example above shows the iterator set up for a file, *Simple.csv*, whose contents look like this:

```
John Smith, 33 Pine Way, 555-2357
Albert Jones, 12 North St, 555-1234
Walter Dennis, 66 Ridge Cr, 555-9842
```

The first time the iterator is run, the variable Name will have the value "John Smith". The second time the iterator runs, the variable Name will have the value "Albert Jones", etc.

The Address and Phone variables will be updated to reflect the Address and Phone fields for each row, as well.

Source File

Specify a file to iterate over.

Field Delimiter

By default, the iterator works with CSV (comma-delimited) files. However, you can use any delimiter character, or even a group of characters, to separate fields.

To use a "tab" character as the delimiter, specify \t.

Field Variables

This is where you specify the list of variables which are set on each iteration. The variables are set in order - ie the first variable is set to the field field, the second variable to the second field, etc, etc.

To add a new variable, select it from the combo box below the list and then click the "Add" button.

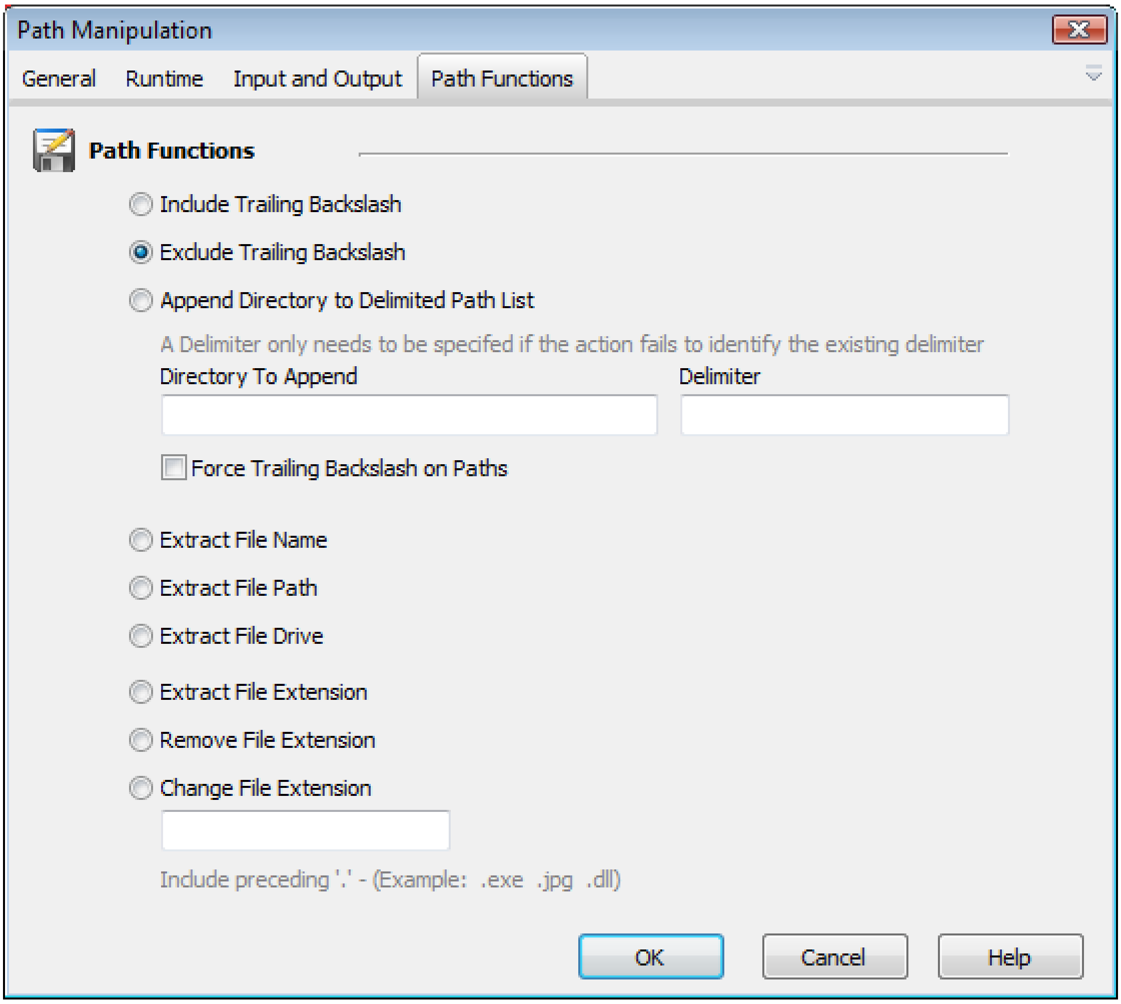
To move a variable up or down in the list, select it and then click the Up or Down button. To remove a variable, select it and click Remove.

At runtime, if there are more field variables to set than there are fields in a row, the extra variables will be cleared.

At runtime, if there are more fields in a row than there are field variables to set, the extra fields will be ignored.

6.24.2 Path Manipulation

The Path Manipulation action allows you to perform common file path functions on a string.



Path Manipulation

Path Functions

○ Include Trailing Backslash

● Exclude Trailing Backslash

○ Append Directory to Delimited Path List

A Delimiter only needs to be specifed if the action fails to identify the existing delimiter

Directory To Append

Delimiter

□ Force Trailing Backslash on Paths

○ Extract File Name

○ Extract File Path

○ Extract File Drive

○ Extract File Extension

○ Remove File Extension

○ Change File Extension

Include preceding '.' - (Example: .exe .jpg .dll)

Include Trailing Backslash
This action will make sure that the output string contains a trailing backslash. If the input string already has a trailing backslash, the string will not be changed.

Exclude Trailing Backslash
This action will make sure that the output string does not have a trailing backslash.

Append Directory to Delimited Path List
This action will append a directory to the end of a delimited directory list. Force trailing backslashes will force each path to end in a backslash.
Example: Appending "C:\Path4" to "C:\Path1;C:\Path2;C:\Path3" will output "C:\Path1;C:\Path2;C:\Path3;C:\Path4".

Extract File Name
This action will extract the filename, including the extension, from the input string.
Example: "C:\Program Files\Program\File.exe" will output "File.exe"

Extract File Path
This action will extract the file path, the filename will be excluded from the string.
Example: "C:\Program Files\Program\File.exe" will output "C:\Program Files\Program\"

Extract File Drive
This action will extract the drive, the file path and name will be excluded from the string.
Example: "C:\Program Files\Program\File.exe" will output "C:"

Extract File Extension
This action will extract the file extension, the drive, path and filename will be excluded from the string.
Example: "C:\Program Files\Program\File.exe" will output ".exe"

Remove File Extension
This action will extract the full path, excluding the file extension.
Example: "C:\Program Files\Program\File.exe" will output "C:\Program Files\Program\File"

Change File Extension
This action will change the file extension of the input string, with the specified extension. The preceding . must be included with the new extension.
Example: "C:\Program Files\Program\File.exe" with new extension as ".zip" will output "C:\Program Files\Program\File.zip"

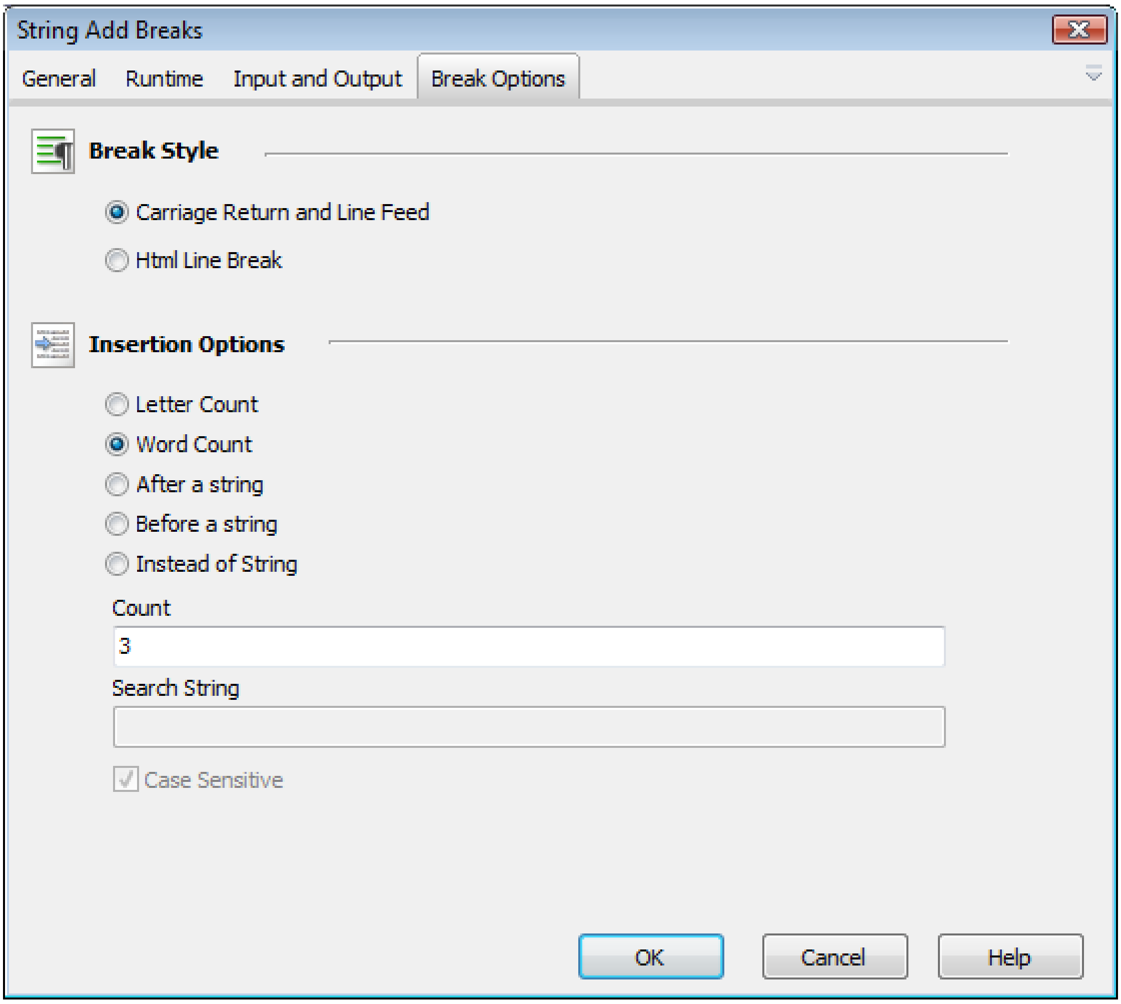
Scripting Info

The Action properties available are :

```
        property  InputString          : string;       // The input string, if not using a variable
        property  OutputVariable       : string;       // The Output Variable name
        property  ApplyToInput         : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable   : boolean;      // True if using an input variable, not input
string
        property  InputVariable        : string;       // Input variable name

        property  IncludeTrailingBackslash    : boolean;      // True to Include Trailing
Backslash, otherwise false
        property  ExcludeTrailingBackslash    : boolean;      // True to Exclude Trailing
Backslash, otherwise false
        property  ExtractFileName             : boolean;      // True to Extract File Name,
otherwise false
        property  ExtractFilePath             : boolean;      // True to Extract File Path,
otherwise false
        property  ExtractFileDrive            : boolean;      // True to Extract File Drive,
otherwise false
        property  ExtractFileExt              : boolean;      // True to Extract File Extension,
otherwise false
        property  RemoveFileExt               : boolean;      // True to Remove File Extension,
otherwise false
        property  ChangeFileExt               : boolean;      // True to Change File Extension,
otherwise false
        property  NewFileExt                  : string;       // The new file extension
```

6.24.3 String Add Breaks

The String Add Breaks action enables you to add breaks at specified locations within a string.



Break Style

● Carriage Return and Line Feed

○ Html Line Break

Insertion Options

○ Letter Count

● Word Count

○ After a string

○ Before a string

○ Instead of String

Count: 3

Search String:

☑ Case Sensitive

Break Style
This action can insert either a Carriage Return and Line Feed, which is the default for Windows operating systems, or a HTML line break.

Letter Count
Inserts a break after a set number of characters.

Word Count
Inserts a break after a set number of words.

Before a String or After a String
Inserts a break before or after the search string.

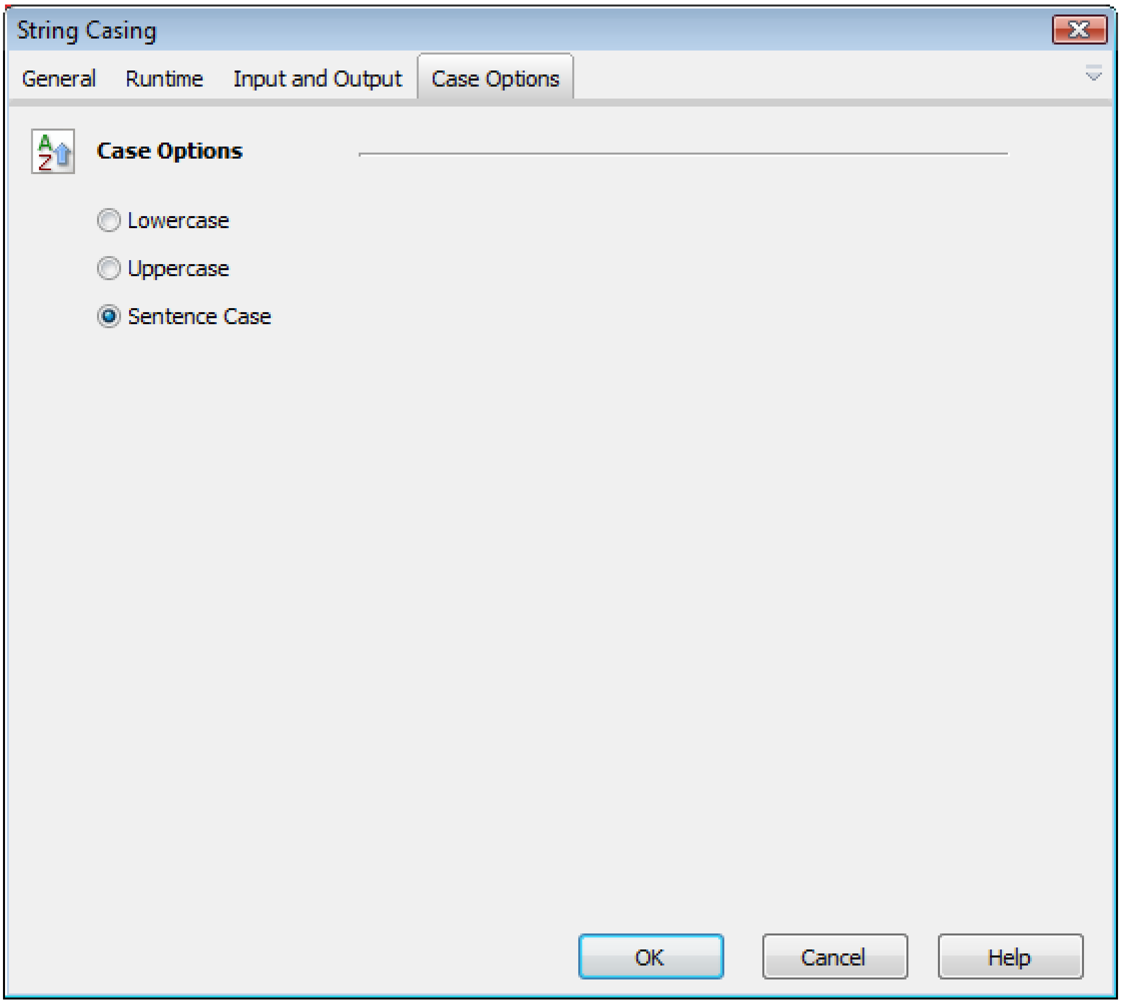
Scripting Info

The Action properties available are :

```
        property  InputString          : string;       // The input string, if not using a variable
        property  OutputVariable       : string;       // The Output Variable name
        property  ApplyToInput         : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable   : boolean;      // True if using an input variable, not input
string
        property  InputVariable        : string;       // Input variable name
        property  UseHtmlBreak            : boolean;      // True to Use Html Break, False to
use CRLF
        property  BreakOnLetterCount      : boolean;      // True to Break On Letter Count,
otherwise false
        property  BreakOnWordCount        : boolean;      // True to Break On Word Count,
otherwise false
        property  BreakOnAfterString      : boolean;      // True to insert Break after string,
otherwise false
        property  BreakOnBeforeString     : boolean;      // True to insert Break before
string, otherwise false

        property  SearchString      : string;       // The search string
        property  Count             : string;       // The Word or Letter count
        property  CaseSensitive     : boolean;      // True for a case sensitive search,
otherwise false.
```

6.24.4 String Casing

The String Casing action allows you to change a string to be Lowercase, Uppercase or Sentence case.



Case Options

○ Lowercase

○ Uppercase

● Sentence Case

Lowercase
All characters within the string will be converted to lowercase.

Uppercase
All characters within the string will be converted to uppercase.

Sentence Case
Each word within the string will have the first character converted to upper case.

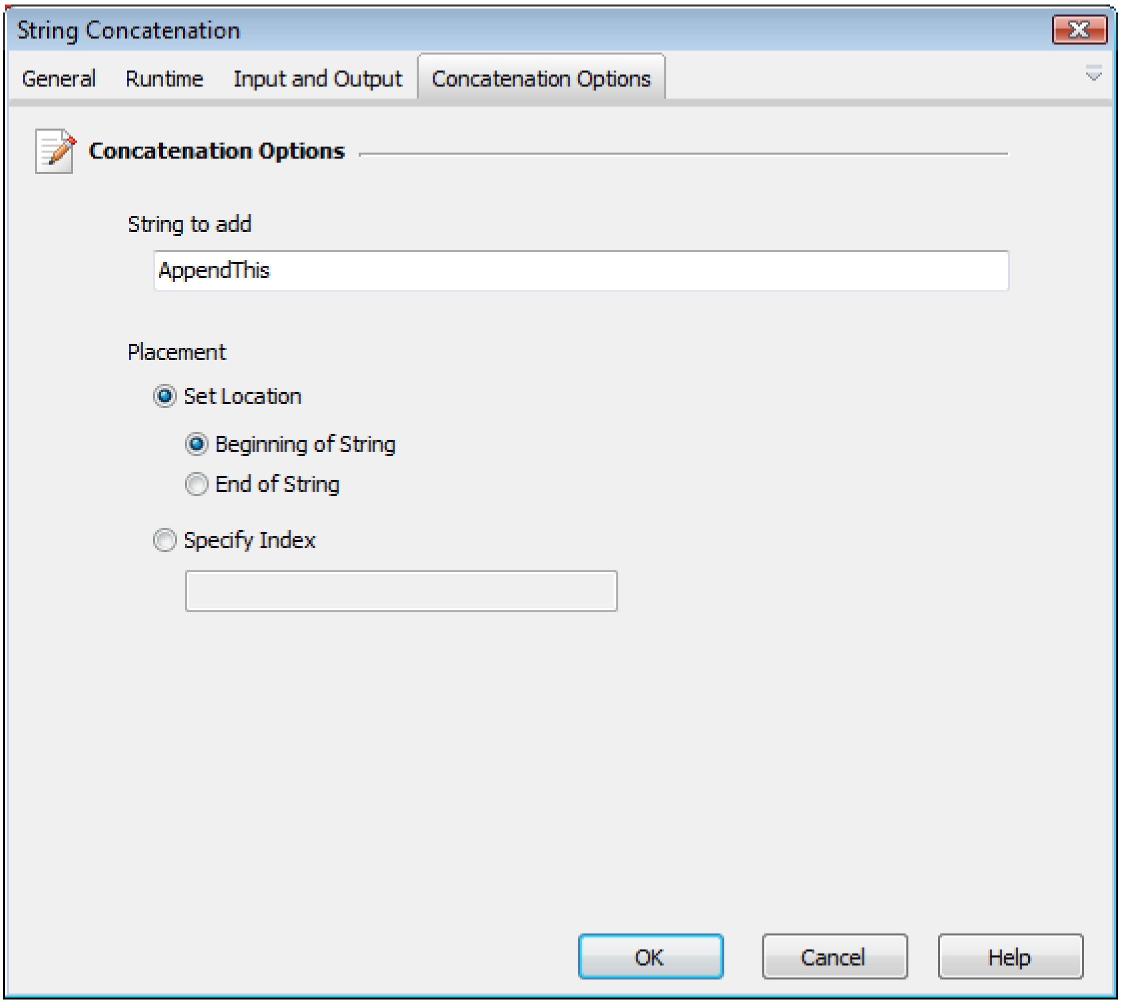
Scripting Info

The Action properties available are :

```
        property  InputString          : string;      // The input string, if not using a variable
        property  OutputVariable       : string;      // The Output Variable name
        property  ApplyToInput         : boolean;     // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable   : boolean;     // True if using an input variable, not input
string
        property  InputVariable        : string;      // Input variable name

        property  CaseUpper            : boolean;     // True to make Uppercase, otherwise
false.
        property  CaseLower            : boolean;     // True to make Lowercase, otherwise
false.
        property  CaseSentence         : boolean;     // True to make Sentence case, otherwise
false.
```

6.24.5 String Concatenation

The String Concatenation action allows you to join two strings together.



Concatenation Options

String to add: AppendThis

Placement

● Set Location

● Beginning of String

○ End of String

○ Specify Index

String to add
The string that will be added to the input string.

Beginning of String
The string will be added to the beginning of the input string.

End of String
The string will be added to the end of the input string.

Specify Index
The string will be added after the specified number of characters. If the specified value is longer then the input string, the string will be added to the end.

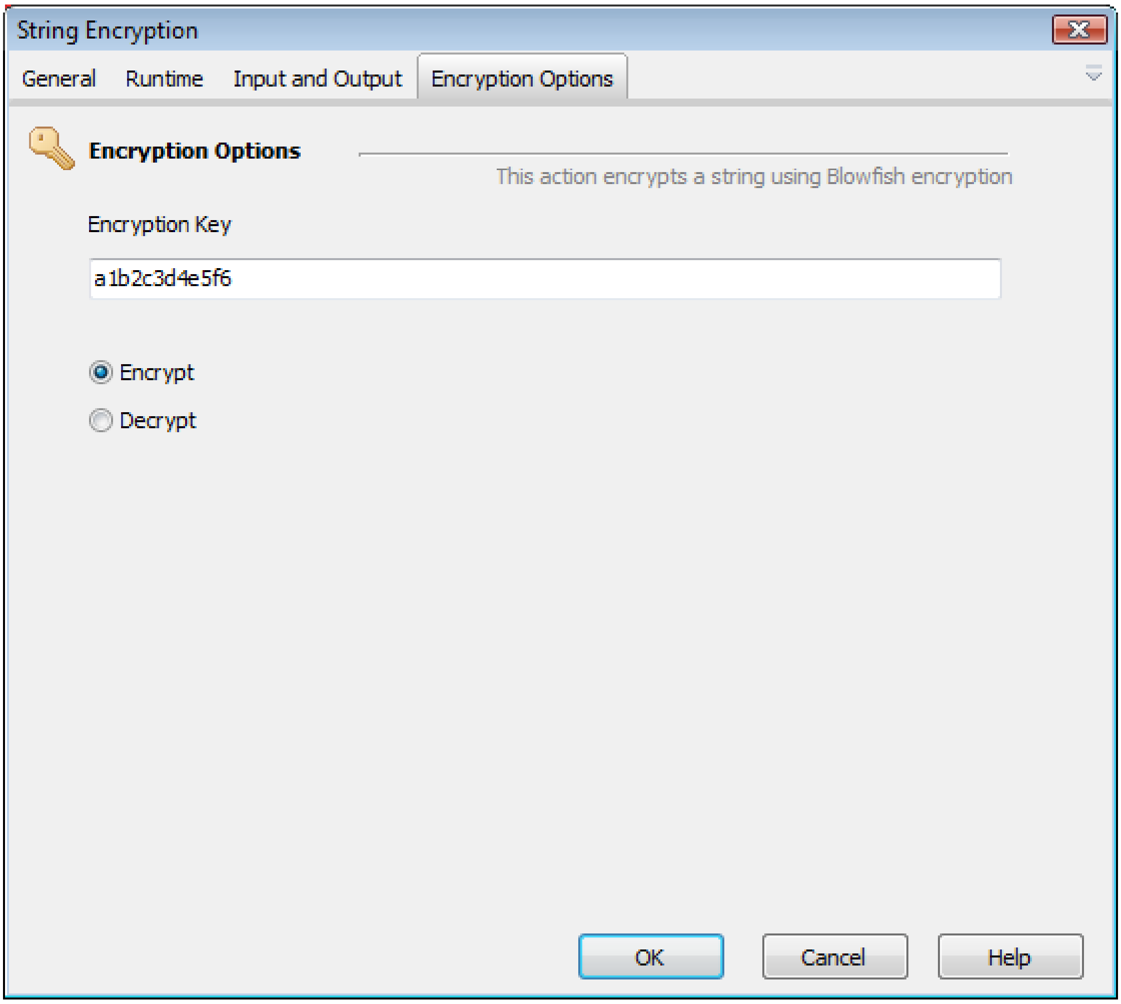
Scripting Info

The Action properties available are :

```
        property InputString         : string;     // The input string, if not using a variable
        property OutputVariable      : string;     // The Output Variable name
        property ApplyToInput        : boolean;    // Apply the changes to the input variable,
only valid if using an input variable
        property UsingInputVariable  : boolean;    // True if using an input variable, not input
string
        property InputVariable       : string;     // Input variable name

        property StringToAdd         : string;     // The string to add
        property UseIndex            : boolean;    // True to specify the index, otherwise
false.
        property InsertIndex         : string;     // The index to insert the string at
        property InsertToLeft        : boolean;    // True to insert at the beginning, false to
insert at the end.
```

6.24.6 String Encryption

The String Encryption action allows you to Encrypt or Decrypt a string using the Blowfish encryption algorithm. For more information regarding Blowfish see - http://en.wikipedia.org/wiki/Blowfish_(cipher)



Encryption Options

This action encrypts a string using Blowfish encryption

Encryption Key

a1b2c3d4e5f6

● Encrypt

○ Decrypt

Encryption Key

An encryption key is used to generate the encrypted string. The same key must be used to Decrypt the string.

Encrypt

The plain text input string is encrypted using the encryption key, the result is then passed to the output string in an encrypted state.

Decrypt

The input string is in an encrypted state and is decrypted using the same key used to encrypt the input string. The decrypted string is then passed to the output string as plain text.

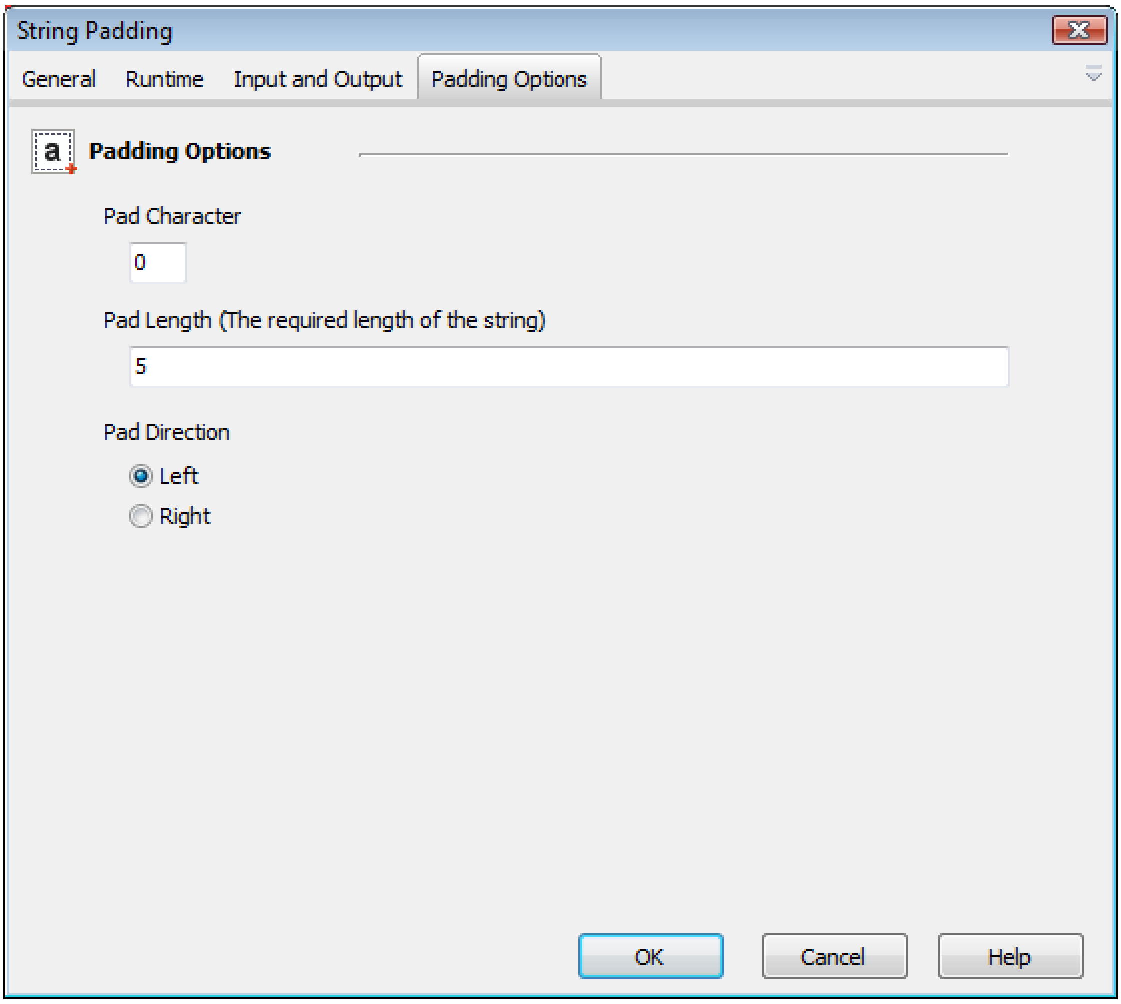
Scripting Info

The Action properties available are :

```
        property  InputString         : string;       // The input string, if not using a variable
        property  OutputVariable      : string;       // The Output Variable name
        property  ApplyToInput        : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable  : boolean;      // True if using an input variable, not input
string
        property  InputVariable       : string;       // Input variable name
property  EncryptionKey      : string;       // The encryption key
property  Encrypt            : boolean;      // True to encrypt, false to decrypt
```

6.24.7 String Padding

The String Padding action allows you to pad a string to a specified length.



Padding Options

Pad Character: 0

Pad Length (The required length of the string): 5

Pad Direction

● Left

○ Right

Pad Character
The character that is used to pad the string.

Pad Length
The required length of the string. If the specified value is shorter then the input string, no change will be made.

Pad Direction
Padding left will insert the pad character to the beginning of the string as many times as needed to make the required length. Padding right will insert the character at the end of the string.

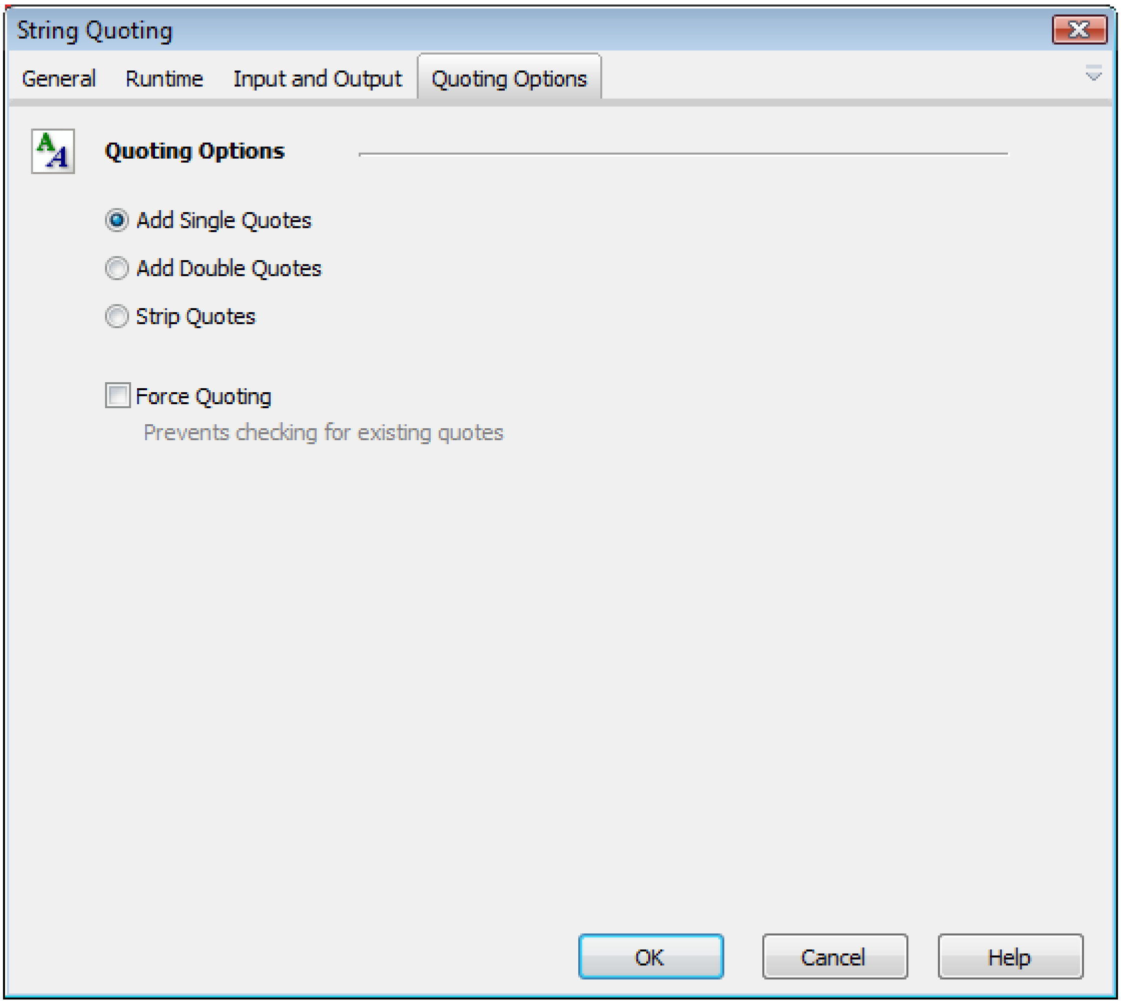
Scripting Info

The Action properties available are :

```
property  InputString        : string;       // The input string, if not using a variable
property  OutputVariable     : string;       // The Output Variable name
property  ApplyToInput       : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable  : boolean;      // True if using an input variable, not input
string
        property  InputVariable       : string;       // Input variable name

        property  PadCharacter        : string;       // The character used to pad
        property  PadLength           : string;       // The length to pad the string to
        property  PadLeft             : boolean;      // True to pad to the left, otherwise false
```

6.24.8 String Quoting

The String Quoting action allows you to add or remove quotes from a string.



Quoting Options

● Add Single Quotes

○ Add Double Quotes

○ Strip Quotes

□ Force Quoting

Prevents checking for existing quotes

Add Single Quotes
Single quotes will be added to the beginning and end of a string.

Add Double Quotes
Double quotes will be added to the beginning and end of a string.

Strip Quotes
This action will remove one pair of matching quotes from a string.

Force Quoting
By default this action checks for quotes before adding them, enabling this option prevents this, which enables you to add extra quotes to a string.

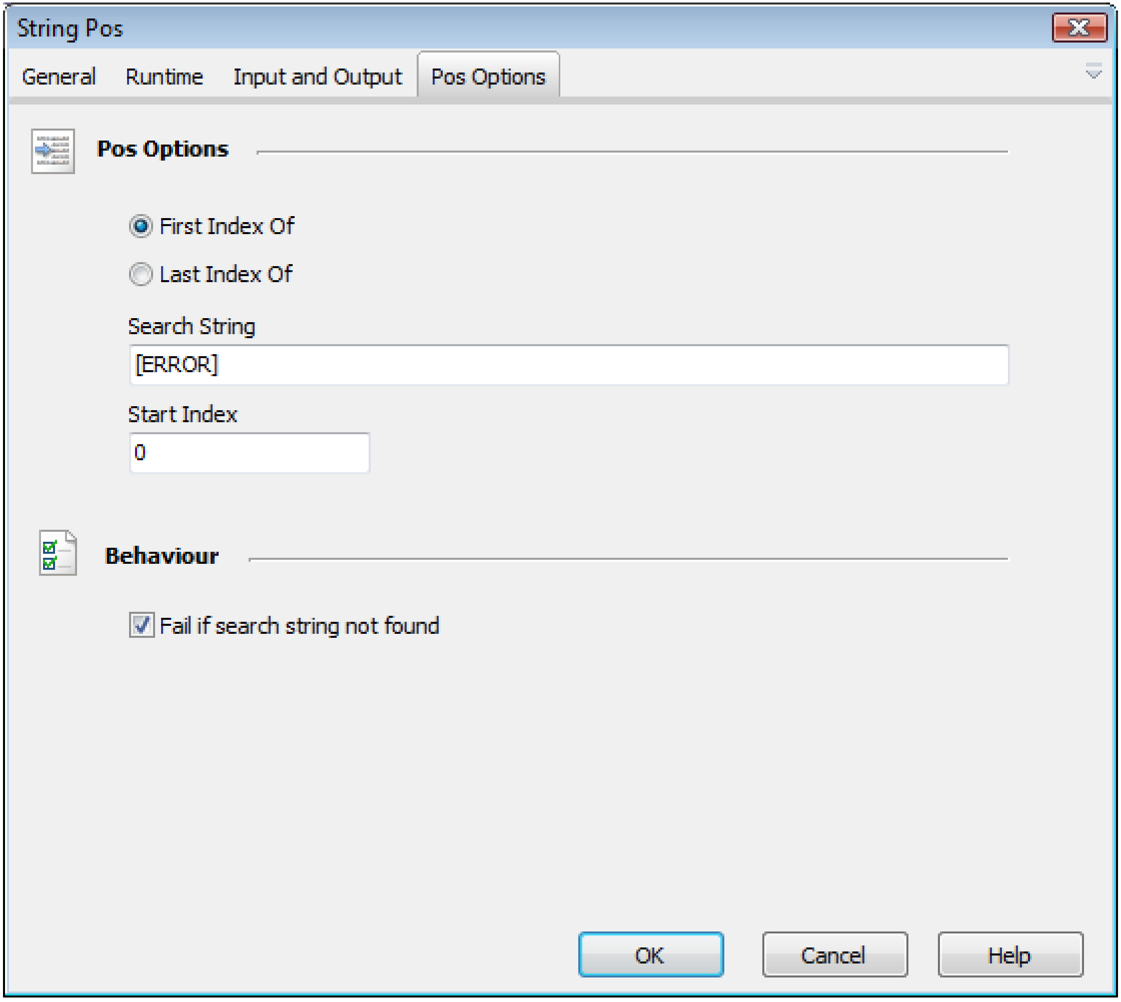
Scripting Info

The Action properties available are :

```
          property  InputString         : string;       // The input string, if not using a variable
          property  OutputVariable      : string;       // The Output Variable name
          property  ApplyToInput        : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
          property  UsingInputVariable  : boolean;      // True if using an input variable, not input
string
          property  InputVariable       : string;       // Input variable name

          property  AddDoubleQuotes     : boolean;      //True to Add Double Quotes, otherwise
false
          property  AddSingleQuotes     : boolean;      //True to Add Single Quotes, otherwise
false
          property  StripQuotes         : boolean;      //True to Strip Quotes, otherwise false
          property  ForceQuoting        : boolean;      //True to Force Quoting
```

6.24.9 String Pos

The String Pos action allows you to find the index of a string or character within a string.



Pos Options

● First Index Of

○ Last Index Of

Search String: [ERROR]

Start Index: 0

Behaviour

☑ Fail if search string not found

First Index Of - Find the first instance of the character or string within the input string.

Last Index Of - Find the last instance of the character or string within the input string.

Search String - The string to search for.

Start Index - The position to start the search from.

Fail if search string not found - Action fails if search string is not found within input.

Scripting Info

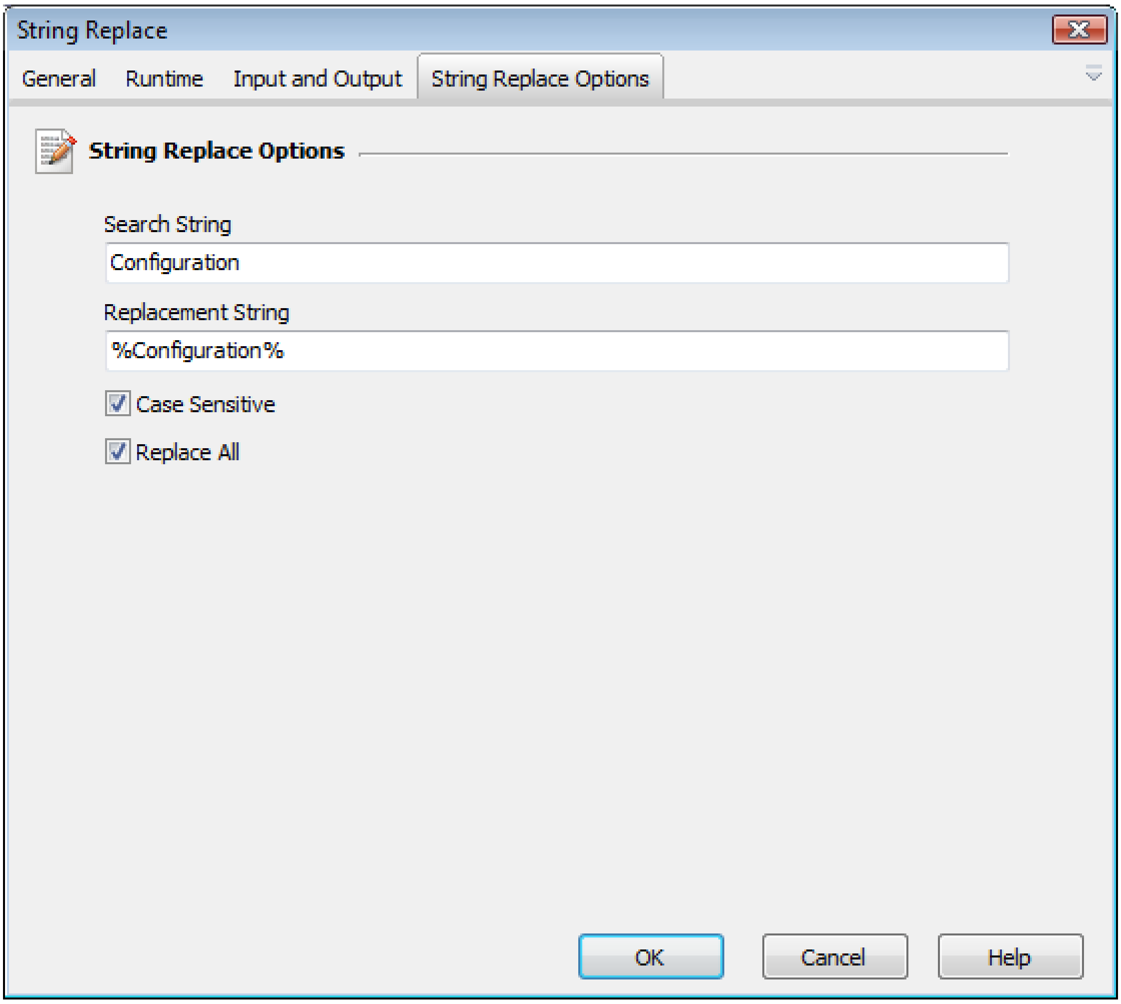
The Action properties available are :

```
        property  InputString          : string;       // The input string, if not using a variable
        property  OutputVariable       : string;       // The Output Variable name
        property  ApplyToInput         : boolean;      // Apply the changes to the input variable, only valid if using an input variable
        property  UsingInputVariable   : boolean;      // True if using an input variable, not input string
        property  InputVariable        : string;       // Input variable name

        property SearchString          : string        // The string to serach for within the input string
        property StartIndex            : string        // The position to start the search from
        property FailIfNotFound        : boolean       // Fail action if search string not found within input
```

6.24.10 String Replace

The String Replace action enables you to replace either all occurrences or the first occurrence of a string, with a new string.

Note: For more advanced text replacement options, see the Text Find / Replace Action.



String Replace Options

Search String: Configuration

Replacement String: %Configuration%

☑ Case Sensitive

☑ Replace All

Search String
The string to find within the input.

Replacement String
The string used to replace found string.

Replace All
This options will force the action to iterate over the entire input string, until no matches are found. By default this action will only replace the first match.

Scripting Info

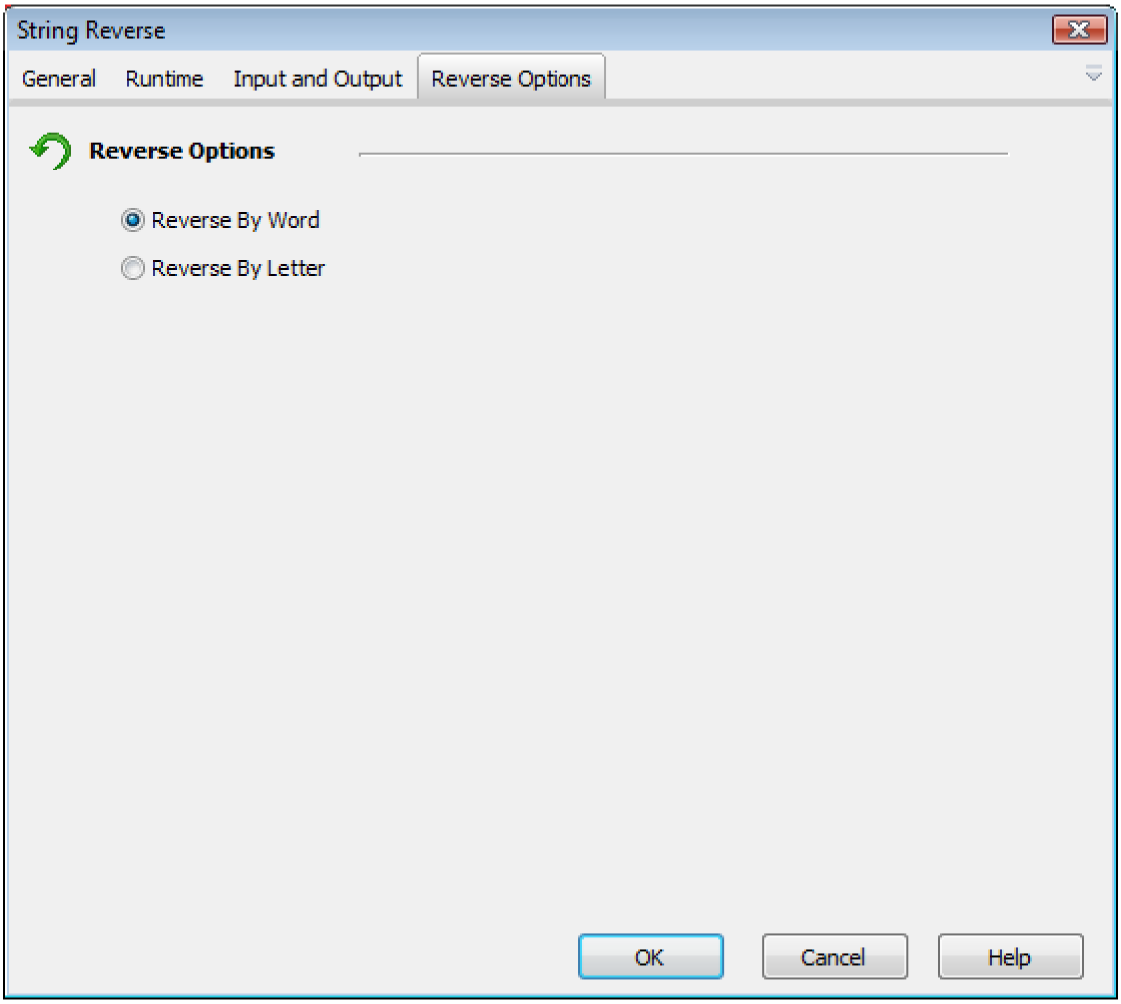
The Action properties available are :

```
        property  InputString          : string;       // The input string, if not using a variable
        property  OutputVariable       : string;       // The Output Variable name
        property  ApplyToInput         : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable   : boolean;      // True if using an input variable, not input
string
        property  InputVariable        : string;       // Input variable name

        property  SearchString         : string;       // String to search for
        property  ReplacementString    : string;       // The replacement string
        property  CaseSensitive        : boolean;      // True to perform case sensitive search,
otherwise false
```

```
        property  ReplaceAll          : boolean;      // True to iterate through entire string,
otherwise false
```

6.24.11 String Reverse

The String Reverse action enables you to reverse a string. The string can be reversed by words or letters.



● Reverse By Word
○ Reverse By Letter

Reverse by Word
The input string will be reversed by words.
Example: "one two three four five" will become "five four three two one"

Reverse by Letter
The input string will be reversed by letters.
Example: "0123456789" will become "9876543210"

Scripting Info

The Action properties available are :

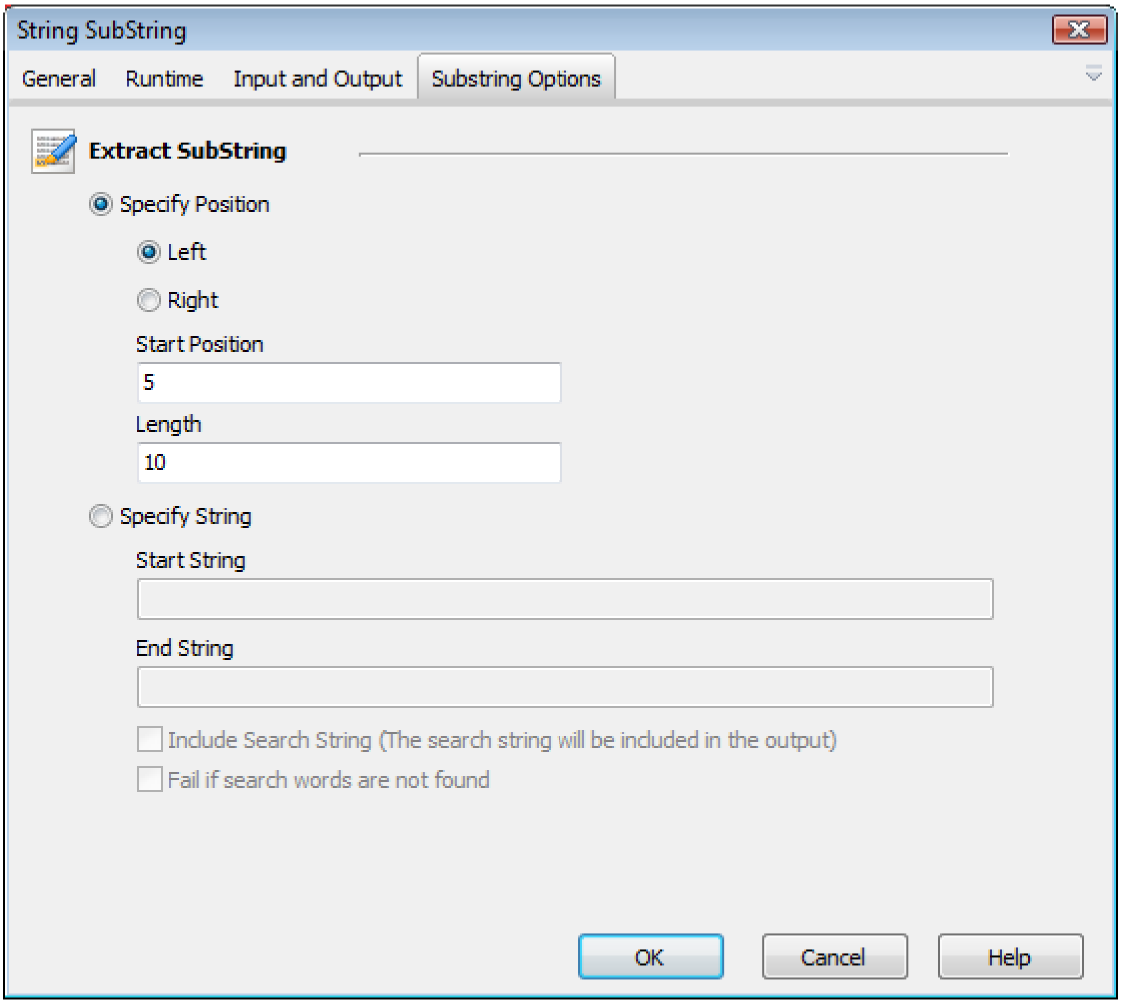
```
        property  InputString         : string;       // The input string, if not using a variable
        property  OutputVariable      : string;       // The Output Variable name
        property  ApplyToInput        : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable  : boolean;      // True if using an input variable, not input
string
```

property InputVariable : string; // Input variable name

property ReverseByWord : boolean; // True to reverse by Words, otherwise false

6.24.12 String SubString

The String SubString action enables you to extract part of a string, where it can be used in subsequent actions.



String SubString

Extract SubString

● Specify Position

● Left

○ Right

Start Position: 5

Length: 10

○ Specify String

Start String:

End String:

□ Include Search String (The search string will be included in the output)

□ Fail if search words are not found

Specify Position
The action will extract a specified number of characters from the beginning or end of the string.

Specify String
This option allows you to extract part of a string between two set characters or words.
Example: Input string "one two three four" extracting between "one" and "four" will return " two three "

Include Search String
This option determines wether the output string includes the search words.

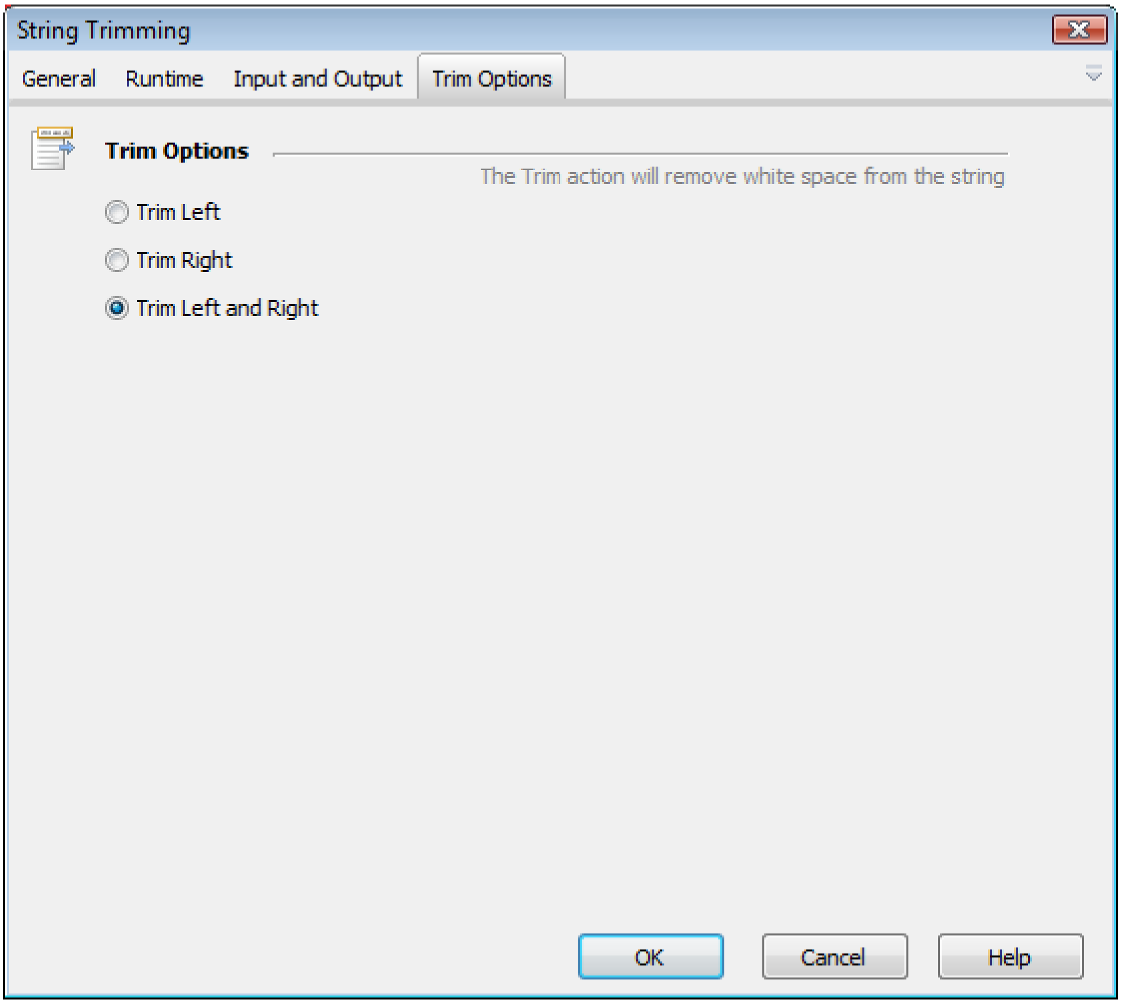
Scripting Info

The Action properties available are :

```
        property  InputString          : string;        // The input string, if not using a variable
        property  OutputVariable       : string;        // The Output Variable name
        property  ApplyToInput         : boolean;       // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable   : boolean;       // True if using an input variable, not input
string
        property  InputVariable        : string;        // Input variable name

        property  Left                         : boolean;       // True to extract from the
beginning, false to extract from the end
        property  Length                       : string;        // Length to extract
        property  CharacterStart               : string;        // Start string to find
        property  CharacterEnd                 : string;        // End string to find
        property  HasSpecifiedIndex            : boolean;       // True to specify index, otherwise
false
        property  HasIncludedSearchWord        : boolean;       // True to include search word in
the output, otherwise false
        property  FailIfStringNotFound         : boolean;       // True to fail if string is not found,
otherwise false
```

6.24.13 String Trimming

The String Trimming Action allows you to remove white space from the beginning or end of a string.



Trim Options

The Trim action will remove white space from the string

○ Trim Left

○ Trim Right

● Trim Left and Right

Trim Left

White space will be removed from the beginning of a string.

Trim Right
White space will be removed from the end of a string.

Trim Left and Right
White space will be removed from both the beginning and end of a string.

Scripting Info

The Action properties available are :

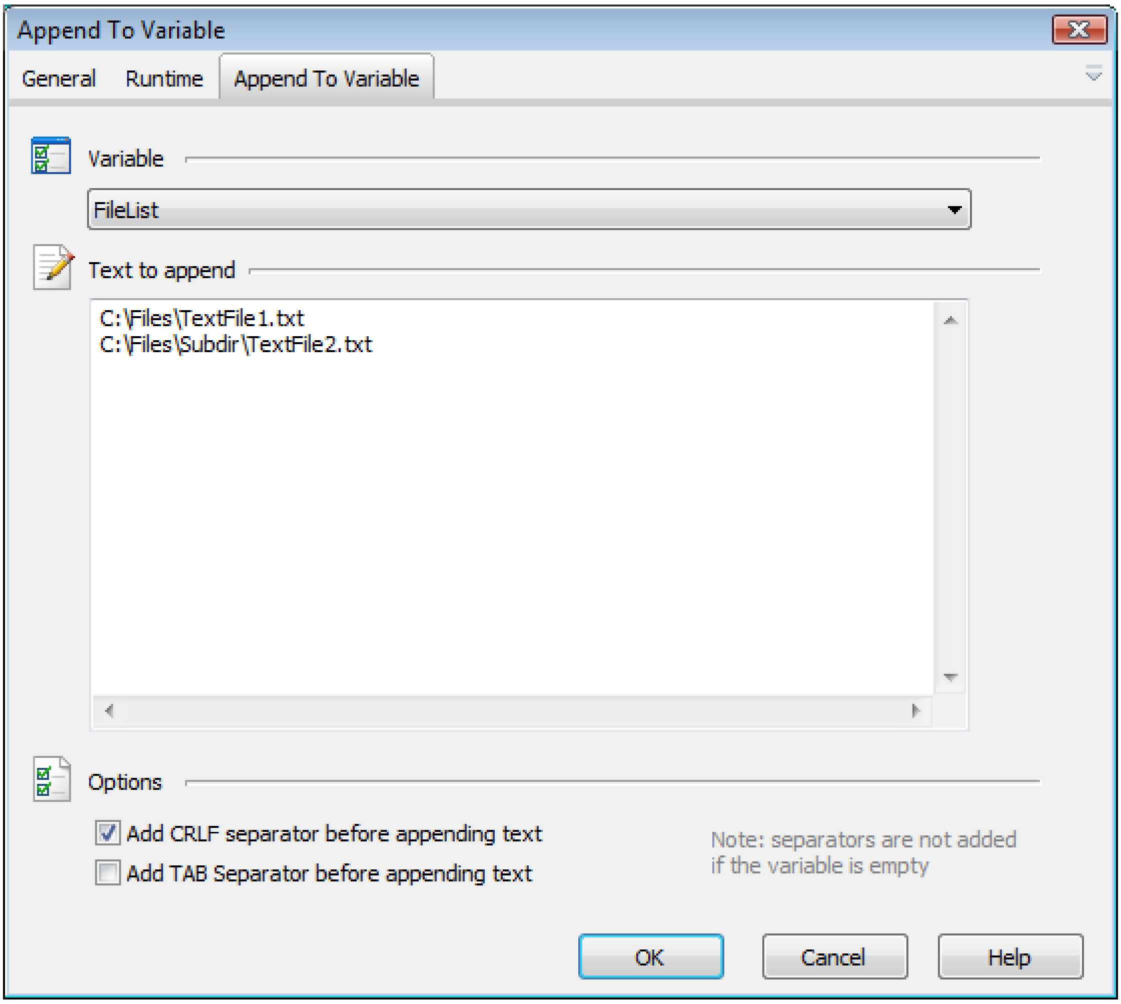
```
        property  InputString          : string;       // The input string, if not using a variable
        property  OutputVariable       : string;       // The Output Variable name
        property  ApplyToInput         : boolean;      // Apply the changes to the input variable,
only valid if using an input variable
        property  UsingInputVariable   : boolean;      // True if using an input variable, not input
string
        property  InputVariable        : string;       // Input variable name

        property  TrimType             : string;       // Valid values: Left, Right, All
```

6.25 Variables

6.25.1 Append to Variable Action

The Append to Variable actions allows you to add more text to the end of a variable, as well as adding a carriage return/linefeed (CRLF) and/or a Tab character to separate the new value.

This action is useful to build up a list of items for use in a List Iterator action, or writing lines to a text file.



Variable: FileList

Text to append:

```
C:\Files\TextFile1.txt
C:\Files\Subdir\TextFile2.txt
```

☑ Add CRLF separator before appending text

□ Add TAB Separator before appending text

Note: separators are not added if the variable is empty

Variable: the variable to update, the new value will be added to the end of the existing value.

Text to Append: this is the text which will be appended to the existing variable. FB Variables in this field will be expanded.

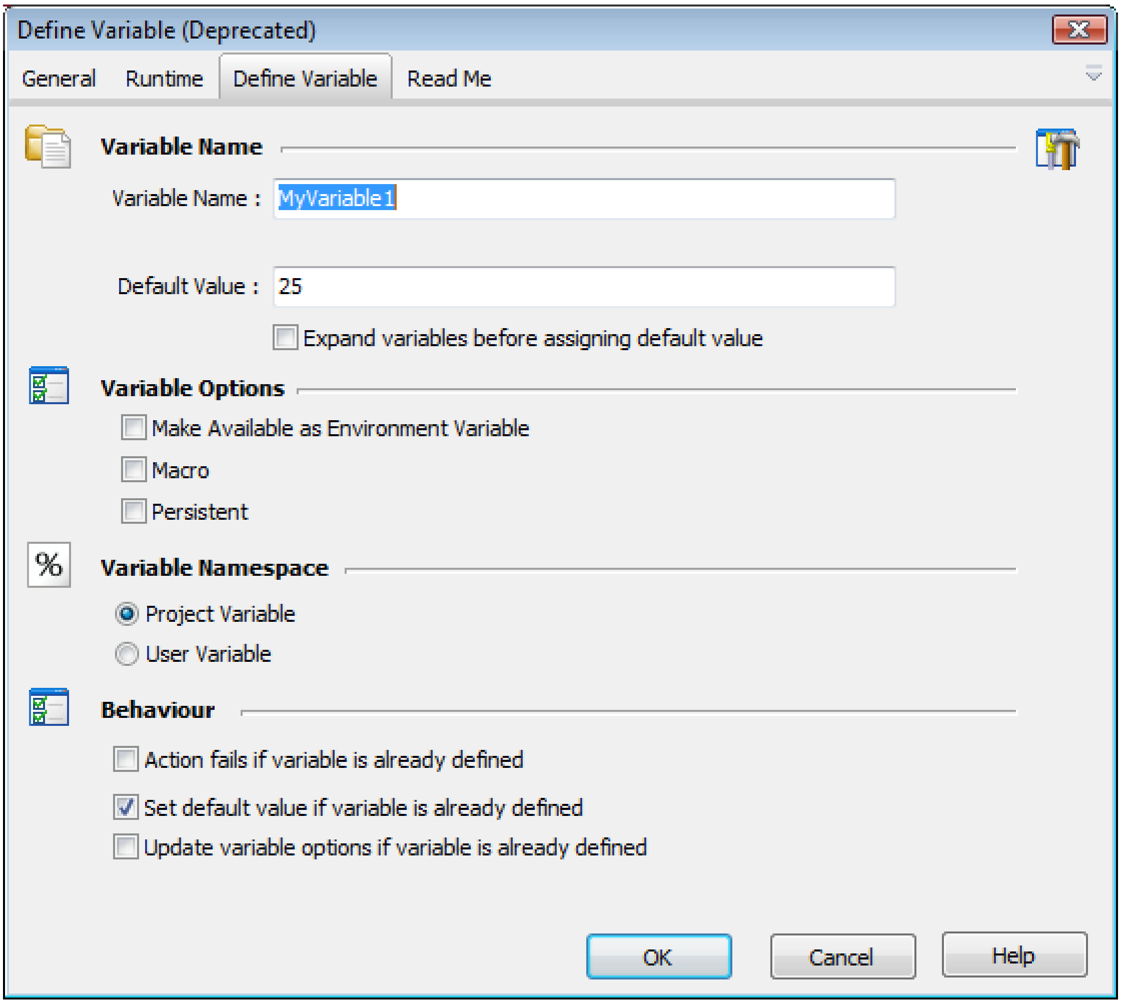
Add CRLF separator: Carriage Return and Line Feed characters will be inserted before the text is appended (if the existing value is not empty.)

Add TAB separator: a tab character will be inserted before the text is appended (if the existing value is not empty.)

6.25.2 Define Variable Action

The define variable action allows you to define a new Automise variable during the execution of the project. Project or User variables are supported.

IMPORTANT: This is not the recommended way to create Automise variables for normal use. See the Variables overview topic for details.


Define Variable (Deprecated)

Variable Name

Variable Name : MyVariable1

Default Value : 25

□ Expand variables before assigning default value

Variable Options

□ Make Available as Environment Variable

□ Macro

□ Persistent

Variable Namespace

● Project Variable

○ User Variable

Behaviour

□ Action fails if variable is already defined

☑ Set default value if variable is already defined

□ Update variable options if variable is already defined

Variable Name

Choose a name for your new variable. Automise will warn you (but not stop you) if you type the name of a preexisting variable.

Default Value

Choose a default value for your new variable. If the 'Set default value if variable already defined' box is checked, this will be the new default value for a preexisting variable, as well.

Expand variables before assigning default value

If you check this box, then variables you enter in the Default Value box (ie %MyVariable %) will be expanded to their current values before being assigned as the default value for the new variable.

Hint: If you want a variable which automatically evaluates other variables each time it is referenced (like a function), you can create a Macro variable (see below.) If you create a macro variable, do not check the *Expand variables before assigning* box.

Make Available as Environment Variable

If checked, the new variable will be available as an environment variable.

Macro

The Is Macro flag forces Automise to re-evaluate the variable whenever it's referenced during the project, otherwise the value of the variable is evaluated when the project starts and that value is used throughout the project. An Is Macro variable is like a function - its value cannot be set during the project using the Set Variable action or any other means.

Persistent

If checked, the new variable will be persistent (the value will be saved between projects.)

Variable Namespace

Choose to create a 'Project' or a 'User' variable.

Behaviour

The action can be set to fail if the variable is already defined (note that variables defined during the project *are* persistent, so if you save a project file after defining a new variable, the new variable will remained defined in the Variables window.)

If the action is set to ignore preexisting variables, it can be set to update the default value. Note that this is the only part of the preexisting variable which can be updated by the action - it is impossible to change any other properties of the variable.

If the "update default value" box is not checked, the action will do nothing if the variable already exists.

If the variable is persistent, then the "set default value if variable already defined" has no effect.

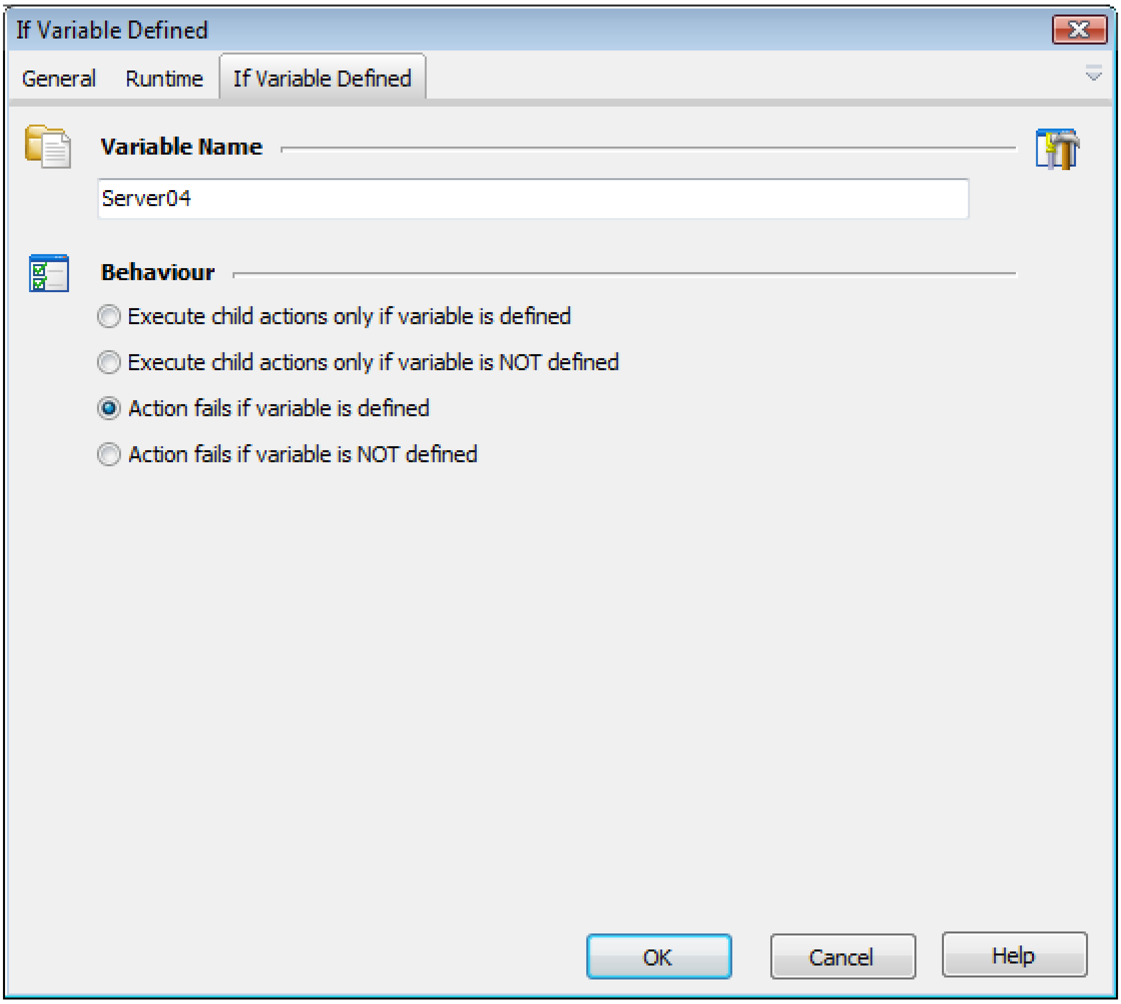
6.25.3 If Variable Defined Action

The If Variable Defined action enables you to control the flow of your project depending on whether a Automise variable is defined. If the variable is defined, then the child actions of the action will be executed.

A variable is defined if it appears in the Edit Variables dialog at all. To test if a defined variable has a value or not, use the If .. Then action.

An Else Action can be used to provide a list of actions to execute if the variable is not defined. Alternatively, you can set the action to fail if the variable is not defined.

Note: The action also tests for Windows Environment variables.



Variable Name

Server04

Behaviour

○ Execute child actions only if variable is defined

○ Execute child actions only if variable is NOT defined

● Action fails if variable is defined

○ Action fails if variable is NOT defined

Variable Name

Specify the name of a Automise variable (or, as in the above picture, a Windows Environment variable) to test for.

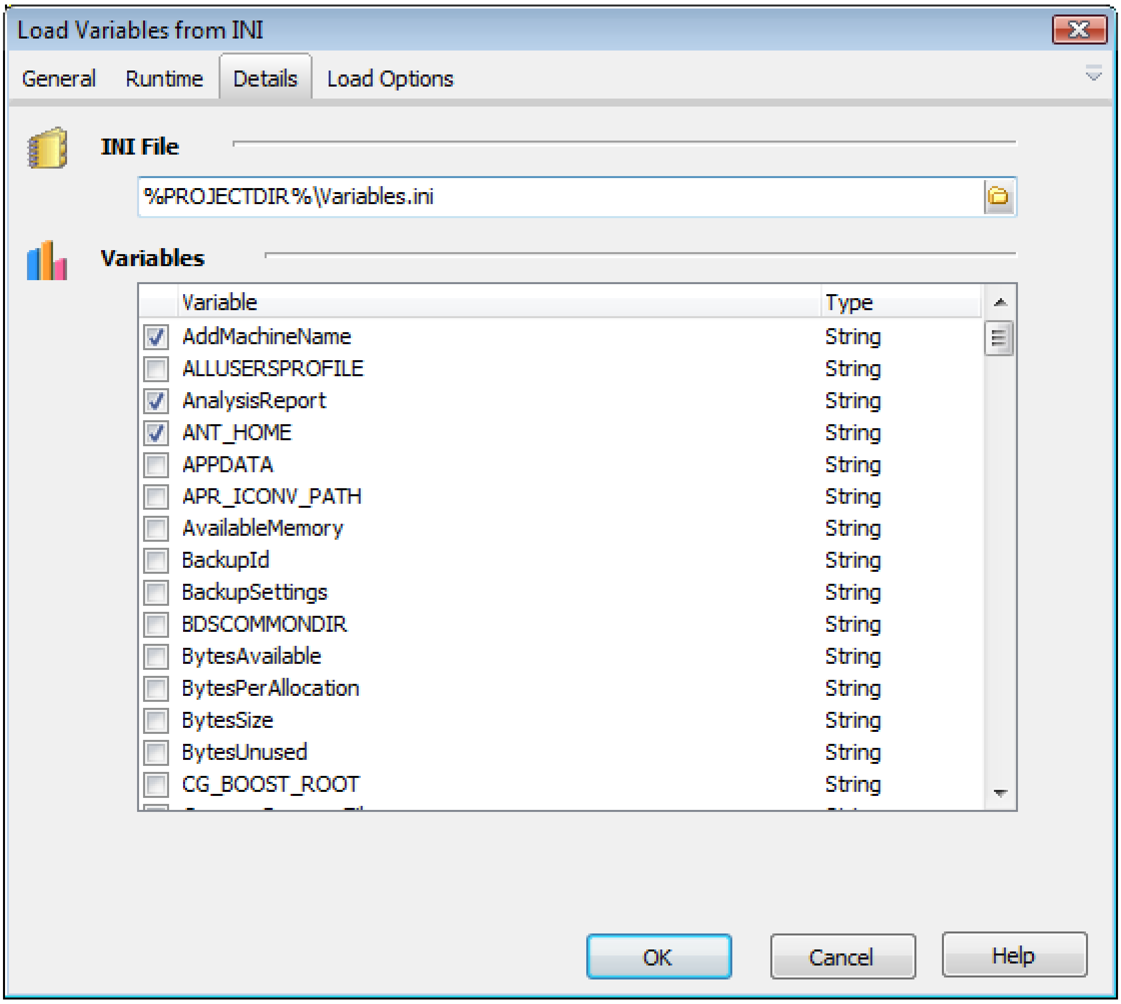
Behaviour

Choose 'Execute child actions' if you want the If Variable Defined Action to behave like an If .. Then Action.

Choose 'Action fails if variable is not defined' if you want the action to fail if the variable is not defined. (Note that if the variable is defined, child actions are still executed.)

6.25.4 Load Variables From Ini

The Load Variables from INI file action enables you to set Automise variables to the values specified in an INI file.



INI File

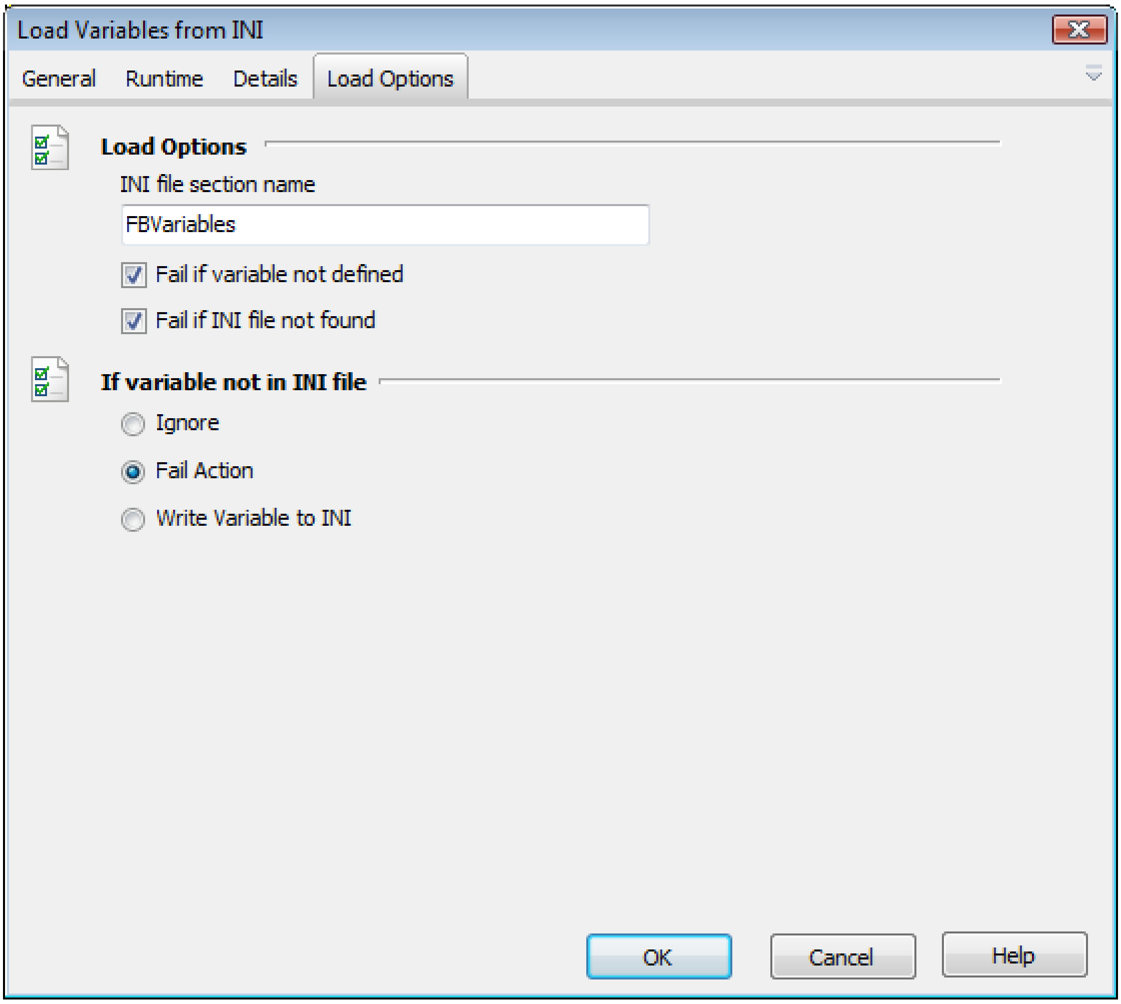
%PROJECTDIR%\Variables.ini

Variables

	Variable	Type
☑	AddMachineName	String
□	ALLUSERSPROFILE	String
☑	AnalysisReport	String
☑	ANT_HOME	String
□	APPDATA	String
□	APR_ICONV_PATH	String
□	AvailableMemory	String
□	BackupId	String
□	BackupSettings	String
□	BDSCOMMONDIR	String
□	BytesAvailable	String
□	BytesPerAllocation	String
□	BytesSize	String
□	BytesUnused	String
□	CG_BOOST_ROOT	String

INI File - Specify the INI file which contains the variables

FB Variables - Specify which variables should be attempted to be set by values in the INI file. You can force the type of the variable to either String, Boolean, or Integer.



Load Options

INI file section name: FBVariables

☑ Fail if variable not defined

☑ Fail if INI file not found

If variable not in INI file

○ Ignore

● Fail Action

○ Write Variable to INI

INI file section name

Specify the section in the INI where the variables are located. This means that you can store different sets of variable values within the one INI file.

Variables must be specified in the following way:
<variableName>=<Variable value>

eg.
[FBVariables]
BuildNumber=10

If you want to use boolean values, then False is 0, True is any other integer value.

See also: INI File format.

Fail if variable not defined

If this option is set and you attempt to load an undefined variable, the action will fail.

Fail if INI file not found

Set this option to false if you want the action to succeed even if the INI File does not exist at all. You will need to set "If Variable not in INI File" to 'Ignore' or 'Write Variable to INI'. If you use 'Write Variable to INI', a new INI file will be created.

If Variable not in INI File

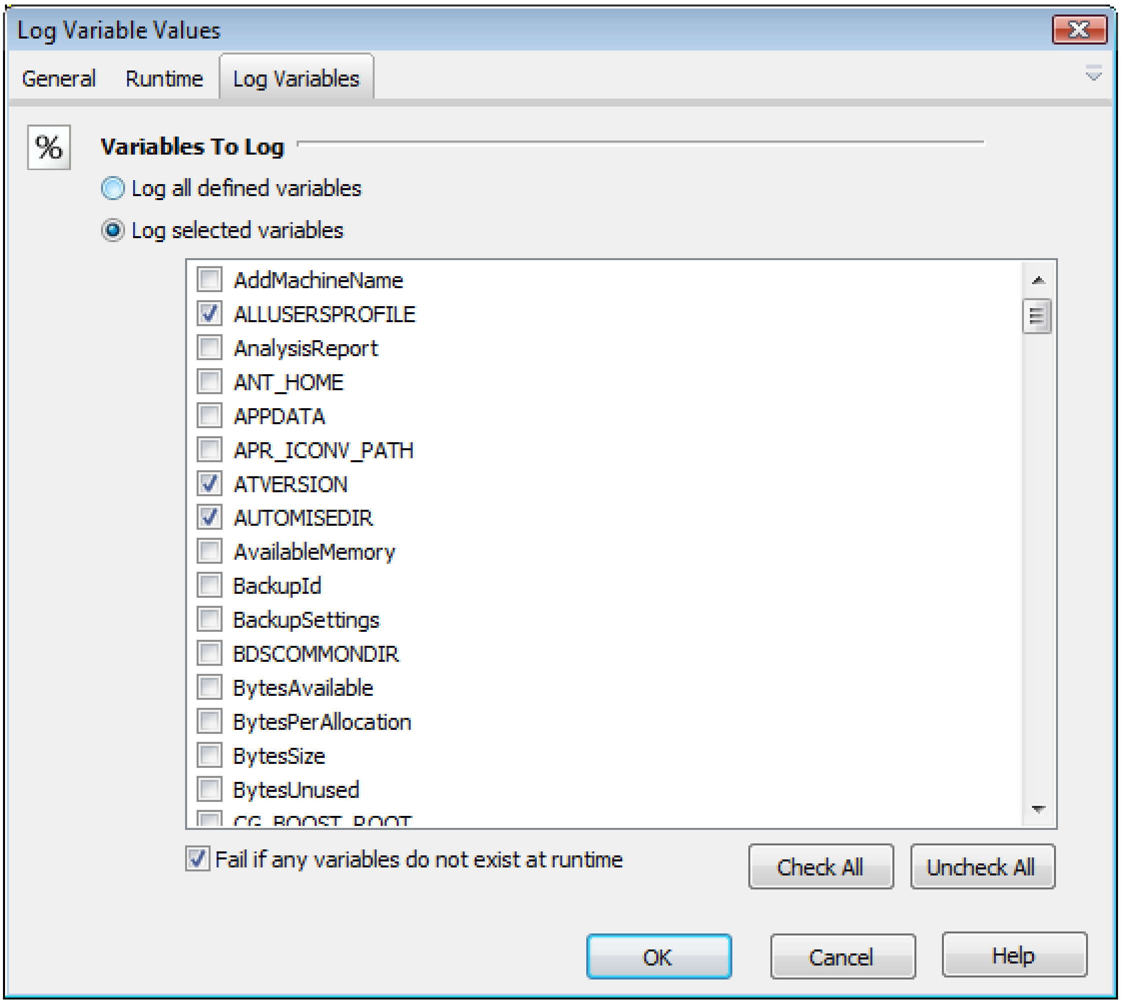
Ignore - Any variables not found in the INI file will be ignored (no failure, and the existing variable value will be kept.)

Fail Action - The action will fail immediately if any variable is not found in the INI file.

Write Variable to INI - If any variable is not found in the INI file, the current value of that variable will be written to the INI file.

6.25.5 Log Variable Values

The Log Variable Values action allows you to log the current values of one or more variables. Logging variable information can be useful when debugging Automise projects (see also Watches) or can be used to make log files easier to follow.



Variables To Log

○ Log all defined variables

● Log selected variables

□ AddMachineName
☑ ALLUSERSPROFILE
□ AnalysisReport
□ ANT_HOME
□ APPDATA
□ APR_ICONV_PATH
☑ ATVERSION
☑ AUTOMISEDIR
□ AvailableMemory
□ BackupId
□ BackupSettings
□ BDSCOMMONDIR
□ BytesAvailable
□ BytesPerAllocation
□ BytesSize
□ BytesUnused
□ CG_BOOST_ROOT

☑ Fail if any variables do not exist at runtime

Log All Defined Variables

If this option is selected, then the list of variables is ignored and the action will log the value of every variable defined at runtime.

Log Selected Variables

Check the names of the variables for the action to log. Use Check All & Uncheck All buttons to quickly select or deselect all variable names.

Fail if any variables do not exist at runtime

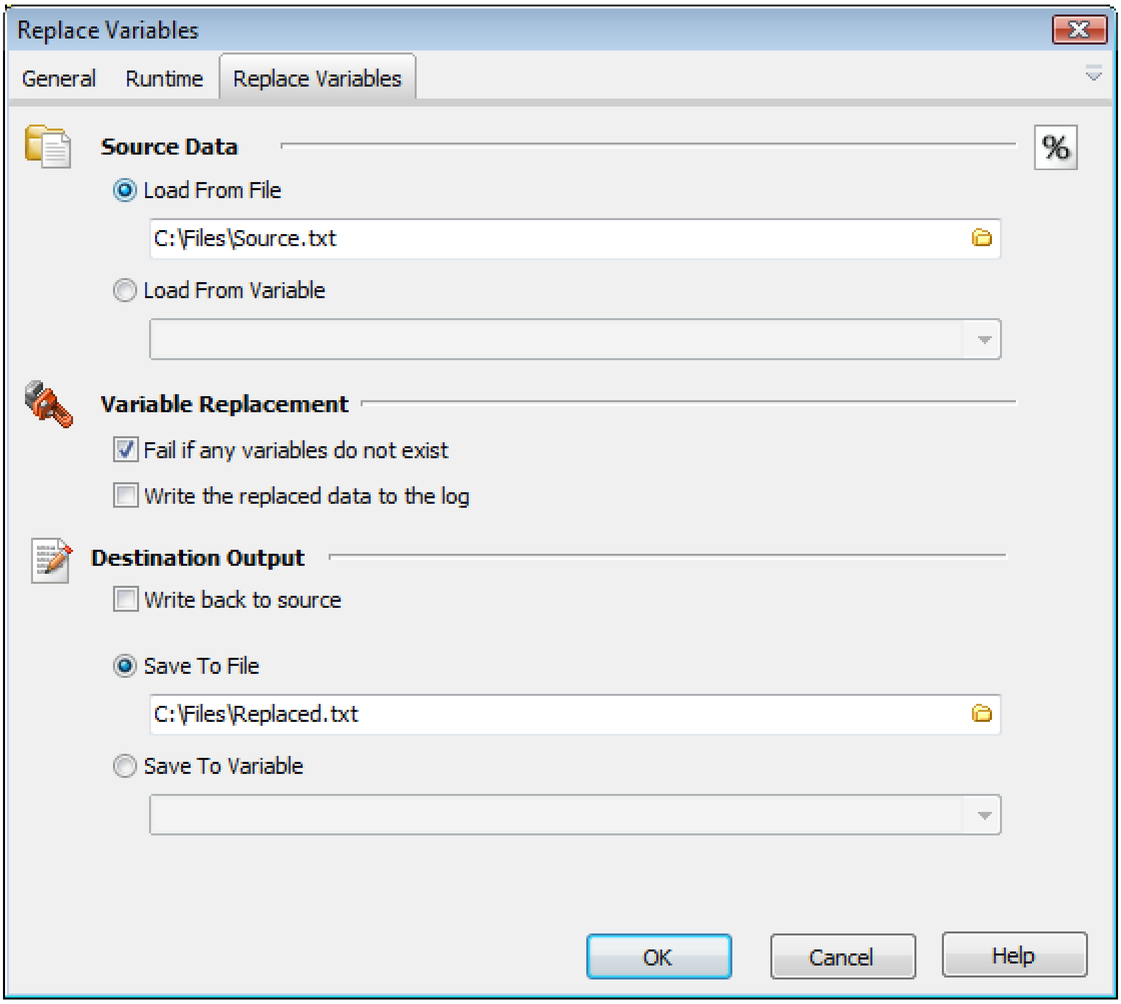
If this checkbox is not selected, variable names which do not exist will be logged as "Variable X is not defined".

See also: Action logging properties

6.25.6 Replace Variables

The Replace Variables action replaces variable reference (ie %VariableName%) in a block of text with their current variable values.

Text can be loaded from a file or from another variable, and can be saved to either a file or a variable.



Replace Variables

General | Runtime | Replace Variables

Source Data

● Load From File

C:\Files\Source.txt

○ Load From Variable

Variable Replacement

☑ Fail if any variables do not exist

□ Write the replaced data to the log

Destination Output

□ Write back to source

● Save To File

C:\Files\Replaced.txt

○ Save To Variable

Source Data

Source text can be loaded from a variable or from a file on disk.

Variable Replacement

"Fail if any variables do not exist"

If this option is set and a variable name is referenced but not defined, the action will fail. Otherwise, the variable reference will be deleted (ie replaced with an empty string.)

"Write the replaced data to the log"

Enable this option in order to echo the replaced text to the action's log output.

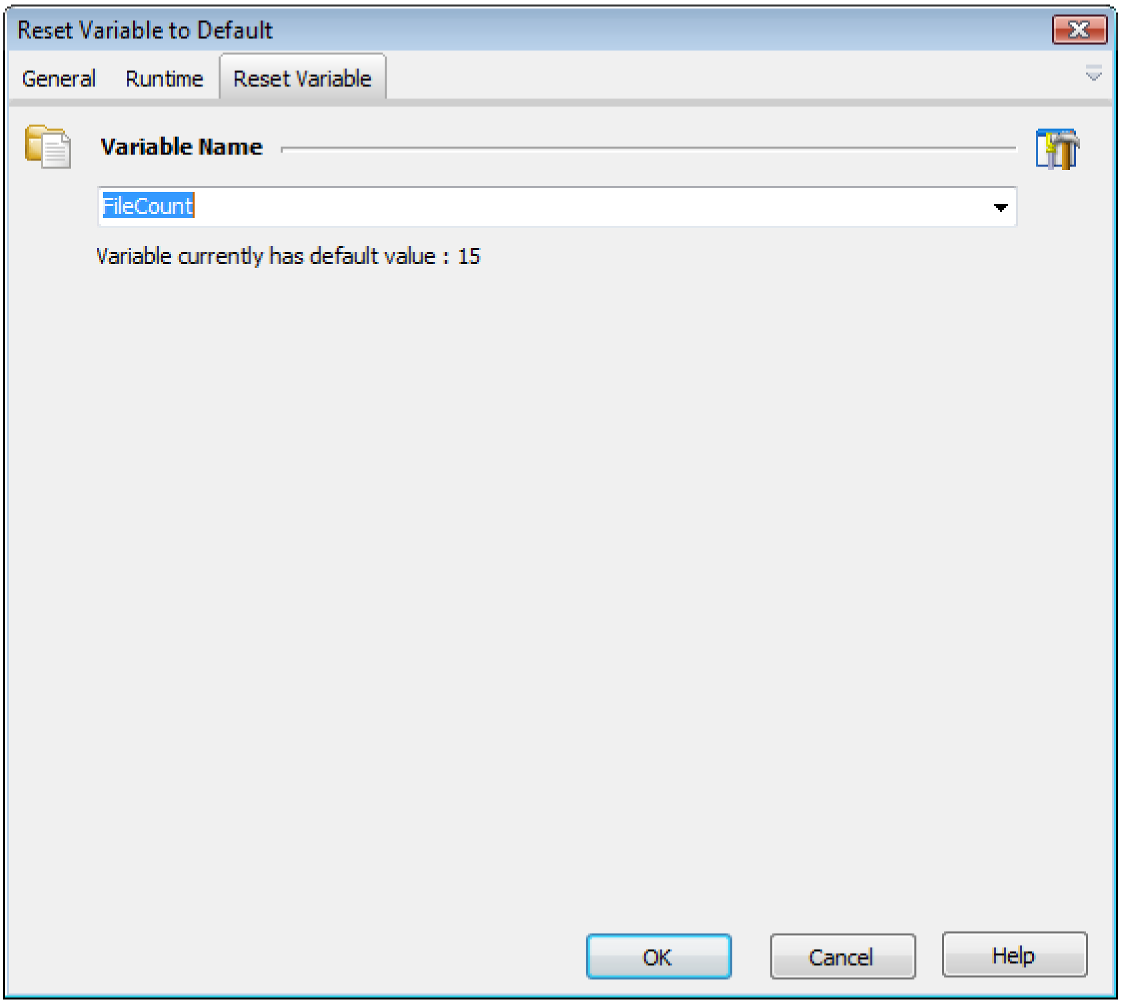
Destination Output

If *"Write back to source"* is selected, the source text (file or variable) will be overwritten with the new text.

If this option is not selected then you can choose a file or variable to write with the new text.

6.25.7 Reset Variable to Default

The Reset Variable to Default action resets a variable to its default value.



FileCount

Variable currently has default value : 15

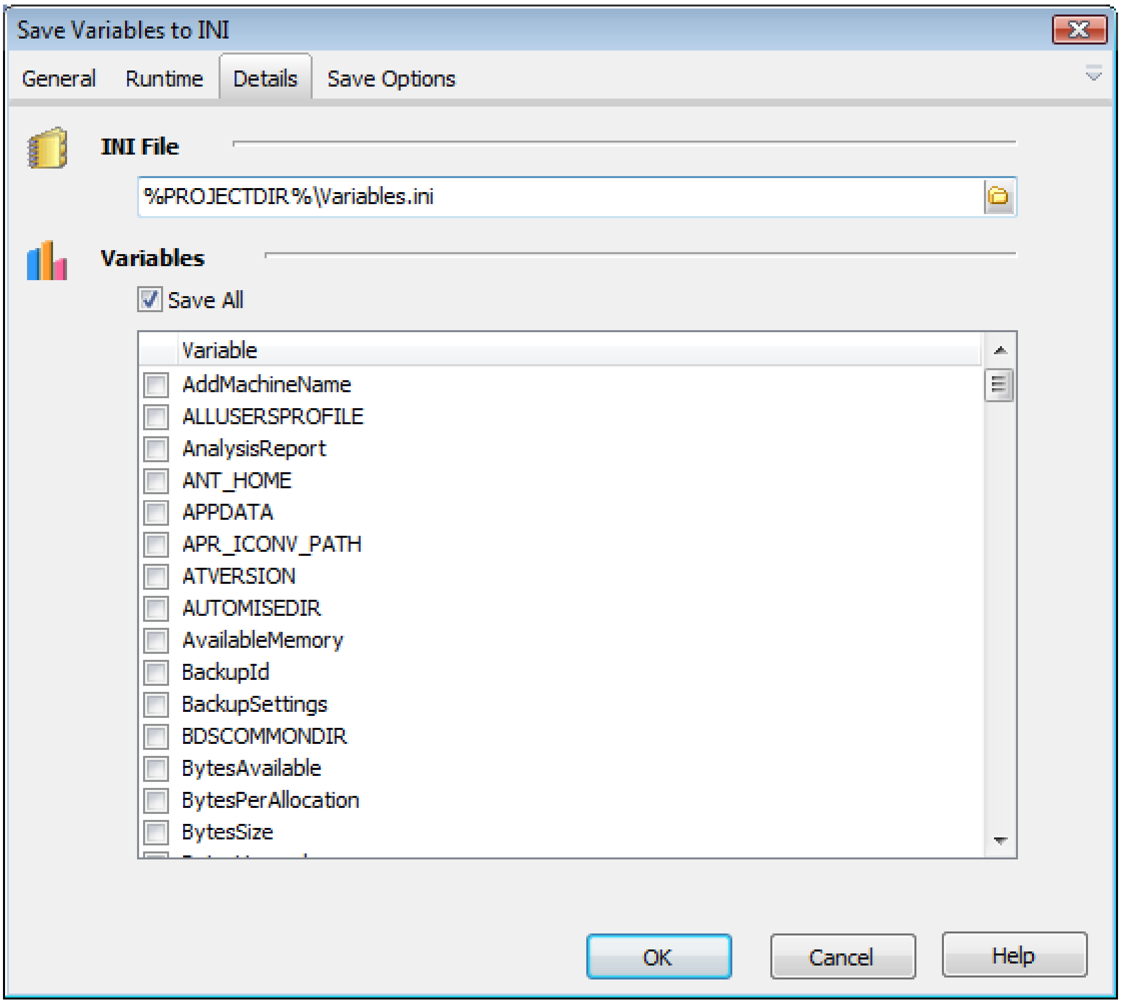
Variable Name

Choose a variable to reset to its default value. If the variable currently has a default value, it will be displayed in the window.

Note that you can reset variables which do not have default values - the variable will be set to blank.

6.25.8 Save Variables To Ini

The Save Variables to INI file action enables you to save a set of Automise variables to an INI file.



INI File

%PROJECTDIR%\Variables.ini

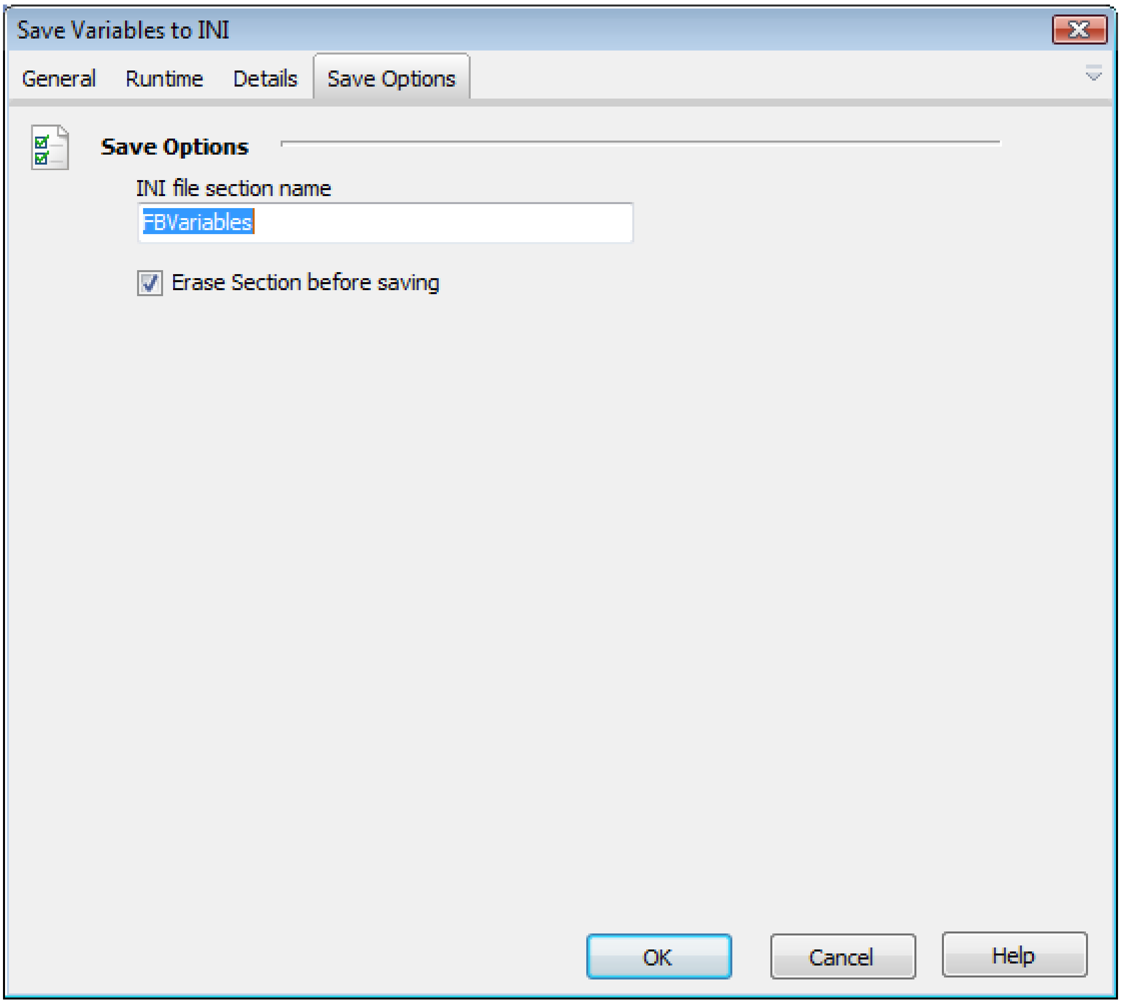
Variables

☑ Save All

Variable
□ AddMachineName
□ ALLUSERSPROFILE
□ AnalysisReport
□ ANT_HOME
□ APPDATA
□ APR_ICONV_PATH
□ ATVERSION
□ AUTOMISEDIR
□ AvailableMemory
□ BackupId
□ BackupSettings
□ BDSCOMMONDIR
□ BytesAvailable
□ BytesPerAllocation
□ BytesSize

INI File - Specify which INI file the variables will be written to

FB Variables - Decide which variables will be written to the file. The variables will be written in the form: <variable name>=<variable value>



Save Options

INI file section name: FBVariables

☑ Erase Section before saving

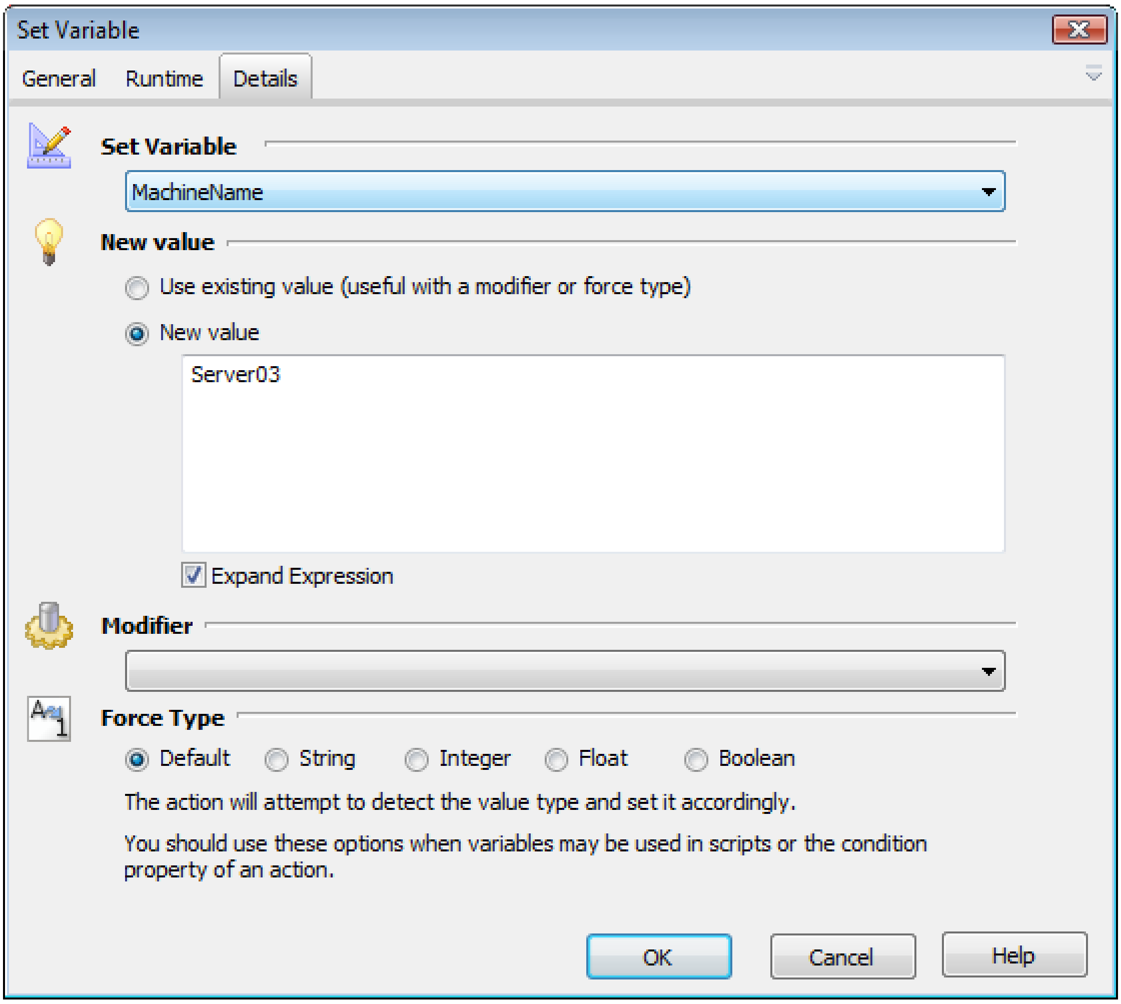
Section - Specify the section in the INI file to write the variables to

Erase Section before saving - This will erase only the specified Section of the INI file. All values will be erased before the new values are written.

6.25.9 Set Variable Action

The Set Variable Action provides a means to set the value of a Project Variable or a User Variable to a new value. The value can include other variables. To append to the existing variable, simply prefix the new value with %VARIABLENAME% where variablename is the name of the variable who's value you are setting.

Note: Although Action List Parameters can be changed, they must be changed from script - not via the Set Variable action.



Set Variable: MachineName

New value

○ Use existing value (useful with a modifier or force type)

● New value

Server03

☑ Expand Expression

Modifier:

Force Type

● Default ○ String ○ Integer ○ Float ○ Boolean

The action will attempt to detect the value type and set it accordingly.

You should use these options when variables may be used in scripts or the condition property of an action.

Set Variable

The name of a Automise variable to set.

Use

New Value

The new value for the variable. Check the Expand Expression box to expand variable and other references in the value. If you want to apply a modifier to the existing value of the variable (eg. maybe you need to make sure the variable has a backslash on the end of it), then select Apply to Existing Value instead.

Modifier

A modifier is a function that can be applied to the new value (after any variables have been replaced).

The available modifiers are :
- None - the default
- Trim - removes white space from the beginning and end of the string
- TrimLeft - removes white space from the left hand side
- TrimRight - removes white space from the right hand side
- IncludeTrailingBackSlash - if a \ doesn't exist on the end of the string, one will be added
- ExcludeTrailingBackSlash - if there's a \ on the end of the string, it will be removed
- ExtractFileName - the drive and path will be removed, eg. "c:\dir\my file.txt" will result in "my

file.txt"
- ExtractFilePath - the file name part will be removed, eg. "c:\dir\my file.txt" will result in "c:\dir\"
- ExtractFileDrive - the file name and path will be removed, eg. "c:\dir\my file.txt" will result in "c:"
- ExtractFileExt - the extension of the filename will be returned, eg. "c:\dir\my file.txt" will result in ".txt"
- ShortFileName - the filename and path will be converted to 8.3 DOS style format
- AddQuotes - the string will be enclosed in single quotes if the string is not already quoted
- AddDoubleQuotes - the string will be enclosed in double quotes if the string is not already quoted
- StripQuotes - matching begin and end quotes will be removed from the string
- Increment - the integer value will be incremented by 1
- Deincrement - the integer value will be decremented by 1
- LowerCase - the string will be converted to lowercase characters
- UpperCase - the string will be converted to uppercase characters
- Encrypt - Encrypts using blowfish with a hard wired key
- Decrypt - Decrypts values previously encrypted with the Encrypt modifier
- RemoveFileExt - removes the file extension from the value, eg. "file.txt" will result in "file"

Apply To Existing Value

If Apply to Existing Value is checked then the New Value field is ignored and the modifier is applied to the existing variable value.

Force Type

There are some instances, usually when using integer and boolean values in scripting code, that you need to force the variable as a certain type. For example, if you are setting a variable to "100", then you might want this to be forced as an integer type, or maybe as a string type. Use the Default for Automise to use it's best guess what the type should be.

6.26 Virtualization Systems

6.26.1 Microsoft Hyper V

The Hyper V actions allow you to control virtual machines that are hosted on a Microsoft Hyper V Server.

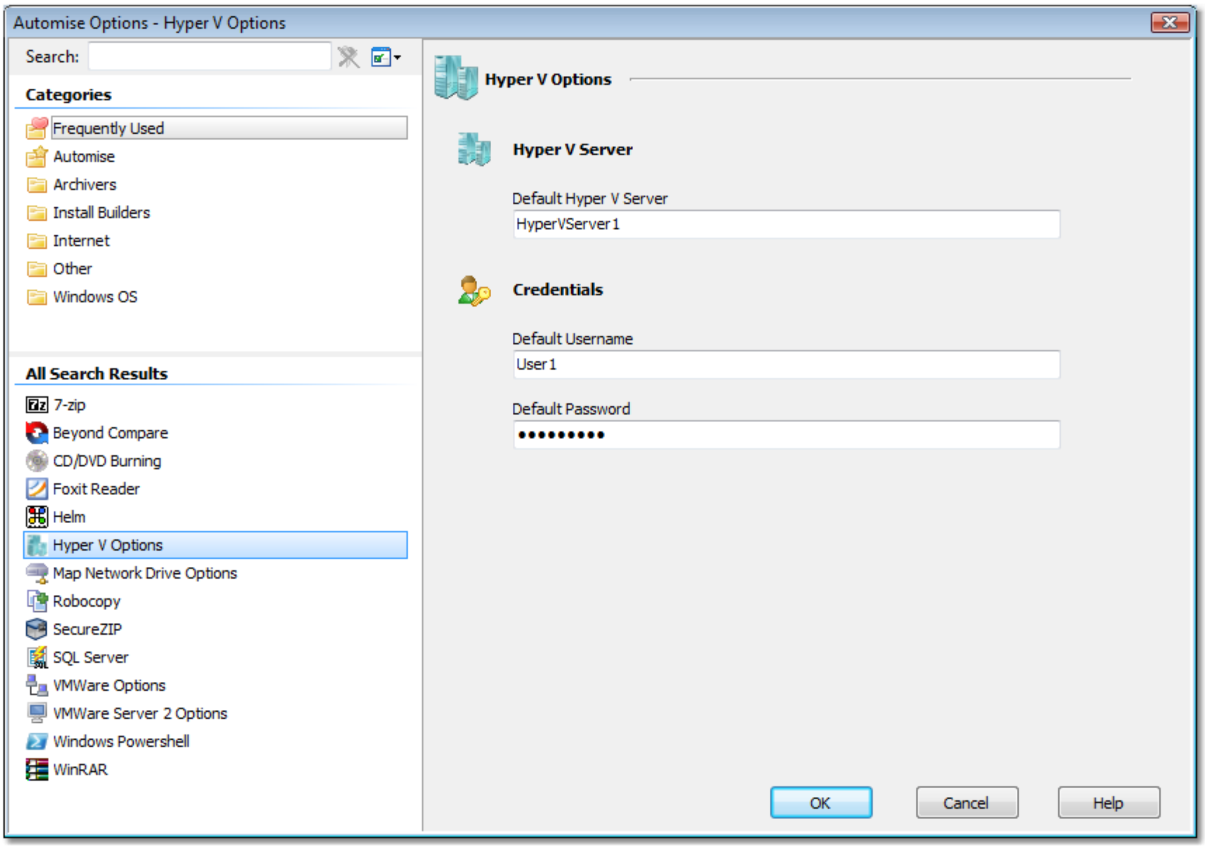
Hyper V Actions

The following actions are available:
- Check VM State
- Start VM
- Stop VM
- Reset VM
- Pause VM
- Suspend VM
- Create VM Snapshot
- Apply VM Snapshot
- Delete VM Snapshot
- List Virtual Machines

Hyper V Options Page

The Hyper V Options page allows you to specify default values for the Hyper V Server name and Credentials. These details are not required but can save time when creating a build that uses multiple Hyper V actions which all use the same Hyper V Server.

For more information about Hyper V Server options see the Hyper V Server page.



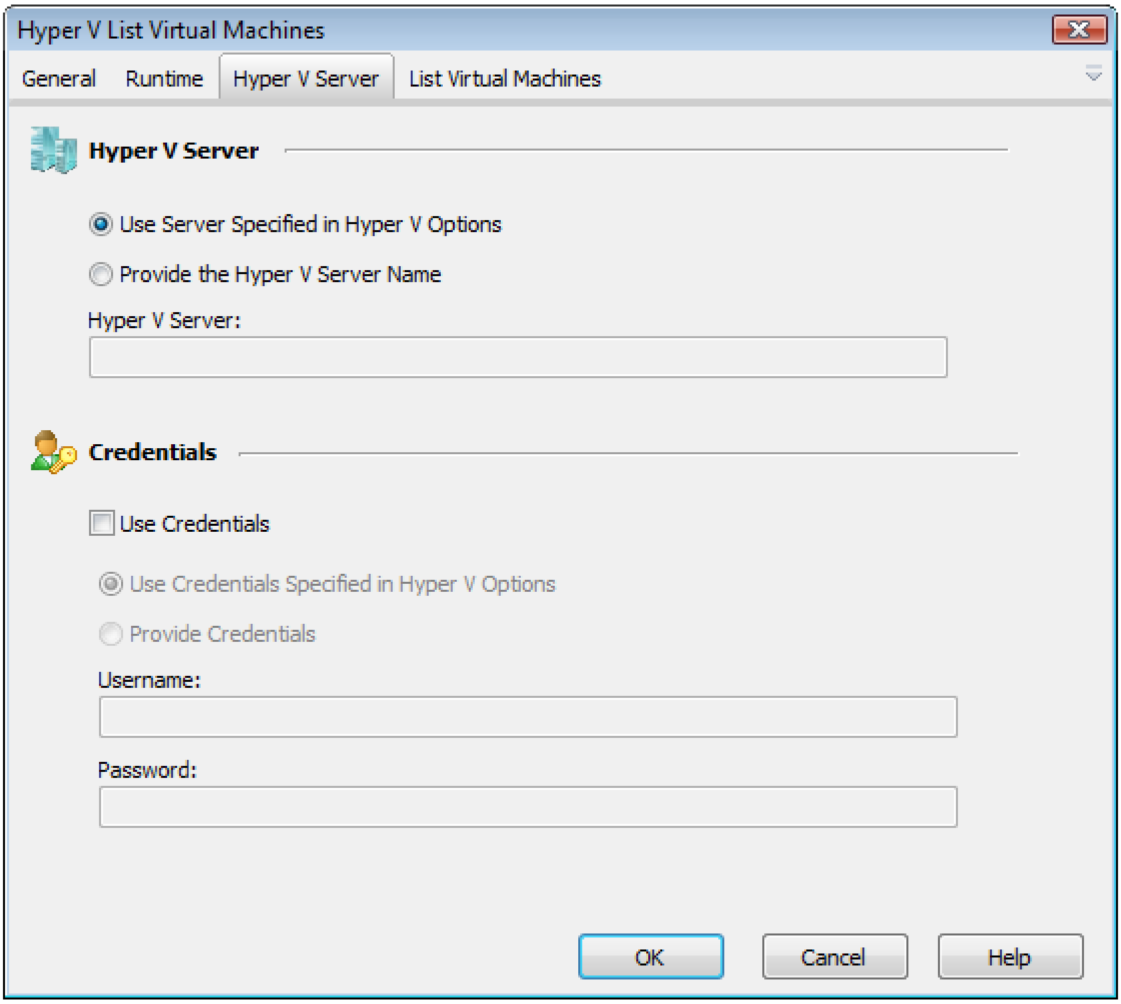
Default Hyper V Server: HyperVServer1

Default Username: User1

Default Password: •••••••••

6.26.1.1 Hyper V Server Page

The *Hyper V Server* page is common to all the Hyper V actions. It allows you to specify the details of the Hyper V Server to connect to in order to manage virtual machines.



Hyper V List Virtual Machines

Hyper V Server

● Use Server Specified in Hyper V Options

○ Provide the Hyper V Server Name

Hyper V Server:

Credentials

□ Use Credentials

● Use Credentials Specified in Hyper V Options

○ Provide Credentials

Username:

Password:

The Hyper V Server is the name of the host machine that you need to connect to when managing virtual machines. There are two options for specifying the Hyper V Server to be used:

- **Use Server Specified in Hyper V Options** - If you have specified a default Hyper V Server on the Hyper V Options page, you can select this option.

- **Provide the Hyper V Server Name** - This option allows you to specify a Hyper V Server to use other than the server specified on the options page.

The Credentials section allows you connect to the Hyper V Server using different credentials to the current account. If the *Use Credentials* option is not selected then the current user's credentials will be used to connect to the Hyper V Server.

When using credentials there are two options available:

- **Use Credentials Specified in Hyper V Options** - If you have specified default credentials on the Hyper V Options page that you wish to use, select this option.

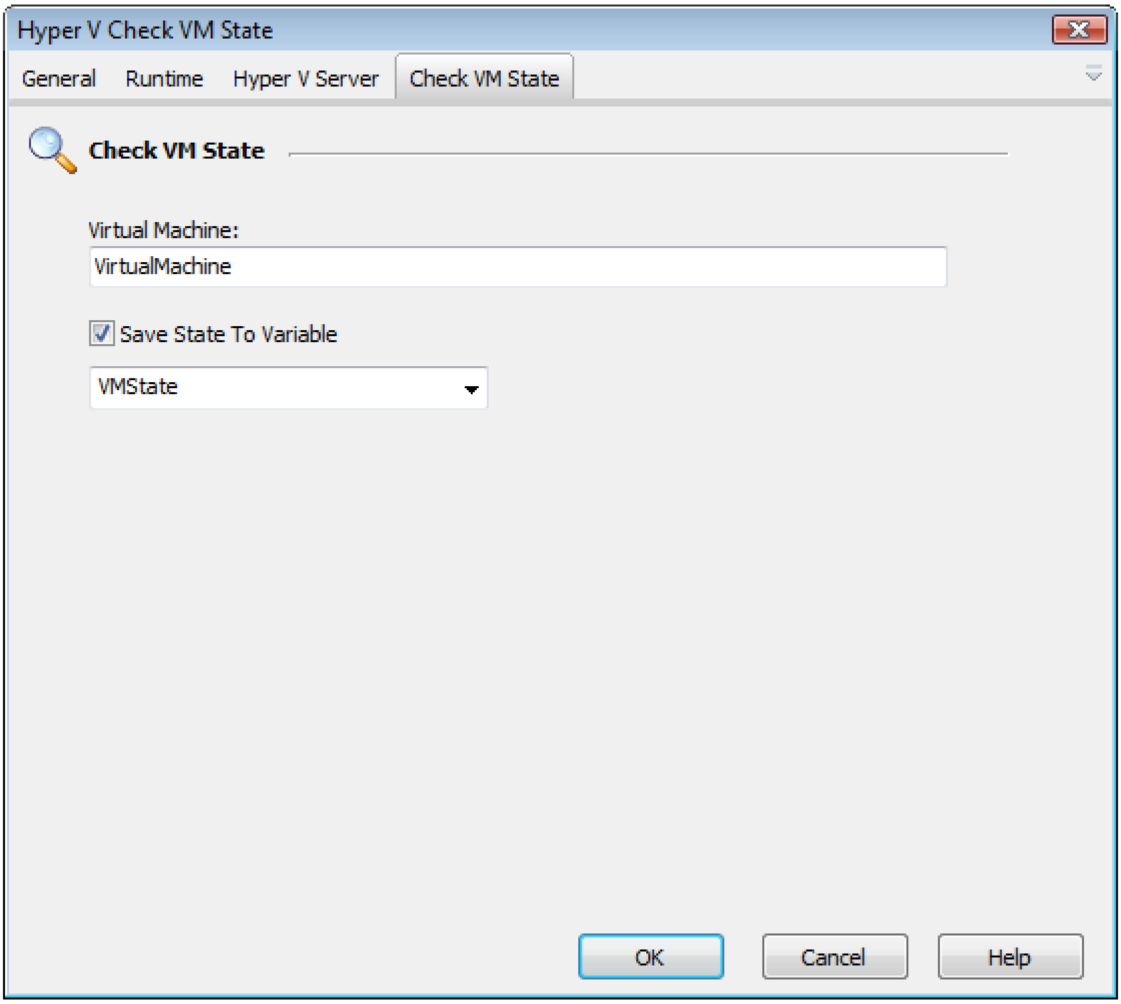
- **Provide Credentials** - This option allows you to specify the credentials to be used.

6.26.1.2 Hyper V Check VM State

The Check VM State action allows you to determine the state of a specific virtual machine running on your Hyper V Management Server.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Check VM State* page provide the name of the virtual machine that you want to check.



Check VM State

Virtual Machine:
VirtualMachine

☑ Save State To Variable

VMState

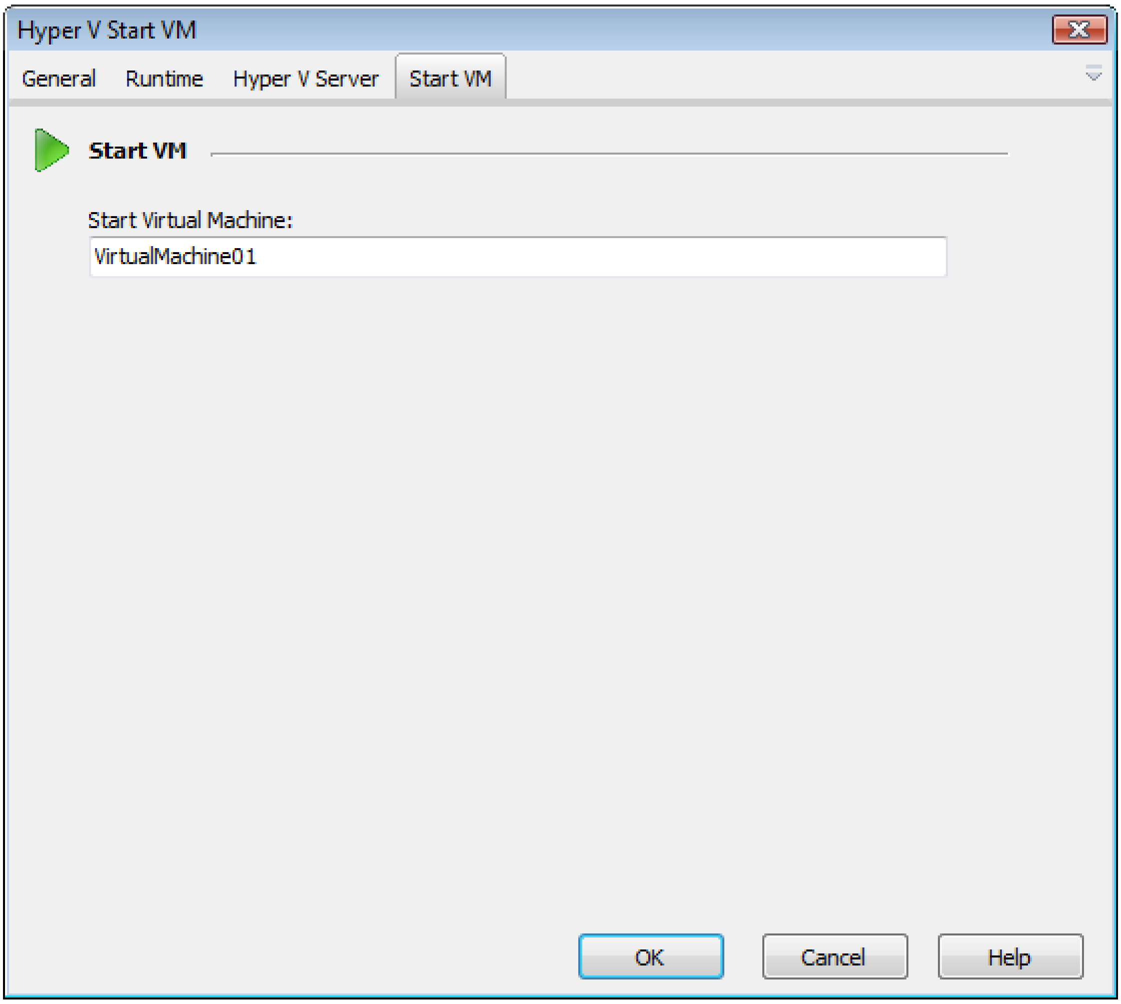
You also have the option saving the machine state into a variable for later use.

6.26.1.3 Hyper V Start VM

The Start VM action allows you to start a virtual machine that is not currently in the running state.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Start VM* page provide the name of the virtual machine to be started.



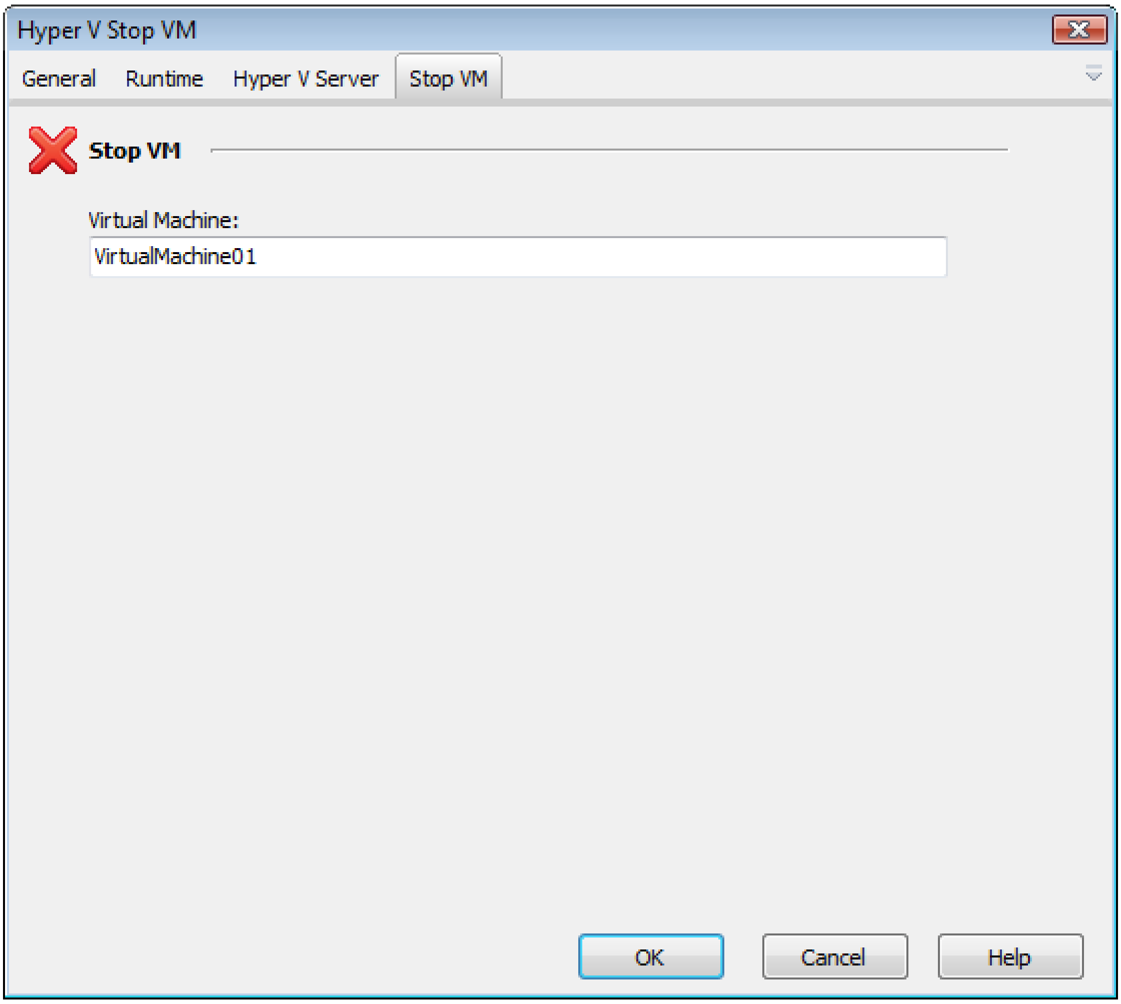
Start Virtual Machine: VirtualMachine01

6.26.1.4 Hyper V Stop VM

The Stop VM action allows you to stop a virtual machine that is not already in the stopped state.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Stop VM* page provide the name of the virtual machine to be stopped.



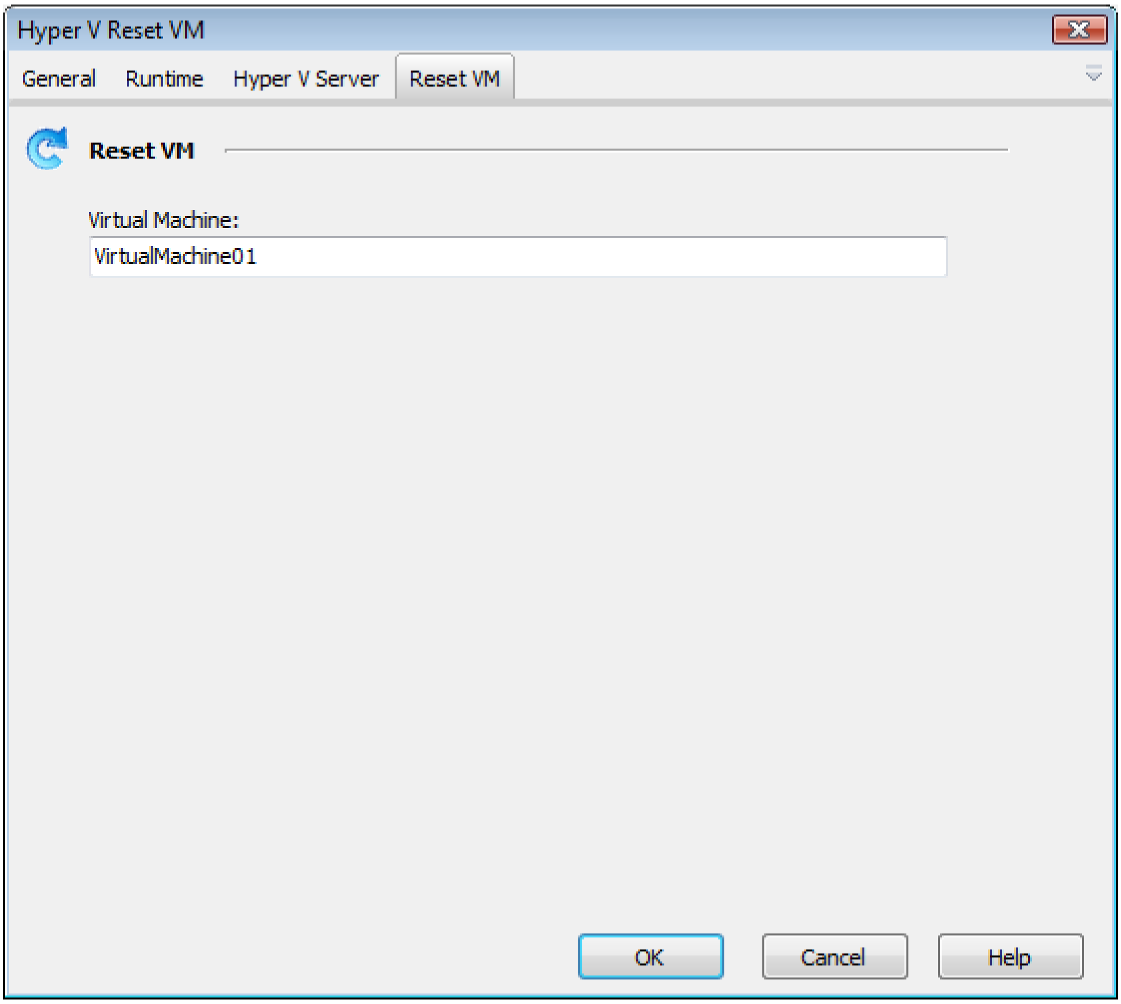
Virtual Machine: VirtualMachine01

6.26.1.5 Hyper V Reset VM

The Reset VM action allows you to reset a virtual machine.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Reset VM* page provide the name of the virtual machine to be reset.



Reset VM

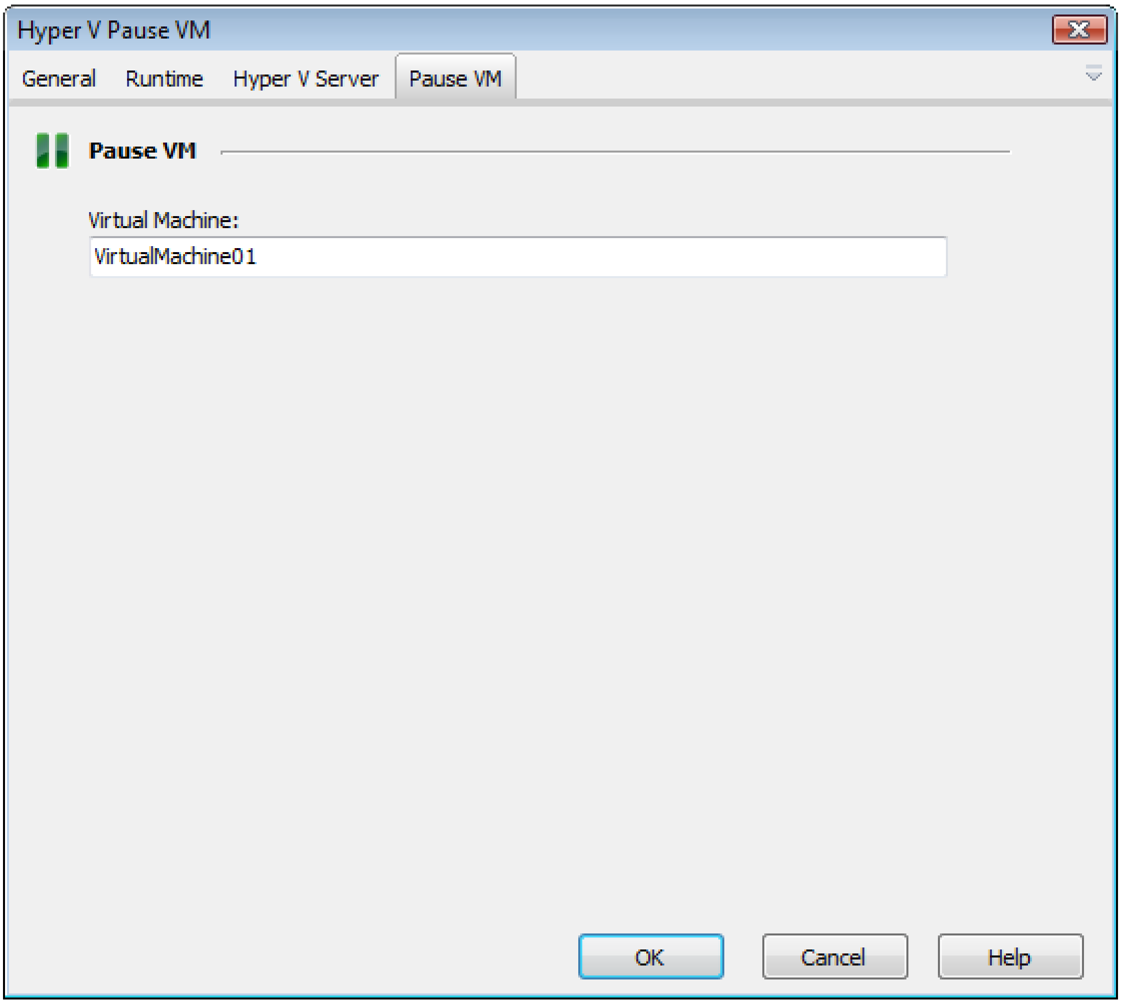
Virtual Machine: VirtualMachine01

6.26.1.6 Hyper V Pause VM

The Pause VM action allows you to pause a virtual machine that is currently in the running state.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Pause VM* page provide the name of the virtual machine to be paused.



Hyper V Pause VM

Pause VM

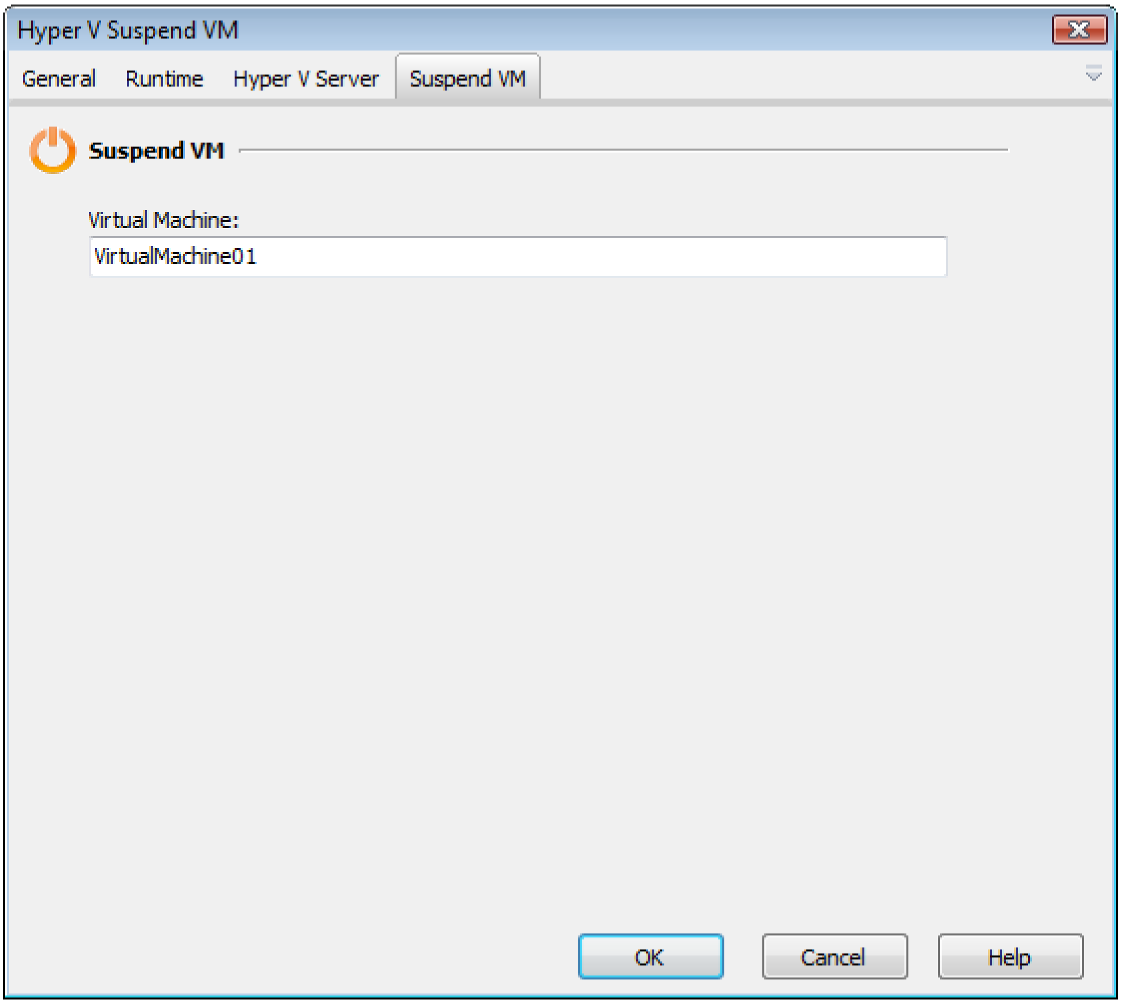
Virtual Machine: VirtualMachine01

6.26.1.7 Hyper V Suspend VM

The Suspend VM action allows you to suspend a virtual machine that is currently in the running state.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Suspend VM* page provide the name of the virtual machine to suspend.



Suspend VM

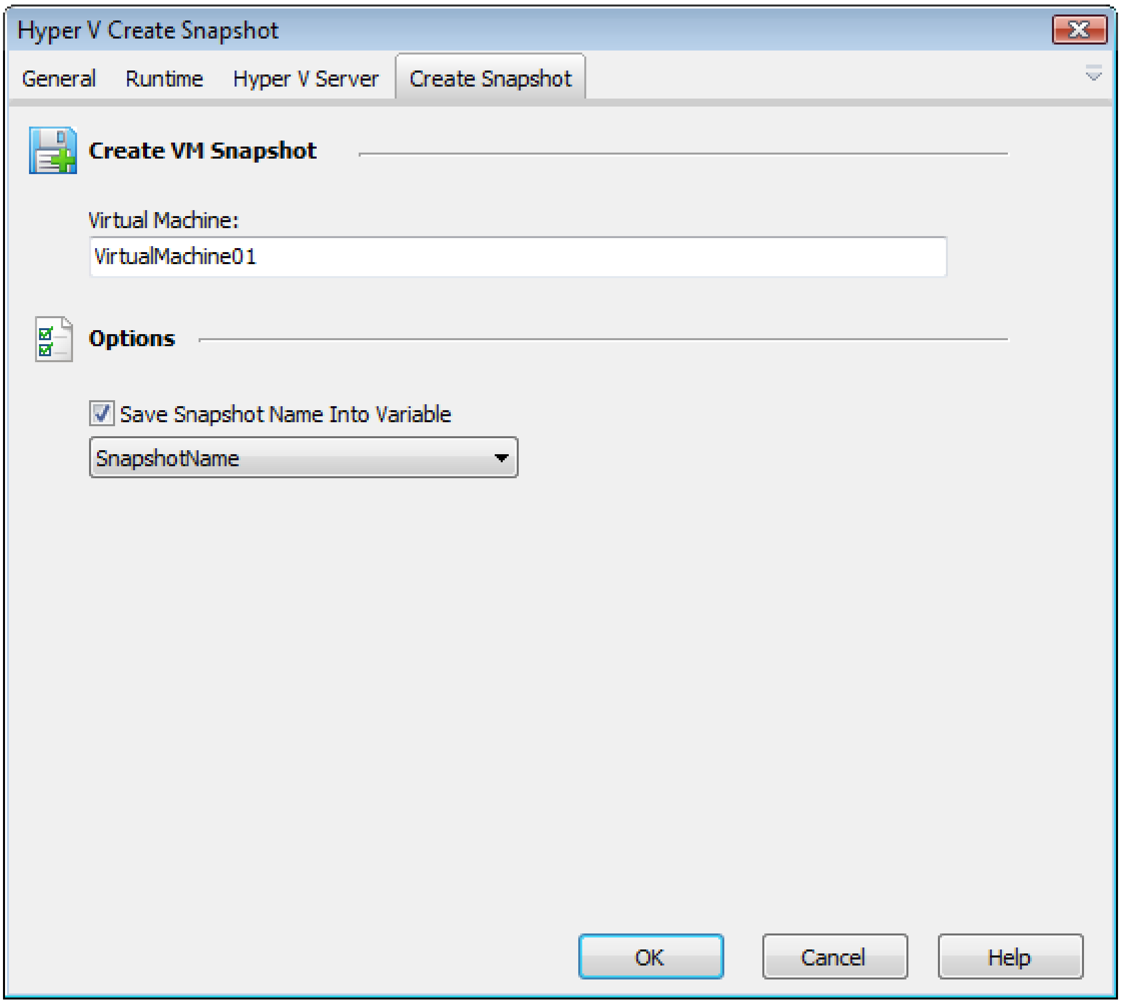
Virtual Machine: VirtualMachine01

6.26.1.8 Hyper V Create VM Snapshot

The Create Snapshot action allows you create a snapshot of the virtual machine's current state that can be restored at a later time.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Create Snapshot* page provide the name of the virtual machine that you want to create a snapshot of.



Hyper V Create Snapshot

Create VM Snapshot

Virtual Machine: VirtualMachine01

Options

☑ Save Snapshot Name Into Variable

SnapshotName

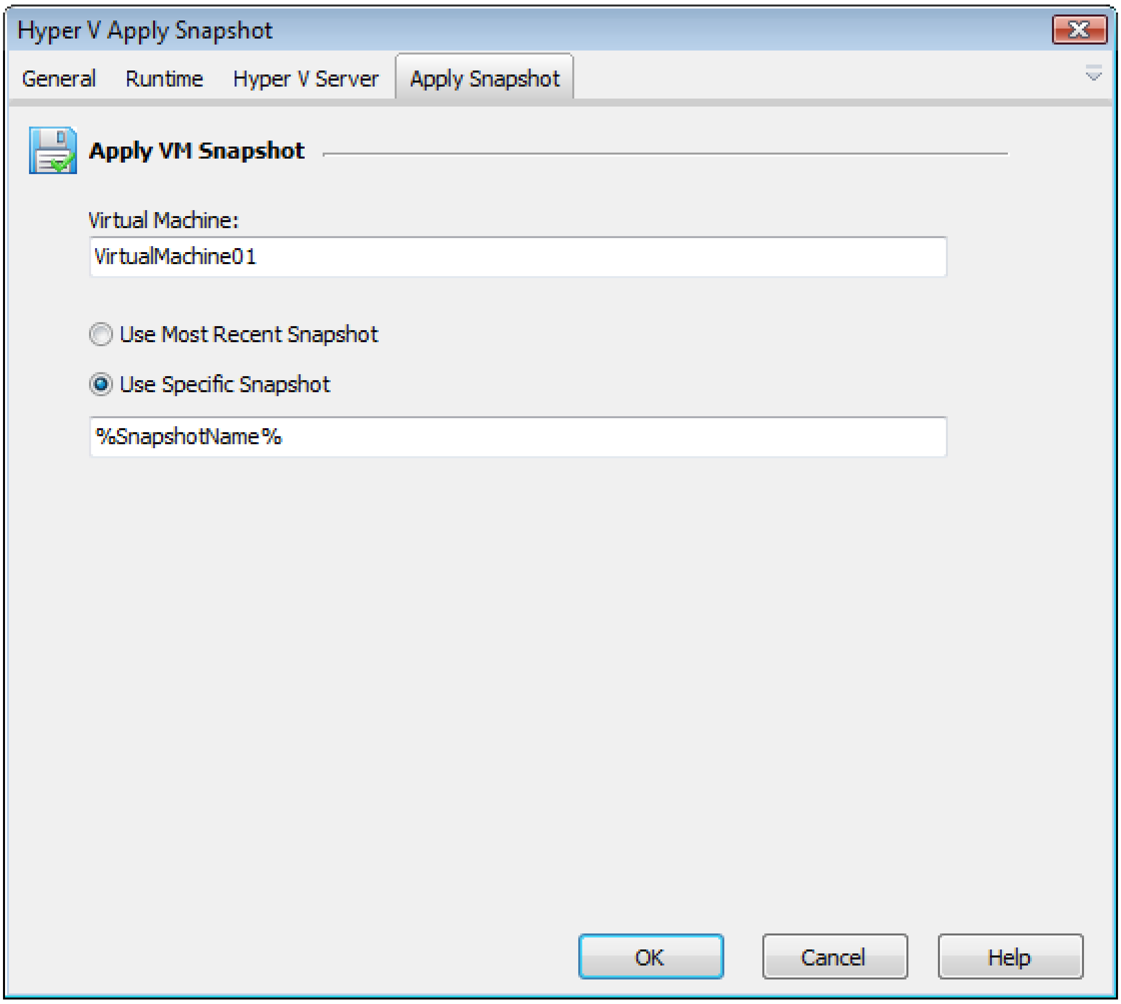
On successful creation of a system snapshot, the name of the new snapshot will be written to the log. By using the **Save Snapshot Name Into Variable** option, the name of the newly created snapshot can be saved into a specified variable for later use.

6.26.1.9 Hyper V Apply VM Snapshot

The Apply Snapshot action allows you to restore a system snapshot of a virtual machine.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Apply Snapshot* page provide the name of the virtual machine that you want to apply the system snapshot to.



Apply VM Snapshot

Virtual Machine:
VirtualMachine01

○ Use Most Recent Snapshot

● Use Specific Snapshot

%SnapshotName%

Select from the following two options:

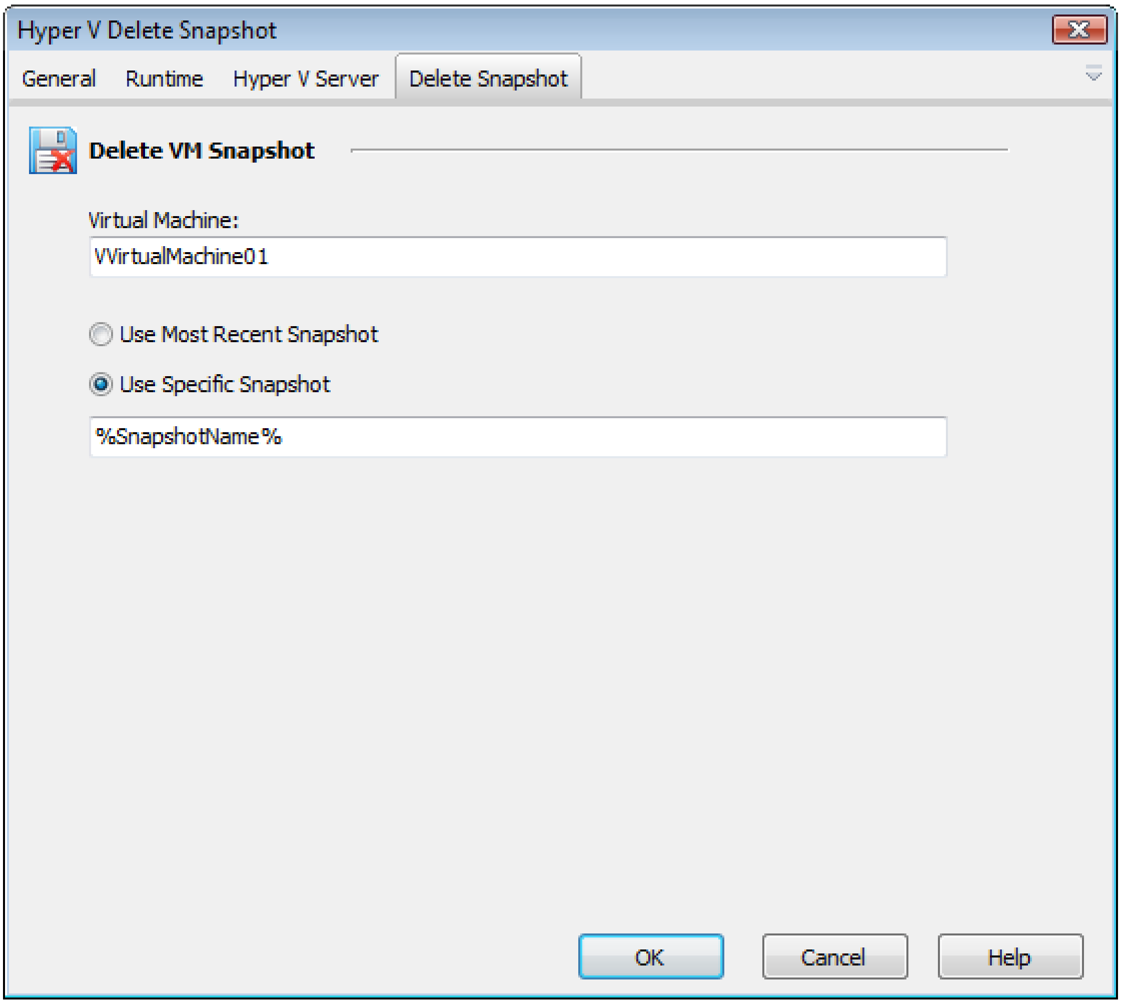
- **Use Most Recent Snapshot** - Use the snapshot that was created most recently.
- **Use Specific Snapshot** - Provide the name of a snapshot to restore the virtual machine to.

6.26.1.10 Hyper V Delete VM Snapshot

The Delete VM Snapshot action allows you to delete a system snapshot

Specify the Hyper V Server settings, see Hyper V Server page.

On the *Delete Snapshot Page* provide the name of the virtual machine that you want to delete the snapshot from.



Hyper V Delete Snapshot

Delete VM Snapshot

Virtual Machine: VVirtualMachine01

○ Use Most Recent Snapshot

● Use Specific Snapshot

%SnapshotName%

Select from the following two options:

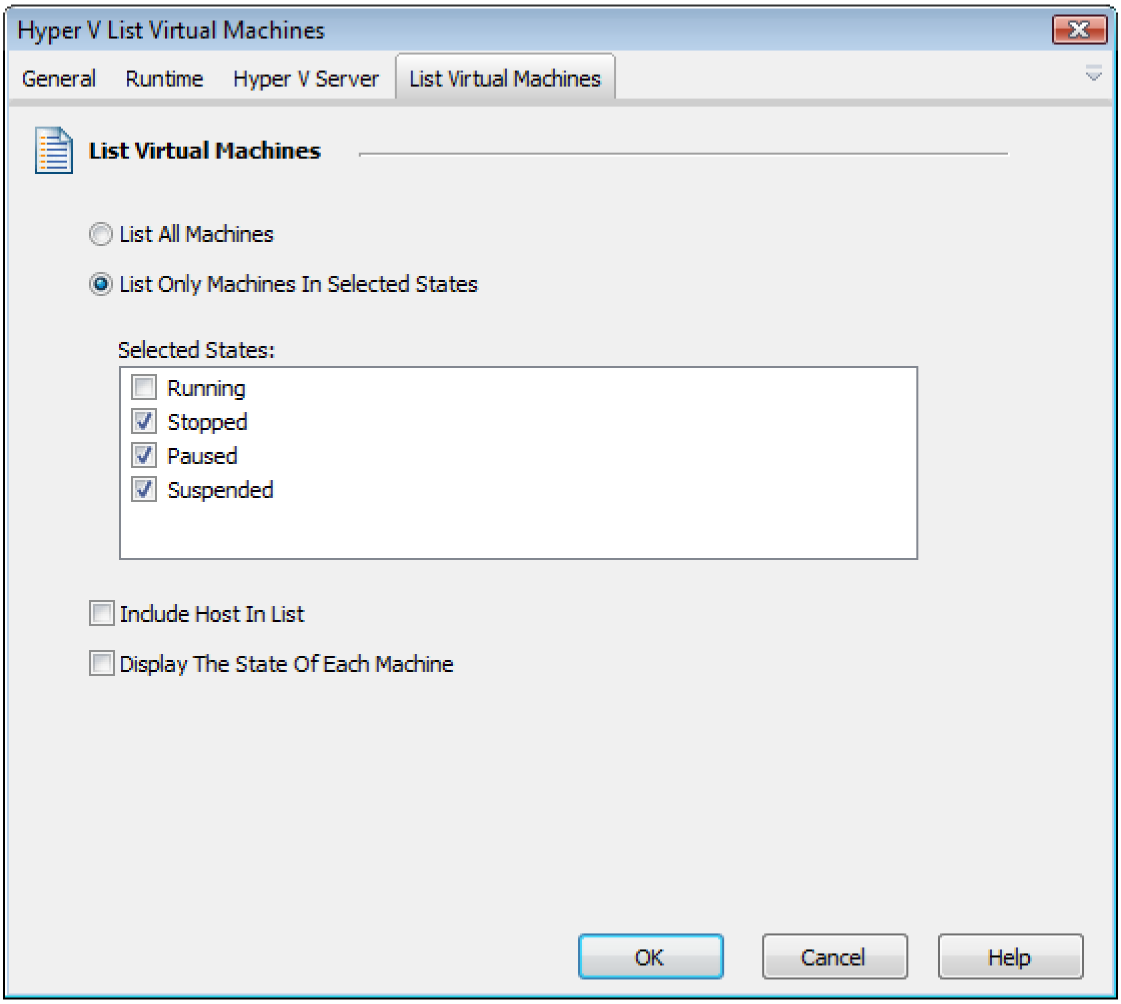
- **Use Most Recent Snapshot** - Delete the snapshot that was created most recently.
- **Use Specific Snapshot** - Provide the name of a snapshot to delete.

6.26.1.11 Hyper V List Virtual Machines

The List Virtual Machines action allows you to list virtual machines on a specific Hyper V Server.

Specify the Hyper V Server settings, see Hyper V Server page.

On the *List Virtual Machines* page select from the following options:



List Virtual Machines

○ List All Machines

● List Only Machines In Selected States

Selected States:

□ Running
☑ Stopped
☑ Paused
☑ Suspended

□ Include Host In List

□ Display The State Of Each Machine

- **List All Machines** - This option will list all the virtual machines that reside on the specified Hyper V Server, regardless of state.
- **List Only Machines In Selected States** - This option will all virtual machines that reside on the specified Hyper V Server, where the virtual machines state is in the selected states list.

The following options are also available:

- **Include Host In List** - This option will include the Hyper V Server in the list of machines.
- **Display The State Of Each Machine** - This option will output the state of each machine listed.

6.26.2 Microsoft Virtual Server

The Virtual Server actions provide an interface to control Microsoft's Virtual Server virtualization product.

Note: You will need the Virtual Server COM Client API installed on your computer in order to use the Virtual Server actions. The COM API is installed as part of a standard Virtual Server installation.

Virtual Machine Property Page

All Virtual Server actions which act on a single Virtual Machine contain the following property page:

Virtual Machine

Enter the name of the virtual machine to connect to. Click the "Load Virtual Machine List" button to load a list of registered virtual machines into the combo box.

Remote Virtual Server (Optional)

If the Virtual Server is located on a remote computer, enter the server name here. The server will be contacted via DCOM (Distributed COM) using the same credentials as the current logged in user. Virtual Server must be configured to allow remote administration.

You will still need the Virtual Server COM Client API installed on the local machine (although you do not need a full Virtual Server installation.)

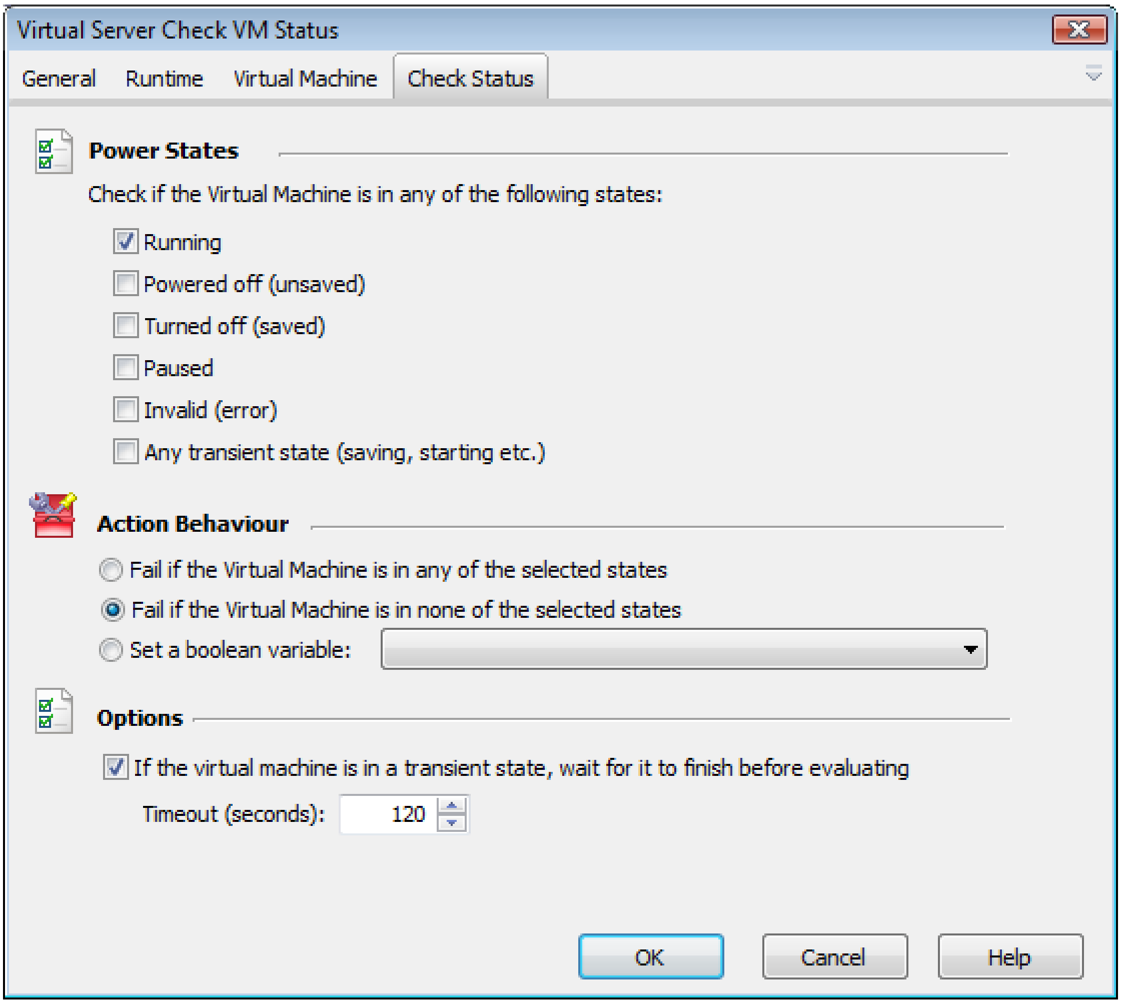
If the Virtual Server is running on the local machine, leave this field blank.

6.26.2.1 Virtual Server Check VM Status

The Virtual Server Check VM Status allows you to modify the behaviour of the action based on the current status of a Virtual Machine.

Virtual Machine Property Page

See the Virtual Server overview topic for details.



Virtual Server Check VM Status

Power States

Check if the Virtual Machine is in any of the following states:

☑ Running
☐ Powered off (unsaved)
☐ Turned off (saved)
☐ Paused
☐ Invalid (error)
☐ Any transient state (saving, starting etc.)

Action Behaviour

○ Fail if the Virtual Machine is in any of the selected states
● Fail if the Virtual Machine is in none of the selected states
○ Set a boolean variable:

Options

☑ If the virtual machine is in a transient state, wait for it to finish before evaluating

Timeout (seconds): 120

Power States

Select the Virtual Machine states you would like to check for.

- The Invalid state should never be encountered unless the Virtual Machine has been deleted while the action is running.

- "Any transient state" is any time when an operation is being performed on a Virtual Machine (ie saving, resuming, deleting.)

Action Behaviour

"Fail if the Virtual Machine is in any of the selected states"
"Fail if the Virtual Machine is in none of the selected states"

These options control the overall result of the action (success or failure) based on the state of the Virtual Machine.

In the above example, the action will only succeed if the Virtual Machine is powered down in a saved state (note that for transient states, the action will wait them out - see below.)

"Set a boolean variable" - Enter the name of a variable to have it set to True if the virtual machine is in one of the chosen states, or False otherwise.

Options

"If the virtual machine is in a transient state, wait for it to finish before evaluating"

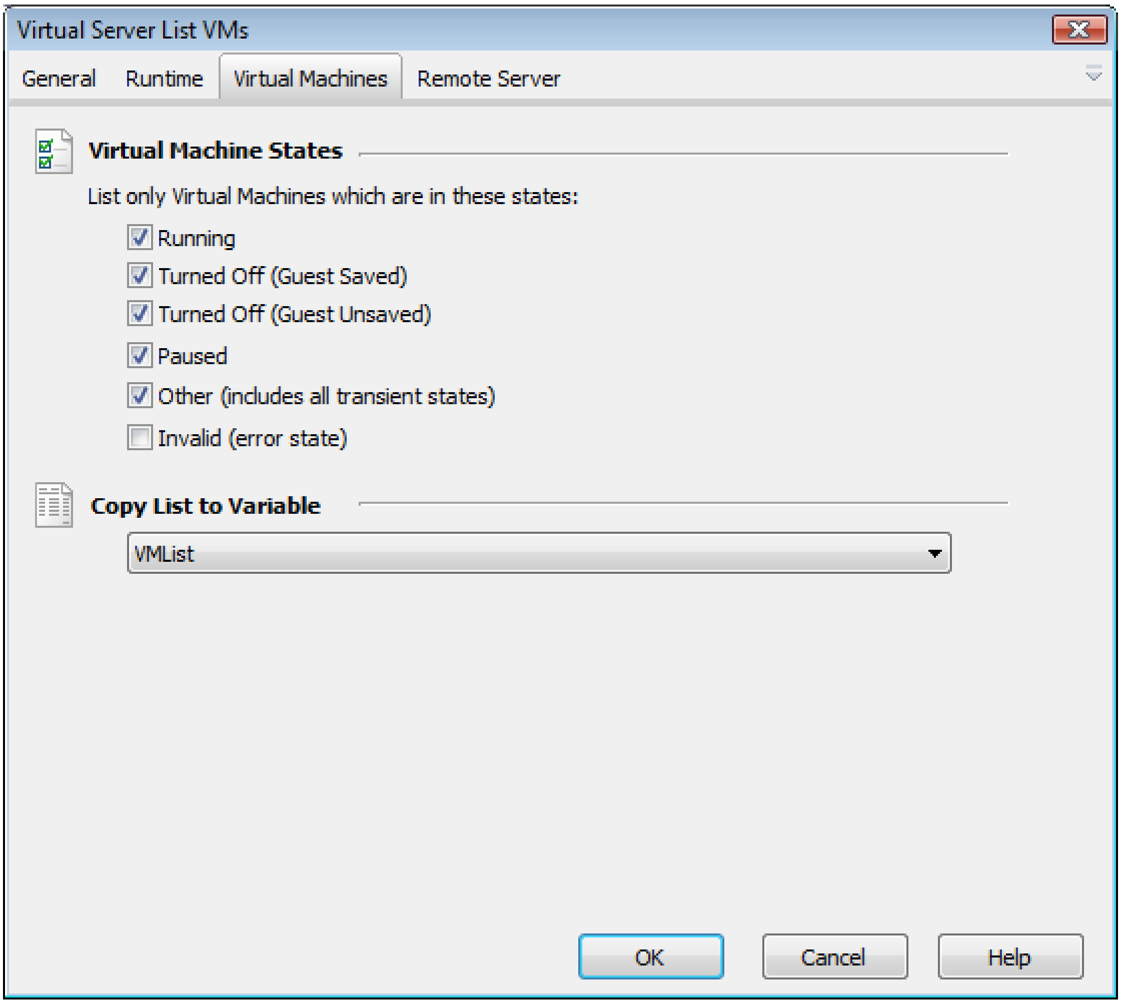
If this option is selected, the action will wait for any "transient" state to have completed before it goes ahead to check the state of the VM. If the action is still in the transient state when the timeout is reached, the action will fail.

(Note that this option does not apply if the "Any transient state" option is selected above.)

6.26.2.2 Virtual Server List VMs

The Virtual Server List VMs action creates a list of Virtual Machines which are in one or more selected states.

The list will be printed to the action's log (along with the state of each action), and can also optionally be saved to a %ProductName% variable.



Virtual Machine States

List only Virtual Machines which are in these states:

- ☑ Running
- ☑ Turned Off (Guest Saved)
- ☑ Turned Off (Guest Unsaved)
- ☑ Paused
- ☑ Other (includes all transient states)
- □ Invalid (error state)

Copy List to Variable

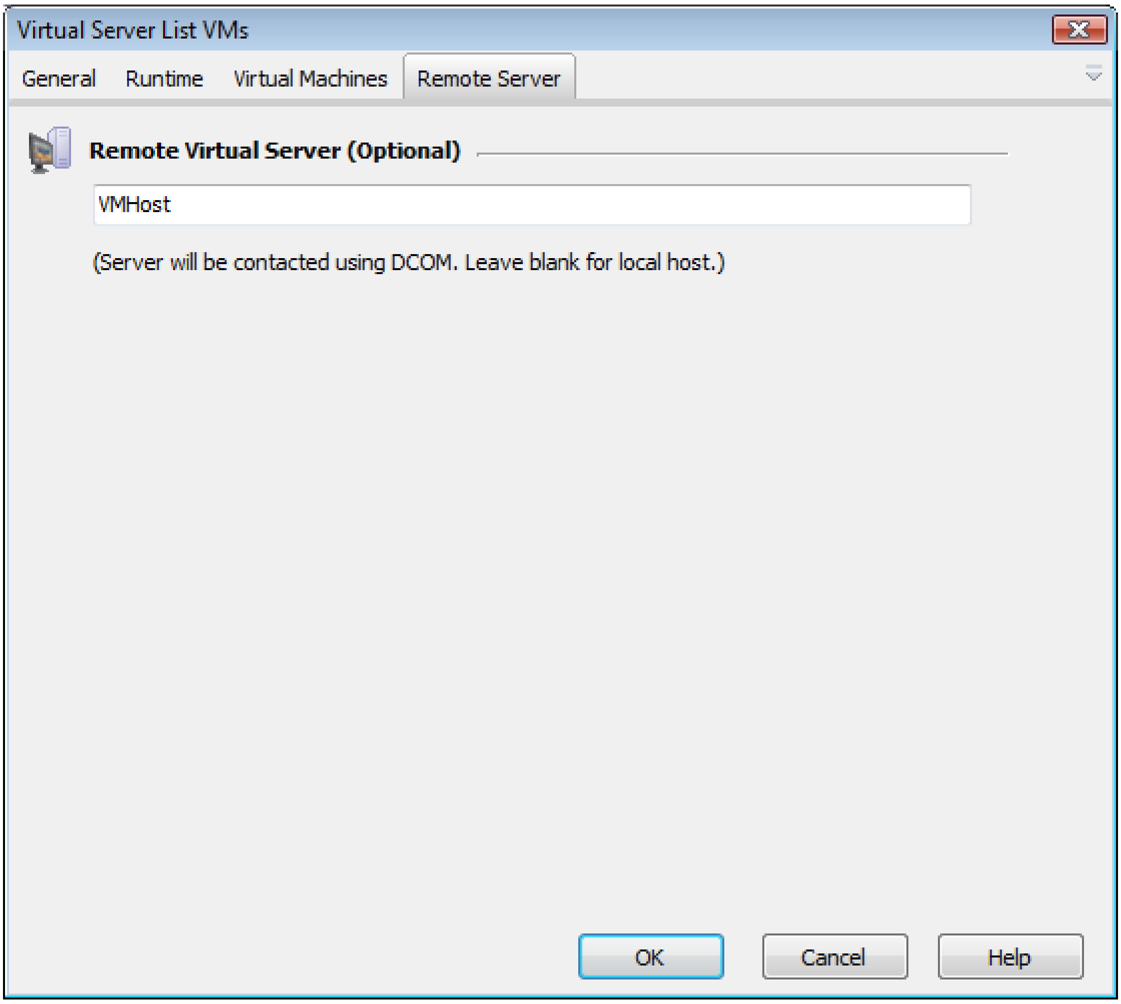
VMList

Virtual Machine States

Specify the states that you want included in the list.

Copy List to Variable

Optionally, specify a Automise variable to write the list of Virtual Machine names to. This list could then be used in a List Iterator.



VMHost

(Server will be contacted using DCOM. Leave blank for local host.)

If the virtual server is running on a remote machine, specify the host name here.

Automise will attempt to connect via DCOM, using the current running user's credentials.

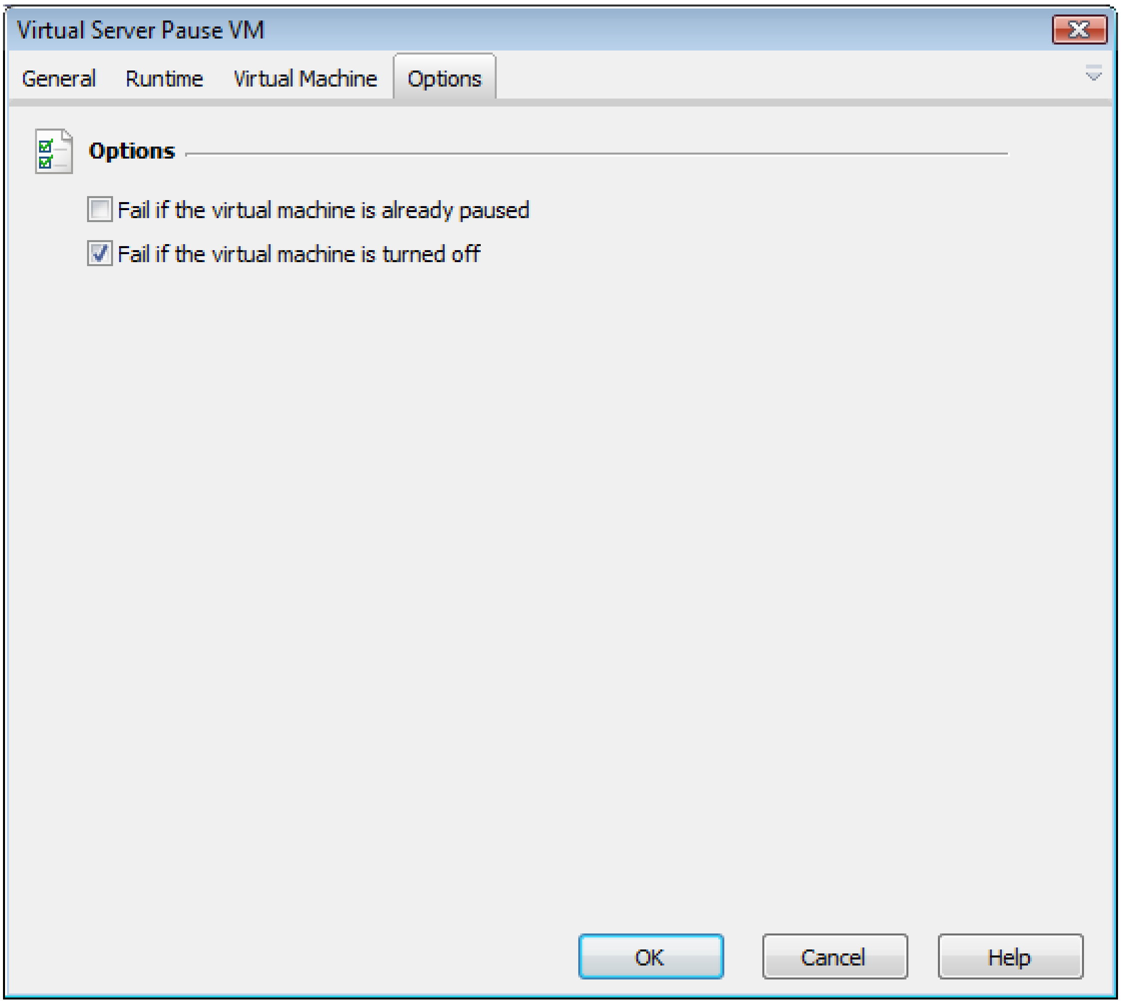
6.26.2.3 Virtual Server Pause VM

The Virtual Server Pause VM action pauses a Virtual Machine (note that if you want to save the contents of a virtual machine and free it from memory, you can use the Save VM State action.)

To resume a paused VM, use the Virtual Server Start VM action.

Virtual Machine Property Page

See the Virtual Server overview topic for details.



Options

□ Fail if the virtual machine is already paused

☑ Fail if the virtual machine is turned off

"Fail if the virtual machine is already paused"

If this option is selected and the virtual machine is already paused, the action will fail. Otherwise, it will continue.

"Fail if the virtual machine is turned off"

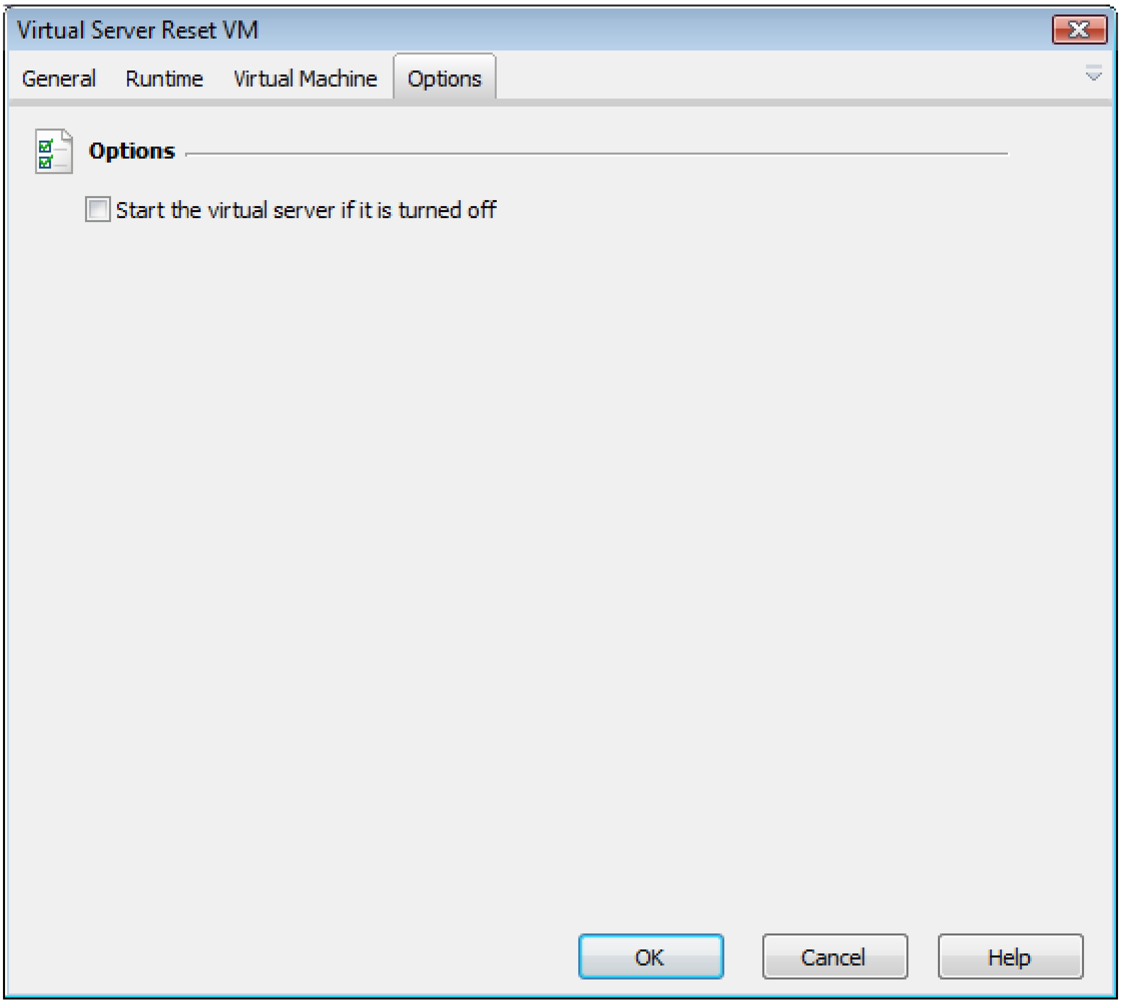
If this option is selected and the virtual machine is powered down, the action will fail. Otherwise, it will continue.

6.26.2.4 Virtual Server Reset VM

The Virtual Server Reset VM action forcibly resets the target virtual machine (the equivalent of pressing the reset button.)

Virtual Machine Property Page

See the Virtual Server overview topic for details.



Options

□ Start the virtual server if it is turned off

"Start the virtual server if it is turned off"

If this option is selected and the Virtual Machine is turned off, the action will turn it on.

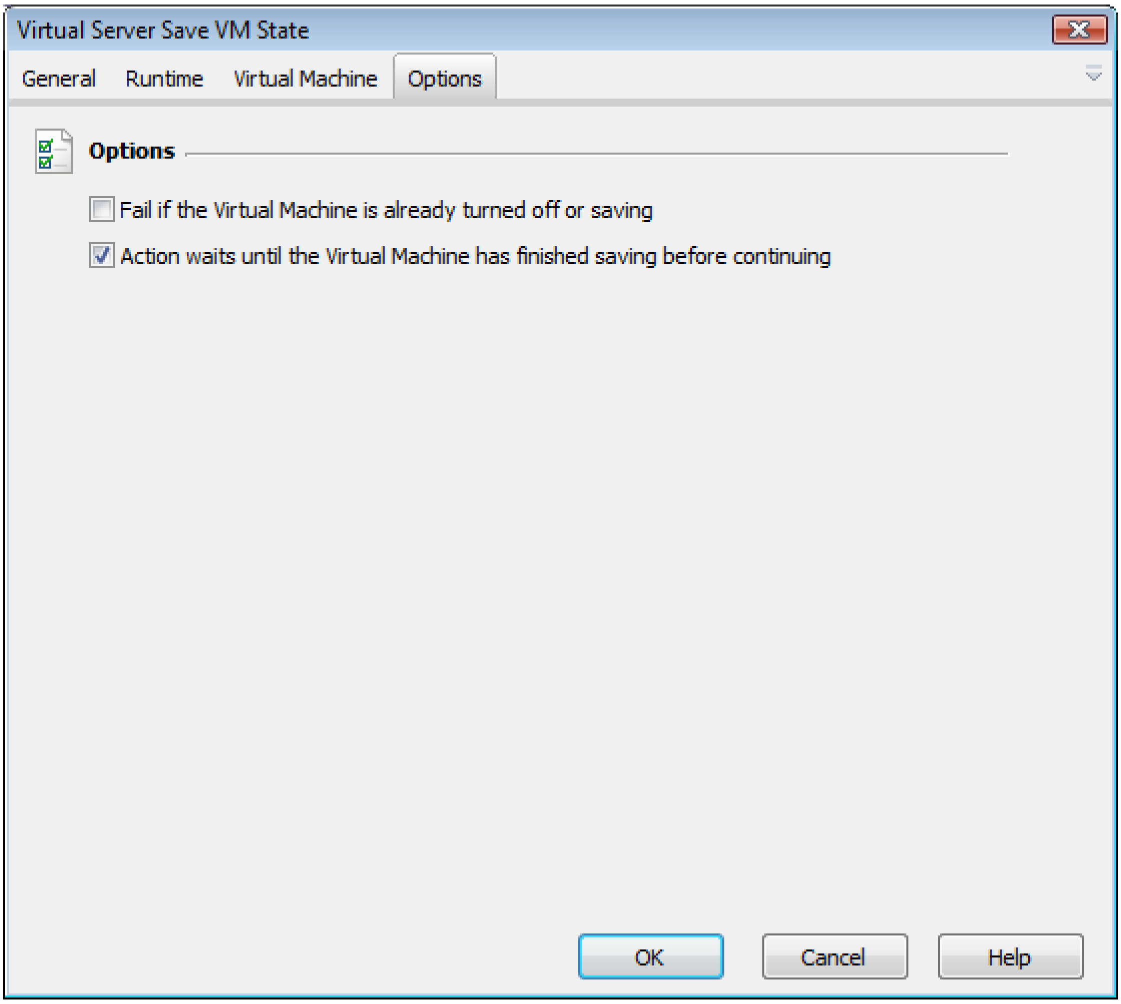
If this option is not selected and the Virtual Machine is turned off, the action will fail.

6.26.2.5 Virtual Server Save VM State

The Virtual Server Save VM State action will save the state of a running Virtual Machine and then stop it.

Virtual Machine Property Page

See the Virtual Server overview topic for details.



Options

□ Fail if the Virtual Machine is already turned off or saving

☑ Action waits until the Virtual Machine has finished saving before continuing

"Fail if the Virtual Machine is already turned off or saving"

If this option is selected, the action will fail if the target VM is already powered down or in the process of saving.

"Action waits until the Virtual Machine has finished saving before continuing"

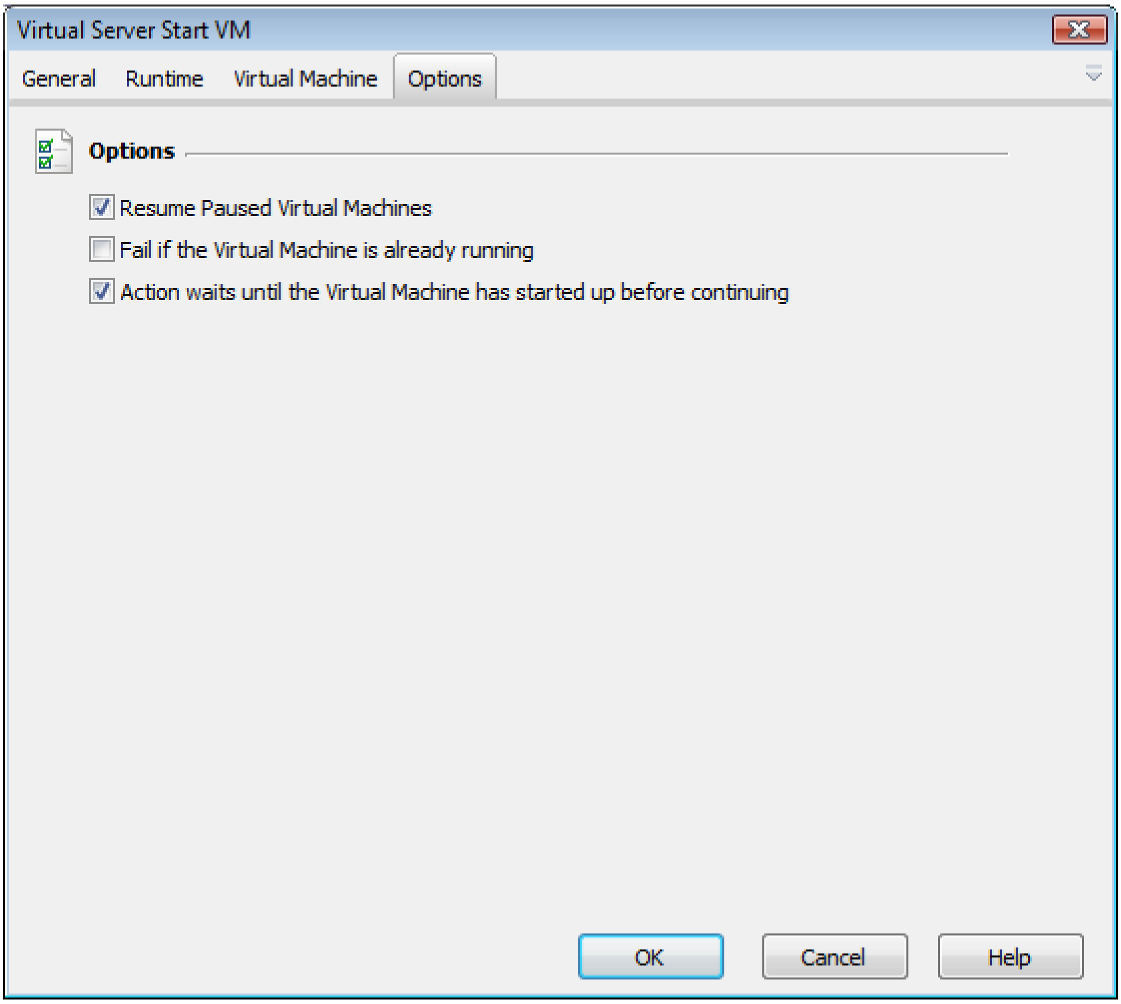
If this option is selected, the action will wait until the Virtual Machine has completed saving and powered down before continuing.

6.26.2.6 Virtual Server Start VM

The Virtual Server Start VM action will start a stopped (or saved) Virtual Machine, or (optionally) resume a paused Virtual Machine.

Virtual Machine Property Page

See the Virtual Server overview topic for details.



Options

☑ Resume Paused Virtual Machines

□ Fail if the Virtual Machine is already running

☑ Action waits until the Virtual Machine has started up before continuing

"Resume Paused Virtual Machines"

If this option is selected and the target VM is paused, it will be resumed.

"Fail if the Virtual Machine is already running"

If this option is selected and the target VM is already running, the action will fail. Otherwise, it will contain.

"Action waits until the Virtual Machine has started up before continuing"

Once the signal is sent to start the stopped VM, it can take some time to start up (especially when saved.)

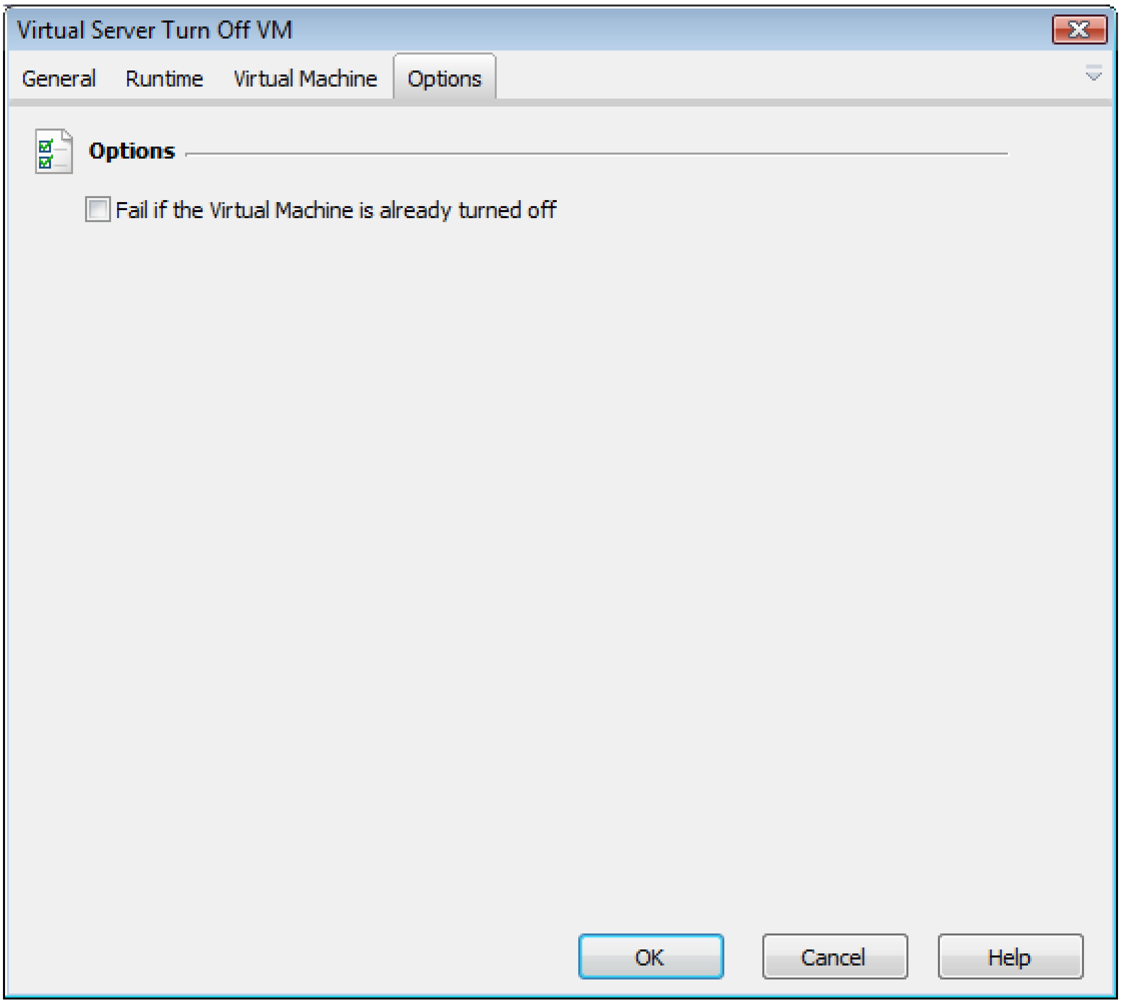
If this option is selected, the action will not continue until the Virtual Machine is up and running. If you are not accessing the Virtual Machine immediately, you can disable this option.

6.26.2.7 Virtual Server Turn Off VM

The Virtual Server Turn Off VM action will power down the target VM instantaneously (if you don't want to lose the contents of the Guest operating system, it is recommended you use the Save VM State action.)

Virtual Machine Property Page

See the Virtual Server overview topic for details.



□ Fail if the Virtual Machine is already turned off

"Fail if the Virtual Machine is already turned off"

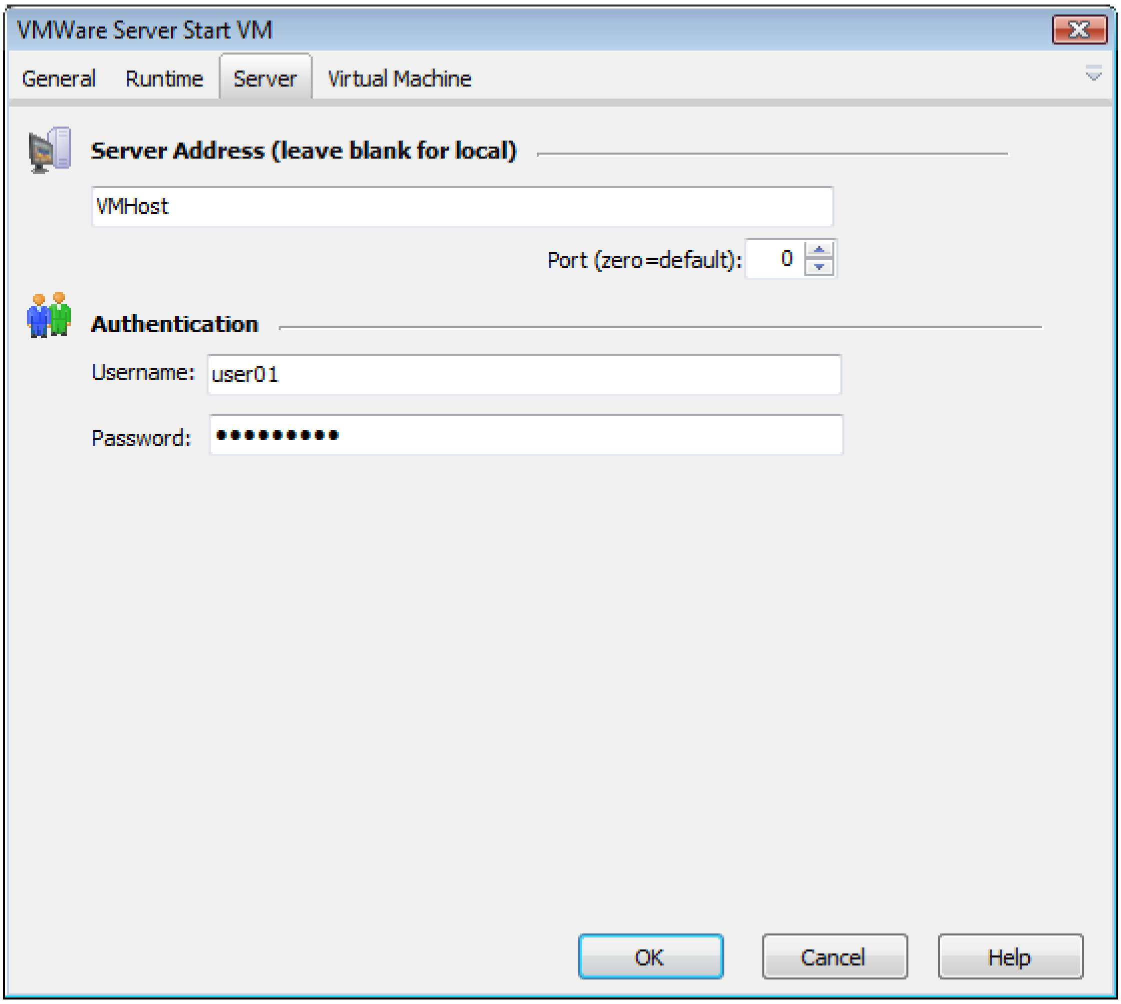
If this option is selected and the VM is already powered down, the action will fail (otherwise it will continue.)

6.26.3 VMWare Server

VMWare Server has a powerful COM interface which can be used to administer Virtual Machines. The VMWare Server actions allow you to use this interface.

Note: You will need the VMWare Server COM Client API installed on your computer in order to use the VMWare Server actions. The COM Client API installed as part of a standard VMWare Server installation.

Server Property Page



VMWare Server Start VM

Server Address (leave blank for local)

VMHost

Port (zero=default): 0

Authentication

Username: user01

Password: ●●●●●●●●●

Server Address

If connecting to a remote instance of VMWare Server, enter the server name (and, optionally, a custom port) in this section.

Note that you will still need the COM Client API (although not necessarily a full VMWare Server installation) installed on the local machine.

Authentication

If you wish to VMWare Server using non-standard credentials, enter them here.

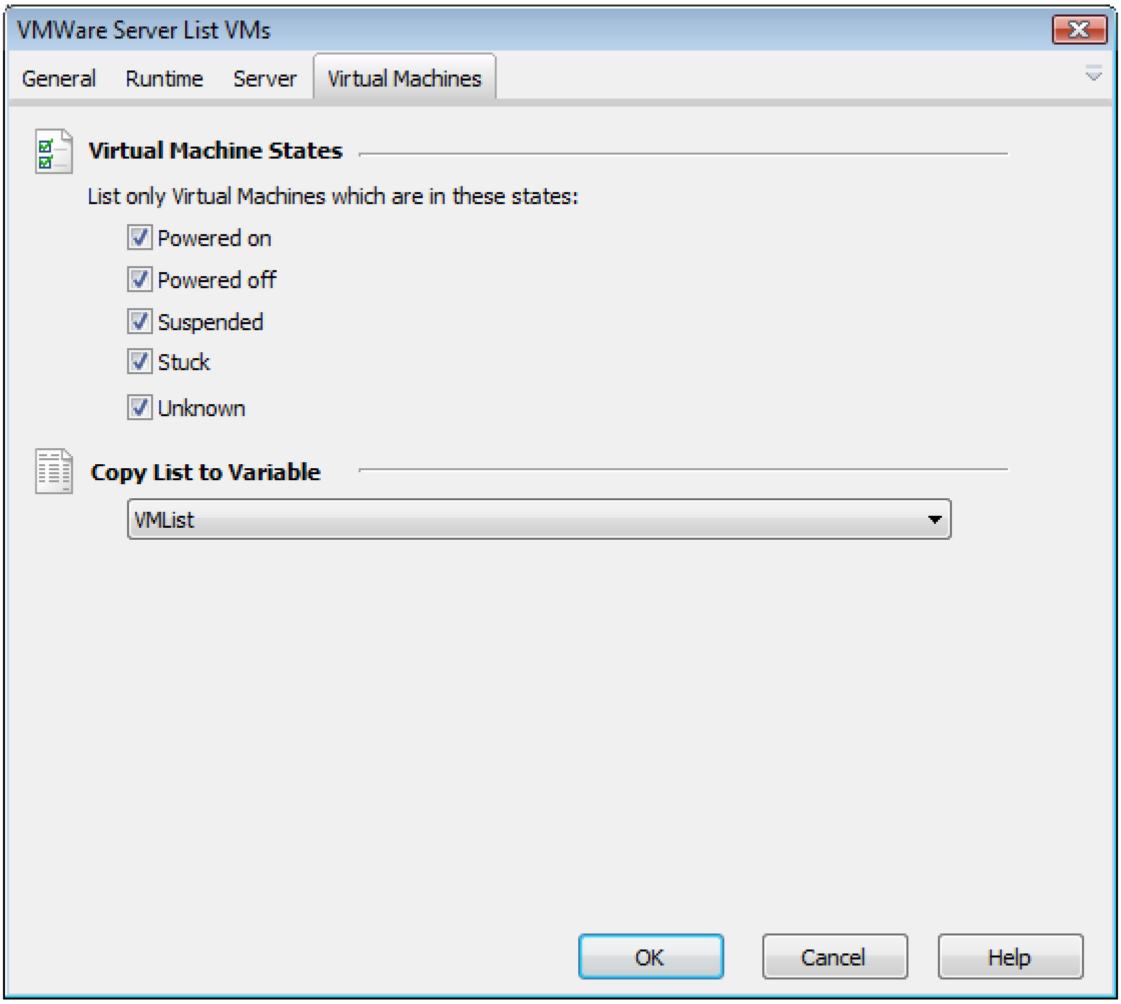
If left blank, the connection will use the credentials of the currently logged in user (recommended.)

6.26.3.1 VMWare Server List VMs

The VMWare Server List VMs action creates a list of registered Virtual Machines. The list is output to the log, and can be optionally written to a variable.

Server Property Page

See the VMWare Server overview topic.



Virtual Machine States

List only Virtual Machines which are in these states:

☑ Powered on

☑ Powered off

☑ Suspended

☑ Stuck

☑ Unknown

Copy List to Variable

VMList

Virtual Machine States

Select the Virtual Machine states you would like to list. The above example will list all VMs which are not currently running.

Copy List to Variable

If a variable name is entered here, the list of Virtual Machine names will be copied to a % ProductName% variable. The list is suitable for use in a List Iterator.

6.26.3.2 VMWare Server Start / Stop / Reset / Suspend VM

This group of VMWare actions modifies the state of a running or stopped Virtual Machine:

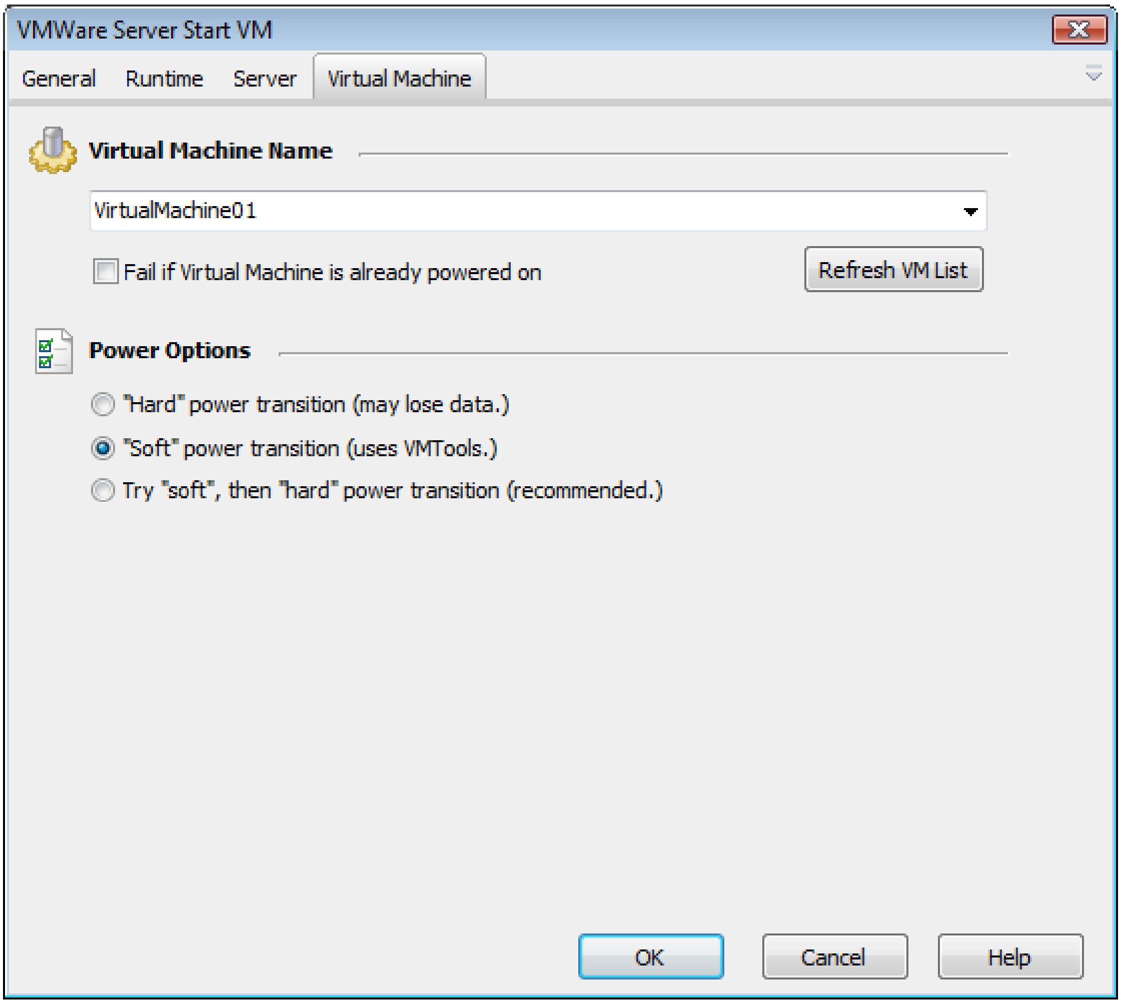
- **VMWare Server Start** starts a virtual machine which is suspended or powered off.
- **VMWare Server Stop** powers off a running virtual machine.
- **VMWare Server Reset** resets a running virtual machine.
- **VMWare Server Suspend** suspends a running virtual machine.

All of these VMWare actions contain the following property pages:

Server Property Page:

See the VMWare Server overview topic.

Virtual Machine Property Page:



Virtual Machine Name

VirtualMachine01

□ Fail if Virtual Machine is already powered on

Power Options

○ "Hard" power transition (may lose data.)

● "Soft" power transition (uses VMTools.)

○ Try "soft", then "hard" power transition (recommended.)

Virtual Machine Name

Enter the name of the Virtual Machine to operate on (in the above example, the VM name is stored in a variable.) Click the Refresh VM List button to populate the combo box with a list of registered Virtual Machines.

"Fail if..."

This option varies between actions, but it implies whether or not to fail if the Virtual Machine is already in the target state (ie, in the above example, if the Virtual Machine is already started.) If this option is not checked, the action will continue as normal if the target state has already been entered when the action runs.

Power Options

Power transitions in VMWare can be hard (ie made directly against the virtual hardware), or soft (using VMTools to attempt to make the operation less damaging.)

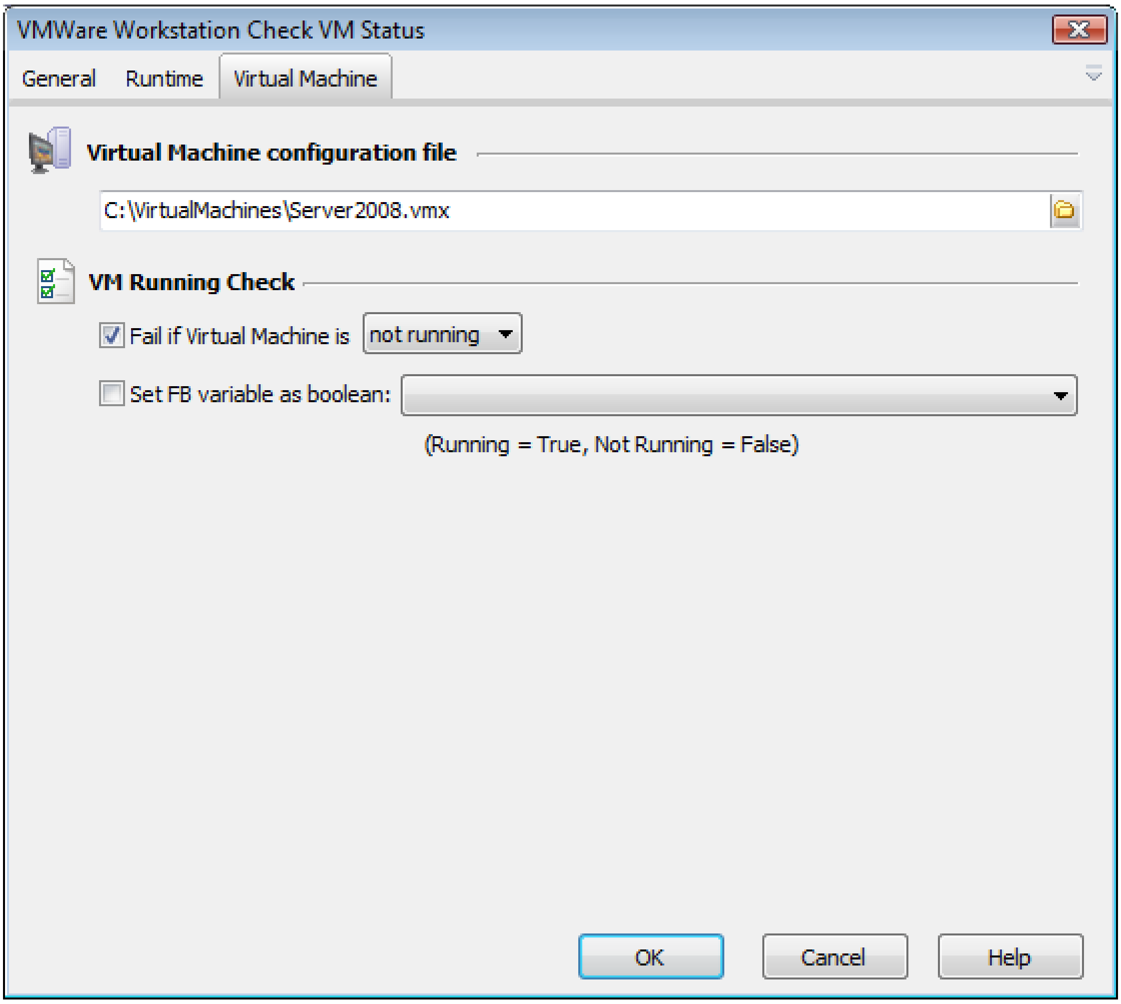
By default, the VMWare Server actions will try a safer soft operation, but revert to a hard operation if the soft operation fails.

6.26.4 VMWare Workstation

6.26.4.1 Check Virtual Machine Status

The Check Virtual Machine Status action allows you to check whether a Virtual Machine is currently running or not.

Note: Due to a bug in VMWare (current in version 5.5), this action sometimes gives incorrect results if more than one VMWare instance is running. Use with caution.



VMWare Workstation Check VM Status

Virtual Machine configuration file: C:\VirtualMachines\Server2008.vmx

VM Running Check

☑ Fail if Virtual Machine is: not running

□ Set FB variable as boolean:

(Running = True, Not Running = False)

Virtual Machine Configuration File

Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team file (.vmtm).

VM Running Check

"*Fail if Virtual Machine is Running / Not Running*"

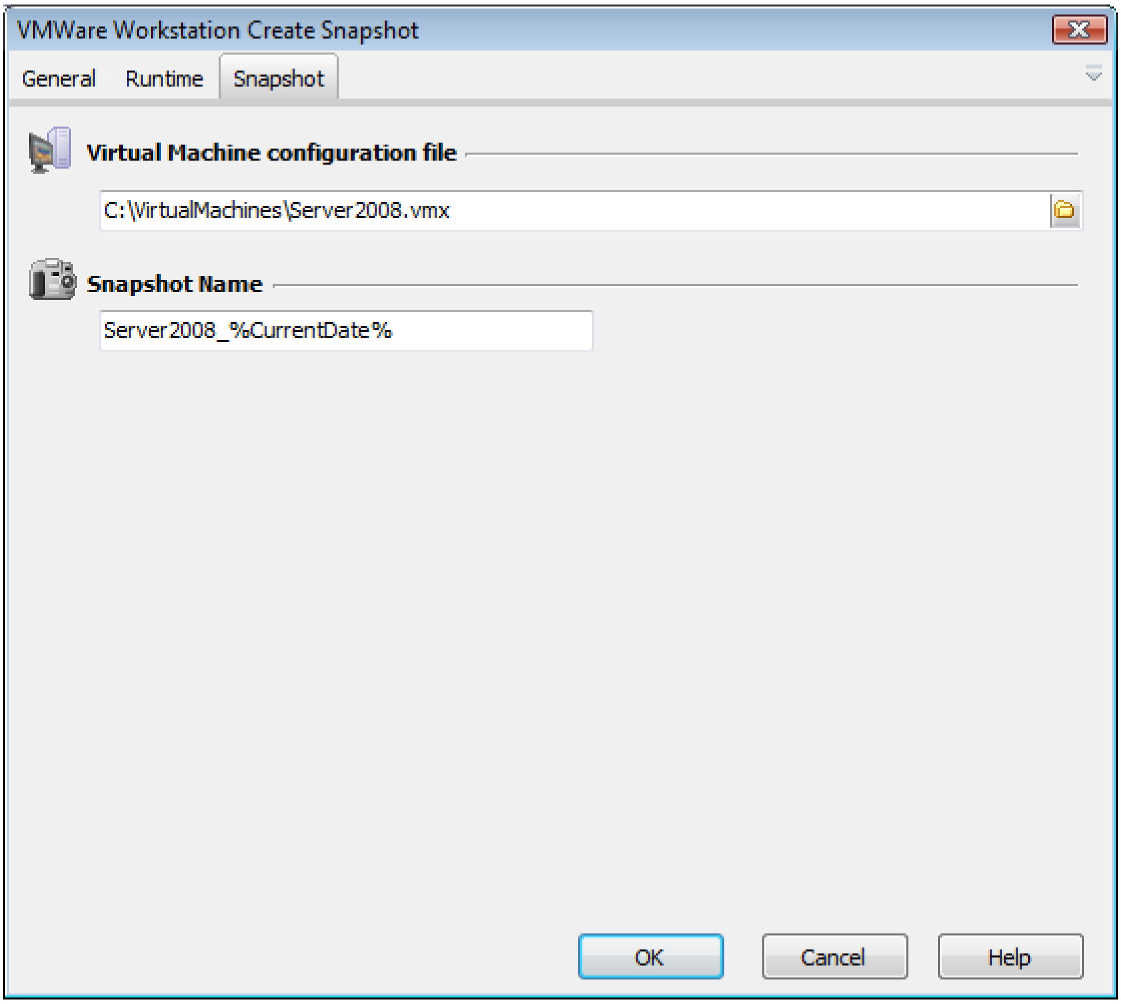
If this box is checked, the action will fail if the VM is either running or not running.

"*Set FB Variable as boolean*"

If this box is checked, the selected FB Variable will be set to "True" if the VM is running or "False" if it is not running.

6.26.4.2 Create Snapshot

The VMWare Create Snapshot action creates a snapshot of a currently running or stopped Virtual Machine.



Virtual Machine configuration file: C:\VirtualMachines\Server2008.vmx

Snapshot Name: Server2008_%CurrentDate%

Virtual Machine configuration file

The configuration file of the VM to snapshot. The VM can be running, suspended or stopped.

Snapshot Name

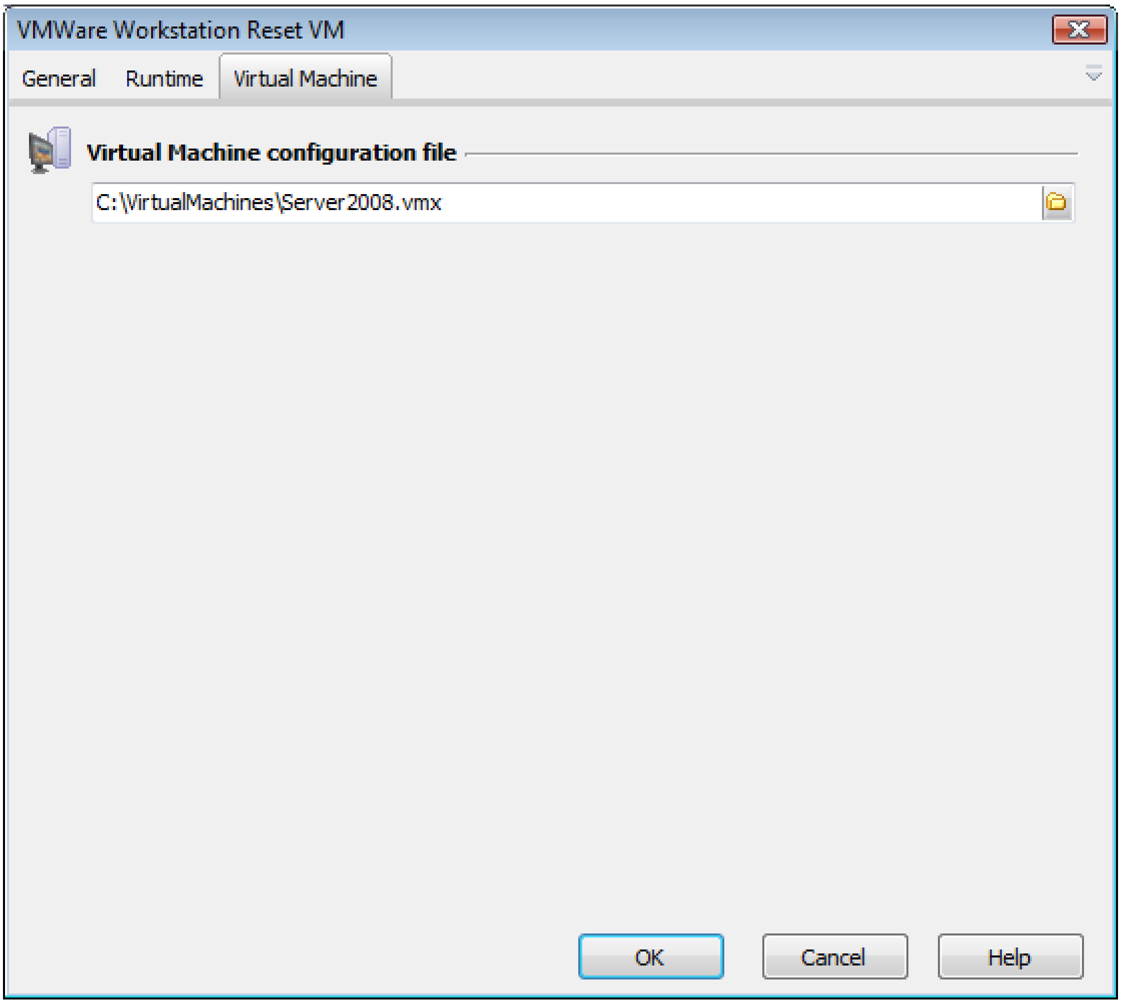
The identifying name for the new snapshot.

NOTE: In recent versions of VMWare, a hint is displayed when creating background snapshots. If the hint dialog is displayed, it will delay an automated snapshot until OK is pressed. If the build is running unattended, it is advisable to check if this hint is displayed and check the "Do Not Display Again" option if it is so.

6.26.4.3 Reset Virtual Machine

The Reset Virtual Machine action allows you to forcibly reset a VMWare Virtual Machine.

Warning: The Reset Virtual Machine action does not take account of the Guest operating system's status. It is possible to lose information through use of this action.



VMWare Workstation Reset VM

Virtual Machine configuration file: C:\VirtualMachines\Server2008.vmx

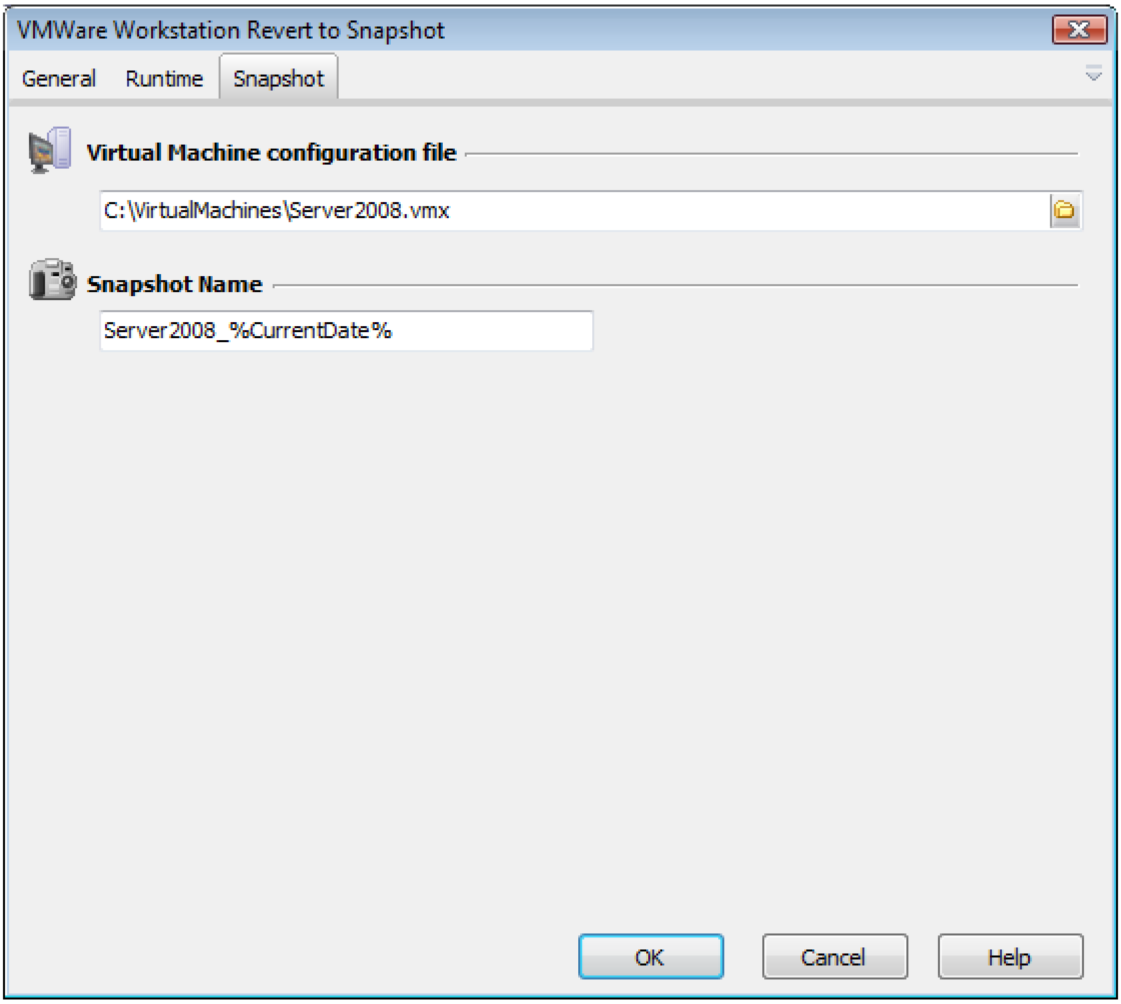
Virtual Machine Configuration File

Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team file (.vmtm).

6.26.4.4 Revert To Snapshot

The Revert To Snapshot action allows you to revert a Virtual Machine to a previously saved state.

Note that reverted virtual machines are normally stopped. To start or resume a reverted virtual machine, use the Start Virtual Machine action.



Virtual Machine configuration file

C:\VirtualMachines\Server2008.vmx

Snapshot Name

Server2008_%CurrentDate%

Virtual Machine configuration file

The configuration file of the VM to revert. The VM can be running, but will need to be started again following the Revert action if you wish to continue using it.

Snapshot Name

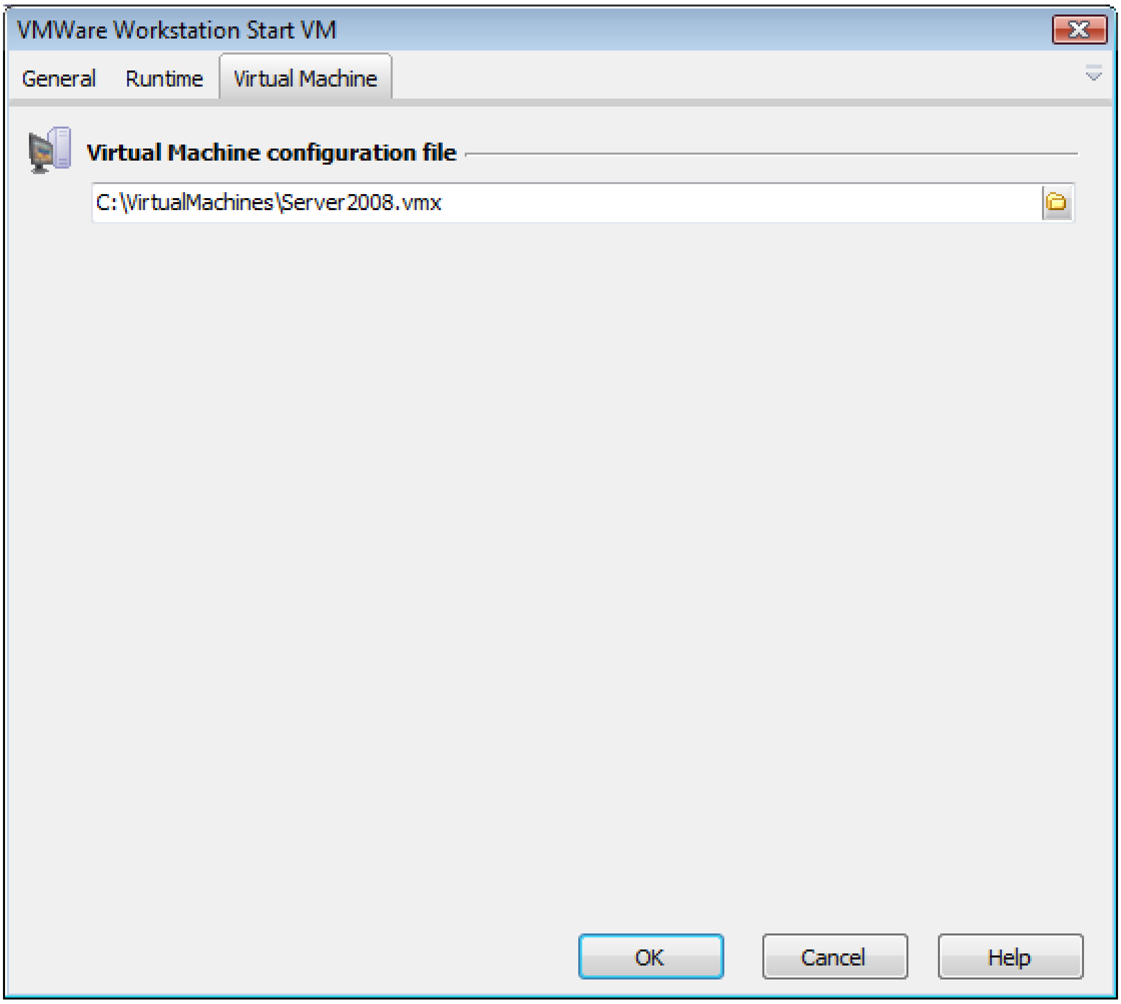
The identifying name for the snapshot to restore to.

NOTE: In recent versions of VMWare, a hint is displayed when restoring snapshots in the background. If the hint dialog is displayed, it will delay the restore until OK is pressed. If the build is running unattended, it is advisable to check if this hint is displayed and check the "Do Not Display Again" option if it is.

6.26.4.5 Start Virtual Machine

The Start Virtual Machine action allows you to start a halted Virtual Machine or resume a suspended Virtual Machine.

Note : Due to a limitation in VMWare Workstation (current in version 5.x), each virtual machine started with the Start Virtual Machine action will start in a new window.



Virtual Machine configuration file

C:\VirtualMachines\Server2008.vmx

Virtual Machine Configuration File

Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team configuration (.vmtm).

Selecting a Virtual Machine which is part of a team (as of VMWare Workstation 5.x) will start the entire team.

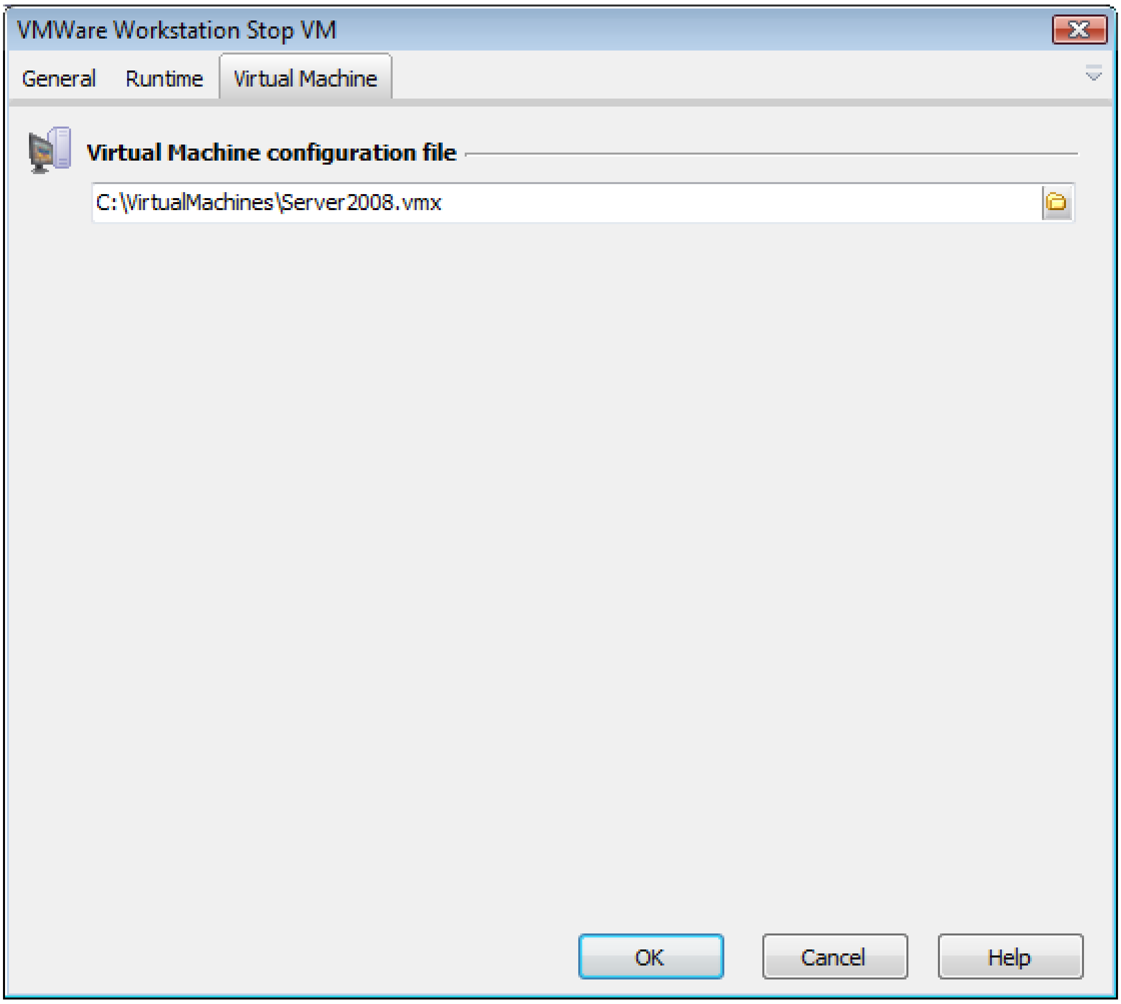
The Virtual Machine must not be already running in a VMWare workstation instance.

6.26.4.6 Stop Virtual Machine

The Stop Virtual Machine action allows you to halt a running VMWare Virtual Machine.

Warning: The Stop Virtual Machine action does not shut down the Guest operating system automatically. It is recommended you use the Suspend Virtual Machine action wherever possible.

Note: Currently (VMWare Workstation 5.5) it is impossible to stop teams, due to a limitation in VMWare. It is also impossible to restart VMs which are part of a team and have already been stopped.



Virtual Machine configuration file

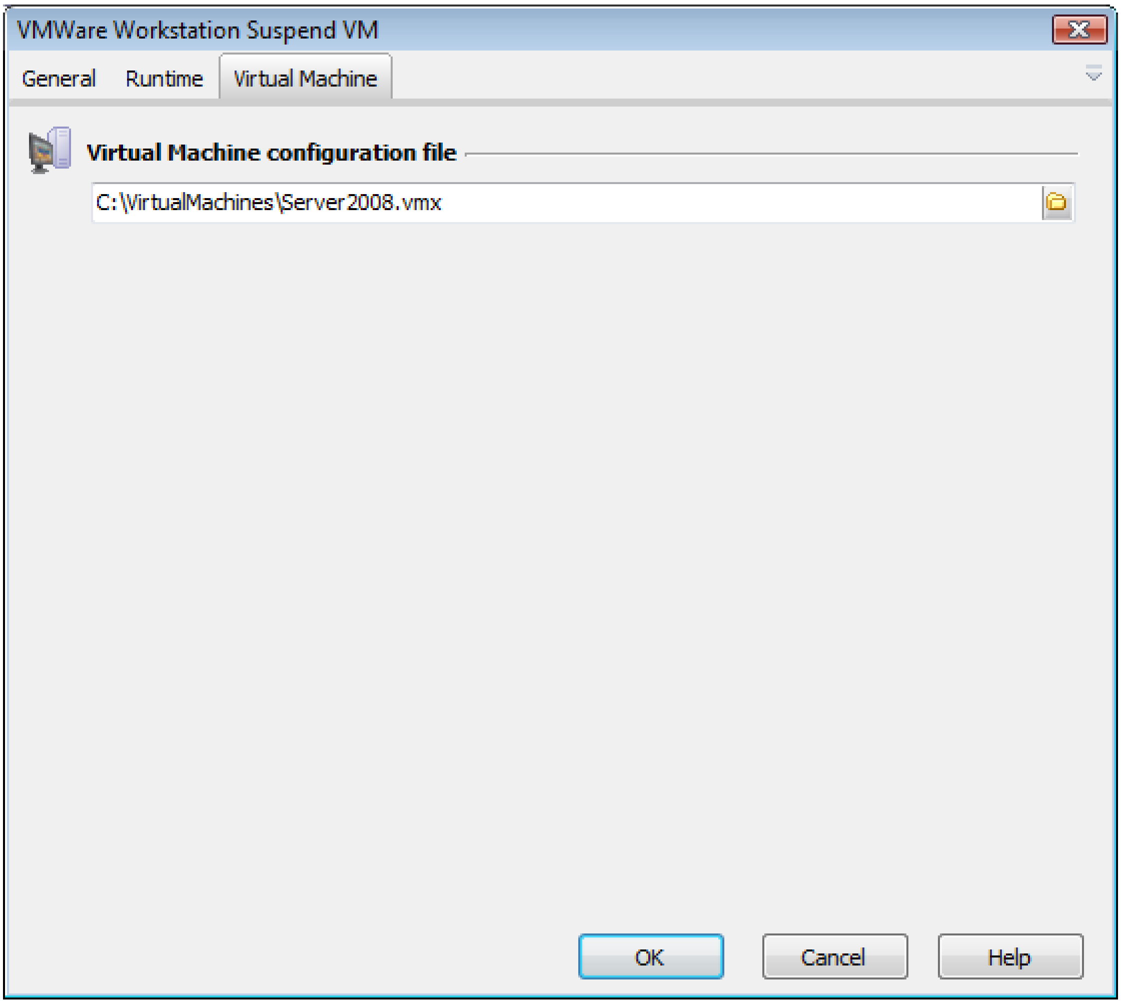
C:\VirtualMachines\Server2008.vmx

Virtual Machine Configuration File

Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team configuration (.vmtm).

6.26.4.7 Suspend Virtual Machine

The Suspend Virtual Machine action allows you to put a VMWare Virtual Machine into suspension. The suspended VM can be started later with the Start Virtual Machine action.



Virtual Machine configuration file

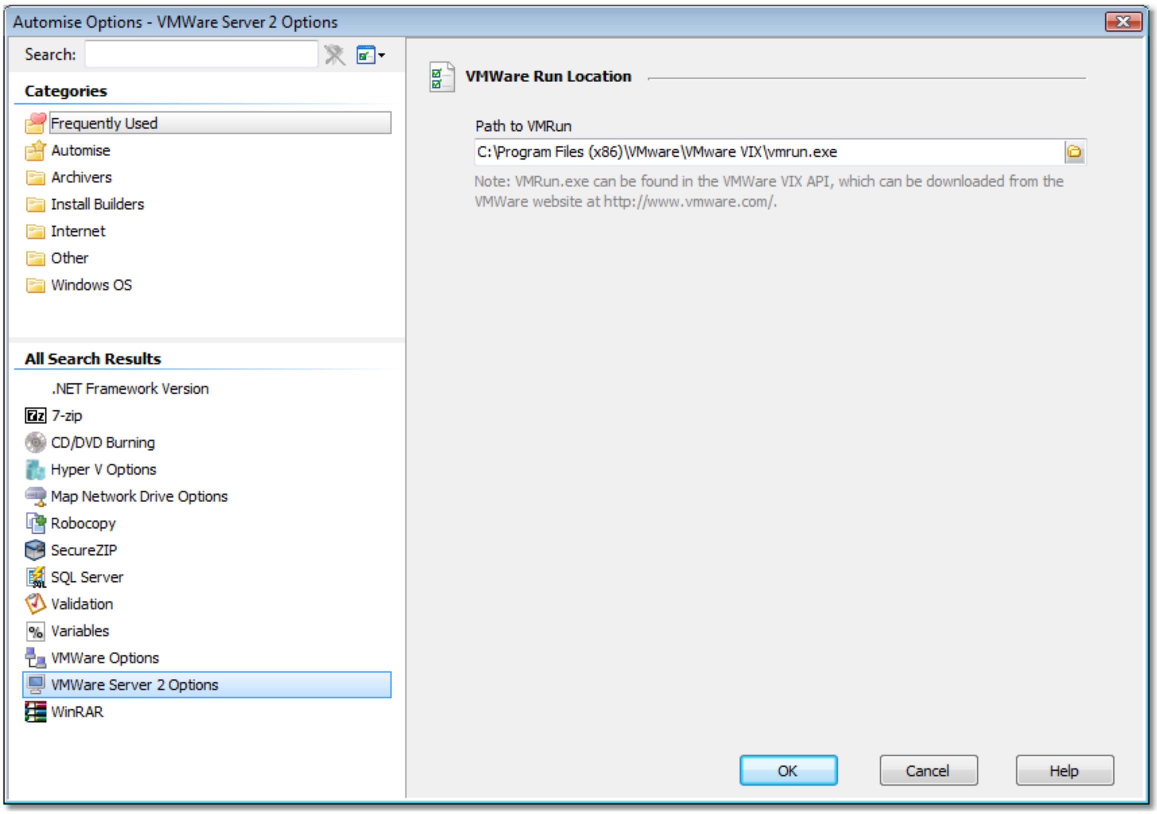
C:\VirtualMachines\Server2008.vmx

Virtual Machine Configuration File

Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team file (.vmtm).

6.26.5 VMWare

The VMWare actions use the VMRun command line utility to control virtual machines either on a VMWare Server installation or VMWare Workstation.

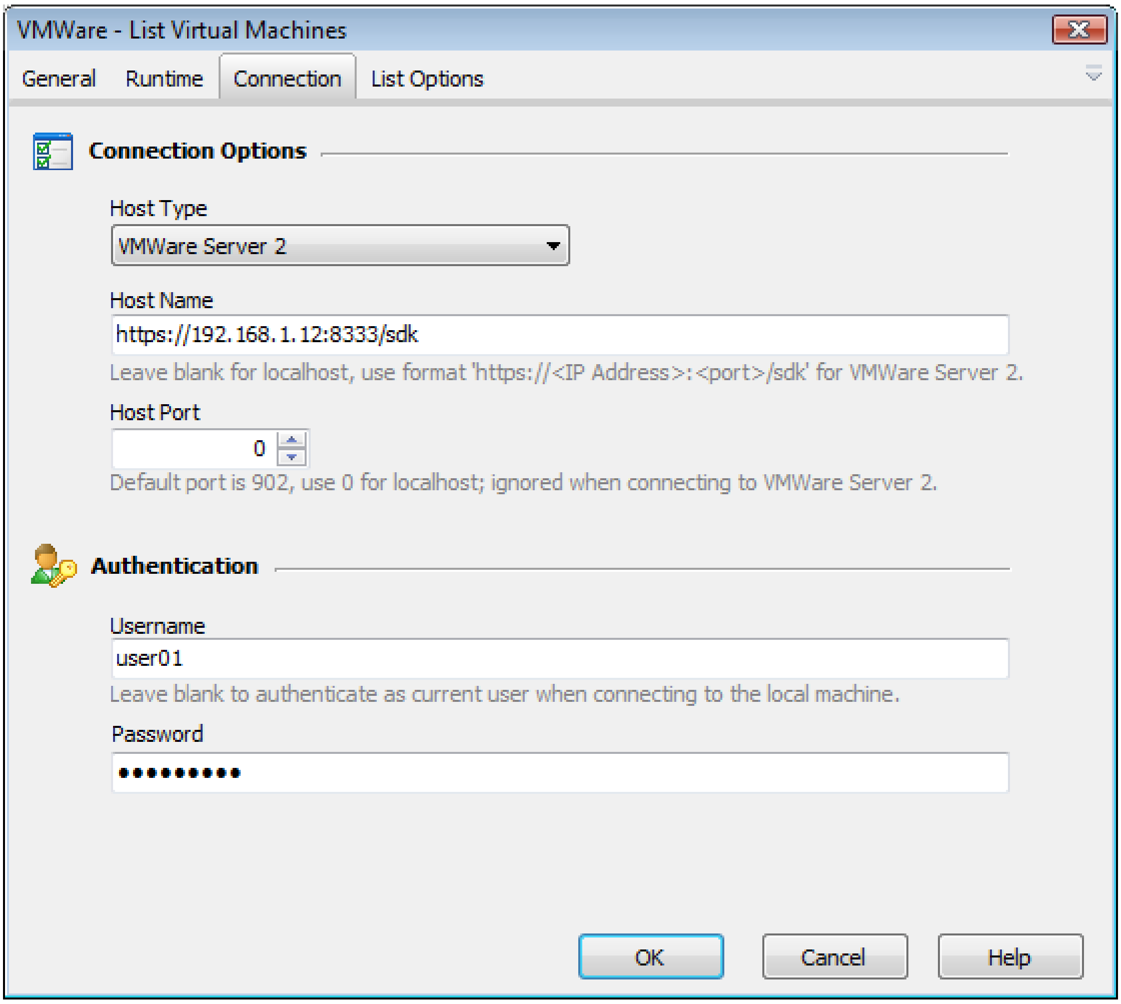


VMWare Run Location

Path to VMRun: C:\Program Files (x86)\VMware\VMware VIX\vmrun.exe

Note: VMRun.exe can be found in the VMWare VIX API, which can be downloaded from the VMWare website at http://www.vmware.com/.

Connection Options



Host Type
VMWare Server 2

Host Name
https://192.168.1.12:8333/sdk
Leave blank for localhost, use format 'https://<IP Address>:<port>/sdk' for VMWare Server 2.

Host Port
0
Default port is 902, use 0 for localhost; ignored when connecting to VMWare Server 2.

Username
user01
Leave blank to authenticate as current user when connecting to the local machine.

Password
●●●●●●●●●

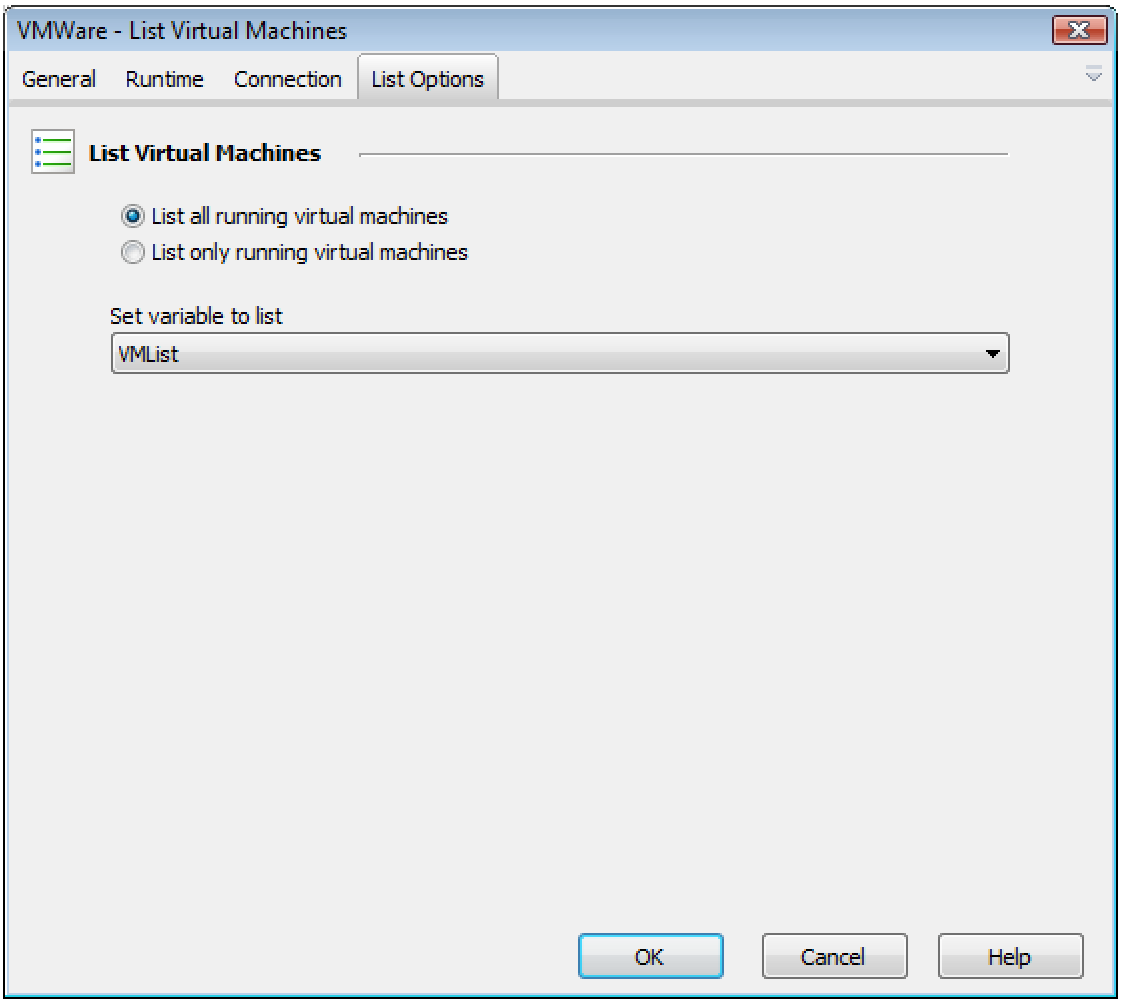
Host Type
The type of host that actions will be connecting to, the possible values include VMWare Server 1, VMWare Server 2 or Workstation.

Host Name
The name of the machine where the VMWare host is located, use *localhost* if you are connecting to an instance on the local machine.

Host Port
The port which the VMWare instance has been configured to use, when connecting to a VMWare Server 2 instance you do not need to specify a port as it's specified within the Host Name.

6.26.5.1 List Virtual Machine

VMWare - List Virtual Machine action enables you to list the virtual machines registered on the host.



VMWare - List Virtual Machines

List Virtual Machines

● List all running virtual machines
○ List only running virtual machines

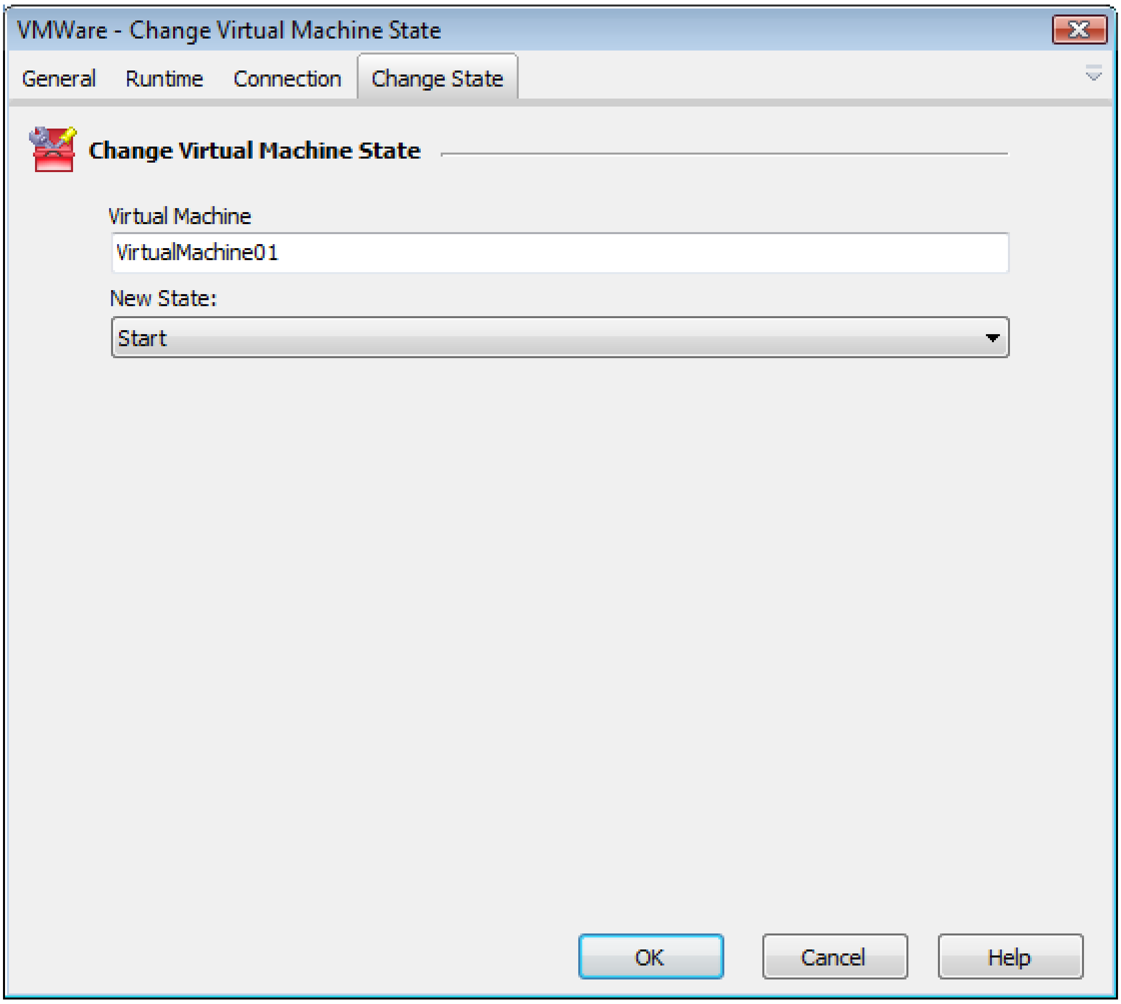
Set variable to list: VMList

Options

You can choose to list all the machines that have been configured on the host, or just the machines that are currently running. A variable can also be specified which the list of virtual machines will be written to and can then be iterated over using the List Iterator action.

6.26.5.2 Change Virtual Machine State

The VMWare - Change Virtual Machine State action enables you to connect to a VMWare instance and change the state of a virtual machine.


VMWare - Change Virtual Machine State
General
Runtime
Connection
Change State
Change Virtual Machine State

Virtual Machine: VirtualMachine01

New State: Start

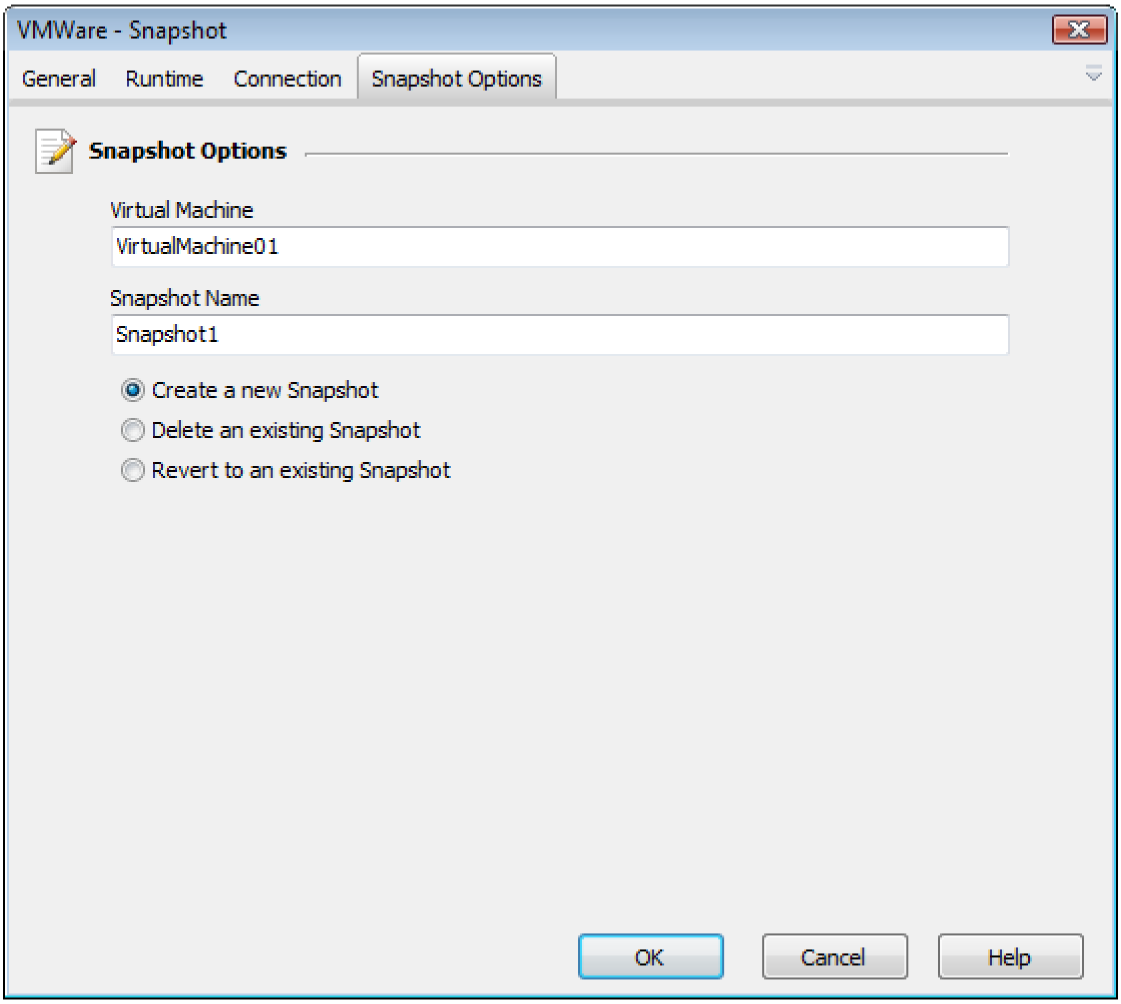
Virtual Machine
The name of the virtual machine which you wish to change the state. A list of the current virtual machines can be obtained using the List Virtual Machines action.

New State
The state you wish the virtual machine to be changed to, the possible values are:

- Start - Starts the virtual machine.
- Stop [Soft] - Stops the virtual machine using the operating system's shutdown procedure.
- Stop [Hard] - Shuts off the virtual machine.
- Reset [Soft] - Resets the virtual machine using the operating system's restart procedure.
- Reset [Hard] - Resets the virtual machine without letting the operating system shutdown.
- Suspend - Suspends the virtual machine.
- Pause - Pauses the virtual machine.
- Unpause - Resumes a previously paused virtual machine.

6.26.5.3 Snapshot

Allows control of the snapshots of a Virtual Machine.



Virtual Machine
VirtualMachine01

Snapshot Name
Snapshot1

● Create a new Snapshot
○ Delete an existing Snapshot
○ Revert to an existing Snapshot

Virtual Machine
The name of the virtual machine which you wish to change the state. A list of the current virtual machines can be obtained using the List Virtual Machines action.

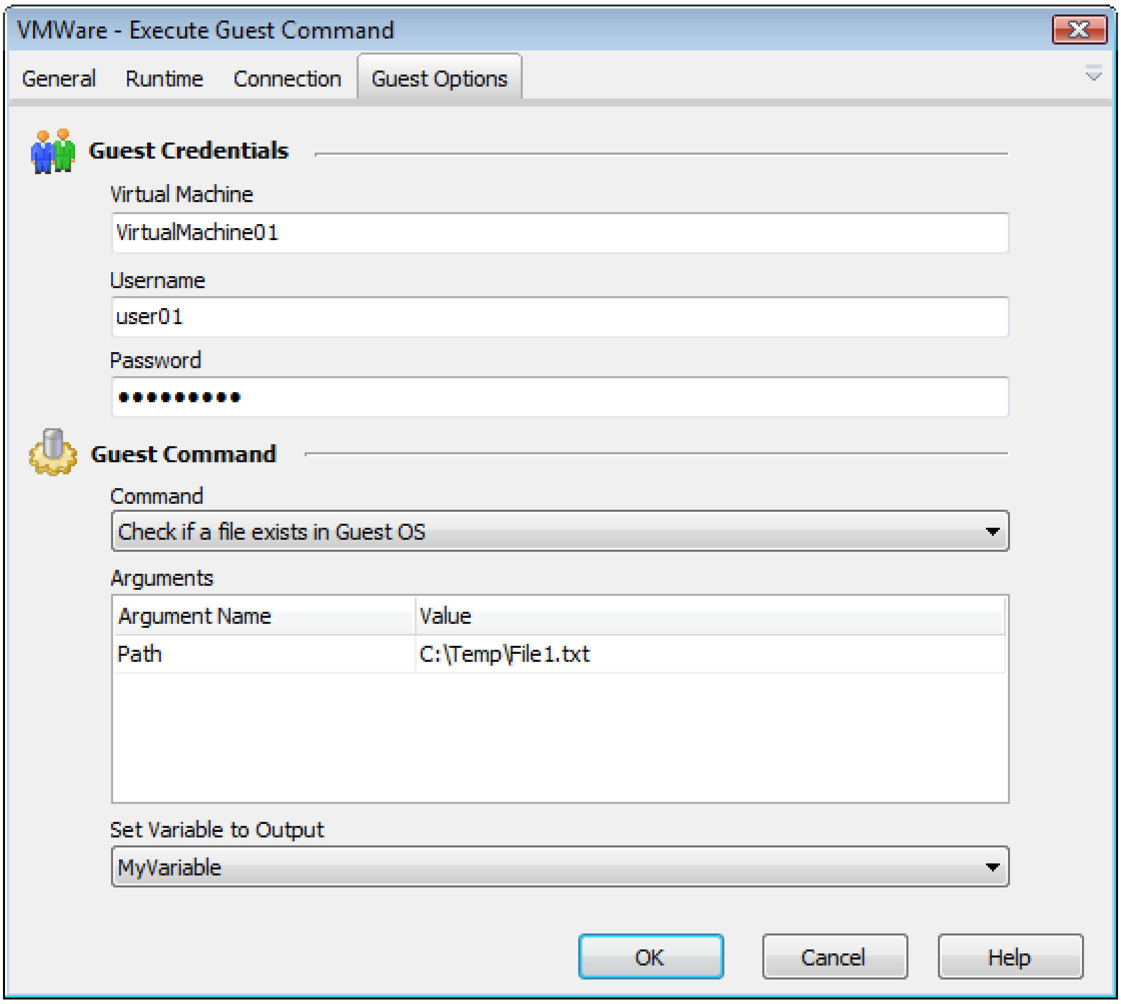
Snapshot Name
The name of the snapshot which you want to manipulate, when connecting to a VMWare Server instance you do not need to provide a snapshot name.

Options

- Create a new Snapshot - Creates a new snapshot of a virtual machine.
- Delete an existing Snapshot - Deletes an existing snapshot of a virtual machine.
- Revert to an existing Snapshot - Reverts the virtual machine back to the snapshot state.

6.26.5.4 Execute Guest Command

Enables a guest command to be executed on the virtual machine using the VMWare Guest Utility.



Virtual Machine: VirtualMachine01

Username: user01

Password: ●●●●●●●●●

Command: Check if a file exists in Guest OS

Arguments

Argument Name	Value
Path	C:\Temp\File1.txt

Set Variable to Output: MyVariable

Virtual Machine
The name of the virtual machine which you wish to change the state. A list of the current virtual machines can be obtained using the List Virtual Machines action.

Username / Password
The credentials to use to connect to the guest operating system.

Guest Command
The command to be executed on the guest operating system, possible commands include:

- Run a program in Guest OS
- Run a program interactively in Guest OS
- Check if a file exists in Guest OS
- List running processes in Guest OS
- Kill a process in Guest OS
- Run a script in Guest OS
- Delete a file in Guest OS
- Rename a file in Guest OS
- Create a directory in Guest OS
- Delete a directory in Guest OS
- List a directory in Guest OS
- Write a variable in the VM state
- Read a variable in the VM state
- Write a variable in the guest environment

- Read a variable in the guest environment
- Save screen capture of virtual machine to host
- Copy a file from the guest operating system to the host
- Copy a file from the host to the guest operating system

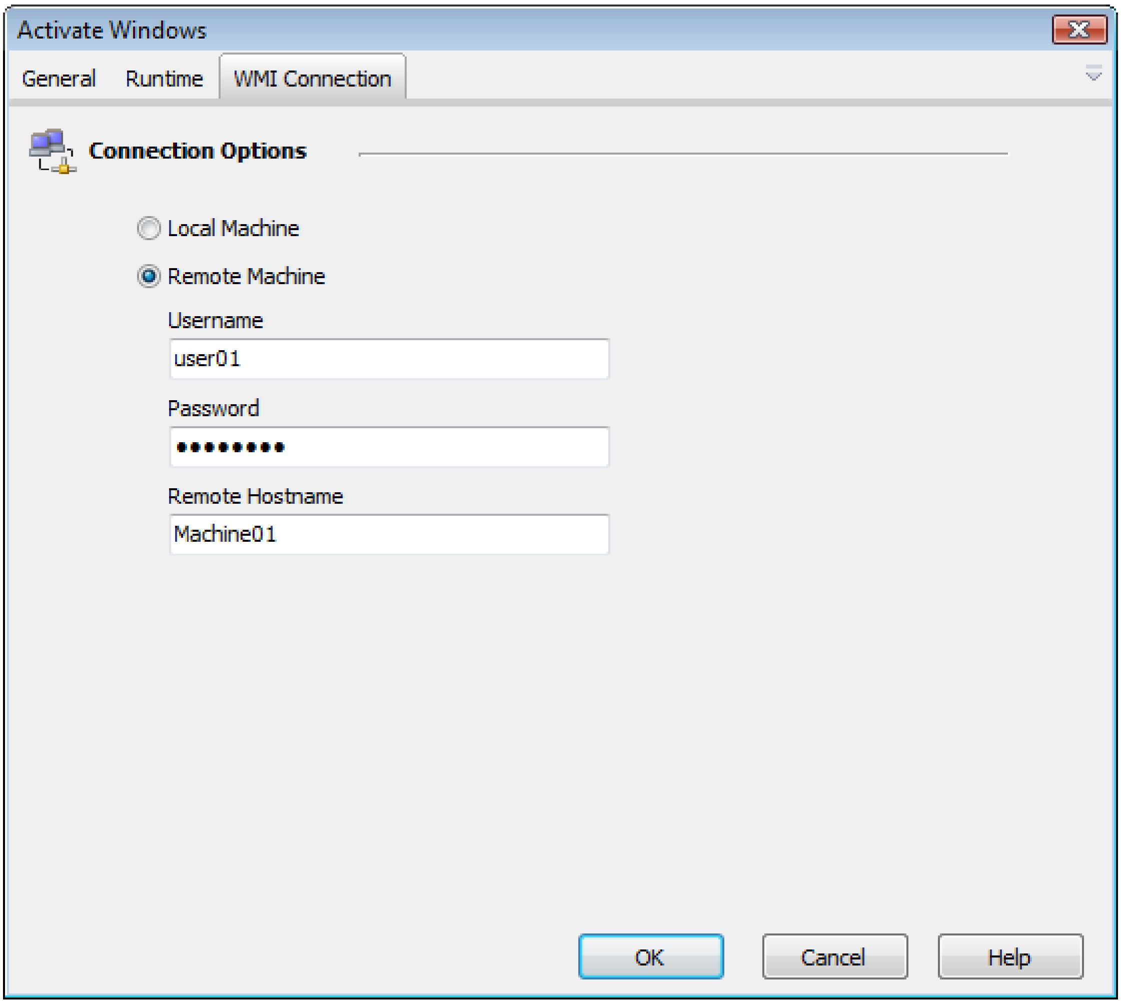
Command Arguments
The arguments that are required by the command.

Variable to Set
The variable to store the output of the guest command.

6.27 Windows OS

6.27.1 Activate Windows

The Activate Windows action enables the activation of Windows either on a local or a remote machine. This is achieved by exchanging license-related data with a Microsoft Clearinghouse license server. If the machine is already activated then no action will be taken.



Connection Options

○ Local Machine

● Remote Machine

Username: user01

Password: ••••••••

Remote Hostname: Machine01

Local Machine
The action will connect to the local machine, with the credentials of the current user.

Remote Machine

The action will connect to a remote machine, using the specified user account.

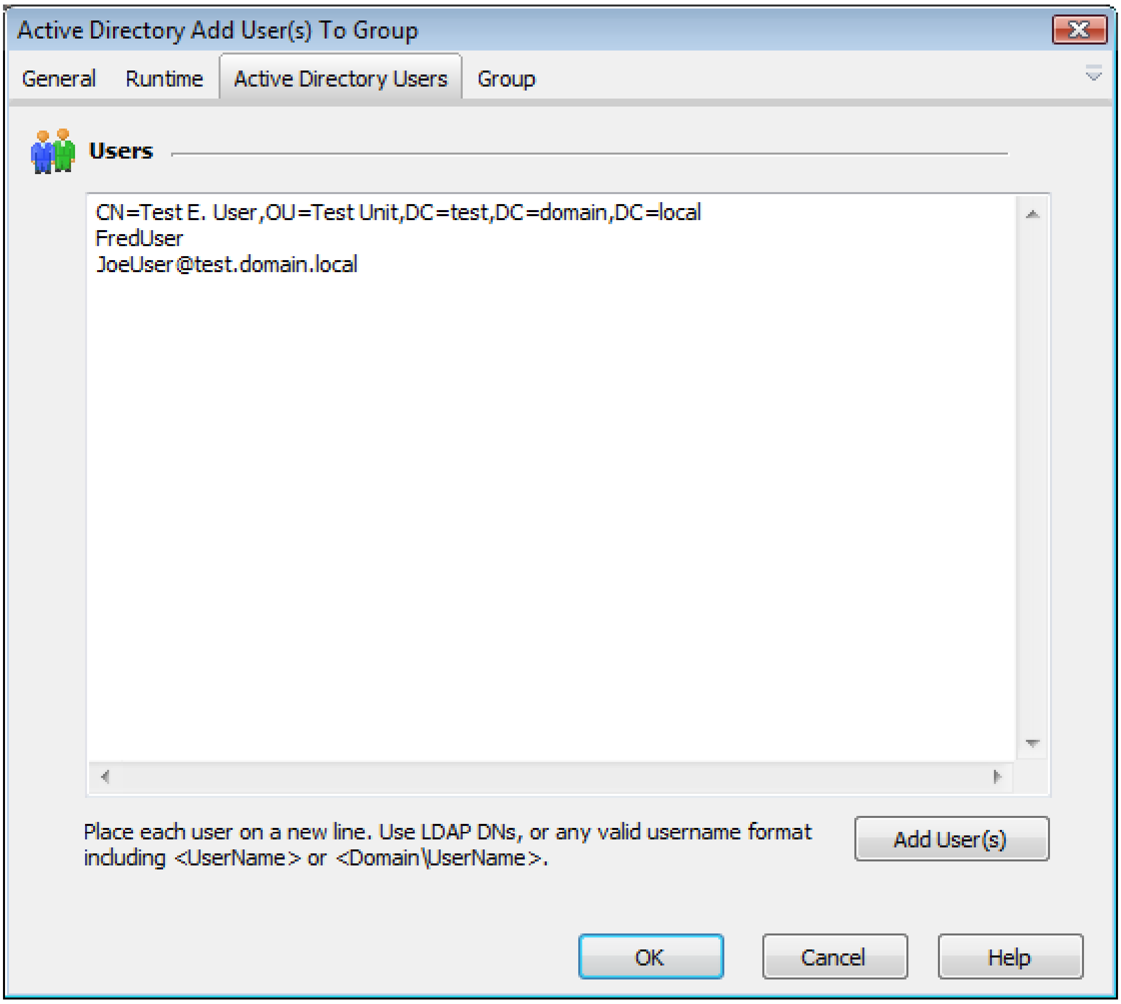
Username
The username can be specified either by just the *username* or *DOMAIN/username*

Remote Host name
The computer which to connect to, if the remote computer is in a different domain then that of the current user account, a Fully Qualified Domain Name must be specified. An IPv4/IPv6 may also be specified.

6.27.2 Active Directory Administration

6.27.2.1 Active Directory Add User(s) To Group

The Active Directory Add User(s) To Group action allows you to add one or more existing users to an Active Directory Group.



```
CN=Test E. User,OU=Test Unit,DC=test,DC=domain,DC=local
FredUser
JoeUser@test.domain.local
```

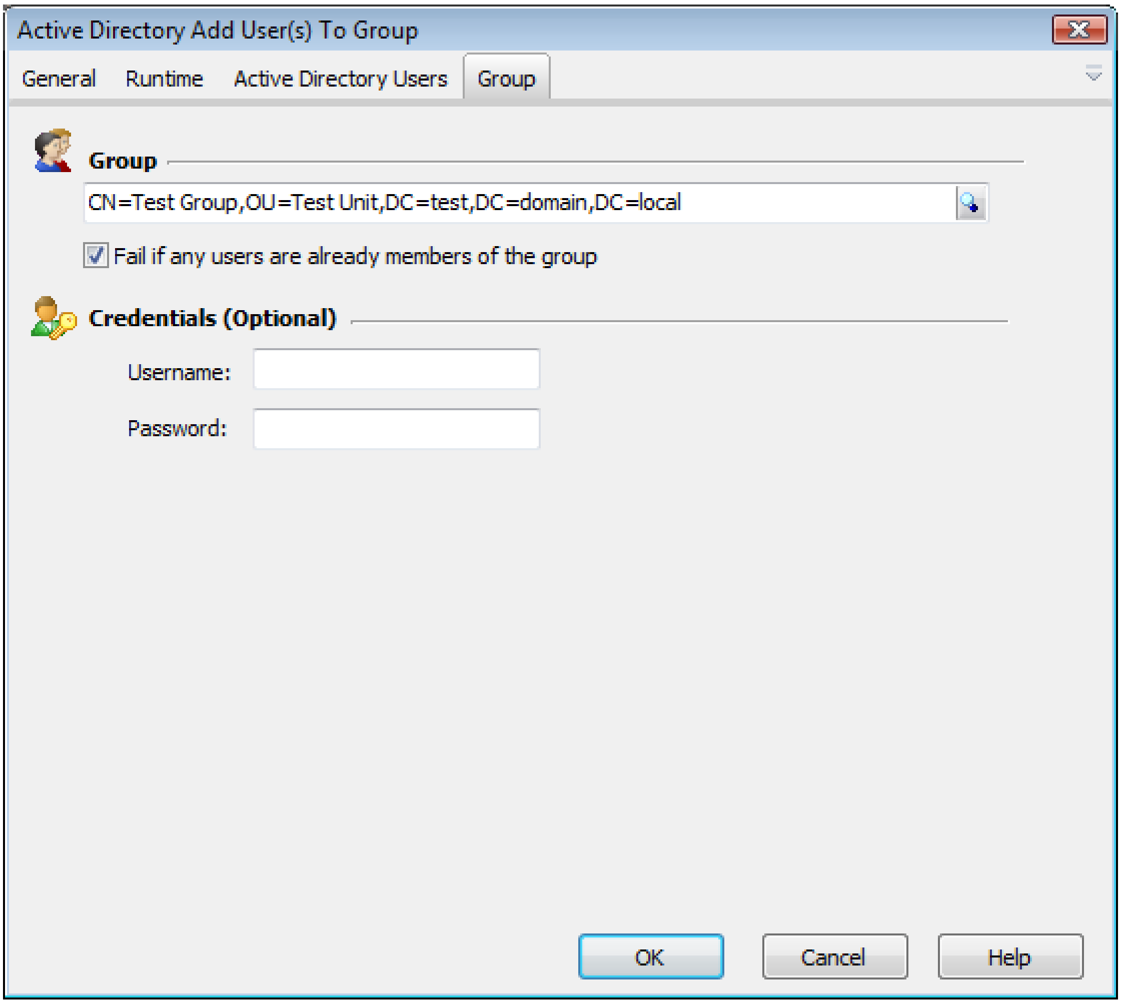
Place each user on a new line. Use LDAP DNs, or any valid username format including <UserName> or <Domain\UserName>.

Users

Users can be specified as LDAP Distinguished Names, or any valid username format - including simple usernames, Domain\UserName syntax, or User@Domain syntax. The LDAP://domainName/ prefix can be used as well.

Click the Add User(s) button to display a dialog allowing you to search for one or more users to add to the list.

Tip: To add all of the users from one group into another, use the Enumerate Group action to write all of the first group members' DNs to a variable, then specify the variable name in the Users field, using the %VarName% syntax.



Group

CN=Test Group,OU=Test Unit,DC=test,DC=domain,DC=local

☑ Fail if any users are already members of the group

Credentials (Optional)

Username:

Password:

Group

Specify the name of the group to add users to. Groups can be specified as a simple group name or an LDAP Distinguished Name.

Fail if any users are already members of the group

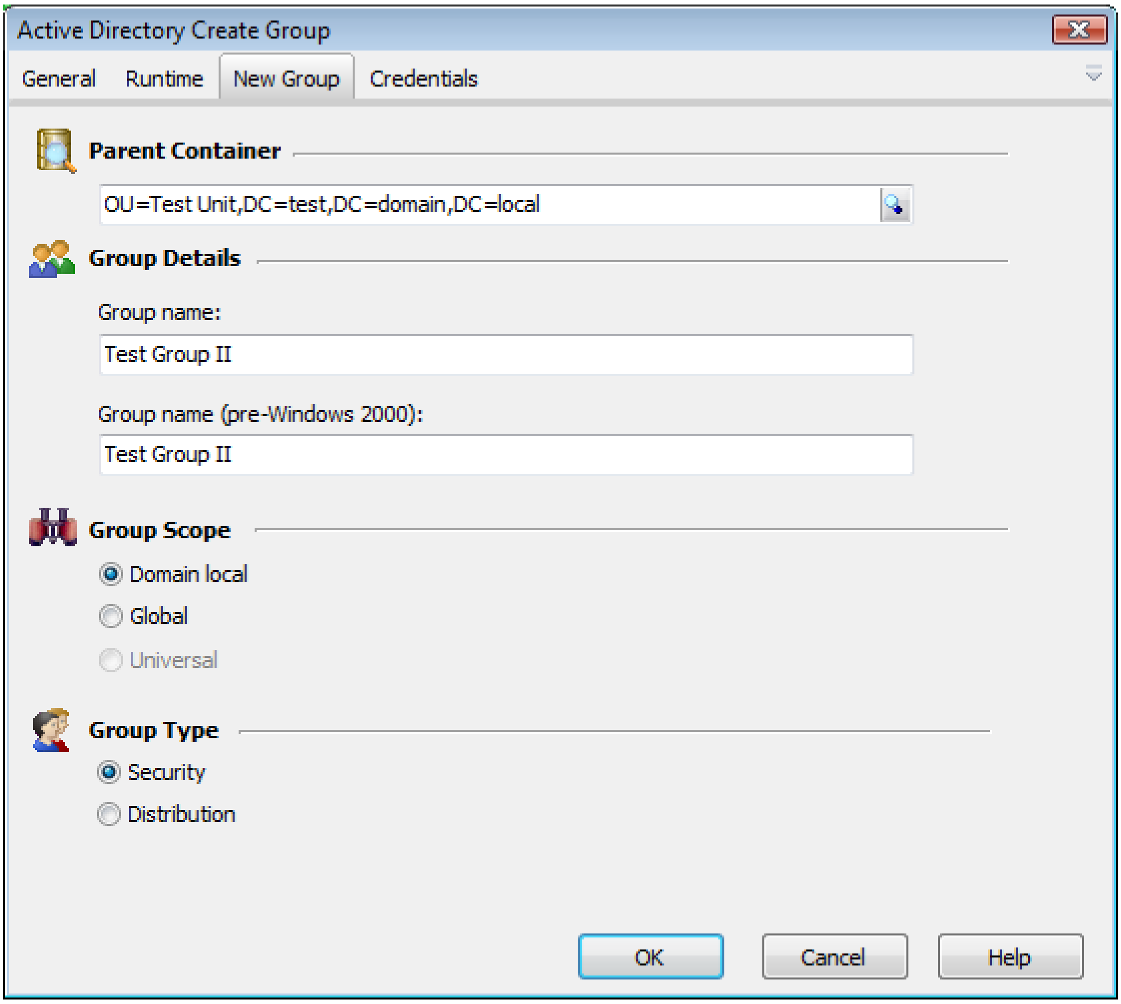
Check this box to fail the action if any users are already members of the group. Note that if the action fails for this reason, some other users may have already been added to the group.

Credentials (Optional)

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.

6.27.2.2 Active Directory Create Group

The Active Directory Create Group action allows you to create a new group.



Parent Container: OU=Test Unit,DC=test,DC=domain,DC=local

Group name: Test Group II

Group name (pre-Windows 2000): Test Group II

Group Scope:
- ● Domain local
- ○ Global
- ○ Universal

Group Type:
- ● Security
- ○ Distribution

Parent Container

Specify the Distinguished Name for the parent container (or OU) in which the new group will be created. Click the magnifying glass to browse for a container.

Group Details

Specify a name (and/or a separate pre-Win2K name) for the new group.

Group Scope

Specify whether you want a domain local group, a global group, or (for distribution groups only) a universal group.

Group Type

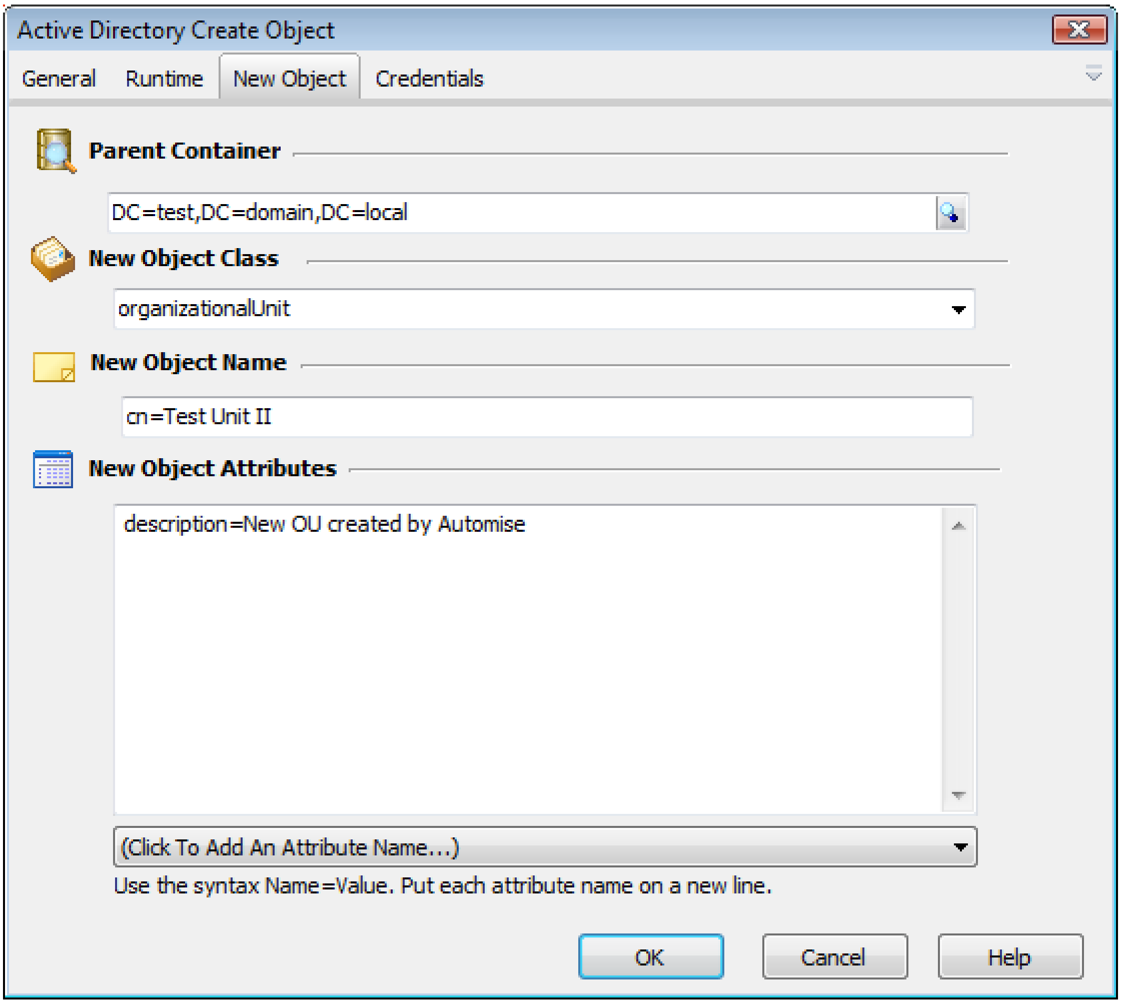
Specify whether you want a Security Group or a Distribution Group.

Credentials

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.

6.27.2.3 Active Directory Create Object

The Active Directory Create Object action allows you to create any Active Directory object, of any class.



Active Directory Create Object

Parent Container: DC=test,DC=domain,DC=local

New Object Class: organizationalUnit

New Object Name: cn=Test Unit II

New Object Attributes:

description=New OU created by Automise

(Click To Add An Attribute Name...)

Use the syntax Name=Value. Put each attribute name on a new line.

Parent Container

Specify the Distinguished Name for the parent container (or OU) in which the new object will be created. Click the magnifying glass to browse for a container.

New Object Class

Specify the class name for the new object. In the example above, we are creating a new Organizational Unit (OU.)

Click on the combo box dropdown to see an automatically generated list of available class names (based on the current Active Directory Schema.)

New Object Name

Specify the name for the new object. For most objects, this will be a common name in the form CN=<Name>. In the above case, Organizational Units are specified OU=.

If no prefix is given, the action will automatically append CN=. The new name must be unique in the parent container.

New Object Attributes

Specify an object attributes you wish to set on the new action. Note that only simple ADS types (strings, booleans, integers) can be set using this dialog (to perform more advanced operations, see below.)

Click on the "(Click To Add...)" combo box dropdown to see an automatically generated list of valid Attribute names for the selected class, based on the Active Directory Schema. Note that not all of the attributes can be set by the action.

Script Events

To enable more advanced operations, this action provides two scripting events - CreatingNewObject and CreatedNewObject. These can be accessed via the Automise Script Editor tab.

The first scripting event is executed before the new object is committed to the Active Directory repository. The second event is executed afterwards. Attributes which are specified by Active Directory itself (ie "CN") are not available to read during the first event.

If an invalid value is set during the first scripting event (CreatingNewObject), the new object will not be created and the action will fail. If an invalid value is set during the second event (CreatedNewObject), the action will still fail but the object will be created (without the changes.)

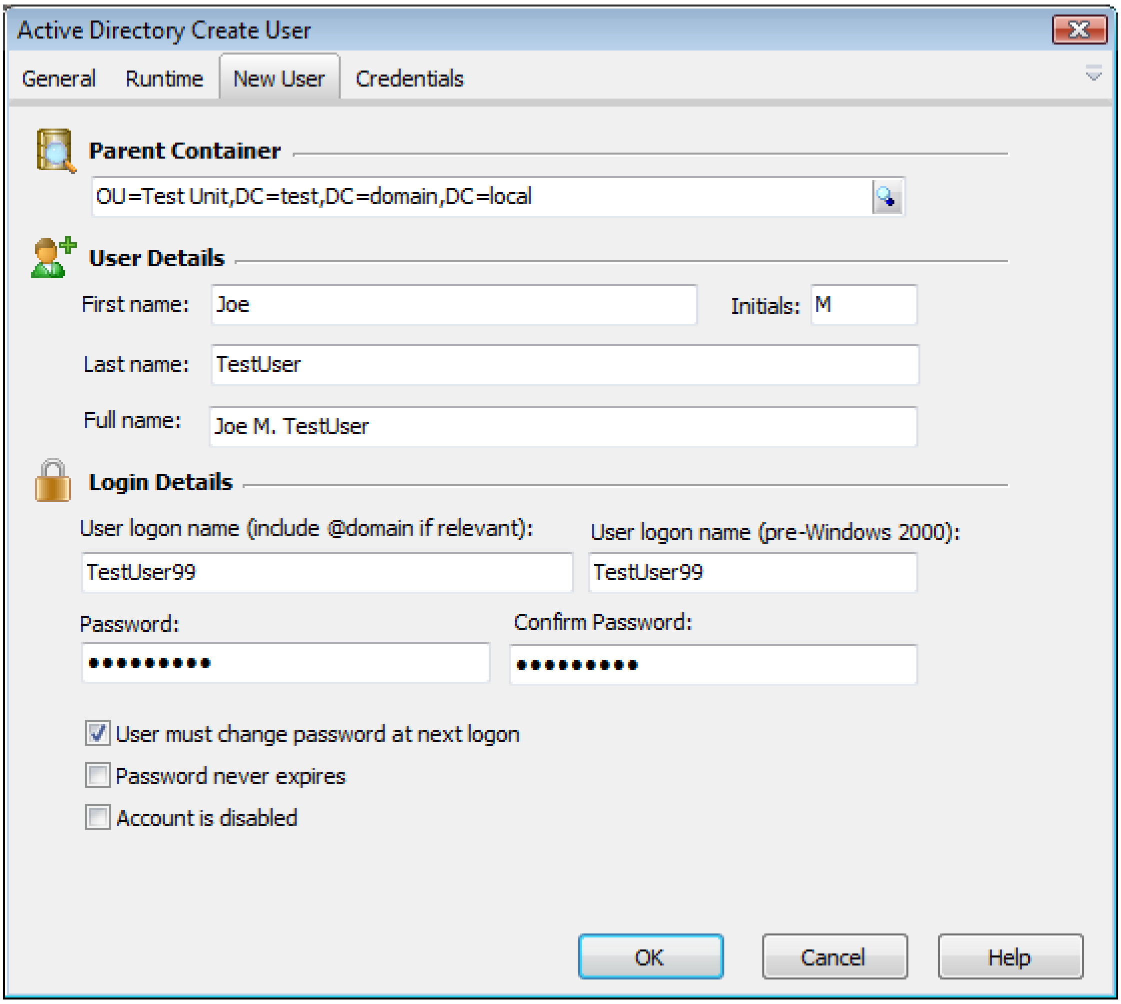
In either event, the Object parameter can be used to access the new Active Directory object. You can use any method available on the IADs interface, and any other interface which is defined for the object class in particular.

Specifically, you may wish to use Object.Put("Name", Value) to set parameters which may not be settable via the **New Object Attributes** field.

6.27.2.4 Active Directory Create User

The Active Directory Create User action allows you to create a new user account.

Tip: To set more properties on the new user account, use the Edit User Properties action.



Active Directory Create User

General | Runtime | New User | Credentials

Parent Container

OU=Test Unit,DC=test,DC=domain,DC=local

User Details

First name: Joe

Initials: M

Last name: TestUser

Full name: Joe M. TestUser

Login Details

User logon name (include @domain if relevant): TestUser99

User logon name (pre-Windows 2000): TestUser99

Password: ●●●●●●●●●

Confirm Password: ●●●●●●●●●

☑ User must change password at next logon

□ Password never expires

□ Account is disabled

Parent Container

Specify the Distinguished Name for the parent container (or OU) in which the new user will be created. Click the magnifying glass to browse for a container.

User Details

Specify the name of the user. By default, full name will fill in automatically as you type first name, last name, and initials.

Login Details

Specify the new user's login name and password (both passwords must match.)

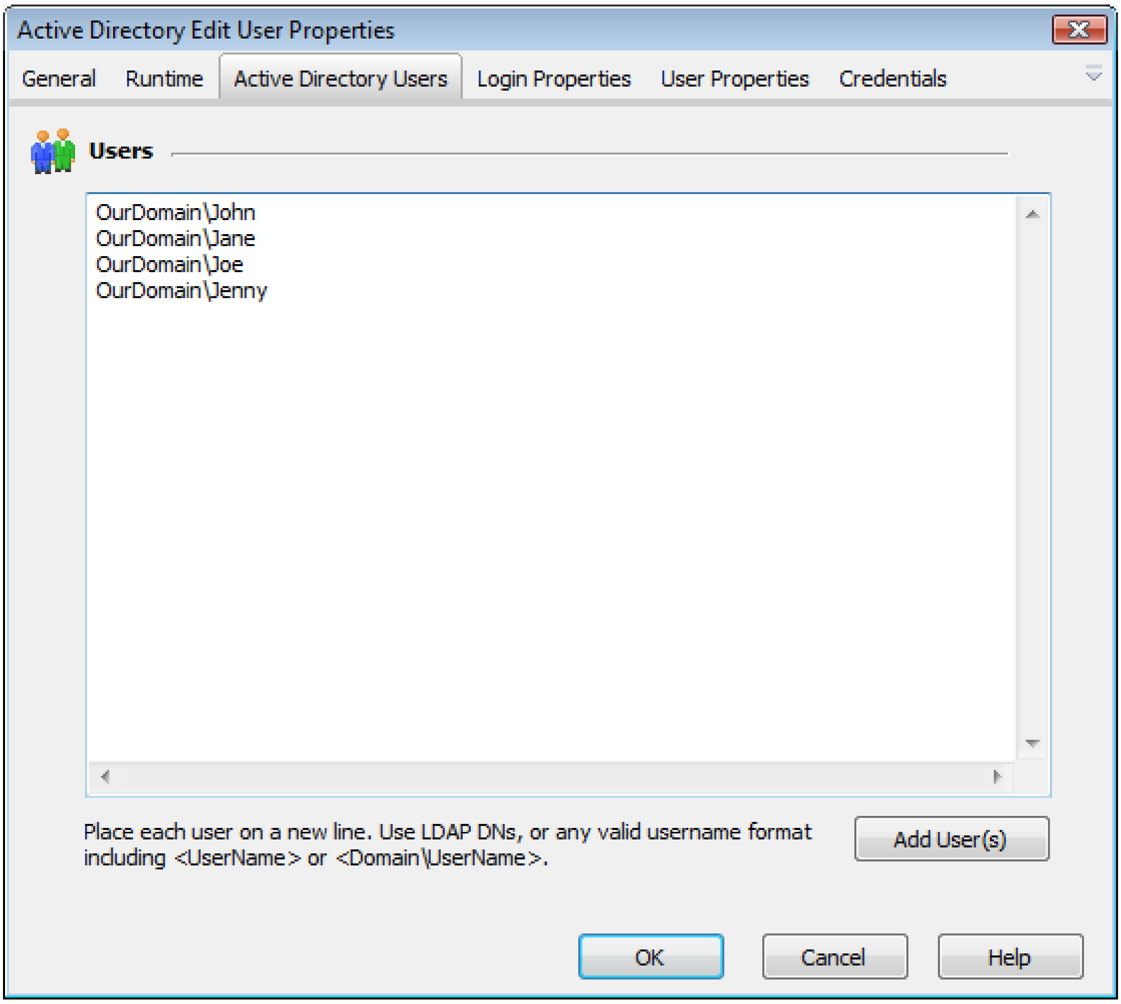
You can also specify whether the user must change their password, whether the password will expire, and whether to create the account enabled or disabled.

Credentials

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.

6.27.2.5 Active Directory Edit User Properties

The Active Directory Edit User Properties action allows you to edit some properties of an existing user.



OurDomain\John
OurDomain\Jane
OurDomain\Joe
OurDomain\Jenny

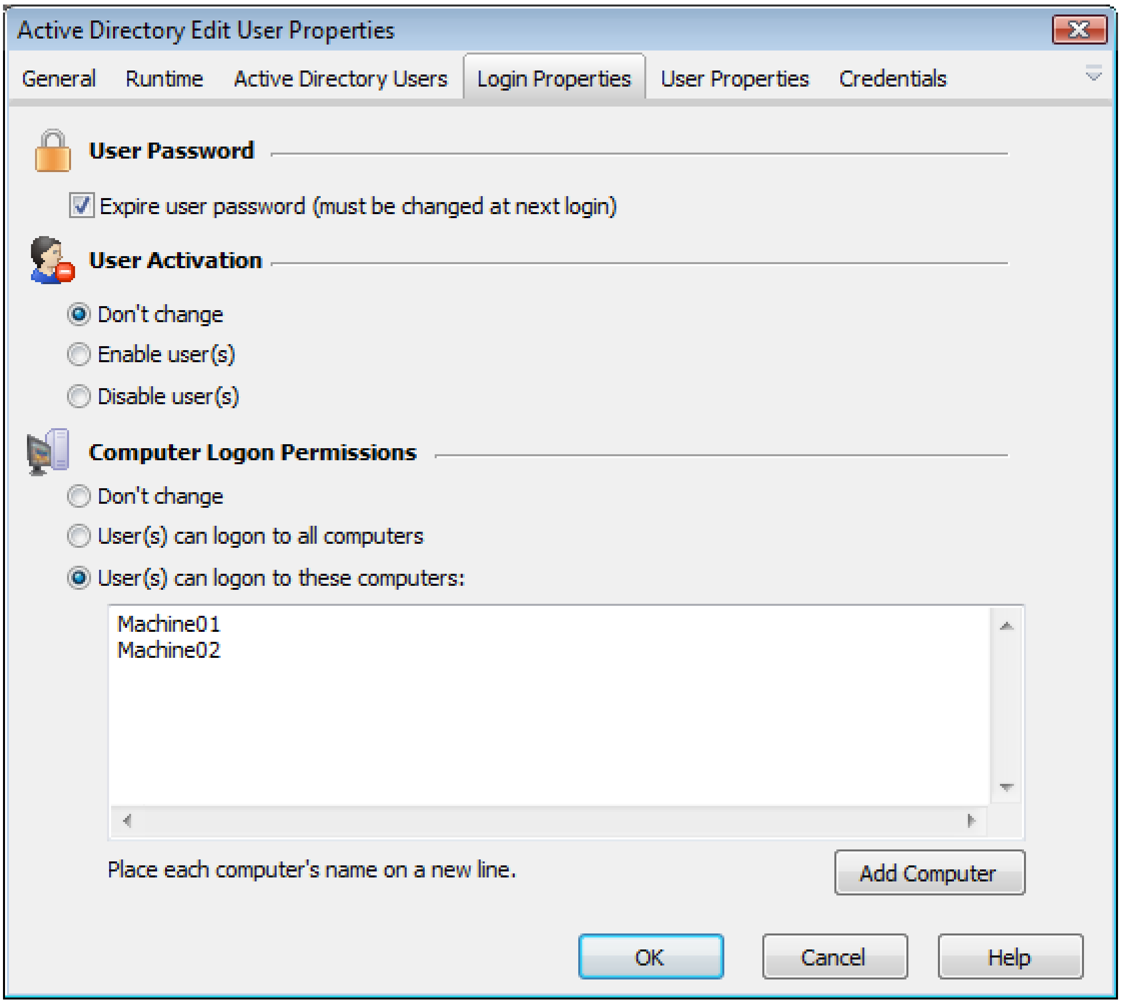
Place each user on a new line. Use LDAP DNs, or any valid username format including <UserName> or <Domain\UserName>.

Active Directory Users

Specify the user(s) that you wish to edit. Users can be specified as LDAP Distinguished Names, or any valid username format - including simple usernames, Domain\UserName syntax, or User@Domain syntax. The LDAP://domainName/ prefix can be used as well.

Click the Add User(s) button to display a dialog allowing you to search for one or more users to add to the list.

Tip: To edit all of the users in a group, use the Enumerate Group action to write all of the group members' DNs to a variable, then specify the variable name in the Users field, using the %VarName% syntax.



User Password

☑ Expire user password (must be changed at next login)

User Activation

● Don't change

○ Enable user(s)

○ Disable user(s)

Computer Logon Permissions

○ Don't change

○ User(s) can logon to all computers

● User(s) can logon to these computers:

```
Machine01
Machine02
```

Place each computer's name on a new line.

User Password

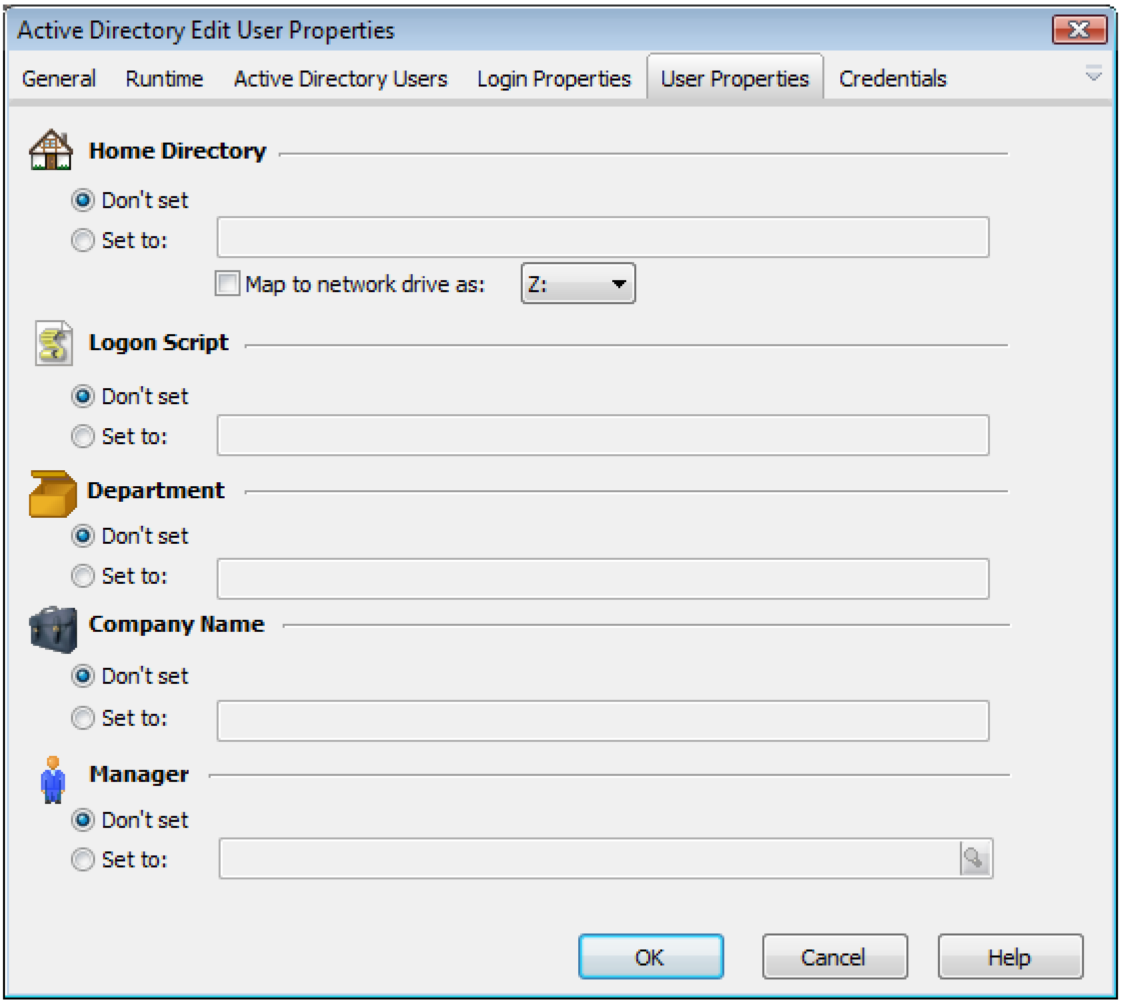
Check this box to force the user(s) to change their password as next login.

User Activation

You can elect to enable or disable users' accounts.

Computer Logon Permissions

This setting corresponds to the "Log On To..." dialog under the Account tab of the Windows "Active Directory Users & Computers" console. You can specify a list of allowed computers, allow all computers, or leave this setting unchanged.



Home Directory

● Don't set

○ Set to:

□ Map to network drive as: Z:

Logon Script

● Don't set

○ Set to:

Department

● Don't set

○ Set to:

Company Name

● Don't set

○ Set to:

Manager

● Don't set

○ Set to:

Home Directory

You can elect to set the home directory of the user(s). You can also elect to set a network drive mapping for the directory name.

Logon Script

You can set a logon script for the user.

Department / Company Name / Manager

To can elect to set the users' department, company and manager names. Note that the manager must also be an Active Directory user (you can specify their name as a username or a Distinguished Name.)

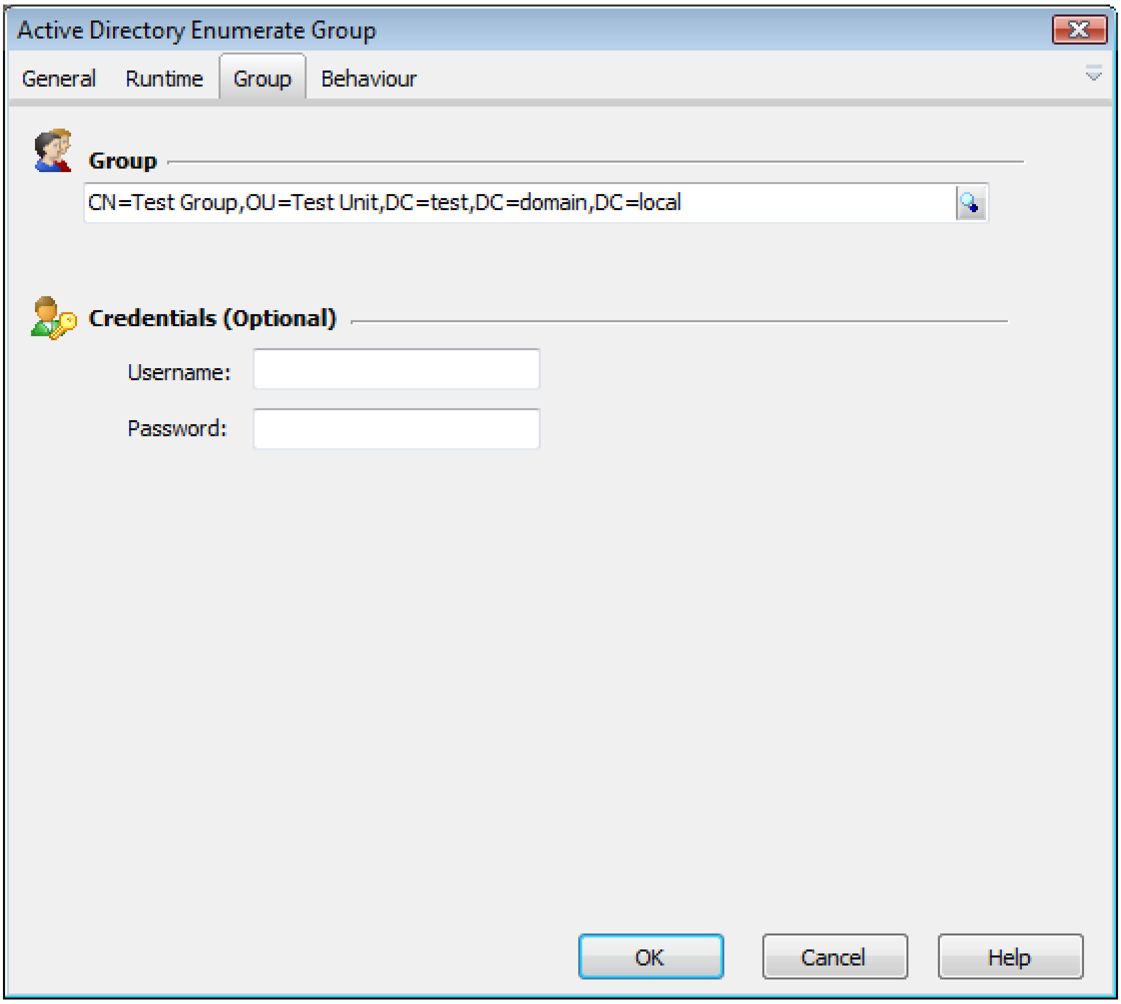
Credentials

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.

6.27.2.6 Active Directory Enumerate Group

The Active Directory Enumerate Group action allows you to list all of the members of a group. The list will be written to the log.

Tip: To list all members of an Organizational Unit (OU) or other container, use the Active Directory Get Info Action.



Group

CN=Test Group,OU=Test Unit,DC=test,DC=domain,DC=local

Credentials (Optional)

Username:

Password:

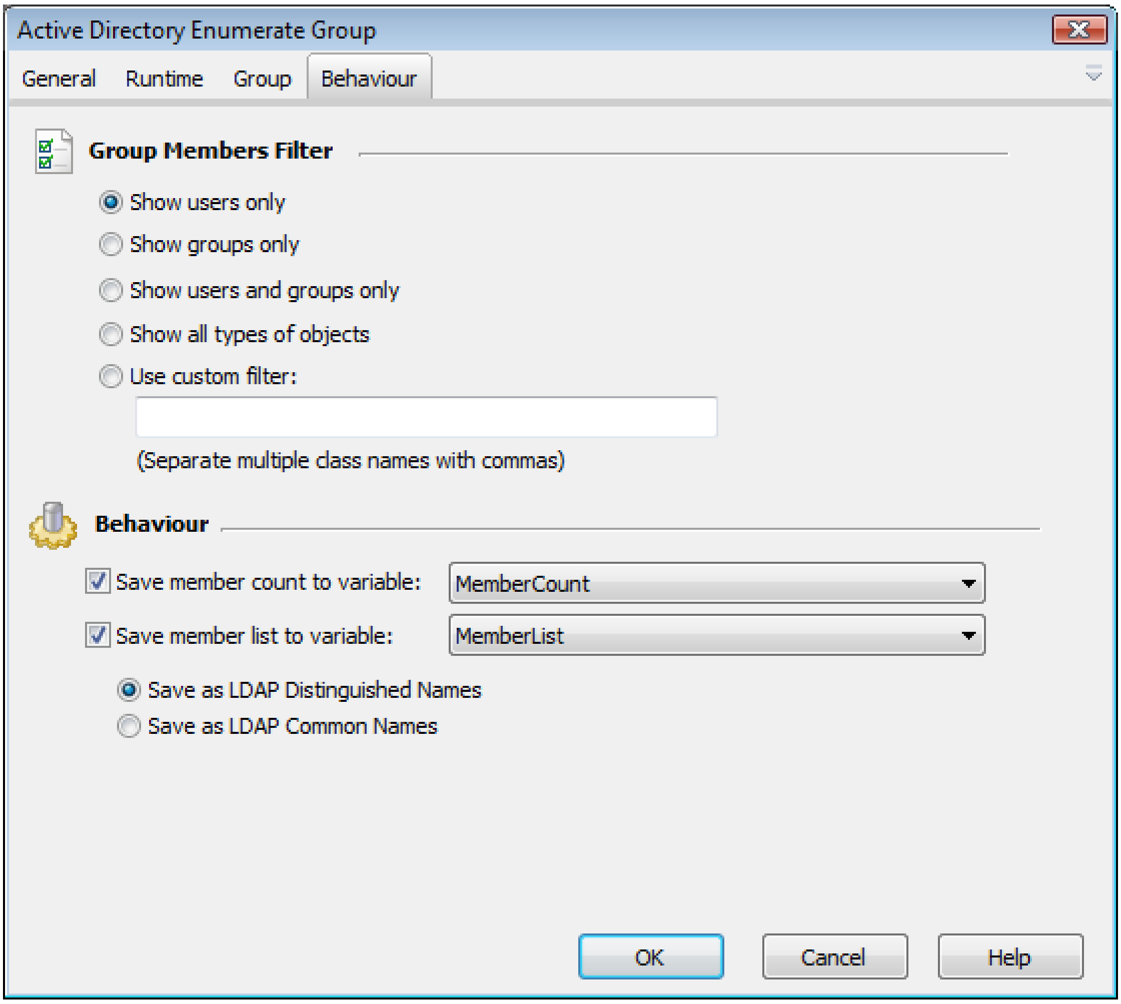
Group

The group to enumerate can be specified as a simple group name or an LDAP Distinguished Name.

Click on the magnifying glass button to display a dialog that you can use to browse for a group.

Credentials (Optional)

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.



Active Directory Enumerate Group

Group Members Filter

● Show users only
○ Show groups only
○ Show users and groups only
○ Show all types of objects
○ Use custom filter:

(Separate multiple class names with commas)

Behaviour

☑ Save member count to variable: MemberCount
☑ Save member list to variable: MemberList

● Save as LDAP Distinguished Names
○ Save as LDAP Common Names

Group Members Filter

You can specify which group members to enumerate. You can choose to show users and/or groups, all types of member objects, or objects matching a specific set of class names.

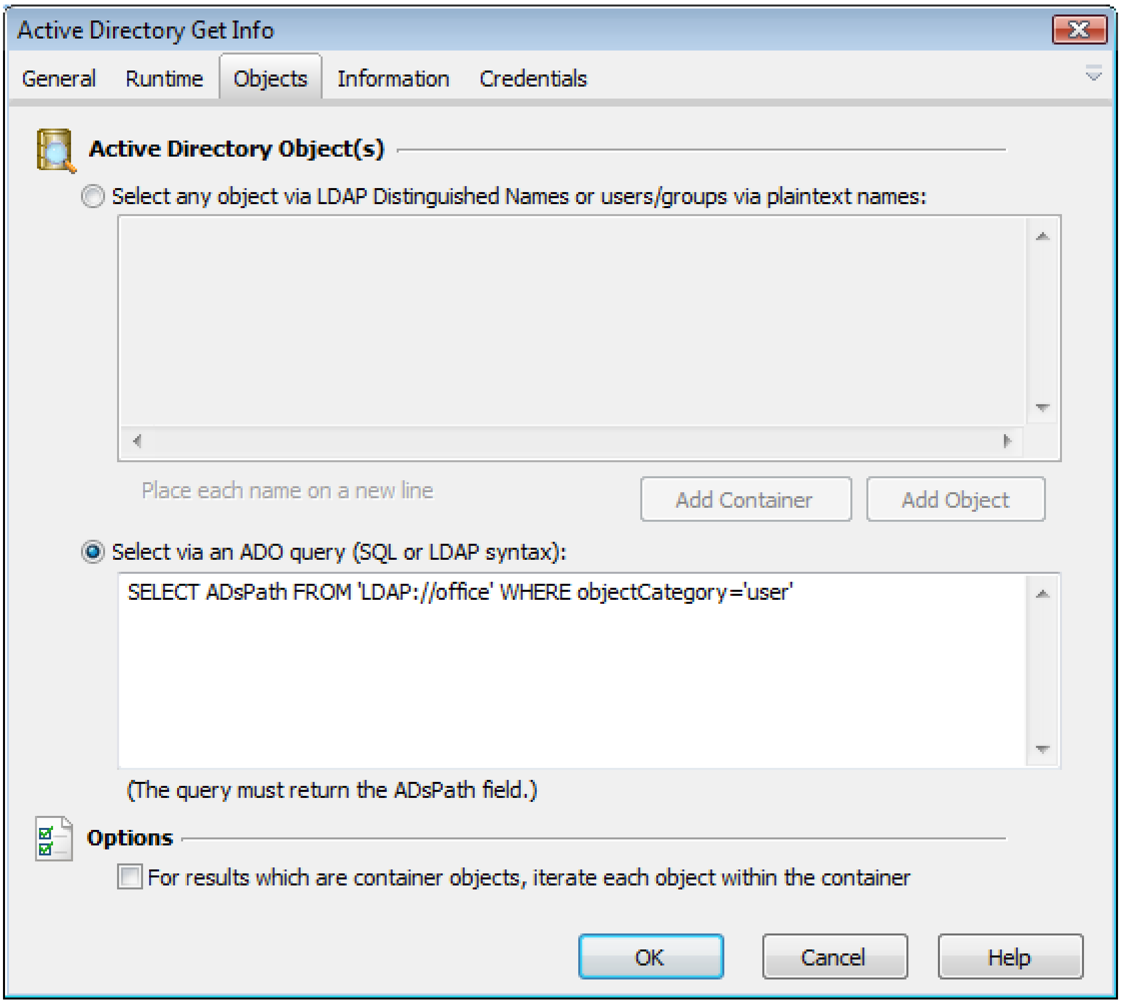
Save member count / list to variable

You can choose to write either the number of matching group members, or the list of group members, to an Automise variable.

If writing a list of members, you can choose to write either Distinguished Names or Common Names for each member. The list will be written as a string, with one member per line.

6.27.2.7 Active Directory Get Info

The Active Directory Get Info action allows you to view properties of Active Directory objects (users, groups, organizational units, computers, printers, etc.) You can find the objects by giving simple names, by providing LDAP Distinguished Name paths, or by executing an ADO query.



○ Select any object via LDAP Distinguished Names or users/groups via plaintext names:

● Select via an ADO query (SQL or LDAP syntax):

```
SELECT ADsPath FROM 'LDAP://office' WHERE objectCategory='user'
```

(The query must return the ADsPath field.)

□ For results which are container objects, iterate each object within the container

"Select any object via LDAP Distinguished Names or users/groups via plaintext names"

If this option is selected, you can specify one or more names of objects to get info. Objects can be fully qualified Distinguished Names (with or without the LDAP:// prefix), or can be a simple user or group name. See the screenshot for some examples.

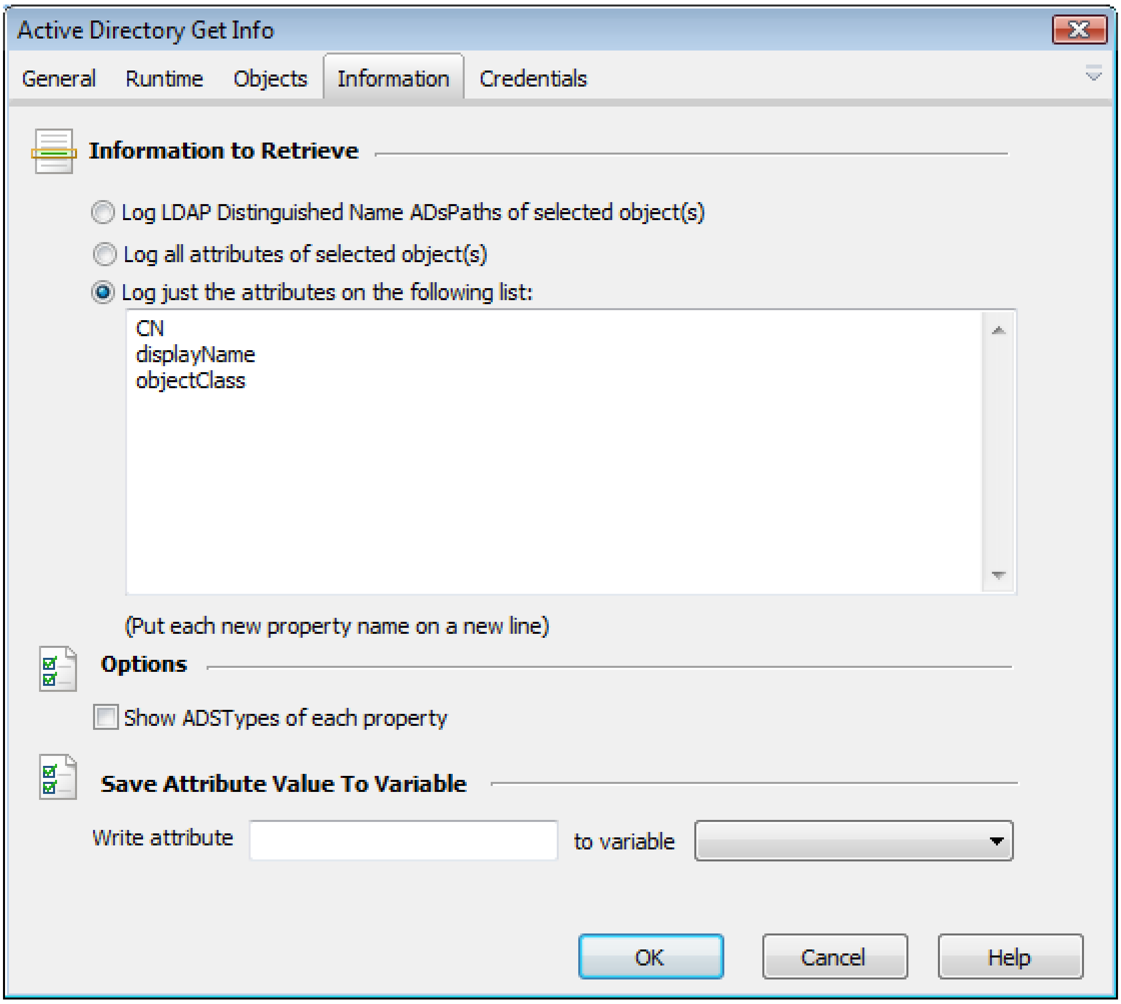
Click the "Add Container" or "Add Object" button to show dialogs allowing you to search for specific objects or containers.

"Select via an ADO query (SQL or LDAP syntax)"

Specify an ADO query (either in SQL syntax, as shown above, or by using an LDAP query.) The query must return the ADsPath field, as shown above.

"For results which are container objects..."

If you want to Get Info on the contents of one or more containers, specify the containers above and then check this box. Information will not be shown for the container objects themselves, just for each object within the container.



Active Directory Get Info

Information to Retrieve

○ Log LDAP Distinguished Name ADsPaths of selected object(s)
○ Log all attributes of selected object(s)
● Log just the attributes on the following list:

CN
displayName
objectClass

(Put each new property name on a new line)

Options

□ Show ADSTypes of each property

Save Attribute Value To Variable

Write attribute to variable

Information to Retrieve

Specify what information you want written to the log. You can choose to log just the ADsPath of each object, all attributes of each object, or a specific list of attributes (as shown in the screenshot.)

Note that some attributes with some types cannot be displayed. (Types which can be displayed include strings, booleans, integers, octet strings (displayed as SSIDs or hex byte strings), and large integers.) These attributes will appear to have no value.

Show ADSTypes of each property

Enable this option to log the ADSType of each attribute shown.

Save Attribute Value To Variable

You can choose to log the values of a specific attribute to an Automise variable. To do so, specify the name of an Active Directory attribute (or ADsPath, which is not an actual attribute but is treated as one for the purposes of this field), and a variable name.

The value of the chosen attribute will be written to the variable when the action runs. If you choose to Get Info on multiple objects, the variable will be written as a string list of values, which each value on a new line.

Note that (as above) only some ADS types can be written to variables.

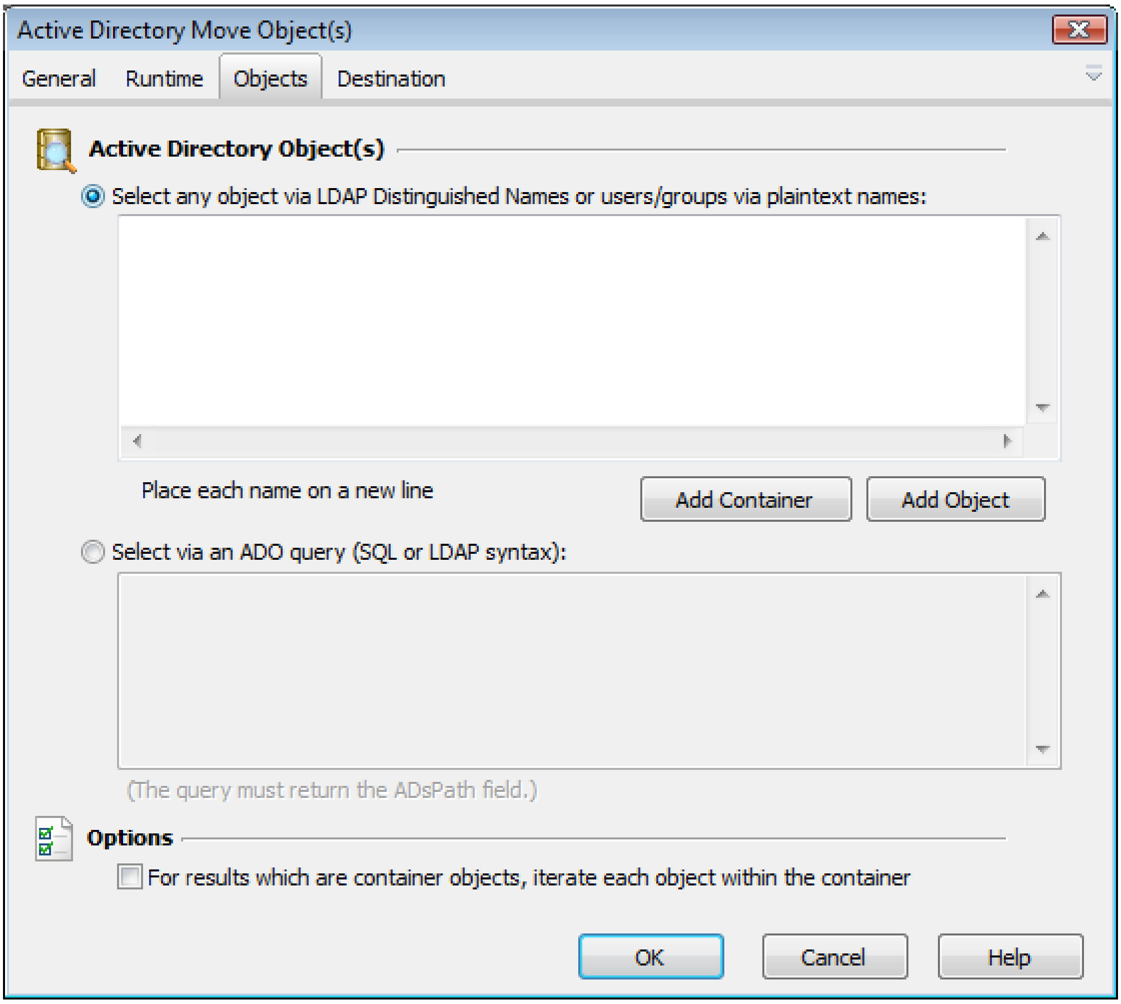
Credentials

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.

6.27.2.8 Active Directory Move Object(s)

This action allows you to move one or more objects from one container (or OU) to another.

Note: This action should be used with caution, as moving objects can severely affect the functionality of Active Directory.



Active Directory Move Object(s)

Active Directory Object(s)

● Select any object via LDAP Distinguished Names or users/groups via plaintext names:

Place each name on a new line

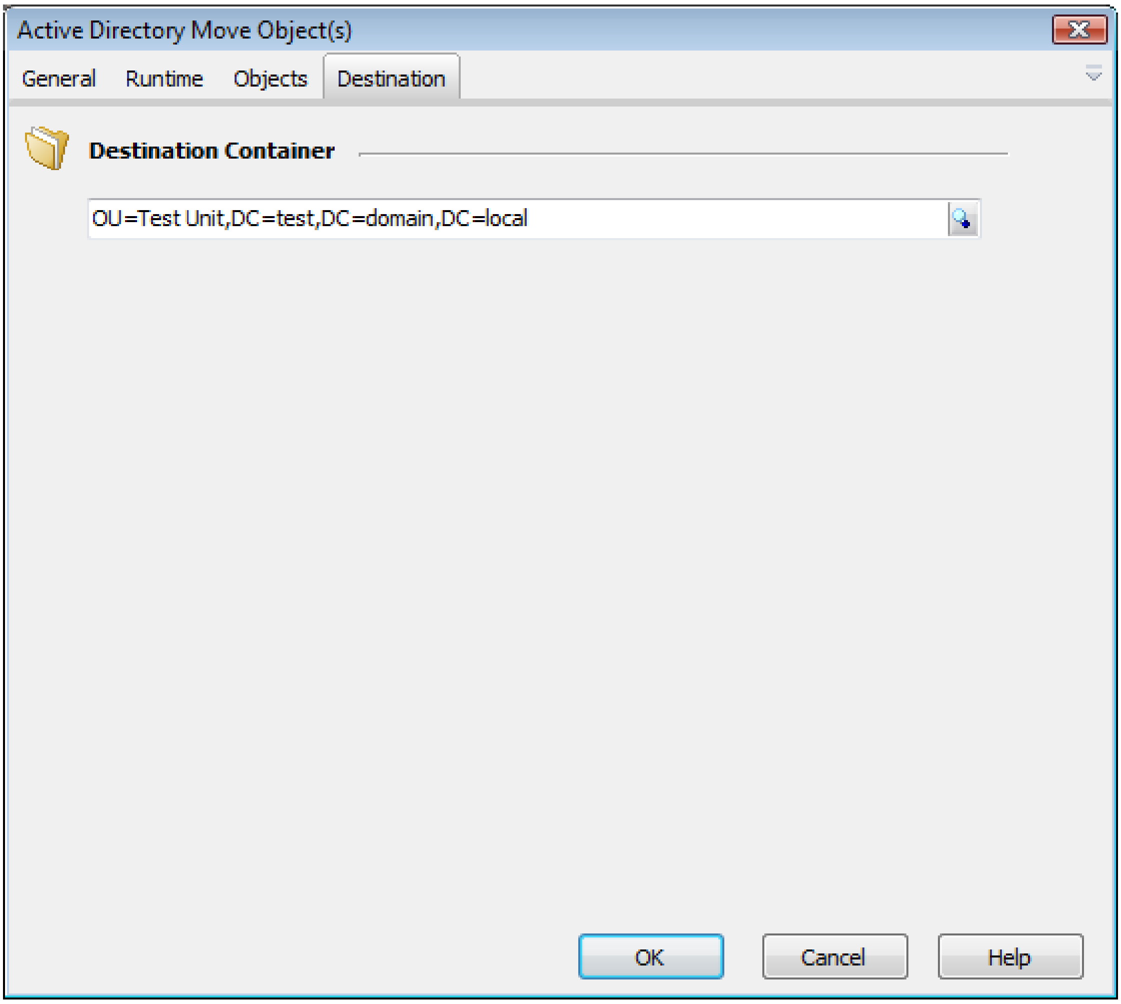
○ Select via an ADO query (SQL or LDAP syntax):

(The query must return the ADsPath field.)

Options

□ For results which are container objects, iterate each object within the container

Select the object(s) to move, either by name or by an ADO query. See the Active Directory Get Info action for details.



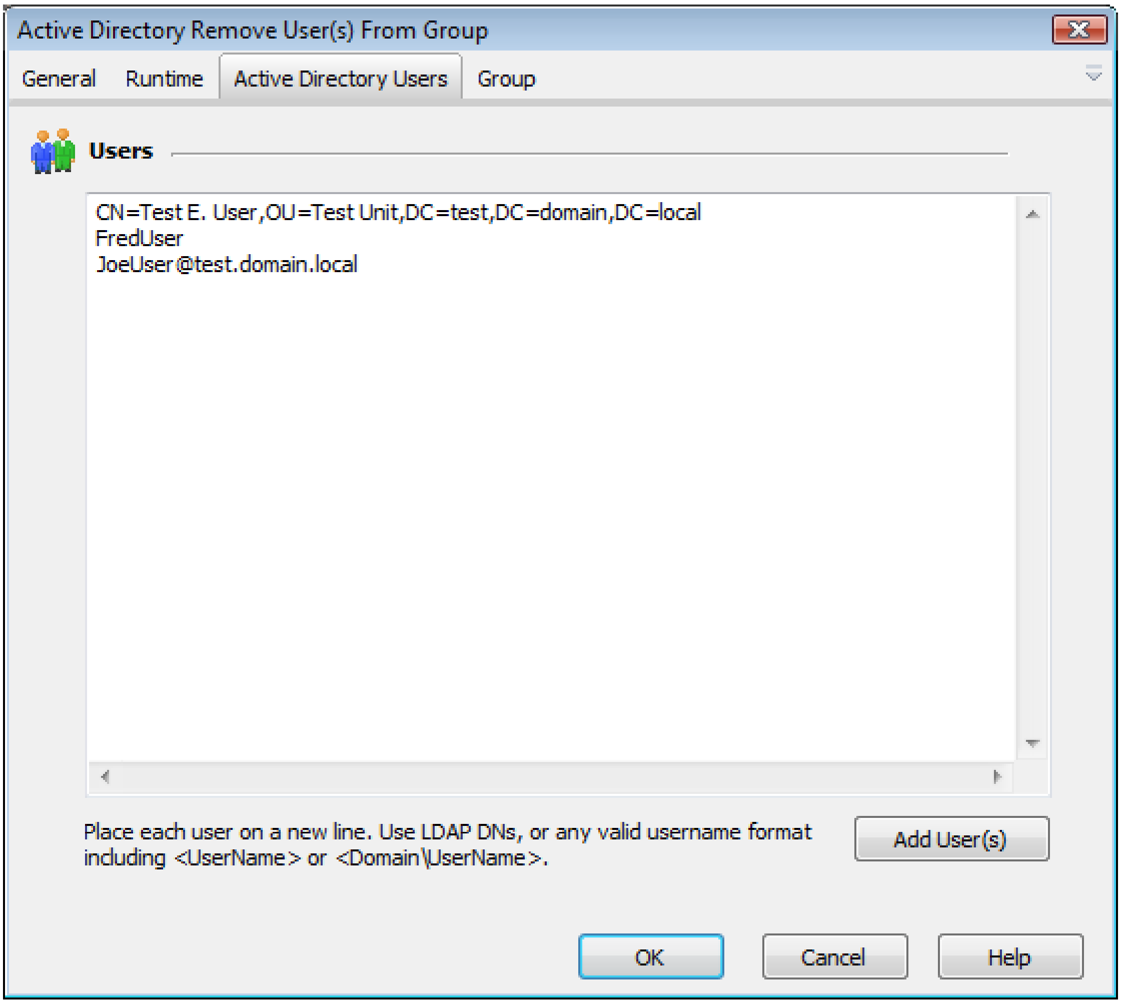
OU=Test Unit,DC=test,DC=domain,DC=local

Destination Container

Specify the Distinguished Name of the destination container. Click the magnifying glass button to browse for a container.

6.27.2.9 Active Directory Remove User(s) From Group

The Active Directory Add User(s) To Group action allows you to remove one or more users to an Active Directory Group.



CN=Test E. User,OU=Test Unit,DC=test,DC=domain,DC=local
FredUser
JoeUser@test.domain.local

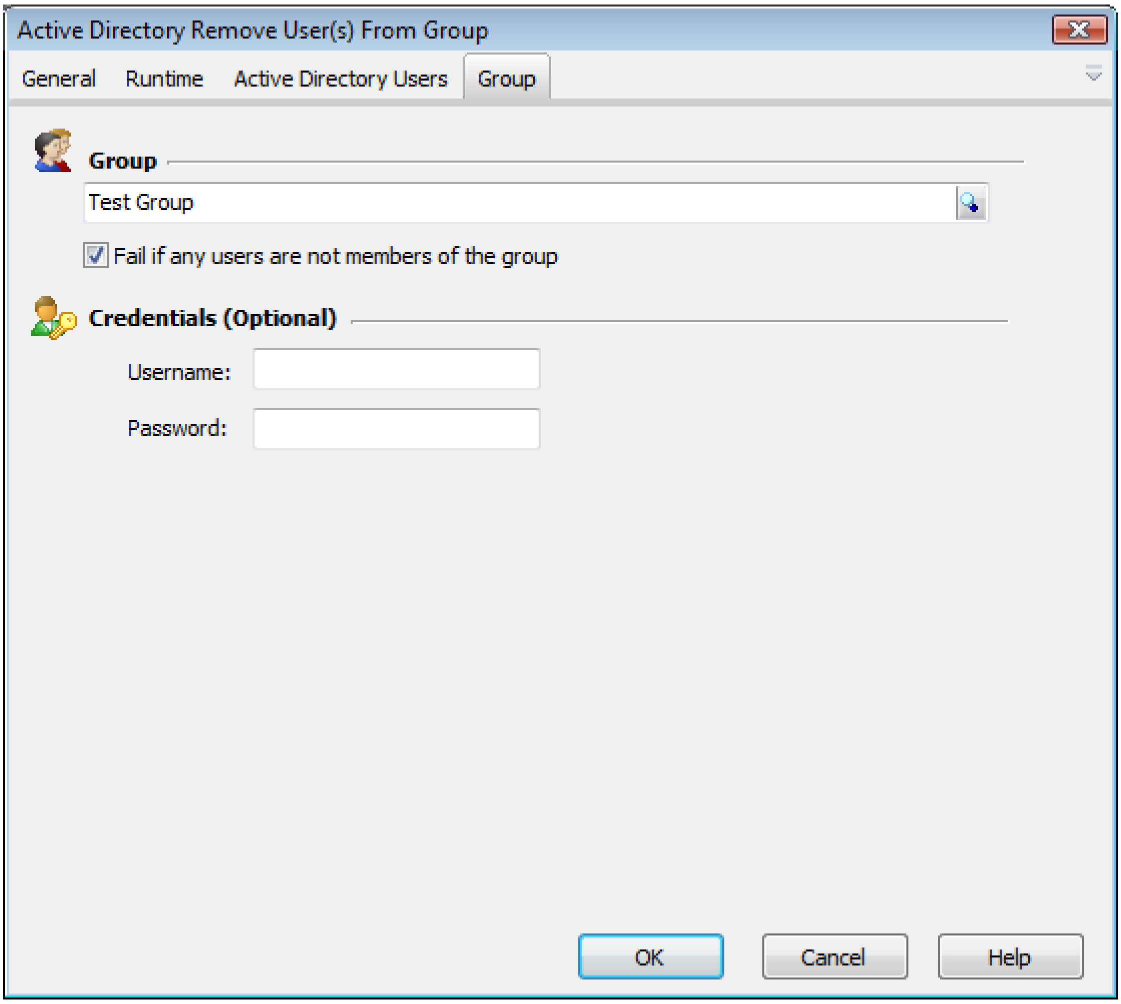
Place each user on a new line. Use LDAP DNs, or any valid username format including <UserName> or <Domain\UserName>.

Users

Users can be specified as LDAP Distinguished Names, or any valid username format - including simple usernames, Domain\UserName syntax, or User@Domain syntax. The LDAP://domainName/ prefix can be used as well.

Click the Add User(s) button to display a dialog allowing you to search for one or more users to add to the list.

Tip: To remove all of the users from a group, use the Enumerate Group action to write all of the group members' DNs to a variable, then specify the variable name in the Users field, using the %VarName% syntax.



Group

Test Group

☑ Fail if any users are not members of the group

Credentials (Optional)

Username:

Password:

Group

Specify the name of the group to remove users from. Groups can be specified as a simple group name or an LDAP Distinguished Name.

Fail if any users are not members of the group

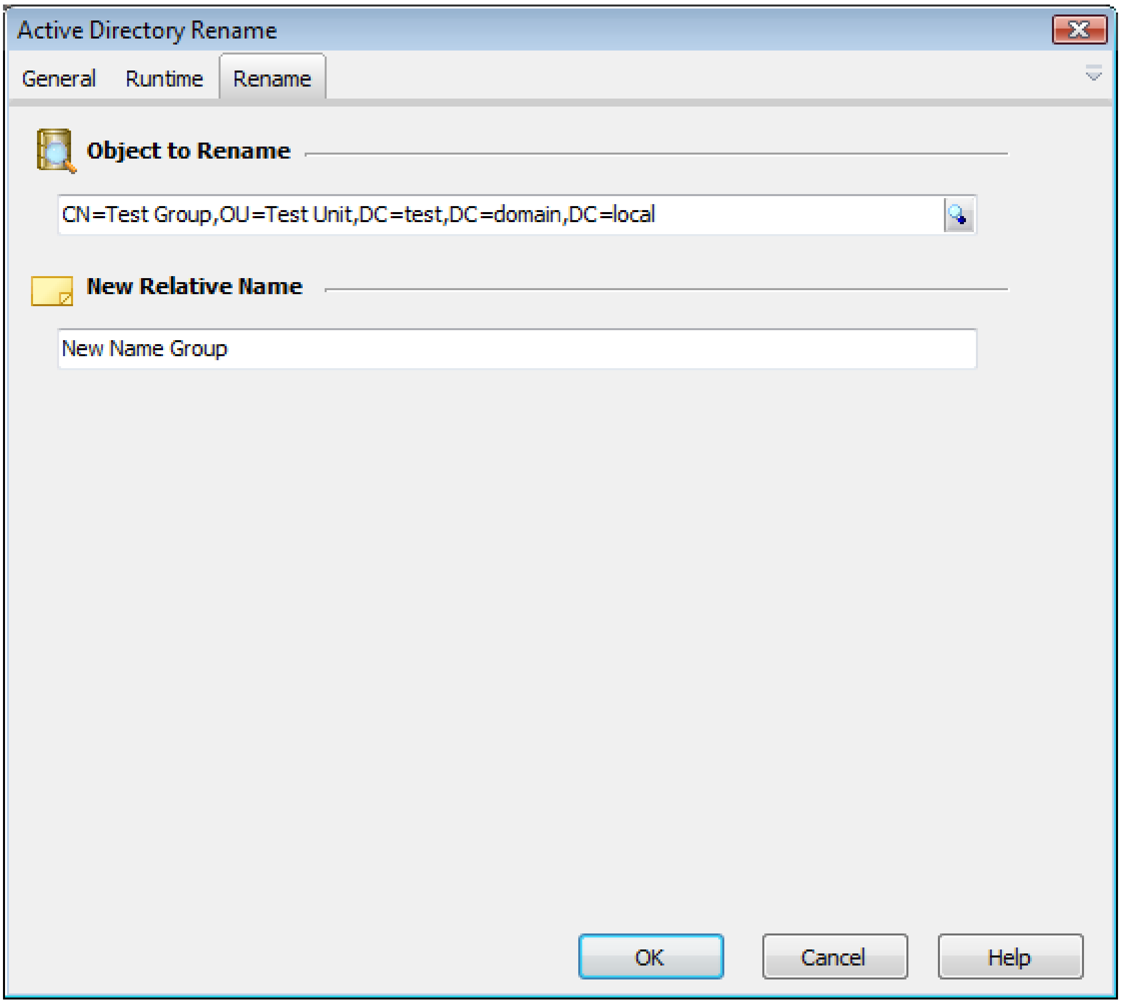
Check this box to fail the action if any users are not members of the group. Note that if the action fails for this reason, some other users may have already been removed from the group.

Credentials (Optional)

You can choose to specify a username and password to use when connecting to the Active Directory. If no username is specified, the current Windows credentials will be used.

6.27.2.10 Active Directory Rename

The Active Directory Rename action allows you to rename an object.



CN=Test Group,OU=Test Unit,DC=test,DC=domain,DC=local

New Name Group

Object to Rename

Specify the Distinguished Name of an object to rename. Click the magnifying glass button to browse for an object.

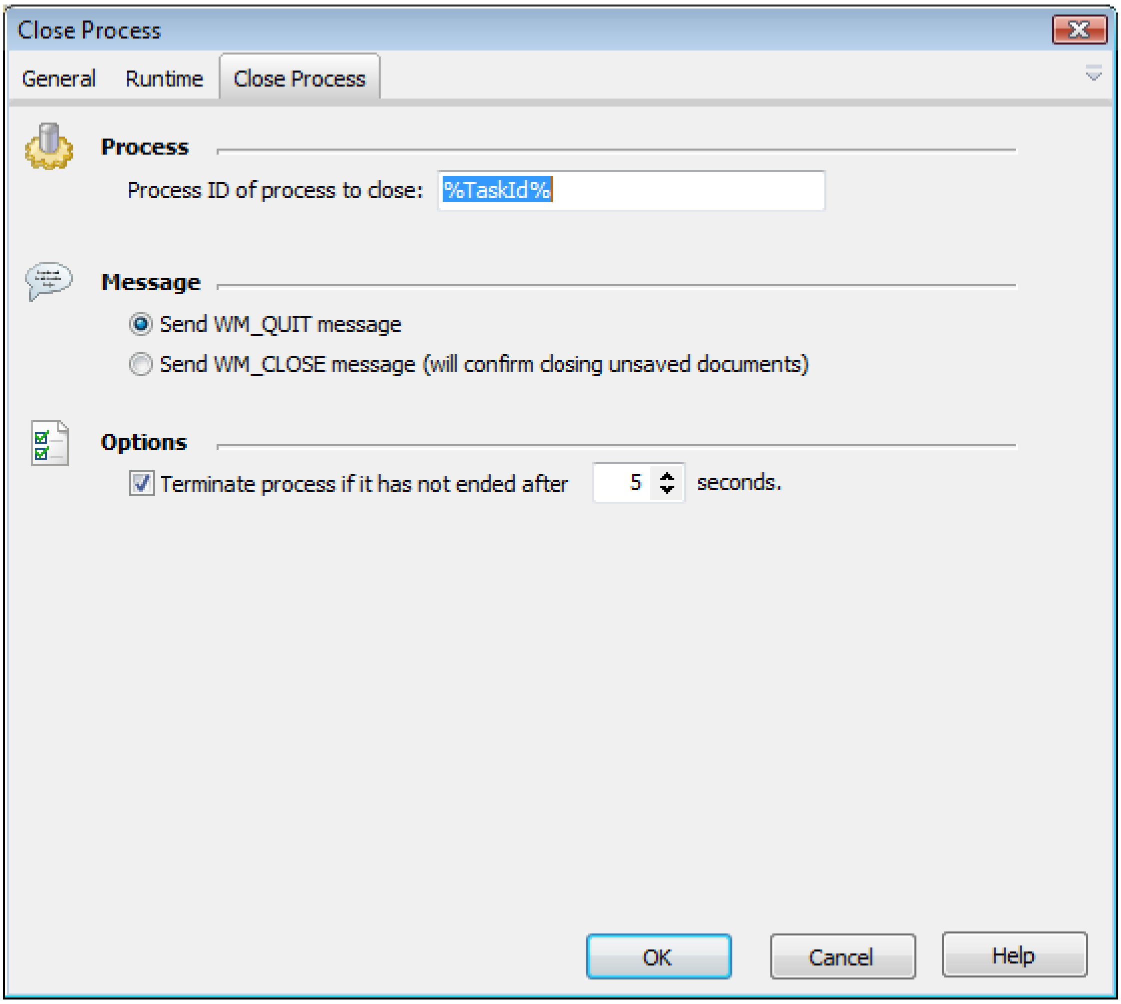
New Relative Name

Specify the new name (CN or relative name) of the object.

6.27.3 Close Process Action

The Close Process action allows you send a quit or a close message to a process running on the local machine.

To terminate an action outright, you can use the WMI Kill Process action.



Process

Process ID of process to close: %TaskId%

Message

● Send WM_QUIT message

○ Send WM_CLOSE message (will confirm closing unsaved documents)

Options

☑ Terminate process if it has not ended after 5 seconds.

Process

Specify the Process ID number of the process to close. The ID can be specified by variable, as in the above example.

To find the Process IDs of an action (or group of actions), you can use the WMI Process Iterator or WMI Process Info actions.

Message

You can choose between sending the WM_QUIT or WM_CLOSE messages. The message will be sent to application's main window (you cannot close command line applications with this action - use WMI Kill Process instead.)

If WM_CLOSE is sent, the application will not pause for confirmation before closing.

"Terminate process if it has not ended..."

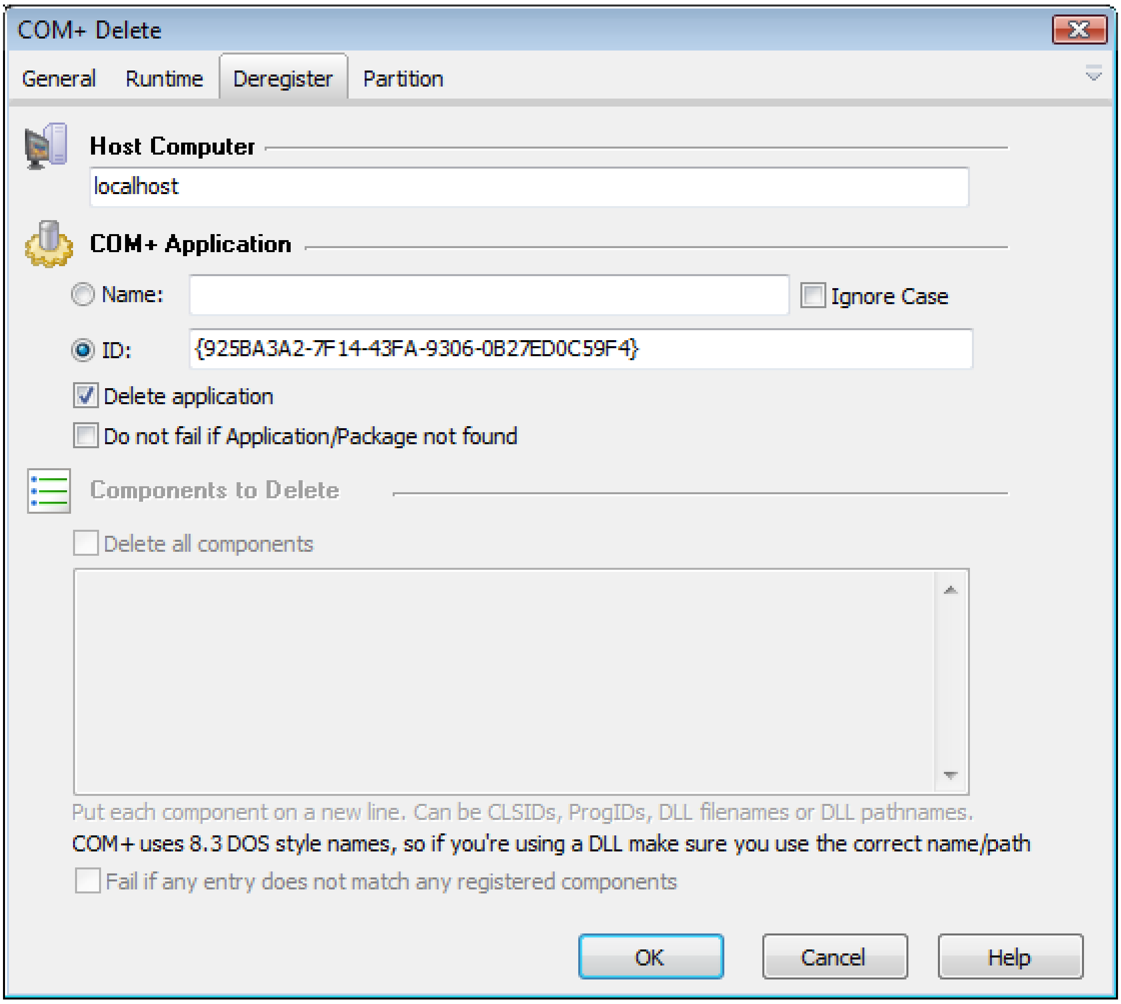
This option sets a timeout after which a process is forcibly terminated. This is useful if you would like to terminate a possibly unstable process, or a process which has paused waiting for confirmation of unsaved data (see WM_CLOSE in the Message section.)

If this option is not checked, the action will not pause to confirm whether or not the process has ended.

6.27.4 COM+ Administration

6.27.4.1 COM+ Delete

The COM+ Delete Action allows you to delete registered COM+ applications or COM+ components.



Host Computer

localhost

COM+ Application

○ Name: □ Ignore Case

● ID: {925BA3A2-7F14-43FA-9306-0B27ED0C59F4}

☑ Delete application

□ Do not fail if Application/Package not found

Components to Delete

□ Delete all components

Put each component on a new line. Can be CLSIDs, ProgIDs, DLL filenames or DLL pathnames.

COM+ uses 8.3 DOS style names, so if you're using a DLL make sure you use the correct name/path

□ Fail if any entry does not match any registered components

Host Computer

The name of the computer you want to administrate. You will need Administrator privileges on this computer. Blank is equivalent to localhost.

COM+ Application

You can specify the COM+ Application by its name or ID (GUID.) The action will fail if no application is found.

Check the "**Delete application**" box to remove this application from the server.

Components To Delete

If the "**Delete application**" option is not set, you will need to specify which components to remove from the server. Put each component on a new line in the memo box.

Components can be specified as component names (ProgIDs, as shown in the Component Services Components list), CLSIDs (GUIDs) or DLL filenames or pathnames (partial pathnames are also acceptable.) If DLL names are specified, all components belonging to that DLL will be deleted.

Check the "**Delete all components**" box to remove all components.

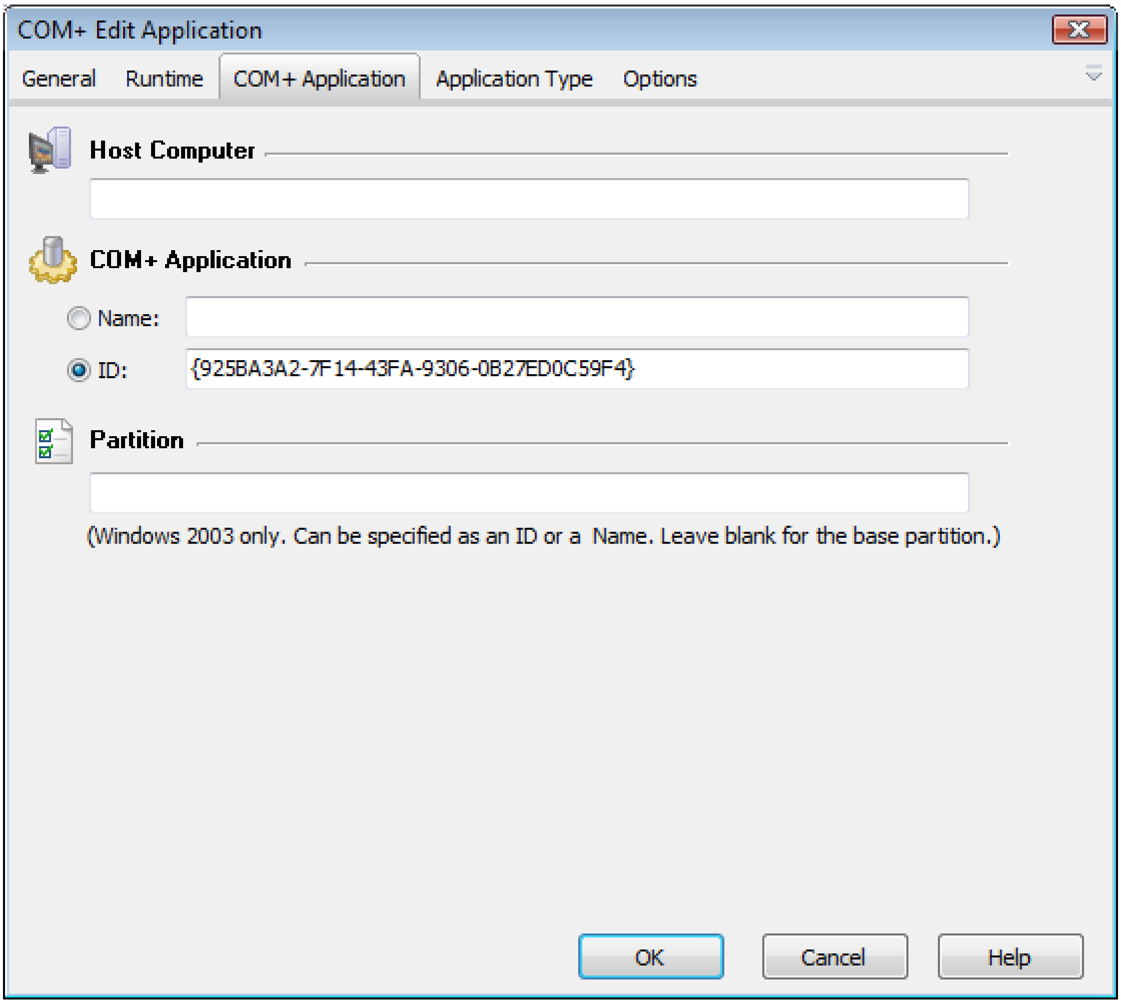
Partition

To specify an application on a different COM+ Partition, enter the name or ID (GUID) of the partition on the partition page. If no partition is specified, the action operates on the base partition.

6.27.4.2 COM+ Edit Application

The COM+ Edit Application action allows you to edit the properties of a COM+ Application. This provides similar functionality to editing the properties in the Component Services Application Properties dialog (obtained by right-clicking on a COM+ Application and choosing "Properties.")

If you wish to edit a property which is not available on the property pages, you can use the EditApplication script event to make custom changes. See below for details.



Host Computer

COM+ Application

○ Name:

● ID: {925BA3A2-7F14-43FA-9306-0B27ED0C59F4}

Partition

(Windows 2003 only. Can be specified as an ID or a Name. Leave blank for the base partition.)

Host Computer

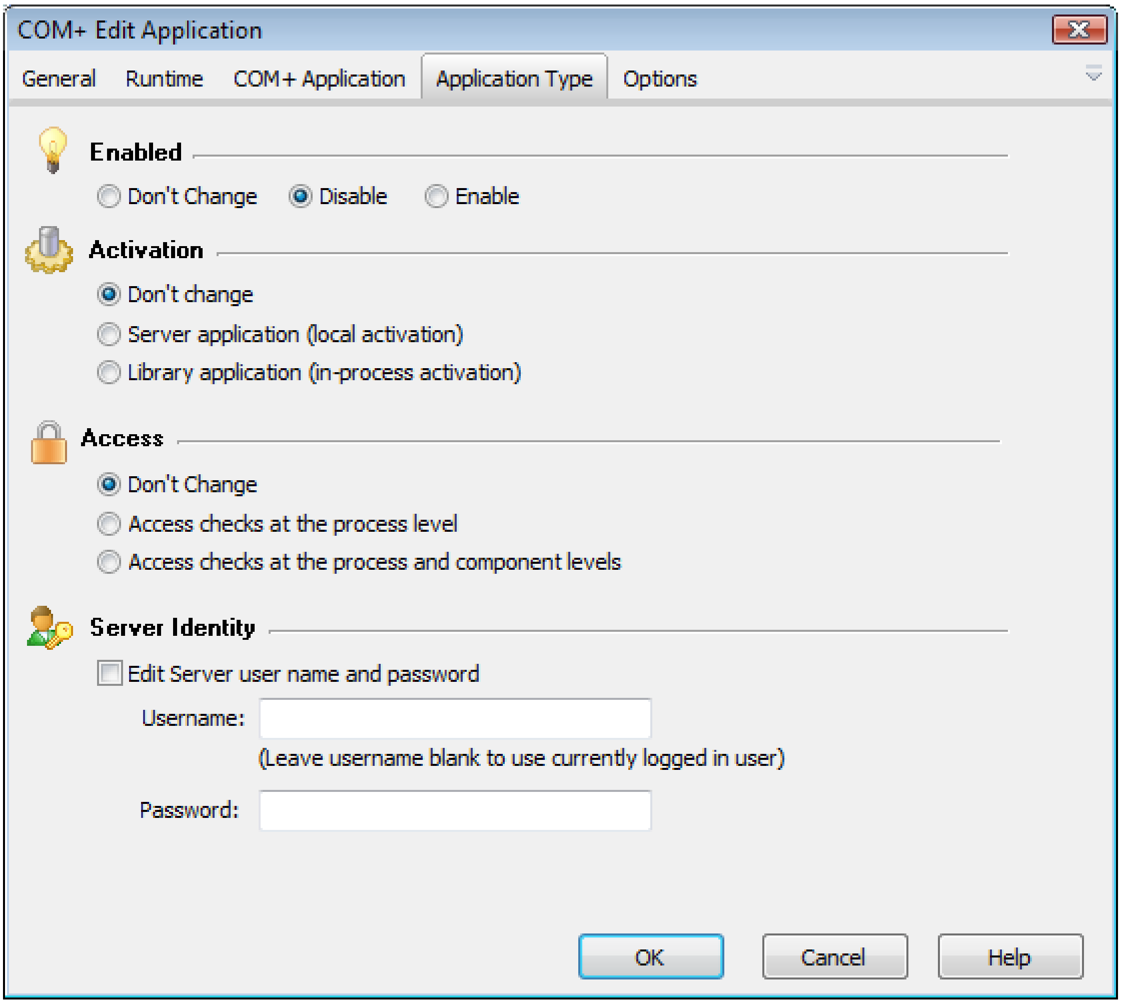
The computer you want to edit the Application on. You will need Administrator privileges on this computer. Blank is equivalent to localhost.

COM+ Application

You can specify an application by its name, or its unique ID (GUID.) At most one application will be edited (if you have more than one application with the same name, only one will be edited.)

Partition

To specify an application on a different COM+ Partition, enter the name or ID (GUID) of the partition. If no partition is specified, the action operates on the base partition.



COM+ Edit Application

Enabled

○ Don't Change ● Disable ○ Enable

Activation

● Don't change
○ Server application (local activation)
○ Library application (in-process activation)

Access

● Don't Change
○ Access checks at the process level
○ Access checks at the process and component levels

Server Identity

□ Edit Server user name and password

Username:

(Leave username blank to use currently logged in user)

Password:

Enabled

You can choose to enable or disable an application, or leave its state alone. This is equivalent to right-clicking on an application in Component Services and choosing "Enable" or "Disable" from the contextual menu.

(Note that enabling and disabling COM+ applications is not the same as Shutting Down and Starting Up COM+ applications. Use the COM+ Shutdown Application and COM+ Start Application actions for these purposes.)

Activation

You can choose to set the Application to run as a Server application (with local process activation) or as a Library application (activated in-process by the caller.)

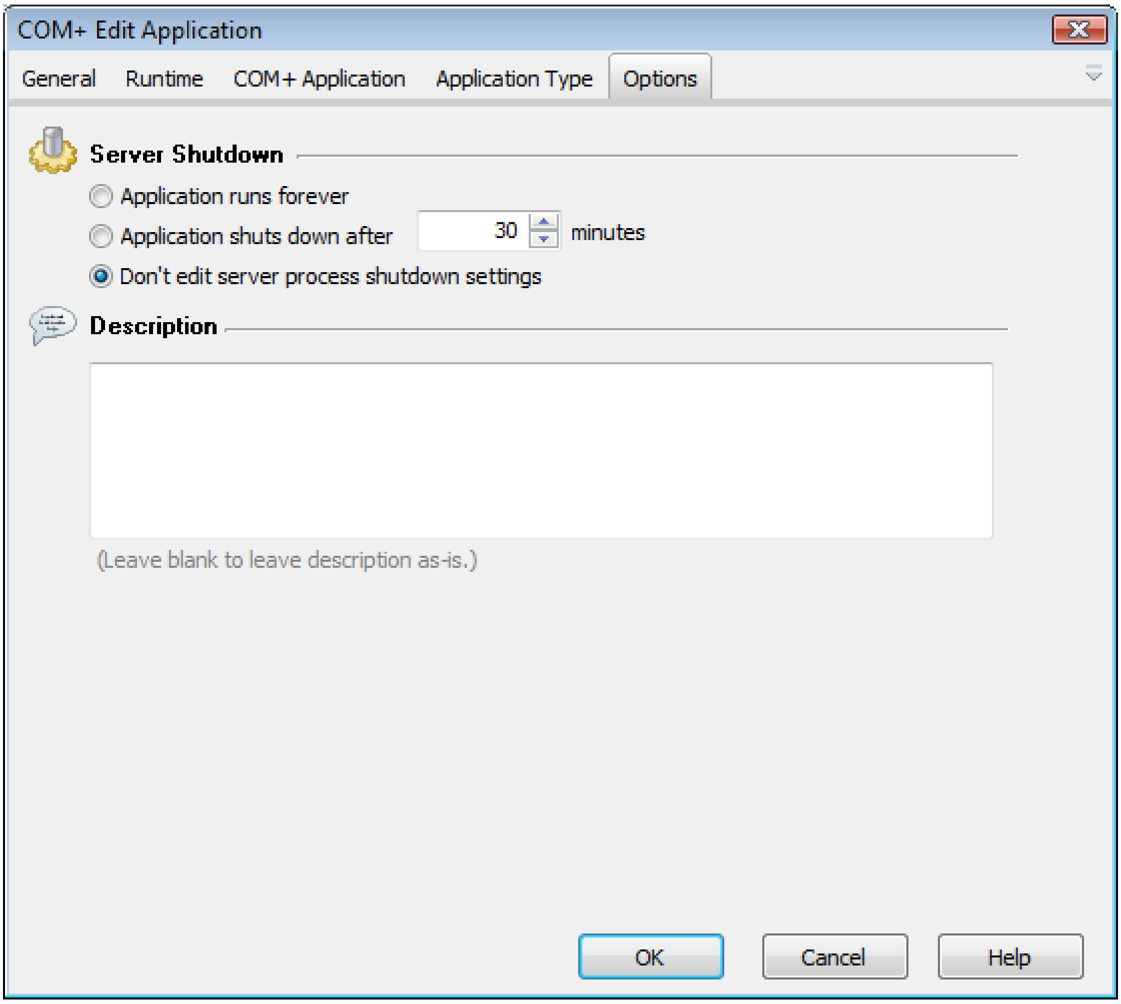
If you set the application to run as a Server application, you can choose credentials for its execution in the Server Identity section.

Access

You can choose to set the Application to enable access checks at the process level, or at both process and component levels. Note that, depending on your security configuration, changes made to this property may or may not take effect.

Server Identity

Check the "Edit Server user name and password" box to set the username and password used when an Application runs as a Server process.



COM+ Edit Application

Server Shutdown

○ Application runs forever

○ Application shuts down after 30 minutes

● Don't edit server process shutdown settings

Description

(Leave blank to leave description as-is.)

Server Shutdown

(This option is only meaningful when an Application is running as a Server.)

Set "Application runs forever" to have an application which never shuts down when idle. Alternatively, set an idle time after which the application will shut down.

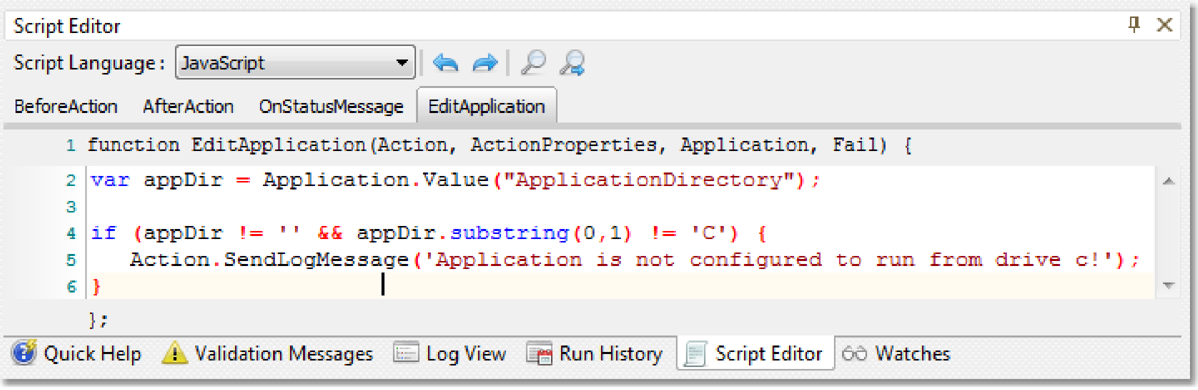
Description

Enter a description to replace the existing description of the application. If left blank, the description will not be updated.

Edit Application Script Event

To perform custom edits on Applications, the COM+ Edit Application action provides a custom script event. To access it, select the Edit Application action, then click on the Script Events tab, then click on "EditApplication."

Here is a sample script event written in JavaScript:



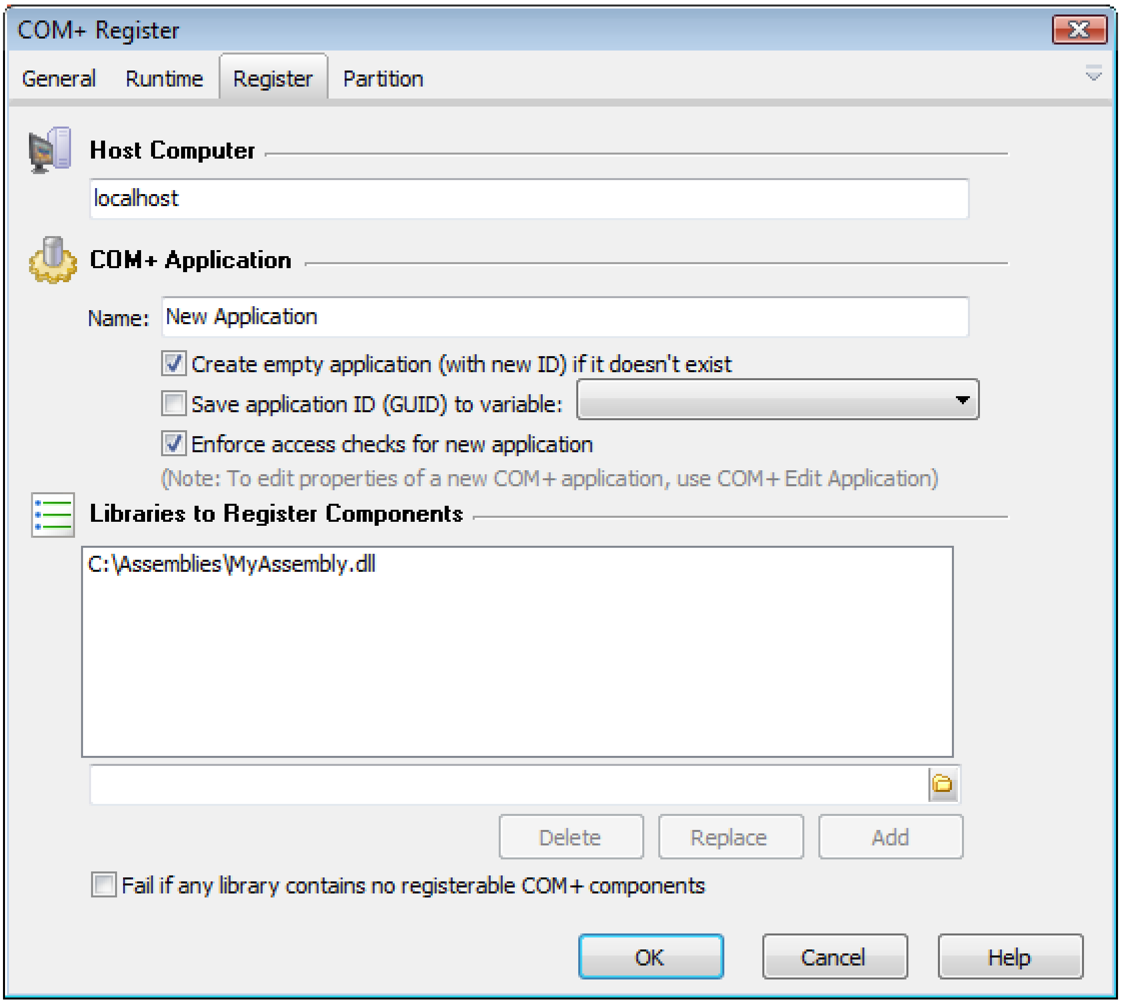
```
function EditApplication(Action, ActionProperties, Application, Fail) {
var appDir = Application.Value("ApplicationDirectory");

if (appDir != '' && appDir.substring(0,1) != 'C') {
   Action.SendLogMessage('Application is not configured to run from drive c!');
}
};
```

To read and write Application properties, use the syntax Application.Value ("PropertyName"). Set the Fail parameter to true to force the action to fail programmatically.

For a full list of available properties for Application administration objects, see the MSDN article at:
MSDN Library -> Win32 and COM Development -> Component Development -> COM+ (Component Services) -> SDK Documentation -> COM+ (Component Services) -> COM+ Reference -> COM+ Administration Reference > COM+ Administration Collections**.**

6.27.4.3 COM+ Register

The COM+ Register action allows you to register one or more components from local DLLs into a COM+ Application. Components can be registered to an existing application or a new application can be created.



Host Computer: localhost

Name: New Application

☑ Create empty application (with new ID) if it doesn't exist

□ Save application ID (GUID) to variable:

☑ Enforce access checks for new application

(Note: To edit properties of a new COM+ application, use COM+ Edit Application)

C:\Assemblies\MyAssembly.dll

□ Fail if any library contains no registerable COM+ components

Host Computer

The computer you want to register the components on. You will need Administrator privileges on this computer. Leave blank for localhost.

COM+ Application

Enter the name of an existing COM+ application, or the name of the new application to create.

Note: If you are creating an application, you will probably want to use the COM+ Edit Application action to set its properties.

Save application ID (GUID) to variable

If this option is checked, then the unique ID string of the COM+ Application (existing or new) will be saved to a Automise variable.

Enforce access checks for new application

This box is equivalent to the checkbox on the Security tab in the Component Services "Application Properties" screen.

Libraries to Register Components

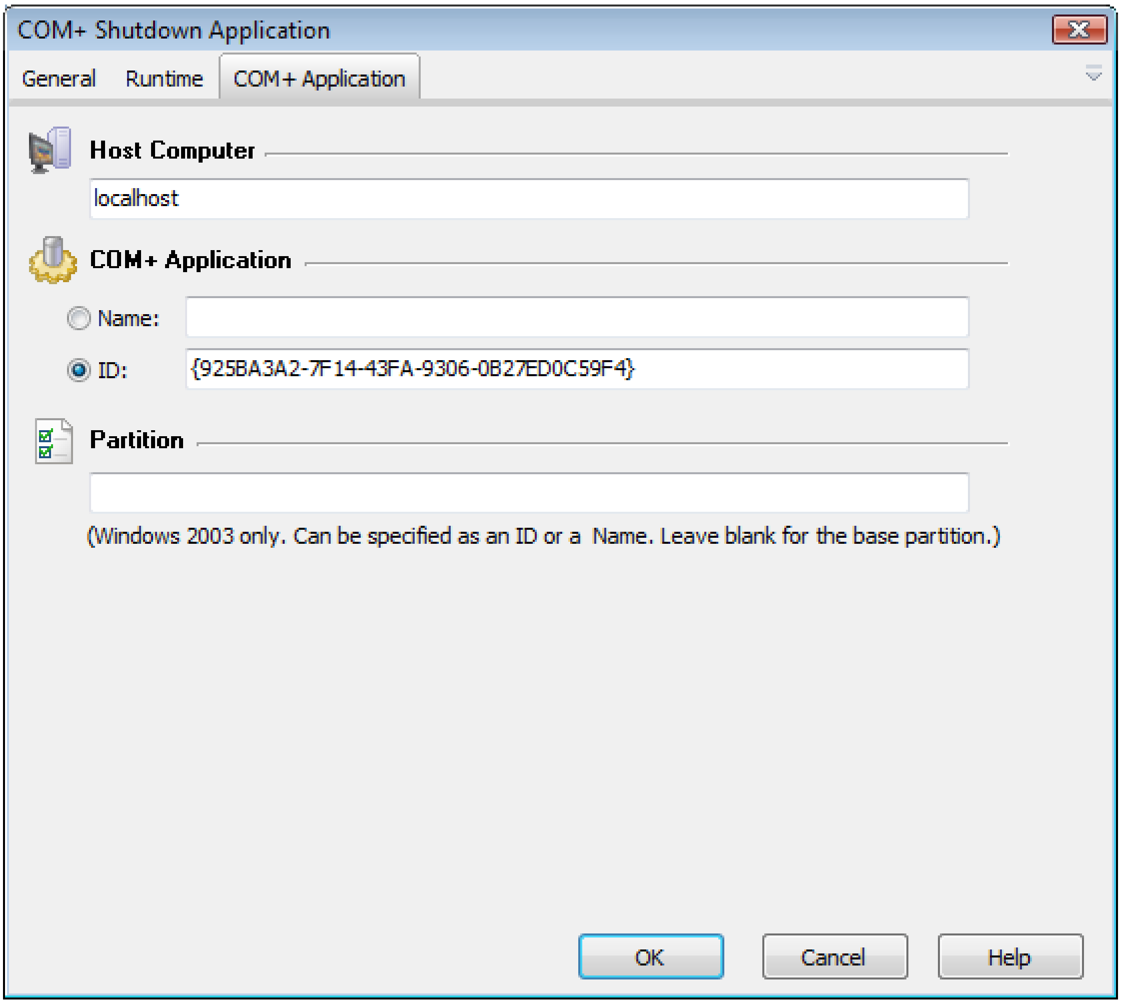
Add the shared libraries that you wish to load COM+ components from. Check the "**Fail if any library contains no registerable COM+ components**" to have the action fail if any DLL does not contain any components.

Partition

To register the application on a COM+ Partition, enter the name or ID (GUID) of the partition on the partition page. If no partition is specified, the action operates on the base partition.

6.27.4.4 COM+ Shutdown Application

The COM+ Shutdown Application action allows you to shut down a running COM+ Application.



COM+ Shutdown Application

General | Runtime | COM+ Application

Host Computer

localhost

COM+ Application

○ Name:

● ID: {925BA3A2-7F14-43FA-9306-0B27ED0C59F4}

Partition

(Windows 2003 only. Can be specified as an ID or a Name. Leave blank for the base partition.)

Host Computer

The computer you want to shut down the application on. You will need Administrator privileges on this computer. Leave blank for localhost.

COM+ Application

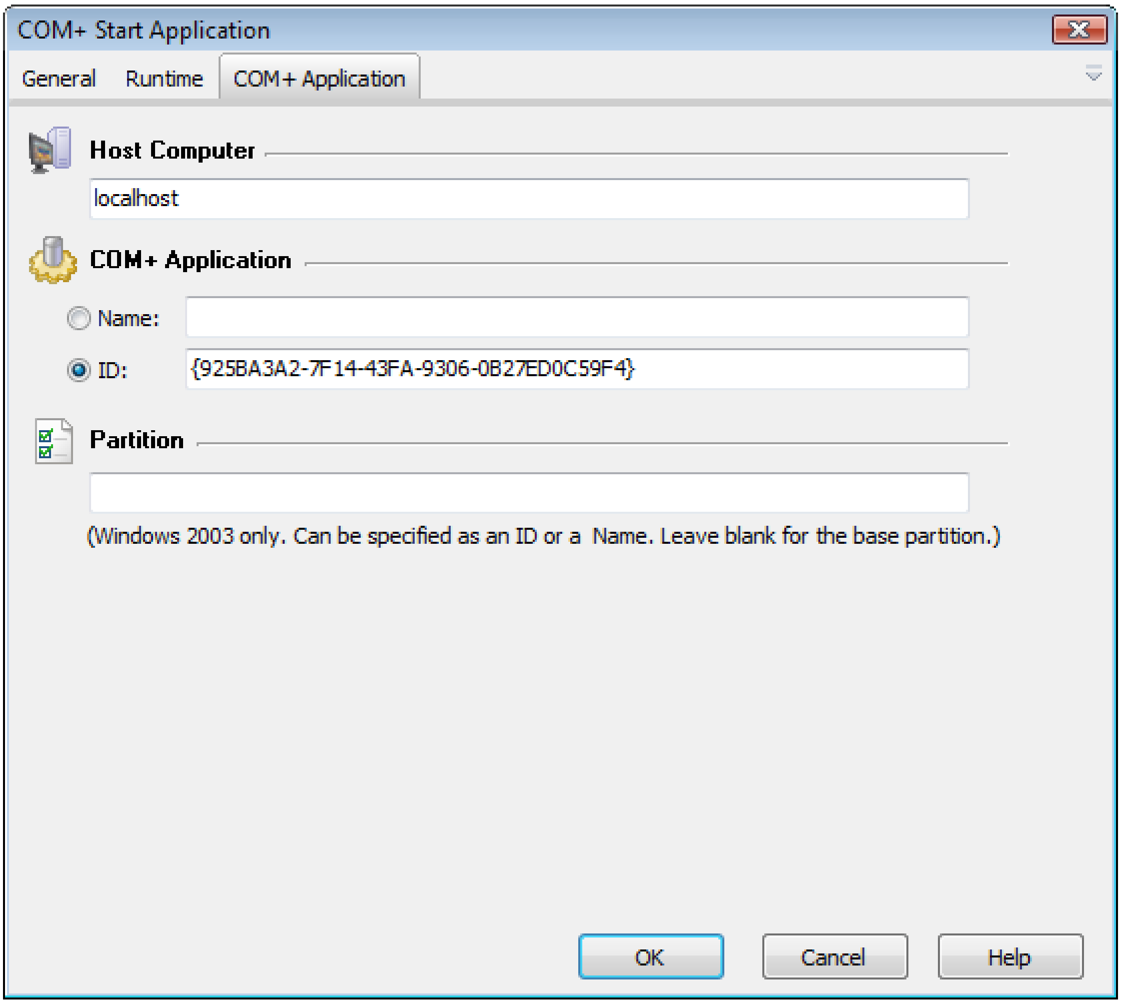
You can specify the application name by its name or its ID (GUID.) In the example above, the application is being referenced via a Automise variable drawn from a COM+ Register action.

Partition

To specify an application on a different COM+ Partition, enter the name or ID (GUID) of the partition. If no partition is specified, the action operates on the base partition.

6.27.4.5 COM+ Start Application

The COM+ Start Application action allows you to start a shut down or inactive COM+ Application.



Host Computer

localhost

COM+ Application

○ Name:

● ID: {925BA3A2-7F14-43FA-9306-0B27ED0C59F4}

Partition

(Windows 2003 only. Can be specified as an ID or a Name. Leave blank for the base partition.)

Host Computer

The computer you want to shut down the application on. You will need Administrator privileges on this computer. Blank is equivalent to localhost.

COM+ Application

You can specify the application name by its name or its ID (GUID.)

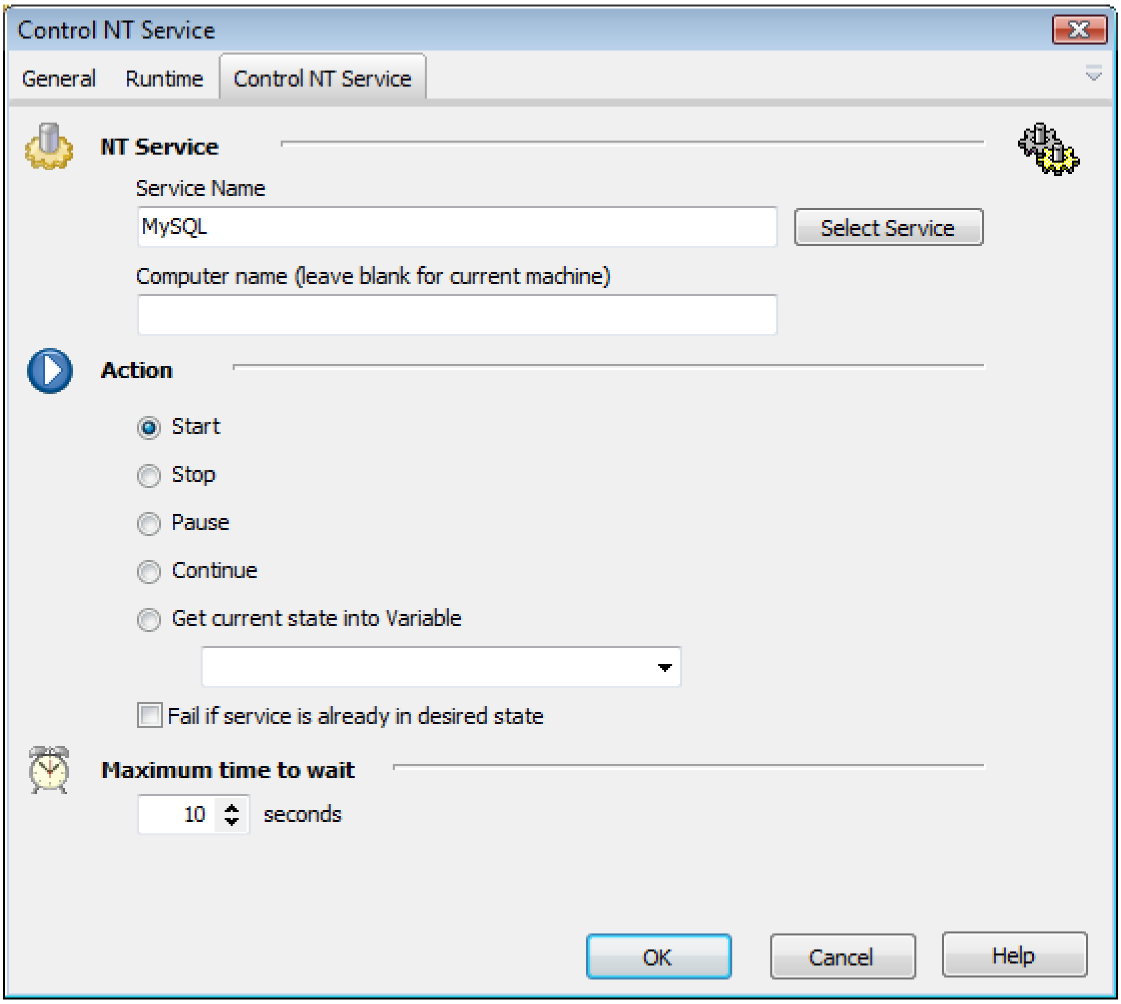
Partition

To specify an application on a different COM+ Partition, enter the name or ID (GUID) of the partition. If no partition is specified, the action operates on the base partition.

6.27.5 Control Service Action

[Automise Professional Edition]

This action allows you to control the state of a windows service on the local machine or a remote machine.



Control NT Service

General | Runtime | Control NT Service

NT Service

Service Name

MySQL

Select Service

Computer name (leave blank for current machine)

Action

● Start

○ Stop

○ Pause

○ Continue

○ Get current state into Variable

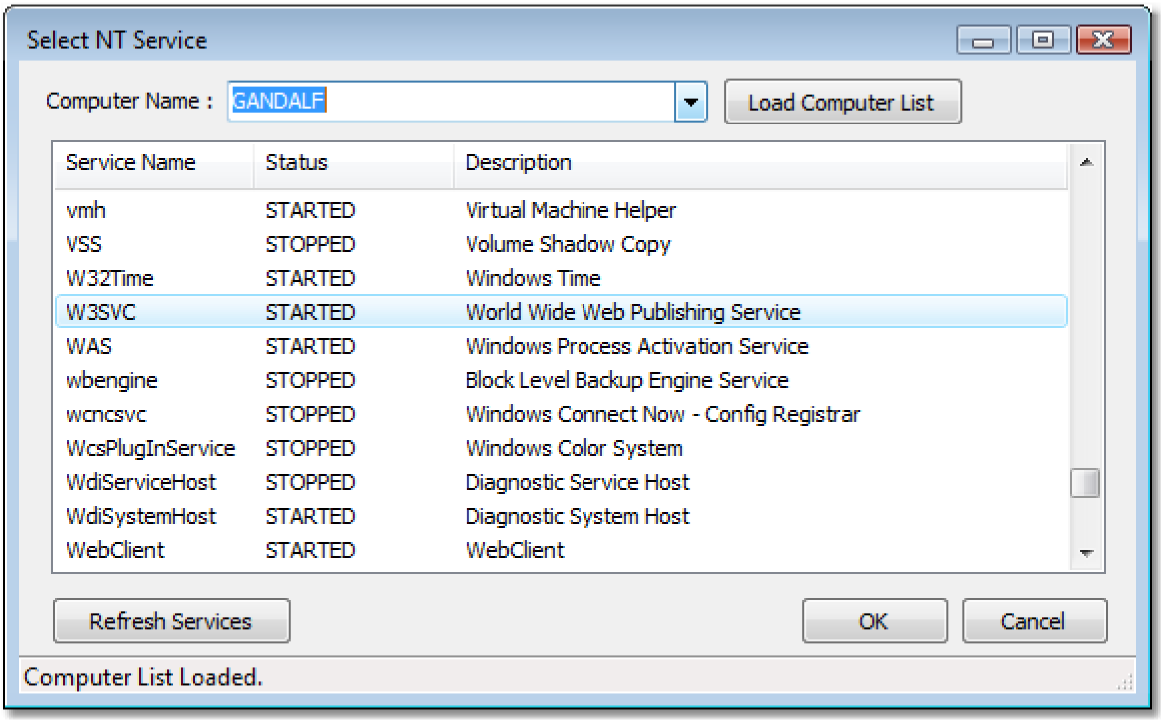
□ Fail if service is already in desired state

Maximum time to wait

10 seconds

OK Cancel Help

Click on the Select service to choose which service to control. By default the registered services of the local machine are listed, to change this either type the name of the computer or select from the drop down list (click on load to get a list, this is not done automatically as it can be slow in a large network) and the click on the Refresh Services button.



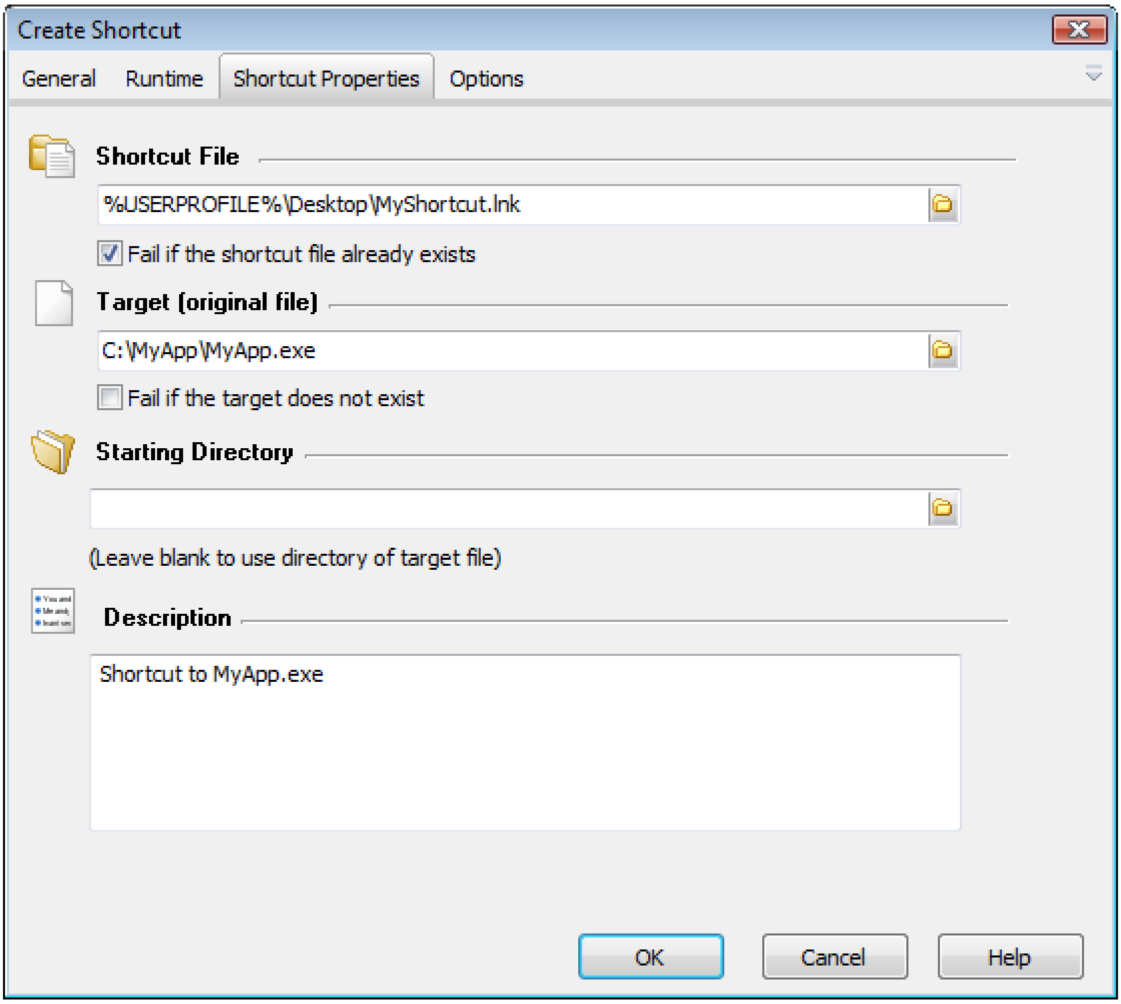
Computer Name : GANDALF

Service Name	Status	Description
vmh	STARTED	Virtual Machine Helper
VSS	STOPPED	Volume Shadow Copy
W32Time	STARTED	Windows Time
W3SVC	STARTED	World Wide Web Publishing Service
WAS	STARTED	Windows Process Activation Service
wbengine	STOPPED	Block Level Backup Engine Service
wcncsvc	STOPPED	Windows Connect Now - Config Registrar
WcsPlugInService	STOPPED	Windows Color System
WdiServiceHost	STOPPED	Diagnostic Service Host
WdiSystemHost	STARTED	Diagnostic System Host
WebClient	STARTED	WebClient

Computer List Loaded.

6.27.6 Create Shortcut

The Create Shortcut action allows you to create an Explorer shortcut (.lnk file) from one file to another.



Shortcut File

%USERPROFILE%\Desktop\MyShortcut.lnk

☑ Fail if the shortcut file already exists

Target (original file)

C:\MyApp\MyApp.exe

□ Fail if the target does not exist

Starting Directory

(Leave blank to use directory of target file)

Description

Shortcut to MyApp.exe

Shortcut File

The location where you want to create the shortcut file. Shortcuts must have the .lnk file extension.

"*Fail if the shortcut file already exists*"

If this option is set, the action will fail if the shortcut already exists. Otherwise, the existing shortcut will be replaced.

Target (original file)

This is the location of the target file that the shortcut links to.

"Fail if the target does not exist"

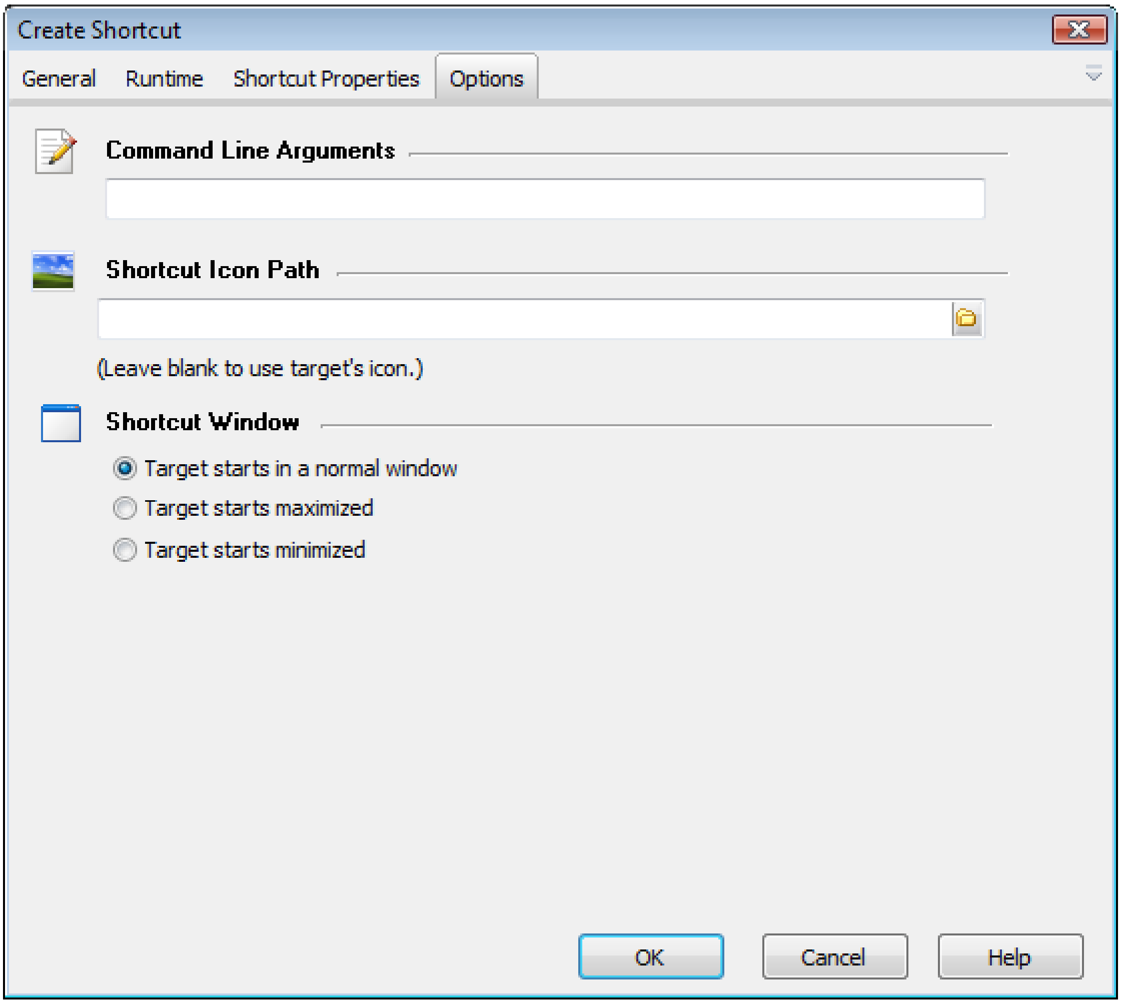
If this option is set, the action will fail if the target is not a valid pathname to an existing file. If this option is not set then the shortcut will be created, regardless.

Starting Directory

The working directory of the shortcut's application (when launched.) Optional. If left blank, the program will be started in the target file's parent directory.

Description

The Description will appear in the "Comment" field in Explorer Properties. Optional.



Command Line Arguments

Shortcut Icon Path

(Leave blank to use target's icon.)

Shortcut Window

● Target starts in a normal window

○ Target starts maximized

○ Target starts minimized

Command Line Arguments

Command line arguments to be passed to the Target (if it is an application.)

Shortcut Icon Path

Optionally specify a different file's icon for the shortcut.

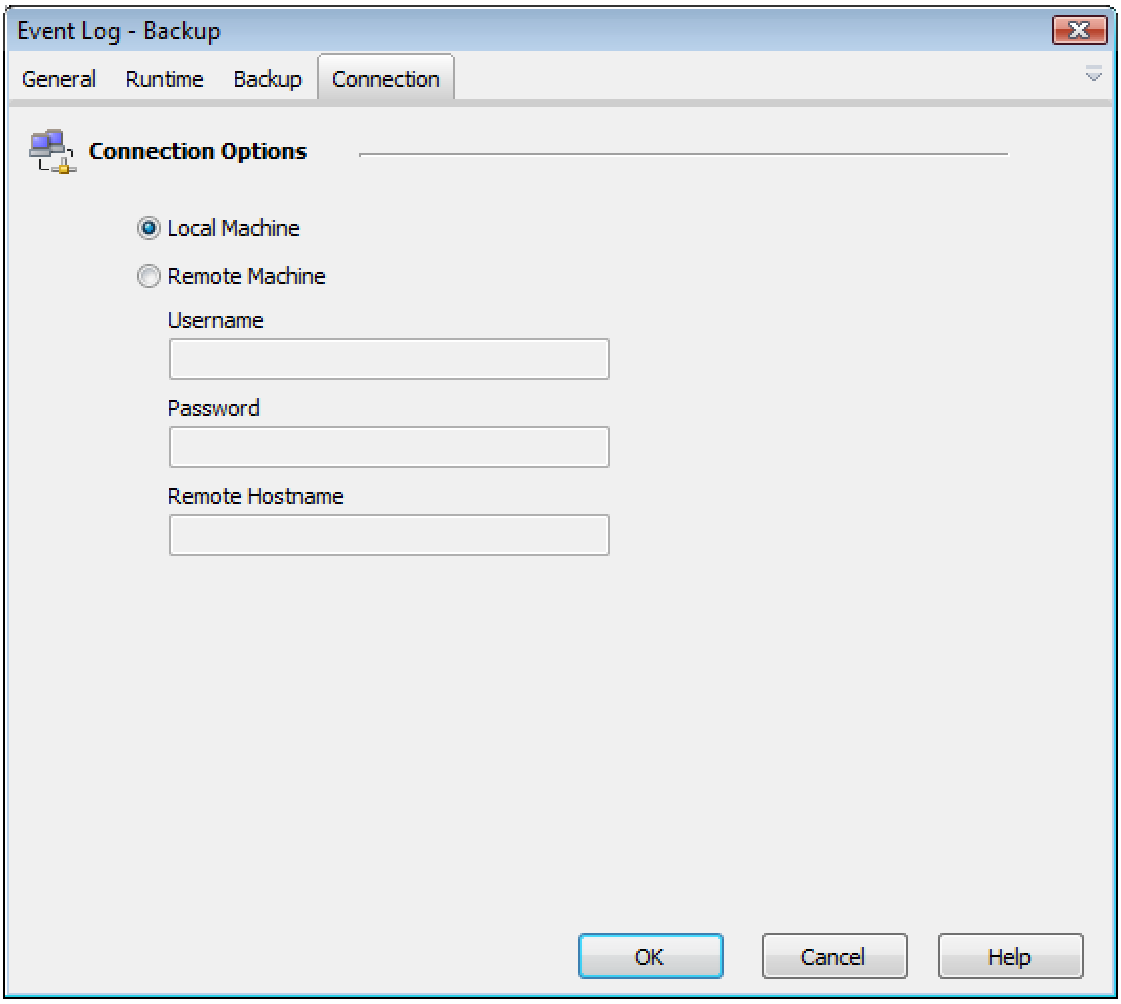
Shortcut Window

This option determines whether the target starts normally, in a maximized window, or in a minimized window in the background.

6.27.7 Event Log

The Event Log actions enable you to backup, clear or search the Windows Event Logs.

Each action has a common options page which allows you specify whether to connect to a remote machine, specifying a username and password, or to connect to the local machine as the current user.



● Local Machine

○ Remote Machine

Username

Password

Remote Hostname

Local Machine
The action will connect to the local machine, with the credentials of the current user.

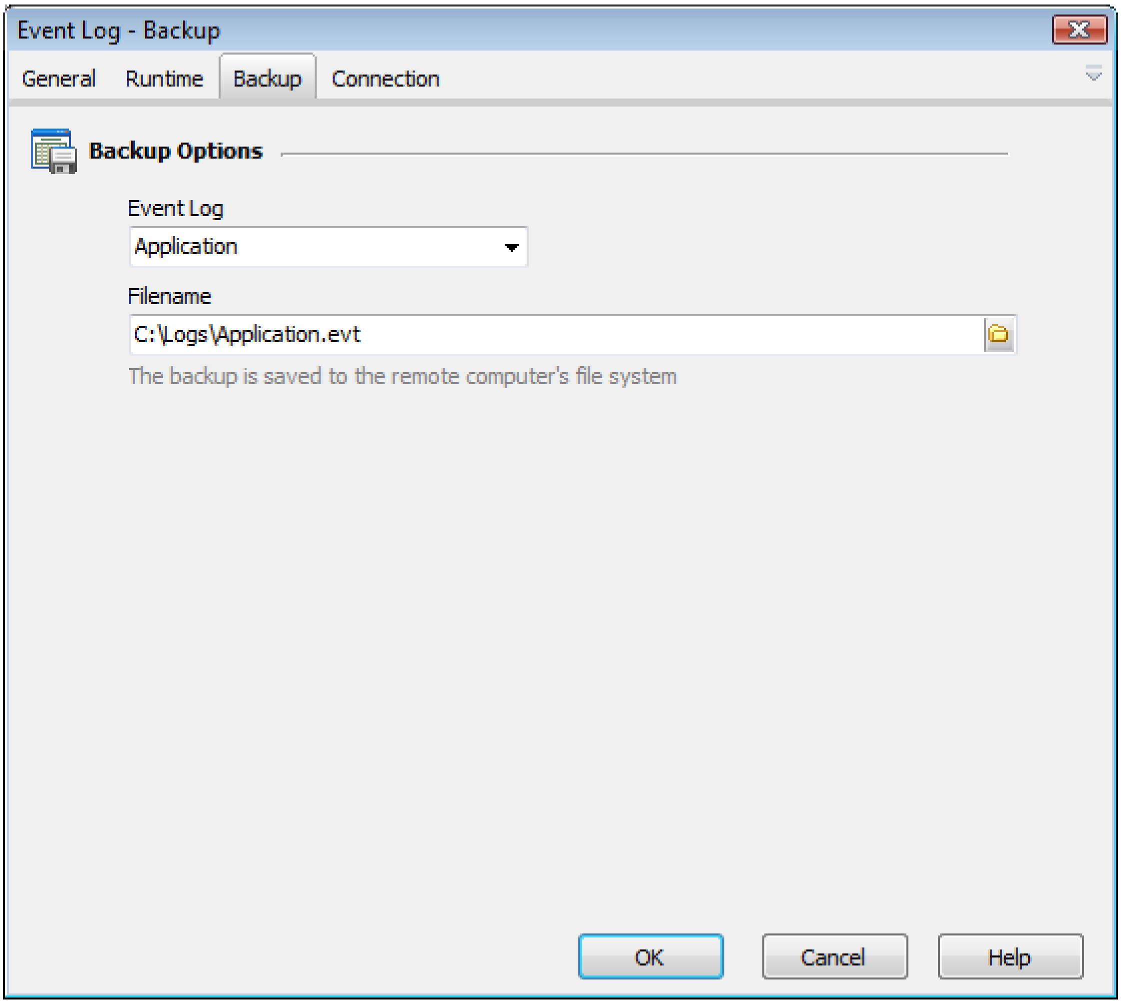
Remote Machine
The action will connect to a remote machine, using the specified user account.

Username
The username can be specified either by just the *username* or *DOMAIN/username*

Remote Hostname
The computer which to connect to, if the remote computer is in a different domain then that of the current user account, a Fully Qualified Domain Name must be specified. An IPv4/IPv6 may also be specified.

6.27.7.1 Event Log - Backup

The Event Log - Backup action enables to perform a backup of the specified event log.



Event Log: Application

Filename: C:\Logs\Application.evt

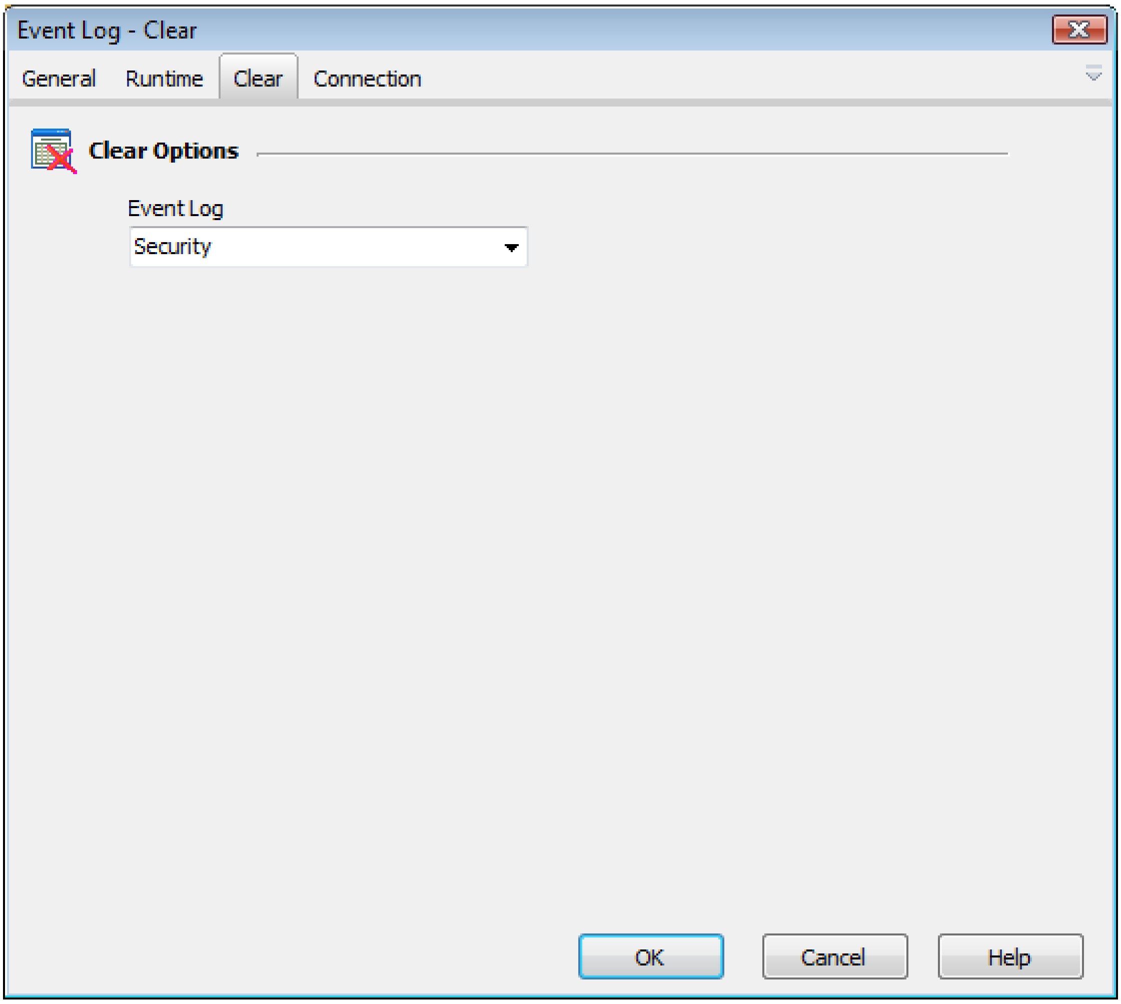
The backup is saved to the remote computer's file system

Event Log
The event log to be backed up. If the required event log does not appear in the list, it may be specified.

Filename
The full path to the location where the backup file will be stored. The file is stored on the remote computers file system, therefore the path must be valid on the remote computer.

6.27.7.2 Event Log - Clear

The Event Log - Clear action enables you to remove all log entries from the specified Event Log.

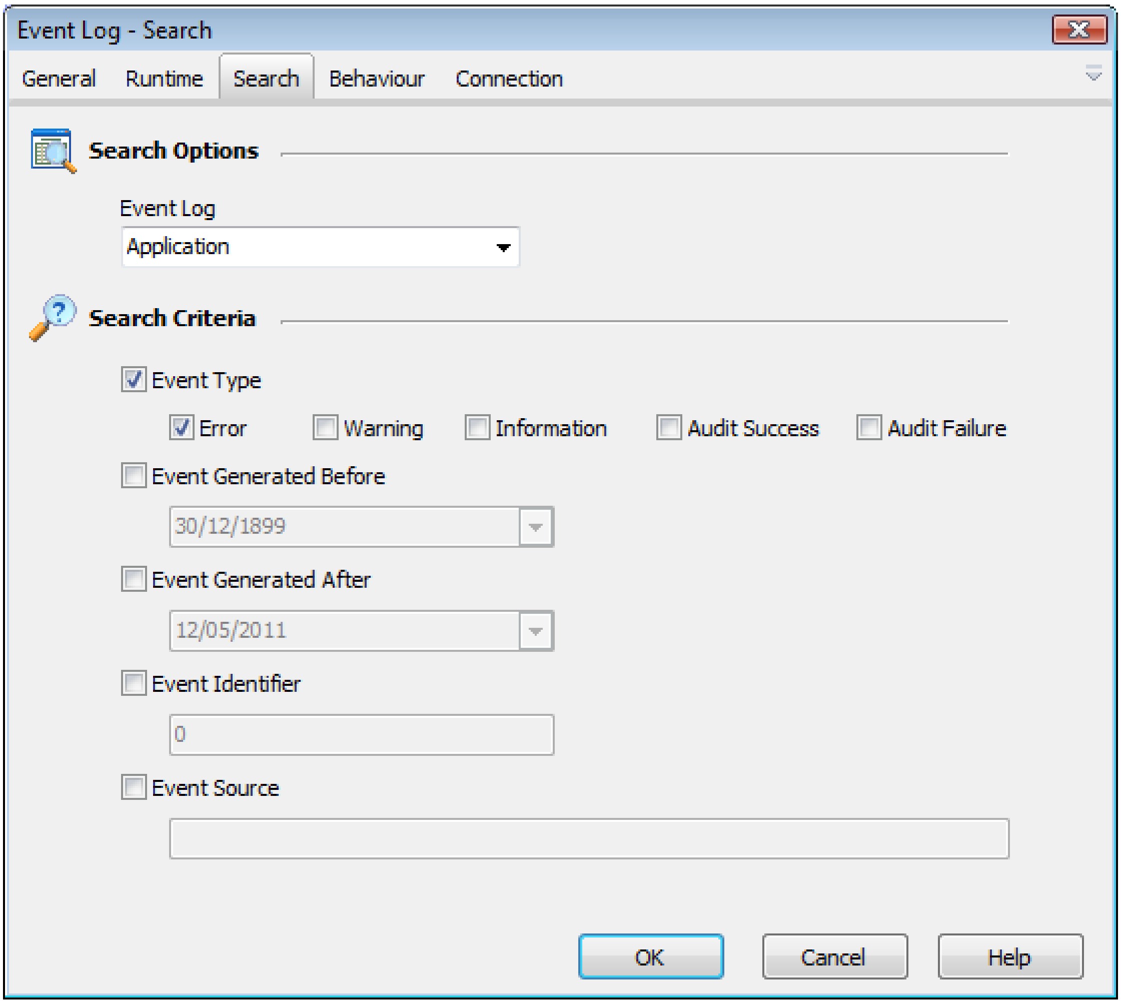


Event Log: Security

Event Log
The event log containing the log entries to be removed. If the required event log does not appear in the list, it may be specified.

6.27.7.3 Event Log - Search

The Event Log - Search action enables you to search the specified event log for entries matching the search criteria.



Search Options

Event Log
Application

Search Criteria

☑ Event Type

☑ Error ☐ Warning ☐ Information ☐ Audit Success ☐ Audit Failure

☐ Event Generated Before

30/12/1899

☐ Event Generated After

12/05/2011

☐ Event Identifier

0

☐ Event Source

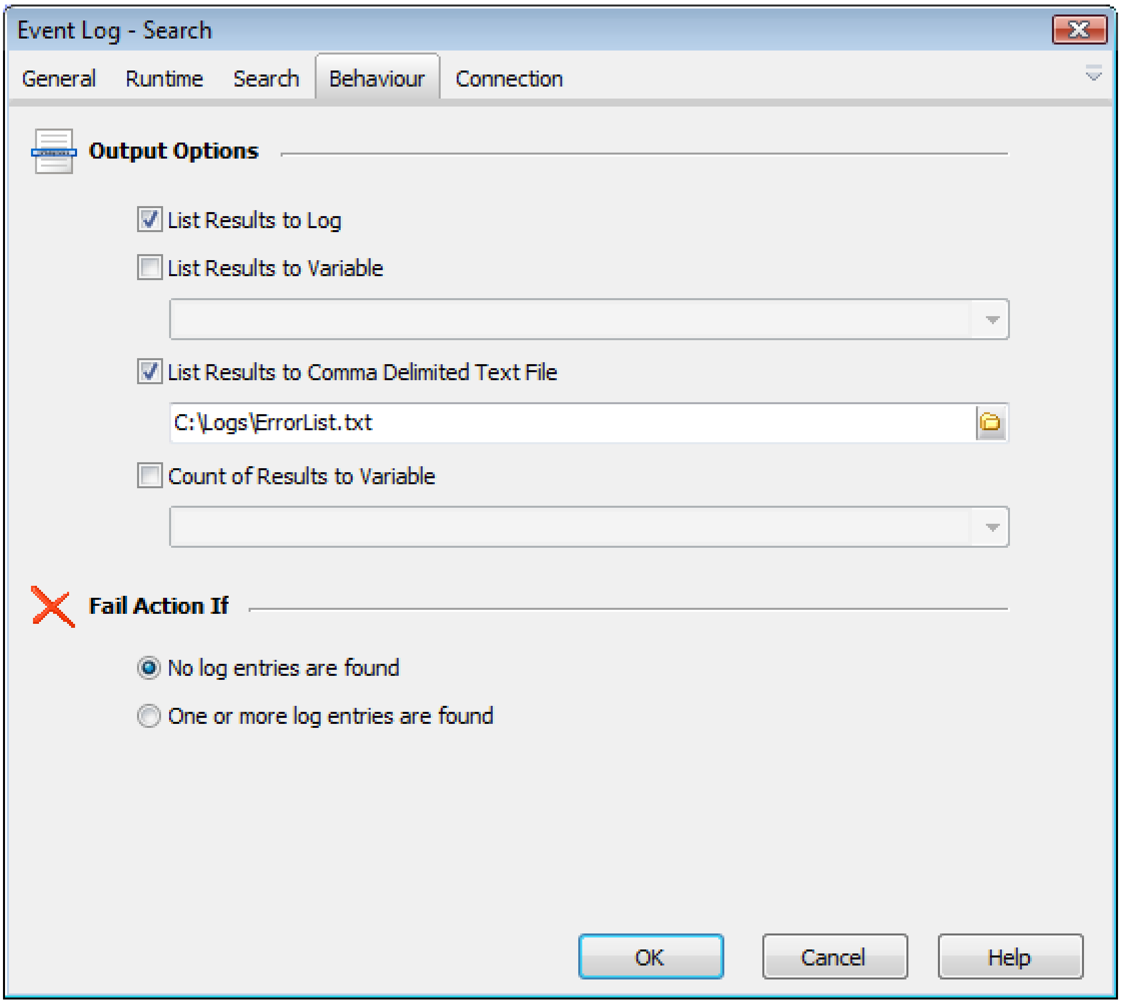
Event Log
The event log to be searched. If the required event log does not appear in the list, it may be specified.

Event Type
Find entries which are of the specified type. More then one type can be selected.

Event Generated Before/After
Find entries which were generated before or after the specified date.

Event Identifier
Find entries which match the specified Event Identifier.

Event Source
Find entries which match the specified source.



☑ List Results to Log

□ List Results to Variable

☑ List Results to Comma Delimited Text File

C:\Logs\ErrorList.txt

□ Count of Results to Variable

● No log entries are found

○ One or more log entries are found

List Results to Log
Enables the listing of the results in the log. Turning this option off is recommend when expecting a high number of results to be returned.

List Results to Variable
Saves the results of the search to the specified variable.

List Results to Comma Delimited Text File
Outputs the log to a text file with each field separated by a comma.

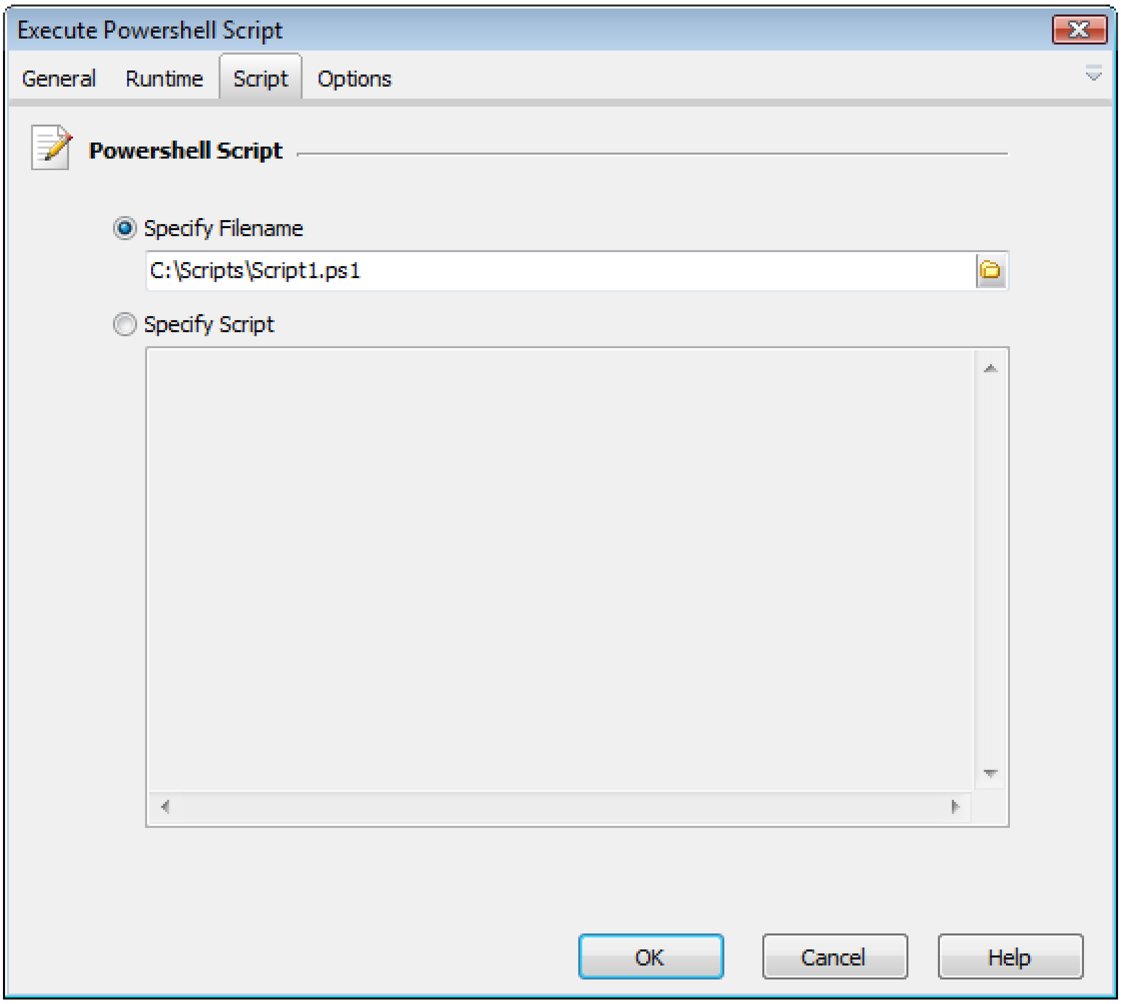
Count of Results to Variable
Writes to number of matched entries to the specified variable.

Fail Action If
Specifies whether the action will fail if there were entries found that matched the criteria or fail if no matches were found.

6.27.8 Execute Powershell Script

The Execute Powershell Script action enables you to execute a Windows Powershell script. The script can be defined in the action or in a script file.

For more information see: http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx



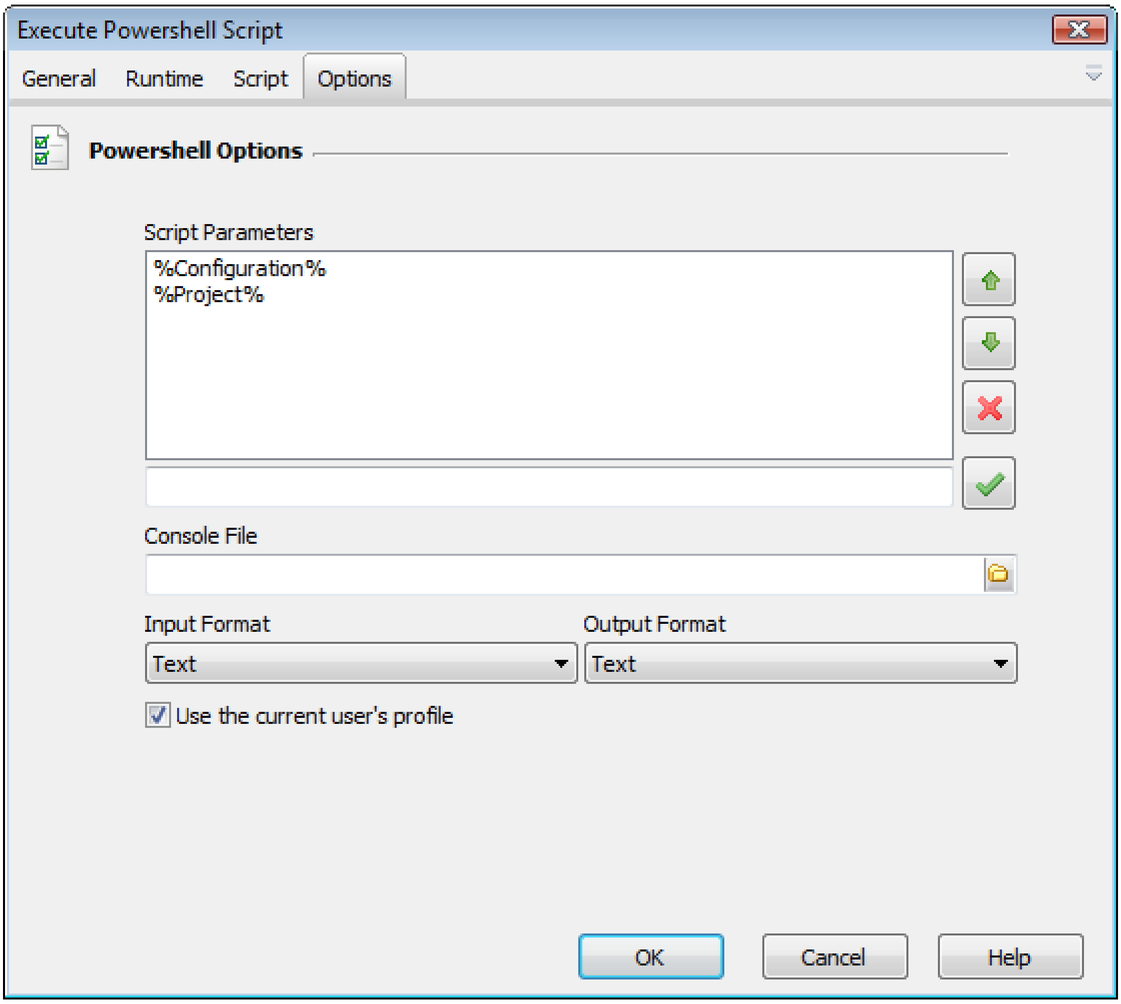
● Specify Filename

C:\Scripts\Script1.ps1

○ Specify Script

Filename
The full path to the script to be executed.

Script
The Powershell script block to be executed.



Script Parameters

%Configuration%
%Project%

Console File

Input Format: Text

Output Format: Text

☑ Use the current user's profile

Script Parameters
The arguments passed to the script. The arguments are passed into the script in the order that they appear in the list.

Console File
The console file to be loaded into the Powershell instance.

Input Format
The format of the data being sent to Windows Powershell.

Output Format
The format of the date being received from Windows Powershell.

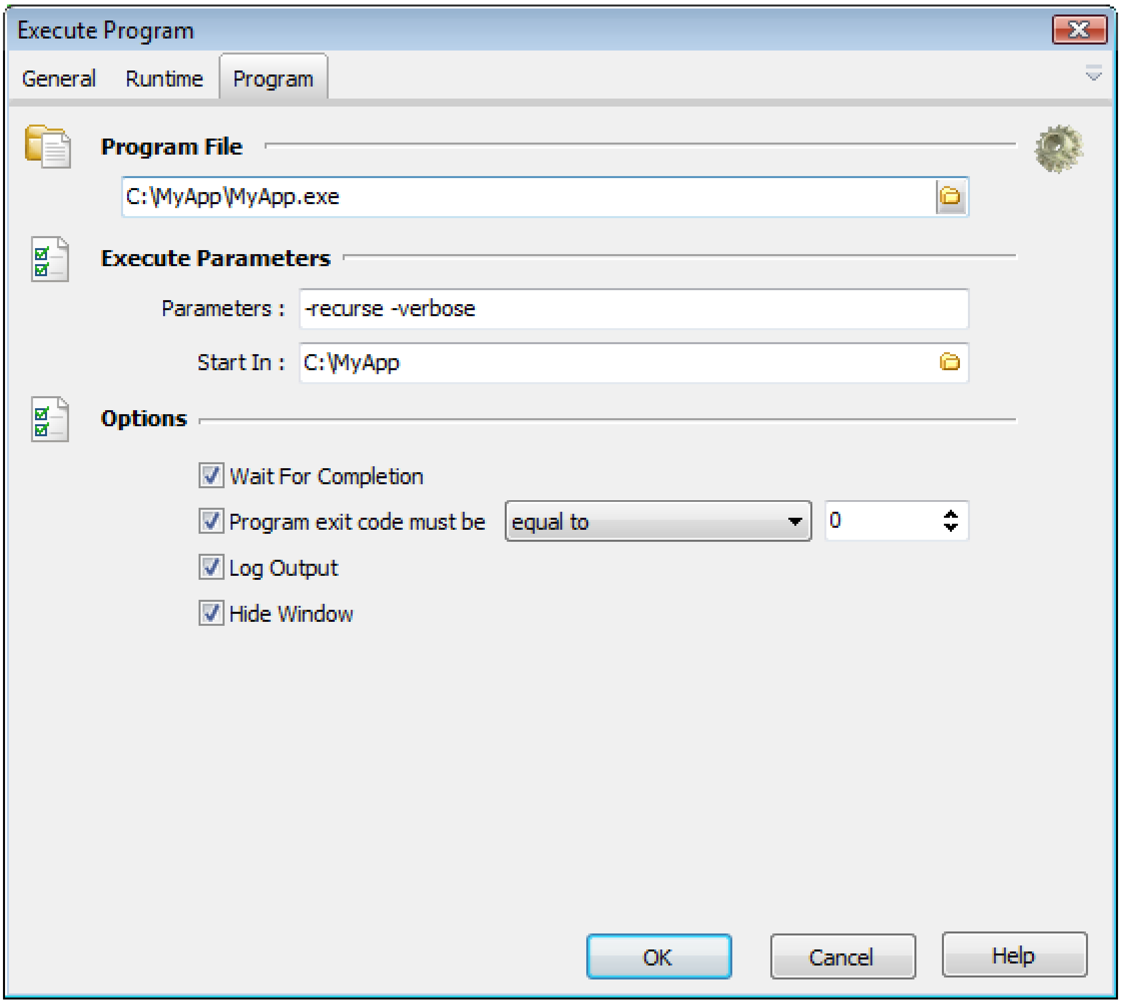
Use the current user's profile
Determines whether the user's profile will be loaded into the Powershell instance.

6.27.9 Execute Program Action

The Execute Program Action allows you to execute nearly any program from a Automise project.

Automise can capture the output of console applications (such as command line compilers) and display the captured output in the Automise output window.

Note: If you are executing a native DOS command (ie 'dir'), or a batch file, or you wish to redirect console output to a file, then use the Run DOS Command action instead.



C:\MyApp\MyApp.exe

Parameters : -recurse -verbose

Start In : C:\MyApp

☑ Wait For Completion

☑ Program exit code must be equal to 0

☑ Log Output

☑ Hide Window

Wait For Completion

By default, Automise will wait for the program to complete before continuing. However you can turn this option off. When Wait For Completion is turned off, the action will complete as soon as the program begins executing. When Wait For Completion is turned off, program output is not available for capture.

Hide Window

If this option is set, console programs will not run in a visible window.

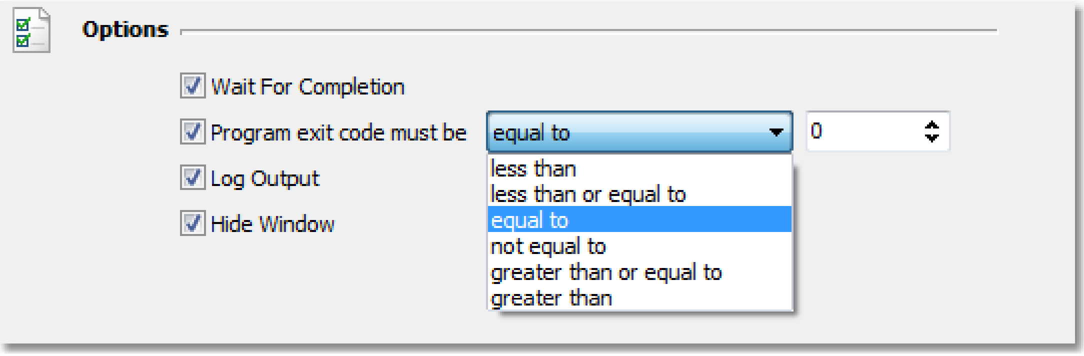
Log Output

If this option is set, Automise will send all console output to the Automise log. This option is not available if WaitForCompletion is set.

You may need to disable **Log Output** on some programs (very few) which do not flush their console output buffer correctly. These programs may appear to hang when executed inside Automise. Unfortunately, this is a problem within the third party programs themselves and it is not possible for Automise to correct the issue externally. If output needs to be captured, you can sometimes use the Run DOS Command action to execute the program and redirect the output to a file.

"Program exit code must be..."

If you enable this option, Automise will require the program's exit code to be within a certain range:



Options

☑ Wait For Completion

☑ Program exit code must be equal to 0

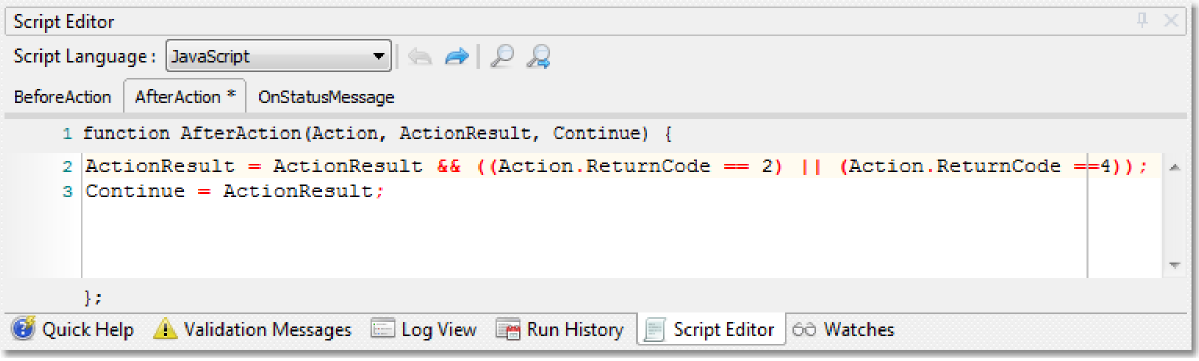
☑ Log Output

☑ Hide Window

If the exit code is outside the specified range, the action will fail.

Advanced Exit Code Checking

If you need more fine-grained control over the action's behaviour, you can read the Action. ReturnCode property from script. The following script fails the action if the return code is any value apart from 2 or 4:



```
function AfterAction(Action, ActionResult, Continue) {
ActionResult = ActionResult && ((Action.ReturnCode == 2) || (Action.ReturnCode ==4));
Continue = ActionResult;
};
```

Parsing The Output of a Program

Some programs do not provide accurate enough return codes to determine if an error has occurred. In these cases you may need to parse the actual program output for errors. You can do this by using script in the OnStatusMessage event handler (assuming you have enabled **Log Output**.)

Scripting Info

The Action properties available are :

property ProgramName : WideString;
property Params : WideString;
property StartInDir : WideString;
property LogOutput : WordBool;

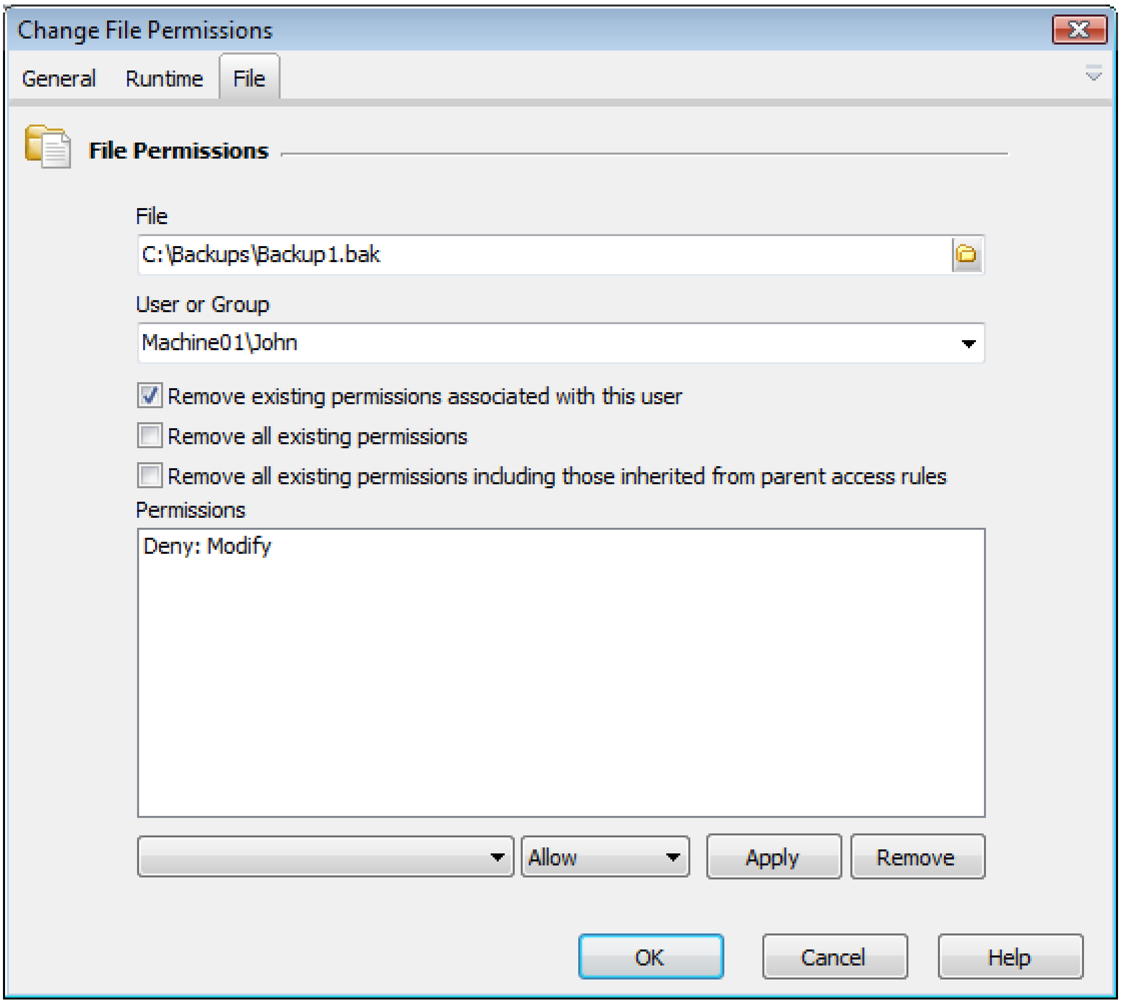
```
property WaitForCompletion : WordBool;
property ReturnCode : integer; // Read only
property HideWindow : WordBool;
property EnableReturnCodeCheck : Boolean;
property ReturnCodeComparator : TFBRunReturnCodeComparator;
property ReturnCodeToCheck : Boolean;
```

These properties may be set in the BeforeAction and AfterAction Script events.

6.27.10 File & Folder Security

6.27.10.1 Change File Permissions

The Change File Permissions action allows the access permissions of a file to be changed.



File: C:\Backups\Backup1.bak

User or Group: Machine01\John

☑ Remove existing permissions associated with this user

☐ Remove all existing permissions

☐ Remove all existing permissions including those inherited from parent access rules

Permissions: Deny: Modify

File
The full path to the file which the permissions are to be changed for.

User or Group
The user or group which the permissions will apply to.

Remove existing permissions associated with this user
Forces all existing permissions that are associated with the specified user to be removed, before the new permissions are added.

Remove all existing permissions
Removes all permissions from the file, except for permissions that are inherited from parent objects.

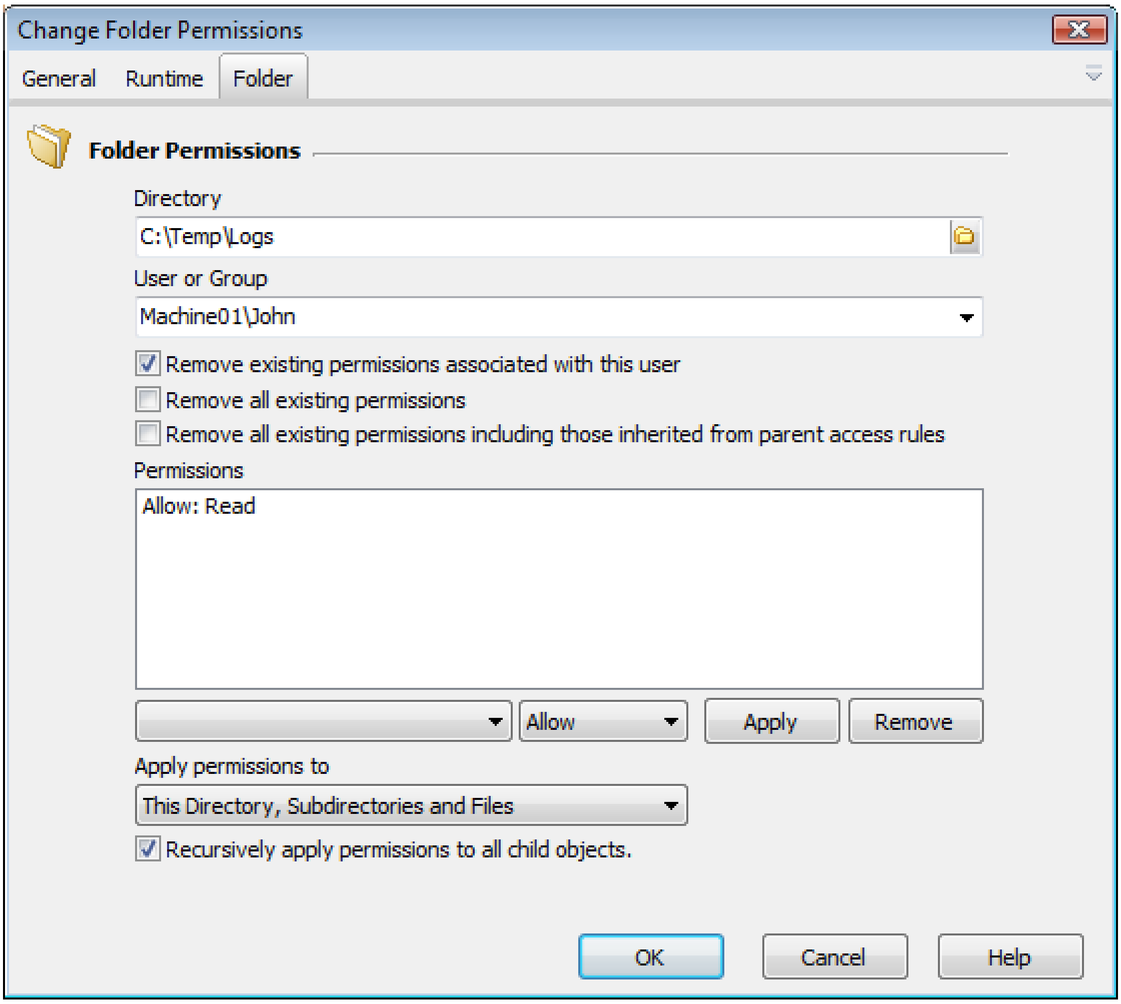
Permissions
The list of the permissions to be applied to the file. Each permission can be toggled between Allow or Deny, possible permissions are

- Full Control
- Modify
- Read And Execute
- Read
- Write

- Append Data
- Change Permissions
- Create Directories
- Create Files
- Delete
- Delete Subdirectories And Files
- Execute File
- List Directory
- Read Attributes
- Read Data
- Read Extended Attributes
- Read Permissions
- Synchronize
- Take Ownership
- Traverse
- Write Attributes
- Write Data
- Write Extended Attributes

6.27.10.2 Change Folder Permissions

The Change Folder Permissions action allows the access permissions of a folder to be changed.



Directory
C:\Temp\Logs

User or Group
Machine01\John

☑ Remove existing permissions associated with this user
□ Remove all existing permissions
□ Remove all existing permissions including those inherited from parent access rules

Permissions
Allow: Read

Apply permissions to
This Directory, Subdirectories and Files

☑ Recursively apply permissions to all child objects.

Directory
The full path to the directory which the permissions are to be applied to.

User or Group
The user or group which the permissions will apply to.

Remove existing permissions associated with this user
Forces all existing permissions that are associated with the specified user to be removed, before the new permissions are added.

Remove all existing permissions
Removes all permissions from the directory, excluding for permissions that are inherited from parent objects.

Permissions
The list of the permissions to be applied to the file. Each permission can be toggled between Allow or Deny, possible permissions are

- Full Control
- Modify
- Read And Execute
- Read
- Write

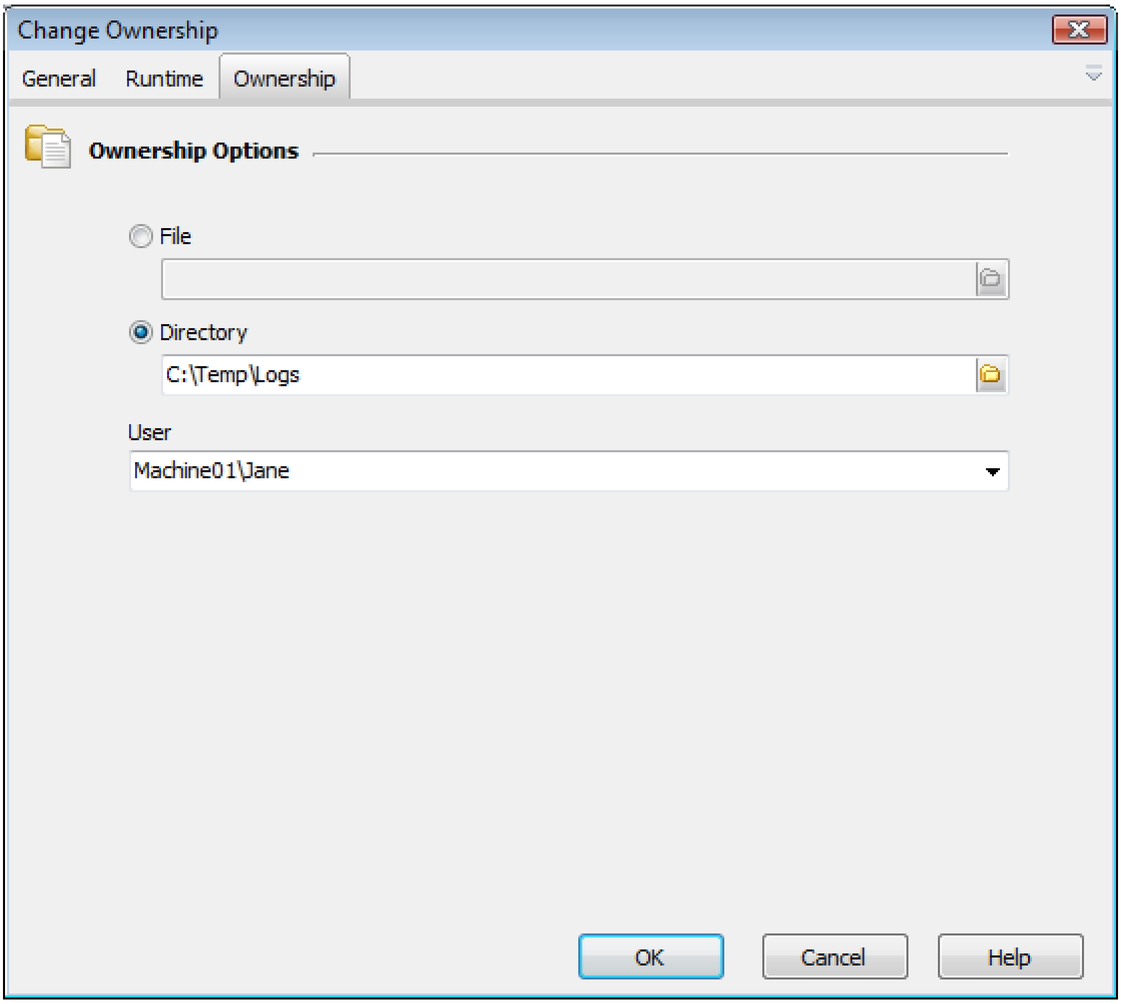
- Append Data
- Change Permissions
- Create Directories
- Create Files
- Delete
- Delete Subdirectories And Files
- Execute File
- List Directory
- Read Attributes
- Read Data
- Read Extended Attributes
- Read Permissions
- Synchronize
- Take Ownership
- Traverse
- Write Attributes
- Write Data
- Write Extended Attributes

Apply permissions to
Determines how the permissions will be inherited by child objects. The possible options are

- This Directory Only
- This Directory and Subdirectories
- This Directory and Files
- This Directory, Subdirectories and Files
- Subdirectories Only
- Subdirectories and Files
- Files Only

6.27.10.3 Change Ownership

The Change Ownership action allows the owner of a file or folder to be changed.



○ File

● Directory

C:\Temp\Logs

User

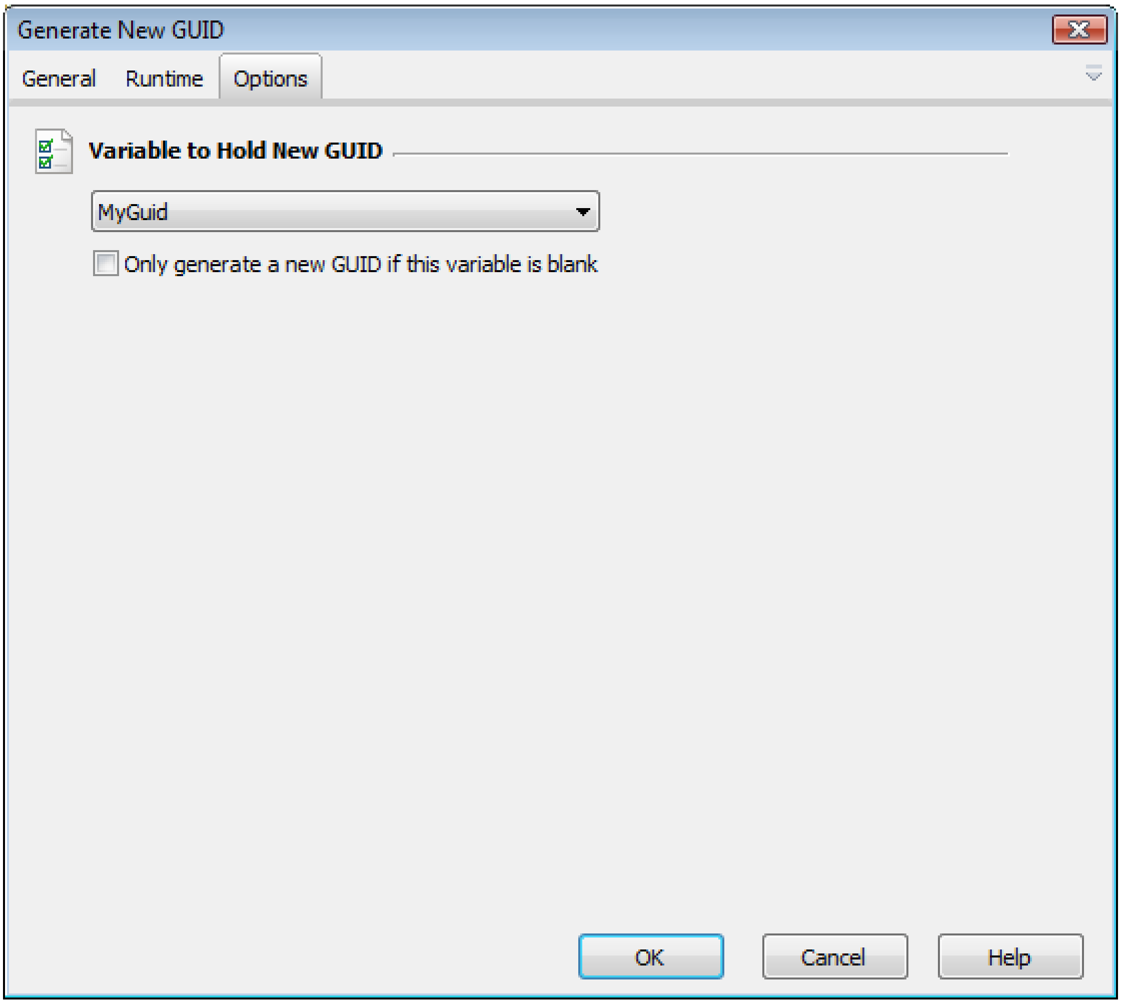
Machine01\Jane

File or Directory
The full path to the file or folder which the owner is to be changed.

User
The user which will take ownership of the specified file or folder. A user can only take ownership if they have the correct security permissions to do so. If they do not have the required permissions, the action will fail.

6.27.11 Generate New GUID

The Generate New GUID action generates a new GUID string and assigns it to a Automise variable.



Variable to Hold New GUID

MyGuid

□ Only generate a new GUID if this variable is blank

Variable to Hold New GUID

Enter the name of a variable to hold the new GUID value.

Only generate a new GUID if this variable is blank

If this option is selected and the chosen variable has a non-blank value when the action is run, a new GUID will not be generated and the variable value will be left as-is.

Script Events

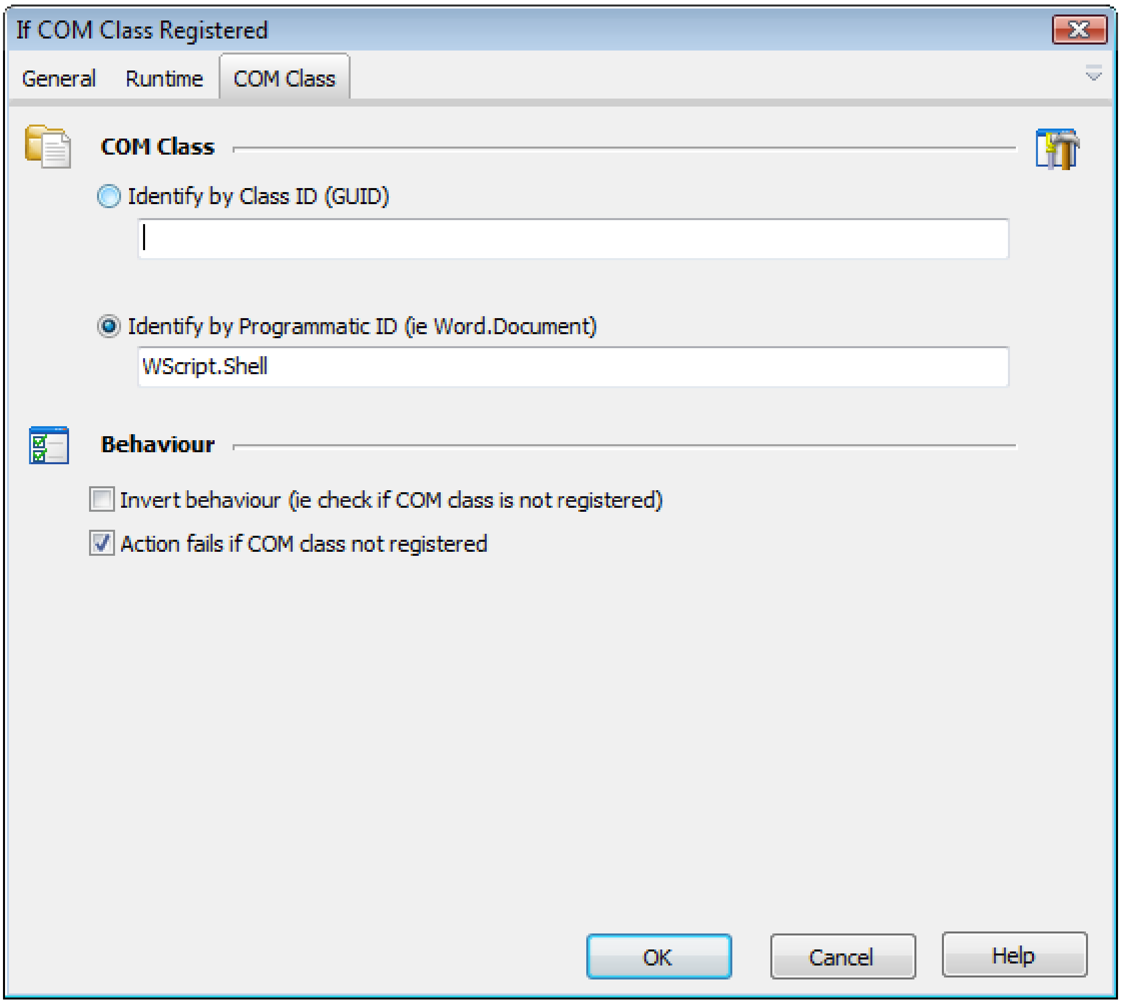
The OnGenerateGUID script event allows you to do some processing when the new GUID is generated. The GUID parameter holds the new GUID value (as a string.)

See also : NewGUIDString() script function.

6.27.12 If COM Class Registered

The If COM Class Registered action enables you to control the flow of your project depending on whether a COM Class is registered. If the class is registered, then the child actions of the action will be executed.

An Else Action can be used to provide a list of actions to execute if the COM class is not registered. Alternatively, enable the Invert behaviour option in the action.



If COM Class Registered

COM Class

○ Identify by Class ID (GUID)

● Identify by Programmatic ID (ie Word.Document)

WScript.Shell

Behaviour

□ Invert behaviour (ie check if COM class is not registered)

☑ Action fails if COM class not registered

Identify by Class ID (GUID)

Specify a GUID in the form {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}. A warning will be displayed if the GUID you have entered does not seem to be syntactically valid.

Identify by Programmatic ID

Programmatic IDs (ProgIDs) are associated with GUIDs in the registry and provide a much easier way to specify a COM Class.

Invert behaviour

If this box is checked, the behavior of the If COM Class Registered action is inverted: ie the child actions will be executed if the COM Class is not registered.

Action fails if COM class not registered

If this box is checked, the action will fail outright if the COM class is not registered. If the

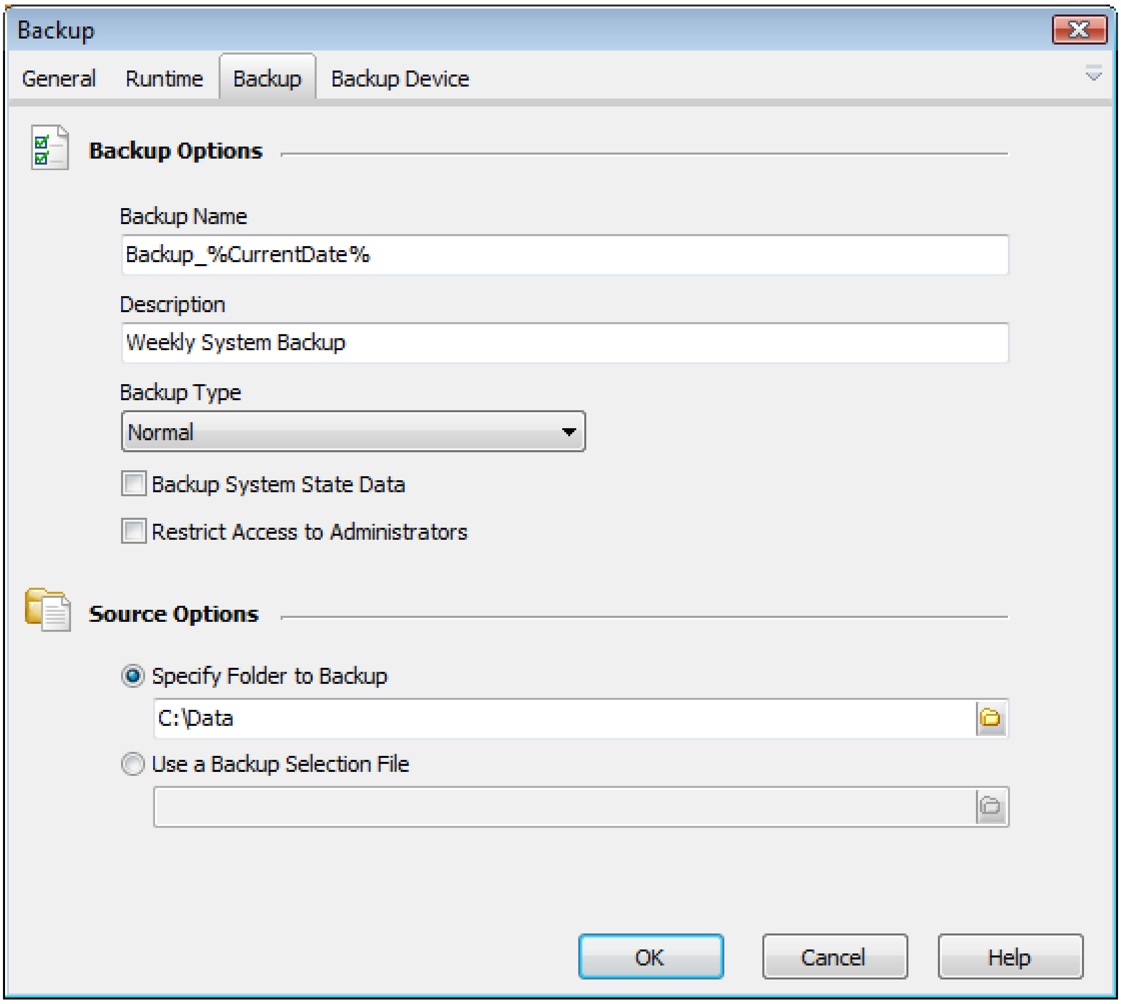
'Invert behaviour' box is checked, this option becomes 'Action fails if COM class *is* registered.'

6.27.13 Logical and Network Drives

6.27.13.1 Backup

The Backup action enables you to perform a system backup, with the ability to write the backup to disk or tape.

Note that this action will not work on Windows Vista/Server 2008 onwards. To perform a backup on these operating systems please use the WBAdmin actions.



Backup

General | Runtime | Backup | Backup Device

Backup Options

Backup Name: Backup_%CurrentDate%

Description: Weekly System Backup

Backup Type: Normal

□ Backup System State Data

□ Restrict Access to Administrators

Source Options

● Specify Folder to Backup: C:\Data

○ Use a Backup Selection File

Backup Name
The name of the backup to be used in the report. Typically the name reflects the items being backed up.

Description
The label of the backup set.

Backup Type
The type of backup to perform, types include:

Normal: Copies all selected files and marks each file as having been backed up.

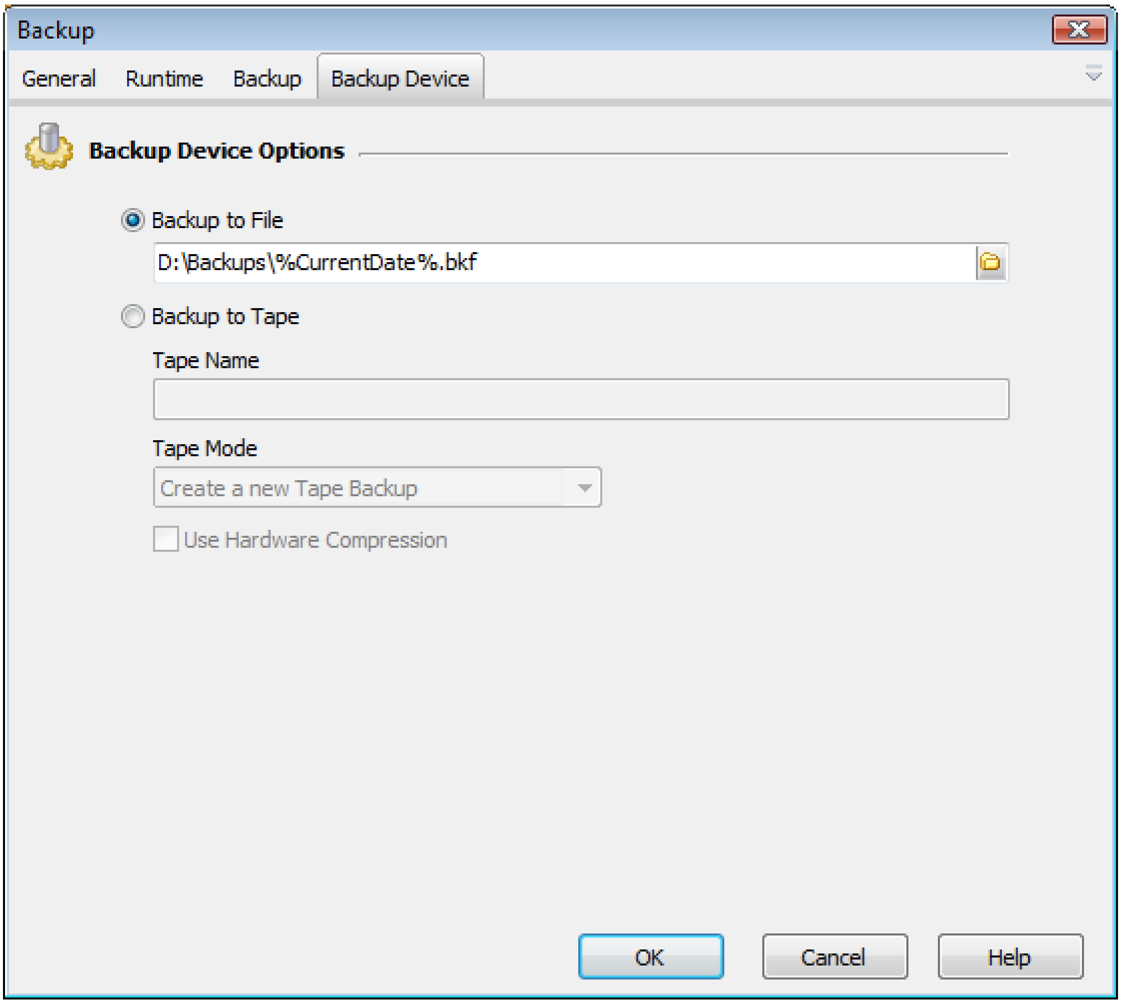
Copy: Copies all selected files but does not mark the files as being backed up.
Differential: Copies all files that have been created or changed since the last normal or incremental backup, but does not mark the files as being backed up.
Incremental: Copies all files that have been created or changed since the last normal or incremental backup, marking each file as having been backed up.
Daily: Copies all files that have been modified the day which the backup is performed, but does not mark the files as being backed up.

Backup System State Data
Backs up the system components relevant the computer system, components can include the Registry, COM+ Class Registration Database, Boot Files, Certificate Services Database, ActiveDirectory Directory Service, SYSVOL Directory, Cluster Service Information, IIS MetaDirectory and System Files protected by Windows File Protection.

Restrict Access to Administrators
Access to the tape is restricted to users in the administrator group.

Specify Folder to Backup
The folder which to perform the backup on.

Use a Backup Selection File
A file created by the NTBackup GUI which contains a list of files to be backed up.


Backup

General | Runtime | Backup | Backup Device

Backup Device Options

● Backup to File
D:\Backups\%CurrentDate%.bkf

○ Backup to Tape
Tape Name

Tape Mode
Create a new Tape Backup

□ Use Hardware Compression

Backup to File

The file to backup the specified files and folders to.

Backup to Tape

Tape Name
The name of the tape to backup the specified files and folders to.

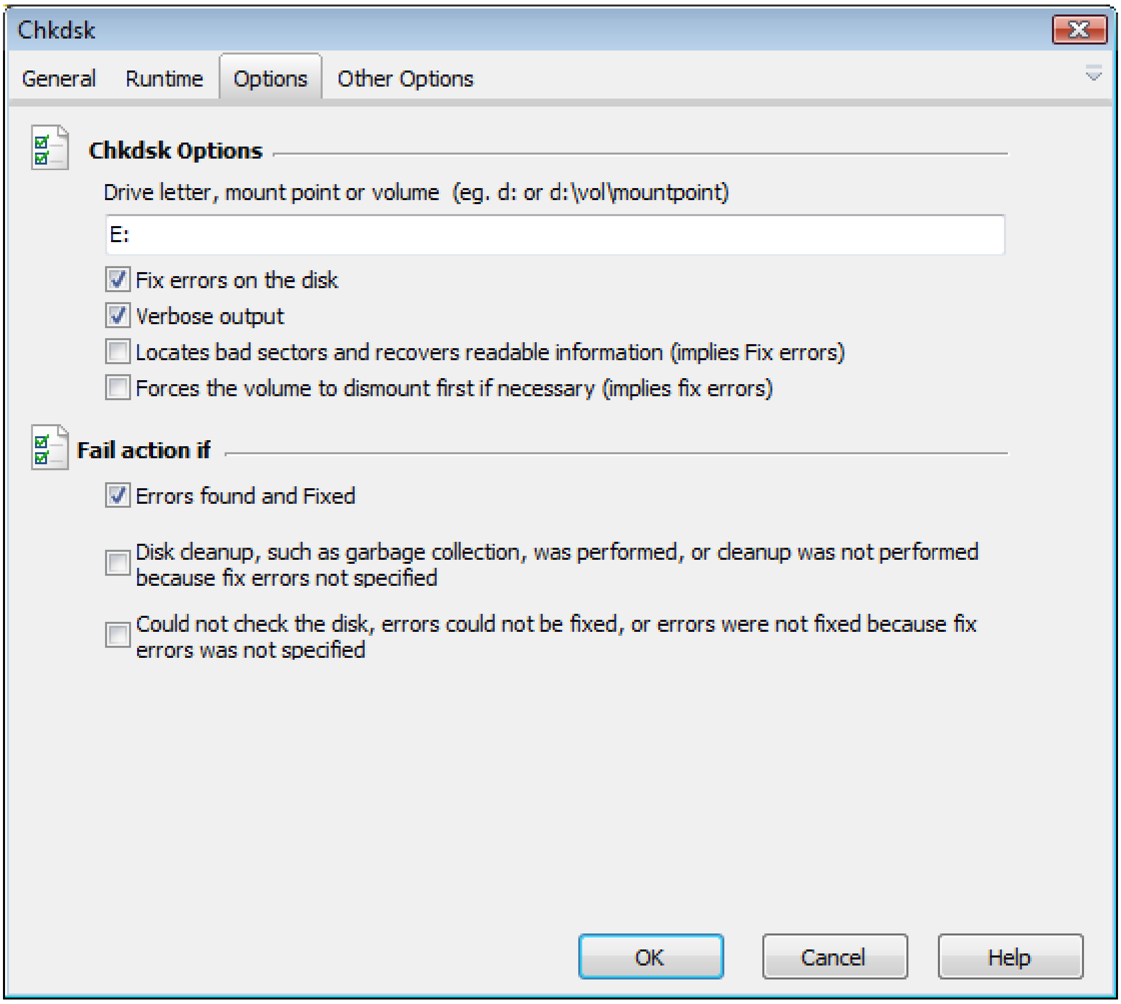
Tape Mode
Changes the behaviour of the way the backup is written to the tape, possible modes are:

Create a new Tape Backup: Use this option when this is the first backup performed.
Append to Existing Tape Backup: Use this option to append the backed backup to the tape.
Overwrite an Existing Tape Backup: Use this option to overwrite any existing backups on the tape.

Use Hardware Compression
Forces the tape drive to use hardware compression if it is supported.

6.27.13.2 Chkdsk Action

The Chkdsk Action runs the chkdsk.exe command line tool to check the integrity of your filesystem and optionally fix errors.



Chkdsk Options

Drive letter, mount point or volume (eg. d: or d:\vol\mountpoint)

E:

☑ Fix errors on the disk
☑ Verbose output
□ Locates bad sectors and recovers readable information (implies Fix errors)
□ Forces the volume to dismount first if necessary (implies fix errors)

Fail action if

☑ Errors found and Fixed
□ Disk cleanup, such as garbage collection, was performed, or cleanup was not performed because fix errors not specified
□ Could not check the disk, errors could not be fixed, or errors were not fixed because fix errors was not specified

Volume - Specify the drive letter (followed by a colon), mount point, or volume name.

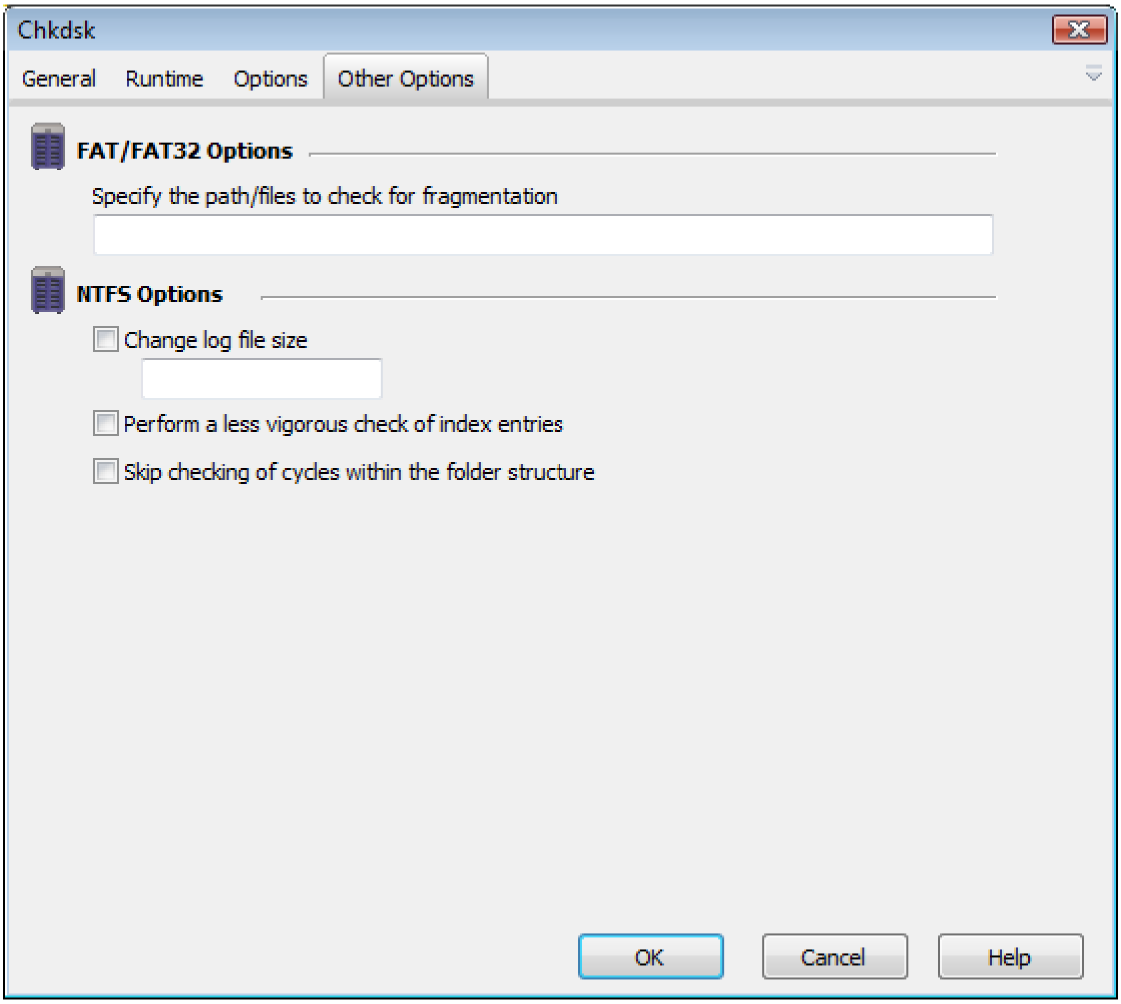
Fix errors on the disk - The disk must be locked. If chkdsk cannot lock the drive, a message appears that asks you if you want to check the drive the next time you restart the computer.

Verbose Output - Displays the name of each file in every directory as the disk is checked and displays cleanup messages

Locates bad sectors and recovers readable information - The Fix errors option is implied. The disk must be locked.

Forces the volume to dismount first, if necessary - All open handles to the drive are invalidated. The Fix errors option is implied.

Fail Options - specify which error conditions reported by chkdsk should fail the action



FAT/FAT32 Options

Specify the path/files to check for fragmentation

NTFS Options

□ Change log file size

□ Perform a less vigorous check of index entries

□ Skip checking of cycles within the folder structure

FileName
Use with file allocation table (FAT) and FAT32 only. Specifies the location and name of a file or set of files that you want chkdsk to check for fragmentation. You can use wildcard characters (that is, * and ?) to specify multiple files.

NTFS Options
Performs a less vigorous check of index entries, reducing the amount of time needed to run chkdsk.
Skips the checking of cycles within the folder structure, reducing the amount of time

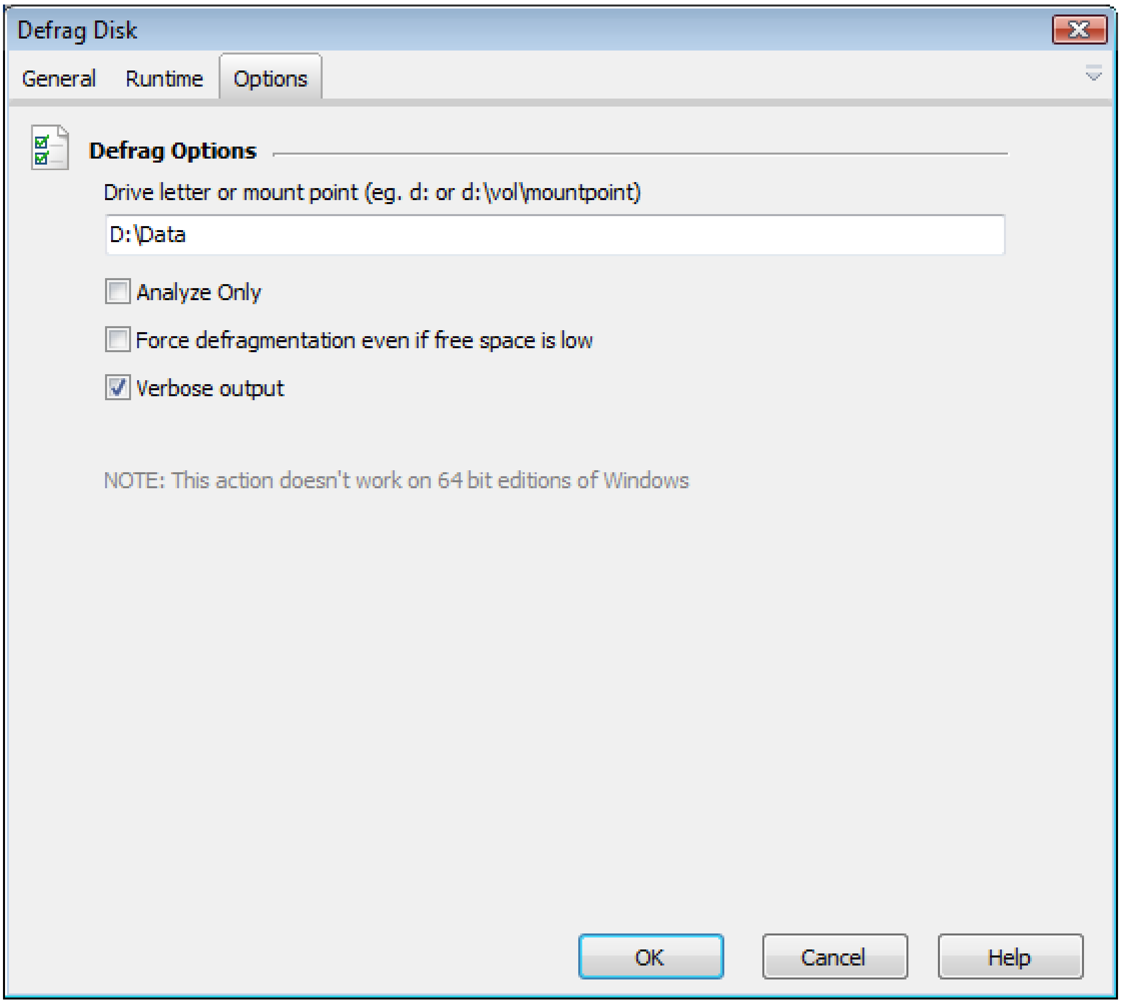
needed to run chkdsk.
Changes the log file size to the size you type.

More detailed information on chkdsk can be found on Microsoft TechNet here:
http://technet2.microsoft.com/WindowsServer/en/library/552ed70a-208d-48c4-8da8-2e27b530eac71033.mspx

6.27.13.3 Defrag Disk

The Defrag Disk action performs an analysis or defragmentation of the specified drive.

Use the WMI Defrag action if you want to Analyze and Defragment drives on Vista and Server 2003



Drive letter or mount point (eg. d: or d:\vol\mountpoint)
D:\Data

☐ Analyze Only

☐ Force defragmentation even if free space is low

☑ Verbose output

NOTE: This action doesn't work on 64 bit editions of Windows

Drive Letter - specify the drive letter or a mount point of the volume to be defragmented

Analyze Only - analyzes the volume, displays a summary of the analysis report, and indicates whether you should defragment the volume

Force - forces defragmentation of the volume when free space is low

Verbose - displays the complete analysis and defragmentation reports. When used in combination with Analyze Only, displays only the analysis report. When used alone, displays both the analysis and defragmentation reports

NOTES:

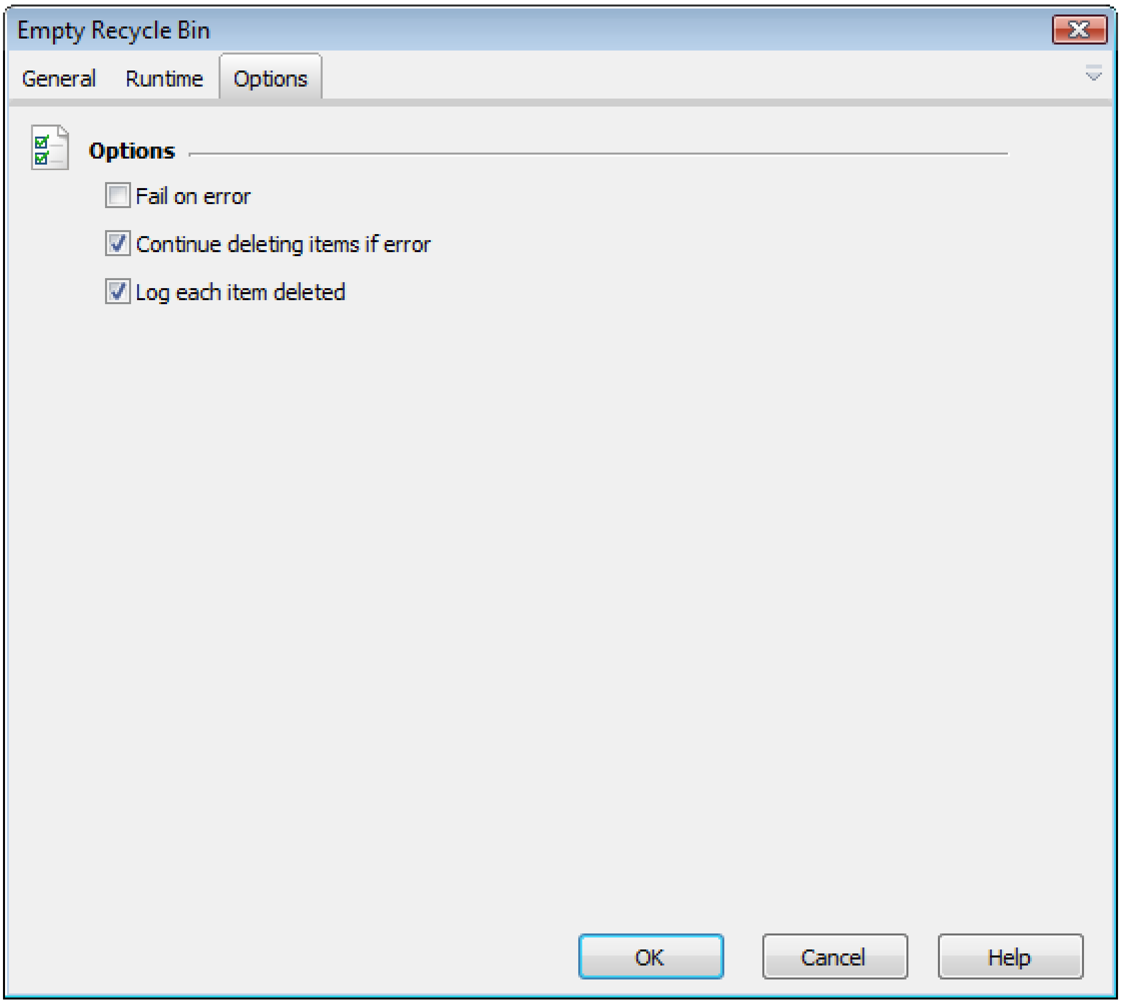
This action is known not to work on 64bit editions of Windows. The defrag.exe utility appears to exist in Explorer, and running "defrag" on the command line works, but if a process tries to run "defrag.exe" via the CreateProcess Windows API it will return in an error condition because the file can't be found.

A volume must have at least 15% free space for defrag to completely and adequately defragment it. Defrag uses this space as a sorting area for file fragments. If a volume has less than 15% free space, defrag will only partially defragment it. To increase the free space on a volume, delete unneeded files or move them to another disk.

You cannot defragment volumes that the file system has marked as dirty, which indicates possible corruption. You must run chkdsk on a dirty volume before you can defragment it. You can determine if a volume is dirty by using the fsutil dirty query command.

6.27.13.4 Empty Recycle Bin

The Empty Recycle Bin action will remove all items from the Recycle Bin of the active user.



Options

□ Fail on error

☑ Continue deleting items if error

☑ Log each item deleted

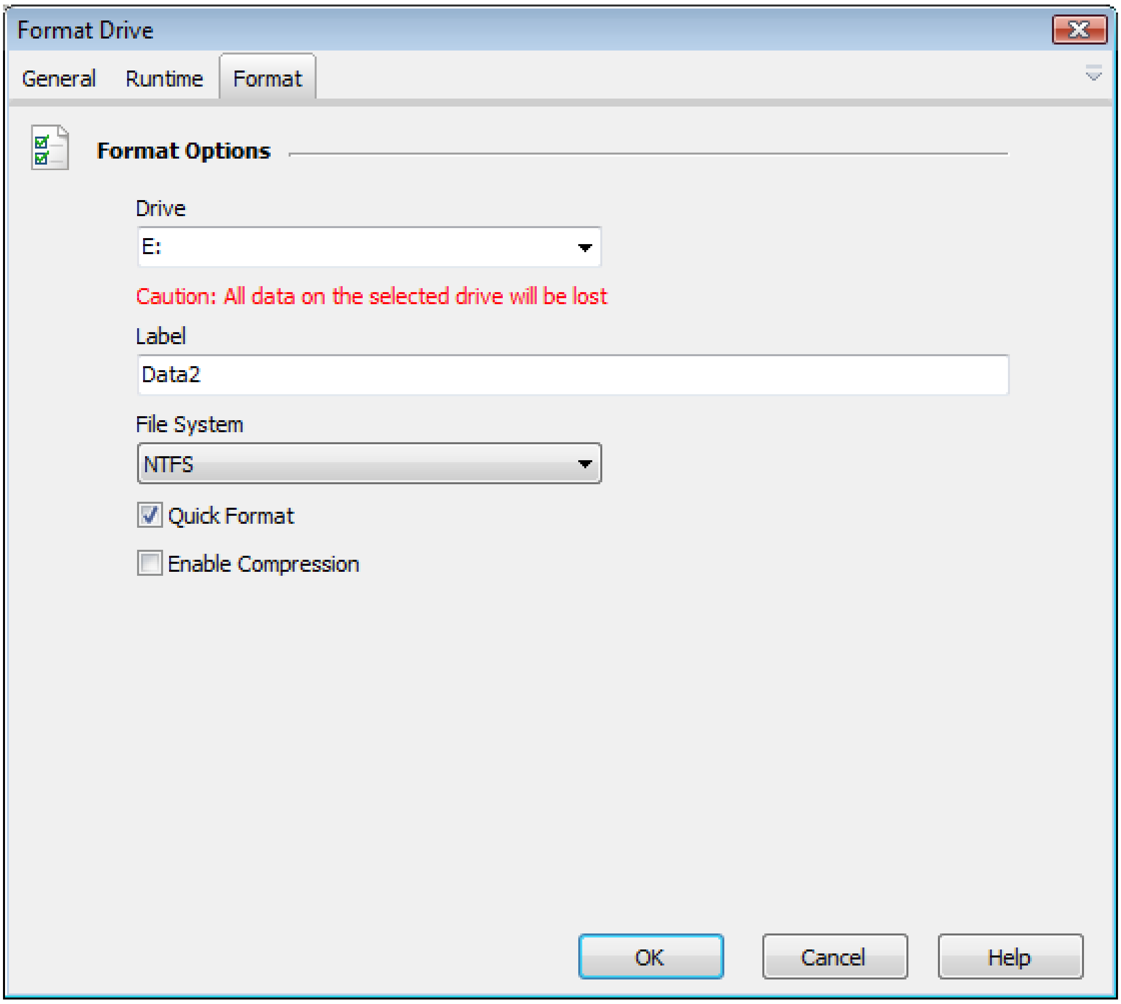
Log each item deleted - writes the name of the file or folder deleted to the log

There is a chance that the action will fail to delete an item in the Recycle Bin. You can choose if the action will continue deleting items if an error is encountered (**Continue deleting items if error**) and also if you want the action to fail if there was any errors (**Fail on error**).

You may also want to check if the recycle bin is empty by using the Is Recycle Bin Empty action.

6.27.13.5 Format Drive

The format drive action allows you to format a local drive.



Drive
E:

Caution: All data on the selected drive will be lost

Label
Data2

File System
NTFS

☑ Quick Format

□ Enable Compression

Drive
The disk to be formatted. The disk can be specified by its mount point, volume name or drive letter.

Label
The volume label of the disk.

File System
The file system which the disk will be formatted as. Possible file systems are Fat, Fat32 and NTFS.

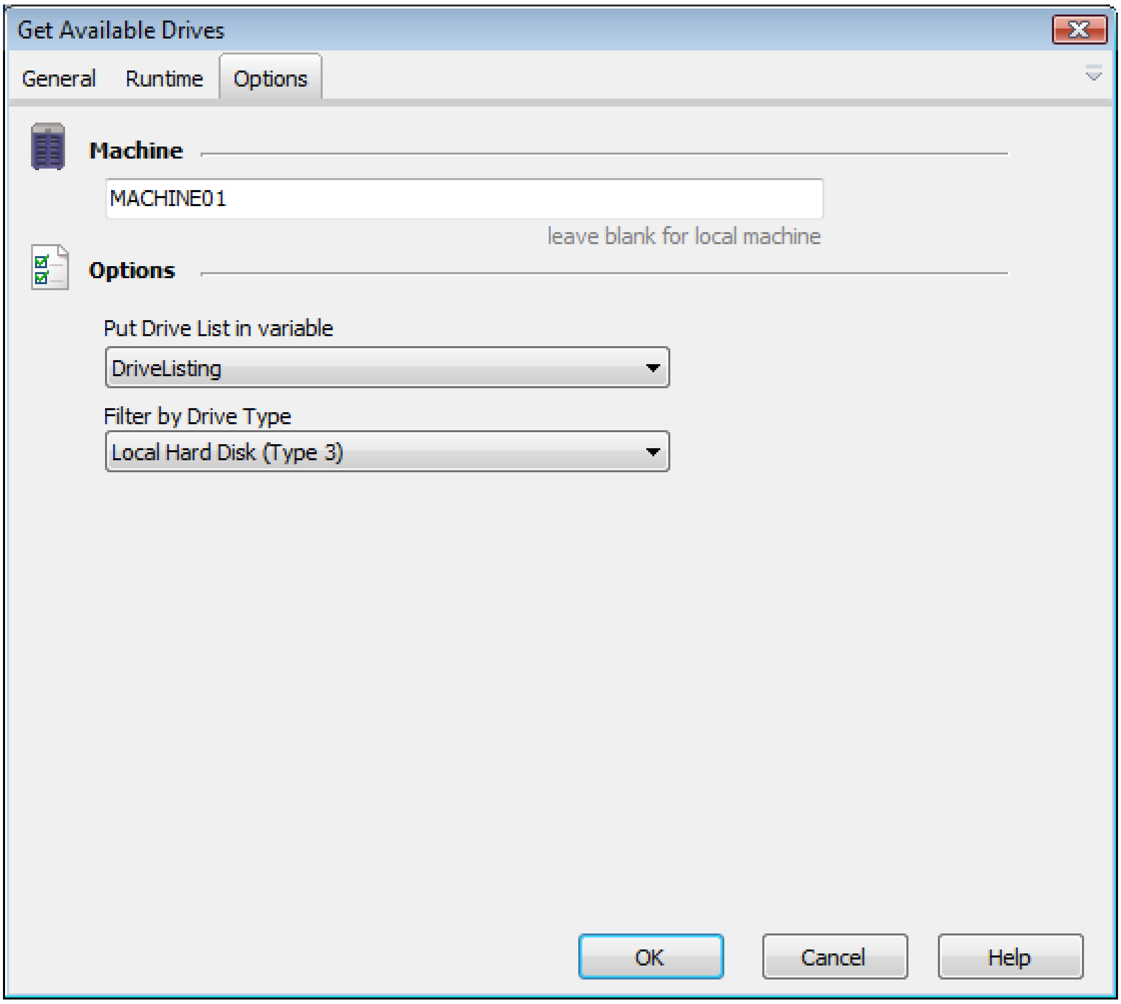
Quick Format
Performs a quick format of the disk by not checking for bad sectors.

Enable Compression

All files created on the disk will be compressed by default. This option is only available when the file system is NTFS.

6.27.13.6 Get Available Drives

The Get Available Drives action retrieves the list of drive letters which are present on the specified machine.



Machine

MACHINE01

leave blank for local machine

Options

Put Drive List in variable: DriveListing

Filter by Drive Type: Local Hard Disk (Type 3)

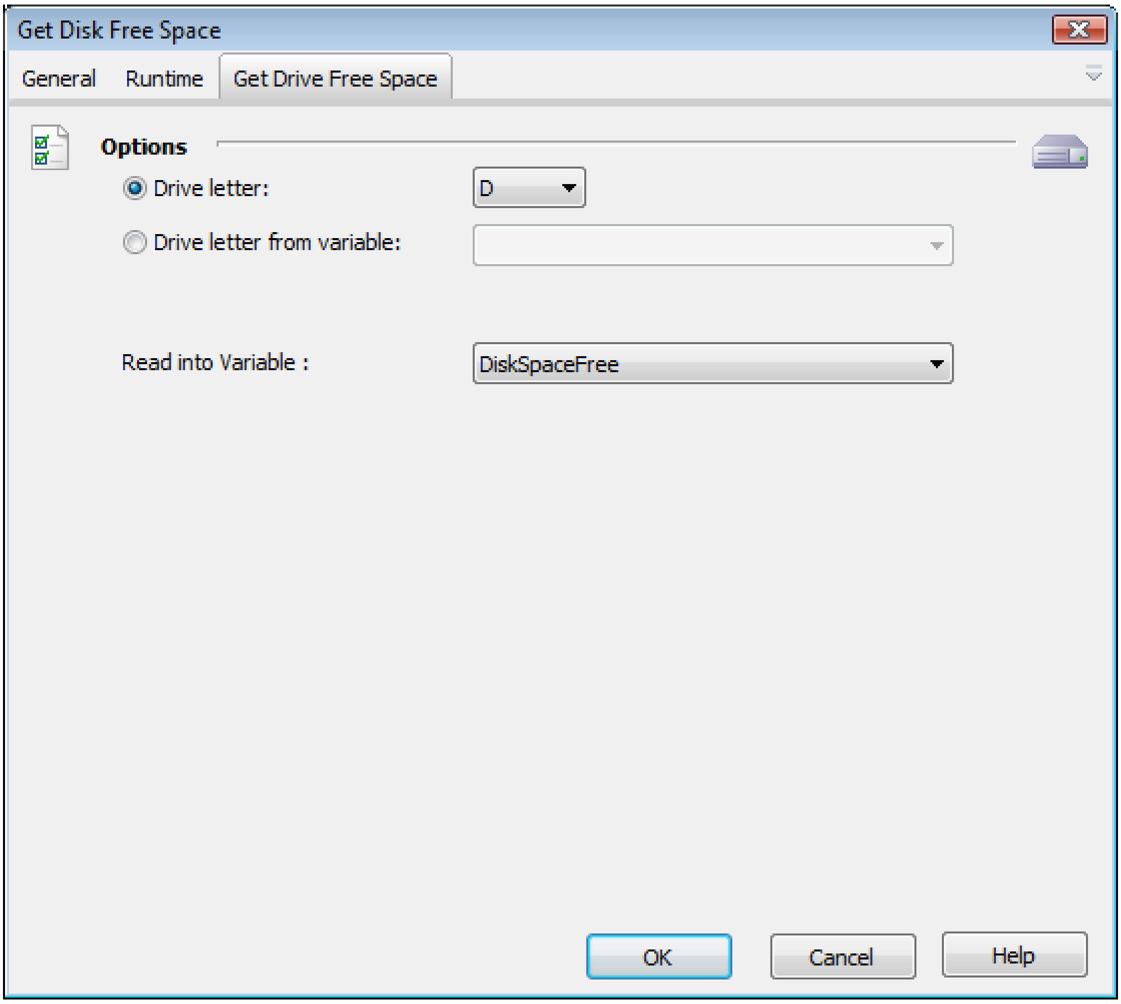
Machine Name - enter the machine name for a remote machine, or leave blank for the local machine

Put Drive List in variable - the list of drives present on the target machine will be put in this variable, one per line

Filter by Drive Type - only retrieve drives matching the chosen filter

6.27.13.7 Get Disk Free Space

Reads the free disk space of a drive into a Automise variable.



● Drive letter: D

○ Drive letter from variable:

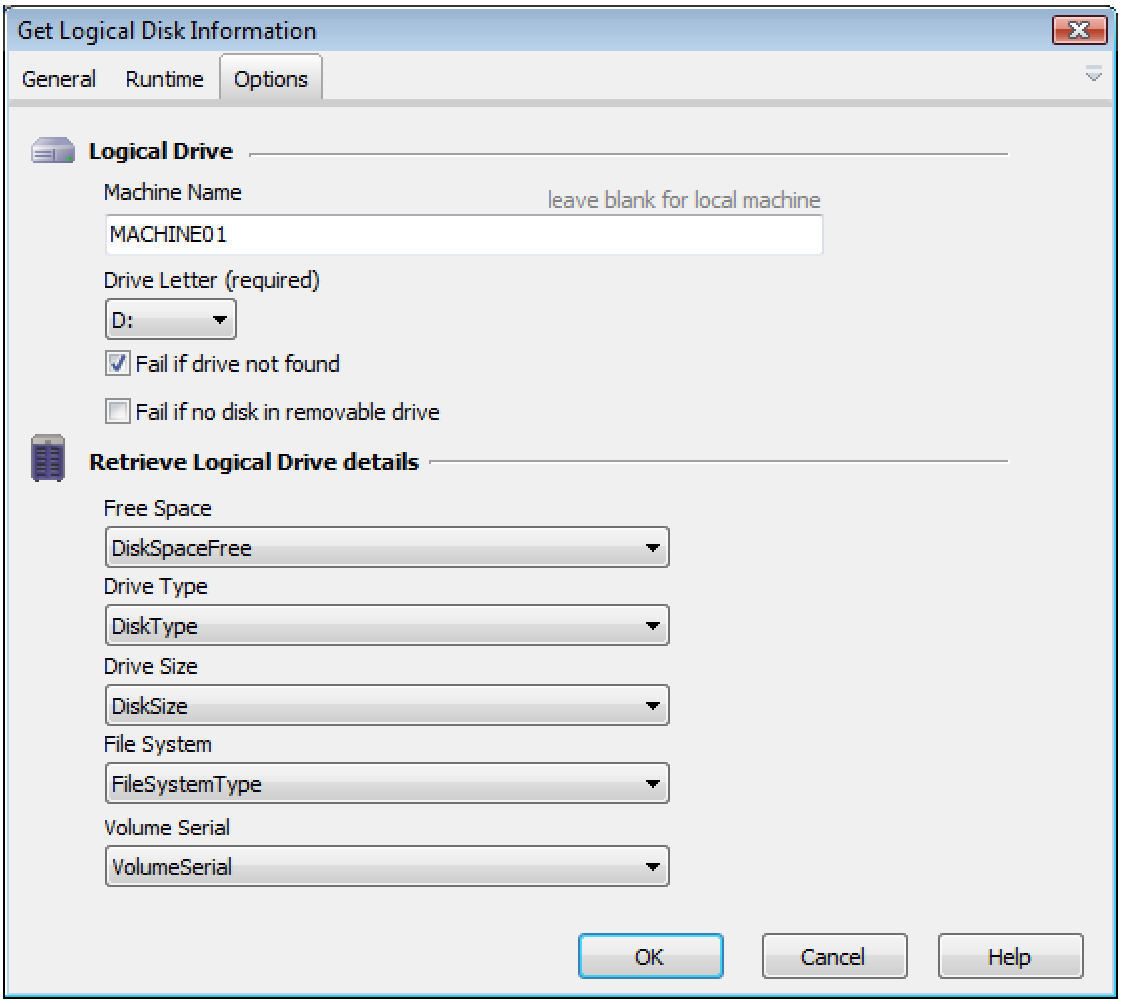
Read into Variable : DiskSpaceFree

The drive letter can be fixed, or loaded from a variable value at runtime. If loaded from a variable, the variable value must be either a single letter (ie "D"), or a single letter plus a colon (ie "D:").

The disk space available on the drive is returned in bytes. If the drive specified is not available, then the action will fail.

6.27.13.8 Get Logical Disk Information

The Get Logical Disk Information action can retrieve certain properties of the specified logical disk.



Get Logical Disk Information

Logical Drive

Machine Name: MACHINE01 (leave blank for local machine)

Drive Letter (required): D:

☑ Fail if drive not found

□ Fail if no disk in removable drive

Retrieve Logical Drive details

Free Space: DiskSpaceFree

Drive Type: DiskType

Drive Size: DiskSize

File System: FileSystemType

Volume Serial: VolumeSerial

Machine Name - enter the machine name for a remote machine, or leave blank for the local machine

Drive Letter - select the drive letter to query

Fail if drive not found - this option will fail the action if the specified drive is not found on the target machine

Fail if no disk in removable drive - only for removable drives, this option will fail the action if the specified drive doesn't contain any media

Free Space - the free space in bytes will be set to the specified variable

Drive Type - the drive type will be set to the specified variable

Drive Size - the drive size in bytes will be set to the specified variable

File System - the file system type (eg. NTFS) will be set to the specified variable

Volume Serial - the volume serial will be set to the specified variable

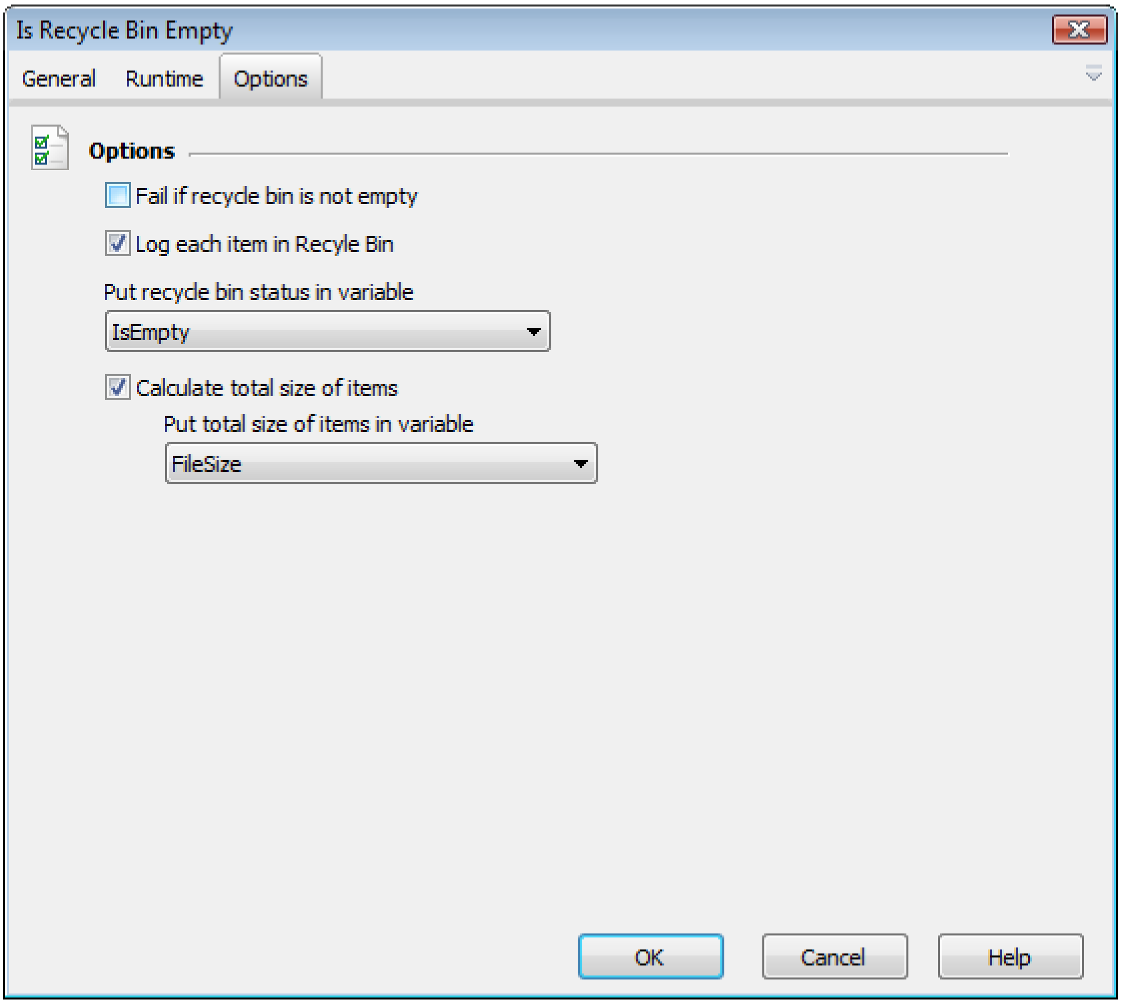
NOTES:
Drive Types:

- Removable drive
- Local hard disk

- Network disk
- Optical disk
- RAM disk

6.27.13.9 Is Recycle Bin Empty

The Is Recycle Bin Empty action will scan the current user's Recycle Bin and report the total items, total size of items and if the bin is empty or not.



Is Recycle Bin Empty

Options

□ Fail if recycle bin is not empty

☑ Log each item in Recyle Bin

Put recycle bin status in variable: IsEmpty

☑ Calculate total size of items

Put total size of items in variable: FileSize

Fail if recycle bin is not empty - choose this option if you want the action to fail if the Recycle Bin still contains at least one item

Log each item - each file and folder in the recycle bin will be written to the log

Put recycle bin status in variable - select the variable to write the status of the recycle bin to, Is Empty = true.

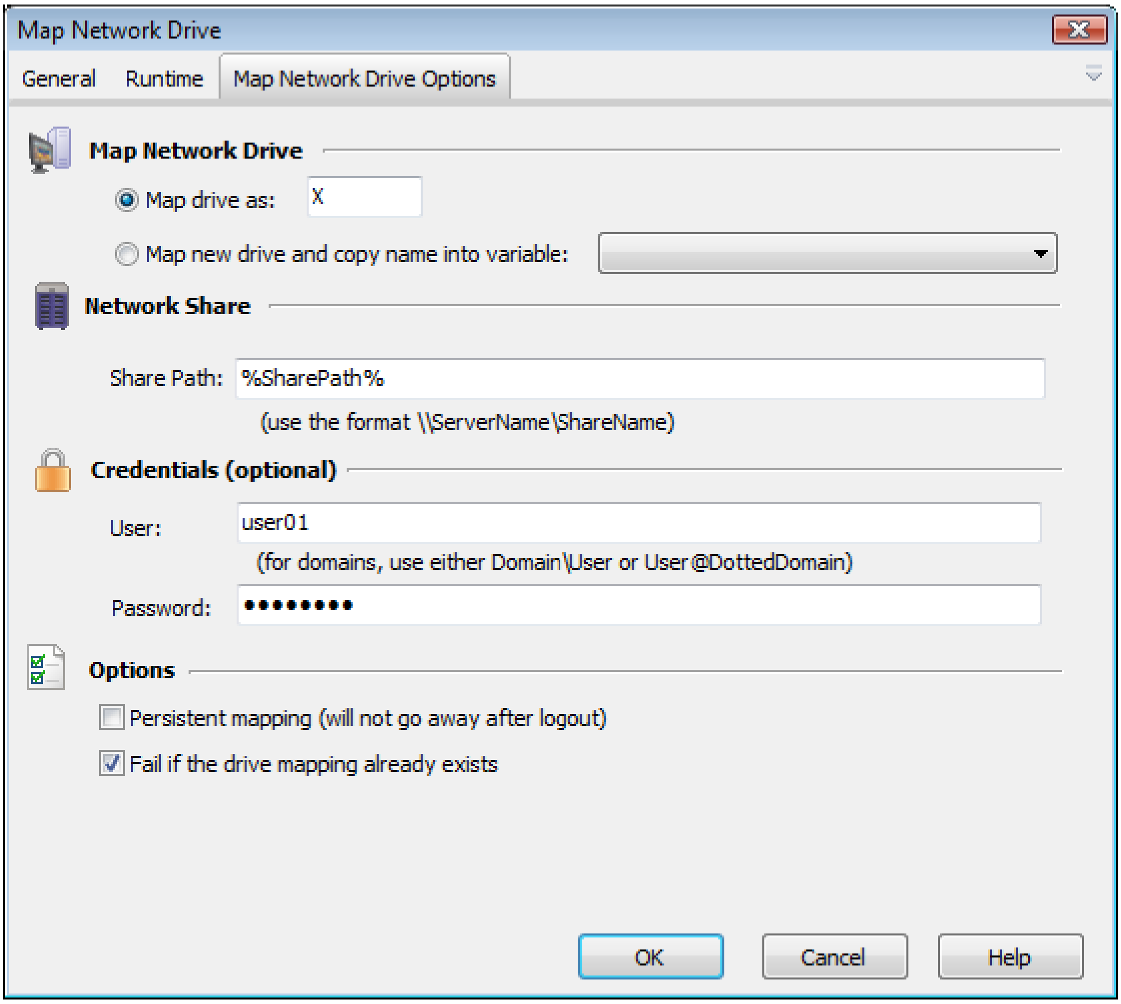
Calculate total size of items - the total size of the items in the recycle bin will be added up and written to the log (in KB)

Put total size in variable - select the variable to write the total size of items to (in KB)

To empty the Recycle Bin, use the Empty Recycle Bin action

6.27.13.1(Map Network Drive Action

The map network drive action creates a network drive mapping to a remote network share.



Map Network Drive

● Map drive as: X

○ Map new drive and copy name into variable:

Network Share

Share Path: %SharePath%

(use the format \\ServerName\ShareName)

Credentials (optional)

User: user01

(for domains, use either Domain\User or User@DottedDomain)

Password: ••••••••

Options

□ Persistent mapping (will not go away after logout)

☑ Fail if the drive mapping already exists

Map Drive As...

Specify a name for the new drive. To retrieve the drive name from a variable, use the form %VariableName%.

Map new drive and copy name...

Use this option to map a new drive to an unused drive name. Optionally, the drive name will be stored in the specified variable. Leave the variable name blank to ignore it (not recommended.)

We recommend you use this option sparingly as it introduces a small degree of non-determinism.

Network Share

Enter the name of a share path to mount.

Credentials

(Optionally) specify a user name and password to connect to the share. If you do not specify a user name and password, the Windows credentials of the currently logged on user will be used to connect.

Options

Persistent Mapping

Persistent mappings will remain after you log out, until deleted. You can use this option along with unchecking "Fail if the drive mapping already exists" to ensure that a persistent mapping is still present when you start your project.

Save Credentials

Saves the login and password for use with future connection attempts.

Fail if the drive mapping already exists

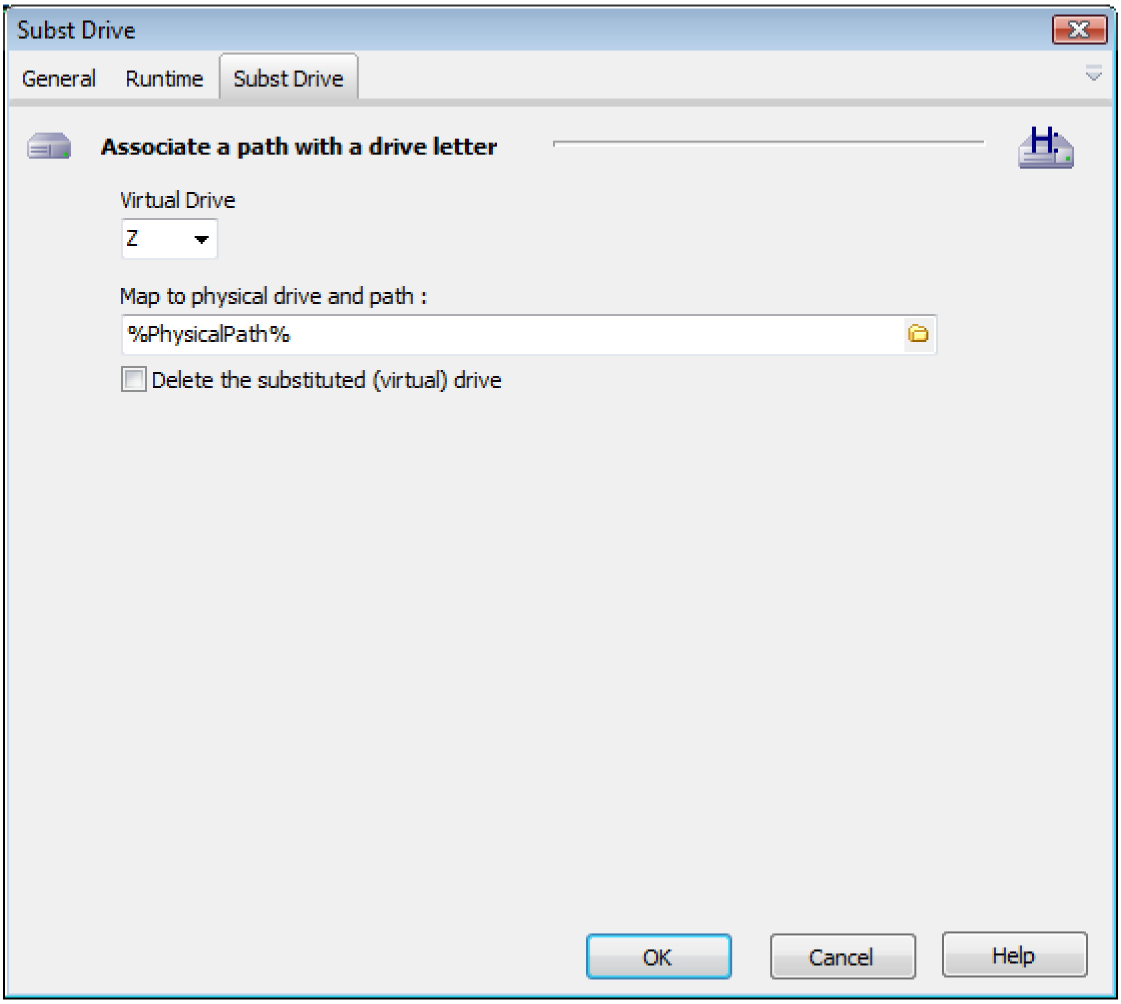
If this option is checked, the action will fail if a drive is already mapped to that drive name. If unchecked, the action will skip the mapping if the drive already exists (note that the action does not check which share is mapped to the drive, nor that the drive is in fact a drive mapping and not a real disk drive!)

This option has no effect if you are mapping a new drive name.

6.27.13.1: Subst Drive Action

This action calls the DOS subst.exe program, to map a folder to a drive letter.

To map a network share to a virtual drive, use the Map Network Drive action.

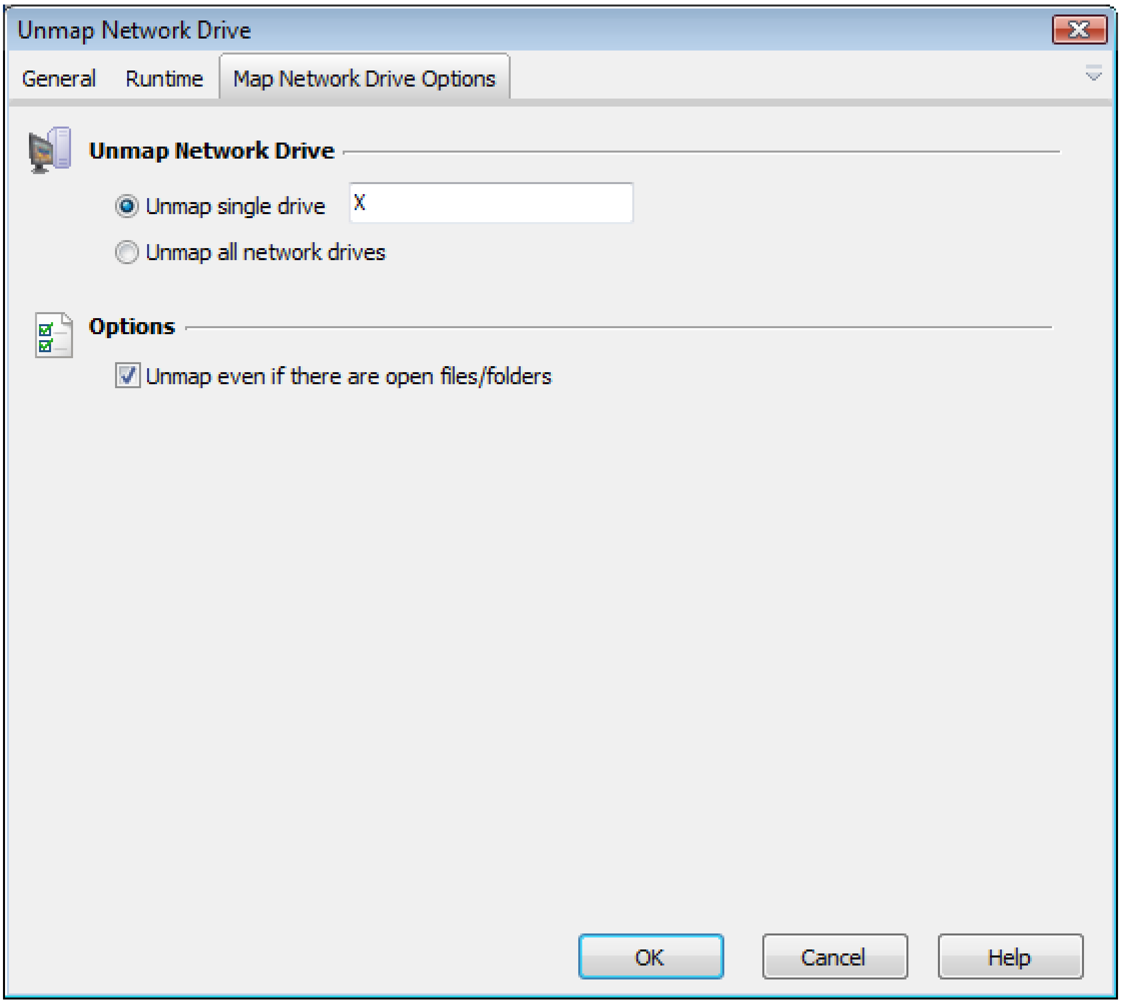


Map to physical drive and path : %PhysicalPath%

☐ Delete the substituted (virtual) drive

6.27.13.11 Unmap Network Drive Action

The unmap network drive action deletes a previously mapped network drive.



Unmap Network Drive

General | Runtime | Map Network Drive Options

Unmap Network Drive

● Unmap single drive: X

○ Unmap all network drives

Options

☑ Unmap even if there are open files/folders

Unmap Single Drive

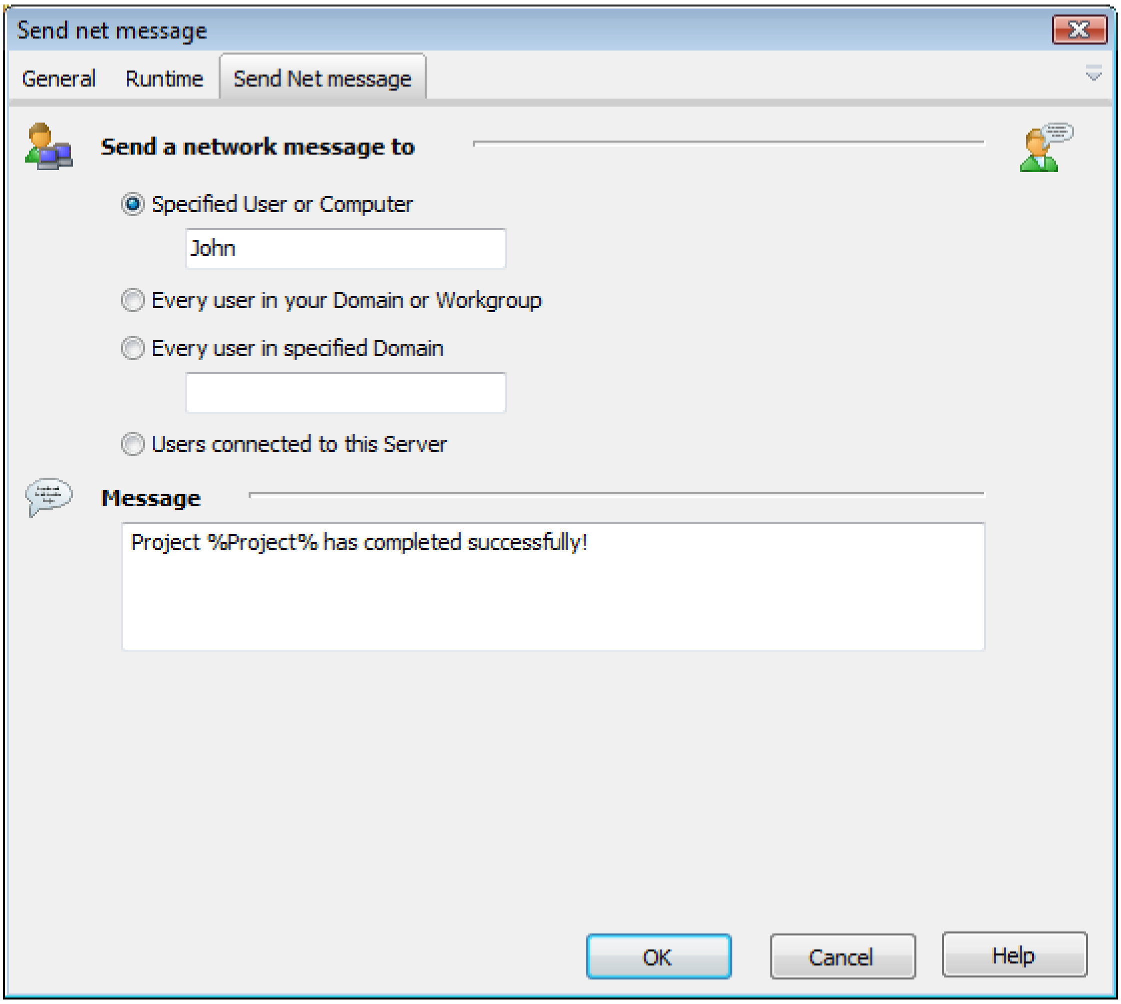
Enter the drive name you wish to unmap, or (optionally, as shown in the picture above) a variable name.

Unmap all network drives

This option will attempt to remove all mapped network drives from the system. Note that the action does not distinguish between drives mapped by Automise and drives mapped by other processes.

6.27.14 Net Send Message Action

Send a Message using windows NET SEND



Send a network message to

● Specified User or Computer

John

○ Every user in your Domain or Workgroup

○ Every user in specified Domain

○ Users connected to this Server

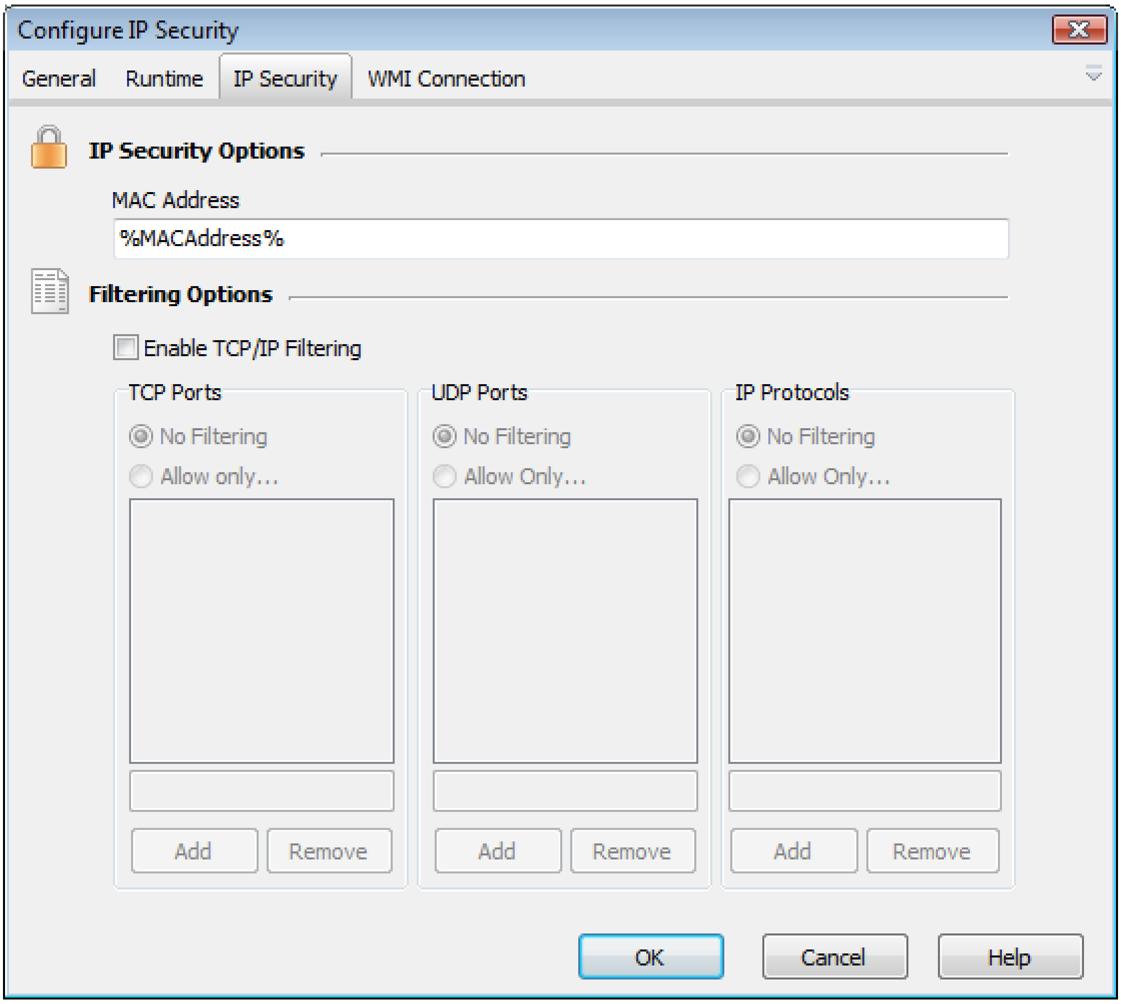
Message

Project %Project% has completed successfully!

6.27.15 Networking Configuration and Diagnostics

6.27.15.1 Configure IP Security

The Configure IP Security action enables the configuration of the security settings of the specified adapter.



MAC Address: %MACAddress%

□ Enable TCP/IP Filtering

TCP Ports: ● No Filtering ○ Allow only...

UDP Ports: ● No Filtering ○ Allow Only...

IP Protocols: ● No Filtering ○ Allow Only...

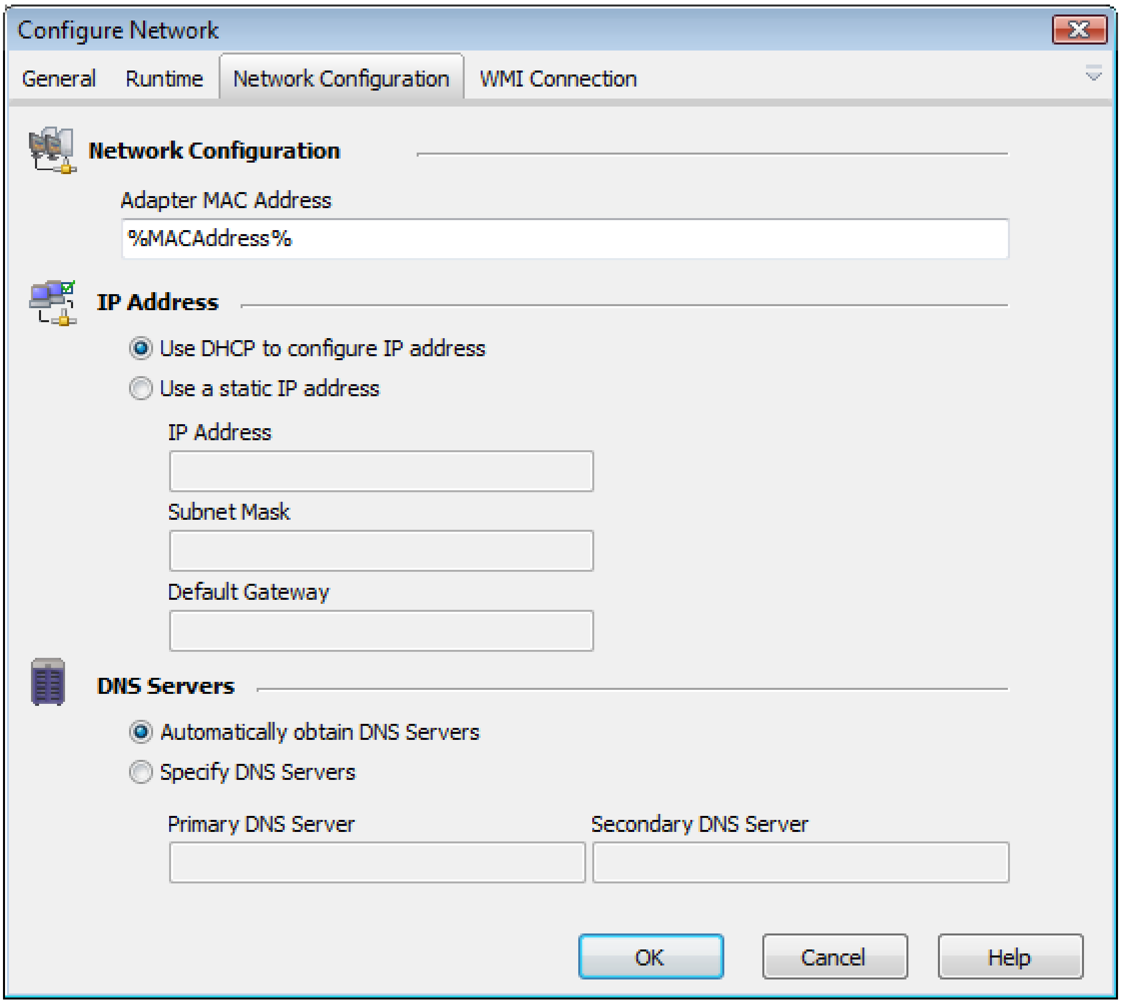
MAC Address
The MAC address of the adapter to change the security settings of.

Filtering Options
The list of ports and protocols that will be filtered.

A list of protocols can be found on the IANA website: http://www.iana.org/assignments/protocol-numbers

6.27.15.2 Configure Network

The Configure Network action enables the configuration of network adapters.



Network Configuration

Adapter MAC Address

%MACAddress%

IP Address

● Use DHCP to configure IP address

○ Use a static IP address

IP Address

Subnet Mask

Default Gateway

DNS Servers

● Automatically obtain DNS Servers

○ Specify DNS Servers

Primary DNS Server

Secondary DNS Server

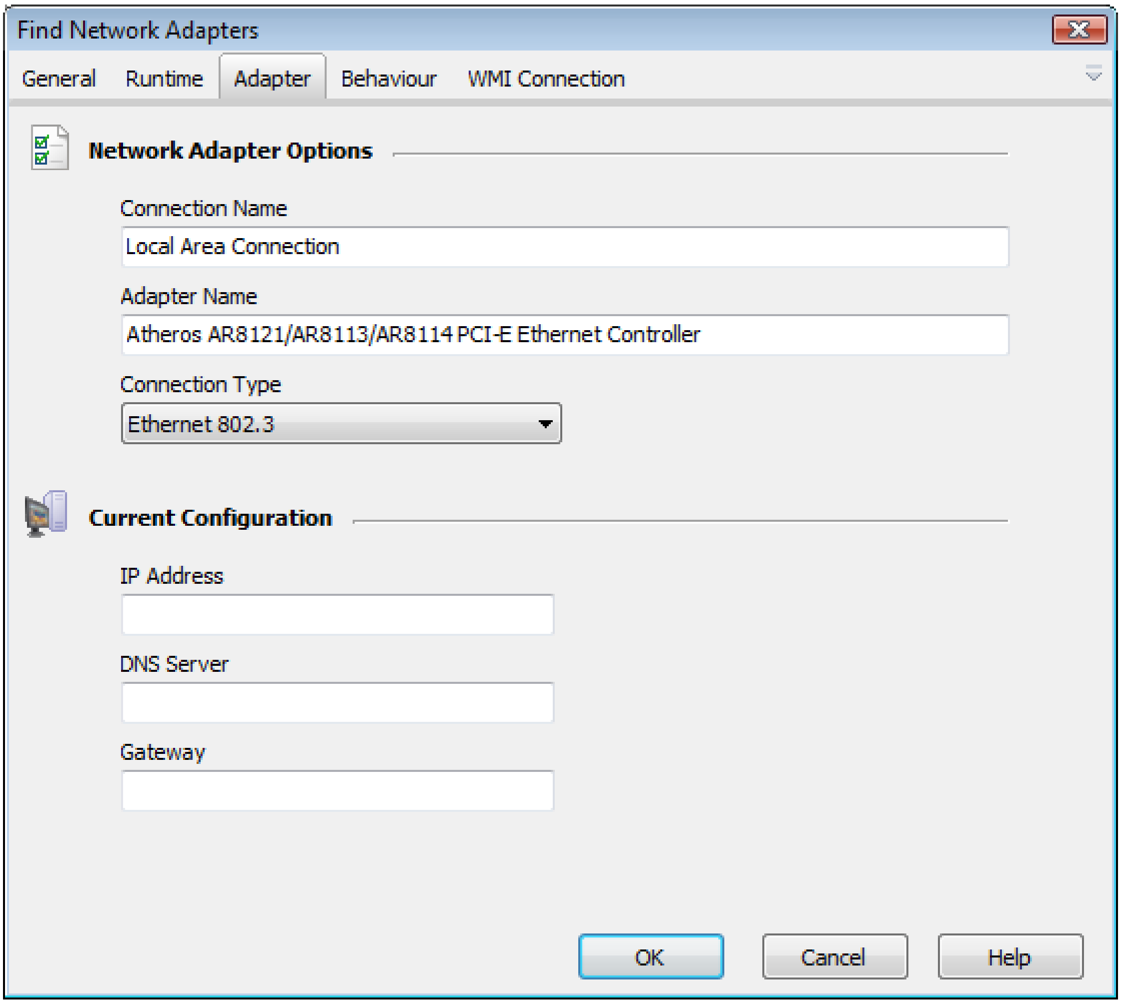
Adapter MAC Address
The MAC address of the adapter to configure.

IP Address
The network adapter can be configured to automatically retrieve its settings from a DHCP server, or the settings can be specified.

DNS Servers
The network adapter can be configured to obtain the DNS server automatically or it be can be specified.

6.27.15.3 Find Network Adapters

The Find Network Adapters action enables you to search for network adapters on the specified machine, returning the current configuration of the adapter.



Find Network Adapters

Network Adapter Options

Connection Name: Local Area Connection

Adapter Name: Atheros AR8121/AR8113/AR8114 PCI-E Ethernet Controller

Connection Type: Ethernet 802.3

Current Configuration

IP Address:

DNS Server:

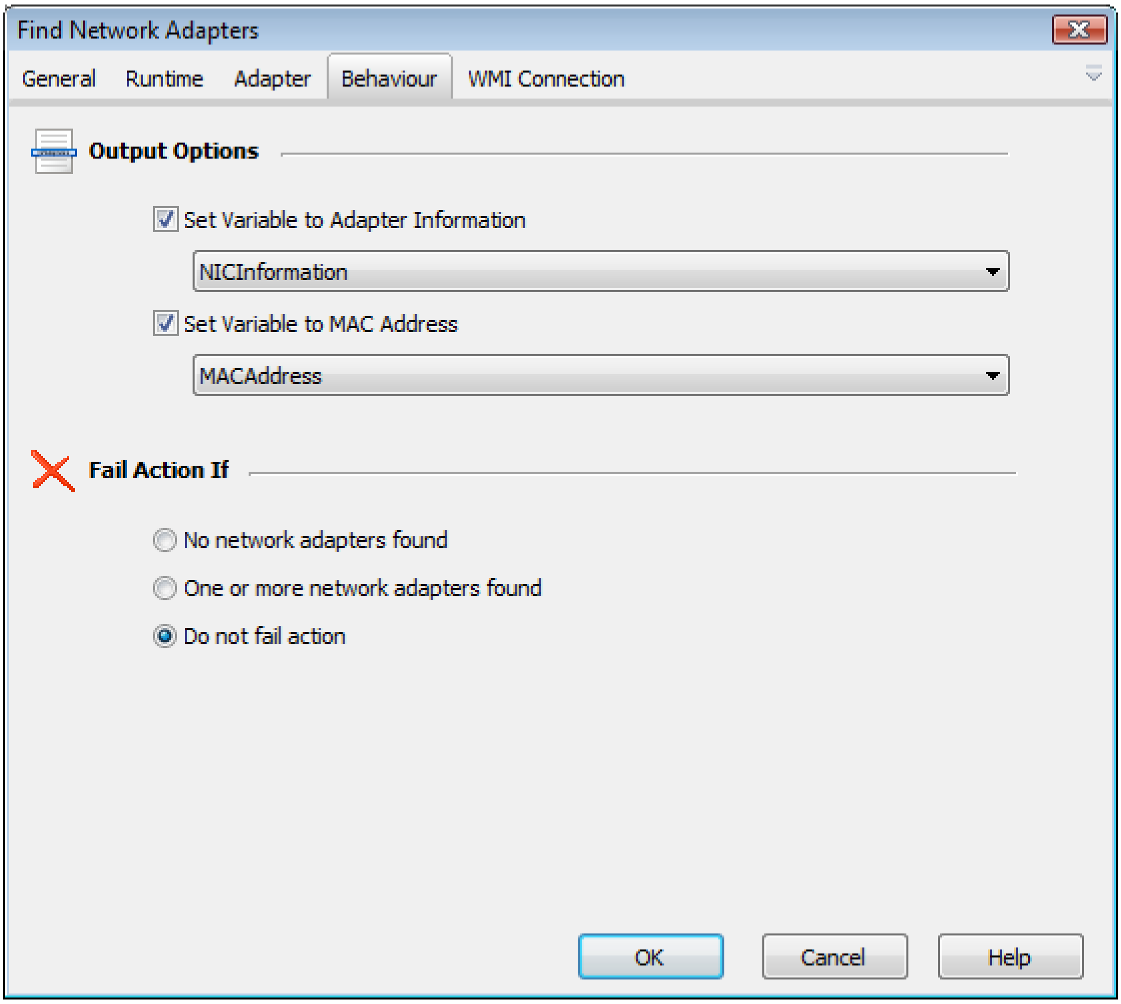
Gateway:

Connection Name
The name of the connection to be found.

Adapter Name
The name of the adapter which the connection is bound to.

Connection Type
The type of connections to find on the specified machine.

Current Configuration
The current network settings to match the connection to.



☑ Set Variable to Adapter Information

NICInformation

☑ Set Variable to MAC Address

MACAddress

○ No network adapters found

○ One or more network adapters found

● Do not fail action

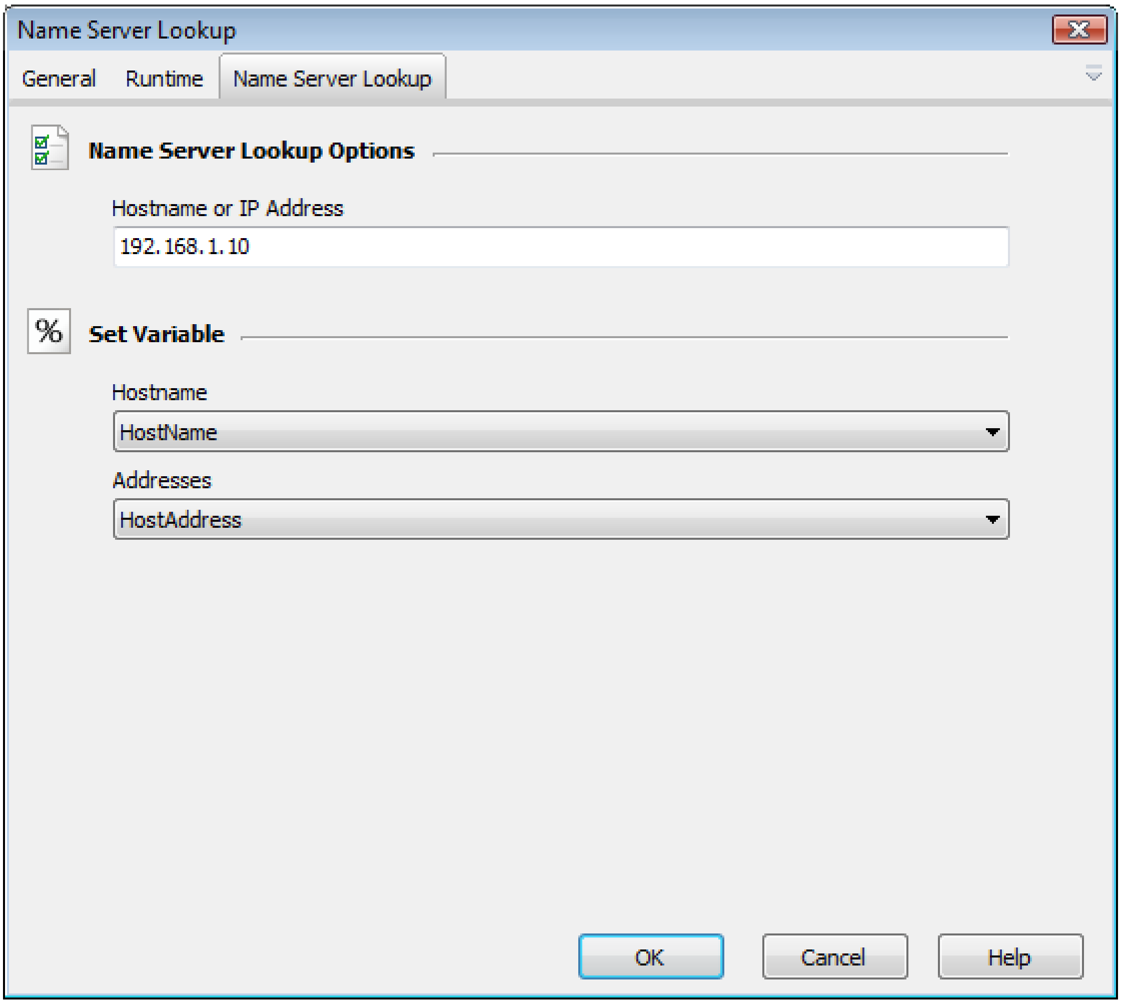
Set Variable to Adapter Information
Sets the specified variable to current configuration of the adapters found.

Set Variable to MAC Address
Sets the specified variable to the MAC Address of the adapter found.

Fail Action If
Alters the behaviour of the action in regards to the result of the search.

6.27.15.4 Name Server Lookup

The Name Server Lookup action enables you to lookup an IP Address or Host name using the default DNS Server.



Name Server Lookup Options

Hostname or IP Address: 192.168.1.10

Set Variable

Hostname: HostName

Addresses: HostAddress

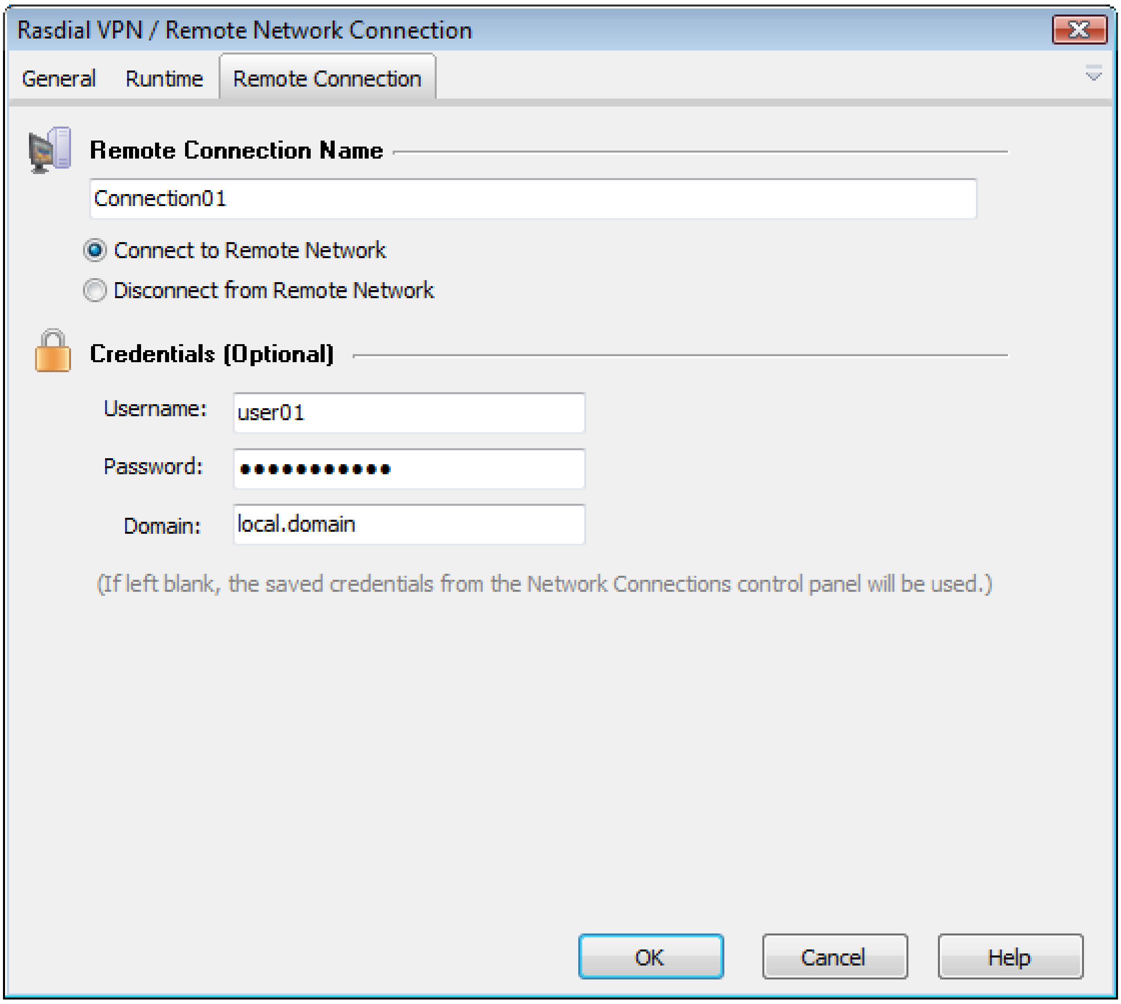
Host name or IP Address
The name or IP Address of the host to lookup on the name server.

Host name
Set the specified variable to the host name.

Addresses
Set the specified variable to a list of the IP Addresses associated with the host.

6.27.15.5 Rasdial VPN / Remote Network Connection

The Rasdial action uses rasdial.exe to connect or disconnect a dial-up network or remote network (VPN) connection, as listed in the Windows Network Connections control panel (note that the rasdial action cannot be used to enable or disable LAN or High Speed Network Connections.)



Remote Connection Name

Connection01

● Connect to Remote Network

○ Disconnect from Remote Network

Credentials (Optional)

Username: user01

Password: ●●●●●●●●●●●●

Domain: local.domain

(If left blank, the saved credentials from the Network Connections control panel will be used.)

Remote Connection Name

Enter the name of the connection as it appears in the Network Connections control panel.

Choose "Connect to Remote Network" to establish a connection to the remote network and "Disconnect from Remote Network" to disconnect. Note that the action will not fail if the connection is already connected/disconnected.

Note that rasdial.exe counts the number of times a connection is established and will require an equal number of disconnections before the network connection is terminated. This means that if the network connection is already connected when the Connect action is run, Disconnect will not terminate the connection (however, a further Disconnect action will terminate it.) For reliable connecting/disconnecting, it is recommended to use a Try... Finally block.

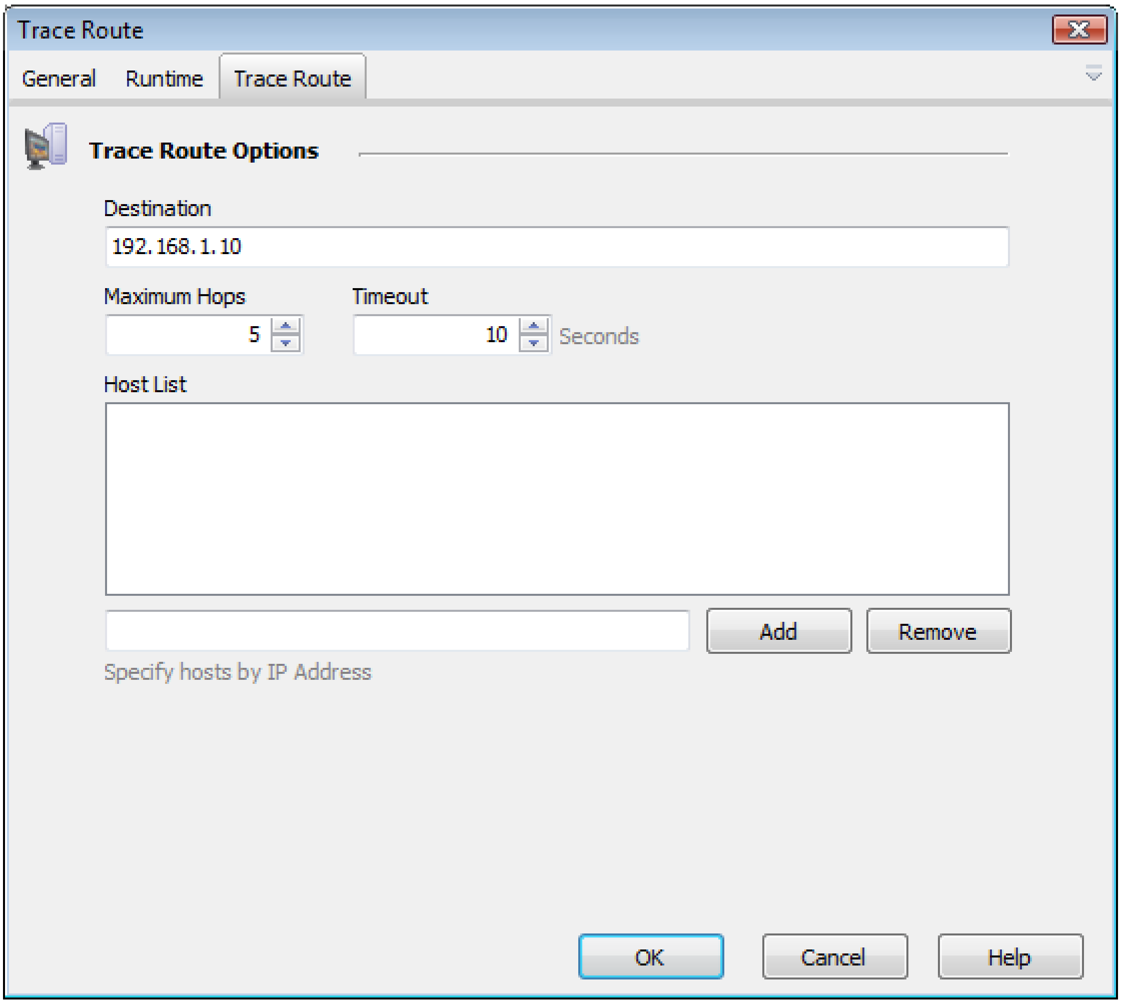
Credentials (Optional)

Credentials can be entered here if they are not already saved in the Network Connections "Connect" dialog.

If the Credentials section is left blank and no username and password is set in the Network Connections control panel, a Connect dialog will be displayed when the action is run.

6.27.15.6 Trace Route

The Trace Route action allows you to determine the path taken to a remote host, by sending ICMP Echo Requests to the destination and decrementing the TTL each time a router is hit.



Destination: 192.168.1.10

Maximum Hops: 5

Timeout: 10 Seconds

Host List

Specify hosts by IP Address

Destination
The destination host to try to be reached. The host can be specified by its host name or an IP address.

Maximum Hops
Specifies the maximum number of hops to take while trying to reach the destination.

Timeout
The amount of time in seconds to wait for an echo reply message is to be sent back.

Host List
The Echo Request message is sent using the Loose Source Route option in the IP header, with the intermediate destinations set by the host list. The hosts can only be specified by an IP address.

6.27.16 NTFS Junction Points

An NTFS junction point (JP) is a type of NTFS reparse point in the NTFS file system. It requires NTFS version 5.0 or later, which can be created (or converted from a FAT partition) using Windows 2000 or later (e.g. Windows XP). Junction Points can be used in a similar way to symbolic links - allowing the creation of a link to a folder that is, for most intents and purposes, the same as the folder itself. This has many benefits over a Windows shell shortcut (.lnk) file, such as allowing you to access files within the shortcut via explorer, the console, etc. Junction points can only link to folders and volumes.

Warning

- Microsoft strongly recommends:

1. Use NTFS ACLs to protect junction points from inadvertent deletion.
2. Use NTFS ACLs to protect files and directories targeted by junction points from inadvertent deletion or other file system operations.
3. Never delete a junction point using Explorer, a del /s command, or other file system utilities that walk recursively into directory trees. These utilities will affect the target directory and all subdirectories. Instead, use the utilities described below to delete junction points.
4. Use caution when applying ACLs or changing file compression in a directory tree that includes NTFS Junction Points.
5. Do not create namespace cycles with NTFS or DFS junction points.
6. Place all your junction points at a secure location in a namespace where you can test them out in safety, and other users will not mistakenly delete them or walk through them.

- Obscure: There are issues relating to junction points on MS Windows 2000 domain controllers & certain Active Directory files.

Actions

The following actions are available for querying, creating and removing Mount Points:

- Mount Volume - create a new mount point
- Get Mount Points Volume Names - get the available mount points on the current system
- Remove Mount Point - removes the volume mount point from the specified directory
- Get Mounted Volume Name - gets the mounted volume name for the specified directory
- Permanent Remove Mount Point - removes the volume mount point from the specified directory, dismounts the volume and makes the volume not mountable
- Clean Old Mount Point Entries - Removes volume mount point directories and registry settings for volumes that are no longer in the system
- Disable Automatic Mounting - disables automatic mounting of new volumes
- Enable Automatic Mounting - re-enables automatic mounting of new volumes

6.27.16.1 Clean Old Mount Point Entries

The Clean Old Mount Point Entries action removes volume mount point directories and registry settings for volumes that are no longer in the system.

6.27.16.2 Disable Automatic Mounting

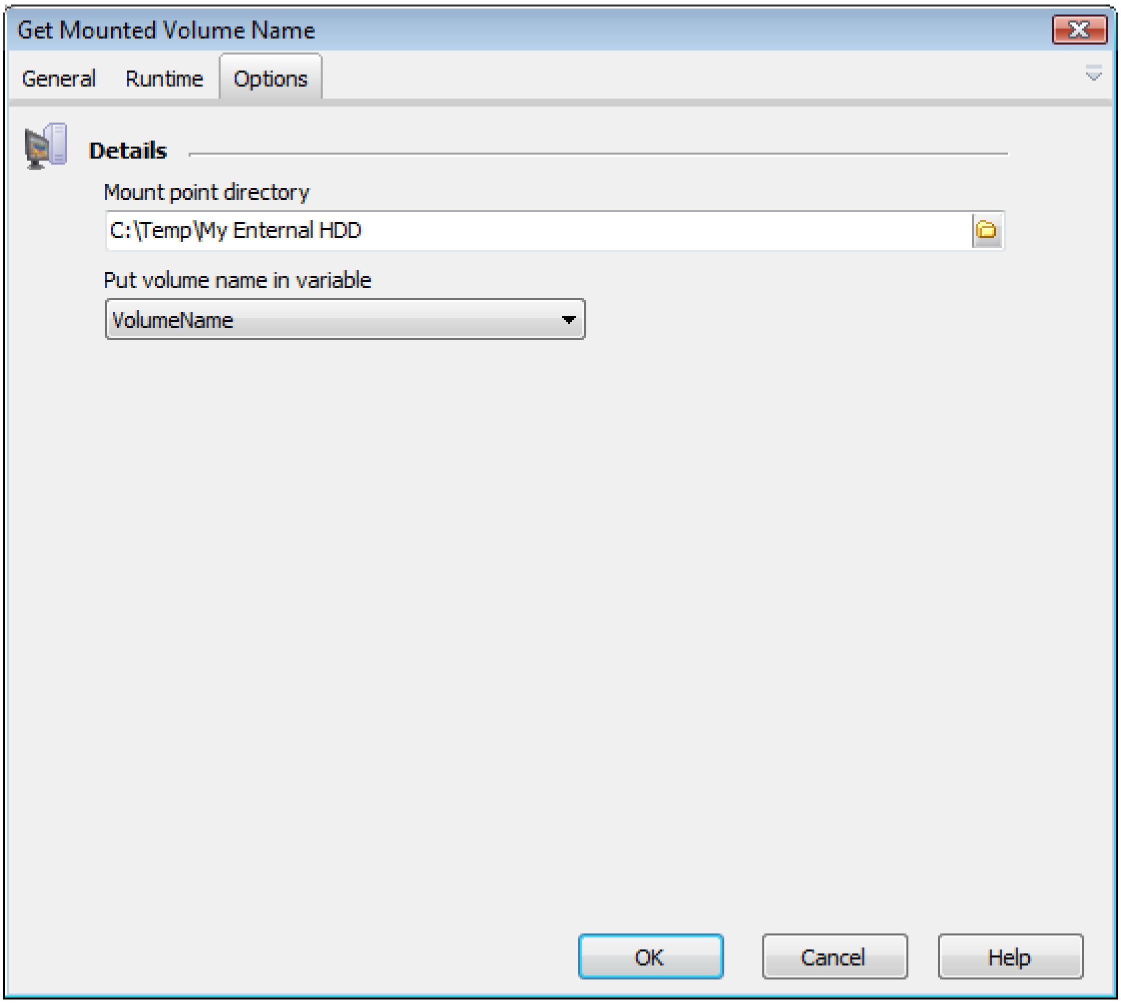
The Disable Automatic Mounting action disables automatic mounting of new volumes.

6.27.16.3 Enable Automatic Mounting

The Enable Automatic Mounting action re-enables automatic mounting of new volumes.

6.27.16.4 Get Mount Points Volume Names

The Get Mount Points Volume Names action will get the available mount points on the current system.



Mount point directory: C:\Temp\My Enternal HDD

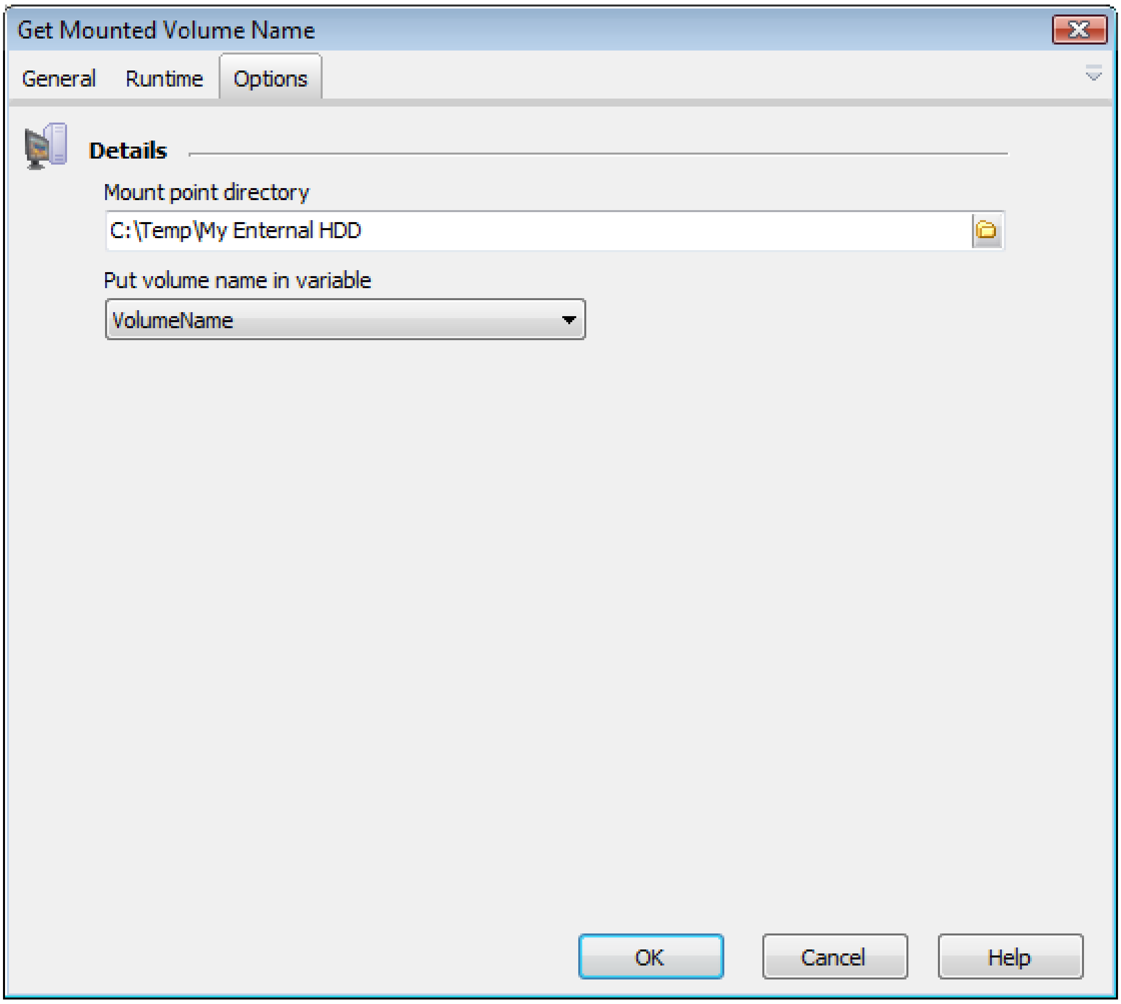
Put volume name in variable: VolumeName

Save Volume names to variable
Specify the variable to save the volume names to.

Include mount points
This option will include the mount points in the log and the information saved to the variable (if specified).

6.27.16.5 Get Mounted Volume Name

The Get Mounted Volume Name action gets the mounted volume name for the specified directory

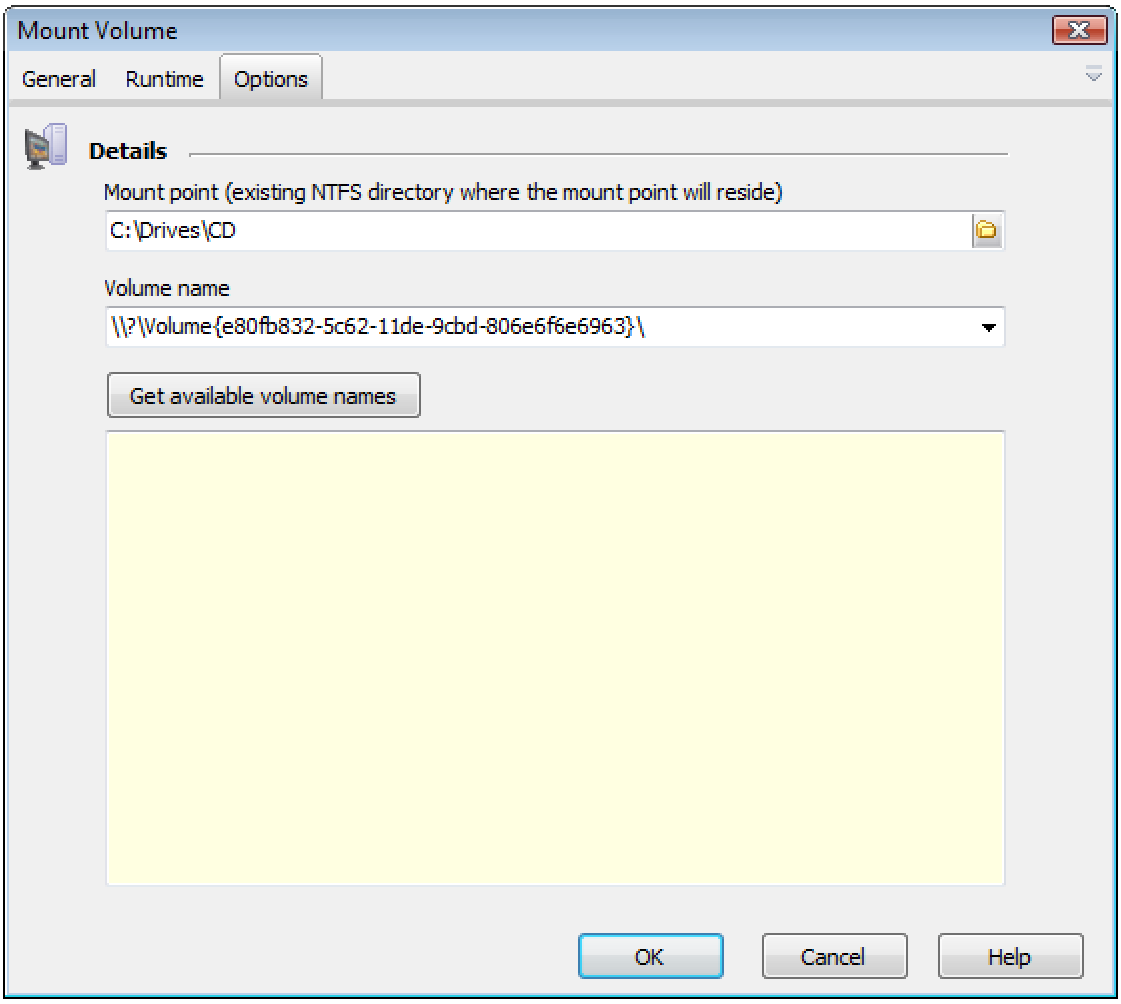


Mount point directory
C:\Temp\My Enternal HDD

Put volume name in variable
VolumeName

6.27.16.6 Mount Volume

The Mount Volume action creates a new mount point.



Mount point (existing NTFS directory where the mount point will reside)
C:\Drives\CD

Volume name
\\?\Volume{e80fb832-5c62-11de-9cbd-806e6f6e6963}\

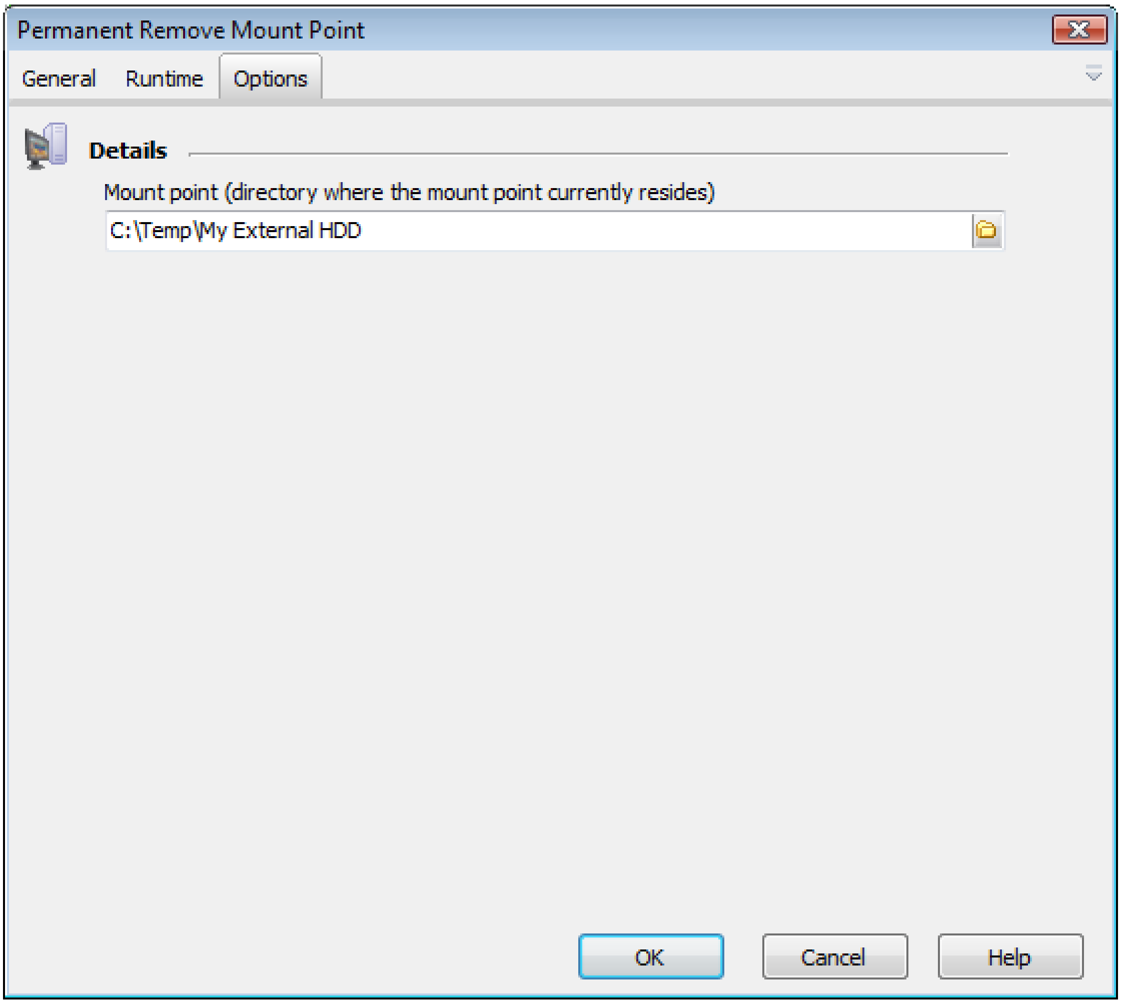
Mount point
Specify an existing directory where you want the new mount point to reside.

Volume name
Select or enter the volume name that will be the target of the mount point.

Get available volume names
Pressing this button will populate the Volume name combo box and the memo with the available volume names

6.27.16.7 Permanent Remove Mount Point

The Permanent Remove Mount Point action removes the volume mount point from the specified directory, dismounts the volume and makes the volume not mountable



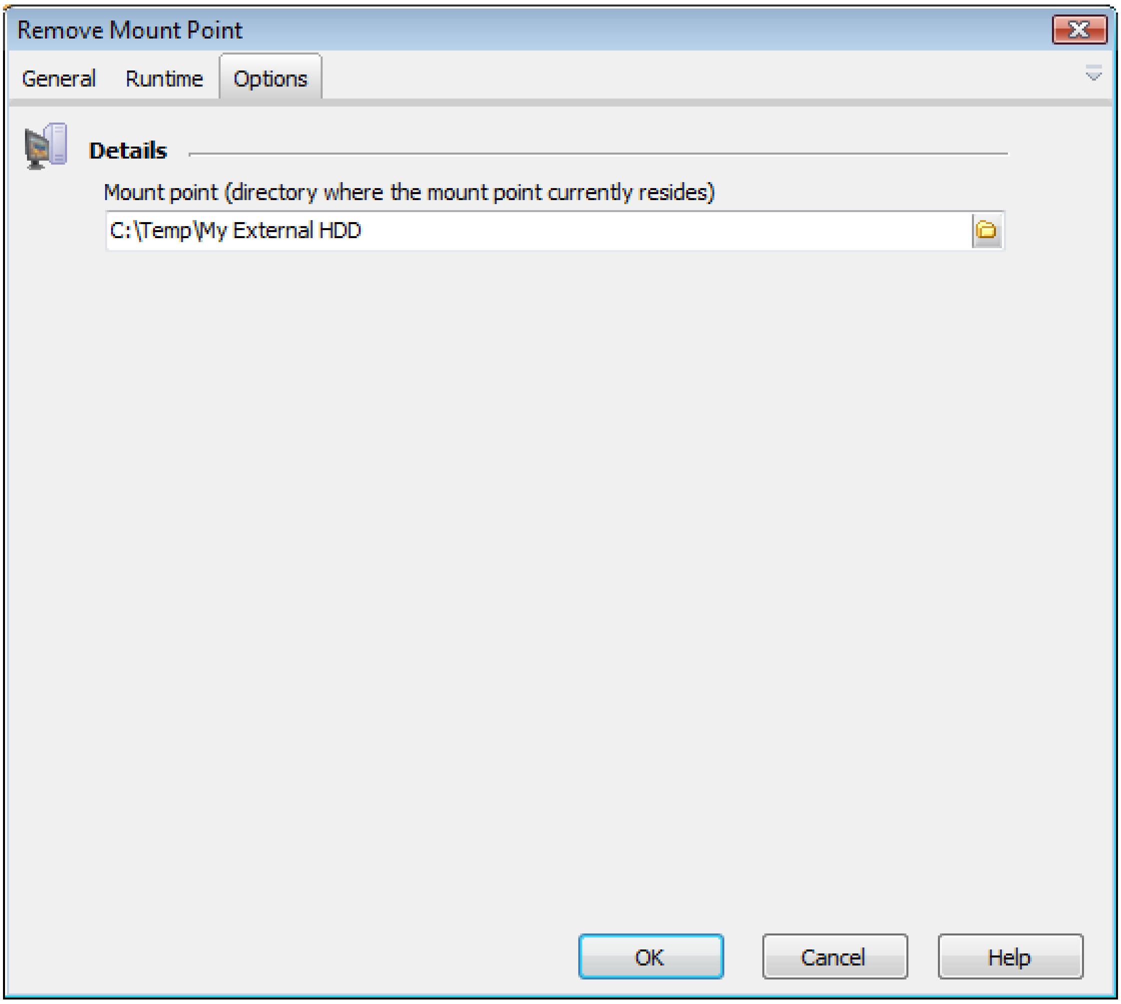
Mount point (directory where the mount point currently resides)

C:\Temp\My External HDD

NOTE: You can only permanently remove a mount point when it's the only mount point for the volume.

6.27.16.8 Remove Mount Point

The Remove Mount Point action removes the volume mount point from the specified directory.



Details

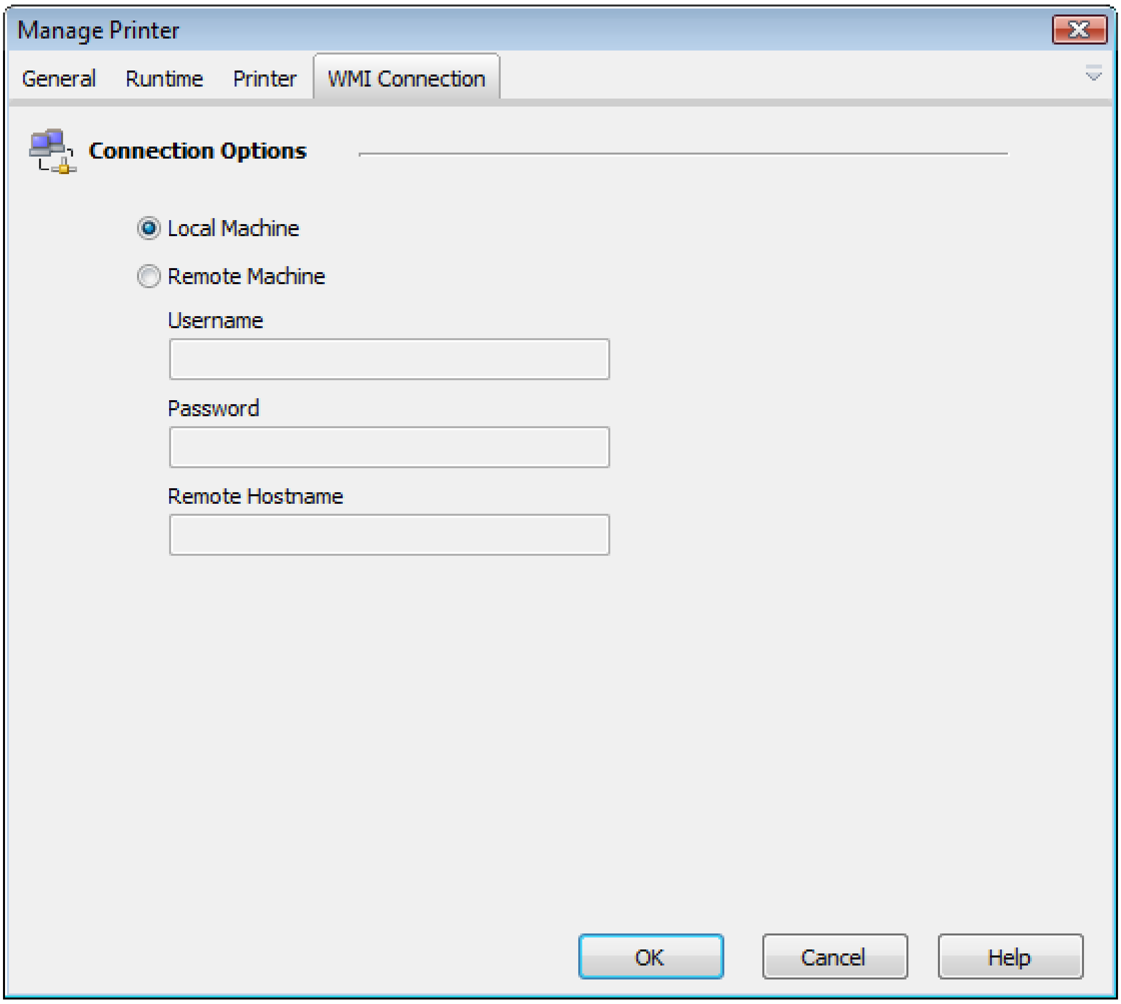
Mount point (directory where the mount point currently resides)

C:\Temp\My External HDD

Specify the mount point (directory) to delete.

6.27.17 Printers

The Printer actions allow for management of printers or print jobs, a common options page is used for the connection details.



● Local Machine

○ Remote Machine

Username

Password

Remote Hostname

Local Machine
The action will connect to the local machine, with the credentials of the current user.

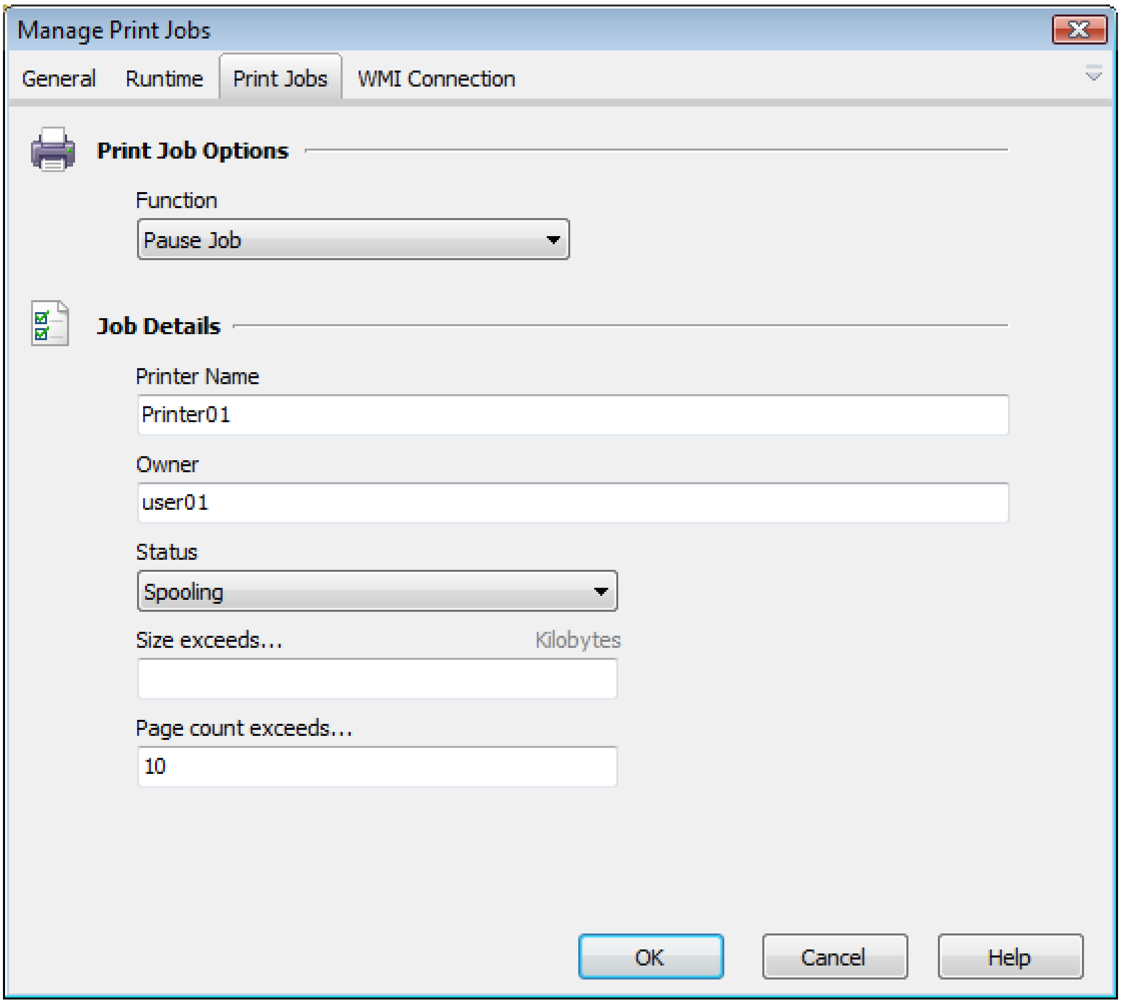
Remote Machine
The action will connect to a remote machine, using the specified user account.

Username
The username can be specified either by just the *username* or *DOMAIN/username*

Remote Host name
The computer which to connect to, if the remote computer is in a different domain then that of the current user account, a Fully Qualified Domain Name must be specified. An IPv4/IPv6 may also be specified.

6.27.17.1 Manage Print Jobs

The Manage Print Jobs action enables you to cancel, pause or resume print jobs.



Print Job Options

Function: Pause Job

Job Details

Printer Name: Printer01

Owner: user01

Status: Spooling

Size exceeds... Kilobytes

Page count exceeds... 10

Function
Pause Job - Pauses the print job.
Resume Job - The print job will be resumed, if it is currently paused.
Change Priority - Allows the print job to be given a priority.
Remove Job - Removes the print job from the queue.

Printer Name
The name of the printer which the print job belongs to.

Owner
The user account who owns the print job.

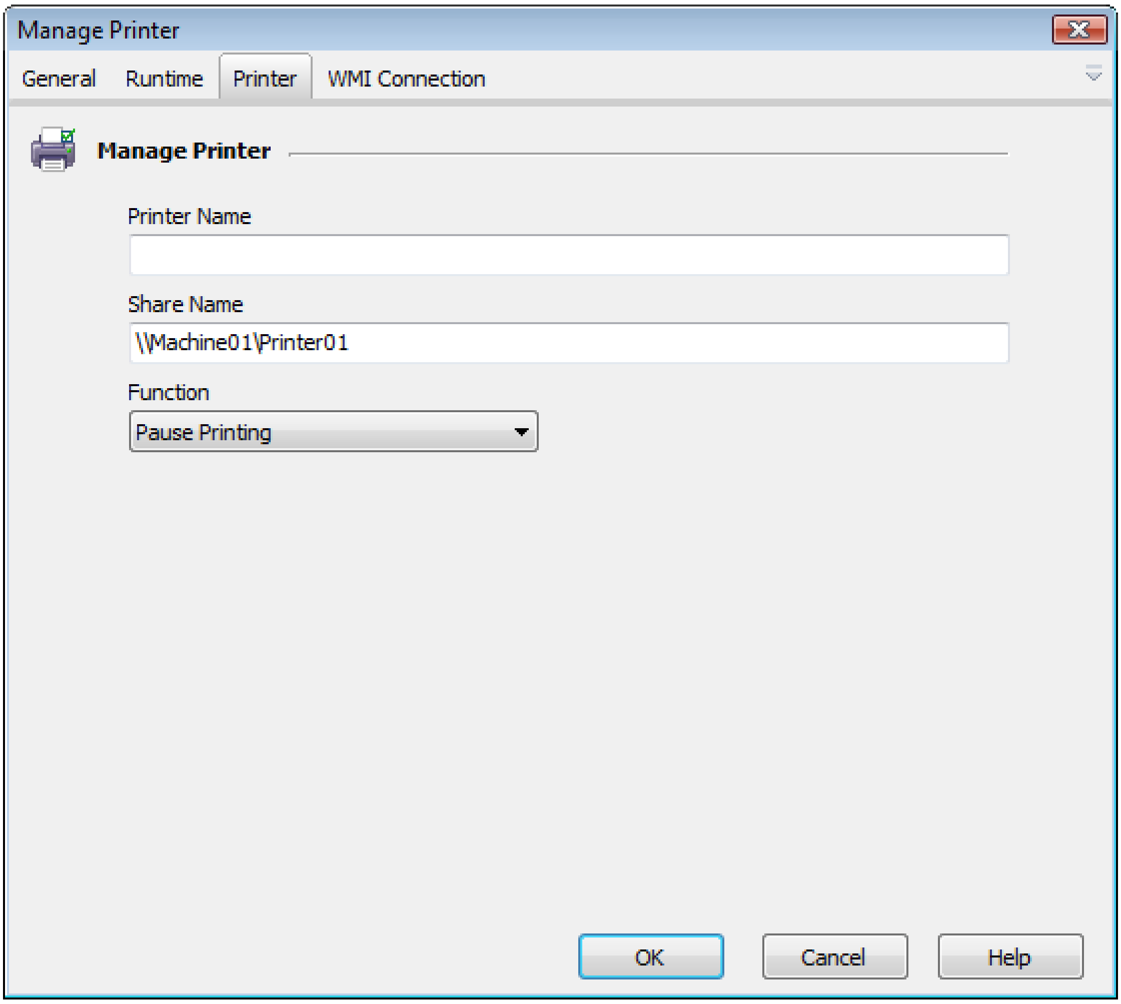
Status
Finds all print jobs with the specified status.

Size Exceeds...
Finds all print jobs which exceed the specified size.

Page Count Exceeds...
Finds all print jobs which exceed the specified page count.

6.27.17.2 Manage Printer

The Manage Printer action enables the management of printers on a local or remote windows based computer.



Printer Name

Share Name \\Machine01\Printer01

Function Pause Printing

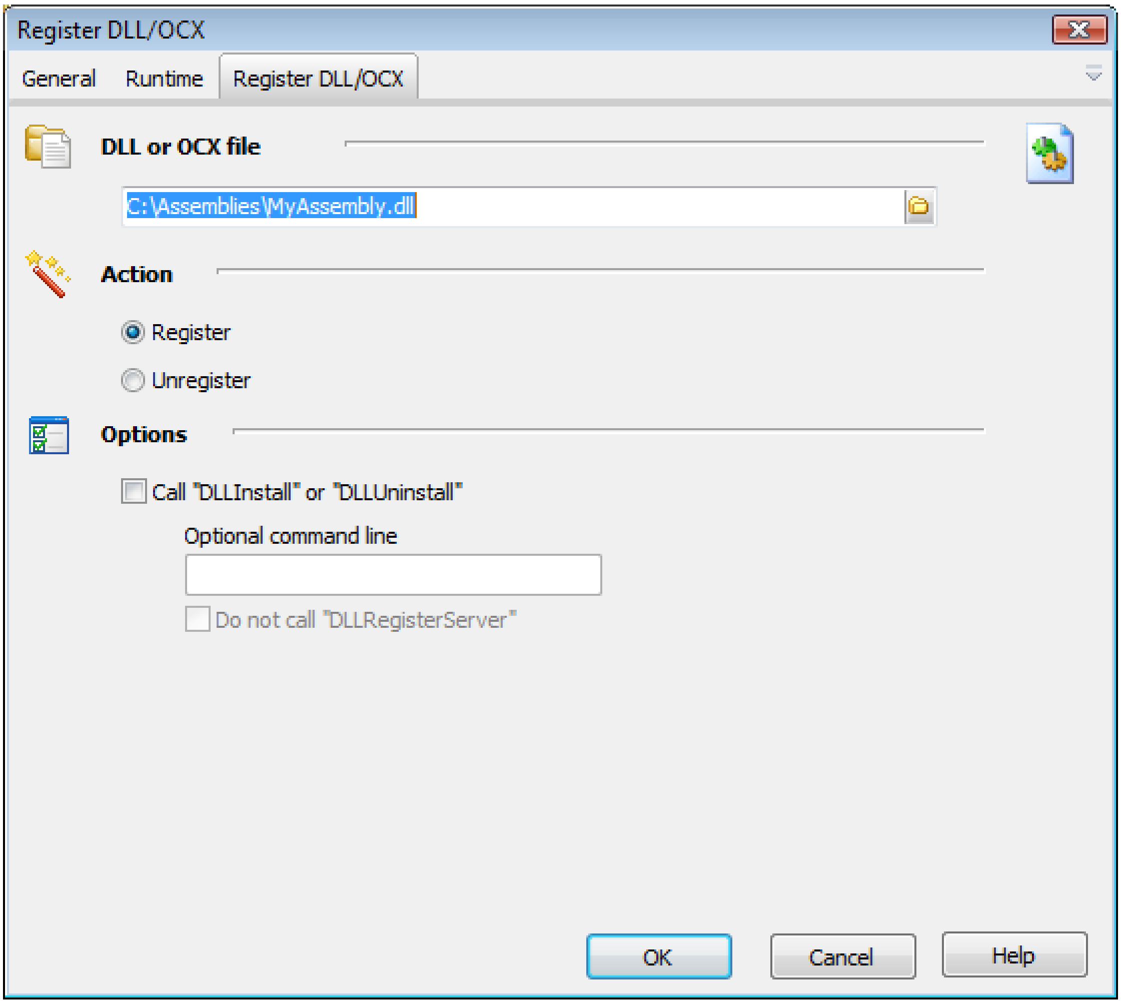
Printer Name
The name of the printer to be managed. This can be left blank if a share name is provided.

Share Name
The share name of the printer to be managed. This can be left blank if a printer name is provided.

Function
Pause Printing - The printers state will be set to paused.
Resume Printing - The printer will resume printing, if it has been paused.
Cancel all Print Jobs - All print jobs in the queue will be removed.
Print Test Page - A test page will be sent to the printer.
Set as Default Printer - The printer will be set as the default printer on the computer.
Rename Printer - Allows the printer to be renamed.

6.27.18 Register DLL/OCX Action

Register an ActiveX control (OCX) or COM DLL.



DLL or OCX file

C:\Assemblies\MyAssembly.dll

Action

● Register

○ Unregister

Options

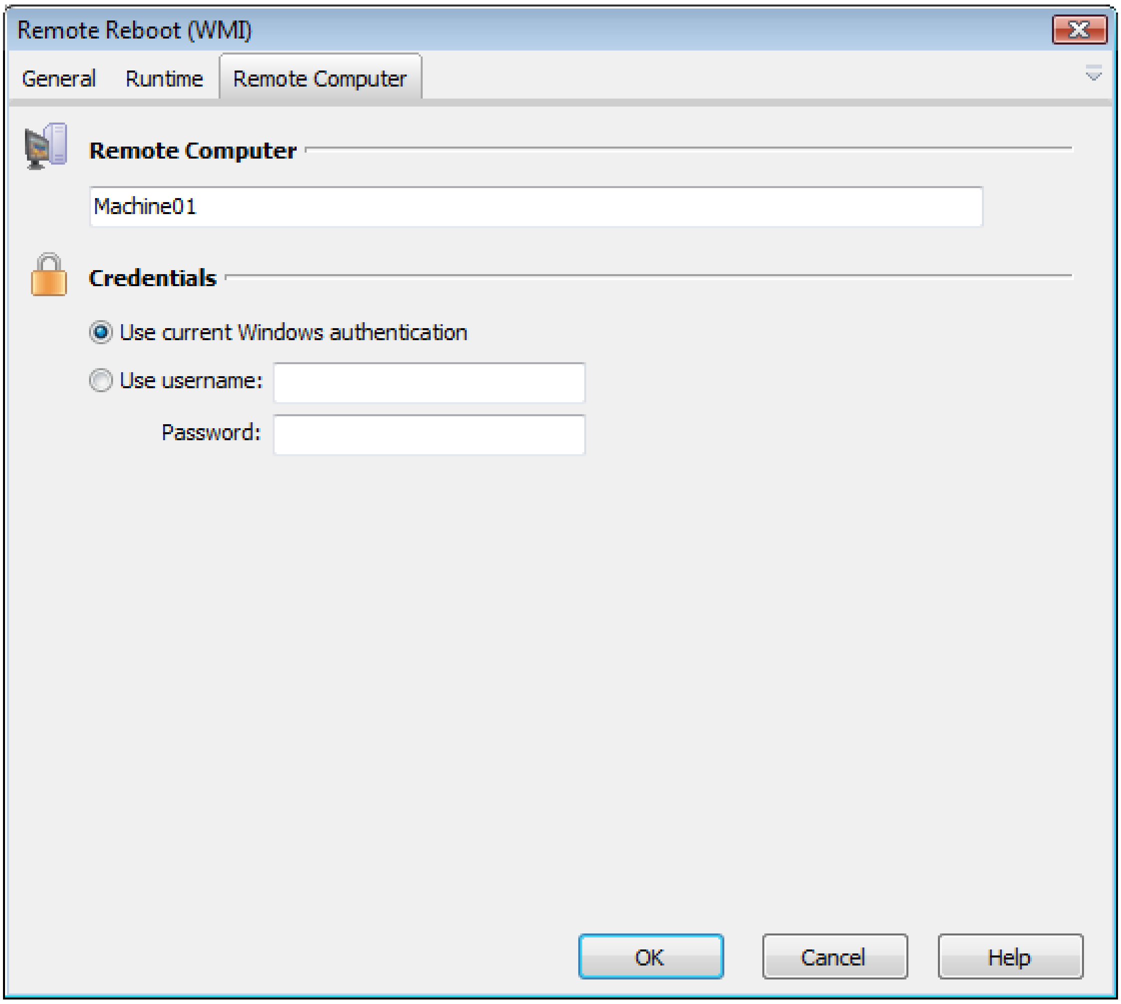
□ Call "DLLInstall" or "DLLUninstall"

Optional command line

□ Do not call "DLLRegisterServer"

6.27.19 Remote Reboot Action

The Remote Reboot action allows you to reboot a Windows computer using WMI.



Remote Computer

Machine01

Credentials

● Use current Windows authentication

○ Use username:

Password:

Remote Computer

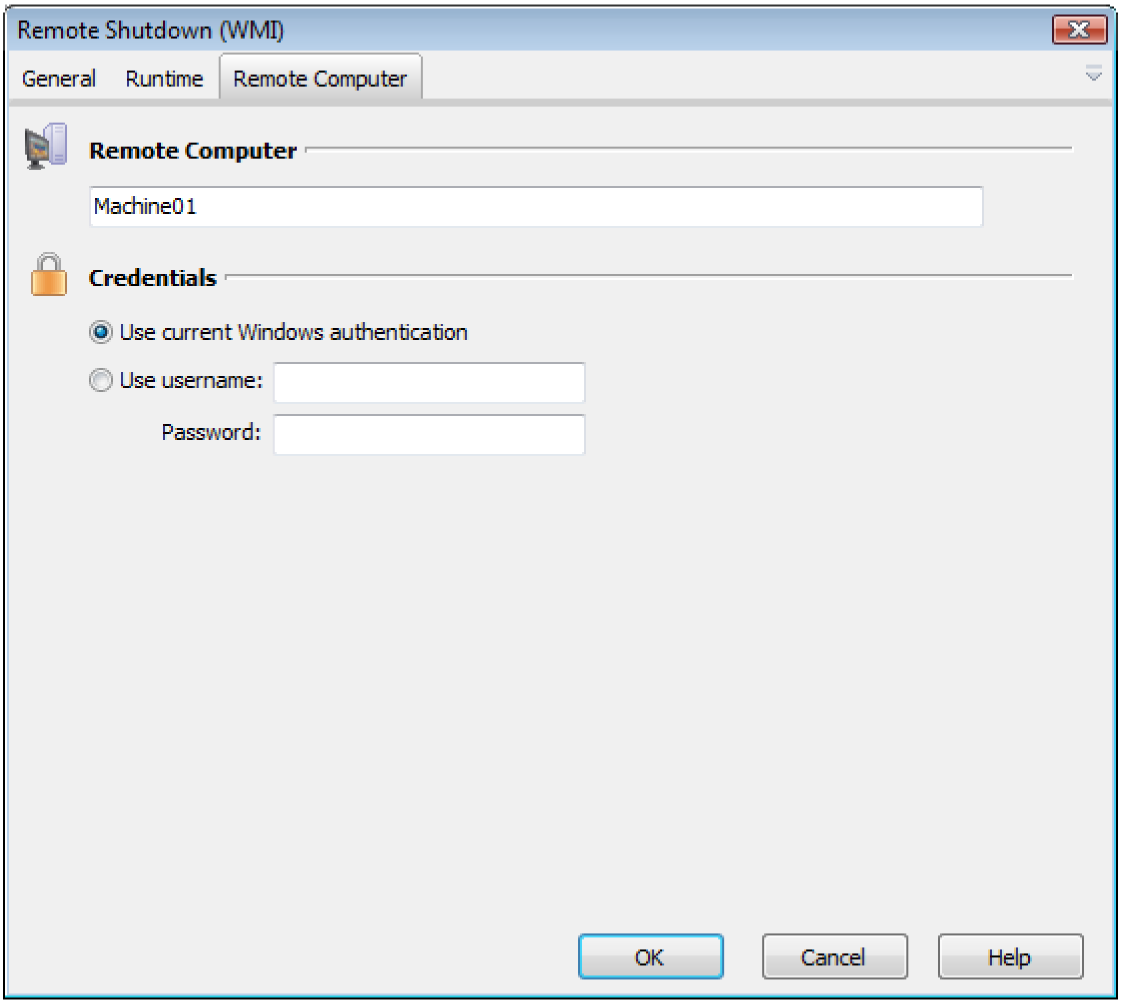
Enter the address of the computer to reboot. Note that the Remote Reboot action cannot be used on the local machine.

Credentials

WMI can either use the current Windows account, or a specific user account, for authentication.

6.27.20 Remote Shutdown Action

The Remote Shutdown action allows you to shut down a Windows computer using WMI.



Remote Shutdown (WMI)

Remote Computer

Machine01

Credentials

● Use current Windows authentication

○ Use username:

Password:

Remote Computer

Enter the address of the computer to shut down. Note that the Remote Shutdown action cannot be used on the local machine.

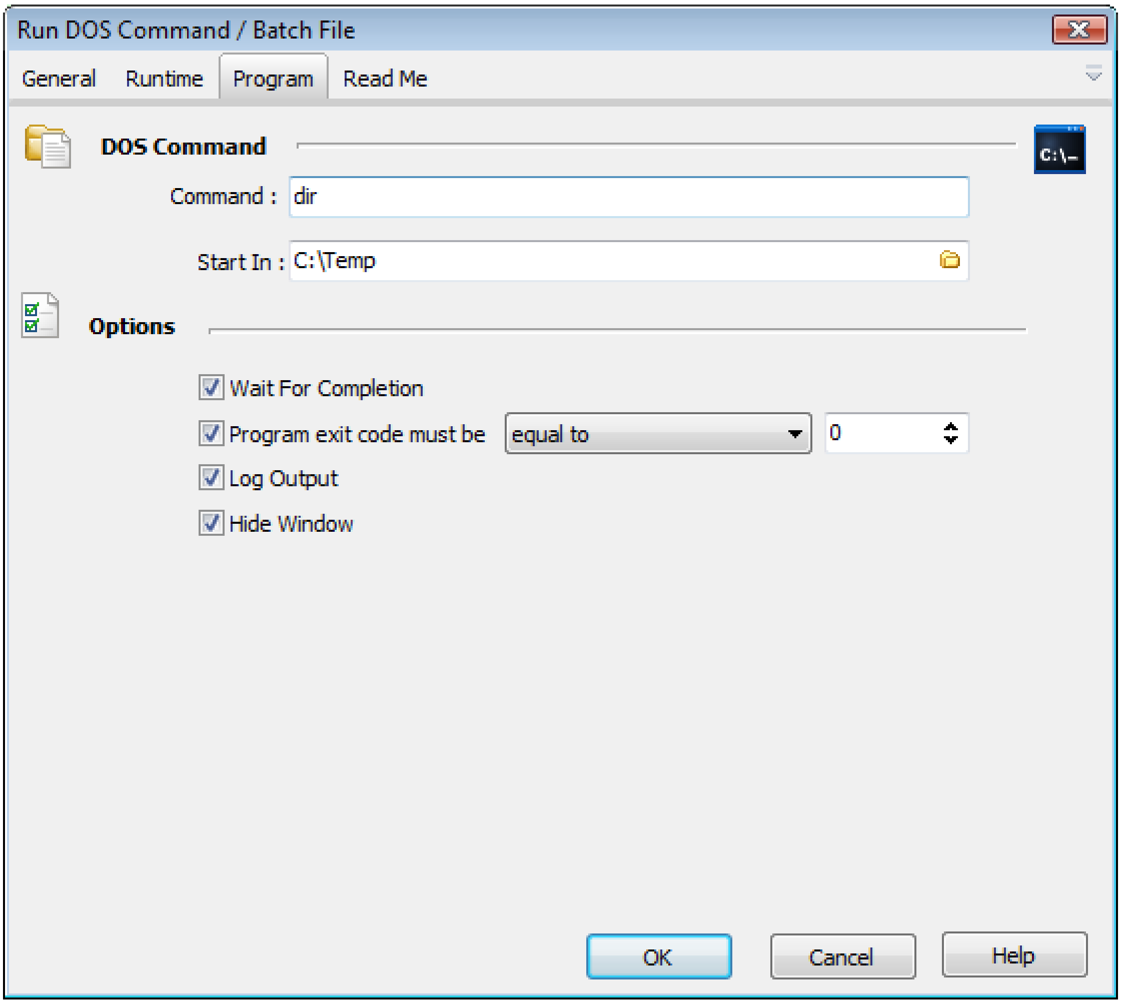
Credentials

WMI can either use the current Windows account, or a specific account, for authentication.

6.27.21 Run DOS Command Action

The Run DOS Command allows you to execute any native DOS command or batch file.

For executing programs (including console programs), use the Execute Program Action (unless you wish to redirect console output to a file with '>'.)



Command : dir

Start In : C:\Temp

☑ Wait For Completion

☑ Program exit code must be equal to 0

☑ Log Output

☑ Hide Window

Command

The full command to execute, including any parameters and output redirection.

Start In

The directory in which to execute the command.

Wait For Completion
Program exit code must be...
Log Output
HideWindow

All of these properties are identical to the Execute Program properties of the same name.

Note that for **Log Output**, some programs (for example, XCopy) cannot have their output captured.

Scripting Info

The Action properties available are :

property StartInDir : WideString;
property LogOutput : WordBool;

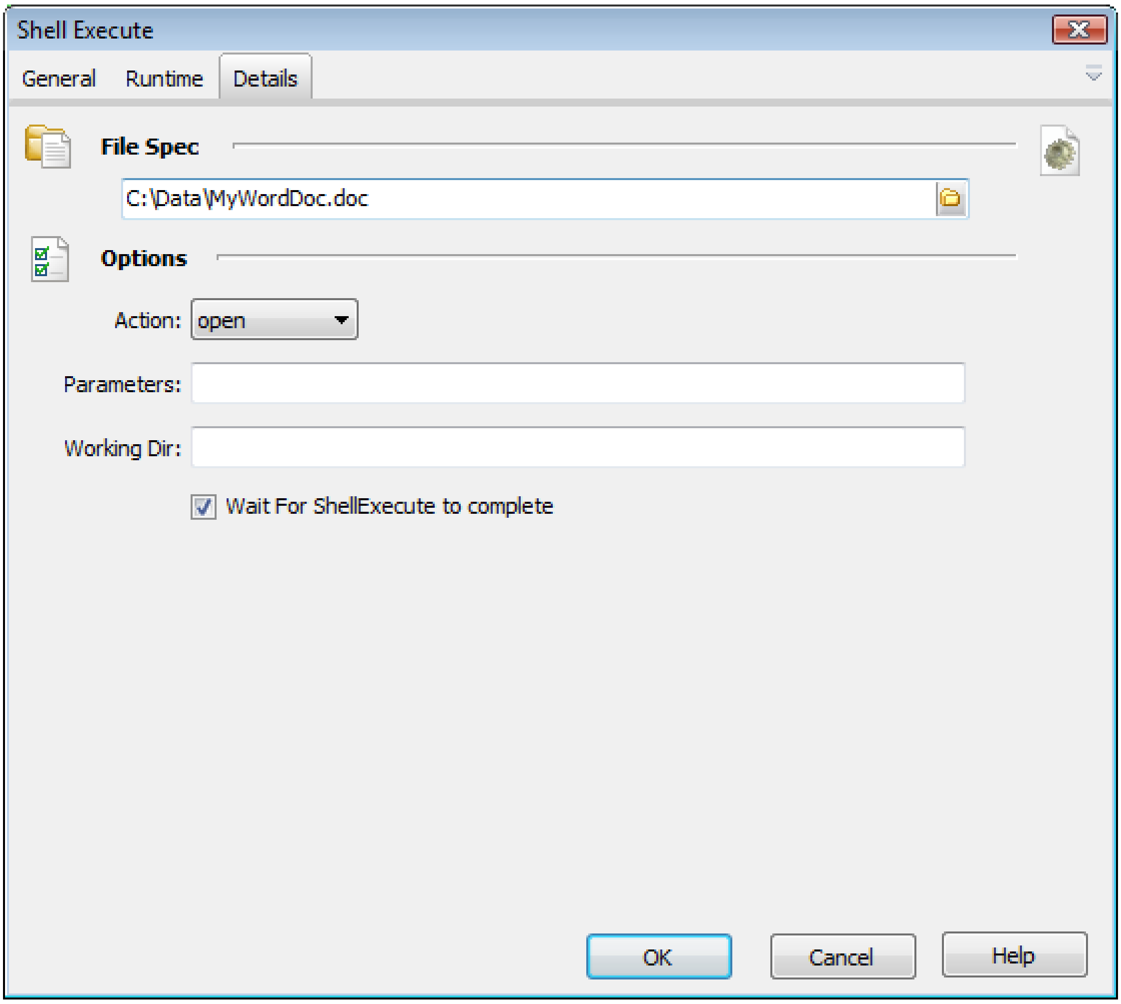
```
property  WaitForCompletion : WordBool;
property  ReturnCode : integer; //read only
property  HideWindow : WordBool;
property  Command : WideString;
 property EnableReturnCodeCheck : Boolean;
 property ReturnCodeComparator : TFBRunReturnCodeComparator;
 property ReturnCodeToCheck : Boolean;
```

These properties may be set in the BeforeAction and AfterAction script events.

6.27.22 Shell Execute

The Shell Execute action enables you to automate use the Windows Shell (Explorer) to act on a certain file.

For example, you could specify a .doc file and set the action to "print". This will ask the windows shell to print the document using the registered application for .doc files (eg. Word), see below:



Shell Execute

File Spec

C:\Data\MyWordDoc.doc

Options

Action: open

Parameters:

Working Dir:

☑ Wait For ShellExecute to complete

6.27.23 Sysinternals PsTools Suite

PsTools is a free suite of Windows administration utilities provided by SysInternals.Com.

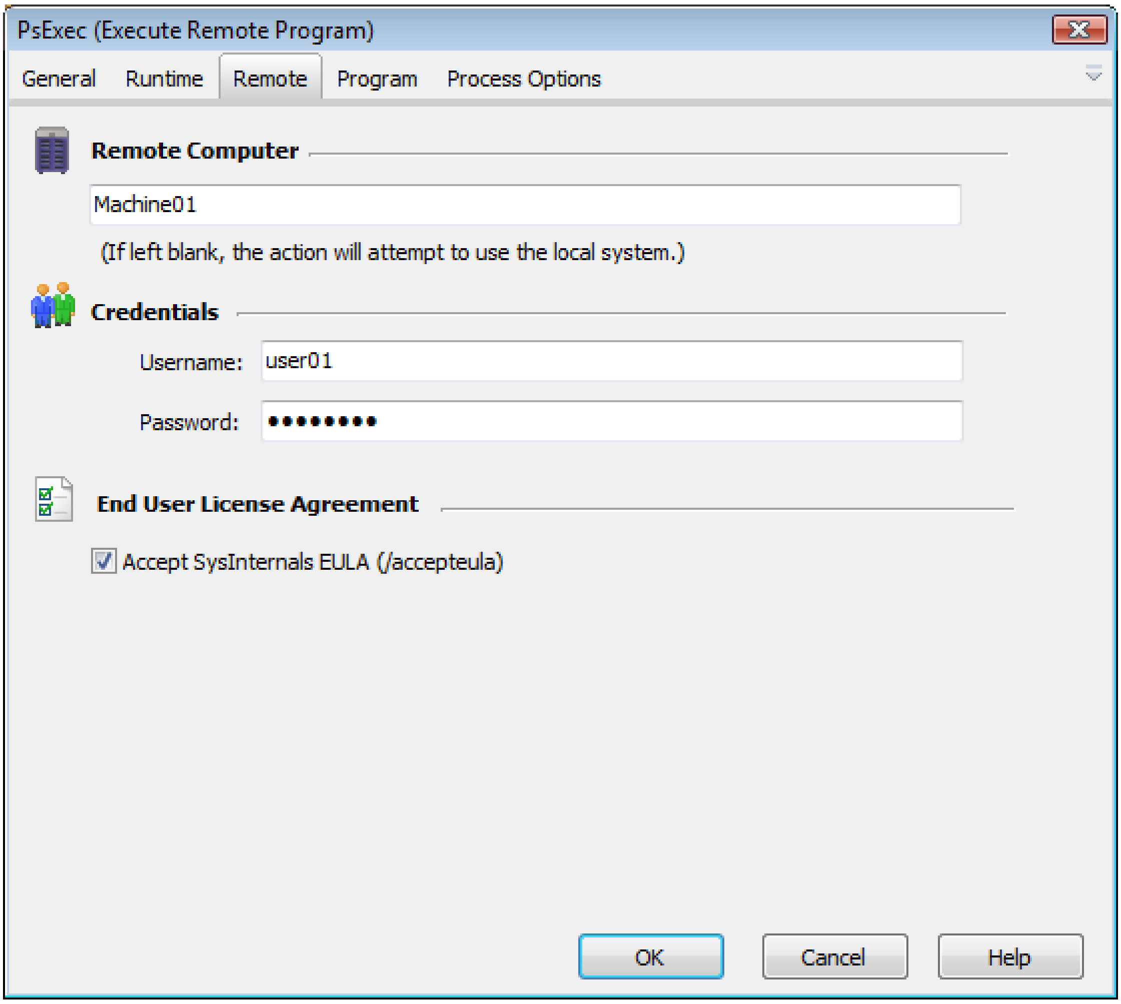
Automise provides full command-line automation for the PsTools suite

Global Options

PSTools Installed Directory
The location of the folder where the PSTools executables are stored.

Common Options

All of the PsTools actions can perform operations on remote computers:



Remote Computer

Machine01

(If left blank, the action will attempt to use the local system.)

Credentials

Username: user01

Password: ••••••••

End User License Agreement

☑ Accept SysInternals EULA (/accepteula)

Remote Computer

Enter the name of a remote computer on the network. If no name is supplied, the action will operate on the local machine.

Credentials

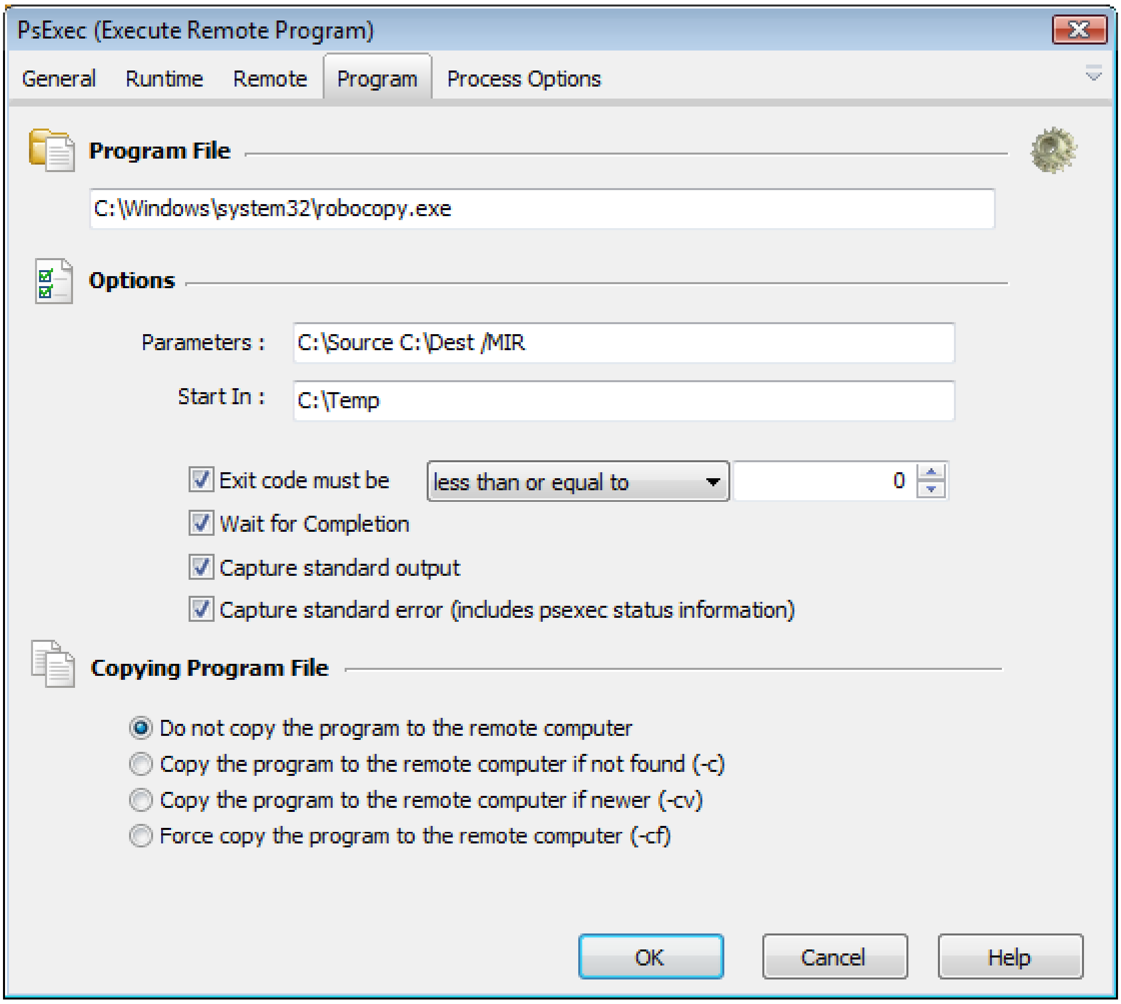
Enter a username and password combination. If no credentials are supplied, the action will attempt to connect using the credentials of the currently logged in user.

6.27.23.1 PsExec (Execute Remote Process)

PsExec allows you to execute a program on a remote computer. It is a more flexible alternative to the WMI Run Process action. PsExec can optionally create an interactive program on a user's desktop, can log output from the target program, and/or can obtain a return code.

"Remote" Tab

See the PsTools Common Options topic.



PsExec (Execute Remote Program)

Program File

C:\Windows\system32\robocopy.exe

Options

Parameters : C:\Source C:\Dest /MIR

Start In : C:\Temp

☑ Exit code must be less than or equal to 0

☑ Wait for Completion

☑ Capture standard output

☑ Capture standard error (includes psexec status information)

Copying Program File

● Do not copy the program to the remote computer

○ Copy the program to the remote computer if not found (-c)

○ Copy the program to the remote computer if newer (-cv)

○ Force copy the program to the remote computer (-cf)

Program File

Enter the path to the target executable (on the remote computer.)

Parameters

Enter any command line parameters for the target executable.

Start In

Enter the directory (on the remote computer) where you would like the process to start. Leave blank to use the directory which contains the program file.

Exit code must be

Specify a range of exit codes for the action to succeed. Note that this option is not available if "Wait For Completion" is enabled.

This option also includes the exit codes from PsExec itself. If PsExec fails to start the process, then it will return a PsExec exit code rather than an exit code from the process.

Wait For Completion

Enable this option for PsExec to not exit before the remote program has completed. If it is not enabled, the action will exit as soon as the remote program is running.

"Wait For Completion" must be enabled in order to log output from the remote action, or in order to receive exit codes from the remote program.

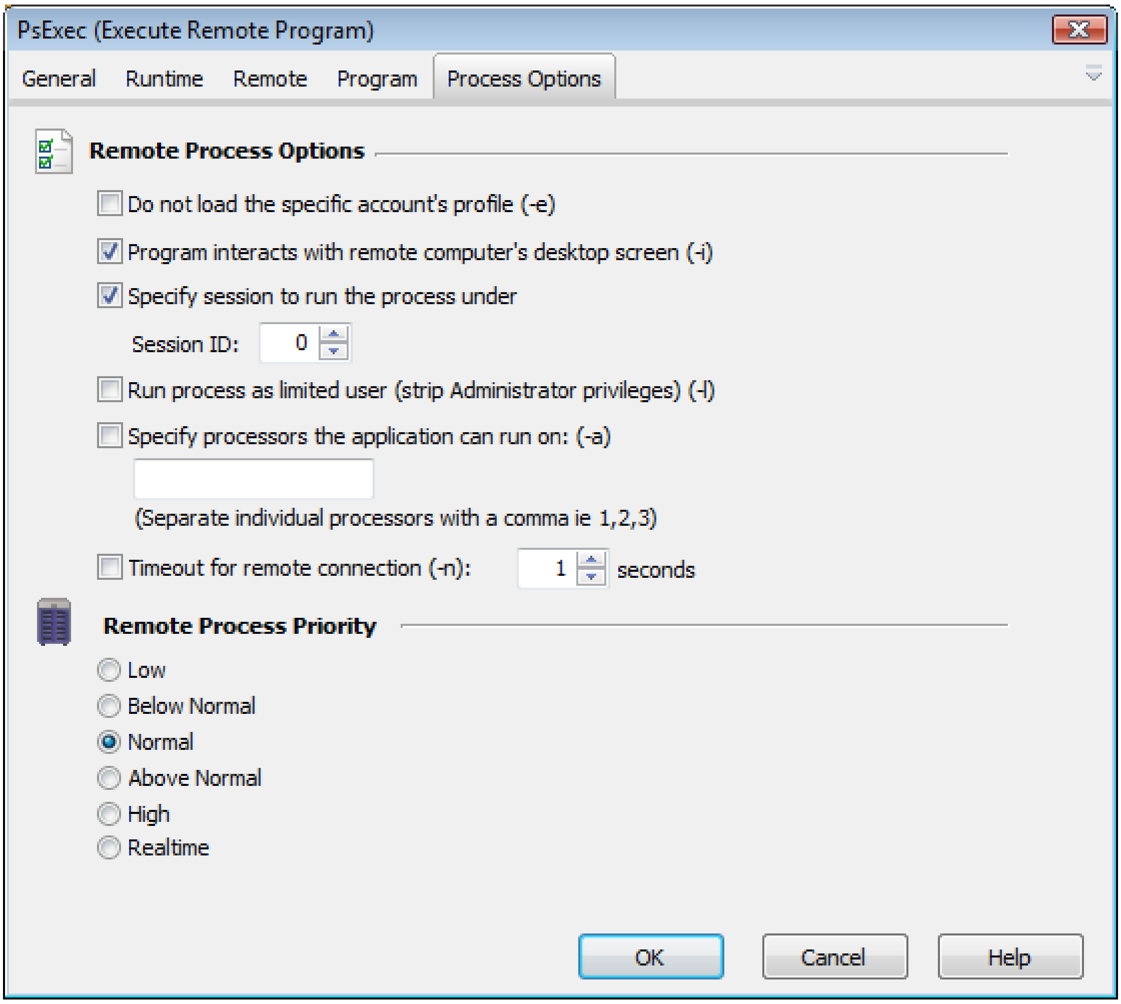
Capture standard output / Standard error

Enable one or both of these options in order to capture output from the remote program.

Note that because of the way in which PsExec outputs information, no "live" information is available before the program has ended. Also, information from Standard Output and Standard Error are not buffered in a chronological fashion - all of standard output is displayed in the log, followed by all of standard error (which includes PsExec's status output.)

Copying Program File

These options allow the target executable to be copied to the remote computer if it does not already exist. The program must exist in the local application PATH, or alternatively at the local pathname specified for "Program".



Remote Process Options

□ Do not load the specific account's profile (-e)

☑ Program interacts with remote computer's desktop screen (-i)

☑ Specify session to run the process under

Session ID: 0

□ Run process as limited user (strip Administrator privileges) (-l)

□ Specify processors the application can run on: (-a)

(Separate individual processors with a comma ie 1,2,3)

□ Timeout for remote connection (-n): 1 seconds

Remote Process Priority

○ Low
○ Below Normal
● Normal
○ Above Normal
○ High
○ Realtime

"Load the specific account's profile"

Enable this option to load account profile information (environment variables, etc.) before executing the program.

"Program interacts with remote computers' desktop screen"

Enable this option to make the remote application show up on the user's desktop. Otherwise, it will run invisibly in the background. Note that some graphical application require this option to be enabled in order to work properly.

"Specify session to run the process under"

When using the "Program interacts with remote computers' desktop screen" option, use this option to specify the ID of the interactive session to run the process within.

"Run process as limited user"

If this option is enabled, then even if the account specified on the "Remote" tab has Administrator privileges, the remote application will not.

"Specify processors the application can run on"

Use this option to specify Processor Affinity for the remote program. List processors by number, and separate individual processors with a comma.

"Timeout for remote connection"

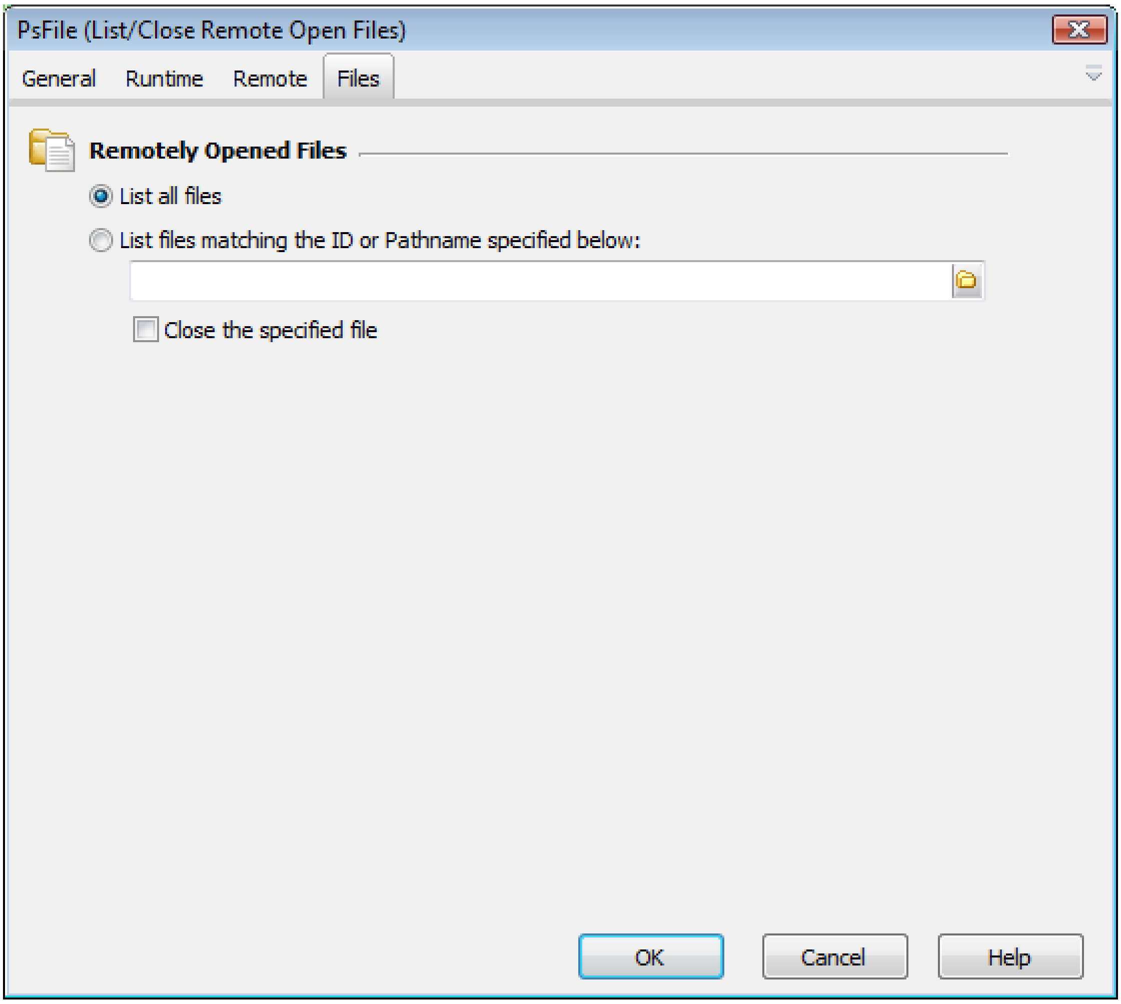
Specify a timeout length for network communications.

"Remote Process Priority"

Specify a priority level for the target process. Note that specifying above Normal process priority may make the remote system unresponsive, and Realtime priority has the potential to crash the remote system completely.

6.27.23.2 PsFile (List/Close Remote Files)

The PsFile command allows you to list files opened remotely on a computer, and optionally close them. See the PsFile topic in the PsTools help file for more details.



PsFile (List/Close Remote Open Files)

Remotely Opened Files

● List all files

○ List files matching the ID or Pathname specified below:

□ Close the specified file

"List all files"

Select this option to have PsFile list all open files on the specified computer.

"List files matching..."

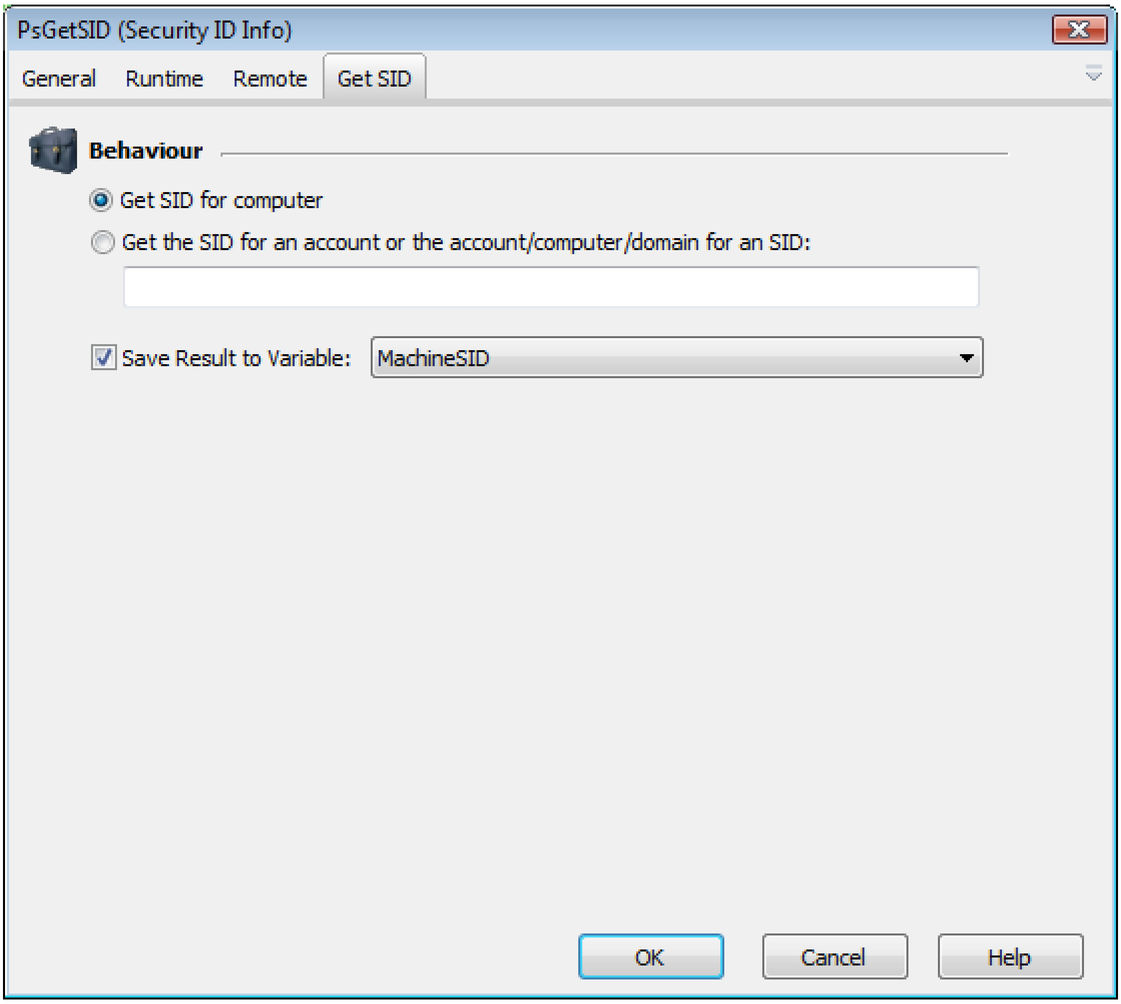
Enter a PsFile ID, or a (partial or complete) pathname, in order to just list certain files. Check the "**Close the specified file**" box to force-close any files which match.

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.3 PsGetSID (Security ID Info)

PsGetSID allows you to retrieve an SID for a computer, account or domain. You can also retrieve an account, computer or domain name from an SID. See the PsGetSID help topic in the PsTools help file for more details.



● Get SID for computer

○ Get the SID for an account or the account/computer/domain for an SID:

☑ Save Result to Variable: MachineSID

Behaviour

Choose "Get SID for computer" to retrieve the SID for the computer specified on the "Remote" tab.

Choose "Get the SID for..." to retrieve the SID for a specified account, computer or domain (in the above example, the Administrator account) or alternatively enter an SID to retrieve the associated account, computer or domain.

Save Result to Variable

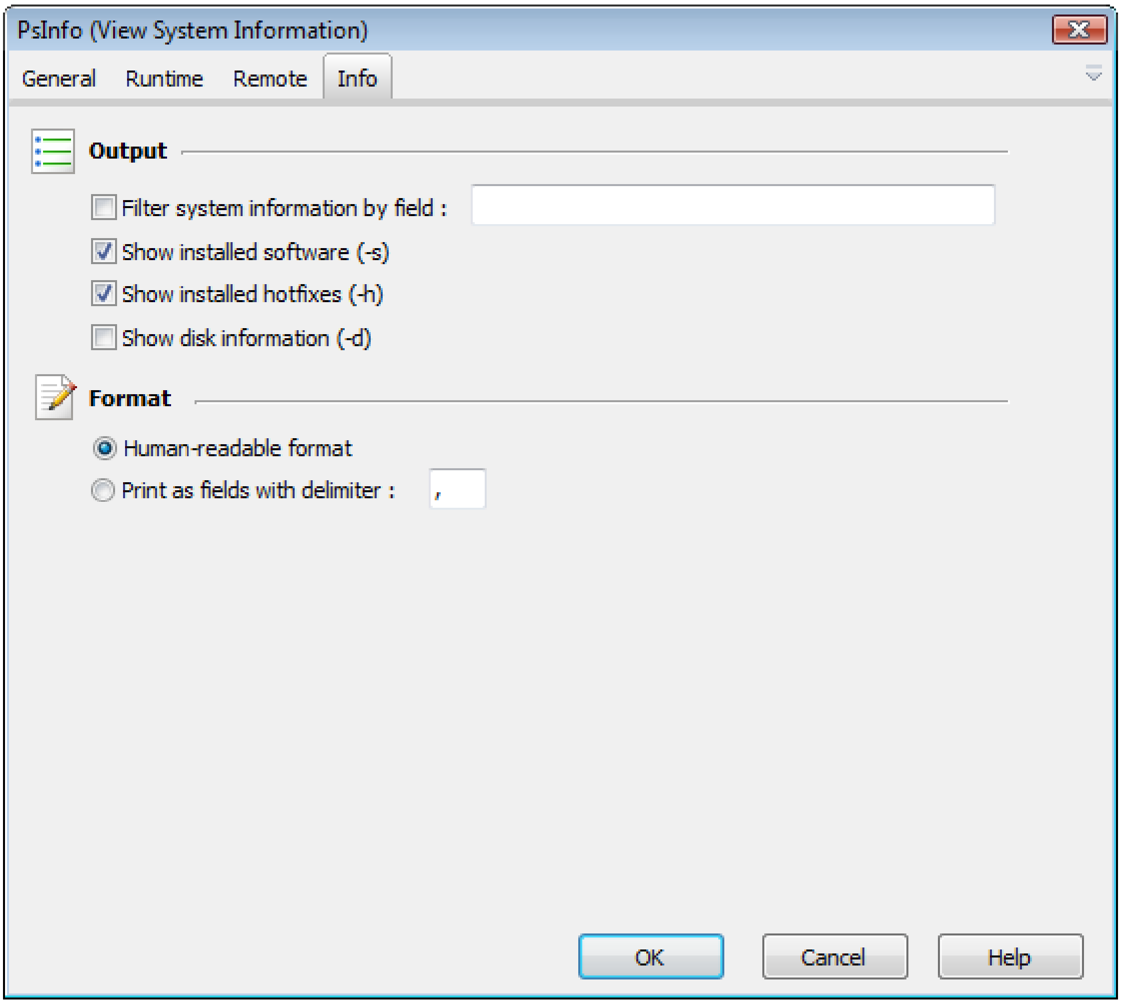
Check this option in order to save the SID (or account/domain/computer name) to a variable.

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.4 PsInfo (View System Information)

PsInfo logs information about the local (or remote) system. See the PsInfo help topic of the PsTools help for details.



Output

□ Filter system information by field :

☑ Show installed software (-s)

☑ Show installed hotfixes (-h)

□ Show disk information (-d)

Format

● Human-readable format

○ Print as fields with delimiter : ,

Filter system information by field...

Enter a field name (or part thereof) to only show information for that field.

Show installed software (-s)

Check this box to show a list of installed software packages.

Show installed hotfixes (-h)

Check this box to show a list of installed Windows hotfixes.

Show disk information (-d)

Check this box to show disk information (free space, files, etc.)

Format

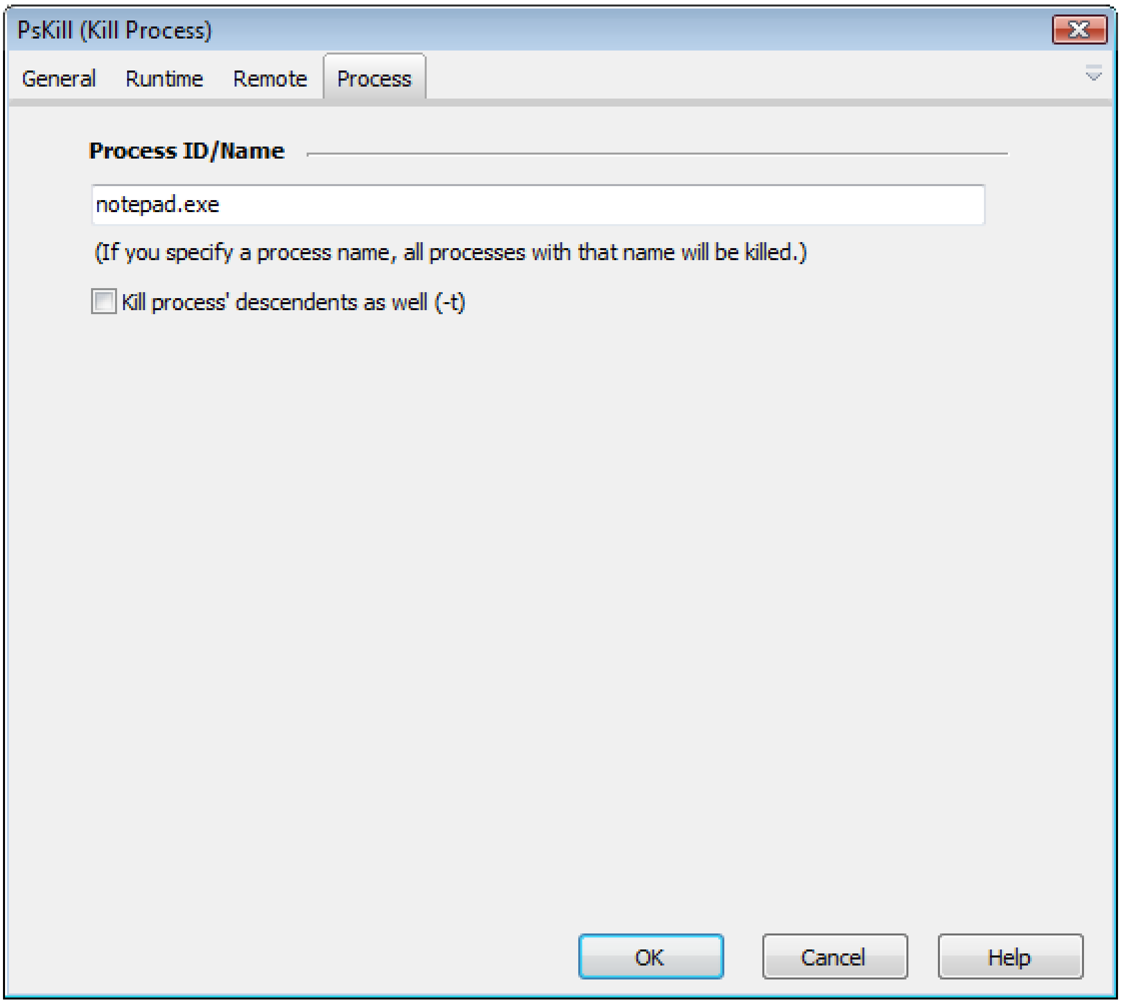
PsInfo output can be printed in human-readable format, or a delimited machine-readable fields (the default is comma-delimited.)

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.5 PsKill (Kill Process)

The PsKill utility allows you to forcibly terminate a running process, similarly to the WMI Kill Process action. See the PsKill help topic in the Pstools help file for more details.



Process ID/Name

notepad.exe

(If you specify a process name, all processes with that name will be killed.)

□ Kill process' descendents as well (-t)

Process ID/Name

Specify a process ID or name. Partial names can be given (ie in the example above, firefox will match firefox.exe or firefox-2.exe.)

Kill process' descendents as well (-t)

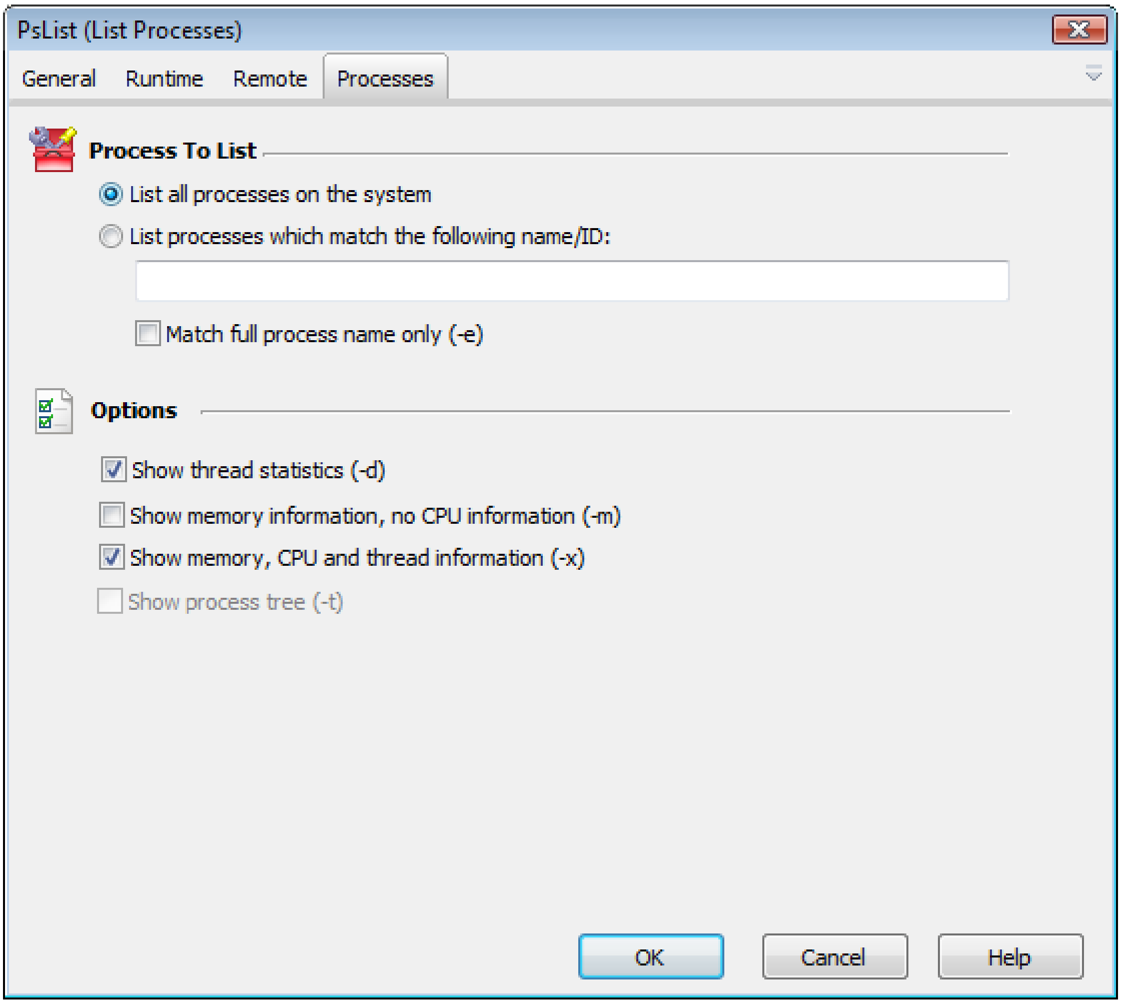
If this option is checked, PsKill will also kill the process' descendents in the process tree.

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.6 PsList (List Processes)

The PsList utility allows you to list processes on a local or remote computer. See the PsList topic in the PsTools help file for more details.



PsList (List Processes)

Process To List

● List all processes on the system

○ List processes which match the following name/ID:

□ Match full process name only (-e)

Options

☑ Show thread statistics (-d)

□ Show memory information, no CPU information (-m)

☑ Show memory, CPU and thread information (-x)

□ Show process tree (-t)

Processes To List

You can choose to list all processes on the system, or only those which match a certain Process ID or Process Name. Partial matches will be allowed, unless you choose the "Match full process name only" option (in which case only exact matches will be made.)

Show thread statistics (-d)

Check this option in order to show accounting information for each thread in each selected process.

Show memory information, no CPU information (-m)

If you choose this option, the default CPU information will not be printed. However, details of memory use will be printed.

Show memory, CPU and thread information (-x)

If you choose this option, all 3 pieces of information (CPU, thread and memory) will be printed for each process.

Show process tree (-t)

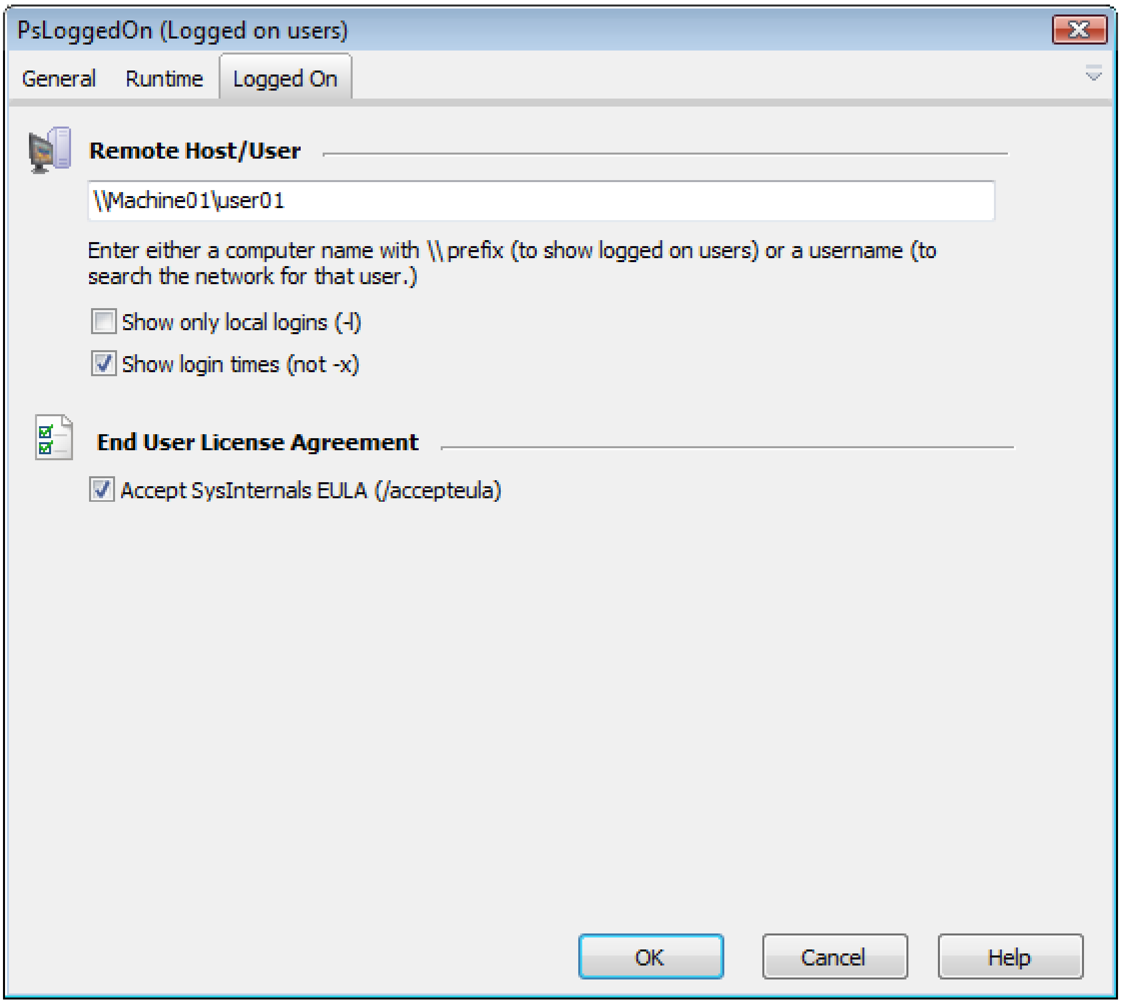
If you choose this option, the processes will be listed in a tree structure showing parent and child processes.

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.7 PsLoggedOn (Logged On Users)

The PsLoggedOn utility lists either all systems a certain user is logged into, or all users logged into a certain system. See the PsLoggedOn topic in the PsTools help file for more details.



Remote Host/User

\\Machine01\user01

Enter either a computer name with \\ prefix (to show logged on users) or a username (to search the network for that user.)

□ Show only local logins (-l)

☑ Show login times (not -x)

End User License Agreement

☑ Accept SysInternals EULA (/accepteula)

Remote Host/User

Specify either a remote host, ie

\\Computer Name

... in order to list all users logged into that host, or a username, ie

Vincent
OFFICE\Gary

... in order to list all computers on the network to which that user is logged in.

"Show only local logins (-l)"

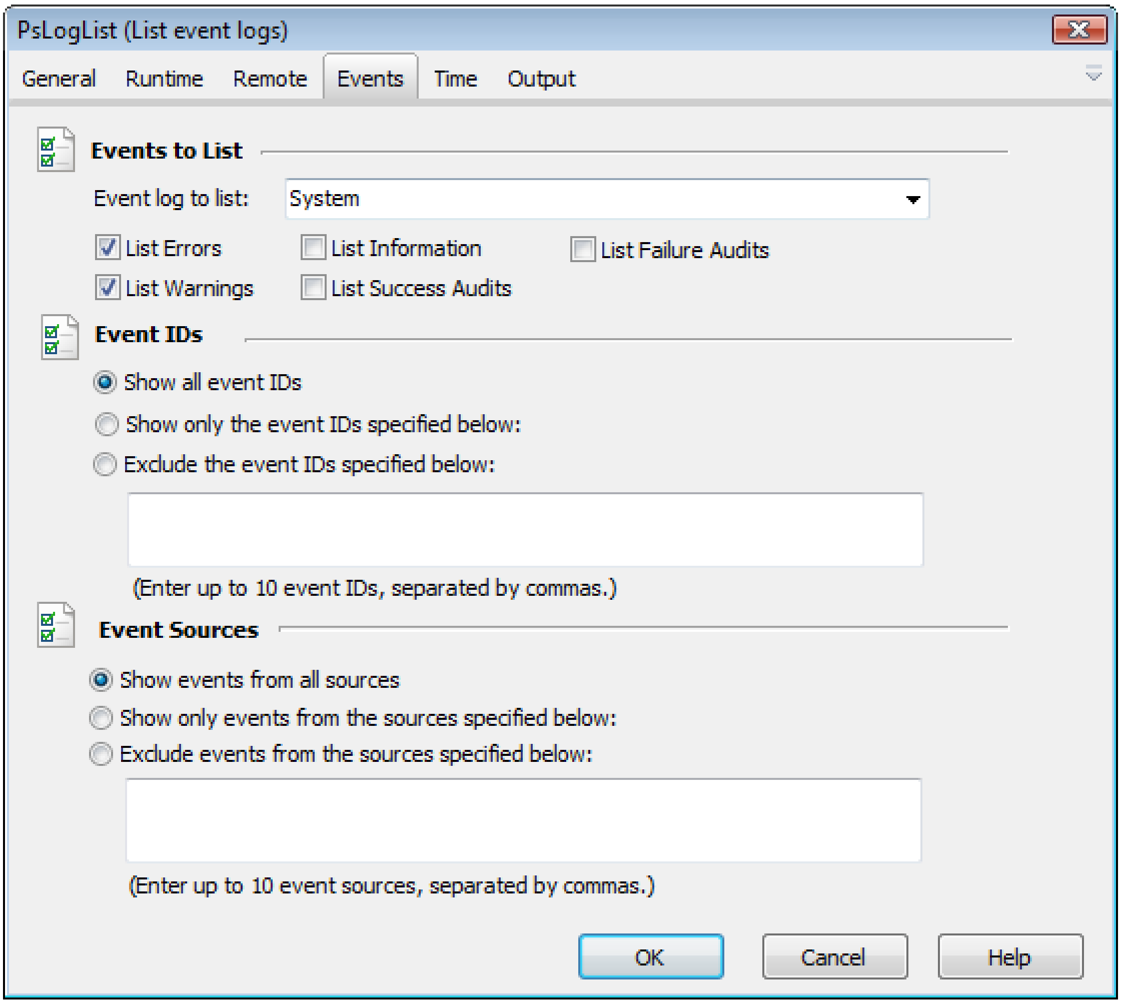
If this option is selected, PsLoggedOn only lists users that are logged in locally.

"Show login times (not -x)"

If this option is selected, PsLoggedOn shows login times as well as usernames.

6.27.23.8 PsLogList (List event logs)

PsLogList allows you to print a filtered section of a Windows Event Log. For full details, see the PsLogList help topic inside PsTools help.



PsLogList (List event logs)

Events to List

Event log to list: System

☑ List Errors ☐ List Information ☐ List Failure Audits

☑ List Warnings ☐ List Success Audits

Event IDs

● Show all event IDs

○ Show only the event IDs specified below:

○ Exclude the event IDs specified below:

(Enter up to 10 event IDs, separated by commas.)

Event Sources

● Show events from all sources

○ Show only events from the sources specified below:

○ Exclude events from the sources specified below:

(Enter up to 10 event sources, separated by commas.)

Events to List

Specify the name of an event log to list. The name can be a Windows event log (ie Application, Security or System), or an application-created Event Log which is viewable in the Windows Event Viewer administration tool.

List Errors / Warnings / Information / Success Audits / Failure Audits

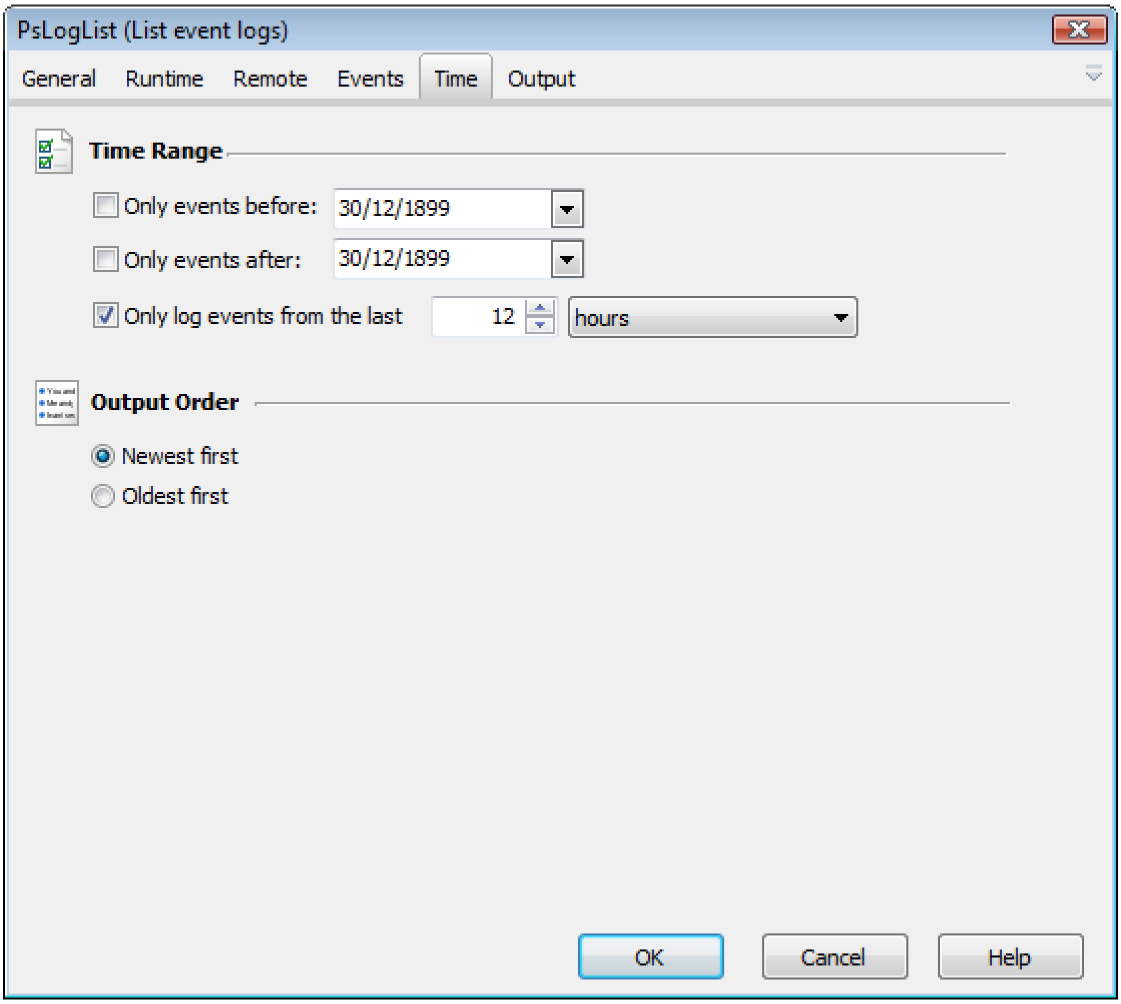
Specify the types of log messages you wish to be printed.

Event IDs

You can choose to show all Event IDs, or filter the events by ID. Filtering can either include or exclude the specified IDs. Enter up to 10 IDs in the event IDs field, separated by commas.

Event Sources

Similar to the Event IDs section, you can choose to show events from all sources, or filter events by specific sources (included or excluded.) Specify up to 10 event sources, separated by commas.



PsLogList (List event logs)

Time Range

□ Only events before: 30/12/1899

□ Only events after: 30/12/1899

☑ Only log events from the last 12 hours

Output Order

● Newest first

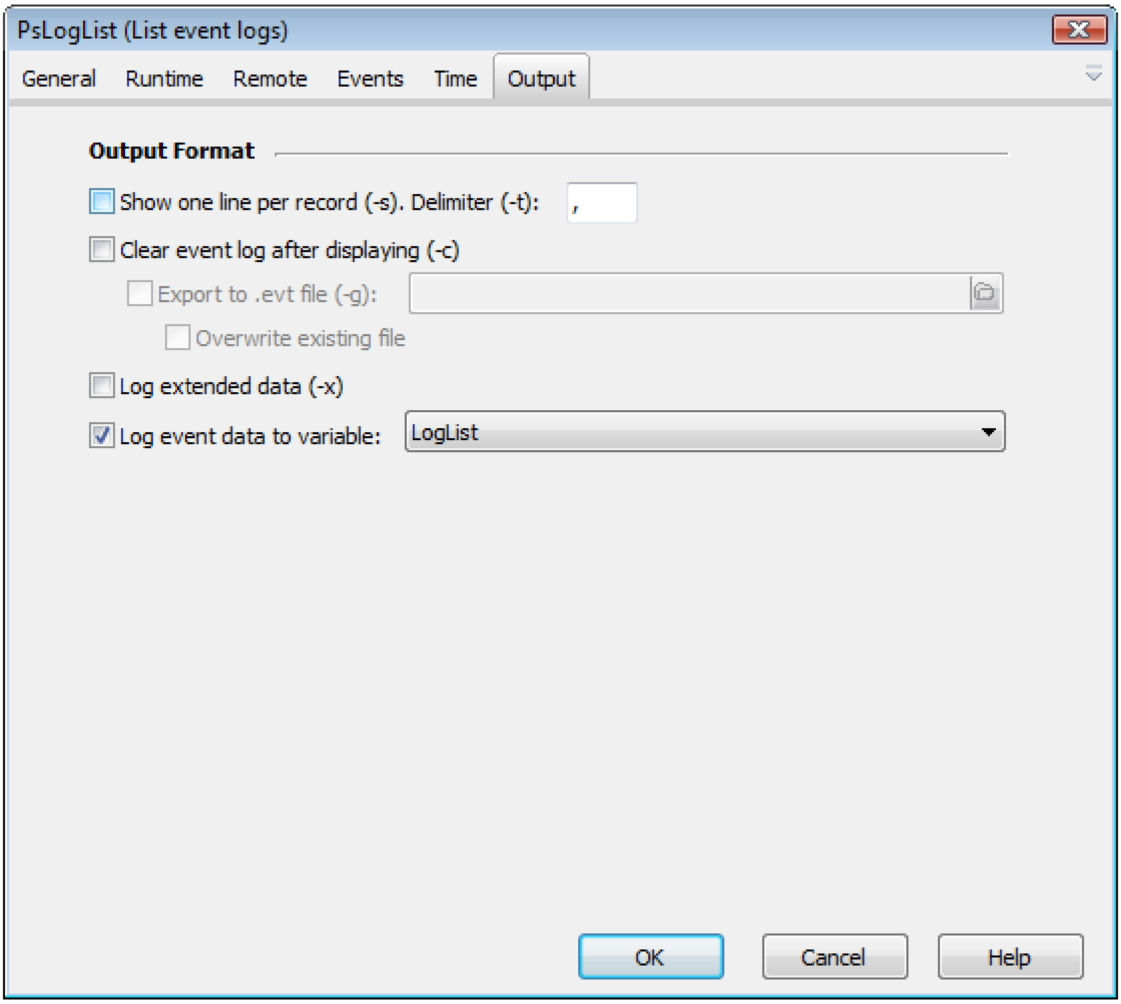
○ Oldest first

Time Range

You can choose to only log events within a certain date range, or within a certain period in the past.

Output Order

You can choose to list events from the newest to the oldest, or the other way around.



Output Format

□ Show one line per record (-s). Delimiter (-t): ,

□ Clear event log after displaying (-c)

□ Export to .evt file (-g):

□ Overwrite existing file

□ Log extended data (-x)

☑ Log event data to variable: LogList

Show one line per record (-s) with Delimiter (-t)

Outputs the event data in "machine readable" delimited field format (default is comma-delimited.)

Clear event log after displaying (-c)

If this option is selected, all of the entries which are selected will be removed from the log.

Export to .evt file (-g)

Enter the path to an .evt file to export the event data as it is removed.

Log extended data (-x)

Check this option to view extended information about each log entry.

Log event data to variable

Check this option to write the log data to a variable. This can be used in conjunction with the "Show one line per record" option to create a table of machine-readable event data which can be parsed by a script.

"Remote" Tab

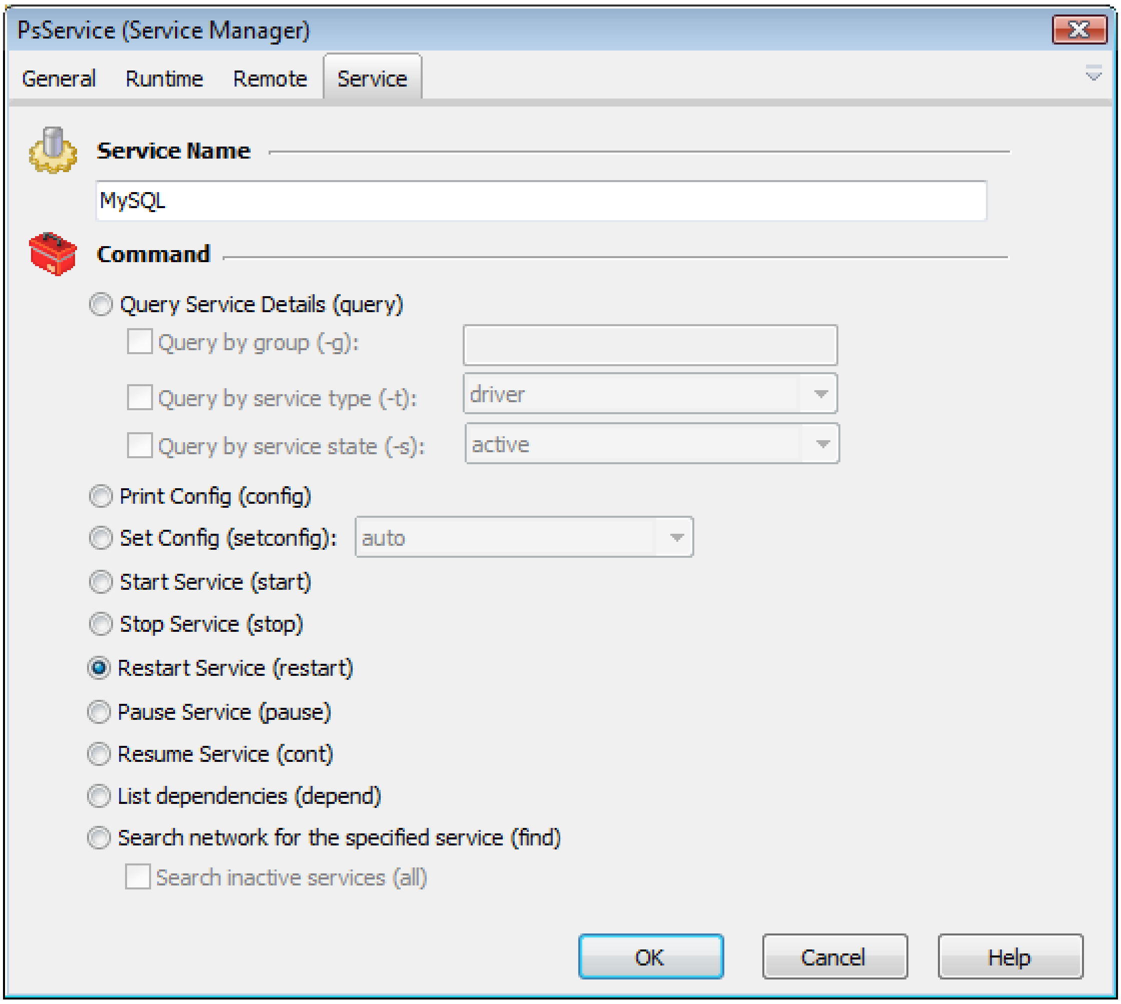
See the PsTools Common Options topic.

6.27.23.9 PsService (Service Manager)

The PsService utility allows you to perform operations on Windows Services. See the PsService topic in the PsTools help file for more details.

WARNING : Because PsService does not provide exit codes to calling programs, Automise is unable to determine success or failure. It is important to manually check PsService output for correct details (or parse the output in the OnStatusMessage script event.)

Consider using the Control Service Action as an alternative to PsService, if possible.



PsService (Service Manager)

Service Name

MySQL

Command

○ Query Service Details (query)
□ Query by group (-g):
□ Query by service type (-t): driver
□ Query by service state (-s): active
○ Print Config (config)
○ Set Config (setconfig): auto
○ Start Service (start)
○ Stop Service (stop)
● Restart Service (restart)
○ Pause Service (pause)
○ Resume Service (cont)
○ List dependencies (depend)
○ Search network for the specified service (find)
□ Search inactive services (all)

Query Service Details (query)

Use this command to output data about one or more services. Services can be queried by service name, group, service type and/or service state.

Print Config (config)

Use this command to log a service's current configuration details.

Set Config (setconfig)

Use this command to set a service to "auto", "demand" or "disabled" configuration state.

Start Service (start)

Use this command to start a stopped service.

Stop Service (stop)

Use this command to stop a running service.

Restart Service (restart)

Use this command to restart a running service.

Pause Service (pause)

Use this command to pause a running service.

Resume Service (cont)

Use this command to resume a paused service.

List Dependencies (depend)

Use this command to list a service's dependencies.

Search network for the specified service (find)

This command ignores any settings on the Remote tab. Instead, it searches all computers on the network for instances of the specified service and lists any which are found.

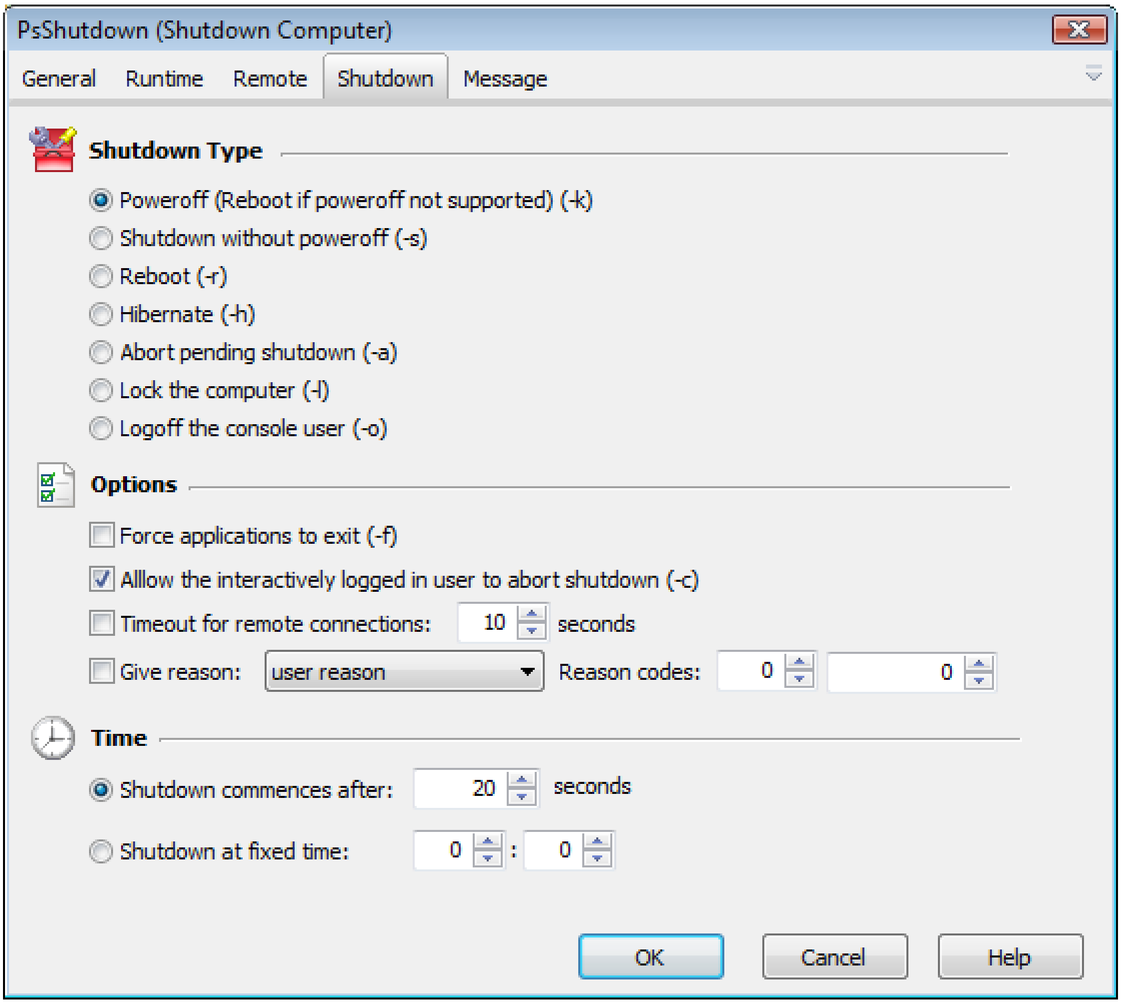
Check the "**Search inactive services (All)**" box to search for both running and inactive services (otherwise, PsService only searches for running instances of the service.)

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.10 PsShutdown (Shutdown Computer)

The PsShutdown utility allows you to shutdown, reboot, hibernate, lock or log a user off from a local or remote computer. In addition, it can be used to cancel an impending shutdown. See the PsShutdown help topic in the PsTools help file for details.



PsShutdown (Shutdown Computer)

Shutdown Type

● Poweroff (Reboot if poweroff not supported) (-k)
○ Shutdown without poweroff (-s)
○ Reboot (-r)
○ Hibernate (-h)
○ Abort pending shutdown (-a)
○ Lock the computer (-l)
○ Logoff the console user (-o)

Options

□ Force applications to exit (-f)
☑ Allow the interactively logged in user to abort shutdown (-c)
□ Timeout for remote connections: 10 seconds
□ Give reason: user reason Reason codes: 0 0

Time

● Shutdown commences after: 20 seconds
○ Shutdown at fixed time: 0 : 0

Shutdown Type:

- Power off / Shutdown without power off - Shut the machine down. (*)

- Reboot - Restart the machine. (*)

- Hibernate - Put the machine into hibernation, if supported. (*)

- Abort pending shutdown - Abort a shutdown which is currently pending.

- Lock the computer - Lock the local terminal and require a password before access is re-allowed.

- Logoff the console user - Log off the locally logged in user.

Options:

Force applications to exit (-f)

Check this box in order to forcibly terminate applications before shutting down. Only applicable to those shutdown types marked (*) above.

Allow the interactively logged in user to abort shutdown (-c)

Check this box to display a window on the local terminal allowing the local user to cancel shutdown. Only applicable to those shutdown types marked (*) above.

Timeout for remote connections

Optionally specify a timeout for all connections to a remote host.

Give reason

Optionally specify a "reason" for shutdown. Reasons are specified as "user reason" or "planned reason", as long as a major and a minor reason code. Only applicable to those shutdown types marked (*) above.

Time

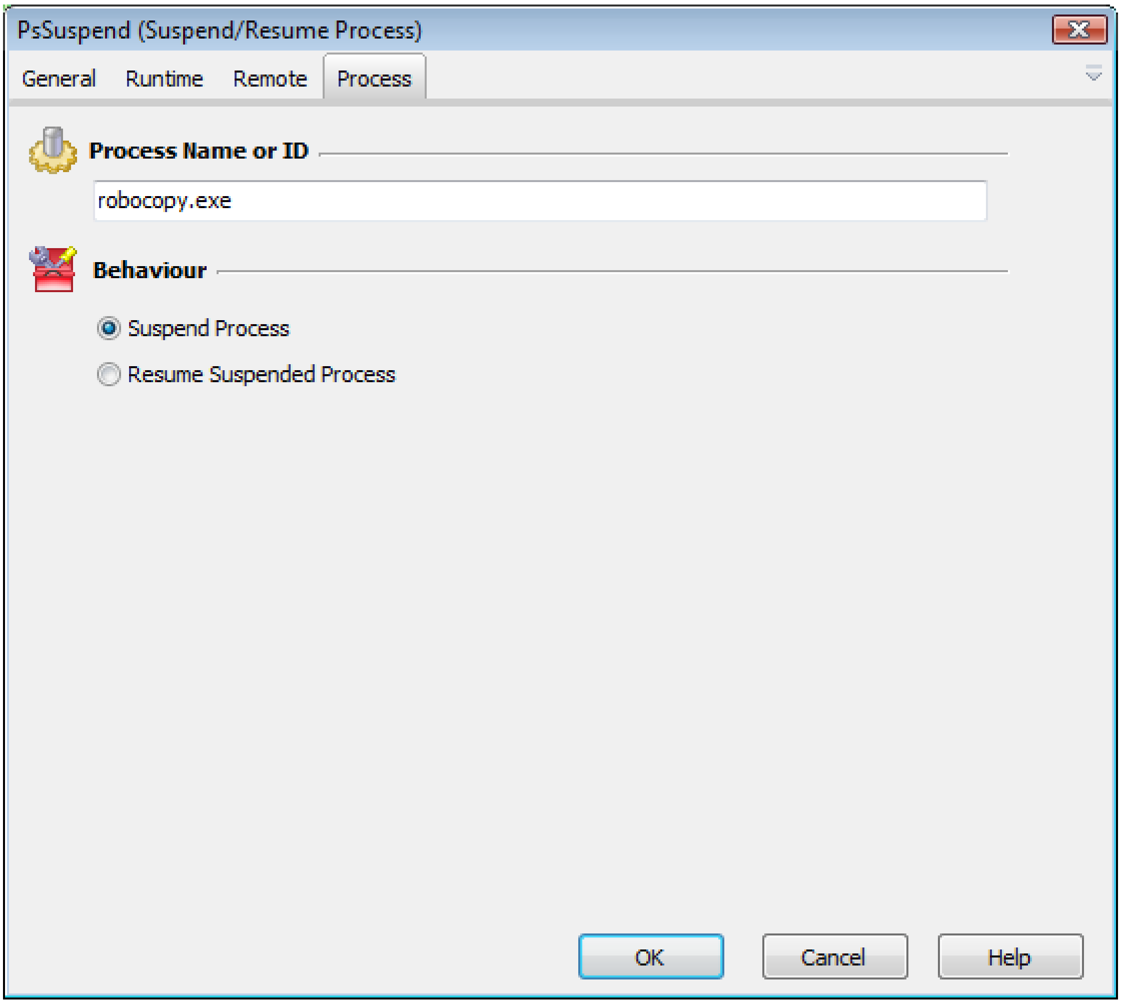
You can specify a delay (in seconds) before shutdown, or an absolute time (hours and minutes) at which to shutdown. Only applicable to those shutdown types marked (*) above.

"Remote" Tab

See the PsTools Common Options topic.

6.27.23.1‌1PsSuspend (Suspend/Resume Processes)

PsSuspend allows you to suspend and resume individual processes running on a system. Suspended processes will not be scheduled until they have been resumed. See the PsSuspend help topic in the PsTools help file for more details.



Process Name or ID

robocopy.exe

Behaviour

● Suspend Process

○ Resume Suspended Process

Process Name or ID

Enter a process name (or part thereof), or a process ID. If a name is specified, all matching processes will be suspended/resumed.

Behaviour

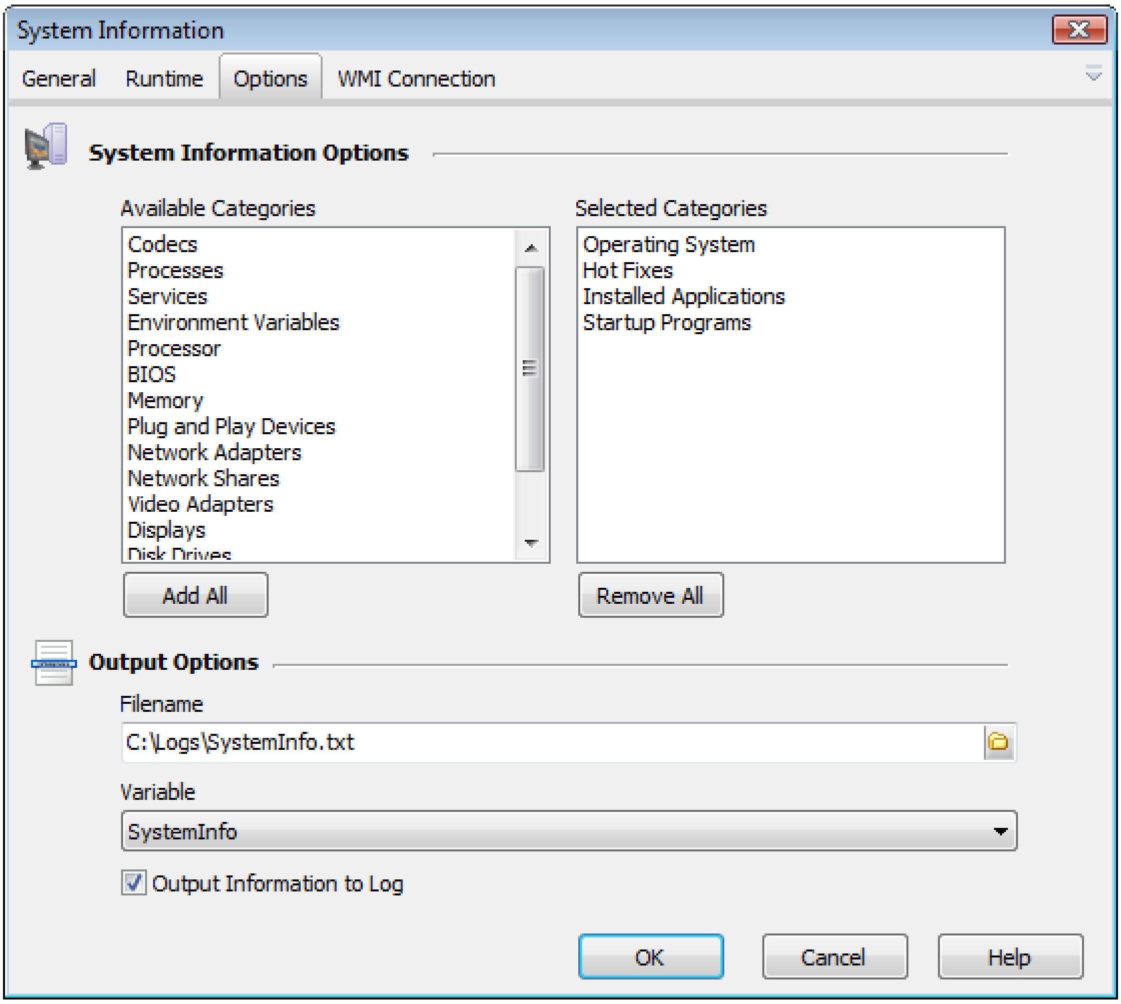
Choose to either suspend or resume the process(es).

"Remote" Tab

See the PsTools Common Options topic.

6.27.24 System Information

The System Information action allows you to retrieve information about the configuration of a windows based computer system.



System Information Options

Available Categories: Codecs, Processes, Services, Environment Variables, Processor, BIOS, Memory, Plug and Play Devices, Network Adapters, Network Shares, Video Adapters, Displays, Disk Drives

Selected Categories: Operating System, Hot Fixes, Installed Applications, Startup Programs

Output Options

Filename: C:\Logs\SystemInfo.txt

Variable: SystemInfo

☑ Output Information to Log

Categories
A list of all the available information that can be retrieved. Categories can be moved to the selected list by selecting them, or by clicking the 'Add All' button which will add all the categories to the selected list.

Filename
The location of the text file to output the system information to.

Variable
The name of the variable which will hold the system information.

6.27.25 System Restore

The System Restore actions allow you to perform various functions against the Windows System Restore services.

The available actions are:
- System Restore Enable
- System Restore Disable
- System Restore Create Restore Point
- System Restore perform Restore
- System Restore Get Last Restore Status
- System Restore Get Configuration
- System Restore Set Configuration
- System Restore List Restore Points

System Restore is only available on Windows Vista, Windows XP, and Windows Me.

For more information on System Restore, see:
http://msdn2.microsoft.com/en-us/library/aa378979.aspx
http://www.microsoft.com/technet/scriptcenter/scripts/desktop/restore/default.mspx

Excerpt from MSDN on System Restore:

As the computer is used over time, restore points are collected in the data archive without any management or intervention required by the user. If the user ever needs to restore the system to a previous state, the available restore points are made visible to the user through the System Restore user interface. The user can choose any of these restore points. The only way to access this archive of restore points is through the System Restore user interface and the System Restore API; this is to protect data integrity and prevent accidental changes made by the user, applications, or other agents.

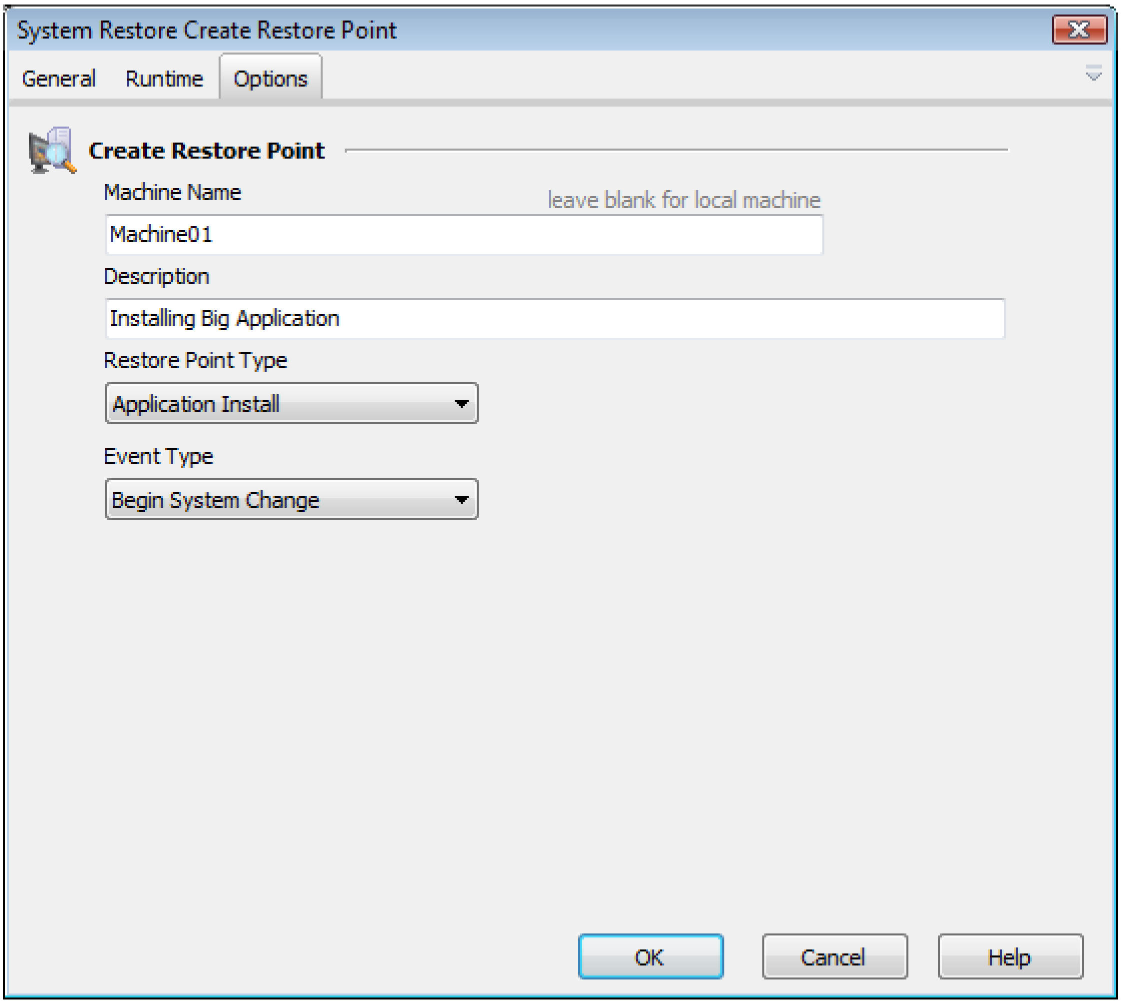
To restore a system, System Restore undoes file changes made to monitored files, recapturing the file state at the time of the selected restore point. It then replaces the current registry with the one saved for the selected restore point.

To ensure that your application has the desired behavior after a restore, do the following:

- Ensure that key application binaries to be protected use extensions consistent with those used in Filelist.xml. For more information, see Monitored File Extensions.
- Do not use monitored extension types for user-editable files. For example, if you name a user's personal data file using the extension .ini, the user may lose work as a result of a system restore.
- Do not store information in the registry that prevents user access to personal data files or applications on system restore. Otherwise, you must provide a mechanism by which the user can download and reinstall the applications without having to pay for them again.
- Use the System Restore API to create meaningful restore points at install and uninstall.

6.27.25.1 System Restore Create Restore Point

The System Restore Create Restore Point Action will create a new System Restore Point on the local machine or a remote machine.


System Restore Create Restore Point
General
Runtime
Options
Create Restore Point
leave blank for local machine
OK
Cancel
Help

Machine Name: Machine01

Description: Installing Big Application

Restore Point Type: Application Install

Event Type: Begin System Change

Machine Name - enter the machine name for a remote machine, or leave blank for the local machine

Description - enter the description that the end user will be able to see in the System Restore control panel applet.

Restore Point Type - select the type of restore point you are creating

Event Type - select the event type.

NOTES:

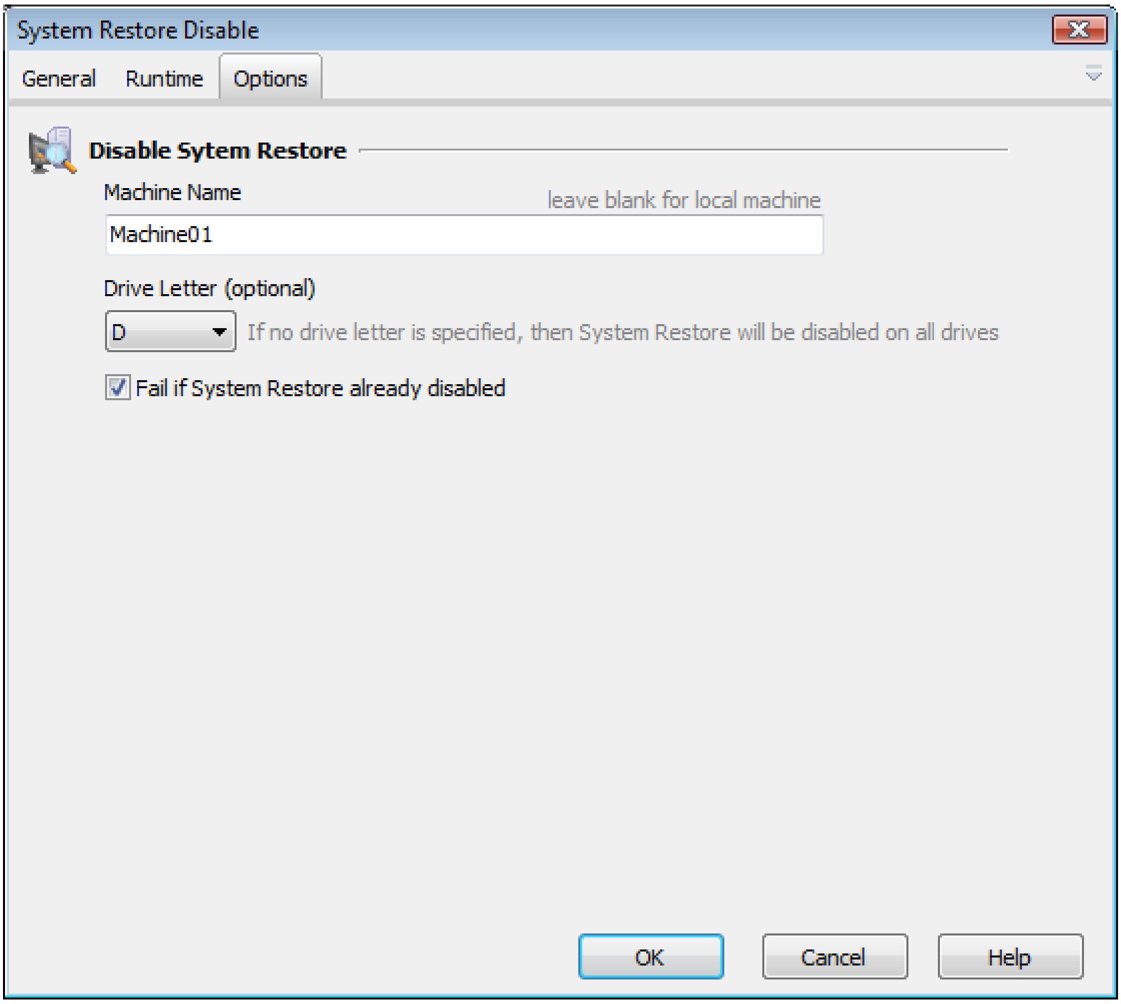
An application can create a restore point before it causes a significant system change, such as an install, uninstall, or feature update.

Installers should create a restore point just prior to installation with the Event Type set to Begin System Change. To notify System Restore that the installation has been completed, use another Create Restore Point action with the Event Type set to End System Change.

If the user cancels the application installation, the installer may remove the restore point it created when the installation began. Removing the restore point is optional and can prevent the user from recovering from unintentional changes made by the installer during the cancellation. To remove a restore point, use a Create Restore Point action with the Restore Point Type set to Cancelled Operation and the Event Type set to End System Change.

6.27.25.2 System Restore Disable

The System Restore Disable Action will disable System Restore on the local machine or a remote machine.



Disable Sytem Restore

Machine Name: Machine01 (leave blank for local machine)

Drive Letter (optional): D (If no drive letter is specified, then System Restore will be disabled on all drives)

☑ Fail if System Restore already disabled

Machine Name - enter the machine name for a remote machine, or leave blank for the local machine

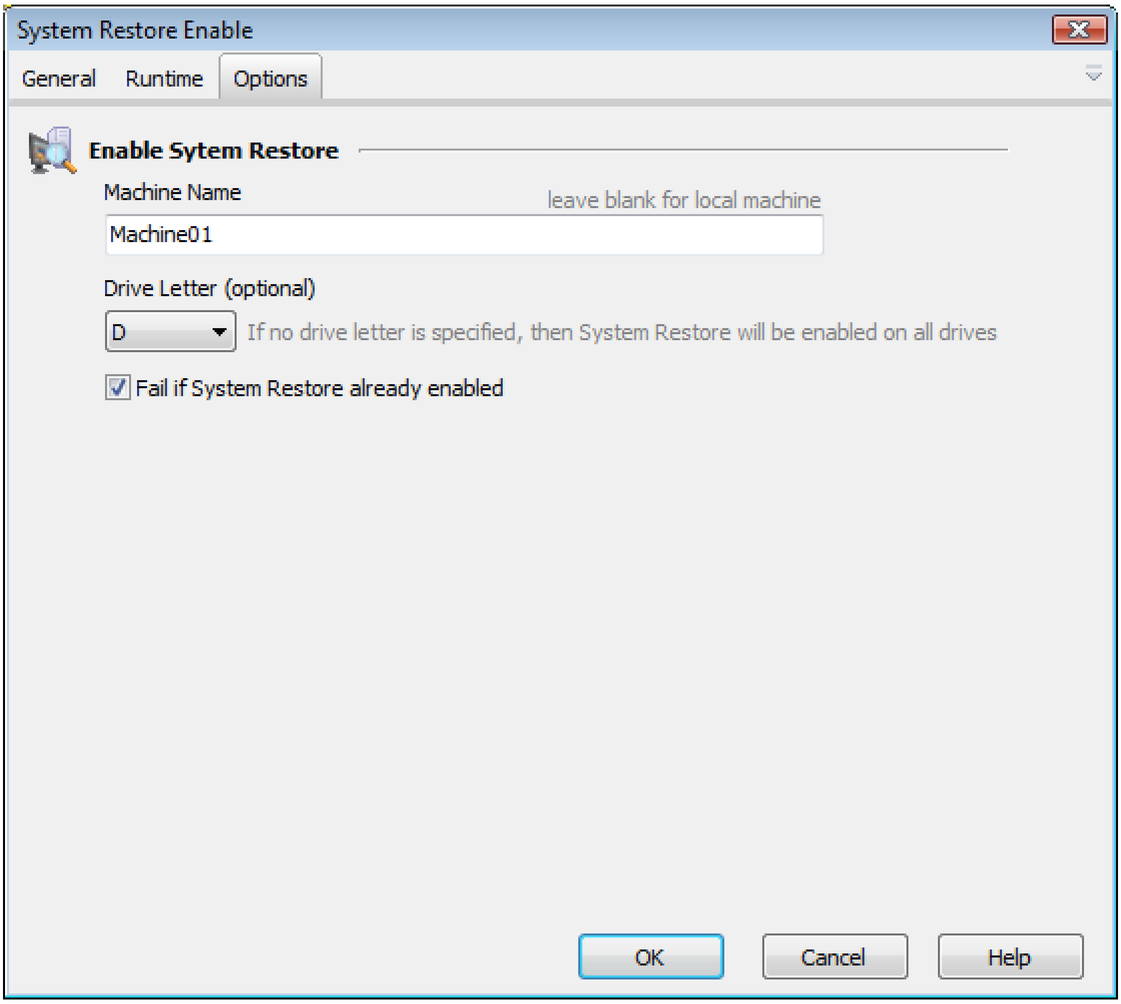
Drive Letter - the System Restore API allows for specifying a drive letter, but during our testing this seems to be ignored. We suggest you leave the drive letter blank, and then System Restore will be disabled on the entire machine.

Fail if System Restore already disabled - the action will fail if System Restore is already disabled

NOTES : After disabling System Restore, the action returns immediately but the target machine may take some time before System Restore is fully disabled. It is advised to wait a few seconds before performing any of the other System Restore actions

6.27.25.3 System Restore Enable

The System Restore Enable Action will enable System Restore on the local machine or a remote machine.



Enable Sytem Restore

Machine Name: Machine01 (leave blank for local machine)

Drive Letter (optional): D (If no drive letter is specified, then System Restore will be enabled on all drives)

☑ Fail if System Restore already enabled

Machine Name - enter the machine name for a remote machine, or leave blank for the local machine

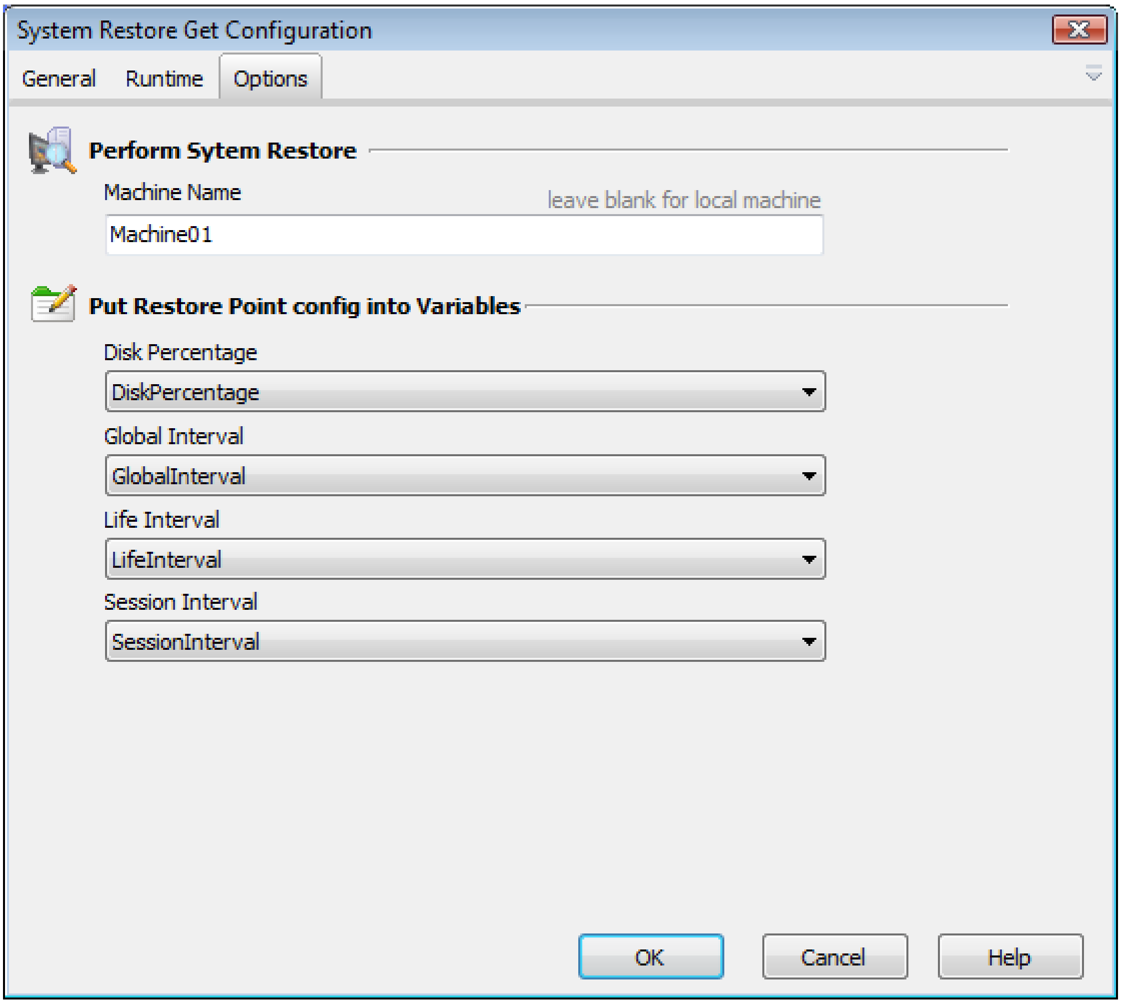
Drive Letter - the System Restore API allows for specifying a drive letter, but during our testing this seems to be ignored. We suggest you leave the drive letter blank, and then System Restore will be enabled on the entire machine.

Fail if System Restore already enabled - the action will fail if System Restore is already enabled

NOTES : After enabling System Restore, the action returns immediately but the target machine may take some time before System Restore is fully functional. It is advised to wait a few seconds before performing any of the other System Restore actions

6.27.25.4 System Restore Get Configuration

The System Restore Get Configuration Action will get the machine System Restore configuration.



Perform Sytem Restore

Machine Name: Machine01 (leave blank for local machine)

Put Restore Point config into Variables

Disk Percentage: DiskPercentage

Global Interval: GlobalInterval

Life Interval: LifeInterval

Session Interval: SessionInterval

Machine Name - specify the machine name or leave blank for local

Disk Percentage - choose a variable to set the Disk Percentage that System Restore is set to use.

Global Interval - choose a variable to set the Global Interval that System Restore is set to use.

Life Interval - choose a variable to set the Life Interval that System Restore is set to use.

Session Interval - choose a variable to set the Session Interval that System Restore is set to use.

NOTES:

Disk Percent (0 - 100)
The maximum amount of disk space on each drive that can be used by System Restore. This value is specified as a percentage of the total drive space. The default value is 12 percent.

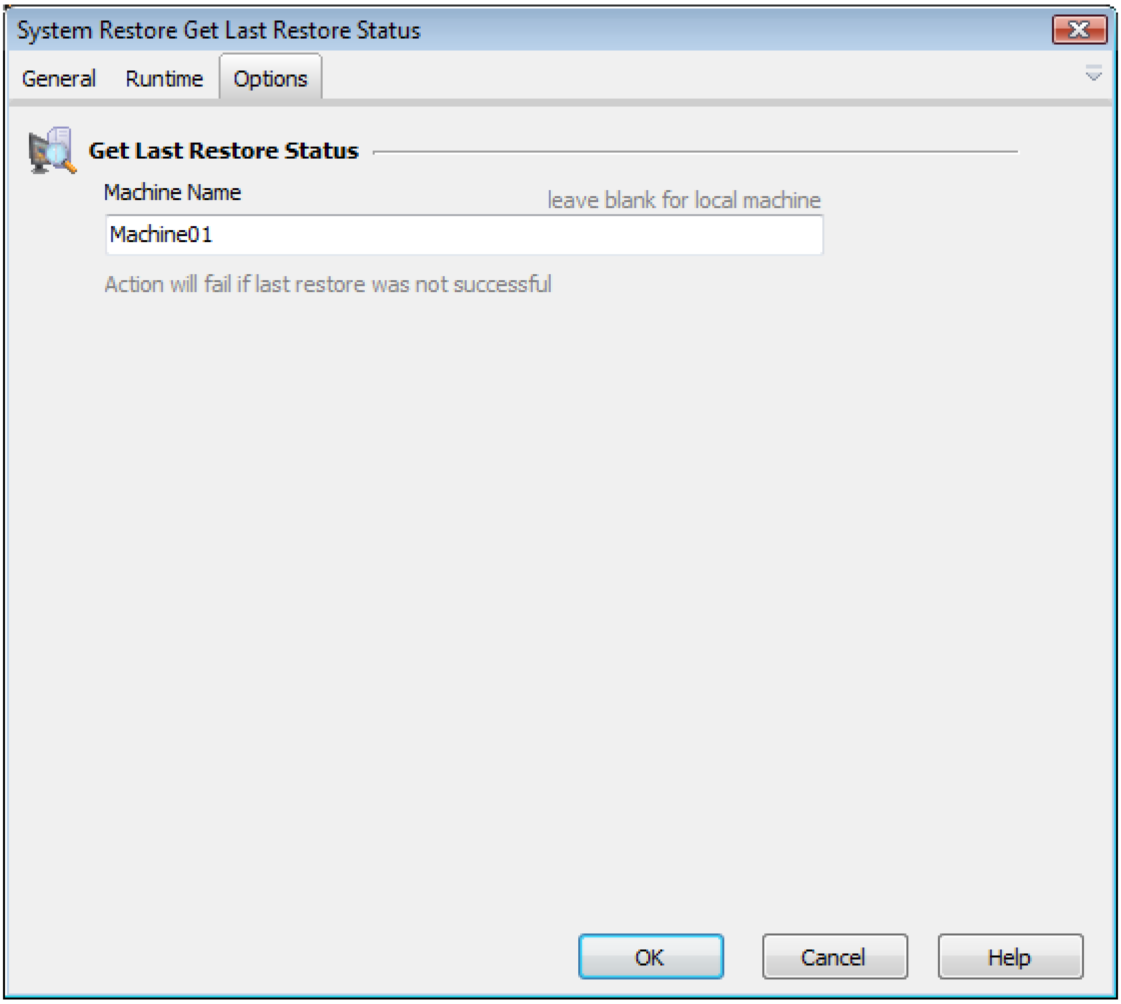
Global Interval (Unsigned 32bit integer)
The absolute time interval at which scheduled system checkpoints are created, in seconds. The default value is 86,400 (24 hours).

Life Interval (Unsigned 32bit integer)
The time interval for which restore points are preserved, in seconds. When a restore point becomes older than this specified interval, it is deleted. The default age limit is 90 days.

Session Interval (Unsigned 32bit integer)
The time interval at which scheduled system checkpoints are created during the session, in seconds. The default value is zero, indicating that the feature is turned off.

6.27.25.5 System Restore Get Last Restore Status

The System Restore Get Last Restore Status will fail if the last System Restore was unsuccessful.



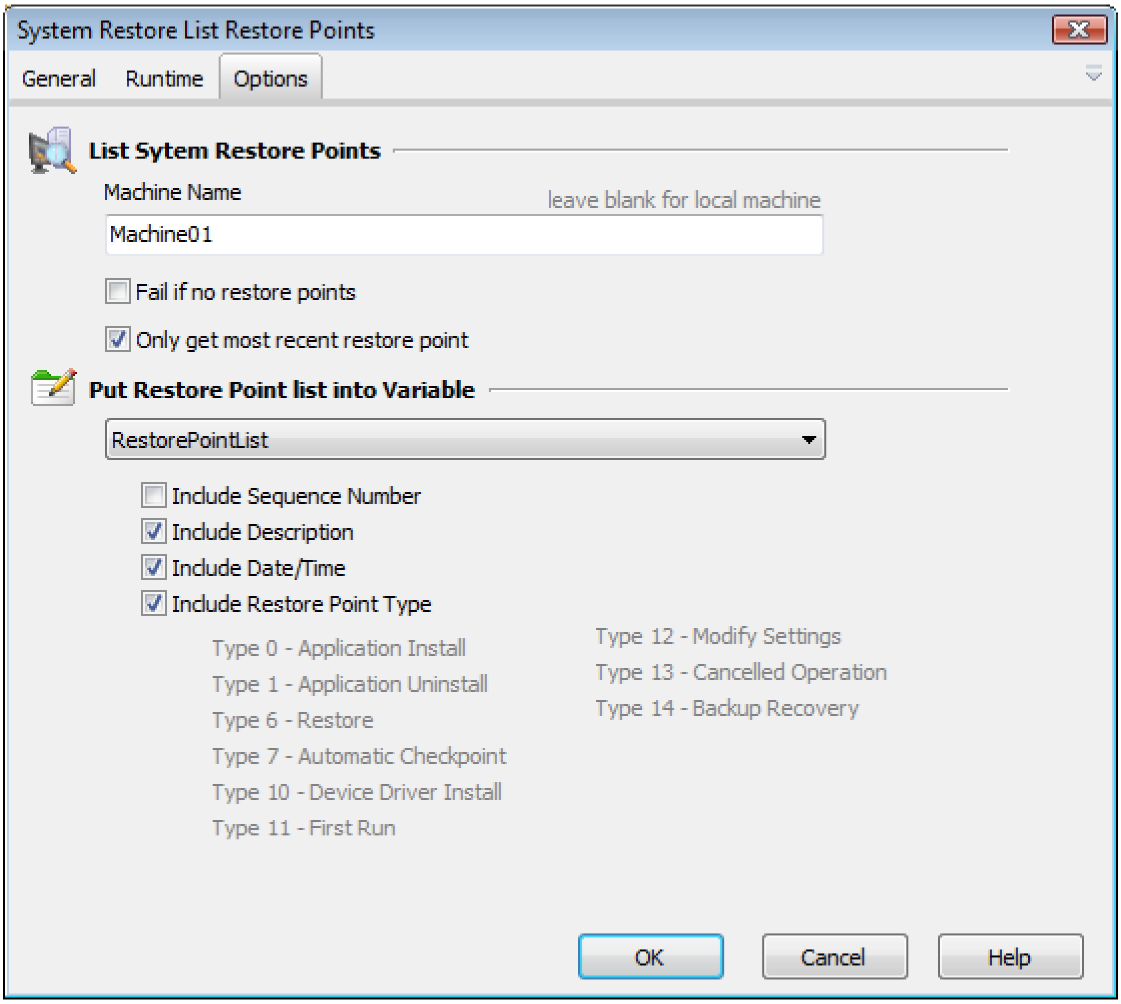
Machine Name: Machine01
leave blank for local machine
Action will fail if last restore was not successful

Machine Name - specify the remote machine, or leave blank for local machine.

NOTES:
The action will fail if the last restore status was unsuccessful

6.27.25.6 System Restore List Restore Points

The System Restore List Restore Points action will retrieve a list of the Restore Points on the specified machine.



List Sytem Restore Points

Machine Name — leave blank for local machine

Machine01

☐ Fail if no restore points

☑ Only get most recent restore point

Put Restore Point list into Variable

RestorePointList

☐ Include Sequence Number

☑ Include Description

☑ Include Date/Time

☑ Include Restore Point Type

Type 0 - Application Install
Type 1 - Application Uninstall
Type 6 - Restore
Type 7 - Automatic Checkpoint
Type 10 - Device Driver Install
Type 11 - First Run
Type 12 - Modify Settings
Type 13 - Cancelled Operation
Type 14 - Backup Recovery

Machine Name - specify the machine or leave blank for the local machine

Fail if no restore points - the action will fail if no restore points are found

Only get most recent restore point - only the most recent restore point will be put in the log and set in the specified variable

Put Restore Point list into Variable - if you need to further process the restore point list, then use a variable so that the list can be used in other actions

Include Sequence Number - choose if the sequence number should be listed (the sequence number is used to perform a restore to this particular restore point)

Include Description - choose if the restore point description should be listed

Include Date/Time - choose if the date/time should be listed

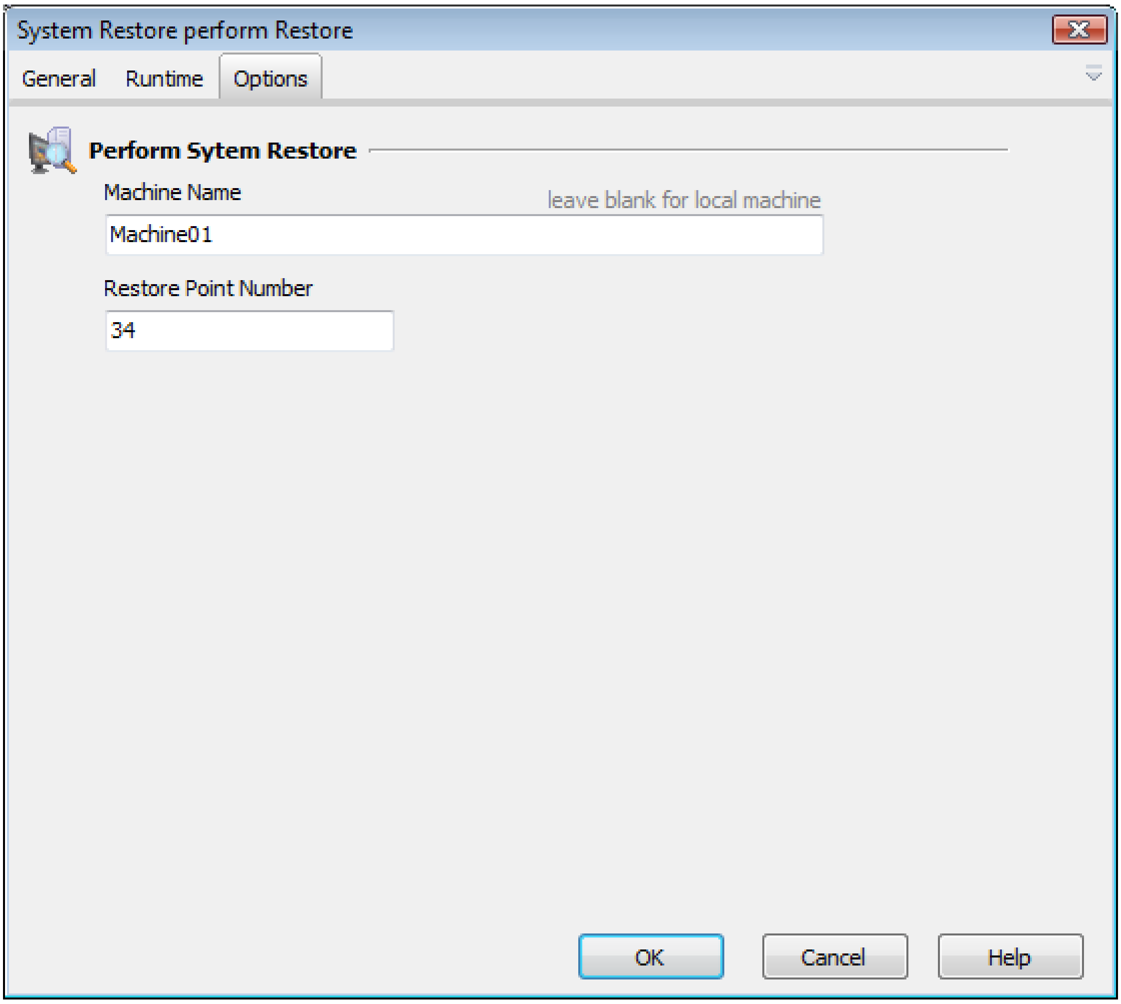
Include Restore Point Type - choose if the restore point type should be listed

NOTES:
The known Restore Point Types are:
0 - Application Install
1 - Application Uninstall

6 - Restore
7 - Automatic Checkpoint
10 - Device Drive Install
11 - First Run
12 - Modify Settings
13 - Cancelled Operation
14 - Backup Recovery

6.27.25.7 System Restore perform Restore

The System Restore perform Restore action will initiate a System Restore on the specified machine. A reboot must be performed, as the actual restoration takes place during the reboot.



Machine Name: Machine01

Restore Point Number: 34

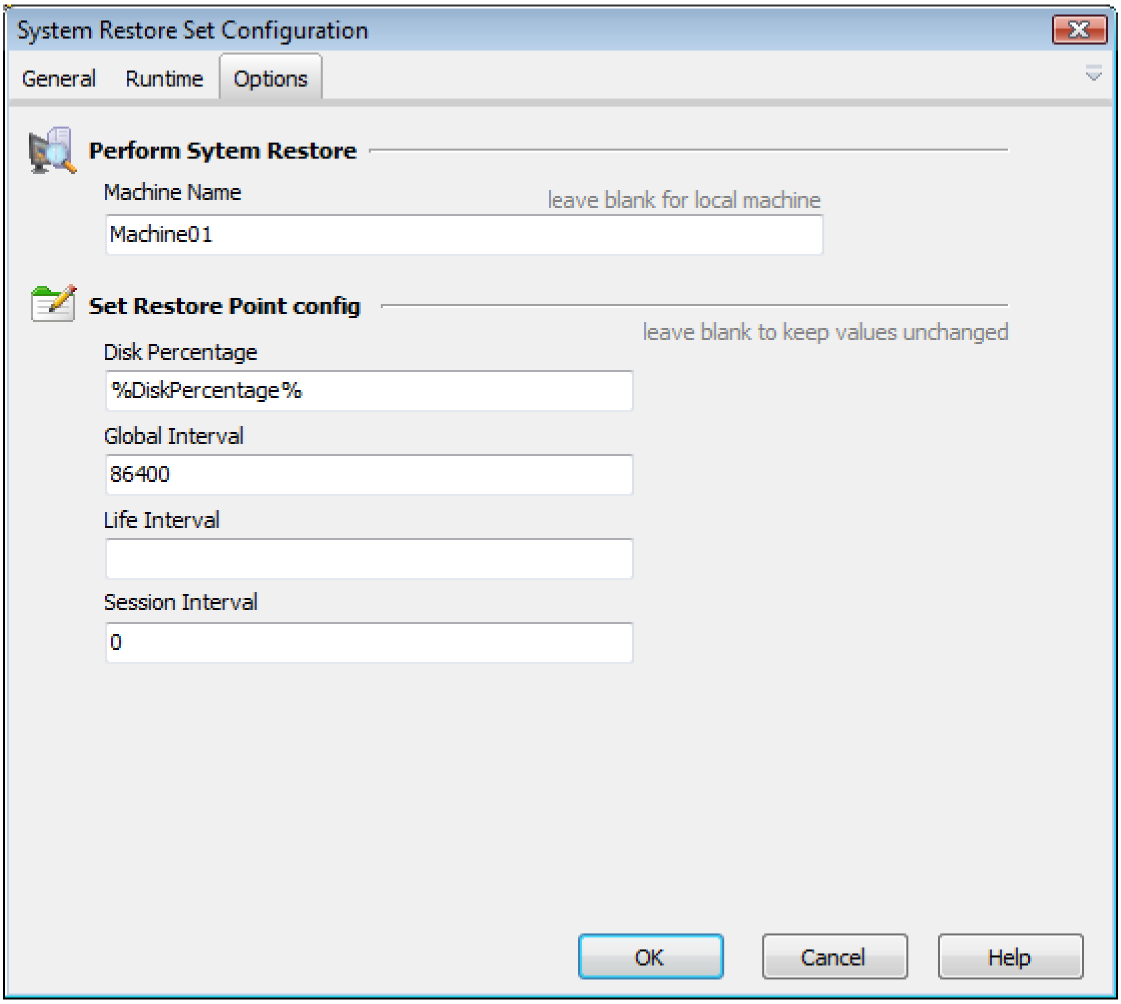
Machine Name - enter the machine name or leave blank for the local machine

Restore Point Number - enter the Restore Point Number to restore the system to. You can get the available restore points by using the System Restore List Restore Points action.

NOTES:
A reboot must be performed, as the actual restoration takes place during the reboot.

6.27.25.8 System Restore Set Configuration

The System Restore Set Configuration Action will set the machine System Restore configuration.



System Restore Set Configuration
General
Runtime
Options
Perform Sytem Restore
leave blank for local machine
Set Restore Point config
leave blank to keep values unchanged
OK
Cancel
Help

Machine Name: Machine01

Disk Percentage: %DiskPercentage%

Global Interval: 86400

Life Interval:

Session Interval: 0

Machine Name - specify the machine name or leave blank for local

Disk Percentage - enter the new disk percentage, or blank to leave value unchanged

Global Interval - enter the new global interval, or blank to leave value unchanged

Life Interval - enter the new life interval, or blank to leave value unchanged

Session Interval - enter the new session interval, or blank to leave value unchanged

NOTES:

Disk Percent (0 - 100)
The maximum amount of disk space on each drive that can be used by System Restore. This value is specified as a percentage of the total drive space. The default value is 12 percent.

Global Interval (Unsigned 32bit integer)
The absolute time interval at which scheduled system checkpoints are created, in

seconds. The default value is 86,400 (24 hours).

Life Interval (Unsigned 32bit integer)
The time interval for which restore points are preserved, in seconds. When a restore point becomes older than this specified interval, it is deleted. The default age limit is 90 days.

Session Interval (Unsigned 32bit integer)
The time interval at which scheduled system checkpoints are created during the session, in seconds. The default value is zero, indicating that the feature is turned off.

6.27.26 WBAdmin

The WBAdmin actions allow you to back and restore your operating system, volumes, drives, folders and applications.

These actions only apply to Windows Vista/Windows Server 2008 onwards. To perform a backup operation on an earlier operating system please use the Backup action.

For more information see WBAdmin at Microsoft Technet.

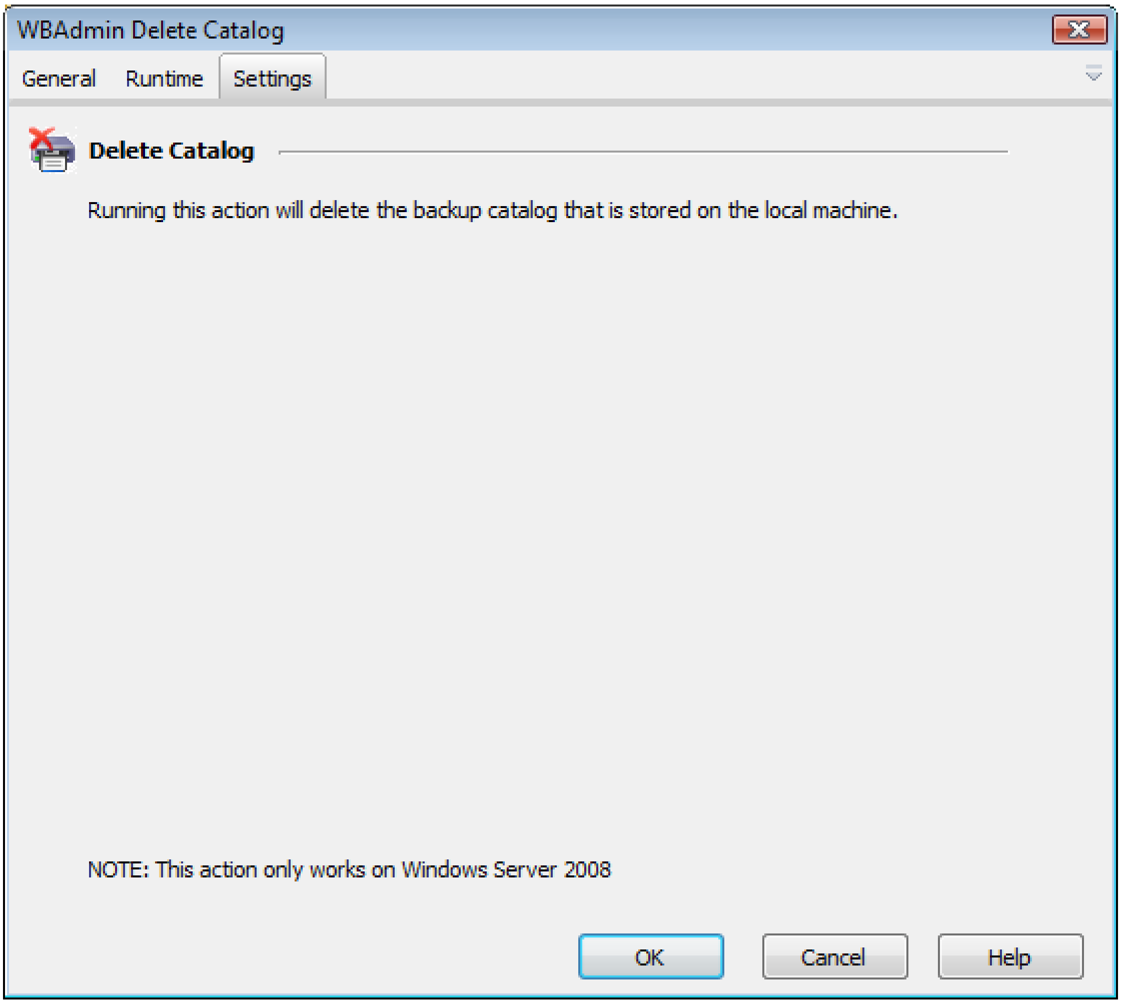
6.27.26.1 WBAdmin Delete Catalog

WBAdmin Delete Catalog

Deletes the backup catalog stored on the local machine.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Running this action will delete the backup catalog that is stored on the local machine.

NOTE: This action only works on Windows Server 2008

For more information see WBAdmin Delete Catalog

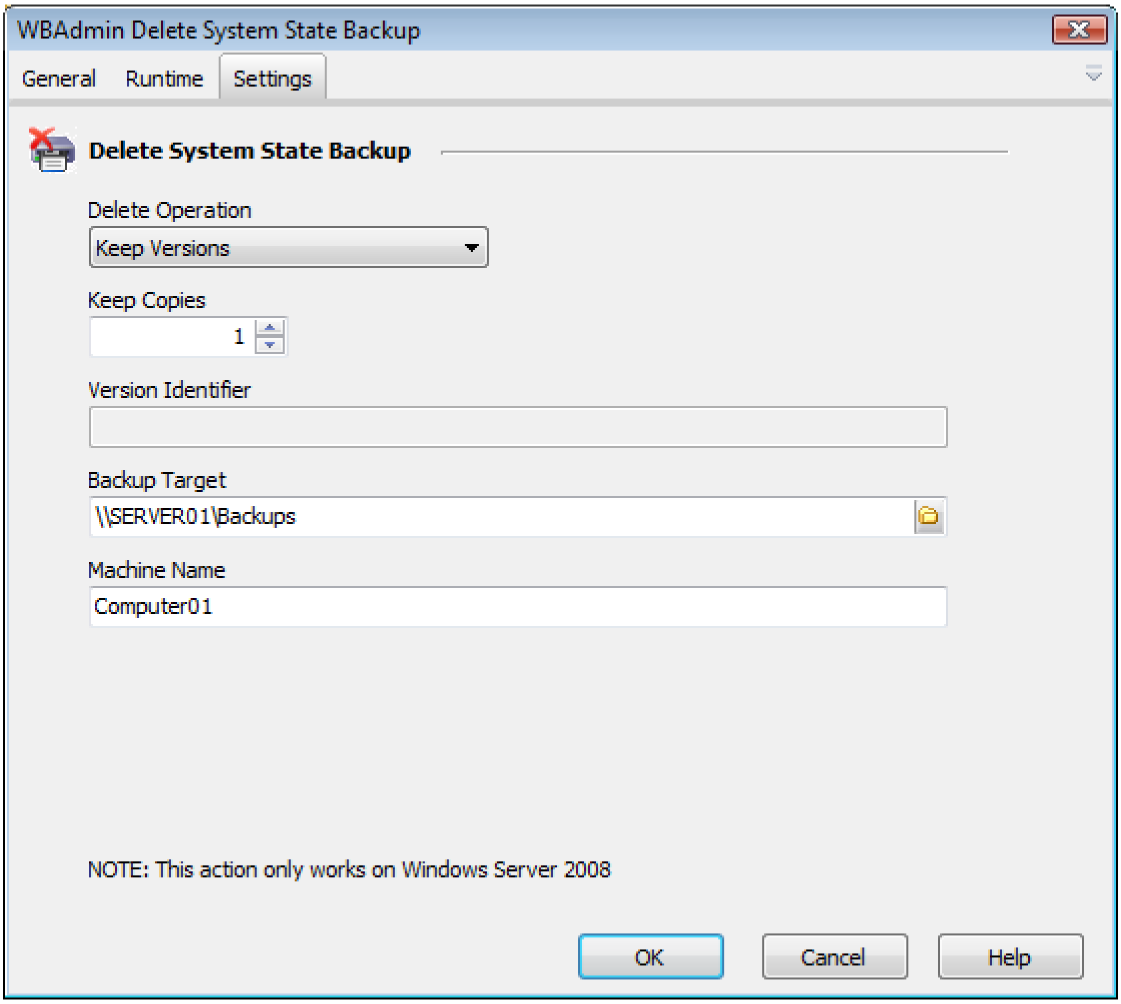
6.27.26.2 WBAdmin Delete System State Backup

WBAdmin Delete System State Backup

Deletes the System State Backups that you specify.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Delete System State Backup

Delete Operation: Keep Versions

Keep Copies: 1

Version Identifier:

Backup Target: \\SERVER01\Backups

Machine Name: Computer01

NOTE: This action only works on Windows Server 2008

Delete Operation - This option allows you to specify how to specify which System State Backups will be deleted.

Keep Versions - When using this option, provide the number of most recent backups to retain, all the others will be deleted.

Version - Specify a particular backup to be deleted.

Delete Oldest - Enabling this option will automatically choose the oldest System State Backup and delete it.

Keep Copies - When using the *Keep Versions* delete operation, this value is the number of latest backups to retain.

Version Identifier - When using the *Version* delete operation, this is the backup identifier of the backup to be deleted. If you do not know the version identifier of the backup, you can use the WBAdmin Get Versions action to get a list of the backups available. You can also use the WBAdmin Get Items action to determine to view the details of the particular backup.

Backup Target - Provide the location where the backups are located. This only required if the backups are not stored on the local machine.

Machine Name - Provide the name of the machine where the backups are stored. Only required when using the Backup Target option.

For more information please see WBAdmin Delete System State Backup

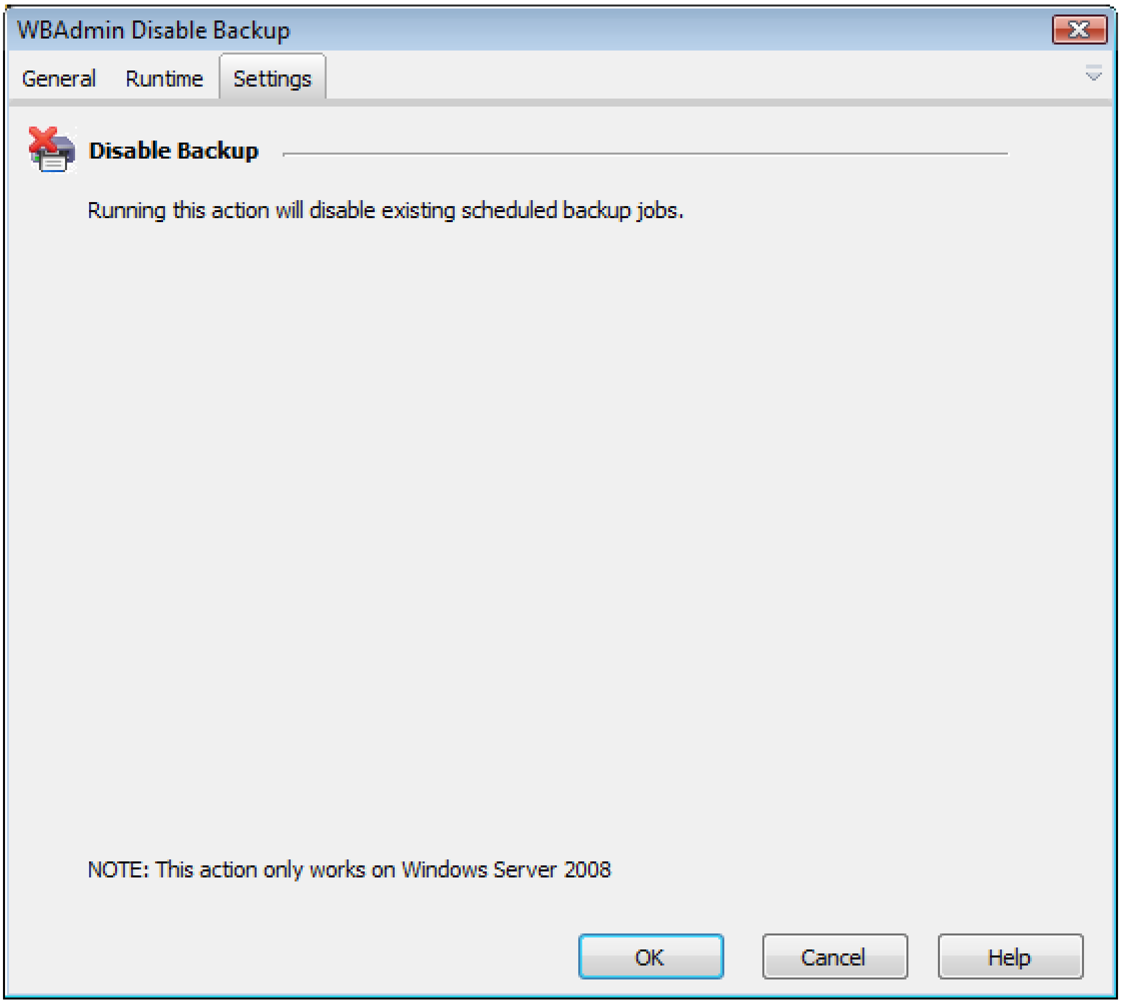
6.27.26.3 WBAdmin Disable Backup

WBAdmin Disable Backup

Disables the current scheduled daily backups from running.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Running this action will disable existing scheduled backup jobs.

NOTE: This action only works on Windows Server 2008

For more information please see WBAdmin Disable Backup

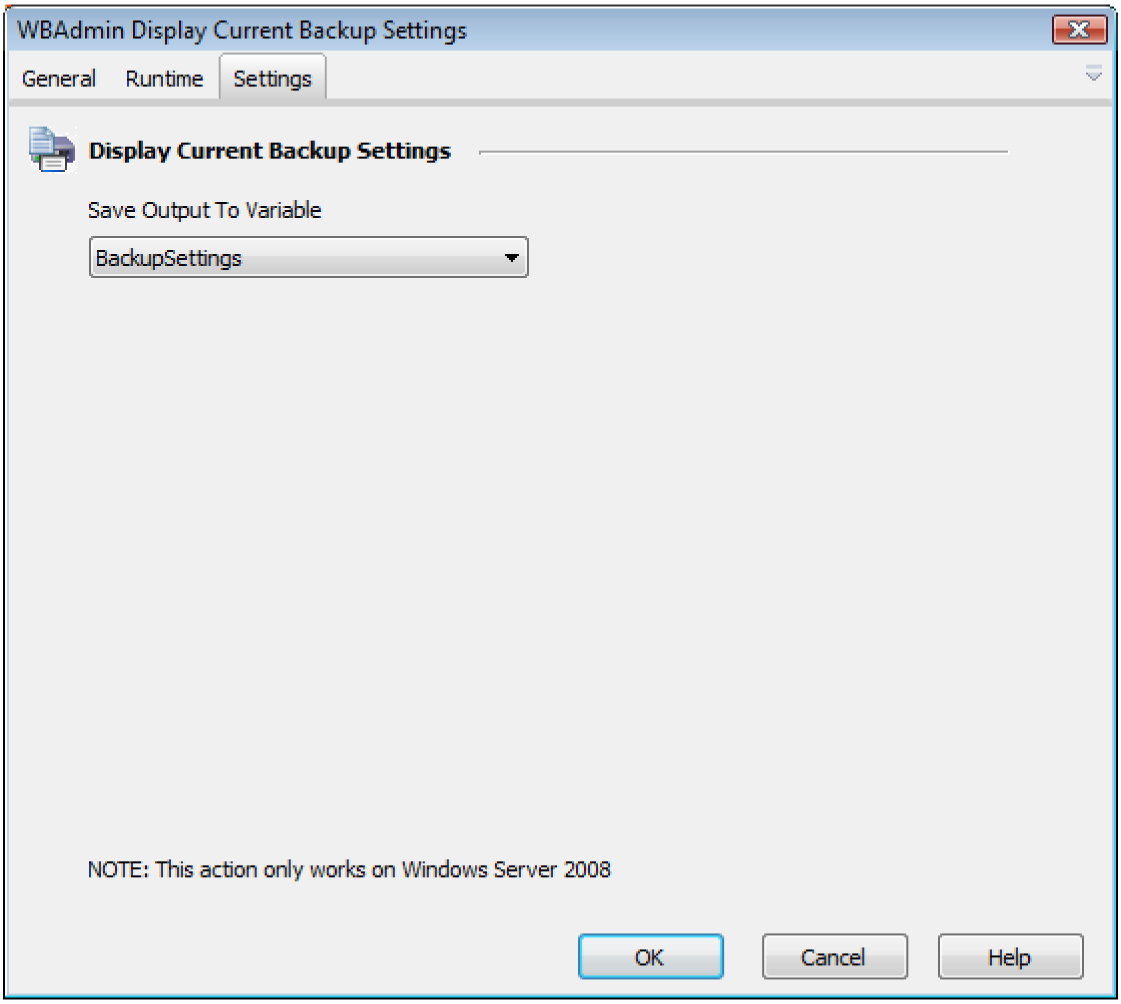
6.27.26.4 WBAdmin Display Current Backup Settings

WBAdmin Display Current Backup Settings

Displays the settings for the scheduled backup.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings


WBAdmin Display Current Backup Settings
General
Runtime
Settings
Display Current Backup Settings

Save Output To Variable: BackupSettings

NOTE: This action only works on Windows Server 2008

For more information see WBAdmin Enable Backup

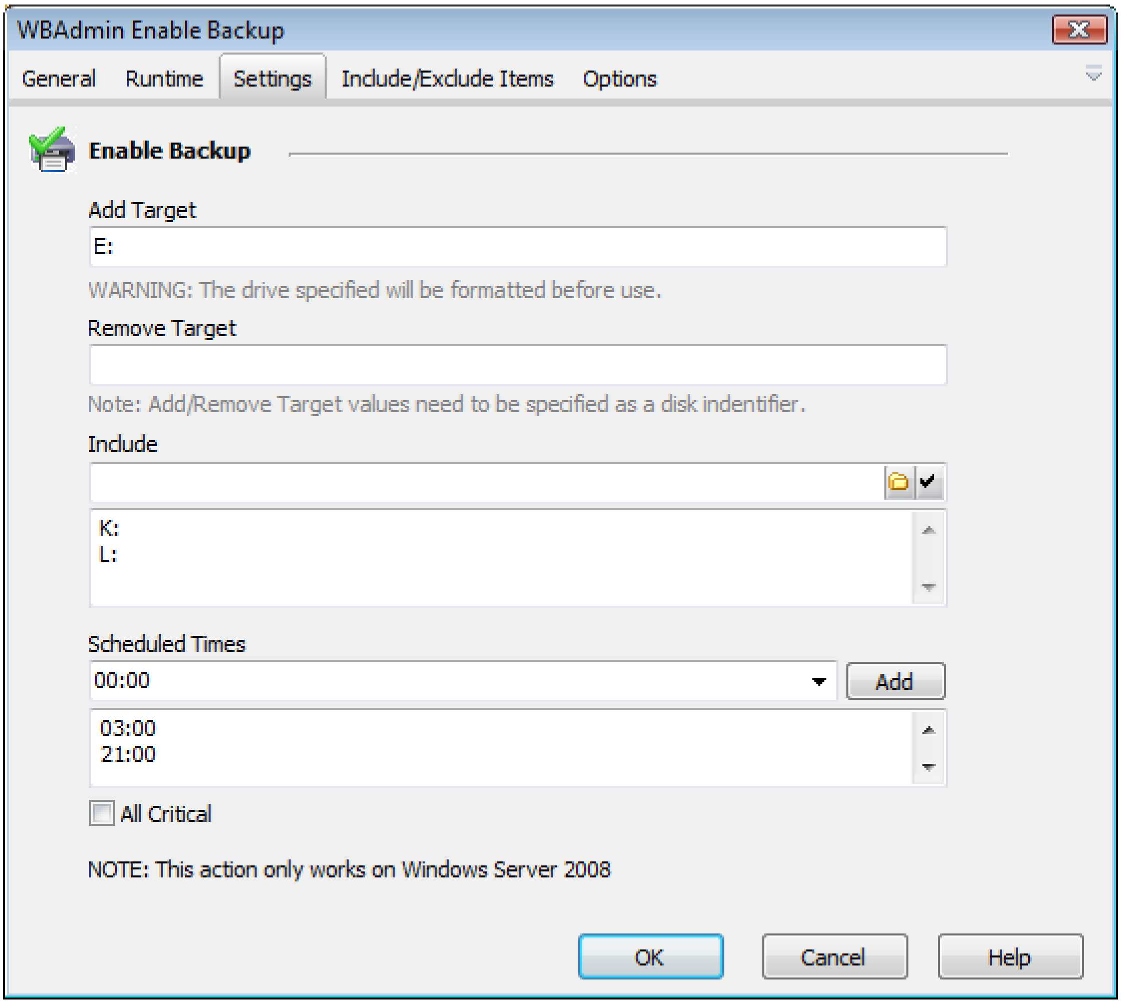
6.27.26.5 WBAdmin Enable Backup

WBAdmin Enable Backup

Enables a new scheduled daily backup operation or modify the existing daily backup schedule.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Enable Backup

Add Target: E:

WARNING: The drive specified will be formatted before use.

Remove Target:

Note: Add/Remove Target values need to be specified as a disk indentifier.

Include:

K:
L:

Scheduled Times: 00:00

03:00
21:00

☐ All Critical

NOTE: This action only works on Windows Server 2008

Add Target - Specify a location for the backup to be stored.

Windows Server 2008 - Target must be specified as a disk identifier.

Windows Server 2008 R2 - Target must be specified disk identifier, volume or UNC share path.

WARNING: The drive specified will be formatted before use, this will result in all data being erased.

Remove Target - Specify a storage location to be removed from existing backup schedule.

Include - Specifies a list of items to be included in backup operation.

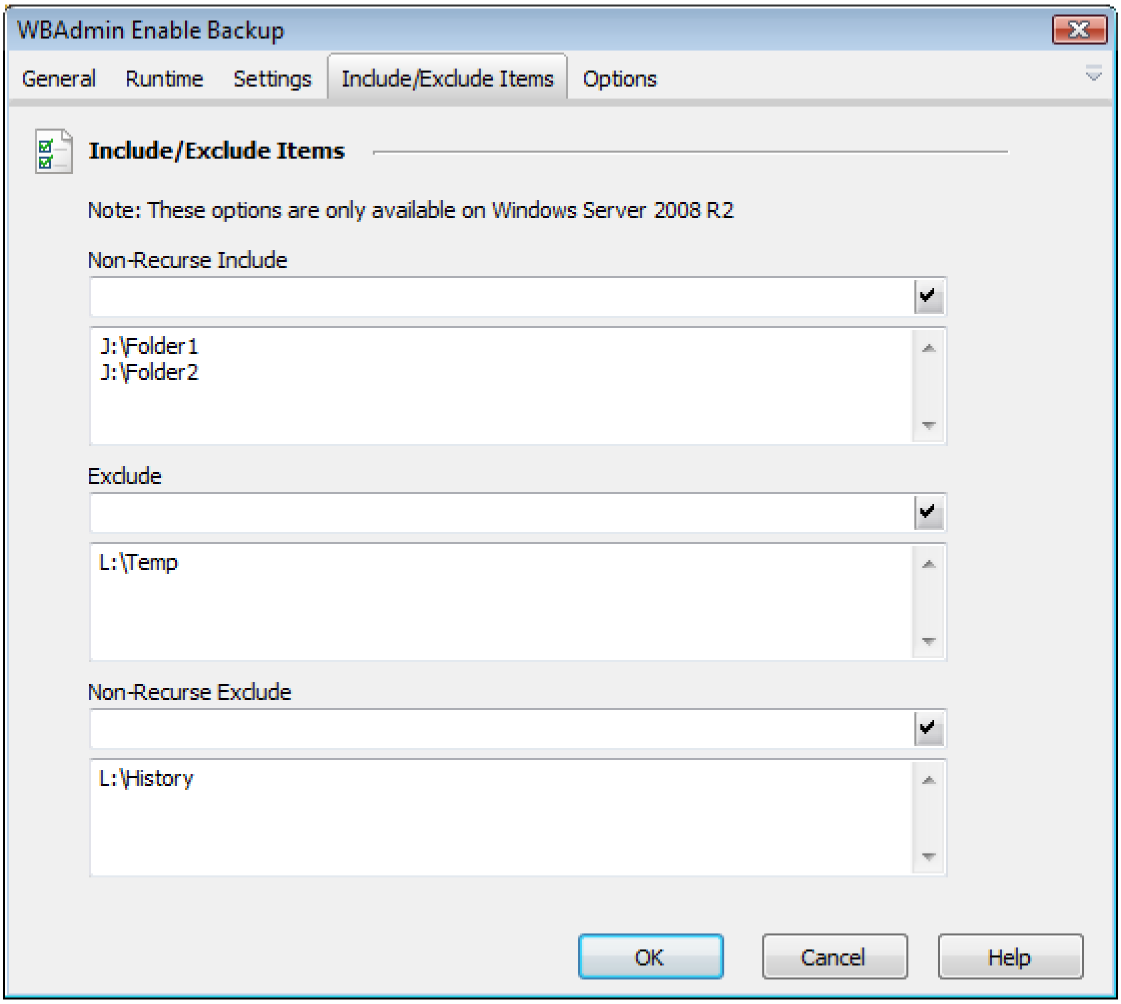
Windows Vista/Server 2008 - List can contain volume drive letters, volume mount points or GUID based volume names.

Windows 7/Server 2008 R2 - List can contain files, folders or volumes. Volumes can be specified via volume drive letters, volume mount points, GUID based volume names.

Scheduled Times - Specify times that backup operation should run.

All Critical - Specifies that all critical volumes are included in the backups.

Include/Exclude Items



Include/Exclude Items

Note: These options are only available on Windows Server 2008 R2

Non-Recurse Include

J:\Folder1
J:\Folder2

Exclude

L:\Temp

Non-Recurse Exclude

L:\History

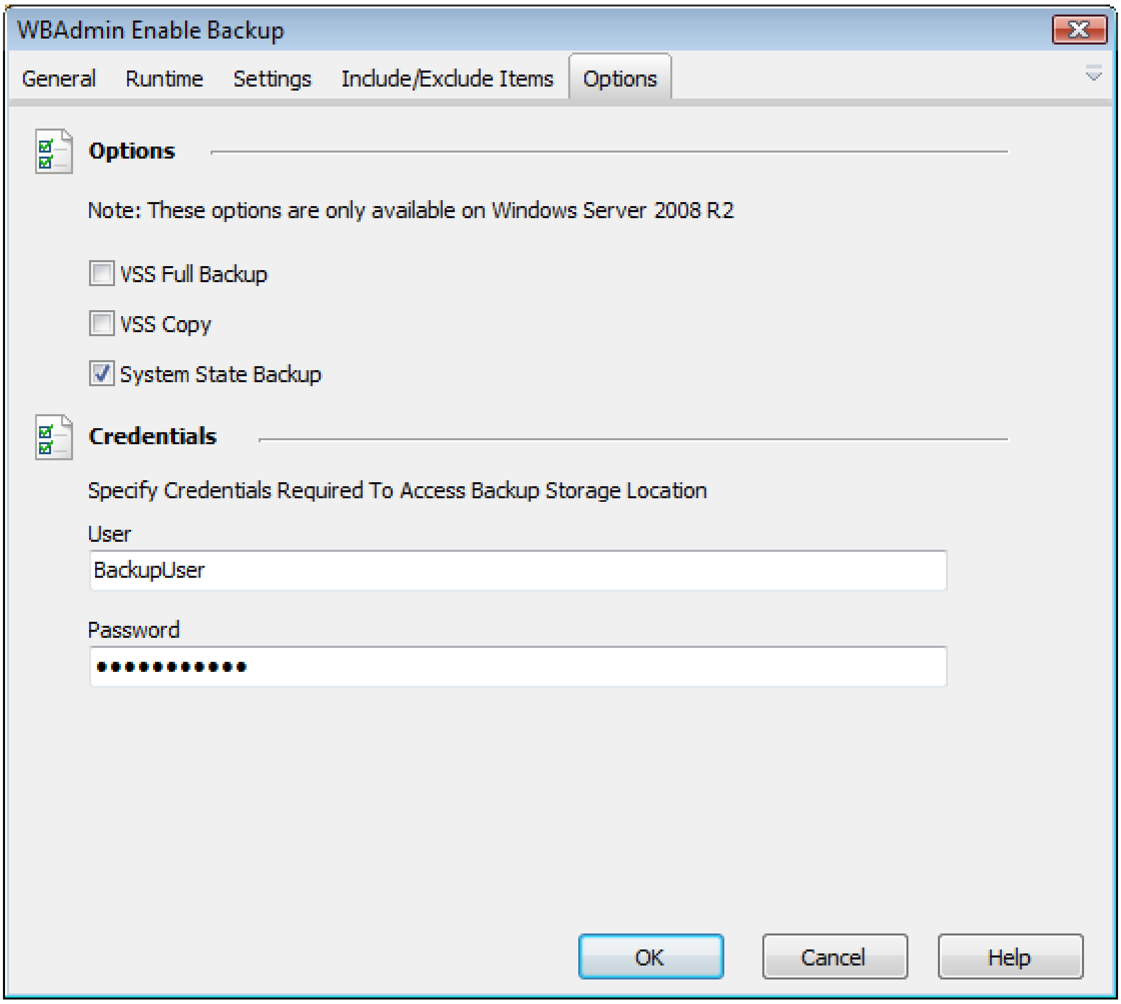
Note: *The options on this page only apply to Windows Server 2008 R2*

Non-Recurse Include - Specify a list of non-recursive items to be included in the backup.

Exclude - Specify a list of items to exclude from the backup.

Non-Recurse Exclude - Specify a list of non-recursive items to be excluded from the backup.

Options



WBAdmin Enable Backup

Options

Note: These options are only available on Windows Server 2008 R2

□ VSS Full Backup

□ VSS Copy

☑ System State Backup

Credentials

Specify Credentials Required To Access Backup Storage Location

User

BackupUser

Password

●●●●●●●●●●●

Note: *The options on this page only apply to Windows Server 2008 R2*

VSS Full Backup - Performs a full backup using the Volume Shadow Copy Service. History of each file is updated to indicate that it was backed up.

VSS Copy - Performs a copy backup using the Volume Shadow Copy Service. History of each file is not updated.

System State Backup - Includes System State in the backup operation.

Credentials - Only required when using a UNC path as the backup target. Specifies the credentials of a user who has write access to the share location.

For more information please see WBAdmin Enable Backup

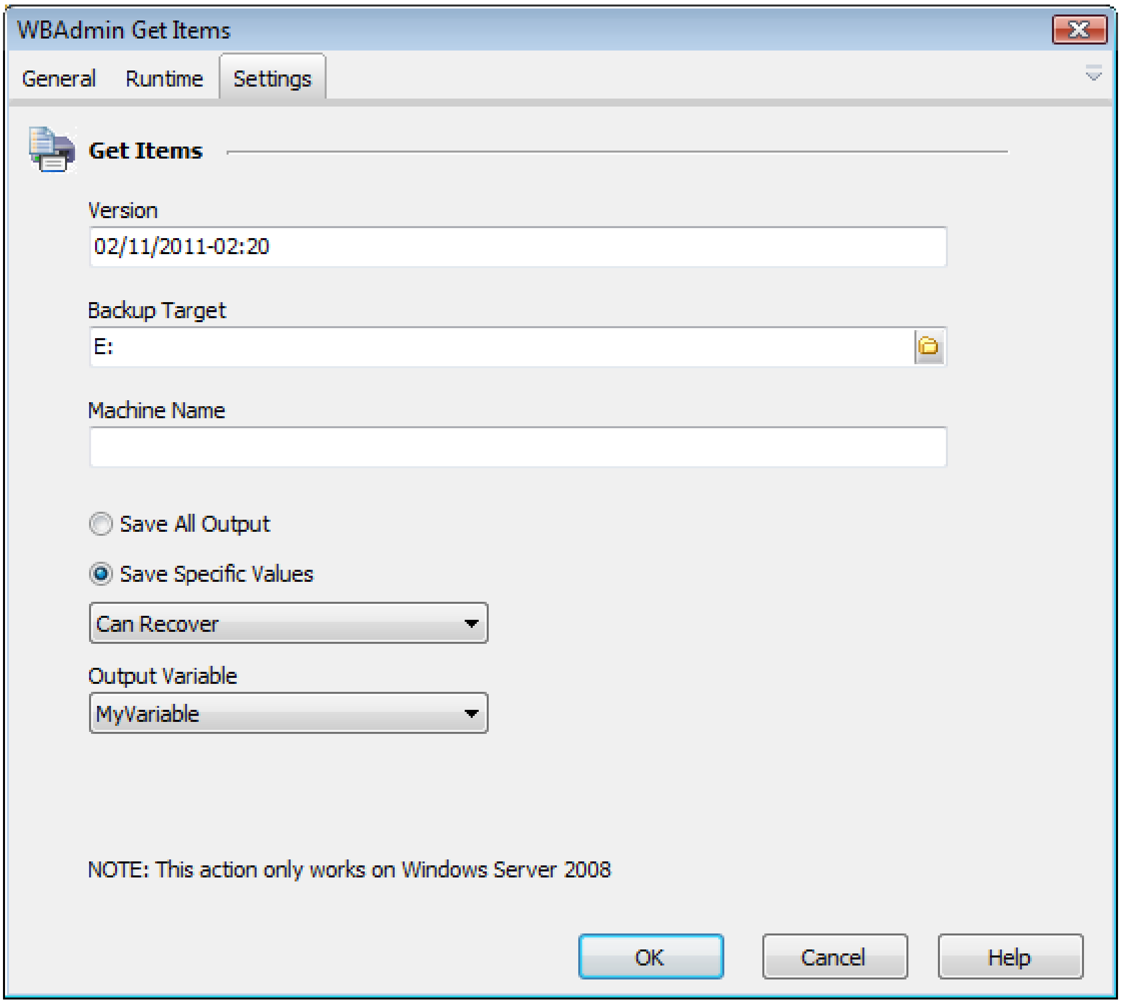
6.27.26.6 WBAdmin Get Disks

WBAdmin Get Disks

Allows you retrieve a list of disks that are currently online for the local machine.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



WBAdmin Get Items

Get Items

Version
02/11/2011-02:20

Backup Target
E:

Machine Name

○ Save All Output

● Save Specific Values

Can Recover

Output Variable
MyVariable

NOTE: This action only works on Windows Server 2008

Save To Output Variable - Specify the variable to save the output to.

Save All Output - Saves entire output of command to selected variable.

Save Specific Values - This option allows you to specify the detail that you want to capture:

Disk Name - The name of the current disk

Disk Number - The index of the current disk

Disk Identifier - The disk identifier (GUID) for the current disk

Total Space - Total disk size for the current disk

Used Space - The amount of space in use for the current disk

Volumes - Volume letter(s) and volume name(s) for the current disk

For more information please see WBAdmin Get Disks

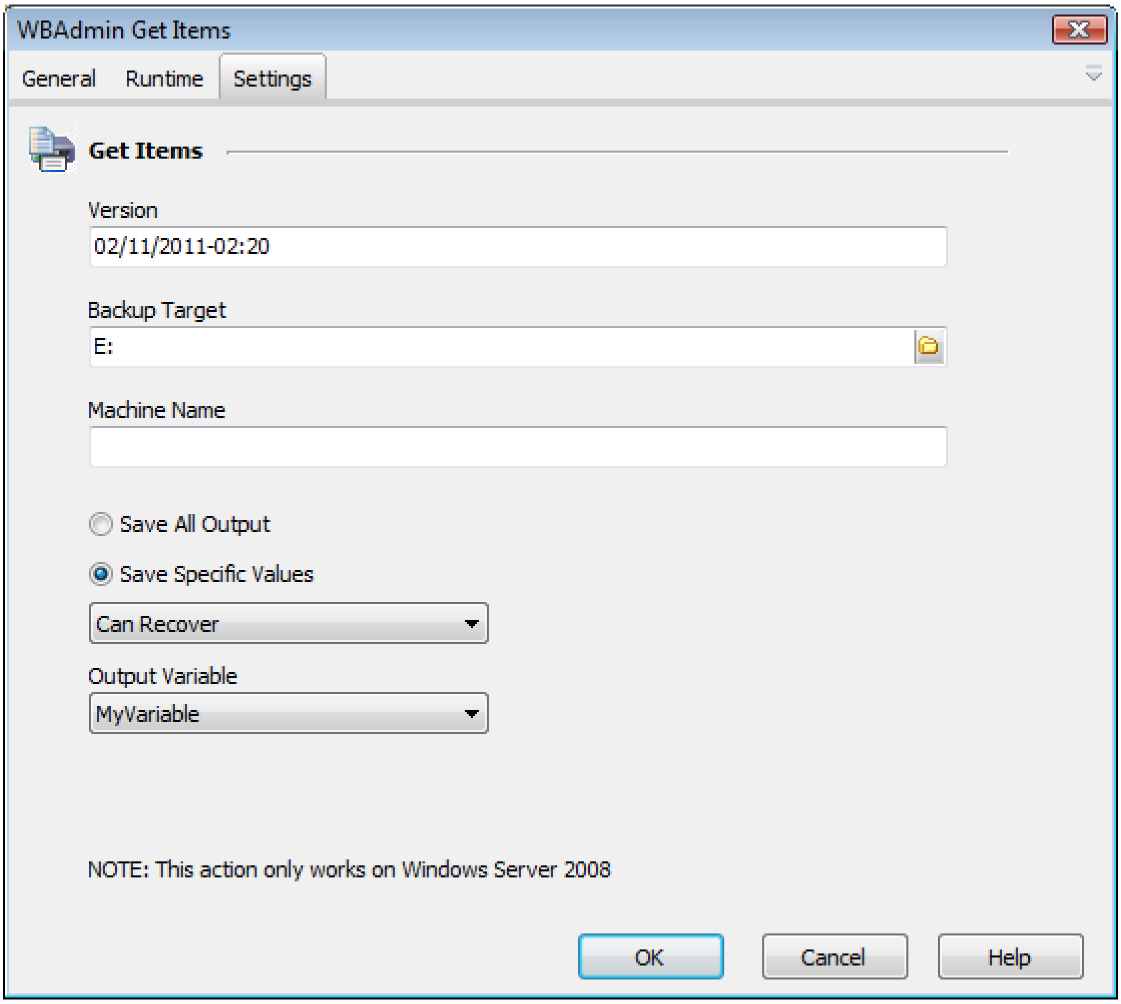
6.27.26.7 WBAdmin Get Items

WBAdmin Get Items

Allows you to list the items available for recovery from a backup operation.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Version
02/11/2011-02:20

Backup Target
E:

Machine Name

○ Save All Output

● Save Specific Values

Can Recover

Output Variable
MyVariable

NOTE: This action only works on Windows Server 2008

Version - Specifies the version of the backup to query in the format MM/DD/YYYY-HH:MM. To query available backups

Backup Target - The storage location that contains the backups which you want to query.

Machine Name - When the Backup Target is remote share that contains backups for multiple machines allows you to specify the name of the machine you want to query.

Save All Output - This option saves the entire output from the command.

Save Specific Values - This option allows you to specify the detail that you want to capture:

Volume ID - The Volume Identifier of the volume that has been backed up

Volume Name - The name of the volume that has been backed up

Volume Size - The size of the volume that was backed up

Can Recover - Items available for recovery

Output Variable - The variable that will save the output from this command.

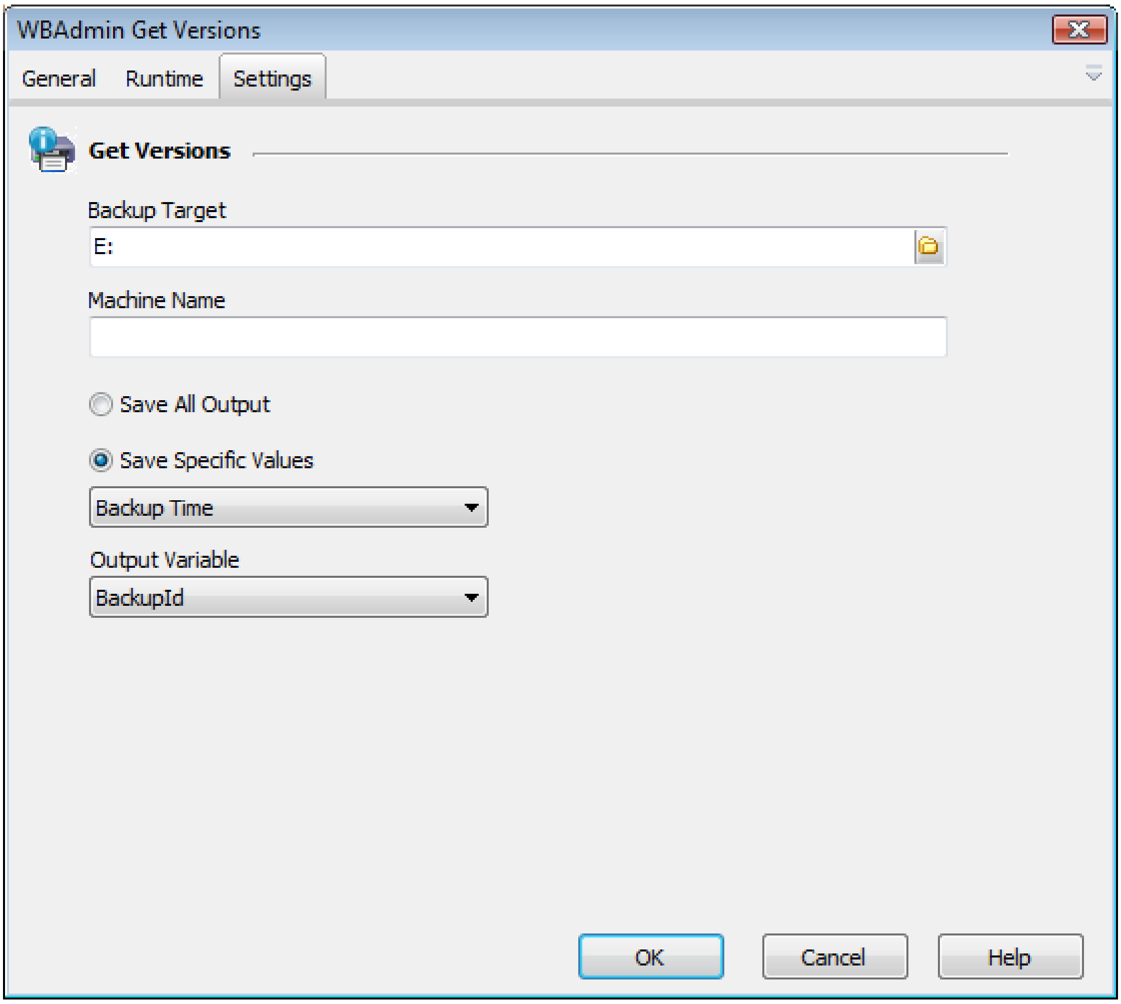
For more information please see WBAdmin Get Items

6.27.26.8 WBAdmin Get Versions

WBAdmin Get Versions

The WBAdmin Get Versions action provides retrieves details about previous backups from the local machine.

Settings



Backup Target: E:

Machine Name:

○ Save All Output

● Save Specific Values

Backup Time

Output Variable: BackupId

Backup Target - The storage location that contains backups that you want to retrieve details about.

Machine Name - When the Backup Target is remote share that contains backups for multiple machines allows you to specify the name of the machine you want to query.

Save All Output - Save the entire output of the command to the selected variable.

Save Specific Values - This option allows you to specify the detail that you want to capture:

Backup Time: The time that the backup operation completed

Backup Target: The target of the backup operation

Version Identifier: The version identifier for the current backup

Can Recover: Specifies what items can be recovered from this backup (Volumes, Files, Applications or System State)

Snapshot ID: The identifier for the snapshot

Output Variable -The variable to store the output of the command.

For more information see WBAdmin Get Versions

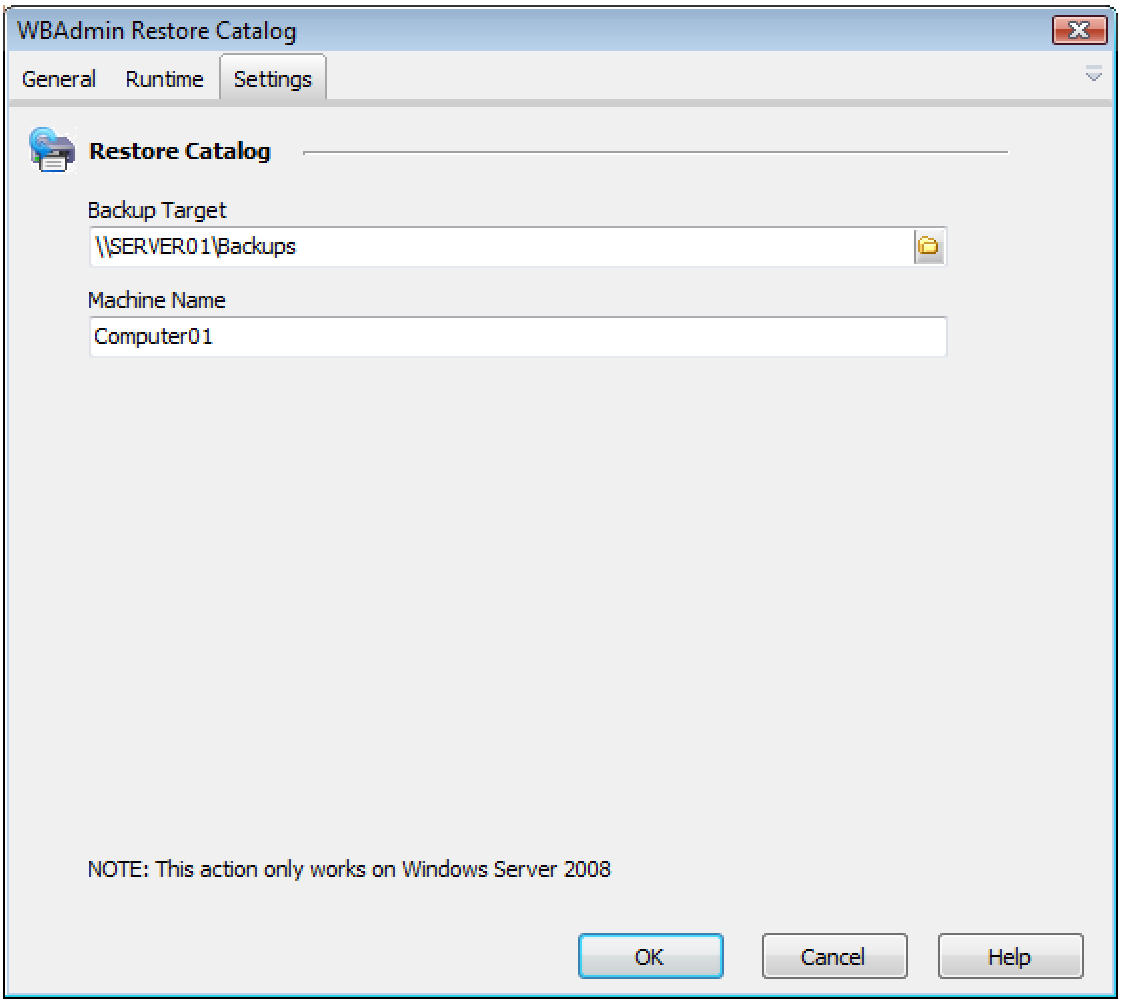
6.27.26.9 WBAdmin Restore Catalog

WBAdmin Restore Catalog

Performs a system recovery from the backup that you specify.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Backup Target
\\SERVER01\Backups

Machine Name
Computer01

NOTE: This action only works on Windows Server 2008

Backup Target - The location of the backup catalog.

Machine Name - The name of the machine that you want restore the catalog for.

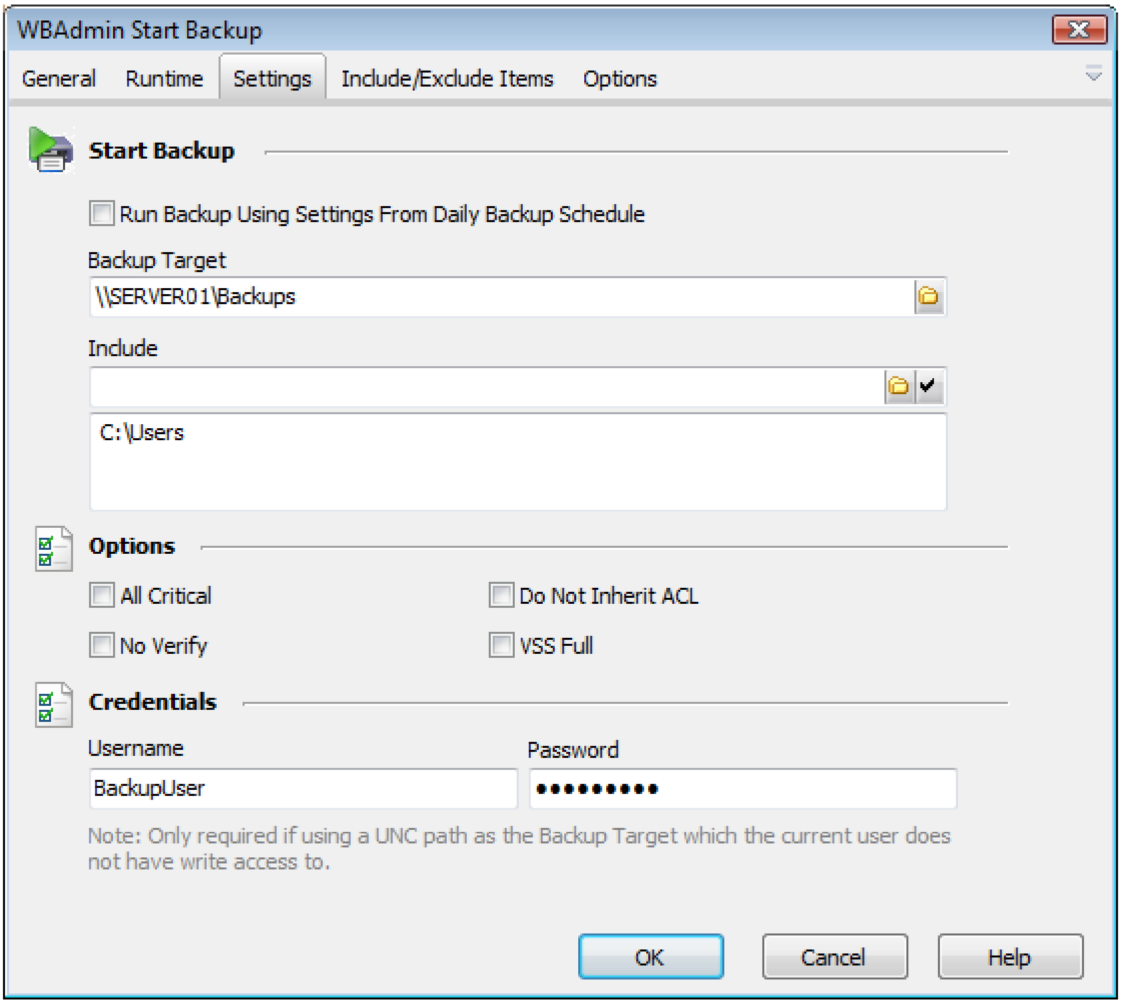
For more information see WBAdmin Restore Catalog

6.27.26.10WBAdmin Start Backup

WBAdmin Start Backup

Allows you to start a backup using the scheduled daily backup settings or using the settings that you provide.

Settings



Start Backup

☐ Run Backup Using Settings From Daily Backup Schedule

Backup Target: \\SERVER01\Backups

Include: C:\Users

Options

☐ All Critical ☐ Do Not Inherit ACL

☐ No Verify ☐ VSS Full

Credentials

Username: BackupUser

Password: •••••••••

Note: Only required if using a UNC path as the Backup Target which the current user does not have write access to.

Run Backup Using Settings From Daily Backup Schedule - Selecting this option disables all other options and runs a backup job using the current daily backup settings. If no daily backup exists the action will fail.

Backup Target - Allows you to specify the storage location for the backup. This value can either be a drive letter, a volume GUID based path or a UNC path.

Include - Specifies a list of items to be included in backup operation.

Windows Vista/Server 2008 - List can contain volume drive letters, volume mount points or GUID based volume names.

Windows 7/Server 2008 R2 - List can contain files, folders or volumes. Volumes can be specified via volume drive letters, volume mount points, GUID based volume names.

All Critical - Specifies that all critical volumes are included in the backups.

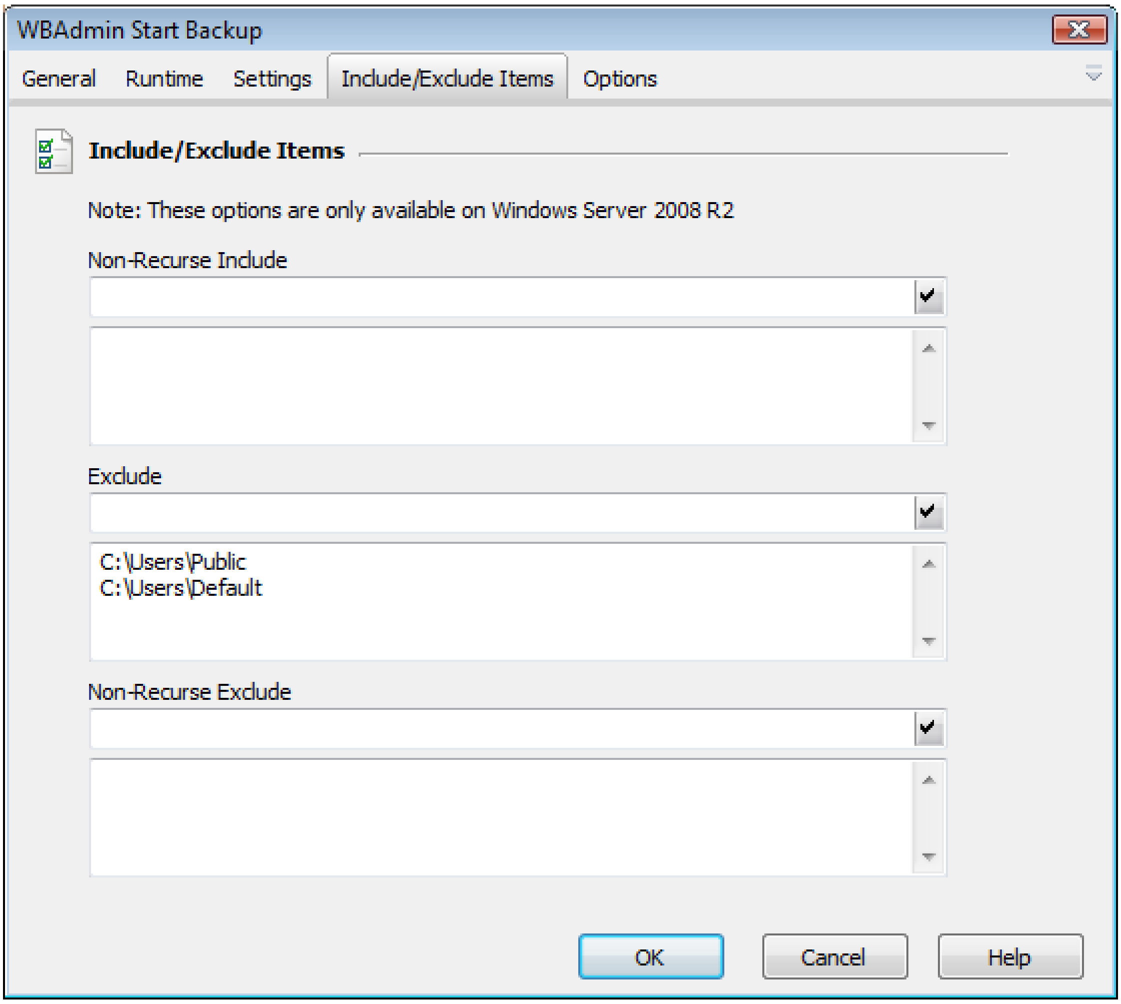
Do Not Inherit ACL - Applies the access control list (ACL) permissions that correspond to the credentials provided by the user credentials (username and password) to the UNC store location. This means that you need to provide these credentials or be a member of the administrators group to access this backup.

No Verify - Backups saved to removable media are not verified for errors.

VSS Full - Performs a full backup using the Volume Shadow Copy Service. History of each file is updated to indicate that it was backed up.

Credentials - Only required when using a UNC path as the backup target. Specifies the credentials of a user who has write access to the share location.

Include/Exclude Items



Note: These options are only available on Windows Server 2008 R2

Non-Recurse Include:

Exclude:
C:\Users\Public
C:\Users\Default

Non-Recurse Exclude:

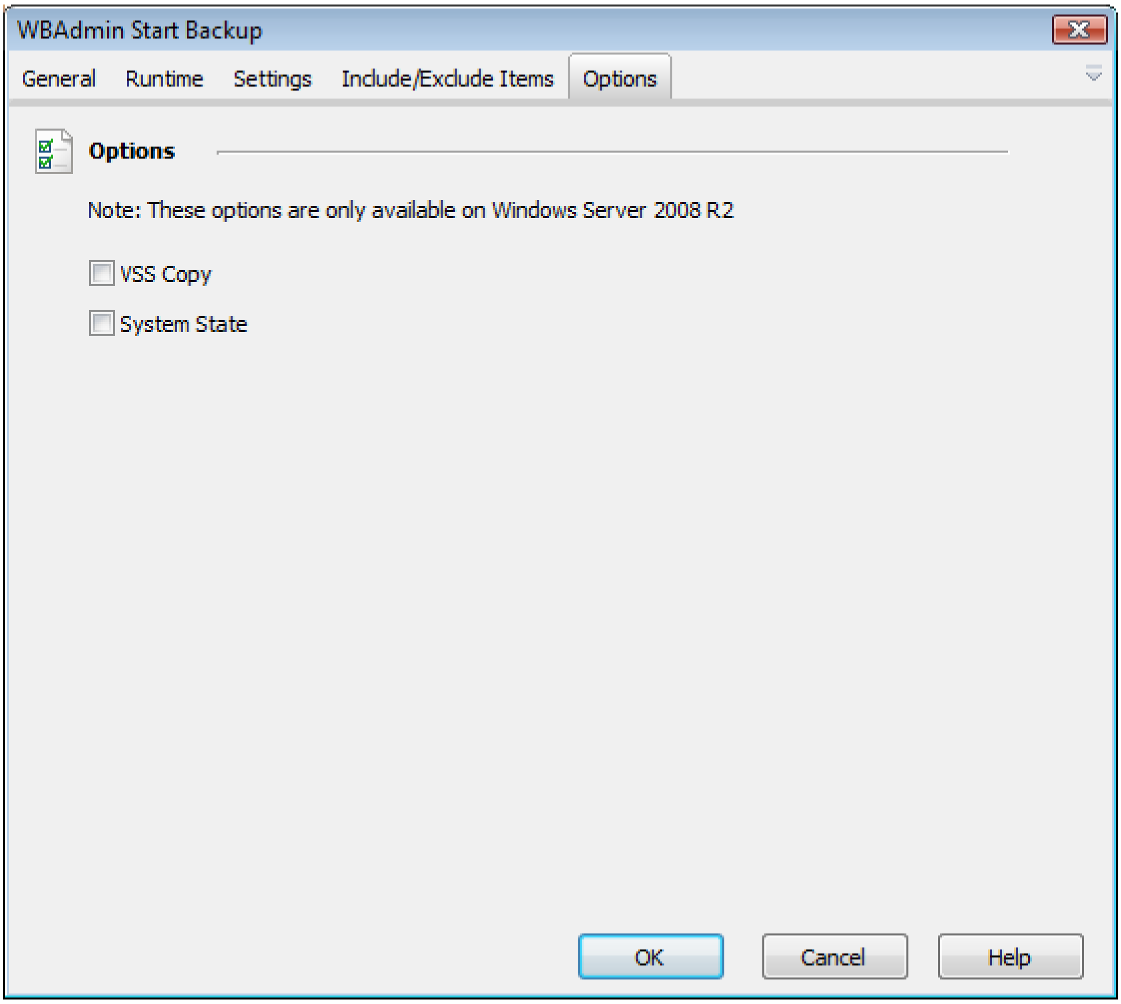
Note: *The options on this page only apply to Windows Server 2008 R2*

Non-Recurse Include - Specify a list of non-recursive items to be included in the backup.

Exclude - Specify a list of items to exclude from the backup.

Non-Recurse Exclude - Specify a list of non-recursive items to be excluded from the backup.

Options



WBAdmin Start Backup

Options

Note: These options are only available on Windows Server 2008 R2

□ VSS Copy

□ System State

Note: *The options on this page only apply to Windows Server 2008 R2*

VSS Copy - Performs a copy backup using the Volume Shadow Copy Service. History of each file is not updated.

System State - Includes System State in the backup operation.

For more information please see WBAdmin Start Backup

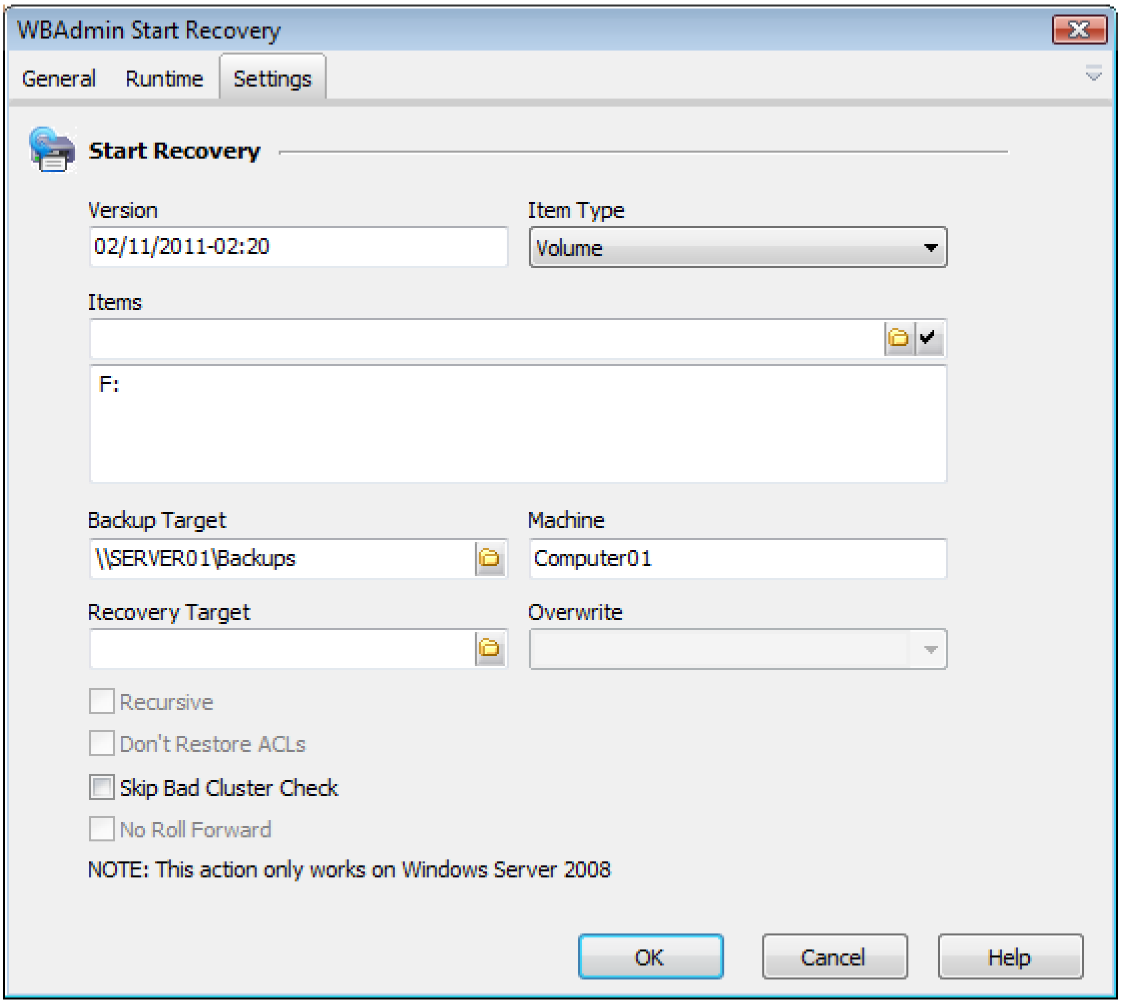
6.27.26.11WBAdmin Start Recovery

WBAdmin Start Recovery

Runs a recovery operation to restore items from a earlier backup operation.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



WBAdmin Start Recovery

Start Recovery

Version: 02/11/2011-02:20

Item Type: Volume

Items: F:

Backup Target: \\SERVER01\Backups

Machine: Computer01

Recovery Target:

Overwrite:

□ Recursive

□ Don't Restore ACLs

□ Skip Bad Cluster Check

□ No Roll Forward

NOTE: This action only works on Windows Server 2008

Version - The version identifier of the backup to restore. To get the correct version identifier you can use the WBAdmin Get Versions action.

Item Type - Select the type of items to recover. Options are Volume, App or File.

Items - Specify the items to restore. If item type is Volume or App then only a single item may be specified. Volumes can be specified via a drive letter, volume mount point or GUID based volume name.

Backup Target - The storage location that contains the backup you want to restore.

Machine - When the Backup Target is remote share that contains backups for multiple machines allows you to specify the name of the machine you want to restore.

Recovery Target - Specify the location where you want to restore items to. Only use this if you want to restore to a different location to the location that was backed up.

Overwrite - This option is only available when recovering files. Allows you to set the behaviour when a file to be recovered already exists in the destination location. There are three options available:

Skip - Don't restore the file and continue restoration with next file.

Create Copy - Creates a copy of the existing file so that the existing file is not overwritten.

Overwrite - The existing file is overwritten with the file from the backup.

Recursive - This option is only available when recovering files. Performs a recursive restore of files and directories.

Don't Restore ACLs - This option is only available when recovering files. Prevents security access control lists from being restored.

Skip Bad Cluster Check - This option is only available when recovering volumes. Disables check of destination volume for bad clusters before performing restore.

No Roll Forward - This option is only available when recovering applications. Allows for previous point-in-time recovery of an application.

For more information see WBAdmin Start Recovery

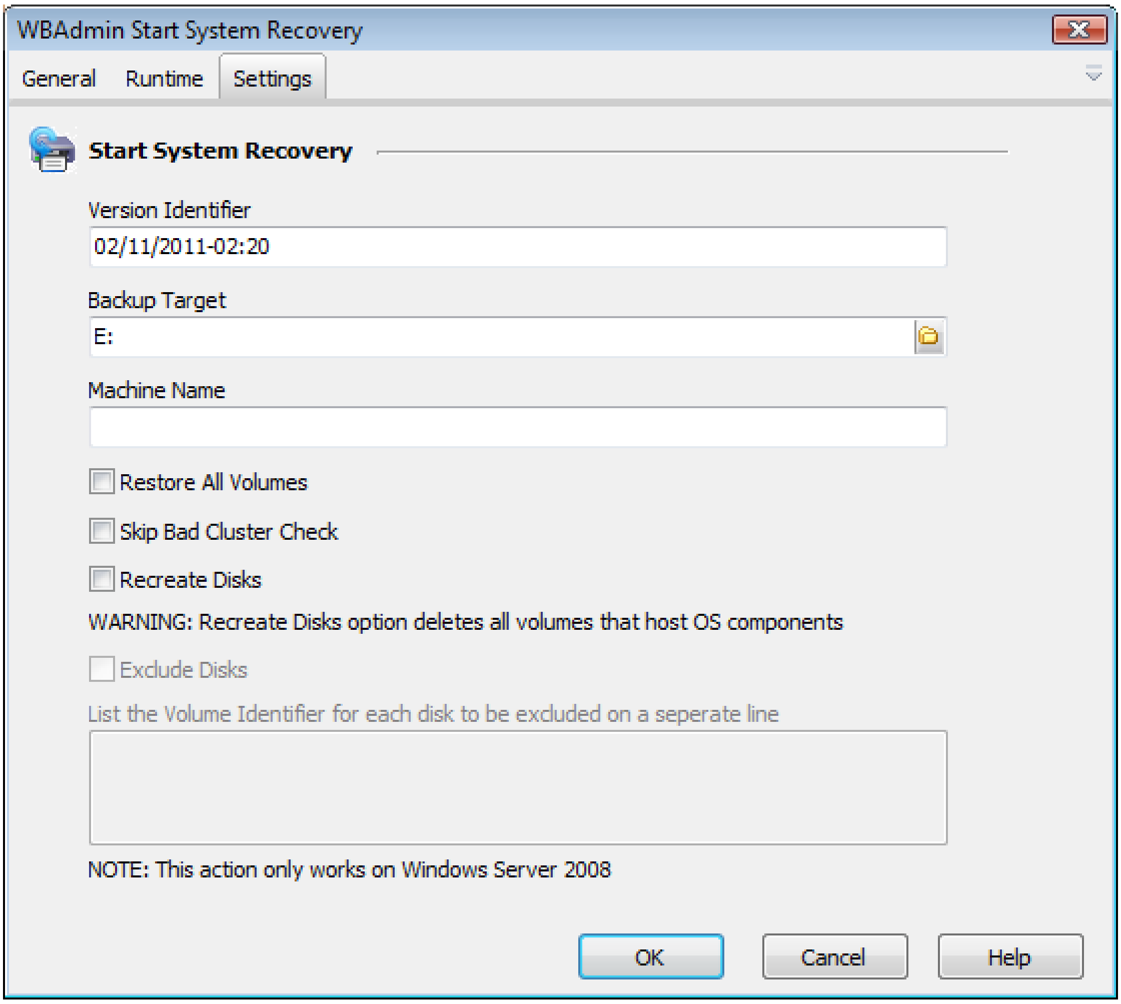
6.27.26.11 WBAdmin Start System Recovery

WBAdmin Start System Recovery

Performs a system recovery from the backup that you specify.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Version Identifier
02/11/2011-02:20

Backup Target
E:

Machine Name

□ Restore All Volumes

□ Skip Bad Cluster Check

□ Recreate Disks

WARNING: Recreate Disks option deletes all volumes that host OS components

□ Exclude Disks

List the Volume Identifier for each disk to be excluded on a seperate line

NOTE: This action only works on Windows Server 2008

Version Identifier - The version identifier of the backup to restore. To get the correct version identifier you can use the WBAdmin Get Versions action.

Backup Target - The storage location that contains the backup you want to restore.

Machine Name - When the Backup Target is remote share that contains backups for multiple machines allows you to specify the name of the machine you want to restore to.

Restore All Volumes - Restores all volumes included in the backup rather than just the critical volumes (which is the default behaviour).

Skip Bad Cluster Check - Disables check of destination disks for bad clusters before performing restore.

Recreate Disks - This option recreates the disk configuration state as it was at the time of backup.

WARNING: This option will delete data on volumes that host operating system components, it may also format data volumes.

Exclude Disks - When using the Recreate Disks option, this list of disks will be excluded from modification.

For more information see WBAdmin Start System Recovery

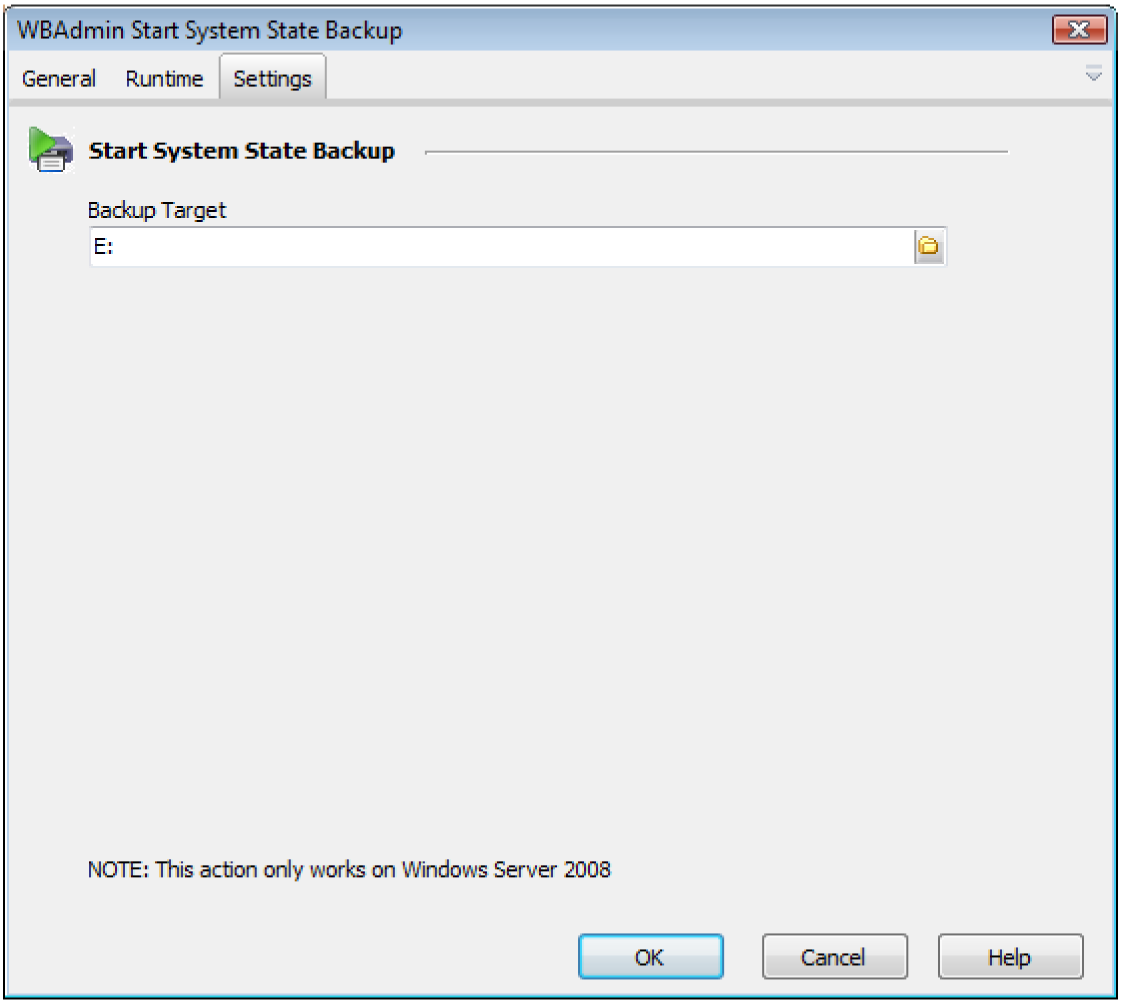
6.27.26.1:WBAdmin Start System State Backup

WBAdmin Start System State Backup

Creates a backup of the System State for the local machine.

This action only applies to Windows Server 2008/Server 2008 R2.

Settings



Start System State Backup

Backup Target

E:

NOTE: This action only works on Windows Server 2008

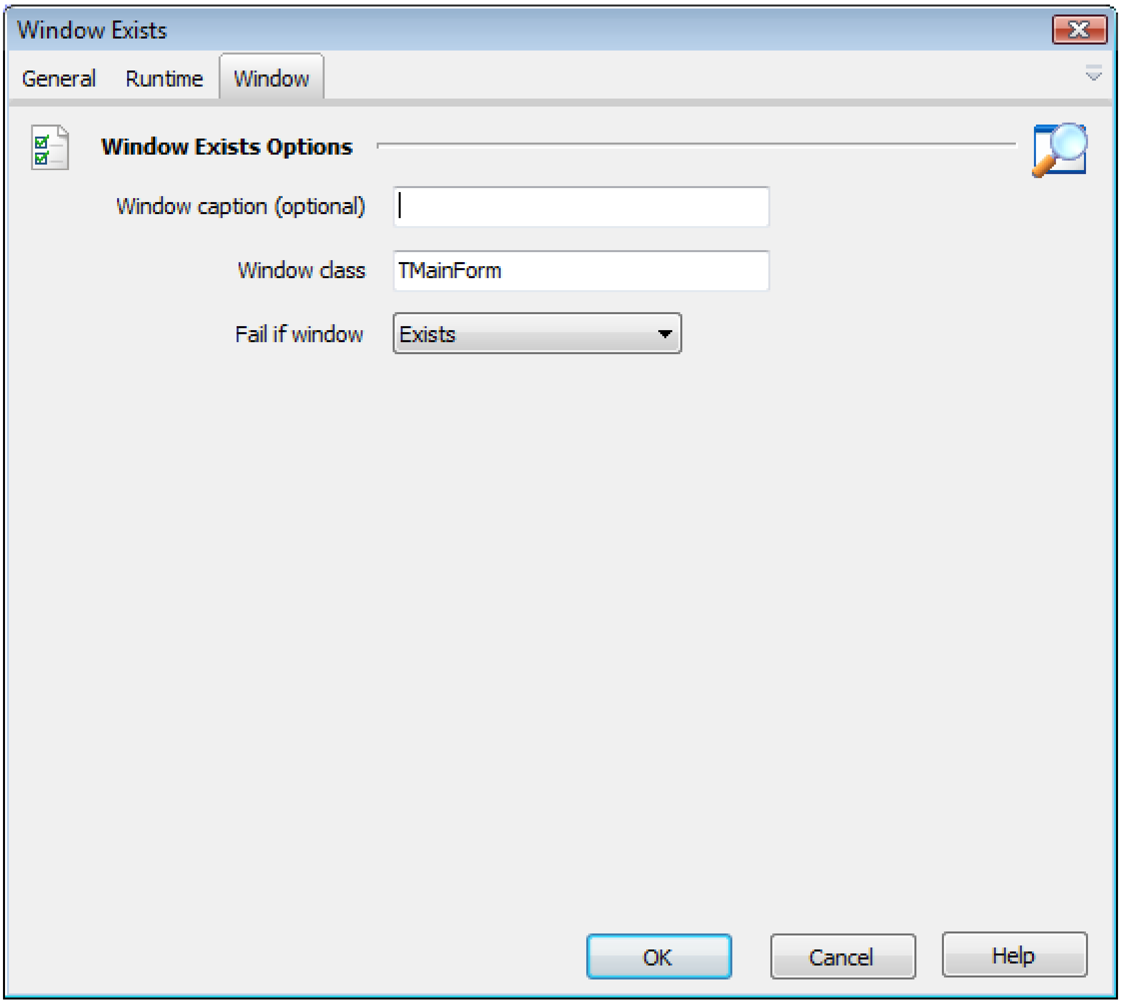
Backup Target - Specifies the location to store the System State Backup.

For more information see WBAdmin Start System State Backup

6.27.27 Window Exists Action

This action enables you to check if an application is running by checking for a Window Caption or Window Class (windows API window class name), and choose to fail if the window exists or fail if the window doesn't exist.

This action was written and kindly donated by Erik Berry, maintainer of GExperts



Window caption (optional)

Window class: TMainForm

Fail if window: Exists

6.27.28 Windows Page File

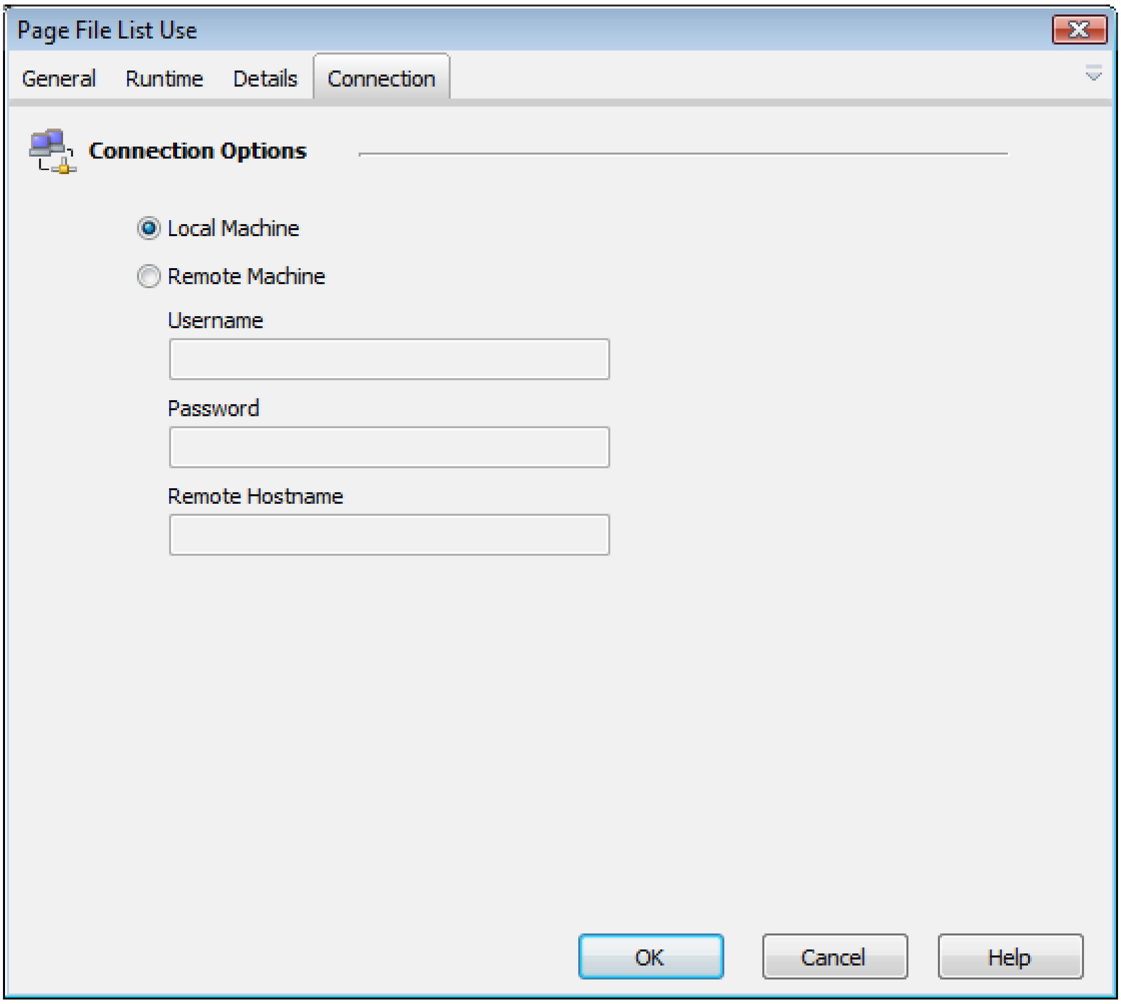
The Windows Page File actions can retrieve various information about the current page file configuration and performance.

The available actions are:

- Page File List Performance - retrieves specified or total page file current and maximum usage
- Page File List Properties - retrieves specified page file detailed configuration properties
- Page File List Use - retrieves specified page file usage information

There is some overlap with the capabilities of each of the actions - this is due to the objects retrieved via WMI queries.

All actions can either access the page file information on the local machine or on a network machine:



Page File List Use

Connection Options

● Local Machine

○ Remote Machine

Username

Password

Remote Hostname

Local Machine
The action will connect to the local machine, with the credentials of the current user.

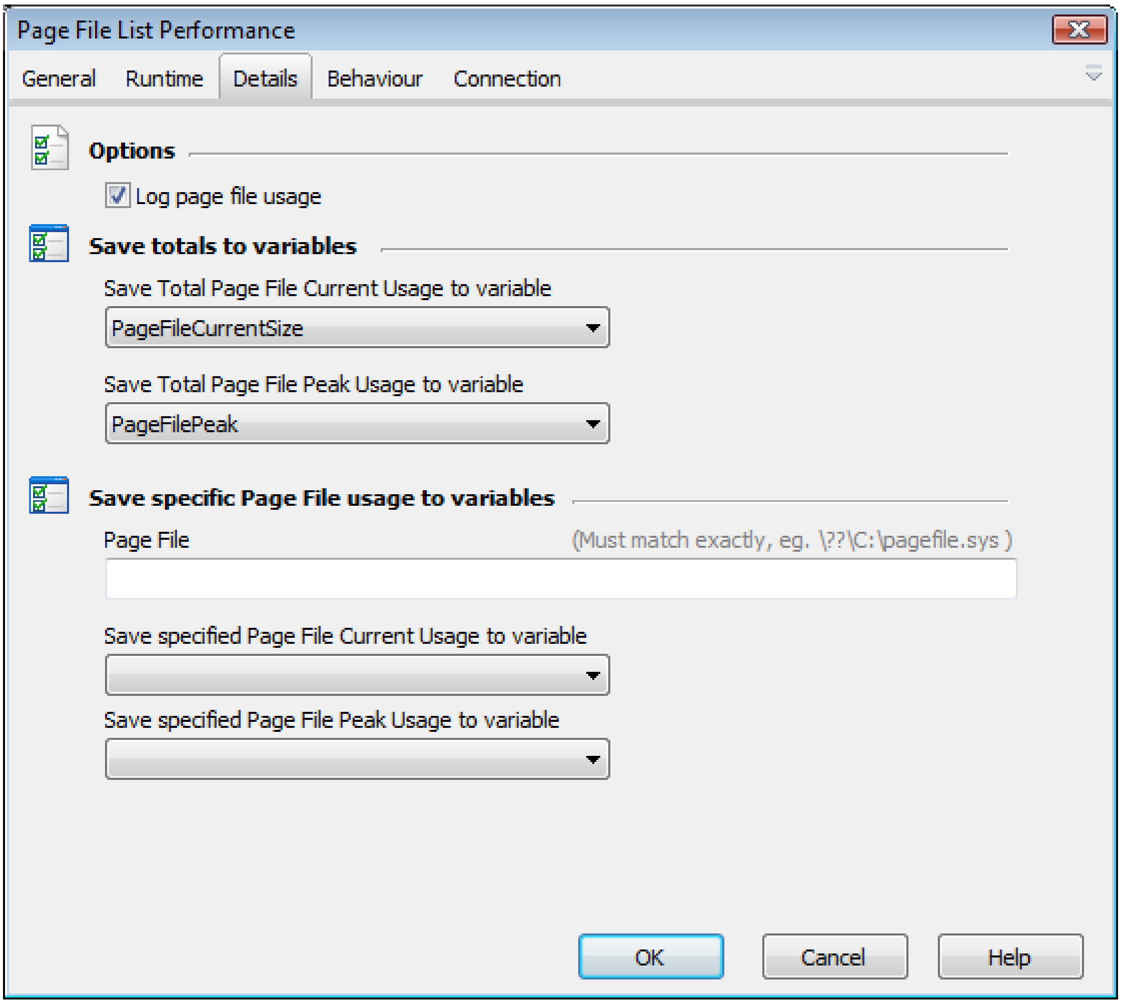
Remote Machine
The action will connect to a remote machine, using the specified user account.

Username
The username can be specified either by just the *username* or *DOMAIN/username*

Remote Host name
The computer which to connect to, if the remote computer is in a different domain then that of the current user account, a Fully Qualified Domain Name must be specified. An IPv4/IPv6 may also be specified.

6.27.28.1 Page File List Performance

The Page File List Performance action retrieves specified or total page file current and maximum usage, and can be used to alter the flow of the project based on certain performance characteristics.



Page File List Performance

Options

☑ Log page file usage

Save totals to variables

Save Total Page File Current Usage to variable: PageFileCurrentSize

Save Total Page File Peak Usage to variable: PageFilePeak

Save specific Page File usage to variables

Page File (Must match exactly, eg. \??\C:\pagefile.sys)

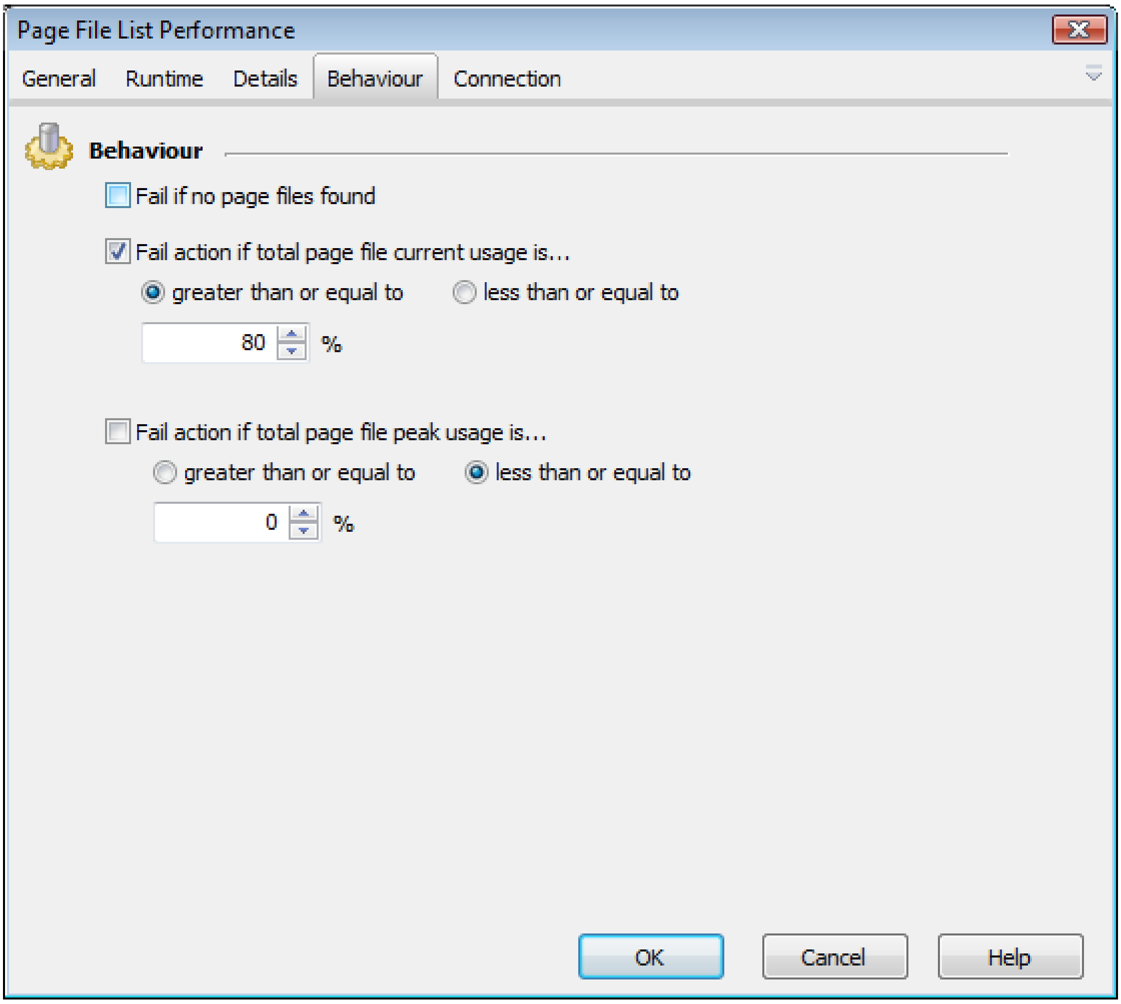
Save specified Page File Current Usage to variable

Save specified Page File Peak Usage to variable

Log page file usage
Will log all page files found along with their current and peak usage.

Save totals to variables
Specify the variables you want the current and peak usage of the total page file usage on the machine saved to.

Save specific page file usage to variables
Specify the page file and the variables you want the current and peak usage of the total page file usage on the machine saved to. If you are unsure of the exact name of the page file, you can run this action with the "Log page file usage" option set and it will report the names of the page files found.



☐ Fail if no page files found

☑ Fail action if total page file current usage is...

● greater than or equal to ○ less than or equal to

80 %

☐ Fail action if total page file peak usage is...

○ greater than or equal to ● less than or equal to

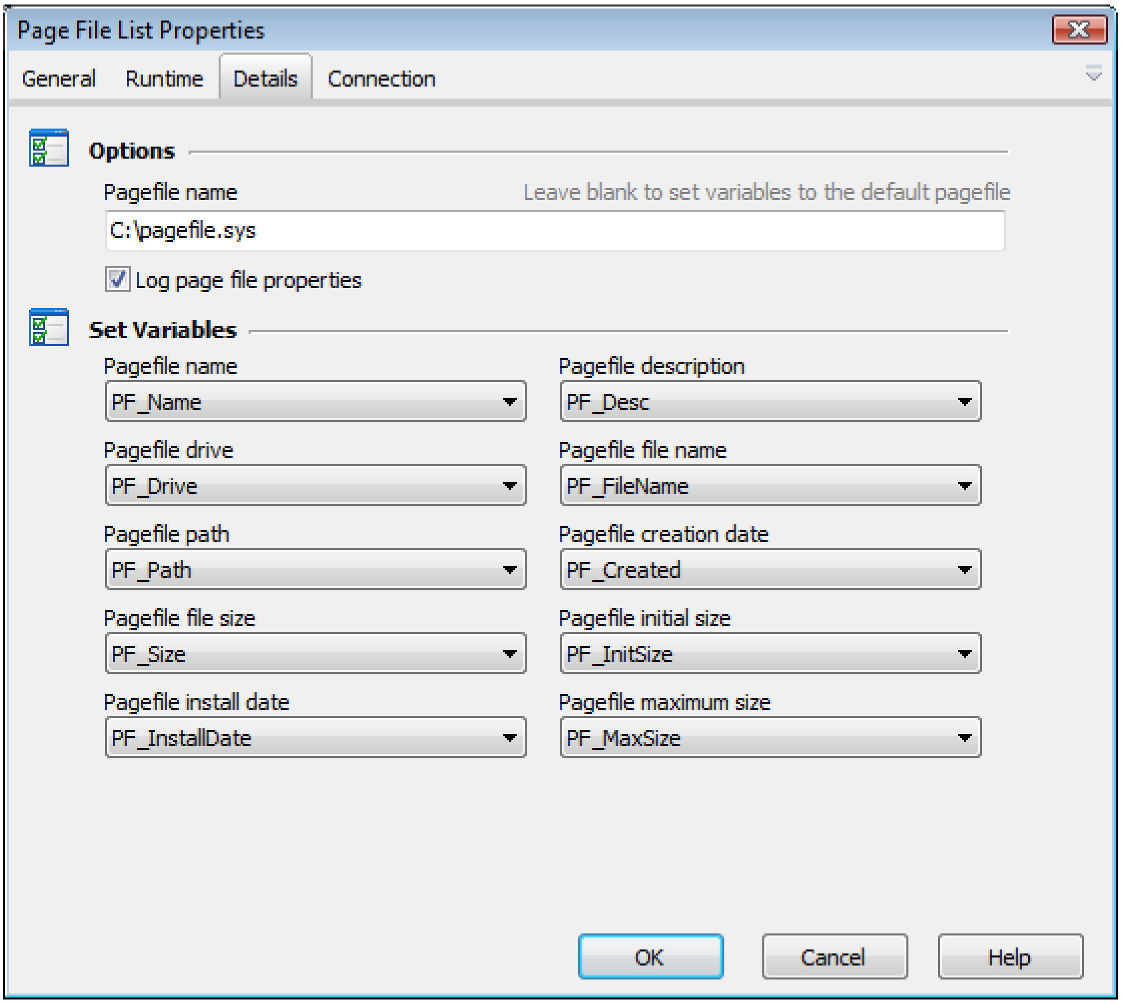
0 %

Fail if no page files found
The action will fail if no page files are found on the machine

The other "fail if..." options allow you to fail the action dependent on the current or peak usage of the total page file usage is above or below a certain threshold. The above screenshot shows that the action will be set to fail if the total peak page file usage is greater than or equal to 90% - an indicator that the page file probably isn't large enough.

6.27.28.2 Page File List Properties

The Page File List Properties action retrieves specified page file detailed configuration properties.



Options

Pagefile name: C:\pagefile.sys — Leave blank to set variables to the default pagefile

☑ Log page file properties

Set Variables

Pagefile name: PF_Name	Pagefile description: PF_Desc
Pagefile drive: PF_Drive	Pagefile file name: PF_FileName
Pagefile path: PF_Path	Pagefile creation date: PF_Created
Pagefile file size: PF_Size	Pagefile initial size: PF_InitSize
Pagefile install date: PF_InstallDate	Pagefile maximum size: PF_MaxSize

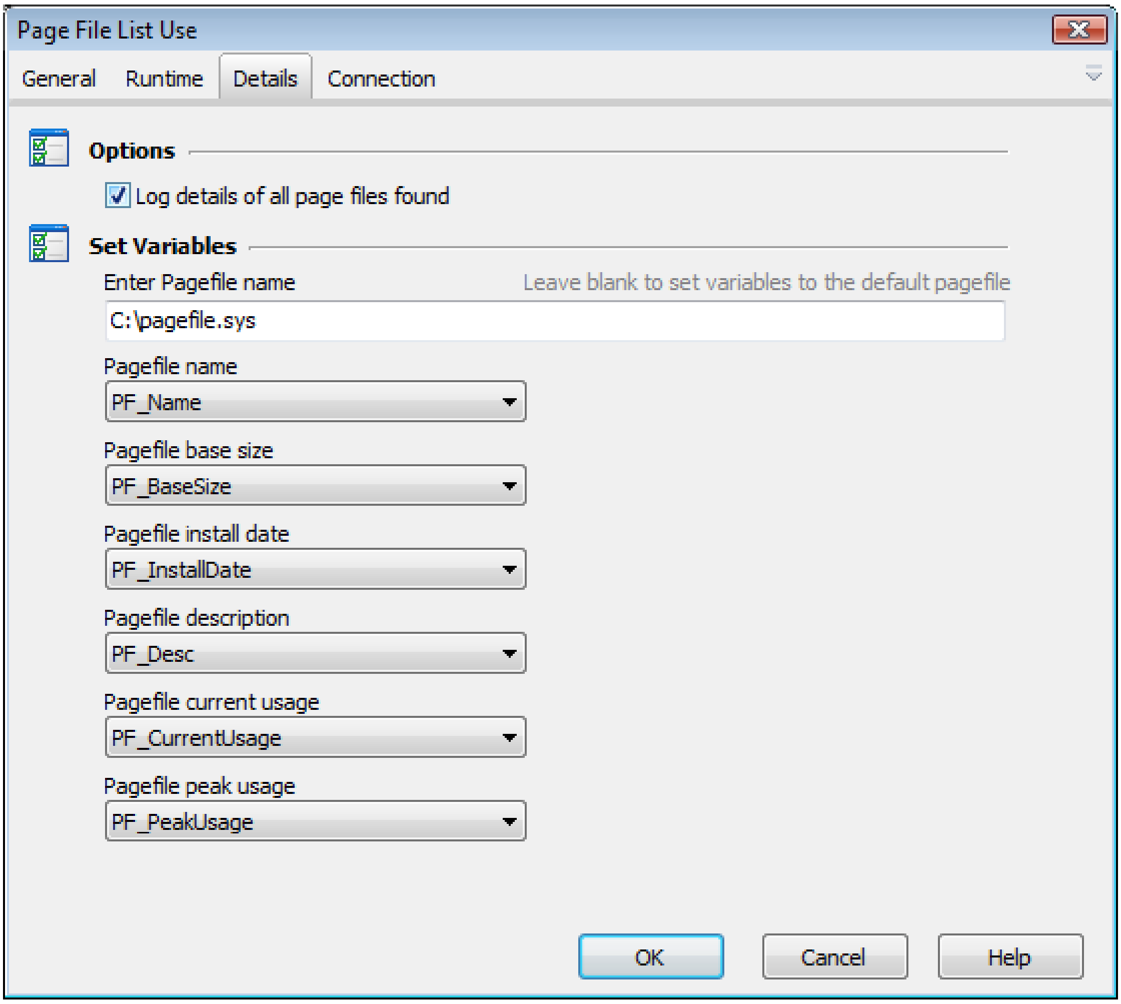
Page file name
Specify the page file name, or leave blank to get the properties of the default page file. The name is the full path to the page file, eg. c:\pagefile.sys

Log page file properties
Log all the available properties of the specified page file

Set Variables
Select variables to save the various page file configuration properties to. The above example shows that the PageFileDrive and PageFileInitialSize variables are going to be set.

6.27.28.3 Page File List Use

The Page File List Use action retrieves specified page file usage information.



Page File List Use

Options

☑ Log details of all page files found

Set Variables

Enter Pagefile name: C:\pagefile.sys (Leave blank to set variables to the default pagefile)

Pagefile name: PF_Name

Pagefile base size: PF_BaseSize

Pagefile install date: PF_InstallDate

Pagefile description: PF_Desc

Pagefile current usage: PF_CurrentUsage

Pagefile peak usage: PF_PeakUsage

Log details of all page files found
Log all the available properties of all page files found

Page file name
Specify the page file name, or leave blank to get the usage properties of the default page file. The name is the full path to the page file, eg. c:\pagefile.sys

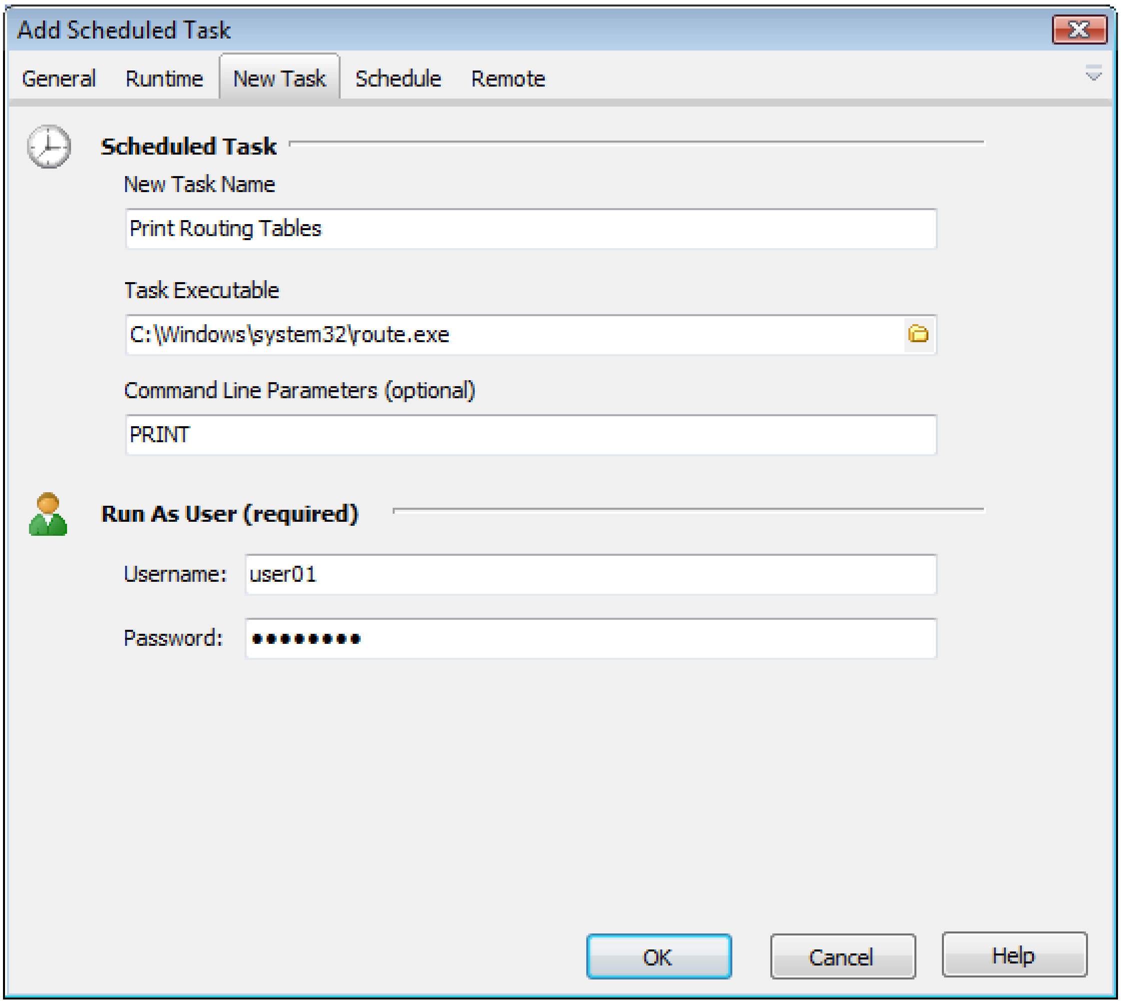
Set Variables
Select variables to save the various page file usage information to. The above example shows that the Pagefile install date information being captured.

6.27.29 Windows Scheduler

6.27.29.1 Add Scheduled Task

The Add Scheduled Task action enables you to schedule a task in the Windows task scheduler. This is equivalent to going to the Scheduled Tasks control panel and clicking Add Scheduled Task.

(Note that you can also schedule Automise projects with the Automise Scheduler.



Add Scheduled Task

Scheduled Task

New Task Name: Print Routing Tables

Task Executable: C:\Windows\system32\route.exe

Command Line Parameters (optional): PRINT

Run As User (required)

Username: user01

Password: ●●●●●●●●

New Task Name

Enter a name for your task. A task with this name must not already exist (if you need to replace a task, you can use a Delete Scheduled Task action before the Add Scheduled Task action.)

Task Executable

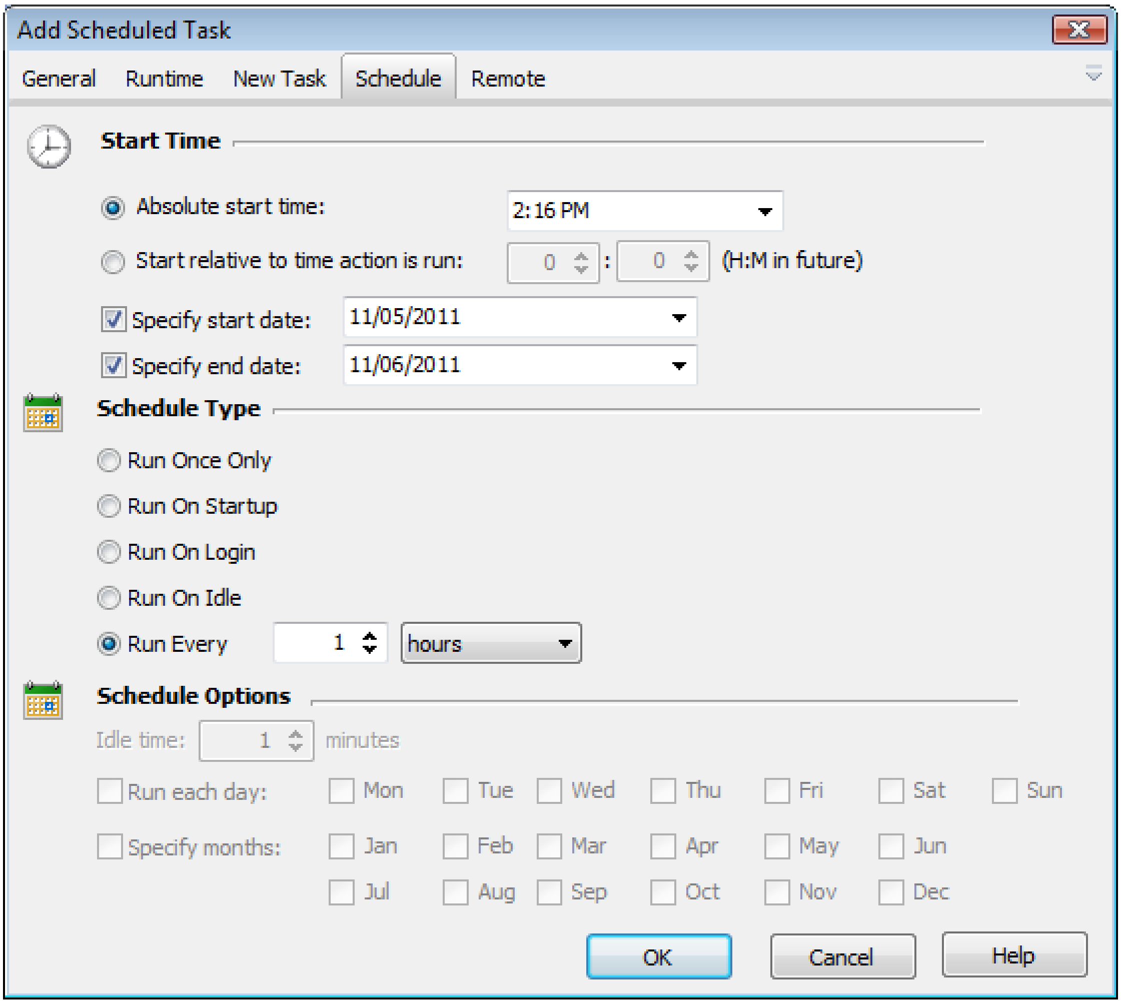
Enter the path to the executable you wish to schedule.

Command Line Parameters (optional)

Enter any parameters for the executable.

Run As User (required)

You must specify a username and password to run the scheduled task.



Add Scheduled Task

General | Runtime | New Task | Schedule | Remote

Start Time

● Absolute start time: 2:16 PM

○ Start relative to time action is run: 0 : 0 (H:M in future)

☑ Specify start date: 11/05/2011

☑ Specify end date: 11/06/2011

Schedule Type

○ Run Once Only

○ Run On Startup

○ Run On Login

○ Run On Idle

● Run Every 1 hours

Schedule Options

Idle time: 1 minutes

□ Run each day: □ Mon □ Tue □ Wed □ Thu □ Fri □ Sat □ Sun

□ Specify months: □ Jan □ Feb □ Mar □ Apr □ May □ Jun □ Jul □ Aug □ Sep □ Oct □ Nov □ Dec

Start Time

(This option is only available for "Once Only" or "Run Every" schedule types.)

You can either specify an absolute start time (ie 11:30PM) or a relative start time, which is calculated relative to the time the action is run. If you specify 24 hours or more, the task will be scheduled one or more days after the action is run.

Specify start date

(This option is only available for "Once Only" or "Run Every" schedule types.)

Specify the date when the scheduled task will first run.

Note that if you use "Start relative", and "Specify start date" together, then the action will start that many hours (or days) ahead of the start date specified.

Specify end date

(This option is only available for the "Run Every" schedule type.)

Specify the date on which the scheduled task will finish and be removed from the scheduler.

Schedule Type

"Run Once Only" - The action will run one time only, at the time/date provided.

"Run On Startup" - The action will run whenever the system starts up.

"Run On Login" - The action will run whenever the specified user logs in.

"Run On Idle" - The action will run once the computer has been idle for the specified number of minutes.

"Run Every" - The action will run repeatedly, first on the time/date specified and then at a certain interval until the end date (if one is given.)

Intervals can be expressed in minutes, hours, days, weeks or months.

Schedule Options

"Idle Time"
(This option is only available for Schedule Type "Run On Idle")

Specify the number of minutes that the computer needs to be idle before the task will run.

"Run each day"
(This option is only available for Schedule Type "Run Every x Weeks")

This option changes the behaviour of the weekly schedule so that, every 'x' week(s), the task will run once on each selected day.

"Specify months"
(This option is only available for Schedule Type "Run Every x Months")

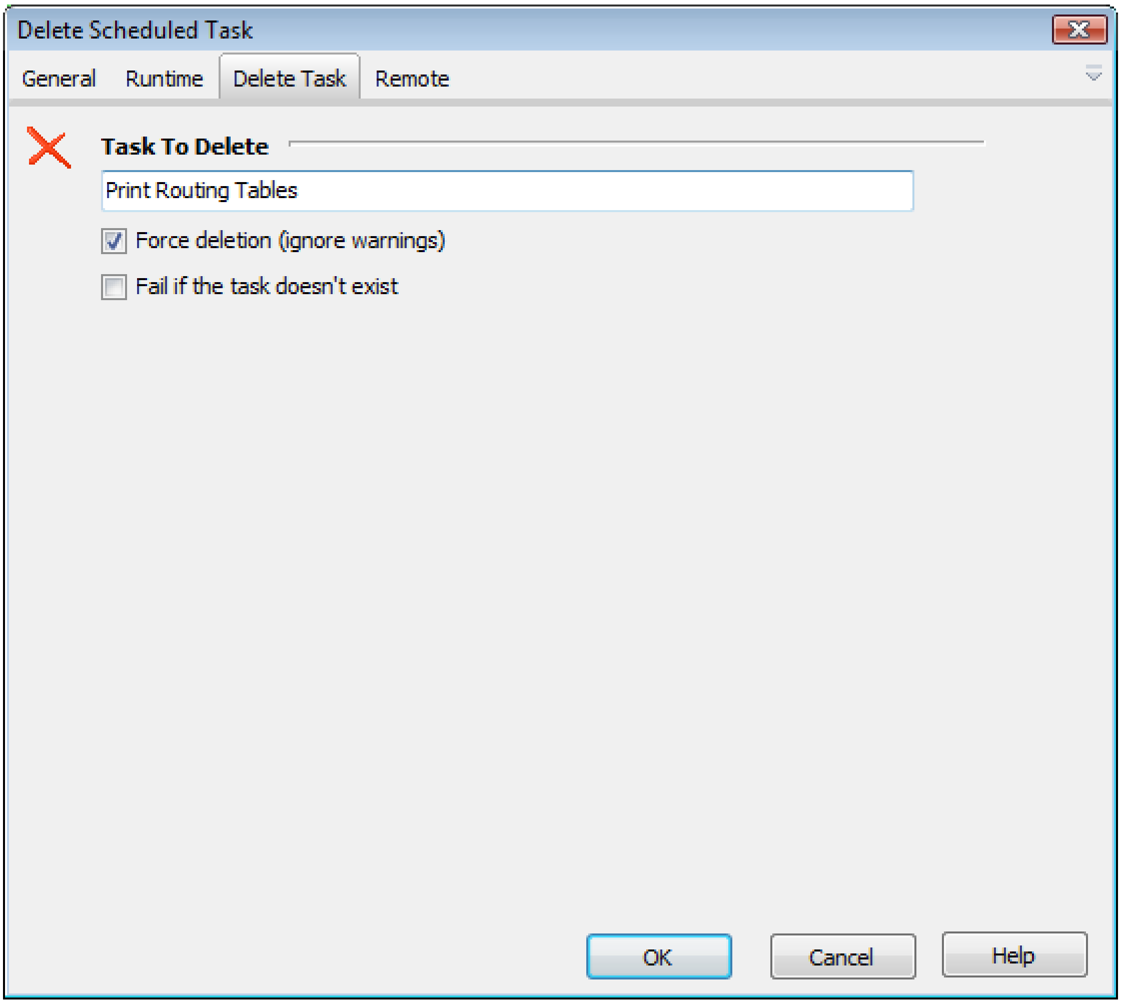
This option changes the behaviour of the monthly schedule so that the task will only run on the first day of certain months.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.27.29.2 Delete Scheduled Task

The Delete Scheduled Task action allows you to remove a scheduled task from the Windows Scheduler.



Delete Scheduled Task

Task To Delete: Print Routing Tables

☑ Force deletion (ignore warnings)

□ Fail if the task doesn't exist

Task To Delete

Enter the name of the task to remove.

Force deletion (ignore warnings)

Check this box to delete tasks which may be running or otherwise not available for deletion.

Fail if the task doesn't exist

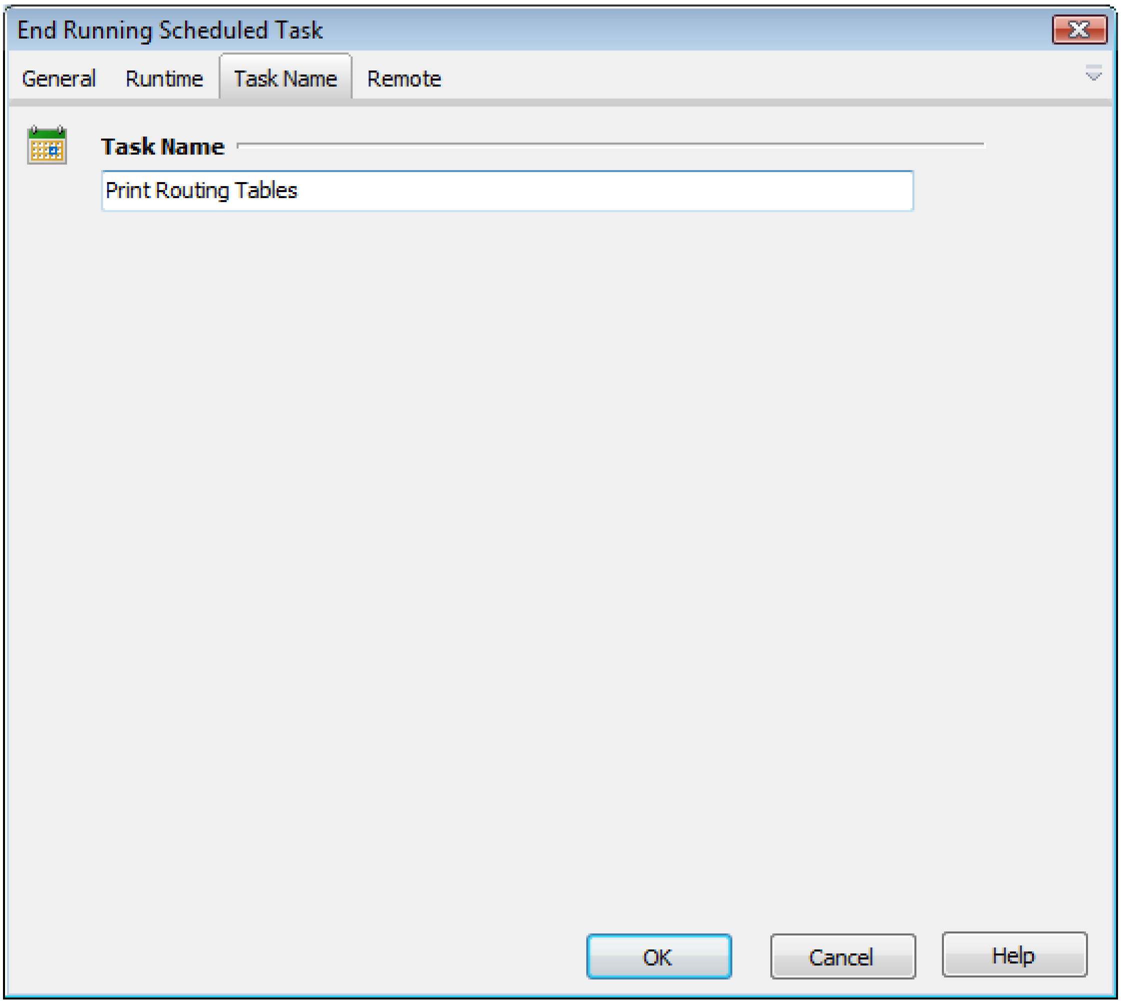
If this box is checked, the action will fail if the task is not found. Otherwise, the error will be ignored and the action will succeed.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.27.29.3 End Running Scheduled Task

The End Scheduled Task explicitly ends a task if it is running. If the task is not running, End Scheduled Task will return an error.



Task Name

Print Routing Tables

Task Name

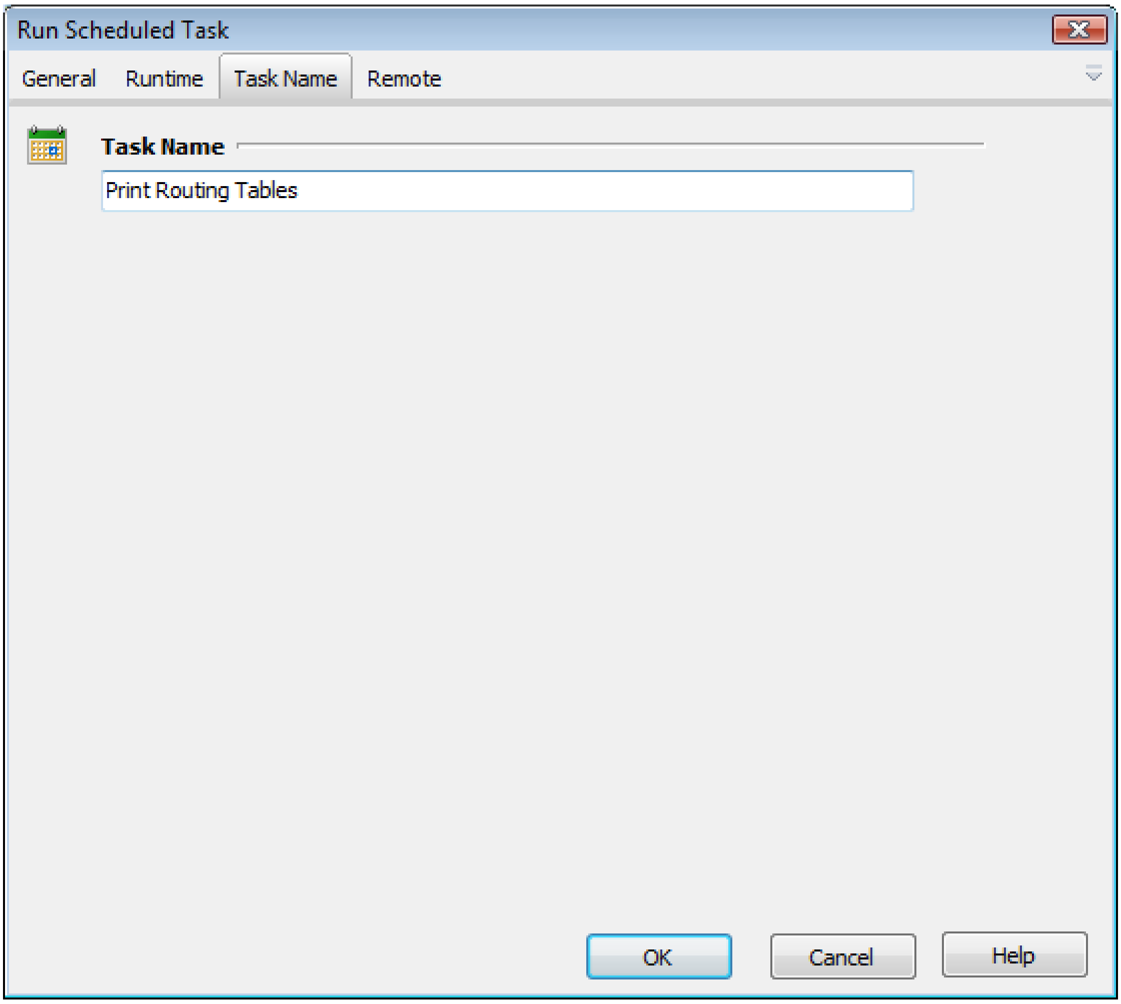
Enter the name of the running task to stop.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.27.29.4 Run Scheduled Task

The Run Scheduled Task action explicitly launches a scheduled task immediately.



Task Name

Print Routing Tables

Task Name

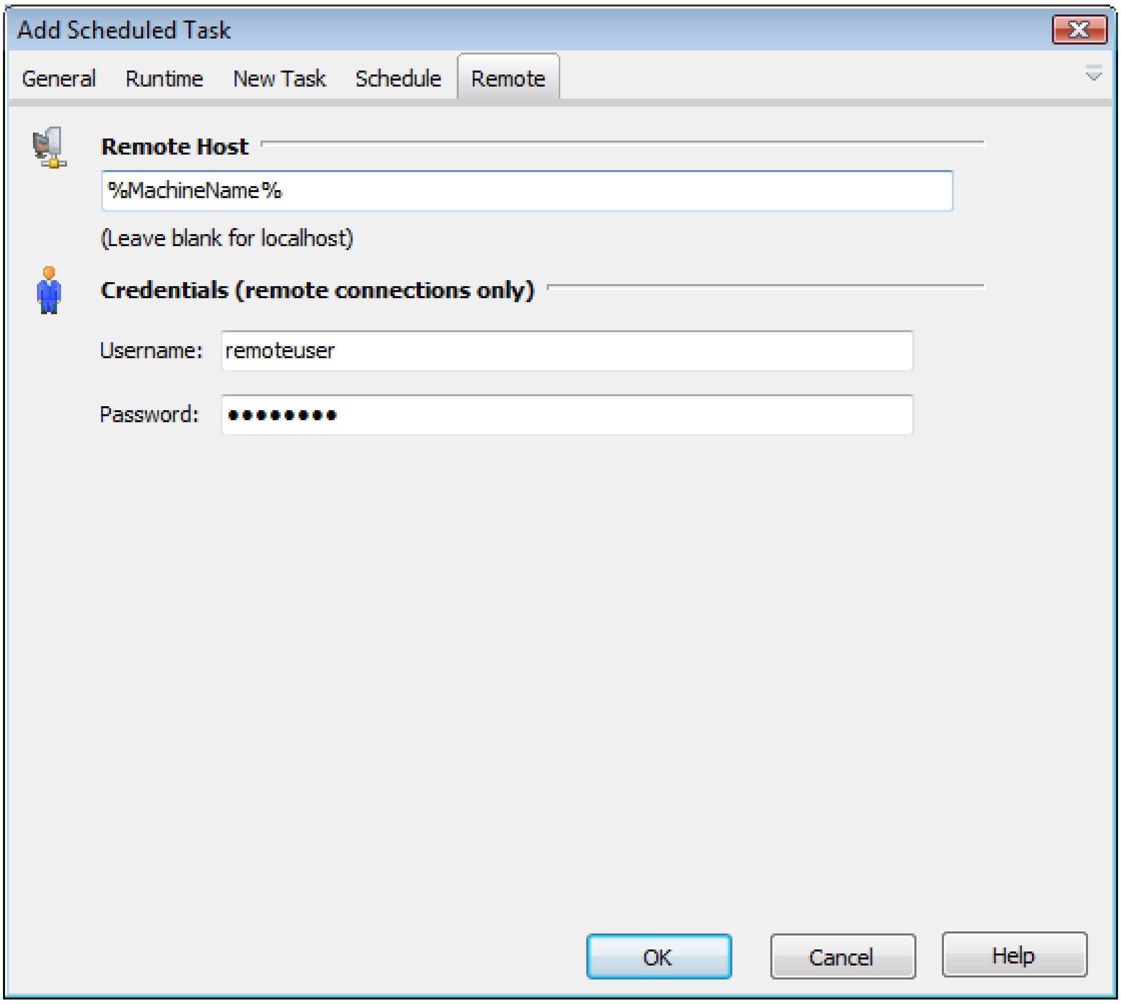
Enter the name of a task to run.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.27.29.5 Scheduling Remote Tasks

All of the Windows Scheduler actions contain a Remote tab, which allows you to schedule tasks on remote machines:



Add Scheduled Task

Remote Host

%MachineName%

(Leave blank for localhost)

Credentials (remote connections only)

Username: remoteuser

Password: •••••••••

Remote Host

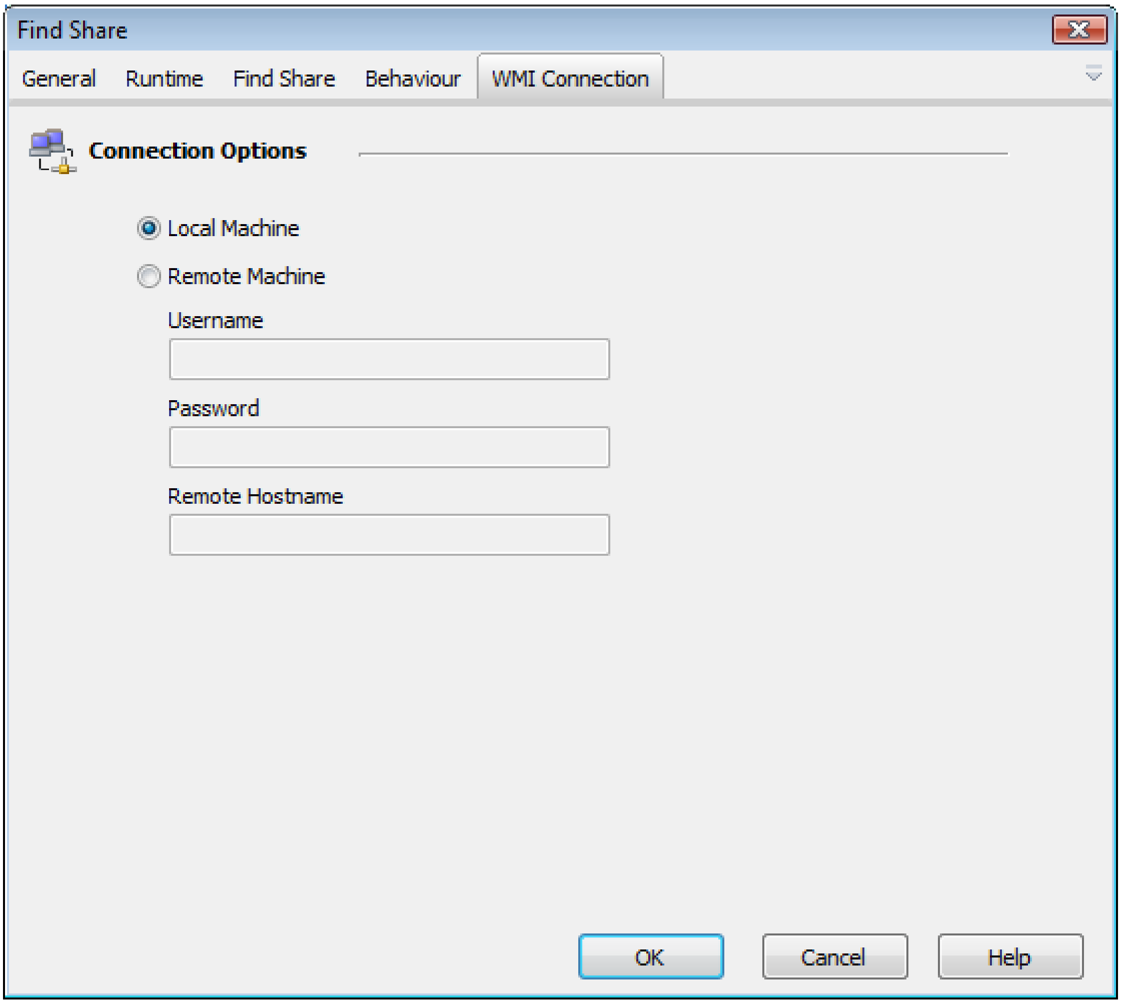
Specify the name of a remote host which allows remote scheduling. Leave blank to select the local machine.

Credentials

Credentials are only needed when connecting to remote computers (they cannot be used locally.) Specify a username and password which is valid on the remote machine. The user will need sufficient privileges to create remote tasks.

6.27.30 Windows Shares

The Windows Shares action enables the administration of shares, either on a local or remote machine.



● Local Machine

○ Remote Machine

Username

Password

Remote Hostname

Local Machine
The action will connect to the local machine, with the credentials of the current user.

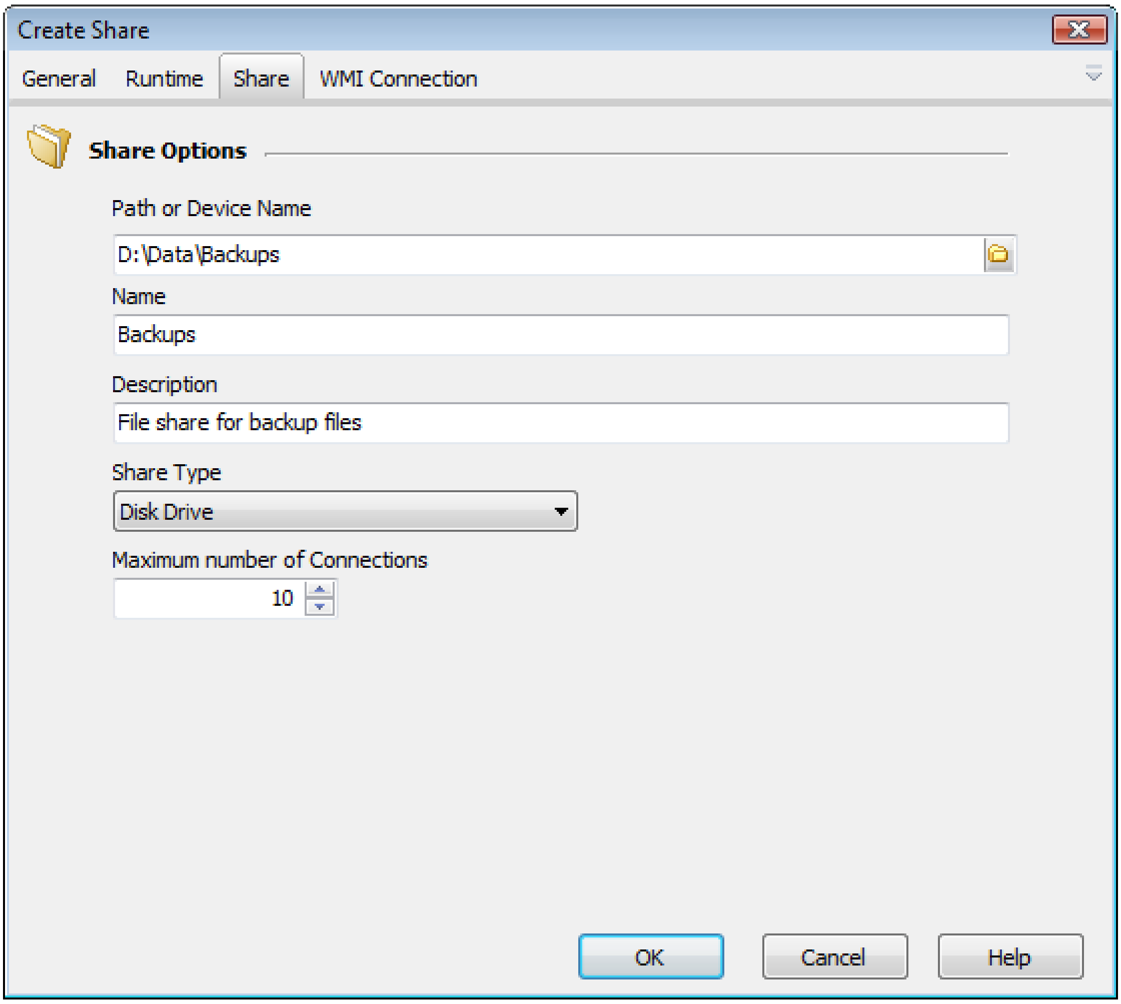
Remote Machine
The action will connect to a remote machine, using the specified user account.

Username
The username can be specified either by just the *username* or *DOMAIN/username*

Remote Host name
The computer which to connect to, if the remote computer is in a different domain then that of the current user account, a Fully Qualified Domain Name must be specified. An IPv4/IPv6 may also be specified.

6.27.30.1 Create Share

The Create Share action enables the creation of a File, Printer, IPC or Device share either on a local or remote machine.



Create Share

Share Options

Path or Device Name: D:\Data\Backups

Name: Backups

Description: File share for backup files

Share Type: Disk Drive

Maximum number of Connections: 10

Path
Local path of the Windows share.

Name
The name of the share.

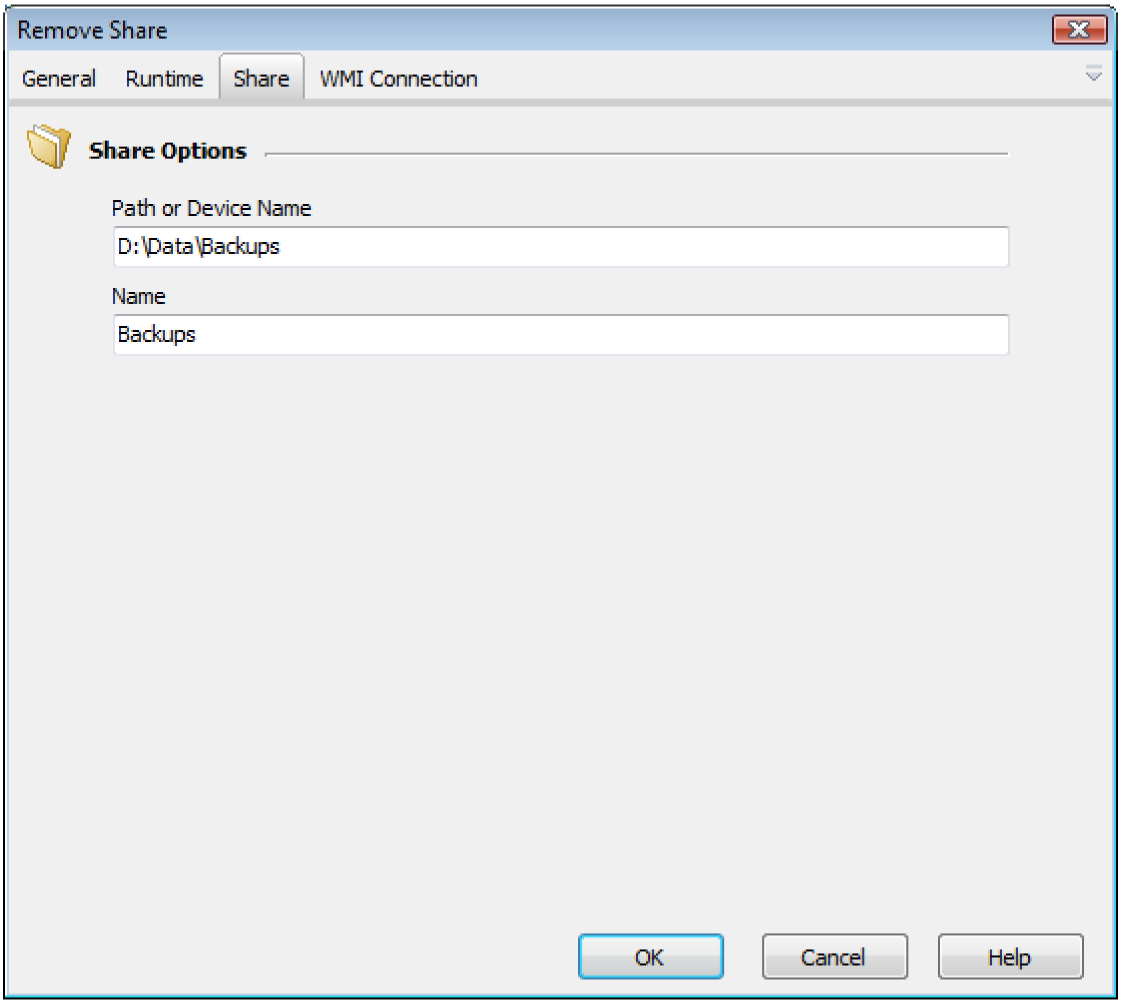
Description
Optional comment to describe the share.

Type
The type of share to create, possible values are Disk Drive, Print Queue, Device or IPC.

Maximum number of Connections
The limit on the maximum number of connections are allowed concurrently to the share.

6.27.30.2 Remove Share

The Remove Share action enables the removal of a File, Printer, IPC or Device share either on a local or remote machine.



Path or Device Name: D:\Data\Backups

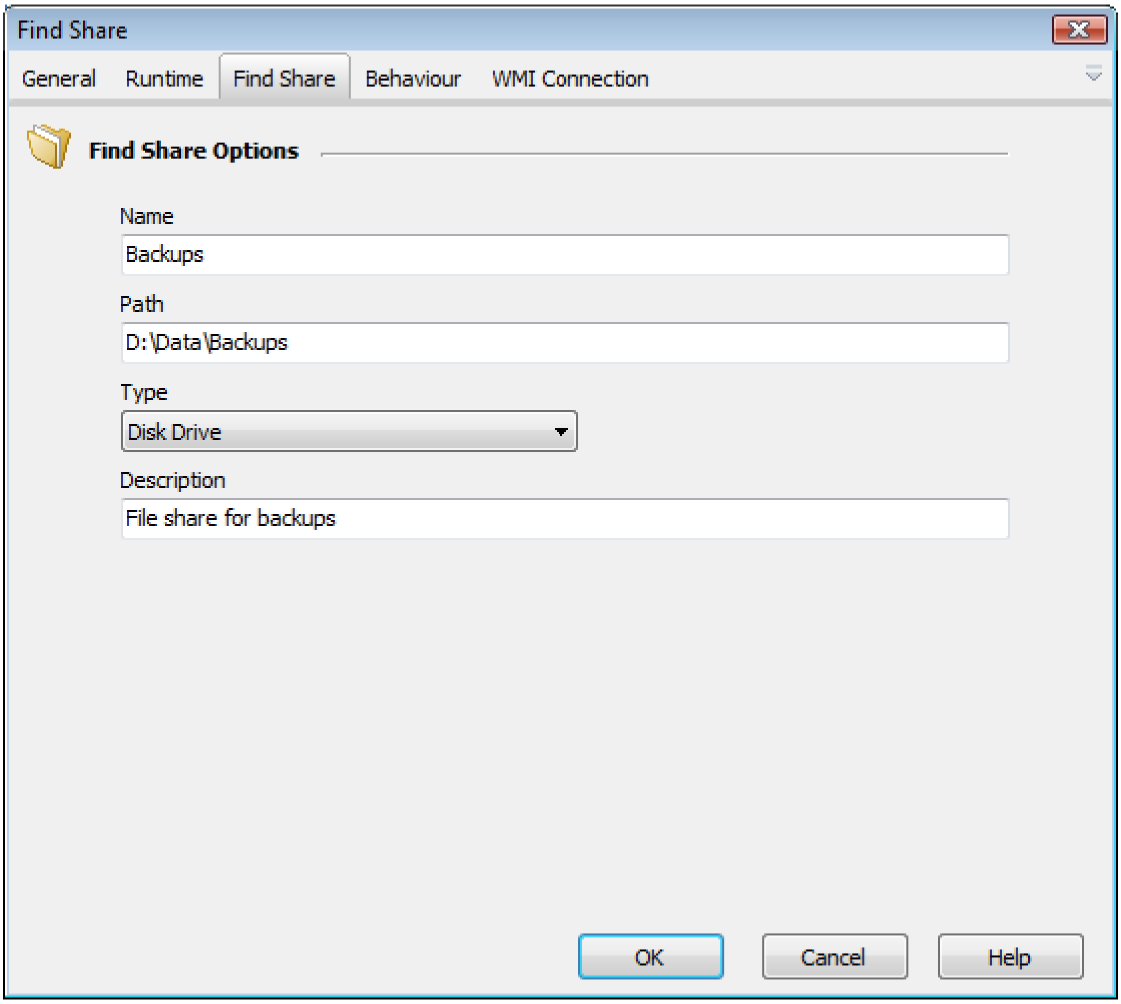
Name: Backups

Path
The local path of the Windows share.

Name
The name of the share.

6.27.30.3 Find Share

The Find Share action allows for searching of shares either on the local or a remote machine.



Name: Backups

Path: D:\Data\Backups

Type: Disk Drive

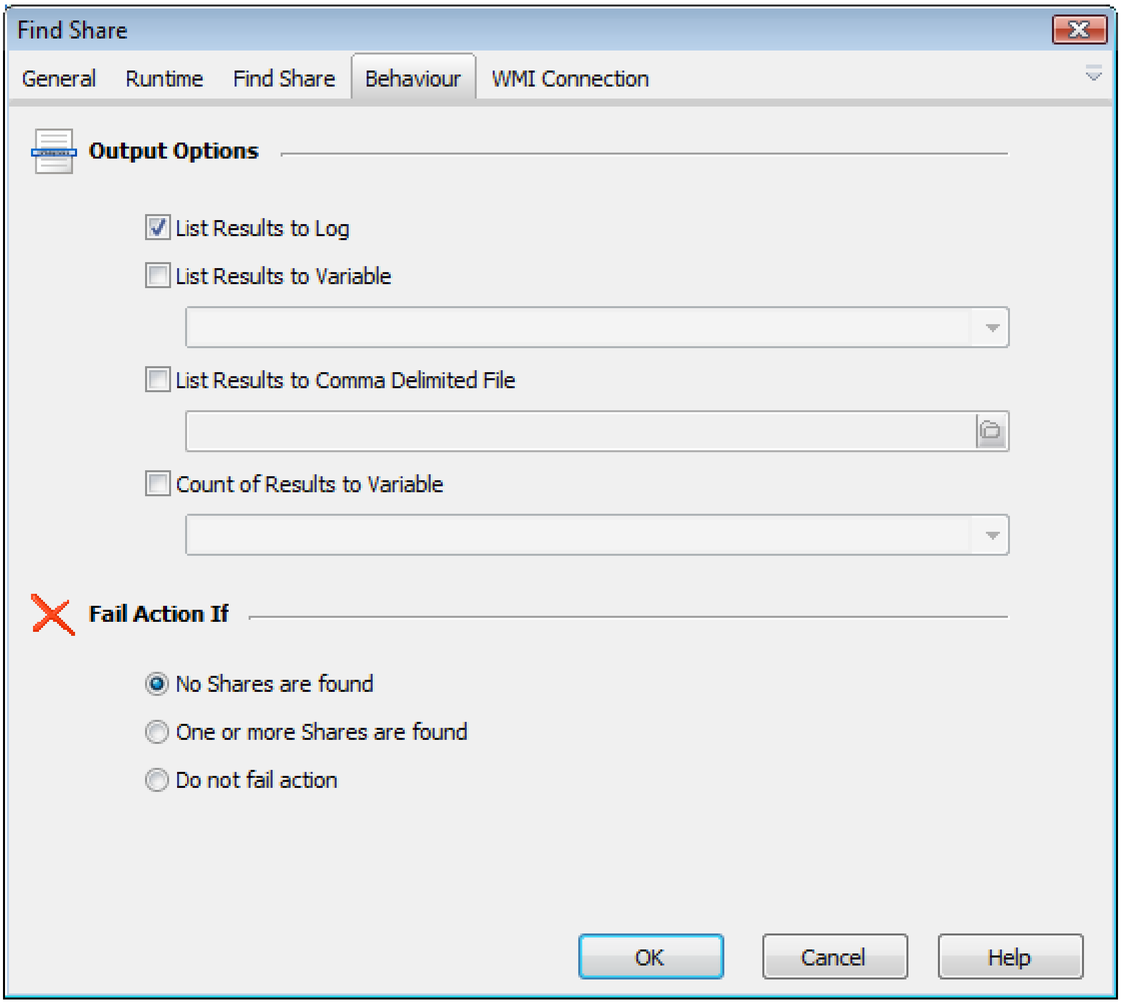
Description: File share for backups

Path
Local path of the Windows share.

Name
The name of the share.

Type
The type of share to find, possible values are Disk Drive, Print Queue, Device or IPC.

Description
The description of the share.



Find Share

Output Options

☑ List Results to Log

□ List Results to Variable

□ List Results to Comma Delimited File

□ Count of Results to Variable

Fail Action If

● No Shares are found

○ One or more Shares are found

○ Do not fail action

List Results to Log
Enables the listing of the results in the log. Turning this option off is recommend when expecting a high number of results to be returned.

List Results to Variable
Saves the results of the search to the specified variable.

List Results to Comma Delimited Text File
Outputs the log to a text file with each field separated by a comma.

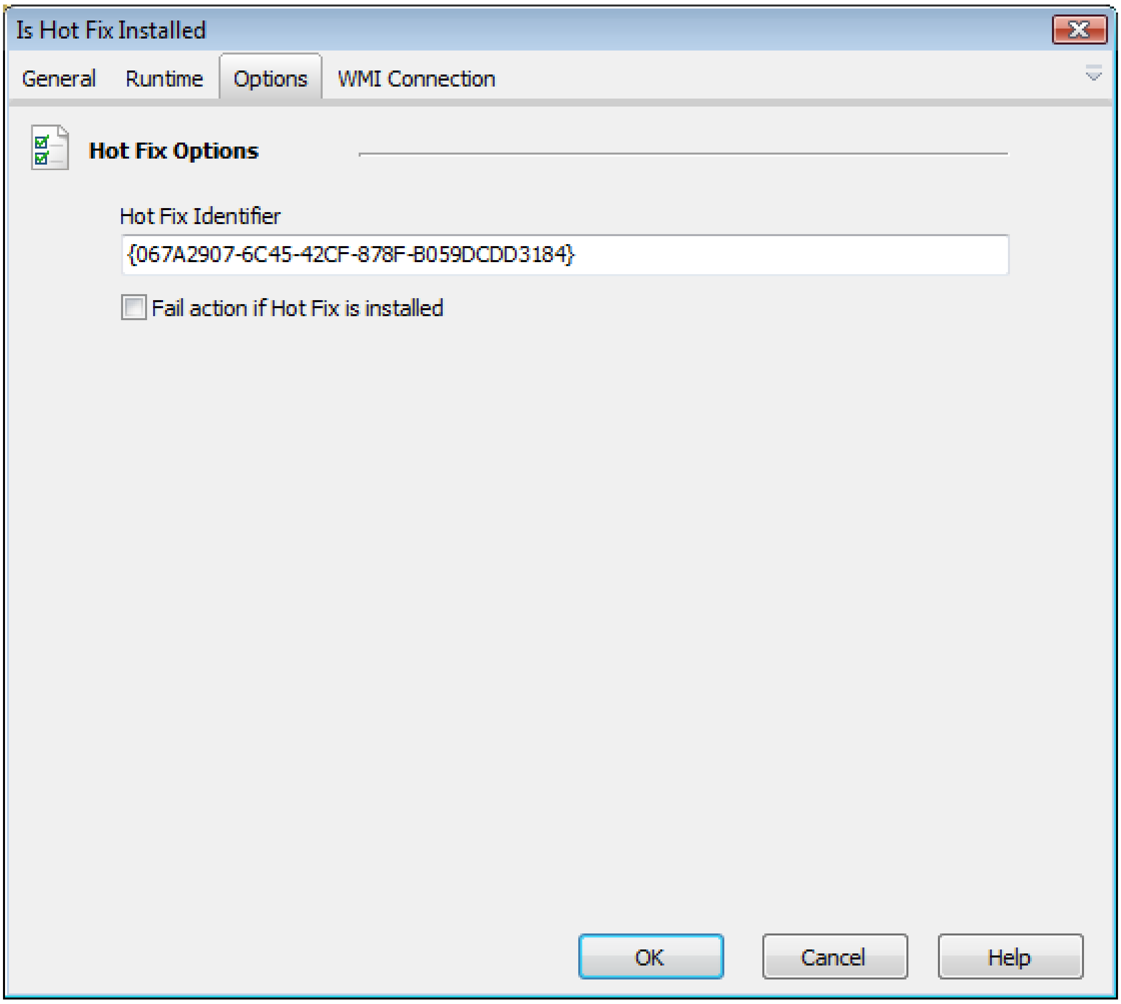
Count of Results to Variable
Writes to number of matched entries to the specified variable.

Fail Action If
Specifies whether the action will fail if there were shares found that matched the criteria or fail if no matches were found.

6.27.31 Windows Updates

6.27.31.1 Is Hot Fix Installed

The Is Hot Fix Installed action checks to see whether a hot fix has been installed on a windows based computer.



Hot Fix Identifier
{067A2907-6C45-42CF-878F-B059DCDD3184}

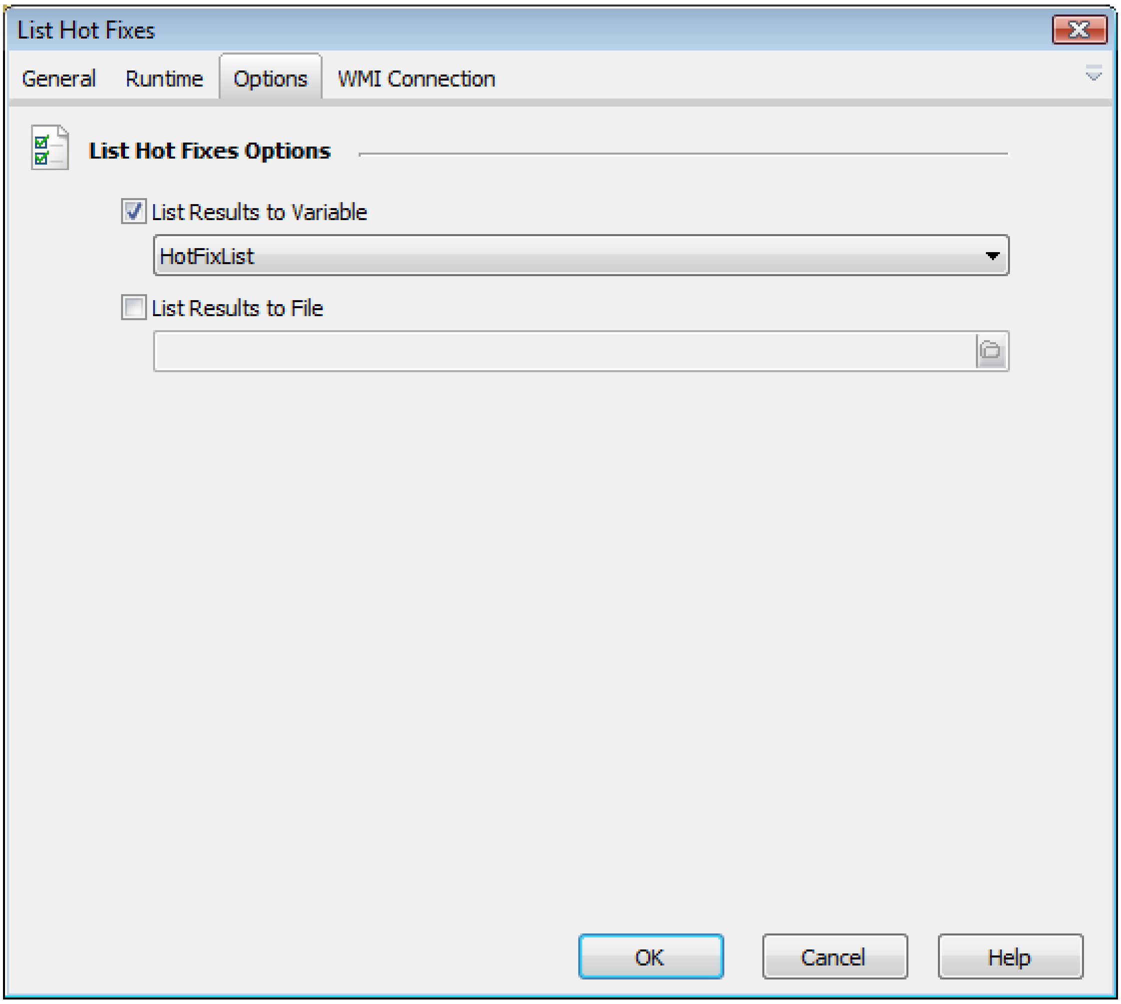
□ Fail action if Hot Fix is installed

Hot Fix Identifier
The ID of the hot fix. For example - 'KB911280'.

Fail action if Hot Fix is installed
Inverts the behaviour of the action so that it will fail if the hot fix is installed.

6.27.31.2 List Hot Fixes

The List Hot Fix action will list all the hot fixes installed on a windows based computer.



☑ List Results to Variable: HotFixList

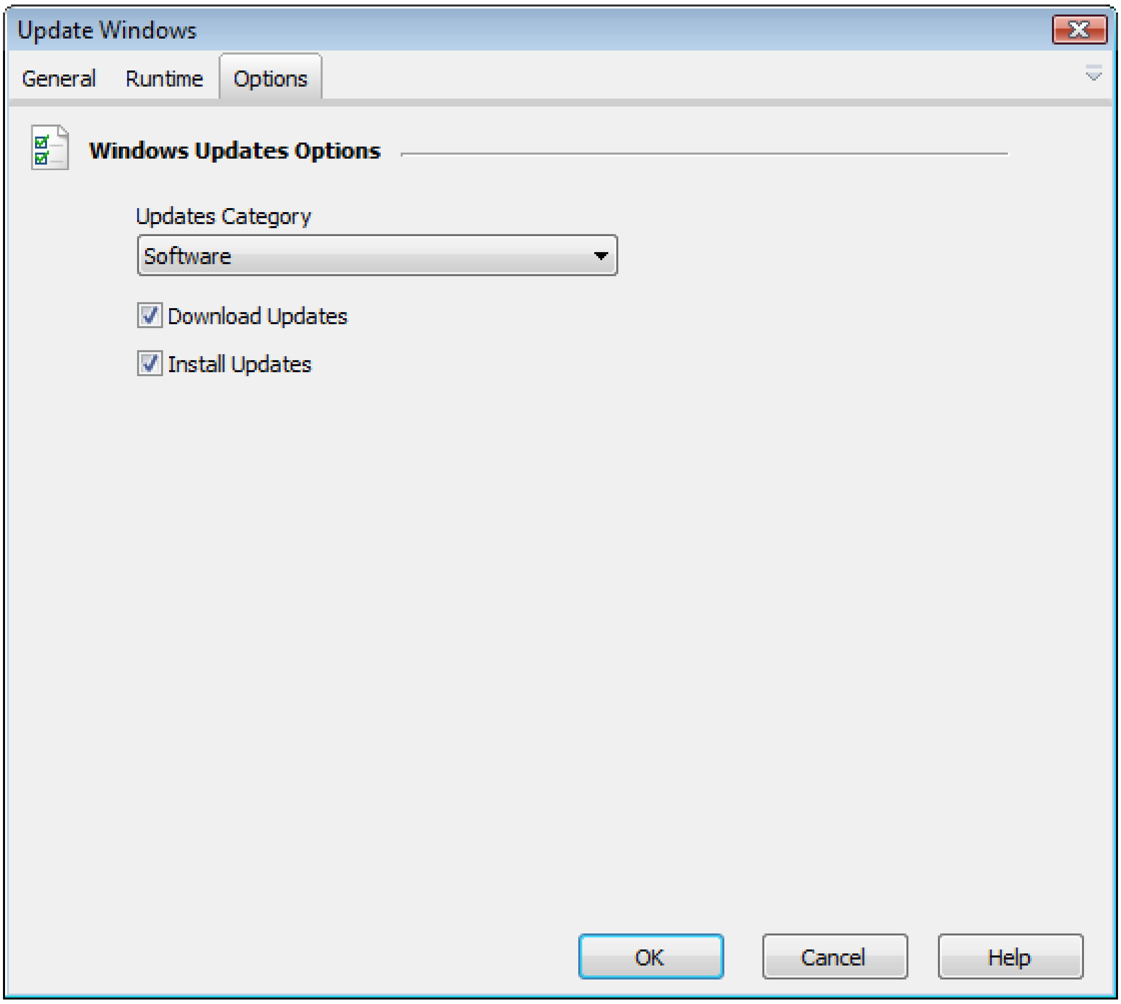
□ List Results to File

List Results to Variable
Lists all the hot fixes to the specified variable.

List Results to File
Appends a list of all the hot fixes installed to the specified text file.

6.27.31.3 Update Windows

The Update Windows action allows you to download and install updates for the Windows Operating System.



Update Windows

Windows Updates Options

Updates Category: Software

☑ Download Updates

☑ Install Updates

Download Updates
All available updates for your system will be downloaded, if the update has already been downloaded but not installed, it will not be downloaded again.

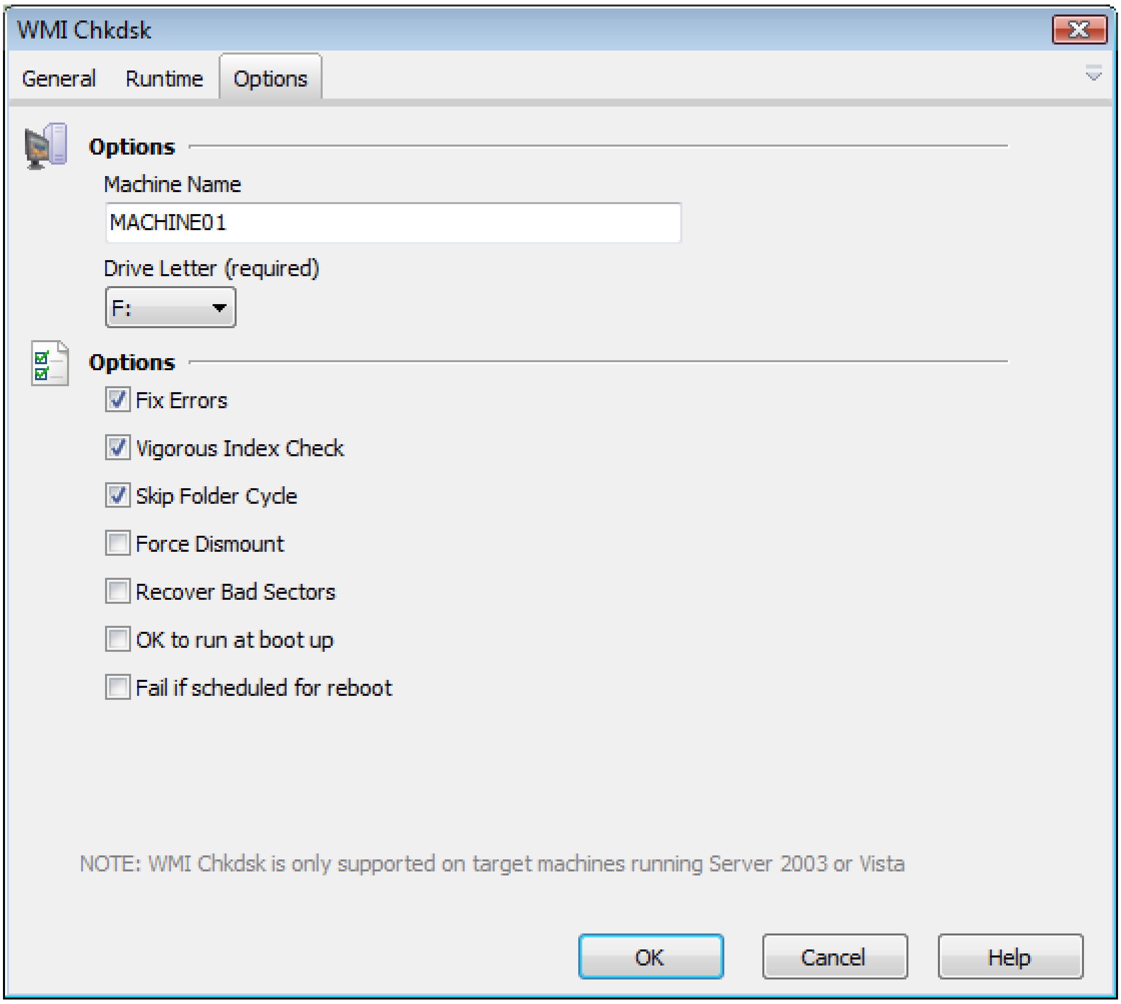
Install Updates
Install the updates after they have been downloaded.

6.27.32 WMI Actions

6.27.32.1 WMI Chkdsk

The WMI Chkdsk action invokes the chkdsk operation on the disk via WMI. The method is applicable to only those instances of logical disk that represent a physical disk in the machine. It is not applicable to mapped logical drives.

Client - Requires Windows Vista or Windows XP.
Server - Requires Windows Server "Longhorn" or Windows Server 2003.



Options

Machine Name: MACHINE01

Drive Letter (required): F:

Options

☑ Fix Errors

☑ Vigorous Index Check

☑ Skip Folder Cycle

□ Force Dismount

□ Recover Bad Sectors

□ OK to run at boot up

□ Fail if scheduled for reboot

NOTE: WMI Chkdsk is only supported on target machines running Server 2003 or Vista

Machine Name - Enter the name of the remote machine, or leave blank for localhost

Drive Letter - Select the drive letter. If you want to set this dynamically, set it in the BeforeAction script event. See below.

Fix Errors - Indicates what should be done to errors found on the disk. If true, then errors are fixed. The default is false.

Vigorous Index Check - If true, a vigorous check of index entries should be performed. The default is true.

Skip Folder Cycle - If true, the folder cycle checking should be skipped. The default is true.

Force Dismount - If true, the drive should be forced to dismount before checking. The default is false.

Recover Bad Sectors - If true, the bad sectors should be located and the readable information should be recovered from these sectors. The default is false.

OK To Run At Boot Up - If true, the chkdsk operation should be performed at next boot up time, in case the operation could not be performed because the disk is locked at time the method is called. The default is false.

Fail if scheduled for reboot - If true and the chkdsk returns that it has scheduled to run at boot up time, then the action will fail

Scripting Notes.

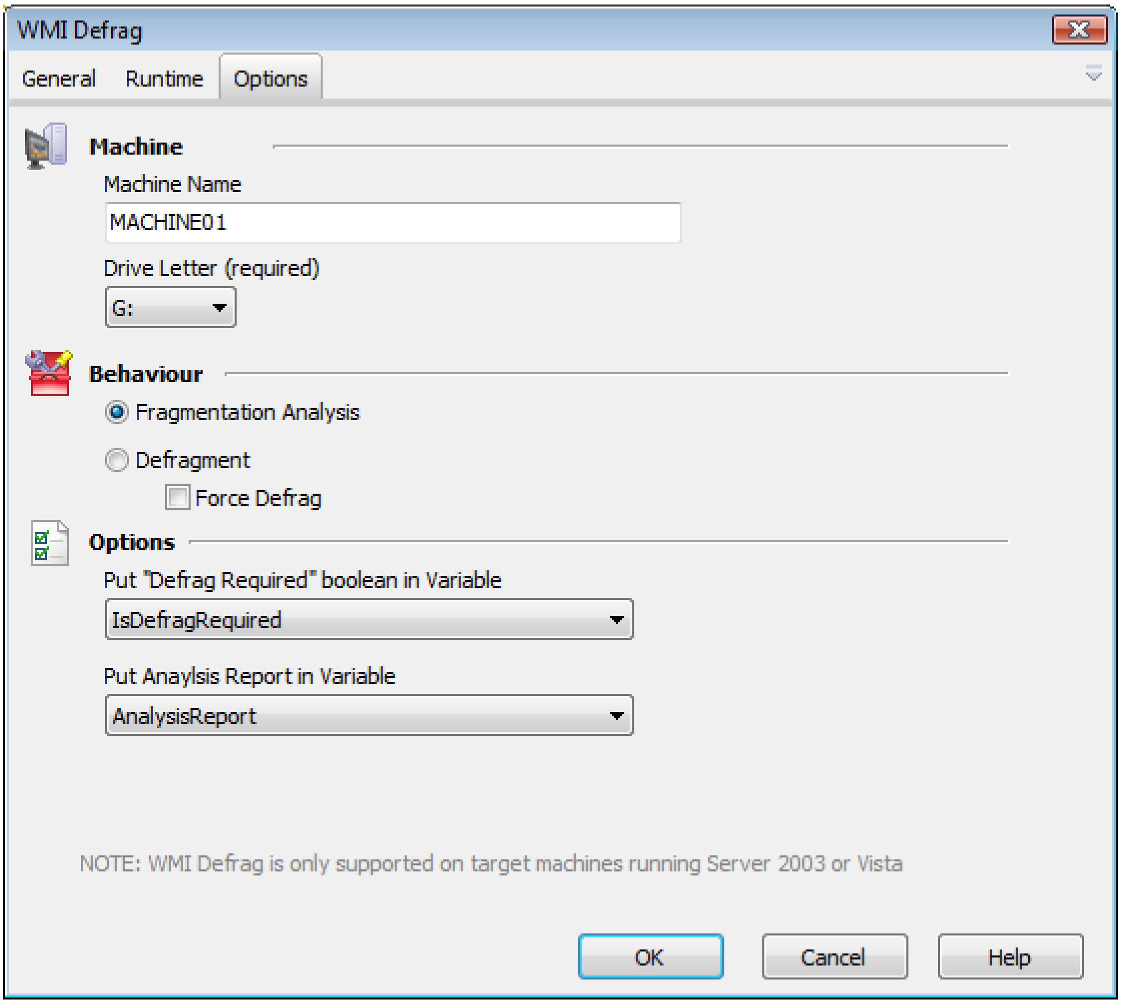
You can set the various properties of the action via script. For example, to set the Drive Letter via script:
BeforeAction Script Event:
Action.PropertyStore.DriveLetter = TheDriveLetter

Where "TheDriveLetter" is a variable that has been set to a valid drive on the machine, in the format <DriveLetter>:
eg. E: or C:

6.27.32.2 WMI Defrag

The WMI Defrag action allows you to analyse the fragmentation and perform a defragmentation of a logical drive via WMI.

The WMI Defrag only works on Windows 2003 and Windows Server "Longhorn" operating systems. To Defrag on other OS's, use the Defrag Disk action.



Machine

Machine Name: MACHINE01

Drive Letter (required): G:

Behaviour

● Fragmentation Analysis

○ Defragment

□ Force Defrag

Options

Put "Defrag Required" boolean in Variable: IsDefragRequired

Put Anaylsis Report in Variable: AnalysisReport

NOTE: WMI Defrag is only supported on target machines running Server 2003 or Vista

Machine Name - Enter the name of the remote machine, or leave blank for localhost

Drive Letter - Select the drive letter. If you want to set this dynamically, set it in the BeforeAction script event. See below.

Behaviour - select if you want to perform an analysis or a defragmentation

Force Defrag - if the volume doesn't require defragmentation, setting this option will force the defrag anyway

Put Defrag Required in variable - if performing an analysis, the recommendation will be set to this variable, true for defrag recommended

Put Analysis Report in variable - select the variable to save the analysis report to. The report will also be output to the log.

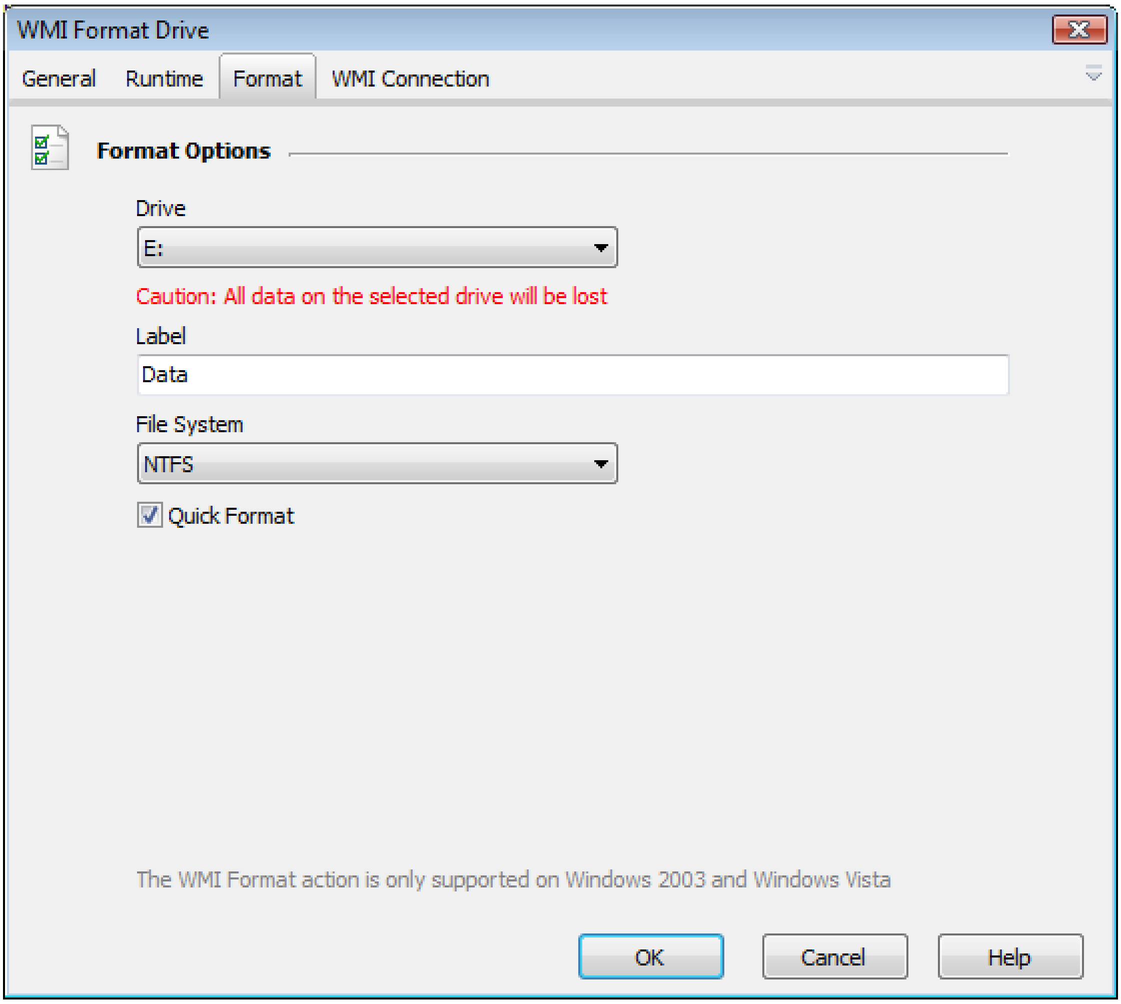
Scripting Notes.

You can set the various properties of the action via script. For example, to set the Drive Letter via script:
BeforeAction Script Event:
Action.PropertyStore.DriveLetter = TheDriveLetter

Where "TheDriveLetter" is a variable that has been set to a valid drive on the machine, in the format <DriveLetter>:
eg. E: or C:

6.27.32.3 WMI Format Drive

The WMI Format Drive action allows you to format a drive on a local or remote system.



WMI Format Drive

Format Options

Drive

E:

Caution: All data on the selected drive will be lost

Label

Data

File System

NTFS

☑ Quick Format

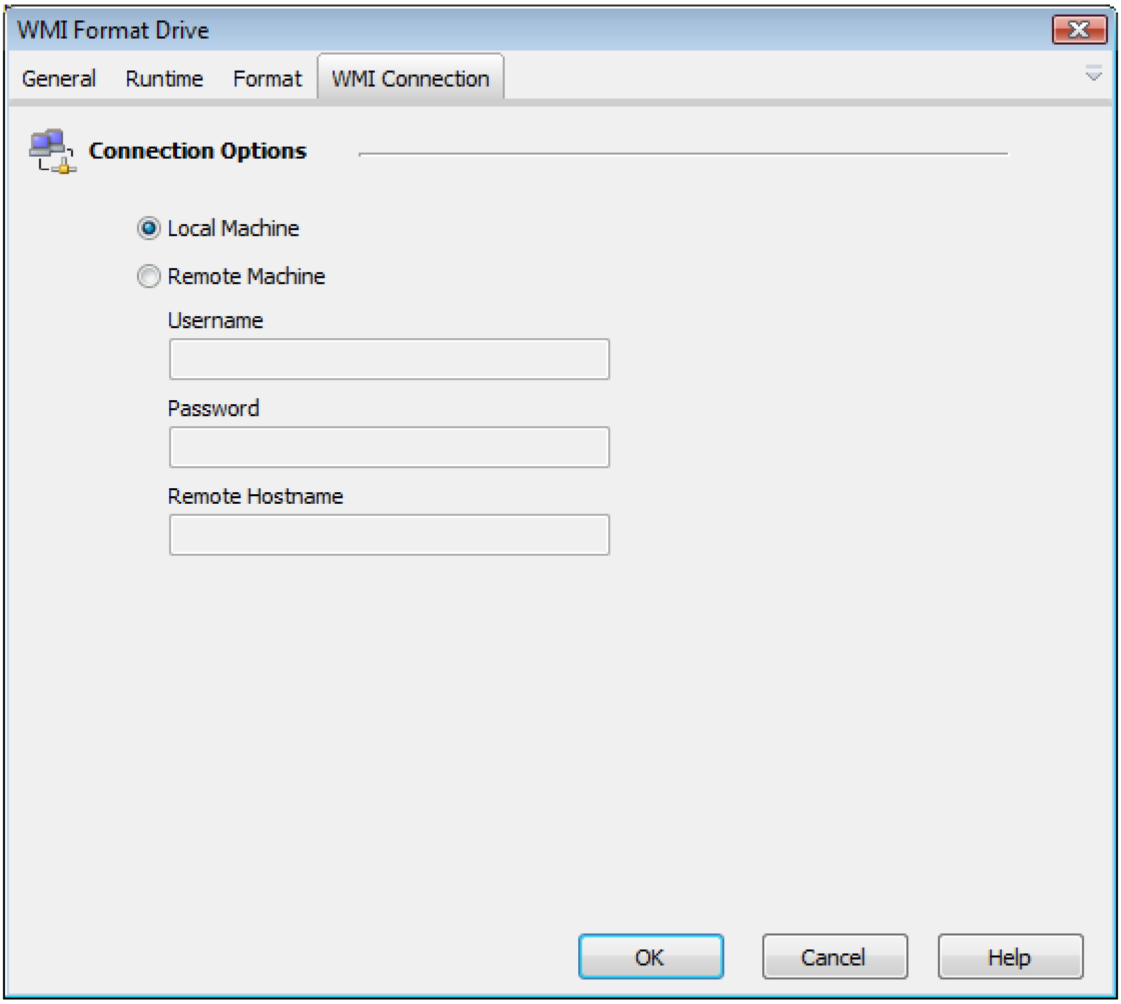
The WMI Format action is only supported on Windows 2003 and Windows Vista

Drive
The disk to be formatted. The disk can be specified by its mount point, volume name or drive letter.

Label
The volume label of the disk.

File System
The file system which the disk will be formatted as. Possible file systems are Fat, Fat32 and NTFS.

Quick Format
Performs a quick format of the disk by not checking for bad sectors.



WMI Format Drive

Connection Options

● Local Machine

○ Remote Machine

Username

Password

Remote Hostname

Local Machine
The action will connect to the local machine, with the credentials of the current user.

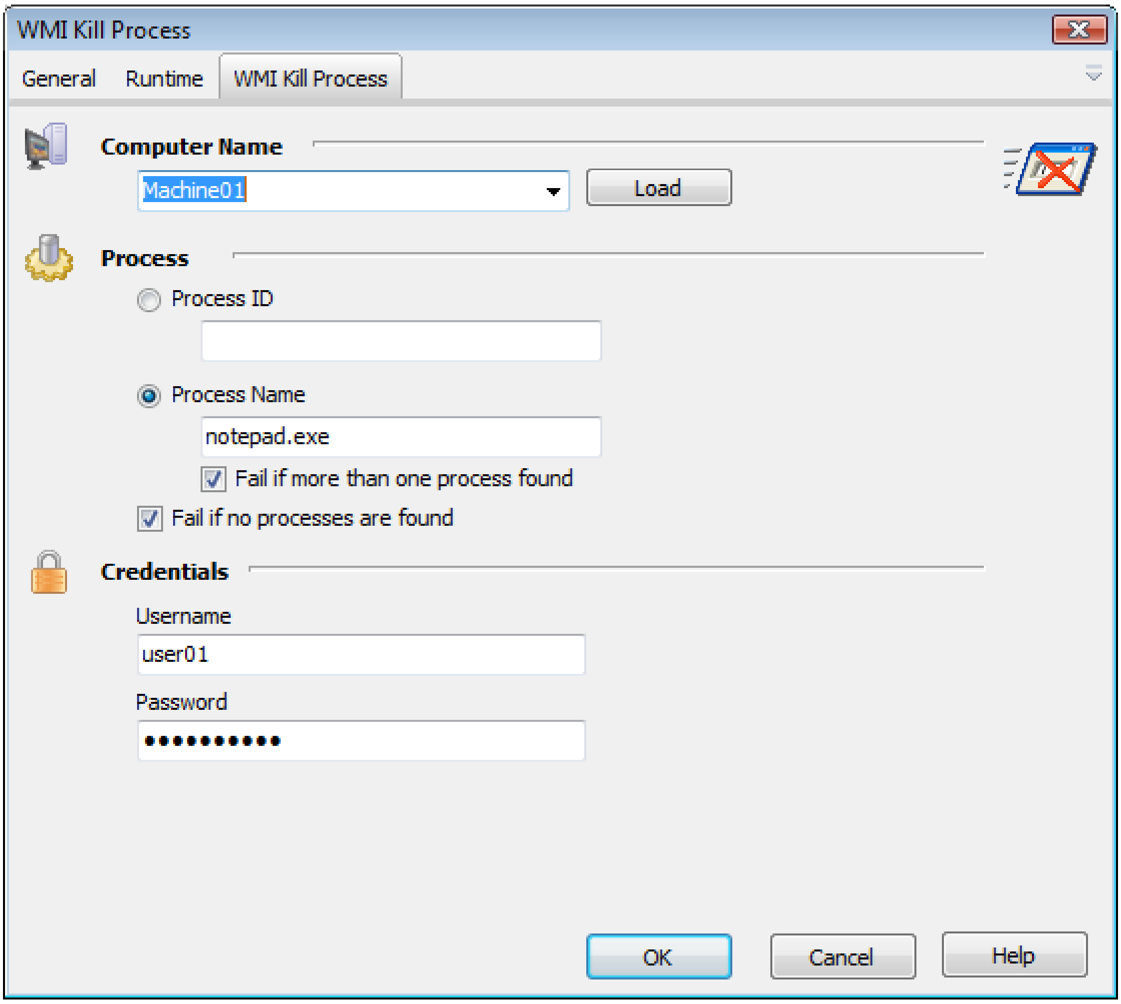
Remote Machine
The action will connect to a remote machine, using the specified user account.

Username
The username can be specified either by just the *username* or *DOMAIN/username*

Remote Host name
The computer which to connect to, if the remote computer is in a different domain then that of the current user account, a Fully Qualified Domain Name must be specified. An IPv4/IPv6 may also be specified.

6.27.32.4 WMI Kill Process Action

This action use the WMI (Windows Management Interface) API to end a process running on a remote or local machine. You can specify a processID or the process name.



Computer Name

Machine01

Process

○ Process ID

● Process Name

notepad.exe

☑ Fail if more than one process found

☑ Fail if no processes are found

Credentials

Username

user01

Password

•••••••••••

Process

You can specify the remote process by Process ID or Process Name. In the case of process name, you can set the action to fail if more than one process is found. Otherwise, all processes by that name will be deleted.

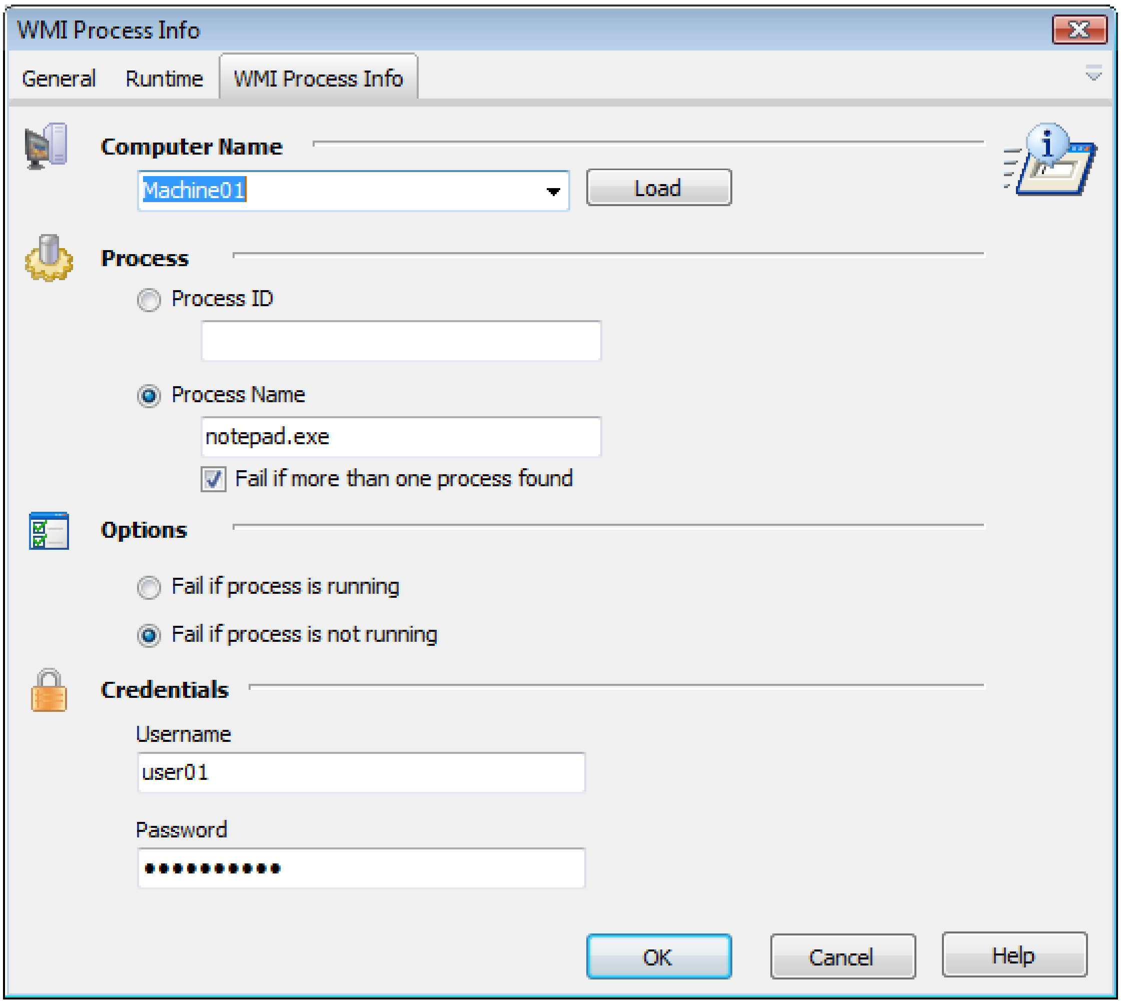
You can also set the action to fail if no processes are found.

Credentials

Specify credentials for the WMI connection. If no credentials are specified, WMI uses the currently logged in user's Windows Authentication credentials.

6.27.32.5 WMI Process Info Action

This action allows you to interrogate a process for information, or just check if the process is running or not.



Computer Name

Machine01

Process

○ Process ID

● Process Name

notepad.exe

☑ Fail if more than one process found

Options

○ Fail if process is running

● Fail if process is not running

Credentials

Username

user01

Password

••••••••••

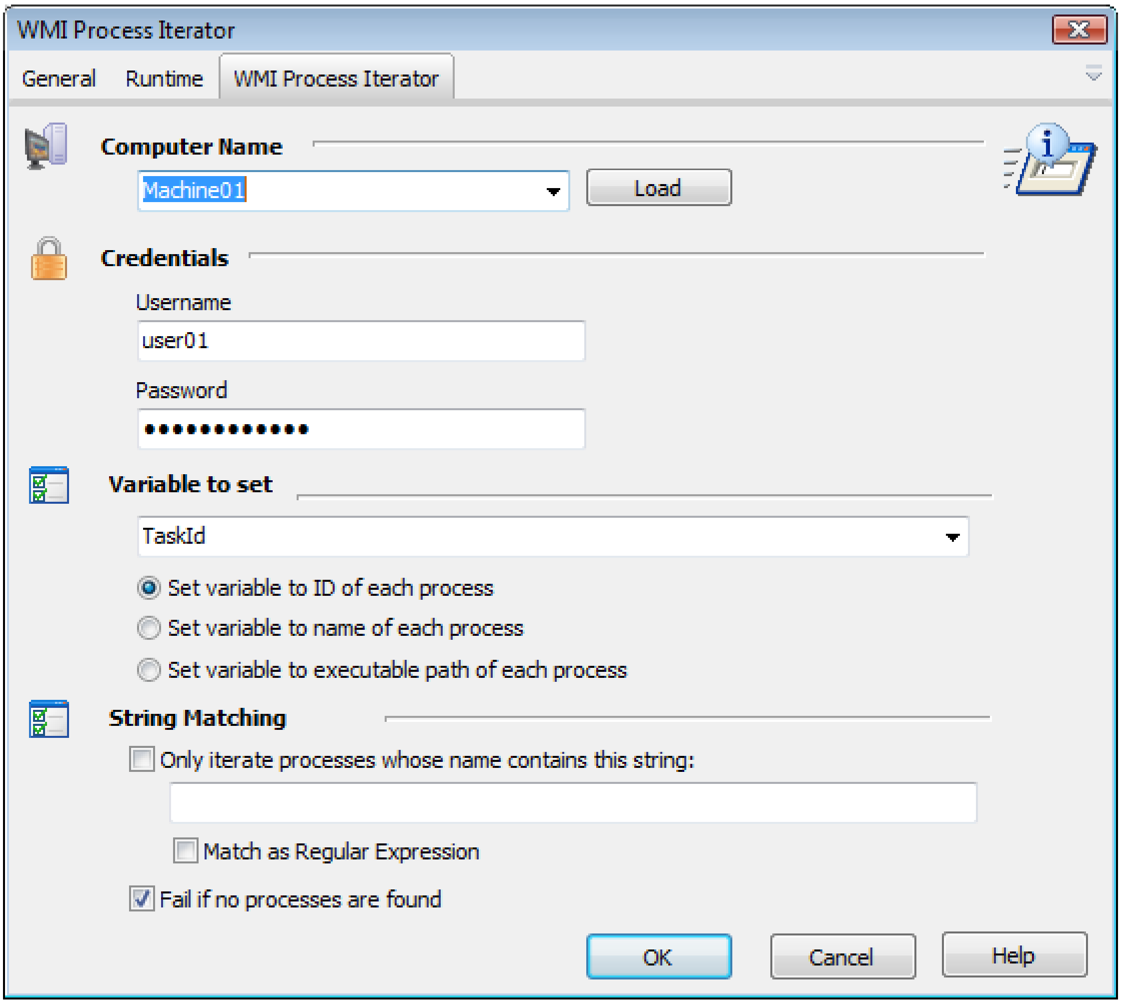
To get access to the process information you need to use script in the AfterAction script event. The following properties can be read from the action in the AfterAction event :

```
property ComputerName : string;
property ProcessID    : string;
property ProcessName  : string;

property ProcessCaption                 : string;
property ProcessDescription             : string;
property ProcessHandleCount             : LongWord;
property ProcessExecutablePath          : string;
property ProcessThreadCount             : LongWord;
property ProcessPriority                : LongWord;
property ProcessWorkingSetSize          : LongWord;
property ProcessPeakWorkingSetSize      : LongWord;

property ProcessPageFaults              : LongWord;
property ProcessPageFileUsage           : LongWord;
property ProcessParentProcessId         : LongWord;
property ProcessPeakPageFileUsage       : LongWord;
property ProcessPeakVirtualSize         : LongWord;
property ProcessCreationDate            : TDateTime;
property ProcessKernelModeTime          : LongWord;
property ProcessMaxWorkingSetSize       : LongWord;
property ProcessMinWorkingSetSize       : LongWord;
property ProcessOtherOperationCount     : LongWord;
property ProcessOtherTransferCount      : LongWord;
property ProcessPrivatePageCount         : LongWord;
property ProcessQuotaNonPagePoolUsage     : LongWord;
property ProcessQuotaPagePoolUsage       : LongWord;
property ProcessQuotaPeakNonPagePoolUsage : LongWord;
property ProcessQuotaPeakPagePoolUsage    : LongWord;
property ProcessReadOperationCount       : LongWord;
property ProcessReadTransferCount        : LongWord;
property ProcessSessionId                : LongWord;
property ProcessUserModeTime             : LongWord;
property ProcessWriteOperationCount      : LongWord;
property ProcessWriteTransferCount       : LongWord;
```

6.27.32.6 WMI Process Iterator

The WMI process iterator allows you to iterate through a list of processes running on any Windows machine. Each process can be identified by it's process ID or name. The list of processes can be filtered to match an executable name or regular expression.

For general information about iterators, click here.



WMI Process Iterator

Computer Name

Machine01

Credentials

Username

user01

Password

●●●●●●●●●●●●

Variable to set

TaskId

● Set variable to ID of each process

○ Set variable to name of each process

○ Set variable to executable path of each process

String Matching

□ Only iterate processes whose name contains this string:

□ Match as Regular Expression

☑ Fail if no processes are found

Computer Name

Enter '.' to browse the local host, otherwise enter the name of a computer on the local network. Press the Load button to load a list of computers into the drop down menu.

Credentials

Credentials are only used if browsing remote computers and are not needed if you have access via a Windows NT Domain.

Variable To Set

Specify a Automise variable to set on each iteration. To add a new variable, go to Tools -> Edit Variables.

The variable can be set to either the Process ID of each process, the Name of each process or the Executable Path of each process.

Note: When setting via Executable Path, system processes (which have no executable path) will not be iterated.

String Matching

The iterator can be set to iterate only processes whose names match a certain substring. The substring can be a full executable name, or a part of one. String matching is case insensitive.

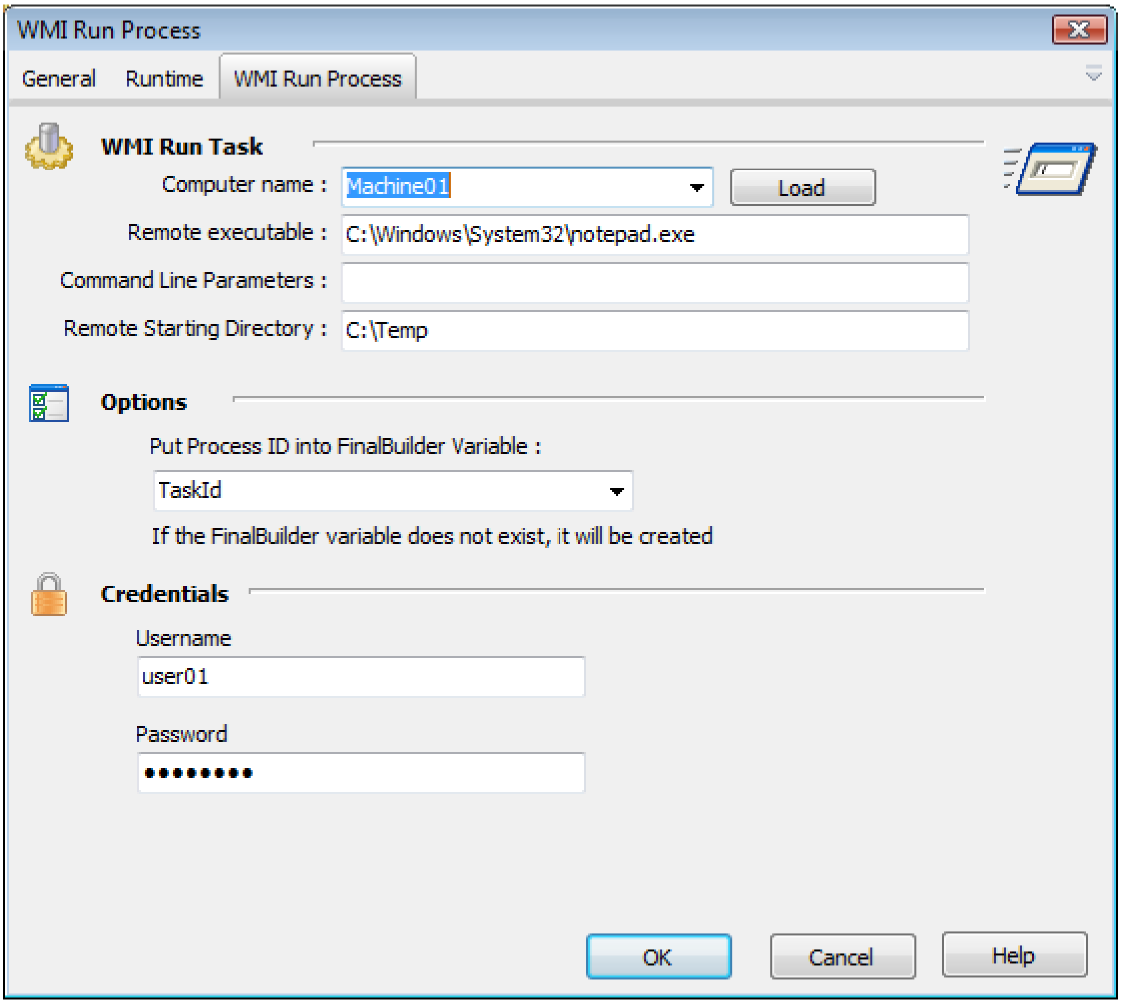
Alternatively, the string can be matched as a Regular Expression.

Fail if no processes are found

If this option is selected, the action will fail outright if no processes are found which match the chosen criteria.

6.27.32.7 WMI Run Process Action

This action use the WMI (Windows Management Interface) API to execute a process on a remote or local machine. Note that the path to the executable should be the path on the specified machine, and the executable must exist on that machine. WMI is supported on NT4 (you need to download it from Microsoft), Windows 2000 and XP.



WMI Run Process

WMI Run Task

Computer name : Machine01

Remote executable : C:\Windows\System32\notepad.exe

Command Line Parameters :

Remote Starting Directory : C:\Temp

Options

Put Process ID into FinalBuilder Variable :

TaskId

If the FinalBuilder variable does not exist, it will be created

Credentials

Username

user01

Password

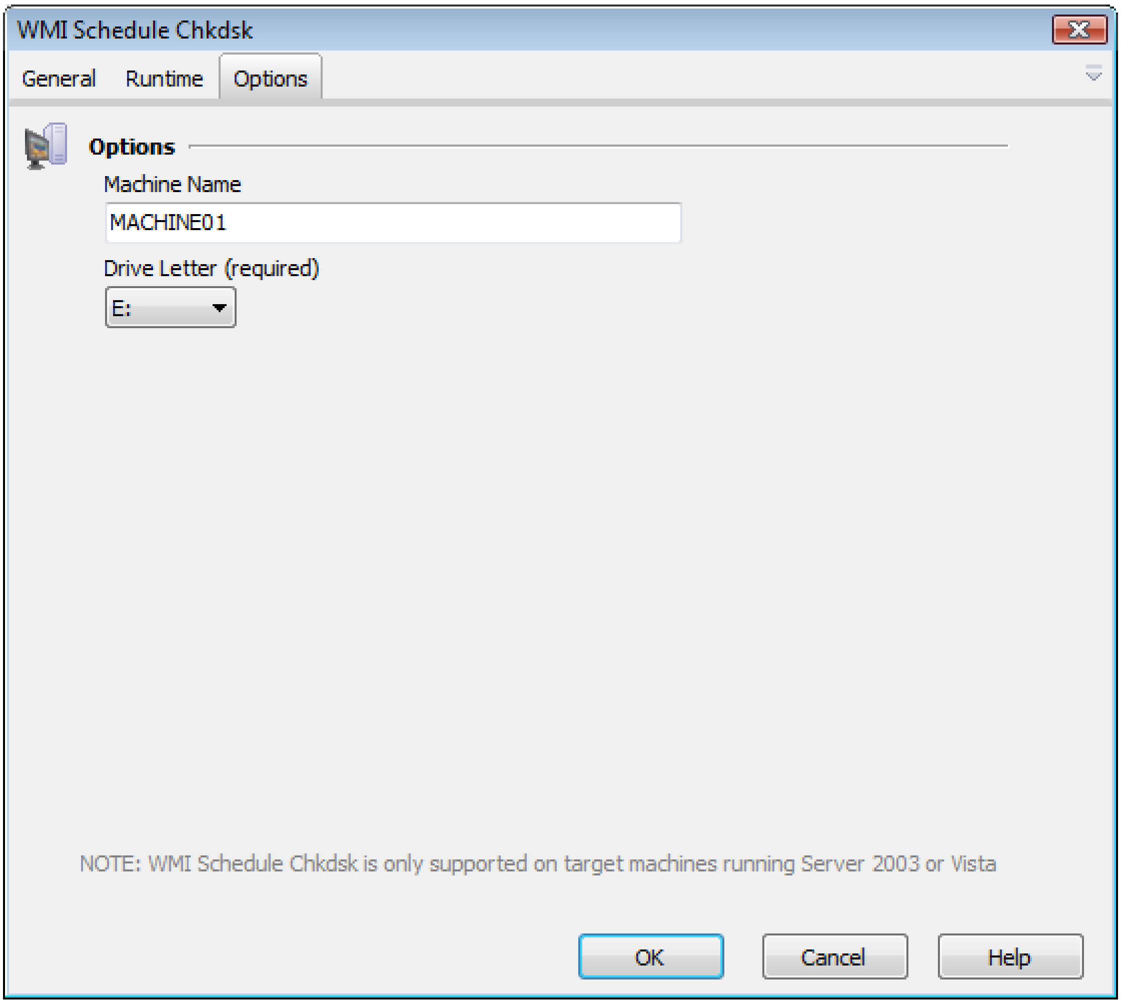
••••••••

You can choose to save the process ID in a Automise variable so that it may be used in later actions such as the WMI Kill Process Action.

6.27.32.8 WMI Schedule Chkdsk

The WMI Schedule Chkdsk action schedules Autochk to be run on the disk drive at the next reboot if the dirty bit is set. The method is applicable to only those instances of logical disk that represent a physical disk in the machine. The method is not applicable to mapped logical drives.

Client - Requires Windows Vista or Windows XP.
Server - Requires Windows Server "Longhorn" or Windows Server 2003.



Machine Name: MACHINE01

Drive Letter (required): E:

NOTE: WMI Schedule Chkdsk is only supported on target machines running Server 2003 or Vista

Machine Name - Enter the name of the remote machine, or leave blank for localhost

Drive Letter - Select the drive letter. If you want to set this dynamically, set it in the BeforeAction script event. See below.

Scripting Notes.

You can set the various properties of the action via script. For example, to set the Drive Letter via script:
BeforeAction Script Event:
Action.PropertyStore.DriveLetter = TheDriveLetter

Where "TheDriveLetter" is a variable that has been set to a valid drive on the machine, in the format <DriveLetter>:
eg. E: or C:

6.28 XML Actions

XML Namespaces

A common issue that occurs when using the Automise XML actions is the following error:

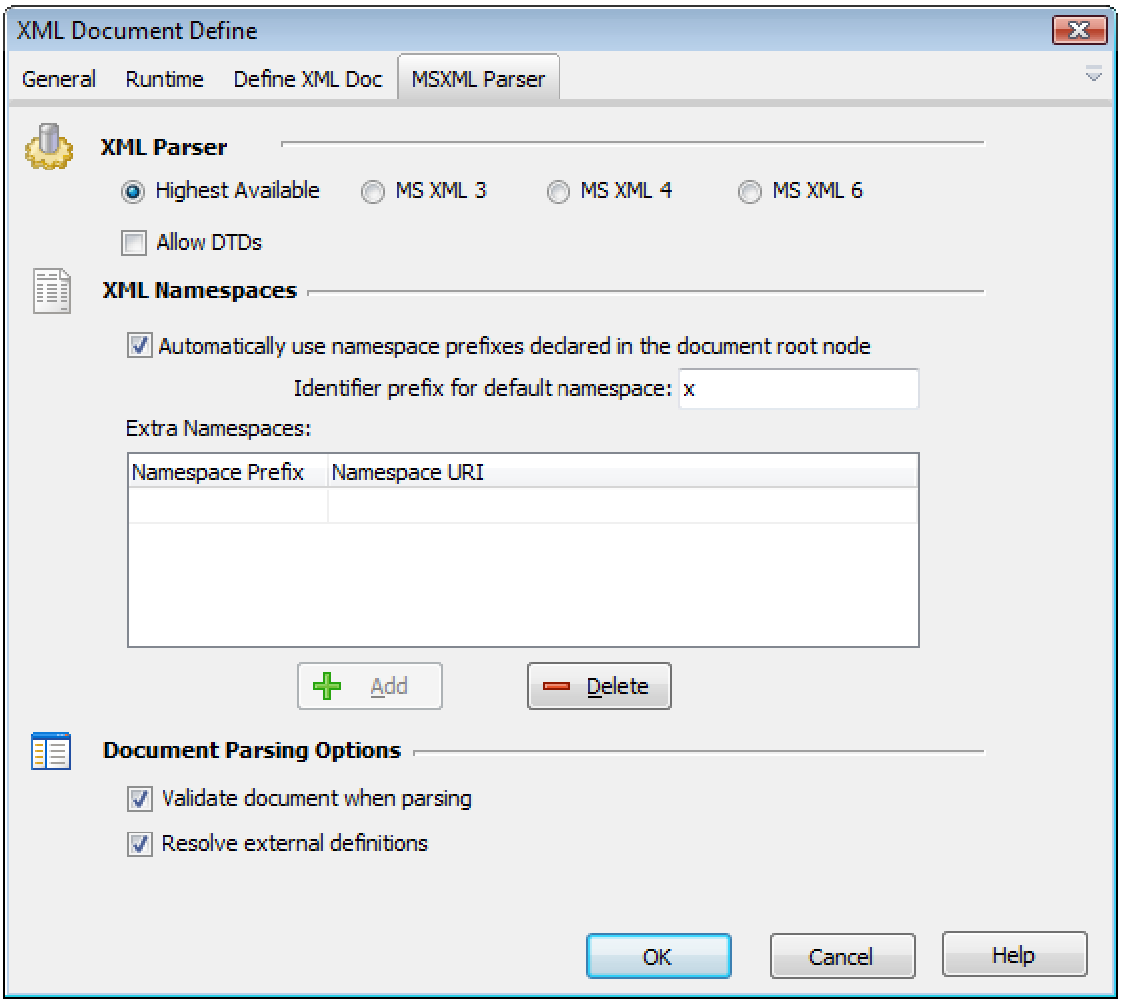
```
XPath returned no node : /Project/Import
```

When using an XML document the uses namespaces, this error can occur even when the XPath to the node is correct. For example if you wanted to iterate over all the imported project names in a .csproj and you defined the XPath to iterate over as */Project/Import* then you would see the error above. If you view .csproj file in a text editor and look at the first line then you will see something resembling the following:

```
<Project ToolsVersion="4.0" DefaultTargets="Build" xmlns="http://schemas.
microsoft.com/developer/msbuild/2003">
```

The xmlns attribute that indicates that this document has a default namespace. This is why MSXML will not be able to find the */Project/Import* nodes given the */Project/Import* XPath.

To allow MSXML to find these nodes, go to the MSXML parser page of the action (or the XML Document Define action that you are using) and enable the *Automatically use the namespace prefixes declared in the document root node* option. You need to provide a letter to prefix the default namespace, to do this enter a letter in the *Identifier prefix for the default namespace* field.



● Highest Available ○ MS XML 3 ○ MS XML 4 ○ MS XML 6

□ Allow DTDs

☑ Automatically use namespace prefixes declared in the document root node

Identifier prefix for default namespace: x

Extra Namespaces:

Namespace Prefix	Namespace URI

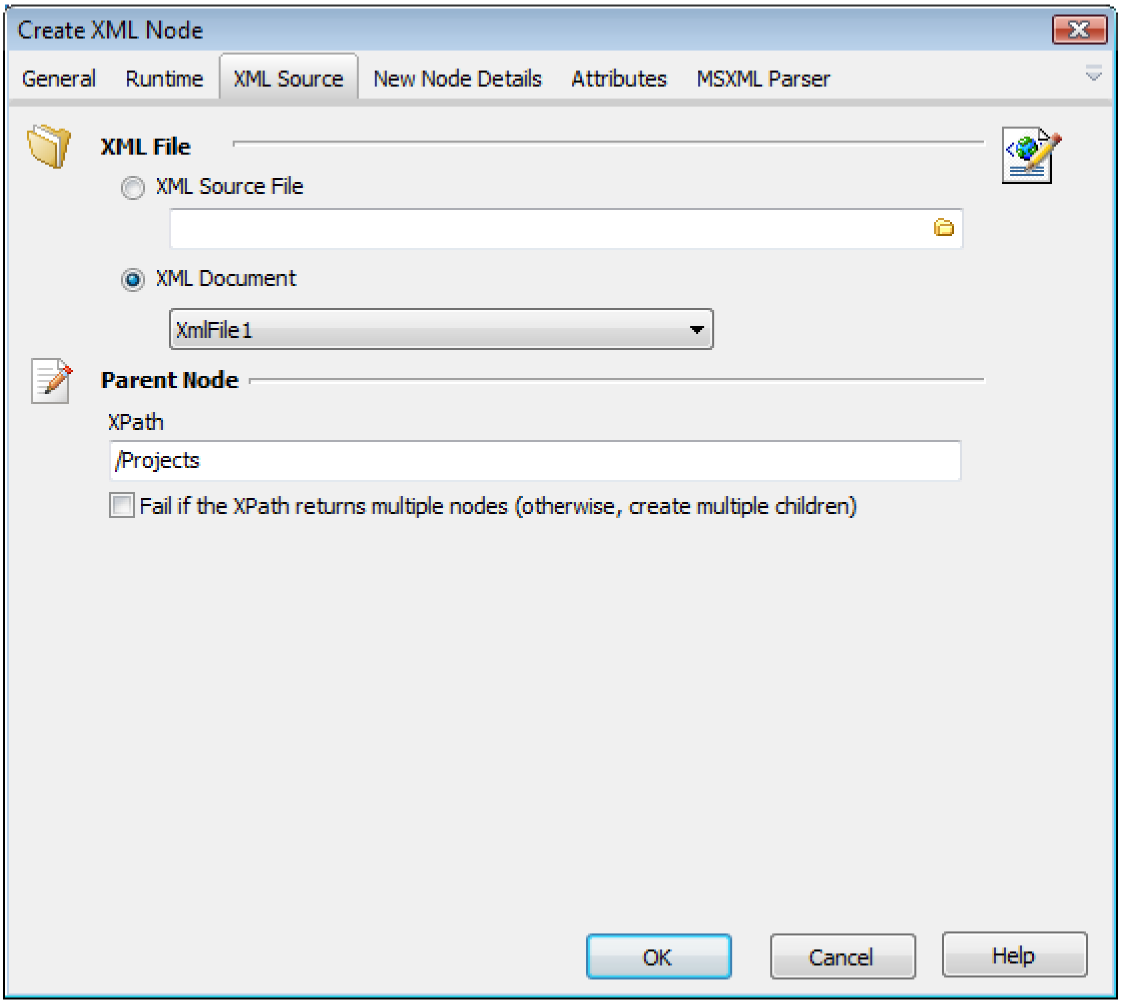
☑ Validate document when parsing

☑ Resolve external definitions

In the screenshot above, *x* has been used as the prefix for the default namespace. To reference the */Project/Import* nodes the XPath used is *x:/Project/x:Import as* x is referencing the default namespace of the document.

6.28.1 Create XML Node

The Create XML Node action allows you to insert an extra node into an existing XML file.



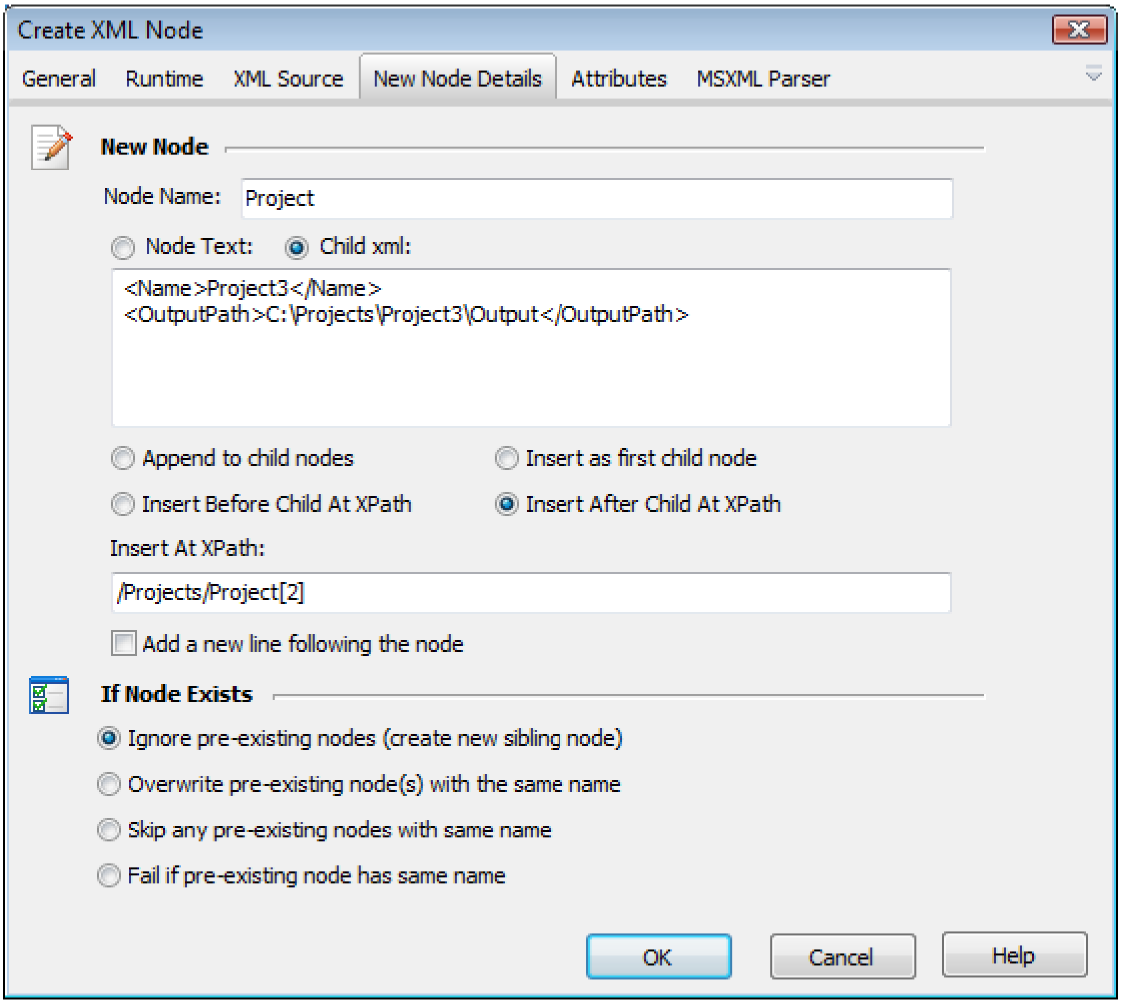
XML File

Specify an XML file or document to modify.

Parent Node

Enter the XPath to the parent node under which to create the new node.

Check the "Fail if the XPath returns multiple nodes" option if you only want to create one node. Otherwise, if the XPath returns multiple nodes than an identical child node will be created under each parent.



New Node

Node Name: Project

○ Node Text: ● Child xml:

```
<Name>Project3</Name>
<OutputPath>C:\Projects\Project3\Output</OutputPath>
```

○ Append to child nodes ○ Insert as first child node

○ Insert Before Child At XPath ● Insert After Child At XPath

Insert At XPath: /Projects/Project[2]

□ Add a new line following the node

If Node Exists

● Ignore pre-existing nodes (create new sibling node)

○ Overwrite pre-existing node(s) with the same name

○ Skip any pre-existing nodes with same name

○ Fail if pre-existing node has same name

Node Name: The name of the new node.

Node Text: The text content of the new node. Leave blank if you do not wish to add any text content.

Append to child nodes
Choose "Append to child nodes" to have the new node created last - after any sibling nodes.

Insert as first child node
Choose "Insert as first child node" to have the new node created before any siblings.

Insert Before Child At XPath
Specify the the Xpath of a sibling node to insert this node directly before.

Insert After Child At XPath
Specify the the Xpath of a sibling node to insert this node directly after.

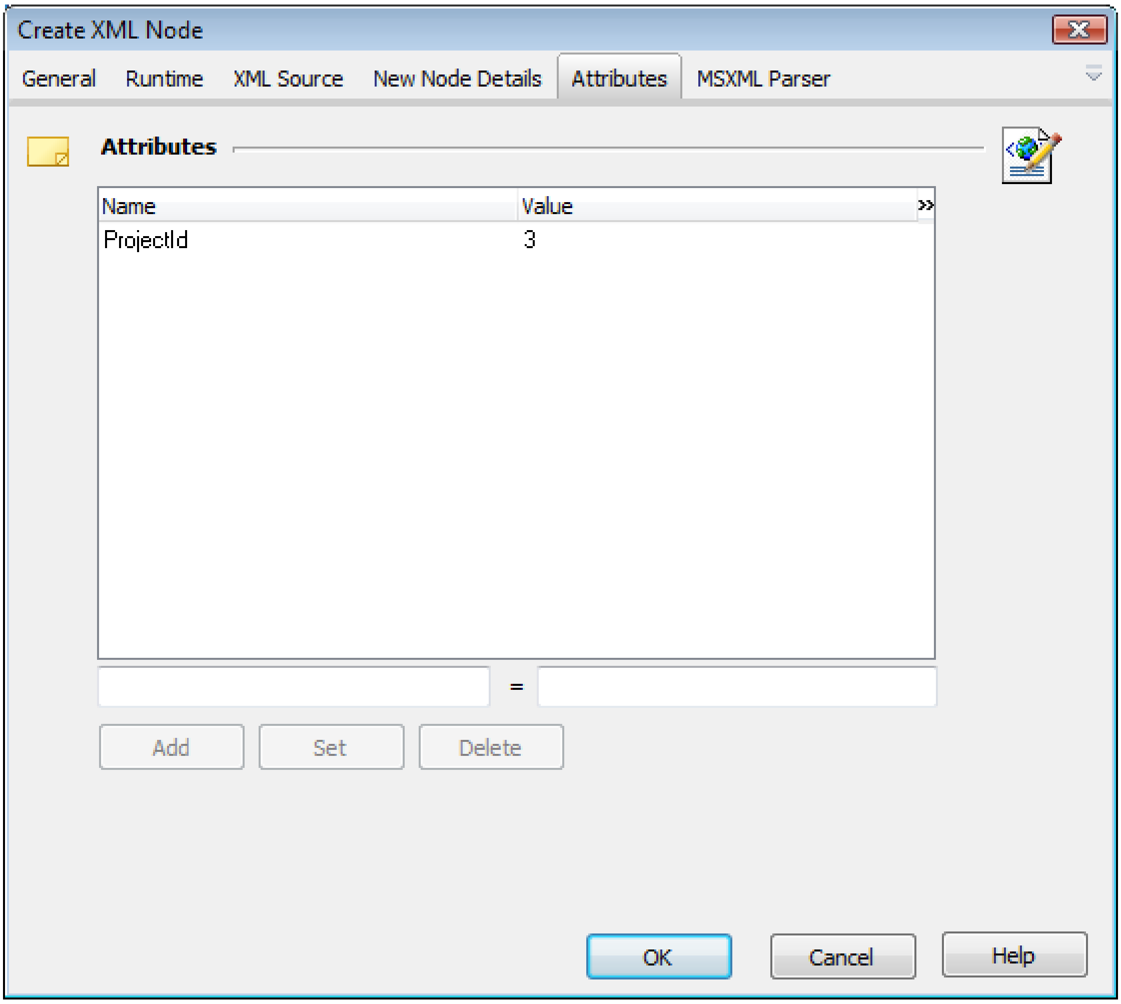
If Node Exists:

"Ignore pre-existing nodes"
The new node will be created regardless of whether other nodes with the same name exist.

"Overwrite pre-existing node(s) with the same name"
The action will delete any child nodes it finds with the same name as the new node. Note that if multiple child nodes with the same name exist, all of them will be erased but only one will be created.

"Skip any pre-existing nodes with same name"
If any child nodes exist with the same name as the new node, that child node will be skipped.

"Fail if any pre-existing node has same name"
If any child nodes exist with the same name, the action will fail.



Name	Value
ProjectId	3

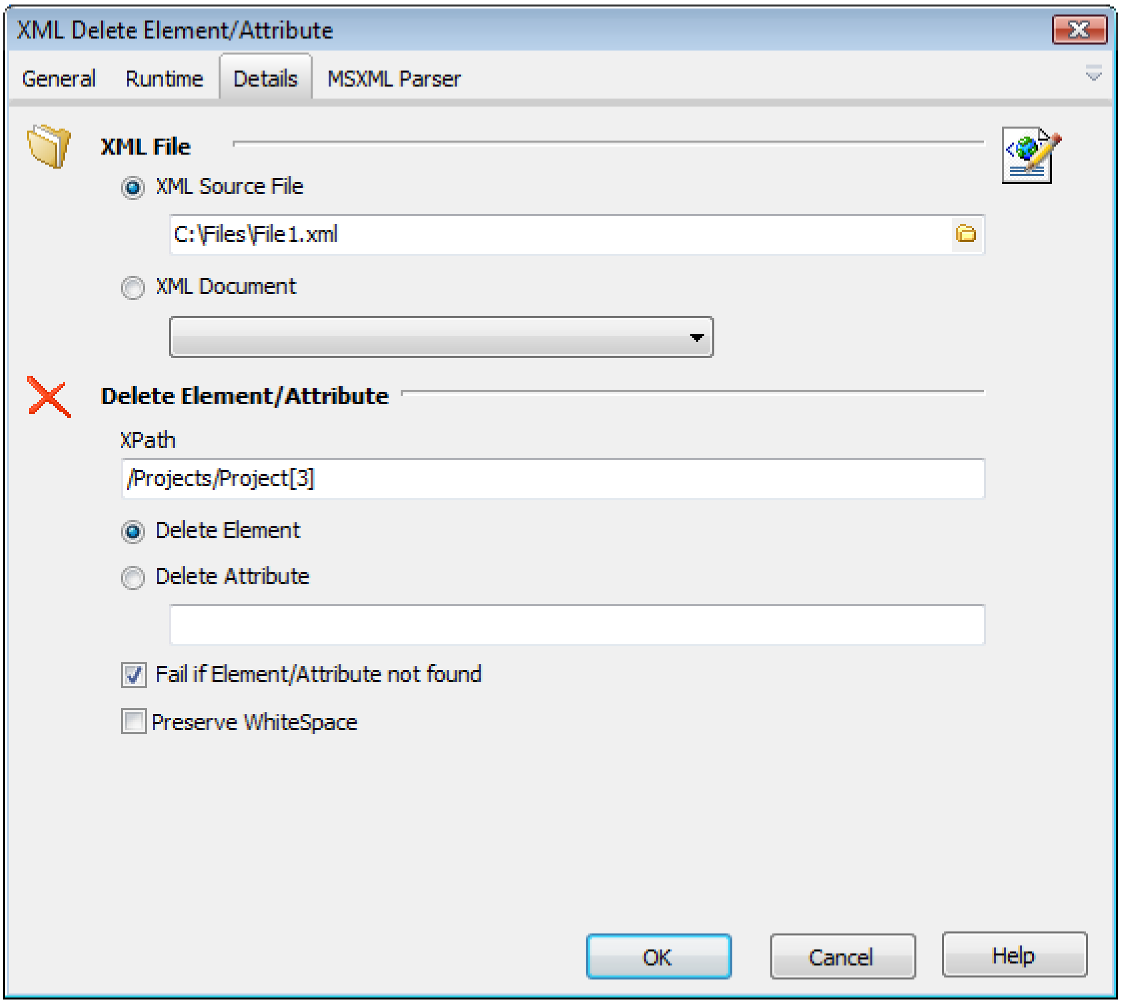
The attributes list shows attribute and value names for the new node. To add a new attribute, type its name and value into the edit fields and then click the Add button.

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.2 Delete XML Element/Attribute

This action allows you to delete a single node or attribute in an XML document. If you want to delete a node list (ie. a list of nodes selected by an xpath query, then use the Delete XML Nodes action).



XML Delete Element/Attribute

XML File

● XML Source File

C:\Files\File1.xml

○ XML Document

Delete Element/Attribute

XPath

/Projects/Project[3]

● Delete Element

○ Delete Attribute

☑ Fail if Element/Attribute not found

□ Preserve WhiteSpace

XML File - specify a xml file, or an XML Document object.

XPath - the xpath statement which specifies the node

Delete Element - the first node selected by the xpath will be deleted if this option is chosen

Delete Attribute - the attribute of the node selected by the xpath will be deleted if this option is chosen

Fail if Element/Attribute not found - the action will fail if the specified node or attribute is not found

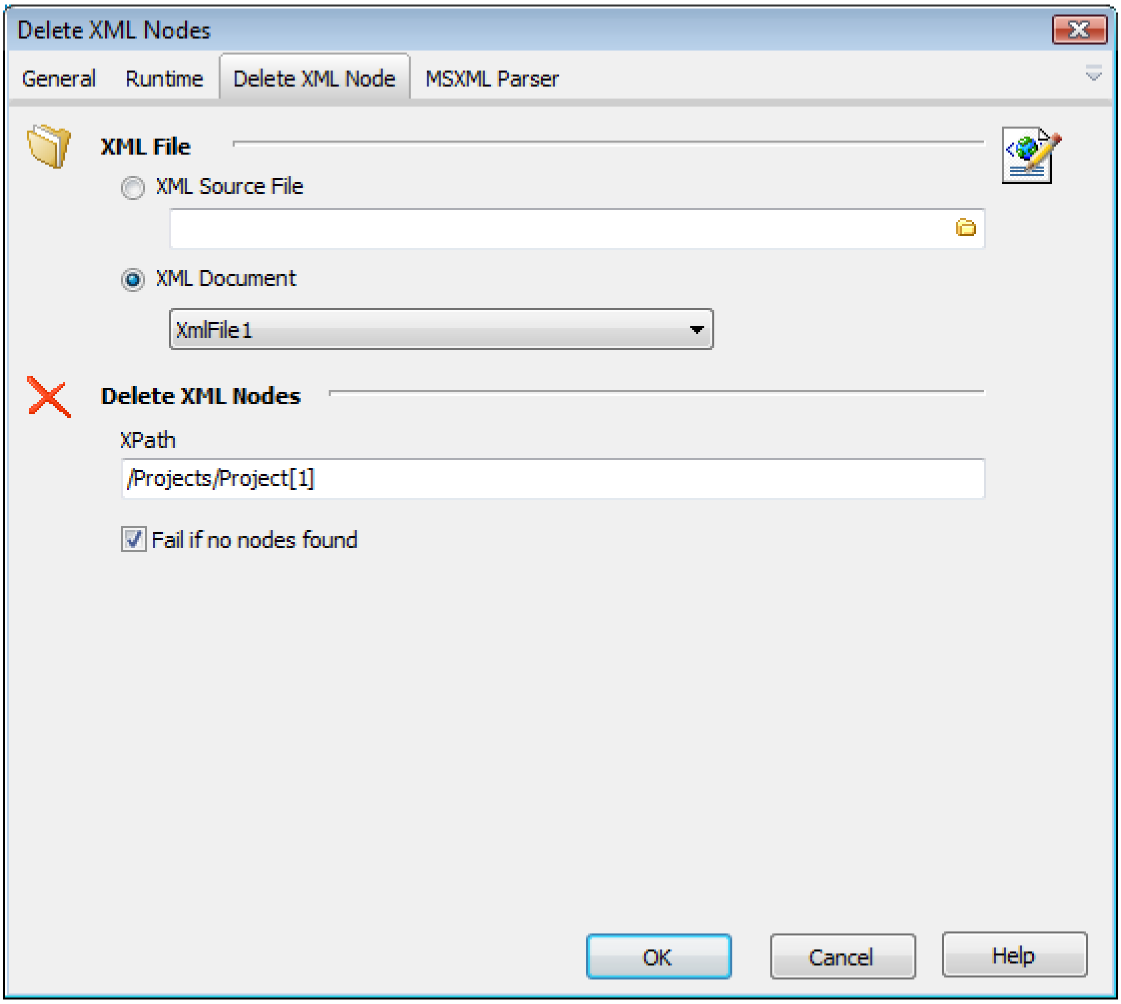
Preserve WhiteSpace - whitespace in the specified xml source file is preserved when the file is written to disk

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.3 Delete XML Nodes

This action allows you to delete a node or set of nodes in an XML document.


Delete XML Nodes

General | Runtime | Delete XML Node | MSXML Parser

XML File

○ XML Source File

● XML Document

XmlFile1

Delete XML Nodes

XPath

/Projects/Project[1]

☑ Fail if no nodes found

XML File

Specify an XML file or document to to remove nodes from.

Delete XML Nodes

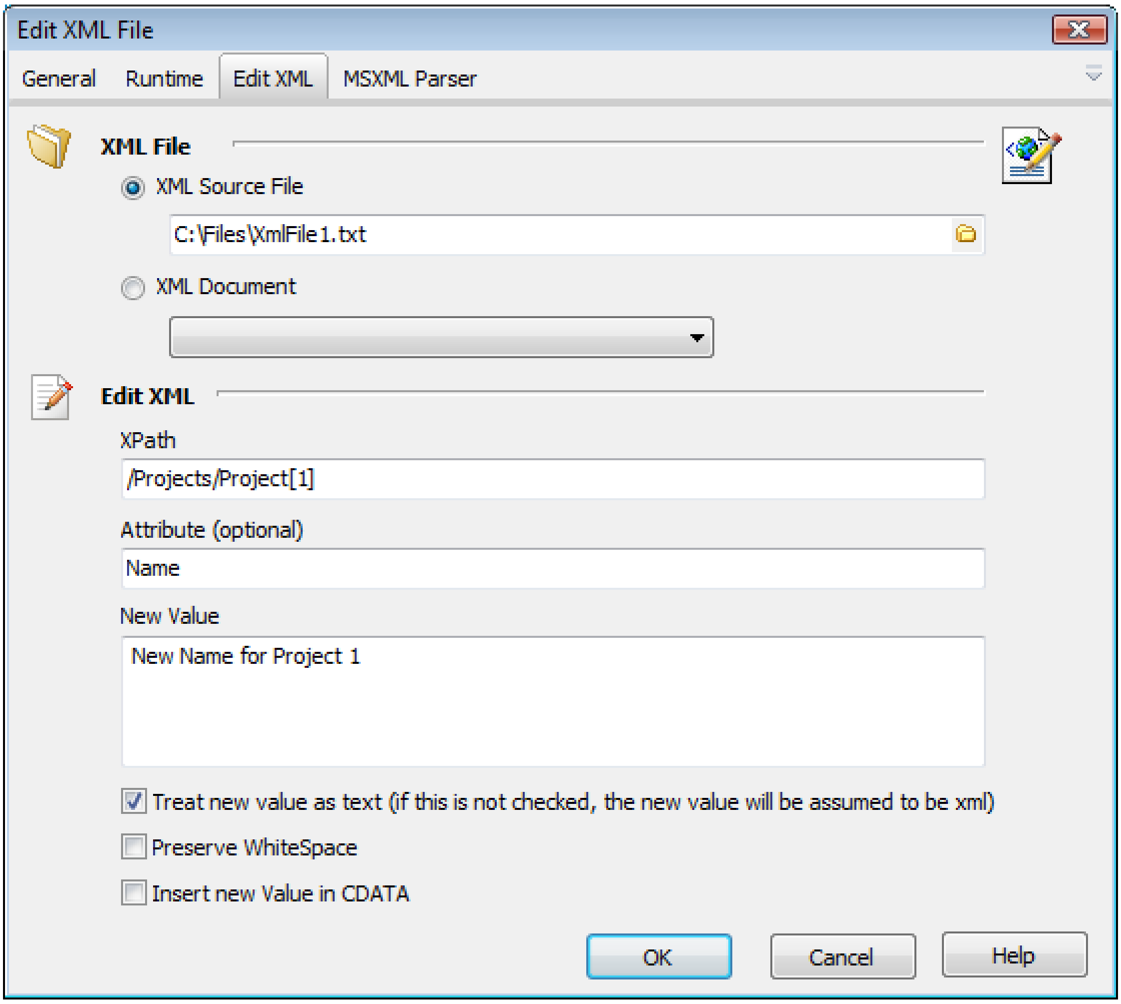
Specify the XPath to the node or nodes you wish to delete. Check the 'Fail if no nodes found' checkbox if you want the action to fail if no nodes match the XPath.

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.4 Edit XML File Action

This action allows you to modify a value in an XML document. The value to modify is selected using XPath, if the Attribute field is empty the node text will be set.



● XML Source File

C:\Files\XmlFile1.txt

○ XML Document

XPath: /Projects/Project[1]

Attribute (optional): Name

New Value: New Name for Project 1

☑ Treat new value as text (if this is not checked, the new value will be assumed to be xml)

□ Preserve WhiteSpace

□ Insert new Value in CDATA

XML File

Specify an XML file or document to edit.

XPath

Specify the XPath of the node you wish to edit.

Attribute (optional)

If you specify an attribute here then that attribute's value will be set to 'New Value.'

If you do not specify an attribute, then the entire contents of the selected node will be set instead.

New Value

Specify the value to write to the XML file.

MSXML Parser Page

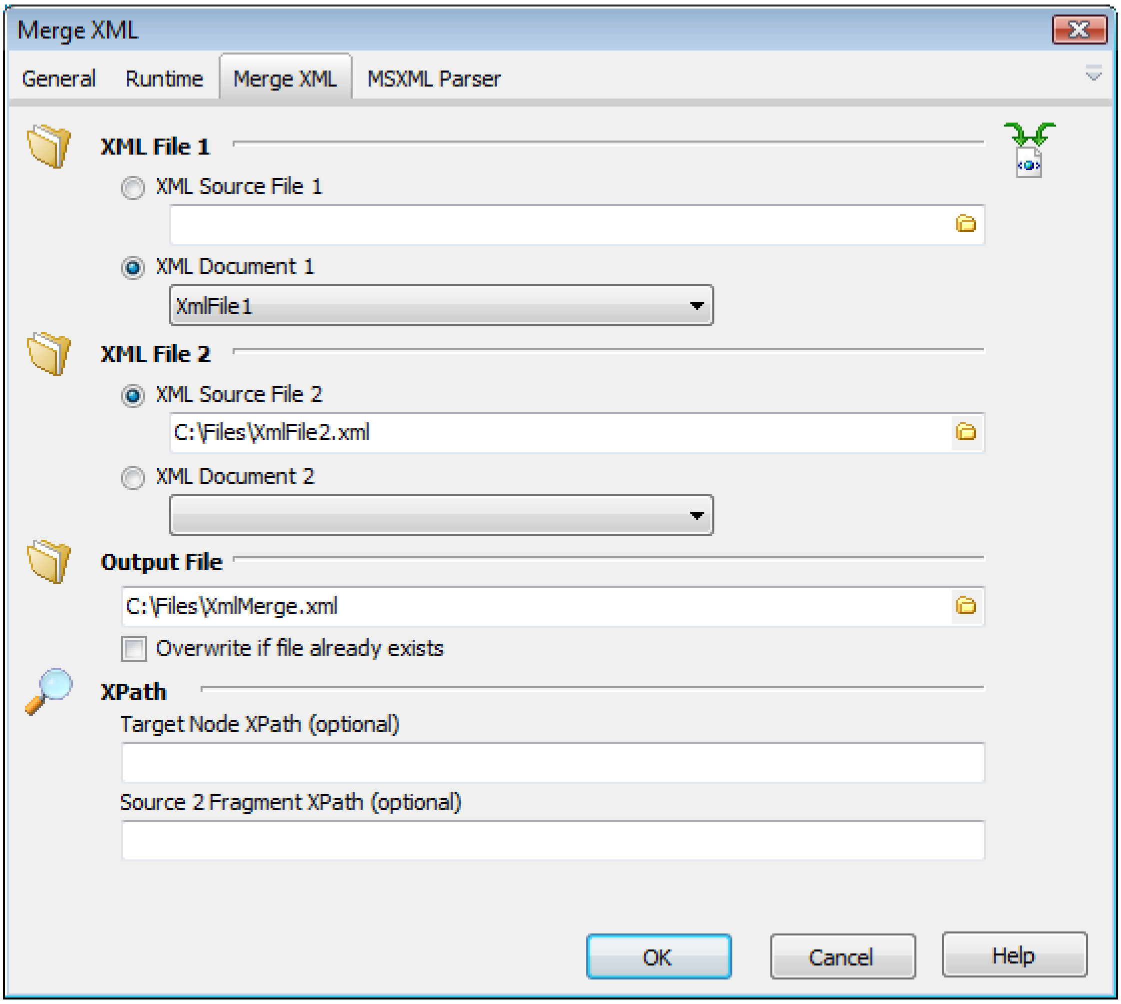
Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

Preserve Whitespace

This option specifies the default white space handling of the MSXML parser. When set to true, all white space is preserved, regardless of any xml:space attributes specified in the document type definition (DTD). It is equivalent to having an xml:space="preserve" attribute on every element. When Preserve Whitespace is False, the values of any xml: space attributes determine where white space is preserved.

6.28.5 Merge XML Action

This action merges two XML documents. Source 2 will be merged into Source 1 and saved as a new document.



Merge XML

XML File 1

○ XML Source File 1

● XML Document 1: XmlFile1

XML File 2

● XML Source File 2: C:\Files\XmlFile2.xml

○ XML Document 2

Output File: C:\Files\XmlMerge.xml

□ Overwrite if file already exists

XPath

Target Node XPath (optional)

Source 2 Fragment XPath (optional)

XML File 1, XML File 2

Specify the XML files or documents to merge from.

Output File

Specify the file to write to.

XPath

The TargetNode XPath option allows you to specify the node where the Source2 document will be inserted.

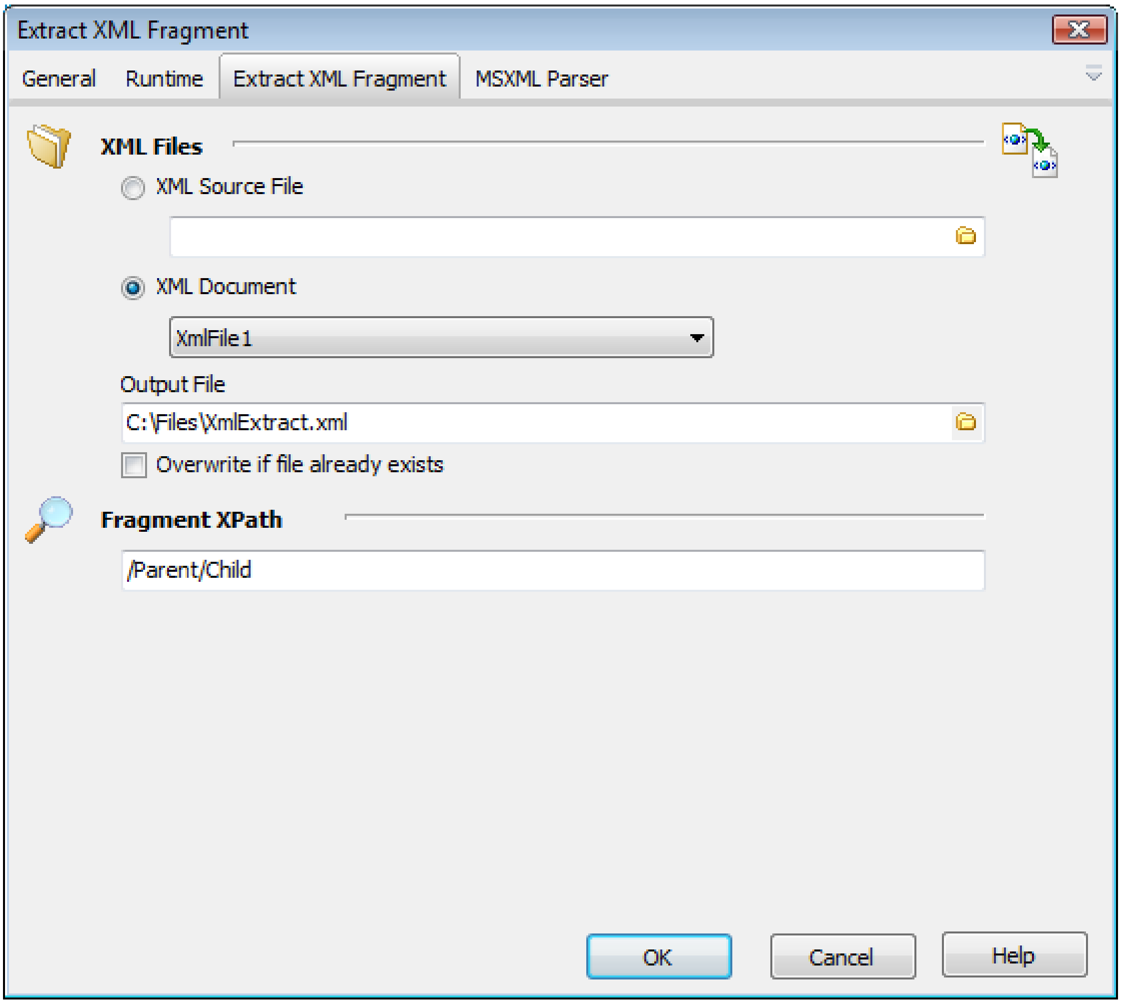
The Source 3 Fragment XPath option allows you to specify a document fragment to merge rather than the whole document.

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.5.1 Extract XML Fragment Action

This action extracts an xml document fragment from a source xml document and saves it as a new xml document.



XML Files

○ XML Source File

● XML Document

XmlFile1

Output File

C:\Files\XmlExtract.xml

□ Overwrite if file already exists

Fragment XPath

/Parent/Child

XML Files

Specify a source file (or document), and a destination file to write the fragment to.

Fragment XPath

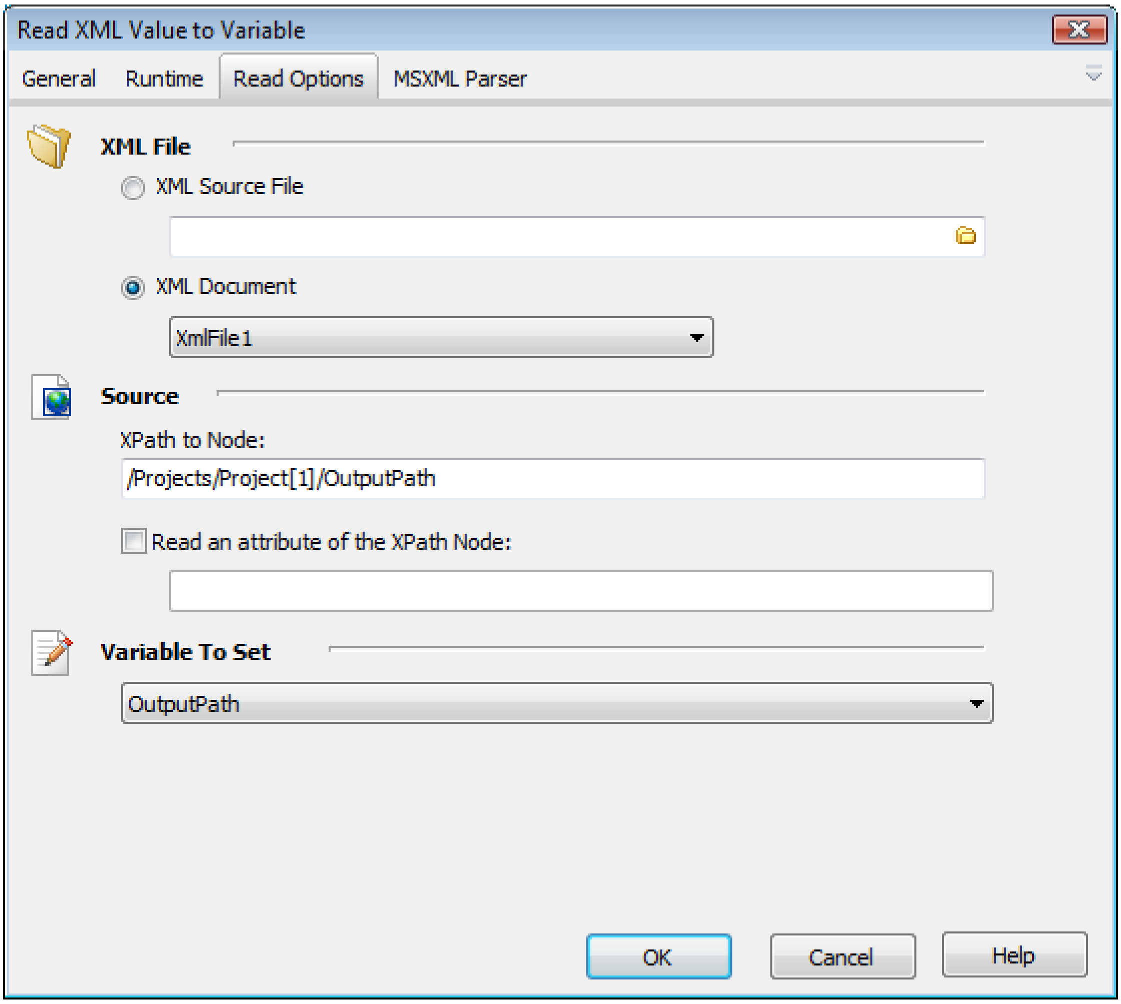
Specify the XPath to the nodes you wish to extract.

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.6 Read XML Value to Variable

Read XML Value to Variable allows you retrieve a text or attribute value from an XML file and store it in a Automise variable.



Read XML Value to Variable

XML File

○ XML Source File

● XML Document

XmlFile1

Source

XPath to Node:

/Projects/Project[1]/OutputPath

□ Read an attribute of the XPath Node:

Variable To Set

OutputPath

XML File

Specify an XML file to read from.

XPath to Node

Specify the XPath to the node you wish to read. If you wish to read from more than one node, try the XML Node Iterator action.

Read an attribute of the XPath node

If you check this box, Automise will read the value of a specified attribute instead of the text value of the node. Type the name of the attribute into the text field.

FB Variable To Set

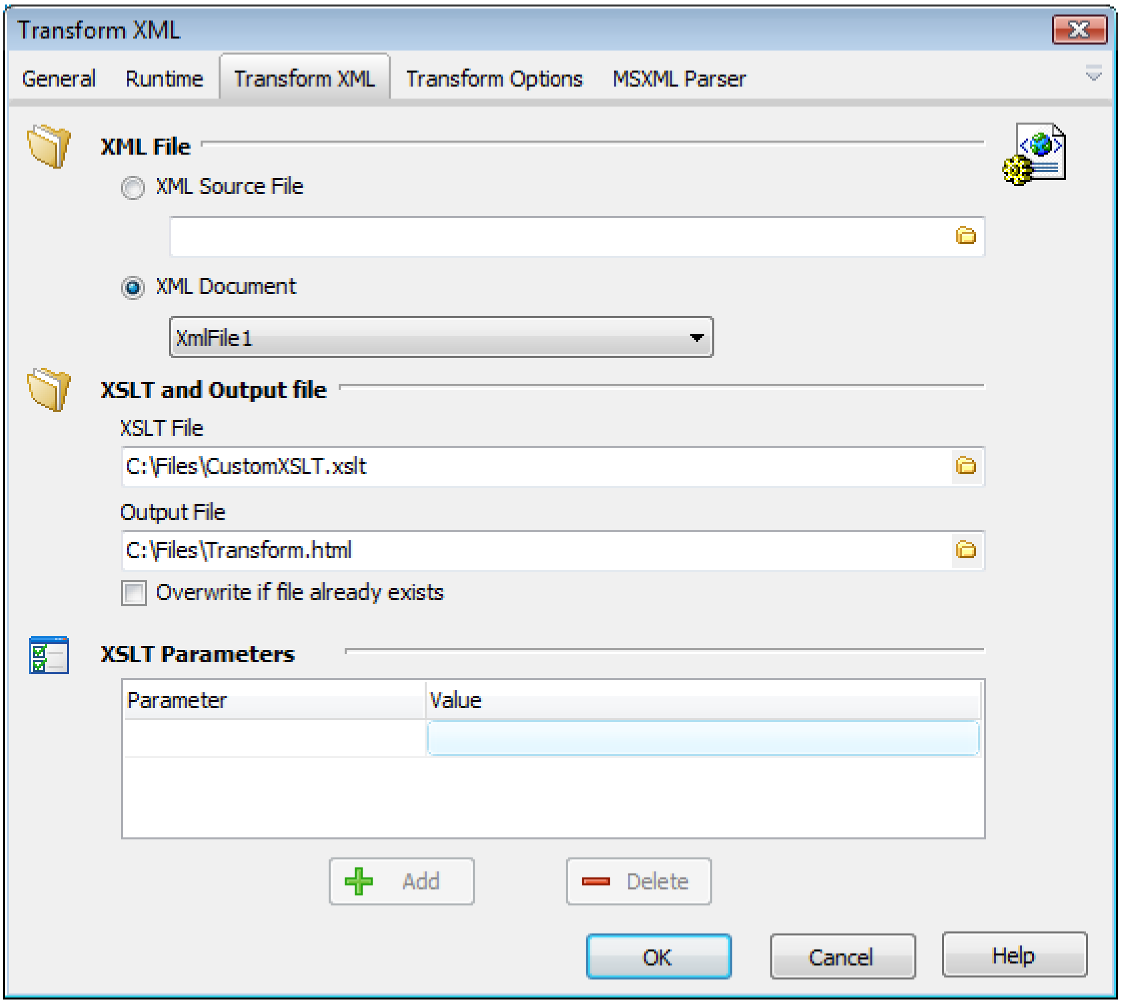
The value will be stored in the specified Automise variable.

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.7 Transform XML

This action performs an XSL Transform using the Microsoft XML parser.



Transform XML

XML File

○ XML Source File

● XML Document

XmlFile1

XSLT and Output file

XSLT File: C:\Files\CustomXSLT.xslt

Output File: C:\Files\Transform.html

□ Overwrite if file already exists

XSLT Parameters

Parameter	Value

XML Source File

Specify an XML file or document to read from.

XSLT and Output File

Specify the XSLT to use for the transform, and the output file to write to.

XSLT Parameters

The XSLT Parameters allows you to provide dynamic values (such as build numbers etc) to your stylesheet. These can be used to alter the output of the transform.

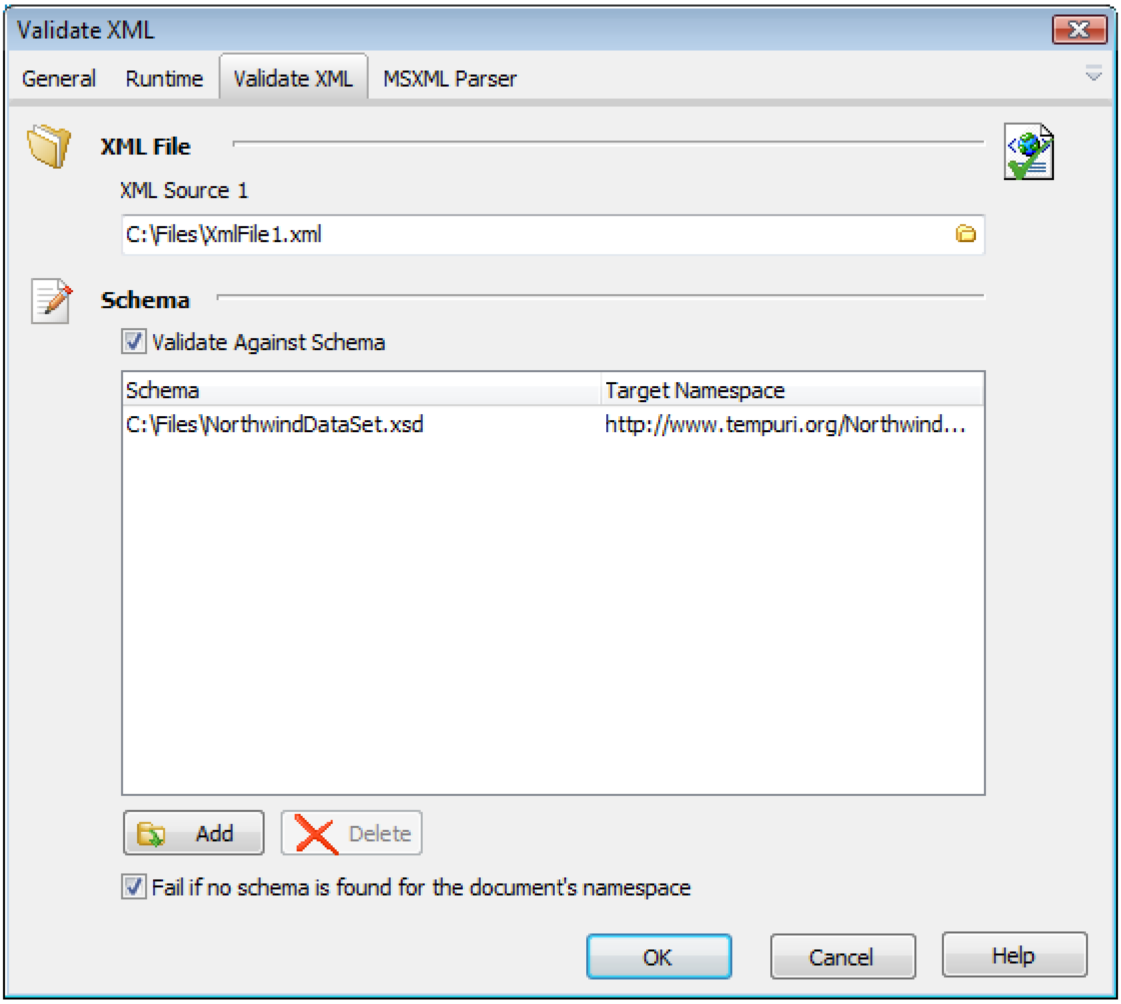
MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.8 Validate XML File

This action validates an XML file using the Microsoft XML parser.

You can validate against one or more XML schemas, or if no schemas are specified then the action parses the xml file and reports any errors found.



XML Source 1: C:\Files\XmlFile1.xml

☑ Validate Against Schema

Schema	Target Namespace
C:\Files\NorthwindDataSet.xsd	http://www.tempuri.org/Northwind...

☑ Fail if no schema is found for the document's namespace

Schema

Enable Validate Against Schema to specify one or more schema. Schema must be specified along with a Target Namespace (if the TargetNamespace attribute is present in the schema file, it will be automatically loaded when the Schema file is added to the list.)

Fail if no schema is found for the document's namespace

Each Schema has a target namespace which it validates against. Enable this option to have the action fail if no available namespaces match the namespace of the target XML file.

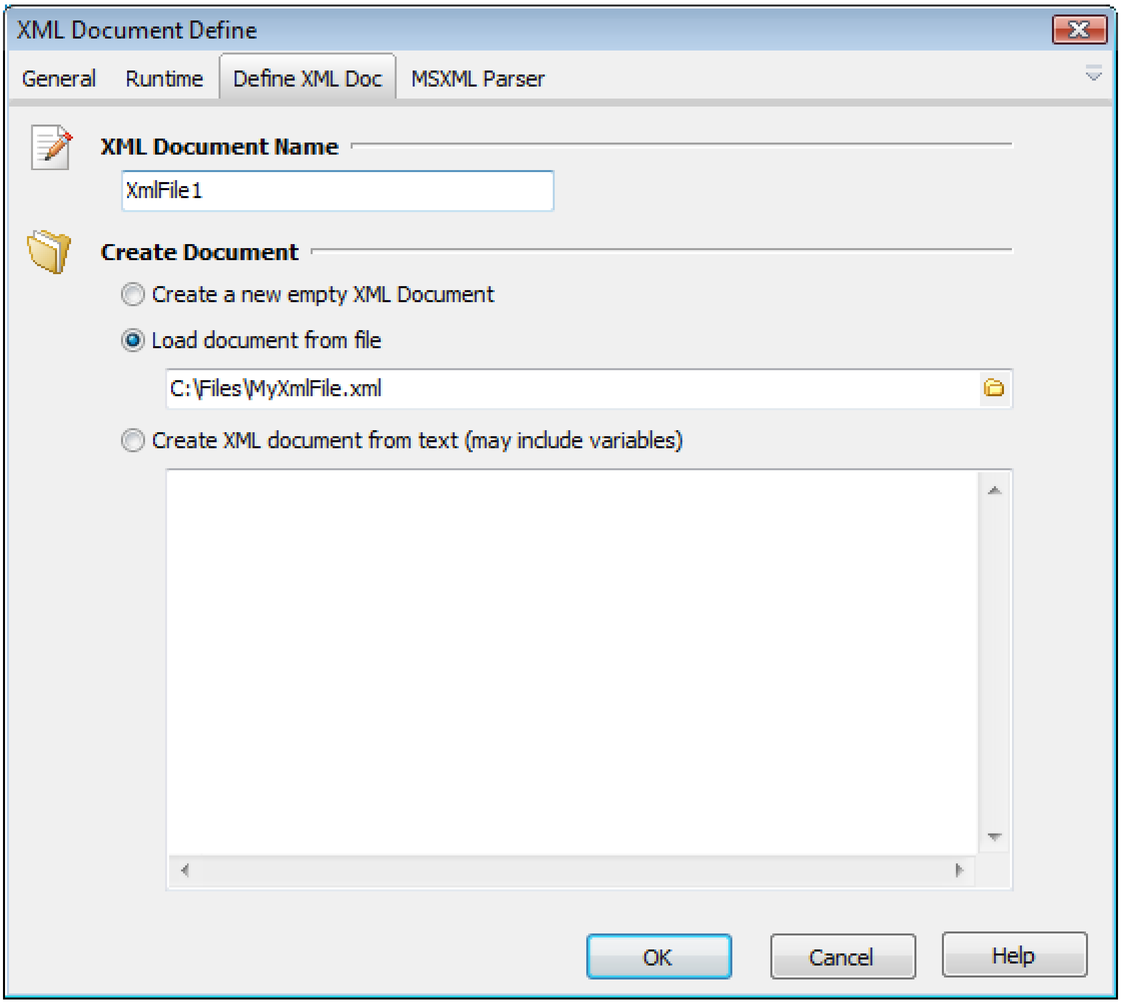
6.28.9 XML Document Define

The XML Document Define action is used to create an xml document object in memory which can then be used with other XML actions that use an XML object.

The actions that can use this XML Document Object are:

Transform XML
Merge XML
Extract XML Fragment
Edit XML File
Delete XML Nodes
Read XML Value to Variable
XML Node Iterator
Create XML Node
XML Delete Element/Attribute
XML Save Document

After performing an operation on an XML Object (eg. editing a value, deleting an attribute, etc) and you wish to save the XML Object as a file, you need to use the XML Save Document action.



XML Document Name

XmlFile1

Create Document

○ Create a new empty XML Document

● Load document from file

C:\Files\MyXmlFile.xml

○ Create XML document from text (may include variables)

XML Document Name - the name to refer to this internal xml document

Create Document - the document can be created by reading an xml file, an empty document, or by entering text

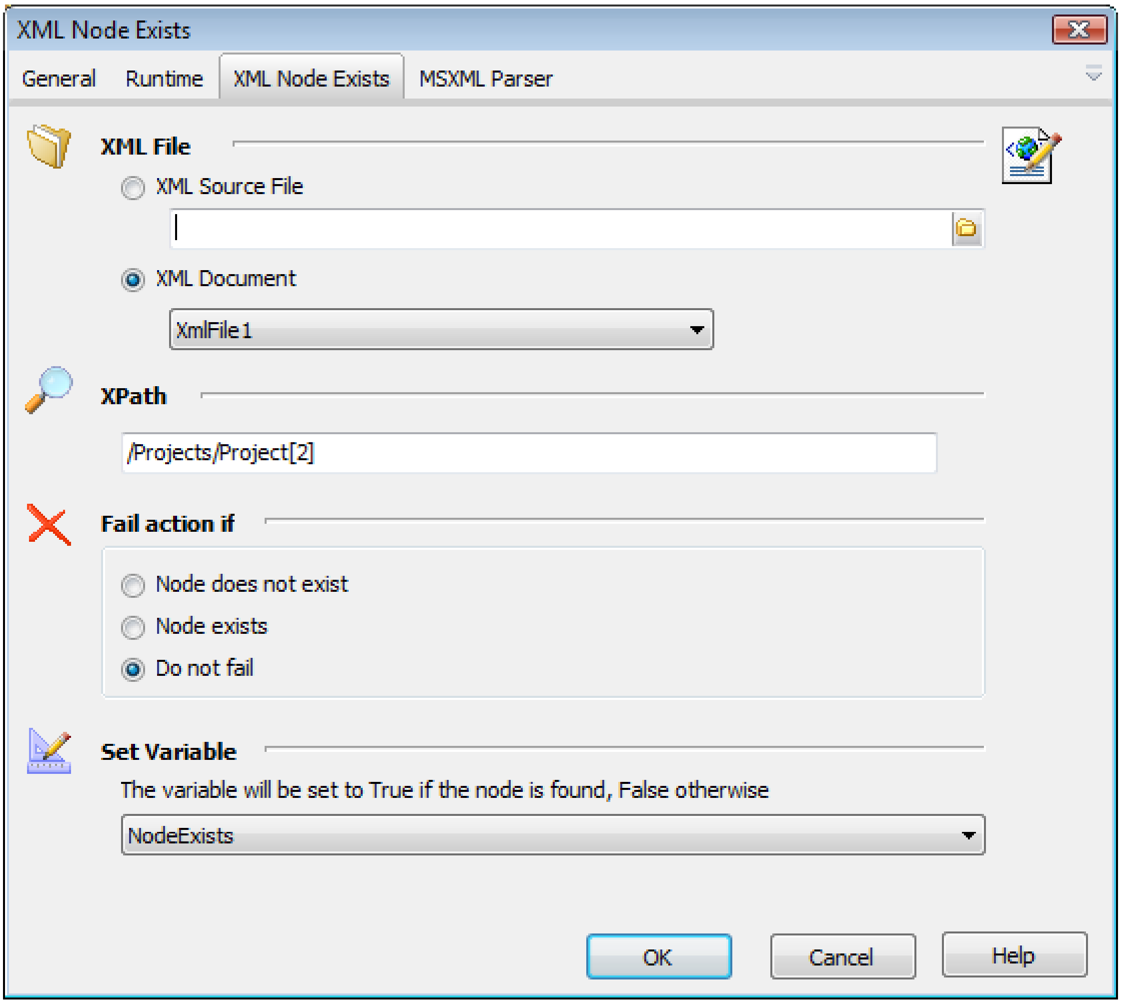
MSXML Parser Page

This page allows you to set some options which affect the parser when it loads the XML. See the XML Parser Options topic for details.

The parser options set which are set by the Document Define action will be used by every action which uses this XML Document object.

6.28.10 XML Node Exists Action

The XML Node Exists action enables you to check if the node specified by an XPath exists in an XML document.



XML File

○ XML Source File

● XML Document

XmlFile1

XPath

/Projects/Project[2]

Fail action if

○ Node does not exist

○ Node exists

● Do not fail

Set Variable

The variable will be set to True if the node is found, False otherwise

NodeExists

XML File

The XML file can be loaded from a file or a document defined using the XML Document Define action.

XPath

Specify an XPath which evaluates to the nodes you wish to check for. For more information on XPath syntax, try the w3schools tutorial.

Fail action if

The action can be set to fail if the node exists or doesn't exist, or to not fail at all.

Set Variable

The result of the search (True or False) can be optionally written to a variable.

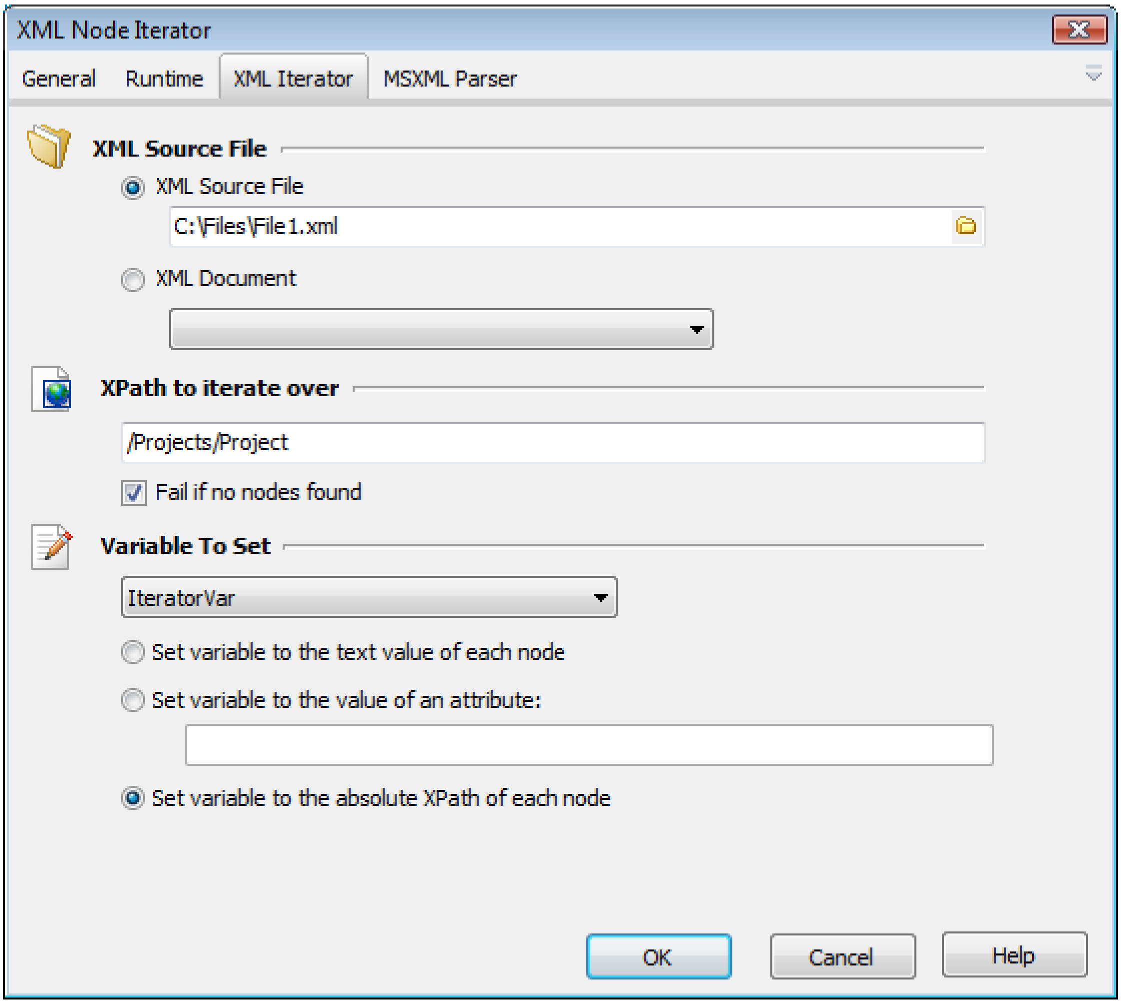
MSXML Parser Page

This page allows you to set some options which affect the parser when it loads the XML.

See the XML Parser Options topic for details.

6.28.11 XML Node Iterator

The XML Iterator action sets a variable by iterating over the nodes of an XML file. For general information about iterators, click here.



XML Source File

● XML Source File

C:\Files\File1.xml

○ XML Document

XPath to iterate over

/Projects/Project

☑ Fail if no nodes found

Variable To Set

IteratorVar

○ Set variable to the text value of each node

○ Set variable to the value of an attribute:

● Set variable to the absolute XPath of each node

XML Source File

Specify an XML file or document to read from.

XPath to iterate over

Specify an XPath which evaluates to the nodes you wish to iterate over. For more information on XPath syntax, try the w3schools tutorial.

FB Variable To Set

Specify the variable name that you wish the iterator to set on each iteration. To create a new variable, choose Edit Variables from the Tools menu.

- **Set variable to the text value of each node**
- **Set variable to the value of an attribute**

  For each iteration, the specified attribute will be evaluated at the given node, and the variable will be set to the value of that attribute.

- **Set variable to the absolute XPath of each node**

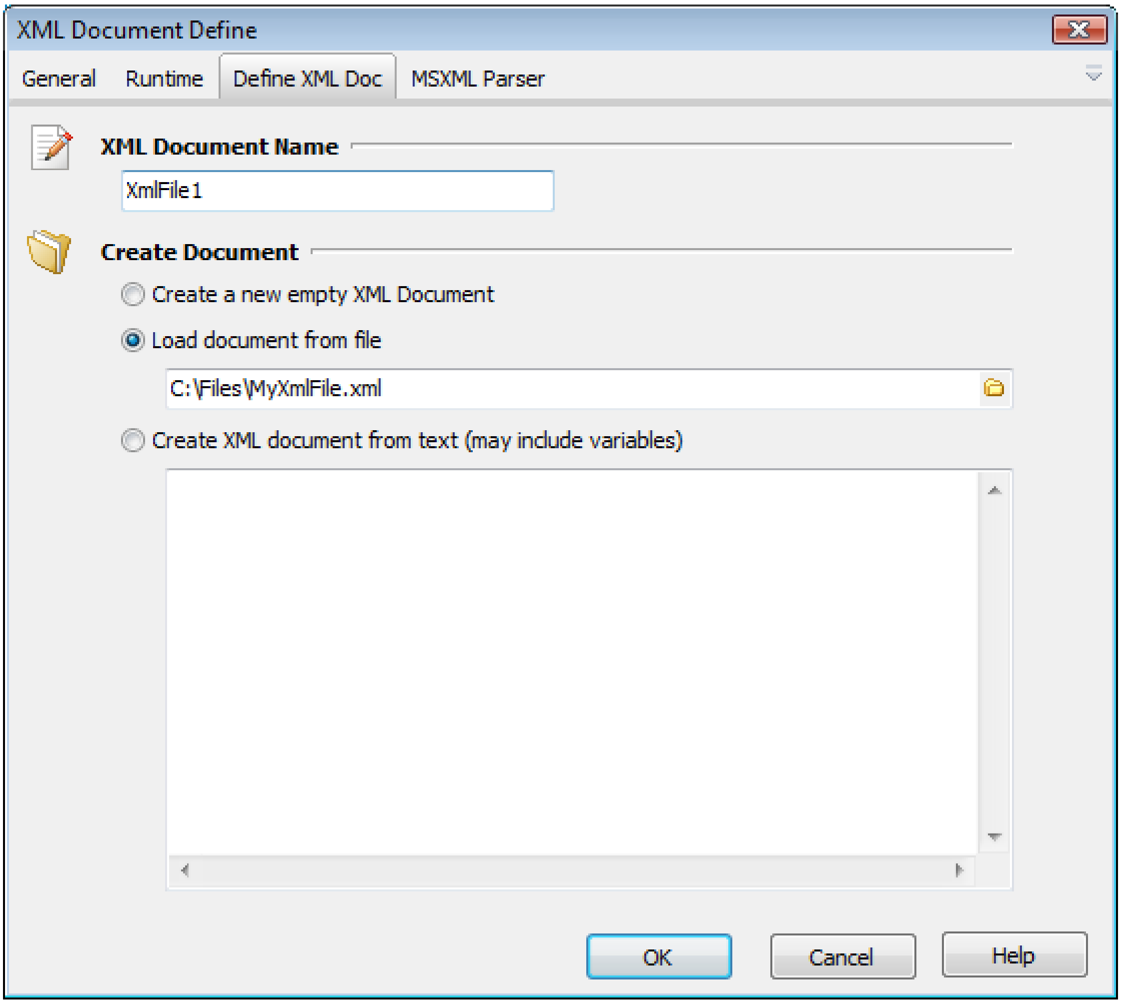
  For each iteration, the variable will be set to the absolute XPath of the selected node. The XPath variable can then be used to provide XPaths to the other XML Actions.

MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.28.12 XML Parser Options

All of the XML Actions include an MSXML Parser property page. This allows you to set the version of the MSXML Parser to use for the file, as well as some parser-specific options.



XML Document Define

XML Document Name

XmlFile1

Create Document

○ Create a new empty XML Document

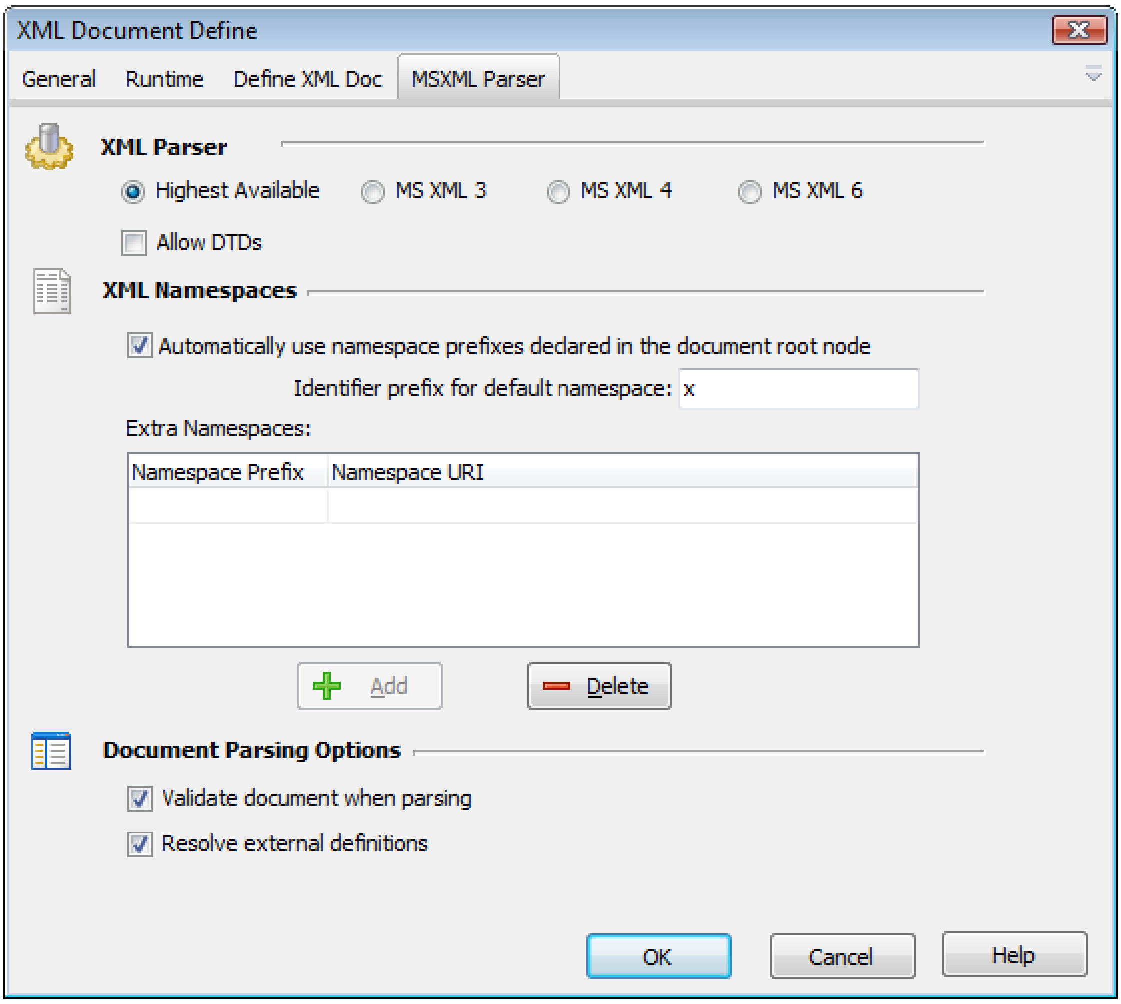
● Load document from file

C:\Files\MyXmlFile.xml

○ Create XML document from text (may include variables)

XML Parser

You can specify which version of the MS XML parser to use to parse the file. In most cases, "Highest Available" will be fine.



XML Parser

● Highest Available ○ MS XML 3 ○ MS XML 4 ○ MS XML 6

□ Allow DTDs

XML Namespaces

☑ Automatically use namespace prefixes declared in the document root node

Identifier prefix for default namespace: x

Extra Namespaces:

Namespace Prefix	Namespace URI

Document Parsing Options

☑ Validate document when parsing

☑ Resolve external definitions

XML Namespaces

"Automatically use namespace prefixes declared in document root node"

If this option is checked then the XML Parser will expand any xmlns: attributes that it finds in the document root node. This enables the use of XPaths which reference nodes with XML Namespace prefixes.

This option is included to overcome a shortcoming in MS XML, whereby XPaths do not automatically take into consideration namespaces or namespace prefixed nodes within a document.

"Identifier prefix for default namespace"

If the document specifies a default namespace, you will still need to use a custom prefix in order to reference it in your XPath. Specify the prefix to use here.

"Extra Namespaces"

If a namespace prefix is declared somewhere other than the document root node, you need to list it here so that you can specify an XPath which references that namespace.

An alternative is to add the relevant declaration to the document root node.

Document Parsing Options

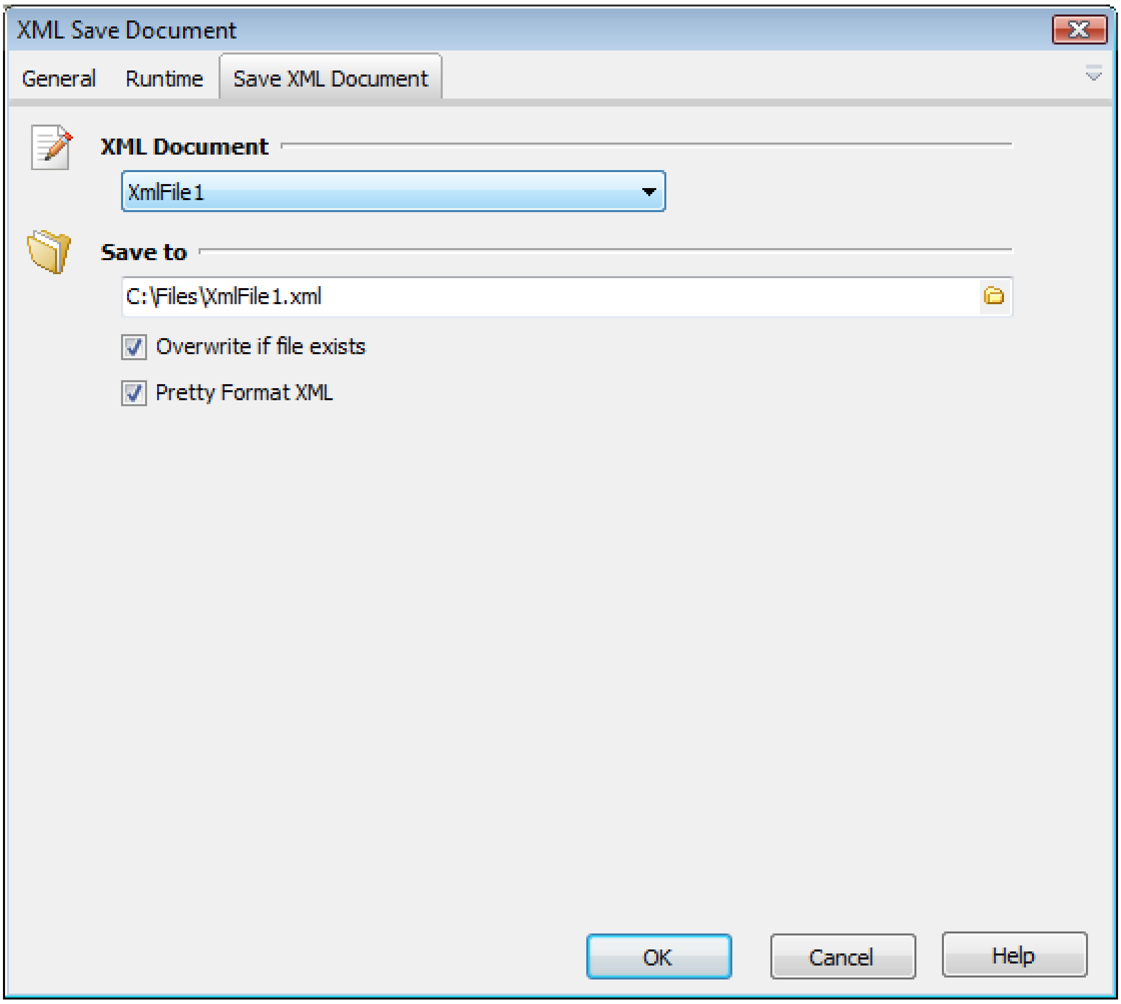
"Validate document when parsing" - If this option is enabled, the XML content will be validated when it is loaded by the parser. If validation fails, the action will fail. Note that XML which is not well formed will always fail the action.

"Resolve external definitions" - If this option is enabled, the XML Parser will attempt to load any external definitions (XML Schema, DTDs, etc.) which are specified inside the XML file. Parsing will fail if any external files are not available.

6.28.13 XML Save Document

Save an XML Document Object created with the XML Document Define action to disk.

Simply select the XML Document Object and a filename to save the xml file to.



XML Document: XmlFile1

Save to: C:\Files\XmlFile1.xml

☑ Overwrite if file exists

☑ Pretty Format XML

7 Automating Automise

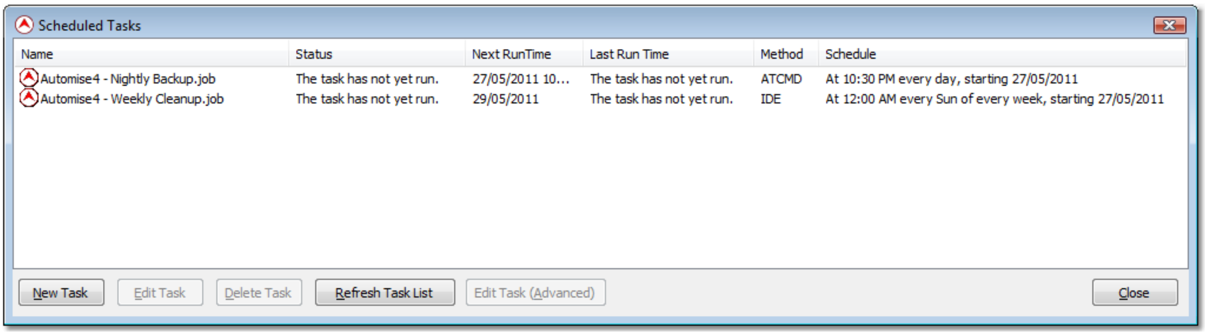
There are a number of ways to automate Automise:

- Executing Automise (the IDE) from the command line
- Using the built in scheduler
- Executing ATCMD on the command line

7.1 Automise IDE

7.1.1 Scheduling projects

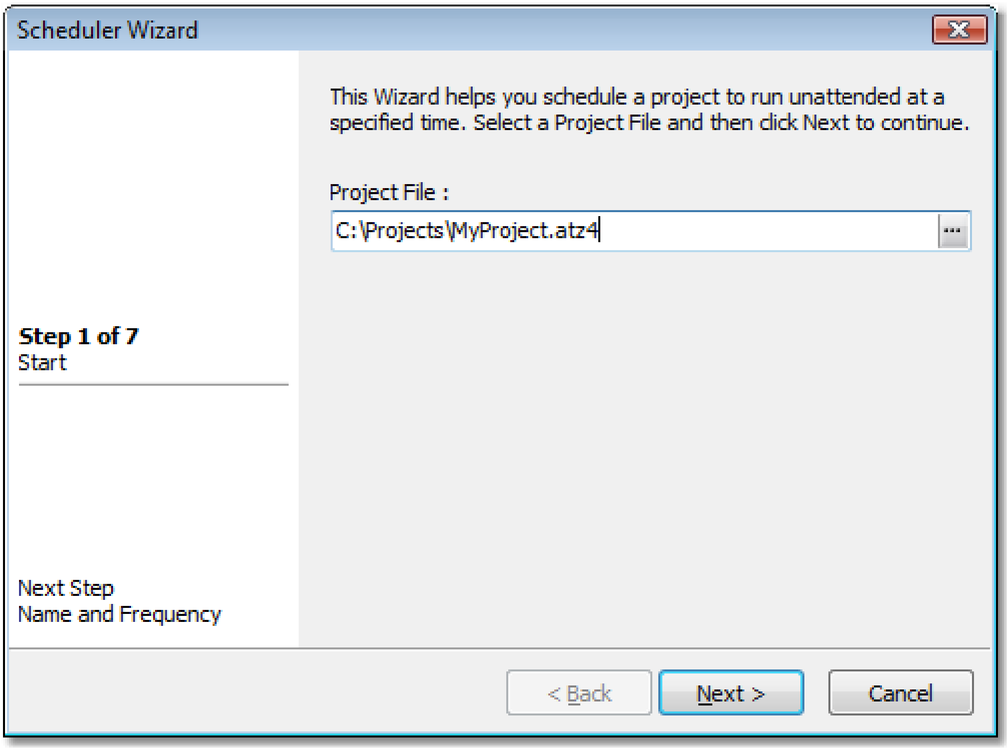
Automise can schedule projects to be run by the Windows Scheduling service. To access the scheduling functions, choose Scheduled projects from the Tools menu.



Scheduled Tasks

Name	Status	Next RunTime	Last Run Time	Method	Schedule
Automise4 - Nightly Backup.job	The task has not yet run.	27/05/2011 10...	The task has not yet run.	ATCMD	At 10:30 PM every day, starting 27/05/2011
Automise4 - Weekly Cleanup.job	The task has not yet run.	29/05/2011	The task has not yet run.	IDE	At 12:00 AM every Sun of every week, starting 27/05/2011

To Create a new scheduled project, click on the New Task button. This will display the wizard that will step you through scheduling a project.



Scheduler Wizard

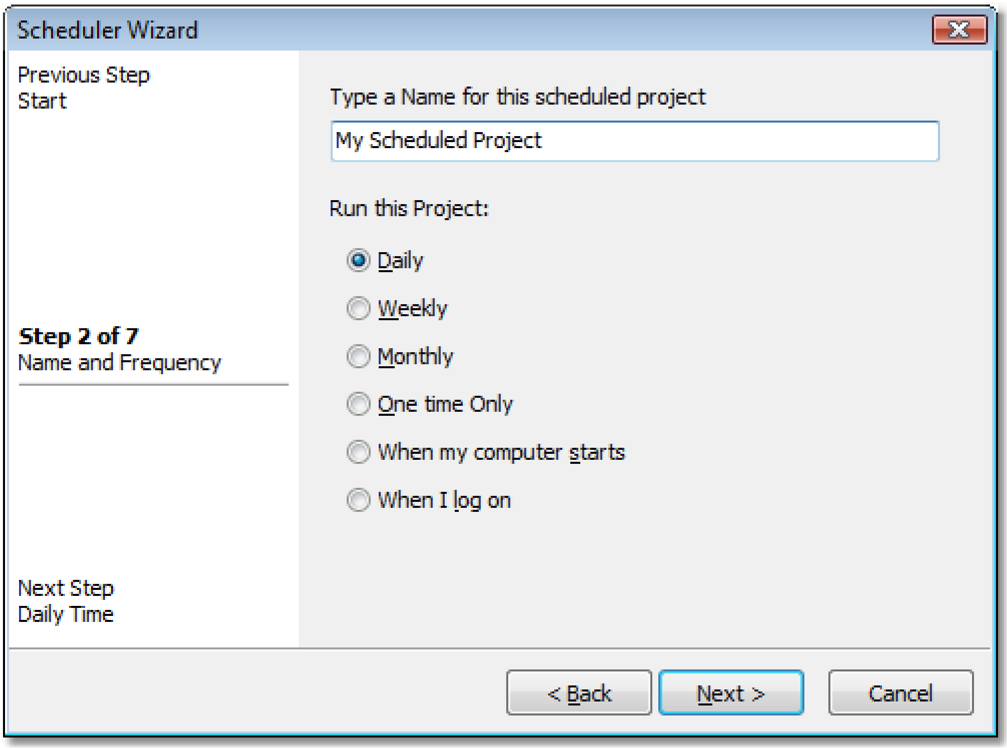
This Wizard helps you schedule a project to run unattended at a specified time. Select a Project File and then click Next to continue.

Project File : C:\Projects\MyProject.atz4

Step 1 of 7
Start

Next Step
Name and Frequency

Select a Automise Project File and then click Next to continue.



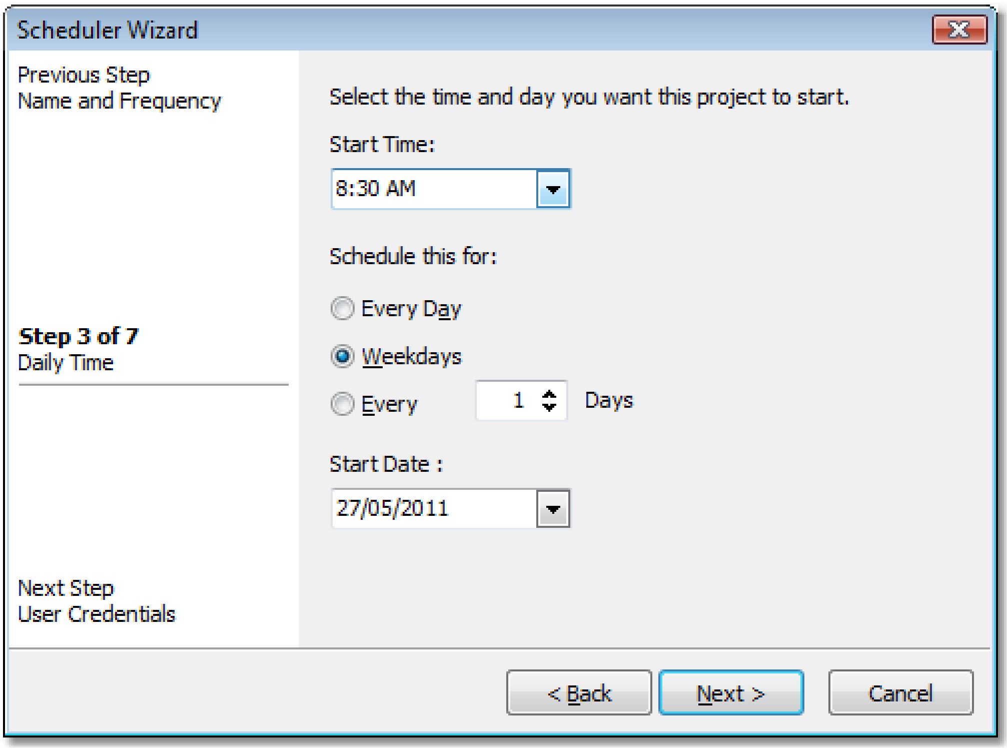
Type a Name for this scheduled project

My Scheduled Project

Run this Project:

- ● Daily
- ○ Weekly
- ○ Monthly
- ○ One time Only
- ○ When my computer starts
- ○ When I log on

Enter a Title for the scheduled project and then specify the frequency of the project, then click on next to continue.



Select the time and day you want this project to start.

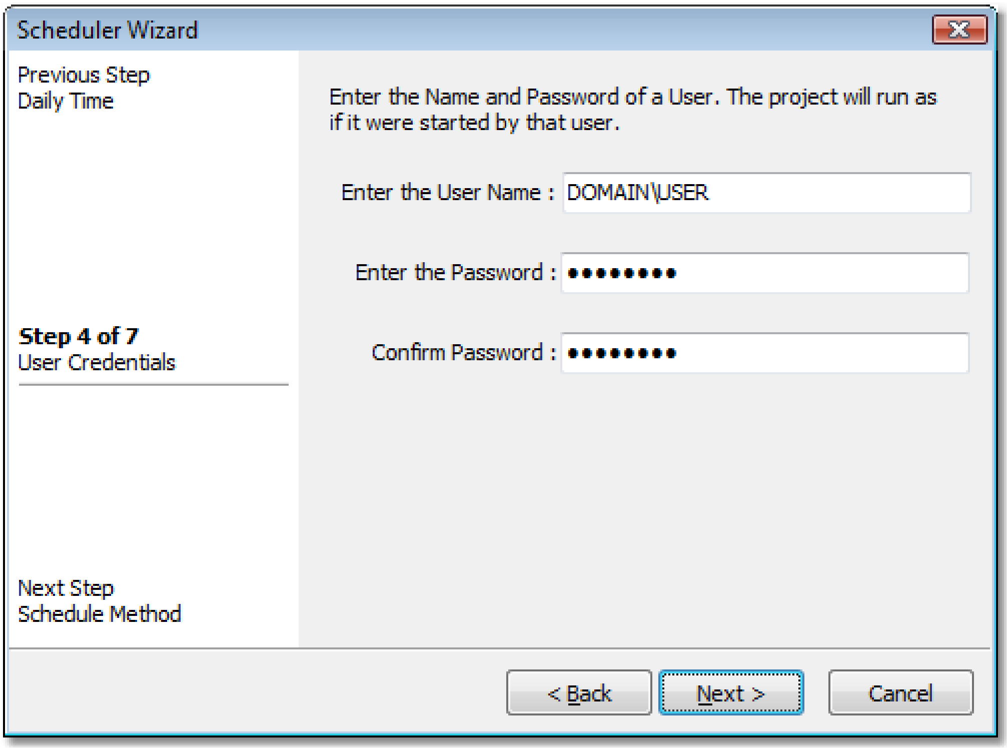
Start Time: 8:30 AM

Schedule this for:

- ○ Every Day
- ● Weekdays
- ○ Every 1 Days

Start Date : 27/05/2011

Specify the Time and starting date for the schedule, then click next to continue.



Scheduler Wizard

Previous Step
Daily Time

Step 4 of 7
User Credentials

Next Step
Schedule Method

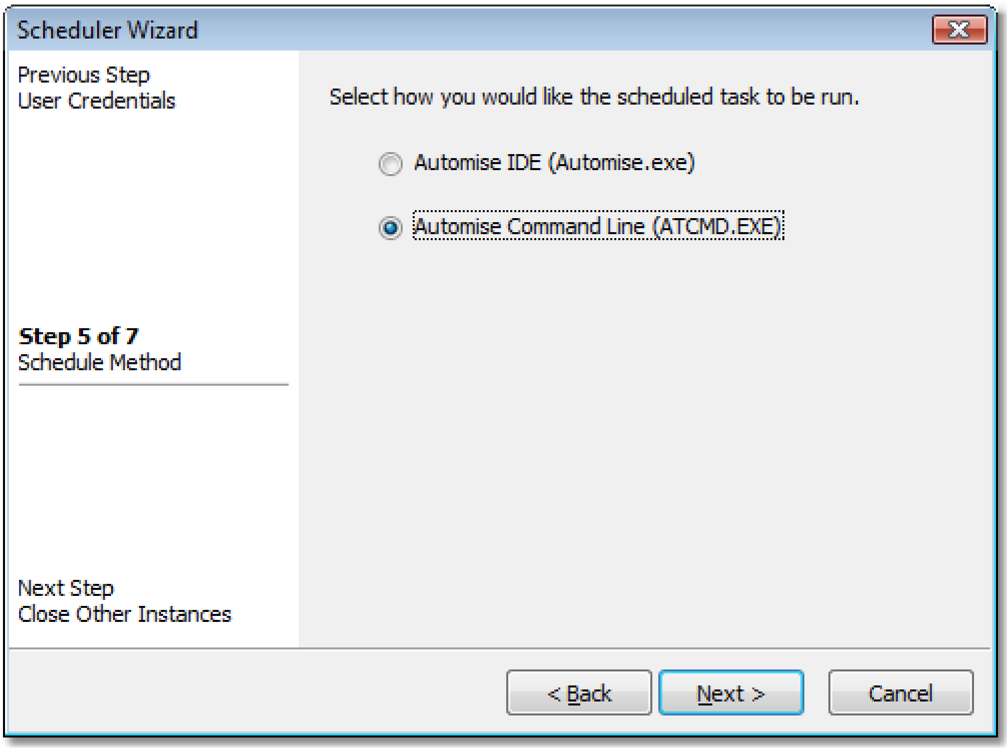
Enter the Name and Password of a User. The project will run as if it were started by that user.

Enter the User Name : DOMAIN\USER

Enter the Password : ••••••••

Confirm Password : ••••••••

Provide the user name that the project will run as and the password, then click continue.



Scheduler Wizard

Previous Step
User Credentials

Step 5 of 7
Schedule Method

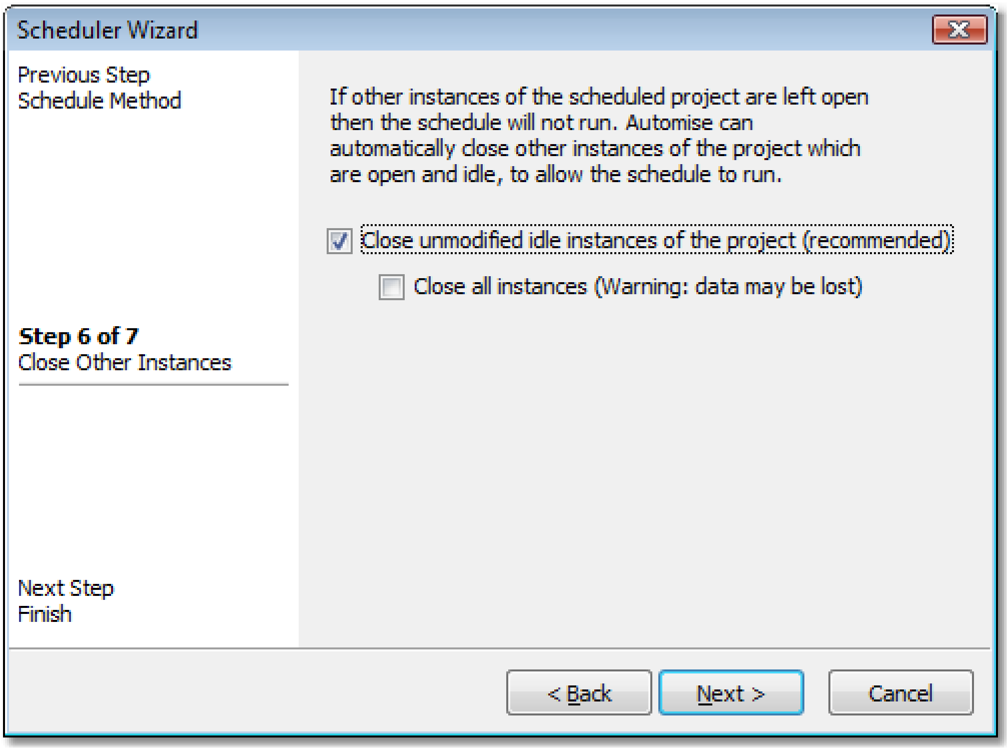
Next Step
Close Other Instances

Select how you would like the scheduled task to be run.

○ Automise IDE (Automise.exe)

● Automise Command Line (ATCMD.EXE)

You can choose to run a scheduled project in the IDE or from the command line tool (ATCMD). ATCMD is recommended for projects that will run unattended.



Scheduler Wizard

Previous Step
Schedule Method

Step 6 of 7
Close Other Instances

Next Step
Finish

If other instances of the scheduled project are left open then the schedule will not run. Automise can automatically close other instances of the project which are open and idle, to allow the schedule to run.

☑ Close unmodified idle instances of the project (recommended)

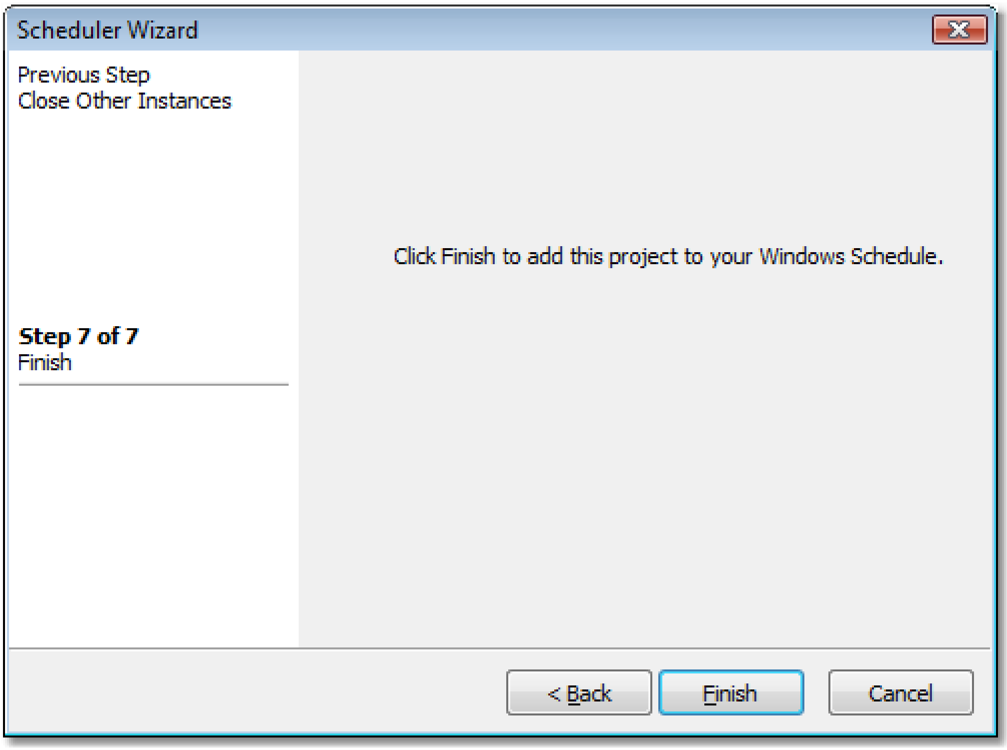
☐ Close all instances (Warning: data may be lost)

Automise can optionally close other instances of itself which are using the scheduled project. This is useful because only one Automise instance can have a single project open at a time.

The **Close unmodified idle instances of the project** option will close any other Automise instances which are not running and have no unsaved changes. This is a safe option, and it is recommended that it is enabled.

The **Close all instances** option will close any Automise instances which are using the same project file. Instances will be closed forcibly if necessary. This option will lose all saved changes, and may forcibly terminate projects in progress if they do not stop cleanly. It is recommended this option is only used sparingly.

For more information on these two options, see the Command Line switches `-o` and `-of` in both the IDE and ATCMD reference topics..



Scheduler Wizard

Previous Step
Close Other Instances

Click Finish to add this project to your Windows Schedule.

Step 7 of 7
Finish

Click Finish to add this project to your Scheduled Tasks.

7.1.2 IDE Command Line Interface

(To run projects from the command line in a console window, use the Command line executable.)

Automise Command Line Options

Usage : Automise.exe [switches] <projectfile>

Switches

Switch	Behaviour	Example
-n or **/n**	Hide the splash screen when starting up.	/n
-r or **/r**	Automatically run the project file passed in on the command line.	/r "MyProject.fbp7"
-e or **/e**	Exit when done running.*	/r /e
-m or **/m**	Minimize IDE when starting. This is useful when you are scheduling a project and you don't want the IDE to appear (instead you will just see the tray icon)	/m
-f or **/f**	Don't exit if an error occurs while running.*	/r /e /f
-v or **/**	Set Automise variables, in the form	/vOutputDIR="d:

v	VarName=VarValue. The variables must be already defined Project or User variables.*
	Separate multiple name/value pairs with semicolons. If the variable value contains spaces, enclose it in quotes.
		\Output\My Output";
		DCUDIR=d:\temp\dcu

-a or **/ a**	Allow interactive actions (prompt for variables, message boxes, etc.) in an automatic project.*	/r /a
-o or / **o**	Close any other instances of the Automise IDE which are idle with unmodified copies of the same project file.*	/r /o
-of or / **of**	Forcibly close any other instances of the Automise IDE which are open with a copy of the same project file. The IDE will first attempt to close cleanly, otherwise will terminate a running project or force terminate a stuck project. Unsaved changes to projects will be lost. Mutually exclusive with -o.*†	/r /of
-tl or **/ tl**	Log to a temporary log file, instead of the project log file. With this option enabled, the same project can be open multiple times concurrently. Log history will be lost when the Automise IDE closes.	-tl
-z or **/ z**	Enable live logging during automated run. Normally live logging is disabled when using **-r** (for performance reasons.) This option can be used for debugging and monitoring purposes.*	-r -z

* = Only valid when Auto Run (**-r**) switch is included.
† = Warning: Forcibly closing other projects may cause loss of data and should be used sparingly.

When running Automise as a scheduled task, you should always use the /r and /e switches.

See Also

ATCMD Command Line Tool | Scheduling projects

7.1.2.1 Exit Codes

The Automise IDE uses the following Windows exit codes to indicate the outcome of the project :

Exit Code	Description
0	The run completed Successfully
1	An Error occurred in the run. Check the log for the details.
2	Invalid project when auto running project (-r) with "exit when done" (-e.)

998	Automise was force terminated following a remote request from another Automise instance (started using the -of option.)
999	The user elected to force terminate Automise after the running project failed to terminate.

See the Command Line Exit Codes topic to see the extended range of exit codes available from ATCMD.exe.

See Also

IDE Command Line Interface | Scheduling projects

7.2 Command Line version

ATCMD is a text mode version of Automise which can run projects from the command line. It is installed in the Automise directory.

ATCMD can be faster at executing than the IDE, as the IDE has more visual overhead.

Command Line Syntax

The command line syntax for ATCMD is as follows:

ATCMD.exe [options**] [/P]**<projectfile>

Options :

Switch	Behaviour	Example
/P	Automise Project File to execute The /P is optional, you can just specify the project file name by itself instead (although in that case it must be after any other options.)	/P"C:\Build\MyProject.atpz" **or** "C:\Build\MyProject.atpz"
/I	Ignore all system message (eg. Ctrl-C, Shutdown.)	/I
/A	Allow interactive actions (message dialogs, prompt for variables, etc.)	/A
/V	Set Automise variables, in the form VarName=VarValue. The variables must be already defined Project or User variables. Separate multiple name/value pairs with semicolons. If the variable value contains spaces, enclose it in quotes.	/VMAJORVER=3;MINORVER=1
/C	Validate project before execution.	/C
/S	Disable logging. No log file will be generated, and an existing log file will not be updated. The Export Log action will not function. This option may give a significant performance speedup if the project runs many actions very quickly. Console output is still enabled (see	/S

	below.) With this option (or **/TL**) enabled, the same project can be opened multiple times concurrently.	
/H	Disable hierarchical logging.	/H
/TL	Logging is still performed, but to a temporary log file. This means actions such as the Export Log action still function. With this option (or **/S**) enabled, the same project can be opened multiple times concurrently.	/TL
@	Read arguments from a parameter text file. Each line of the file should specify a different command line parameter.	@"MyParameters.txt"
/O	Close any other instances of the %ProductName% IDE which are idle with unmodified copies of the same project file.	
/OF	Forcibly close any other instances of the %ProductName% IDE which are open with a copy of the same project file. The IDE will first attempt to close cleanly, otherwise will terminate a running project or force terminate a stuck project. Unsaved changes to projects will be lost. Mutually exclusive with **/o**.[†]	

† = Warning: Forcibly closing other projects may cause loss of data and should be used sparingly.

Options for console output (this affects the data written to the console, not to the project's log file:

Switch	**Behaviour**	**Example**
/L	Output console messages to a specified text file instead of the console.	/L"C:\Temp\MyProjectOutput.txt"
/LA	Like **/L**, but append to the text file instead of overwriting.	/L"C:\Temp\ExistingOutput.txt"
/LN	Like **/L**, but fail if the file already exists (instead of overwriting.)	/L"C:\Temp\MoreOutput.txt"
/B	Suppress Automise banner.	/B
/XL	Don't output Action List start and end messages to the console.[*°]	/XL
/XA	Don't output action start messages to the console.[*°]	/XA
/XM	Don't send action output to the console.[*°]	/XAM

* = Or the specified text file, if using **/L**, **/LA** or **/LN**.
° = Can be combined in a single switch, ie **/XLAM**.

See Also

Scheduling projects | IDE Command Line Interface

7.2.1 Exit Codes

ATCMD uses the following Windows exit codes to indicate the outcome of the project run:

Exit codes:

Exit Code	Reason
0	No error, build succeeded.
1	Build Failed.
2	Invalid arguments on command line.
3	Project file not found.
4	Expected output file name (With **/L**, **/LN** or **/LA**.)
5	Output file error (file already exists, cannot be overwritten, or is not writable. (With **/L**, **/LN** or **/LA.**)
6	Invalid project.
7	Unhandled Internal Exception.
8	Aborted.
9	Project Load Error.
10	Initialization error.
11	Project failed validation (with **/C**.)
12	Package loading error.
13	Required package not loaded.
14	Error setting variable (with **/V**.)
15	Unknown unhandled exception.
16	Parameter file missing (with **@**.)
17	Parameter file error (With **@**.)
18	Wrong Automise version.

See Also

IDE Exit Codes

8 Tips and Tricks

8.1 Two Environments

A common situation is developing an Automise project on one machine - the *development* machine - and needing it to also run on a second machine - the *production* machine. However, unless the two machines are set up identically, this can be tricky. One easy solution is to use User Variables: on each machine, simply create the necessary variables and define them as User. However, for centralised control, you may wish to use a single shared INI file as follows.

Let's assume that whenever you run a project on your local machine (named *John*), you want it to take place in C:\Projects\MyProjects. When you run the project on the production machine (named *Server01*), it takes place in J:\Projects\Latest

1. Create an INI file with the differences

First, create a .ini file in a shared location with the parameters. Give each section the name of the machine it applies to:

```
[Server01]
ProjectHome=J:\Projects\Latest

[John]
ProjectHome=C:\Projects\MyProjects
```

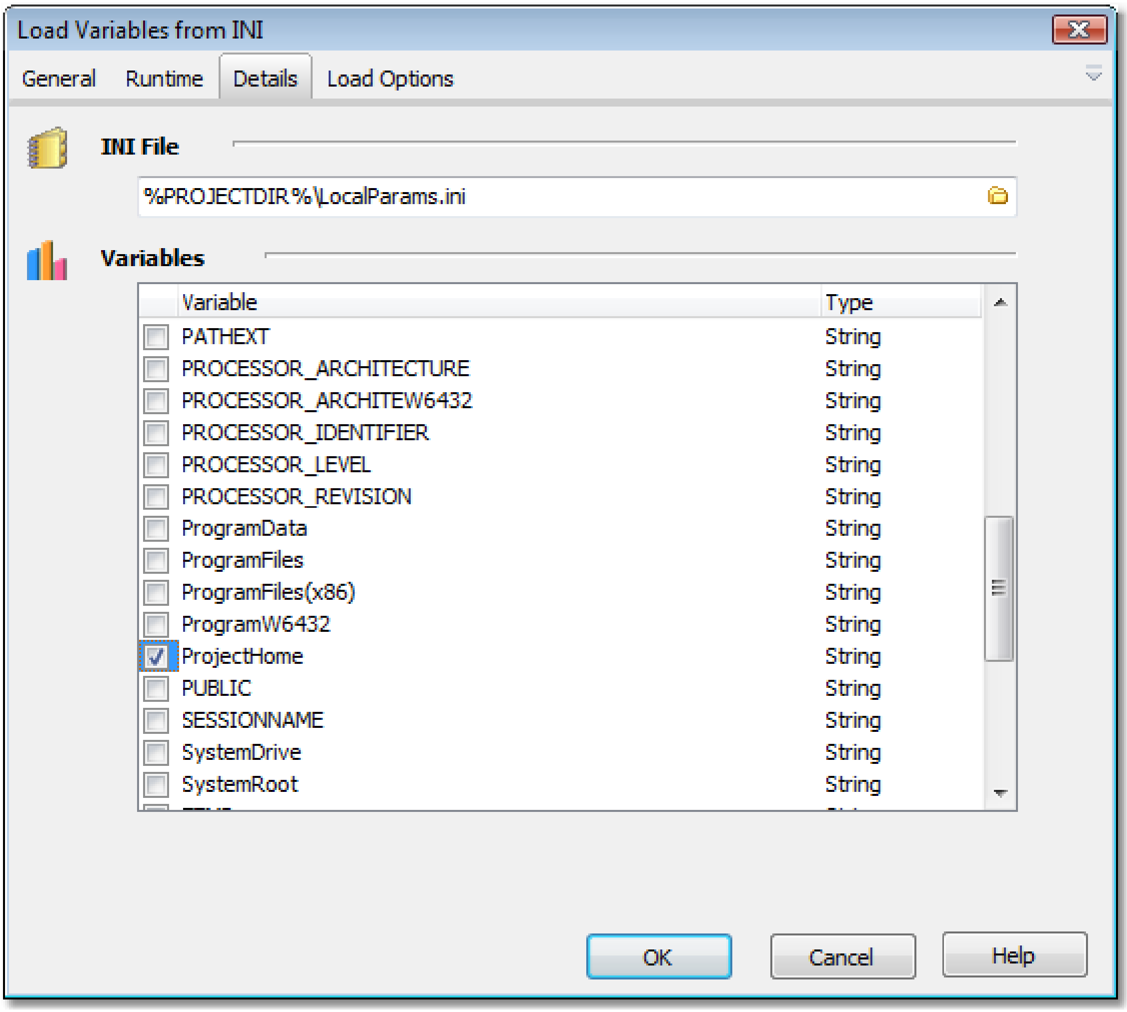
Name the file something like "LocalParams.ini".

2. Make your project depend on these parameters

Define variables with the names you used in the INI file. Update all actions that use these locations to use the variables instead.

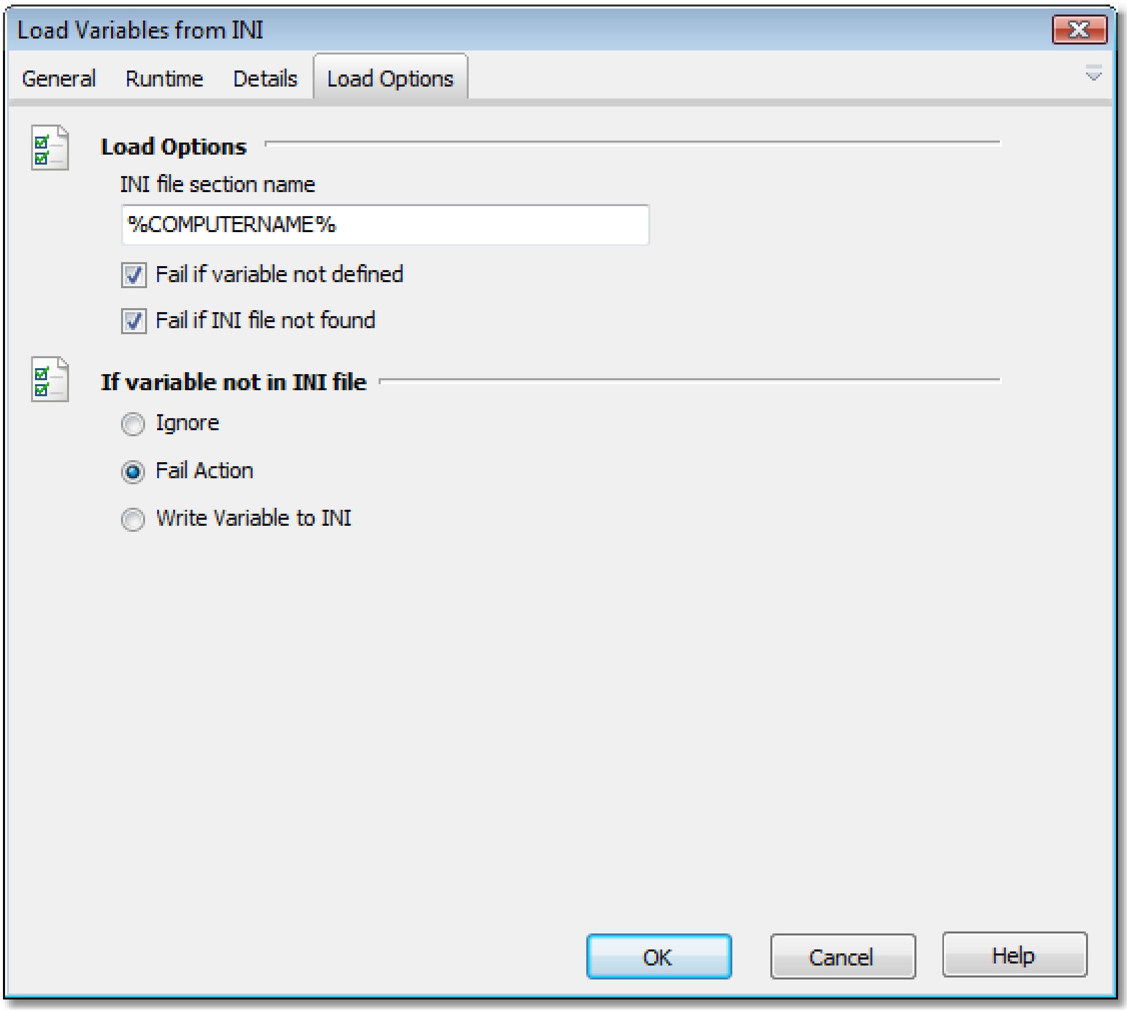
3. Load the INI file as the first step in the project

At the start of the project, add a "Load Variables from INI" that loads the relevant variables from the INI file. Use the %COMPUTERNAME% variable to load from the right section:



%PROJECTDIR%\LocalParams.ini

	Variable	Type
□	PATHEXT	String
□	PROCESSOR_ARCHITECTURE	String
□	PROCESSOR_ARCHITEW6432	String
□	PROCESSOR_IDENTIFIER	String
□	PROCESSOR_LEVEL	String
□	PROCESSOR_REVISION	String
□	ProgramData	String
□	ProgramFiles	String
□	ProgramFiles(x86)	String
□	ProgramW6432	String
☑	ProjectHome	String
□	PUBLIC	String
□	SESSIONNAME	String
□	SystemDrive	String
□	SystemRoot	String



Load Variables from INI

Load Options

INI file section name: %COMPUTERNAME%

☑ Fail if variable not defined

☑ Fail if INI file not found

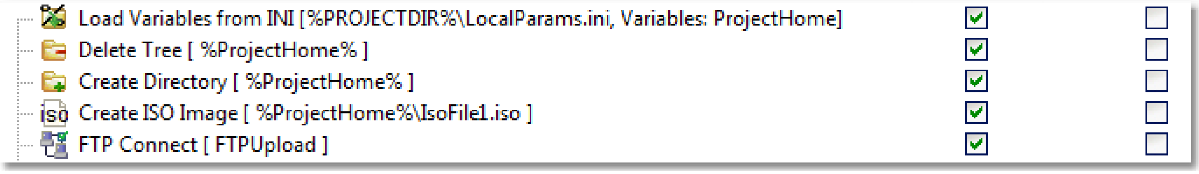
If variable not in INI file

○ Ignore

● Fail Action

○ Write Variable to INI

4. The finished product



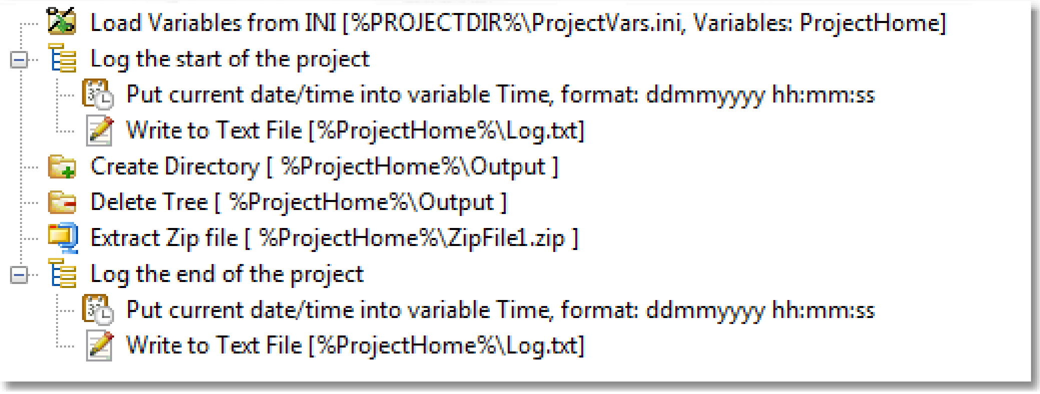
Load Variables from INI [%PROJECTDIR%\LocalParams.ini, Variables: ProjectHome]	☑	□
Delete Tree [%ProjectHome%]	☑	□
Create Directory [%ProjectHome%]	☑	□
Create ISO Image [%ProjectHome%\IsoFile1.iso]	☑	□
FTP Connect [FTPUpload]	☑	□

8.2 Reusing Logic

When you have come up with a useful sequence of actions, you may want to reuse it. Here are several methods, in increasing levels of sophistication.

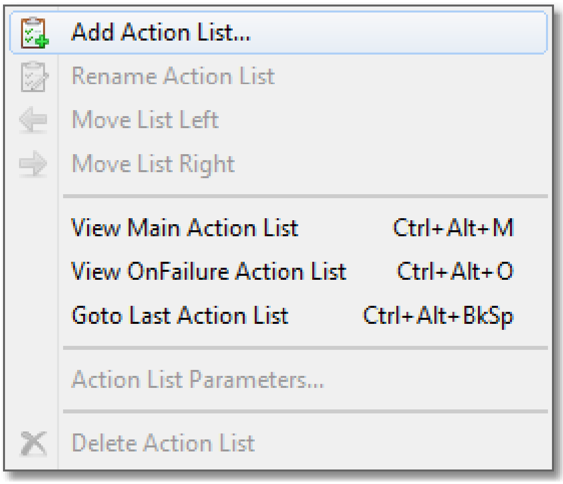
Action lists

Let's take the example of a couple of actions used to log to an external log file with a date and time stamp. Before writing to the log with the Write Text File action, the Time variable is updated using the Get Date Time action.



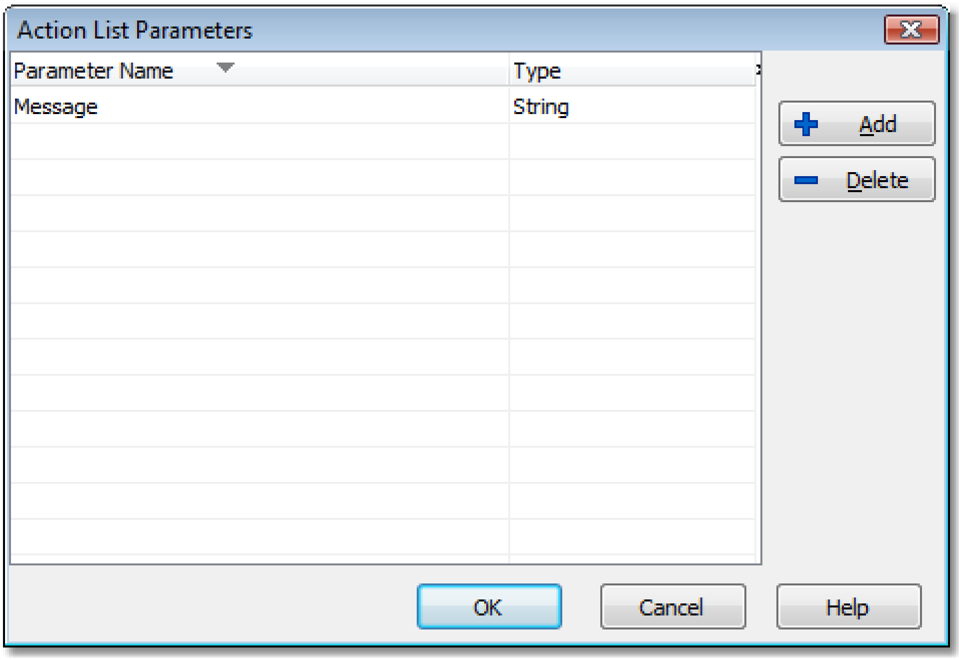
Load Variables from INI [%PROJECTDIR%\ProjectVars.ini, Variables: ProjectHome]
Log the start of the project
Put current date/time into variable Time, format: ddmmyyyy hh:mm:ss
Write to Text File [%ProjectHome%\Log.txt]
Create Directory [%ProjectHome%\Output]
Delete Tree [%ProjectHome%\Output]
Extract Zip file [%ProjectHome%\ZipFile1.zip]
Log the end of the project
Put current date/time into variable Time, format: ddmmyyyy hh:mm:ss
Write to Text File [%ProjectHome%\Log.txt]

Here, the log actions have been copied. This is not ideal, as if we improve the logic later on, we'll have to edit all those actions. Instead, create a new action list, called Log:

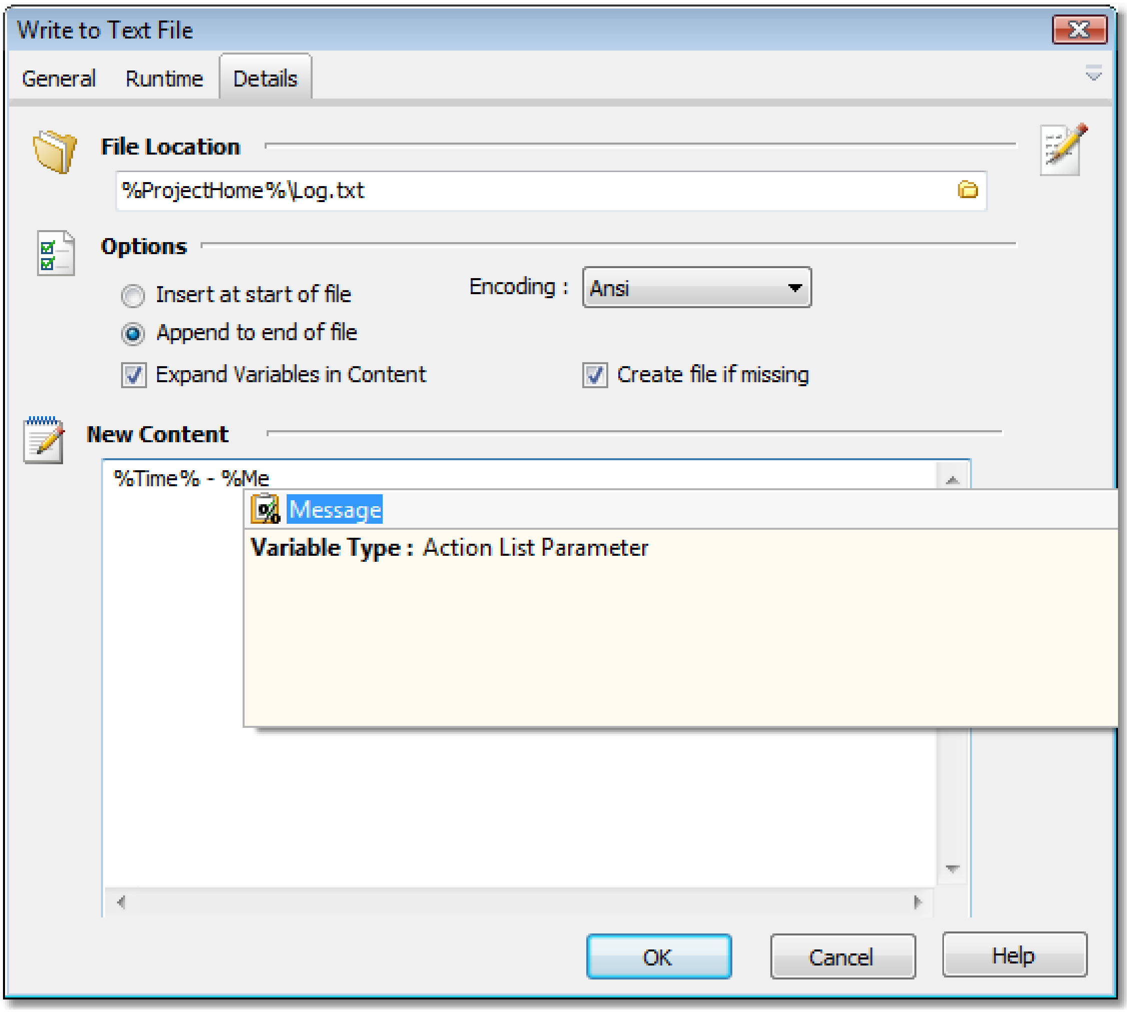


Add Action List...
Rename Action List
Move List Left
Move List Right
View Main Action List Ctrl+Alt+M
View OnFailure Action List Ctrl+Alt+O
Goto Last Action List Ctrl+Alt+BkSp
Action List Parameters...
Delete Action List

Move the actions to the new action list. Now, create a parameter to contain the log message, by right clicking on the action list title.



Replace the hard-coded message in the "Write to Text File" action. Action list parameters behave just like normal variables.



Write to Text File

File Location

%ProjectHome%\Log.txt

Options

○ Insert at start of file

● Append to end of file

☑ Expand Variables in Content

Encoding : Ansi

☑ Create file if missing

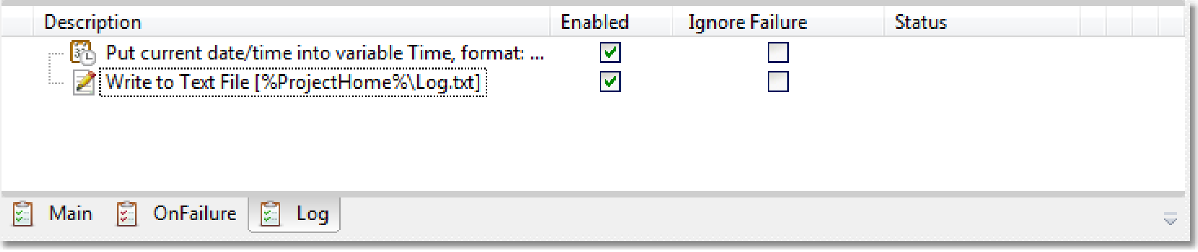
New Content

%Time% - %Me

Message

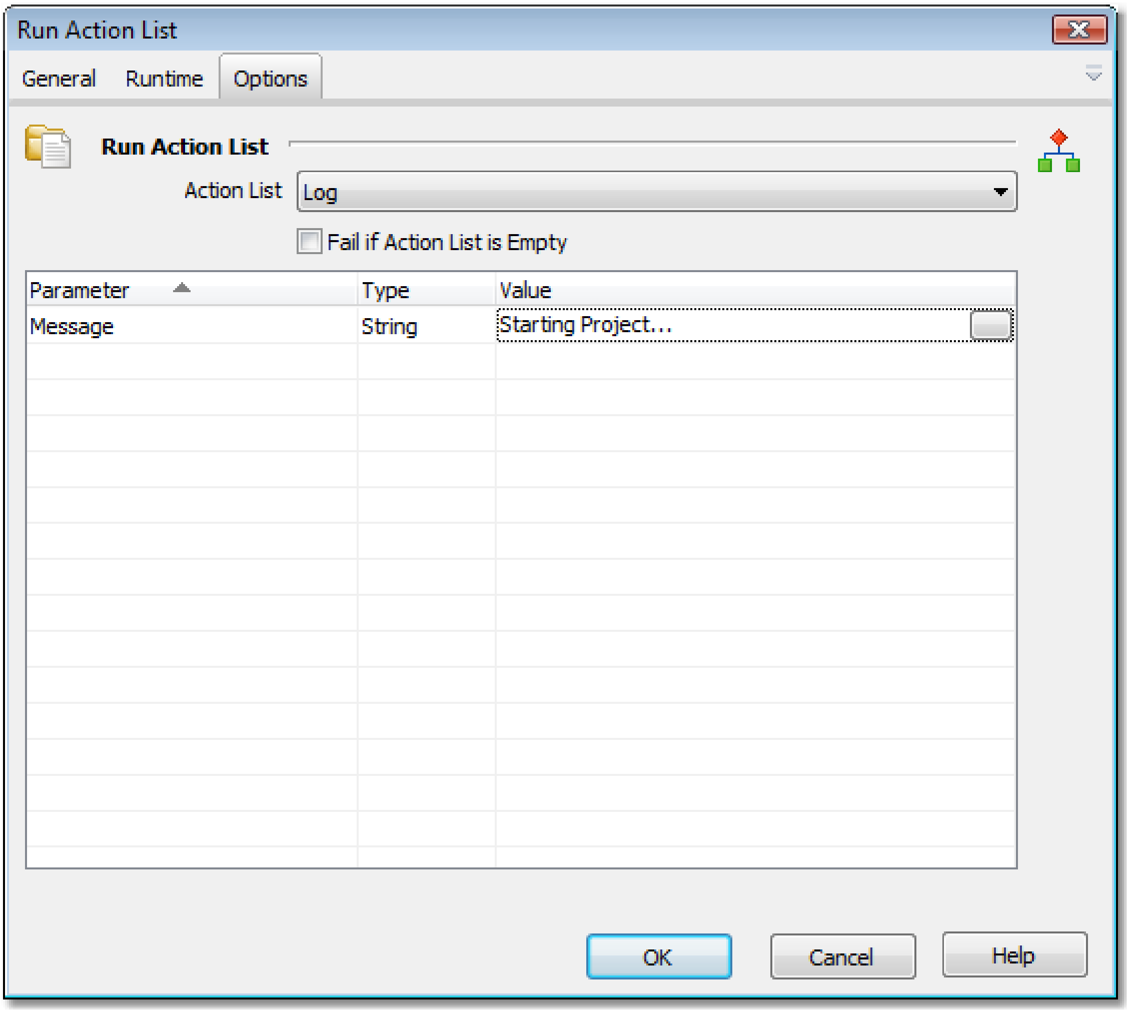
Variable Type : Action List Parameter

The action list now looks like this:



Description	Enabled	Ignore Failure	Status
Put current date/time into variable Time, format: ...	☑	□	
Write to Text File [%ProjectHome%\Log.txt]	☑	□	

The final step is to replace the original calls with Run Action List actions.



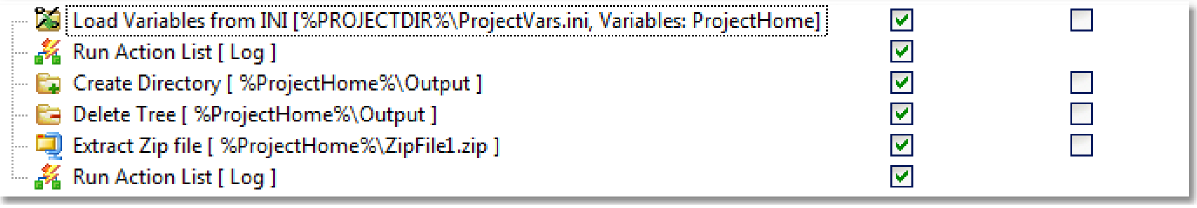
Run Action List

Action List: Log

□ Fail if Action List is Empty

Parameter	Type	Value
Message	String	Starting Project...

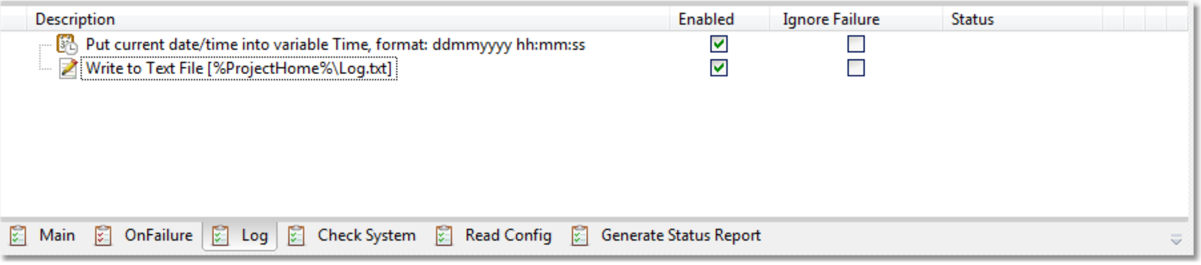
The result is this:



Load Variables from INI [%PROJECTDIR%\ProjectVars.ini, Variables: ProjectHome]	☑	□
Run Action List [Log]	☑	
Create Directory [%ProjectHome%\Output]	☑	□
Delete Tree [%ProjectHome%\Output]	☑	□
Extract Zip file [%ProjectHome%\ZipFile1.zip]	☑	□
Run Action List [Log]	☑	

External action lists

The next step in reusing action logic is to group these utility action lists into one project:



Description	Enabled	Ignore Failure	Status
Put current date/time into variable Time, format: ddmmyyyy hh:mm:ss	☑	□	
Write to Text File [%ProjectHome%\Log.txt]	☑	□	

You can now call these action lists from any project.

8.3 A Configurable project

Frequently, when you set up a Automise project to perform a task, you have to perform the same task in several different environments requiring different parameters. There are several ways to solve this problem. This tutorial demonstrates one approach, using INI files, prompts and switch statements to make one project configurable.

Lets say we have a project that deletes a set of old log files, creates a database backup and uploads the backup to an FTP server. Rather than create a different project with fixed values for each environment, we can define an INI file that contains the required values for each environment.

First we need to define a short code for each environment. If we wanted to run this project in *Development A*, *Development B* and *Production* then we could use the the following short codes: *DEVA*, *DEVB* and *PROD*.

INI File

Create an INI file with some parameters for each project and save it in the same location as your Automise project named *Configuration.ini*:

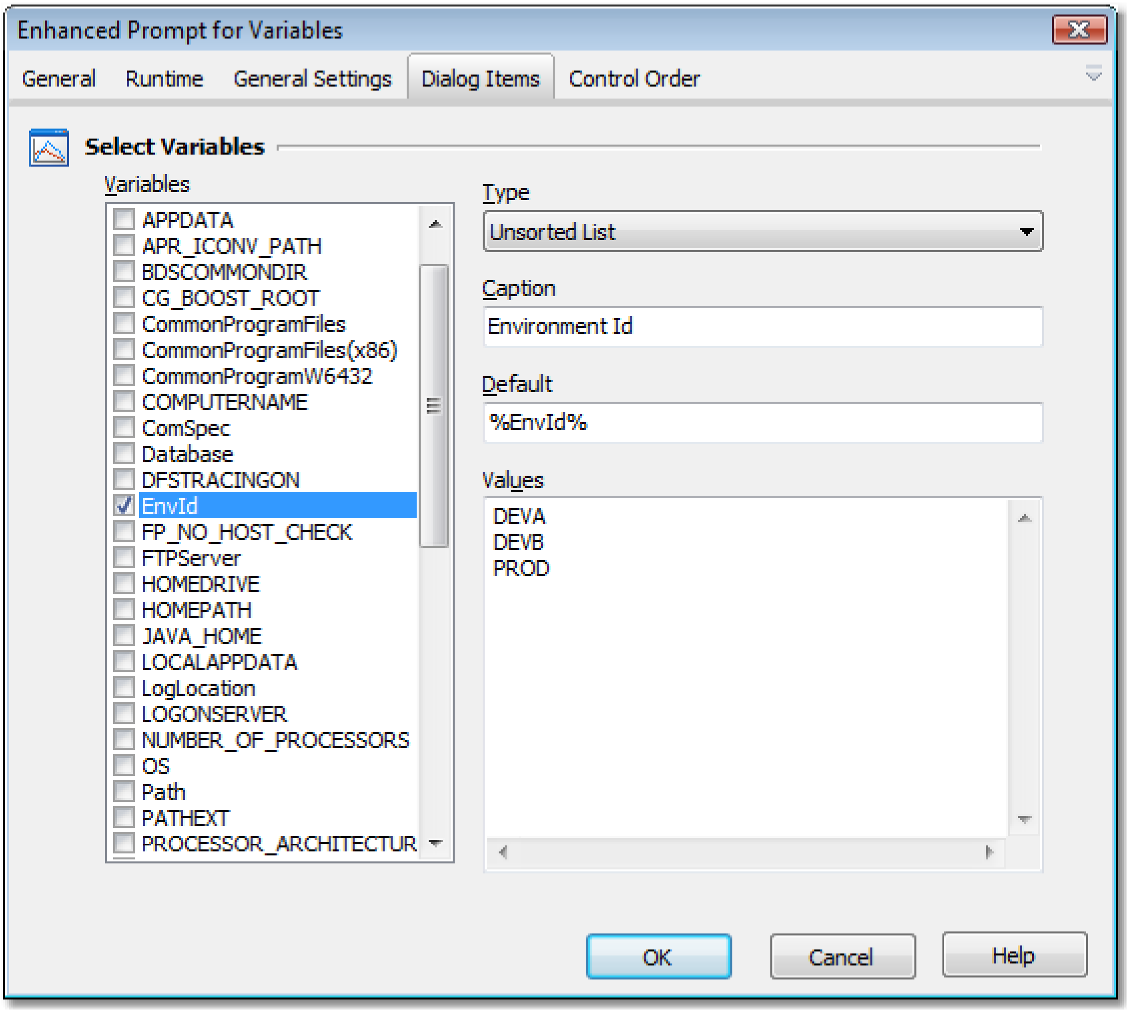
```
[DEVA]
LogLocation=D:\Data\Logs
Database=DEVData\Database01
FTPServer=DEVFTP01

[DEVB]
LogLocation=J:\Public\Data\Logs
Database=DEV2Data\Database01
FTPServer=DEV2FTP01

[PROD]
LogLocation=L:\Public\Logs
Database=ProdData\Data01
FTPServer=ProdFTP01
```

Variables

Create a "EnvId" variable. Make it persistent, so that each time the project is run it can default to the environment as the previous run.

Prompt

Create a "Enhanced prompt for variables" action. Here you will give the user the choice of which environment to use. By using the "unsorted list" type with the current value as the default, a drop down list is shown with the current value already selected.



☑ EnvId

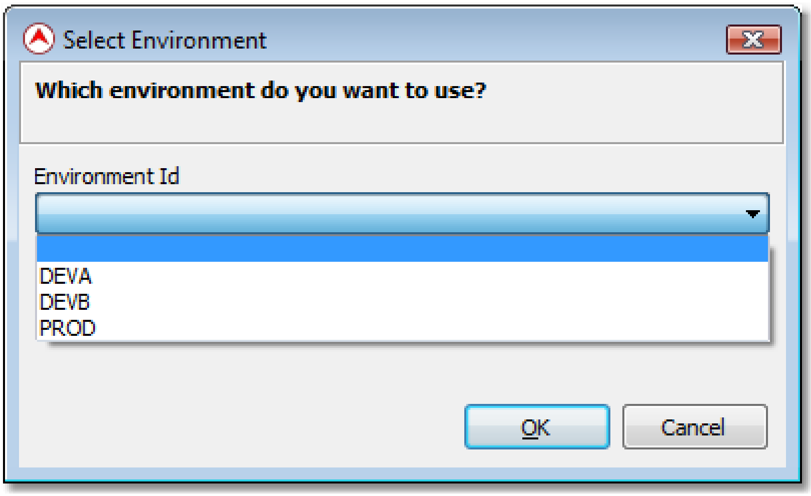
Type: Unsorted List

Caption: Environment Id

Default: %EnvId%

Values:
DEVA
DEVB
PROD

At runtime, this will look as follows:



Which environment do you want to use?

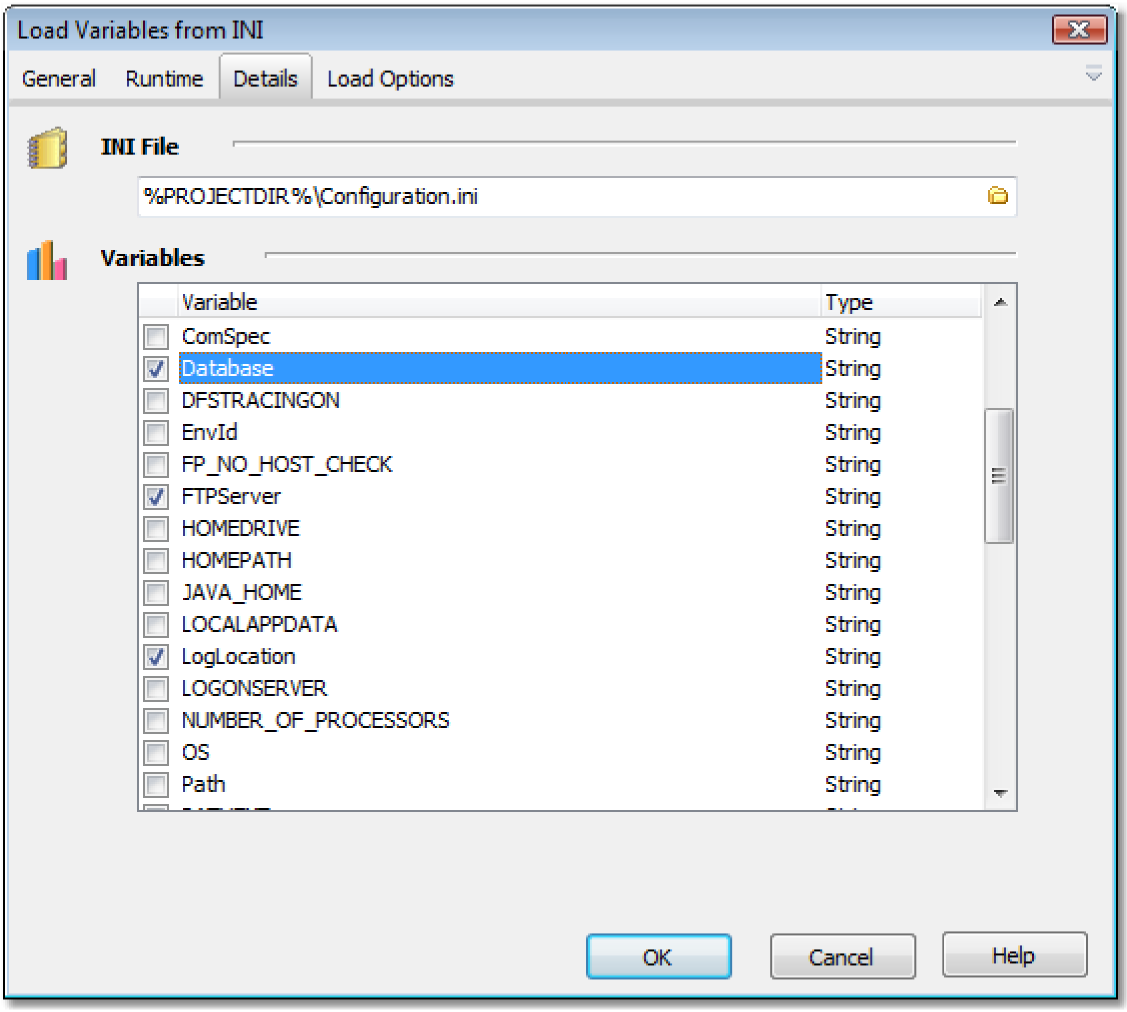
DEVA
DEVB
PROD

Load INI file

Next, we need to load the settings for the chosen environment.

Use the *Load Variables from INI* action with these settings:

- INI file: %PROJECTDIR%\Configuration.ini
- Variables: Database, FTPServer, LogLocation
- INI file section name: %EnvId%
- If variable not in INI file: Fail Action

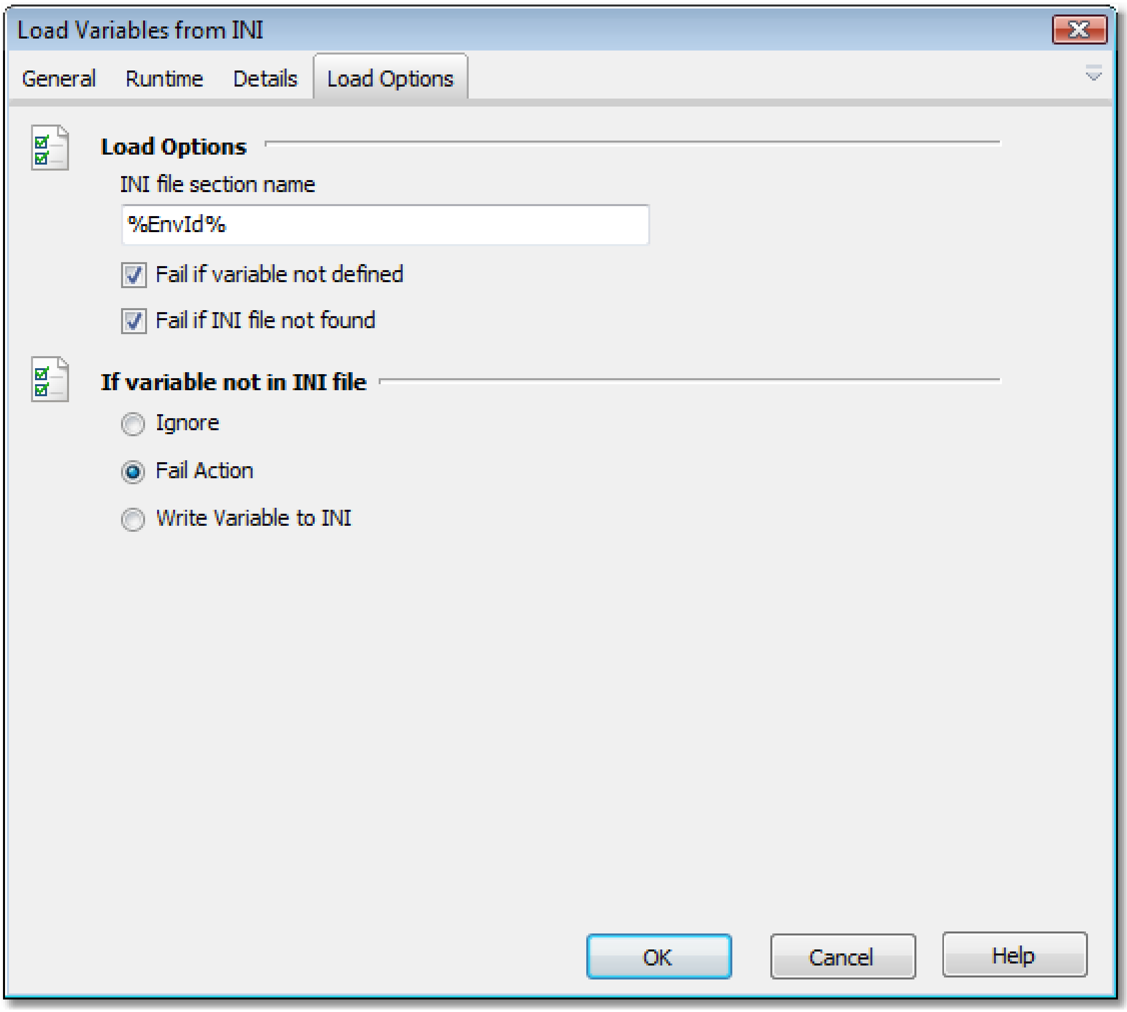


INI File

%PROJECTDIR%\Configuration.ini

Variables

	Variable	Type
□	ComSpec	String
☑	Database	String
□	DFSTRACINGON	String
□	EnvId	String
□	FP_NO_HOST_CHECK	String
☑	FTPServer	String
□	HOMEDRIVE	String
□	HOMEPATH	String
□	JAVA_HOME	String
□	LOCALAPPDATA	String
☑	LogLocation	String
□	LOGONSERVER	String
□	NUMBER_OF_PROCESSORS	String
□	OS	String
□	Path	String



Load Variables from INI

Load Options

INI file section name: %EnvId%

☑ Fail if variable not defined

☑ Fail if INI file not found

If variable not in INI file

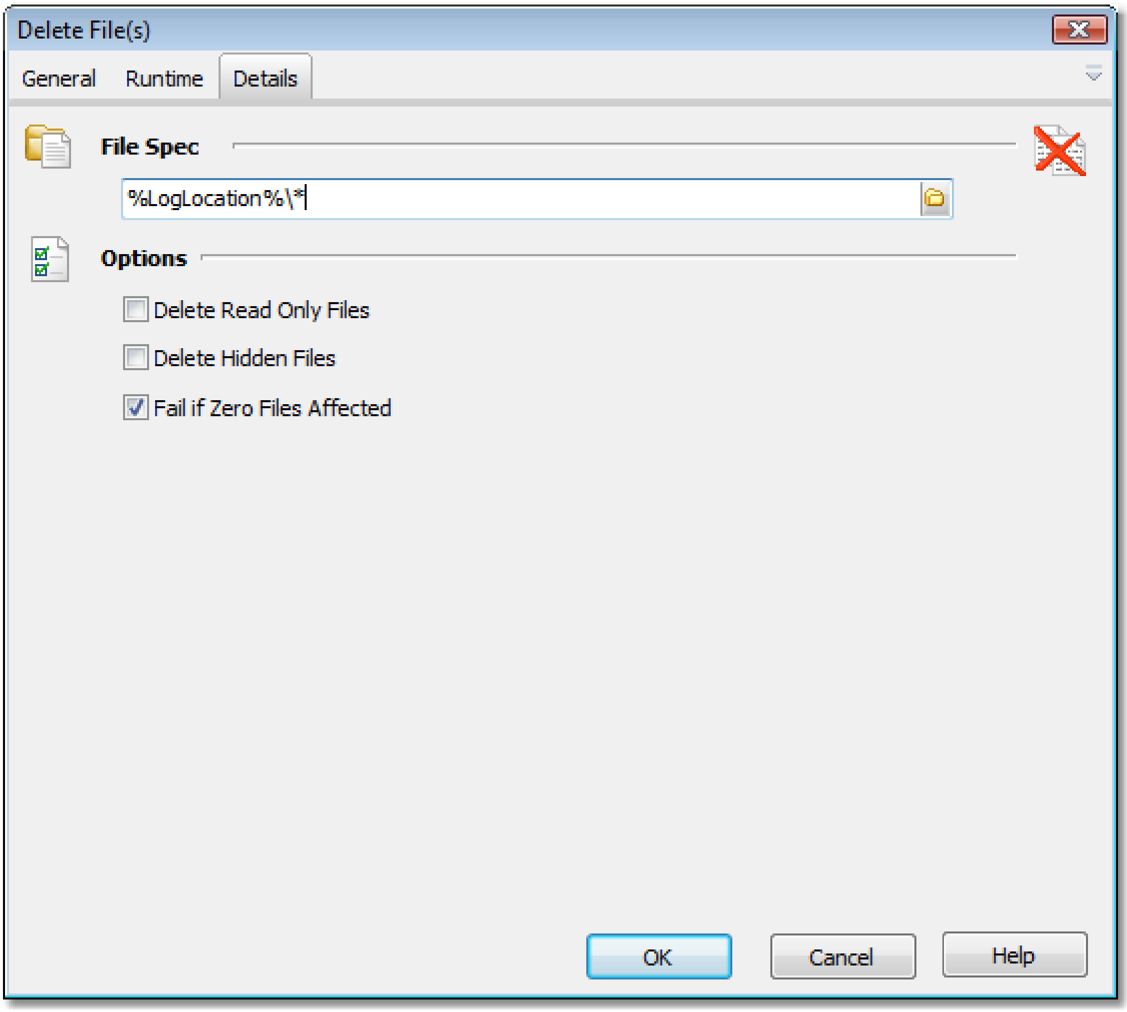
○ Ignore

● Fail Action

○ Write Variable to INI

Use the variables

Now construct your project, using these variables everywhere:



Delete File(s)

File Spec

%LogLocation%*

Options

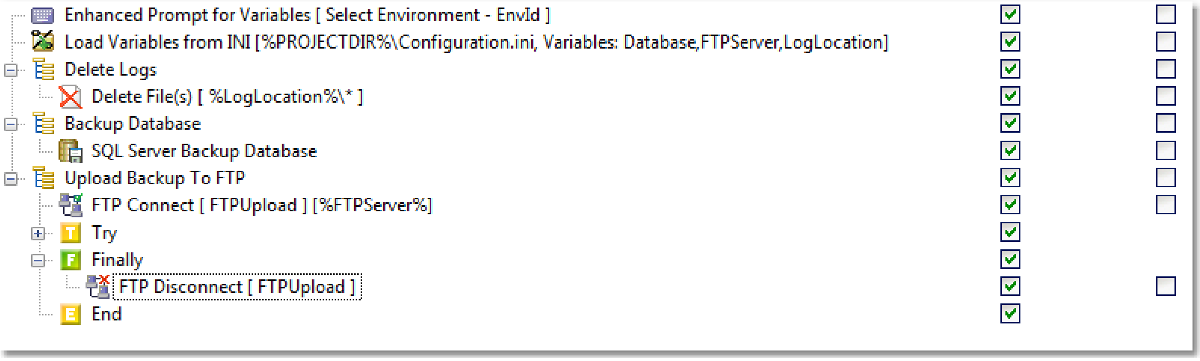
□ Delete Read Only Files

□ Delete Hidden Files

☑ Fail if Zero Files Affected

Conclusion

The overall result looks like this:



```
Enhanced Prompt for Variables [ Select Environment - EnvId ]                                              ☑  □
Load Variables from INI [%PROJECTDIR%\Configuration.ini, Variables: Database,FTPServer,LogLocation]       ☑  □
Delete Logs                                                                                               ☑  □
    Delete File(s) [ %LogLocation%\* ]                                                                    ☑  □
Backup Database                                                                                           ☑  □
    SQL Server Backup Database                                                                            ☑  □
Upload Backup To FTP                                                                                      ☑  □
    FTP Connect [ FTPUpload ] [%FTPServer%]                                                               ☑  □
    Try                                                                                                   ☑
    Finally                                                                                               ☑
        FTP Disconnect [ FTPUpload ]                                                                      ☑  □
    End                                                                                                   ☑
```

You now have a single project which is capable of running in different environments. All the core logic - loading variables, deleting files, performing database backup, uploading files to FTP server - is stored in the one place. This is much better than having a separate Automise project for each environment. In that situation, if you found a problem in one project, you would have to fix it in every project individually, making your project more error prone and labour intensive.

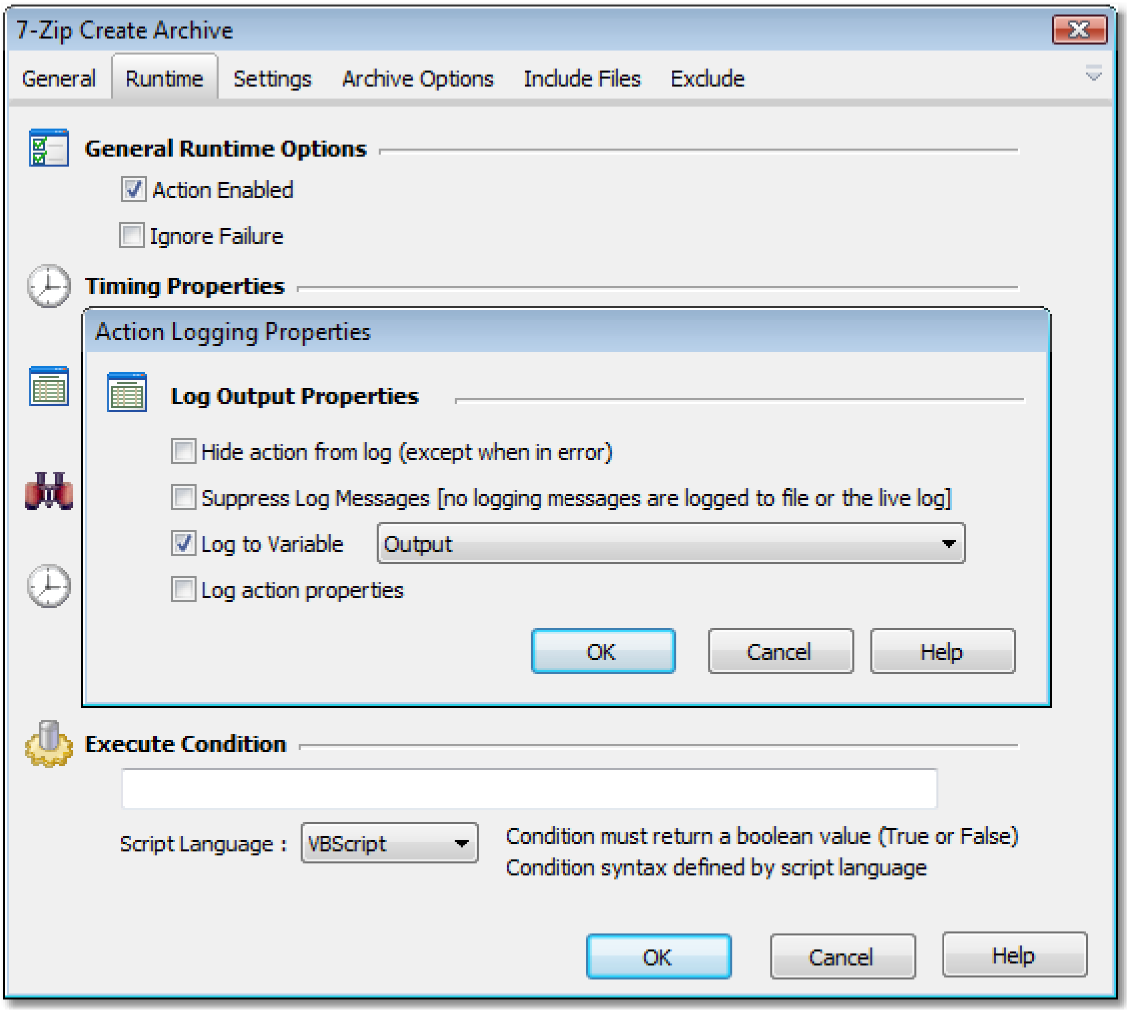
8.4 Analysing Output

Action output monitors make it easy to react to the presence of a word, such as "error" in the output from an action. But what if you want to abort the build if there are more than five errors, for example, or you want to process each line of the output somehow? Here are some solutions to those two problems.

Counting warnings

The key is to log the output of the action to a variable, then analyse the contents of the variable. You'll need two variables, "Output" and "Count"

First, on the **Runtime** tab of the relevant action, go to "Logging Properties" and select "Log to Variable".

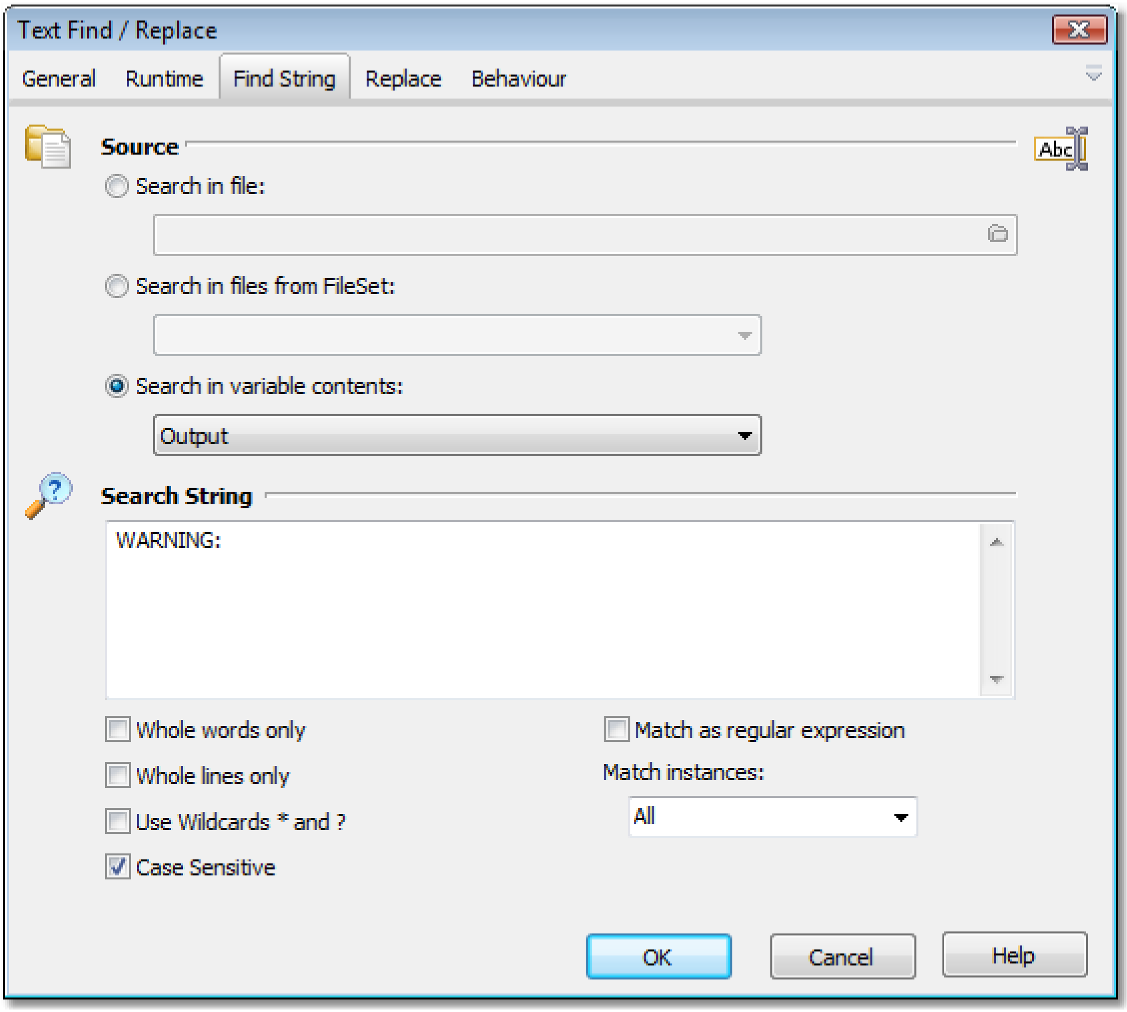


☑ Action Enabled
□ Ignore Failure

□ Hide action from log (except when in error)
□ Suppress Log Messages [no logging messages are logged to file or the live log]
☑ Log to Variable: Output
□ Log action properties

Script Language : VBScript
Condition must return a boolean value (True or False)
Condition syntax defined by script language

Next, use a "Text Find / Replace" action to count the number of times the string appears:



Text Find / Replace

Source

○ Search in file:

○ Search in files from FileSet:

● Search in variable contents:

Output

Search String

WARNING:

□ Whole words only

□ Whole lines only

□ Use Wildcards * and ?

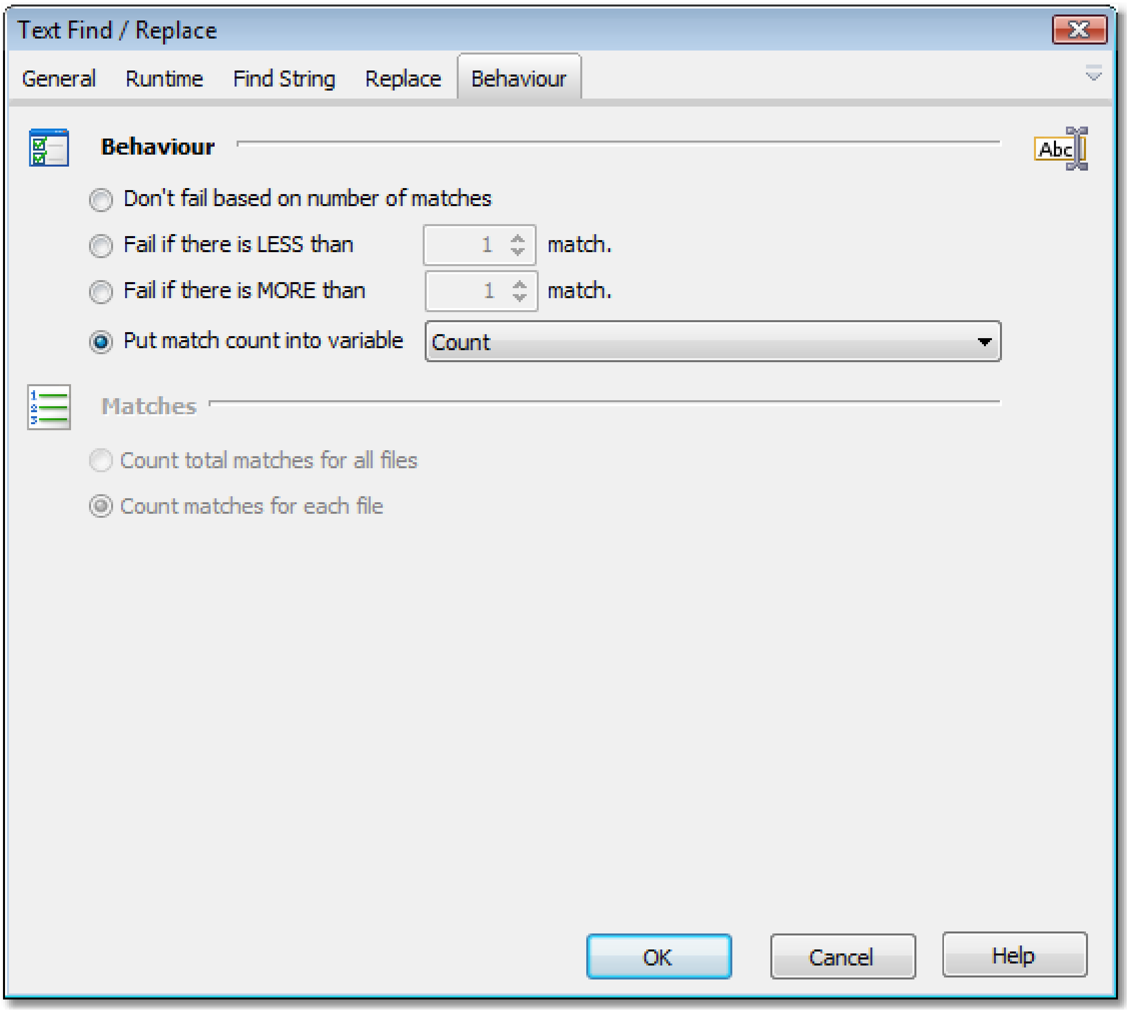
☑ Case Sensitive

□ Match as regular expression

Match instances:

All

On the **Behaviour** tab, set the variable to hold the number of matches:



Text Find / Replace

Behaviour

○ Don't fail based on number of matches

○ Fail if there is LESS than 1 match.

○ Fail if there is MORE than 1 match.

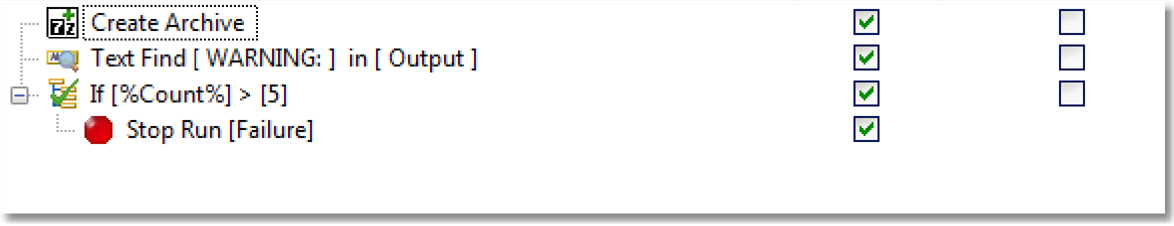
● Put match count into variable: Count

Matches

○ Count total matches for all files

● Count matches for each file

Now you're all set to use the variable however you like:



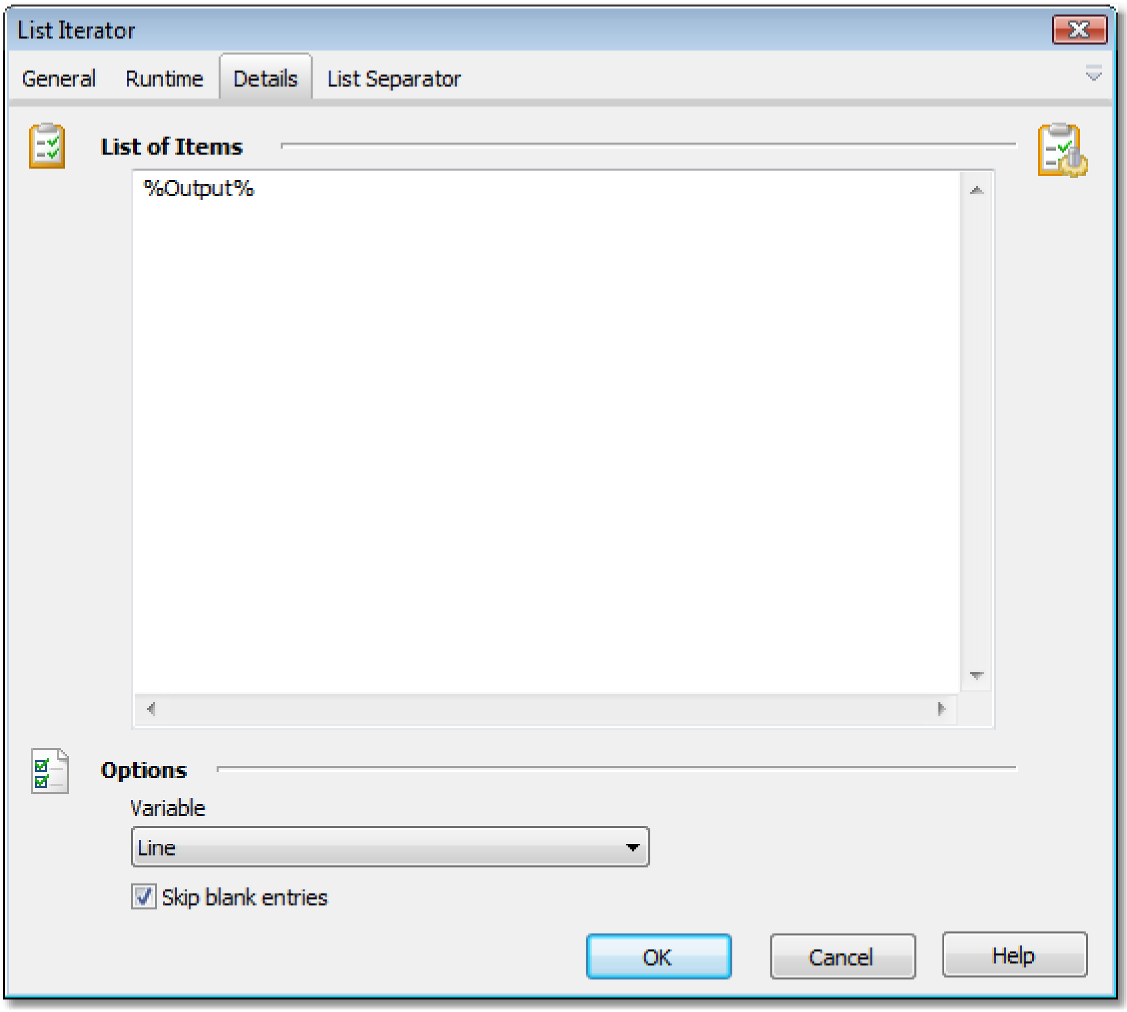
Create Archive	☑	□
Text Find [WARNING:] in [Output]	☑	□
If [%Count%] > [5]	☑	□
Stop Run [Failure]	☑	

Processing a log line by line

Let's say the output from some external program is very verbose, and all you want is lines that contain "Image: " followed by a filename.

Start by logging the output of action to the Output variable. You will also need a variable to hold each line of output. Call it "Line".

Next, use a List Iterator action. Use %Output% as the "List of Items" value. At runtime, it will be expanded to the full value of the log. Don't worry about the size, Automise has a very large upper limit on variable size.



List of Items

%Output%

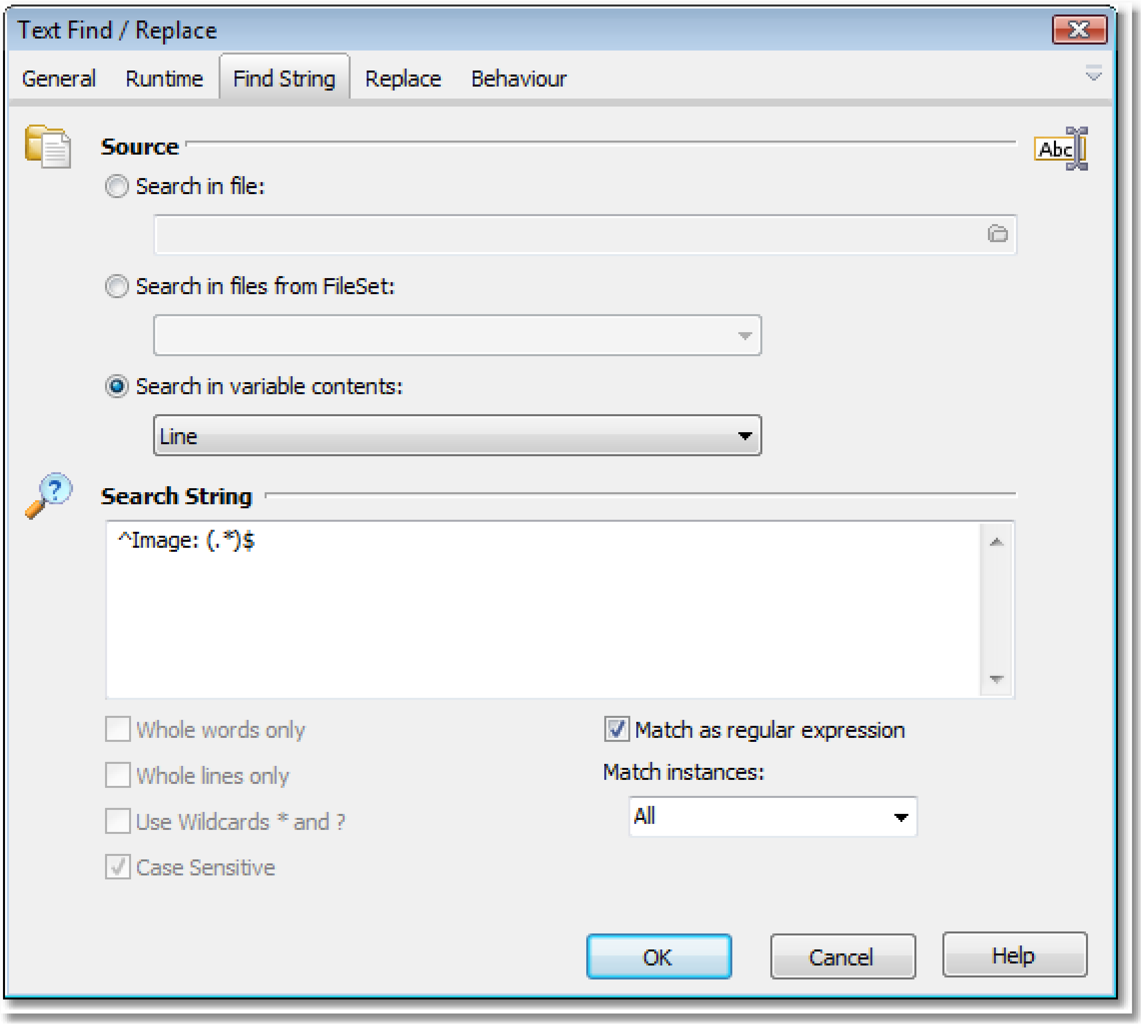
Options

Variable: Line

☑ Skip blank entries

Leave the "List Separator" settings as the default: a carriage return/line feed.

Now for each line, we use a Text Find / Replace action to reduce a line containing the key string down to just the image filename itself:



Source

○ Search in file:

○ Search in files from FileSet:

● Search in variable contents:

Line

Search String

```
^Image: (.*)$
```

□ Whole words only

□ Whole lines only

□ Use Wildcards * and ?

☑ Case Sensitive

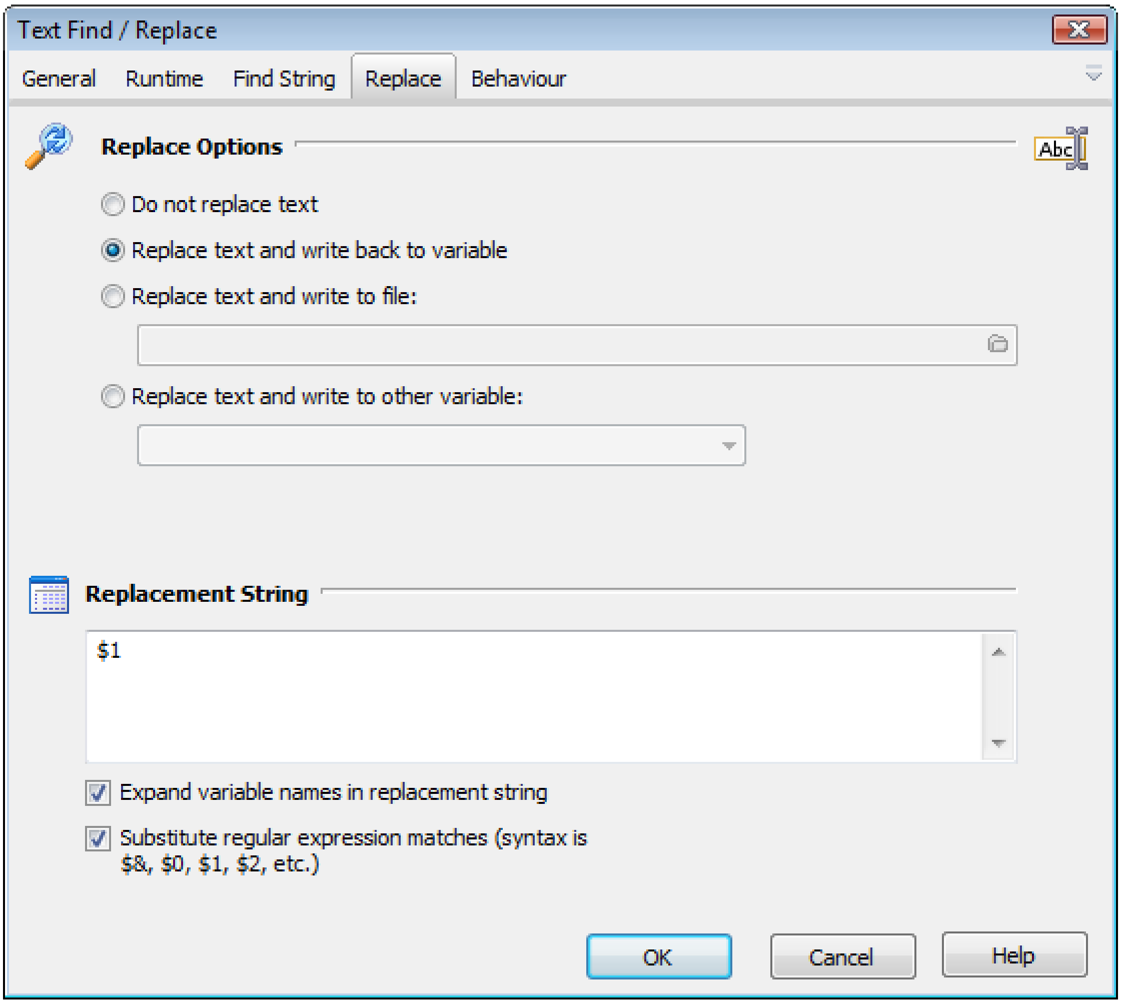
☑ Match as regular expression

Match instances:

All

This regular expression means "*beginning at the start of the line, match the word Image, a colon, a space, then store the whole rest of the line as subexpression 1*".

On the **Replace** tab, we write that subexpression back to the same variable:



Replace Options

○ Do not replace text

● Replace text and write back to variable

○ Replace text and write to file:

○ Replace text and write to other variable:

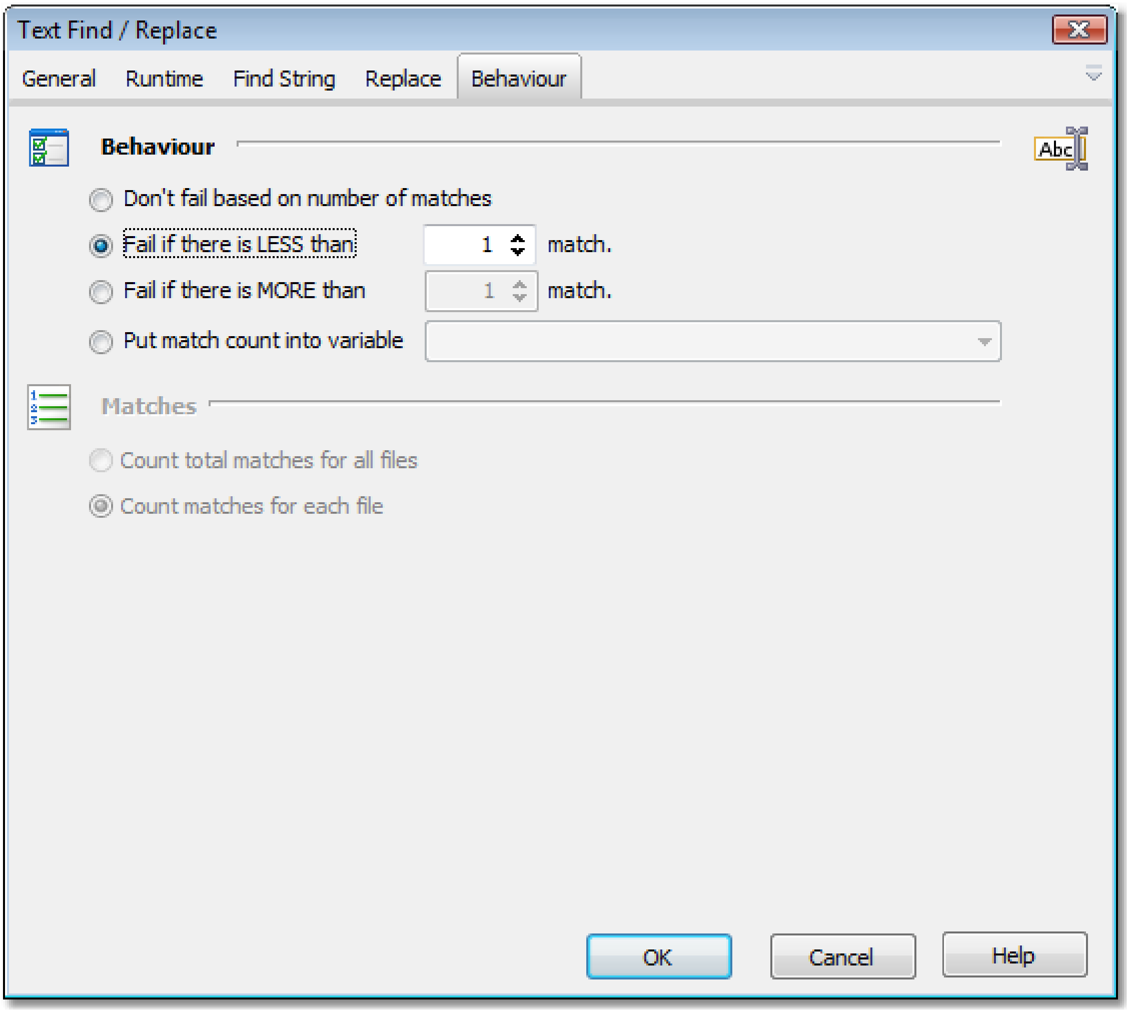
Replacement String

```
$1
```

☑ Expand variable names in replacement string

☑ Substitute regular expression matches (syntax is $&, $0, $1, $2, etc.)

Finally, on the **Behaviour** tab, we set the action to fail if it didn't match. We do this because we want to do further processing on a line that matches.



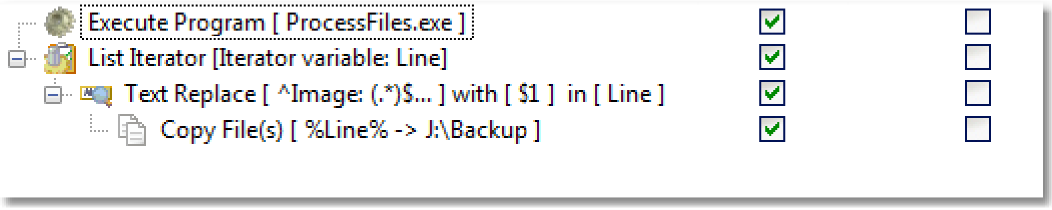
Behaviour

○ Don't fail based on number of matches
● Fail if there is LESS than 1 match.
○ Fail if there is MORE than 1 match.
○ Put match count into variable

Matches

○ Count total matches for all files
● Count matches for each file

Now, we can add whatever processing we like. The "line" variable at this point contains just the name of the image found in the output. Note that we set the "Text Replace" action to ignore failure. The loop should carry on for each line that doesn't have the text we're looking for.



Execute Program [ProcessFiles.exe]	☑	□
List Iterator [Iterator variable: Line]	☑	□
Text Replace [^Image: (.*)$...] with [$1] in [Line]	☑	□
Copy File(s) [%Line% -> J:\Backup]	☑	□

There we have it!

Summary

1. The program runs, logging its output to a variable called "Output"
2. The list iterator cycles over that output, placing each line in a variable called "Line"
3. The Text Replace action then reduces that line down to just the image name, or fails if it's not an image line.
4. If the text is found, the file is then copied somewhere.

8.5 Documenting your Project

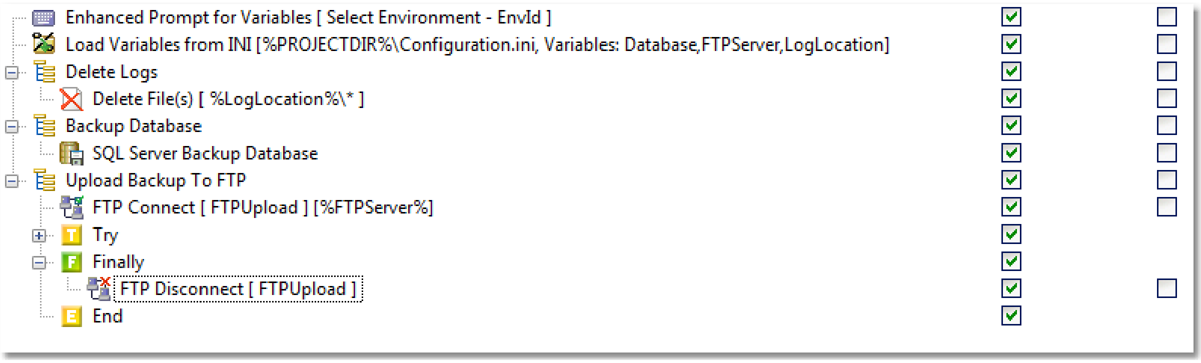
As your Automise project gets larger and more complex, you should document it, to help you and others maintain it in the future. Here are some of the ways Automise lets you do that.

Topics covered:

- Action groups
- Comment actions
- Action comments
- Variable comments
- Project information notes

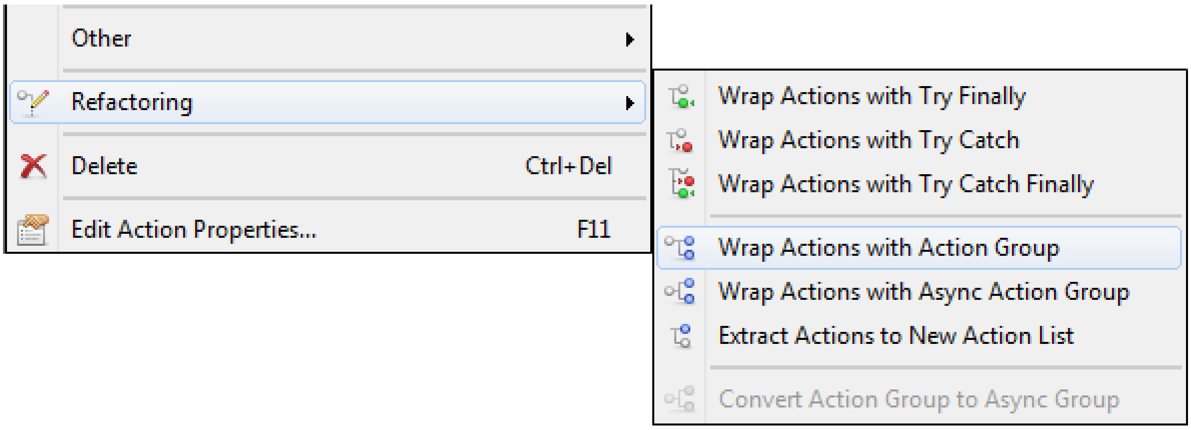
Actions

Action groups are the best way to describe the purpose of a sequence of actions. It's clear to the reader exactly which actions are covered by the comment, and you can collapse the action group and still see roughly what it's doing.

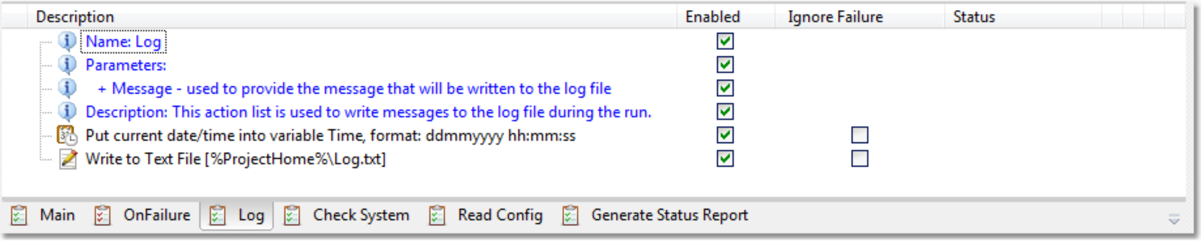


```
Enhanced Prompt for Variables [ Select Environment - EnvId ]                                          ☑  □
Load Variables from INI [%PROJECTDIR%\Configuration.ini, Variables: Database,FTPServer,LogLocation]   ☑  □
⊟ Delete Logs                                                                                         ☑  □
    Delete File(s) [ %LogLocation%\* ]                                                                ☑  □
⊟ Backup Database                                                                                     ☑  □
    SQL Server Backup Database                                                                        ☑  □
⊟ Upload Backup To FTP                                                                                ☑  □
    FTP Connect [ FTPUpload ] [%FTPServer%]                                                           ☑  □
  ⊞ Try                                                                                               ☑
  ⊟ Finally                                                                                           ☑
      FTP Disconnect [ FTPUpload ]                                                                    ☑  □
    End                                                                                               ☑
```

To quickly create an action group, select a group of actions, right click, and choose "Refactoring > Wrap Actions with Action Group". Then press F2 to give the action group a meaningful name.

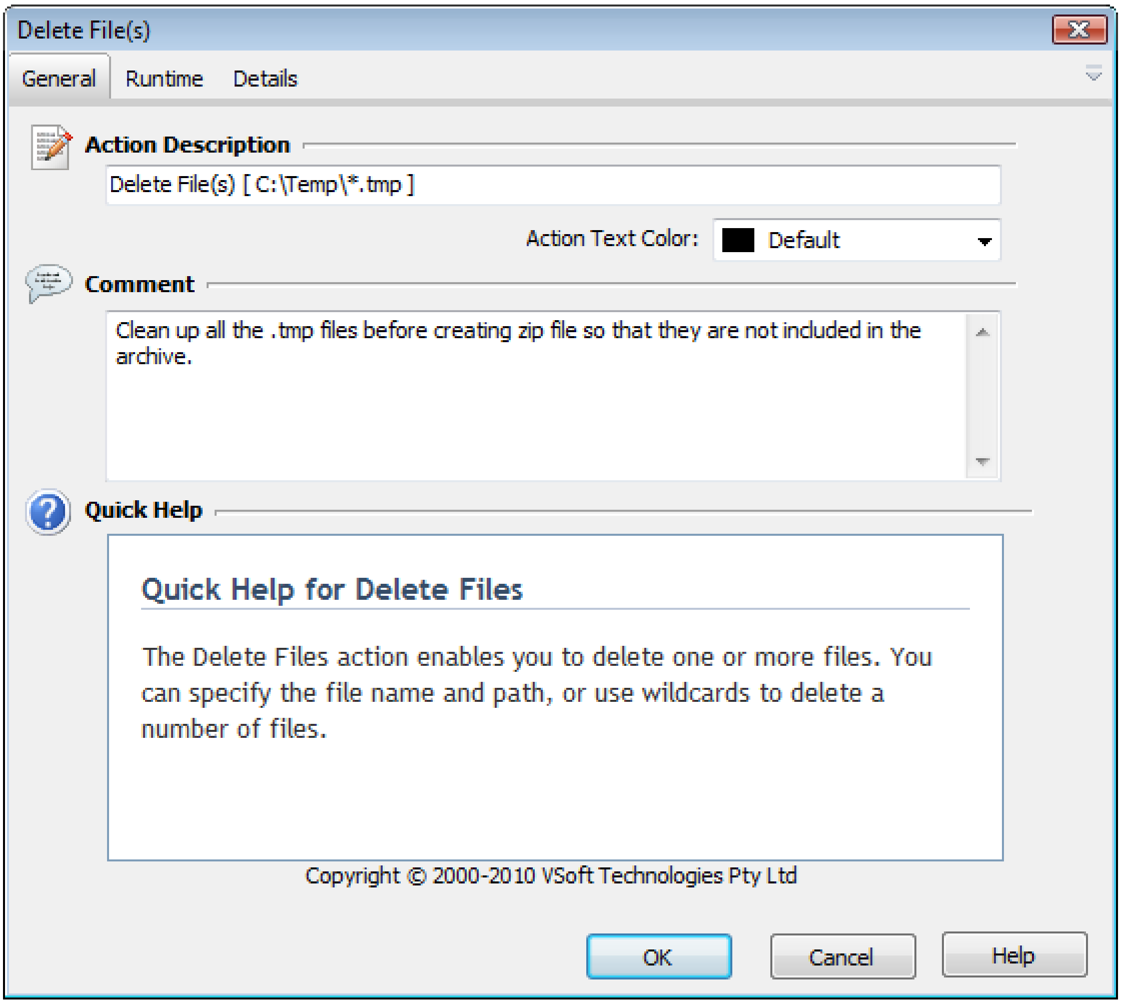

Other
Refactoring
Delete Ctrl+Del
Edit Action Properties... F11
Wrap Actions with Try Finally
Wrap Actions with Try Catch
Wrap Actions with Try Catch Finally
Wrap Actions with Action Group
Wrap Actions with Async Action Group
Extract Actions to New Action List
Convert Action Group to Async Group

Comment actions are useful for providing further information about a part of your project. They draw the maintainer's attention to some important information about a part of the project. It's also good practice to place a comment at the start of each action list (apart from Main), describing the purpose of the action list and its parameters.



Description	Enabled	Ignore Failure	Status
Name: Log	☑		
Parameters:	☑		
+ Message - used to provide the message that will be written to the log file	☑		
Description: This action list is used to write messages to the log file during the run.	☑		
Put current date/time into variable Time, format: ddmmyyyy hh:mm:ss	☑	□	
Write to Text File [%ProjectHome%\Log.txt]	☑	□	

Finally, actions themselves can have comments. These are useful for explaining the rationale behind a specific action or option, or to warn the maintainer about any important side effects.



Action Description

Delete File(s) [C:\Temp\*.tmp]

Action Text Color: ■ Default

Comment

Clean up all the .tmp files before creating zip file so that they are not included in the archive.

Quick Help

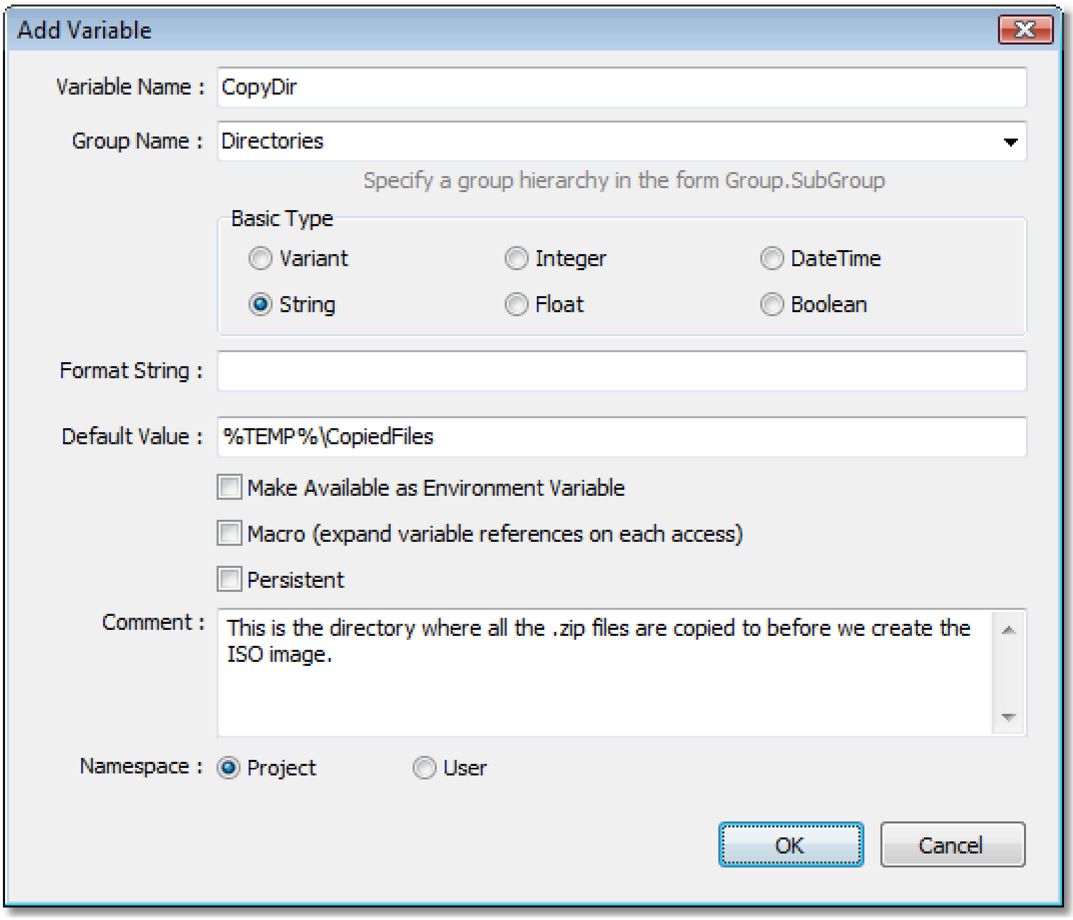
Quick Help for Delete Files

The Delete Files action enables you to delete one or more files. You can specify the file name and path, or use wildcards to delete a number of files.

Copyright © 2000-2010 VSoft Technologies Pty Ltd

Variables

While system variables come with predefined comments, it's up to your to document your own variables using the Comment field. Good comments explain what a variable is used for, where its value comes from, and what a typical value looks like.



Add Variable

Variable Name : CopyDir

Group Name : Directories

Specify a group hierarchy in the form Group.SubGroup

Basic Type

○ Variant ○ Integer ○ DateTime

● String ○ Float ○ Boolean

Format String :

Default Value : %TEMP%\CopiedFiles

□ Make Available as Environment Variable

□ Macro (expand variable references on each access)

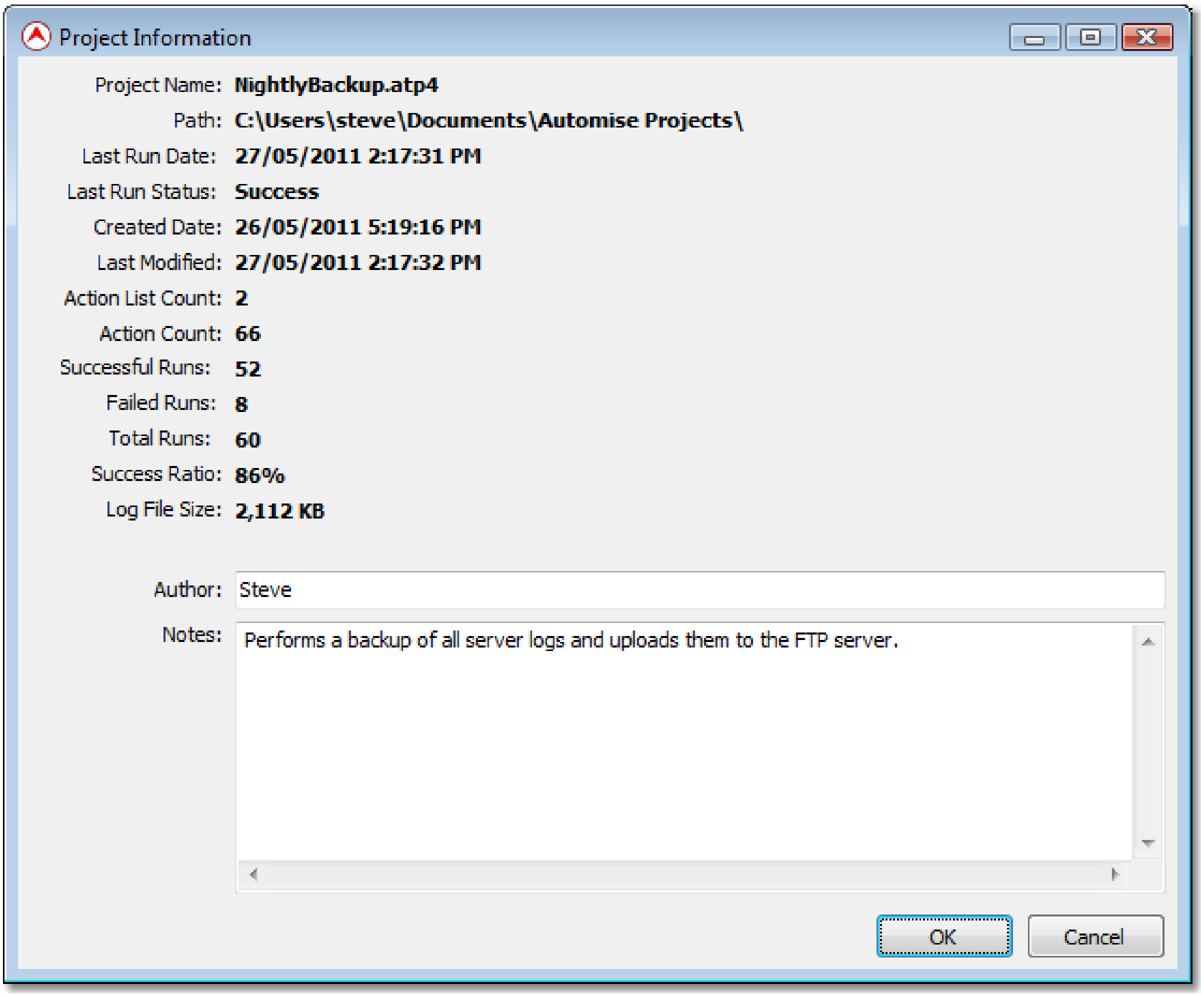
□ Persistent

Comment : This is the directory where all the .zip files are copied to before we create the ISO image.

Namespace : ● Project ○ User

Project

The project itself has notes that can be edited on the Project Information page. This field is by default shown on the Welcome Page. You can put a brief summary of the purpose of the project, and possibly some history or other notes.



Project Name:	NightlyBackup.atp4
Path:	C:\Users\steve\Documents\Automise Projects\
Last Run Date:	27/05/2011 2:17:31 PM
Last Run Status:	Success
Created Date:	26/05/2011 5:19:16 PM
Last Modified:	27/05/2011 2:17:32 PM
Action List Count:	2
Action Count:	66
Successful Runs:	52
Failed Runs:	8
Total Runs:	60
Success Ratio:	86%
Log File Size:	2,112 KB

Author: Steve

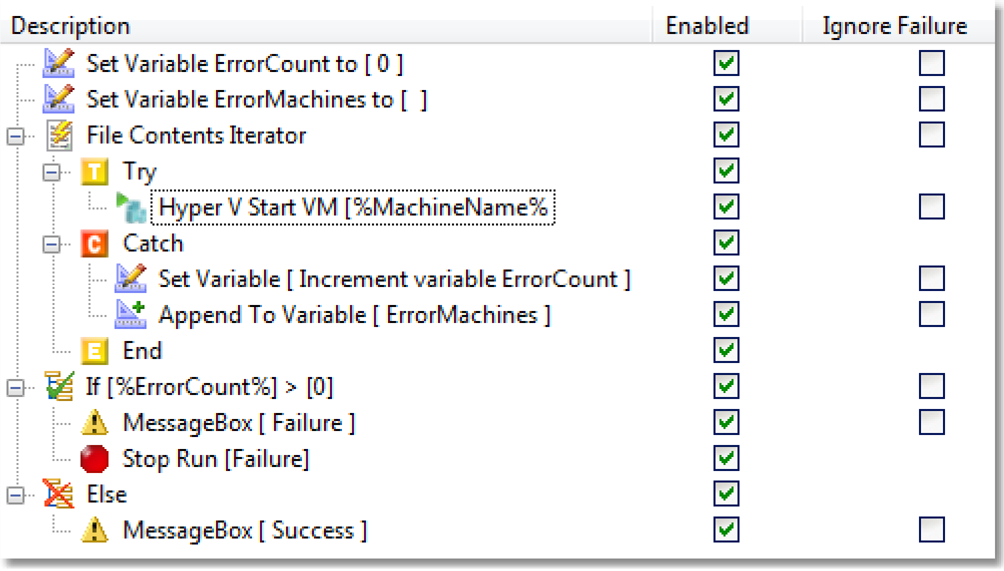
Notes: Performs a backup of all server logs and uploads them to the FTP server.

8.6 Counting Errors with Try/Catch Blocks

One good use of Try/Catch blocks is to record information about errors, in order to generate a report later in the project. There are several advantages to doing this:

- The project doesn't abort at the first error, so you get more information if several steps fail.
- You can recover from minor errors.
- You can treat different errors differently, for example, by emailing different responsible people.
- You can record you own statistics and logs.

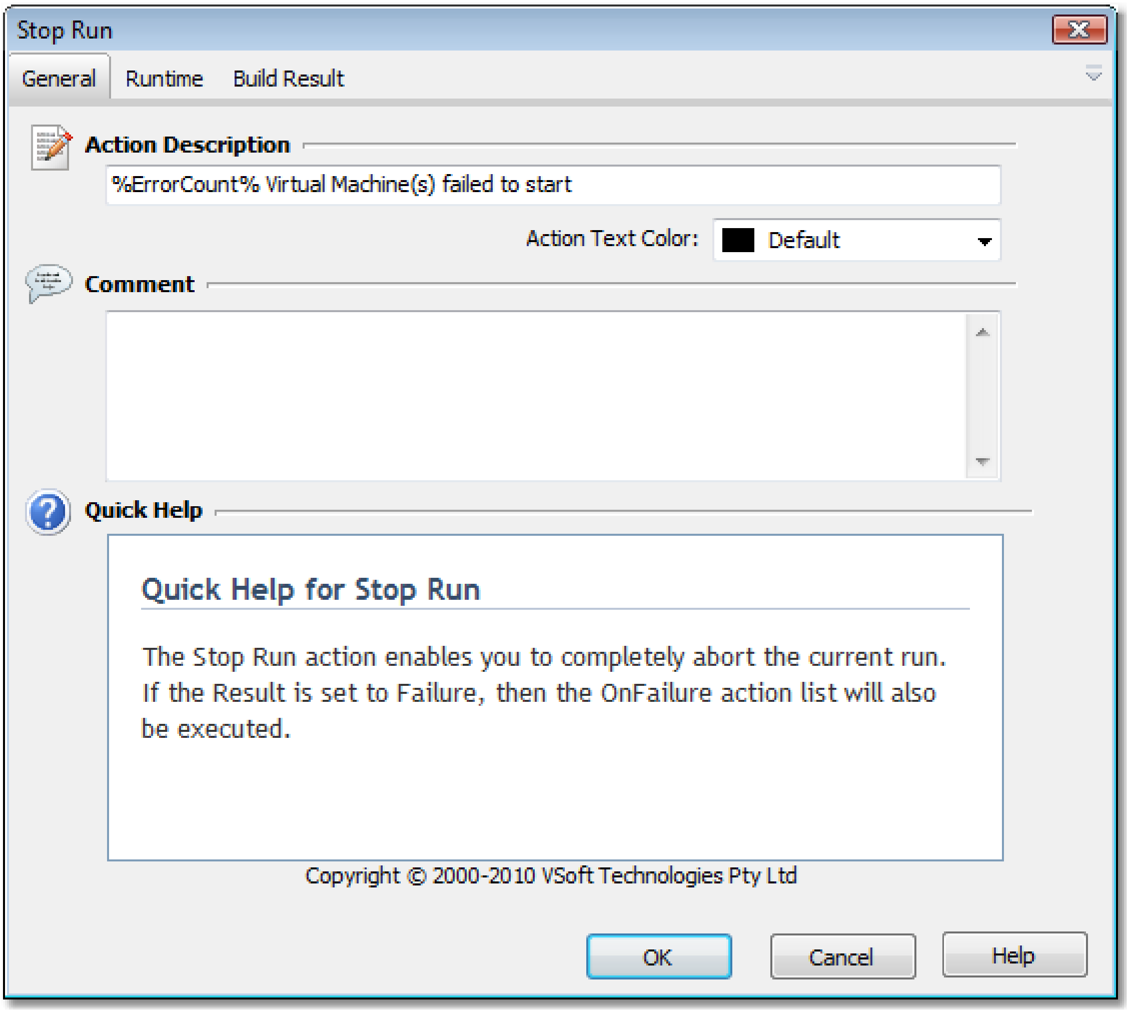
Here's a simple example which starts all the Virtual Machines listed in a text file. The variable ErrorCount stores the number of machines that failed to start, while ErrorMachines creates a list of machines that failed to start.



Description	Enabled	Ignore Failure
Set Variable ErrorCount to [0]	☑	□
Set Variable ErrorMachines to [ ]	☑	□
File Contents Iterator	☑	□
Try	☑	
Hyper V Start VM [%MachineName%	☑	□
Catch	☑	
Set Variable [Increment variable ErrorCount]	☑	□
Append To Variable [ErrorMachines]	☑	□
End	☑	
If [%ErrorCount%] > [0]	☑	□
MessageBox [Failure]	☑	□
Stop Run [Failure]	☑	
Else	☑	
MessageBox [Success]	☑	□

The steps are as follows:
1. Initialise the two variables.
2. Iterate over the contents of the file.
3. Use a Try action to wrap around the Hyper V Start Machine action. If the virtual machine starts successfully then the Catch part is not run.
4. If the virtual machine does not start, the Catch part is run: the ErrorCount variable is incremented, and the ErrorMachines variable is appended to. The project then continues on the next loop of the iterator.
5. Once the project has finished attempting to start all the machines (reached the end of the file), a message will be shown if there was at least one error. We then use a Stop Run action to signal that the project as a whole failed.
6. If there were no errors, a different message is shown. By default, projects terminate with a success code, so we don't need a Stop Run action here.

More ideas:
- Instead of showing a message, you could record the count and list of failed machines to a text file.
- To gain more information about any error, you could use Log To Variable. See the Analysing Output tutorial.
- You can use Try/Catch blocks at a very high level, wrapping calls to Action Lists or even other projects with the Include Project action.
- Set a custom Action Log Title on the Stop Run action to explain why the build is stopping.



Action Description

%ErrorCount% Virtual Machine(s) failed to start

Action Text Color: ■ Default

Comment

Quick Help

Quick Help for Stop Run

The Stop Run action enables you to completely abort the current run. If the Result is set to Failure, then the OnFailure action list will also be executed.

Copyright © 2000-2010 VSoft Technologies Pty Ltd

8.7 Using Output Monitors

Here we use **Output Monitors** to solve the problem of finding the most recent file in a directory. In this example, we want to find the most recently changed file in the c:\projects\source directory. This DOS command lists the files in order from oldest to newest: `dir /b /od /a-d c:\projects\source\*.*`

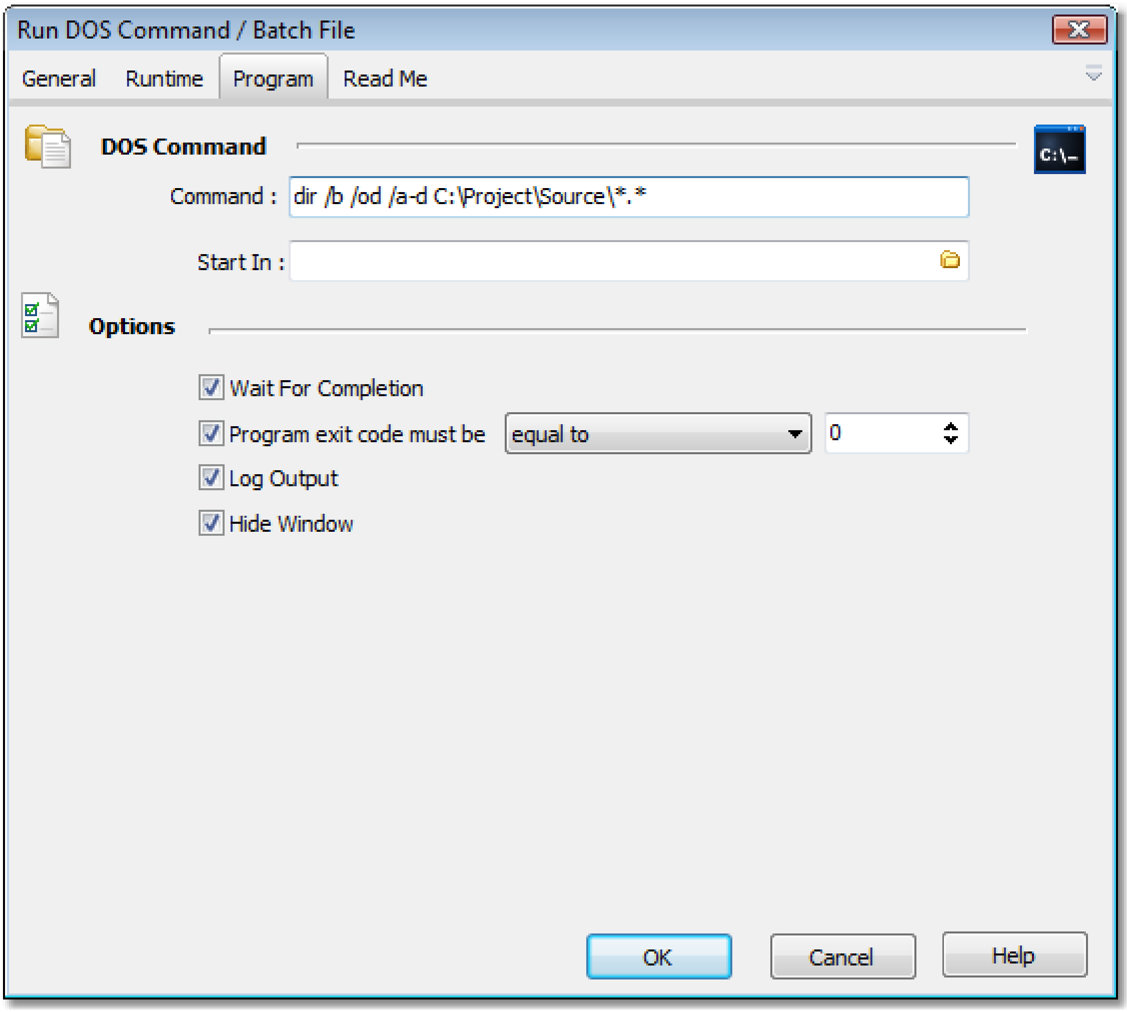
The output is as follows:

```
ExcelDoc1.xlsx
ExcelDoc2.xlsx
SourceDoc3.txt
FileWithNoExt2
WordDoc2.docx
SourceDoc2.txt
WordDoc1.docx
SourceDoc1.txt
FileWithNoExt
```

We thus simply want to retrieve the last line of that output. An Output Monitor do this.

First, create a variable to hold the name of the most recent file. Call it "filename".

Next, create a Run DOS Command / Batch File action.



Command : dir /b /od /a-d C:\Project\Source\*.*

Start In :

☑ Wait For Completion

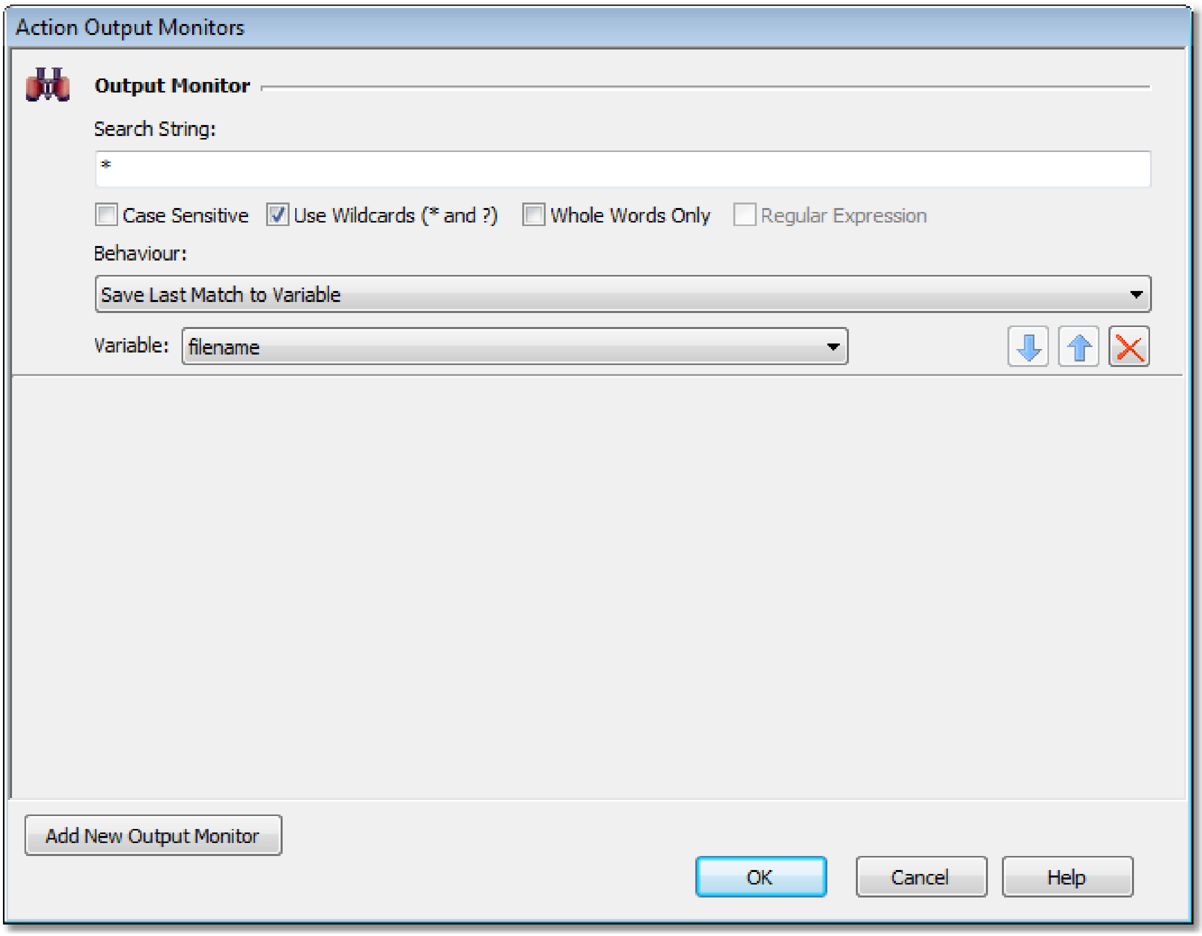
☑ Program exit code must be equal to 0

☑ Log Output

☑ Hide Window

On the **Runtime** tab, click the **Output Monitors** ellipsis button (...).

Click "Add a New Output Monitor" then set the options as shown:


Action Output Monitors

Output Monitor

Search String:

*

□ Case Sensitive ☑ Use Wildcards (* and ?) □ Whole Words Only □ Regular Expression

Behaviour:

Save Last Match to Variable

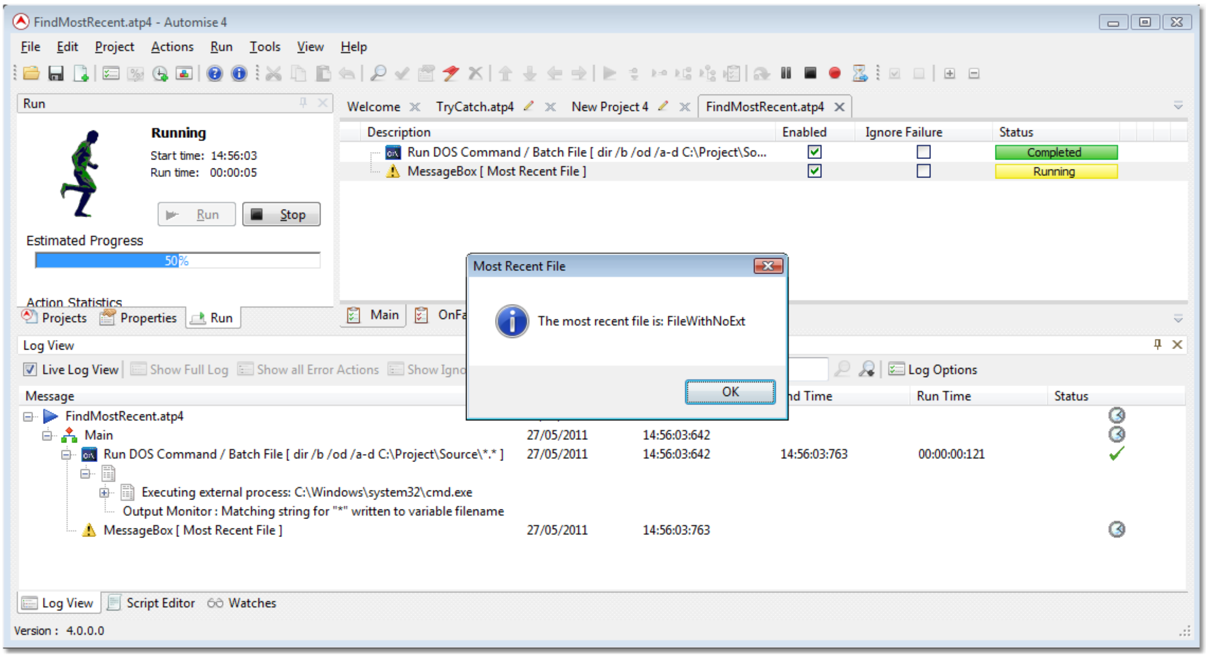
Variable: filename

The **Search String** defines what lines of text are selected. In this case, we don't care what the text is - we just want the last one. So the *, combined with **Use Wildcards** matches any text.

The **Behaviour** lets you choose options like failing if the text doesn't match. The **Save Last Match to Variable** option is obviously the appropriate one here.

Lastly, choose the **Variable** you created - Filename.

The final result looks like this:



Running
Start time: 14:56:03
Run time: 00:00:05

Description	Enabled	Ignore Failure	Status
Run DOS Command / Batch File [dir /b /od /a-d C:\Project\So...	☑	□	Completed
MessageBox [Most Recent File]	☑	□	Running

Most Recent File

The most recent file is: FileWithNoExt

OK

Message			nd Time	Run Time	Status
FindMostRecent.atp4					
Main	27/05/2011	14:56:03:642			
Run DOS Command / Batch File [dir /b /od /a-d C:\Project\Source\*.*]	27/05/2011	14:56:03:642	14:56:03:763	00:00:00:121	✓
Executing external process: C:\Windows\system32\cmd.exe					
Output Monitor : Matching string for "*" written to variable filename					
MessageBox [Most Recent File]	27/05/2011	14:56:03:763			

Version : 4.0.0.0

9 Reference

9.1 Regular Expression Reference

Introduction

Regular Expressions are a widely-used method of specifying patterns of text to search for. Special meta characters allow you to specify, for instance, that a particular string you are looking for occurs at the beginning or end of a line, or contains n recurrences of a certain character.

Regular expressions look ugly for novices, but are really very simple, handy and powerful.

Regular expressions can be used in some actions and can also be used by plugin developers. This reference documents the particular regular expression library used in Automise: TRegExpr, see http://www.regexpstudio.com/

Simple Matches

Any single character matches itself, unless it is a meta character with a special meaning described below.

A series of characters matches that series of characters in the target string, so the pattern "bluh" would match "bluh" in the target string. Quite simple, eh ?

You can cause characters that normally function as meta characters or escape sequences to be interpreted literally by 'escaping' them by preceding them with a backslash "\", for instance: meta character "^" match beginning of string, but "\^" match character "^", "\\" match "\" and so on.

Examples:

foobar	matches string 'foobar'

```
  \^FooBarPtr     matches '^FooBarPtr'
```

Escape Sequences

Characters may be specified using a escape sequences syntax much like that used in C and Perl: "\n" matches a new line, "\t" a tab, etc. More generally, \xnn, where nn is a string of hexadecimal digits, matches the character whose ASCII value is nn. If You need wide (Unicode) character code, You can use '\x{nnnn}', where 'nnnn' - one or more hexadecimal digits.

```
  \xnn     char with hex code nn
  \x{nnnn} char with hex code nnnn (one byte for plain text and two bytes for Unicode)
  \t       tab (HT/TAB), same as \x09
  \n       newline (NL), same as \x0a
  \r       car.return (CR), same as \x0d
  \f       form feed (FF), same as \x0c
  \a       alarm (bell) (BEL), same as \x07
  \e       escape (ESC), same as \x1b
```

Examples:

```
  foo\x20bar   matches 'foo bar' (note space in the middle)
  \tfoobar     matches 'foobar' predefined by tab
```

Character Classes

You can specify a character class, by enclosing a list of characters in [], which will match any one character from the list.

If the first character after the "[" is "^", the class matches any character not in the list.

Examples:

```
  foob[aeiou]r   finds strings 'foobar', 'foober' etc. but not 'foobbr', 'foobcr' etc.

  foob[^aeiou]r  find strings 'foobbr', 'foobcr' etc. but not 'foobar', 'foober' etc.
```

Within a list, the "-" character is used to specify a range, so that a-z represents all characters between "a" and "z", inclusive.

If You want "-" itself to be a member of a class, put it at the start or end of the list, or escape it with a backslash. If You want ']' you may place it at the start of list or escape it with a backslash.

Examples:

```
  [-az]      matches 'a', 'z' and '-'

  [az-]      matches 'a', 'z' and '-'
  [a\-z]     matches 'a', 'z' and '-'
  [a-z]      matches all twenty six small characters from 'a' to 'z'
  [\n-\x0D]  matches any of #10,#11,#12,#13.
  [\d-t]     matches any digit, '-' or 't'.
  []-a]      matches any char from ']'..'a'.
```

Meta Characters

Meta characters are special characters which are the essence of Regular Expressions. There are different types of meta characters, described below.

Meta characters - line separators

^	start of line
$	end of line
\A	start of text
\Z	end of text
.	any character in line

Examples:

^foobar	matches string 'foobar' only if it's at the beginning of line
foobar$	matches string 'foobar' only if it's at the end of line
^foobar$	matches string 'foobar' only if it's the only string in line
foob.r	matches strings like 'foobar', 'foobbr', 'foob1r' and so on

The "^" meta character by default is only guaranteed to match at the beginning of the input string/text, the "$" meta character only at the end. Embedded line separators will not be matched by "^" or "$".
You may, however, wish to treat a string as a multi-line buffer, such that the "^" will match after any line separator within the string, and "$" will match before any line separator. You can do this by switching On the modifier /m.

The \A and \Z are just like "^" and "$", except that they won't match multiple times when the modifier /m is used, while "^" and "$" will match at every internal line separator.

The "." meta character by default matches any character, but if you switch off the modifier /s, then '.' won't match embedded line separators.

TRegExpr works with line separators as recommended at http://www.unicode.org/.

"^" is at the beginning of a input string, and, if modifier /m is On, also immediately following any occurrence of \x0D\x0A or \x0A or \x0D (if You are using Unicode version of TRegExpr, then also \x2028 or \x2029 or \x0B or \x0C or \x85). Note that there is no empty line within the sequence \x0D\x0A.

"$" is at the end of a input string, and, if modifier /m is On, also immediately preceding any occurrence of \x0D\x0A or \x0A or \x0D (if You are using Unicode version of TRegExpr, then also \x2028 or \x2029 or \x0B or \x0C or \x85). Note that there is no empty line within the sequence \x0D\x0A.

"." matches any character, but if You switch Off modifier /s then "." doesn't match \x0D\x0A and \x0A and \x0D (if You are using Unicode version of TRegExpr, then also \x2028 and \x2029 and \x0B and \x0C and \x85).

Note that "^.*$" (an empty line pattern) doesn't match the empty string within the sequence \x0D\x0A, but matches the empty string within the sequence \x0A\x0D.

Multiline processing can be easily tuned for Your own purpose with help of TRegExpr properties LineSeparators and LinePairedSeparator, You can use only Unix style separators \n or only DOS/Windows style \r\n or mix them together (as described above and used by default) or define Your own line separators!

Meta Characters - Predefined Classes

\w an alphanumeric character (including "_")
\W a non alphanumeric
\d a numeric character
\D a non-numeric
\s any space (same as [\t\n\r\f])
\S a non space

You may use \w, \d and \s within custom character classes.

Examples:
foob\dr matches strings like 'foob1r', ''foob6r' and so on but not 'foobar', 'foobbr' and so on

foob[\w\s]r matches strings like 'foobar', 'foob r', 'foobbr' and so on but not 'foob1r', 'foob=r' and so on

TRegExpr uses properties SpaceChars and WordChars to define character classes \w, \W, \s, \S, so You can easily redefine it.

Meta Characters - Word Boundaries

\b Match a word boundary
\B Match a non-(word boundary)

A word boundary (\b) is a spot between two characters that has a \w on one side of it and a \W on the other side of it (in either order), counting the imaginary characters off the beginning and end of the string as matching a \W.

Meta Characters - Iterators

Any item of a regular expression may be followed by another type of meta characters - iterators. Using this meta characters You can specify number of occurrences of previous character, meta character or sub-expression.

* zero or more ("greedy"), similar to {0,}
+ one or more ("greedy"), similar to {1,}
? zero or one ("greedy"), similar to {0,1}

{n} exactly n times ("greedy")
{n,} at least n times ("greedy")
{n,m} at least n but not more than m times ("greedy")
*? zero or more ("non-greedy"), similar to {0,}?
+? one or more ("non-greedy"), similar to {1,}?
?? zero or one ("non-greedy"), similar to {0,1}?
{n}? exactly n times ("non-greedy")
{n,}? at least n times ("non-greedy")
{n,m}? at least n but not more than m times ("non-greedy")

So, digits in curly brackets of the form {n,m}, specify the minimum number of times to match the item n and the maximum m. The form {n} is equivalent to {n,n} and matches exactly n times. The form {n,} matches n or more times. There is no limit to the size of n or m, but large numbers will chew up more memory and slow down r.e. execution.

If a curly bracket occurs in any other context, it is treated as a regular character.

Examples:

foob.*r   matches strings like 'foobar', 'foobalkjdflkj9r' and 'foobr'

foob.+r   matches strings like 'foobar', 'foobalkjdflkj9r' but not 'foobr'
foob.?r   matches strings like 'foobar', 'foobbr' and 'foobr' but not 'foobalkj9r'
fooba{2}r   matches the string 'foobaar'
fooba{2,}r matches strings like 'foobaar', 'foobaaar', 'foobaaaar' etc.
fooba{2,3}r matches strings like 'foobaar', or 'foobaaar' but not 'foobaaaar'

A little explanation about "greediness". "Greedy" takes as many as possible, "non-greedy" takes as few as possible. For example, 'b+' and 'b*' applied to string 'abbbbc' return 'bbbb', 'b+?' returns 'b', 'b*?' returns empty string, 'b{2,3}?' returns 'bb', 'b{2,3}' returns 'bbb'.

You can switch all iterators into "non-greedy" mode (see the modifier /g).

Meta Characters - Alternatives

You can specify a series of alternatives for a pattern using "|" to separate them, so that fee|fie|foe will match any of "fee", "fie", or "foe" in the target string (as would f(e|i|o)e). The first alternative includes everything from the last pattern delimiter ("(", "[", or the beginning of the pattern) up to the first "|", and the last alternative contains everything from the last "|" to the next pattern delimiter. For this reason, it's common practice to include alternatives in parentheses, to minimize confusion about where they start and end.

Alternatives are tried from left to right, so the first alternative found for which the entire expression matches, is the one that is chosen. This means that alternatives are not necessarily greedy. For example: when matching foo|foot against "barefoot", only the "foo" part will match, as that is the first alternative tried, and it successfully matches the target string. (This might not seem important, but it is important when you are capturing matched text using parentheses.)

Also remember that "|" is interpreted as a literal within square brackets, so if You write [fee|fie|foe] You're really only matching [feio|].

Examples:
foo(bar|foo) matches strings 'foobar' or 'foofoo'.

Meta Characters - Sub-Expressions

The bracketing construct (...) may also be used for define r.e. sub-expressions (after parsing You can find sub-expression positions, lengths and actual values in MatchPos, MatchLen and Match properties of TRegExpr, and substitute it in template strings by TRegExpr.Substitute).

sub-expressions are numbered based on the left to right order of their opening parenthesis.
First sub-expression has number '1' (whole r.e. match has number '0' - You can substitute it in TRegExpr. Substitute as '$0' or '$&').

Examples:
(foobar){8,10} matches strings which contain 8, 9 or 10 instances of the 'foobar'
foob([0-9]|a+)r matches 'foob0r', 'foob1r' , 'foobar', 'foobaar', 'foobaar' etc.

Meta characters - Back References

Meta characters \1 through \9 are interpreted as back references. \<n> matches previously matched sub-expression #<n>.

Examples:
(.)\1+ matches 'aaaa' and 'cc'.
(.+)\1+ also match 'abab' and '123123'
(['"]?)(\d+)\1 matches '"13" (in double quotes), or '4' (in single quotes) or 77 (without quotes) etc

Modifiers

Modifiers are for changing behaviour of TRegExpr.

There are many ways to set up modifiers.
Any of these modifiers may be embedded within the regular expression itself using the (?...) construct.
Also, You can assign to appropriate TRegExpr properties (ModifierX for example to change /x, or ModifierStr to change all modifiers together). The default values for new instances of TRegExpr object defined in global variables, for example global variable RegExprModifierX defines value of new TRegExpr instance ModifierX property.

i

Do case-insensitive pattern matching (using installed in you system locale settings), see also InvertCase.

m

Treat string as multiple lines. That is, change "^" and "$" from matching at only the very start or end of the string to the start or end of any line anywhere within the string, see also Line separators.

s

Treat string as single line. That is, change "." to match any character whatsoever, even a line separators (see also Line separators), which it normally would not match.

g

Non standard modifier. Switching it Off You'll switch all following operators into non-greedy mode (by default this modifier is On). So, if modifier /g is Off then '+' works as '+?', '*' as '*?' and so on

x

Extend your pattern's legibility by permitting whitespace and comments (see explanation below).

r

Non-standard modifier. If is set then range à-ÿ additional include Russian letter '¸', À-ß additional include '¨', and à-ß include all Russian symbols.
Sorry for foreign users, but it's set by default. If you want switch if off by default - set false to global variable RegExprModifierR.

The modifier /x itself needs a little more explanation. It tells the TRegExpr to ignore whitespace that is neither backslashed nor within a character class. You can use this to break up your regular expression into (slightly) more readable parts. The # character is also treated as a meta character introducing a comment, for example:

```
(
(abc) # comment 1
  |   # You can use spaces to format r.e. - TRegExpr ignores it
(efg) # comment 2
)
```

This also means that if you want real whitespace or # characters in the pattern (outside a character class, where they are unaffected by /x), that you'll either have to escape them or encode them using octal or hex escapes. Taken together, these features go a long way towards making regular expressions text more readable.

Perl Extensions

(?imsxr-imsxr)
You may use it in the regular expression for modifying modifiers on the fly. If the extension is inlined into a sub-expression, then it effects only into that sub-expression.

Examples:

(?i)Saint-Petersburg	matches 'Saint-petersburg' and 'Saint-Petersburg'
(?i)Saint-(?-i)Petersburg	matches 'Saint-Petersburg' but not 'Saint-petersburg'
(?i)(Saint-)?Petersburg	matches 'Saint-petersburg' and 'saint-petersburg'
((?i)Saint-)?Petersburg	matches 'saint-Petersburg', but not 'saint-petersburg'

(?#text)
A comment, the text is ignored. Note that TRegExpr closes the comment as soon as it sees a ")", so there is no way to put a literal ")" in the comment.

9.2 INI Files

INI files are a widely used format for storing text data. They have a specific format which must be adhered to.

The format is:

```
[section name]
name=value
name2=value
```

All other text is ignored. Values can contain spaces.

Notes:
The section name must be unique and is required.
Each name in a section must be unique, and must contain a = sign

10 Support

10.1 Known Problems

Retrying Try Blocks Inside Concurrent Action Lists

Problem: If copies of a single action list is running concurrently inside an Async Action Group (using Run Action List actions), and that action list contains a Try block with Retry enabled, then the retry count will not be properly updated. This can lead to retries being run an unspecified number of times. Although it should never retry indefinitely, it is possible that a Try block will fail without ever being tried.

Note that this issue does not appear when you are simply using different Retry Try Blocks in parallel, or inside different Action Lists. The project must be running multiple copies of the same action list.

Workaround : Do not use retry try blocks inside action lists which might be run concurrently (single action retries will work fine.)

10.2 Automise Support

VSoft Technologies provide support for Automise™ on our forums, and via email.

Web forum interface : http://www.Automise.com/forums.aspx

Support Email : support@Automise.com

Index

- . -

.Net Framework Tools 537
- Precompile ASP.NET 2.0 538
- Register Assembly in COM 541
- Run ASPNET_REGIIS.EXE Action 541
- Serviced Components Installation Tool 542

.Net SDK Tools 544
- GAC Download Cache Action 544
- GAC Install Action 545
- GAC Uninstall Action 546

- A -

ACL
- Change File Permissions 675
- Change Folder Permissions 676

Action Behaviour 84

Action Group Action 225

Action Inspector 12

Action List
- Columns 10
- Exit Action List 229
- Run Action List 236
- Working with Actions 10

Action List Parameters 74

Action Lists 5, 6, 13

Action Logging Properties 88

Action Object 61

Action output monitors 89, 842

Action Properties 61, 84
- Logging 88
- Output Monitors 89
- Processor Affiinity and Process Priority 94
- Run Action as User 92
- Timing 87

Action Types 6, 14

Action Usage Information 40

ActionComment Property 61

ActionLogTitle Property 61

ActionName Property 61

Actions 5
- Adding Actions 10
- Copying Actions 10
- Deleting Actions 10
- Moving Actions 10
- Selecting Actions 10

Actions Summary 81

Actions Tab 14

ActionStudio 83

Activation 633

Active Directory
- Add Users to Group 634
- Create Group 636
- Create Object 637
- Create User 639
- Edit User Properties 640
- Enumerate Group 643
- Get Info 644
- Move Objects 647
- Remove Users From Group 648
- Rename 650

Active Script 52

Add Action List 13

Add Scheduled Task 773

Adding & Editing Variables 68

Adding Actions 10

ADO Dataset Iterator 147

ADO Execute SQL Action 149

ADO Execute Stored Procedure Action 150

AfterAction Event 60

Analsyse Fragmentation on Drive (WMI) 790

Application Variables 73

Archiving 112, 116, 117, 119, 120, 121, 123
- 7Zip 97
- Create Archive 97
- Create Zip File Action 106
- Delete from Archive 101
- Extract Archive 102
- Extract Zip File Action 110
- List Archive 103
- Test Archive 104
- Update Archive 105
- WinRAR Action 123

Ask Question Action 292

ASP.NET Compiler 538

ASPNET_REGIIS 541

Async Action Group 237

Automatic Mounting Disable 705

Automatic Mounting Enable 705

Automatic Usage Submission 40

Automating Automise 820
- Command Line Interface 825
- Console Build Tool 827
- Console Exit Codes 829
- IDE Exit Codes 826
- Scheduling Builds 821

Automating GUI Applications 262

Automise System Variables 73

Auxillary project files 46
Available Drives 689

- B -

Back up Event Log 666
Backup 682
Backup Database Action 137
Backup IIS5 Metabase Action 432
Backup IIS6 Metabase Action 450
Backup Registry Key 279
Basic Concepts 5
Beep Action 294
BeforeAction Event 60
BMP Image Manipulation 519
BR 567
Build History 25
Build Log 6, 28
Build Status Message 156
Burn CD Action 127
Burn ISO Action 130

- C -

Calculate File CRC32 Action 170
Calculate File MD5 Action 171
Calculate Size of Files 194
Carriage Return 567
Case Action 227
Case Statement
  Case Action 227
  Switch Action 241
Catch Action 241
Category 14
CD/DVD Burning
  Burn CD Action 127
  Burn ISO Action 130
  Check Ready Action 130
  Create ISO Action 131
  Erase CDRW Action 132
Change String Case 569
Check Catalogue Action 140
Check Database Action 141
Check Disk 684
Check File Exists Action 172
Check for Updates 42
Check If Host Exists 548
Check if Recycle Bin is Empty 692
Check Ready Action 130
Check Spelling 529
Check Virtual Machine Status 620
Check VM Status (Virtual Server) 608
Checkdisk (WMI) 788
Checkdisk Schedule (WMI) 798
ChildActionCount Property 61
ChildActions Function 61
Chkdsk (WMI) 788
Chkdsk Action 684
Chkdsk Schedule (WMI) 798
Choose One Action 294
Clean Old Mount Point Entries 705
Clear Build History 25
Clear Event Log 667
Clear Recycle Bin 687
Close Process Action 651
COM Class Registered 680
COM+ Delete 653
COM+ Edit Appliction 654
COM+ Register 658
COM+ Shutdown Application 660
COM+ Start Application 661
Command Line Interface
  Console Build Tool 827
  Console Build Tool Exit Codes 829
  IDE Command Line Arguments 825
Comment Action 515
Concatenate Files Action 174
Concetenation of Strings 570
Control Service Action 662
Convert file to PDF 550
Convert HTML to MHT File 309
Convert Images 519
Convert PDF to another format 550
Copy Files Action 174
Copy/Move File List Action 176
Copy/Move FileSet 165
Copying Actions 10
Create Archive 97
Create Directory Action 178
Create IIS5 Virtual Directory Action 433
Create IIS5 Web Site Action 436
Create IIS6 Application Pool 452
Create IIS6 FTP Directory Action 471
Create IIS6 FTP Site Action 474
Create IIS6 Virtual Directory Action 453
Create IIS6 Web Site Action 455
Create ISO Action 131
Create Shortcut 663
Create Snapshot of Virtual Machine 621
Create Sytem Restore Point 739
Create Text File Action 179
Create XML Node 801
Create Zip File Action 106

Creating Custom Actions 83
CRLF 567
CSV Iterator
  Comma 564
  Delimited 564
Cursor Position 270
Custom Actions 83

- D -

Database
  ADO Dataset Iterator 147
  ADO Execute SQL Action 149
  ADO Execute Stored Procedure Action 150
  SQL Server Backup Database Action 137
  SQL Server Best Practices Analyzer 138
  SQL Server Check Catalogue Action 140
  SQL Server Check Database Action 141
  SQL Server DTSRun Action 133
  SQL Server Execute SQL Action 134
  SQL Server Rebuild Indexes Action 142
  SQL Server Remove Unused Space Action 143
  SQL Server Restore Database Action 144
  SQL Server Set Access Mode Action 146
  SQL Server Update DB Statistics Action 147
Date of File/Folder 195
DateTime 58
DateTime Action 517
Debugger (Script) Options 55
Debugging 24
Decrypt String 571
Define XML Document Object 813
Defrag Disk 686
Defrag using WMI 790
Delay Action 227
Delete Directory Action 180, 182
Delete Files Action 182
Delete from Archive 101
Delete IIS5 Virtual Directory Action 438
Delete IIS5 Web Site Action 438
Delete IIS6 Application Pool 457
Delete IIS6 FTP Directory Action 478
Delete IIS6 FTP Site Action 478
Delete IIS6 Virtual Directory Action 459
Delete IIS6 Web Site Action 459
Delete Log Entry 25
Delete Scheduled Task 776
Delete XML Element/Attribute 804
Delete XML Nodes 805
Deleting Actions 10
Depend on a file 189
Description Property 61
Design Time Options 33
Disable Automatic Mounting 705
Disable Sytem Restore 741
Disk Defragment 686
Disk Free Space 689
Disk Information 690
Disk Management
  Format 688, 791
DNS Lookup 701
DOS Command Action 716
Double Quotes 574
Drag & Dropping Actions 10
Drive Defrag (WMI) 790
Drive Free Space 689
Drive Letters 689
DTSRun Action 133
DTSRun Location 141
DVD Burning 127
DVD Check Ready 130

- E -

EC2
  EC2 Attach Volume 420
  EC2 Change Instance States 410
  EC2 Create Image 416
  EC2 Create Snapshot 425
  EC2 Create Volume 422
  EC2 Delete Snapshot 426
  EC2 Delete Volume 423
  EC2 Deregister Image 417
  EC2 Detach Volume 421
  EC2 Get Availability Zones 427
  EC2 Get Image Properties 414
  EC2 Get Instance Properties 409
  EC2 Get Snapshot Properties 424
  EC2 Get Volume Properties 418
  EC2 Run Instances 411
  EC2 Terminate Instance 413
Echo Procedure 61
Edit Field 96
Edit Password 49
Edit XML File Action 806
Editor (Script) Options 53
Else Action 228
EMail Action 346
Empty Recycle Bin 687
Enable Automatic Mounting 705
Enable Sytem Restore 741
Encrypt String 571

End Action 241
End Running Scheduled Task 777
Environment Variables 73
Erase CDRW Action 132
Error Codes
    Console Error Codes 829
    IDE Error Codes 826
Error Handling 23
Errors 23
Escaping Variable References 72
Estimated Progress 18, 155
Event 60
Event Log 665
    Backing Up 666
    Clearing 667
    Searching 668
Exceptions 23
Execute Condition 60, 63
Execute Program Action 672
Execute SQL Action 134
Exit Action List Action 229
Exit Codes
    Console Exit Codes 829
    IDE Exit Codes 826
ExpandExpression Function 61
Export Log 25, 153
Export Log Fragment 153
Export Registry Key 279
Extension of Files 565
Extract Archive 102
Extract File Version Action 183
Extract String 579
Extract XML Fragment Action 808
Extract Zip File Action 110

- F -

FBCmd 827
Field Editor 96
File & Folder Security
    Change File Permissions 675
    Change Folder Permissions 676
    Change Ownership 678
File Contents Iterator Action 501
File Dependency 189
File Iterator Action 502
File Share 780
    Create Share 781
    Find Share 783
    Remove Share 782
File Size Action 194
File Types 46
File/Folder Date 195
Filename Manipulation 565
Filepath Manipulation 565
Files & Directories
    Calculate File CRC32 Action 170
    Calculate File MD5 Action 171
    Check File Exists Action 172
    Concatenate Files Action 174
    Copy Files Action 174
    Copy/Move File List Action 176
    Create Directory Action 178
    Create Text File Action 179
    Delete Directory Action 180
    Delete Files Action 182
    Extract File Version Action 183
    File Dependency 189
    Find File(s) 192
    Get File Size 194
    Get File/Folder Date 195
    Move Directory Action 197
    Move Files Action 198
    Read Text File Action 200
    Rename File or Directory Action 201
    RoboCopy Actions 202
    Set File Attributes Action 215
    Touch Files Action 217
    Write Text File Action 219
    XCopy Action 220
FileSet Copy/Move 165
FileSet Define 160
FileSet Delete Files 167
FileSet Filters 164
FileSet Log Files 167
FileSet Patterns 162
FileSet Refresh 168
FileSet to Variable 169
FileSets 159
    Copy/Move FileSet 165
    FileSet Accessing 159
    FileSet Define 160
    FileSet Delete Files 167
    FileSet Filters 164
    FileSet Log Files 167
    FileSet Refresh 168
    FileSet Specifying Patterns 162
    FileSet to Variable 169
Filter 14
Filter Log By Action 28, 30
Finally Action 241
Find / Replace
    In IDE 22

Find / Replace
- Text Find / Replace Action 532

Flow Control 225
- Async Action Group 237
- Case Action 227
- Catch Action 241
- Conditional Execution 63
- Delay Action 227
- Else Action 228
- End Action 241
- Exit Action List 229
- Finally Action 241
- If .. Then Action 231
- If Prev Action Failed Action 233
- Include Automise Project 234
- Iterators 499
- Parallelization 237
- Raise Exception Action 235
- Run Action List 236
- Stop Run Action 240
- Switch Action 241
- Try Action 241
- Waiting For Events 244
- While Loop 261

Folder Iterator Action 503

Folder name Manipulation 565

Folder Share 780
- Create Share 781
- Find Share 783
- Remove Share 782

Format 688, 791

Format DateTime 58

Free Space of Disk 689

FTP Actions 310

FTP Client 338

FTP Mirror
- FTP 326
- Mirror 326
- Sync 326

FTP Upload 333

FTPS Change Directory
- Change Directory 386
- FTPS 386

FTPS Change Up Directory
- Change Up Directory 387
- FTPS 387

FTPS Check If Connected
- Check If Connected 387
- FTPS 387

FTPS Connect
- Connect 388
- FTPS 388

FTPS Create Directory
- Create Directory 392
- FTPS 392

FTPS Delete File
- Delete File 393
- FTPS 393

FTPS Disconnect
- Disconnect 394
- FTPS 394

FTPS Dowload Files
- Download Files 396
- FTPS 396

FTPS Download File
- Download File 395
- FTPS 395

FTPS Get File Date
- FTPS 398
- Get File Date 398

FTPS Get File Size
- FTPS 399
- Get File Size 399

FTPS Get Remote Checksum
- FTPS 400
- Get Remote Checksum 400

FTPS Get Remote Directory
- FTPS 401
- Get Remote Directory 401

FTPS List Directory
- FTPS 401
- List Directory 401

FTPS No-op
- FTPS 404
- No-op 404

FTPS Options 385

FTPS Remove Directory
- FTPS 405
- Remove Directory 405

FTPS Rename File
- FTPS 405
- Rename File 405

FTPS Send Command
- FTPS 406
- Send Command 406

FTPS Upload Files
- FTPS 408
- Upload Files 408

- G -

GAC Download Cache Action 544

GAC Install Action 545

GAC Uninstall Action 546
Generate New GUID 679
Generate Random Number 516
Get Available Drives 689
Get DateTime Action 517
Get Disk Free Space Action 689
Get Drive Letters 689
Get File Size Action 194
Get File/Folder Date 195
Get Last Sytem Restore Status 744
Get Logical Disk Information 690
Get Mount Points for Volume Names 705
Get Mounted Volume Name 706
Get Sytem Restore Configuration 742
Getting Started 20
    Basic Concepts 5
    IDE Modes 6
GIF Image Manipulation 519
Global Script 47
GUI Automation 262
    Perform Mouse Click 268
    Perform Mouse Move 270
    Send Keyboard Input 271
    Set Scrollbar Position 274
    Specifying target controls 263, 264
    Wait for Window or Control 275
GUID (Generate new) 679

- H -

Hot Fixes
    Check if Installed 785
    List Installed 786
HTML to MHT File 309
HTTP Get Action 342
Hyper V 595
    Apply VM Snapshot 604
    Check VM State 598
    Create VM Snapshot 603
    Delete VM Snapshot 605
    List Virtual Machines 606
    Pause VM 601
    Reset VM 600
    Server 596
    Start VM 598
    Stop VM 599
    Suspend VM 602
Hyper V Options 595

- I -

ICQ Action 352
IDE Debug Mode 24
    Watches 25
IDE Design Mode 9
    Action Lists 13
    Actions Tab 14
    Project Tree 16
    Properties Tab 12
    Quick Help 20
    Script Editor 20
    Working with Actions 10
IDE Modes 6
IDE Running Mode 26
    Build History 25
    Estimated Progress 18
    Run History 25
    Run Log 28
    Running a project 27
    Validation 31
If .. Then Action 231
If COM Class Registered 680
If Prev Action Failed Action 233
Ignore Error 23
IgnoreFailure Property 61
IIS 429
IIS5 429
    Backup IIS Metabase Action 432
    Create Virtual Directory Action 433
    Create Web Site Action 436
    Delete Virtual Directory Action 438
    Delete Web Site Action 438
    Restore IIS Metabase Action 439
    Start/Stop/Pause Website Actions 440
    Start/Stop/Restart WWW Service Actions 443
    Unload IIS Application Action 446
IIS6 447
    Backup IIS Metabase Action 450
    Create Application Pool 452
    Create Virtual Directory Action 453
    Create Web Site Action 455
    Delete Application Pool 457
    Delete Virtual Directory Action 459
    Delete Web Site Action 459
    Restore IIS Metabase Action 460
    Start/Stop/Pause Website Actions 461
    Start/Stop/Restart WWW Service Actions 464
    Unload IIS Application Action 467
IIS6 FTP 468

IIS6 FTP 468
- Create FTP Directory Action 471
- Create FTP Site Action 474
- Delete FTP Directory Action 478
- Delete FTP Site Action 478
- Start/Stop/Pause FTP Site Actions 479
- Start/Stop/Restart FTP Service Actions 482

Image Manipulation 519

Images
- Convert, Resize, Rotate 519
- EXIF 523
- Lossless Transform 523
- Rotate 523

Include Automise Project Action 234

Information 737

INI File Iterator Action 504

Ini Files
- Load Variables to INI Action 585
- Read 279
- Save Variables to INI Action 591
- Write 284

INI Files Reference 863

Input Box 296

Insert Text in PDF file 551

Install Application 290

Interactive
- Ask Question Action 292
- Choose One Action 294
- Input Box 296
- MessageBox 297
- Multi Question Action 298
- Prompt for File or Directory 300
- Prompt for Password 301
- Prompt for Variables 303
- Prompt for Variables (Enhanced) 305

Internet
- Convert HTML to MHT File 309
- Download 361
- FTP Actions 310
- FTP Client 338
- FTP Upload 333
- HTTP Get Action 342
- ICQ Action 352
- NNTP News Post Action 343
- Send Email Action 346
- Send MSN Message 353
- SSH Batch Execute Action 366
- SSH Close Action 364
- SSH Execute Command Action 365
- SSH Open Action 363
- Telnet Action 356

Internet Information Services 429
- IIS5 429
- IIS6 447
- IIS6 FTP 468

Internet Information Services 5
- Backup IIS Metabase Action 432
- Create Virtual Directory Action 433
- Create Web Site Action 436
- Delete Virtual Directory Action 438
- Delete Web Site Action 438
- Restore IIS Metabase Action 439
- Start/Stop/Pause Website Actions 440
- Start/Stop/Restart WWW Service Actions 443
- Unload IIS Application Action 446

Internet Information Services 6
- Backup IIS Metabase Action 450
- Create Application Pool 452
- Create Virtual Directory Action 453
- Create Web Site Action 455
- Delete Application Pool 457
- Delete Virtual Directory Action 459
- Delete Web Site Action 459
- Restore IIS Metabase Action 460
- Start/Stop/Pause Website Actions 461
- Start/Stop/Restart WWW Service Actions 464
- Unload IIS Application Action 467

Internet Information Services 6 FTP
- Create FTP Directory Action 471
- Create FTP Site Action 474
- Delete FTP Directory Action 478
- Delete FTP Site Action 478
- Start/Stop/Pause FTP Site Actions 479
- Start/Stop/Restart FTP Service Actions 482

Is Recycle Bin Empty 692

Is Rubbish Bin Empty 692

ISO Buring 130

ISO Creation 131

ISQL Location 141

Iterators
- ADO Dataset Iterator 147
- File Contents Iterator 501
- File Iterator 502
- Folder Iterator 503
- For Loop 229
- General Information 499
- INI File Iterator 504
- List Iterator 505
- While Loop 261
- XML Node Iterator 816

- J -

JavaScript 20, 52
JPEG Image Manipulation 519
JPEG Lossless Transform 523
JPG Image Manipulation 519
JScript 20, 52
Junction Points
    Clean Old Mount Point Entries 705
    Disable Automatic Mounting 705
    Enable Automatic Mounting 705
    Get Mount Points for Volume Names 705
    Get Mounted Volume Name 706
    Mount Volume 707
    Overview 704
    Permanent Remove Mount Point 708
    Remove Mount Point 709

- K -

Kill Process (WMI) 793
Known Problems 864

- L -

License 1
Line Feed 567
List Archive 103
List Iterator Action 505
List Sort 528
List System Restore Points 744
List VMs (Virtual Server) 610
List VMs in (VMWare Server) 617
Live Logging 28
Load Variables From INI Action 585
Lock file 526
Log Action 153
Logging Options 36
Logical Disk Defragment 686
Logical Disk Information 690
Logical Drive Defrag (WMI) 790
LogToVariable Property 61
Loop 501, 502, 503, 504, 505
Lossless Transform 523
Lowercase 569

- M -

Main 5, 13
Map Network Drive 693
Merge PDF Files 553
Merge XML Action 807
Message Queuing 507
    Clear Queue 508
    Create Queue 509
    Delete Queue 510
    Peek Message 511
    Receive Message 512
    Send Message 513
    Wait for Message 514
MessageBox 297
MessageBox Constants 59
Messages 42
MHT File Conversion 309
Microsoft Message Queuing 507
    Clear Queue 508
    Create Queue 509
    Delete Queue 510
    Peek Message 511
    Receive Message 512
    Send Message 513
    Wait for Message 514
Microsoft Powershell 670
Modified date 195
Mount Point Entries Cleanup 705
Mount Point Permanent Remove 708
Mount Point Remove 709
Mount Points for Volume Names 705
Mount Volume 707
Mounted Volume Name 706
Mouse Clicks 268
Mouse Movement 270
Move Directory Action 197
Move Files Action 198
Moving Actions 10
MSI 285
    Install Application 290
    MSI Generic 285
    MSI Merge 286
    MSI Transform 287
    MSI Update Properties 288
    MSI Validation 289
    Reinstall Application 290
    Uninstall Application 291
MSMQ 507
    Clear Queue 508
    Create Queue 509
    Delete Queue 510
    Peek Message 511
    Receive Message 512
    Send Message 513

MSMQ 507
- Wait for Message 514

MSN Send Message 353
Multi Question Action 298
Mutex 526

- N -

Name Server Lookup 701
Natural sort 528
Net Send Message Action 696
Network 548
- Configure IP Security 697
- Configure Network 698
- Find Network Adapters 699

Network Configuration
- Configure IP Security 697
- Configure Network 698
- Find Network Adapters 699

Network Diagnostics 704
Network Drive Information 690
Network drive letters 689
Network Drive Mapping 693
Network Drive Unmapping 695
New Line 567
News Post Action 343
NNTP News Post Action 343
NSLookup 701
NT Backup 682
NTFS Junction Points
- Clean Old Mount Point Entries 705
- Disable Automatic Mounting 705
- Enable Automatic Mounting 705
- Get Mount Points for Volume Names 705
- Get Mounted Volume Name 706
- Mount Volume 707
- Overview 704
- Permanent Remove Mount Point 708
- Remove Mount Point 709

Number (Random) 516

- O -

OnFailure 5, 13
Options
- Design Time 33
- Logging 36
- Run Time 35
- Script Debugger 55
- Script Editor 53
- Setting via scripting 64
- SQL Server Options 141
- Validation 39

Options Dialog 32
Options screen 25
Order 528
Orientation 14
OSQL Location 141
Output Monitors 89
Output Tree 28
Override progress 155
Overview 1

- P -

Package Manager 42
Pad String 573
Page File List Configuration Properties 771
Page File List Performance 769
Page File List Usage 772
Page File Overview 768
pagefile.sys 768
Parallel Build Process 237
Parent Function 61
Password for Project 49
Password Prompt 301
Patches
- Check if Installed 785
- List Installed 786

Path Manipulation 565
Pause (Delay Action) 227
Pause VM (Virtual Server) 611
PauseInterval Property 61
PDF
- Convert PDF to another format 550
- Convert to PDF 550
- Insert Text 551
- Merge PDF files 553

PDF Convert to another format 550
PDF Insert Text 551
PDF Merge files 553
Perform Mouse Click 268
Perform Mouse Move 270
Perform Sytem Restore 746
Permanent Remove Mount Point 708
Permissions
- Change File Permissions 675
- Change Folder Permissions 676

Plugins 42
Powershell 52, 670
Precompile ASP.NET 2.0 538
Print Queue 712

Printer Share 780
- Create Share 781
- Find Share 783
- Remove Share 782

Printers 710
- Manage Print Queue 711
- Manage Printer 712

Process Close Action 651
Process Info Action 794
Process Iterator (WMI) 796
Process Priority 94
Processor Affiinity 94
Progress 156
Progress bar 18, 155
Project file types 46
Project Global Script 47
Project Password 49
Project Summary 50
Project Tree 16
Project Variables 72
Prompt for File or Directory Action 300
Prompt for Password 301
Prompt for Variables (Enhanced) Action 305
Prompt for Variables Action 303
Properties Property 61
Properties Tab 12
PsTools 718
- PsExec 719
- PsFile 723
- PsGetSID 724
- PsInfo 725
- PsKill 726
- PsList 727
- PsLoggedOn 728
- PsLogList 729
- PsService 733
- PsShutdown 734
- PsSuspend 736

- Q -

Queue 553
Queue Clear 555
Queue Define 554
Queue get count 555
Queue Insert 556
Queue is empty 557
Queue Iterator 558
Queue log items 560
Queue Peek 561
Queue Pop 562
Queue Push 563
Quick help 20
Quotes in Strings 574

- R -

Raise Exception Action 235
Random 528
Random Number 516
RAR Files 123
Rasdial VPN 702
Read Ini 279
Read Text File Action 200
Read XML Value to Variable 809
Reboot Action 714
Rebuild Indexes Action 142
Recycle Bin Empty 687
Recycle Bin Is Empty 692
Recycle Bin size and item count 692
REGASM 541
Register Assembly in COM 541
Register COM+ 658
Register DLL/OCX Action 713
Registry
- Delete Registry Value 280
- Read Registry Value 280
- Set Registry Value 280

Registry Export 279
Registry Key Save 279
REGSVCS 542
Regular Expression Reference 857
Reinstall Application 290
Remote Network Connection 702
Remote Reboot Action 714
Remote Shutdown 715
Remote Task Scheduling 779
Remove (Permanent) Mount Point 708
Remove Mount Point 709
Remove Unused Space Action 143
Remove whitespace 580
Rename File or Directory Action 201
Replace String 576
Reset Virtual Machine 622
Reset VM 618
Reset VM (Virtual Server) 612
Resize Images 519
Restore Database Action 144
Restore IIS5 Metabase Action 439
Restore IIS6 Metabase Action 460
Reverse String 578
Revert to Snapshot of Virtual Machine 622

Robocopy 208, 209, 210, 212, 213
RoboCopy Actions 202
Rotate Images 519
RTF convert to PDF 550
Rubbish Bin Empty 687
Run Action as User 92
Run Action List Action 236
Run ASPNET_REGIIS.EXE Action 541
Run Command 27
Run Process (WMI) 797
Run Scheduled Task 778
Run Scipt Action 527
Run Status Message 156
Run Time Options 35
Running 156

- S -

Save Log to File 153
Save Variables To INI Action 591
Save VM State (Virtual Server) 613
Save XML Document Object 820
Scandisk 684
Scandisk (WMI) 788
Scandisk Schedule (WMI) 798
Schedule Chkdsk (WMI) 798
Scheduled Task Add 773
Scheduled Task Delete 776
Scheduled Task End Running 777
Scheduled Task Run 778
Scheduling Builds 821
Script
    Global Script 47
Script Debugger Options 55
Script Editor 20
Script Editor Options 53
Script Functions 56
Scripting 52
    Accessing Options 64
    Accessing TStrings 65
    Action Methods 61
    Action Properties 61
    Events 60
    Execute Condition 63
    Global Functions 56
Scrollbar Position 274
Search 14
Search / Replace
    In IDE 22
    Text Find / Replace Action 532
Search Event Log 668
SecureZIP 112, 116, 117, 119, 120, 121, 123
    SecureZIP Options 111
SecureZIP Add Files To Archive 112
SecureZIP Common Pages 119
SecureZIP Exclude List 123
SecureZIP Extract Archive 116
SecureZIP Filters 121
SecureZIP List Archive Contents 117
SecureZIP Security 120
SecureZIP Test Archive Integrity 119
Selecting Actions 10
Send Email 346
Send Keyboard Input 271
Send Message Action 696
Send MSN Message 353
SendInput 268
SendKeys 271
SendLogMessage Procedure 61
SendProgress Procedure 61
Sentence 569
Serviced Components Installation Tool 542
Set Build Status Message 156
Set File Attributes Action 215
Set Run Status Message 156
Set Scrollbar Position 274
Set Sytem Restore Configuration 747
Set Variable Action 593
SetLogTitle Procedure 61
Shares 780
    Create Share 781
    Find Share 783
    Remove Share 782
Shell Execute Action 718
Shortcut create 663
Shutdown Action 715
Single Quotes 574
Size of Files Action 194
Size of Recycle Bin 692
SMTP 346
Sort Text List 528
Sound 294
Space 580
Spell Check 529
SQL Server
    Backup Database Action 137
    Best Practices Analyzer 138
    Check Catalogue Action 140
    Check Database Action 141
    DTSRun Action 133
    Execute SQL Action 134
    Rebuild Indexes Action 142
    Remove Unused Space Action 143

SQL Server
  Restore Database Action 144
  Set Access Mode Action 146
  Update DB Statistics Action 147
SQL Server Best Practices Analyzer 138
SQL Server Options 141
SQLCmd Location 141
SQLMaint Location 141
SSH Batch Execute Action 366
SSH Close Action 364
SSH Execute Command Action 365
SSH Open Action 363
Stack 553
Stack Clear 555
Stack Define 554
Stack get count 555
Stack Insert 556
Stack is empty 557
Stack Iterator 558
Stack log items 560
Stack Peek 561
Stack Pop 562
Stack Push 563
Start Virtual Machine 623
Start VM 618
Start VMs (Virtual Server) 614
Start/Stop/Pause IIS5 Website Actions 440
Start/Stop/Pause IIS6 FTP Site Actions 479
Start/Stop/Pause IIS6 Website Actions 461
Start/Stop/Restart IIS5 WWW Service Actions 443
Start/Stop/Restart IIS6 FTP Service Actions 482
Start/Stop/Restart IIS6 WWW Service Actions 464
Status 156
Stepping 24
Stop Command 27
Stop Run Action 240
Stop Virtual Machine 624
Stop VM 618
Stopping a build 27
String Manipulation
  Breaks 567
  Case 569
  Concatenation 570
  Encryption 571
  Find / Replace 532
  Index 575
  Pad 573
  Path 565
  Pos 575
  Position 575
  Quotes 574
  Replace 576
  Reverse 578
  Substring 579
  Trim 580
Subst Drive Action 694
Substring 579
Support 864
SuppressStatusMessages Property 61
Suspend Virtual Machine 625
Suspend VM 618
Switch Action 241
Sysinternals 718
  PsExec 719
  PsFile 723
  PsGetSID 724
  PsInfo 725
  PsKill 726
  PsList 727
  PsLoggedOn 728
  PsLogList 729
  PsService 733
  PsShutdown 734
  PsSuspend 736
System Information 737
System Restore
  Create Restore Point 739
  Disable 741
  Enable 741
  Get Configuration 742
  Get Last Status 744
  List Restore Points 744
  Overview 738
  Perform Restore 746
  Set Configuration 747
System Tray 43
System Variables 73

- T -

Target Property Page 263, 264
Task Scheduling
  Add Scheduled Task 773
  Delete Scheduled Task 776
  End Running Scheduled Task 777
  Run Scheduled Task 778
Task Scheduling Remote Tasks 779
Telnet Action 356
Terminate Command 27
Test Archive 104
Text Find / Replace 532
Text Sort 528
TIFF Image Manipulation 519

Timeout (Delay Action) 227
Timing 87
Touch Files Action 217
Trace Route 704
Tracert 704
Transform Image 523
Transform XML Action 811
Tray Icon 43
Trim String 580
Try Action 241
TStrings 65
Turn off VM (Virtual Server) 615

- U -

Uninstall Application 291
Unload IIS5 Application Action 446
Unload IIS6 Application Action 467
Unmap Network Drive 695
Update Archive 105
Update DB Statistics Action 147
Updates 20, 42, 787
Uppercase 569
User Action Lists 5
User Variables 73
Using Variables 71

- V -

Validate XML Action 812
Validation 31
Validation Options 39
Variable Options 75
Variable Sense 75
Variables
- Action List Parameters 74
- Adding & Editing Variables 68, 75
- Application Variables 73
- Environment Variables 73
- Escaping References 72
- Load from INI Action 585
- Overview 66
- Predefined Automise Variables 73
- Project Variables 72
- Save To INI Action 591
- Set Variable Action 593
- User Variables 73
- Using Variables 71

VBScript 20, 52
Version History 1
View Log 25
Virtual Server 607
- Check VM Status 608
- List VMs 610
- Pause VM 611
- Reset VM 612
- Save VM State 613
- Start VMs 614
- Turn off VM 615

VMWare 626, 628, 629, 630, 631
VMWare Server 616, 626, 629, 630, 631
- List Virtual Machines 628
- List VMs 617
- Reset VM 618
- Start VM 618
- Stop VM 618
- Suspend VM 618

VMWare Workstation
- Check Virtual Machine Status 620
- Create Snapshot 621
- Reset Virtual Machine 622
- Revert to Snapshot 622
- Start Virtual Machine 623
- Stop Virtual Machine 624
- Suspend Virtual Machine 625

Volume Mount 707
Volume name mount points 705
VPN (Rasdial) 702

- W -

Wait (Delay Action) 227
Wait for Window or Control 275
Waiting For Events
- Automise Variables 258
- Certain Date or Time 259
- Command Output 246
- Files 247
- Ini File Change 249
- Overview 244
- Process 251
- Registry Change 252
- Remote Computer 254
- Script Result 256

Watches 24, 25
WGet Download Action 361
While Loop Action 261
Whitespace 580
Window Exists Action 767
Windows Activation 633
Windows Administration
- Active Directory 634, 636, 637, 639, 640, 643, 644, 647, 648, 650

Windows Installer 285
- Install Application 290
- MSI Generic 285
- MSI Merge 286
- MSI Transform 287
- MSI Update Properties 288
- MSI Validation 289
- Reinstall Application 290
- Uninstall Application 291

Windows OS
- Chkdsk 684
- Defrag Disk 686
- Empty Recycle Bin 687
- Is Recycle Bin Empty 692

Windows Page File 768
- Page File List Configuration Properties 771
- Page File List Performance 769
- Page File List Usage 772

Windows Powershell 670

Windows Scheduler Actions
- Add Scheduled Task 773
- Delete Scheduled Task 776
- End Running Scheduled Task 777
- Run Scheduled Task 778

Windows Scheduling Remote Tasks 779

Windows Share 780

Windows Shares
- Create Share 781
- Find Share 783
- Remove Share 782

Windows Update 787

Windows Updates
- Check if Installed 785
- List Installed 786

WinRAR Action 123

WMI Checkdisk 788

WMI Checkdisk Schedule 798

WMI Chkdsk 788

WMI Chkdsk Schedule 798

WMI Defrag 790

WMI Kill Process Action 793

WMI Process Info Action 794

WMI Process Iterator 796

WMI Run Process Action 797

Working with Actions 10

WPA 633

Write Ini 284

Write Text File Action 219

- X -

XCopy Action 220

XML
- Create XML Node 801
- Define XML Document Object 813
- Delete XML Element/Attribute 804
- Delete XML Nodes 805
- Edit XML File Action 806
- Extract XML Fragment Action 808
- Merge XML Action 807
- Node Iterator 816
- Read XML Value to Variable 809
- Save XML Document Object 820
- Transform XML Action 811
- Validate XML Action 812

XML Define Document Object 813

XML Node Iterator 816

XML Save Document Object 820

- Z -

Zip Files 97, 101, 102, 103, 104, 105, 106, 110