

FinalBuilder

Automating the Build Process

Table of Contents

Part I FinalBuilder	1
1 FinalBuilder Overview	1
2 FinalBuilder Server	2
3 Version History	3
4 License	4
Part II Getting Started	7
1 FinalBuilder Basic Concepts	7
2 FinalBuilder IDE Modes	8
3 Upgrading Projects From Previous Versions	9
Part III FinalBuilder IDE Reference	11
1 Welcome Page	11
2 IDE Design Mode	13
Working with Actions	14
Properties tab	17
Action Lists	18
Actions Tab	20
Project Tree	22
Run Tab	24
Quick Help	26
Script Editor	27
Keyboard Shortcuts	28
Search & Replace	30
Error Handling	32
3 IDE Debug Mode	33
Watches	34
4 History & Statistics	35
Build Statistics	37
Success Ratio Chart.....	37
Main Chart Types.....	38
Chart Data Options.....	39
Chart Visual Options.....	40
Exporting Build Statistics.....	41
Printing Statistics Chart.....	43
5 IDE Running Mode	44
Running a Project	45
Build Log	47
Validation	50
6 IDE Features	51
Options Dialog	51
Design Time Options.....	52
Runtime Options.....	54
Build Statistics Options.....	55
Logging Options.....	57
Validation Options.....	60

Welcome Page Options	61
Source Control Integration	62
Source Control Options	65
Automatic Action Usage Updates	67
Check for Updates	69
Messages	70
Package Manager	71
Tray Icon	73
Keyboard Shortcuts	75
7 FinalBuilder Projects	77
Project Files & Other Files	77
Project Global Script	79
Project Edit Password	81
Project Summary	83
Project Information	84
Configuration Wizard	85

Part IV Scripting 86

1 Scripting in FinalBuilder	86
2 Script Editor Options	88
3 Script Debugging Options	90
4 Global Script functions	91
Format DateTime Formatting Options	94
MessageBox Constants	97
5 Action Script Events	98
Action Properties and Methods	100
Execute Condition	102
6 Accessing the Options settings via scripting	104
7 Accessing TStrings based properties	105

Part V Variables 106

1 Variables Overview	106
2 Adding Variables	109
3 Using Variables	112
4 Variable Evaluation and Recursion	113
5 Escaping Variable References	114
6 Variable Types	115
Project Variables	115
User Variables	116
Environment Variables	117
Application Variables	118
7 Action List Parameters	121
8 Variable Options	123
9 Variable Format Strings	126
10 DateTime Format Strings	128

Part VI Actions Reference 130

1 Custom Actions	133
2 Common Action Properties	134
Timing Properties	137

Logging Properties	138
Action Output Monitors	139
Run Action as User	143
Setting Processor affinity & process priority	145
Error Dialog Monitor	146
3 Edit Field	148
4 Archiving	149
7Zip	149
Create Archive.....	149
Delete from Archive.....	153
Extract Archive.....	153
List Archive.....	154
Test Archive.....	155
Update Archive.....	155
Create Zip File	156
Extract Zip File Action	160
SecureZIP	162
SecureZIP Add Files To Archive.....	163
SecureZIP Extract Archive.....	166
SecureZIP List Archive Contents.....	167
SecureZIP Test Archive Integrity.....	169
SecureZIP Common Pages	170
SecureZIP Security Page.....	170
SecureZIP Filters Page.....	173
SecureZIP Exclude List Page.....	174
WinRAR Action	176
5 Build Tools	177
Ant Project Action	177
MSBuild Project Action	180
Nant Project Action	183
6 CD/DVD Burner Actions	184
Burn CD/DVD Action	185
Burn ISO Action	188
Check Ready Action	189
Create ISO Action	190
Erase CD/DVD RW	191
7 Compilers	192
AssemblyInfo Updater Action	192
Embarcadero	194
Embarcadero C++ Builder Action.....	194
Embarcadero C++Builder Compiler Options.....	195
Embarcadero C++Builder Advanced Compiler Options.....	195
Embarcadero C++Builder Advanced Linker Options.....	196
Embarcadero C++Builder CodeGuard Options.....	196
Embarcadero C++Builder Corba Options.....	196
Embarcadero C++Builder CPP Options.....	197
Embarcadero C++Builder Linker Options.....	197
Embarcadero C++Builder Pascal Options.....	198
Embarcadero C++Builder Scripting Reference.....	198
Embarcadero C++Builder TASM Options.....	200
Embarcadero C++Builder TLib Options.....	200
Using PropertySets w ith Embarcadero C++ Builder.....	201
Build Delphi Project.....	201
Delphi Options	205
Using PropertySets w ith Delphi.....	206
Using EurekaLog.....	207

Embarcadero C# Builder Project Compiler Action.....	208
Delphi for .NET Action.....	208
BPG File Iterator.....	213
Compile Embarcadero Resource Script.....	215
MadExcept Compiler Action.....	216
Chrome	218
Incredibuild	219
Java Compiler Action	220
Embarcadero Compiler Options	222
JDK Configurations.....	222
Microsoft	223
Compile Visual Basic Project.....	223
VB6 Version Compatibility.....	226
Using PropertySets w ith VB6.....	227
Visual C++ 6 Action.....	228
Using PropertySets w ith Visual C++.....	231
Visual Studio .NET Action.....	232
Using PropertySets w ith VisualStudio.Net.....	242
Visual Studio .NET Solution Iterator	244
Microsoft C# Compiler Action.....	246
Microsoft C# Project Compiler Action.....	247
Microsoft VB.NET Compiler Action.....	249
Microsoft VB.NET Project Compiler Action.....	249
Microsoft J# Compiler Action.....	251
Microsoft J# Project Compiler Action.....	252
MSIL Assembler (ILASM) action	253
8 Database	255
SQL Server Actions	255
DTSRun Action.....	255
Execute SQL Action.....	256
SQL Server Backup Database.....	258
SQL Server Best Practices Analyzer.....	259
SQL Server Check Catalogue.....	261
SQL Server Check Database.....	261
SQL Server Options	261
SQL Server Rebuild Indexes.....	262
SQL Server Remove Unused Space.....	263
SQL Server Restore Database.....	264
SQL Server Set Access Mode.....	265
SQL Server Update DB Statistics.....	266
ADO Dataset Iterator	267
ADO Execute SQL	269
ADO Execute Stored Procedure	270
9 Executable Packers	272
ASPack Action	272
UPX Pack Executable	274
10 FinalBuilder IDE	277
Save FinalBuilder Project Action	277
Export Log Action	278
Set Estimated Progress Total Action	281
Set Build Status Message Action	282
Create ActionStudio Documentation Action	284
11 FileSets	285
Accessing FileSets	286
FileSet Define	288
FileSet Patterns.....	289
FileSet Filters	291

FileSet Copy/Move Action	293
FileSet Delete Files	295
FileSet Log Files	296
FileSet Refresh	297
FileSet to Variable	298
12 Files & Directories	299
Authenticode	299
Calculate File CRC32	301
Calculate File MD5	303
Check if File Exists Action	305
Concatenate Files Action	307
Copy File(s) Action	308
Copy/Move File List	310
Create Directory Action	312
Create Text File	313
Delete Directory Action	314
Delete File(s) Action	316
Extract Version Info	318
File Comparison Tools	320
Araxis Merge Compare Actions	320
Beyond Compare.....	322
Beyond Compare Action.....	322
Run Beyond Compare Script.....	323
File Compare Action.....	324
File Dependency Action	326
Find File(s)	330
Get File Size	332
Get File/Folder Date	333
Move Directory Action	335
Move File(s) Action	336
Read Text File	338
Rename File or Directory Action	340
RoboCopy	341
Robocopy	341
Robocopy Mirror.....	343
Robocopy Move.....	343
Run Robocopy Job.....	344
Robocopy Common Pages.....	347
Set File Attributes Action	355
Signtool Actions	356
Signtool Sign Files.....	356
Signtool Timestamp Files.....	359
Signtool Verify Digital Signatures	360
Text Replace Action	363
Touch File(s)	365
Write Text File	367
XCopy	368
13 Flow Control	373
Action Group	374
Case Action	376
Delay Action	377
Else Action	378
Exit Action List Action	379
For Loop Action	380
If .. Then Action	382
If Prev Action Failed Action	386
Include FinalBuilder Project	387
Raise Exception	389

Run Action List Action	390
Running actions in Parallel	391
Stop Build Action	395
Switch Action	396
Try/Catch/Finally/End Actions	397
Waiting For Events (Wait For actions)	400
Wait For Command.....	401
Wait For File.....	403
Wait For Ini File.....	405
Wait For Process.....	407
Wait For Registry.....	409
Wait For Remote Computer.....	411
Wait For Script Result.....	413
Wait For Variable.....	416
Wait Until.....	417
While Loop Action	420
14 GUI Automation	421
Overview	421
Target Property Page	422
Automatic Configuration.....	422
Manual Configuration.....	423
Actions	428
Perform Mouse Click.....	428
Perform Mouse Move.....	429
Send Keyboard Input.....	430
Set Scrollbar Position.....	433
Wait For Window or Control.....	435
15 Help Compilers	439
Build Doc-O-Matic Project	439
Help & Manual Action	441
HTML Help Compiler	443
NDoc Action	444
Sandcastle - BuildAssembler	445
Sandcastle - MRefBuilder	446
Sandcastle - XslTransform	448
WinHelp Compiler	449
16 Ini Files & Registry	451
Export Registry Key	451
Read Ini File	452
Read/Set/Delete Registry Value	453
Write Ini File	457
17 Installers	458
GPInstall Action	458
Inno Setup	459
InstallAnywhere .Net	460
InstallAnywhere Enterprise	461
InstallAware	462
InstallAware Group Policy Wizard	465
InstallShield	467
InstallShield 2008, 2009 & 2010.....	467
InstallShield Developer.....	468
InstallShield Global Options.....	470
InstallShield Pro - Std Edition.....	471
InstallShield Pro - Windows Installer Edition.....	474
InstallShield Universal Installer.....	475
Nullsoft NSIS	477
SetupBuilder Compile Project	478

SetupBuilder Run Installer	483
Setup Factory	486
Windows Installer	488
MSI Generic	488
MSI Merge.....	489
MSI Transform.....	490
MSI Update Properties	491
MSI Validation.....	492
Windows Installer - Install.....	493
Windows Installer - Reinstall.....	494
Windows Installer - Uninstall.....	495
Wise For Windows Installer	496
Wise InstallBuilder/InstallMaster	497
WiX (Windows Installer XML Toolkit)	499
WiX Compiler (candle.exe)	499
WiX Linker (light.exe).....	501
18 Interactive	504
Ask Question Action	505
Beep Action	507
Choose One Action	508
Input Box Action	510
Message Box Action	511
Multi Question	512
Prompt for File or Directory	514
Prompt for Password Action	516
Prompt for Variables Action	518
Enhanced Prompt for Variables Action	520
19 Internet	524
Convert HTML to MHT File	524
FTP Actions	525
FTP Mirror	528
FTP Upload.....	535
FTP Client (deprecated)	541
HTTP Get Action	545
NNTP News Post Action	548
Send Email (SMTP)	549
Send ICQ Message Action	555
Send MSN Message Action	556
Telnet Client Action	560
Web Service Action	561
Web Service Iterator Action	564
WGet Download	566
MSDeploy	568
MSDeploy Sync.....	569
MSDeploy Delete.....	575
MSDeploy Dump.....	576
MSDeploy Get Dependencies.....	576
MSDeploy Get System Info.....	578
SSH	580
SSH Open Connection.....	580
SSH Close Connection.....	581
SSH Execute Commands.....	582
SSH Batch Execute.....	583
SFTP	585
SFTP Connect.....	585
SFTP Disconnect.....	587
SFTP Check if Connected.....	587
SFTP Create Directory.....	588

SFTP Upload FileSet.....	589
SFTP Upload File.....	590
SFTP Upload Files.....	591
SFTP Download File.....	592
SFTP Download Files.....	593
SFTP Remove File.....	594
SFTP Remove Files.....	595
SFTP Remove Directory.....	596
SFTP Rename File.....	598
SFTP List Files.....	599
SFTP Query Available Space.....	600
SFTP Query Home Directory.....	601
SFTP Copy Remote File.....	603
FTPS Actions	604
FTPS Change Directory.....	604
FTPS Change Up Directory.....	605
FTPS Check If Connected.....	606
FTPS Connect.....	607
FTPS Create Directory.....	611
FTPS Delete File.....	612
FTPS Disconnect.....	613
FTPS Download File.....	614
FTPS Download Files.....	615
FTPS Get File Date.....	616
FTPS Get File Size.....	617
FTPS Get Remote Checksum.....	618
FTPS Get Remote Directory.....	619
FTPS List Directory.....	620
FTPS No-op.....	622
FTPS Remove Directory.....	623
FTPS Rename File.....	624
FTPS Send Command.....	625
FTPS Upload Files.....	626
EC2 Actions	628
EC2 Get Instance Properties.....	628
EC2 Change Instance States.....	629
EC2 Run Instances.....	630
EC2 Terminate Instance.....	632
EC2 Get Image Properties.....	633
EC2 Create Image.....	635
EC2 Deregister Image.....	636
EC2 Get Volume Properties.....	637
EC2 Attach Volume.....	639
EC2 Detach Volume.....	640
EC2 Create Volume.....	641
EC2 Delete Volume.....	642
EC2 Get Snapshot Properties.....	643
EC2 Create Snapshot.....	645
EC2 Delete Snapshot.....	646
EC2 Get Availability Zones.....	647
20 Internet Information Services (IIS)	649
IIS 5	650
Server Selection.....	650
Website Selection.....	651
Backup IIS Metabase Action.....	652
Create Virtual Directory Action.....	653
Create Web Site Action.....	656
Delete Virtual Directory Action.....	659

Delete Web Site Action.....	660
Restore IIS Metabase Action.....	660
Start / Stop / Pause Website Actions	661
Start / Stop / Restart WWW Service Actions	661
Unload IIS Application Action.....	661
IIS 6	663
Server Selection.....	663
Website Selection.....	665
Backup IIS Metabase Action.....	666
Create Application Pool.....	668
Create Virtual Directory Action.....	669
Create Web Site Action.....	671
Delete Application Pool.....	674
Delete Virtual Directory Action.....	675
Delete Web Site Action.....	676
Restore IIS Metabase Action.....	676
Start / Stop / Pause Website Actions	677
Start / Stop / Restart WWW Service Action.....	677
Unload IIS Application Action.....	677
IIS 6 FTP	679
Server Selection.....	679
FTP Site Selection.....	680
Create FTP Directory Action.....	681
Create FTP Site Action.....	684
Delete FTP Directory Action.....	689
Delete FTP Site Action.....	690
Start / Stop / Pause FTP Site Actions	690
Start / Stop / Restart FTP Service Actions	690
IIS 7	691
Server Selection.....	691
Website Selection.....	693
Create Website Action.....	694
Delete Website Action.....	696
Start / Stop / Restart Website.....	696
Create Application.....	696
Delete Application.....	697
Create Virtual Directory.....	698
Delete Virtual Directory.....	699
Create Application Pool.....	700
Delete Application Pool.....	701
Start / Stop / Recycle Application Pool.....	702
21 Iterators	704
File Contents Iterator	706
File/Fileset Iterator	707
Folder Iterator	709
INI File Iterator	710
List Iterator	711
22 Licensing Tools	713
Armadillo Action	713
ASProtect Action	714
ProActivate Action	716
23 Localization Tools	717
Sisualizer	717
24 Misc Actions	719
CityDesk Action	719
Comment Action	720
Encrypt ActionPackage	721

Generate Random Number	722
Get DateTime Action	723
Image File Manipulation	725
Image Manipulation.....	725
Lossless JPEG Transform.....	729
Mutex/Lock File Action	732
Run Script Action	735
Sort Text List Action	736
Spell Checking	738
Text Find / Replace Action	741
Simple Maths Action	747
25 .NET Actions	748
.Net Framework Tools	748
NGen 2.0 Native Image Install Action.....	748
NGen 2.0 Native Image Uninstall Action.....	750
NGen 2.0 Native Images Update Action.....	752
Precompile ASP.NET 2.0 Application.....	753
Register Assembly in COM [REGASM].....	756
Run AL.EXE.....	757
Run ASPNET_REGIIS.EXE.....	758
Serviced Components Installation Tool [REGSVCS].....	759
.Net SDK Tools	762
Extract Public Key [SN].....	762
GAC Download Cache [GACUTIL].....	763
GAC Install [GACUTIL].....	763
GAC Uninstall [GACUTIL].....	764
Generate Key Pair [SN].....	766
Install Key in Container [SN].....	766
Manifest Generation and Editing [MAGE].....	767
Re-sign Assembly [SN].....	771
Run ResGen.exe.....	771
Run SN.EXE.....	772
Type Library Export [TLBEXP].....	773
Type Library Import [TLBIMP].....	774
Verify Strong Name [SN].....	775
3rd Party Tools	777
Demeanor.....	777
Dotfuscator.....	777
FxCop	778
XenoCode.....	778
NCover	780
NCoverExplorer.....	783
NCover Reporting.....	785
NuGet	791
NuGet Delete.....	791
NuGet Install.....	792
NuGet Pack.....	794
NuGet Push.....	795
NuGet Spec.....	796
Other	797
Fix TLBImp Project Reference.....	797
ILMerge	798
VS.NET Setup & Deployment Project Updater	800
.NET Framework Options	803
26 Network	804
Check If Host Exists	804
27 PDF Files	806

Convert PDF	806
Convert to PDF	807
PDF Insert Text	808
PDF Merge Files	810
28 Property Sets	811
Property Set Definitions	813
Win32 Property Sets.....	813
.Net Property Sets	814
VB Property Sets.....	814
Accessing PropertySets	816
PropertySet define	817
PropertySet assign properties	819
PropertySet increment value	820
PropertySet load from AssemblyInfo	821
PropertySet load from FBD	823
PropertySet load from INI file	824
PropertySet load from Registry	825
PropertySet log values	826
PropertySet save to FBD	827
PropertySet save to INI file	828
PropertySet save to Registry	829
PropertySet to Variable	830
29 Source Code Tools	832
CodeHealer	832
Pascal Analyzer	834
NDepend	836
30 Stack and Queue Actions	838
Stack/Queue Define	839
Stack/Queue Clear	841
Stack/Queue Get Count	842
Stack/Queue Insert Item	843
Stack/Queue Is Empty	844
Stack/Queue Iterator	845
Stack/Queue Log Items	847
Stack/Queue Peek Item	849
Stack/Queue Pop Item	850
Stack/Queue Push Item	851
31 String Manipulation	852
CSV Field Iterator	852
Path Manipulation	854
String Add Breaks	856
String Casing	858
String Concatenation	860
String Encryption	862
String Padding	864
String Quoting	866
String Pos	868
String Replace	870
String Reverse	872
String SubString	874
String Trimming	876
32 Testing Tools	878
AQTest	878
AutomatedQA TestComplete 2.x/3.x Actions	880
AutomatedQA TestComplete 4 and onwards	881
MSTest	883

NUnit Action	887
Run QA Wizard Batch File	889
MbUnit Action	891
33 Variables	894
Append to Variable Action	894
Define Variable Action	895
If Variable Defined Action	897
Load Variables From Ini	899
Log Variable Values	902
Replace Variables	904
Reset Variable to Default	906
Save Variables To Ini	907
Set Variable Action	909
34 Version Control Actions	911
AccuRev	911
AccuRev Add.....	913
AccuRev Add a Change Package Entry.....	914
AccuRev Change User.....	915
AccuRev Check Out.....	916
AccuRev Create Snapshot.....	917
AccuRev Create Stream.....	918
AccuRev Defunct.....	919
AccuRev Generic.....	920
AccuRev Keep.....	921
AccuRev Lock.....	922
AccuRev Promote.....	923
AccuRev Purge.....	924
AccuRev Remove.....	925
AccuRev Remove a Change Package Entry	926
AccuRev Unlock.....	927
AccuRev Update.....	928
Borland StarTeam Actions	929
StarTeam Add Files Action.....	931
StarTeam Apply Label Action.....	931
StarTeam Check In Action.....	931
StarTeam Check Out Action.....	931
StarTeam Create Label Action.....	931
StarTeam Delete Files Action.....	932
StarTeam Generic Action.....	932
StarTeam List Files Action.....	932
StarTeam Lock/Unlock Files Action.....	932
StarTeam Update Status Action.....	932
ClearCase	933
Base ClearCase.....	933
ClearCase Apply Attribute Action.....	933
ClearCase Apply Label Action.....	934
ClearCase Check In Action.....	935
ClearCase Check Out Action.....	936
ClearCase Find Checkouts Action.....	937
ClearCase Get Config Spec Action.....	938
ClearCase Lock Action.....	939
ClearCase Make Attribute Type Action.....	940
ClearCase Make Element Action.....	942
ClearCase Make Label Type Action.....	943
ClearCase Run Cleartool (Generic) Action.....	945
ClearCase Set Config Spec Action.....	946
ClearCase Undo Checkouts Action.....	947
ClearCase Unlock Action.....	947

ClearCase Update Snapshot View Action.....	948
ClearCase Object Selector	949
ClearCase Pathnames	951
UCM	953
UCM Change Baseline.....	953
UCM Change Stream.....	954
UCM Check In Files For Activity Action.....	956
UCM Make Activity Action.....	957
UCM Make Baseline Action.....	958
UCM Make Baseline from Label Action.....	960
UCM Set Current Activity Action.....	961
UCM Undo Checkouts for Activity Action.....	961
CVS Actions	963
CVS Command Action.....	964
Git	965
Git Options.....	965
Git Add Files To Repository.....	967
Git Checkout.....	970
Git Clone Repository.....	972
Git Commit.....	974
Git Create Branch.....	978
Git Create Repository	980
Git Delete Branch.....	983
Git Fetch.....	984
Git Generic.....	987
Git List Branches.....	988
Git Merge	990
Git Move	993
Git Pull	995
Git Push	998
Git Remove Files.....	1001
Git Rename.....	1003
Git Repository Status.....	1005
Git Reset.....	1006
Git Show Commit Logs	1009
JediVCS	1013
Scripting JediVCS actions	1013
Mercurial	1016
Choosing Repositories.....	1017
Specifying Files	1020
Mercurial Add.....	1020
Mercurial Annotate.....	1020
Mercurial Archive.....	1022
Mercurial Branch.....	1023
Mercurial Bundle.....	1023
Mercurial Cat.....	1025
Mercurial Clone.....	1025
Mercurial Commit.....	1027
Mercurial Copy.....	1028
Mercurial Create Repository.....	1029
Mercurial Export.....	1030
Mercurial Forget.....	1031
Mercurial Generic.....	1032
Mercurial Heads.....	1032
Mercurial Import.....	1033
Mercurial Log.....	1034
Mercurial Manifest.....	1034
Mercurial Merge.....	1034

Mercurial Parents.....	1035
Mercurial Pull.....	1036
Mercurial Push.....	1036
Mercurial Remove.....	1038
Mercurial Revert.....	1038
Mercurial Rollback.....	1039
Mercurial Status.....	1039
Mercurial Summary.....	1041
Mercurial Tag.....	1041
Mercurial Tip.....	1042
Mercurial Unbundle.....	1042
Mercurial Update.....	1043
Mercurial Verify.....	1044
MKS Source	1045
MKS Add.....	1046
MKS Add Label.....	1047
MKS Check In.....	1048
MKS Check Out.....	1049
MKS Checkpoint.....	1050
MKS Close Change Package.....	1051
MKS Connect.....	1052
MKS Create Change Package.....	1053
MKS Create Sandbox.....	1054
MKS Delete Label.....	1055
MKS Disconnect.....	1056
MKS Drop.....	1057
MKS Generic.....	1058
MKS Lock.....	1059
MKS Open Change Package.....	1060
MKS Project Add.....	1061
MKS Project Check In.....	1062
MKS Resynchronize.....	1063
MKS Unlock.....	1064
Perforce	1065
Perforce Add Files.....	1065
Perforce Create Branch.....	1065
Perforce Create Changelist.....	1066
Perforce Create Label.....	1067
Perforce Delete Branch.....	1067
Perforce Delete Changelist.....	1068
Perforce Delete Label.....	1068
Perforce Generic.....	1068
Perforce Labelsync.....	1069
Perforce Lock.....	1070
Perforce Old Actions.....	1070
Perforce Synchronise with View Action (Deprecated).....	1070
Perforce Command Action (Deprecated).....	1072
Perforce open for Delete.....	1072
Perforce open for Edit.....	1073
Perforce Opened.....	1074
Perforce Options.....	1074
Override Perforce Global Options.....	1075
Perforce Revert.....	1075
Perforce Submit.....	1076
Perforce Sync.....	1076
Perforce Tag.....	1077
Perforce Unlock.....	1077
Perforce Update Label.....	1078

Perform Login Action.....	1078
Plastic SCM	1080
Plastic SCM Options.....	1080
Plastic SCM Add New Items	1081
Plastic SCM Check In Items.....	1083
Plastic SCM Check Out Items.....	1085
Plastic SCM Configure Workspace.....	1087
Plastic SCM Create New Branch.....	1089
Plastic SCM Create Workspace.....	1090
Plastic SCM Generic	1091
Plastic SCM Label Workspace.....	1093
Plastic SCM Update.....	1093
PureCM	1095
PureCM Options.....	1095
PureCM Add File.....	1096
PureCM Checkout File.....	1097
PureCM Delete File.....	1097
PureCM Revert File.....	1098
PureCM Submit.....	1099
PureCM Create Workspace	1100
PureCM Update Workspace.....	1102
PureCM Remove Workspace.....	1103
PureCM Create Stream.....	1104
PureCM Delete Stream.....	1105
PureCM Create Stream Folder	1106
PureCM Delete Stream Folder.....	1107
PureCM Lock Stream.....	1107
PureCM Unlock Stream.....	1108
PureCM Generic.....	1110
PVCS	1111
PVCS Add Label.....	1111
PVCS Get.....	1112
PVCS List Files	1113
PVCS Promote	1114
PVCS Put.....	1115
QSC Team Coherence Actions	1116
Team Coherence Attach Label Action.....	1118
Team Coherence Check In Action.....	1118
Team Coherence Check Out Action.....	1119
Team Coherence Connect Action.....	1121
Team Coherence Create Label Action.....	1122
Team Coherence Create View	1123
Team Coherence Delete View	1124
Team Coherence Detach Label Action.....	1124
Team Coherence Generic.....	1125
Team Coherence Get Action.....	1126
Team Coherence Promote Action.....	1126
Team Coherence Set View Action.....	1128
Team Coherence Sync.....	1128
Team Coherence Update View	1129
QVCS Actions	1130
QVCS Add File Action.....	1130
QVCS Check In File(s) Action.....	1130
QVCS Check Out File(s) Action.....	1130
QVCS Get Latest Version Action.....	1131
QVCS Labels File(s) Action.....	1131
QVCS Undo Check Out File(s) Action.....	1131
Seapine Surround SCM	1132

Surround SCM CheckIn.....	1132
Surround SCM CheckOut.....	1133
Surround SCM Checkout Report.....	1134
Surround SCM Create Branch.....	1137
Surround SCM Freeze Branch.....	1138
Surround SCM Generic.....	1138
Surround SCM Get.....	1139
Surround SCM Global Options.....	1141
Surround SCM Override Global Options.....	1142
Surround SCM Label.....	1143
Surround SCM Remove Working Directory.....	1145
Surround SCM Set Working Directory.....	1146
Surround SCM Unfreeze Branch.....	1147
Source Safe	1148
Source Safe Add Files Action.....	1148
Source Safe Branch.....	1149
Source Safe Check File Status.....	1151
Source Safe Check In File(s) Action.....	1152
Source Safe Check Out File(s) Action.....	1154
Source Safe Get Latest Version Action.....	1155
Source Safe Get Working Directory.....	1156
Source Safe Label File(s).....	1156
Source Safe Override Global Options.....	1157
Source Safe Project Checkouts.....	1158
Source Safe Share.....	1159
Source Safe Undo CheckOut Action.....	1160
Source Safe Create Action.....	1161
SourceGear Vault Actions	1163
Vault Add Action.....	1165
Vault Branch Action.....	1165
Vault Check In Action.....	1166
Vault Check Out Action.....	1168
Vault Cloak Action.....	1169
Vault Commit Action.....	1170
Vault Create Folder Action.....	1171
Vault Create Label Action.....	1172
Vault Delete File/Folder Action.....	1173
Vault Diff.....	1174
Vault File Status.....	1174
Vault Get Action.....	1175
Vault Get Label.....	1177
Vault Get using Wildcards Action.....	1177
Vault Get Version Action.....	1179
Vault GetLabelDiffs.....	1180
Vault ListCheckouts.....	1180
Vault Move File/Folder Action.....	1182
Vault Pin File/Folder Action.....	1183
Vault Rename File/Folder Action.....	1184
Vault Set Working Folder.....	1185
Vault Share File/Folder Action.....	1186
Vault UnCloak Action.....	1187
Vault Undo Checkout Action.....	1187
Vault UnPin File/Folder Action.....	1189
Subversion Actions	1190
Subversion Add.....	1195
Subversion Checkout.....	1196
Subversion Cleanup.....	1197
Subversion Commit.....	1197

Subversion Copy.....	1198
Subversion Delete.....	1200
Subversion Export.....	1201
Subversion Generic Action.....	1202
Subversion Import.....	1203
Subversion Info.....	1204
Subversion Mkdir.....	1205
Subversion Revert.....	1206
Subversion Status.....	1207
Subversion Switch.....	1209
Subversion Update.....	1210
Team Foundation Source Control	1211
Options	1211
Team Foundation Add.....	1212
Team Foundation Apply Label.....	1213
Team Foundation Branch.....	1214
Team Foundation Check In Pending Changes.....	1216
Team Foundation Check Out.....	1217
Team Foundation Delete.....	1219
Team Foundation Delete Shelveset.....	1220
Team Foundation Get.....	1220
Team Foundation Lock.....	1223
Team Foundation Map Working Folder.....	1224
Team Foundation Merge.....	1225
Team Foundation Remove Label.....	1226
Team Foundation Rename/Move.....	1227
Team Foundation Shelve Pending Changes	1229
Team Foundation Undelete.....	1231
Team Foundation Undo Pending Changes.....	1232
Team Foundation Unlock.....	1233
Team Foundation Unmap Working Folder	1234
Team Foundation Unshelve Pending Changes	1234
Team Foundation Create Workspace.....	1236
Team Foundation Delete Workspace.....	1237
Team Foundation Work Items.....	1238
Team Foundation Query Work Items	1242
Team Foundation Create Work Item.....	1243
Team Foundation Edit Work Item(s).....	1245
Team Foundation Copy Work Item(s)	1245
Bazaar	1247
Bazaar Init.....	1248
Bazaar Add.....	1249
Bazaar Ignore.....	1250
Bazaar Move.....	1251
Bazaar Rename.....	1252
Bazaar Status.....	1253
Bazaar Merge.....	1254
Bazaar Checkout.....	1255
Bazaar Commit.....	1256
Bazaar Update.....	1257
Bazaar Branch.....	1258
Bazaar Pull.....	1259
Bazaar Push.....	1260
Bazaar Remove.....	1261
Bazaar Revert.....	1262
Bazaar Whoami.....	1263
Bazaar Remove Branch.....	1264
Bazaar Remove Tree.....	1265

Select Files	1266
35 Virtualization Systems	1267
Microsoft Hyper V	1267
Hyper V Server Page	1268
Hyper V Check VM State	1270
Hyper V Start VM	1270
Hyper V Stop VM	1271
Hyper V Reset VM	1272
Hyper V Pause VM	1273
Hyper V Suspend VM	1274
Hyper V Create VM Snapshot	1275
Hyper V Apply VM Snapshot	1276
Hyper V Delete VM Snapshot	1277
Hyper V List Virtual Machines	1278
Microsoft Virtual Server	1280
Virtual Server Check VM Status	1281
Virtual Server List VMs	1282
Virtual Server Pause VM	1284
Virtual Server Reset VM	1285
Virtual Server Save VM State	1286
Virtual Server Start VM	1287
Virtual Server Turn Off VM	1289
VMWare Server	1290
VMWare Server List VMs	1291
VMWare Server Start / Stop / Reset / Suspend VM	1292
VMWare Workstation	1294
Check Virtual Machine Status	1294
Create Snapshot	1295
Reset Virtual Machine	1296
Revert To Snapshot	1296
Start Virtual Machine	1298
Stop Virtual Machine	1299
Suspend Virtual Machine	1300
VMWare	1301
List Virtual Machine	1302
Change Virtual Machine State	1303
Snapshot	1305
Execute Guest Command	1306
36 Windows OS	1308
Close Process Action	1308
COM+ Administration	1310
COM+ Delete	1310
COM+ Edit Application	1311
COM+ Register	1315
COM+ Shutdown n Application	1317
COM+ Start Application	1318
Control Service Action	1320
Create Shortcut	1322
Execute Powershell Script	1325
Execute Program Action	1327
Generate New GUID	1330
If COM Class Registered	1331
Logical and Network Drives	1333
Get Disk Free Space	1333
Map Network Drive Action	1333
Subst Drive Action	1335
Unmap Network Drive Action	1337
Net Send Message Action	1338

Networking Configuration and Diagnostics	1339
Rasdial VPN / Remote Network Connection	1339
Register DLL/OCX Action	1341
Remote Reboot Action	1342
Remote Shutdown Action	1343
Run DOS Command Action	1344
Shell Execute	1346
Sysinternals PsTools Suite	1347
PsExec (Execute Remote Process)	1348
PsFile (List/Close Remote Files)	1351
PsGetSID (Security ID Info)	1352
PsInfo (View System Information)	1353
PsKill (Kill Process)	1354
PsList (List Processes)	1356
PsLoggedOn (Logged On Users)	1357
PsLogList (List event logs)	1358
PsService (Service Manager)	1362
PsShutdown (Shutdown Computer)	1364
PsSuspend (Suspend/Resume Processes)	1365
Window Exists Action	1367
Windows Scheduler	1368
Add Scheduled Task	1368
Delete Scheduled Task	1371
End Running Scheduled Task	1372
Run Scheduled Task	1373
Scheduling Remote Tasks	1373
WMI Actions	1375
WMI Kill Process Action	1375
WMI Process Info Action	1376
WMI Process Iterator	1377
WMI Run Process Action	1379
37 XML Actions	1381
Altova DiffDog Action	1383
Create XML Node	1384
Delete XML Element/Attribute	1388
Delete XML Nodes	1390
Edit XML File Action	1391
Merge XML Action	1393
Extract XML Fragment Action	1394
Read XML Value to Variable	1396
Transform XML	1398
Validate XML File	1400
XML Document Define	1401
XML Node Exists Action	1403
XML Node Iterator	1405
XML Parser Options	1407
XML Save Document	1409

Part VII Wizards 1410

1 Import Embarcadero Project Wizard	1410
Select Project Group Page	1411
Select Compiler Version Page	1412
Common Settings Page	1413
Version Info Page	1414
Import Options Page	1415
Finish Page	1416
2 Import VB6 Project Group Wizard	1417

Select Project Group	1418
Master Project Selection	1419
Common Output Path Setting	1420
Compiler Settings	1421
Make Settings	1422
Grouping	1423
Finish	1424
3 Embarcadero Developer Studio Project Import Wizard	1425
Select Projects	1426
Common Settings	1427
Common Information	1428
Win32 Property Set	1429
Action Grouping	1430
Finish	1431

Part VIII Automating FinalBuilder 1432

1 FinalBuilder IDE	1433
Scheduling builds	1433
IDE Command Line Interface	1438
Exit Codes.....	1439
2 Command Line version	1440
Exit Codes	1443
3 MSBuild Task	1444
4 Team Foundation Server Integration	1446
Team Build (TFS 2008)	1447
Visual Studio Addin.....	1447
New Build Definition With FinalBuilder.....	1447
Edit FinalBuilder Project(s).....	1450
FinalBuilder Task dialog.....	1452
Team Build Functionality.....	1454
Get Team Foundation Build Parameters Action.....	1455
Windows Workflow (TFS 2010)	1457
Adding FinalBuilder templates to TFS.....	1457
Using TFS data in FinalBuilder	1459
Making your project accessible to TFS.....	1460
Configure and Queue a Build.....	1460

Part IX Tips and Tricks 1462

1 Two Environments	1462
2 Reusing Logic	1465
3 A Configurable build	1470
4 Analysing Output	1475
5 Documenting your Project	1482
6 Counting Errors with Try/Catch Blocks	1486
7 Running Builds in Sequence	1488
8 Using Output Monitors	1492

Part X Reference 1496

1 Regular Expression Reference	1496
2 INI Files	1503

Part XI Support	1504
1 Known Problems	1504
2 FAQ	1506
3 FinalBuilder Support	1507

1 FinalBuilder

1.1 FinalBuilder Overview



Welcome to FinalBuilder™

FinalBuilder provides one tool for automating the build, test and release processes. With FinalBuilder, you can define, debug, maintain, run and schedule a build process for your software.

How does FinalBuilder Work?

FinalBuilder features built-in 'actions' for automating many tasks and interfacing with 3rd party applications that are commonly used in a build process. Most development tools such as compilers, install builders, version controls systems, support some sort of automation interface, for example a command line compiler, or a COM interface. FinalBuilder leverages these interfaces into a consistent and easy to use GUI application. Each of FinalBuilder's actions provide an interface to a third party tool or internal function. Simply chain actions together to create a customised workflow to automate each step in your build, test and release process.

To extend your build process, FinalBuilder also supports scripting and ActionStudio, a custom action authoring tool. Scripting can be used with each action as it exposes events which can be coded in VBScript, JavaScript or PowerShell. ActionStudio, provides an IDE for developing custom actions in any of these languages that you can the plug-in to FinalBuilder.

Make the most of FinalBuilder

FinalBuilder is sophisticated tool which can automate many tasks. To help you leverage FinalBuilder, VSoft encourage you to take the time to read this help file and explore the other resources we have for learning to use FinalBuilder. If you have any questions not answered in this help file, please contact us on support@finalbuilder.com.

1.2 FinalBuilder Server

FinalBuilder Server is bundled with FinalBuilder Professional. FinalBuilder Professional also includes one user license for FinalBuilder Server - additional user licenses need to be purchased separately.

FinalBuilder Server is a web based interface for FinalBuilder that enables centralized management of automated software builds. This allows software developers and members of development teams to easily monitor and control all of their builds.

FinalBuilder Server can start builds in a number of ways:

- On a time schedule (eg. Mon-Fri at 1pm)
- Interactively - a logged in user can start the build at any time
- Triggered - Continuous Integration style by monitoring changes in your version control system

For more information on FinalBuilder Server, please see either the website, <http://www.finalbuilder.com> or the online FinalBuilder Server help file.

1.3 Version History

Version history is available at: <http://www.finalbuilder.com/version-history.aspx>

1.4 License

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

FinalBuilder End User Licence Agreement
Copyright © 2010 VSoft Technologies Pty Ltd
All Rights Reserved.

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: FinalBuilder, FinalBuilder Server and FinalBuilder ActionStudio, herein referred to as the Software.

FinalBuilder also incorporates binary files provided by Microsoft Corporation, herein referred to as IronPython files. These files are contained within the IronPython subdirectory of the FinalBuilder program directory. The IronPython files are explicitly not covered by this EULA. Microsoft provides the IronPython files under the license described in the "License.html" file, found in the IronPython subdirectory.

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 that is treated identically to a physical machine in respect to licensing of the Software.

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. License Types

4.1 Named User/Build Machine License: Under the terms of a Named User/Build Machine License, either one Named User or one Build Machine is permitted to use the Software.

A Named User is a specific individual designated by you to use the Software. The individual may install the software on a desktop development machine and notebook computer; however, they are the only entity, human or machine, permitted to use the software. If more than one entity is required to use the software, then a concurrent license is required.

A Build Machine is a non-human operated device that utilizes the Software, for example, a computer running triggered or scheduled builds. The build machine is the only entity the software may be installed on; however, a single user can remotely login to the machine to use the software. If more than one user is required to log into the machine then a concurrent license is required.

A Named User/Build Machine License may not be used concurrently on different computers or by

different entities. Only Professional edition Named User/Build Machine licenses permit the named user to install and access FinalBuilder Server.

4.2 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. However, the number of users accessing or using the Software simultaneously must not exceed the number of Concurrent User Licenses purchased for such use. Only Professional edition Concurrent User licenses permit you to install and access FinalBuilder Server.

4.3 Site License: Under the terms of a Site License, you may distribute and install an unlimited number of copies of the Software within once specified Site in your organisation. An unlimited number of users may access the Software simultaneously within the specified Site. Only Professional edition Site licenses permit you to install and access FinalBuilder Server.

4.4 Server License: a Server License is included with all Professional edition licenses. Under the terms of a Server License, you may install and use the Server Software on an unlimited number of machines. The Server Software requires a Professional Named, Concurrent or Site license to the Software. User accounts for the Server Software are limited to the number of Server User licenses.

4.5 Server User License: Under the terms of a Server User License, one individual is permitted to use the Server Software through a named account. Each user license is for use by only one specific individual. Licenses can be transferred to a different individual; however, they cannot be used concurrently by multiple individuals.

4.6 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.

4.7 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.

5. License Restrictions

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

5.1 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.

5.2 The Software is licensed as a single product. The Software may not be separated for use on more than one computer.

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

5.4 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.

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

6. License Enforcement and Software Updates

6.1 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.

6.2 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.

7. 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.

8. 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.

9. 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.

If this clause breaches your company policy or you do not wish to be bound by this clause, VSoft are happy to remove it upon request.

10. 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 7488

Email: salesinfo@vsoft-tech.com.au

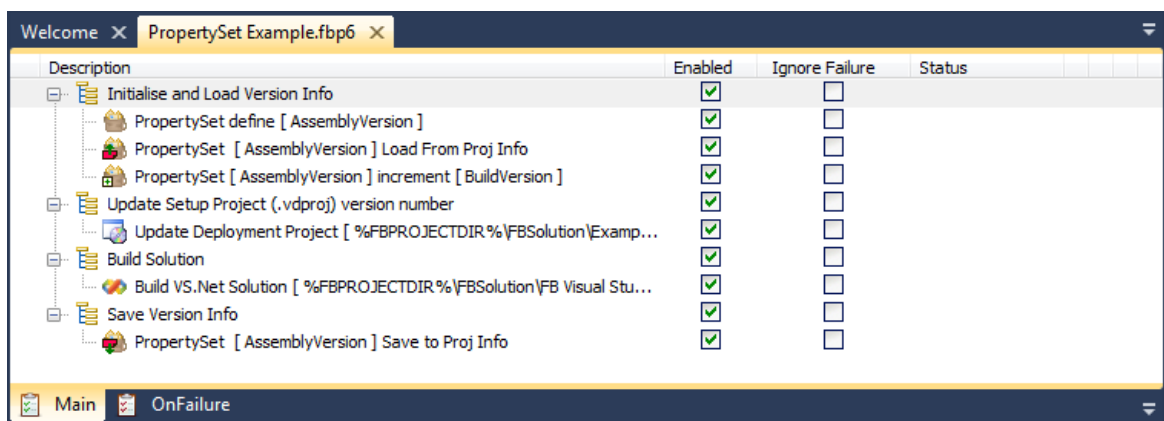
2 Getting Started

2.1 FinalBuilder Basic Concepts

- Use the FinalBuilder IDE to design your build using Actions (see below)
- Test and Debug your build in the IDE
 - Run your build in the IDE or using FBCMD
 - Use FinalBuilder Server for centralized build management and continuous integration

Actions and Action Lists

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



Main and OnFailure

By default each FinalBuilder 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 build 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 build fails, such as deleting temporary or intermediate files etc.

User Action Lists

In addition to the default Action Lists, FinalBuilder 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 FinalBuilder IDE Modes

The FinalBuilder IDE supports three main modes of operation:

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

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

2.3 Upgrading Projects From Previous Versions

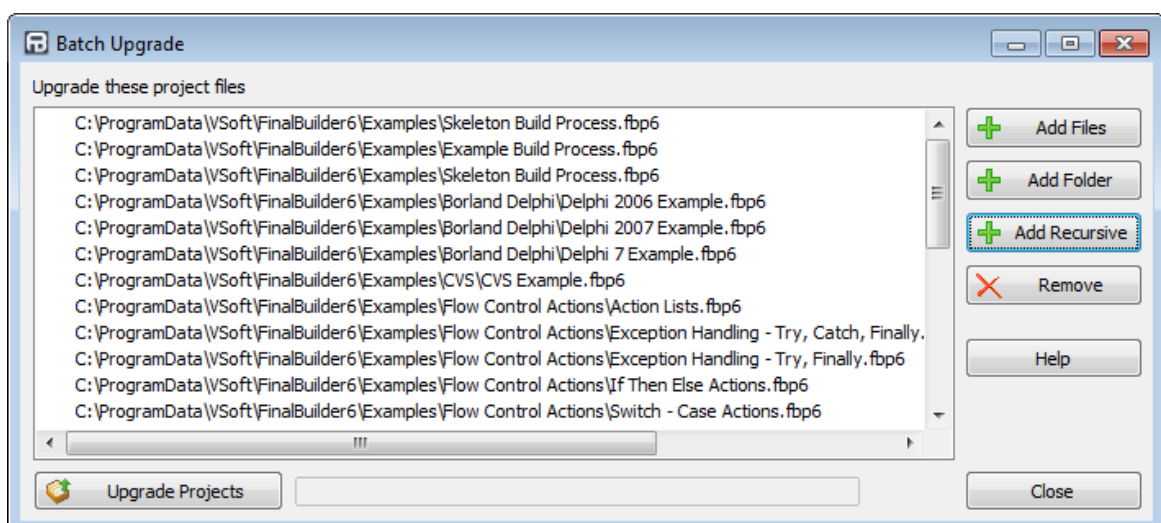
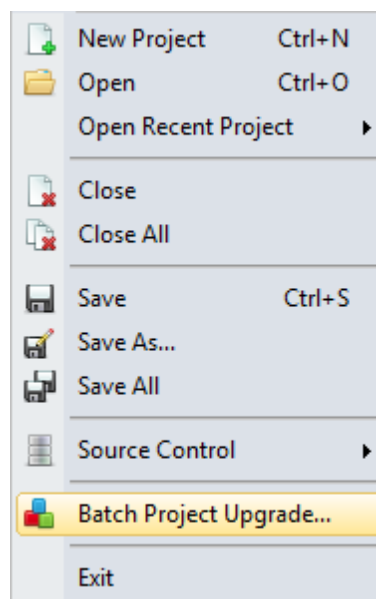
Upgrading One Project File

FinalBuilder can load any project file from any previous version of FinalBuilder. 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.



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 FinalBuilder projects, project files are converted as follows.

Previous file	Saved as
.fbp6	.fbp7
.fbz6	.fbz7
.fbv, .fbw, .fbd	.fbpinf
.fbI5, .lck	(not converted)

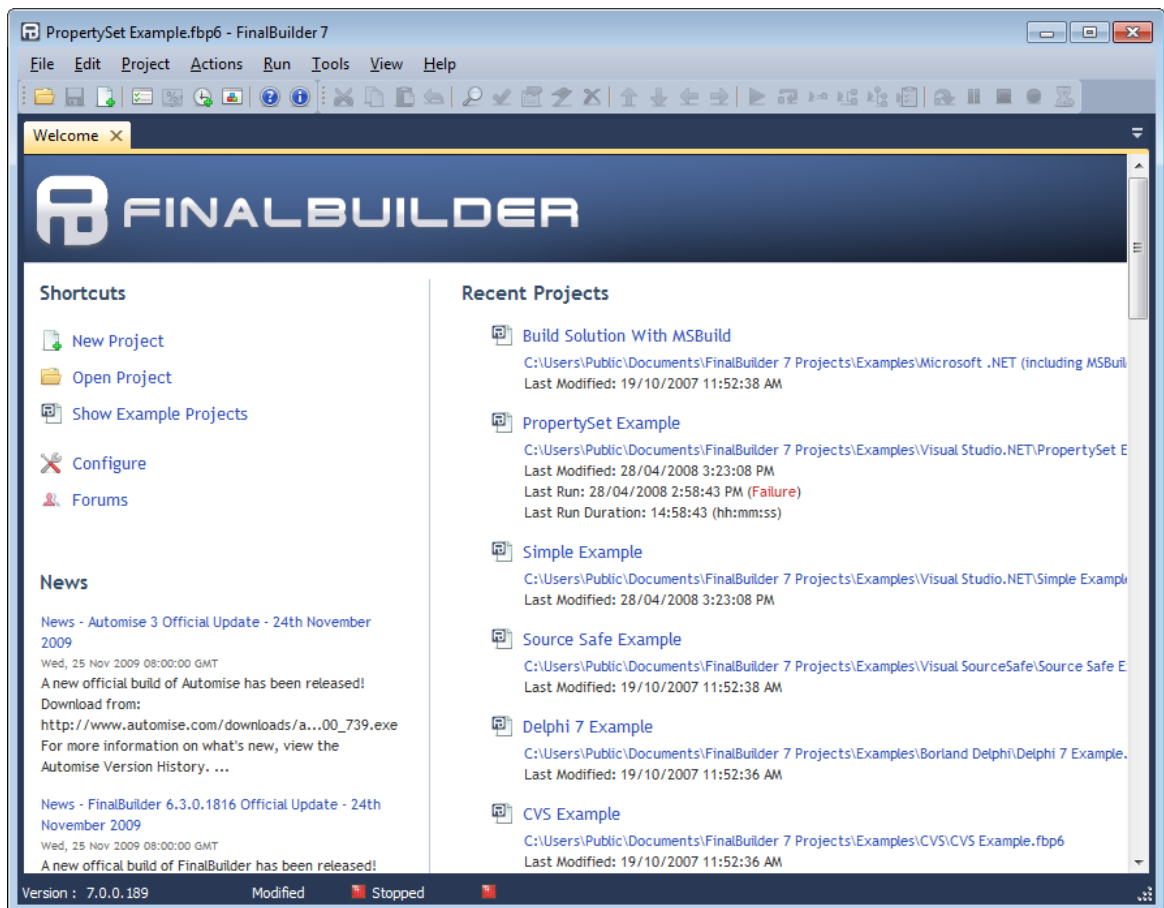
See also

Project Files

3 FinalBuilder IDE Reference

3.1 Welcome Page

The Welcome Page is a kind of portal within FinalBuilder, 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.



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
- FinalBuilder 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 FinalBuilder.

Learn More About

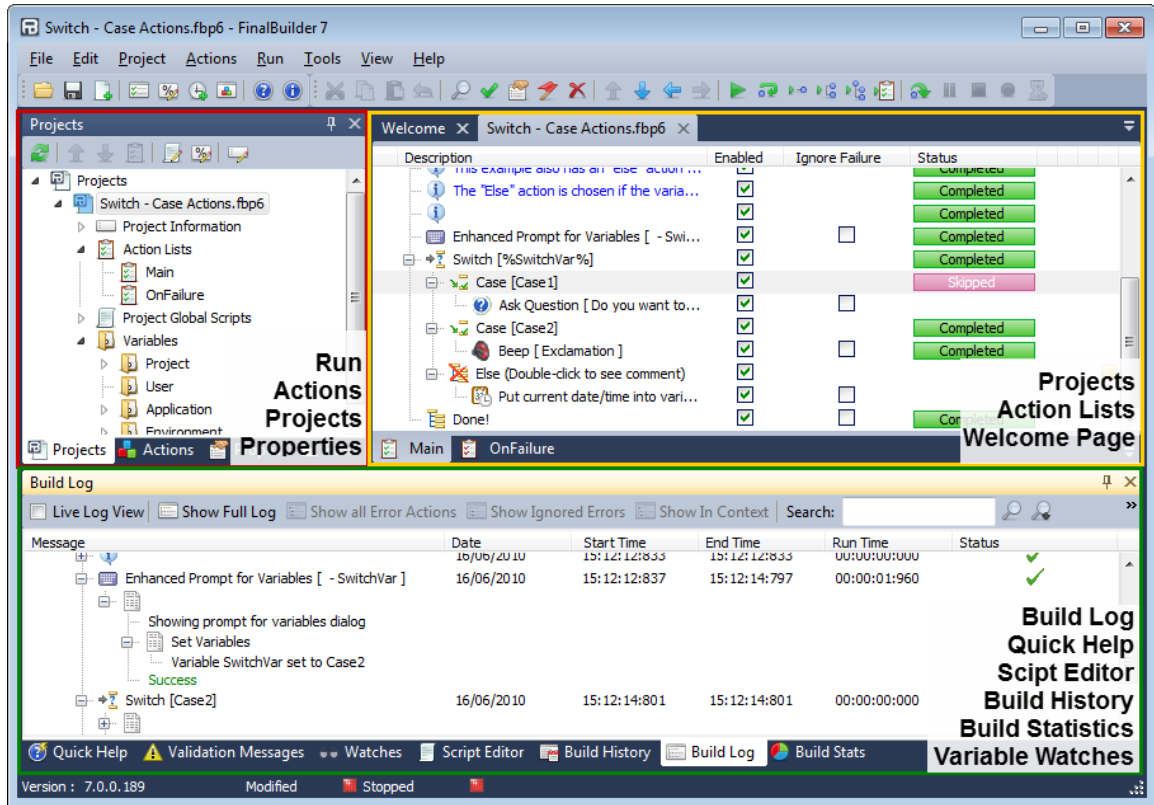
These links point to help on the most important concepts you need to know about. If you're new to FinalBuilder, we suggest having a read.

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

FinalBuilder's Design mode consists of three main sections:



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

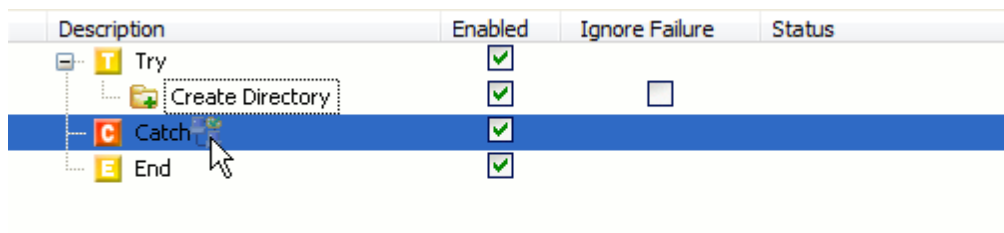
There are several tabs in the bottom section, but the most relevant ones during design of your project are the Script Editor and the Quick Help.

3.2.1 Working with Actions

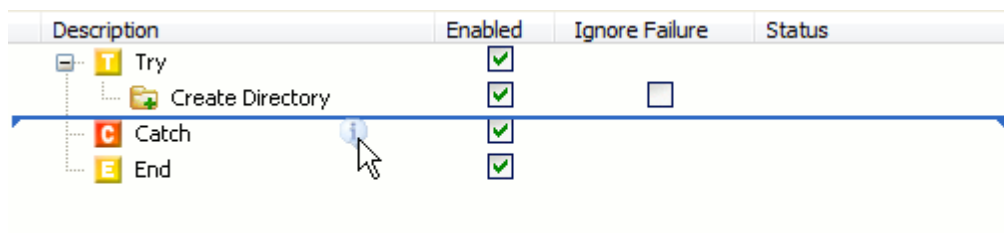
Adding Actions to an Action List

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.

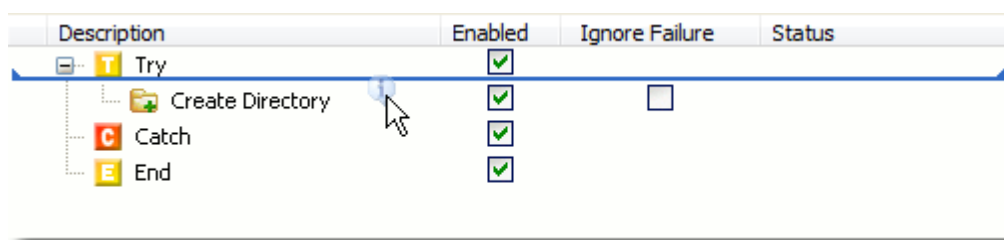
The Action List Tree view provides guide lines to indicate where the action will be dropped. In this example the action will be dropped as a child to the highlighted action, because the mouse is over the icon or name of the action.










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



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



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

Description	Enabled	Ignore Failure	Status
 Try	<input checked="" type="checkbox"/>		
 Comment (Action dropped above line)	<input checked="" type="checkbox"/>		
 Create Directory	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 Comment (Action dropped below line)	<input checked="" type="checkbox"/>		
 Catch	<input checked="" type="checkbox"/>		
 FTP Create Directory	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 End	<input checked="" type="checkbox"/>		

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

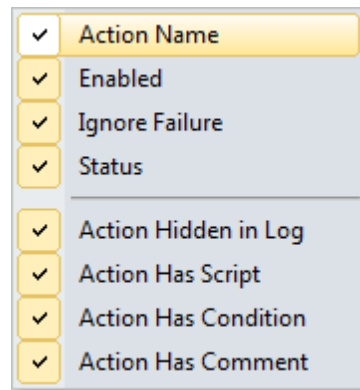
Actions can be copied and pasted using the clipboard, in which case the actions are pasted after the currently selected Action. You can also use Drag and Drop with the Control Key down to copy the dragged Actions.

Deleting Actions

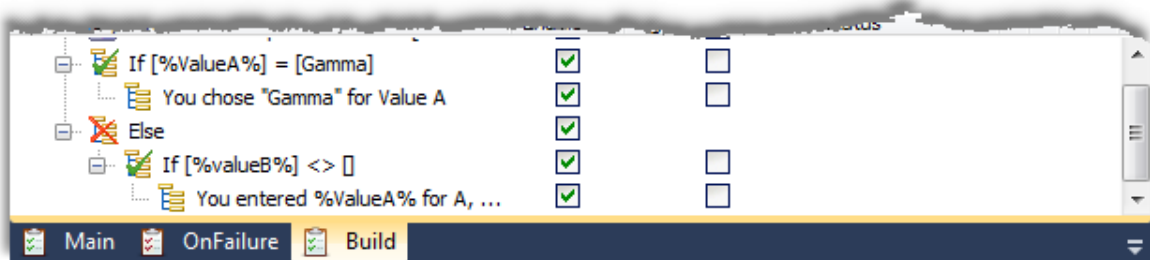
To delete an Action, select it and press the delete key, or use the Delete button on the Actions Toolbar.

Action List Columns

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



Columns:



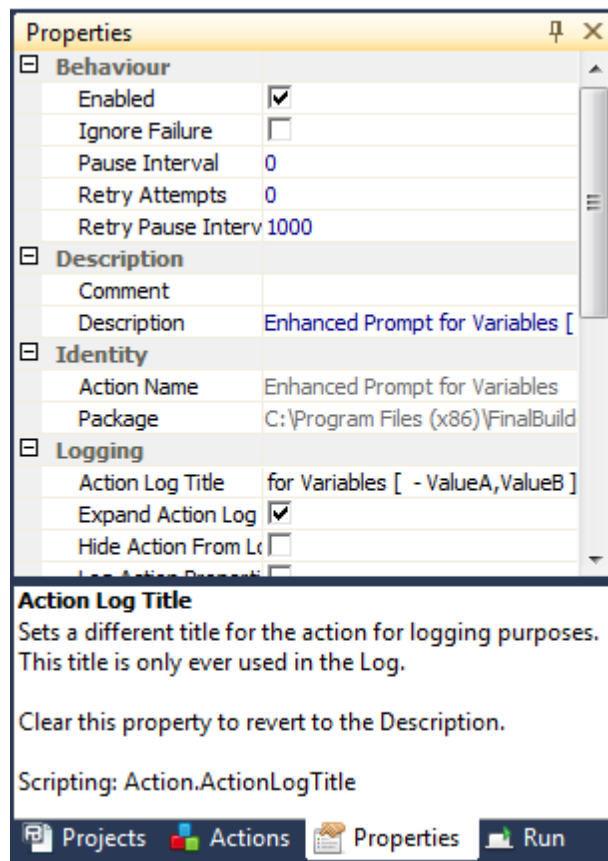
- Action: shows the underlying name of the action, such as "Action Group" or "List Iterator". Useful if you frequently rename actions.
- Enabled: shows a checkbox allowing you to quickly enable or disable actions.
- Ignore Failure: shows a checkbox that specifies whether execution continues even if this 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.

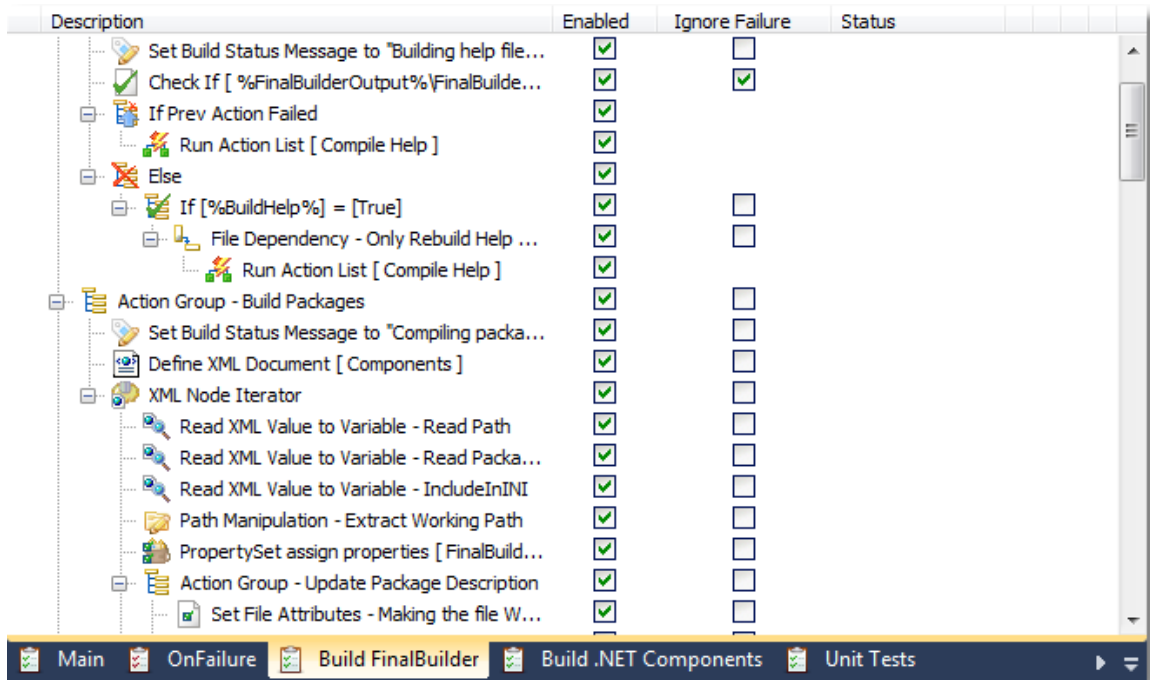
Every action property can also be accessed through scripting. The context-sensitive description at the bottom shows the name

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



3.2.3 Action Lists

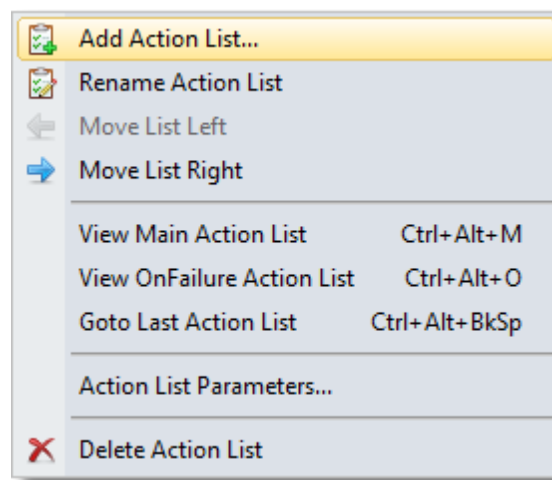
By Default, a new FinalBuilder project has two Action Lists, Main and OnFailure. These default Action Lists cannot be deleted or moved. When a project starts, the first enabled action in the Main Action List is executed. If any action fails, execution switches to the OnFailure Action List, if it contains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



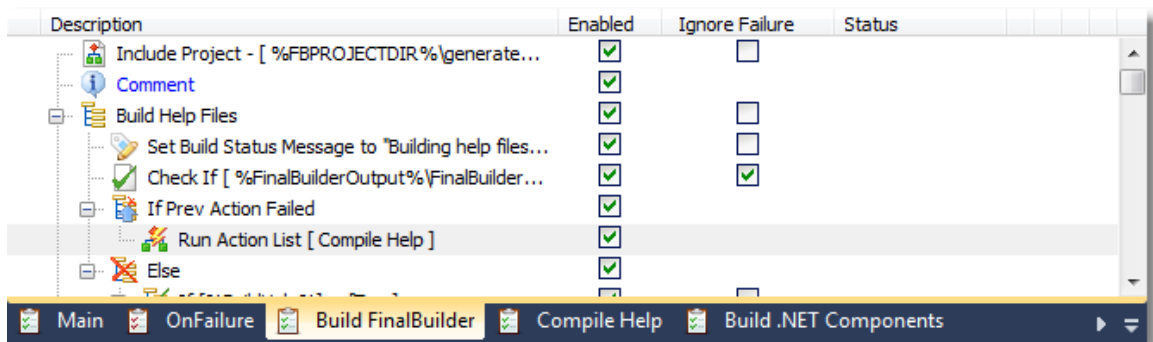
(Tip : You can drag actions from one action list to another by dragging them over the action list tab, then dropping them on the other action list.)

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:



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.



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

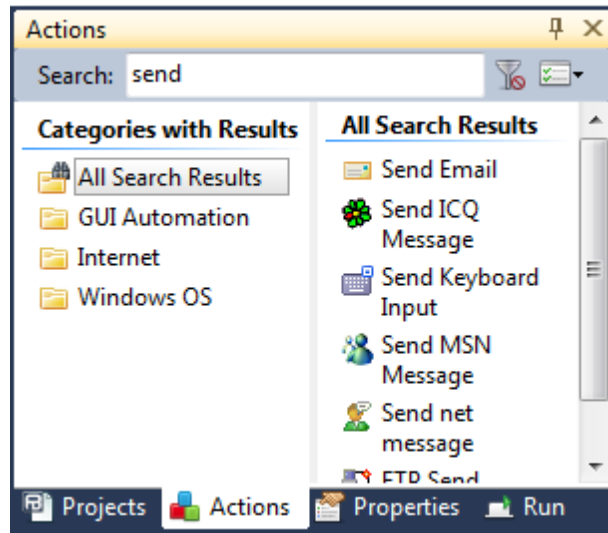
Action Lists can be treated like subroutines - you can call them as often as you require. Through the use of Action List Parameters and FinalBuilder variables, Action Lists can be used like functions or method calls.

See Also

[FinalBuilder 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:



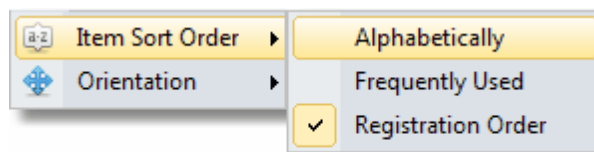
To clear the search filter click on the "Clear Search Field" button or press Escape :



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.

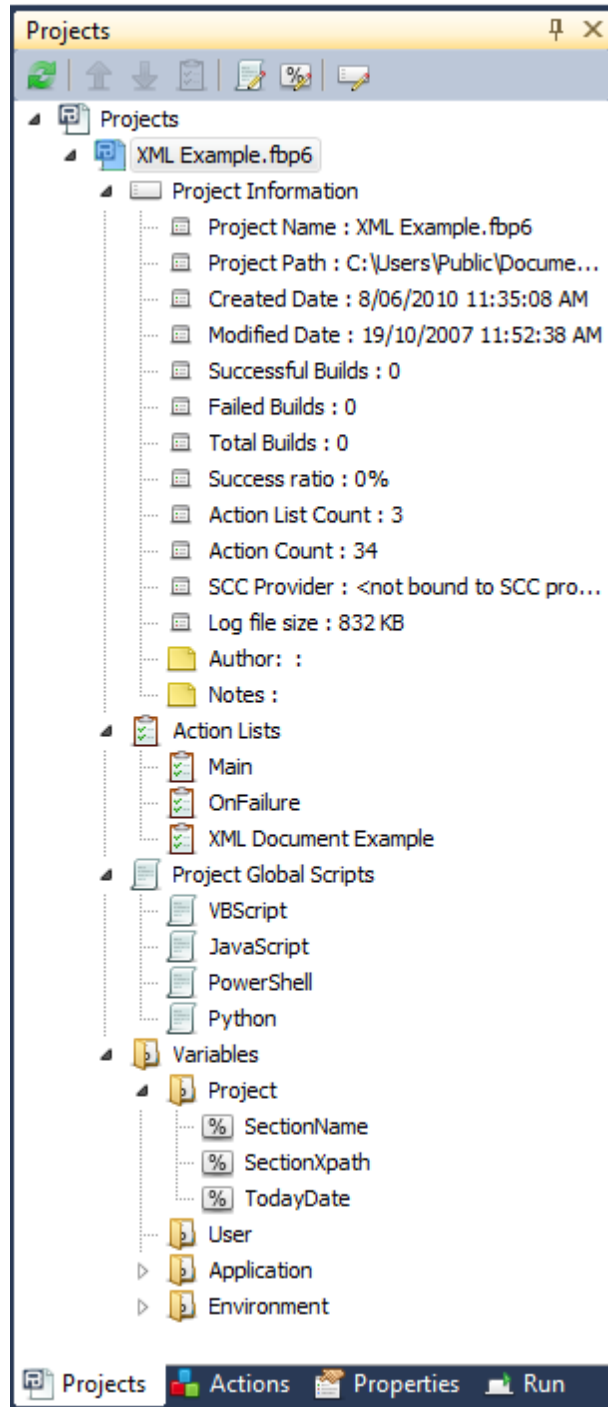
See Also

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



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

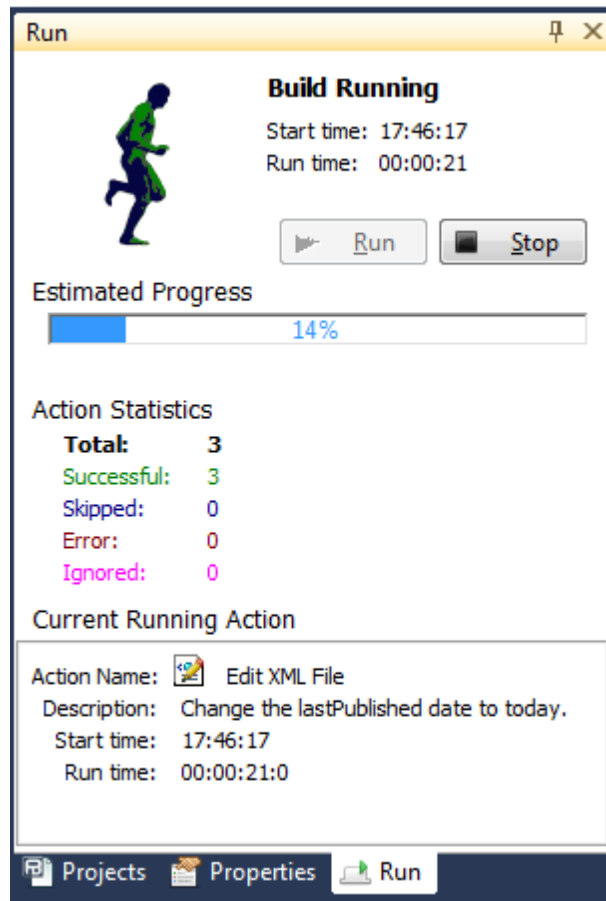
To add a variable, right click Variables and select "Add Variable"

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:



Estimated Progress

The calculation of progress is only an estimate - these are the steps FinalBuilder 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 FinalBuilder 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 BugFix). 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 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 Build 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 BugFix modes. You know (because you've run the Full mode enough times) that Full mode will run 469 actions, and that BugFix runs only 210 actions. Your project will have some logic at the start which figures out if Full or BugFix is required, and using that same logic it can call the script function to set the estimated progress total (in JavaScript):

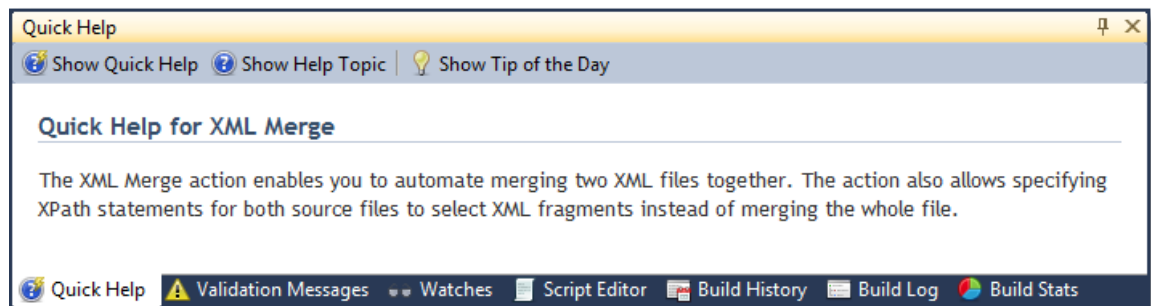
```
if (FullBuild){
    SetEstimatedProgressTotal(469)
} else {
    SetEstimatedProgressTotal(210)
}
```

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"



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 FinalBuilder, 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.

For more information on scripting see [Scripting in FinalBuilder](#).

3.2.9 Keyboard Shortcuts

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

	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 Build 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

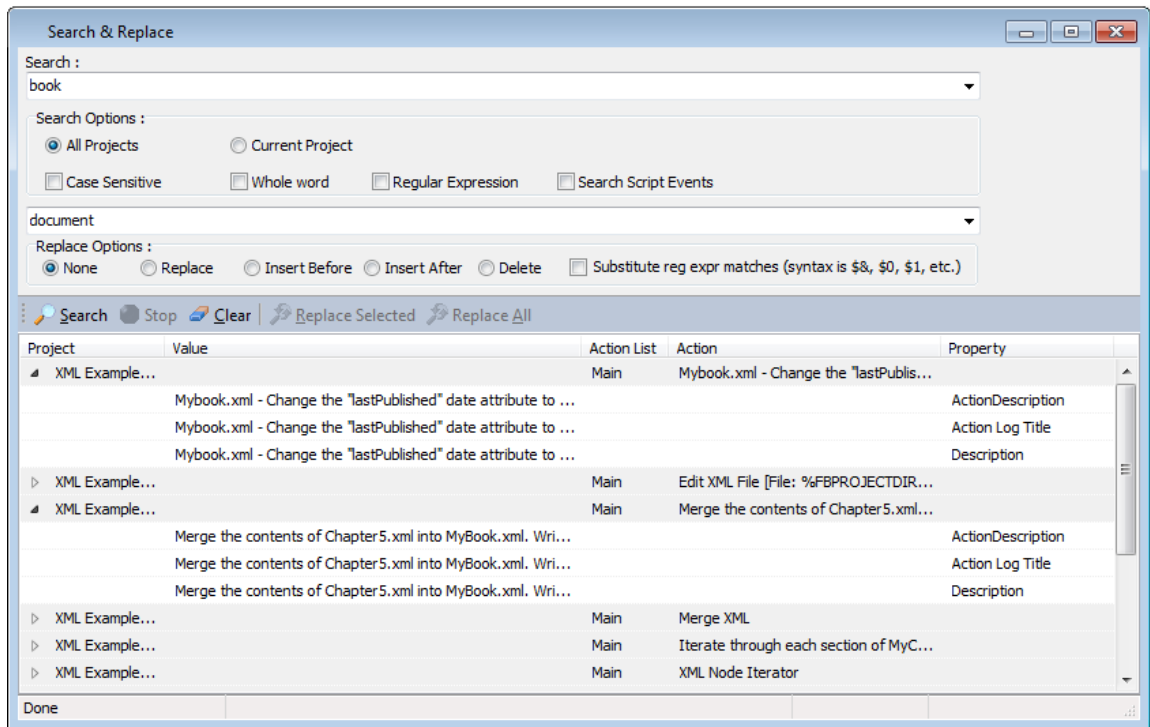
New project	Ctrl+N
Open project	Ctrl+O
Save project	Ctrl+S

Source control:

Get latest version	Shift+Ctrl+G
Check out project	Shift+Ctrl+O
Check in project	Shift+Ctrl+I

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.



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. FinalBuilder 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	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Iteration 1 of 44
List Iterator [Iterator vari...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Do something	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

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:

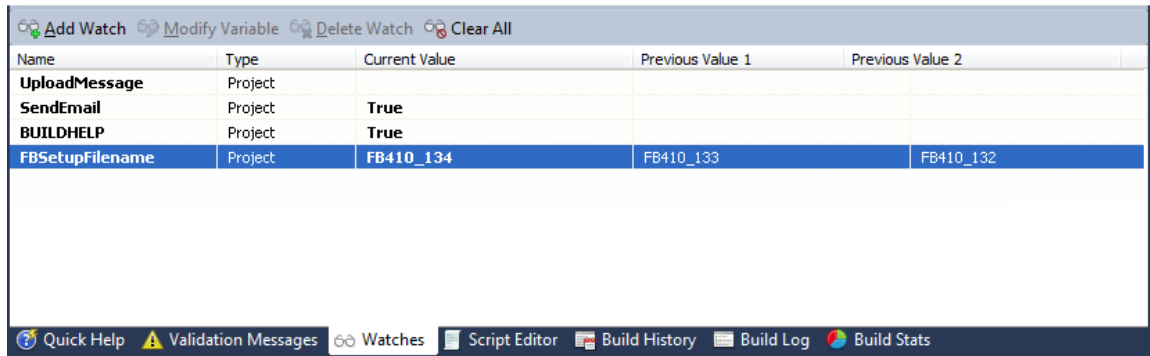
- Click the toolbar icon; or
- Press the F8 (or F10 with the Microsoft style bindings) key; or
- Choose Step from the Run menu.

The screenshot shows the FinalBuilder IDE interface. On the left, the 'Run' window is open, displaying 'Build Paused' with a progress bar at 50%. It includes buttons for 'Run' and 'Stop', and 'Action Statistics' showing 2 total actions, 2 successful, 0 skipped, 0 errors, and 0 ignored. On the right, the 'New Project 1.fbz7' window shows the project actions: 'File Contents Iterator' (21/500), 'List Iterator [Ite...]' (Paused), and 'Run Action ...' (Paused).

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.



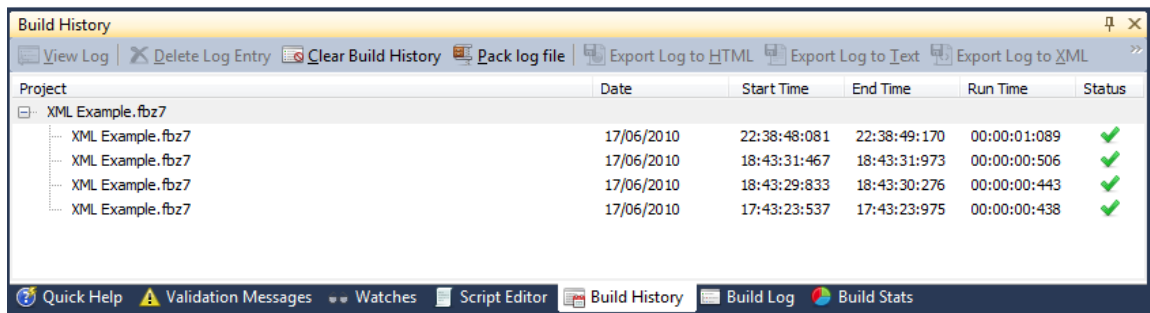
To add a new watch, click on the Add Watch button or drag a variable from the Project tab.

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.

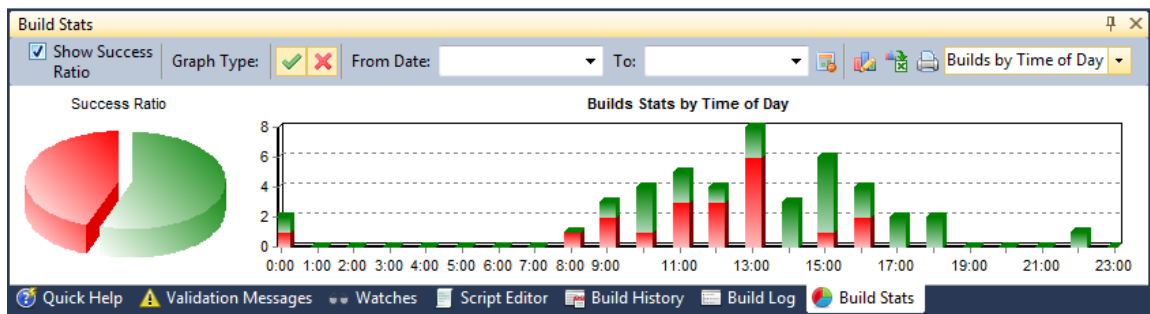
3.4 History & Statistics

The History & Statistics Tabs display a summary of the previous builds of the current project.

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



Project	Date	Start Time	End Time	Run Time	Status
XML Example.fbz7	17/06/2010	22:38:48:081	22:38:49:170	00:00:01:089	✓
XML Example.fbz7	17/06/2010	18:43:31:467	18:43:31:973	00:00:00:506	✓
XML Example.fbz7	17/06/2010	18:43:29:833	18:43:30:276	00:00:00:443	✓
XML Example.fbz7	17/06/2010	17:43:23:537	17:43:23:975	00:00:00:438	✓



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 Build 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.

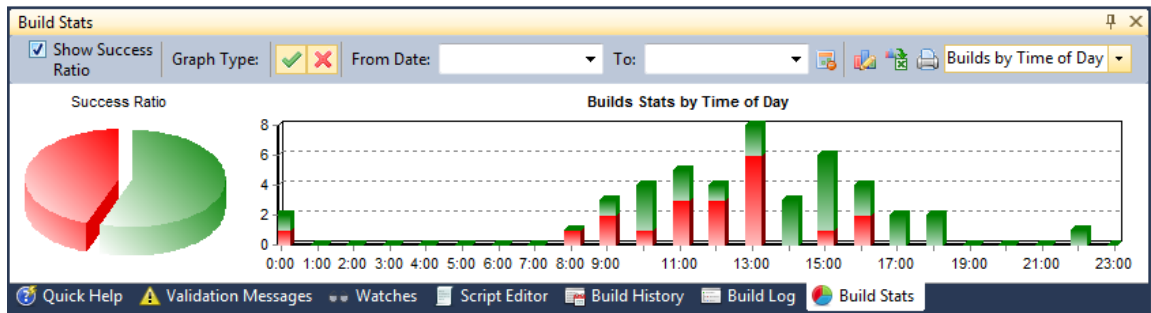
See also: Logging Options

3.4.1 Build Statistics

The build Statistics displays various graphs based on the history of the current project. This is useful if you want some overall metrics and trends for the builds you run. These statistics can be printed out and exported as a picture or raw data to various formats.

The build Statistics are automatically updated after a build completes, and when any setting is changed (eg. changing the graph type).

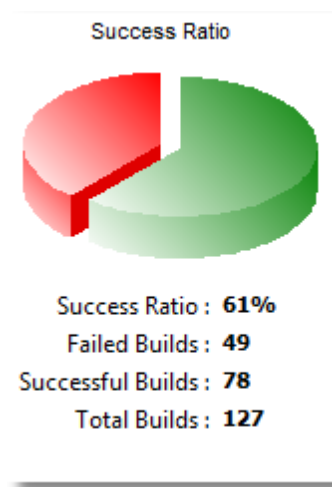
NOTE: Statistics are only available in the Professional Edition of FinalBuilder



Success Ratio Chart
Main Chart Types
Chart Data Options
Chart Visual Options
Exporting Build Statistics
Printing Statistics Chart

3.4.1.1 Success Ratio Chart

Success Ratio Chart



The Success Ratio pie chart display an overall graphical representation of the success ratio of the project for a particular time period. Underneath the success ratio graph is

a total of the failed builds, successful builds, overall total, and the success ratio as a percentage. The success ratio chart cannot be exported or printed.

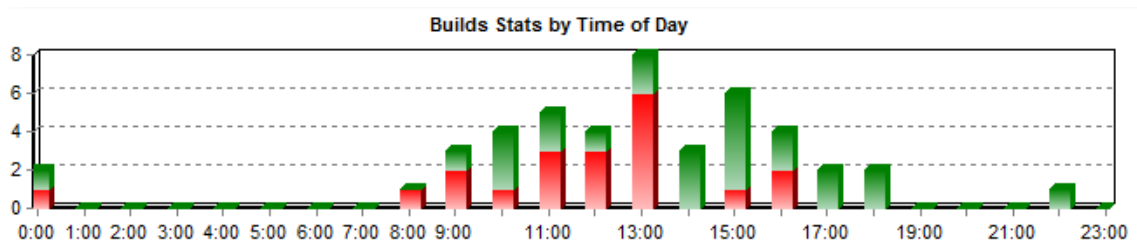
3.4.1.2 Main Chart Types

Main Chart Types

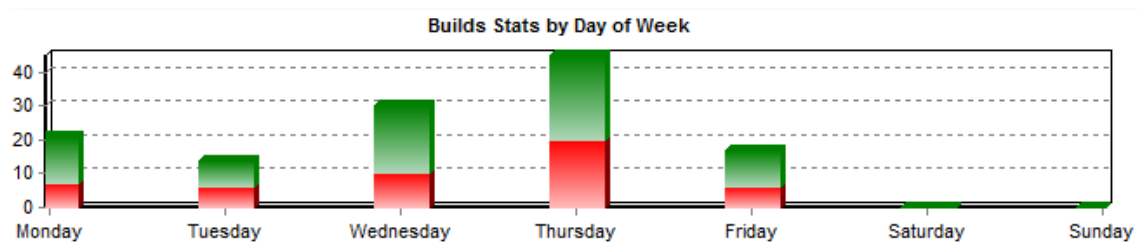
The main build statistics chart can display a wide variety of data. The available graph types are as follows:

- Builds by Time of Day
- Builds by Day of Week
- Builds by Day
- Builds by Week
- Builds by Month

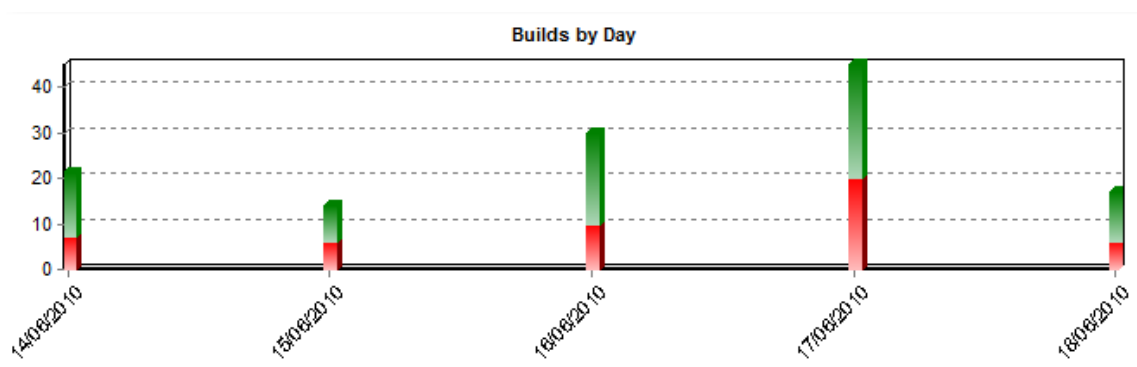
Builds by Time of Day



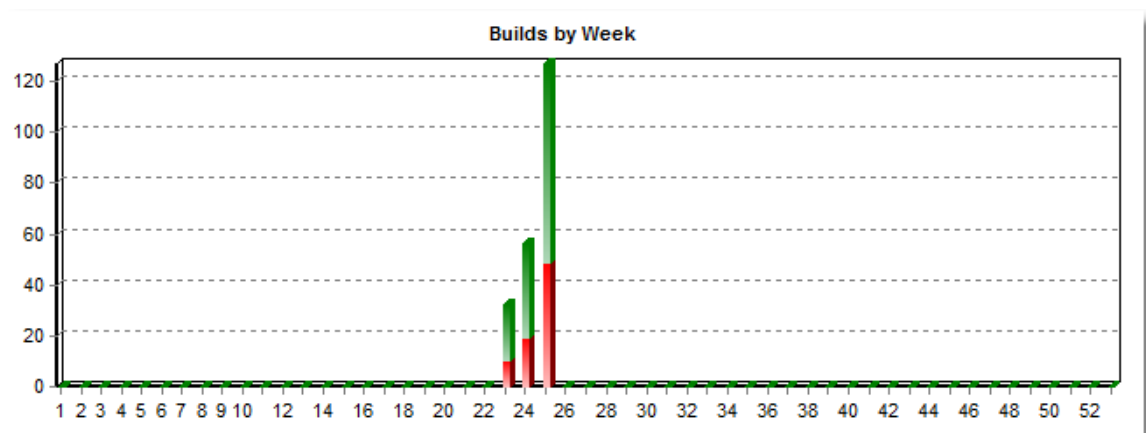
Builds by Day of Week



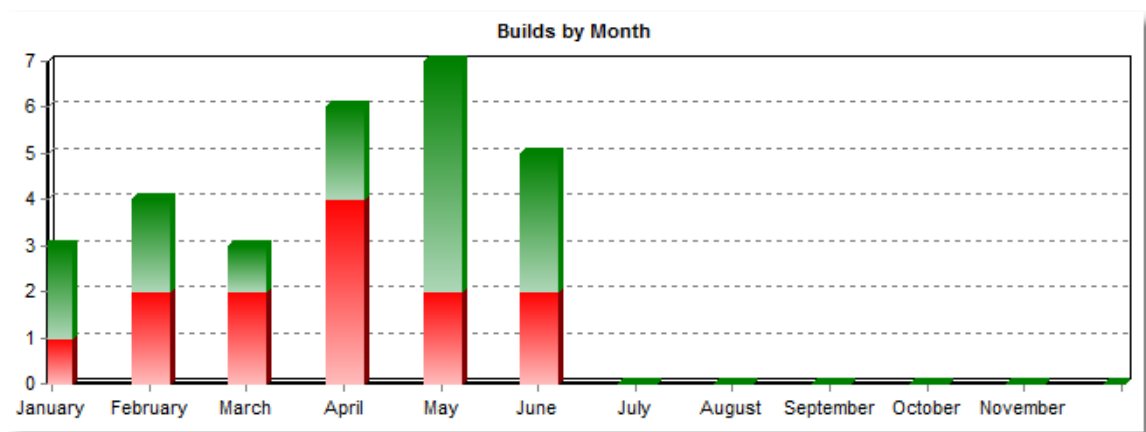
Builds by Day



Builds by Week

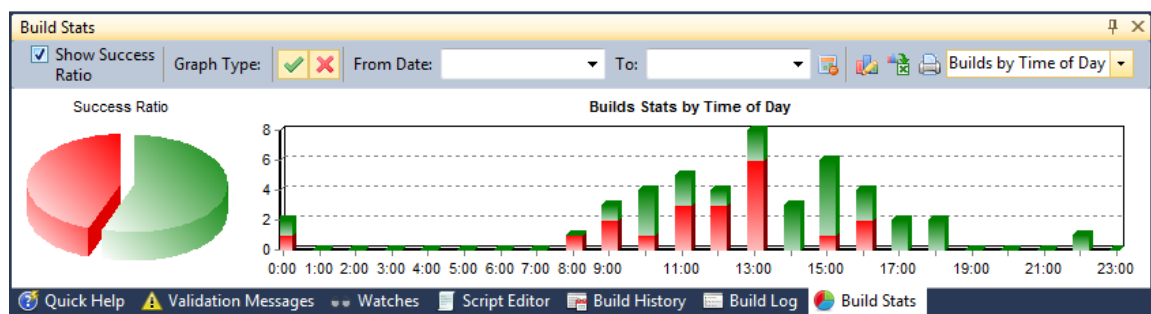


Builds by Month



3.4.1.3 Chart Data Options

Data options



The following data options are available:

- Show Completed Builds, Show Failed Builds

- From Date
- To Date

Show Completed Builds - this can be toggled by clicking the "Tick" icon. A blue border represents that the option is selected

Show Failed Builds - this can be toggled by clicking the "X" icon. A blue border represents that the option is selected

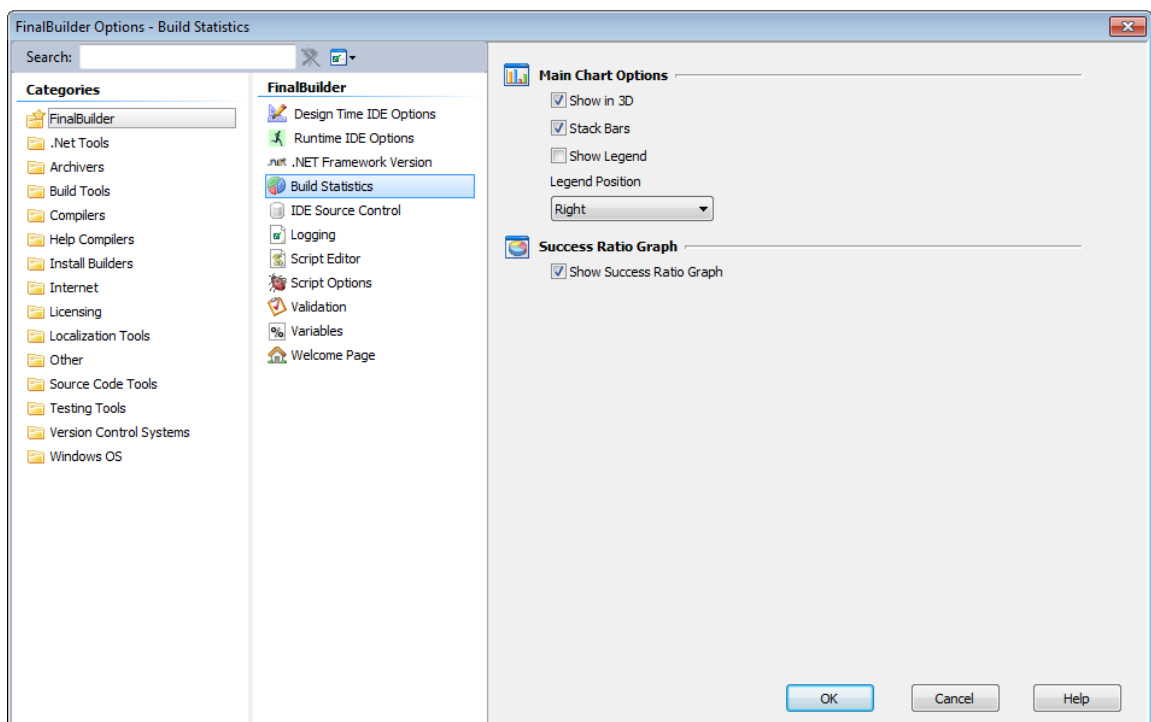
From Date - If you specify a from date, only builds which occur after this date will be shown on the graph

To Date - If you specify a to date, only builds which occur before this date will be shown on the graph

It is valid to specify a date range by selecting both a from date and a to date. To clear the date range, click the "Clear Date Range" icon.

3.4.1.4 Chart Visual Options

There are a number of visual options for you to change the graph layout to suit your requirements. The options are available in the main FinalBuilder options dialog (in the FinalBuilder category) or can be accessed by clicking the Graph Options button in the Build Stats tab.



Show in 3D - the bars are shown with a three dimensional visual effect.

Stack Bars - the failure and success bars are stacked on top of each other.

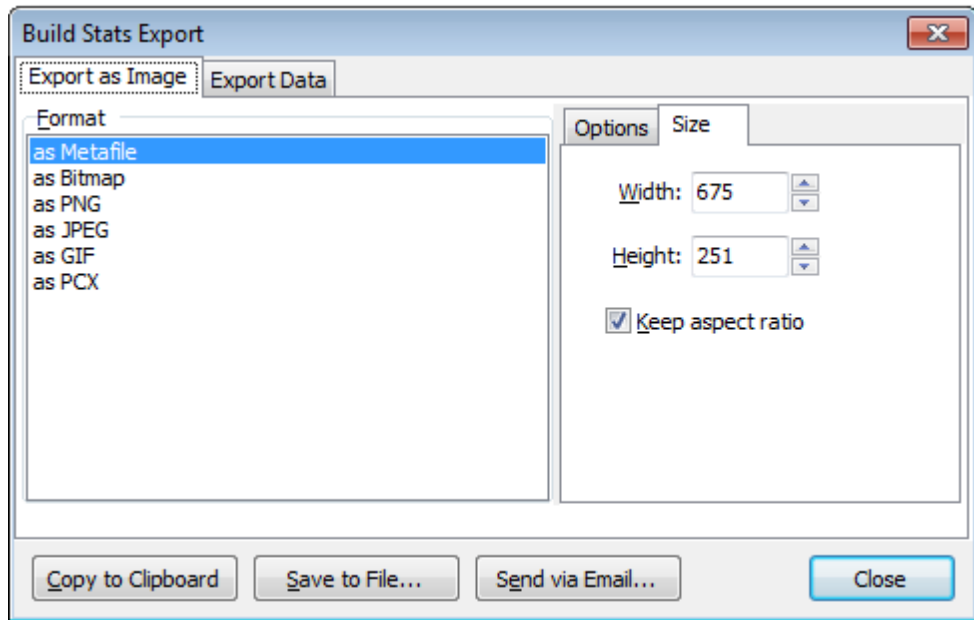
Show Legend - the legend is displayed on the graph.

Legend Position - the position to place the legend if it's enabled.

Show Success Ratio Graph - shows a pie chart of successful and failed builds, and related statistics.

3.4.1.5 Exporting Build Statistics

To export the build statistics as a chart or raw data, click the Export Graph button on the Build Stats tab.



There are two main options for exporting, either the chart as an image, or the raw data.

Export as Image

The available image formats are:

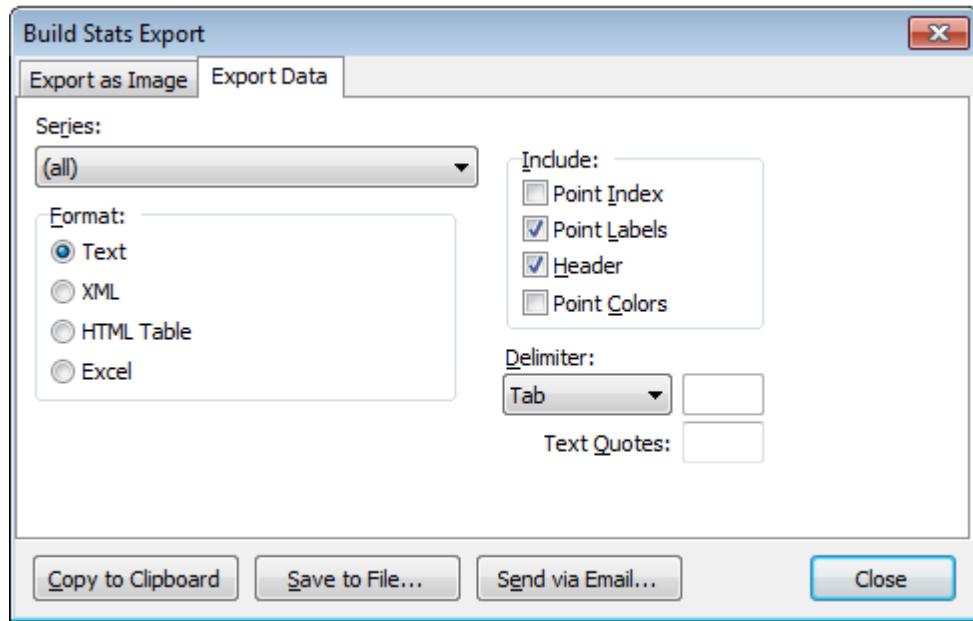
- Metafile
- Bitmap
- PNG
- JPEG
- GIF
- PCX

For each image format, there are different options which can be set via the Options and Size tabs on the right hand side.

When you are ready to export, you can export via the following methods:

- Copy to Clipboard - the image will be placed in the Windows clipboard
- Save to File - A file save as dialog will prompt for the file name to save the image as
- Send via Email - uses MAPI to attach the image file to a new email message

Exporting raw Data



Series - select if you want to export the failed data, success data, or both

Format - select the desired format (text, XML, HTML, or Excel)

Include - various data to include in the exported data

Delimiter - select the delimiter character if exporting to text

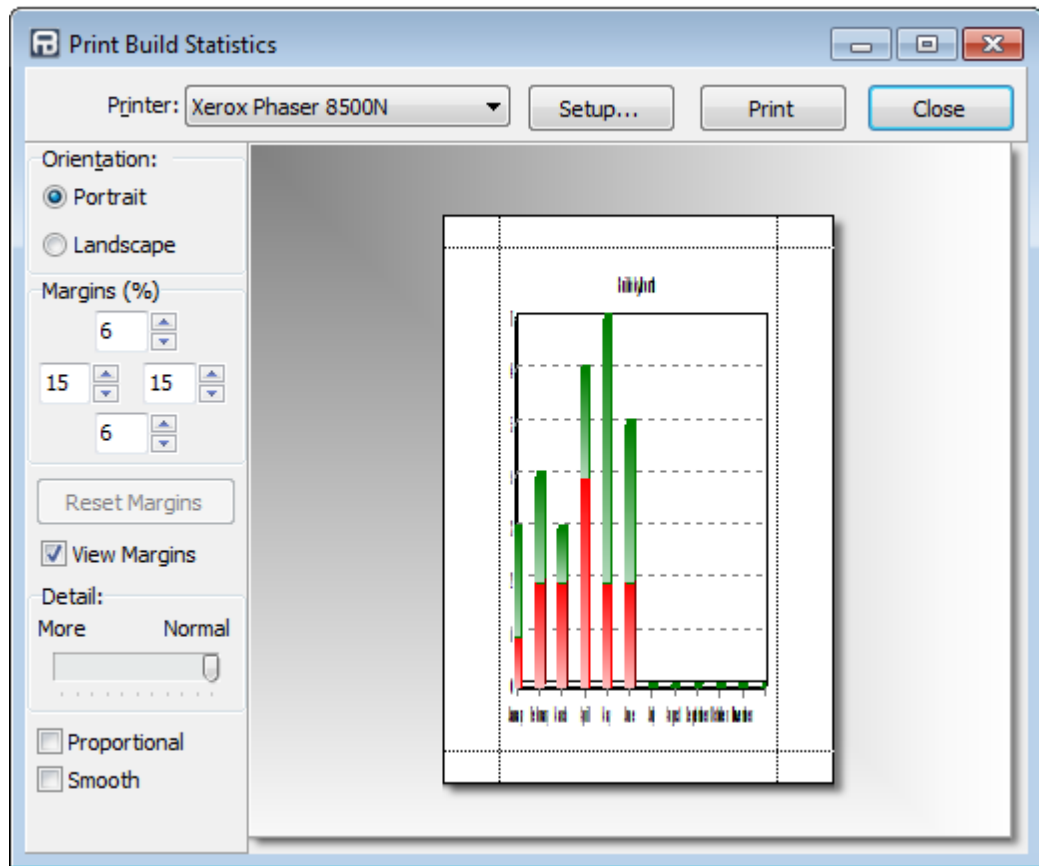
Text Quotes - select the quote character to enclose data in

When you are ready to export, you can export via the following methods:

- Copy to Clipboard - the data will be placed in the Windows clipboard (not available for Excel)
- Save to File - A file save as dialog will prompt for the file name to save the data as
- Send via Email - uses MAPI to attach the data file to a new email message

3.4.1.6 Printing Statistics Chart

To print the main chart as it currently appears, click the Print Chart button on the Build Stats tab.



This dialog shows a print preview of the chart. You may change the margins, orientation, detail level, the printer, and then when you are happy with the preview press the Print button to send the chart to the printer.

Orientation - Portrait or Landscape.

Margins - change the margins to make the graph larger or smaller on the page

View Margins - only for helping you layout the graph, will not be printed

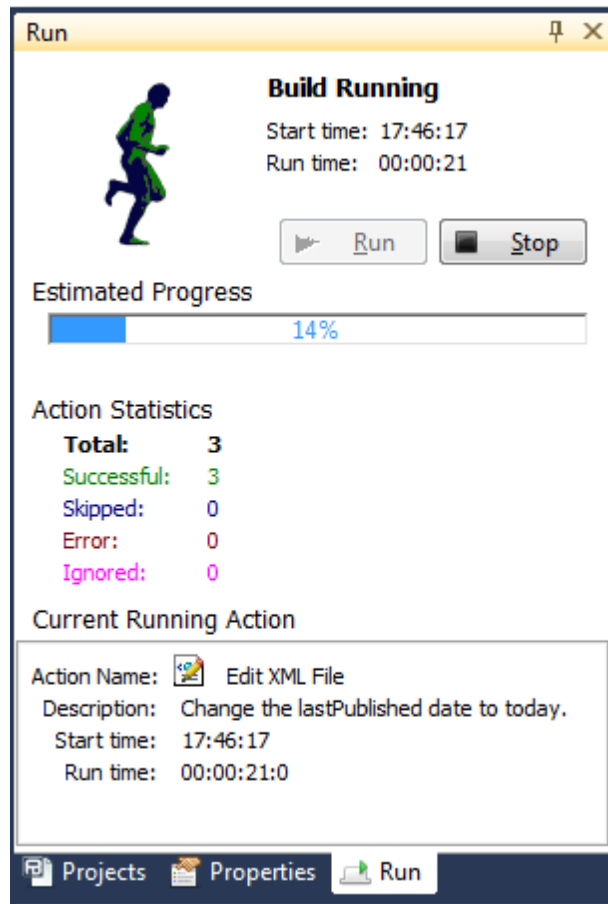
Detail - changes the text size relative to the graph. More detail will make the font size smaller

Proportional - will adjust the margins so the original aspect ratio is maintained

Smooth - anti-alias the graph (makes lines and text appear smoother)

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 build.



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

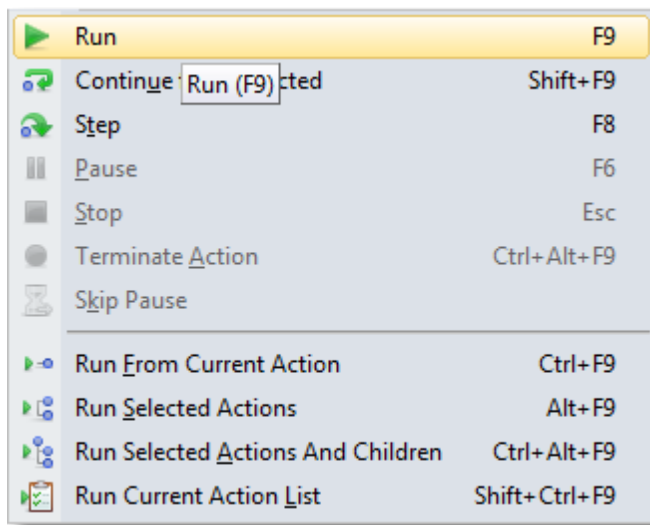
3.5.1 Running a Project

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 build will append to the existing log instead of starting a new log entry.
- **Run from Current Action** - the build 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:



2. Shortcut keys (eg. F9 for Run)

3. Toolbar



4. Right-click context menu of an action.

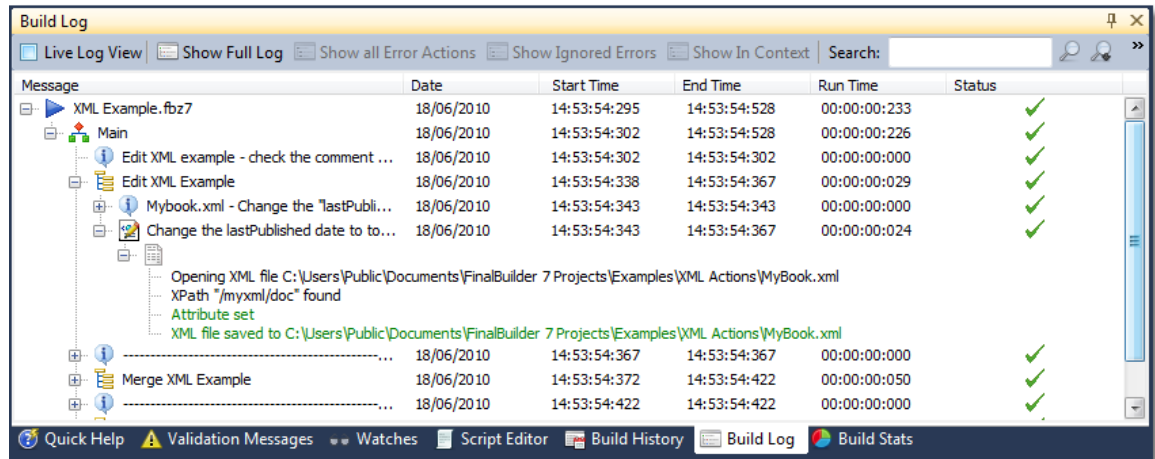
In addition to these run commands, you may want to step, pause or stop the build using the following commands:

- **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 build will pause after the current action completes.
- **Stop** - The build will stop after the current action completes. Some actions will stop prematurely if Stop is pressed, but most will not.
- **Terminate** - FinalBuilder will attempt to terminate the currently running action, and immediately stop the build. Terminate is a more abrupt version of Stop. Some actions don't support being terminated.
- **Continue** - The build will continue running if it's in the paused state






3.5.2 Build Log

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



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 FinalBuilder IDE. To show the logs for a particular action, right-click and choose "Filter Log By Action".

Live Log View

When a build 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 build stops.
- Live logging on: the log is displayed with every change.

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

If Live Logging is switched on while a build is running, then the full build 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 build. To see the complete hierarchy, it is necessary to wait until the build 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.

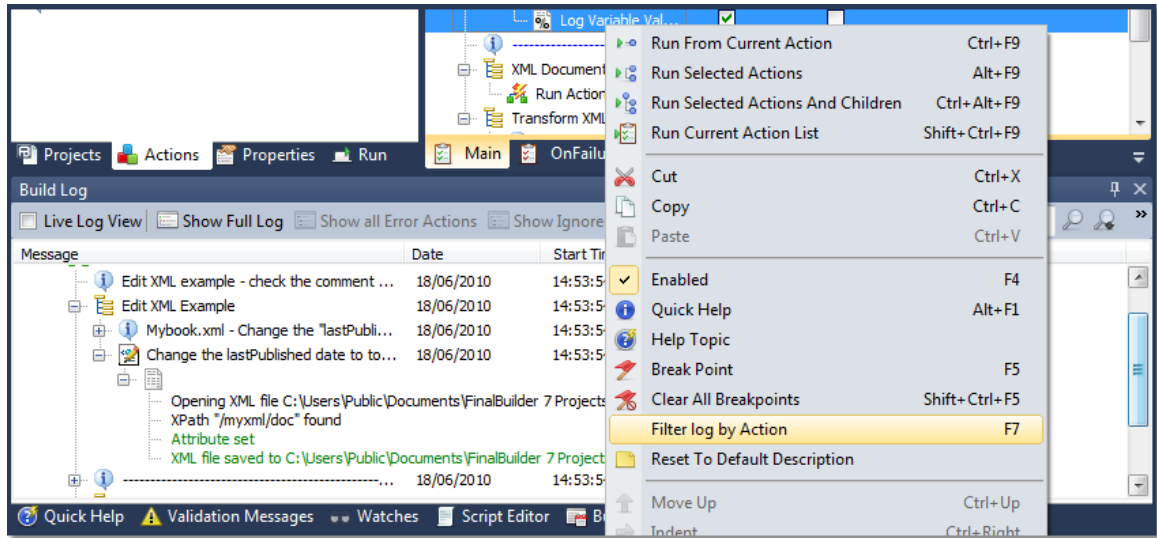
Note: searching action names or messages is equivalent to adding "action:" or "message" to the front of the search query.

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**.

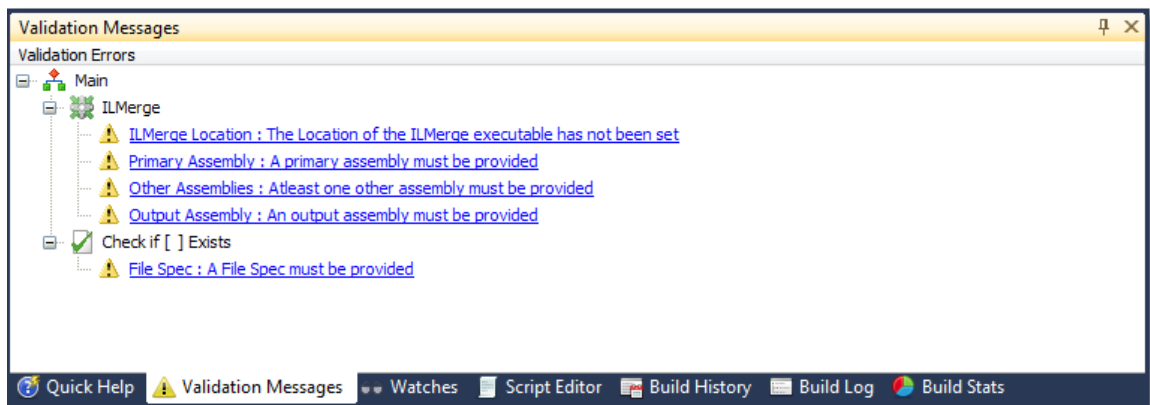


3.5.3 Validation

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 build 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 FBCMD is started with the /C option set.



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

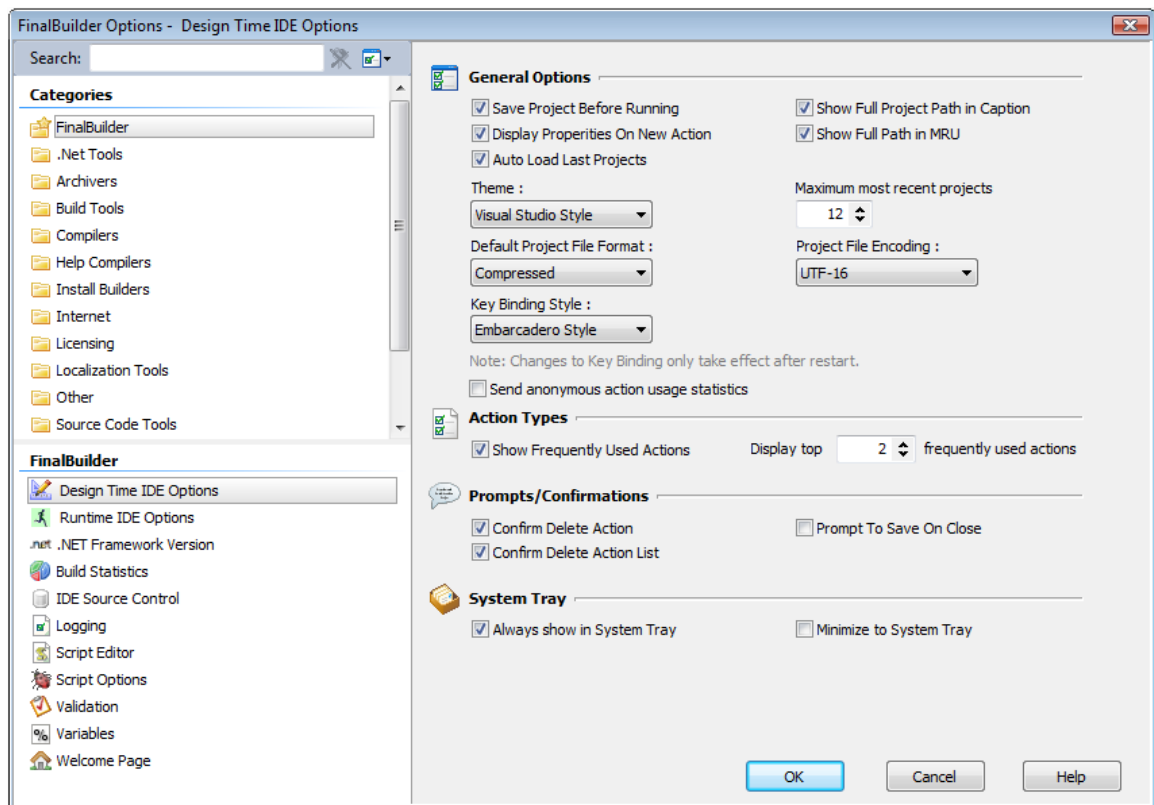
Validation does not prevent the project from being saved, but will prevent the project from being run. You can turn off pre-run validation in the Validation Options page.

3.6 IDE Features

3.6.1 Options Dialog

The options dialog allows you to set general FinalBuilder 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.

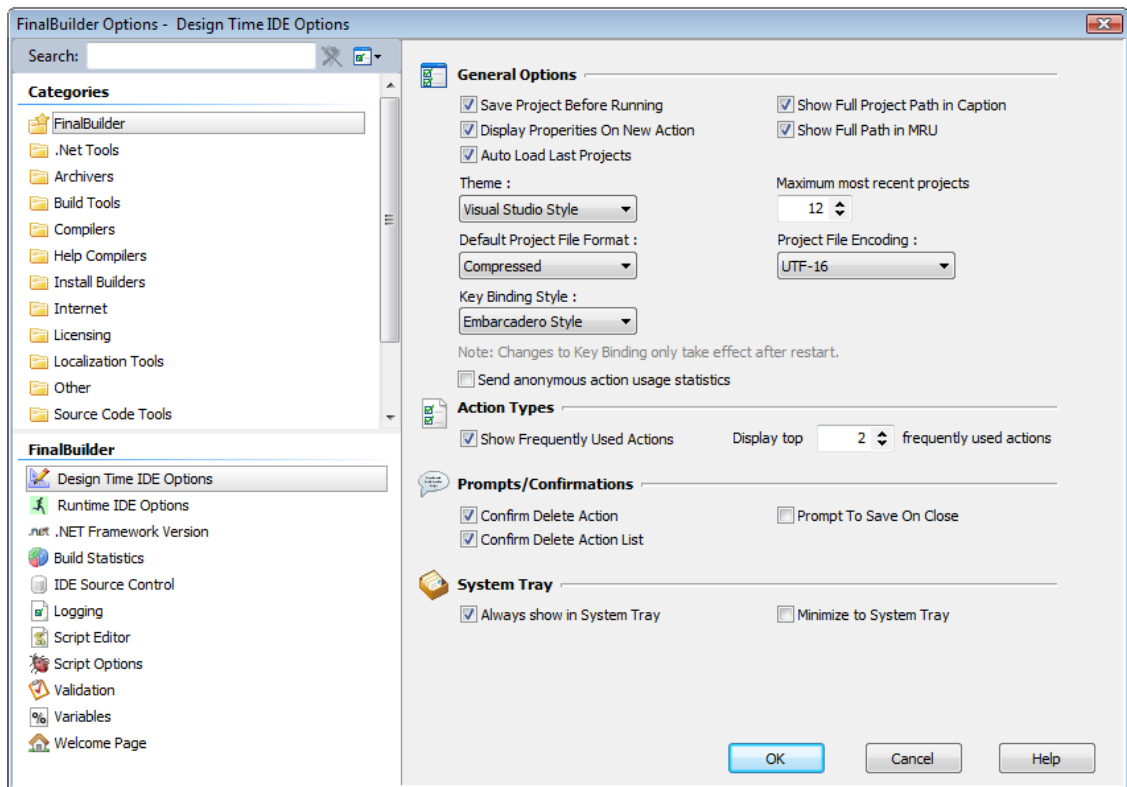


See also:

Design Time Options
 Runtime Options
 Validation Options
 Variables Options
 Logging Options
 Scripting Debugger Options
 Scripting Editor Options

Build Statistics Options
 Source Control Options

3.6.1.1 Design Time Options



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 FinalBuilder 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 FinalBuilder 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.

Theme

This option allows you to set the Style of the IDE. Themes available are *Visual Studio* and *RAD Studio*.

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 build, 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 FinalBuilder for the change to take effect.

Default Project File Format

If "compressed", projects are saved in the compressed .fbz7 format. Otherwise they are saved in the uncompressed .fbp7 format by default.

Project File Encoding

Selects the encoding that FinalBuilder 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 FinalBuilder each day.

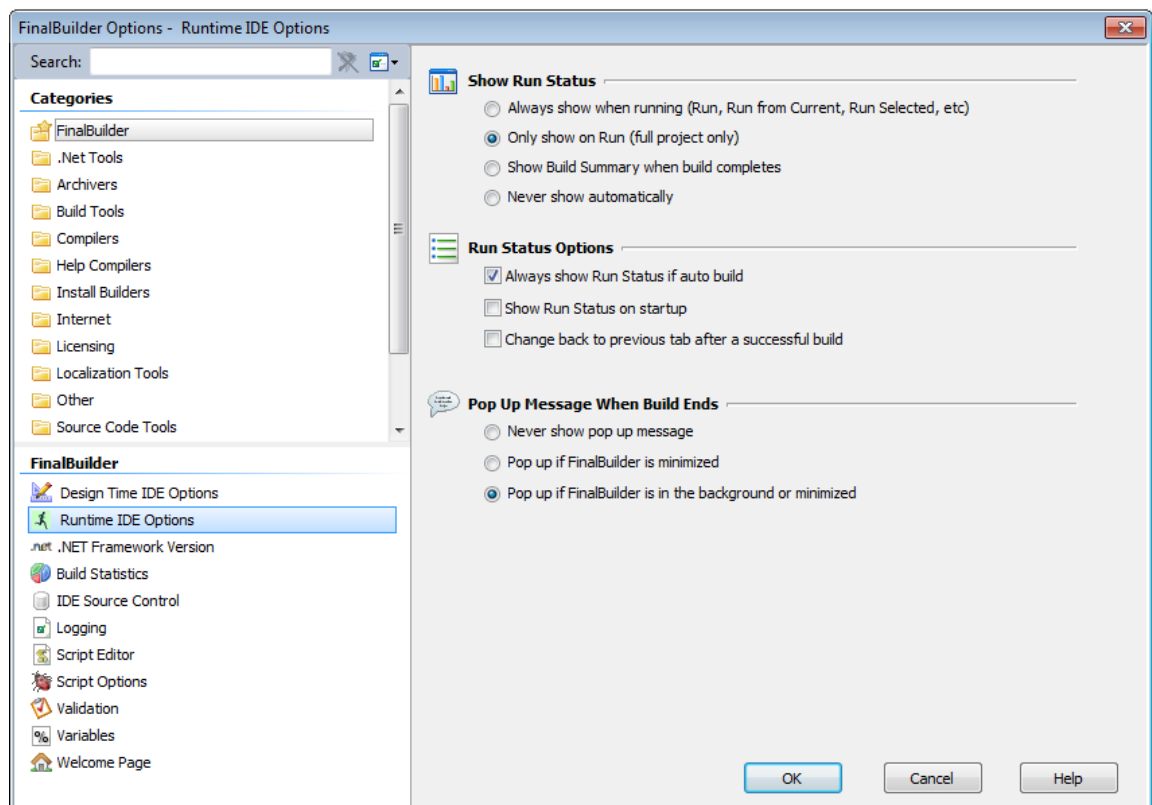
System Tray

FinalBuilder 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 FinalBuilder is minimized. Double-clicking on the system tray icon will restore the FinalBuilder IDE as the current focused application.

Minimize to System Tray

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

3.6.1.2 Runtime Options



Show Run Status

Automatically switches to the Run Tab in certain situations:

- Anytime a build is running
- Only when a build is run from scratch ("Run" menu item)

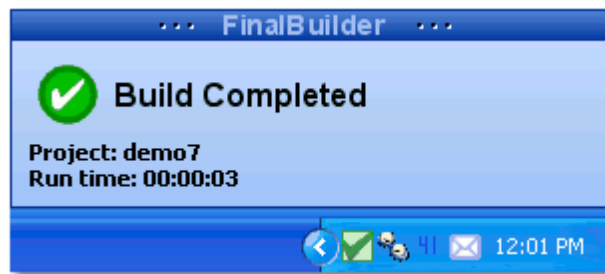
- When the build completes
- Never

Run Status Options

- **Always show Run Status if auto Build:** if running from the command line, shows the Run Tab.
- **Show Run Status on startup:** automatically chooses the Run Tab when FinalBuilder is first run.
- **Change back to previous tab after a successful Build:** if the Run Tab is automatically chosen for a build which completes successfully, returns to the previously selected tab.

Pop Up Message When Build Ends

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

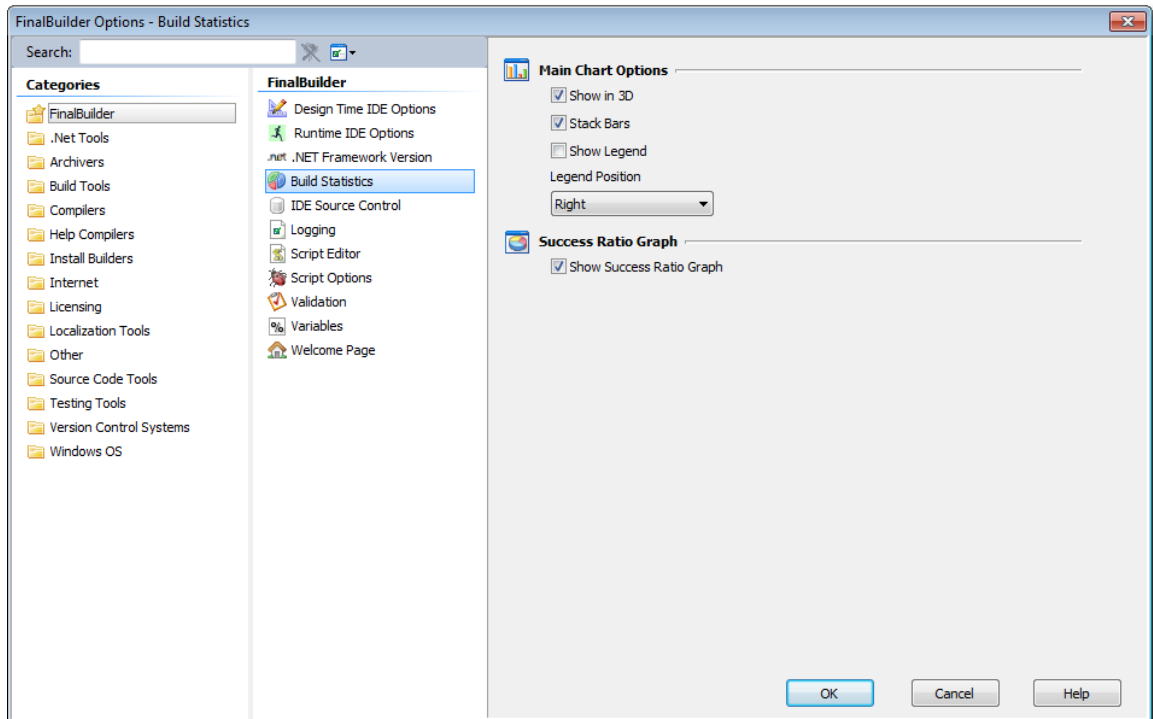


The message disappears when you click on it, or on the FinalBuilder IDE window.

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

3.6.1.3 Build Statistics Options

There are a number of visual options for you to change the graph layout to suit your requirements. The options are available in the main FinalBuilder options dialog (in the FinalBuilder category) or can be accessed by clicking the Graph Options button in the Build Stats tab.



Show in 3D - the bars are shown with a three dimensional visual effect.

Stack Bars - the failure and success bars are stacked on top of each other.

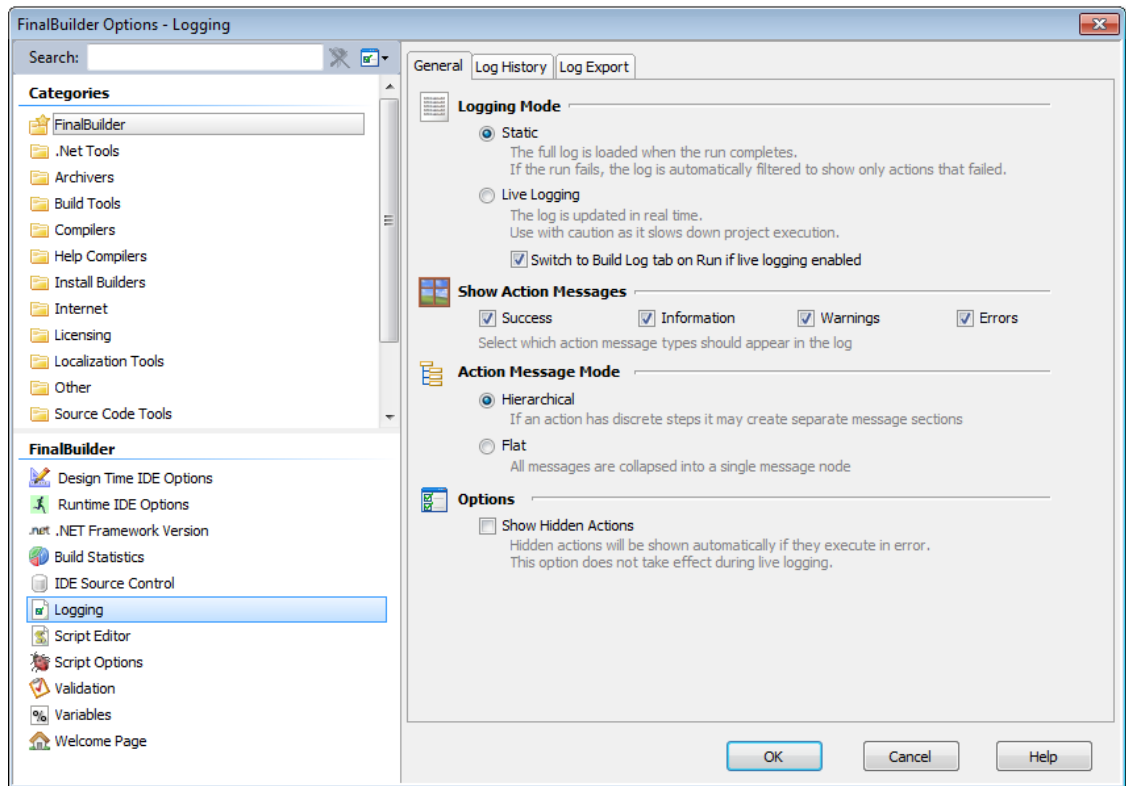
Show Legend - the legend is displayed on the graph.

Legend Position - the position to place the legend if it's enabled.

Show Success Ratio Graph - shows a pie chart of successful and failed builds, and related statistics.

3.6.1.4 Logging Options

General Options



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 builds and also a summary record of all past builds 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 Build Log tab when the project stops.

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

- **Switch to Build Log tab on Run if live logging enabled:** Automatically switches to the Build 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 Warning 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.

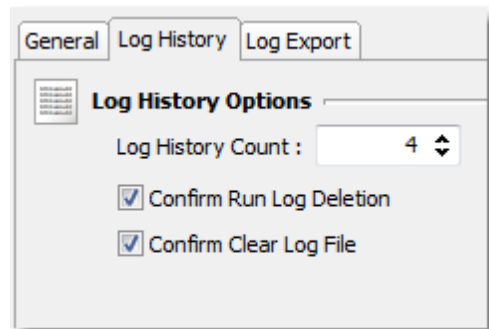
Action Message Mode

- 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.

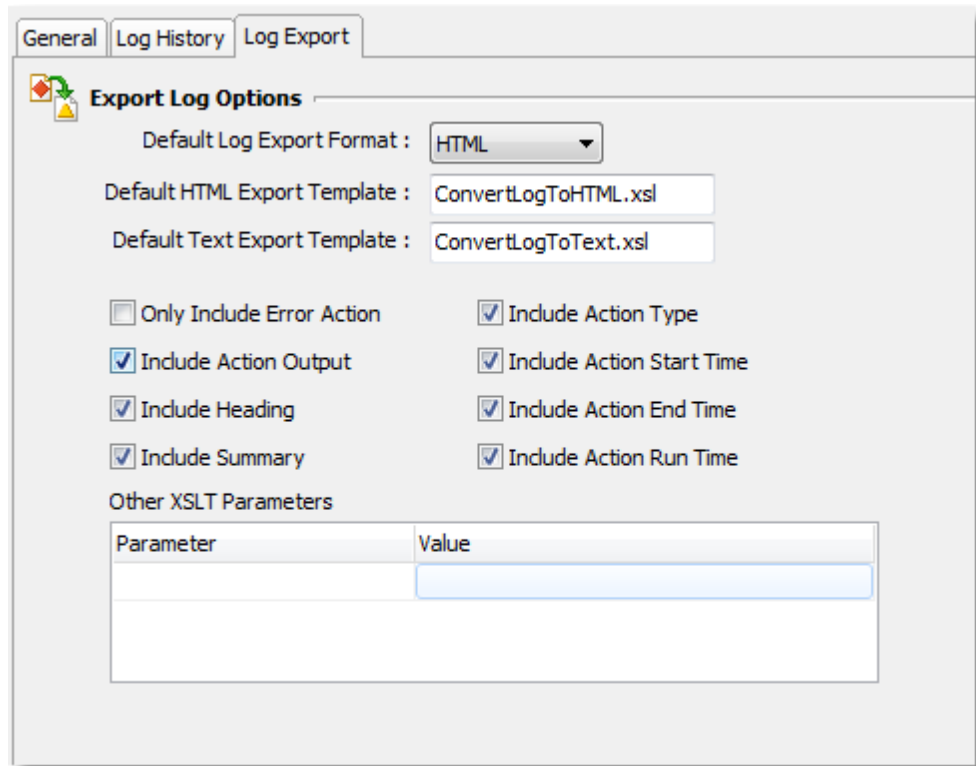
Log History Options



- Log History 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. This number has no effect on build statistics.
- 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.

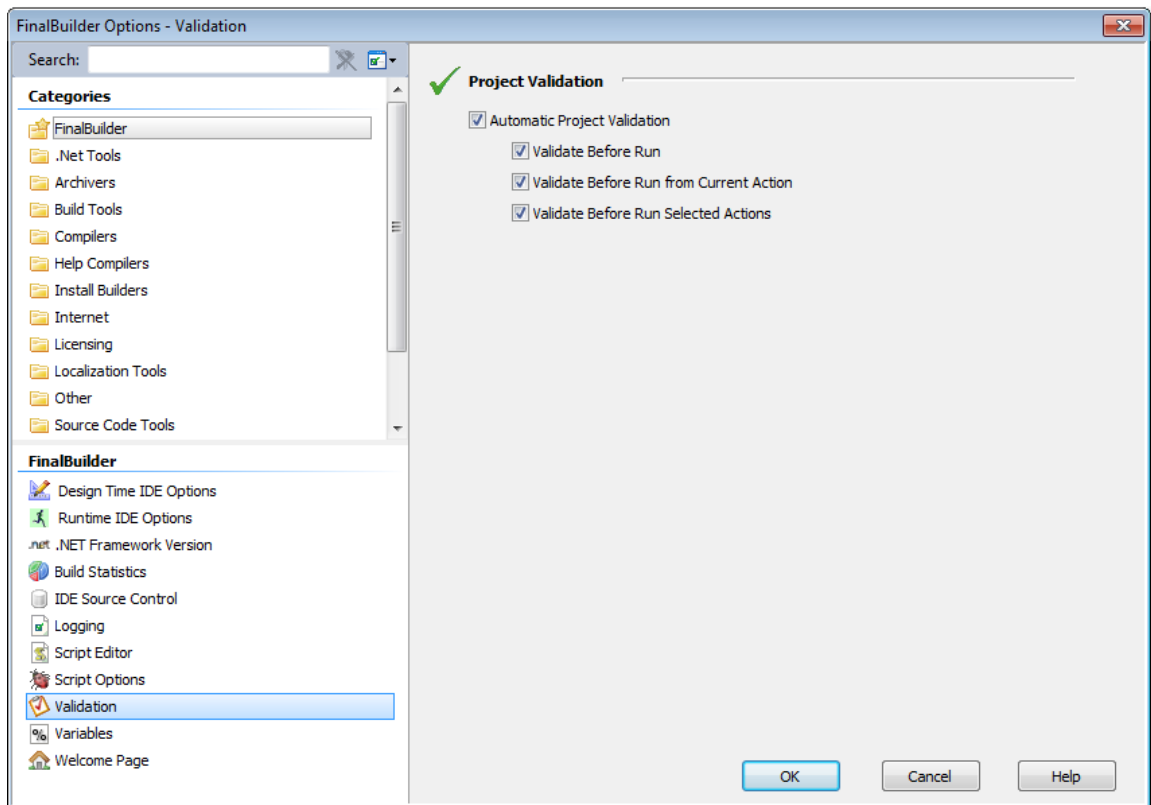
Export Log Options

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



- 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.

3.6.1.5 Validation Options



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

Three options are available:

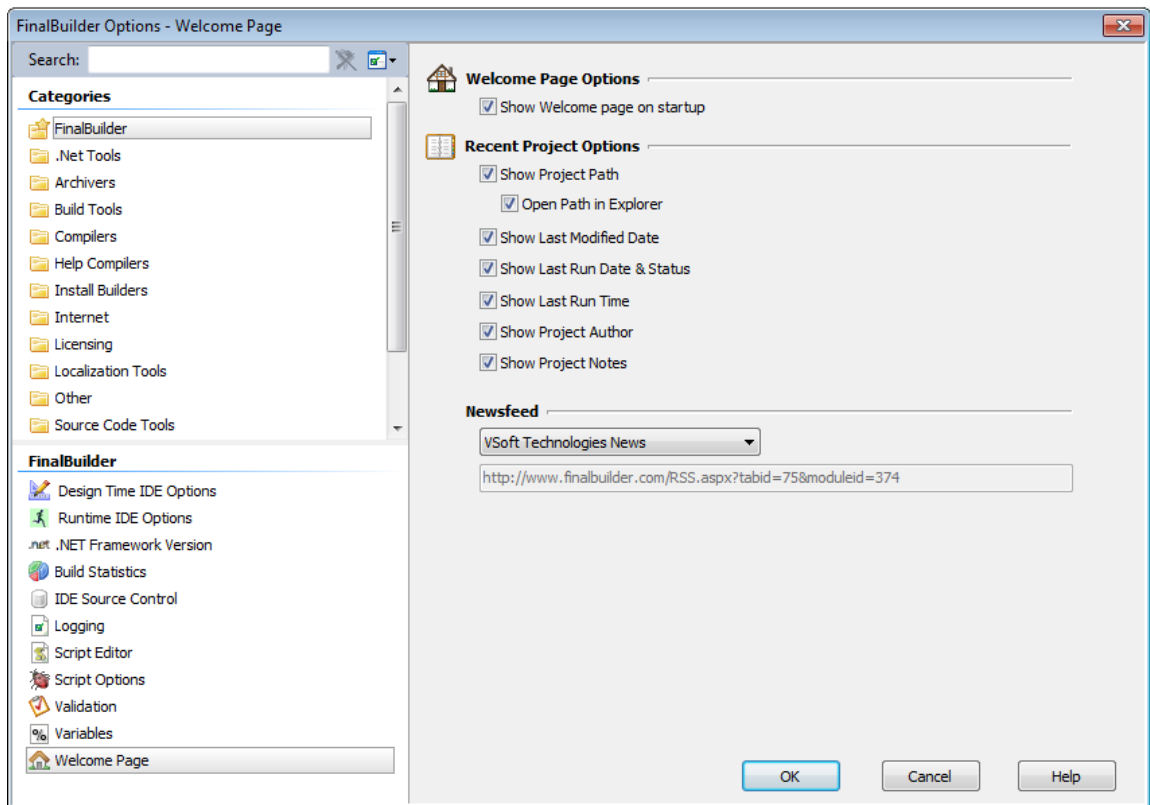
- Validate Before Run
- Validate Before Run from Current Action
- Validate Before Run Selected Actions

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](#).

3.6.1.6 Welcome Page Options



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

Welcome Page Options

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

Recent Project Options

Controls which 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 Source Control Integration

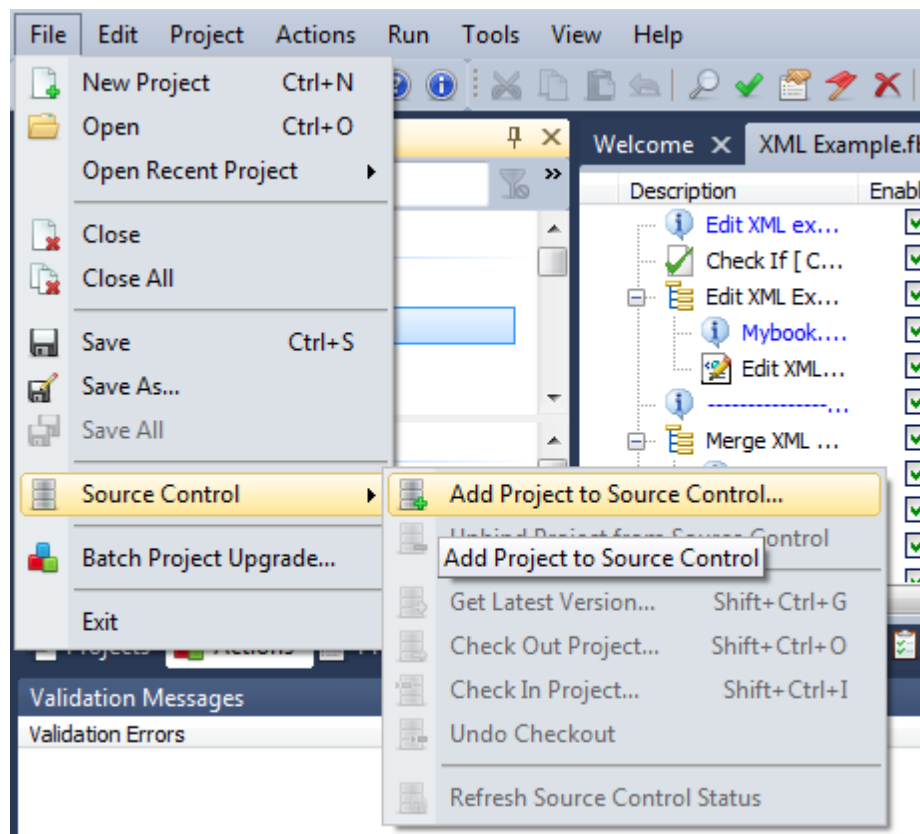
[FinalBuilder Professional Edition only]

FinalBuilder Professional features automatic integration with any Version Control System which supports the Microsoft SCC API. Projects can be automatically added to source control, checked out, edited, and checked in from inside FinalBuilder.

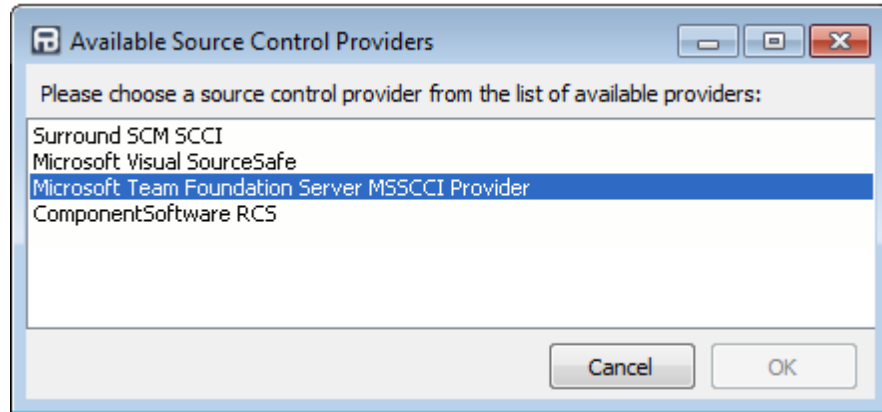
To use the Source Control Integration features of FinalBuilder, you will need to be using a Version Control System which provides a Microsoft SCC API Provider Plugin.

Adding/Binding New Projects

To use a project from inside FinalBuilder's SCC Integration, you first need to add it to source control inside FinalBuilder. Select "Add Project To Source Control" from the File -> Source Control menu.



You will be prompted to choose an SCC Integration Plugin from those which are available. If the plugin you wish to use is not shown, you may need to download and install an additional "SCC Integration Plugin" package from the Version Control System vendor.

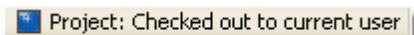


When you have chosen a provider, click OK. The next stages will depend upon the provider which you have installed. The source control provider may ask you for a source control project to use with the working directory that contains the FinalBuilder project.

It may also ask you for other information relating to adding the project file to Source Control.

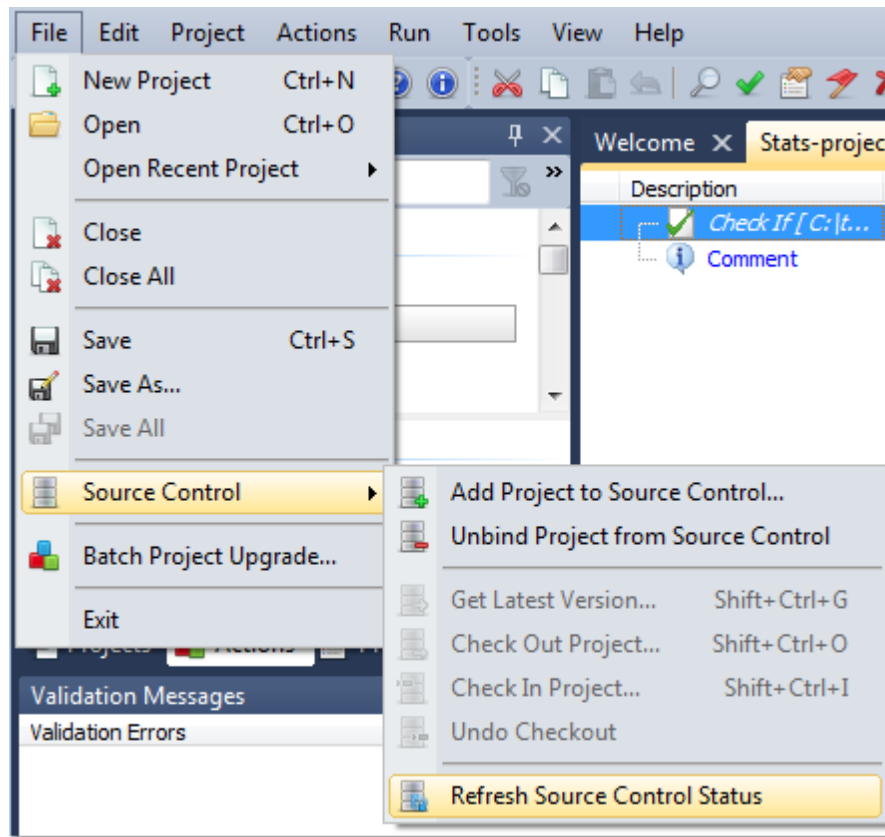
Source Control Status

Once the FinalBuilder project has been successfully added, you will see something similar to the following at the bottom of the FinalBuilder IDE window:



Project: Checked out to current user

This status information will update whenever you perform a source control operation. To manually refresh the status information, choose the Refresh Source Control Status option from the Source Control menu:



Unbinding A Project

To unbind a project from source control, choose the "Unbind Project from Source Control" command from the Source Control menu. Note that unbinding a project does not remove the file(s) from source control, it merely removes the connection between FinalBuilder and the source control project. If you wish to delete the files, you will need to do so manually.

Note: Moving, renaming, or "Saving As..." a FinalBuilder project will automatically unbind it from source control. To rebind the file, simply choose "Add Project to Source Control" again.

"Get Latest Version..."

Select this menu item to fetch the latest version of the project file from source control. The project will be reloaded if a new version is fetched.

"Check Out Project..."

Select this menu item to check out the latest version of the project file from source control.

"Check In Project..."

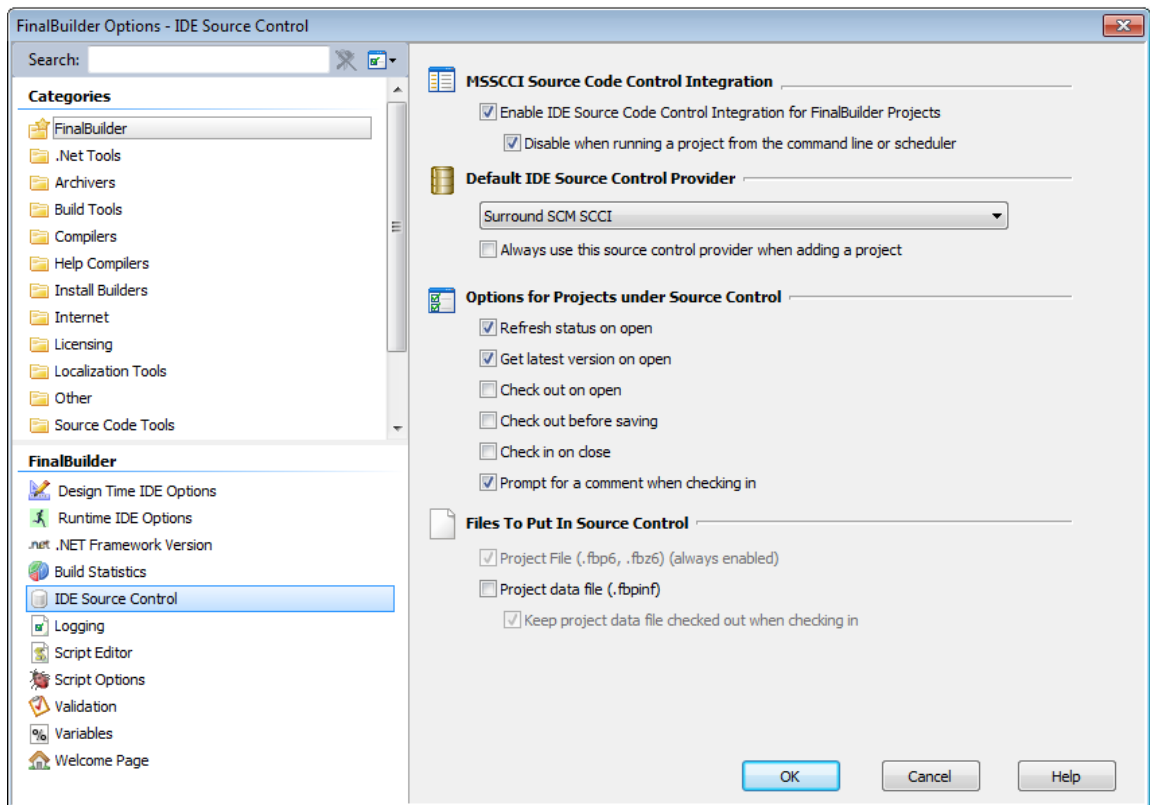
Select this menu item to check in a checked out version of the project file. Your source control provider plugin may prompt you for checkin information such as comments and workflow details.

"Undo Checkout"

Select this menu item to undo a checkout on an checked out FinalBuilder project. The project will be reloaded with the original, checked in, version.

3.6.2.1 Source Control Options

To edit FinalBuilder's Source Control Integration options, go to Tools menu | Options then click on the FinalBuilder category and the "IDE Source Control" item.



MSSCCI Source Code Control Integration

- **Enable Source Code Control Integration:** If this option is disabled, no SCC integration will be offered. FinalBuilder Projects which are bound to Source Control will have these bindings ignored.
- **Disable when running a project from the command line or scheduler:** If this option is enabled, no SCC integration will be offered when the FinalBuilder IDE is running a project passed in on the command line, or running from the scheduler.

Note that this option only applies to the FinalBuilder IDE. No SCC Integration is available at all when running from the FBCMD command line application. This is because most SCC Integration Provider Plugins display graphical dialogs which FinalBuilder cannot control.

Default Source Control Provider

This option will default to the last source control provider you selected, and is the provider which will be selected the next time the "Choose Provider" dialog is shown. Select "**Always use this source control provider...**" to skip the "Choose Provider" dialog when binding new projects.

Options for Projects under Source Control

FinalBuilder can be set to automatically perform certain SCC operations when bound projects are opened, saved and/or closed.

- Refresh status on open: Connects to the SCC repository and checks the project's status when opening a project.
- Get latest version on open: Automatically retrieves the latest version of the project from the SCC repository when opening.
- Check out on open: Checks out (locks) the project when opening it.
- Check out before saving: Checks out (locks) the project when saving it.
- Check in on close: Checks the project back into the SCC repository when the project is closed.
- Prompt for a comment when checking in: Automatically prompts you to enter a check-in comment on check-in. This is always bypassed for some SCC Providers which are known to always provide their own check-in dialog (like Team Foundation Server.)

Files To Put In Source Control

By default, only Project Files are operated on by the SCC operations. You can optionally include the Project Data File (.fbpinf)

Note: When using this option, if a new project data file appears is not yet in source control, the FinalBuilder IDE will report that the entire project is not bound to source control. Choose the "Add Project to Source Control..." command to add the missing file and restore normal SCC status. It should not be necessary to re-enter provider or binding information.

Note Also: Log Files cannot be automatically added to Source Control. It is recommended that FinalBuilder Log Files are never added to Source Control under any circumstances, as they are likely to impact detrimentally on the Source Control system's performance. Log files are large binary files which change substantially during a run.

- Keep project data file checked out when checking in: Enable this option to add the Project Data File file to source control, but keep it permanently checked out (checkins will still update the copy in Source Control.)

3.6.3 Automatic Action Usage Updates

What Are Automatic Usage Updates?

In order to continue making FinalBuilder 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 FinalBuilder 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:

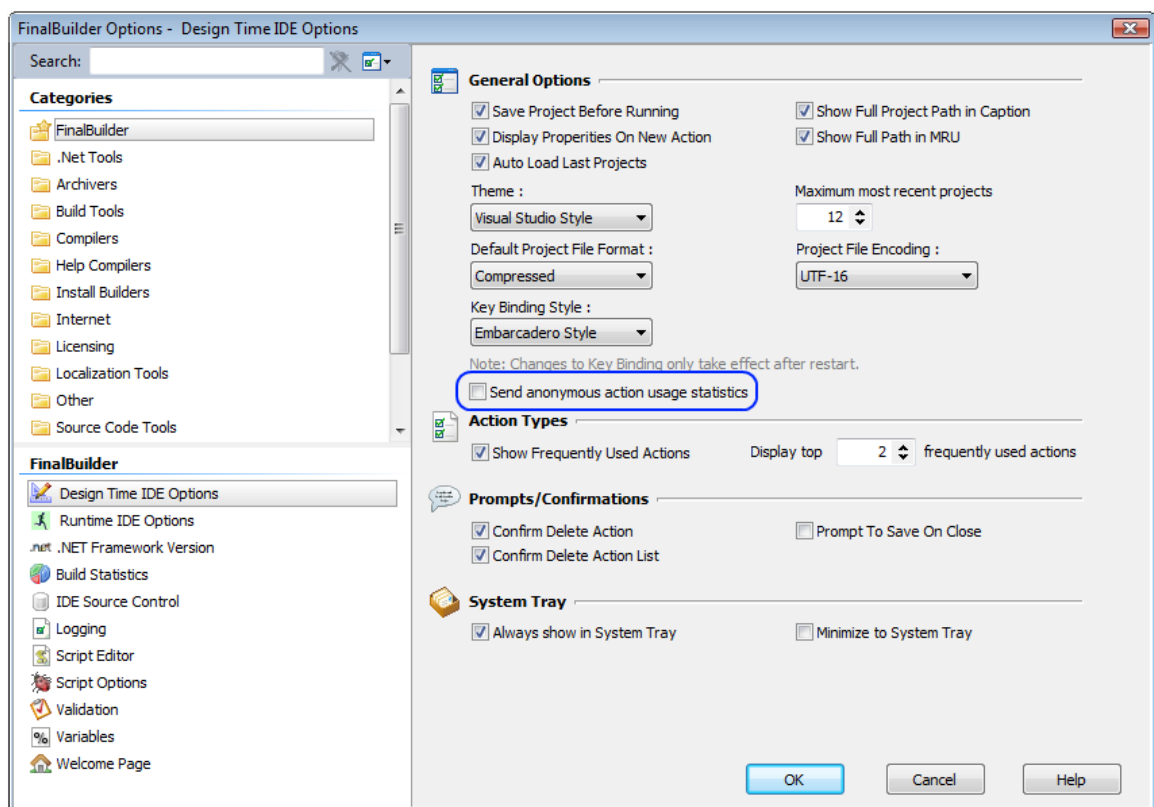
- The version of FinalBuilder that you are using.
- The names (and total numbers) of actions which were added to projects.

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 FinalBuilder, 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 "FinalBuilder" and then "General IDE Options":



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 FinalBuilder is launched. If you are consistently receiving errors about failed submissions, then you may need to consider disabling the automatic usage updates feature.

3.6.4 Check for Updates

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

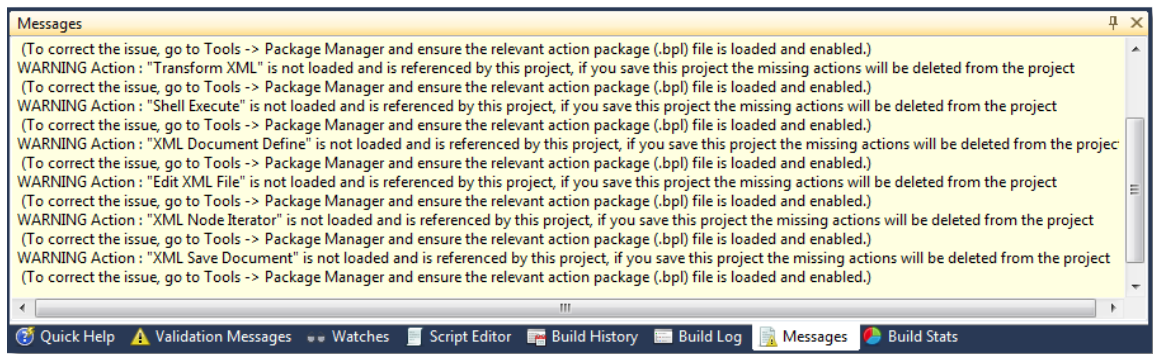
If an update is available, the Quick Help tab will display the details of the update and provide a link to any updated files and further instructions.

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

3.6.5 Messages

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

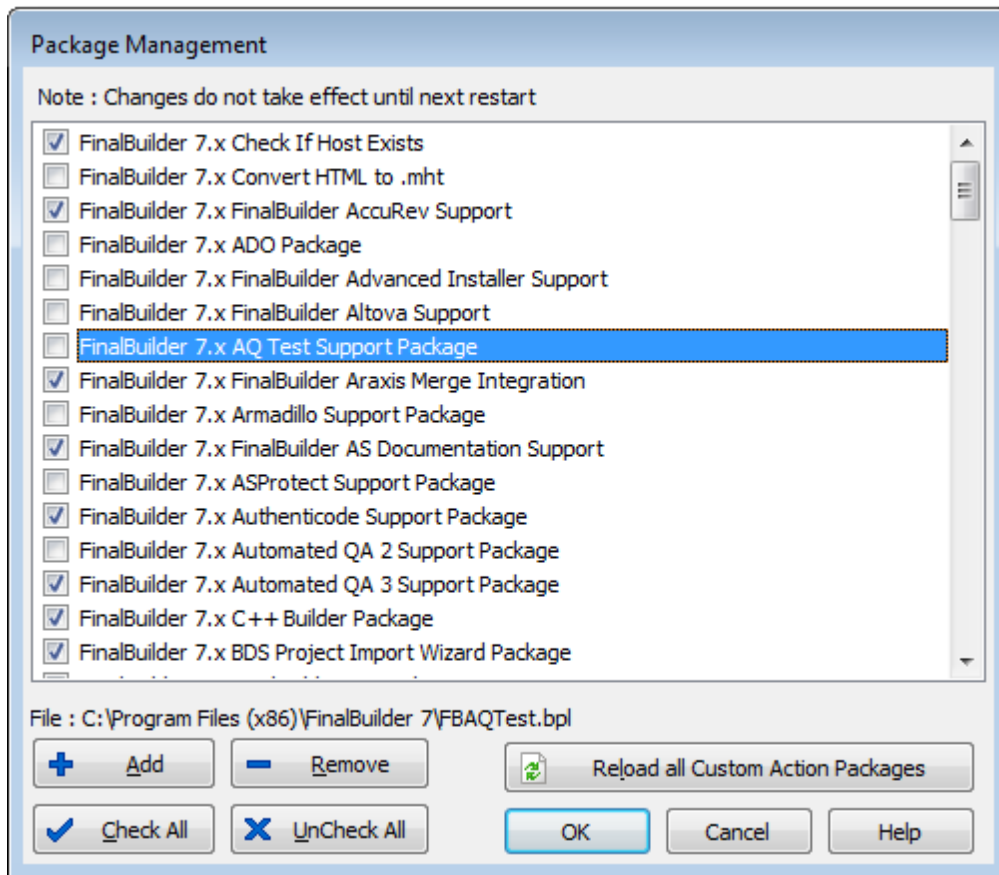
For example, if a project is loaded and it contains an action which is has not been loaded in the IDE (because the package has been disabled or removed), then it shows an error message such as the one below:



3.6.6 Package Manager

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

A simpler alternative to the Package Manager is the Configuration Wizard.



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

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

FinalBuilder action packages come in three kinds:

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

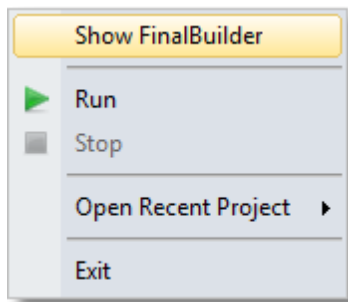
(Note that the FinalBuilder package manager will automatically add new custom action packages from the ActionDefs folder, although custom action packages can be added from any location.)

3.6.7 Tray Icon

FinalBuilder can optionally show in the System Tray:



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



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

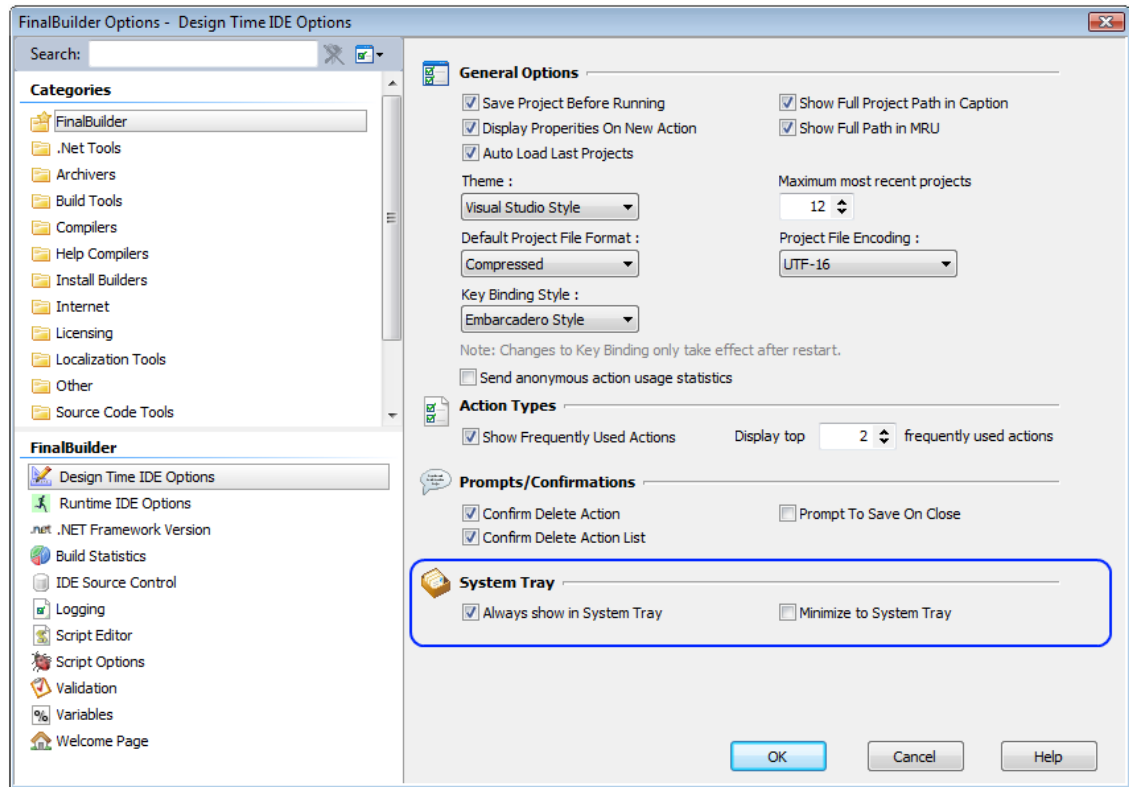
Run <project> - This will start the build.

Stop - Stops the build

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

Exit - Closes FinalBuilder

To change the System Tray settings:









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, FinalBuilder 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 build:

-  - FinalBuilder is currently idle
-  - build is currently running
-  - build completed successfully
-  - build completed with an error
-  - build failed validation
-  - build is paused

3.6.8 Keyboard Shortcuts

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

	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 Build 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

Source control:

Get latest version	Shift+Ctrl+G
Check out project	Shift+Ctrl+O
Check in project	Shift+Ctrl+I

3.7 FinalBuilder Projects

3.7.1 Project Files & Other Files

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

<project>.fbz7

Compressed FinalBuilder 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>.fbp7

Uncompressed FinalBuilder 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

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

<project>.fbpInf

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>.fbl7

The FBL7 file is the log archive file. It records the logs of any previous builds, up to the "Log History Count" option (default is 4). The log file must be writeable in order to open the project file in FinalBuilder.

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>.fb7lck

The lock file is used to indicate when the project is currently opened in FinalBuilder. Other instances of FinalBuilder cannot open the same project at the same time. The lock file is deleted when a project is closed, or FinalBuilder closes. If FinalBuilder 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.

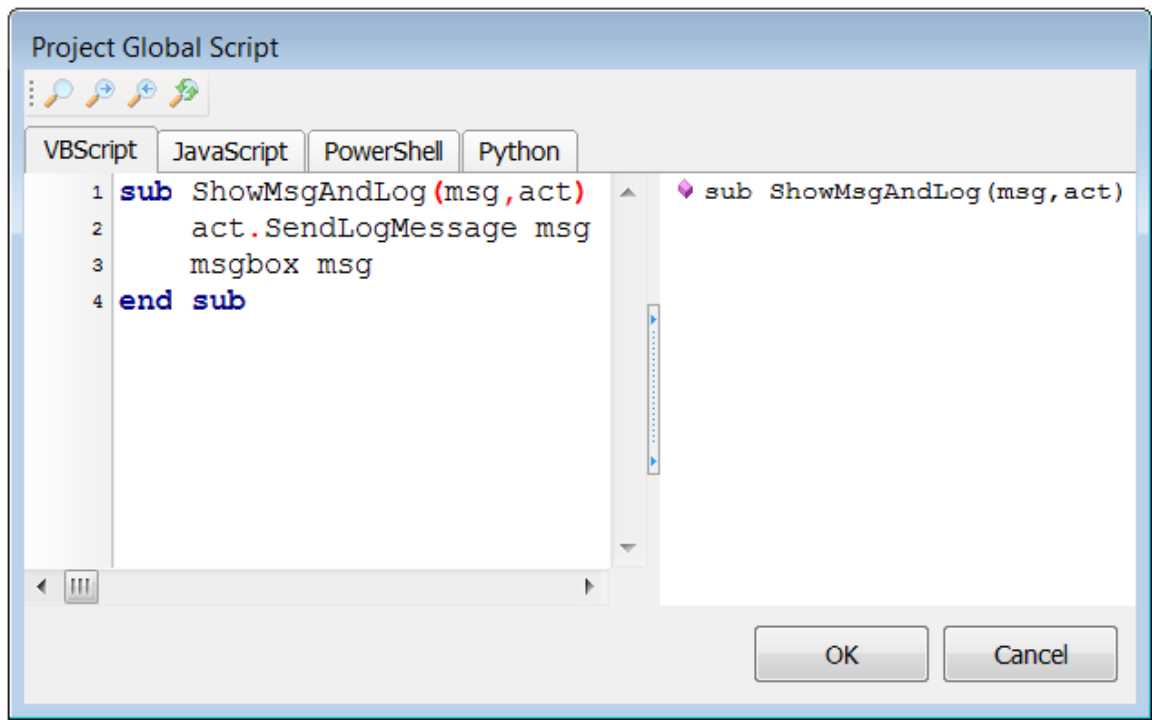
See also

Upgrading Projects

3.7.2 Project Global Script

FinalBuilder 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:



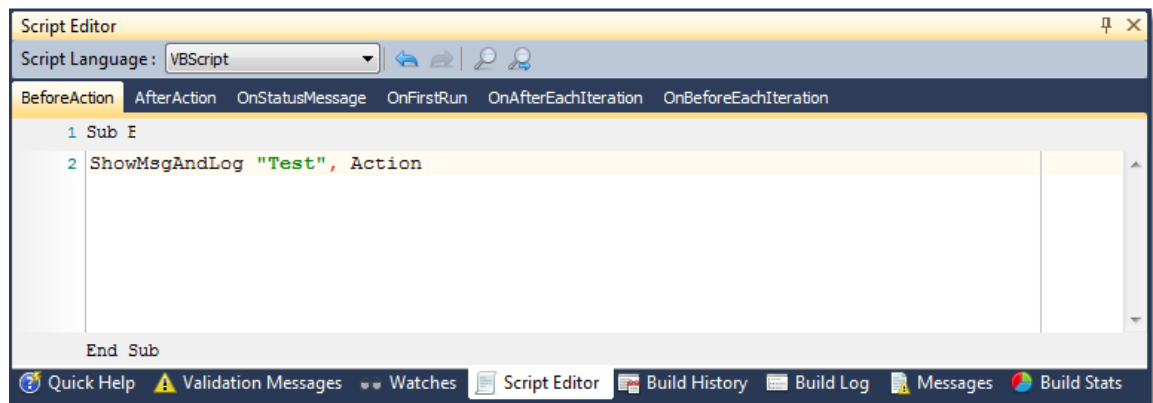
This dialog supports syntax highlighting, code completion, parameter hints and also will display subs, functions and constants in the right hand side section to make it easy to see an overview of your code at a glance as well as easy navigation within the script (double-click an item and it scrolls the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

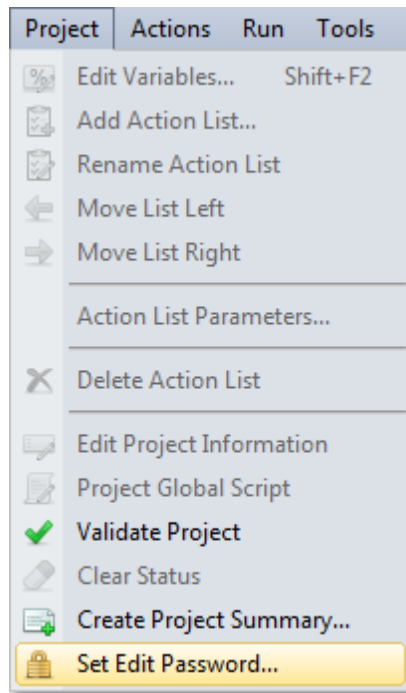


3.7.3 Project Edit Password

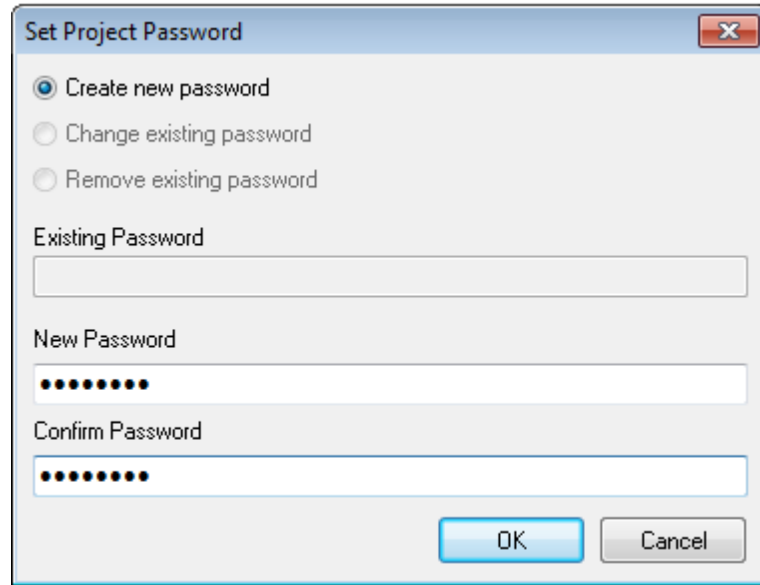
[FinalBuilder Professional Edition only]

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 build 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.

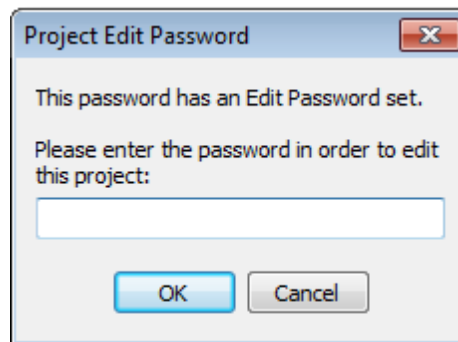
To set an edit password, choose "Set Edit Password" from the Project menu:



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.



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

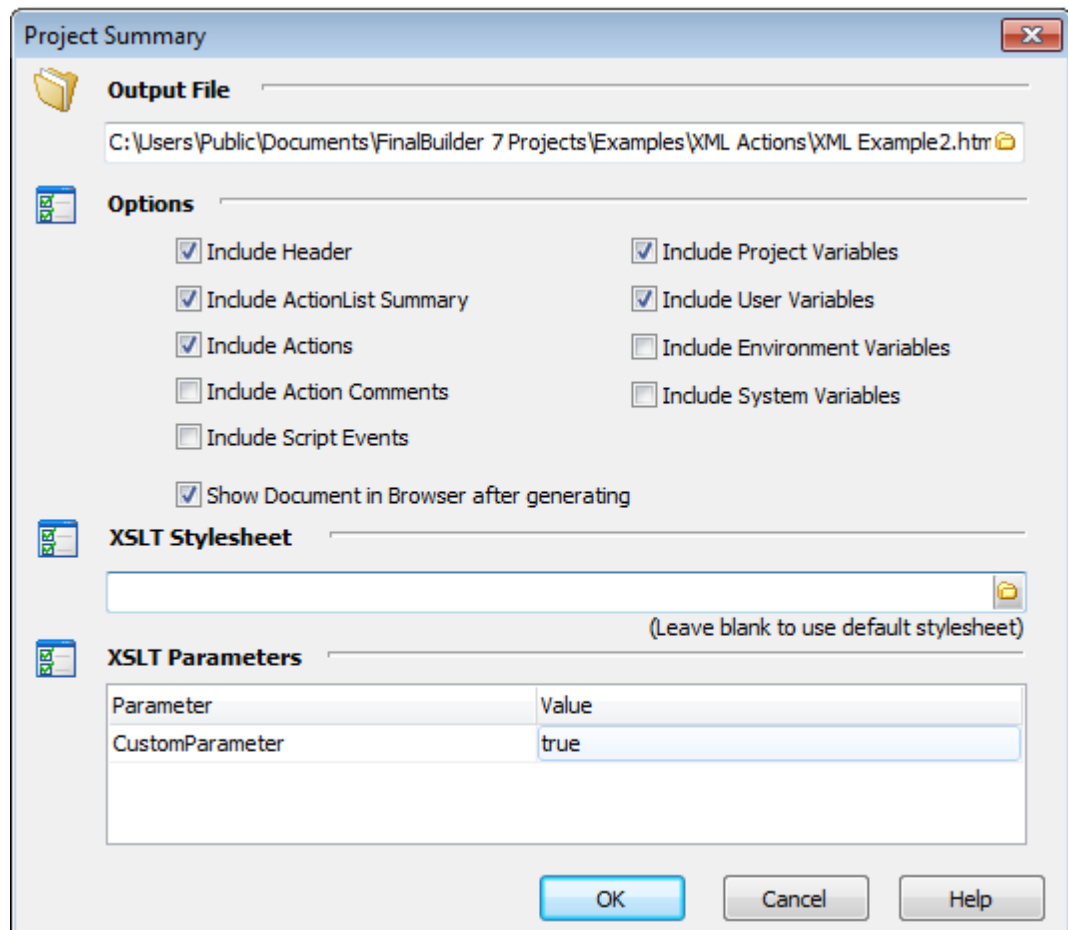


One of the limitations with project edit passwords is that the project file must be saved as a compressed project file (.fbz7) - as this prevents the encrypted password in the project file being removed easily.

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.

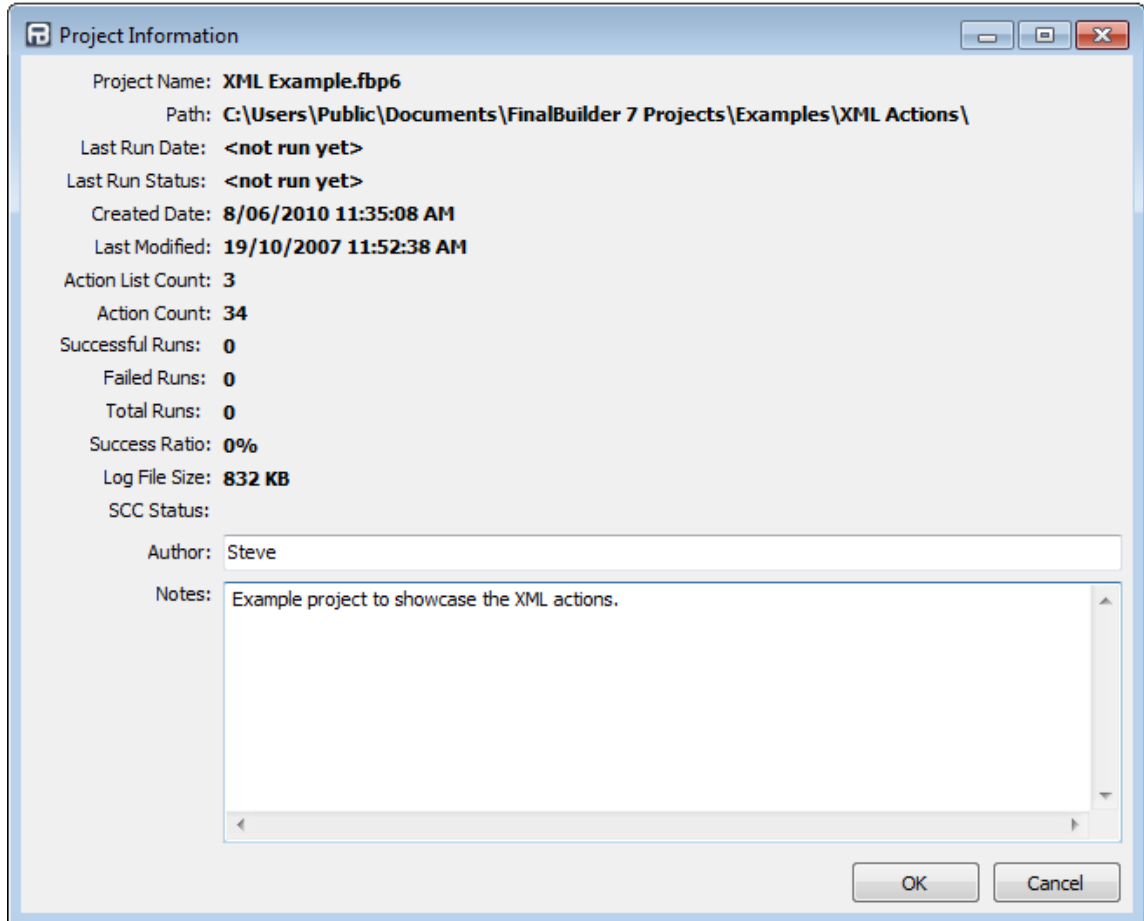


The XSLT Stylesheet option allows you to use your own custom stylesheet to control the format of the output. If this field is left blank the default stylesheet (<install directory>\Stylesheets\CreateProjectSummaryFromProject.xsl) will be used.

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.



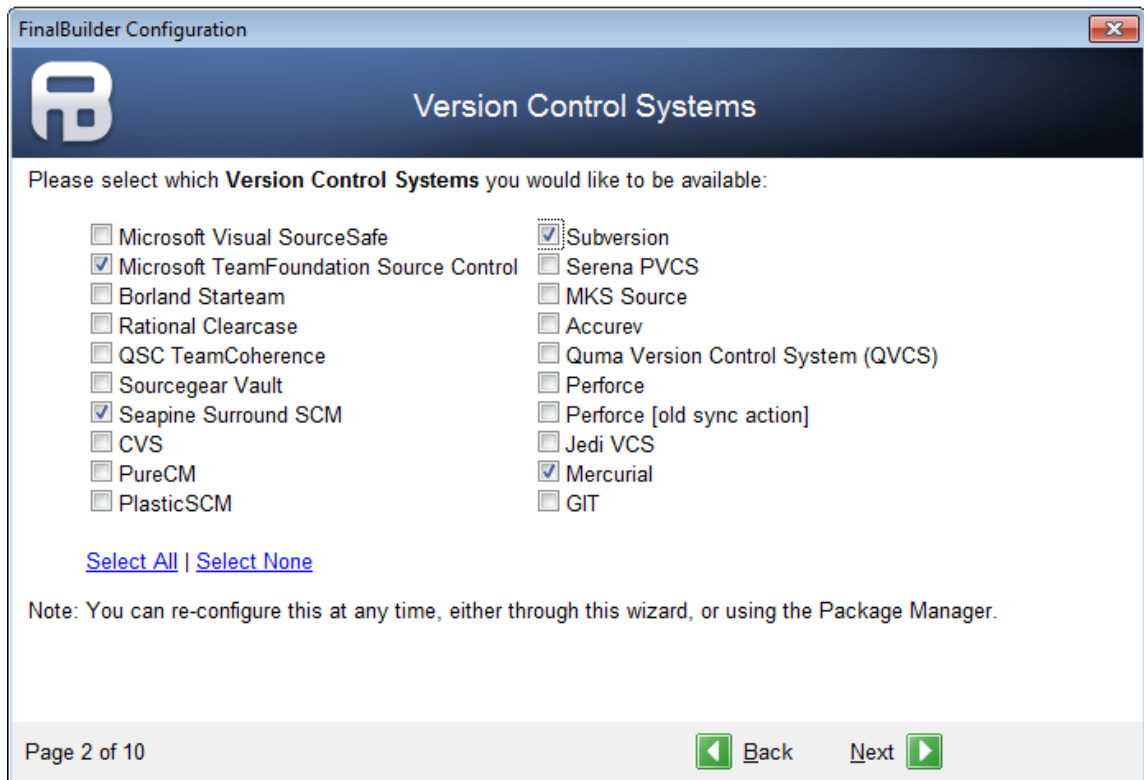
The **Author** and **Notes** fields can contain any information you like. They are shown by default on the Welcome Tab.

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.

3.7.6 Configuration Wizard



The Configuration Wizard is an efficient way to set up FinalBuilder to suit your needs. It sets up the Package Manager by asking a series of questions, and finally sets a few default options.

The Configuration Wizard is run the first time you launch FinalBuilder, and you can run it again by choosing **Run Configure FinalBuilder Wizard** from the Tools menu. It is a good idea to run the wizard periodically to check whether you have packages selected that you don't need, and to see if there are packages that might have become useful to you that were disabled. For example, if you have begun using Subversion, run the wizard to quickly add that package.

The wizard consists of the following pages:

1. Introduction
2. Version Control Systems
3. Compilers
4. Help Compilers
5. Installers
6. Testing and Obfuscation Tools
7. Licensing Tools
8. Other Packages
9. Other Options

On the "Other Options" page you can set the default script language, maximum number of logs, and the key bindings to use. These correspond to options on the Script Editor Options, Logging Options and Design Time Options pages respectively.

4 Scripting

4.1 Scripting in FinalBuilder

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

PowerShell

To use PowerShell in FinalBuilder 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 FinalBuilder, 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.

FinalBuilder 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 FinalBuilder Variables in Scripts

FinalBuilder 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 FinalBuilder 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 FinalBuilder 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:\FinalBuilder\scripts\iis_stuff.vbs
```

Example - JavaScript

```
//USEUNIT c:\FinalBuilder\scripts\iis_stuff.js
```

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

You can also use FinalBuilder variables in the path, for example : 'USEUNIT "%SCRIPTPATH%\Test.vbs"

Note that this is the only time you would use the %variable% syntax for FinalBuilder 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 FinalBuilder.

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>

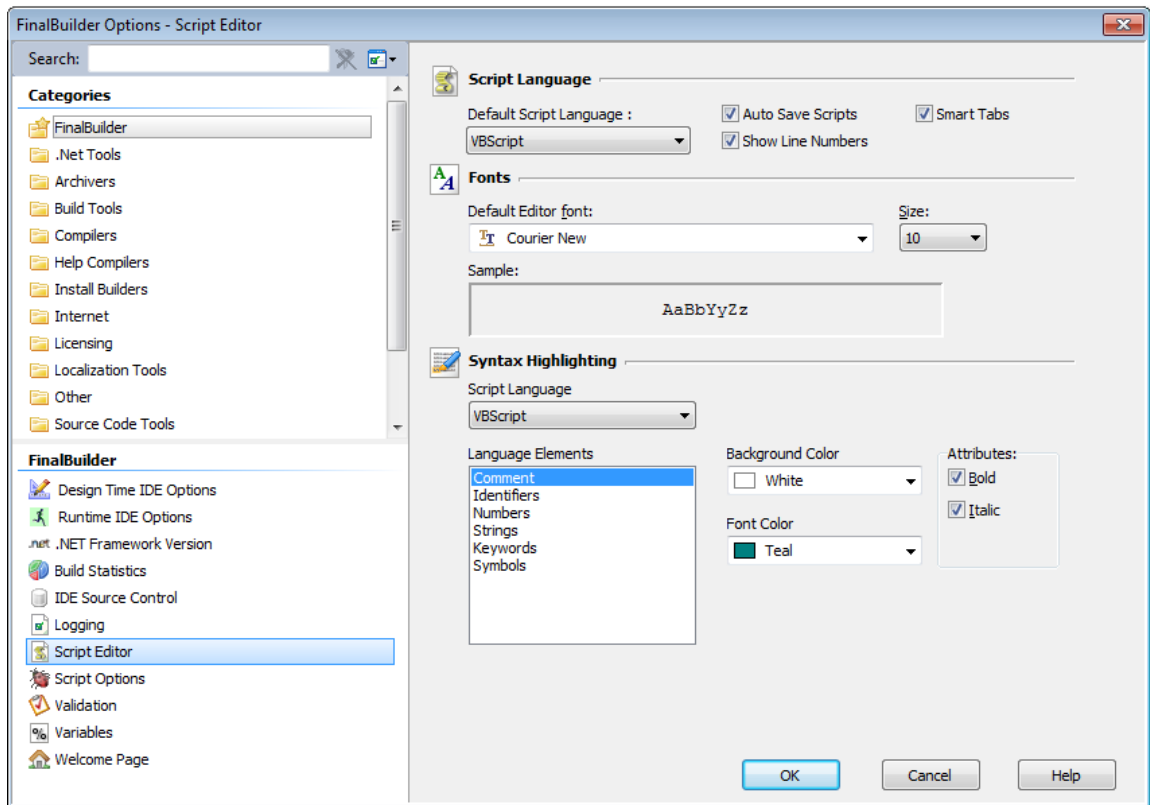
Information about IronPython can be found at the IronPython home page on Codeplex:

<http://www.codeplex.com/IronPython>

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

4.2 Script Editor Options

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



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

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

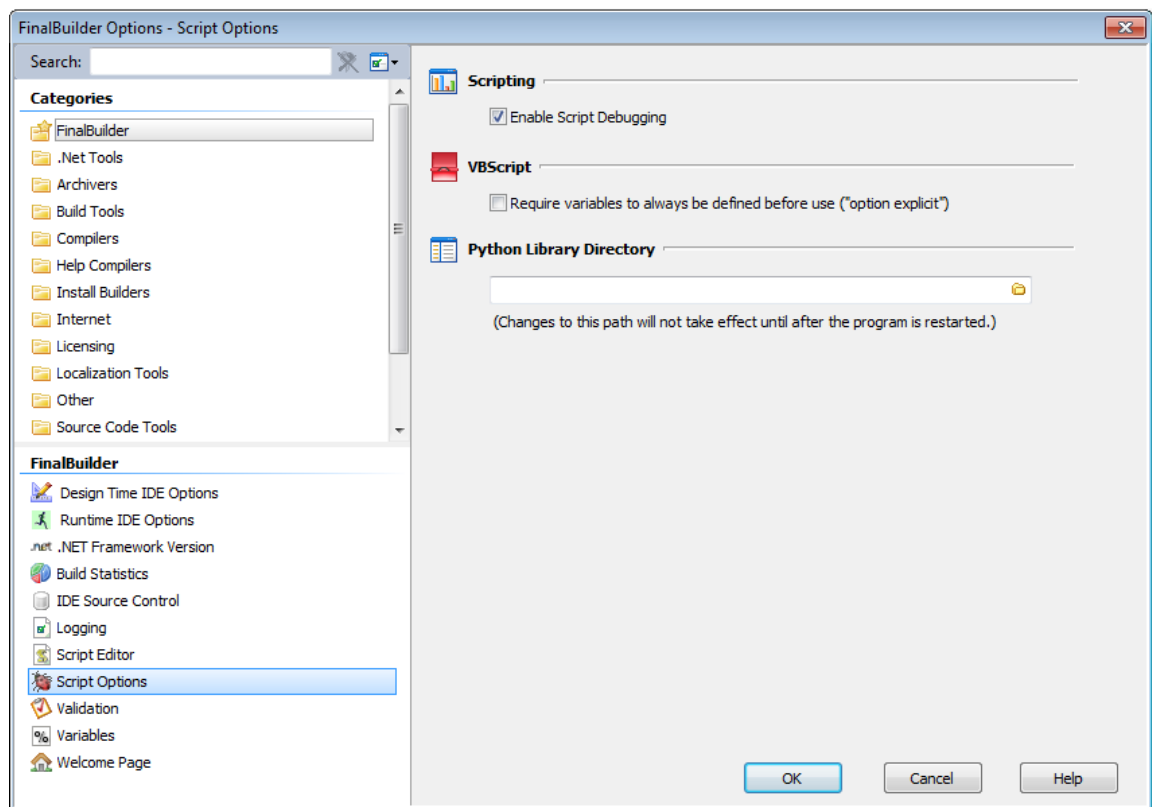
Background Color - choose the background colour of the selected language element

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



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

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, FinalBuilder exposes the following global functions :

procedure SaveProject;

Saves the current FinalBuilder 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 FinalBuilder

FBSetCaption does nothing when FinalBuilder is run from the scheduler.

function GetClipboardText : string;

Gets the text currently on the clipboard

GetClipboardText does nothing when FinalBuilder is run from the scheduler.

procedure CopyToClipboard(const value : string)

Copies the string to the clipboard

CopyToClipboard does nothing when FinalBuilder 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 specified directory if it exists.

function ExpandFileName(value : string) : string;
Expands the short filename and path to the full 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.
dddddd	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.

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

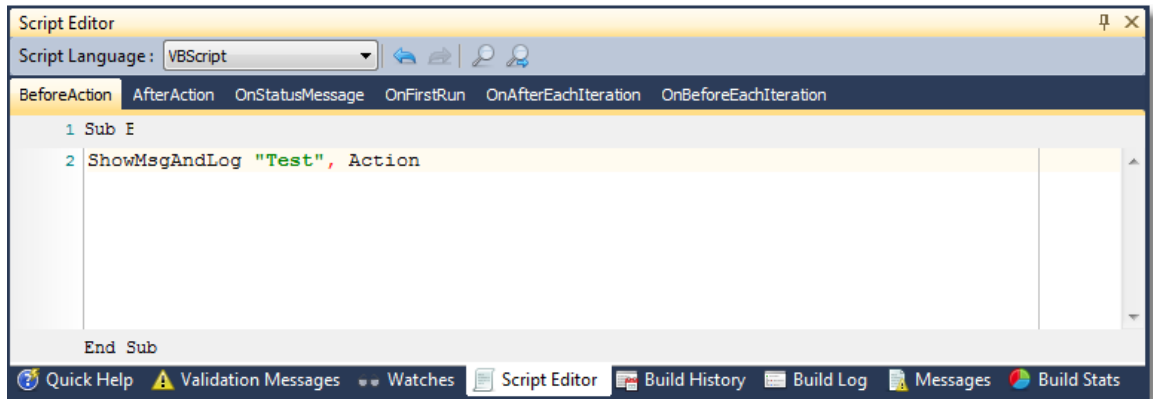
eg.

```
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 FinalBuilder 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 FinalBuilder IDE in the Script Editor tab, see screenshot.



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 build.

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 FinalBuilder 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 OnStatusMessage 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 FinalBuilder 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 : Turn off debug info on all child Delphi compiler actions... (in this example IncludeDebugInfo is a FinalBuilder 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 = "Compile Delphi Win32 Project" then
      child.CompilerOpt.DebugInfo = IncludeDebugInfo
    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

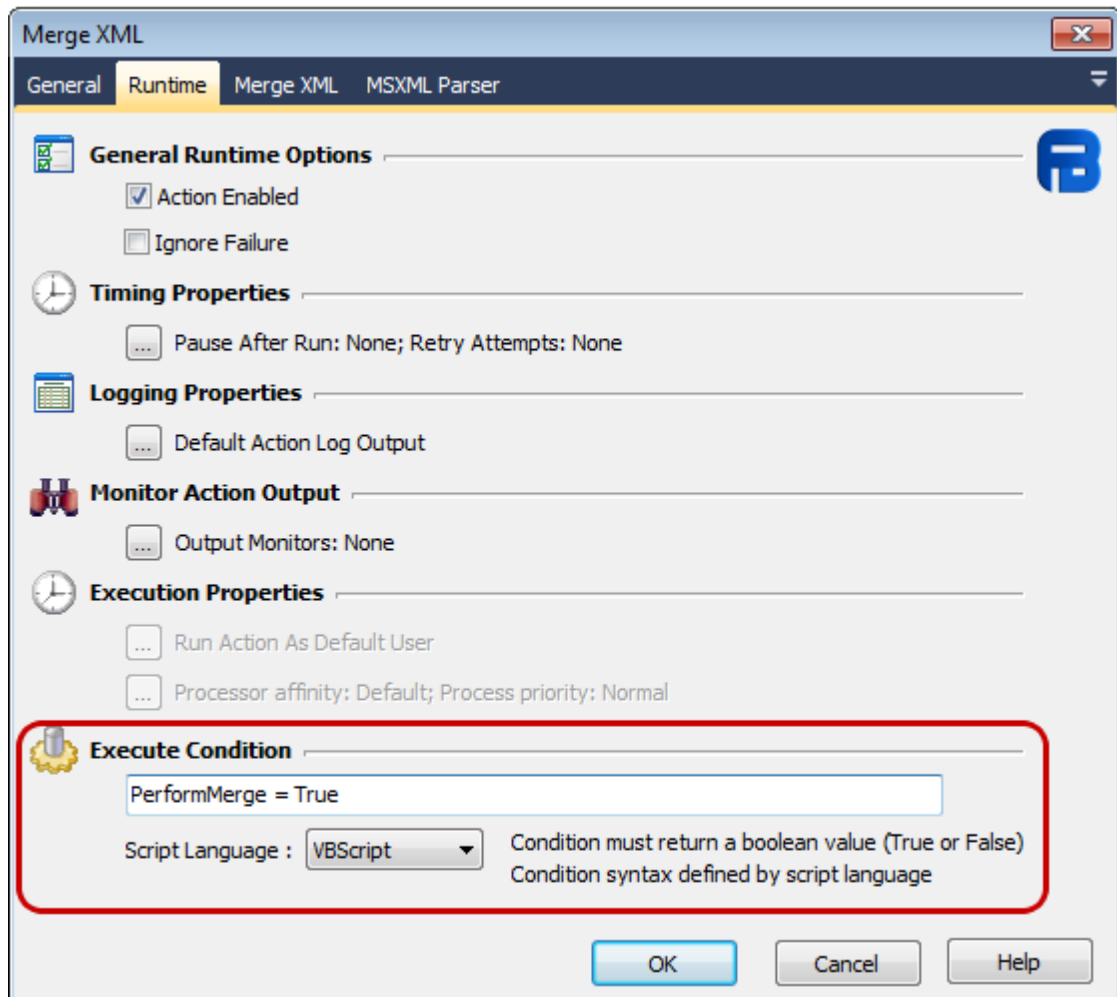
Provides access to the Action's PauseInterval property

property: **SuppressStatusMessages** As Boolean

Provides access to the Action's SuppressStatusMessages property

4.5.2 Execute Condition

Every action in FinalBuilder has an Execute Condition property:



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.

FinalBuilder Variables

Execute conditions are specified in a scripting language, so %'s are not used when referring to FinalBuilder variables. For example, if you have a FB variable "PerformMerge" then you can simply specify `PerformMerge` and not `%PerformMerge%`. In the example above, previous actions should have set the PerformMerge variable to either True or False.

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

```
(VBScript)
BuildType = "Full"
```

(JavaScript)
BuildType == "Full"

(PowerShell)
\$FBVariables.GetVariable("BuildType") -eq "Full"

where BuildType is a FinalBuilder variable.

4.6 Accessing the Options settings via scripting

FinalBuilder allows you to modify FinalBuilder options at runtime using scripting. This makes it possible to specify the path to a third party tool at runtime so that a FinalBuilder project can for example use a different version of a third party tool.

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 access the Delphi options you would use this code (VBScript example) :

```
dim delphiOptions
set delphiOptions = GetOptionsObject("Embarcadero Delphi")
delphiOptions.D6LibraryPath =delphiOptions.D6LibraryPath & ";$(DELPHI)
\Components\Lib"
```

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

FinalBuilder 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

FinalBuilder Script Editor

Name : Script Editor

- property LineNumbers: boolean
- property DefaultScriptLanguage : String
- property AutoSaveScripts : boolean

FinalBuilder 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).

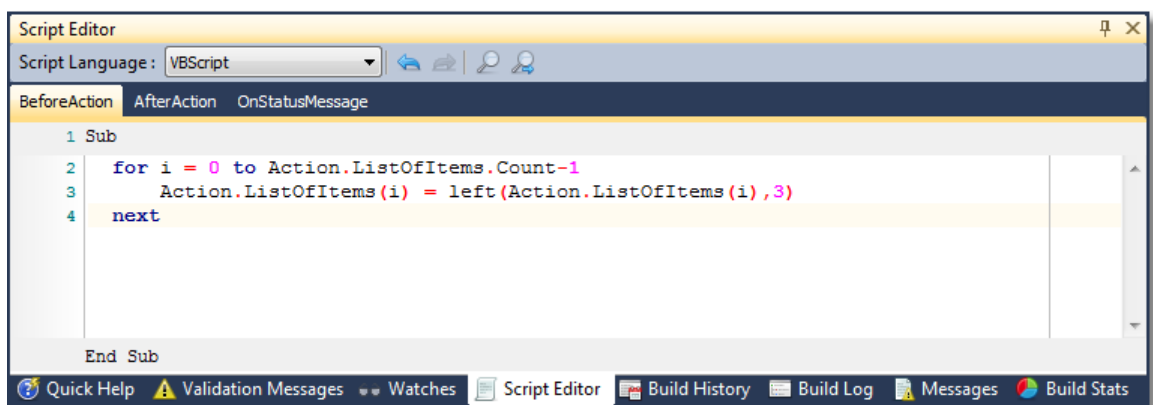
4.7 Accessing TStrings based properties

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 reduces each item in a List Iterator's list of items down to its leftmost three characters:



5 Variables

5.1 Variables Overview

Variables in FinalBuilder are the key to making your builds dynamic. Variables can be used in almost every text property of every FinalBuilder 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 FinalBuilder, 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.

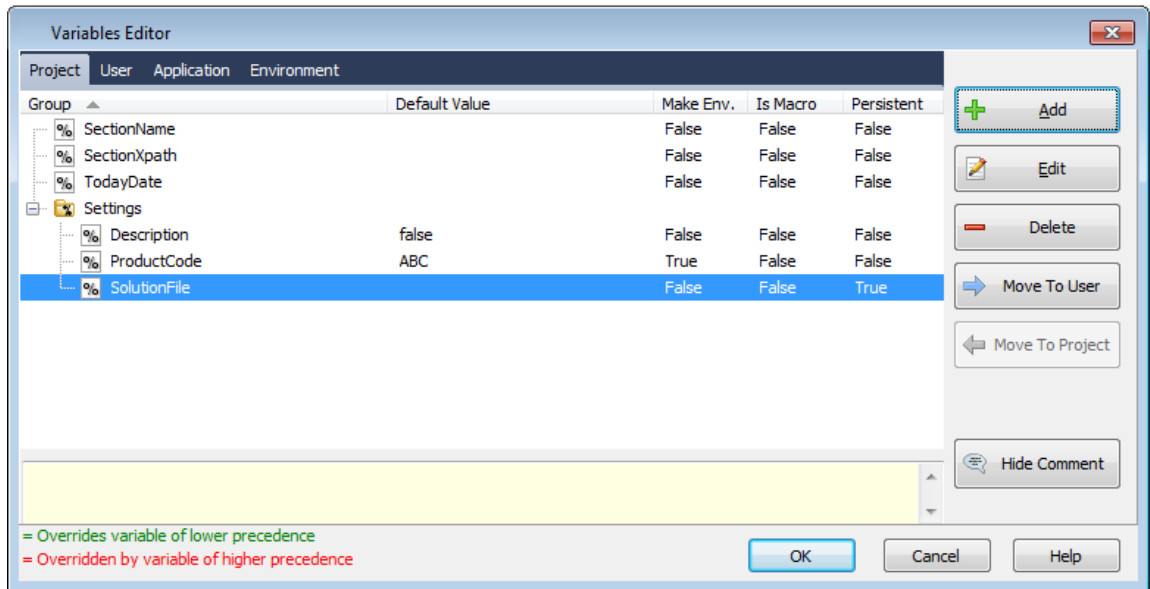
System and Environment variables cannot be edited in any way from FinalBuilder.

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 FinalBuilder can be referenced in FinalBuilder 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.



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 FinalBuilder. 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 FinalBuilder 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 build using the Set Variable action or any other means.

Persistent

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

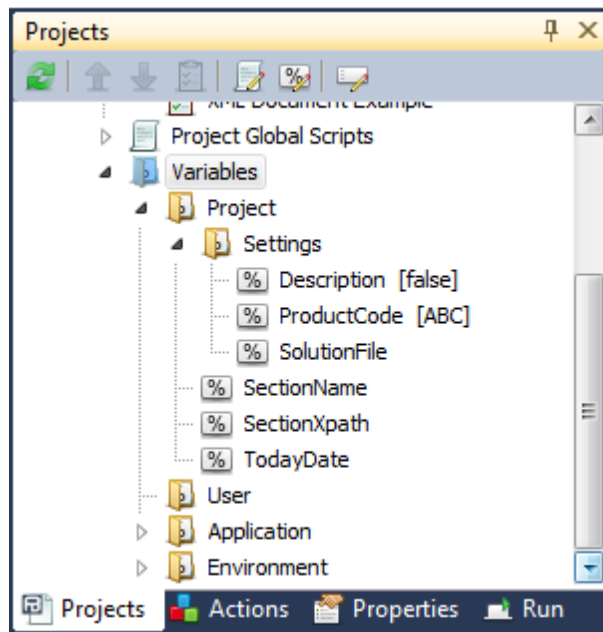
Persistent variable values are stored in .fbpinf files. Each fbpinf file has the same name as the FinalBuilder project (to persist variables, FinalBuilder needs write access to the directory containing the project and the ability to create or write to the .fbpinf file.)

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

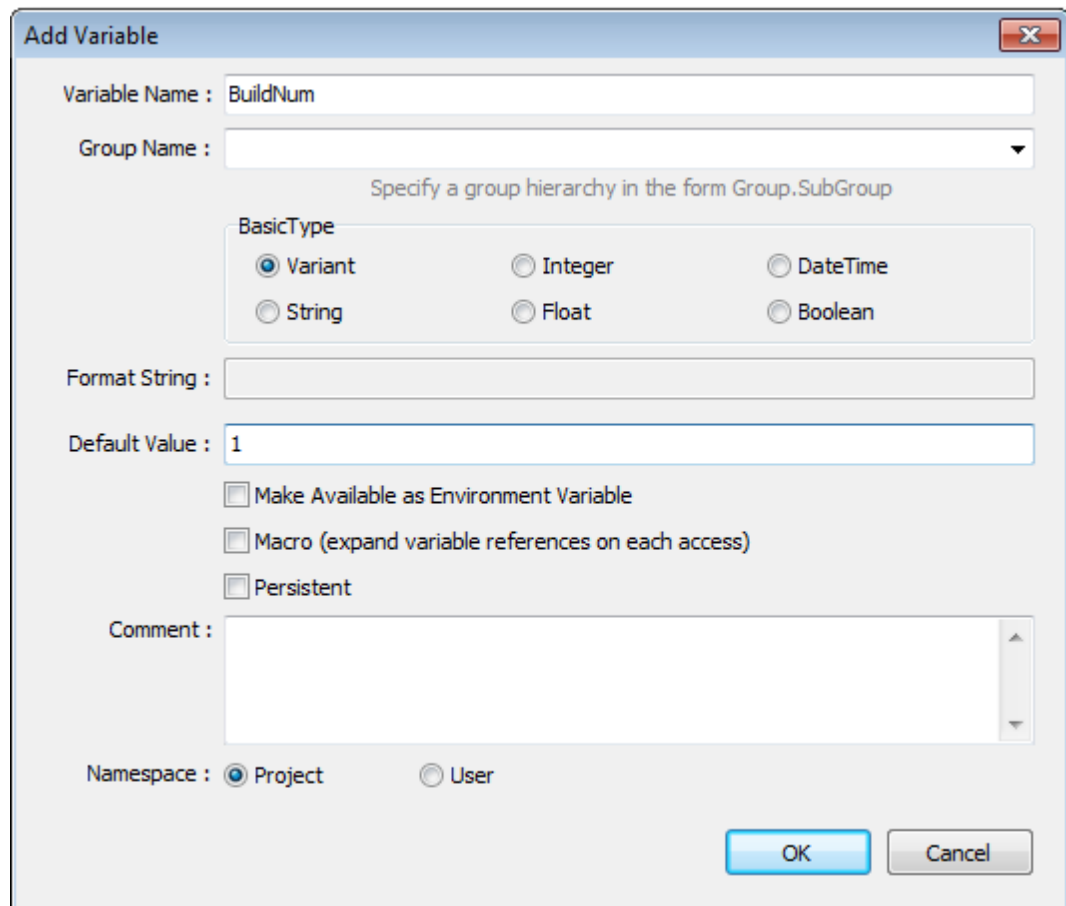
Using Variables
Project Variables
User Variables
Environment Variables
Application Variables
Action List Parameters
Escaping Variable References

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:



The 'Add Variable' dialog box is shown with the following fields and options:

- Variable Name :** BuildNum
- Group Name :** (empty dropdown menu)
- Specify a group hierarchy in the form Group.SubGroup**
- BasicType**
 - ☒ Variant
 - ☐ Integer
 - ☐ DateTime
 - ☐ String
 - ☐ Float
 - ☐ Boolean
- Format String :** (empty text field)
- Default Value :** 1
- ☐ Make Available as Environment Variable
- ☐ Macro (expand variable references on each access)
- ☐ Persistent
- Comment :** (empty text area)
- Namespace :** ☒ Project ☐ User
- Buttons:** OK, Cancel

Variable Name

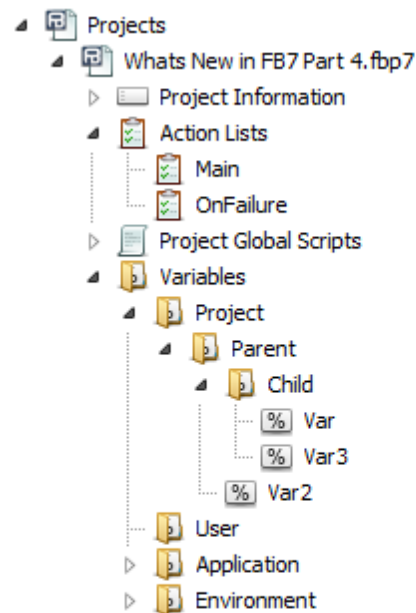
The name by which you will refer to the variable in your project.

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



Basic Type

By default, variables in FinalBuilder 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 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 FinalBuilder. 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 FinalBuilder.

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 FinalBuilder executions.

See the Variables Overview for details on the variable flags.

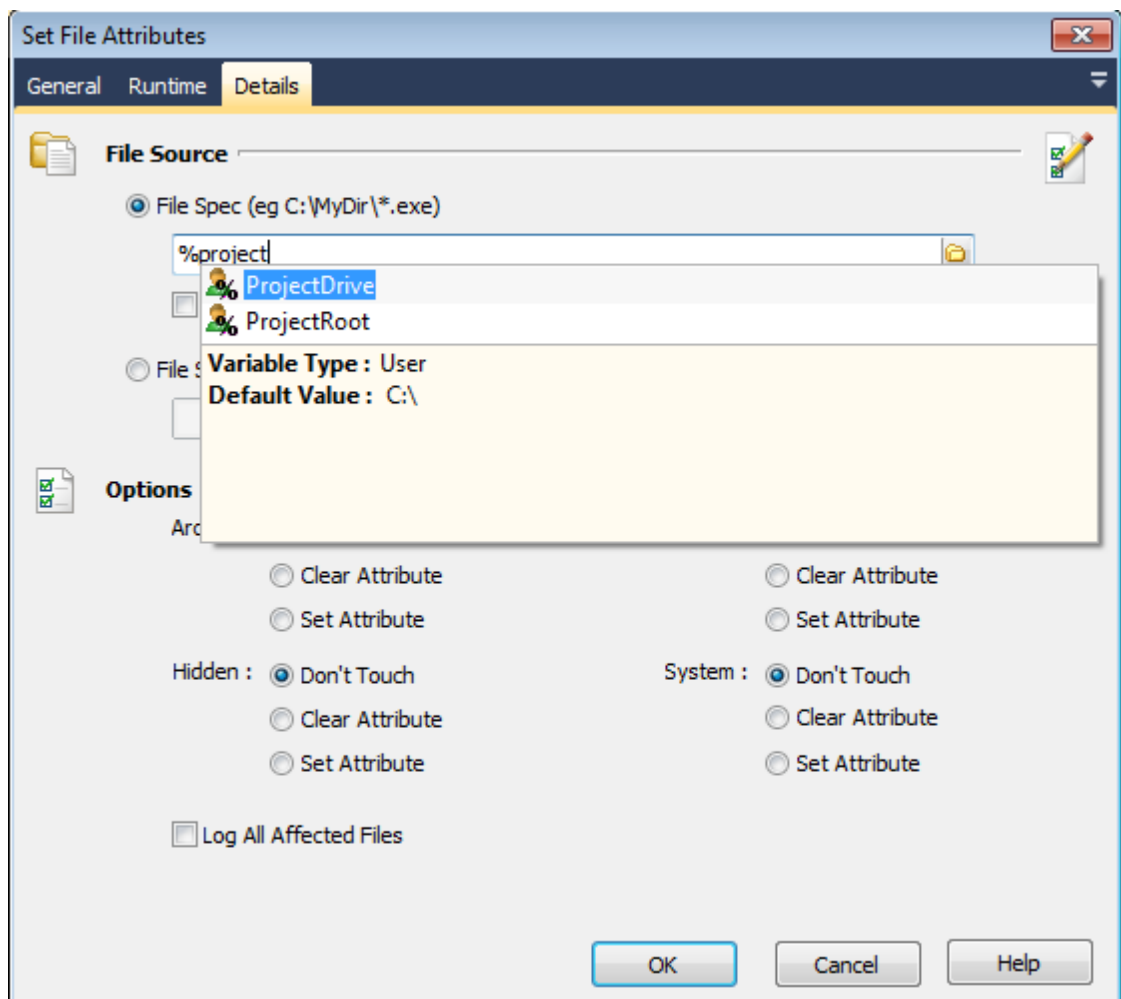
5.3 Using Variables

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:



Variables can also be used in Script Events.

See also: Variable Sense.

5.4 Variable Evaluation and Recursion

By default, when a FinalBuilder variable's value is referenced in a text field (using the %myvariable% syntax), FinalBuilder 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 FinalBuilder 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 FinalBuilder 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 be 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 .fbpinf 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\FinalBuilder\FBUserVariables.ini`

5.6.3 Environment Variables

FinalBuilder 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 FinalBuilder during a run are not saved after a run. FinalBuilder sets them temporarily and then restores them to their original values at the end of the run.

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

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

5.6.4 Application Variables

Application Variables are defined by FinalBuilder 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 FinalBuilder on.
USERNAME	The Name of the currently logged on User
DOSCMD	The path to the Dos command interpreter. This is used internally by FinalBuilder.

The following Application Variables are created by FinalBuilder:

ISAUTOBUILD	True if the build is being run automatically from a command line parameter.
ISCONSOLEBUILD	True if the build is being run from the console under FBCMD.EXE.
ISINTERACTIVEBUILD	True if the build is being run under the FinalBuilder IDE.
ISINCLUDEDPROJECT	True if the current project is being run as a child project (via the Include FinalBuilder Project Action)
FBPROJECT	The full path to the currently running FinalBuilder project file.
FBPROJECTDIR	The dir where the currently running FinalBuilder project lives.
FBDIR	The directory in which FinalBuilder is installed.
FBIGNOREDERRORS	The number of actions in error during the build that were ignored.
FBBUILDTYPE	When running in the IDE, is this a full run or a run started part-way through? Possible values are script constants btFull, btFromCurrent, btSelected or btSelectedAndChildren.
ISSERVERBUILD	True if the project is being run by FinalBuilder Server.
FBSERVERPROJECTNAME	If the project is run by FinalBuilder Server, holds the name of the project as it is defined there

These Variables may appear on your system, depending on whether you have the software installed :

VB6	The path to the Visual Basic 6 compiler
DELPHI3DIR	The directory where Delphi 3 is installed
DELPHI3	The path to the Delphi 3 command line compiler
DELPHI3_BRCC32	The path to the Delphi 3 Resource compiler
DELPHI4DIR	The directory where Delphi 4 is installed
DELPHI4	The path to the Delphi 4 command line compiler
DELPHI4_BRCC32	The path to the Delphi 4 Resource compiler
DELPHI5DIR	The directory where Delphi 5 is installed
DELPHI5	The path to the Delphi 5 command line compiler
DELPHI5_BRCC32	The path to the Delphi 5 Resource compiler

DELPHI6DIR	The directory where Delphi 6 is installed
DELPHI6	The path to the Delphi 6 command line compiler
DELPHI6_BRCC32	The path to the Delphi 6 Resource compiler
DELPHI7DIR	The directory where Delphi 7 is installed
DELPHI7	The path to the Delphi 7 command line compiler
DELPHI7_BRCC32	The path to the Delphi 7 Resource compiler
DELPHI8DIR	The directory where Delphi.NET 8 is installed
DELPHI8	The path to the Delphi 8 command line compiler (dccil.exe)
DELPHI2005	The path to the Delphi 2005 command line compiler
DELPHI2005DIR	The directory where Delphi 2005 is installed
DELPHI2005_BRCC32	The path to the Delphi 2005 Resource compiler
DELPHI2005DOTNET	The path to the Delphi 2005 .Net command line compiler (dccil.exe)
DELPHI2006DIR	The directory where Delphi 2006 is installed
DELPHI2006_BRCC32	The path to the Delphi 2006 Resource compiler
DELPHI2006DOTNET	The path to the Delphi 2006 .Net command line compiler (dccil.exe)
DELPHI2007DIR	The directory where Delphi 2007 is installed
DELPHI2007_BRCC32	The path to the Delphi 2007 Resource compiler
DELPHI2007DOTNET	The path to the Delphi 2007 .Net command line compiler (dccil.exe)
BDSUSERDIR	The Embarcadero Studio Projects user directory
AQTestErrors	The AQ Test Complete error count
AQTestWarnings	The AQ Test Complete warning count
CB2007DIR	The directory where CBuilder 2007 is installed
CB2007	The path to CBuilder 2007 make.exe
CB2007DCC	The path to the CBuilder 2007 Delphi command line compiler (dcc32.exe)
CB2007_BRCC32	The path to the CBuilder 2007 command line compiler (brcc32.exe)
BCB2006DIR	The directory where CBuilder 2006 is installed
BCB2006	The path to CBuilder 2006 make.exe
BCB2006DCC	The path to the CBuilder 2006 Delphi command line compiler (dcc32.exe)
BCB2006_BRCC32	The path to the CBuilder 2006 command line compiler (brcc32.exe)
BCB3DIR	The directory where CBuilder 3 is installed
BCB3	The path to CBuilder 3 make.exe
BCB3DCC	The path to the CBuilder 3 Delphi command line compiler (dcc32.exe)
BCB3_BRCC32	The path to the CBuilder 3 command line compiler (brcc32.exe)
BCB4DIR	The directory where CBuilder 4 is installed
BCB4	The path to CBuilder 4 make.exe
BCB4DCC	The path to the CBuilder 4 Delphi command line compiler (dcc32.exe)
BCB4_BRCC32	The path to the CBuilder 4 command line compiler (brcc32.exe)
BCB5DIR	The directory where CBuilder 5 is installed
BCB5	The path to CBuilder 5 make.exe
BCB5DCC	The path to the CBuilder 5 Delphi command line compiler (dcc32.exe)
BCB5_BRCC32	The path to the CBuilder 5 command line compiler (brcc32.exe)

BCB6DIR	The directory where CBuilder 6 is installed
BCB6	The path to CBuilder 6 make.exe
BCB6DCC	The path to the CBuilder 6 Delphi command line compiler (dcc32.exe)
BCB6_BRCC32	The path to the CBuilder 6 command line compiler (brcc32.exe)
DOCOMATICDIR	The directory where Doc-O-Matic is installed
DOCOMATIC	The path to the Doc-O-Matic command line compiler

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.

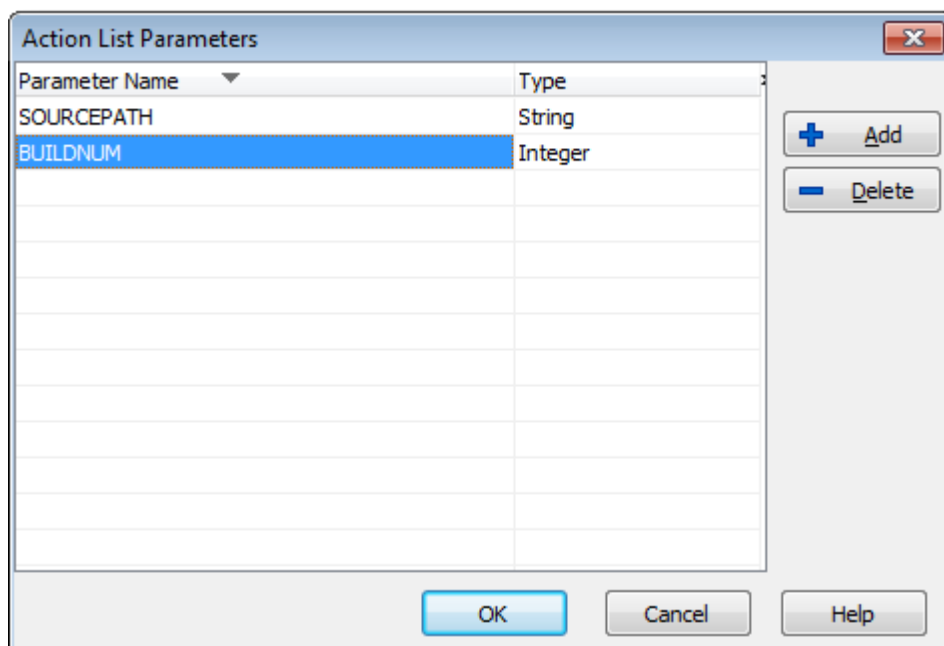
The scope of an action list parameter is confined to the actions within the Action List. Action List Parameters can be modified from script in the same manner as FinalBuilder variables.

Action List Parameters can be accessed anywhere that FinalBuilder variables can be used. Action List Parameters take precedence over FinalBuildervariables 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.

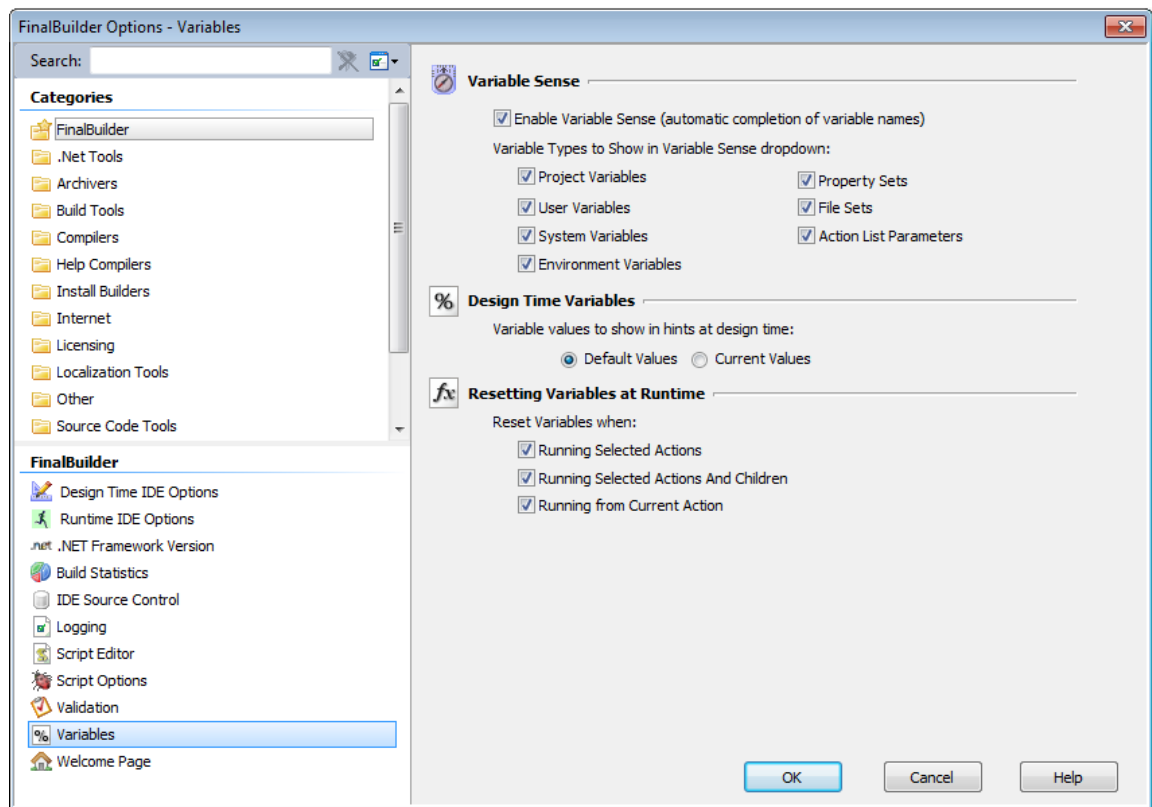


In the above example, the variables %SOURCEPATH% and %BUILDNUM% are available to any actions in the action list with these parameters.

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:

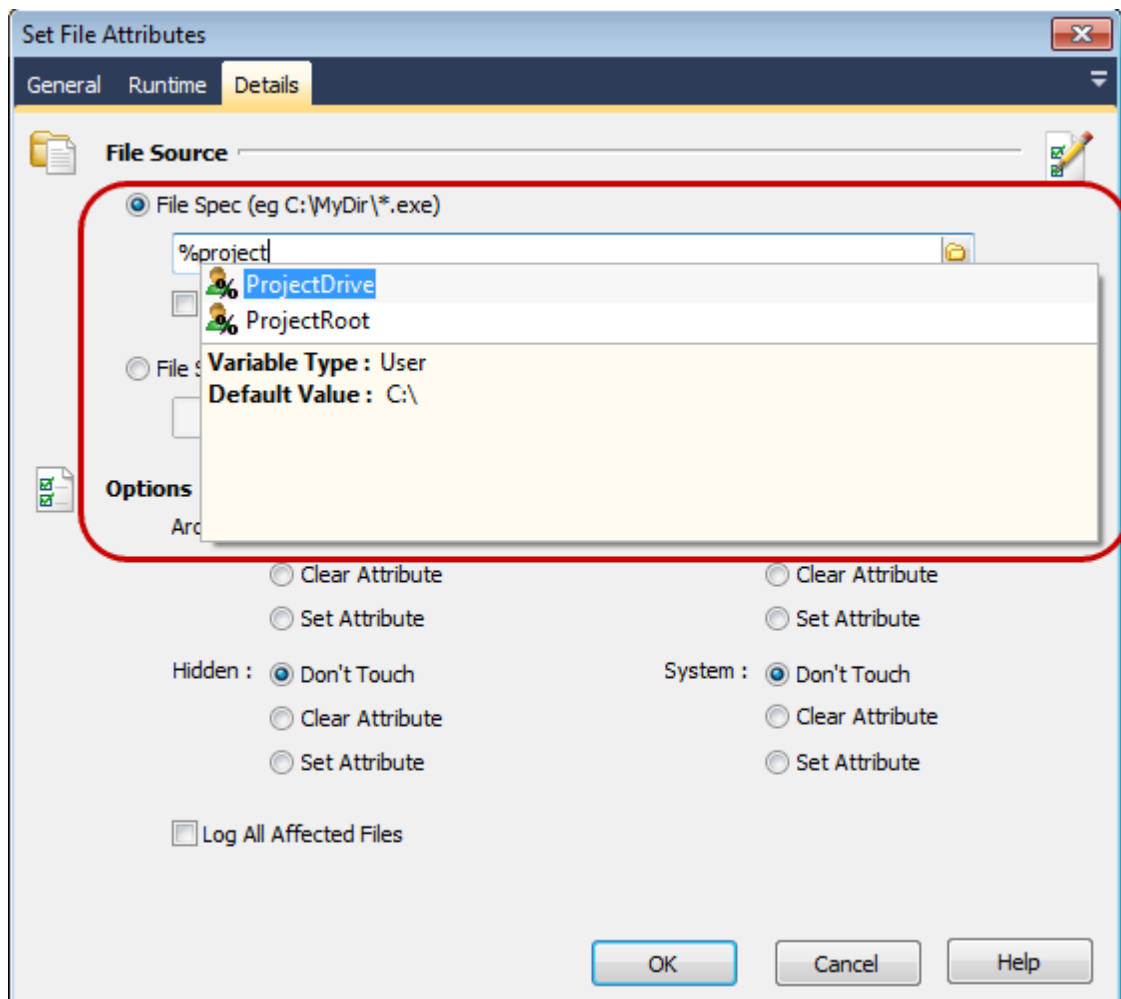
5.8 Variable Options

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



Variable Sense

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



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.

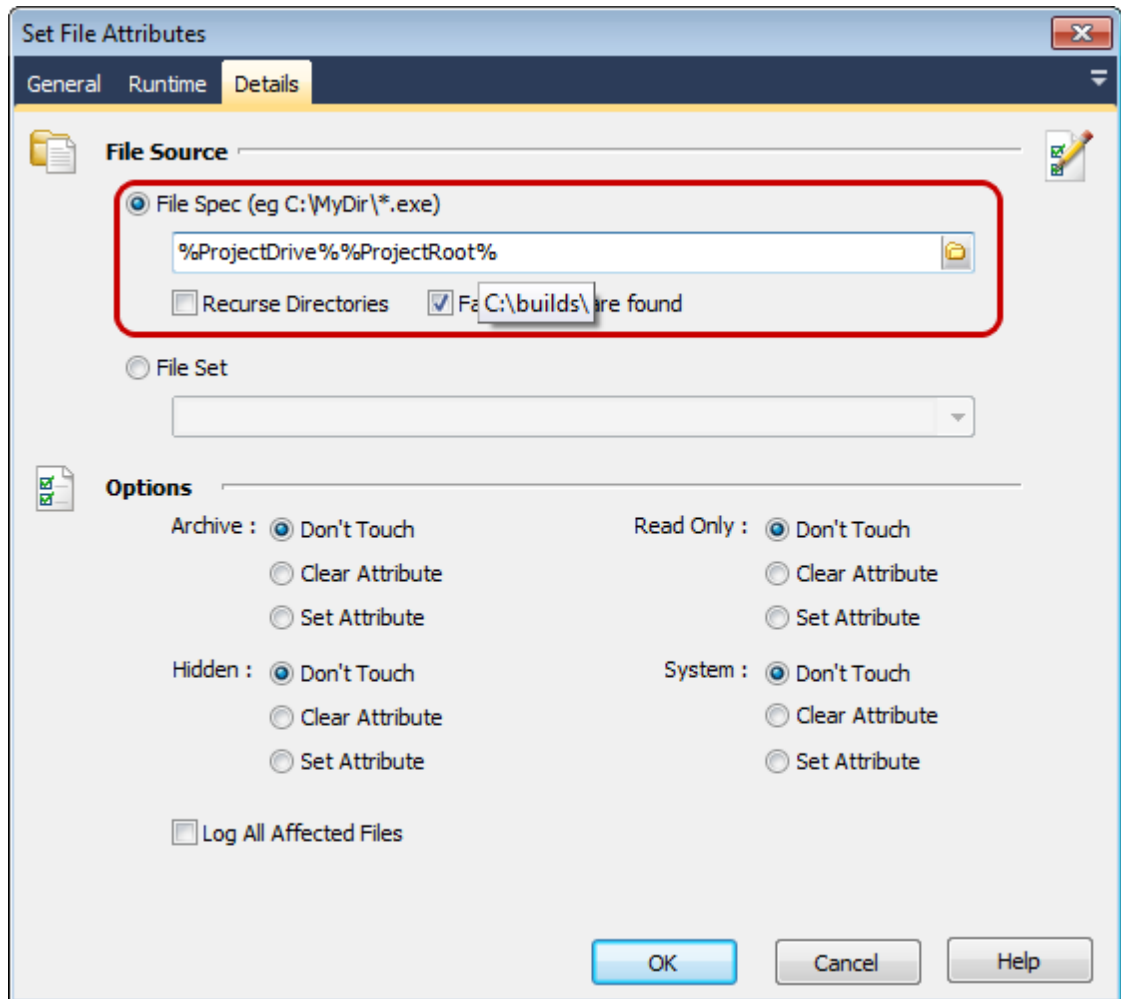
By default, all variable types are shown, however you can narrow this to only include certain variable types, and/or other items such as Property Sets, File Sets and Action List Parameters.

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:



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 "-" or 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.
- dddddd** 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?

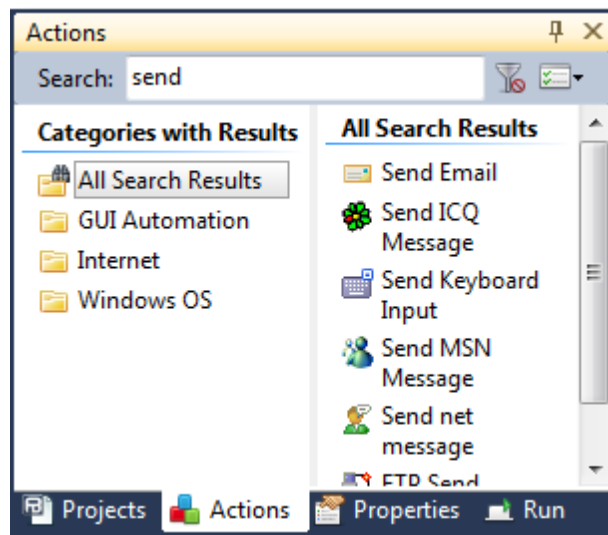
Actions are the cornerstone of FinalBuilder. FinalBuilder has built in actions to do things such as Copy Files, Move files, run programs etc. In addition, FinalBuilder has actions to interface with a number of third party products such as Embarcadero Delphi, Microsoft Visual Studio, Subversion and InstallShield Professional.

Action Lists

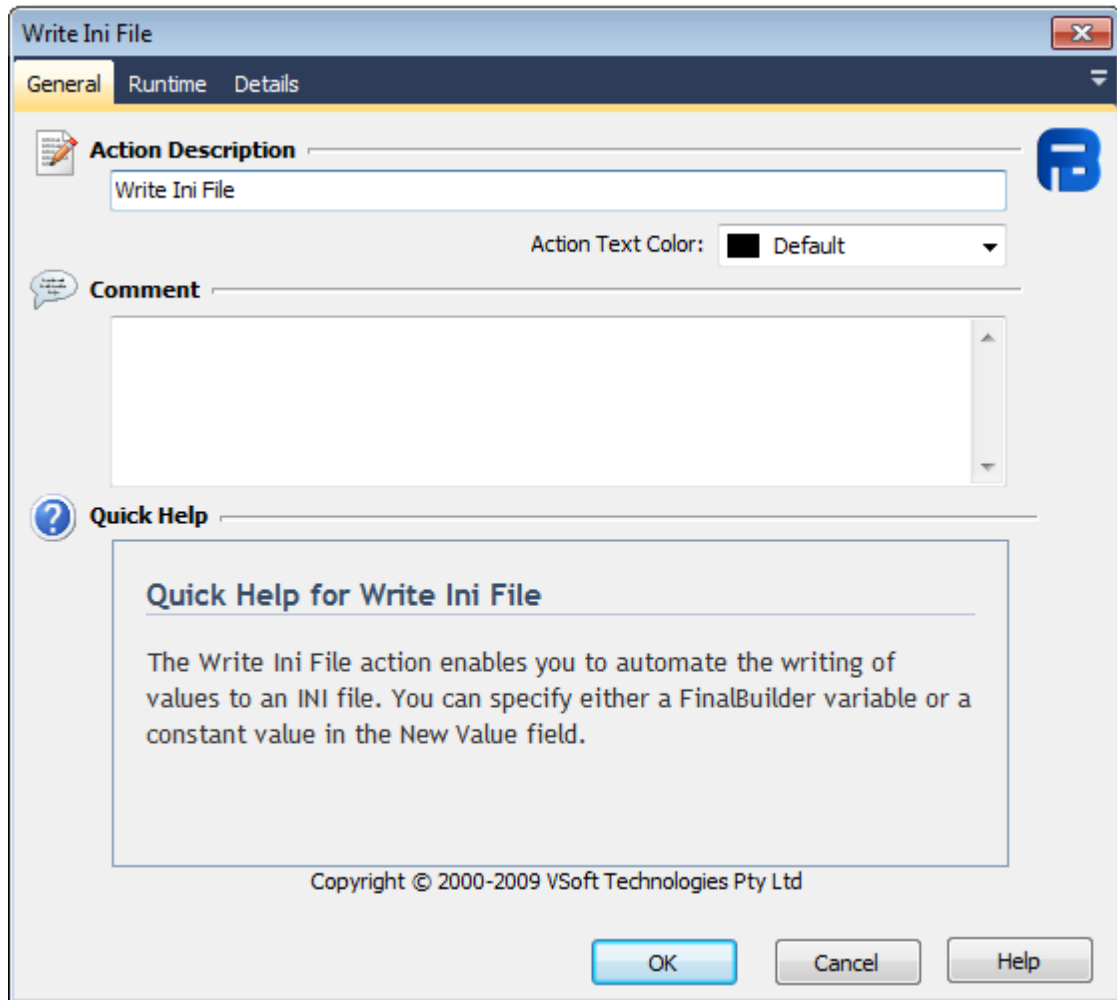
FinalBuilder has two default Action Lists that you can add actions to, the Main Action List and the OnFailure Action List. When you are creating your build, you will start off by adding actions to the Main Action list. See the Action Lists topic for more information.

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.

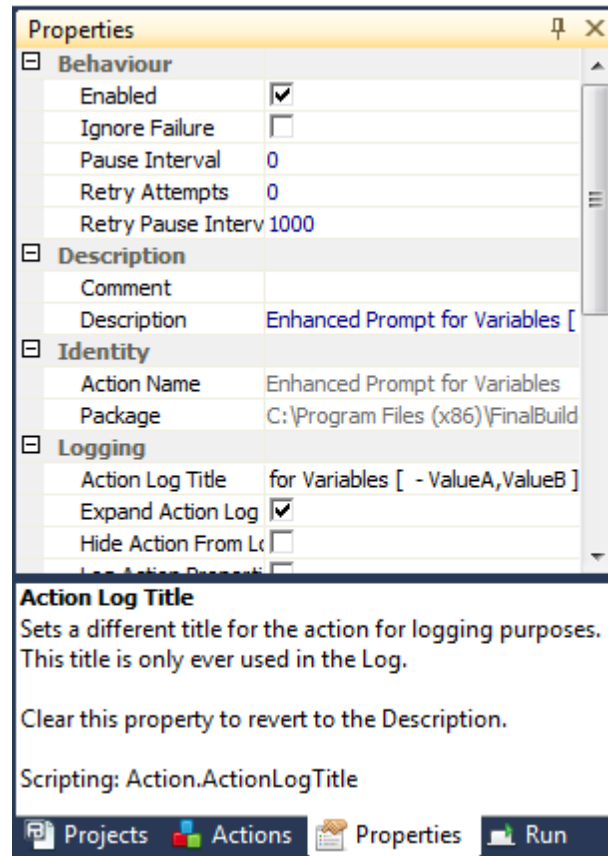


After clicking on the action type, by default the properties dialog will be displayed:



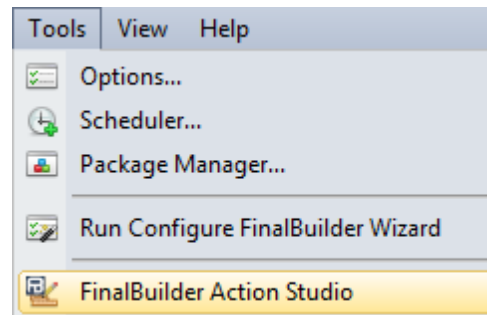
All actions have a common set of properties, displayed on the Properties tab. The number of additional tabs in this dialog will depend on which action type you selected.

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

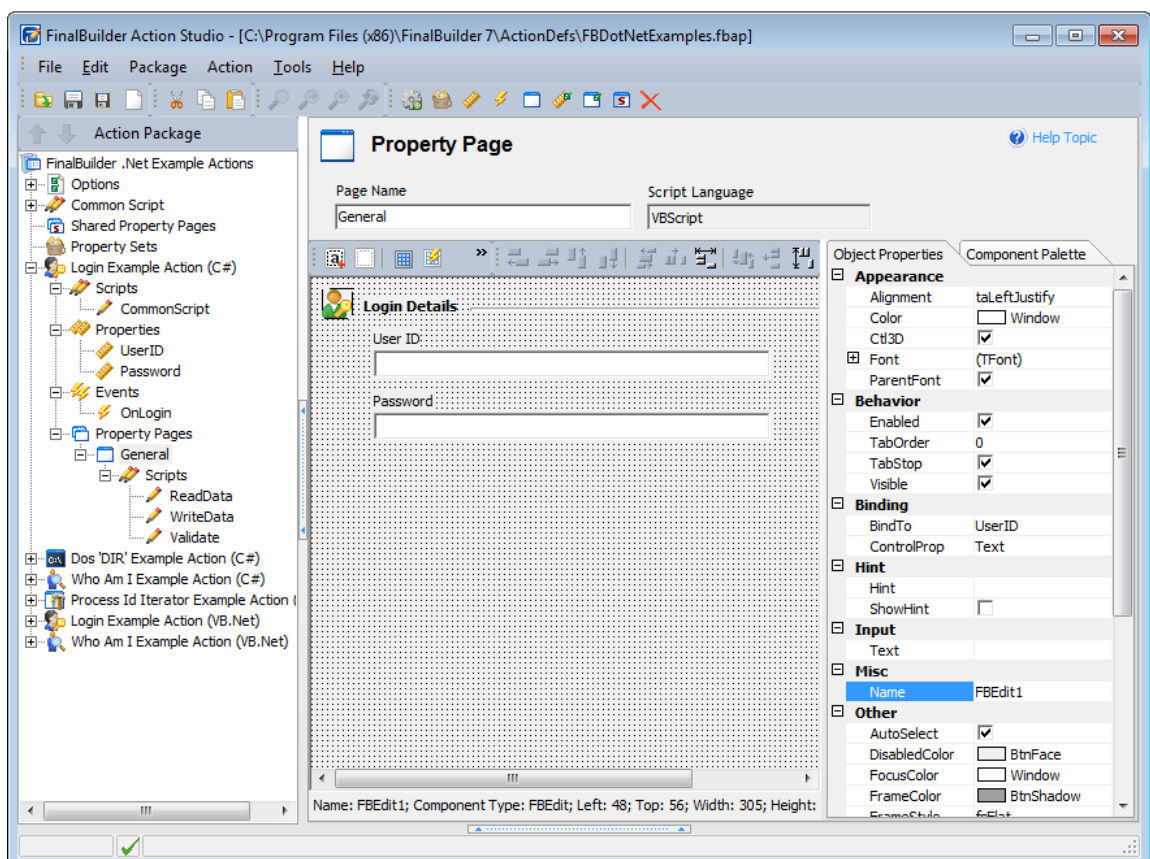


6.1 Custom Actions

FinalBuilder 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 FinalBuilder.



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 FinalBuilder main directory.

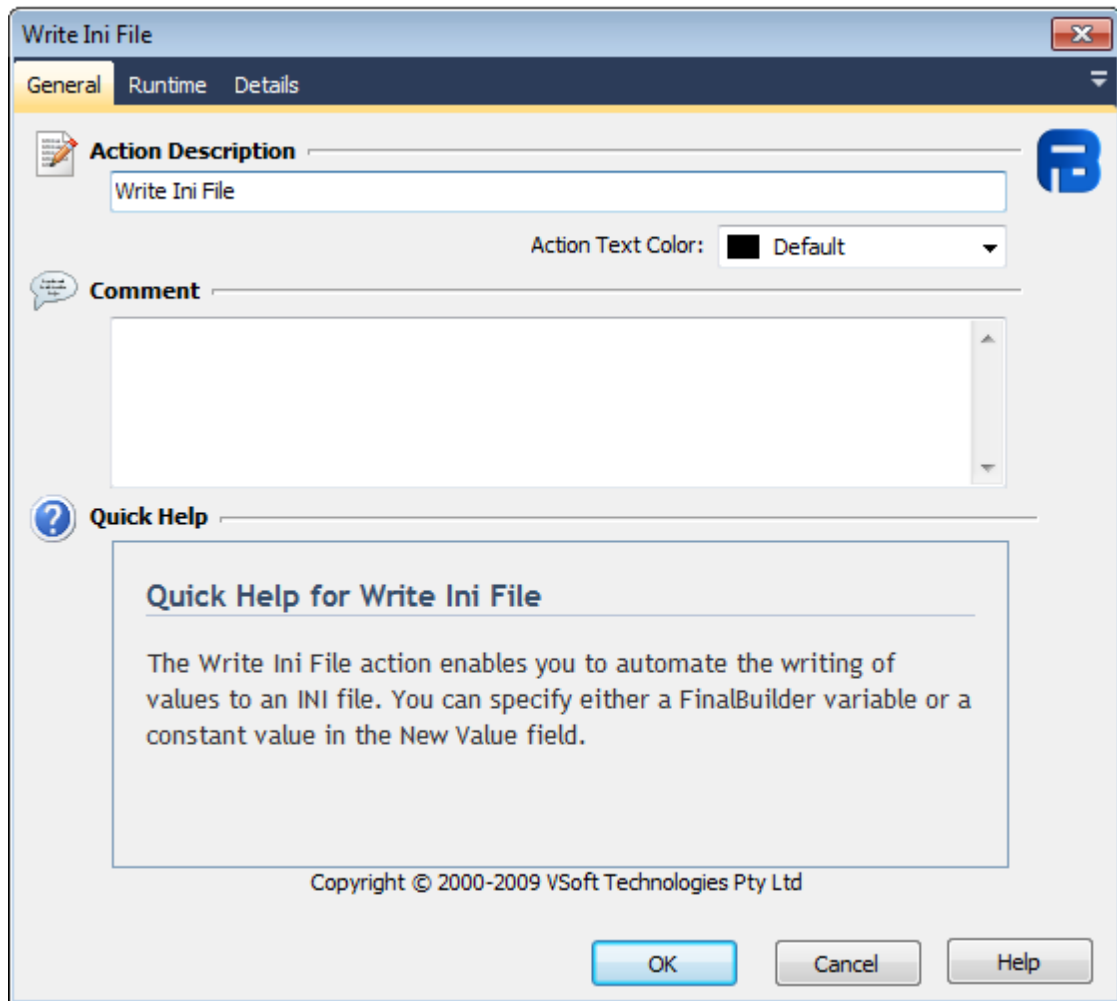


6.2 Common Action Properties

All FinalBuilder 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

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.

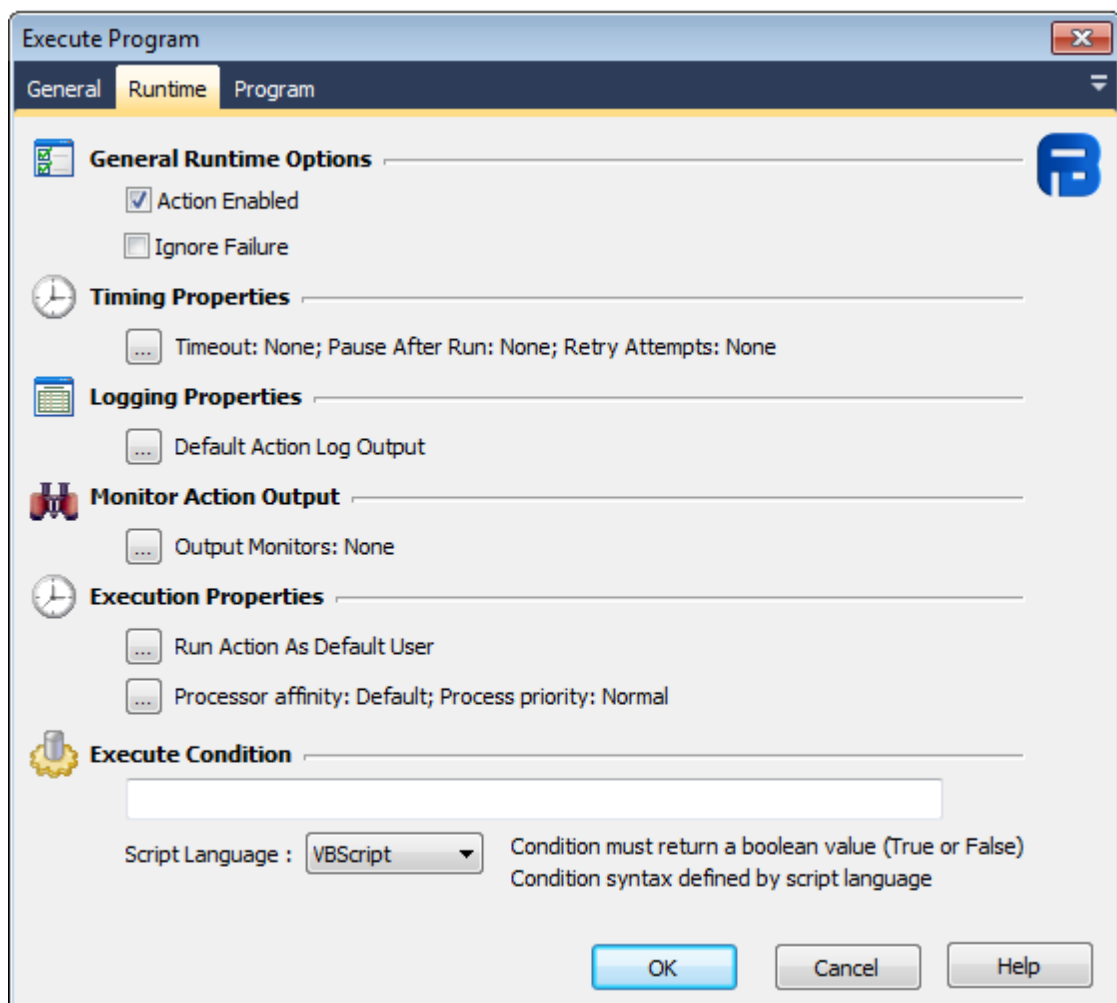
Quick Help

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.



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, and only when using FinalBuilder Professional Edition.

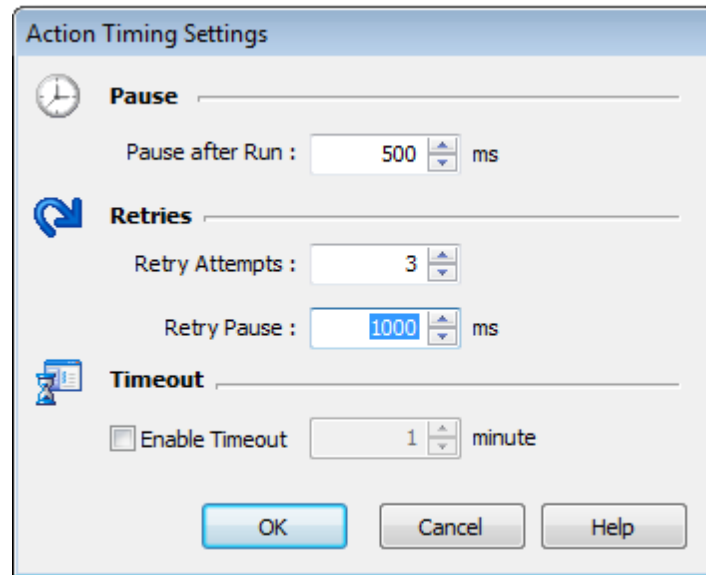
See the property pages for Run Action as User, and Setting Processor Affinity and Priority, for more information.

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

After the action completes, the build 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.

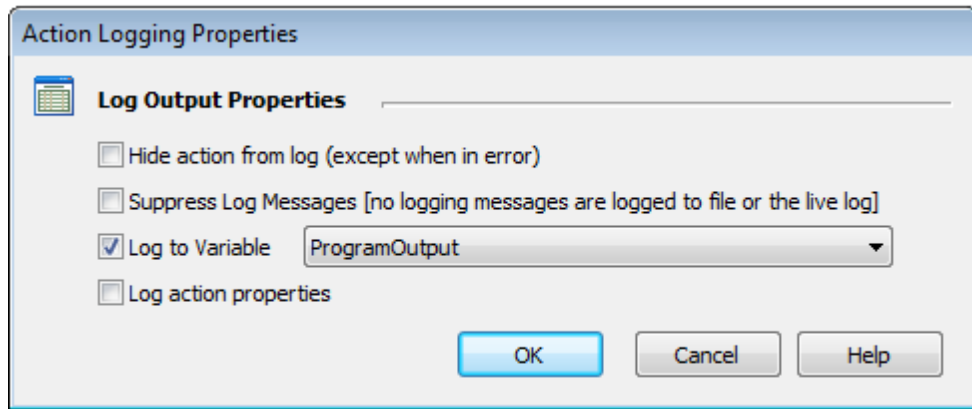
Set the Retry Pause to delay between each retry. Setting Retry on "Try" actions allows you to retry an entire group of actions as one. See the Try action topic for more details.

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.



Hide action from log

If this property is enabled, the action not appear at all in the FinalBuilder 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 FinalBuilder variable specified. This option is not available if "Suppress Log Messages" is enabled.

Log action properties

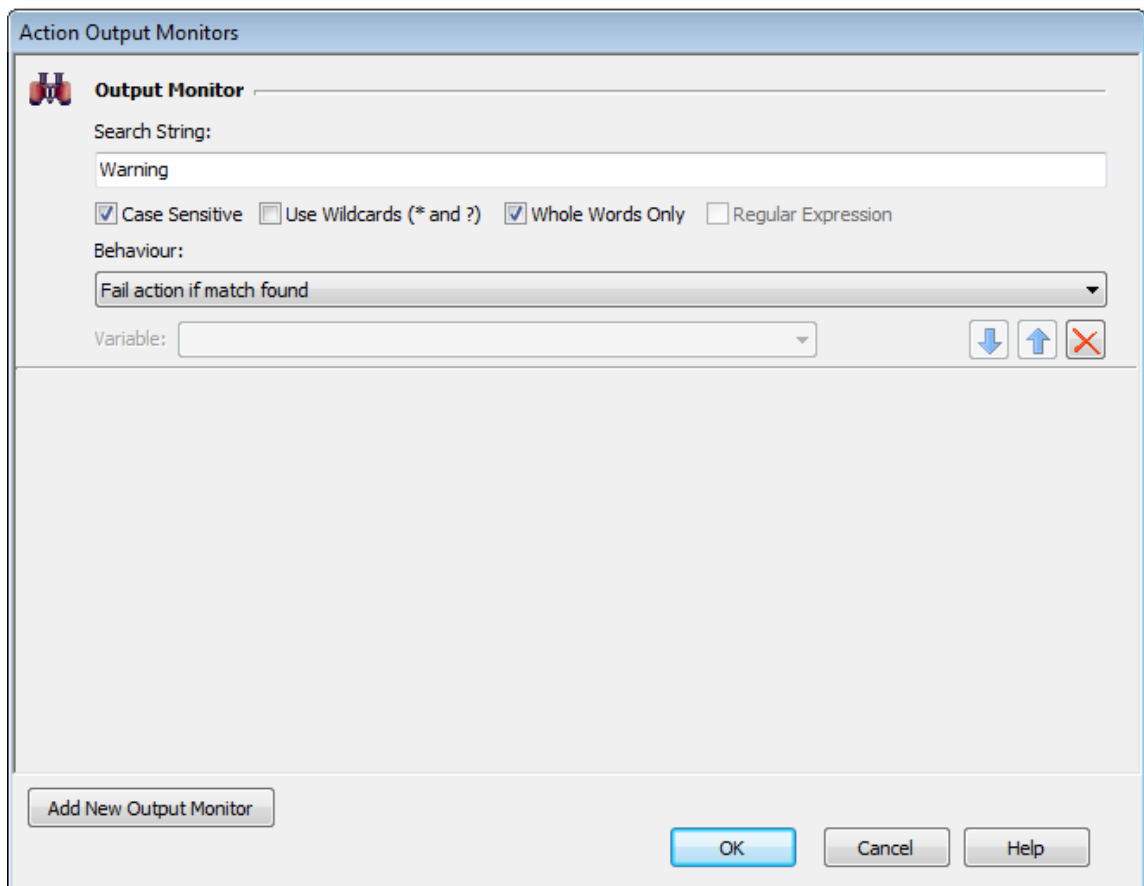
If this property is enabled, then all of the action's properties will be written to the log before the action begins running. The properties will be written in a special log message group called "Action Properties". This is useful for debugging. This option is not available if "Suppress Log Messages"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



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.

Whole Words Only

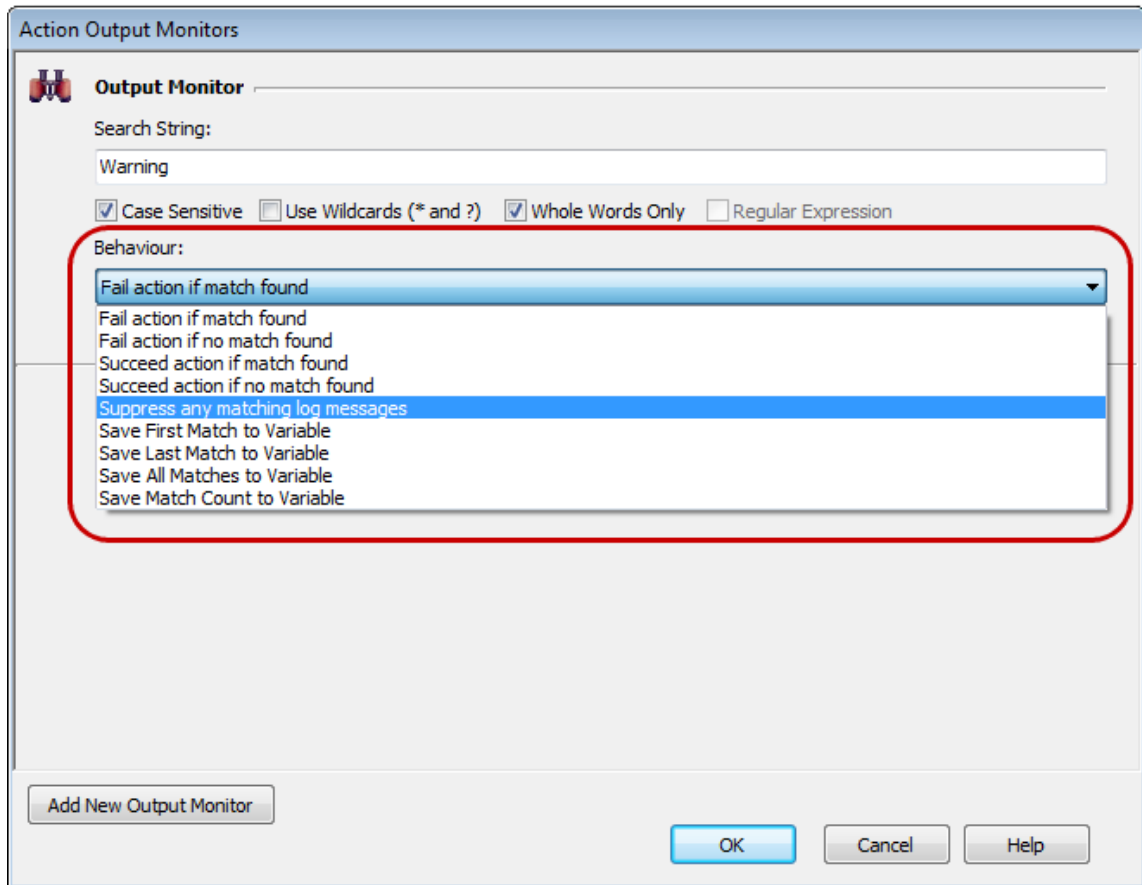
If this box is checked, the search string must match a whole word - not part of a word.

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.



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 further modify the action result.

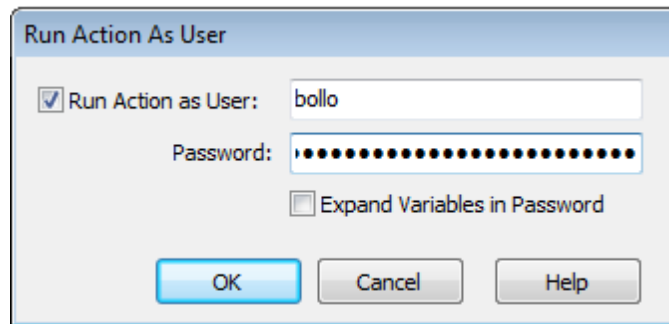
Note: Monitors Are Applied Live

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

[FinalBuilder Professional Edition only]

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

By setting the Username and Password, FinalBuilder 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.

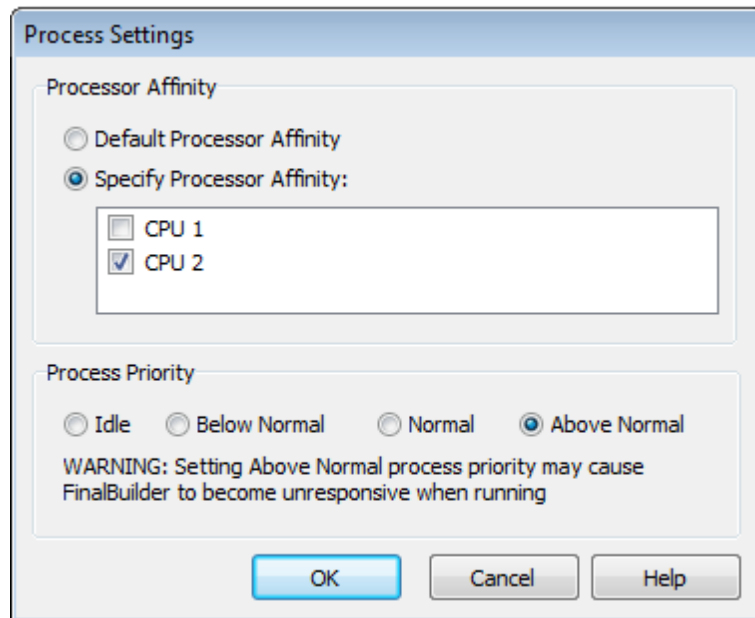
The application failed to initialize properly (0xc0000142). Click on OK to terminate the application.

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

[FinalBuilder Professional Edition only]

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.



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 FinalBuilder. Unless you are running FinalBuilder 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. FinalBuilder 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. FinalBuilder will not change the priority of these programs.

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

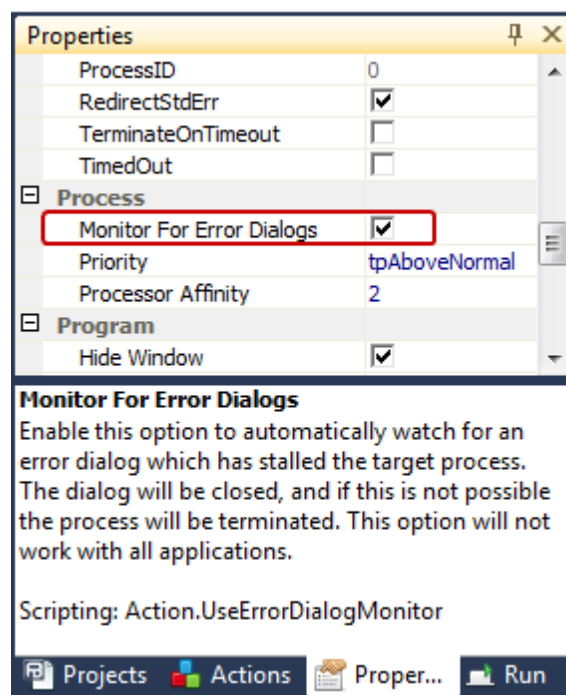
6.2.6 Error Dialog Monitor

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

Sometimes when FinalBuilder 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.



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, FinalBuilder will attempt to close the dialog. If the dialog cannot be closed, the process will be terminated.

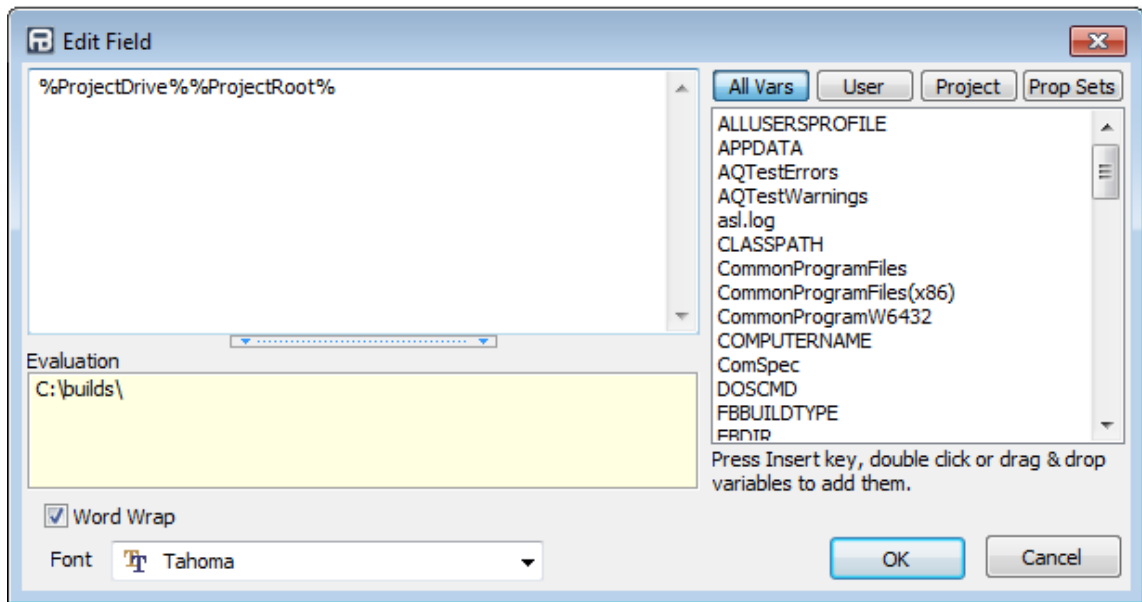
FinalBuilder will also attempt to record any text which was visible in the dialog before it was closed. This approach works for some dialogs and not for others, depending on the target application.

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 FinalBuilder 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.



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)

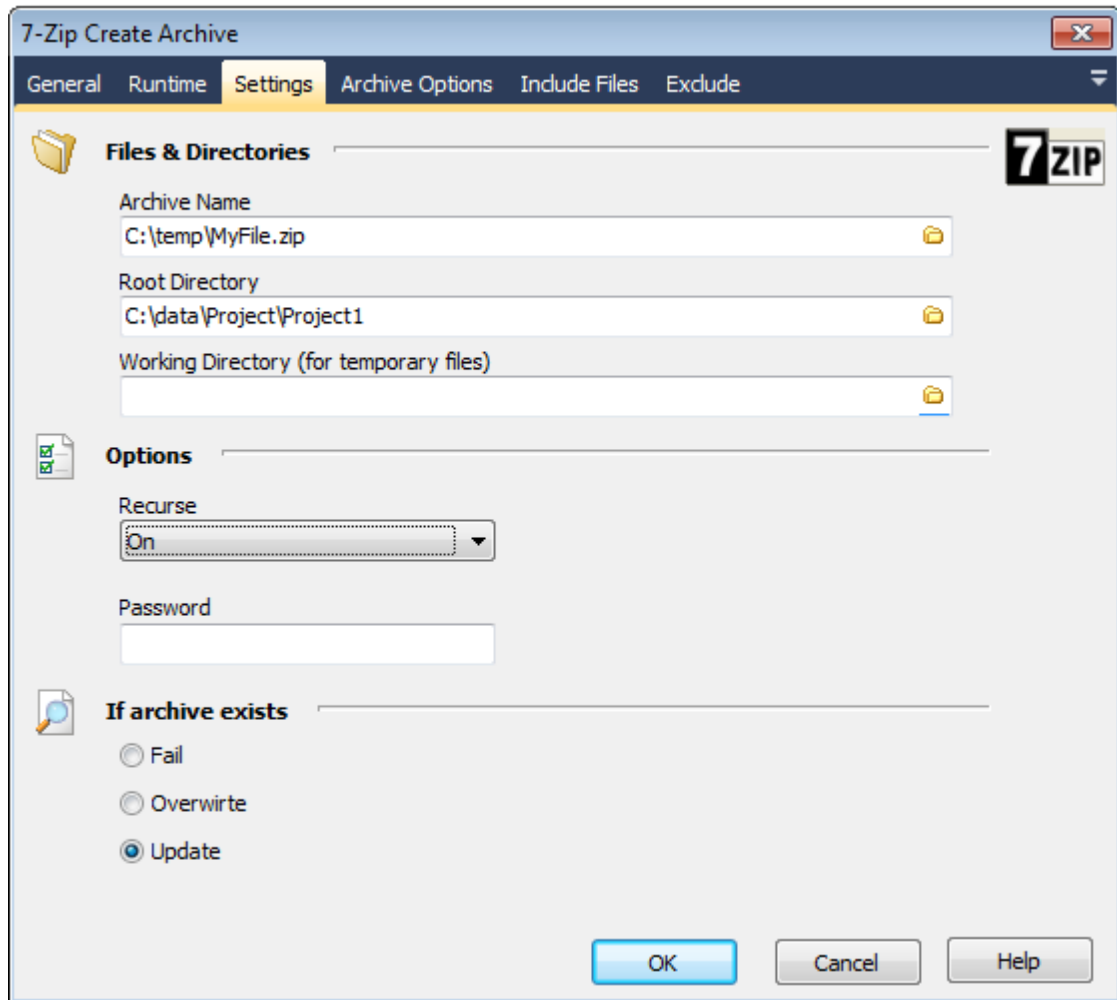
These actions require the 7Zip tool, available at <http://www.7-zip.org/>

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



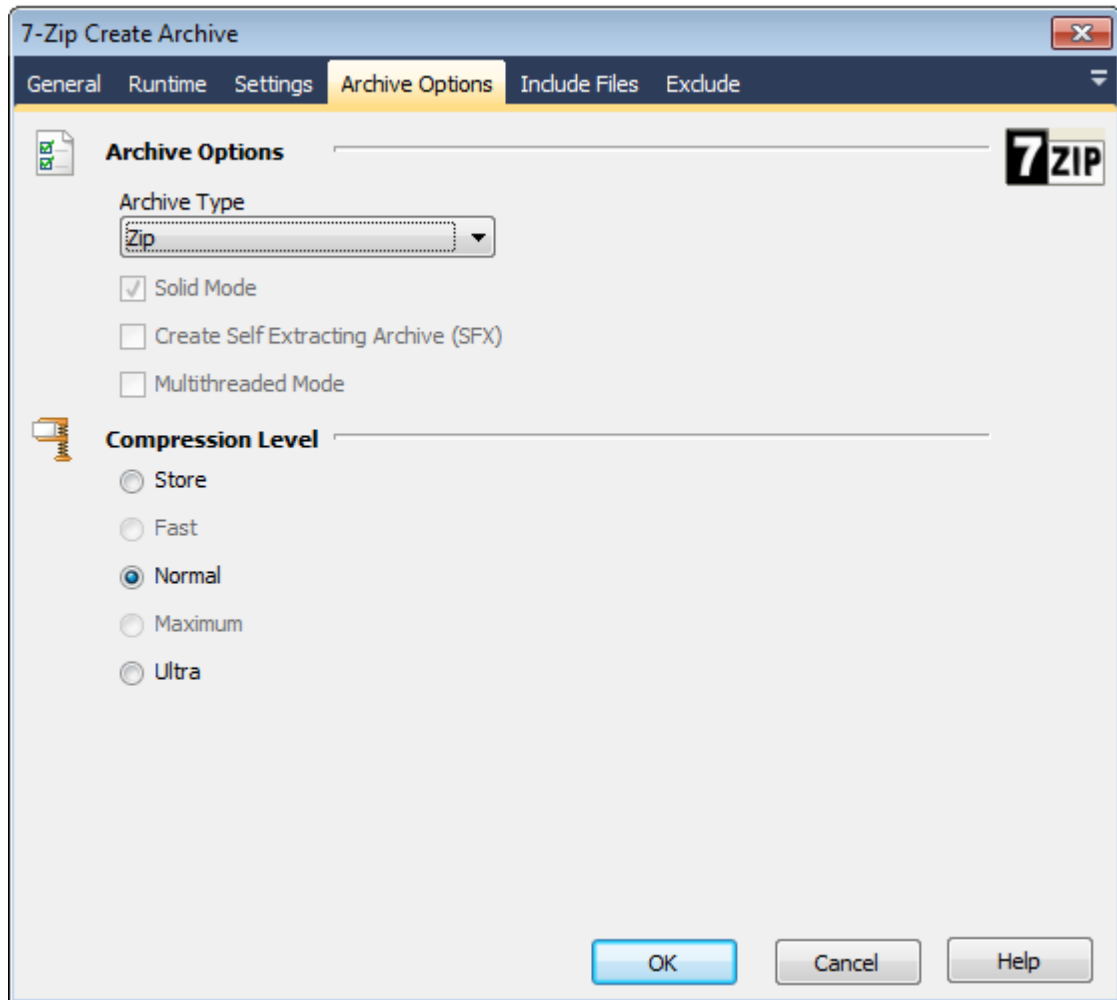
Archive Name - specify the name of the new archive. The file extension should match the type of archive you are creating (eg. ZIP)

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

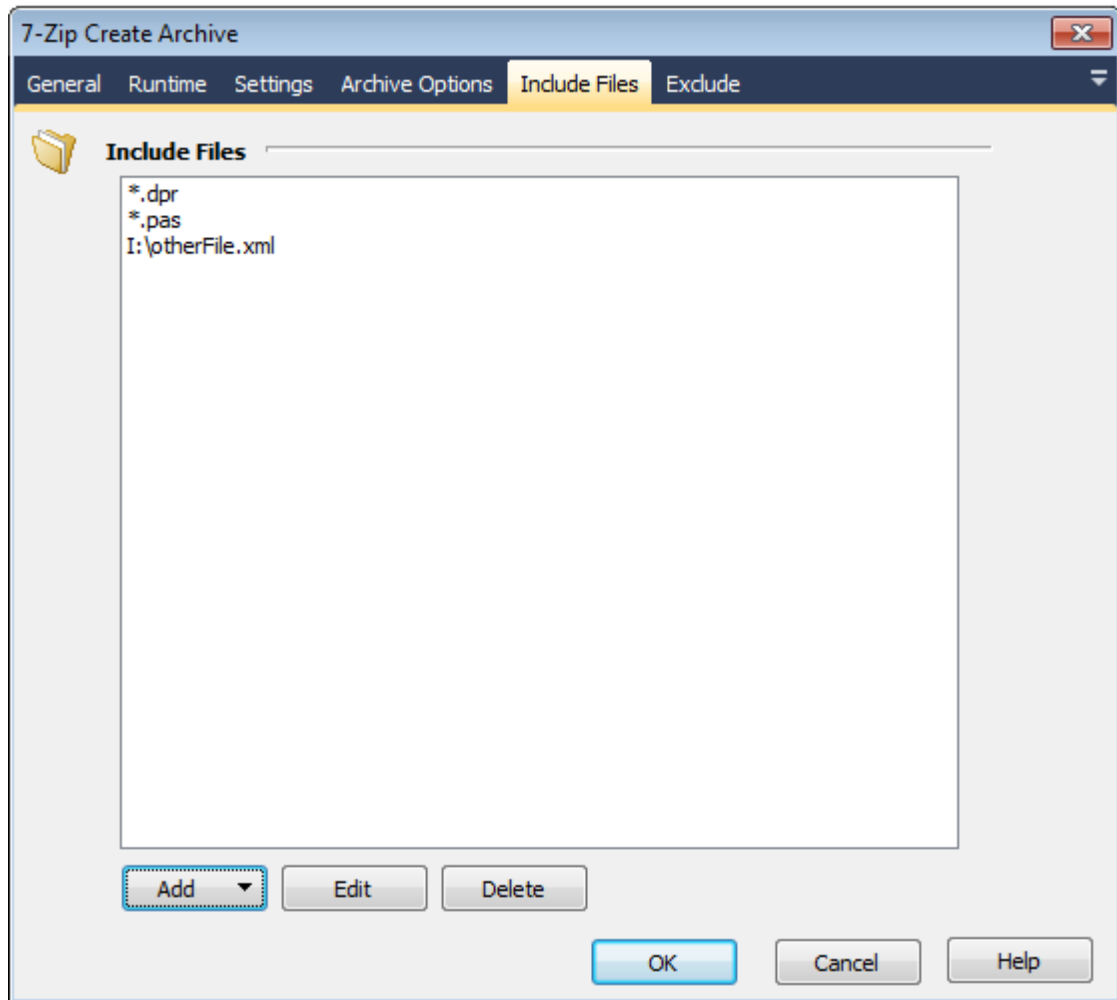
Password - specify a password to protect the archive. You'll need to supply this password to decrypt the archive



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

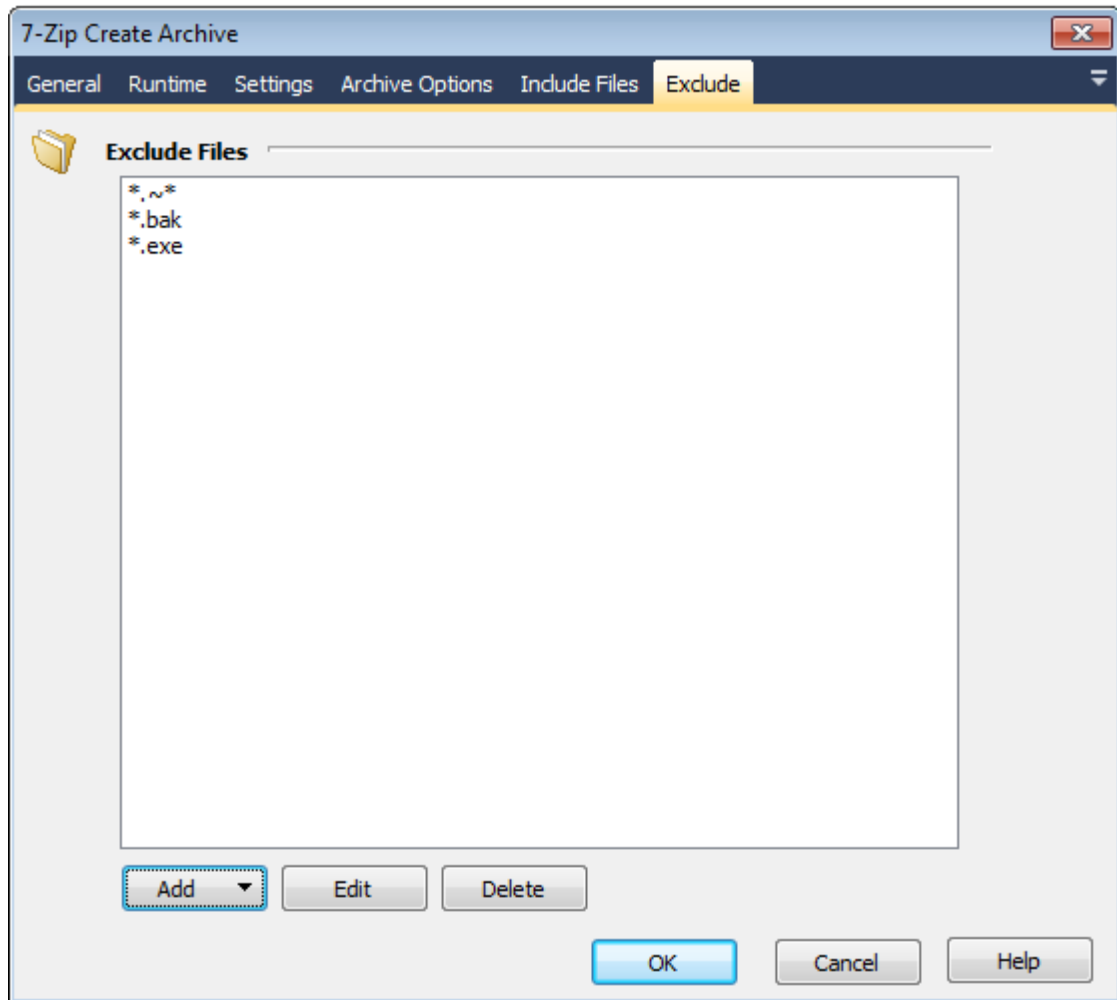
Solid Mode, SFX, and Multithreaded are modes available if you choose the 7zip format

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.



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



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

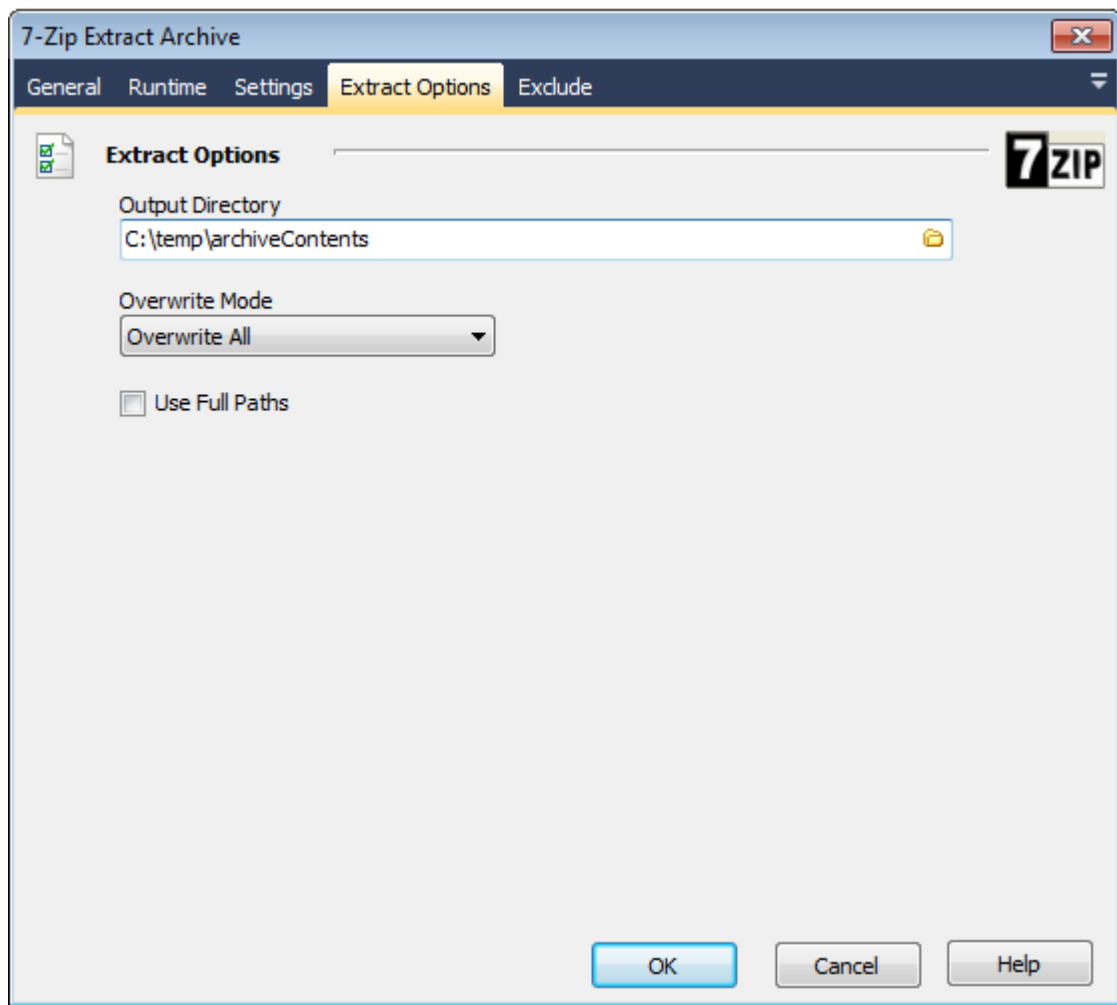
More Info on the 7Zip based actions

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



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

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

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:

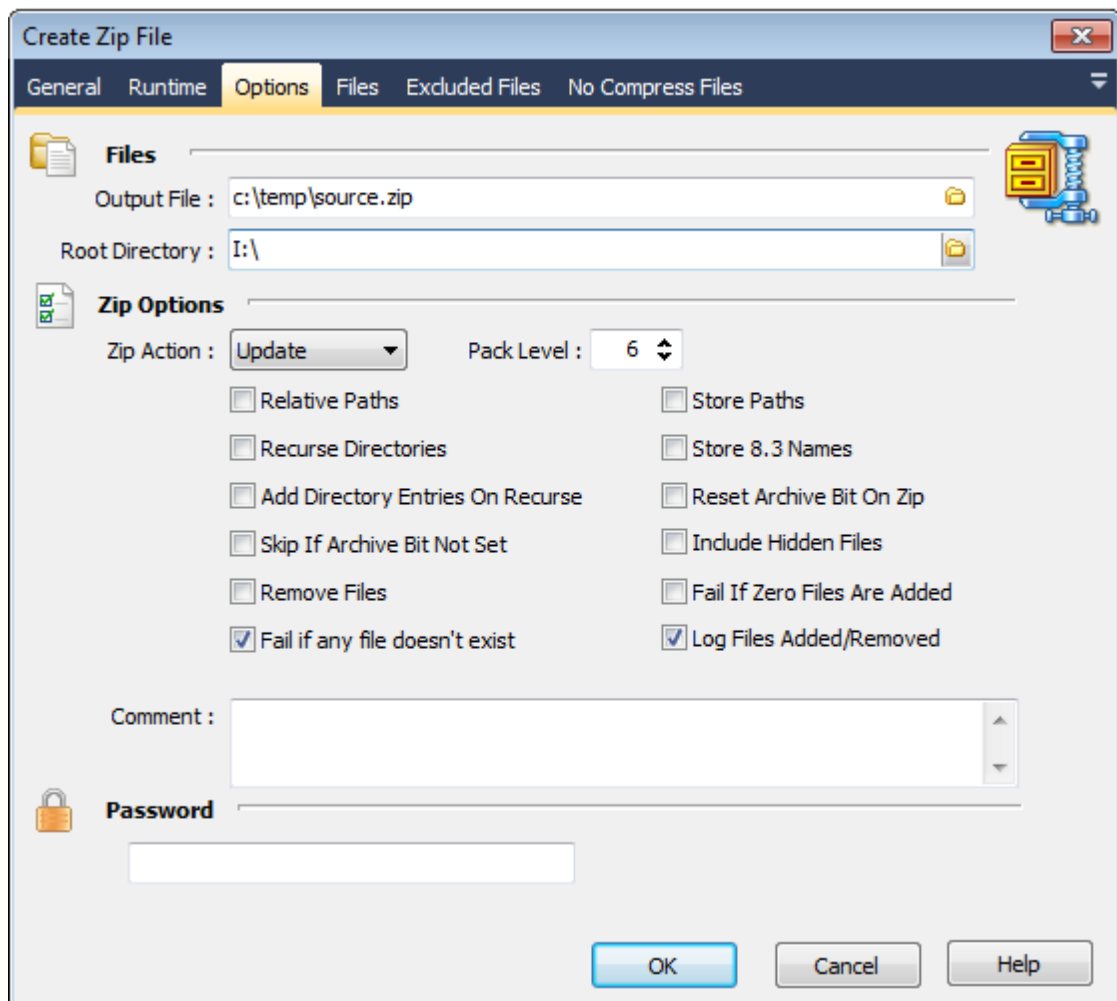
Zip, 7z, GZip, BZip2, TAR

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.



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.

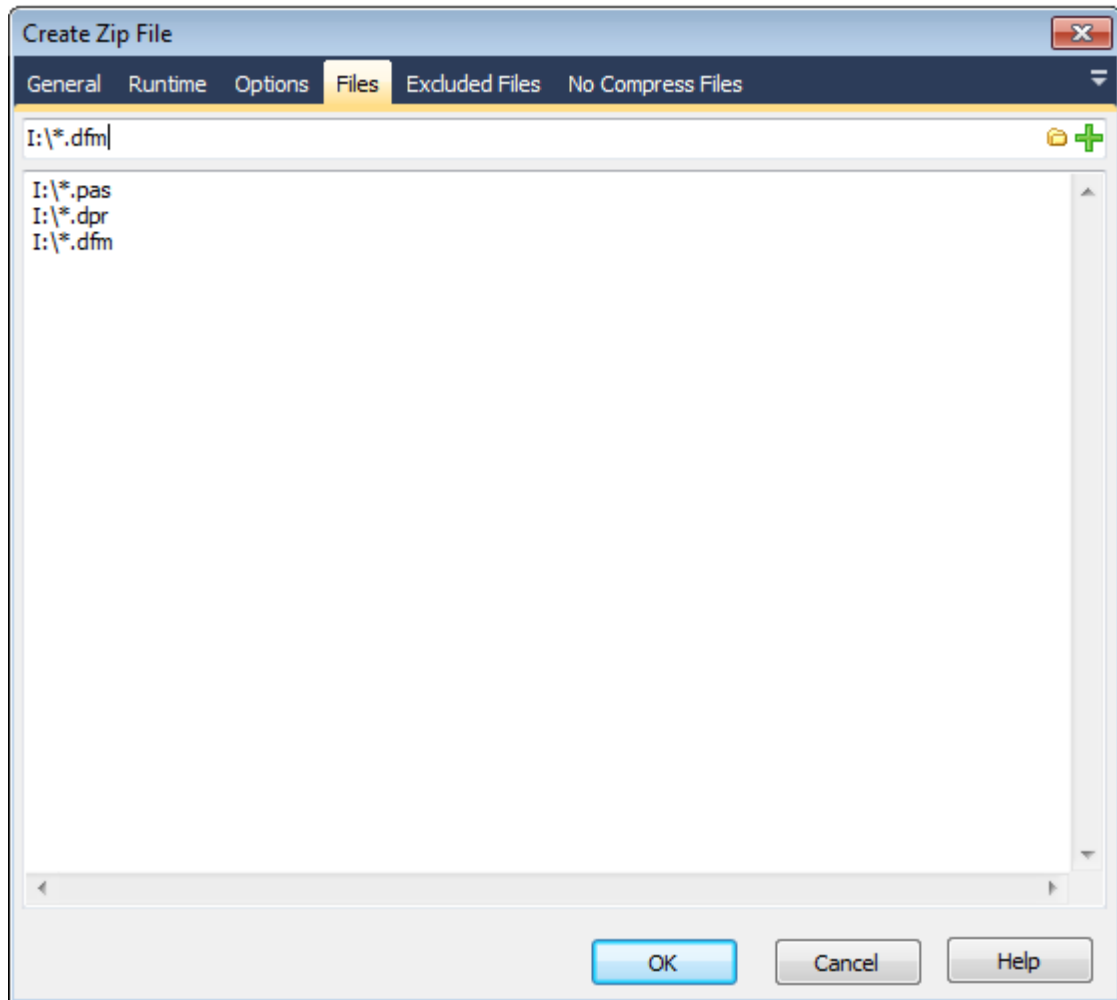
Root directory : The RootDir property determines where zipping will start for any wildcard entries or filenames in the FilesList Property that do not already include path information. Essentially, the value of RootDir will be prepended to anything in the FilesList that does not have any path information 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.



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

Scripting Info

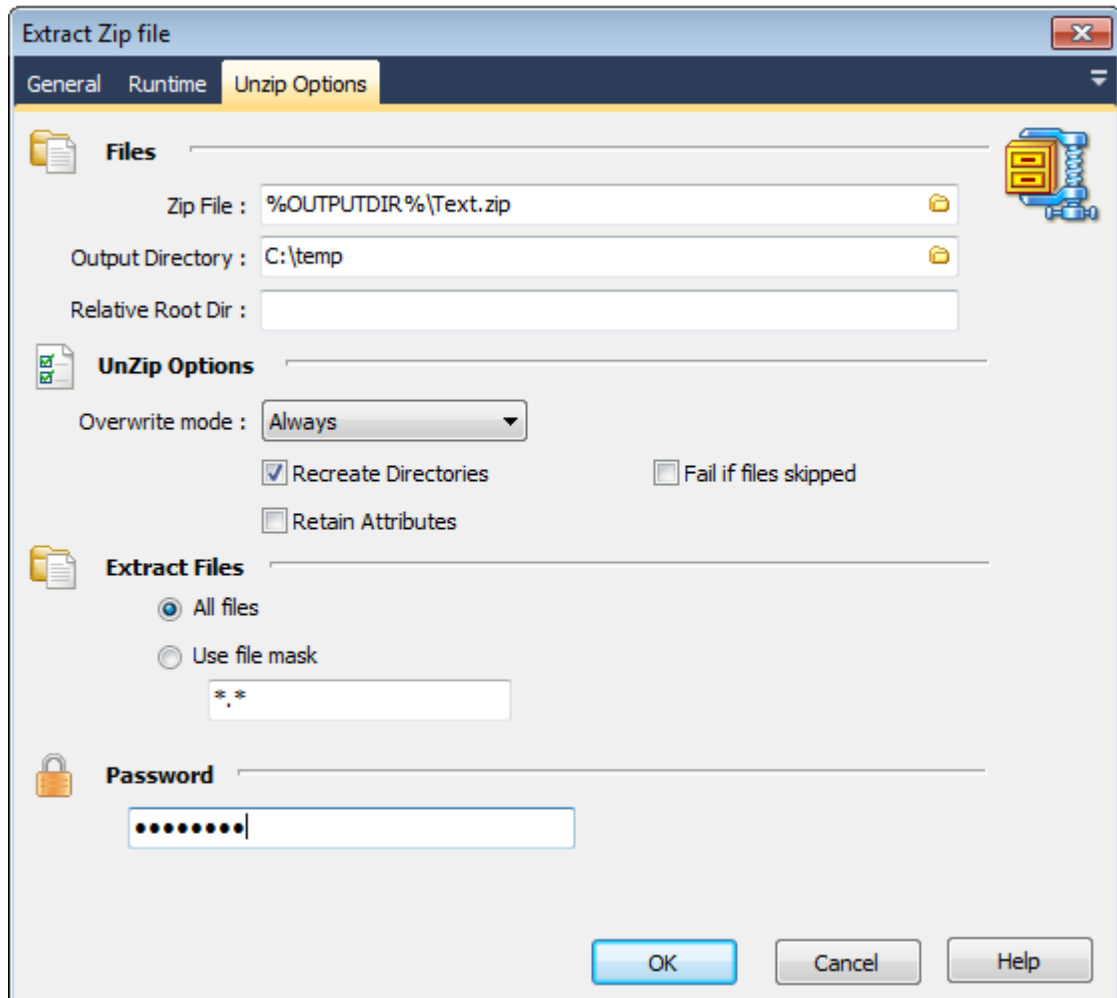
The Action properties available are :

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 - 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).

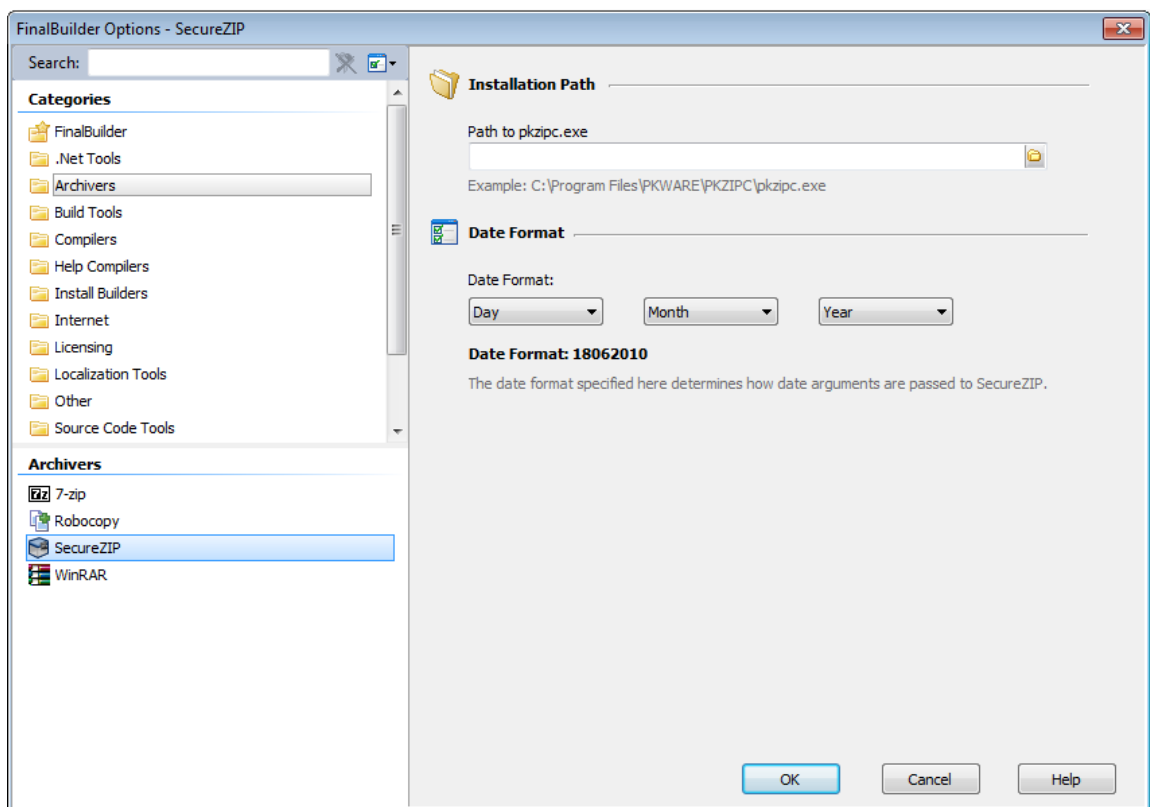
SecureZIP CLI is available from www.pkware.com

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.



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

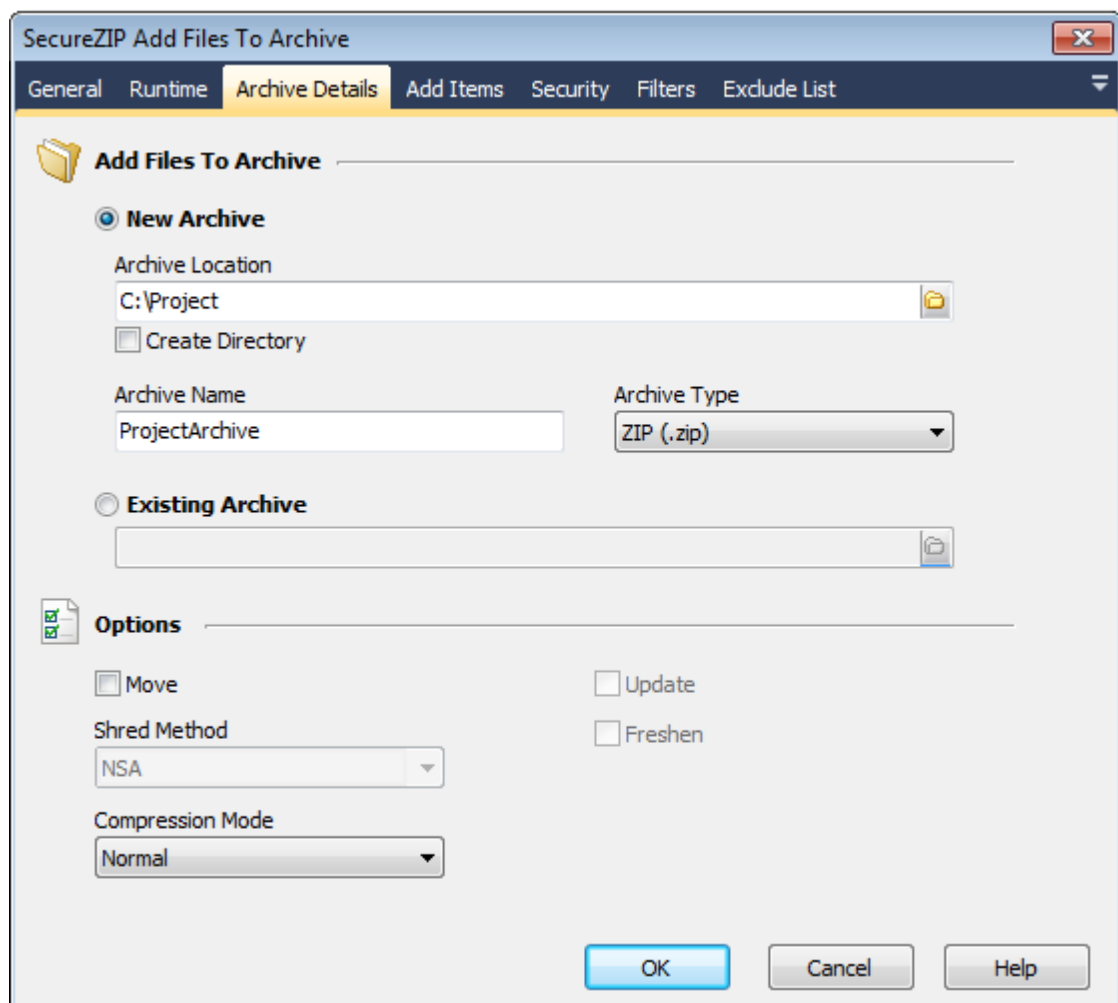
the date format so that the current date is displayed below the controls in the format specified.

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.



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.

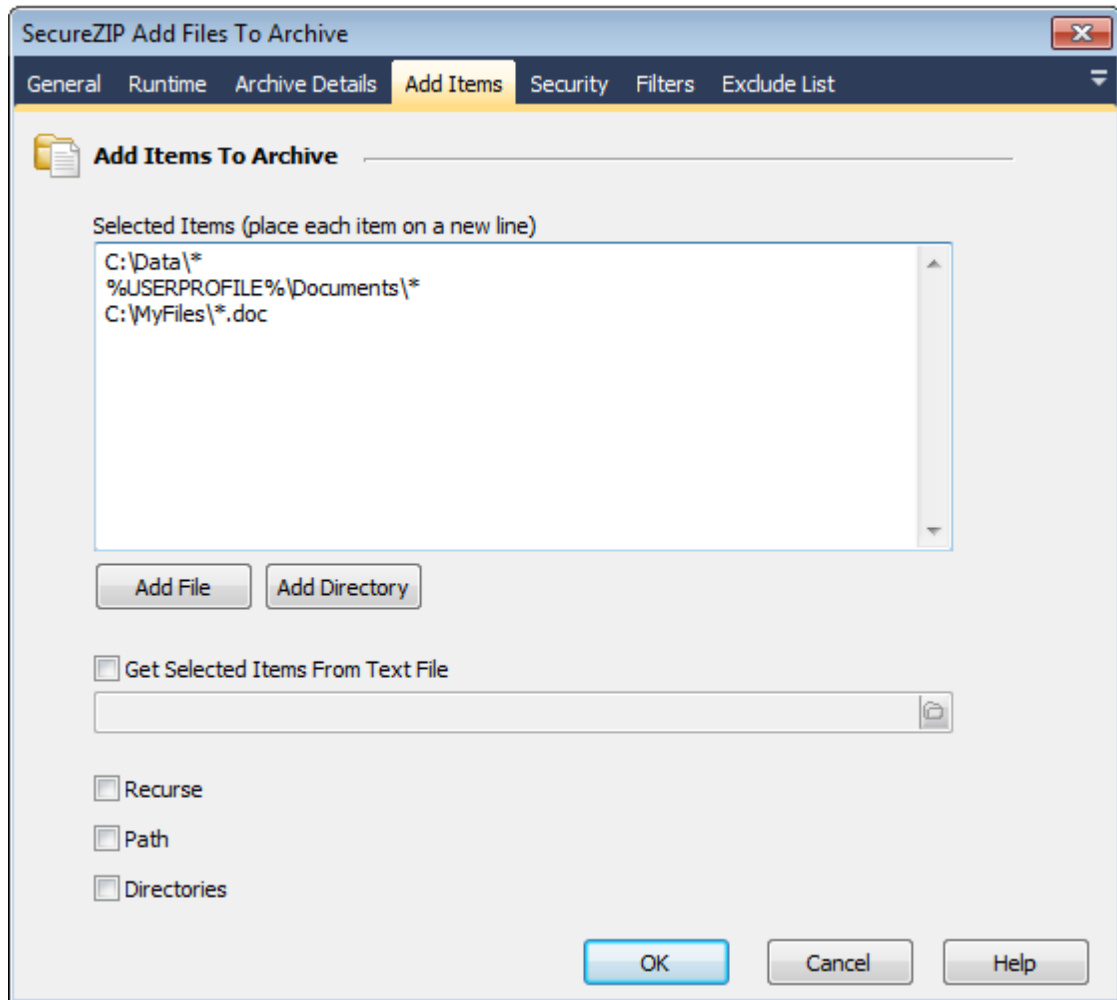
- **Archive Name** - The name of your archive.
- **Archive Type** - This determines the type of archive to create.

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.



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.

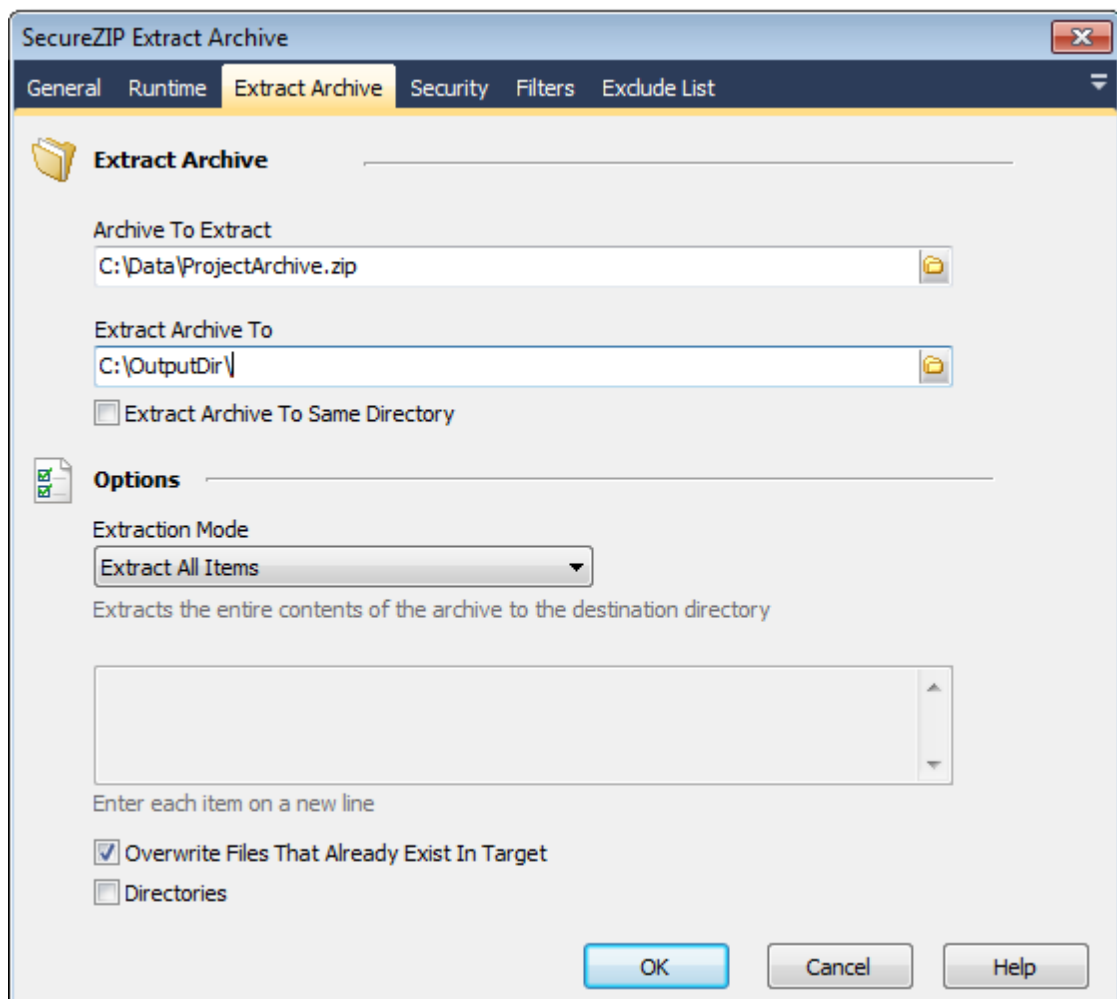
Additional Information

- 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 exclusion 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.



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.

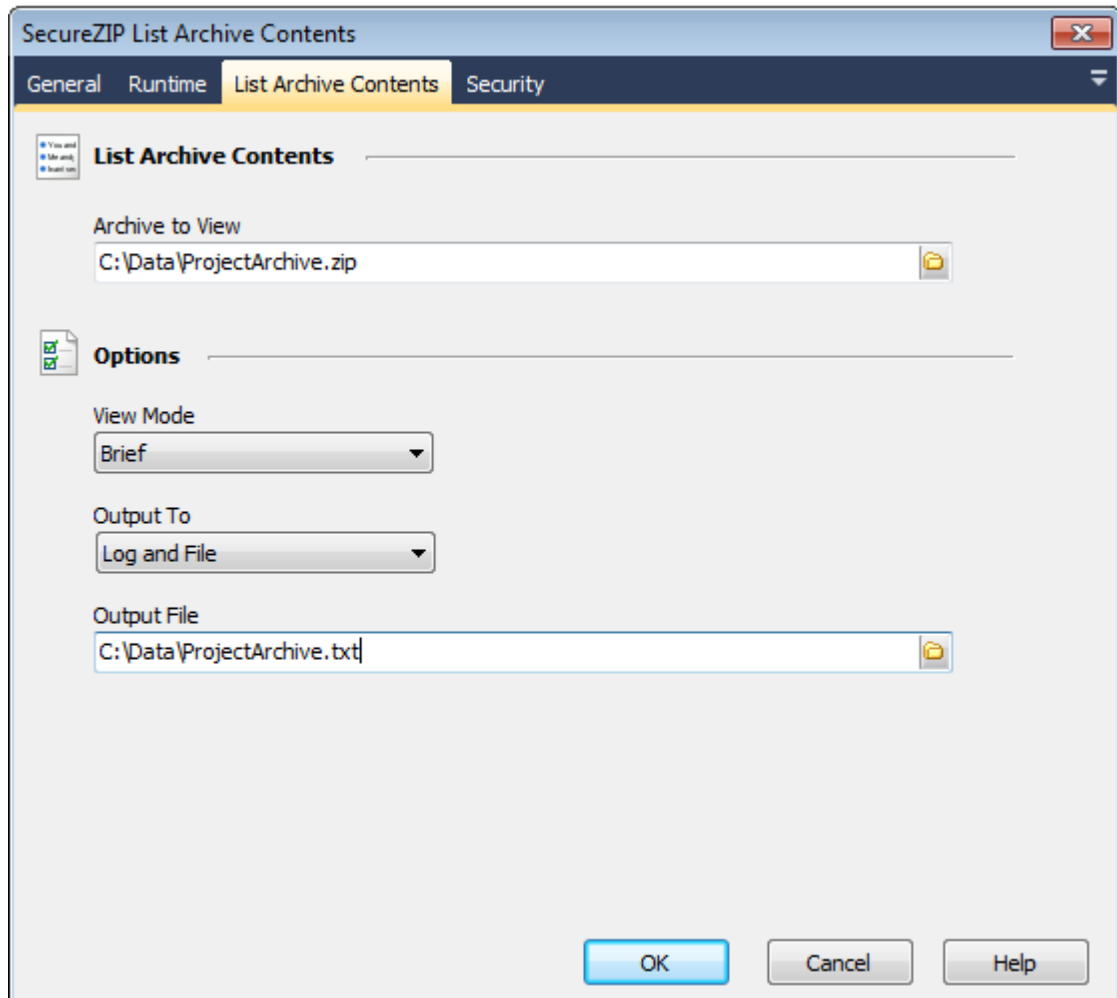
Additional Information

- 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 exclusion 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.



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.

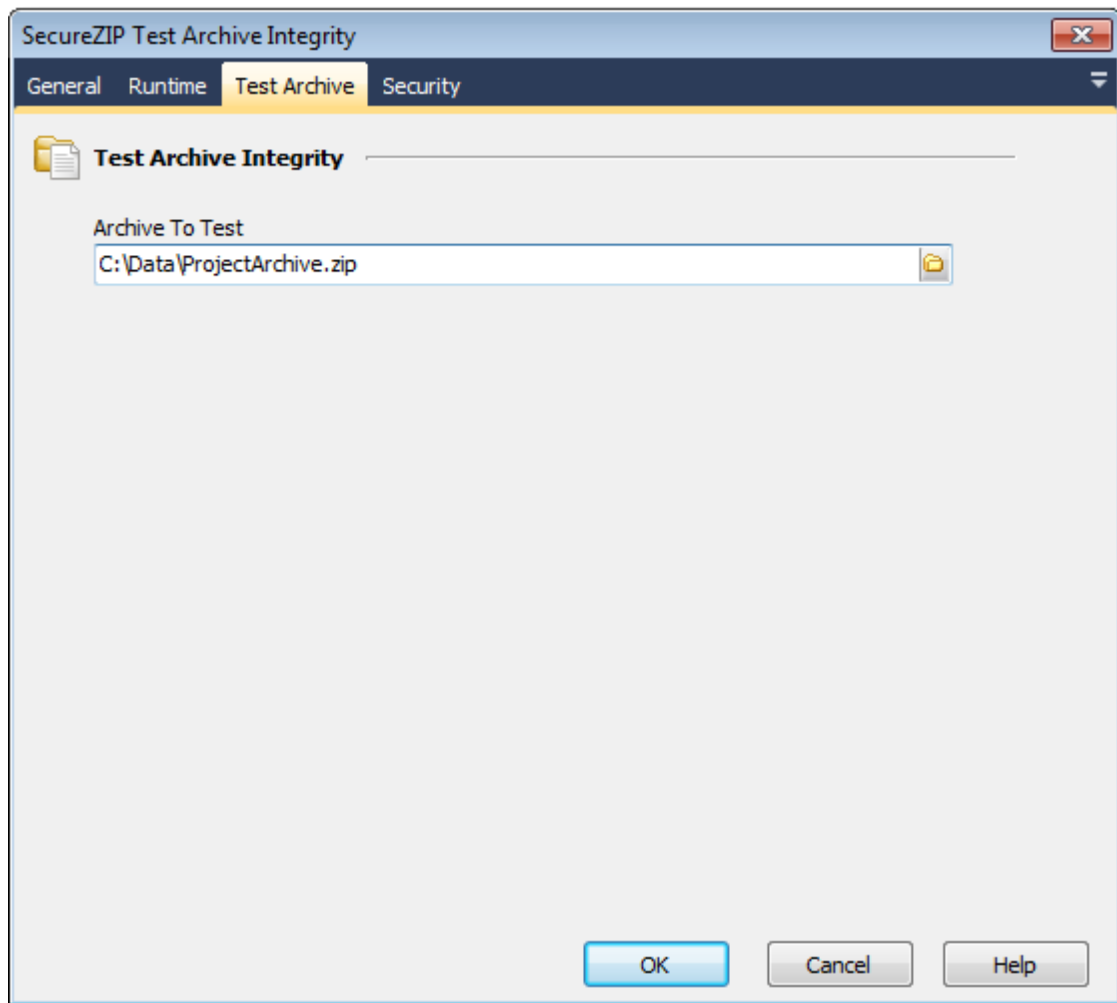
Additional Information

- 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.



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

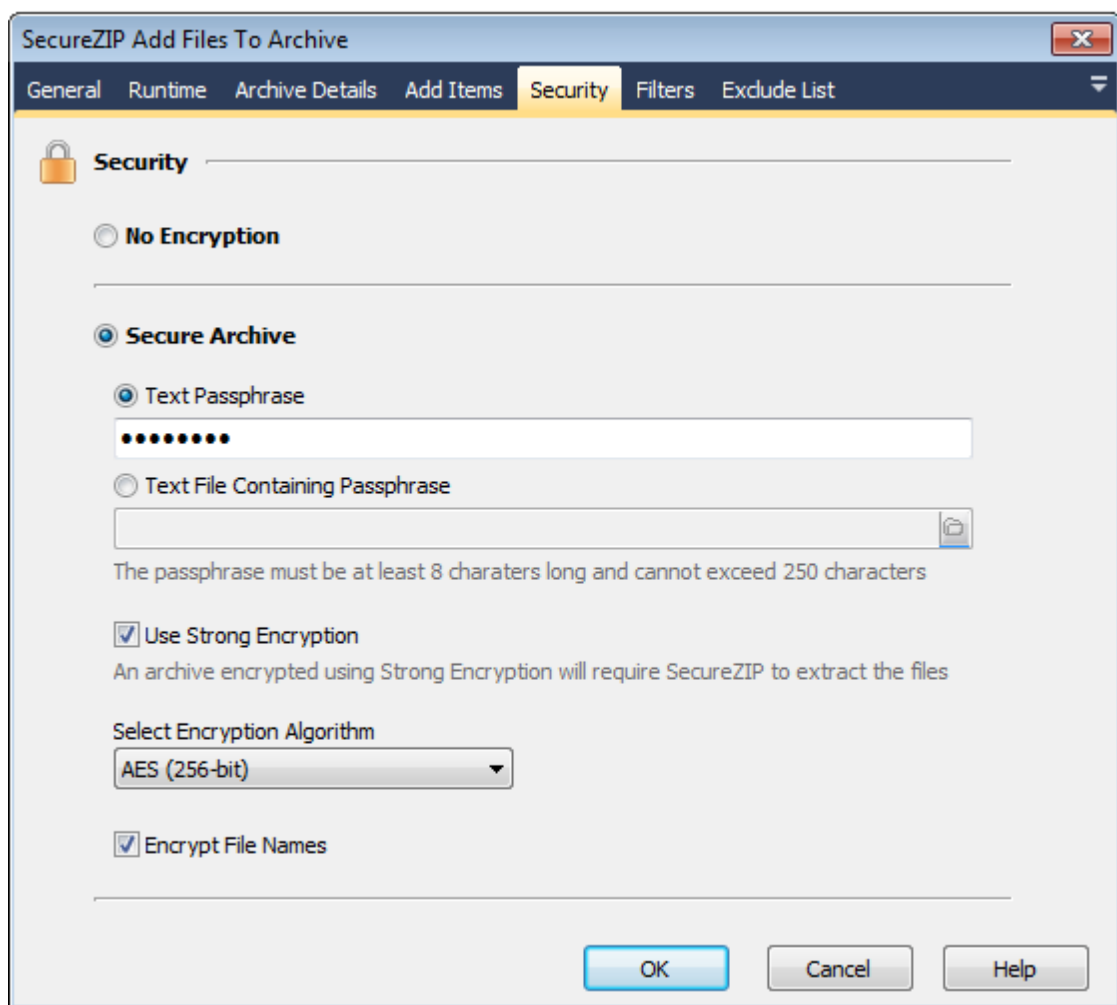
6.4.4.5.1 SecureZIP Security Page

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.



To create a secure archive select the **Secure Archive** option and use the following

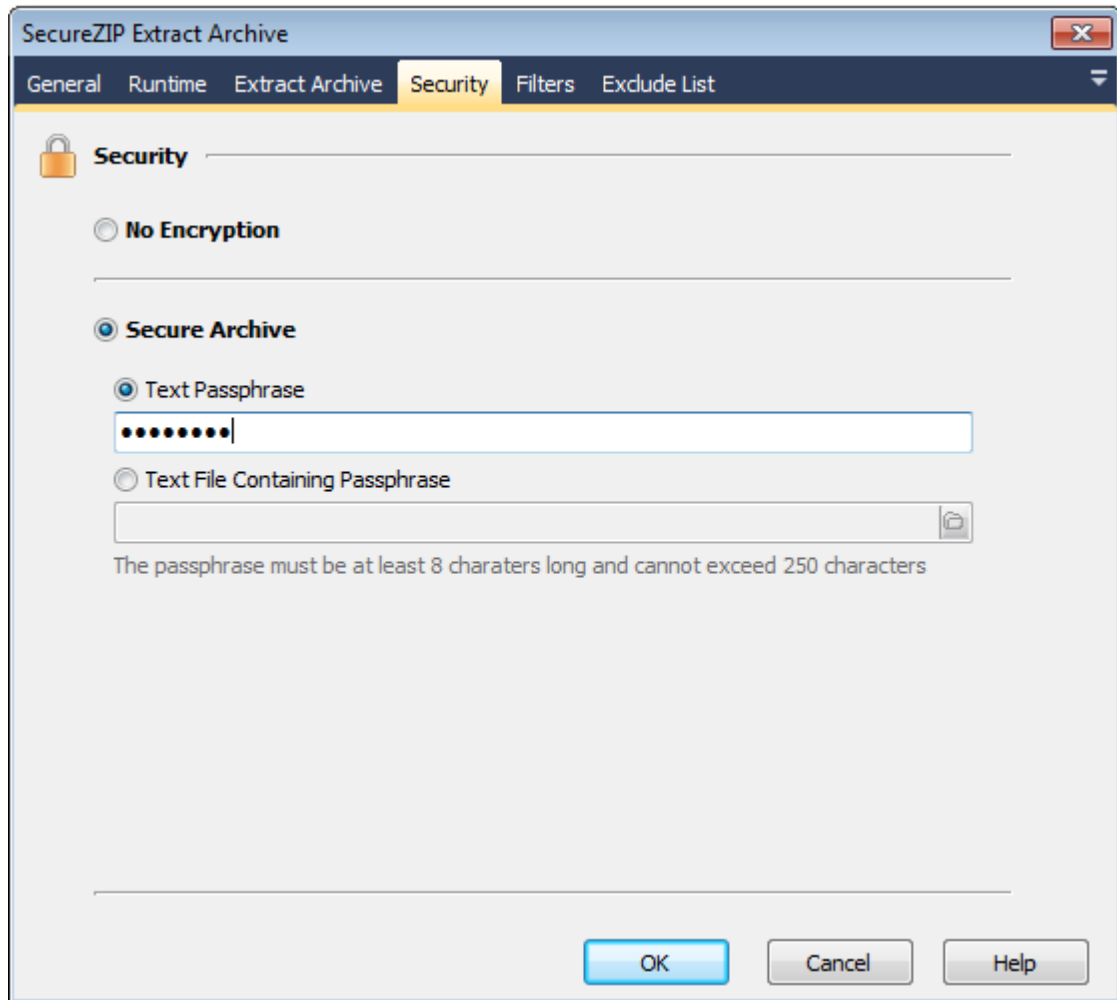
security options provided:

- **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-authorized 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.

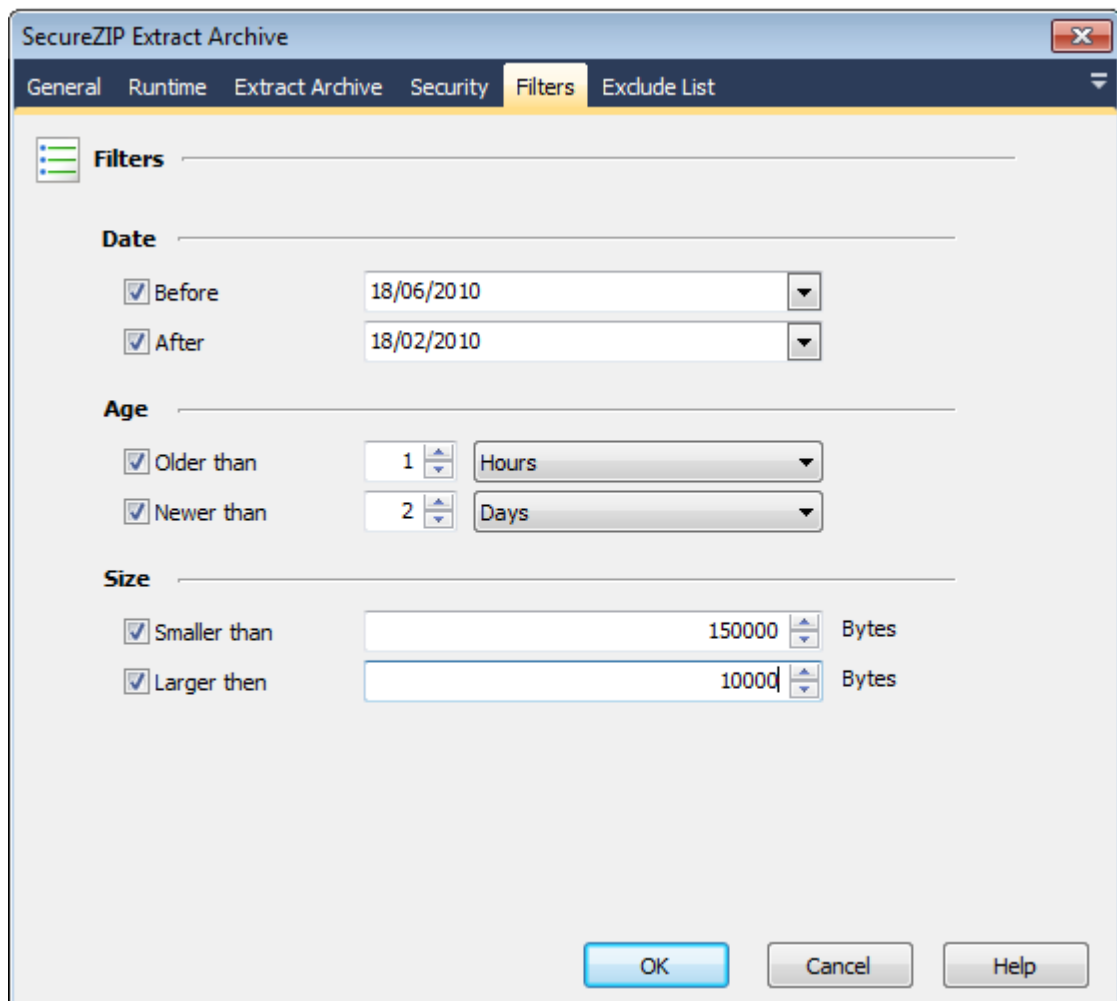


On the *Security* Page specify the passphrase that was used to secure the archive. This can be done either of the two methods:

- **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.



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.

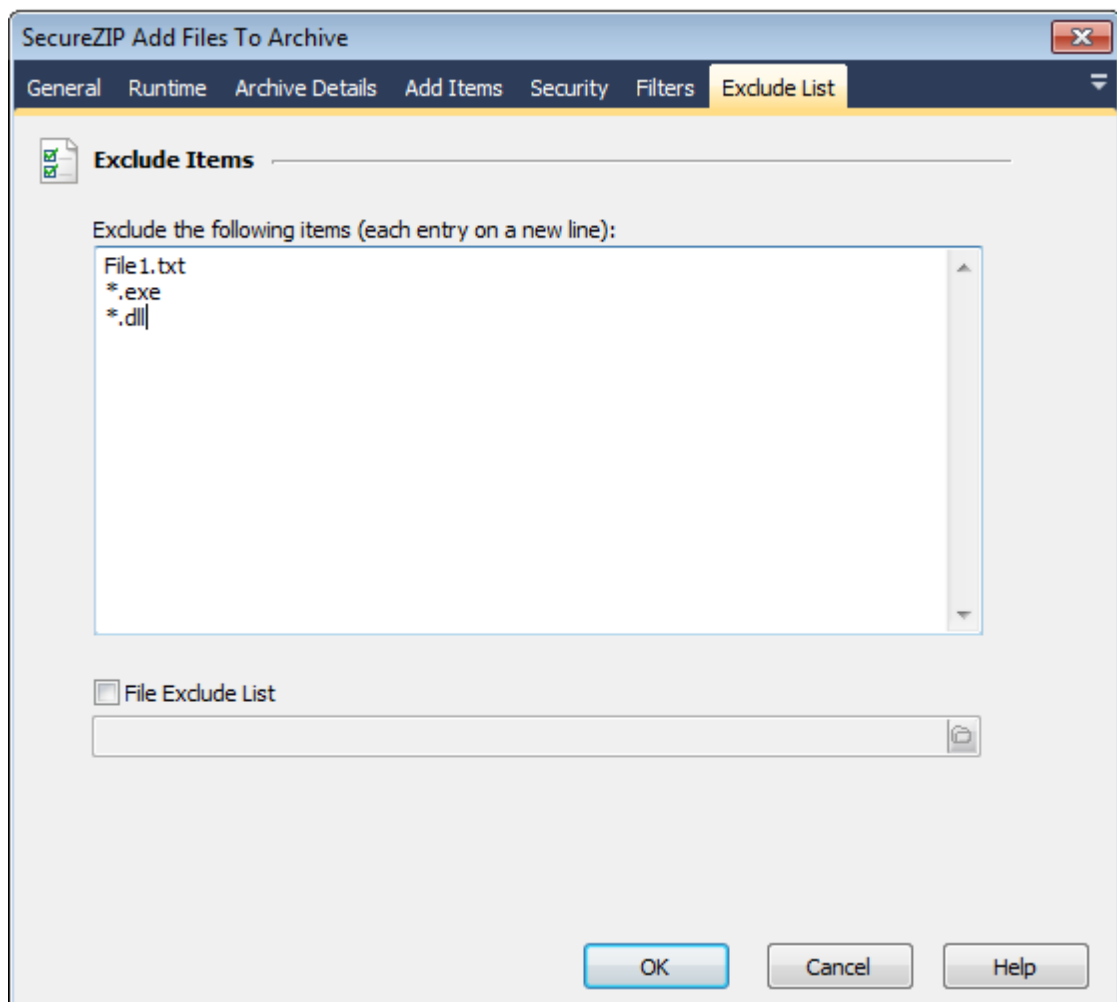
- **Smaller than** - Only select items smaller than the specified number of bytes.
- **Larger than** - Only select items larger than the specified number of bytes.

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.



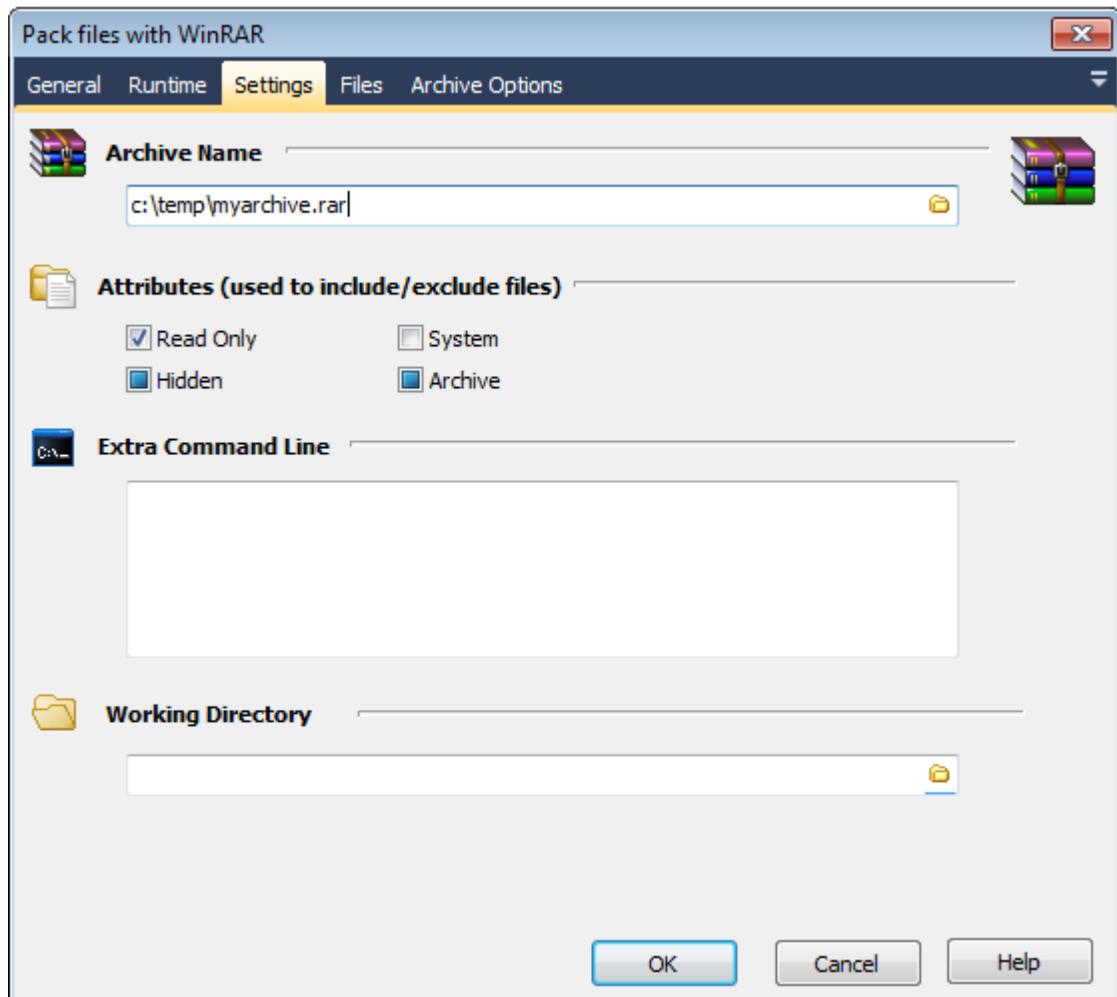
On the *Exclude List* page enter the names of the files that you wish to exclude from the process and when the action is run these files will be ignored.

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 - specify the name of the archive to create

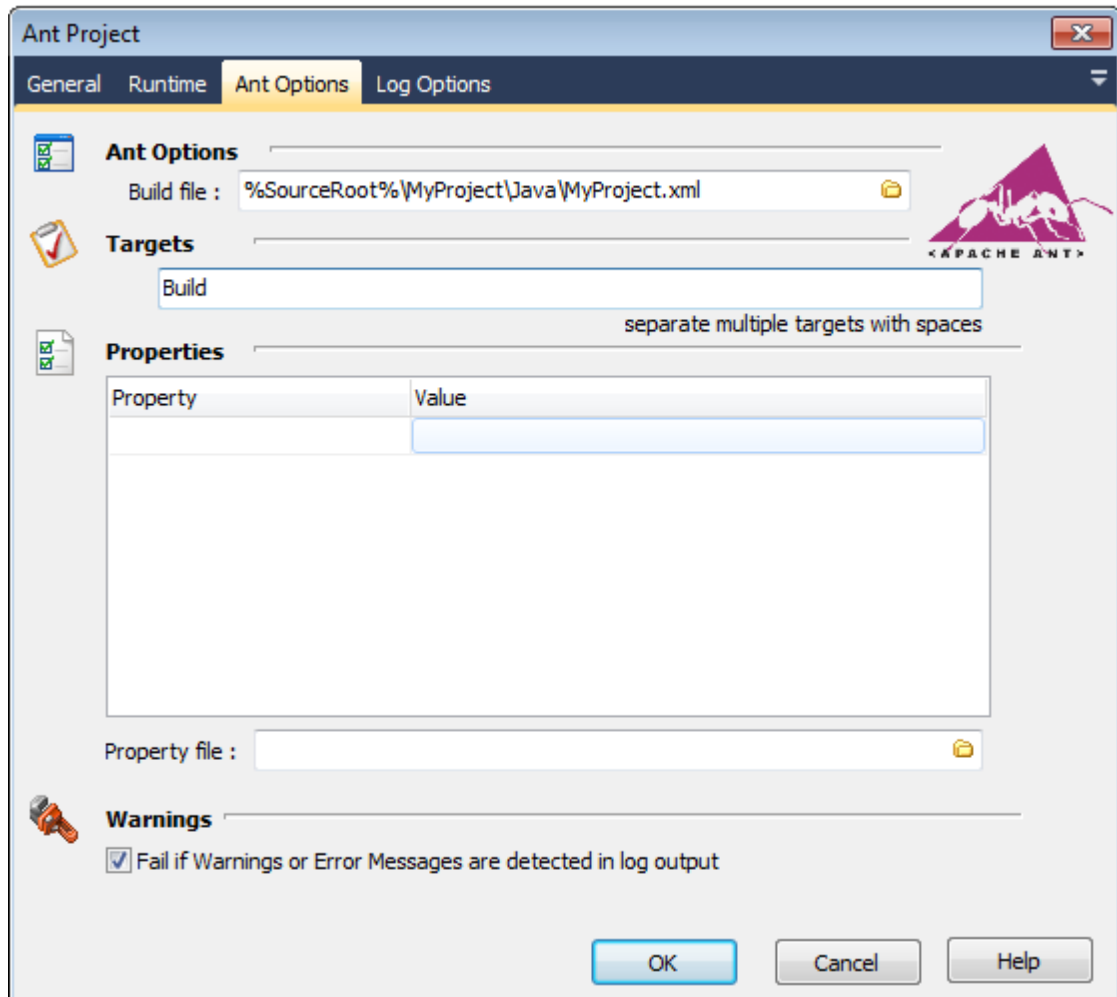
Attributes - In the above screenshot, only files with the Read Only attribute, and not system files will be added to the archive.

Extra Command Line - If there are any WinRAR options which the FB action doesn't surface, then you can manually specify them

6.5 Build Tools

6.5.1 Ant Project Action

This action executes the Apache Ant command line build tool, making it easy to integrate legacy build scripts into your FinalBuilder build.



Build File

The path to an Ant project file to build.

Targets

One or more targets to build. Separate multiple targets with spaces.

Properties

Properties to pass into Ant on the command line. Press the "Insert" key to add new Properties.

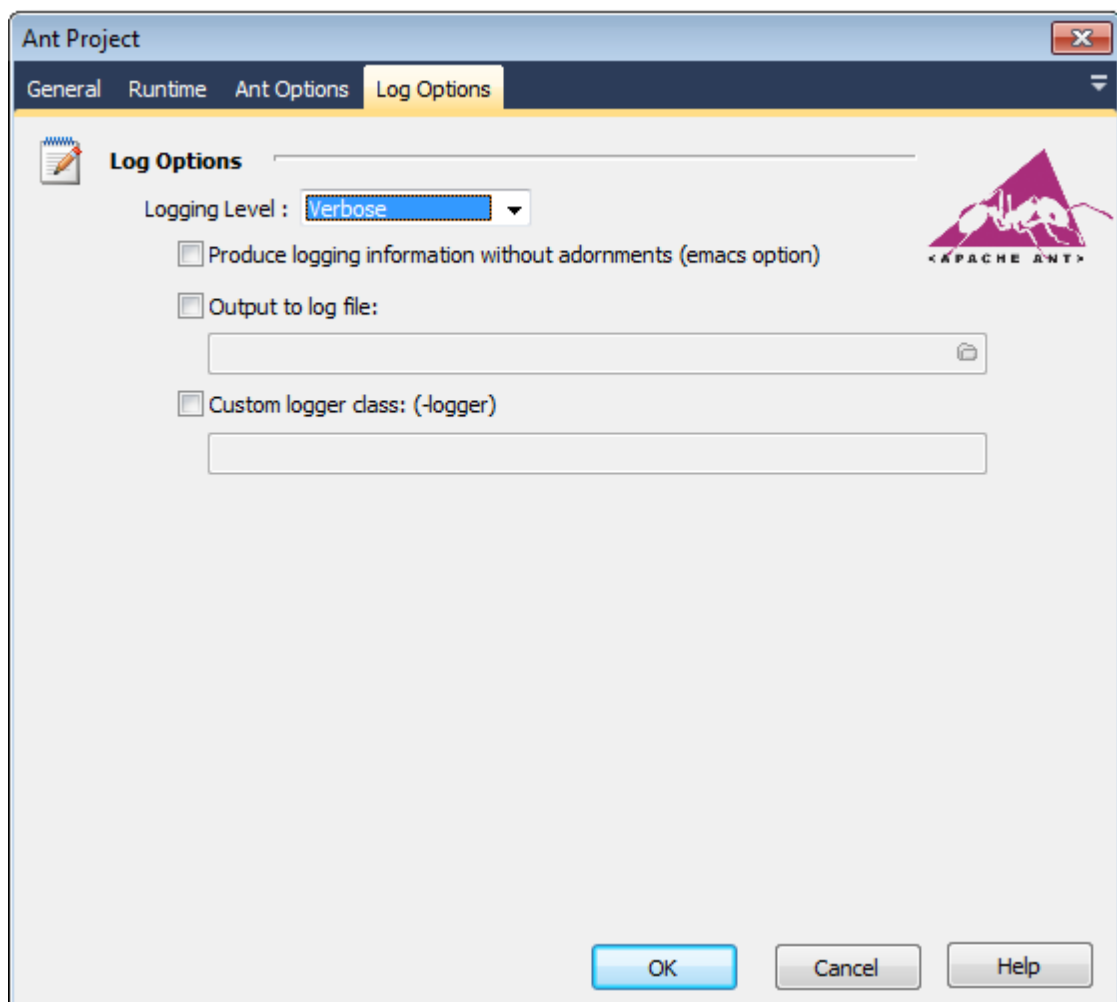
Property File

To pass properties from a text file, specify one here

Fail if Warnings or Error Messages are detected in log output

If this option is enabled, the Ant Project Action will fail if any error or warning messages are detected in the log output. This means the action may fail even though the overall result from Ant is "Success".

In some older versions of Ant, this option may be required in order to properly detect failed projects.



Logging Level

Choose a logging level for the output from the action.

Produce logging information without adornments

This is equivalent to passing the -emacs option to Ant.

Output to log file

Check this box to specify a path for Ant to write a dedicated log file.

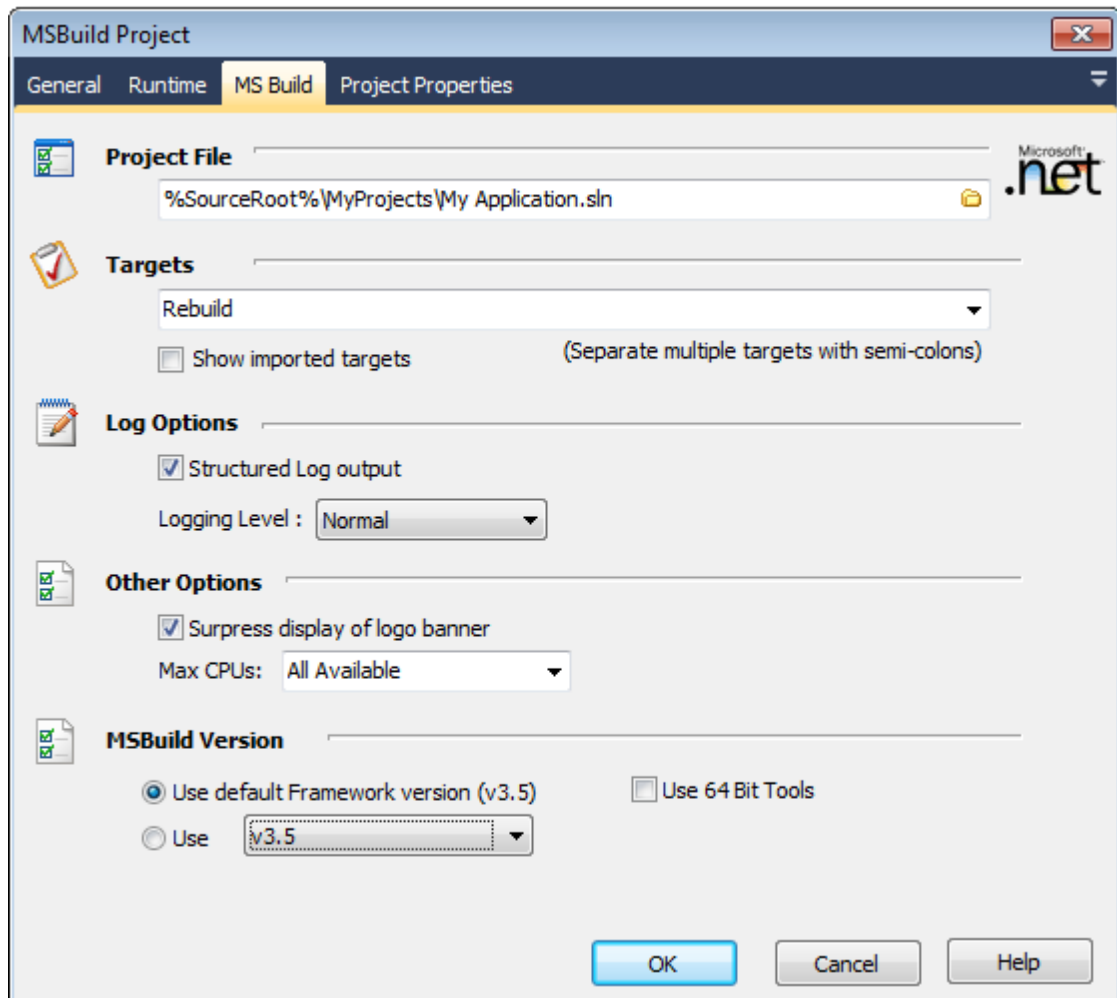
Custom logger class

Check this box to specify the name of a custom logger class.

6.5.2 MSBuild Project Action

This action allows you to build MSBuild projects as part of your FinalBuilder build. MSBuild is Microsoft's build technology for .NET, and is used by Visual Studio 2005 and 2008.

The MSBuild Project action can build project files and solutions created by these versions of Visual Studio (ie csproj, vbproj, sln.) Alternatively, you can use the Visual Studio action to build these projects. Note that if you use the MSBuild Project action, you do not need Visual Studio to be installed on the build machine.



Project File

The MSBuild project file to build. Files can be .proj, .csproj, .vbproj, etc. or solution files with the extension .sln.

Targets

Select one or more targets to build. Separate multiple target names with semi-colons.

If this field is left blank, the action will build the default targets.

The dropdown list is automatically populated with any default target names for the specified project, as well as any targets defined in the project file itself.

Show imported targets

Check this box to have the dropdown list also show targets imported into this project from other files (note that this list can be very long.)

Structured Log Output

Check this box to have FinalBuilder log the output from MSBuild in a structured and hierarchical manner. This can be very useful when viewing FinalBuilder logs.

If the box is unchecked, MSBuild will use the standard console logger - so the output will be the same as if MSBuild had been run from the command line.

Logging Level

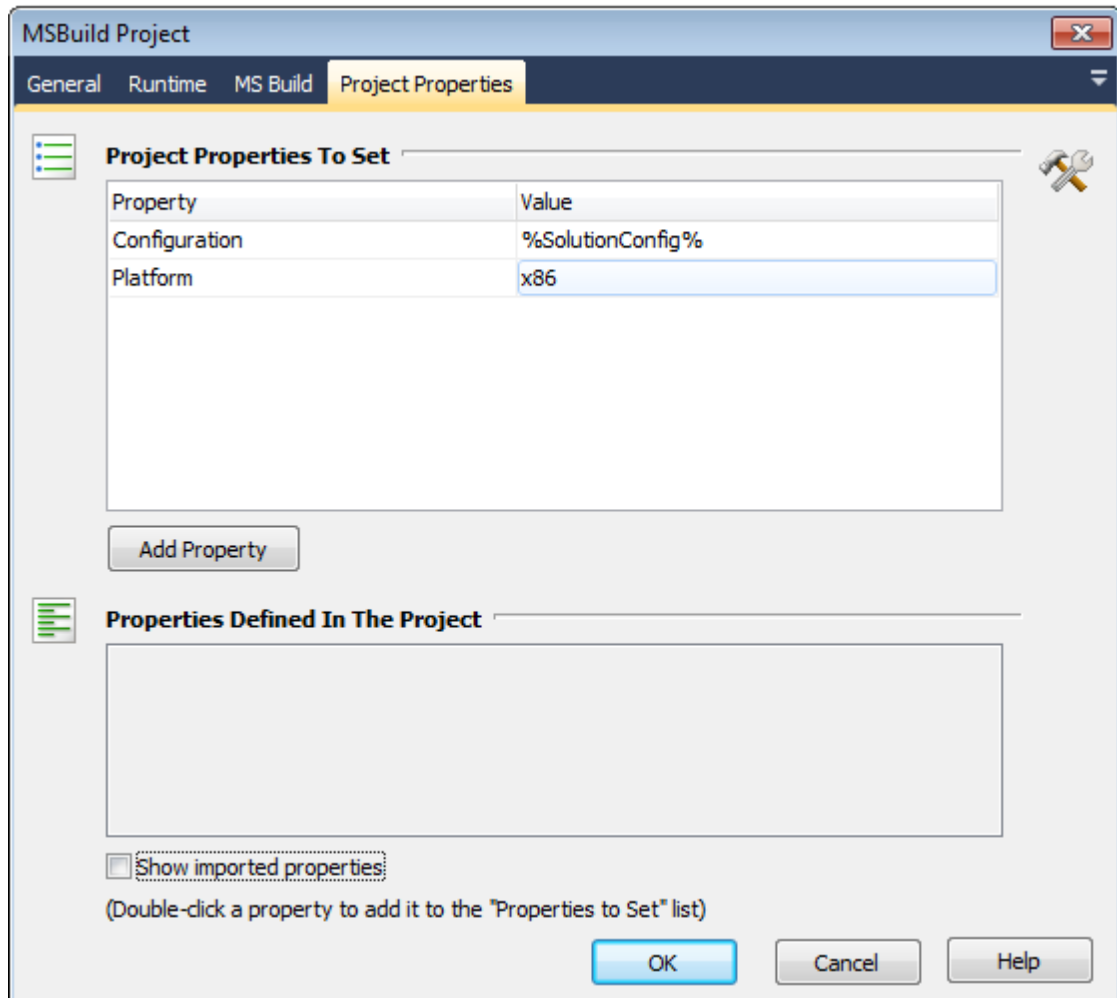
MSBuild supports different logging levels. Choose "Detailed" or "Diagnostic" to see lots of extra output in the logs (useful for debugging.) Choose "Minimal" or "None" to minimize the size of the logged information.

Suppress display of logo banner

By default, MSBuild logs the current MSBuild and .NET Framework versions. Check this box to exclude this information from the log.

Framework Version

The MSBuild action can use the default FinalBuilder Framework version (set in the .NET Options), or you can choose a specific framework version to use.



On this property page, you can set any project property values to be passed to the MSBuild project.

The "Properties Defined In the Project" list automatically shows all of the properties found in the project file. Double-click a property to add it to the "Project Properties To Set" list.

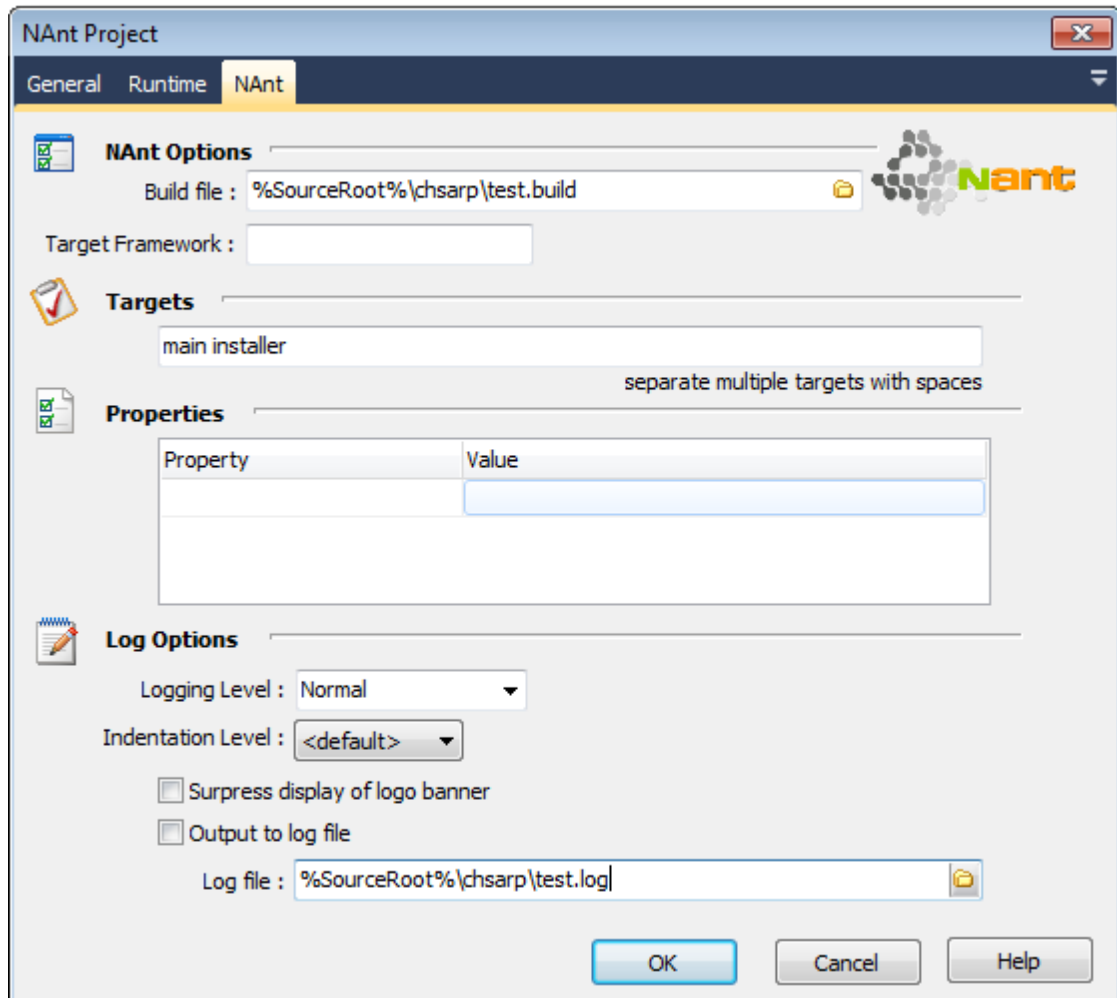
To add other properties to the list, focus the last item in the Project Properties list box (as shown) and press the "down" arrow key.

Show imported properties

By default, only properties which are defined in the actual project file are shown in the list. Checking this box will also show properties which are imported from other files. Note that this list may be very long.

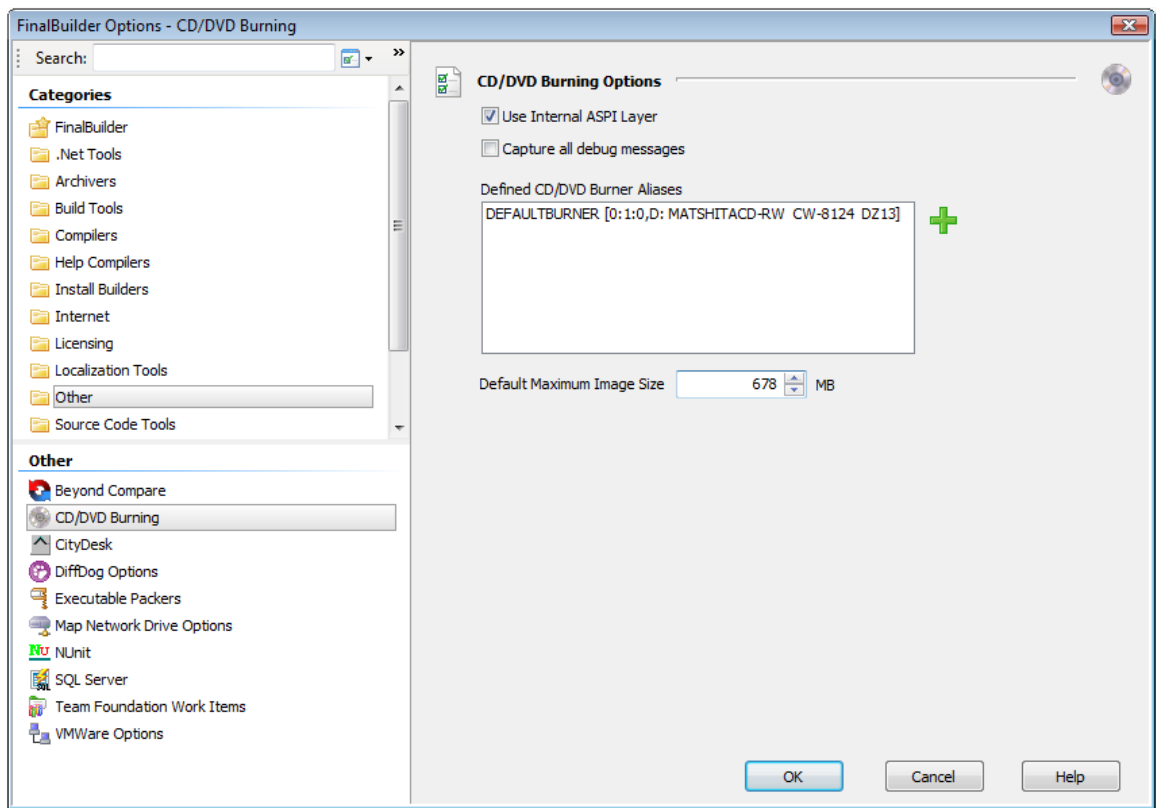
6.5.3 NAnt Project Action

This action executes the NAnt command line build tool, making it easy to integrate legacy build scripts into your FinalBuilder build.



6.6 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.



FinalBuilder 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

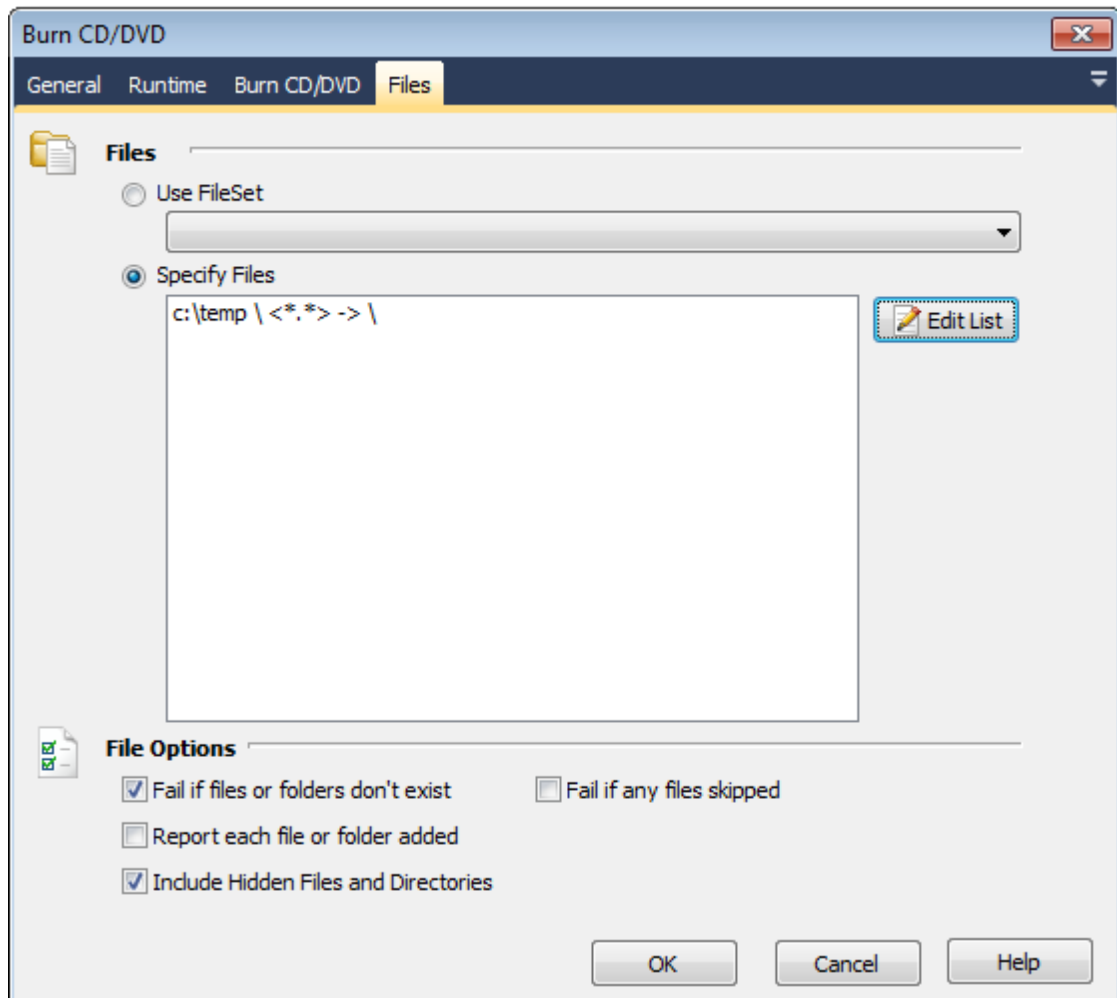
Windows Vista RTM, SP1

Windows Server 2008

Windows Feature Pack for Storage 1.0 Download - Adds bluray and burn verification support, recommended.

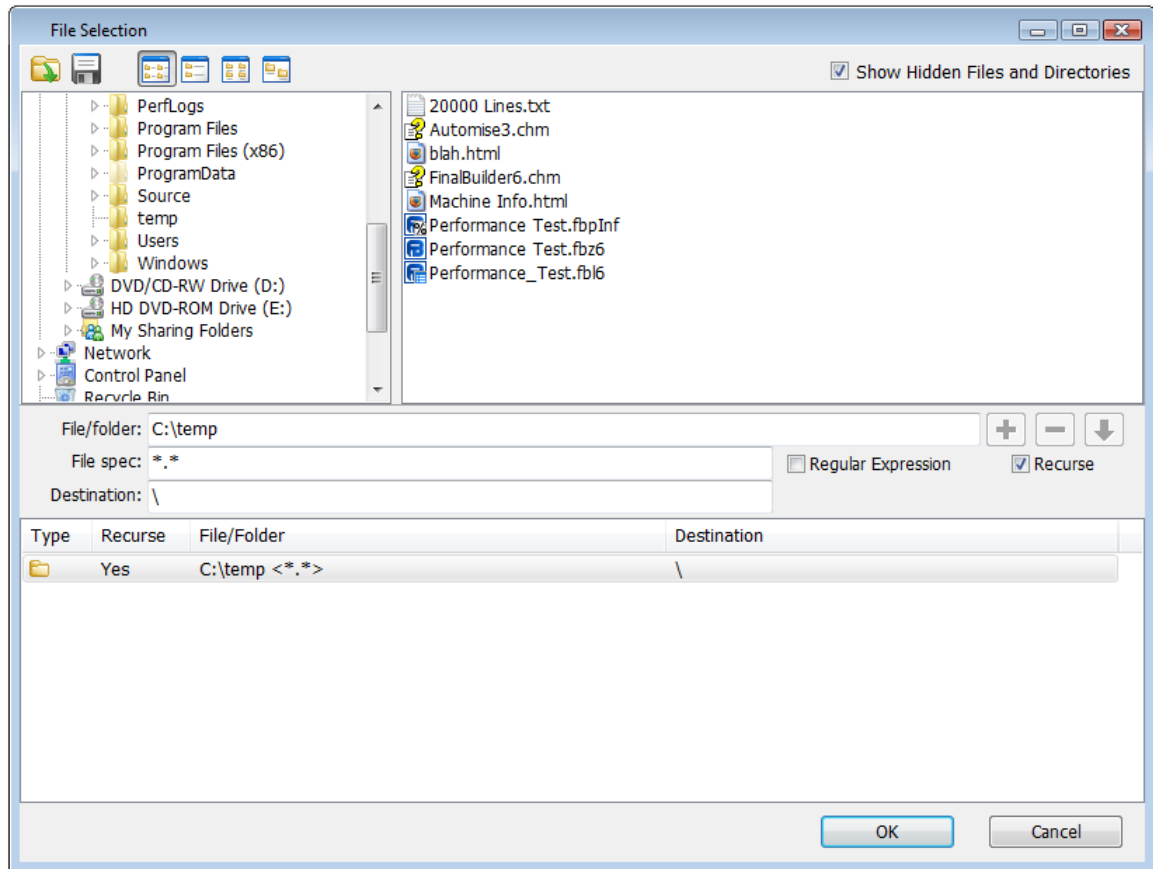
6.6.1 Burn CD/DVD Action

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).



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.



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.

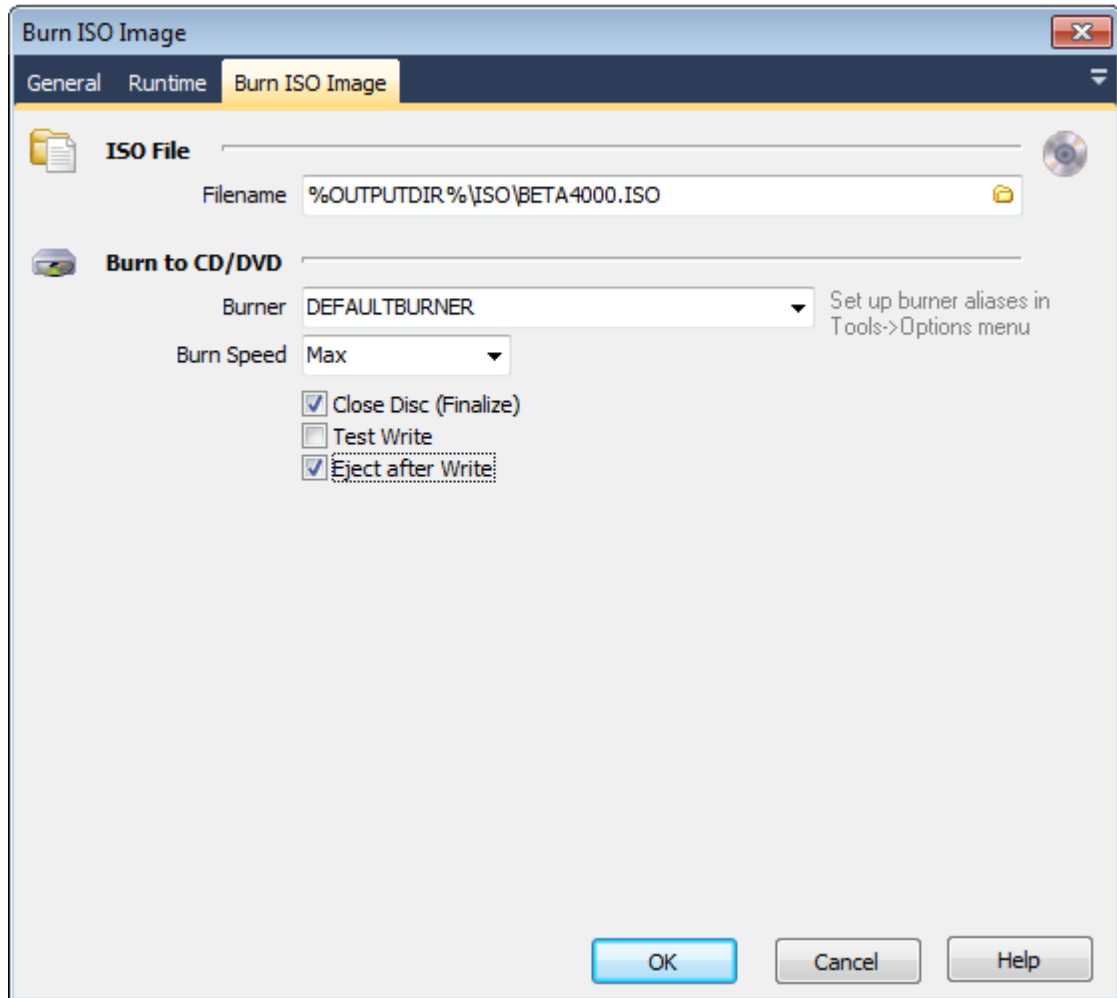
"Eject after write"

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.

6.6.2 Burn ISO Action

[FinalBuilder Professional Edition]

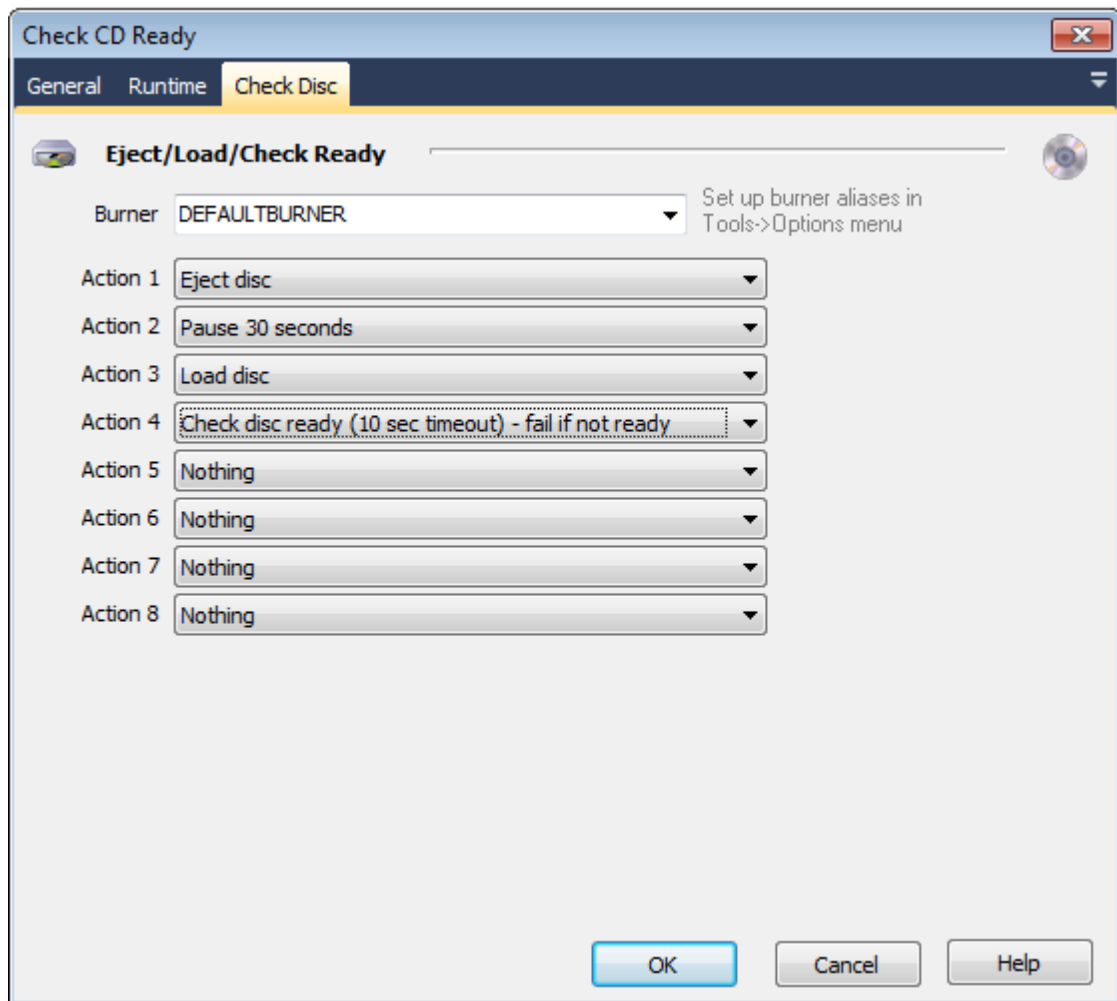
This action supports burning an ISO file generated by FinalBuilder(or any other tool that generates valid ISO Files) to CDR/CDRW/DVDR & DVDRW Media (depending on your burner hardware). Note that to use this action you need to define an alias for your burner first (see here for more info).



6.6.3 Check Ready Action

[FinalBuilder Professional Edition]

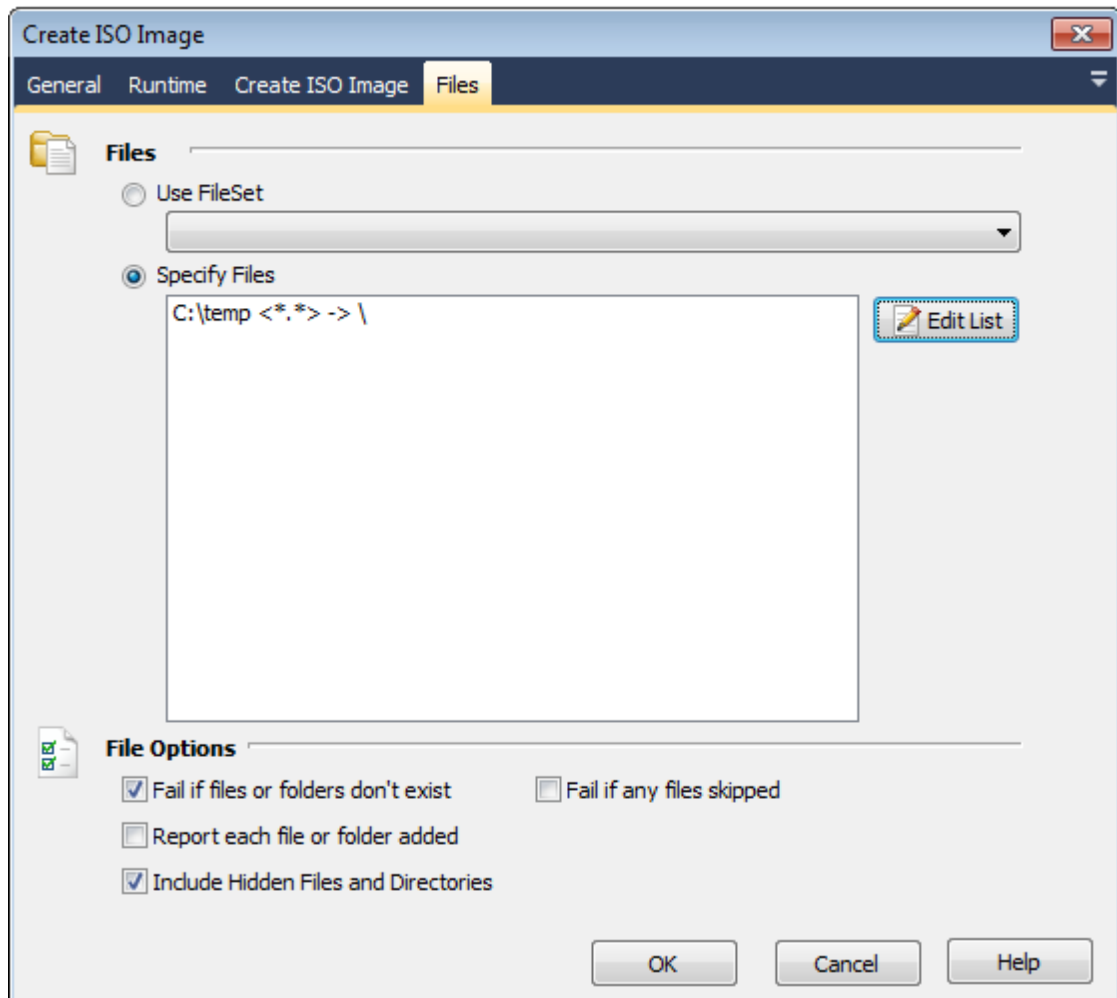
This action allows you to automate to some degree the process of loading a CD into the burner, and making sure the burner is ready to burn before attempting to burn to CD. An example which shows the use of this action is installed in the FinalBuilder\Examples\CDBurner directory. Note that to use this action you need to define an alias for your burner first (see here for more info).



6.6.4 Create ISO Action

[FinalBuilder Professional Edition]

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 FinalBuilder does not impose any limits on the size of the resulting ISO File.



"Joliet File System"

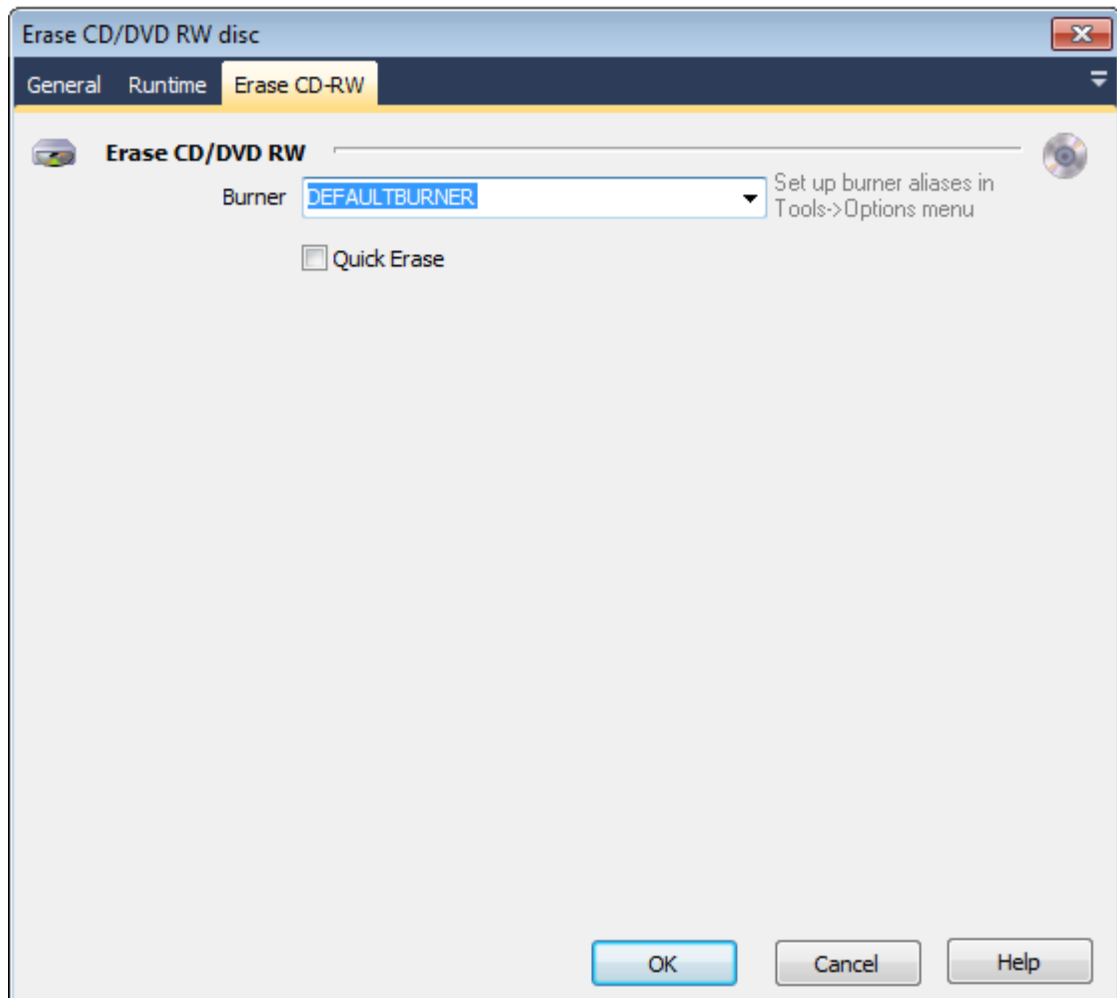
Enables the Joliet extensions to the ISO9660 file system. This allows you to have file names longer than 8.3 characters. In almost all instances, you'll want to leave this option enabled.

6.6.5 Erase CD/DVD RW

[FinalBuilder Professional Edition]

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.

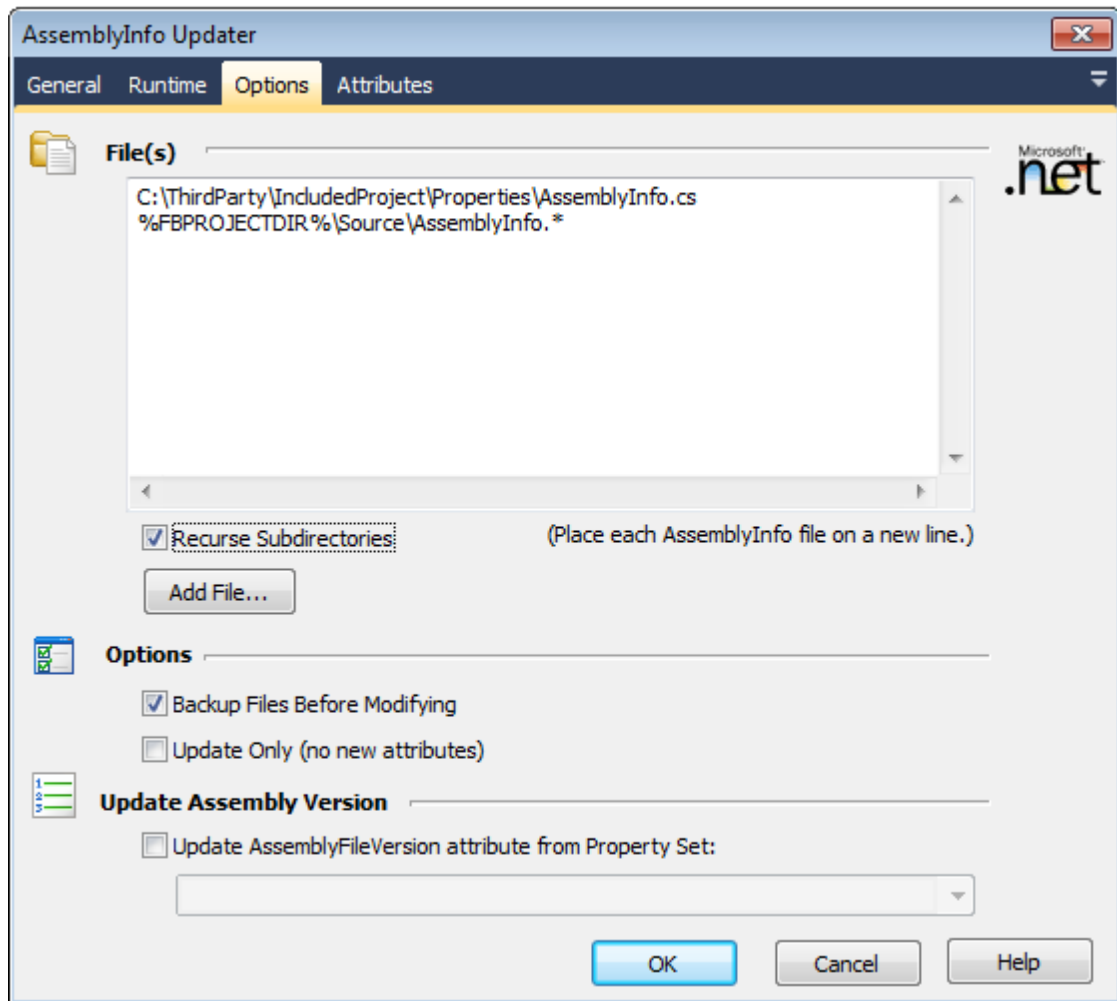
Quick Erase is recommended for most media and is substantially faster (full erase can take a long time on a DVDRW!).



6.7 Compilers

6.7.1 AssemblyInfo Updater Action

This action allows you to update one or more AssemblyInfo source files. Supported languages are C#, VB.NET, J#, C++ and Delphi.NET.



File(s)

Enter one or more AssemblyInfo files to update. Wildcards * and ? are allowed (as shown above.)

Recurse Subdirectories will recurse subdirectories for any entry which contains a Wildcard character. In the example above, all AssemblyInfo files within subdirectories of "%FBPROJECTDIR%\Source" will be updated.

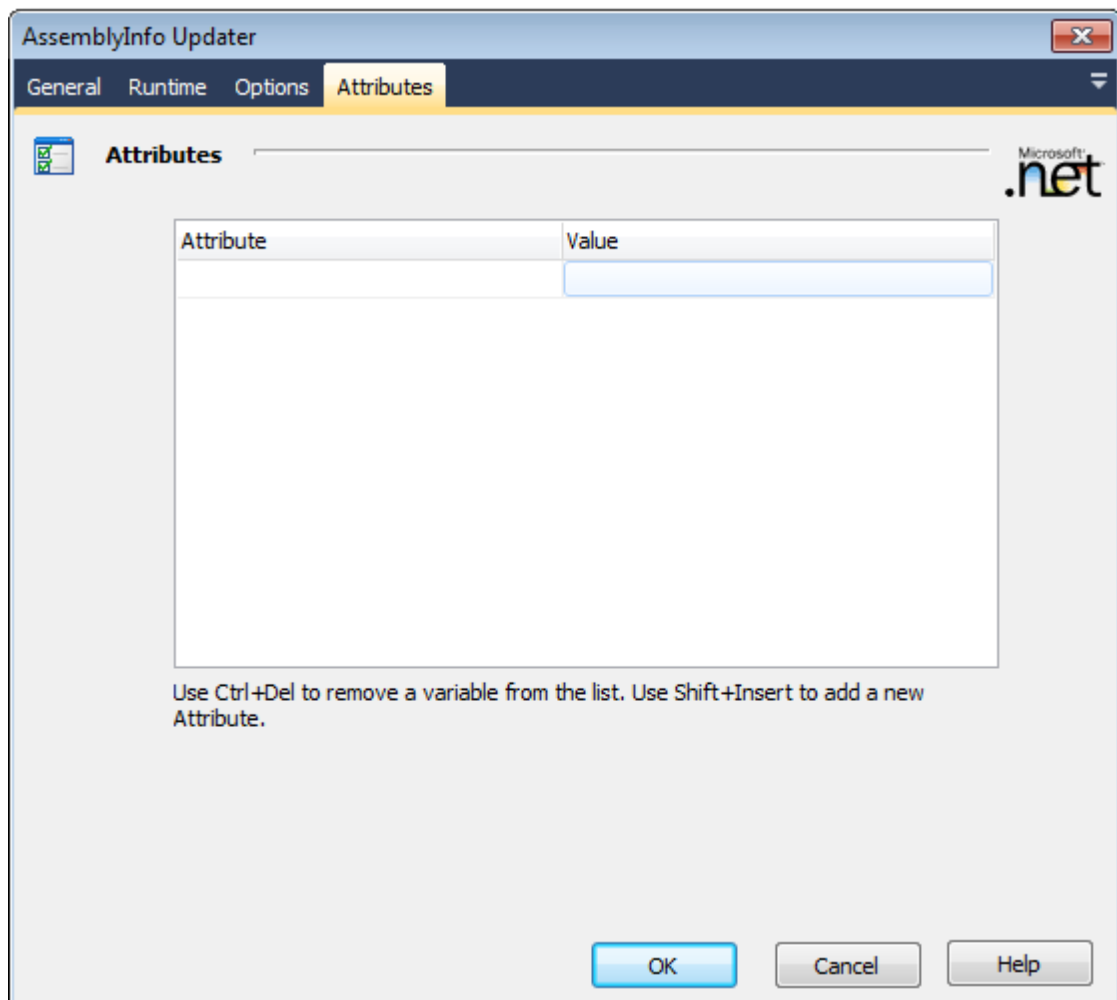
Options

Backup Files Before Modifying - A .bak file will be created before each AssemblyInfo file is modified. For instance, AssemblyInfo.cs will be backed up as AssemblyInfo.cs.bak.

Update Only - No new attributes will be added to the file. Only existing attributes will be modified.

Update Assembly Version

If this option is enabled, the AssemblyVersion attribute will be automatically updated using the value of the specified .NET Version Property Set.



If the value you want to add to the assembly requires quotes, eg.
[assembly: AssemblyFileVersion("1.0.0.3")]

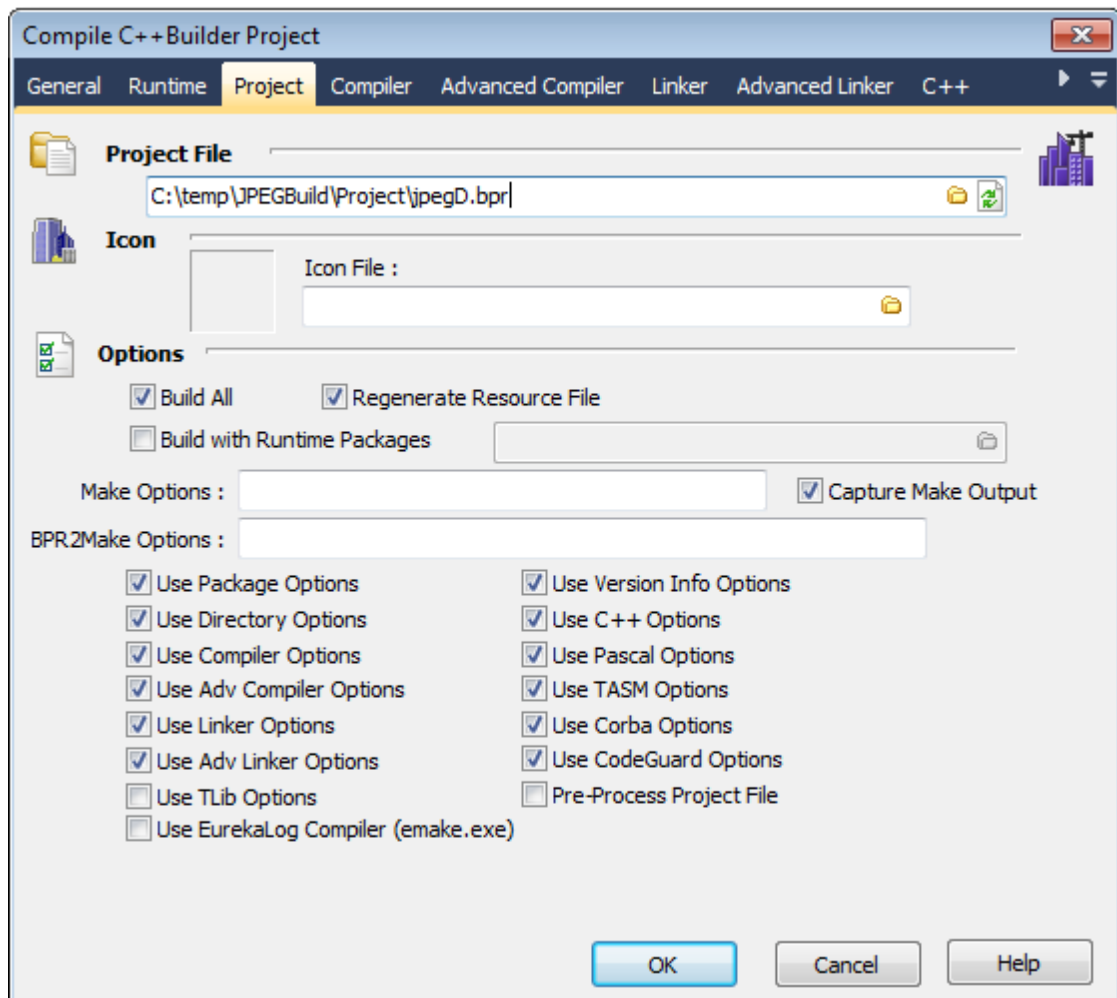
Then you need to specify the quotes around the Value. FinalBuilder doesn't know any specifics about the attributes you are adding, so you have to manually specify the quotes. Although, if you are updating an existing value, then FinalBuilder will look to see if the value is quoted and preserve the quotes.

6.7.2 Embarcadero

6.7.2.1 Embarcadero C++ Builder Action

This action provides support for Embarcadero's C++ Builder versions 5 & 6. The action allows you to selectively override the settings in the project file with the settings in FinalBuilder. It provides the same interface as the Project Options dialog in BCB.

Note: there are some Known Problems with this action.



This action also allows you to generate Version Info for the project, and auto increment build numbers.

The way the action works is this :

- 1) In the BCB Action's properties dialog, select a BCB 5 or 6 project. FinalBuilder will read the settings from the project file.
- 2) When you run the action, it will again read in the project file, and then apply the settings from the Action in FinalBuilder.

You can choose which settings from FinalBuilder are used on the project tab, so for example you can choose to use the compiler & linker settings from the project file and

the directories & version info settings from the action.

3) If you have FinalBuilder generate version info, it will save a .rc file and invoke brcc32

4) FinalBuilder saves a temporary copy of the project file, with the merged settings, then calls BPR2MAK and the Make.

Embarcadero C++Builder Scripting Reference

6.7.2.1.1 Embarcadero C++Builder Compiler Options

Scripting Info

The Action.CompilerOpt properties available are :

property Optimisation : TBCBCompilerOptimisations;
Valid values are combinations of : coPentiumScheduling,coInlineIntrinsic,
coInductionVariables,coCommonSubExpression

property OutputWarnings : TBCBCompilerWarnings;
valid values are : cwNone,cwAll,cwSelected

property DebugInfo : boolean;
property LineNoInfo : boolean;
property DisableInline: boolean;
property PreCompiledHeaders : TBCBCompilerPreCompiledHeader;
Valid Values are pcNone,pcUse,pcCache

property PreCompFile : string;
property StopAfterFile : string;

property MergeDupStrings : boolean;
property StackFrames : boolean;
property EnumsAsInt : boolean;
property ShowGeneralMessages : boolean;
property ExtendedErrorInfo : boolean;

6.7.2.1.2 Embarcadero C++Builder Advanced Compiler Options

Scripting Info

The Action.AdvCompOpt properties available are :

property InstructionSet : TCBuilderInstructionSet;
Valid values are : is386,is486,isPentium,isPentiumPro

property DataAlignment : TCBuilderDataAlignment;
Valid values are : daByte,daWord,daDoubleWord,daQuadWord

property CallingConvention : TCBuilderCallingConvention;
Valid values are : ccC,ccPascal,ccRegister,ccStdCall

property RegisterVariables : TCBuilderRegisterVariables;

Valid values are : rvNone,rvAuto,rvKeyword

property OutputAutoDepInfo: boolean;
property GenerateUnderscores: boolean;
property FloatingPointNone : boolean;
property FloatingPointFast : boolean;
property FloatingPointFixFDIV : boolean;
property LanguageCompliance : TCBuilderLanguageCompliance;
Valid values are : lcEmbarcadero,lcANSI,lcUnixV,lcKR

property NestedComments : boolean;
property MFCCompatibility : boolean;
property IdentifierLength : integer;

6.7.2.1.3 Embarcadero C++Builder Advanced Linker Options

Scripting Info

The Action.AdvLinkerOpt properties available are :

property CaseInsensitiveLink: boolean;
property CalcChecksum: boolean;
property ReplaceResources: boolean;
property UserMajorVersion: integer;
property UserMinorVersion: integer;
property ImageComment : string;
property DelayLoadDLLS : string;

6.7.2.1.4 Embarcadero C++Builder CodeGuard Options

Scripting Info

The Action.CodeGuardOpt properties available are :

property CodeGuardValidation: boolean;
property ValidateGlobalAndStackAccess: boolean;
property ValidateThisPointer: boolean;
property ValidatePointerAccesses: boolean;

6.7.2.1.5 Embarcadero C++Builder Corba Options

Scripting Info

The Action.CorbaOpt properties available are :

property GenerateObjectWrappers: boolean;
property GenerateTie: boolean read;
property GenerateVirtualImplInh: boolean;
property GenerateTypecodeInfo: boolean;
property GenerateIncludeFilesCode: boolean;
property GenerateStreamOperators: boolean;
property HeaderExtension: string;
property AdditionalIDLOptions: string;
property IDLAddServerUnit: boolean;

property IDLAddClientUnit: boolean;
property IDLPrecompiledHeaders: boolean;
property CorbaIncludeIR: boolean;
property CorbaIncludeDSI: boolean;

6.7.2.1.6 Embarcadero C++Builder CPP Options

Scripting Info

The Action.CppOpt properties available are :

property MemberPointers : TCBuilderMemberPointers;
 Valid values are : mpAllCases,mpMultipleInheritance,mpSingleInheritance,
 mpSmallest

property HonourMemberPrecision: boolean;
property DontRestrictForLoop: boolean;
property DontMangleCodeModifiers: boolean;
property ZeroLengthEmptyBaseClasses: boolean;
property ZeroLengthEmptyClassMembers: boolean;
property ExternalTemplates: boolean;
property EnabledRTTI: boolean read;
property EnabledExceptions: boolean;
property LocalInformation: boolean
property DestructorCleanup: boolean;
property FastExceptionPrologs: boolean;
property VirtualTables : TCBuilderVirtualTables;
 Valid values are : vtSmart,vtLocal,vtExternal,vtPublic

property ForceCPPCompile: boolean

6.7.2.1.7 Embarcadero C++Builder Linker Options

Scripting Info

The Action.LinkerOpt properties available are :

property DebugInfo : boolean;
property UseDynamicRTL: boolean;
property UseDebugLibraries: boolean;
property GenerateImportLib: boolean;
property GenerateLibFile: boolean;
property DontGenerateStateFile: boolean;
property ShowMangledNames: boolean;
property MaxErrors: integer;
property MapFile : TCBuilderMapFile;
 Valid values are : cmOff,cmSegments,cmPublics,cmDetailed

property LinkerWarnings : TCBuilderLinkerWarnings;
 Valid Values are : lwAll,lwSelected

property MaxStackSize: Cardinal;
property MinStackSize: Cardinal;
property MinHeapSize: Cardinal;
property MaxHeapSize: Cardinal;
property ImageBase: Cardinal;
property SubSystemMinor : Cardinal;
property SubSystemMajor : Cardinal;

6.7.2.1.8 Embarcadero C++Builder Pascal Options

Scripting Info

The Action.PascalOpt properties available are :

property Optimisation : boolean;
property AlignFields : TCBUILDERPascalFieldAlign;
Valid values are : fa1,fa2,fa4,fa8

property StackFrames : boolean;
property SafeDivide : boolean;

property VarStringChecks : boolean;
property BoolEval : boolean;
property ExtendedSyntax : boolean;
property TypedPointers : boolean;
property OpenStrings : boolean;
property HugeStrings : boolean;
property AssignableConst : boolean;

property RangeChecking : boolean;
property IOChecking : boolean;
property OverflowChecking : boolean;
property DebugInfo : boolean;
property LocalSymbols : boolean;
property ReferenceInfo : boolean;
property DefinitionsOnly : boolean;
property Assertions : boolean;

property ShowHints : boolean;
property ShowWarnings : boolean;

6.7.2.1.9 Embarcadero C++Builder Scripting Reference

Scripting Info

The Action properties available are :

procedure SetupRelease; //same as the Release button on the compiler options page

procedure SetupDebug; //same as the Debug button on the compiler options page

procedure SetVersionInfoKey(name : WideString; Value : WideString);

function GetVersionInfoKey(name : WideString) : WideString;

property ProjectFile : string;
property IncludePath : string;
property LibraryPath : string;
property IntermediateOutputPath : string;
property FinalOutputPath : string;
property BPILIBOutputPath : string;
property Packages : string;
property UsePackages : boolean;

```

property MakeOptions      : string;
property BuildAll         : boolean;
property Conditionals     : string;
property BPR2MakOptions   : string;
property RegenerateResource: boolean;
property IncludeVerInfo: boolean;
property MajorVersion: integer;
property MinorVersion: integer;
property ReleaseVersion: integer;
property BuildVersion: integer;
property AutoIncBuild: boolean;
property IsDebug: boolean;
property IsPreRelease: boolean;
property IsPrivate: boolean;
property IsSpecial: boolean;
property IsDLL: boolean;
property VersionInfoKeys: string;

property Locale: integer;
property CodePage: integer;
property IncludeCompileDate: boolean;
property IconFile: string;
property CompilerOpt      : OleVariant;
property AdvCompilerOpt   : OleVariant;

property LinkerOpt        : OleVariant;
property AdvLinkerOpt     : OleVariant;
property CppOpt           : OleVariant;
property PascalOpt        : OleVariant;
property TASMOpt          : OleVariant;
property CorbaOpt         : OleVariant;
property CodeGuardOpt     : OleVariant;
property TLibOpt          : OleVariant read GetTLibOptions;

property CaptureMakeOutput : boolean;

property UseRegForBPLPath : boolean;
property UseRegForBPIPath : boolean ;

property UseDirectorySettings: boolean;
property UseCompilerSettings: boolean;
property UseAdvCompilerSettings: boolean;
property UseTLibSettings : boolean;
property UseLinkerSettings: boolean;
property UseAdvLinkerSettings: boolean;
property UseCPPSettings: boolean;
property UsePascalSettings: boolean;
property UseVersionInfo: boolean;
property UseTASMSettings: boolean;
property UseCorbaSettings: boolean;
property UsePackageSettings: boolean;
property UseCodeGuardSettings: boolean;

```

Embarcadero C++ Builder Compiler Options Scripting Reference

Embarcadero C++ Builder Linker Options Scripting Reference

Embarcadero C++ Builder Advanced Compiler Options Scripting Reference

Embarcadero C++ Builder Advanced Linker Options Scripting Reference

Embarcadero C++ Builder TLib Options Scripting Reference

Embarcadero C++ Builder CPP Options Scripting Reference

Embarcadero C++ Builder Pascal Options Scripting Reference

Embarcadero C++ Builder TASM Options Scripting Reference

Embarcadero C++ Builder Corba Options Scripting Reference

Embarcadero C++ Builder CodeGuard Options Scripting Reference

6.7.2.1.10 Embarcadero C++Builder TASM Options

Scripting Info

The Action.TASMOpt properties available are :

property DebugInfo : TCBuilderTASMDDebugInfo;
Valid values are : tdNone,tdLineNumbers,tdFull

property CaseSensitivity : TCBuilderTASMCASESensitivity;
Valid values are : csNone,csGlobals,csAll

property Warnings : TCBuilderTASMWarnings;
Valid values are : twNone,twLevel1,twLevel2

property HashTableCapacity : Cardinal read;
property MaxPasses : Cardinal read;
property MaxSymbolLength : Cardinal;
property GenerateListing: boolean;
property ListingType: TCBuilderTASMListingType;
Valid values are : ltNormal,ltExpanded

property CrossReference: boolean read;
property SymbolTablesInfo: boolean;
property IncludeErrorMessages: boolean;
property IncludeFalseConditionals: boolean;
property AddDirective : string;

6.7.2.1.11 Embarcadero C++Builder TLib Options

Scripting Info

The Action.TLibOpt properties available are :

property CaseSensitive : boolean;
property CreateExtendedDictionary : boolean;
property PurgeComments : boolean;
property PageSize : integer;

```

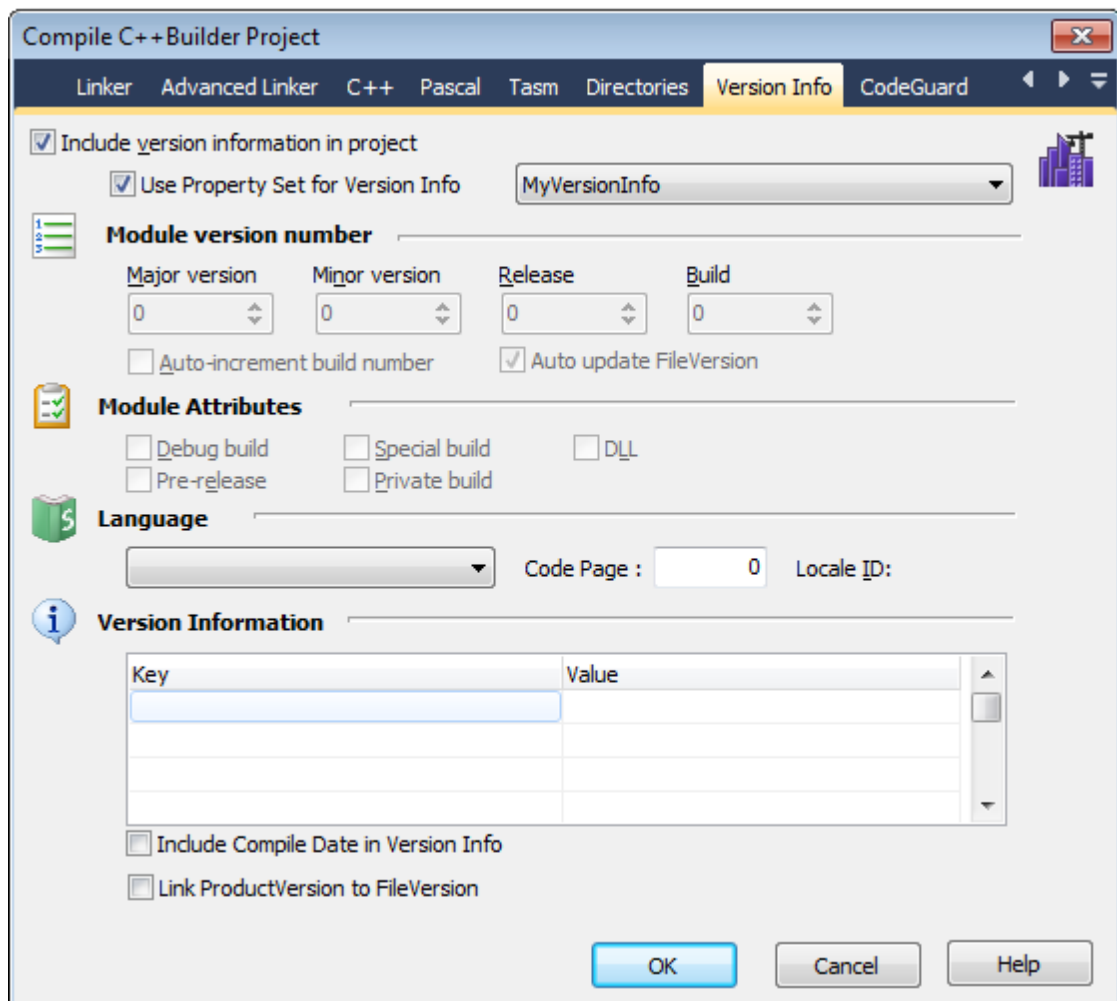
property ListFile           : string;
property UseRTL             : boolean;

```

6.7.2.1.12 Using PropertySets with Embarcadero C++ Builder

To use PropertySets with Embarcadero C++ Builder, first you need to define a property set using the PropertySet define action and choose either Win32 Version Numbers or Win32 Version Info.

Then in the Compile Embarcadero C++ Builder action on the Version Info property page, select "Use property set for version info" and select the PropertySet to use:



When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

6.7.2.2 Build Delphi Project

This action allows you to automate the compilation of Delphi projects. FinalBuilder supports the following versions of Embarcadero/Codegear Delphi:

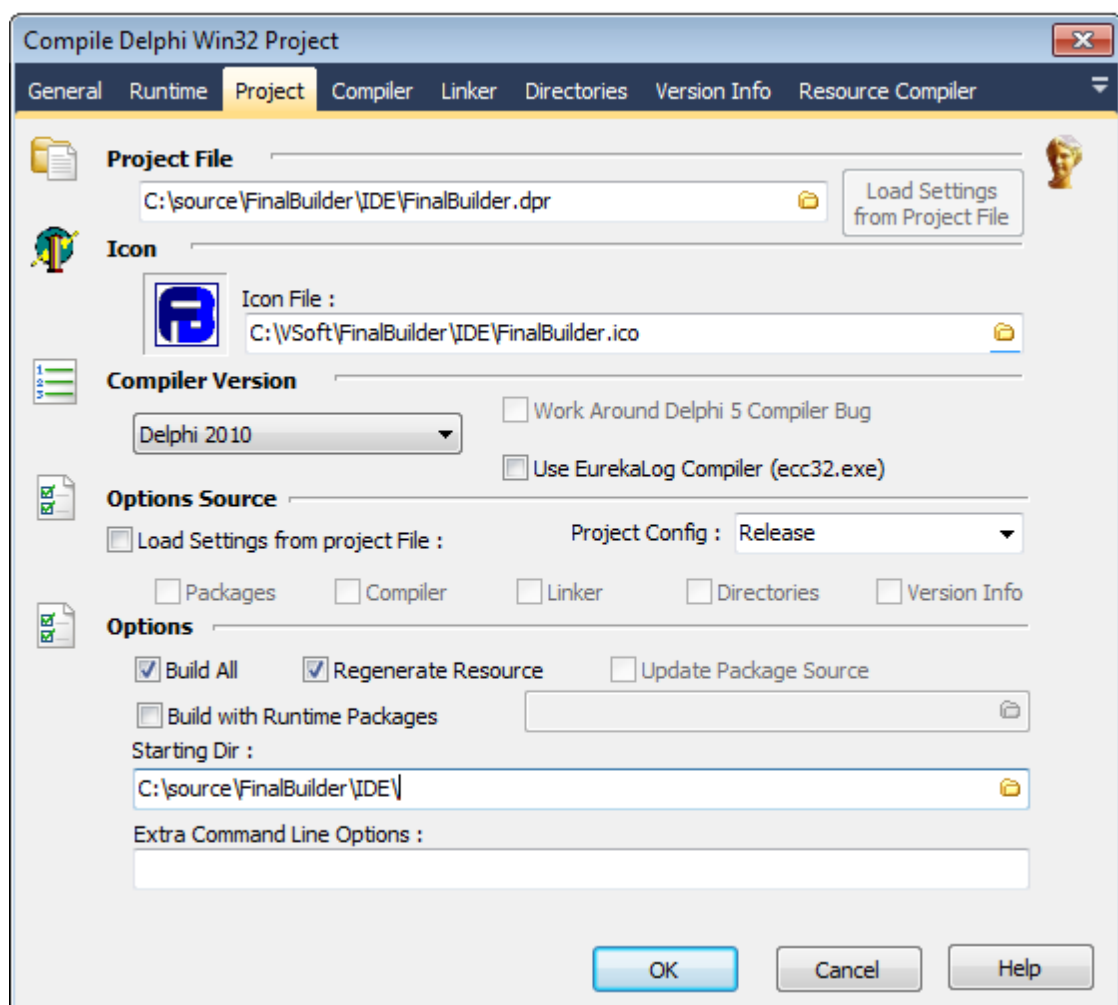
- Delphi 3, 4, 5, 6 & 7
- Delphi 2005
- Delphi 2006

- Delphi 2007

Set the compiler version you need from the "Compiler Version" section of the "Project" tab (see below.)

Many of the property pages for this action use properties which are identical to the properties in Delphi IDE. The properties on the Compiler, Linker, Directories and Version Info pages are described in the Delphi Help File. Note that you may use FinalBuilder Variables in the Directories page but not on the Version Info page.

This action can also be used to maintain the version info for the selected Delphi project. When the AutoIncrement property (on the version info tab) is enabled, the version number properties (Major, Minor, Release and Build) are persisted to a file (yourproject.fbd) after the action runs (and after the AfterAction script event). These values will be restored when the project is loaded. If the action fails, the values are not persisted. This makes the AutoIncrement property function the same as it does in the Delphi IDE.



The options that are specific to FinalBuilder are provided on the Project property page, pictured above.

Project File : This is the path to the Delphi project, Delphi package or unit file you wish to compile. Note that you must choose the **.dpr** or **.dpk** file for a project or package, not the **.bdsproj/.dproj/.dof** file (all of the latter are project settings files.)

Load Settings From Project File : The Delphi IDE maintains a file with the extension **.dof**, .

bdsproj or .dproj with the project settings. Press this button to automatically populate all of the action's property pages with the settings from this file. You should do this when you first create the action. Once the settings are loaded, they will not be reloaded into FinalBuilder from the Delphi IDE unless you press this button again, or check the "Use Settings From Project File" option (see below.)

Icon file : This is the path to the icon that will be used as the main icon for the application. When you select a project file, FinalBuilder checks to see if there is a resource file with the same name as the project (ie the default one generated by the Delphi IDE). If the file exists, FinalBuilder extracts the MAINICON resource and saves it in the same folder as mainicon.ico. This is needed for the resource compiler.

Compiler Version: Choose the version of Delphi to use to compile.

Work around Delphi 5 Compiler Bug : Check this property (which is only enabled when the compiler version is D5) to work around the problem where the project compiles fine but no executable is produced. The will save having to work around it manually by adding a second action.

Use EurekaLog Compiler : Check this property to use the EurekaLog compiler instead of Delphi. See the Using EurekaLog topic for more information.

Use Settings from Project File : Checking this box will cause FinalBuilder to ignore any settings made on the action's property pages, and instead always use the values from the project settings file. You can choose to apply this option only to certain groups of settings, any combination of Packages, Compiler, Linker, Directories and/or Version Info can be loaded from settings. Note that checking this option disables the relevant property page(s).

Project Config : (Delphi 2007 only) Select which configuration of the project you want to build.

Build All : Tells the command line compiler to rebuild all files in the project. This is checked by default. When not checked, the command line compiler will only recompile modified units.

Regenerate Resource : When checked, FinalBuilder will regenerate the project resource file before compiling the project. It calls the Embarcadero Resource Compiler to do this. You should leave this option checked if you use version info in your project and change any version numbers between builds

Update Package Source : Enable this option to update the package source with changes when the action runs.

Build with runtime Packages : This is the same as the equivalent option in the Delphi project options.

Starting Dir : Select the starting directory for the action.

Extra Command Line Options : Enter any extra command line options to be passed to dcc32.exe.

NOTE: You need write access to the directory you are compiling the project from, as FinalBuilder generates temporary .cfg file for the command line compiler to use.

Scripting Info

The Action properties available via scripting are :

```
property CompilerOpt : IDelphiCompilerOptions; // See the Compiler Options section below
property ProjectFile : WideString;
//version info stuff
property IncludeVerInfo : WordBool;
property AutoIncBuild : WordBool;
property MajorVersion : integer;
property MinorVersion : integer;
property ReleaseVersion : integer;
property BuildVersion : integer;
property IsDebug : WordBool;
property IsPreRelease : WordBool;
property IsSpecial : WordBool;
property IsPrivate : WordBool;
property IsDLL : WordBool;
property Locale : integer;
property CodePage : integer
property VersionInfoKeys : WideString; // in the format :
    CompanyName=VSoft Technologies Pty Ltd
    FileDescription=FinalBuilder Core API
    FileVersion=0.0.1.541
    InternalName=FBCore10.bpl
    LegalCopyright=© 2000 - 2001 VSoft Technologies Pty Ltd
    LegalTrademarks=FinalBuilder™
    OriginalFilename=FBCore10.bpl
    ProductName=FinalBuilder™
    ProductVersion=1.0.0.0
    Comments=
```

```
property BuildAll : WordBool;
property CompilerVersion : integer; // D3 = 0, D4 = 1, D5 = 2, D6 = 3, D7 = 4, D2005 = 10,
D2006 = 11, D2007 = 12
property ExtraCommandline : WideString;;
property IconFile : WideString;
property RegenerateResource : WordBool;
property WarningsAsError : WordBool;
property HintsAsError : WordBool;
property Packages : WideString
property UsePackages : WordBool
```

Compiler Options

```
property Optimisation : WordBool;
property AlignFields : WordBool;
property StackFrames : WordBool;
property SafeDivide : WordBool;
property VarStringChecks : WordBool;
property BoolEval : WordBool
property ExtendedSyntax : WordBool
property TypedPointers : WordBool
property OpenStrings : WordBool
property HugeStrings : WordBool
property AssignableConst : WordBool
property RangeChecking : WordBool
property IOChecking : WordBool
property OverflowChecking : WordBool
property DebugInfo : WordBool
```

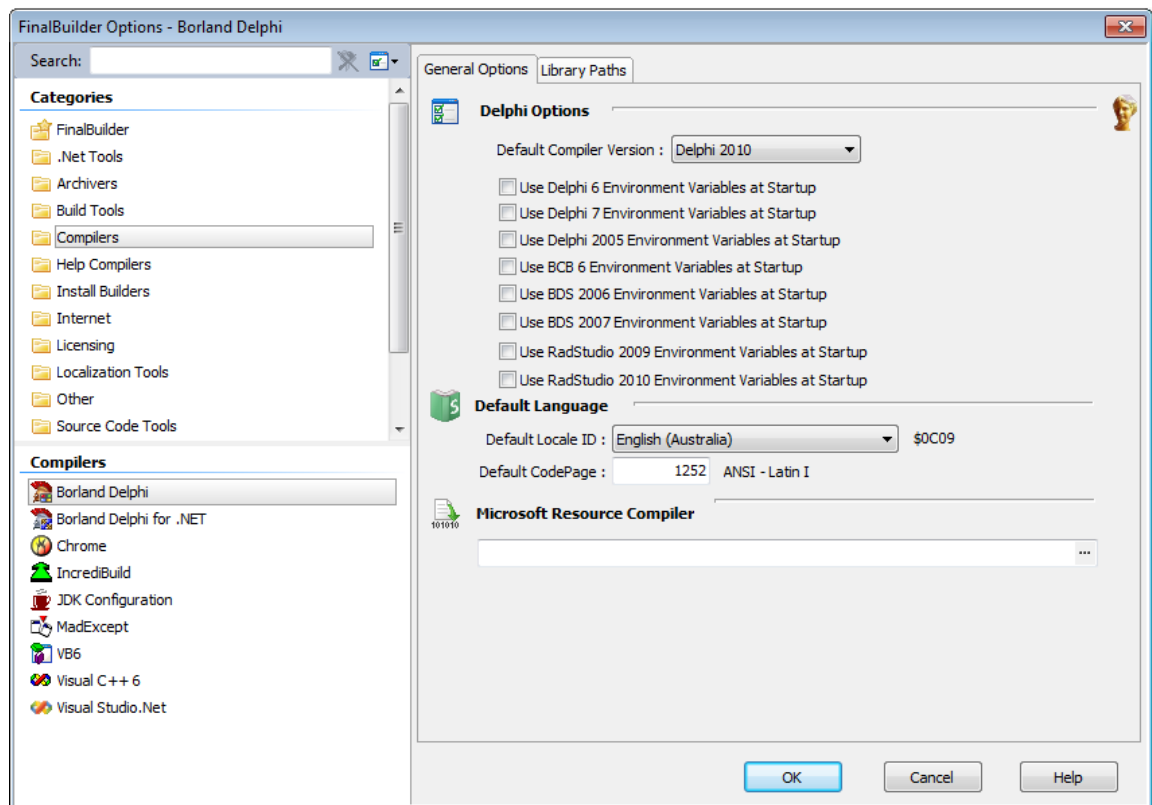
property LocalSymbols : WordBool
property ReferenceInfo : WordBool
property DefinitionsOnly : WordBool
property Assertions : WordBool
property UseDebugDCU : WordBool
property ShowHints : WordBool
property ShowWarnings : WordBool
property Mapfile : integer // valid values are : mfNone, mfSegments, mfPublics, mfDetailed
property ConsoleApp : WordBool
property IncludeTD32 : WordBool
property IncludeRemoteSymbols : WordBool
property LinkerOutput : integer; // valid values are loGenerateDCU, loGenerateCOBJ,
 loGenerateCPlusObj
property IncludeNamespaces : WordBool
property ExportAllSymbols : WordBool
property MinStackSize : Cardinal
property MaxStackSize : Cardinal
property ImageBase : Cardinal
property ExeDescription : WideString
property OutputDir : WideString
property UnitOutputDir : WideString
property BPLOutputDir : WideString
property DCPOutputDir : WideString
property Conditionals : WideString
property SearchPath : WideString
property LibraryPath : WideString
property UnitAliases : WideString

Action methods :

procedure SetVersionInfoKey(name : WideString; Value : WideString);
function GetVersionInfoKey(name : WideString) : WideString;

6.7.2.2.1 Delphi Options

Access the Delphi Options from the Tools Menu | Options | Compilers category and then choose the Embarcadero Delphi item.



Delphi Options

Choose the default Delphi compiler version to use in new Delphi Compiler actions, as well as set the default state of using certain environment variables at startup.

Default Language

Choose the default Locale and CodePage to apply to new Delphi Compiler actions added to your project.

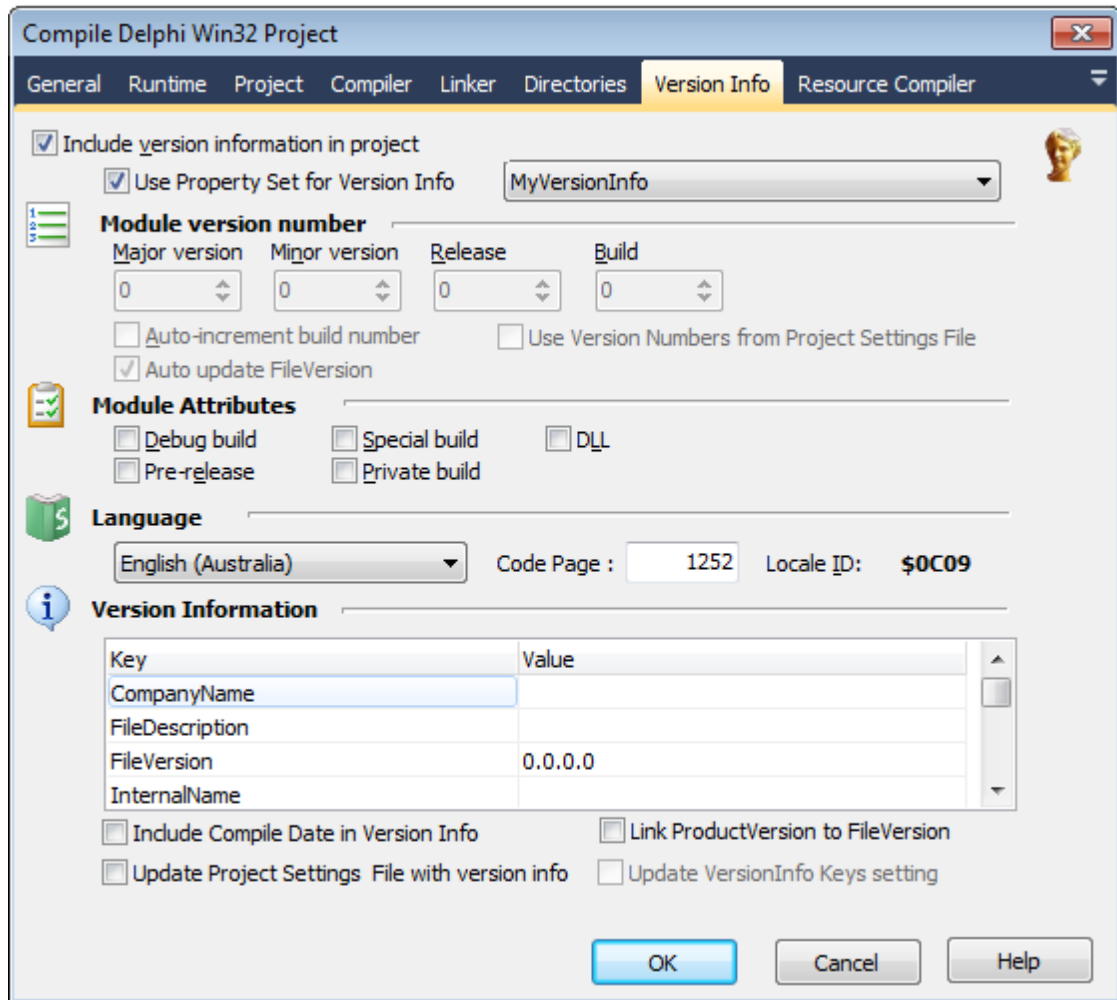
Microsoft Resource Compiler

Enter the location of the Microsoft Resource Compiler (rc.exe). The Microsoft resource compiler can be used instead of the Embarcadero provided one.

6.7.2.2.2 Using PropertySets with Delphi

To use PropertySets with Delphi, first you need to define a property set using the PropertySet define action and choose either Win32 Version Numbers or Win32 Version Info.

Then in the Compile Delphi Win32 action on the Version Info property page, select "Use property set for version info" and select the PropertySet to use:



When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

6.7.2.2.3 Using EurekaLog

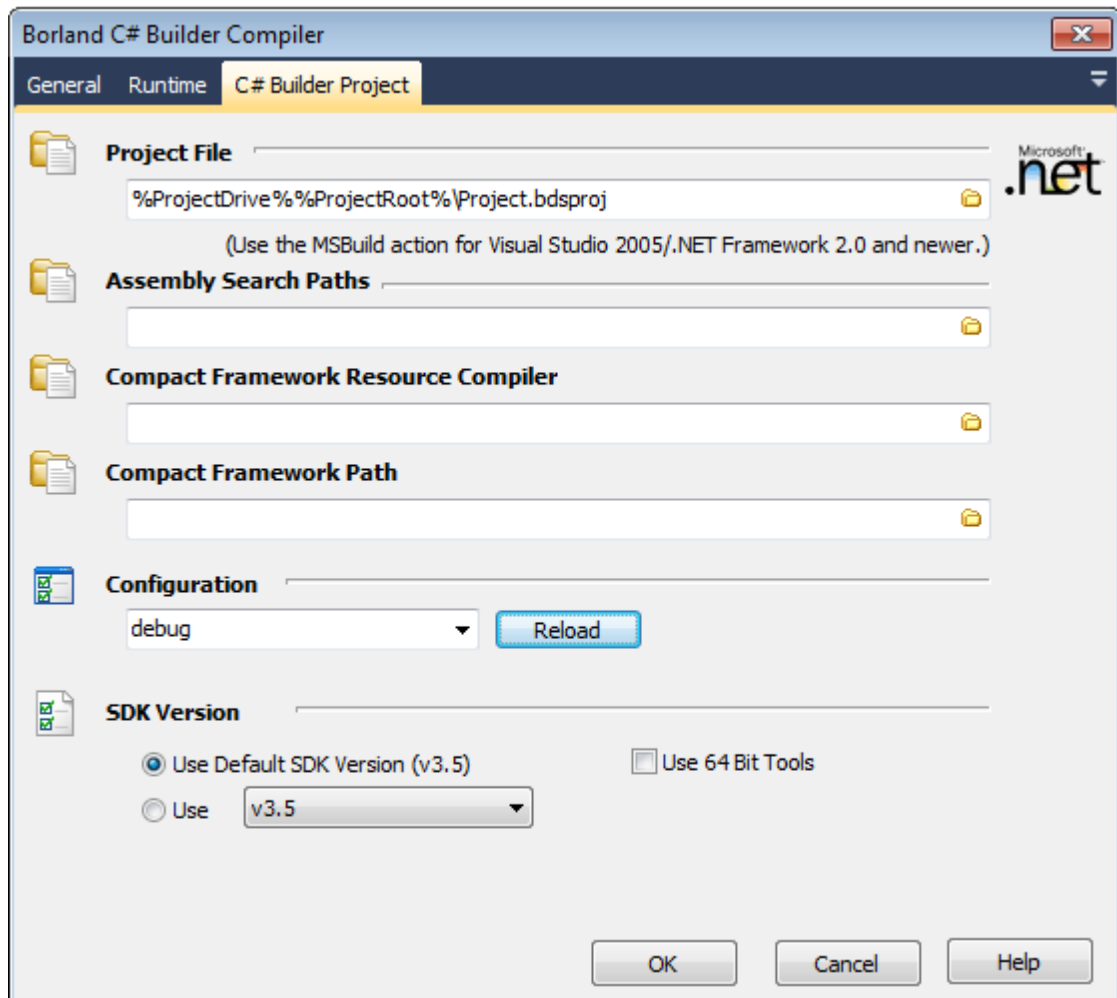
If you have EurekaLog installed, you can set the "Use EurekaLog Compiler" option on the Compile Delphi action Project tab, in order to have the project built with EurekaLog.

In addition to doing this, two options must be set before you get full EurekaLog results:

- "Activate EurekaLog" must be enabled on the project in the Delphi IDE. To do this, open the project in the IDE, go to Project -> EurekaLog Settings and verify that "Activate EurekaLog" is enabled. You will need to save and close the project before the changes take effect.
- This will give you EurekaLog logging, however the EUREKALOG and EUREKALOG_Vx conditional defines will not be set in the project unless you enable the "Use Conditionals from Project File" option on the Directories tab.

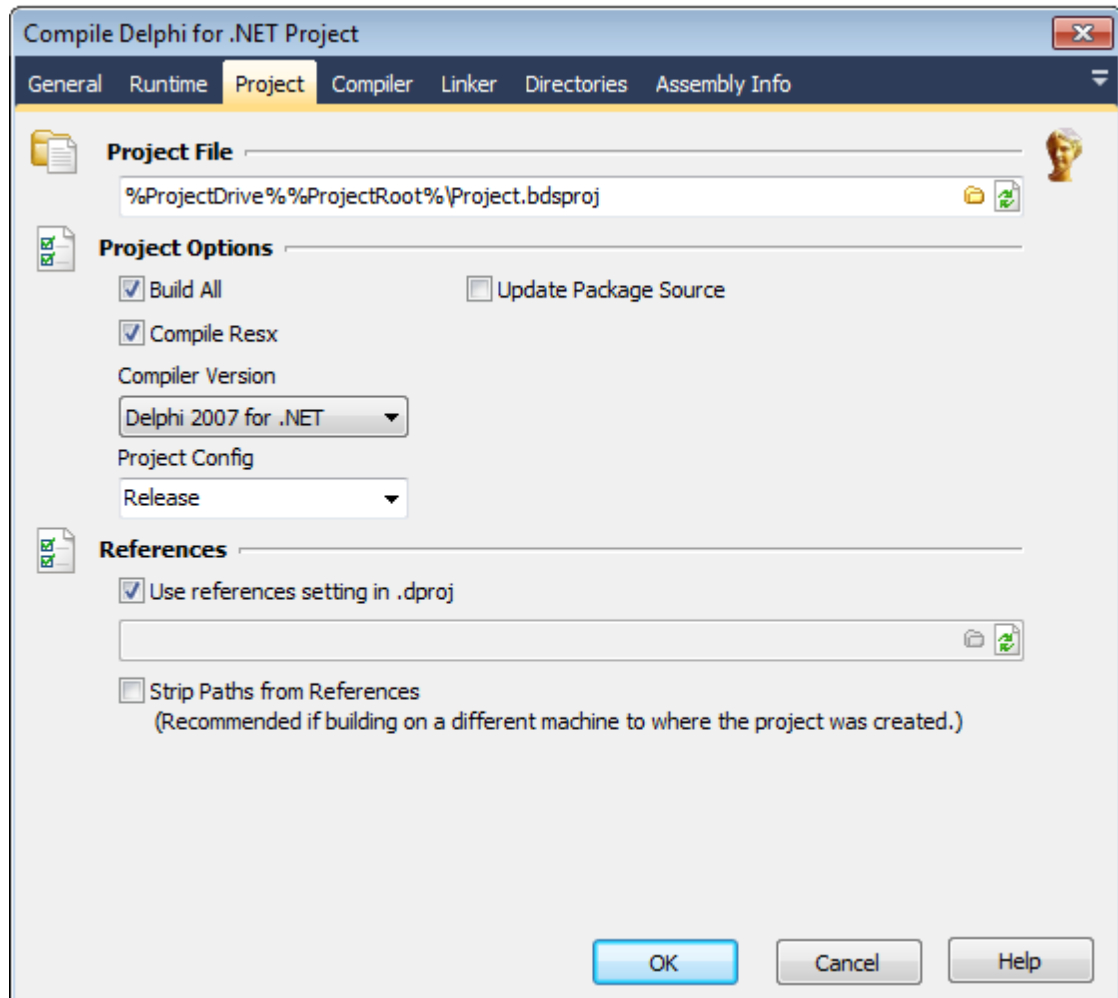
6.7.2.3 Embarcadero C# Builder Project Compiler Action

This action will build Embarcadero C# Builder 1.x projects (.bdsproj) without requiring that C#Builder be installed on your build machine.



6.7.2.4 Delphi for .NET Action

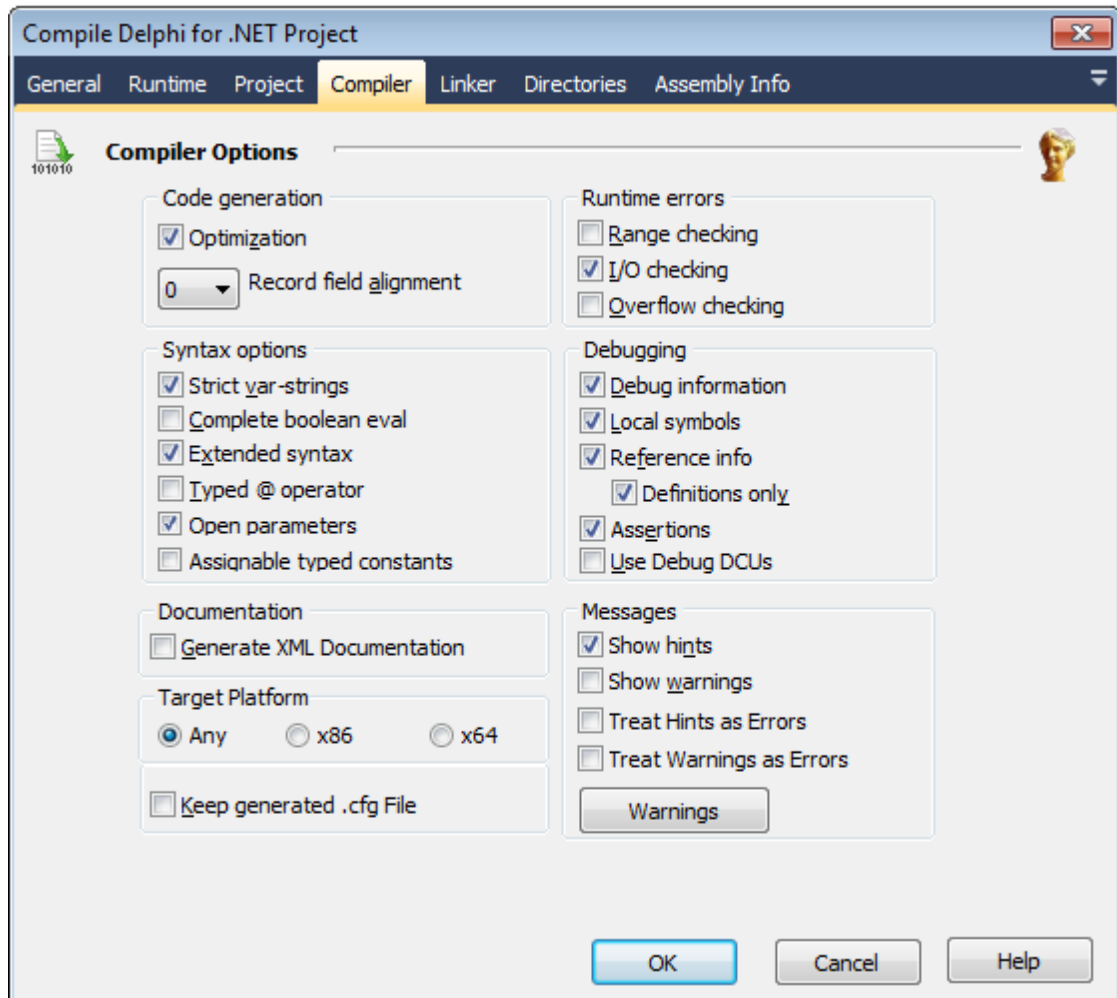
This action will build Embarcadero Delphi™ for .NET projects (.bdsproj). Delphi for .NET is significantly different from earlier versions of Delphi, thus requiring a separate action.

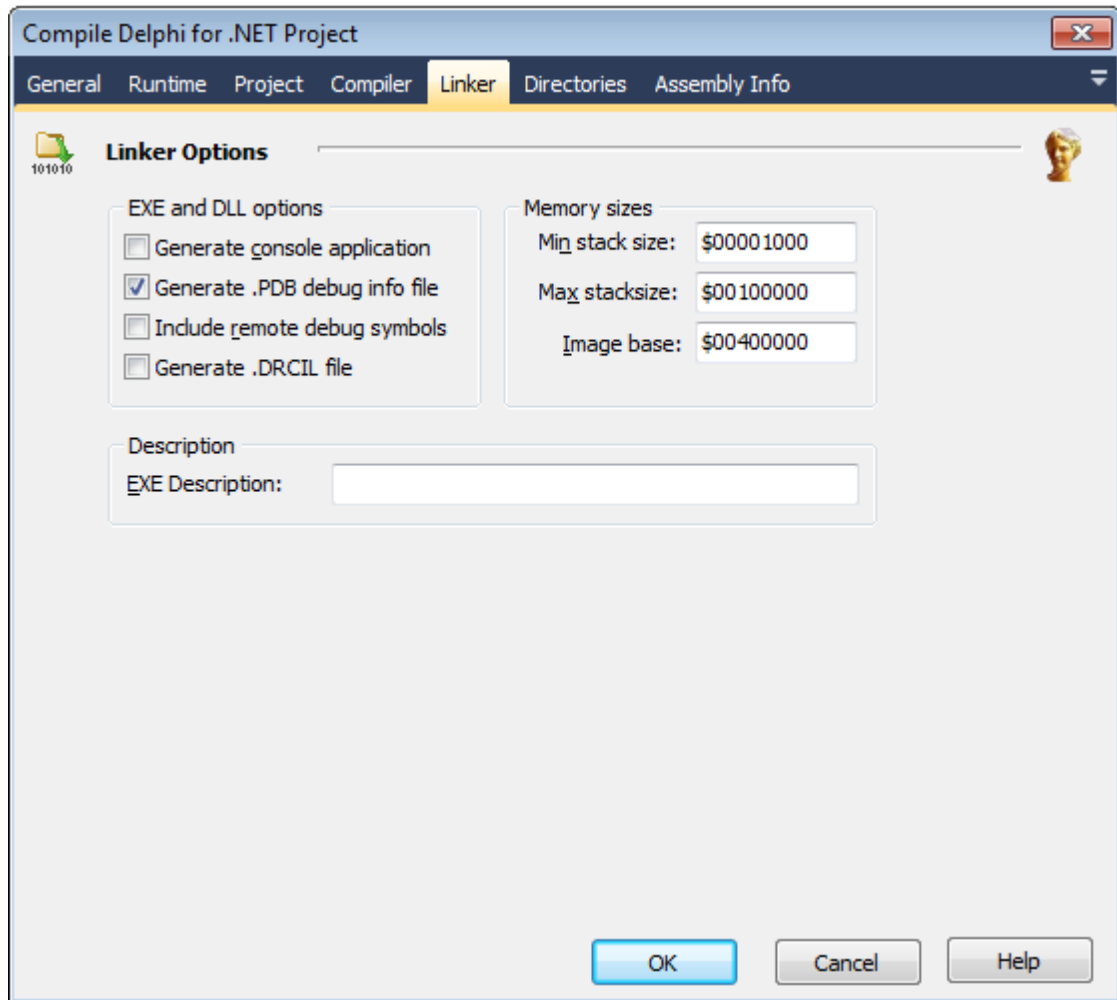


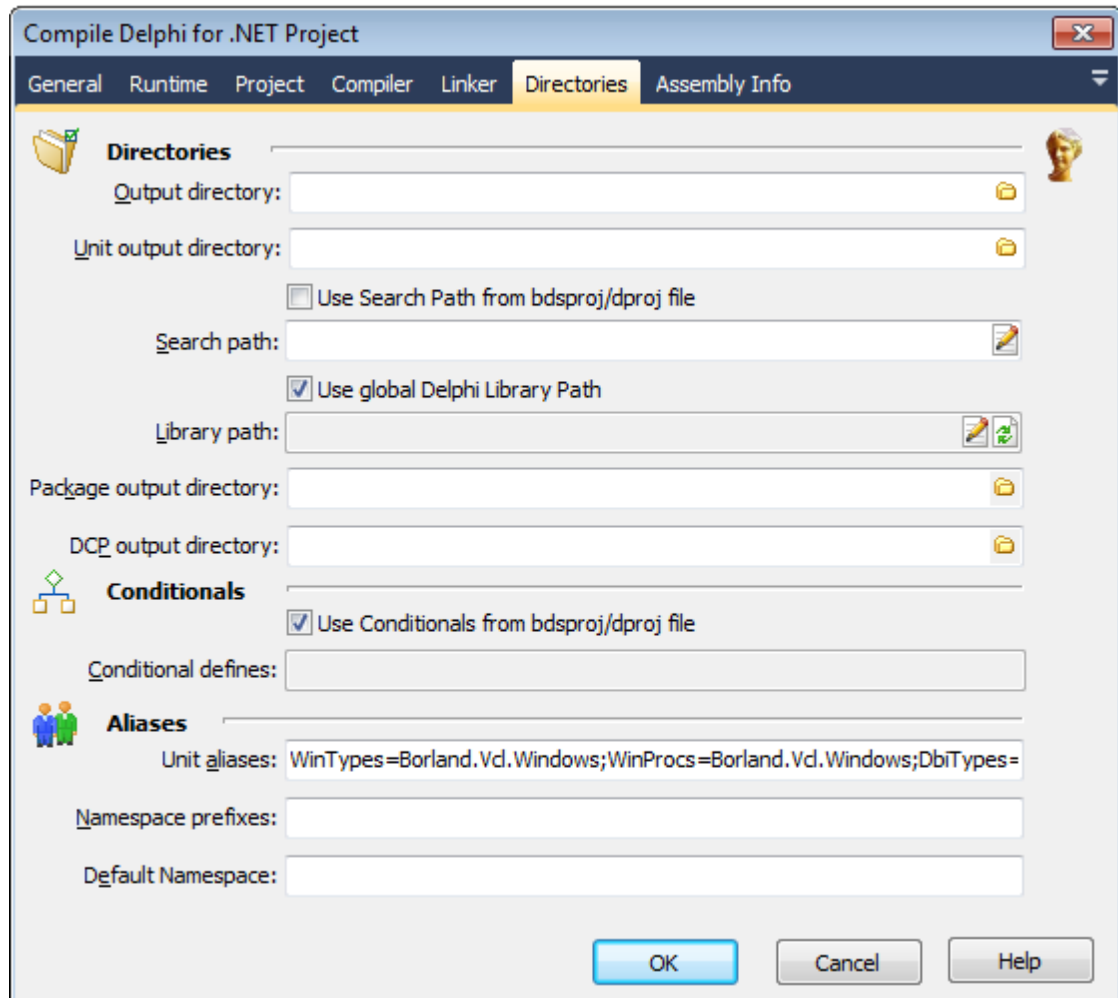
Project File - the path to your .bdsproj file (only for Delphi for .NET projects, for Win32 see the Compile Delphi Project action).

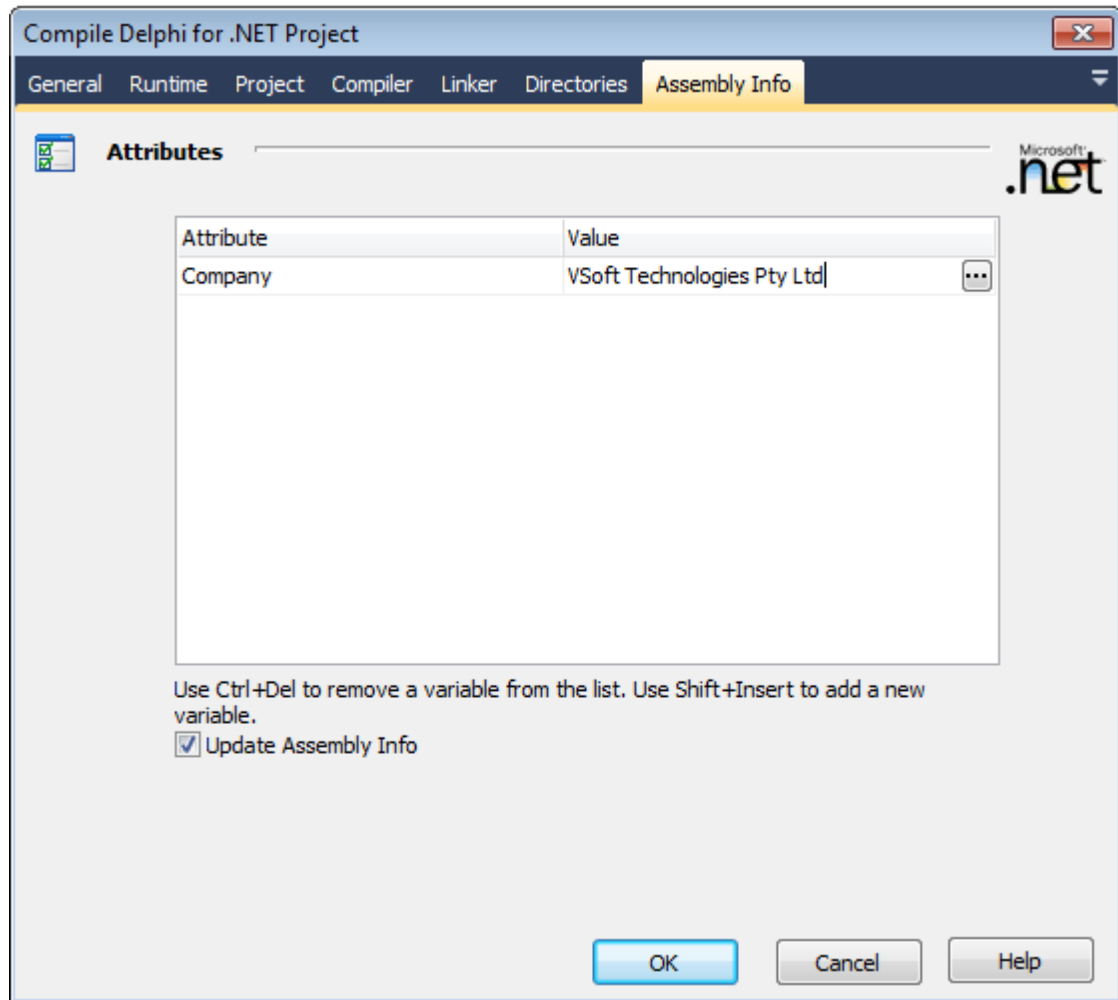
Compiler Version - choose the compiler version to compile your project

Build All - perform a complete rebuild all



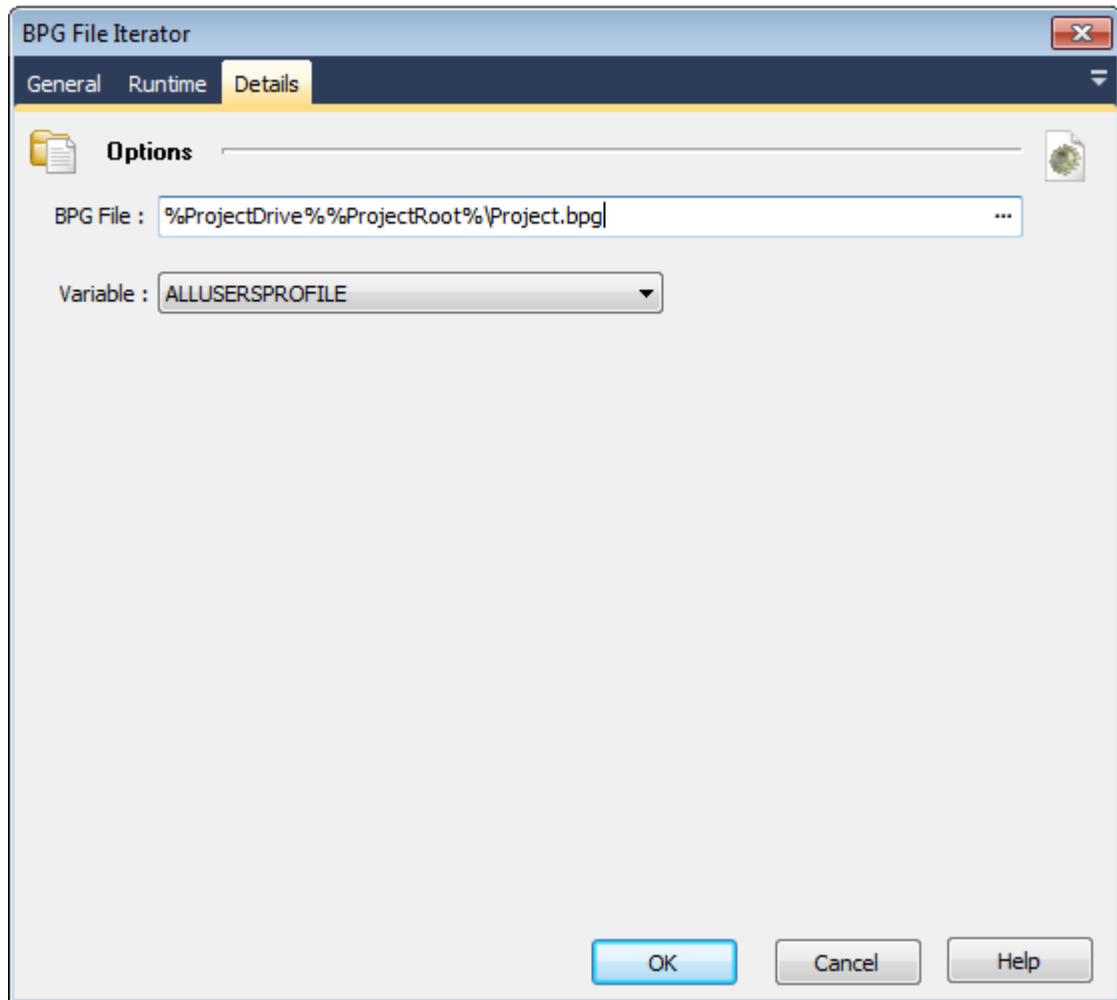






6.7.2.5 BPG File Iterator

The BPG File Iterator action will run each of its child actions for each project found in the specified Embarcadero Project Group (bpg) file specified.

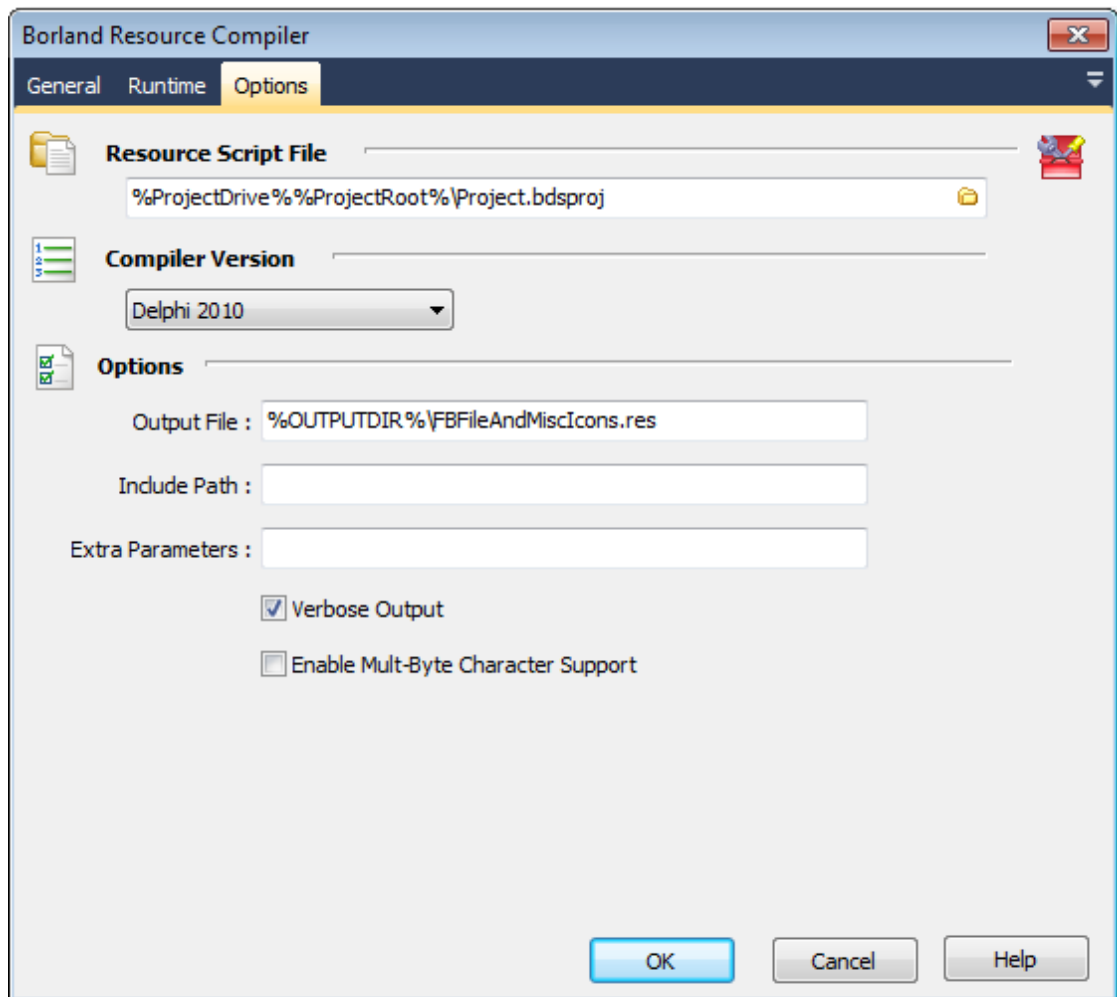


BPG File - specify the project group file which you want to iterate over

Variable - this variable will be set to the project file for each iteration. So you would use this variable in the child actions to perform some function (eg. Compile the project, or check it out of your version control system for example).

6.7.2.6 Compile Embarcadero Resource Script

This action provides an easy to use interface for the Embarcadero Resource Compiler.



Resource Script : The .rc file that you wish to compile.

Output File : The path to the Output .res File.

Include Path : The path to use for resource files included with the #include command

Compiler Version : The Embarcadero Resource compiler is provided as part of Delphi And C++Builder. This option allows you to choose which version of dcc32 to use.

Verbose Output : Tells the resource compile to report its compile status for each line. Otherwise, it will only report errors.

Enable Multi-Byte Character Sets : Enable multi-byte character support

Scripting Info

The Action properties available are :

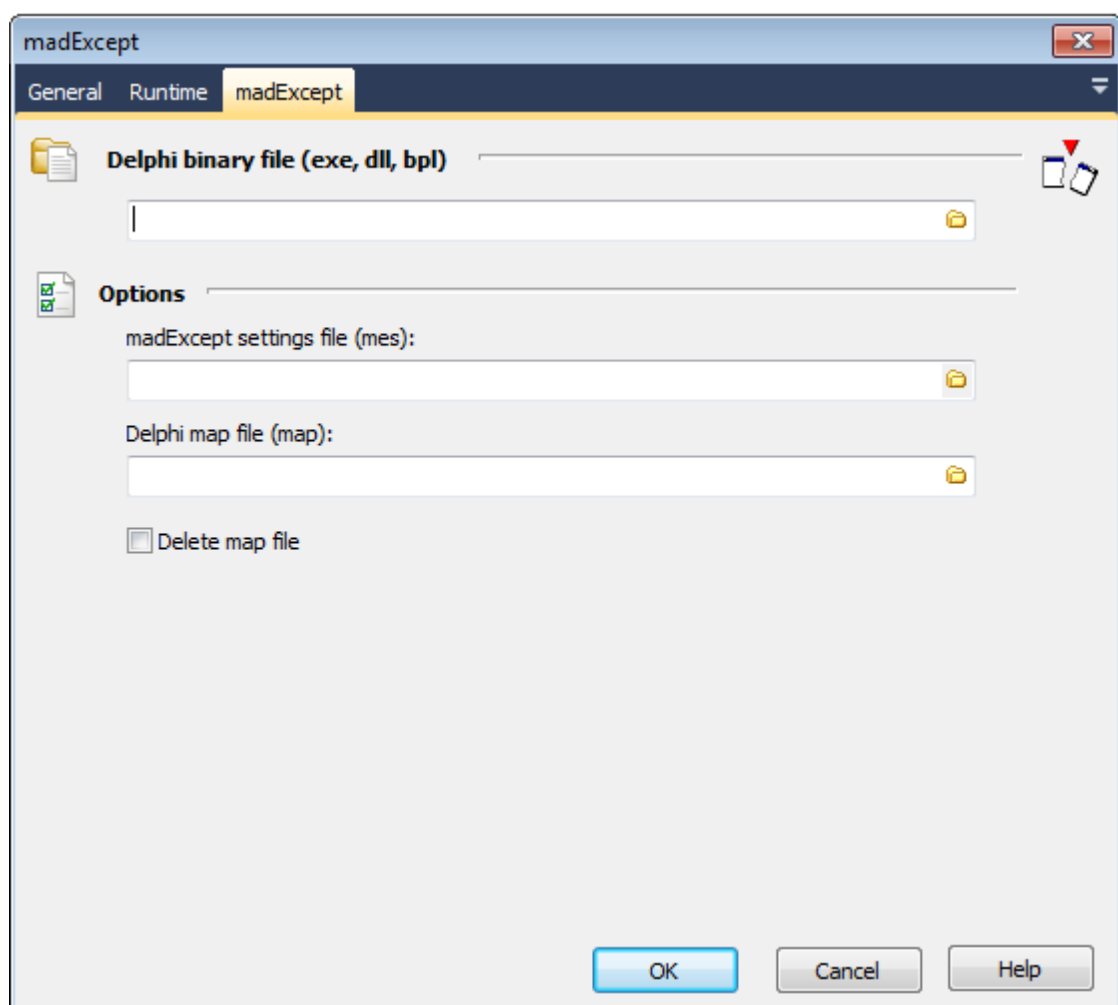
property ScriptFile : WideString;

property OutputFileName : WideString;
property VerboseOutput : WordBool;
property CompilerVersion : integer; // the valid values are :
 0 = Delphi 3
 1 = Delphi 4
 2 = Delphi 5
 3 = Delphi 6
 4 = Delphi 7
 5 = BCB 3
 6 = BCB 4
 7 = BCB 5
 8 = BCB 6
 9 = BCB 7
property IncludePath : WideString;
property EnableMBCS : WordBool;

6.7.2.7 MadExcept Compiler Action

This Action provided support for the MadExcept product from <http://www.madshi.net>

(while not strictly a compiler, MadExcept is a commonly used addon to the Embarcadero Delphi compiler which is why it's in the Compiler category.)



Specify a Delphi binary file and corresponding .Map and MadExcept settings (.Mes) file.

6.7.3 Chrome

The Chrome action enables you to automate compilation of your Chrome projects

Chrome™ is RemObjects' next generation Object Pascal language for the .NET and Mono Platforms. While implementing a language that stays true to the beauty and elegance of Object Pascal, Chrome adds useful design elements from other languages such as C#, Java and Eiffel, and it introduces its own language innovations.

Use Chrome to write fully managed native .NET applications for the Microsoft .NET Framework, the Compact Framework or the Mono Platform, and develop your applications inside the well known Visual Studio .NET IDE.

For more information on Chrome, see <http://www.chromesville.com/>

6.7.4 Incredibuild

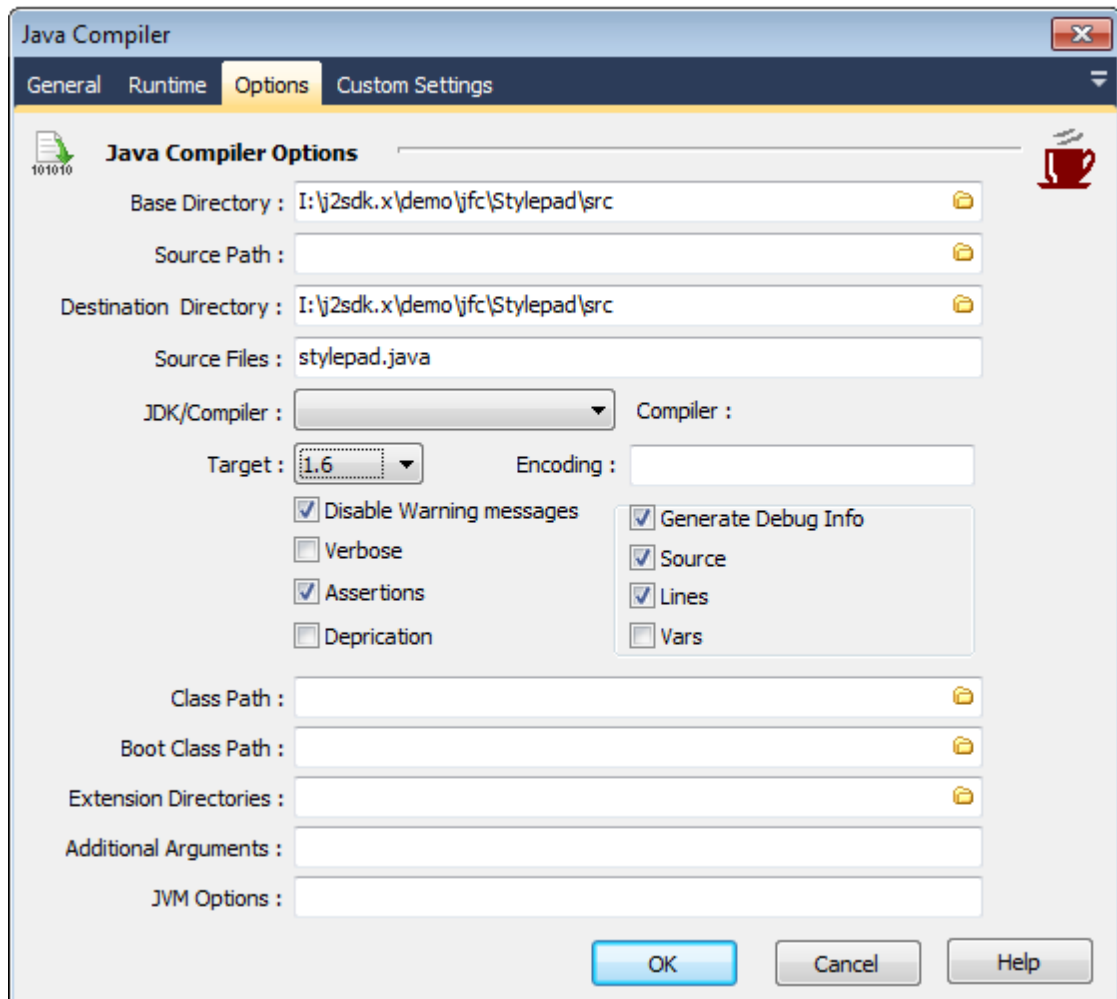
The IncrediBuild actions enable you to automate IncrediBuild distributed compiles.

IncrediBuild Actions:

- IncrediBuild - executes an IncrediBuild distributed compilation
- Enable IncrediBuild Agent - enables the agent on the client machine
- Disable IncrediBuild Agent - disables the agent on the client machine
- Stop Current IncrediBuild Compile - stops any running IncrediBuild compilation
- Reset IncrediBuild Swapfile - resets the client machines IncrediBuild swap file

For more information on Xoreax IncrediBuild, please see <http://www.xoreax.com/>

6.7.5 Java Compiler Action



The Java Compiler Action invokes the `javac` compiler which is included in the Sun Java JDK. It also supports invoking other java compilers such as Embarcadero Java Compiler and IBM Jikes.

Base Directory - The current directory when the compiler is invoked.

Source Path - Specify the source code path to search for class or interface definitions. As with the user class path, source path entries are separated by semicolons (;) and can be directories, JAR archives, or ZIP archives. If packages are used, the local path name within the directory or archive must reflect the package name. Note that classes found through the classpath are subject to automatic recompilation if their sources are found.

Destination Directory - Sets the destination directory for class files. The destination directory must already exist; `javac` will not create the destination directory. If a class is part of a package, `javac` puts the class file in a subdirectory reflecting the package name, creating directories as needed. For example, if you specify `-d c:\myclasses` and the class is called `com.mypackage.MyClass`, then the class file is called `c:\myclasses\com\mypackage\MyClass.class`.

If not specified, `javac` puts the class file in the same directory as the source file.

Note that the directory specified is automatically added to your user class path.

Source Files - One or more source files to be compiled (such as MyClass.java).

JDK/Compiler - The JDK Config to use. See the JDK Configuration Section in the Options Dialog (under compilers).

Target - Generate class files that will work on VMs with the specified version. The default is to generate class files to be compatible with the 1.2 VM in the Java 2 SDK. The versions supported by **javac** in the Java 2 SDK are:

- 1.1 Ensure that generated class files will be compatible with 1.1 and VMs in the Java 2 SDK.
- 1.2 Generate class files that will run on VMs in the Java 2 SDK, v 1.2 and later, but will not run on 1.1 VMs. This is the default.
- 1.3 Generate class files that will run on VMs in the Java 2 SDK, v 1.3 and later, but will not run on 1.1 or 1.2 VMs.
- 1.4 Generate class files that are compatible only with 1.4 VMs.

ClassPath - Set the user class path, overriding the user class path in the CLASSPATH environment variable. If neither CLASSPATH or -classpath is specified, the user class path consists of the current directory.

BootClassPath = Allows cross compilation using the bootstrap and extension classes of a different java platform implementation.

Extension Directories - Cross compile against the specified extension directories

Additional Arguments - Allows you to pass additional command line arguments to the compiler (ie arguments not exposed as properties of the action).

JVM Options - Allows you to specify arguments that will be passed to the jvm when it is invoked.

Generate Debug Info - Generates Debug info during the compile

- Source - Source file debugging information
- Lines - Line number debugging information
- Vars - Local variable debugging information

Verbose - Verbose output. This includes information about each class loaded and each source file compiled

Assertions - Enables support for compiling source code containing assertions. When Target is set to 1.4, the compiler accepts code containing assertions. Assertions were introduced in J2SE 1.4. When Target is set to 1.3, the compiler does not support assertions. Only valid if Assertions option enabled.

Deprecation - Show a description of each use or override of a deprecated member or class. Without **-deprecation**, **javac** shows the names of source files that use or override deprecated members or classes.

6.7.5.1 Embarcadero Compiler Options

The Embarcadero Compiler does not have any extra settings.

6.7.5.2 JDK Configurations

The JDK Configurations enable the Java action to execute using different JDK configurations, and makes it easy to switch between those configurations.

To Add a new JDK Config, open the FinalBuilder Options dialog, navigate to Compilers, JDK Configurations.

Click on the Add button, then fill in the Name field, for example "Sun JDK 1.4.2.0.5".

The JDK Home Path field should point to the Root folder where the jdk is installed, for example G:\j2sdk1.4.2_05

Select the Compiler Type, in this example "Sun Javac 1.4"

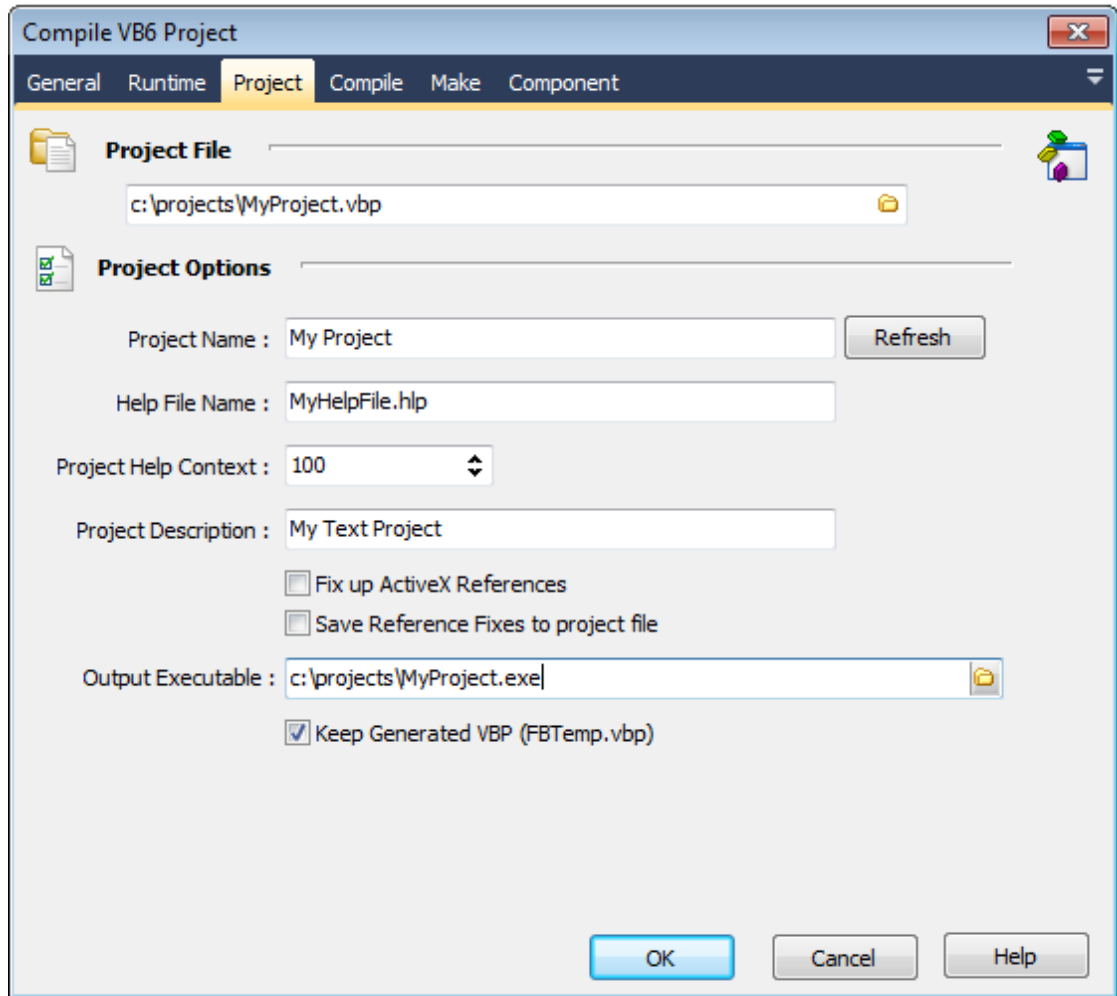
Now we need to add the jdk classpath setting. in the edit box at the bottom of the page, click on the browse button and navigate to the JDK lib folder, eg. G:\j2sdk1.4.2_05\lib and click on ok. Then click on the Add Button, then close the options dialog.

You should now see the jdk config you just created in the dropdown list on the Java Compiler action's property page. You can create as many configurations as you like. Note that these configs are stored in the registry, if you move your project to another machine you should first define those configurations on the new machine.

6.7.6 Microsoft

6.7.6.1 Compile Visual Basic Project

This action provides an interface to the VB6 compiler. This action maintains the Version info and some other settings separate from the visual basic project file (.vbp). Before running the VB compiler, FinalBuilder creates a temporary .vbp file using the settings from the FinalBuilder action and from the actual project.vbp file.



Project File : Specifies the path and file name of the Visual Basic project file that you wish to compile.

Refresh Button : Refreshes the settings from the project.vbp file. This will overwrite the changes that you made in FinalBuilder.

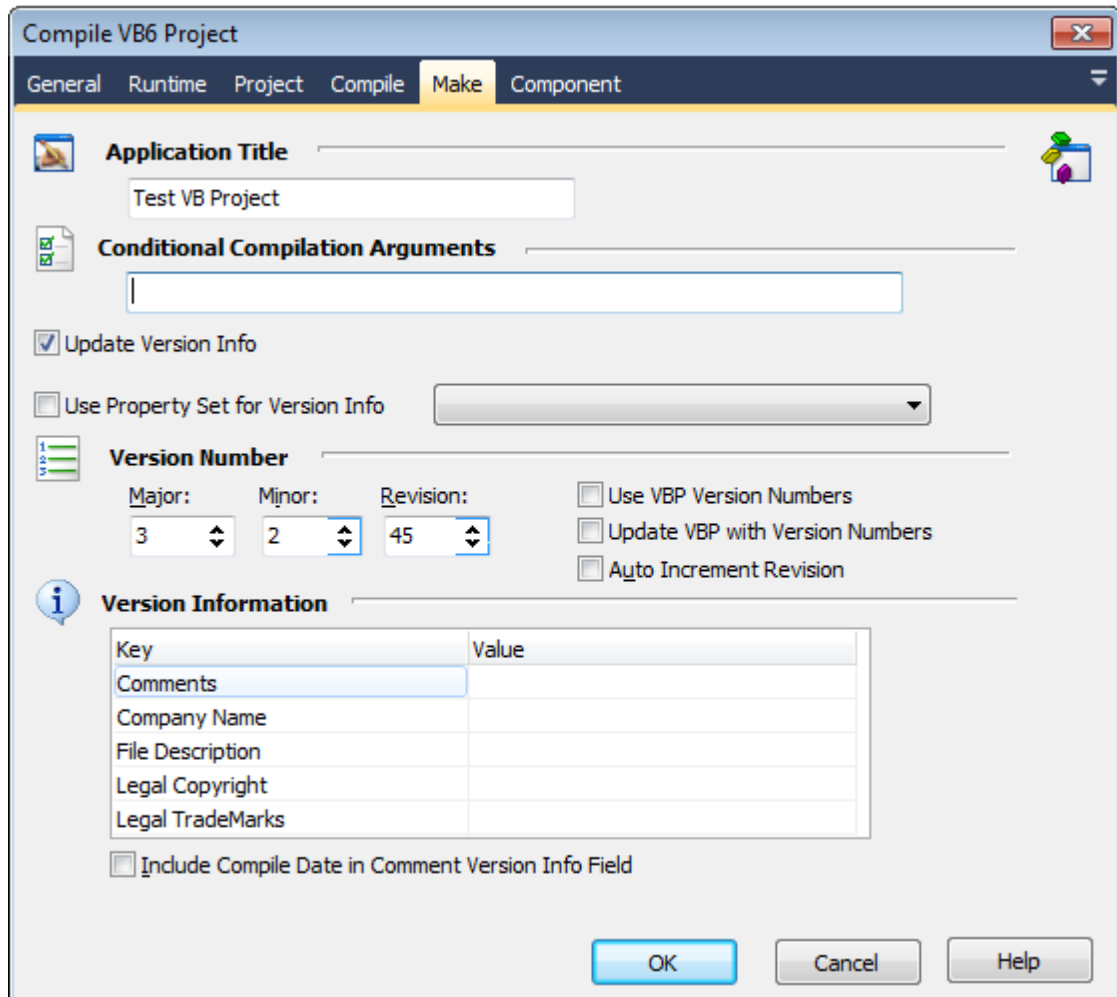
Project Name : Identifies the project in code. It can't contain periods (.), spaces, or start with a non-alphabetic character. For a public class name, the project name and class name cannot exceed a total of 37 characters.

Help File Name : The name of the Help file associated with the project.

Help Context ID : The context ID for the specific Help topic to be called when the user selects the "?" button while the application's object library is selected in the Object Browser.

Project Description : A user-friendly name for the project. Displayed in the References and Object Browser dialog boxes.

Output Executable : The full path and file name that the project will be compiled to.



Note that if you use the AutoInc Version Number, you need to save the FinalBuilder project after building for this to be saved. Alternatively, use a FinalBuilder Project variable and read the value from an ini file before compiling and then write the variable back out to the ini file after compiling.

Note : Some important info on the "Fix up ActiveX References" setting on the Project Tab

When using Project or No Compatibility with ActiveX DLL projects, the CLSID's for those projects change with each compile or interface change (depending on the compatibility setting). This will invalidate references to those dll's in other projects. When the Fix up ActiveX References option is checked, FinalBuilder will check that the references are correct, and if not correct them. It does this by locating the type library and extracting the CLSID and comparing it to the reference entry, if they are not the same then the reference entry is corrected. This will enable FinalBuilder to compile the projects correctly, and also help you avoid the dreaded 429 Ole Automation error.

Scripting Info

The Action properties available are :

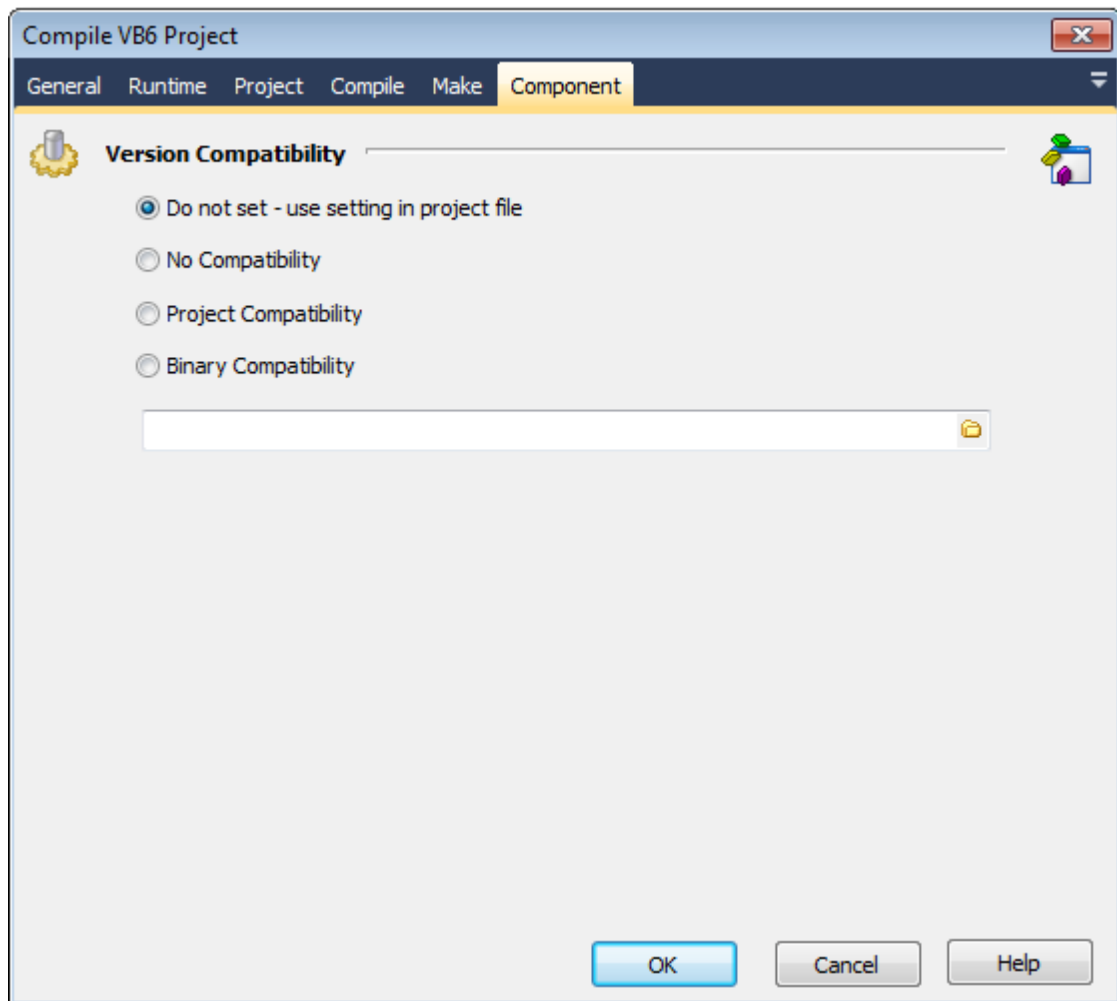
property ProjectFile : WideString
property OutputPath : WideString
property Conditionals: WideString

property ProjectName: WideString
property ProjectDescription: WideString
property ProjectHelpFile: WideString
property ProjectHelpContextID: integer

property Title: WideString
property ExeName: WideString
property MajorVersion: integer
property MinorVersion: integer
property RevisionVersion: integer
property AutoIncrementVersion: WordBool
property VersionCompanyName: WideString
property VersionComments: WideString
property VersionFileDescription: WideString
property VersionLegalCopyright: WideString
property VersionLegalTrademarks: WideString
property VersionProductName: WideString

property CompilationType: integer // valid values are ctPCode or ctNativeCode
property OptimizationType: integer //valid values are otNone, otFast or otSmall
property FavorPentiumPro: WordBool
property CodeViewDebugInfo: WordBool
property NoAliasing: WordBool
property RemoveBoundsCheck: WordBool
property RemoveOverflowCheck: WordBool
property RemoveFPointCheck: WordBool
property RemoveFDIVCheck: WordBool
property AllowUnroundedFP: WordBool
property DLLBaseAddress: Integer

6.7.6.1.1 VB6 Version Compatibility



This page allows you to decide which version compatibility mode you want to use when compiling.

Do not set - this option uses whatever compatibility mode has been set in the project file

No Compatibility

With this setting, new class ID's, new interface ID's and a new type library ID will be generated by VB each time the ActiveX component project is compiled. This will cause any compiled client components to fail (with error 429!) and report a missing reference to the 'VB ActiveX Test Component' when a client project is loaded in the VB IDE.

TIP: Use this setting to compile the initial release of a component to other developers.

Project Compatibility

With this setting, VB will generate new interface ID's for classes whose interfaces have changed, but will not change the class ID's or the type library ID. This will still cause any compiled client components to fail (with error 429!) but will not report a missing reference to the 'VB ActiveX Test Component' when a client project is loaded in the VB IDE. Recompilation of client components will restore them to working order

again.

TIP: Use this setting during the initial development and testing of a component within the IDE and before the component is released to other developers.

Binary Compatibility

VB makes it possible to extend an existing class or interface by adding new methods and properties etc. and yet still retain binary compatibility. It can do this, because it silently creates a new interface ID for the extended interface and adds registration code to register the original interface ID but with a new Forward key containing the value of this new interface ID. COM will then substitute calls having the old ID with the new ID and hence applications built against the old interface will continue to work (assuming the inner workings of the component remain backward compatible!).

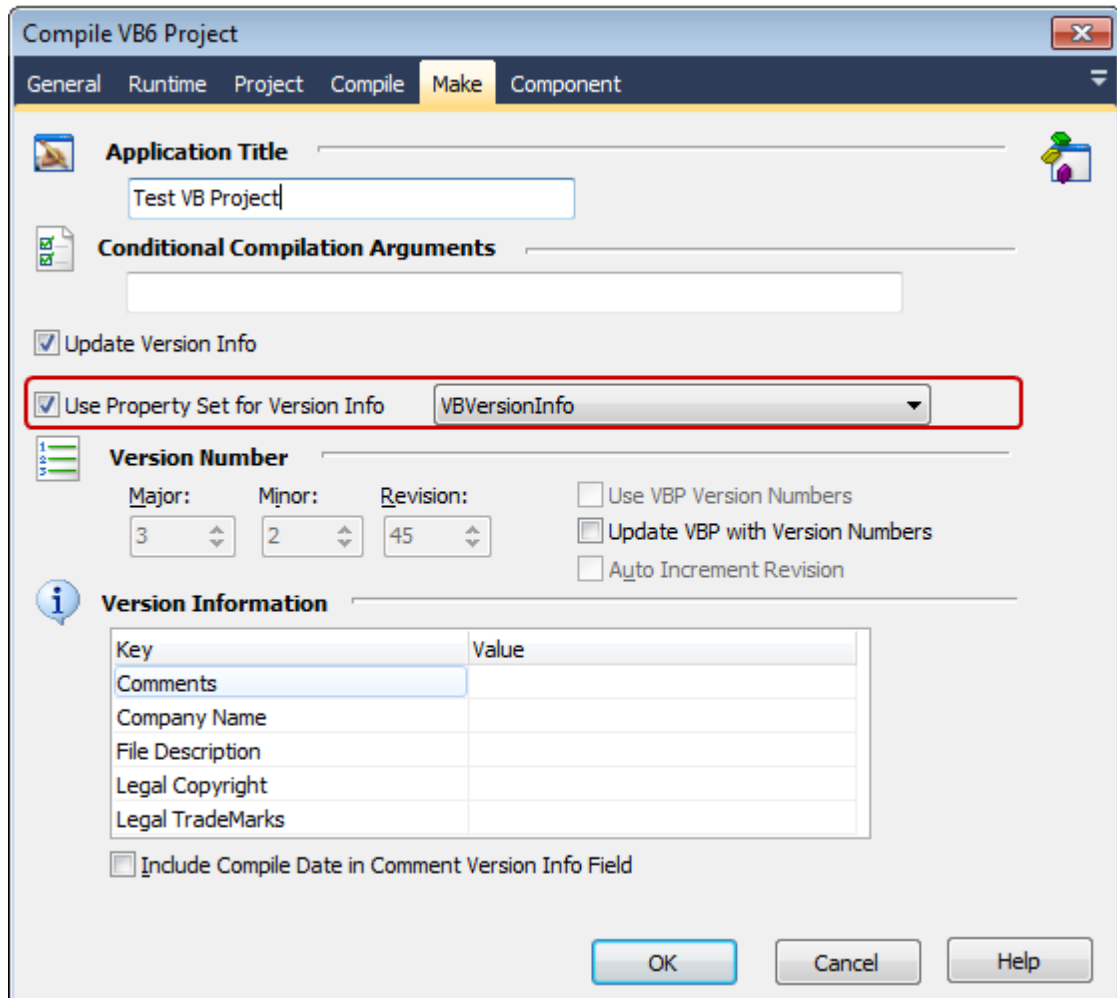
With this setting, VB will not change any of the existing class, interface or type library ID's, however in order that it can do so, VB requires the project to specify an existing compiled version that it can compare against to ensure that existing interfaces have not been broken.

TIP: Use this setting following the release of a component to other developers.

6.7.6.1.2 Using PropertySets with VB6

To use PropertySets with VisualBasic, first you need to define a property set using the PropertySet define action and choose either VisualBasic Version Numbers or VisualBasic Version Info.

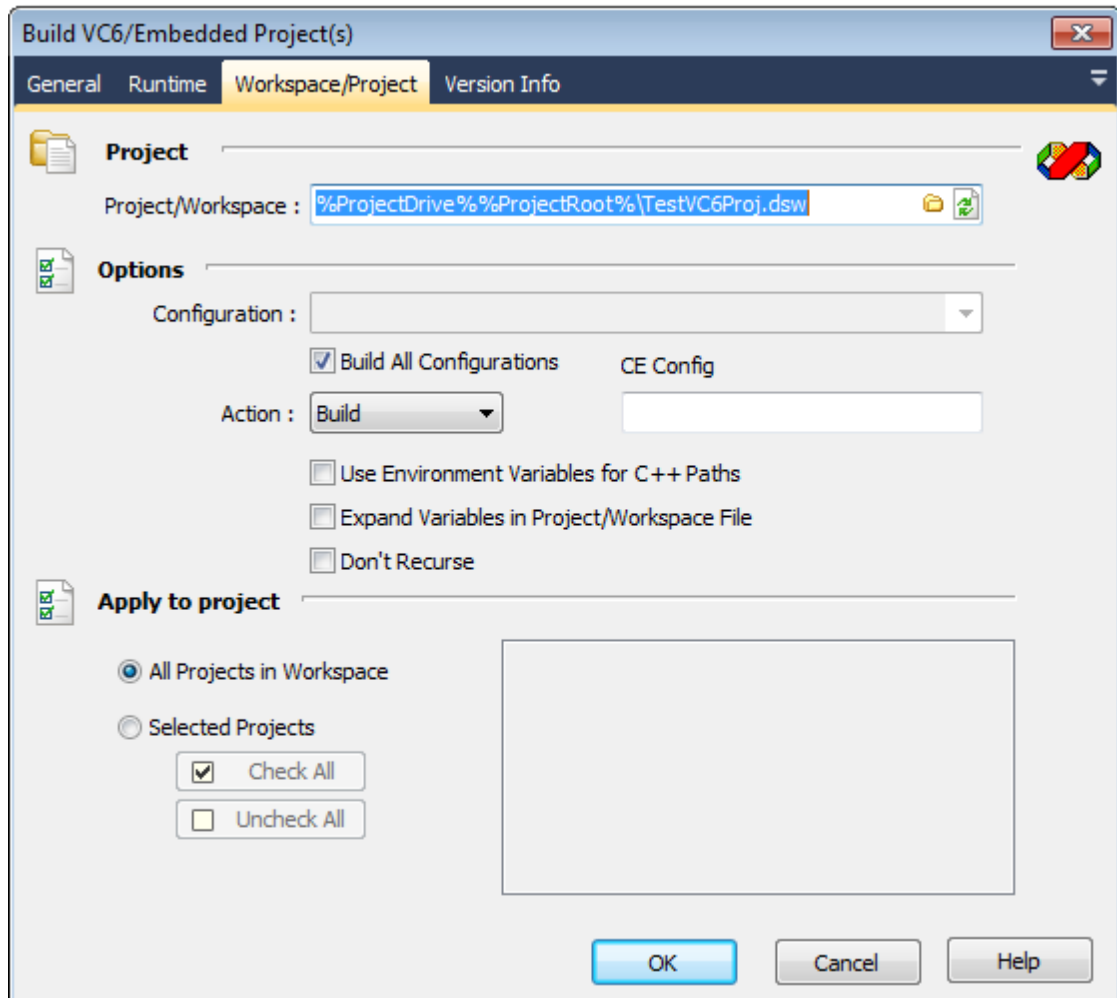
Then in the VisualBasic action on the Make property page, select "Use property set for version info" and select the PropertySet to use:



When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

6.7.6.2 Visual C++ 6 Action

This action provides the ability to build Visual C++ 6 projects and Workspaces. You can choose to build/rebuild/clean all projects in a workspace or selected projects. You can also choose the Configuration to be used (this configuration must exist in all projects in the workspace).



Project/Workspace - enter the name of your project or workspace. The project/workspace types allowed are Visual Studio project file (.dsp), Visual Studio Workspace (.dsw), Embedded VC++ 4.x Workspace (.vcw) or Embedded VC++ 4.x Project (.vcp).

Configuration/Build All - If Build All Configurations is not selected, then enter the name of the configuration you want to build

Action - Choose to Build, Rebuild or Clean the project

CE Config - If building an embedded workspace or project, enter the Config to use

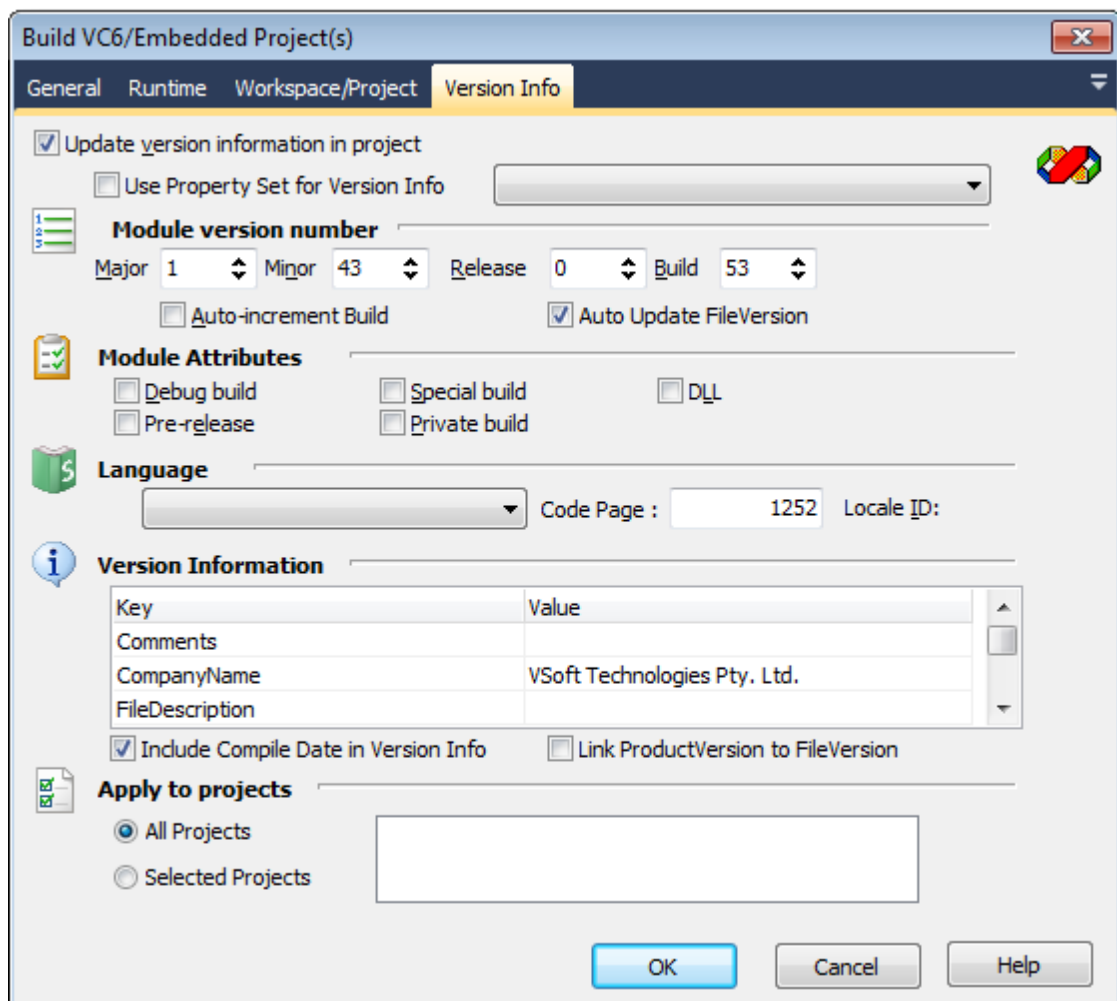
Use Environment Variables for C++ Paths - Use PATH, INCLUDE, LIBPATH, and LIB environment variables instead of IDE paths for VC++ builds.

Expand Variables in Project/Workspace File - By default FinalBuilder variables aren't expanded for the Project/Workspace files, turning this option on will expand variables.

Don't Recurse - Only valid for embedded projects, turning this on will build the specified project configurations without building any dependent projects

Apply to Project - you may choose to build all projects in the workspace, or build only selected projects

FinalBuilder can also update the version info in the project. It does this by searching the .rc files in the project for a VERSIONINFO structure. At runtime it will backup the original xxxx.rc file to xxxx.rc.orig and generate a new one with the updated version info. You can choose to apply the version info to all projects or selected projects in the workspace (the list of projects is dependant on what is selected on the project/workspace tab). To Auto Increment the build number, make sure the Auto-Increment check box is checked.



Scripting Info

The Action properties available are :

property ProjectFile : string;
property Configuration : string;
property UseEnv : boolean;
property BuildAction : TVC6BuildAction; Valid values are : baBuild,baRebuild and baClean.

The BuildType property determines whether All or selected projects in a workspace are built.

property BuildType : TVC6BuildType; Valid values are : btAllProjects and btSelectedProjects

The ConfigType property determines which configurations are built, Selected or All Configurations

property ConfigType : TVC6ConfigType; Valid values are : ctSelected,ctAll

property SelectedProjects : TStrings; //The selected projects that will be built.

property SelectedVI : TStrings; // the projects the version info will be applied to.

property VIType : TVC6VIType; Valid Values are : viAllProjects, viSelectedProjects //determines whether version info will be applied to all or selected projects in the workspace.

property InvokeMacro : string;

property IncludeVerInfo : boolean;

property AutoIncBuild : boolean;

property MajorVersion : integer;

property MinorVersion : integer;

property ReleaseVersion : integer;

property BuildVersion : integer;

property IsDebug : boolean;

property IsPreRelease : boolean;

property IsSpecial : boolean;

property IsPrivate : boolean;

property IsDLL : boolean;

property Locale : integer;

property CodePage : integer;

property VersionInfoKeys : WideString; // in the format :

CompanyName=VSoft Technologies Pty Ltd

FileDescription=FinalBuilder Core API

FileVersion=0.0.1.541

InternalName=FBCore10.bpl

LegalCopyright=© 2000 - 2003 VSoft Technologies Pty Ltd

LegalTrademarks=FinalBuilder™

OriginalFilename=FBCore10.bpl

ProductName=FinalBuilder™

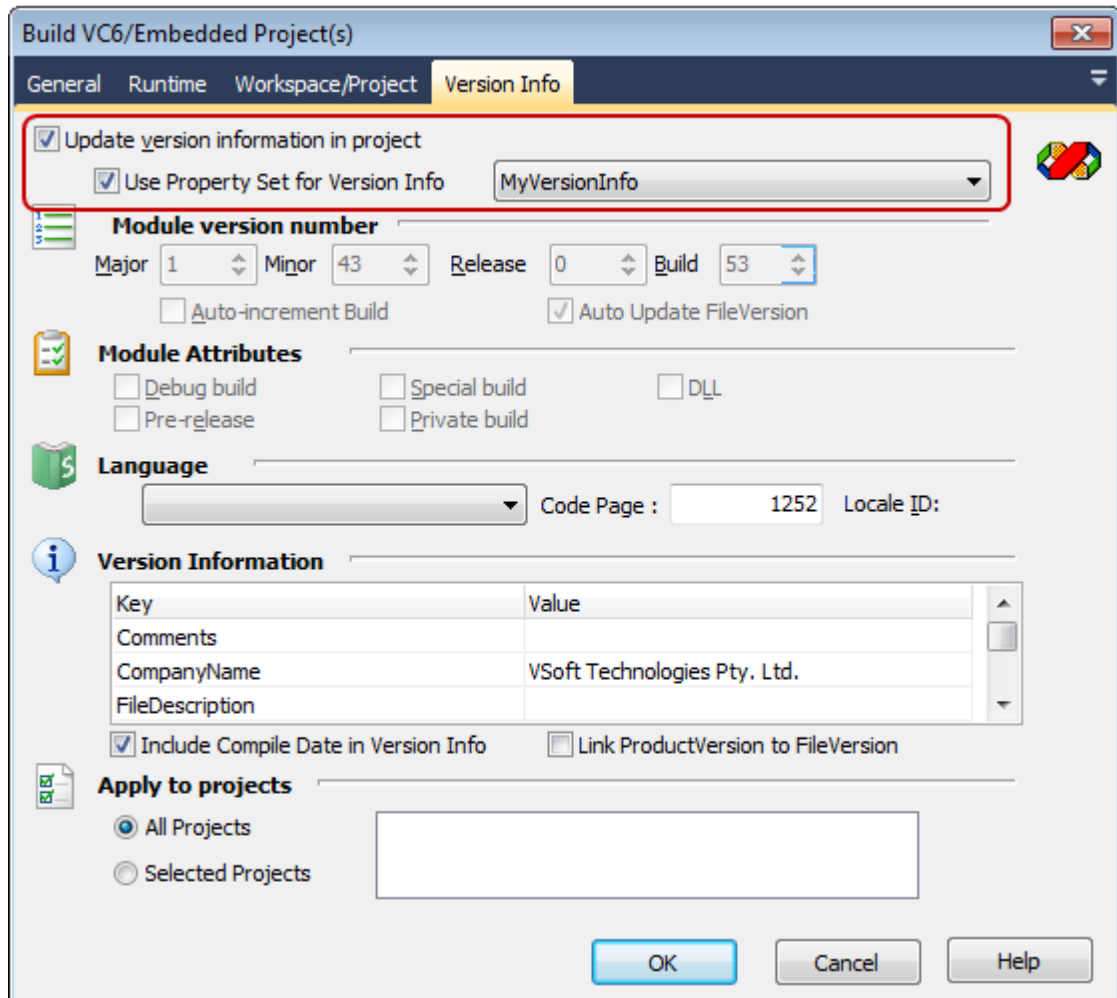
ProductVersion=1.0.0.0

Comments=

6.7.6.2.1 Using PropertySets with Visual C++

To use PropertySets with Visual C++, first you need to define a property set using the PropertySet define action and choose either Win32 Version Numbers or Win32 Version Info.

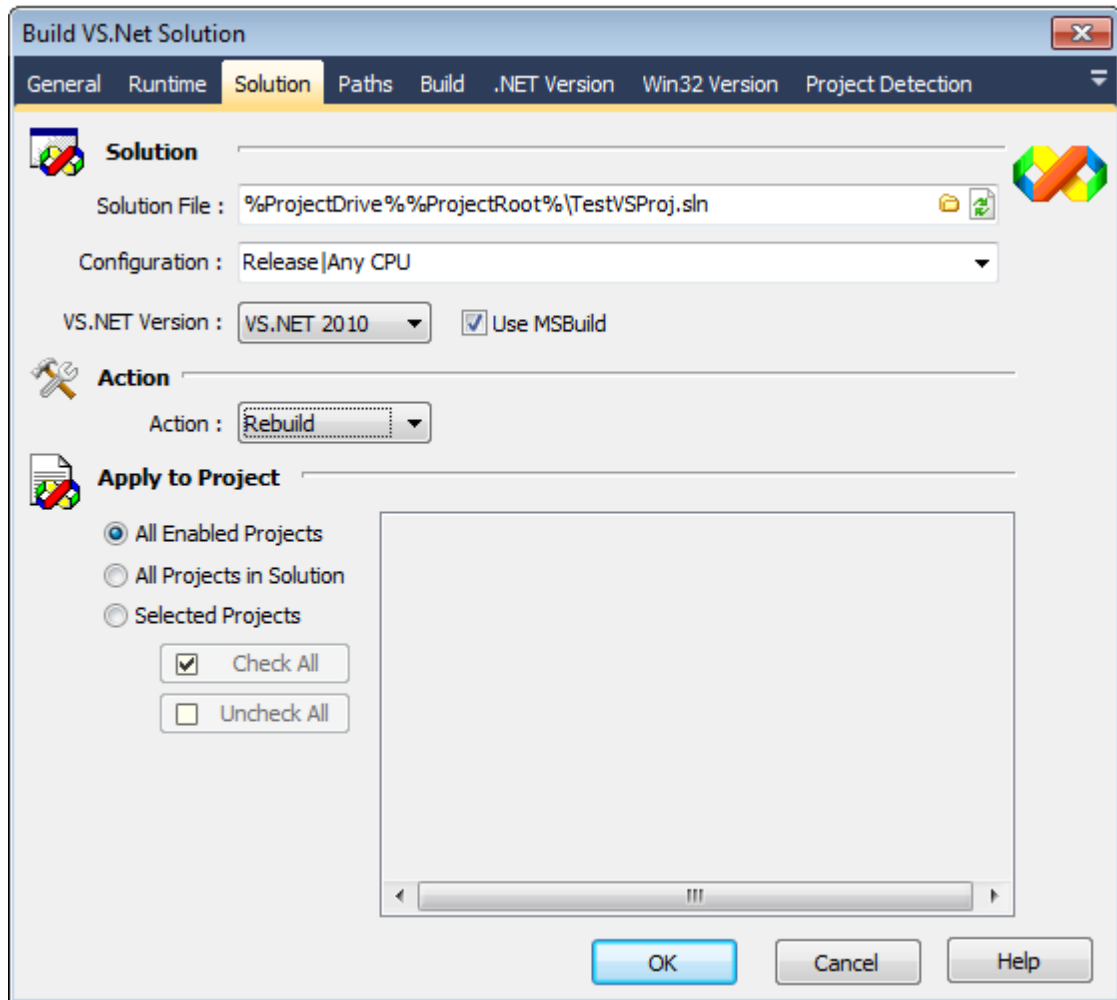
Then in the Build VC6/Embedded Projects action on the Version Info property page, select "Use property set for version info" and select the PropertySet to use:



When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

6.7.6.3 Visual Studio .NET Action

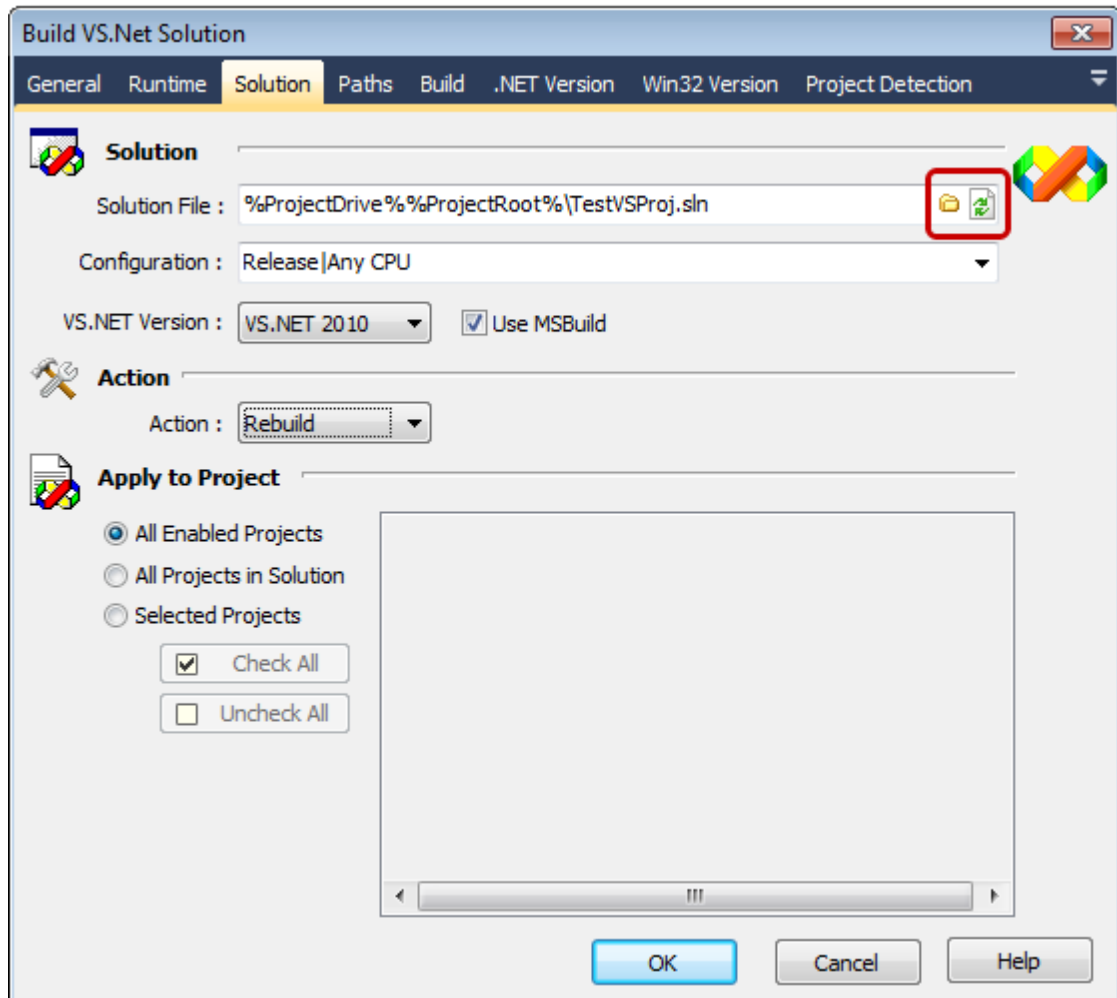
This action provides the ability to build Visual Studio .NET solutions using FinalBuilder. You can choose to build, rebuild or clean the whole solution or selected projects in the solution. You can also choose which Solution Configuration to use.



Solution File

The path to the solution file to build. If the solution file is found at design-time, the configuration and project lists will be automatically populated.

Click the refresh button (as shown) to reload the solution and refresh the configuration, version and project names.



Configuration

Choose the solution configuration to build. Available configurations are automatically loaded from the solution file and shown in the dropdown list.

If the solution file cannot be found, the dropdown shows the default Visual Studio.NET configurations.

VS.NET Version

Specify the version of Visual Studio to build the solution. A solution file can only be built by the version of Visual Studio that created it.

The correct version is automatically chosen in the solution file is loaded at design-time.

Use MSBuild

Solution files created by Visual Studio 2005 and newer can be built by MSBuild. Check the "Use MSBuild" checkbox to build with MSBuild instead of Visual Studio. If you

enable this option, you do not need Visual Studio installed on the build machine. Choosing this option also enables a lot of the custom properties of the action, and allows you to set custom MSBuild Properties as well.

Note that not all project types can be built with MSBuild. There are also differences involving Unloaded Projects (see below for details.)

Action

Choose the action (Build, Rebuild, Clean or Deploy) to perform. These are all the equivalent of taking the same action in the Visual Studio IDE.

Apply To Project

You can choose to build all projects in a solution, or only build certain projects.

All Enabled Projects

Selecting this option will build the chosen configuration, as defined in the Configuration Manager in Visual Studio. Only projects with the "Build" flag set in Configuration Manager will be built.

All Projects in Solution

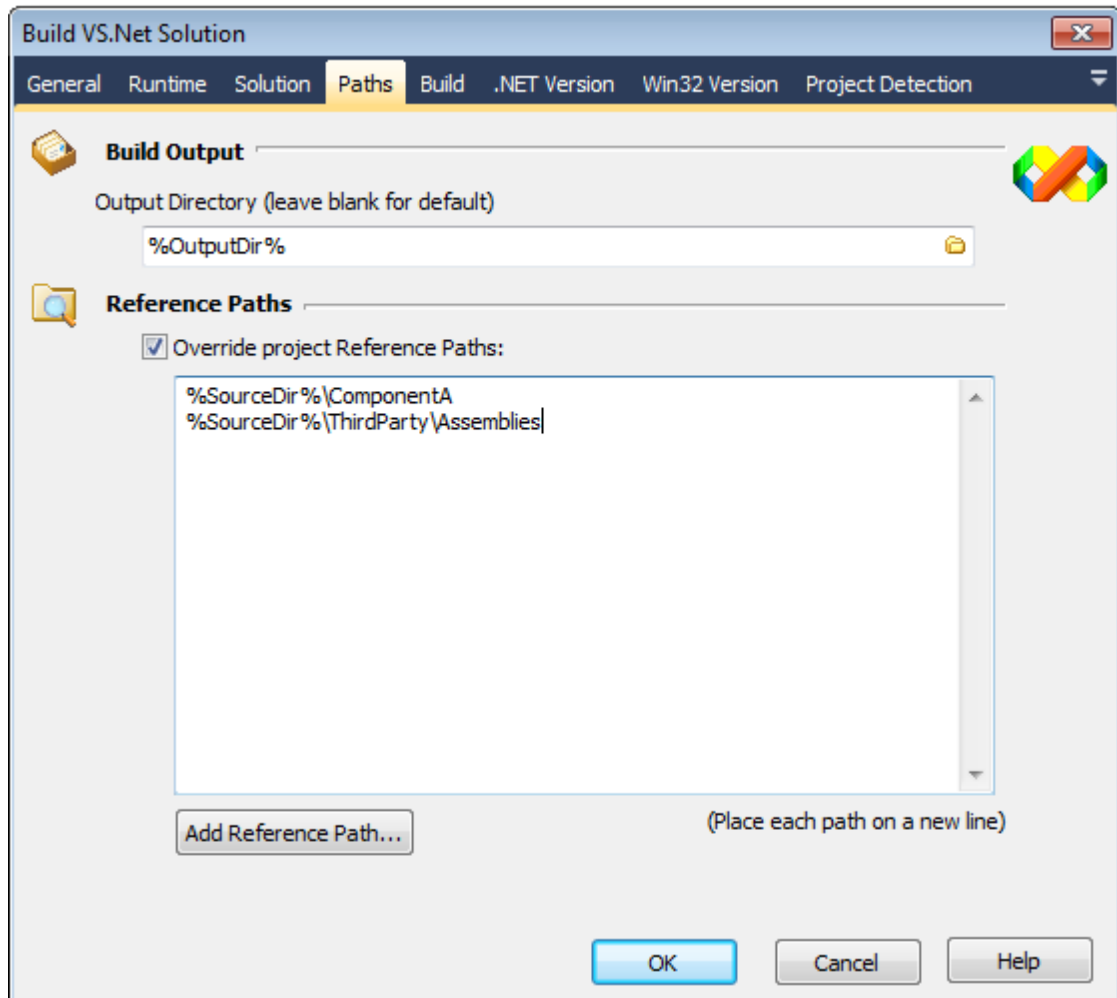
Selecting this option means that every single project in the solution will be built, regardless of whether or not it is set to "Build" as part of the selected configuration.

Selected Projects

Selecting this option allows you to choose which projects to build, using the listbox shown to the right of the option. Only checked projects will be built.

See below for some discussion on how to choose which projects to build dynamically at runtime (using script.)

The only time this behaviour changes is when there are Unloaded Projects (see below for details.)



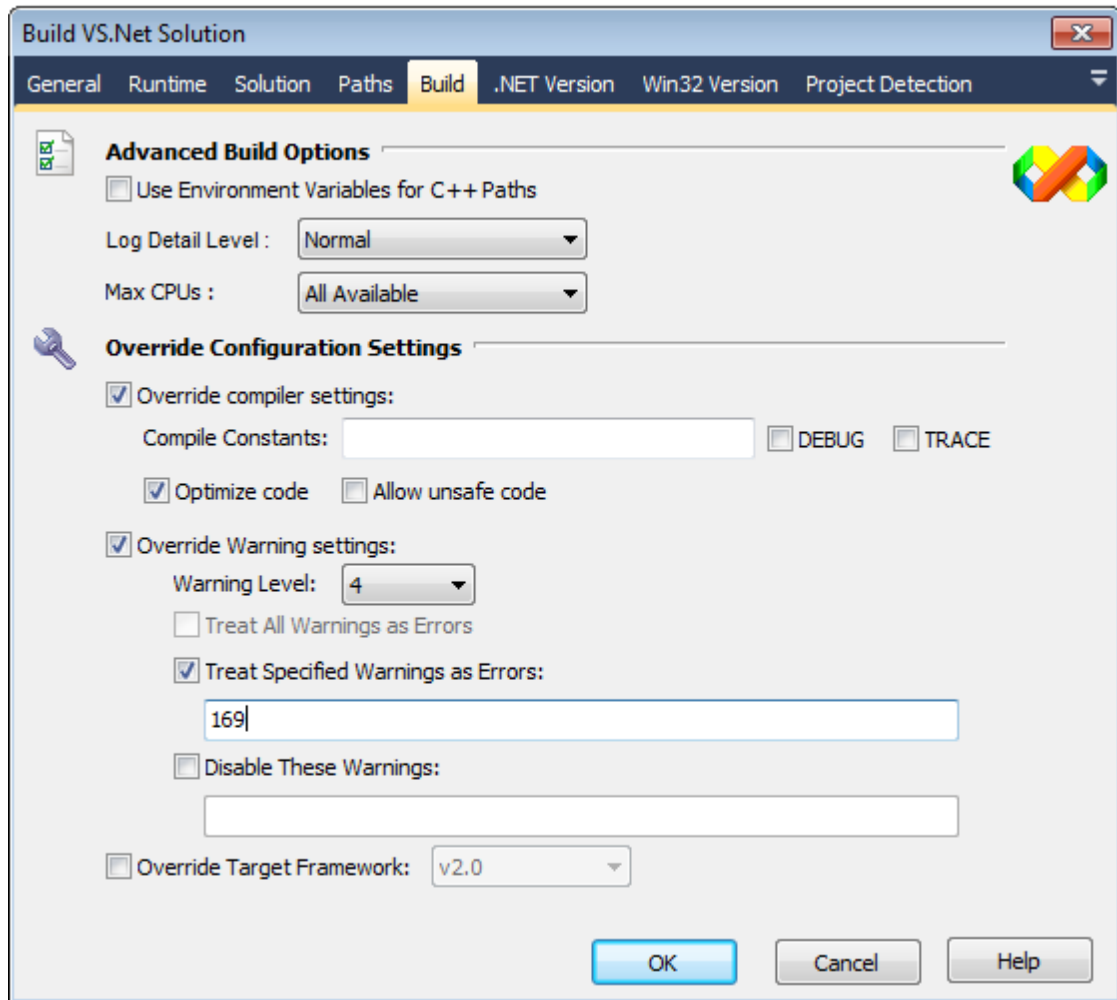
(The options on this page are only available when the "Use MSBuild" option is enabled.)

Build Output

Enter an output directory here to override the default output directory for each project in the solution. If this field is left blank, the default output directory will be used.

Reference Paths

If the "Override project Reference Paths" option is enabled, this list of paths will be used to resolve search for assembly references. This list overrides the list of paths set on the "Reference Paths" tab for a Visual Studio project (this list is normally stored in each project's *.user* file.)



(Apart from "Use Environment Variables...", all of the options on this page are only available when "Use MSBuild" is enabled.)

Use Environment Variables for C++ Paths

When building Visual C++ projects, enabling this option will use the environment variable values PATH, INCLUDE, LIBS, and LIBPATH when starting Visual Studio. This is equivalent to passing the /useenv option to Visual Studio, or the /p:useenv=1 argument to MSBuild.

Log Detail Level

Specify the level of output detail you would like to see in the log. Minimal will only show basic compilation steps. Normal shows all major targets. Detailed and Diagnostic can be used to analyse complex build scenarios.

Max CPUs

This option is only enabled when Visual Studio 2008 is being used for the build. You can limit the maximum number of CPUs to use for building, or set it to "All Available" to use all available CPUs.

Override compiler settings

If this checkbox is selected, then all of the following configuration properties will be overridden:

Compile Constants: Specify conditional constants to define. Separate multiple constants with semicolons.

DEBUG / TRACE: Check these boxes to specifically enable these two constants.

Optimize Code: Check this box to enable code optimization.

Allow unsafe code: Check this box to allow unsafe code.

Override Warning settings

If this checkbox is selected, then all of the following Warning-related properties will be overridden:

Warning Level: Specify the level of warning output to display.

Treat All Warnings as Errors: If this option is enabled, any compiler warning will cause the build to fail.

Treat Specified Warnings as Errors: Specify one or more warning numbers to treat as errors. Separate multiple warning numbers with semicolons.

Disable These Warnings: Specify one or more warning numbers to completely ignore. Separate multiple warning numbers with semicolons.

Override Target Framework

This option is only enabled when using Visual Studio 2008. It allows you to build projects against an earlier version of the .NET Framework.

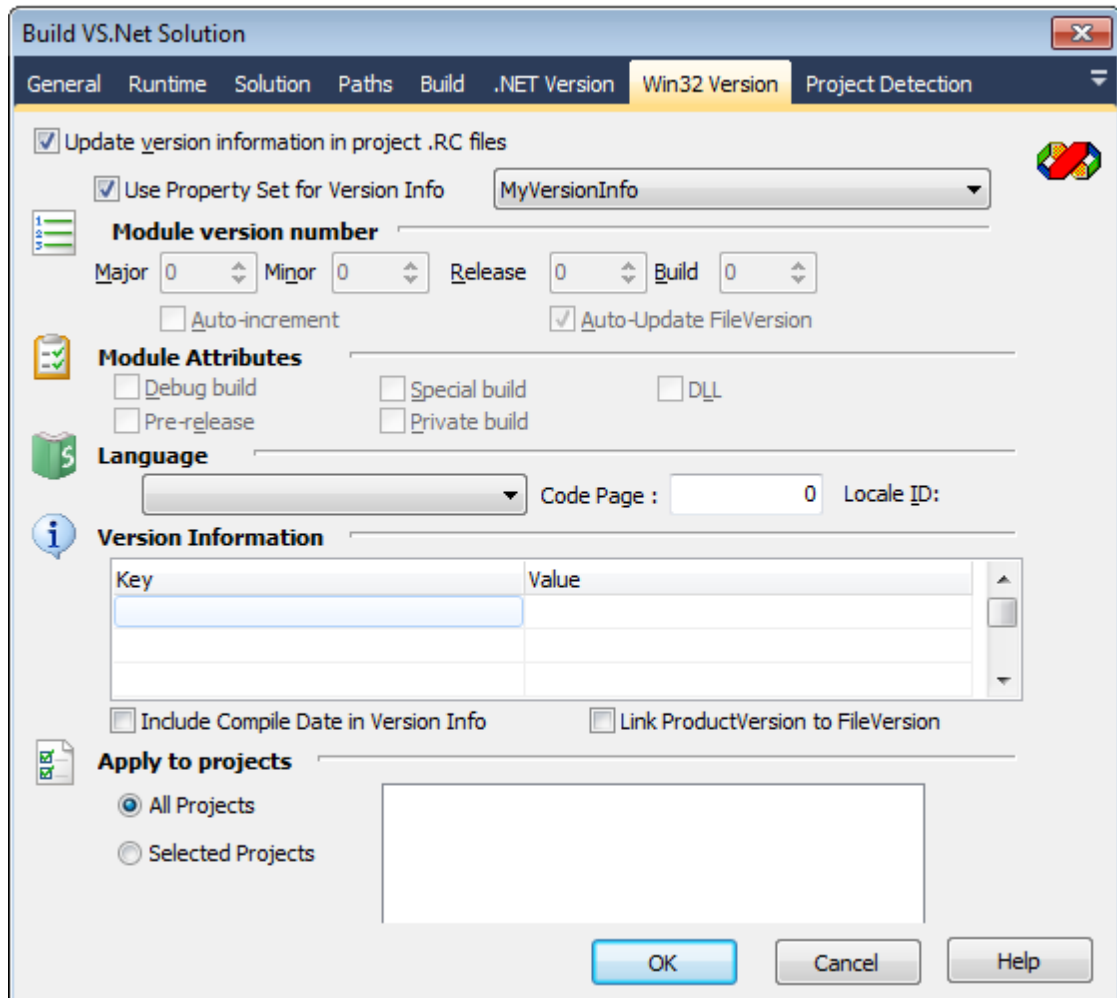
The screenshot shows the 'Build VS.Net Solution' dialog box with the '.NET Version' tab selected. The dialog has a menu bar with 'General', 'Runtime', 'Solution', 'Paths', 'Build', '.NET Version', 'Win32 Version', and 'Project Detection'. The '.NET Version' tab contains the following options:

- ☒ Update Assembly Info for .NET projects
 - ☐ Use Property Set for Version Info (dropdown menu)
- ☒ Title: Test Proj
- ☐ Description
- ☒ Company: VSoft Technologies Pty. Ltd.
- ☐ Product
- ☐ Copyright
- ☐ Trademark
- ☒ Assembly Version
 - Major: 1, Minor: 5, Build: 263, Revision: 0
 - ☐ Auto Increment Build, ☐ Auto Increment Revision
- ☐ File Version
 - ☐ Auto Increment Build, ☐ Auto Increment Revision
- ☒ Link File Version to Assembly Version
 - Apply to:
 - ☒ All Projects in Solution
 - ☐ Selected Projects

Buttons at the bottom: OK, Cancel, Help.

The .NET Version page allows the action to update AssemblyInfo files in all or selected .NET projects in the solution. Note that this will not work for any ASP.NET Web Sites in the solution.

Specify the details that you would like to set. The "Use PropertySet for Version Info" dropdown allows you to use a PropertySet to populate the version information (see dedicated topic.)



The Win32 Version page allows the action to update .RC files in all or selected C++ projects in the solution.

Specify the details that you would like to set. The "Use PropertySet for Version Info" dropdown allows you to use a PropertySet to populate the version information (see dedicated topic.)

Scripting

In addition to the common scripting properties, this action supports 3 additional script functions

Action.GetProjectID

This function takes the path to a project file, or the name of a project, and returns the internal ID (GUID) for that project. If the project is not found in the current solution, the function returns an empty string. Paths can be absolute, or relative to the solution file.

Action.GetProjectName

This function takes the ID of a project inside the solution, and returns the name (as shown in the Visual Studio IDE.) If the project ID is not found in the current solution, the function returns an empty string.

Action.GetProjectPath

This function takes the ID of a project inside the solution, and returns the project file path (relative to the solution directory.) If the project ID is not found in the current solution, the function returns an empty string.

Dynamically Choosing Which Projects to Build at Runtime

It is possible to dynamically configure the list of projects to build at runtime, using the Action.SelectedProjects property. This property is a TStrings property which contains the list of project IDs to build, if "Build Selected Projects" is enabled.

Another way to do this is to use Visual Studio .NET Solution Iterator.

Although the list normally contains Project IDs, it is possible to add project names and paths to the list, ie Action.SelectedProjects.Add("My Project Name").

Therefore, to set the list of projects to build from a CRLF-delimited string variable "MyProjects", which contains a list of project names or paths, it is possible to use script like this in the BeforeAction event of the action:

```
Action.SelectedProjects = FBVariables.MyProjects;
```

Note that the names are automatically converted to project IDs when the value is set. If the name does not represent a valid project, the entry is not added

Solutions Which Contain Unloaded Projects

It is possible to unload a project in Visual Studio by right-clicking in Solution Explorer and selecting "Unload Project". Unloaded projects build differently, depending on if you have chosen "Use MSBuild" or not.

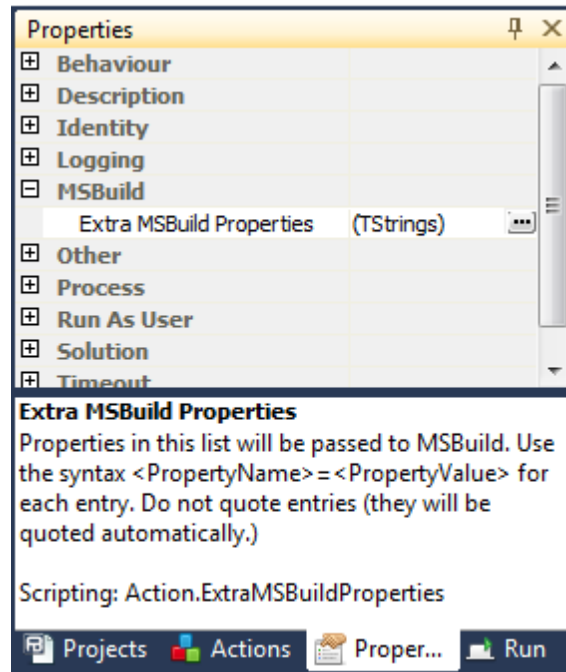
With "Use MSBuild" enabled, unloaded projects behave exactly the same as loaded projects. If you choose to build "All Projects", then all projects (including unloaded projects) will build. If you choose to build "Selected Projects", then all the selected projects will build.

Without "Use MSBuild" enabled, unloaded projects will never be built, even if they are explicitly selected.

Extra MSBuild Properties

It is possible to pass custom MSBuild Properties to the build process when "Use MSBuild" is selected. This allows you to set build properties which are not supported natively by the action.

To set these properties, select the action and click on the Properties tab. Under the MSBuild section is a property called "Extra MSBuild Properties":



Click the ellipsis button to edit the list of extra properties. Place each property on a new line. Properties take the form `<Name>=<Value>`.

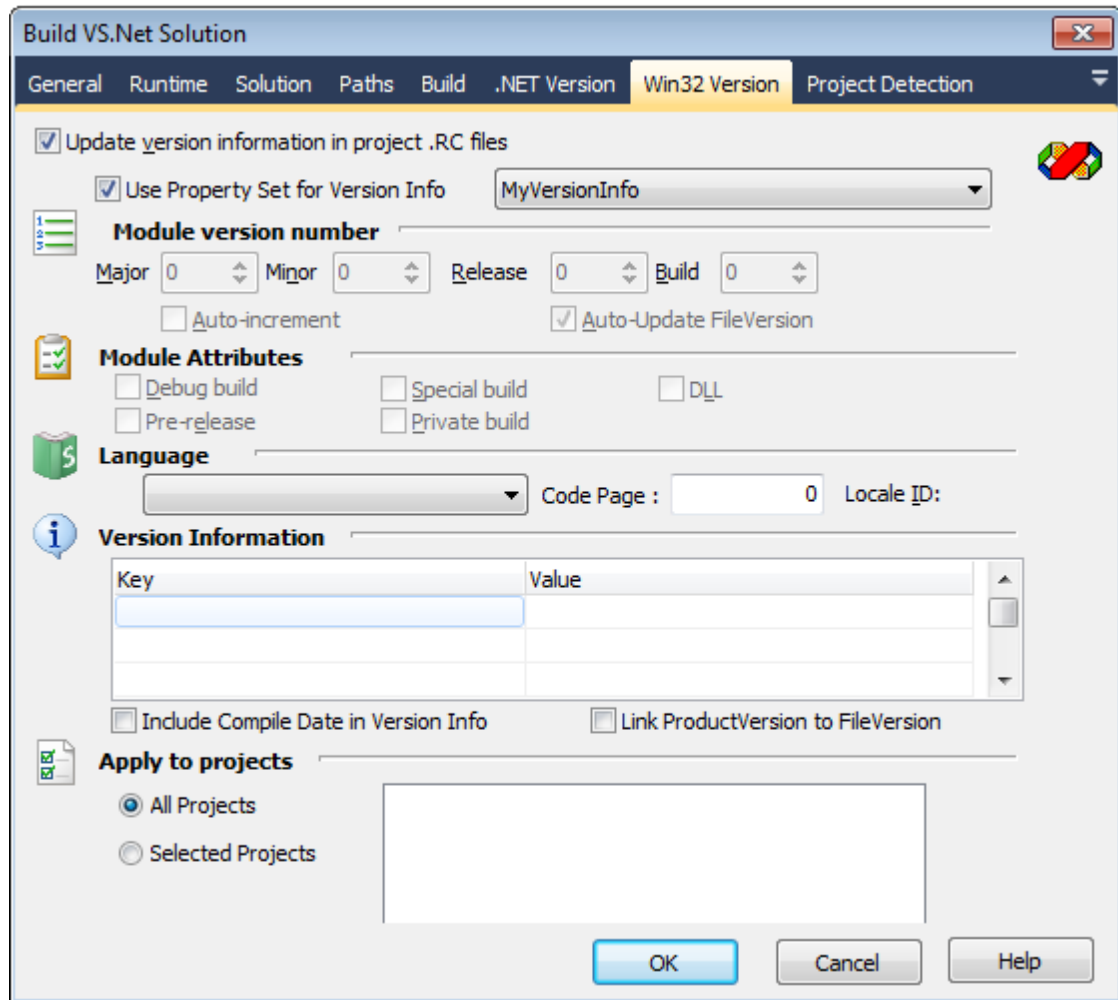
6.7.6.3.1 Using PropertySets with VisualStudio.Net

To use PropertySets with VisualStudio.Net, first you need to define a property set using the PropertySet define action and choose either Win32 Version Numbers or Win32 Version Info if you want to update the Version Info, and/or .Net Assembly Numbers or .Net Assembly Info if you want to update the Assembly info.

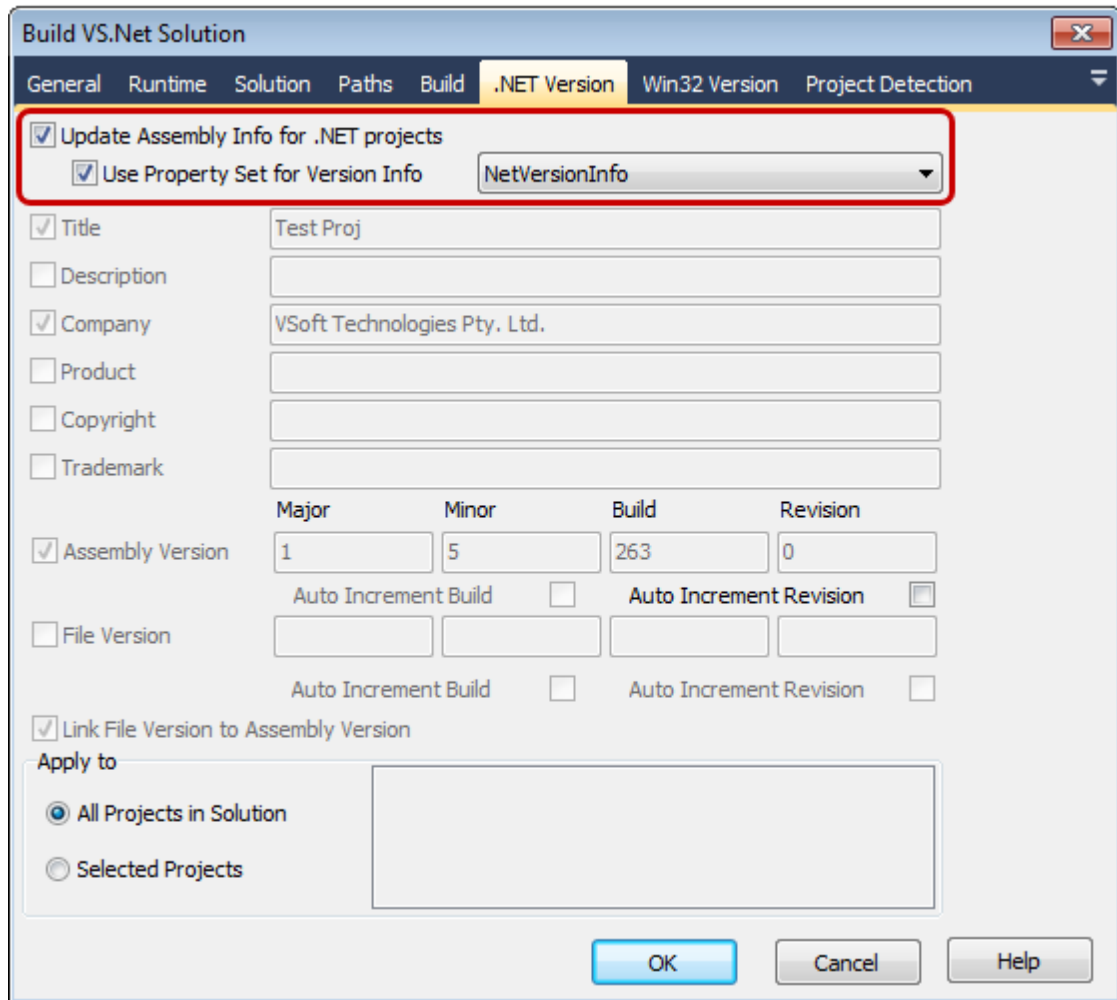
Then in the Build VS.Net Solution action on the Version Info or Assembly Info property page, select "Use property set for version info" and select the PropertySet to use.

When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

eg. For Version Info (must use a Win32 PropertySet)

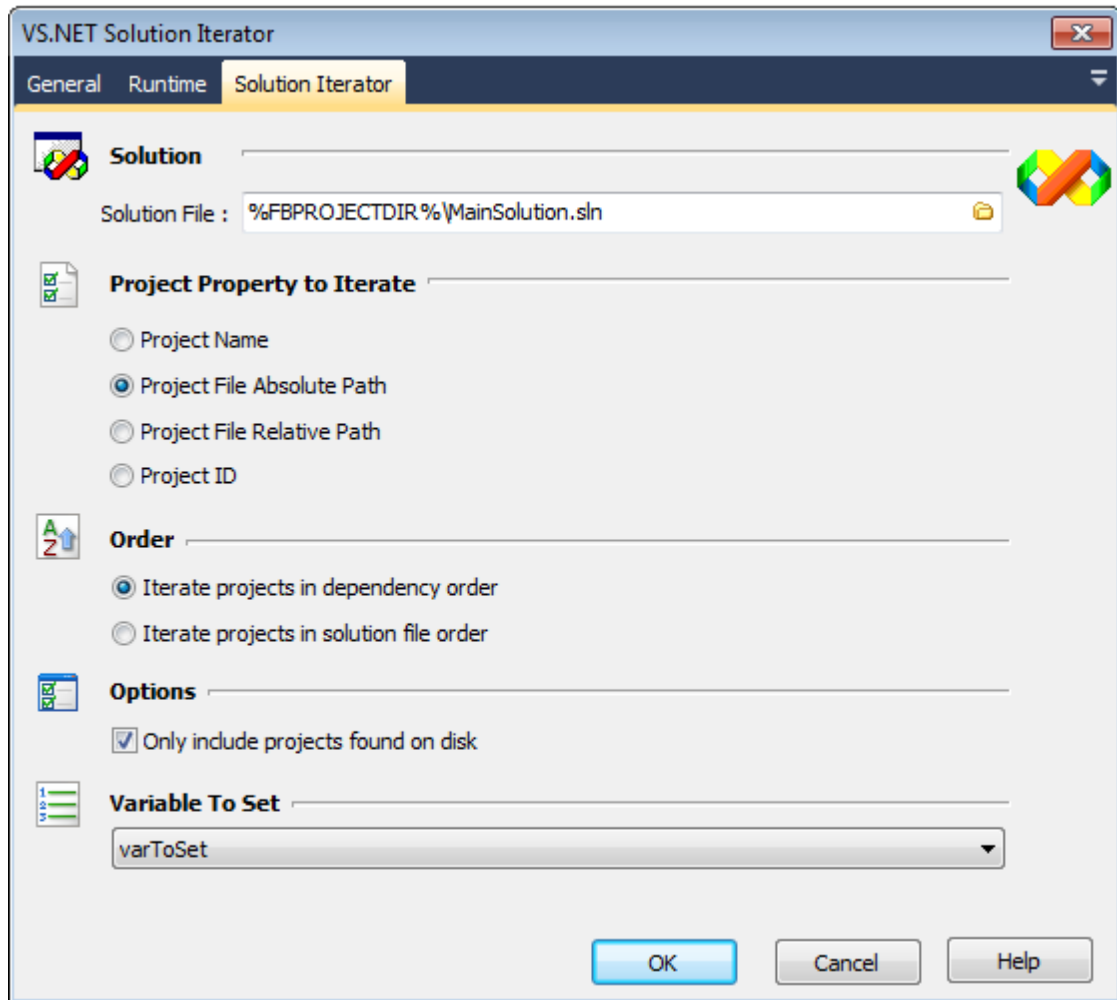


eg. For Version Info (must use a Dot Net PropertySet)



6.7.6.4 Visual Studio .NET Solution Iterator

The VS.NET Solution Iterator action allows you to iterate over each of the projects in a Visual Studio solution. For general information about iterators, see [here](#).



Solution

The path to the solution file to iterate over.

Project Property to Iterate

You can set which property of the project is used as the value for the iterator.

Project Name - Use the name of each project, as shown in Visual Studio.

Project File Absolute Path - Use the full path to each project. If this option is used, the project file paths can be used with the MSBuild Project action.

Project File Relative Path - Use the path to each project, relative to the solution directory.

Project ID - Use the ID of each project (this is a GUID value, used internally by Visual Studio.)

Order

By default, projects are sorted in dependency order (so that they can be successfully built as they are iterated.) If necessary, projects can be iterated in the order they appear in the solution file instead.

Options - Only include projects found on disk

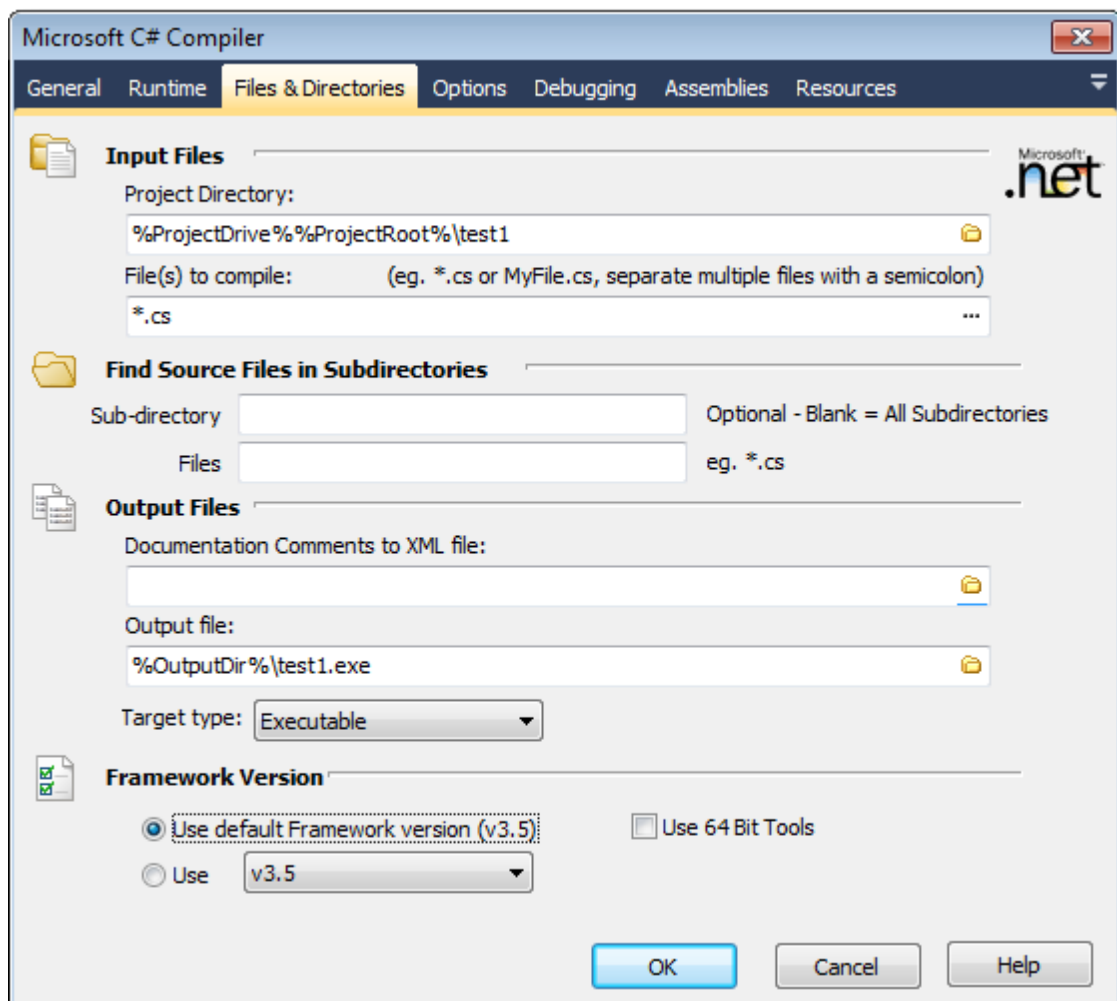
By default, if a project file is not found on disk then it is excluded from the iterator. Disable this option if you would like to iterate project files which are not found, as well.

Variable to Set

Choose a FinalBuilder variable to set for each iteration.

6.7.6.5 Microsoft C# Compiler Action

This action executes the Microsoft .NET Framework C# command line compiler.



Information on the compiler options can be found in the .NET Framework documentation on MSDN :

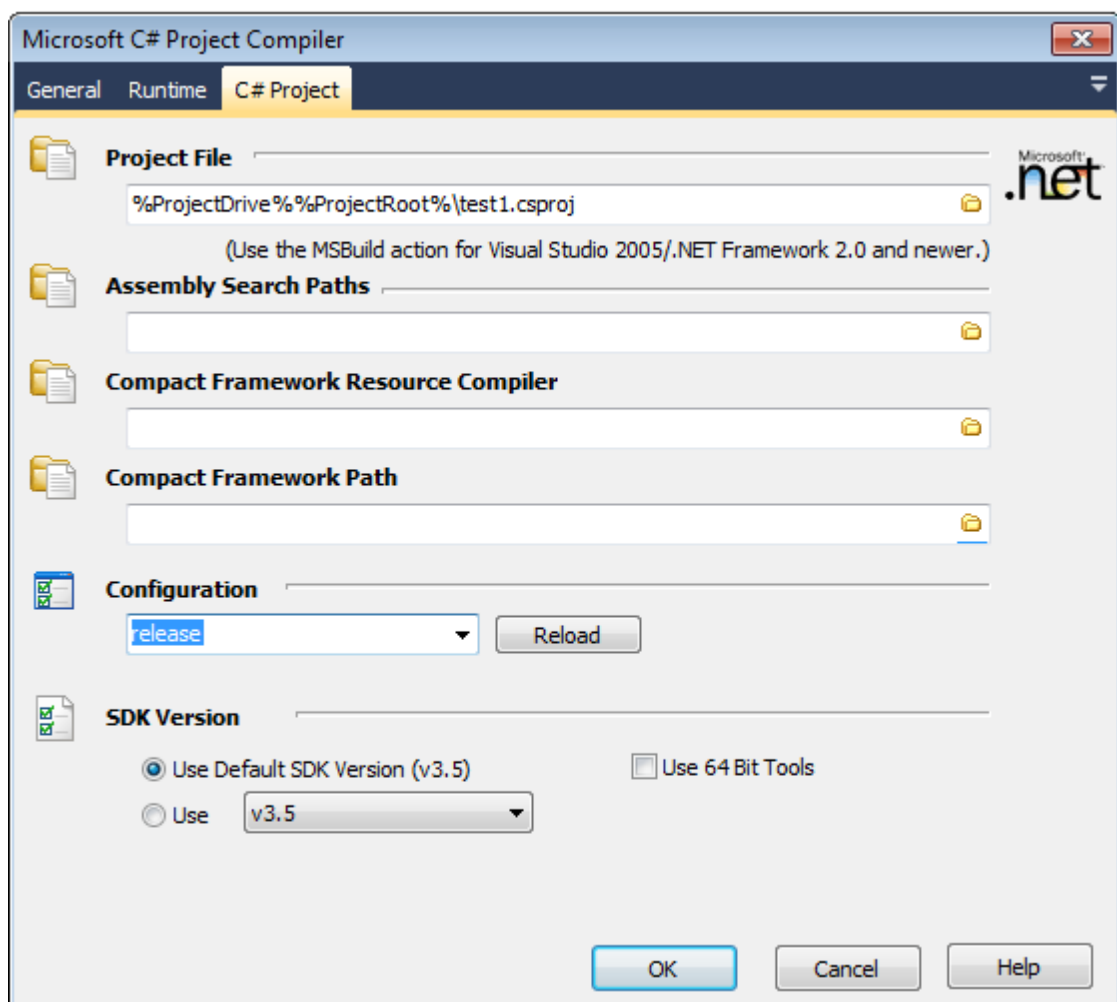
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cscomp/html/vcrefCompilerOptions.asp>

6.7.6.6 Microsoft C# Project Compiler Action

This action enables you to compile Microsoft C# Projects (.csproj) produced by Visual Studio.NET without having Visual Studio installed on your build machine.

Note that this action is only for Visual Studio.NET 2003 and earlier, and/or .NET Framework 1.1 and earlier.

To build C# projects from newer versions of Visual Studio / .NET Framework, use the MSBuild Project action.



Project File

Select the .csproj file to build.

Assembly Search Paths

Specify optional additional search paths for .NET assemblies. Click the folder button to display a list dialog with all search paths.

Compact Framework Resource Compiler / Path

Only required if building with the .NET Compact Framework. Select a resource compiler and framework path for the required version of .NET CF.

Configuration

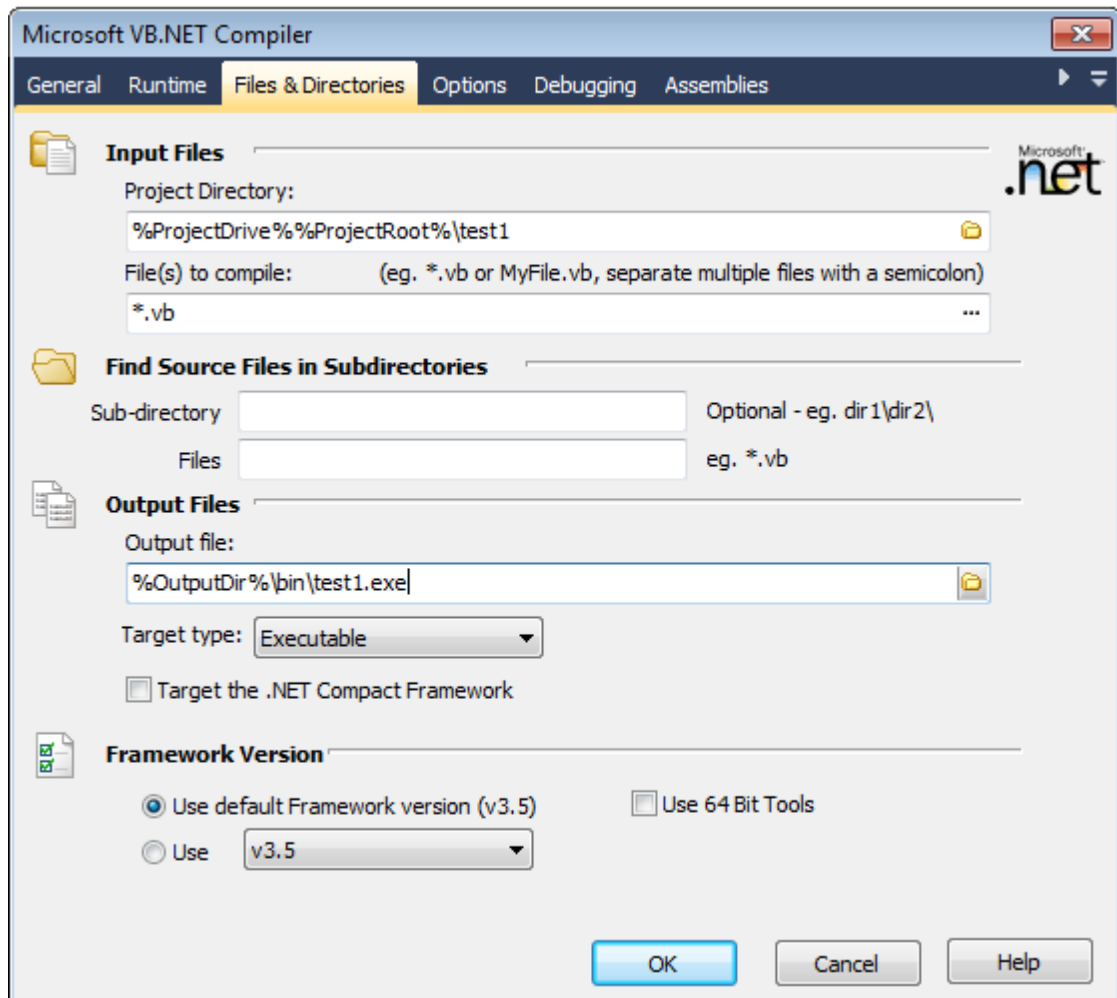
Choose the Configuration to build (ie Debug/Release.) Click the Reload button to reload available configurations from the project file.

SDK Version

Choose the SDK Version to build. Supported SDKs are 1.0 and 1.1. For newer Framework versions, use the **MSBuild Project action**.

6.7.6.7 Microsoft VB.NET Compiler Action

This action executes the Microsoft .NET Framework VB.NET command line compiler.



Information on the compiler options can be found in the .NET Framework documentation on MSDN :

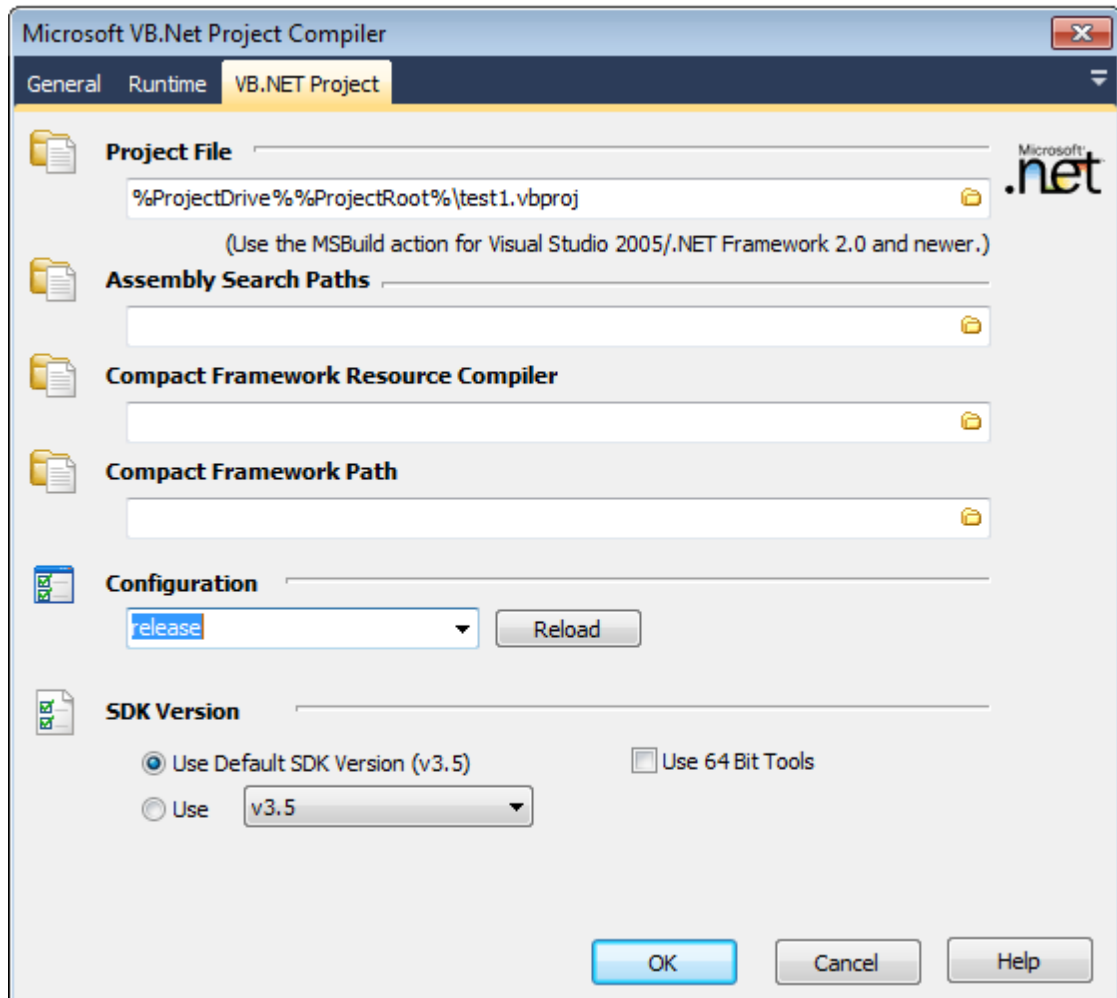
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbclr7/html/vaconbuildingfromcommandline.asp>

6.7.6.8 Microsoft VB.NET Project Compiler Action

This action enables you to compile Microsoft VB.NET Projects (.vbproj) produced by Visual Studio.NET without having Visual Studio installed on your build machine.

Note that this action is only for Visual Studio.NET 2003 and earlier, and/or .NET Framework 1.1 and earlier.

To build VB.NET projects from newer versions of Visual Studio / .NET Framework, use the MSBuild Project action.



Project File

Select the .vbproj file to build.

Assembly Search Paths

Specify optional additional search paths for .NET assemblies. Click the folder button to display a list dialog with all search paths.

Compact Framework Resource Compiler / Path

Only required if building with the .NET Compact Framework. Select a resource compiler and framework path for the required version of .NET CF.

Configuration

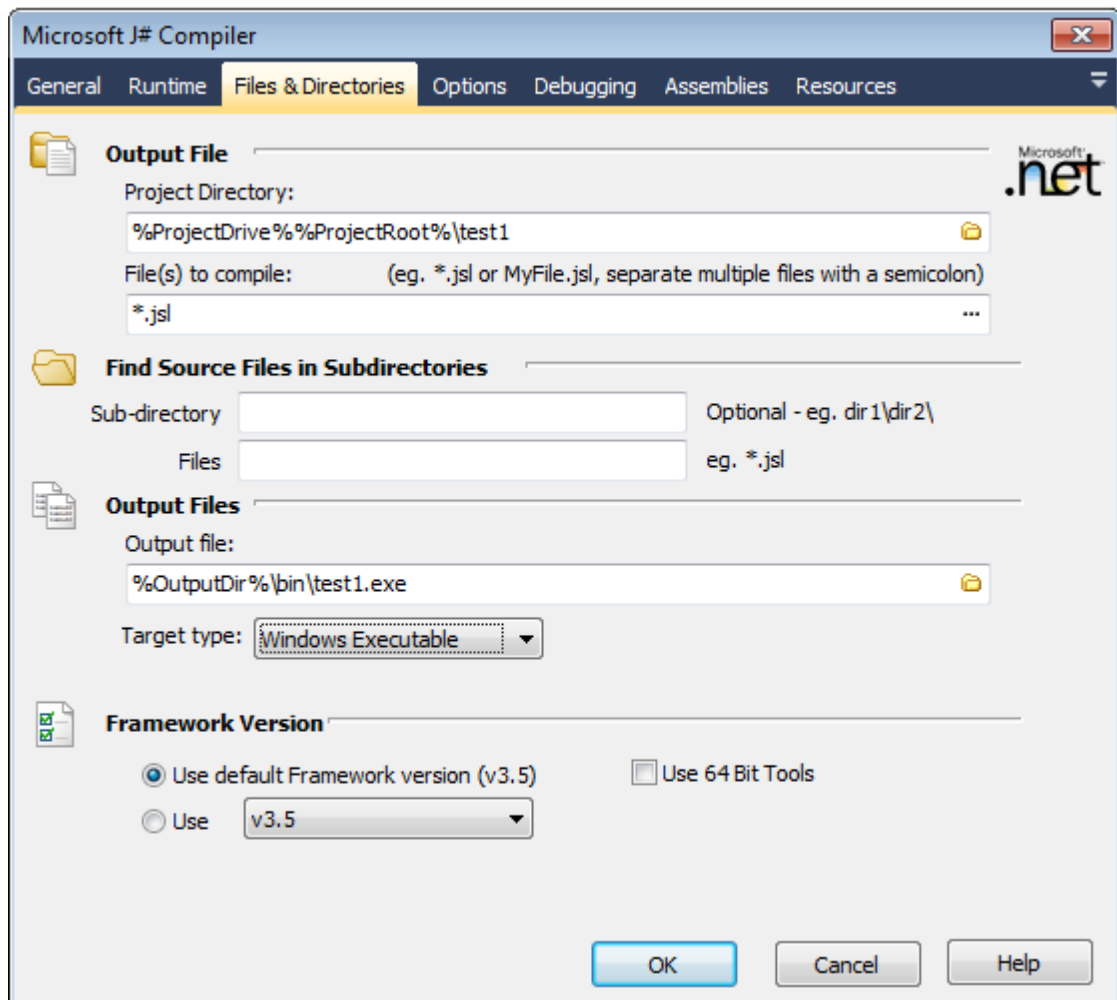
Choose the Configuration to build (ie Debug/Release.) Click the Reload button to reload available configurations from the project file.

SDK Version

Choose the SDK Version to build. Supported SDKs are 1.0 and 1.1. For newer Framework versions, use the **MSBuild Project action**.

6.7.6.9 Microsoft J# Compiler Action

This action executes the Microsoft .NET Framework J# command line compiler.

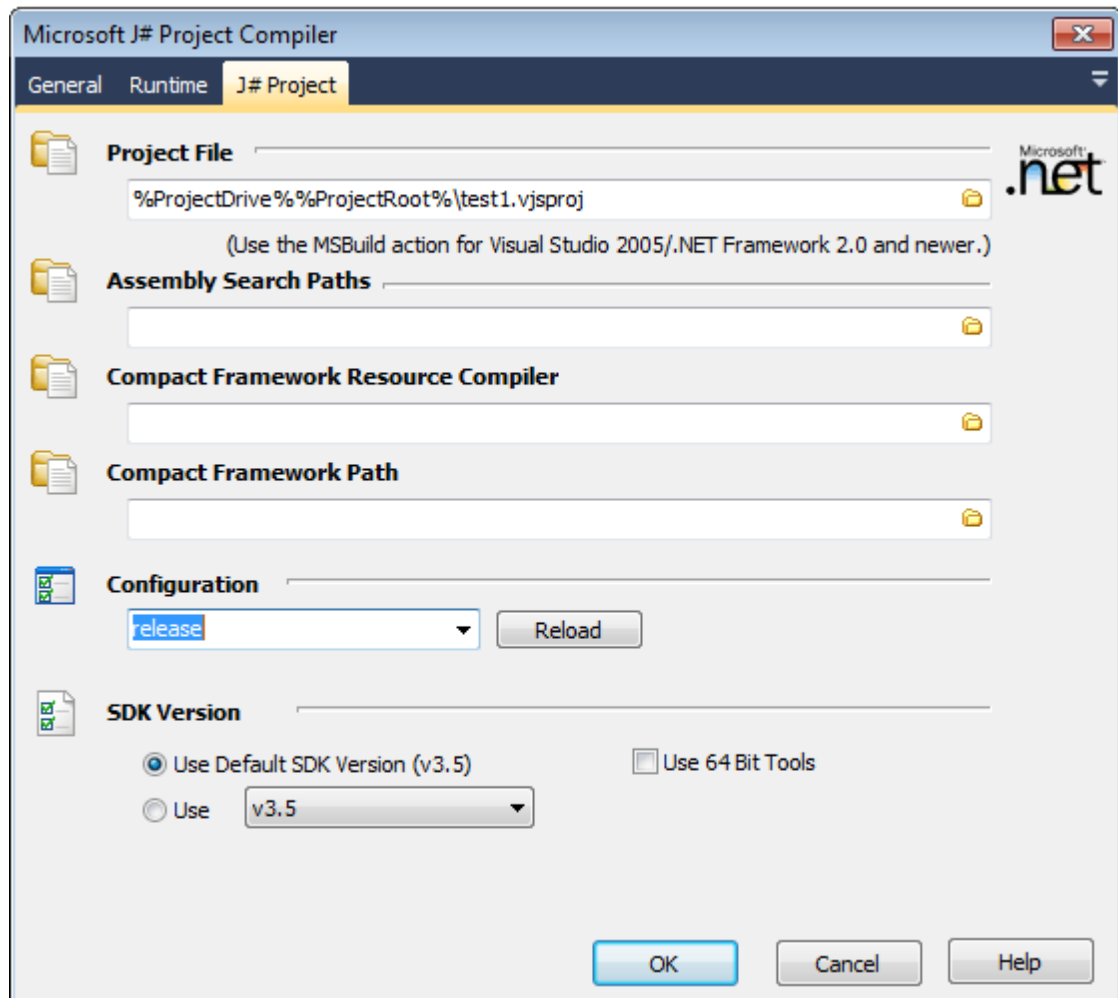


Information on the compiler options can be found in the .NET Framework documentation on MSDN :

http://msdn.microsoft.com/library/en-us/dv_vjsharp/html/vjgrfVisualJCompilerOptions.asp

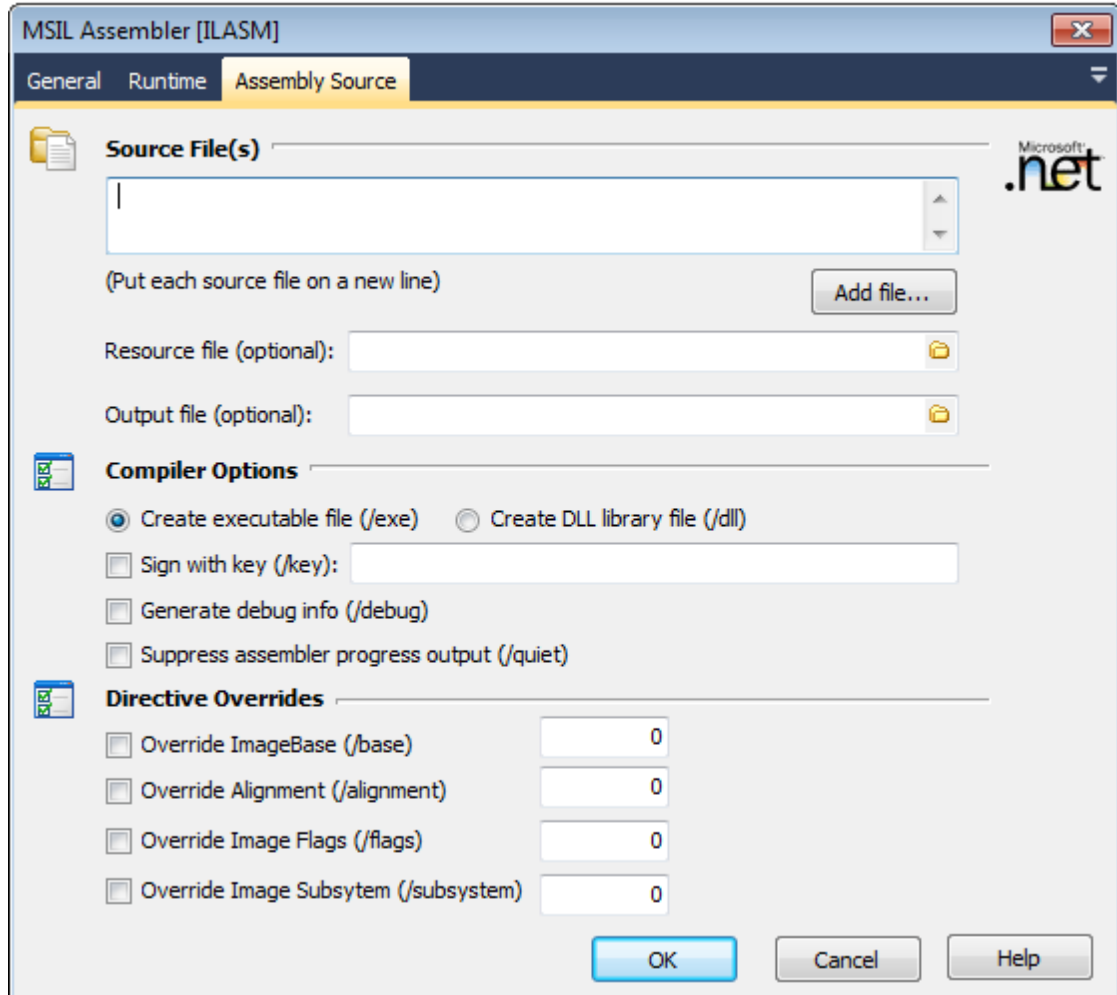
6.7.6.10 Microsoft J# Project Compiler Action

This action enabled you to compile Microsoft J# Projects (.vjsproj) produced by Visual Studio.NET without having Visual Studio installed on your build machine.



6.7.7 MSIL Assembler (ILASM) action

The MSIL Assembler Action allows you to automate running of the `ilasm.exe` .IL assembler.



Source Files(s)

Type the pathnames of one or more .IL files to assemble (place each file on a new line.) Click the "Add file..." button to add a new source file.

Resource File (optional)

Optionally specify a .res file to include in the output file.

Output file (optional)

Optionally specify the output file path. If no path is specified, `ilasm` will output `<sourcename>.exe` or `<sourcename>.dll`, depending on the target type.

Create executable file (/exe) | Create DLL library file (/dll)

Specify whether to output an executable or a DLL file.

Sign with key (/key)

Specify whether to sign the compiled assembly with a key. Keys can be specified as key files (type the pathname to the key file) or as Key Sources (use the syntax @KeySourceName.)

Generate debug info (/debug)

If this option is selected, a .pdb file will be generated allowing you to debug the built assembly.

Suppress assembler progress output (/quiet)

Suppresses the progress details which are normally output by the assembler.

Directive Overrides

Each of these overrides corresponds to a directive which can be given in the .IL files. If an override value is specified here, it replaces the value(s) given in the source file (s).

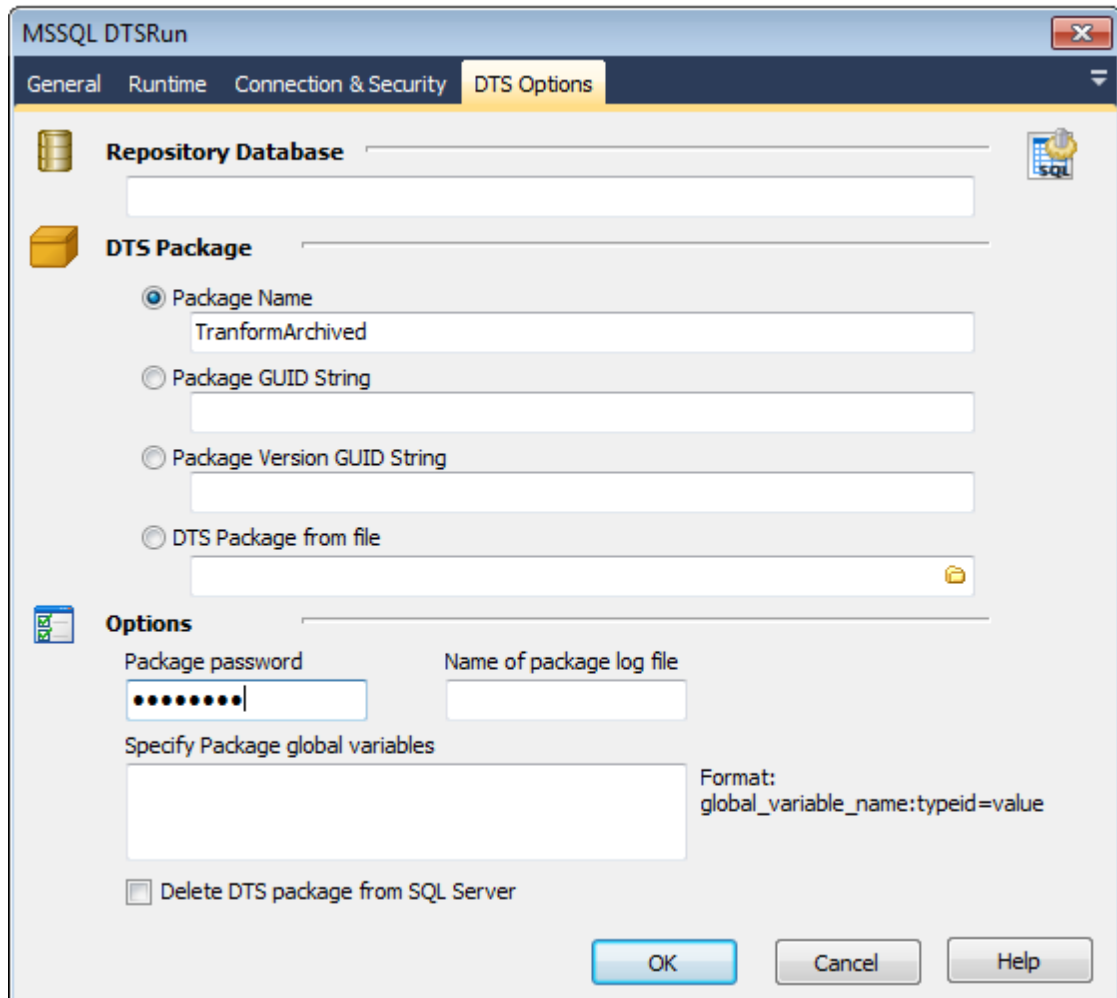
6.8 Database

6.8.1 SQL Server Actions

6.8.1.1 DTSRun Action

[FinalBuilder 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.



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

6.8.1.2 Execute SQL Action

[FinalBuilder Professional Edition]

Execute an SQL statement against Microsoft SQL Server and capture the result set.

Connection & Security

The screenshot shows the 'MSSQL Execute SQL' dialog box with the 'Connection & Security' tab selected. The dialog has four tabs: General, Runtime, Connection & Security, and Input. The 'Connection & Security' tab contains three sections: 'SQL Server', 'SQL Server Tool', and 'Security'. In the 'SQL Server' section, the 'Server' field is set to 'scarecrow' and the 'Use database' field is set to 'northwind'. A 'View List' button is next to the 'Server' field. In the 'SQL Server Tool' section, three radio buttons are present: 'ISQL (uses DB-Library to communicate with Microsoft® SQL Server)', 'OSQL (uses ODBC to communicate with the server)' (which is selected), and 'SQLCmd (uses SQL Native Client OLE DB to communicate with a SQL 2005 Server)'. In the 'Security' section, the 'Use integrated security' checkbox is checked. Below it are fields for 'Username' and 'Password', and a 'Login Timeout' spinner set to 0. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

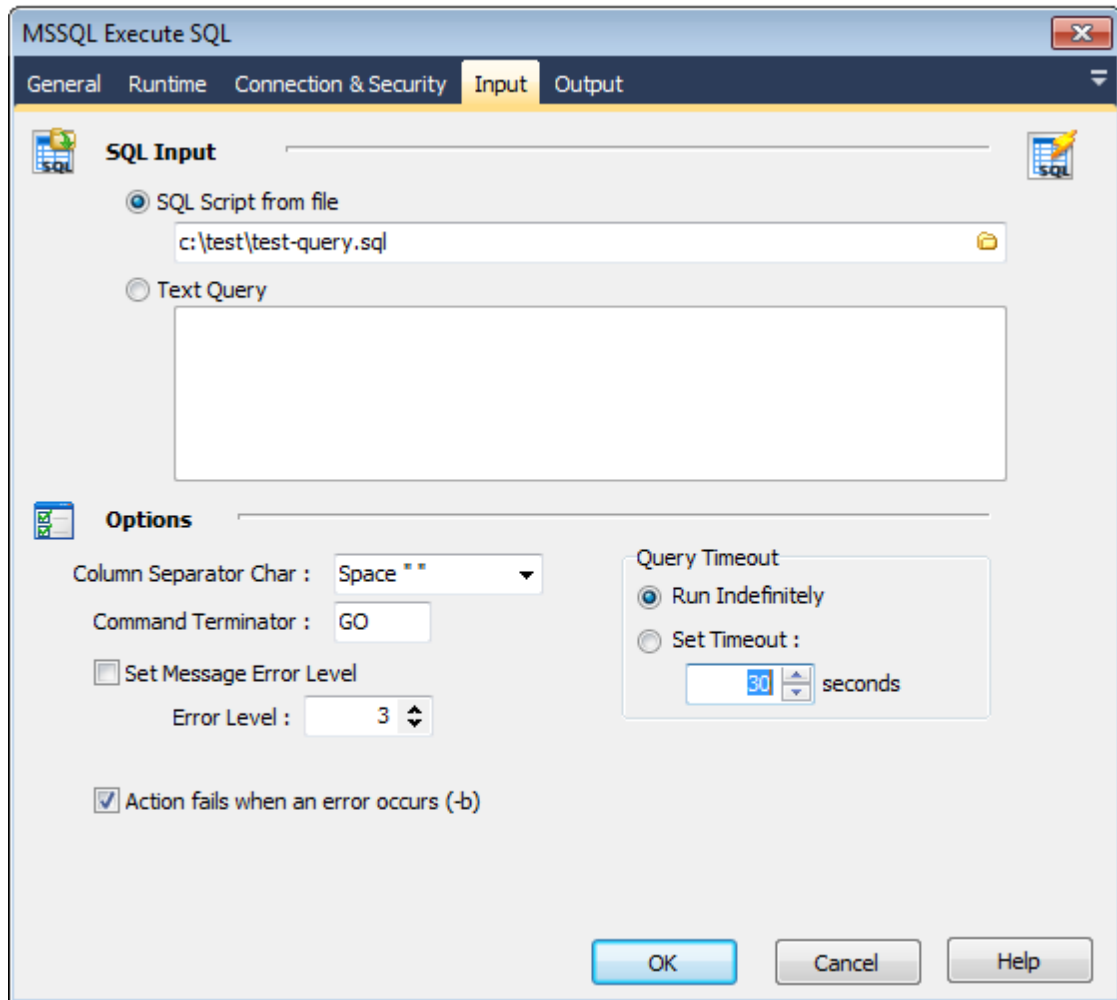
Server - Specify the MS SQL Server to use. Clicking "View List" will attempt to locate any SQL Servers on the network.

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 - specify a file which contains an SQL Script.
- Text Query - Specify the SQL Query to execute.

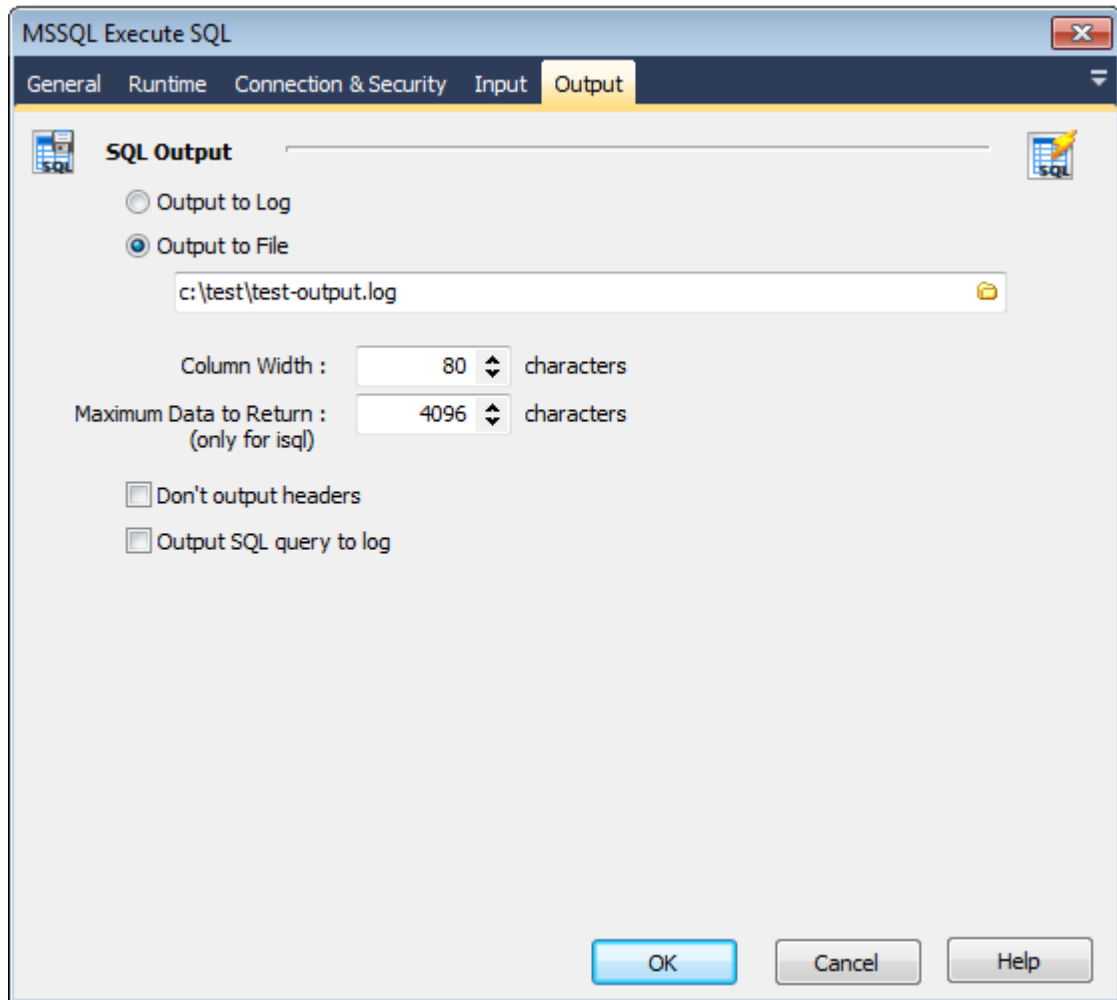
For information on the other options see:

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



The query results can either be output to a file or to the FinalBuilder Log.
For information on the other options see:

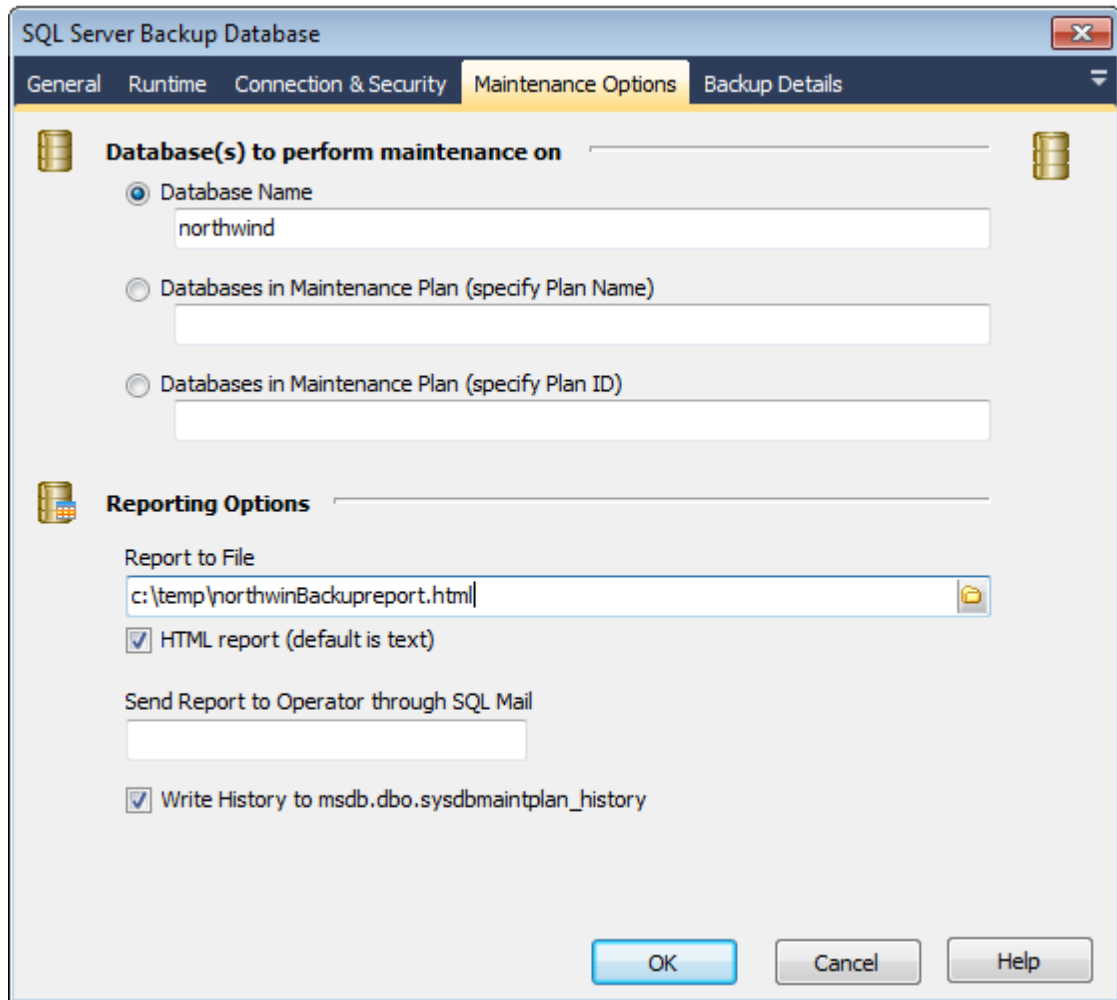
isql : http://msdn.microsoft.com/library/en-us/coprompt/cp_isql_8r39.asp
osql : http://msdn.microsoft.com/library/en-us/coprompt/cp_osql_1wxi.asp
sqlcmd : <http://msdn2.microsoft.com/en-us/library/ms162773.aspx>

To set the location of ISQL, OSQL and SQLCmd see SQL Server Options

6.8.1.3 SQL Server Backup Database

[FinalBuilder 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



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

6.8.1.4 SQL Server Best Practices Analyzer

[FinalBuilder 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 FinalBuilder.

Specify the SQL Server Best Practices Analyzer installation directory from the SQL Server tab in the Options.

The screenshot shows the 'SQL Best Practices Analyzer' dialog box with the 'Best Practices Analyzer' tab selected. The dialog is divided into three main sections: 'BPA Repository Location', 'Best Practices Groups to execute', and 'Options'. In the 'BPA Repository Location' section, the 'SQL Server Instance' is set to 'localhost' and the 'BPA Repository Database' is 'sqlbpa'. The 'Use integrated authentication' radio button is selected. In the 'Best Practices Groups to execute' section, a list box contains 'Security Policy' and 'Miscellaneous Practices'. In the 'Options' section, the checkbox 'Action fails if any group fails to achieve' is checked, and the dropdown menu is set to 'partial compliance'. The 'Log SQL BPA operations to msbpa.log' checkbox is unchecked. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

SQL Best Practices Analyzer

General Runtime **Best Practices Analyzer**

BPA Repository Location

SQL Server Instance: localhost

BPA Repository Database: sqlbpa

☒ Use integrated authentication

☐ Use username:

Password:

Best Practices Groups to execute

Security Policy
Miscellaneous Practices

(specify each Best Practices Group on a new line)

Options

☒ Action fails if any group fails to achieve partial compliance

☐ Log SQL BPA operations to msbpa.log

OK Cancel Help

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

6.8.1.5 SQL Server Check Catalogue

[FinalBuilder 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.

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

6.8.1.6 SQL Server Check Database

[FinalBuilder 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.

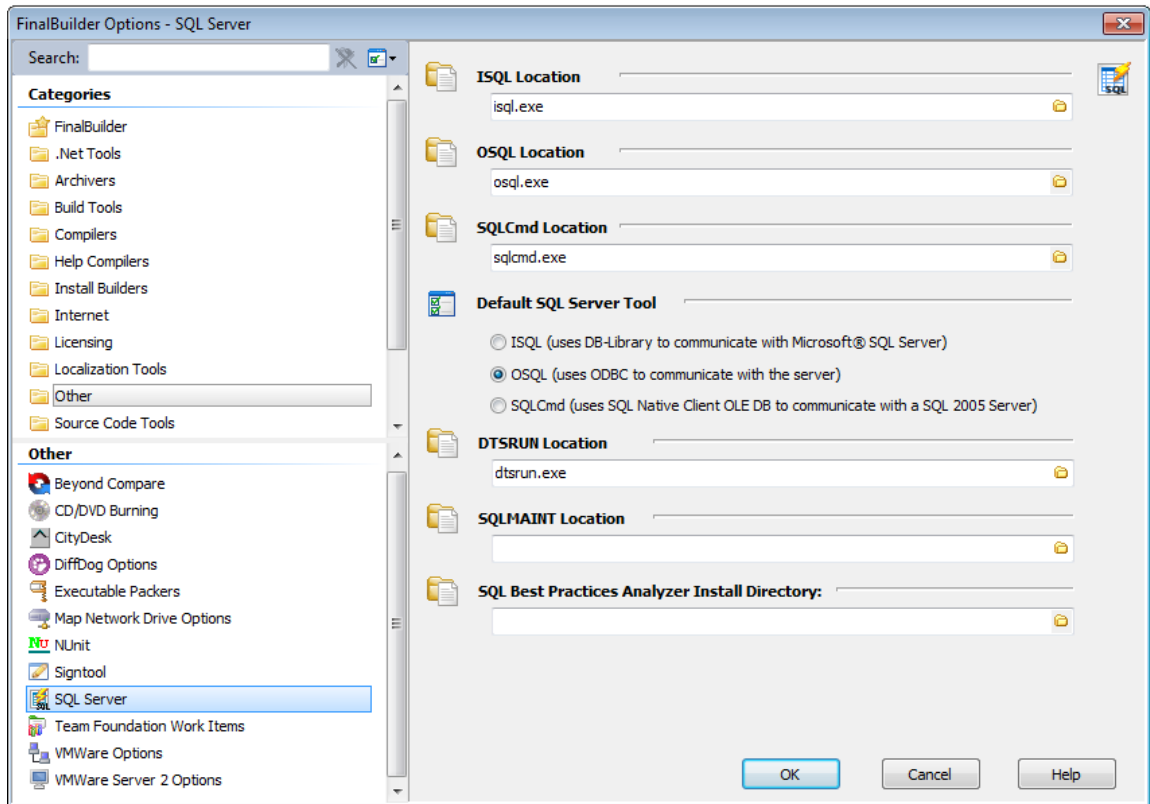
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.8.1.7 SQL Server Options

[FinalBuilder 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



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

6.8.1.8 SQL Server Rebuild Indexes

[FinalBuilder 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.

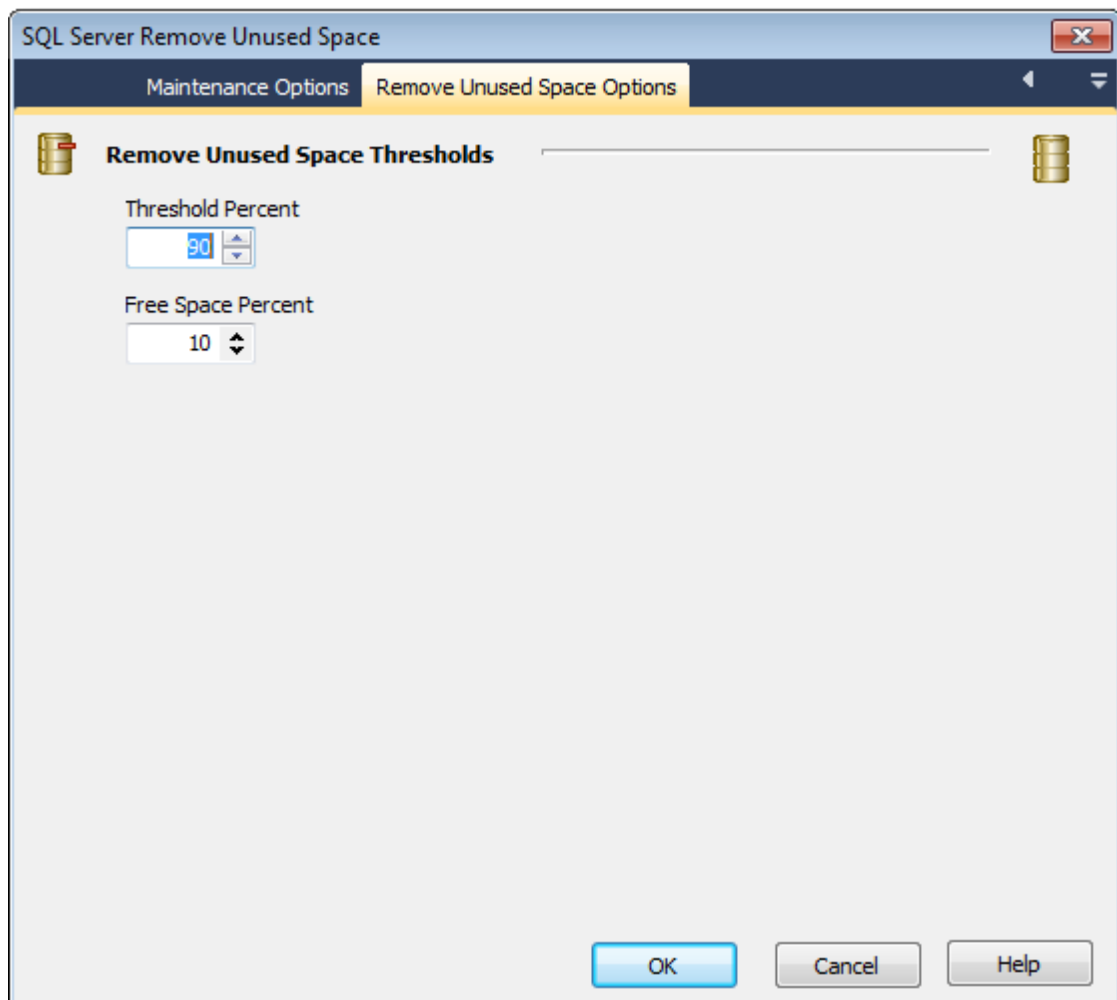
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.8.1.9 SQL Server Remove Unused Space

[FinalBuilder 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.



Specify the threshold percent and free space percent (see above).

For more information on the sqlmaint tool see: <http://msdn.microsoft.com/library/en->

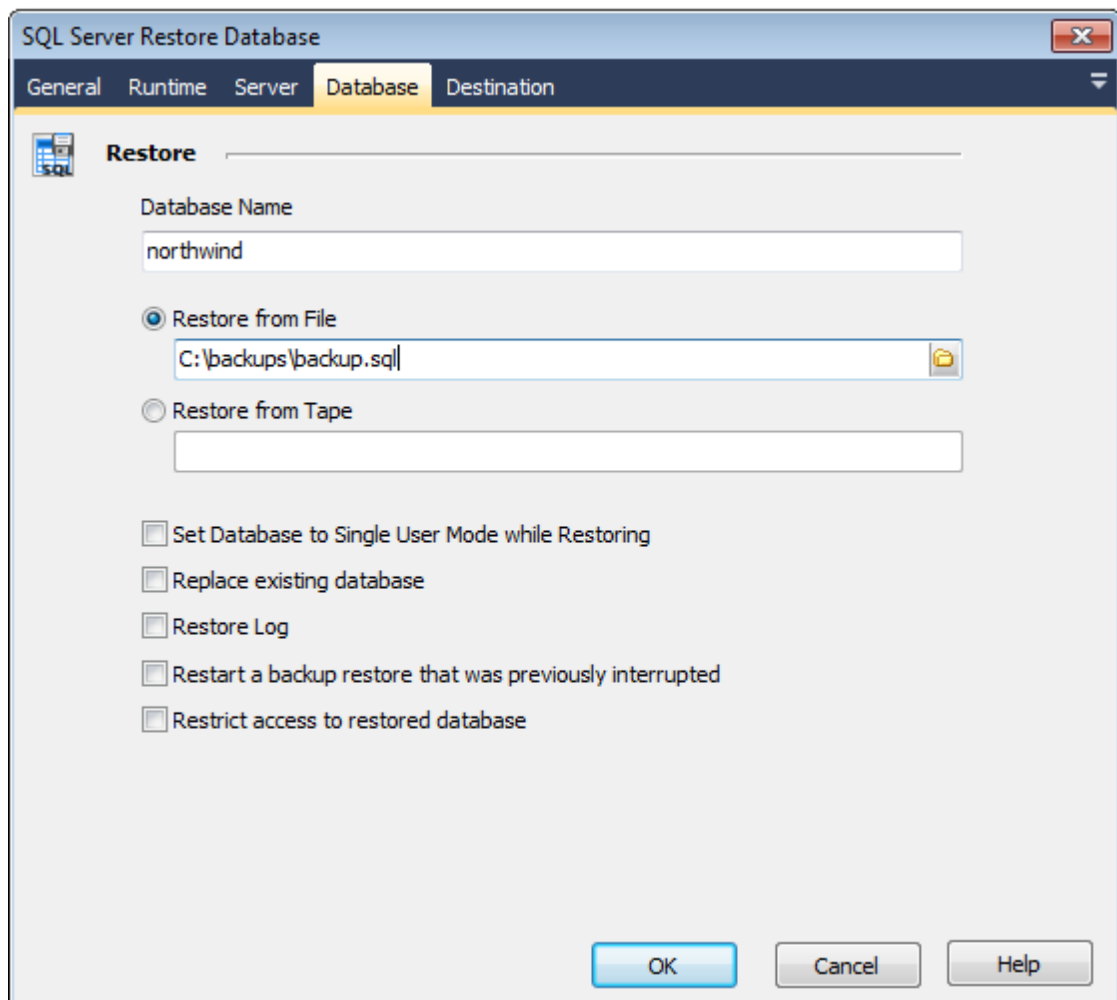
us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.8.1.10 SQL Server Restore Database

[FinalBuilder Professional Edition]

The SQL Server Restore Database action enables you to restore a SQL database from a disk or tape drive.



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.

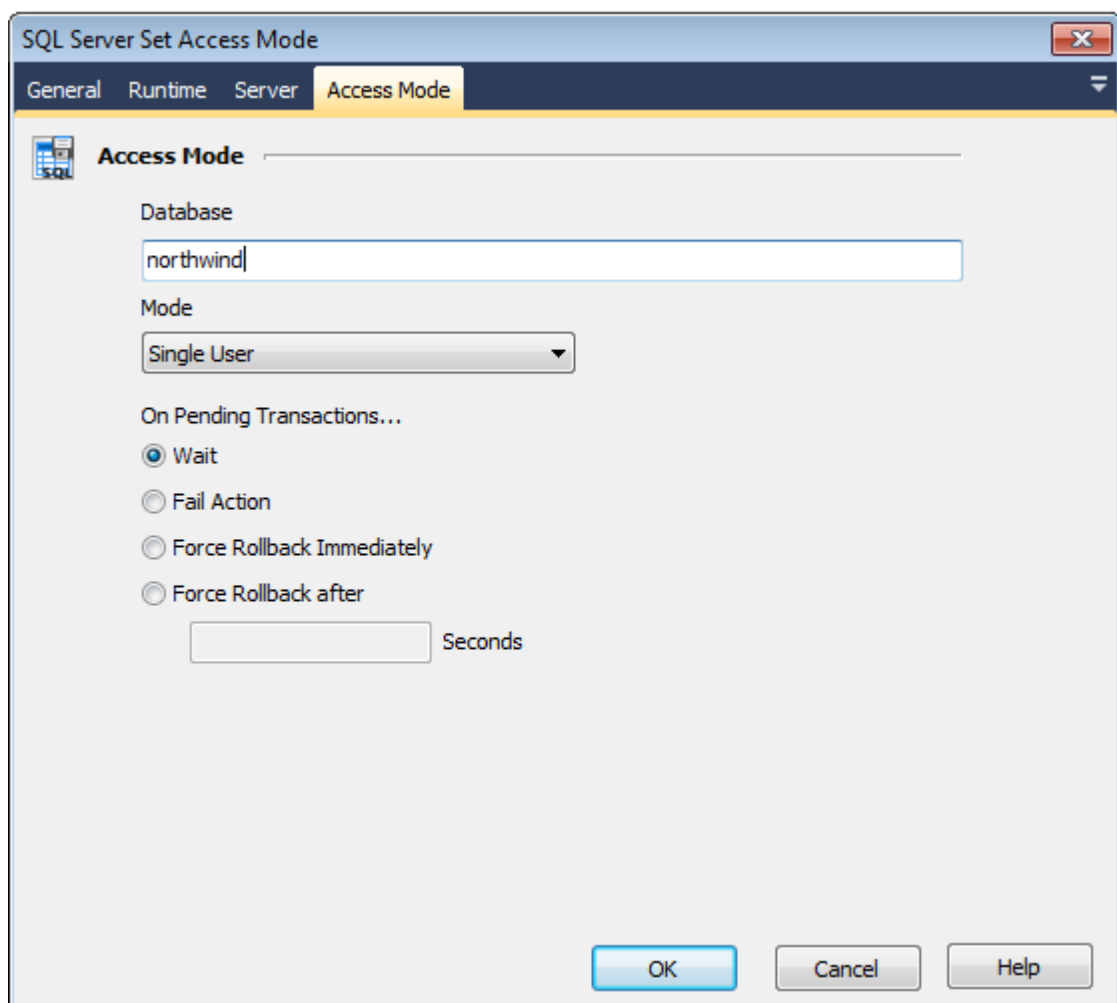
Restrict access to restored database

Restricts access to the newly restored database to members of the *db_owner*, *dbcreator* and *sysadmin* roles.

6.8.1.11 SQL Server Set Access Mode

[FinalBuilder Professional Edition]

The SQL Server Set Access Mode action enables you to change the access mode of a database.



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.

6.8.1.12 SQL Server Update DB Statistics

[FinalBuilder 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.

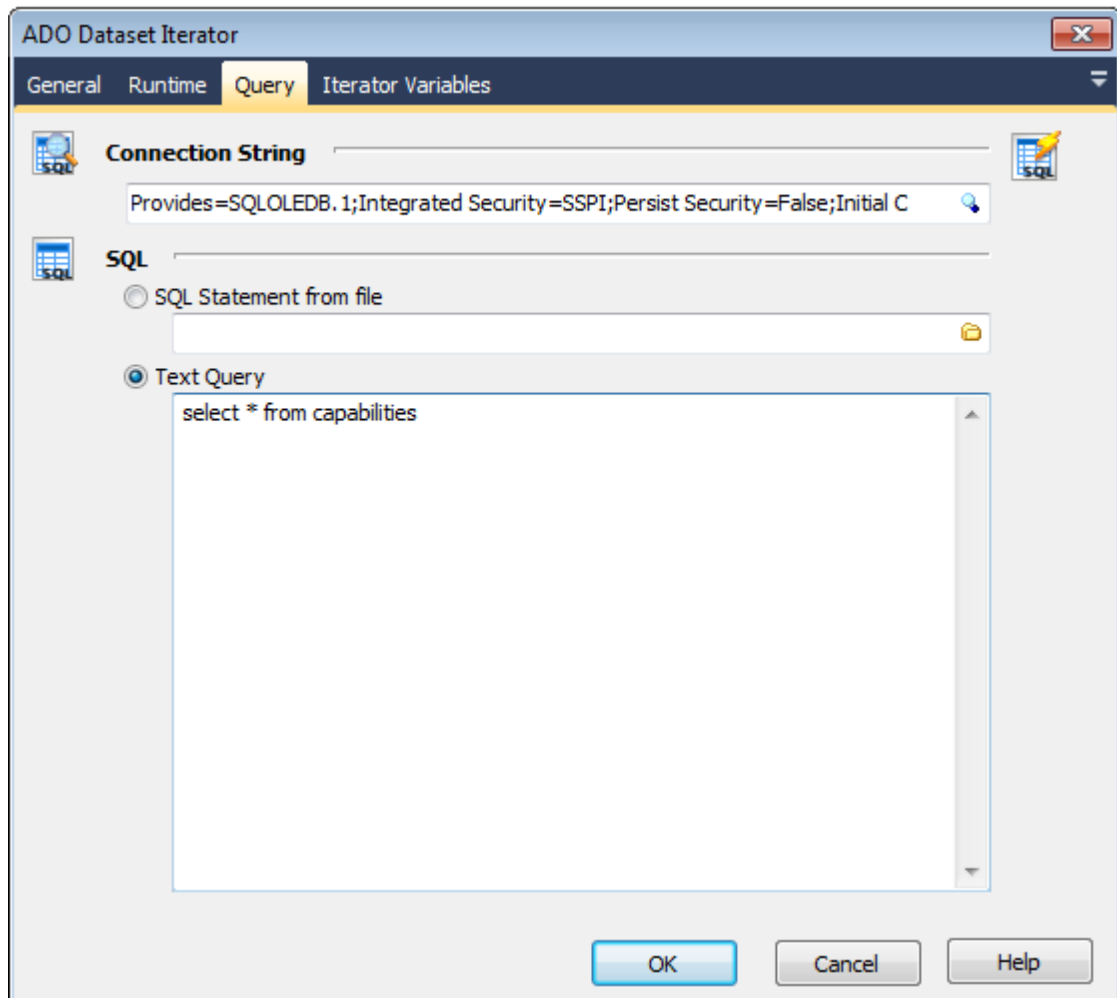
For more information on the sqlmaint tool see: http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

To set the location of SQLMaint see SQL Server Options

6.8.2 ADO Dataset Iterator

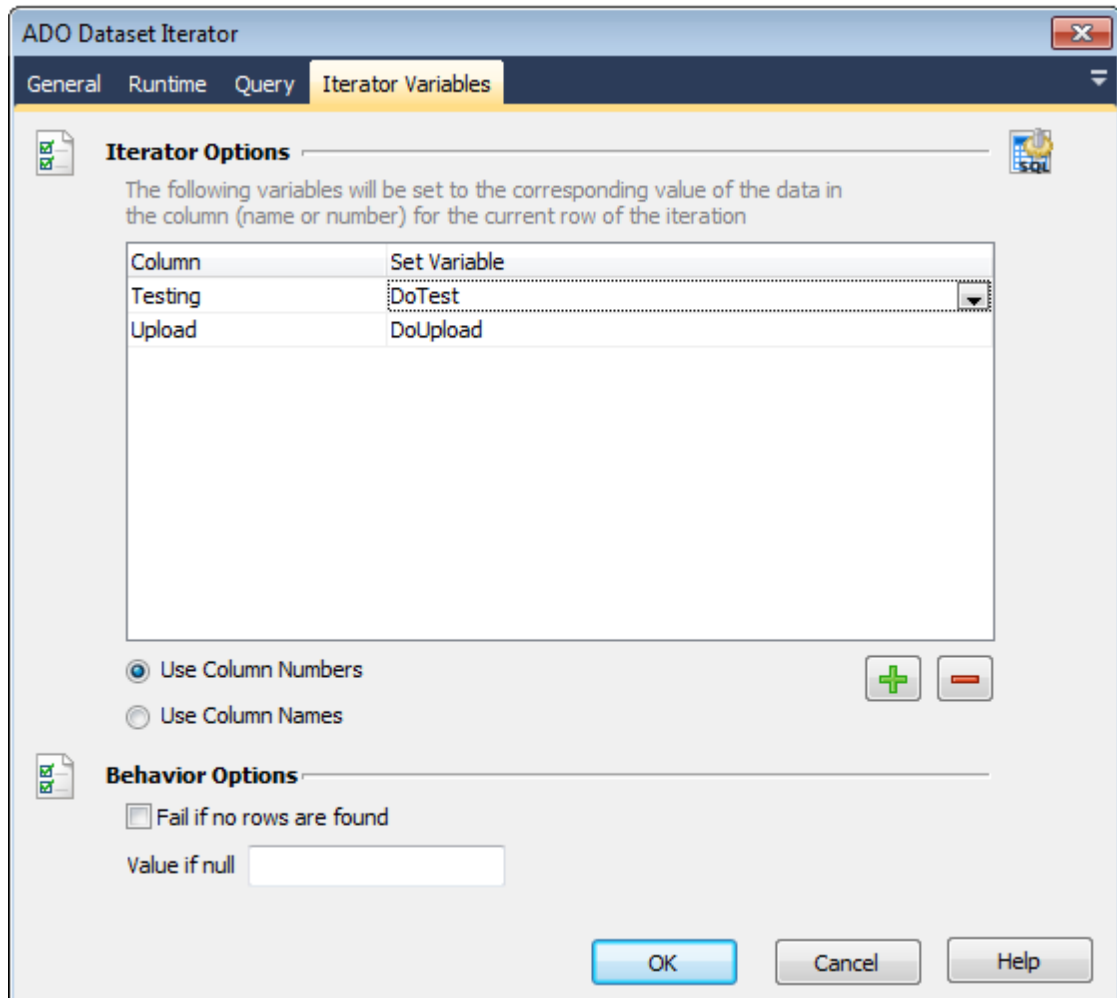
The ADO Dataset Iterator action allows you to perform a group of actions for each row returned from a SQL query on a database using ADO.

For more information on Iterator style actions, see Iterators.



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

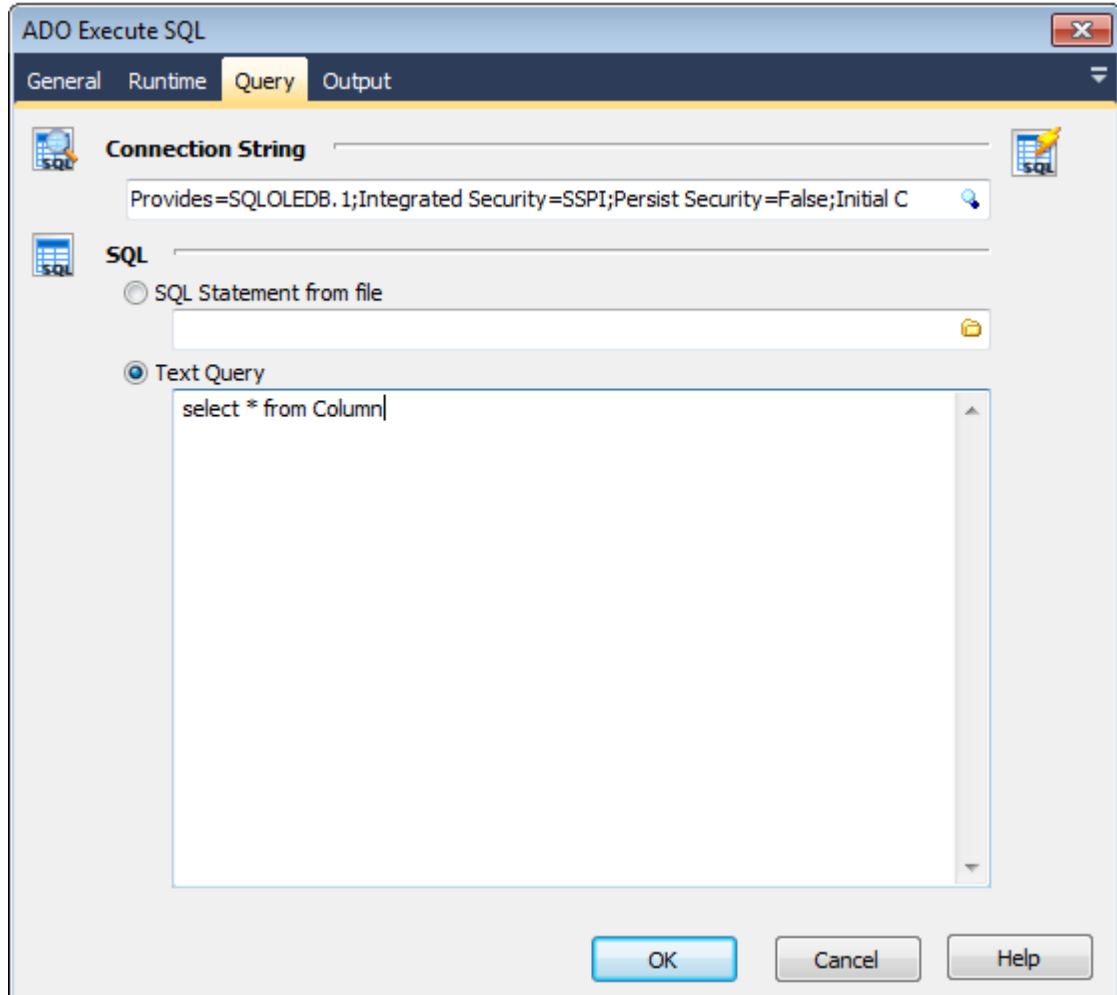


Use the grid to set up which column values you want to put in variables for each iteration of the action. The above example shows that the Capability column and Version column values will be placed into the CapabilityName and VersionNum FinalBuilder variables. Use the Plus and Minus buttons to add and 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.

6.8.3 ADO Execute SQL

The ADO Query action enables you to execute a SQL statement against an ADO database connection



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.

6.8.4 ADO Execute Stored Procedure

The ADO Stored Procedure action enables you to automate the execution of stored procedures using ADO.

The action property pages help 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 output 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 FinalBuilder

ADO Execute Stored Procedure

General Runtime **Details**

Connection String

curity=SSPI;Persist Security=False;Initial Catalog=NorthWind;Data Source=scarecrow

Stored Procedure

CustOrdHist Load Procedures

Parameter Name	Type	Parameter Input Value	Output To
@RETURN_VALUE	Return		
@CustomerID	In	BERGS	

+ Add - Delete Load Parameters

☒ 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>

OK Cancel Help

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 be 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.

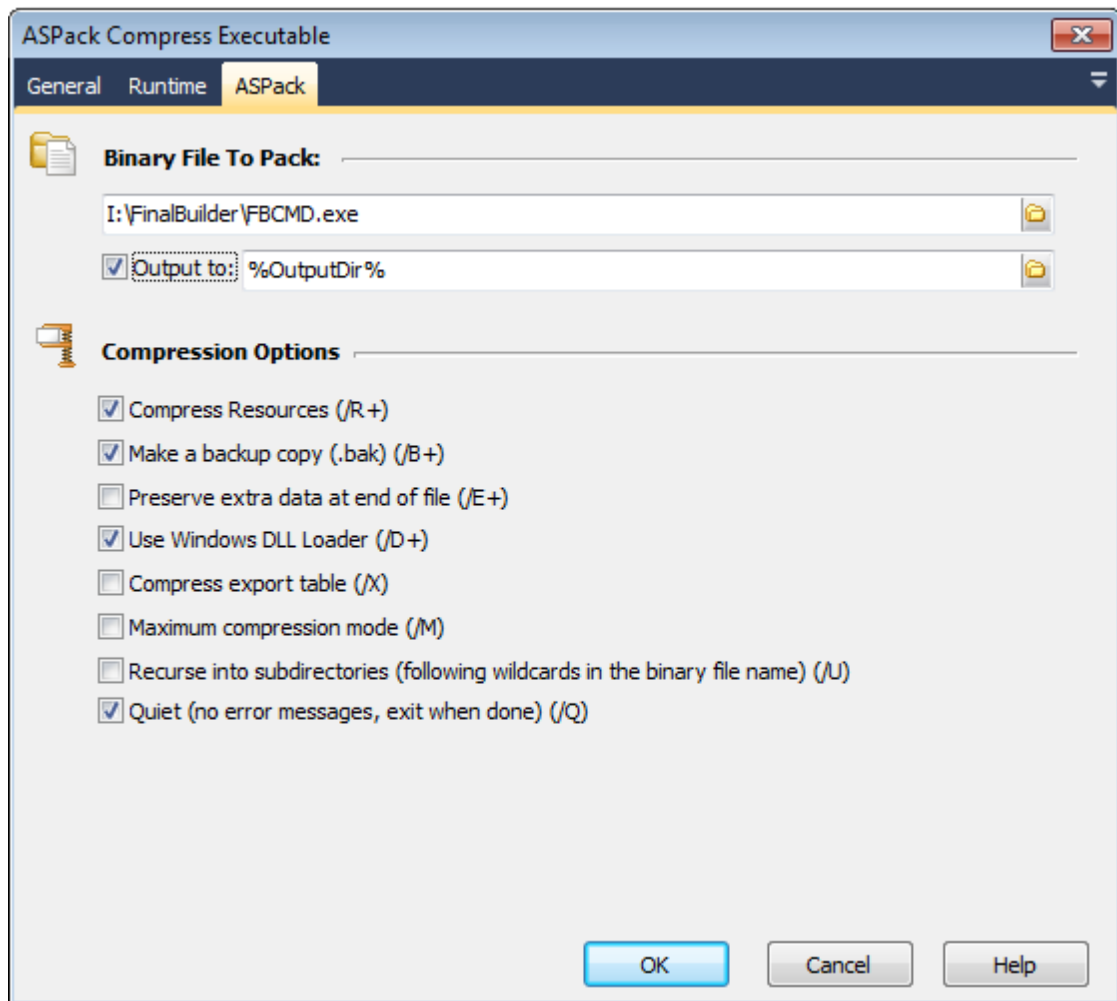
SQL Output - You can output a result recordset to the FB log, and/or to an XML file using the ADO XML file format.

6.9 Executable Packers

6.9.1 ASPack Action

The ASPack action uses the ASPack utility to compress a binary executable or library file.

For more information on ASPack, see the ASPack website <http://www.aspack.com/>.



Binary File to Pack

Specify the name of an executable or library file to packed. Optionally, you can also specify an output path for the packed file.

Compression Options

Each of these options corresponds to a different ASPack command line option (each switch is shown in the label for the checkbox.) See the ASPack documentation for more details.

For options where a switch is listed with a + sign (ie /R+), the unchecked option will

result in a minus sign (ie /R-.)

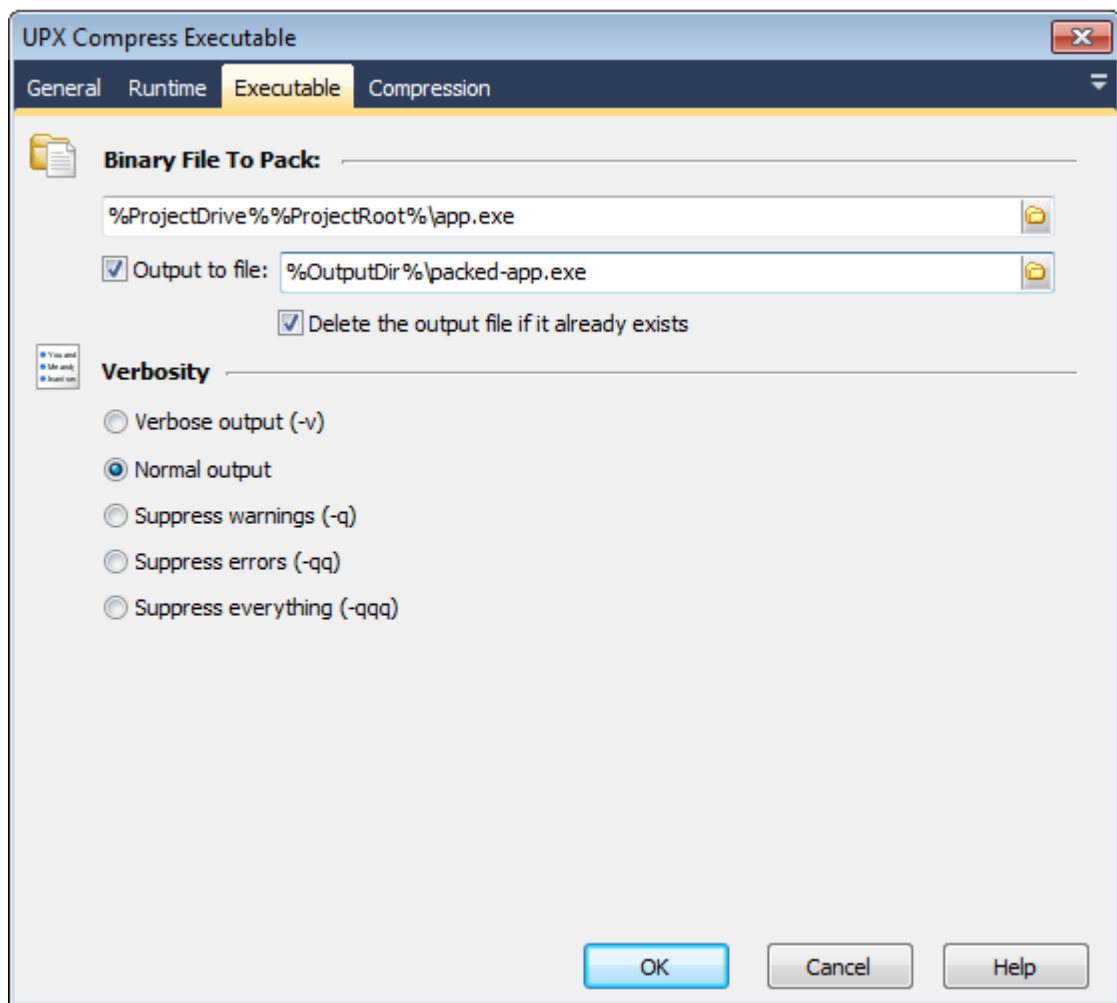
It is not recommended that the "Quiet" option is unchecked when running automated builds, as this may lead to ASPack pausing and displaying a modal dialog.

6.9.2 UPX Pack Executable

UPX ([The Ultimate Packer for Executables](http://upx.sourceforge.net/)) is a free (open source) packer for executable programs. UPX packed applications take up far less space on disk, in exchange for slightly higher memory use on startup. UPX can also pack Win32 DLLs.

For more information on UPX, see <http://upx.sourceforge.net/>

For more *specific technical information* on UPX options, see the 'upx.doc' file that comes with UPX.



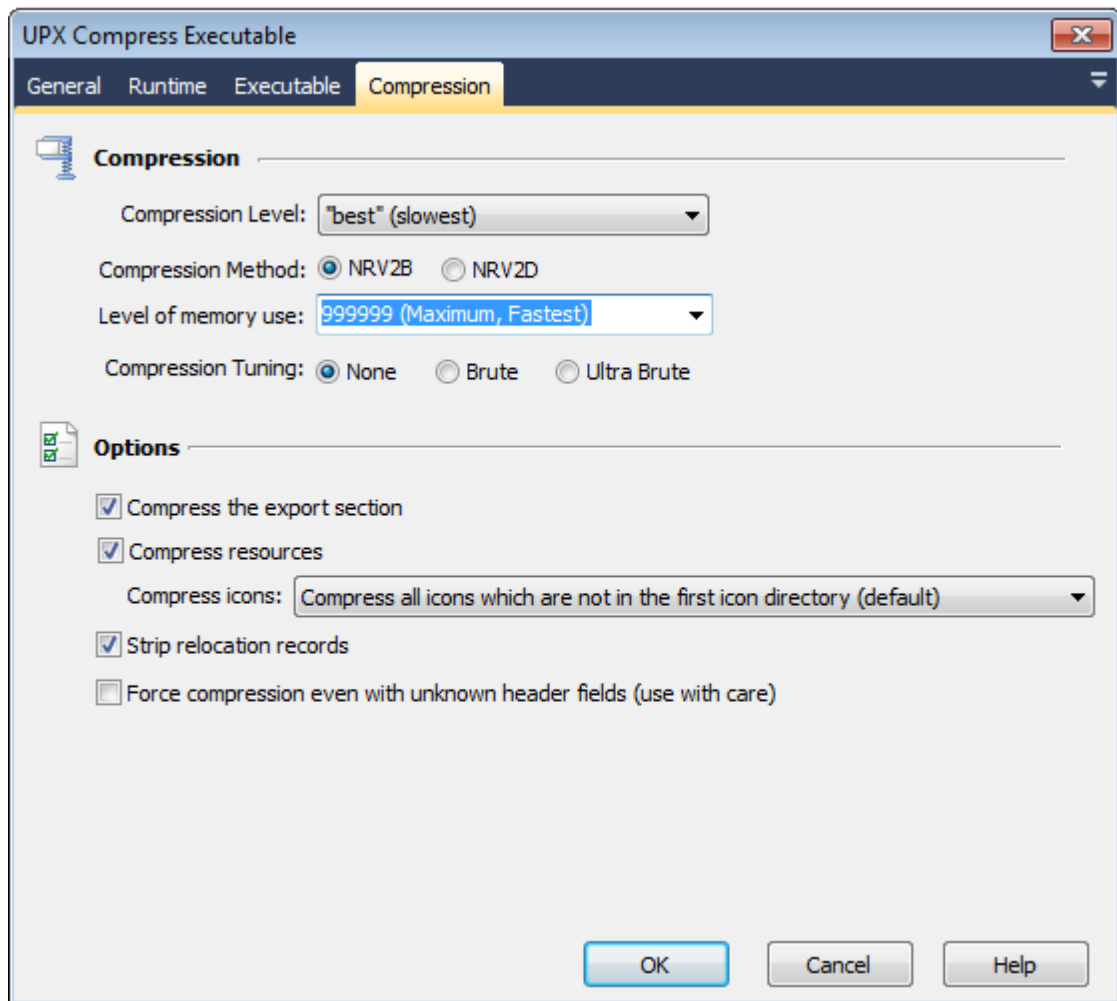
Binary File to Pack

UPX can pack executable files, Win32 DLLs and screensavers.

If you do not specify an output file, the compressed executable will be written over the original file. Note that you cannot compressed an already-compressed executable.

Verbosity

Specify the level of output from the UPX utility.



Compression Level

Compression levels range from 1 (fastest) to 9 (slower, better) and then "best" (maximum compression.)

Compression Method

UPX currently supports two compression algorithms, NRV2B and NRV2D. UPX's creators recommend trying both and seeing which gives better results for your application.

Level of memory use

Specify the amount of memory which UPX will use for packing the executable.

Higher numbers will use more memory to provide faster results and (slightly) better compression. Setting the value to 999999 will use as much available memory as required for the best results.

Compress the export section

Equivalent of (`--compress-exports=0/1`)

If checked, the export section of the Win32/PE executable will be compressed. This gives a good saving in size, but does not work with all applications. See the UPX documentation for more details.

Compress resources / Compress icons

Equivalent of (`--compress-resources=0/1` and `--compress-icons=0/1/2`)

UPX can be set to compress application resources, including icons. See the UPX documentation for more details.

Strip Relocation Records

Equivalent of (`--strip-relocs=0/1`)

If checked, UPX will strip the relocation records of executables. See the UPX documentation for more details.

Force compression even with unknown header fields

Equivalent of (`--force`)

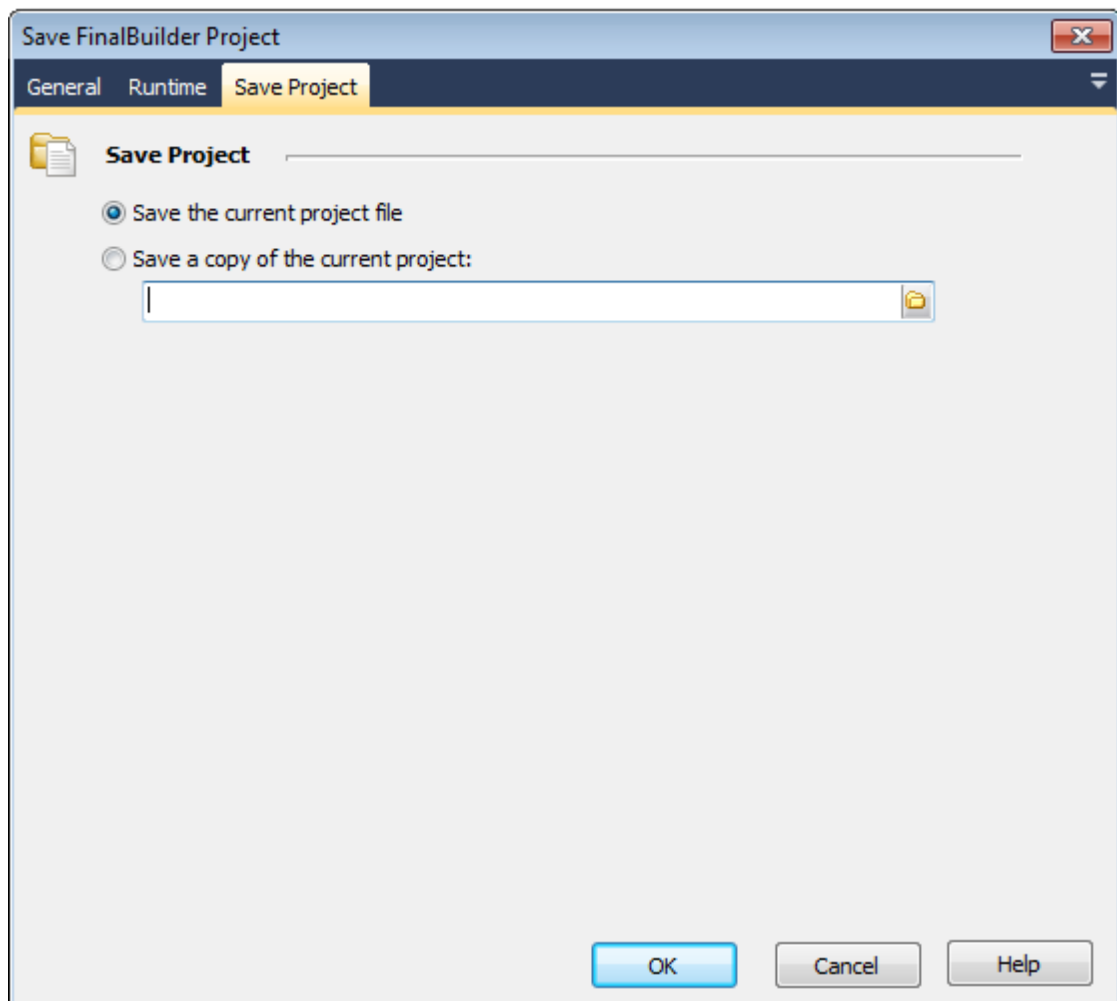
If this option is checked, it forces UPX to compress the executable even if it finds unknown data in the executable header fields. This may render your application unusable. See the UPX documentation for more details.

6.10 FinalBuilder IDE

6.10.1 Save FinalBuilder Project Action

The **Save FinalBuilder Project** Action allows you to save the current FinalBuilder project from inside your build.

Reasons you may want to do this include saving changes to action properties, and saving the status of persistent variables.



Save the current project file

Saves the current project to the current project file. The equivalent of choosing File - > Save in the FinalBuilder IDE.

Save a copy of the current project

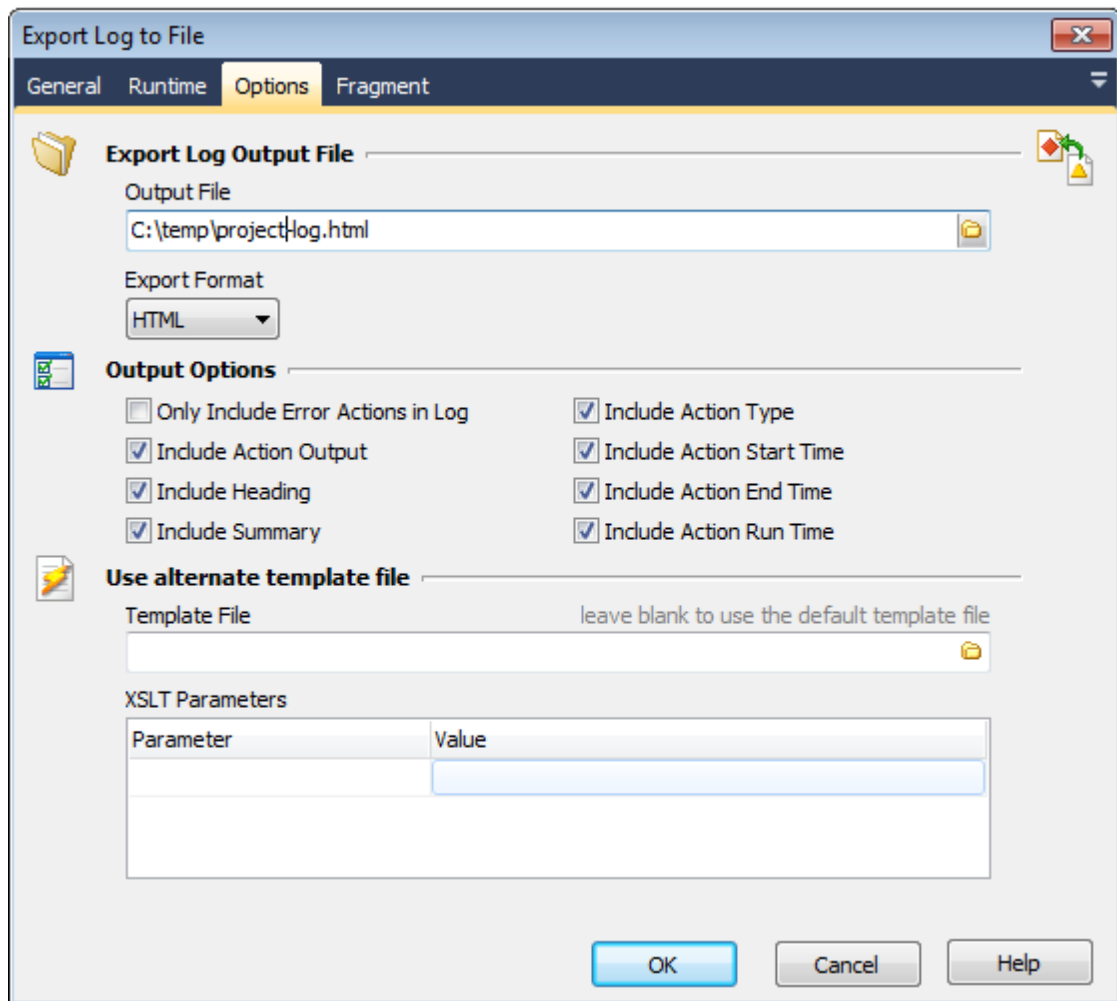
A copy of the current project will be written to the file you specify.

6.10.2 Export Log Action

The **Export Log Action** exports a FinalBuilder 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.



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 - FinalBuilder Install Directory\stylesheets\ConvertLogToHTML.xsl
- XML - FinalBuilder Install Directory\stylesheets\ConvertLogToXML.xsl
- Text - FinalBuilder 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): FinalBuilder 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

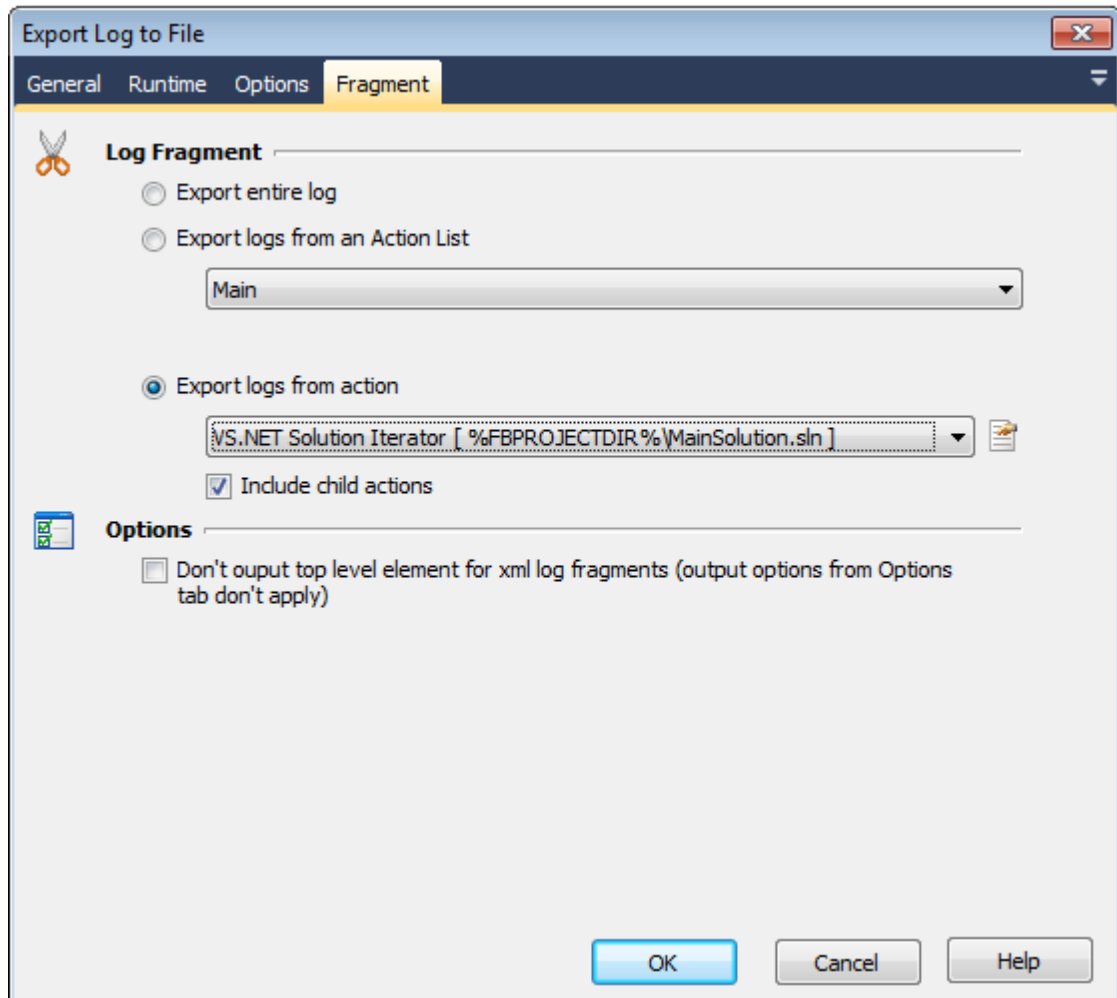
Template File - You can also specify a different XSL Stylesheet template file to alter the output format of the log file. When you leave Template File blank, FinalBuilder will choose the appropriate stylesheet depending on the Export Format chosen (see Export Format above).

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 FinalBuilder 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.



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 be exported to the log.

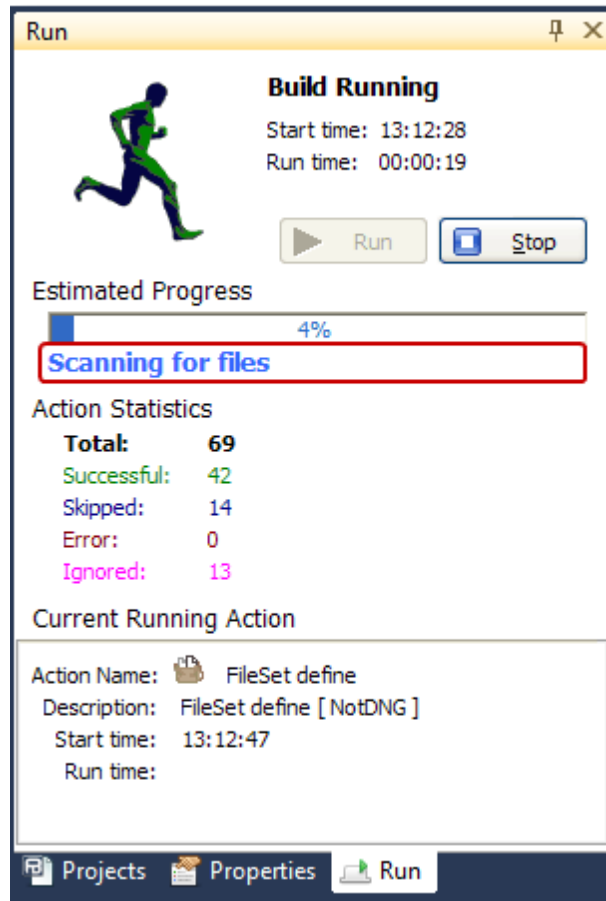
6.10.3 Set Estimated Progress Total Action

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.

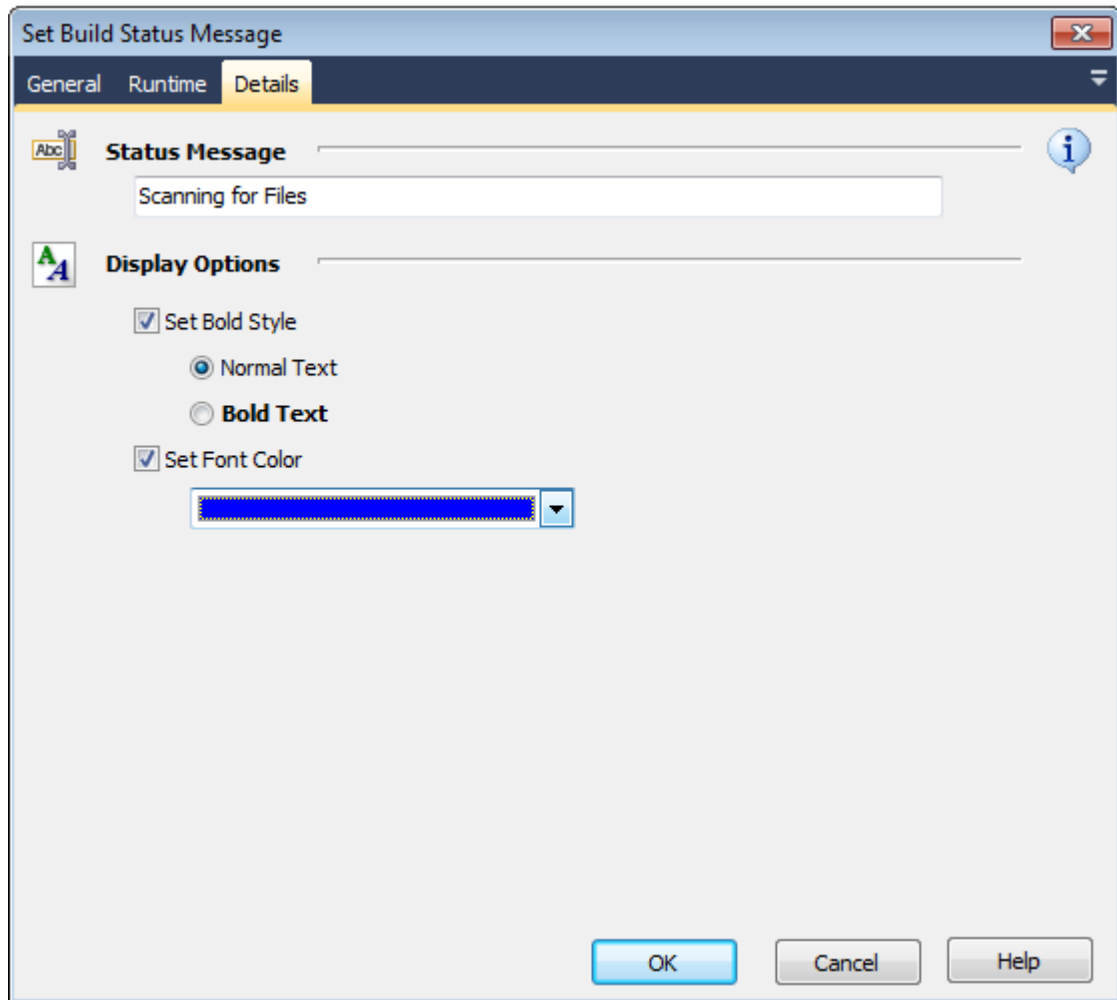
6.10.4 Set Build Status Message Action

The Set Build Status Message action allows you to add a custom message to the build 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 build is.



The above screen shot shows the custom status message "Scanning for files".

The options for Set Build Status Message action are:



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.

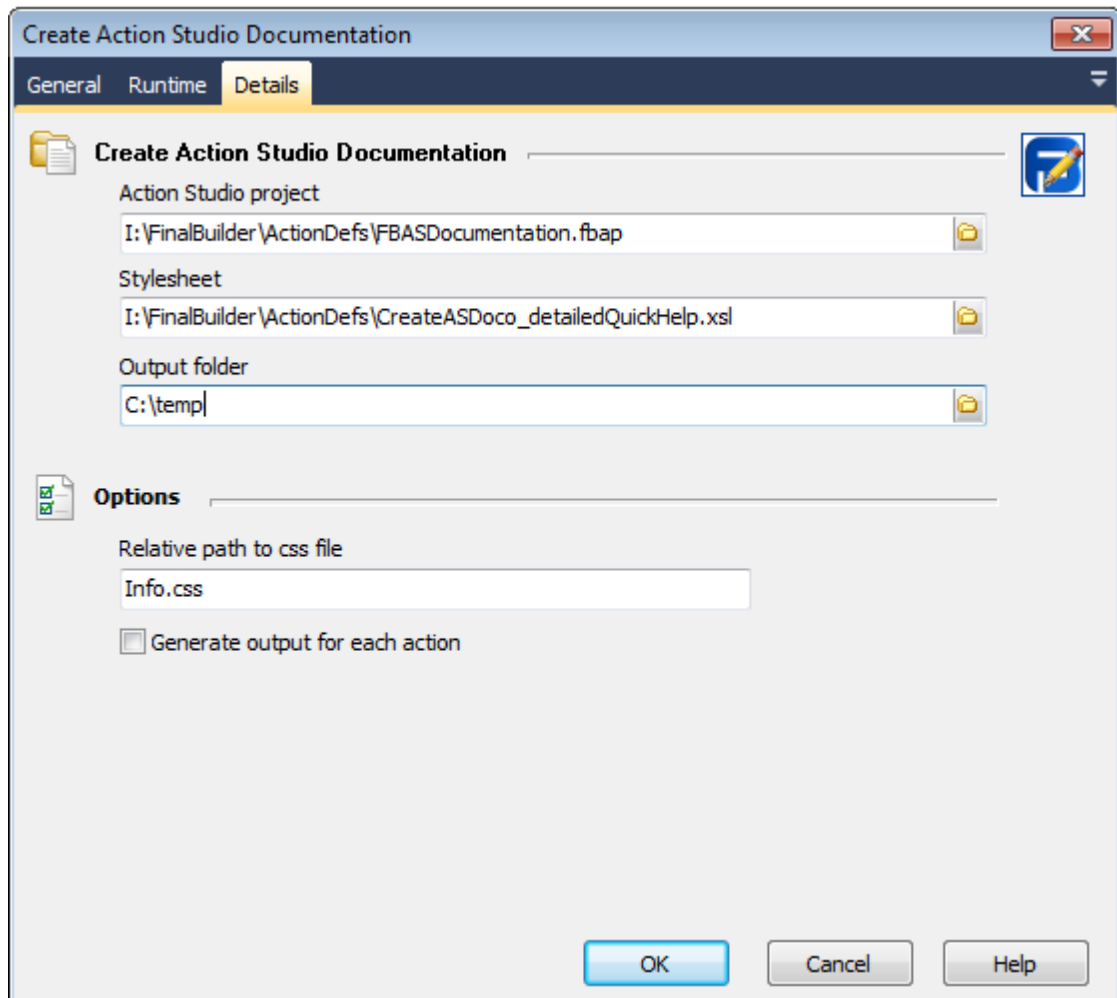
Set Font Color: sets the color of the text (and then set the colour using the colour combo box).

When a new build is started, the status message will be cleared and the font and style set back to default (ie. Normal and Black).

6.10.5 Create ActionStudio Documentation Action

This action will generate documentation for your actions written in ActionStudio.

There are two xslt stylesheets included in the installation which can be used. They are located in the Stylesheets directory, and called "CreateASDoco_ListActions.xsl" and "CreateASDoco_DetailedQuickHelp.xsl"



Action Studio Project - the path to your ActionStudio based action package

Stylesheet - the XSLT stylesheet used to transform the xml to the desired html file

Output - the output location

Relative path to css file - specify the path to the .css file you want to use for the html. The default .css file is the same one used by the other actions.

Generate output for each action - this will generate a new file for each action found in the .fbap

6.11 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, FinalBuilder 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 build. See Accessing FileSets to see how other actions in the build can use the list of files selected.

6.11.1 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 FinalBuilder 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 FinalBuilder 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

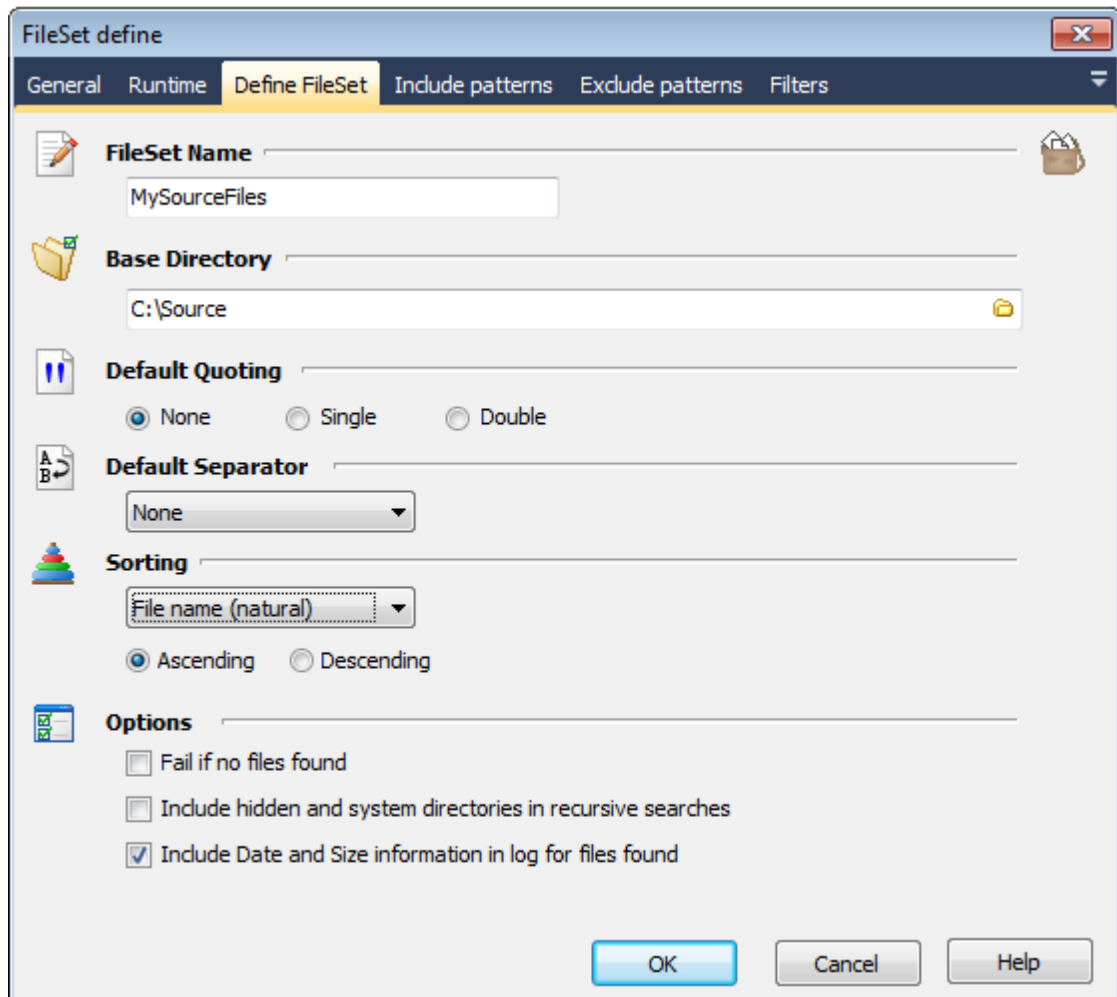
To retrieve all files in the FileSet overriding the default separator and quote type:

Variable Syntax: %MyFileSet.File5:quote=single:sep=comma%

Script: MyFileSet.OutputQuoting = qtSingle
 MyFileSet.OutputSeparator = spComma
 MyFileSet.AllFiles

6.11.2 FileSet Define

The FileSet Define action is used to create a new FileSet.



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 Patterns 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

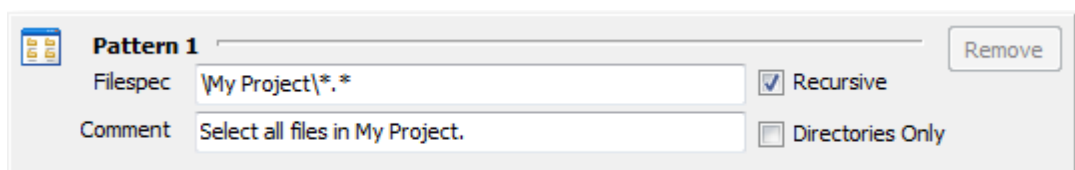
For more information on Filters, see FileSet Filters

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

6.11.2.1 FileSet Patterns

FileSet Patterns are used to specify which files are included and excluded from the FileSet.



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\

6.11.2.2 FileSet Filters

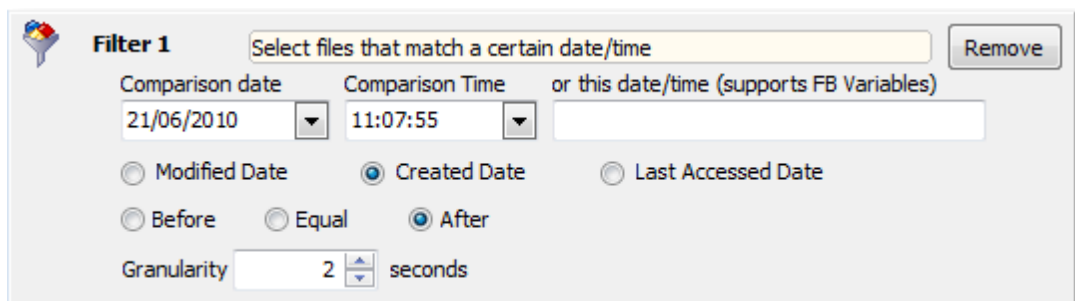
Filters are used to further refine the files chosen by properties other than the file name or directory. For example, you may want to only include files of a certain size, containing particular text, matching particular attributes, or based on date and time (or a combination of all of these). When a 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

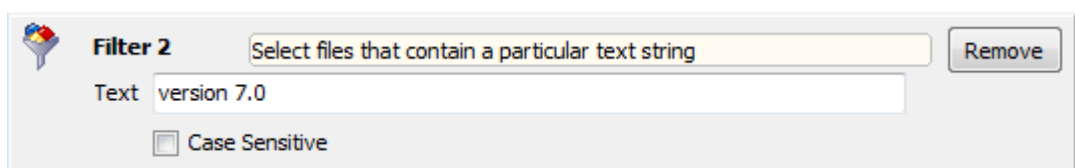
The Date/Time Filter allows you to filter the FileSet files based on their modified date, accessed date, or creation date.



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

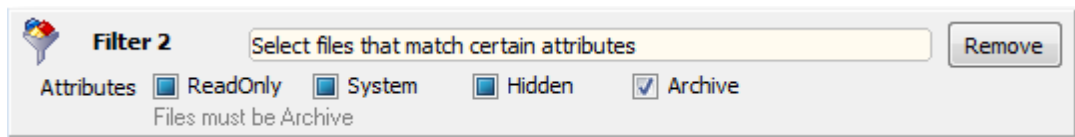
The Contents filter allows you to filter the FileSet files depending if their contents contain certain text. This filter needs to open each file and scan for the text, so the performance of creating the FileSet may be poor, especially if it contains large files.



This example shows that only files which contain the text "version 4.0" will be added to the FileSet.

Attributes Filter

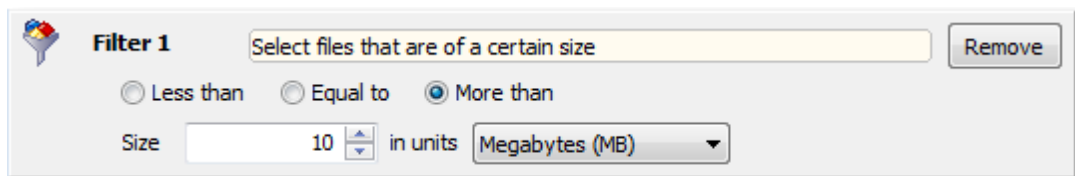
The Attributes filter allows you to filter the FileSet files depending on the file system attributes of the files.



This example shows that files must have the Archive bit set.

Size Filter

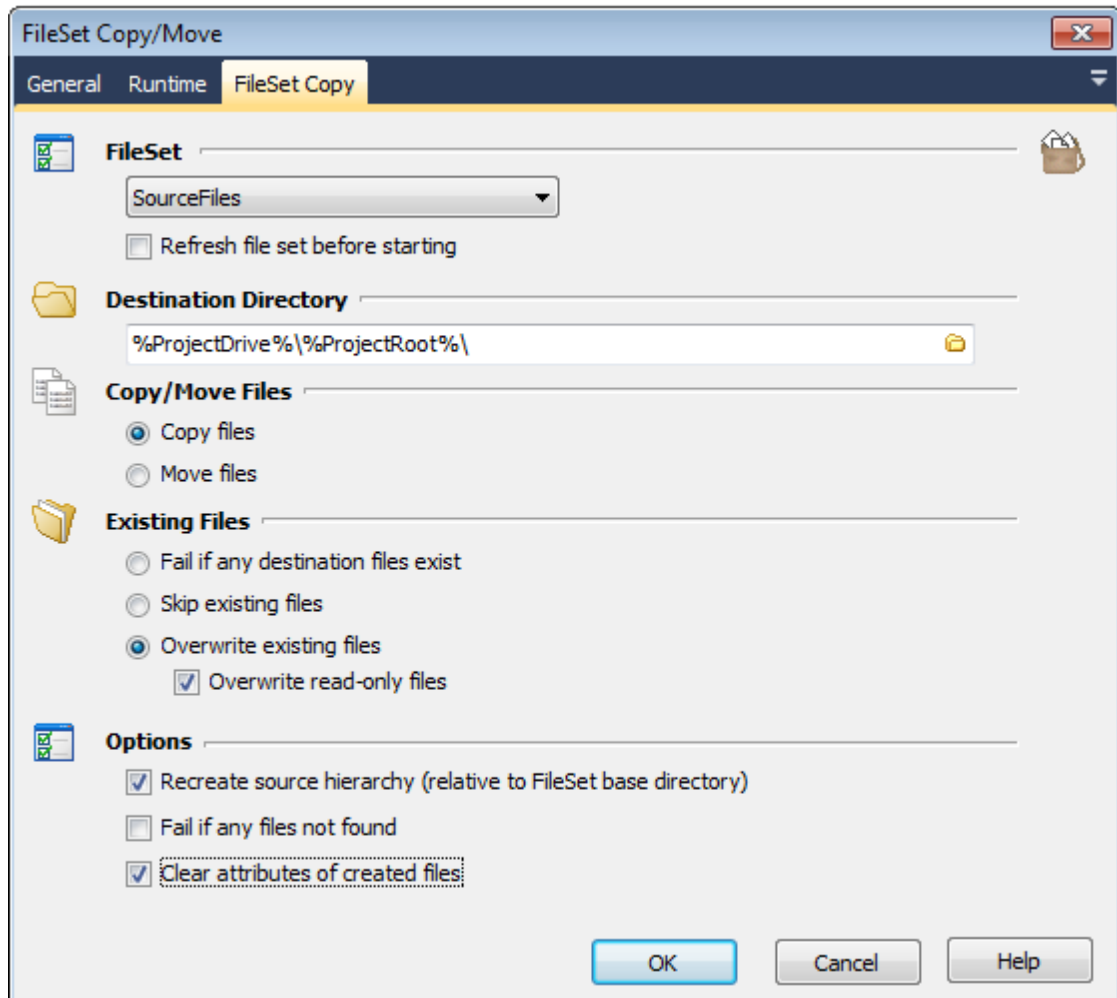
The Size filter allows you to filter the FileSet files depending on the size of the file specified in bytes, MB, KB or GB.



This example shows that only files bigger than 10 MB will be added to the fileset.

6.11.3 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 - 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

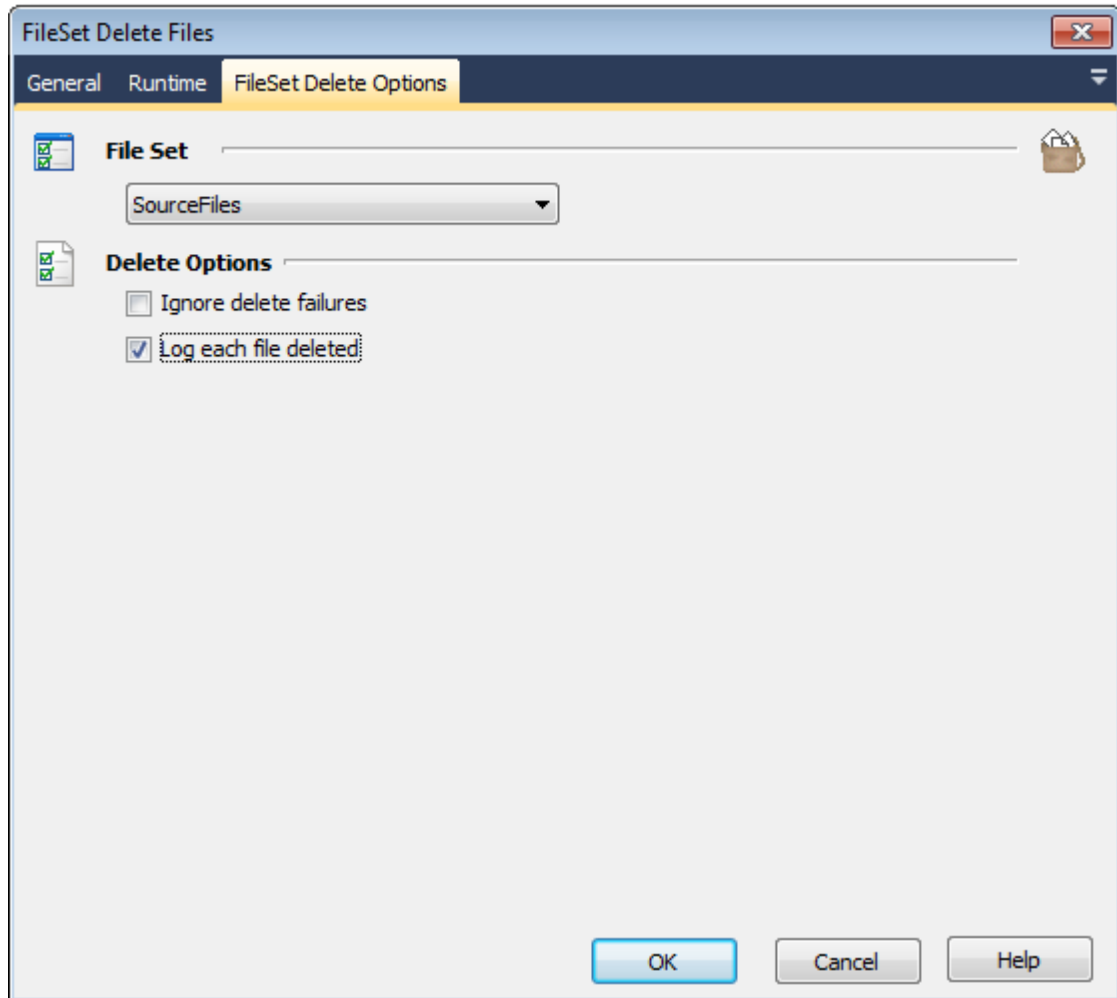
Recreate source hierarchy - if this option is not set, then all the files in the FileSet (regardless of their path) will be placed in the destination directory.

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

6.11.4 FileSet Delete Files

The FileSet Delete Files action will delete all the files in the specified FileSet.



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

6.11.5 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.

6.11.6 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 build it may be necessary to Refresh the FileSet as files may have changed (eg. more files added, files deleted, attributes changed, etc.)

6.11.7 FileSet to Variable

The FileSet to Variable action allows you to create a listing of the files specified by the FileSet in a FinalBuilder 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

6.12 Files & Directories

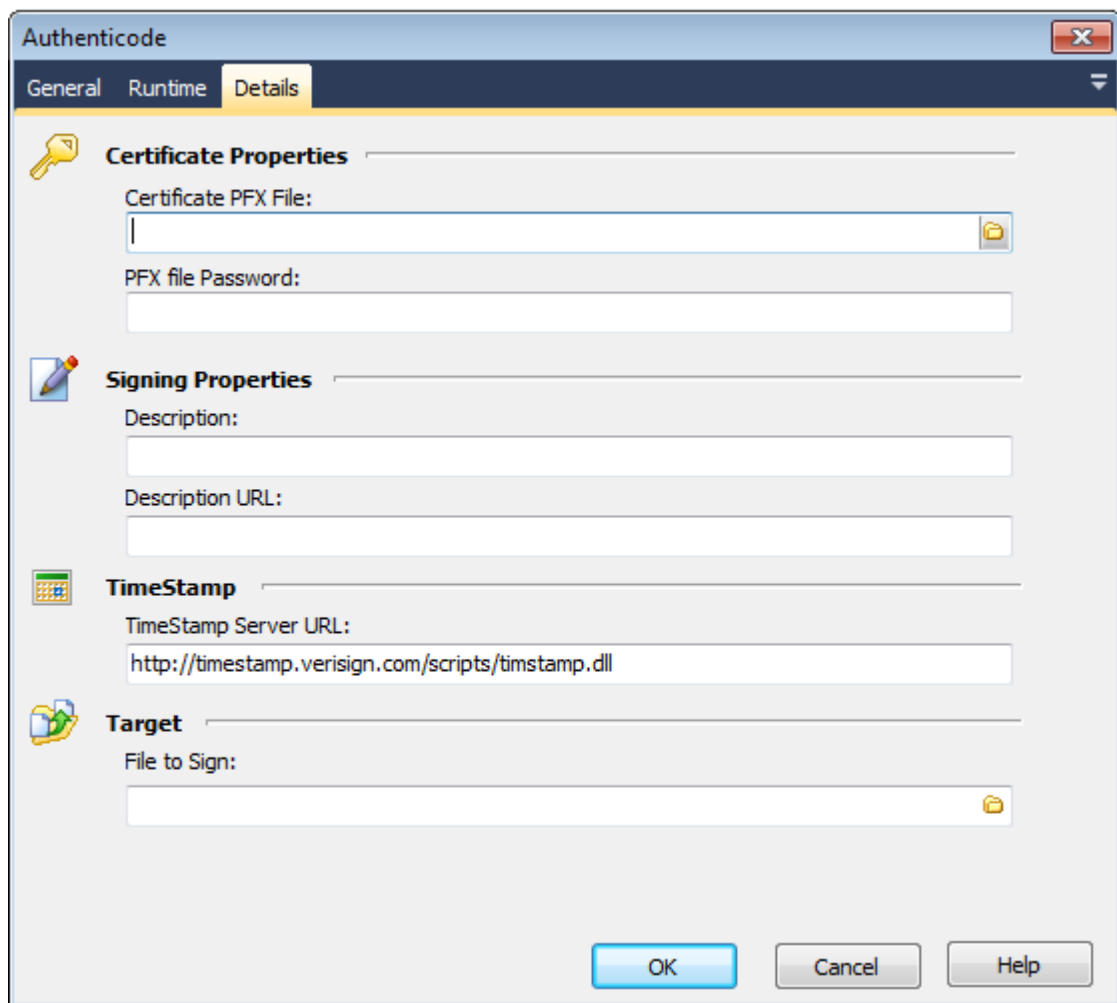
6.12.1 Authenticode

The Authenticode File action enables you to automate Authenticode signing of your executable files during your build.

This action requires your Authenticode Signing Certificate to be in a PFX file.

Many CAs provide certificates in two files: an SPC and a PVK. See below for details on converting these to a PFX file.

The Authenticode action also contains fields for "Description" and "Description URL". These fields contain the information which is shown when Windows prompts the user to confirm they want to run the executable (ie the first time a downloaded program is launched, or when a UAC confirmation dialog is shown.)



Creating a PFX File

To make the PFX, you need to use a Microsoft tool called PVKIMPRT. PVKIMPRT is included as part of the Windows Platform SDK, or can be obtained directly at <http://office.microsoft.com/downloads/2000/pvkimprt.aspx>.

Open a Command Prompt, switch to the directory that contains your certificate files and type:

```
pvkimprt -PFX mycert.spc mykey.pvk
```

(Replace the *mycert* and *mykey* file names where required).

You will be prompted to enter your private key password (if set.) A dialog will appear with the certificate export wizard. Make sure you select 'Yes, export the private key', and 'Include all certificates in the certification path if possible' options. You are then prompted to enter a password: this is the password you will use for code signing with the created PFX (it should be secure.) Finally, you will be prompted for a location to save the new PFX file.

License Information

This action includes source code from StreamSec. Its license is reproduced here.

StreamSec LICENSE

Copyright (c) 2004, Henrick Wibell Hellström, StreamSec

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 name of StreamSec 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

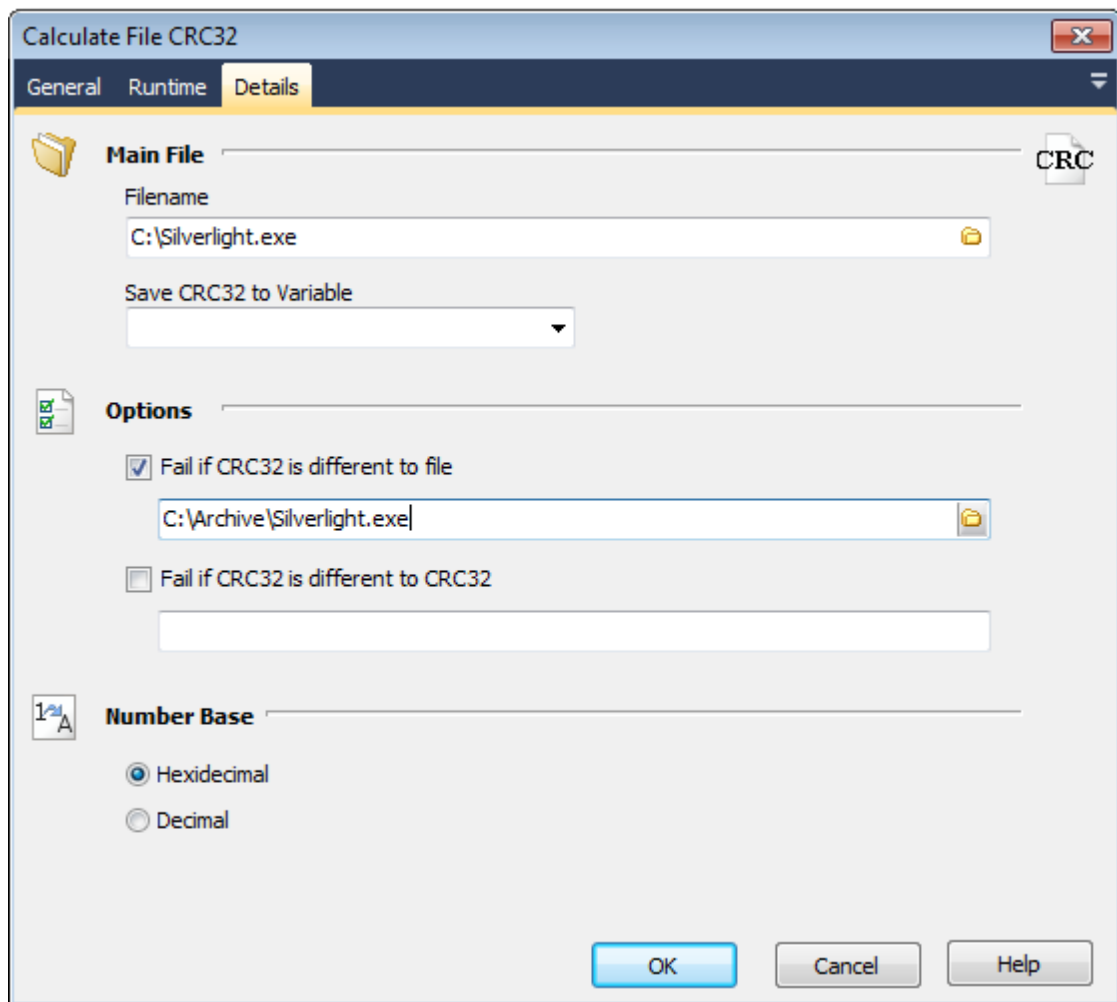
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.12.2 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 $7 / 3 = 2 + \text{rest } 1 \Rightarrow 1$ 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).



Filename - specify the file which you want to calculate the CRC32 for.

Variable - specify the FinalBuilder 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.

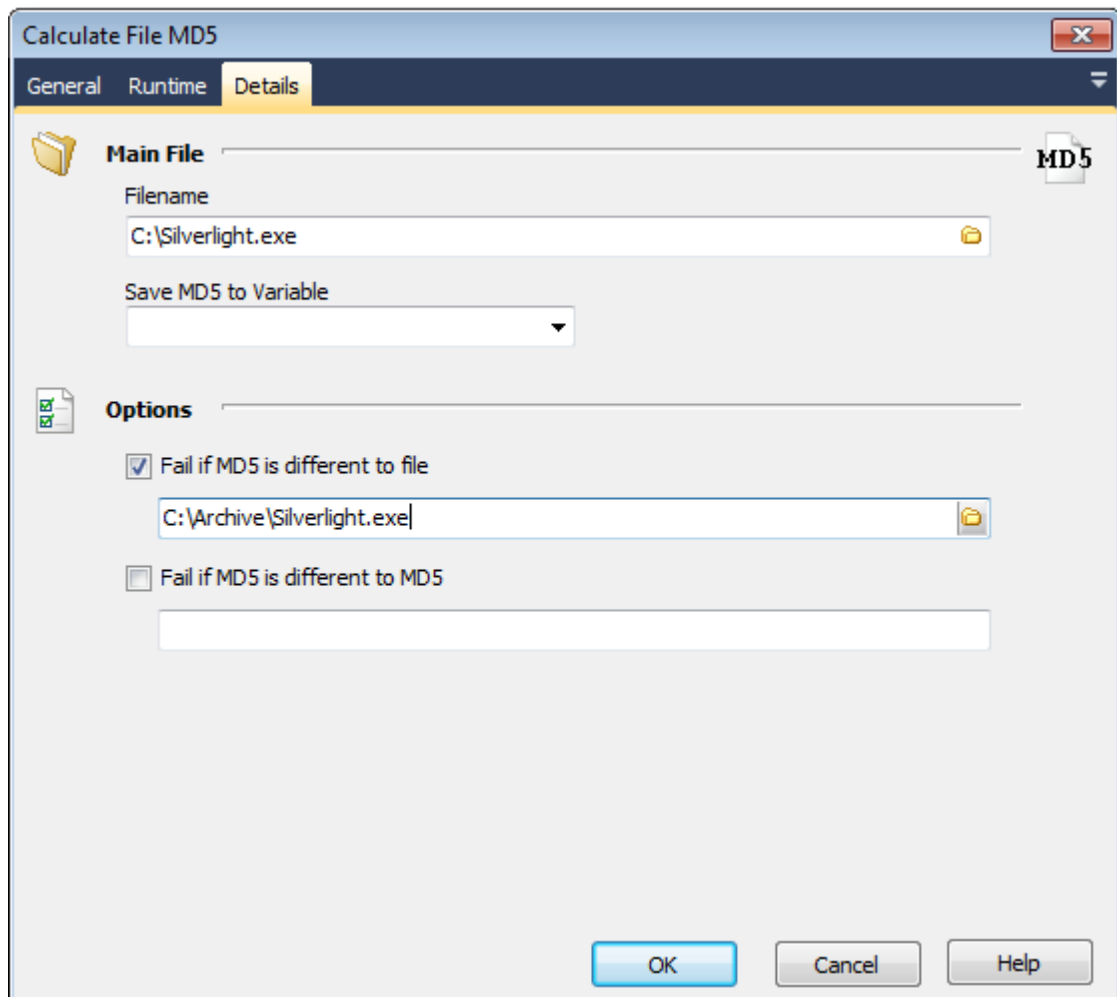
6.12.3 Calculate File MD5

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.



Filename - specify the file which you want to calculate the MD5 for.

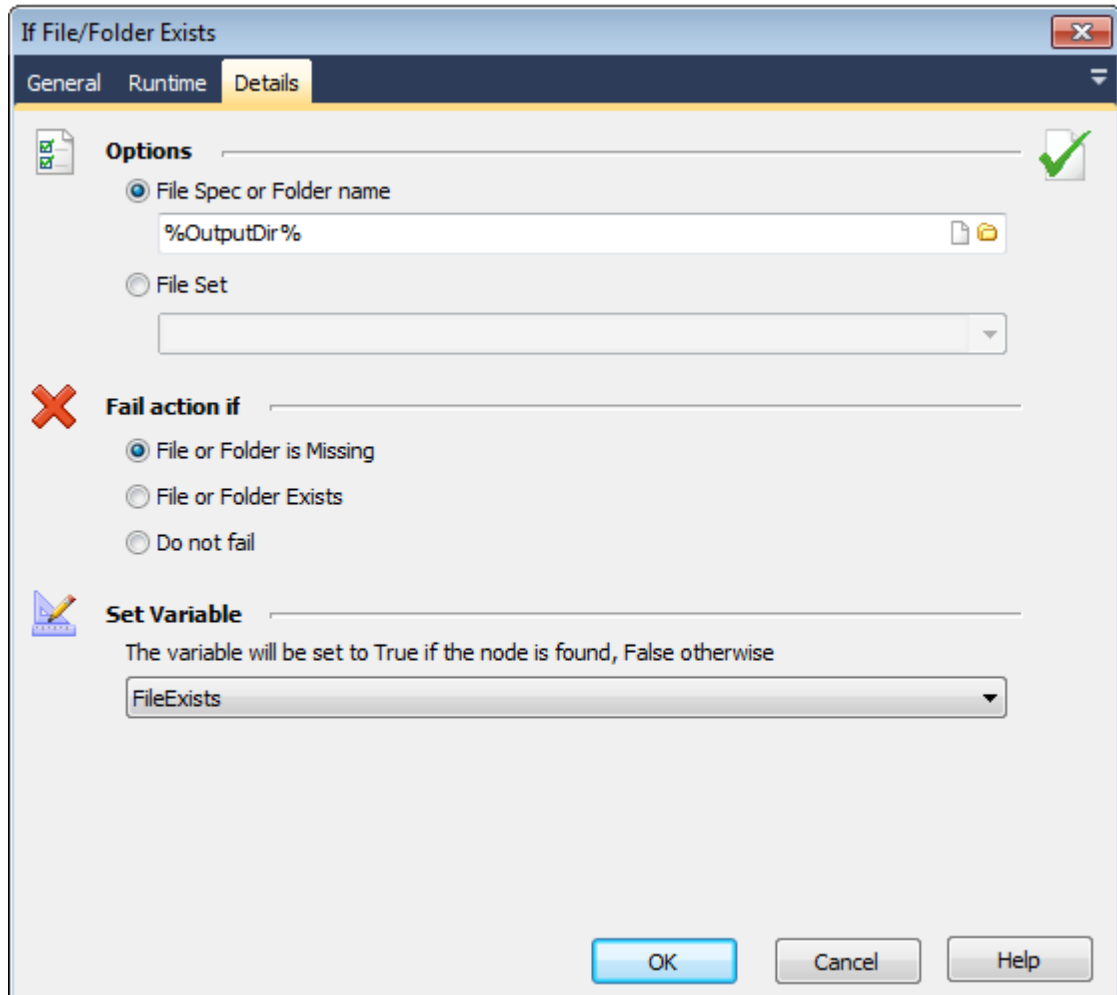
Variable - specify the FinalBuilder variable to save the calculated file MD5 to.

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

6.12.4 Check if File Exists Action

This action allows you to check for the existence of a specific file or folder.



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:\builds\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.

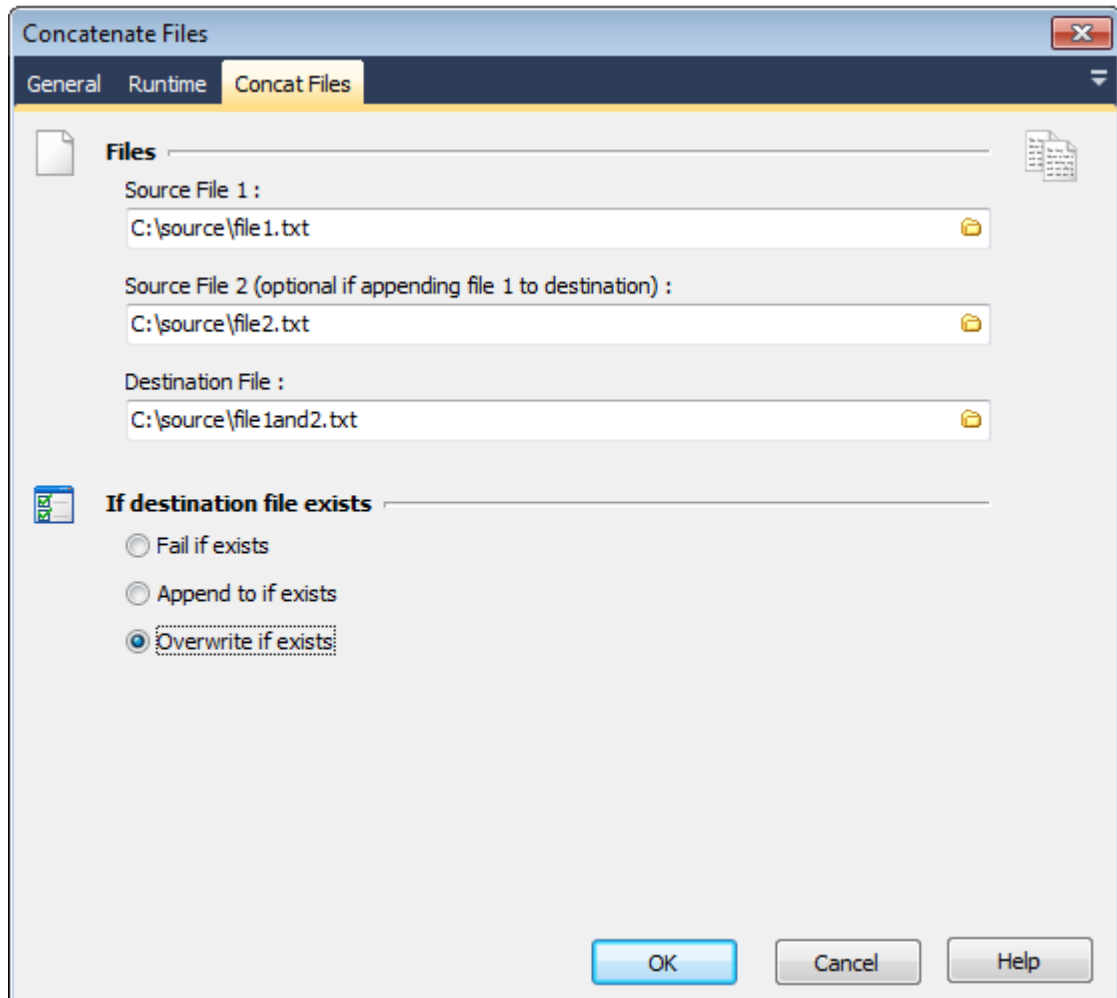
The build will skip the sub-node if the file is not present, but will keep running.

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.

6.12.5 Concatenate Files Action

This action concatenates two text files into a single new file, or merges/append a file into the destination file.



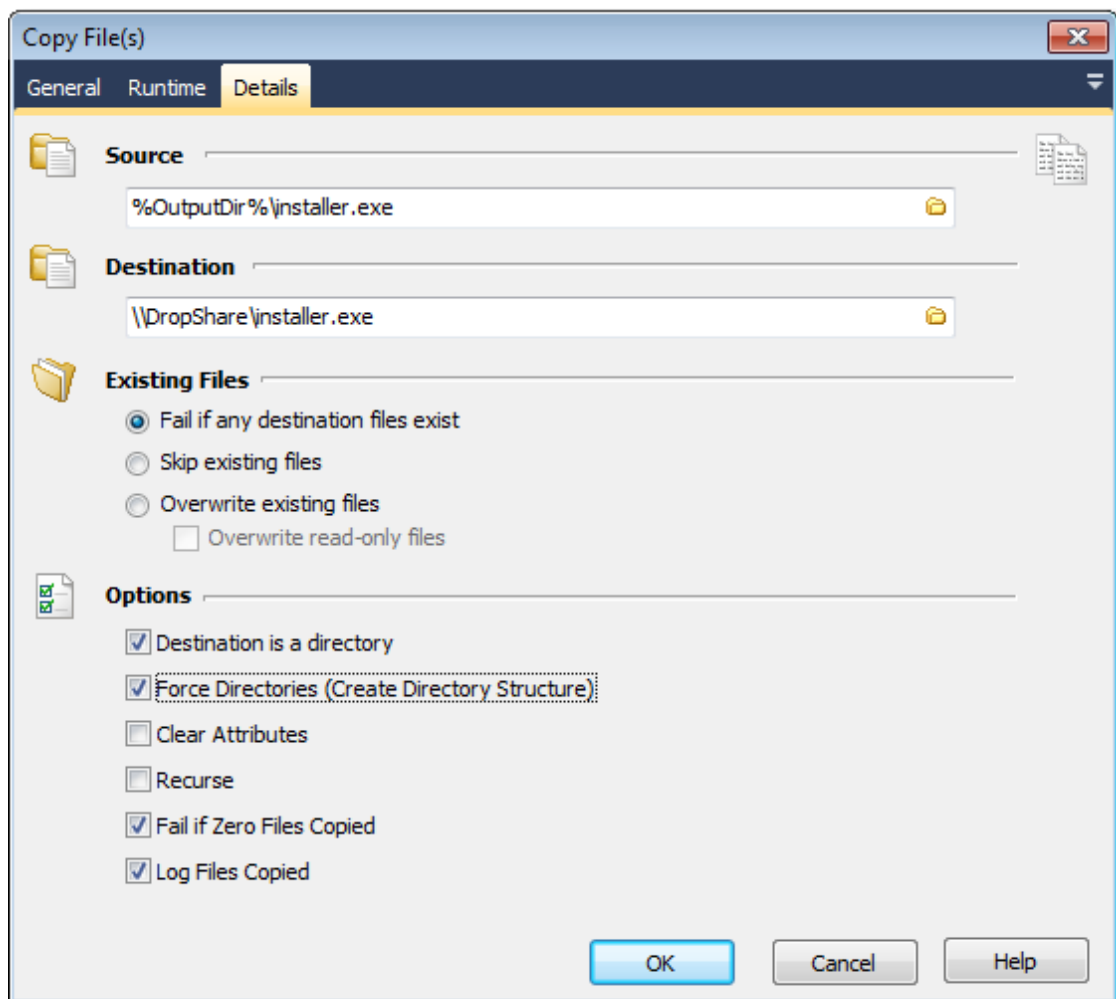
Note that Source Files 1 & 2 cannot be the same file, and they cannot be the same as the destination file. Source 2 may be blank if appending to an existing file.

6.12.6 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 FinalBuilder 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.



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;

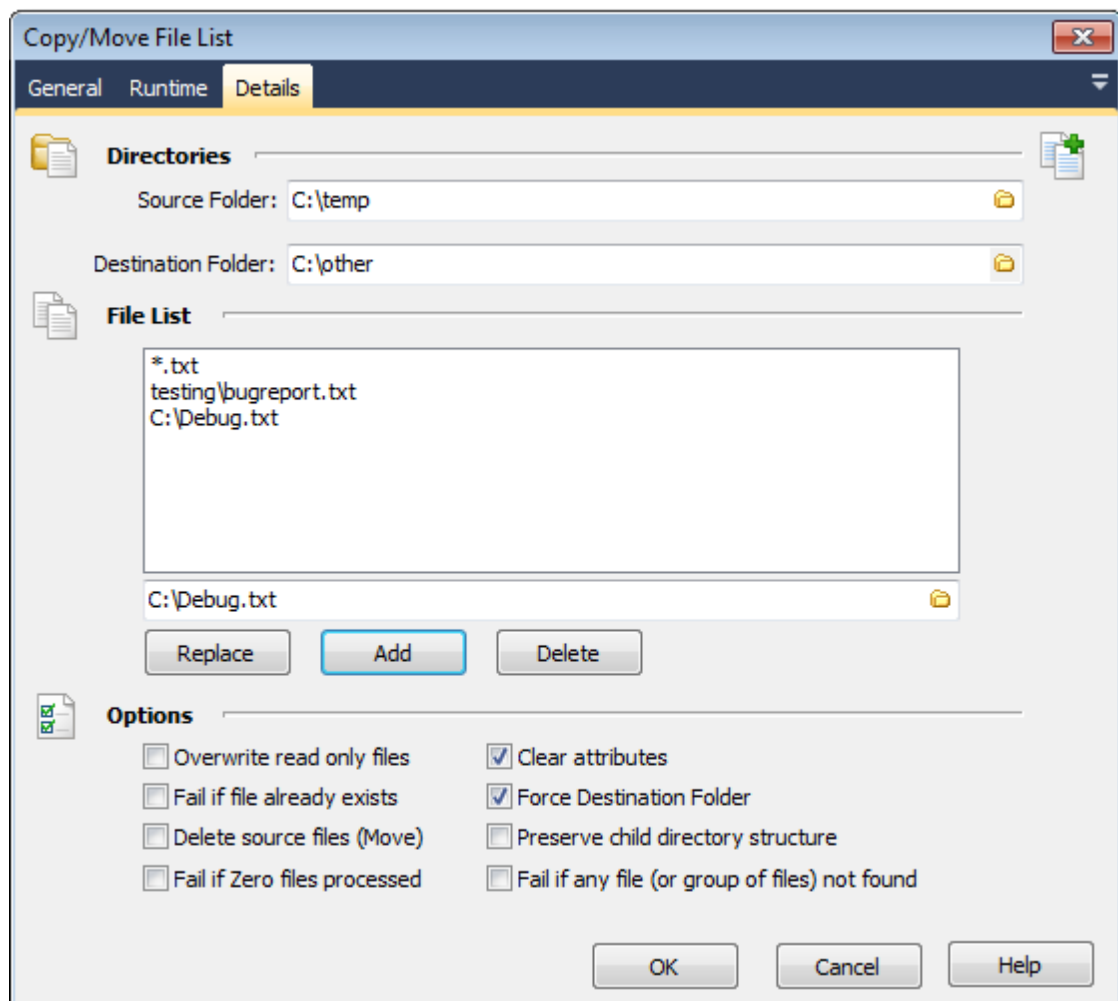
property LogFilesAffected : WordBool;

6.12.7 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 FinalBuilder. 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 FinalBuilder users. The source for this action is installed as an example of creating custom FinalBuilder Actions.



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.

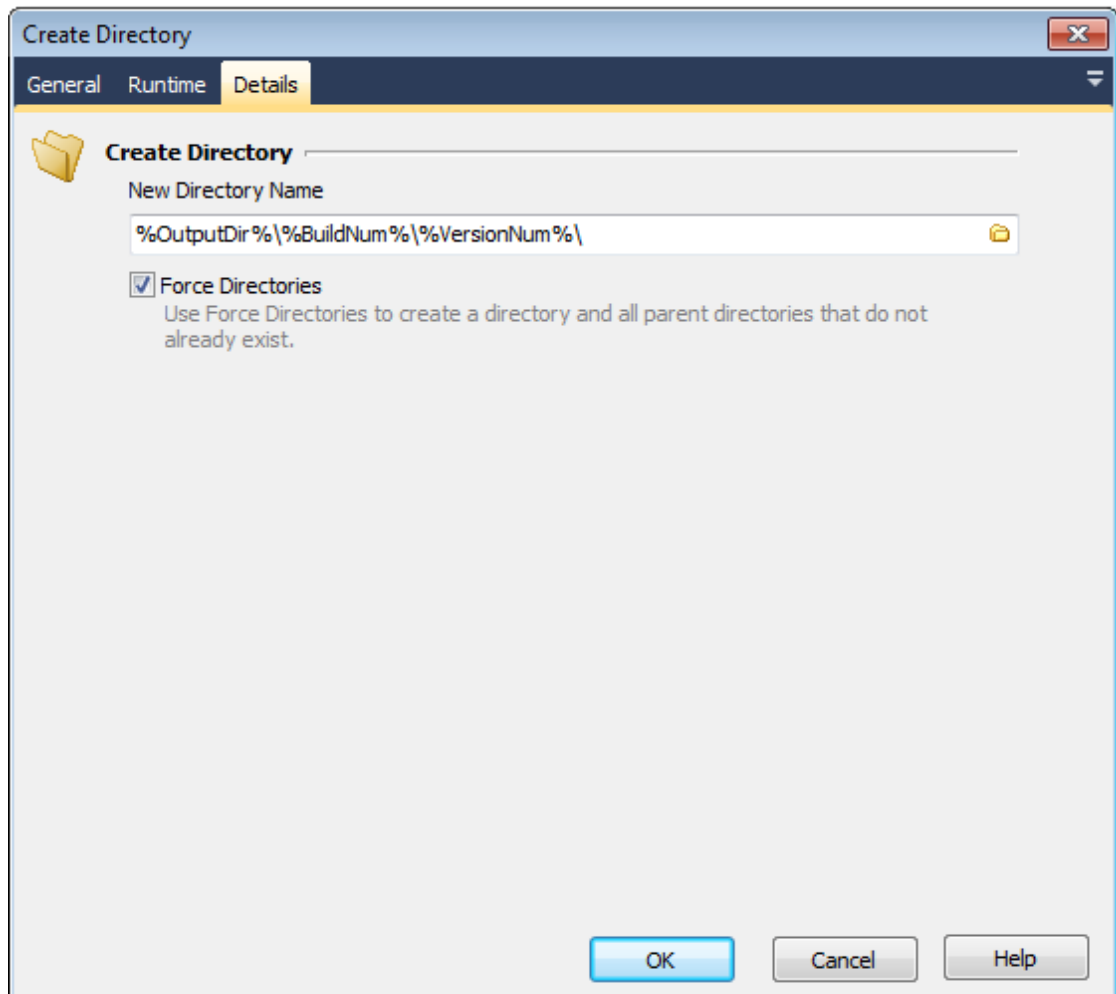
File List

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.)

6.12.8 Create Directory Action

This action will create the specified directory, if the specified directory already exists then the action will not do anything.



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

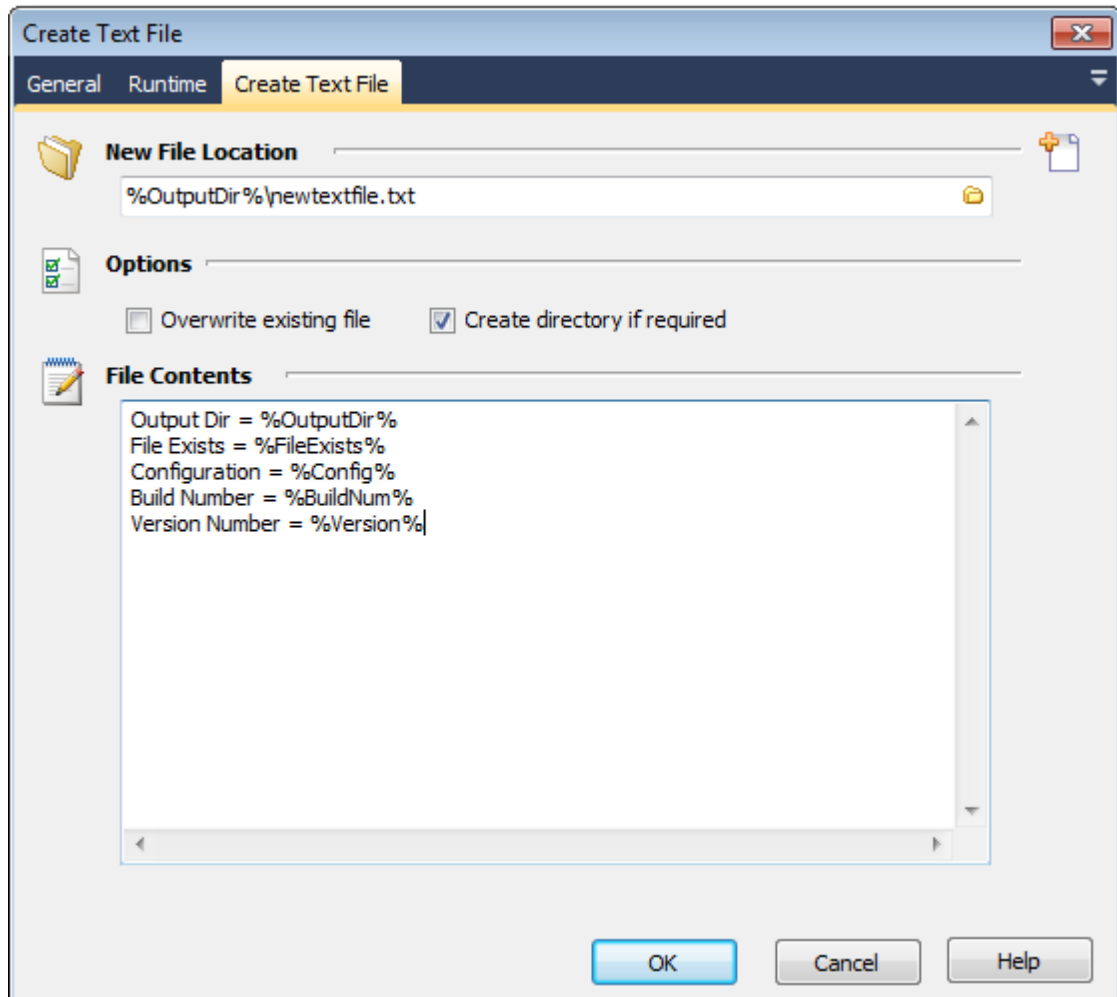
The Action properties available are :

property FileOrDirectory : WideString; // The Directory to Create
property Force : WordBool; // Force the creation of directories.

6.12.9 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**

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

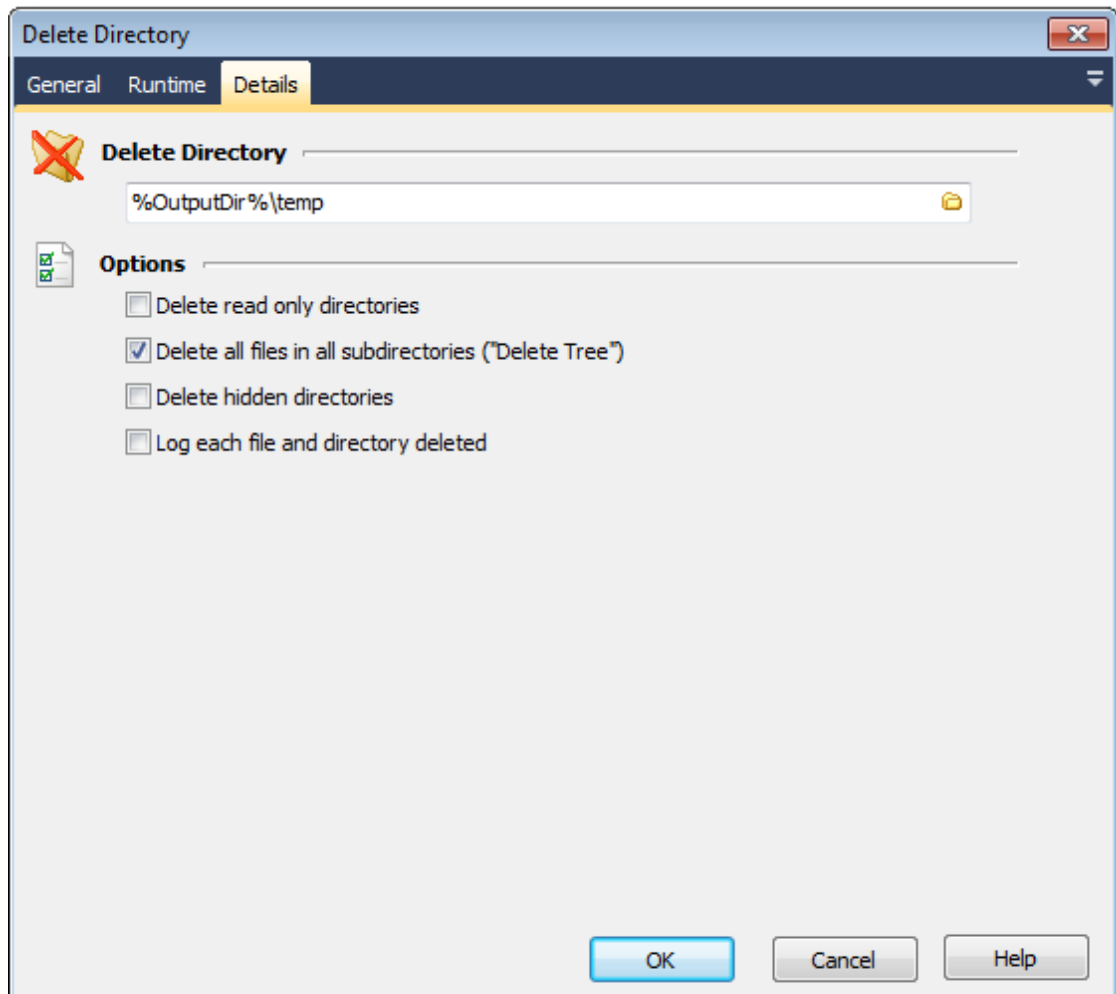
If any directory doesn't exist in the new file location, then it will be automatically created.

File Contents

Specify the text to be written to the file. You can use FinalBuilder variables in the File Contents field to customise the file based on the build.

6.12.10 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 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

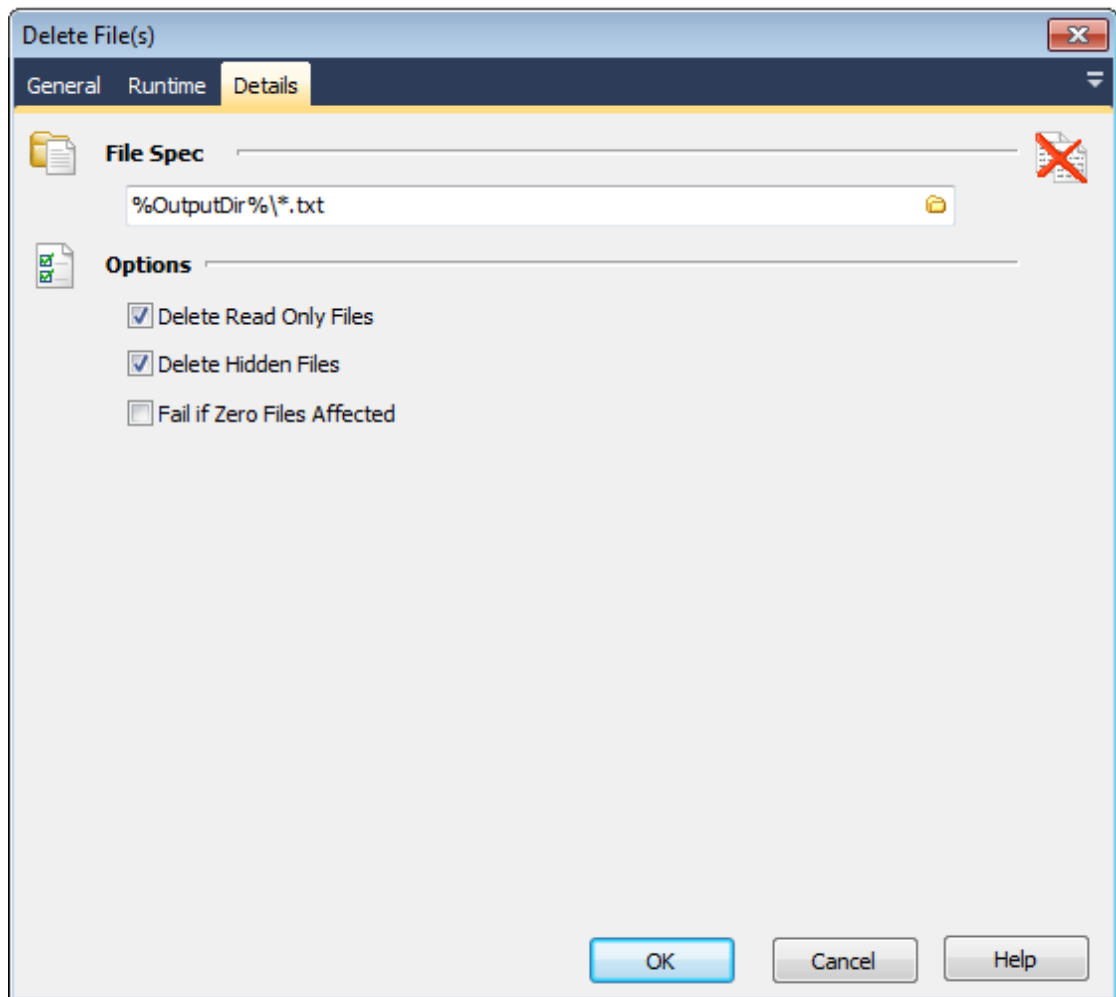
The Action properties available are :

property FileOrDirectory : WideString;// The Directory to Delete
property DeleteReadOnly : WordBool;
property DeleteHidden : WordBool;
property DeleteTree : WordBool;

6.12.11 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.



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.

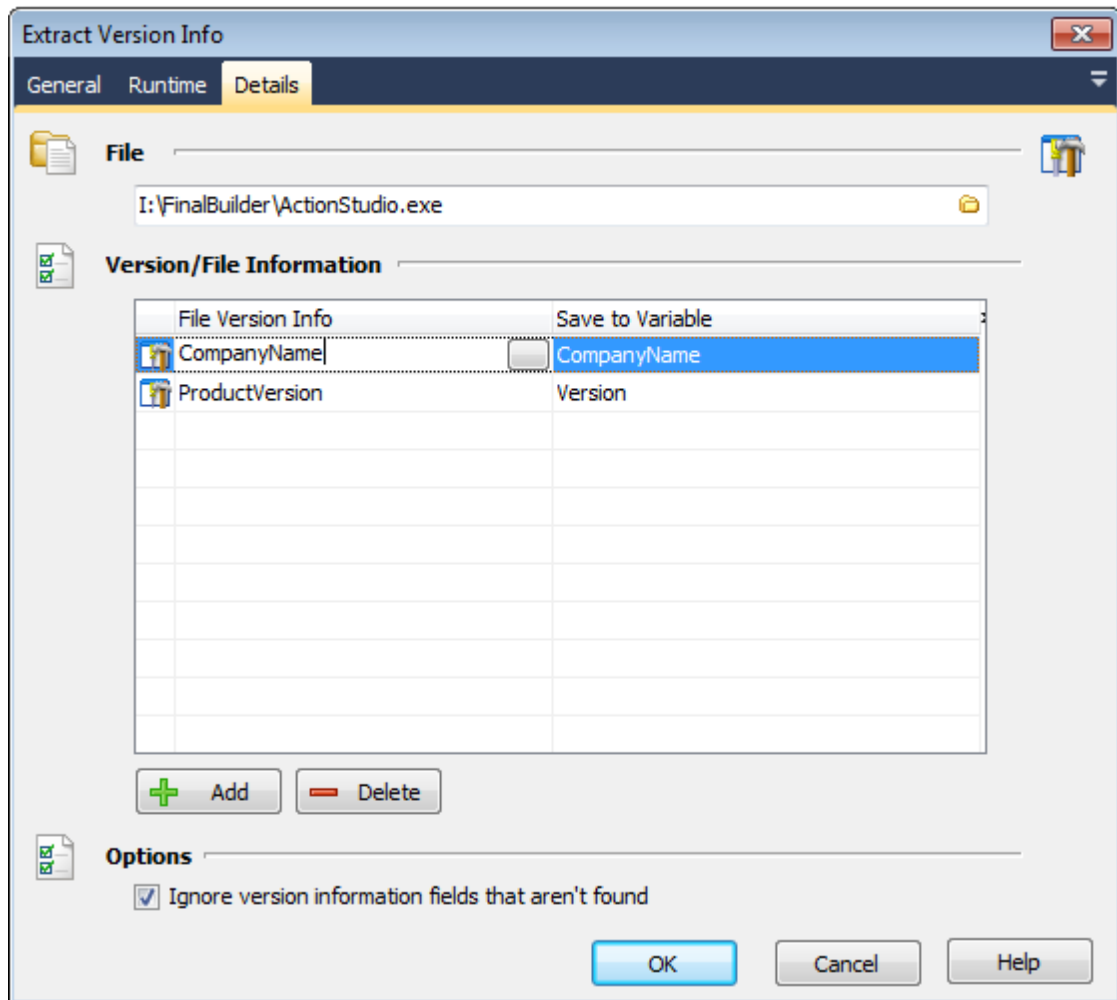
property DeleteHidden : WordBool; //delete hidden files.

property LogFilesDeleted : Boolean; //logs each file that gets deleted

6.12.12 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.

**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 FinalBuilder variable to save the value to.

Ignore Version Information fields that aren't found

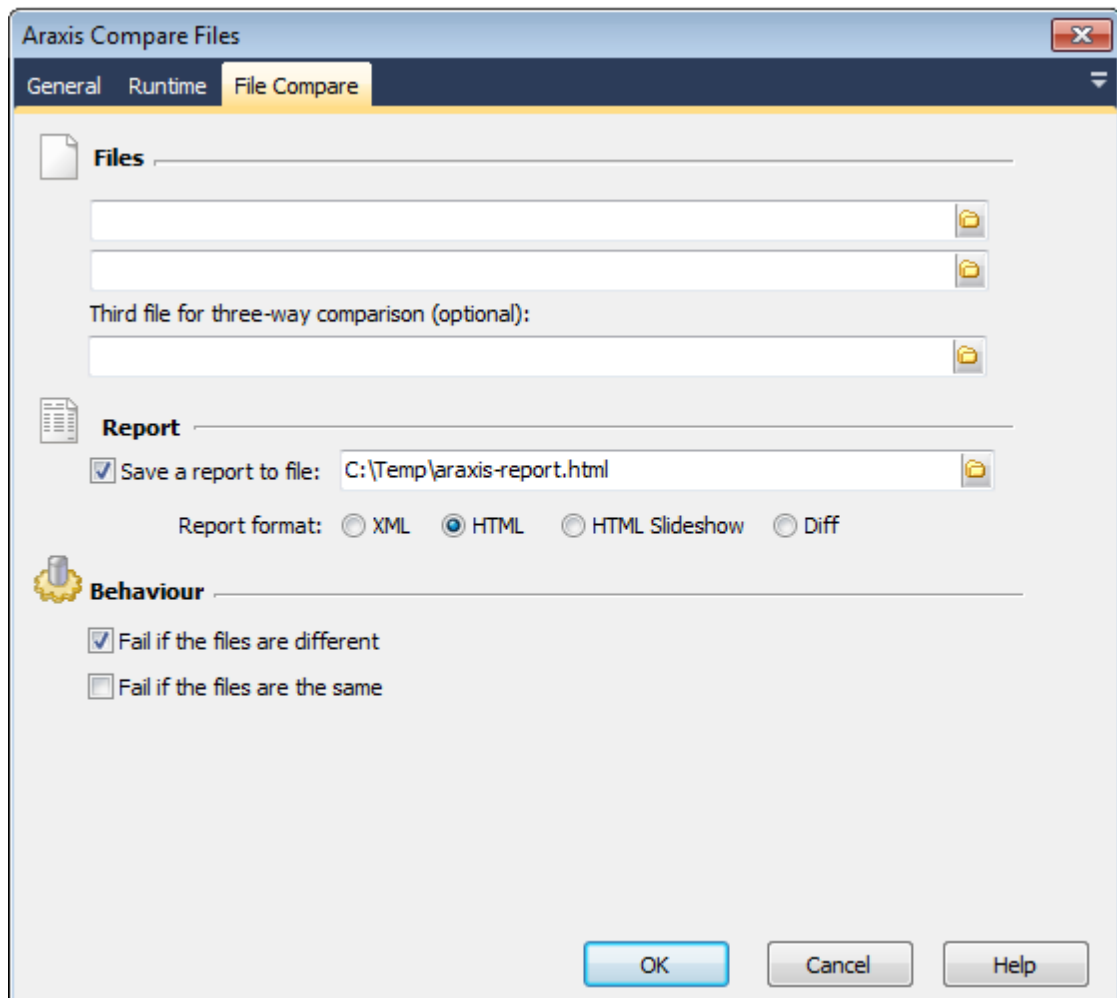
Set this to ignore any fields that are not found. The variables will be set to a blank string.

6.12.13 File Comparison Tools

6.12.13.1 Araxis Merge Compare Actions

FinalBuilder 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.



Files

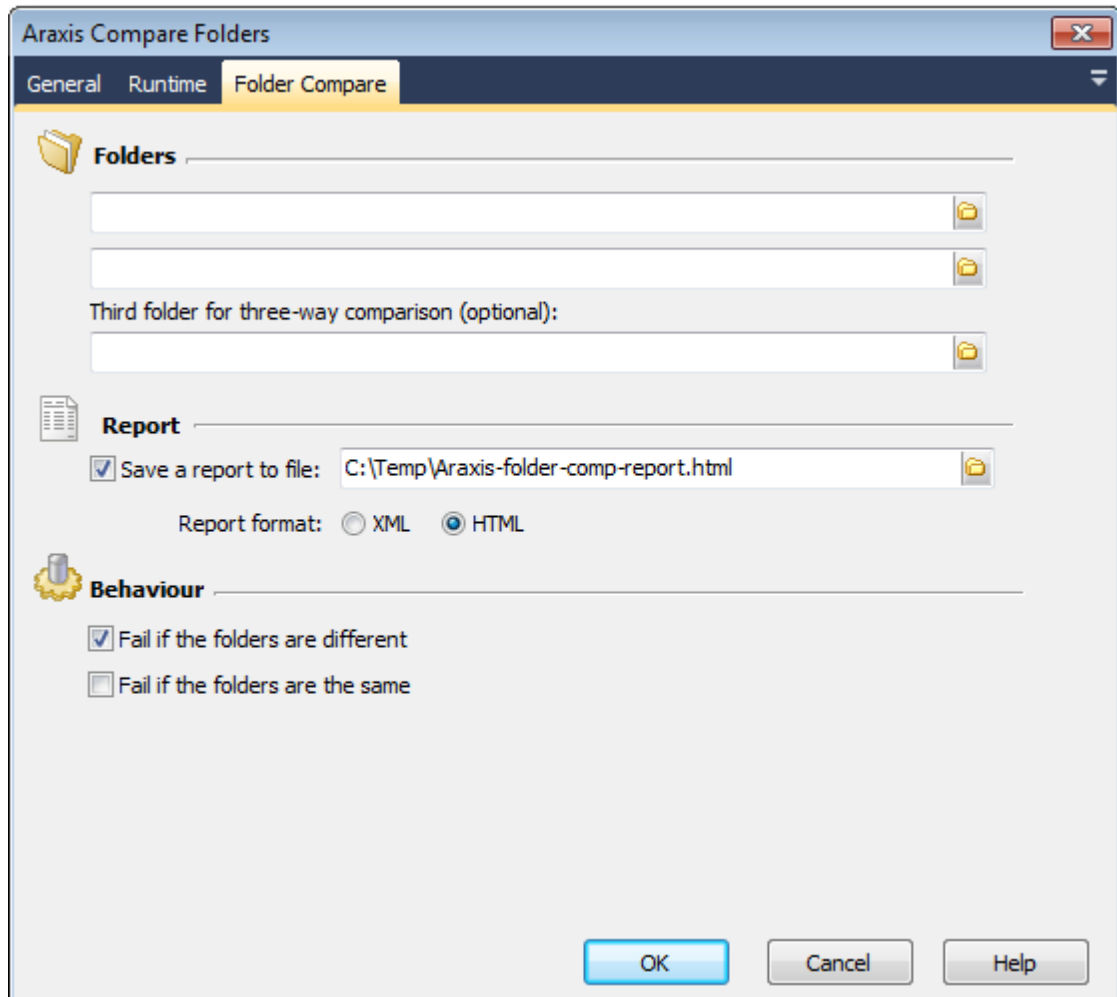
Araxis Merge can compare either two or three files, generating a two- or three-way comparison report.

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.)



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.

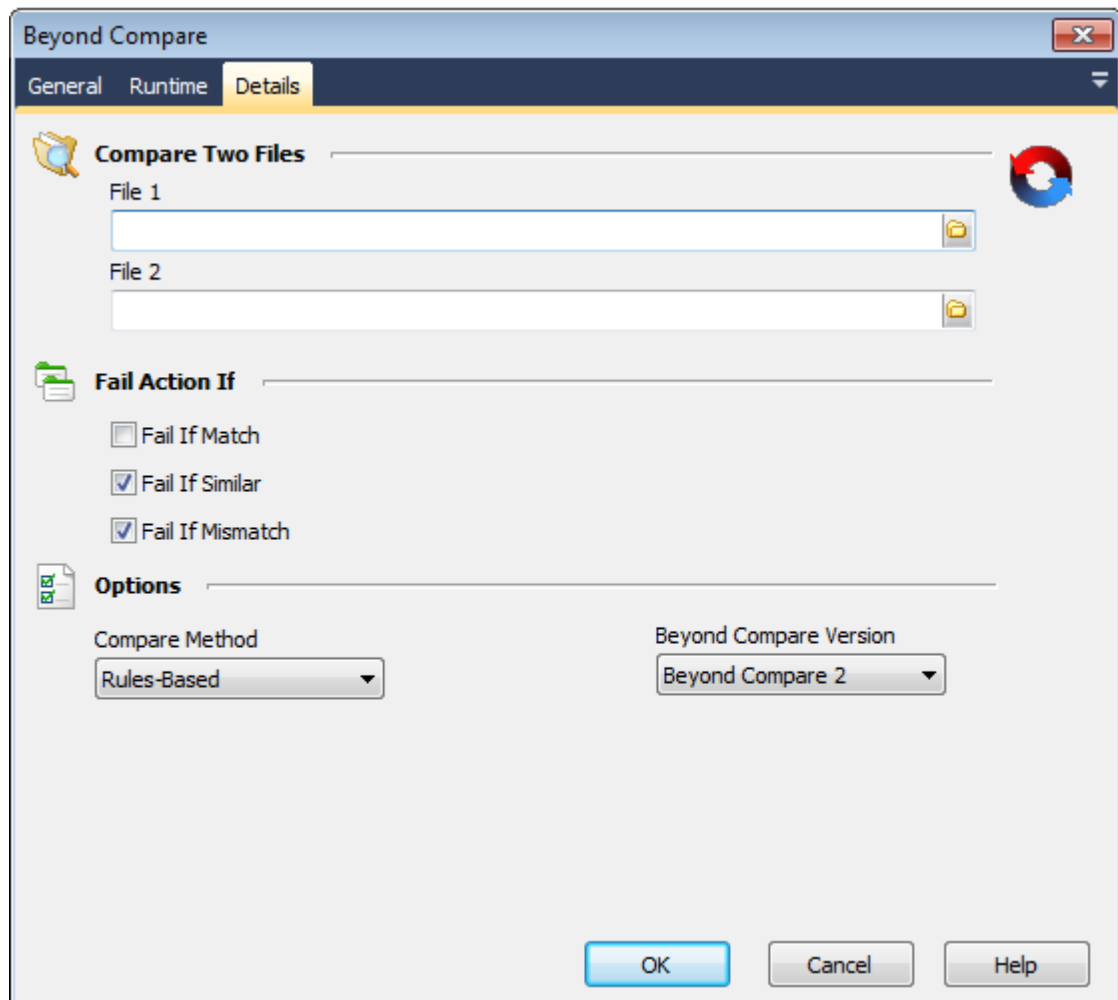
Behaviour

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.)

6.12.13.2 Beyond Compare

6.12.13.2.1 Beyond Compare Action

The Beyond Compare uses the QuickCompare mode of [Scooter Software's Beyond Compare](#) to compare the contents of two files.



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

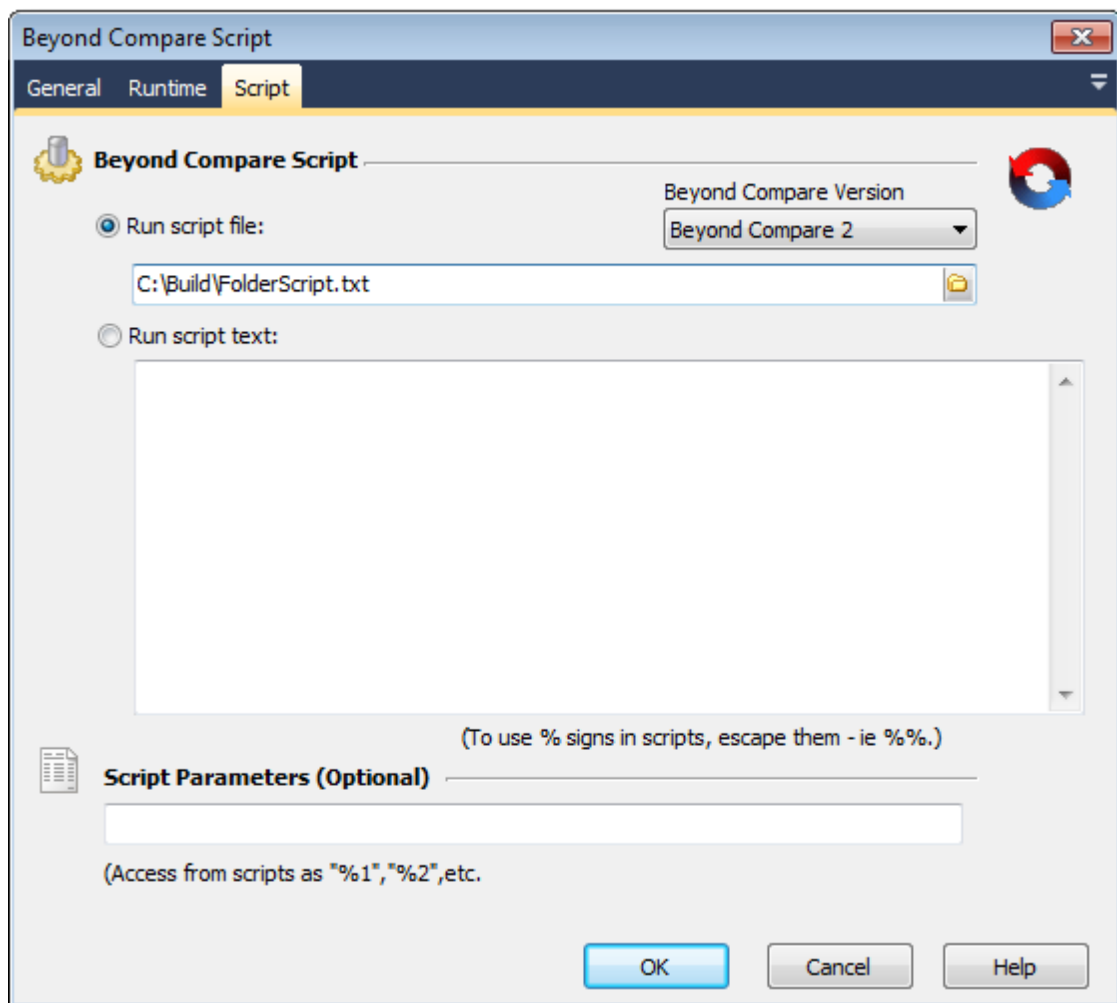
Comparisons can be made based on rules-based, size, CRC or full binary comparison.

See the Beyond Compare help file for the details of each comparison method.

6.12.13.2.2 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.



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 FinalBuilder variables in the script by using the % syntax. If you want to use % signs in the script itself, escape them as %%.

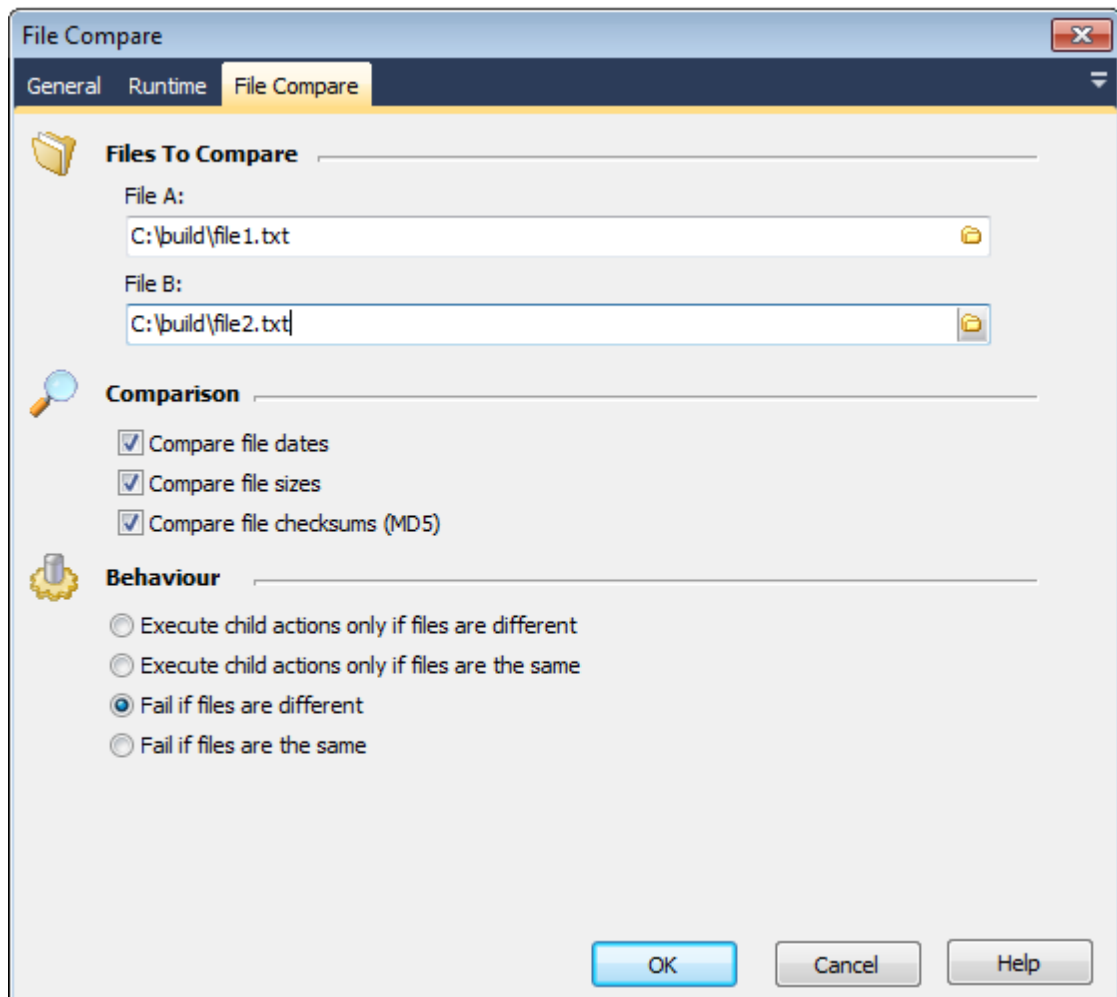
Script Parameters (Optional)

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.)

6.12.13.3 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 or Araxis Merge actions.



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.

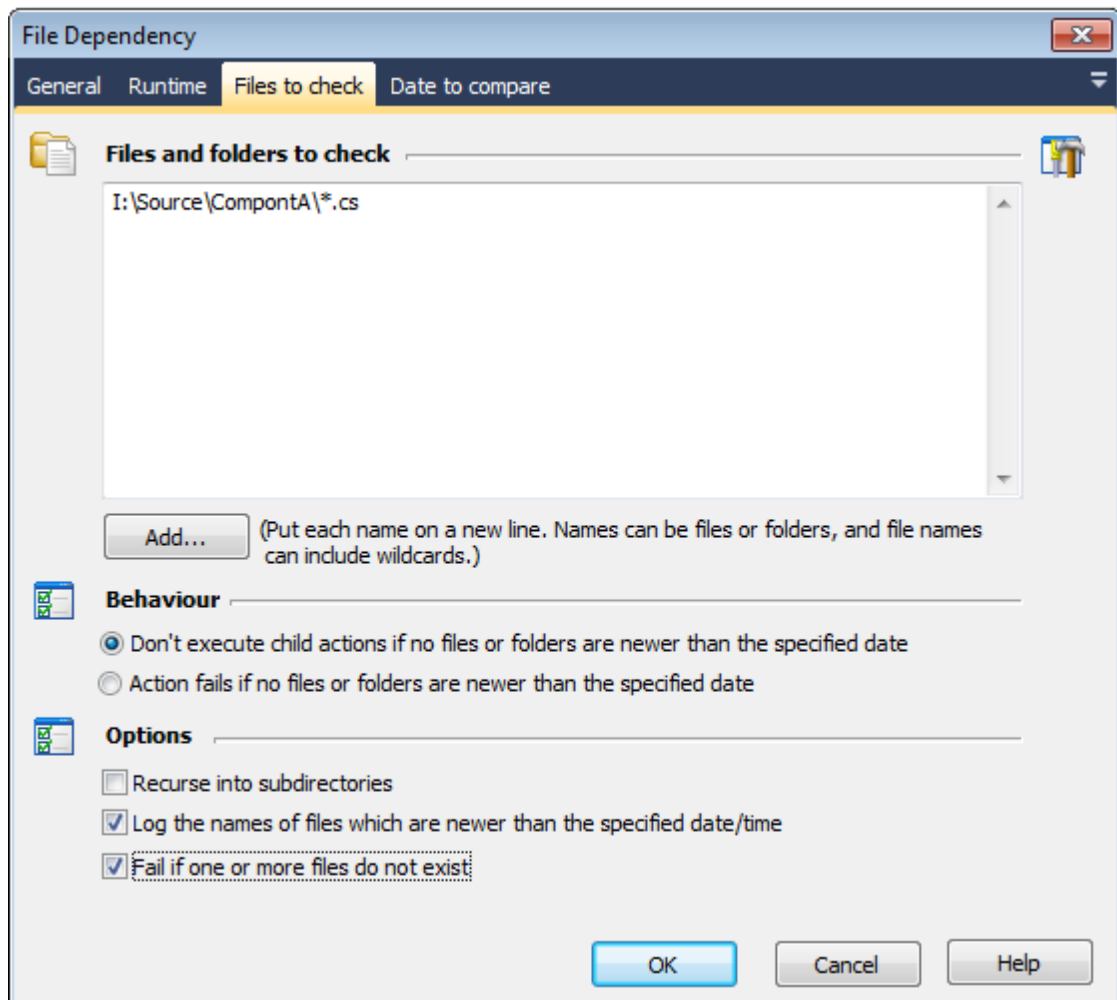
6.12.14 File Dependency Action

[FinalBuilder 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

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, FinalBuilder 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 FinalBuilder 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.

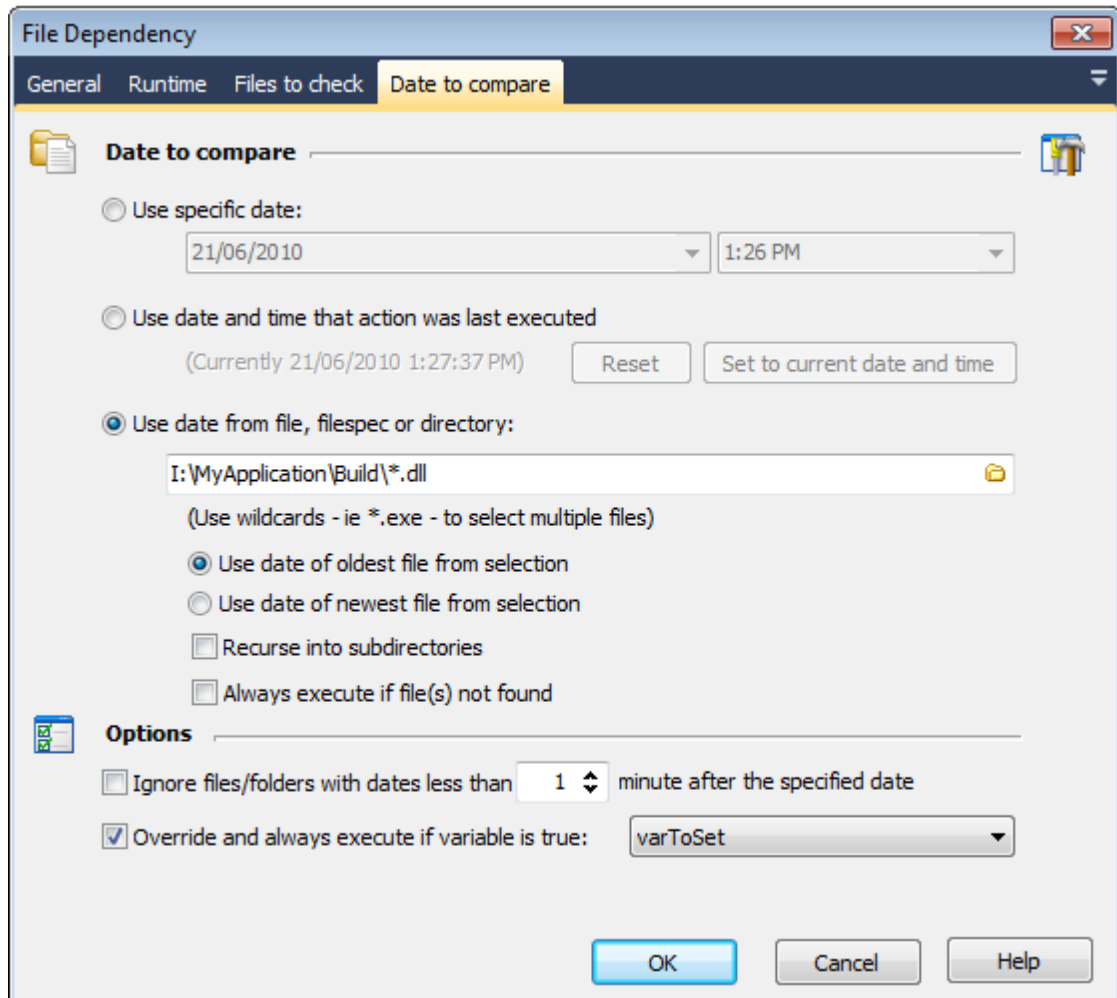
"Log the names of files which are newer than the specified date/time"

If you choose this option, then FinalBuilder 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.



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

FinalBuilder 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 FinalBuilder 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.)

6.12.15 Find File(s)

The Find File(s) action enables you to search for files matching a specified search criteria.

The screenshot shows the 'Find File(s)' dialog box with the 'Search Options' tab selected. The dialog has four tabs: 'General', 'Runtime', 'Search Options', and 'Behaviour'. The 'Search Options' tab contains the following sections:

- File Location:** Includes a 'File Name' field with the wildcard '*.*', a 'Search Directory' field with 'W:\', and a 'Recursive' checkbox which is unchecked.
- Time Stamps:** Includes a clock icon, a dropdown menu, and two date/time fields. The first field shows '30/12/1899' and the second shows '12:00 AM'.
- Size:** Includes a folder icon, a 'Greater than:' field with '1024' KB, and a 'Less than:' field with '0' KB. Below these fields, it says 'Search for files with a size greater than 1024KB'.
- Attributes:** Includes a folder icon and six checkboxes: 'Read Only', 'System', 'Encrypted', 'Archive', 'Hidden', and 'Compressed'. All checkboxes are unchecked.

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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

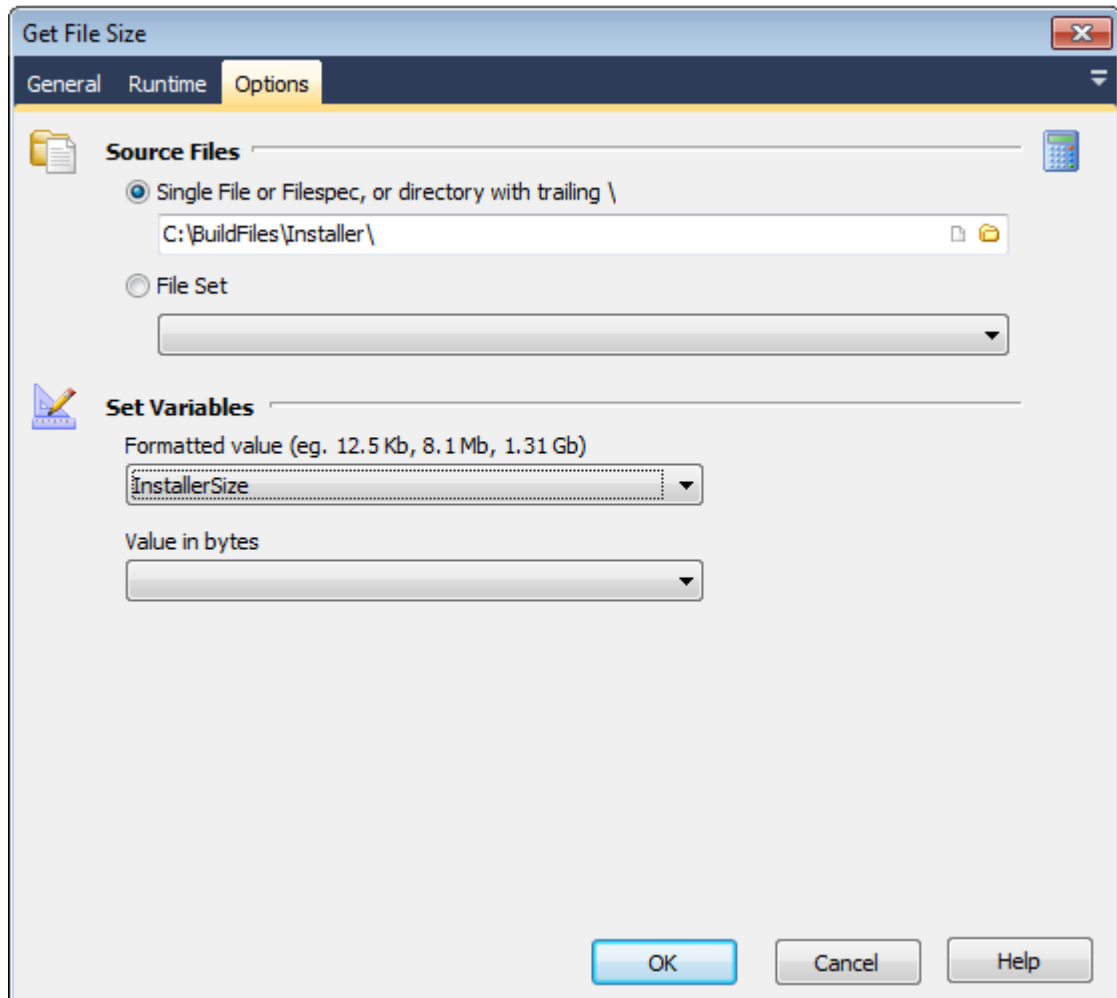
Allows file to be found based on the attributes set.

Fail Action If

Alters the behaviour of the action regarding the success or failure when files are found.

6.12.16 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 - 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

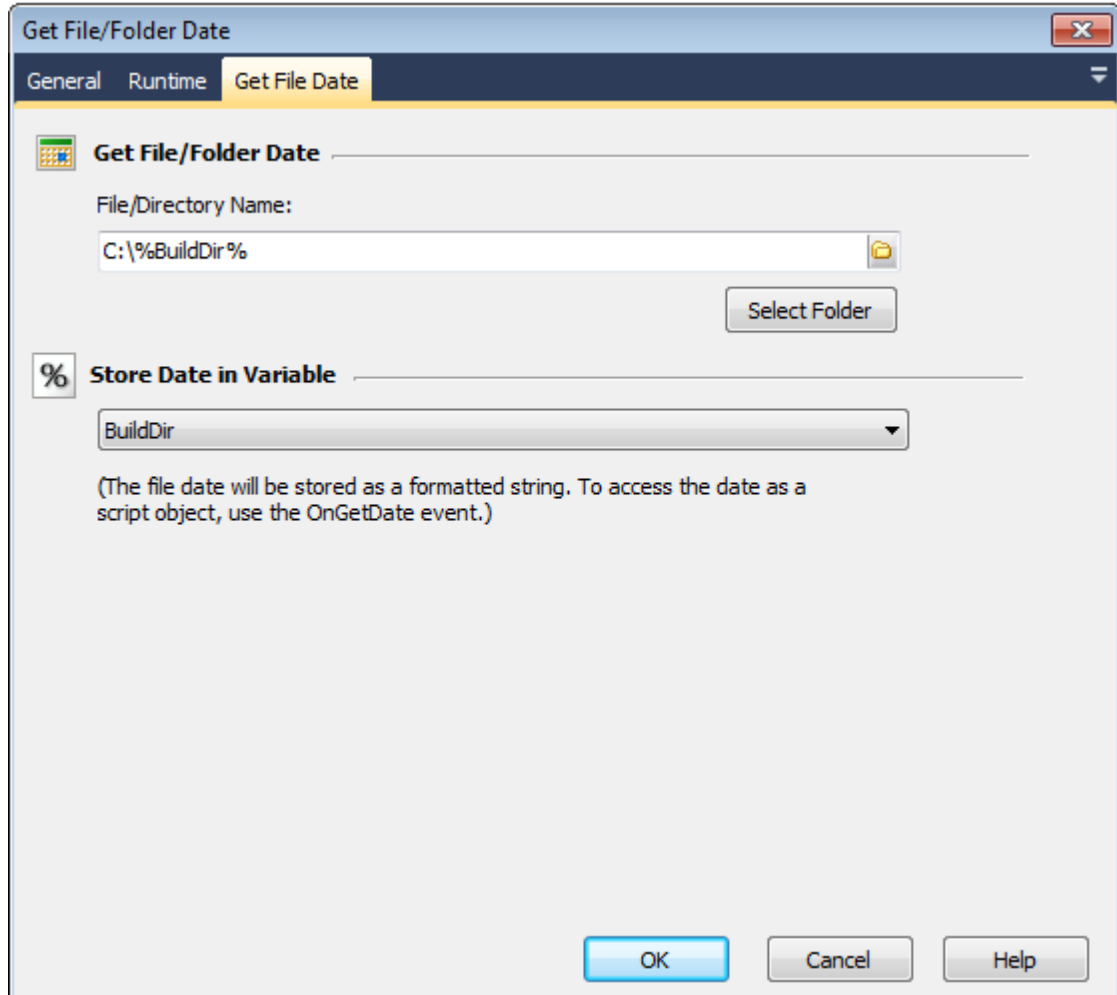
Set Variables

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

6.12.17 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

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

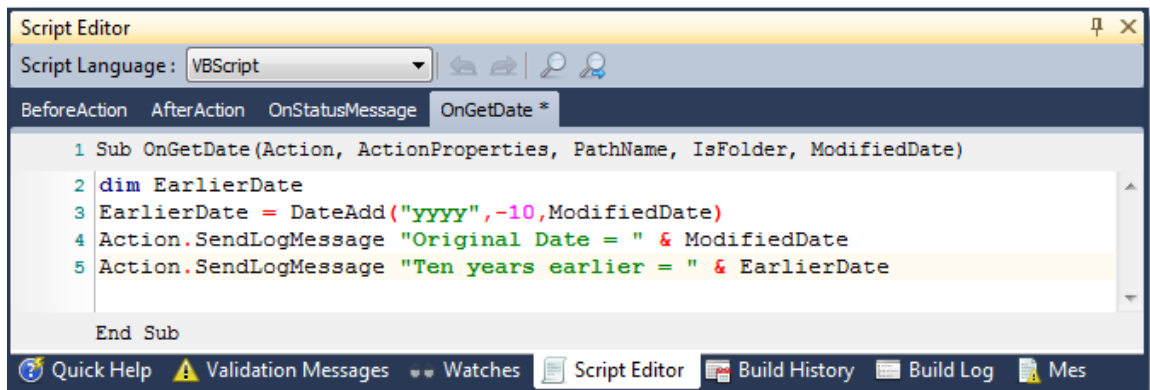
Optionally, choose a variable to store the date in. The date will be stored as a formatted date/time string.

Modification Date As Object

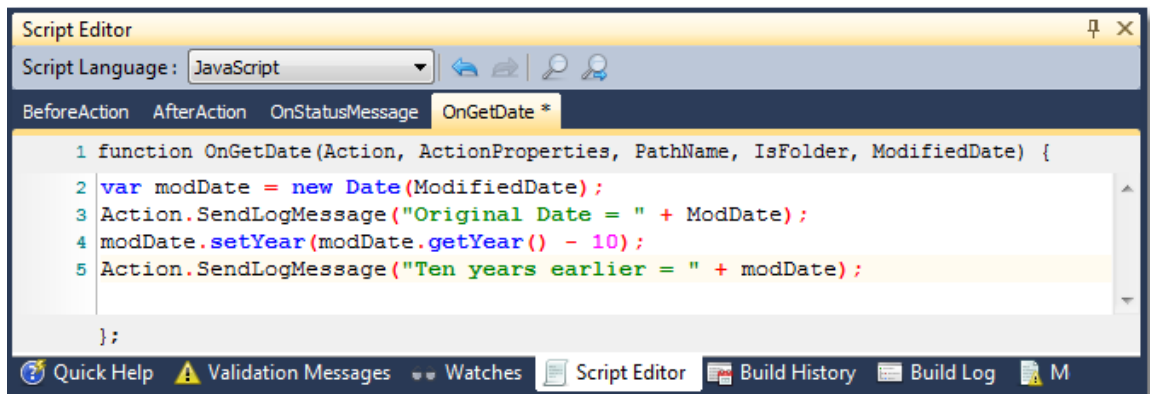
If you wish to access the date as an object, you can use the OnGetDate script event.

For VBScript, you can use Date functions directly to operate on the ModifiedDate

parameter:



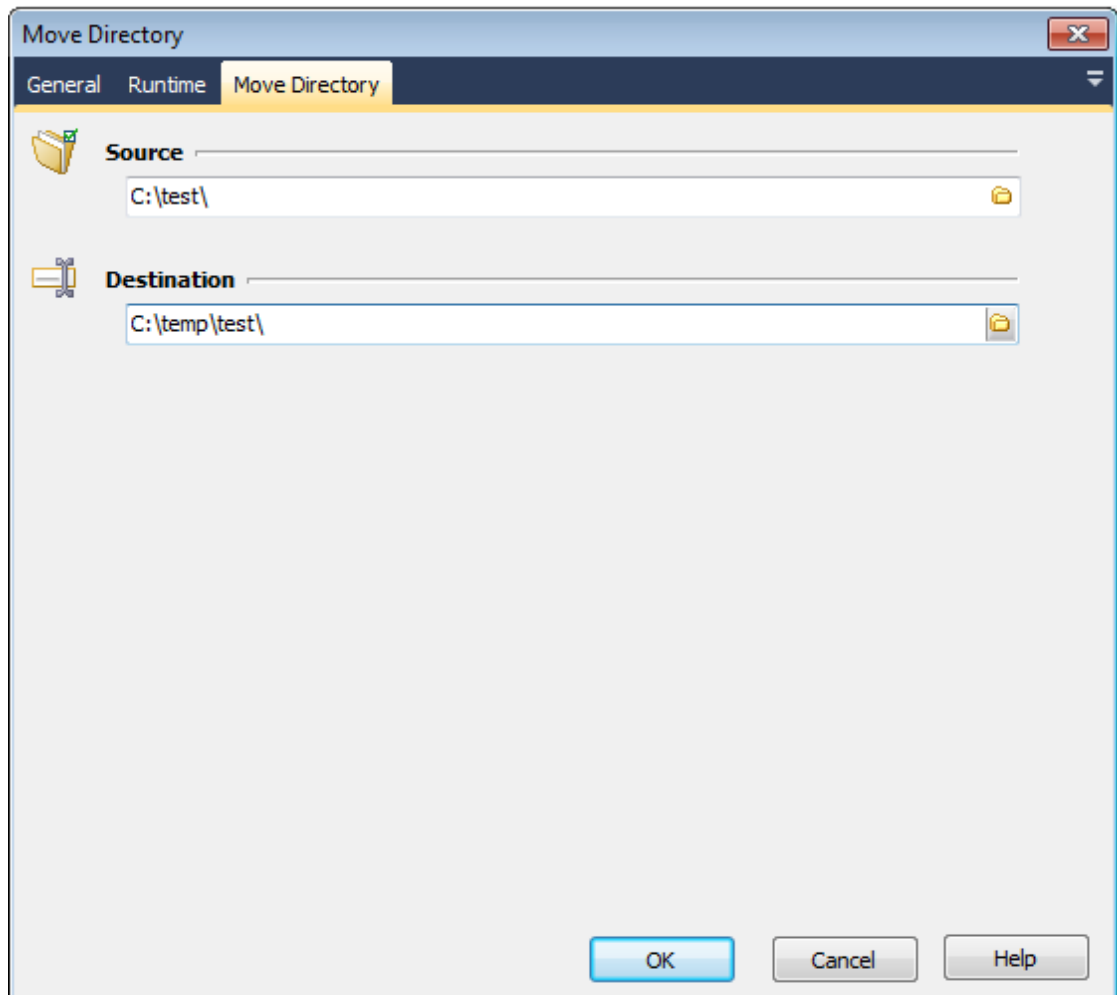
For JavaScript, you need to create a new JavaScript Date object to wrap the ModifiedDate parameter:



6.12.18 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.



Scripting Info

The Action properties available are:

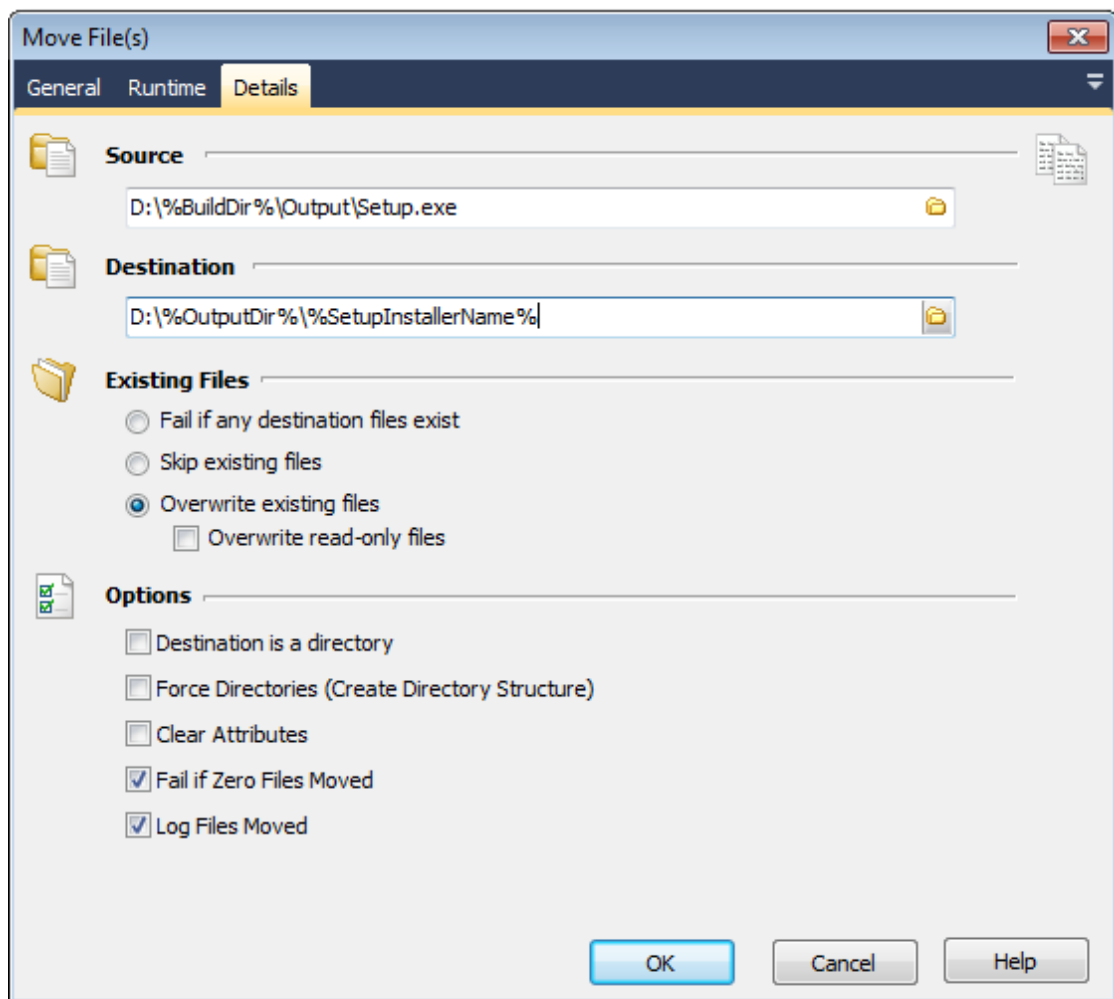
- property** Directory : String; // the directory you wish to move.
- property** DestinationDirectory : String; // the new path for the directory.

6.12.19 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 FinalBuilder 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.



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.

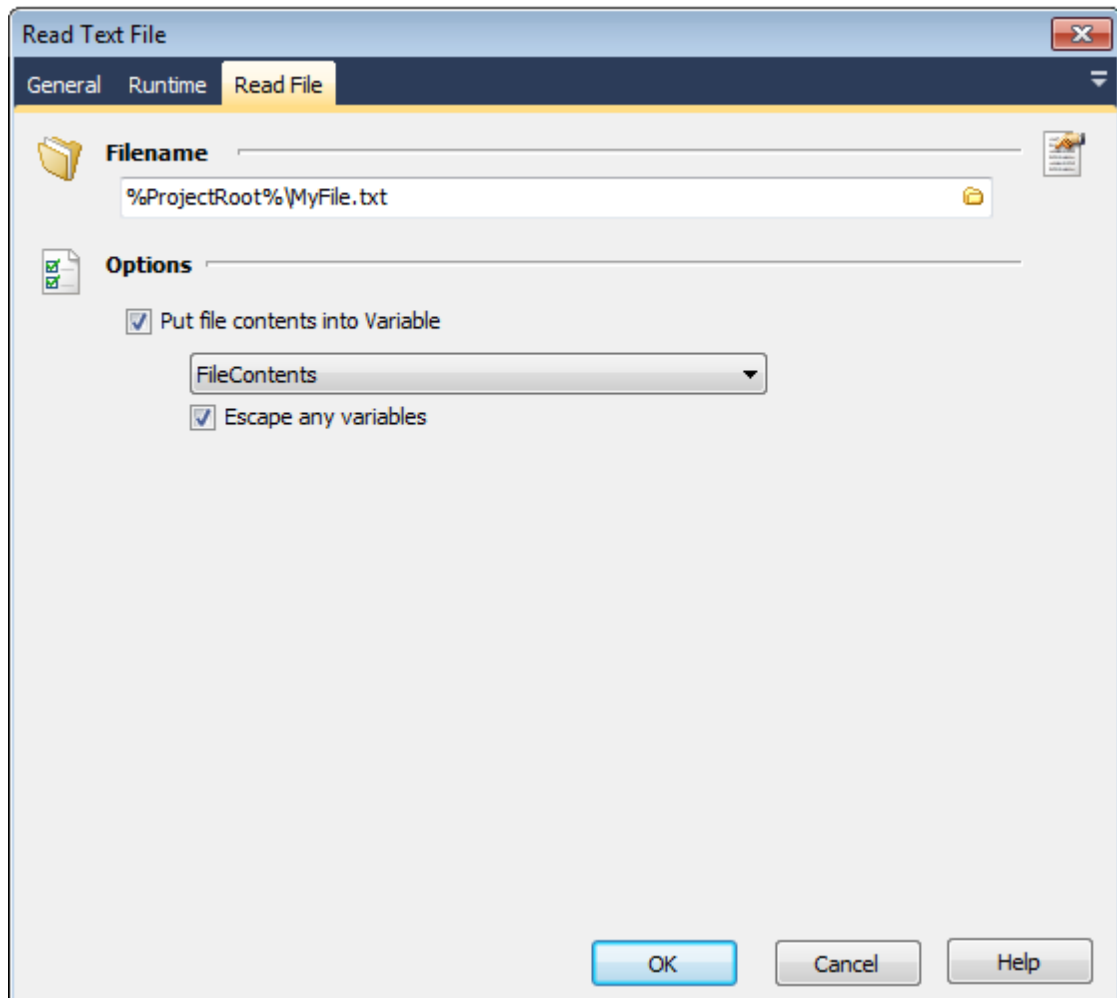
property FailIfZeroFiles : WordBool;

property LogFilesAffected : WordBool;

6.12.20 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 FinalBuilder variable, and also process each line in the OnReadLine event.



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

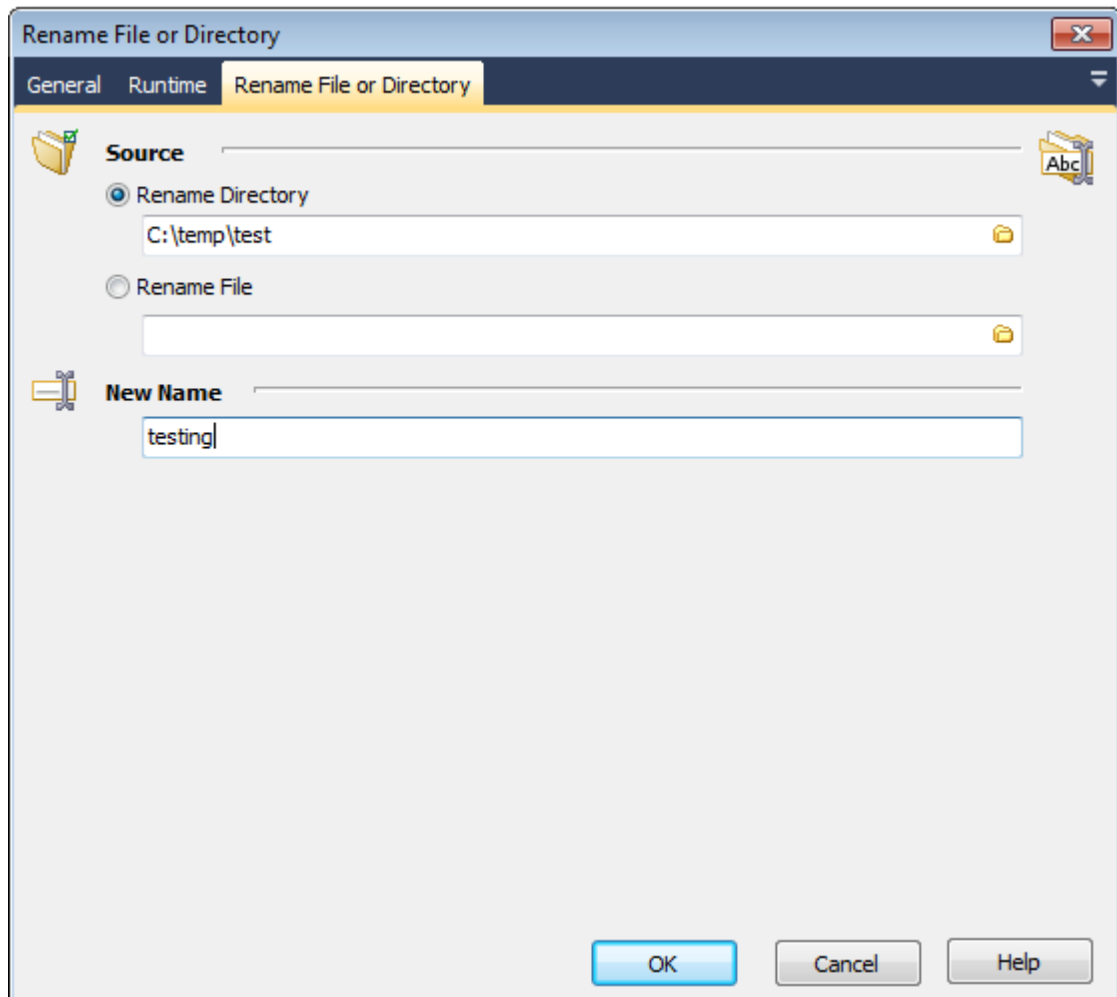
Specify this option to escape any variable references in the source file.

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](#)

6.12.21 Rename File or Directory Action

This action allows you to rename a File or a Directory.



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)

6.12.22 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.

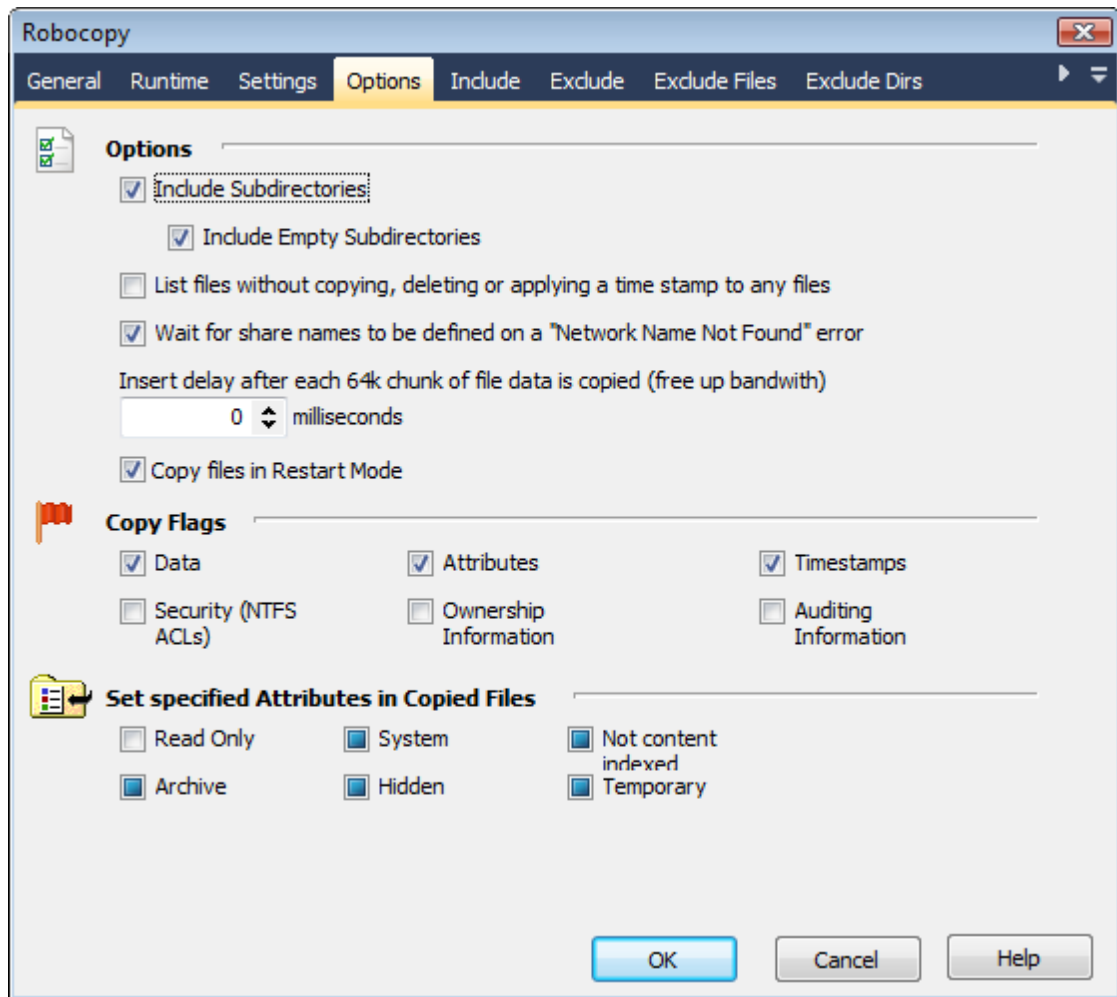
6.12.22.1 Robocopy

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 - 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.

6.12.22.2 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

6.12.22.3 Robocopy Move

The Robocopy Move action allows you to move files and directories using Microsoft's Robocopy.

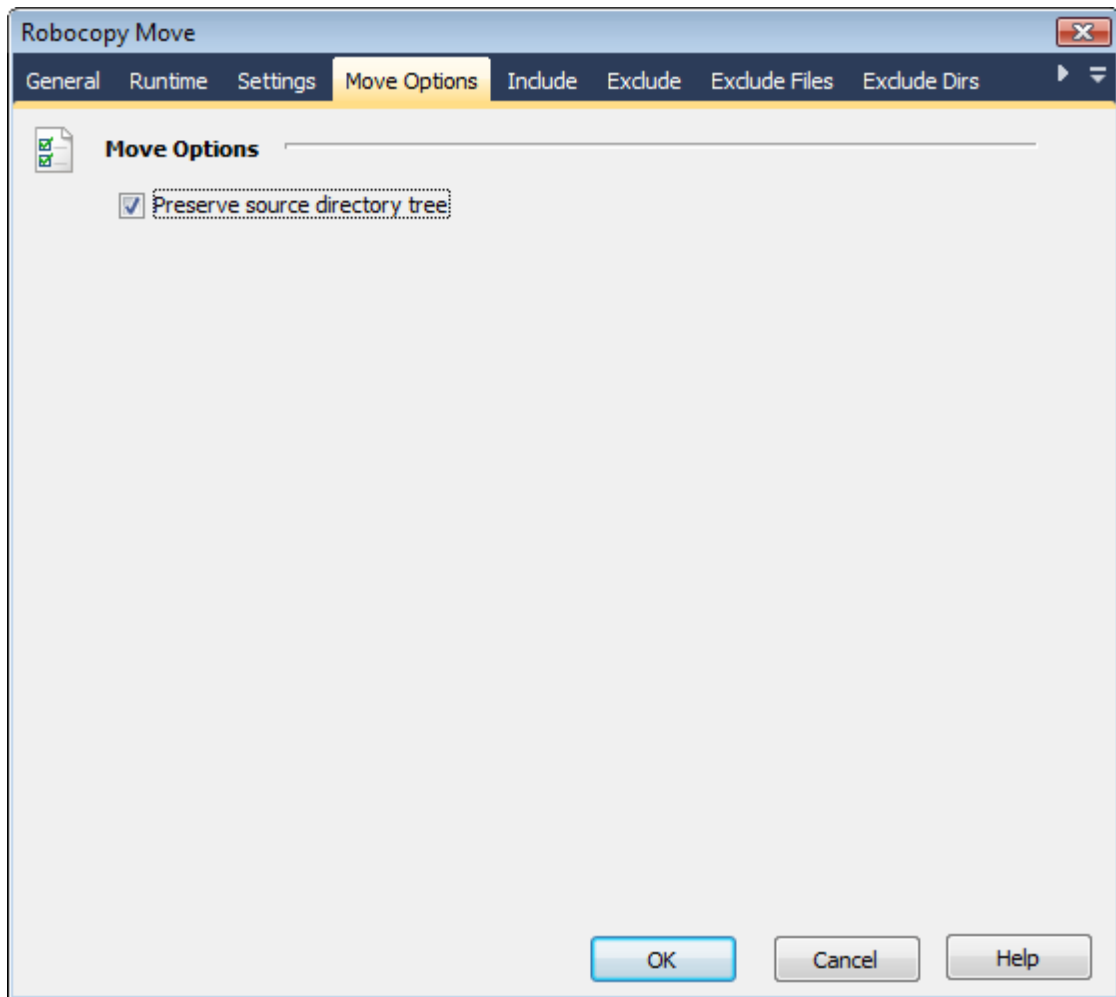
This action allows you to move the contents of one directory to another.

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.



6.12.22.4 Run Robocopy Job

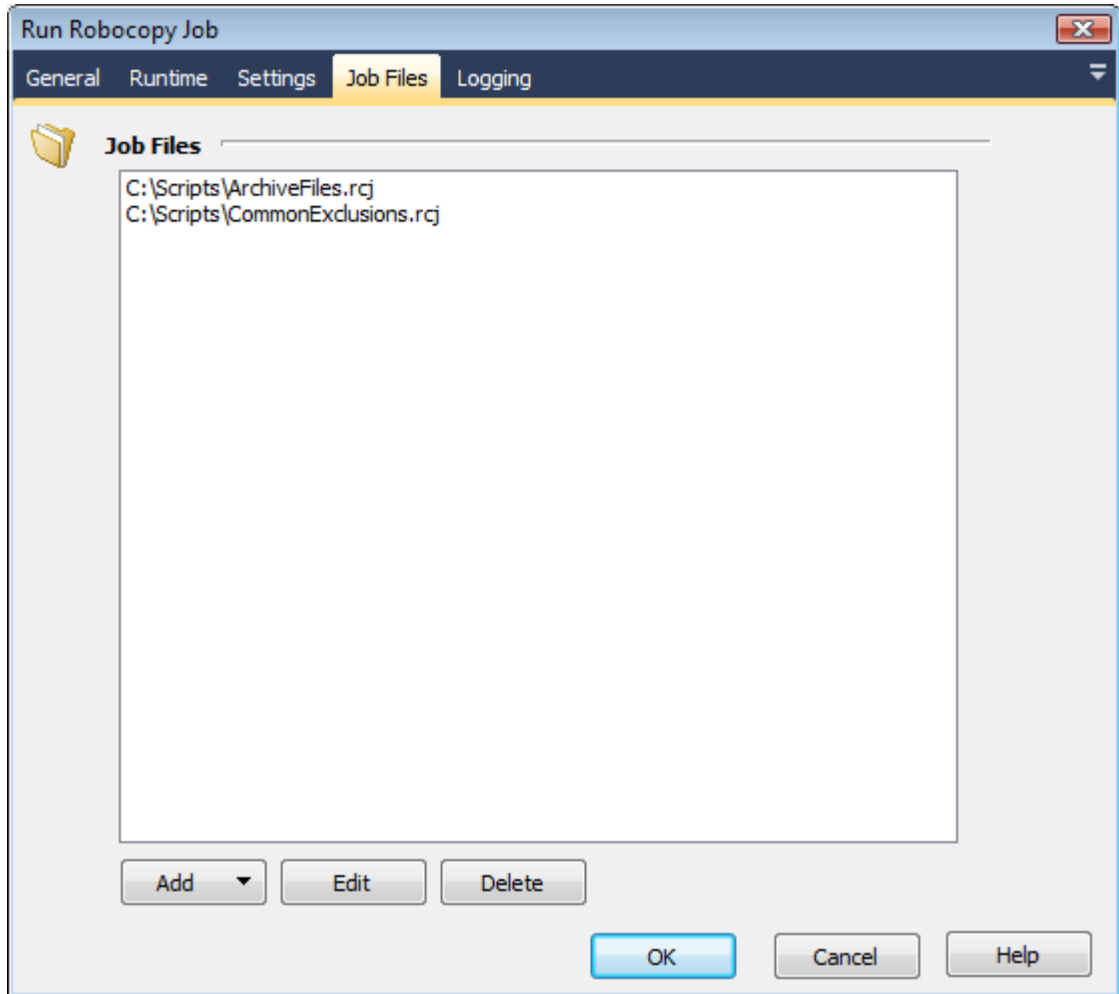
The Run Robocopy Job action allows you to pass in Robocopy Job Files (.rcj) that contain the parameters for the Robocopy operation.

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.

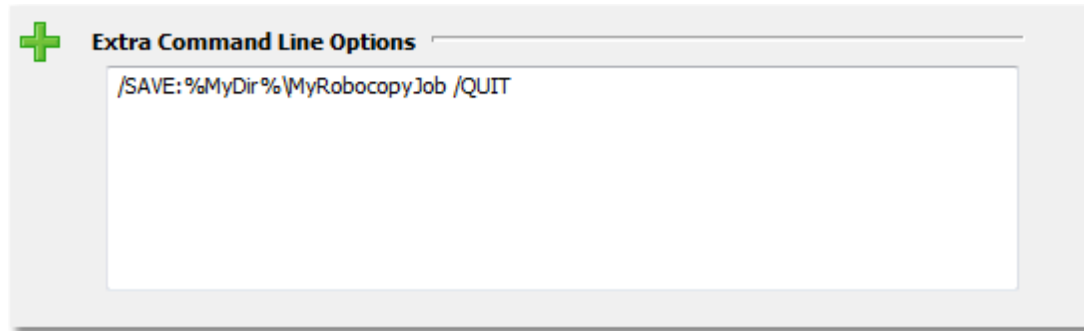


About Robocopy Jobs

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 precede 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 FinalBuilder by using the desired action and adding the `/SAVE:<filename>` and `/QUIT` switches in the extra command line arguments field.

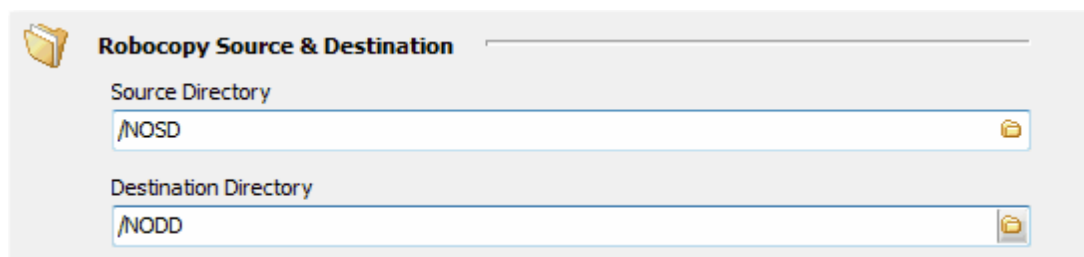


Creating a Robocopy job file from a FinalBuilder action.

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 operation 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).



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 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

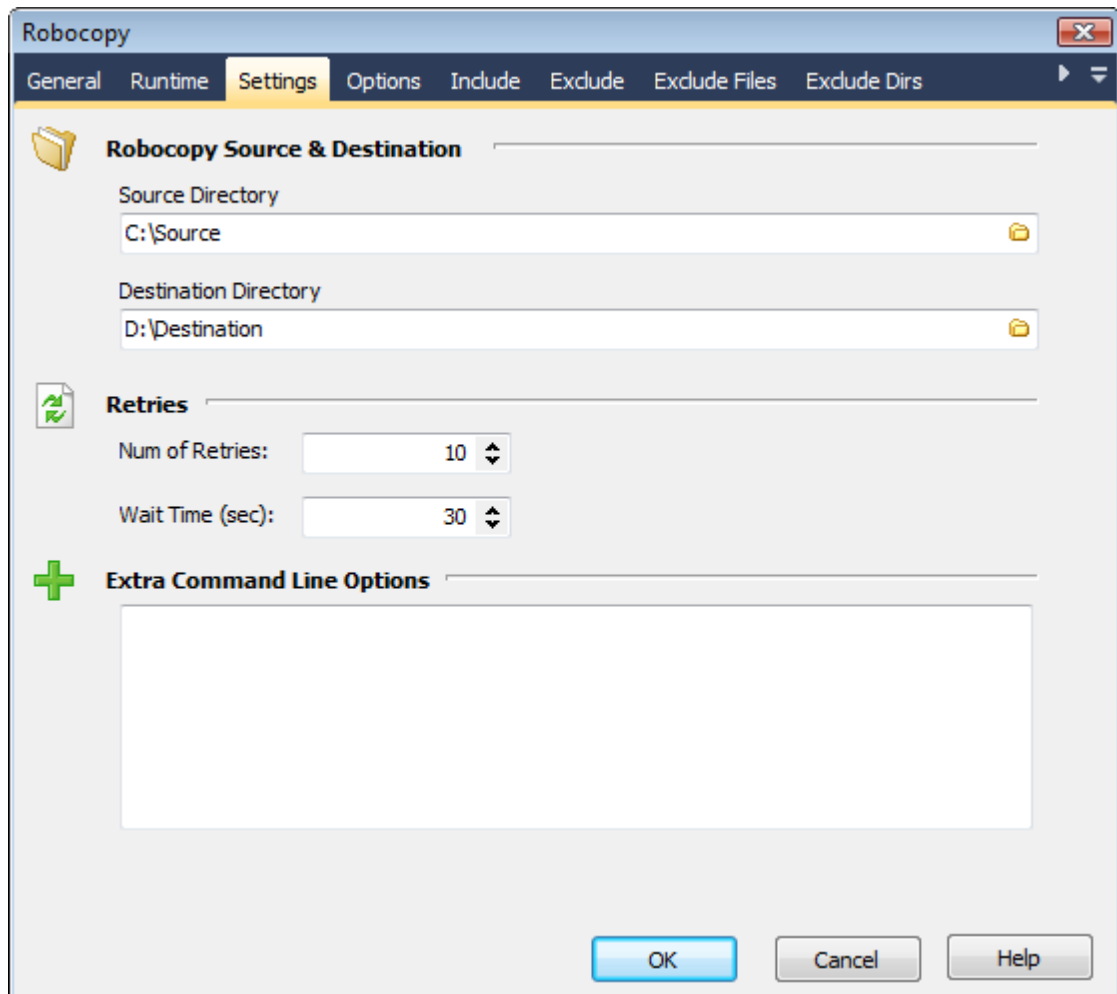
:: Excluded Files
/XF
    Readme.txt  :: Any file called 'Readme.txt' will not be included
in the operation
    *.tmp      :: Any file with the extension .tmp will be excluded
```

6.12.22.5 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 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.

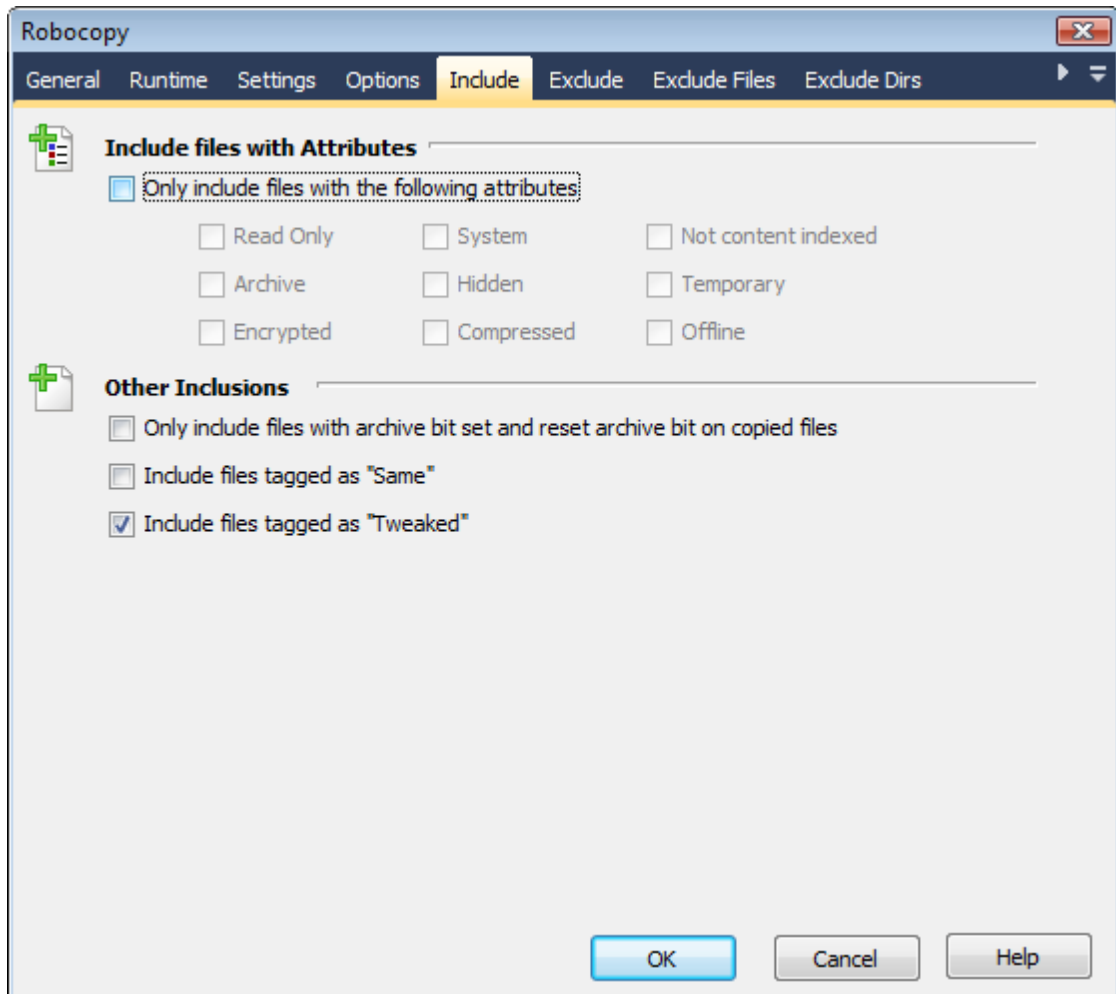
Wait Time (sec) - This allows you to specify the number of seconds between retries of failed copy attempts.

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 - 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

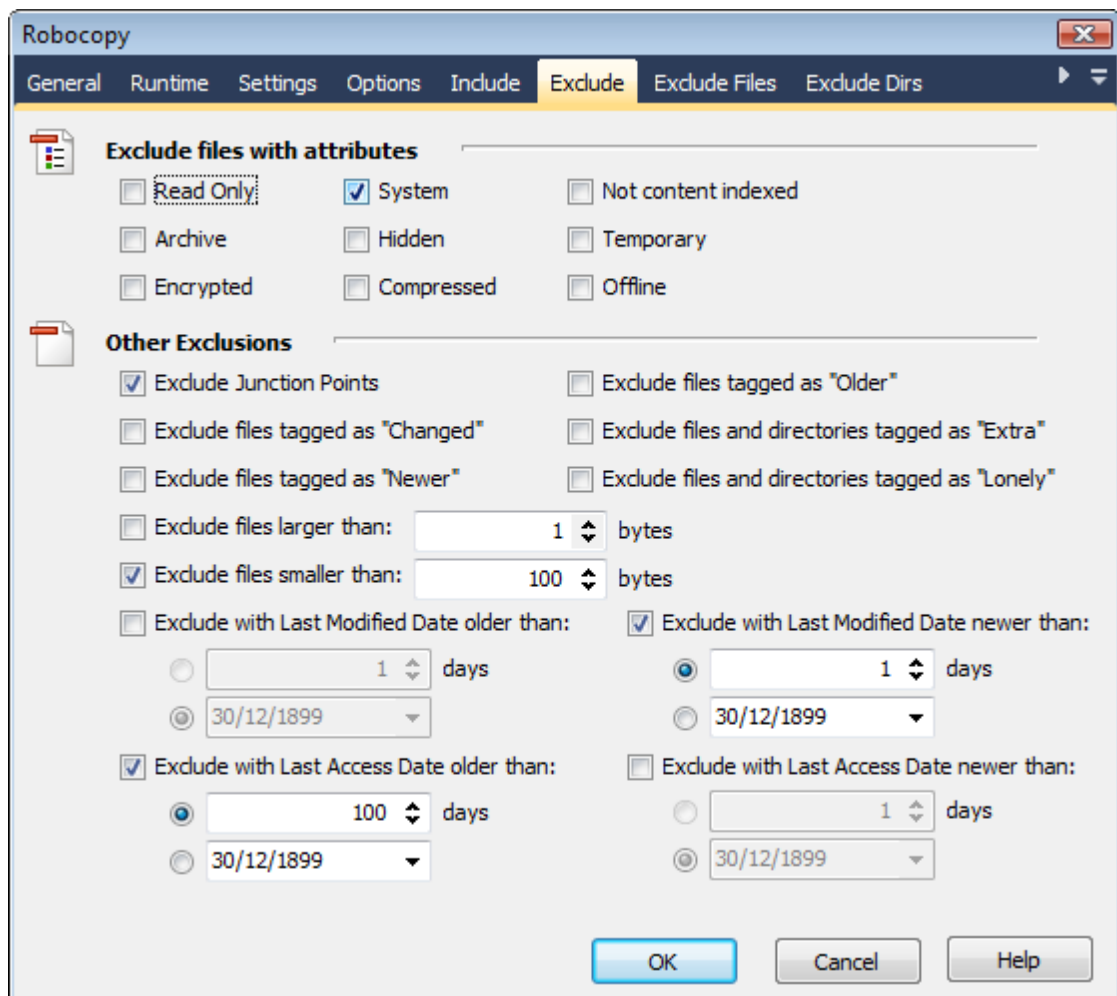
- Only source files that have the archive bit set will be copied. Once the copy has completed the archive bit will be reset on these files.

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 exclude files from the Robocopy operation.



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.

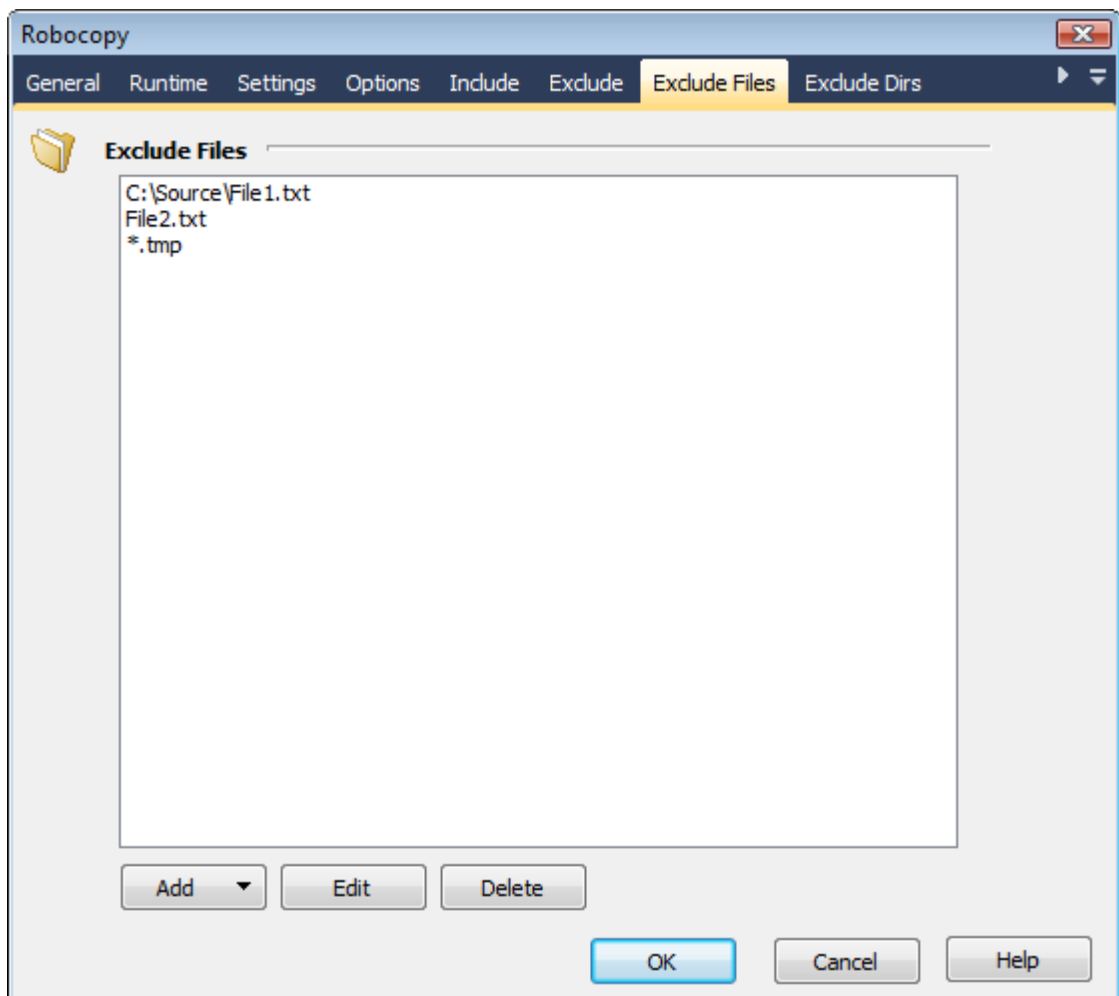
Exclude with Last Modified Date newer than - Self explanatory.

Exclude with Last Access Date older than - Self explanatory.

Exclude with Last Access Date newer than - Self explanatory.

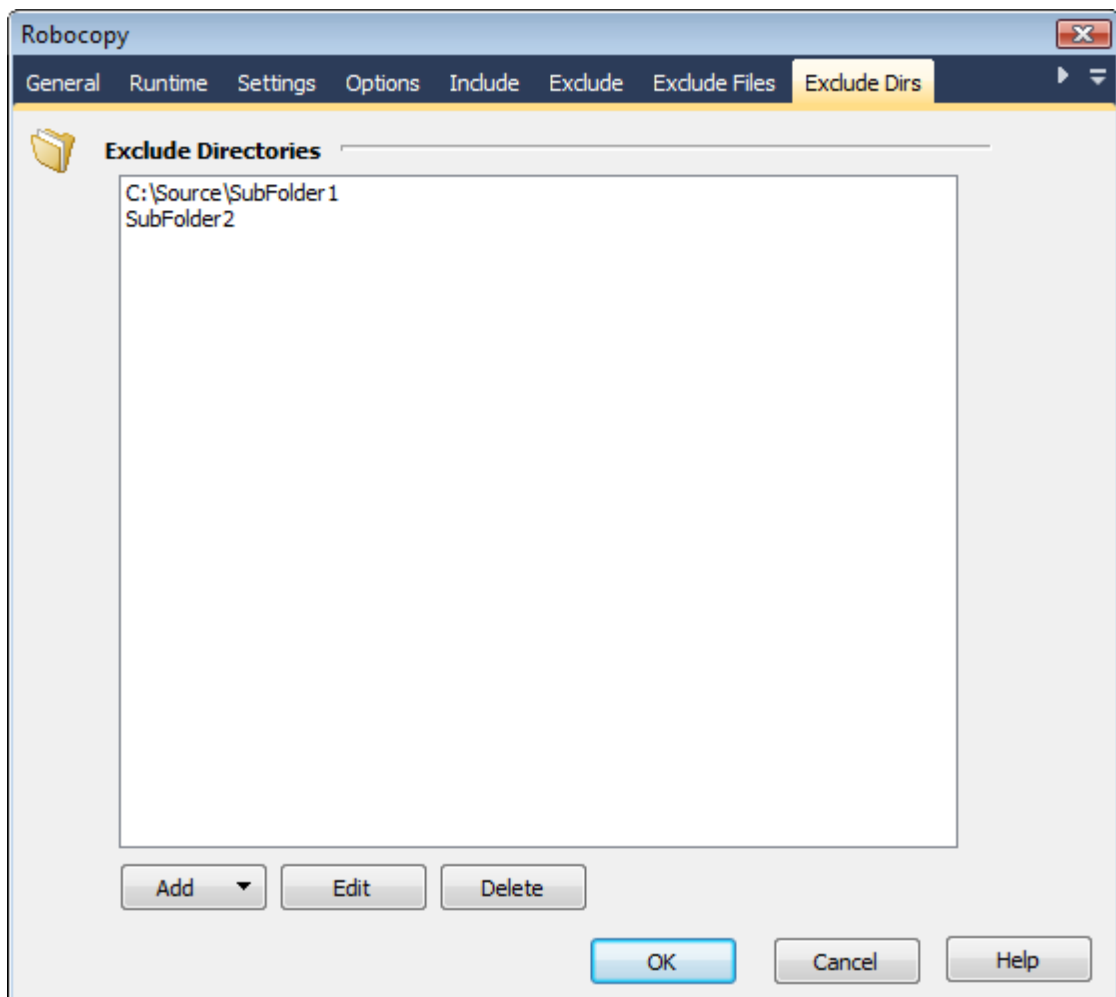
Exclude Files Page

The Exclude Files page allows you to specify file names and file masks to exclude files from the Robocopy operation.



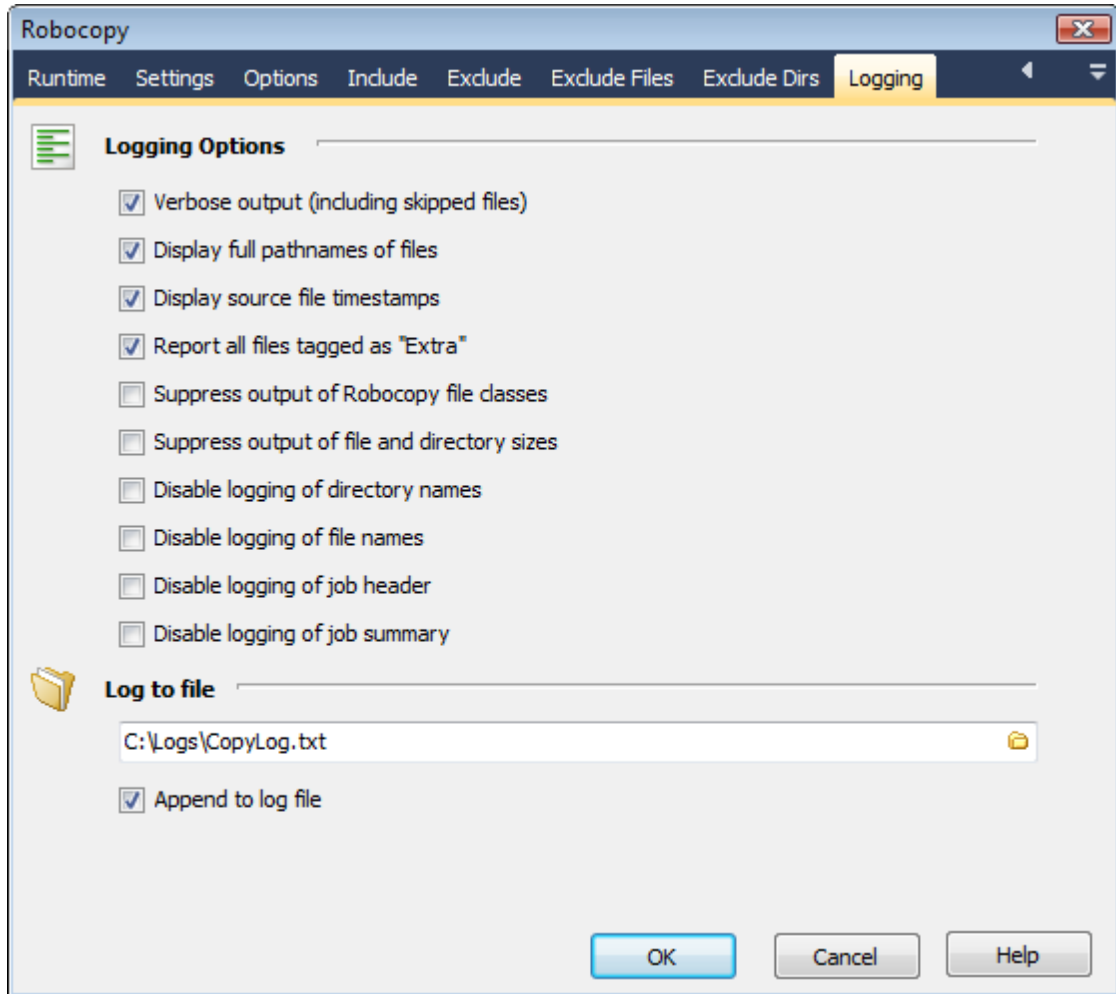
Exclude Dirs Page

The Exclude Dirs page allows you to specify directories to be excluded from the Robocopy operation.



Logging Page

The logging page provides options to customise the output from Robocopy.



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.

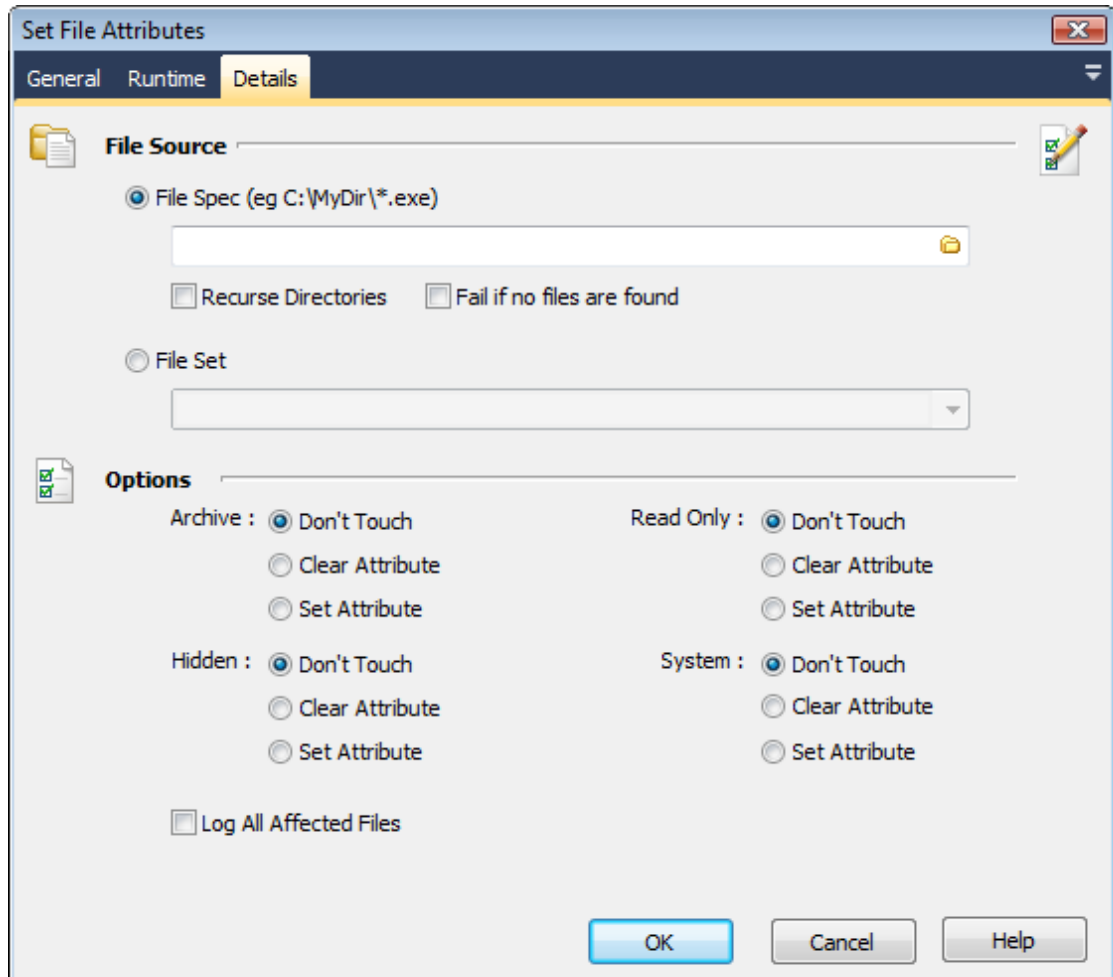
Disable logging of job summary - Self explanatory.

Log To File - Provide a file path to log Robocopy output to a file as well as a the FinalBuilder log.

Append to log file - Append to the file specified rather than overwriting it each time the action is run.

6.12.23 Set File Attributes Action

This action enables you to set a file or a set of file's attributes.



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 FinalBuilder Output Tree.

6.12.24 Signtool Actions

The Signtool actions allow you to automate digitally signing files, applying a timestamp to signed files and verifying signatures of signed files.

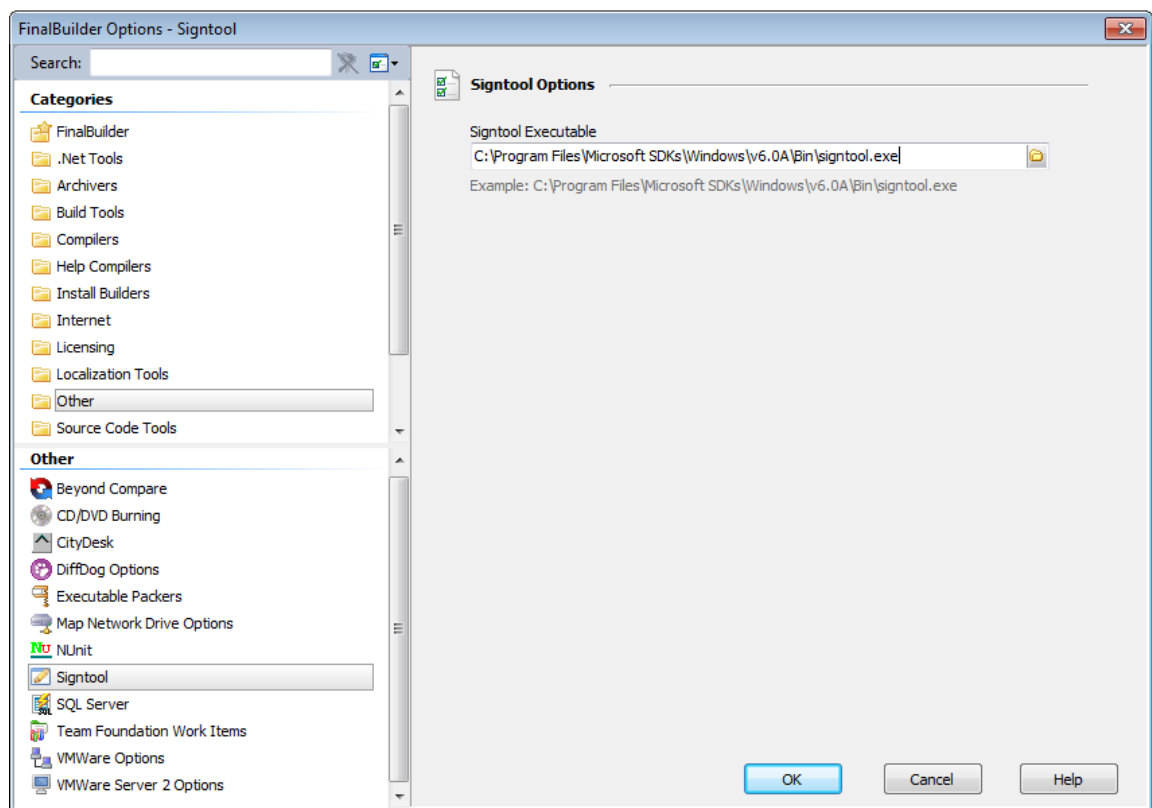
These actions use the Signtool executable (signtool.exe) which is located in the \Bin directory of the Microsoft Windows SDK.

The following Signtool actions are available:

- Signtool Sign Files
- Signtool Timestamp Files
- Signtool Verify Digital Signatures

Signtool Options Page

Before using any of the Signtool actions you must first specify the path to the Signtool executable via the options page.

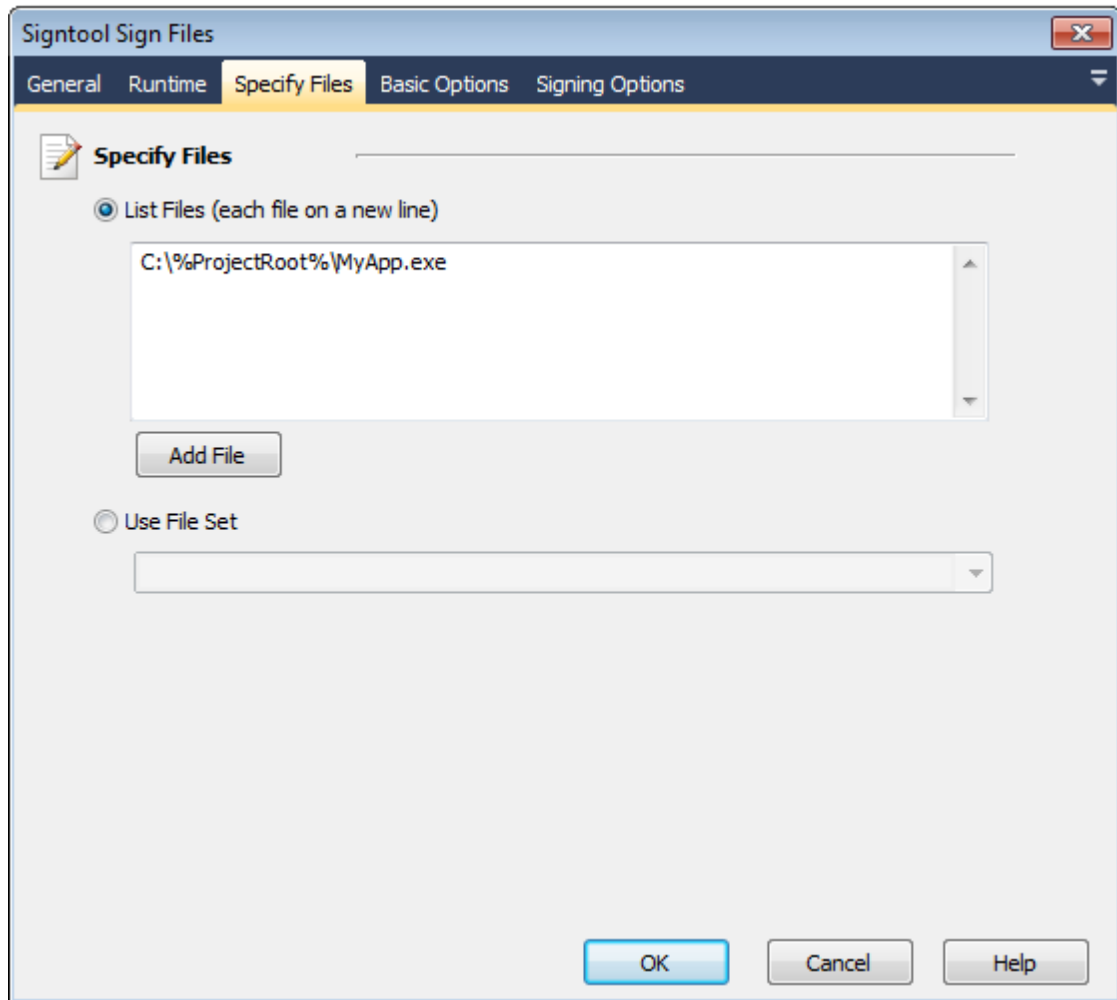


For more information on Signtool see MSDN.

6.12.24.1 Signtool Sign Files

The Signtool Sign Files action allows you to digitally sign files.

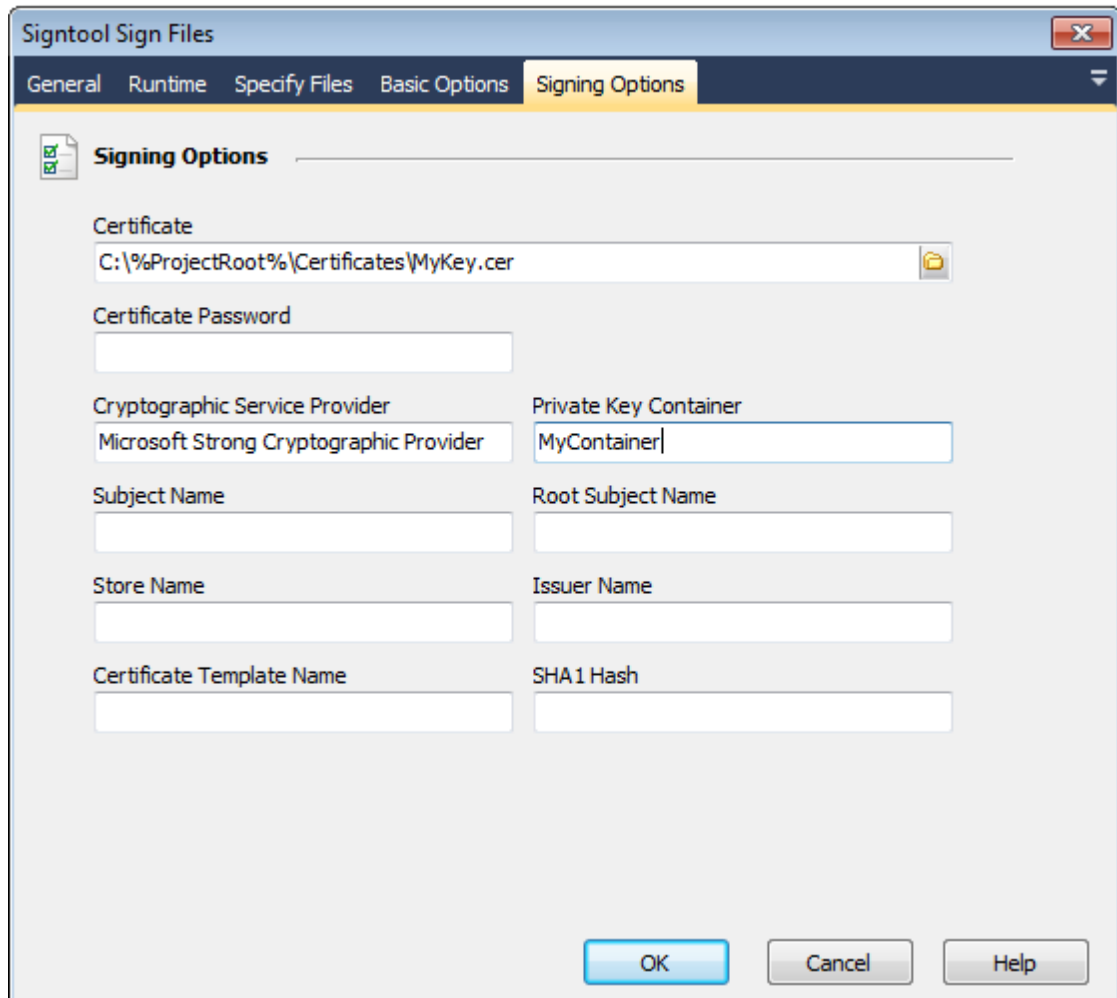
From the *Specify Files* tab specify the files that you wish to sign (each entry on a new line).



From the options section select any of the following options that you would like executed:

- **Verbose Output** - Self Explanatory.
- **Automatically select the best signing certificate** - Self Explanatory.
- **Use Computer store rather than User store** - Self Explanatory.
- **Signed Content Description** - A text description of the signed content.
- **Signed Content Description (URL)** - A URL to the full description of the signed content.
- **Use Timestamp Service URL** - The URL of the timestamp server.
- **Enhanced Key Usage** - Specifies the Enhanced Key Usage that must be present in the signing certificate.

The *Signing Options* tab allows you to specify extra details about the signature:



- **Certificate** - Specify the file to use as the signing certificate. When using a Personal Information Exchange (.pfx) which is protected by a password, use the *Certificate Password* option to specify the password. If the file does not contain private keys use the *Cryptographic Service Provider* and *Private Key Container* options to specify the private keys to be used.
- **Certificate Password** - The password to use when opening a password protected Personal Information Exchange (.pfx) file.
- **Cryptographic Service Provider** - The Cryptographic Service Provider that contains the private key container.
- **Private Key Container** - The private key container name.
- **Subject Name** - The subject name of the signing certificate.
- **Root Subject Name** - Specify the subject name of the root certificate that the signing certificate must link to.
- **Store Name** - Specify the name of the store to open when searching for the certificate.

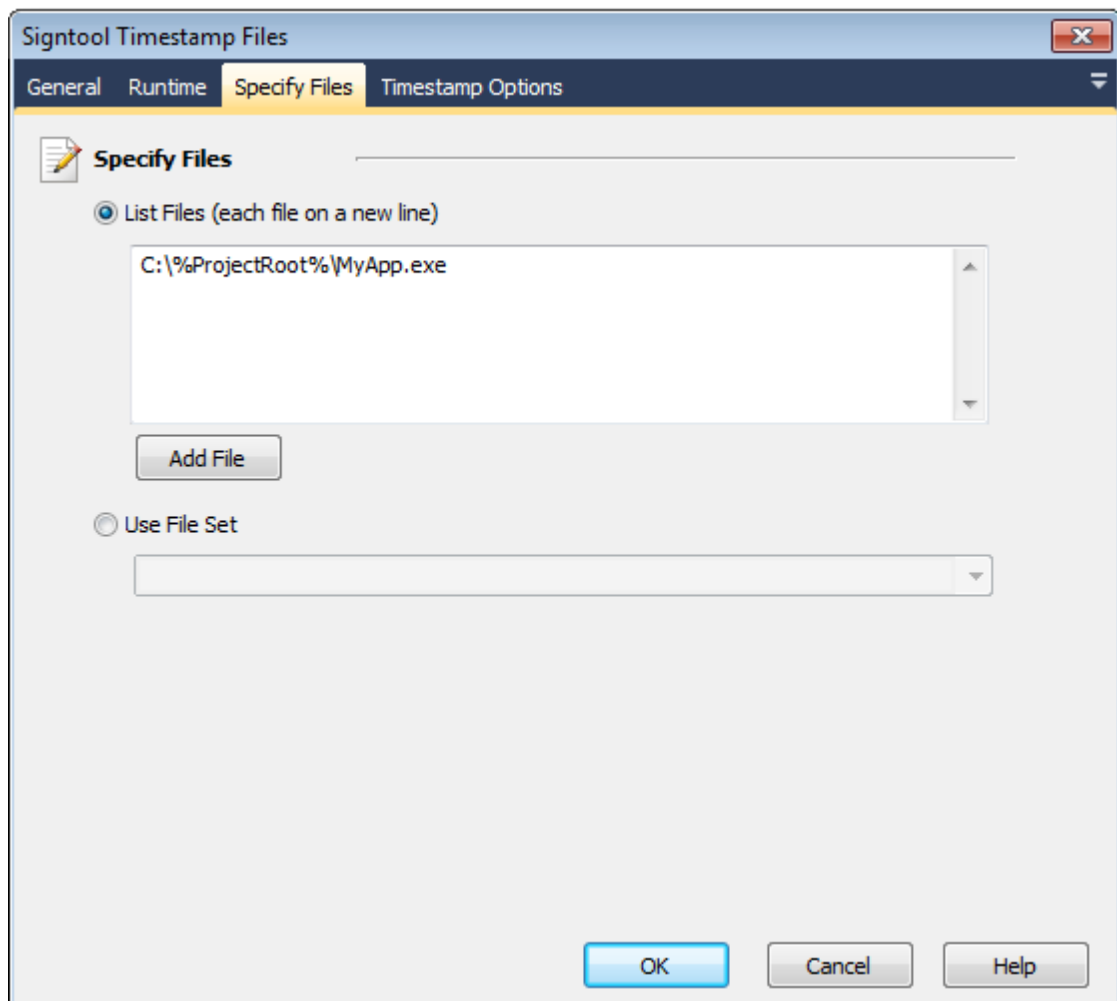
- **Issuer Name** - Specify the issuer name of the signing certificate.
- **Certificate Template Name** - The certificate template name for the signing certificate.
- **SHA1 Hash** - Specify the SHA1 hash of the signing certificate.

For more information on Signtool see MSDN.

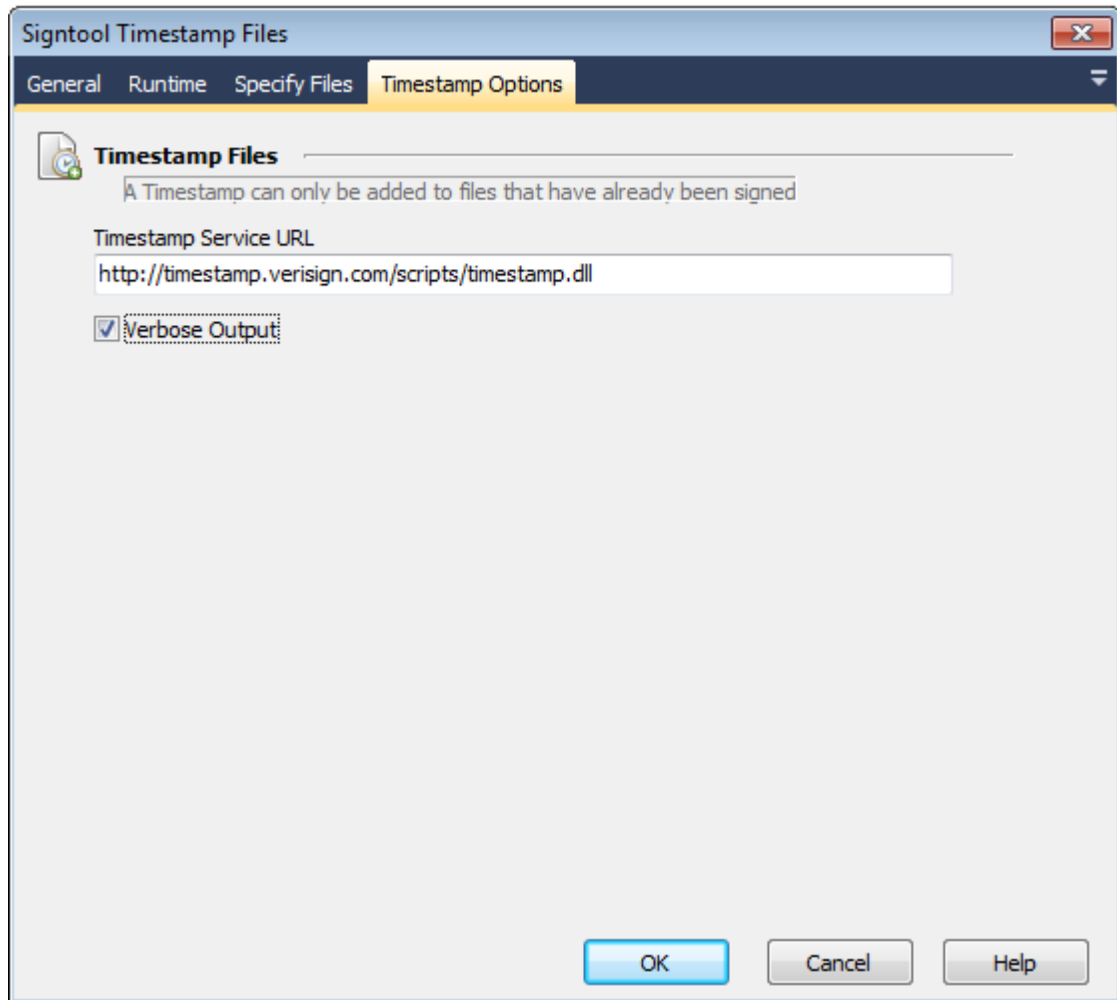
6.12.24.2 Signtool Timestamp Files

The Signtool Timestamp Files action allows you to apply a timestamp to files that have already been signed.

Specify the files that you want to apply the timestamp to (each entry on a new line). The files specified must have already been digitally signed otherwise the action will fail.



Specify the *Timestamp Service URL* that will be used obtain the timestamp.



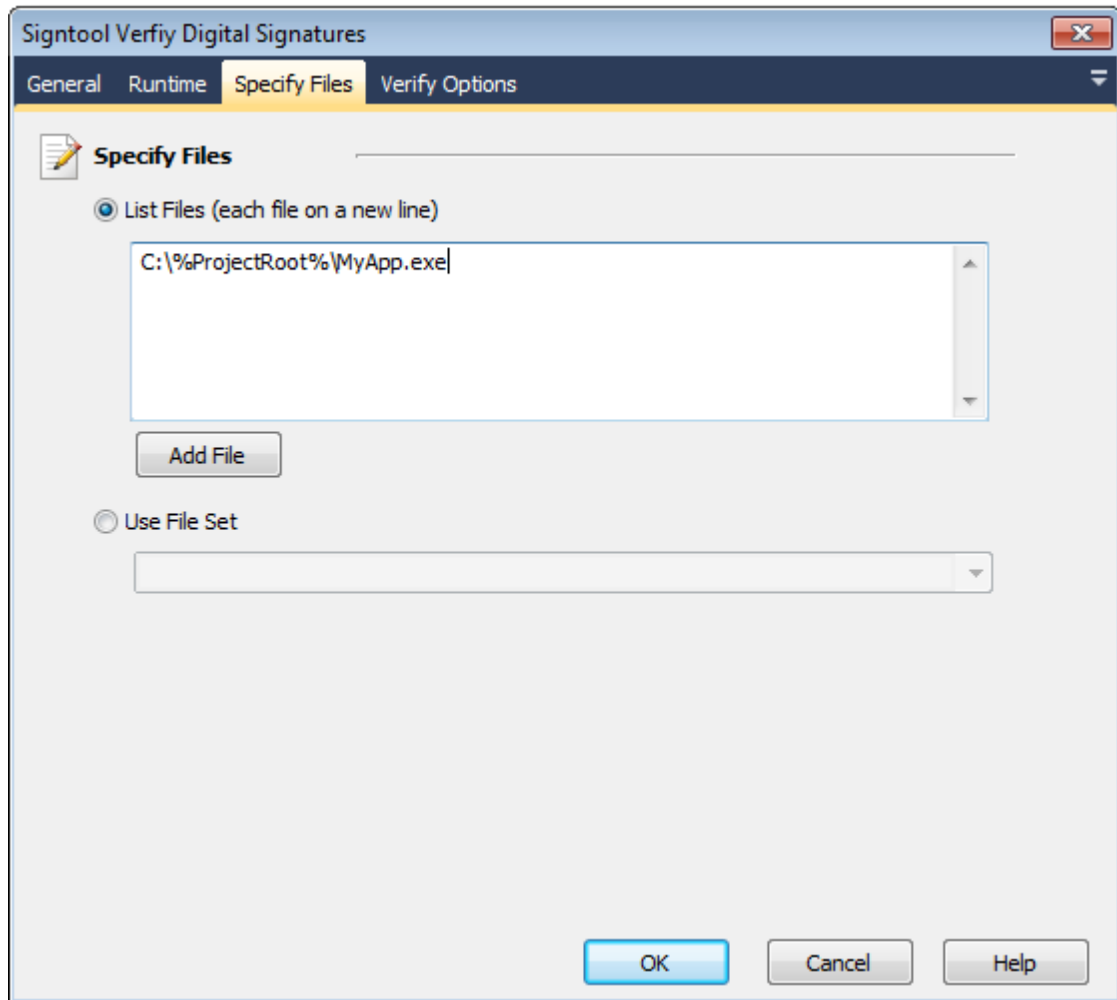
You can also use the **Verbose Output** option with this action to obtain extended output from the Signtool executable.

For more information on Signtool see MSDN.

6.12.24.3 Signtool Verify Digital Signatures

The Verify Digital Signatures action allows you to verify digital signatures that have been applied to files.

Specify the files which you want to verify (each entry on a new line).



Set any of the following options that you want to execute:

- **All Methods** - Use all verification methods to verify the files.
- **Default Catalog Database** - Find catalog from default catalog database.
- **System Component Catalog Database** - Finds the catalog from the System Component (Driver) catalog database.
- **Catalog Database GUID** - Provide a GUID to identify the catalog in the catalog database.
- **Catalog File** - Specify the catalog file.
- **OS Version** - Use the operating system version to verify the file. The format for this value needs to be passed in the following way: *PlatformId:MajorVersion:MinorVersion.BuildNumber* (build number is optional).
Note: This option can only be used when one of the options above is used.
- **Verbose Output** - Self Explanatory.

- **Default Authentication Verification Policy** - Self Explanatory.

Note: Cannot be used with the Verification Policy GUID option.

- **Warning If Signature Not Time Stamped** - Self Explanatory.
- **Verification Policy GUID** - Specify the GUID that matches the Action Id of the verification policy to be used.

Note: Cannot be used with the Default Authentication Verification Policy option.

- **Subject Name of Root Certificate** - Specify the subject name of the root certificate that the signing certificate must link to.

For more information on Signtool see MSDN.

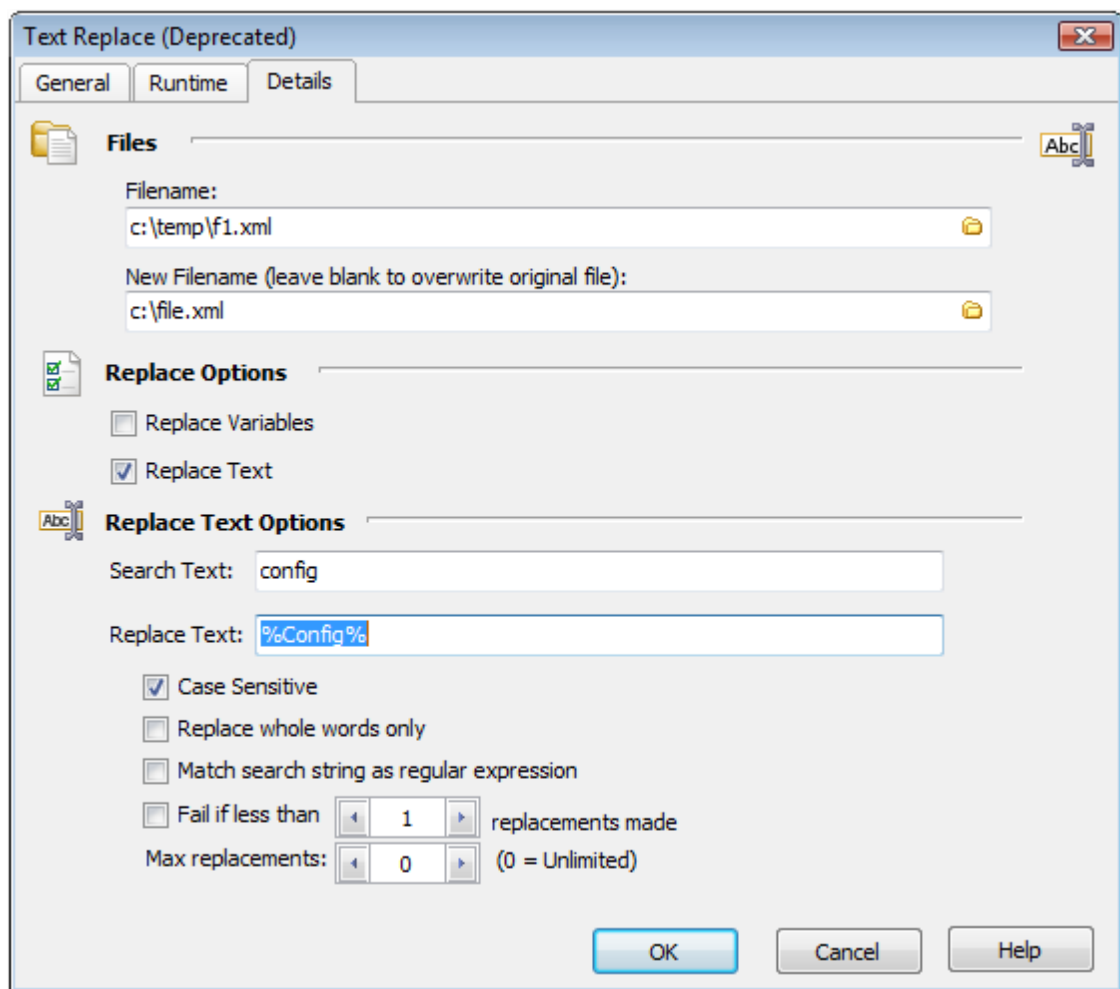
6.12.25 Text Replace Action

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 FinalBuilder 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.



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 FinalBuilder Variables

If this box is checked, FinalBuilder 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

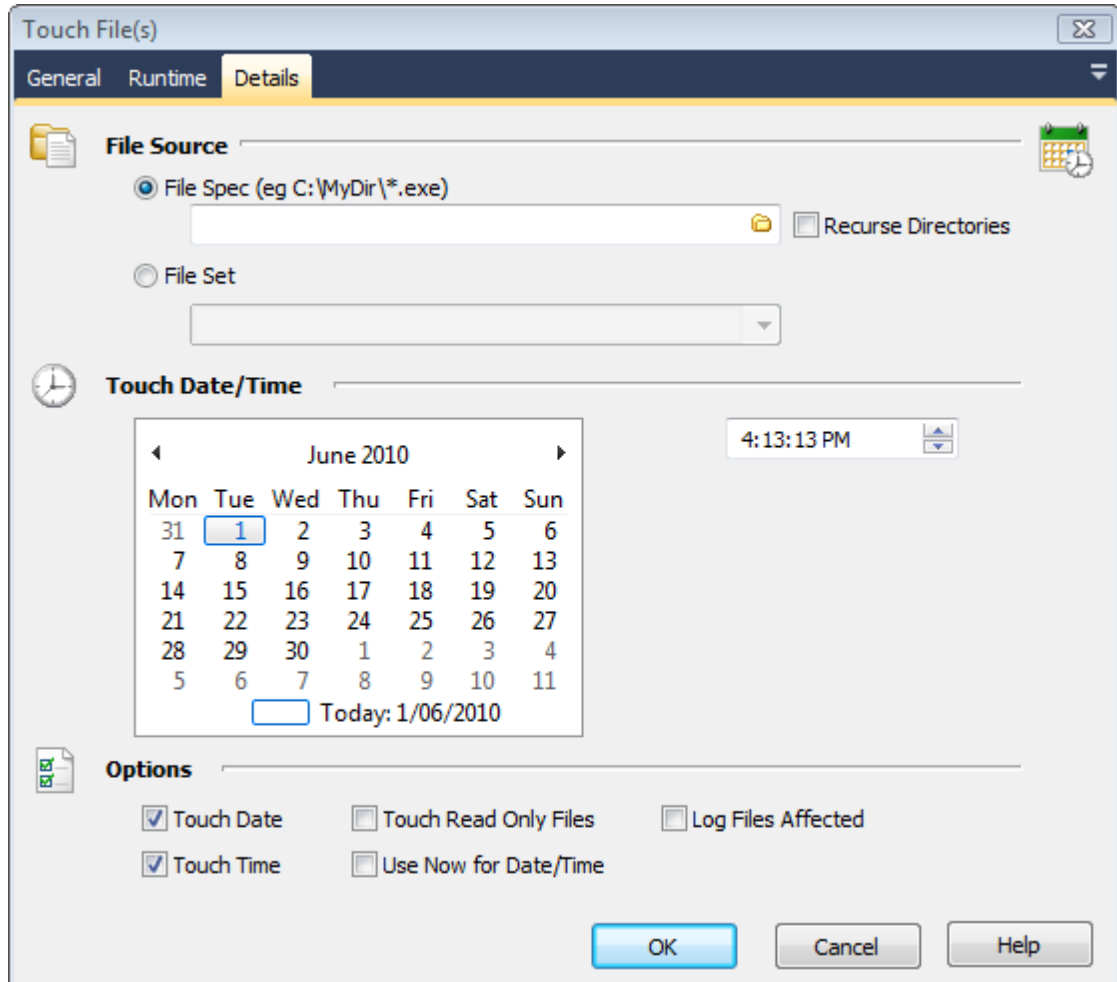
The action can be set to fail if an insufficient amount of replacements are made. This is useful to check that there isn't a syntax error or other mistake in the text file.

Max Replacements: N

The action will replace only the first N matches it finds. Set to zero to replace all matches.

6.12.26 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.



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

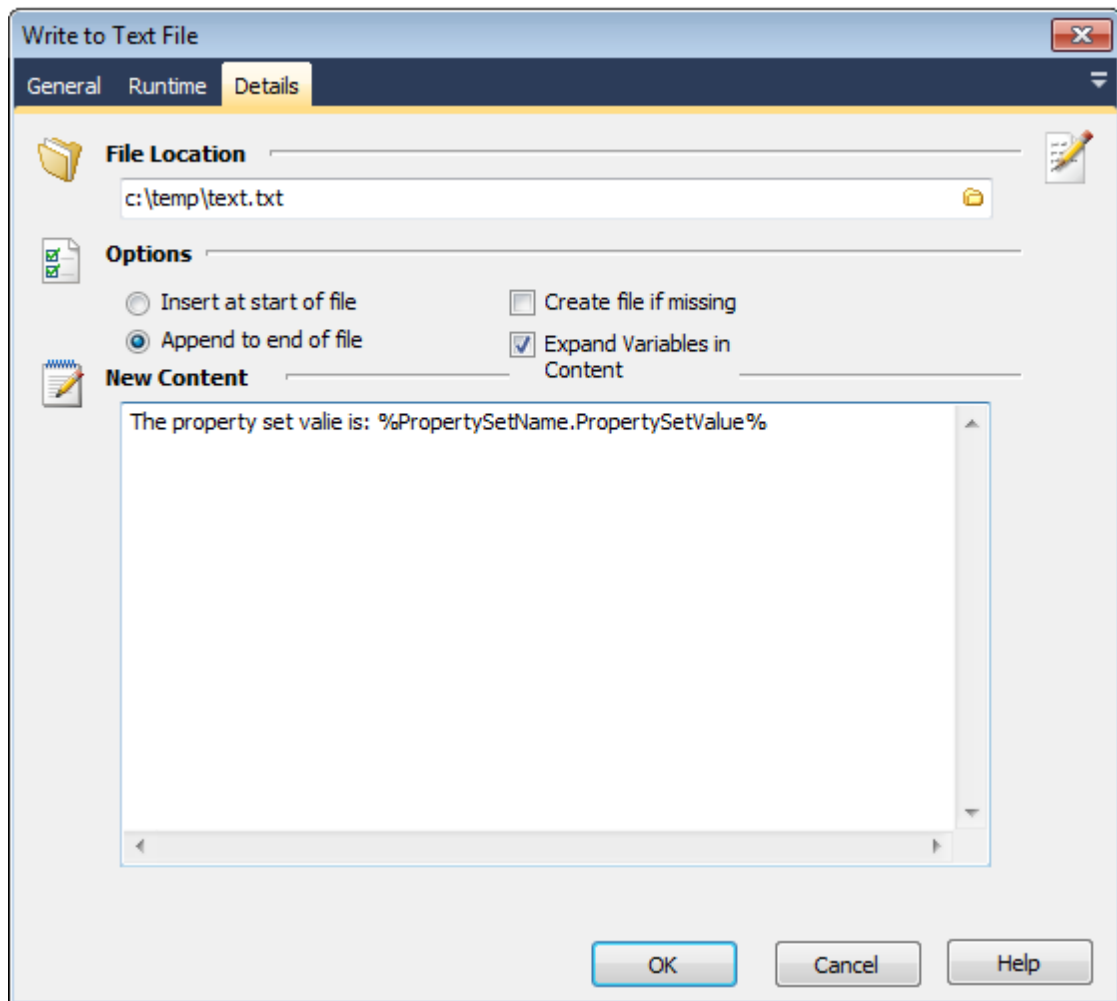
The Action properties available are :

- 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

6.12.27 Write Text File

The Write to Text File action enables you to automate inserting and appending text to an existing file.

You can use FinalBuilder 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 - 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

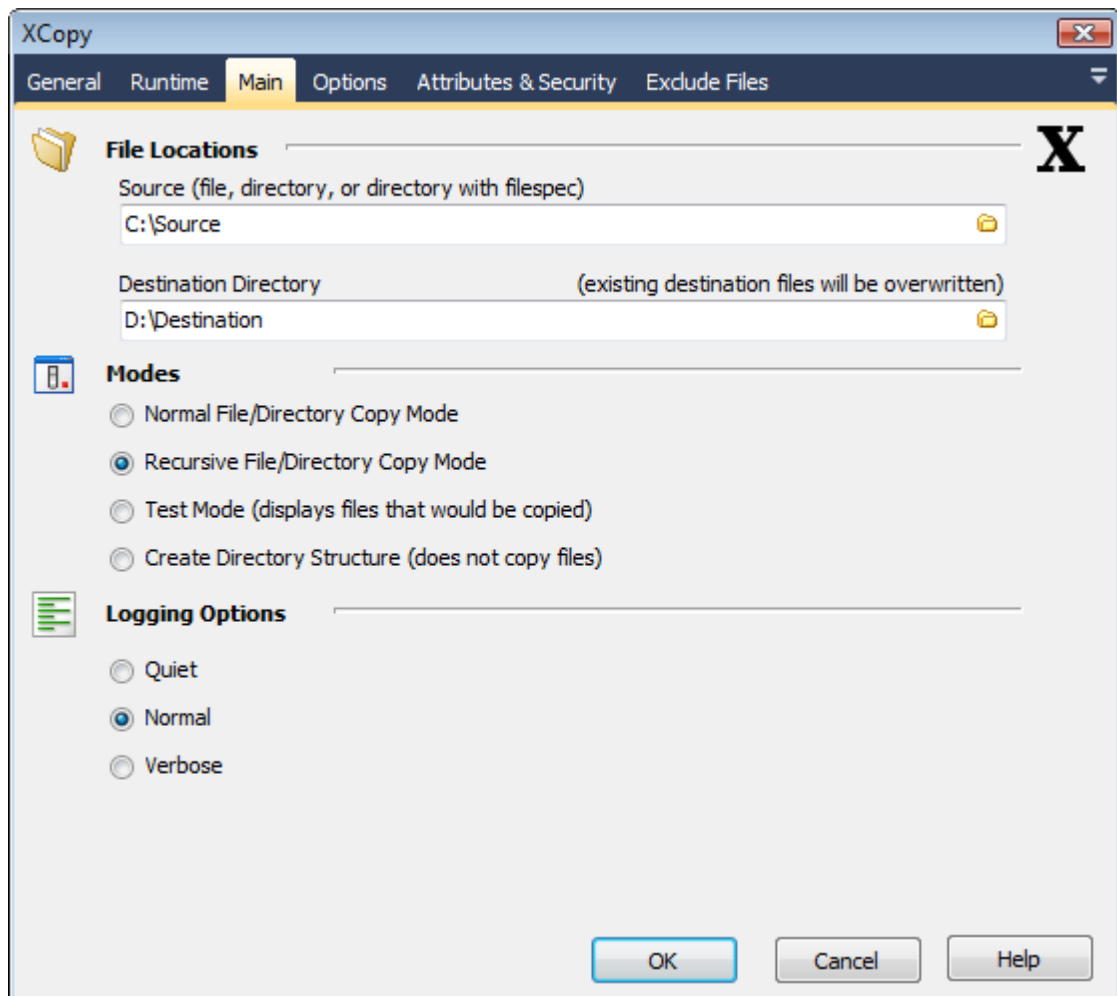
Create file if missing - the action will normally fail if the file doesn't already exist.

New Content - specify the content to write to the text file

6.12.28 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



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.

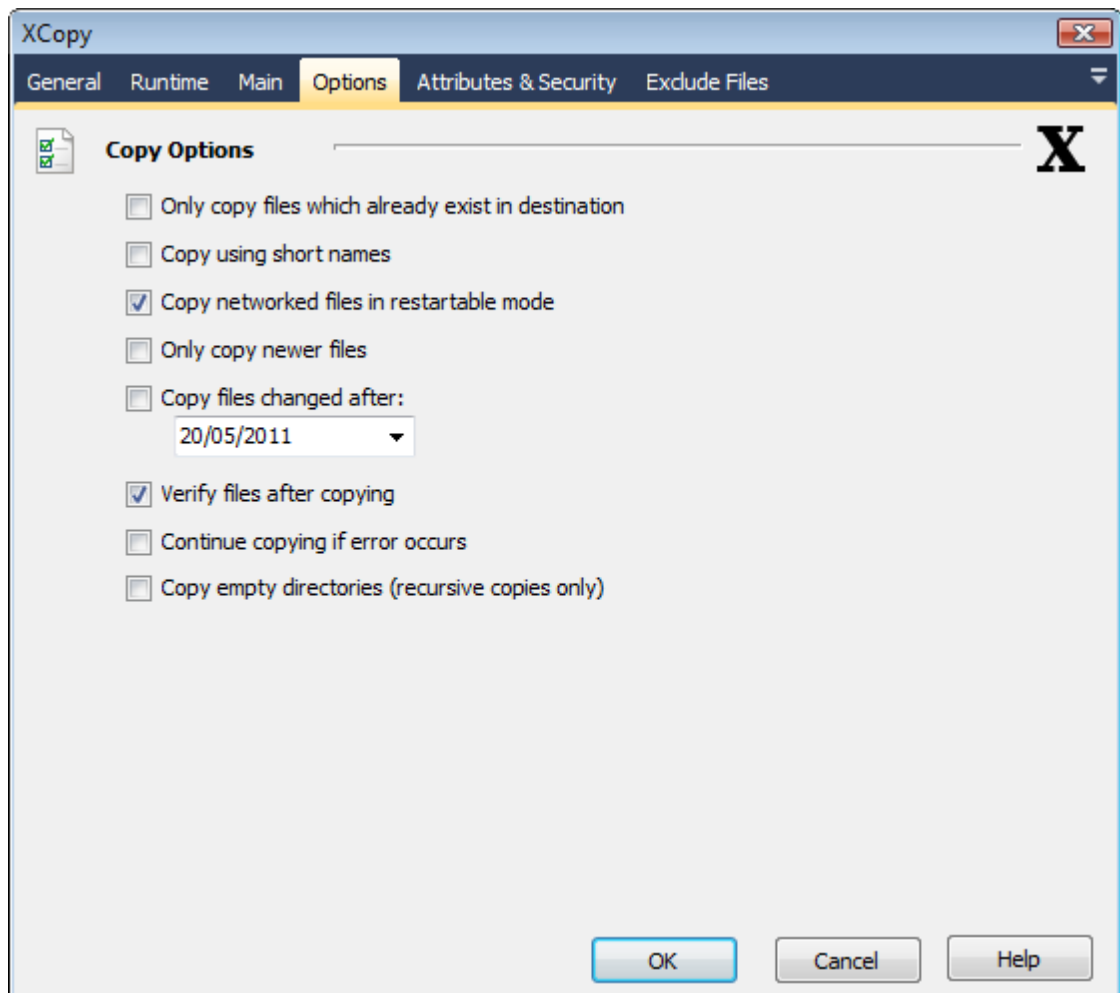
Logging Options

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



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.

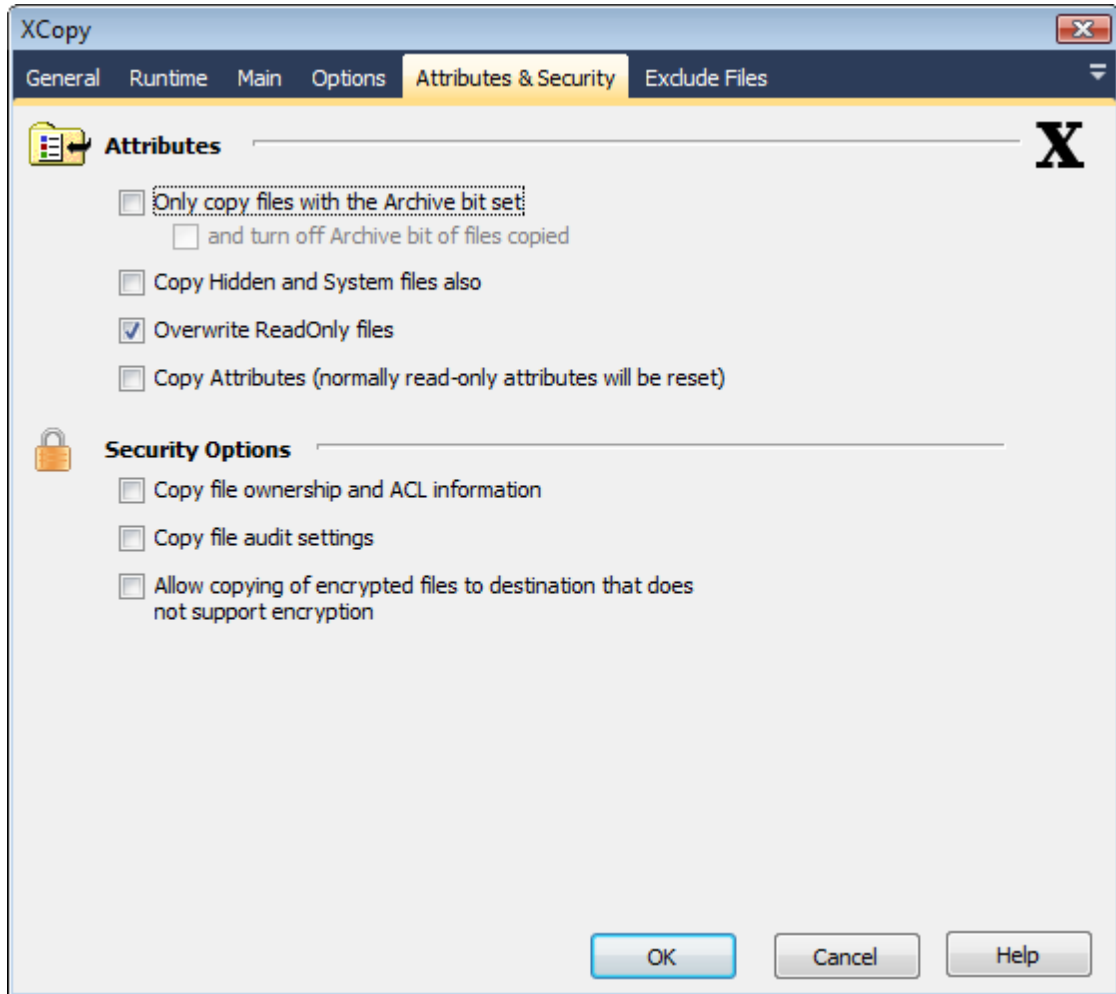
Copy files that are changed after - Enable this option to only copy files that exist in both the source and the destination where the source files have changed after the date specified.

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



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.

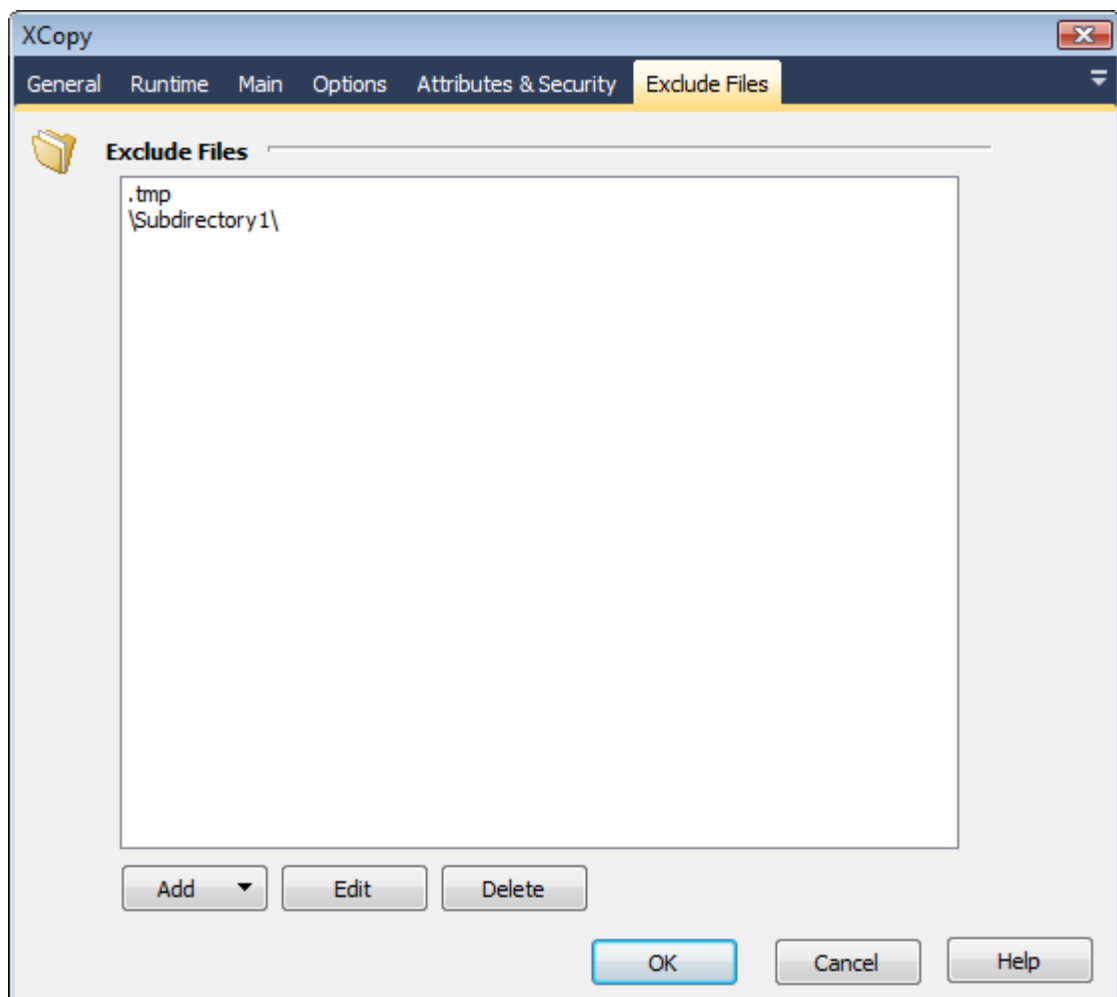
Copy file audit settings - Copies the file audit settings as well as the system

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 any text files).



For more information on XCopy see: <http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/xcopy.msp>

6.13 Flow Control

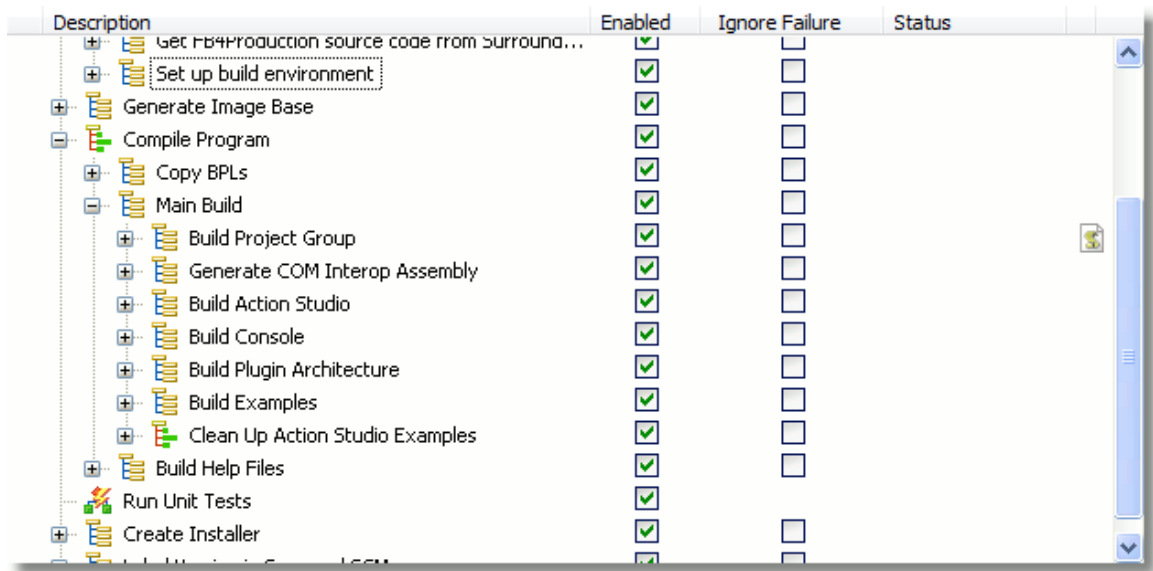
FinalBuilder includes many actions which let you control which actions are executed in your project. The Flow Control category in the Actions Pane includes the following flow control actions:

- 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 FinalBuilder 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.

6.13.1 Action Group

The Action Group action type has no specific functionality, but is extremely useful for structuring your project.



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:

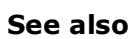
The Action Group's BeforeAction, AfterAction, and OnStatusMessage script events execute the same as for any other action type.

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.



© 2001-2010 VSoft Technologies Pty Ltd

6.13.2 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.

See Also

Switch Action

6.13.3 Delay Action

This Action will pause for the time specified in the Delay property (in milliseconds).

6.13.4 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	<input checked="" type="checkbox"/>	Skipped
(This will never run)	Action Group	<input checked="" type="checkbox"/>	
Else	Else	<input checked="" type="checkbox"/>	Completed
(This will run)	Action Group	<input checked="" type="checkbox"/>	Completed

For switch actions:

Description	Action	Enabled	Status
Enhanced Prompt for Variables [- SwitchVar]	Enhanced Prompt f...	<input checked="" type="checkbox"/>	
Switch [%SwitchVar%]	Switch	<input checked="" type="checkbox"/>	
Case [Case1]	Case	<input checked="" type="checkbox"/>	
Ask Question [Do you want to fail the build?]	Ask Question	<input checked="" type="checkbox"/>	
Case [Case2]	Case	<input checked="" type="checkbox"/>	
Beep [Exclamation]	Beep	<input checked="" type="checkbox"/>	
Else (Double-click to see comment)	Else	<input checked="" type="checkbox"/>	
Put current date/time into variable TheDate, format: ddmmyyyy	Get DateTime	<input checked="" type="checkbox"/>	
Done!	Action Group	<input checked="" type="checkbox"/>	

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

6.13.5 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

6.13.6 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)

The screenshot shows the 'For Loop' dialog box with the following configuration:

- Loop From:** Constant value: 0
- Loop To:** Variable value: ProjectCount, Subtract 1 checkbox is unchecked
- Step:** Constant value: 1
- Set Loop Value to Variable:** varToSet, Show Loop Value in Action Log Title checkbox is checked

Loop From

The initial value for the loop value. The value can be either a constant or a FinalBuilder 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 FinalBuilder 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 FinalBuilder variable.

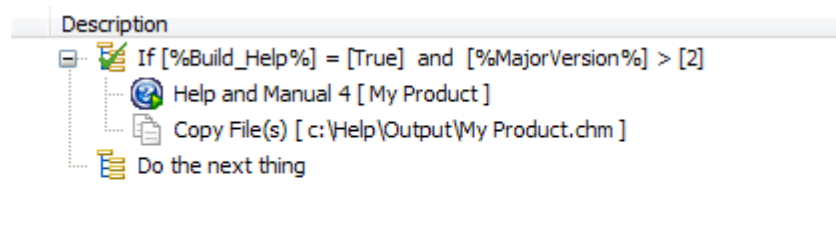
Set Loop Value to Variable

Specify a FinalBuilder variable to use for the loop value. If no variable is specified, the loop value will be stored internally.

TIP: If you use FinalBuilder 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.

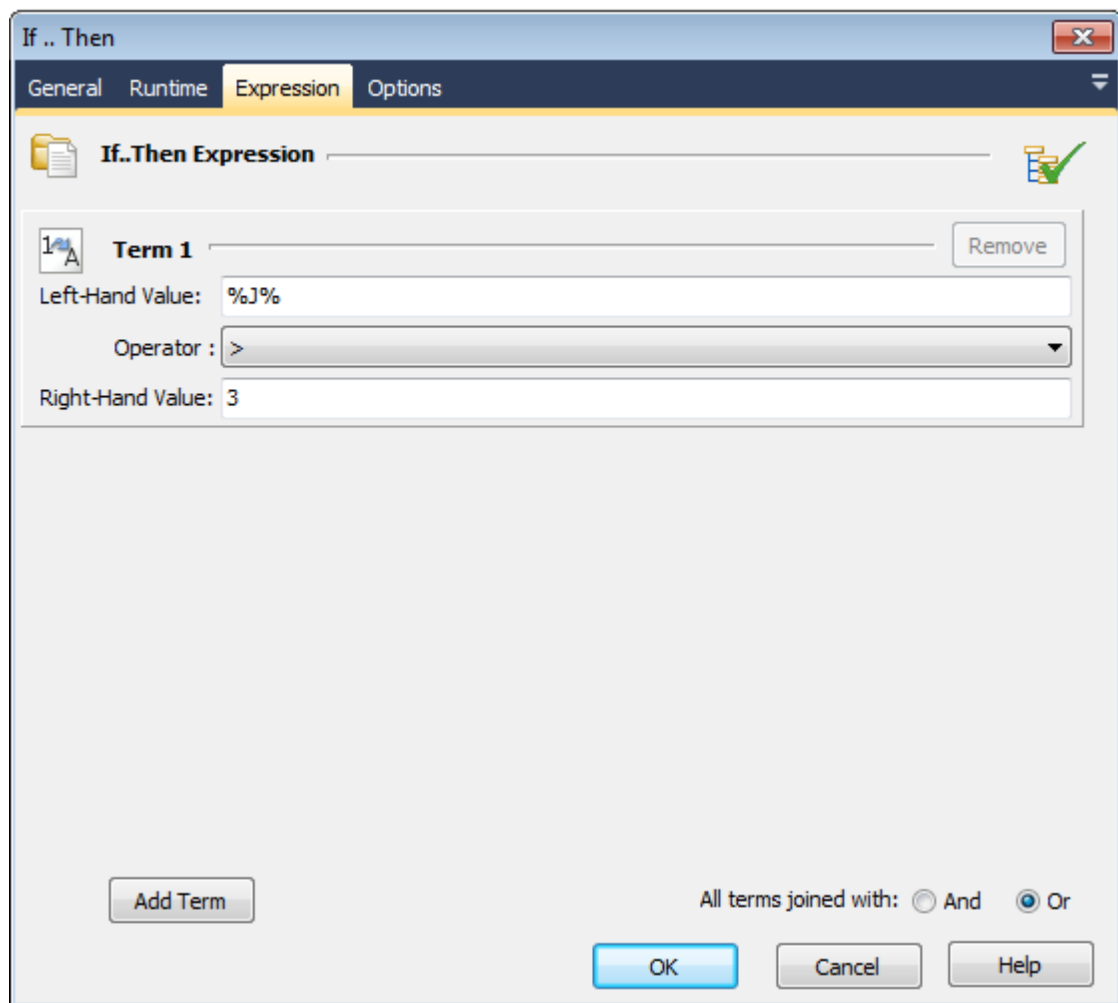
6.13.7 If .. Then Action

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 its child actions or not.

If..Then Properties Dialog



Each term is comprised of a left hand side, a right hand side, 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

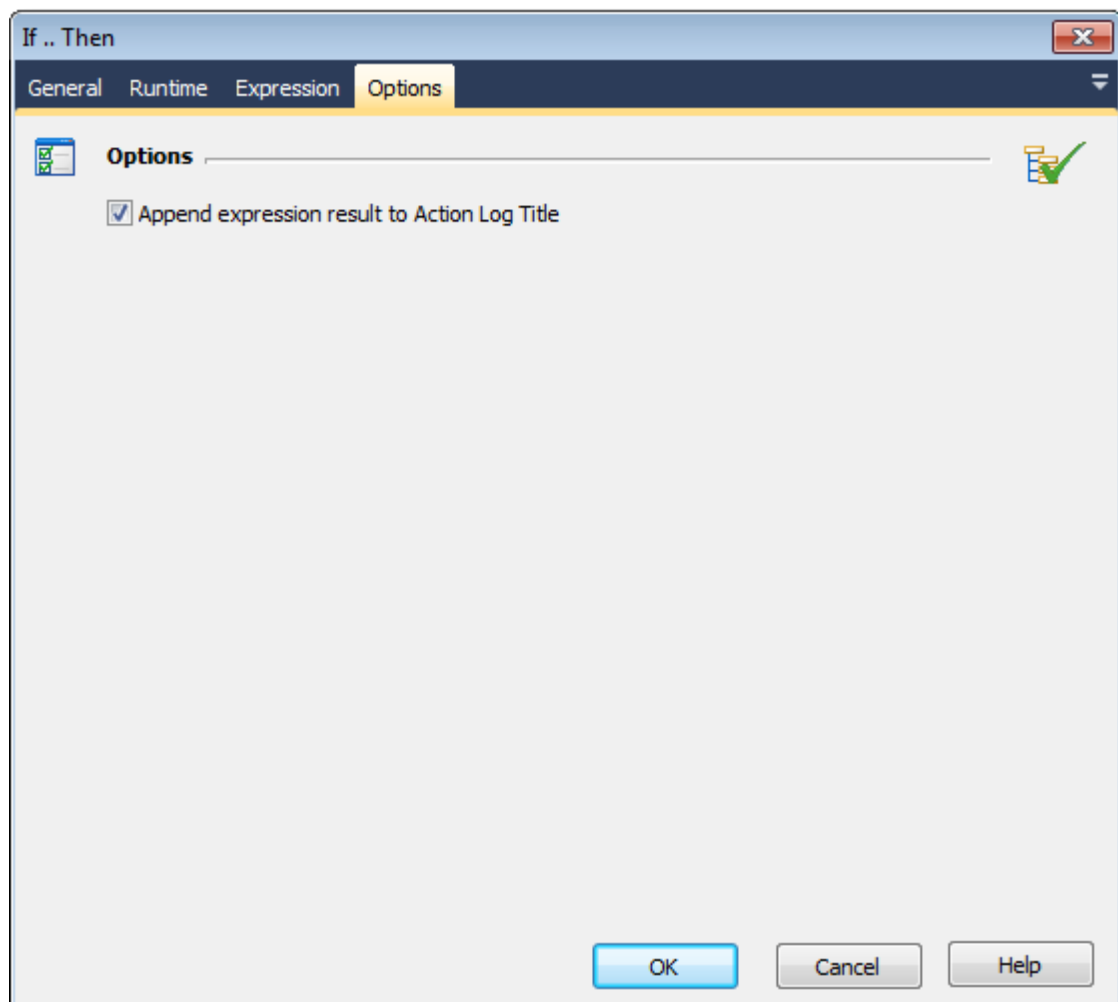
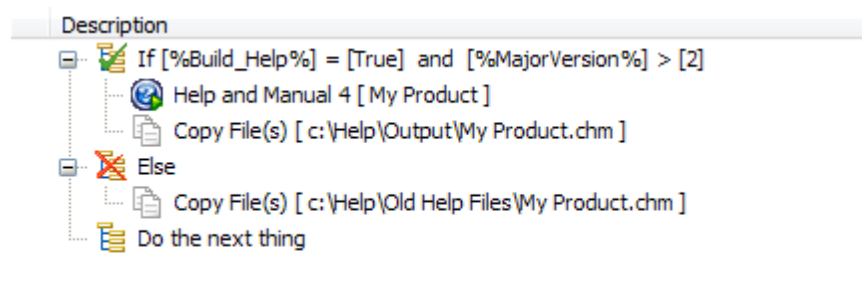
The following If... Then action will execute its children when variable J is equal to 4, 5, 6, 7 or 9:

The screenshot shows the 'If..Then' dialog box with the 'Expression' tab selected. The dialog contains three terms for the variable '%J%':

- 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.

At the bottom, the 'All terms joined with:' section has the 'Or' radio button selected. The 'Add Term' button is visible on the left, and 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

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.






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

6.13.8 If Prev Action Failed Action

This action will execute its child actions only if the previous sibling action failed. For this to happen the previous sibling action needs Ignore Failure enabled:

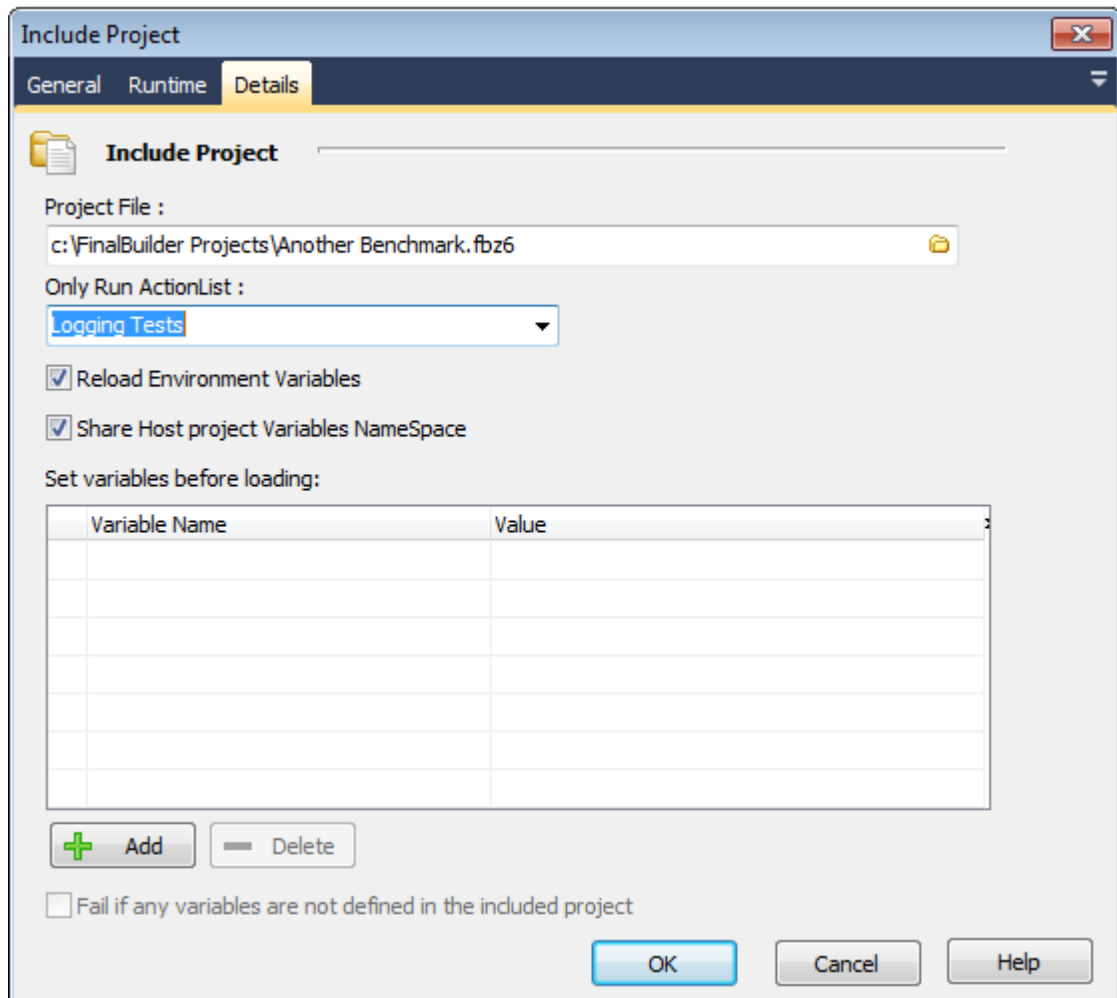
Description	Enabled	Ignore Failure	Status
 An action which might fail	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Error Ignored
 If Prev Action Failed	<input checked="" type="checkbox"/>		Completed
 The action failed, so do some other stuff	<input checked="" type="checkbox"/>	<input type="checkbox"/>	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.

6.13.9 Include FinalBuilder Project

This action type allows you to include other FinalBuilder projects in your project. This allows you to modularise your project by including subprojects, as an alternative to using many Action Lists.



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 FinalBuilder 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.

6.13.10 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
 Try	<input checked="" type="checkbox"/>		Completed
 Raise Exception	<input checked="" type="checkbox"/>		Error
 Catch	<input checked="" type="checkbox"/>		Completed
 The exception above will be handled here	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
 Finally	<input checked="" type="checkbox"/>		Completed
 This will be executed before the action list finishes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
 End	<input checked="" type="checkbox"/>		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.

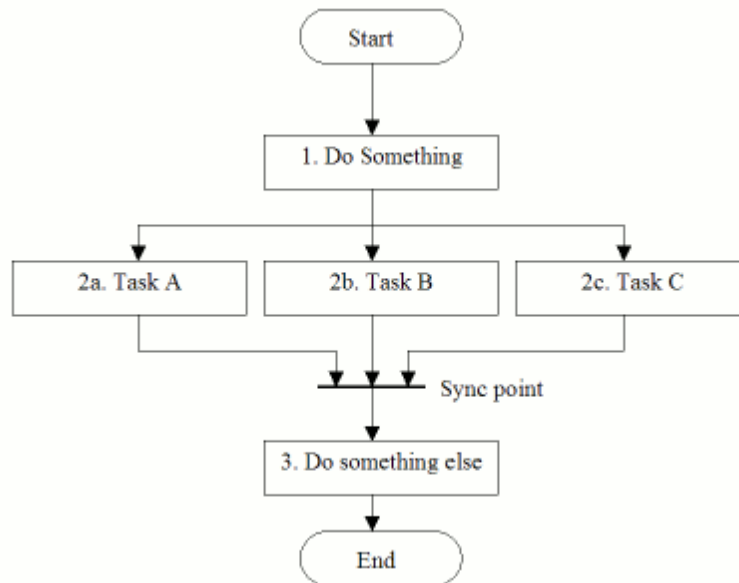
In the example above, the project stopped with status "Succeeded" because the exception was handled in the Try/Catch block.

6.13.12 Running actions in Parallel

[FinalBuilder Professional Edition]

Motivation

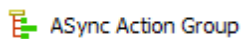
FinalBuilder allows you to structure your project so that multiple actions can be run simultaneously. A graphical representation of multiple actions running in parallel is represented by this flow diagram:



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).



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 FinalBuilder as follows:

Description	Enabled	Ignore Failure	Status
1. Do Something	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
ASync Action Group	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
2a. Task A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
2b. Task B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
2c. Task C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
3. Do something else	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

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.

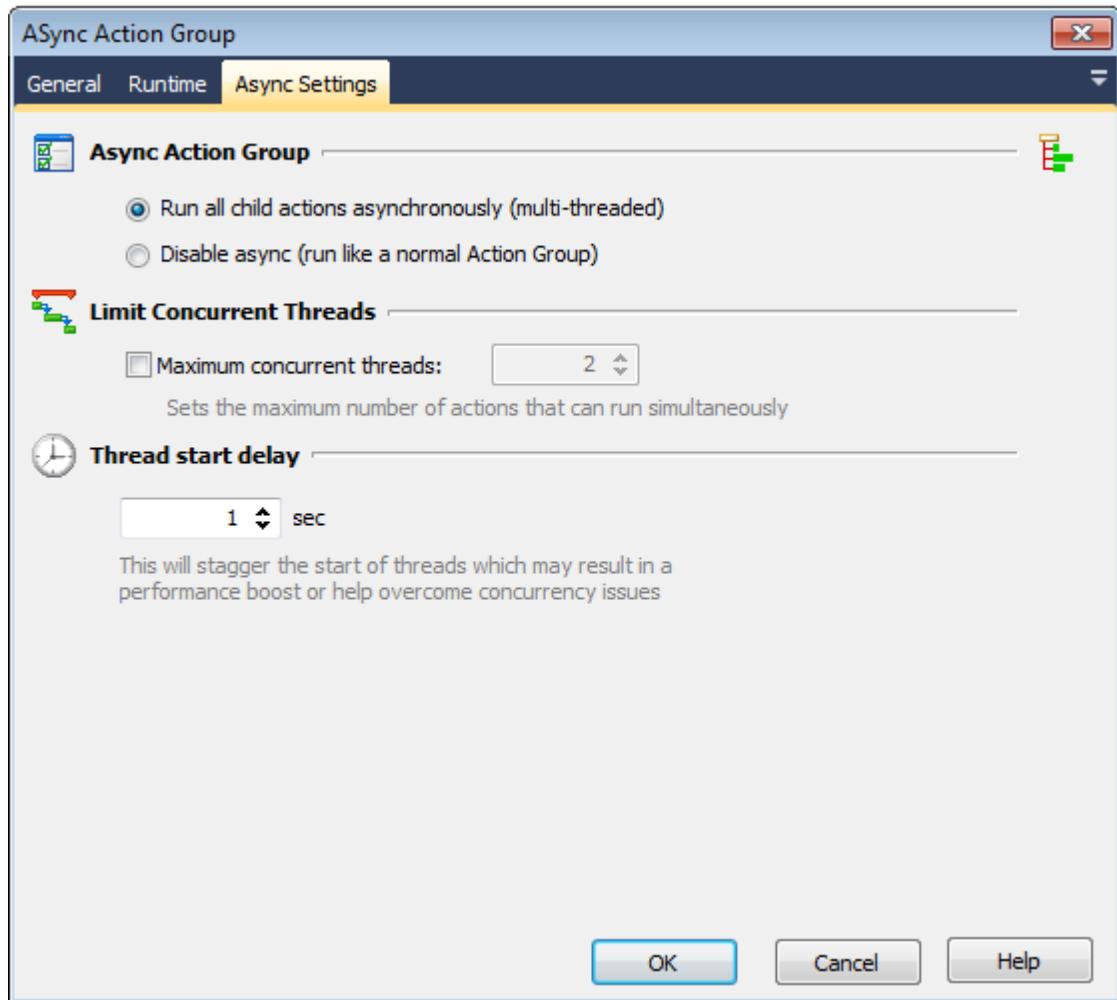
Grouping actions under ASync

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	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
ASync Action Group	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
2a. Task A	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
2b. Task B	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
Action 2ba	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
Action 2bb	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
Action 2bc	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
Action 2bd	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2c. Task C	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Running
3. Do something else	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

(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



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 FinalBuilder 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.
- The Visual SourceSafe actions cannot be used under Async group as the actions temporarily modify the SourceSafe INI files.

6.13.13 Stop Build Action

The "Stop Build" action will cause the project to stop running.

Description	Action	Enabled	Status
Try	Try	<input checked="" type="checkbox"/>	
Stop Run [Success]	Stop Run	<input checked="" type="checkbox"/>	
Catch	Catch	<input checked="" type="checkbox"/>	
This won't be executed before the project stops.	Action Group	<input checked="" type="checkbox"/>	
Finally	Finally	<input checked="" type="checkbox"/>	
This will be executed before the project stops.	Action Group	<input checked="" type="checkbox"/>	
End	End	<input checked="" type="checkbox"/>	

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

6.13.14 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...	<input checked="" type="checkbox"/>	
Switch [%SwitchVar%]	Switch	<input checked="" type="checkbox"/>	
Case [Case1]	Case	<input checked="" type="checkbox"/>	
Ask Question [Do you want to fail the build?]	Ask Question	<input checked="" type="checkbox"/>	
Case [Case2]	Case	<input checked="" type="checkbox"/>	
Beep [Exclamation]	Beep	<input checked="" type="checkbox"/>	
Else (Double-click to see comment)	Else	<input checked="" type="checkbox"/>	
Put current date/time into variable TheDate, format: ddmmyyyy	Get DateTime	<input checked="" type="checkbox"/>	
Done!	Action Group	<input checked="" type="checkbox"/>	

See Also






Else Action

6.13.15 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	<input checked="" type="checkbox"/>		Completed
 Raise Exception	<input checked="" type="checkbox"/>		Error
 Catch	<input checked="" type="checkbox"/>		Completed
 The error created above will be handled here	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
 End	<input checked="" type="checkbox"/>		Completed

2. Try ... Finally block

Description	Enabled	Ignore Failure	Status
 Try	<input checked="" type="checkbox"/>		Completed
 Raise Exception	<input checked="" type="checkbox"/>		Error
 Finally	<input checked="" type="checkbox"/>		Completed
 This action will run regardless of whether the try fails or succeeds.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
 End	<input checked="" type="checkbox"/>		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

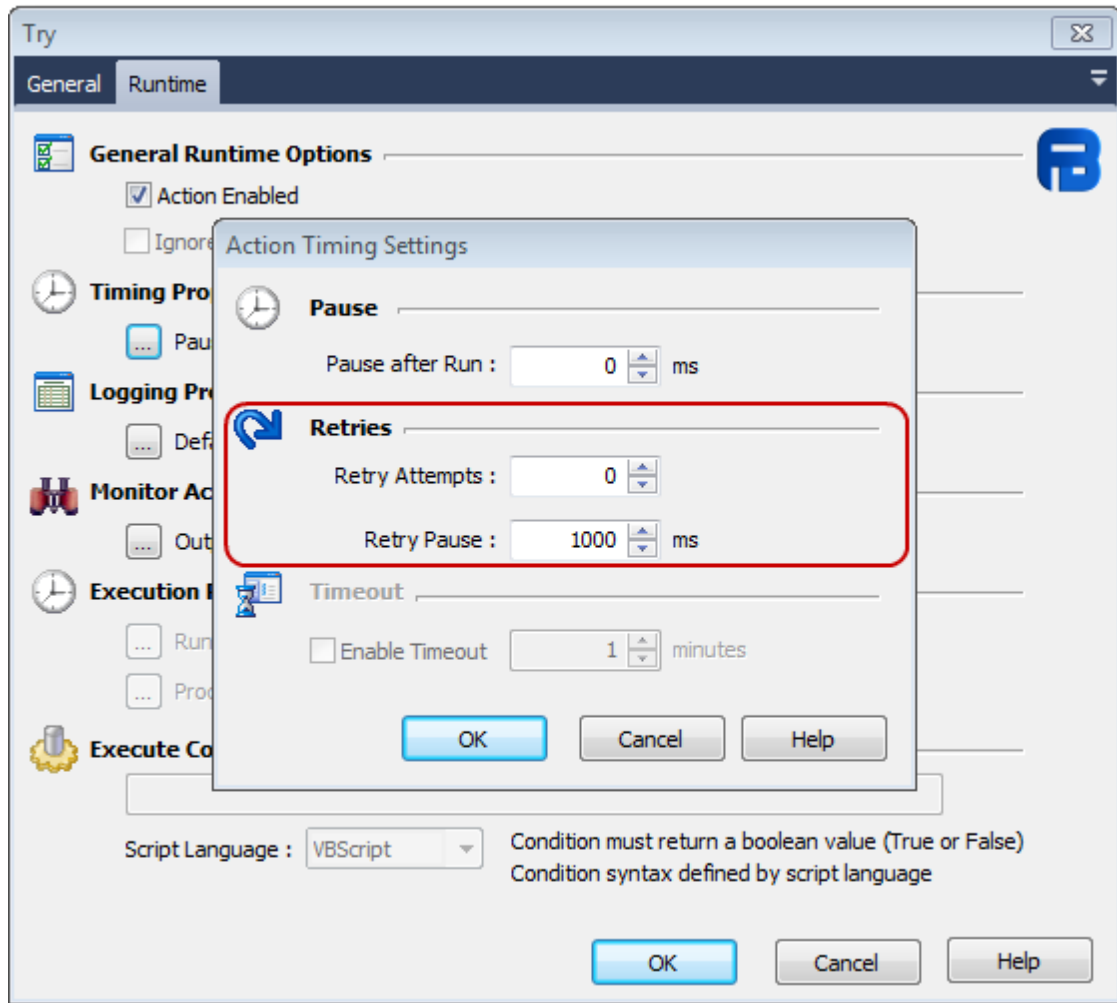
Description	Enabled	Ignore Failure	Status
 Try	<input checked="" type="checkbox"/>		Completed
 Raise Exception	<input checked="" type="checkbox"/>		Error
 Catch	<input checked="" type="checkbox"/>		Completed
 The exception above will be handled here	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
 Finally	<input checked="" type="checkbox"/>		Completed
 This will be executed before the action list finishes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed
 End	<input checked="" type="checkbox"/>		Completed

A more complete example:

Description	Enabled	Ignore Failure	Status
Read my comment	<input checked="" type="checkbox"/>		
T Try	<input checked="" type="checkbox"/>		
Create Directory [c:\TempFinalBuilderDirectory]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Write Ini File	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Read Ini File	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
An action group which contains an error [Initially Disabl...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Read my comment	<input checked="" type="checkbox"/>		
Check if File Exists [c:\TempFinalBuilderDirectory\...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Beep	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
C Catch	<input checked="" type="checkbox"/>		
Handle the error here.	<input checked="" type="checkbox"/>		
Set Variable ERRORLOCATION to [CreateDirectory]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
F Finally	<input checked="" type="checkbox"/>		
Read me too	<input checked="" type="checkbox"/>		
Delete Directory [c:\TempFinalBuilderDirectory]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Make sure temporary directory has been deleted	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
E End	<input checked="" type="checkbox"/>		
Read me	<input checked="" type="checkbox"/>		

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.



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.

6.13.16 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 FinalBuilder log when something actually happens.

Polling Type

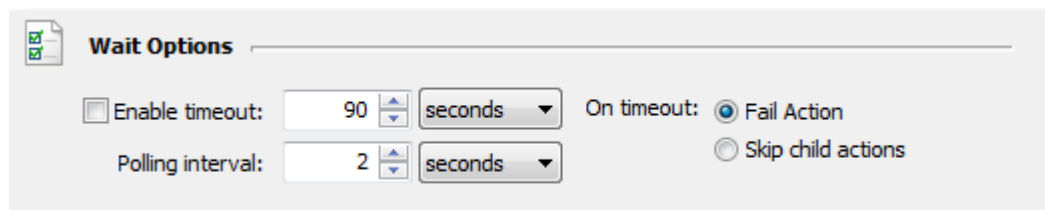
Note that the "Wait For..." action is a "busy-waiting" or "polling" wait - a particular wait condition is checked every so many seconds to see if it has changed.

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 On timeout: ☒ Fail Action

☐ Polling interval: 2 seconds ☐ 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.)

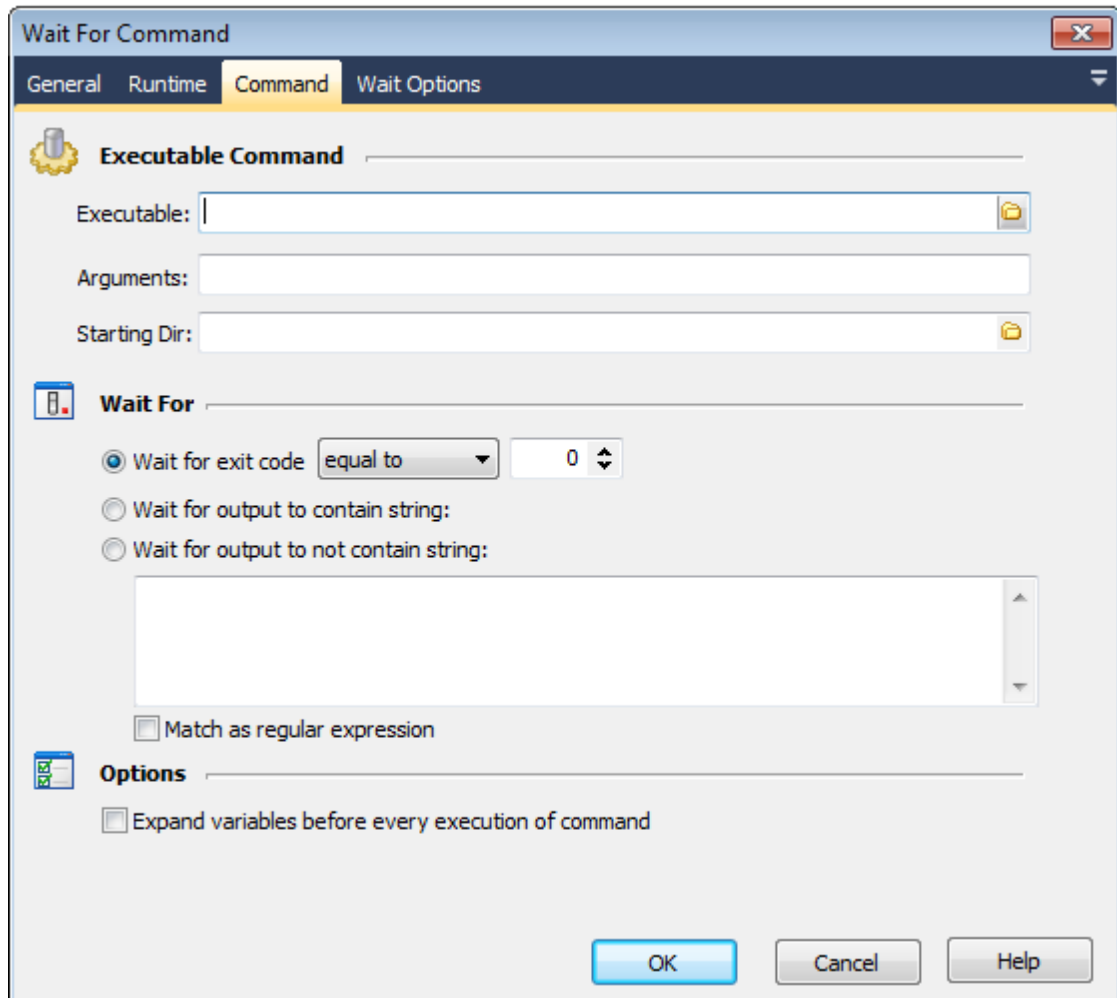
Timing Limitations

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 $X + (\text{Length of last polling event})$ seconds, and in some actions (ie Wait For Remote Computer) the polling event itself may take several seconds.

6.13.16.1 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 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

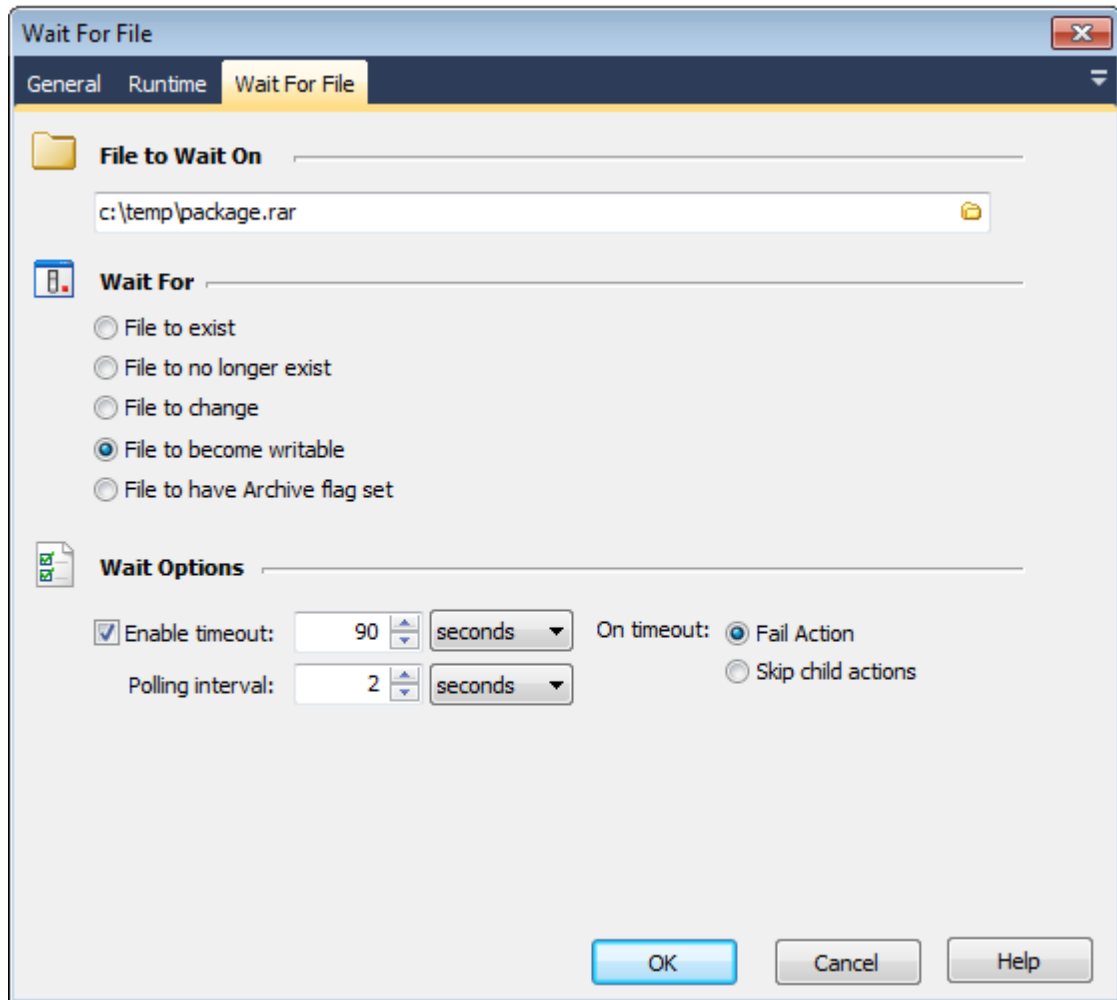
In addition to the normal Wait script events, Wait For Command provides an OnCommandOutput event. This script event allows you to do more advanced parsing of the command output (available in the Output parameter) or the exit code (available in the Exit Code parameter.) Set the Continue parameter to True 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.)

6.13.16.2 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.



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

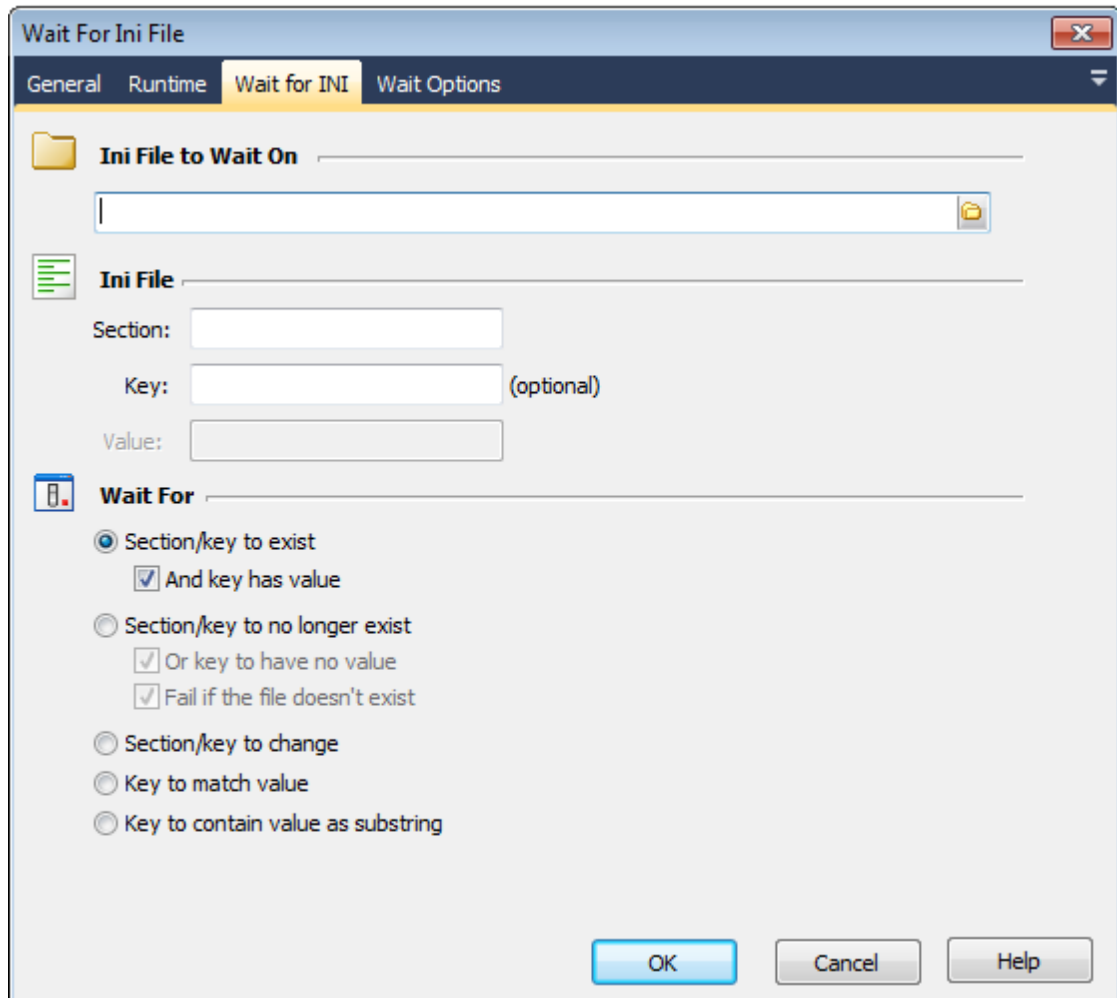
The project run will continue when the file has its Archive (A) bit set.

Wait Options

See the Waiting For Events topic.

6.13.16.3 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.

**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"

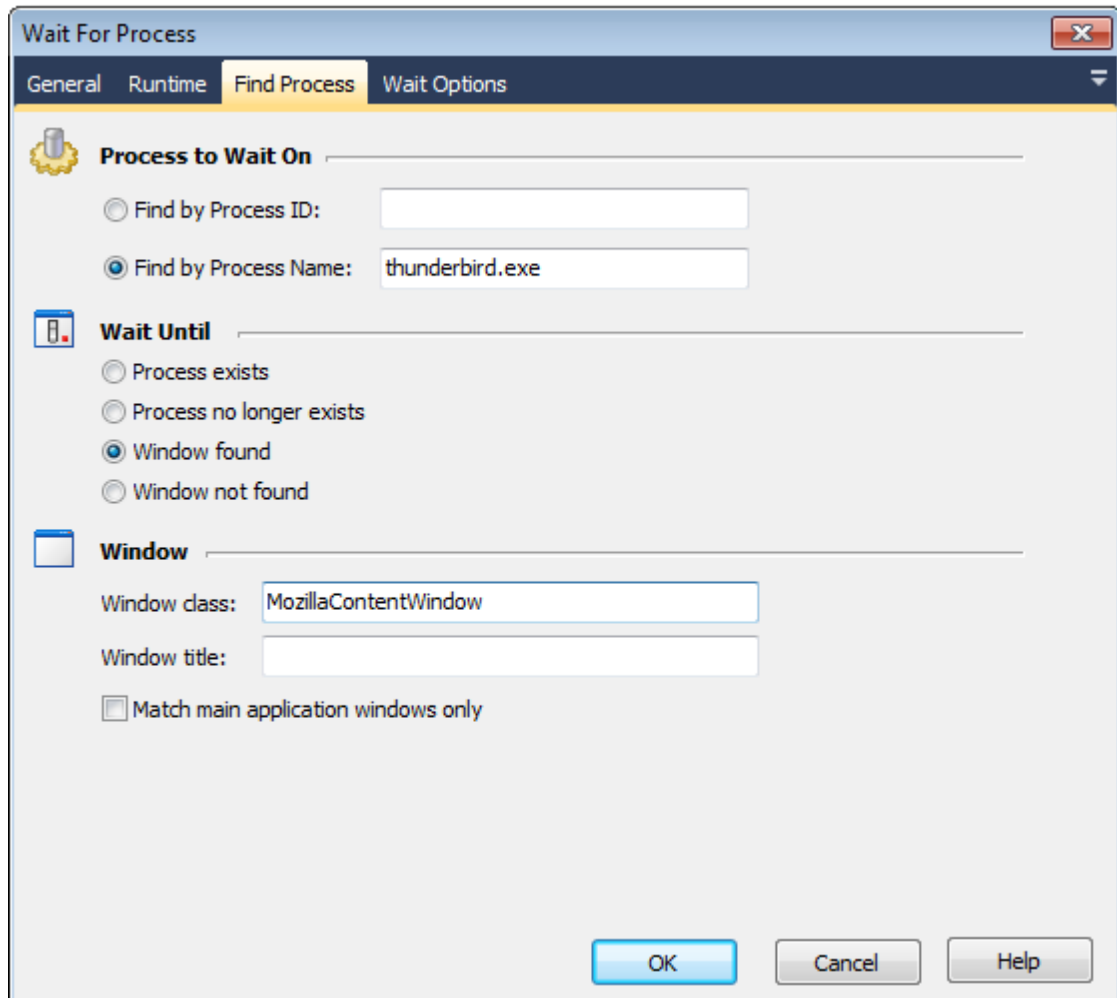
You need to specify a key name and a value to wait for.

The action will continue when the value for the key contains the substring you specify in the "Value" field.

6.13.16.4 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

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.

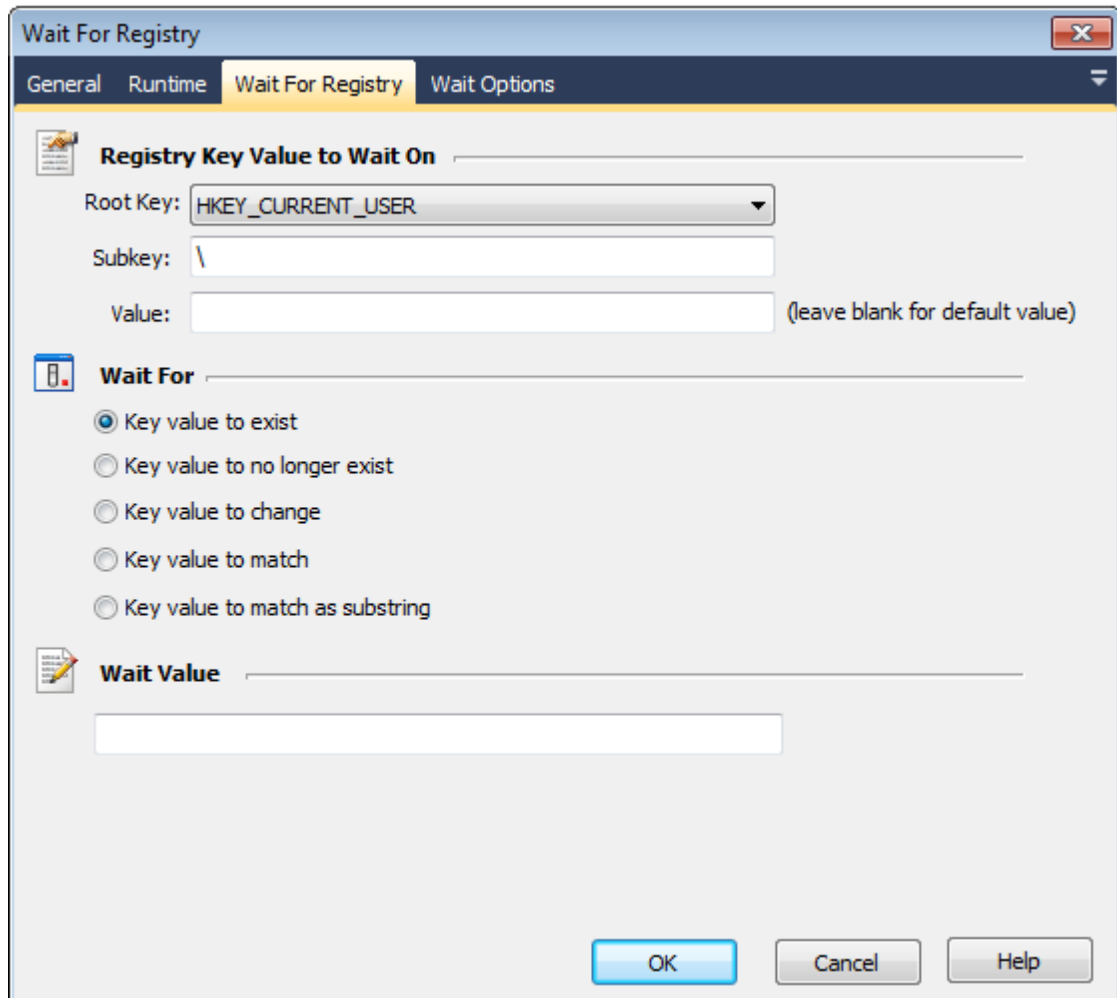
Check the *"Main application windows only"* box to only match main (ie parent) application windows. If the box is unchecked, all child windows will be searched.

Wait Options

See the Waiting For Events topic.

6.13.16.5 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

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.

6.13.16.6 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

General Runtime **Wait For Remote Computer** Credentials Wait Options

Remote Computer

Remote address:

☒ Fail if this host name cannot be found

Wait For

☒ Computer to respond to ping request (ping packet timeout ms.)

☐ WMI available and process running on remote host

☐ WMI available and process not running on remote host

☐ Network share available

WMI

Process Name:

☒ 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

OK Cancel Help

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, FinalBuilder 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, FinalBuilder 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

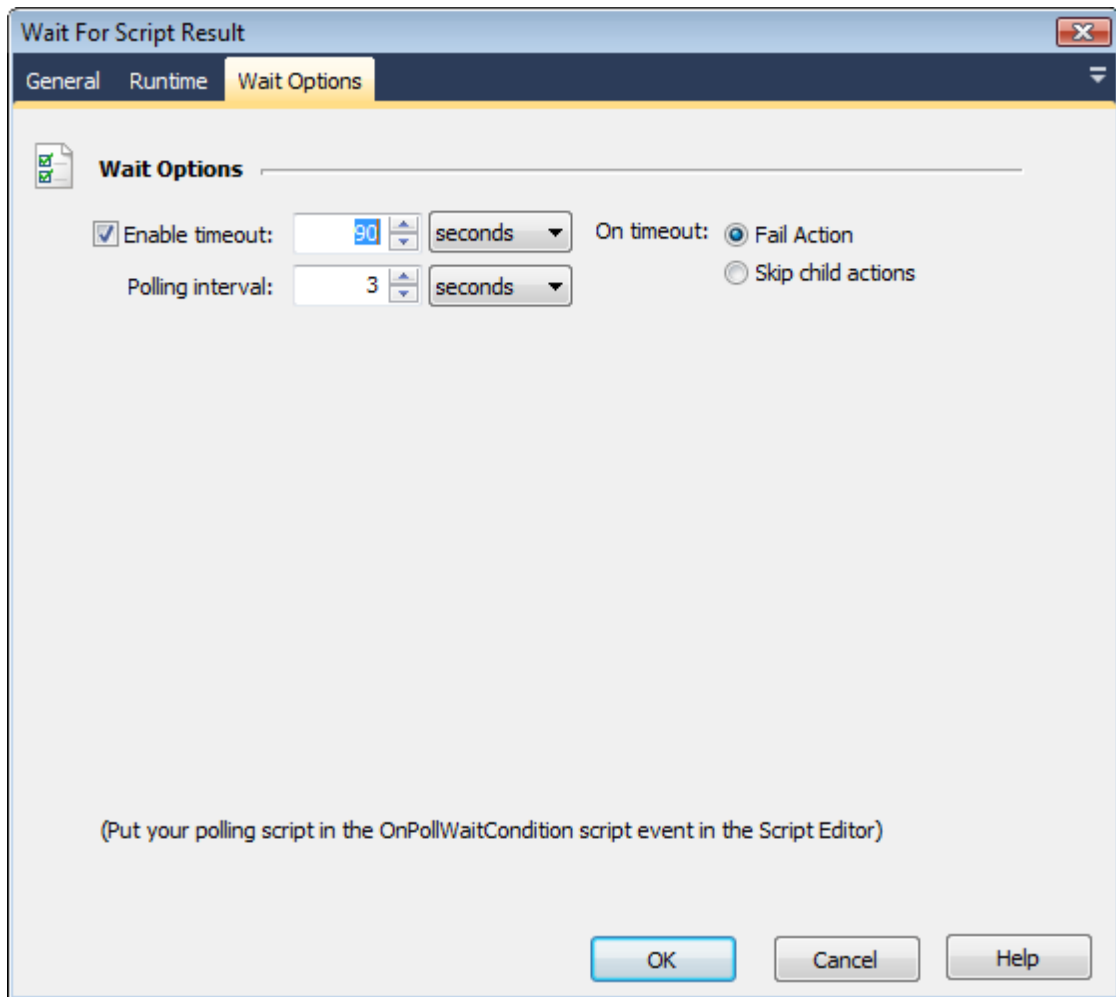
See this topic.

6.13.16.7 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.

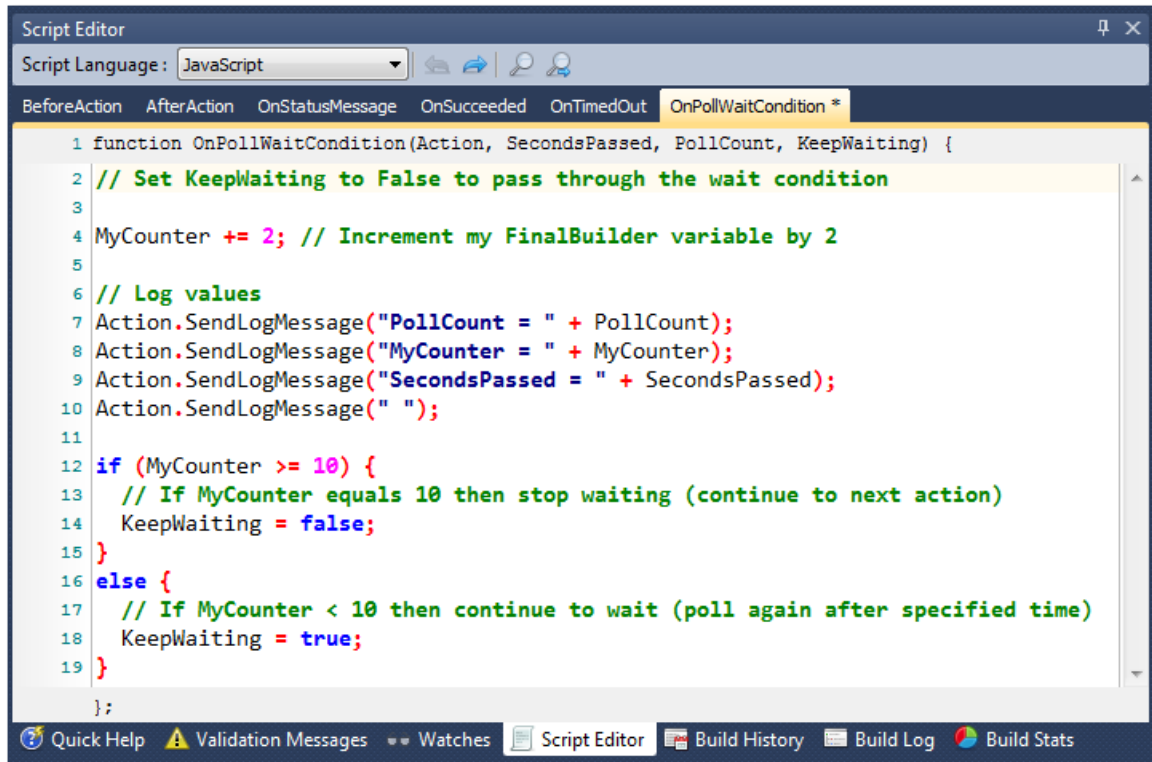
Wait Options

See the Waiting For Events topic for more information on Wait For actions.

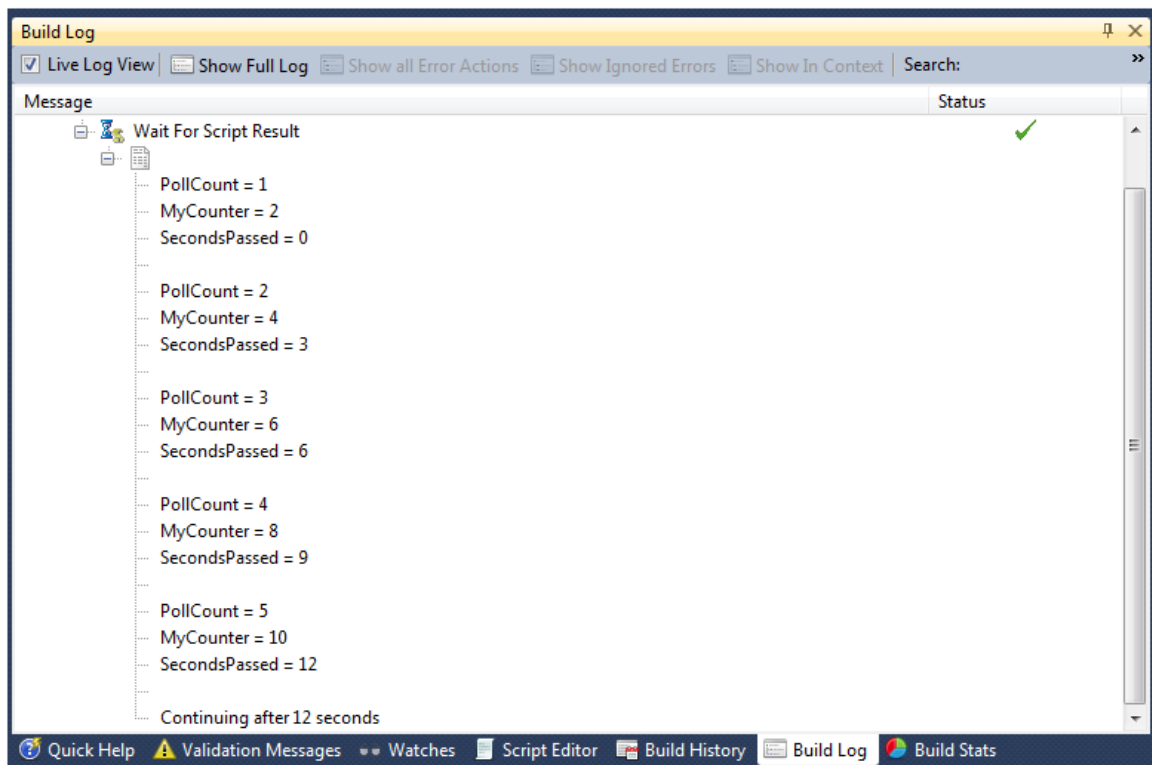


In this example the FinalBuilder variable *MyCounter* is incremented by 2 and then evaluated to see if the value is greater or equal to 10. If the value is less than 10 then the action will poll again after the specified period (set via the Wait Options tab of the action).

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.



The output would look as follows (assuming variable *MyCounter* was set to zero before the Wait For Script Result action commenced):



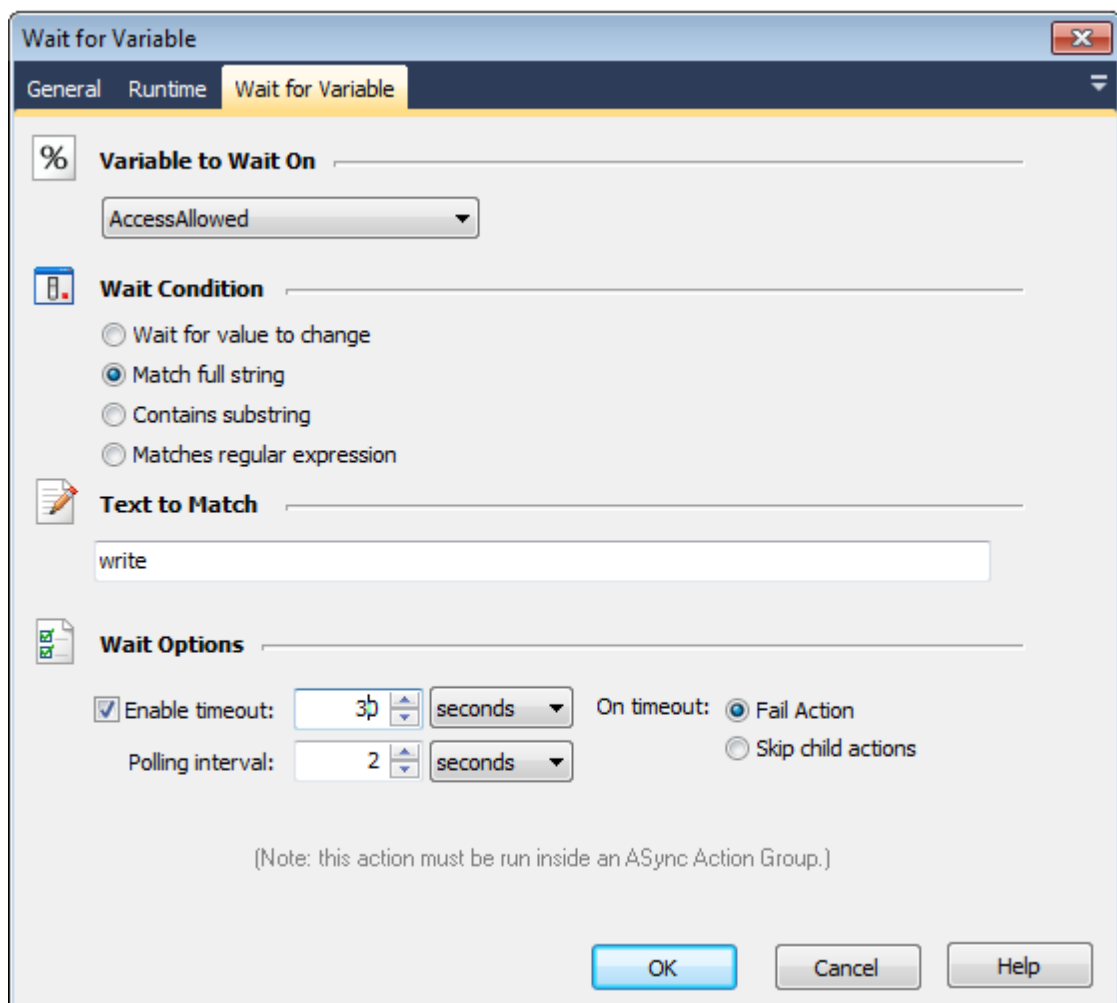
6.13.16.8 Wait For Variable

[FinalBuilder Professional Edition]

Wait for Variable pauses the project run until a given FinalBuilder 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.)



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

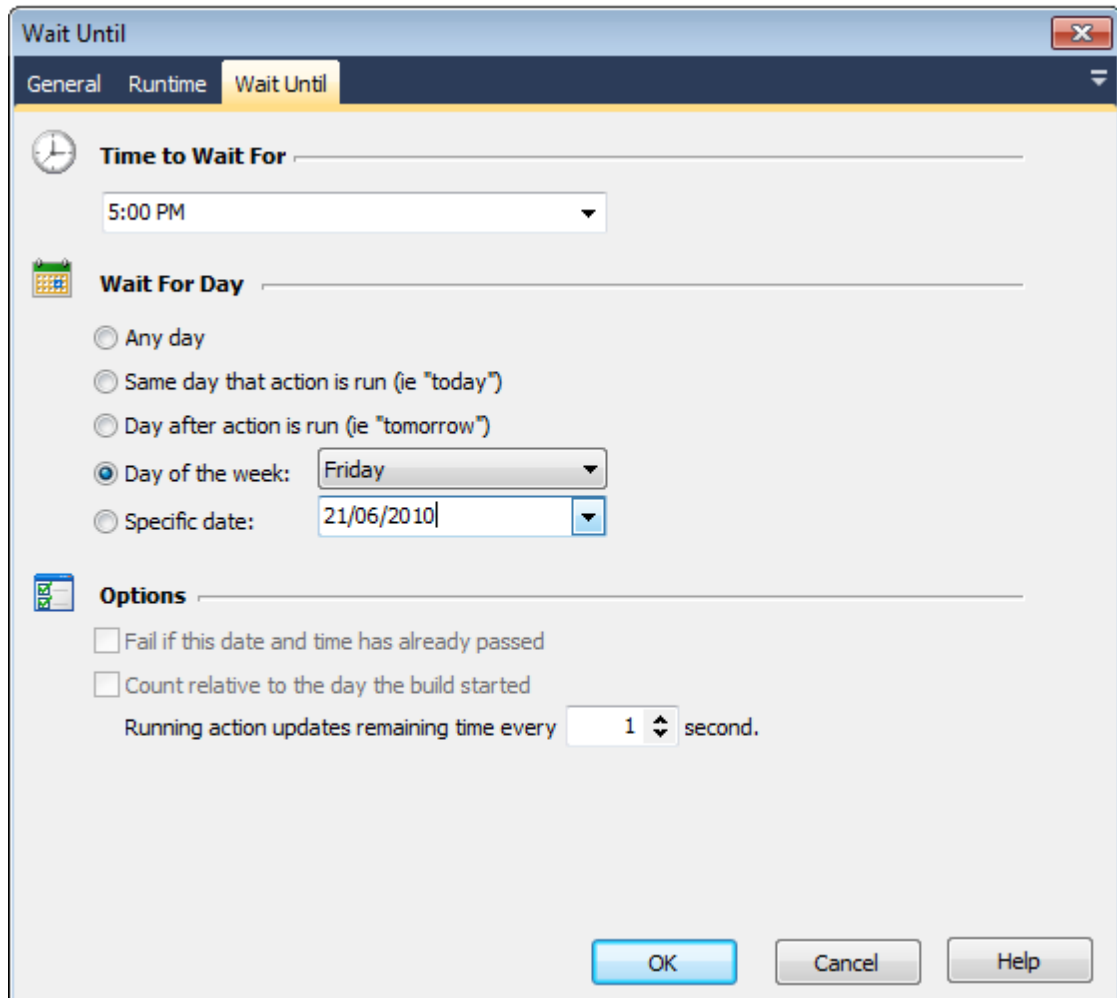
See the Waiting For Events topic.

6.13.16.9 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

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.

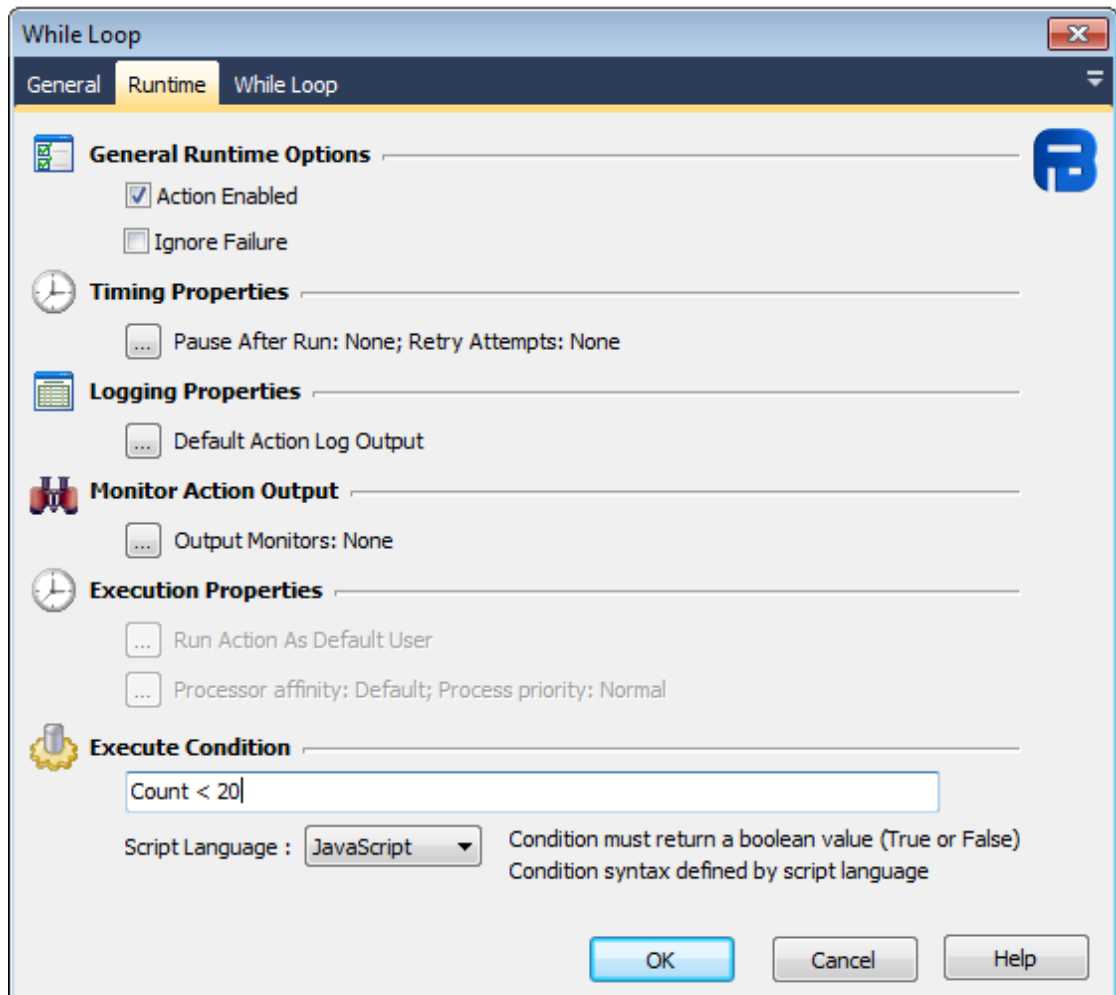
Running action updates remaining time every...

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.

6.13.17 While Loop Action

The While Loop action will run a set of actions (the child actions of the While Loop) while its Condition (specified on the Runtime tab) evaluates to True. The While action has no specific functionality other than its looping capability (specifically it tells the stepping engine which action to run next, overriding the default behaviour).



NOTE: Note that it will execute its script events (like any other action) first before executing its child actions. What this means is that the While Loop Action will execute its 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.

6.14 GUI Automation

6.14.1 Overview

[FinalBuilder Professional Edition]

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

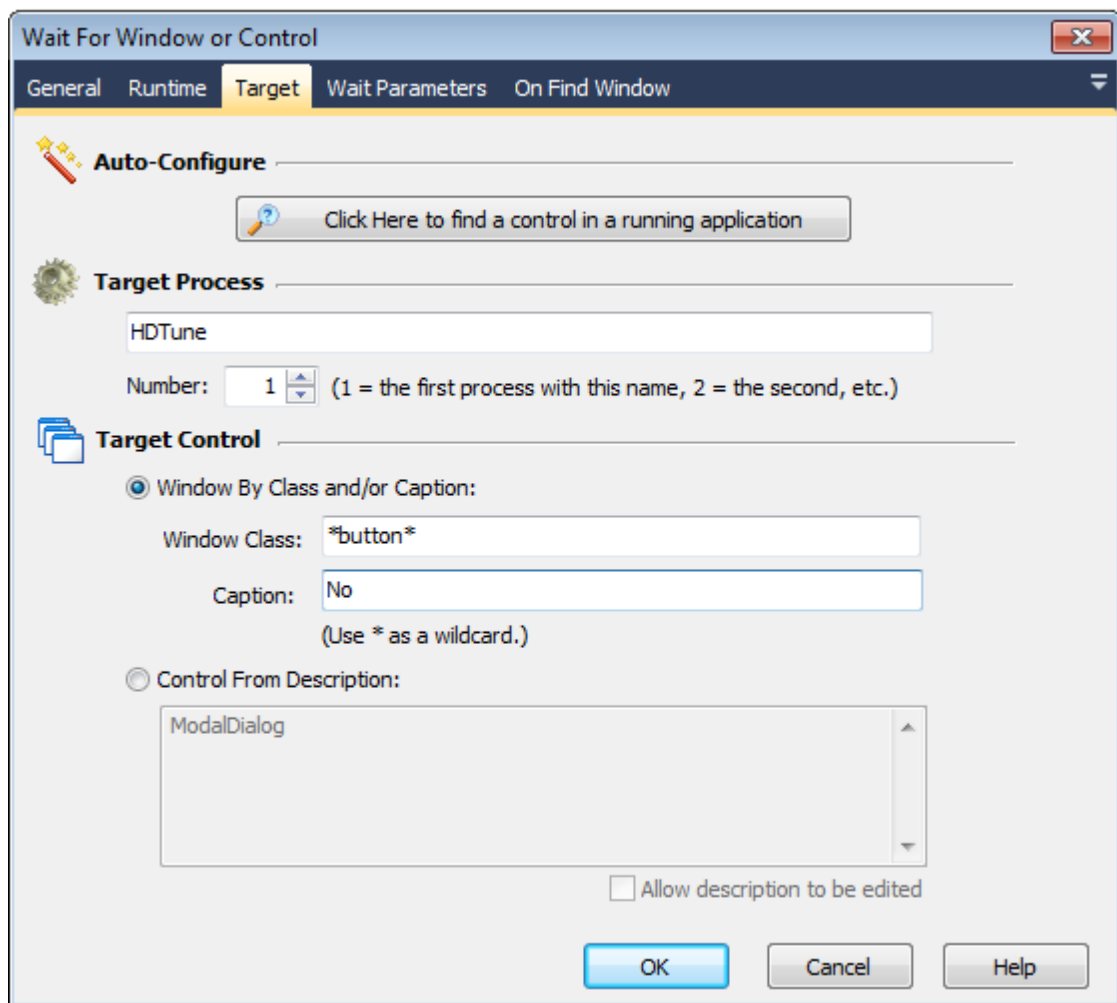
The actions use an object model called the "GUI Scripting Model" to perform these actions. Complete documentation for the Scripting Model will be made available in a future version of FinalBuilder.

6.14.2 Target Property Page

6.14.2.1 Automatic Configuration

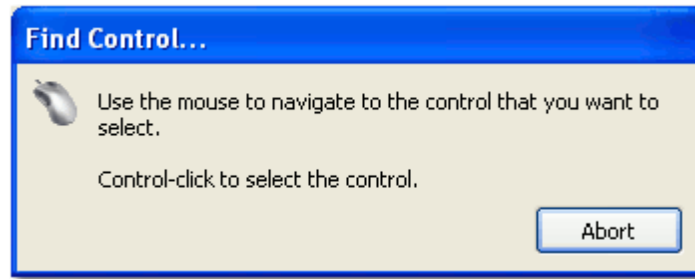
All of the GUI Automation Actions use the "Target" property page to describe the window or control that the action targets.

Here is an example of an automatically configured Target Property Page:



Auto-Configure

This is the recommended way to configure this property page. When you click on the "Click Here..." button, FinalBuilder is hidden and the following dialog pops up:



While you search, FinalBuilder 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.**

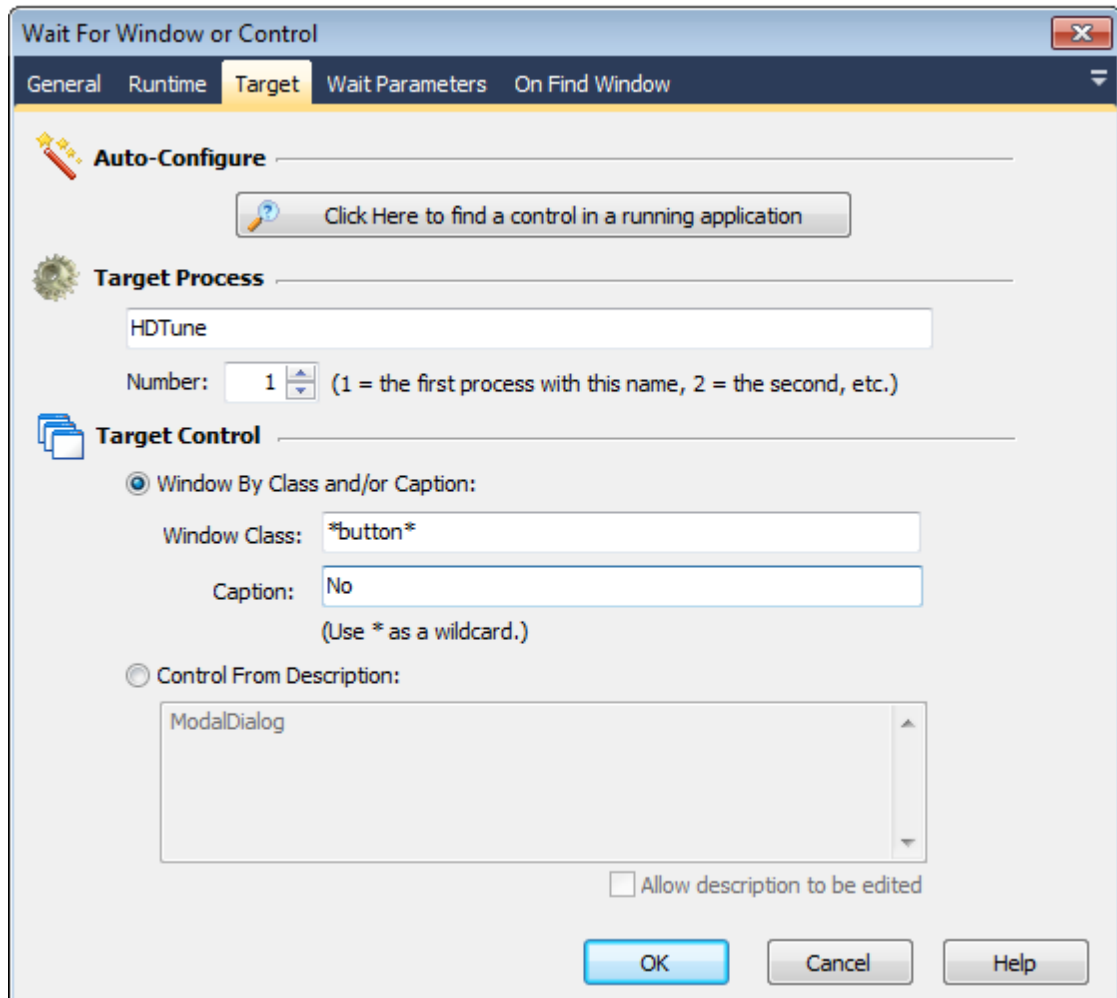
However, you can fine tune the selection by editing each field on the property page. This is described in the Manual Configuration topic.

6.14.2.2 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.



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:

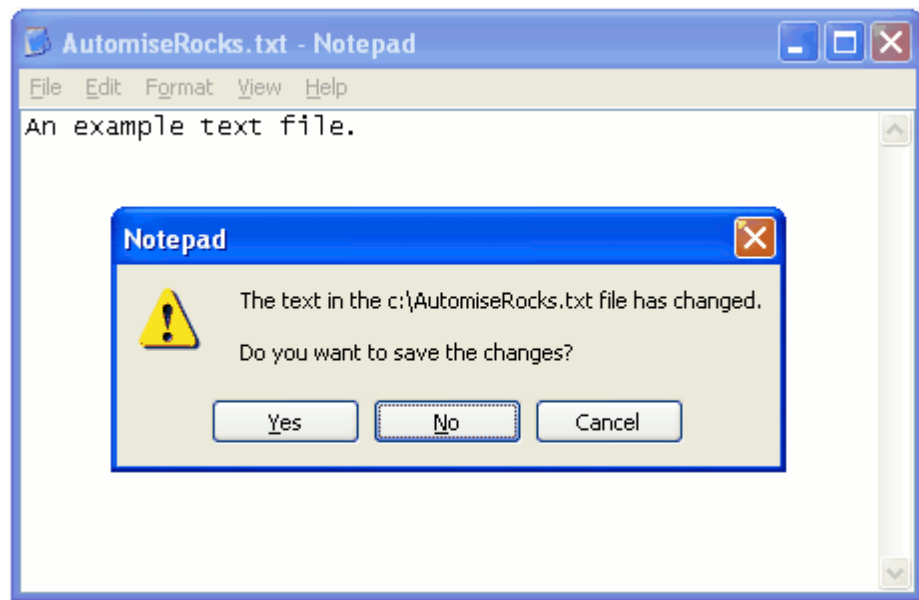
- The Window Caption. This is normally the text displayed in the window.
- The Window Class. This is normally set by the developer of the application.

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.

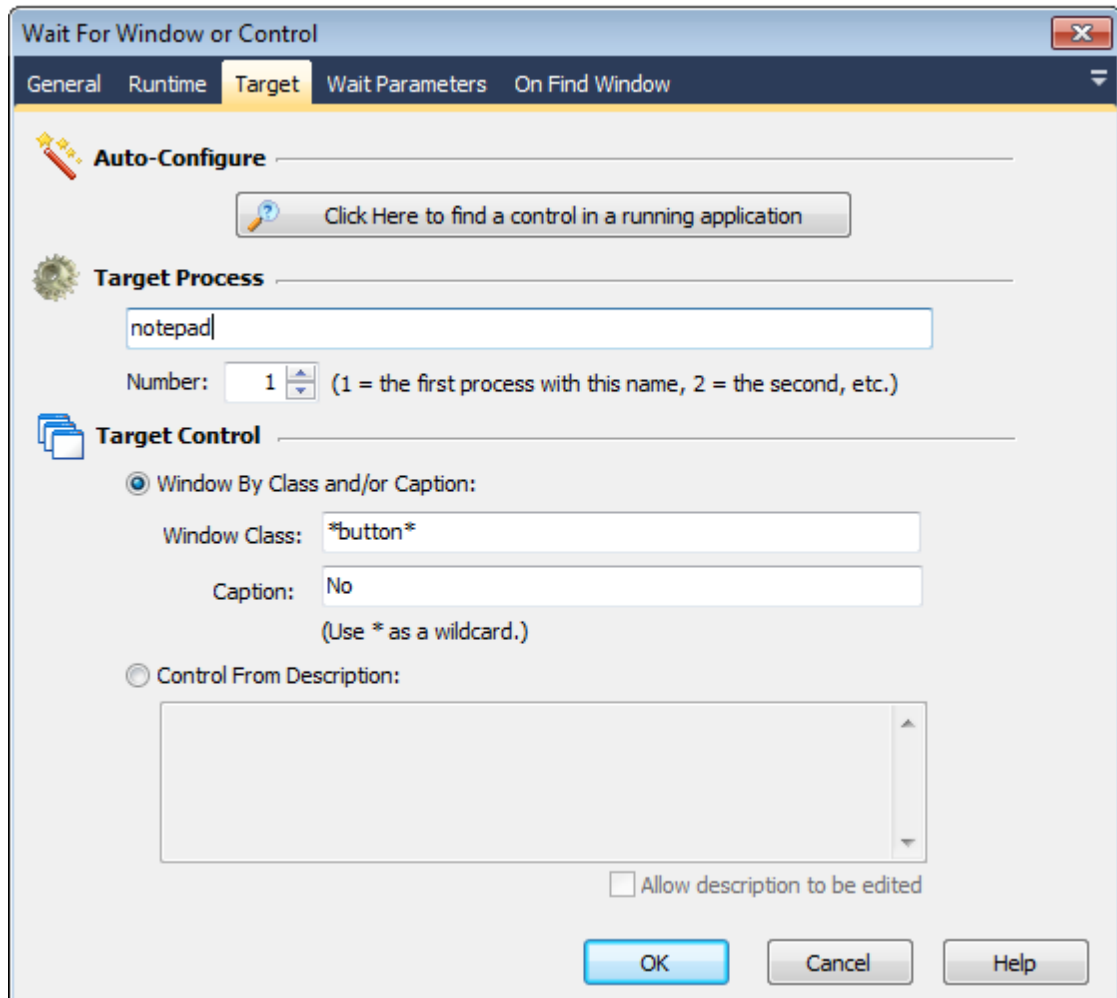
The target window can exist anywhere in the target process. Window classes and captions are not case sensitive. "*" is a wildcard which matches any string.

The following example will return the "No" button in this Notepad modal dialog:

Here is the modal dialog:



Here is the property page:

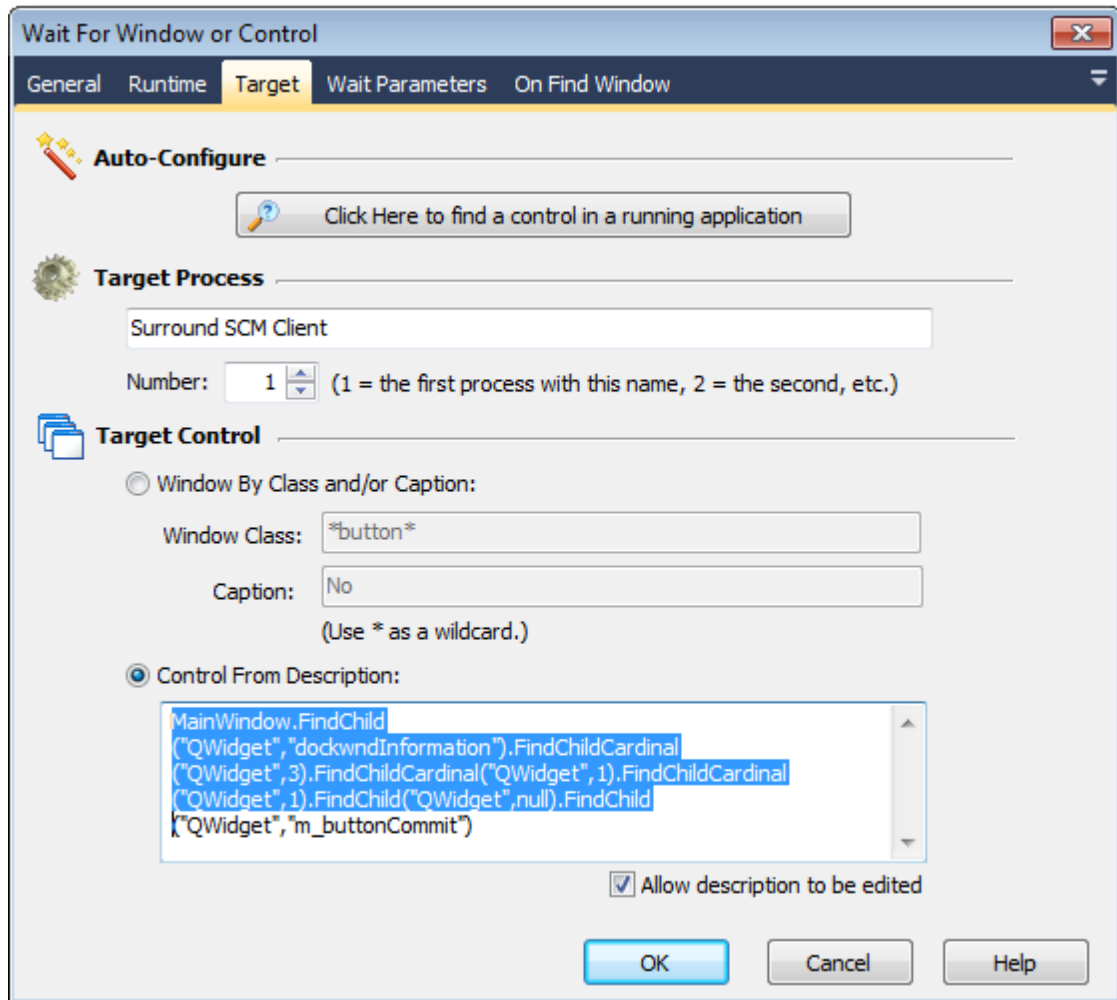


Control from Description

The Control From Description option uses a description string to explain to FinalBuilder 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, FinalBuilder 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:



Descriptions are specified using the structure of the FinalBuilder 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.

6.14.3 Actions

6.14.3.1 Perform Mouse Click

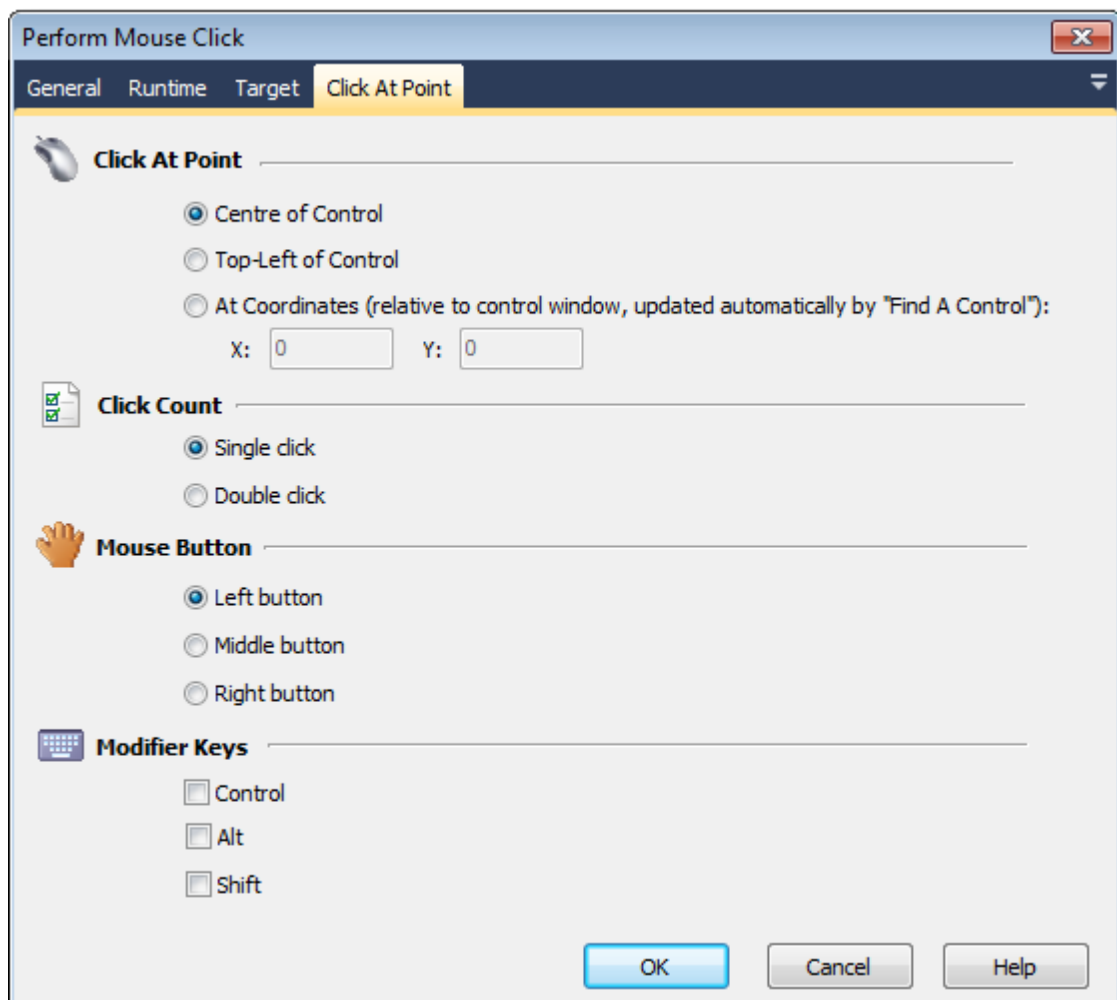
[Professional Edition Only]

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

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 FinalBuilder 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.

6.14.3.2 Perform Mouse Move

[Professional Edition Only]

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.

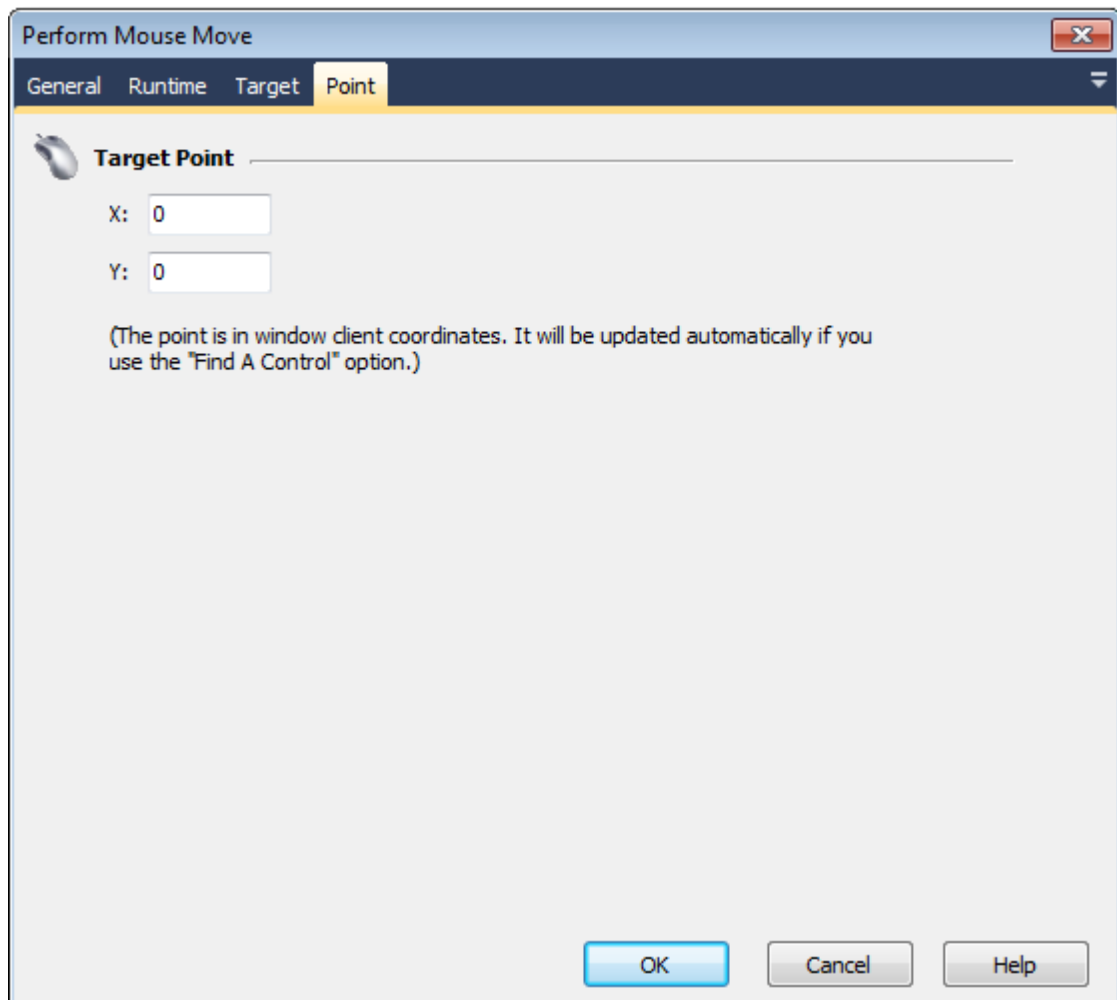
*Note: It is **not** necessary to use Perform Mouse Move each time you use Perform Mouse Click.*

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



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.

6.14.3.3 Send Keyboard Input

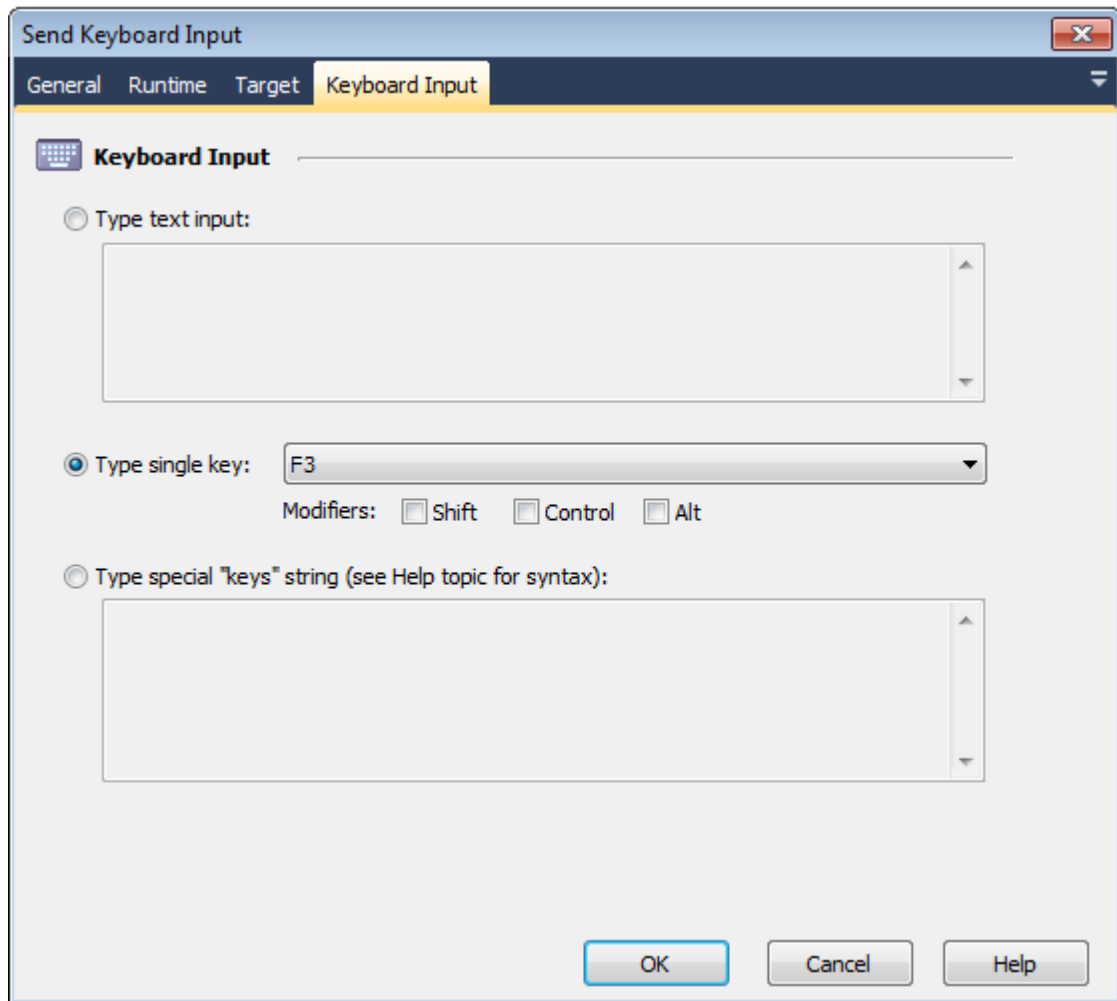
[Professional Edition Only]

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

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	{SCROLLLOCK}
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})".

6.14.3.4 Set Scrollbar Position

[Professional Edition Only]

The Set Scrollbar Position action allows you to move a scrollbar to a desired position.

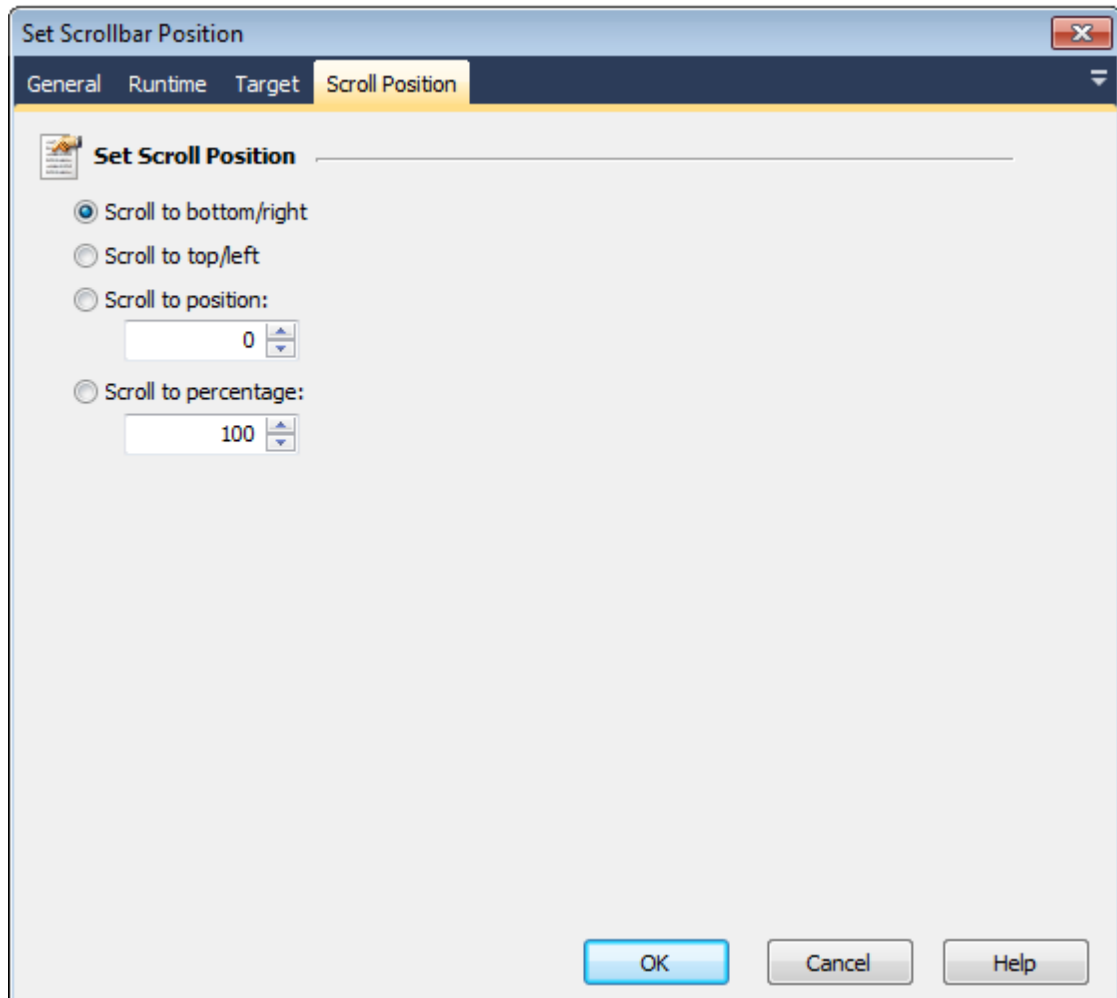
Any scrollbar which implements the standard Windows scrollbar behaviour can be moved (note that some non-standard scrollbars do not support this functionality.)

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**

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.

6.14.3.5 Wait For Window or Control

[Professional Edition Only]

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:

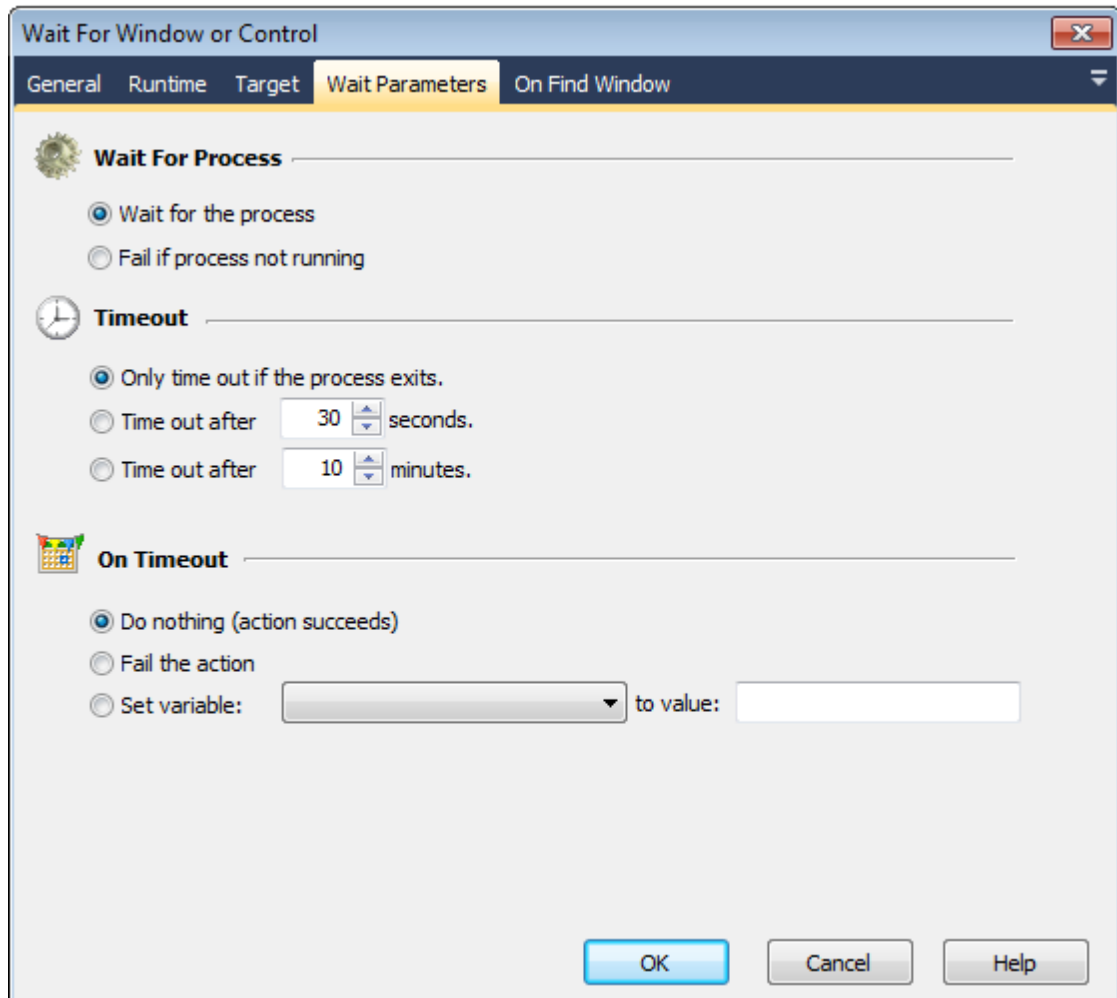
- Monitoring a running application in case it pops up an Error or Warning dialog.
- Waiting for a process to finish starting up.
- Waiting for a process to exit after an operation completes.

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, FinalBuilder 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"

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.

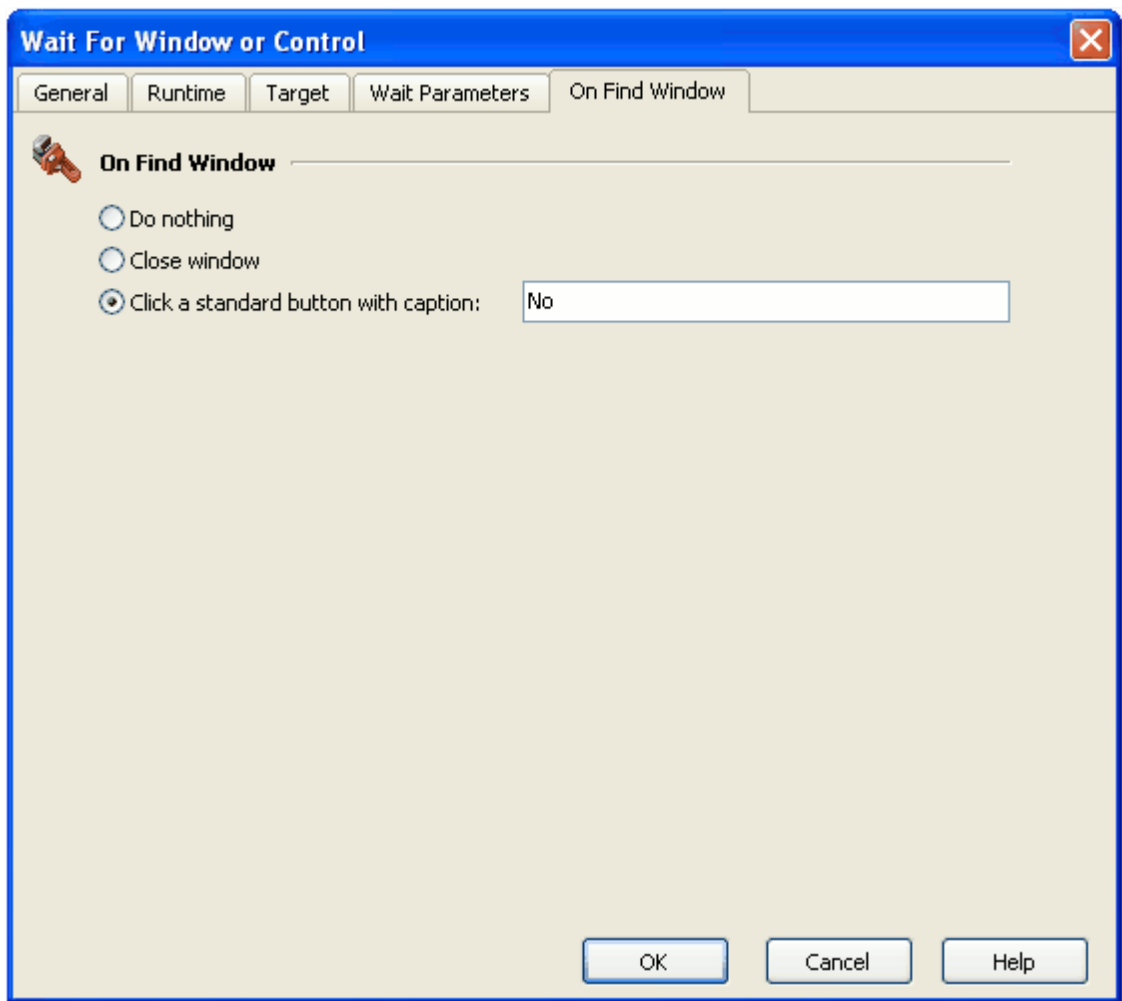
"Fail the action"

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



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..."

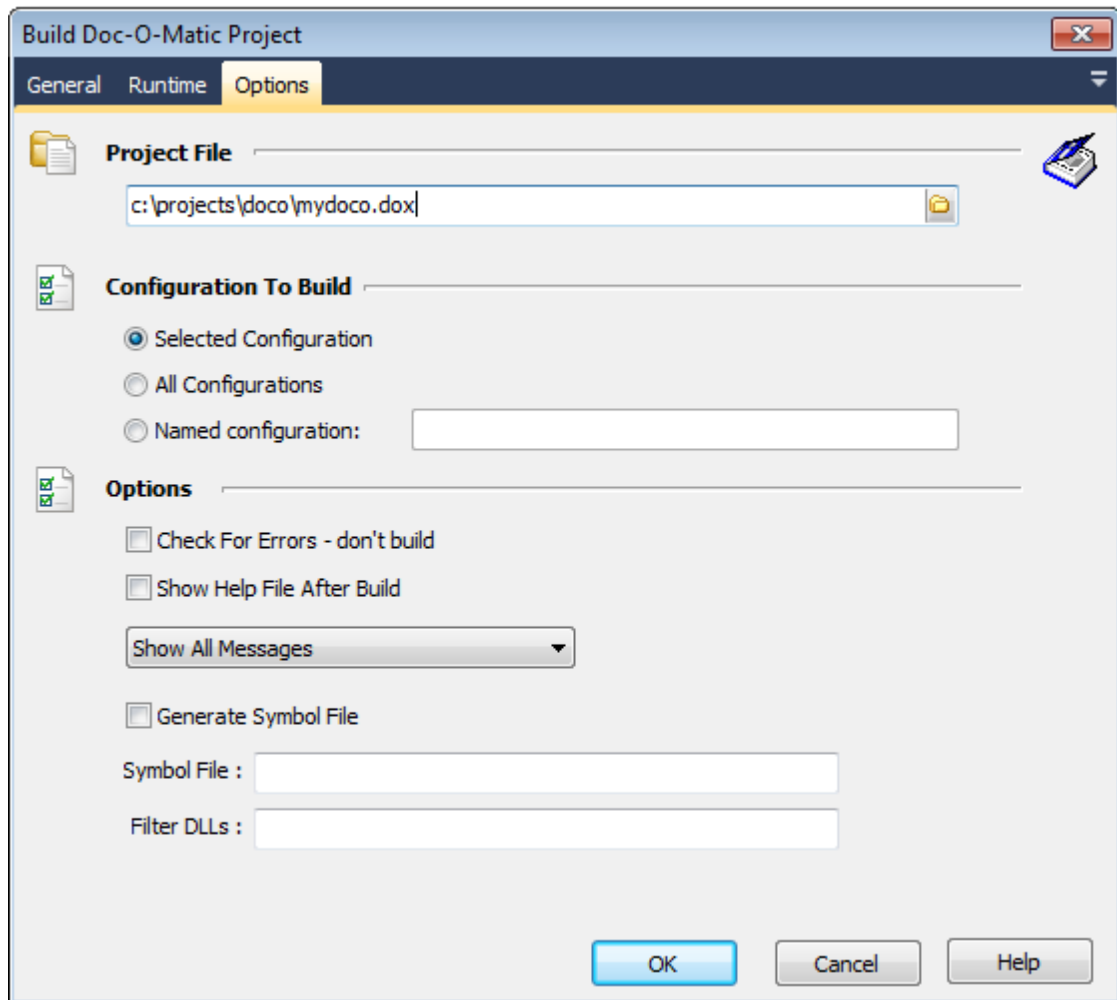
If the target control is a window or a modal dialog which contains "standard" Windows buttons, then this option will click on the button with the specified name. This is useful for modal warning or error dialogs which may need to be automatically dismissed.

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.

6.15 Help Compilers

6.15.1 Build Doc-O-Matic Project

This action provides an interface with the Doc-O-Matic command line compiler.



Project File : The Doc-O-Matic project file.

Check for Errors - Don't Build : The file will be checked for errors before starting and the build will be cancelled if any are found.

Show Help File After Build : Show the help file after build, not recommended

Reporting Level : Show All messages : Show Errors only : Show Hints, Warnings and Errors : Show Warnings and Errors.

Generate Symbol File : Alternate usage, generates a symbol file.

Symbol File : The fully qualified path to the generated symbol file.

Scripting Info

The Action properties available are :

property CheckForErrors : WordBool
property BuildAll : WordBool
property WarningLevel : integer // valid values are : wShowAll,
wShowHintsWarningsAndErrors, wShowWarningsAndErrors , wShowErrorsOnly

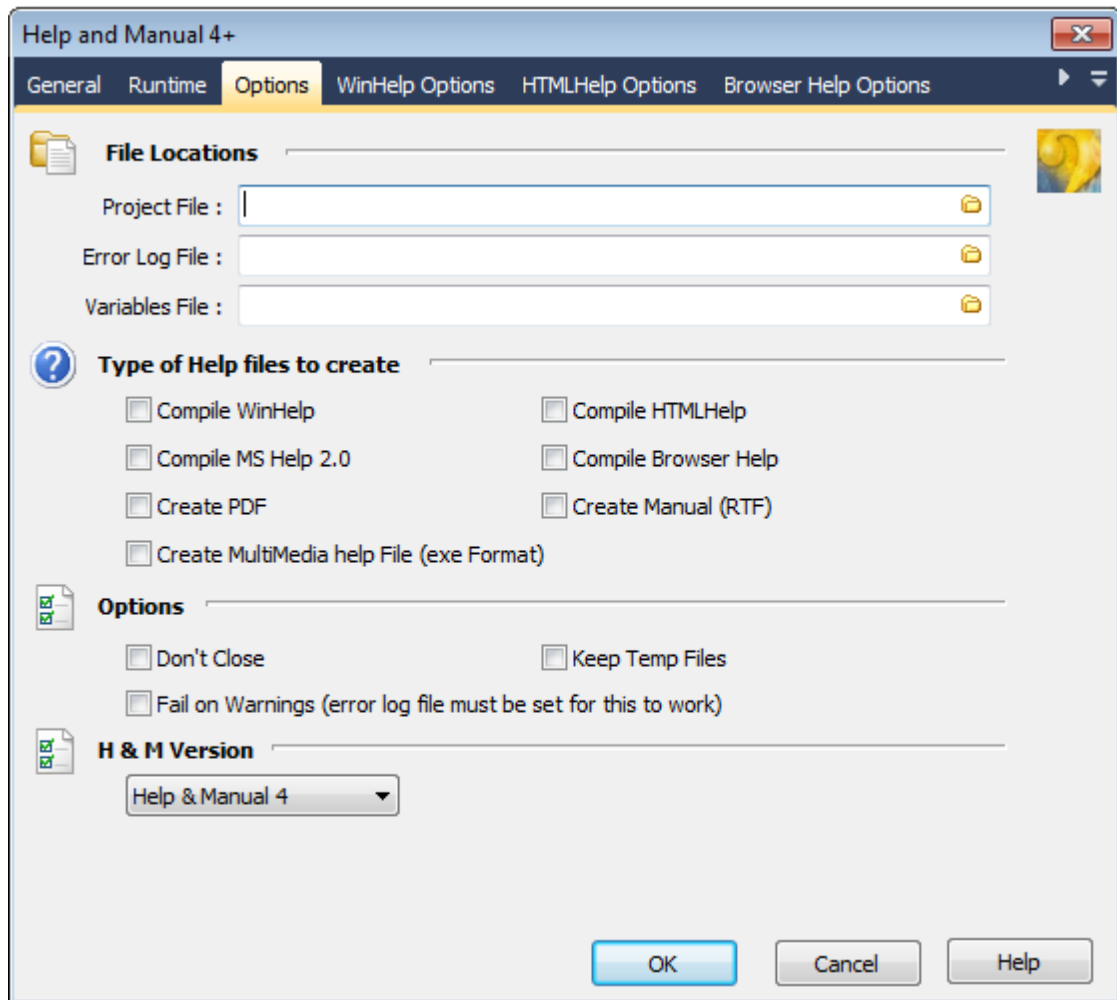
property ProjectFile : WideString
property ShowAfterBuild : WordBool
property SymbolFile : WideString
property GenerateSymbolFile : WordBool

Doc-O-Matic can be found at <http://www.toolsfactory.com/>

6.15.2 Help & Manual Action

This Action enables the automated builds of Help & Manual projects - you can build multiple output formats in one action.

Help & Manual is a product of [ECSoftware](#) and is the tool we use and recommend for help file and manual generation.



Project File

The Help & Manual project file to compile.

Error Log File

Outputs the compile log to the specified text file.

Variables File

An external text file, which redefines some or all the variables in the specified project file. The file must only contain one definition per line.

Types of Help Files to Create

A list of the available project types to be built. More than one type of help file can be built. The specific options for each Help file type can be found under the relevant options tab.

Don't Close

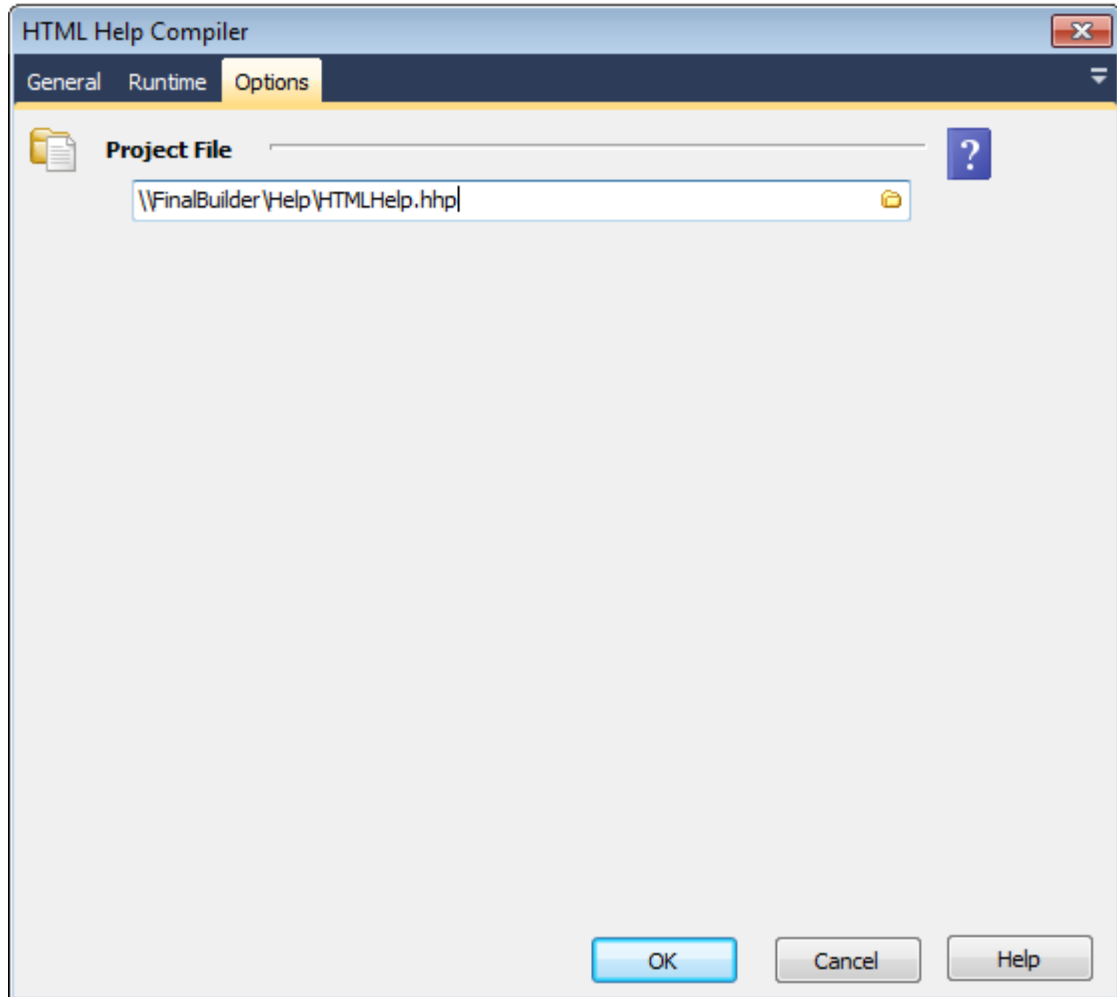
The Help & Manual program will be left running after the action finishes.

Keep Temp Files

Ensures the files generated by the 'HTML Help', 'WinHelp' and 'Visual Studio Help' help file compilers are not deleted.

6.15.3 HTML Help Compiler

This action provides an interface to the HTML Help Compiler. Note that to use this action, you should first set the HTML Help Compiler location in the Options dialog.



Project File : The fully qualified path to the html help project file (hhp).

Scripting Info

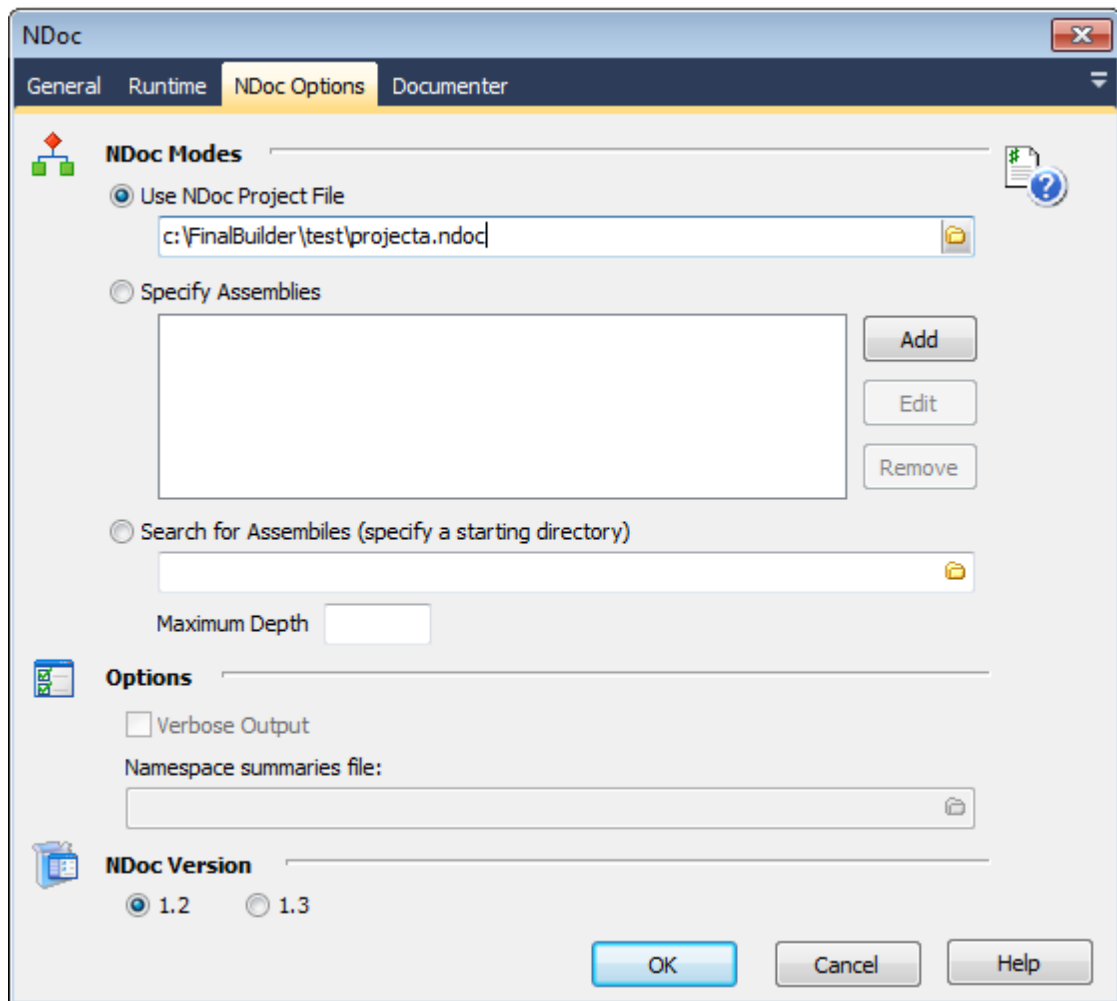
The Action properties available are :

property ProjectFile : WideString

6.15.4 NDoc Action

NDoc generates class libraries documentation from .NET assemblies and the XML documentation files generated by the C# compiler (or an add-on tool for VB.NET).

NDoc uses add-on documenters to generate documentation in several different formats, including the MSDN-style HTML Help format (.chm), the Visual Studio .NET Help format (HTML Help 2), and MSDN-online style web pages.

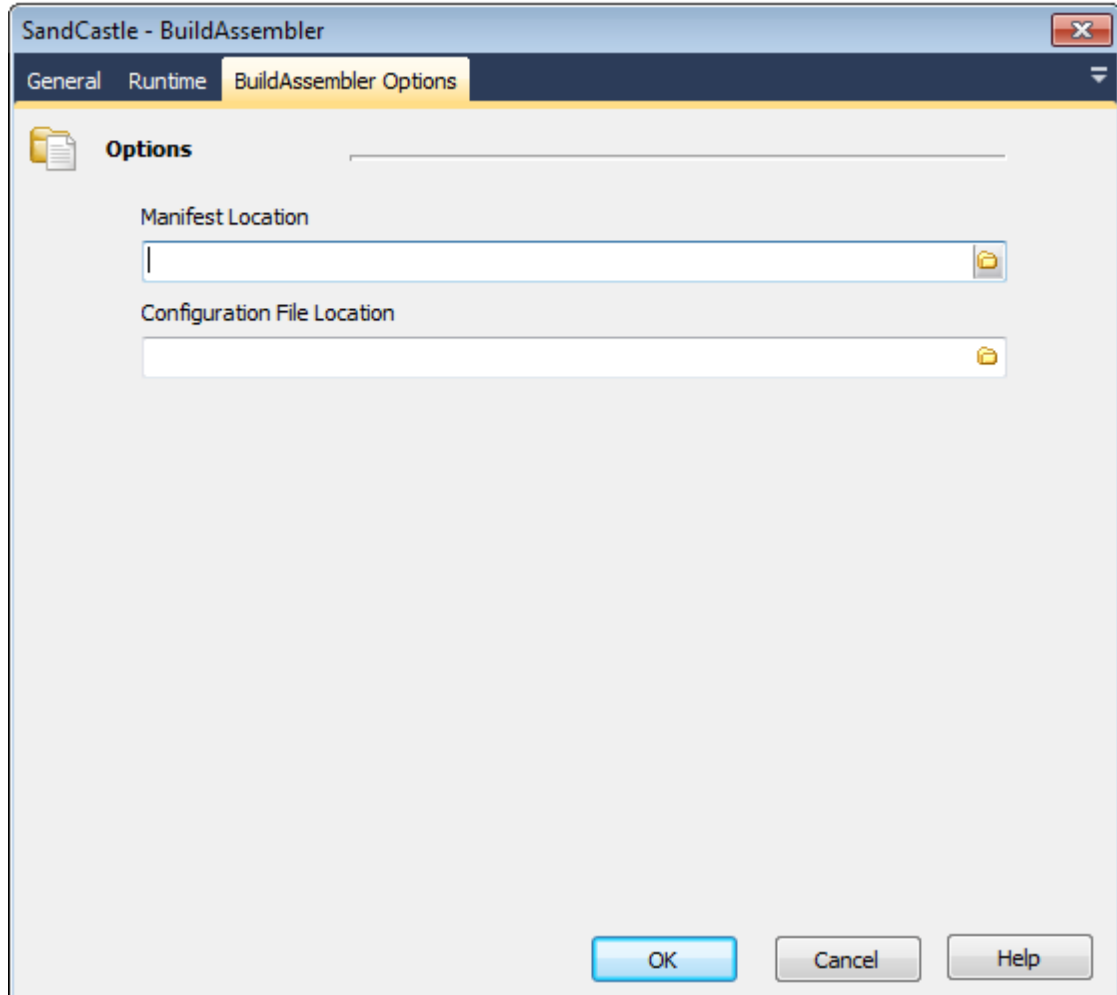


For more information see the NDoc homepage:

<http://ndoc.sourceforge.net/wiki>

6.15.5 Sandcastle - BuildAssembler

The Sandcastle - BuildAssembler action enables you to collate information from comment files and reflection information files into a set of HTML pages.

**Manifest Location**

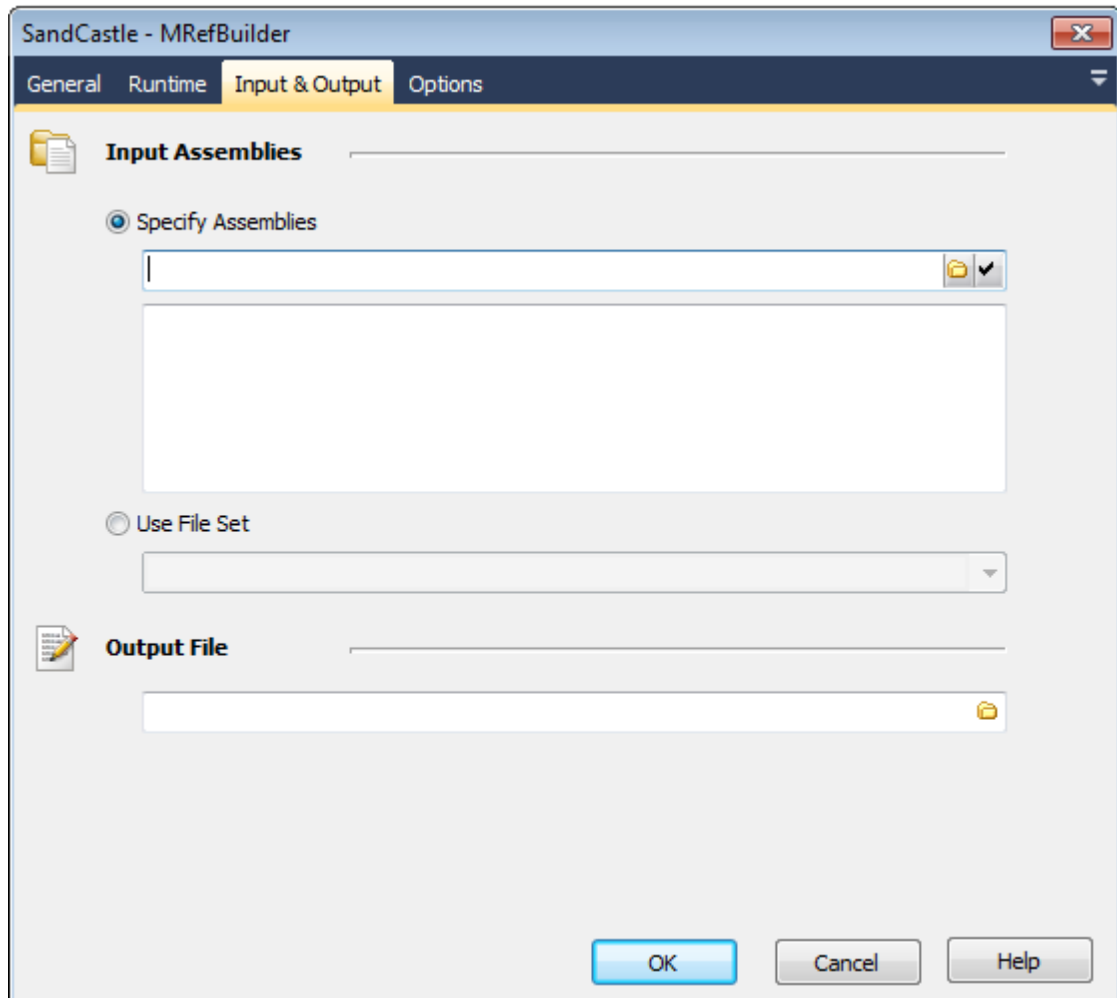
The full path to the manifest file.

Configuration File Location

The location of the configuration file, which is where the component stack is defined. The directory that contains the configuration file will be used as the working directory for the operation.

6.15.6 Sandcastle - MRefBuilder

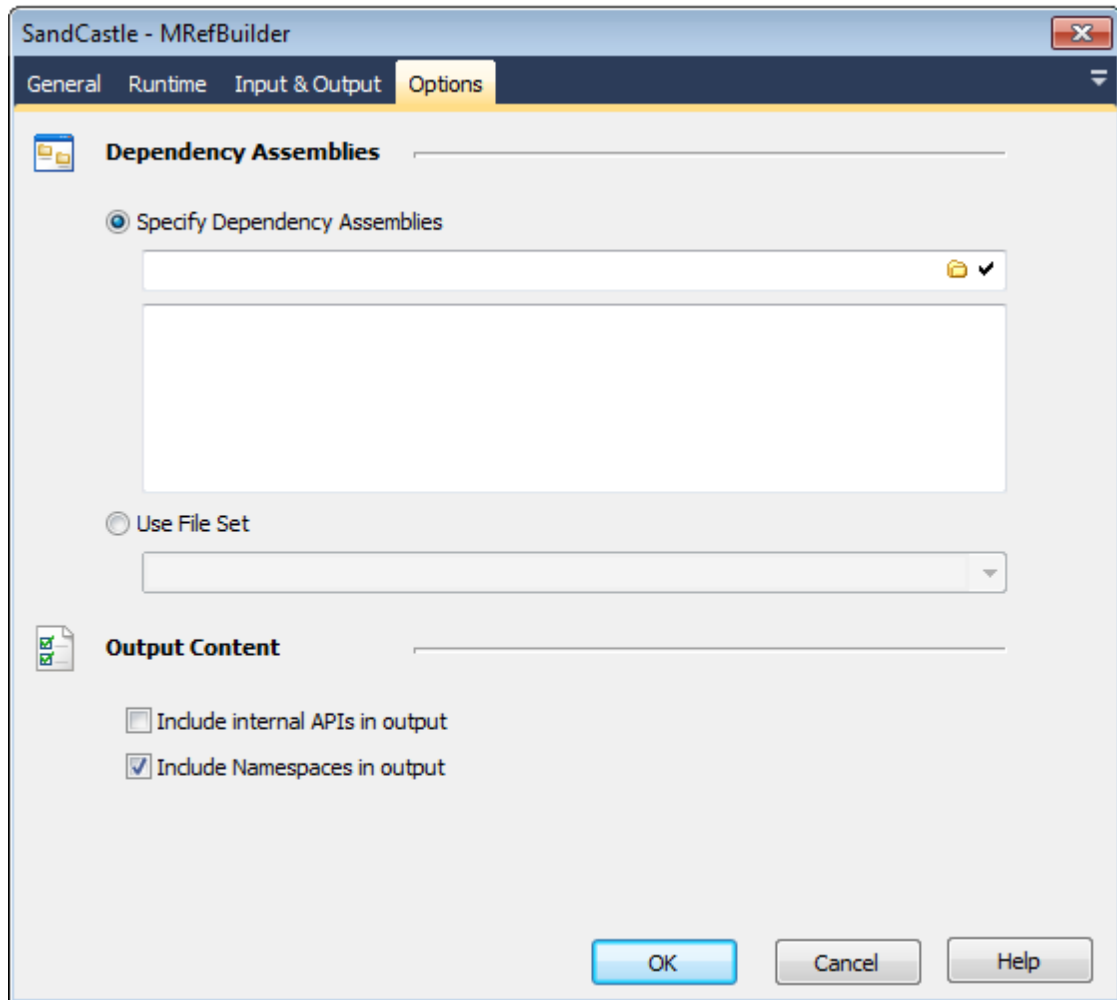
The Sandcastle - MRefBuilder action enables you to reflect over managed assemblies to produce an XML-formatted reflection information file.

**Input Assemblies**

One or more assembly files to reflect over to produce an XML-formatted reflection information file. The assemblies can be specified on the property page or from a file set.

Output File

The filename for the reflected information file.

**Dependency Assemblies**

The assemblies that are depended on by the input assemblies. These can be specified on the property page or from a file set.

Include Internal APIs in Output

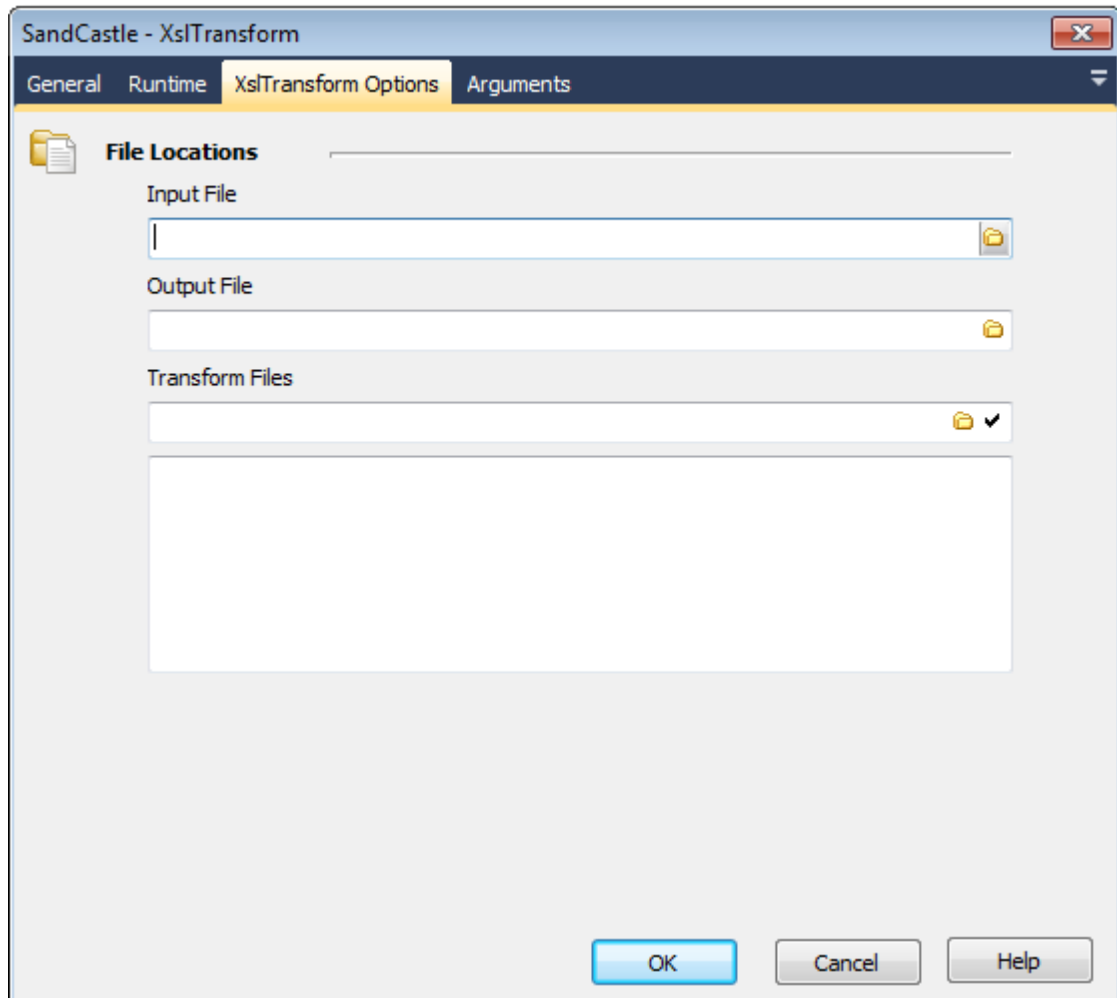
Determines whether the output will include API's that are only visible internally.

Include Namespaces in Output

Determines whether namespace information is included in the output.

6.15.7 Sandcastle - XslTransform

The Sandcastle - XslTransform action enables you to apply XSL transforms to XML files.

**Input File**

The XML file that will be transformed by XSL transform file.

Output File

The file that will hold the transformed data.

Transform Files

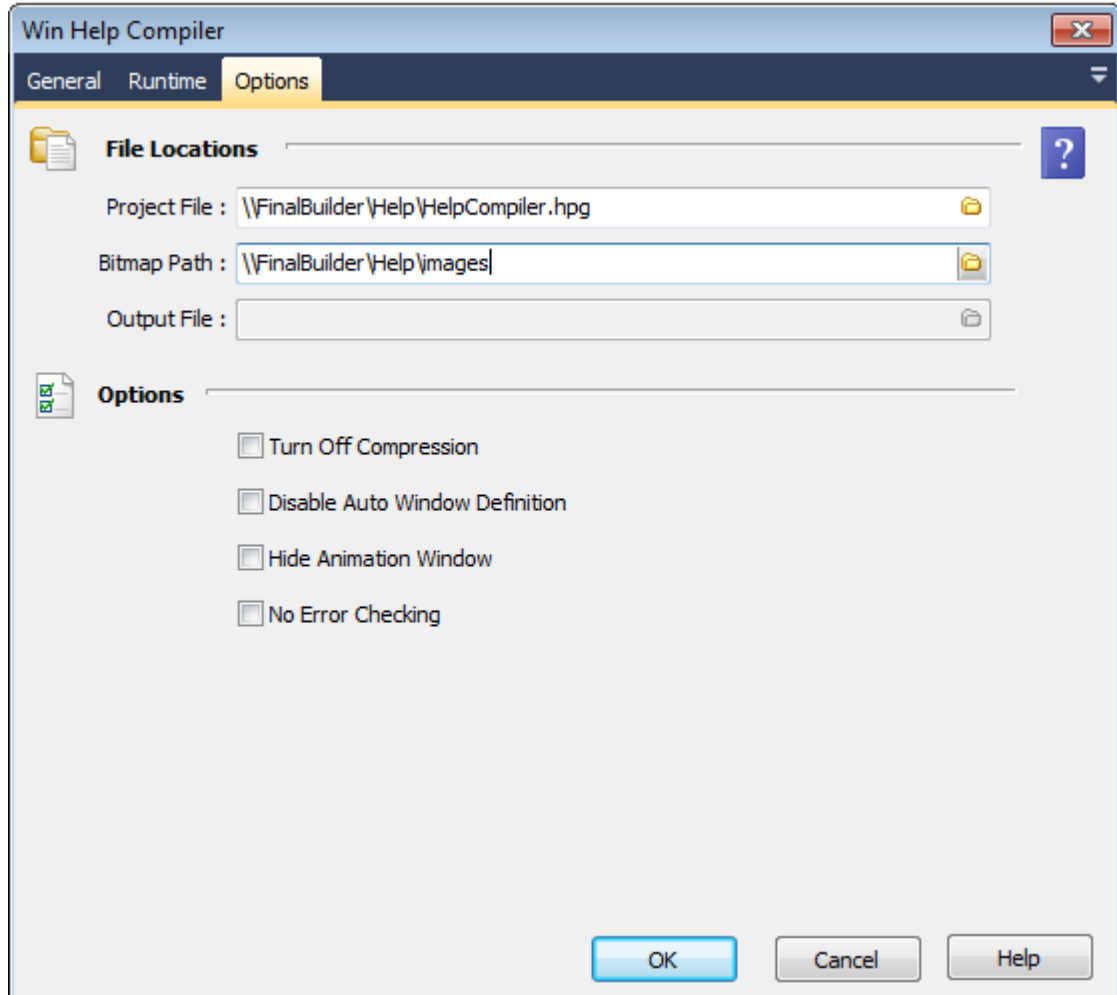
One or more files to use to transform the input file. When using multiple transform files, the output of the previous transform is used as the input for the next.

Arguments

Extra arguments to provide to the XslTransform tool.

6.15.8 WinHelp Compiler

This action provides an interface to the Windows Help Compiler. Note that to use this action, you should first set the WinHelp compiler location in the Options dialog.



Project File : The fully qualified path to the winhelp project file (*.hpg) or Rich Text File (*.rtf)

Bitmap Path : The search path for bitmaps referenced in the help project. Separate entries with a semi-colon.

Turn Off Compression : Instructs the compiler not to compress the help file.

Hide Animation Window : Hides the animated window while compiling

Disable Auto Window Definition : Prevents the automatic Creation of window definition, if a window specified in a hotspot or macro was not defined in the project file.

No Error Checking : Specifies that there are no errors in the rtf files. Some error checking will be turned off to speed up the compilation process. This option should only be used where compilation speed is crucial.

Output File : Specifies the name of the hlp file to create. Use this only when not compiling an hpj file.

Scripting Info

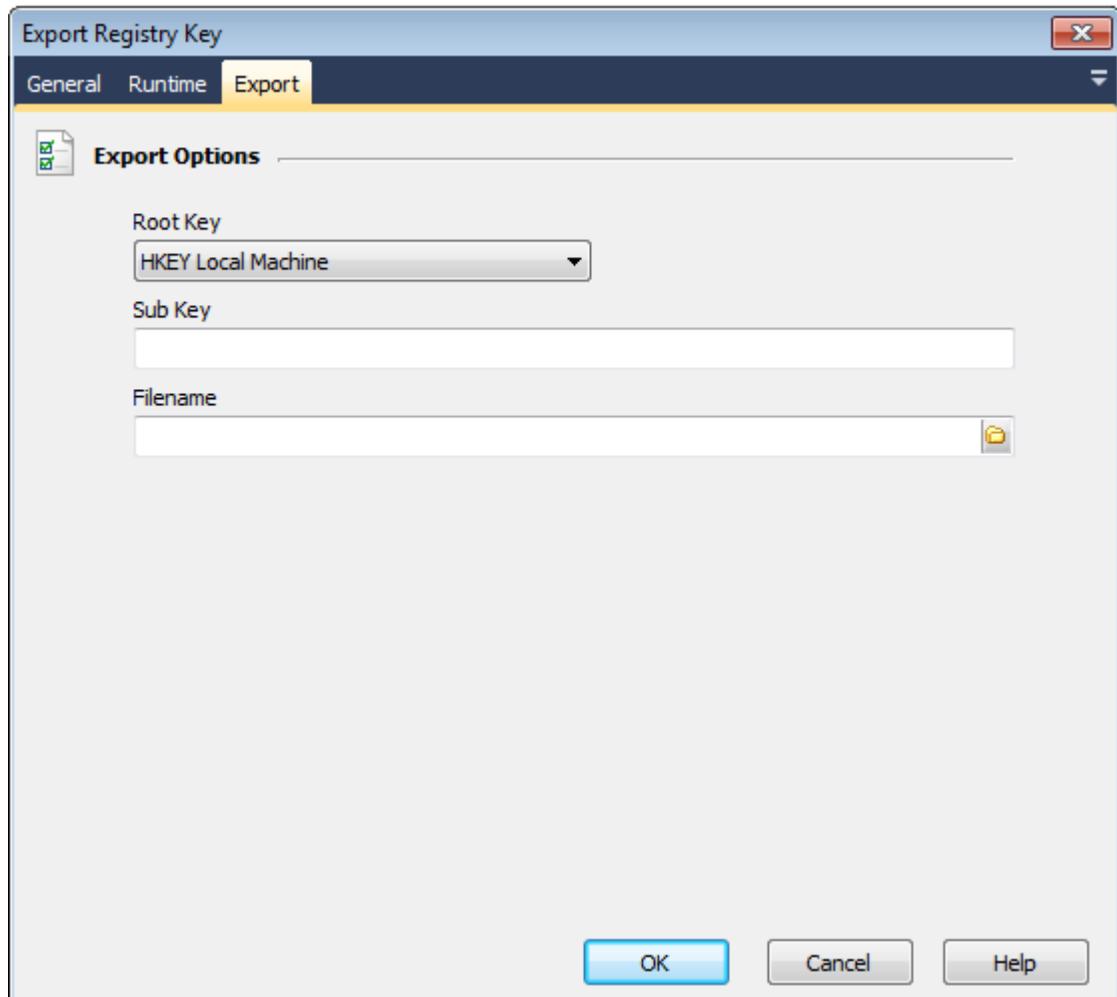
The Action properties available are :

property	ProjectFile : WideString
property	BitmapPath : WideString
property	TurnOffCompression : WordBool
property	HideAnimationWin : WordBool
property	DisableAutoWinDef : WordBool
property	NoErrorCheck : WordBool
property	OutputFile : WideString

6.16 Ini Files & Registry

6.16.1 Export Registry Key

The Export Registry Key action enables you to export a registry key as registration (.reg) file.

**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

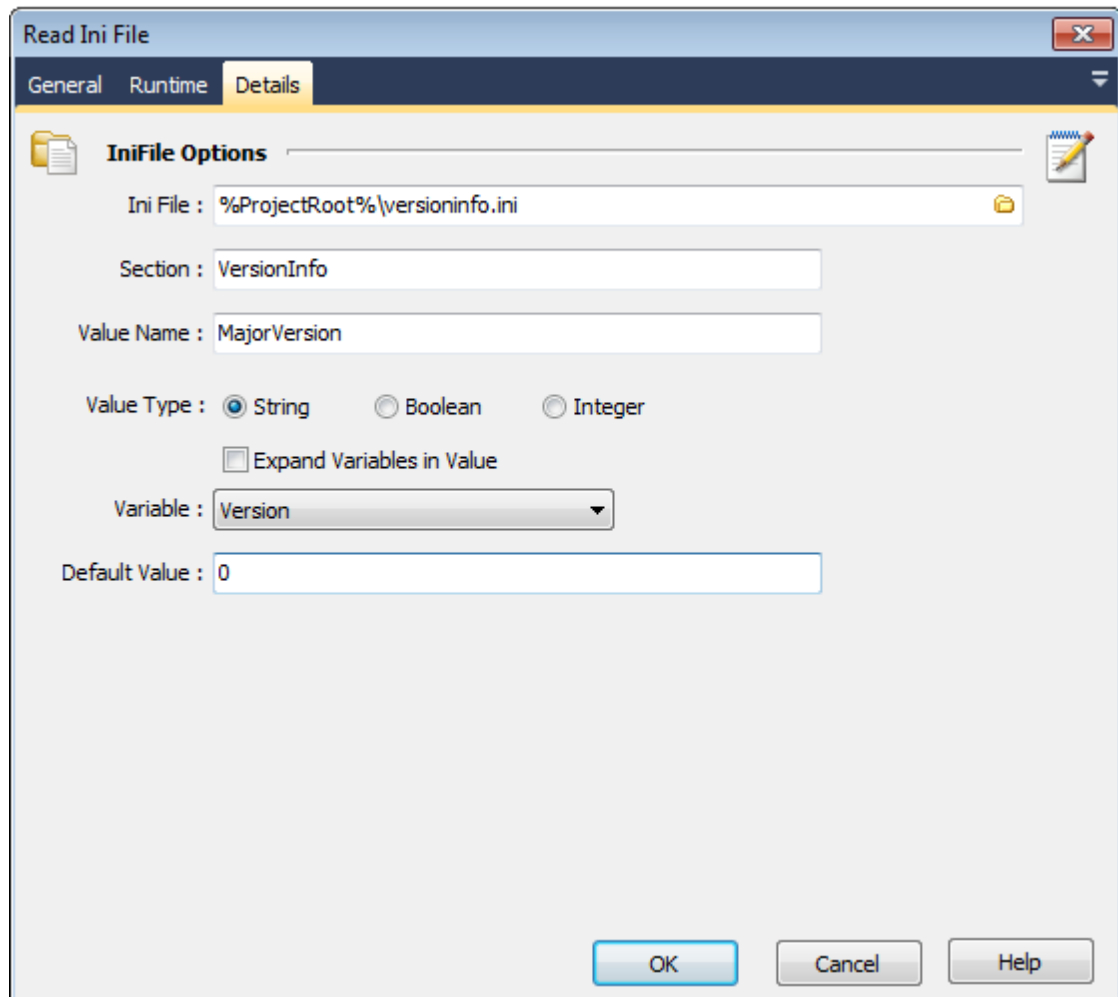
The remaining path to the key that is to be exported. For example 'Software\Microsoft\Windows'.

Filename

The registration file that the key will be exported to.

6.16.2 Read Ini File

This action allows you to read values from an ini file into FinalBuilder variables.



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 FinalBuilder 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

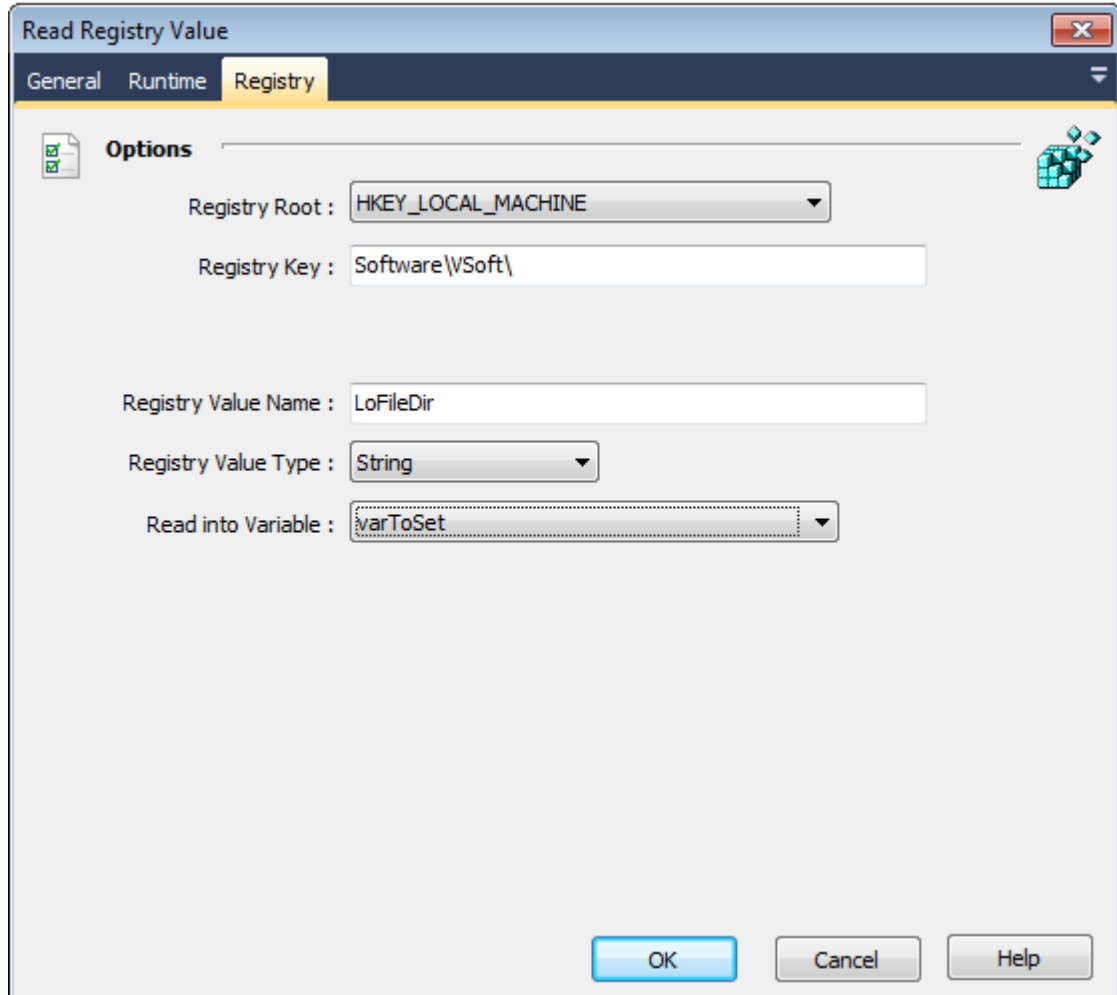
property VariableName : WideString

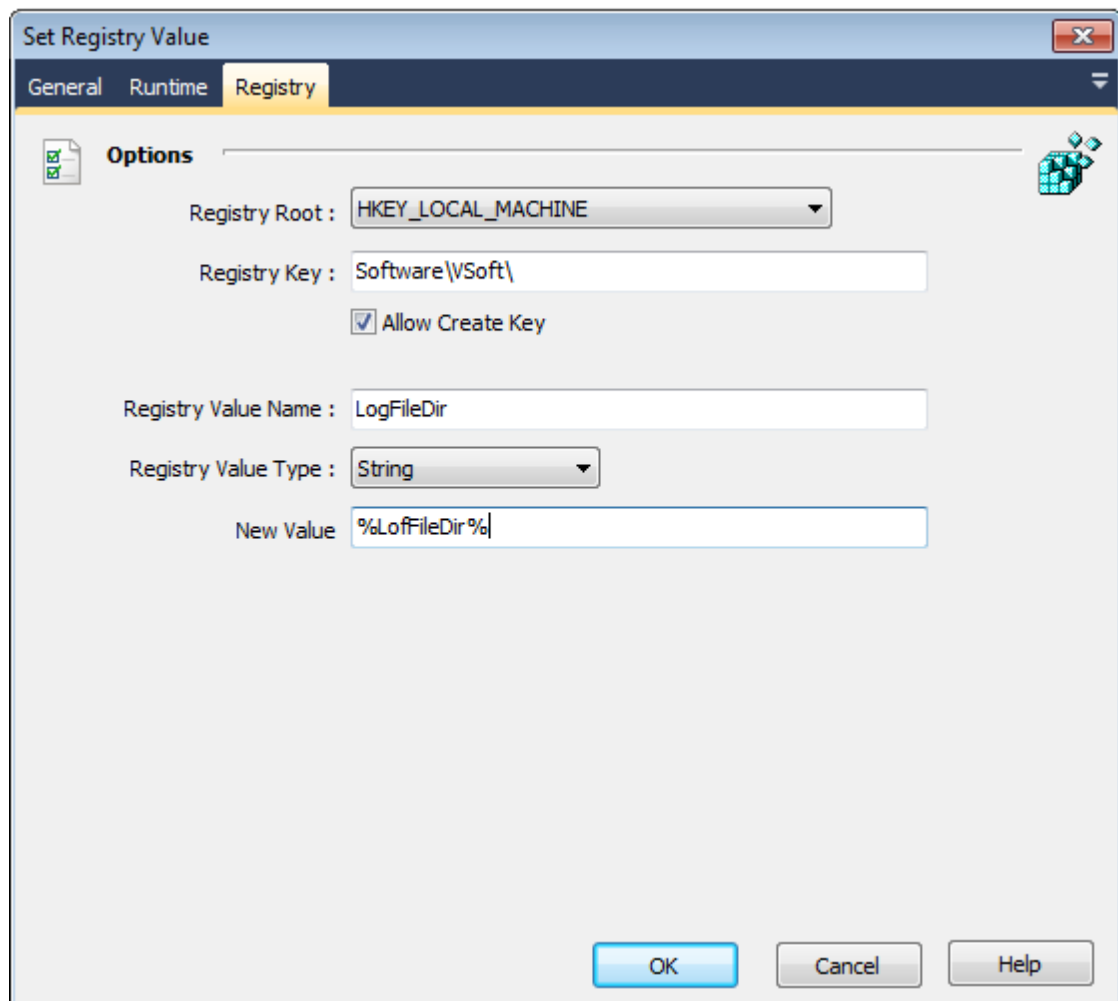
property DefaultValue : WideString

See also INI file format information

6.16.3 Read/Set/Delete Registry Value

These actions allow you to Read a Registry Value into a FinalBuilder variable, Set a Registry Value or Delete a Registry Key/Value.





The image shows a 'Set Registry Value' dialog box with three tabs: 'General', 'Runtime', and 'Registry'. The 'Registry' tab is selected. The dialog has a title bar with a close button. Below the tabs, there is a section titled 'Options' with a green checkmark icon. The 'Options' section contains the following fields and controls:

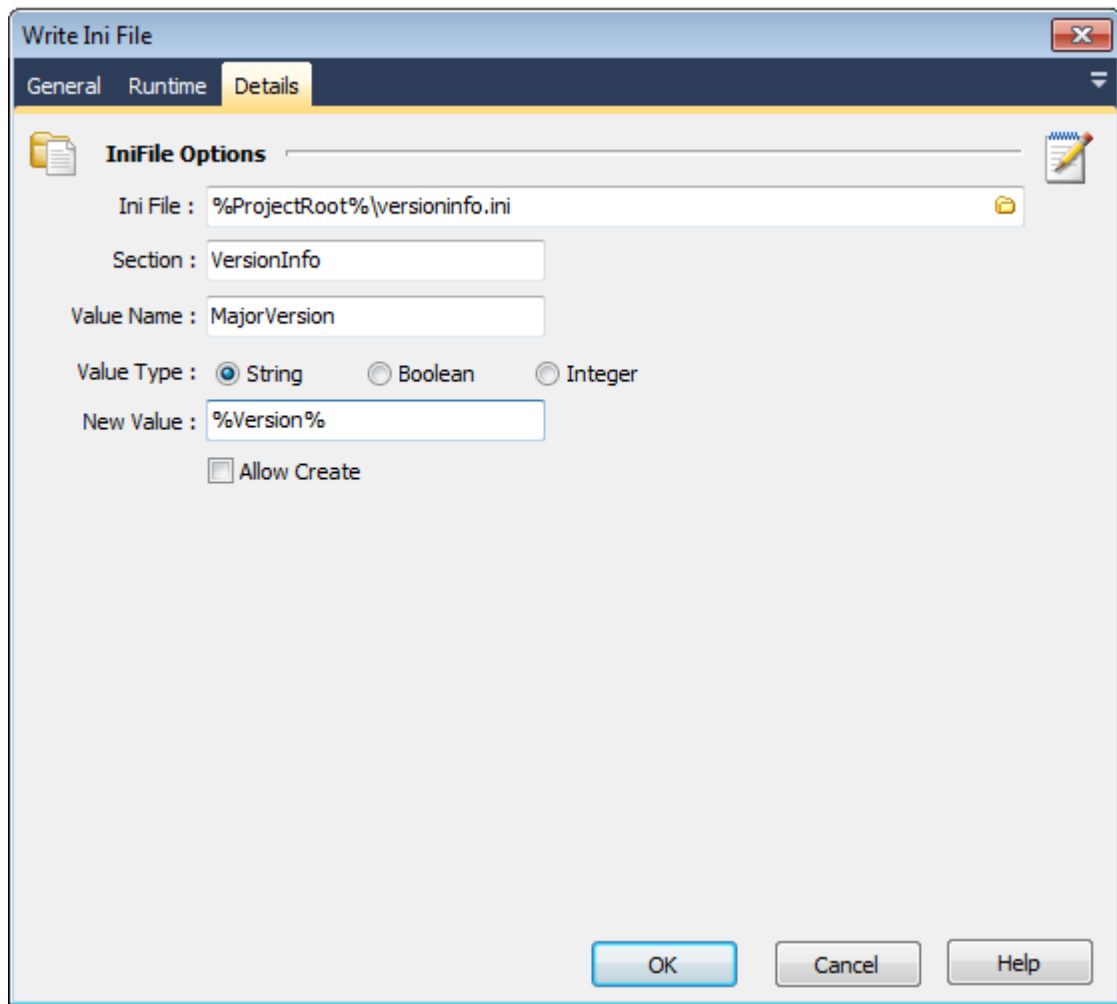
- Registry Root :** A dropdown menu showing 'HKEY_LOCAL_MACHINE'.
- Registry Key :** A text box containing 'Software\VSoft\'. Below it is a checked checkbox labeled 'Allow Create Key'.
- Registry Value Name :** A text box containing 'LogFileDir'.
- Registry Value Type :** A dropdown menu showing 'String'.
- New Value :** A text box containing '%LoffileDir%'.

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

For Delete Value/Key:
property DeleteKey;

6.16.4 Write Ini File

This action allows you to write values to an ini file.



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 FinalBuilder variables here.

Allow Create : Allow the ini file to be created if it does not already exist.

Scripting Info

The Action properties available are :

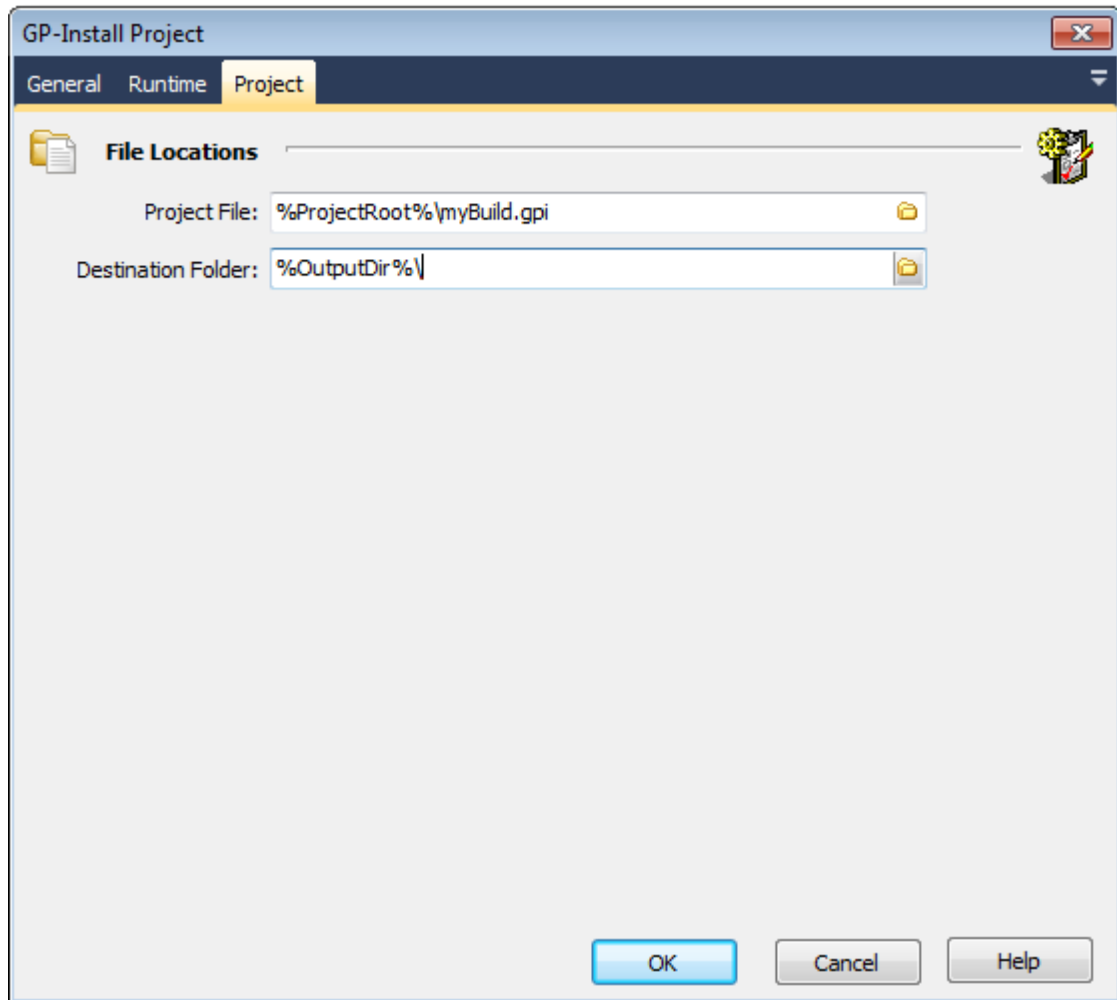
property IniFile : WideString
property Section : WideString
property ValueName : WideString
property NewValue : WideString
property AllowCreate : WordBool

See also INI file format information

6.17 Installers

6.17.1 GPInstall Action

This Action provides support for the popular Freeware install tool GPInstall.



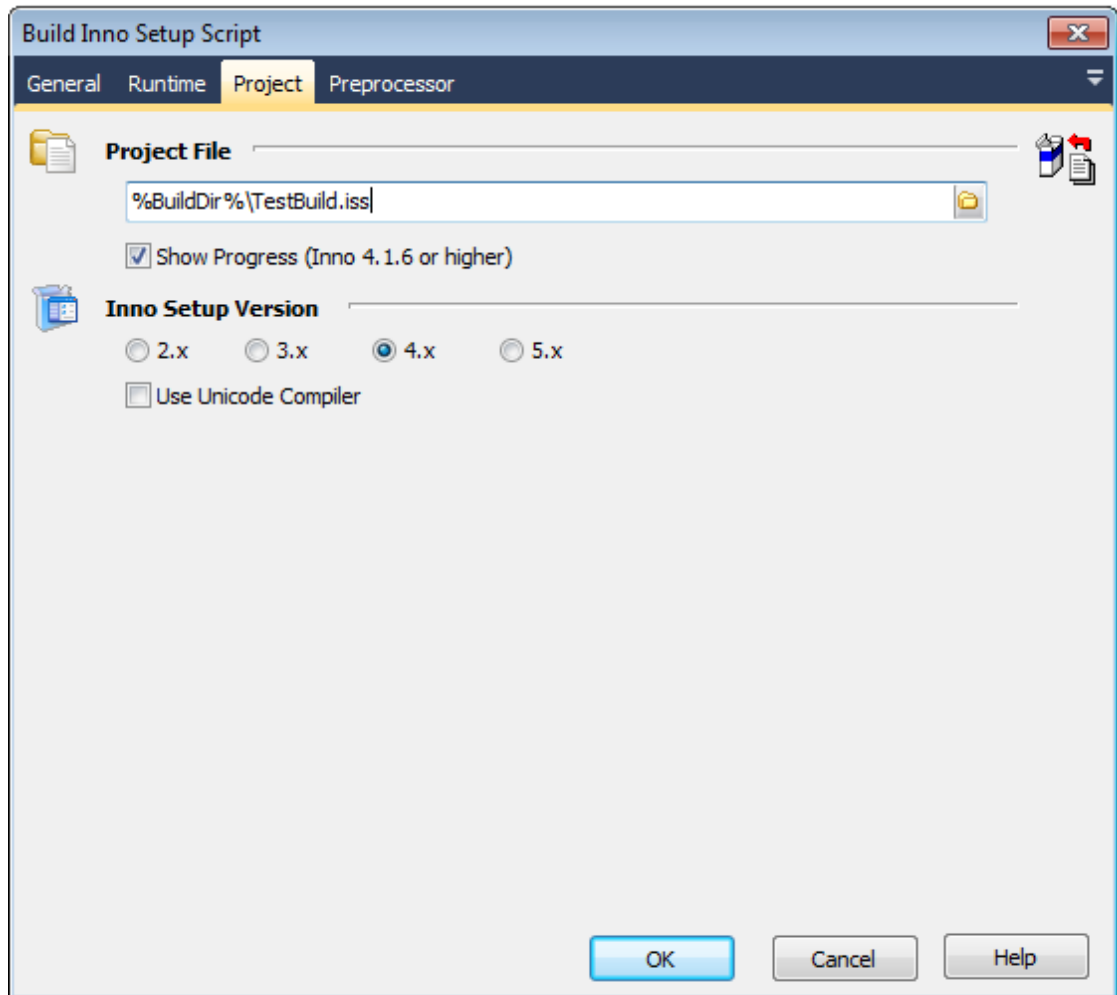
Scripting Info

The Action properties available are :

property DestinationFolder : WideString;
property ProjectFile : WideString;

6.17.2 Inno Setup

This action provides an interface to the Inno Setup Compiler. To use this action you need to specify the path to the compiler dll (ISComplr.dll) in the FinalBuilder Options dialog. FinalBuilder supports InnoSetup versions 2.x - 5.x



Project file : The fully qualified path to the setup script file (*.iss)

Show Progress : This option is only supported with InnoSetup version 4.1.6 or higher. It enables FinalBuilder to display the progress of InnoSetup as it creates the installer and is displayed on the Running screen.

Scripting Info

The Action properties available are :

property ScriptFile : WideString

property InnoVersion : integer // 0 = 2.x, 1 = 3.x, 2 = 4.x, 3 = 5.x

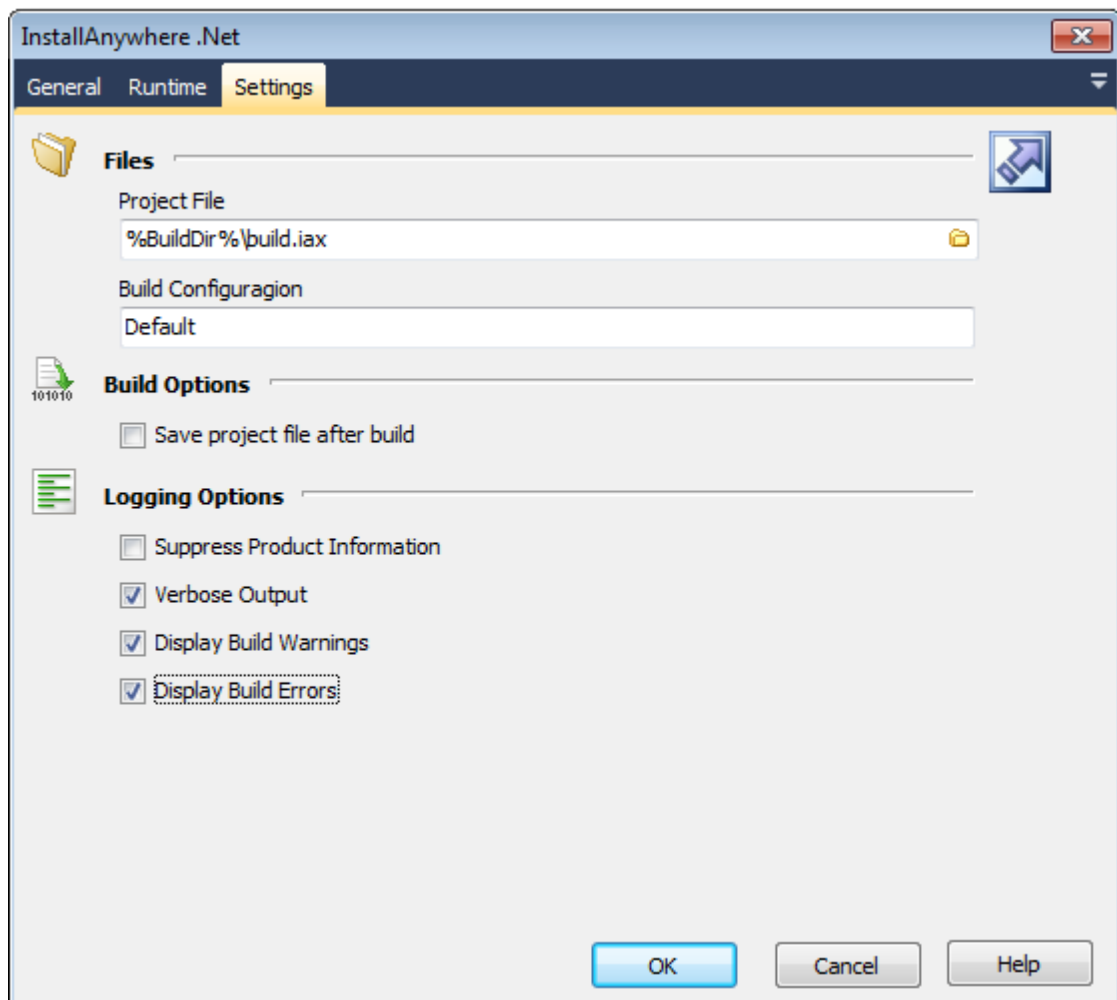
Inno Setup is a freeware setup compiler which can be downloaded from : <http://www.jrsoftware.org/isinfo.htm>

6.17.3 InstallAnywhere .Net

The InstallAnywhere.Net action enables you to automate the process of building ZeroG InstallAnywhere.Net installations.

Zero G's multiplatform solutions make deploying, delivering, and updating software applications easier and faster. Our award-winning solutions are full-featured, multiplatform, and perfect for enterprise development teams.

For more information, see <http://www.zerog.com>

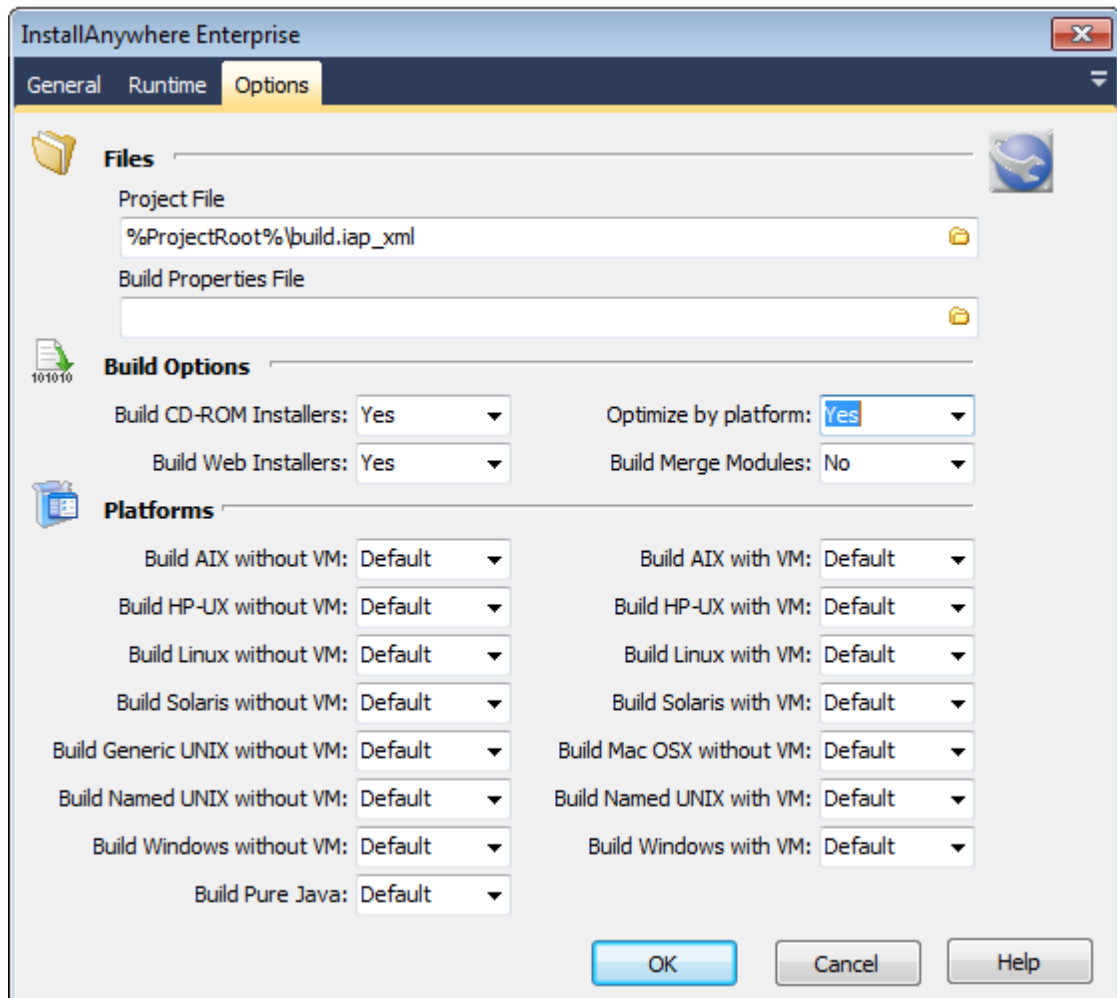


6.17.4 InstallAnywhere Enterprise

The InstallAnywhere Enterprise action enables you to automate the process of building ZeroG InstallAnywhere Enterprise installations.

Zero G's multiplatform solutions make deploying, delivering, and updating software applications easier and faster. Our award-winning solutions are full-featured, multiplatform, and perfect for enterprise development teams.

For more information, see <http://www.zerog.com>

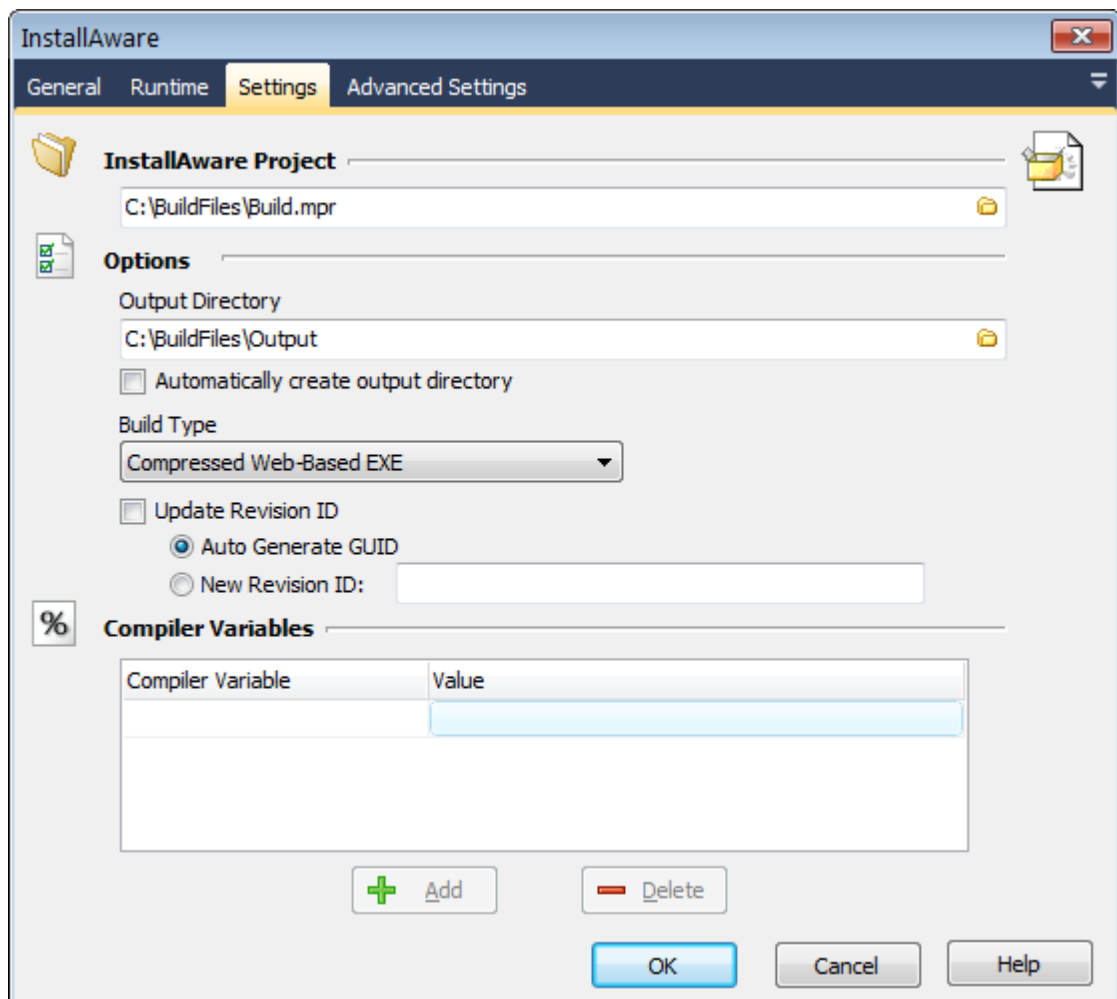


6.17.5 InstallAware

The InstallAware action enables you to automate the process of building InstallAware installations.

The InstallAware product family provides a complete set of installation authoring tools and environments targeting Microsoft Windows and Microsoft .NET platforms

For more information, see <http://www.installaware.com>



Update Revision ID

This will modify the InstallAware project file with a new Revision ID. The revision ID uniquely identifies the setup to Windows Installer. Each revision should have a new Revision ID otherwise Windows Installer will attempt to do a Repair/Remove instead of uninstalling an old build and then running the new build.

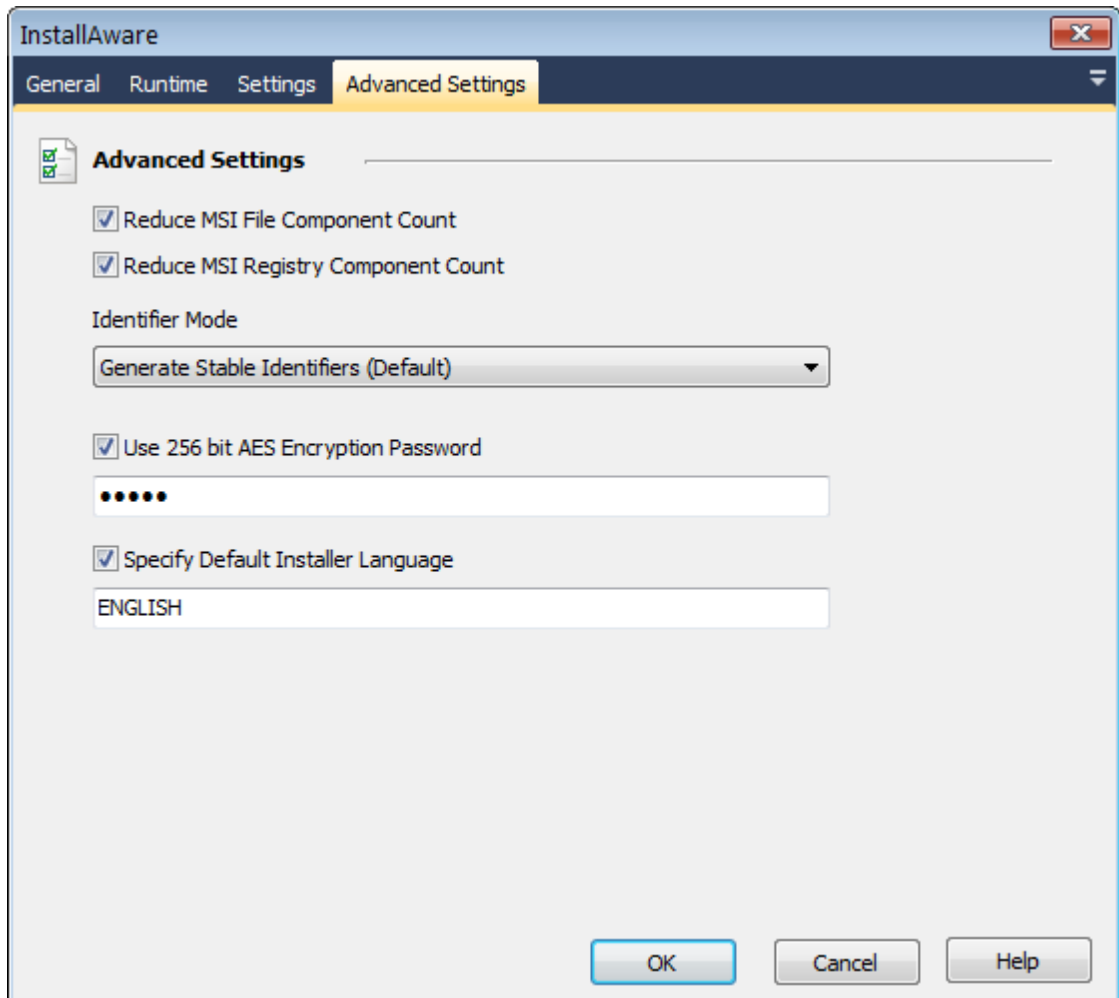
Auto Generate GUID

FinalBuilder will automatically generate a new GUID for the RevisionID each time the action runs.

New Revision ID

Specify the new Revision ID that you want this version to have.

The Advanced Settings page allows you to specify additional details about the build of your InstallAware package.



The following options are available from this page:

Reduce MSI File Component Count

Enabling this option allows InstallAware to group multiple files inside a single MSI component, this will improve installer performance.

Reduce MSI Registry Component Count

Enabling this option allows InstallAware to group multiple registry values into a single MSI component, this will improve installer performance.

Identifier Mode

The Identifier Mode option allows you to specify how InstallAware generates MSI code statement identifiers. There are three options available:

Generate Stable Identifiers

This is the default mode. A stable statement identifier is generated for each line of script which results in unique MSI Code statements. There is a performance cost to build the installer package when using this option, however runtime performance is not affected.

Use Dynamic Identifiers

Using this options will improve the build speed of your setup project. The InstallAware compiler will not attempt to generate unique statement identifiers, this may result in problems with patching and may break Web Media Block reuse.

Blend

This setting provides a combination of the previous two settings. A unique statement identifier for each statement that is associated with a source file, however all other statements, unstable identifiers will be used.

Use 256 bit AES Encryption Password

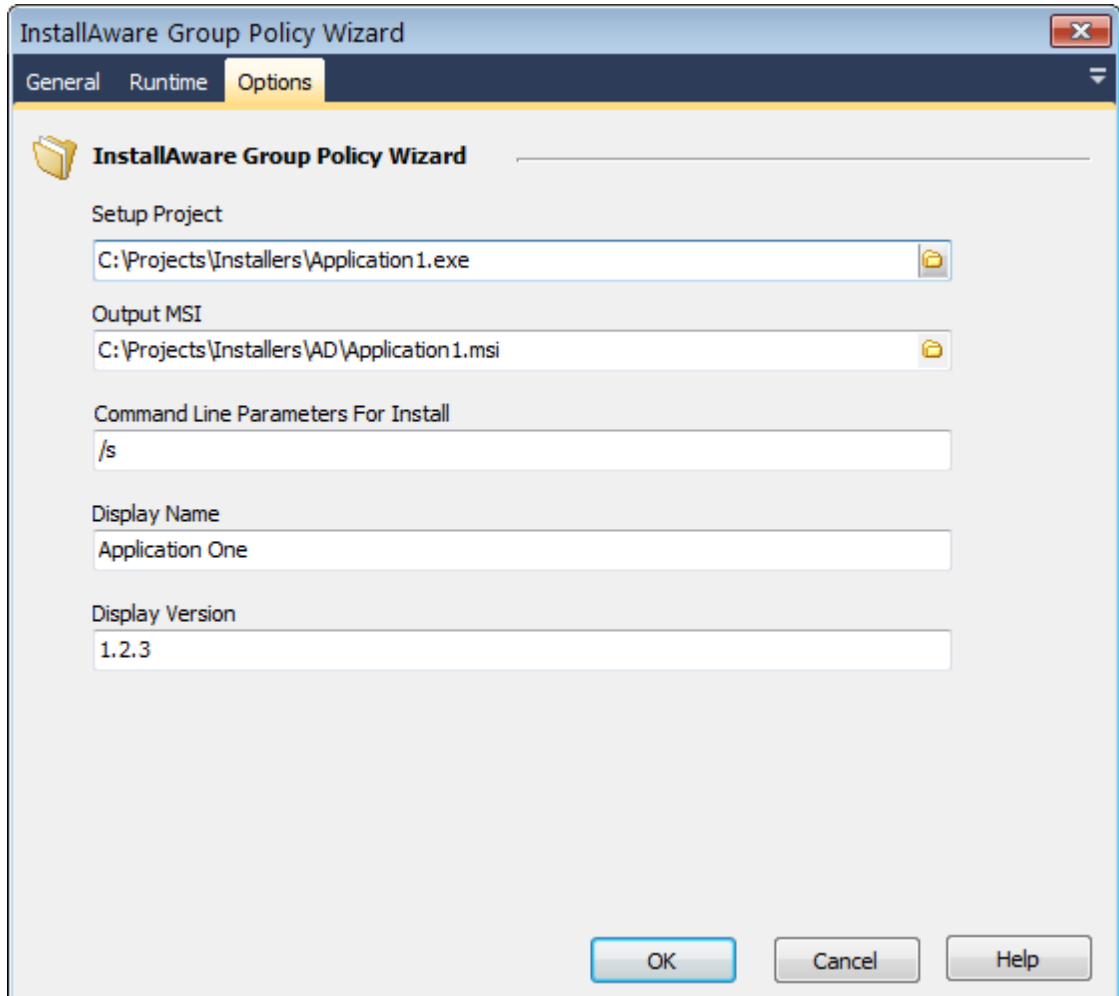
Use this option to encrypt your compressed setup packages with 256 bit AES encryption. This will encrypt both the setup files and the included data files.

Specify Default Installer Language

This option allows you to specify the default language for the installer package that is created.

6.17.6 InstallAware Group Policy Wizard

The InstallAware Group Policy Wizard action allows you to create a single MSI file that works with Group Policy and Active Directory deployment.



Setup Project

Provide the path to the InstallAware setup project that you wish to use for your Group Policy deployment. This setup project must be already built and can only be a Single File Compressed project, Compressed Web-Based EXE or a Patch.

Output MSI

Specify the the location and name for the Output MSI file to be created. The output folder for the new MSI must exist.

Command Line Parameters For Install (Optional)

Specify any command line parameters that you wish to pass to the installer package when the MSI is run.

Display Name (Optional)

Enter the name of the product to be displayed when the installation occurs.

Display Version (Optional)

Enter the version of the product to be displayed when the installation occurs.

For more information please see www.installaware.com

6.17.7 InstallShield

6.17.7.1 InstallShield 2008, 2009 & 2010

The InstallShield 2008, 2009, 2010 & 2011 action allows you to build InstallShield 2008, 2009, 2010 & 2011 projects.

Options / Stand Alone

The options on these property pages are the same as for the InstallShield Developer action. See the linked help topic for details.

Override Parameters

The options on this property page are unique to InstallShield 2008, 2009, 2010 & 2011.

InstallShield Dev 7 - 2011 Project (MSI)

General Runtime Options Stand Alone **Override Parameters**

Set Product Version

☐ Set the product version

☒ Set to values:

Major: Minor: Build: Revision:

☐ Set from PropertySet:

☒ Use Major, Minor, Build & Release/Revision Version Numbers

☐ Use Major, Minor, Release/Revision & Build Version Numbers

Override MSI Properties

Name	Value

Override Path Variables

Name	Value

OK Cancel Help

Set Product Version

This overrides the product version set in the .ISM project file with an explicit product version. This is equivalent to the -y command line option.

InstallShield 2010/2011 also allows you to specify the 'Revision' version number, this

option is not available for 2008/2009.

Set the product version via numeric values, or by specifying a PropertySet.

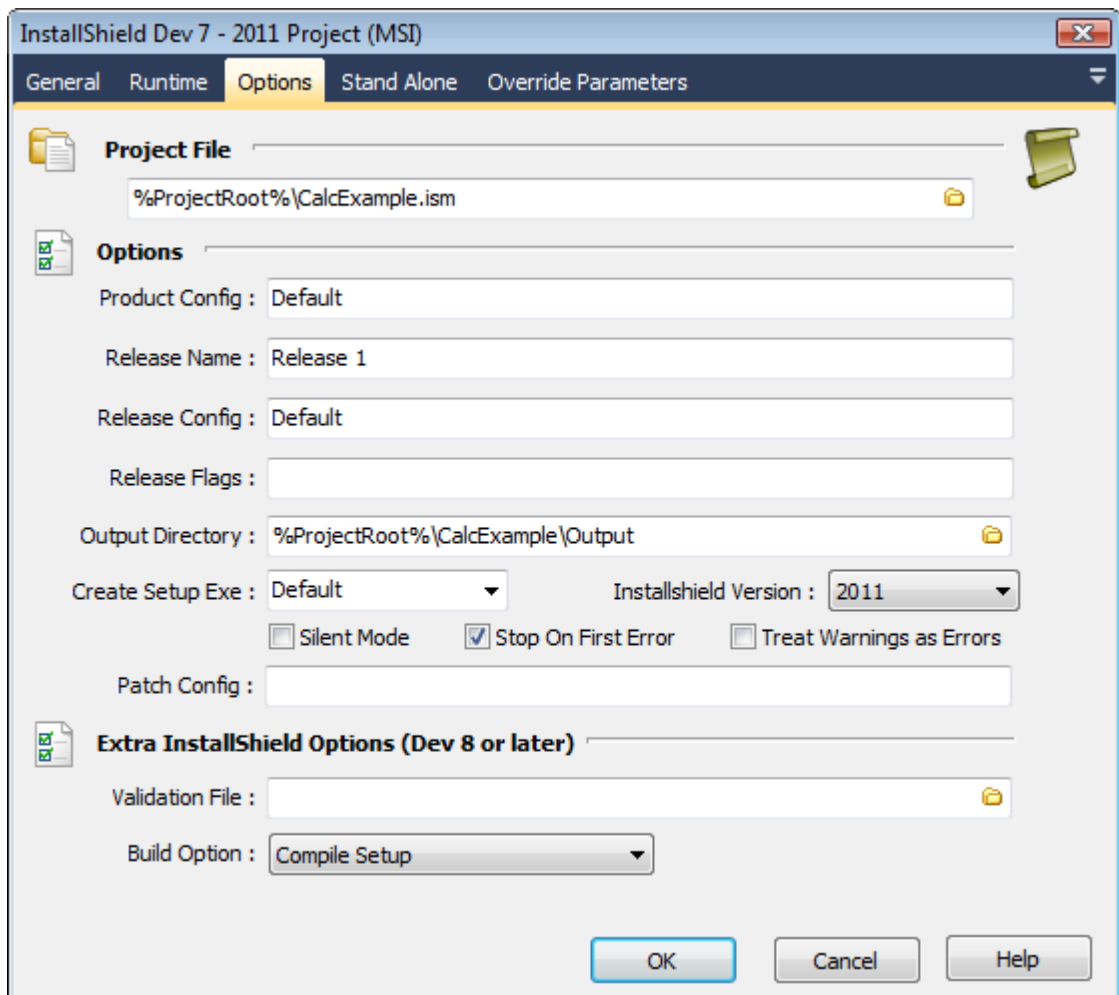
Override MSI Properties

This option allows you to override any MSI Property. If the property does not exist in the project, it will be created. This is equivalent to the -z command line option.

Type property names and values into the blank row of the grid. To remove an entry, clear the name and value fields.

6.17.7.2 InstallShield Developer

This Action provides an interface to InstallShield Developer 7.x - 12



Project File : Project File (.ism)

Product Config : Product Configuration for the release. If it does not exist it will be created. Although this parameter is not required, it is a good idea to include it if you are specifying a Release Name.

Release Name : The Release Name as specified in the Release Wizard (in the InstallShield IDE) You can use an existing release name or create a new one. Although this option is not required, it is a good idea to include it if you are specifying the flag for product configurations.

Release Config : This option allows to specify whether you would like to have your release compressed into one file or remain uncompressed in multiple files. This is optional, if the release name already exists, the configuration will be based on what is specified in the InstallShield IDE. If this is ignored for a new release, the new package will be uncompressed.

Release Flags : Use this option to specify any Release Flags that you would like to include in your build. Separate multiple flags with a comma.

Output Directory : Qualified path to where you want the output folders to be placed. This is optional, if not specified then the build will place the build package and files in the directory specified in the Project Location section of the Options panel in InstallShield.

Create Setup.Exe : Specify whether or not you would like to create a setup.exe along with your setup project.

InstallShield Version: Use this option to specify which version of InstallShield you are using.

Stop On First Error: Build will stop if an error is encountered.

Silent Mode : Run the InstallShield compiler in silent mode, this will disable the output of compiler messages.

Treat Warnings as Errors: Any warnings issued by the Installshield compiler will be treated as errors.

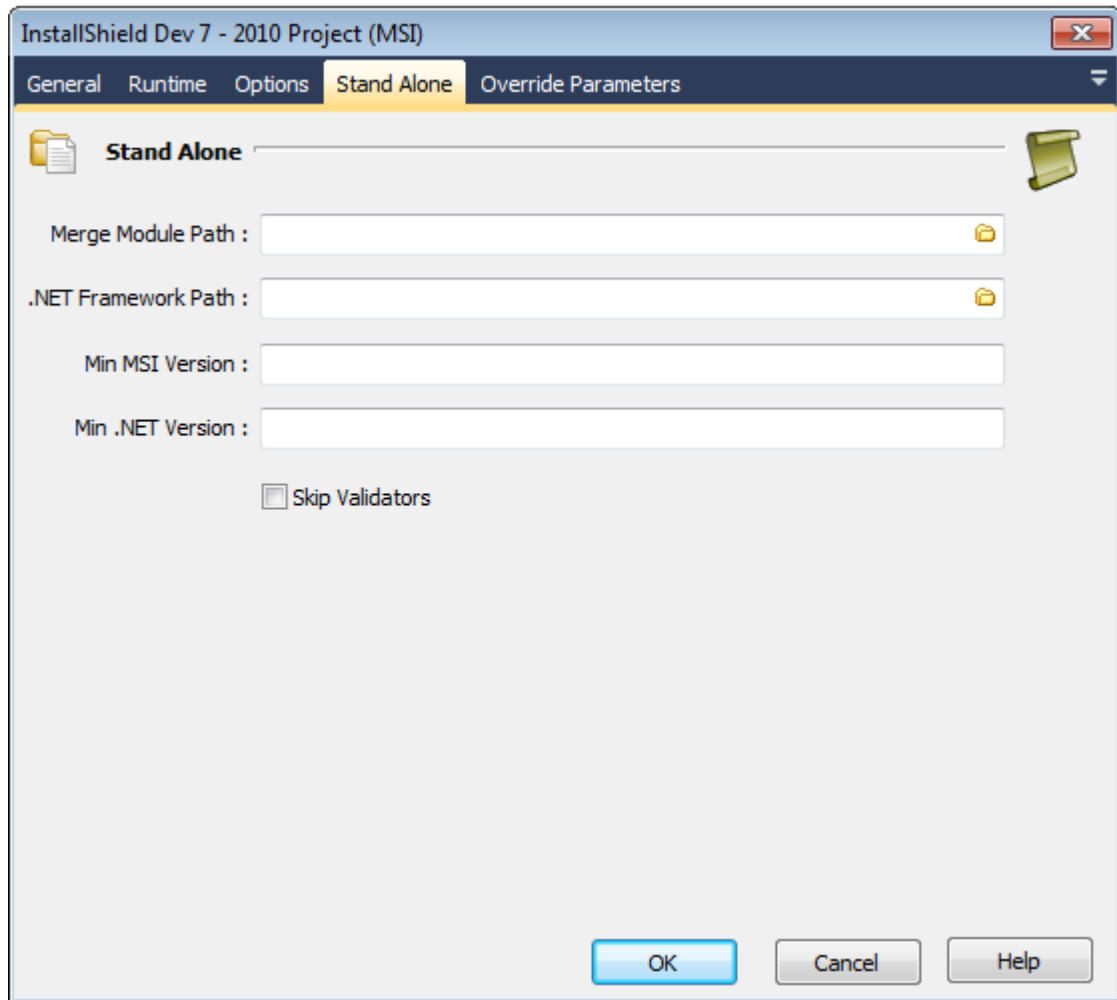
Stop on First Error : By default, the Installshield compiler does not stop when it encounters an error, it just keeps a count of errors to report at the end. Enable this option to stop the build as soon as the compiler encounters any error.

Patch Config: This option allows you to specify the name of the patch configuration that you wish to build.

Validation File: Specify a CUB file (.cub) to validate the installer package once it is built.

Build Option: This drop down box offers the following options:

- None
- Tables Only - Builds only the Windows Installer Tables for your release.
- Tables and Refresh - Builds Windows Installer Tables and refreshes files.
- Compile Setup - Compiles only Setup.rul



Merge Module Path : Enter a single path or multiple comma separated paths to the merge modules. You will need to manually quote this field if you are entering multiple paths (individually quoted) or your path contains spaces. The first path is what is selected as the path to copy the MSM file.

Note:

The copy will only work if IsSABld.exe is set as the IS executable. This is because it has the -o option w/in InstallShield

6.17.7.3 InstallShield Global Options

The InstallShield Global options are required for FinalBuilder to locate the InstallShield compilers. Before using the InstallShield actions you will need to set the path to the InstallShield compilers. The options are located in Tools Menu -> Options, Install Builders category, InstallShield.

The InstallShield Global options can also be set via script (for example if you need to dynamically change the location of the compiler to support more than one version of InstallShield Developer).

Example (VBScript):

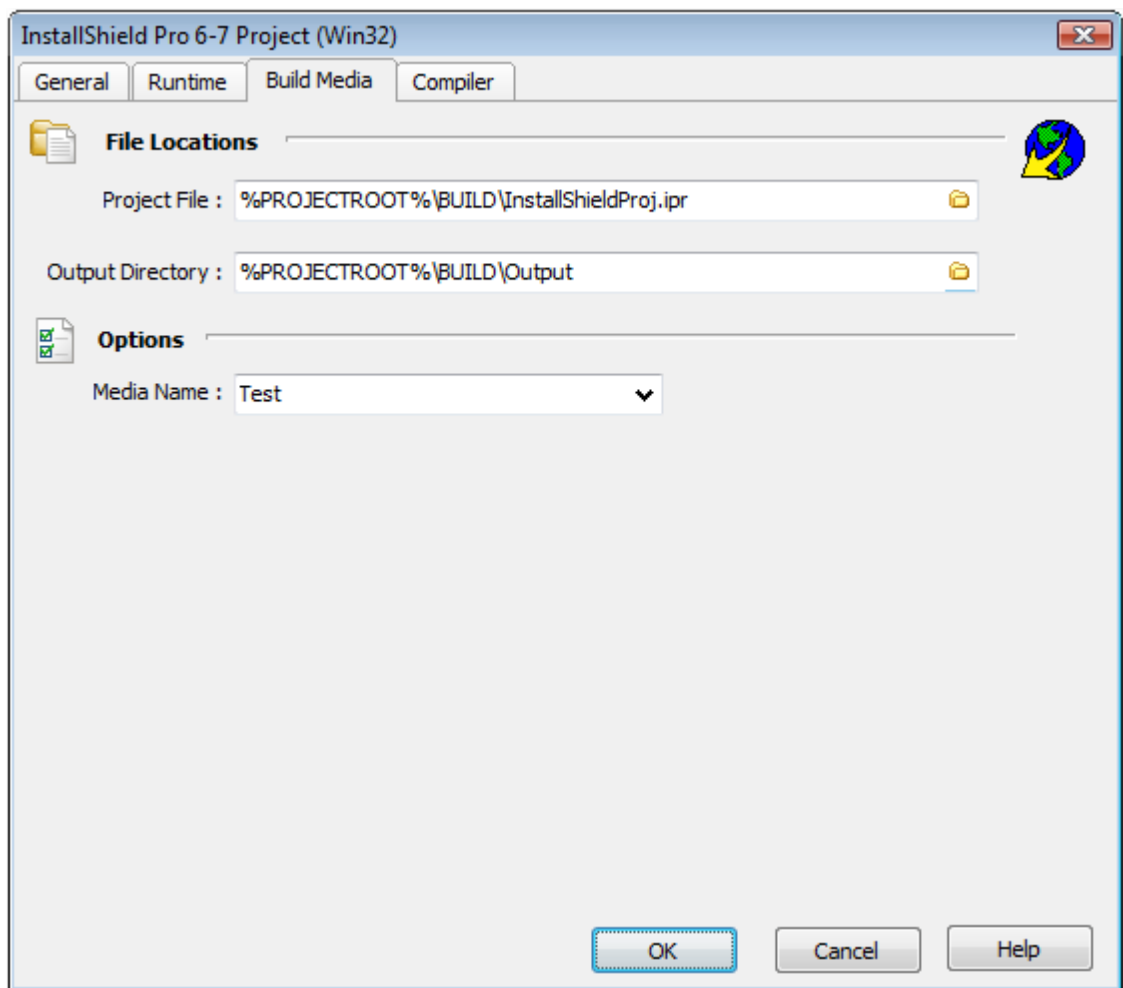
```
dim opt
opt = GetOptionsObject("InstallShield")
opt.DevISCmdBldLocation = "path to ISCmdBld"
```

The available properties of the InstallShield object are:

- CompilerLocation // required for InstallShield Professional 6.x to 7.x
- IFXLocation // required for InstallShield Professional 6.x to 7.x
- ISRTLLocation // required for InstallShield Professional 6.x to 7.x
- IncludePath // required for InstallShield Professional 6.x to 7.x
- ISBuildLocation // required for InstallShield Professional 6.x to 7.x
- ISCmdBldLocation // for Windows Installer Edition
- DevISCmdBldLocation // for InstallShield Developer
- UniversalLocation // for InstallShield Universal

6.17.7.4 InstallShield Pro - Std Edition

This Action provides an interface to InstallShield Pro - Standard Edition version 6.2.x - 7.x



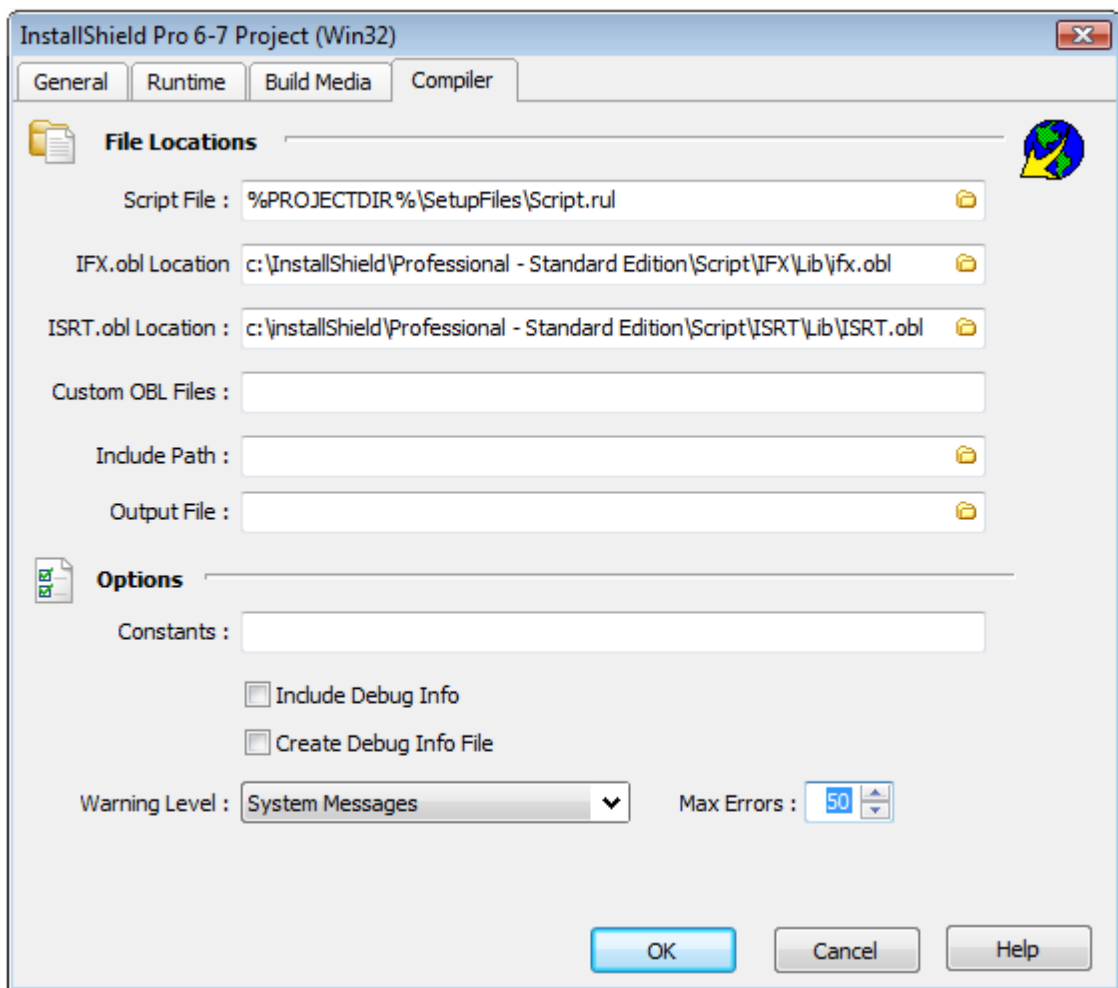
Project File : The fully qualified path to the setup project file (*.ipr).

You can also compile Installshield Objects (.ipo) but you'll have to specify the .ipo file

without using the file dialog. You will also have to manually enter the Media name, as the automatically detected media names only work for .ipr files.

Output Directory : The fully qualified path to where you want the output folders and files to be placed. The built setup files will be placed in the "Disk Images\Disk1" subfolder in the location you specify.

Media Name : The media name that you entered into the Media Wizard - Media Name Panel in InstallShield.



Script File : Specifies the name of the Setup Script.

IFX.obl Location : The location of the ifx.obl file, defaults to the value specified in the Options dialog.

ISRT.obl Location : The location of the isrt.obl file, defaults to the value specified in the Options dialog.

Custom OBL Files : The fully qualified path (separated by semi-colons) of any custom library files your setup requires

Include Path : Specifies the search path for source files that have been included in the setup by means of #include statements.

Constants : Specify any constants you require in the form identifier=value , separate multiple constants with semicolons

Output File : Specifies the file name to assigned to the compiled script.

Include Debug Info : Specifies that debugging information should be included in the compiled script file, so a debugging information file is not needed.

Create Debug Info File : Specifies that a debugging information file should be produced.

Warning Level : Sets error reporting level for the compiler.

Max Errors : The maximum number of errors before the compiler aborts.

Scripting Info

The Action properties available are :

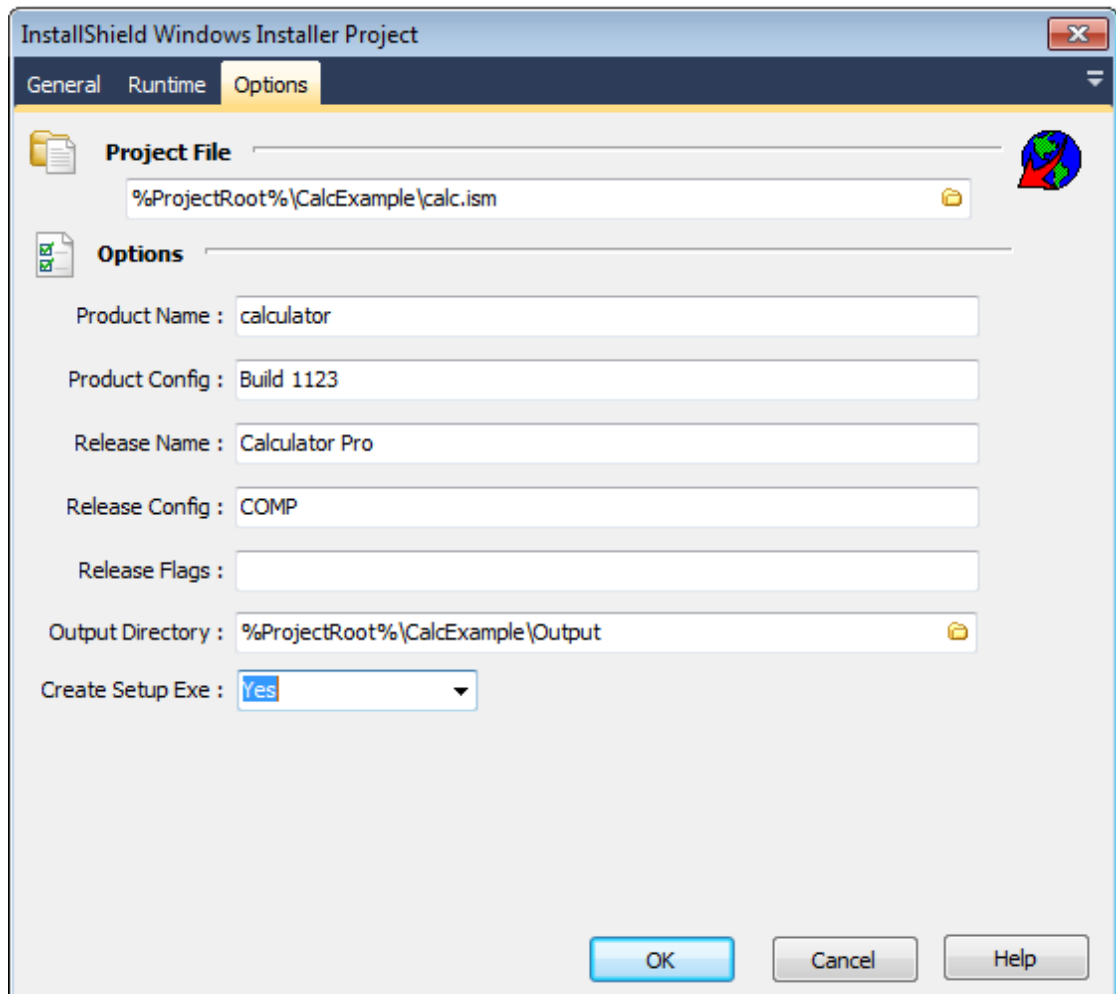
property BuildMedia : WordBool // Build the Media
property CompileSetup : WordBool //CompileSetup // compile the setup program

property ProjectFile : WideString
property OutputDirectory : WideString
property MediaName : WideString;

property SetupScriptFile : WideString
property IFXLocation : WideString
property ISRTLLocation : WideString
property CustomOBLFiles : WideStringSetCustomOBLFiles;
property Constants : WideString
property MaxErrors : integer
property CreateDebugInfoFile : WordBool
property IncludeDebugInfo : WordBool
property IncludePath : WideString
property WarningLevel : integer
property OutputFileName : WideString

6.17.7.5 InstallShield Pro - Windows Installer Edition

This Action provides an interface to InstallShield Pro - Windows Installer Edition



Project File : Project File (.ism)

Product Name : The Product Name in the .ism file

Product Config : Product Configuration for the release. If it does not exist it will be created. Although this parameter is not required, it is a good idea to include it if you are specifying a Release Name.

Release Name : The Release Name as specified in the Release Wizard (in the InstallShield IDE) You can use an existing release name or create a new one. Although this option is not required, it is a good idea to include it if you are specifying the flag for product configurations.

Release Config : This option allows to specify whether you would like to have your release compressed into one file or remain uncompressed in multiple files. This is optional, if the release name already exists, the configuration will be based on what is specified in the InstallShield IDE. If this is ignored for a new release, the new package will be uncompressed.

Release Flags : Use this option to specify any Release Flags that you would like to include in your build. Separate multiple flags with a comma.

Output Directory : Qualified path to where you want the output folders to be placed. This is optional, if not specified then the build will place the build package and files in the directory specified in the Project Location section of the Options panel in InstallShield.

Create Setup.Exe : Specify whether or not you would like to create a setup.exe along with your setup project.

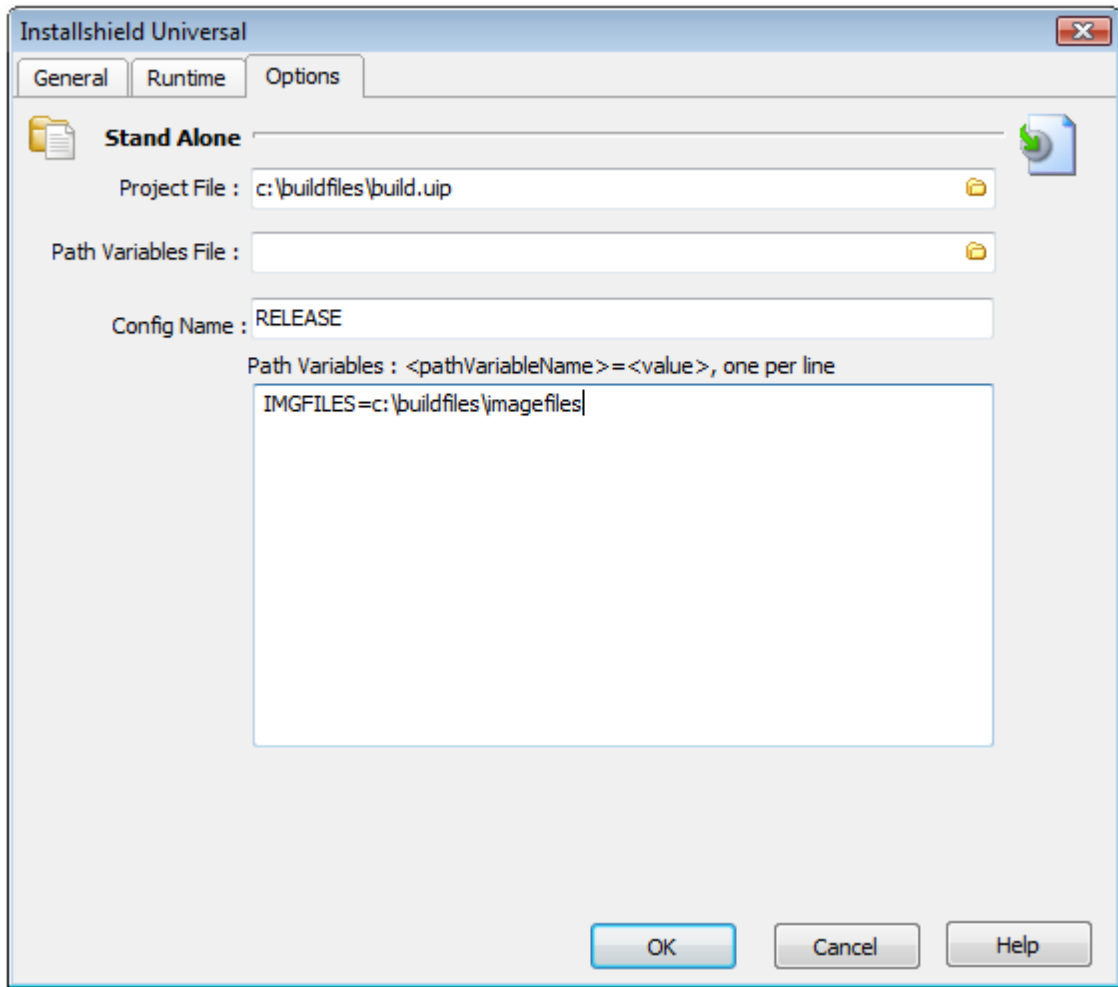
Scripting Info

The Action properties available are :

property ProjectFile : WideString
property ProductName : WideString
property ReleaseName : WideString
property ProductConfig : WideString
property ReleaseConfig : WideString
property OutputDirectory : WideString
property ReleaseFlags : WideString
property CreateSetupExe : WordBool

6.17.7.6 InstallShield Universal Installer

The InstallShield Universal action enables you to automate the process of building InstallShield Universal installations.



Project File - specify your InstallShield Universal project file

Path Variables File - specify the fully qualified path to your variables file

Config Name - enter the configuration you want to run. The configurations are defined in your project file.

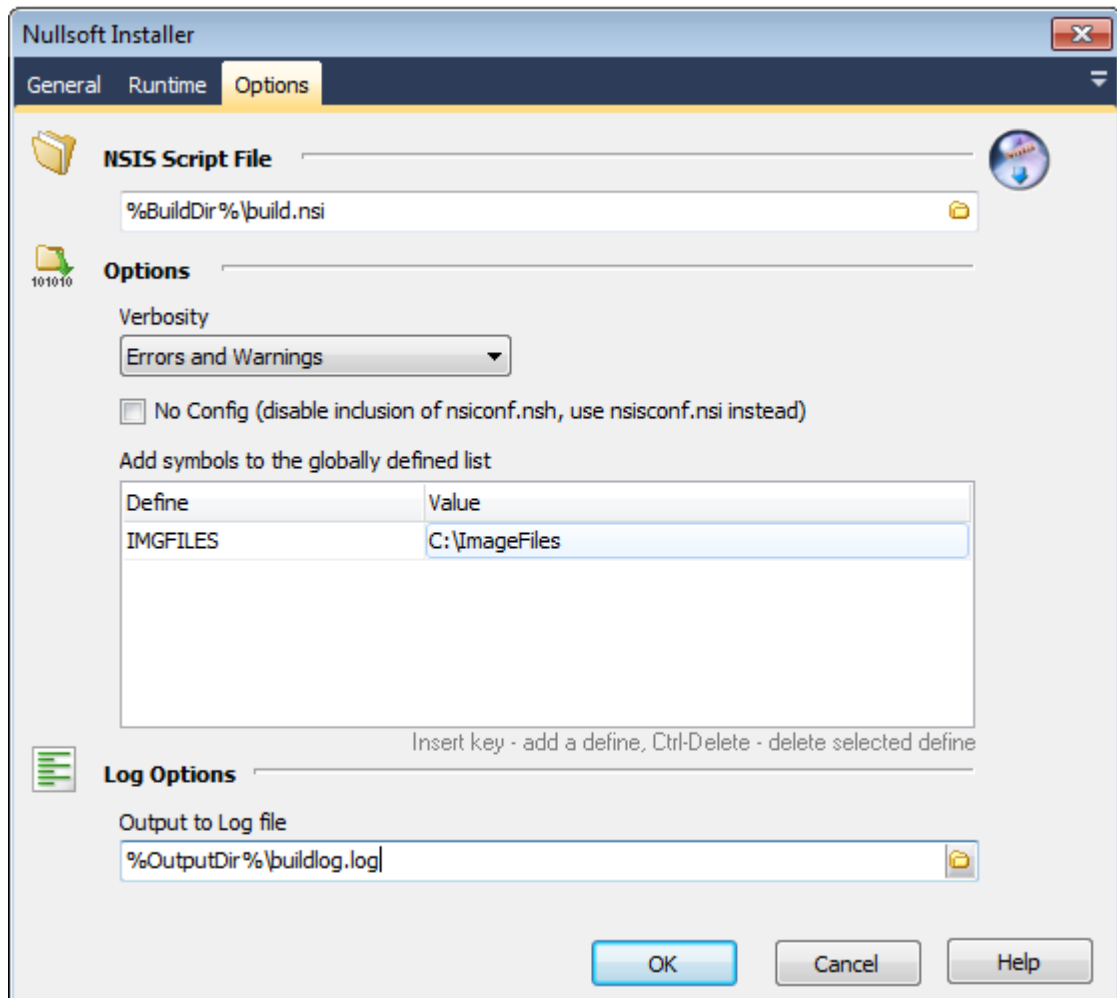
Path Variables - enter name/value pairs defining the values for directory variables

6.17.8 Nullsoft NSIS

The Nullsoft Installer action enables you to automate building Nullsoft Installer projects.

NSIS (Nullsoft Scriptable Install System) is a tool that allows programmers to create installers for Windows. It is released under an open source license and is completely free for any use.

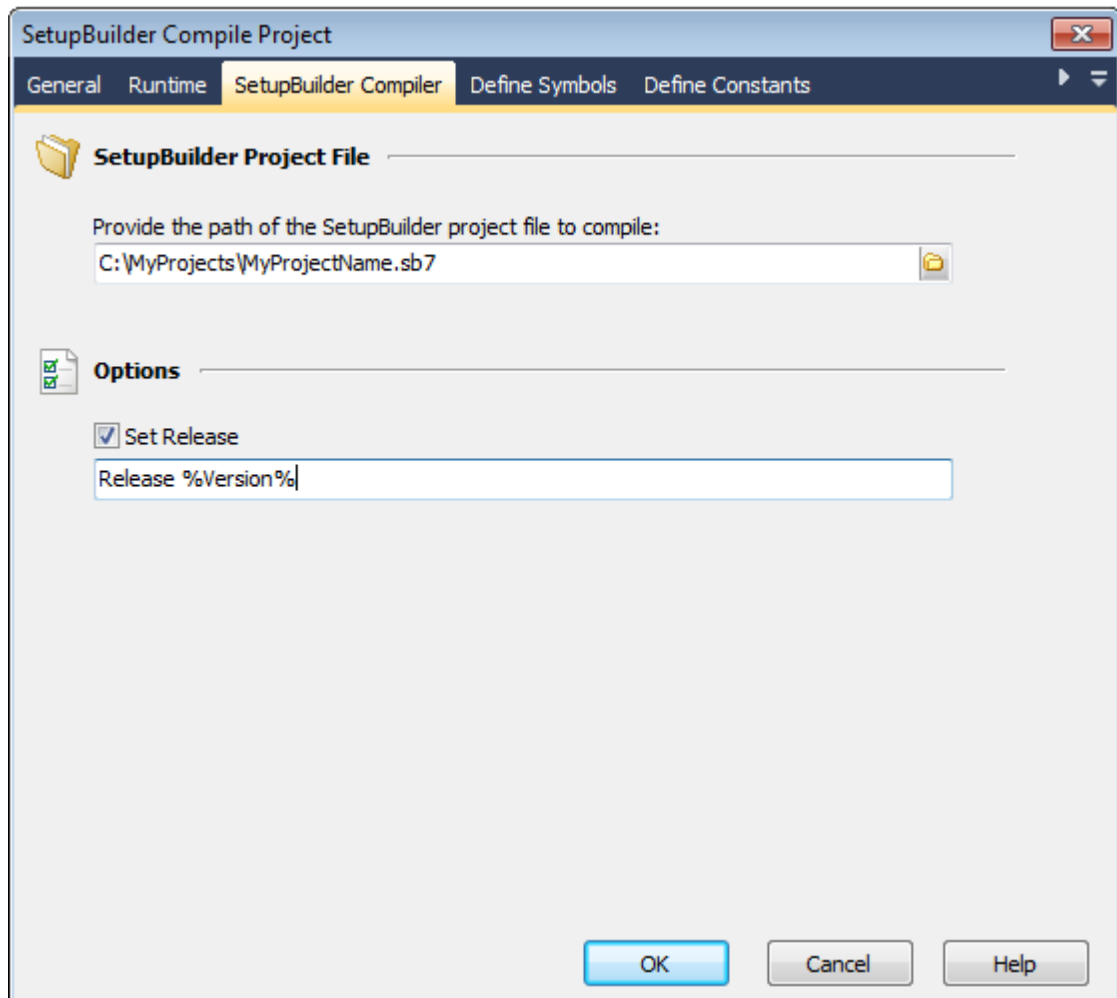
For more information see <http://nsis.sourceforge.net/>



6.17.9 SetupBuilder Compile Project

The SetupBuilder Compile Project Action allows you to compile a SetupBuilder project file in order to create an Installer executable.

From the *SetupBuilder Compiler* tab, specify the location of the project which you want to compile.



Also from this tab the *Set Release* option allows you to specify the Release which you would like to compile.

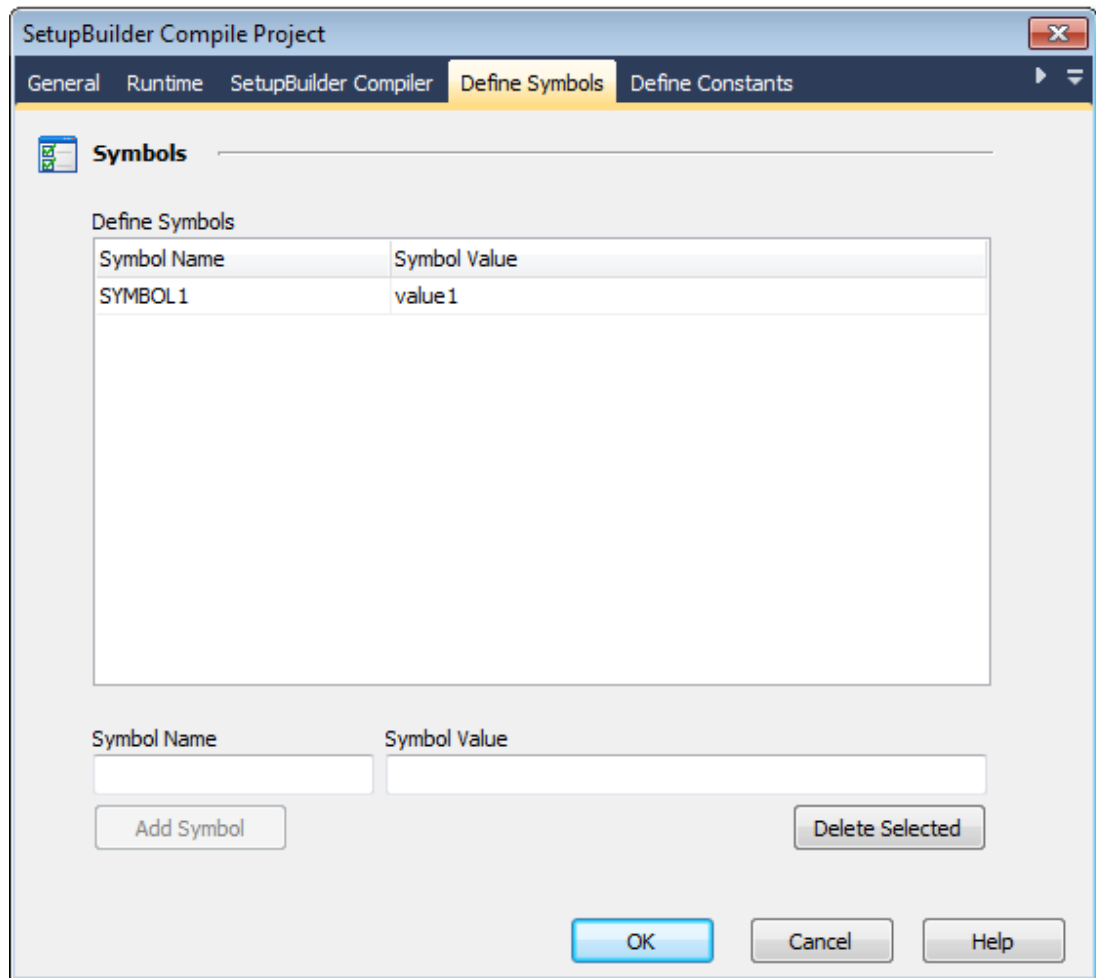
Note: An invalid release value will cause the compilation/action to fail.

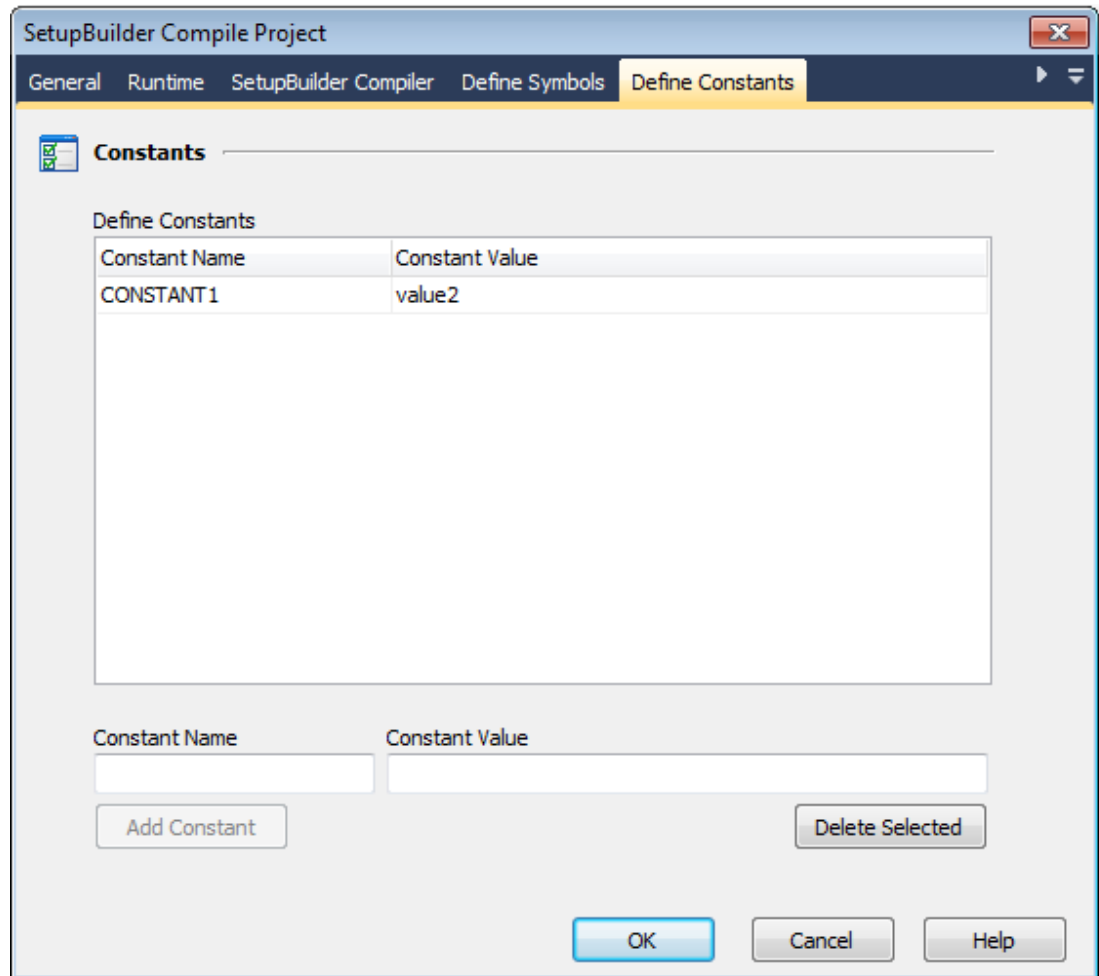
Depending on the requirements of your project it may be necessary to define symbols, constants and compiler variables when compiling your project.

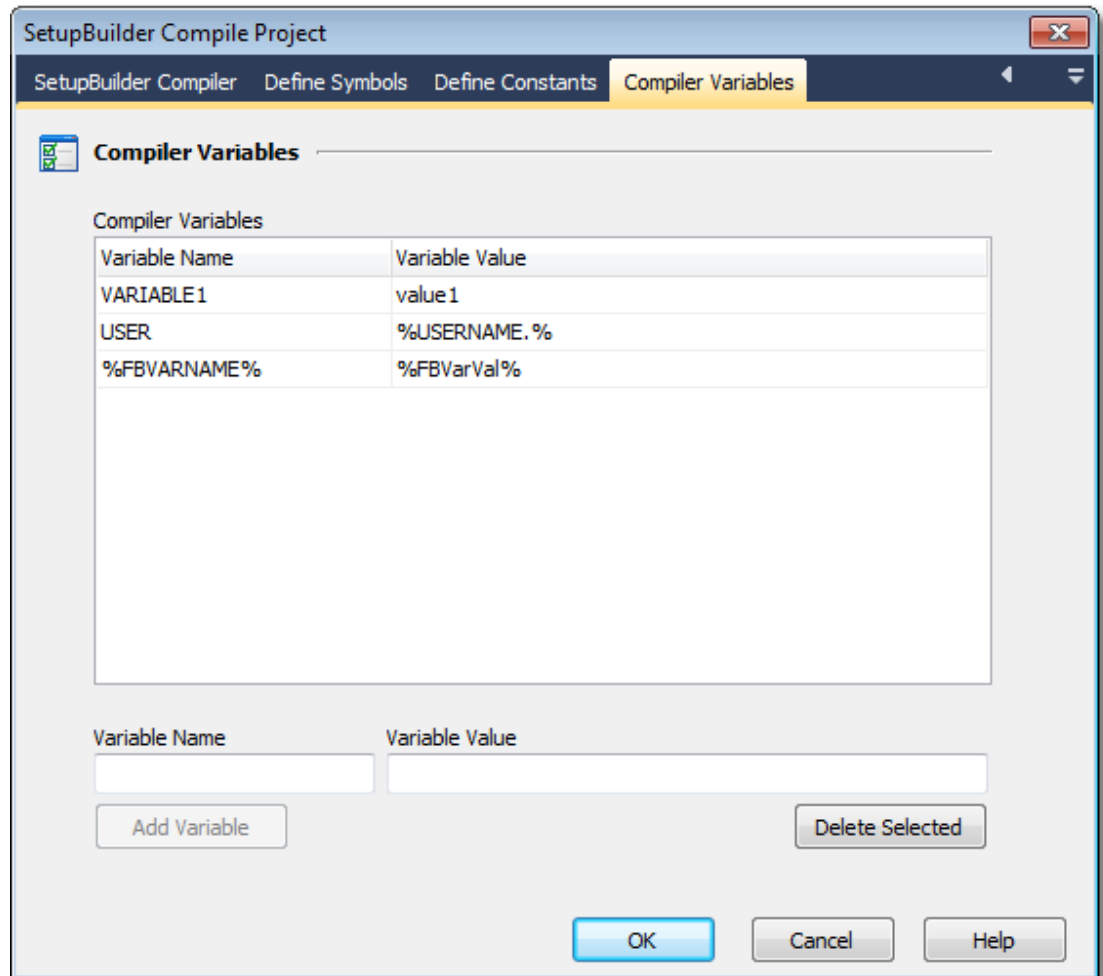
There are three separate tabs that allow you to do all of this when running this action:

- Define Symbols
- Define Constants

- Compiler Variables







These three tabs are much the same, to define a compiler variable enter the name in the text box *Variable Name* and an associated value in the *Variable Value* and click the *Add Variable* button. The *Add Variable* button will remain disabled until both text boxes are populated.

Once added to the list you can modify the value of the variable by selecting the value in the grid view and typing a new value. The variable name cannot be modified once added, to change a variable name you must delete that entry and add the name/value pair again.

Note: An equals sign (=) will not be accepted within a symbol, constant or variable name. If you want to define a name that contains an equals sign you can create a FinalBuilder project variable (which contains the desired name) and pass this variable as the symbol/constant/compiler variable name.

All symbol, constant and variable names are added in uppercase.

Note: Any message boxes in the SetupBuilder compiler script will need to either be commented out or set to be a 'timed message box' (which will only be displayed for two seconds) otherwise an unattended build will 'hang' waiting for user to acknowledge the message box.

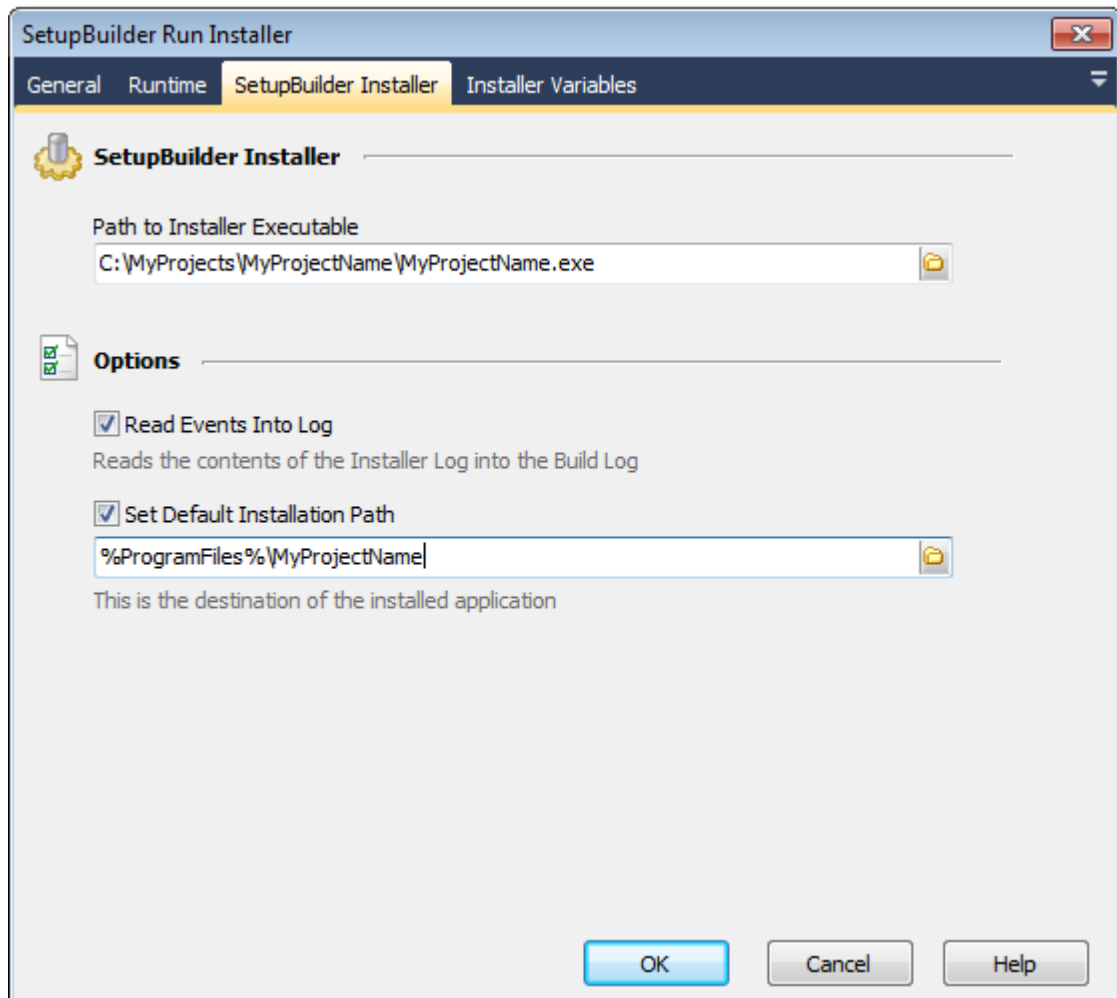
This action requires SetupBuilder Developer Edition.

For more information about SetupBuilder see the Lindersoft website.

6.17.10 SetupBuilder Run Installer

The SetupBuilder Run Installer Action allows you to run an installer executable that has been compiled by SetupBuilder.

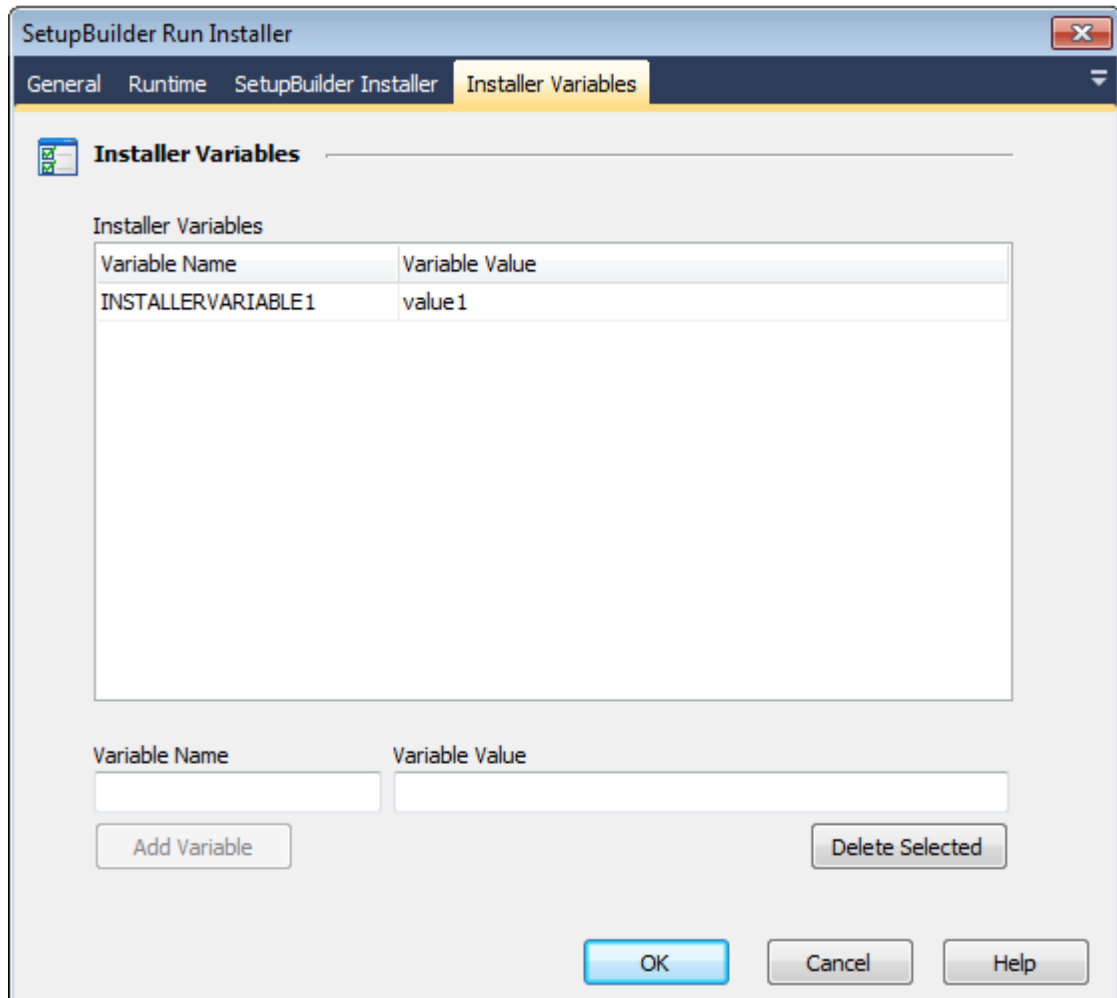
On the *SetupBuilder Installer* tab specify the path to the Installer Executable. If the executable is not an installer package generated by SetupBuilder then the action will fail.



There are also a number of options available from this tab:

- Read Events Into Log - At the completion of execution read the contents of the Installer Log into the FinalBuilder Build Log.
- Set Default Installation Path - The SetupBuilder will contain a default install path (which is the destination for the installation). This option lets you override this value so you can specify a custom location for the application to be installed.

The SetupBuilder Run Installer Action also allows you to define Installer Variables via the *Installer Variables* tab.



Enter a name and a value for the Installer Variable in the text boxes provided (once both text boxes are populated the *Add Variable* button will be enabled) then click the *Add Variable* button to add the variable to the list.

Once added to the list you can modify the value of the variable by selecting the value in the grid view and typing a new value. The variable name cannot be modified once added, to change a variable name you must delete that entry and add the name/value pair again.

Note: An equals sign (=) will not be accepted within a variable name. If you want to define a name that contains an equals sign you can create a FinalBuilder project variable (which contains the desired name) and pass this variable as the variable name.

All variable names are added in uppercase.

Note: This action runs in silent mode, which means the installer dialogs will be suppressed and default values will be used. Any values that you want to populate can be passed via an Installer Variable or by using an INI file to define values.

This action requires SetupBuilder Developer Edition.

For more information about SetupBuilder see the Lindersoft website.

6.17.11 Setup Factory

The Setup Factory action allows you to automate building of Setup Factory installers as part of the build process.

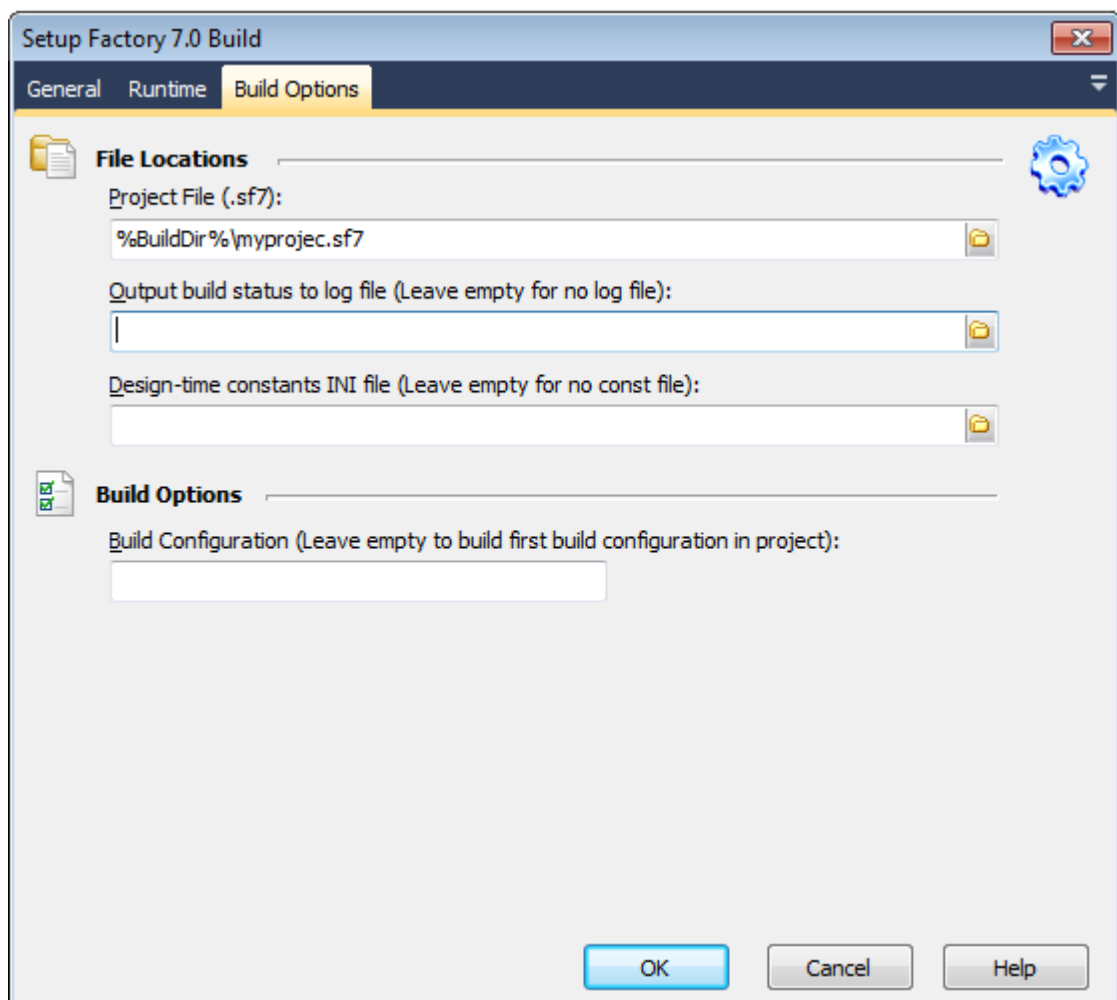
Before running this action, you need to set up the Setup Factory Options. This is accessed via the Tools menu, Options.

Setup Factory 7.0 Location

The fully qualified path and filename of the Setup Factory 7.0 design environment executable. The file is always called "SUF70Design.exe" and is usually located in "C:\Program Files\Setup Factory 7.0".

Setup Factory 7.0 Build Action

Setup Factory 7.0 Action



Project File (.sf7)

The fully qualified path and filename of the Setup Factory 7.0 project file that you want to build. This file must have been previously created and saved using the Setup Factory 7.0 design environment. By default, Setup Factory project files are saved in "My Documents\Setup Factory 7.0 Projects".

Output build status to log file (Leave empty for no log file)

Specifies a log file to output build status to. If the file exists, the output will be appended to it. If the file does not exist, it will be created. For example, "C:\Output\Setup Build Log.txt". Leave this field empty if you do not want to create a log file.

Design-time constants INI file (Leave empty for no const file)

Lets you specify an INI file that contains design-time constants to override the ones in the project. You can define as many design-time constants as you want in the INI file, with each constant on a separate line beneath the [Constants] section. Each constant that is defined in the INI file must already be defined in the project file. For example:

```
[Constants]
#OUTPUTDIR#=C:\Output\Foobar 2002\Release
#SETUPNAME#=foobar2002setup.exe
#BUILD#=release
```

Leave this field empty if you do not want to use a constants INI file.

Build Configuration (Leave empty to build the first build configuration in project)

Specifies the build configuration to use in the build. If left blank, the first build configuration found in the project is used. If your project only has one build configuration, you can leave this field empty.

For more information on Setup Factory, visit <http://www.setupfactory.com>

6.17.12 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.

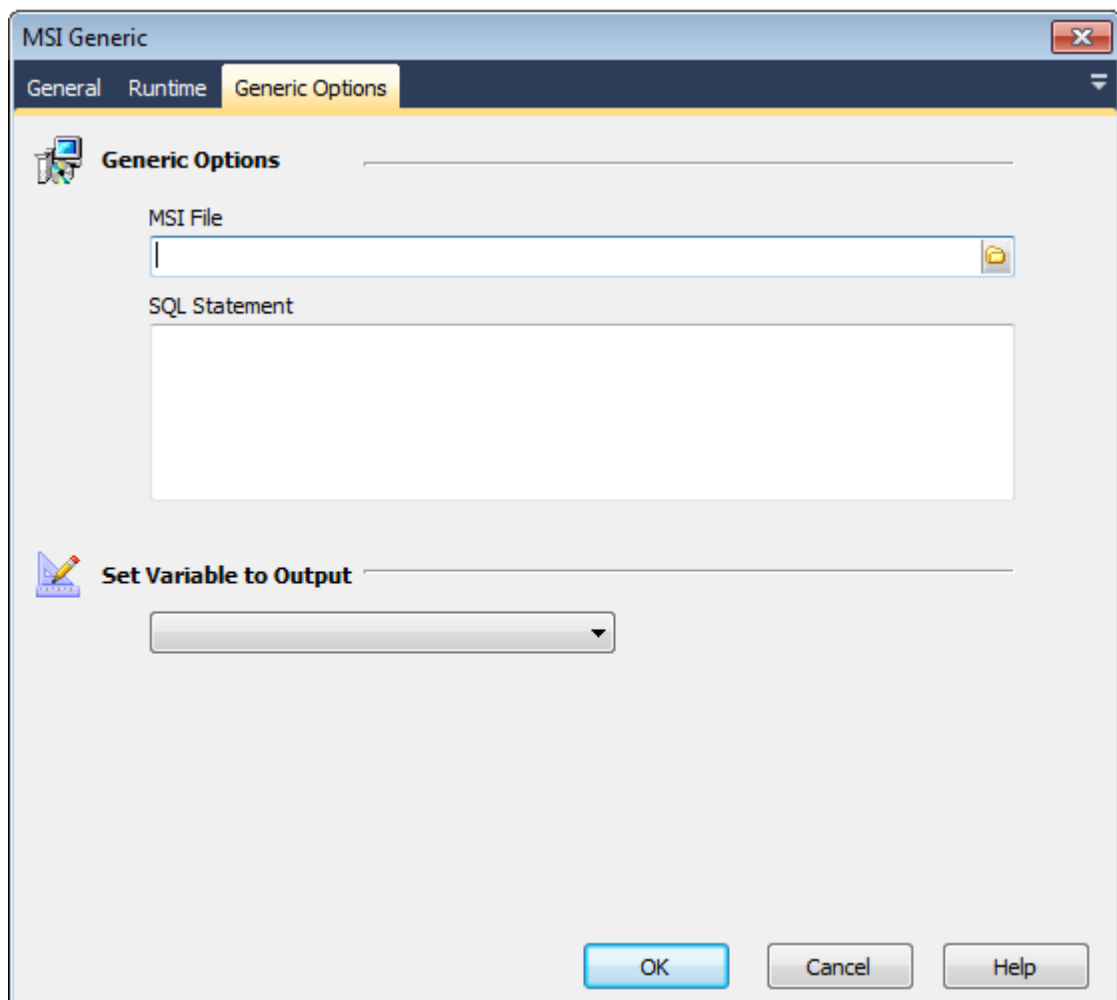
Global Options

The location of MsiVal2 can be specified in the global options dialog: Tools - Options - Install Builders - MSI Sdk.

6.17.12.1 MSI Generic

[FinalBuilder 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

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'
```

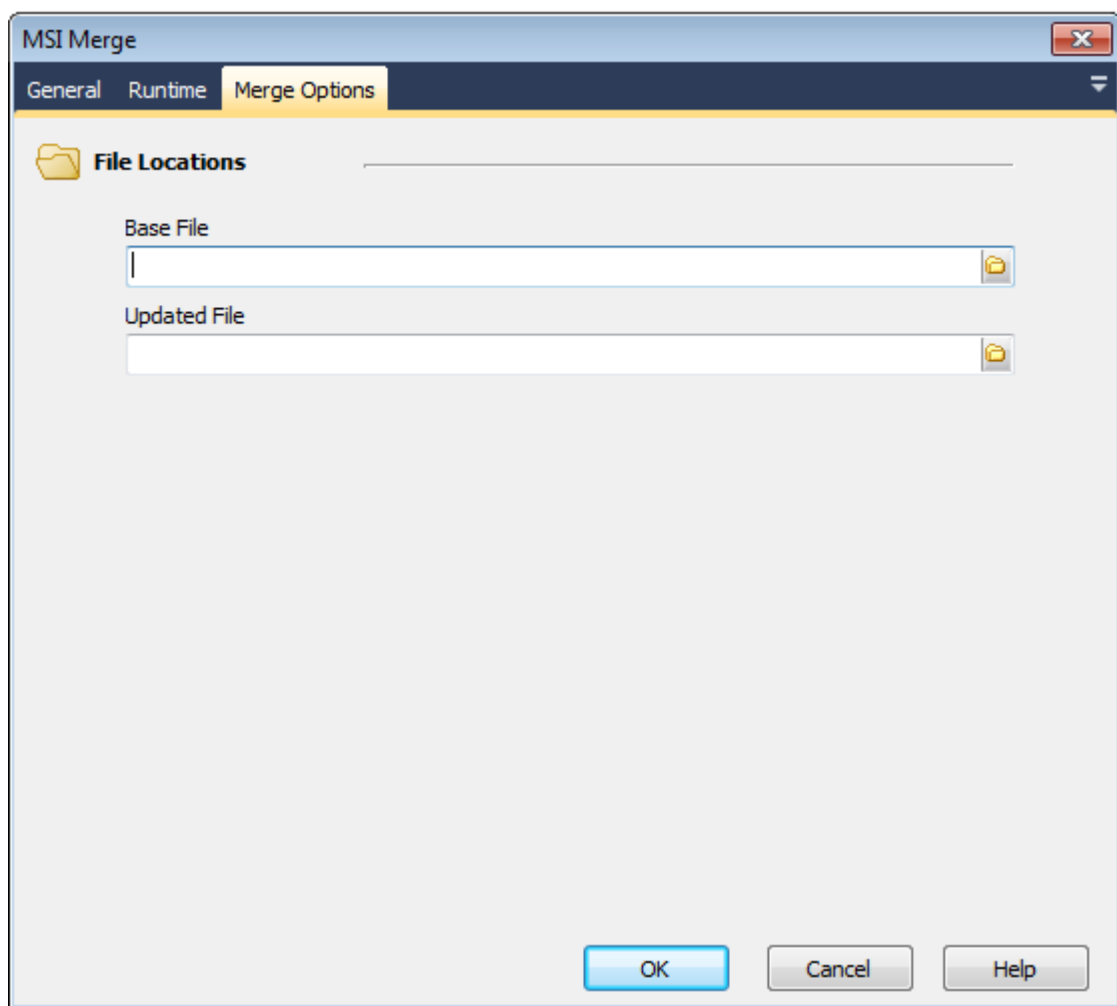
Change the Dialog Font to 'Courier':

```
UPDATE TextStyle SET FaceName='Courier', Size='10' WHERE  
TextStyle='DlgFont8'
```

6.17.12.2 MSI Merge

[FinalBuilder Professional Edition]

The MSI Merge action enables you to merge two MSI databases into one.



Base File

The location of the MSI file that will be merged.

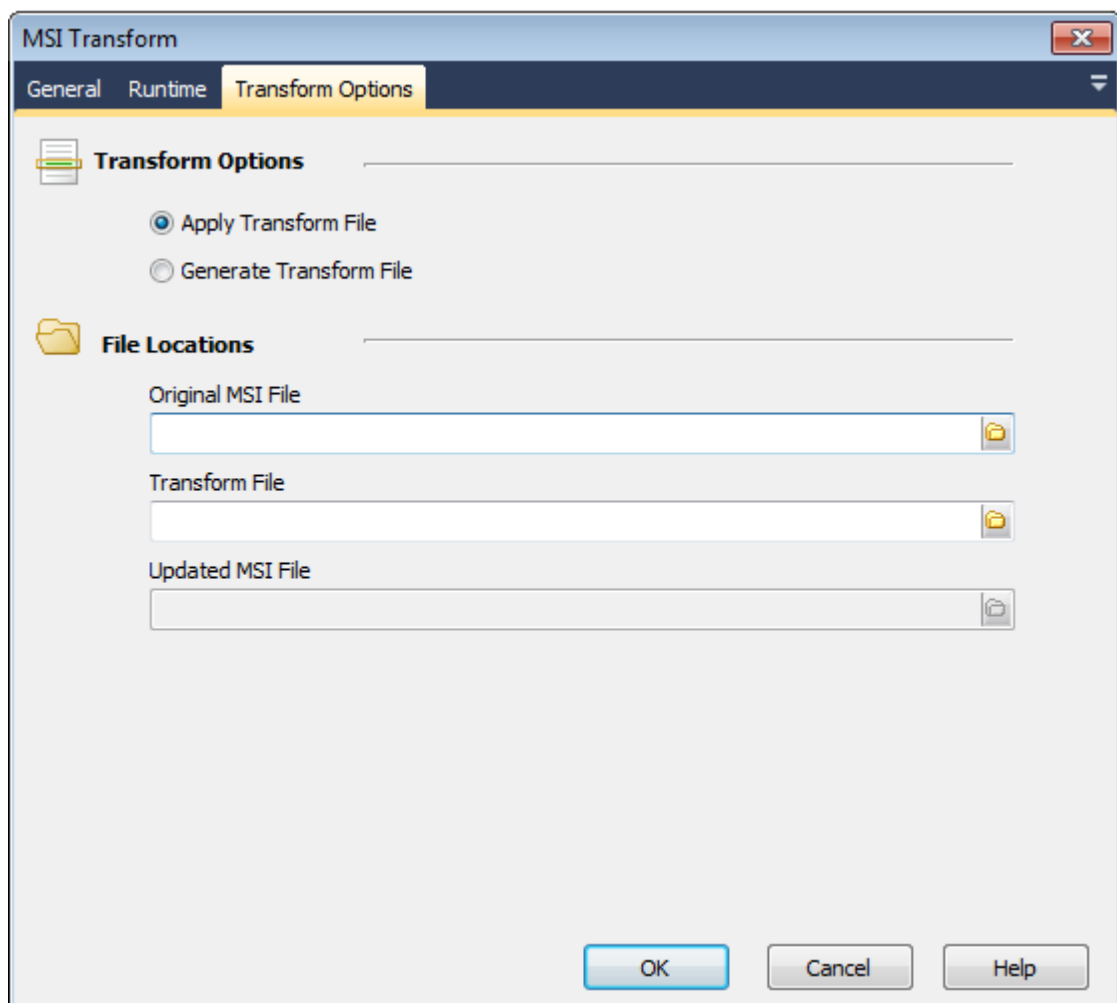
Updated File

The location of the MSI file that contains the changes that should be merged into the Base file.

6.17.12.3 MSI Transform

[FinalBuilder Professional Edition]

The MSI Transform action enables you to generate a MSI transform files, or apply transforms to existing MSI databases.

**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

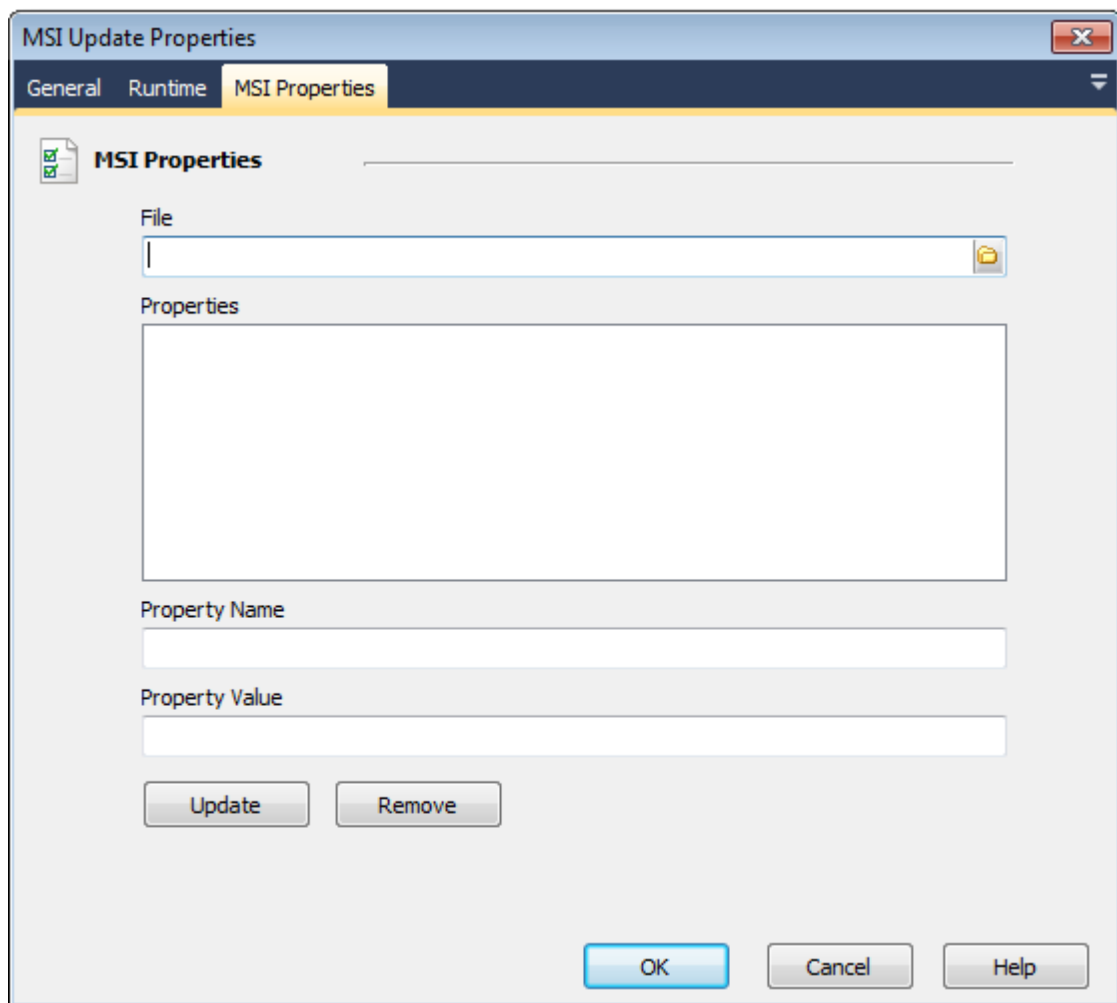
The MSI file which contains the changes to the original file, which will be used when generating a transform file.

6.17.12.4 MSI Update Properties

[FinalBuilder 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

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.

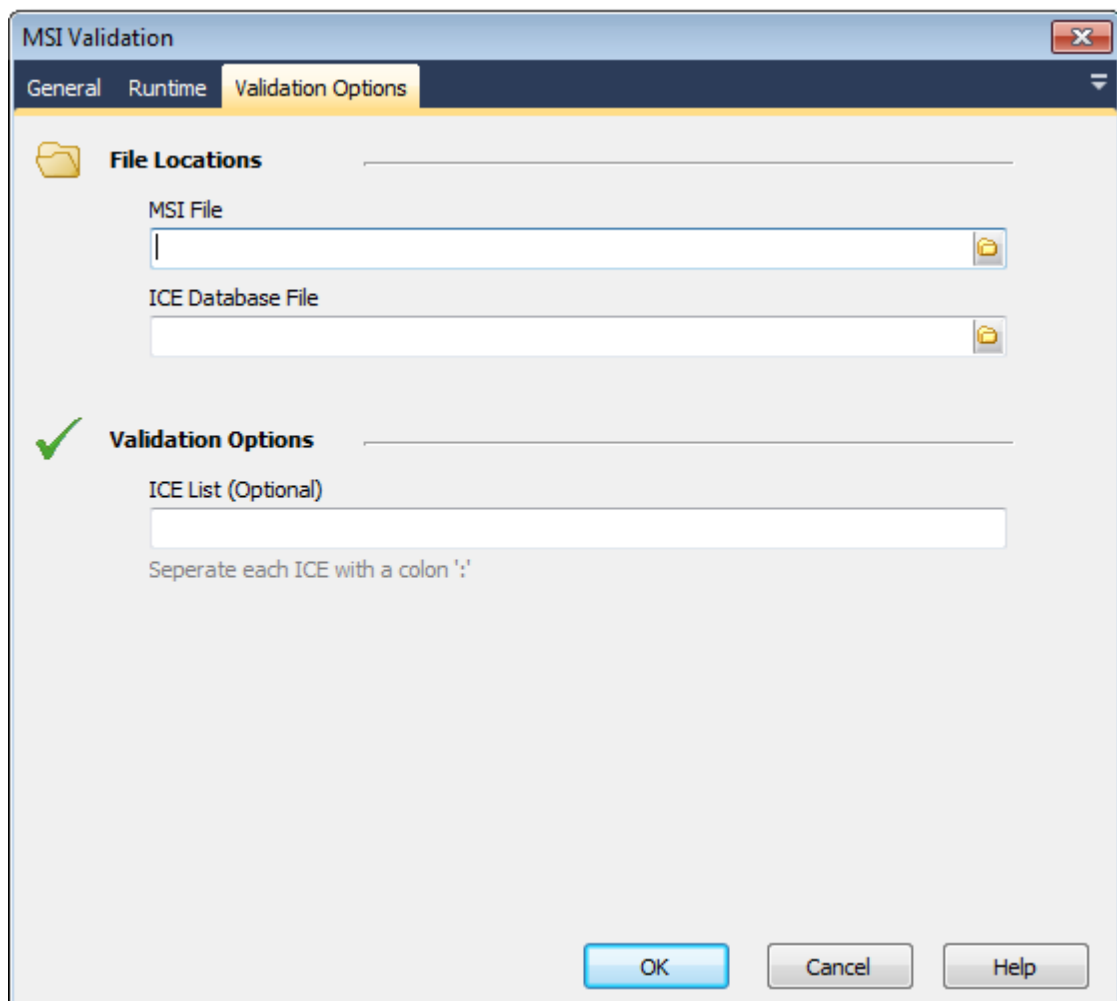
Neither of these buttons will actually remove items from or insert items into the database, they are simply there to allow alteration of the properties list.

6.17.12.5 MSI Validation

[FinalBuilder 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**

The MSI file to perform the validation on.

ICE Database File

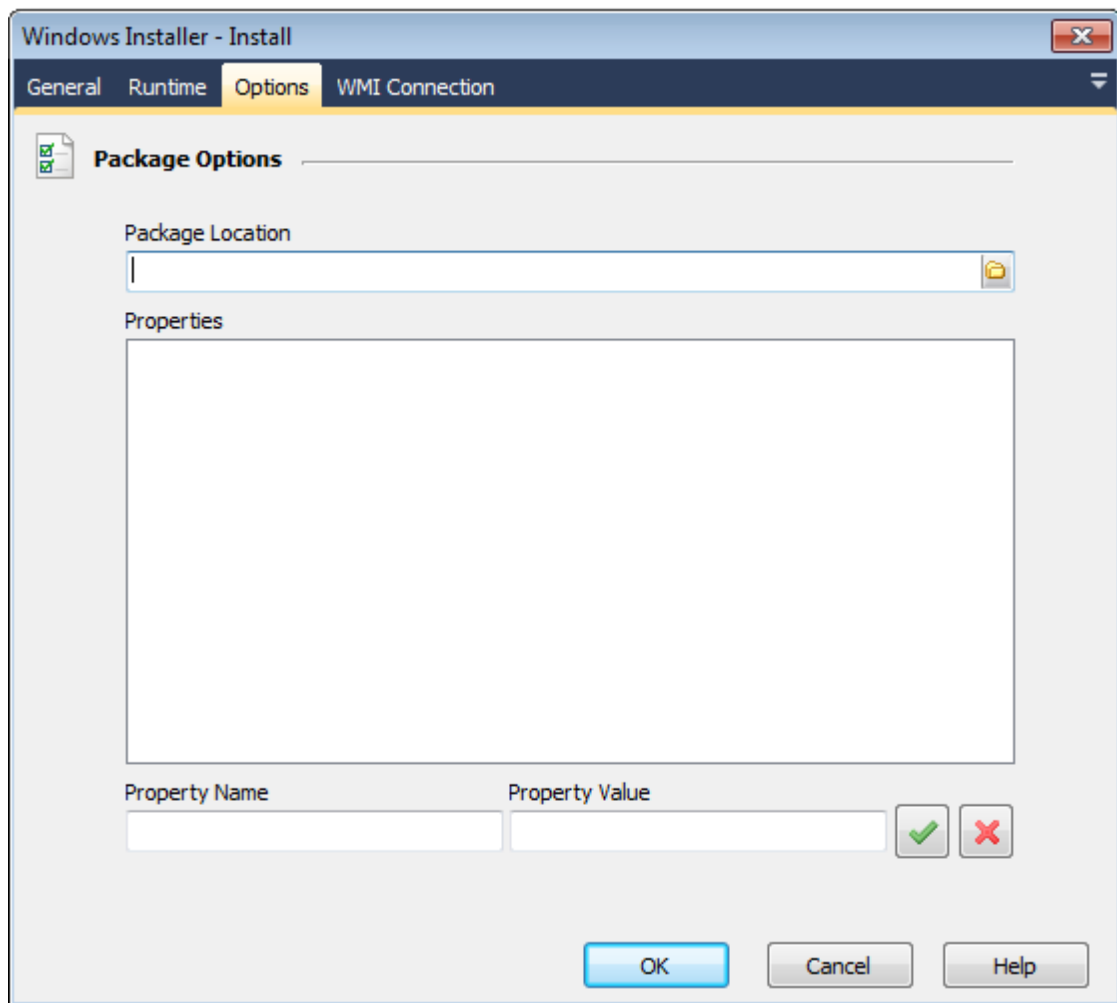
The location of the ICE Database (.Cub) file to use. The MsiVal2 directory contains several .Cub files which can be used.

ICE List

A colon delimited list of the required ICE Modules to be used when performing the validation.

6.17.12.6 Windows Installer - Install

The Windows Installer - Install action enables a windows installer package to be installed on a local or remote computer.

**Package Location**

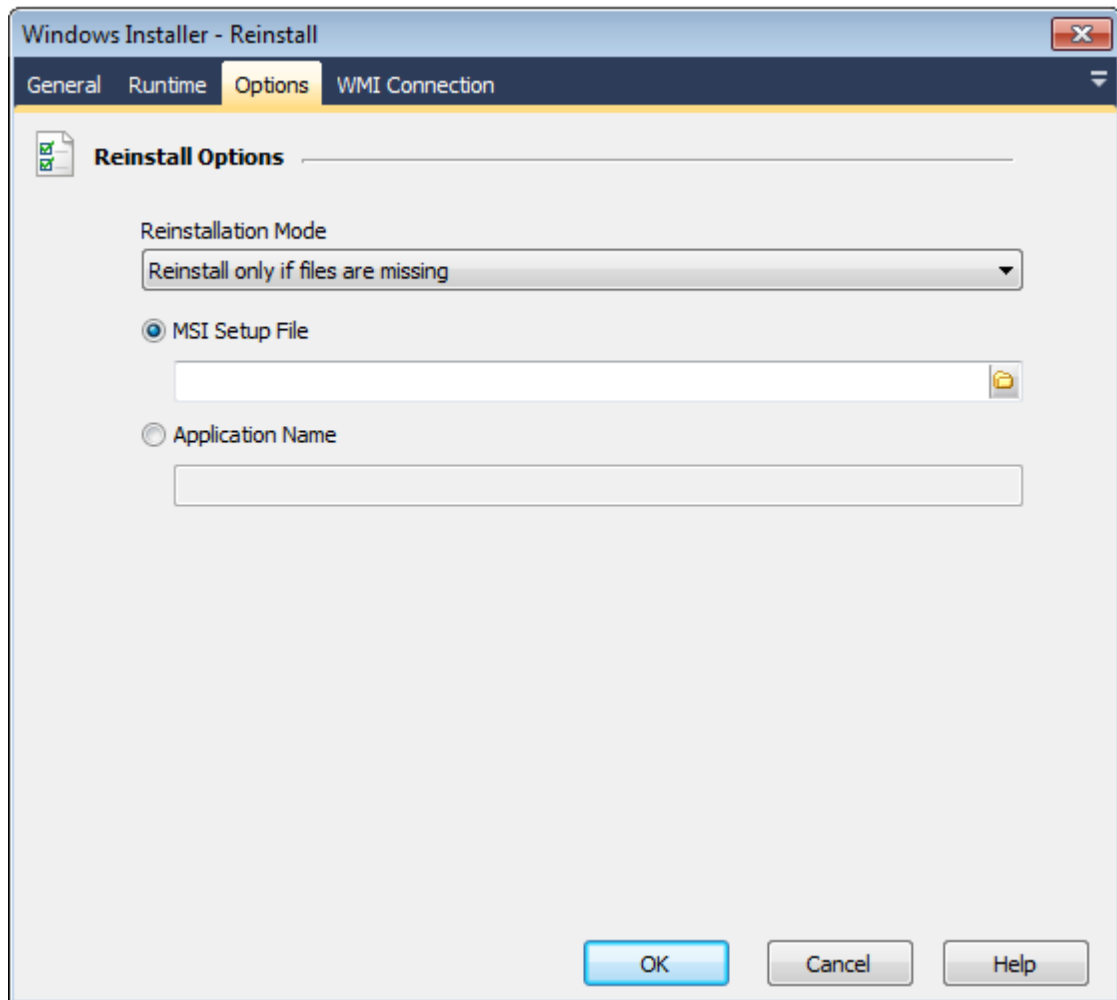
The location of the package to be installed. The package must be accessible by the computer on which it is being installed.

Properties

The properties to be passed into the installer.

6.17.12.7 Windows Installer - Reinstall

The Windows Installer - Reinstall action enables the reinstallation of an application installed by a windows installer package.

**Reinstallation Mode**

The type of reinstallation to be performed.

MSI Setup File

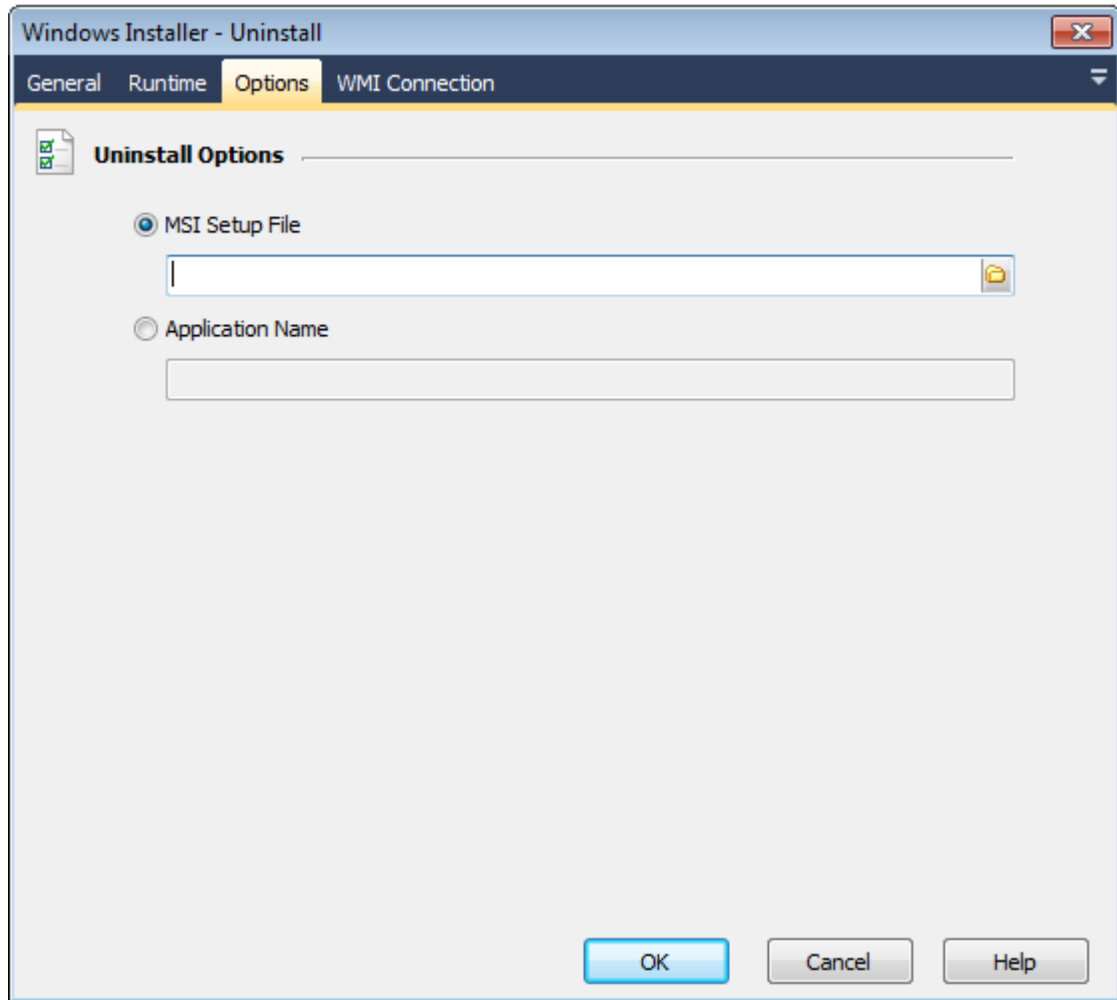
The setup file used to install the application. The setup file does not need to be accessible by the computer which the application is being uninstalled on, as it is only used to get the products identification number.

Application Name

The name of the application to uninstall. If more than one application is found with the same name, only one of the applications will be uninstalled.

6.17.12.8 Windows Installer - Uninstall

The Windows Installer - Uninstall action allows a application installed by a windows installer package to be uninstalled.



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.

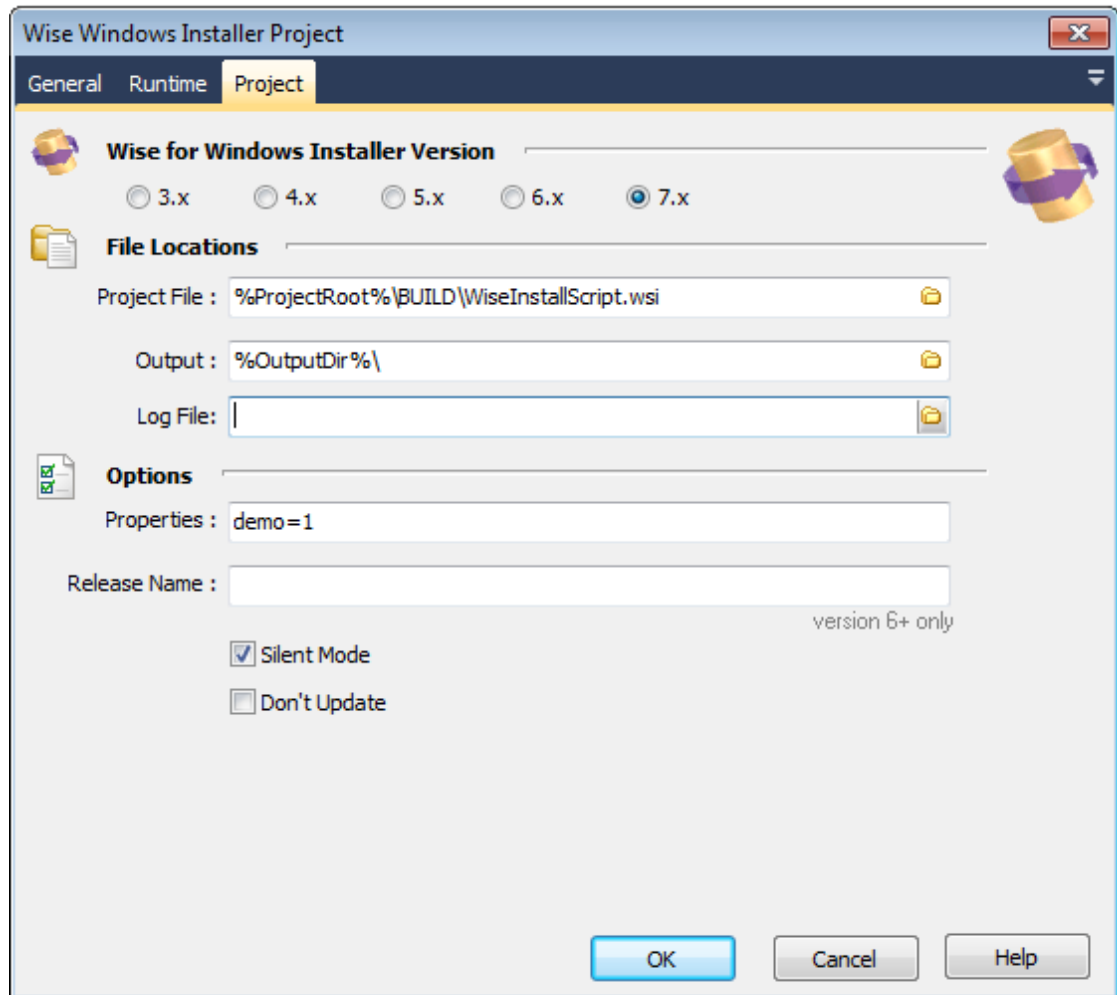
Application Name

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.

6.17.13 Wise For Windows Installer

This action provides an interface to Wise for Windows Installer versions 3, 4 and 5.

For newer versions of Wise, see the Wise InstallBuilder/InstallMaster action.



Project File : The full path and file name of the wise project file (.wse).

Output Directory : Output File Path.

Properties : Project Properties, format is name=value;name=value

Scripting Info

The Action properties available are :

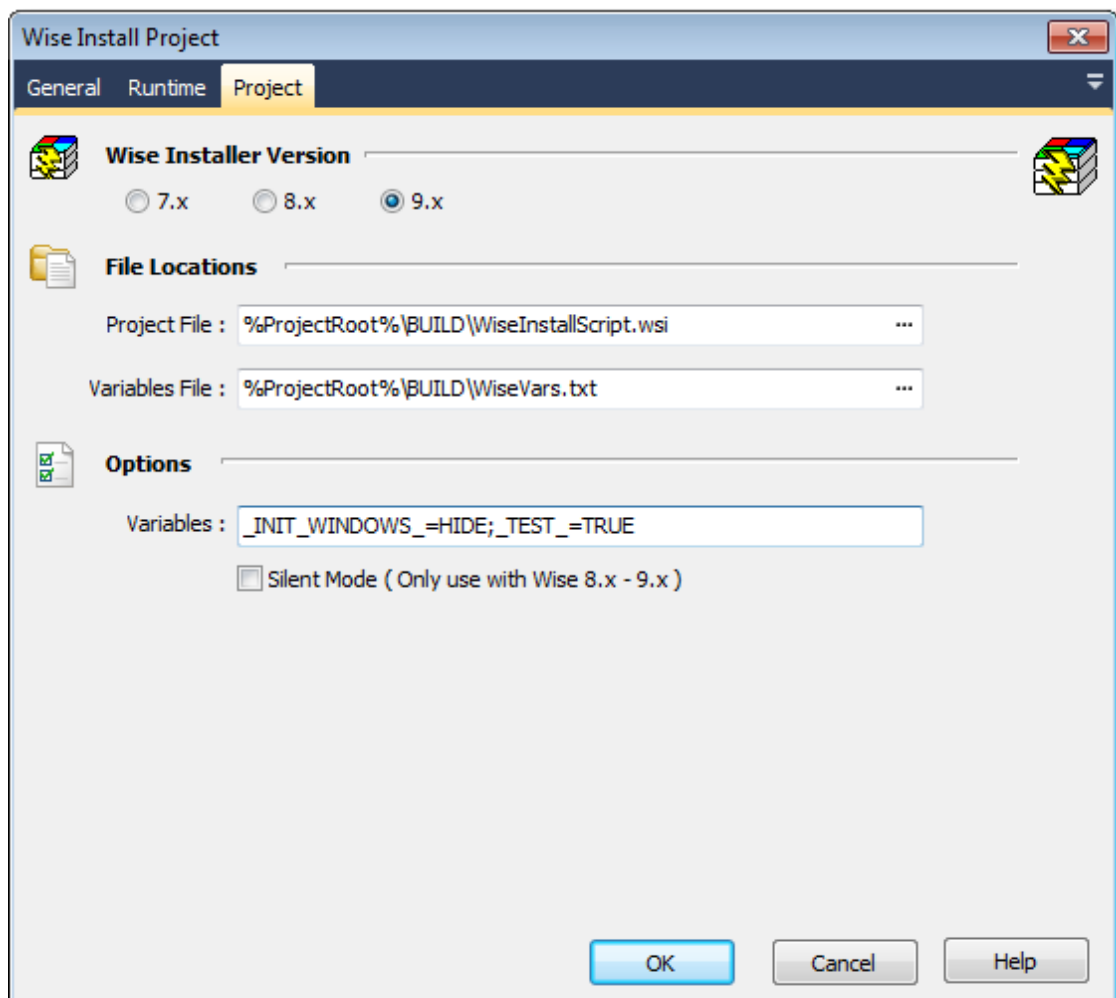
property ProjectFile : WideString
property Properties : WideString
property OutputFilePath : WideString

6.17.14 Wise InstallBuilder/InstallMaster

This action provides an interface to Wise InstallMaster/InstallBuilder 7.0, 8.x and 9.x.

Later versions work as well, although new features may not be supported.

For versions 3.x, 4.x and 5.x, see the Wise For Windows Installer action.



Project File : The full path and file name of the wise project file (.wse).

Variables File : The full path and file name to a file that contains compiler variables. Format is `_VAR_=value`, one entry per line.

Variables : Compiler variables, format is `_VAR_=value;_ANOTHER_VAR_=value`

NOTE: The Wise compiler does not return a non zero return code when an error occurs during compilation. This means that FinalBuilder has no way of detecting if the compile of the installer failed or not. Usually Wise displays a message box when an error occurred.

Scripting Info

The Action properties available are :

property ProjectFile : WideString
property Variables : WideString
property VariablesFile : WideString

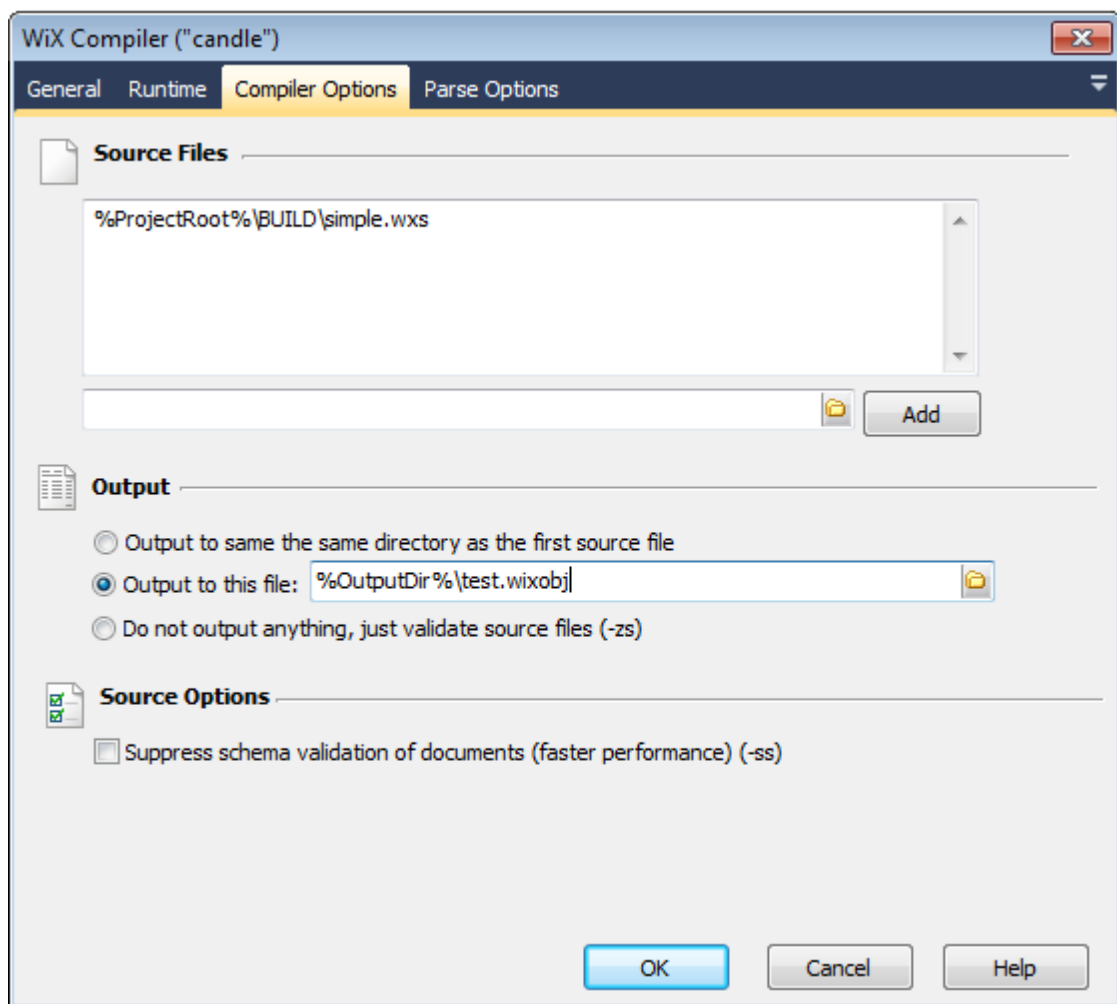
6.17.15 WiX (Windows Installer XML Toolkit)

WiX - The Windows Installer XML Toolset - builds Windows Installer Packages from XML source code. WiX generates a .MSI Windows Installer Package in two steps - compiling & linking.

WiX is an open source application. For more information, please see the [WiX project homepage](#).

6.17.15.1 WiX Compiler (candle.exe)

This action provides an interface to the WiX compiler (candle.exe.) The compiler compiles .wxs source files to .wixobj object files, which are then linked by the WiX Linker (light.exe)



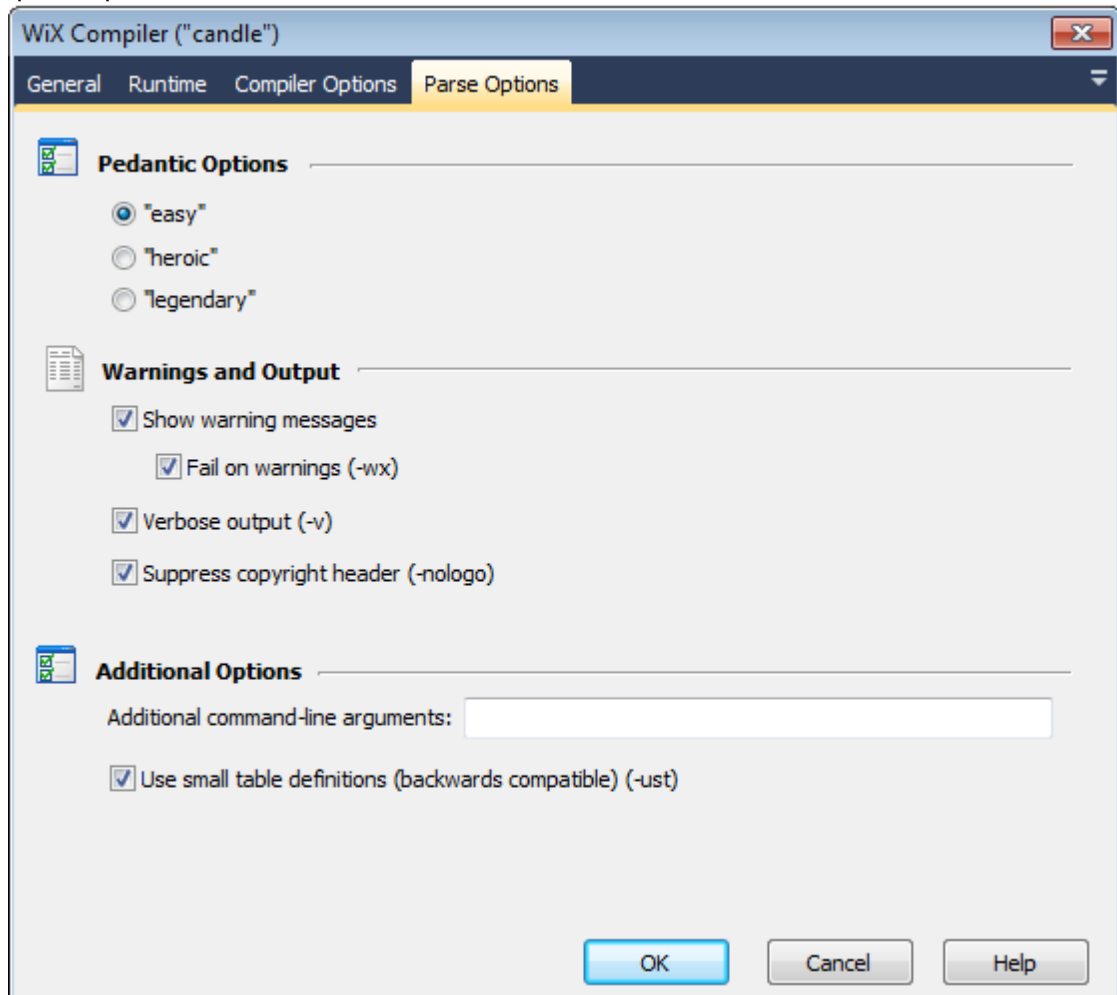
Source Files

Specify one or more .wxs files to compile. For information on the .wxs format, see the 'doc' directory of your WiX installation.

Output

Output can be delivered to the default directory or a specific object file. Alternatively, the source files can just be parsed and no object files generated.

#parseoptions



Pedantic Checks

Specifies how closely the compiler should check source file syntax. "Legendary" is more pedantic than "Heroic", and "Heroic" is more pedantic than "Easy" (the names correspond to the command line options.)

Show Warning Messages

Warning messages will be recorded in the FinalBuilder log.

Fail on Warnings

Warnings will be treated as errors and the build will stop if any occur.

Verbose Output

More progress data will be logged (useful for debugging.)

Suppress copyright header

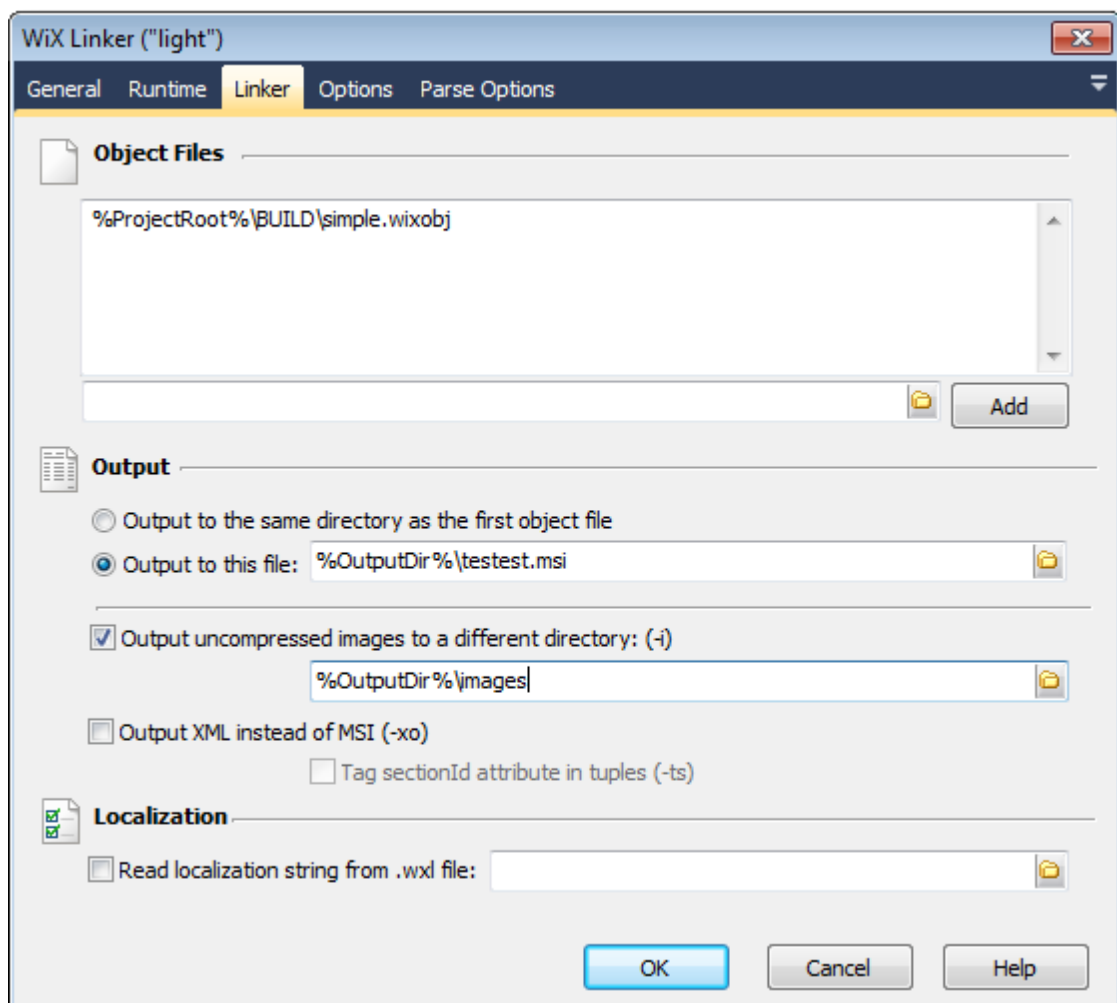
The WiX copyright header will not be displayed. Useful for a smaller log size.

Additional Command-Line Arguments

Type any additional command line arguments that you want to pass to WiX.

6.17.15.2 WiX Linker (light.exe)

This action provides an interface to the WiX linker (light.exe) The linker links .wixobj object files into a .msi Windows Installer Package.



Object Files

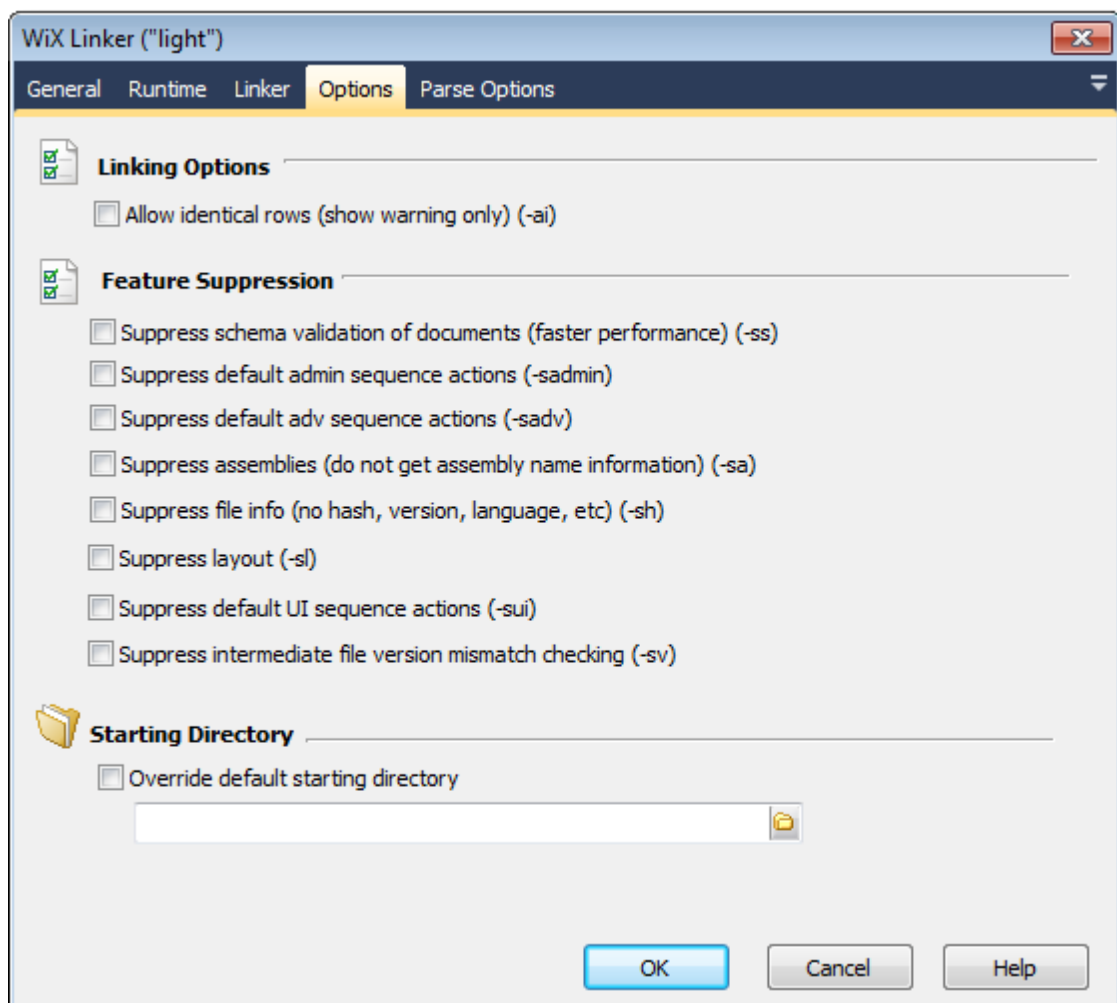
Specify one or more .wixobj object files that you wish to link.

Output

Output can be sent to the default directory, or a specific .msi installer package file. Alternatively, uncompressed images can be output to a different directory. The linker can also output raw XML instead of MSI (useful for debugging.)

Localization

A .wxl file can be used to create a localized installer package.

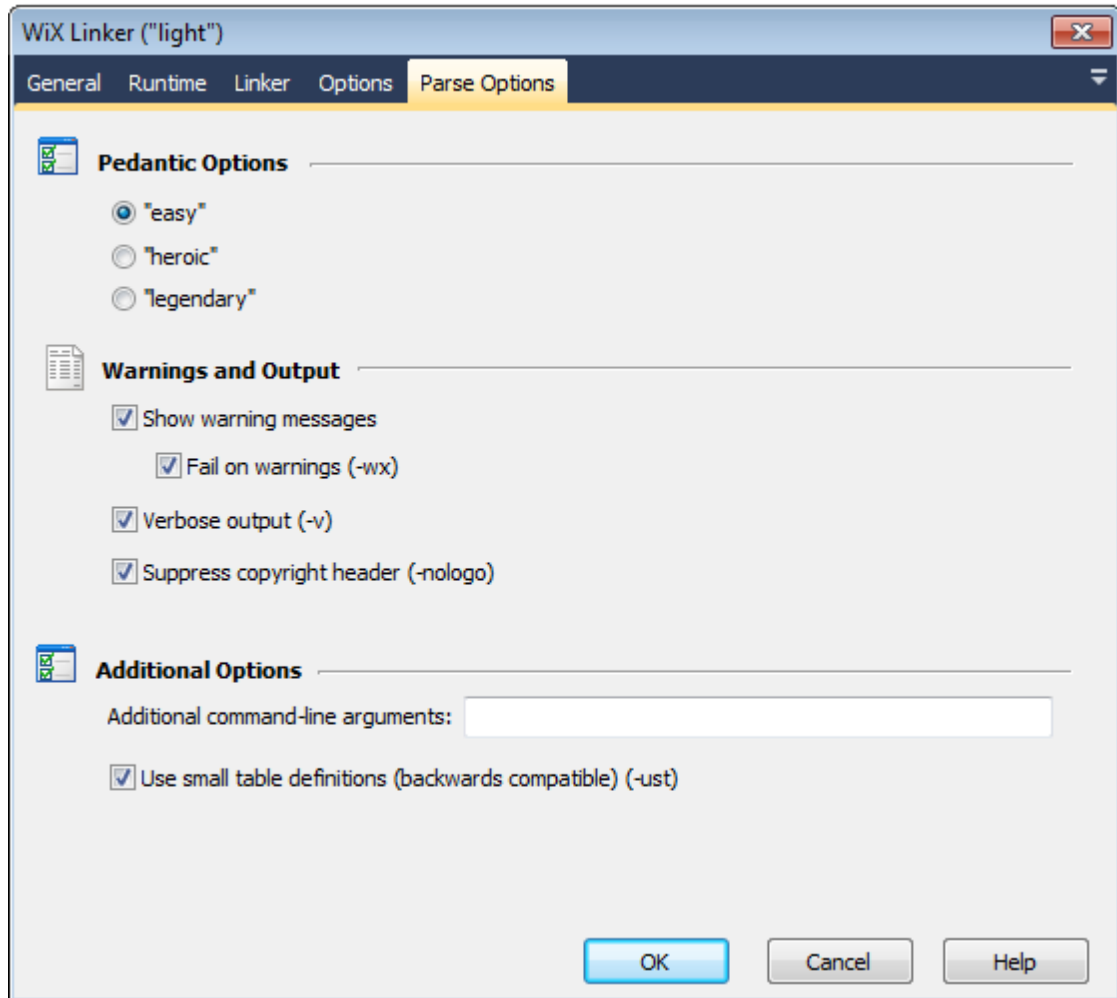


Allow Identical Rows

Normally the WiX Linker will fail if it detects identical rows. With this option checked it will only emit a warning.

Feature Suppression

Various aspects of the WiX Link process can be bypassed. It is not recommended that you use these options unless you are sure you do not need the features specified.



For information on these options, see the WiX Compiler help topic.

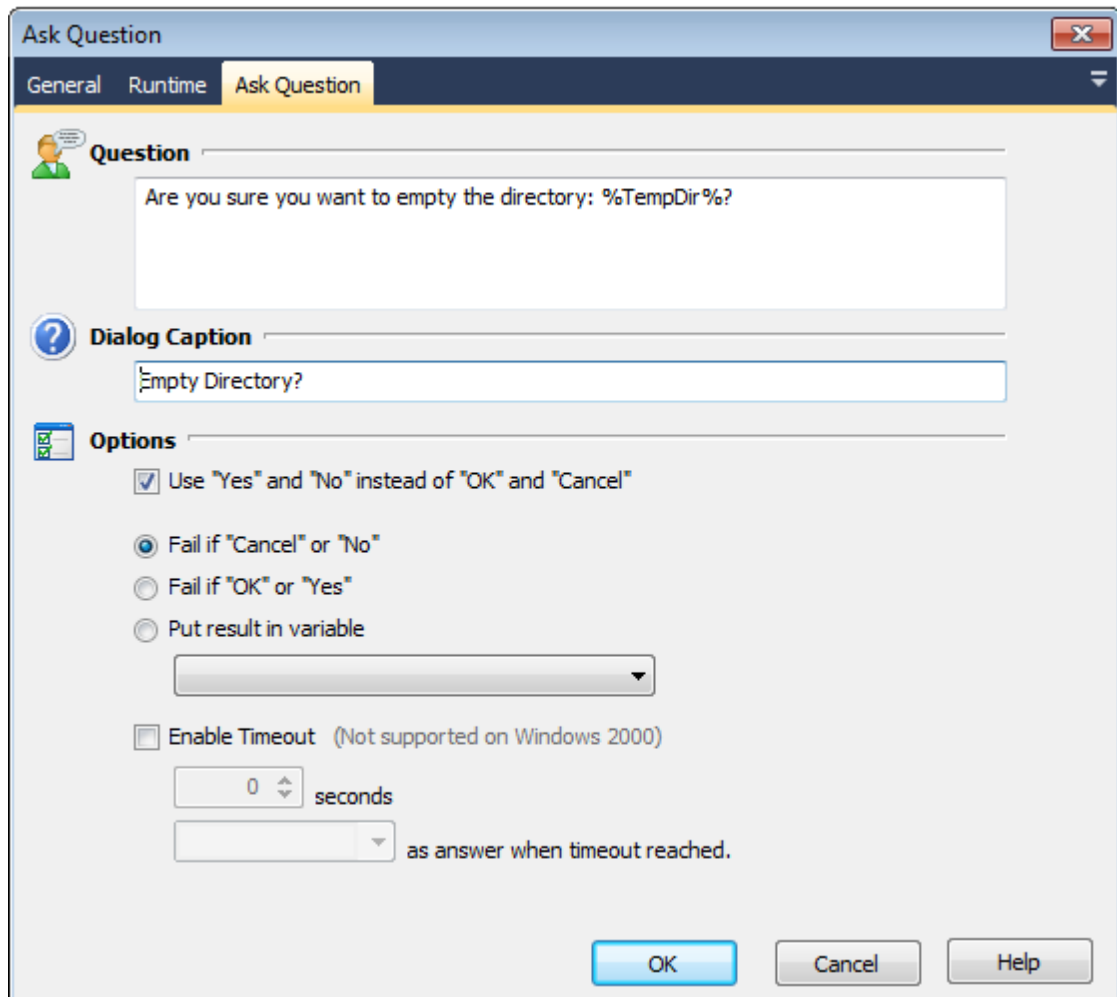
6.18 Interactive

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.

6.18.1 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 FinalBuilder 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 FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

**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.

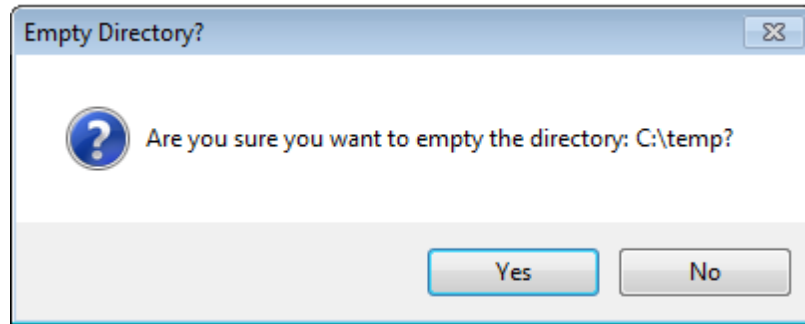
Options

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

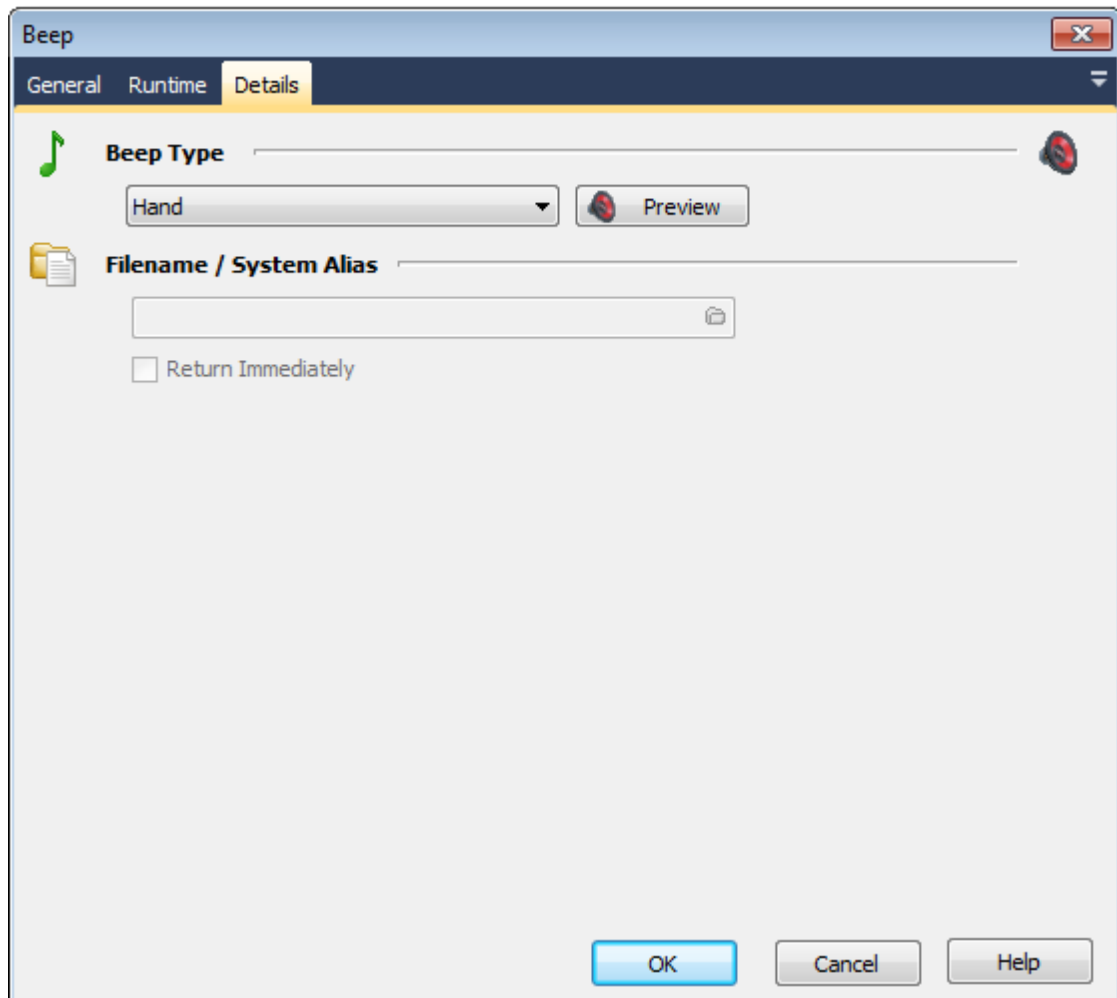
fail if the user chooses OK/Yes, or not to fail at all and instead save the result to a variable. If you choose to store the result in a FinalBuilder variable, then OK / Yes will set the variable to True, and Cancel / No will set the variable to False

Below is an example of what the dialog will look like at runtime:



6.18.2 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. FinalBuilder creates two System Aliases, Build Complete and Build Error. You can change the sounds assigned to these aliases in the windows control panel.

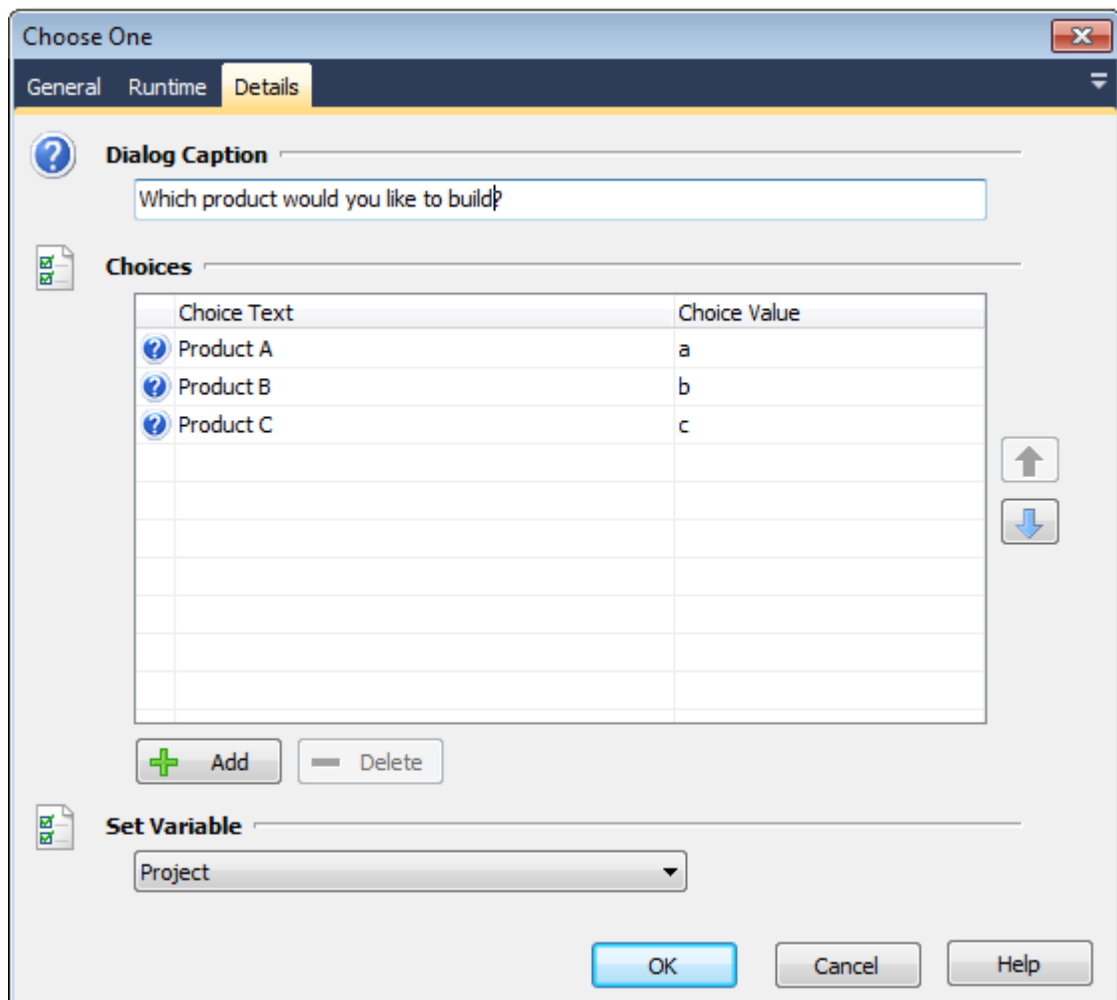


6.18.3 Choose One Action

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 FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.



Dialog Caption

The text to display above the choices

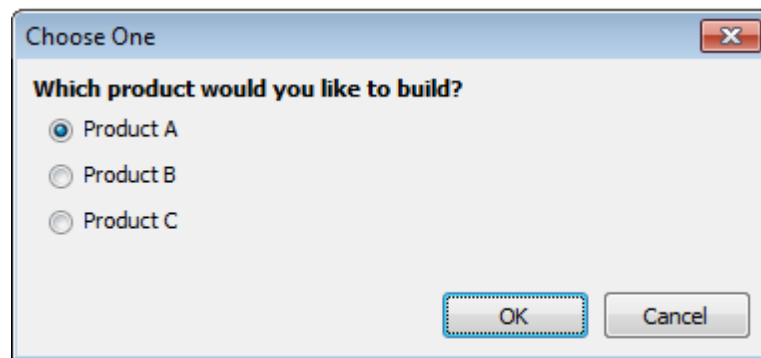
Choices

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:



The first option is always chosen as default. Pressing OK will set the variable, whereas Cancel will fail the action.

6.18.4 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](#).

Note: This action should not be used in unattended builds as it will cause them to hang waiting for user input. By default, interactive actions are not available from FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

This action was donated by Peter Thornqvist.

6.18.5 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](#).

Note: This action should not be used in unattended builds as it will cause them to hang waiting for user input. By default, interactive actions are not available from FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

This action was donated by Peter Thornqvist.

6.18.6 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 FinalBuilder 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](#).

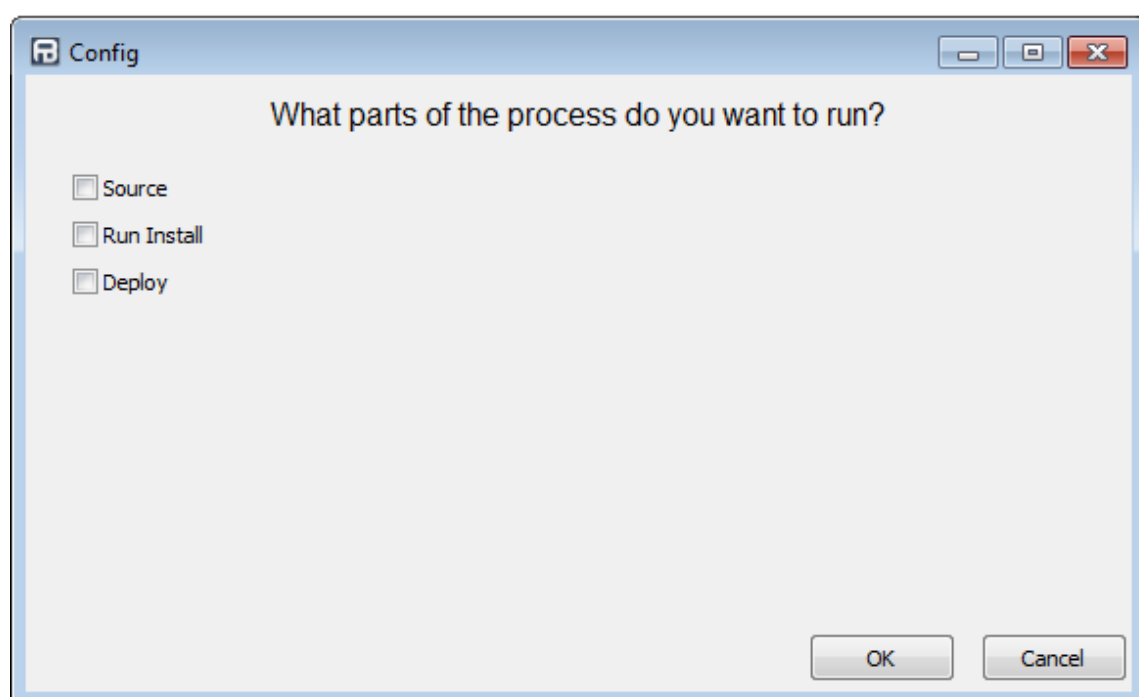
Note: This action should not be used unattended as it will cause them to hang waiting for user input. By default, interactive actions are not available from the FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

Answer Text	Set Boolean Variable	Default
Source	DoCompileSource	<input type="checkbox"/>
Run Install	DoInstallers	<input type="checkbox"/>
Deploy	DoInstallers	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

For Example, the main question could be "What parts of the build 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.

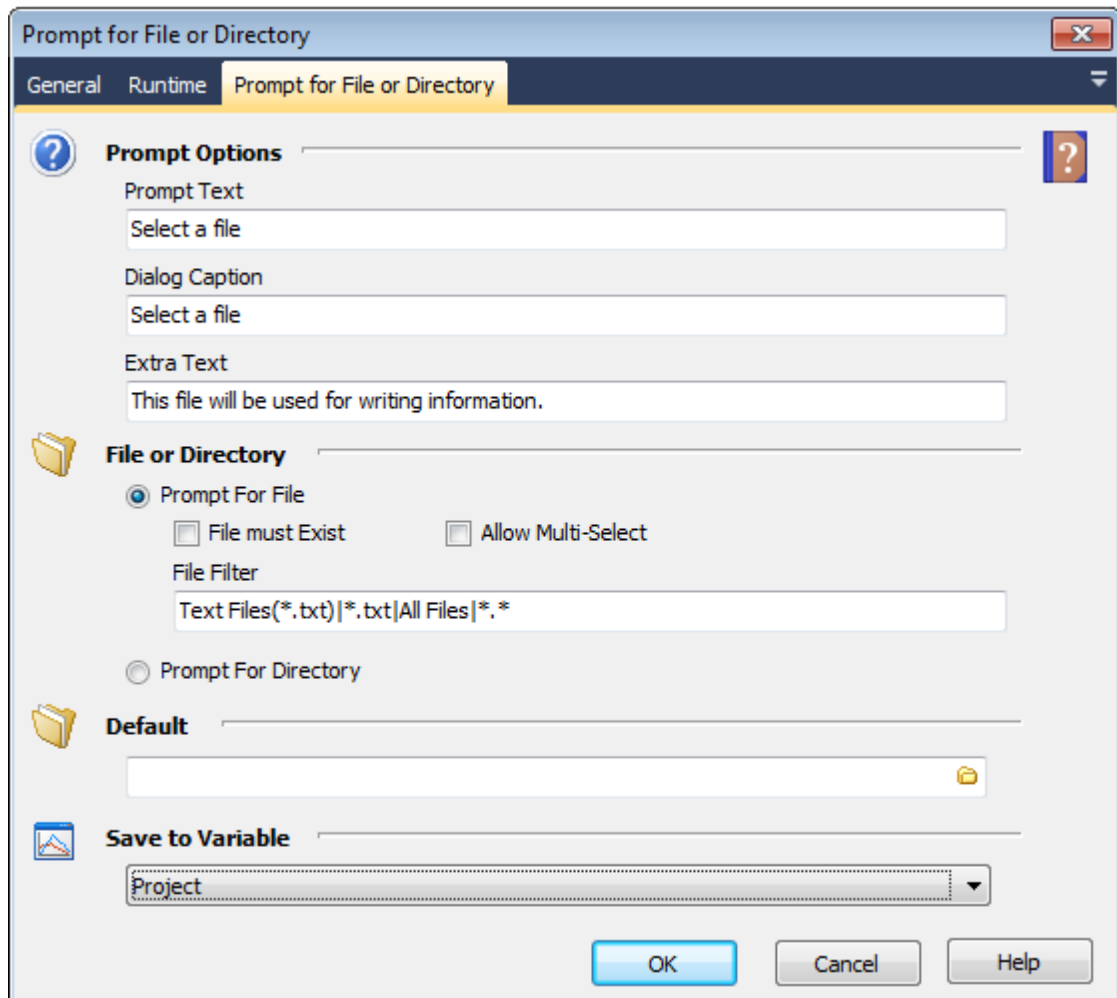
This is what the action looks like at run time:



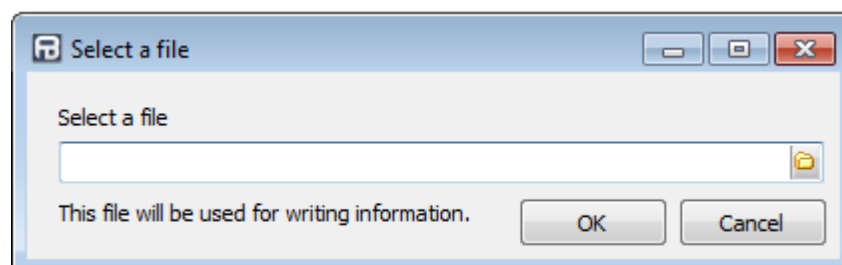
6.18.7 Prompt for File or Directory

The Prompt for File or Directory action enables you to ask the user to specify a file or directory during your build. 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 builds as it will cause them to hang waiting for user input. By default, interactive actions are not available from FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.



When the action is executed it looks something like this:



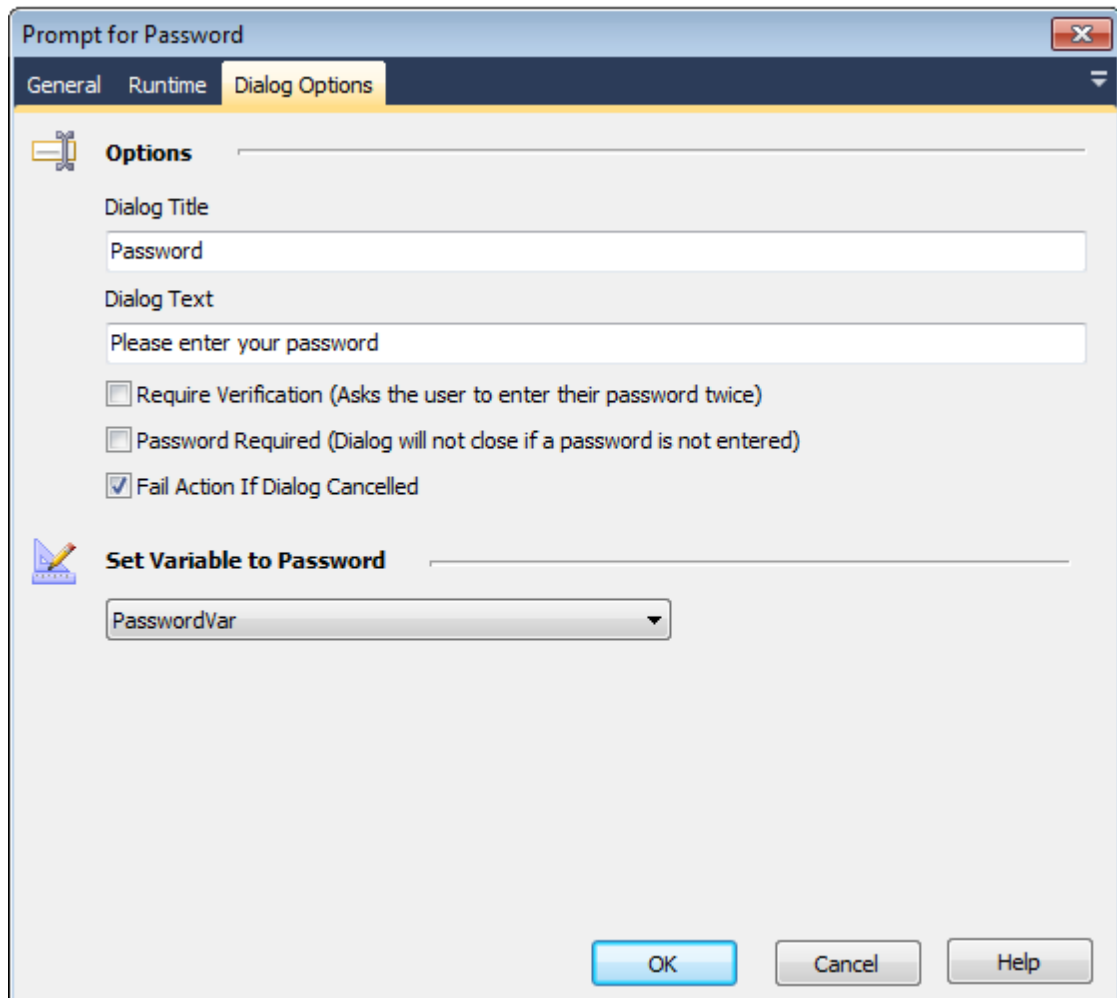
6.18.8 Prompt for Password Action

This action allows you to prompt the user to enter a password at run time.

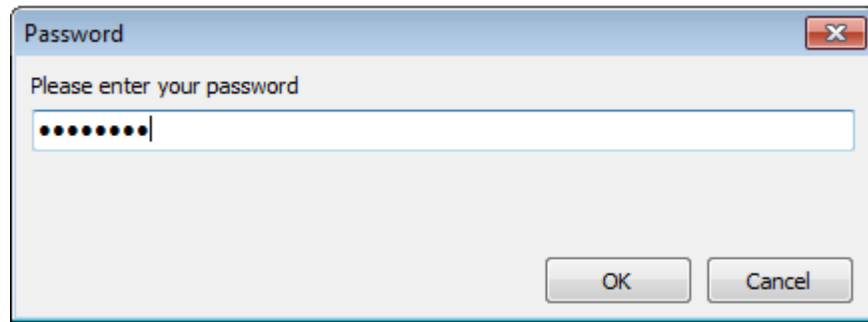
For more information on variables, including creating your own project and user variables, see [here](#).

Note: By default, interactive actions are not available from the FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

Prompt for Password Options



Prompt for Password Dialog

**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

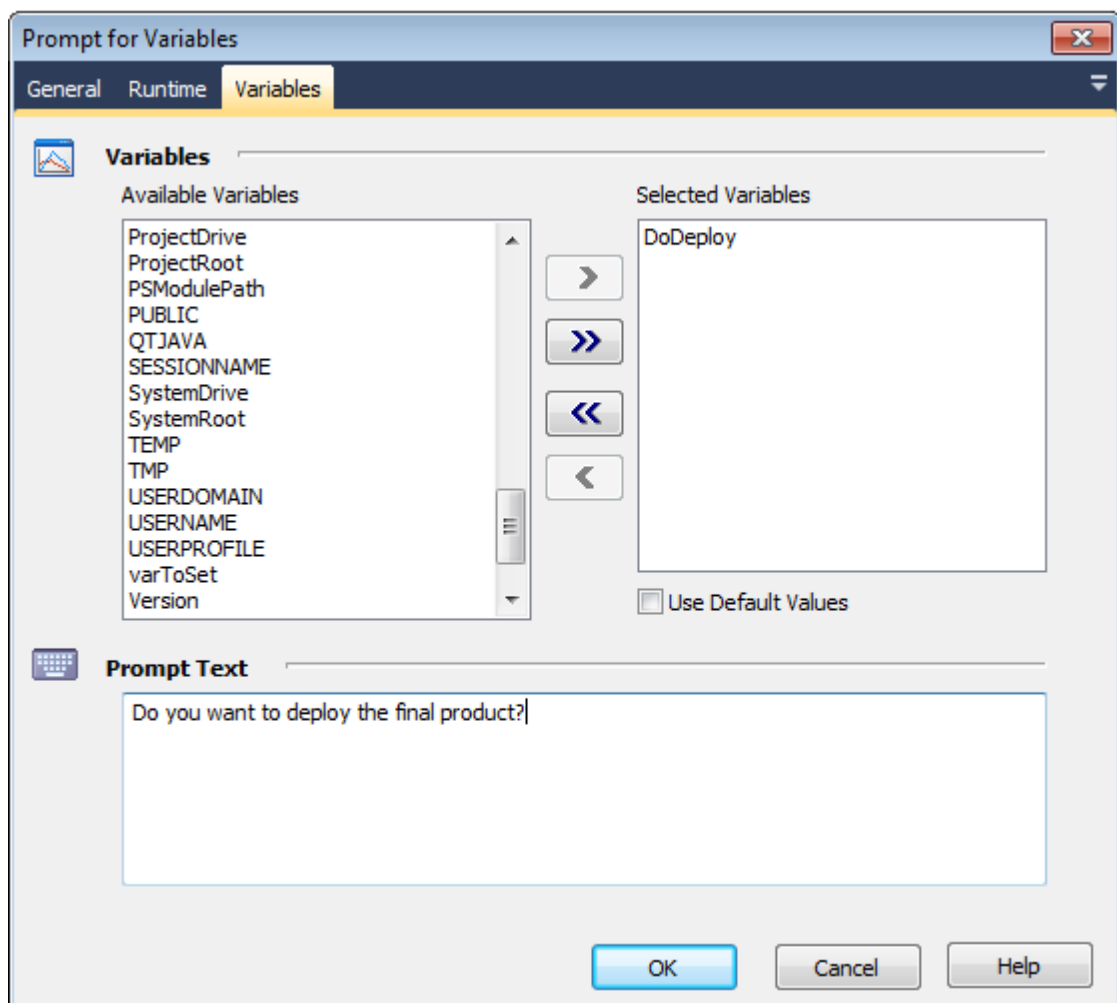
Select the variable for the password to be saved back into. This variable may then be subsequently used in other actions that require the password.

6.18.9 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 FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

For more sophisticated prompts requirements, use the Prompt For Variables (Enhanced) action.

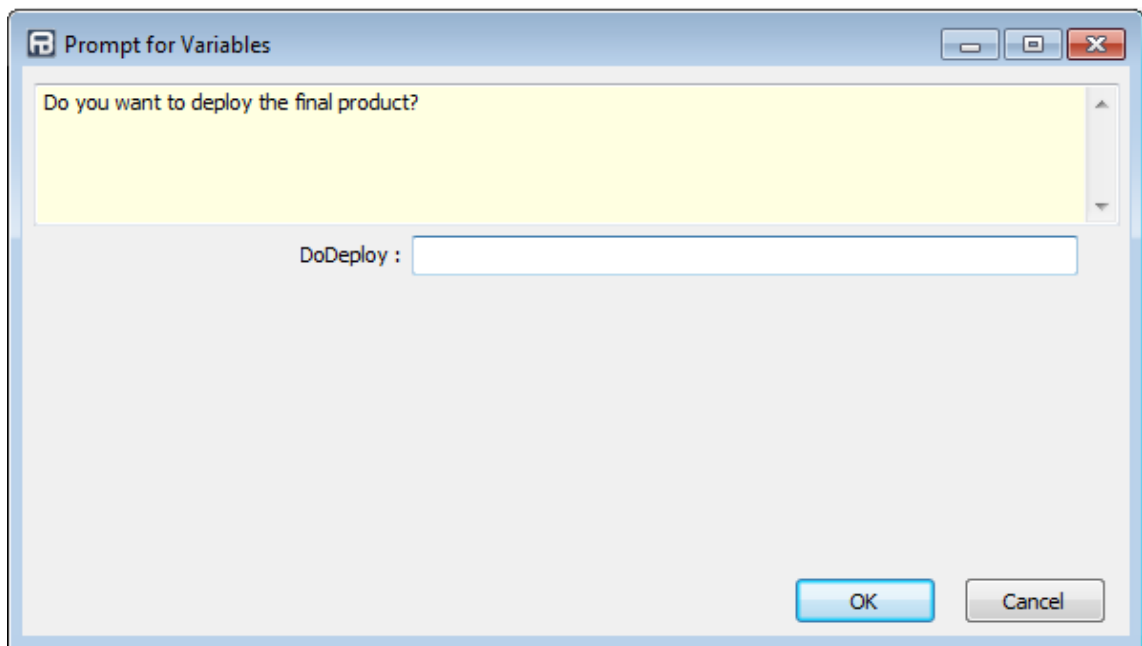


Available Variables & Selected Variables - Move Variables from the Available Variables side to the Selected Variables side to prompt for them when the action runs

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.



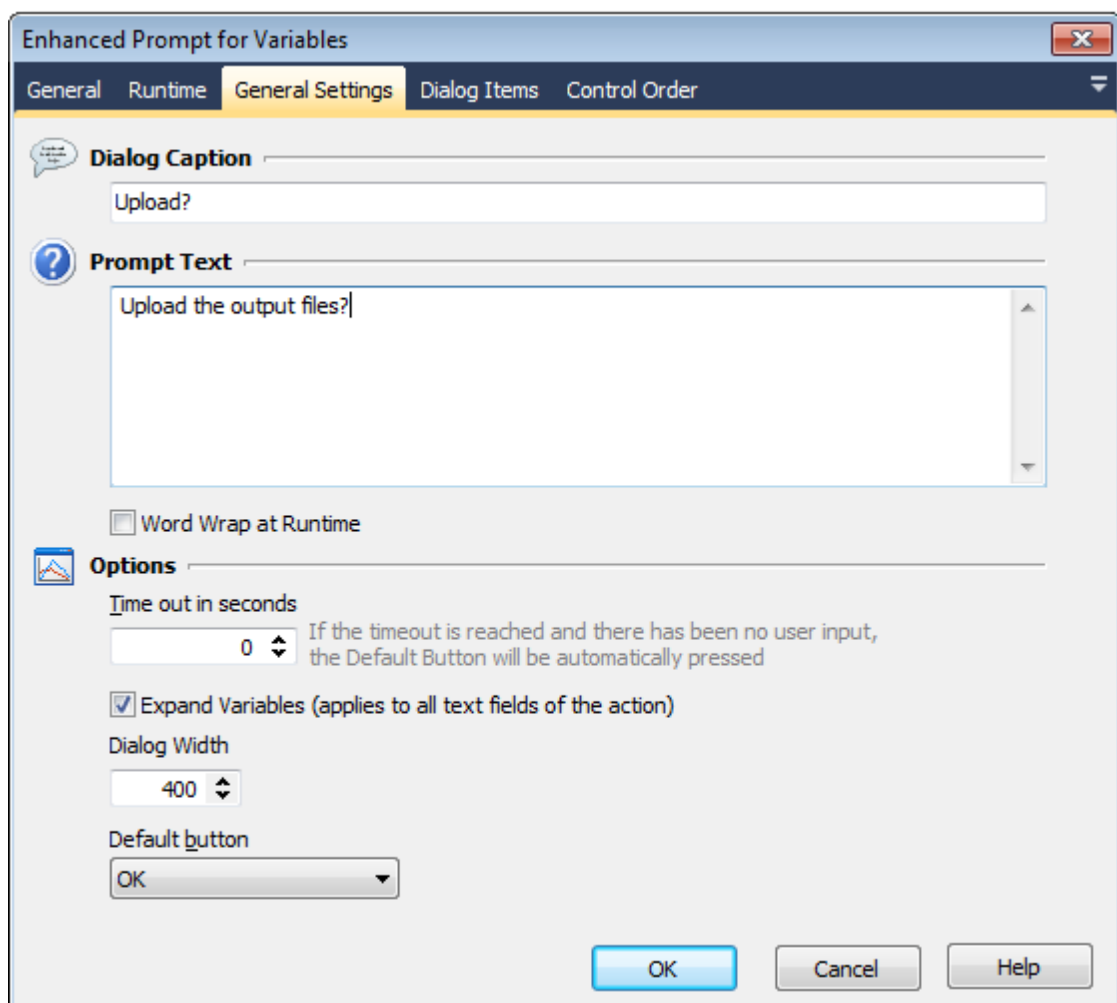
6.18.10 Enhanced Prompt for Variables Action

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 FinalBuilder console. Interactive actions cannot be used inside Async Action Groups.

The general settings lets you customise the dialog appearance.



Dialog Caption - the text to appear in the dialog header.

Prompt Text - the text to appear at the top of the dialog (optional).

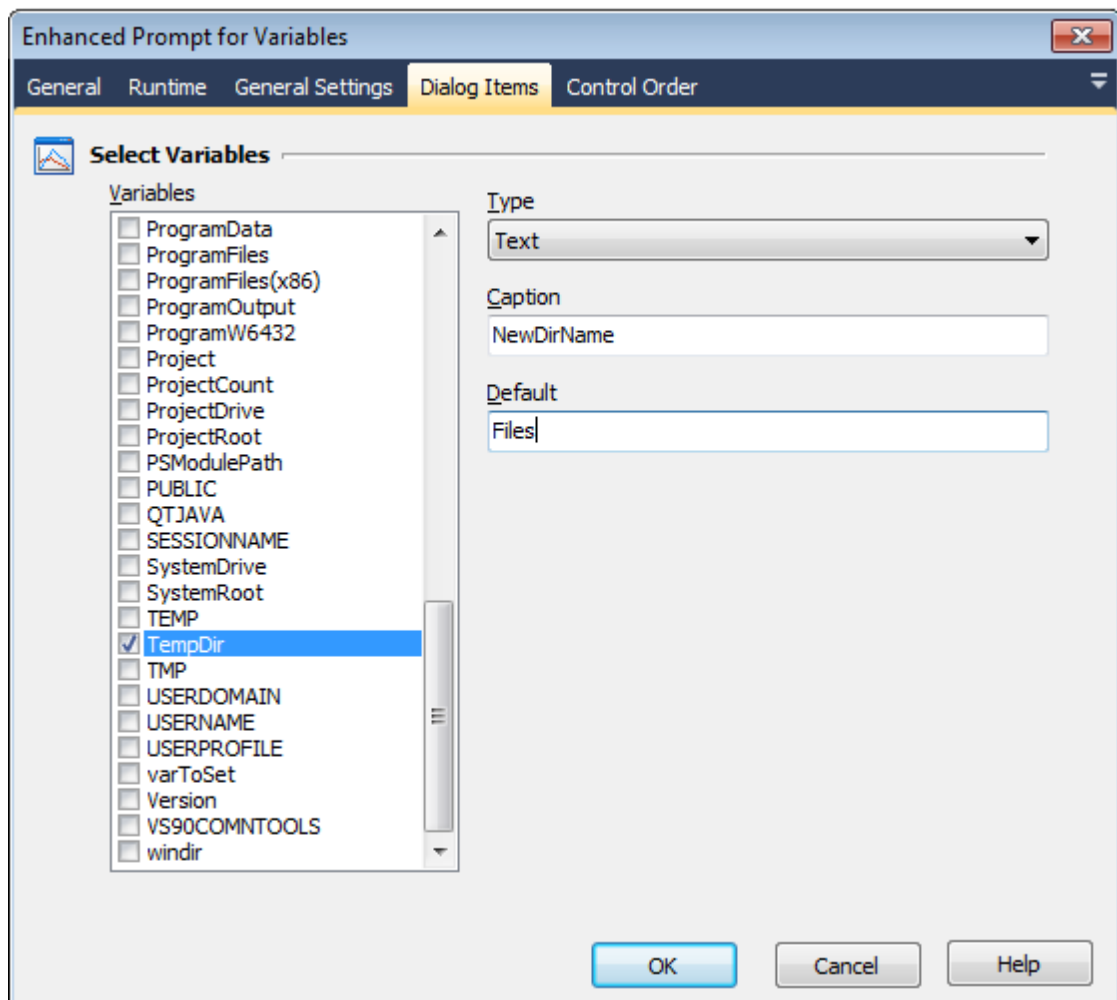
Word Wrap at Runtime - set this to automatically wrap the prompt text according to the width of the dialog.

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 - 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

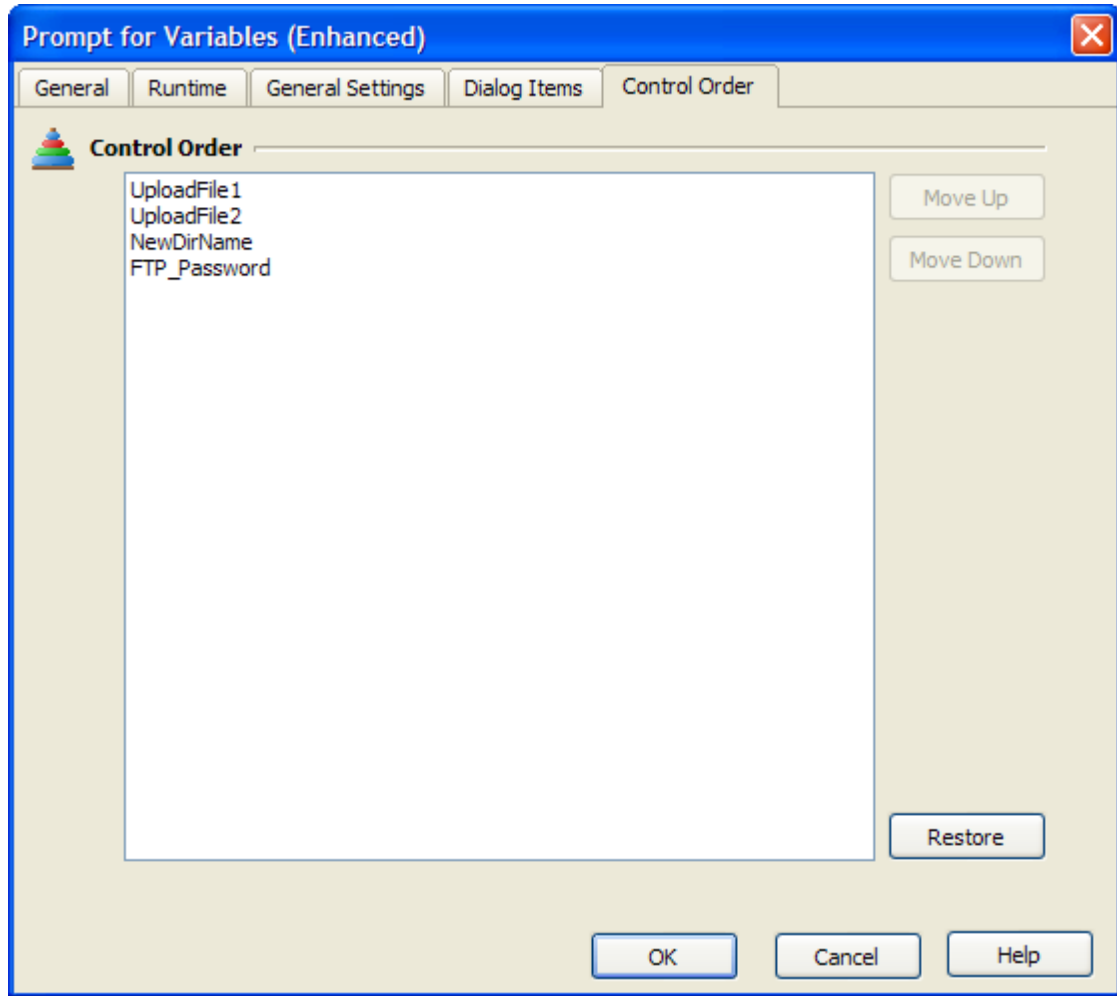
Default - the default value of the control

Values can have initial defaults (which can be variable values or fixed values.)

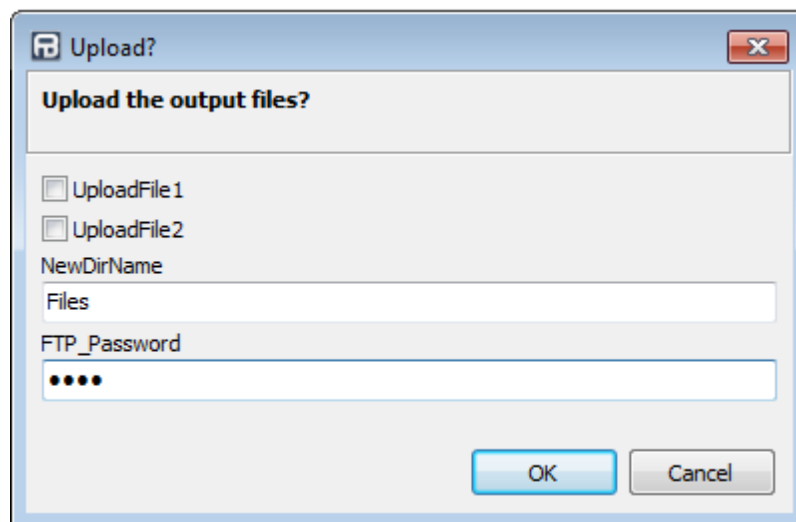
For "Check List" type prompts, specify default values as a comma-delimited list of values to enable at runtime.

Control Order

This tab allows you to set the order of the various controls as they will appear on the runtime dialog



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.



6.19 Internet

6.19.1 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.

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]

6.19.2 FTP Actions

The FTP Actions allow you to add various FTP commands to your build.

Use the FTP Connection to define a connection, and then specify this connection in the other FTP actions. Your build should include an FTP Disconnect action to close the FTP connection.

Example:

Description	Action	Enabled	Status
 FTP Connect [website]	FTP Connect	<input checked="" type="checkbox"/>	Completed
 FTP Change Directory [website] [DNN]	FTP Change Directory	<input checked="" type="checkbox"/>	Completed
 FTP Upload [website] [AT-icon.png]	FTP Upload	<input checked="" type="checkbox"/>	Completed
 FTP Disconnect [website]	FTP Disconnect	<input checked="" type="checkbox"/>	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 - 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 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

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\FinalBuilder\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

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

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 FinalBuilder 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 FinalBuilder variable - specify the variable to hold the remote directory.

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 FinalBuilder log.
- Put listing in FB variable - if a variable is specified, then the listing of files is saved to the FB variable

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.

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

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 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.

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.

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.

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 build, 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)

6.19.2.1 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

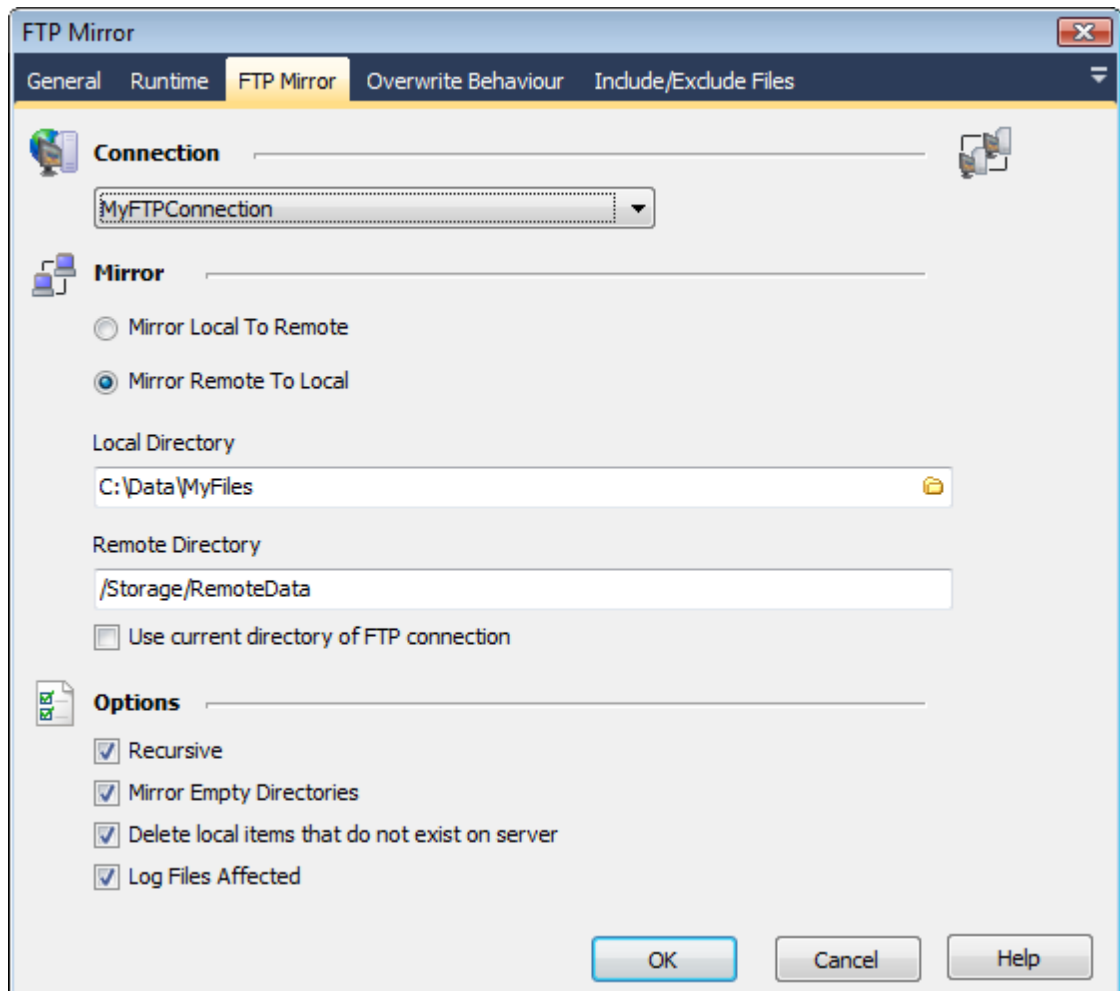
This action also allows you to specify whether files that exist in both source and destination should be overwritten in the destination based on the following rules:

- 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.



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.

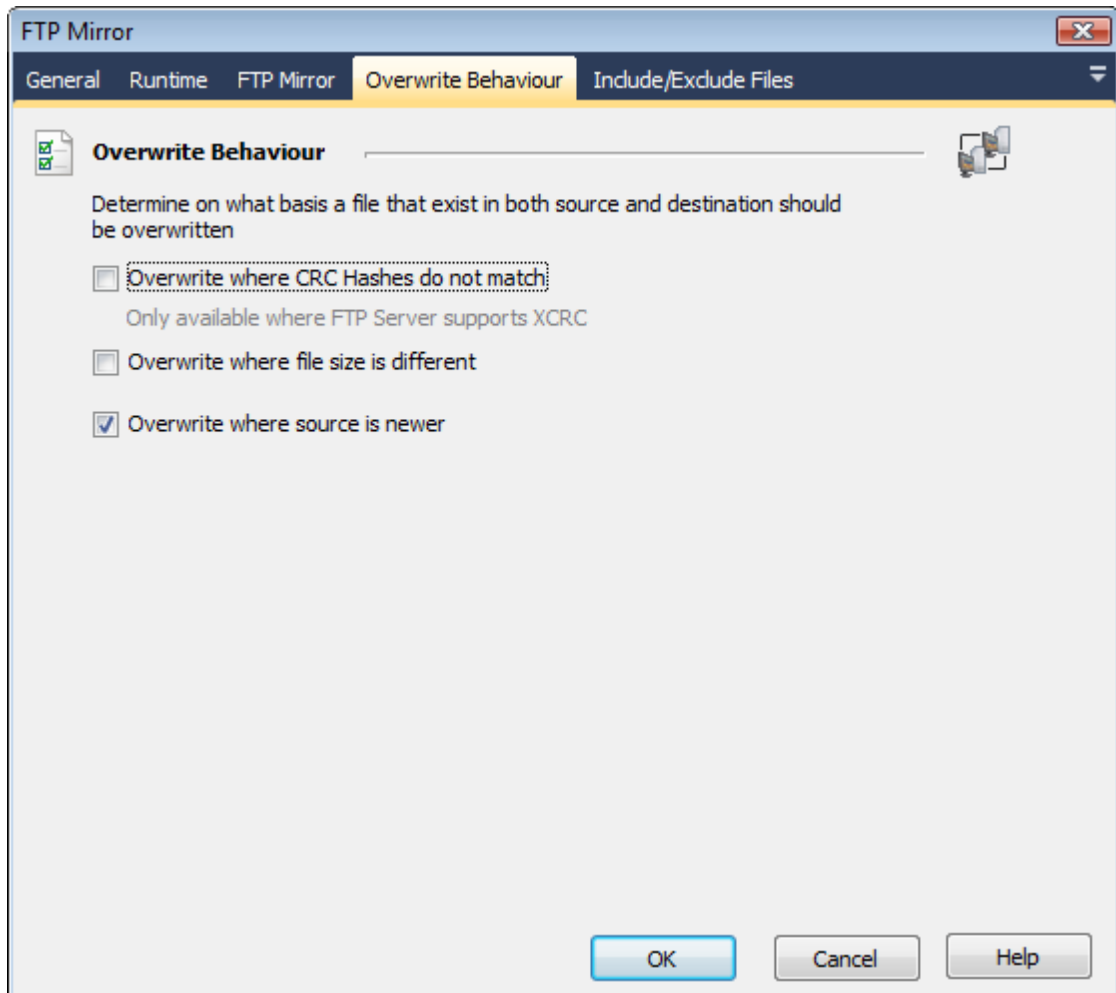
Mirror Empty Directories - Enabling this option allows directories that are empty on the source end to be recreated in the destination, by default empty directories are not re-created.

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.



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.

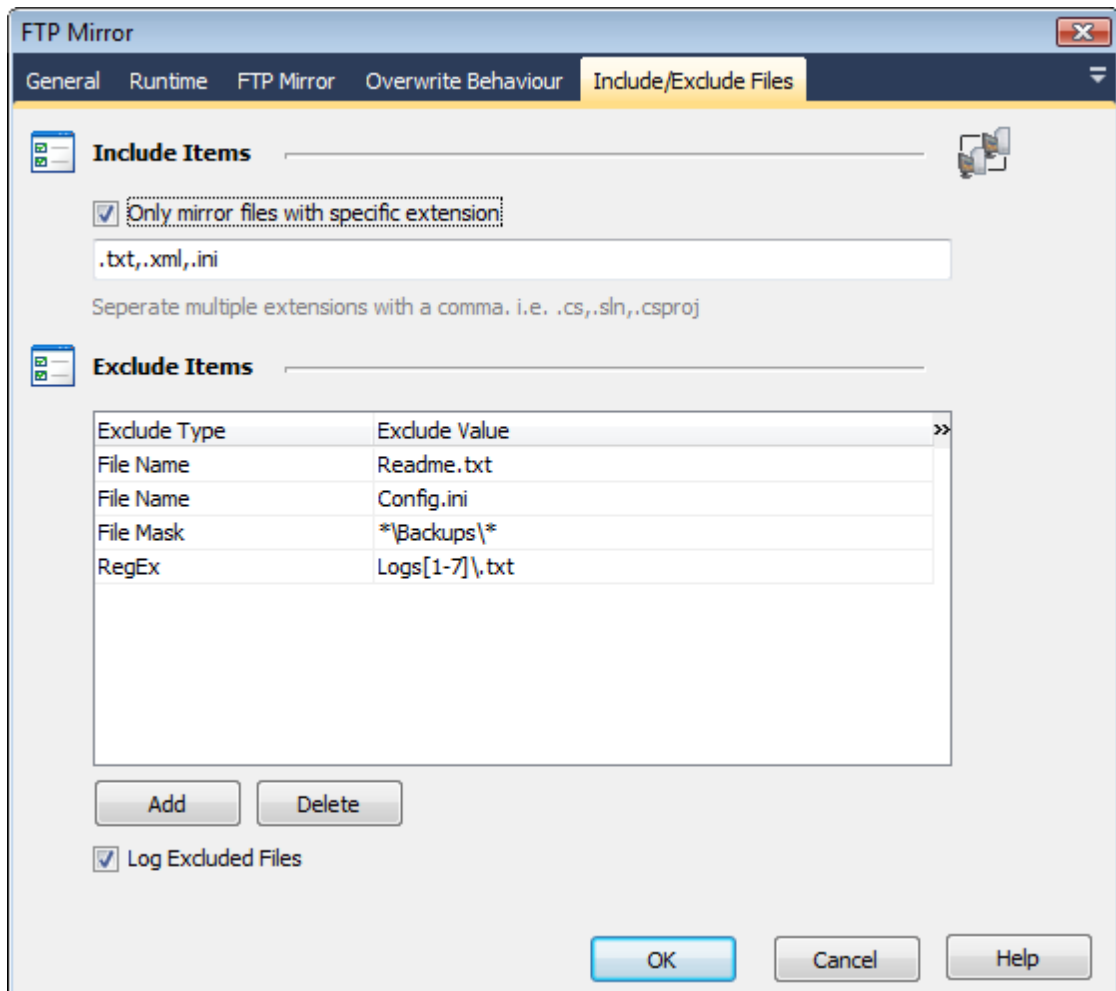
Overwrite where source is newer - Enable this option to compare the file date when the file

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.



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.

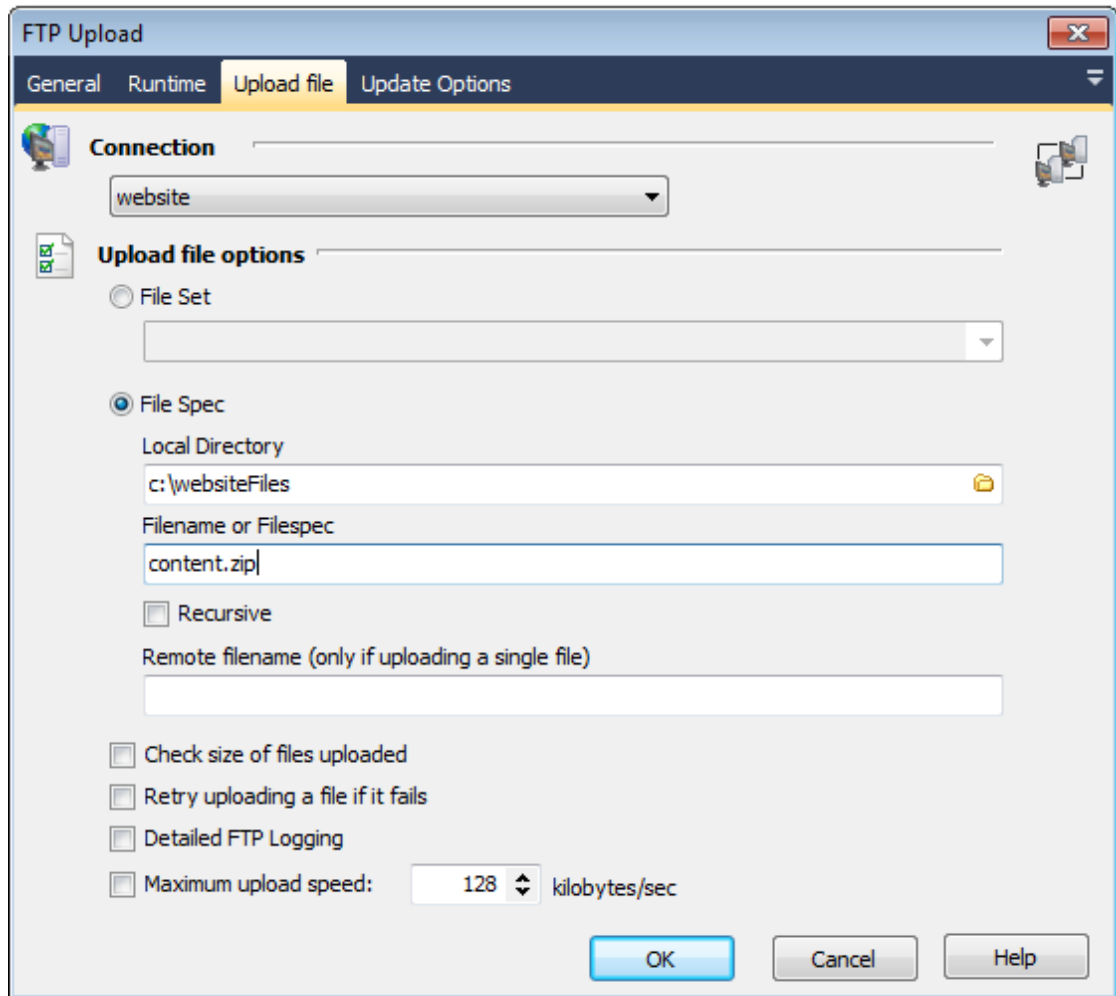
See the Regular Expression reference for more on using regular expressions.

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.

6.19.2.2 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 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

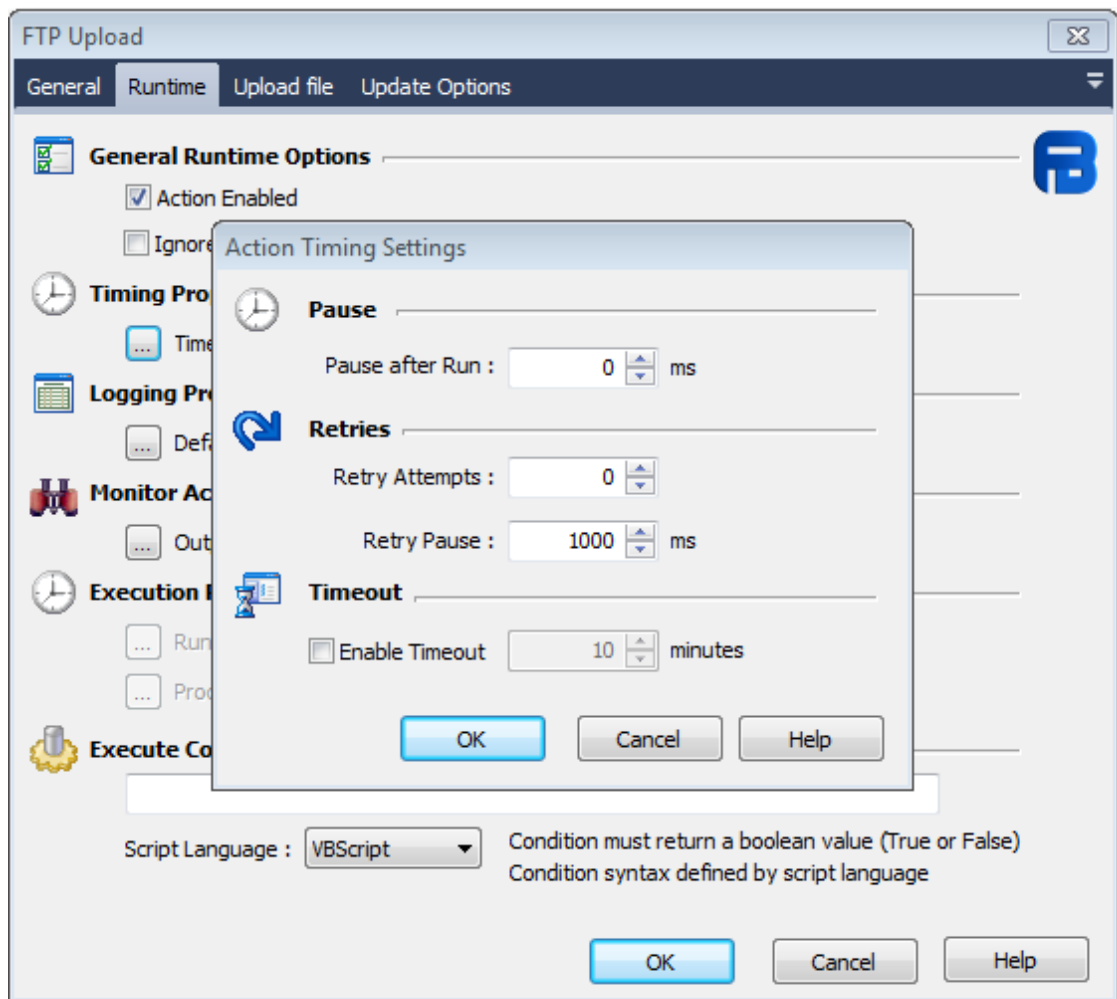
Outputs the raw FTP commands and log messages to the log.

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

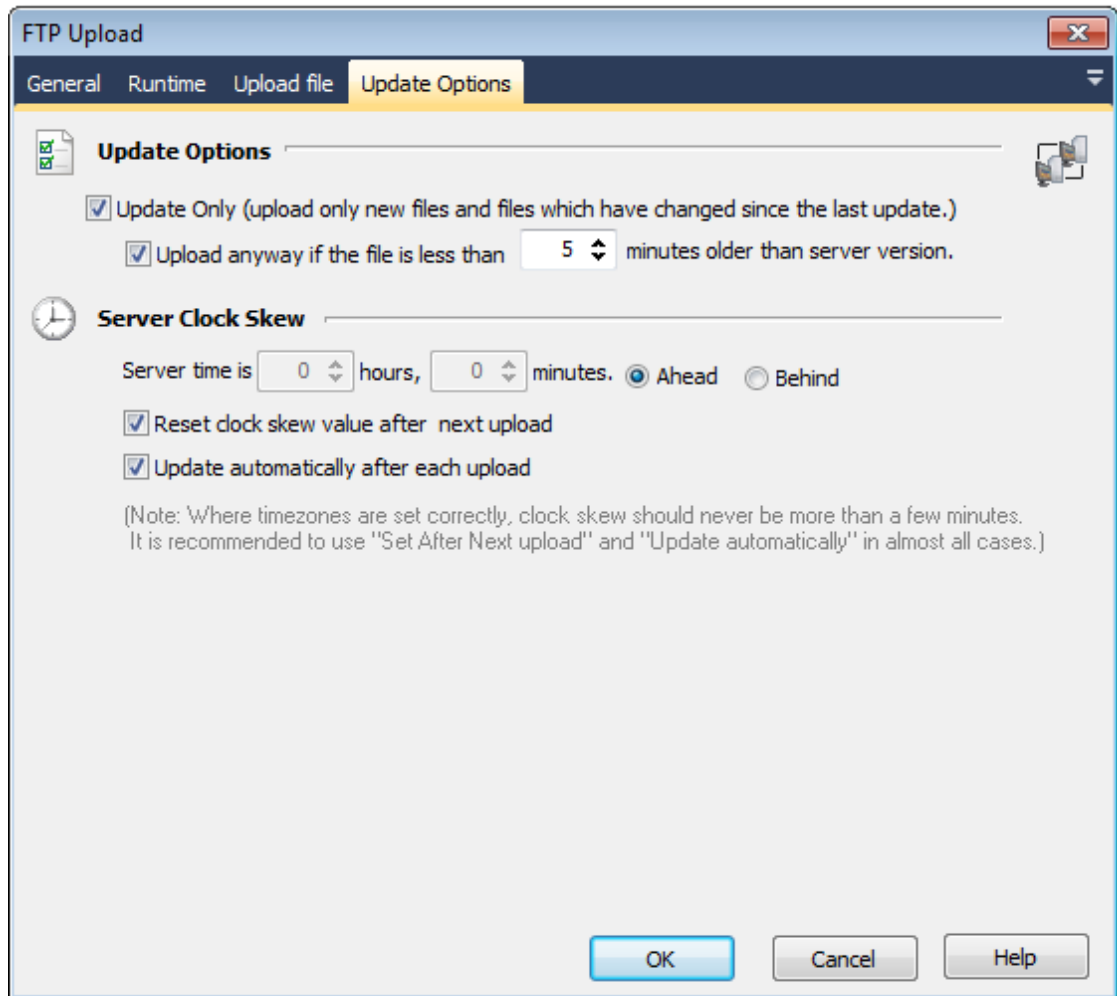
You can set a maximum time for the upload to complete in:



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 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

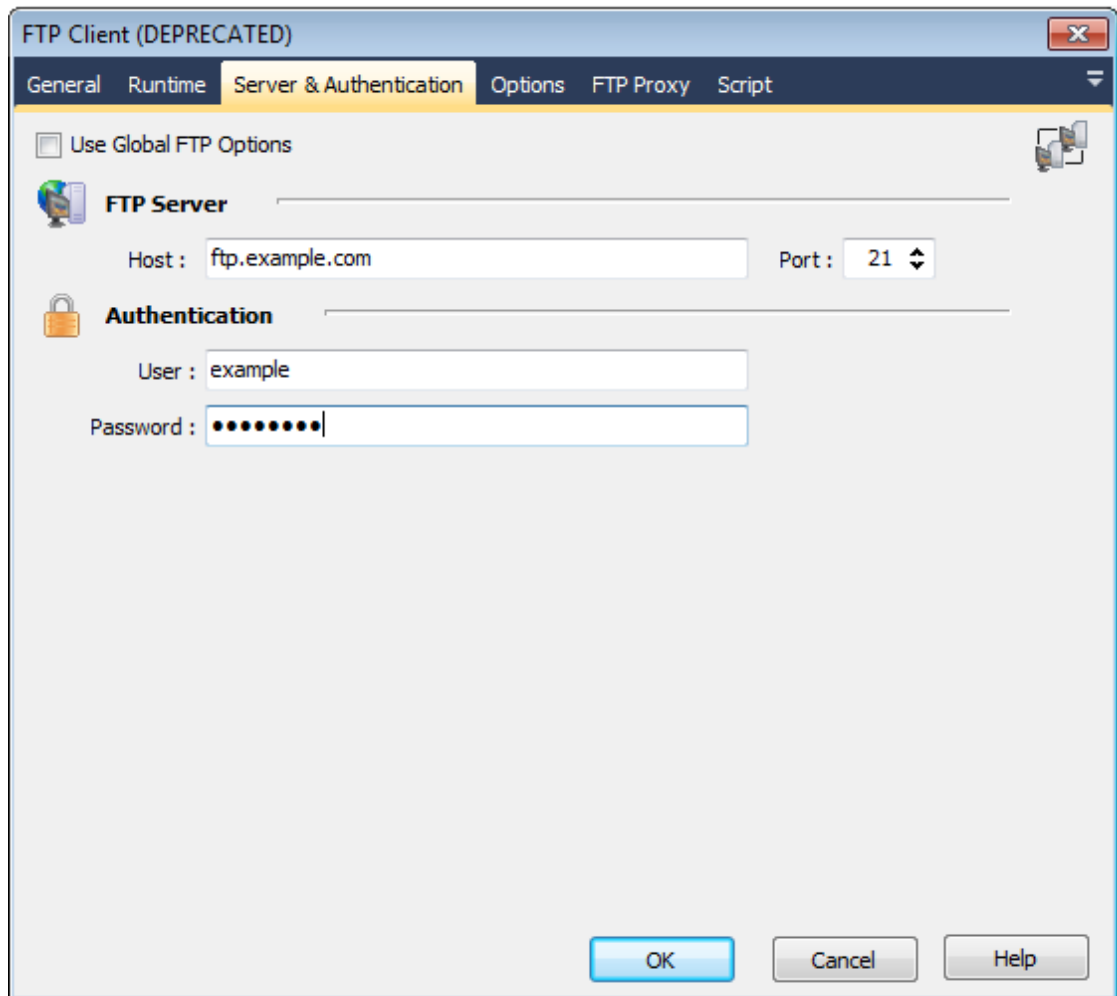
If this option is checked, then each time a file is updated its timestamp is used to reset the clock skew value if it has changed. This helps overcome "clock drift" over time.

6.19.3 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.



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

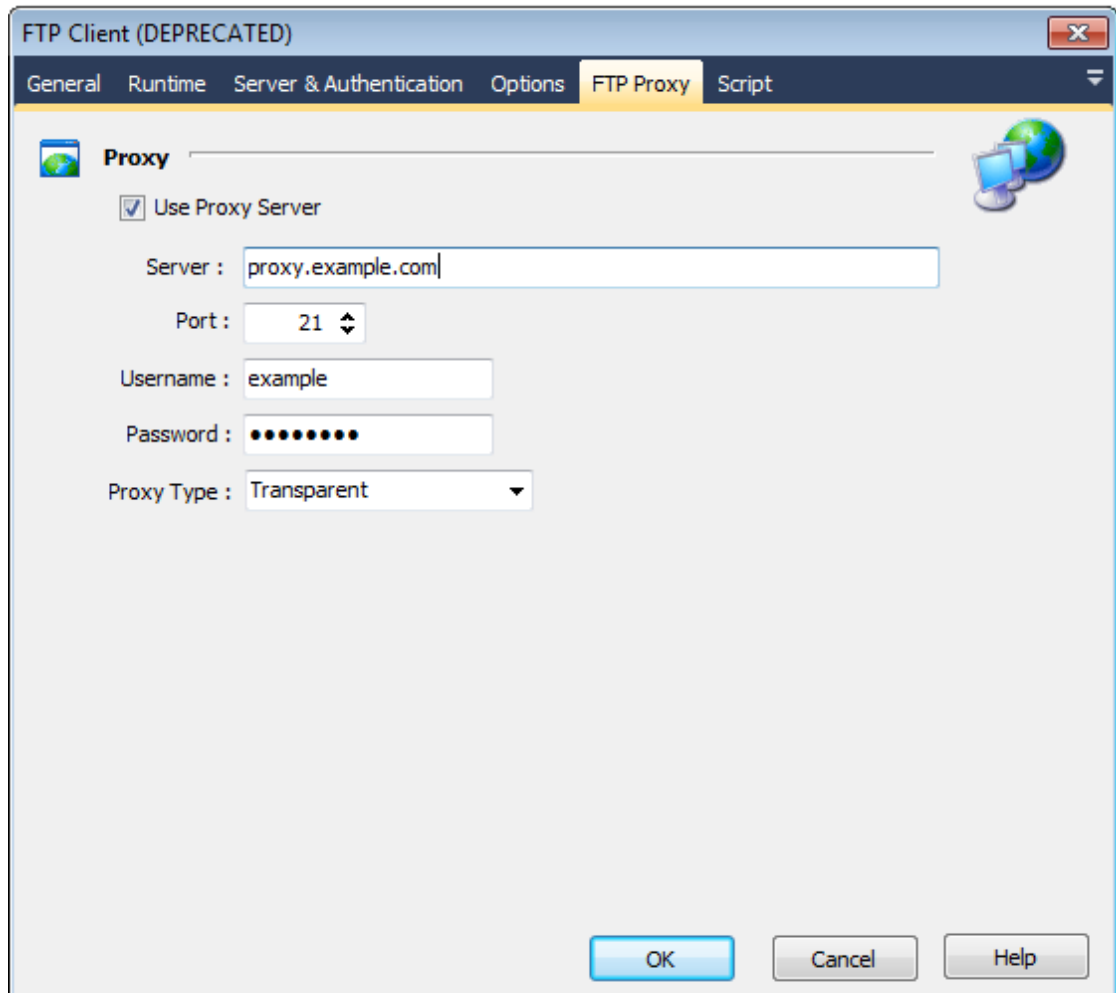
Passive : Instructs the FTP client and server to use passive mode transfers, this is needed when going through some firewalls

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



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

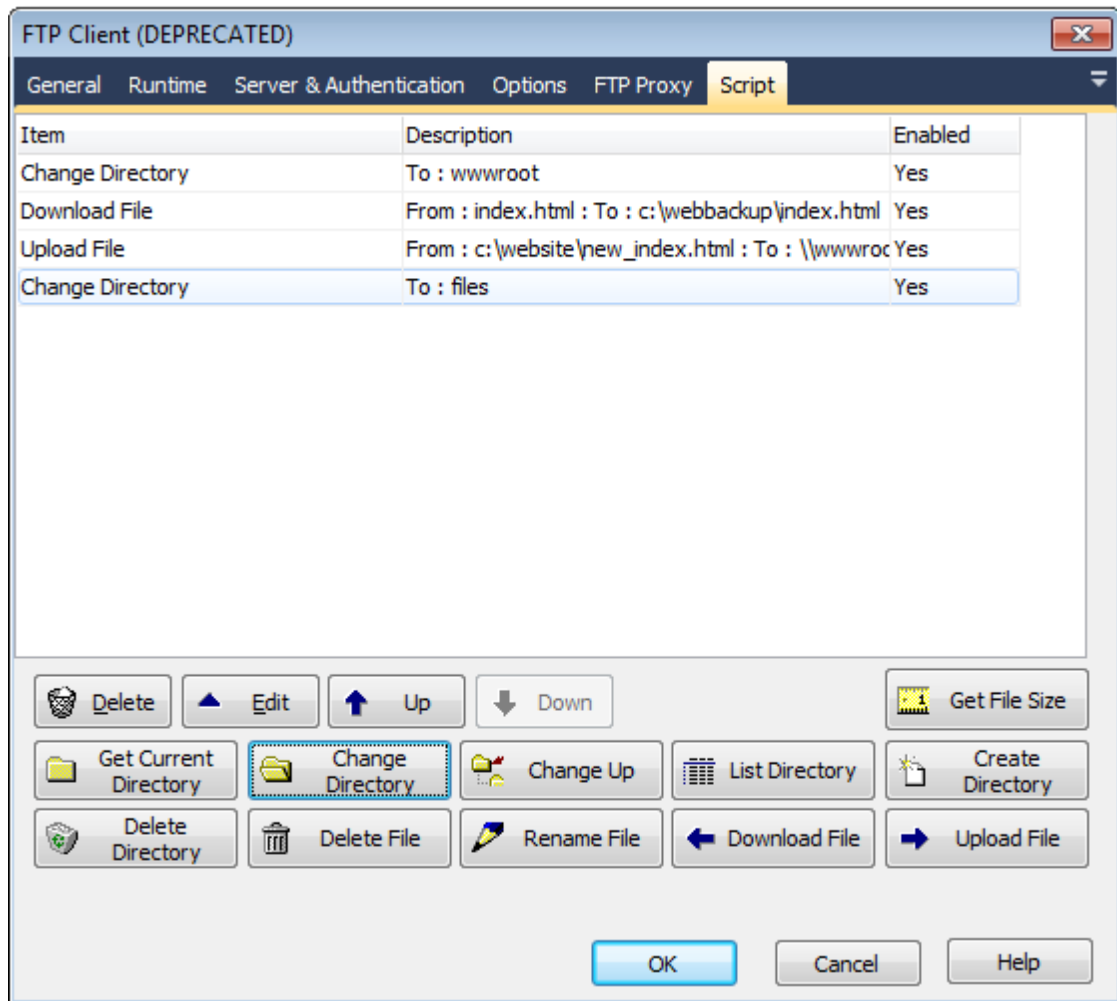
UserPass - USER user@firewalluser@hostname / PASS pass@firewallpass

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



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 FinalBuilder 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>

6.19.4 HTTP Get Action

This action enables you to download a file using the http protocol.

HTTP Get File

General Runtime **HTTP Get File** HTTP Proxy

Location

URL

Port ☐ Accept & Process Redirects (3xx Responses)

Security

Username

Password

Download to

☐ Just check that file exists

☒ Save to file

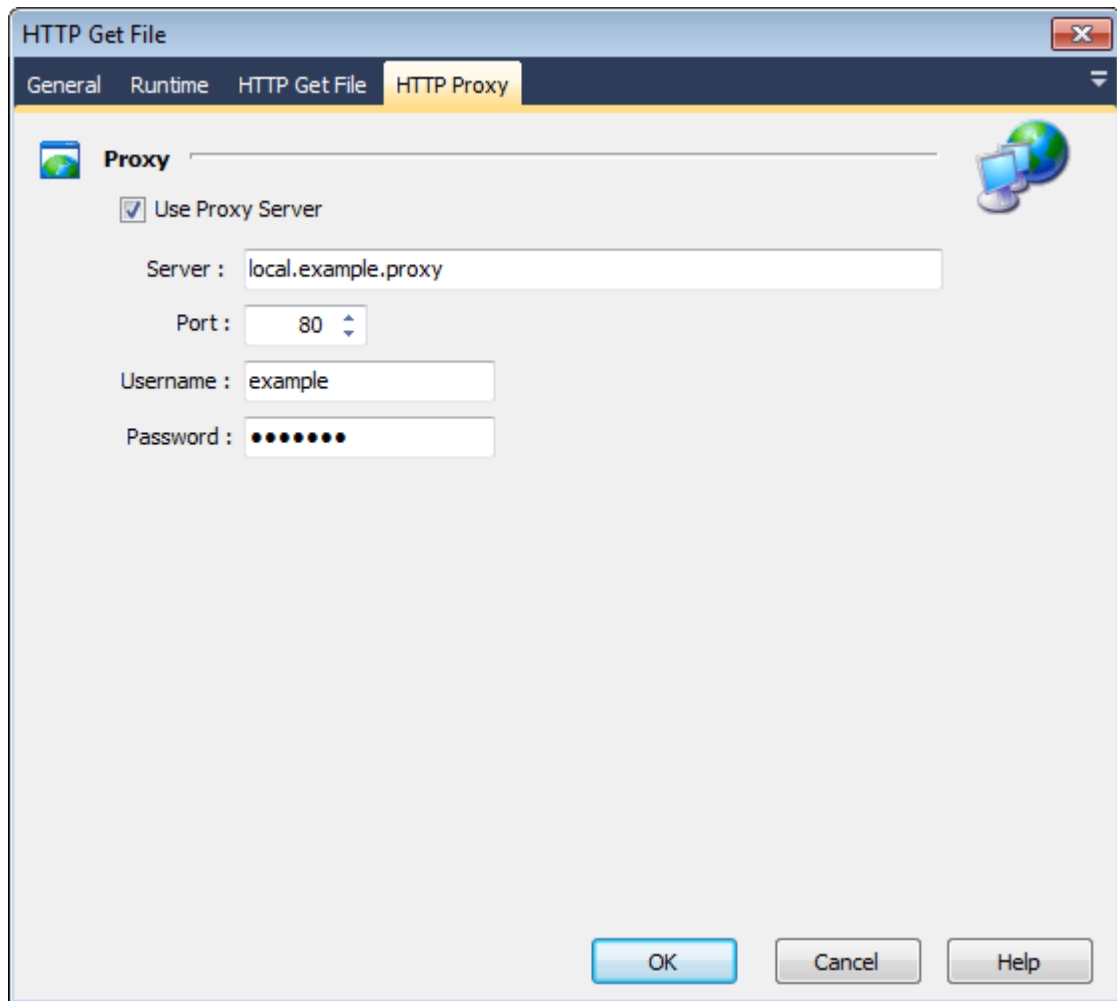
...

☐ Overwrite existing file

☐ Save to variable

If the variable does not exist, it will be created

OK Cancel Help



HTTP Get File

URL : The web address location of the file or resource to be downloaded or existence checked.

Port : Port of the server to communicate on. Typically 80 for http.

Accept & Process Redirects : Redirect responses (in the 300 range) will be treated as successes, and then processed to find the actual location of the resource.

Username : The username for any security challenge on the server holding the resource in question.

Password : The password required for the above username.

Just check that file exists : The resource isn't saved to a variable or file however the full GET response is received and checked that it is.

Save to file : Saves response to the specified file. Check the "Overwrite existing file" option to override the file each time.

Save to variable : Saves the response to a project variable for use later as needed.

Proxy

Use Proxy Server : Check to use a proxy for the HTTP GET requests.

Port : Port to use on the proxy server for HTTP GET requests.

Username : The username for any security challenge on the proxy server.

Password : The password required for the above username.

HTTP Responses & Action Success

Typically anything with a response less than 300 will be treated as a successful response. These are namely:

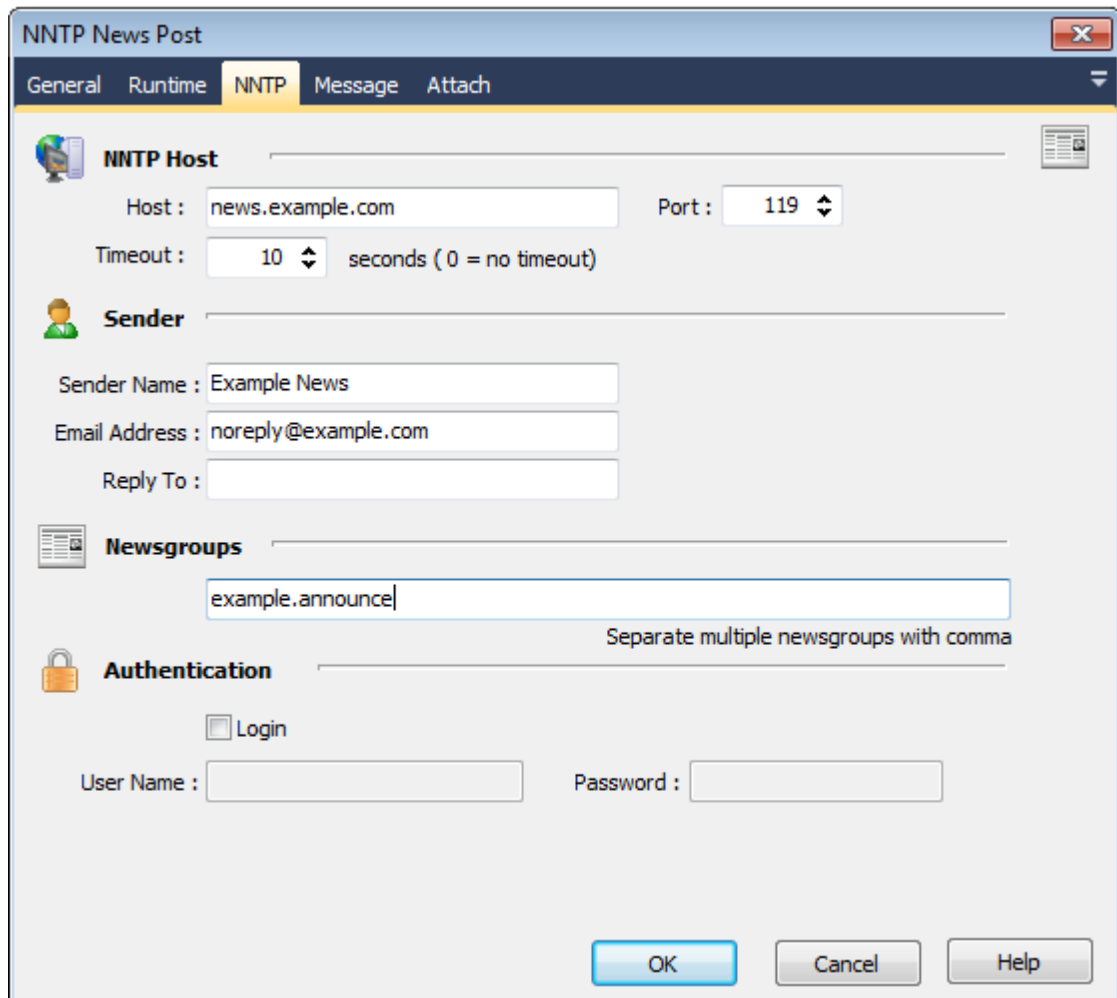
100	Continue
101	Switching Protocols
102	Processing (WebDAV)
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
207	Multi-Status (WebDAV)
208	Already Reported (WebDAV)
226	IM Used (RFC 3229)

Any other responses will be treated as errors. If the handling of redirects is turned on then all 300 range (300..308) response codes will be handled internally to redirect the HTTP GET Request to the new location.

6.19.5 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.

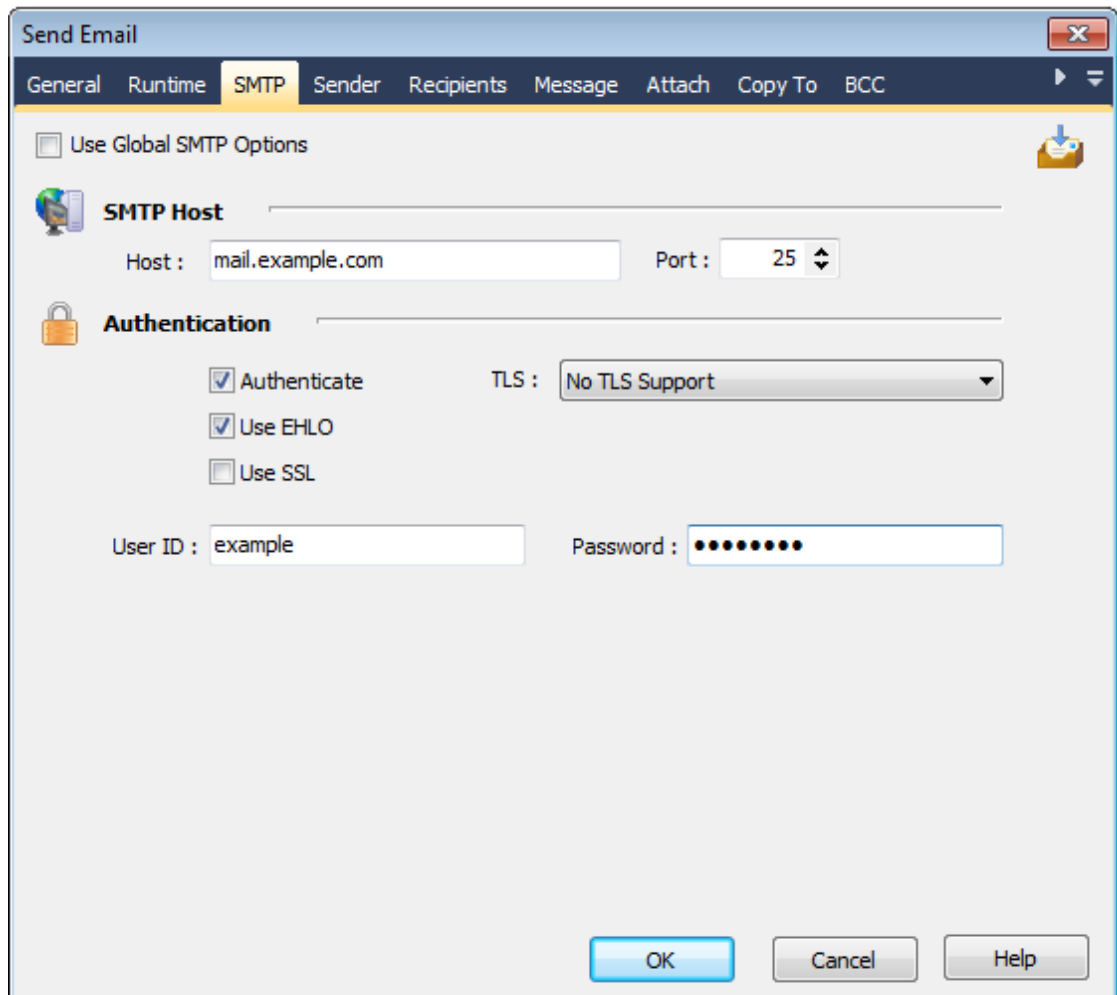


The screenshot shows the 'NNTP News Post' dialog box with the following fields and options:

- General** tab selected.
- NNTP Host** section:
 - Host:
 - Port:
 - Timeout: seconds (0 = no timeout)
- Sender** section:
 - Sender Name:
 - Email Address:
 - Reply To:
- Newsgroups** section:
 -
 - Separate multiple newsgroups with comma
- Authentication** section:
 - ☐ Login
 - User Name:
 - Password:
- Buttons: OK, Cancel, Help

6.19.6 Send Email (SMTP)

This action allows your FinalBuilder projects to send email via an SMTP server.



The image shows a 'Send Email' dialog box with a tabbed interface. The 'SMTP' tab is selected. At the top, there is a checkbox for 'Use Global SMTP Options'. Below this, the 'SMTP Host' section includes a 'Host' text field with 'mail.example.com' and a 'Port' spinner box set to '25'. The 'Authentication' section, marked with a lock icon, contains three checkboxes: 'Authenticate' (checked), 'Use EHLO' (checked), and 'Use SSL' (unchecked). To the right of these is a 'TLS' dropdown menu showing 'No TLS Support'. At the bottom of the authentication section are 'User ID' and 'Password' text fields; 'User ID' contains 'example' and 'Password' is masked with dots. At the very bottom are 'OK', 'Cancel', and 'Help' buttons.

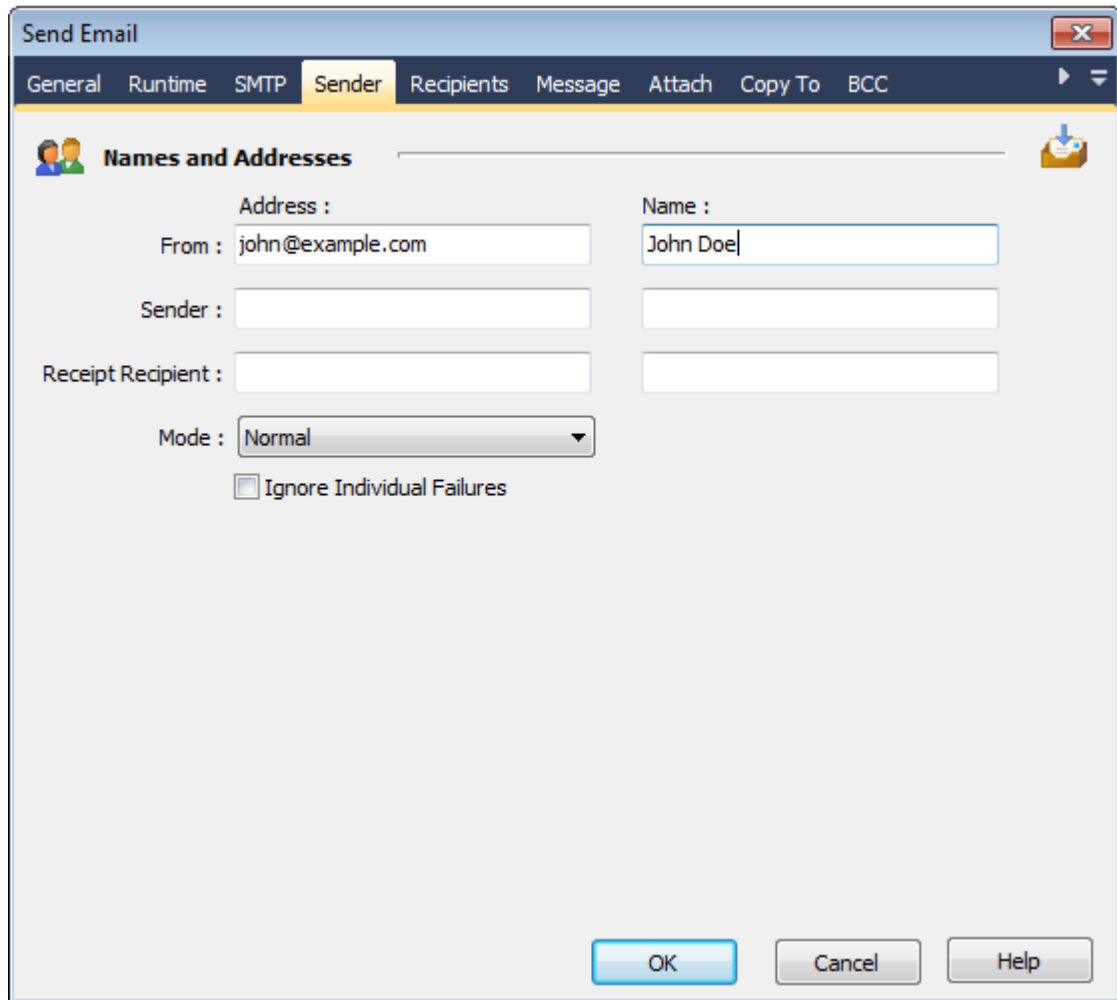
Host : The host name (or IP address) of the SMTP server

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 FinalBuilder), and then the client can adjust it's protocol to support the reported capabilities.

It is generally a good idea to enable this option.



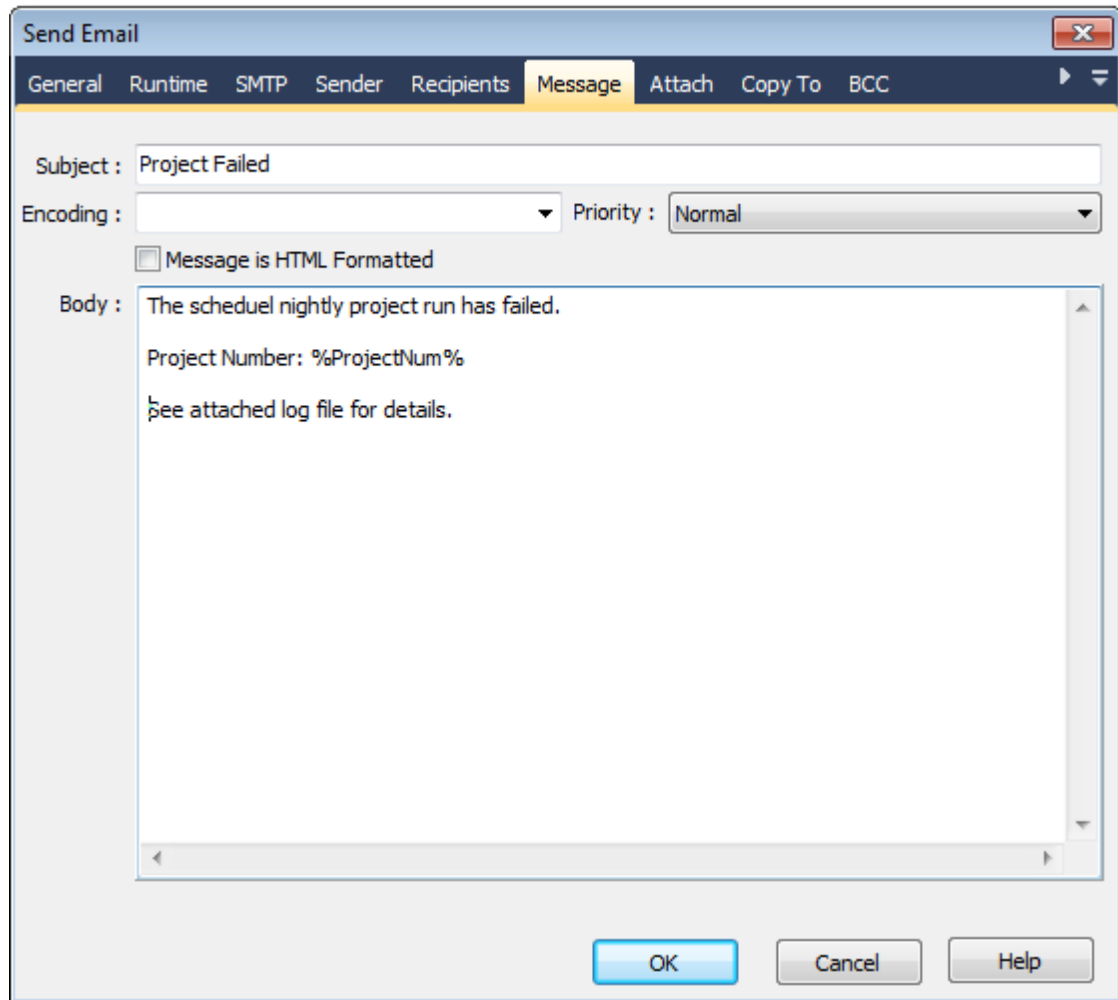
The image shows a 'Send Email' dialog box with a tabbed interface. The 'Sender' tab is selected. The dialog has a title bar with a close button. Below the tabs, there is a section titled 'Names and Addresses' with a small icon of two people. The main area contains several input fields: 'From' (with 'john@example.com' entered), 'Name' (with 'John Doe' entered), 'Sender', 'Receipt Recipient', and 'Mode' (a dropdown menu set to 'Normal'). There is also an unchecked checkbox labeled 'Ignore Individual Failures'. At the bottom right, there are three buttons: 'OK', 'Cancel', and 'Help'.

From : This must be set to a valid email address

Sender : Set this when sending mail on behalf of someone else

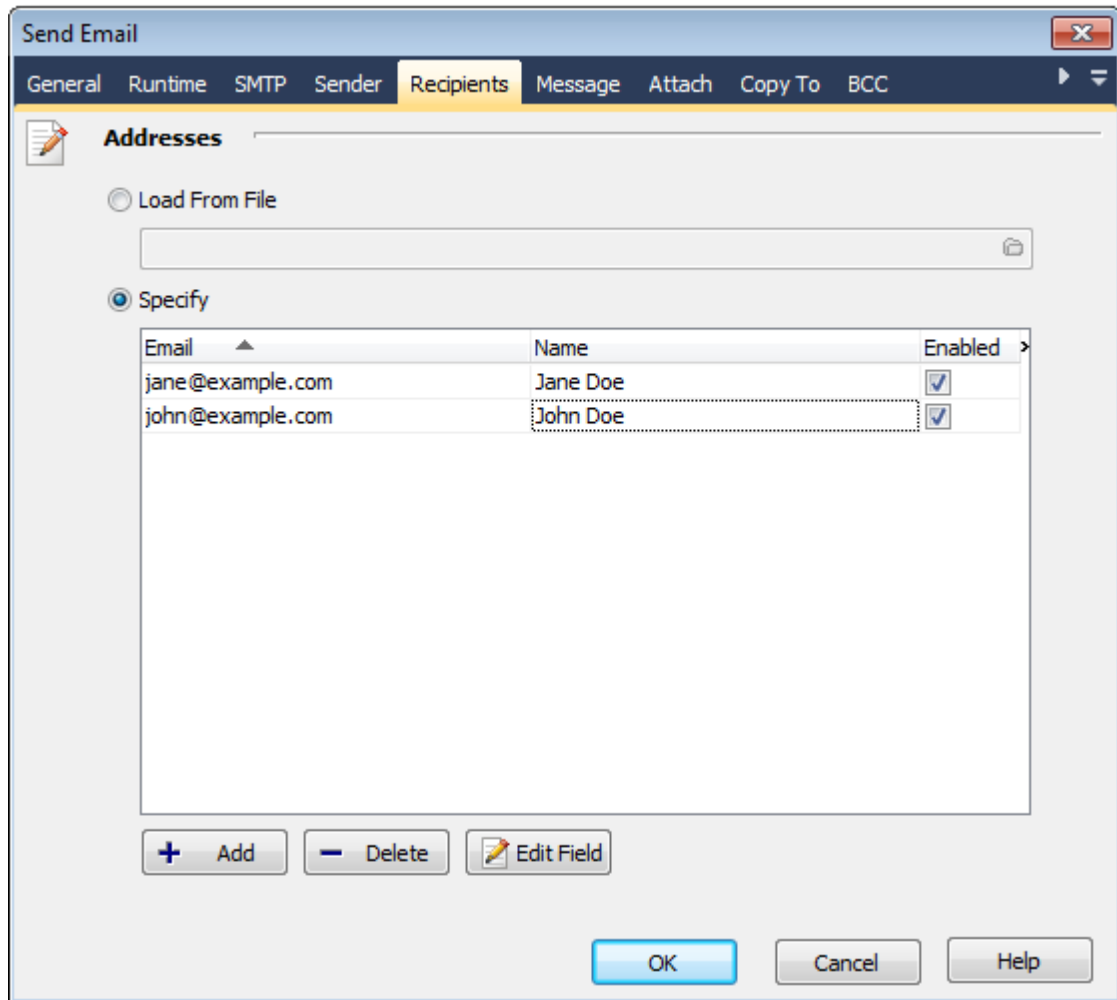
Receipt Recipient : If you require a notice receipt then set the address for the receipt message to be sent to

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.



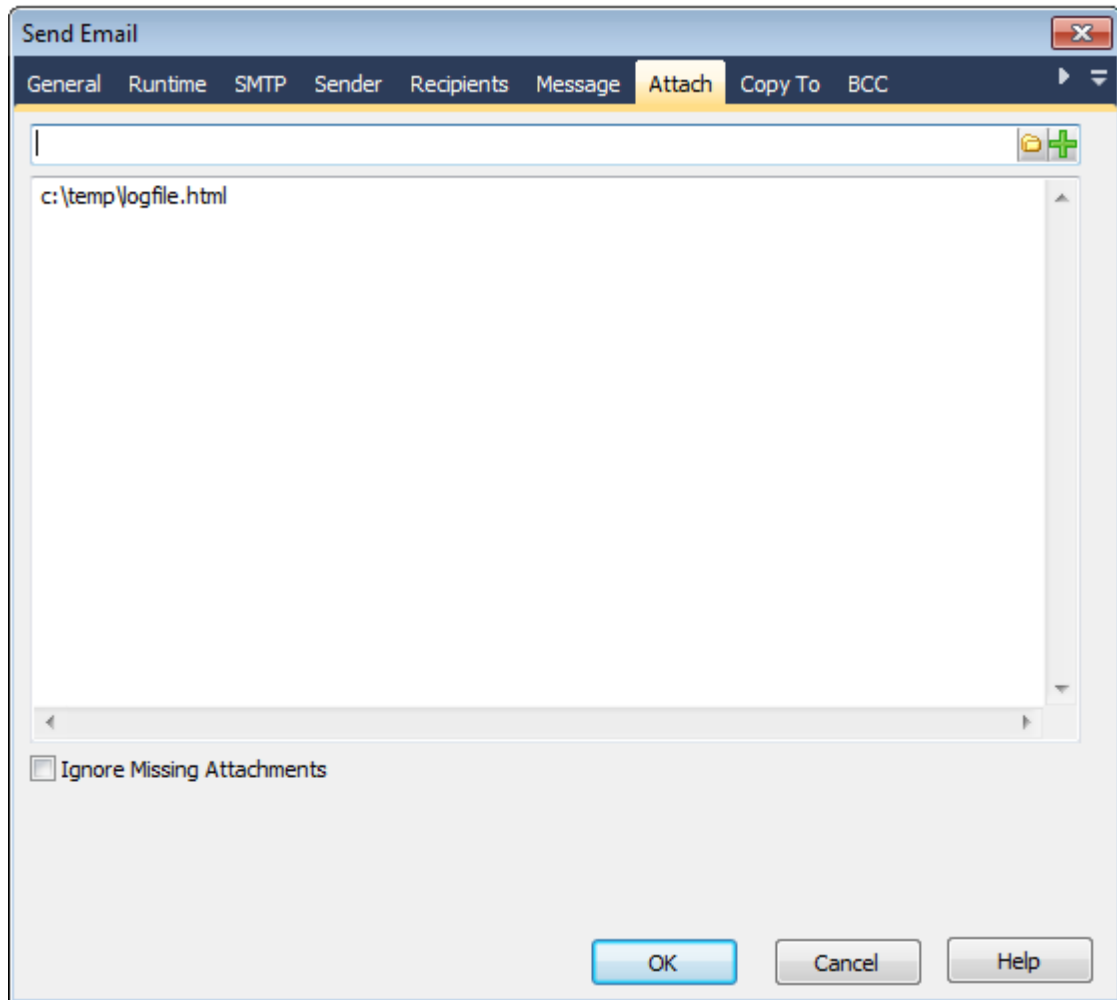
Subject : The email message's subject

Body : The actual message.



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.



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

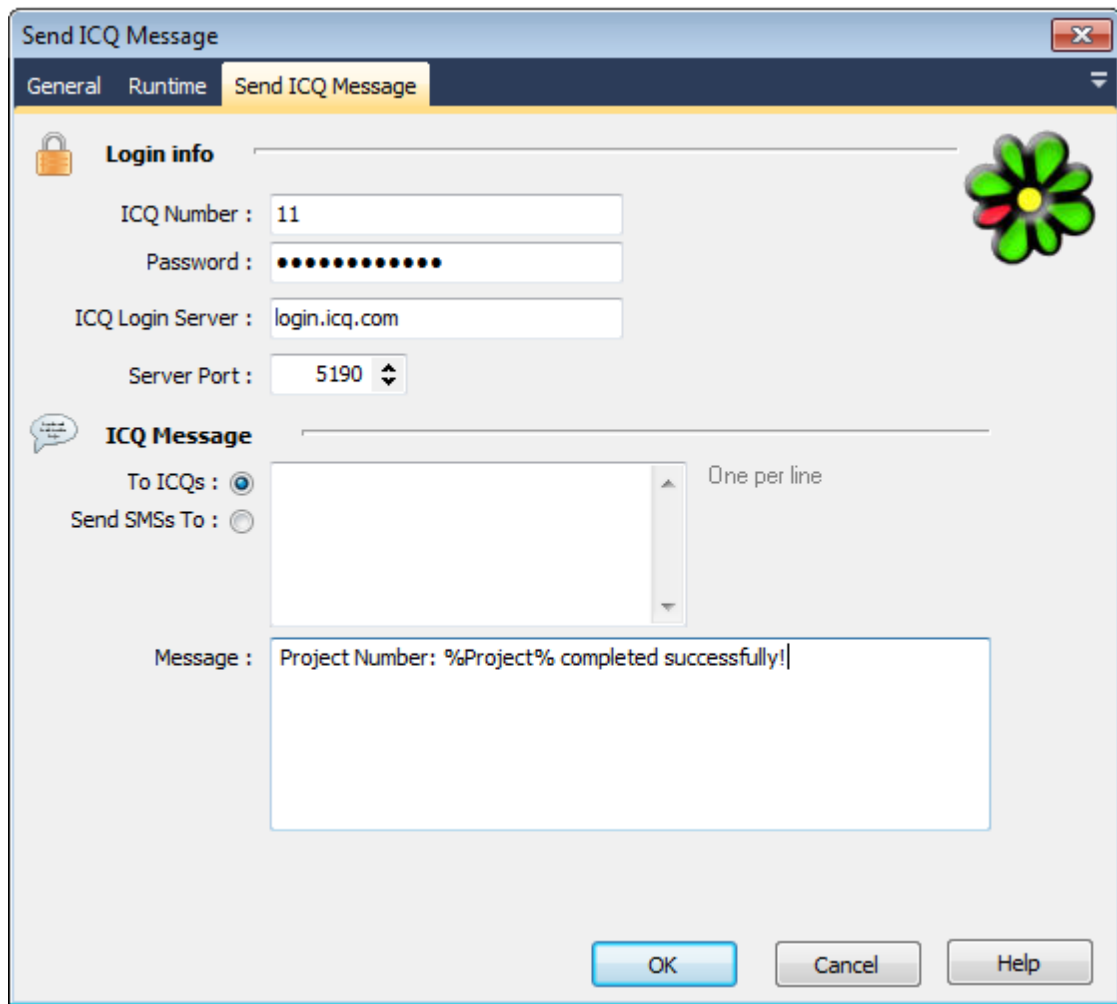
property Address : WideString

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>

6.19.7 Send ICQ Message Action

This action enables you to send ICQ messages.

The image shows a 'Send ICQ Message' dialog box with a blue title bar and a close button. It has three tabs: 'General', 'Runtime', and 'Send ICQ Message', with the last one being active. The 'Login info' section, marked with a padlock icon, contains fields for 'ICQ Number' (11), 'Password' (masked with dots), 'ICQ Login Server' (login.icq.com), and 'Server Port' (5190). The 'ICQ Message' section, marked with a speech bubble icon, has radio buttons for 'To ICQs' (selected) and 'Send SMSs To'. Below these is a list box with a vertical scrollbar and the text 'One per line'. At the bottom, there is a 'Message' text area containing the text 'Project Number: %Project% completed successfully!'. The dialog box also features 'OK', 'Cancel', and 'Help' buttons at the bottom right.

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 or 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.

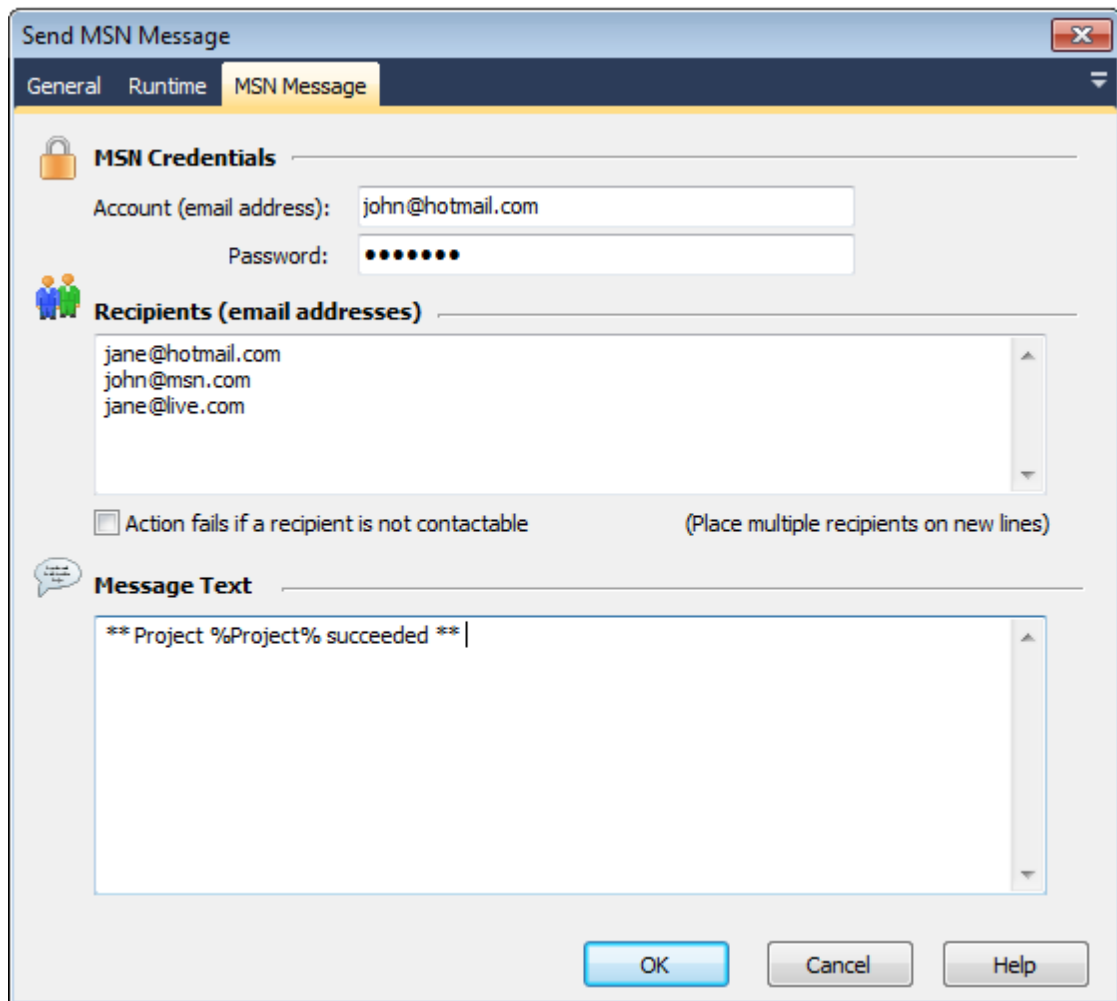
Message

The text of the ICQ message.

6.19.8 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 build.

The MSN action uses the open source [dotMSN library](#) to access MSN Messenger (see below for copyright details.)



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 FinalBuilder.

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@FinalBuilder.com does not need vincent@FinalBuilder.com on his contact list, it is recommended that vincent@FinalBuilder.com adds angus@FinalBuilder.com to his contact list.)

Action fails if a recipient is not contactable

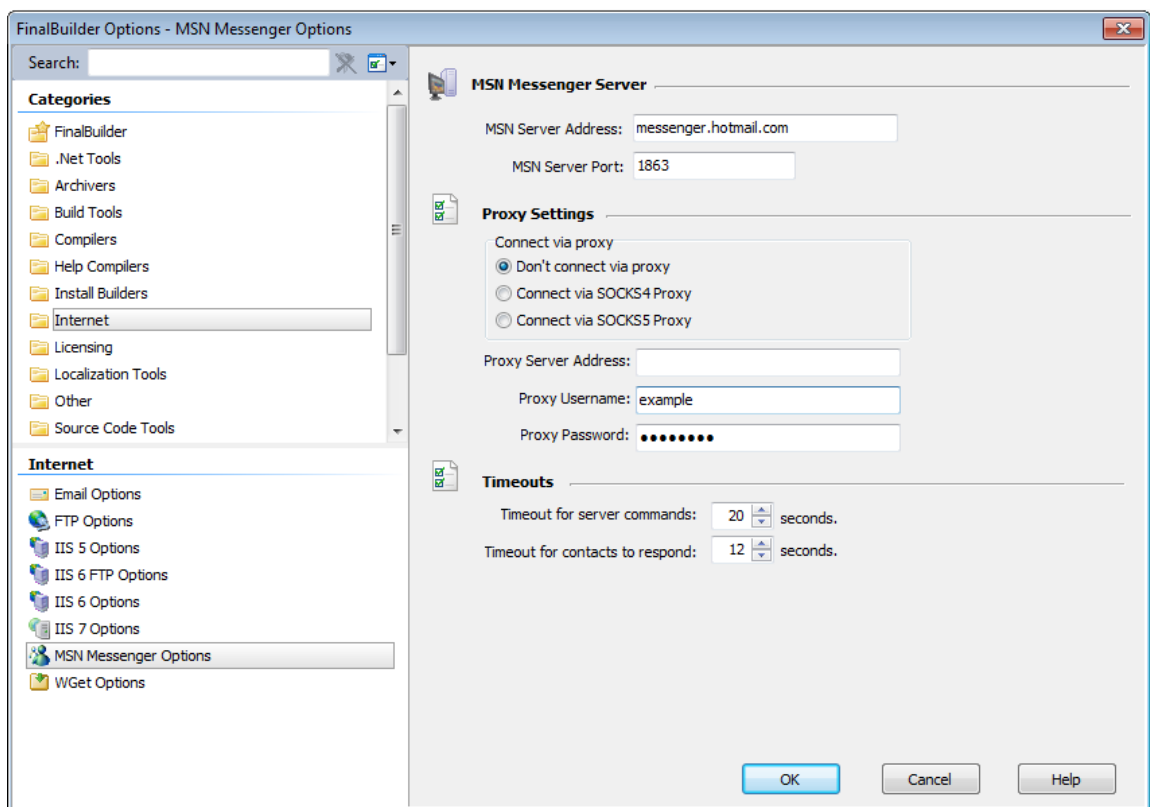
The action can be set to fail if one or more contacts could not be reached (either because they are offline or because that email address is not registered with MSN.)

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



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 FinalBuilder 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:
```

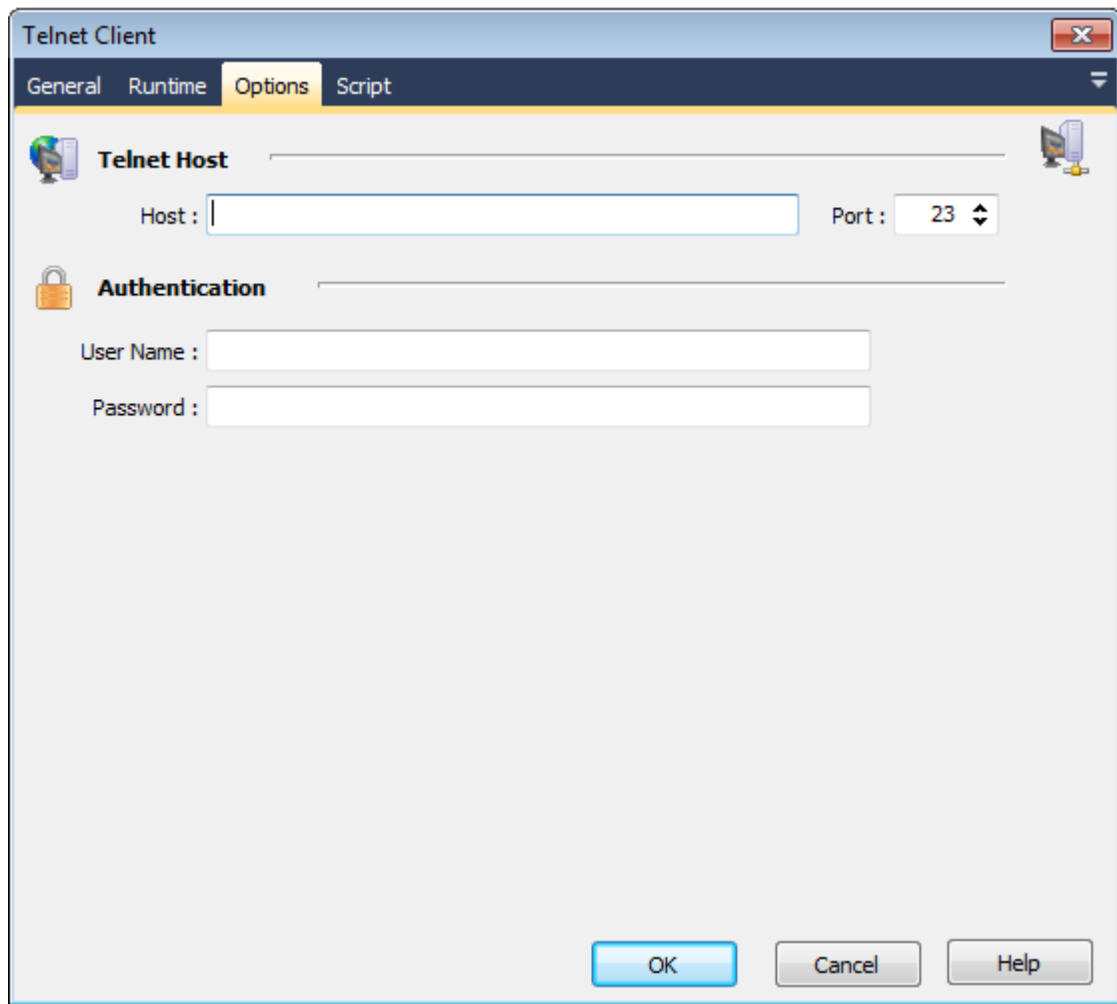
- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials 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
```

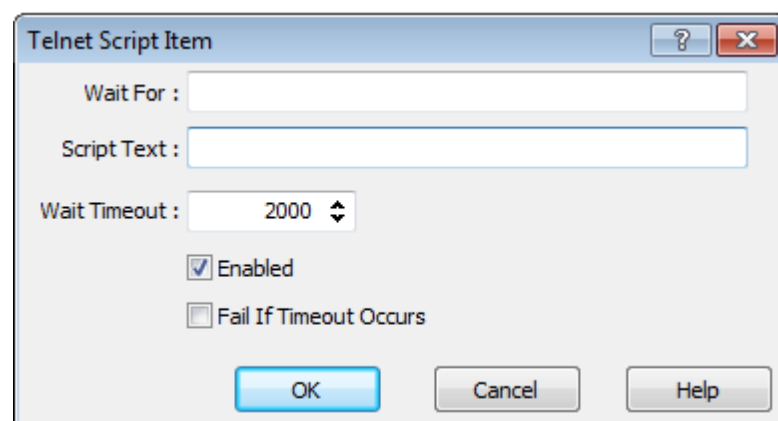
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6.19.9 Telnet Client Action

This action provides a simple scriptable Telnet client.



The script items allow you to wait for a string and (optionally) respond with a string.



To send the UserName set the ScriptText field to @USERNAME

To send the Password set the ScriptText field to @PASSWORD

6.19.10 Web Service Action

The Web Service Action allows you to call a method of a web service and read results into your running project.

Web Service

General Runtime **Service** Output

Web Service

Service Description Url
es.daelab.net/temperatureconversion/TemperatureConversion.wso?WSDL Load
The Url to the web service description (Example: http://mydomain.com/myService.asmx?wsdl)

Domain Username Password

Method
WindChillInCelcius

Parameters

nCelcius %Celcius%
nWindSpeed 18|

☐ Fixed Parameter Names ☒ Editable Parameter Names Reload Parameters

OK Cancel Help

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.)

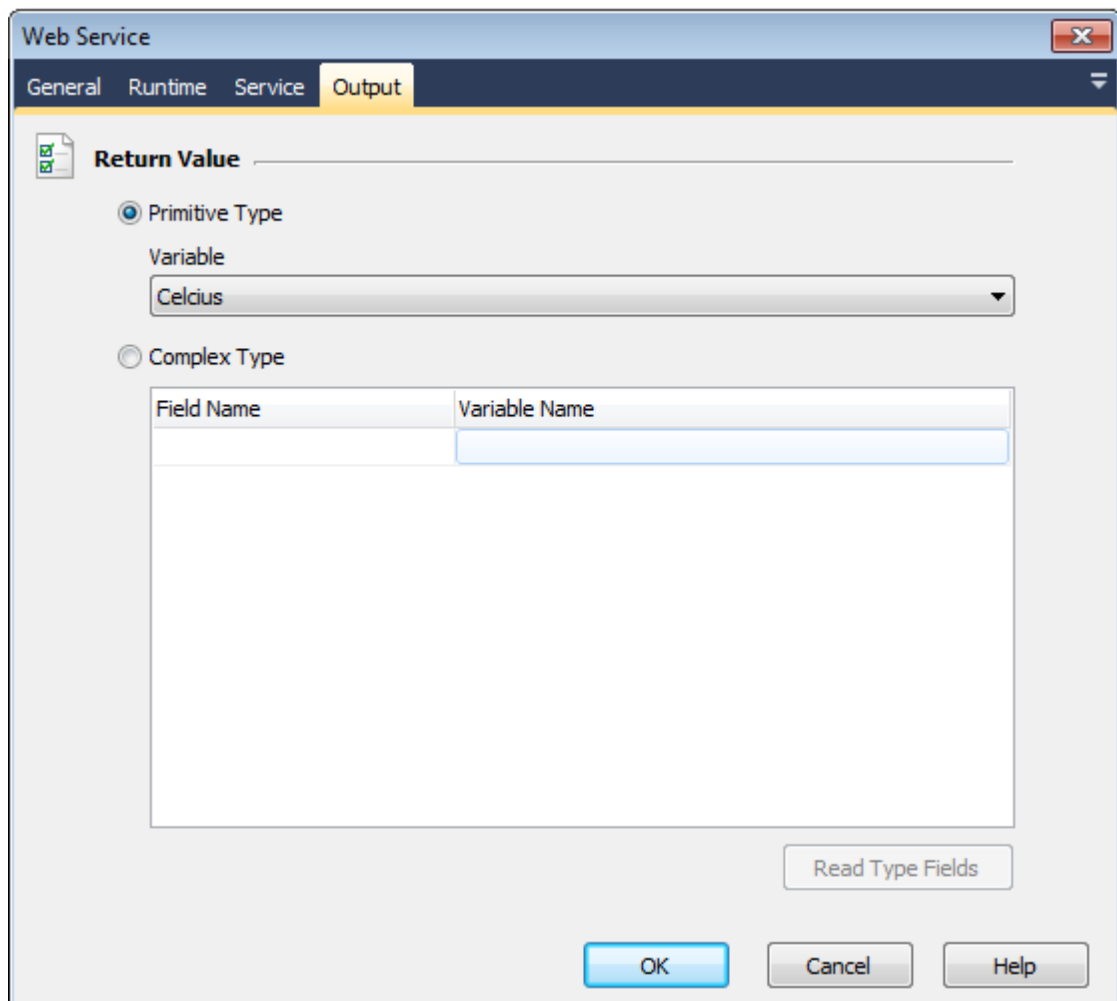
This dropdown is automatically populated when you click the Load button.

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.



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.

6.19.11 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

The screenshot shows the 'Web Service Iterator' dialog box with the 'Output' tab selected. The 'Return Value' section has two radio buttons: 'Array of Primitive Types' (unselected) and 'Array of Complex Types' (selected). Under 'Array of Primitive Types', there is a 'Variable' dropdown menu with 'Celcius' selected. Under 'Array of Complex Types', there is a table with two columns: 'Field Name' and 'Variable Name'. The table is currently empty. At the bottom right of the table area is a 'Read Type Fields' button. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

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.

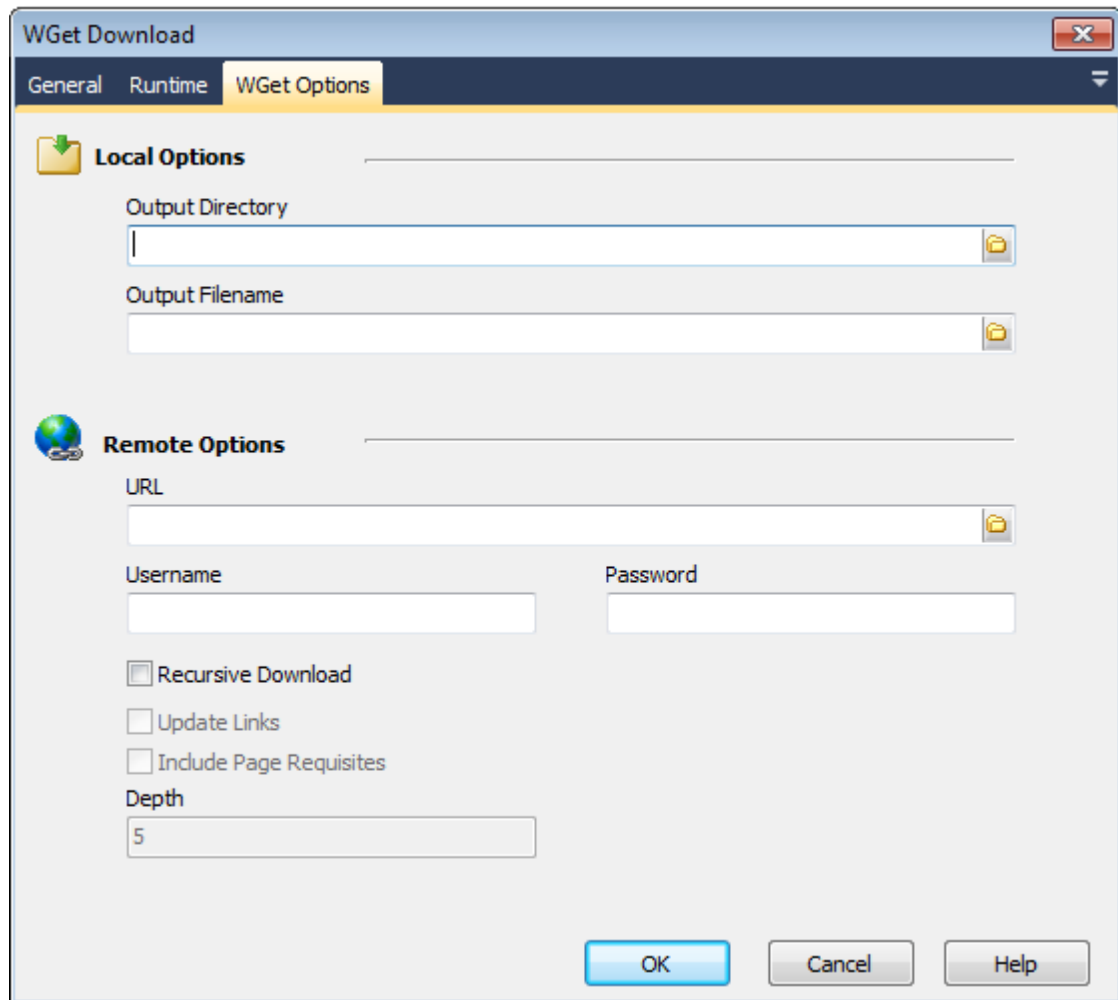
If the method returns an array of complex types (as shown above), then the iterator will iterate over the values of the array and set each variable to the corresponding field of the complex type. Click on "Read Type Fields" to populate the list of available fields for the method. You do not need to specify a variable for each field, just the fields which you need to read.

6.19.12 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

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;

6.19.13 MSDeploy

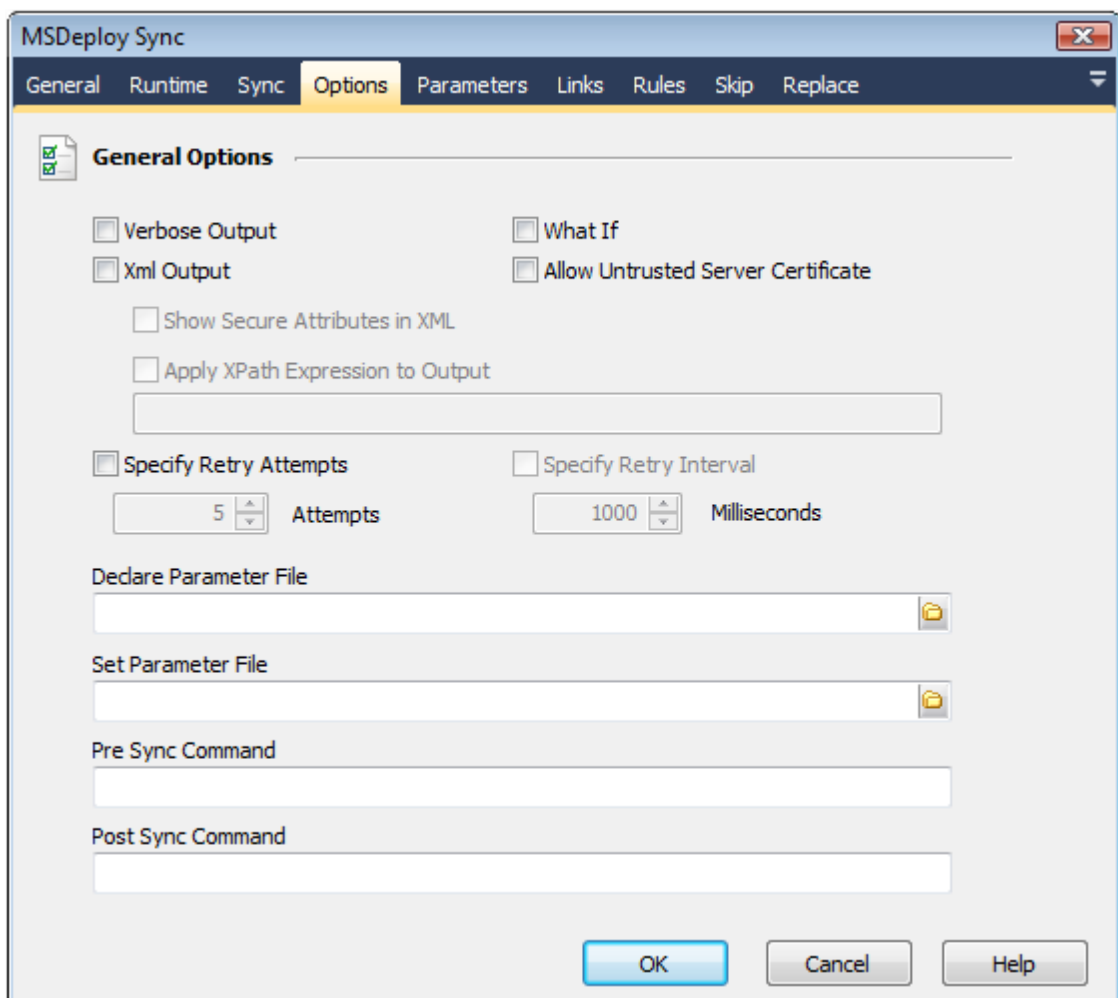
The MSDeploy actions allow you to manage IIS Web Applications, Web Sites, Web Server Content and configuration as a part of your FinalBuilder project.

The MSDeploy actions consist of five individual actions:

- MSDeploy Sync
- MSDeploy Delete
- MSDeploy Dump
- MSDeploy Get Dependencies
- MSDeploy Get System Info

Before using any of these actions you must have the Web Deploy Tool installed on your machine. Specify the path to msdeploy.exe via Tools -> Options -> Internet -> MSDeploy.

All the actions include a common **Options** page, which allow to you set common options for the actions.



The **Xml Output** option cannot be disabled for the Get Dependencies and Get System Info actions as they automatically return output in Xml Format. The Xml Output argument will not be passed to the command line for these actions.

For more information on these options see Web Deploy Operation Settings.

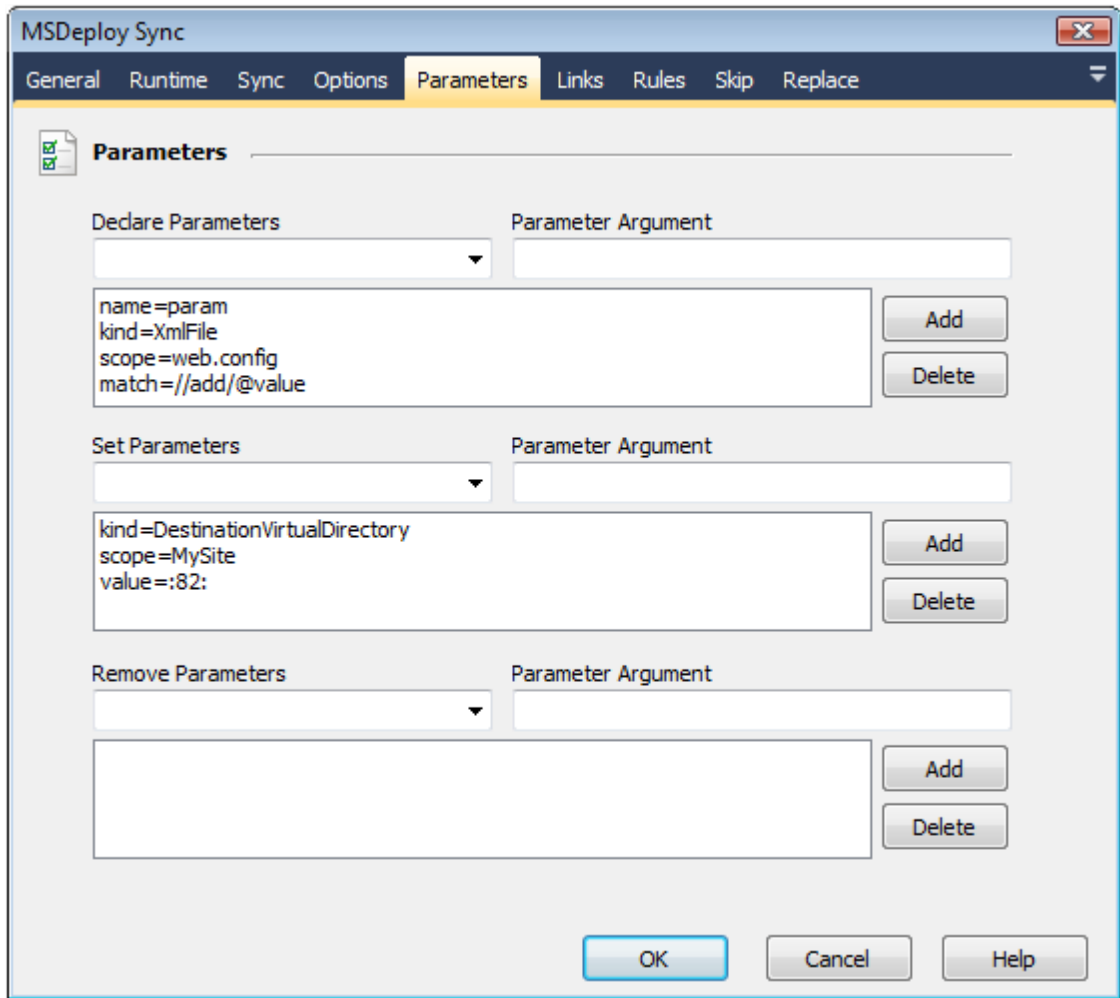
6.19.13.1 MSDeploy Sync

The **MSDeploy Sync** action allows you to synchronise one Web Deploy Provider object (destination) with another (source). It also allows you to package (backup) existing IIS components into a zip archive. You can use these packages to deploy content to other machines at a later stage. The Sync action also allows you to deploy content and configurations from one location to another.

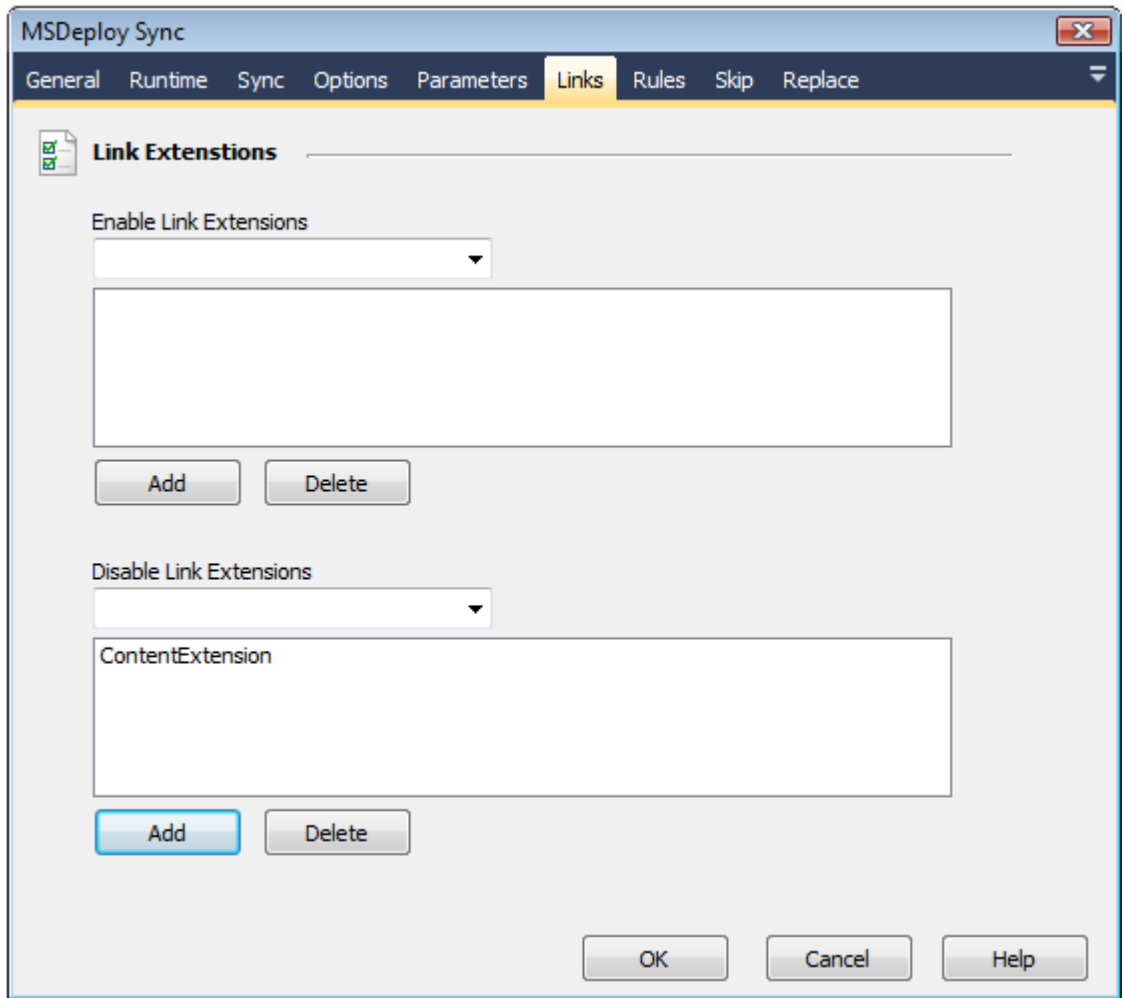
The screenshot shows the MSDeploy Sync dialog box. The 'Sync' tab is active. The 'Source Object' is set to 'webServer'. The 'Destination Object' is also set to 'webServer'. The 'Source Provider Setting' and 'Destination Provider Setting' are both empty. The 'Source Object Value' and 'Destination Object Value' are also empty. There are two text boxes for 'computerName=Server2' and 'computerName=Server3', each with 'Add' and 'Delete' buttons. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

To perform a Sync using a remote server, the *Web Deploy Remote Service* needs to be installed and running on the remote machine. See here for more information on the service [Web Deploy Remote Service](#).

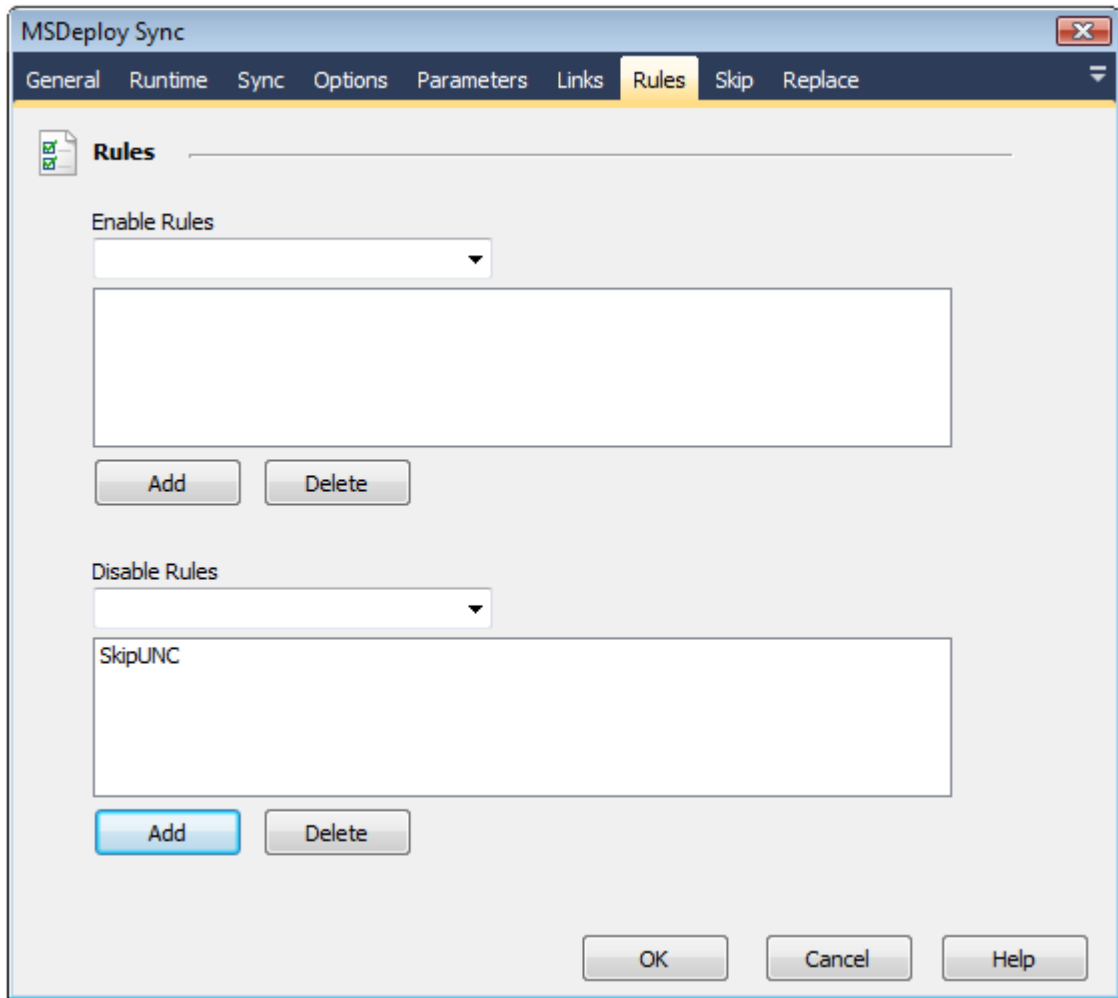
The **Parameters** page allows you to declare new parameters, set existing parameters and remove parameters within the package or archive. For more information on using parameters as a part of the synchronisation process see [Using DeclareParam](#) and [SetParam](#).



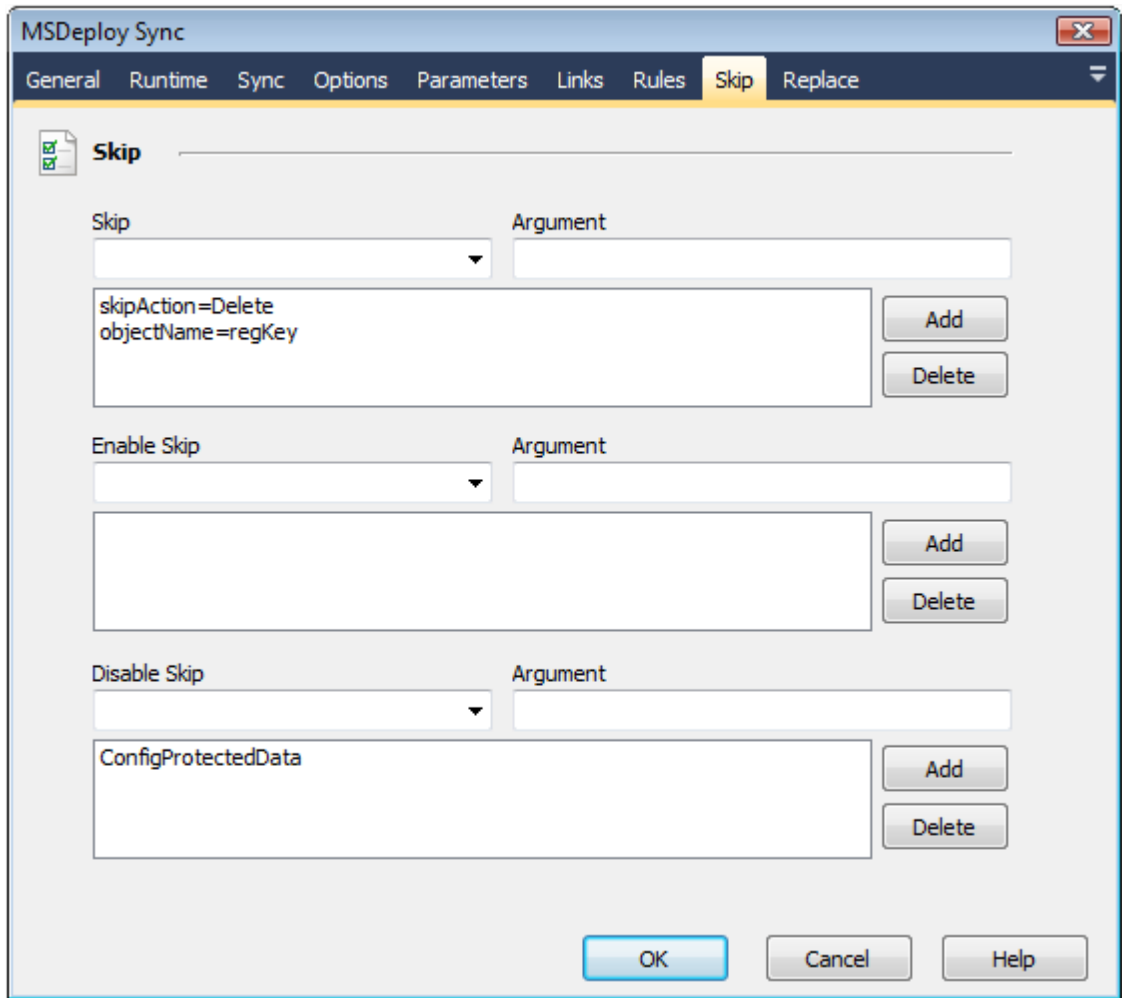
The **Links** page allows you to enable/disable link extensions which provide an overview of a Web Server that can be dumped or synchronised. For more information on Web Deploy Link Extensions see [Web Deploy Link Extensions](#).



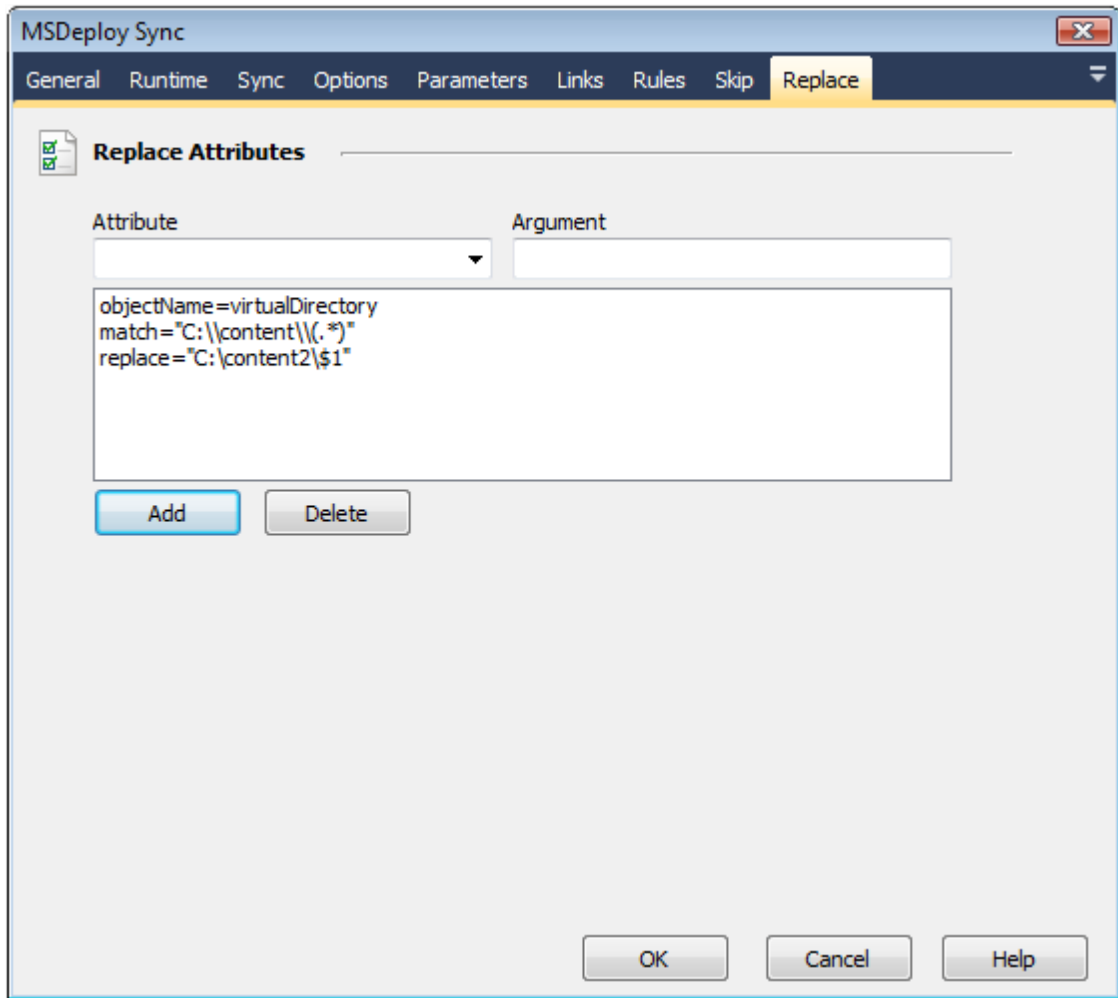
The **Rules** page allows you to specify which rules are executed as a part of the synchronisation process. For more information on Web Deploy Rules see [Web Deploy Rules](#).



The **Skip** page allows you to specify objects or actions to be excluded from the synchronisation process, it also allows you to enable and disable skip directives. For more information on skipping objects and actions as well as skip directives see Web Deploy Operation Settings.



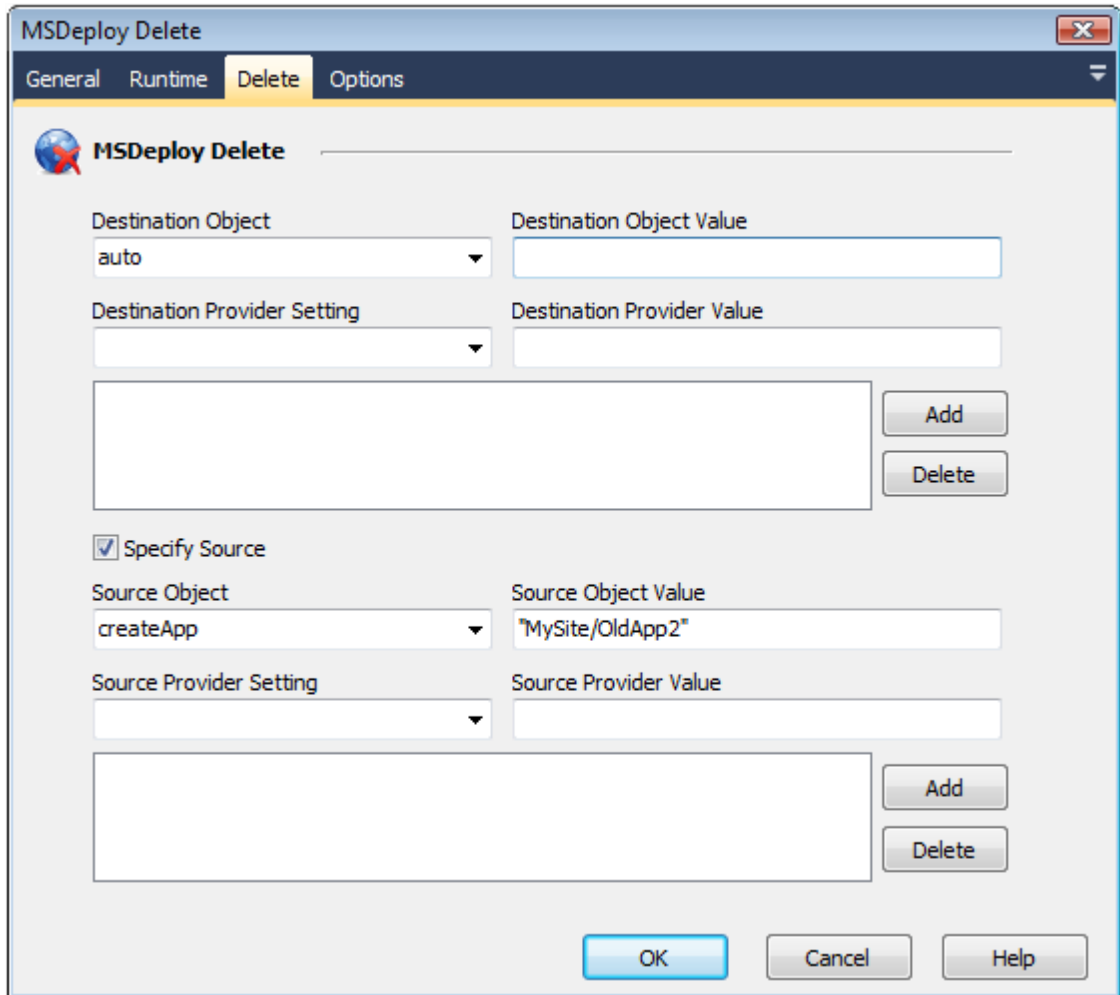
The **Replace** page allows you to replace object or attributes during a synchronisation process. It allows you to use a regular expression to match an object or attribute and replace the value of this item. For more information on using replace during synchronisation see Web Deploy Operation Settings.



For more information see Web Deploy Sync Operation.

6.19.13.2 MSDeploy Delete

The **MSDeploy Delete** action allows you to delete Web Deploy Provider objects (specified as the destination).



The image shows the 'MSDeploy Delete' dialog box. It has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Delete' (which is selected), and 'Options'. The main area of the dialog is titled 'MSDeploy Delete' and contains two sections. The first section is for the destination, with fields for 'Destination Object' (a dropdown menu showing 'auto'), 'Destination Object Value' (a text box), 'Destination Provider Setting' (a dropdown menu), and 'Destination Provider Value' (a text box). Below these fields is a list box with 'Add' and 'Delete' buttons. The second section is for the source, with a checked checkbox labeled 'Specify Source'. It has fields for 'Source Object' (a dropdown menu showing 'createApp'), 'Source Object Value' (a text box containing 'MySite/OldApp2'), 'Source Provider Setting' (a dropdown menu), and 'Source Provider Value' (a text box). Below these fields is another list box with 'Add' and 'Delete' buttons. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Note: Not all providers support the delete operation, the values available from the Source and Destination drop down boxes in the Delete action have been limited to the providers that support the delete function.

The *What If* option (available via the Options page) allows you preview the operation so that you can see the result without actually performing a delete.

If you specify the Source (optional) you need to specify either the same destination settings or set the destination to *auto*. The destination cannot be set to *auto* if no source has been provided.

For more information see Web Deploy Delete Operation.

6.19.13.3 MSDeploy Dump

The **MSDeploy Dump** action allows you to retrieve information about a specified Web Deploy Provider object. Extra information can be obtained from this action via enabling the *Xml Output* option via the Options page, this will provide additional attribute information.

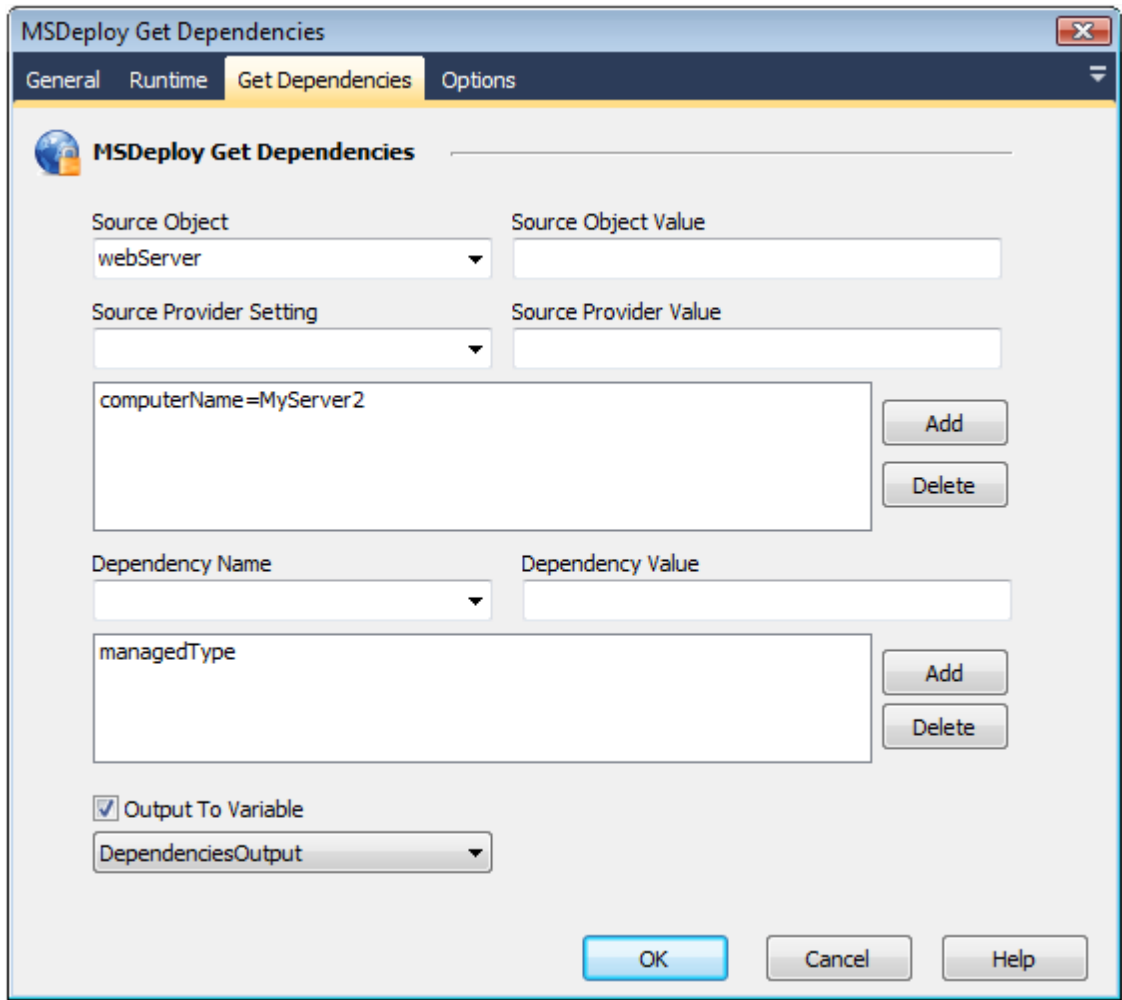
The screenshot shows the 'MSDeploy Dump' dialog box with the 'Dump' tab selected. The 'Source Object' is set to 'contentPath' and the 'Source Object Value' is 'C:\\inetpub\\wwwroot'. The 'Source Provider Setting' and 'Source Provider Value' fields are empty. There is a list box for settings with 'Add' and 'Delete' buttons. The 'Output To Variable' checkbox is checked, and the variable name is 'DumpOutput'. The 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

The information returned from this action can be saved into a variable.

For more information see [Web Deploy Dump Operation](#).

6.19.13.4 MSDeploy Get Dependencies

The **MSDeploy Get Dependencies** action allows you retrieve a list of dependencies for components that may need to be installed at the destination location before you can synchronise a website/web server to this location.



The image shows the 'MSDeploy Get Dependencies' dialog box. It has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Get Dependencies' (which is selected and highlighted in yellow), and 'Options'. The main content area is titled 'MSDeploy Get Dependencies' with a globe icon. It contains two sections for configuring dependencies. The first section has 'Source Object' (a dropdown menu with 'webServer' selected) and 'Source Object Value' (a text input field). Below these are 'Source Provider Setting' (a dropdown menu) and 'Source Provider Value' (a text input field). A large text area contains the text 'computerName=MyServer2'. To the right of this text area are 'Add' and 'Delete' buttons. The second section has 'Dependency Name' (a dropdown menu) and 'Dependency Value' (a text input field). Below these is another large text area containing 'managedType'. To the right of this text area are 'Add' and 'Delete' buttons. At the bottom left, there is a checked checkbox labeled 'Output To Variable' and a dropdown menu showing 'DependenciesOutput'. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

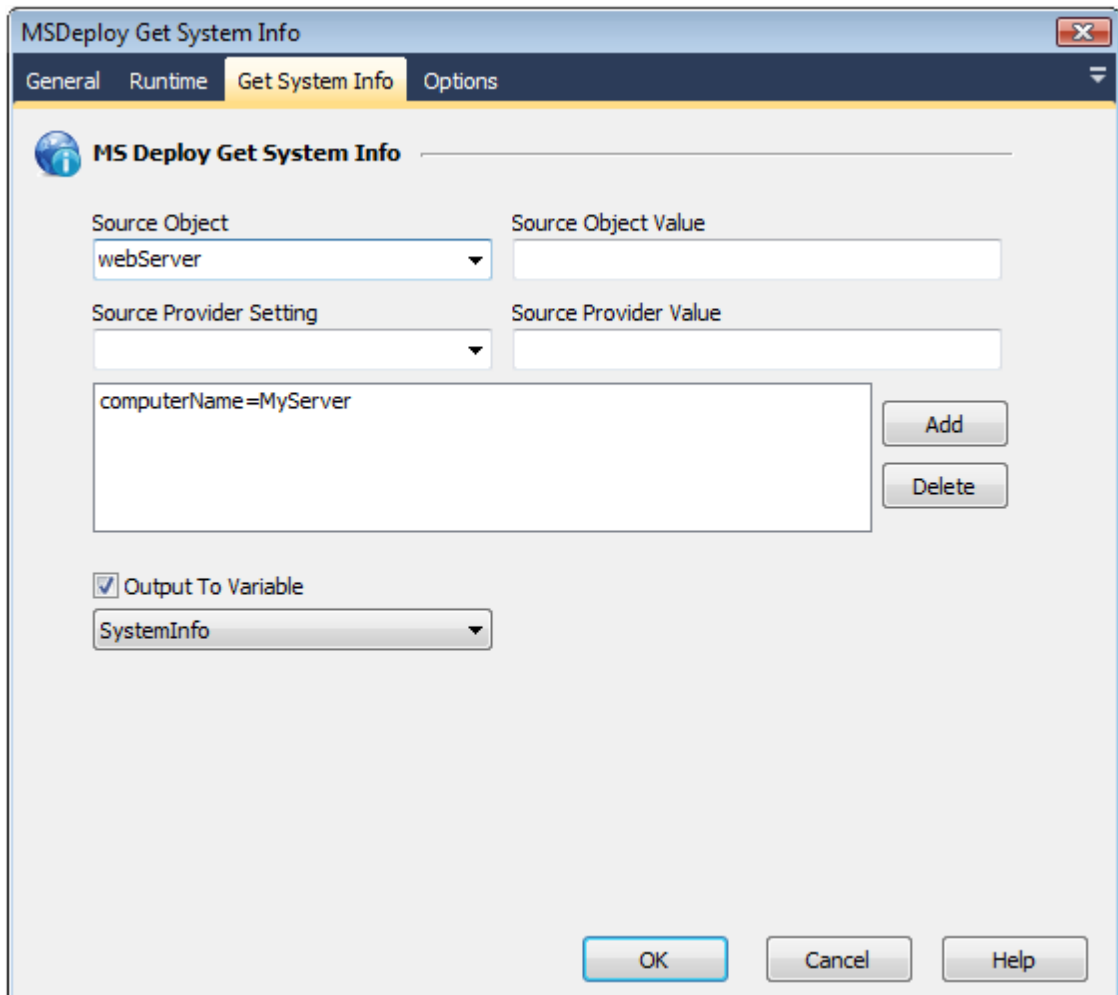
This information is returned in Xml and can be saved into a variable. MSDeploy will also accept an XPath value that can be applied to the results before they are returned. This feature is available via the Options page.

You can specify the dependencies that the action will return via the dependency settings.

For more information see [Web Deploy Get Dependencies Operation](#).

6.19.13.5 MSDeploy Get System Info

The **MSDeploy Get System Info** action allows you to retrieve IIS related information from a system in Xml format.



The information retrieved can be saved into a variable. MSDeploy will also accept an XPath value that can be applied to the results before they are returned. This feature is available via the Options page.

The information returned from this action will include details such as:

- Operating System Version
- Windows Directory Location
- Computer Name
- Processor Architecture
- IIS Version
- ASP .NET version

- List of the IIS Components Installed

For more information see Web Deploy Get System Info Operation.

6.19.14 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.

6.19.14.1 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.

SSH Open Connection

General Runtime **Connection**

Connection Settings

Connection Name:

Server: Port:

Server Fingerprint (optional): ☐ Set from server if null

User settings

User Name: Password:

Private Key File (optional):

Options

Shell type: Connect Timeout (ms):

☐ Verbose logging

OK Cancel Help

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 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

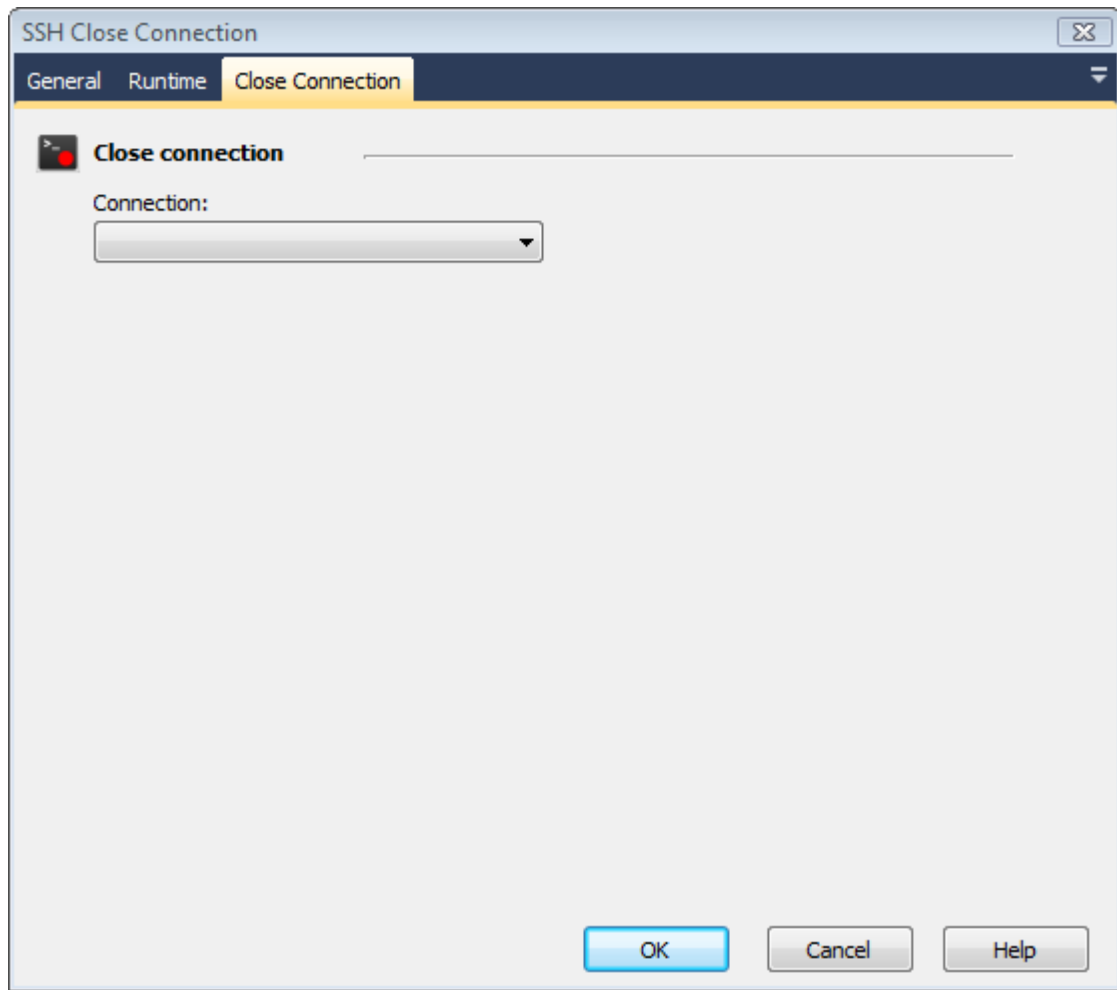
The amount of time (in ms) that the action should wait for the server to connect. If set too low, the connection may fail, or the setting of the shell type may fail. If you are seeing "garbage" characters in the output from commands, try changing shell types and/or increasing the timeout.

Verbose logging

Enable verbose logging

6.19.14.2 SSH Close Connection

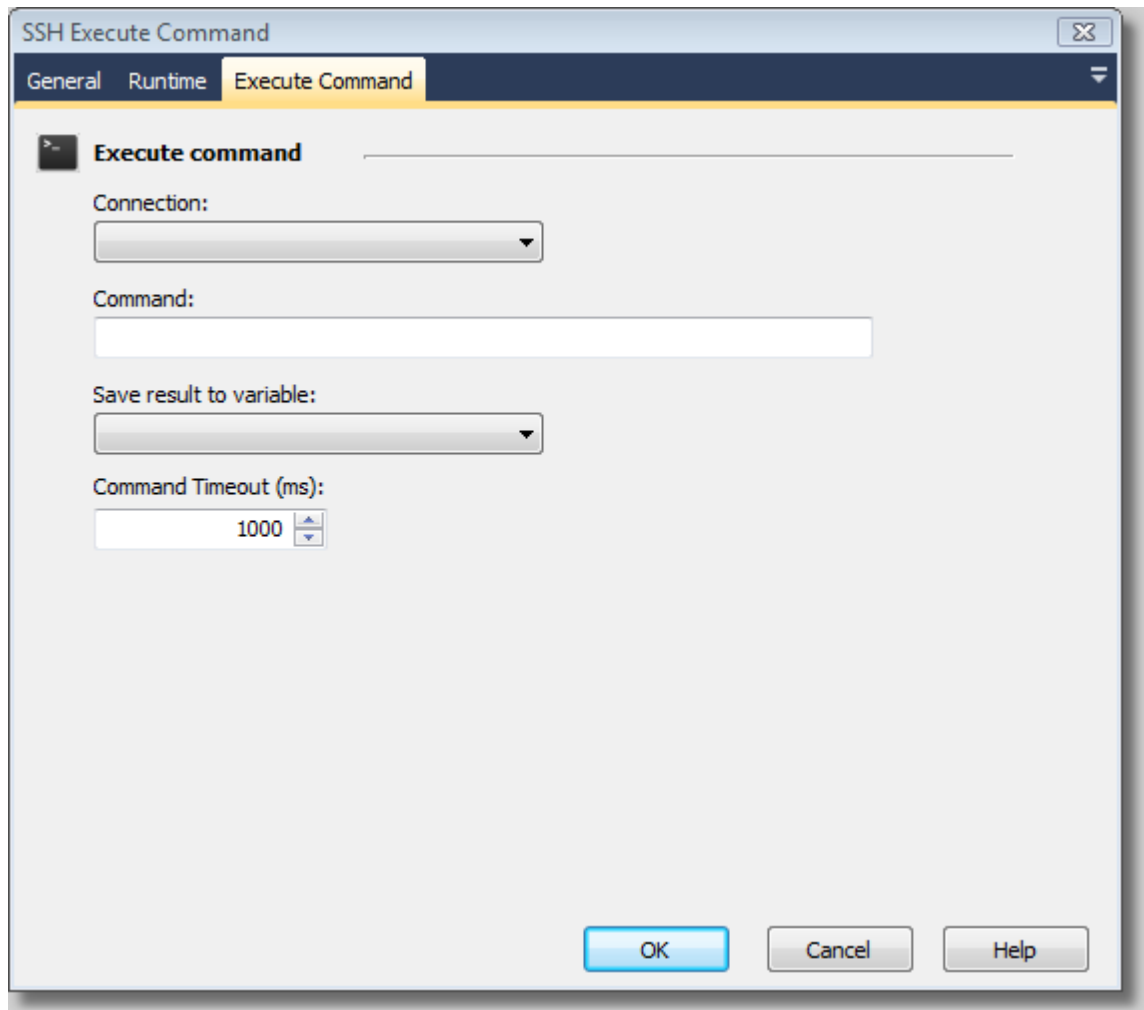
The SSH Close Connection action closes a connection opened by SSH Open Connection. If you do not use this action to close a connection, the connection will remain open until the FinalBuilder project is closed.

**Connection**

The connection to close. If the connection is already closed, this action will fail.

6.19.14.3 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.

**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.

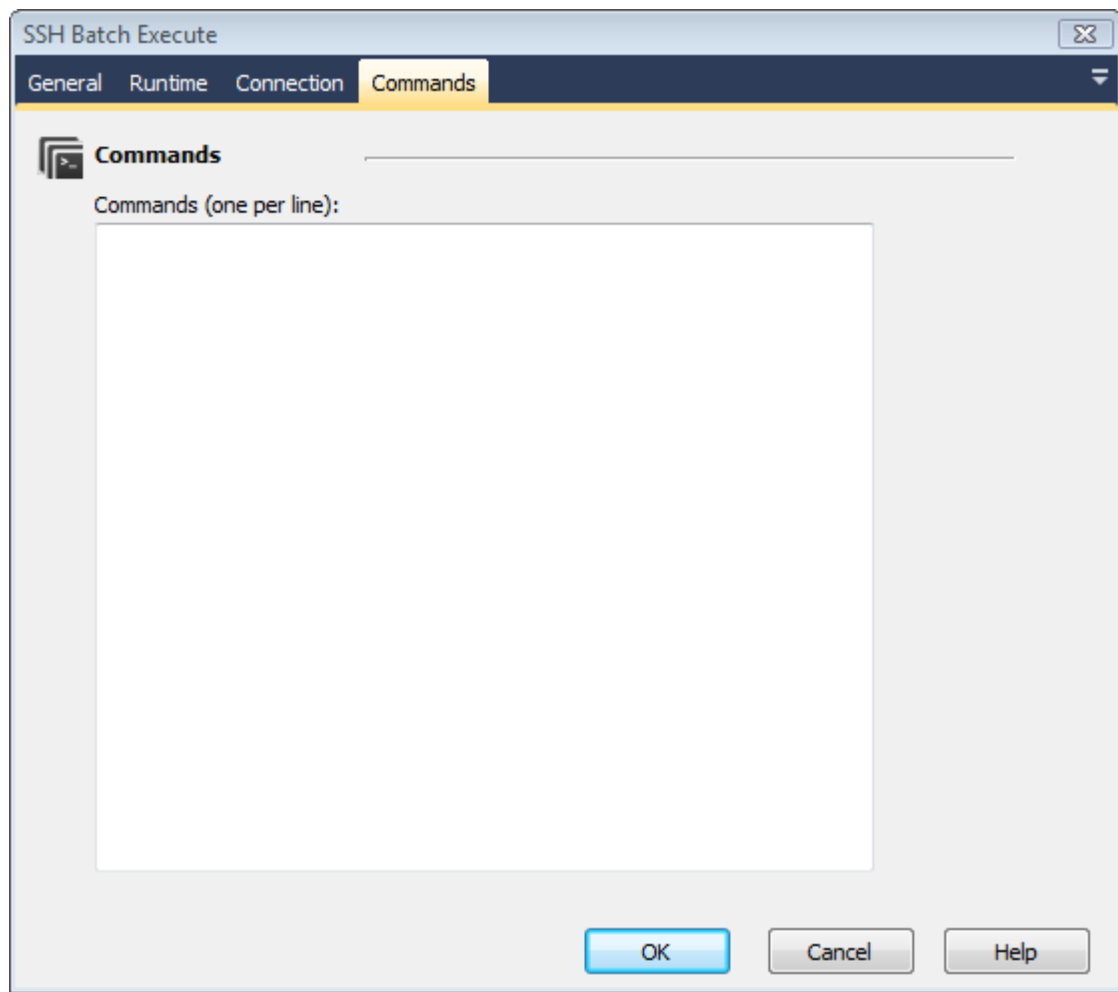
Timeout

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.

6.19.14.4 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**

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.

6.19.15 SFTP

The SFTP actions in FinalBuilder provide client-side functionality for SFTP (Secure File Transfer Protocol).

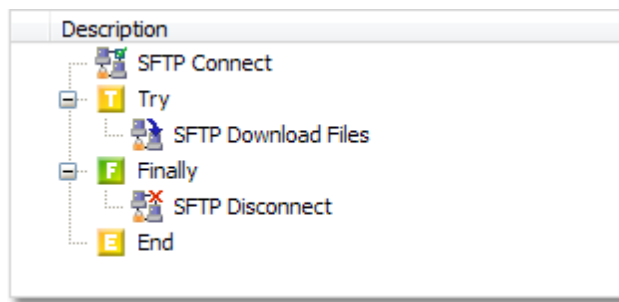
The actions implement SSH File Transfer Protocol. It works over secure SSH channel and has nothing common with FTP protocol. SFTP works as a subsystem of SSHv2 protocol, i.e. SFTP negotiation flow is the following:

- 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:



6.19.15.1 SFTP Connect

Use the SFTP Connect action to create a connection to an SFTP server which you can then use with other SFTP actions.

The screenshot shows the 'SFTP Connect' dialog box with the 'Server & Authentication' tab selected. The dialog has four main sections: 'Connection Name', 'SFTP Server', 'Authentication', and 'Options'.
- **Connection Name:** A text field containing 'TestConnection'.
- **SFTP Server:** Contains 'Host name' (text field with 'localhost') and 'Port' (spin box with '22'). Below these is 'Server Public Key' (text field with '678aaa24378b05cfaacf271ce0e8a966') and a checkbox 'Cache public key on first connection (or when key blank)' which is checked. The text 'RSA 1024 bit MD5 hash' is to the right of the key field.
- **Authentication:** Contains 'Username' (text field with 'example') and 'Password' (password field with 10 dots). Below is 'Private Key File (optional)' with an empty text field and a folder icon.
- **Options:** Contains a checkbox 'Keep connection alive' which is unchecked. Below it is a spin box with '10000' and the text 'Tunnel inactivity period (in ms), after which the keep-alive signal will be sent'.
At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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.

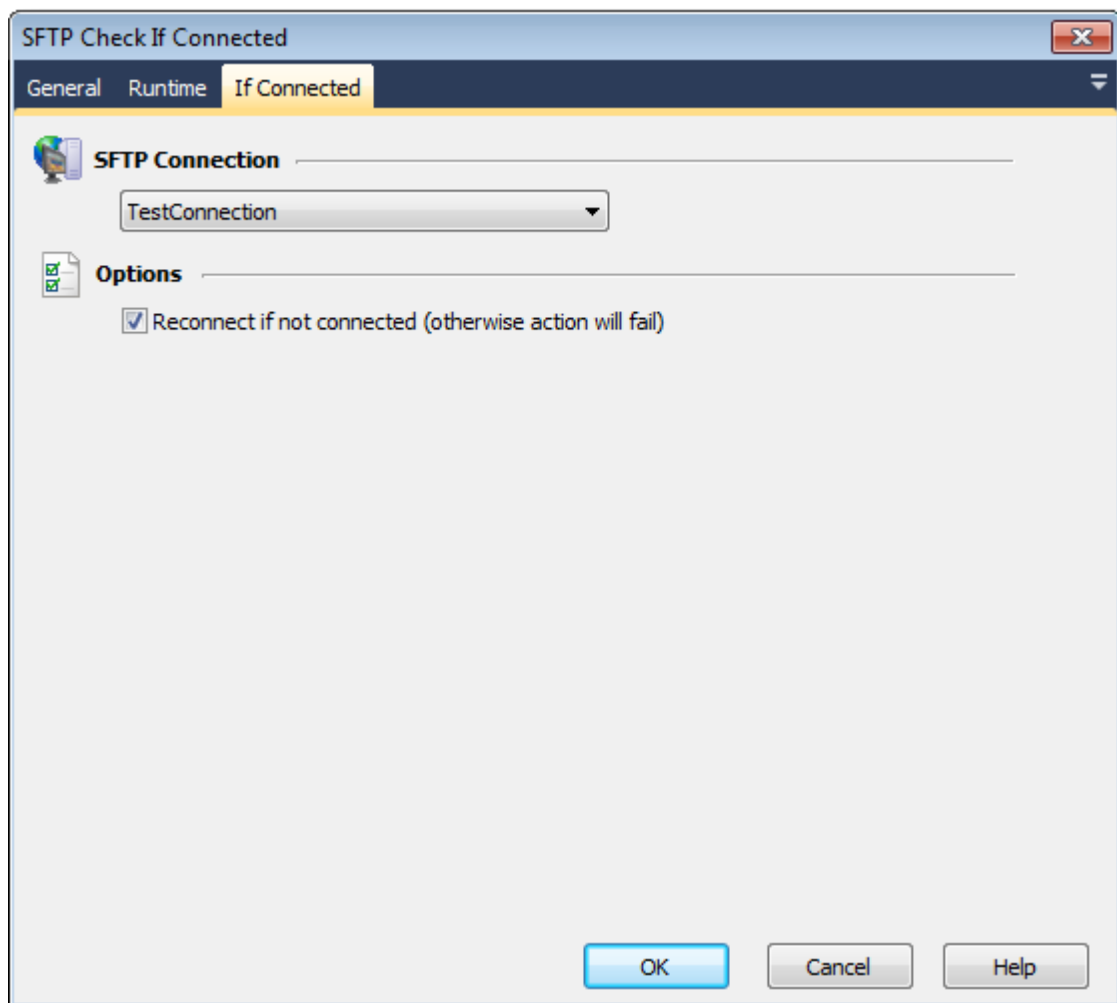
6.19.15.2 SFTP Disconnect

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.

6.19.15.3 SFTP Check if Connected

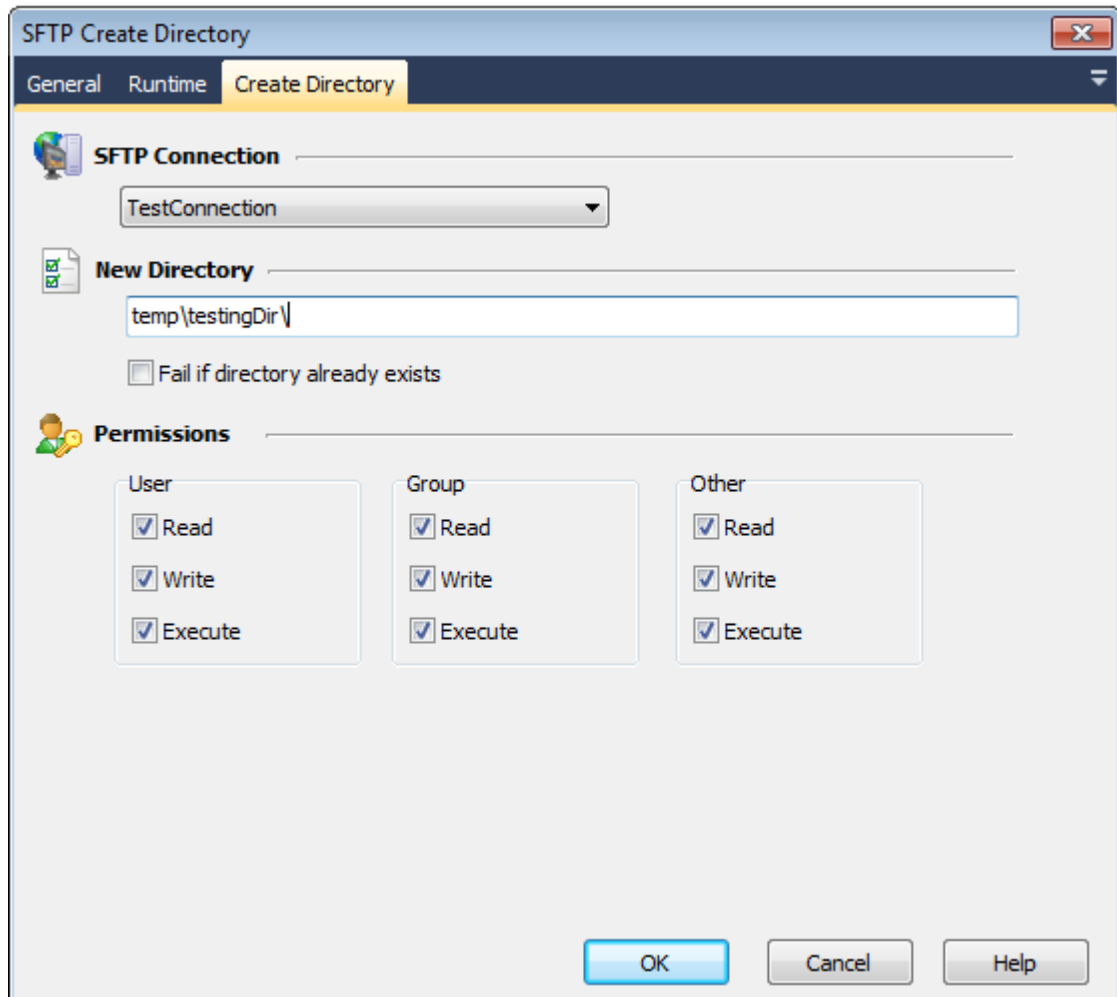
Use the SFTP Check if Connected action to validate if the connection to the SFTP server is still valid.

Optionally reconnect to the SFTP server (using the details from the SFTP Connect action) if the connection has been lost.



6.19.15.4 SFTP Create Directory

Use the SFTP Create Directory action to create a directory on the remote server.



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

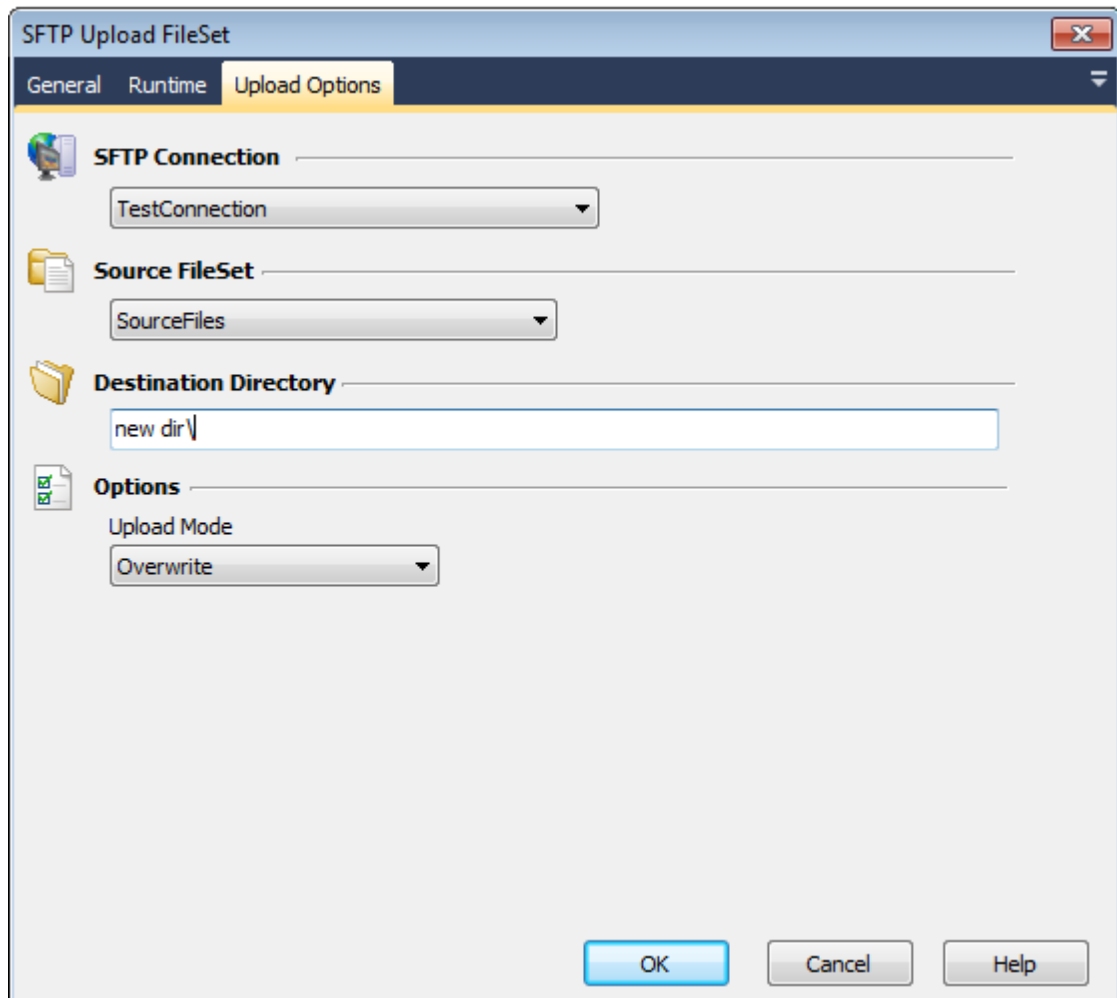
Optionally fail the action if the directory to create already exists on the server.

Permissions

Specify the permissions that the new directory should have.

6.19.15.5 SFTP Upload FileSet

Use the SFTP Upload FileSet action to upload all the files in an existing FileSet.



Source FileSet

Select the FileSet that contains the files that you want to transfer.

Destination Directory

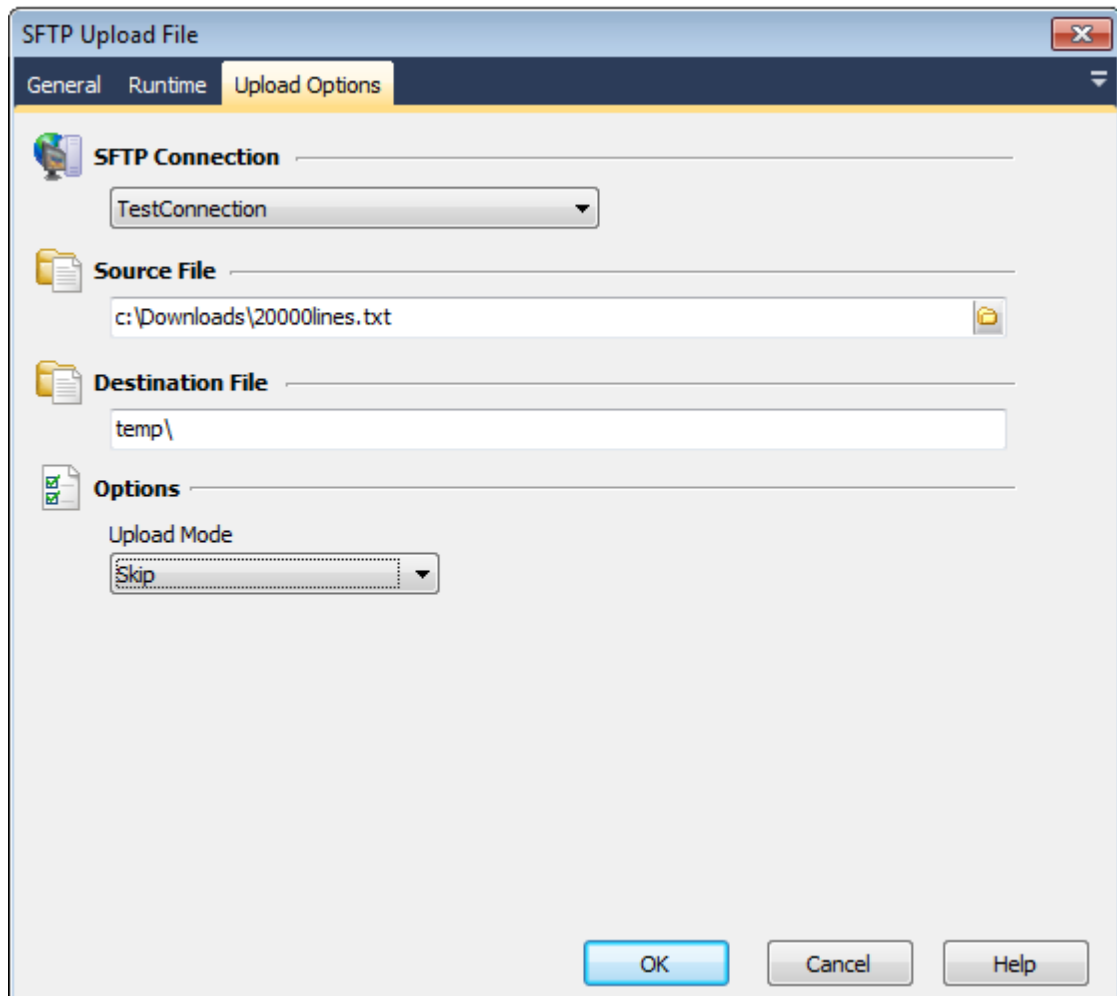
Specify the name of the directory on the SFTP server. You may specify a directory that doesn't exist and it will be created for you. Make sure the directory is a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept of a "current directory" like FTP does.

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.

6.19.15.6 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

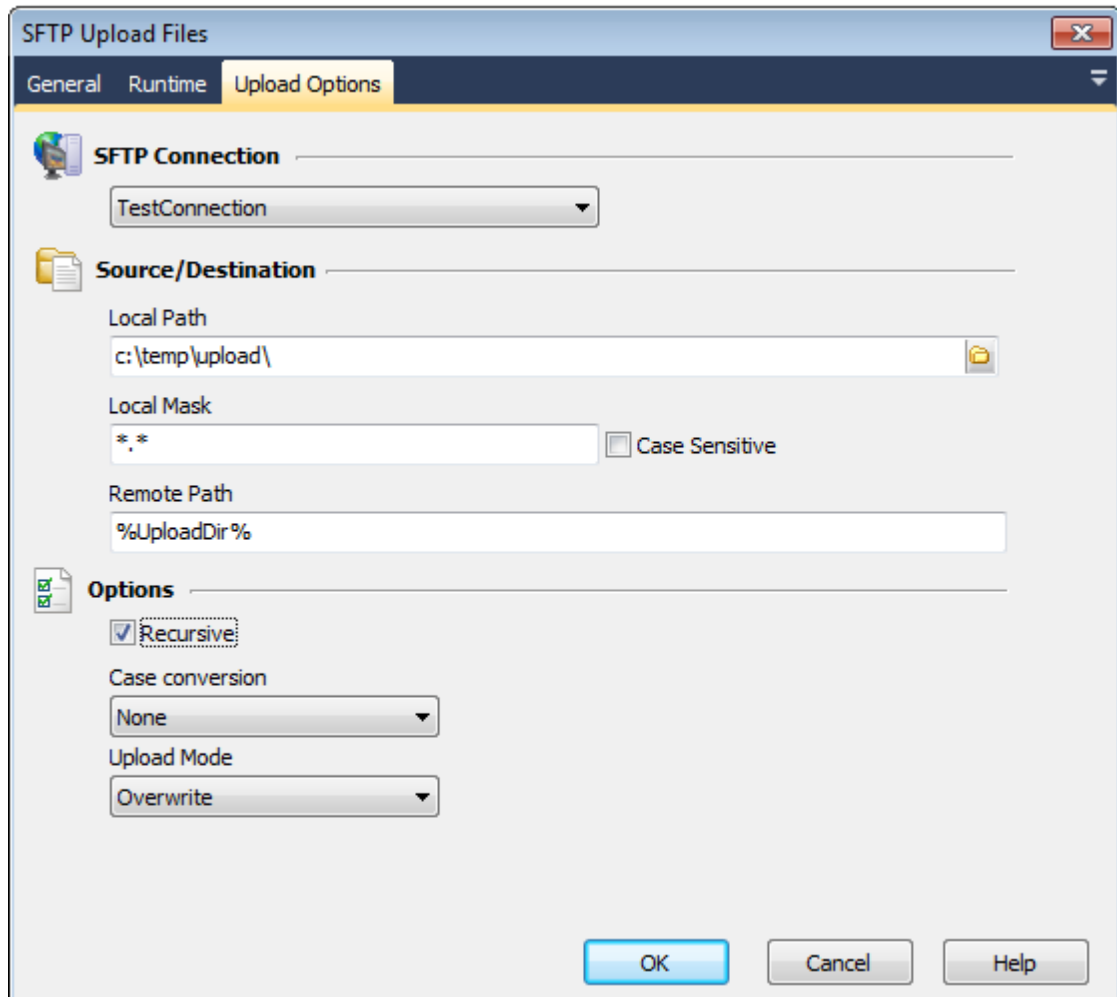
Specify the destination directory or destination directory including the filename on the SFTP server. You may specify a directory that doesn't exist and it will be created for you. Make sure the directory is a full path (specified from the user's home directory). The SFTP protocol doesn't have the concept 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.

6.19.15.7 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.



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

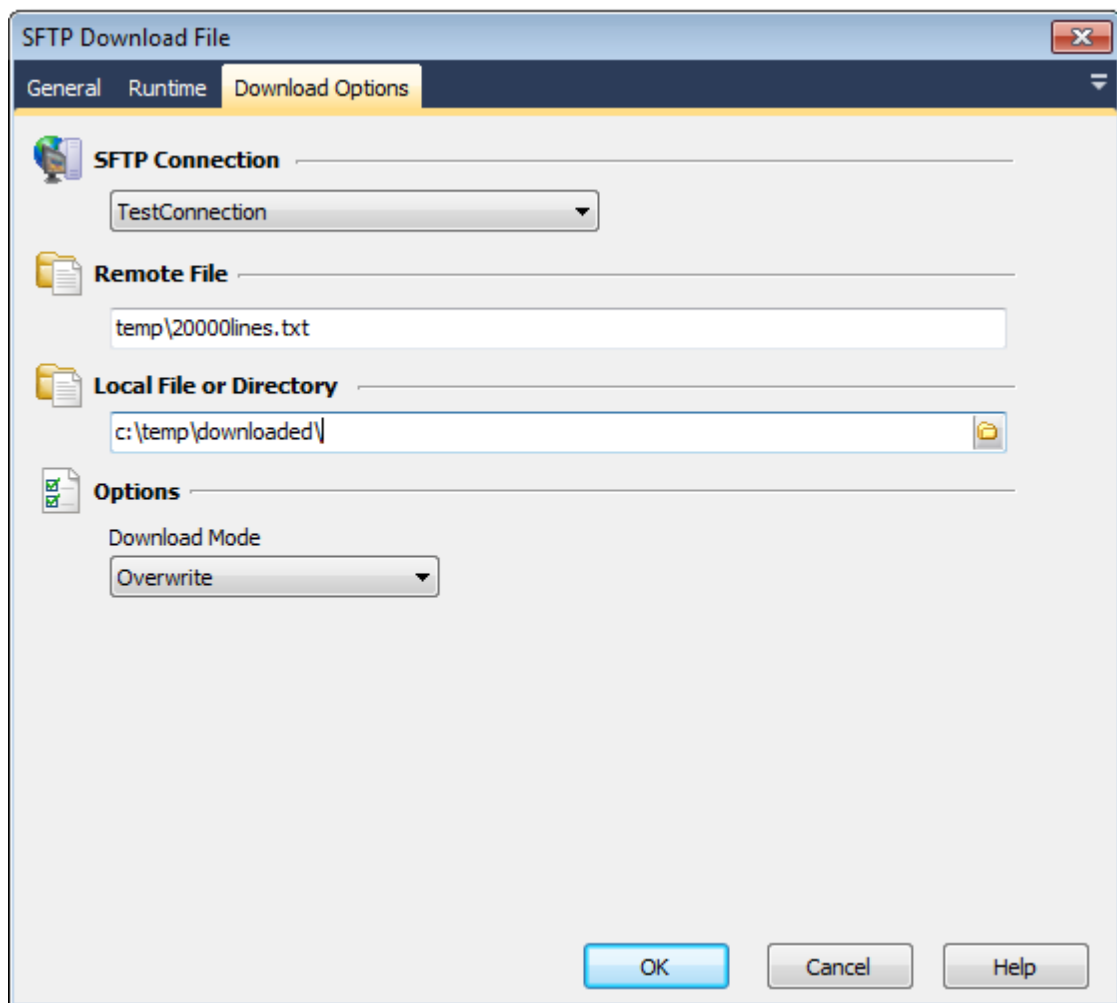
To convert the filenames to either lowercase or uppercase choose the corresponding option, otherwise choose "None" to leave the filenames as-is.

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.

6.19.15.8 SFTP Download File

Use the SFTP Download File action to download a single file from the SFTP server.



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

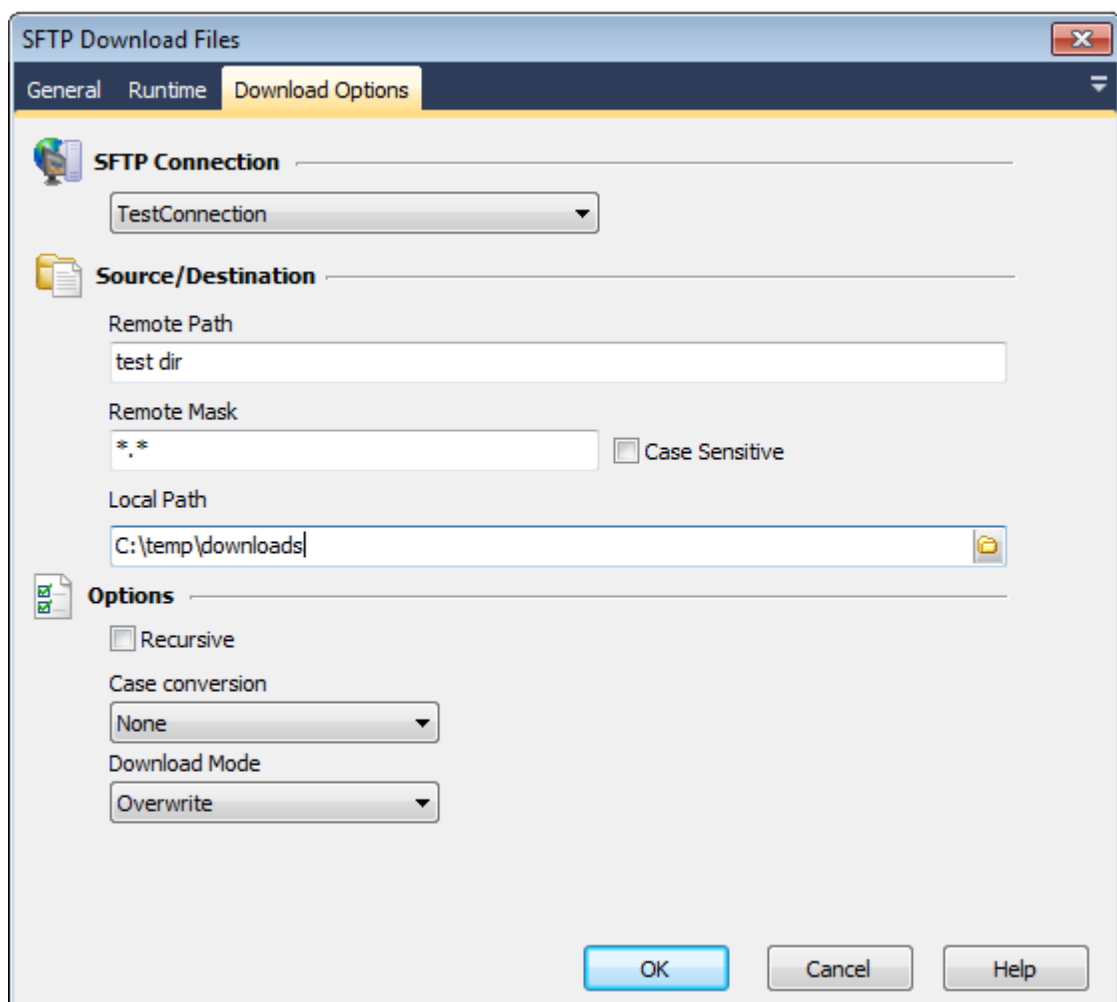
Specify either a local filename (including path) or a local directory. If specifying the local filename, then the file will be transferred and renamed to this new name.

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.

6.19.15.9 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.



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

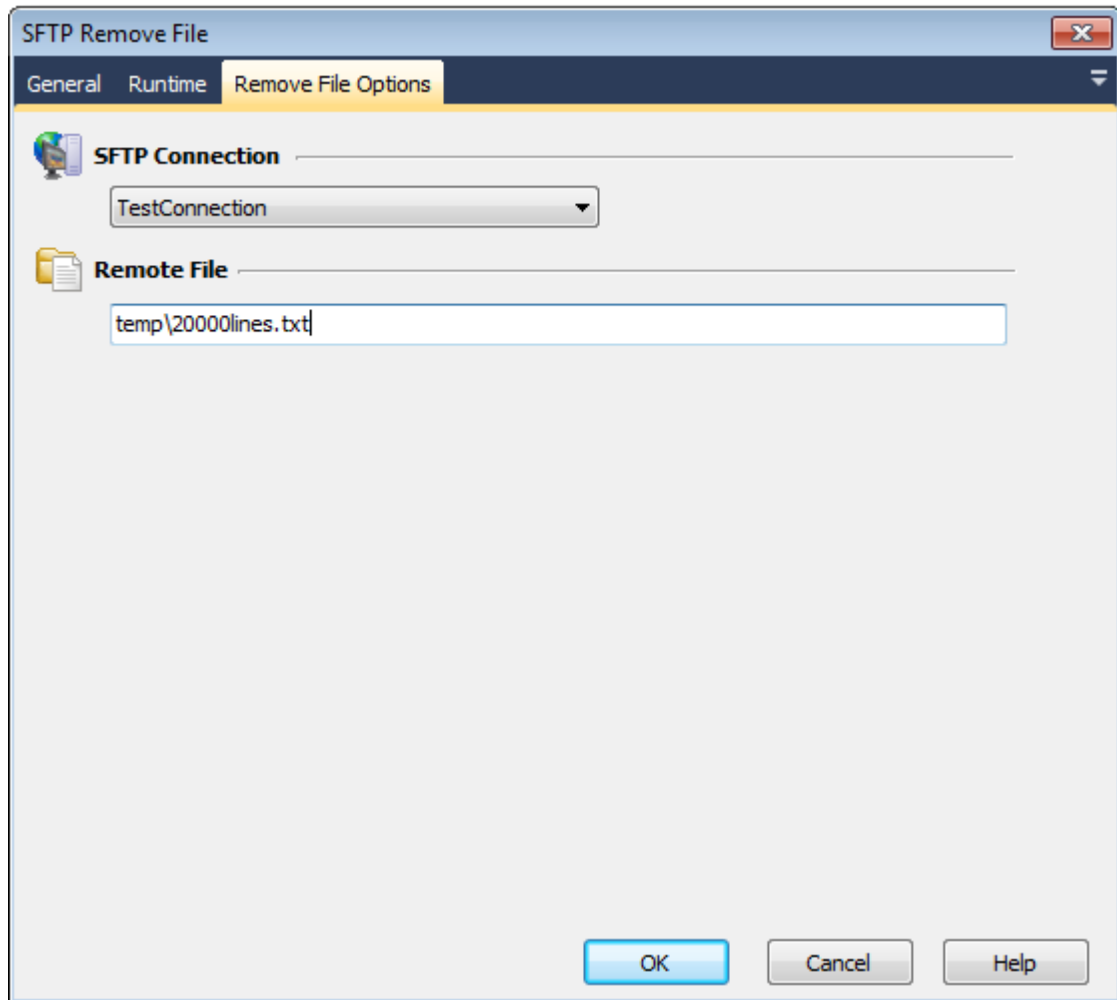
To convert the filenames to either lowercase or uppercase choose the corresponding option, otherwise choose "None" to leave the filenames as-is.

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.

6.19.15.1(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.

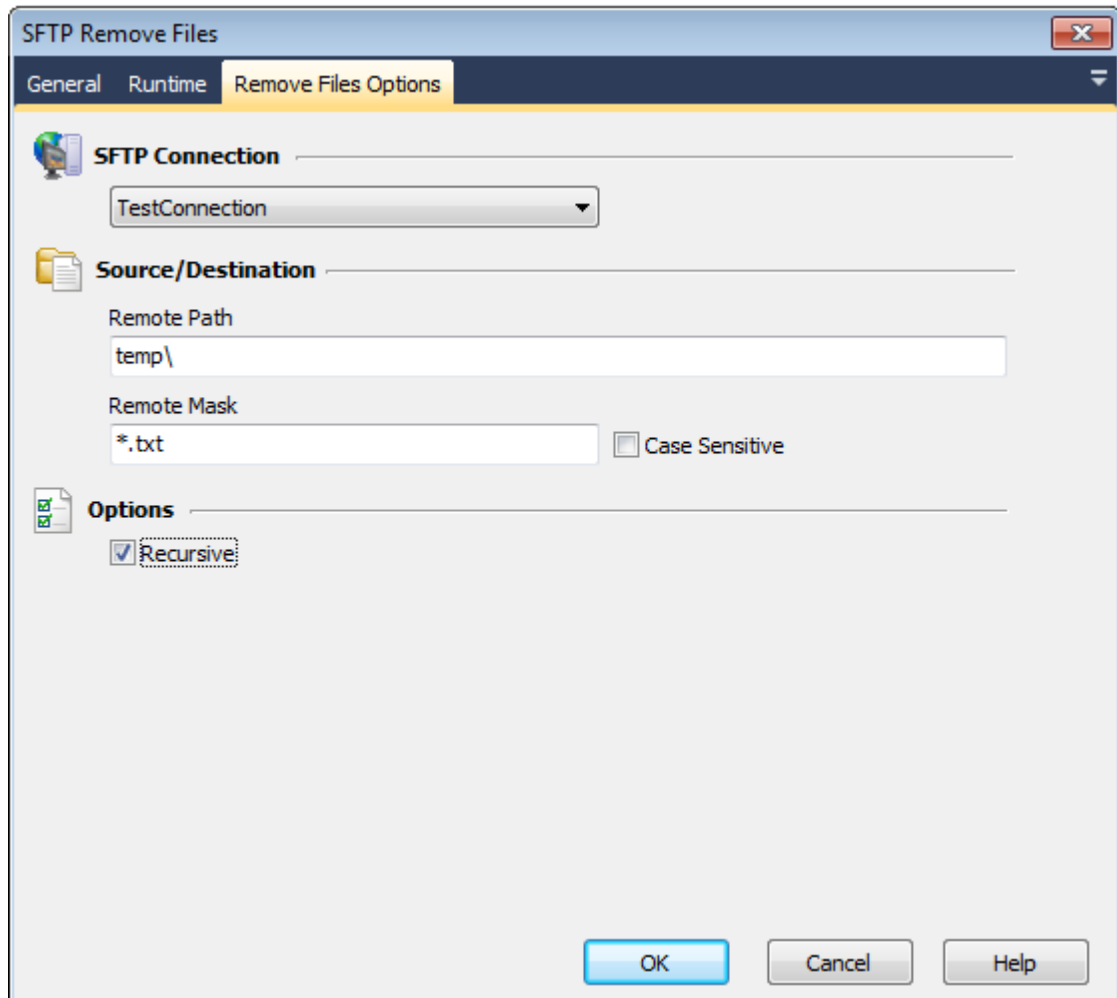


Remote File

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.

6.19.15.1 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.

**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

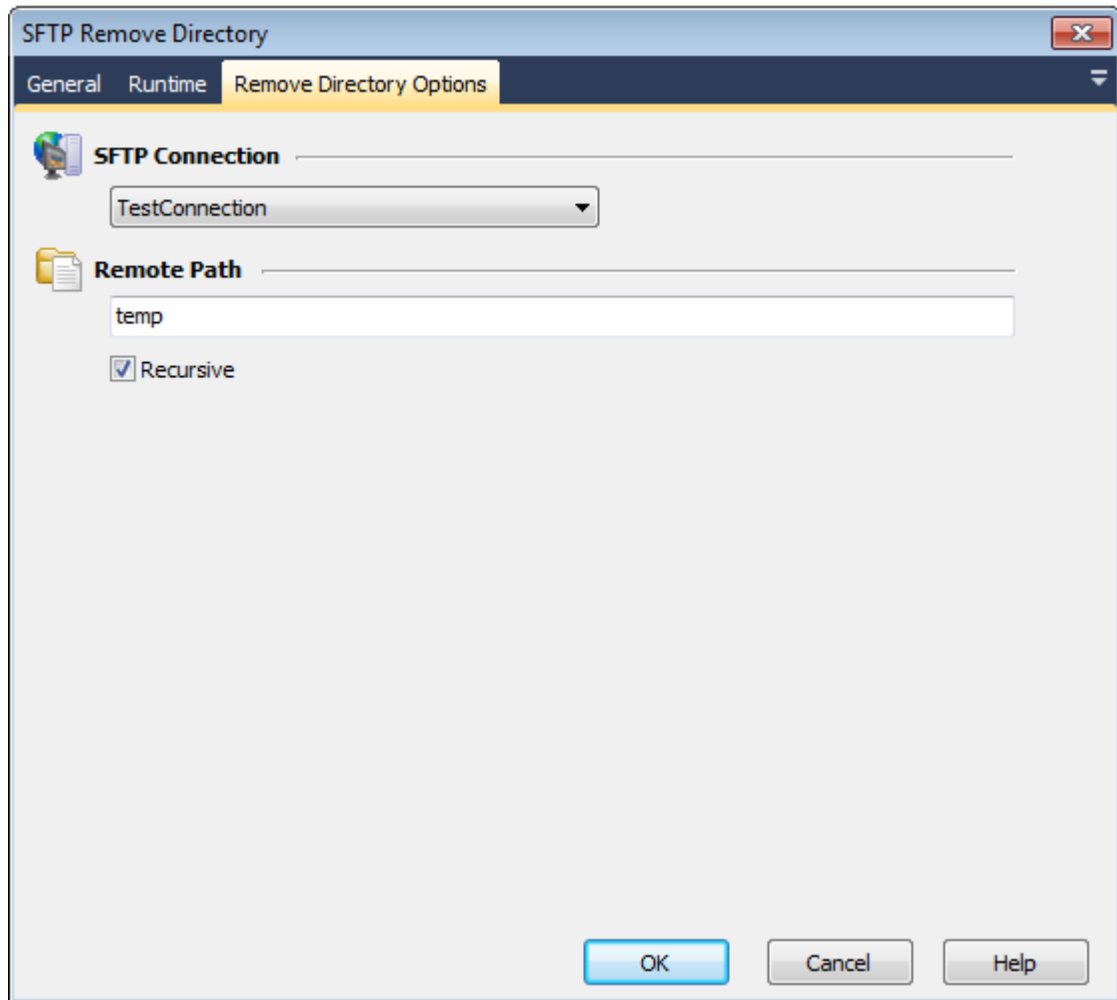
Enter the remote mask to match against the files and directories to delete. Case Sensitive specifies whether to treat filenames case-sensitively.

Recursive

Specify the recursive option to delete files in child directories also.

6.19.15.1 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.

**Remote Path**

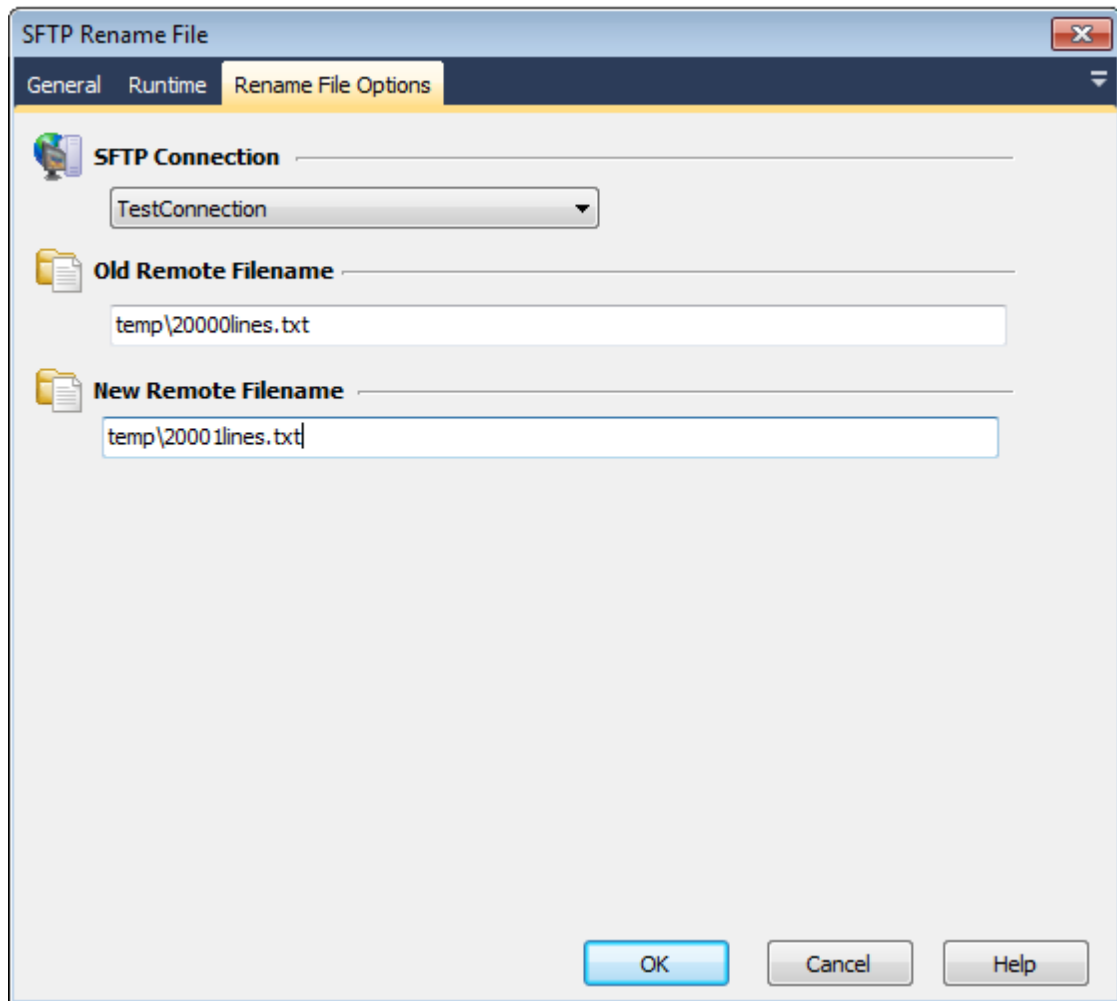
Enter the path to the directory to be removed on the remote server. Make sure the full path is included (specified from the user's home directory) as the SFTP protocol doesn't have the concept of a "current directory" like FTP does.

Recursive

Specifies whether to remove directories recursively, i.e. including subdirectories

6.19.15.1 SFTP Rename File

Use the SFTP Rename File action to change the name of a file on the SFTP server.



Old Remote Filename

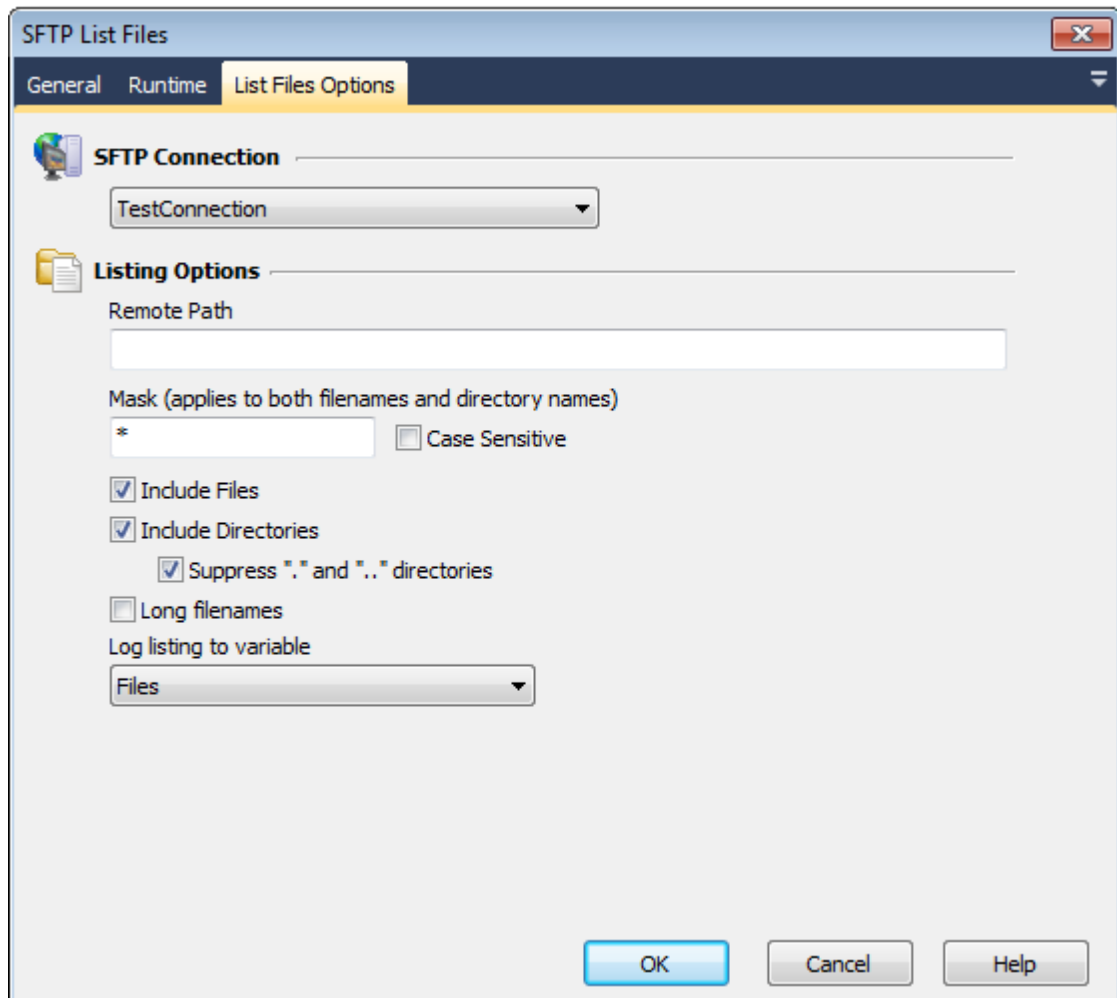
Path to existing file on the remote SFTP server.

New Remote Filename

Specifies the new name/location of the file on the remote SFTP server.

6.19.15.1 SFTP List Files

Use the SFTP List Files action to list the files and directories on the SFTP server.



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

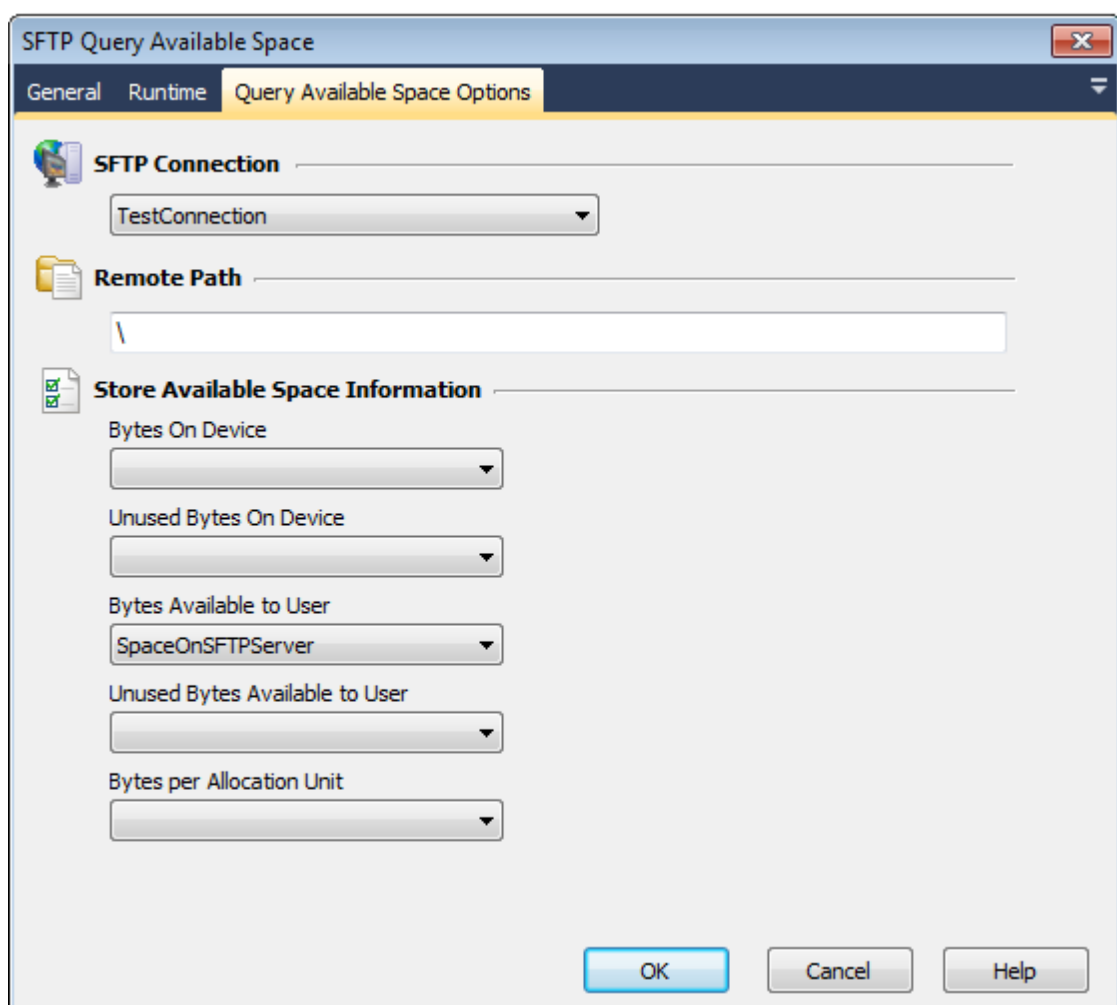
Specify to return a long listing. The long listing includes the permissions, size of file, ownership and other properties.

Log listing to variable

Specify a variable name to copy the listing into.

6.19.15.1 SFTP Query Available Space

Use the SFTP Query Available Space action to change the name of a file on the SFTP server.



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

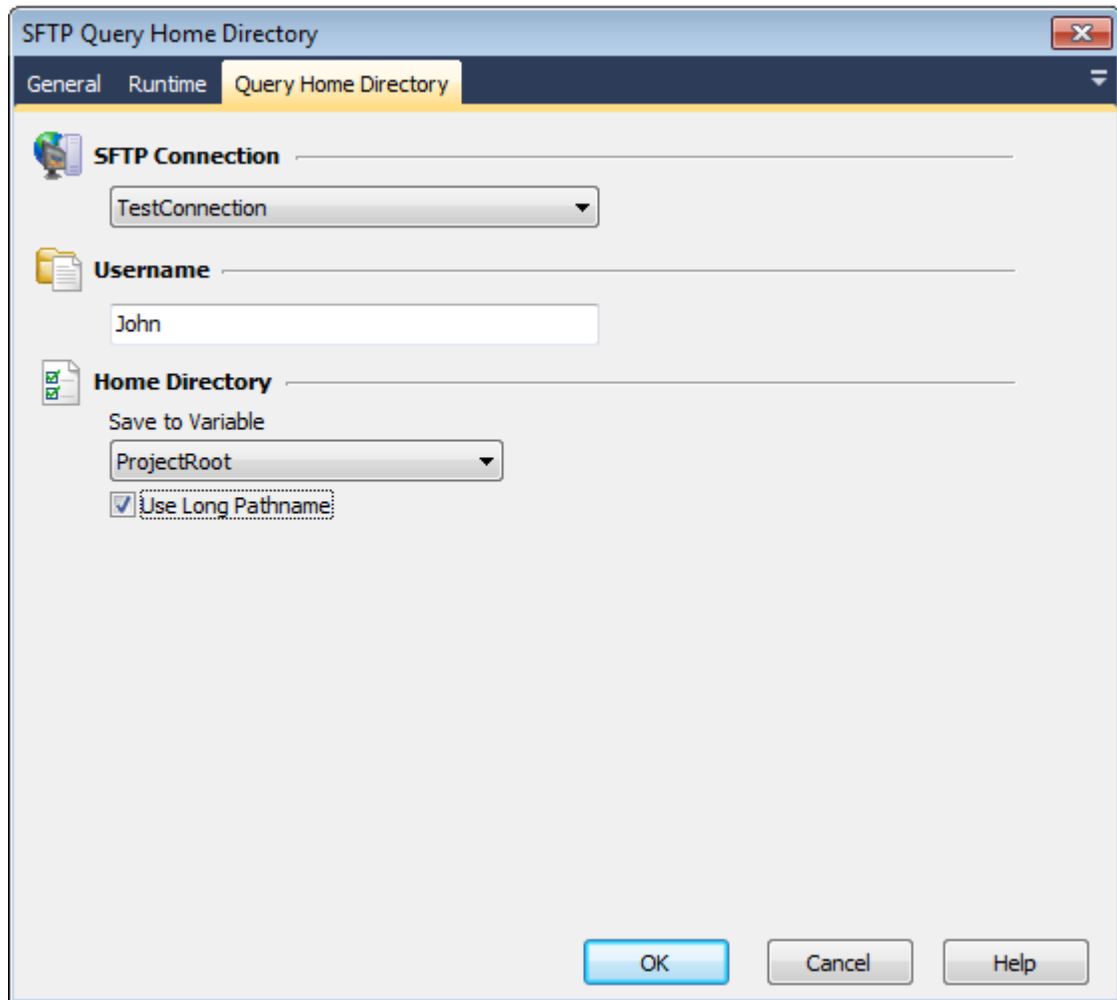
Specifies unused space available to the current user in bytes. Specify the variable to store the information in.

Bytes per Allocation Unit

Specifies number of bytes per allocation unit on the device. Specify the variable to store the information in.

6.19.15.1 SFTP Query Home Directory

Use the SFTP Query Home Directory action to request the path to the user's home directory.

**Username**

Specify the username for which the home directory is requested

Save to variable

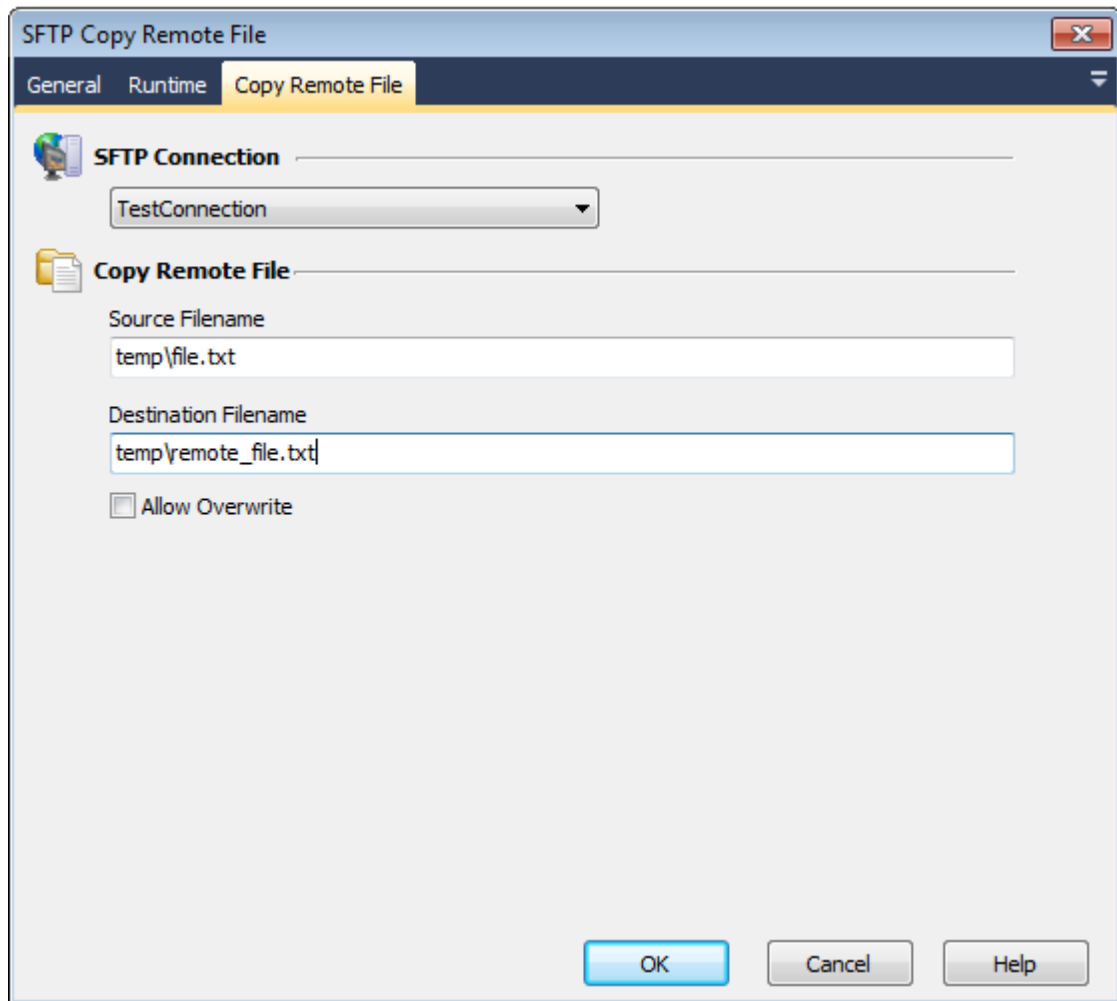
Specify a variable to save the home directory information into.

Use Long Pathname

Specify to return a long listing. The long listing includes the permissions, size of file, ownership and other properties.

6.19.15.1 SFTP Copy Remote File

Use the SFTP Copy Remote File action to copy a remote file into a new location.



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.

6.19.16 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.

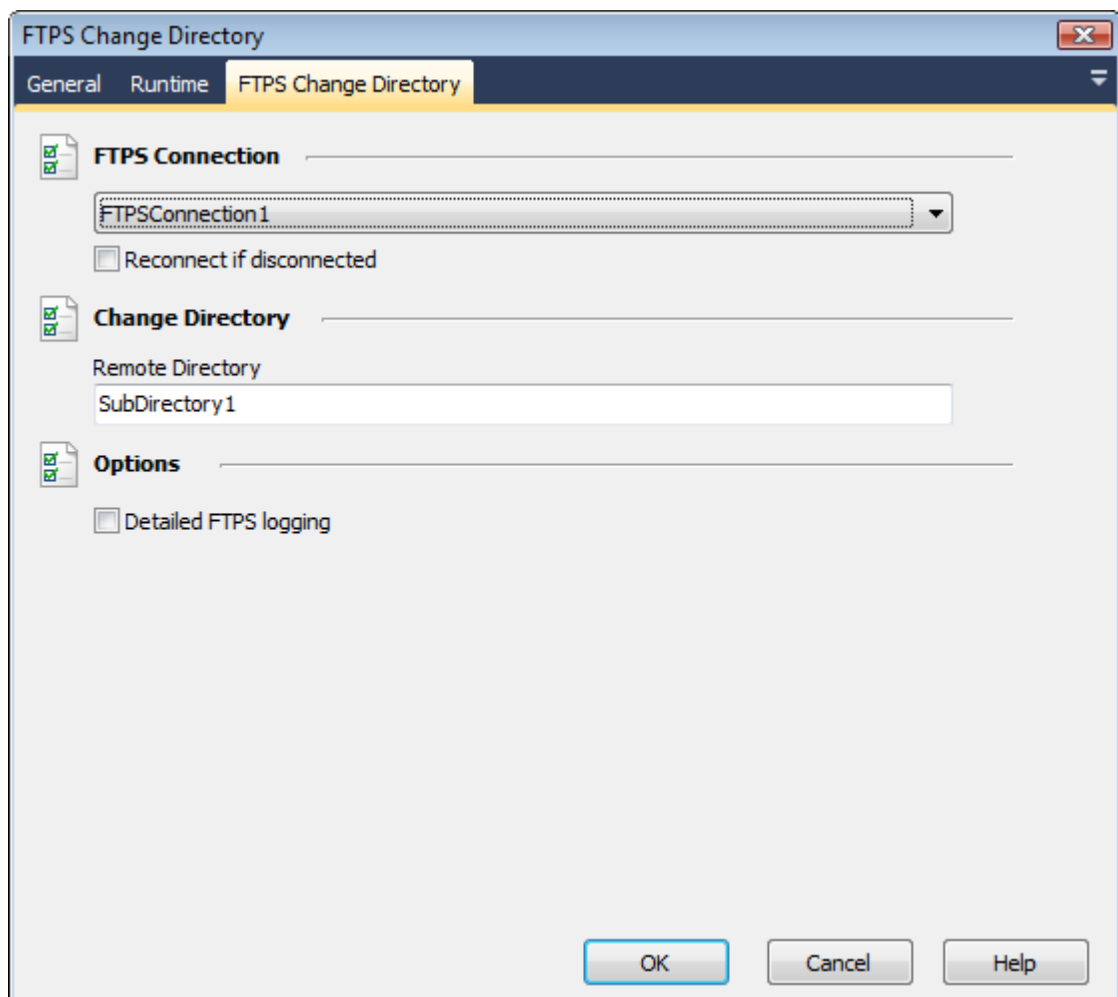
FTPS Options

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.

6.19.16.1 FTPS Change Directory

The FTPS Change Directory action allows you to change the current directory of the FTPS server.



Connection Name - The name of the FTPS connection to be used for the operation.

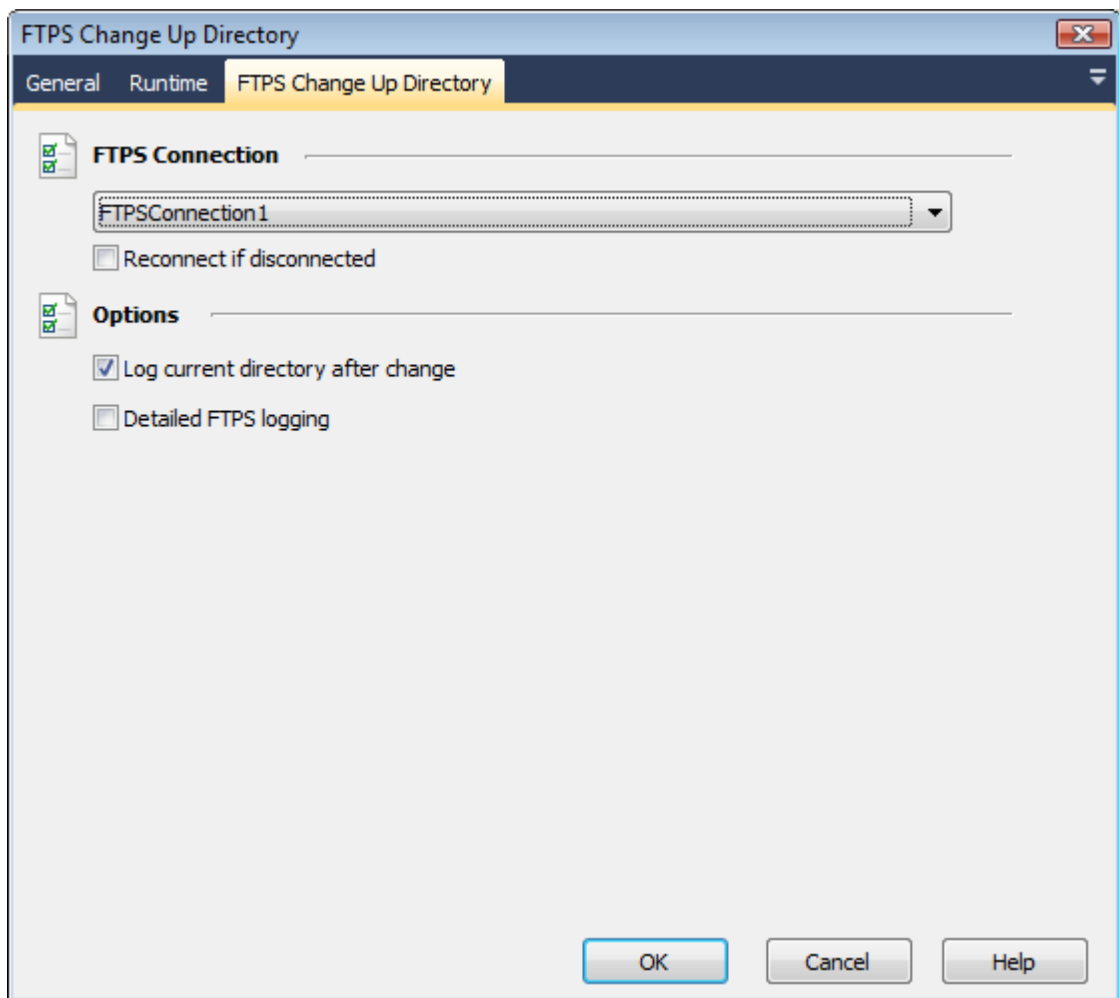
Reconnect if disconnected - If the connection has become disconnected, attempt to reconnect and then execute the action.

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.

6.19.16.2 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.



Connection Name - The name of the FTPS connection to be used for the operation.

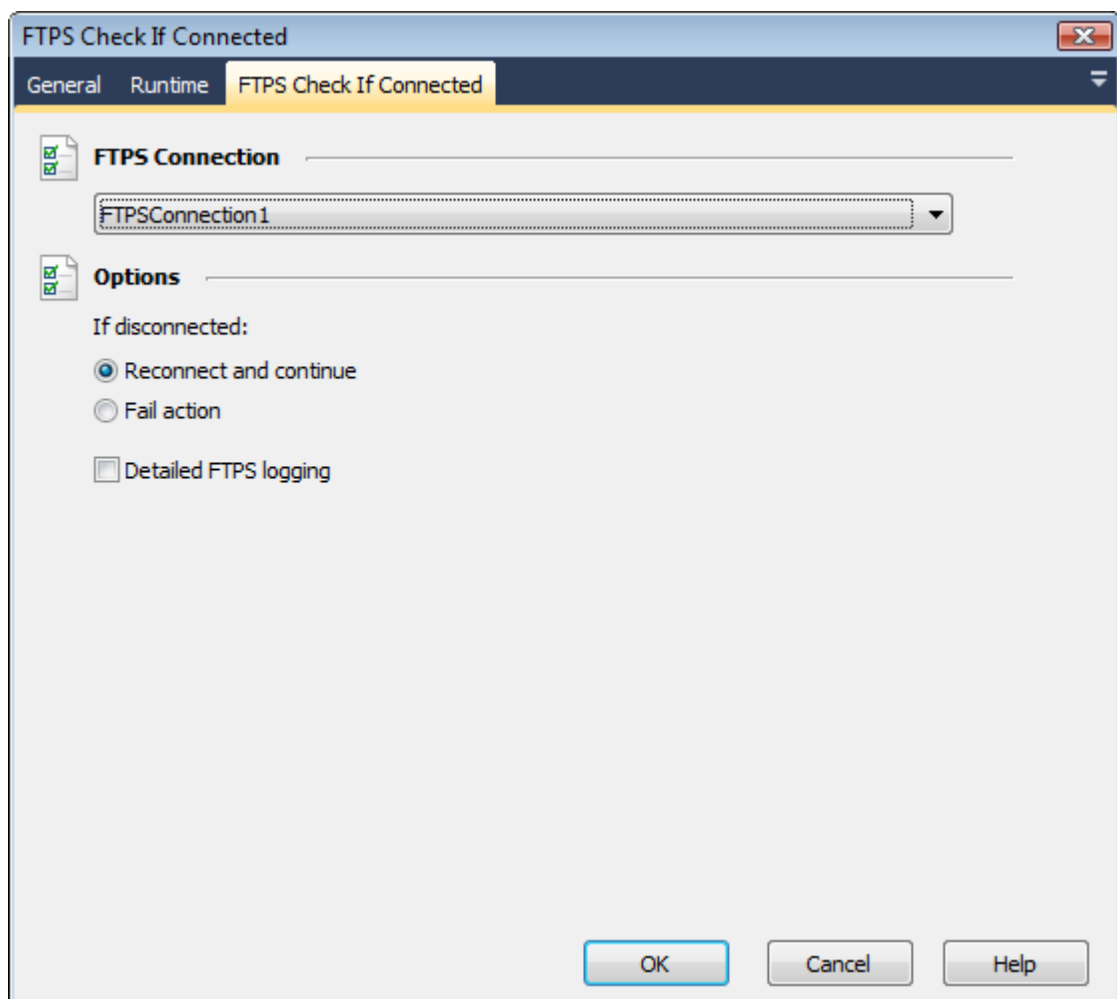
Reconnect if disconnected - If the connection has become disconnected, attempt to reconnect and then execute the action.

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.

6.19.16.3 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.

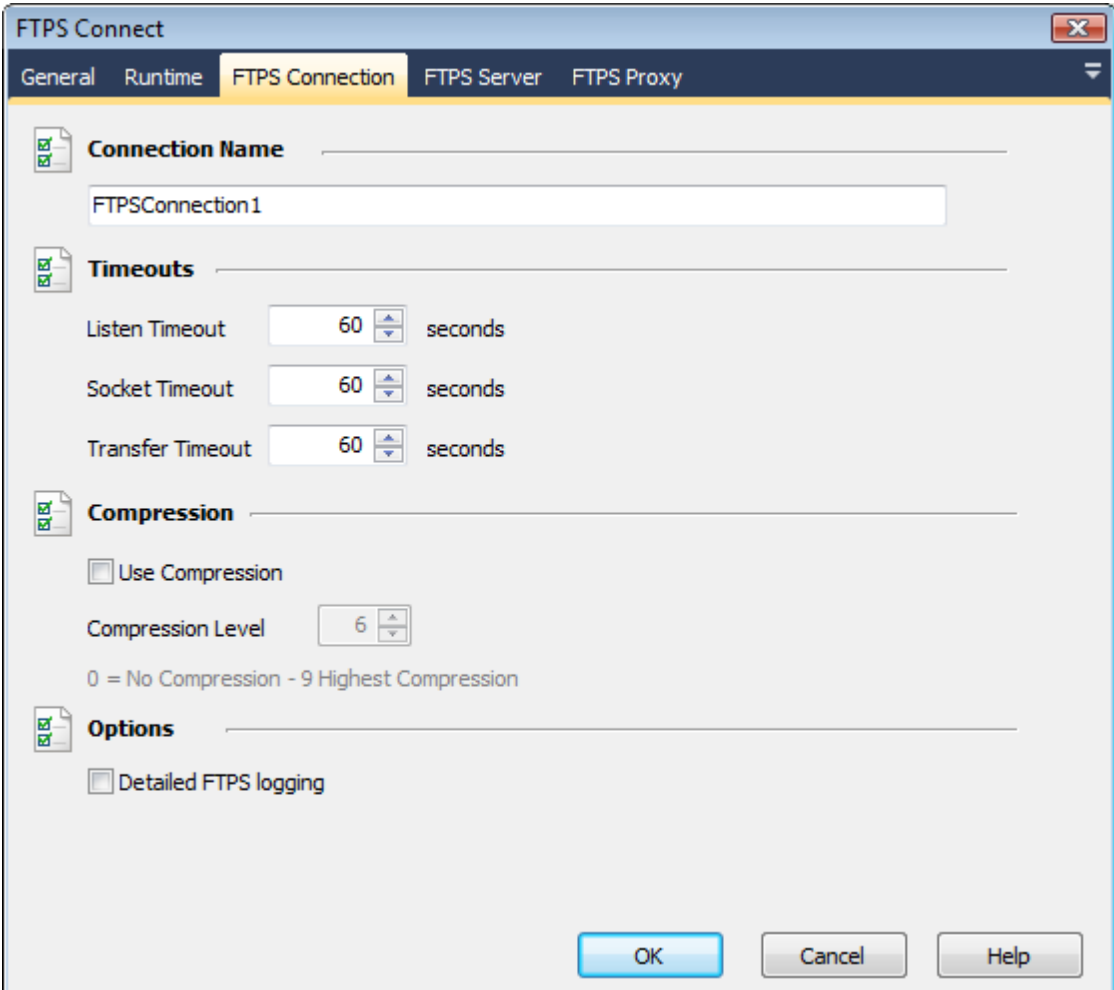
Fail action - Action will fail immediately if connection has been disconnected.

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.

6.19.16.4 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



The screenshot shows the 'FTPS Connect' dialog box with the 'FTPS Connection' tab selected. The dialog has four tabs: 'General', 'Runtime', 'FTPS Connection', 'FTPS Server', and 'FTPS Proxy'. The 'FTPS Connection' tab contains the following sections:

- Connection Name**: A text field containing 'FTPSConnection1'.
- Timeouts**: Three spinners for 'Listen Timeout', 'Socket Timeout', and 'Transfer Timeout', all set to '60' seconds.
- Compression**: A checkbox for 'Use Compression' (unchecked), a 'Compression Level' spinner set to '6', and a note '0 = No Compression - 9 Highest Compression'.
- Options**: A checkbox for 'Detailed FTPS logging' (unchecked).

At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

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

The screenshot shows the 'FTPS Connect' dialog box with the 'FTPS Server' tab selected. The dialog has a title bar with a close button. Below the title bar are five tabs: 'General', 'Runtime', 'FTPS Connection', 'FTPS Server' (selected), and 'FTPS Proxy'. The 'FTPS Server' tab contains the following sections:

- FTPS Server**:
 - ☐ Use Global Server Settings
 - Server: Port:
- Authentication**:
 - User:
 - Password:
 - Certificate:
 - Certificate Password:
 - Authentication:
 - ☐ Passive Mode ☒ Encrypt Data Channel
 - ☐ Implicit SSL ☐ ASCII Transfer Mode
 - ☒ Allow Self-Signed Certificates
- Allowed SSL/TLS Modes**:
 - ☒ SSL 2 ☒ SSL 3 ☐ TLS 1
 - ☐ TLS 1.1 ☐ TLS 1.2

At the bottom right are three buttons: 'OK', 'Cancel', and 'Help'.

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

The screenshot shows the 'FTPS Connect' dialog box with the 'FTPS Proxy' tab selected. The dialog contains the following elements:

- FTPS Proxy** (Section Header)
- ☒ Use Proxy Server
- ☐ Use Global Proxy Settings
- Proxy Host: ProxyHost1
- Proxy Port: 21
- Username: john
- Password: (masked with dots)
- Proxy Type: UserSite (dropdown menu)
- Buttons: OK, Cancel, Help

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.

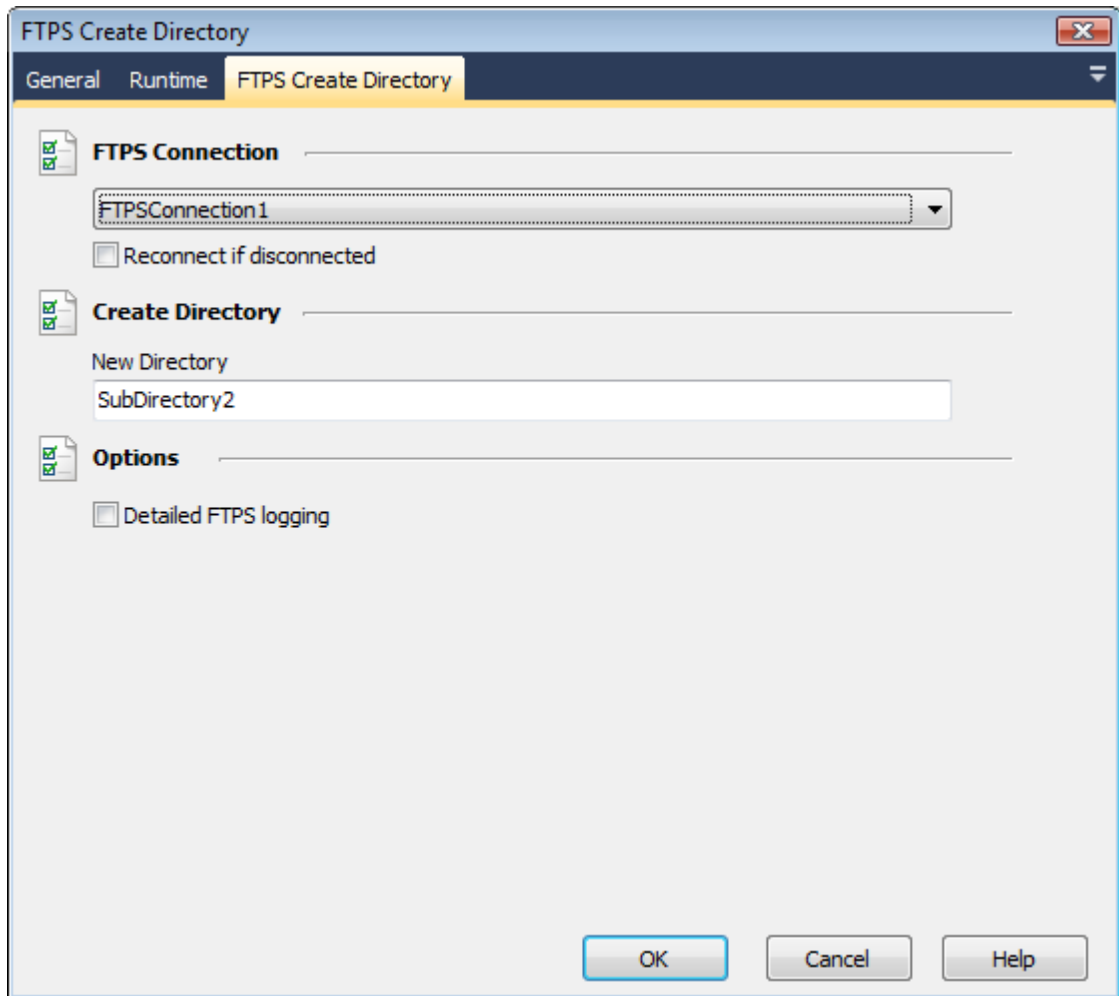
Username - The username for the proxy connection.

Password - The password for the proxy connection.

Proxy Type - The type of proxy server being used.

6.19.16.5 FTPS Create Directory

The FTPS Create Directory action creates a new directory on the FTPS Server.



Connection Name - The name of the FTPS connection to be used for the operation.

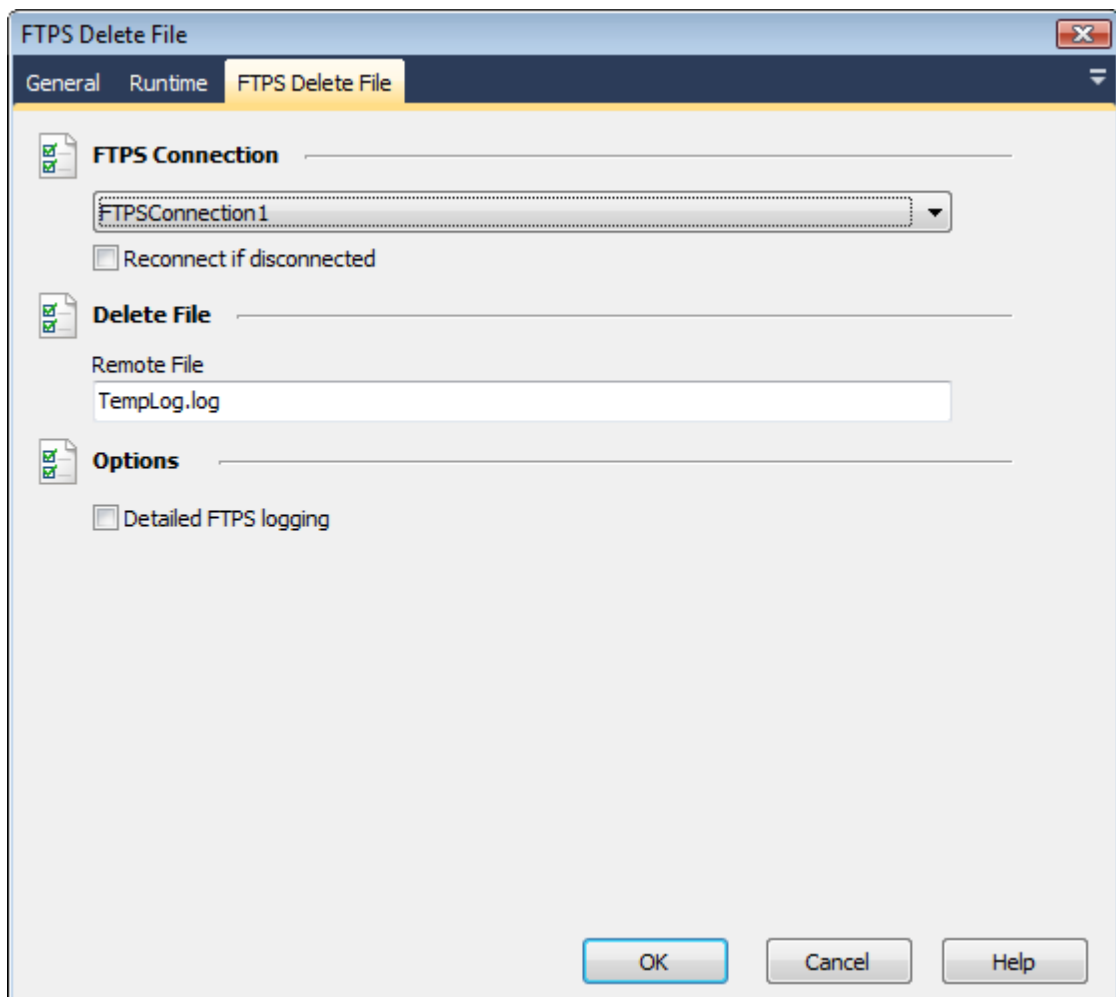
Reconnect if disconnected - If the connection has become disconnected, attempt to reconnect and then execute the action.

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.

6.19.16.6 FTPS Delete File

The FTPS Delete File action allows you to delete a file from the FTPS server.



Connection Name - The name of the FTPS connection to be used for the operation.

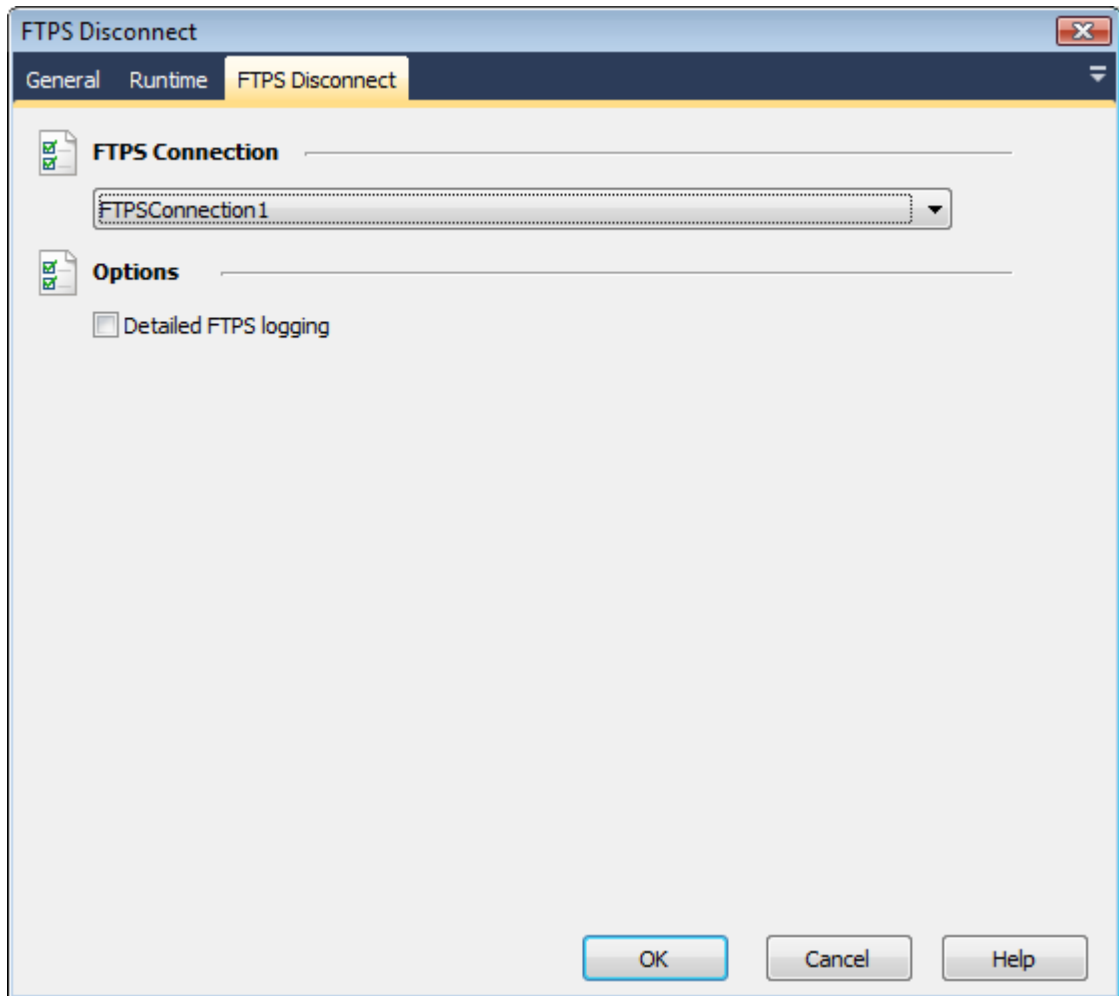
Reconnect if disconnected - If the connection has become disconnected, attempt to reconnect and then execute the action.

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.

6.19.16.7 FTPS Disconnect

The FTPS Disconnect action is used to end the specified FTPS session.



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.

6.19.16.8 FTPS Download File

The FTPS Download File action is used to download a single file from the FTPS server to the local filesystem.

The screenshot shows the 'FTPS Download File' dialog box. It has three tabs: 'General', 'Runtime', and 'FTPS Download File'. The 'FTPS Download File' tab is selected. The dialog is divided into three sections: 'FTPS Connection', 'Download File', and 'Options'. In the 'FTPS Connection' section, there is a dropdown menu showing 'FTPSConnection1' and a checkbox 'Reconnect if disconnected'. In the 'Download File' section, there are two text fields: 'Remote File' with the value 'Log1.txt' and 'Destination Directory' with the value 'C:\Temp\Logs'. In the 'Options' section, there are three checkboxes: 'Specify new file name' (checked), 'Overwrite existing file' (checked), and 'Detailed FTPS logging' (unchecked). Below the 'Specify new file name' checkbox is a text field with the value 'FTPSLog.txt'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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.

6.19.16.9 FTPS Download Files

The FTPS Download Files action can be used to download multiple files from the server to the local file system.

The screenshot shows the 'FTPS Download Files' dialog box. It has three tabs: 'General', 'Runtime', and 'FTPS Download Files'. The 'FTPS Download Files' tab is selected. The dialog is divided into three main sections: 'FTPS Connection', 'Download Files', and 'Options'. In the 'FTPS Connection' section, there is a dropdown menu showing 'FTPSConnection1' and a checkbox for 'Reconnect if disconnected'. The 'Download Files' section includes a 'Remote Directory' field with 'SubDirectory1', a 'File Spec' field with '*.log.txt', and a 'Local Directory' field with 'C:\Temp\Logs'. There is also a checkbox for 'Use RegEx to select files'. The 'Options' section contains several checkboxes: 'Create local directory if missing', 'Ignore individual failures', 'Overwrite existing files', 'Fail if zero files downloaded', and 'Detailed FTPS logging'. At the bottom right, there are 'OK', 'Cancel', and 'Help' buttons.

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.

6.19.16.1(FTPS Get File Date

The FTPS Get File Date allows you to get the modified date of a file on the FTPS server.

The screenshot shows the 'FTPS Get File Date' dialog box. It has three tabs: 'General', 'Runtime', and 'FTPS Get File Date'. The 'FTPS Get File Date' tab is selected. The dialog is divided into three main sections, each with a green checkmark icon and a title bar. The first section, 'FTPS Connection', has a dropdown menu showing 'FTPSConnection1' and a checkbox labeled 'Reconnect if disconnected'. The second section, 'Get File Date', has a text field labeled 'Remote File' containing the text 'ServerLog01.txt'. The third section, 'Options', has a dropdown menu labeled 'Save Date To Variable' with 'MyVariable' selected, a dropdown menu labeled 'Output Format' with 'Short Date' selected, a note 'When output variable is DateTime type set output format as Date Time Object', and a checkbox labeled 'Detailed FTPS logging'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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 reconnect and then execute the action.

Remote File - The name of the file to retrieve the date of.

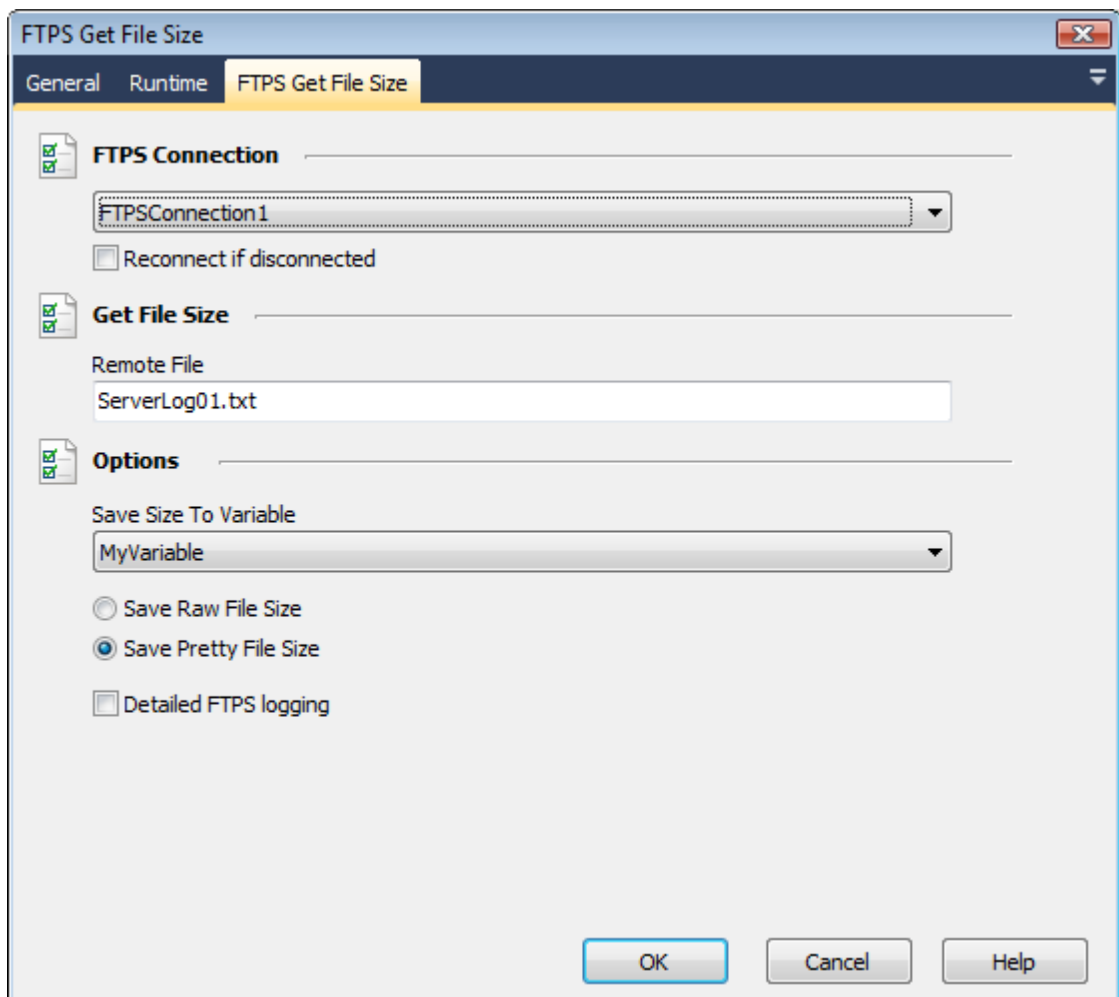
Save Date To Variable - Specify a variable to store the file date in.

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.

6.19.16.1 FTPS Get File Size

The FTPS Get File Size allows you to retrieve the size of a file from the FTPS server.



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 reconnect 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.

6.19.16.1 FTPS Get Remote Checksum

The FTPS Get Remote Checksum action allows you to retrieve the remote checksum for a selected file.

The screenshot shows the 'FTPS Get Remote Checksum' dialog box. It has three tabs: 'General', 'Runtime', and 'FTPS Get Remote Checksum'. The 'FTPS Get Remote Checksum' tab is selected. The dialog is divided into three main sections, each with a green icon and a title bar. The first section, 'FTPS Connection', contains a dropdown menu with 'FTPSConnection1' selected and a checkbox labeled 'Reconnect if disconnected'. The second section, 'Get Remote Checksum', contains a text field for 'Remote File' with 'ServerLog01.txt' entered, a dropdown for 'Checksum Method' with 'CRC' selected, and a note: 'Note: This action will only work where the FTPS Server supports checksum calculation. The action will fail if checksum calculation is not supported.' The third section, 'Options', contains a dropdown for 'Save output to variable' with 'MyVariable' selected and a checkbox labeled 'Detailed FTPS logging'. At the bottom right are three buttons: 'OK', 'Cancel', and 'Help'.

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 reconnect and then execute the action.

Remote File - The name of the file to get the remote checksum of.

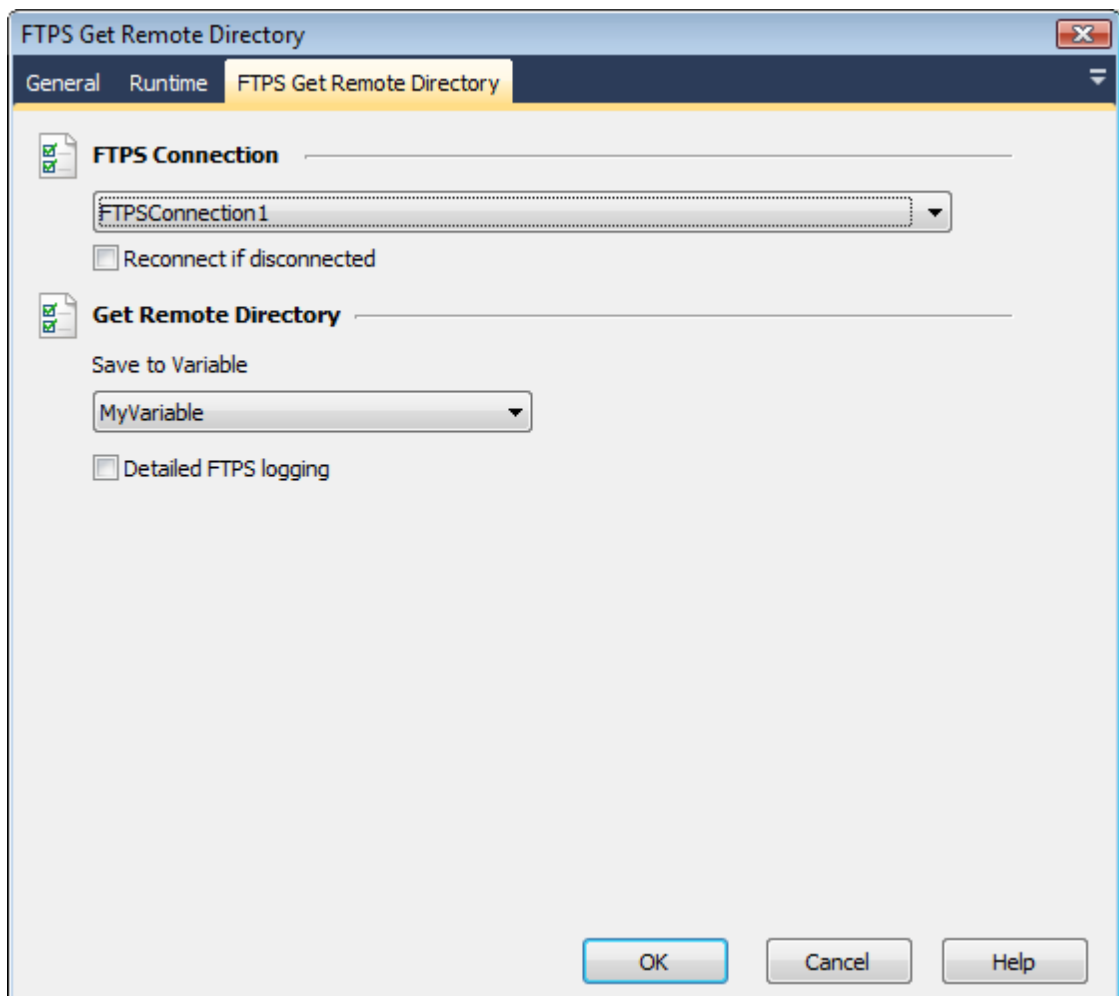
Checksum Method - This is the method used to specify the checksum method (method must be supported by the FTPS server).

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.

6.19.16.1 FTPS Get Remote Directory

The FTPS Get Remote Directory action allows you to get the name of the current directory from the FTPS server.



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.

6.19.16.14 FTPS List Directory

The FTPS List Directory action lists the contents of the specified directory.

The screenshot shows the 'FTPS List Directory' dialog box. It has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'FTPS List Directory' (which is selected and highlighted in yellow), and 'Format Options'. The main area of the dialog is divided into three sections, each with a green checkmark icon and a title: 'FTPS Connection', 'List Directory', and 'Options'. In the 'FTPS Connection' section, there is a dropdown menu currently showing 'FTPSConnection1' and a checkbox labeled 'Reconnect if disconnected'. In the 'List Directory' section, there is a text box labeled 'Remote Directory (relative to current directory)' containing 'SubDirectory1', a text box labeled 'Filename/FileSpec' which is empty, and a note below it saying 'Leave blank to list all items in the current directory'. In the 'Options' section, there is a dropdown menu labeled 'Save Listing To Variable' showing 'MyVariable', and two checkboxes: 'Write listing to log' (which is checked) and 'Detailed FTPS logging' (which is unchecked). At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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.

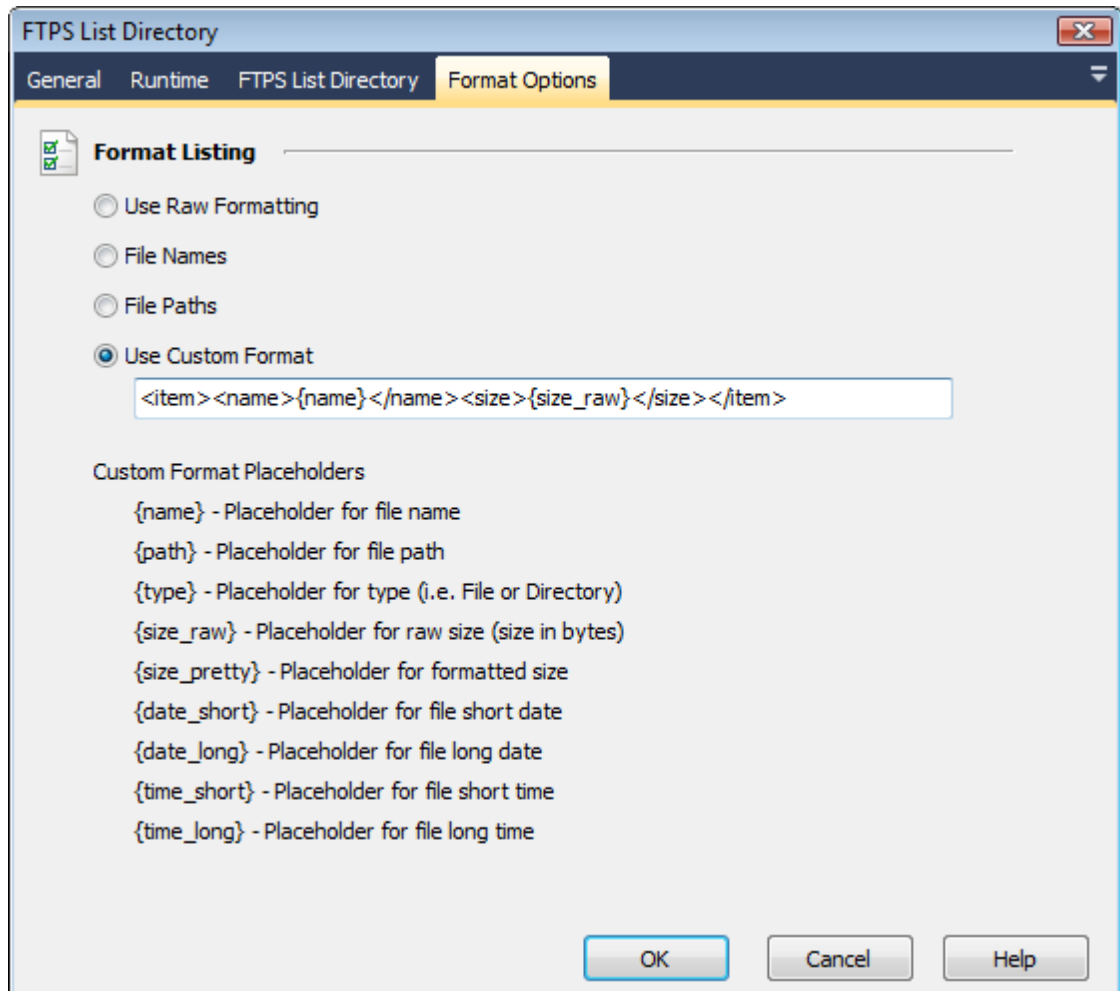
Remote Directory - The name of the remote directory to list the contents of (leave blank to list the current directory).

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 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.

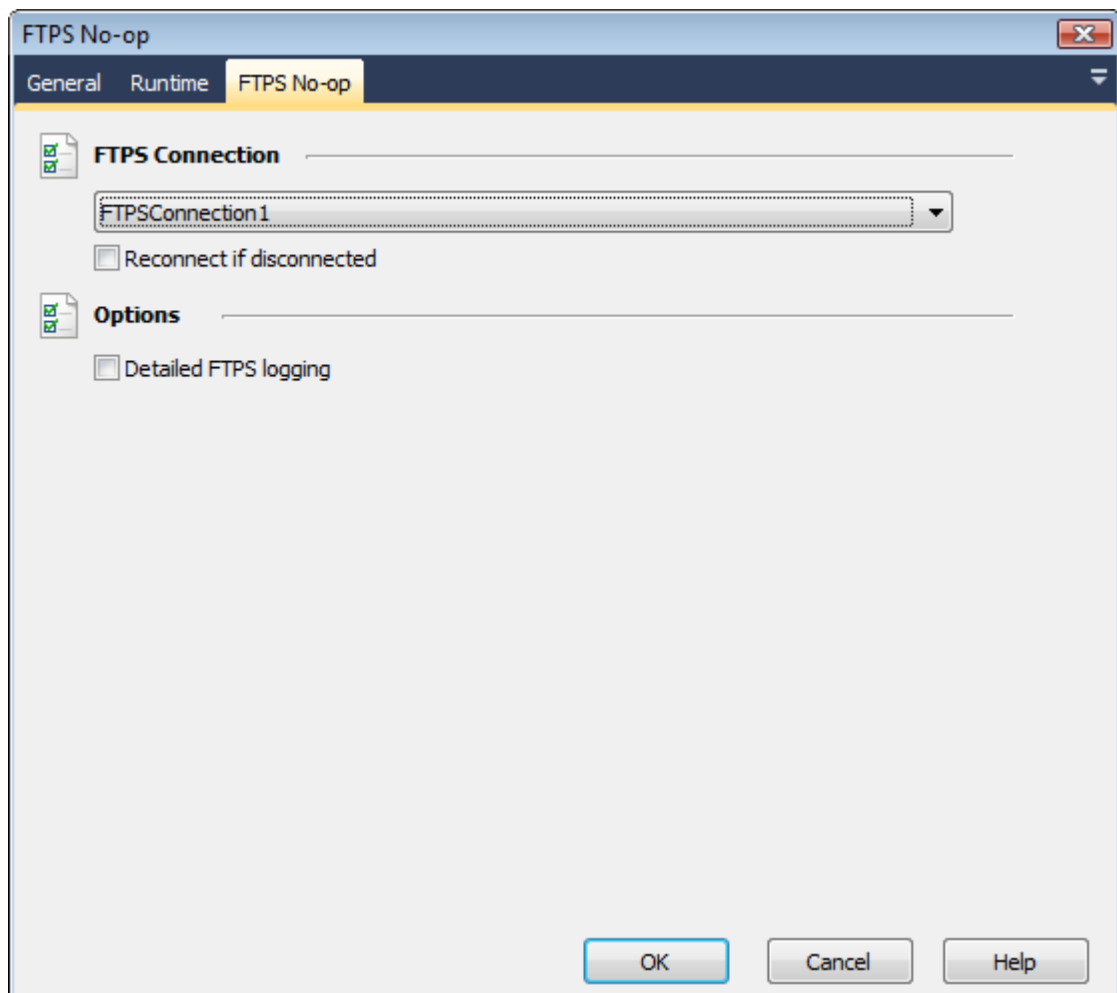
{date_long} - The modified date as a long date.

{time_short} - The modified time as a short time.

{time_long} - The modified time as a long time.

6.19.16.1!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.



Connection Name - The name of the FTPS connection to be used for the operation.

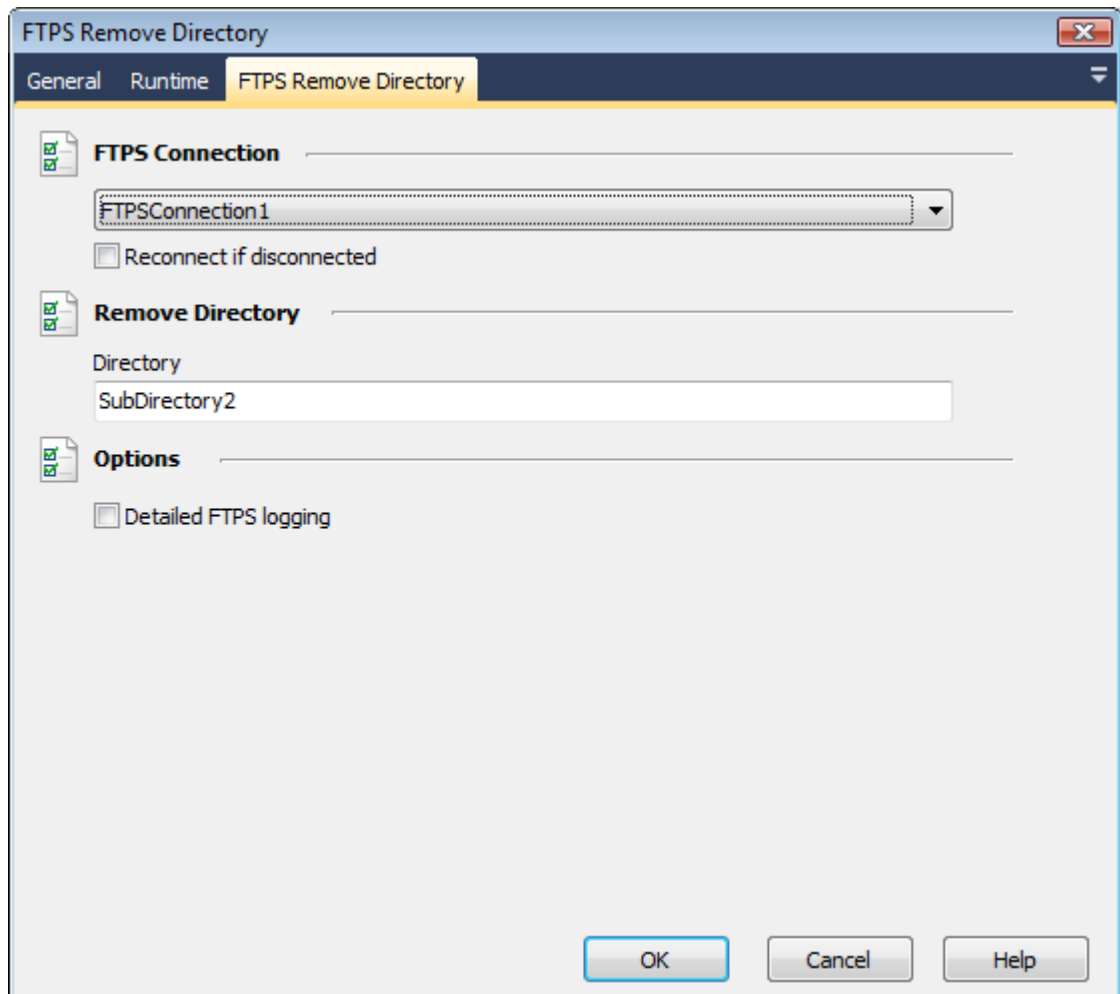
Reconnect if disconnected - If the connection has become disconnected, attempt to re-

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.

6.19.16.1 FTPS Remove Directory

The FTPS Remove Directory action is used to remove a directory from the FTPS server.



Connection Name - The name of the FTPS connection to be used for the operation.

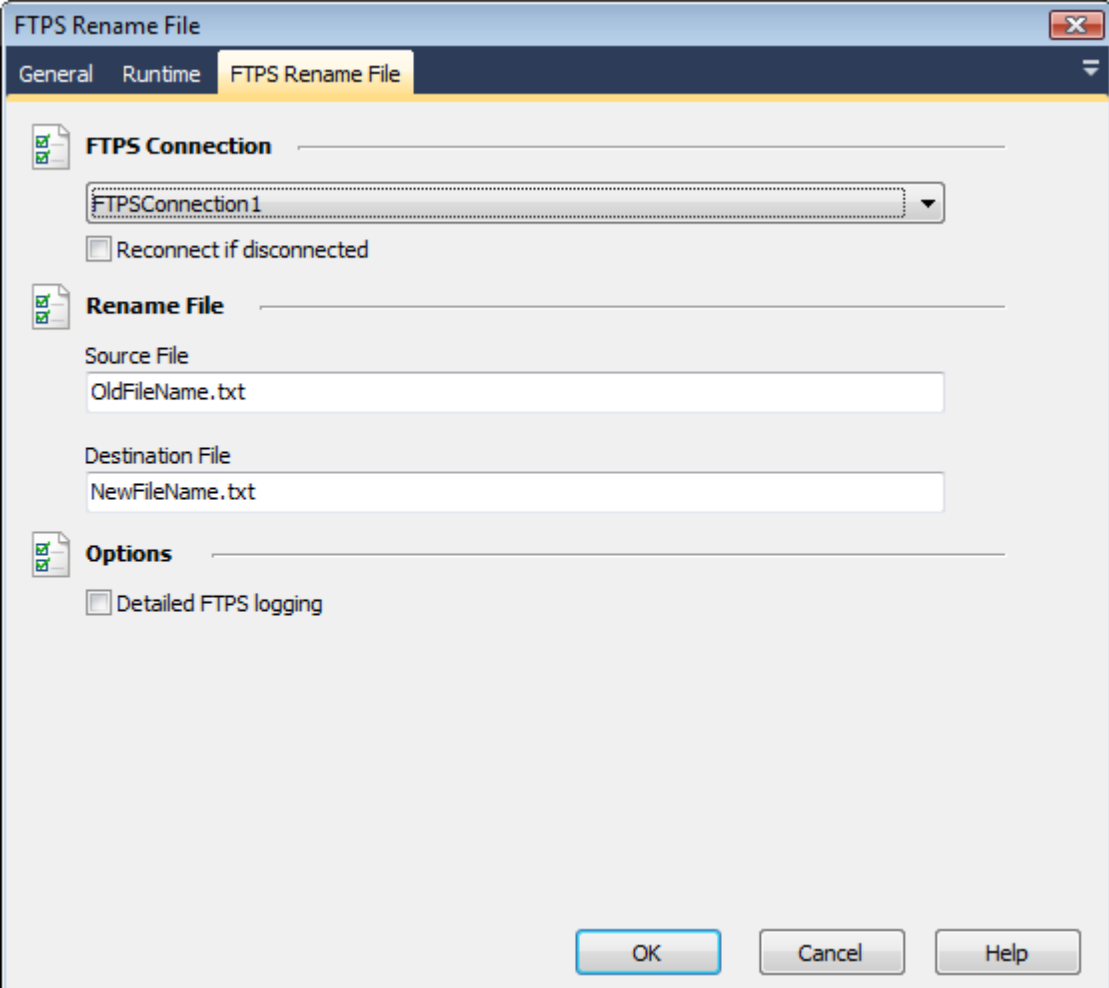
Reconnect if disconnected - If the connection has become disconnected, attempt to re-connect and then execute the action.

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.

6.19.16.1 FTPS Rename File

The FTPS Rename File action is used to rename a file on the FTPS server.



The screenshot shows the 'FTPS Rename File' dialog box with three tabs: 'General', 'Runtime', and 'FTPS Rename File'. The 'FTPS Rename File' tab is active. It contains three sections: 'FTPS Connection', 'Rename File', and 'Options'. The 'FTPS Connection' section has a dropdown menu set to 'FTPSConnection1' and a checkbox for 'Reconnect if disconnected'. The 'Rename File' section has two text fields: 'Source File' with 'OldFileName.txt' and 'Destination File' with 'NewFileName.txt'. The 'Options' section has a checkbox for 'Detailed FTPS logging'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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 reconnect 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.

6.19.16.1 FTPS Send Command

The FTPS Send Command action allows you to send a custom command to the FTPS server.

The screenshot shows the 'FTPS Send Command' dialog box with the following configuration:

- FTPS Connection:** A dropdown menu showing 'FTPSConnection1'. Below it is an unchecked checkbox for 'Reconnect if disconnected'.
- Send Command:** A text field for 'Command' containing 'STAT'. Below it is a text field for 'Accepted Response Codes' containing '211'. A note below this field says 'Seperate multiple codes with a comma'.
- Options:** A dropdown menu for 'Save output to variable' showing 'MyVariable'. Below it is an unchecked checkbox for 'Detailed FTPS logging'.

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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.

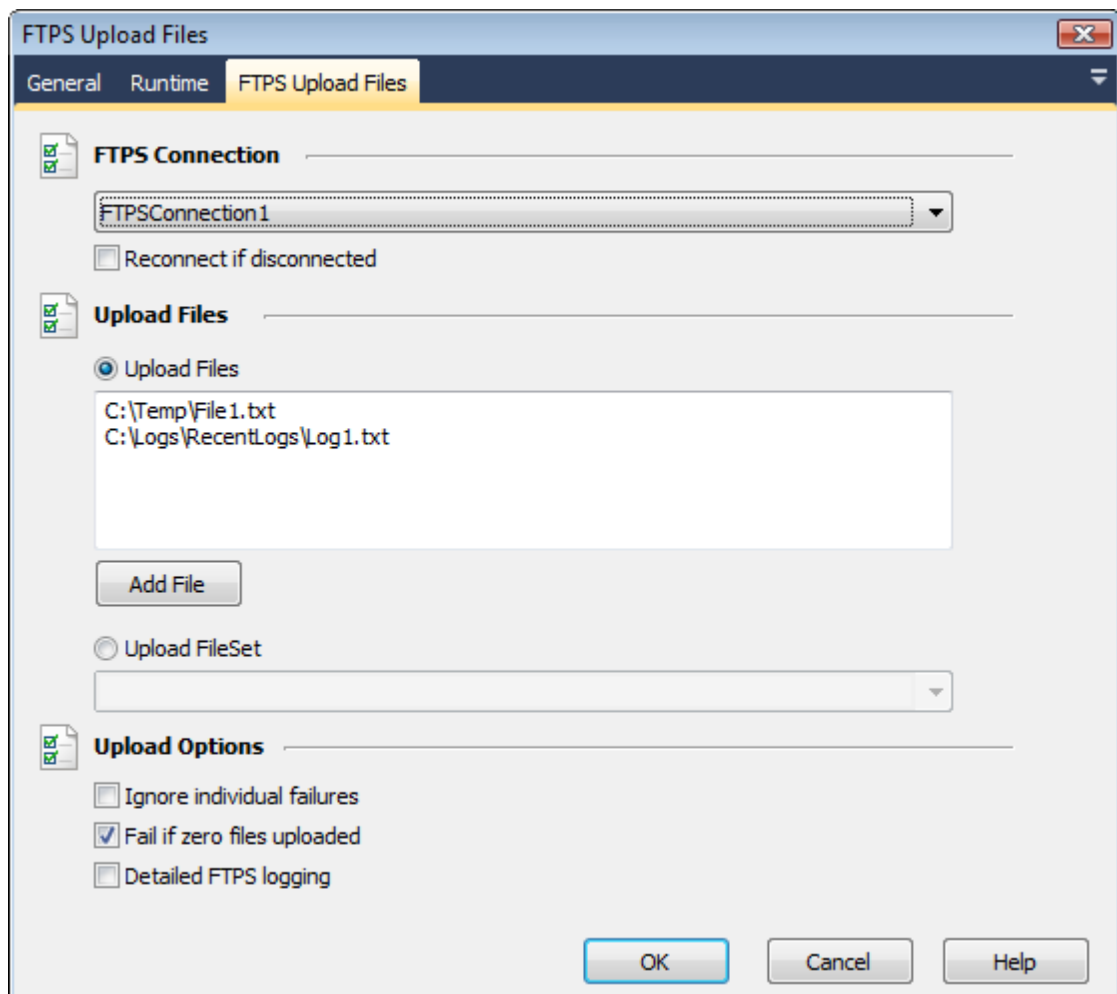
Using the detailed logging option will help you determine the accepted response code that you will need to use for this action. The accepted response code will vary depending on the command being sent.

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.

6.19.16.1!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.



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.

6.19.17 EC2 Actions

The EC2 actions allow you to manage your EC2 instances from your project.

6.19.17.1 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.

The screenshot shows the 'EC2 Get Instance Properties' dialog box with the 'Settings' tab selected. The dialog has three tabs: 'General', 'Runtime', and 'Settings'. The 'Settings' tab contains the following sections:

- AWS Connection:** Includes fields for 'AWS Access Key' (containing 'AKIBJYEXAMPLEV48GBDQ') and 'AWS Secret Key' (masked with dots). There is a checkbox for 'Use Global Values' which is currently unchecked.
- Options:** Includes a 'Return Property' dropdown menu set to 'ID' and an 'Output Variable' dropdown menu set to 'output1'.
- Filters:** There are four filter sections labeled 'Filter 1' through 'Filter 4'. 'Filter 1' has 'Name' set to 'instance-state-name' and 'Value' set to 'running'. 'Filter 2', 'Filter 3', and 'Filter 4' have empty 'Name' and 'Value' fields.

Below the filters, it says '(all filters are optional)'. At the bottom right, there are 'OK', 'Cancel', and 'Help' buttons.

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 separated 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/APIReference/index.html?ApiReference-query-DescribeInstances.html>

6.19.17.2 EC2 Change Instance States

This action enables you to change the state of one or more instances.

The screenshot shows a dialog box titled "EC2 Change Instance States" with three tabs: "General", "Runtime", and "Settings". The "Settings" tab is selected. It contains two sections: "AWS Connection" and "Options".

AWS Connection

- AWS Access Key:
- AWS Secret Key:
- ☐ Use Global Values

Options

- Instance ID(s): (comma seperated)
- State Change:

At the bottom right are three buttons: "OK", "Cancel", and "Help".

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

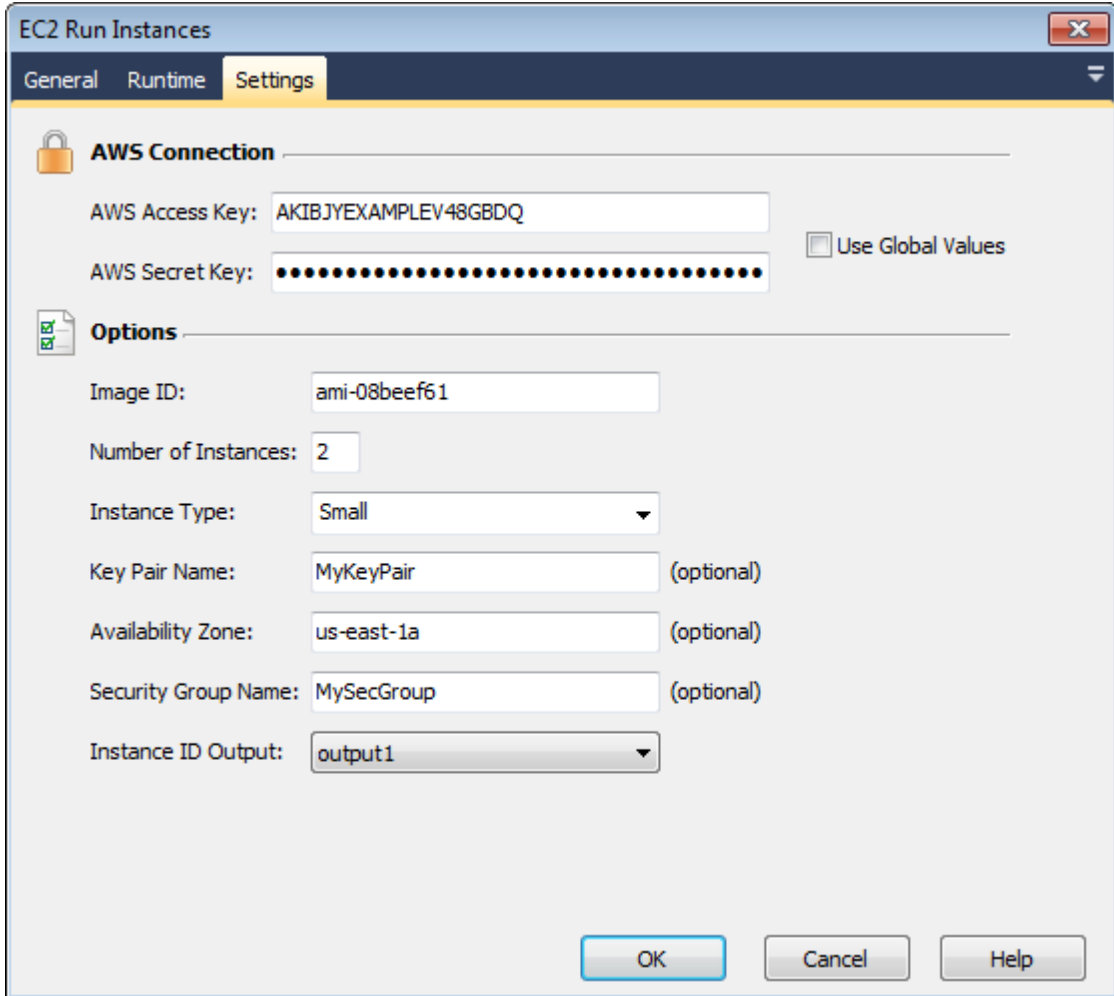
A comma seperated list of the IDs of the instances to change state.

State Change

The state change to apply to the specified instances. Choose between start, stop, and reboot.

6.19.17.3 EC2 Run Instances

This action enables you to run (create) one or more instances.



The screenshot shows the 'EC2 Run Instances' dialog box with the 'Settings' tab selected. The 'AWS Connection' section includes fields for 'AWS Access Key' (AKIBJYEXAMPLEV48GBDQ) and 'AWS Secret Key' (masked with dots), with a 'Use Global Values' checkbox. The 'Options' section includes fields for 'Image ID' (ami-08beef61), 'Number of Instances' (2), 'Instance Type' (Small), 'Key Pair Name' (MyKeyPair), 'Availability Zone' (us-east-1a), 'Security Group Name' (MySecGroup), and 'Instance ID Output' (output1). The 'Key Pair Name', 'Availability Zone', and 'Security Group Name' fields are marked as optional. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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 instances 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

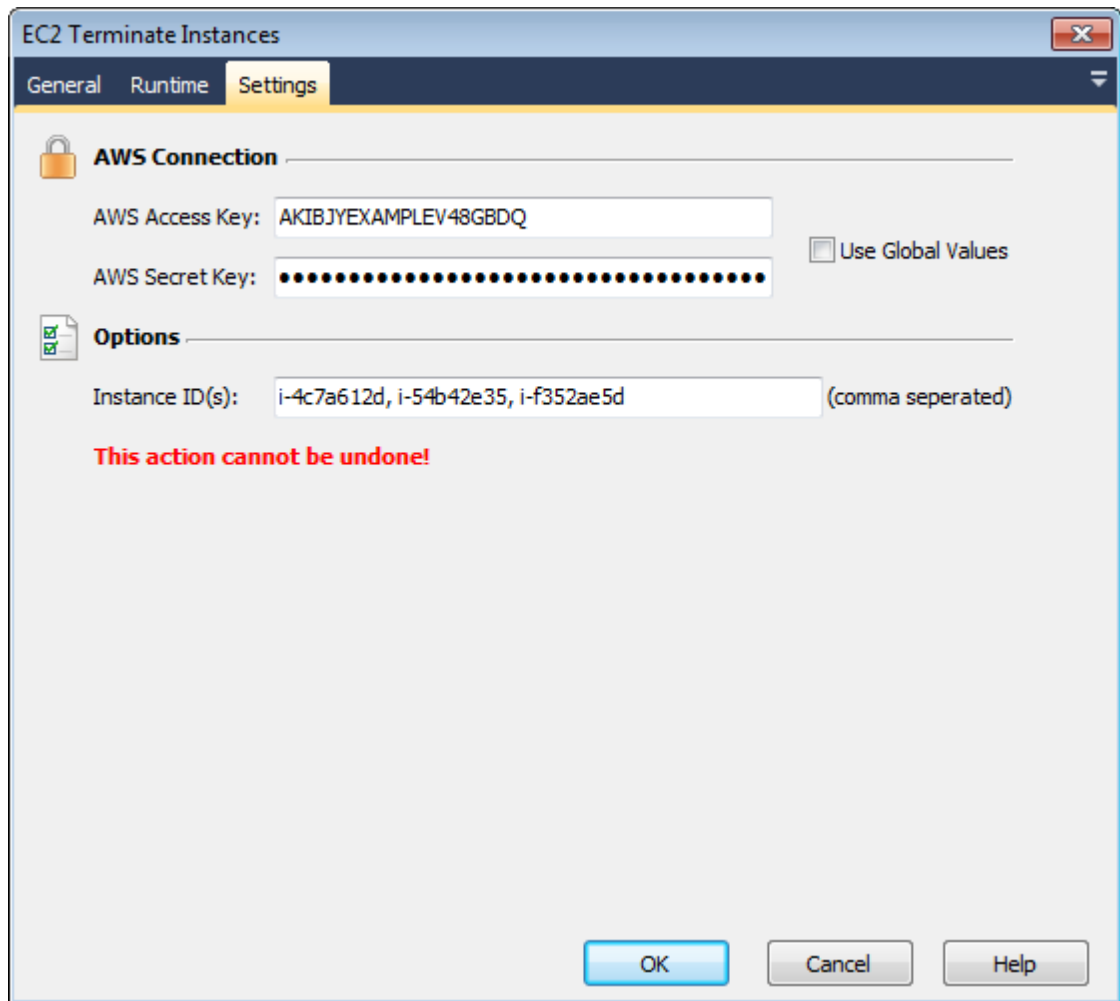
Optional. The name of a security group you have previously created via the AWS web interface. If not specified, your default security group will be used.

Instance ID Output

The variable to return the ID(s) of the newly created instances to. Will return a comma separated list if more than one instance was created.

6.19.17.4 EC2 Terminate Instance

This action enables you to terminate one or more instances. Note that this action is permanent and cannot be undone.



AWS 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 separated list of the IDs of the instances to terminate.

6.19.17.5 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.

The screenshot shows the 'EC2 Get Image Properties' dialog box with the 'Settings' tab selected. The 'AWS Connection' section has a lock icon and fields for 'AWS Access Key' (AKIBJYEXAMPLEV48GBDQ) and 'AWS Secret Key' (masked with dots). A 'Use Global Values' checkbox is present. The 'Options' section has a document icon and dropdowns for 'Return Property' (ID) and 'Output Variable' (output1). Below are four filter sections: Filter 1 (Name: owner-id, Value: 272153855122), Filter 2, Filter 3, and Filter 4, each with 'Name' and 'Value' fields. A note states '(all filters are optional, however without them many images will be returned)'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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 separated 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/APIReference/index.html?ApiReference-query-DescribeImages.html>

6.19.17.6 EC2 Create Image

This action enables you to create an image from an instance.

The screenshot shows the 'EC2 Create Image' dialog box with the 'Settings' tab selected. The 'AWS Connection' section includes fields for 'AWS Access Key' (AKIBJYEXAMPLEV48GBDQ) and 'AWS Secret Key' (masked with dots), with a 'Use Global Values' checkbox. The 'Options' section includes fields for 'Instance ID' (i-4c7a612d), 'Name' (Example Image), 'Description' (This is an example image), a 'Do not shut down instance' checkbox, and an 'Image ID Output' dropdown menu set to 'output1'. The dialog has 'OK', 'Cancel', and 'Help' buttons at the bottom.

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

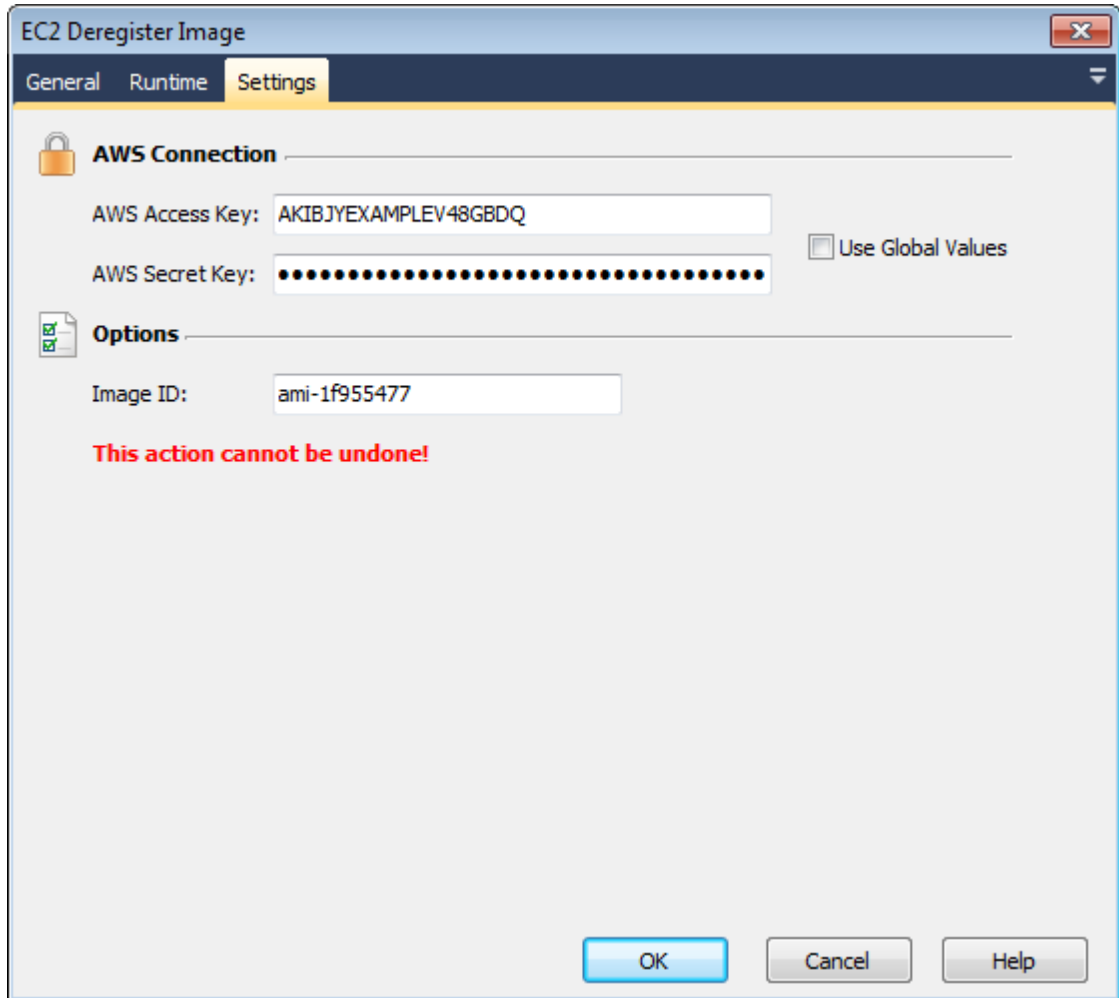
Checking this will create the image without stopping the instance first. This is not recommended as it may cause I/O errors in the created image.

Image ID Output

The variable to return the ID of the newly created image to.

6.19.17.7 EC2 Deregister Image

This action enables you to deregister (delete) an image. Please note that this action is permanent and 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.

6.19.17.8 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.

The screenshot shows the 'EC2 Get Volume Properties' dialog box with the 'Settings' tab selected. The 'AWS Connection' section includes fields for 'AWS Access Key' (AKIBJYEXAMPLEV48GBDQ) and 'AWS Secret Key' (masked with dots), with a 'Use Global Values' checkbox. The 'Options' section includes 'Return Property' (ID) and 'Output Variable' (output1). There are four filter sections (Filter 1 to Filter 4), each with 'Name' and 'Value' fields. Filter 1 has 'Name: size' and 'Value: 10'. The other filters are empty. A note at the bottom states '(all filters are optional)'. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

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 separated list of properties will be returned.

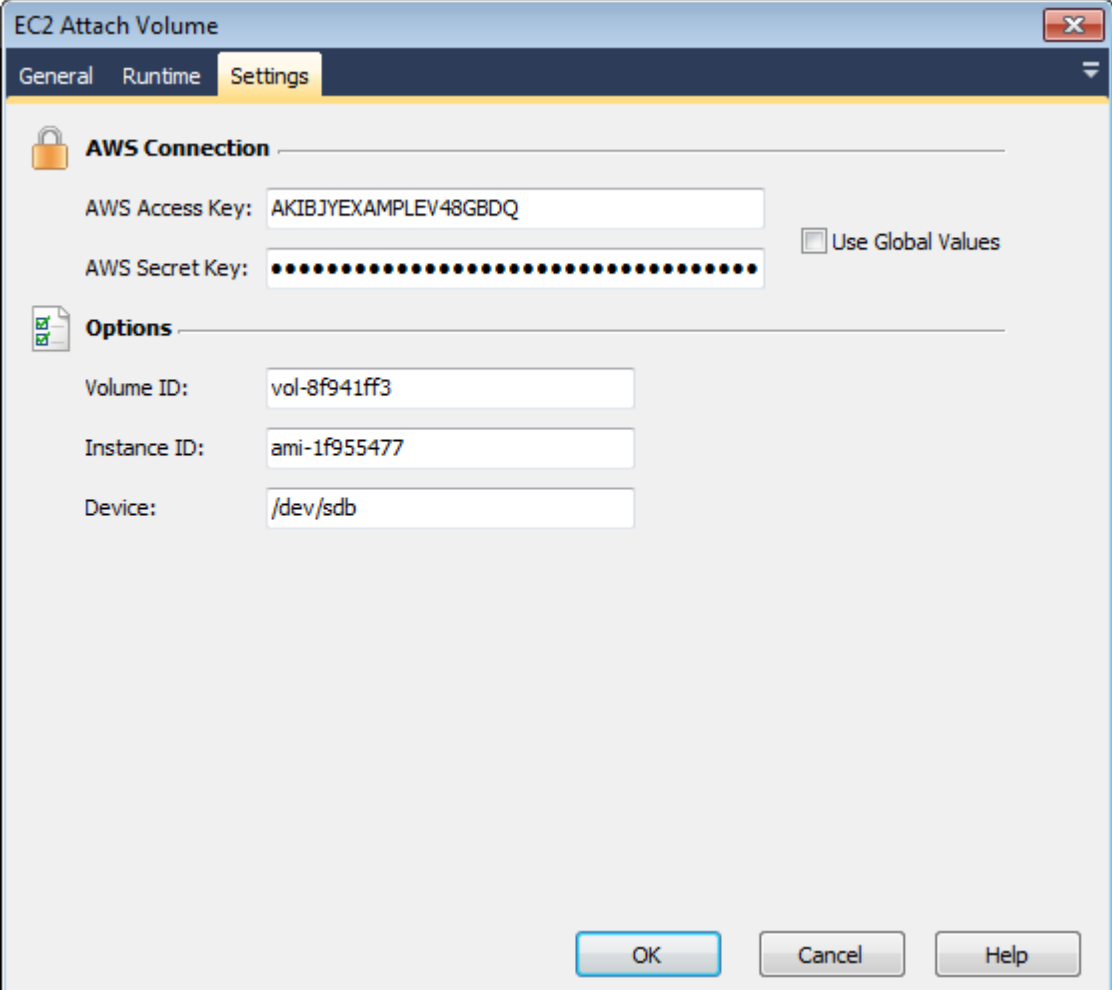
Filters

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/APIReference/index.html?ApiReference-query-DescribeVolumes.html>

6.19.17.9 EC2 Attach Volume

This action enables you to attach a volume to an instance.



The screenshot shows the 'EC2 Attach Volume' dialog box with the 'Settings' tab selected. The dialog has three tabs: 'General', 'Runtime', and 'Settings'. Under the 'Settings' tab, there are two main sections: 'AWS Connection' and 'Options'. The 'AWS Connection' section includes fields for 'AWS Access Key' (containing 'AKIBJYEXAMPLEV48GBDQ') and 'AWS Secret Key' (masked with dots), and a checkbox for 'Use Global Values' which is unchecked. The 'Options' section includes fields for 'Volume ID' (containing 'vol-8f941ff3'), 'Instance ID' (containing 'ami-1f955477'), and 'Device' (containing '/dev/sdb'). At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

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

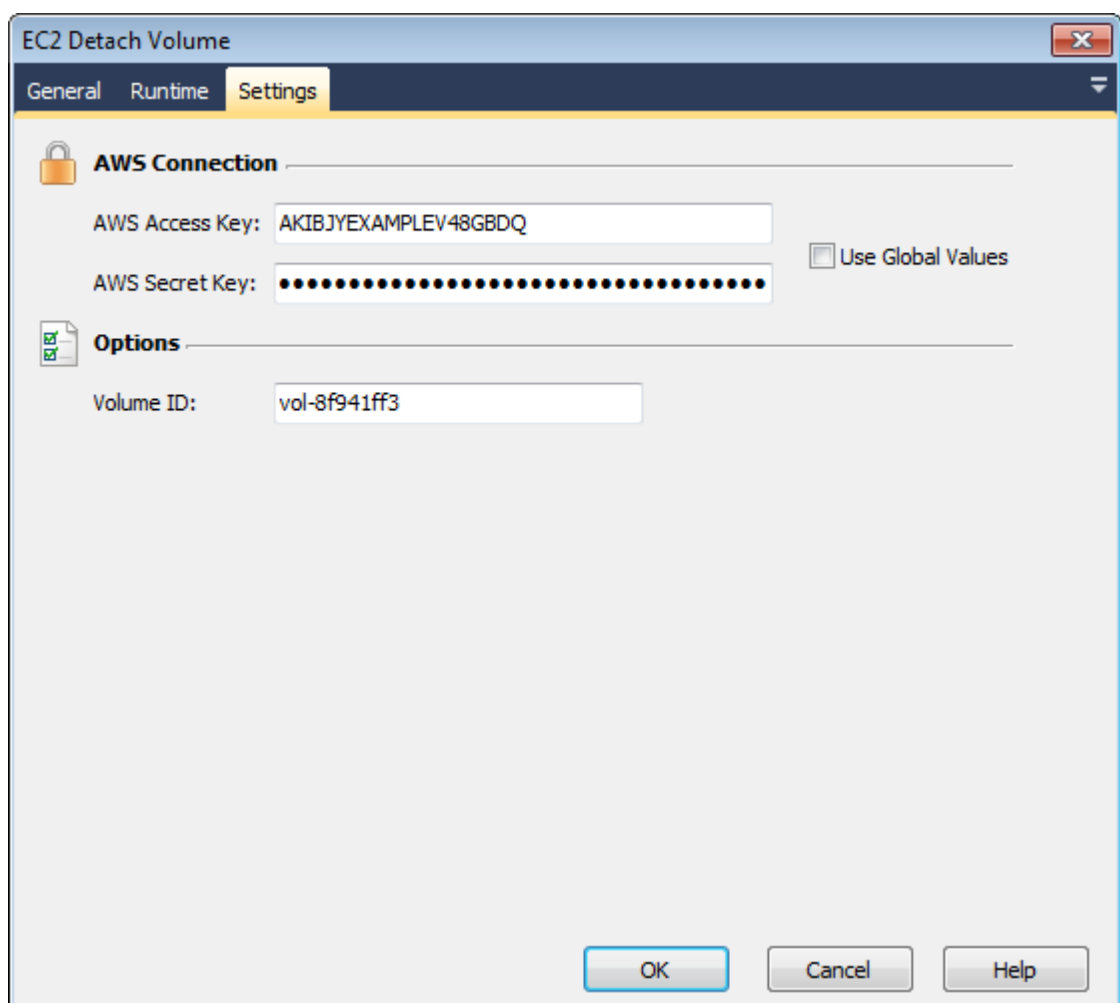
The ID of the instance to attach to.

Device

The device to expose the volume to the instance as. For example, /dev/sdh, or xvdh.

6.19.17.1 EC2 Detach Volume

This action enables you to detach a volume from any instances it is attached to.

**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

6.19.17.1 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.

The screenshot shows the 'EC2 Create Volume' dialog box with the 'Settings' tab selected. The 'AWS Connection' section has fields for 'AWS Access Key' (AKIBJYEXAMPLEV48GBDQ) and 'AWS Secret Key' (masked with dots), with a 'Use Global Values' checkbox. The 'Options' section has radio buttons for 'Size' (selected, 10 GiB) and 'Snapshot ID' (empty), with a '(choose one)' label. Below these are fields for 'Availability Zone' (us-east-1a) and 'Volume ID Output' (output1). At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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

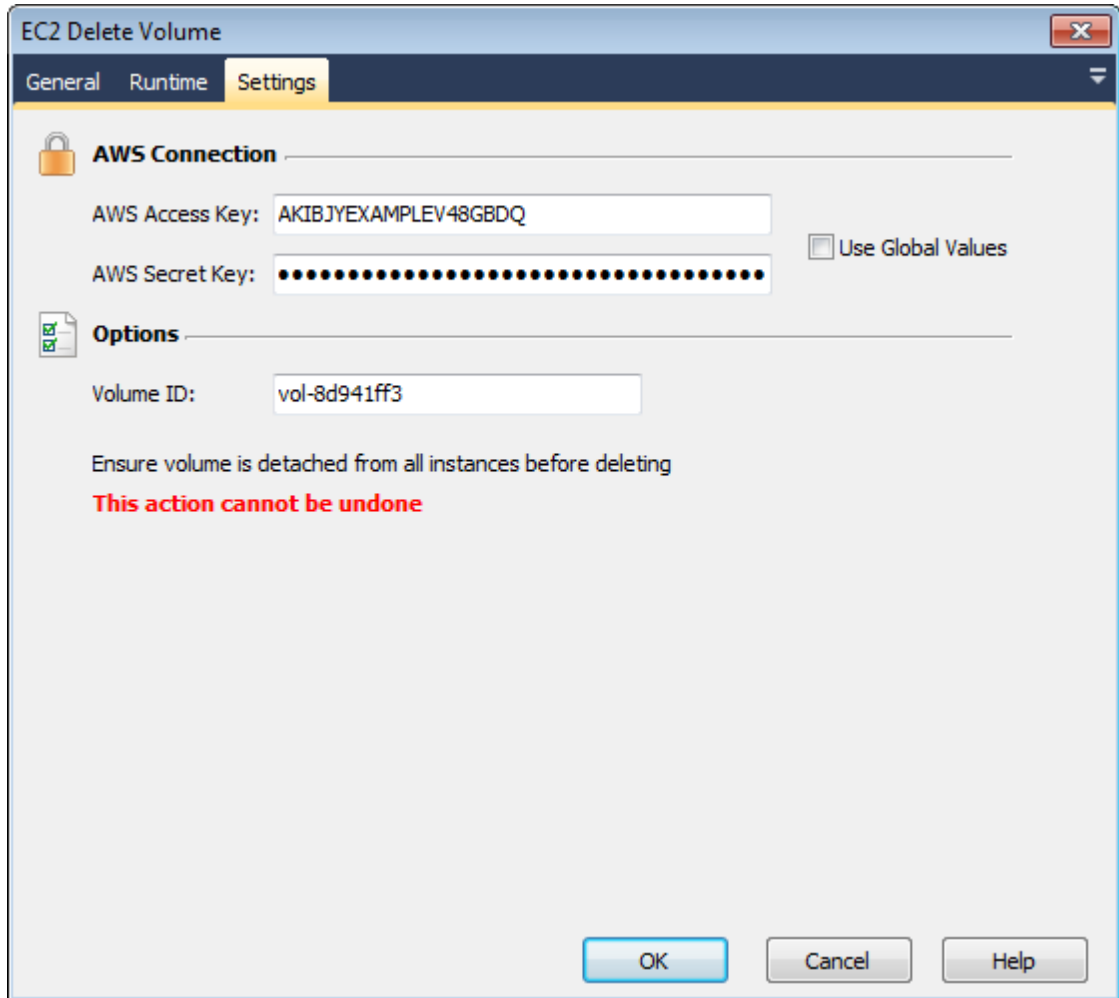
The availability zone for the volume to be created in. Use the Get Availability Zones action to get a list of availability zones that are available to you.

Volume ID Output

The variable to return the ID of the newly created volume to.

6.19.17.1 EC2 Delete Volume

This action enables you to delete a volume. Note that this action is permanent and 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.

6.19.17.1 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.

The screenshot shows the 'EC2 Get Snapshot Properties' dialog box with the 'Settings' tab selected. The 'AWS Connection' section includes fields for 'AWS Access Key' (AKIBJYEXAMPLEV48GBDQ) and 'AWS Secret Key' (masked with dots), with a 'Use Global Values' checkbox. The 'Options' section features a 'Return Property' dropdown set to 'ID' and an 'Output Variable' dropdown set to 'output1'. Below these are four filter sections: 'Filter 1' (Name: owner-id, Value: 272153855122), 'Filter 2' (Name: , Value:), 'Filter 3' (Name: , Value:), and 'Filter 4' (Name: , Value:). A note states '(all filters are optional, however many results may be returned)'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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 separated 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/APIReference/index.html?ApiReference-query-DescribeSnapshots.html>

6.19.17.14 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.

The screenshot shows the 'EC2 Create Snapshot' dialog box with the 'Settings' tab selected. The dialog has three tabs: 'General', 'Runtime', and 'Settings'. The 'Settings' tab contains two sections: 'AWS Connection' and 'Options'. The 'AWS Connection' section has a lock icon, a text field for 'AWS Access Key' containing 'AKIBJYEXAMPLEV48GBDQ', a text field for 'AWS Secret Key' filled with dots, and a checkbox for 'Use Global Values' which is unchecked. The 'Options' section has a document icon, a text field for 'Volume ID' containing 'vol-8f941ff3', a text field for 'Description' containing 'This is an example snapshot.' with '(optional)' to its right, and a dropdown menu for 'Snapshot ID Out' showing 'output1'. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

Volume ID

The volume ID to create a snapshot of.

Description

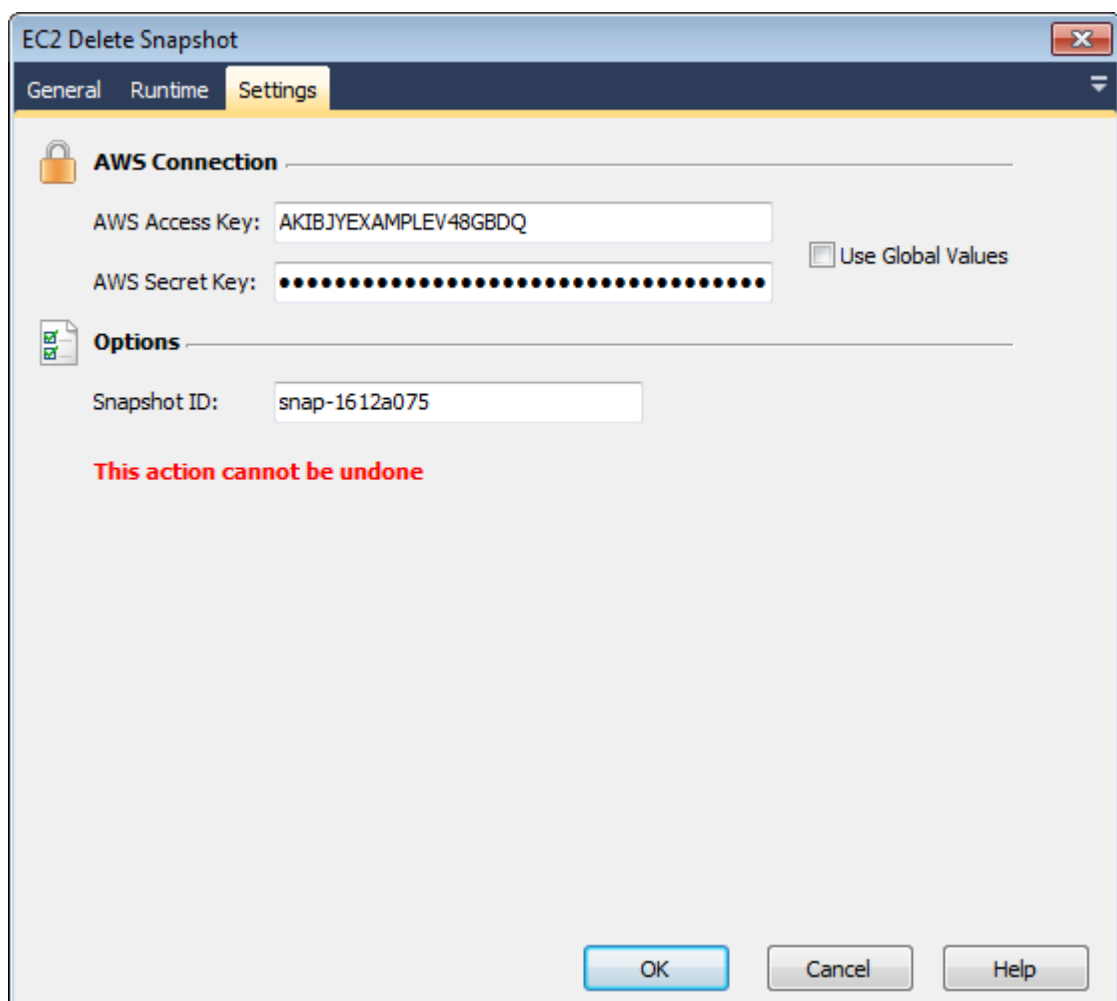
Optional. A description of the snapshot, visible in the AWS web interface, and available through the Get Snapshot Properties method.

Snapshot ID Output

The variable to return the ID of the newly created snapshot to.

6.19.17.1!EC2 Delete Snapshot

This action enables you to delete a snapshot. Note that this action is permanent and 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.

6.19.17.1 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.

The screenshot shows the 'EC2 Get Availability Zones' dialog box with the 'Settings' tab selected. The dialog has three tabs: 'General', 'Runtime', and 'Settings'. The 'AWS Connection' section includes fields for 'AWS Access Key' (containing 'AKIBJYEXAMPLEV48GBDQ') and 'AWS Secret Key' (masked with dots), and a checkbox for 'Use Global Values'. The 'Options' section includes a 'Zone Names Out' dropdown menu set to 'output1', and four filter sections (Filter 1 through Filter 4). Filter 1 has 'Name' set to 'state' and 'Value' set to 'available'. Filters 2, 3, and 4 have empty 'Name' and 'Value' fields. A note at the bottom states '(all filters are optional)'. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

Section	Field	Value
AWS Connection	AWS Access Key	AKIBJYEXAMPLEV48GBDQ
	AWS Secret Key
Options	Zone Names Out	output1
	Filter 1 Name	state
	Filter 1 Value	available
	Filter 2 Name	
	Filter 2 Value	
Filter 3 Name		
Filter 3 Value		
Filter 4 Name		
Filter 4 Value		

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 separated 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/APIReference/index.html?ApiReference-query-DescribeAvailabilityZones.html>

6.20 Internet Information Services (IIS)

The IIS actions allow you to administer Microsoft IIS 5 and IIS 6 servers from your build. 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.

6.20.1 IIS 5

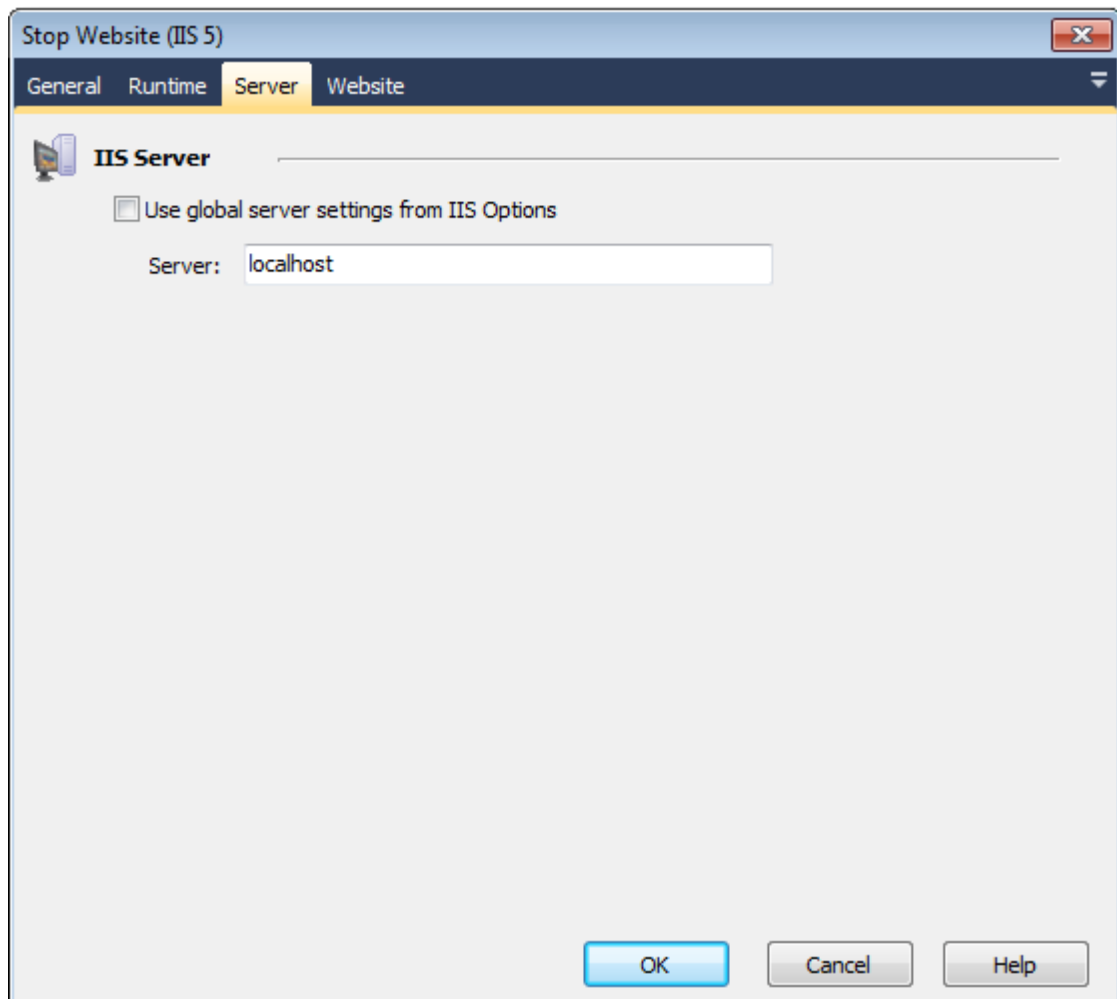
The IIS 5 actions use the ADSI interface to administer features of IIS.

These actions also work with IIS 6.0, but some IIS 6.0 administration features (such as user/password authentication) are not available.

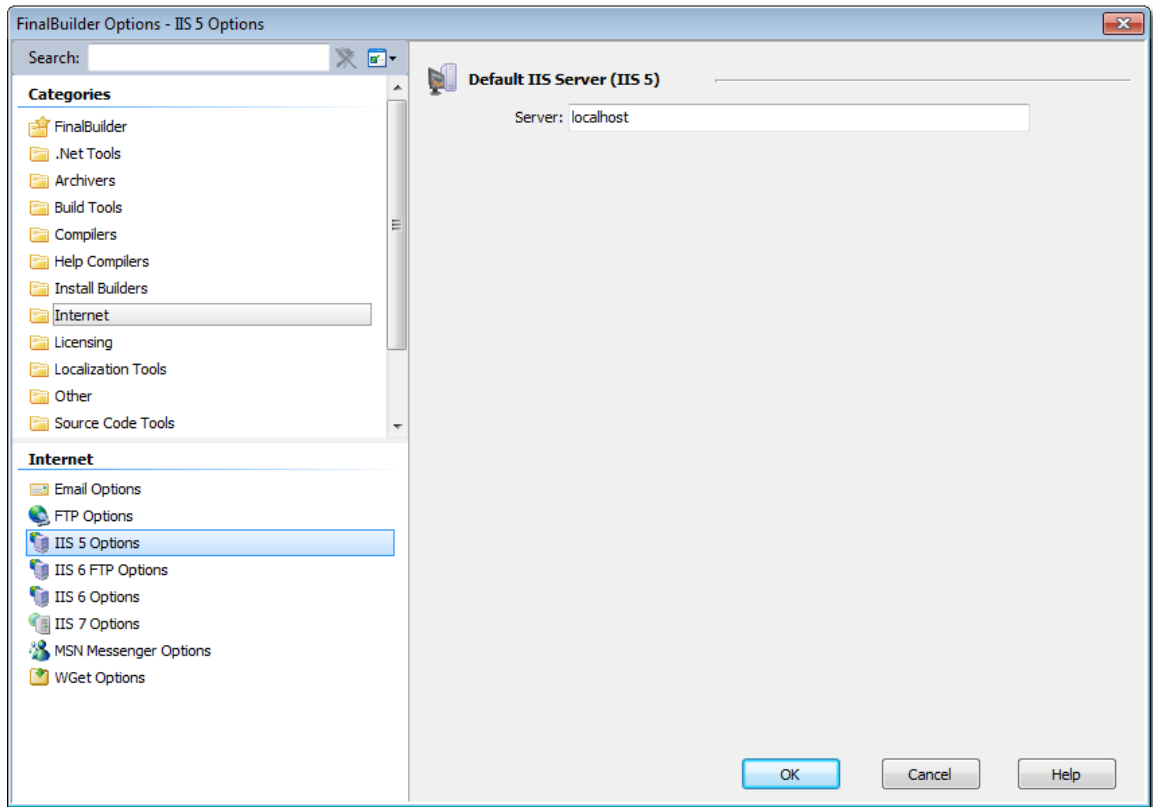
6.20.1.1 Server Selection

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.

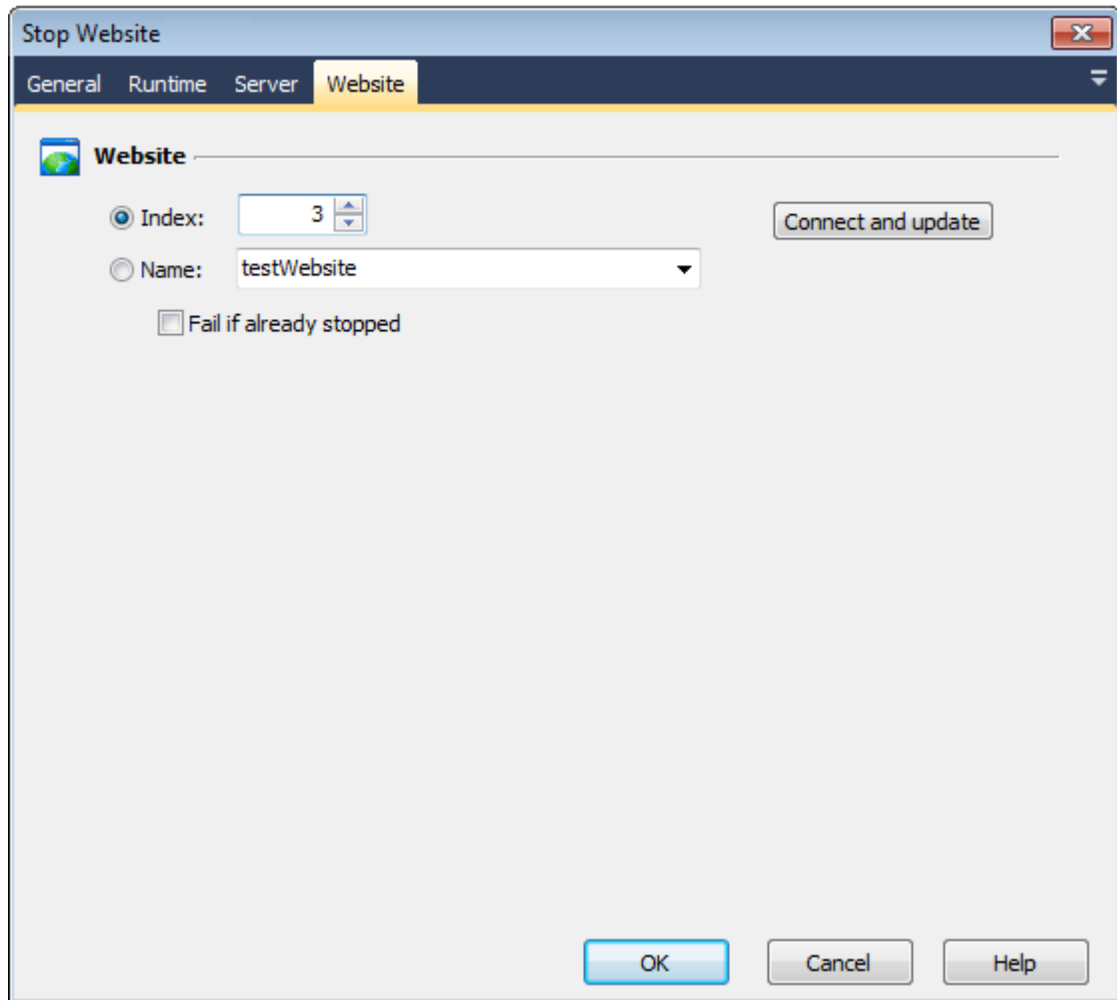


To select the global server name, choose Options from the Tools menu and then click on the IIS 5 Options tab under 'Internet'.



6.20.1.2 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.)



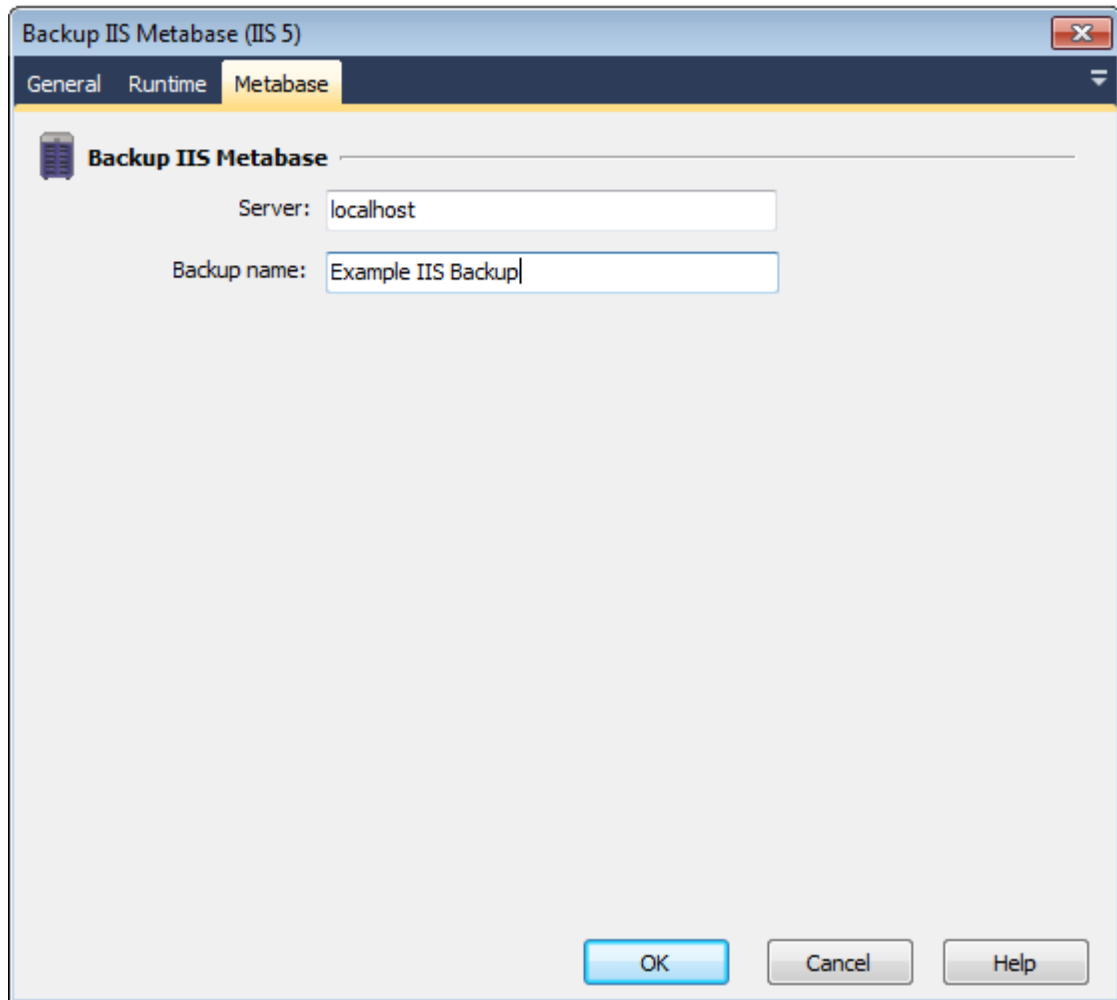
To connect to the server and update the list of available web site names, click "Connect and update." Alternatively, just type a name into the combo box.

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.

6.20.1.3 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**

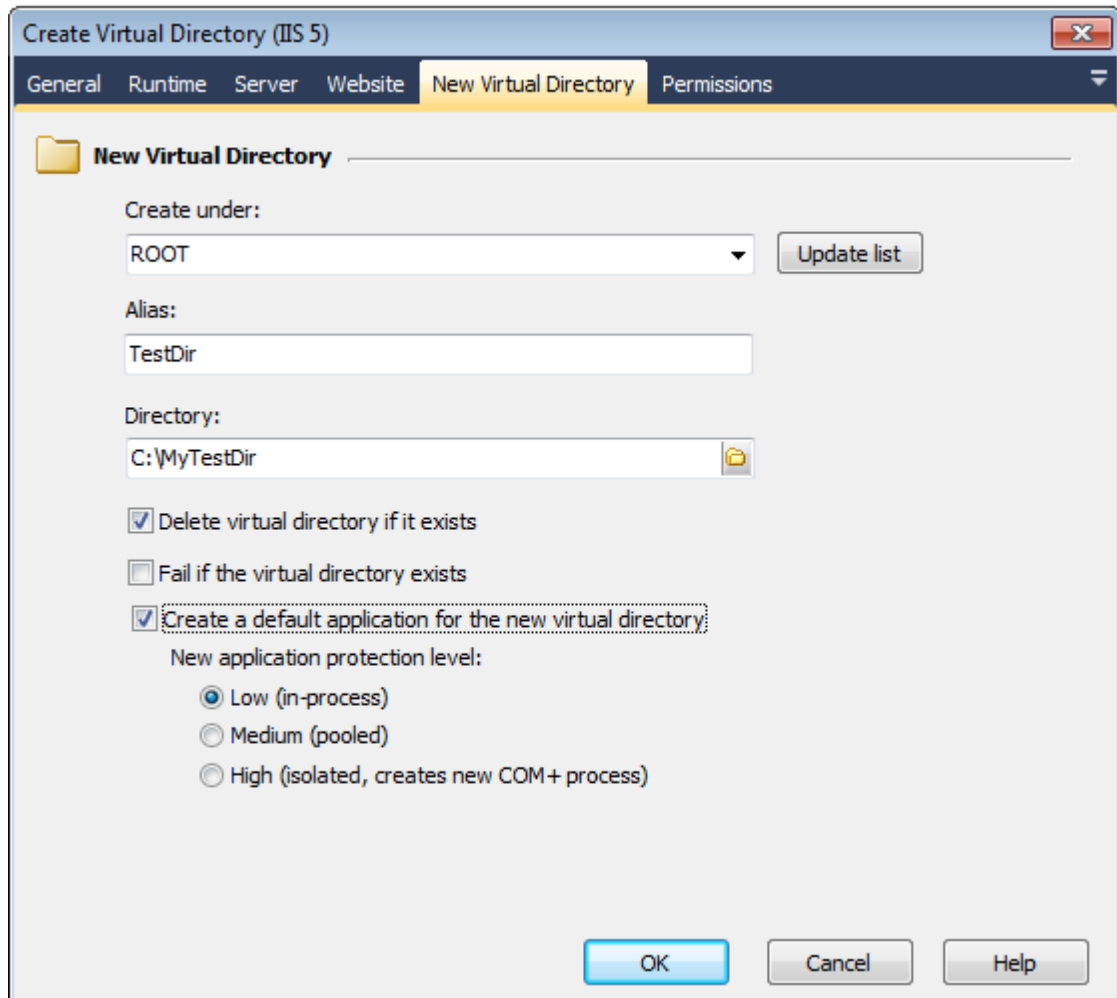
Specify the name of the server you wish to back up.

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 FinalBuilder IIS Backup 0, FinalBuilder IIS Backup 1, FinalBuilder IIS Backup 2...)

6.20.1.4 Create Virtual Directory Action

The Create Virtual Directory action allows you to add a new virtual directory to an existing IIS web site.



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.

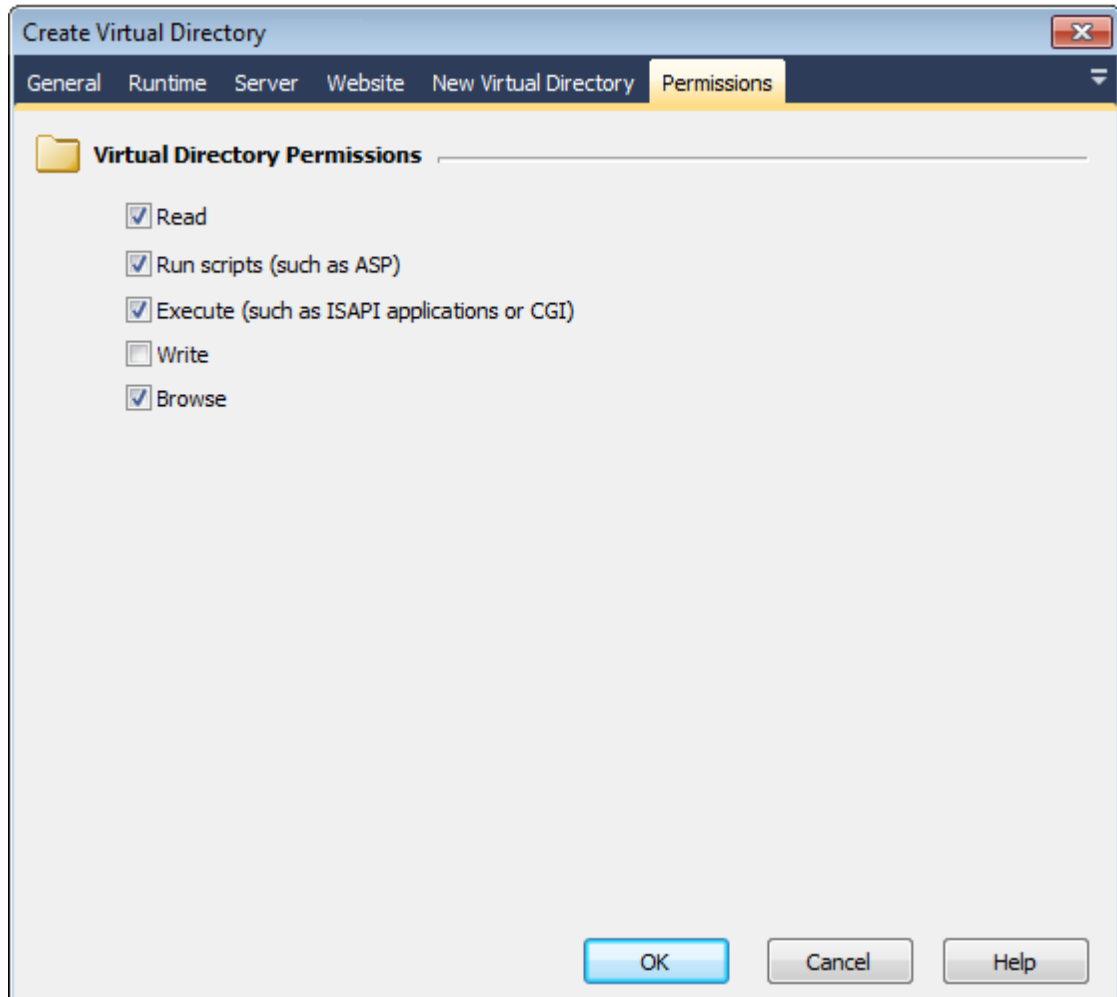
Directory

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.

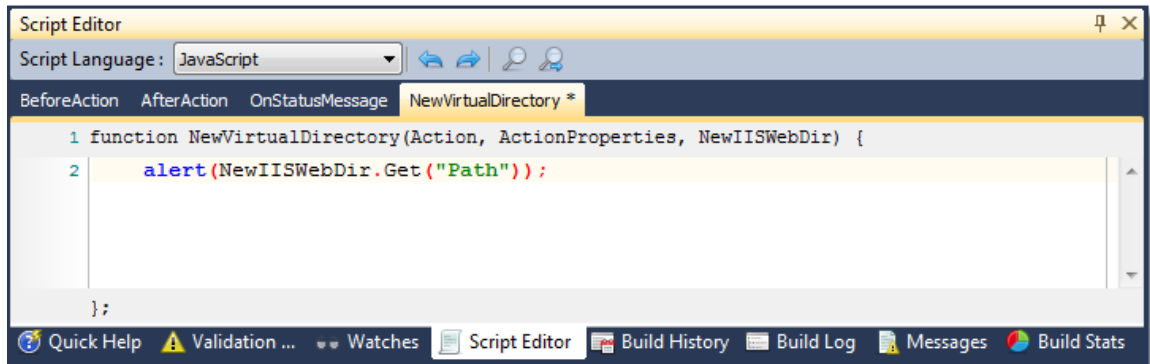
Permissions Tab



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.



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.

6.20.1.5 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.

The screenshot shows the 'Create Web Site (IIS 5)' dialog box with the 'Website Properties' tab selected. The dialog has four tabs: 'General', 'Runtime', 'Website Properties', and 'Permissions'. The 'Website Properties' tab contains the following sections:

- IIS Server**: A text box labeled 'Server:' containing the text 'localhost'.
- New Web Site Details**:
 - A text box labeled 'Description (ServerComment):'.
 - Two checkboxes: ☒ 'Fail if a web site with this description exists' and ☐ 'Delete and overwrite any web site with this description'.
 - A text box labeled 'Root Directory:' with a folder icon button to its right.
 - A checked checkbox labeled 'Start new website after creating it'.
- Server Bindings**:
 - Three text boxes labeled 'Bind to IP:', 'Port:', and 'Hostname:'.
 - The 'Port:' box contains the text '80'.
 - A note below the boxes: '(Leave fields blank to bind to any of a given parameter)'.

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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

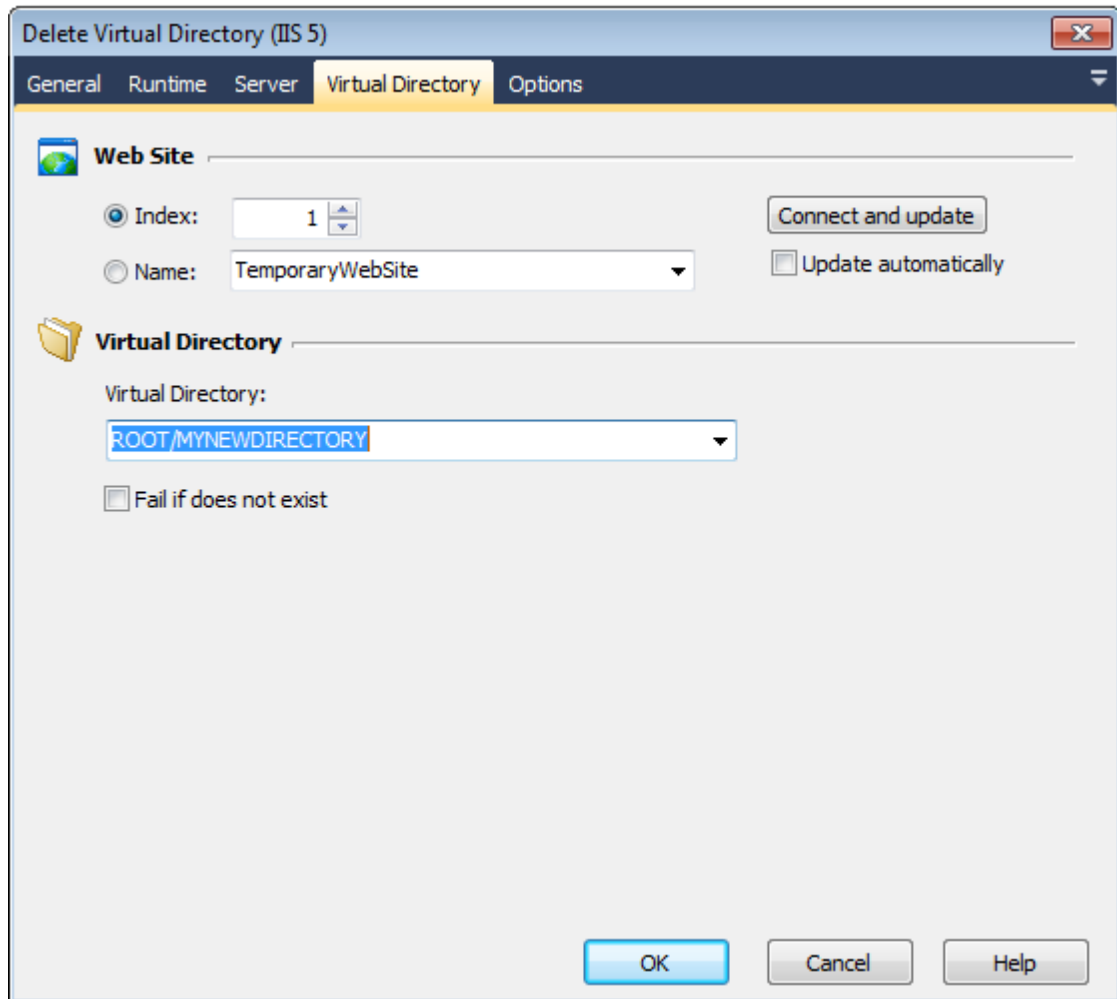
You can specify a specific IP, hostname, or Port combination to bind the new server to. Leave values blank to match all-inclusive wild cards.

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.

6.20.1.6 Delete Virtual Directory Action



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.

6.20.1.7 Delete Web Site Action

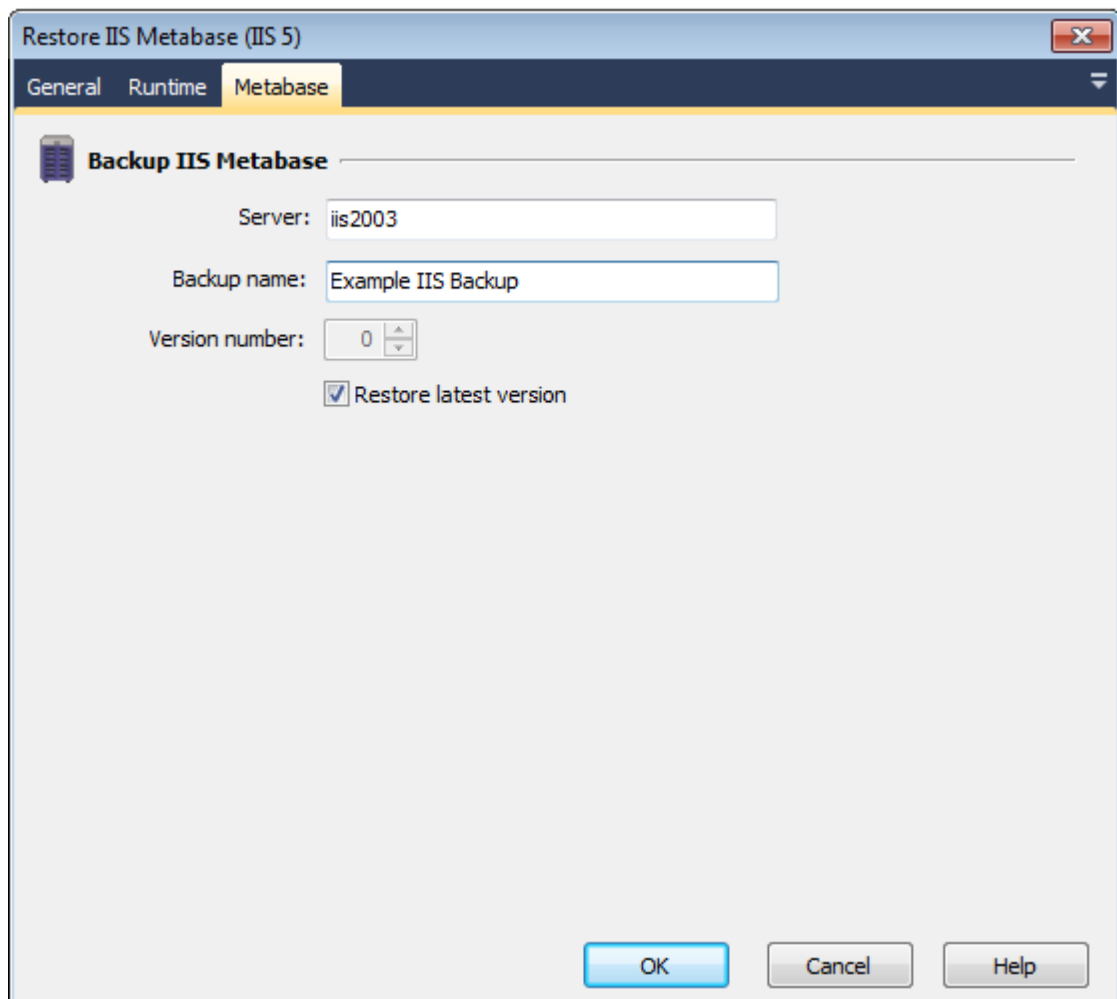
The Delete Web Site action allows you remove a web site from the IIS metabase.

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.

6.20.1.8 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

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.

6.20.1.9 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.)

6.20.1.10 Start / Stop / Restart WWW Service Actions

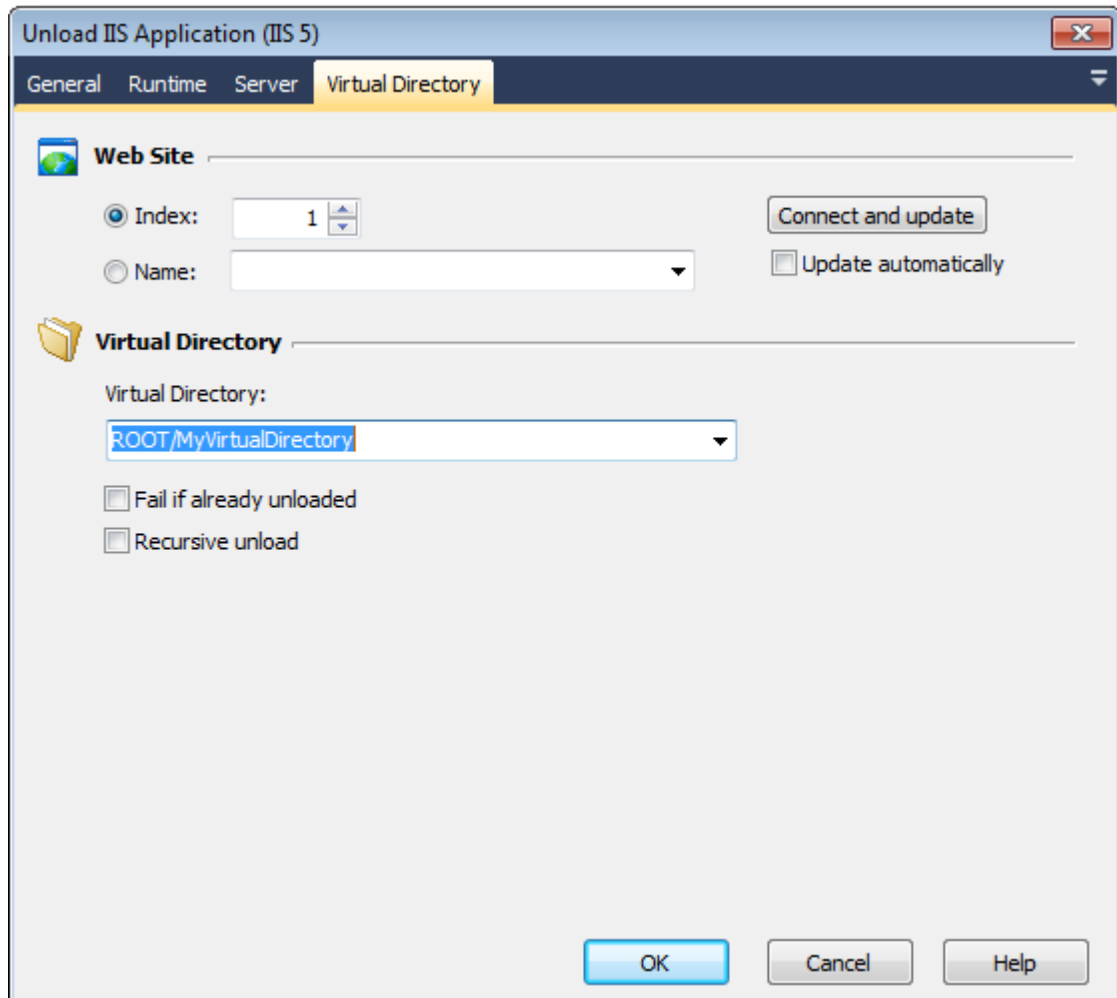
The Start, Stop & Restart WWW Service actions use WMI to start and stop the IIS WWW service (W3SVC) on a server running IIS.

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.)

6.20.1.11 Unload IIS Application Action

The Unload IIS Application actions allows you unload a given IIS web application from memory.



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.

6.20.2 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.

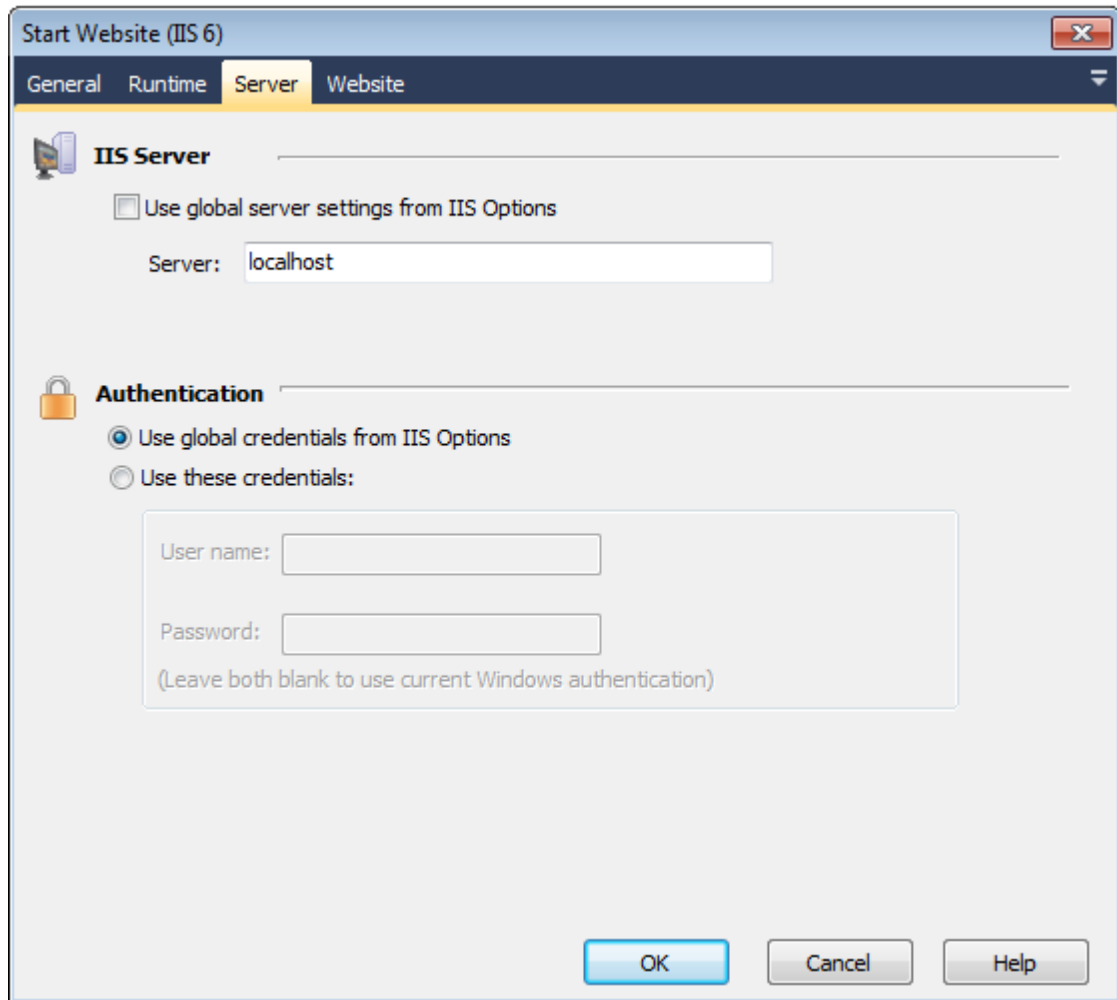
(The WMI interface is not available in IIS versions below 6.0.)

6.20.2.1 Server Selection

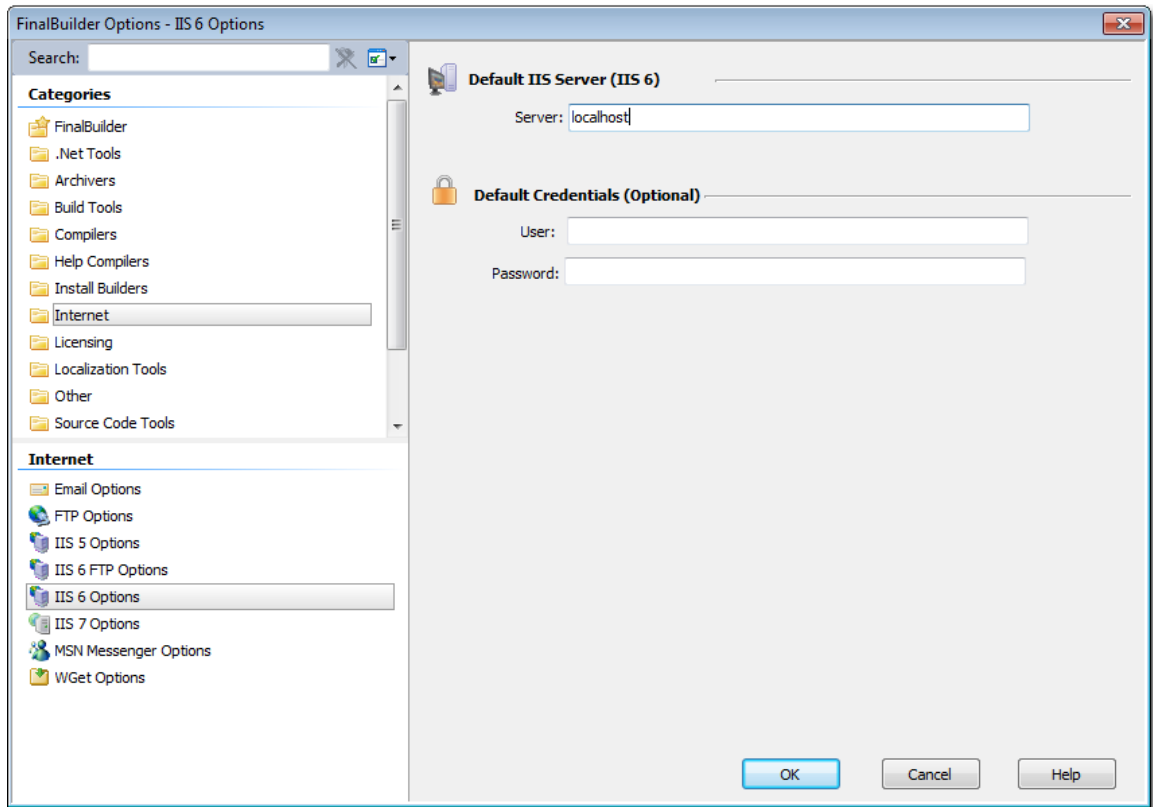
For each IIS action, you can specify the IIS Server as either a global server (specified in the options) or a particular server name. You can also specify credentials as either the credentials specified in the Options, or a particular set of credentials.

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.

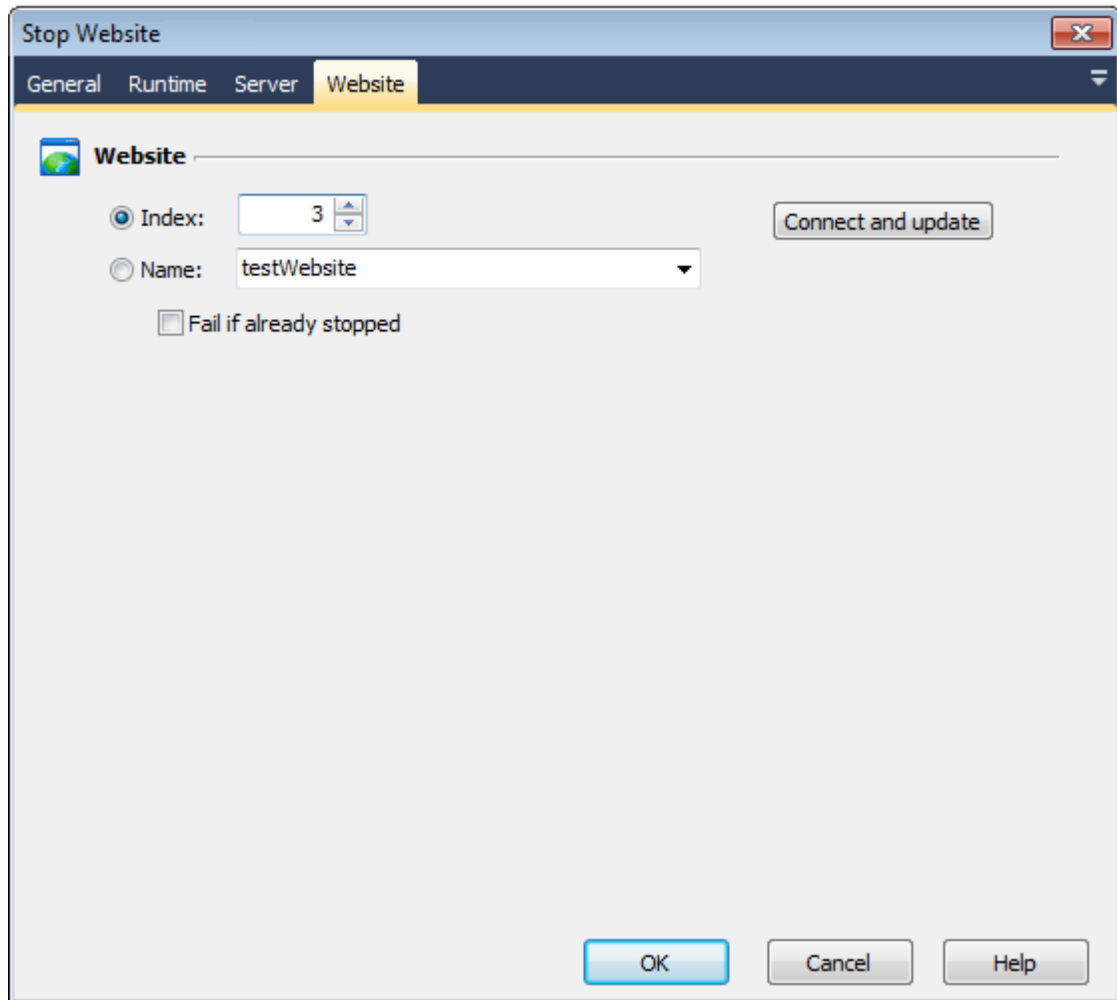


To select the global server name and credentials, choose Options from the Tools menu and then click on the IIS 6 Options tab under 'Internet':



6.20.2.2 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.)



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.

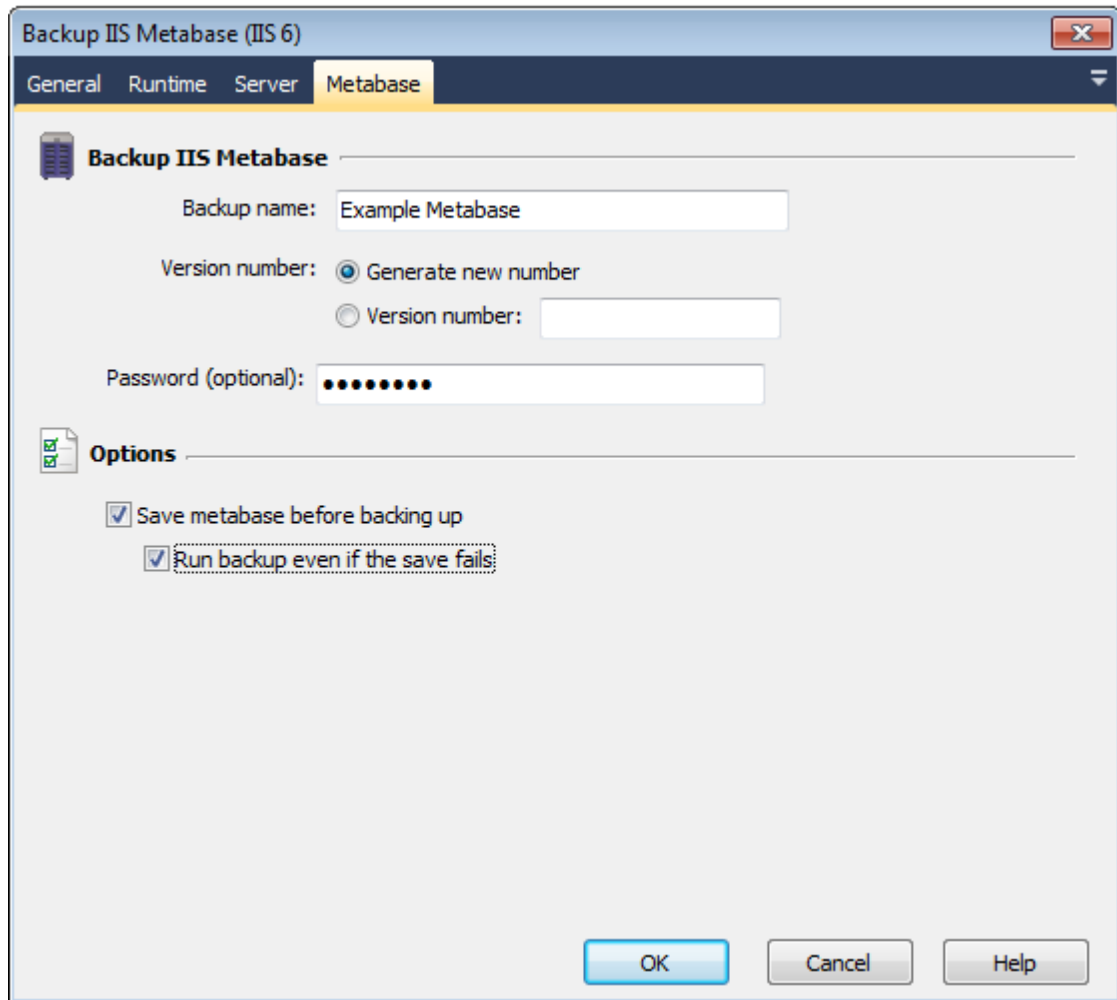
The "**Fail if already...**" checkbox allows you to specify whether some actions will abort if a given action (ie stopping or starting a web site) has already been taken.

6.20.2.3 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 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

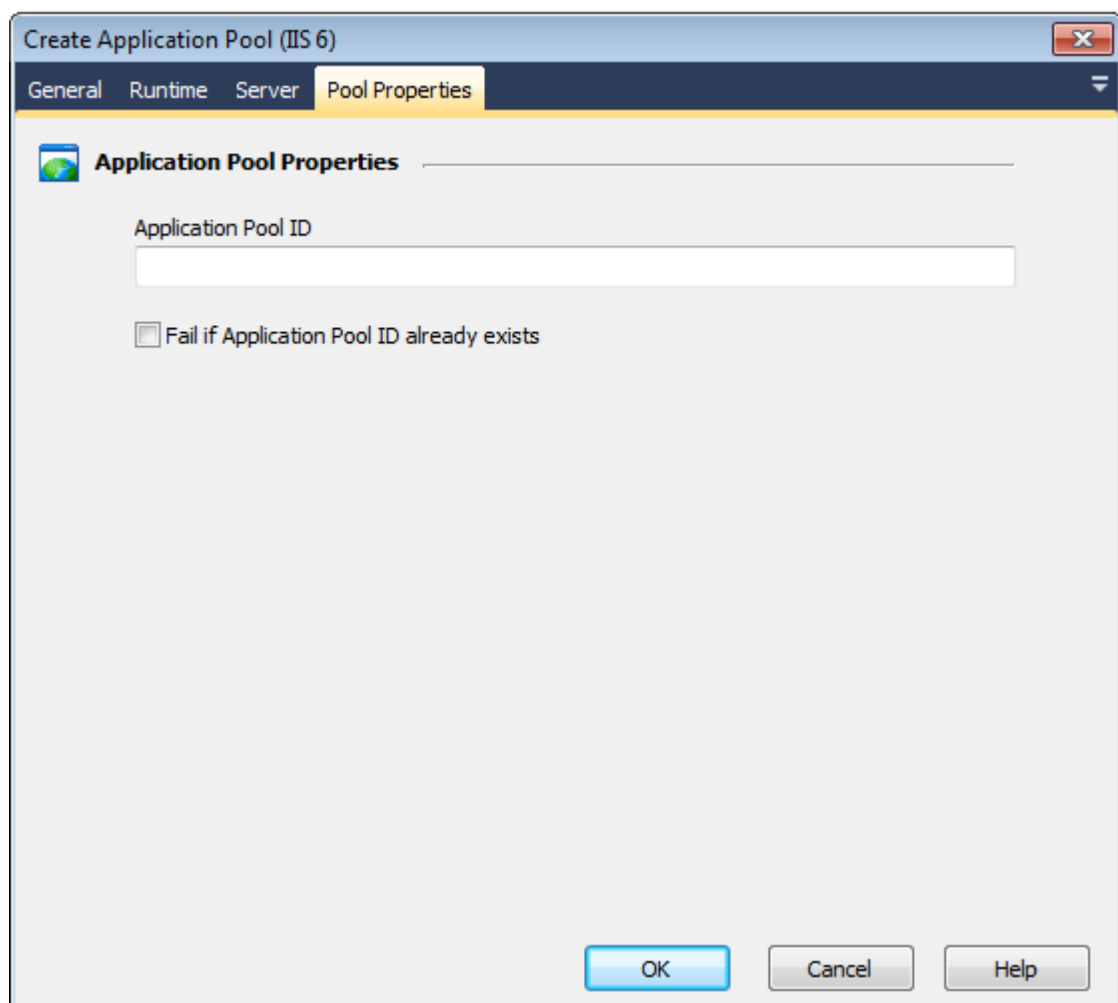
The contents of the metabase will be saved before the backup is undertaken.

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."

6.20.2.4 Create Application Pool

The IIS Create Application Pool action allows you to create new Application Pools on an IIS Server.

**Application Pool ID**

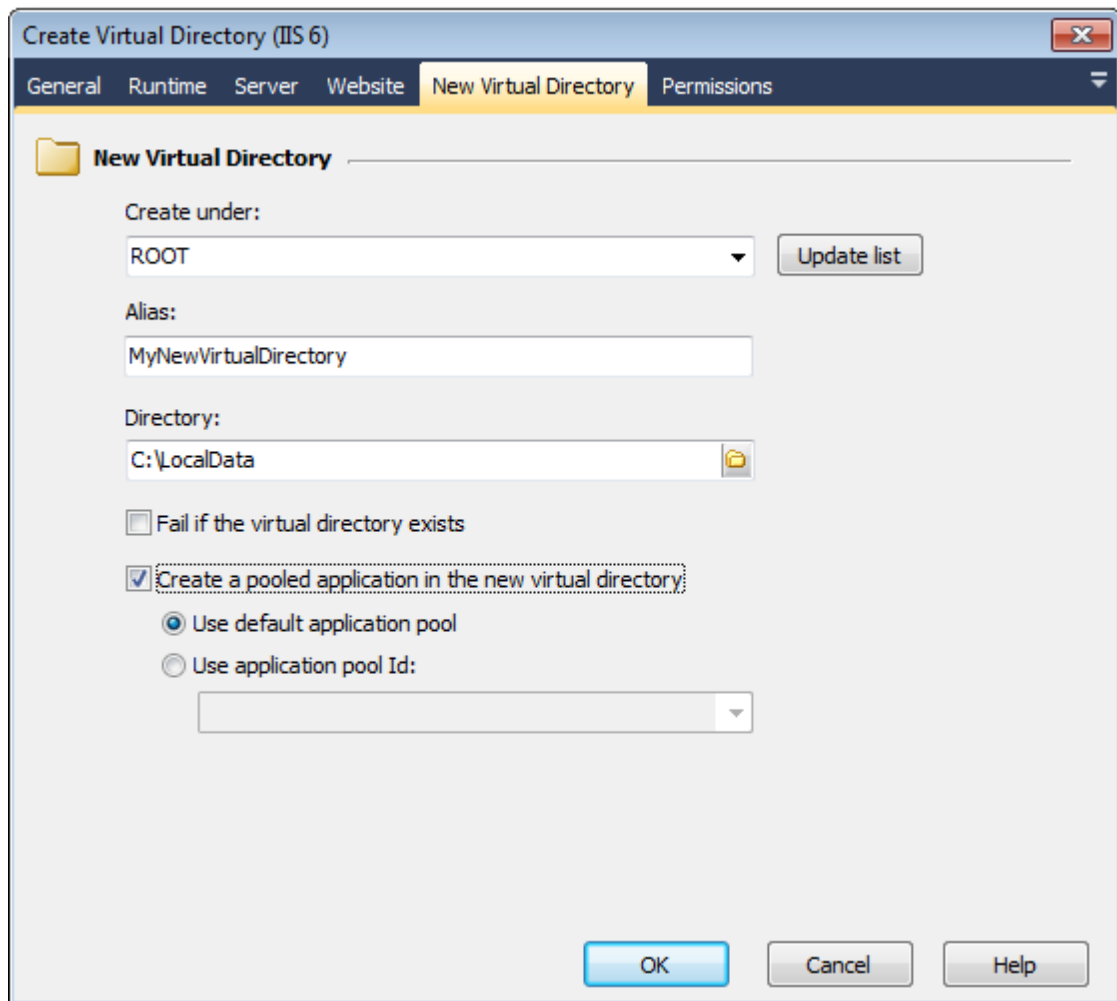
The identifier to associate with the new Application Pool.

Fail if Application Pool ID already exists

Performs a check on the server to see whether an existing application pool has the same identifier.

6.20.2.5 Create Virtual Directory Action

The Create Virtual Directory action allows you to add a new virtual directory to an existing IIS web site.

**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.

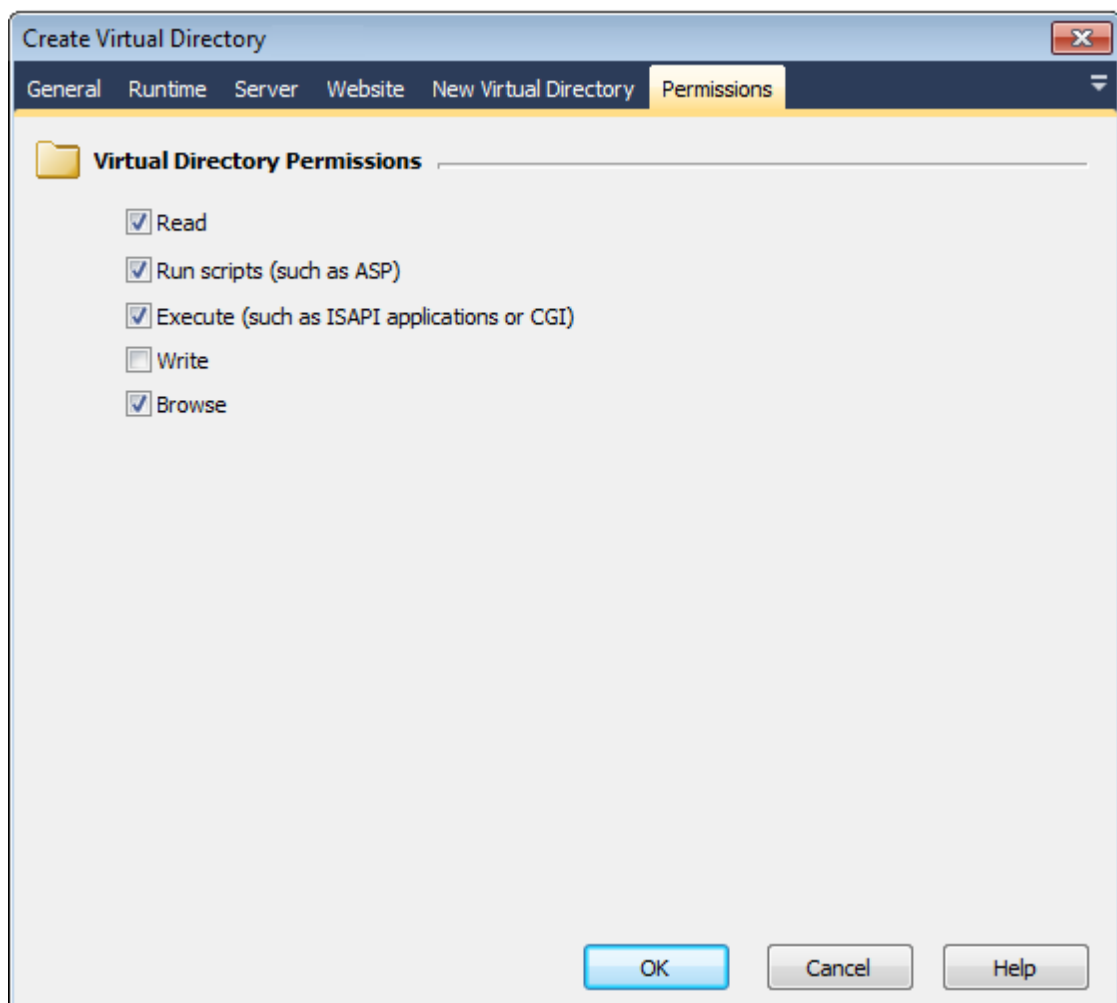
Directory

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

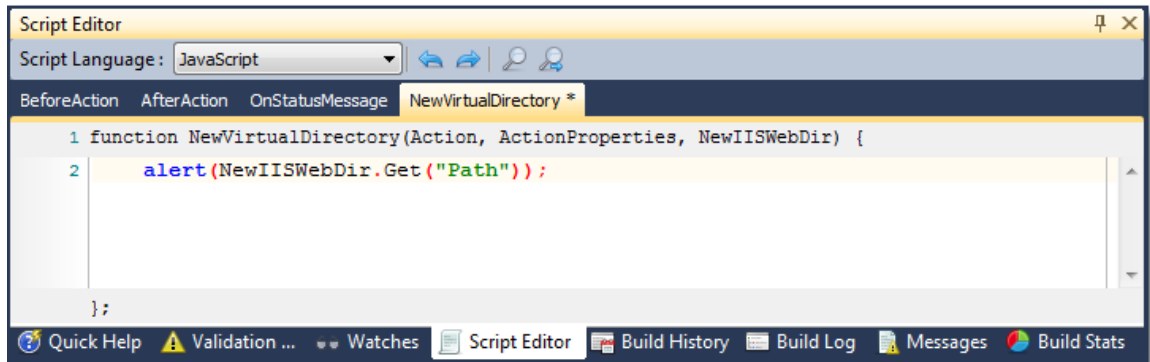


Set the permissions you want IIS users to have on the new Virtual Directory.

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.



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.)

6.20.2.6 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.

The screenshot shows the 'Create Web Site (IIS 6)' dialog box with the 'Website Properties' tab selected. The 'New Web Site Details' section contains a 'Description (ServerComment)' field with the text 'Example Website'. Below it are two checkboxes: 'Fail if a web site with this description exists' (unchecked) and 'Delete and overwrite any web site with this description' (checked). The 'Root Directory' field shows 'c:\MyWebsiteData' with a folder icon to its right. A checkbox 'Start new website after creating it' is checked. The 'Server Bindings' section has three fields: 'Bind to IP:' (empty), 'Port: 80', and 'Hostname:' (empty). A note below these fields says '(Leave fields blank to bind to any of a given parameter)'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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.

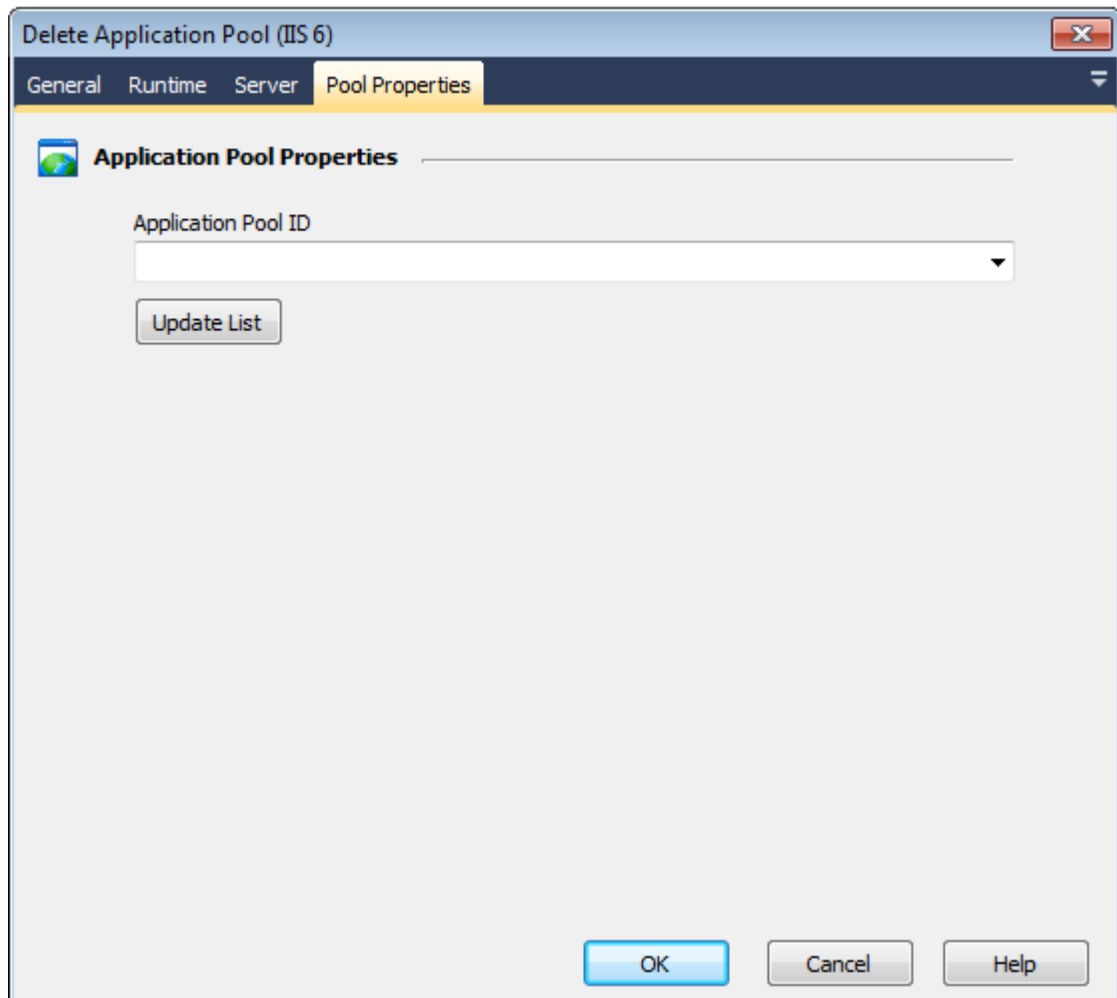
Note that if any of these values overlap with an existing web site, the two cannot be run concurrently.

Permissions Tab

Set the permissions for the new web site's root Virtual Directory.

6.20.2.7 Delete Application Pool

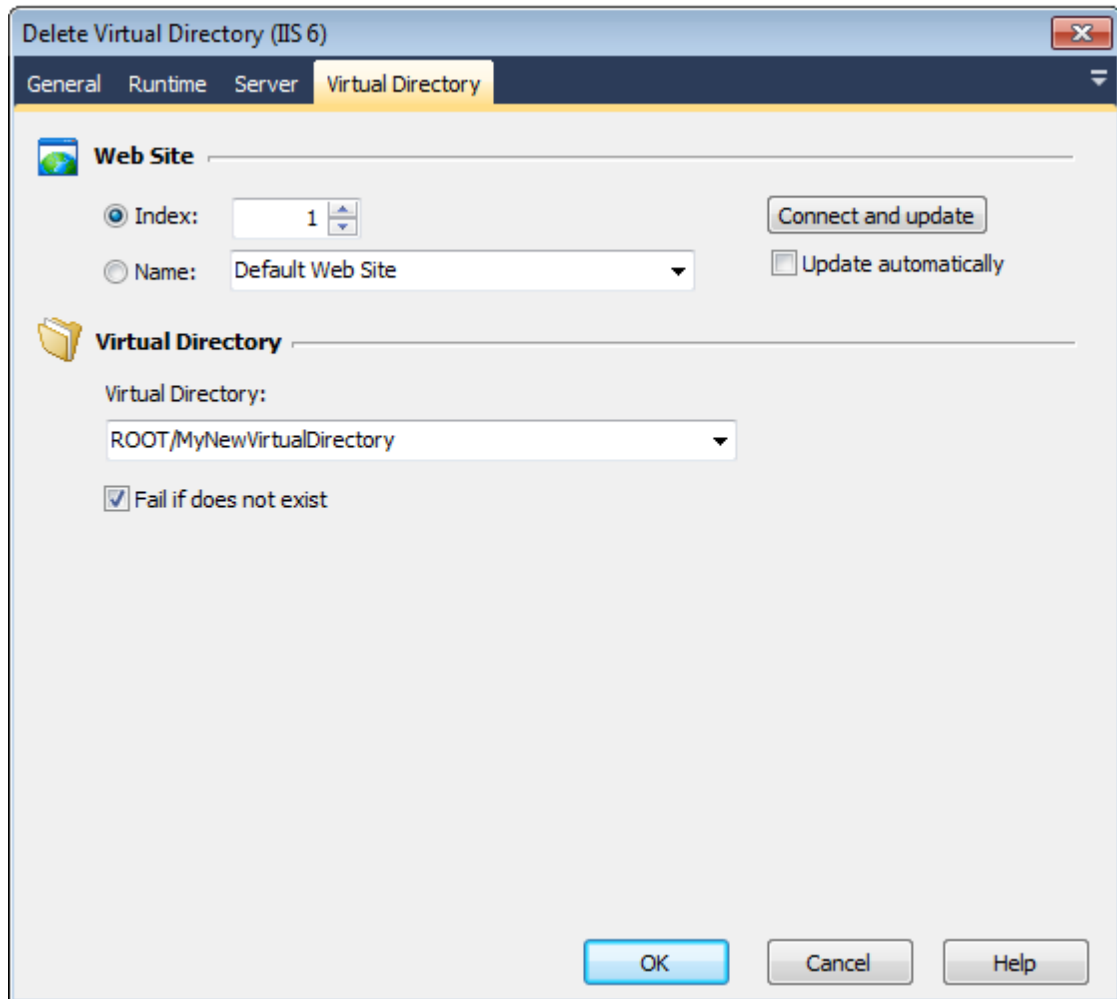
The IIS Delete Application Pool action enables you to remove existing application pools from an IIS server.



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.

6.20.2.8 Delete Virtual Directory Action



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.

6.20.2.9 Delete Web Site Action

The Delete Web Site action allows you remove a web site from the IIS metabase.

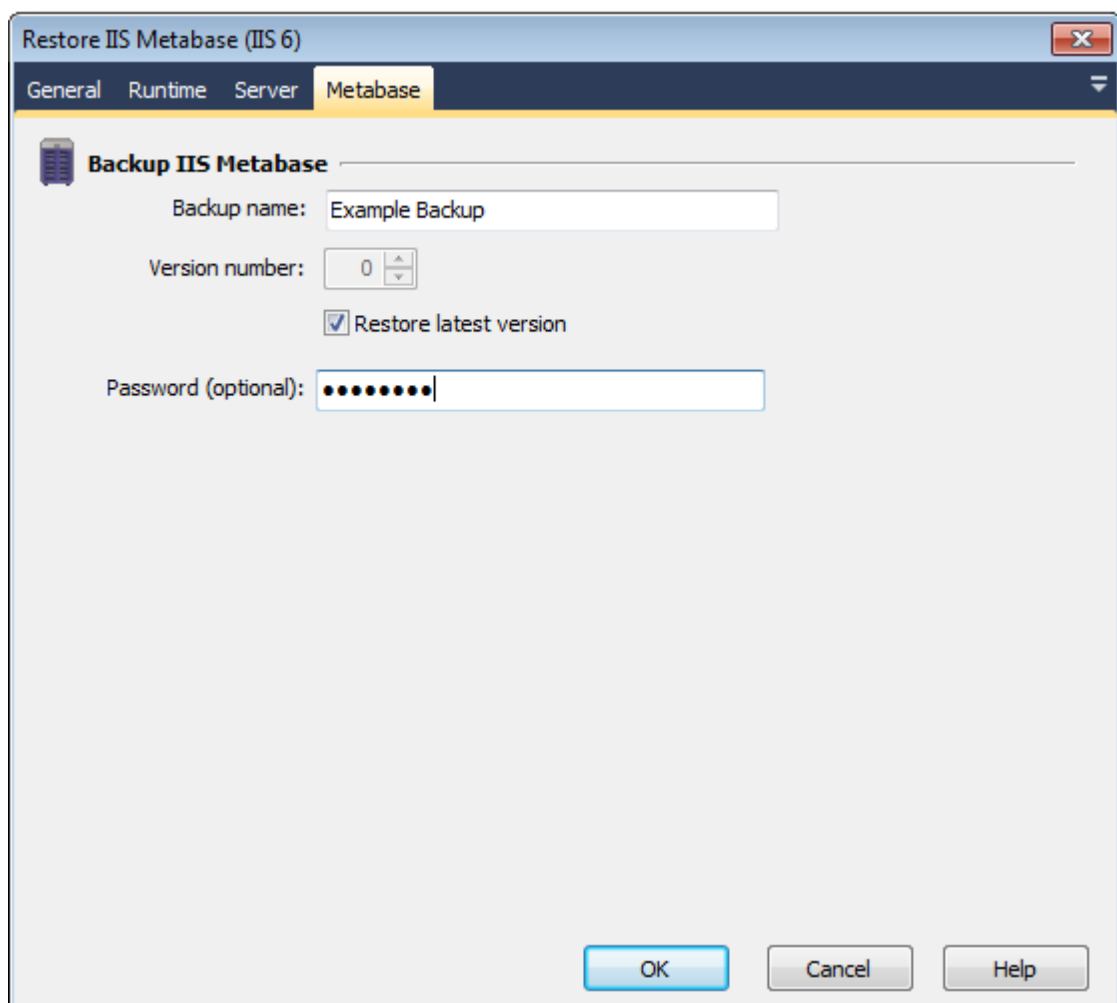
Use the Server Selection tab to choose the IIS server to delete the web site from & credentials for that server. Then use the Website Selection tab to choose the web site to delete.

6.20.2.10 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 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.

6.20.2.11 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.)

6.20.2.12 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.

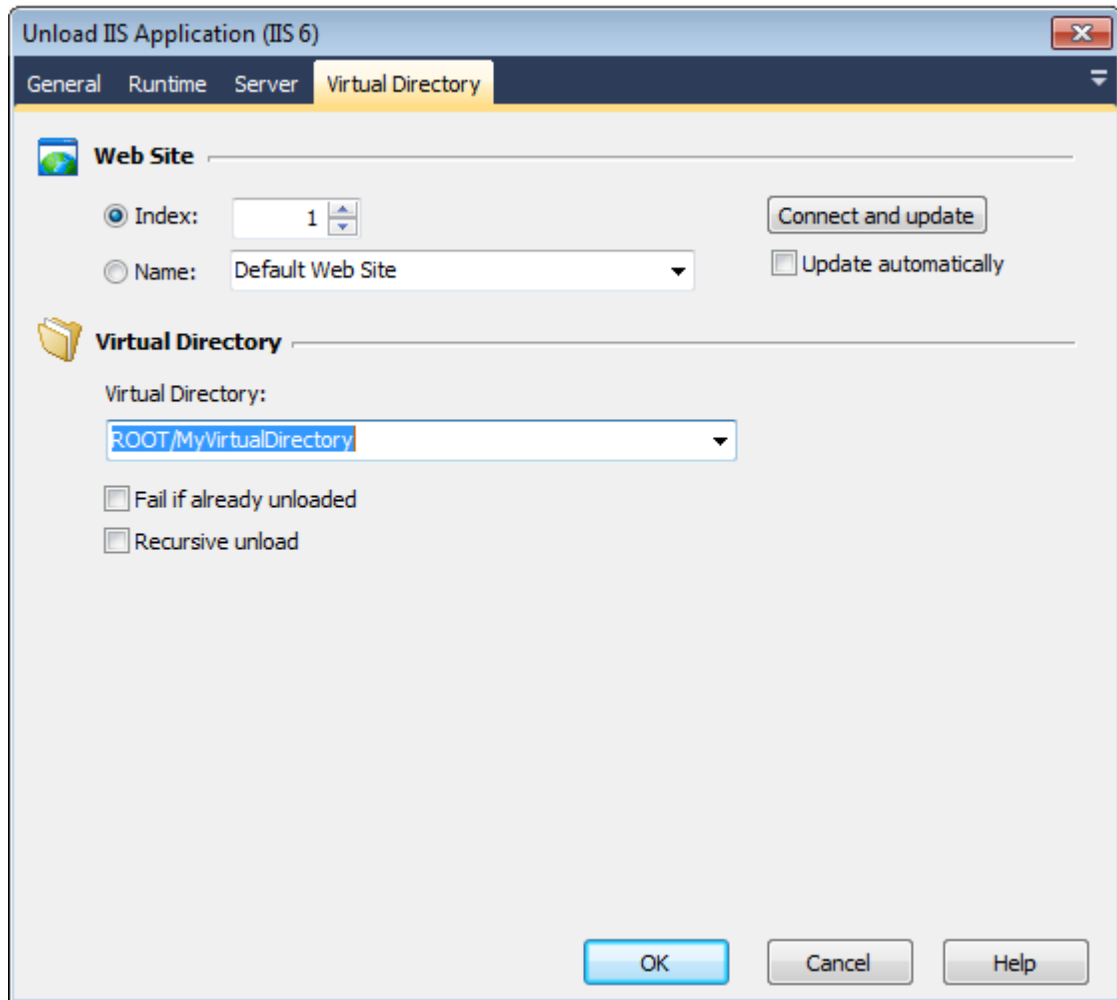
Use the Server Selection tab to choose the server you wish to administer, and to specify credentials for connecting to the server.

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.)

6.20.2.13 Unload IIS Application Action

The Unload IIS Application actions allows you unload a given IIS web application from memory.

**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.

6.20.3 IIS 6 FTP

The IIS 6 FTP actions use the WMI interface to administer the FTP server features of IIS 6.

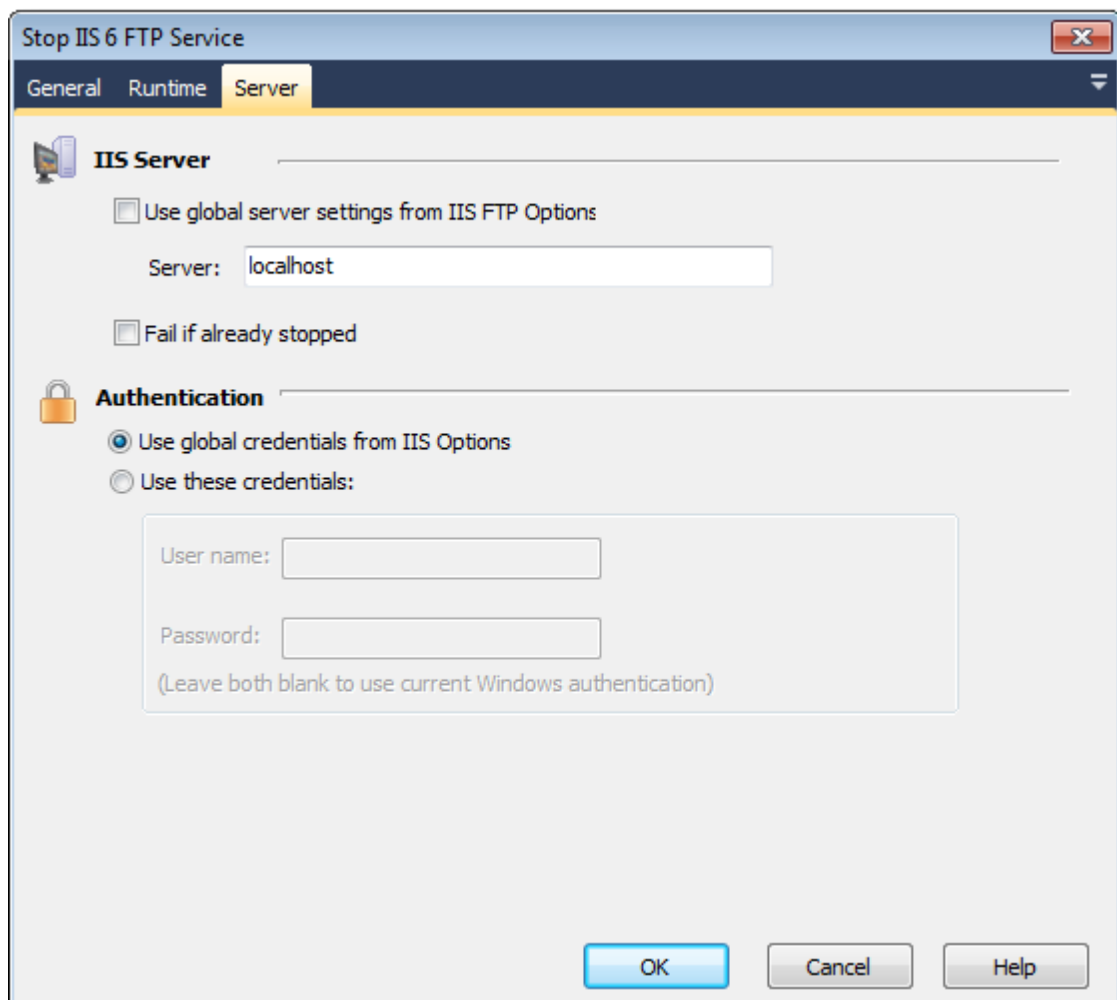
(The WMI interface is not available in IIS versions below 6.0.)

6.20.3.1 Server Selection

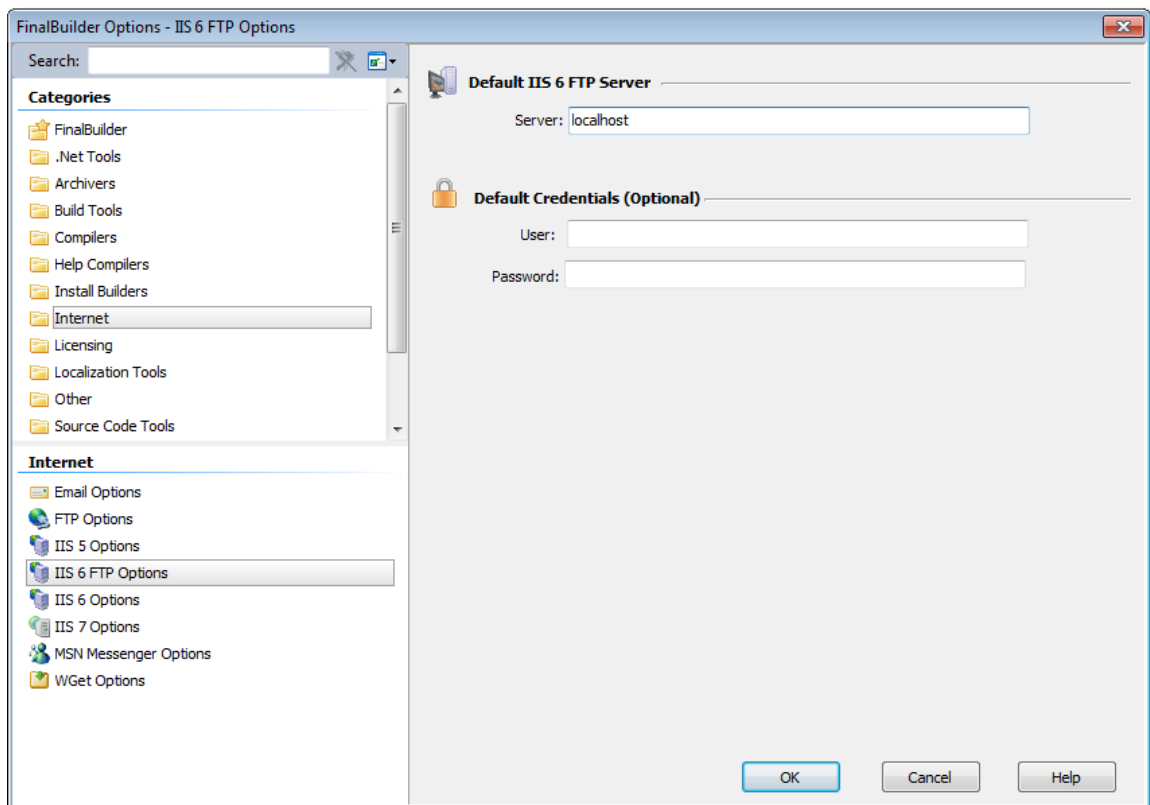
[FinalBuilder Professional Edition]

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.



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':

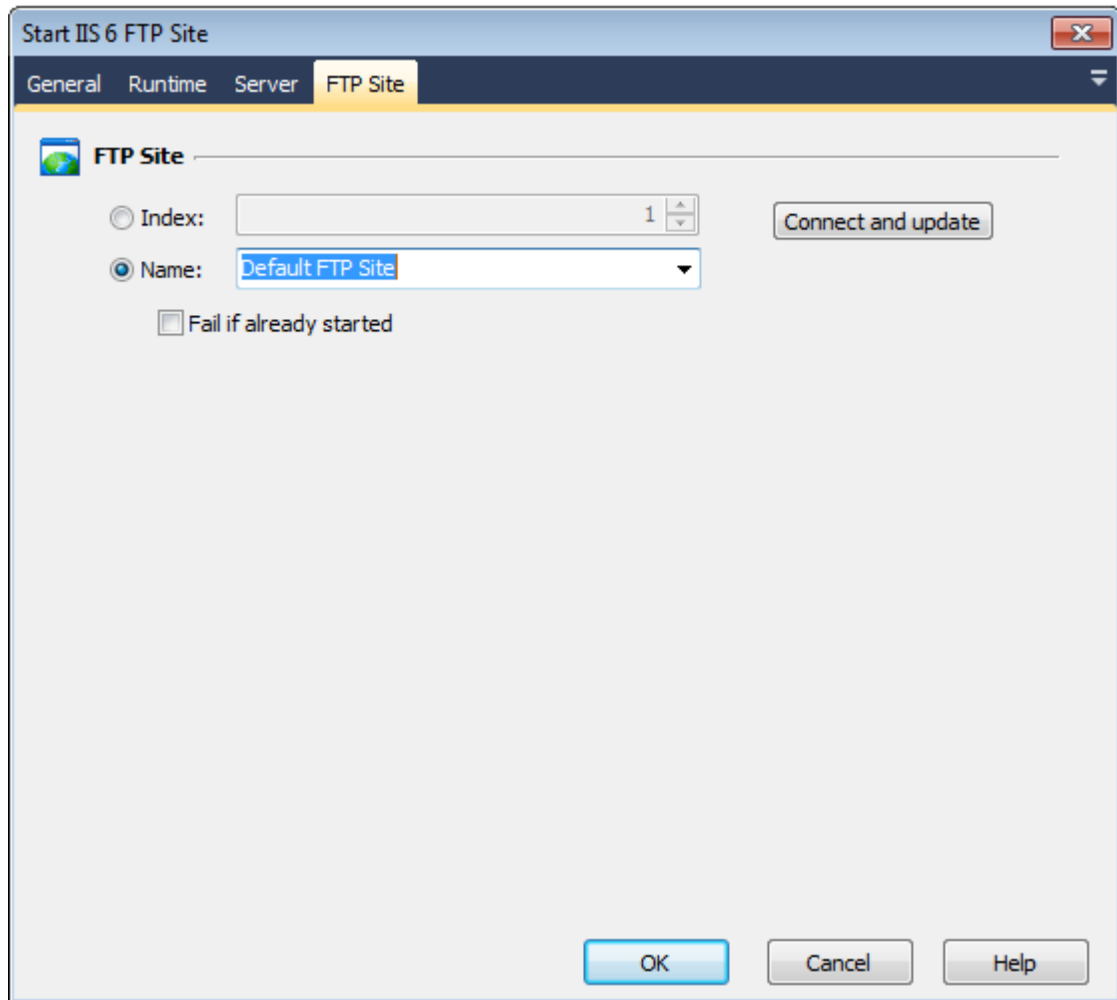


6.20.3.2 FTP Site Selection

[FinalBuilder 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.)

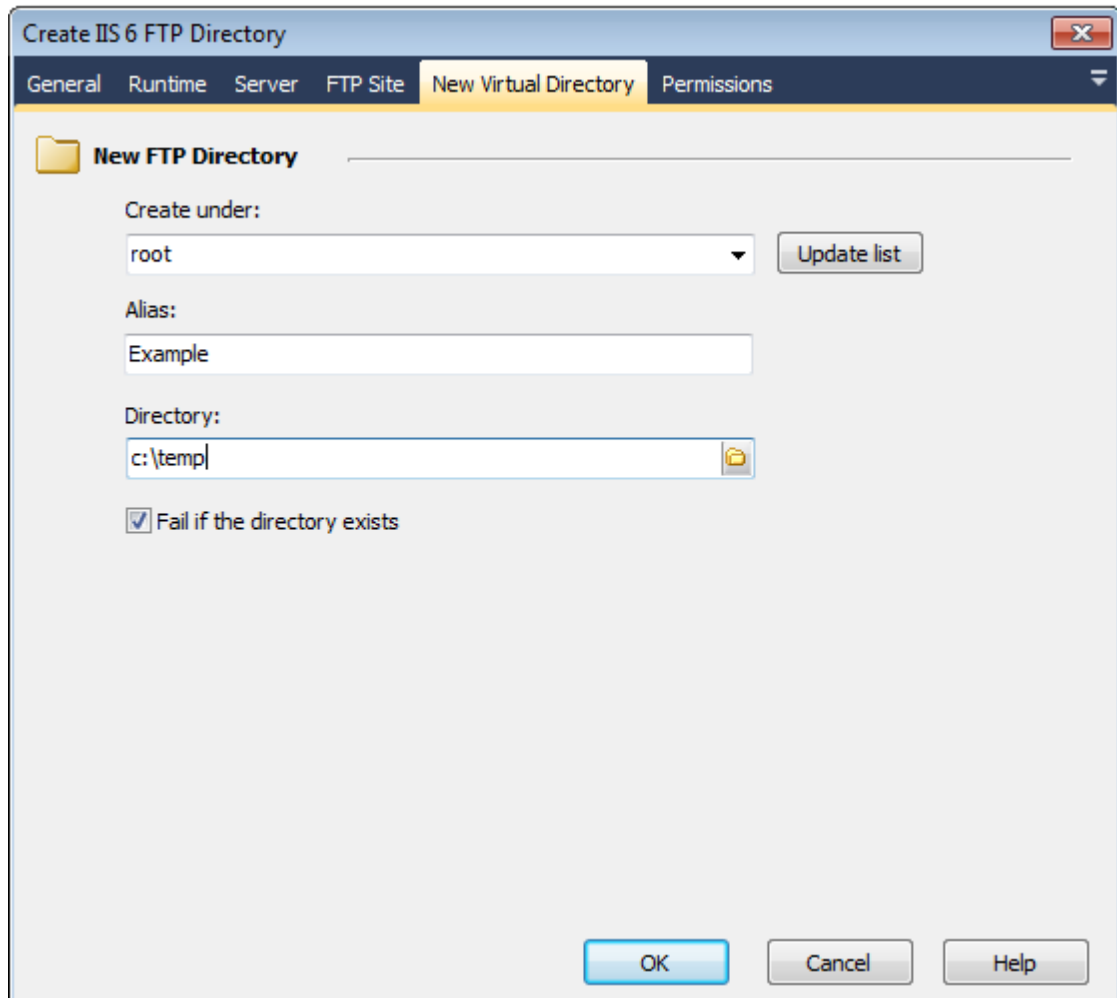


6.20.3.3 Create FTP Directory Action

[FinalBuilder Professional Edition]

Server & Website

Specify the Server and FTP site to use via the Server Selection & FTP Site Selection tabs, respectively.

**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.

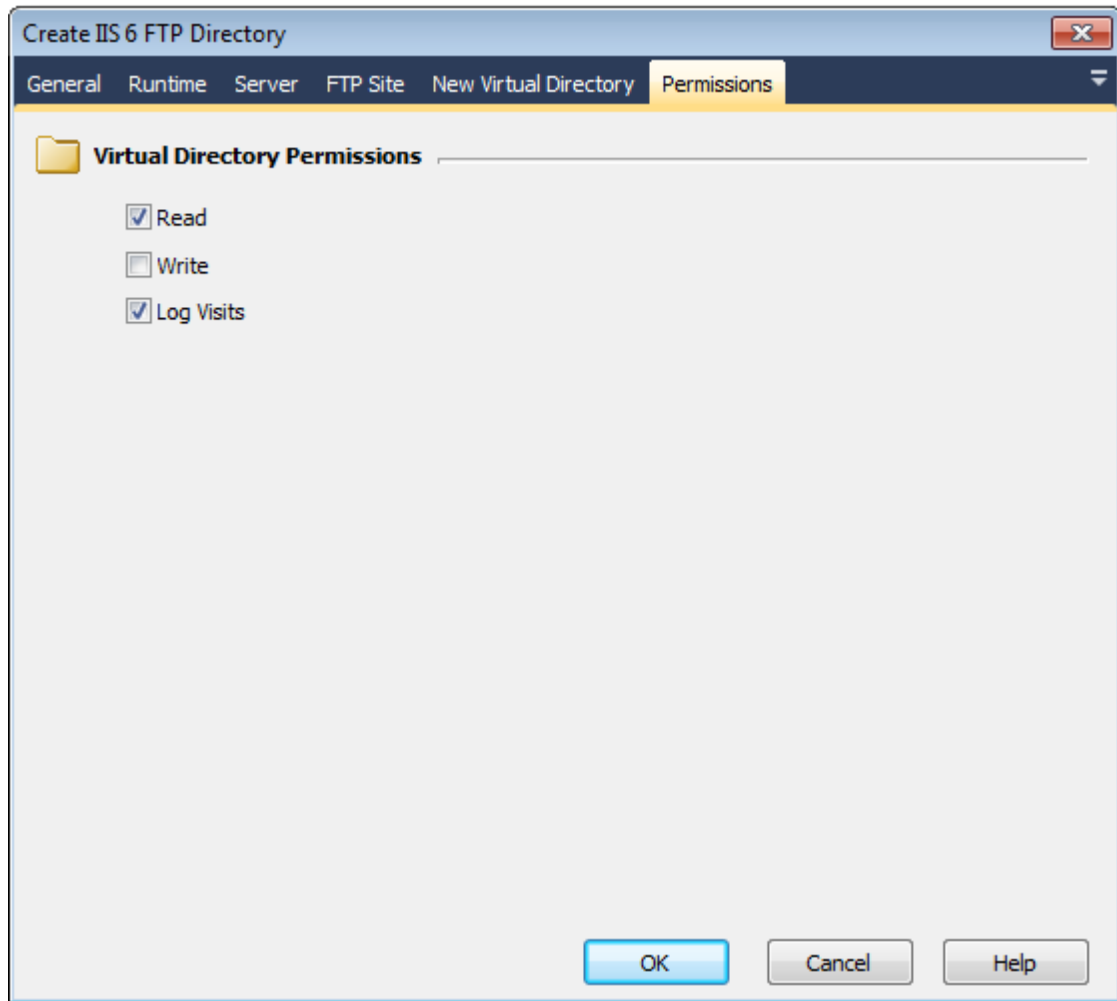
Directory

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

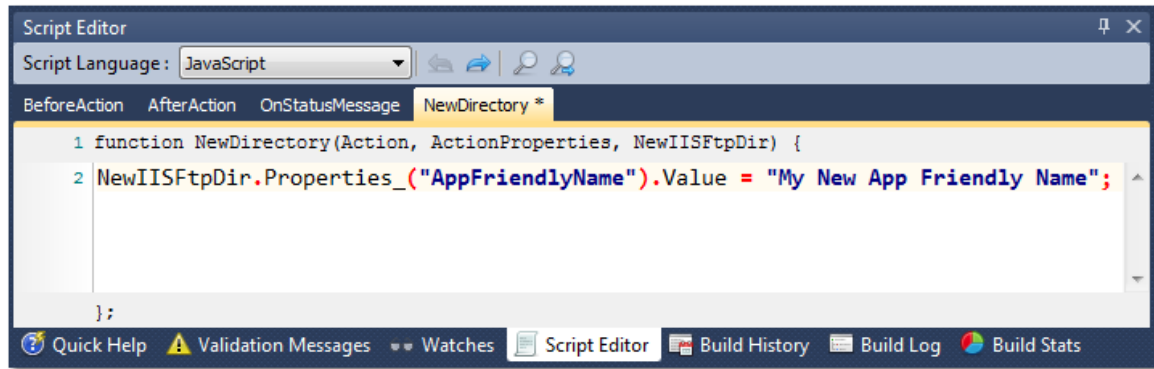


Set the permissions you want FTP users to have on the new Directory.

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.



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.)

6.20.3.4 Create FTP Site Action

[FinalBuilder 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.

The screenshot shows the 'Create IIS 6 FTP Site' dialog box with the 'Site Properties' tab selected. The 'New FTP Site Details' section contains a 'Description (ServerComment)' field with the text 'Example FTP'. Below it are two checkboxes: 'Fail if an FTP site with this description exists' (unchecked) and 'Delete and overwrite any FTP site with this description' (checked). The 'Root Directory' field contains 'c:\temp' with a folder icon to its right. Below that is a checked checkbox 'Start new FTP site after creating it'. The 'Server Bindings' section has three fields: 'Bind to IP:' (empty), 'Port:' (21), and 'Hostname:' (empty). At the bottom, there is a note: '(Leave fields blank to bind to any of a given parameter)'. The dialog has 'OK', 'Cancel', and 'Help' buttons at the bottom right.

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

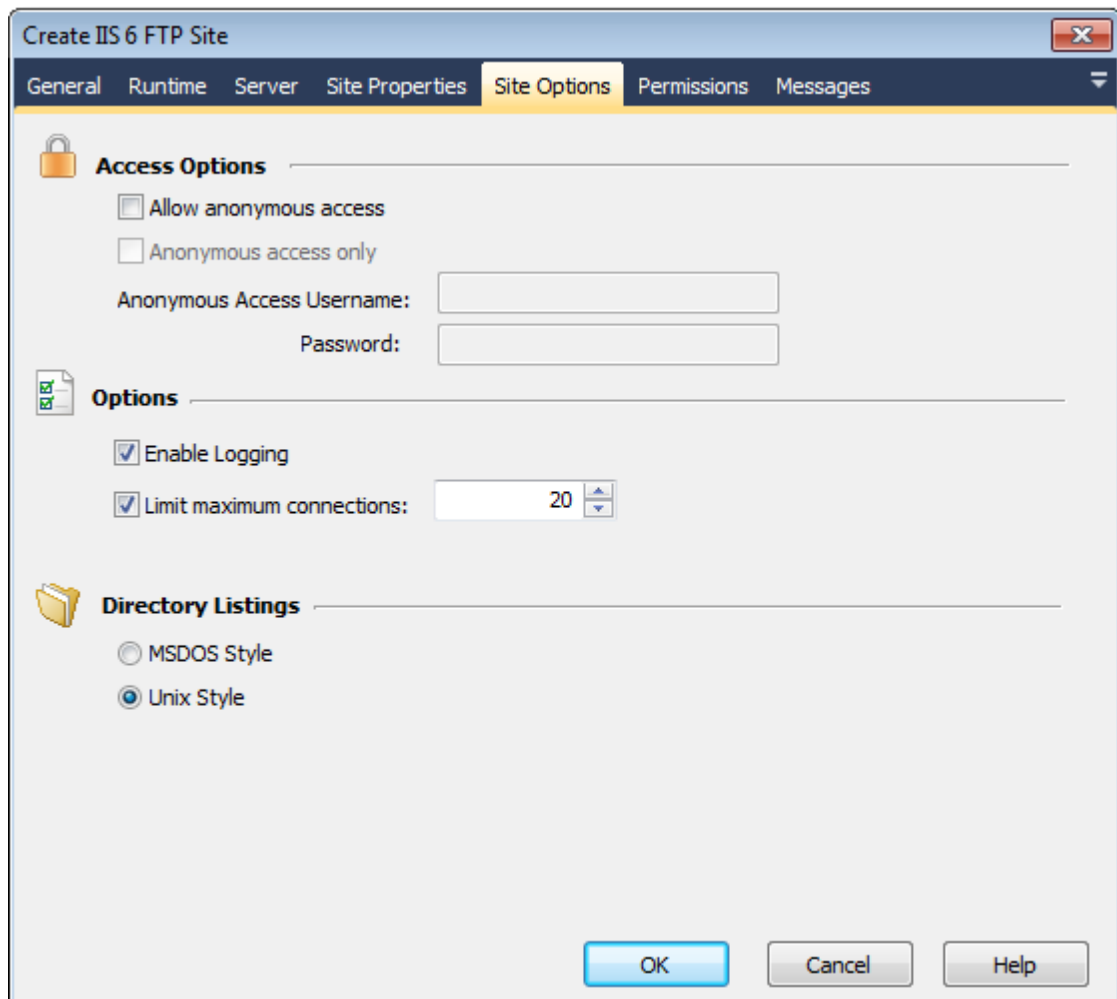
Specify the directory (on the IIS server) where you want files to be hosted for the root FTP directory.

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

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.

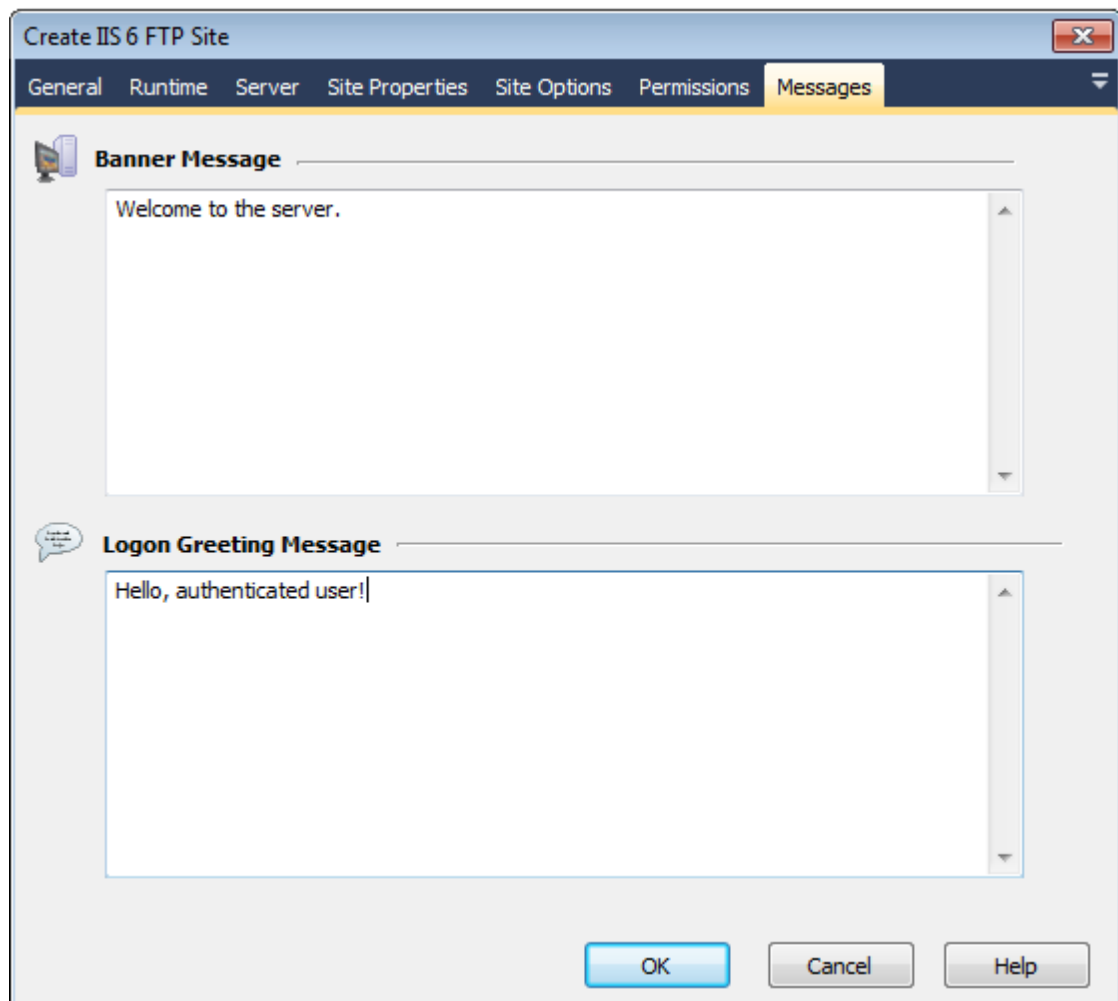
Directory Listings

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.

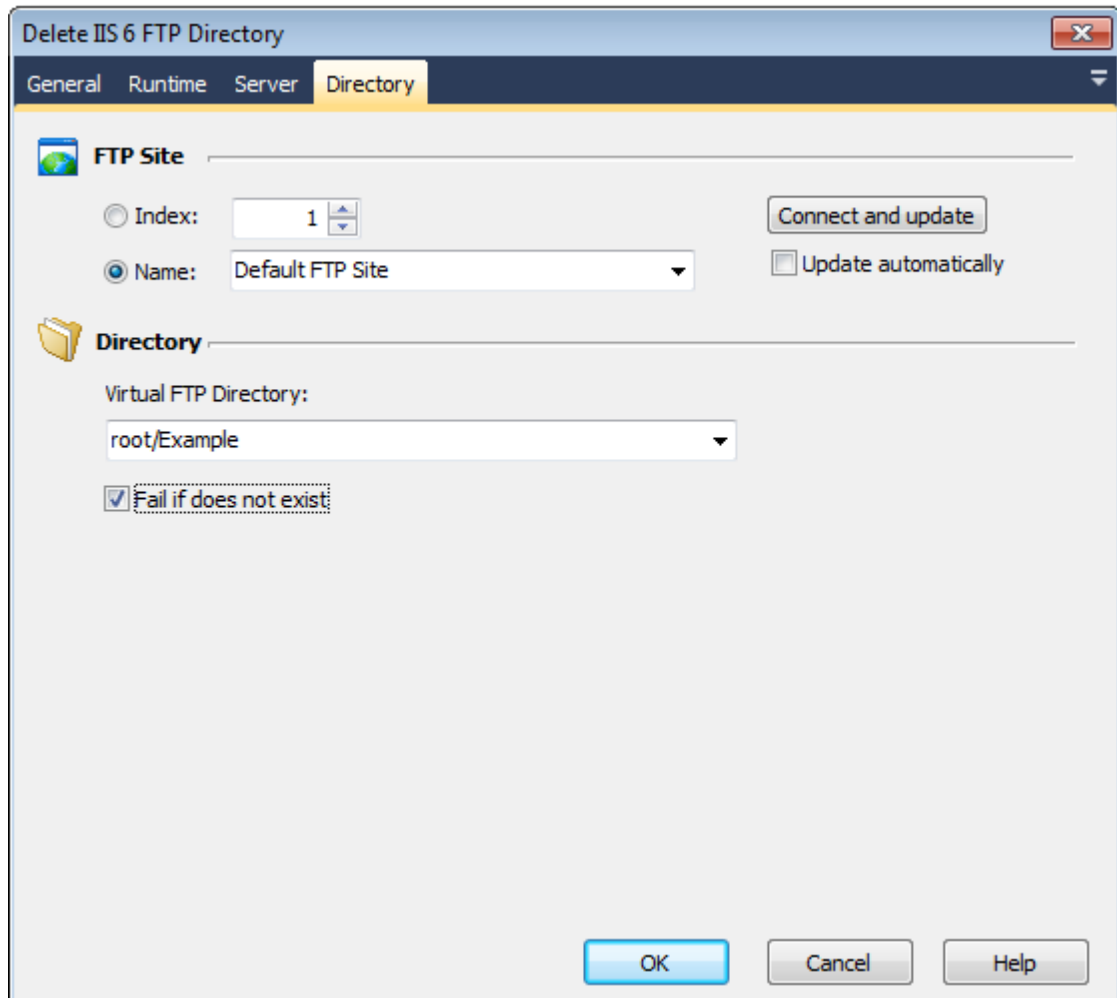
NB: There is an observed bug in IIS 6.0 whereby even though "Read" access is set on the root directory of a new FTP site, the Read checkbox appears unchecked when viewed in the IIS Management console. Despite the checkbox appearing unchecked, the root directory does support Read access.



The Banner Message and Logon Greeting Message are sent to the user when they first connect to site, and when they log in (respectively.)

6.20.3.5 Delete FTP Directory Action

[FinalBuilder Professional Edition]



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.

6.20.3.6 Delete FTP Site Action

[FinalBuilder 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.

6.20.3.7 Start / Stop / Pause FTP Site Actions

[FinalBuilder 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.)

6.20.3.8 Start / Stop / Restart FTP Service Actions

[FinalBuilder 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.)

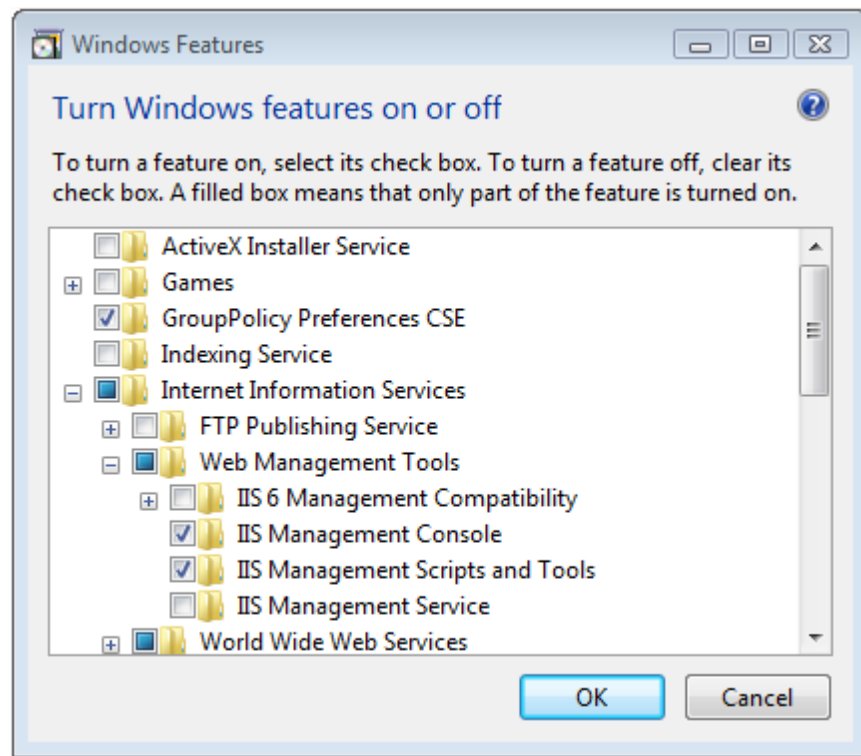
It is recommended that you use the Restart action instead of a Stop followed immediately by a Start action (the Restart action waits to ensure the service is fully stopped before restarting it.)

6.20.4 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.



6.20.4.1 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.

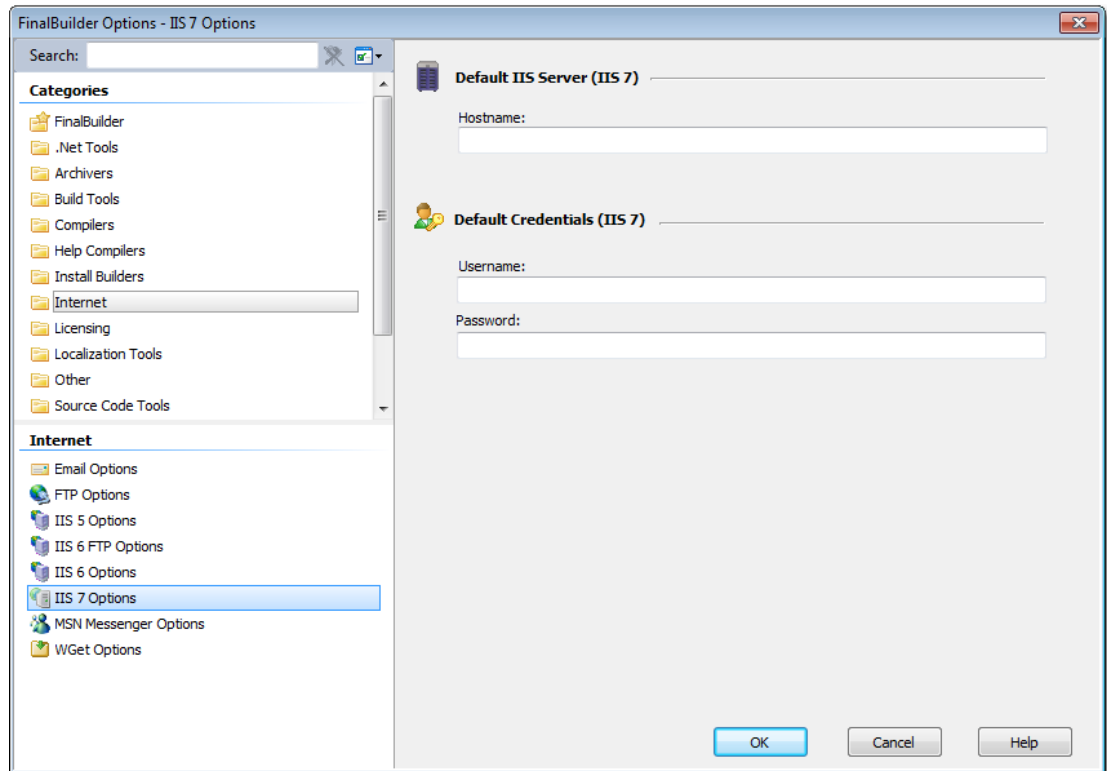
The screenshot shows a Windows-style dialog box titled "Create Website (IIS 7)". It has four tabs: "General", "Runtime", "Server" (which is selected and highlighted in yellow), and "Website Options".

Under the "Server" tab, there are two main sections:

- IIS Server (IIS 7)**: This section contains two radio buttons. The first, "Use the hostname specified in the global IIS options.", is selected. The second, "Provide the hostname of the IIS server.", is unselected. Below these is a text field labeled "Hostname:". A note below the field says "Leave blank to connect to the local machine."
- Credentials (IIS 7)**: This section also contains two radio buttons. The first, "Use the credentials specified in the global IIS options.", is selected. The second, "Provide the credentials to the IIS server.", is unselected. Below these are two text fields: "Username:" and "Password:". A note below the "Username:" field says "Leave blank to use the credentials of the logged in user."

At the bottom right of the dialog box are three buttons: "OK", "Cancel", and "Help".

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 FinalBuilder options and navigate to the IIS 7 options page.

**Hostname**

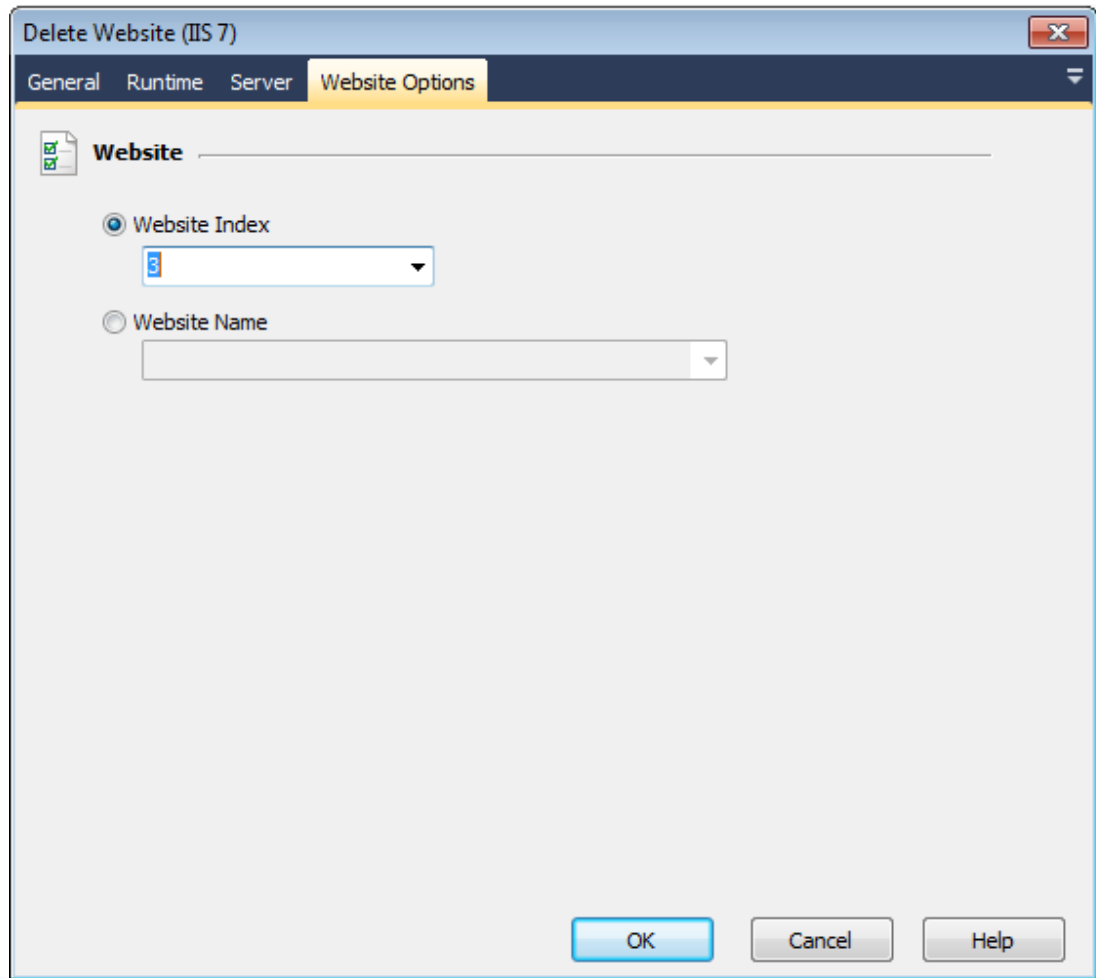
The name of the machine to connect to, leave blank if you wish to connect to the local machine.

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.

6.20.4.2 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.



6.20.4.3 Create Website Action

The Create Website (IIS 7) actions allows you to create a new website on an IIS 7 server.

The screenshot shows a Windows-style dialog box titled "Create Website (IIS 7)". It has four tabs: "General", "Runtime", "Server" (which is selected and highlighted in yellow), and "Website Options".

Under the "Server" tab, there are two main sections:

- IIS Server (IIS 7)**: This section contains two radio buttons. The first, "Use the hostname specified in the global IIS options.", is selected. The second, "Provide the hostname of the IIS server.", is unselected. Below the second option is a text field labeled "Hostname:". A note below the text field says "Leave blank to connect to the local machine."
- Credentials (IIS 7)**: This section also contains two radio buttons. The first, "Use the credentials specified in the global IIS options.", is selected. The second, "Provide the credentials to the IIS server.", is unselected. Below the second option are two text fields: "Username:" and "Password:". A note below the "Username:" field says "Leave blank to use the credentials of the logged in user."

At the bottom right of the dialog box are three buttons: "OK", "Cancel", and "Help".

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.

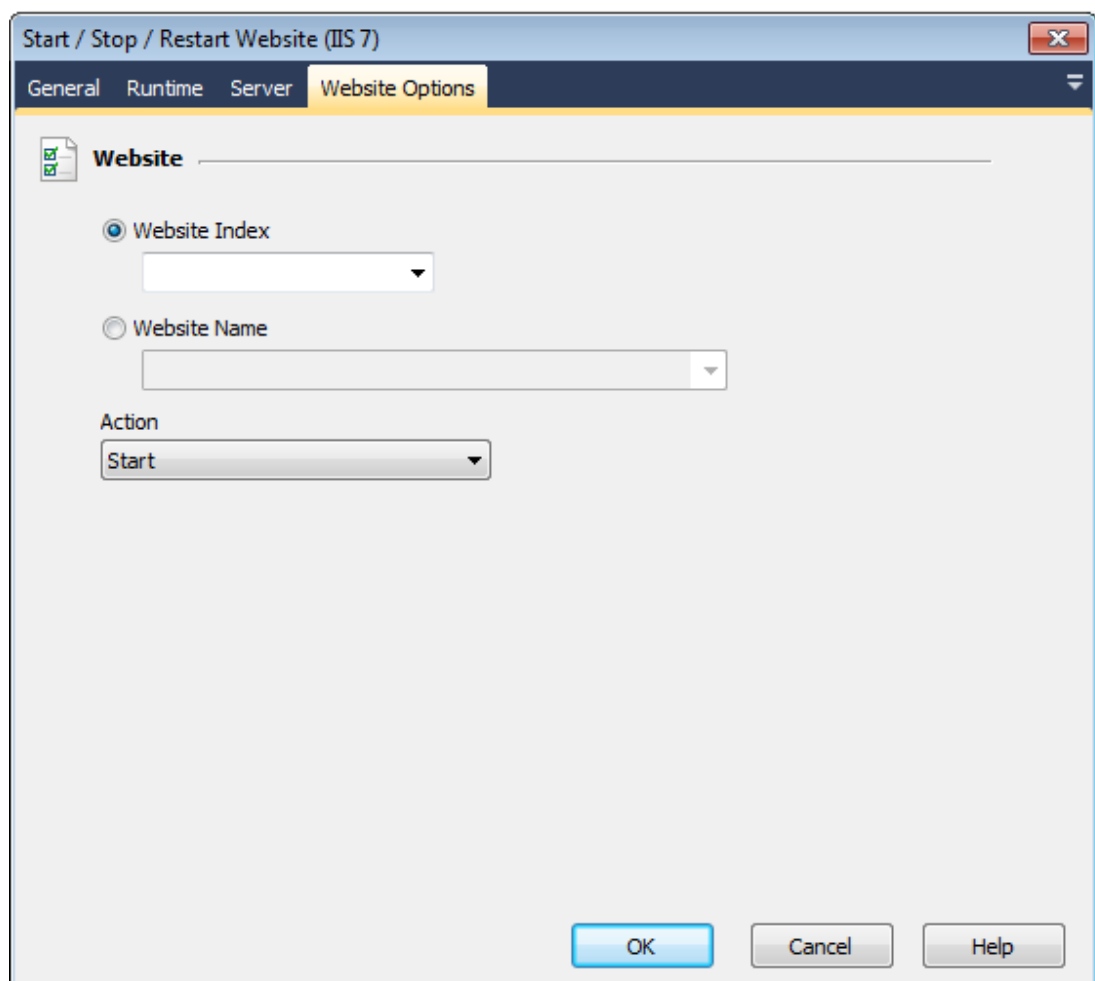
6.20.4.4 Delete Website Action

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.

6.20.4.5 Start / Stop / Restart Website

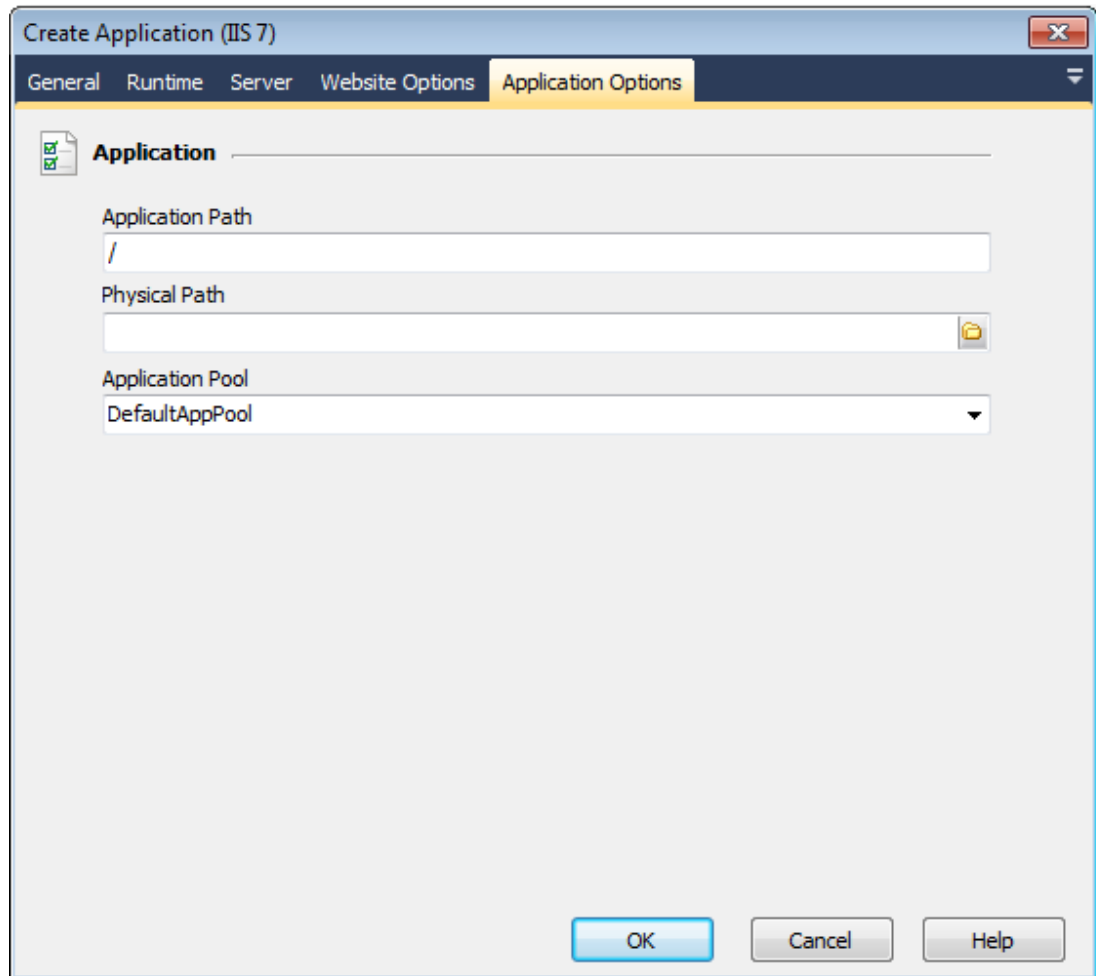
The Start / Stop / Restart Website (IIS 7) action enables you to change the current state of an existing website.



You can choose to start, stop or restart the web site by selected the relevant action in the drop down list.

6.20.4.6 Create Application

The Create Application (IIS 7) action enables you to create a new application under an existing website.

**Application Path**

The path of the new application.

Physical Path

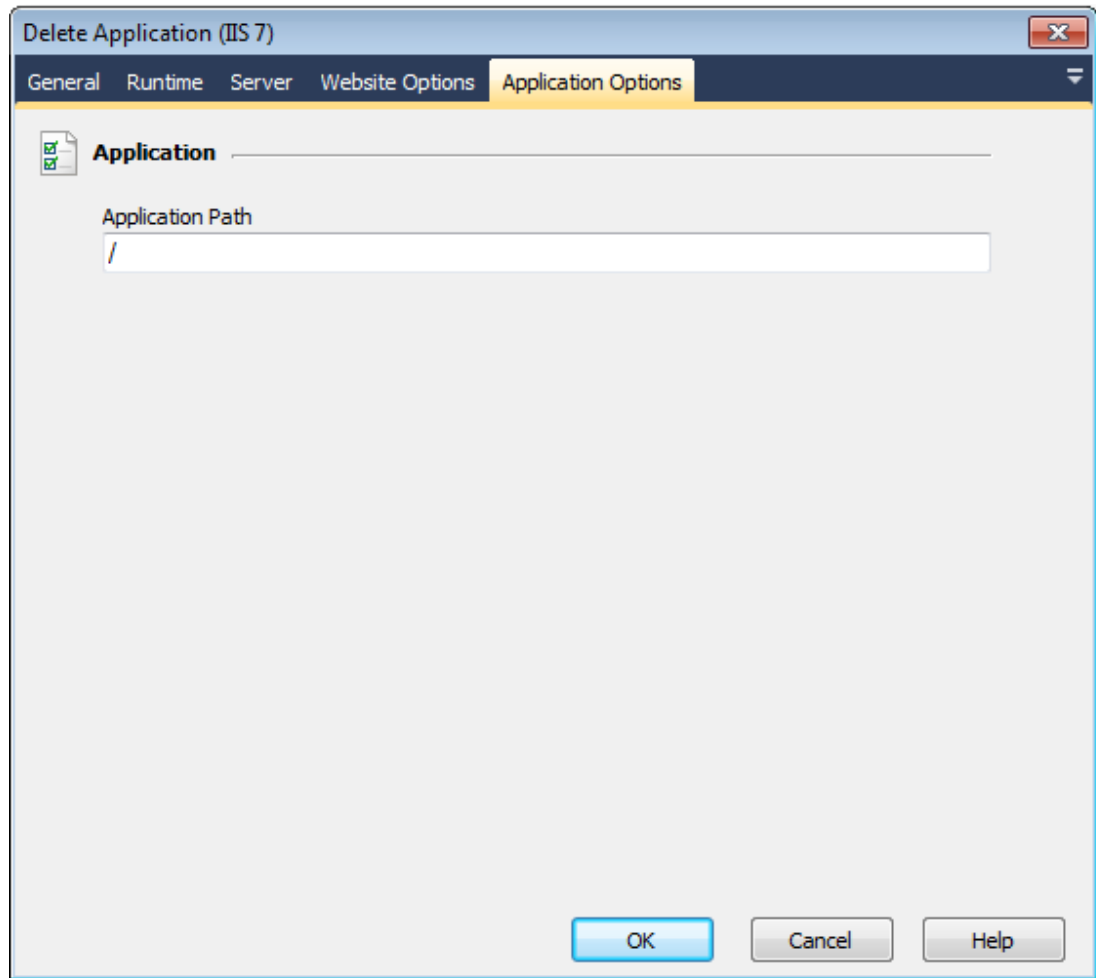
An optional value that specifies the physical path for the application's root virtual directory.

Application Pool

The application pool which the new application will use, the default is the 'DefaultAppPool'.

6.20.4.7 Delete Application

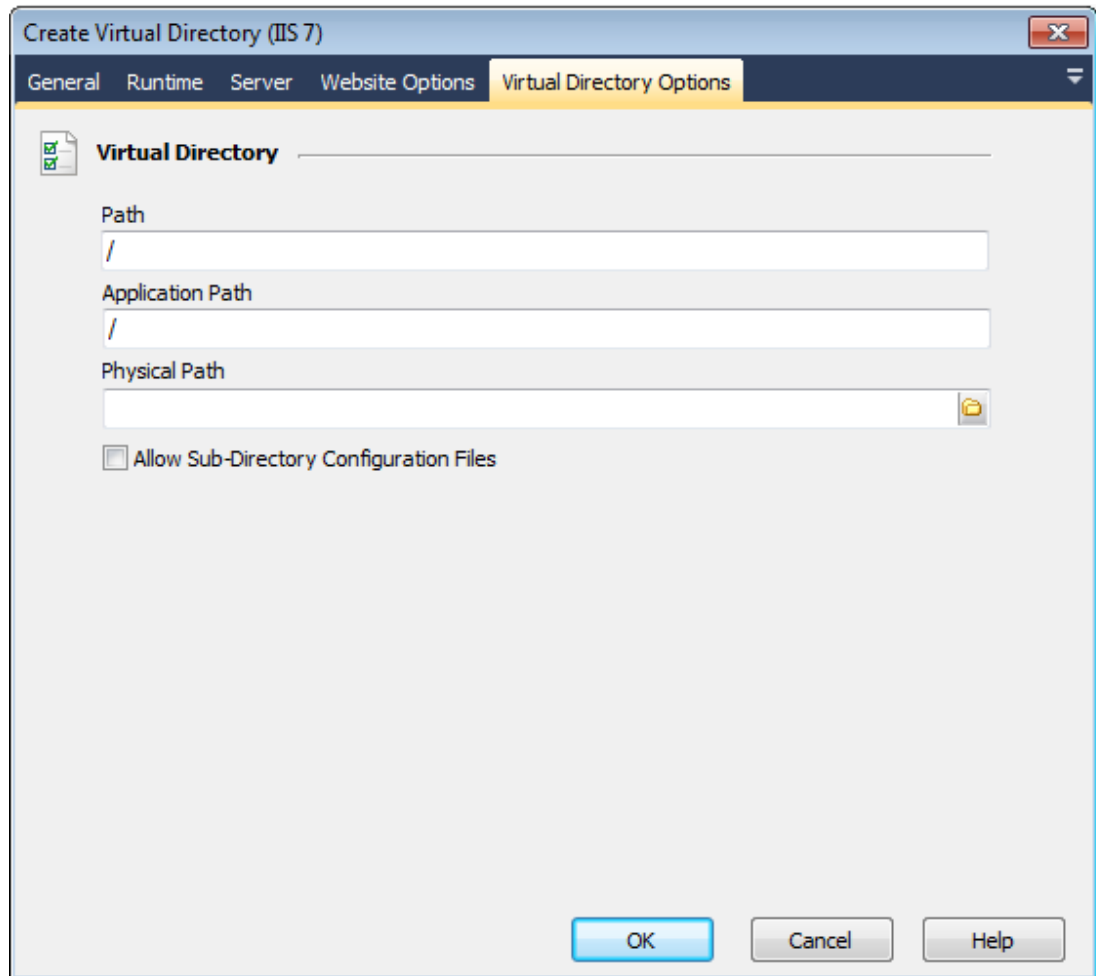
The Delete Application (IIS 7) action enables you to delete an existing application on a website.

**Application Path**

The path to the application that you wish to delete.

6.20.4.8 Create Virtual Directory

The Create Virtual Directory (IIS 7) action enables you to create a new virtual directory on an existing website.

**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

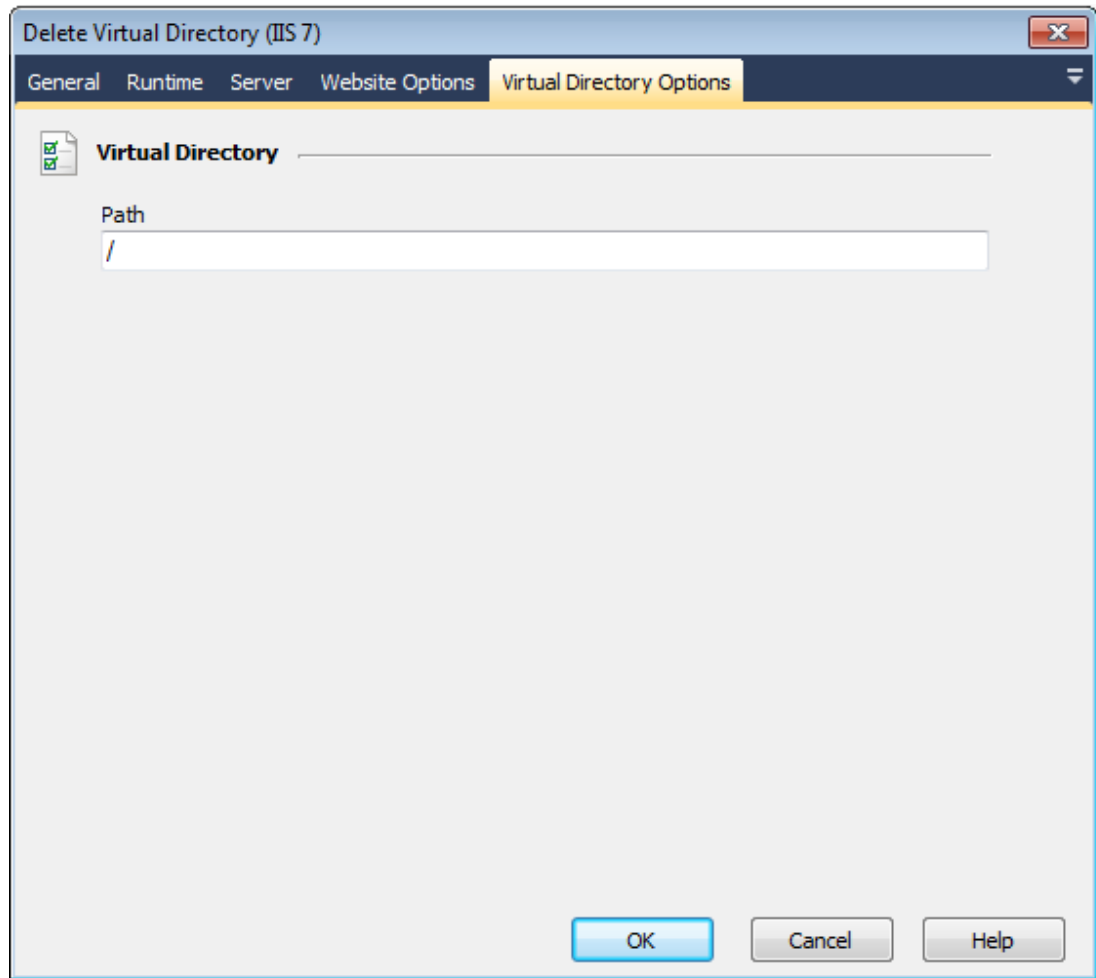
The physical path to the file directory where the content for the virtual directory is stored.

Allow Sub-Directory Configuration Files

Determines whether IIS will look for web.config in content directories lower than the current level.

6.20.4.9 Delete Virtual Directory

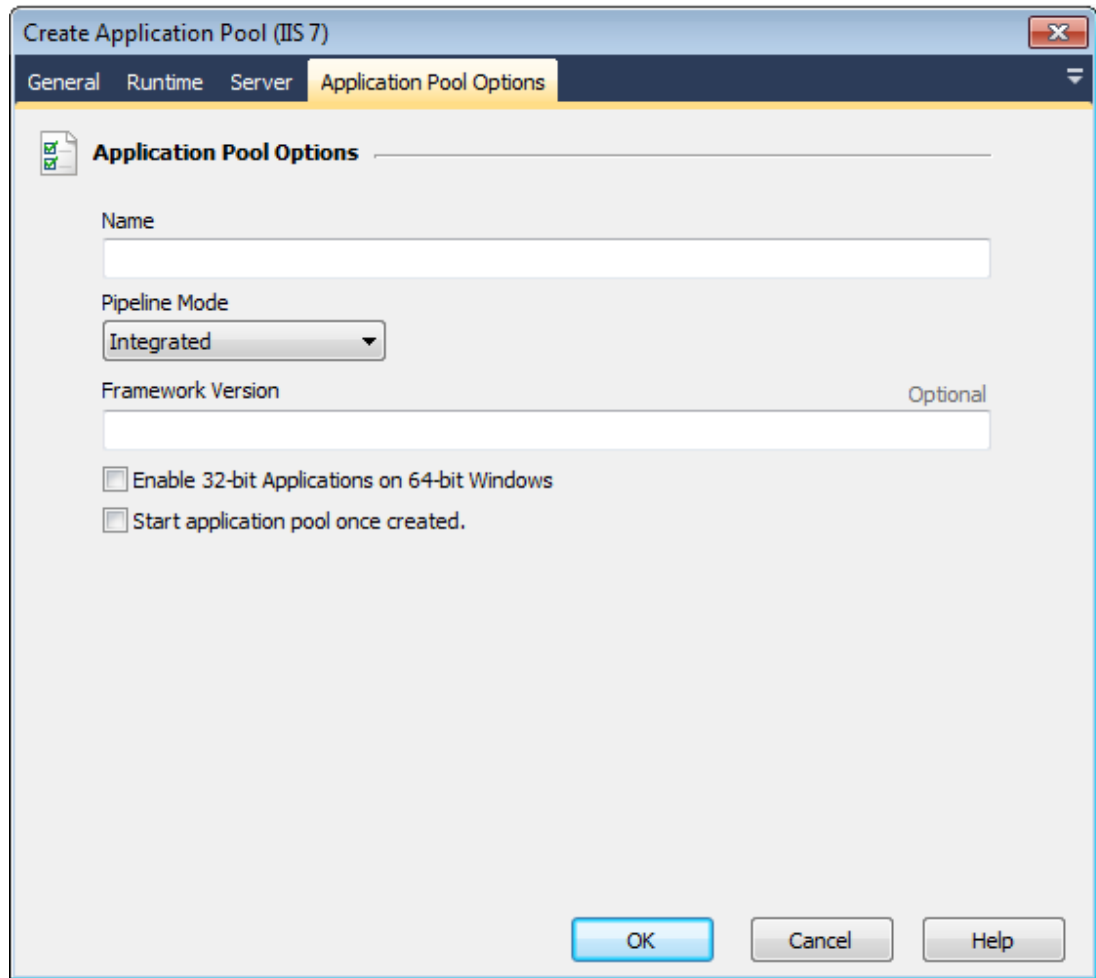
The Delete Virtual Directory (IIS 7) action enables you to delete an existing virtual directory from a website.

**Path**

The path of the virtual directory that you wish to delete.

6.20.4.10 Create Application Pool

The Create Application Pool (IIS 7) actions enables you to create a new application pool on an IIS server.

**Name**

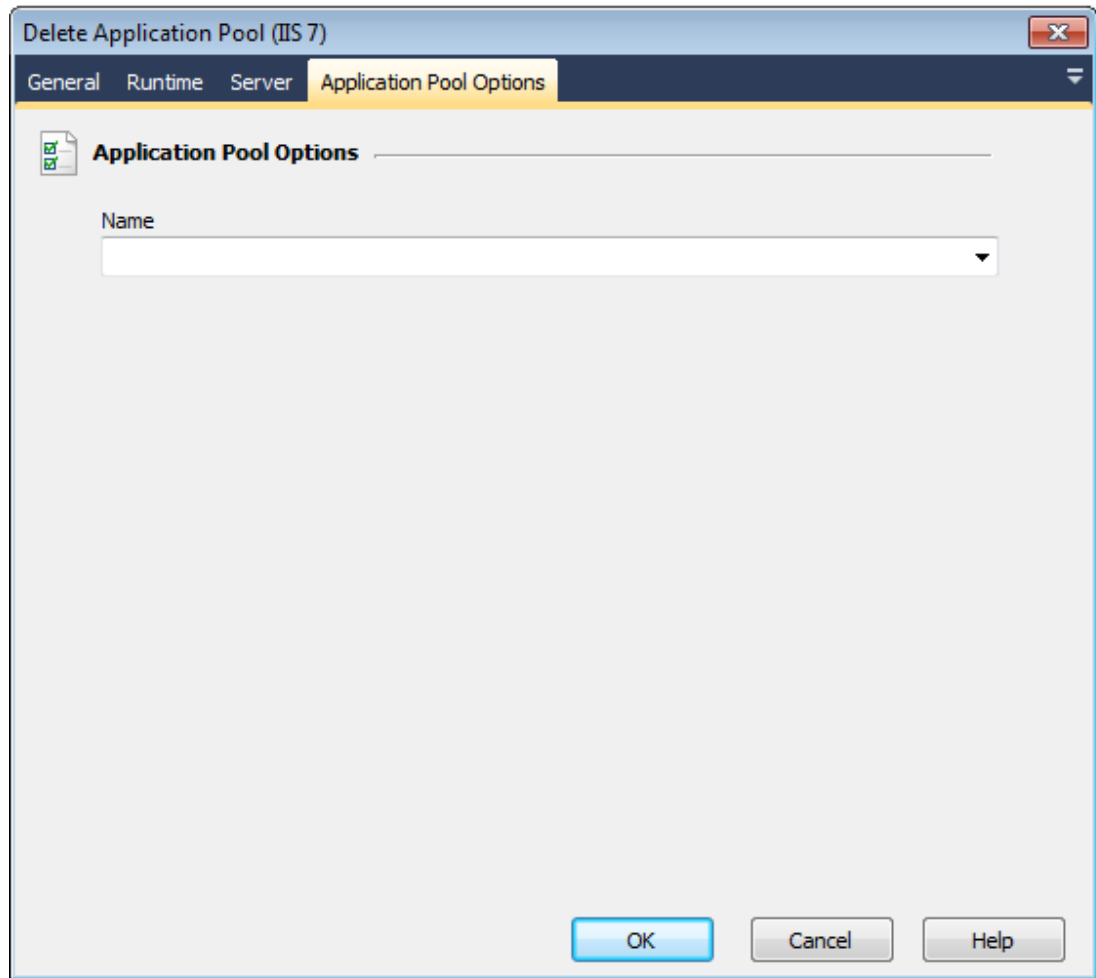
The name that you wish to give the new application pool. This name must be unique on the server.

Start application pool once created

Determines whether the application pool will be started once it has been created.

6.20.4.11 Delete Application Pool

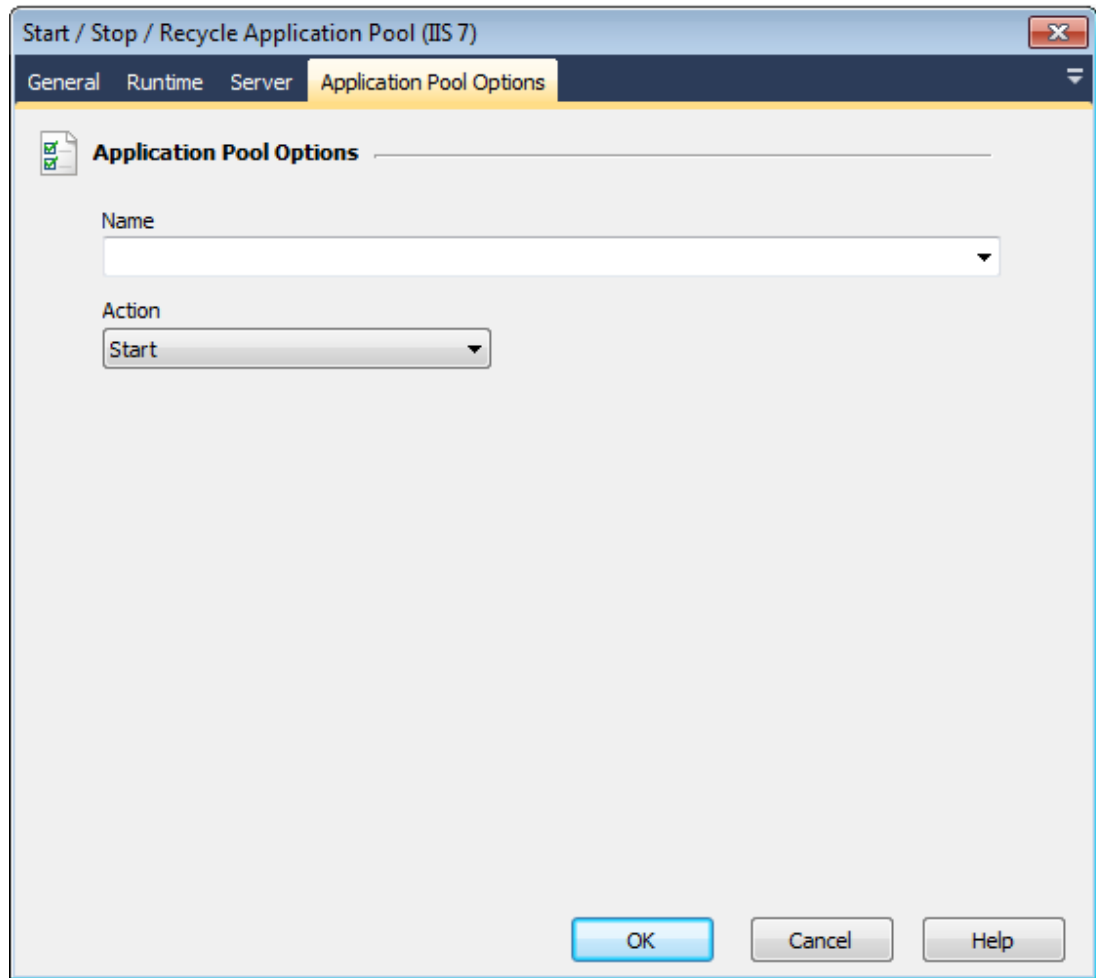
The Delete Application Pool (IIS 7) options enables you to delete an existing application pool on an IIS 7 server.

**Name**

The name of the existing application pool that is to be deleted.

6.20.4.12 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.

**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.

6.21 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	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Iteration 5 of 6
 Do Something	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Completed x 5
 Do Something Else	<input checked="" type="checkbox"/>	<input type="checkbox"/>	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.
Visual Studio.NET Solution Iterator	Iterate through each project in a Visual Studio.NET Solution file.

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

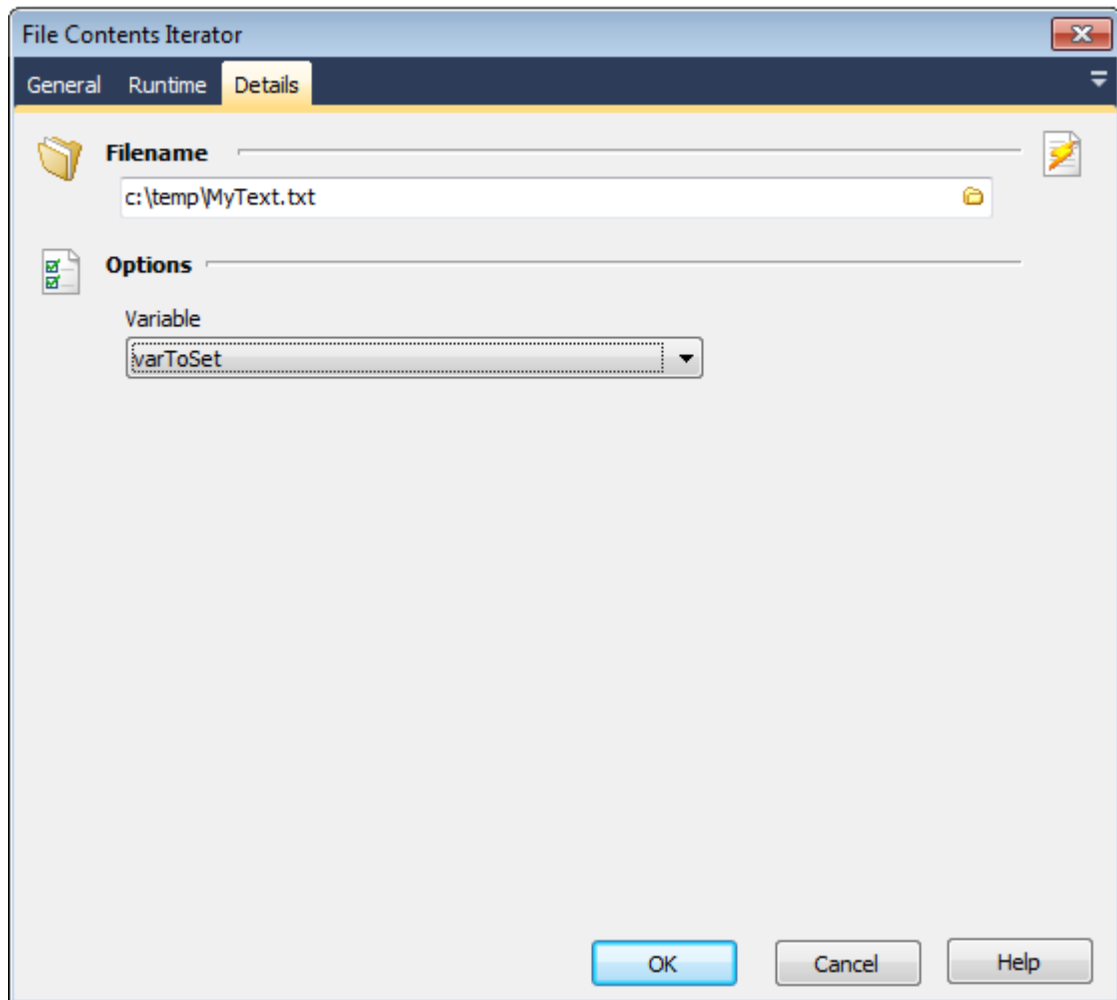
This script event fires the first time the action is run, before the variable has been set to its first value. If the iterator relies on some variables to generate the list of iterated values, you can set them from the OnFirstRun event.

This is equivalent to having a Run Script action placed immediately before the iterator action.

6.21.1 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 Iterator action.



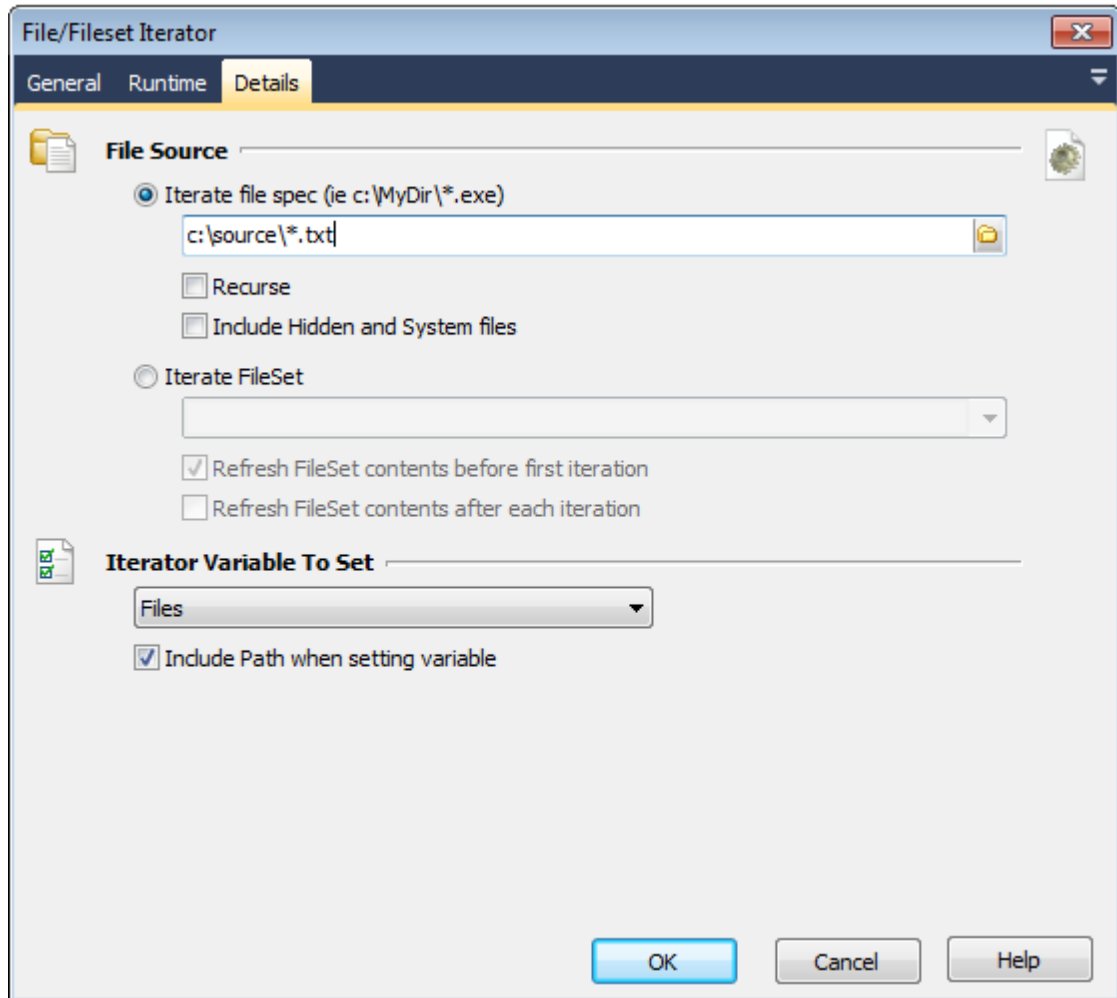
Filename - the name of the text file to read

Variable - the FinalBuilder variable to place the current line of the file in for each iteration

6.21.2 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.



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 FinalBuilder Project/User Variable. On each iteration, the variable value will be set to the path/name of each file.

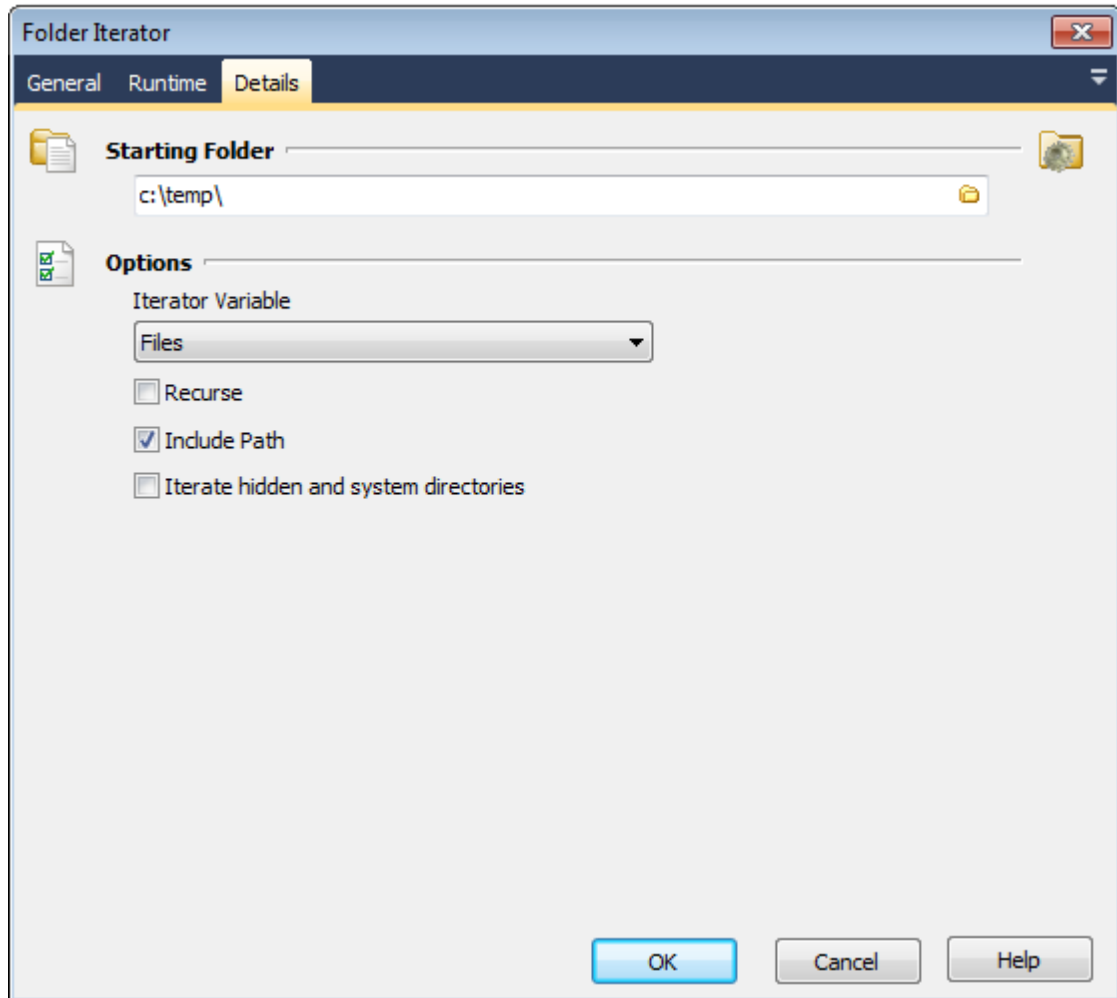
Include Path when setting variable

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.

6.21.3 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.)



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.

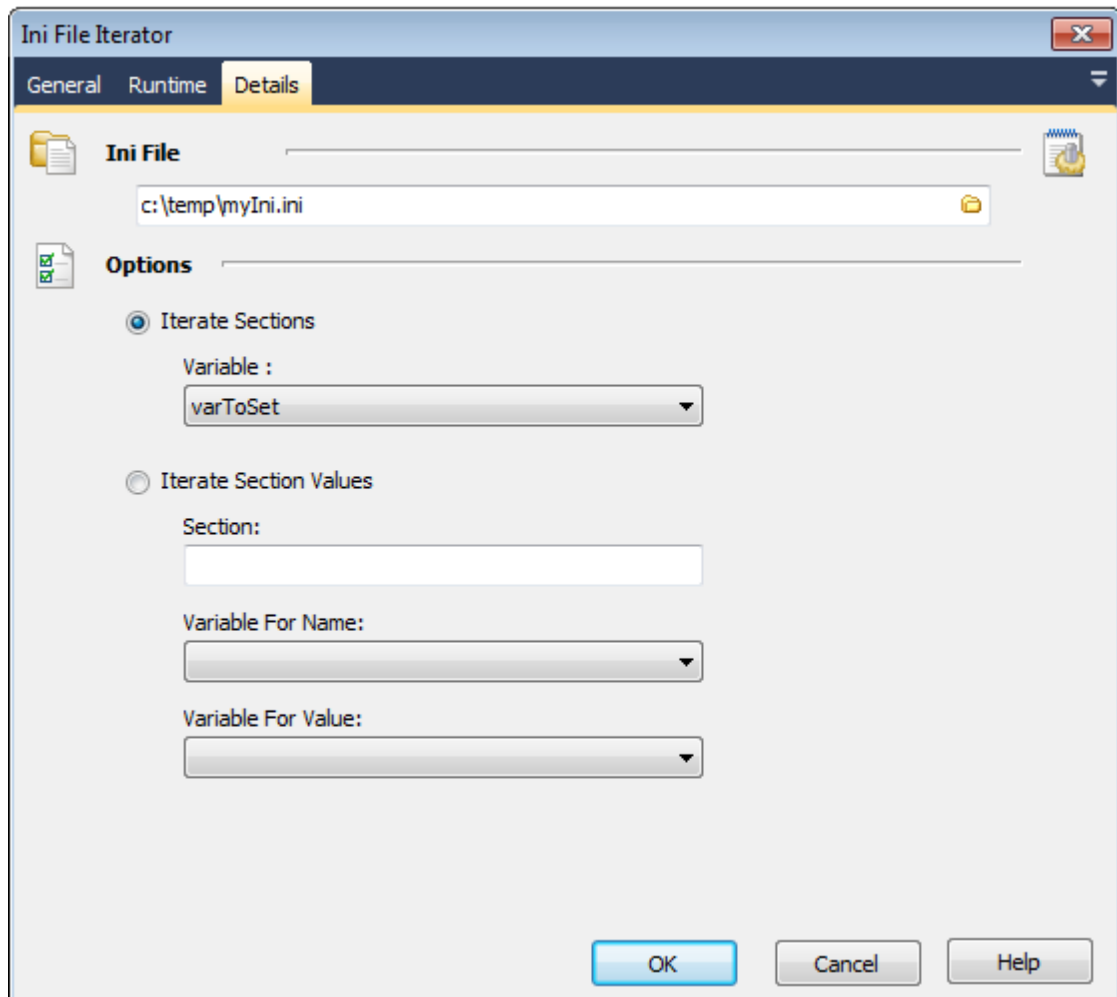
Include Path - If this property is set then the variable will be set to the complete path to the folder. Eg. "c:\temp\myfolder", instead of just the folder name ie "myfolder"

Iterate hidden and system directories - Set this option to also include directories marked with the hidden and/or system attributes.

6.21.4 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 - Specify the INI file to read

Iterate Sections - Iterates for each section found in the INI file.

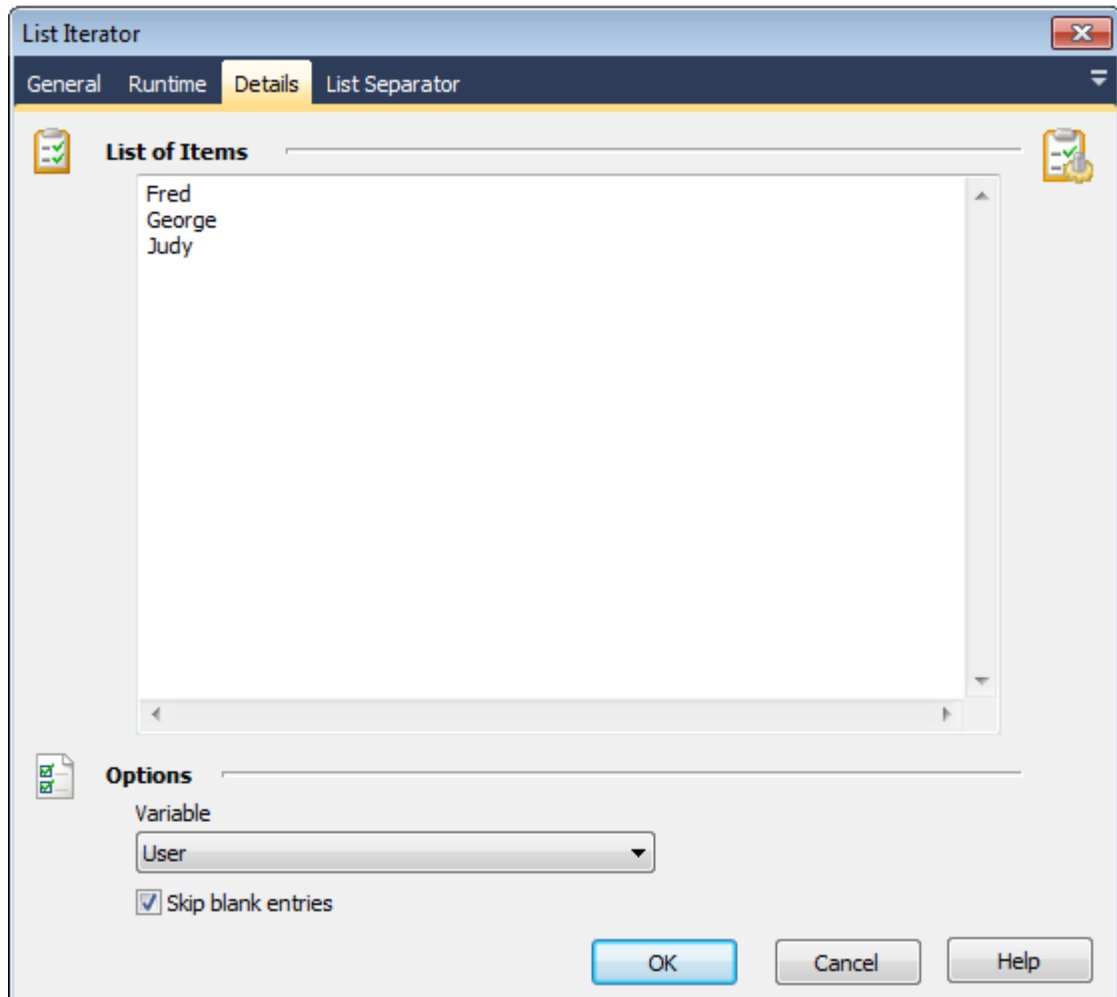
Iterate Section Values - You need to specify which section in the INI file, and then the action will iterate for each Name=Value pair found in the specified section

See also INI file format information

6.21.5 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

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 its child actions), then to George (and run its child actions) and then to Judy (and run its child actions).

FinalBuilder Variables (ie %ListContents%) can be used as part (or all) of the list.

Variable

The name of the FinalBuilder Project/User Variable to set for each value.

Skip Blank Entries

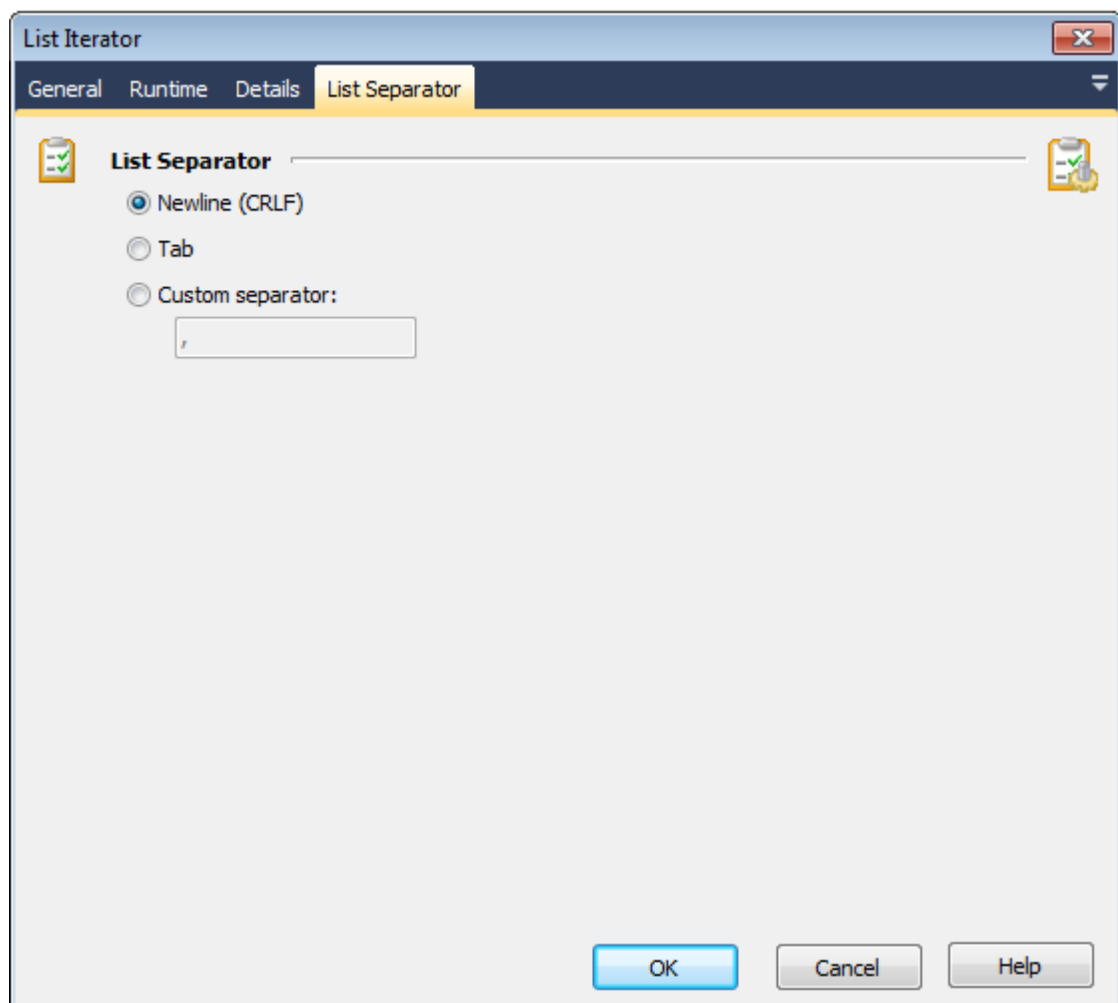
If this option is checked, empty lines will be skipped.

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



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.)

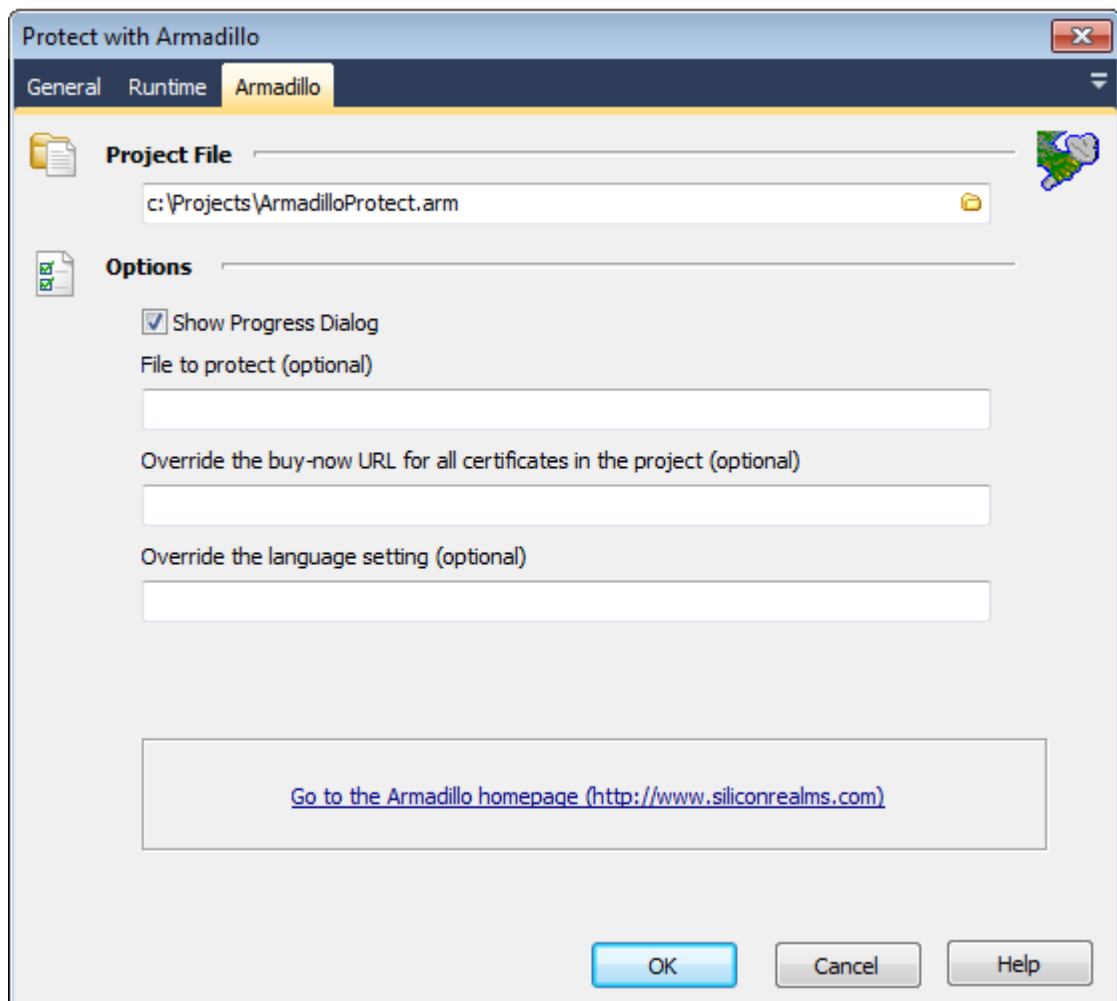
Note that if you use a list separator other than newline, scripting of the ListOfItems property will not work as expected.

6.22 Licensing Tools

6.22.1 Armadillo Action

This action provides support for the Armadillo Software protection system. For more info on Armadillo, please visit the Silicon Realms website - <http://www.siliconrealms.com> Note that to use this action you must set the Armadillo location in the options dialog.

This action was written and kindly provided by Peter Thörnqvist.

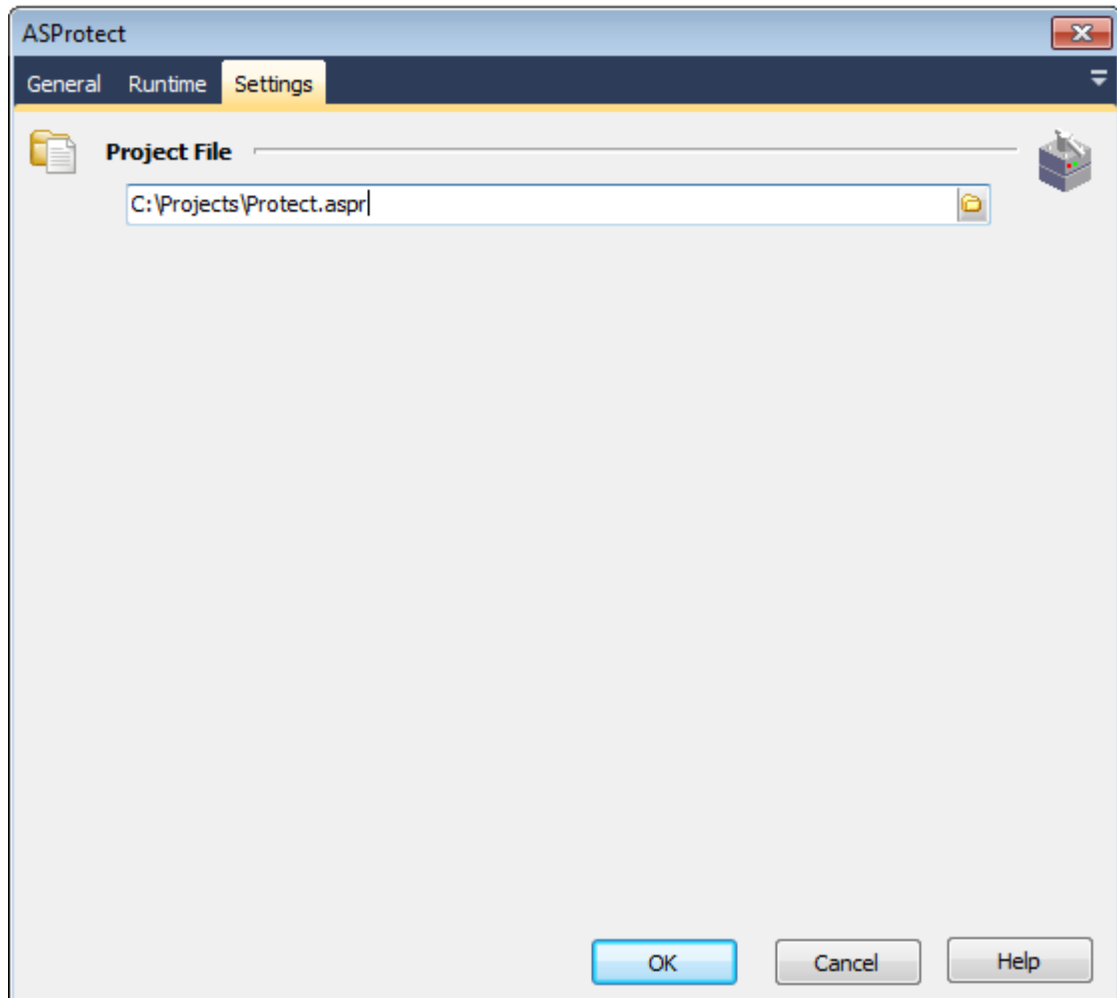


Armadillo Options (in Tools menu | Options)

Armadillo Location - set the location to Armadillo Command (ArmCLine.exe) or Armadillo.exe

6.22.2 ASProtect Action

This Action provides support for the ASProtect Software protection product. For more information on ASProtect, please visit the ASPack website <http://www.aspack.com>



Scripting Info

The Action properties available are :

property ProjectFile : String;

ASProtect Options (in Tools menu | Options)

ASProtect Location - set the location to ASProtect.exe

The Options Object is exposed to scripting as:

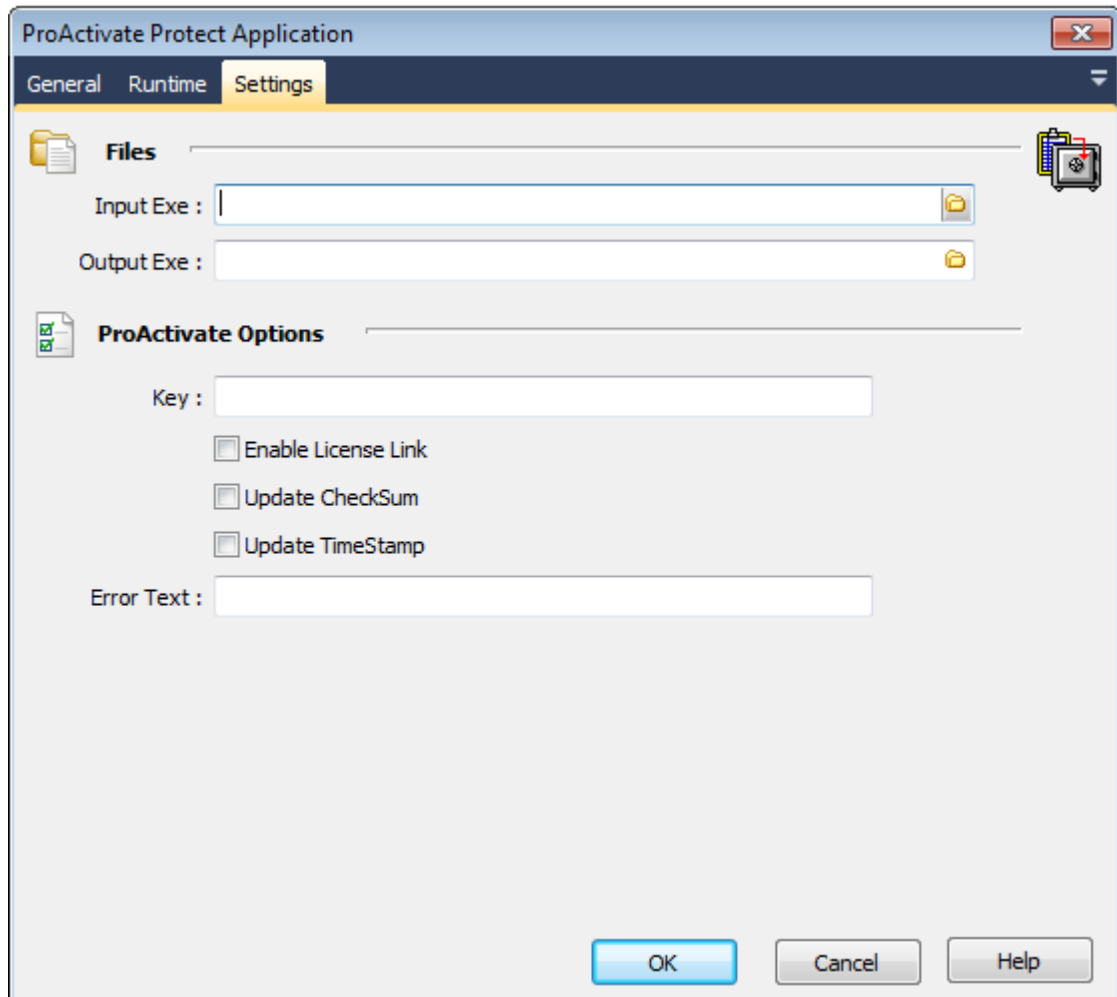
function ASProtectOptions : IFBASProtectOptions

It has one property:

property ASProtectLocation: string

6.22.3 ProActivate Action

This Action provides support for TurboPower's ProActivate software Licensing product. For more information on ProActivate please visit the TurboPower website <http://www.turbopower.com>



6.23 Localization Tools

6.23.1 Sisulizer

Sisulizer Build action builds target files with Sisulizer Enterprise's SLMake utility.

Before using it you should set the path to slmake.exe under Tools- >Options- >Localizations Tools- >Sisulizer.

Parameter Name	Type	Description	Empty Field means
Project Path	Mandatory	This is the most important parameter. It points to the Sisulizer project (.slp) to be build. The project must exist. Hint: Please check if everything works as expected in the Sisulizer GUI before using it from FinalBuilder.	Not valid.
Languages	Optional	This optional parameter select the languages to be build. Of course the languages must exist in the Sisulizer project given in Project Path. Separate multiple codes with semicolon . Code format is either two character ISO-639 language code (e.g. "en"), or two character ISO-3166 country code (e.g. "US").	Build all languages.
Include Source	Optional	This optional parameter select the sources to be build. The sources must exist in the Sisulizer project given in Project Path. Separate multiple sources to include with semicolon . Please enter the source names as printed in Sisulizer's object tree, e.g. mypath\myapp.exe	Include all sources.
Exclude Source	Optional	This optional parameter select the sources not to be build. The sources must exist in the Sisulizer project given in Project Path. Separate multiple sources to include with semicolon . Please enter the source names as printed in Sisulizer's object tree, e.g. mypath\myapp.exe Hint: Sometimes you want to exclude only one or two sources. You could list all sources to include in Include Source	Exclude no source.

		but this list can be rather long and then hard to maintain.	
--	--	---	--

Quick reference ISO-639 codes:

AA = Afar; AB = Abkhazian; AF = Afrikaans; AM = Amharic; AR = Arabic; AS = Assamese; AY = Aymara; AZ = Azerbaijani; BA = Bashkir; BE = Byelorussian; BG = Bulgarian; BH = Bihari; BI = Bislama; BN = Bengali / Bangla; BO = Tibetan; BR = Breton; CA = Catalan; CO = Corsican; CS = Czech; CY = Welsh; DA = Danish; DE = German; DZ = Bhutani; EL = Greek; EN = English / American; EO = Esperanto; ES = Spanish; ET = Estonian; EU = Basque; FA = Persian; FI = Finnish; FJ = Fiji; FO = Faeroese; FR = French; FY = Frisian; GA = Irish; GD = Gaelic" = Scots Gaelic; GL = Galician; GN = Guarani; GU = Gujarati; HA = Hausa; HI = Hindi; HR = Croatian; HU = Hungarian; HY = Armenian; IA = Interlingua; IE = Interlingue; IK = Inupiak; IN = Indonesian; IS = Icelandic; IT = Italian; IW = Hebrew; JA = Japanese; JI = Yiddish; JW = Javanese; KA = Georgian; KK = Kazakh; KL = Greenlandic; KM = Cambodian; KN = Kannada; KO = Korean; KS = Kashmiri; KU = Kurdish; KY = Kirghiz; LA = Latin; LN = Lingala; LO = Laothian; LT = Lithuanian; LV = Latvian / Lettish; MG = Malagasy; MI = Maori; MK = Macedonian; ML = Malayalam; MN = Mongolian; MO = Moldavian; MR = Marathi; MS = Malay; MT = Maltese; MY = Burmese; NA = Nauru; NE = Nepali; NL = Dutch; NO = Norwegian; OC = Occitan; OM = Oromo / Afan; OR = Oriya; PA = Punjabi; PL = Polish; PS = Pashto / Pushto; PT = Portuguese; QU = Quechua; RM = Rhaeto-Romance; RN = Kirundi; RO = Romanian; RU = Russian; RW = Kinyarwanda; SA = Sanskrit; SD = Sindhi; SG = Sangro; SH = Serbo-Croatian; SI = Singhalese; SK = Slovak; SL = Slovenian; SM = Samoan; SN = Shona; SO = Somali; SQ = Albanian; SR = Serbian; SS = Siswati; ST = Sesotho; SU = Sudanese; SV = Swedish; SW = Swahili; TA = Tamil; TE = Tegulu; TG = Tajik; TH = Thai; TI = Tigrinya; TK = Turkmen; TL = Tagalog; TN = Setswana; TO = Tonga; TR = Turkish; TS = Tsonga; TT = Tatar; TW = Twi; UK = Ukrainian; UR = Urdu; UZ = Uzbek; VI = Vietnamese; VO = Volapuk; WO = Wolof; XH = Xhosa; YO = Yoruba; ZH = Chinese; ZU = Zulu

Copyright © 2007 Sisulizer Ltd & Co KG. Reproduced here with permission.

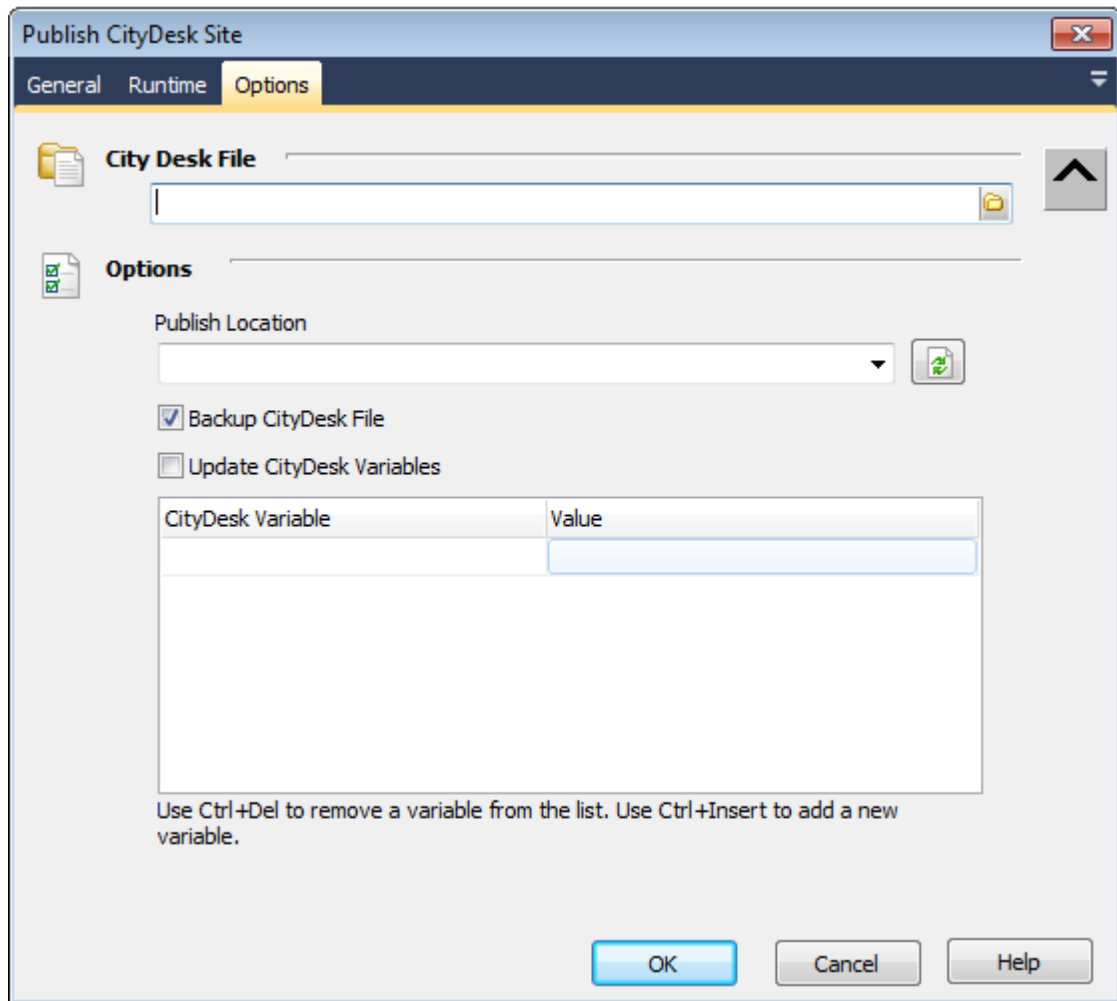
For more information on Sisulizer, visit <http://www.sisulizer.com>

For technical support of slmake.exe please visit the support forum on <http://www.sisulizer.net>

6.24 Misc Actions

6.24.1 CityDesk Action

This Action will publish a [CityDesk 2](#) Site to a predefined location. Before using this action for the first time you will need to set the CityDesk location option in the FinalBuilder Options dialog.



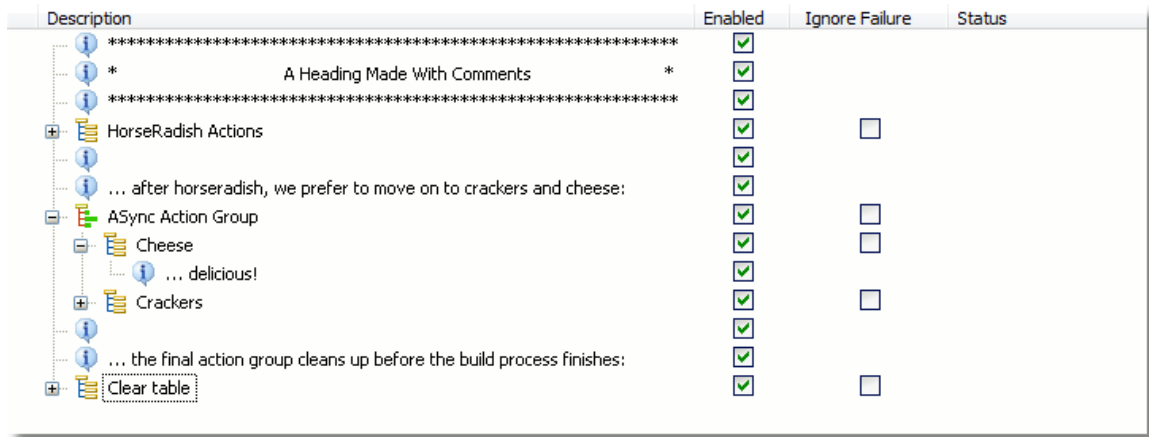
When you select a CityDesk file, FinalBuilder will read the defined publish locations into the combo box. Alternatively, you can type the name of a different publish location.

Note however that when the action runs this field must have a valid location, otherwise CityDesk will display a message box with an error (which will cause unattended build to hang!)

Note : CityDesk 2 Files are Microsoft Access databases, and this action uses the Microsoft Jet 4.x OLEDB provider. This provided is not installed in Windows by default. If you have Office XP/2003 or Access XP/2003 you should already have it, otherwise you can download it from the Microsoft web site.

6.24.2 Comment Action

Comment Actions provide a way to place descriptive text inside action lists.



Description	Enabled	Ignore Failure	Status
***** i ***** * A Heading Made With Comments * i *****	<input checked="" type="checkbox"/>		
HorseRadish Actions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
i ... after horseradish, we prefer to move on to crackers and cheese:	<input checked="" type="checkbox"/>		
ASync Action Group	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Cheese	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
i ... delicious!	<input checked="" type="checkbox"/>		
Crackers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
i ... the final action group cleans up before the build process finishes:	<input checked="" type="checkbox"/>		
Clear table	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Comments do not get executed at run time and do not have script events.

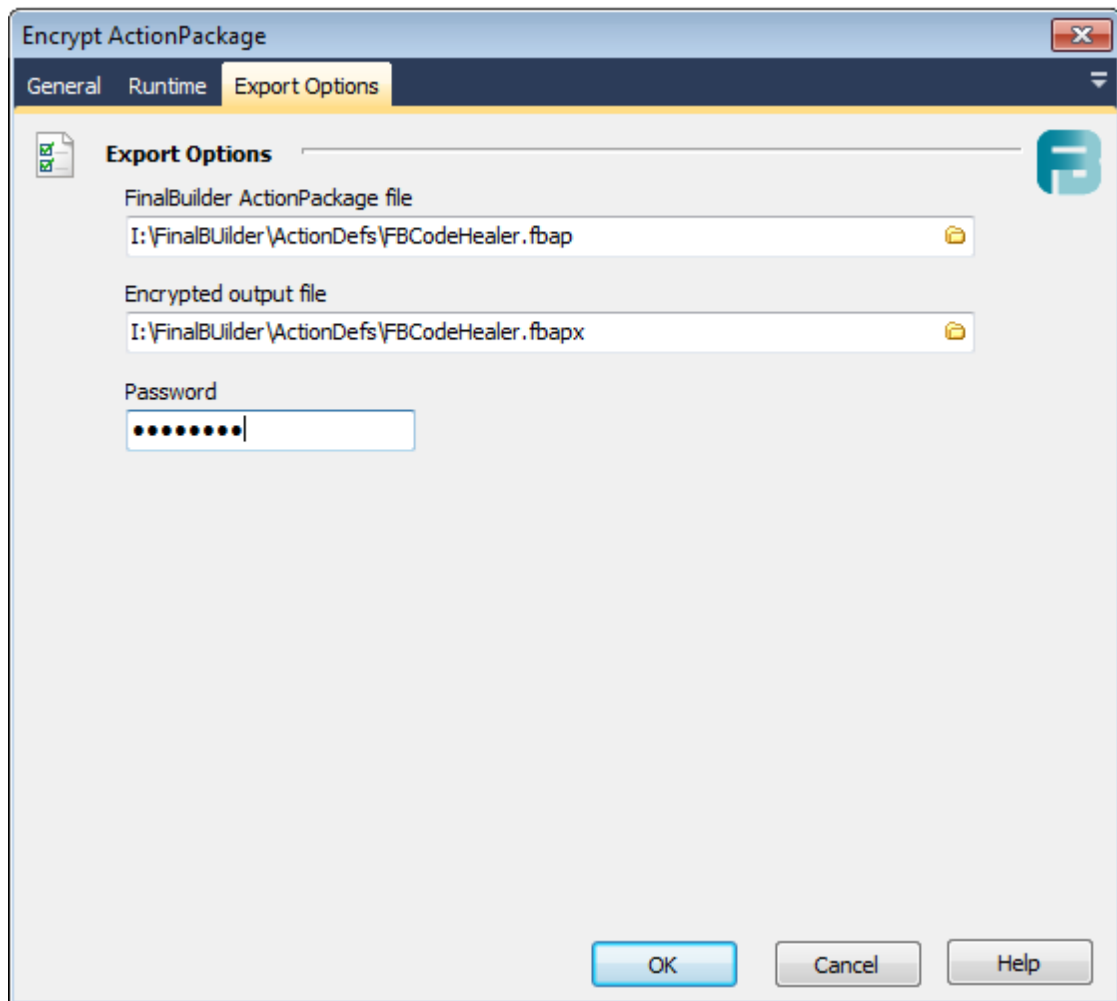
Action Groups can also be used to make your build easy to understand.

6.24.3 Encrypt ActionPackage

[Professional Edition Only]

The Encrypt ActionPackage action allows you to convert a .fbap (standard xml file) into an encrypted ActionPackage (encrypted xml file) which has the extension .fbapx

This can be used to hide the implementation of the ActionPackage from the user, but still allows FinalBuilder to load and use the action.



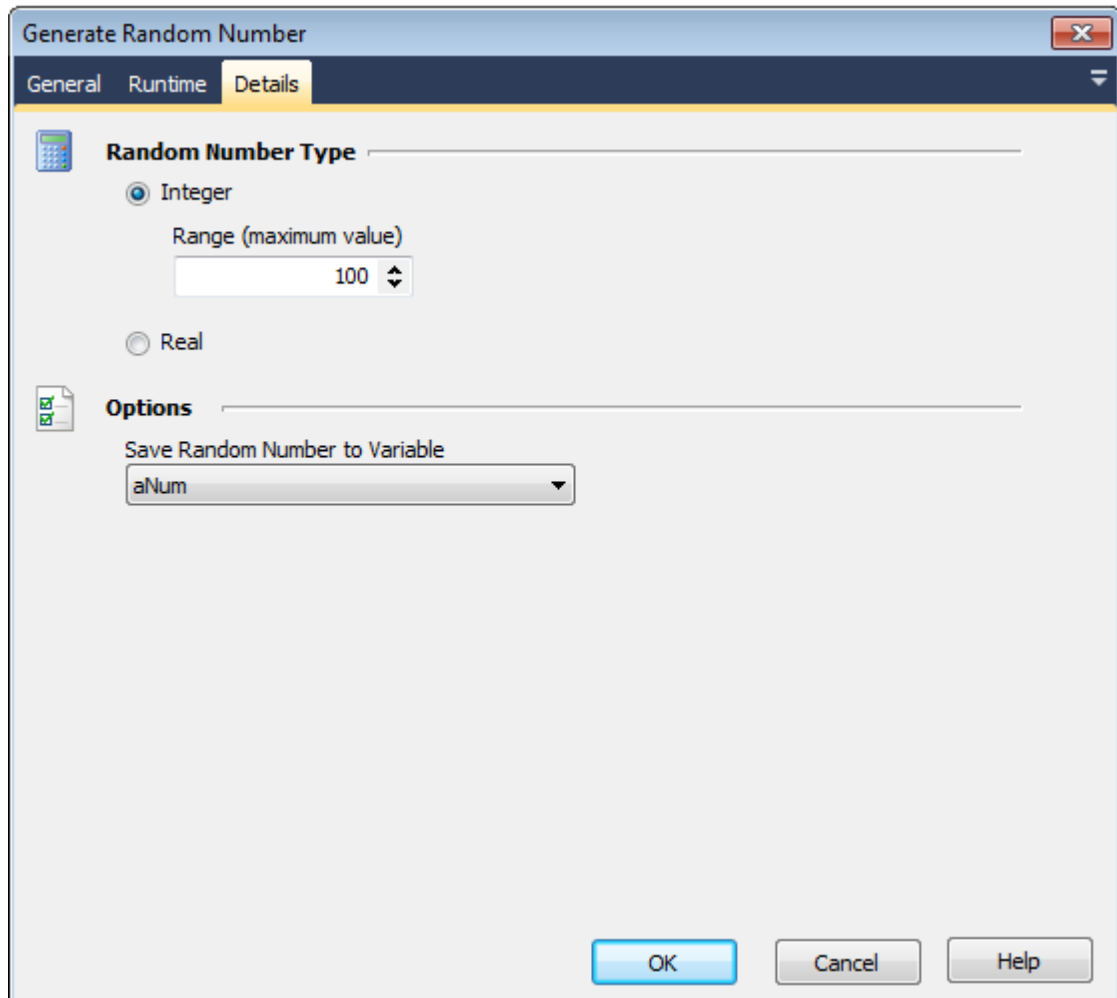
ActionPackage file - the path to the .fbap

Encrypted Output File - the path to where the .fbapx will be saved

Password - the password to encrypt the ActionPackage with. This password will need to be used to decrypt the ActionPackage for editing in ActionStudio.

6.24.4 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

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

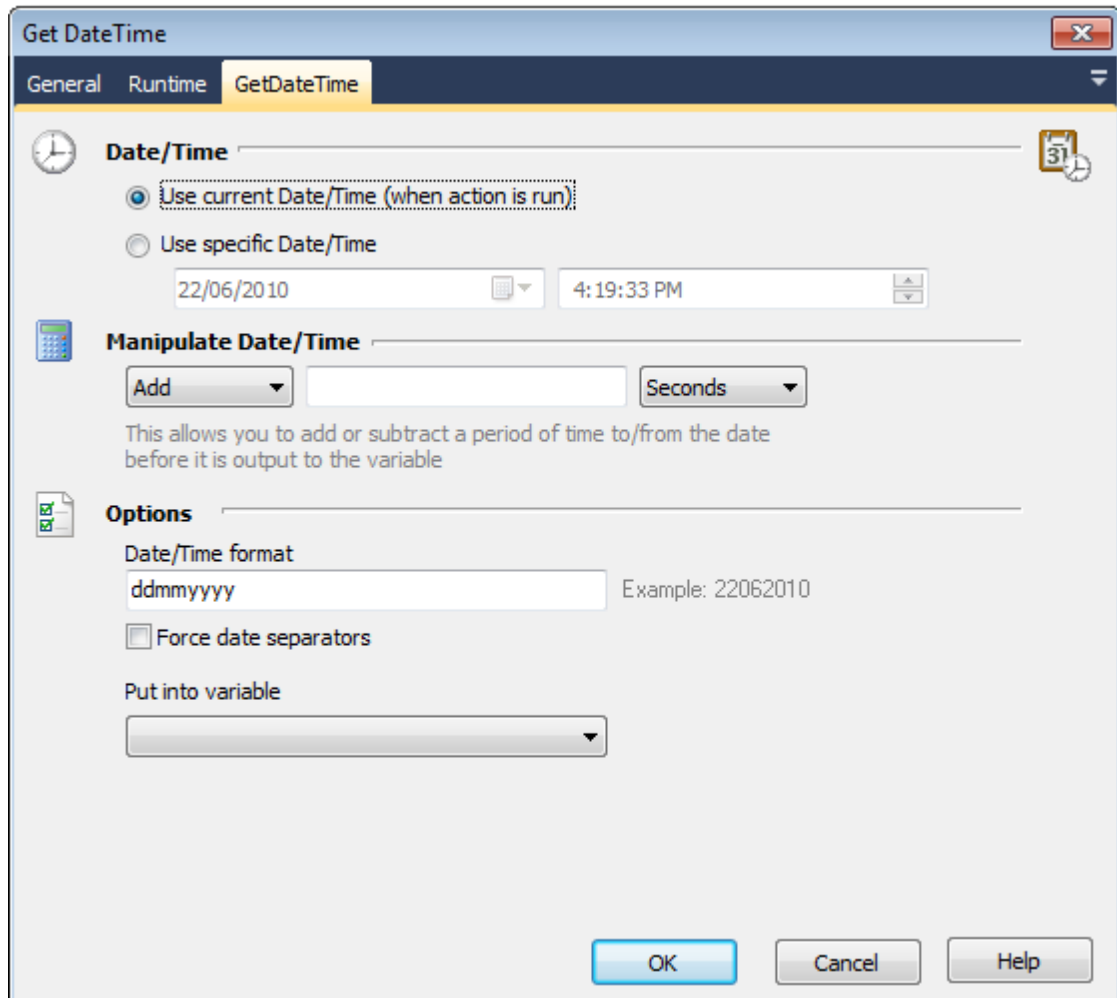
Options

Variable

Specify the variable to save the random number to

6.24.5 Get DateTime Action

This action formats a specific Date or Date/Time value into a FinalBuilder Variable.



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

6.24.6 Image File Manipulation

6.24.6.1 Image Manipulation

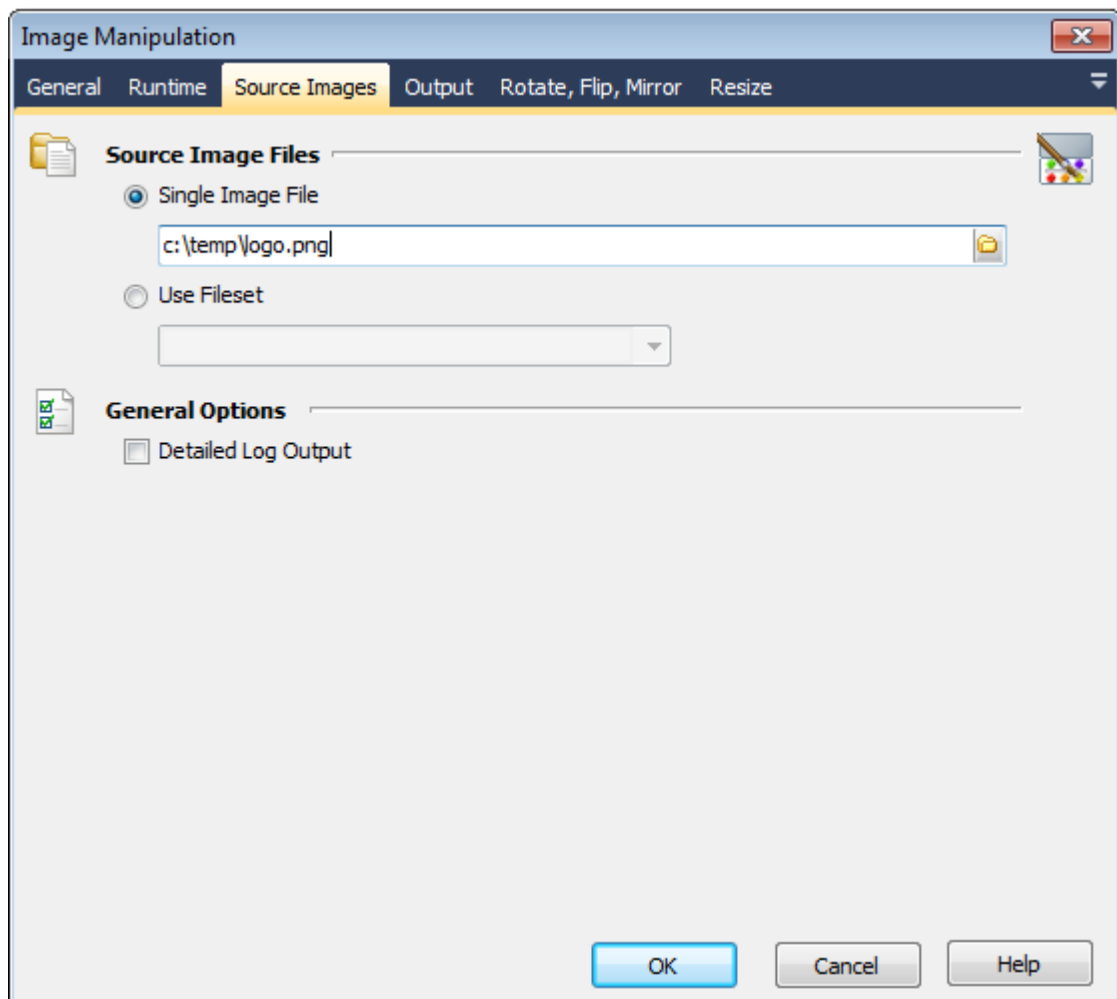
[FinalBuilder Professional Edition]

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)



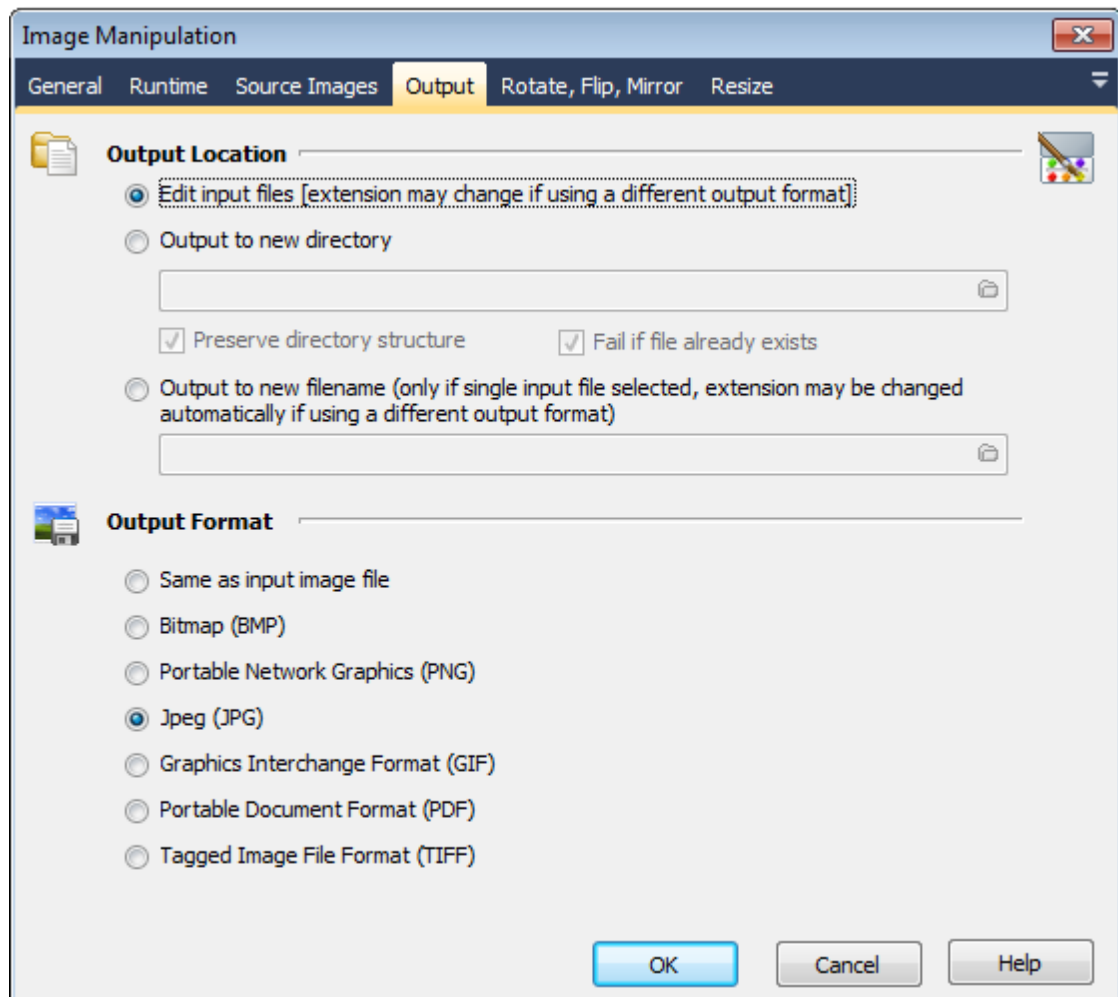
Source Images

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.



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.

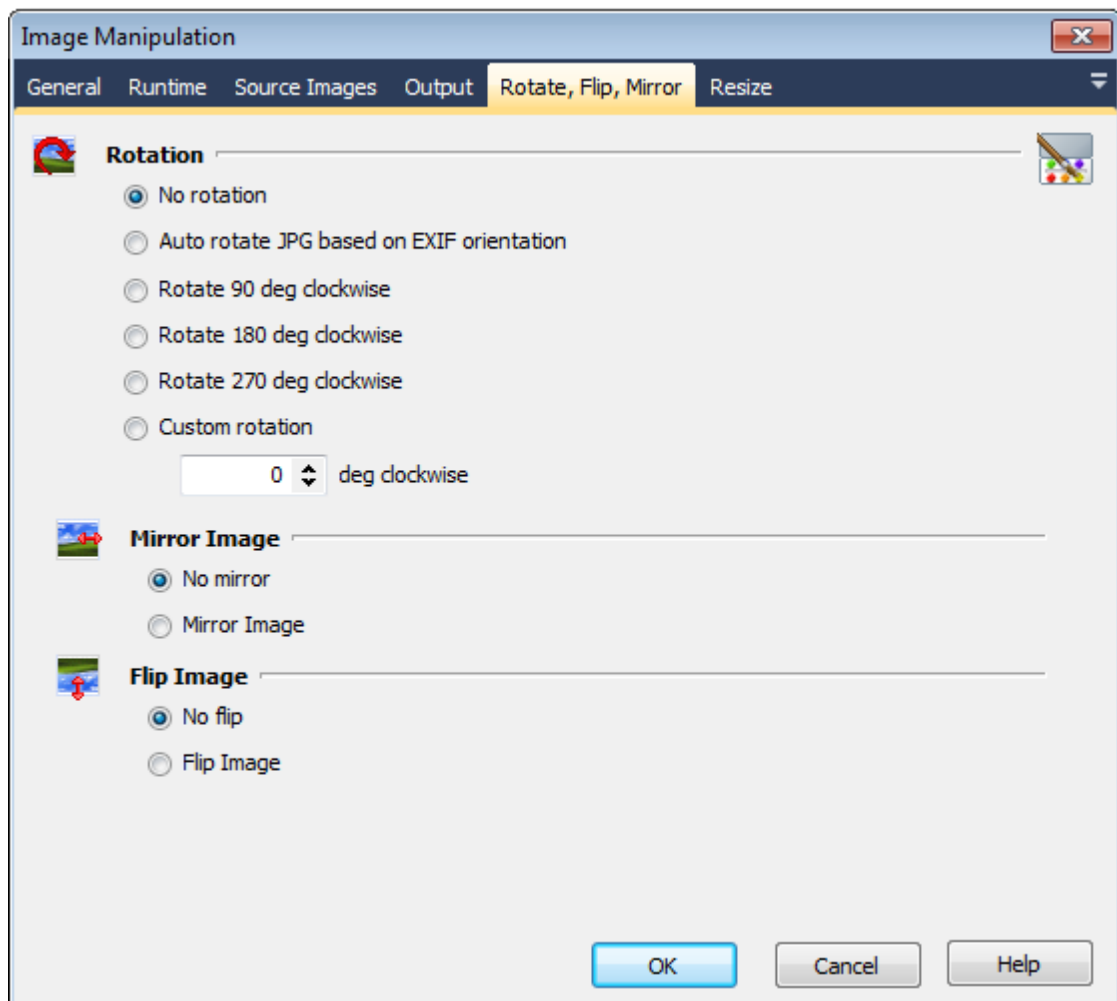
Output to New Directory : This option will leave your source files untouched, and will create new files in the chosen directory. The preserve output directory option can be useful when using a FileSet that includes multiple directories - the directory structure of the source files will be recreated under 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

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

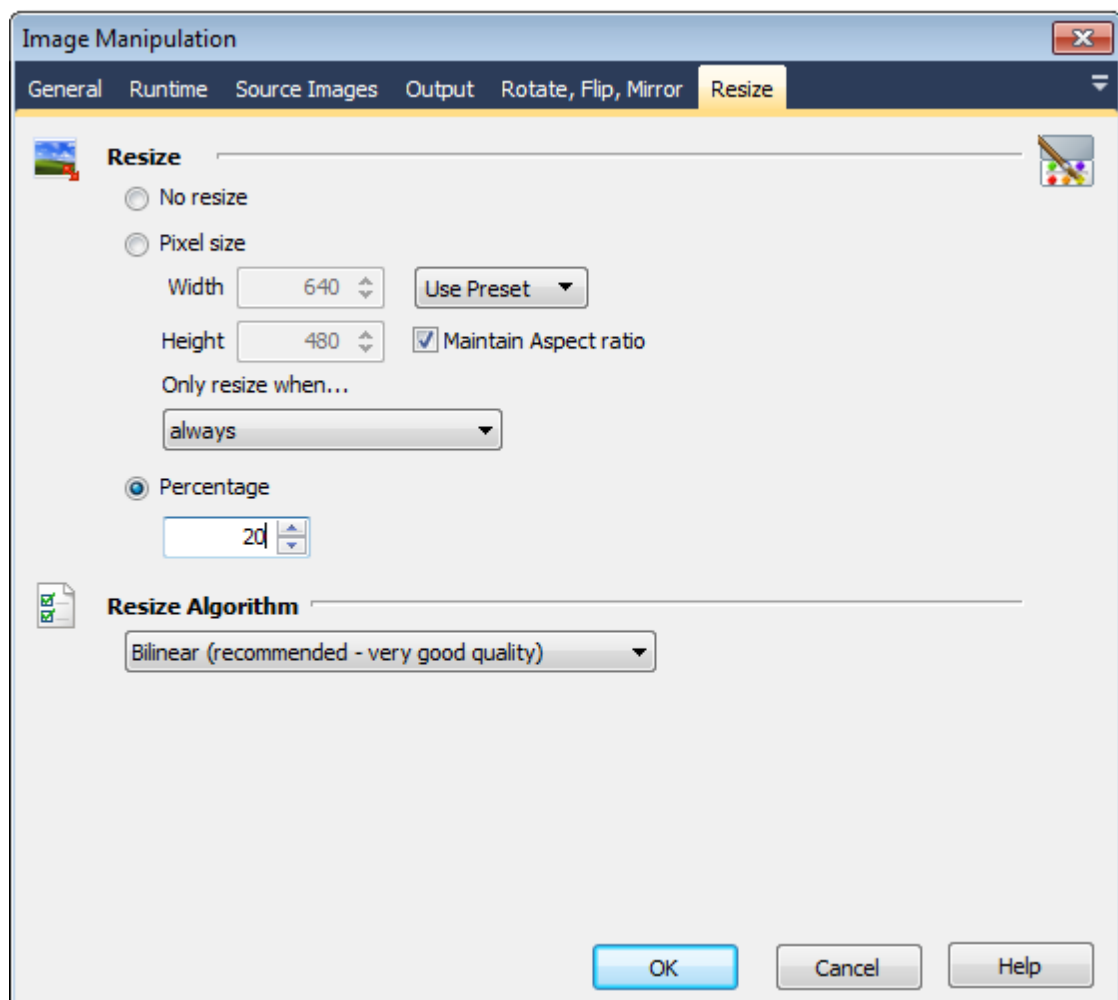
Custom rotation : Rotate the image between 0 and 360 degrees

Mirror Image

Mirroring the image will perform a horizontal flip

Flip Image

Flipping the image will perform a vertical flip



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.

6.24.6.2 Lossless JPEG Transform

[FinalBuilder 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 result 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

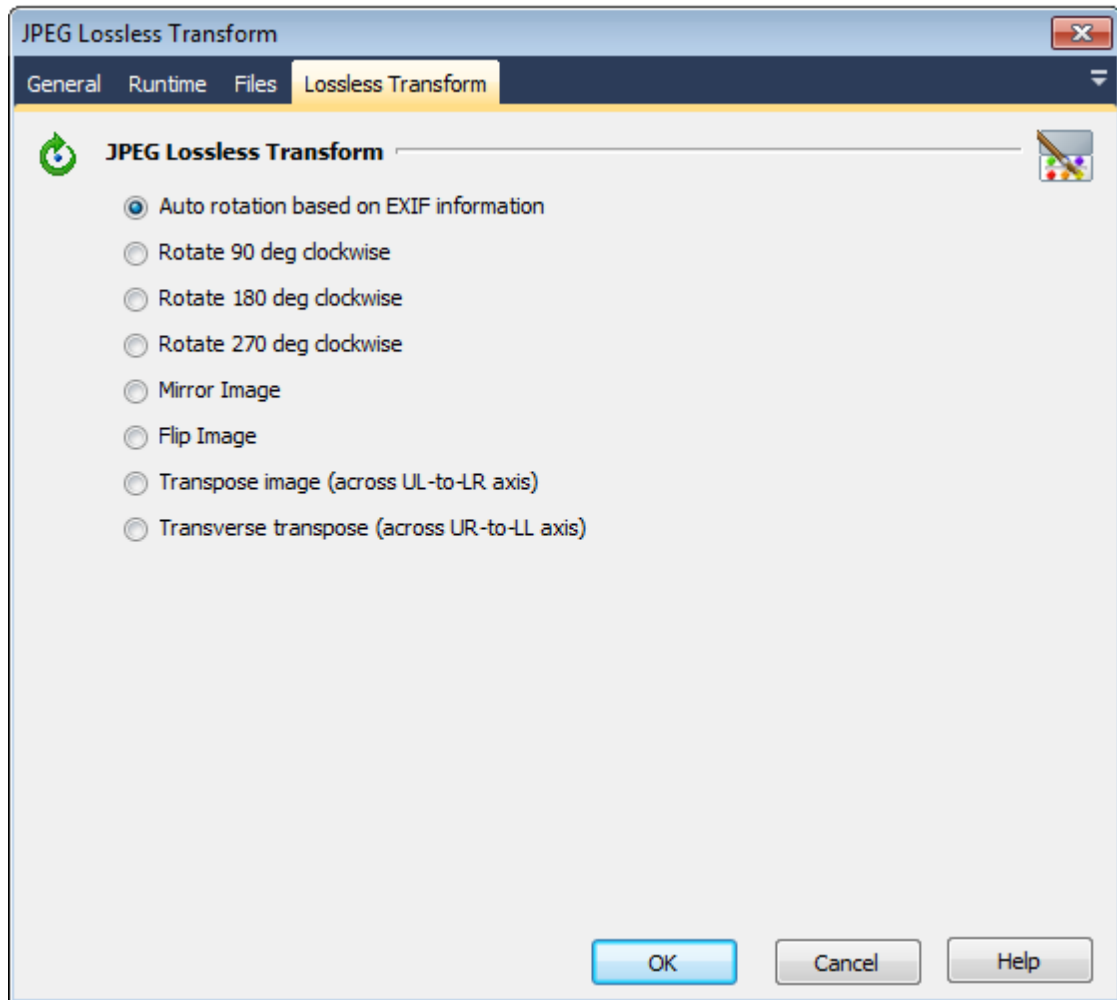
Rotate 270 deg clockwise : Perform a lossless 270 degrees clockwise rotation

Mirror Image : Perform a lossless horizontal flip

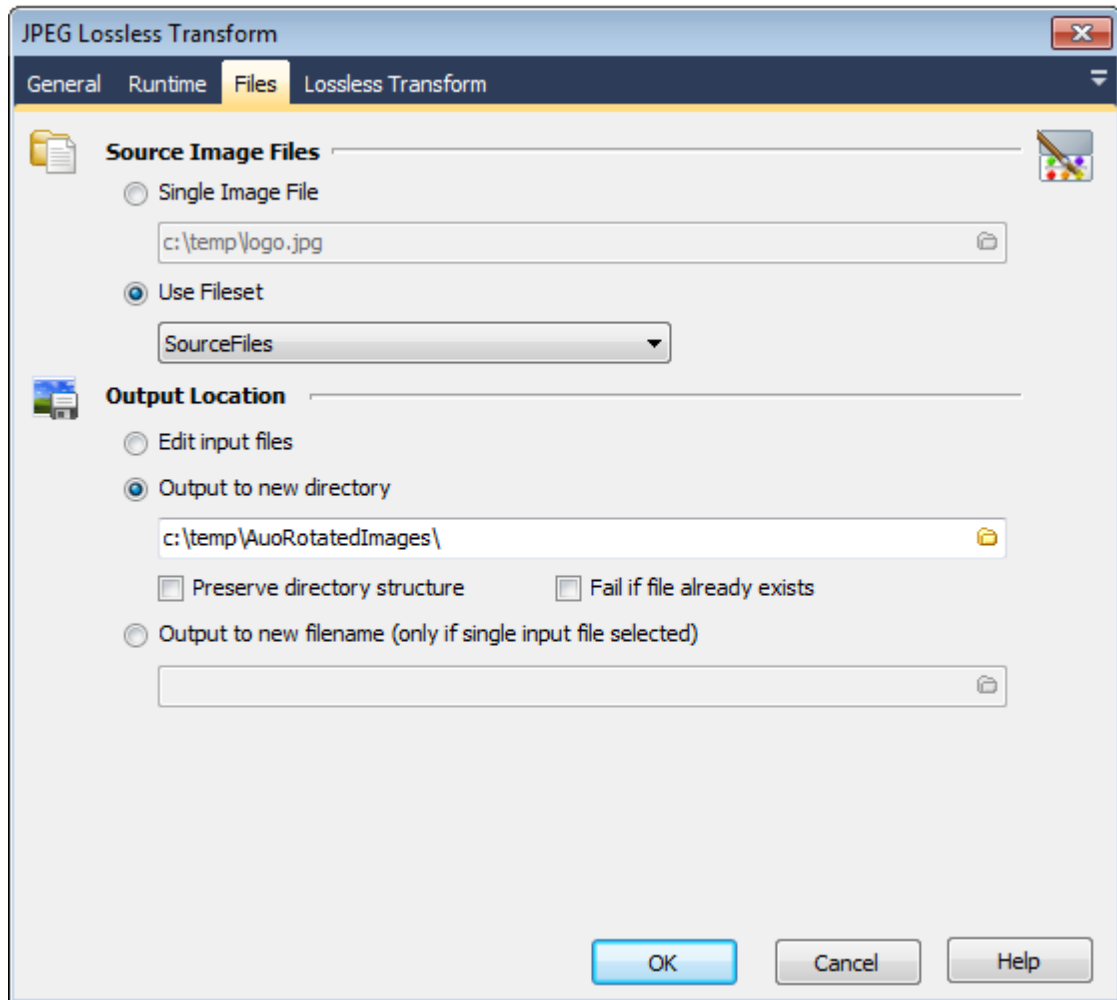
Flip Image : Perform a lossless vertical flip

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



Use the Files tab to specify the source images and the output location



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.

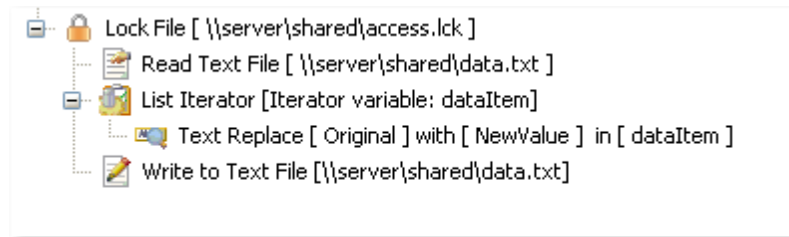
Output to New File Name : This option can only be used when the source image is a single file. This option allows you to rename the output file.

6.24.7 Mutex/Lock File Action

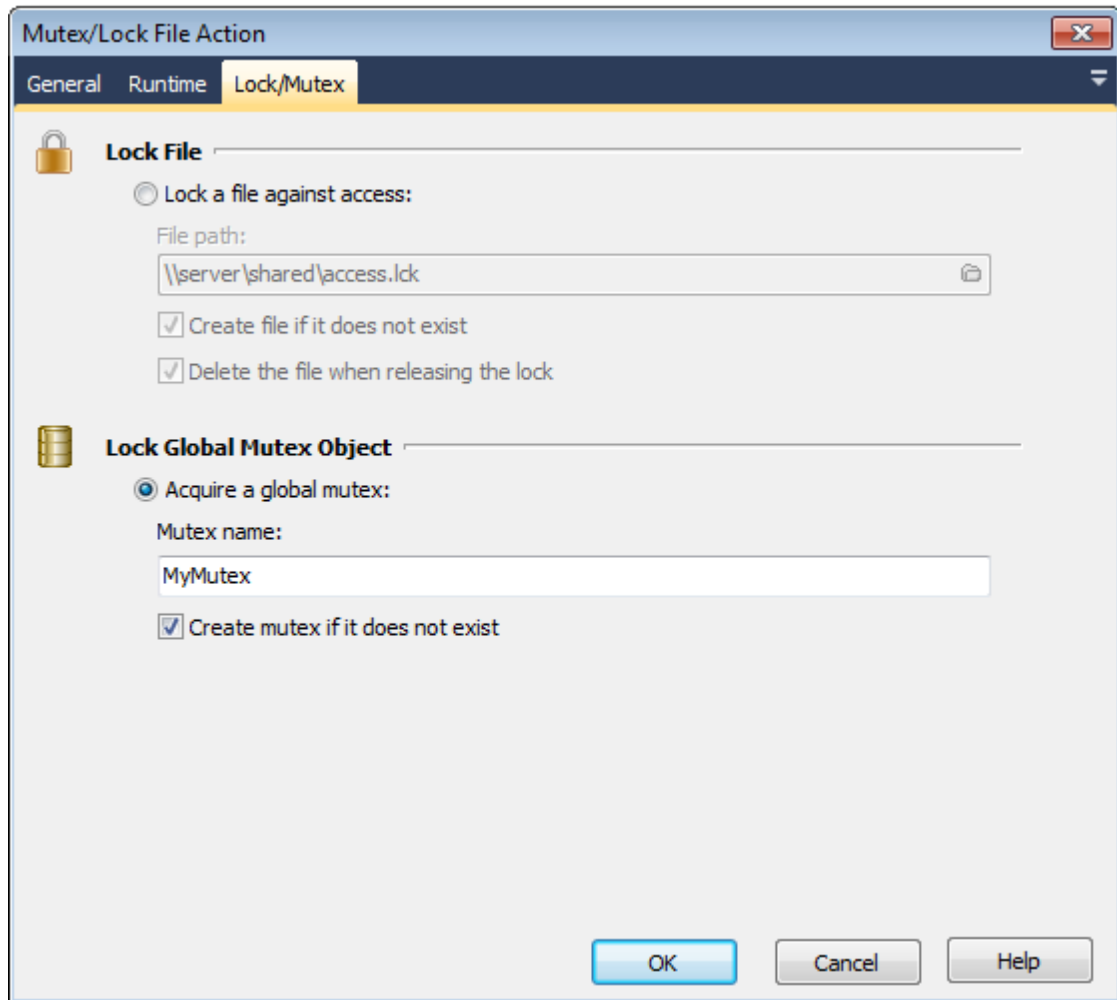
The Mutex/Lock File Action allows you to establish access to an exclusive resource. The resource can be either a lock file or a global Windows mutex object. The action can be used to control exclusive access within a FinalBuilder project, between multiple concurrent instances of FinalBuilder, or between FinalBuilder and a different application.

The exclusive access lasts until all child actions have finished running.

Example



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

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 FinalBuilder will need full access to the network share.

Lock Global Mutex Object

Mutexes are a class of Windows operating system object. FinalBuilder 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.

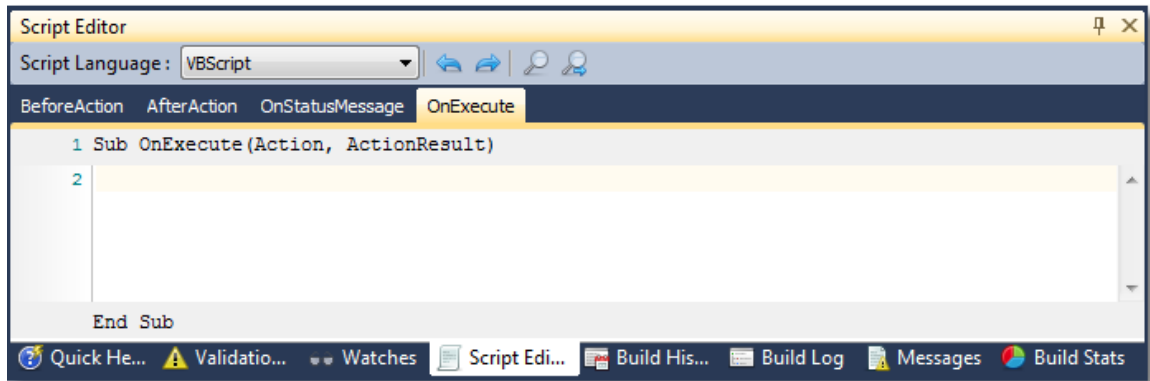
Timeout

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.

6.24.8 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:

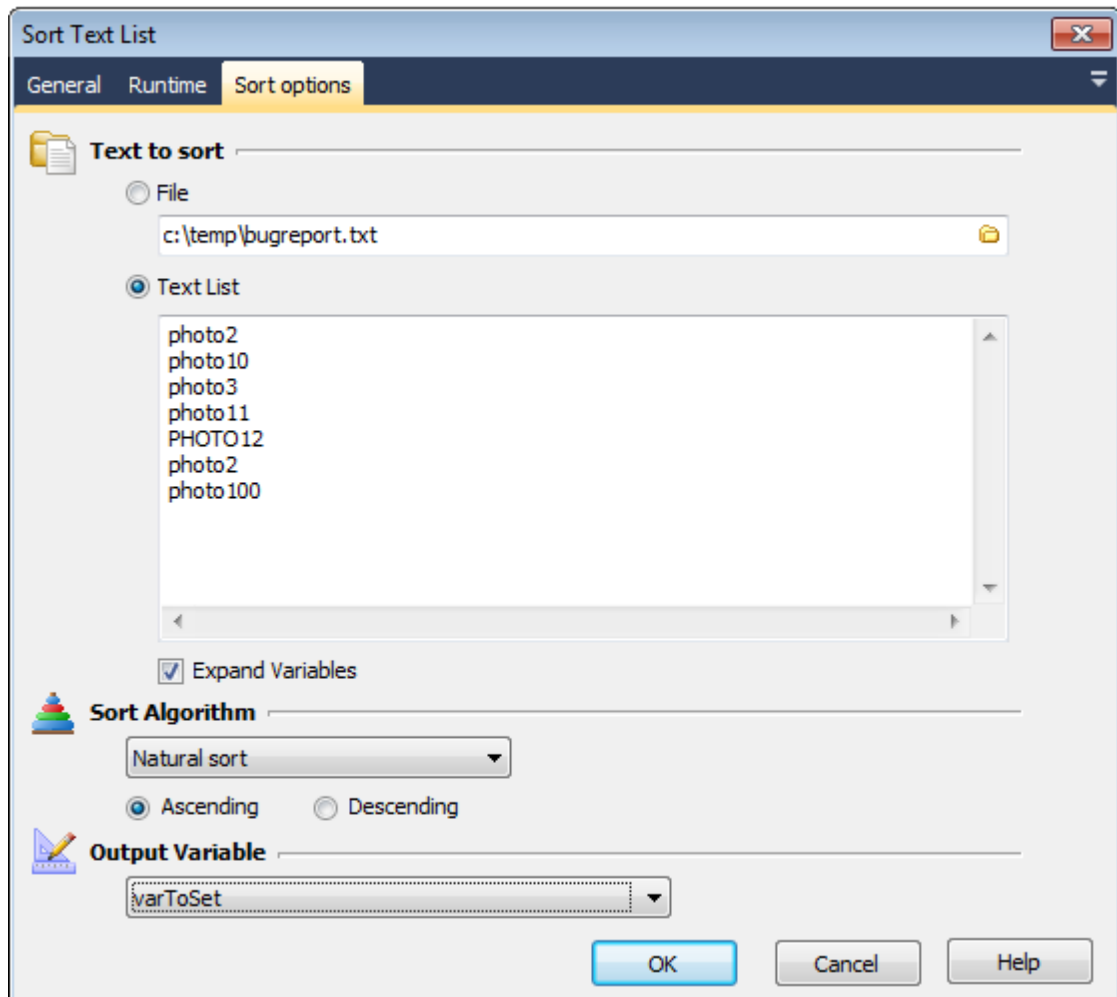
(For more information on FinalBuilder scripting and action script events, see the Scripting in FinalBuilder overview topic.)



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.

6.24.9 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.



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)

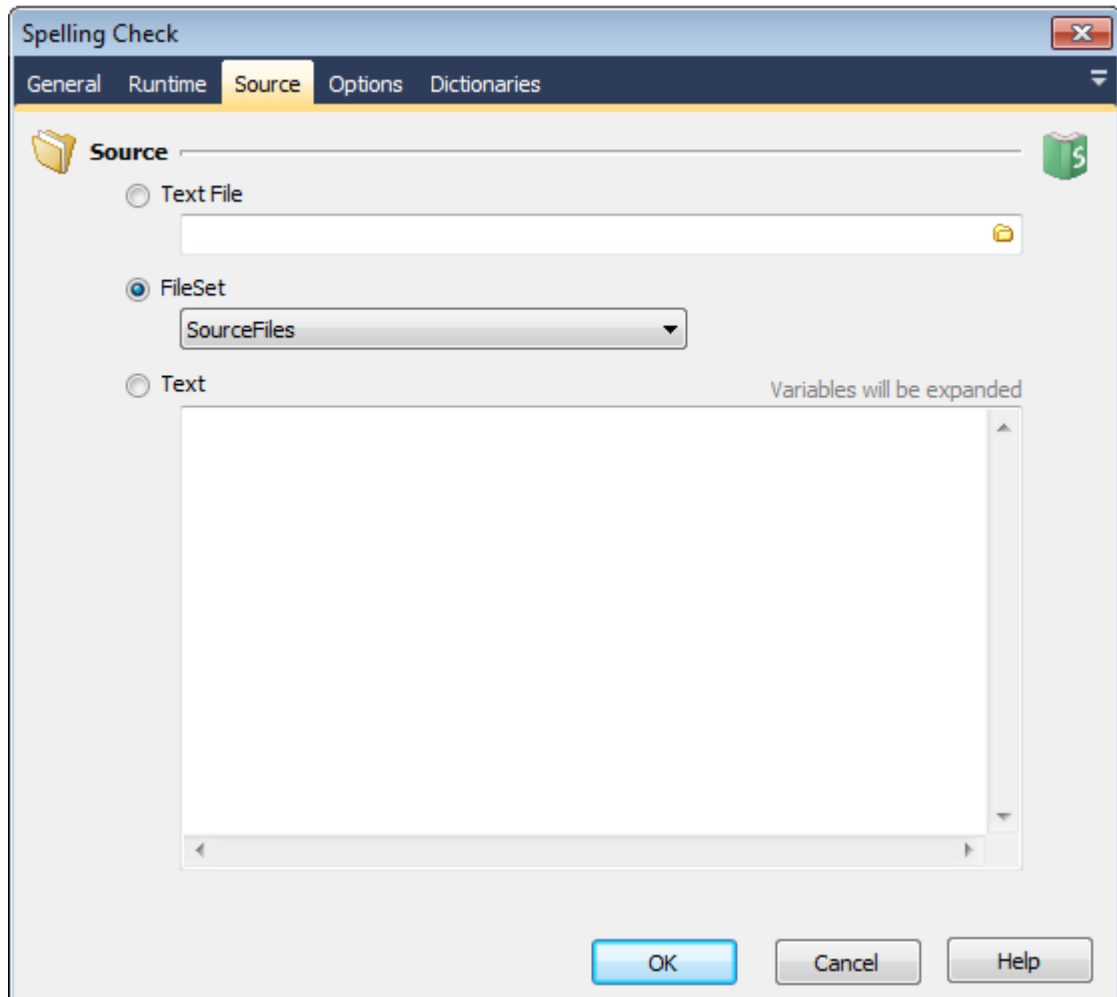
Output Variable

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.

6.24.10 Spell Checking

[FinalBuilder Professional Edition]

The Spell Check action allows you to check a text file for any spelling errors (ie. unknown words).

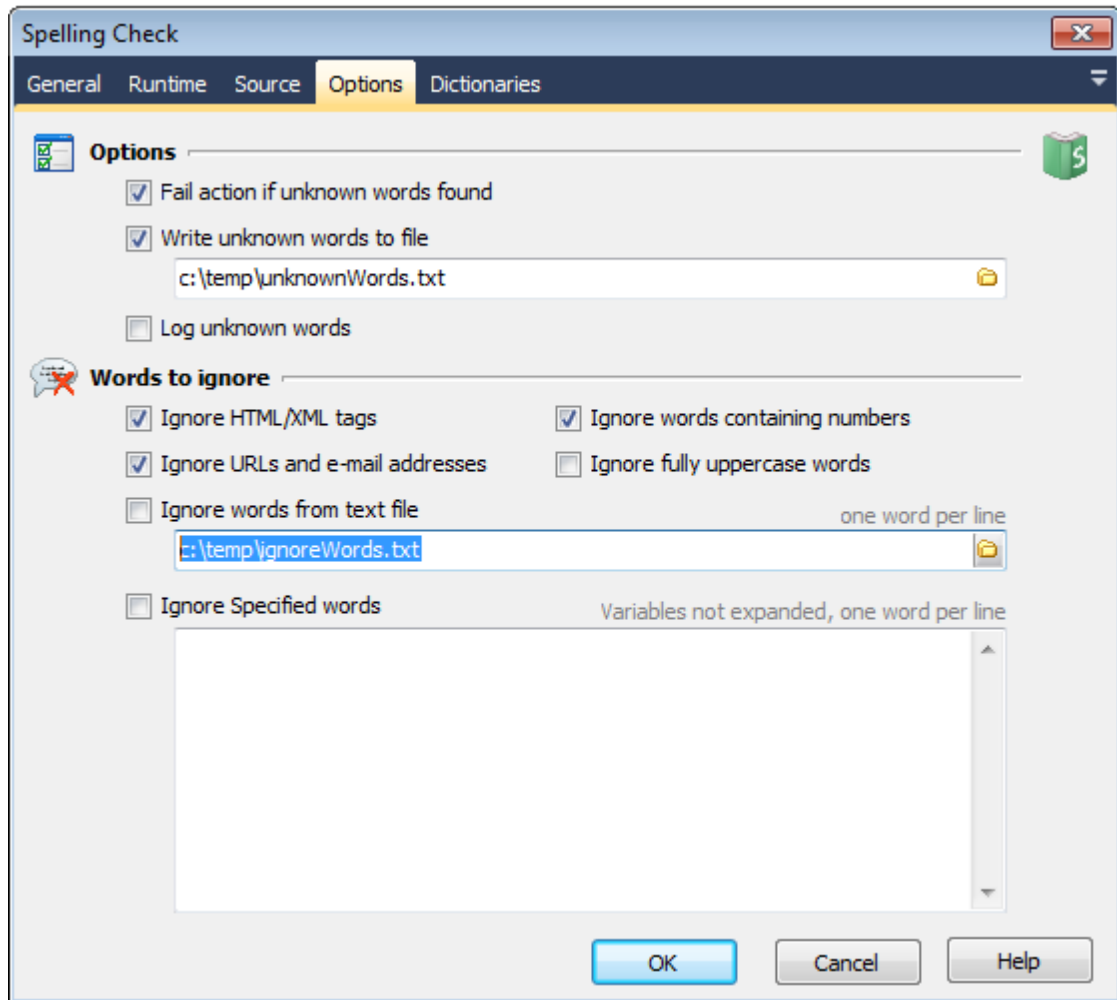


Select the source to check spelling of:

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 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 exist.

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>error</speeling>` will only test the word "error" 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"

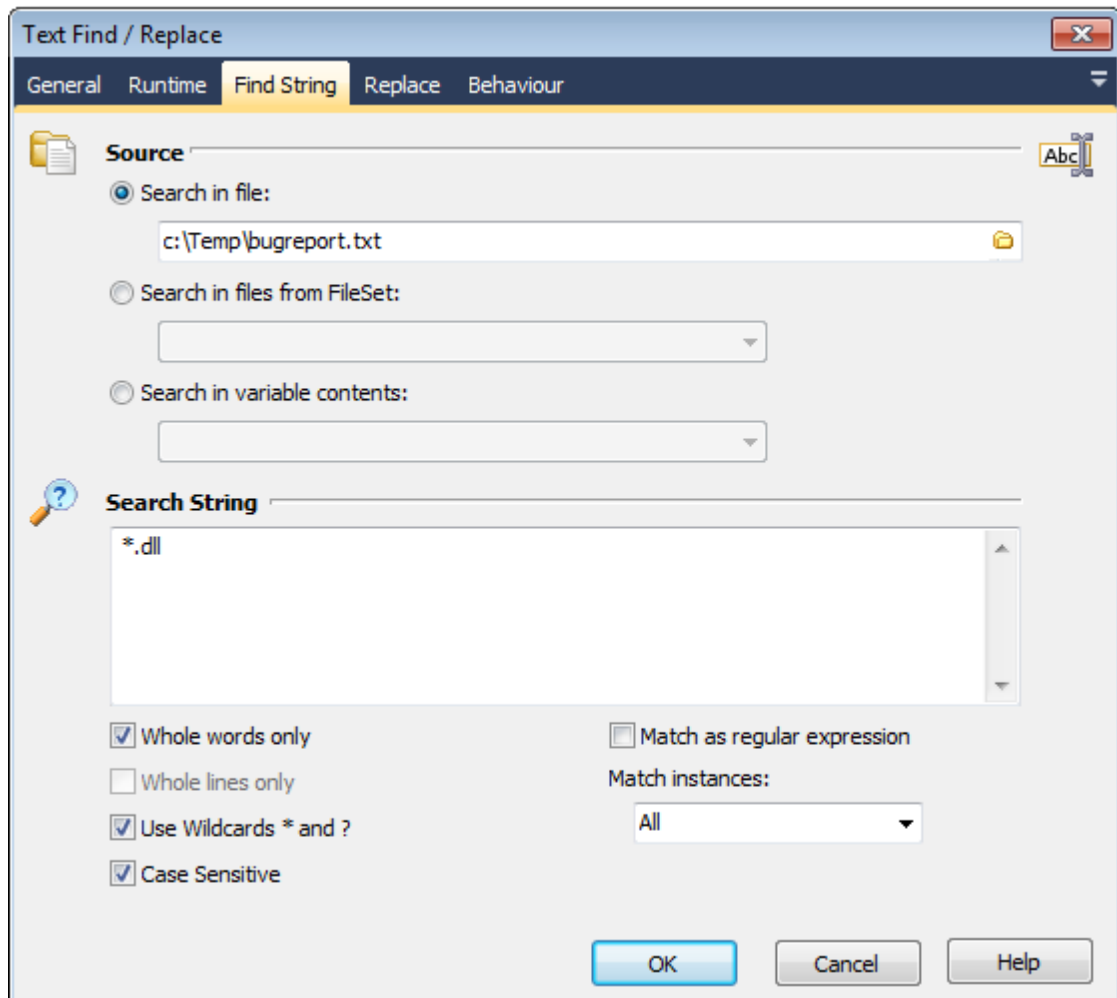
Ignore words from text file - each line of the file is interpreted as a word and will be ignored

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

6.24.11 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.



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"*

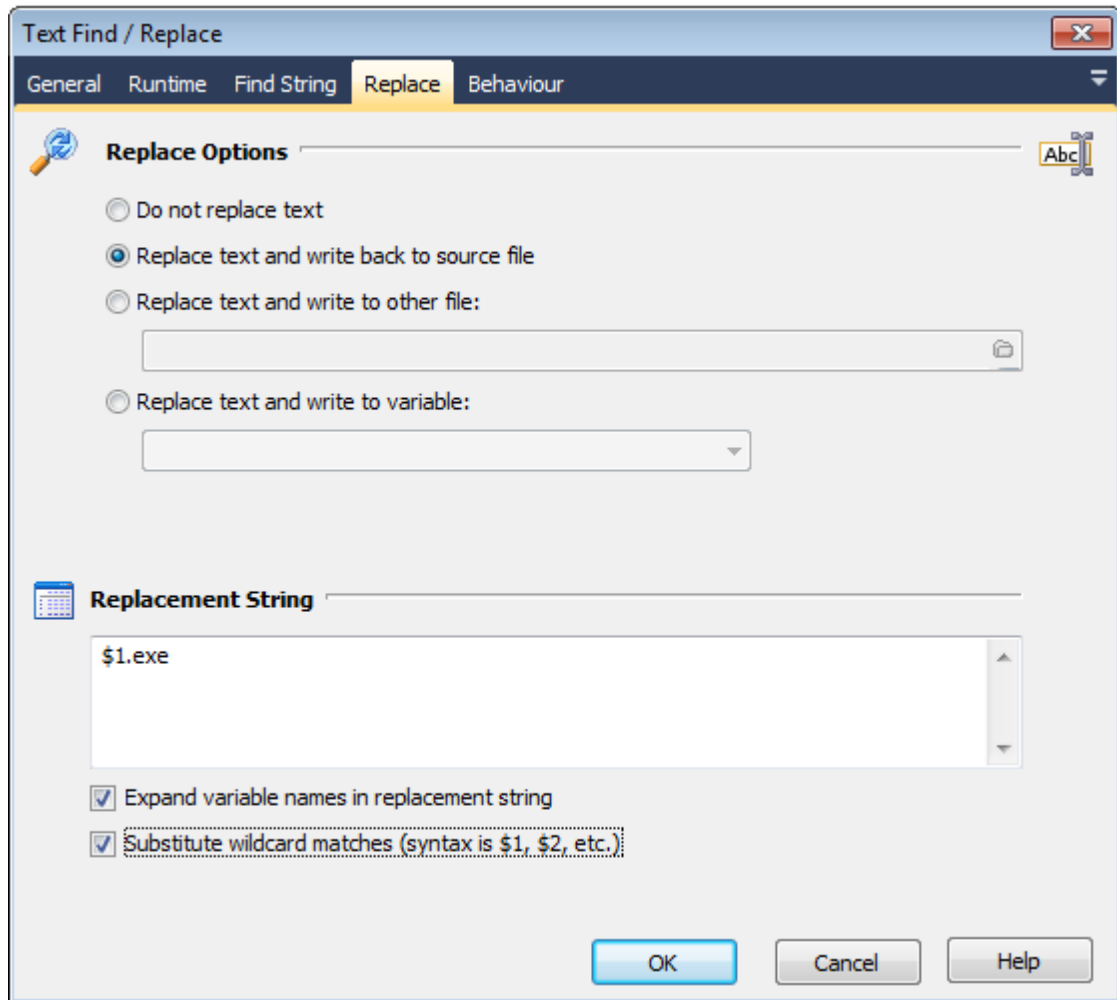
If this box is checked, the search string can contain the wildcard characters "*" and "?". * will match any number of characters (including none), whereas ? will match any single character.

"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"

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.)"

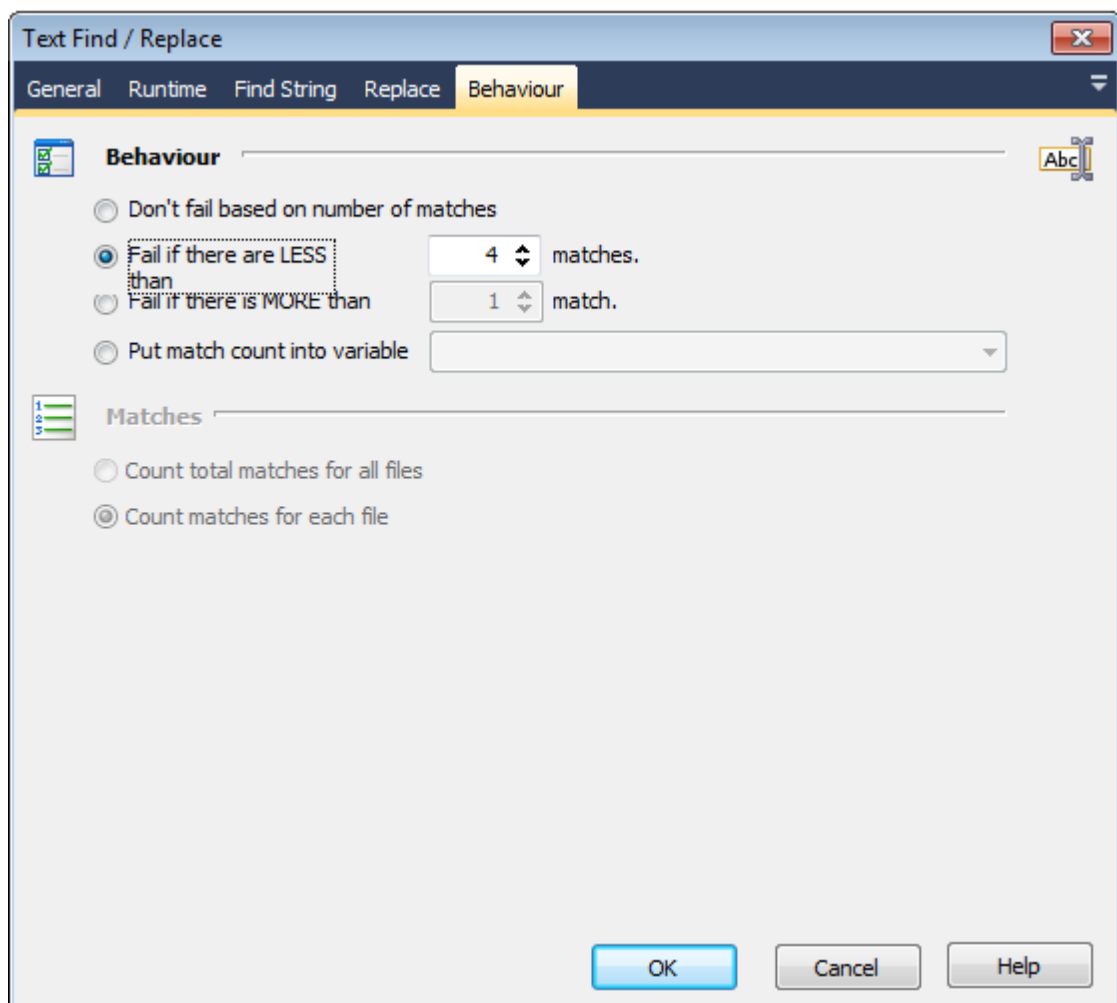
Enable this option to replace \$1, \$2, etc. with the text of the corresponding wildcard (*) match from the search string. *"Wildcard * matches any substring"* must be selected for this option to take effect. In the example above, \$1 would be replaced with the value which matched the leading * from "*.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

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.

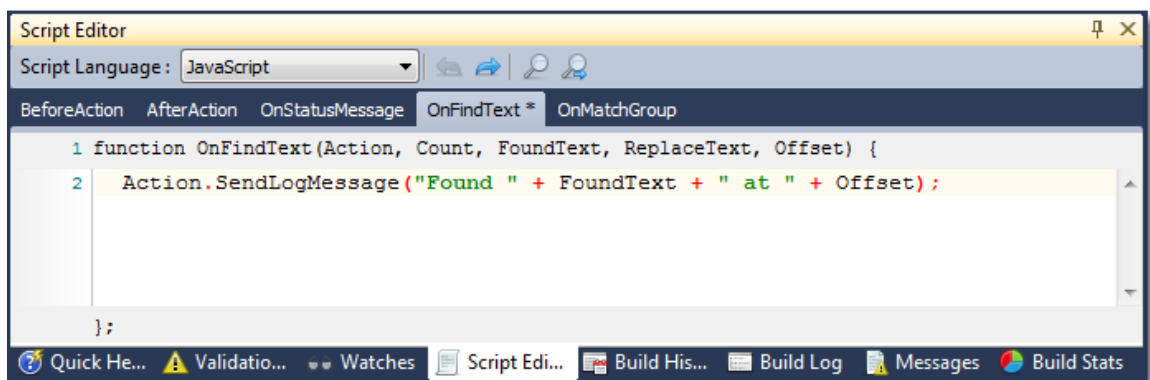
Matches

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.



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

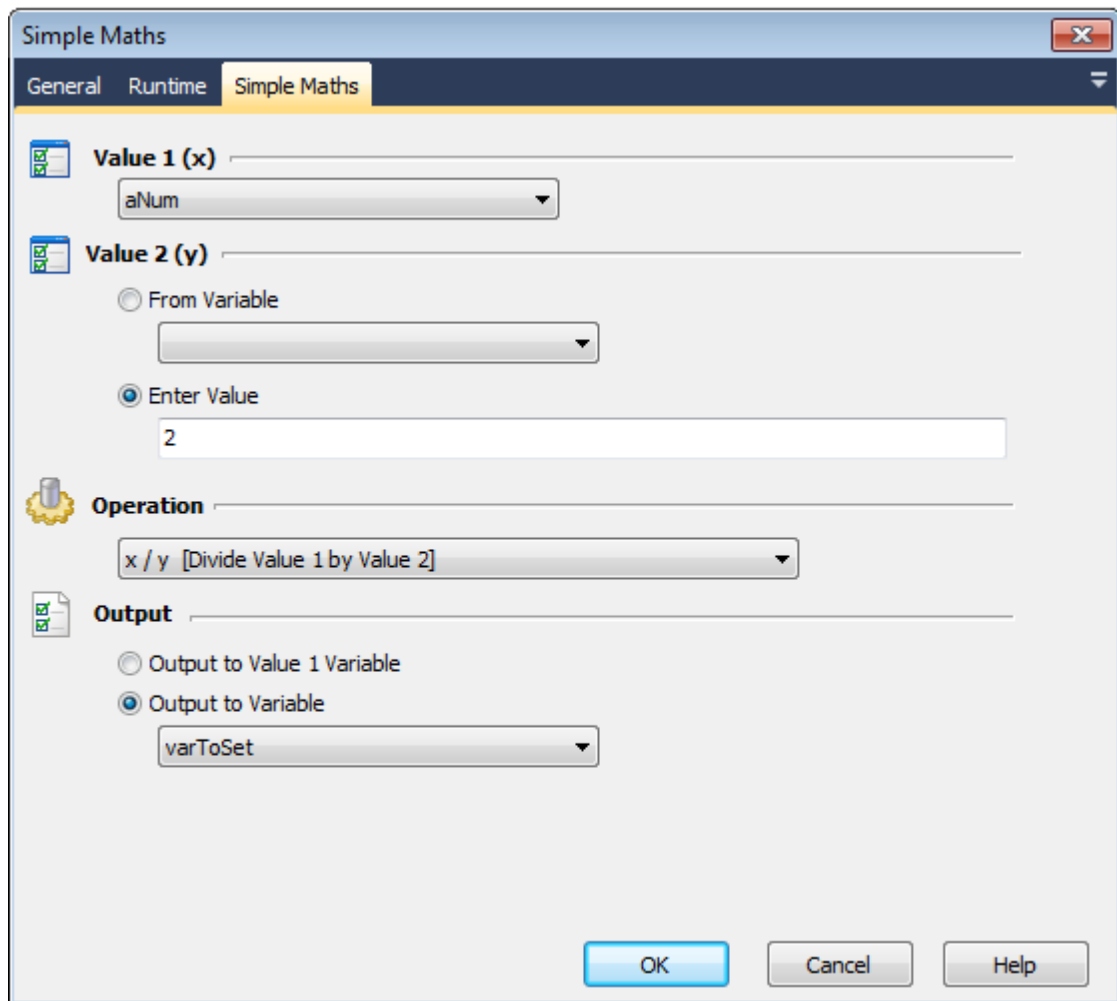
The offset in the source at which the text was found. *Offset* is zero-indexed.

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.

6.24.12 Simple Maths Action

Use the Simple Maths action to perform mathematical calculations.



The image shows a 'Simple Maths' dialog box with three tabs: 'General', 'Runtime', and 'Simple Maths'. The 'Simple Maths' tab is selected. It contains the following fields and options:

- Value 1 (x)**: A dropdown menu showing 'aNum'.
- Value 2 (y)**: Two options: 'From Variable' (unselected) with an empty dropdown, and 'Enter Value' (selected) with a text field containing '2'.
- Operation**: A dropdown menu showing 'x / y [Divide Value 1 by Value 2]'.
- Output**: Two options: 'Output to Value 1 Variable' (unselected), and 'Output to Variable' (selected) with a dropdown menu showing 'varToSet'.

At the bottom right are three buttons: 'OK', 'Cancel', and 'Help'.

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.

6.25 .NET Actions

6.25.1 .Net Framework Tools

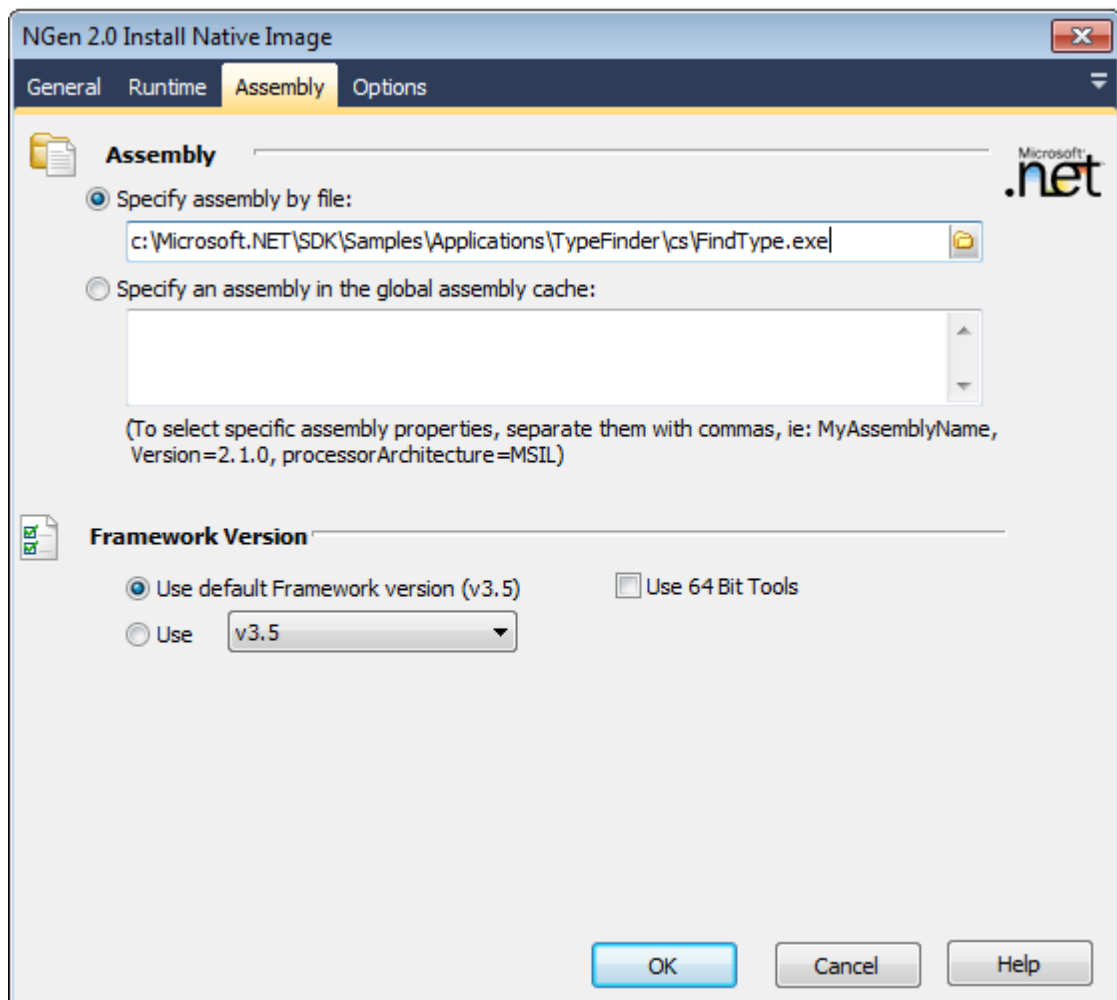
The .Net Framework tools are installed as part of the .Net framework (ie. if you have .Net installed, you have the framework tools)

6.25.1.1 NGen 2.0 Native Image Install Action

The NGen 2.0 Native Image Install Action generates native images for an assembly (and its dependencies) and installs them in the Native Images Cache. Assemblies with native images run faster on the current machine because they can bypass the .NET Just-In-Time compiler.

Note: This action requires the Microsoft .NET 2.0 Framework.

For more information see [http://msdn2.microsoft.com/en-us/library/6t9t5wcf\(en-us,VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/6t9t5wcf(en-us,VS.80).aspx)



Specify Assembly By File

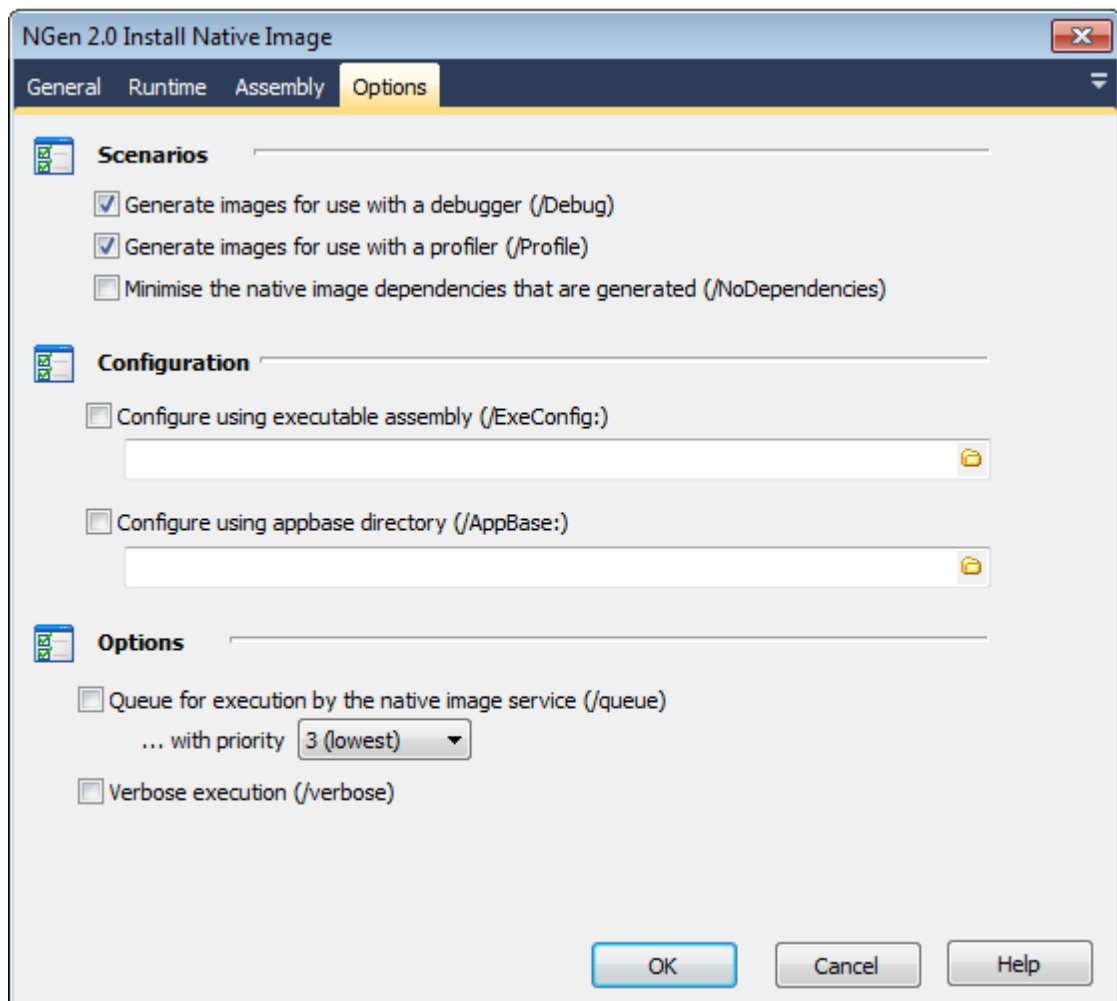
Enter the path of an assembly to precompile.

Specify an assembly in the global assembly cache

Specify the name of an assembly in the global assembly cache. Separate extra properties with commas.

Framework Version

Specify the version of the .NET framework to use for NGen. The minimum is 2.0.



Scenarios

Native images can be compiled for various scenarios. Debugging and/or profiling information can be included. Alternatively, the native image can be generated with a minimum of dependencies (not recommended because non-native dependencies will cause the Just In Time compiler to be invoked, negating the benefits of Native Images.)

Configuration

The assembly configuration can be optionally taken from an executable configuration or an application base path (note that these two options are mutually exclusive.)

Queue for execution...

Rather than installing directly, the generation & installation can be queued for execution by the Native Image Service.

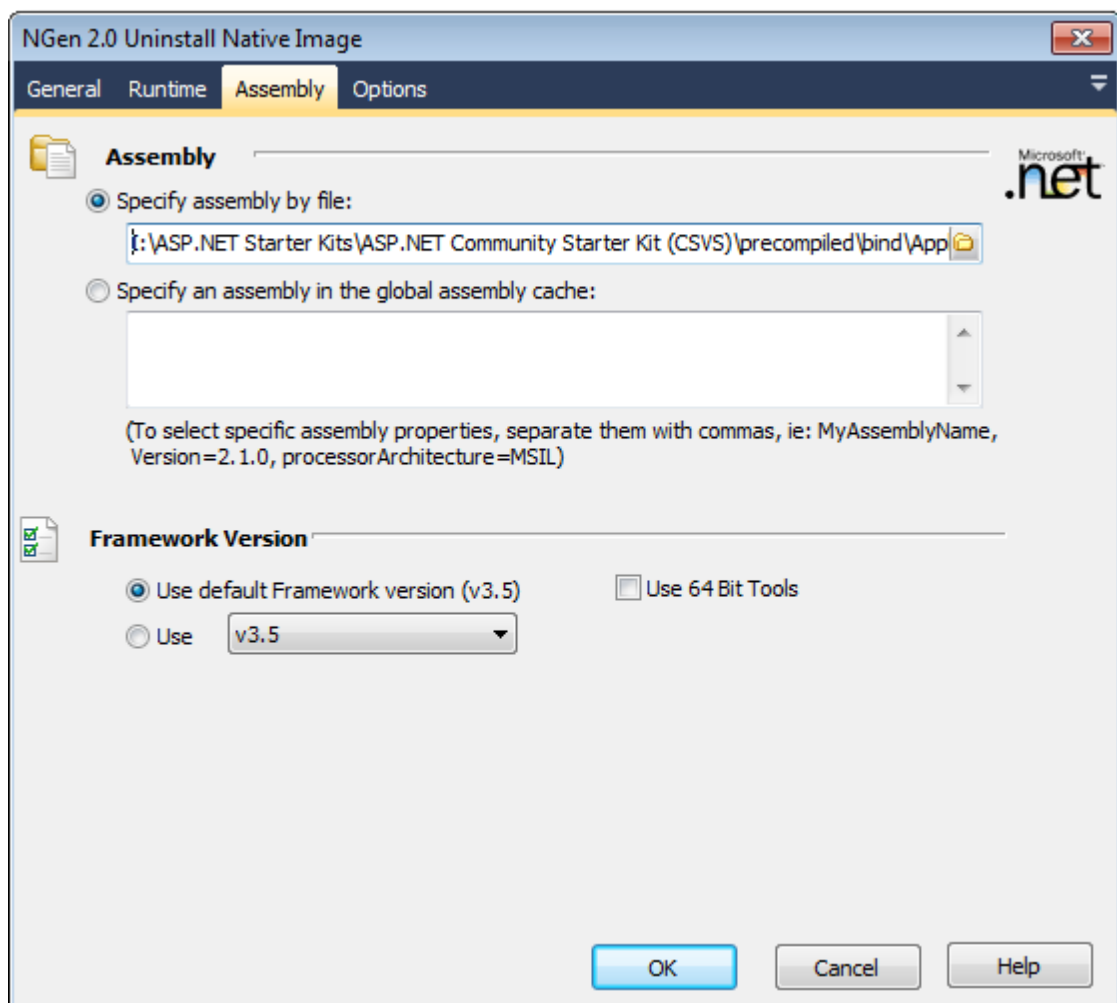
6.25.1.2 NGen 2.0 Native Image Uninstall Action

The NGen 2.0 Native Image Uninstall Action removes native images of an assembly (and its dependencies) from the Native Images Cache.

Note that NGen maintains a reference count, so native assemblies will only be removed if they are not depended upon by other native assemblies.

Note: This action requires the Microsoft .NET 2.0 Framework.

For more information see [http://msdn2.microsoft.com/en-us/library/6t9t5wcf\(en-us,VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/6t9t5wcf(en-us,VS.80).aspx)



Specify Assembly By File

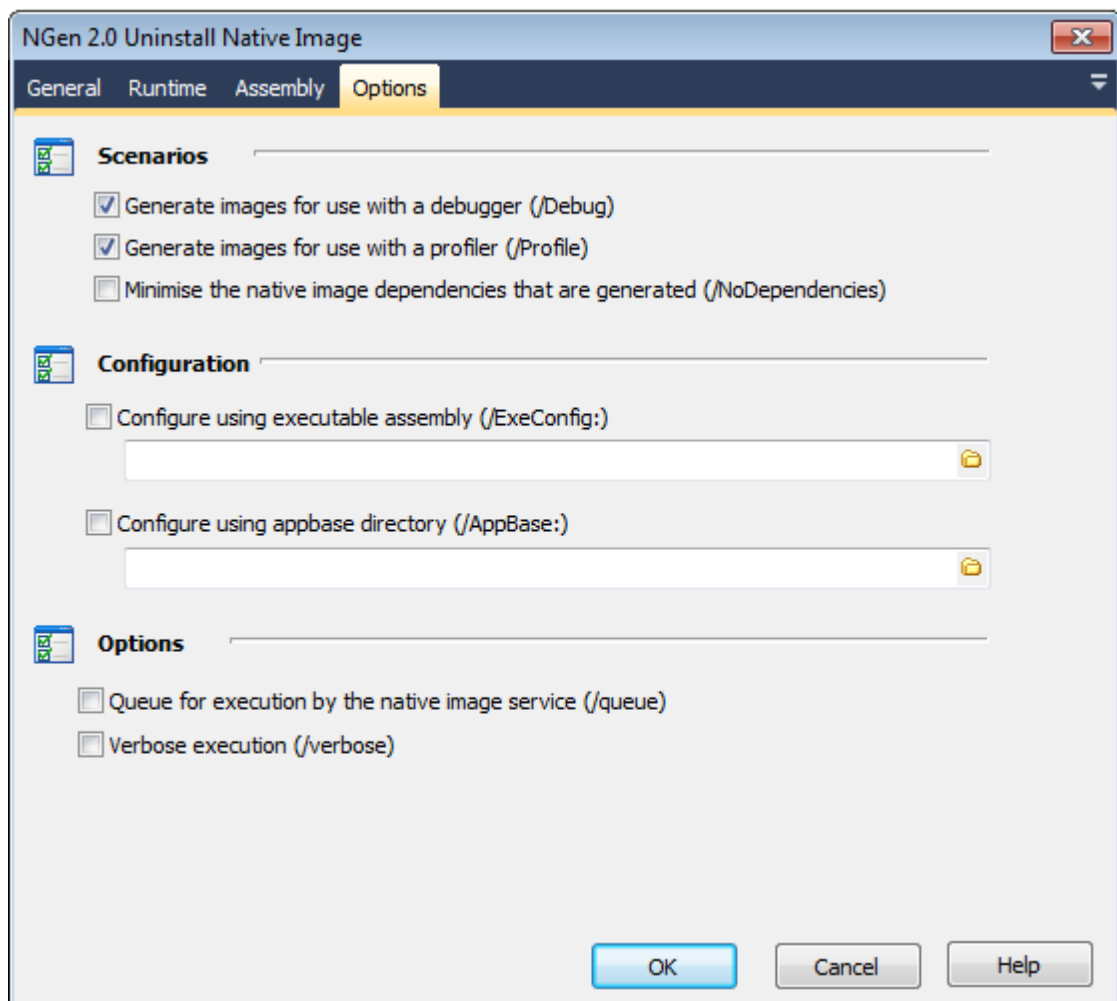
Enter the path of an assembly to remove.

Specify an assembly in the global assembly cache

Specify the name of an assembly in the global assembly cache. Separate extra properties with commas.

Framework Version

Specify the version of the .NET framework to use for NGen. The minimum is 2.0.



Scenarios & Configuration

Use the same options that were used to install the assembly.

Queue for execution...

Rather than being directly removed, the removal job will be queued for execution by

the Native Image Service.

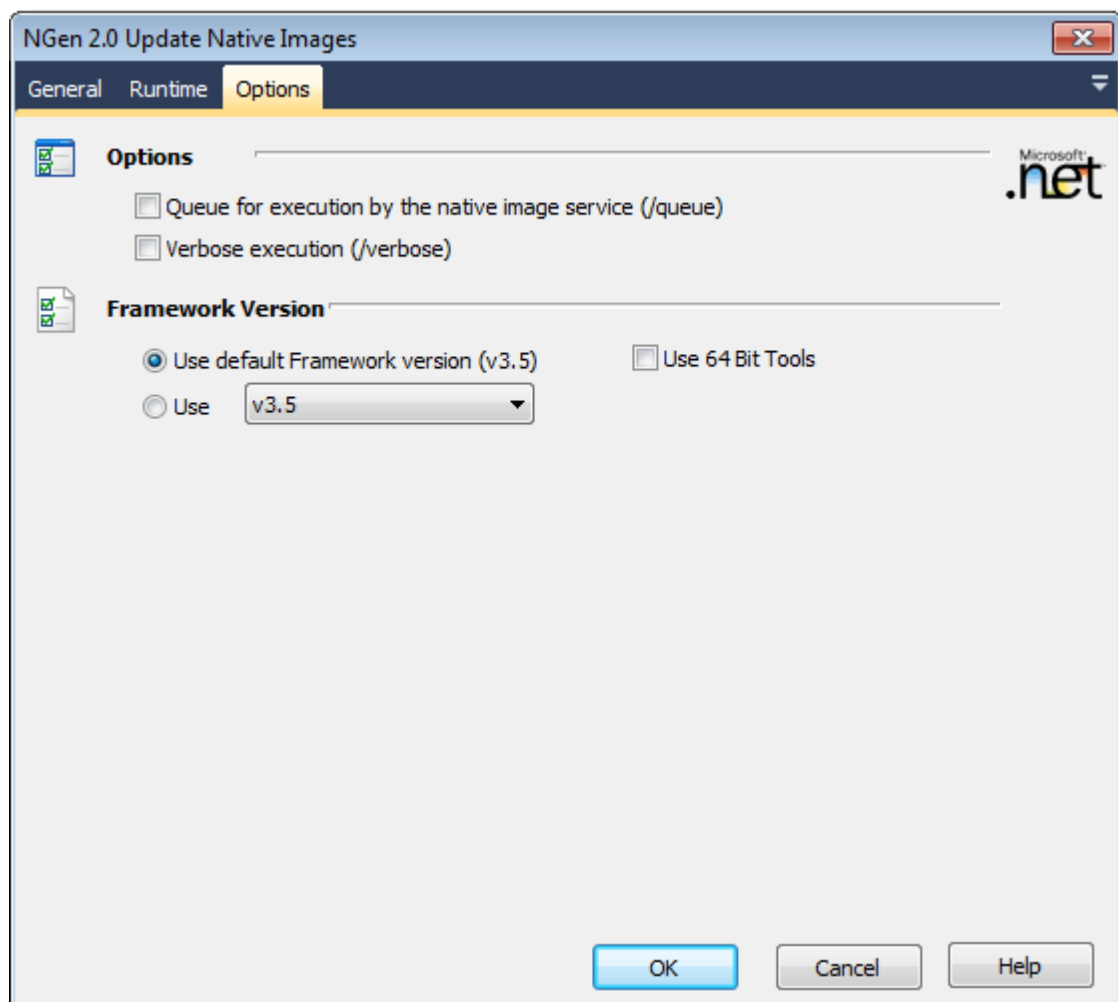
6.25.1.3 NGen 2.0 Native Images Update Action

The NGen 2.0 Native Images Update Action updates any invalid images from the Native Images Cache.

Updating invalid images is required because if an assembly image is out of date, the Just In Time compiler will be invoked when an assembly references it (which negates the benefits of Native Images.)

Note: This action requires the Microsoft .NET 2.0 Framework.

For more information see [http://msdn2.microsoft.com/en-us/library/6t9t5wcf\(en-us,VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/6t9t5wcf(en-us,VS.80).aspx)



Queue for execution...

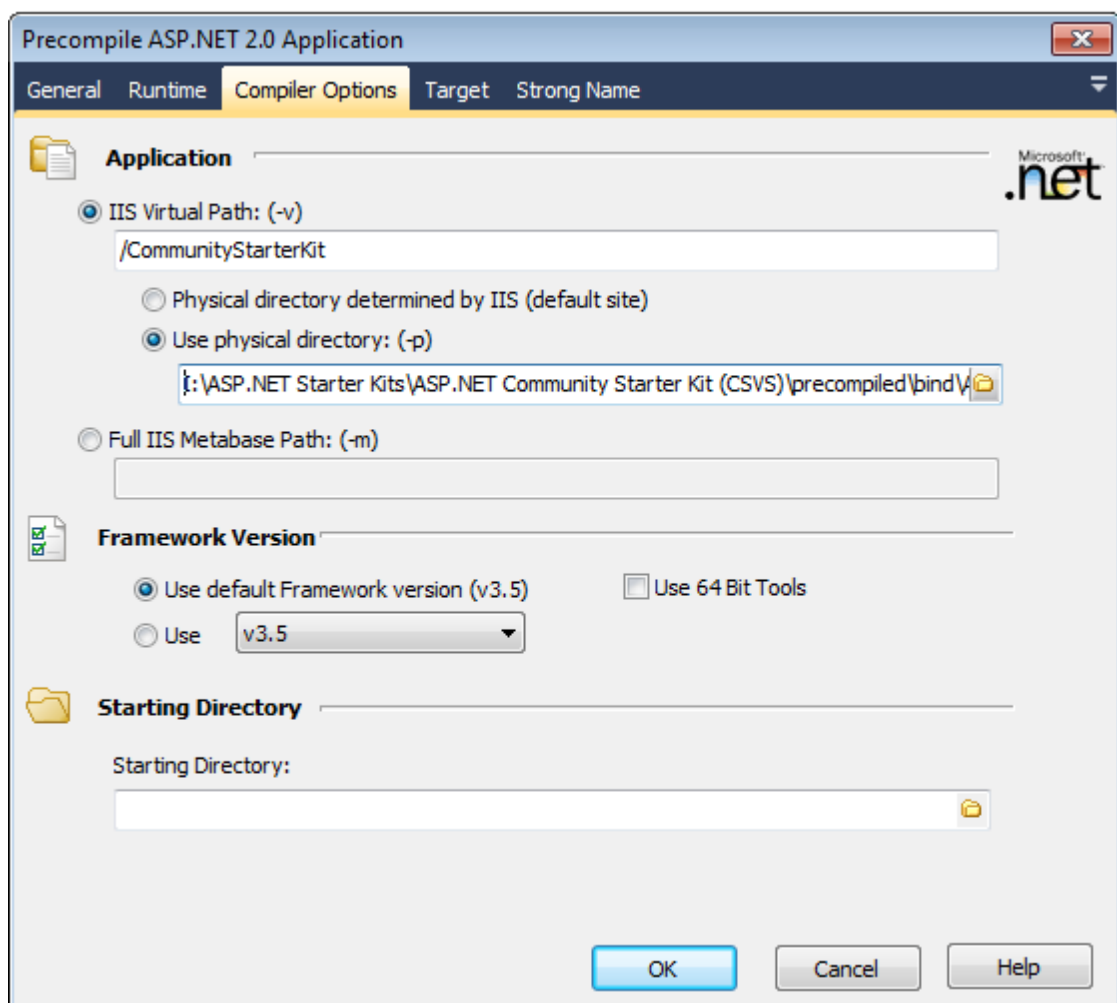
Rather than being updated immediately, the updates will be queued for execution by the Native Image Service.

6.25.1.4 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 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'

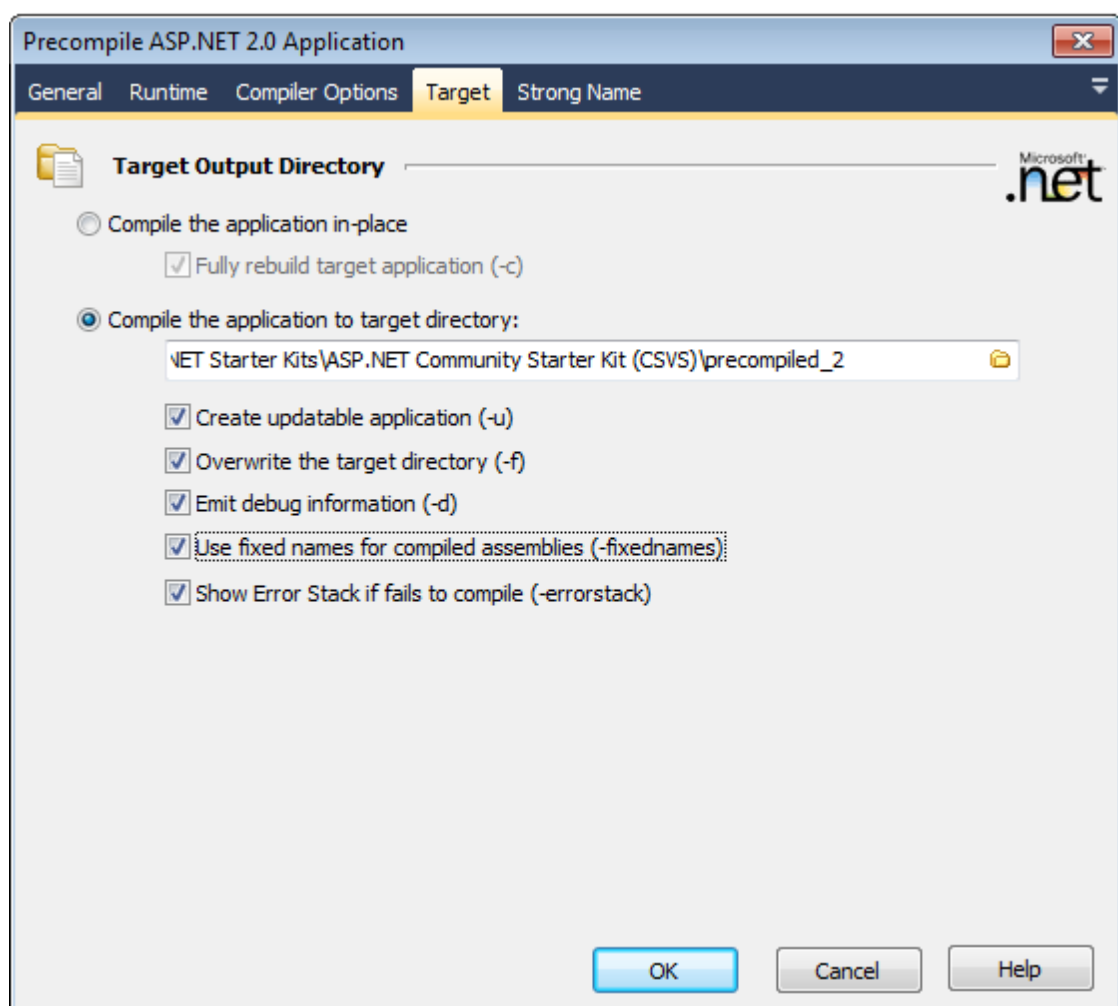
The application will be compiled from the specified physical (local) directory, but as if it was installed at the IIS virtual directory given by the path. This is useful if you want to compile your ASP.NET application from one (offline) directory, but deploy the assemblies in another directory.

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.



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."

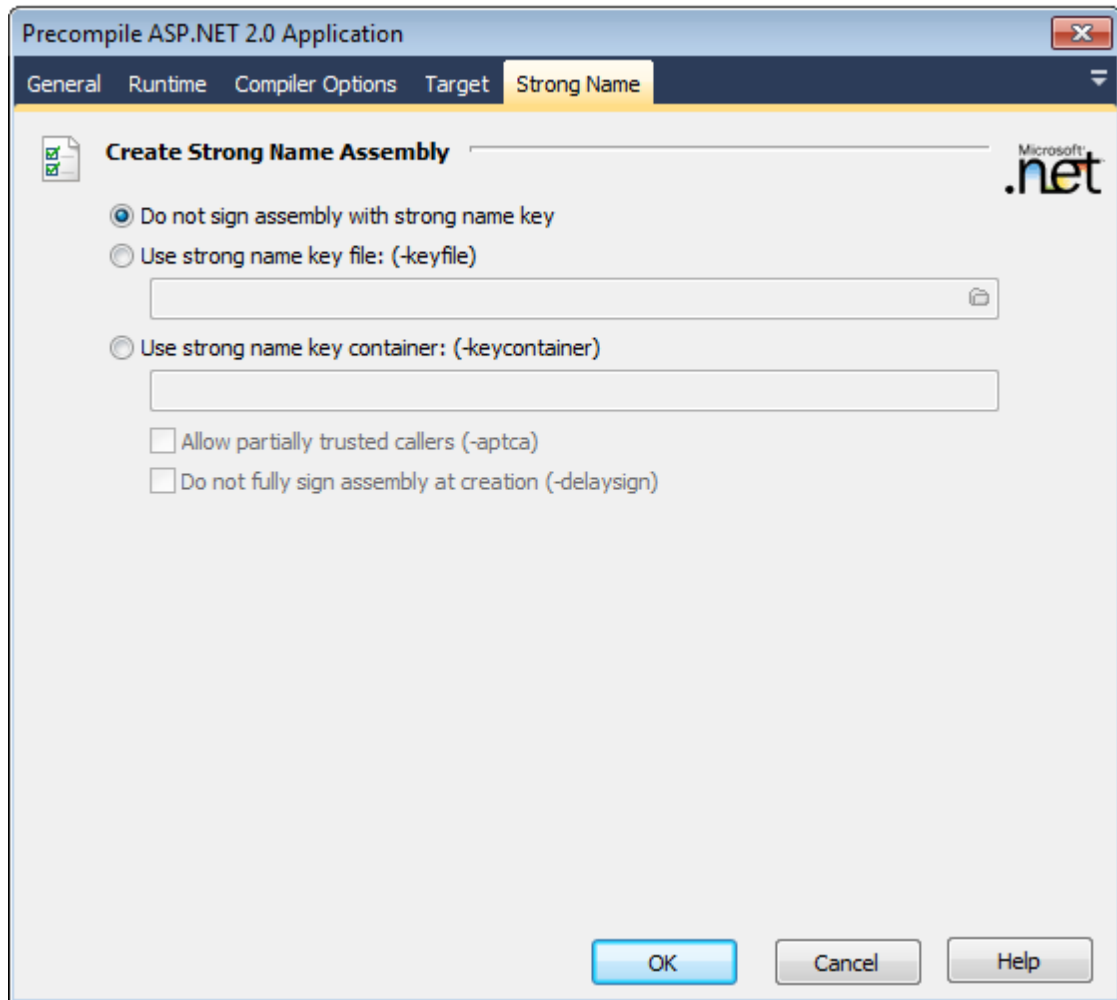
Create updatable application : The compiled application will be updatable.

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.

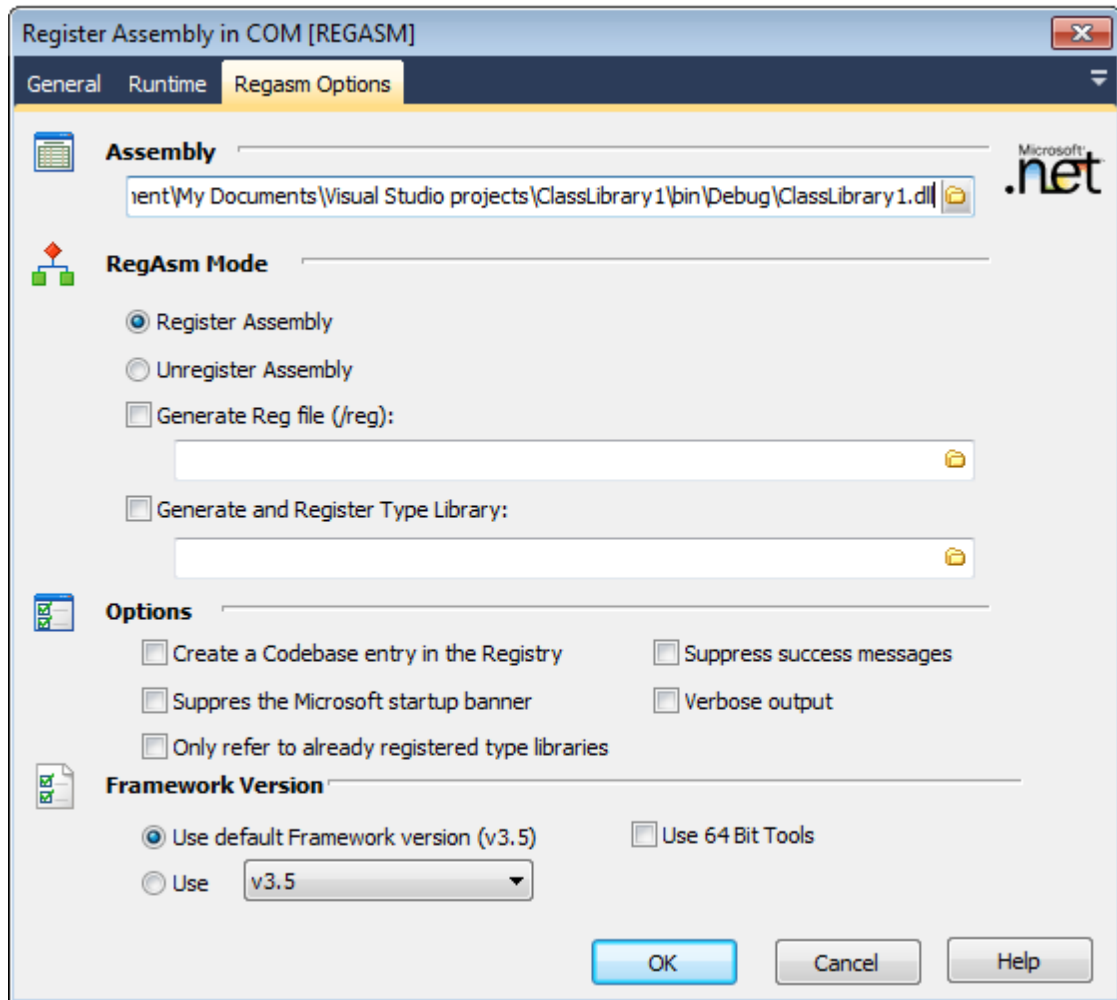
(If this option is not set, the names will be autogenerated.)



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.

6.25.1.5 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.

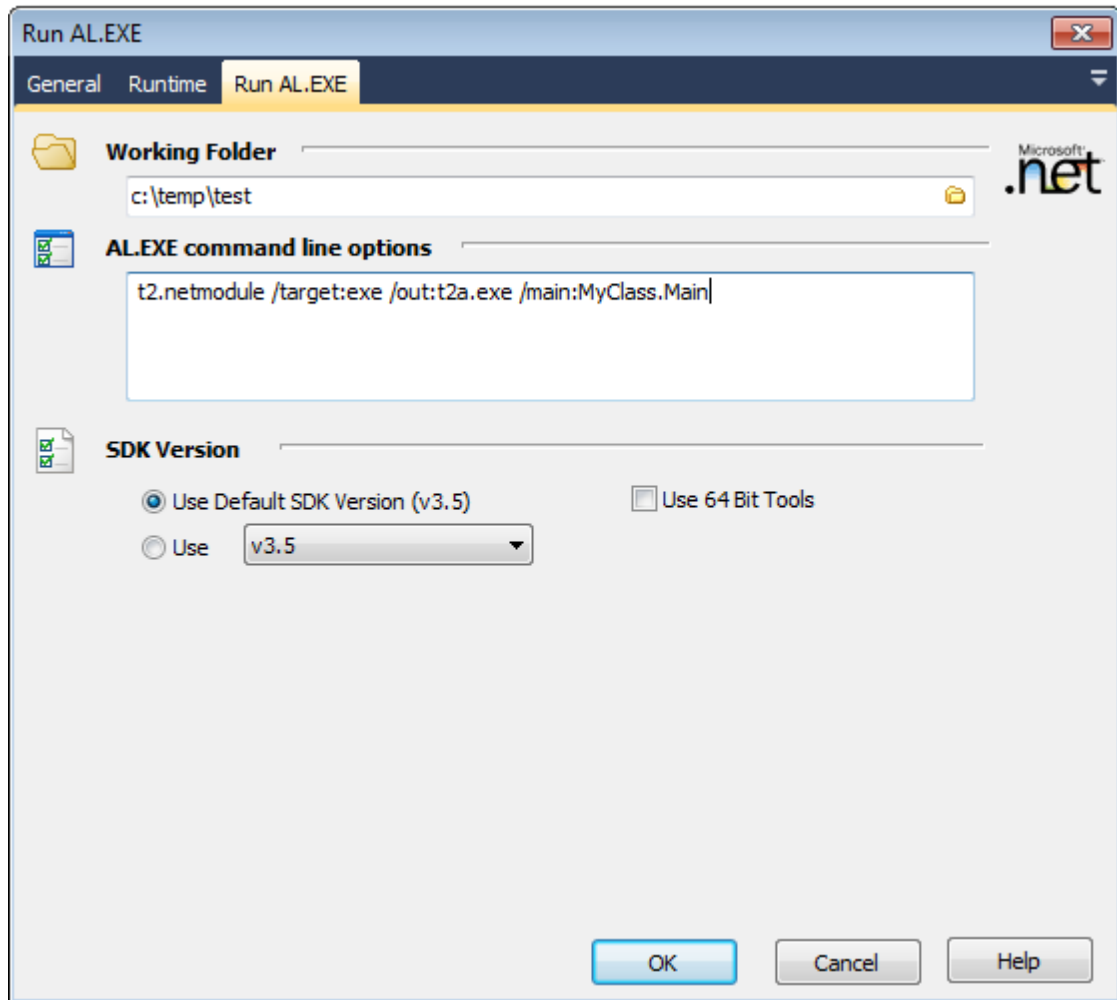


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfAssemblyRegistrationToolRegasmexe.asp>

6.25.1.6 Run AL.EXE

The Assembly Linker generates a file with an assembly manifest from one or more files that are either modules or resource files. A module is a Microsoft intermediate language (MSIL) file that does not have an assembly manifest.

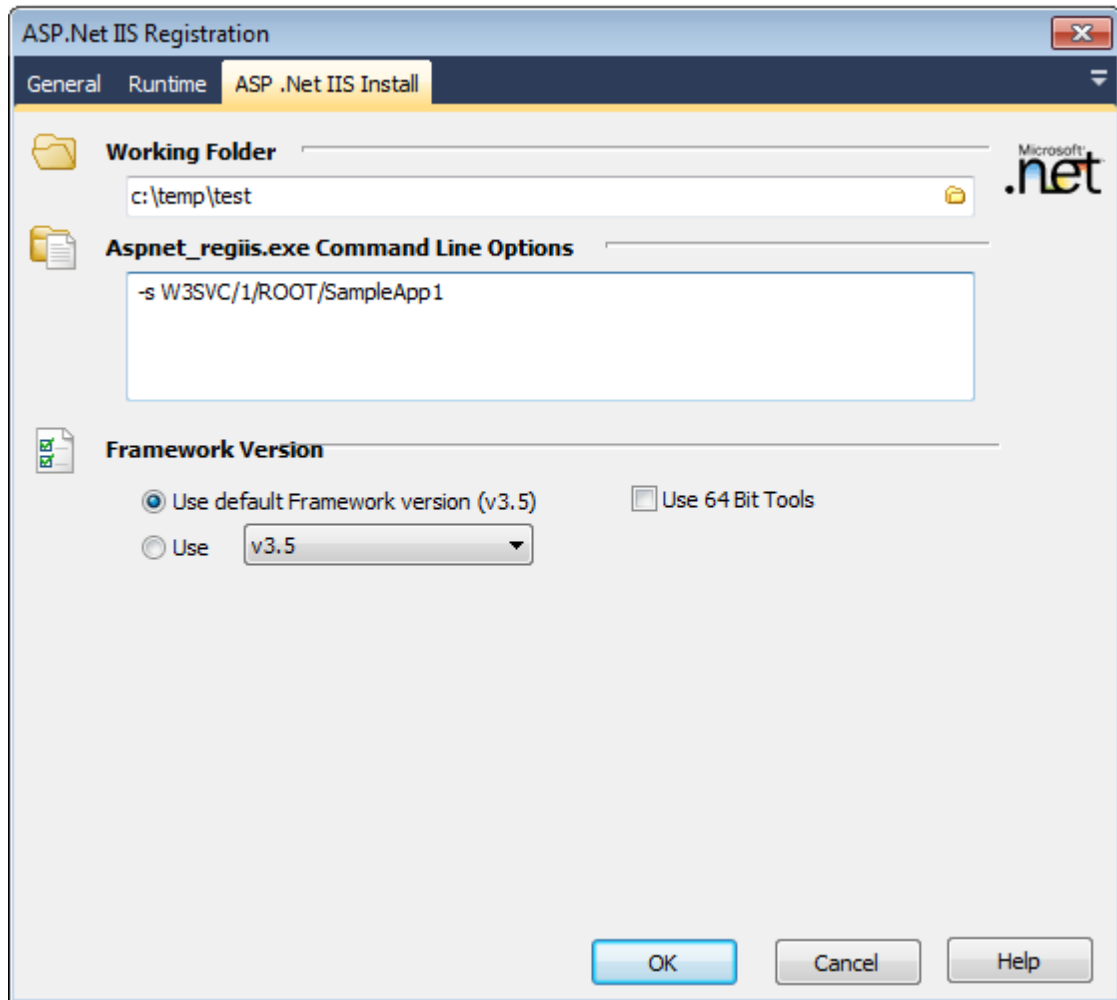


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfAssemblyGenerationUtilityAlexe.asp>

6.25.1.7 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.

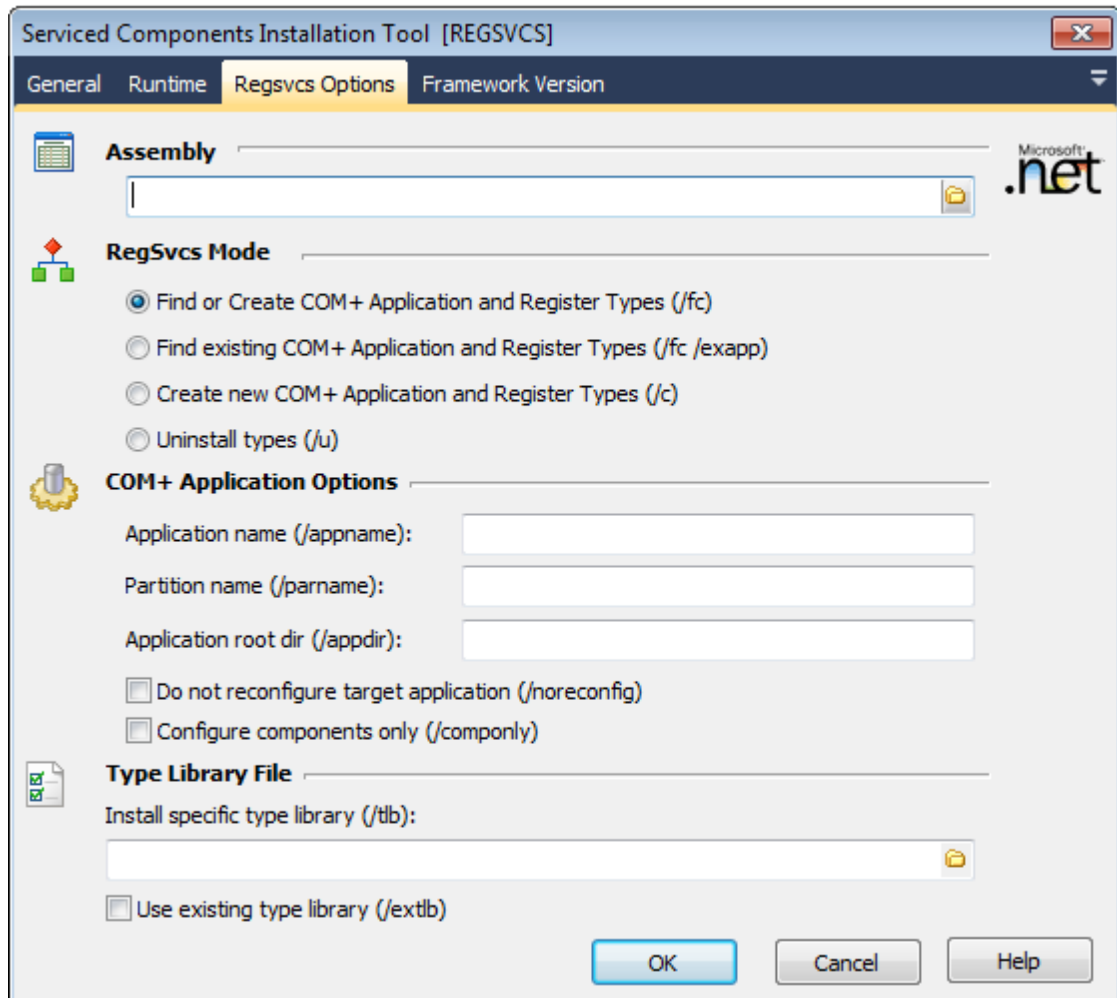


For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfASPNETIISRegistrationToolAspnet_regiisexe.asp

6.25.1.8 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.



Assembly

The name of the assembly to install/uninstall from.

RegSvc 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.

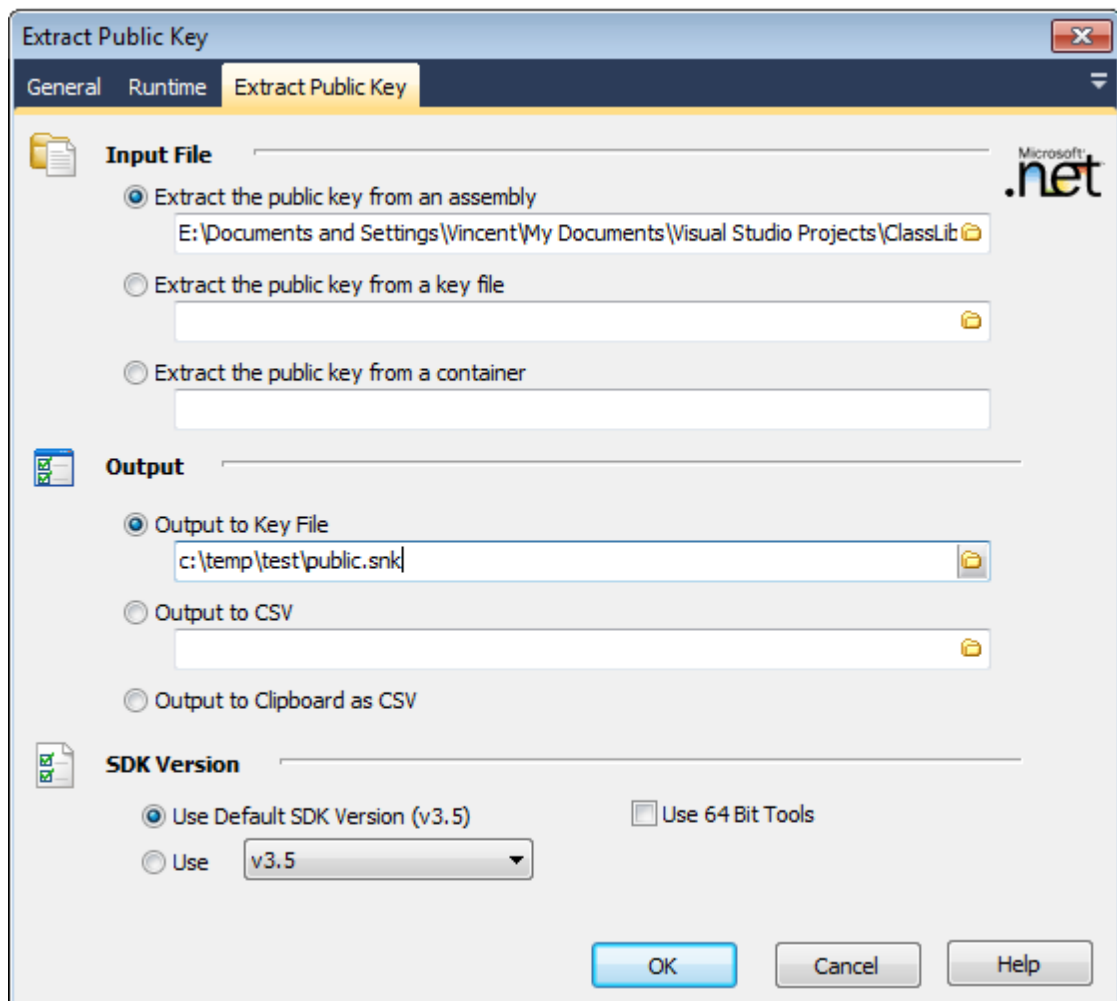
If **"Use existing type library"** is checked, the type library will not be regenerated.

6.25.2 .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

6.25.2.1 Extract Public Key [SN]

Extracts the public key from assembly and stores it in outfile.

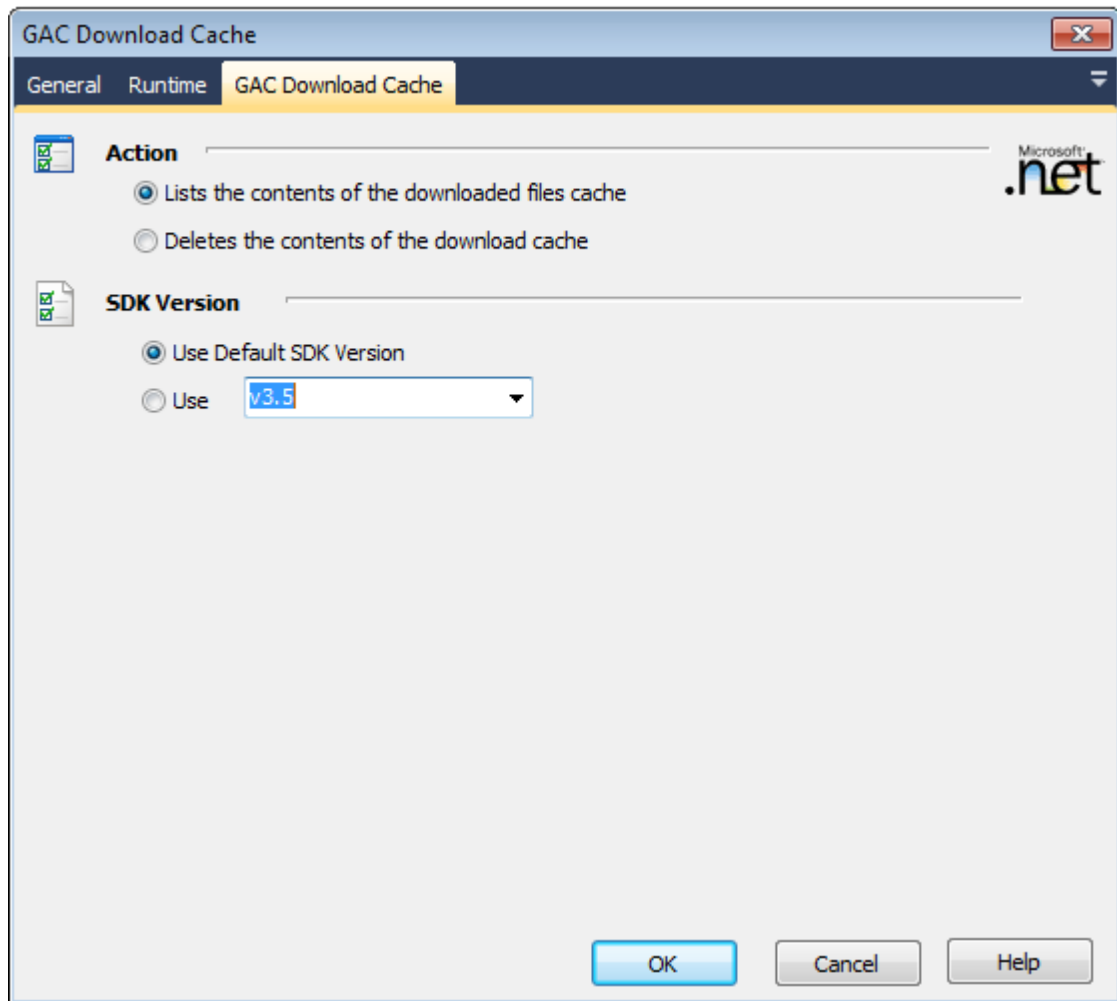


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp>

6.25.2.2 GAC Download Cache [GACUTIL]

Lists or deletes the contents of the downloaded files cache.



For more information see:

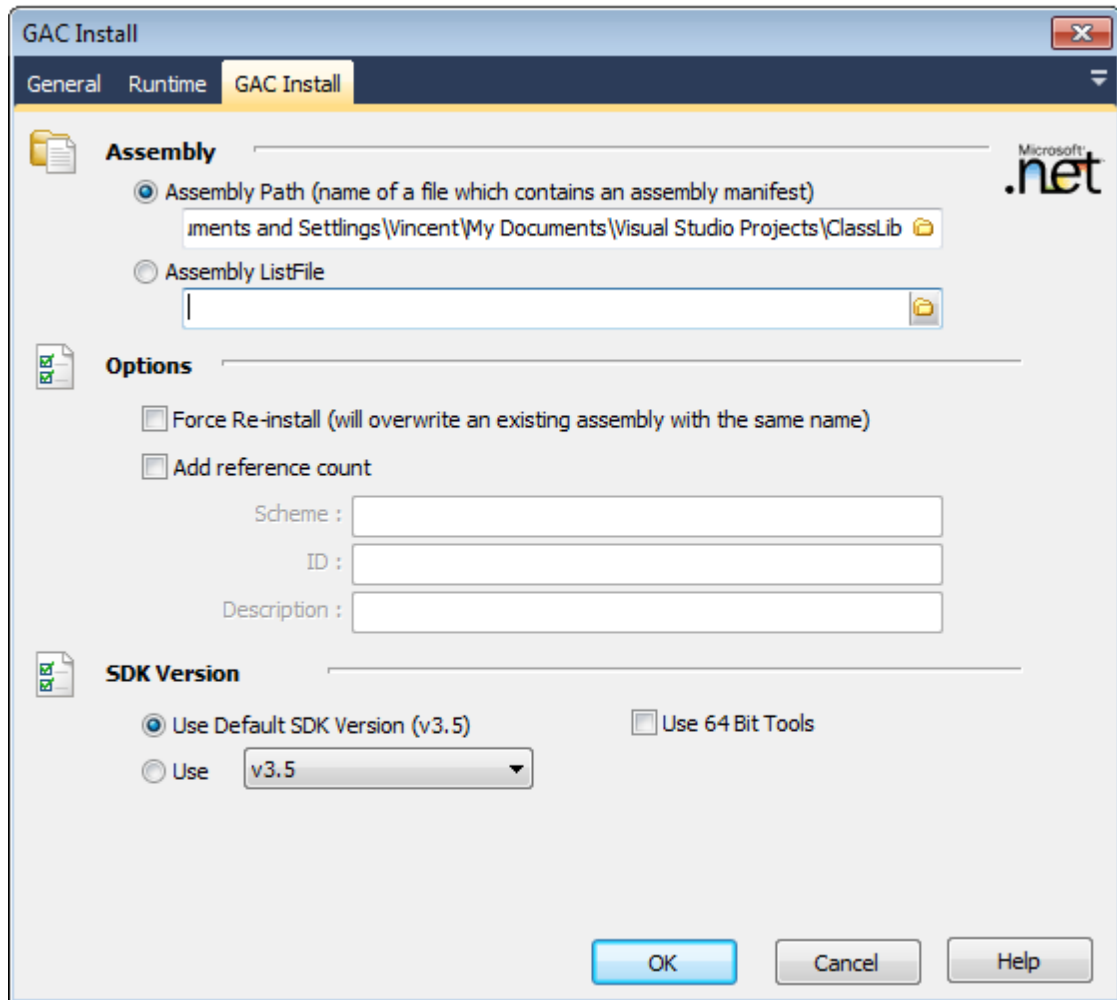
<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutil.exe.asp>

6.25.2.3 GAC Install [GACUTIL]

Installs an assembly into the global assembly cache.

This action automates the following options:

/i
/if
/il
/ir



For more information see:

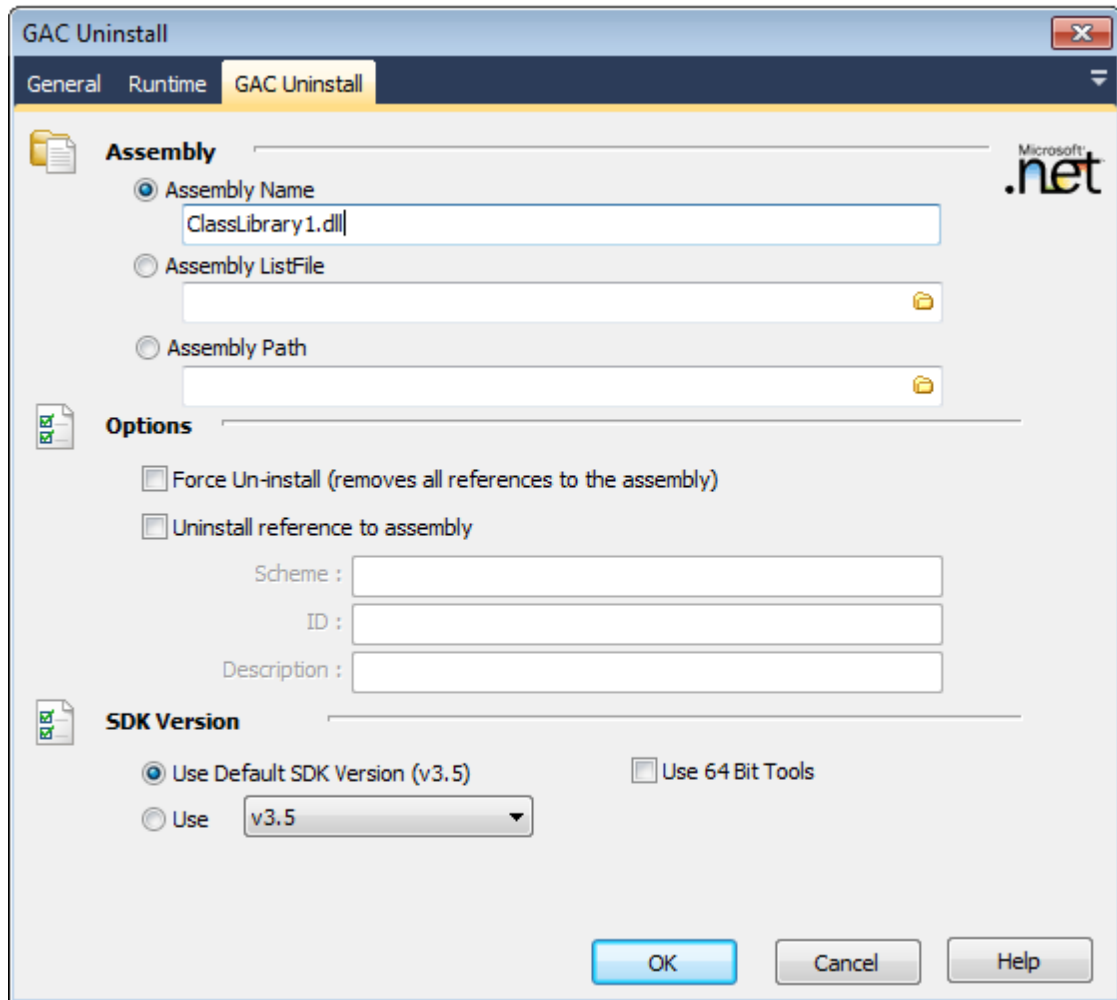
<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutil.exe.asp>

6.25.2.4 GAC Uninstall [GACUTIL]

Uninstalls an assembly from the global assembly cache.

This action automates the following options:

/u
/uf
/ul
/ur

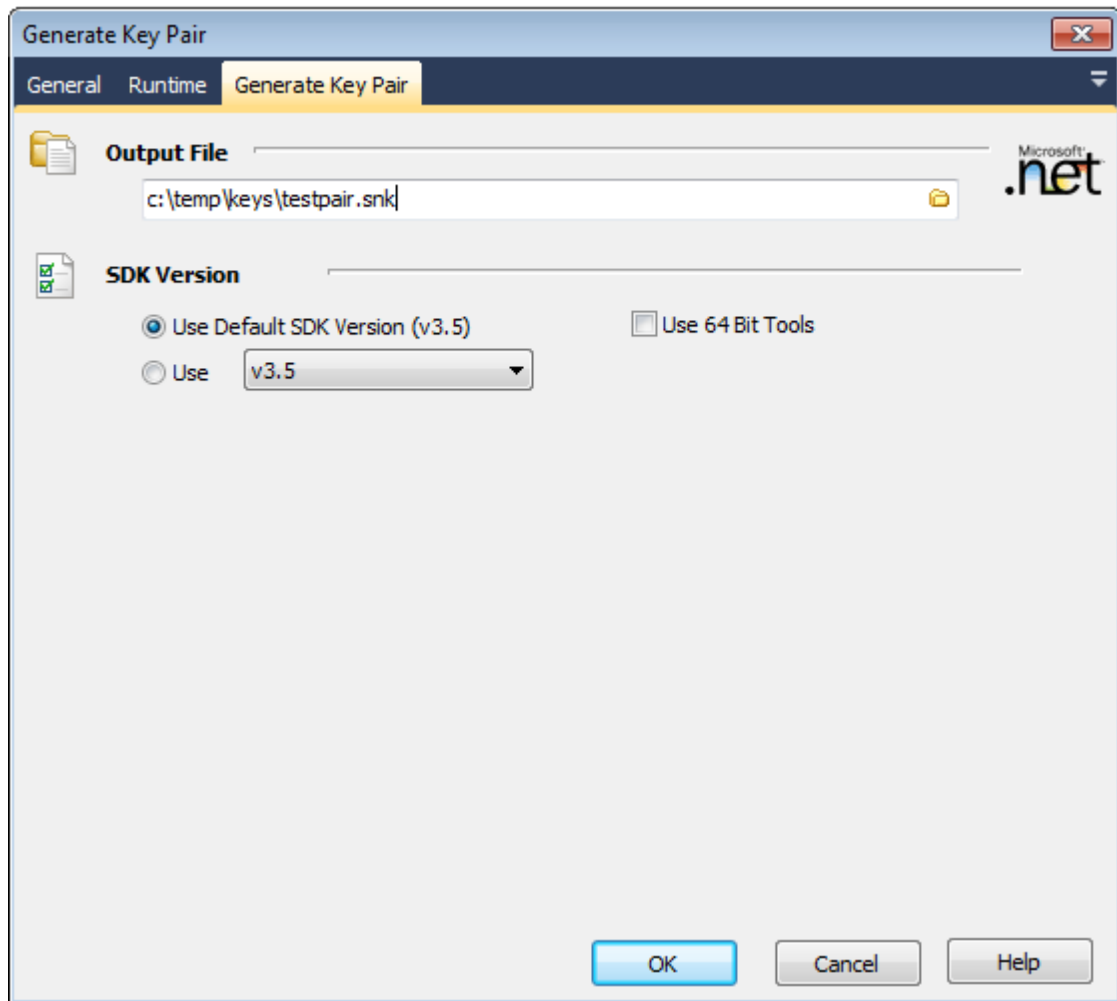


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutil.exe.asp>

6.25.2.5 Generate Key Pair [SN]

Generates a new key pair and writes it to the specified file.

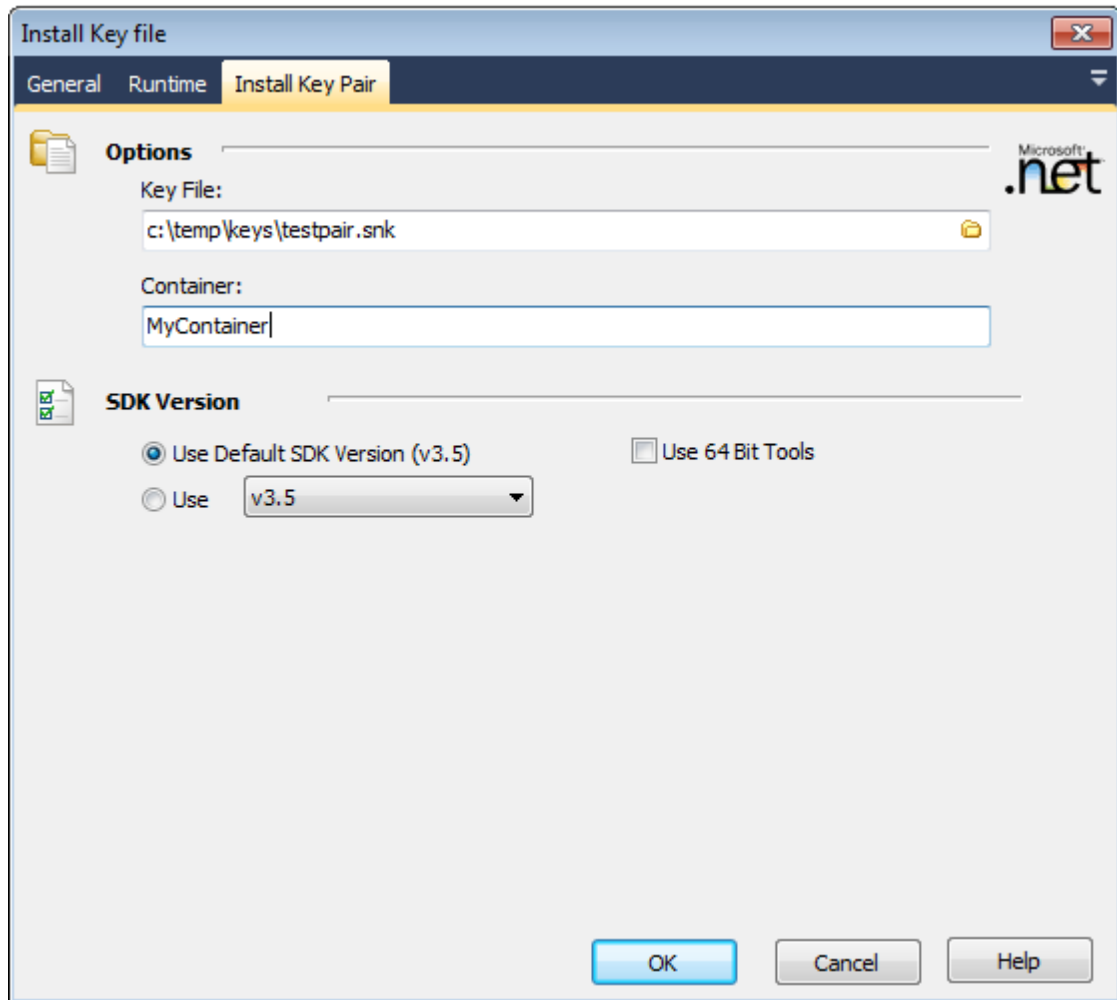


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp>

6.25.2.6 Install Key in Container [SN]

Installs the key pair from Key File in the specified key container. The key container resides in the strong name CSP.



For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp>

6.25.2.7 Manifest Generation and Editing [MAGE]

The Manifest Generation and Editing action enables the creation and editing of application and deployment manifests.

For more information see <http://msdn2.microsoft.com/en-us/library/acz3y3te.aspx>

The screenshot shows the 'Manifest Generation and Editing Tool [MAGE]' window. The 'Manifest Options' tab is selected. Under 'General Options', there are input fields for 'Project Path', 'Project Name', and 'Project Version'. Below these are two dropdown menus: 'Target Processor' (set to 'msil') and 'Trust Level' (set to 'Intranet'). Under 'Manifest Signing', there are input fields for 'Certificate Path' and 'Certificate Password'. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

Project Path

The full path of the project files.

Project Name

The name property will be used to identify the application within the Start Menu, and in the Permission Elevation dialog boxes.

Project Version

The version of the deployment. This value does not correspond to the applications version. The value must be entered in the format of *N.N.N.N*, for example 1.0.0.0

Target Processor

The target architecture on which the application will be run. Default value is 'msil' which is platform-independent.

Trust Level

The level of trust to allow the application on client computers.

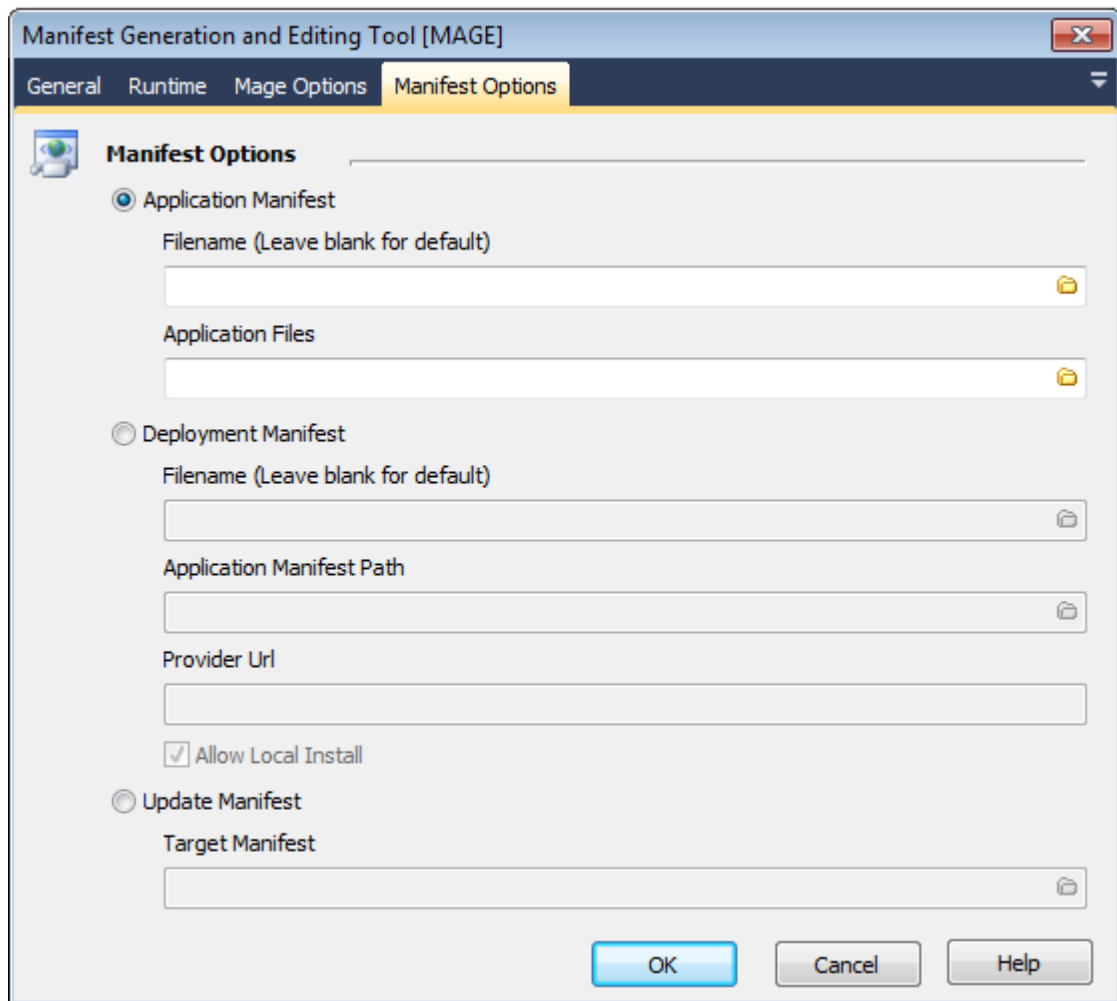
Certificate Path

The location of the certificate that will be used to sign the manifest.

Certificate Password

Used only in conjunction with the certificate path, allows the use of password

protected certificates.



Application Manifest

An application manifest is an XML file that describes an application that is deployed using ClickOnce

Filename

Specify the output file for the manifest. Default filename is 'application.exe.manifest'.

Application Files

The full path of the files that will be deployed using ClickOnce.

Deployment Manifest

A deployment manifest is an XML file that describes a ClickOnce deployment, including the identification of the current ClickOnce application version to deploy.

Filename

Specify the output file for the manifest. Default filename is 'deploy.application'.

Application Manifest Path

The full path to the application manifest file, which will be used to generate the deployment manifest.

Provider Url

The Url that ClickOnce will use to check for application updates.

Allow Local Install

Indicates whether the application should be installed on the client machines, or run from the web.

Update Manifest**Target Manifest**

The full path to the manifest that is to be updated.

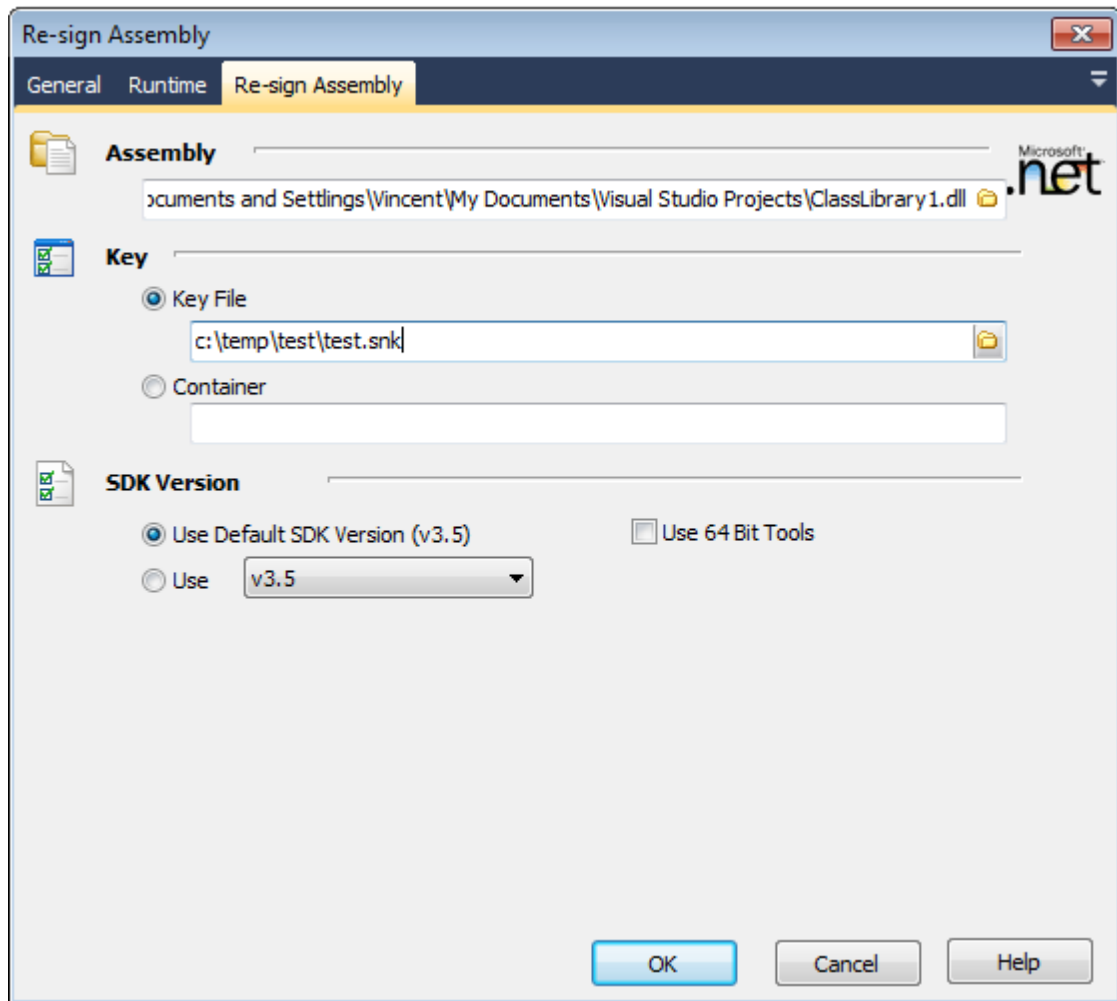
Scripting Info

The Action properties available are :

property	ProjectPath	: string
property	ProjectName	: string
property	ApplicationFilePath	: string
property	Version	: string
property	InstallLocal	: boolean
property	ManifestType	: TFBDotNetManifestMode - mmApplication, mmDeployment, mmUpdate
property	Certificate	: string
property	CertPassword	: string
property	SignManifest	: boolean
property	ProviderUrl	: string
property	AppManifestPath	: string
property	AppToFile	: string
property	DeployToFile	: string
property	UpdateTarget	: string
property	TargetProcessor	: string
property	TrustLevel	: string

6.25.2.8 Re-sign Assembly [SN]

Re-signs a previously signed or delay signed assembly.

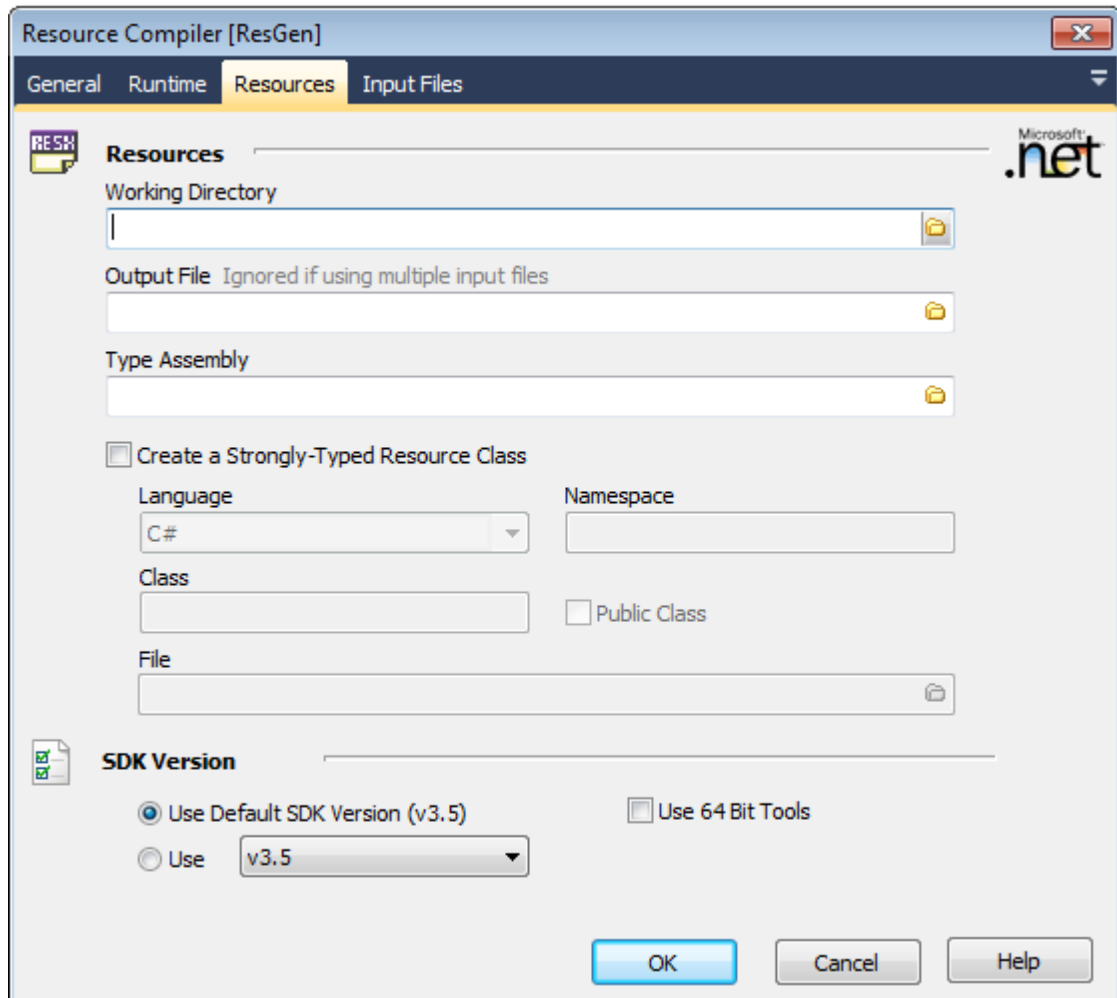


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp>

6.25.2.9 Run ResGen.exe

Resource File Generator converts .txt files and .resx (XML-based resource format) files to common language runtime binary .resources files that can be embedded in a runtime binary executable or compiled into satellite assemblies.

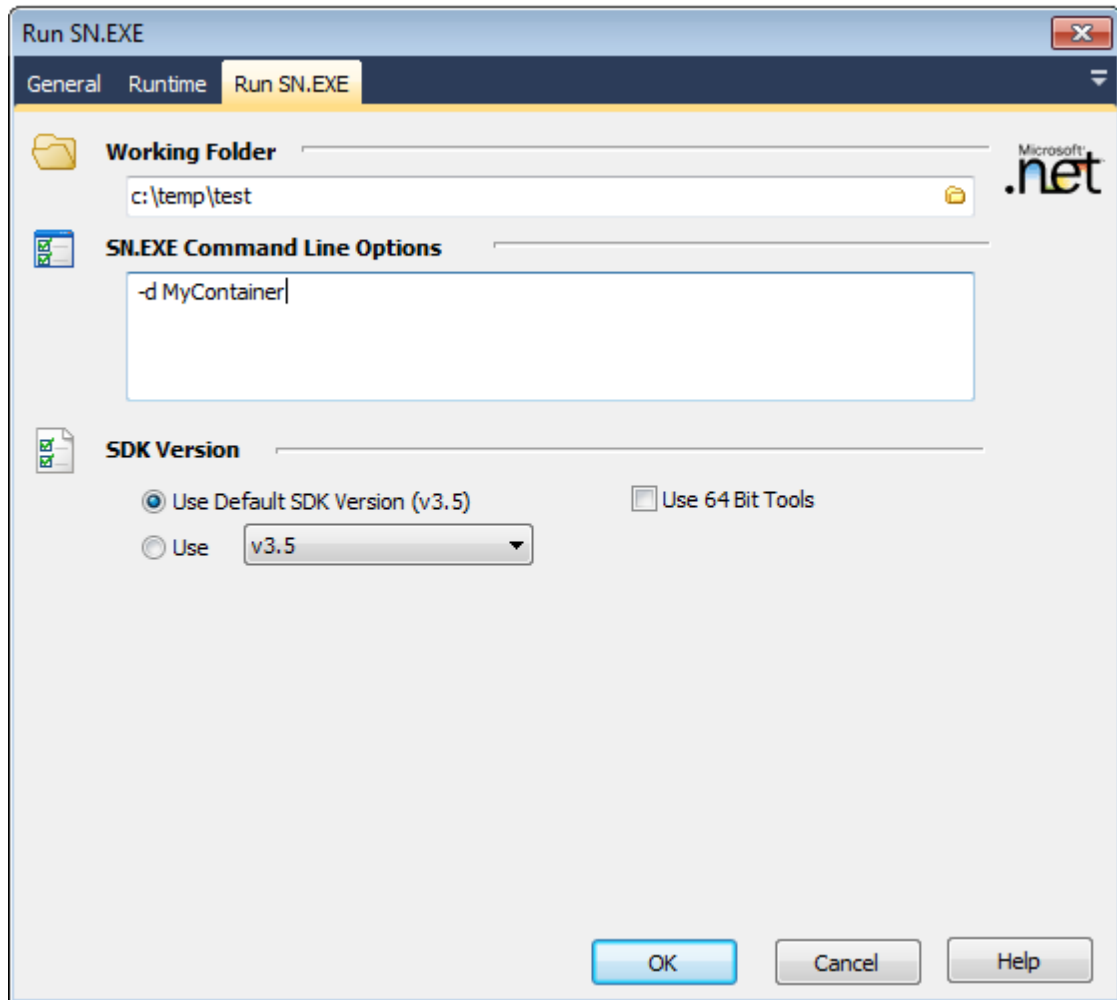


For more information see :

<http://msdn2.microsoft.com/en-us/library/cccec7sz1.aspx>

6.25.2.10 Run SN.EXE

The Strong Name tool helps sign assemblies with strong names. Sn.exe provides options for key management, signature generation, and signature verification. You should only need to use this action if the Generate Key Pair, Verify Strong Name, Install Key in Container, Extract Public Key, and Re-sign Assembly can't do what you require.

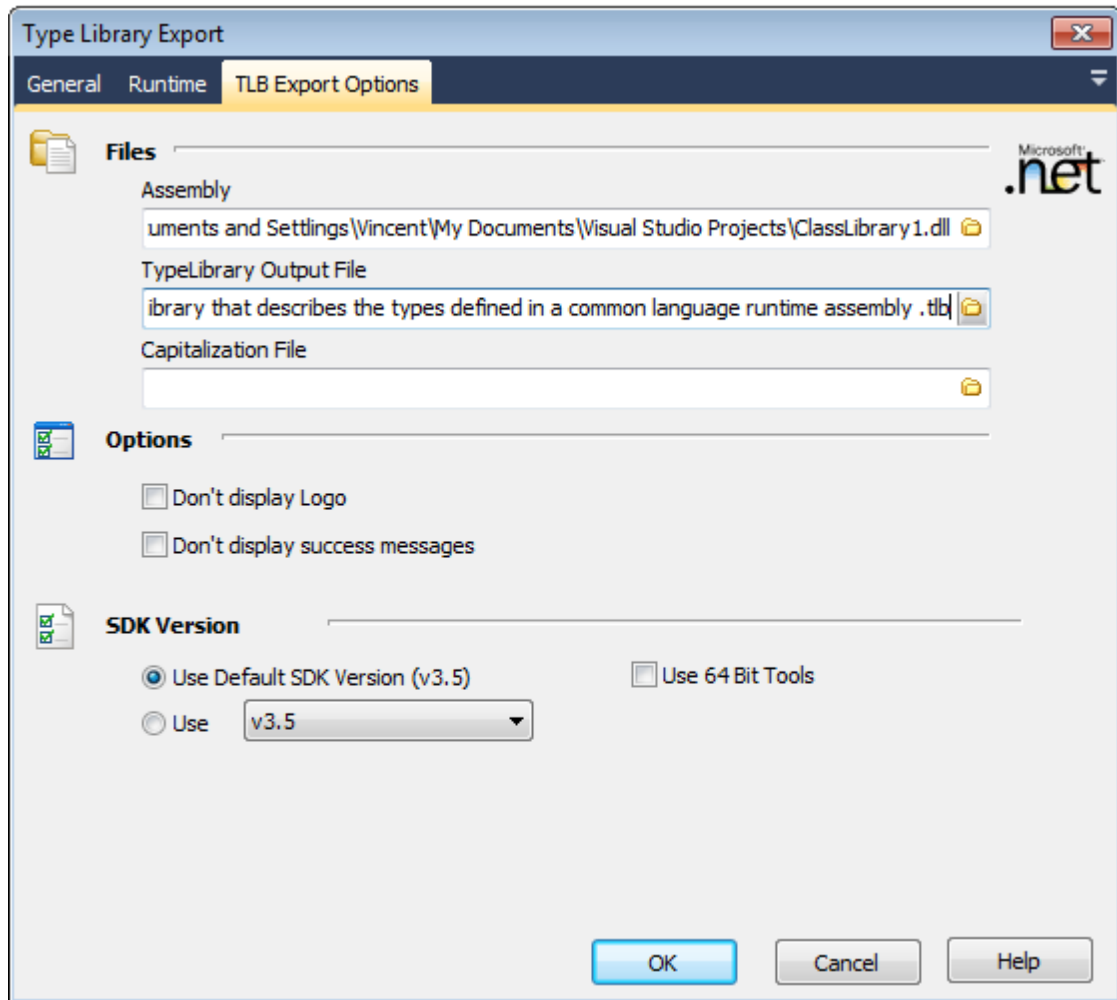


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp>

6.25.2.11 Type Library Export [TLBEXP]

The Type Library Exporter generates a type library that describes the types defined in a common language runtime assembly.

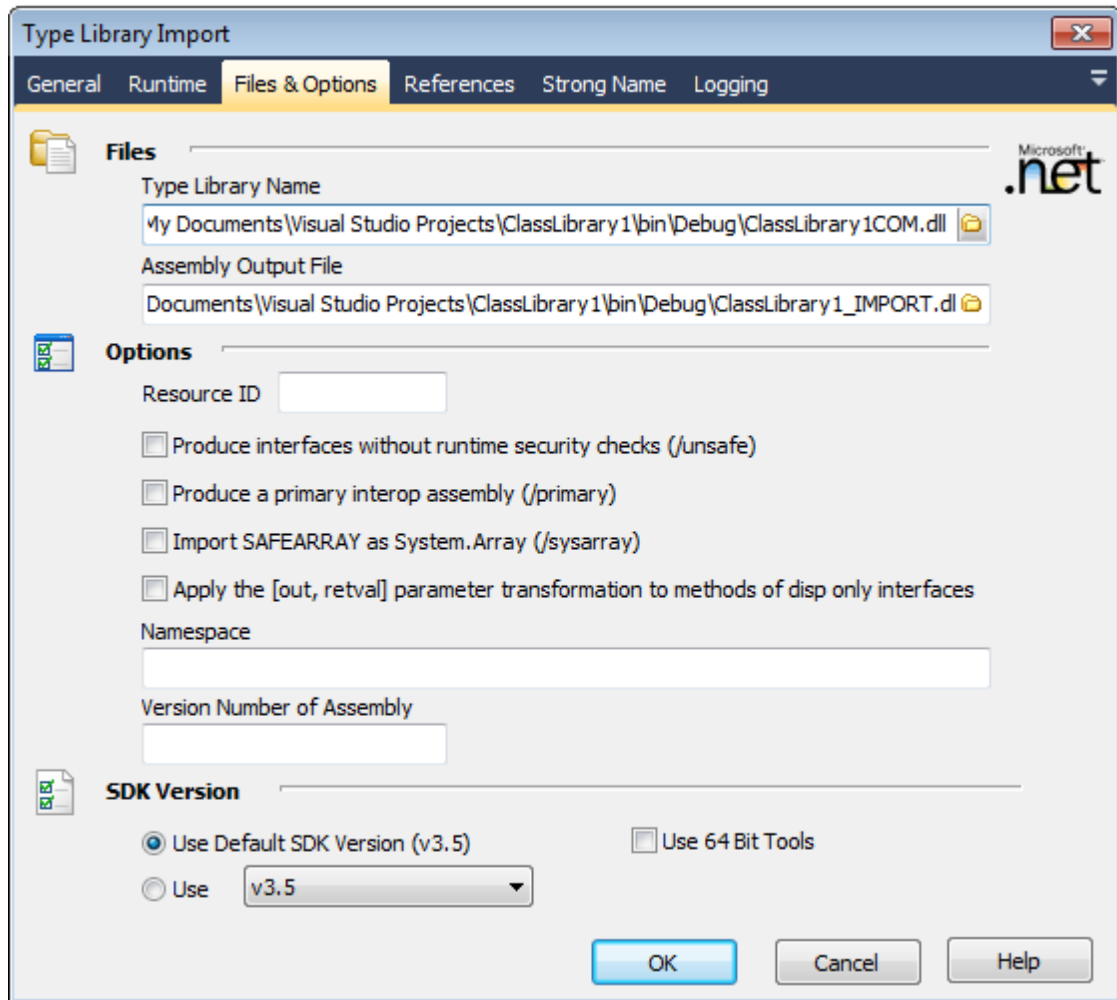


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfTypeLibraryExporterTlbExpexe.asp>

6.25.2.12 Type Library Import [TLBIMP]

The Type Library Importer converts the type definitions found within a COM type library into equivalent definitions in a common language runtime assembly. The output of Tlbimp.exe is a binary file (an assembly) that contains runtime metadata for the types defined within the original type library. You can examine this file with tools such as Ildasm.exe.

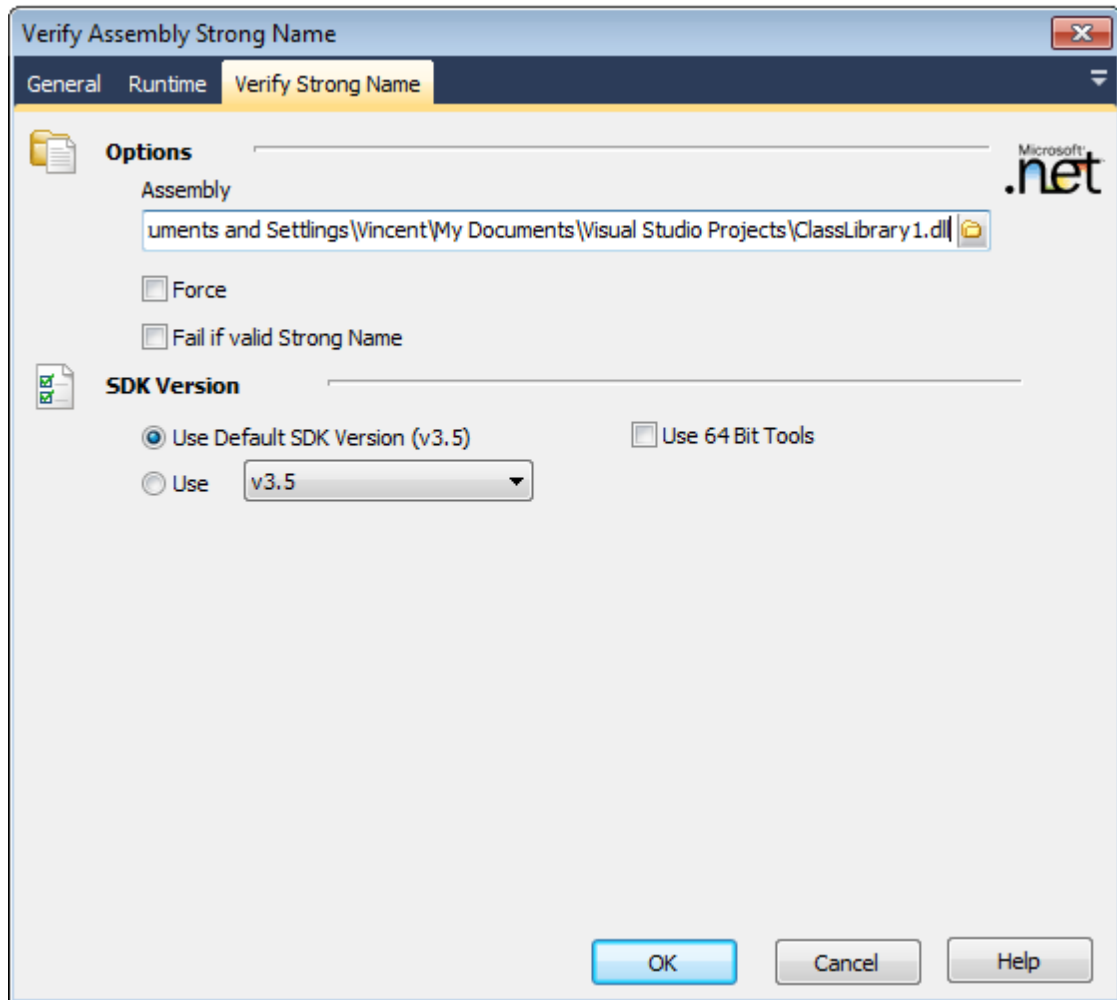


For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfTypeLibraryImporterTlbimpexe.asp>

6.25.2.13 Verify Strong Name [SN]

Verifies the strong name in assembly, where assembly is the name of a file that contains an assembly manifest.



For more information see:

<http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp>

6.25.3 3rd Party Tools

Some .Net tools have been placed in other categories:

NDoc Action

NUnit Action

MSTest

Nant Project Action

MSBuild Project Action

6.25.3.1 Demeanor

The Demeanor action enables you to automate obfuscation of your .Net assemblies using Wise Owl Demeanor

Demeanor for .NET protects your intellectual property by making it extremely difficult to reverse engineer your .NET applications. Unprotected .NET applications can be easily reverse engineered via decompilation and inspection by many decompiler products.

To counter this threat, WiseOwl has developed Demeanor for .NET -- the best .NET obfuscator available. Demeanor for .NET applies many transformations to your .NET applications that makes them much more difficult to reverse engineer.

Demeanor for .NET obfuscates the names of your types, fields, methods, properties and events by changing their names to meaningless symbols. Demeanor for .NET also obfuscates the metadata of your application, discarding all types and members that aren't needed during runtime. Demeanor for .NET also alters the control flow of your methods so that the resulting code is much harder to understand.

See the Wise Owl Demeanor homepage for more information, <http://www.wiseowl.com/products/products.aspx>

6.25.3.2 Dotfuscator

The Dotfuscator action enables you to automate obfuscation of your .Net assemblies using Dotfuscator

The .NET environment provides unprecedented flexibility and power in developing windows applications fast. Once that application is done, you better protect it. That's where Dotfuscator comes in. Dotfuscator provides powerful protection for your .NET code to protect your valuable intellectual property. After all, if your code was worth writing, isn't it worth protecting?

NOTE : The Community Edition of Dotfuscator displays a Register dialog when unregistered and an upgrade dialog when registered. This will cause the action to hang indefinitely until aborted. This action will only work correctly with the full commercial version.

See the Dotfuscator homepage for more information, <http://www.preemptive.com/products/dotfuscator/>

6.25.3.3 FxCop

The FxCop action enables you to automate compliance testing of your .Net assemblies using FxCop

FxCop is a code analysis tool that checks .NET managed code assemblies for conformance to the Microsoft .NET Framework Design Guidelines. It uses reflection, MSIL parsing, and callgraph analysis to inspect assemblies for more than 200 defects in the following areas:

- Library design
- Localization
- Naming conventions
- Performance
- Security

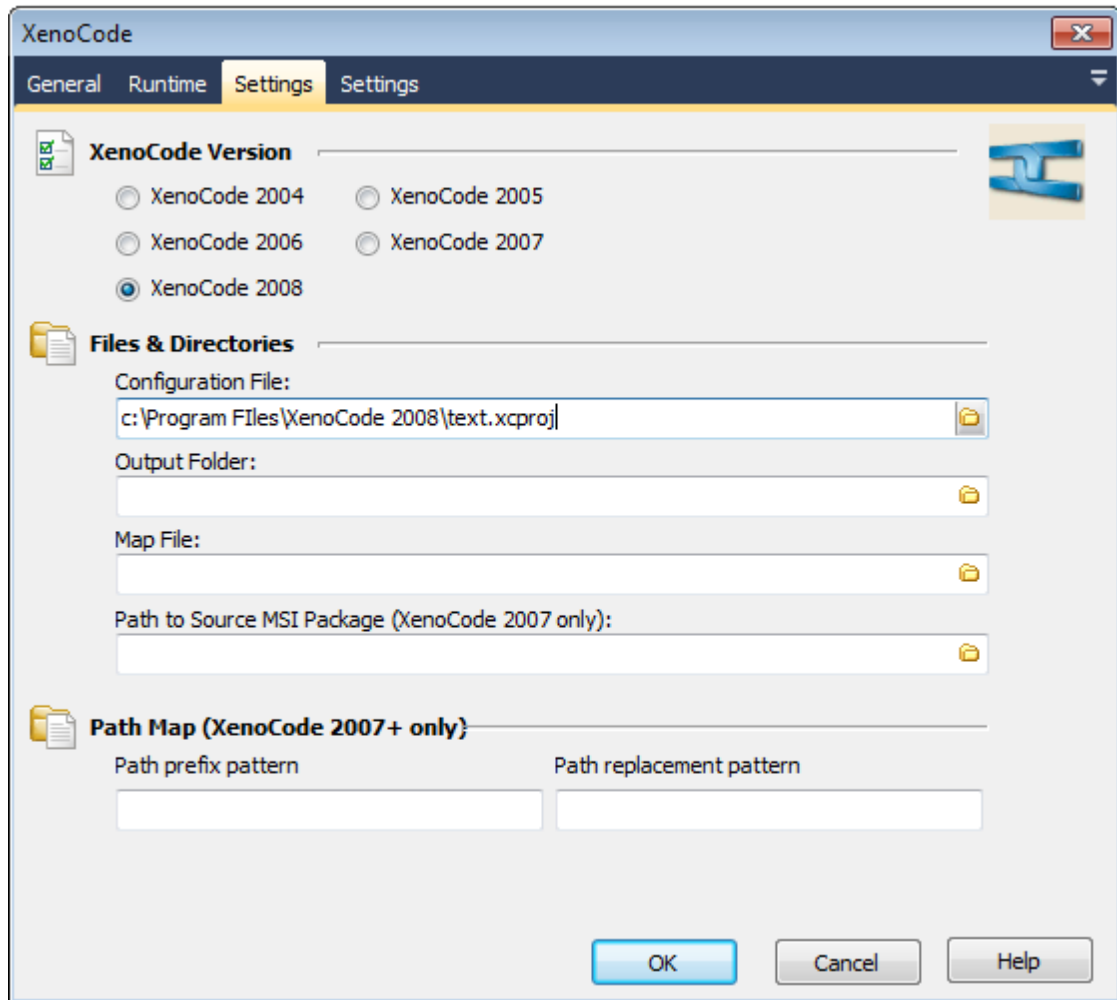
See the FxCop homepage for more information, <http://www.getdotnet.com/team/fxcop/>

6.25.3.4 XenoCode

The XenoCode action enables you to automate obfuscation and optimization of your .Net assemblies using XenoCode

XenoCode is the powerful, flexible, and easy-to-use code protection and deployment solution for .NET developers.

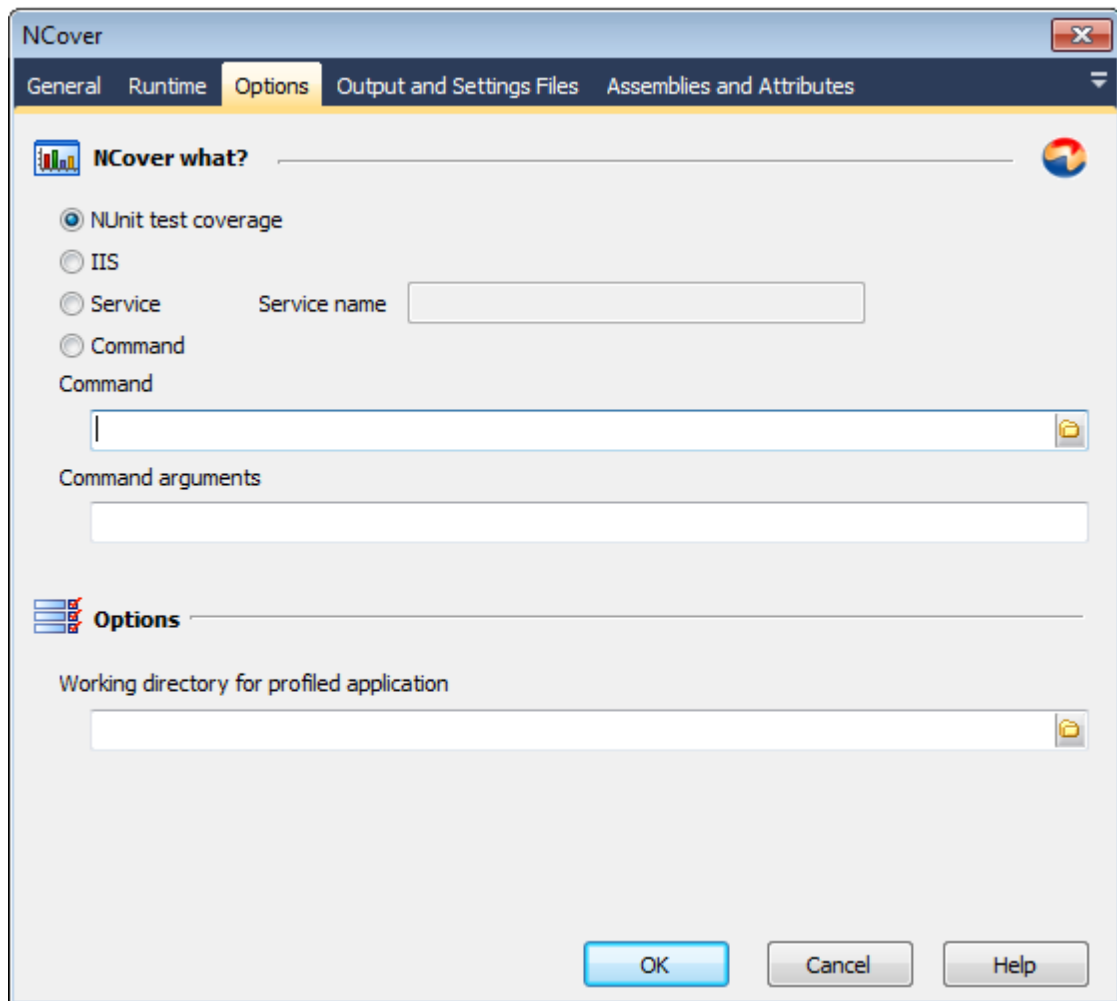
See the XenoCode homepage for more information, <http://www.xenocode.com/>



The XenoCode action supports both XenoCode 2004 (version 2.x) and XenoCode 2005 (version 3.x). Set up the locations for the XenoCode executable location and default XenoCode version in the Tools->Options menu.

If no output folder is specified, then the output folder is set to a XenoCode subdirectory of the configuration folder.

6.25.3.5 NCover



NCover What?

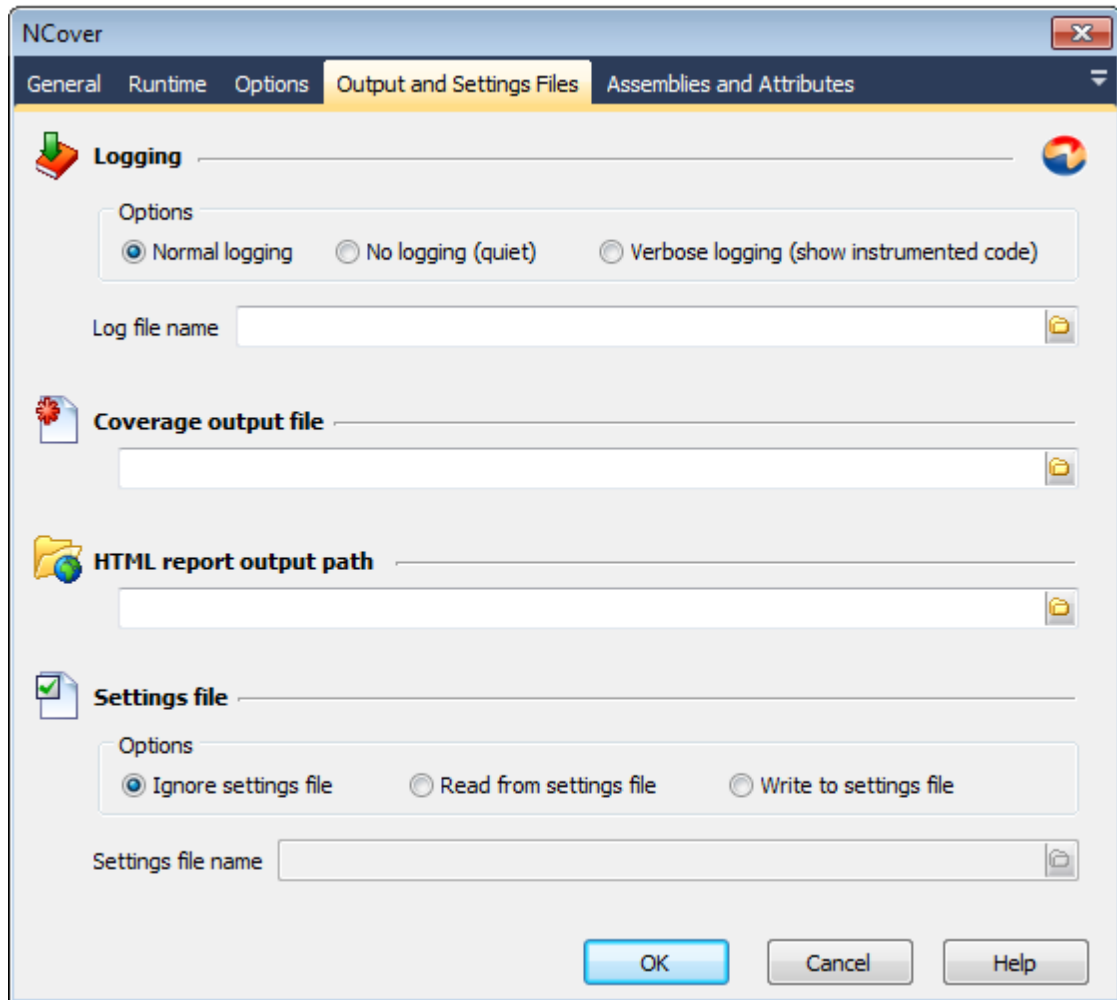
Choose from the following items to run NCover over:

- NUnit - Allows coverage of NUnit unit tests.
- IIS - Allows coverage of IIS.
- Windows Services - Allows coverage of a service.
- Command - Allows you to specify the command line that NCover will use to run the external tool that will be used for code coverage.

Note: When selecting NUnit as the application to execute the code you need to provide the command arguments to NUnit in the 'Command Arguments' field. If you're testing a single assembly it's enough to just specify the assembly path relative to the working directory (i.e. Program.Tests.dll).

Options

Working directory - specify the working directory that will be set when running NCover.



Logging

Specify the level of logging and the file name that will be used for logging.

Coverage Output File

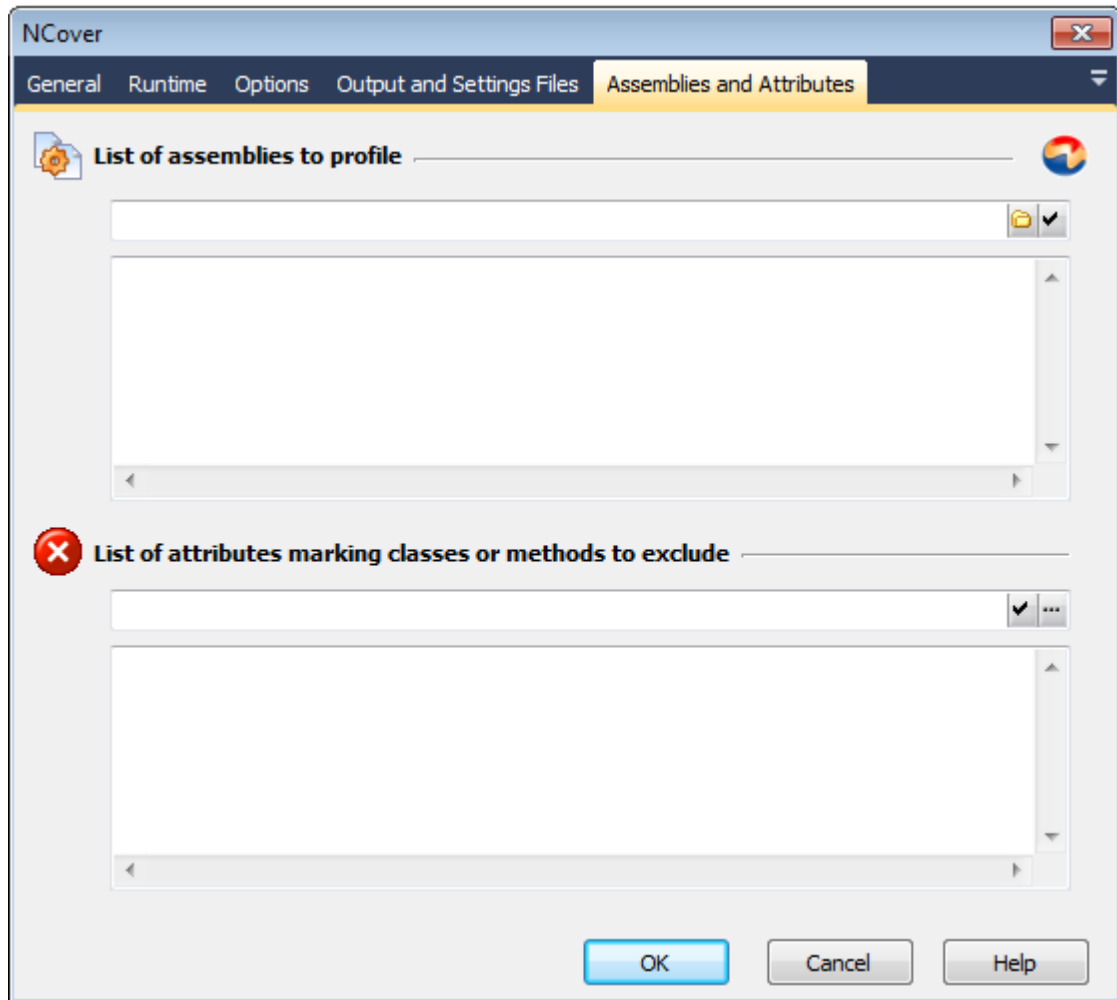
Specify the name of the file for coverage output.

HTML Report output path

Specify the path to be used for HTML reports.

Settings File

- Save Settings - Save coverage settings to a file.
- Read Settings - Set coverage settings based on a settings file.



List of assemblies to profile

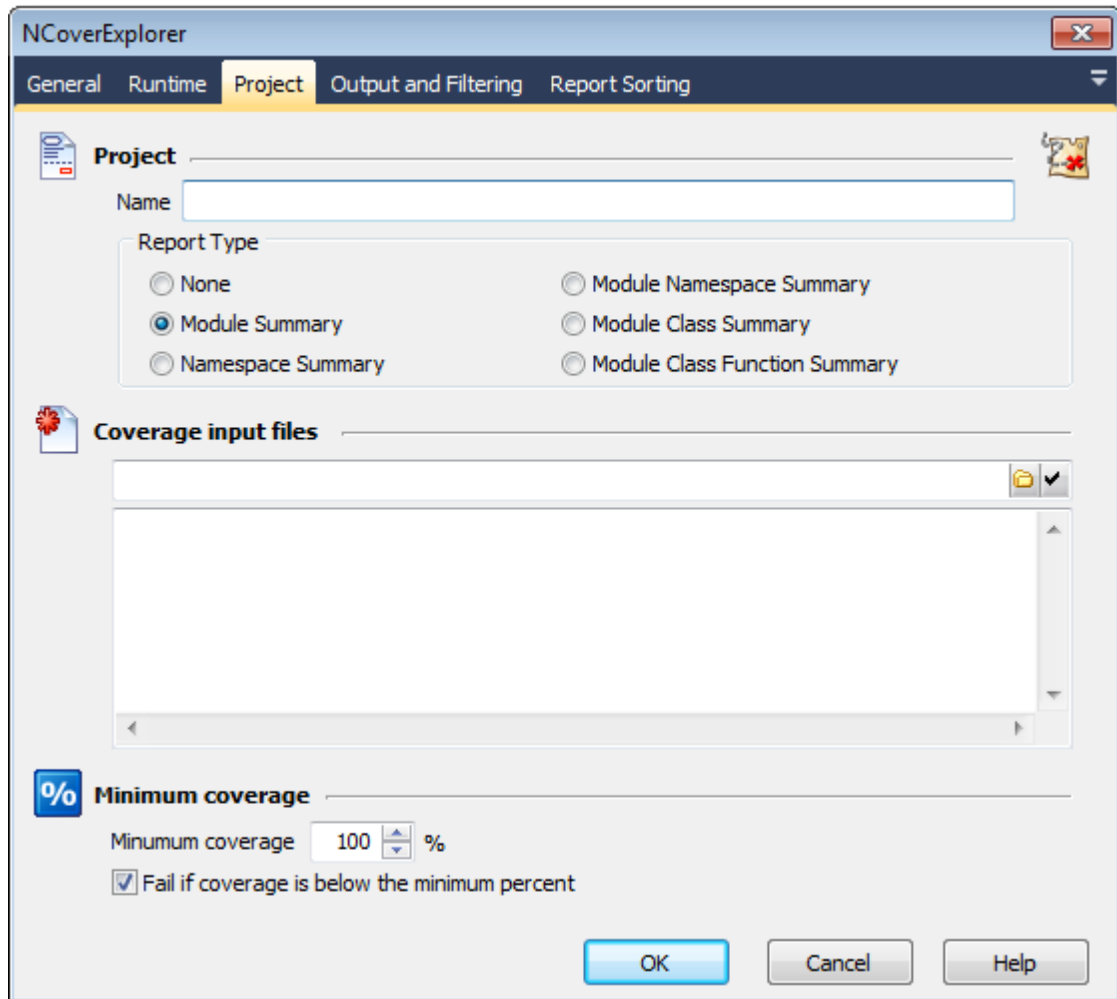
Browse or enter an assembly to profile and then add it to the list by clicking the tick button.

List of attributes marking classes or methods to execute

Click the ellipse (...) button to add NUnit attributes.

For more information on NCover options, please visit: <http://www.ncover.com/documentation/console/flags>

6.25.3.6 NCoverExplorer



Project

Name - The name of the project to appear in the report.

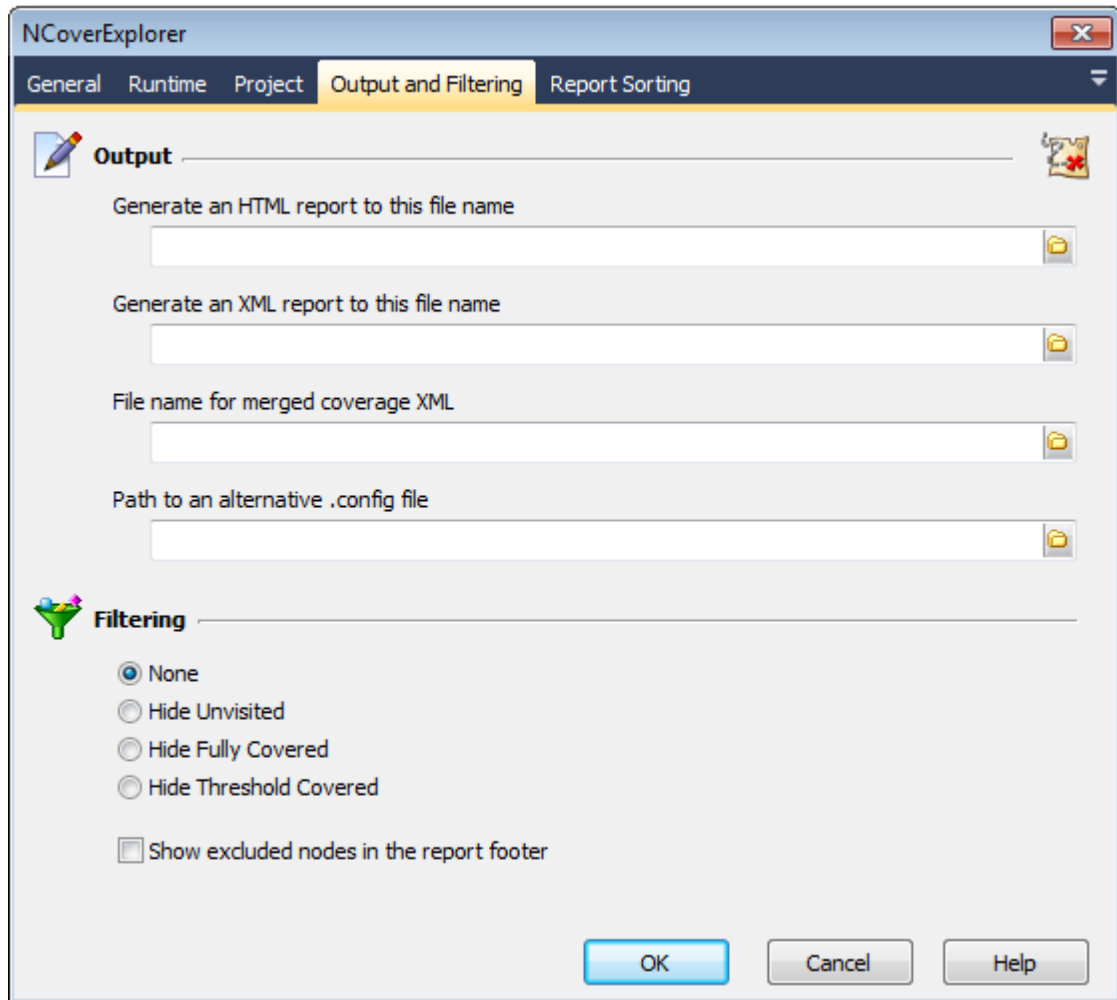
Report Type - Specifies report type to be generated.

Coverage input files - A list of input files, one per line. Can include wildcards, like *.Xml

Minimum Coverage

Minimum coverage - Specifies the minimum coverage percentage

Fail - Fail if any assembly's coverage is lower than the specified minimum.



Output

Enter the name of the HTML and/or XML report required.

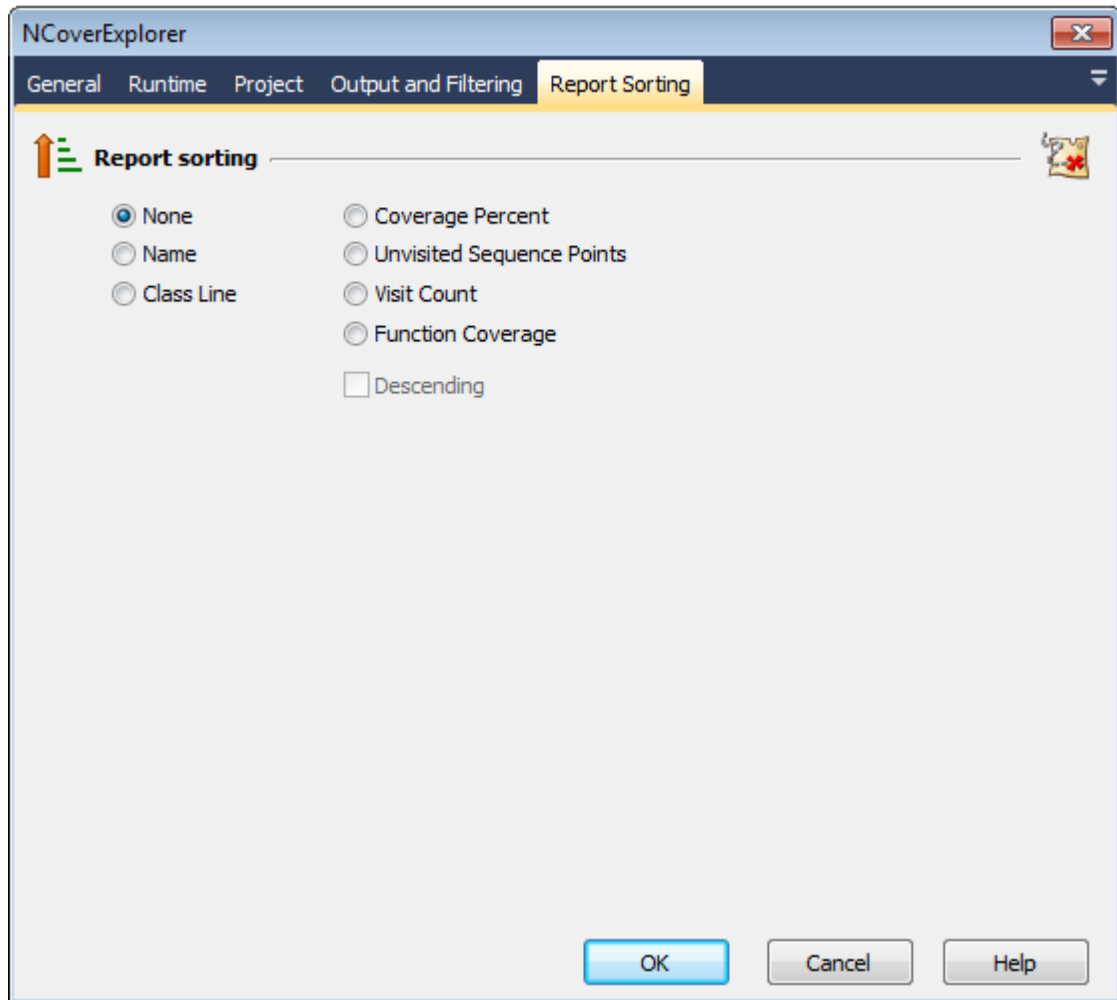
File name for merged coverage XML - Specifies the file to save merged coverage data to

Path to an alternative .config file - Path to an alternative settings file

Filtering

Specify the filtering for the report.

Specify whether or not to show excluded nodes in the report footer



Report sorting

Specify the sorting for the report

For more information on NCoverExplorer options, please visit: <http://www.ncover.com/documentation/explorerconsole/flags>

6.25.3.7 NCover Reporting

The NCover Reporting action allows you to generate a range of reports from code coverage data (which can be generated using the NCover action). Over 20 different reports can be generated within a single execution of the NCover Reporting action.

The **Reporting** page allows you to specify the Coverage files to be included in the report.

NCover Reporting

General Runtime **Reporting** Output Filters Thresholds Options

NCover Reporting

Coverage Files

%FBPROJECTDIR%\NCover\CoverageFiles\Coverage1.xml
%FBPROJECTDIR%\NCover\CoverageFiles\Coverage2.xml
%FBPROJECTDIR%\NCover\CoverageFiles\Coverage3.xml

Append Trend to a File (Optional):
%FBPROJECTDIR%\NCover\Trends\TrendFile.trend

Load Trend From a File (Optional):

Custom Stylesheet (Optional):

Project Name (Optional):
My Application

Build Id (Optional):

Sort Report Elements By:
Name

Hide Report Elements:
HideFullyCovered

Top Uncovered To Report
10

☐ Ignore Failed Builds

OK Cancel Help

Coverage Files: Specify a list of coverage files to be included in the output report. Each coverage file needs to be listed on a separate line within the field provided. Coverage files can be generated by the NCover action.

Append To Trend File: Specify a file to append the coverage trend data to. If the file does not exist it will be created.

Load From Trend File: To load existing trend data to be included in the report specify an existing trend file.

Custom Stylesheet: Specify a custom XSLT file to be used to format the report.

Project Name: The value entered here will appear in the heading of the report.

Build Id: Specify a Build Id for the Merged Coverage Data File Name.

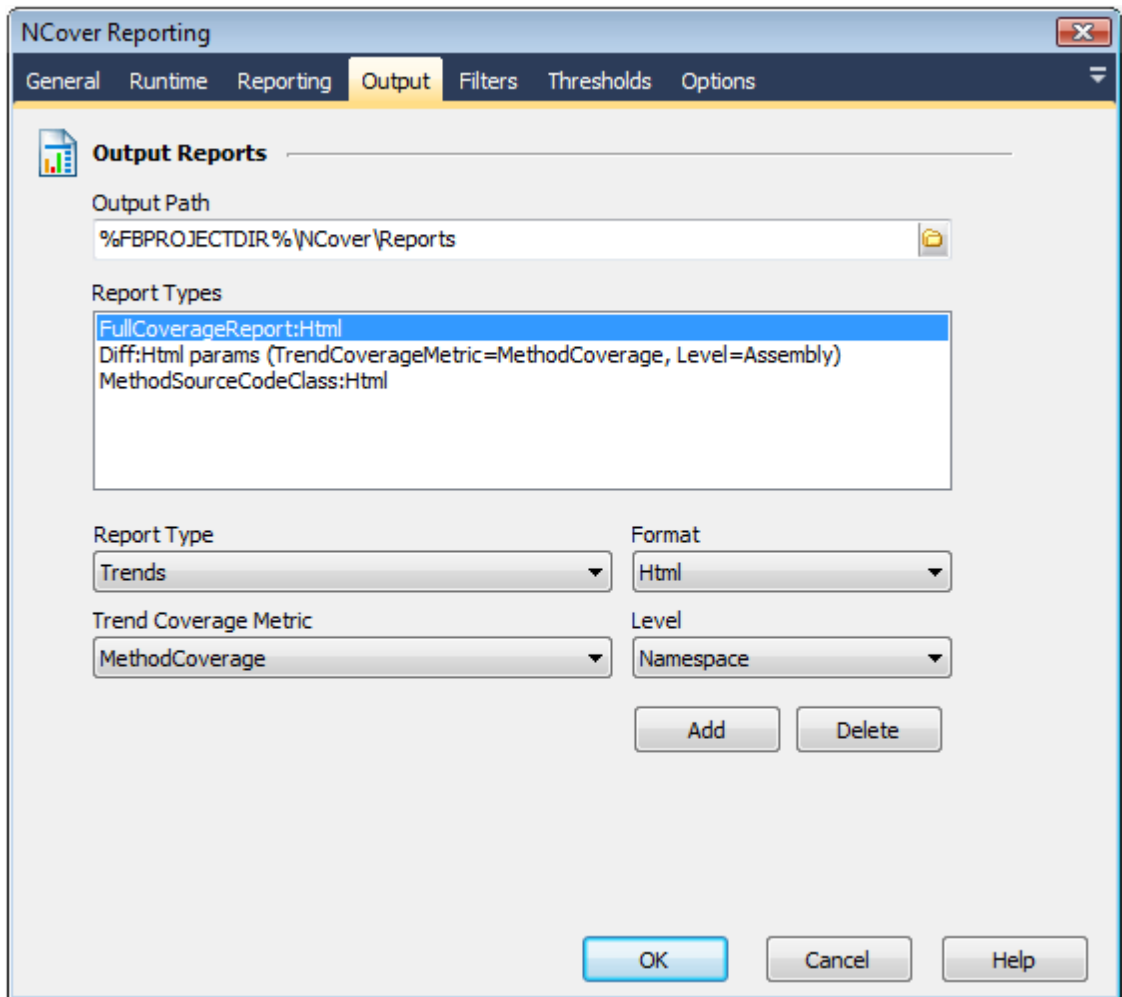
Sort Report Elements: Allows you to specify how the data in the report is sorted.

Hide Report Elements: Allows you to hide specific elements within the report.

Top Uncovered To Report: Control how many of the Top Uncovered items are displayed in the report.

Ignore Failed Builds: When including trend data, builds with failing tests will be included. Enabling this option will exclude data where tests failed.

The **Output** page allows you to specify the output directory where the reports will be stored. You can also generate a list of the reports that will be generated when the action runs.

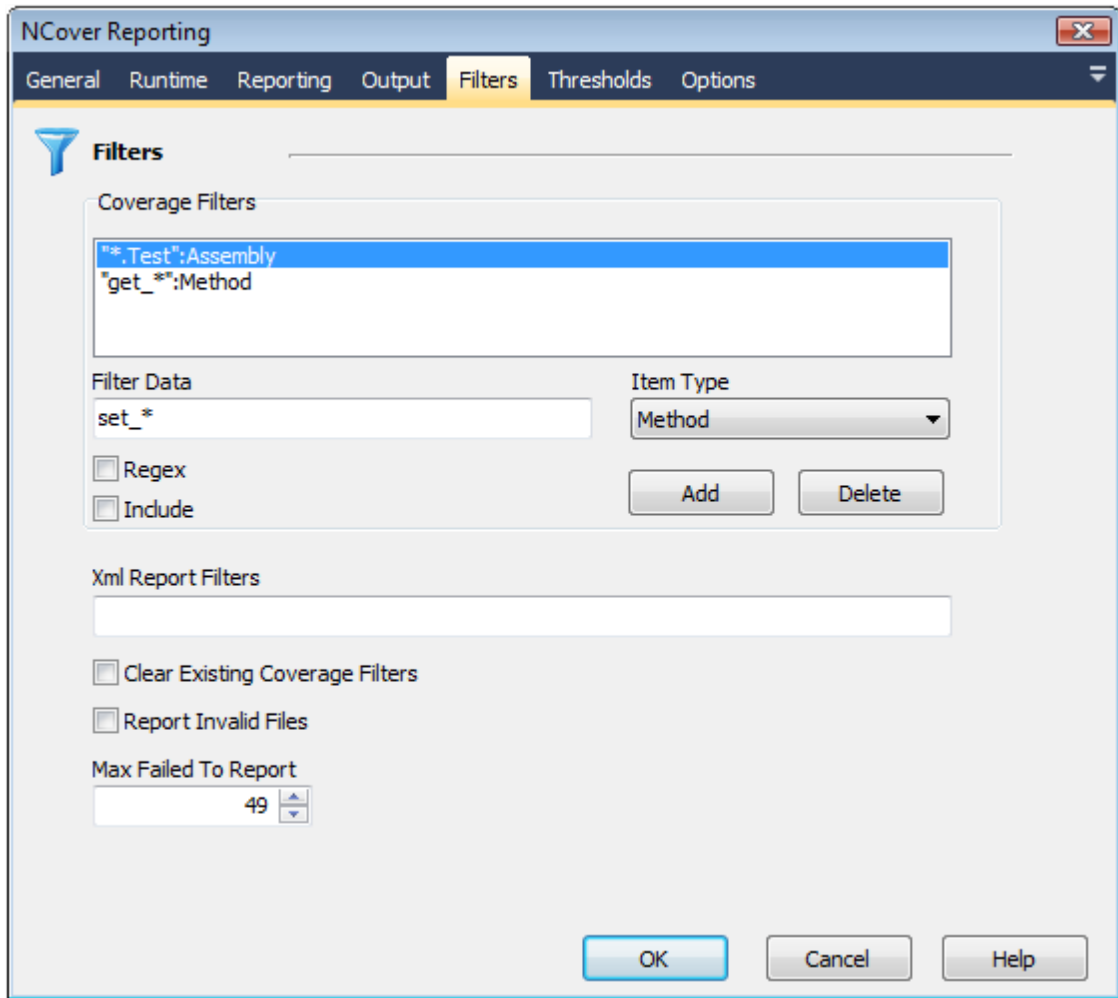


Output Path: Specify the path where the reports will be stored.

Report Types: Use the controls provided to add the desired report types to the list. Multiple reports can be generated at once.

For more information see NCover Report Types.

The **Filters** page provides a means of excluding specific data from being included in the report.



Coverage Filters: Use the controls provided to set filters to include/exclude specific elements from the report.

Example:

The following filter will exclude all assemblies that end with 'Test':

```
"*Test":Assembly
```

The following filter excludes all assemblies that don't begin with 'myproject':

```
"myproject*":Assembly:False:True
```

For more information see NCover Advanced Reporting Features.

Xml Report Filters: Allows you specify which details should be omitted from Xml reports, may be necessary when generating large reports.

Clear Existing Coverage Filters: Clear existing filters on the view.

Report Invalid Files: Rather than fail the action if one of the coverage files is invalid, enabling this option will report the invalid coverage files and create the report with the remaining

valid coverage files.

Max Failed To Report: Specify the maximum amount of items that fail the minimum coverage thresholds.

The **Thresholds** page allows you to specify minimum and satisfactory threshold levels. Minimum coverage thresholds allow you to fail the action if the coverage criteria specified is not satisfied. The satisfactory thresholds allow you to report on values that do not meet specified criteria.

Note: The Threshold features are only available in NCover Complete Edition.

The screenshot shows the 'NCover Reporting' window with the 'Thresholds' tab selected. The window has a menu bar with 'General', 'Runtime', 'Reporting', 'Output', 'Filters', 'Thresholds', and 'Options'. The 'Thresholds' section is divided into two main areas: 'Satisfactory Coverage Thresholds' and 'Minimum Coverage Thresholds'. Each area has a list box for existing thresholds, a 'Coverage Metric' dropdown, a 'Threshold' spinner, an 'Item Type' dropdown, and a 'Pattern' text box. In the 'Satisfactory' section, the list box contains 'CydomaticComplexity:60:Namespace', 'SymbolCoverage:65', and 'MethodCoverage:80'. The 'Coverage Metric' is 'MethodCoverage', the 'Threshold' is '80', and the 'Item Type' is empty. In the 'Minimum' section, the list box contains 'MethodCoverage:100:Class: ". *Test\$"', the 'Coverage Metric' is 'MethodCoverage', the 'Threshold' is '100', and the 'Item Type' is empty. At the bottom of the dialog is a checkbox labeled 'Use Minimum Coverage' which is unchecked. 'Add' and 'Delete' buttons are present next to the list boxes. At the very bottom are 'OK', 'Cancel', and 'Help' buttons.

Satisfactory Coverage Thresholds: Use the controls provided to set the Satisfactory Thresholds.

Minimum Coverage Thresholds: Use the controls provided to set the Minimum Thresholds.

Examples:

The following example will fail any namespace that has a cyclomatic complexity greater than 60:

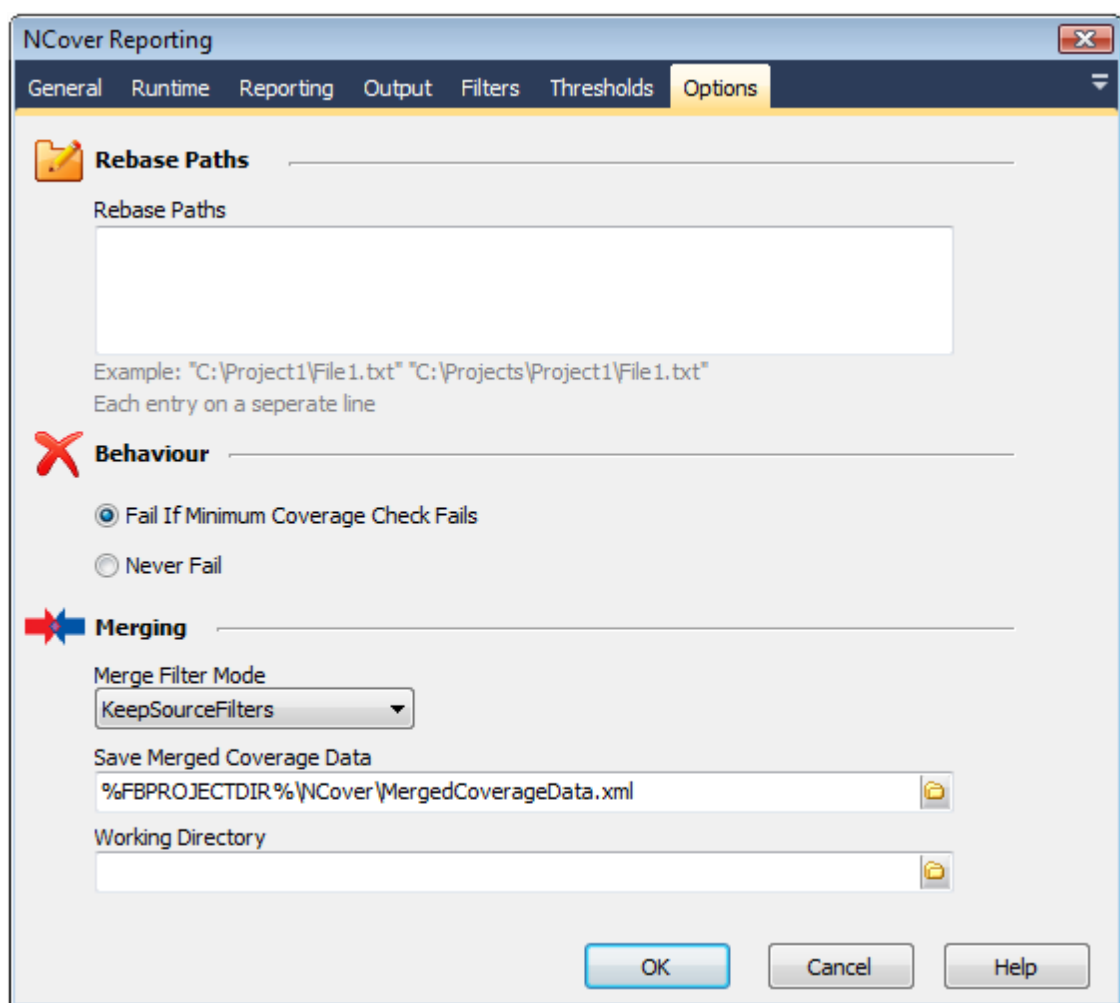
CyclomaticComplexity:60:Namespace

The following example will fail any Class ending in 'Test' where all methods have not been executed:

*MethodCoverage:100:Class:". *Test\$"*

For more information see NCover Advanced Reporting Features.

The **Options** page allows you to rebase source files, determine action behaviour and set merge options.



Rebase Paths: Specify source paths to be rebased in the form of "<Source Path>" "<Destination Path>" with each entry on a separate line.

Behaviour: This option allows you to determine whether the action will fail if the Minimum

Coverage Checks fail or whether the action will pass regardless of the Minimum Coverage Check outcome.

Merge Filter Mode: This determines how the coverage data is merged into the base coverage file.

Save Merged Coverage Data: Allows you to save the merged coverage data and the filters to the specified file.

Working Directory: Allows you to set a working directory so that relative paths can be used.

For more information see NCover Reporting.

6.25.3.8 NuGet

NuGet Overview

NuGet makes it easy to add, remove, update and share libraries and tools in Visual Studio projects that use the .NET Framework.

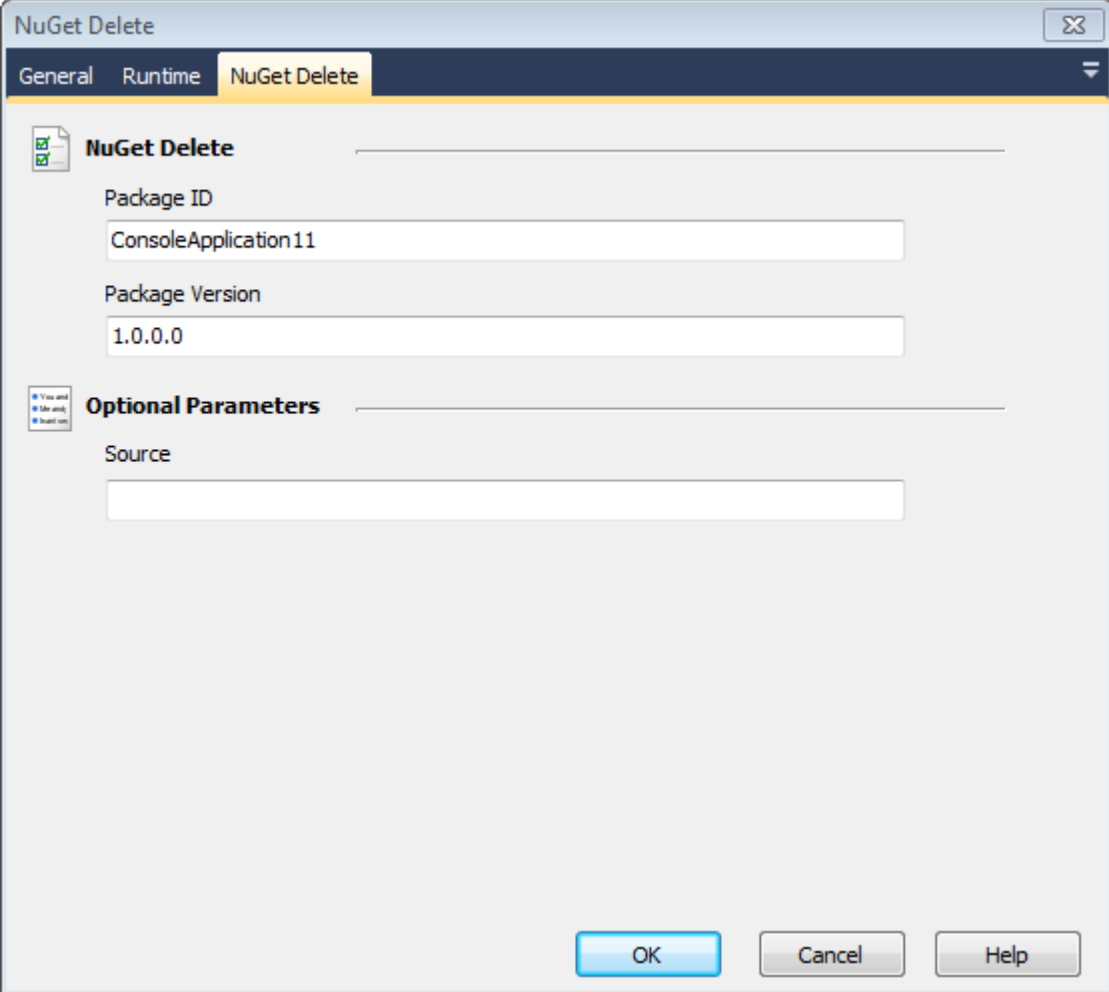
To create and publish a package, the general workflow would be to use Spec to create a .nuspec file, create a .nupkg package file using Pack, and then publish the package to a NuGet server using Push.

For more information please see nuget.org

6.25.3.8.1 NuGet Delete

NuGet Delete

The Delete action deletes a package from either the default or specified feed.

The image shows a 'NuGet Delete' dialog box with a title bar and a close button. It has three tabs: 'General', 'Runtime', and 'NuGet Delete', with the last one selected. The 'NuGet Delete' tab contains a 'NuGet Delete' section with a green checkmark icon, followed by 'Package ID' (text 'ConsoleApplication11') and 'Package Version' (text '1.0.0.0'). Below this is an 'Optional Parameters' section with a blue icon and a 'Source' text box. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

NuGet Delete

General Runtime **NuGet Delete**

NuGet Delete

Package ID
ConsoleApplication11

Package Version
1.0.0.0

Optional Parameters

Source

OK Cancel Help

Package Id

The Id of the package that you wish to delete.

Version

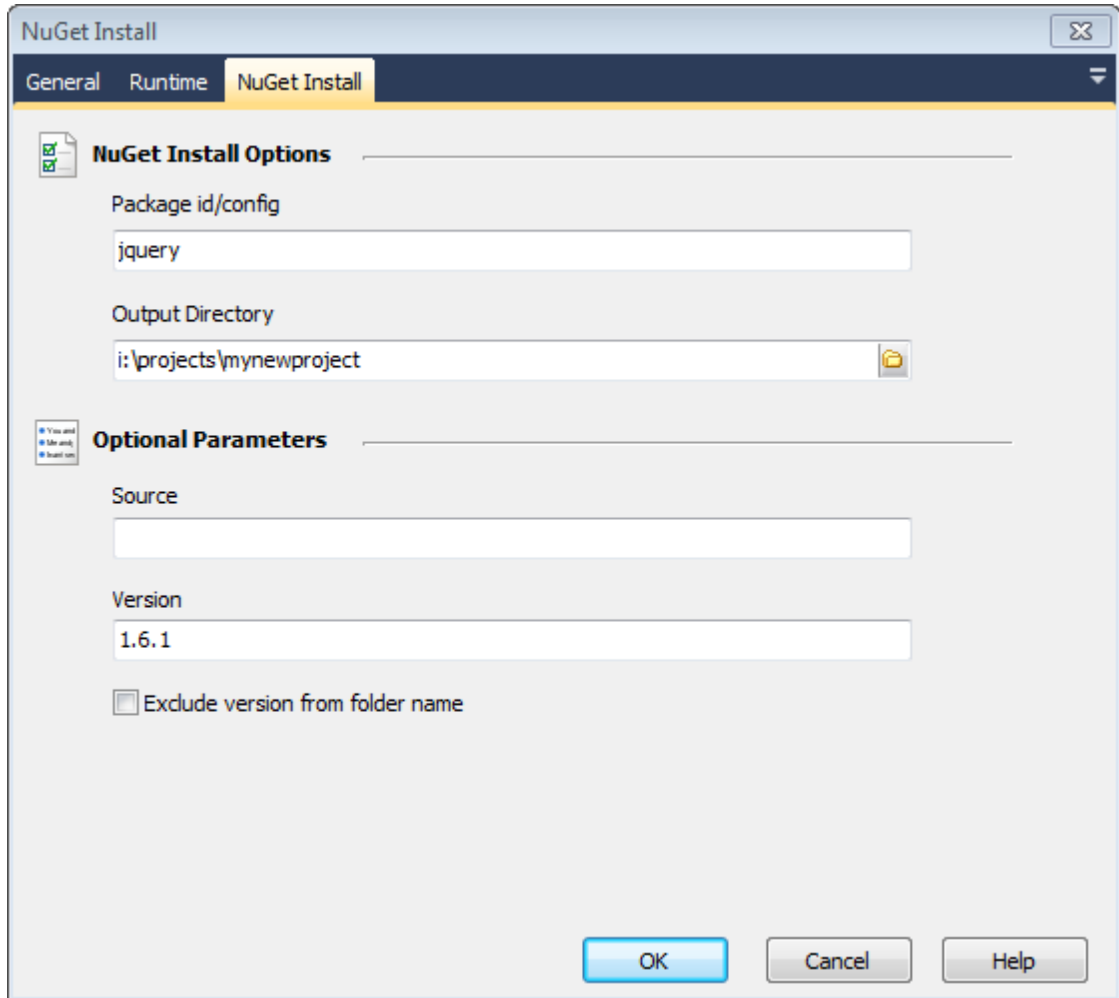
The version of the package that you wish to delete.

Source

The server that you wish to delete the package from.

6.25.3.8.2 NuGet Install**NuGet Install**

The Install action downloads and installs the package from either the default or specified feed. You can use this action to ensure your automated build is using the most recent, or a specific, version of a library.

**Package Id/Config**

The id of the package to install, or the path to the a packages.config file.

Output Directory

The directory to place to downloaded package

Source

You can use this field to specify a different source feed. For example, your company may host its own internal NuGet package feed for sharing internal components. If left blank the package will be installed from the default package gallery, which you can view at <http://nuget.org/List/Packages>

Version

You can use this field to specify a version of the package. If left the latest version will be installed.

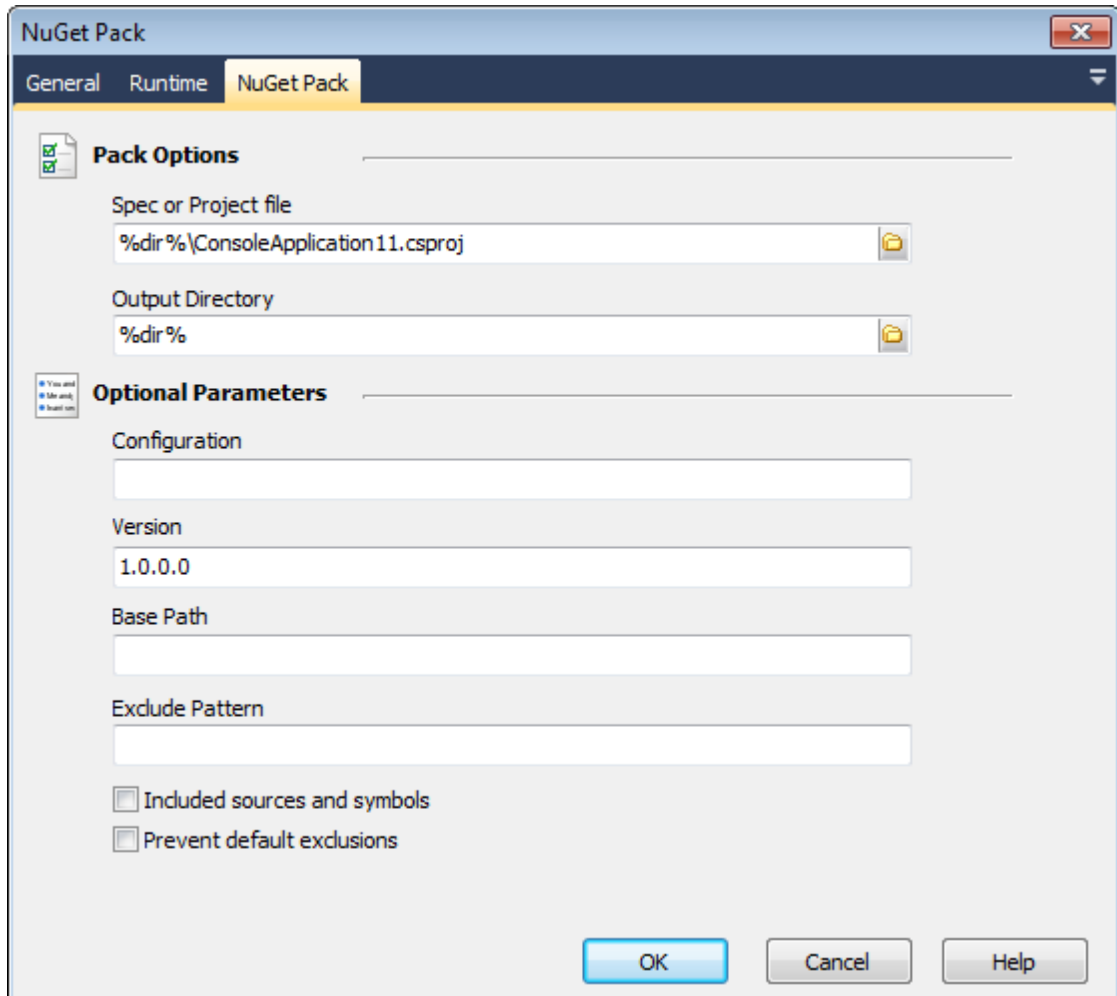
Exclude version from folder name

By default, the package will be downloaded into a folder which is named <packageid><version>. Selecting this option causes the folder to be named <packageid>.

6.25.3.8.3 NuGet Pack

Nuget Pack

The Pack action creates a NuGet package based on the specified spec or project file.

**Spec or Project file**

The path to a .nuspec or .csproj file to create the package from.

Output Directory

The directory where the package should be created.

Configuration

Determines which configuration should be used when building the package from a project (.csproj) file.

Version

Overrides the version number provided in the nuspec file.

Base Path

The base path of the files defined in the nuspec file.

Exclude Pattern

One or more wildcard patterns to exclude when creating a package.

Include sources and symbols

When building a package from a project (.csproj) file, this option causes the source files and symbols to be included in the package.

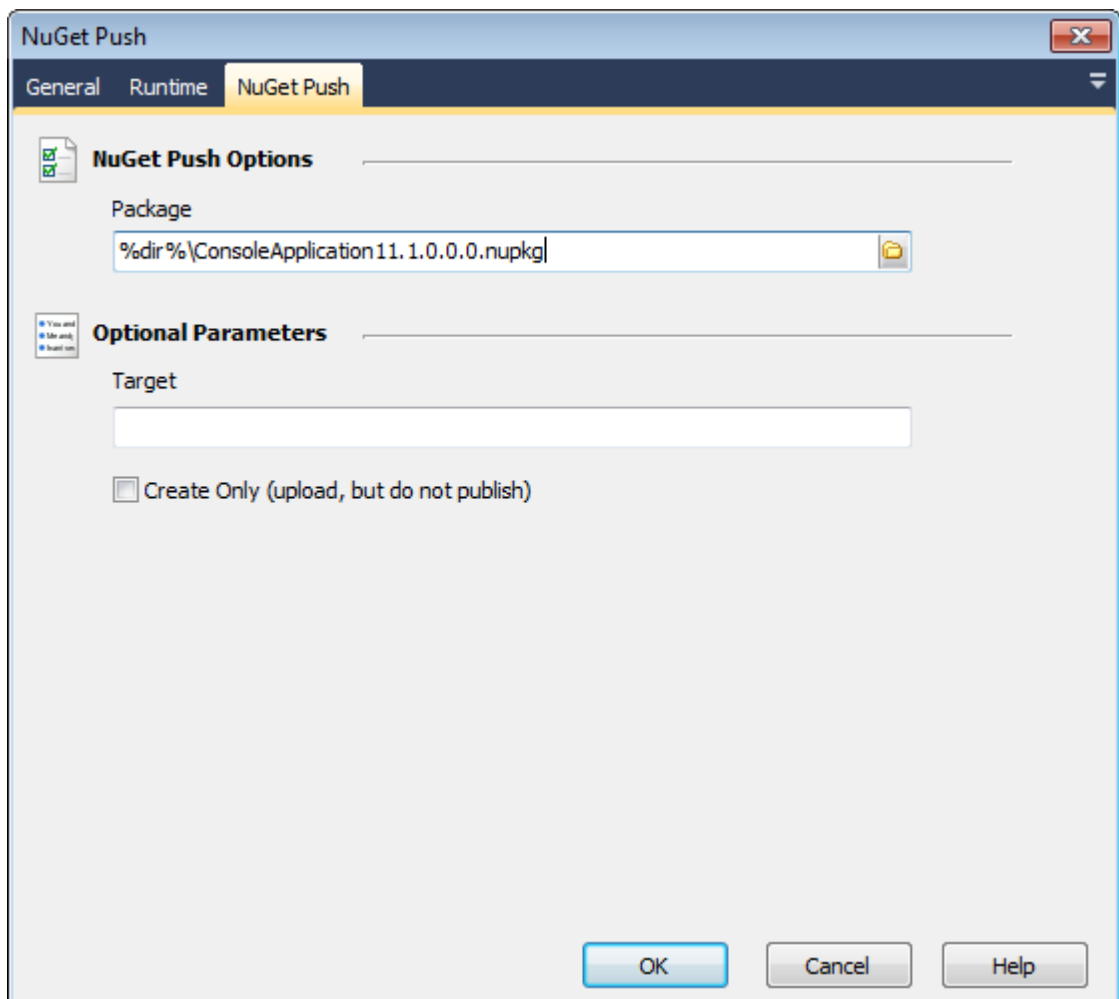
Prevent default exclusions

Prevents the default exclusion of files and folders beginning with a period (eg .hg/, .hgignore)

6.25.3.8.4 NuGet Push

NuGet Push

The Push action pushes a package to either the default or specified feed.

**Package**

The path to the package that you wish to publish.

Target

The server you wish to publish the package to. If not specified the default server is used.

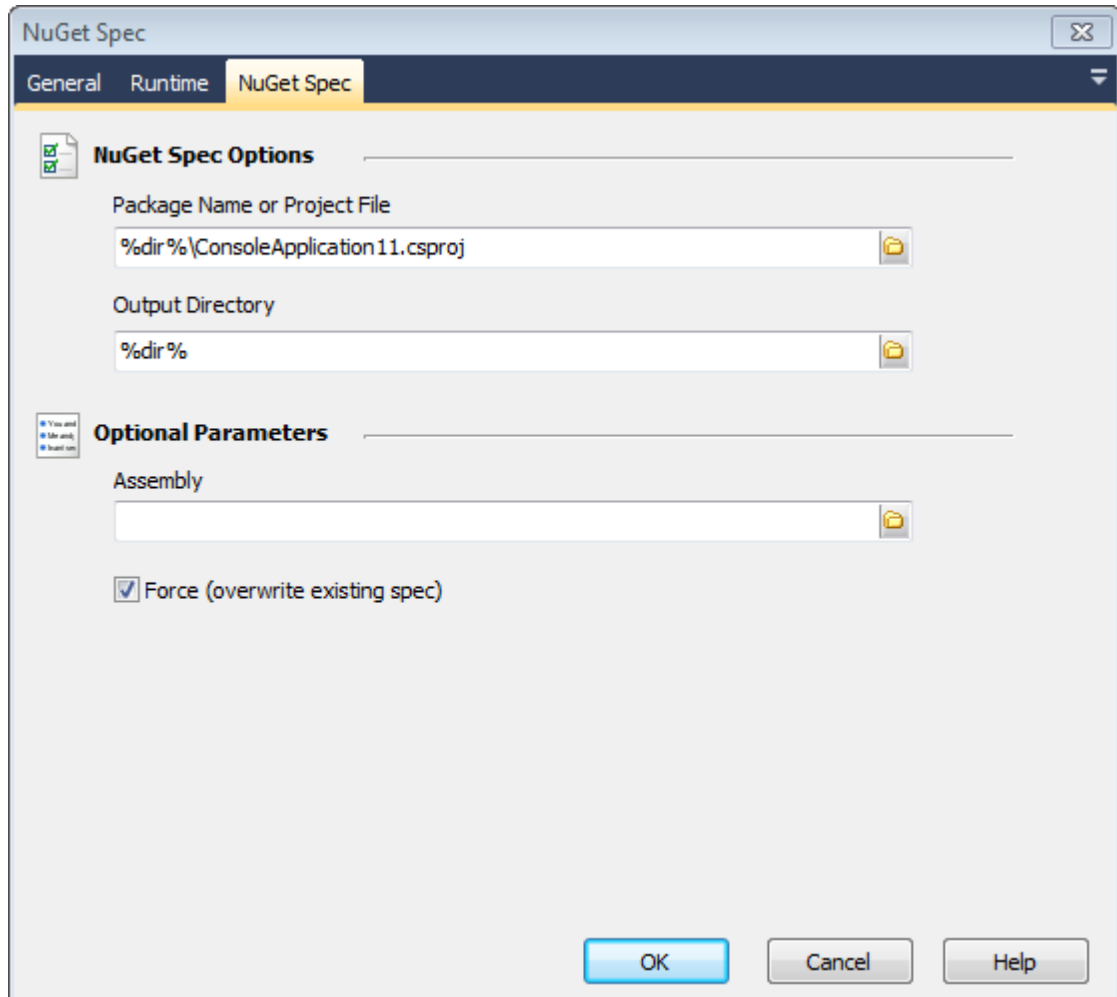
Create Only

If checked, the package is uploaded to the server but not made publicly available.

6.25.3.8.5 NuGet Spec

NuGet Spec

The Spec action creates a nuspec for a package. The nuspec is later used by the Pack action to create the NuGet package.

**Package Name or Project File**

You can specify either a project name or a project file. A project name will cause a new blank spec to be created. If you specify a Project File the nuspec will be based on the project metadata.

Output Directory

The directory where the nuspec file should be created.

Assembly

You can optionally specify an assembly from which to get metadata.

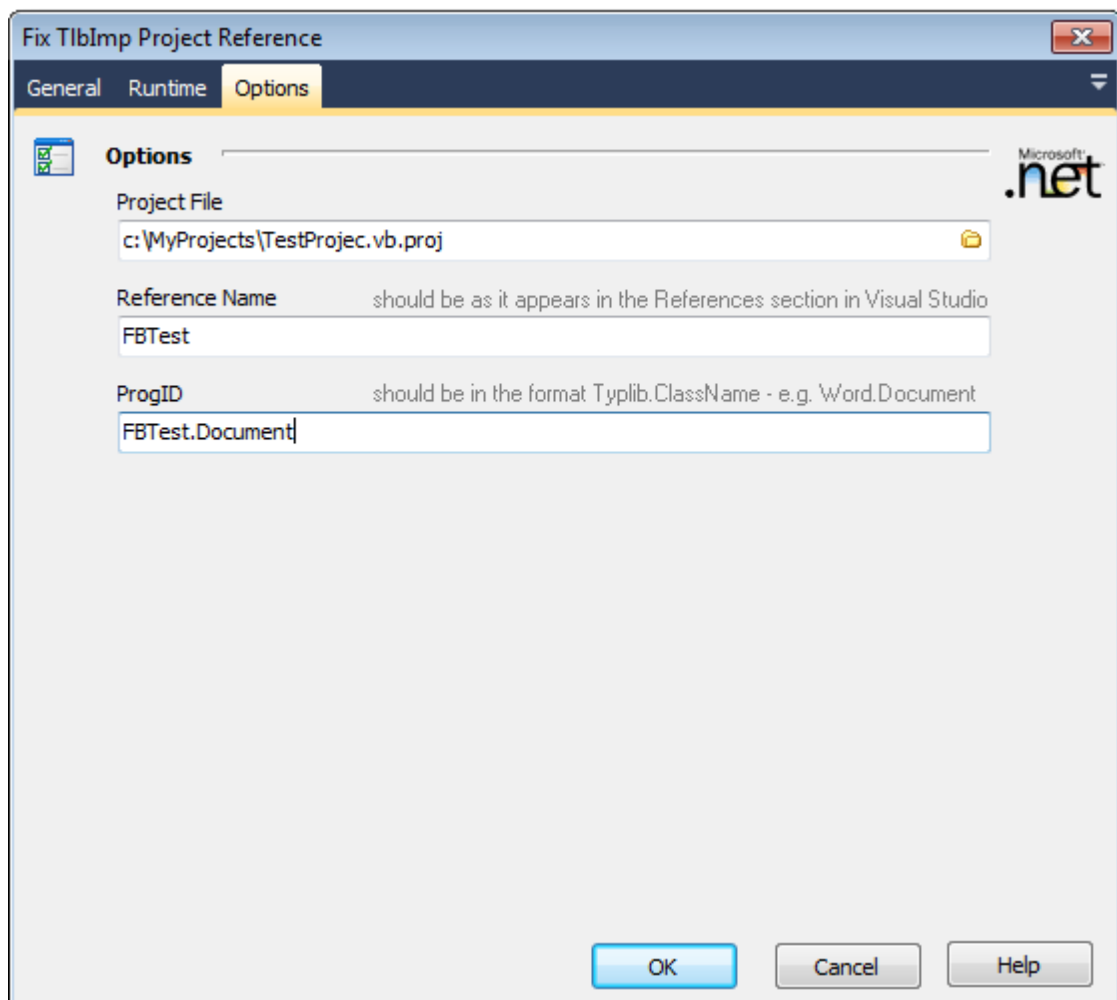
Force

This will overwrite the nuspec file if it already exists.

6.25.4 Other

6.25.4.1 Fix TLBImp Project Reference

This action will attempt to fix com interop references in a C#, VB.NET or J# Visual Studio project File. The most reason this might need to be done is if the Typelibary guids have changed since the reference was added to the project in Visual Studio. An example of when this might happen is when a .NET project is importing a VB6 COM DLL that is not compiled with binary compatibility enabled. VB6 will change the typelibary guid's every time the project is built. This action should be used after the VB6 Action and before the VS.NET action or the C#/VB.NET or J# Project compiler actions.

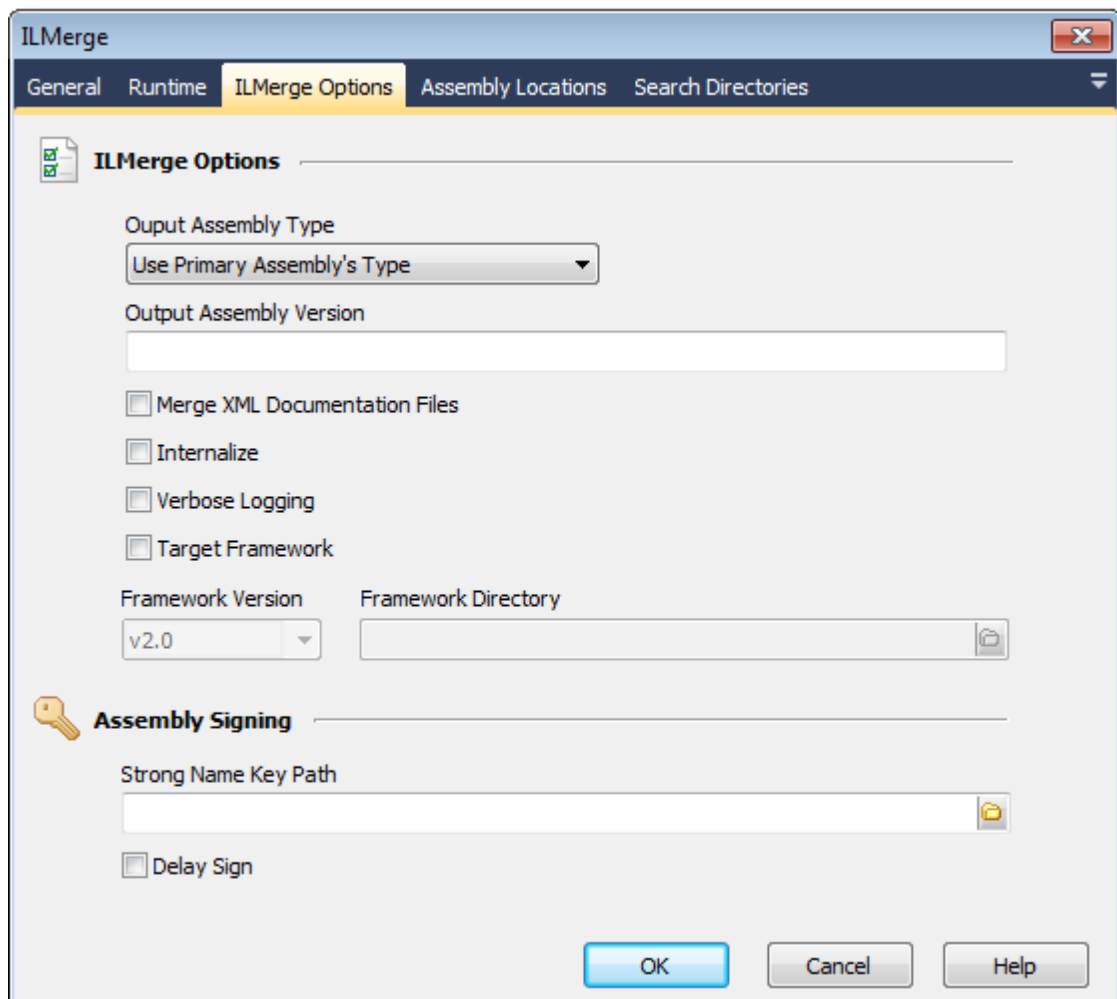


The project File property should point to the VS.NET project file (not the solution) that references the com dll. The reference name is the name of the reference as it appears in the references section of the solution explorer in Visual Studio. The ProID should be set to the format TypibName.ClassName, for example Word.Application. This prog id enables the action to find the typelibary entry in the registry and find the correct guid and version numbers for the com dll. The vs.net project file must be writeable as the action will update the reference entry in the file.

6.25.4.2 ILMerge

The ILMerge action enables you to merge multiple .Net assemblies into a single assembly.

For more information regarding ILMerge, visit <http://research.microsoft.com/~mbarnett/ILMerge.aspx>



Output Assembly Type

Controls whether the output assembly is a Windows Application, Console Application or a Library. The default setting, Use Primary Assembly's Type, will make the output assembly the same type as the primary assembly.

Output Assembly Version

Allows the output assemblies version to be specified. The value should be in standard .Net versioning notation (example: 1.0.0.0).

Merge XML Documentation Files

Controls whether the XML Documentation files will be merged together to produce a single XML Documentation file for the output assembly.

Internalize

Controls whether the non-primary assemblies will have their visibility changed to internal.

Verbose Logging

When enabled, the action will produce a full log of the merging process.

Target Framework

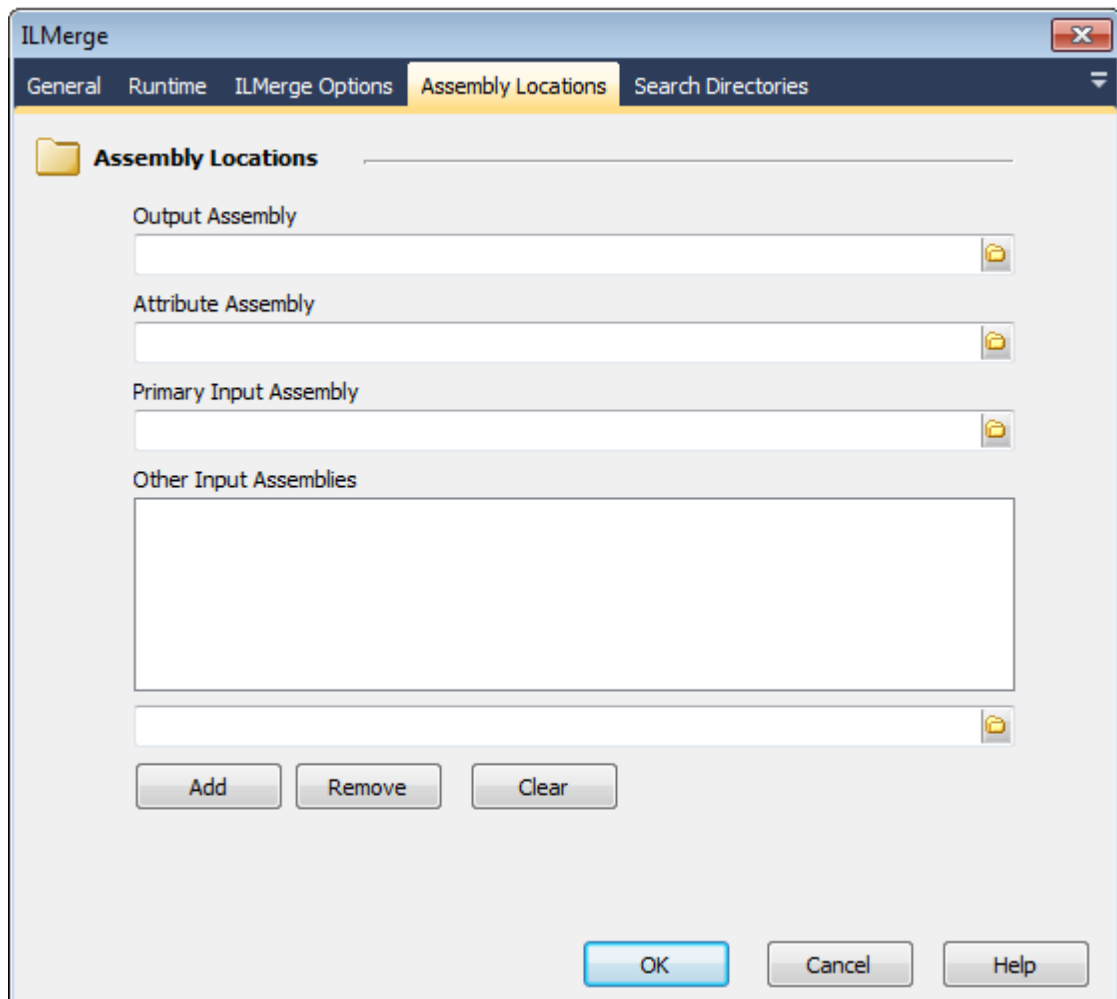
Enable this option to specify the version of the .NET Framework to use. Use the Framework Version drop down list to select the version and Framework Directory field to specify the directory of the framework to be used.

Strong Key Name Path

The path to the strong key name file to be used to sign the output assembly. If no key is specified the output assembly will not be signed.

Delay Sign

Controls whether the output assembly will be delay signed.

**Output Assembly**

The full path to the required location of the output assembly, including a filename.

Attribute Assembly

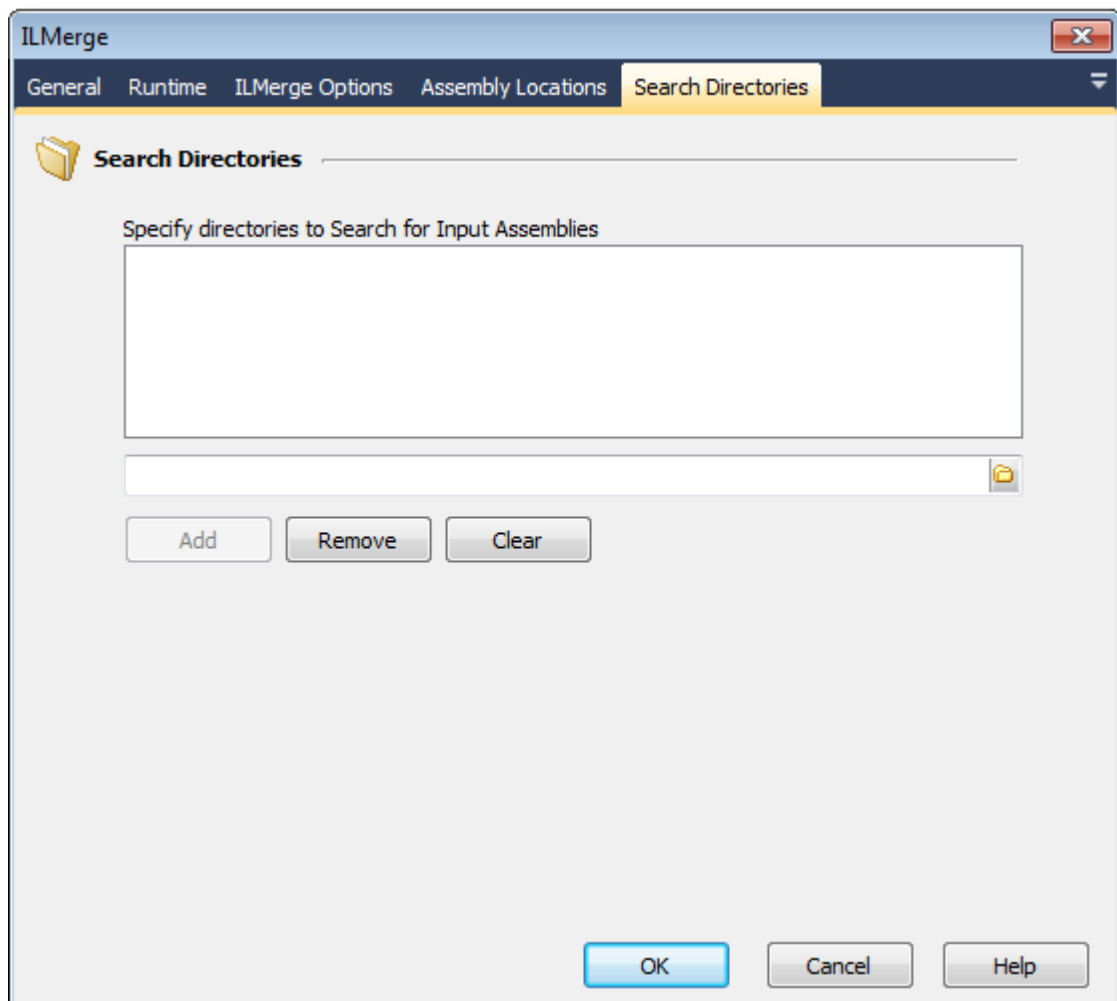
The full path to the assembly used for assembly-level attributes such as Culture, Version etc...

Primary Input Assembly

The primary assembly to use for merging. If this assembly is an executable the output assembly will be an executable with the same entry point.

Other Input Assemblies

This is a list of all the assemblies to be merged with the primary input assembly.

**Search for Input Assemblies**

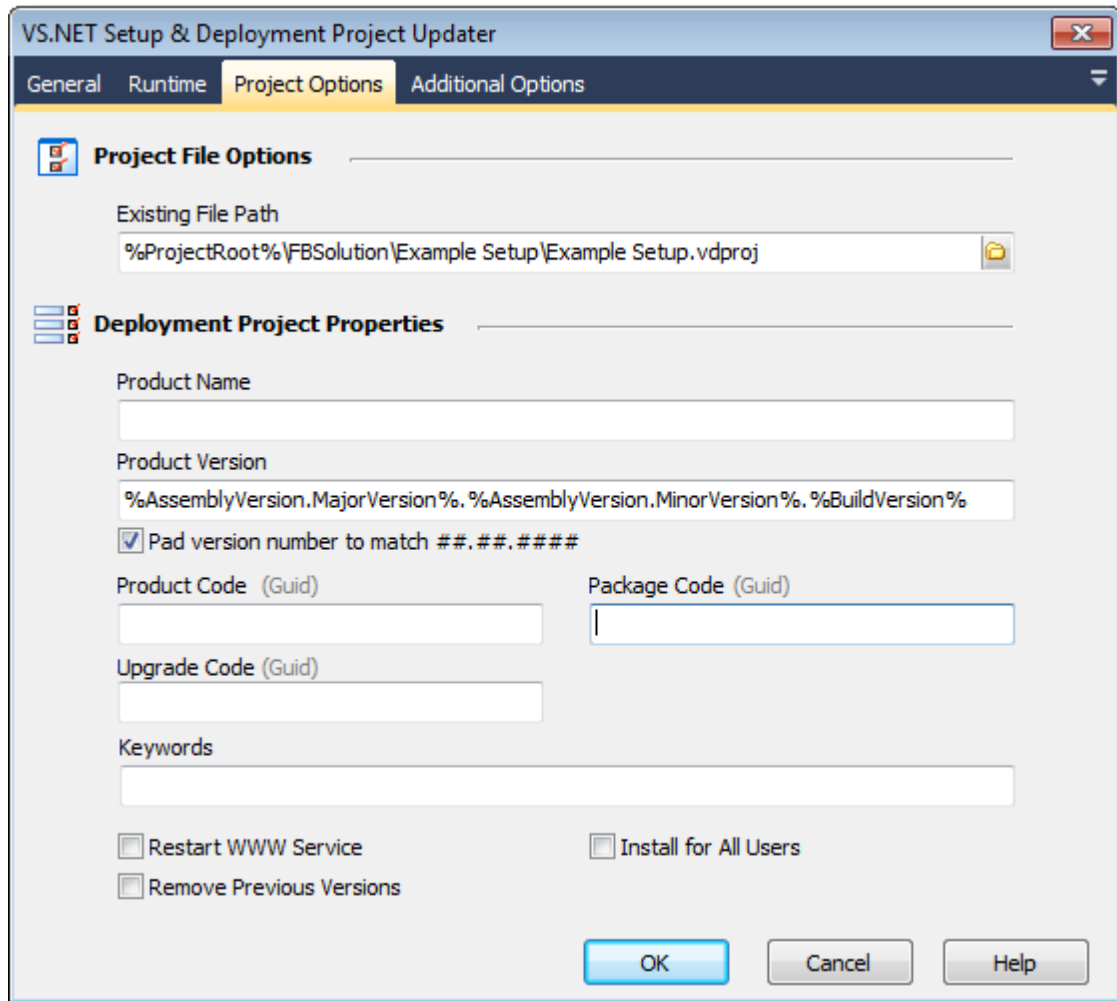
Use the controls provided to add Directories that ILMerge should search for input assemblies.

6.25.4.3 VS.NET Setup & Deployment Project Updater

The VS.NET Setup & Deployment Project Updater Action allows you to edit the metadata contained in Visual Studio Setup and Deployment project files (.vdproj)

To build .vdproj project files, use the Build VS.NET Solution action.

(Note: This action was formerly titled the Microsoft Deployment Project action.)

**Existing File Path**

The full path to the deployment project file to update.

Deployment Project Properties

All of these properties are available to update. Any field left blank will not be updated. In the example shown above, only the Product Version will be updated.

Product Name

The name of the product. This value will be used within the installer to make reference to your product.

Product Version

The version of the product. The version number format must be ##.##.####; for example 1.0.453

Pad version number to match

If you check the box then the action will try to modify the version number so it can be

used. For example, "1.0.3" will become "01.00.003".

Product Code

A Guid used to identify a product release. Windows Installer requires that each installed product has a unique ID.

Package Code

A Guid used to identify the MSI package.

Restart WWW Service

Check this box if the built installer needs to restart the WWW Service after the install runs.

Remove Previous Versions

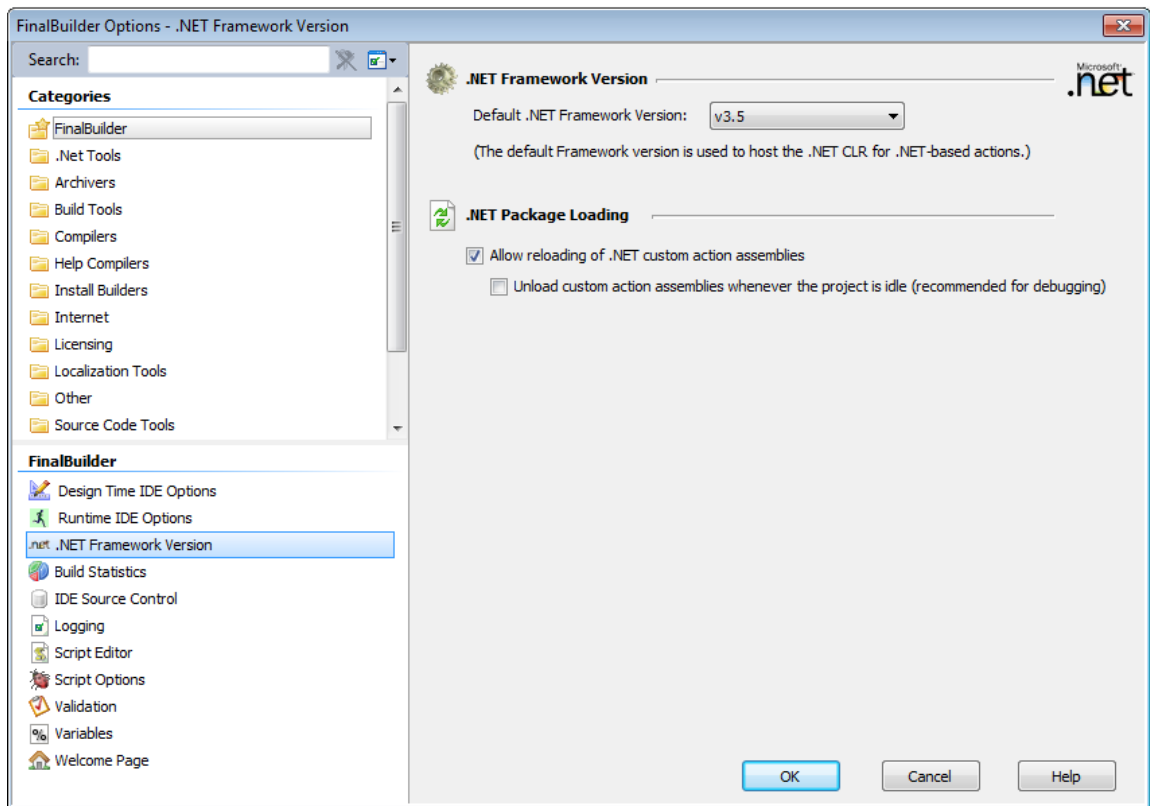
Check this box to remove any previous versions when the built product is installed.

Install for All Users

If this box is checked, the installer show Install for All Users as the default choice. Otherwise, Install for Current User is the default choice.

6.25.5 .NET Framework Options

The .NET Framework Version options control settings used by .NET actions.



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 FinalBuilder 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.

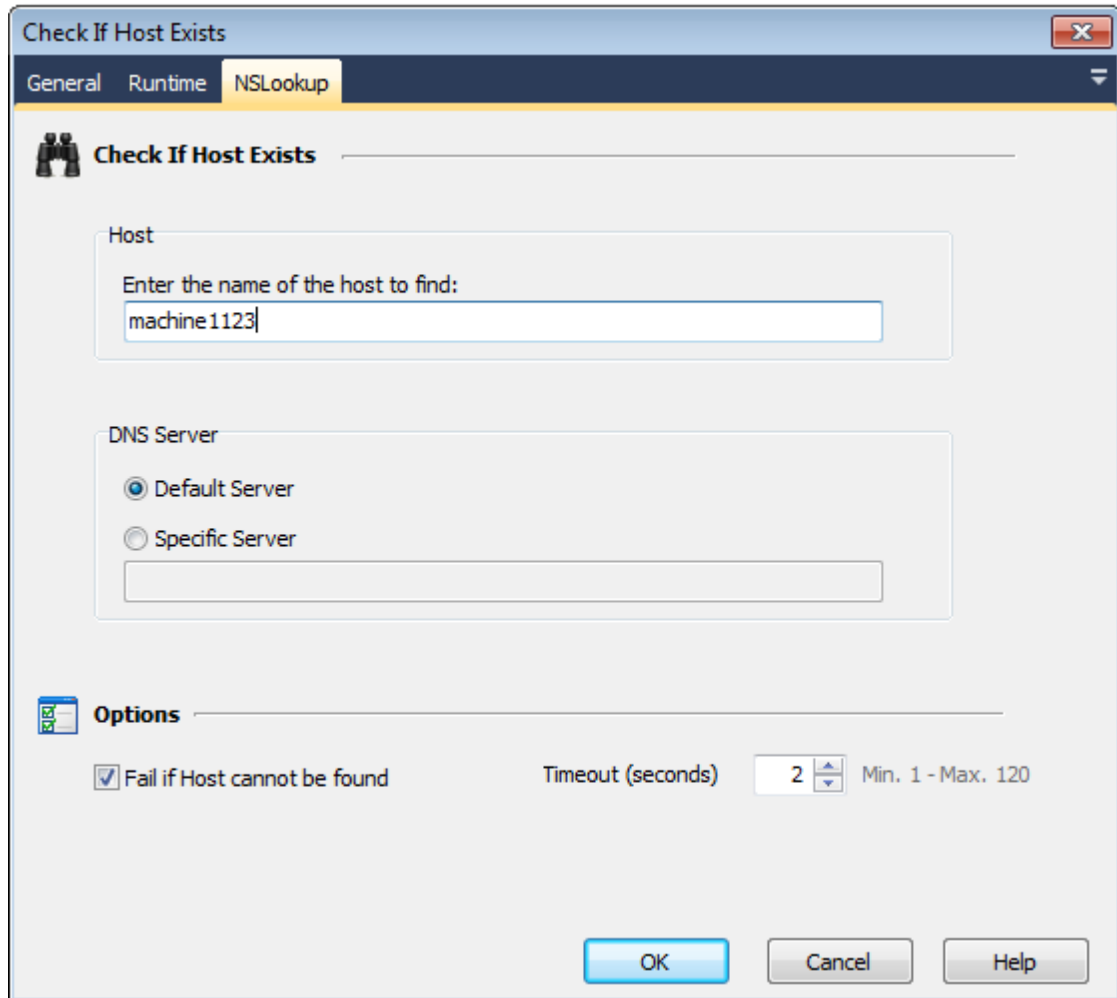
Unload custom action assemblies whenever the project is idle

Checking this option means that FinalBuilder will unload custom action assemblies when FinalBuilder is idle. For example, after a build have completed or when the action properties dialog is closed.

6.26 Network

6.26.1 Check If Host Exists

The Check If Host Exists action allows you to query a DNS server to determine if a host exists.



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).

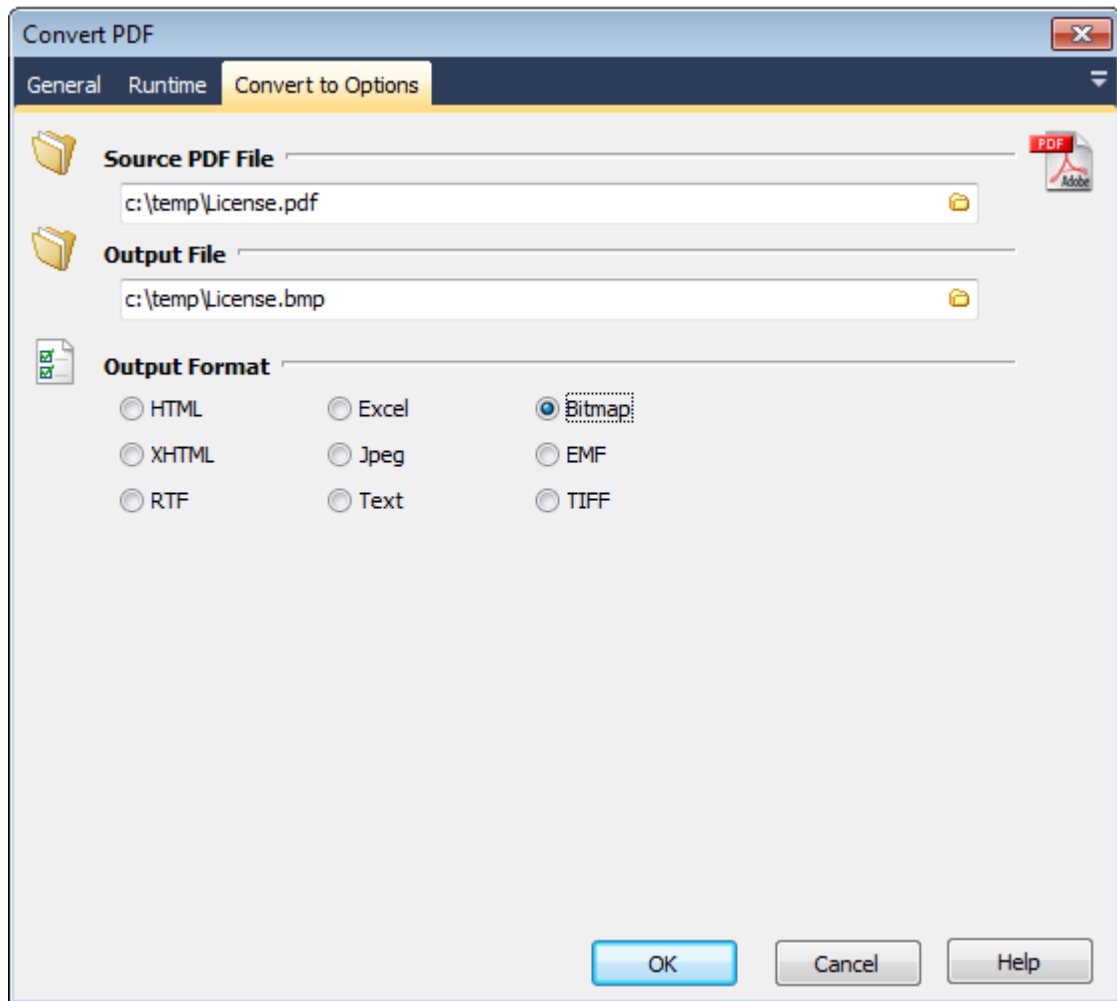
- Timeout - Specify the timeout value at which point the query will be abandoned if the host has not been found.

6.27 PDF Files

6.27.1 Convert PDF

[FinalBuilder 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 - specify the source PDF file to convert

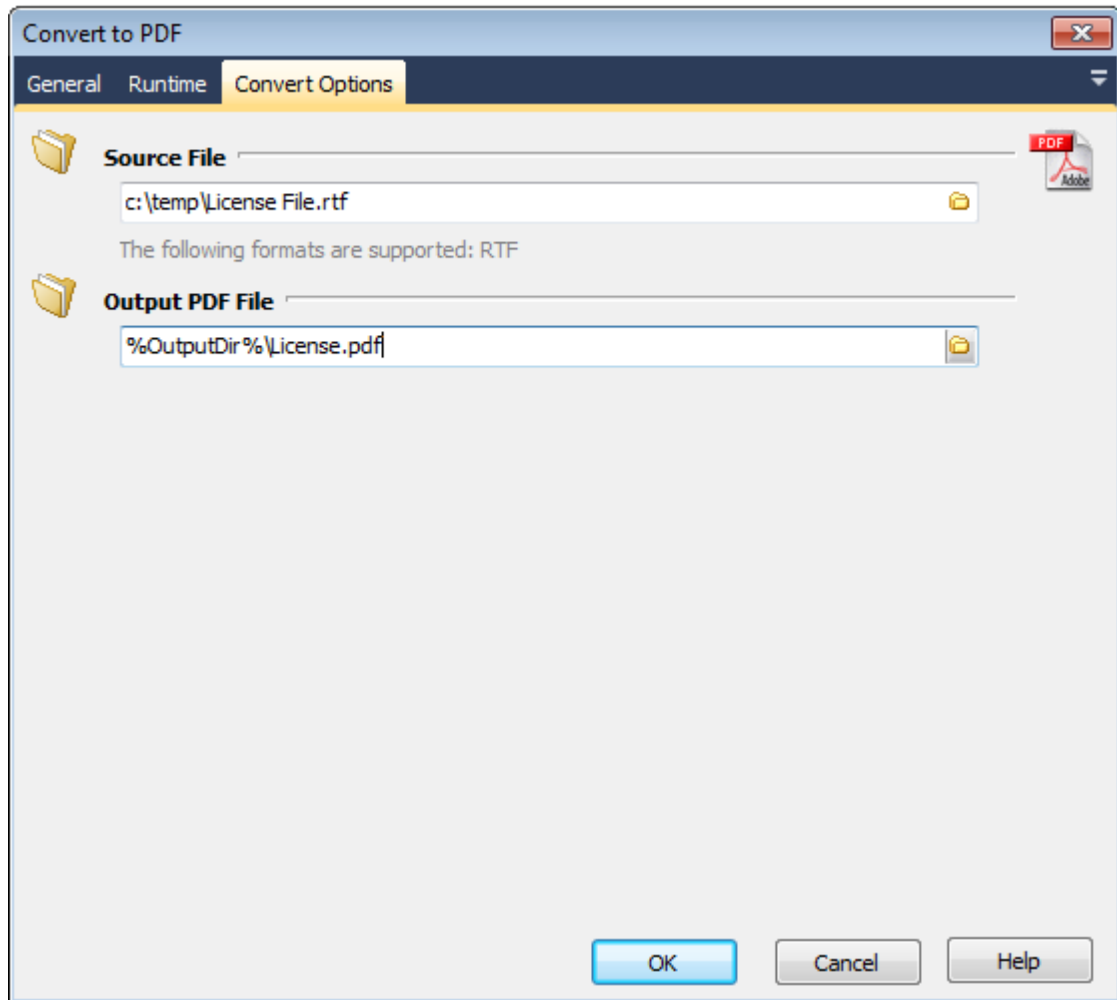
Output File - Specify the output filename

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.

6.27.2 Convert to PDF

[FinalBuilder Professional Edition]

The Convert to PDF action will convert an RTF (Rich Text Format) action into a PDF File.



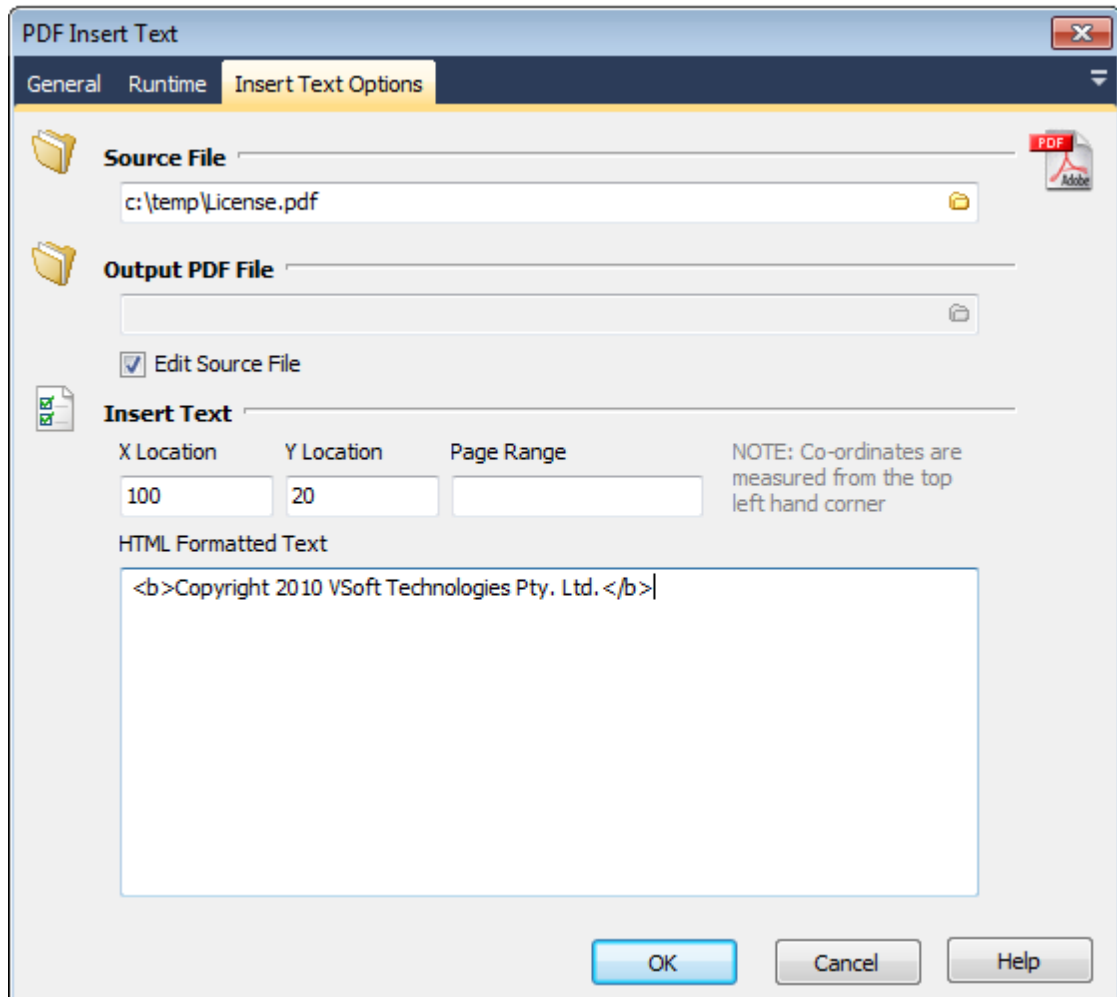
Source File - specify the RTF file to convert to PDF

Output PDF File - specify the PDF filename to write to

6.27.3 PDF Insert Text

[FinalBuilder Professional Edition]

The PDF Insert Text action will add HTML formatted text to a PDF file at the specified X, Y location.



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 **....**

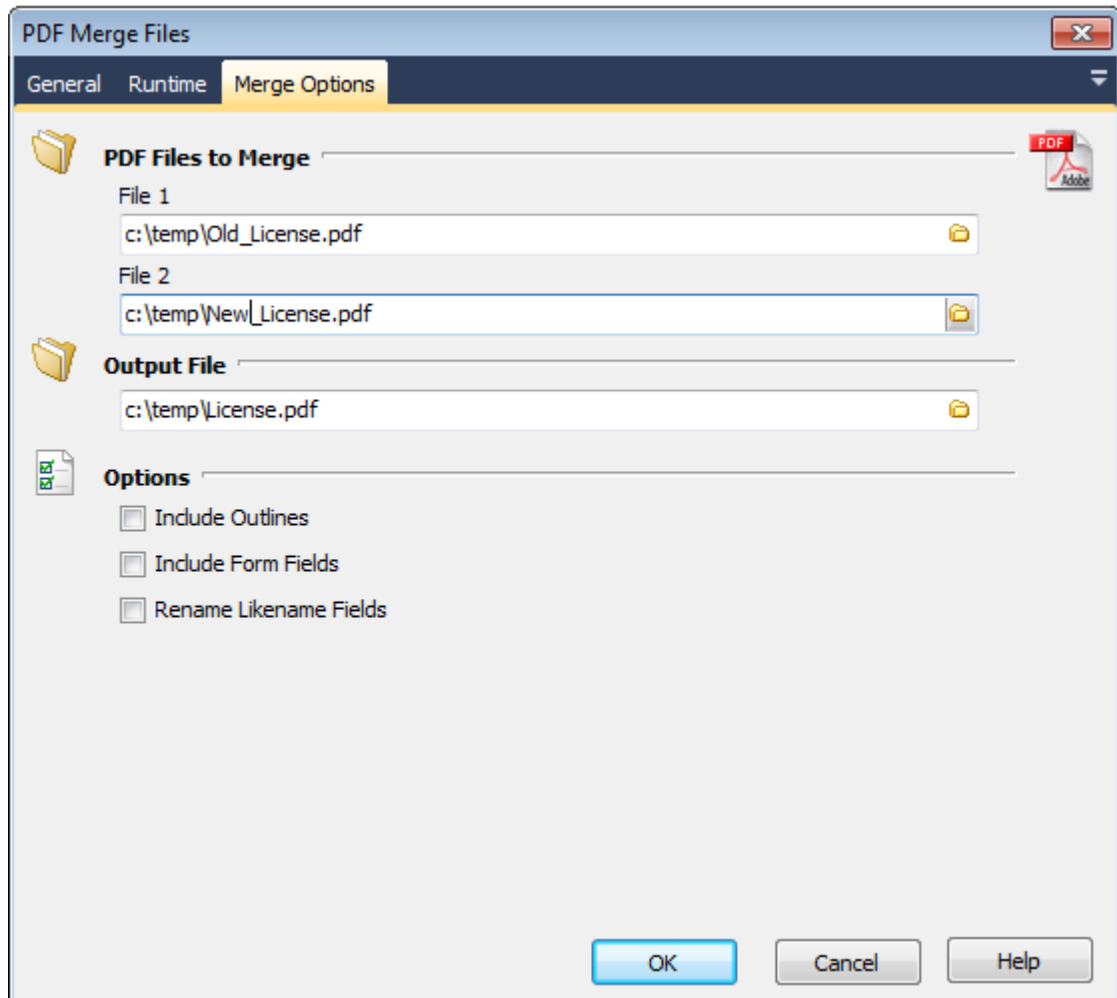
Italic *<i>....</i>*

Underlined <u>....</u>

6.27.4 PDF Merge Files

[FinalBuilder Professional Edition]

The PDF Merge Files action will merge two PDF files into a new single PDF File.



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.

6.28 Property Sets

FinalBuilder Property Sets are a group of values which belong together. A common use of property sets is to use them for the version information for compilers as the version information consists of a set of integer values, eg. Major, Minor, Release, Build. In most cases these four values need to be treated as one, eg. Load from INI, apply to compiler, use them to generate a filename, Save to INI. But in other cases a single property of the property set might be singled out (eg. to increment the Build number).

There are a set of actions which deal with Property sets:

- PropertySet define
- PropertySet load from INI file
- PropertySet load from Registry
- PropertySet load from FBD
- PropertySet save to INI file
- PropertySet save to Registry
- PropertySet save to FBD
- PropertySet increment value
- PropertySet log values
- PropertySet assign properties
- PropertySet to Variable

In all cases the property set needs to be defined, so use a PropertySet define action at the start of your build. Once the PropertySet define action has been added to your build, the other PropertySet actions can access the named property set.

A common practise would be to load a property set at the beginning of the build, possibly increment a property, use the property set in various actions, and then at the end of the build save the property set if the build was successfully. Property sets can be saved and loaded from INI files, the system Registry, or the FBD file. The values of the Property Set can also be set using hard coded values or FinalBuilder variables by using the PropertySet assign properties action. A good way to record the values of the property set is to use the PropertySet log values action, which outputs the current values of the property set to the FinalBuilder log.

To build up a string value using the different properties of a property set, use the PropertySet to Variable action. This action allows you to define a prefix, suffix, separator and any of the properties in any order to build up a value which is saved in a FinalBuilder variable.

The following PropertySets are always available:

- Win32 Version Numbers
- Win32 Version Info
- .Net Assembly Numbers
- .Net Assembly Info

Some PropertySet types are registered by certain packages:

VisualBasic Package

- VisualBasic Version Info
- VisualBasic Version Numbers

For detailed information on the values of the property sets, see Property Set Definitions

PropertySets can also be accessed in Script and any text field of an action that also allows FinalBuilder variables. See [Accessing PropertySets](#) for more information.

6.28.1 Property Set Definitions

The following PropertySets are always available:

- Win32 Version Numbers
- Win32 Version Info
- .Net Assembly Numbers
- .Net Assembly Info

The following PropertySet types are only available if the relevant package is used:

VisualBasic Package

- VisualBasic Version Info
- VisualBasic Version Numbers

6.28.1.1 Win32 Property Sets

This property set is designed for compilers targeting the Windows32 platform (eg. Delphi, VisualStudio 6). VisualBasic cannot use these property sets because it uses a different numbering scheme - see here.

There are two Win32 specific property sets:

Win32 Version Numbers

This property set defines the following properties:

- Major Version (Integer)
- Minor Version (Integer)
- Release Version (Integer)
- Build Version (Integer)

Win32 Version Info

This property set defines the following properties:

- Major Version (Integer)
- Minor Version (Integer)
- Release Version (Integer)
- Build Version (Integer)
- CompanyName (string)
- FileDescription (string)
- FileVersion (string)
- InternalName (string)
- LegalCopyright (string)
- LegalTrademarks (string)
- OriginalFilename (string)
- ProductName (string)
- ProductVersion (string)
- CompileDate (string)
- DebugBuild (boolean)
- SpecialBuild (boolean)
- PrivateBuild (boolean)
- PreReleaseBuild (boolean)
- DLL (boolean)

6.28.1.2 .Net Property Sets

This property set is designed for compilers targeting the Microsoft Dot Net platform (eg. VisualStudio.Net).

There are two Dot Net specific property sets:

Dot Net Assembly Numbers

This property set defines the following properties:

- Major Version (Integer)
- Minor Version (Integer)
- Build Version (Integer)
- Revision Version (Integer)

Dot Net Assembly Info

This property set defines the following properties:

- Major Version (Integer)
- Minor Version (Integer)
- Build Version (Integer)
- Revision Version (Integer)
- Title (string)
- Description (string)
- Company (string)
- Product (string)
- Copyright (string)
- Trademark (string)

6.28.1.3 VB Property Sets

This property set is specifically designed for VisualBasic. The VB specific property sets are only available if the FinalBuilder VisualBasic package is loaded.

There are two VB specific property sets:

VisualBasic Version Numbers

This property set defines the following properties:

- MajorVersion (integer)
- MinorVersion (integer)
- RevisionVersion (integer)

VisualBasic Version Info

This property set defines the following properties:

- MajorVersion (integer)
- MinorVersion (integer)
- RevisionVersion (integer)
- Comments (string)
- CompanyName (string)
- FileDescription (string)
- LegalCopyright (string)

- LegalTradeMarks (string)
- ProductName (string)

6.28.2 Accessing PropertySets

The values of a PropertySet can be accessed in Script Events and also in text fields of actions that support FinalBuilder variables.

Accessing PropertySet values in Script Events

The syntax to access a PropertySet value in script is:

```
<PropertySetName>.<PropertySetValue>
```

eg. (VBScript)

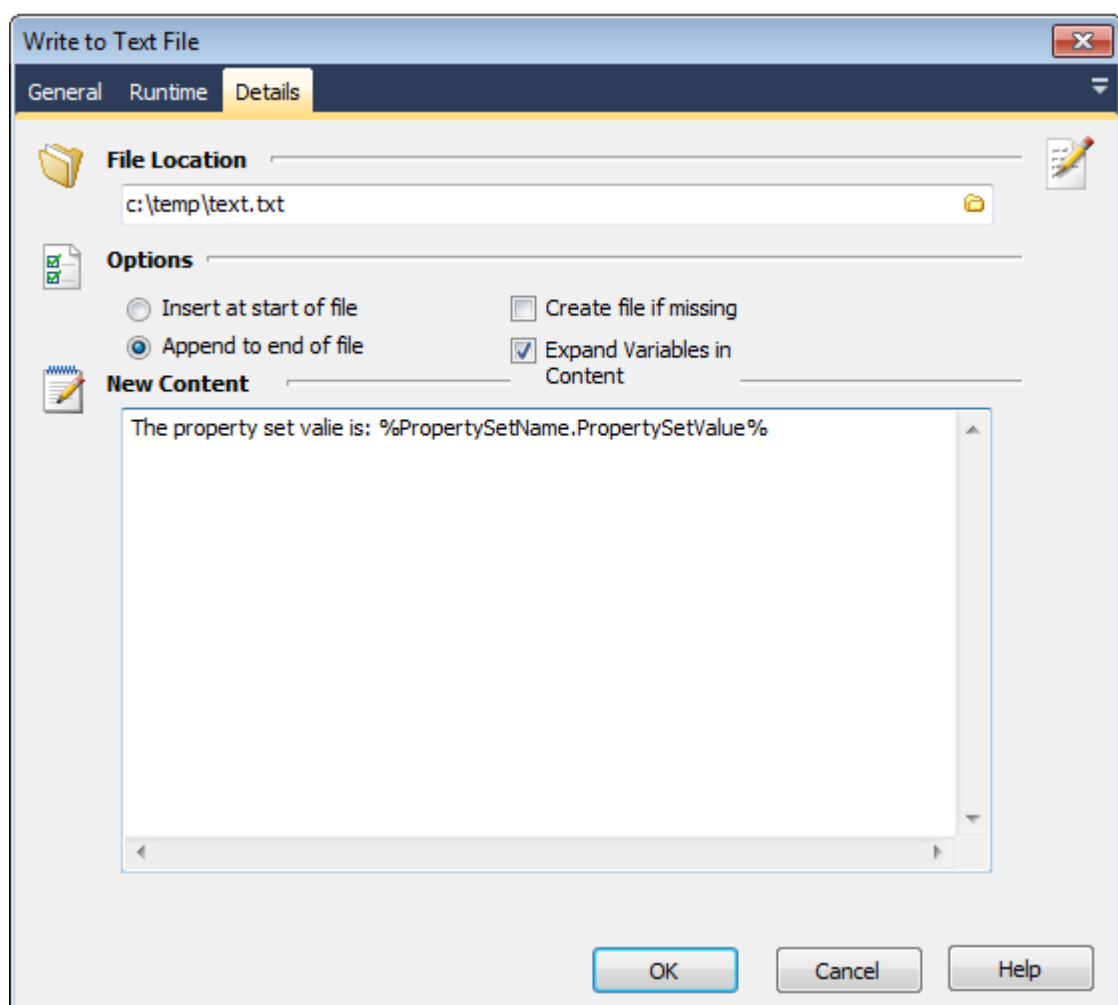
```
msgbox MyVersionNumbers.MajorVersion
```

Accessing PropertySet values in Action fields

The syntax to access a PropertySet value in an action field is:

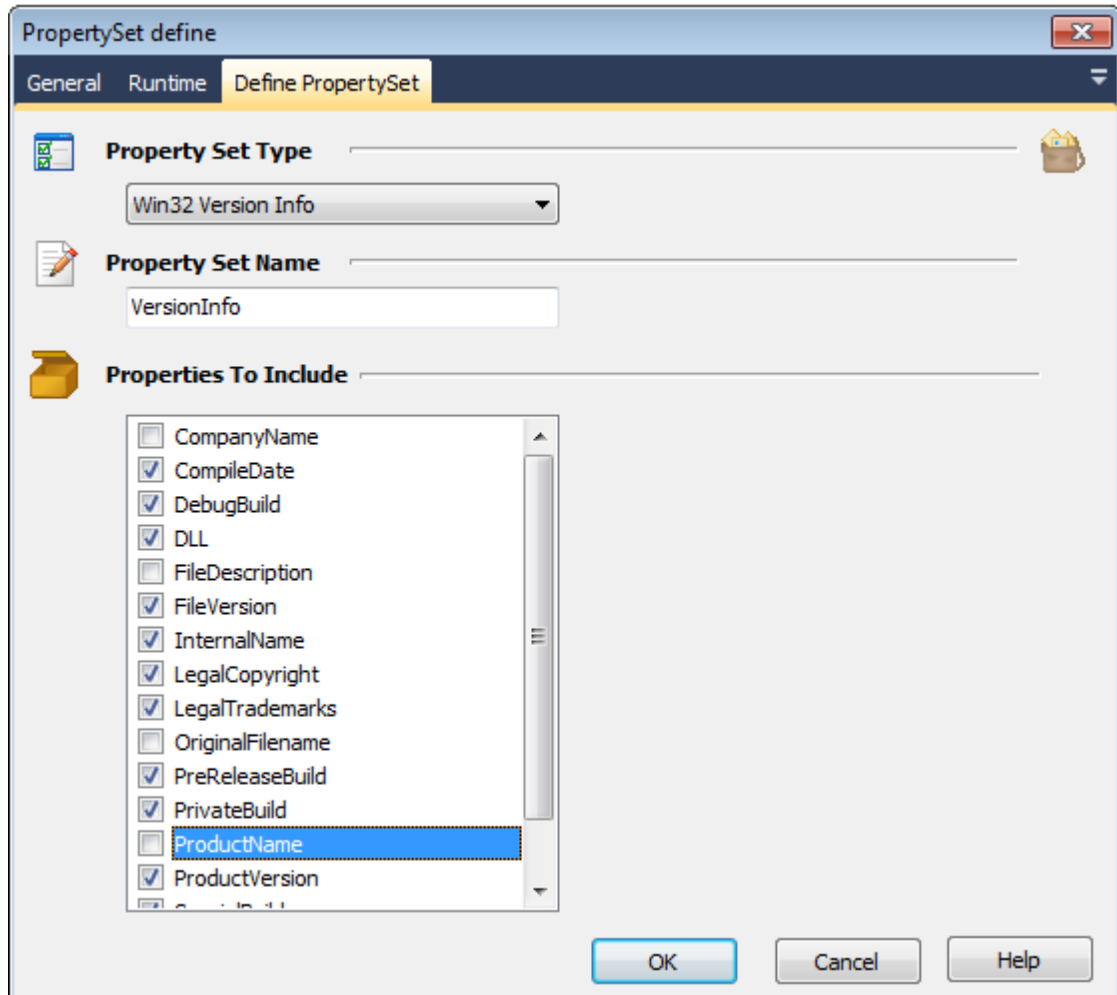
```
%<PropertySetName>.<PropertySetValue>%
```

The example below shows how to access the BuildVersion property in an action:



6.28.3 PropertySet define

Use the PropertySet define action to create a new property set for your build.



Property Set Type

The Property Set Type lists all available property sets. Property sets are registered with FinalBuilder by certain action packages. See Property Sets for the list of registered Property sets.

Property Set Name

Give your property set a name, the name must not contain blanks.

As soon as the PropertySet action has been added to your project and given a name, the PropertySet is available to other actions in your project (eg. in the Delphi action to set the version information).

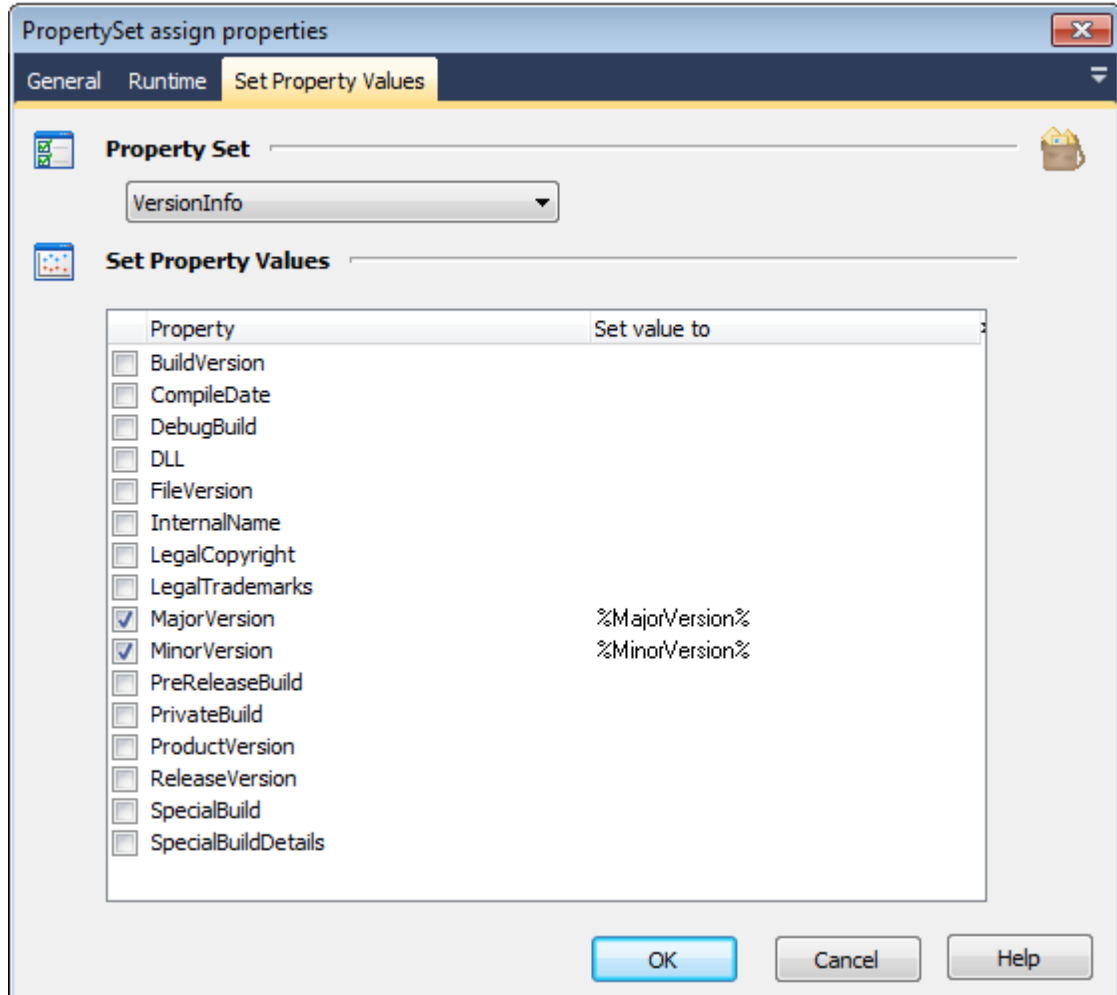
NOTE: If you change the name of a property set, you'll need to update any actions in your project which reference the PropertySet, as the reference is done via the PropertySet's name.

Properties To Include

Each property set type contains a list of properties. By default, all of these properties are included in the set. It is also possible to "trim down" a PropertySet by excluded some properties. In the screenshot above, the "CompanyName", "OriginalFileName" and "ProductName" are excluded from the set. This allows you to "mix and match" properties, setting some from the PropertySet but leaving others with their default values.

6.28.4 PropertySet assign properties

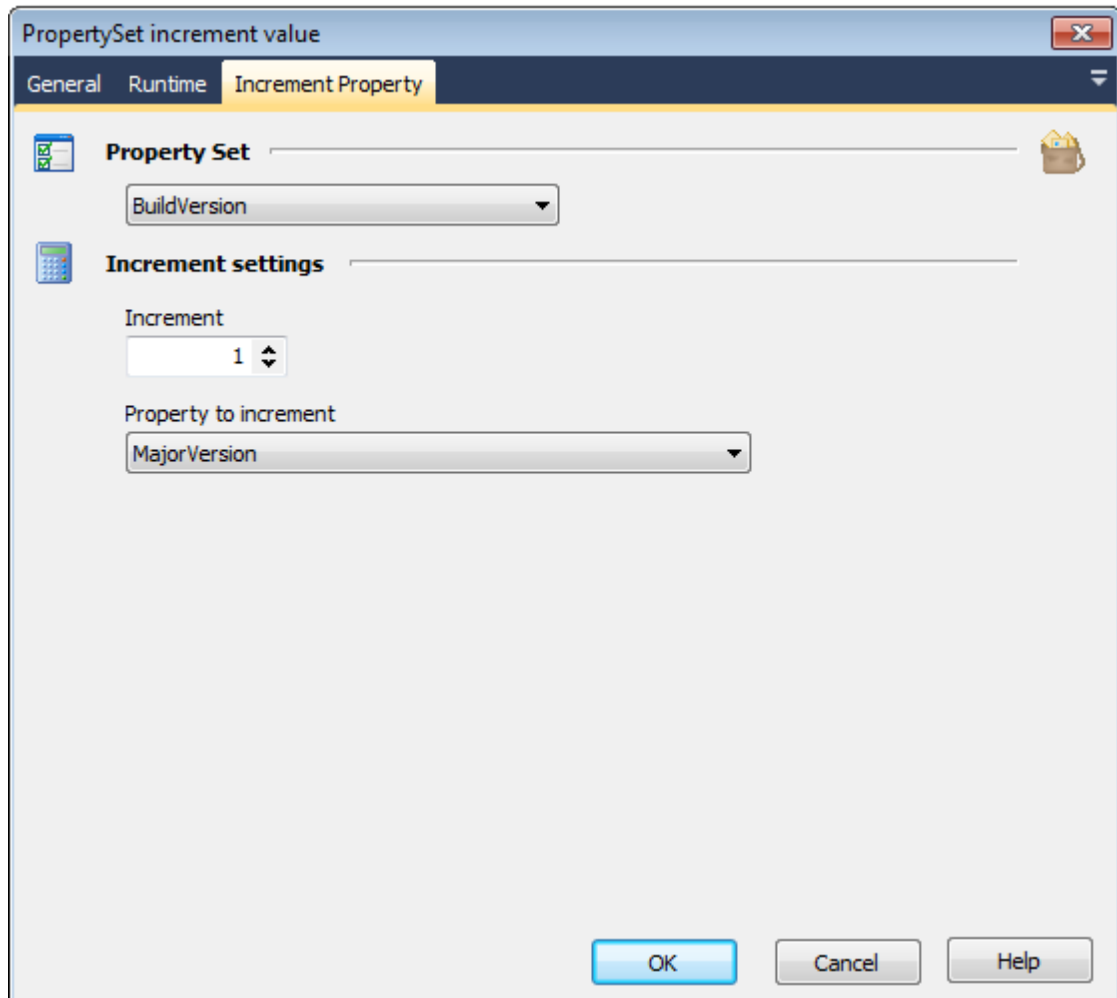
The PropertySet assign properties action allows each property of the Property Set to be assigned to either a particular value or to the value of a FinalBuilder variable.



The checkbox denotes which properties will be assigned to new values when the action executes. Enter the new value to set the properties to in the "Set value to" column. The expression will be evaluated, therefore allowing FinalBuilder variables.

6.28.5 PropertySet increment value

The PropertySet increment value action allows you to increment or de-increment an integer property of a property set.



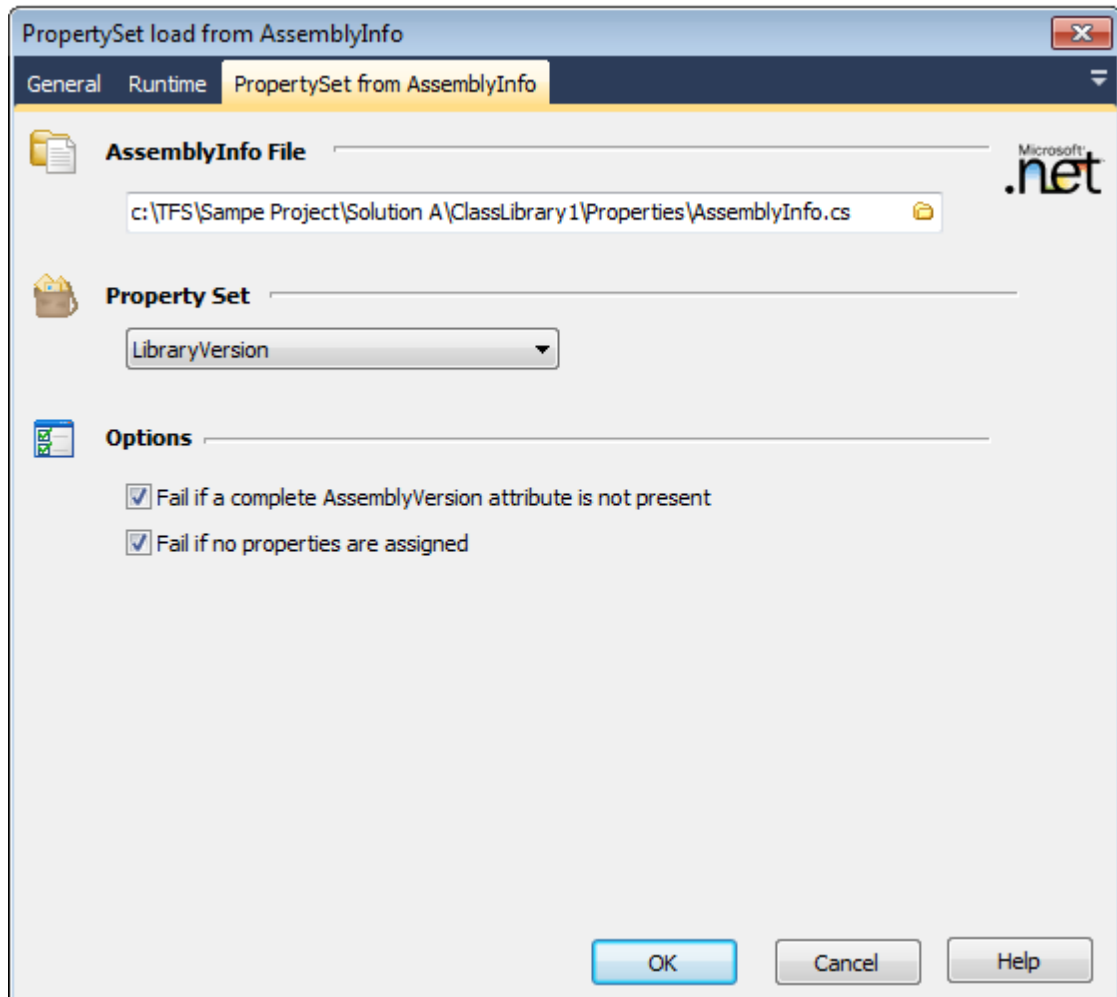
Property Set - select the property set which contains the property you wish to increment.

Increment - Select a positive or negative value which will be added to the integer property.

Property to increment - Select the property to increment. If the property is not an integer type the action will fail

6.28.6 PropertySet load from AssemblyInfo

The "PropertySet load from AssemblyInfo" action allows you to load the values of .NET property set from a .NET AssemblyInfo file.



AssemblyInfo File

The path to the AssemblyInfo file to load details from. Supported file formats include C#, VB.Net, Managed C++, J# and Delphi.NET.

Property Set

The name of the PropertySet to load the values into. Property sets of type ".NET Version Information" and ".NET Version Numbers" are supported. When using ".NET Version Information", both the AssemblyVersion numbers and other relevant metadata will be loaded from the AssemblyInfo file. For ".NET Version Numbers", only the AssemblyVersion numbers will be loaded.

The PropertySet must already be defined in the project, by using a PropertySet Define action.

Options

"Fail if a complete AssemblyVersion attribute is not present"

If this option is enabled, the action will fail if it does not find a complete AssemblyVersion attribute in the AssemblyInfo file. A complete attribute contains a value of the form "a.b.c.d". Values of the form "a.b.c.*" and "a.b.*" do not count as complete.

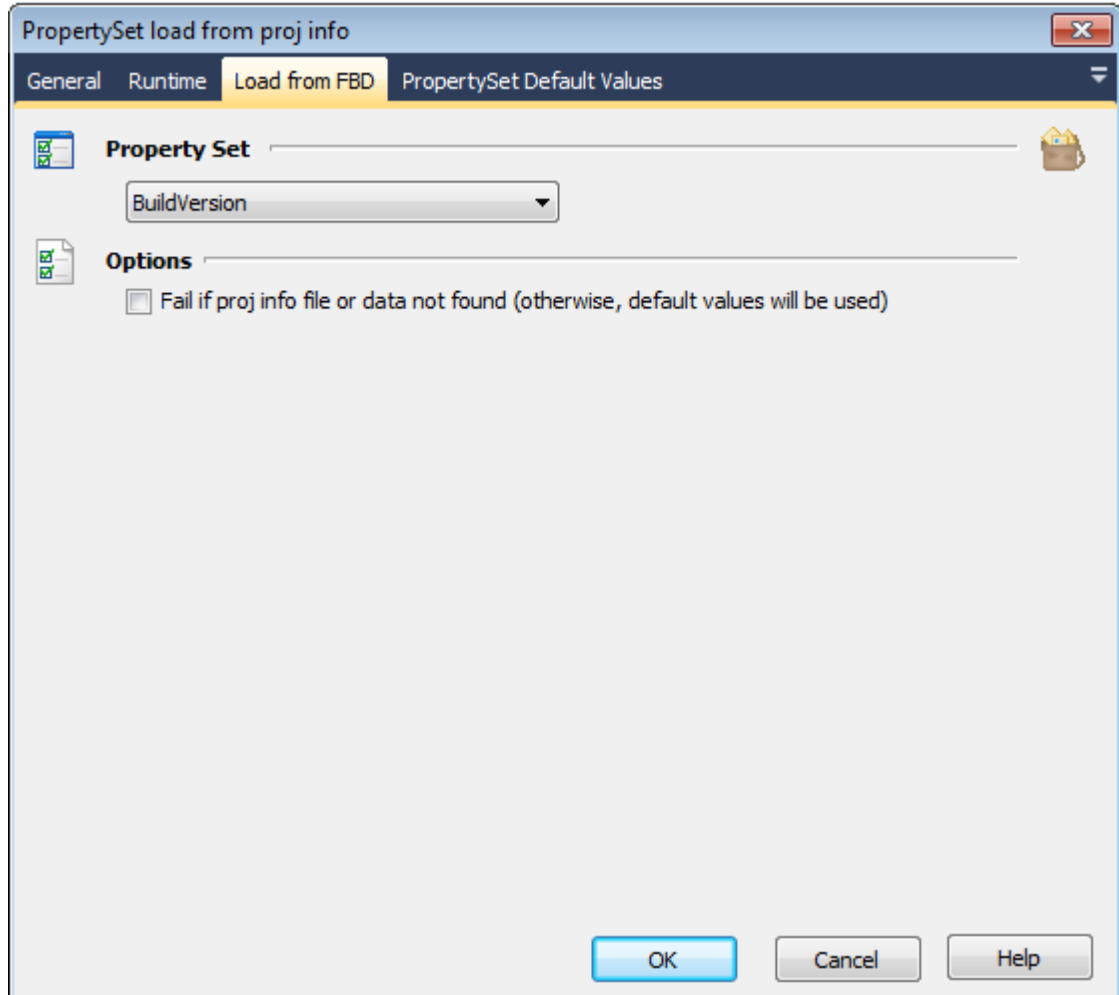
If this option is disabled, the action will assign only those version number parts that are present in the AssemblyVersion attribute. For instance, "5.4.*" will assign MajorVersion to 5 and MinorVersion to 4, but will not assign BuildVersion or RevisionVersion.

"Fail if no properties are assigned"

If this option is enabled, the action will fail if none of the properties of the PropertySet are assigned by the action.

6.28.7 PropertySet load from FBD

The PropertySet load from FBD action will load the values of your property set from your project's FBD file.

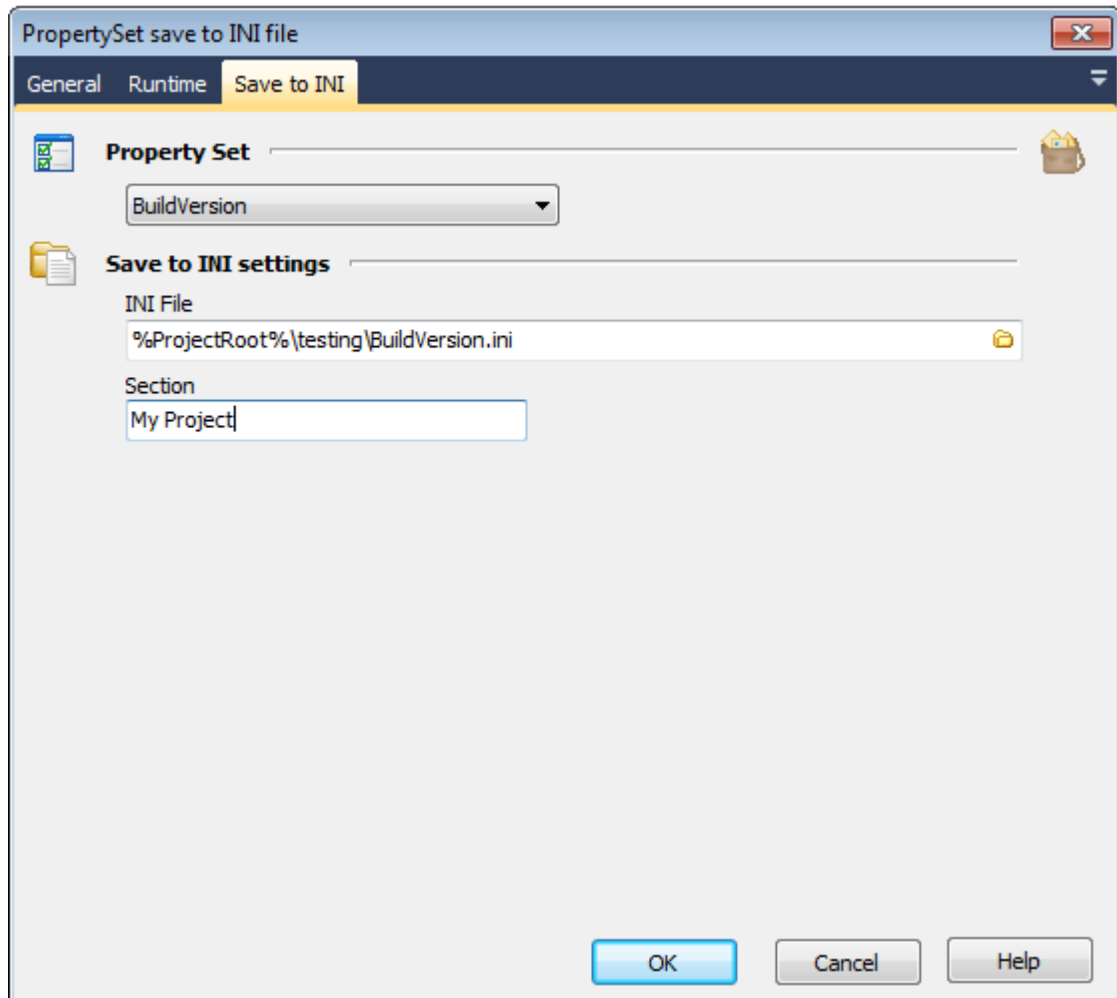


For more information on the FBD file, see [Project Files & Other Files](#).

The action will fail if the FBD file doesn't exist.

6.28.8 PropertySet load from INI file

The PropertySet load from INI file action will read the values of your property set from an INI file.



Property Set - select the property set which you want to load.

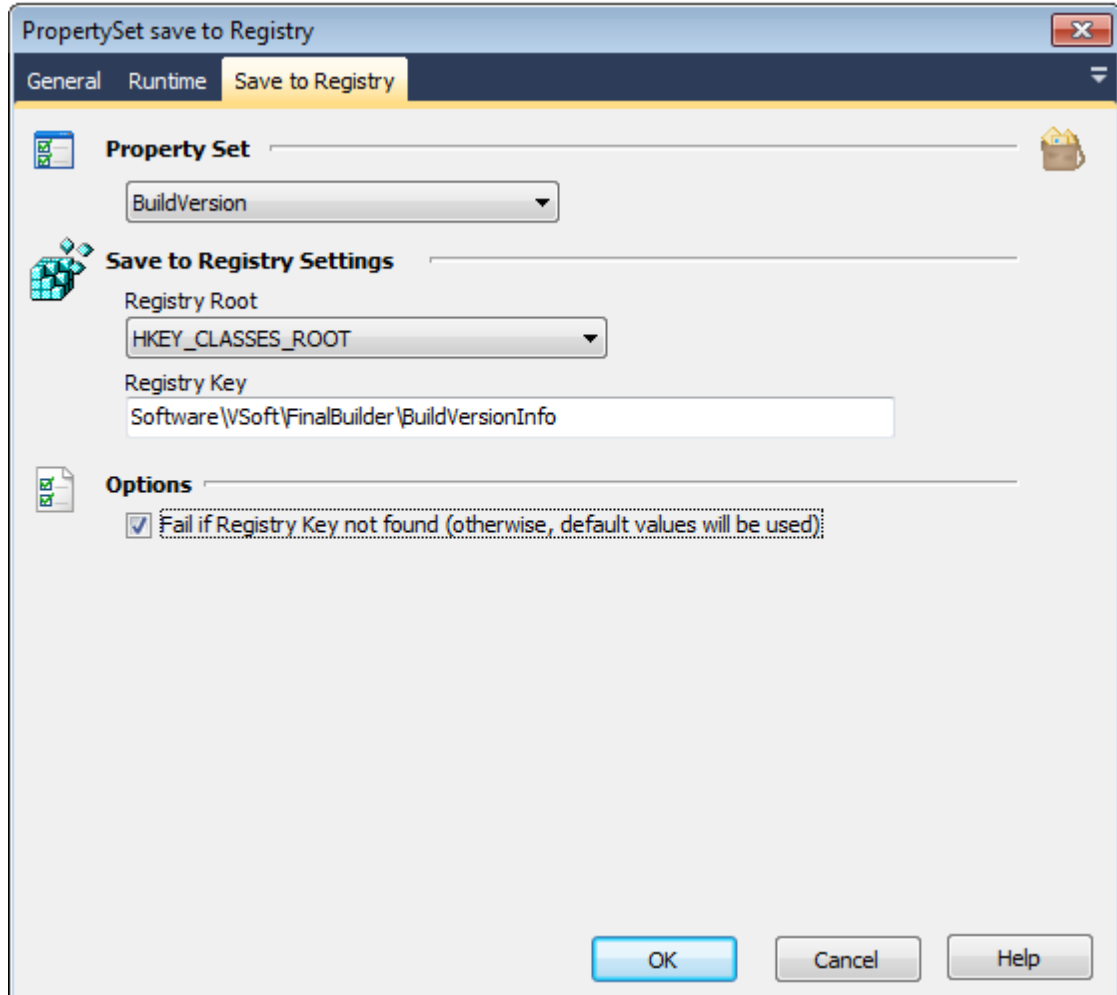
INI File - The INI file to load the PropertySet from.

Section - The INI file section in which to load the PropertySet values

The action will fail if the INI file doesn't exist or if the Section doesn't exist in the INI file.

6.28.9 PropertySet load from Registry

The PropertySet load from Registry action will load the values of your property from the system registry.



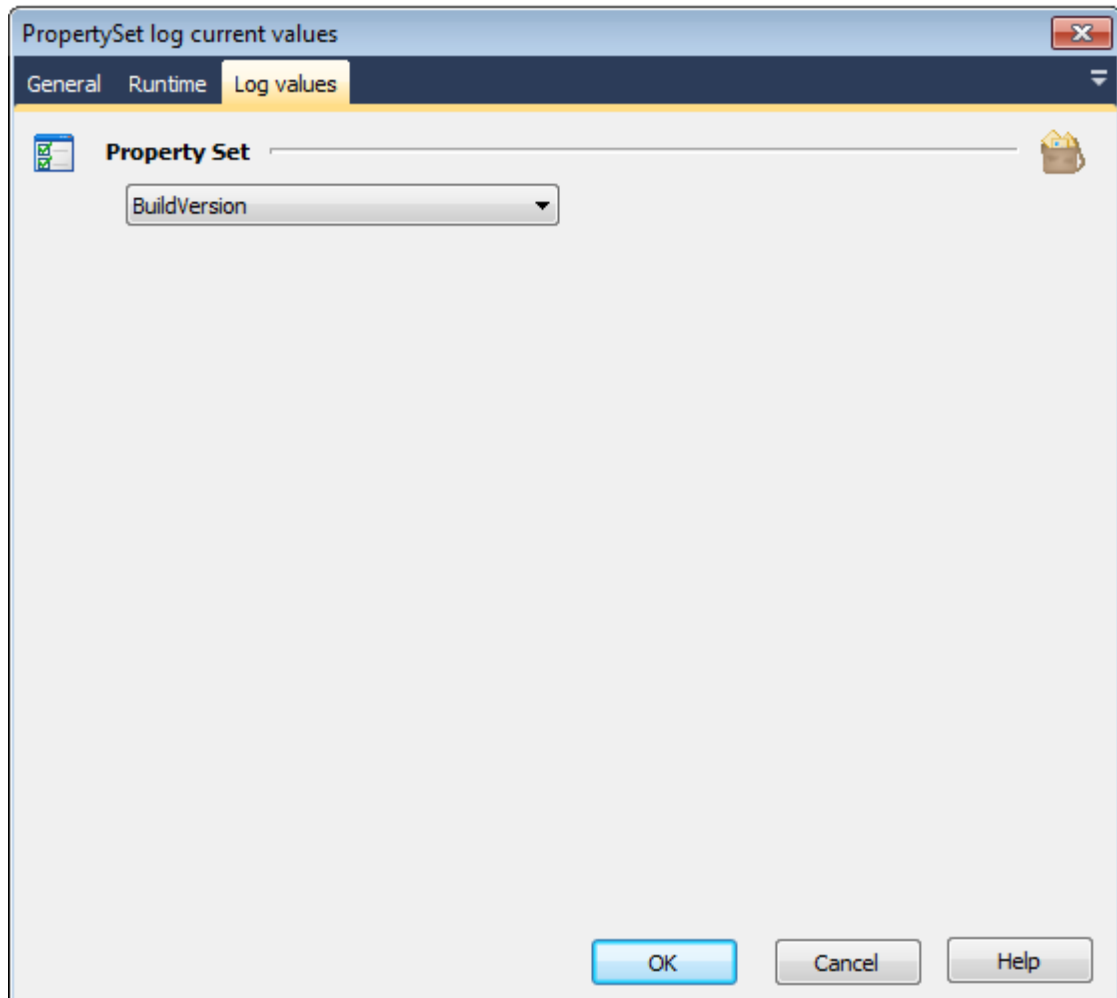
Property Set - select the property set which you want to load.

Registry Root - Select the Registry Root, usually Current_User or Local_Machine

Registry Key - Enter the registry key where you want the PropertySet values to be loaded from. If the key does not exists, then the action will fail.

6.28.10 PropertySet log values

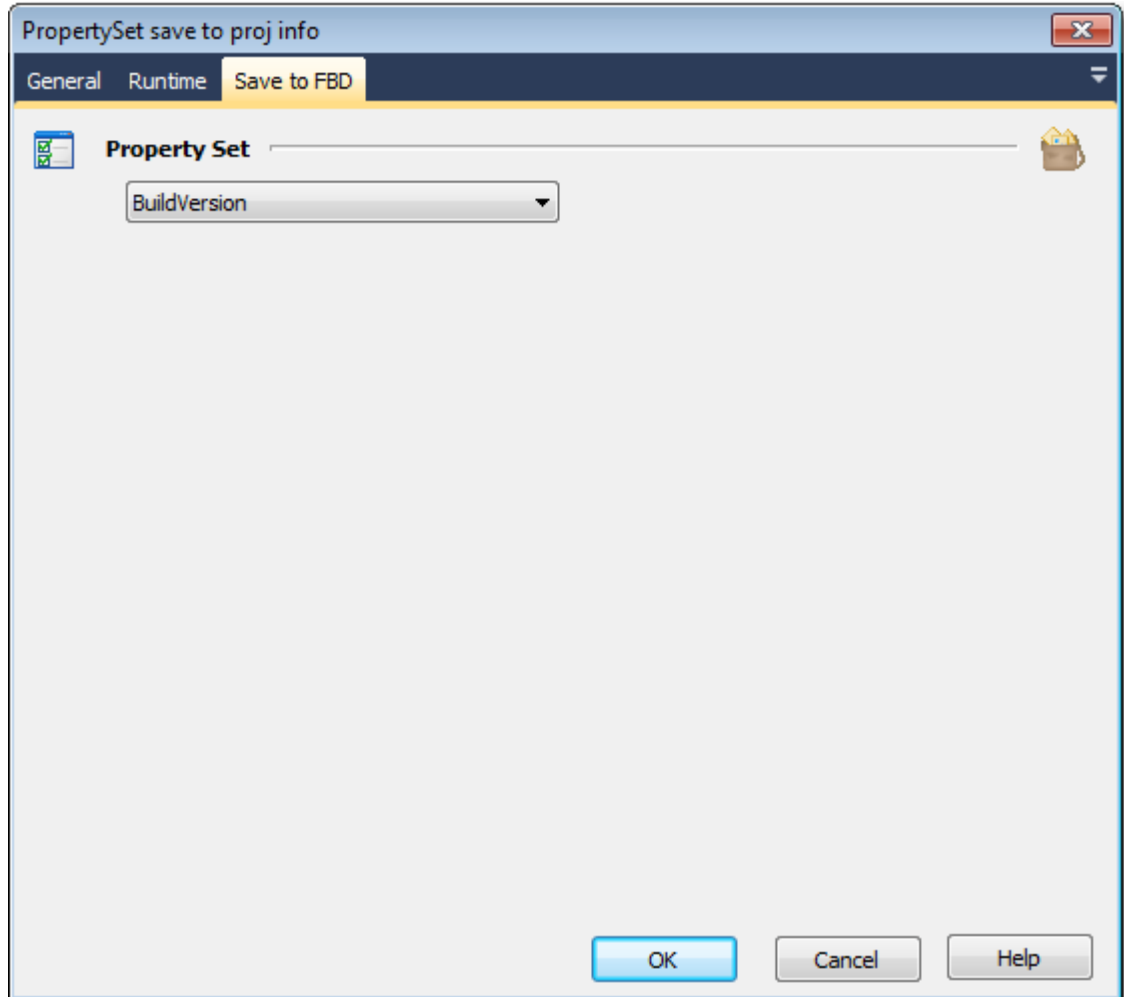
The PropertySet log values action outputs the current values of the property set to the FinalBuilder log.



Simply select the Property Set, and during your build process the action will log each property name along with it's current value. This is useful during debugging, as well as to record the values of the property set in your build log.

6.28.11 PropertySet save to FBD

The PropertySet save to FBD action will persist the values of your property set to your project's FBD file.

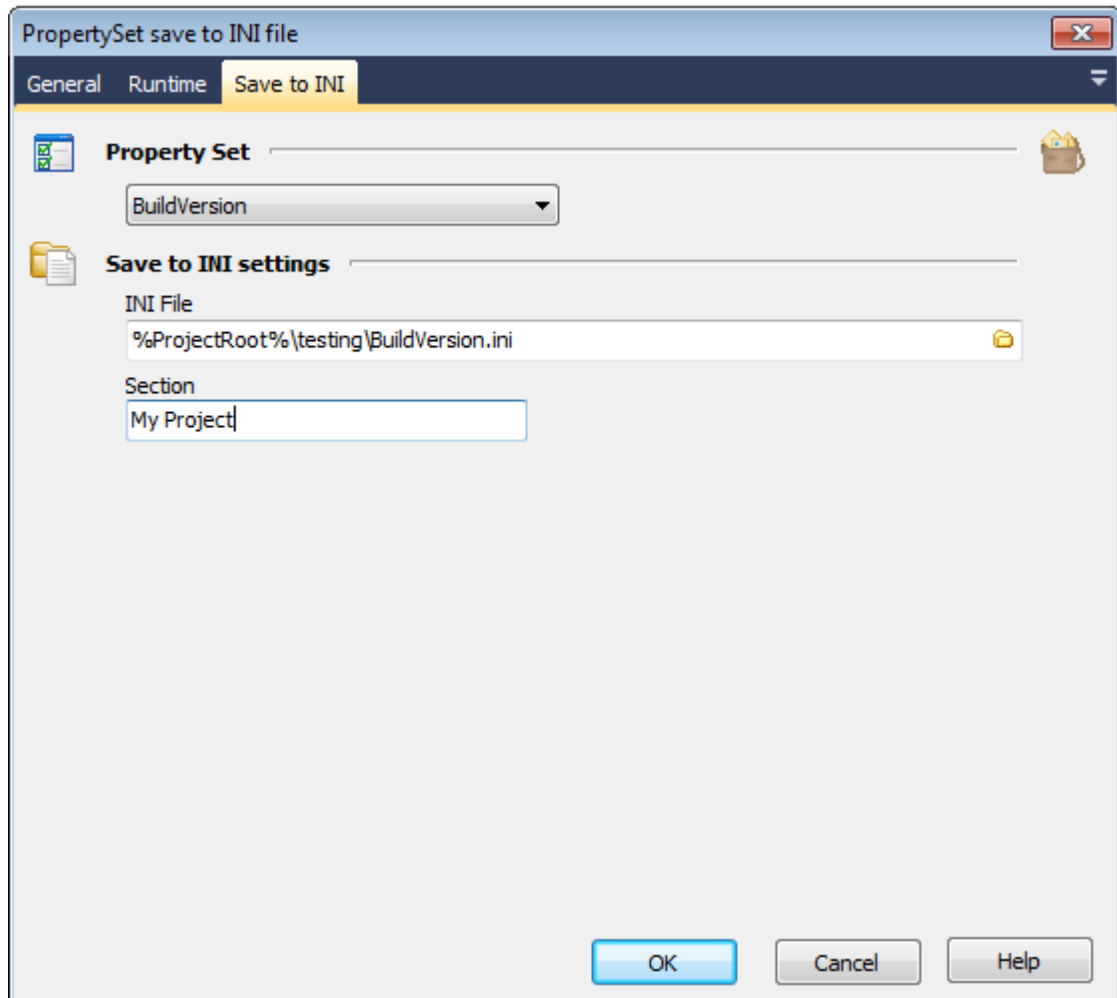


For more information on the FBD file, see [Project Files & Other Files](#).

The FBD file will be automatically created if it doesn't already exist.

6.28.12 PropertySet save to INI file

The PropertySet save to INI file action will persist the values of your property set to an INI file.



Property Set - select the property set which you want to save.

INI File - The INI file to save the PropertySet to.

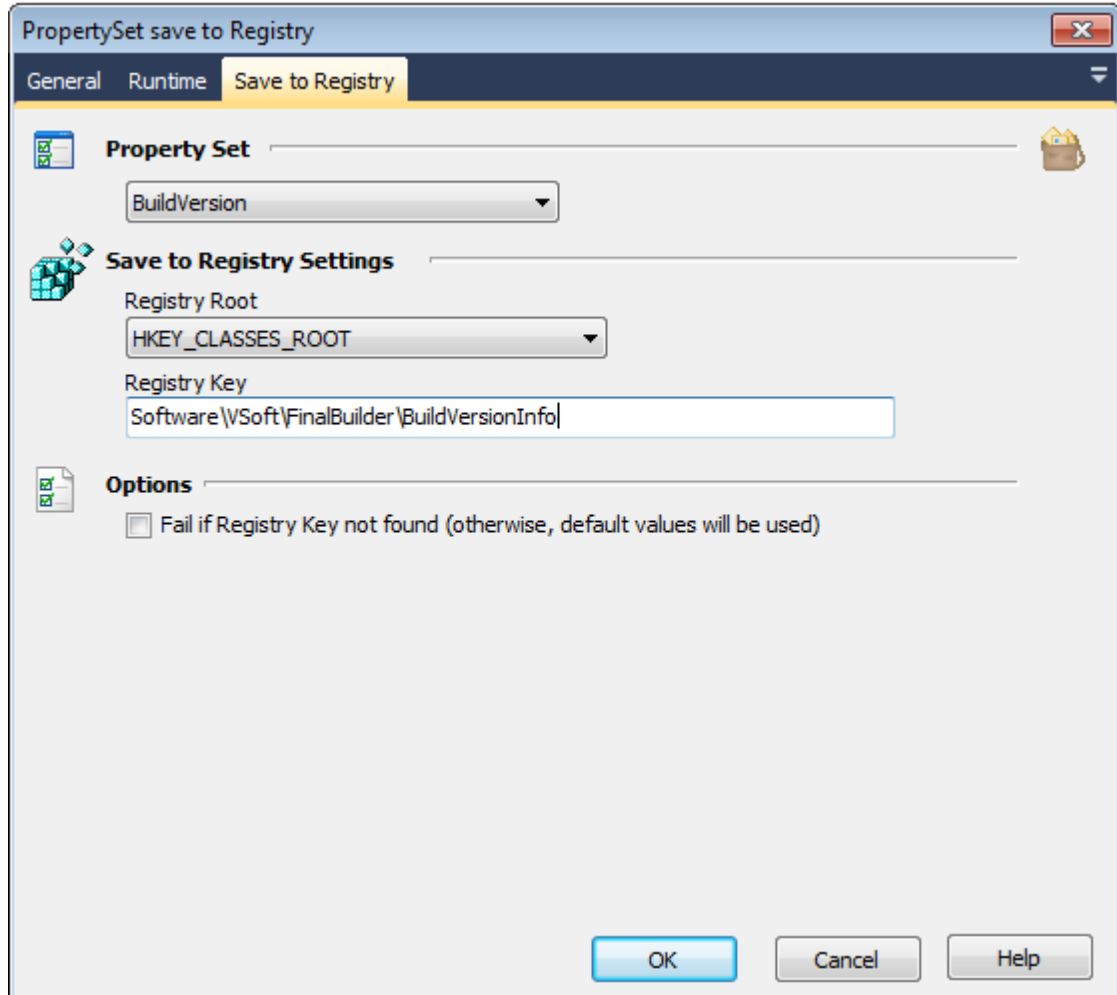
Section - The INI file section in which to store the PropertySet values

The INI file will automatically be created if it doesn't exist.

The Section in the INI file will also be created if it doesn't exist.

6.28.13 PropertySet save to Registry

The PropertySet save to Registry action will persist the values of your property set to the system registry.



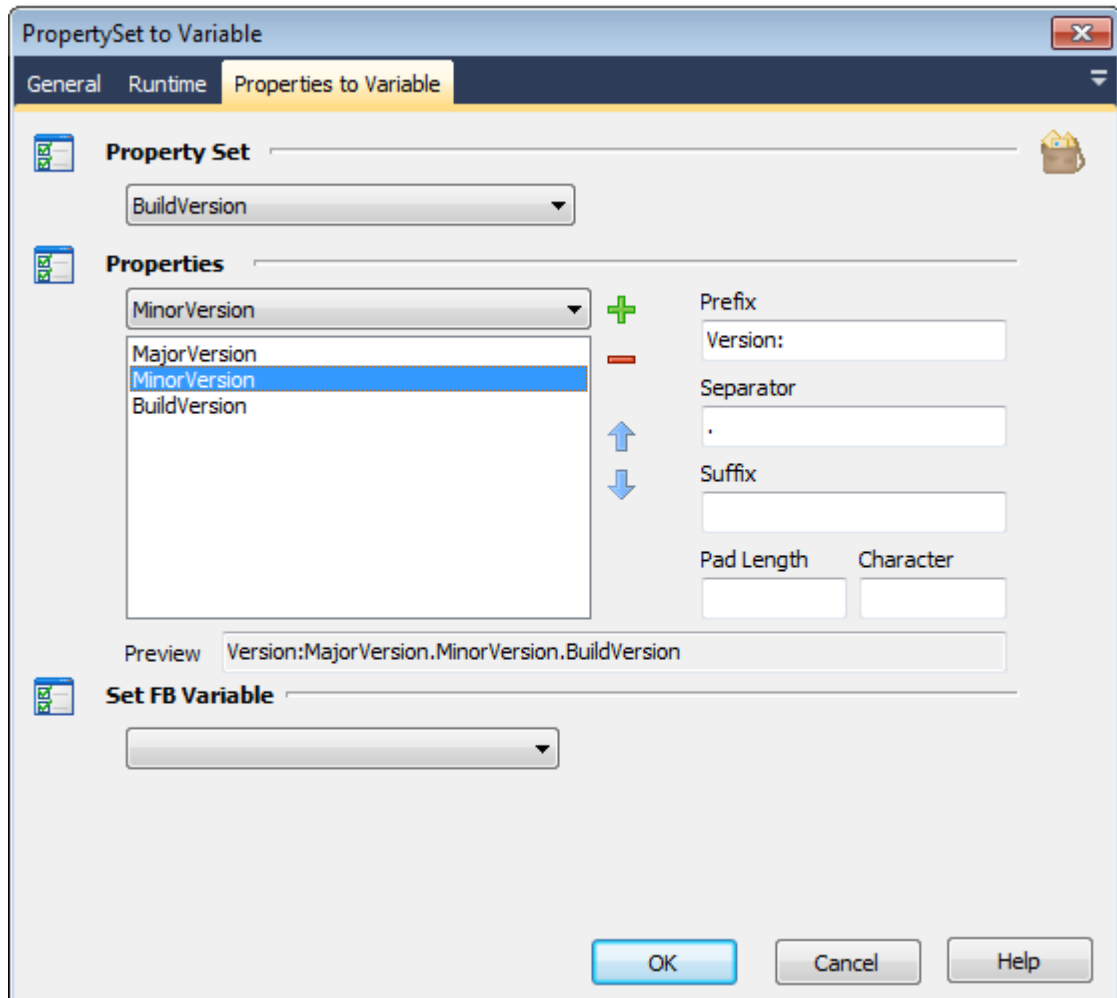
Property Set - select the property set which you want to save.

Registry Root - Select the Registry Root, usually Current_User or Local_Machine

Registry Key - Enter the registry key where you want the PropertySet values to be saved. If the key does not exists, then it will be automatically created.

6.28.14 PropertySet to Variable

Use the PropertySet to Variable action to create a text string from the values of your property set.



Property Set - Select the property set to use.

Properties - Select the property from the combobox and click the Plus button to add to the list. Any number of properties can be added to the list (duplicate properties are allowed). Use the up/down arrows to reorder the properties in the list.

Prefix - Optionally add a character string prefix.

Separator - This value is placed between each of the properties.

Suffix - Optionally add a character string suffix.

Pad Length & Character - Optionally pad the each property to the specified length.

Preview - the preview displays what the resultant text string will be according to the properties selected, prefix, separator, and suffix values.

Set FB Variable - the resultant text string will be saved in the selected FB variable.

Example.

In the above screen pic, if MajorVersion = 3, MinorVersion = 0, BuildVersion = 43, and ReleaseVersion = 9 then the variable would be set to:

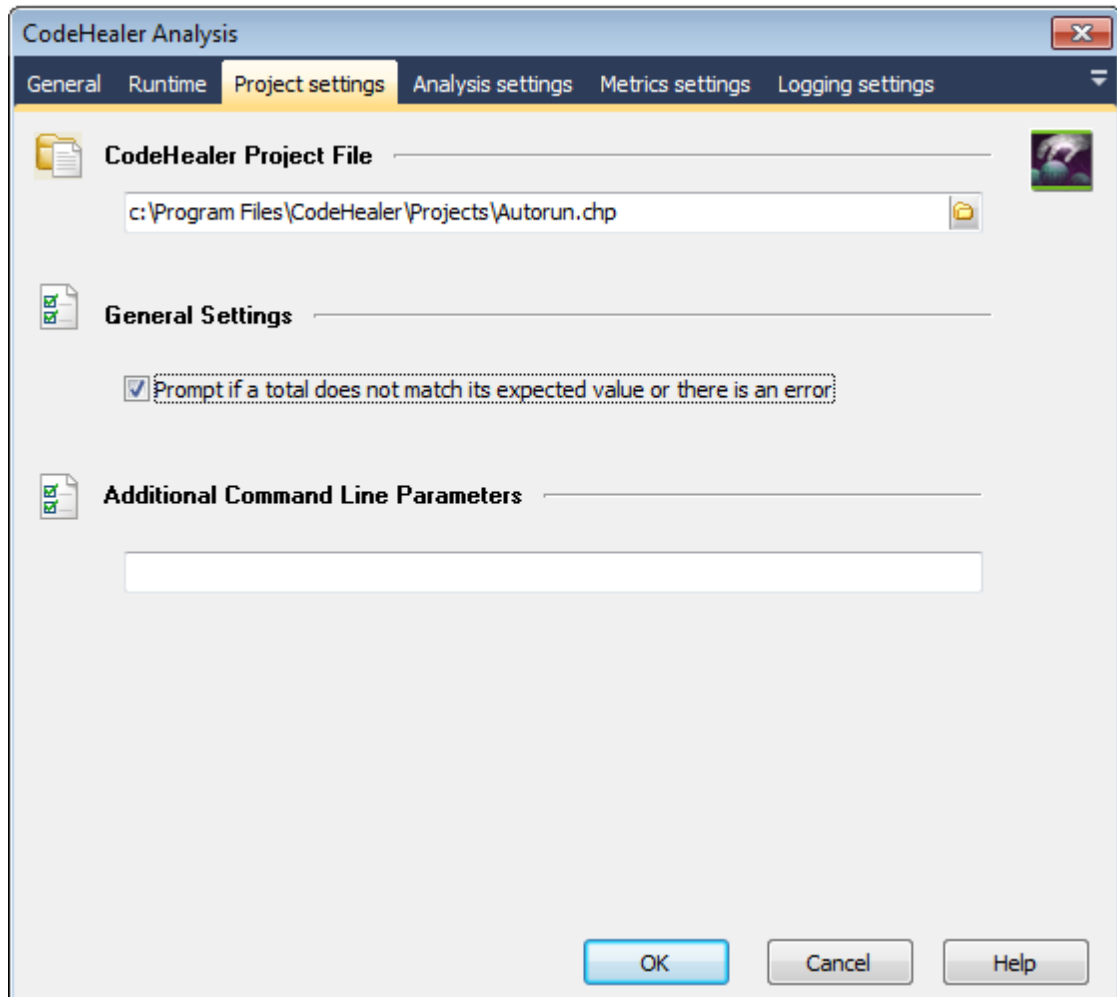
FB3.0.9.43_

Then other actions could use the %uservar% variable - for example the file may be renamed using the uservar variable as the new name of the file.

6.29 Source Code Tools

6.29.1 CodeHealer

This action provides an automated interface to CodeHealer, the source code analysis and verification tool, allowing the integration of automatic generation of source code analysis and metrics reports into your build process.



Please select the CodeHealer project file (.CHP) to be analysed, and also ensure that the project can be analysed successfully by manually running a full analysis from within CodeHealer itself before attempting to execute a CodeHealer Analysis action on a project from within FinalBuilder.

A number of options can be set from within this CodeHealer Analysis action, but many can only be set from within CodeHealer. Please ensure that all the required or appropriate preferences have been set from within CodeHealer before attempting to execute a CodeHealer Analysis action on a project.

Each CodeHealer Analysis action must perform either one or both of a full analysis or a calculation of metrics for the selected project, and the results can be checked and exported to various file formats using the settings on the pages of this action.

The available settings should be fairly self explanatory, but for more details on the

meaning and usage of each of the settings that are available in this action please refer to the **Command Line Parameters** section of the **CodeHealer Help** file.

Finally, make sure that you have correctly configured the location of the CodeHealer program (CODEHEALER.EXE) and its default options by selecting **Tools | Options | Source Code Tools | CodeHealer** from the FinalBuilder main menu.

What is CodeHealer

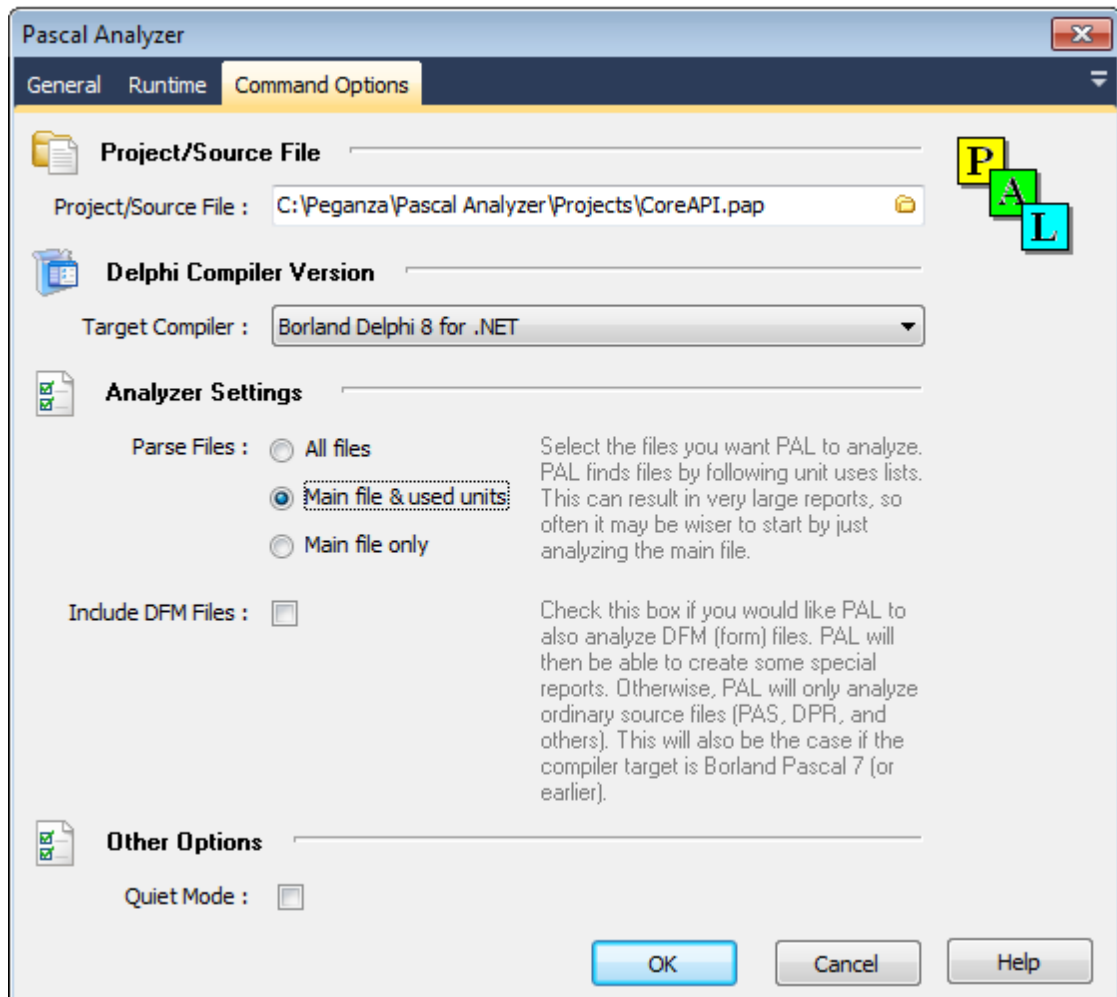
CodeHealer is an efficient, powerful and easy to use program source code analysis and verification tool that will help find and fix a significant number of programming bugs, mistakes and inconsistencies in Delphi programs before they are released.

The information provided by CodeHealer will also help developers understand and improve existing code that they have not written. This is especially helpful for corporate developers who take over the management of legacy applications.

For more information on CodeHealer, and to download a trial version, please visit <http://www.socksoftware.com>.

6.29.2 Pascal Analyzer

The Pascal Analyzer action enables you to analyze your Pascal source code using Peganza Pascal Analyzer.



Select the main file to be analyzed, this can be a Pascal Analyzer project file (.PAP), or Delphi source file (.DPR, .DPK, .PAS), and optionally select the scope of the analysis, and the Embarcadero compiler version you are using.

Make sure that you have configured the location of PALCMD.EXE (the Pascal Analyzer command-line utility), and set other default options via Tools | Options | Source Code Tools | Pascal Analyzer. Other settings such as style and destination of output, and which reports will be included are read from the Pascal Analyzer configuration file which can be configured via the GUI version of Pascal Analyzer (PAL.EXE).

What is Pascal Analyzer

Pascal Analyzer, or PAL for short, is a utility program that analyzes, documents, debugs, and helps you optimize your source code. It will help you better understand your code and support you in producing code of higher quality, consistency, and reliability. PAL quickly pays itself back in easier maintenance, less errors and improved quality, not only during development, but also throughout the entire life cycle of your code.

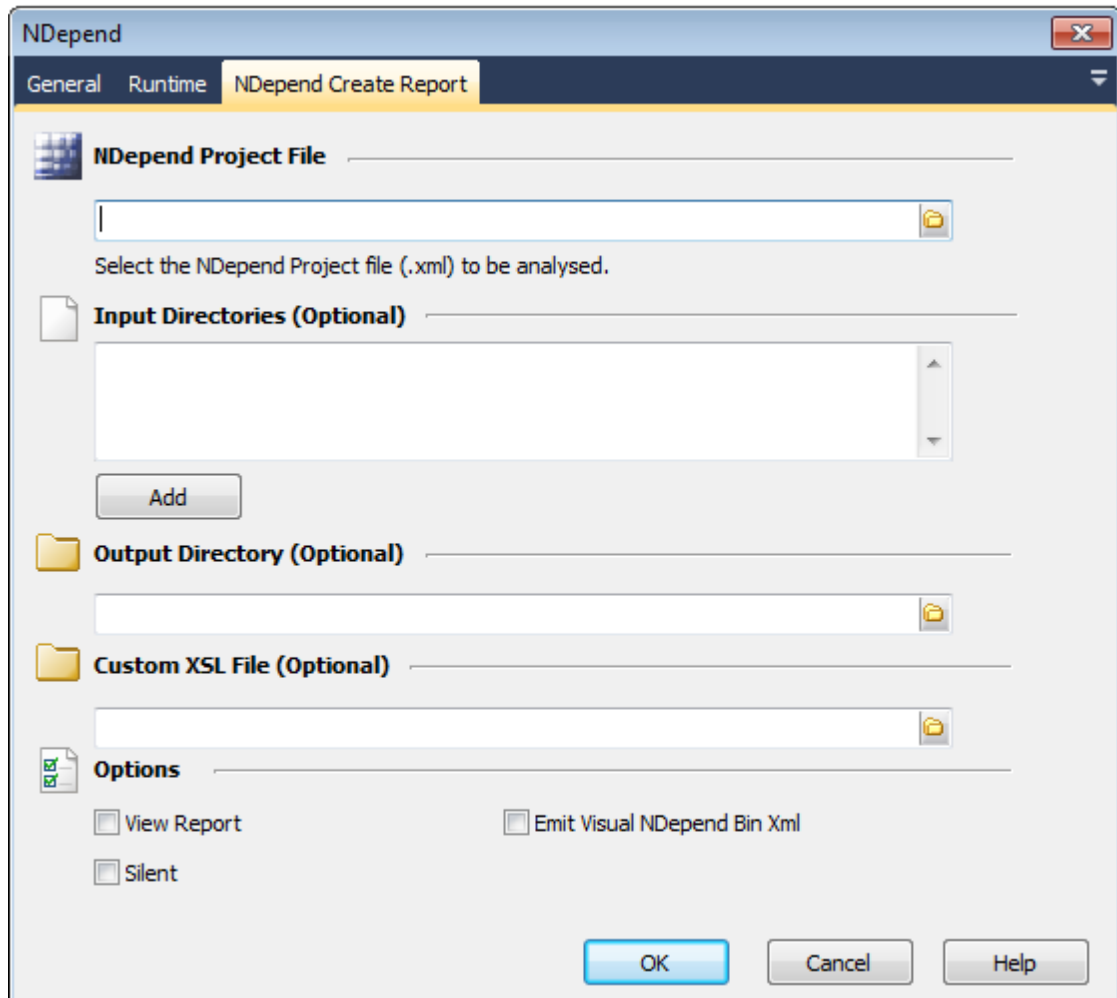
For more information see <http://www.peganza.com>

How to use PALCMD.EXE

The FinalBuilder Pascal Analyzer plugin uses the standalone command-line version PALCMD.EXE. PALCMD.EXE uses the same engine as the GUI version PAL.EXE and produces the same output.

6.29.3 NDepend

NDepend allows you to automate the analysis of a NDepend Visual project file.

**NDepend Project File ***

Specify the path of the NDepend Visual Project file (.xml) to be analysed. This field is a required value.

Input Directories (Optional)*

Override the Input directories specified in the NDepend Visual Project file. The directories in the Project file will be ignored if you specify your own input directories.

Output Directory (Optional)*

Override the default output directory (which is the NDepend Visual project directory).

Custom XSL File (Optional)*

Specify a custom XSL file to allow custom formatting of the output report.

View Report

Display the HTML report once the action is run.

Silent

Suppress log output from NDepend console.

Emit Visual NDepend Bin Xml

Include the VisualNDepend.bin.xml file in the output files.

** Note: All paths must be specified as an absolute reference, relative paths will not work.*

For more information on NDepend options please visit: [NDepend console options](#)

6.30 Stack and Queue Actions

[FinalBuilder Professional Edition]

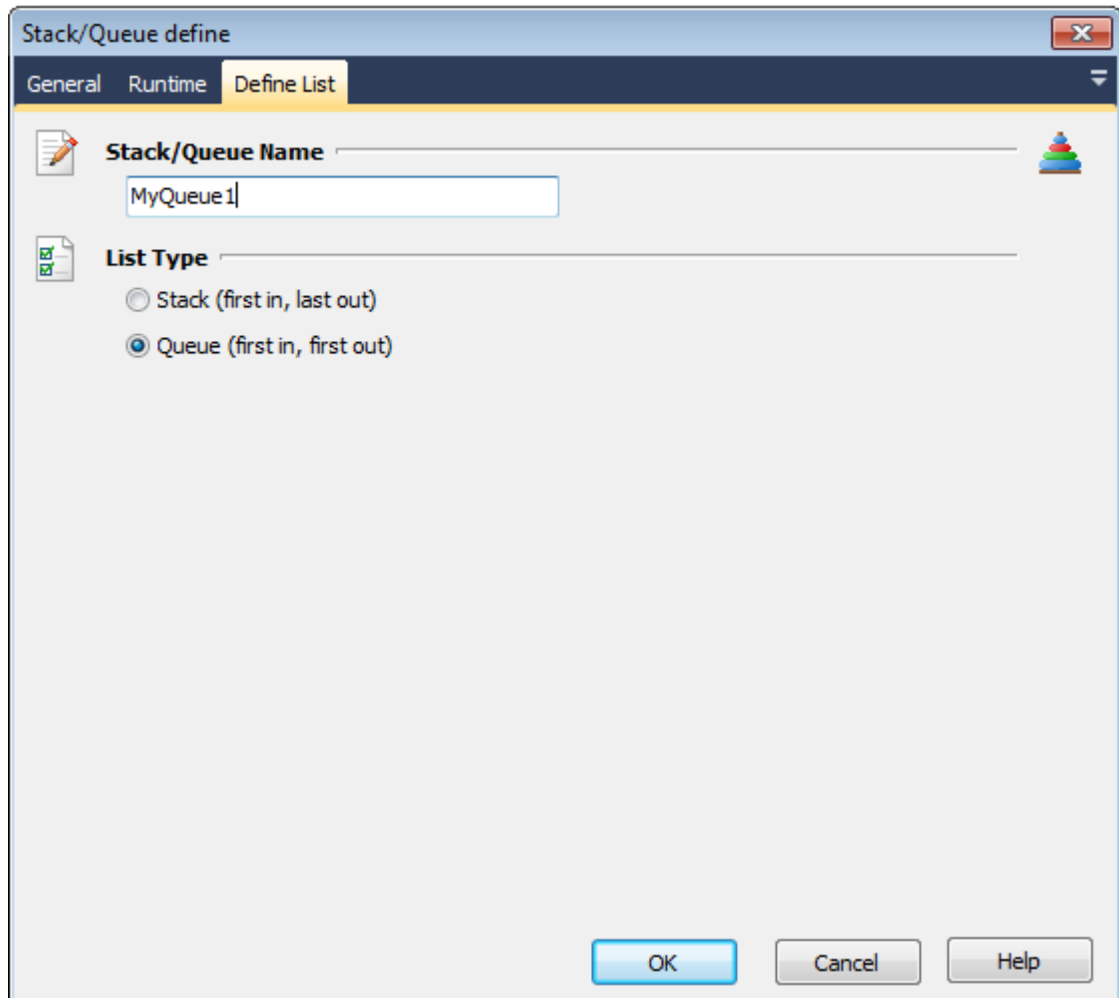
The Stack and Queue actions allow you to build up a list of items and then access the items in either a first in, first out (Queue) or first in, last out (Stack) manner.

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.

6.30.1 Stack/Queue Define

[FinalBuilder Professional Edition]

The Stack/Queue Define action is used for creating a new list and giving it a name.



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

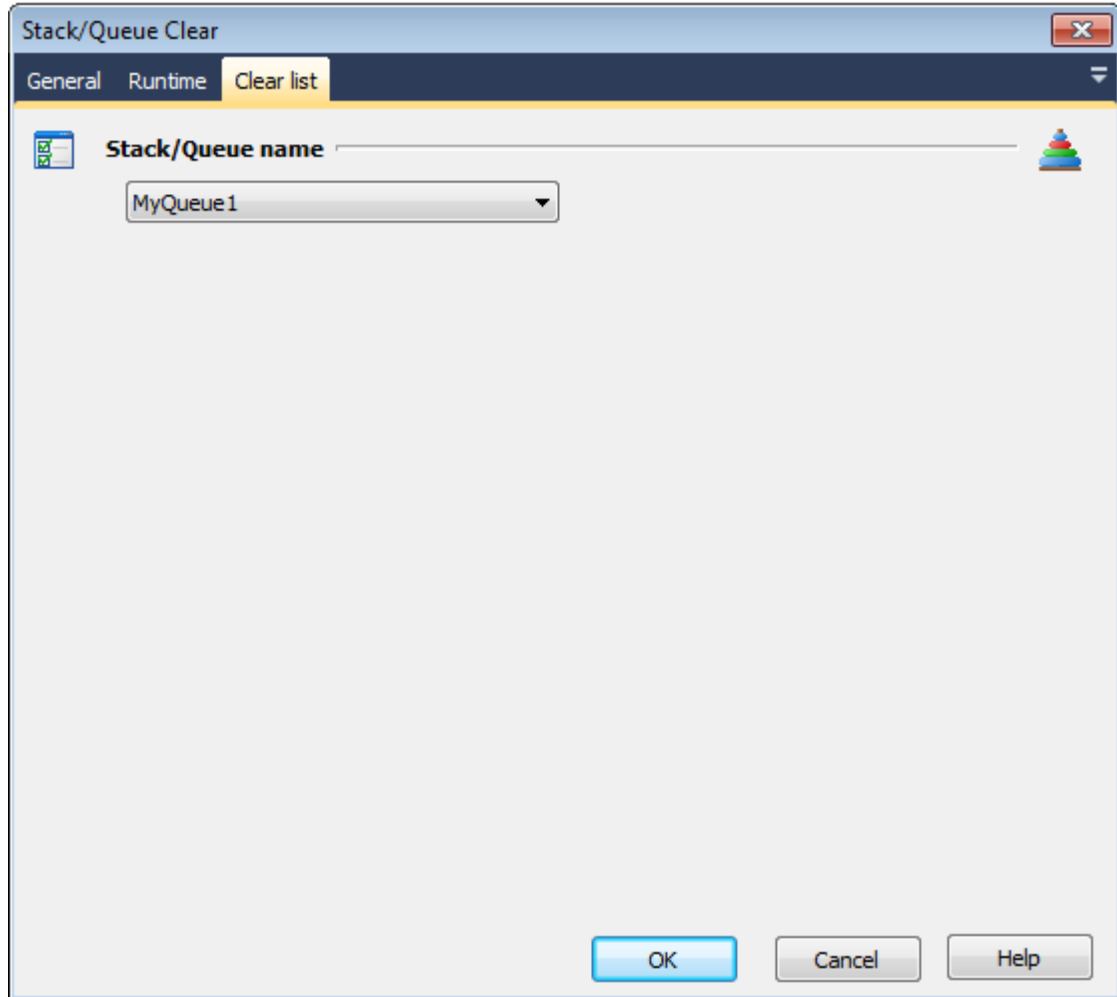
If we pop the items off a Queue, we get:

- Pop a
- Pop b
- Pop c

6.30.2 Stack/Queue Clear

[FinalBuilder Professional Edition]

The Stack/Queue Clear action is used to remove all values from the list.

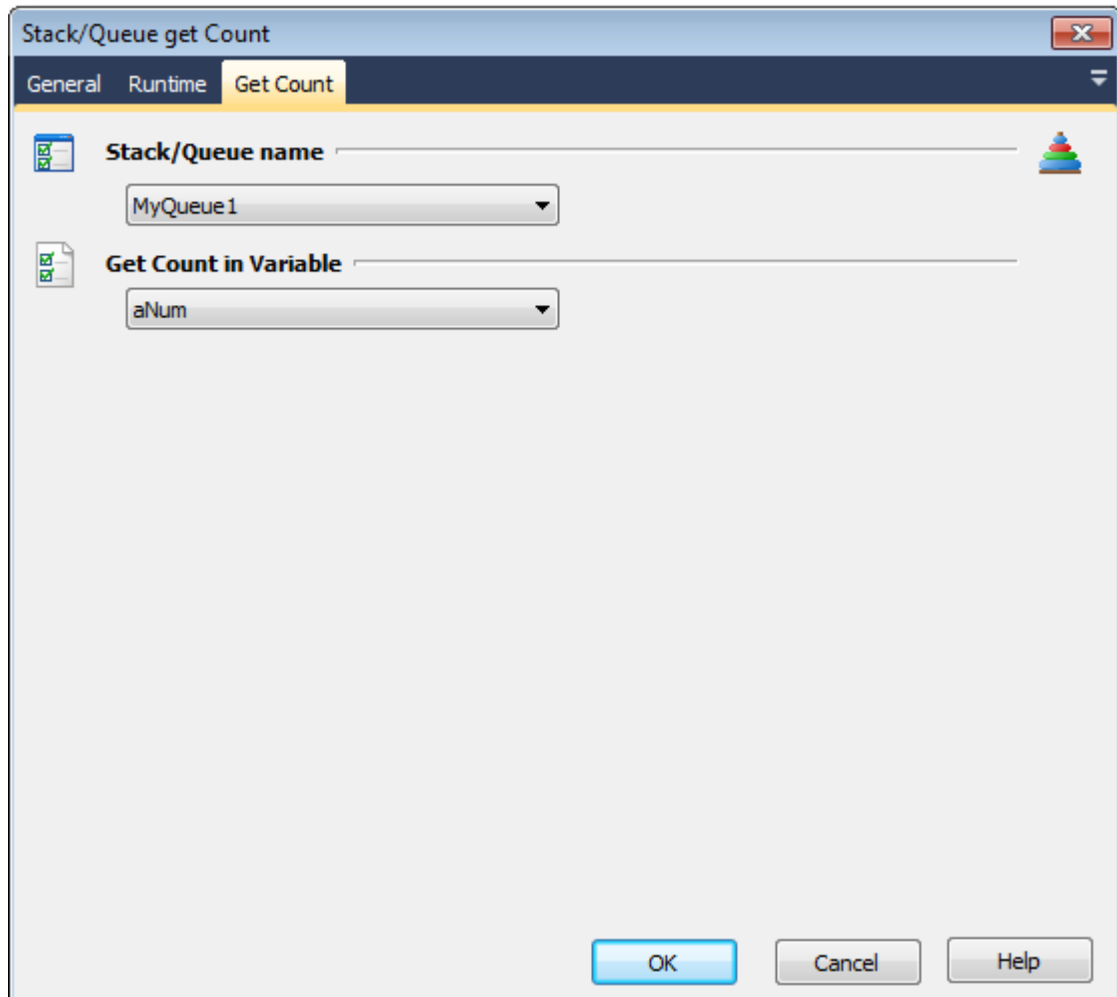


Stack/Queue name - select the name of the list which you want to clear. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

6.30.3 Stack/Queue Get Count

[FinalBuilder Professional Edition]

The Stack/Queue Get Count action is used to get the amount of items that is currently on the list.



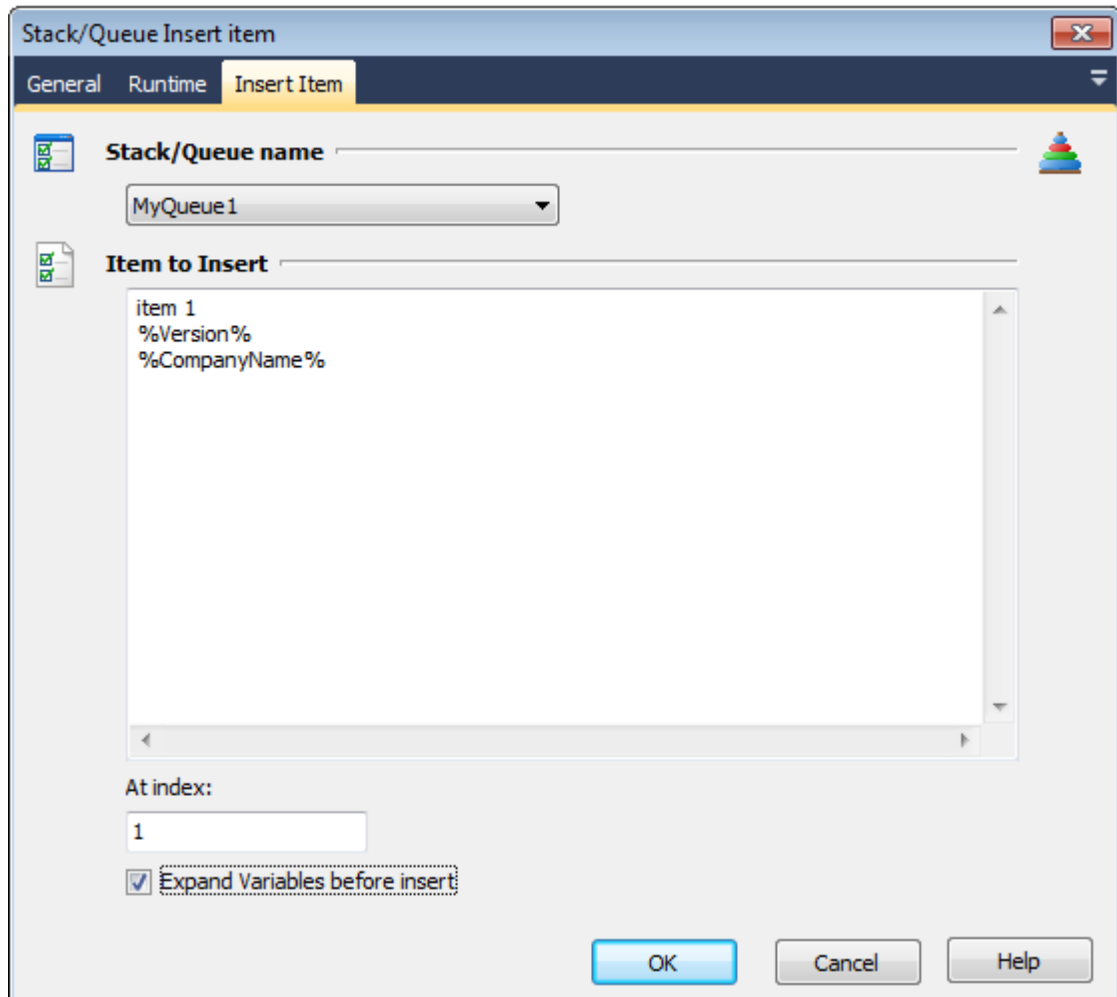
Stack/Queue name - select the name of the list which you want to get the count of items in. If there aren't any items in this list, you need to use the Stack/Queue Define action to create a new list.

Get Count in Variable - specify the variable to set the count to

6.30.4 Stack/Queue Insert Item

[FinalBuilder Professional Edition]

The Stack/Queue Insert Item action is used to push an item at a particular location in the list.



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.

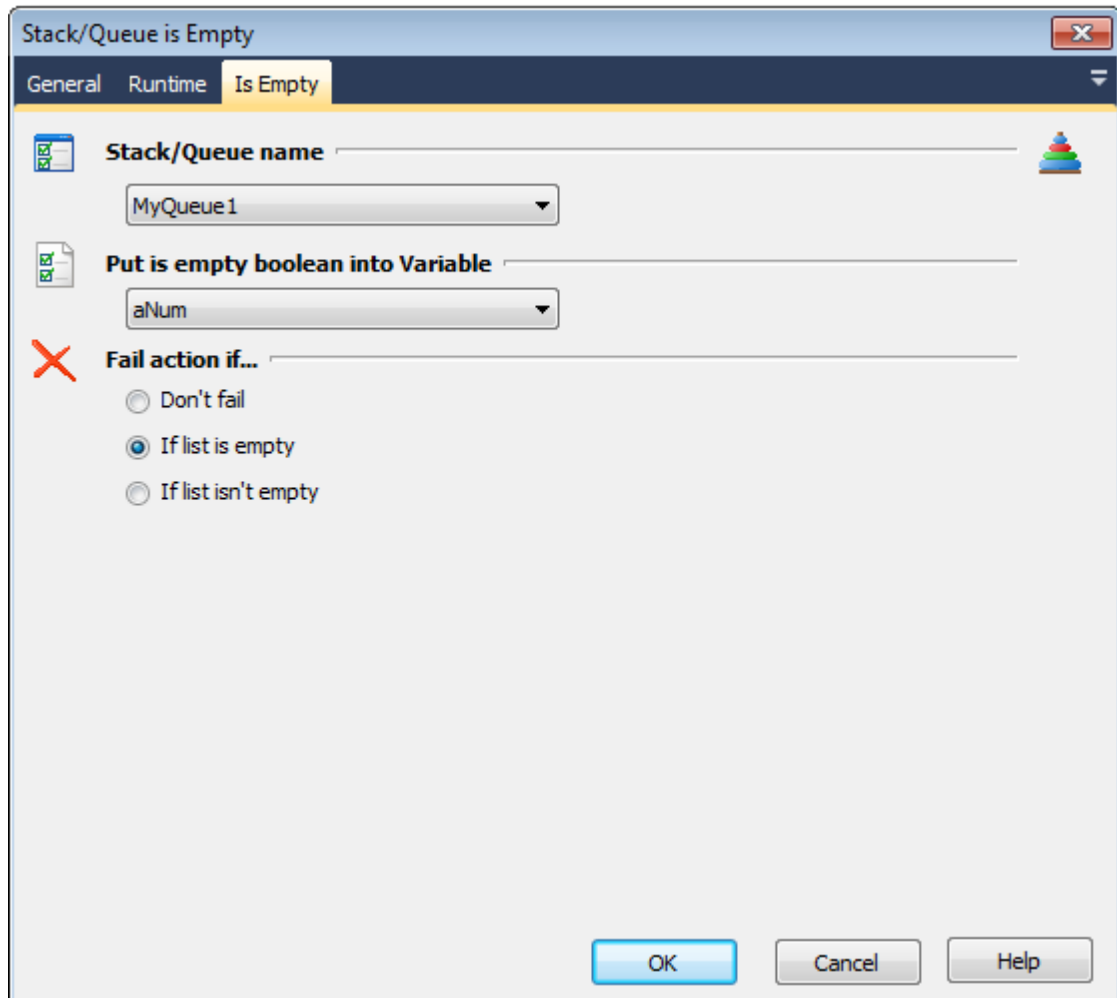
Expand variables before insert - if you specify variables as part of the text to push, eg. %MyVariable% then selecting this option will expand the variables before the item is pushed onto the list.

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.

6.30.5 Stack/Queue Is Empty

[FinalBuilder Professional Edition]

The Stack/Queue Is Empty action is used to test if the list contains any items or not.



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.

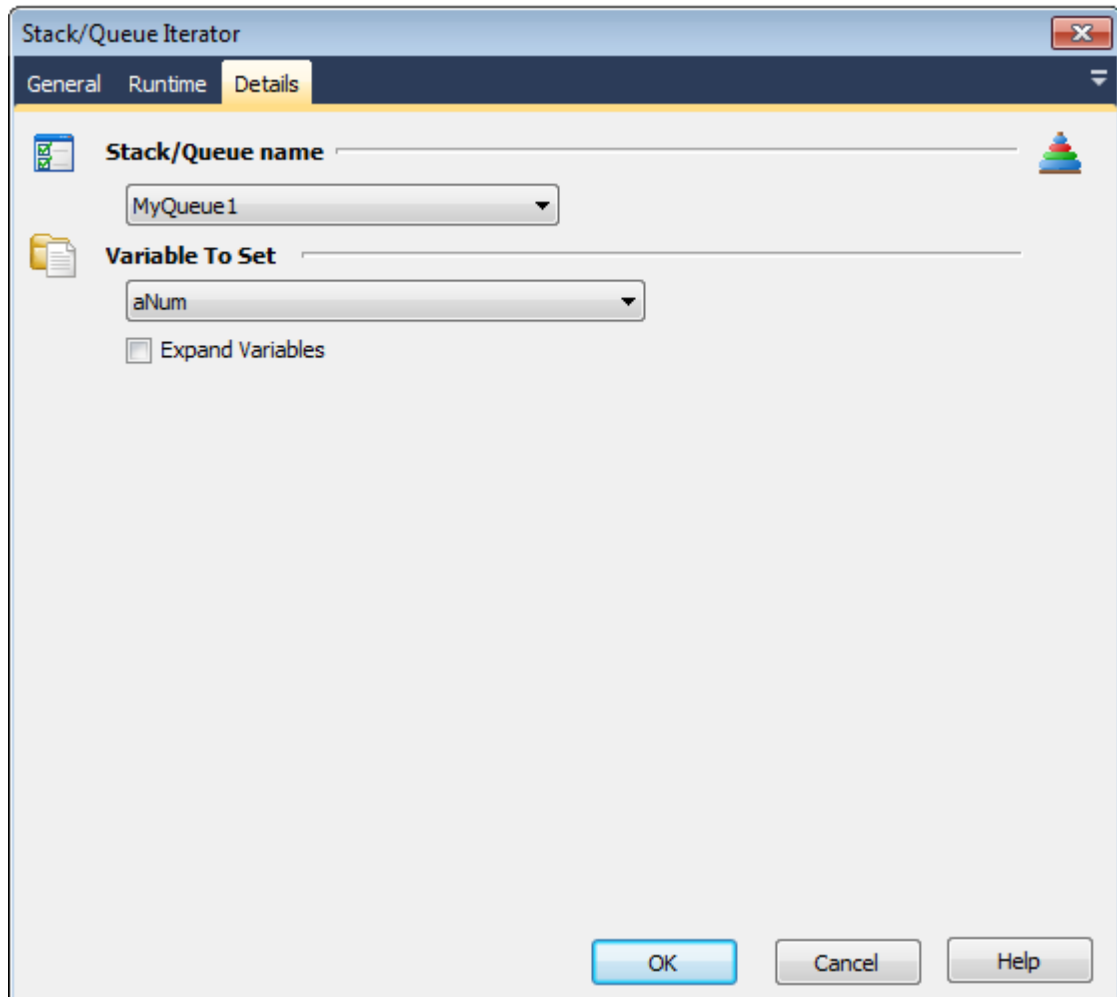
Put is empty boolean into Variable - specify a variable to set to a boolean indicating if the list is empty or not (True means list is empty).

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.

6.30.6 Stack/Queue Iterator

[FinalBuilder Professional Edition]

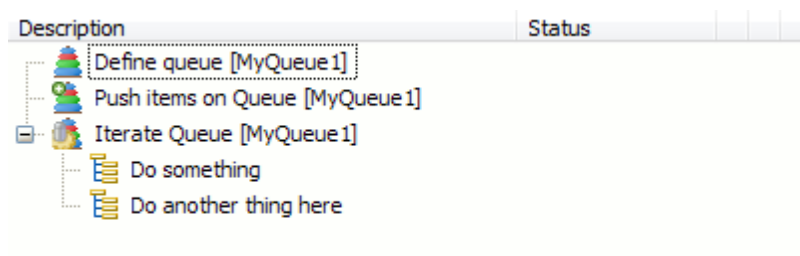
The Stack/Queue Iterator action is used to run a set of actions for each of the items in the list.



Stack/Queue 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.



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.

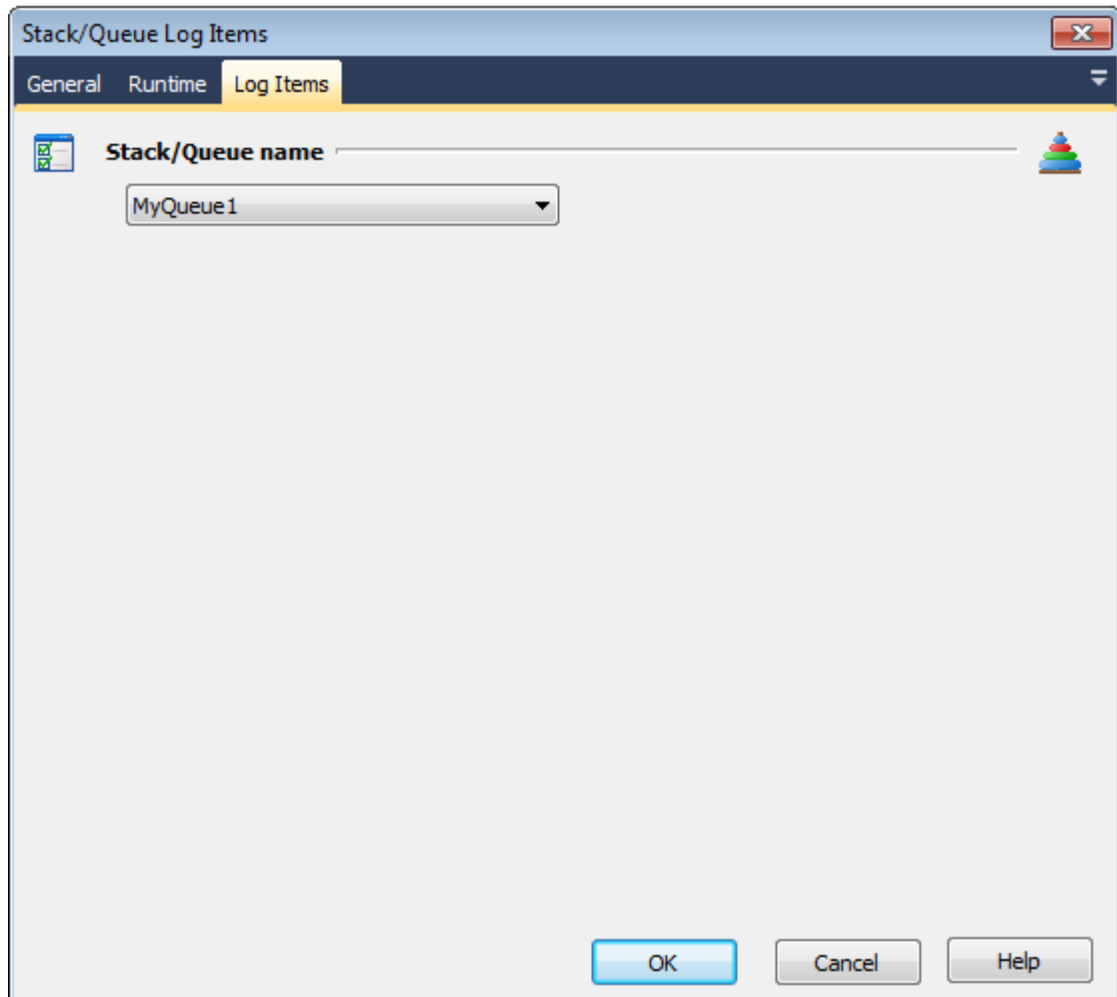
If the list includes the following values: "a" and "b". Then, the Iterator Variable will be set to "a" and then the "Do..." actions will be executed. And then for the next iteration, Iterator Variable will be set to "b" and then the "Do..." actions run again.

After the iterator action runs, the list will be empty.

6.30.7 Stack/Queue Log Items

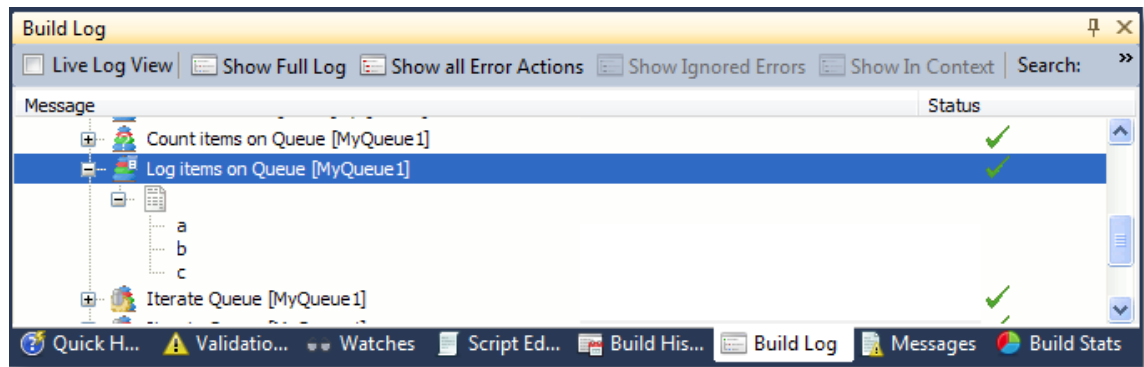
[FinalBuilder Professional Edition]

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 - 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.

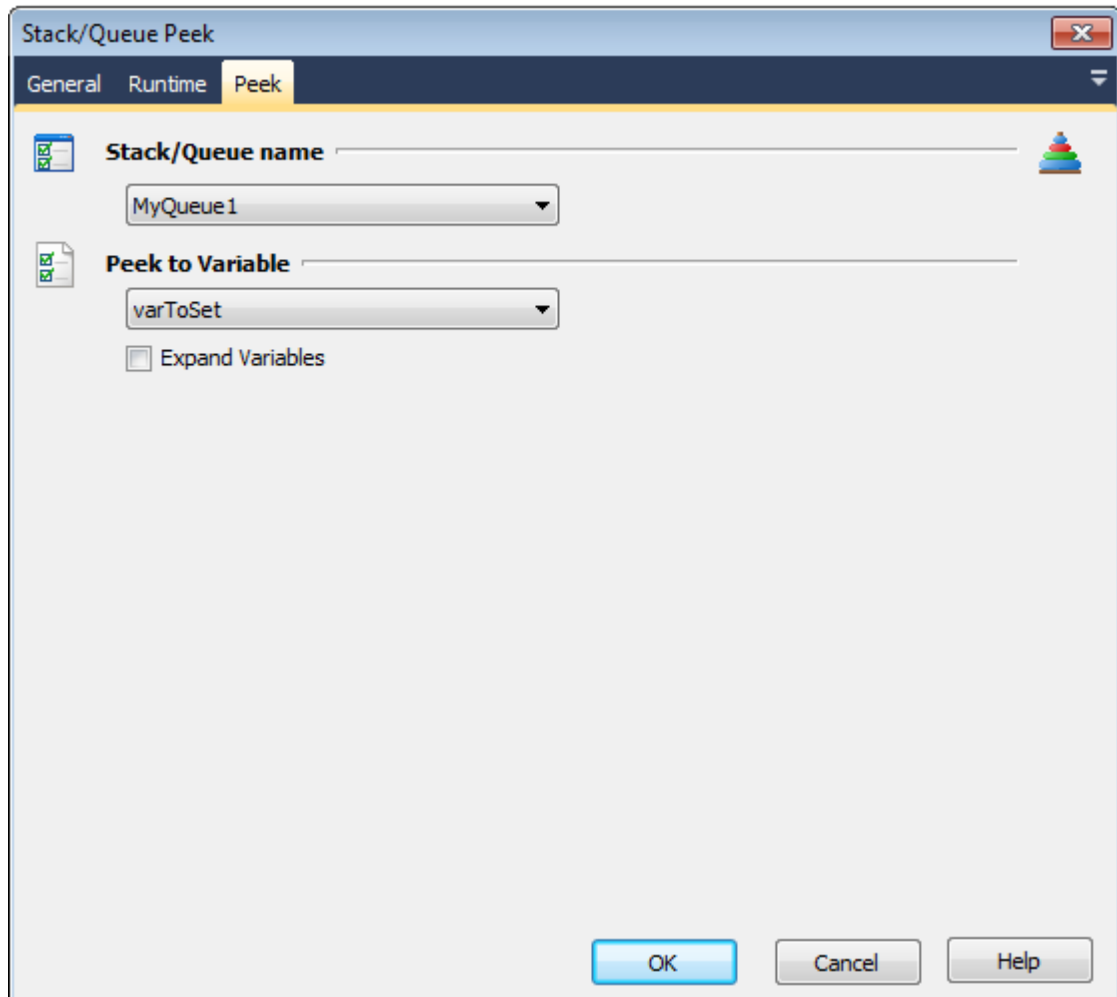
Below is an example of the log output of a queue that contains 3 items, "a", "b", and "c".



6.30.8 Stack/Queue Peek Item

[FinalBuilder 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 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.

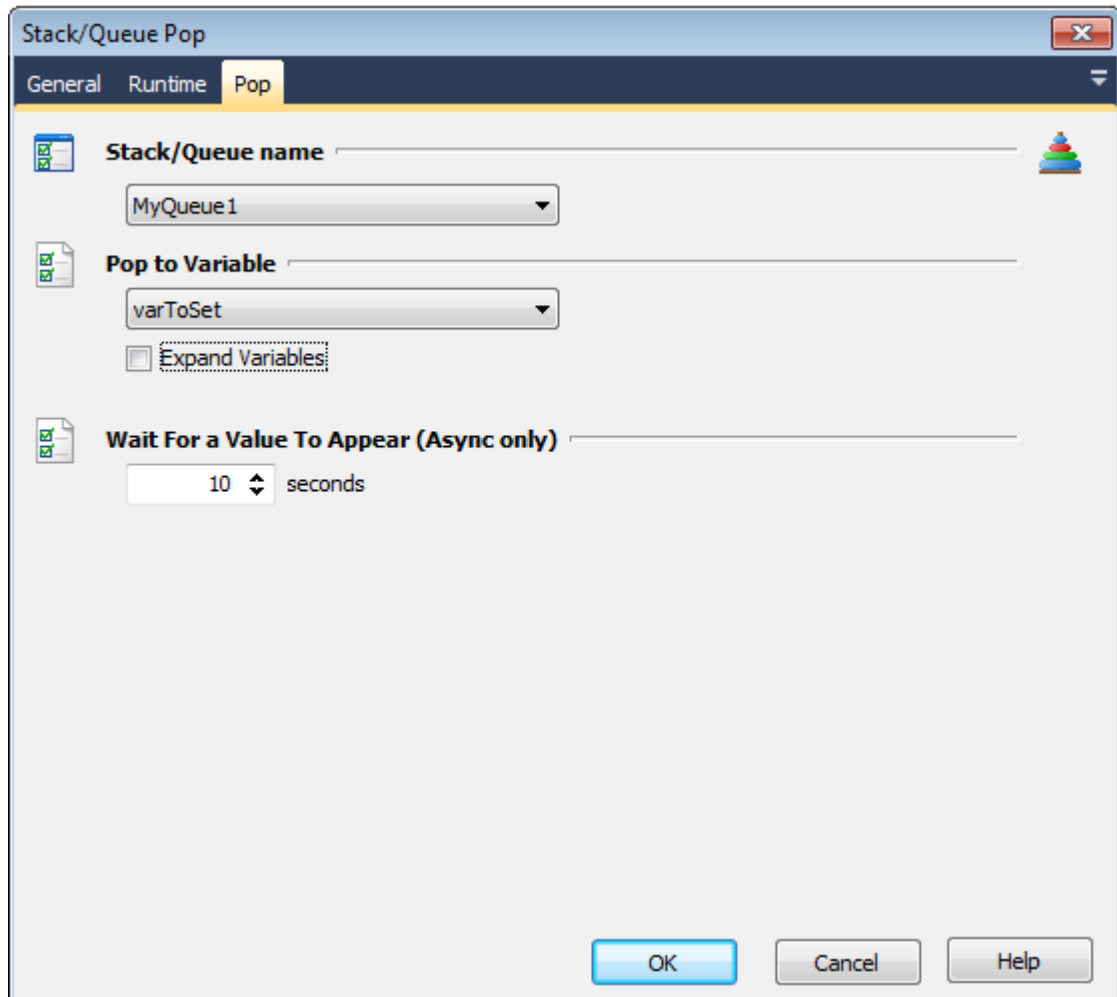
Expand variables - this will expand any variables in the item after it has been retrieved.

NOTE: If there are no more items to peek, then the action will fail.

6.30.9 Stack/Queue Pop Item

[FinalBuilder 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 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.

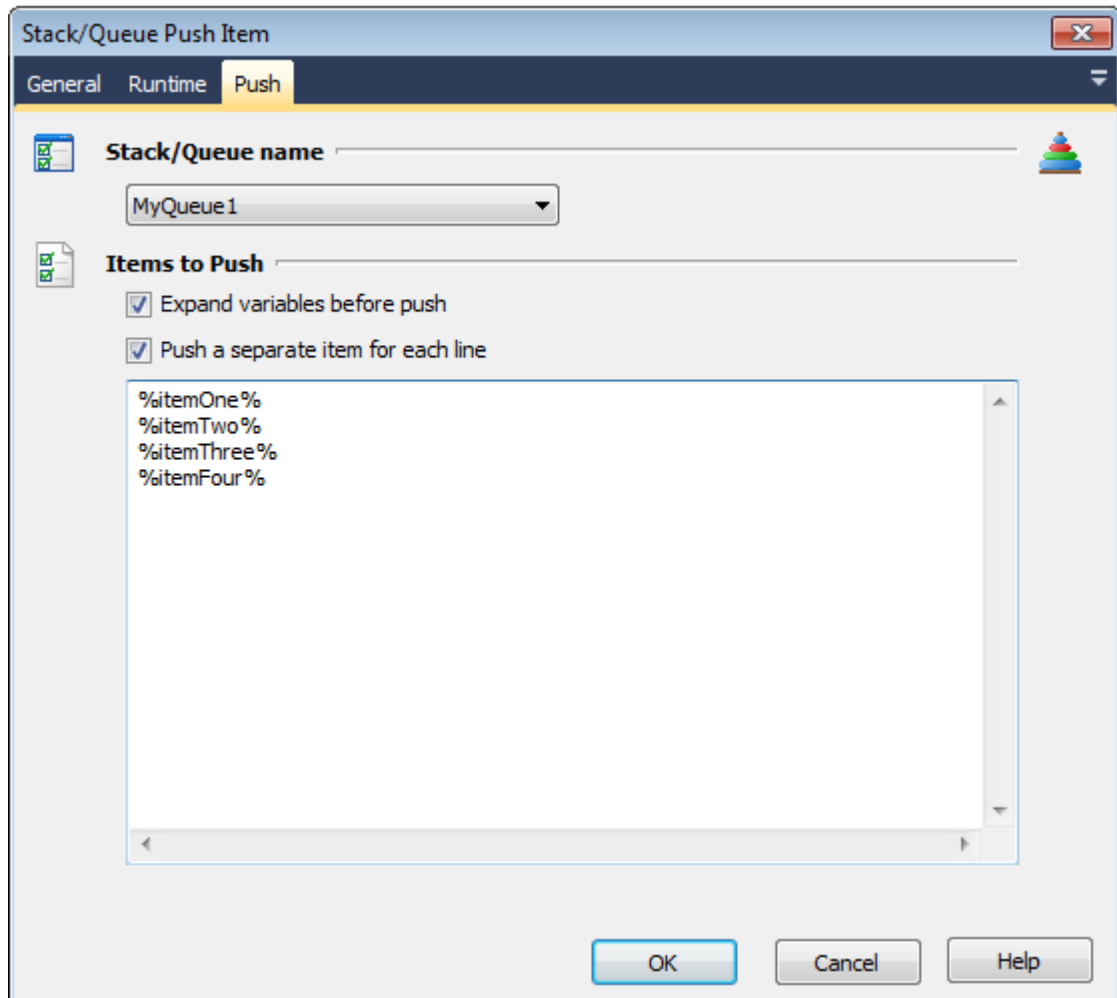
Wait For a Value to Appear - If there are no items to pop, then the action will wait for this amount of times for an item to appear in the queue. If no item appears within the specified time, the action will fail.

NOTE: If there are no more items to pop, then the action will fail.

6.30.10 Stack/Queue Push Item

[FinalBuilder Professional Edition]

The Stack/Queue Push Item action is used to push one or more items into the list.



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

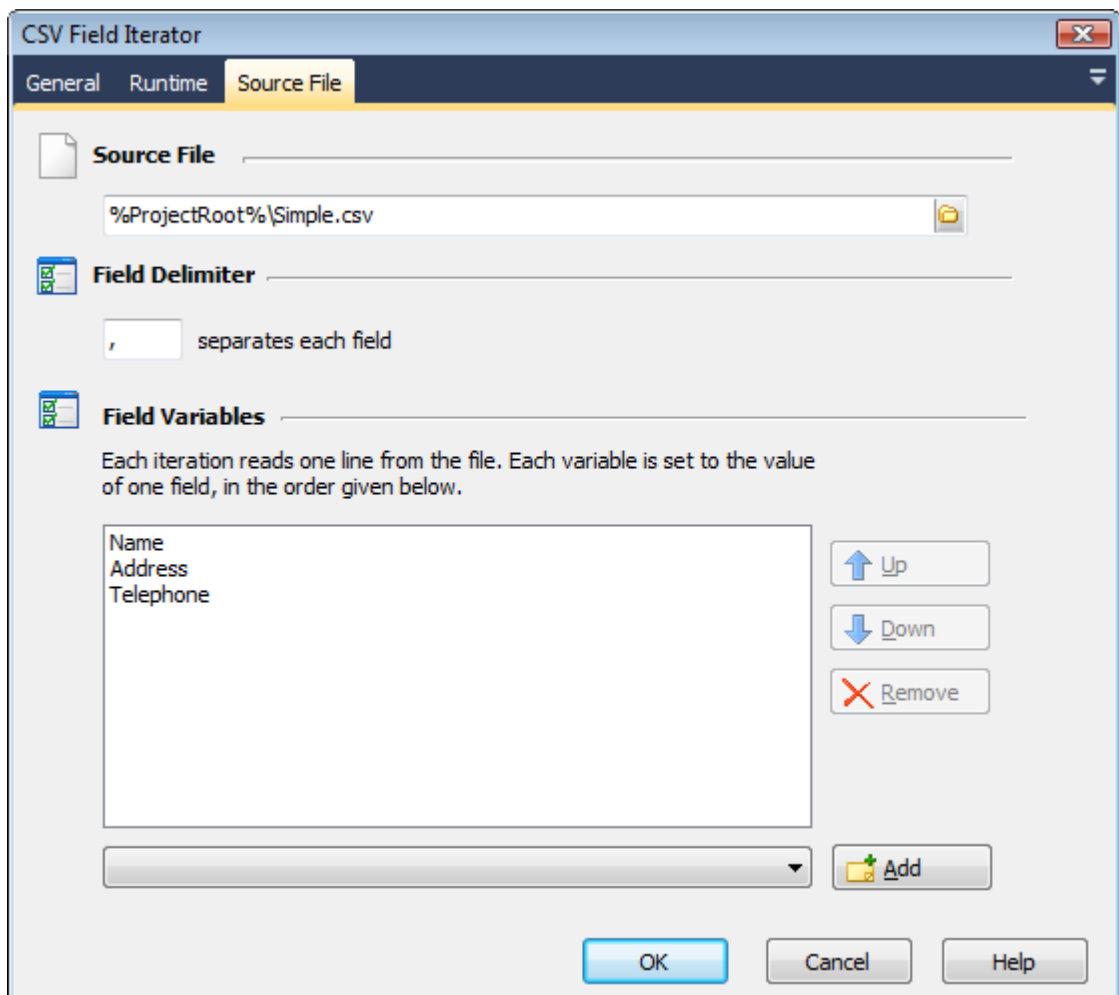
6.31 String Manipulation

6.31.1 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.



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.

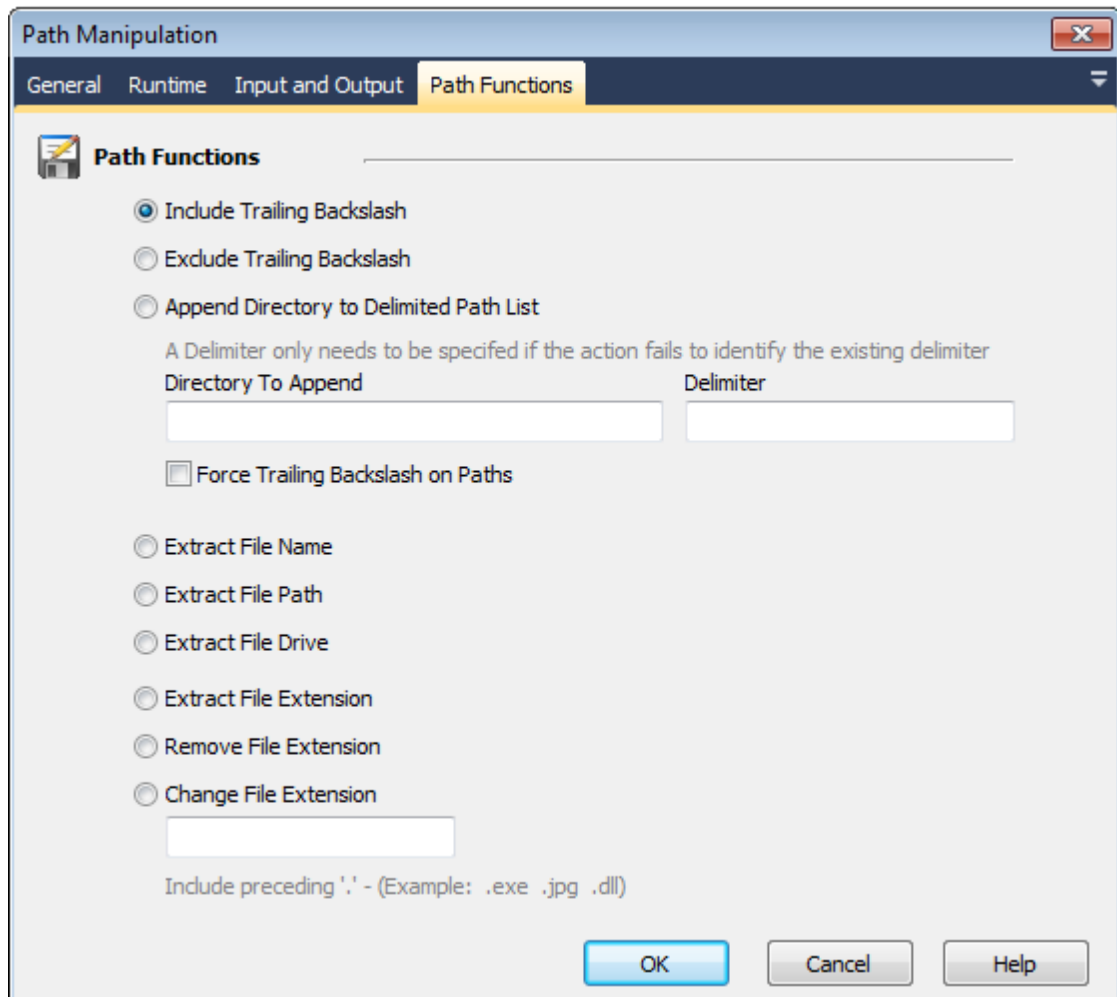
To move a variable up or down in the list, select it and then click the Up or Down button. To remove a variable, select it and click Remove.

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.

6.31.2 Path Manipulation

The Path Manipulation action allows you to perform common file path functions on a string.



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"

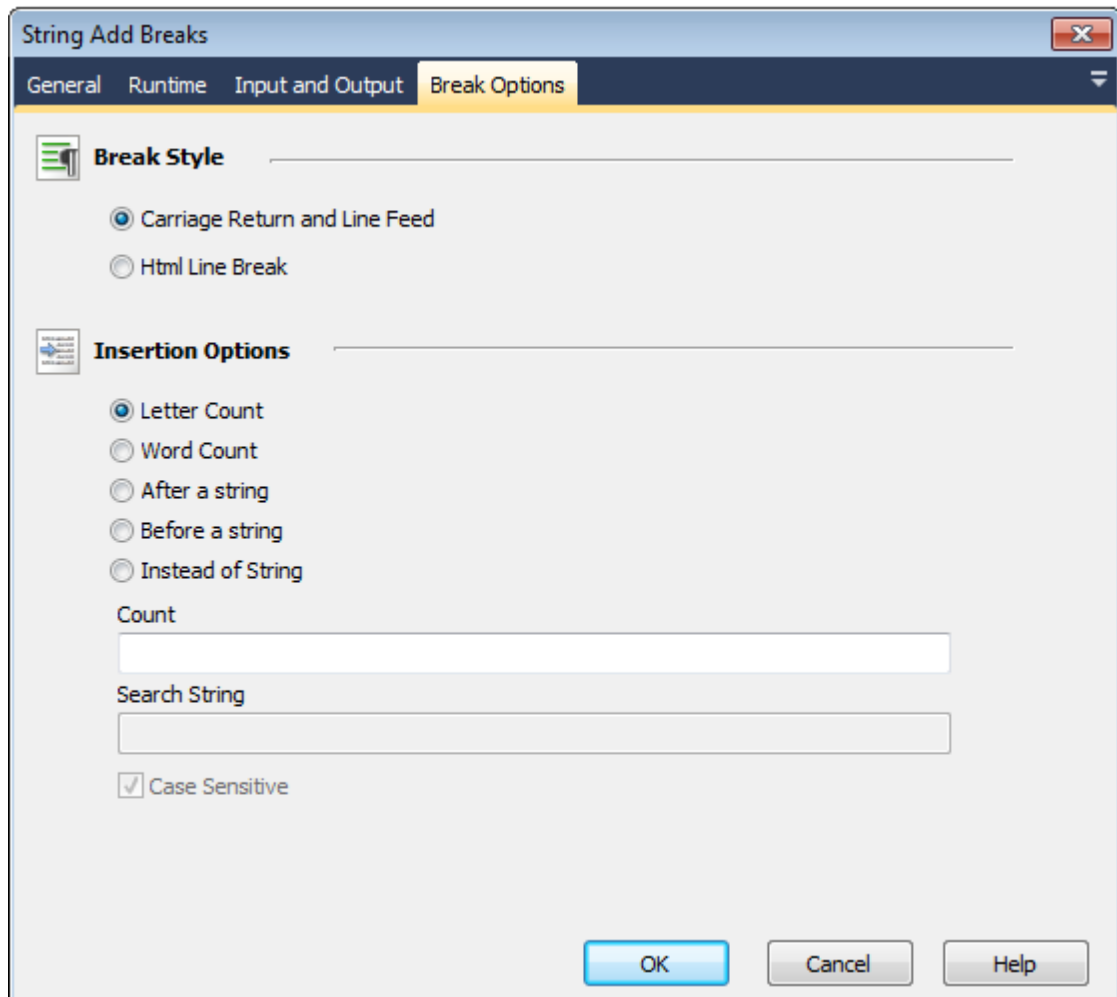
Scripting Info

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

6.31.3 String Add Breaks

The String Add Breaks action enables you to add breaks at specified locations within a string.



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

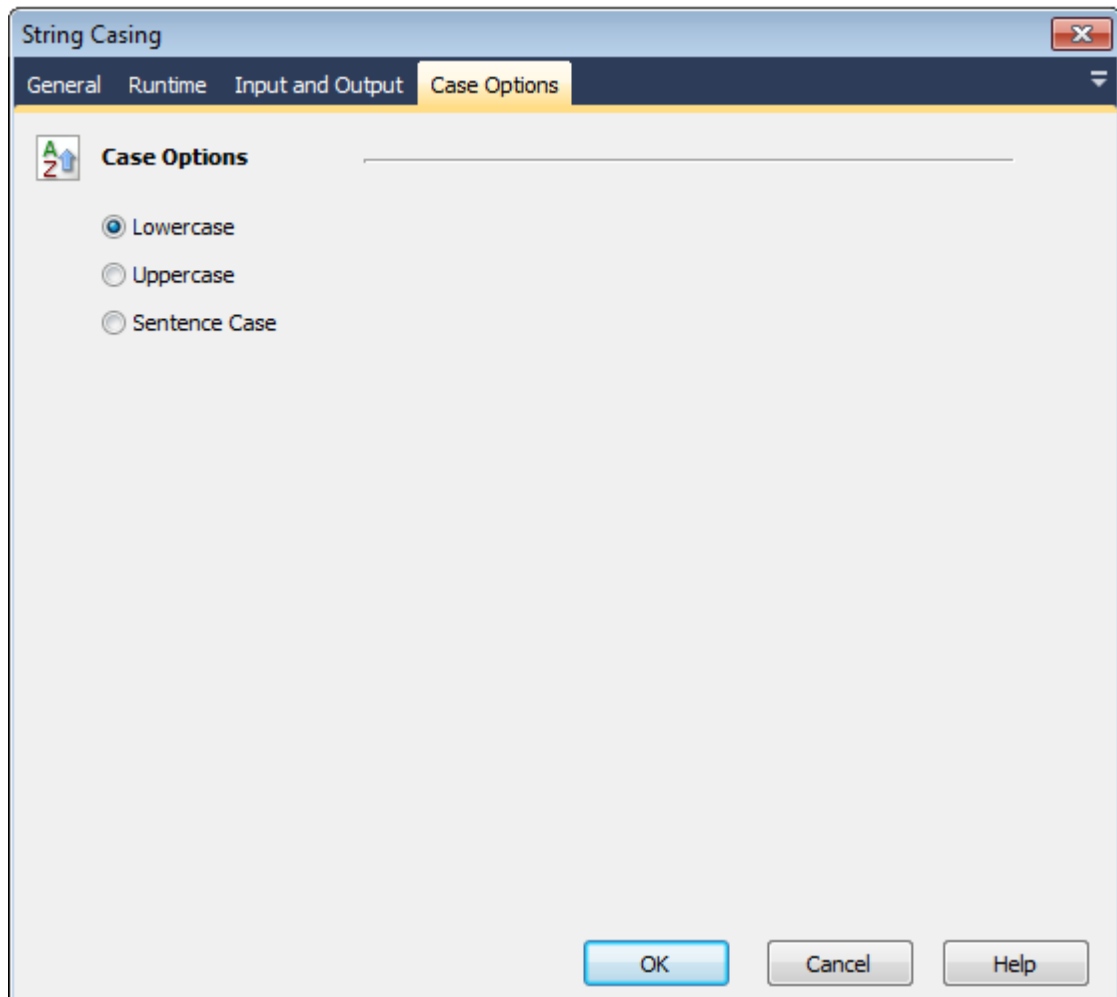
property UseHtmlBreak        : boolean;      // True to Use Html Break,
False to use CRLF
property BreakOnLetterCount   : boolean;      // True to Break On Letter
Count, otherwise false
property BreakOnWordCount     : boolean;      // True to Break On Word
Count, otherwise false
property BreakOnAfterString   : 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.

```

6.31.4 String Casing

The String Casing action allows you to change a string to be Lowercase, Uppercase or 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 :

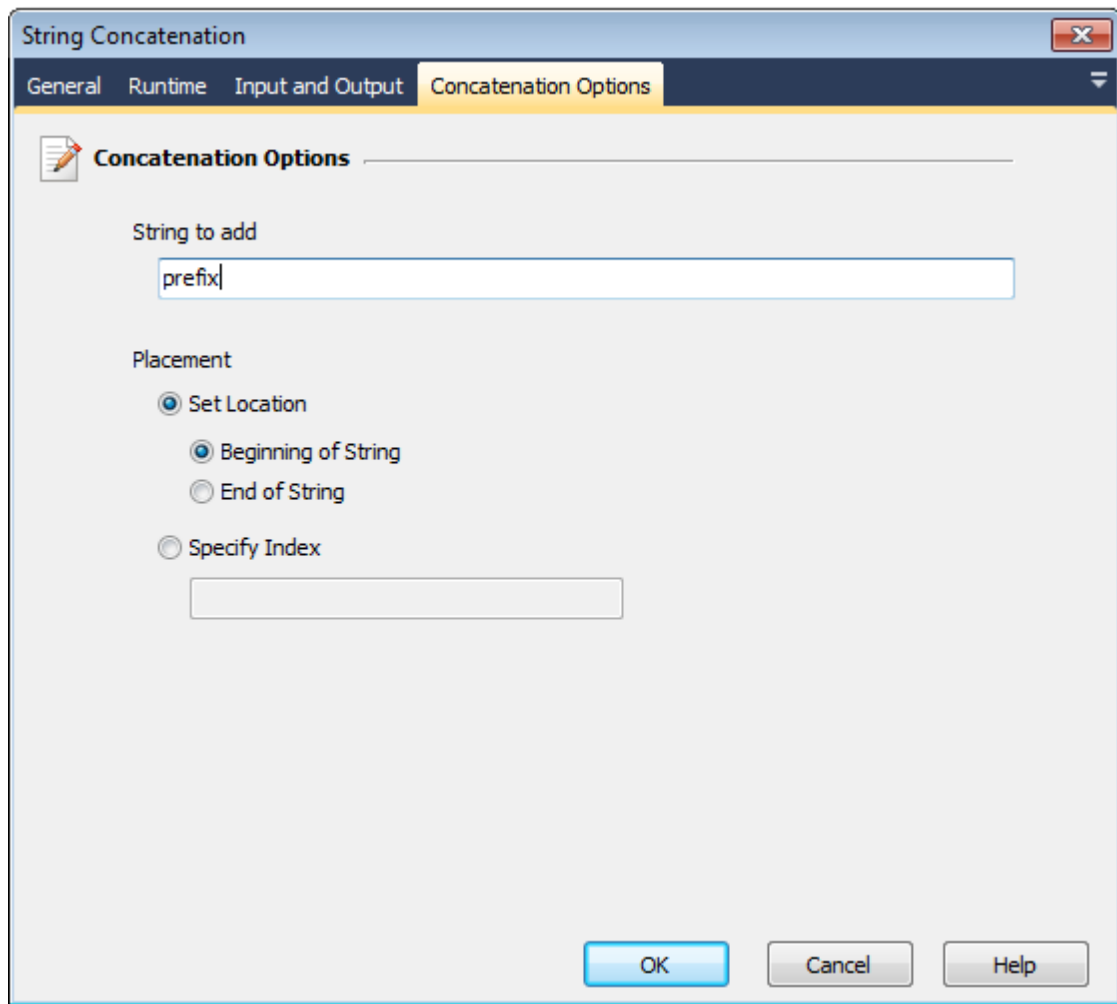
property	InputString	: string;	// The input string, if not using a variable
property	OutputVariable	: string;	// The Output Variable name
property	ApplyToInput	: boolean;	// Apply the changes to the input variable, only valid if using an input variable
property	UsingInputVariable	: boolean;	// True if 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.
```

6.31.5 String Concatenation

The String Concatenation action allows you to join two strings together.

**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 than 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

```

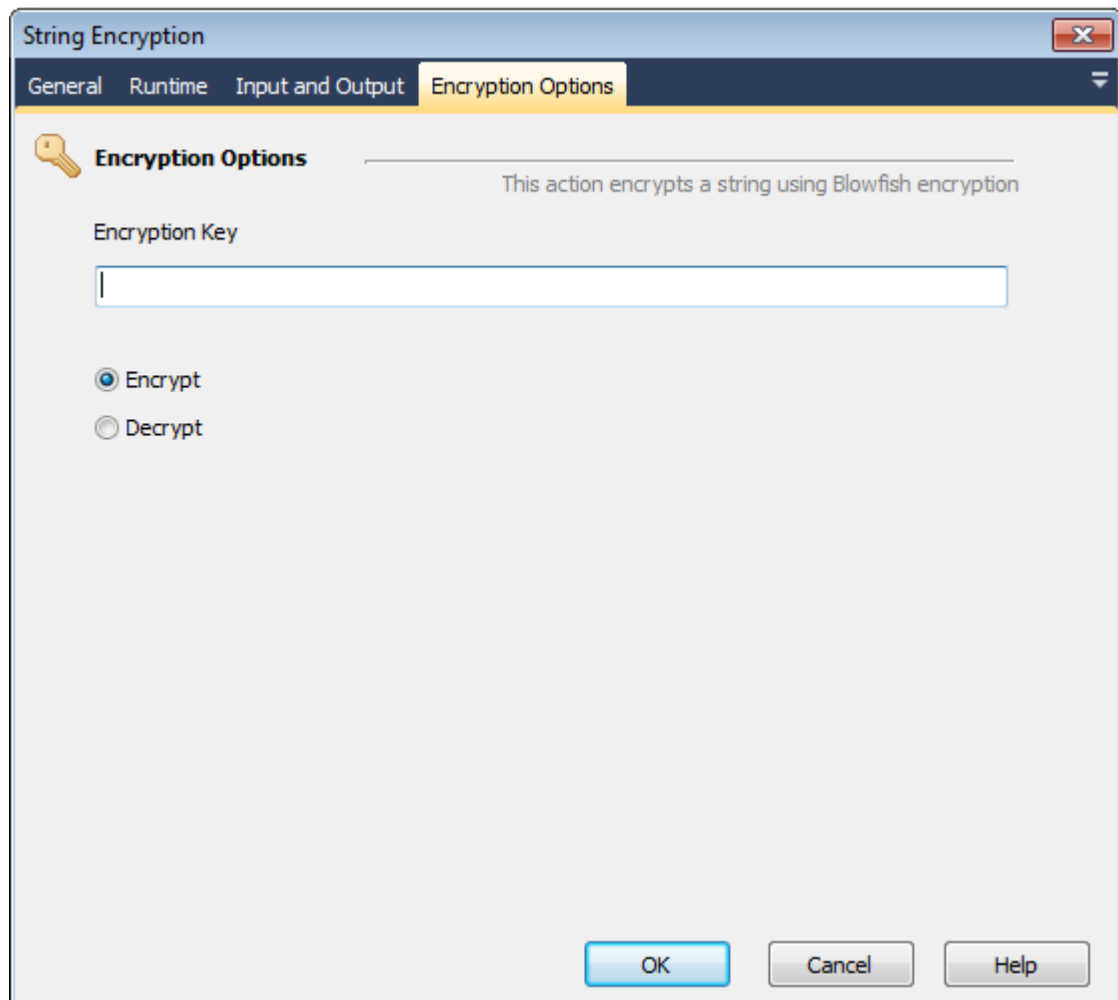
    property ApplyToInput      : boolean;    // Apply the changes to the input
variable, only valid if using an input variable
    property UsingInputVariable : boolean;    // True if using an input variable, not
input string
    property InputVariable     : string;      // Input variable name

    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.

```


6.31.6 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_\(cipher\)](http://en.wikipedia.org/wiki/Blowfish_(cipher))



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

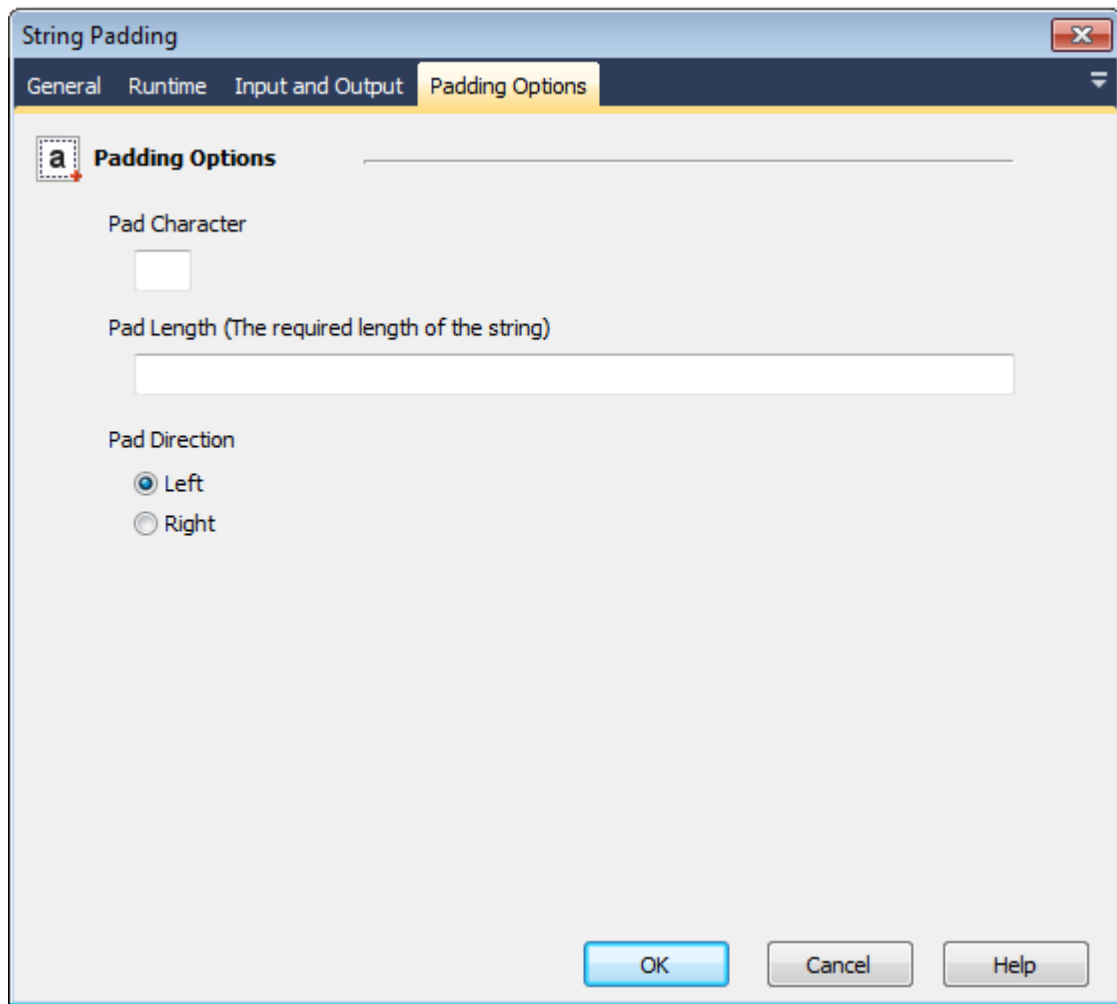
property InputVariable : string; // Input variable name

property EncryptionKey : string; // The encryption key

property Encrypt : boolean; // True to encrypt, false to decrypt

6.31.7 String Padding

The String Padding action allows you to pad a string to a specified length.



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 than 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 :

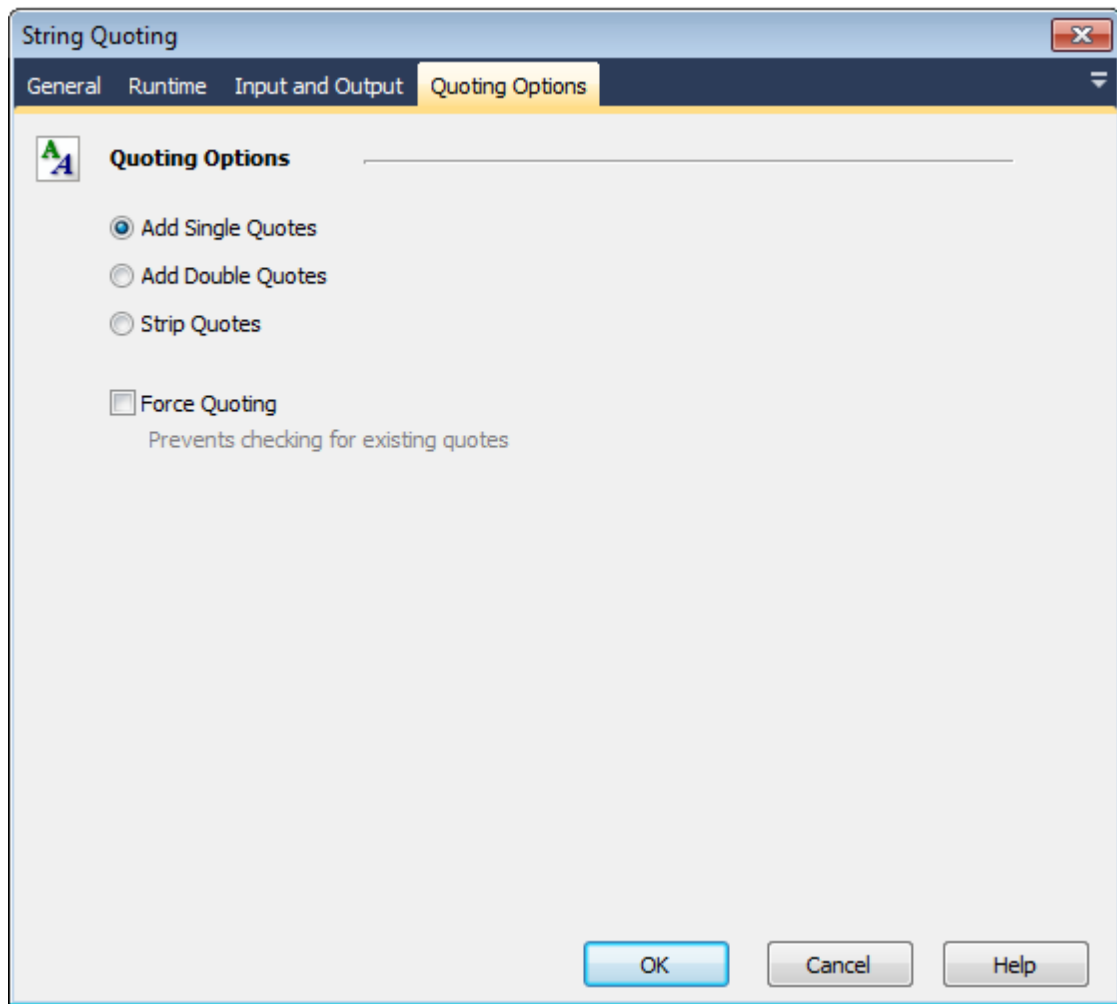
property variable	InputString	: string;	// The input string, if not using a
property	OutputVariable	: string;	// The Output Variable name
property	ApplyToInput	: boolean;	// Apply the changes to the input

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

6.31.8 String Quoting

The String Quoting action allows you to add or remove quotes from a string.



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 :

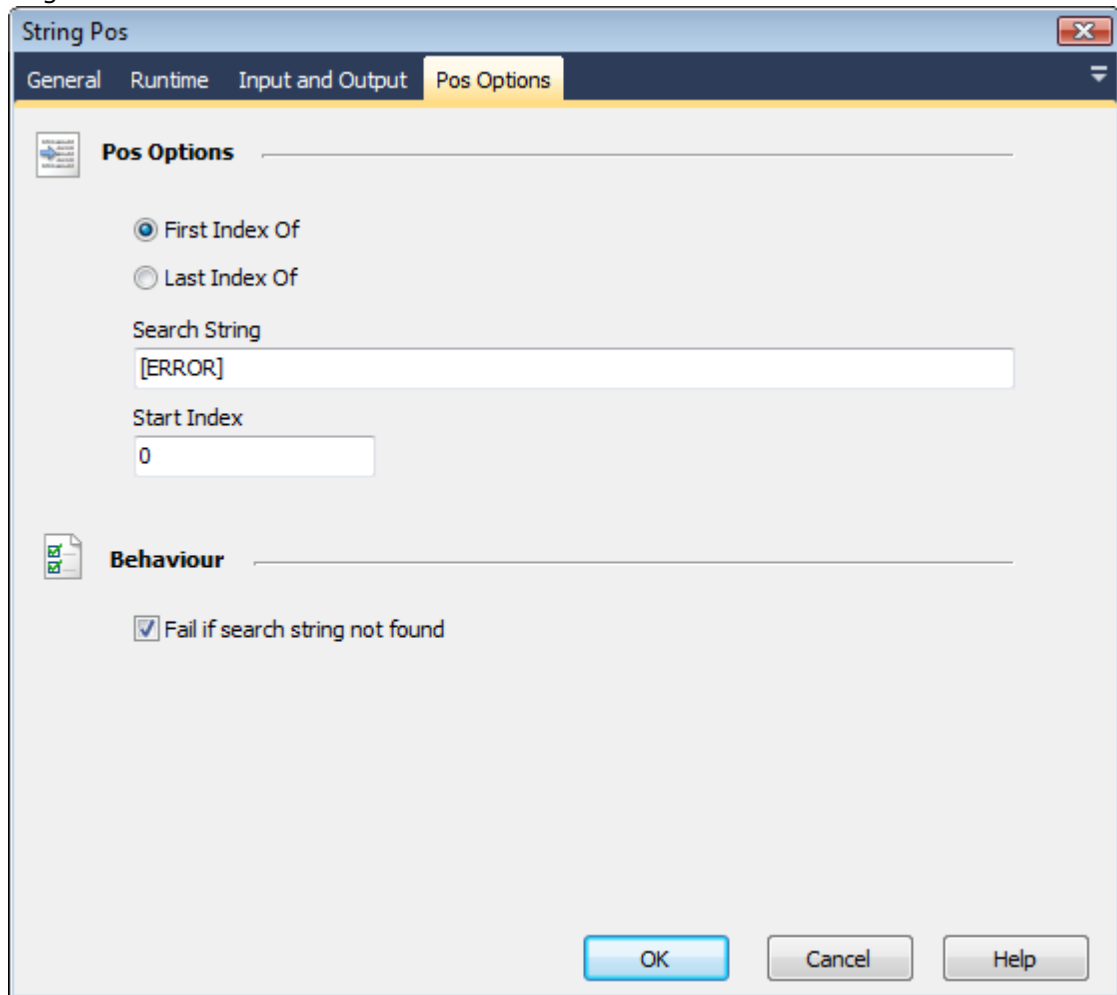
property InputString	: string;	// The input string, if not using a
variable		
property OutputVariable	: string;	// The Output Variable name

```
    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
```

6.31.9 String Pos

The String Pos action allows you to find the index of a string or character within a string.



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 search for within the
input string

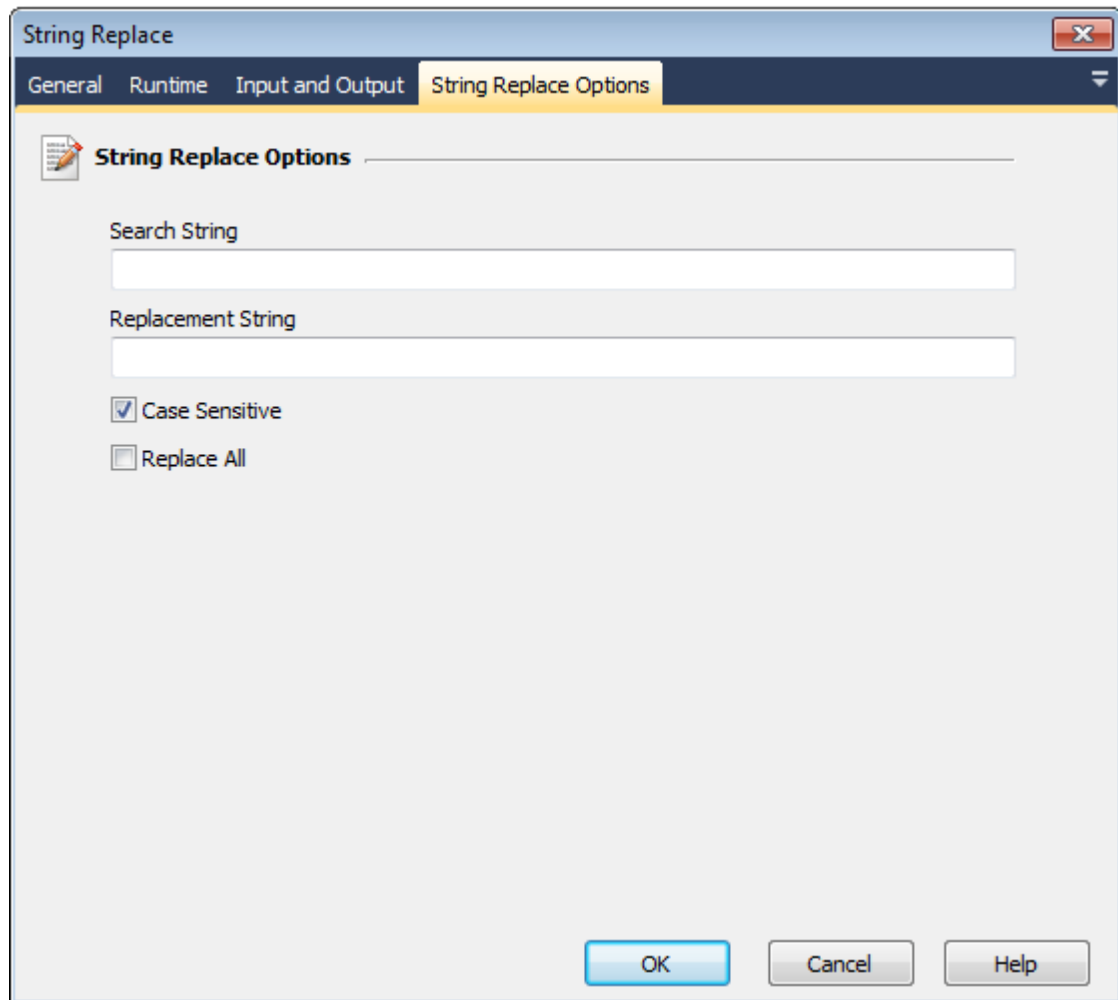
property StartIndex : string // The position to start the search
from

property FailIfNotFound : boolean // Fail action if search string not found
within input

6.31.10 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.



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

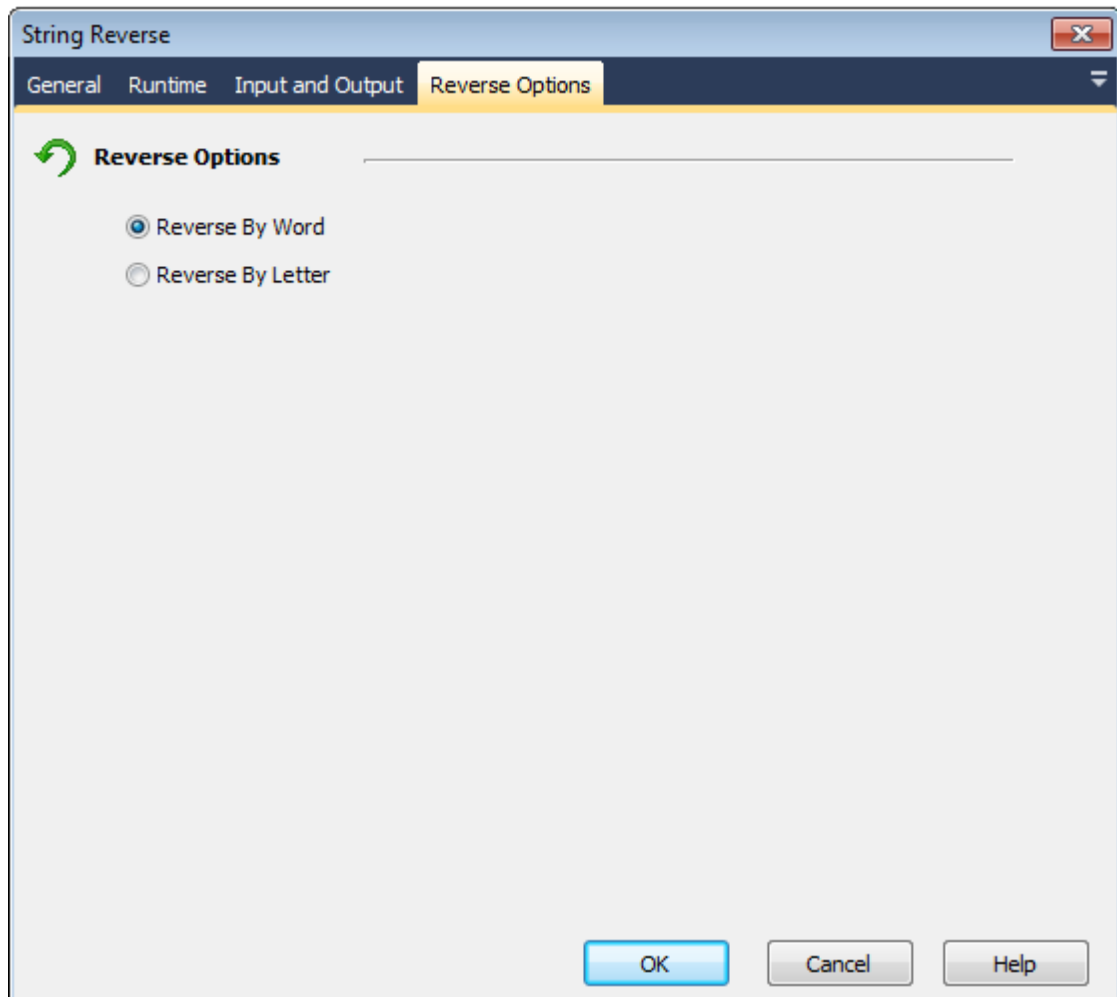
search, otherwise false

property ReplaceAll : boolean; // True to iterate through entire string,

otherwise false

6.31.11 String Reverse

The String Reverse action enables you to reverse a string. The string can be reversed by words or letters.



Reverse by Word

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

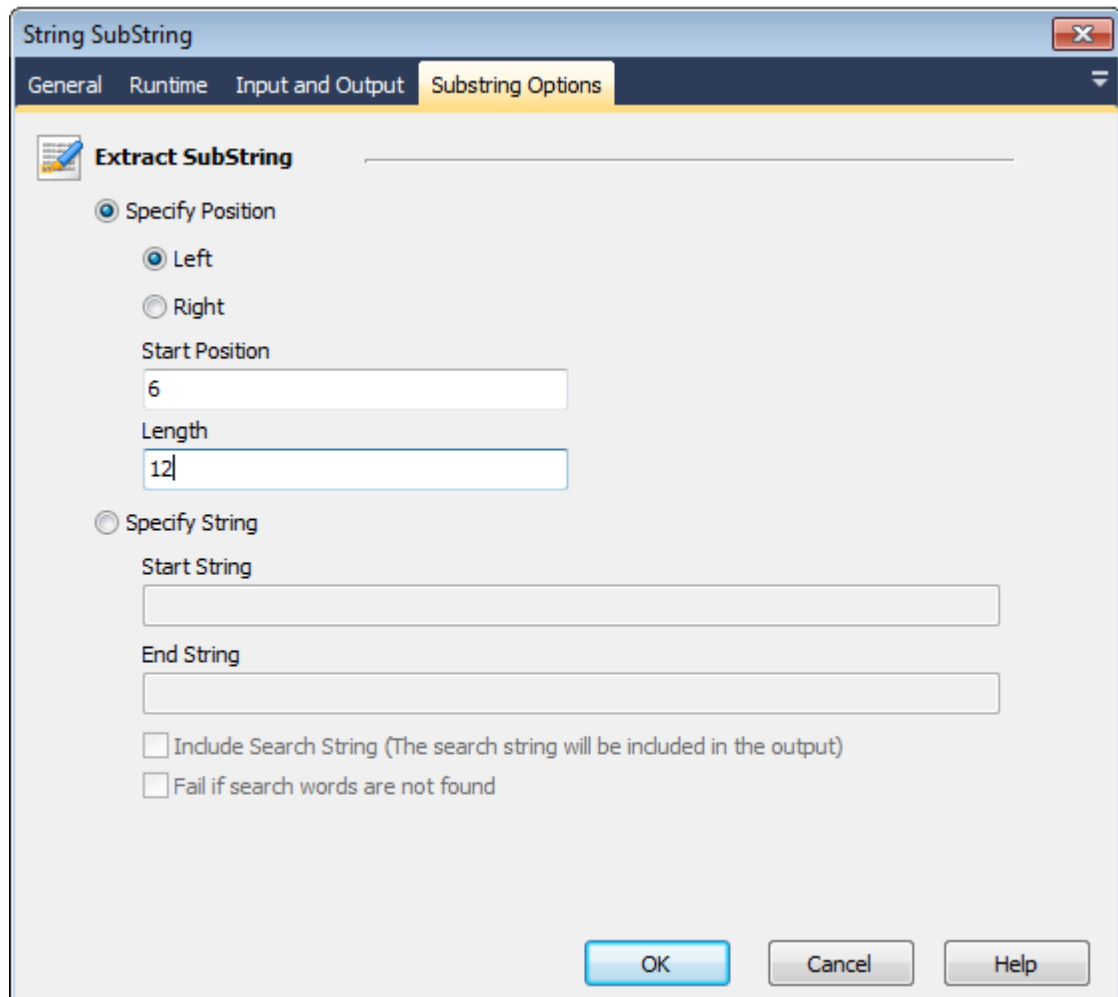
The Action properties available are :

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
```

6.31.12 String SubString

The String SubString action enables you to extract part of a string, where it can be used in subsequent actions.



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 whether 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			

```

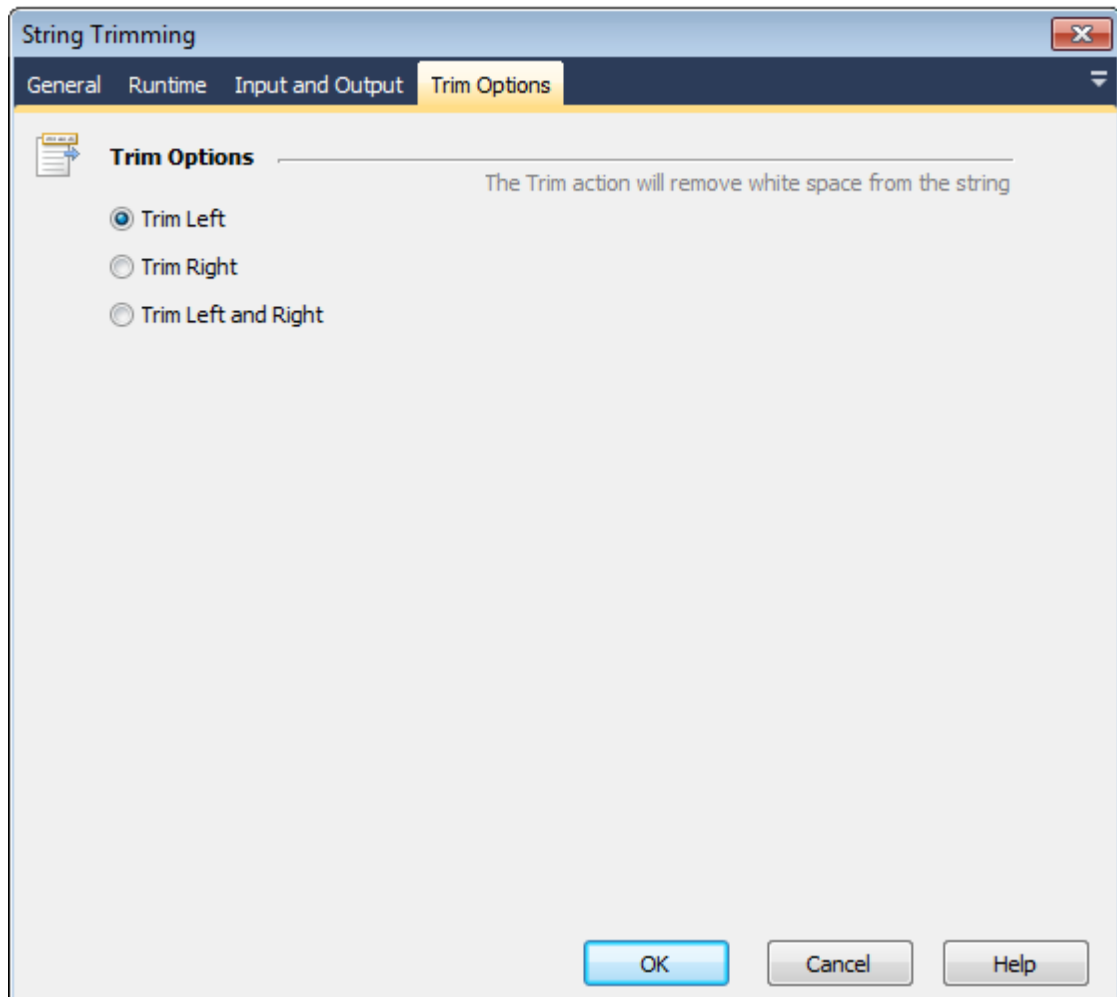
property OutputVariable      : string;      // The Output Variable name
property ApplyToInput       : boolean;      // Apply the changes to the input
variable, only valid if using an input variable
property UsingInputVariable : boolean;      // True if using an input variable, not
input string
property 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

```

6.31.13 String Trimming

The String Trimming Action allows you to remove white space from the beginning or end of a string.



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

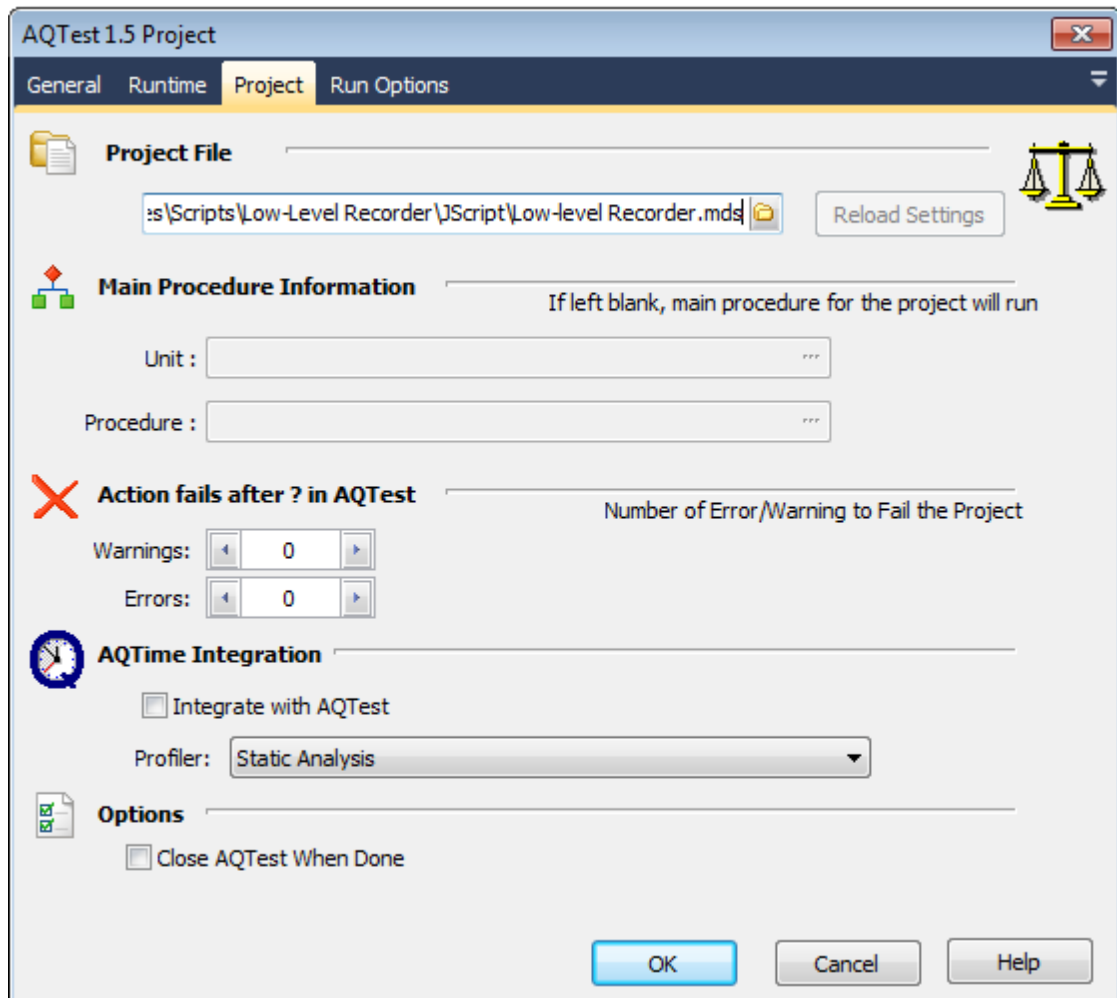
property InputVariable : string; // Input variable name

property TrimType : string; // Valid values: Left, Right, All

6.32 Testing Tools

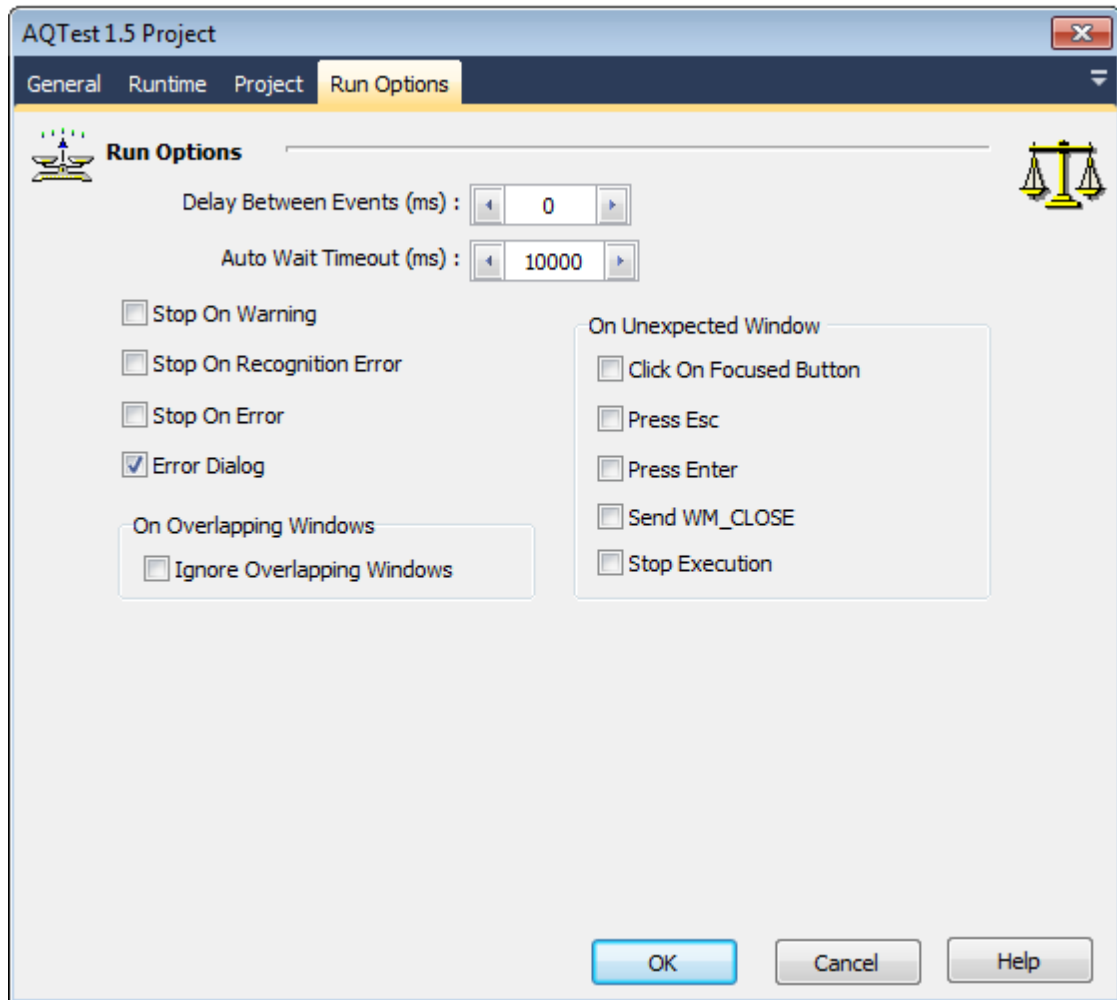
6.32.1 AQTest

This action integrates Automated QA's AQTest automated testing tool with FinalBuilder. This makes it possible to perform Regression testing as part of the build process.



Project File : The fully qualified path to the AQTest project file (.mds).

Close AQTest When Done : This action will start AQTest if it is not already loaded. If you are only doing one test, you can check this option to have FinalBuilder close AQTest when it is done testing.



For a full description of the options, please consult the AQTest 1.5 manual.

Scripting Info

The Action properties available are :

property ProjectFile : WideString
property CloseWhenDone : WordBool;

This action was written by Eric Holton and VSoft Technologies. AQTest can be found at <http://www.automatedqa.com>

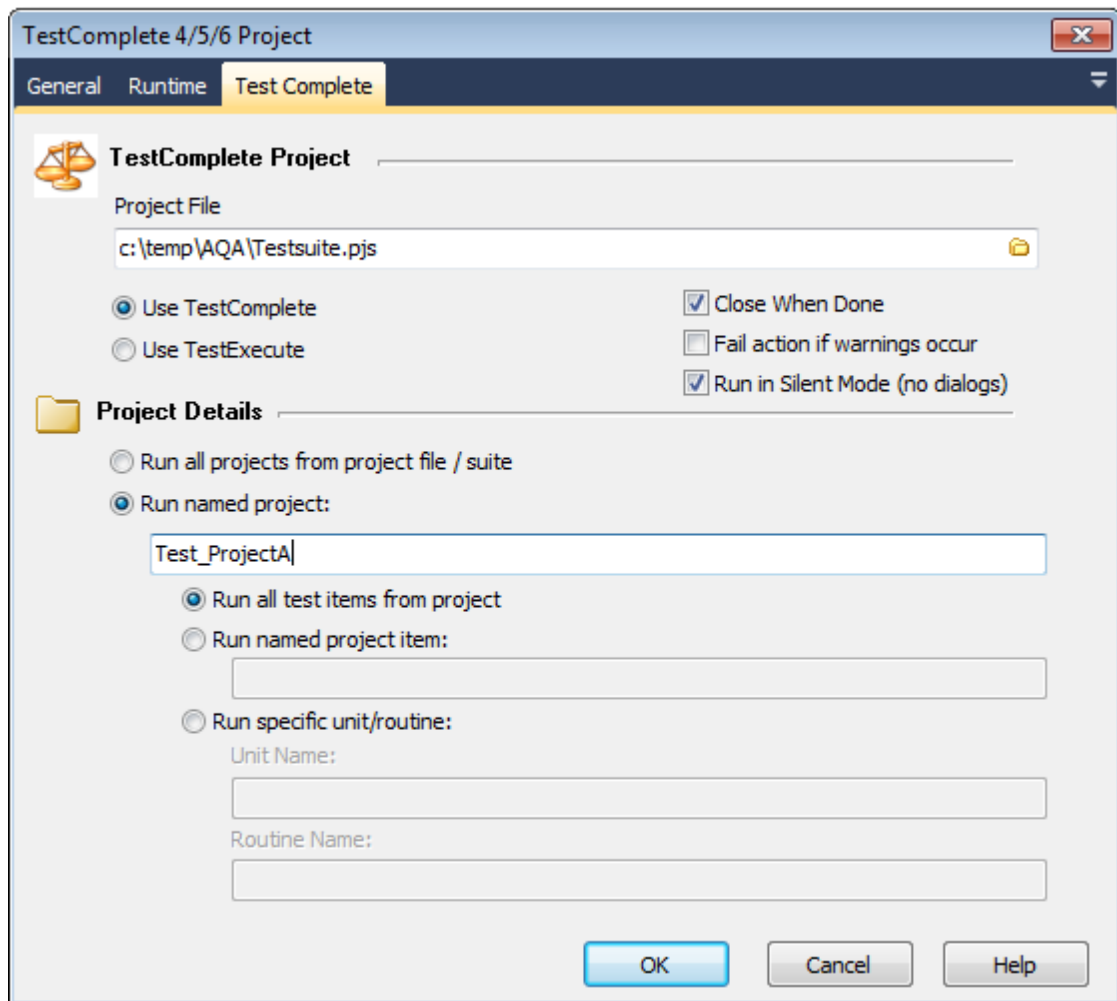
6.32.2 AutomatedQA TestComplete 2.x/3.x Actions

These Actions provide integration with AutomatedQA's TestComplete 2.x & 3.x products. They were kindly provided by Eric Holton - <http://www.holtonsystems.com>

6.32.3 AutomatedQA TestComplete 4 and onwards

The Test Complete 4/5/6 Project action provides support for execution of AutomatedQA TestComplete projects for version 4 and newer.

The path to your TestComplete installation should be automatically detected. If it is not, you will need to configure it under Tools menu -> Options -> Testing Tools.



TestComplete Project

The full path to a TestComplete project or project suite.

Use TestComplete / TestExecute

Choose whether to run the project in TestComplete or TestExecute. The chosen application must be installed, and the path to the executable set under the Tools menu -> Options -> Testing Tools -> TestComplete.

Close When Done

Select to close TestComplete when the test is completed. Note that TestComplete does not output verbose status information to the FinalBuilder log, so you will need to check the TestComplete log file for details.

Fail action if warnings occur

TestComplete returns four levels of error information:

- No errors (test completed successfully.)
- Warnings (test completed with warnings.)
- Errors (errors were detected.)
- Fatal Errors (a fatal error prevented the test script from running.)

Normally, the action will fail on all but "No errors." If this option is selected, TestComplete will only fail if errors were encountered.

Run in silent mode

If this option is selected, TestComplete will not display warning, hint or error dialogs while running. Recommended if a build is to be run unattended.

Project Details:

Run all projects / Run named project

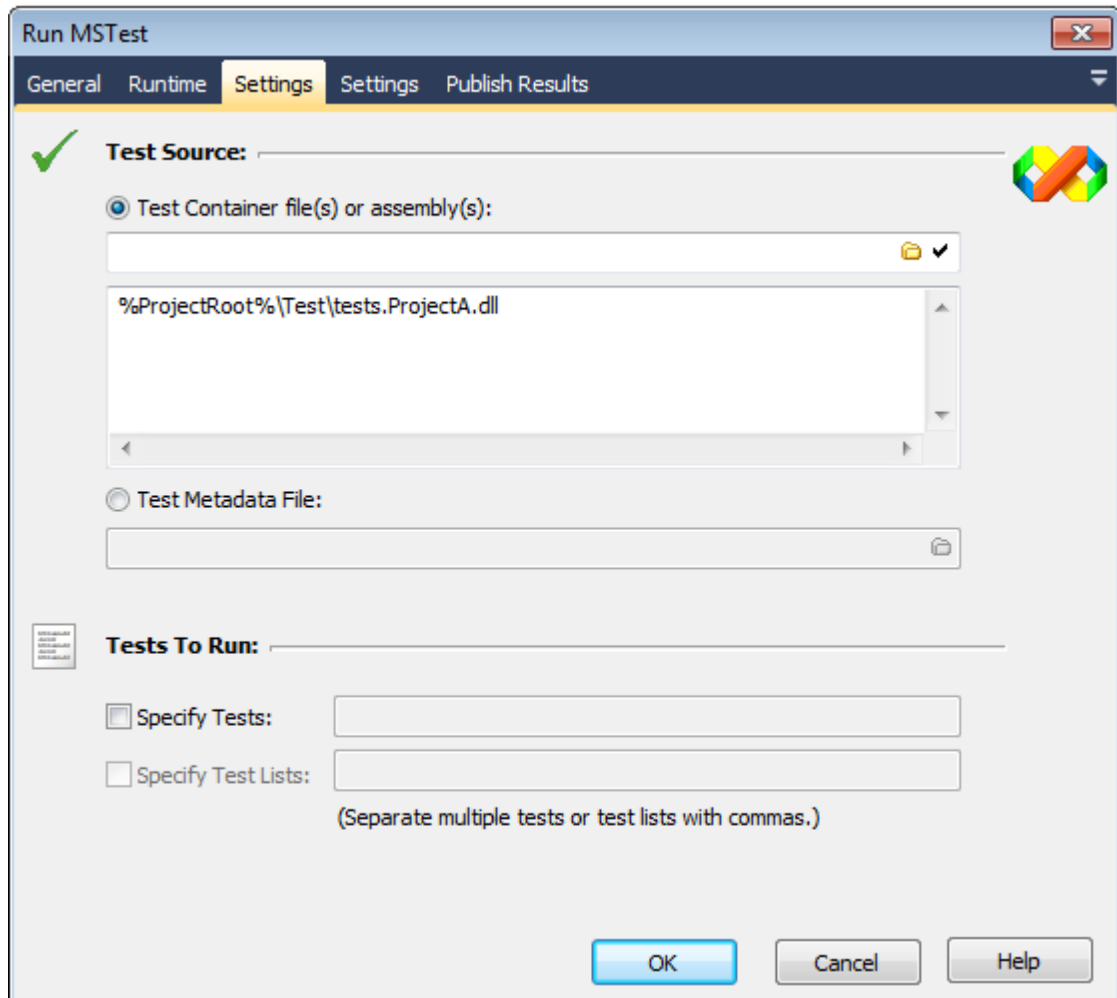
You can choose to run one test project or all projects from the project suite.

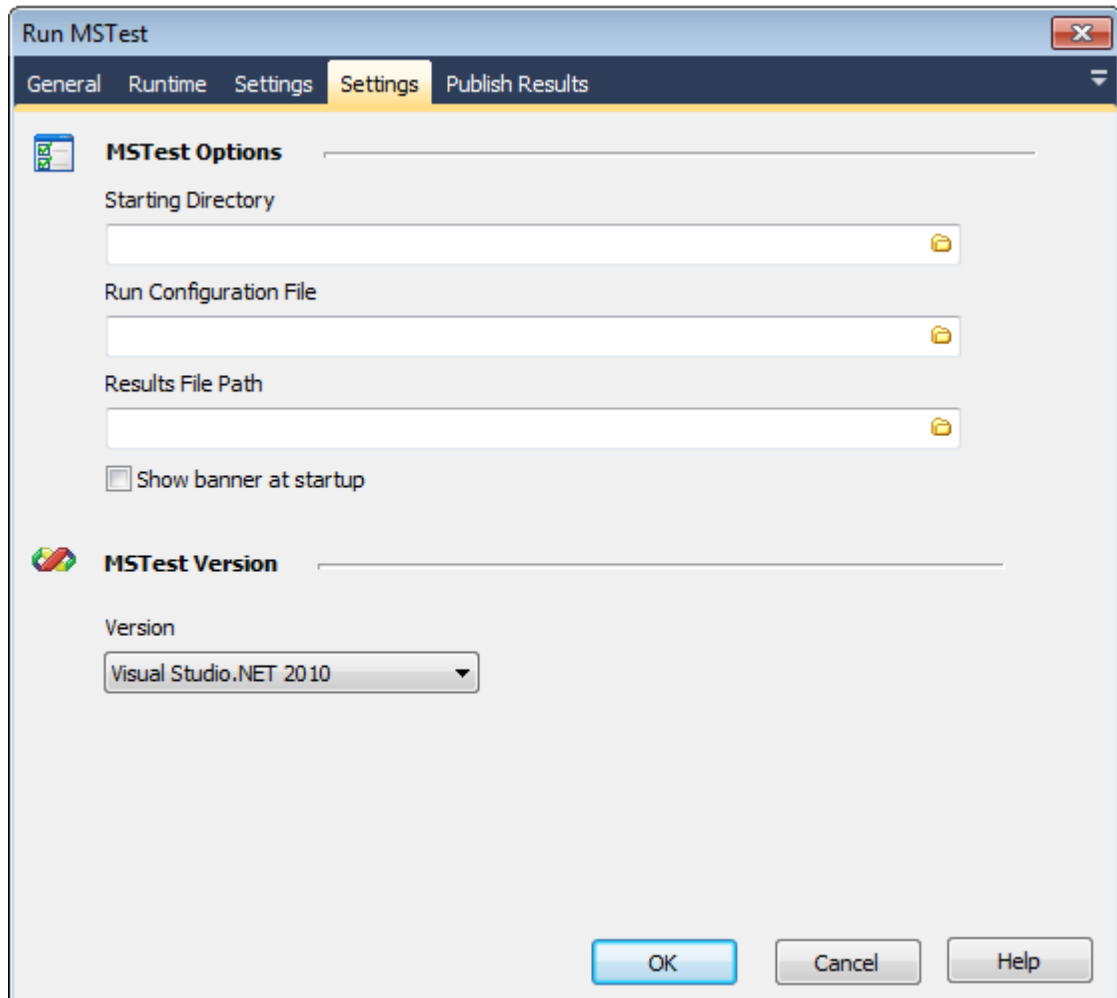
Run all test items / Run named project item / Run specific unit/routine

If running a named project, you can select to run all items in the suite, a single named "project item" from the suite, or a specific test script routine from a single test script unit.

6.32.4 MSTest

The Run MSTest action allows you to run tests created with the Visual Studio MSTest framework.





Test Source

You can choose to load tests from either an assembly containing tests or a test metadata file. A test metadata file is an XML file which is created when you use Test Manager to create tests for a Visual Studio solution file.

MSTest Version

Choose the applicable version of MSTest for your Visual Studio version. The Run MSTest action uses the path to Visual Studio given under Tools -> Options -> Visual Studio.NET.

There is also the option to choose "Standalone MSTest" version, if your copy of MSTest is not installed as part of a Visual Studio installation. The standalone path is also set from the Visual Studio options.

Tests to Run

By default, all tests in the container/metadata file are executed. Optionally, you can specify one or more test names or one or more test list names to run. Test lists only

apply when running tests from a test metadata file.

Run Configuration File

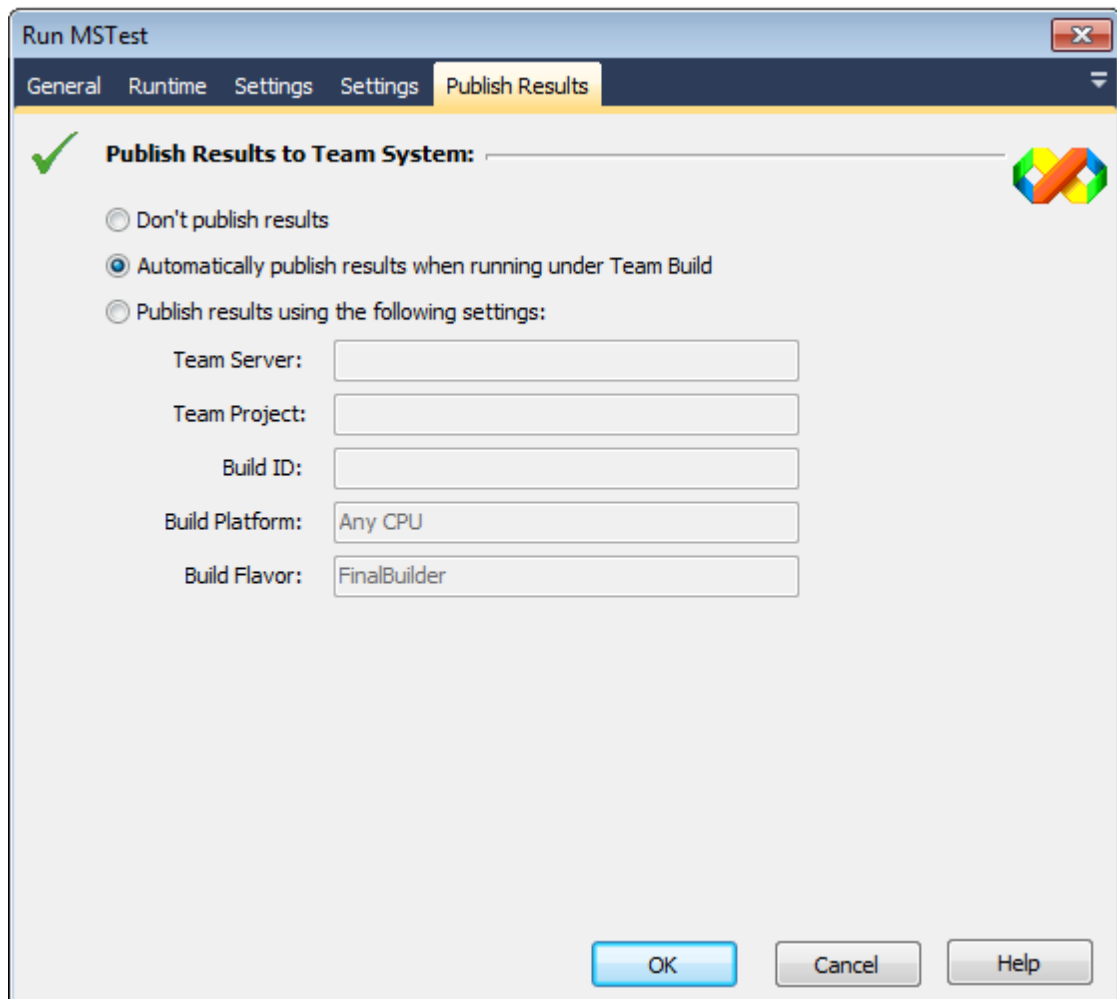
You can optionally specify a .testrunconfig file containing test run configuration details.

Results file path

You can optionally specify a .TRX results file path for the result details. This file can be opened and viewed in Visual Studio.

Show banner at startup

If this option is enabled, the MSTest banner will be logged when the action is run.



The options on the "Publish Results" are for publishing test results to a Team Foundation Server. By default, results are automatically published, but only if

FinalBuilder detects it is being run as part of a Team Build process.

Alternatively, you can choose "Don't publish results" to never publish results to TFS.

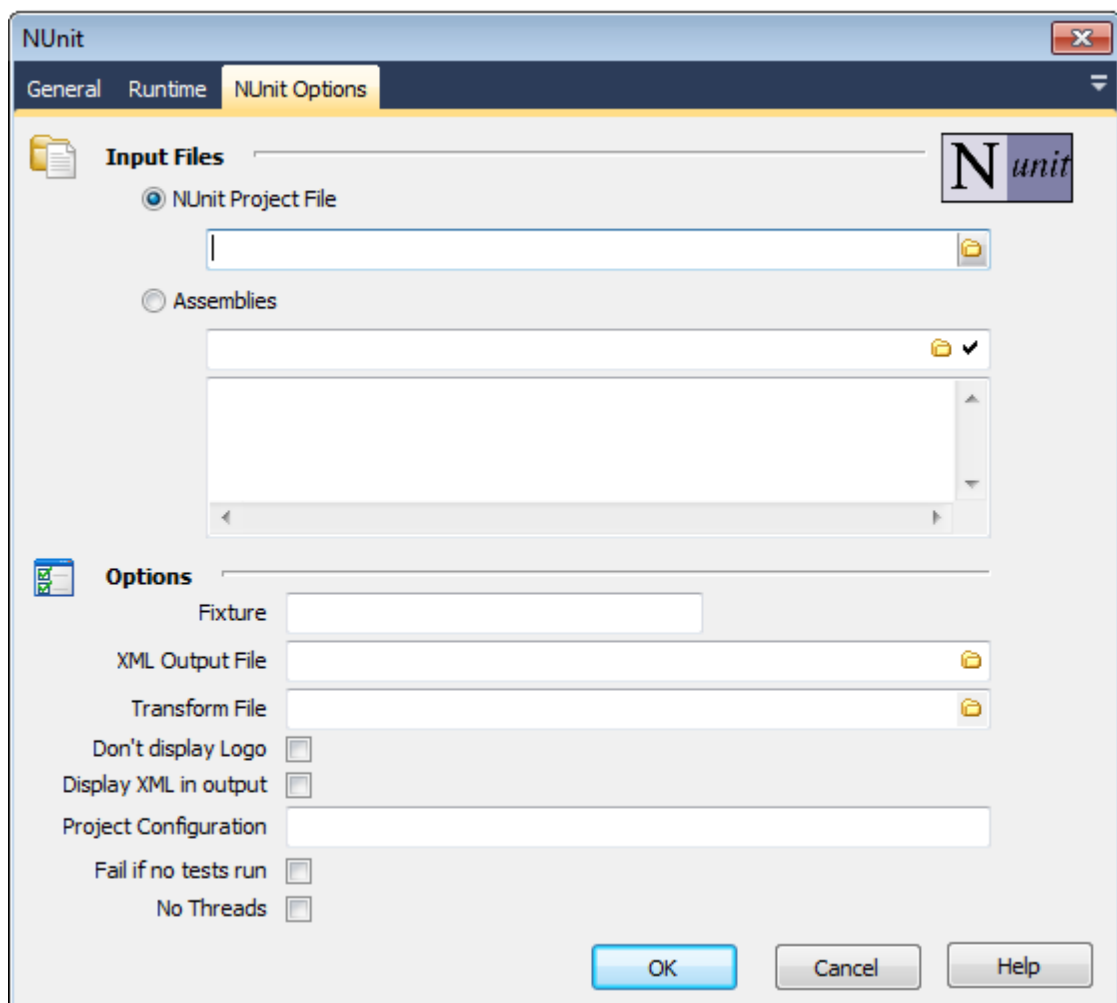
Finally, you can optionally specify explicit settings to have results published to a specific TFS server, for a particular team project and team build ID.

6.32.5 NUnit Action

NUnit is a unit-testing framework for all .Net languages. Initially ported from JUnit, the current version, 2.1 is the third major release of this xUnit based unit testing tool for Microsoft .NET. It is written entirely in C# and has been completely redesigned to take advantage of many .NET language features, for example custom attributes and other reflection related capabilities. NUnit brings xUnit to all .NET languages.

For more information see the NUnit homepage:

<http://www.nunit.org/>



Assemblies

The individual assemblies which contain the TestFixtures to be executed.

Fixture

The full name of the TestFixture to run. A namespace can be provided without the class name, in this instance all the TestFixtures under the namespace will be tested.

XML Output File

The XML file to output the results to.

Transform File

The path to a transform file, which will be used to display the results to the build log. If 'Display XML in Output' is enabled, the results will not be transformed.

Project Configuration

The name of the configuration for NUnit to run. If no project is specified NUnit will load the first configuration found as the default. This option has no effect if a Project File is not being used.

NUnit Options (in Tools menu | Options)

NUnit Location - set the location to NUnit-console.exe

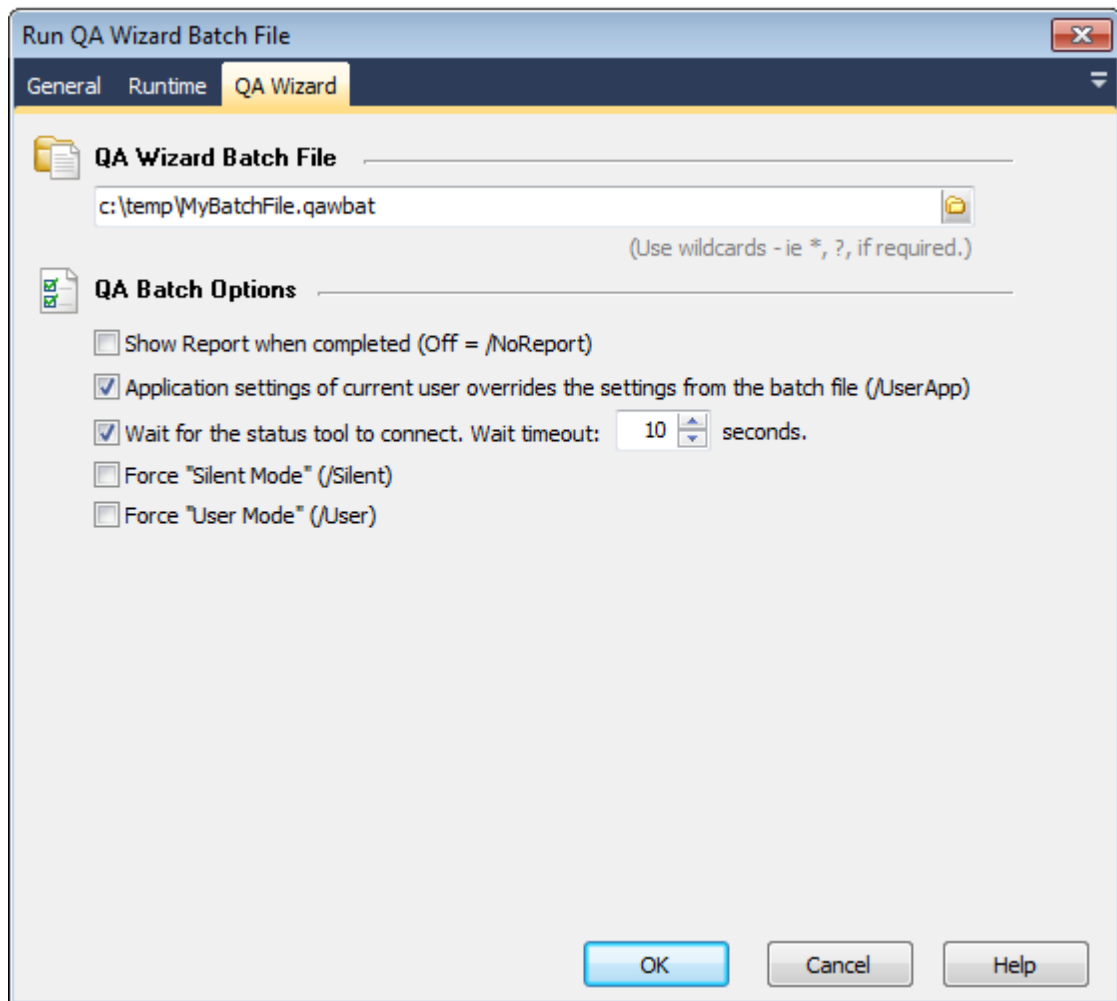
Normally located in:

C:\Program Files (x86)\NUnit-Net-2.0 2.2.8\bin\nunit-console.exe

6.32.6 Run QA Wizard Batch File

The Run QA Wizard Batch File provides a frontend to QABatch in order to run QA Wizard batch files.

(To create QA Wizard batch files, go to the Script menu in QA Wizard and choose Manage Batch Files. See [Seapine KB Article 39](#) for details.)



QA Wizard Batch File

Enter the path to a QA Wizard batch file (see above for details on creating these.) Use wildcards (ie *, ?) to specify multiple batch files.

Show Report When Completed

If this option is selected, QA Batch will open a web browser to view the status report when the tests finish. If not selected, this option is equivalent to the /NoReport switch.

Application Settings of Current User Overrides the Settings from the Batch File

Use this option to override options from the batch file with the user's default settings in QA Wizard. Equivalent to the /UserApp switch.

Wait for the status too to connect

If this option is set, QABatch will wait the specified number of seconds for the QA Wizard Status Tool (tray icon) to establish a connection. Choose this option if you would like the QA Wizard status to be shown on the status bar, disable it otherwise.

Force "Silent Mode"

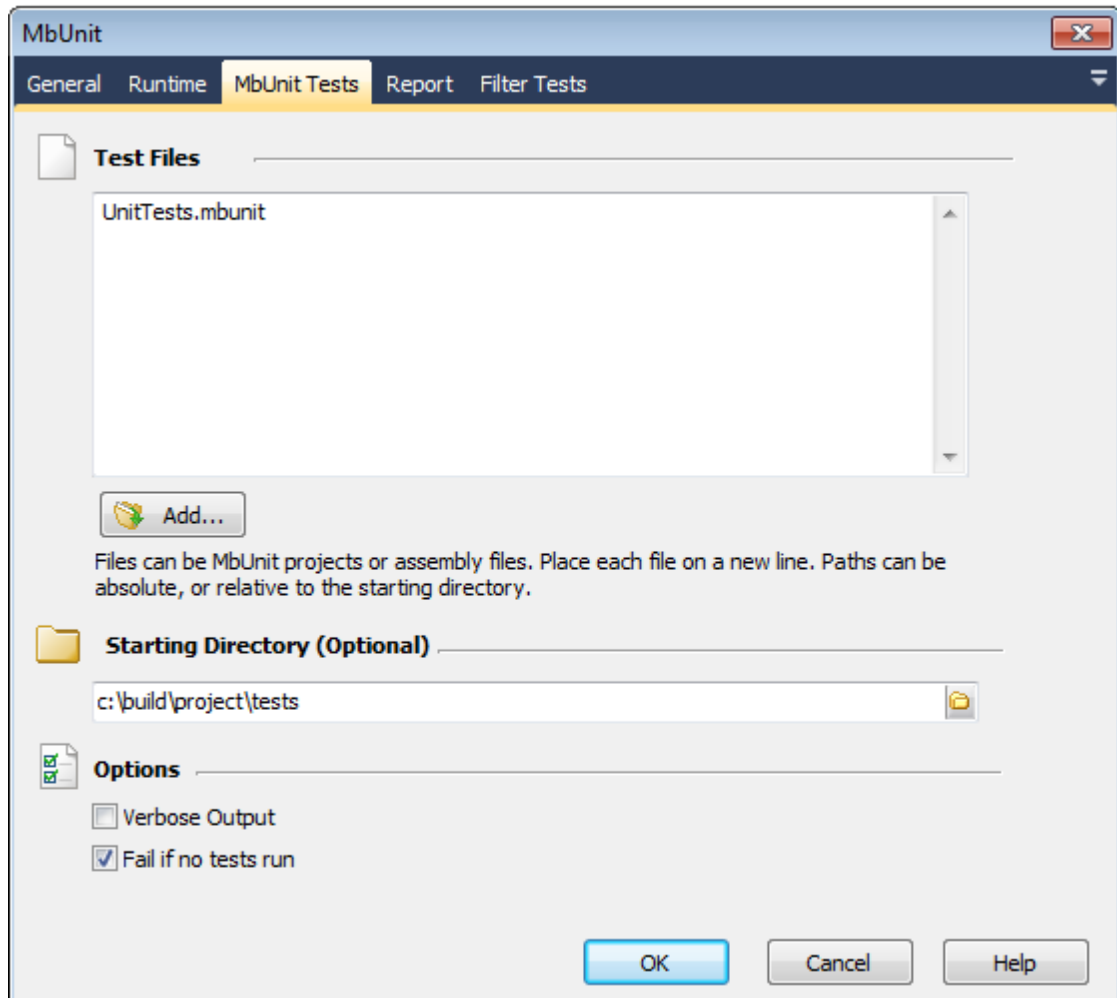
Forces the batch scripts to start in "silent mode". Warning : On our test machine this option caused some scripts to hang QABatch. Equivalent to the /Silent switch. Mutually exclusive with "Force User Mode".

Force "User Mode"

Forces the batch script to start in "user" mode. Equivalent to the /User switch. Mutually exclusive with "Silent Mode".

6.32.7 MbUnit Action

The MbUnit action allows you to run tests using the MbUnit test framework.



Test Files

Test files can be MbUnit project files (.mbunit), or assembly files which contain MbUnit test classes.

Place each file name on a new line. Paths can be absolute, or relative to the starting directory (as shown above.)

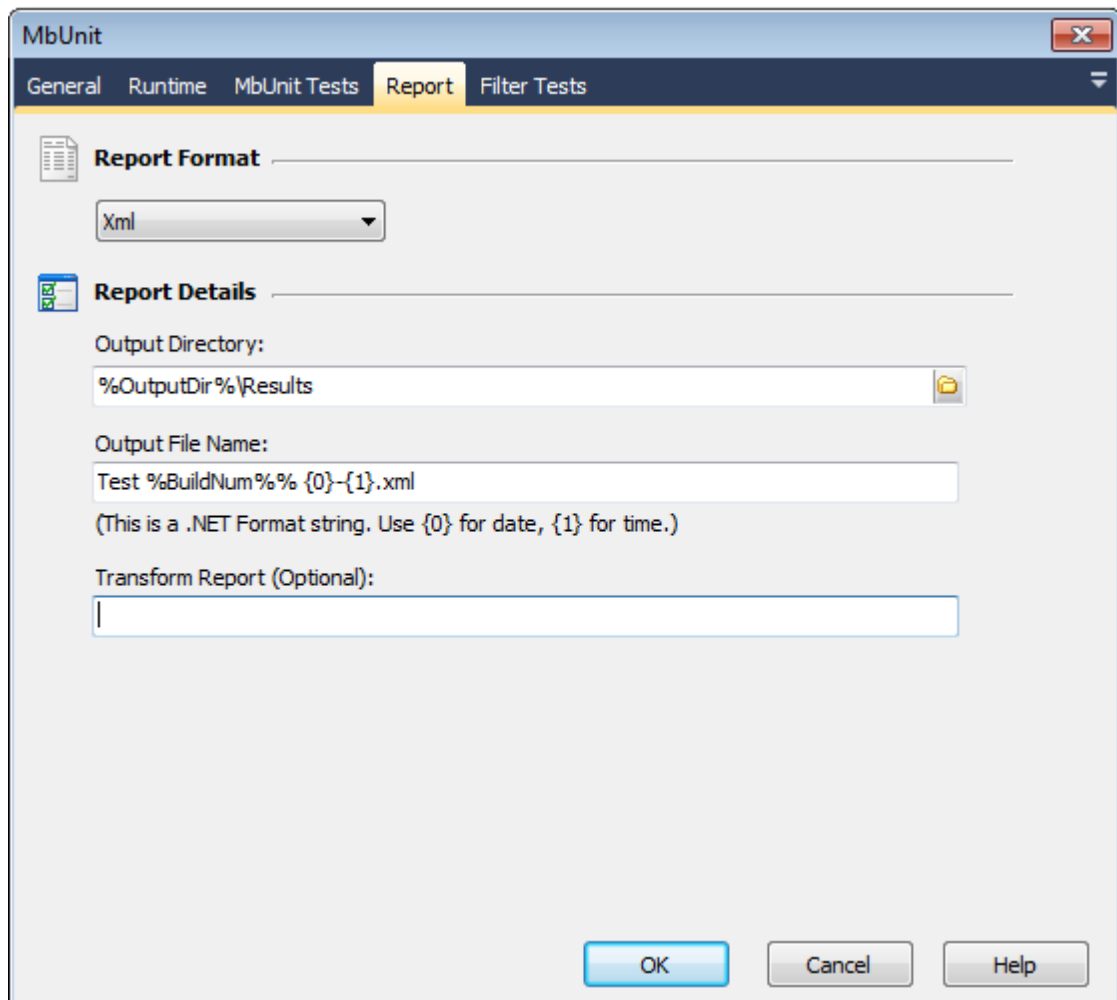
Starting Directory

Path to the starting directory for MbUnit. This allows you to specify relative paths to test files, and it can also help MbUnit resolve Assembly References.

Options

Verbose Output - Enable this option to see extra details about test output.

Fail if no test run - Enable this option to have the action fail if no tests are run (this includes the scenario when assemblies are loaded successfully, but do not contain any tests.)



Report Format

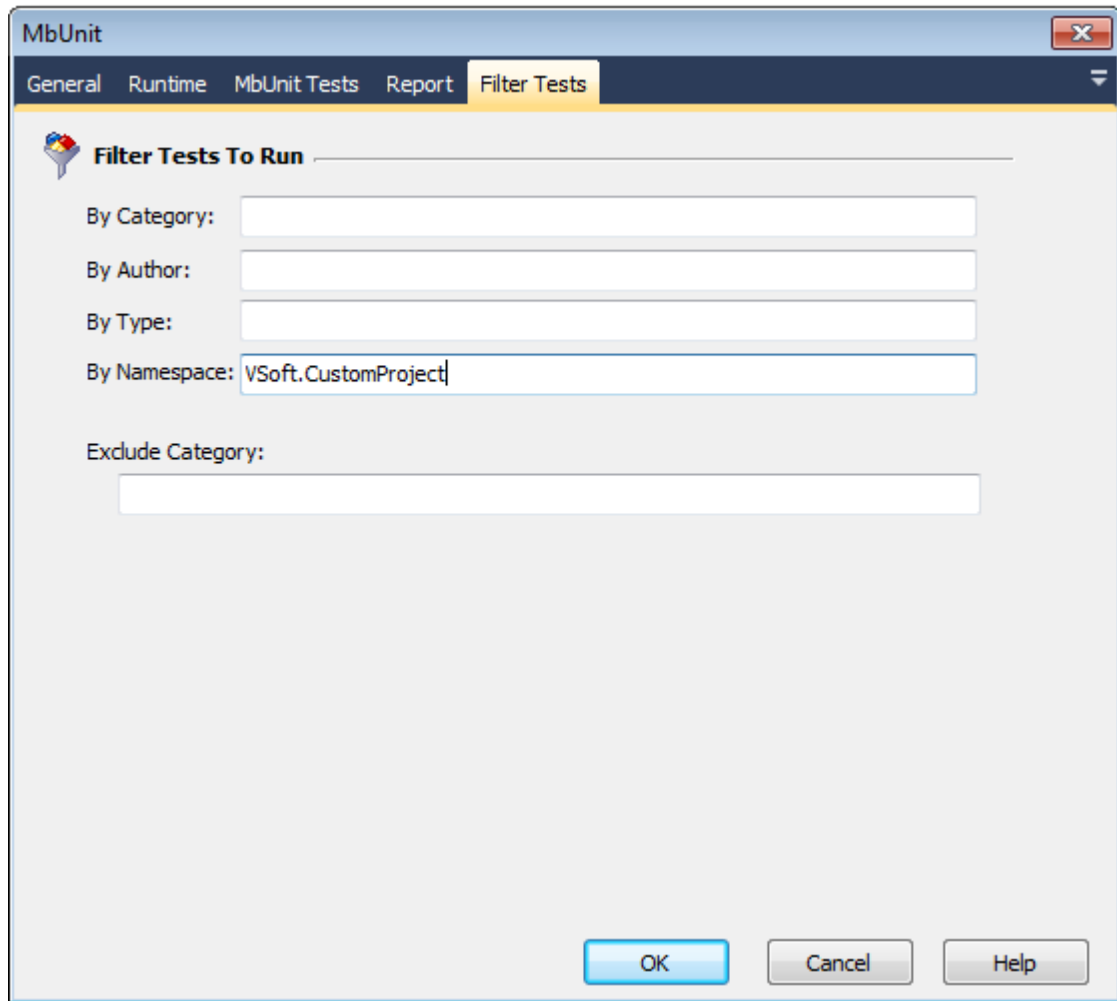
The default Report Format is "None" (no report.) Optionally, you can generate a report in Xml, Html, Text or Dox format.

Report Details

Output Directory - This is the directory where the report will be written.

Output File Name - This is the base file name for the report. This is passed as an argument to the .NET function `System.String.Format()`. Parameter `{0}` is the current date, and `{1}` is the current time. To include a `{` character in the file name, escape it as `{{`.

Transform Report - Optionally specify the path to an XSL file to transform the report output.



All of the above fields are optional.

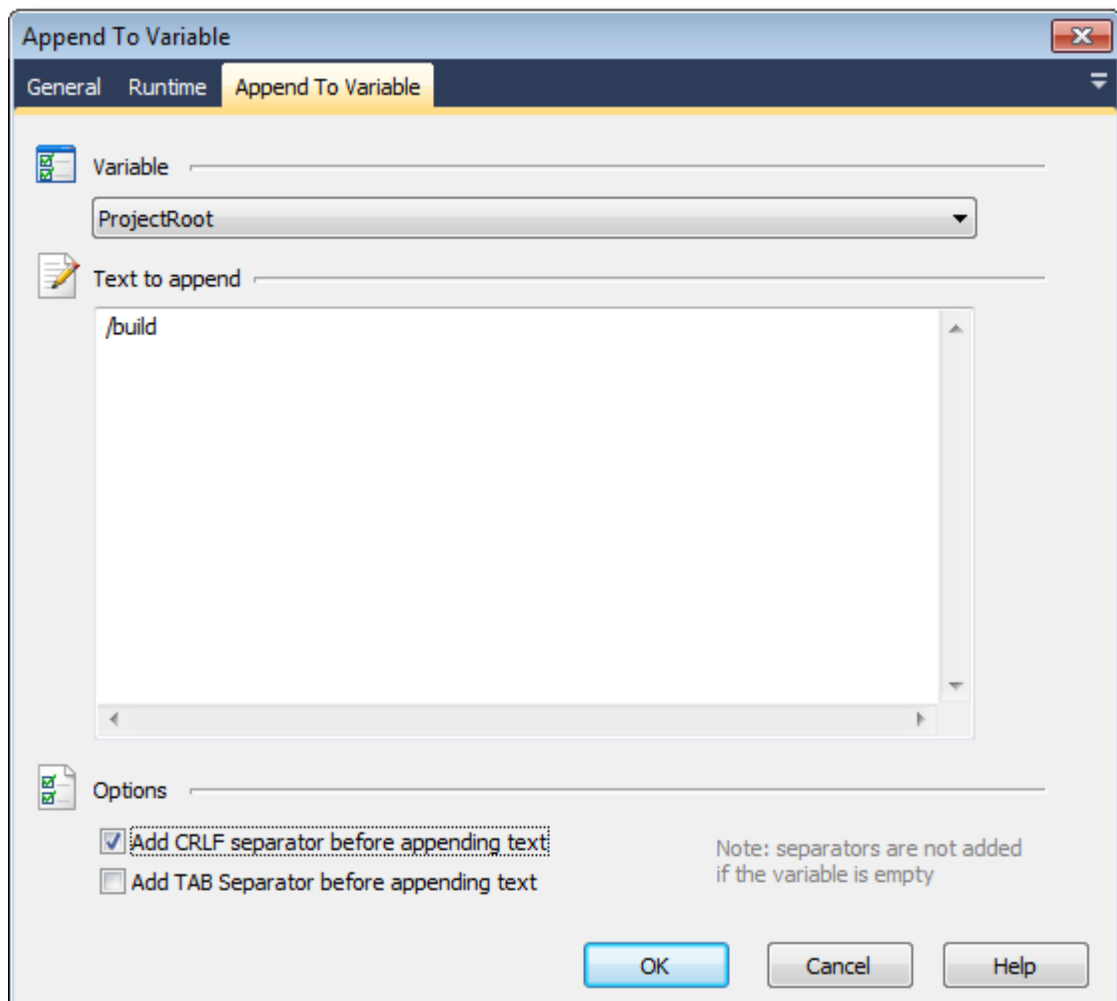
You can filter the tests to run by various parameters. Tests can be filtered by their MbUnit category, or author. Alternatively, filters can be applied based on the type or parent namespace of the test fixture classes. You can also specify a category to exclude from testing.

6.33 Variables

6.33.1 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: 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.

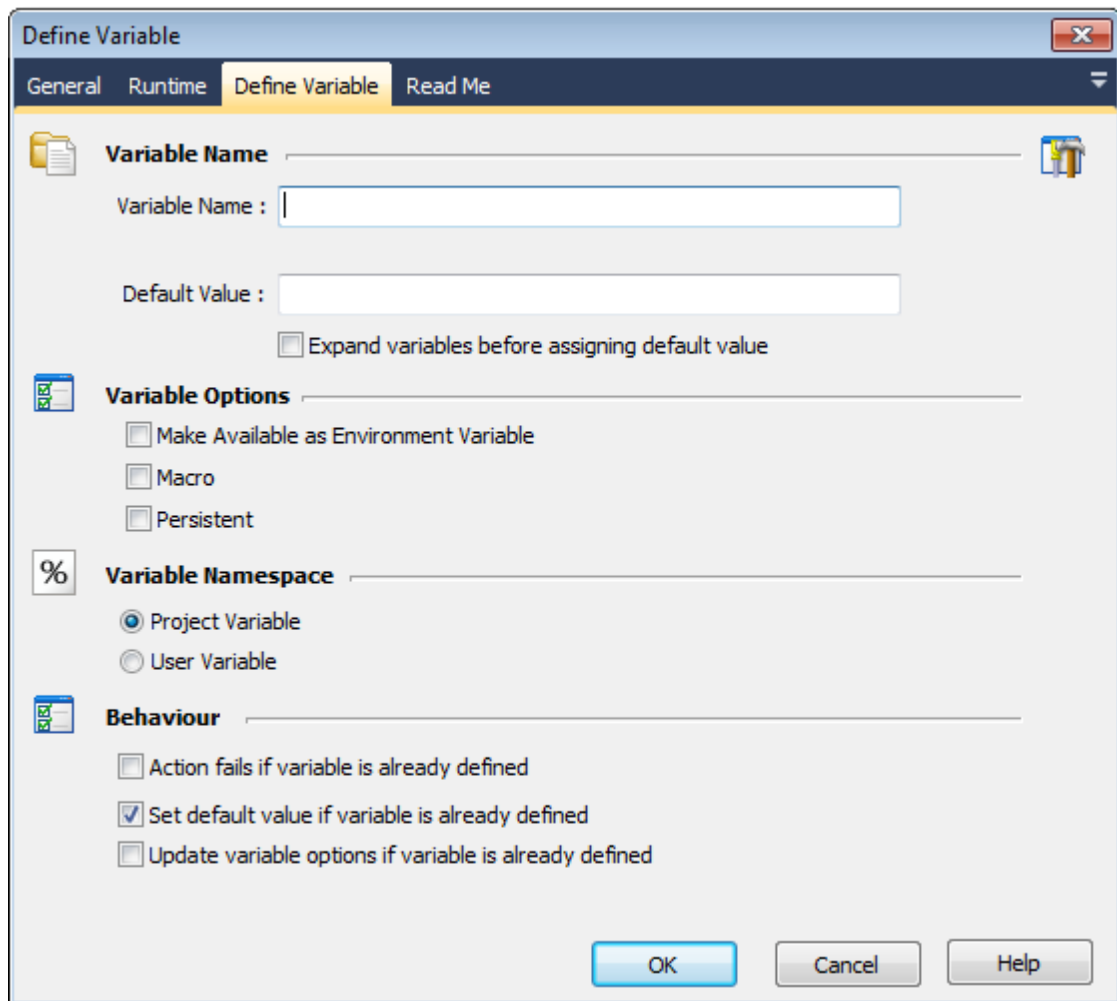
Add CRLF separator: Carriage Return and Line Feed characters will be inserted before the text is appended (if the existing value is not empty.)

Add TAB separator: a tab character will be inserted before the text is appended (if the existing value is not empty.)

6.33.2 Define Variable Action

The define variable action allows you to define a new FinalBuilder variable during the execution of the project. Project or User variables are supported.

IMPORTANT: This is not the recommended way to create FinalBuilder variables for normal use. See the Variables overview topic for details.

The image shows a 'Define Variable' dialog box with a title bar and a close button. It has four tabs: 'General', 'Runtime', 'Define Variable' (which is selected), and 'Read Me'. The 'Define Variable' tab contains several sections. The 'Variable Name' section has a text box for the variable name and a 'Default Value' text box below it. There is a checkbox labeled 'Expand variables before assigning default value'. The 'Variable Options' section has three checkboxes: 'Make Available as Environment Variable', 'Macro', and 'Persistent'. The 'Variable Namespace' section has a percentage icon and two radio buttons: 'Project Variable' (selected) and 'User Variable'. The 'Behaviour' section has three checkboxes: 'Action fails if variable is already defined', 'Set default value if variable is already defined' (checked), and 'Update variable options if variable is already defined'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Variable Name

Choose a name for your new variable. FinalBuilder 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 FinalBuilder to re-evaluate the variable whenever it's referenced during the build, otherwise the value of the variable is evaluated when the build starts and that value is used throughout the build. An Is Macro variable is like a function - its value cannot be set during the build using the Set Variable action or any other means.

Persistent

If checked, the new variable will be persistent (the value will be saved between builds.)

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 build *are* persistent, so if you save a project file after defining a new variable, the new variable will remain 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.

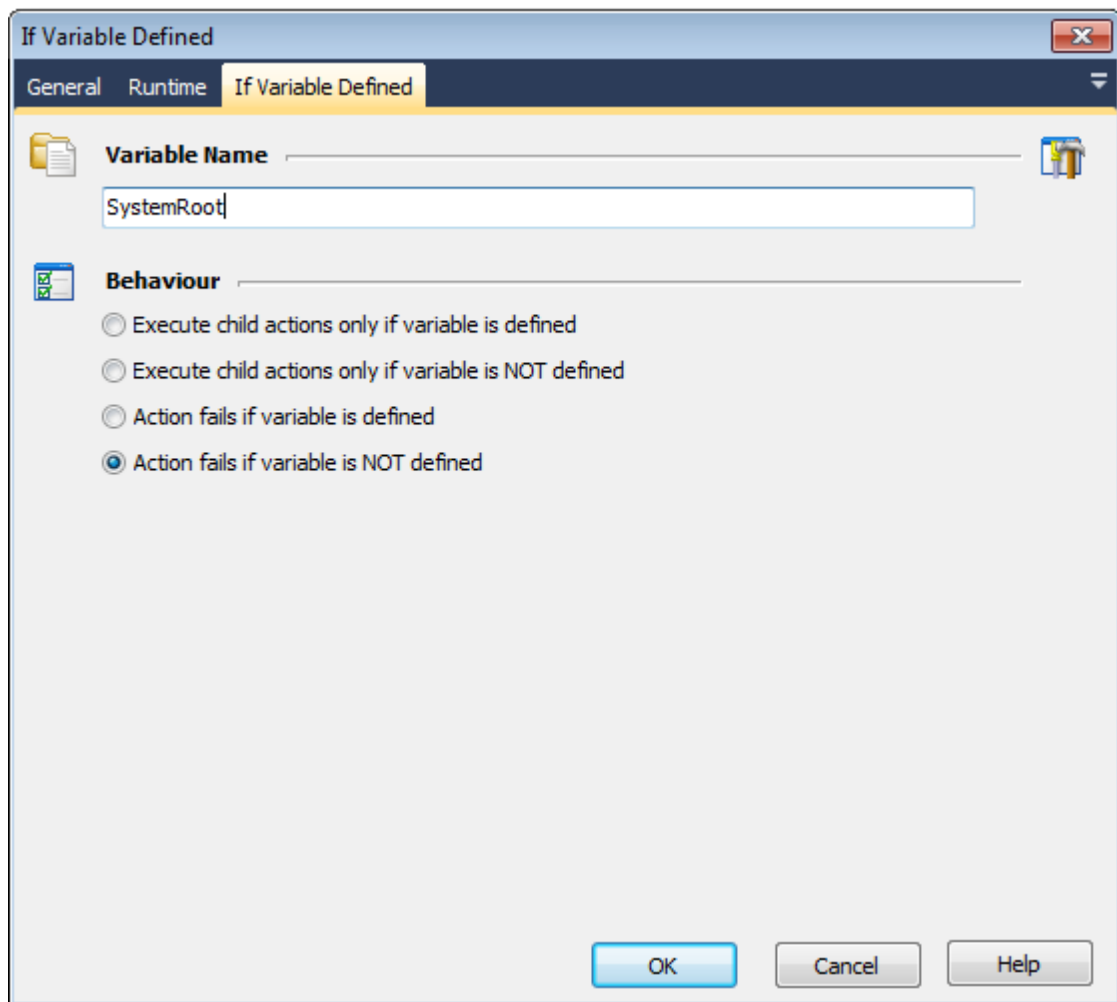
6.33.3 If Variable Defined Action

The If Variable Defined action enables you to control the flow of your build depending on whether a FinalBuilder 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

Specify the name of a FinalBuilder variable (or, as in the above picture, a Windows Environment variable) to test for.

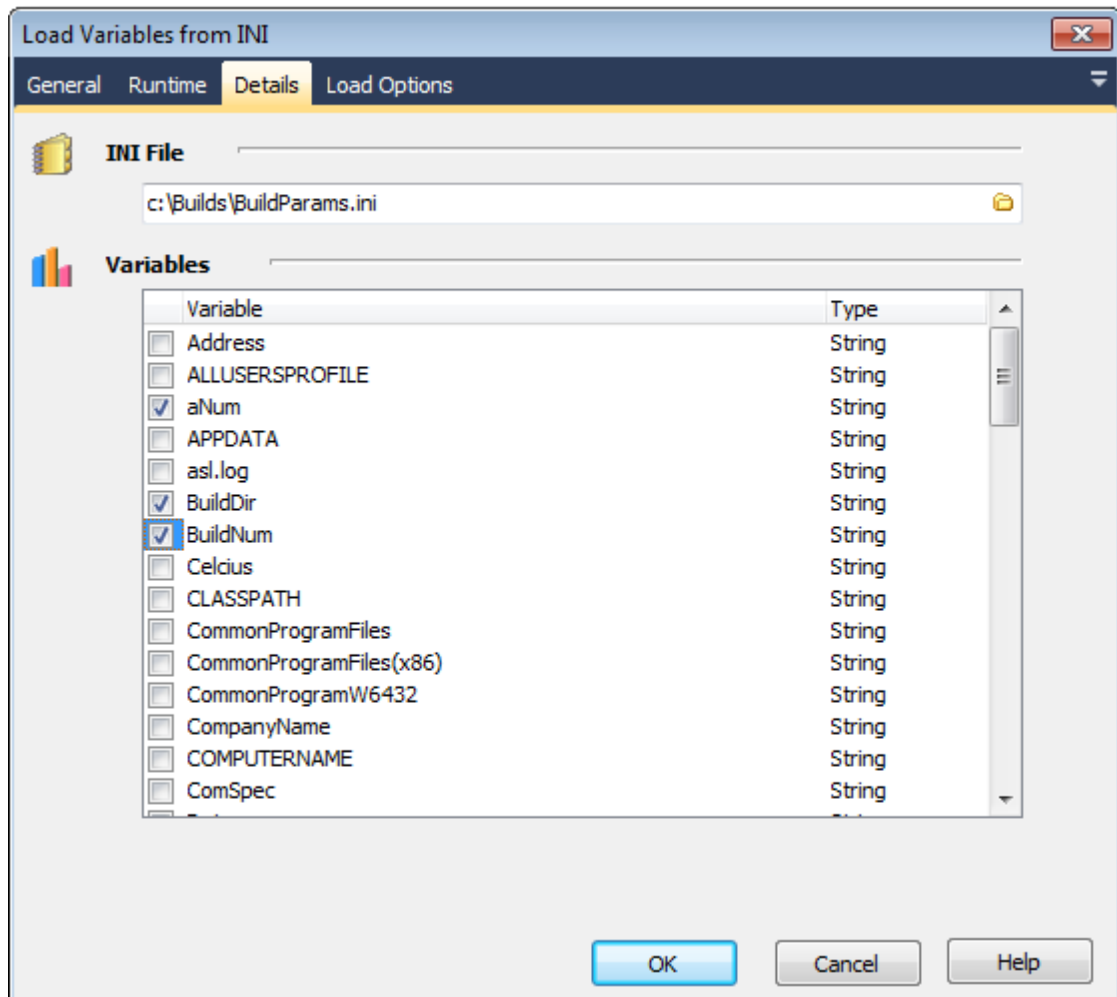
Behaviour

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.)

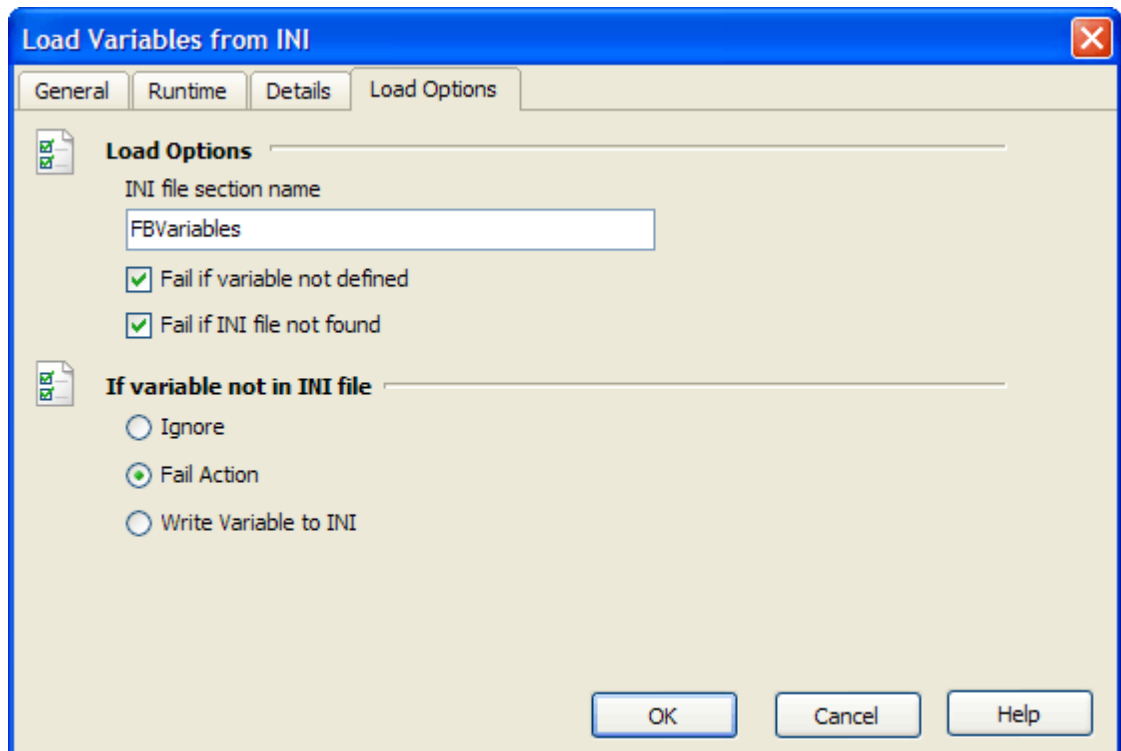
6.33.4 Load Variables From Ini

The Load Variables from INI file action enables you to set FinalBuilder variables to the values specified in an INI file.



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.



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

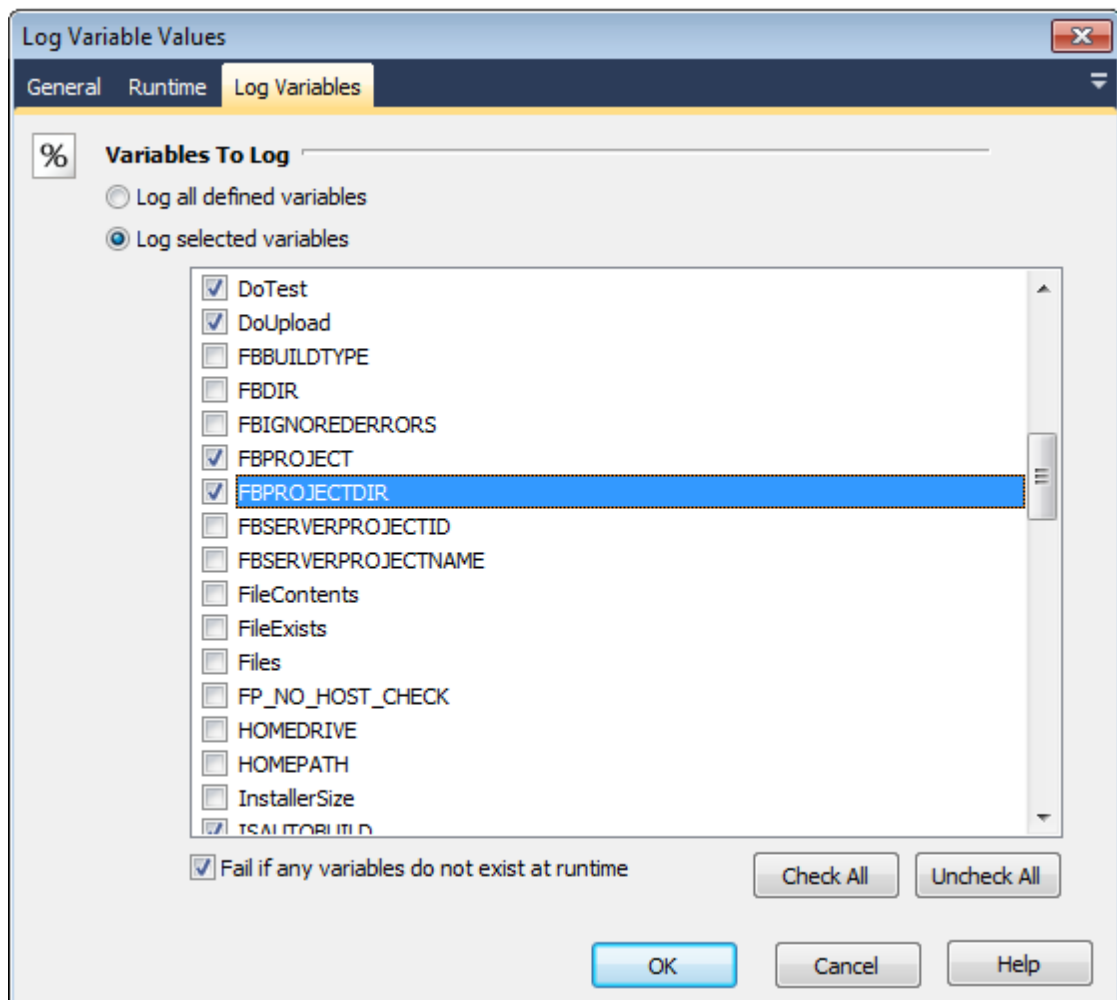
Ignore - Any variables not found in the INI file will be ignored (no failure, and the existing variable value will be kept.)

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.

6.33.5 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 FinalBuilder projects (see also Watches) or can be used to make log files easier to follow.



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

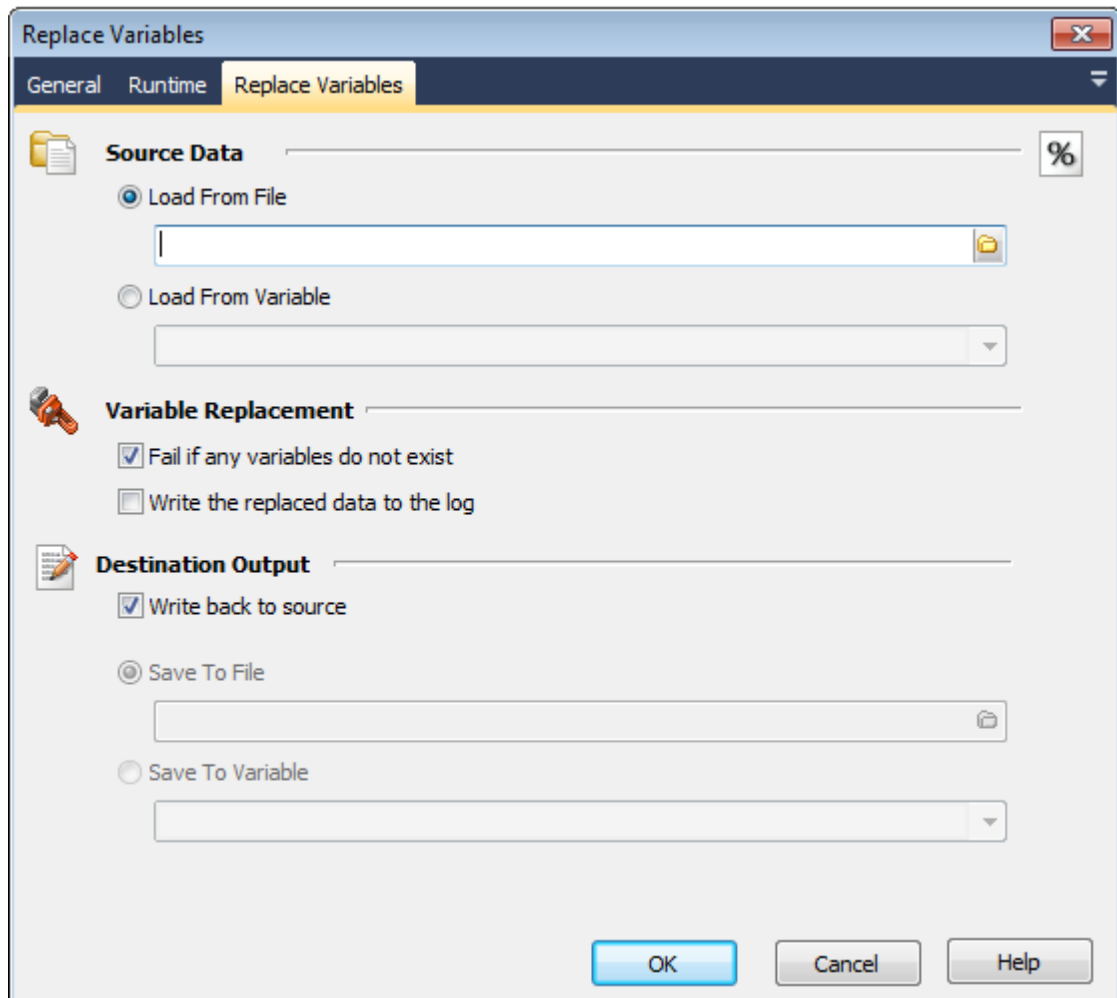
If this checkbox is not selected, variable names which do not exist will be logged as "Variable X is not defined".

See also: Action logging properties

6.33.6 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.



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.

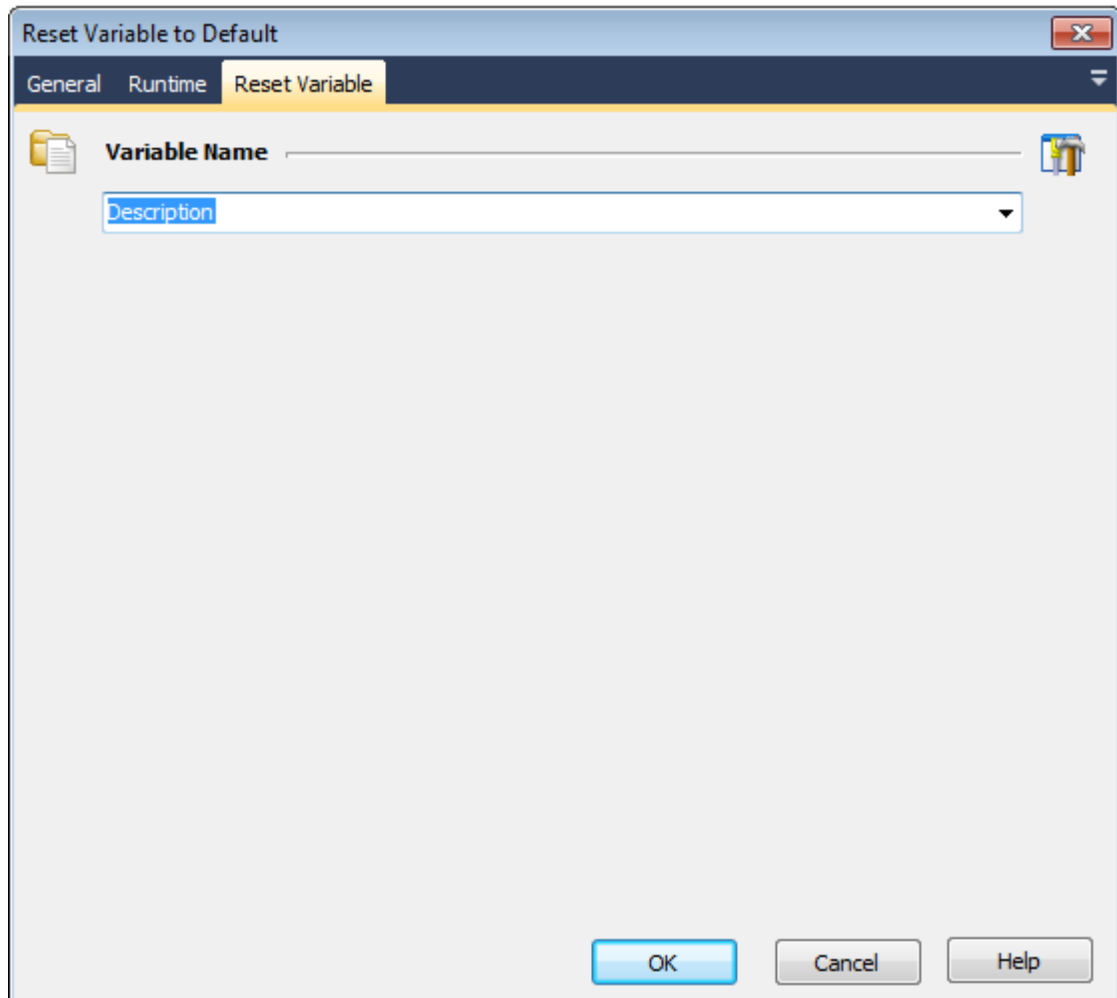
Destination Output

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.

6.33.7 Reset Variable to Default

The Reset Variable to Default action resets a variable to its default value.



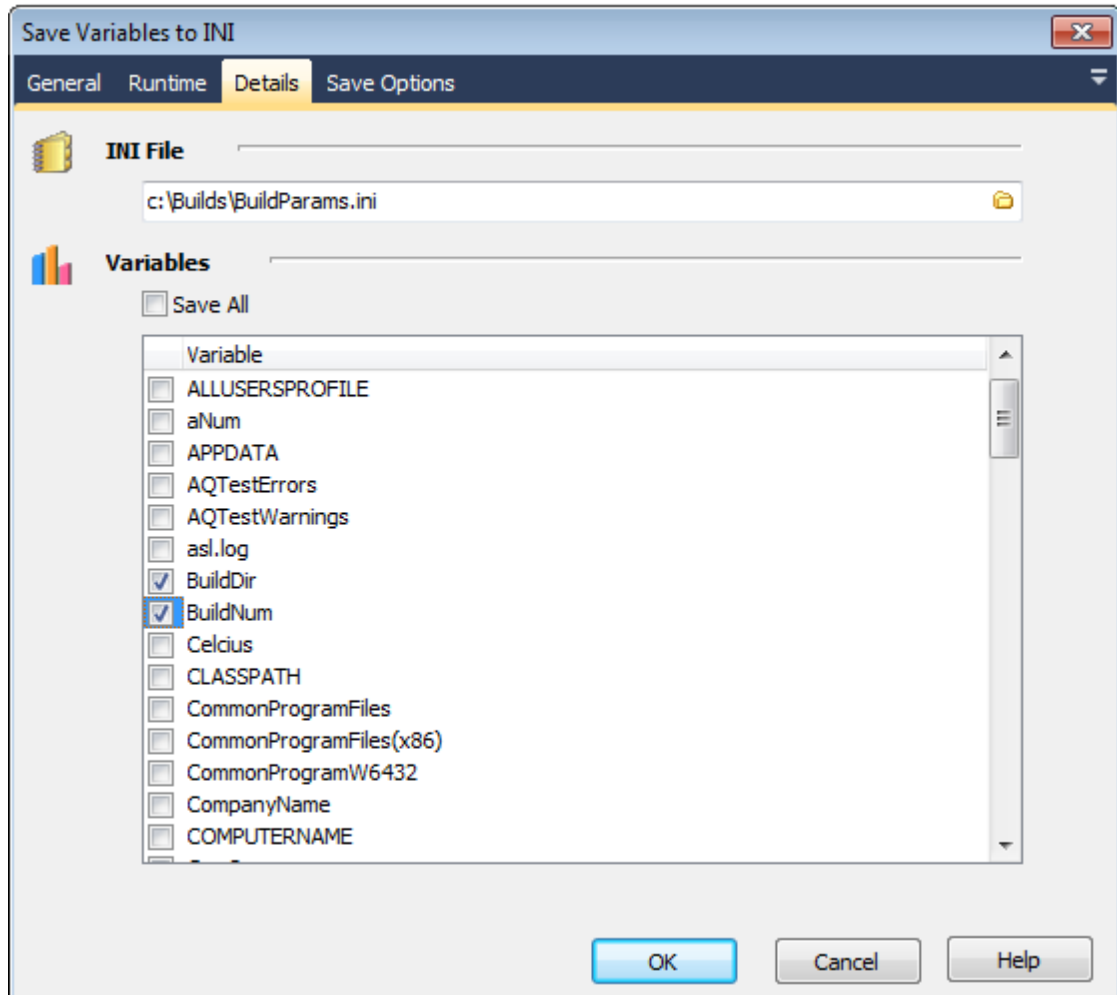
Variable Name

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.

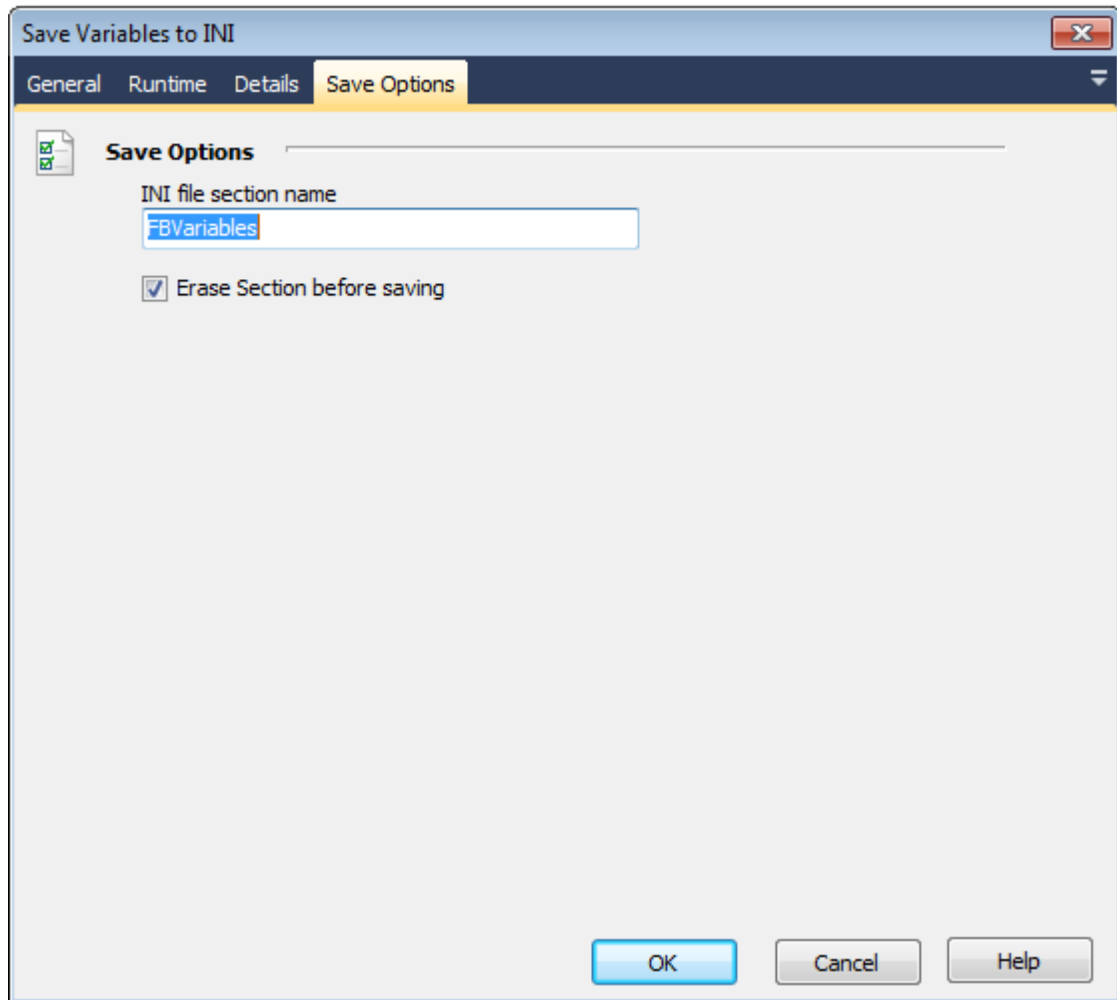
6.33.8 Save Variables To Ini

The Save Variables to INI file action enables you to save a set of FinalBuilder variables to an INI file.



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>



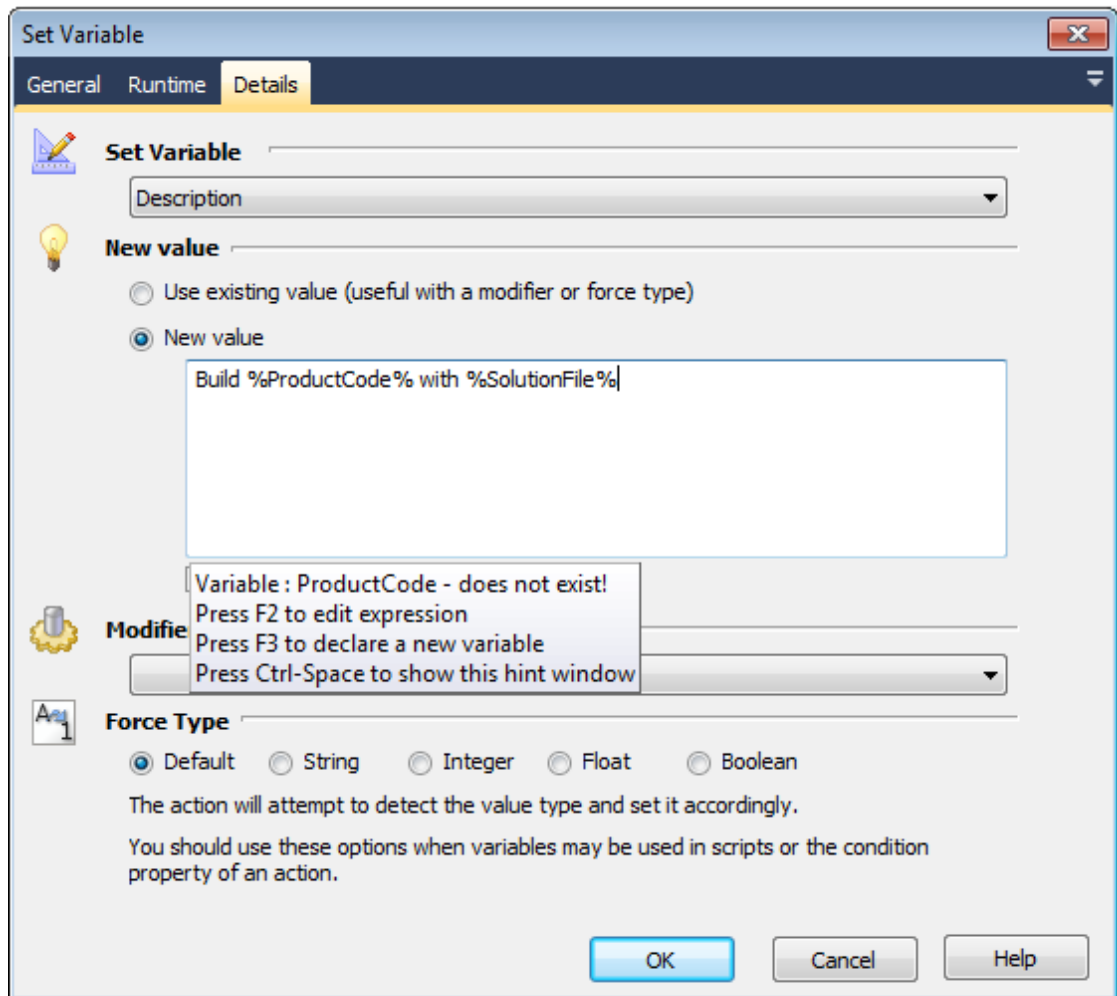
Section - Specify the section in the INI file to write the variables to

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.

6.33.9 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 whose 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

The name of a FinalBuilder 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
- Decrement - 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 FinalBuilder to use it's best guess what the type should be.

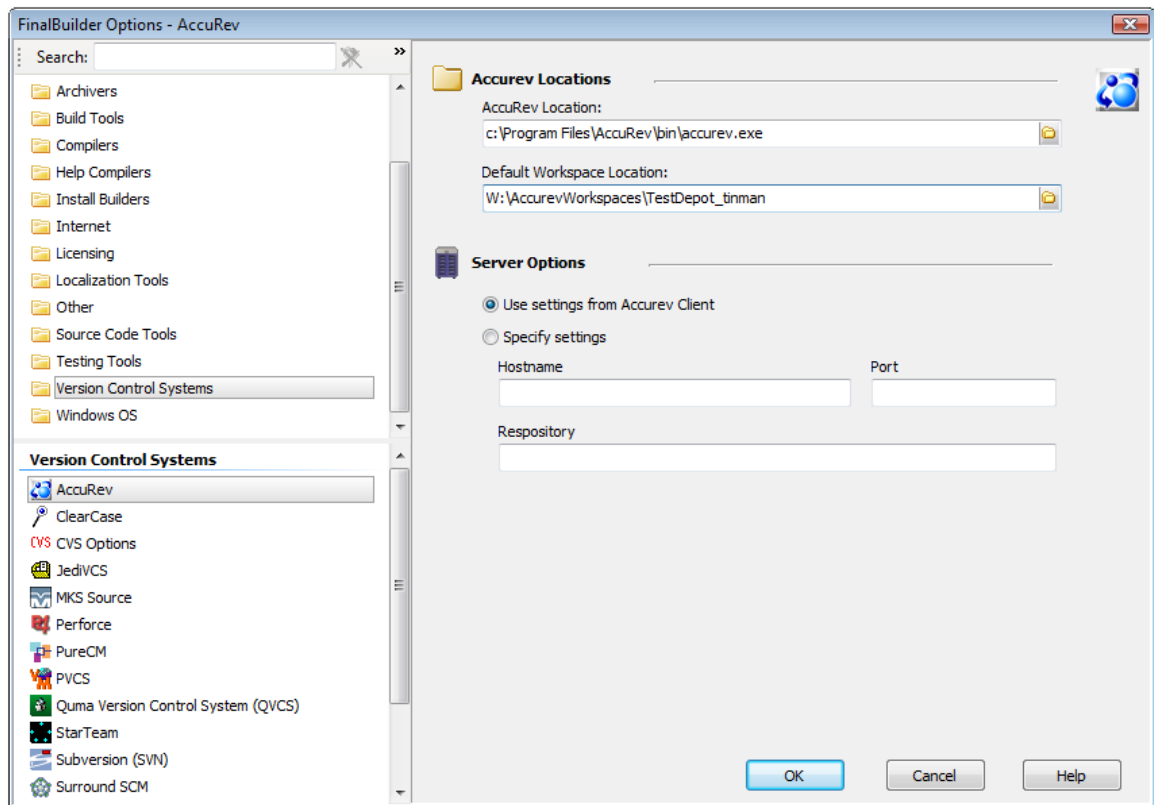
6.34 Version Control Actions

6.34.1 AccuRev

The AccuRev actions allow interaction with an AccuRev SCM server.

For more information regarding AccuRev, visit <http://www.accurev.com/>.

Global Options



AccuRev Location

The full path to the accurev.exe application.

Default Workspace Location

The local path for the default workspace, this option can be overridden in each action.

Use settings from AccuRev Client

This option will allow AccuRev to use settings previously defined in the Client GUI, which are stored in acclient.cnf.

Specify Settings

Specifying the server settings here will allow all actions to use the specified server.

Common Options

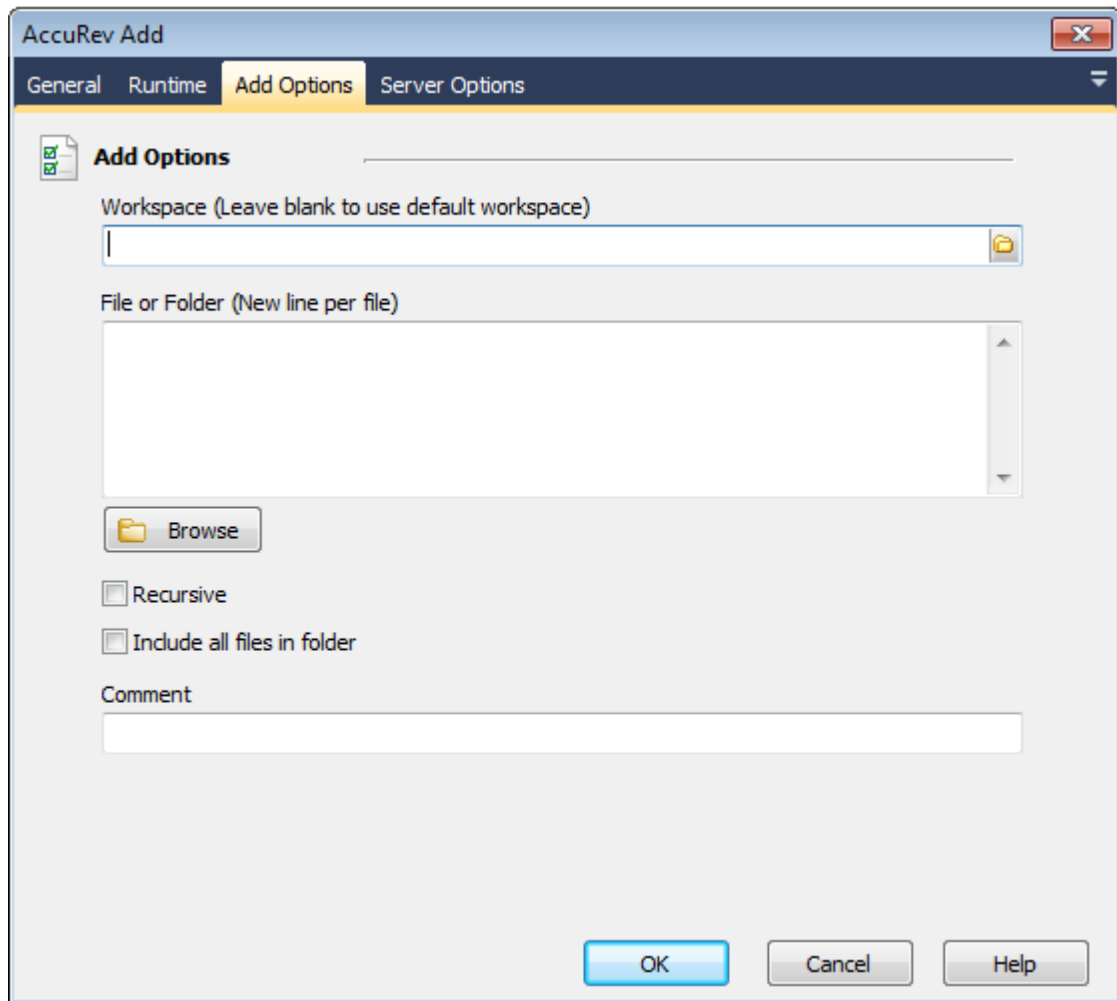
The screenshot shows a Windows-style dialog box titled "AccuRev Generic". It has four tabs: "General", "Runtime", "Generic Options", and "Server Options". The "Server Options" tab is selected. Inside the tab, there is a section titled "Server Options" with a small server icon. Below this, there are three radio buttons: "Use settings defined in global options" (which is selected), "Use settings from AccuRev Client", and "Specify Server Settings". Below the radio buttons, there are three text input fields: "Hostname" and "Port" are side-by-side, and "Repository" is below them. At the bottom right of the dialog, there are three buttons: "OK", "Cancel", and "Help".

These options allow each action to have different server settings.

Using settings from AccuRev Client will ignore any settings defined in the Global Settings.

6.34.1.1 AccuRev Add

AccuRev Add action enables you to convert one or more existing files and/or directories in a workspace to version controlled elements.

**Workspace**

The local path to the workspace.

File or Folder

The path to the file or folder which you are adding. This file must reside in your workspace.

Recursive

Recurse into specified directory and add all external files in that directory.

Include all files in folder

Selects all external files and directories in the workspace.

Comment

The comment to be associated with the transaction.

6.34.1.2 AccuRev Add a Change Package Entry

The AccuRev Add a Change Package Entry action enables you to create or update an entry for an element in the specified change package.

The screenshot shows a dialog box titled "AccuRev Add a Change Package Entry" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Runtime", "Add Entry Options" (which is selected and highlighted in yellow), and "Server Options". Below the tabs, there is a section titled "Add Entry Options" with a green checkmark icon. This section contains five input fields, each with a label and a description: "Workspace (Leave blank to use default workspace)" with a folder icon on the right; "Depot Name (Leave blank to use the Depot of the current workspace)" with a text input field; "Issue Record Number" with a text input field; "Version (Leave blank to use version of current workspace)" with a text input field; and "File" with a text input field and a folder icon on the right. At the bottom right of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Workspace

The local path to the workspace.

Depot Name

The depot in which the issue record is located.

Issue Record Number

The issue record whose change packaged will be processed.

Version

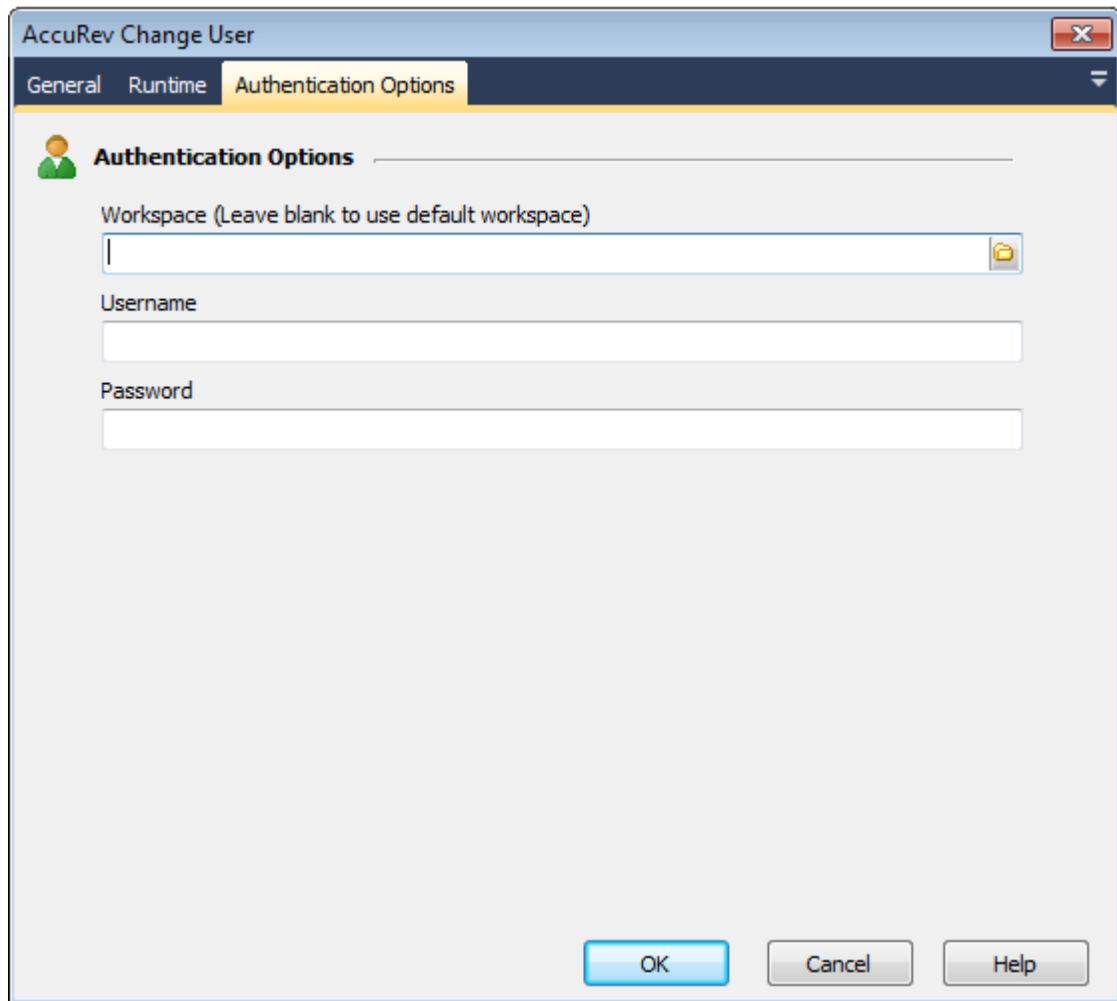
The head version of the change.

File

The element name.

6.34.1.3 AccuRev Change User

The AccuRev Change User action enables you to change the current principal and password.



Username

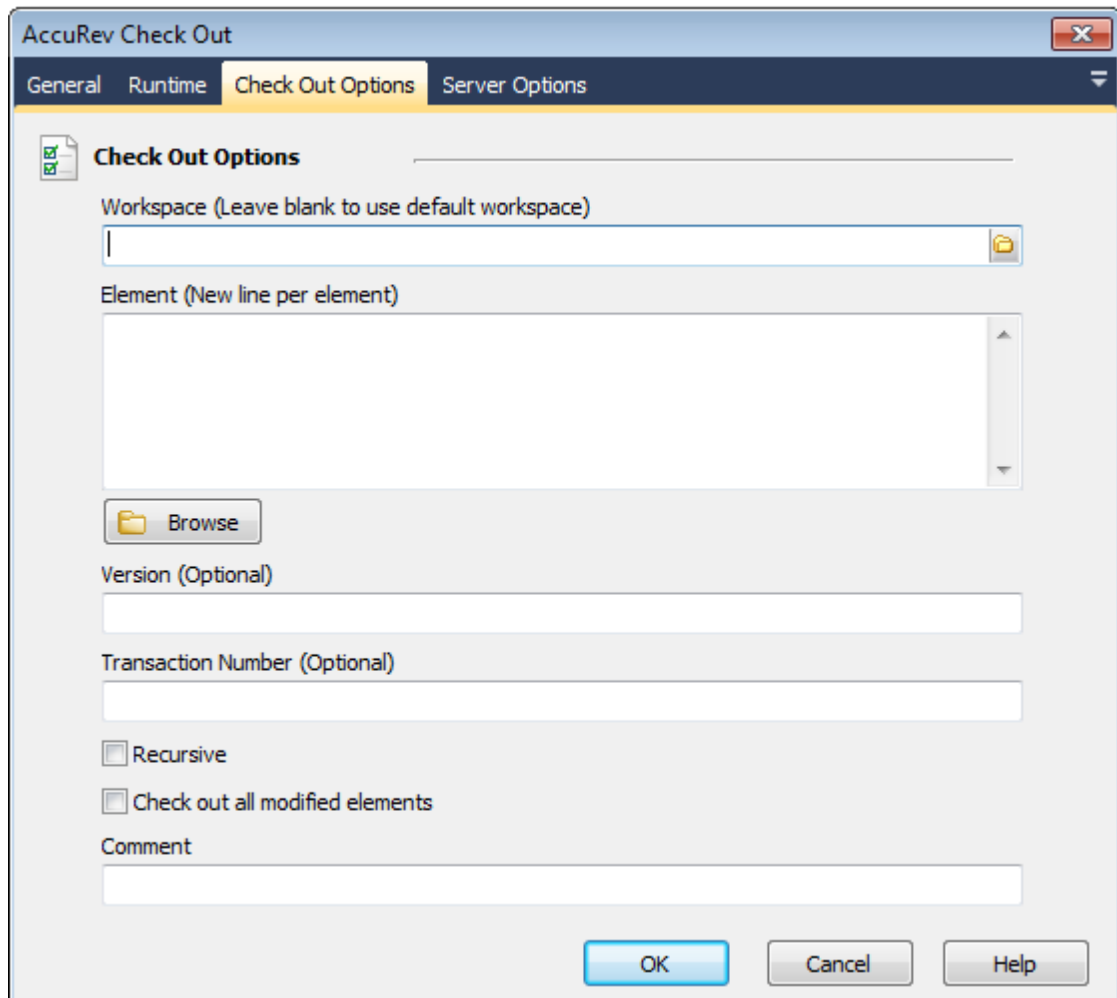
Used to set the AccuRev_Principal environment variable. This variable is used as the user identifier for all AccuRev commands executed from your project.

Password

The password is saved in the 'authn' file, which is read and authenticated along with the username on the AccuRev server.

6.34.1.4 AccuRev Check Out

The AccuRev Check Out action enables you to add elements to your workgroup's default group.

**Workspace**

The local path to the workspace.

Element

One or more elements, separated by white space.

Version

Check out a specified version of the element, copying it to your workspace.

Transaction Number

Check out all the versions associated with the specified transaction.

Recursive

Recurse in the directory specified, and check out all files that are not in the default group.

Check Out all modified elements

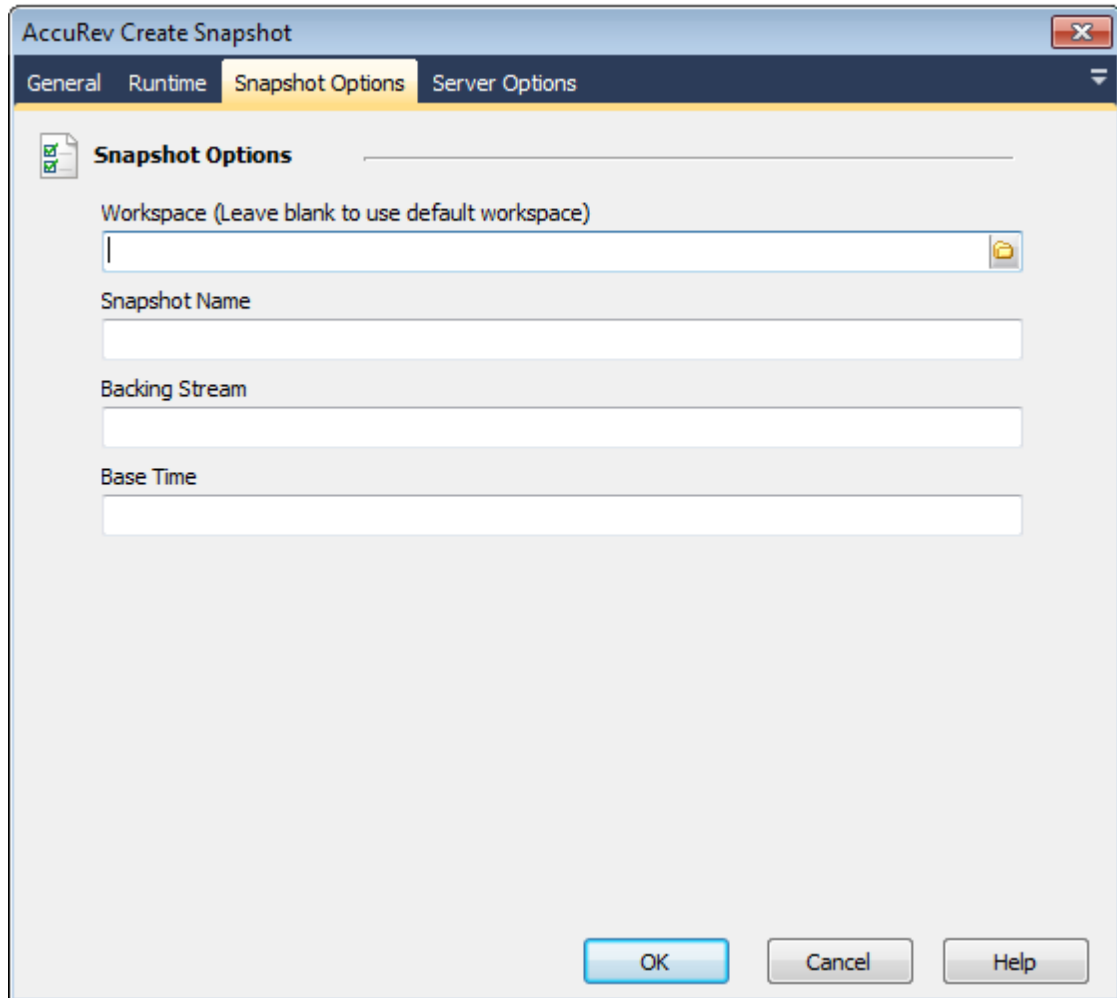
Check out all modified elements in the work space that are not already in the default group.

Comment

The comment to be associated with the transaction.

6.34.1.5 AccuRev Create Snapshot

The AccuRev Create Snapshot action enables you to create a new static stream (Snapshot), based on an existing steam.

**Workspace**

The local path to the workspace.

Snapshot Name

The name of the new snapshot. Must begin with a non-digit character and cannot contain slashes.

Backing Stream

The existing stream which the new snapshot will be based on.

Base Time

The date at which to take the snapshot. The snapshot will only contain the elements that were in backing stream on this date.

6.34.1.6 AccuRev Create Stream

The AccuRev Create Stream action enables you to create a new dynamic stream based on an existing backing stream.

The screenshot shows the 'AccuRev Create Stream' dialog box with the 'Create Stream Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Create Stream Options' (highlighted), and 'Server Options'. The 'Create Stream Options' tab contains the following fields and controls:

- Create Stream Options**: A section header with a green checkmark icon.
- Workspace**: A text field with the placeholder text '(Leave blank to use default workspace)' and a folder icon button on the right.
- New Stream name**: A text field.
- Backing Stream**: A text field.
- Pass Through Stream**: A checkbox.
- Base Time**: A text field.

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Workspace

The local path to the workspace.

New Stream Name

The name of the new Stream.

Backing Stream

The existing stream, that the new stream will be based on.

Pass Through Stream

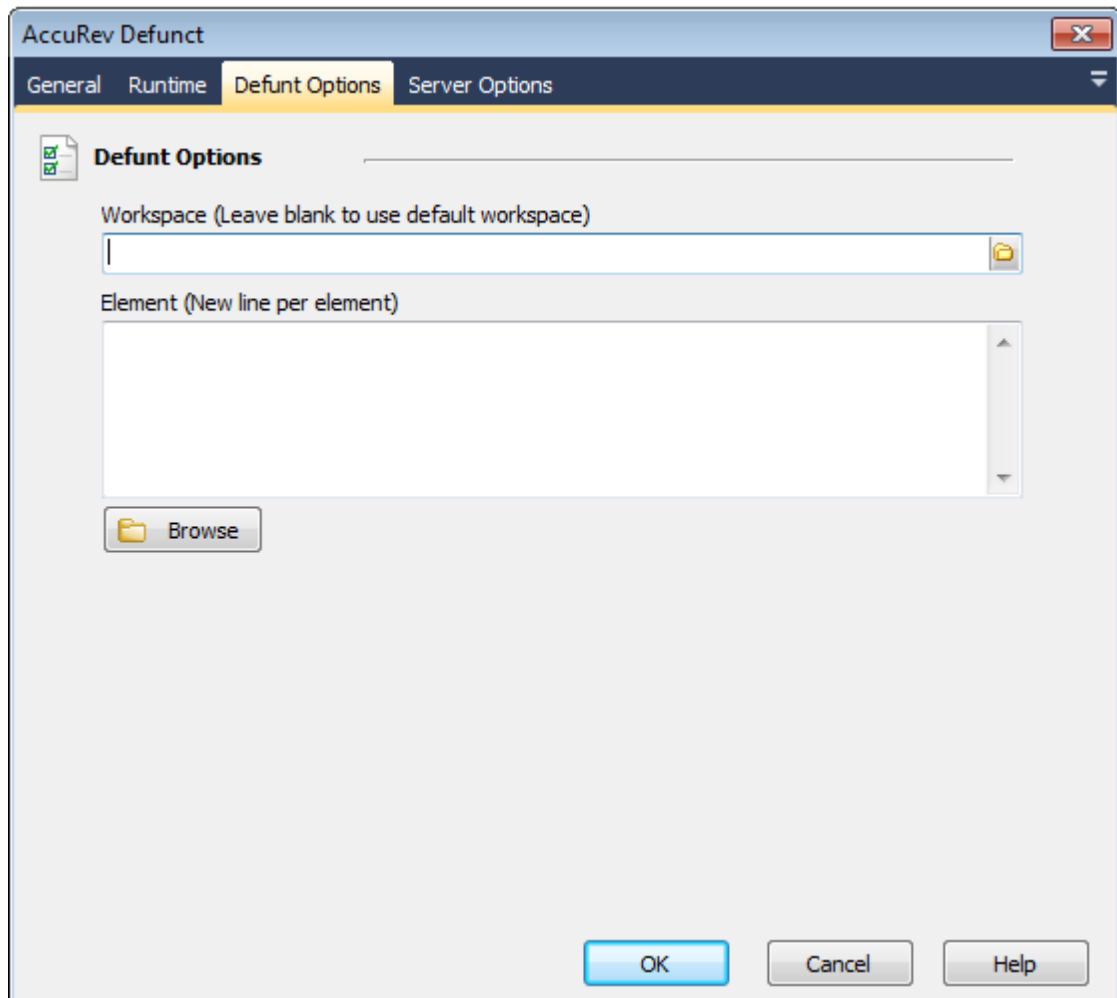
Make the new stream a 'Pass Through Stream'.

Snapshot Time

Base the new stream on the state of backing stream at the specified time.

6.34.1.7 AccuRev Defunct

The AccuRev Defunct action enables you to remove elements from active use in your workspace.

**Workspace**

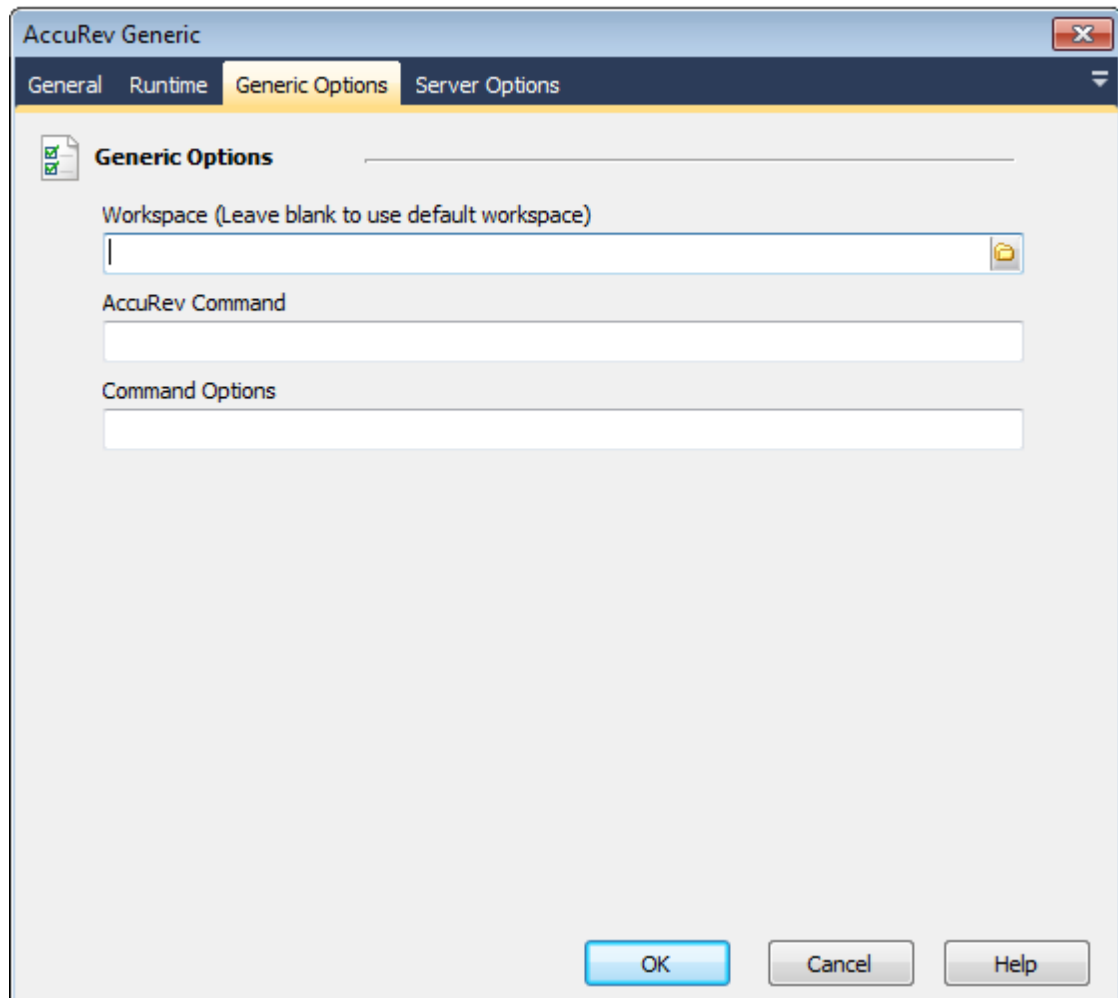
The local path to the workspace.

Files

The files to be made defunct.

6.34.1.8 AccuRev Generic

The AccuRev Generic action enables you to call any command used by the AccuRev CLI, and pass in any command line parameters.



Workspace

The local path to the workspace.

AccuRev Command

The command to pass to the command line version of AccuRev.

Command Options

These are the options passed to the AccuRev command.

Examples

accurev unlock [-kf | -kt] <stream>

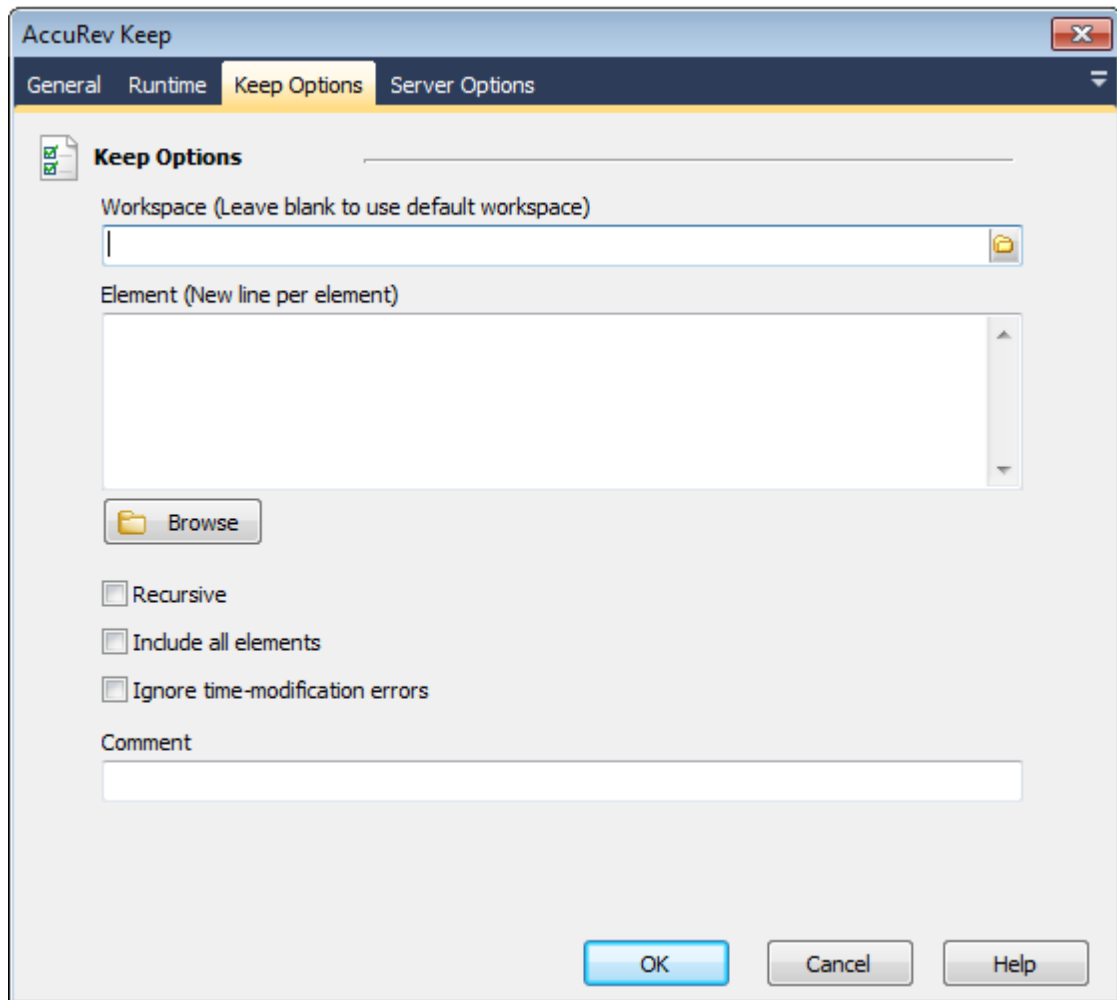
AccuRev Command: unlock
Command Options: -kf MyStream

accurev excl [-s <stream>] <element>

AccuRev Command: `excl`
Command Options: `-s MyStream SomeElement`

6.34.1.9 AccuRev Keep

The AccuRev Keep action enables you to create a new version of an element, using the copy of the element that is in your workspace.



Workspace

The local path to the workspace.

File or Folder

The list of elements that will be kept, separated by white space.

Recursive

Recurse in to the specified directory and keep all files that are not in the default group.

Include all elements

Select all elements in the default group.

Ignore time-modification errors

Ignore any errors regarding modification-time timewarps. Allows a keep on a file with a

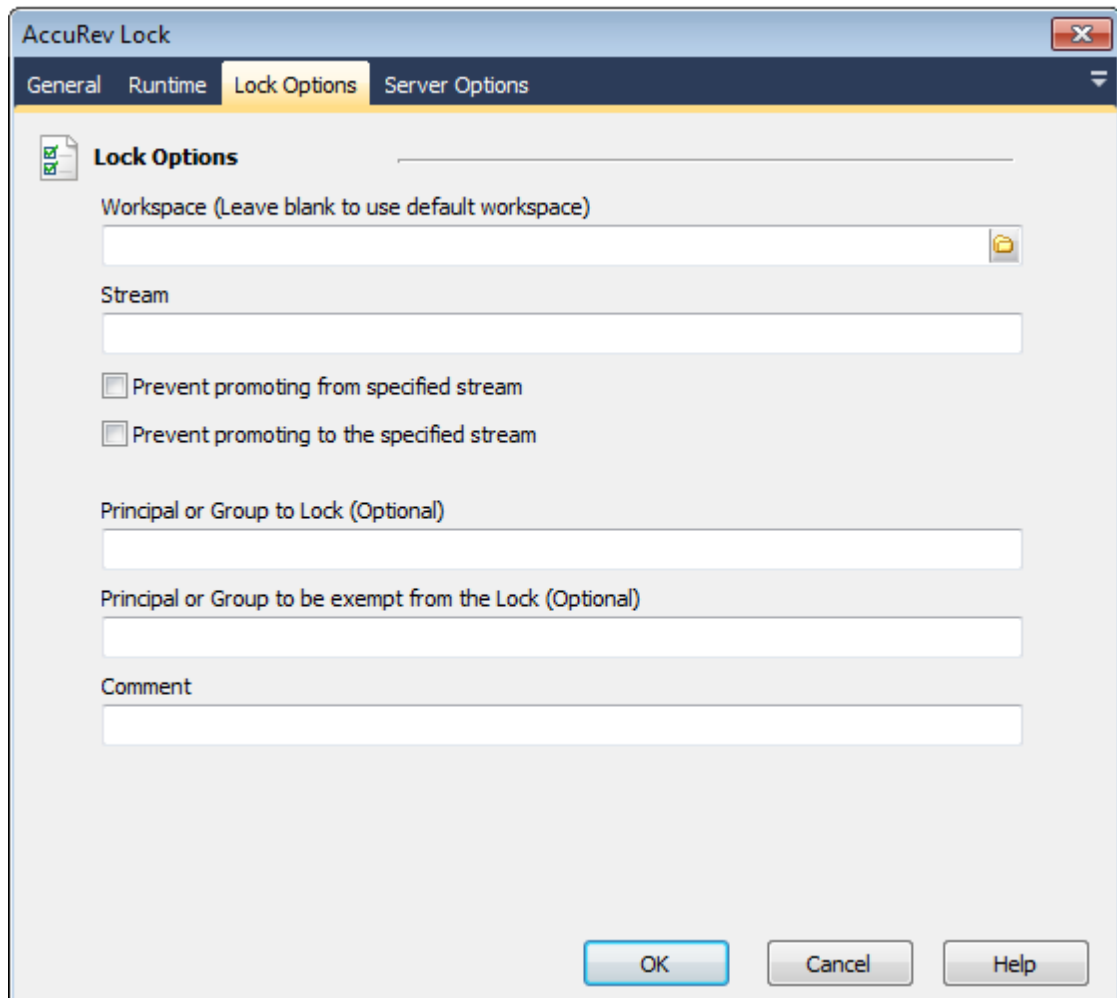
modification time older than its ancestor's modification time.

Comment

The comment to be associated with the transaction.

6.34.1.10 AccuRev Lock

The AccuRev Lock action allows you to prevent users from making various changes to the specified dynamic stream.

**Workspace**

The local path to the workspace.

Stream

The stream on which the lock will be placed.

Prevent Promoting from Specified Stream

Users will be prevented from promoting versions from the specified stream to other streams.

Prevent Promoting to the Specified Stream

Users will be prevented from promoting versions to the specified stream, from other

streams.

Principal or Group to Lock

The lock will only apply to the specified principal or group.

Principal or Group to exempt from the Lock

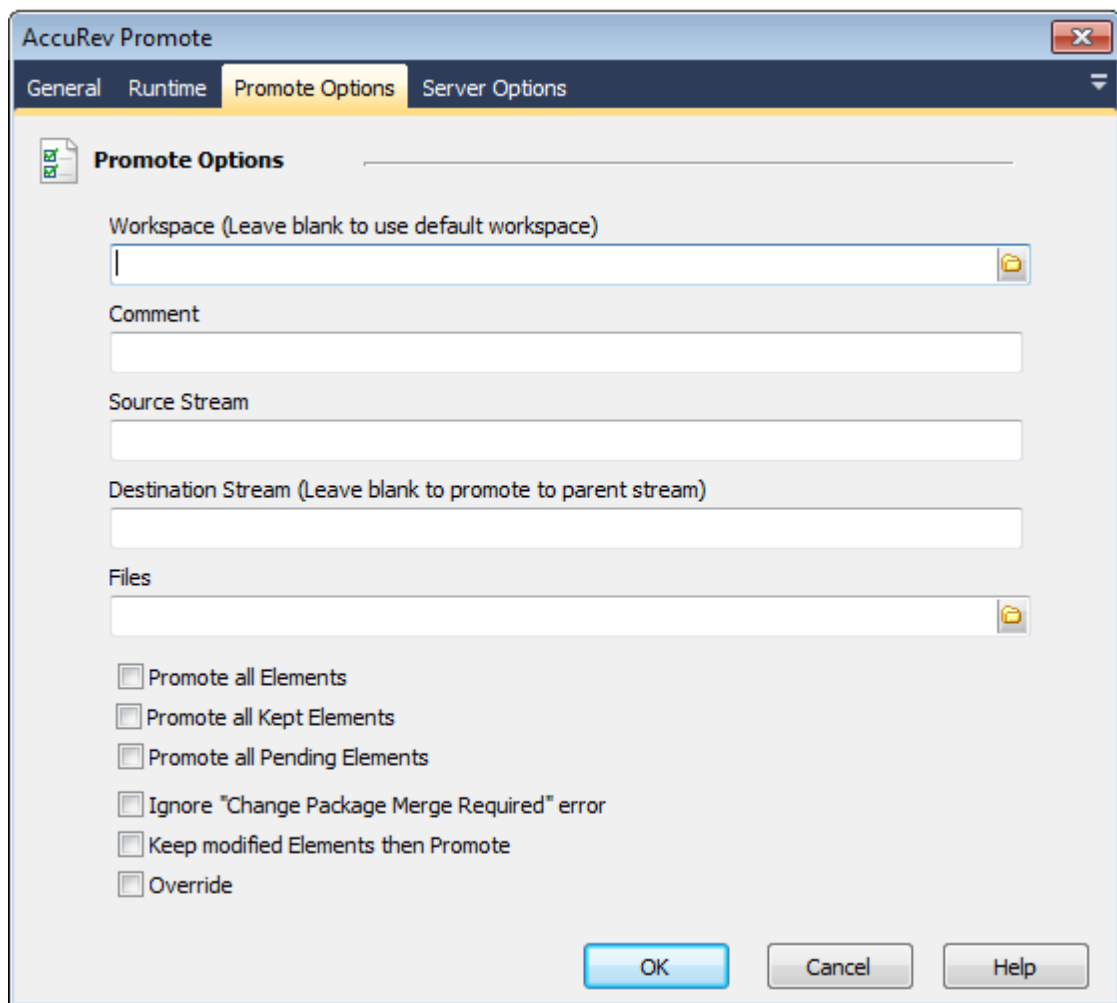
The lock will apply to everyone, except the specified principal or group.

Comment

The comment to be associated with the transaction.

6.34.1.11 AccuRev Promote

The AccuRev Promote action enables you to send a stream's changes to its parent stream.

**Workspace**

The local path to the workspace.

Comment

The comment to be associated with the transaction.

Source Stream

The stream to promote.

Destination Stream

The stream to promote the change to.

Files

The elements to promote to the destination stream.

Keep modified Elements then Promote

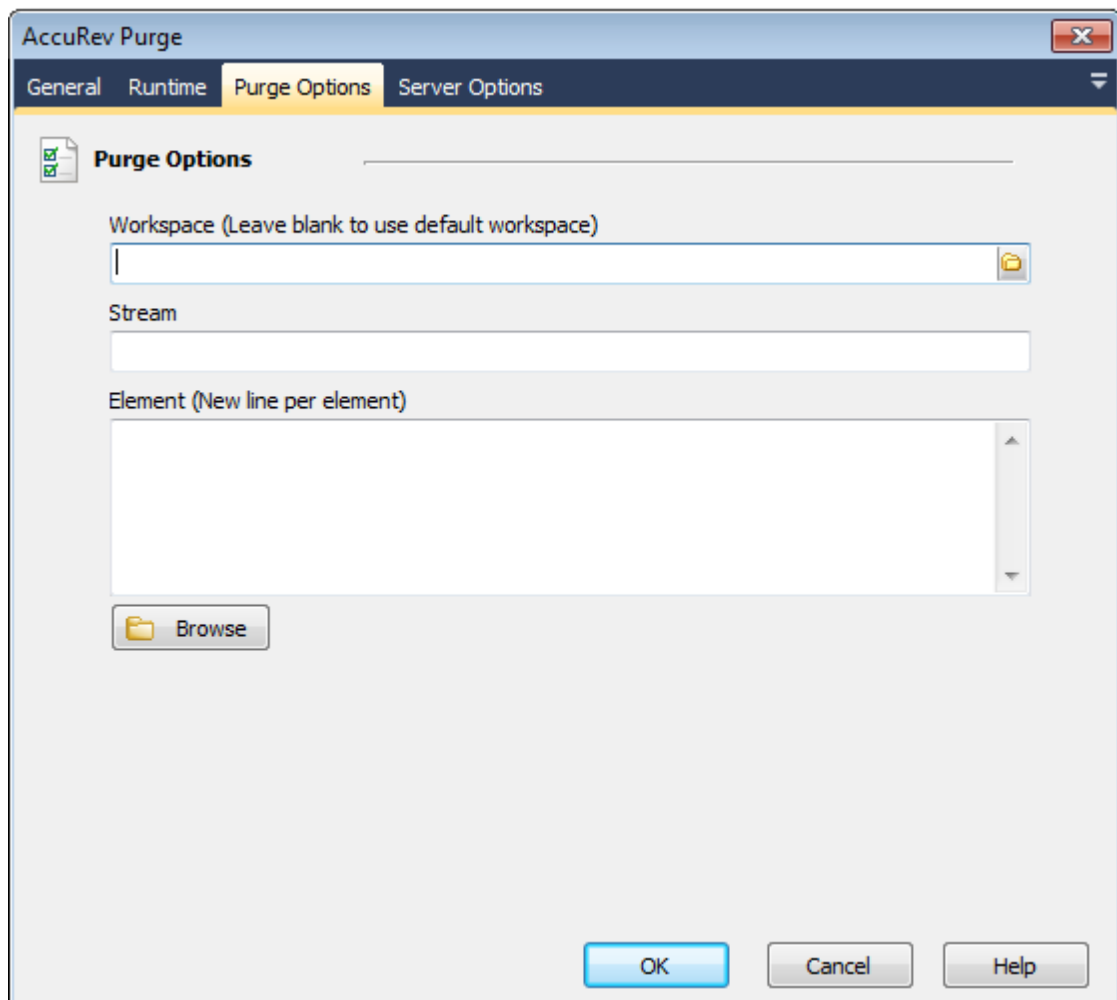
If an elements status is set to modified, then the element will be kept first, then promoted.

Override

Changes the behaviour of AccuRev when dealing with name discrepancies, and forces the promotion even if a merge would normally be required.

6.34.1.12 AccuRev Purge

The AccuRev Purge action enables you to undo all the changes made to an element since activation of the workspace.

**Workspace**

The local path to the workspace.

Stream

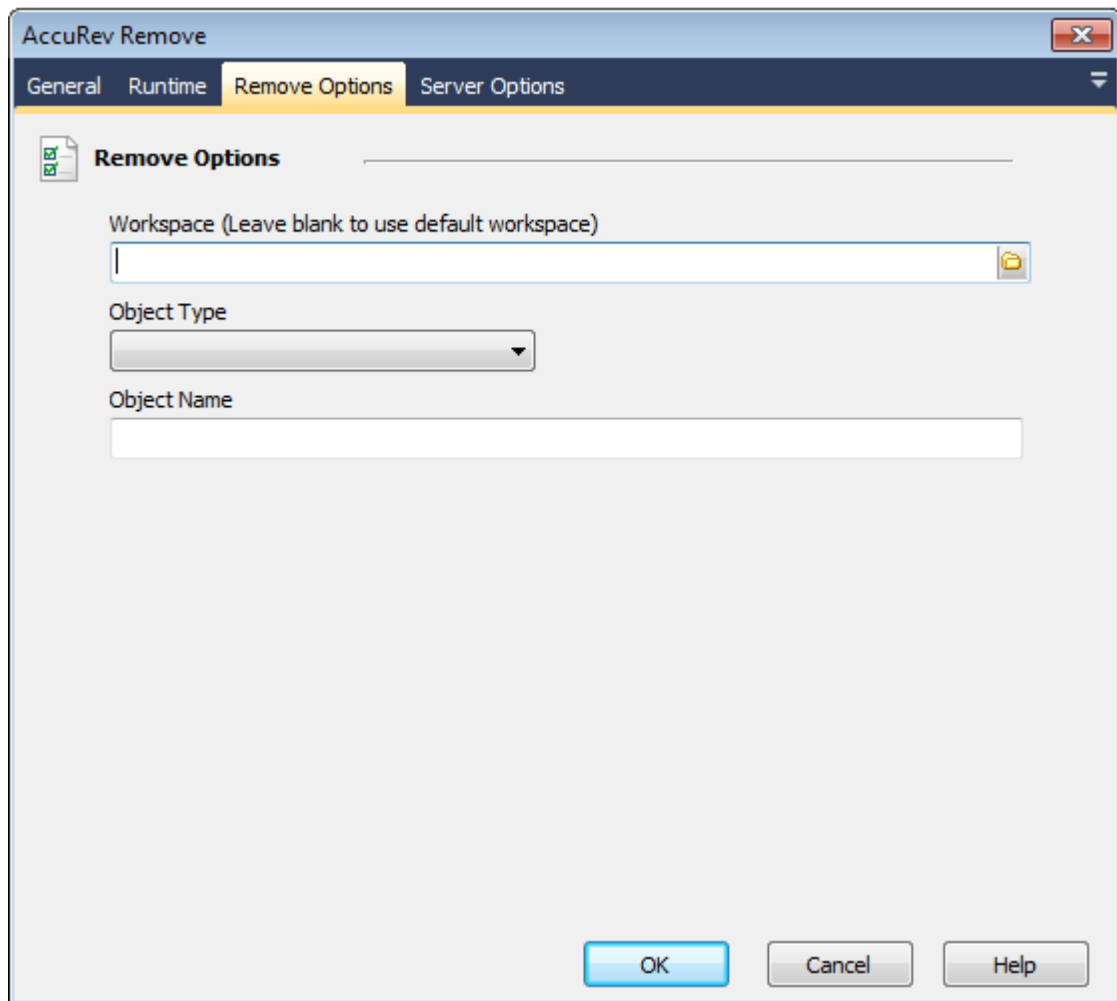
The stream to promote the changes from.

Files

The elements to purge.

6.34.1.13 AccuRev Remove

The AccuRev Remove action enables you to hide a reference tree, stream, user, group or workspace. It does not actually remove the object from the Depot.

**Workspace**

The local path to the workspace.

Object Type

The type of the object to remove, possible types are Reference Tree, Stream, User, Group or Workspace.

Object Name

The name of the object to remove.

6.34.1.14 AccuRev Remove a Change Package Entry

The AccuRev Remove a Change Package Entry action enables you to remove an entry from the specified change package.

The screenshot shows a dialog box titled "AccuRev Remove a Change Package Entry" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog has four tabs: "General", "Runtime", "Remove Entry Options" (which is selected and highlighted in yellow), and "Server Options". Below the tabs, there is a section titled "Remove Entry Options" with a small icon of a document with a green checkmark. This section contains four input fields: "Workspace (Leave blank to use default workspace)" with a folder selection icon on the right; "Depot Name (Leave blank to use the Depot of the current workspace)" with a text input field; "Issue Record Number" with a text input field; and "File" with a text input field and a folder selection icon on the right. At the bottom right of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Workspace

The local path to the workspace.

Depot Name

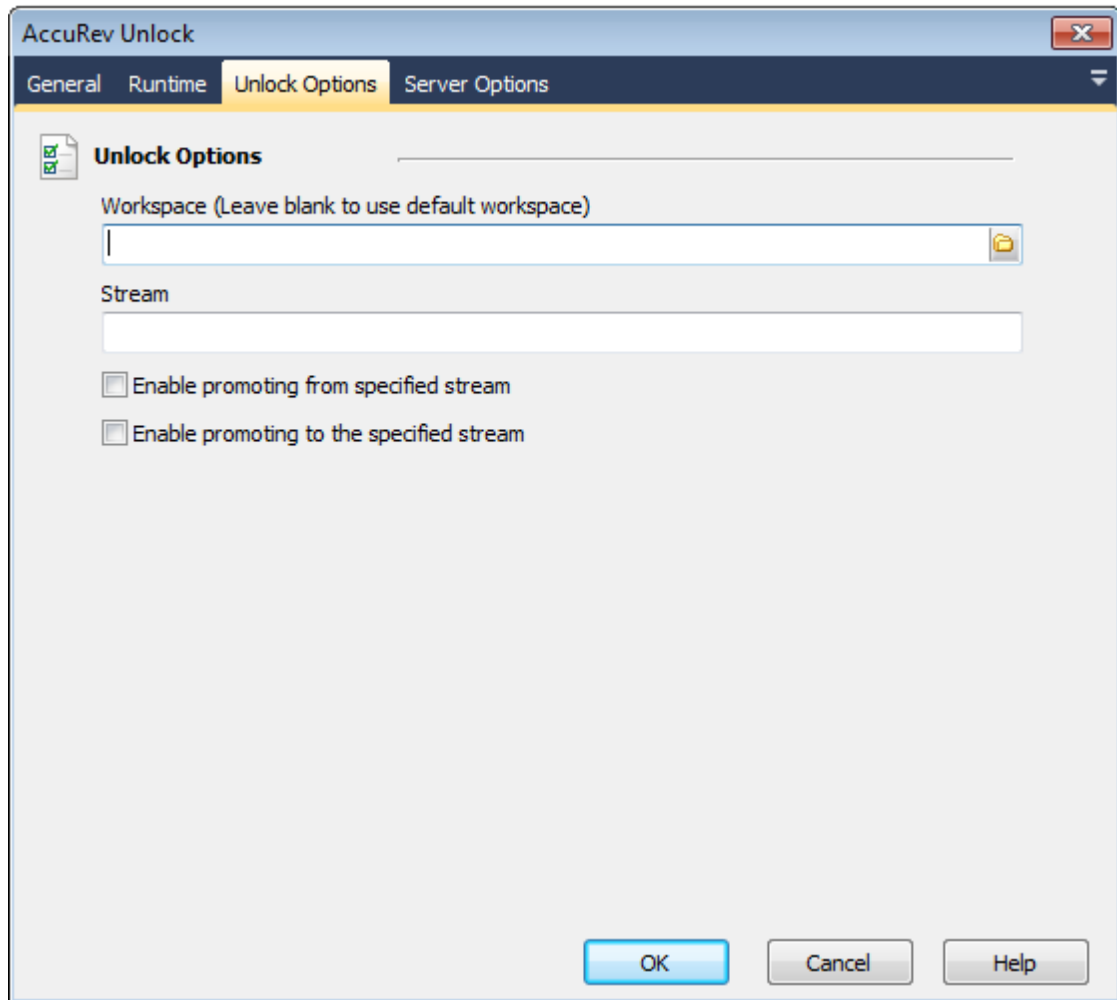
The depot in which the issue record is located.

Issue Record Number

The issue record to be removed.

6.34.1.15 AccuRev Unlock

The AccuRev Unlock action enables you to remove a lock from a dynamic stream.

**Workspace**

The local path to the workspace.

Stream

The stream to remove the lock from.

Enable promoting from specified stream

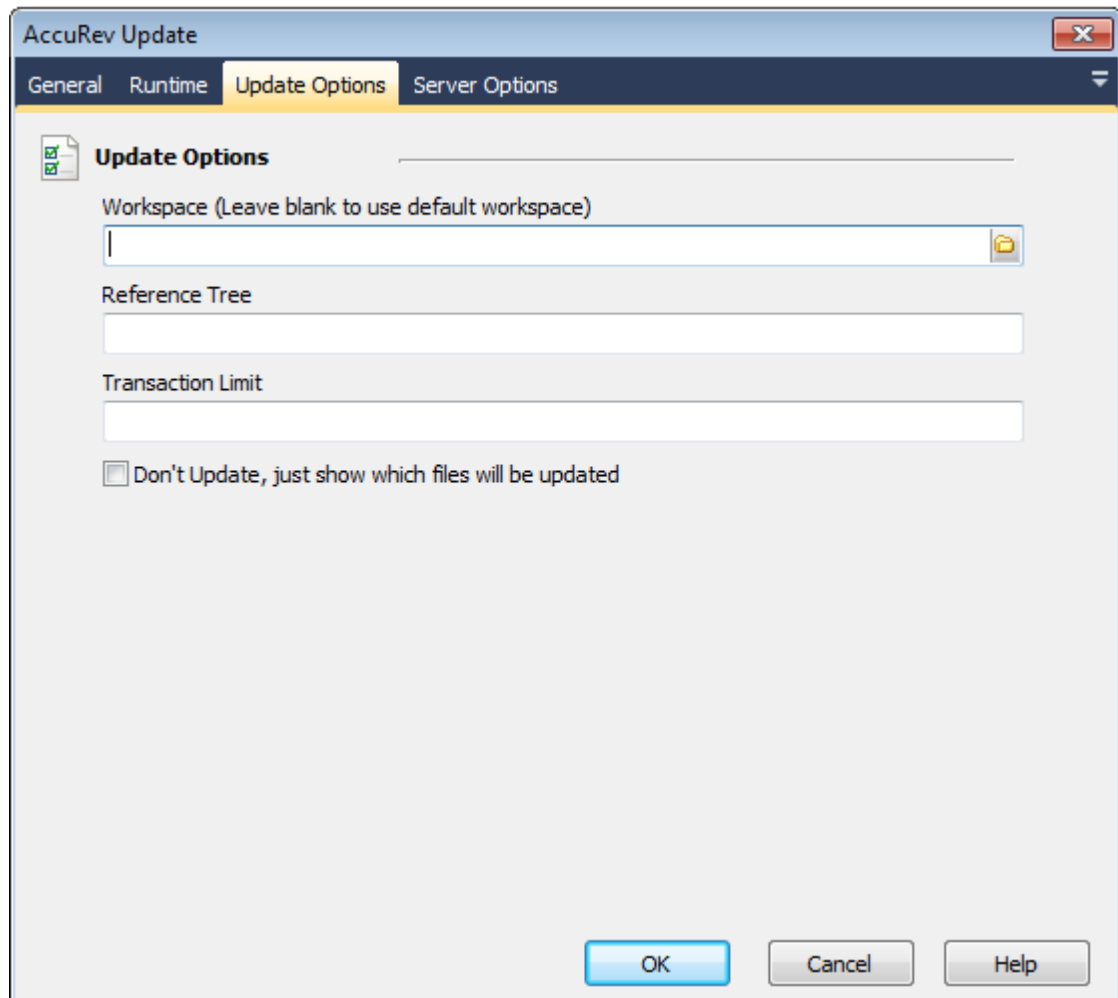
Promoting from the specified stream will be enabled.

Enable promoting to the specified stream

Promoting to the specified stream will be enabled.

6.34.1.16 AccuRev Update

The AccuRev Update action allows you to copy versions from your workspace's backup stream into your workspace

**Workspace**

The local path to the workspace.

Reference Tree

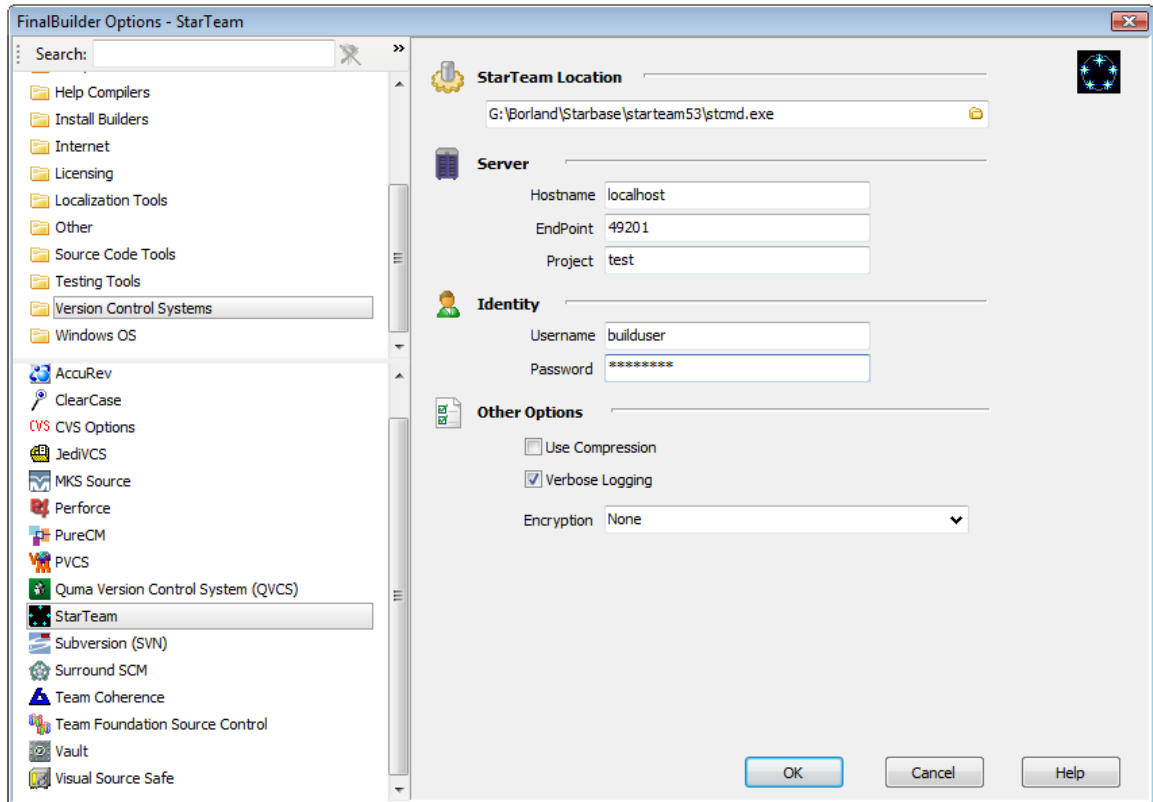
The Reference Tree to perform the update on.

Transaction Limit

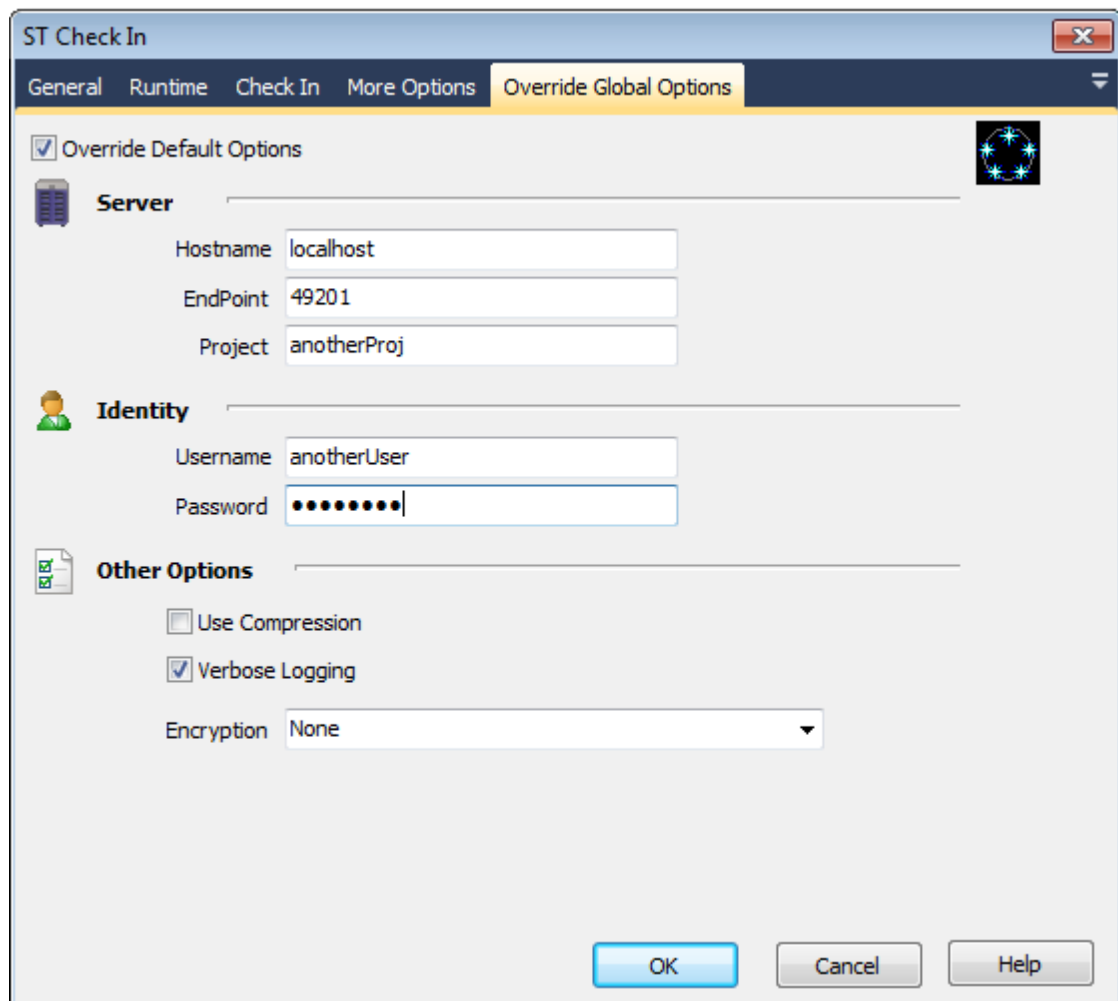
Update the workspace only up to the specified transaction. Any changes made prior to the transaction will not be updated.

6.34.2 Borland StarTeam Actions

These Actions support Borland StarTeam 5.2.x or higher. They provide an easy to use wrapper around the StarTeam command line tool too. Before using these actions you need to check that the path to the StarTeam command line tool is set in the FinalBuilder Options Dialog. FinalBuilder will attempt to find the path automatically however it is not always possible (depends on your installation).



You can also set defaults which will be used for the StarTeam actions in your projects, you can override the defaults if needed in the individual Actions.



The image shows a software dialog box titled "ST Check In". It has a tabbed interface with four tabs: "General", "Runtime", "Check In", and "Override Global Options". The "Override Global Options" tab is currently selected and highlighted in yellow. At the top left of the dialog, there is a checked checkbox labeled "Override Default Options" and a small icon of a network or server cluster. The dialog is organized into three sections, each with a header and a small icon: "Server" (server rack icon), "Identity" (person icon), and "Other Options" (checklist icon). The "Server" section contains three text input fields: "Hostname" with the value "localhost", "EndPoint" with the value "49201", and "Project" with the value "anotherProj". The "Identity" section contains two text input fields: "Username" with the value "anotherUser" and "Password" which is masked with ten black dots. The "Other Options" section contains two checkboxes: "Use Compression" (unchecked) and "Verbose Logging" (checked), followed by a dropdown menu for "Encryption" which is currently set to "None". At the bottom right of the dialog, there are three buttons: "OK", "Cancel", and "Help".

ST Check In

General Runtime Check In More Options **Override Global Options**

☒ Override Default Options

Server

Hostname localhost

EndPoint 49201

Project anotherProj

Identity

Username anotherUser

Password ••••••••••

Other Options

☐ Use Compression

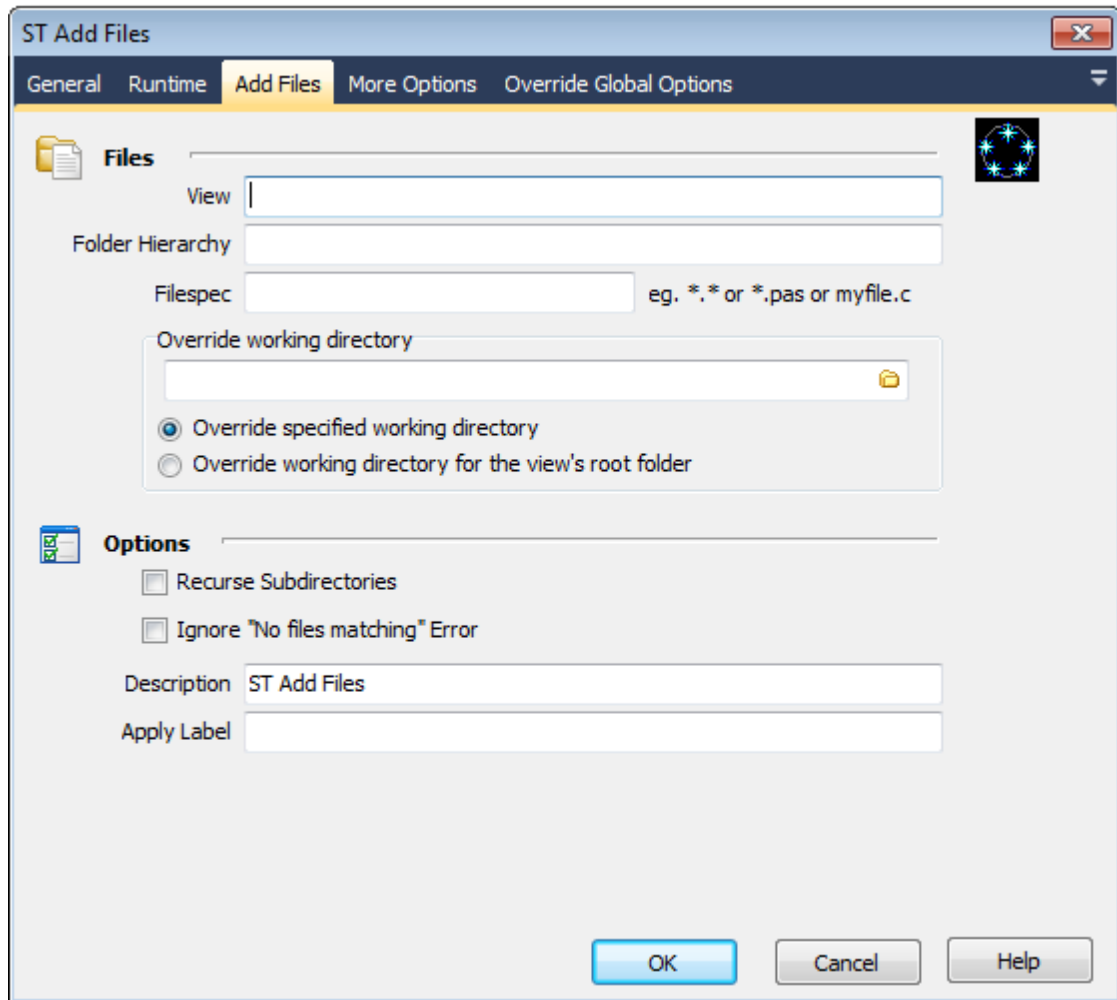
☒ Verbose Logging

Encryption None

OK Cancel Help

6.34.2.1 StarTeam Add Files Action

This action calls the StarTeam command line tool to Add files.



6.34.2.2 StarTeam Apply Label Action

This action calls the StarTeam command line tool to Apply a Label.

6.34.2.3 StarTeam Check In Action

This action calls the StarTeam command line tool to check in files.

6.34.2.4 StarTeam Check Out Action

This action calls the StarTeam command line tool to check out files.

6.34.2.5 StarTeam Create Label Action

This action calls the StarTeam command line tool to create a Label

6.34.2.6 StarTeam Delete Files Action

This action calls the StarTeam command line tool to delete files.

6.34.2.7 StarTeam Generic Action

Use the StarTeam Generic Action to call StarTeam commands that are not natively supported by the other StarTeam actions.

6.34.2.8 StarTeam List Files Action

This action calls the StarTeam command line tool to List files.

6.34.2.9 StarTeam Lock/Unlock Files Action

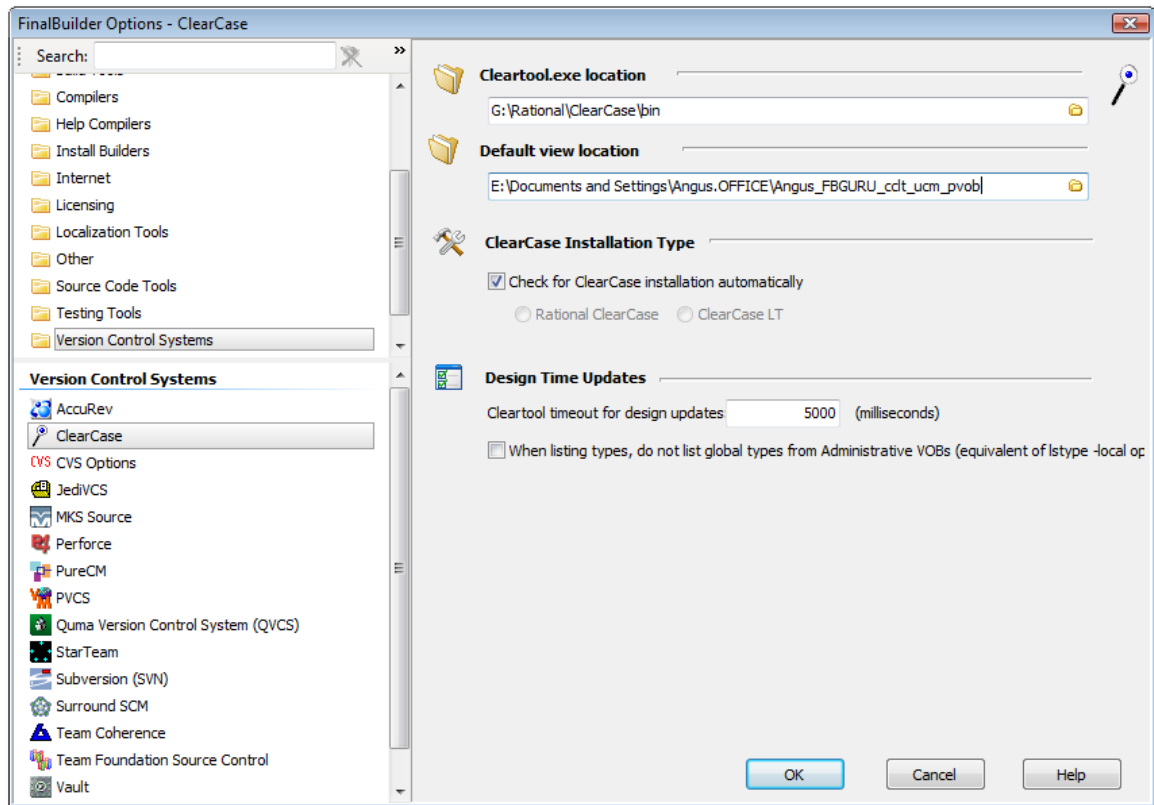
This action calls the StarTeam command line tool to Lock/Unlock files.

6.34.2.10 StarTeam Update Status Action

This action calls the StarTeam command line tool to Update the status of files.

6.34.3 ClearCase

The Rational ClearCase actions provide an interface via Rational's Cleartool. Each action wraps almost all the functionality of the relevant cleartool command. For unsupported commands, there is a generic Cleartool interface which allows Cleartool to be called directly.



Default View Location allows you to specify a "view context" directory to be used as the default for all ClearCase actions which use a view context.

Cleartool.exe location & ClearCase Installation Type should be set automatically on a computer with a ClearCase installation, although they can be overridden manually if desired. ClearCase Installation Type enables and disables some options which are only available in ClearCase or ClearCase LT, respectively.

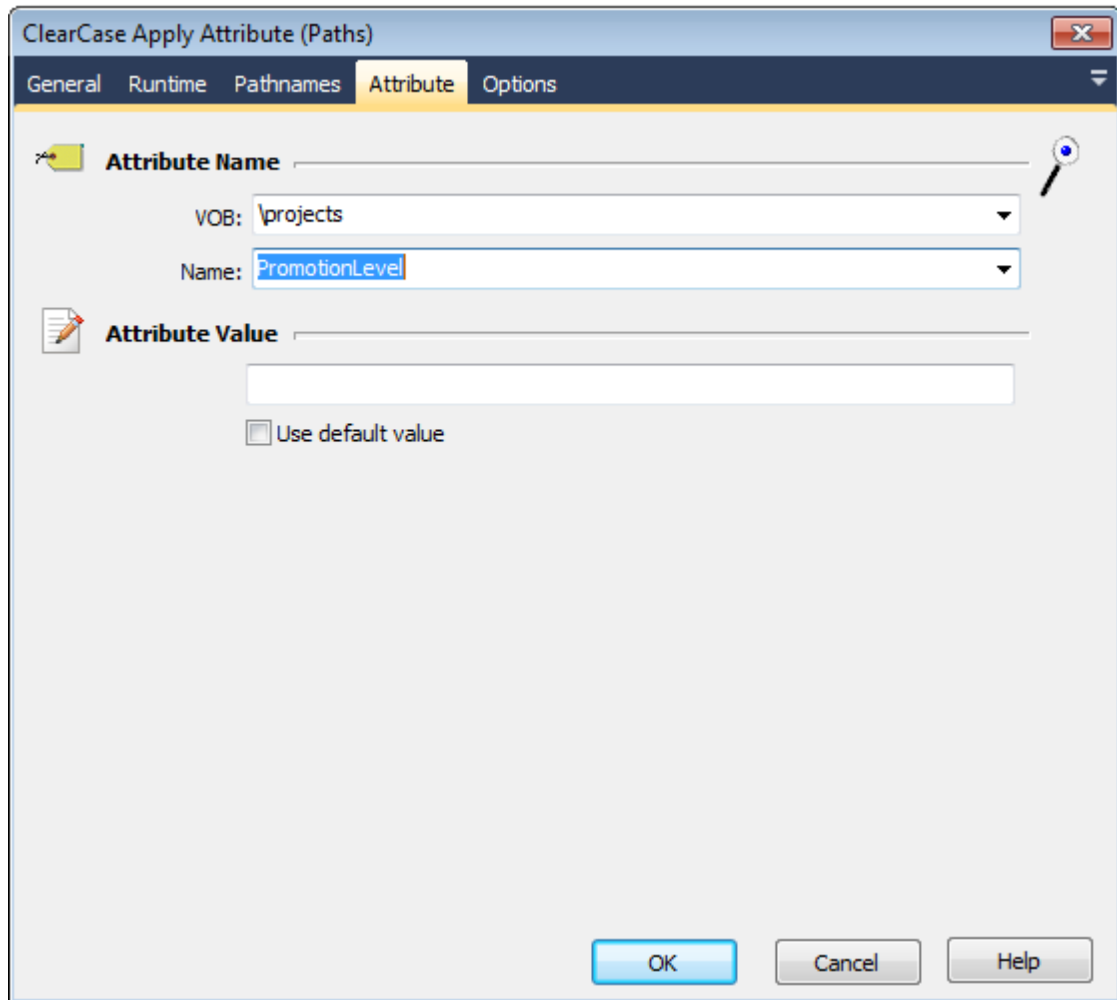
6.34.3.1 Base ClearCase

6.34.3.1.1 ClearCase Apply Attribute Action

Use this action to apply an attribute (with value) to one or more elements in a VOB.

For full details of the mkattr command, see the ClearCase manual page (type *cleartool man mkattr*.)

The Apply Attribute Action comes in two forms: Apply Attribute (Paths) and Apply Attribute (Objects.) The (Paths) form chooses the elements via ClearCase Pathnames, while the (Objects) form uses the object selector.



Attribute Name

Click on the VOB dropdown menu to see a list of available VOBs. Once a VOB has been selected, click on the Name dropdown menu to see a list of available attribute type names.

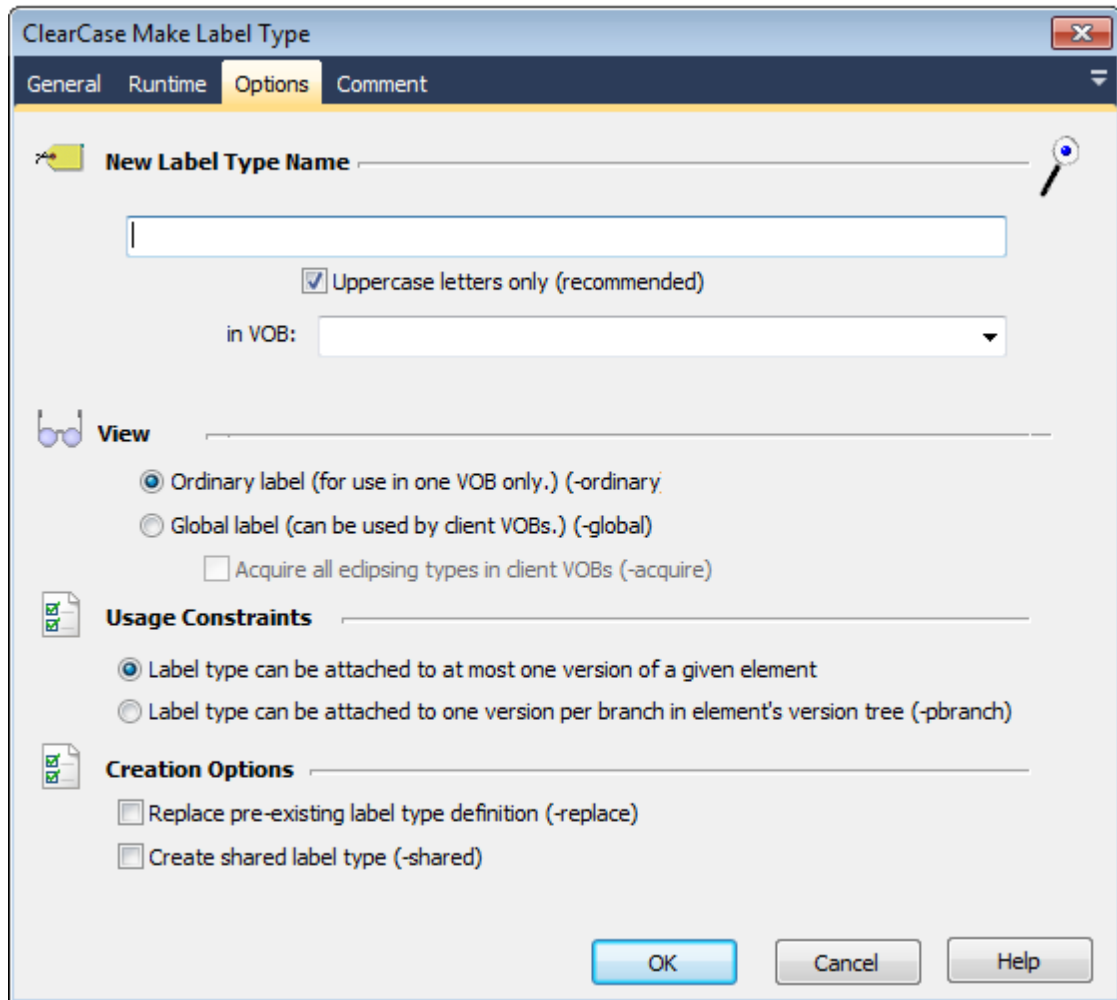
Once an attribute name has been chosen, FinalBuilder will try and determine the type and the default value (if applicable.)

6.34.3.1.2 ClearCase Apply Label Action

Use this action to apply a label to one or more elements in a VOB.

For full details of the `mklabel` command, see the ClearCase manual page (type `cleartool man mklabel`.)

The ClearCase Apply Label Action can use View- or Extended- pathnames.



Label Type Selector

If you have chosen a VOB, Click on the Label drop down menu to see a list of labels available in that VOB.

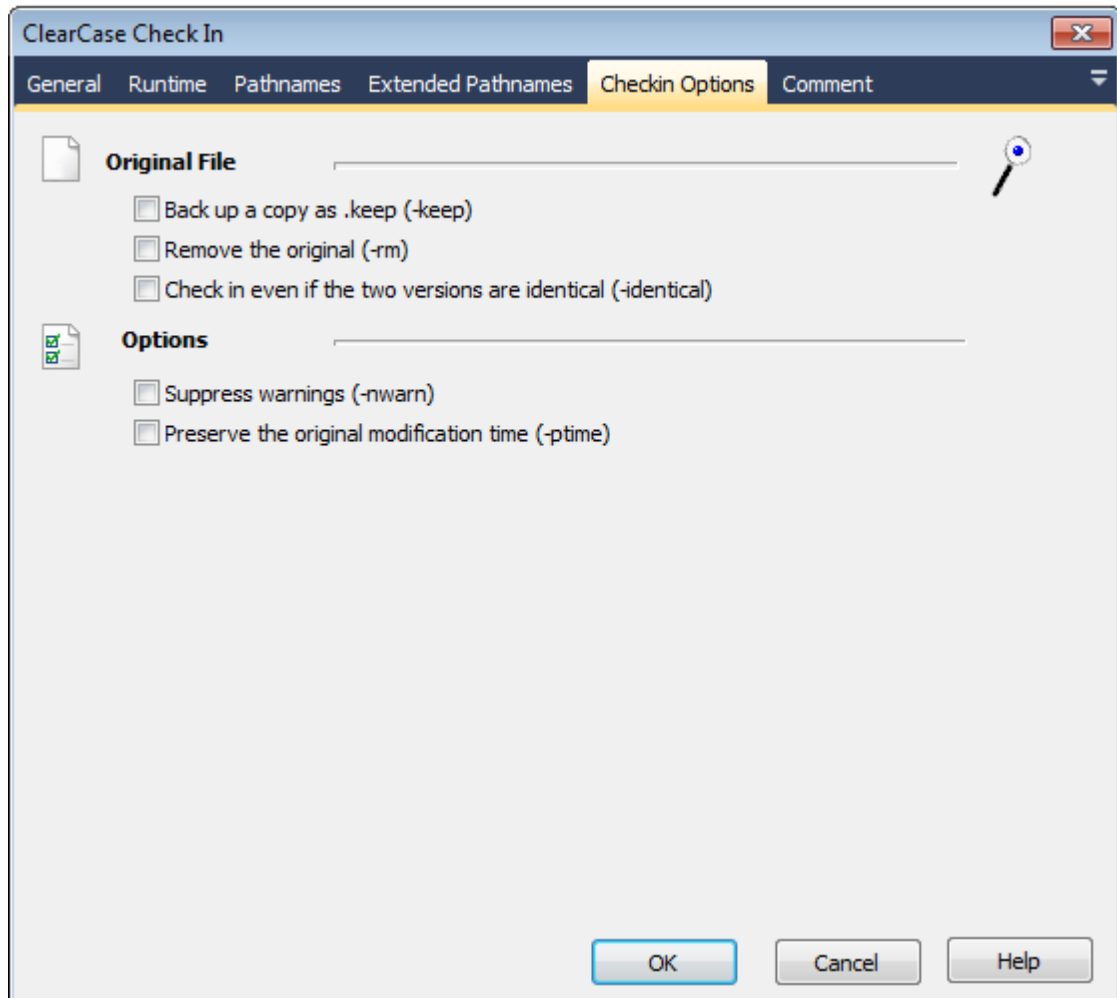
A VOB selector is not required if the Label exists in the default VOB for the view context.

6.34.3.1.3 ClearCase Check In Action

Use this action to check in elements (files or directories) to a VOB. To add new elements to a VOB, use the ClearCase Make Element action.

For full details of the checkin command, see the ClearCase manual page (type *cleartool man checkin*.)

The ClearCase Check In Action can use View- or Extended- pathnames.



Original File

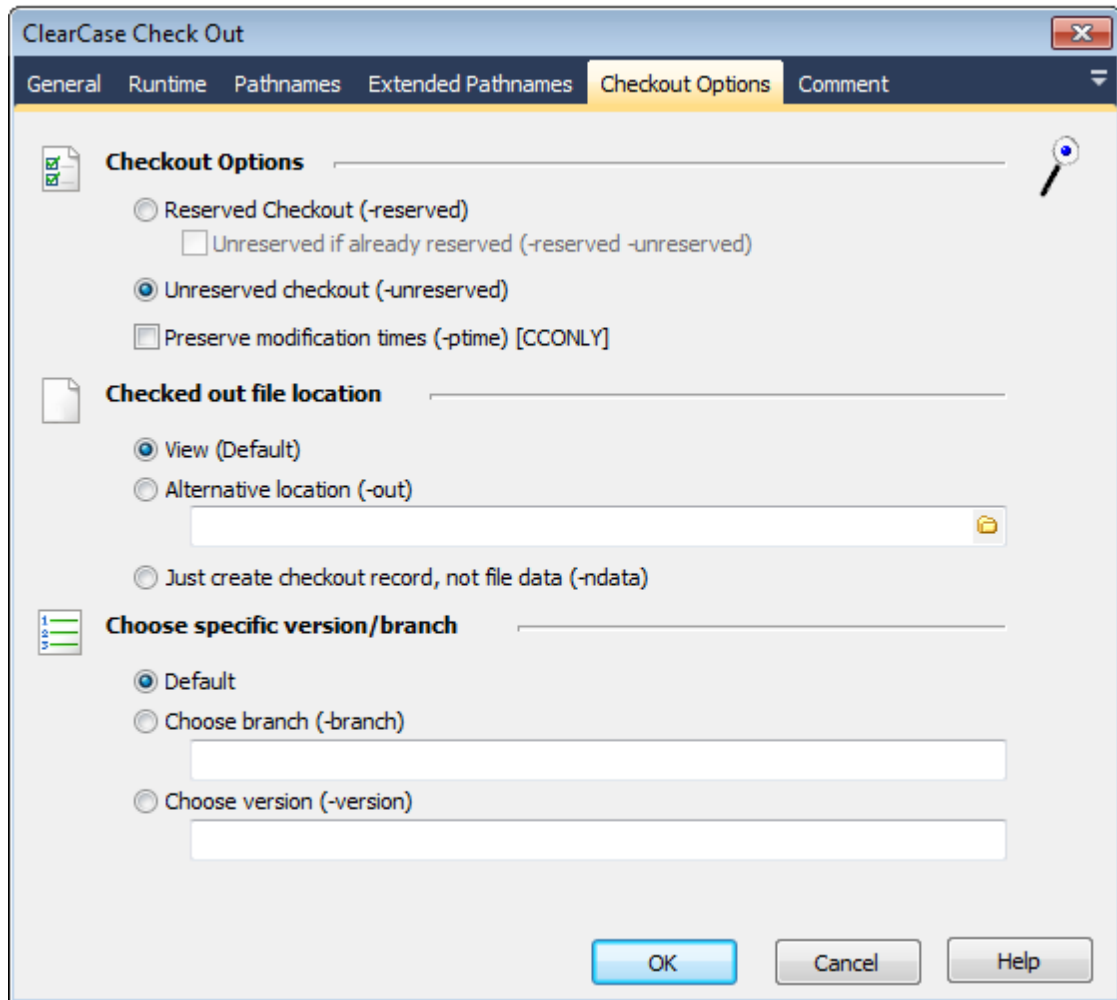
By default, in a dynamic view, cleartool deletes each view-private, checked-out file after using it to create a new version in the VOB. In a snapshot view, cleartool uses the checked-out file to create a new version, then loads the new version into the view.

6.34.3.1.4 ClearCase Check Out Action

Use this option to check out elements (files or directories) from VOBs.

For full details of the checkout command, see the ClearCase manual page (type *cleartool man checkout*.)

The ClearCase Check Out Action can use View- or Extended- pathnames.



Checkout Options

Checked out files can be reserved or unreserved. Only one user can hold a reserved checkout on a single element. See the manual page for more details.

Checked Out File Location

By default, files are checked out to the view they were accessed from.

Choose specific version/branch

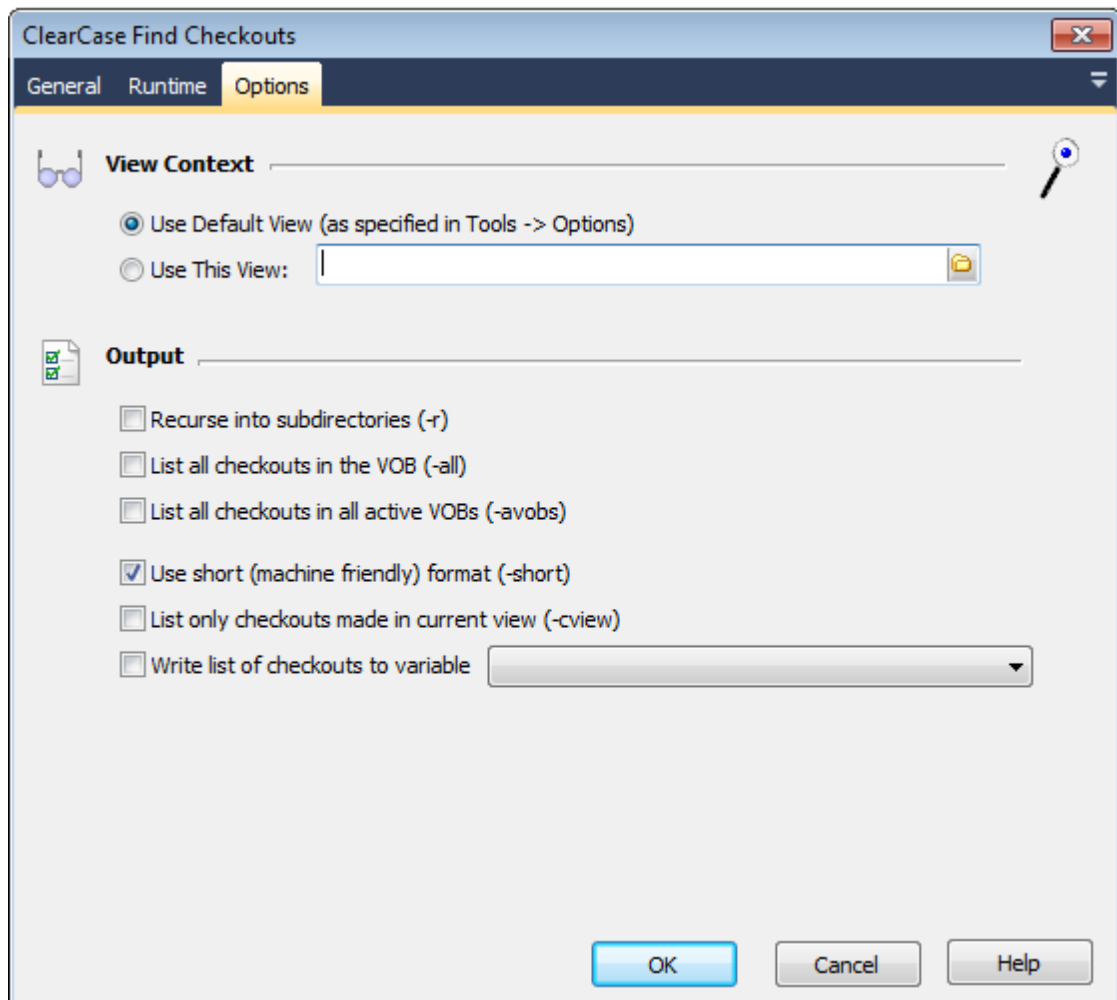
Different versions/branches to the one specified by the config spec can only be checked out from dynamic views. This option is not available on ClearCase LT.

6.34.3.1.5 ClearCase Find Checkouts Action

Use this action to find all the checked out elements from a view.

For full details of the find command, see the ClearCase manual page (type *cleartool man find*.)

The ClearCase Find Checkouts Action can use View- or Extended- pathnames.



The view to search in can be either the default view (specified on the Options panel) or a view specific to that action.

Output

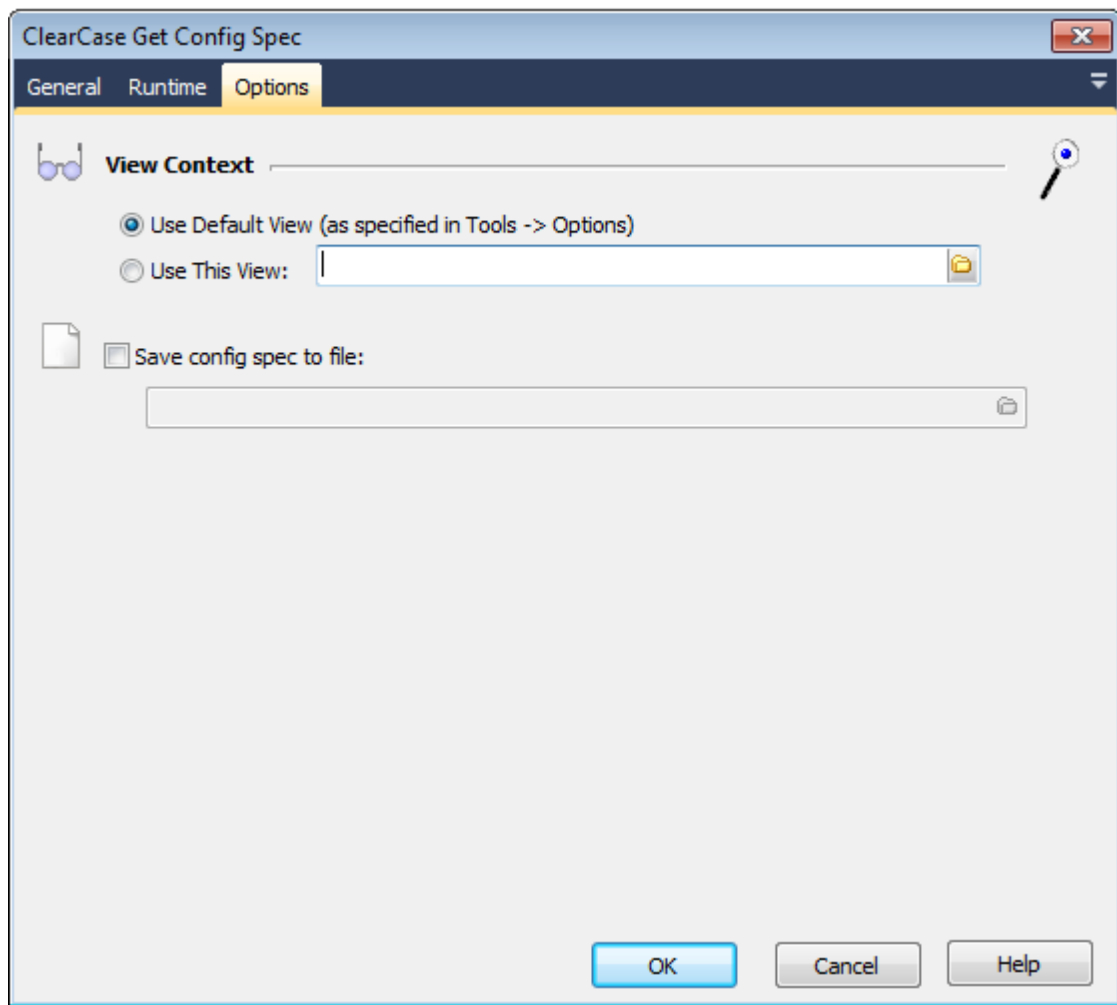
The list of checkouts can be either long (with details of the checkout) or a short summary of the checked out files.

The list of checkouts can also be written to a variable and used (for example) with a list iterator. If you are writing to a variable, it is recommended that you use the short (machine friendly) format.

6.34.3.1.6 ClearCase Get Config Spec Action

Use this action to output the config spec for a view.

For full details of the `catcs` command, see the ClearCase manual page (type `cleartool man catcs`.)



The config spec can be either that of the default view (specified on the Options panel) or a view specific to that action.

If the 'Save config spec to file' option is not checked, the config spec will be output to the log.

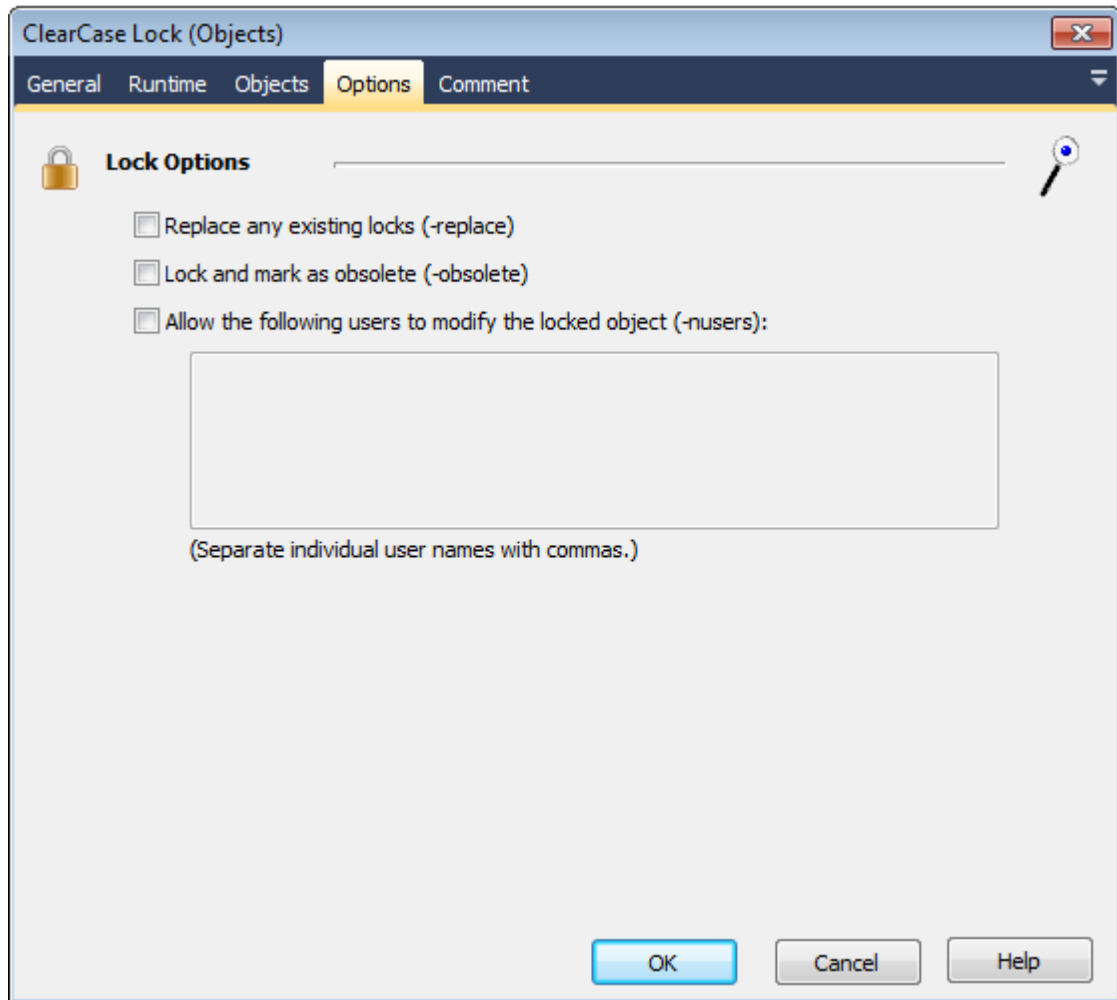
Config specs which have been saved to a file can be set with the Set Config Spec Action.

6.34.3.1.7 ClearCase Lock Action

Use this action to lock one or more elements in a VOB.

For full details of the lock command, see the ClearCase manual page (type *cleartool man lock*.)

The Lock Action comes in two forms: Lock (Paths) and Lock (Objects.) The (Paths) form chooses the elements to lock via ClearCase Pathnames, while the (Objects) form uses the object selector.

**Replace any existing locks**

The action will fail if an element is already locked and this option is unchecked.

Lock and mark as obsolete

Obsolete elements are no longer used in the VOB.

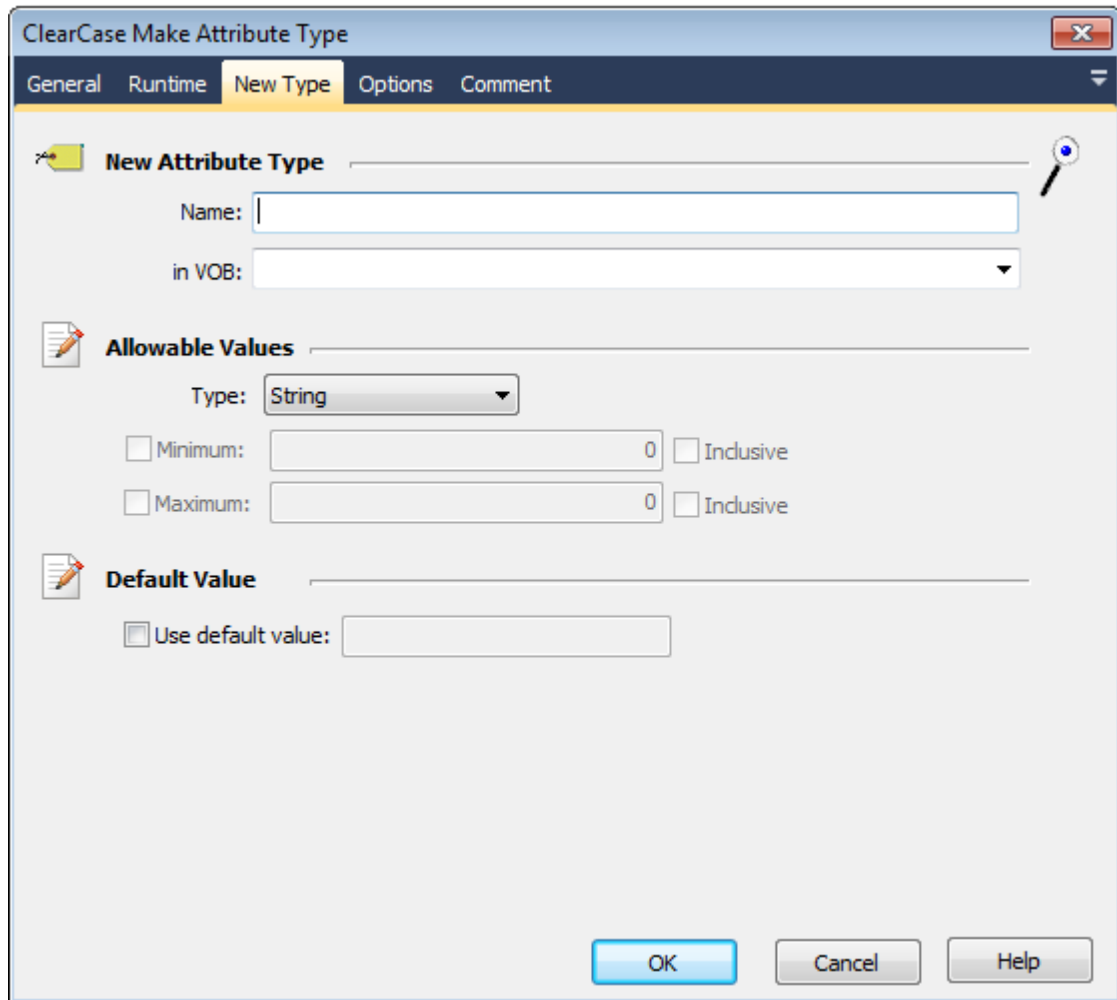
Allow the following users...

Locks can provide access control by allowing only certain users to modify the locked contents.

6.34.3.1.8 ClearCase Make Attribute Type Action

Use this action to make a new attribute type in a VOB. The new attribute type can then have values assigned to elements and components within that VOB.

For full details of the `mkatttype` command, see the ClearCase manual page (type *cleartool man mkatttype*.)



The image shows the 'ClearCase Make Attribute Type' dialog box. It has a title bar with a close button (X) and a menu bar with 'General', 'Runtime', 'New Type' (selected), 'Options', and 'Comment'. The 'New Type' tab is active, showing three sections: 'New Attribute Type', 'Allowable Values', and 'Default Value'. The 'New Attribute Type' section has a 'Name:' text box and an 'in VOB:' dropdown menu. The 'Allowable Values' section has a 'Type:' dropdown menu set to 'String', and two rows of 'Minimum:' and 'Maximum:' text boxes, each followed by a '0' and an 'Inclusive' checkbox. The 'Default Value' section has a 'Use default value:' checkbox and a text box. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

ClearCase Make Attribute Type

General Runtime **New Type** Options Comment

New Attribute Type

Name:

in VOB:

Allowable Values

Type:

☐ Minimum: ☐ Inclusive

☐ Maximum: ☐ Inclusive

Default Value

☐ Use default value:

OK Cancel Help

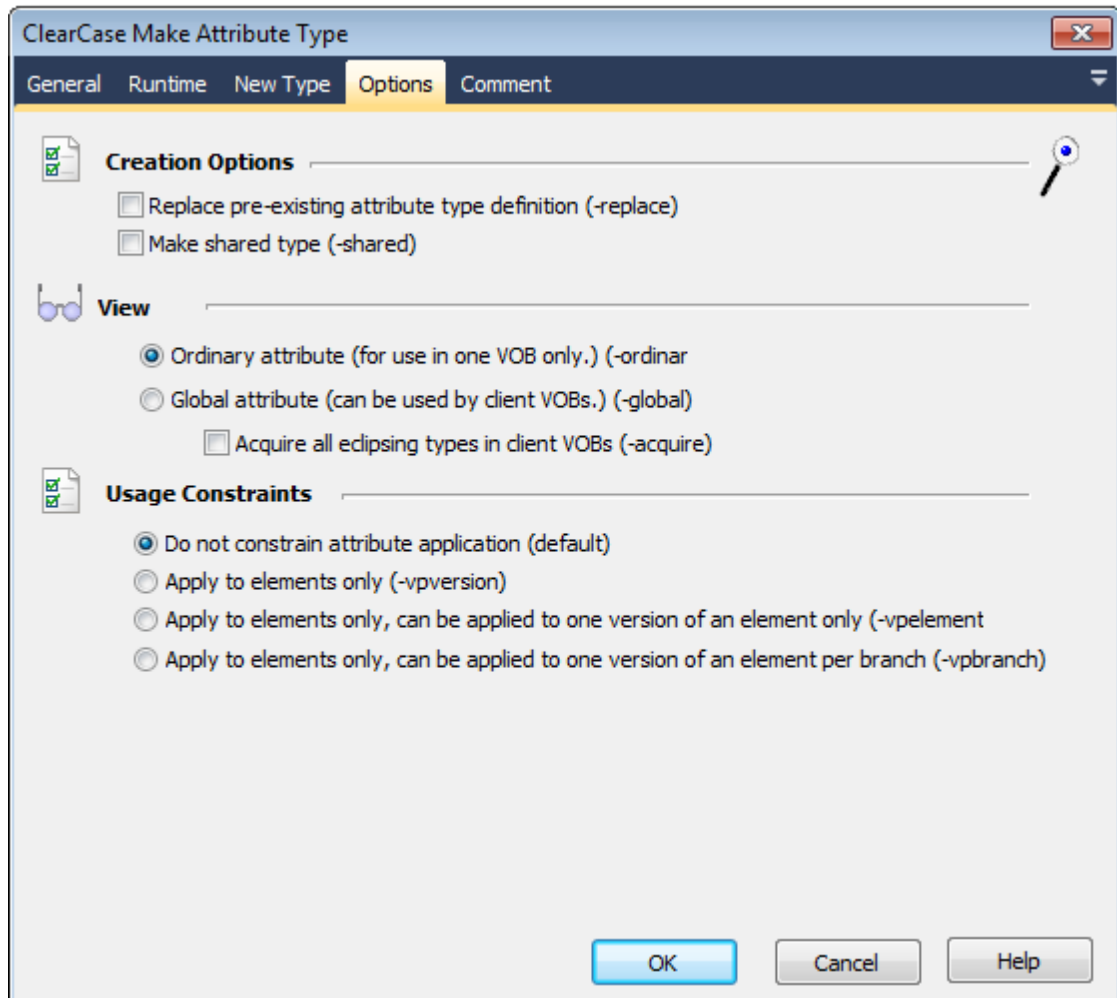
New Attribute Type

Click on the dropdown menu to see a list of VOBs.

Allowable Values

Attributes can be of type String, Integer, Real, Date or Opaque (a single unsigned byte.)

For cardinal types (all apart from String), minimum and maximum values can be specified.



Creation Options

If the attribute type already exists, 'Replace pre-existing...' must be checked or the action will fail at runtime.

View

The attribute can be scoped either for a single VOB (Ordinary) or for all client VOBS as well (Global.)

Usage Constraints

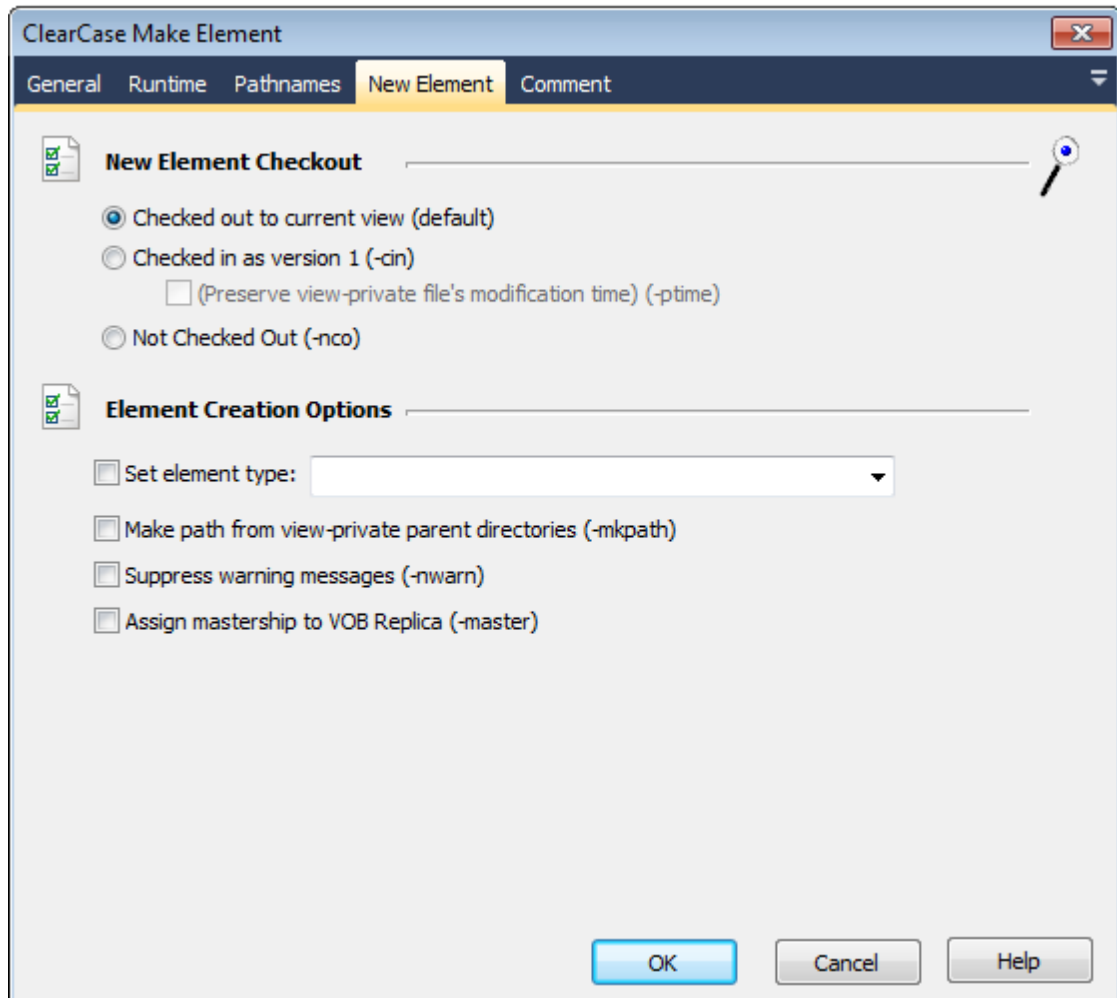
Application of the attribute type can be restricted in order to avoid ambiguity.

6.34.3.1.9 ClearCase Make Element Action

Use this action to make a new element (file, directory, etc.) The new element is created in both the VOB and the view directory.

For full details of the mkelem command, see the ClearCase manual page (type *cleartool man mkelem*.)

The ClearCase Make Element Action can only create elements specified by View-extended pathnames.



Element Creation Options

Set Element Type

The element type is not required, but allows ClearCase to automatically manage the contents of some files. Set the element type to 'Directory' if you are creating a directory. Click on the dropdown menu to get a list of all element types in all VOBs.

Make path from view-private parent directories

Use this option if the new element has a parent directory which also does not exist in the VOB.

Assign Mastership to VOB Replica

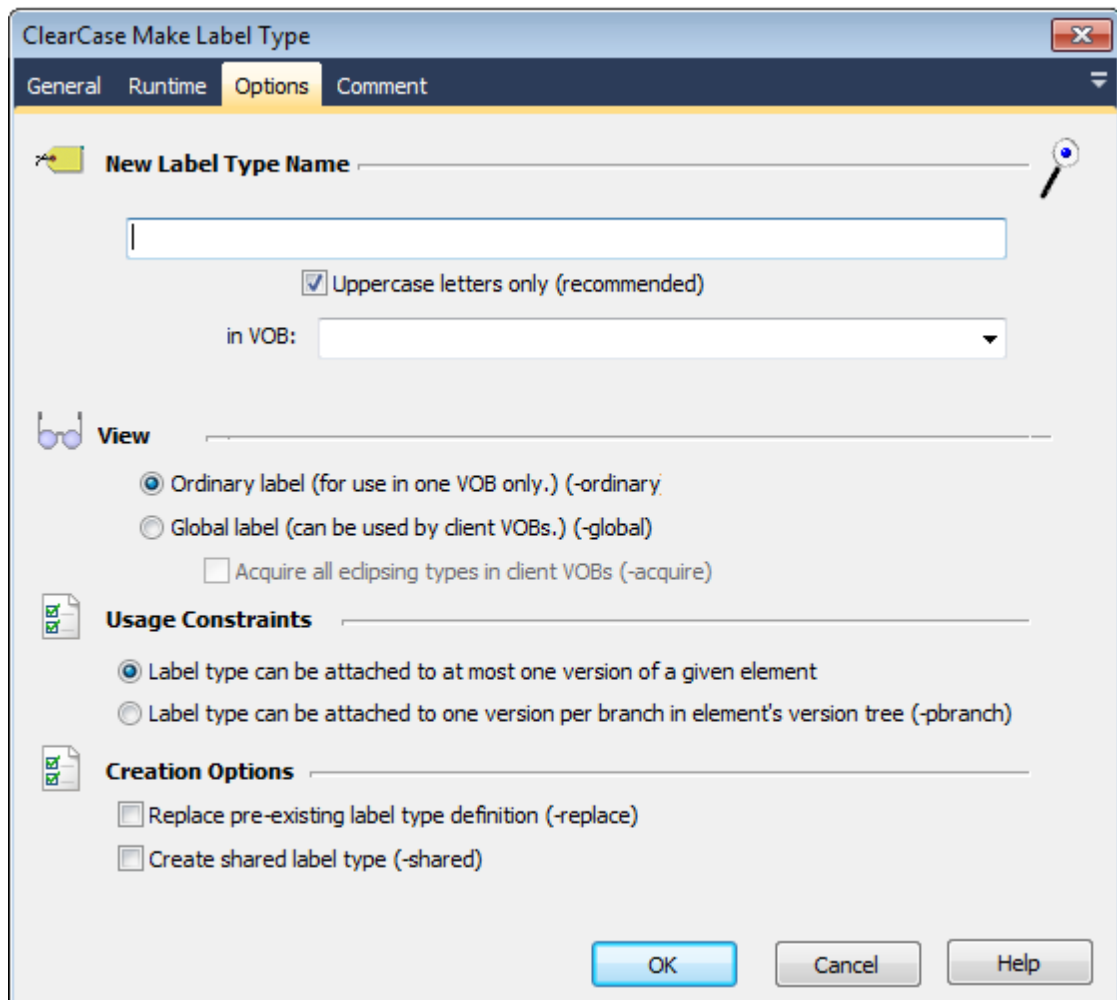
This option is not available in ClearCase LT.

6.34.3.1.10 ClearCase Make Label Type Action

Use this action to make a new label type in a VOB. The new label type can then be assigned to elements and components within that VOB.

For full details of the `mklbtype` command, see the ClearCase manual page (type

cleartool man mklbtype.)



New Label Type Name

By convention, and to avoid conflict with branch names, ClearCase Label Type Names are normally given in all uppercase letters.

Click on the dropdown menu to see a list of all available VOBs.

View

The label can be scoped either for a single VOB (Ordinary) or for all client VOBs as well (Global.)

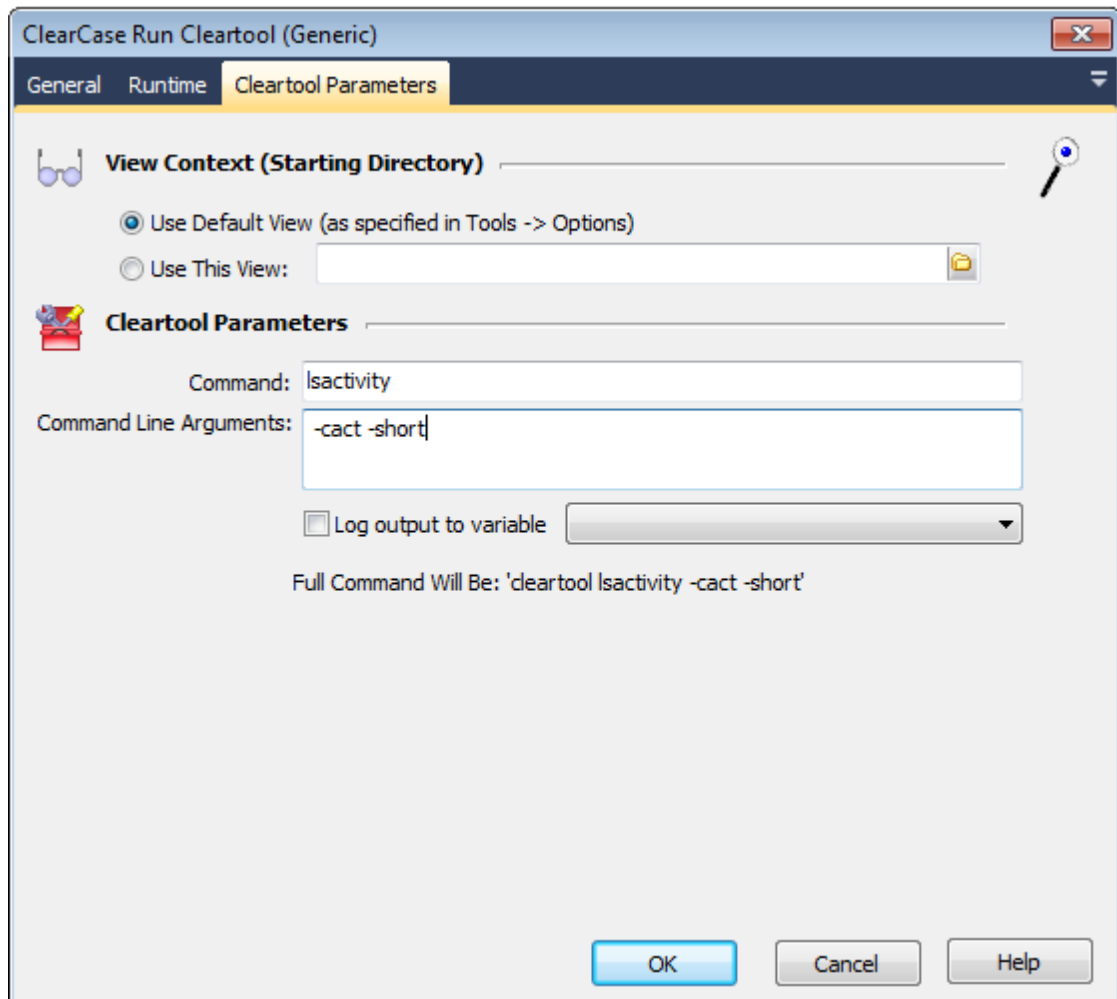
Usage Constraints

By default, and to avoid ambiguity, label types can be attached to only one version of a given element. This constraint can be relaxed to allow each branch to contain a labelled version.

6.34.3.1.11 ClearCase Run Cleartool (Generic) Action

The ClearCase Run Cleartool (Generic) action allows you to specify any cleartool command and any set of parameters to pass to cleartool.

For full details of Cleartool's capabilities, and a list of commands, type '*cleartool man*'



When the action is run, its behaviour will correspond to the following DOS commands:

```
cd <<View Context (Starting Directory)>>
cleartool <<Command>> <<Command Line Arguments>>
```

A preview of the full command string is available on the property page (see above.) All output will be recording in the log. The action will fail if cleartool fails (ie returns a non-zero return code.)

Log output to variable

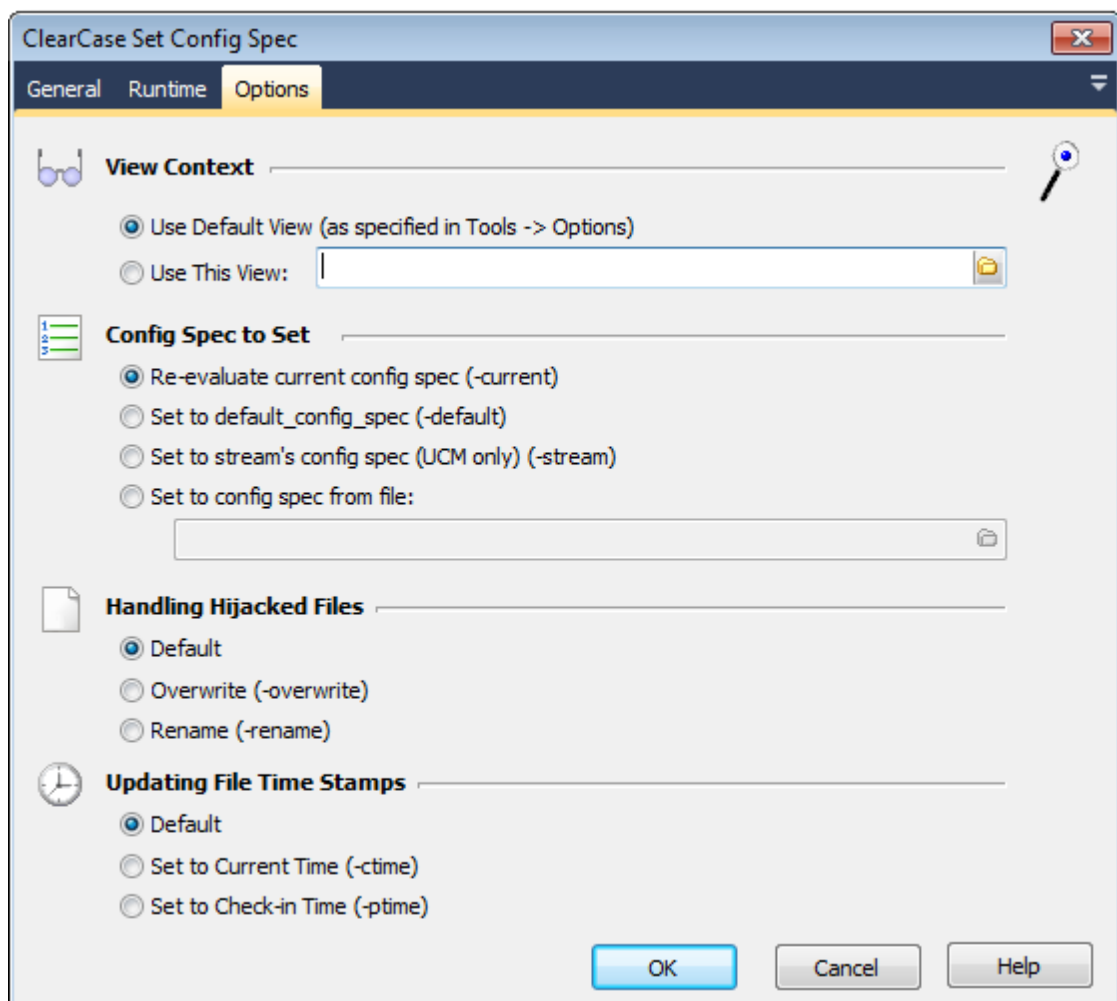
Log output to variable works in a similar way to the 'Log output to Variable' on the Properties tab, except the Cleartool-specific "Log output" only saves the cleartool output and trims any leading and trailing whitespace. This means that, in the example above, the variable 'Output' will be written with the exact string of the current activity name, ready to be passed into another command.

Note: If you find you are using a certain command very frequently, and would like to see an action based on that command, then please [email us](#) and request it!

6.34.3.1.12 ClearCase Set Config Spec Action

Use this action to set the config spec for a view.

For full details of the setcs command, see the ClearCase manual page (type *cleartool man setcs*.)



The config spec can be either that of the default view (specified on the Options panel) or a view specific to that action.

The config spec that is set can be one of:

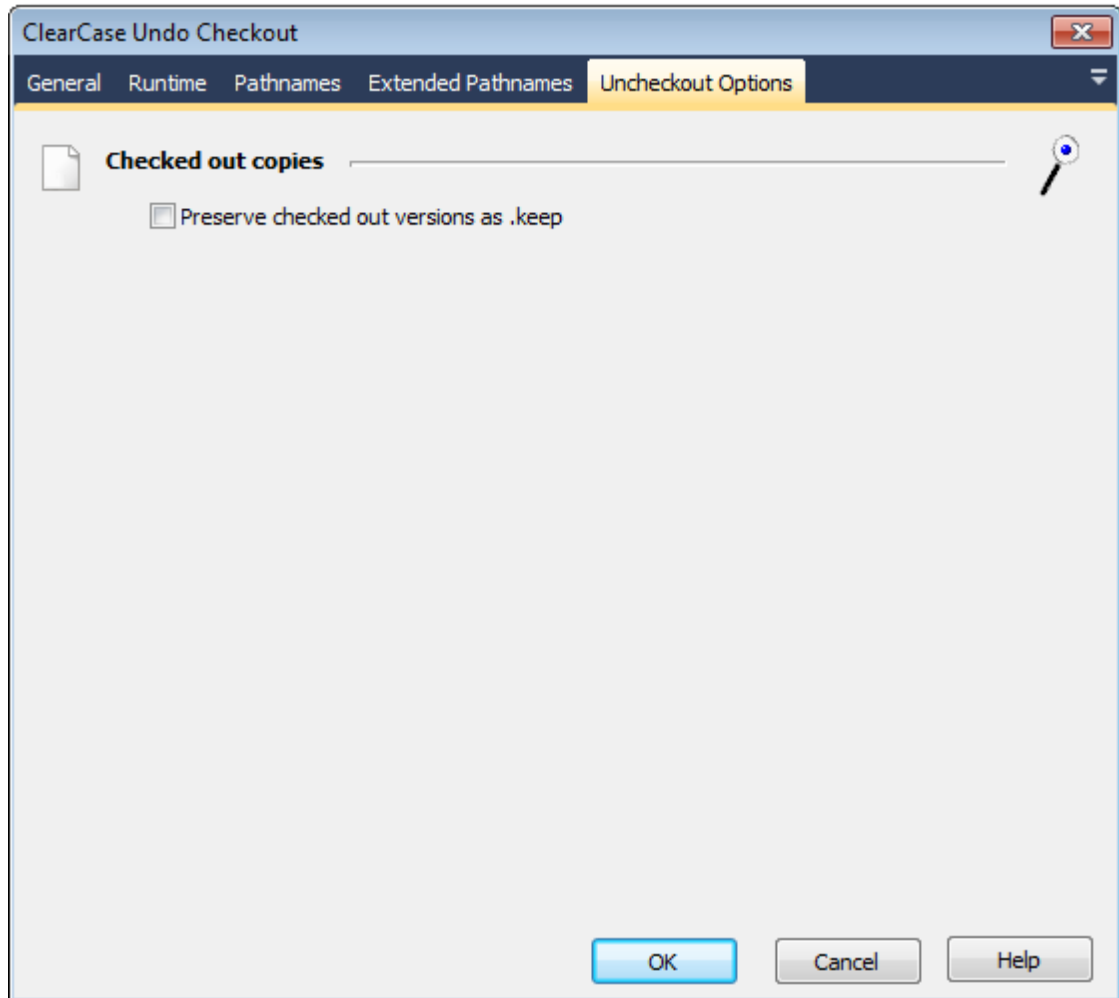
- The current config spec (which is re-evaluated.)
- The default config spec for that view
- A config spec given in a file
- The config spec for the View's UCM Stream

6.34.3.1.13 ClearCase Undo Checkouts Action

Use this action to undo the check out of some checked out files. For users using UCM, there is also a UCM Undo Checkouts for Activity action..

For full details of the uncheckout command, see the ClearCase manual page (type *cleartool man uncheckout.*)

The ClearCase Undo Checkouts Action can use View- or Extended- pathnames.



Checked out copies can be deleted or preserved as *filename.keep*.

6.34.3.1.14 ClearCase Unlock Action

Use this action to unlock one or more elements in a VOB.

For full details of the unlock command, see the ClearCase manual page (type *cleartool man unlock.*)

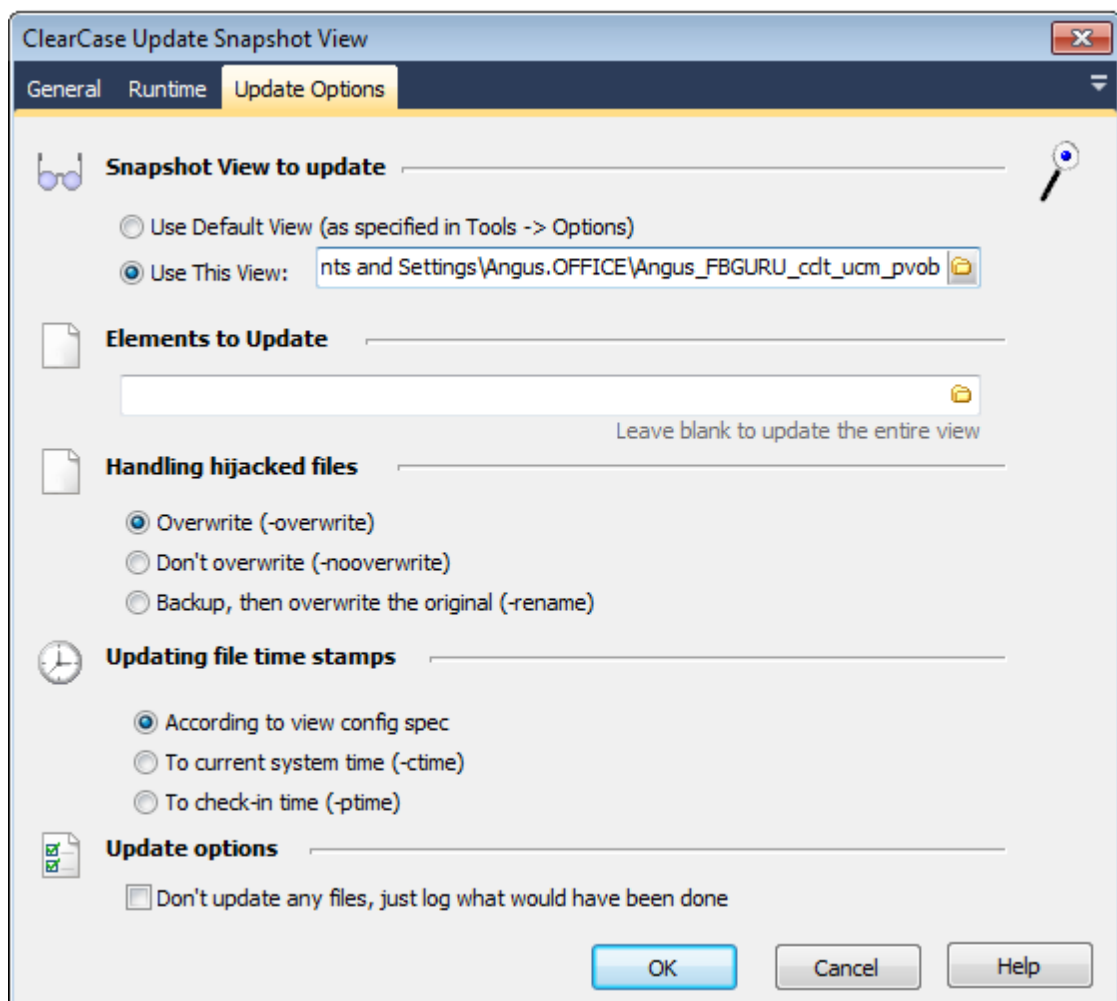
The Unlock Action comes in two forms: Unlock (Paths) and Unlock (Objects.) The (Paths) form chooses the elements to lock via ClearCase Pathnames, while the (Objects) form uses the object selector.

There are no specific options for this action.

6.34.3.1.15 ClearCase Update Snapshot View Action

Use this action to update a ClearCase LT Snapshot View (this action is not necessary when using ClearCase with MVFS.)

For full details of the update command, see the ClearCase manual page (type *cleartool man update*.)



The snapshot view to update can be either the default view (specified on the Options panel) or a view specific to that action.

Handling Hijacked Files

Hijacked files can either be overwritten with the copy from the VOB, left alone, or backed up (to *filename.keep*.) See the manual page for complete details.

Updating File Time Stamps

Updated files can be updated to local system time, or kept with the timestamp from the VOB.

Update Options

Update can also be used to just log the differences between the VOB and the current view directory.

6.34.3.2 ClearCase Object Selector

The Object Selector allows some ClearCase actions to target abstract ClearCase objects.

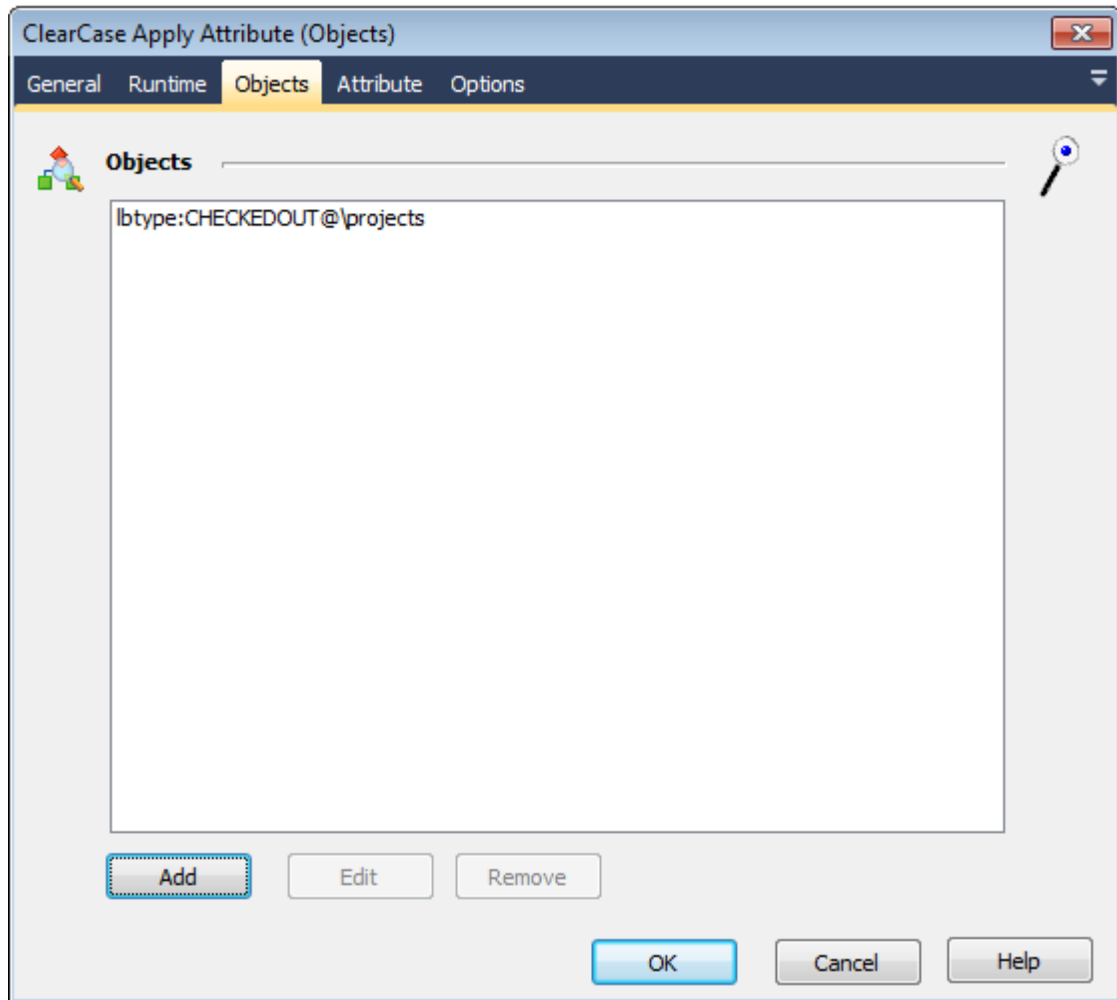
Objects that can be targeted include:

- VOBs
- Attribute Types
- Branches
- Elements
- Hyperlinks
- Labels
- Triggers
- Pools
- Object IDs

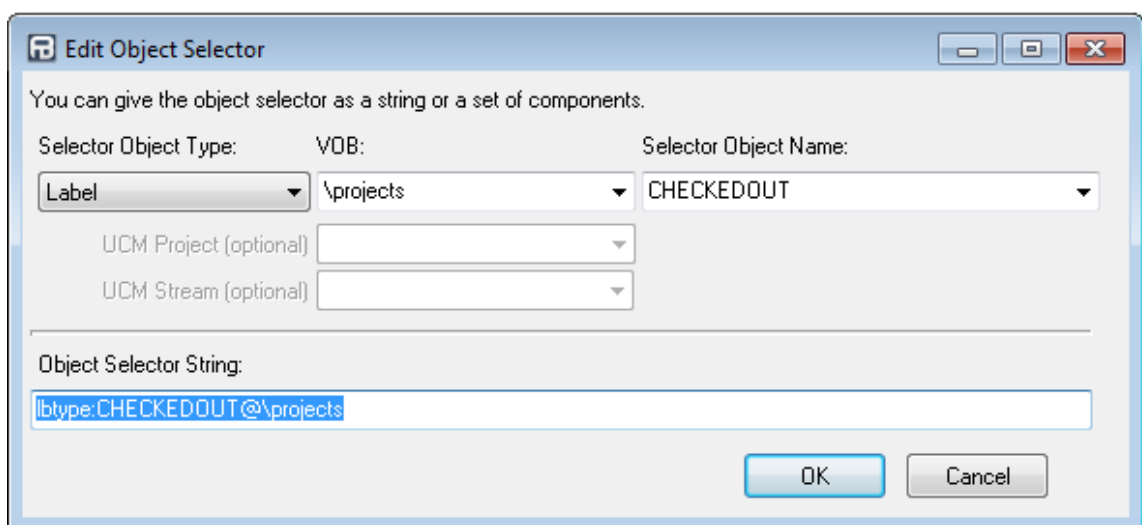
Also,

- UCM Activities
- UCM Baselines
- UCM Components
- UCM Folders
- UCM Projects
- UCM Streams

In the screenshot below, a label type has been selected to apply an attribute:



Clicking the Add button brings up the object selection dialog:



Object selectors can be specified as an object selector string (bottom row) or as a combination of constituent components (top row.) 'Object Name' dropdown menu in

the top row is context sensitive (ie its contents will depend on what VOB name and object type has been selected.)

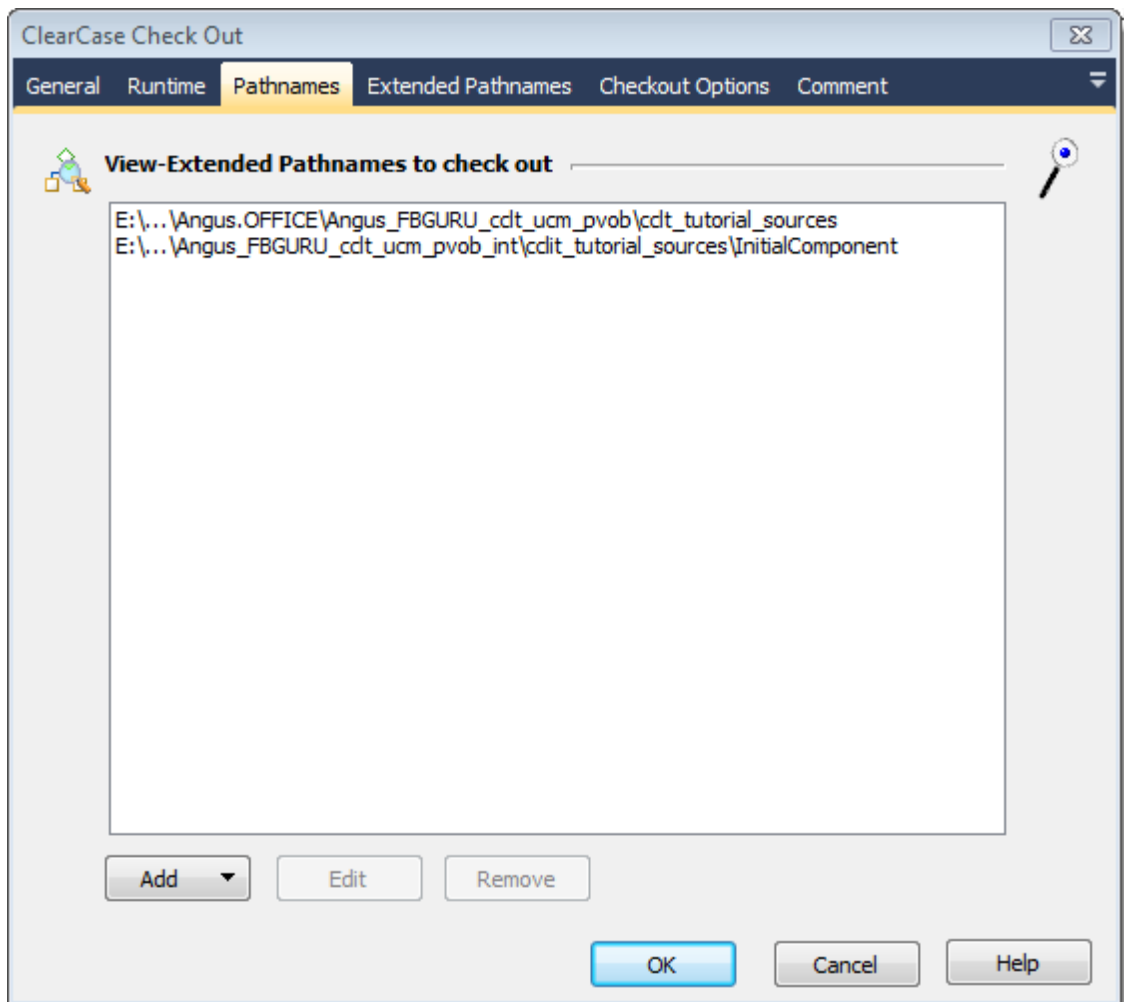
Typing a selector into the bottom edit field changes the contents of the top fields automatically, and vice versa.

6.34.3.3 ClearCase Pathnames

When choosing elements from a ClearCase VOB, selections can be made via View-Extended, Version-Extended or VOB-Extended pathnames.

For a full discussion of Pathnames in ClearCase, see the Pathnames section of the ClearCase manual (run *cleartool man pathnames_ccase*).

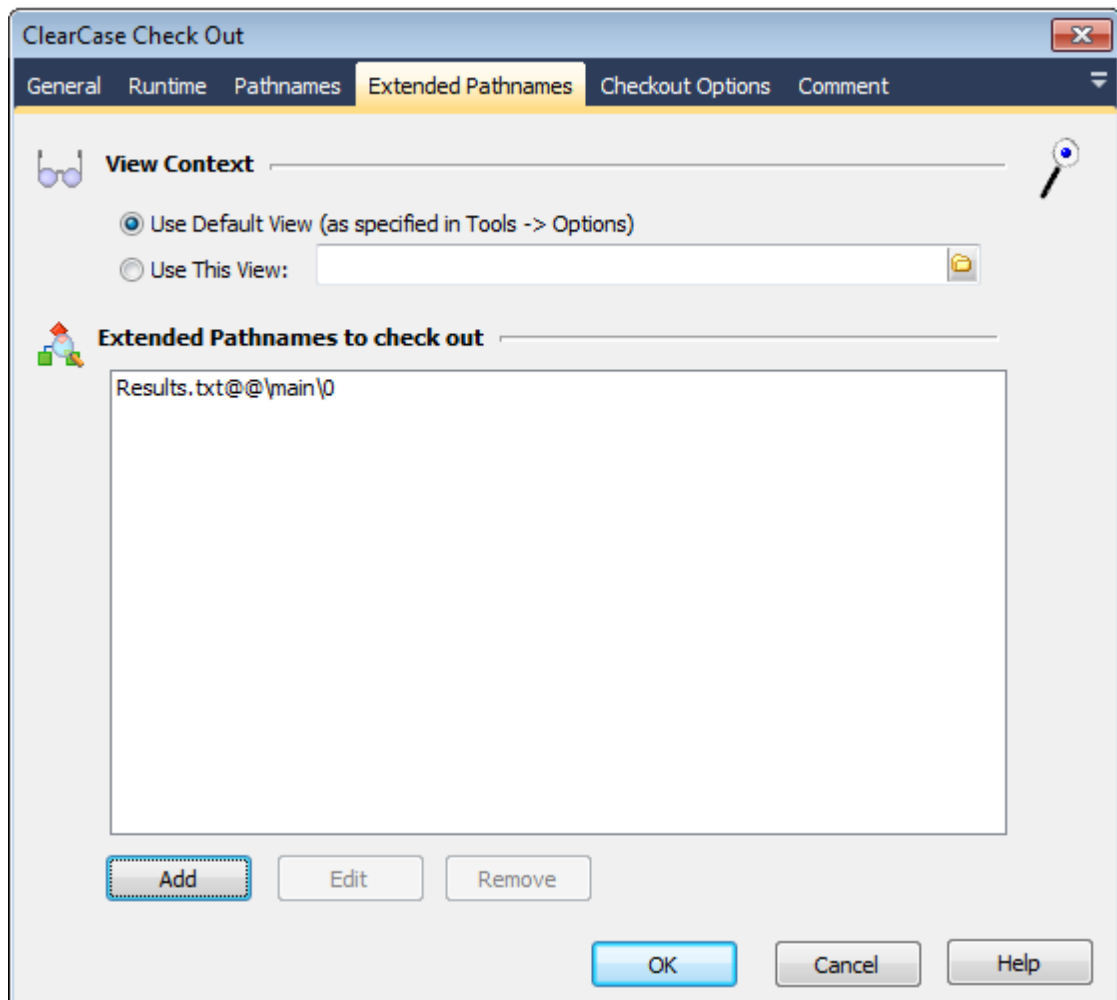
View-Extended Pathnames can be found under the '**Pathnames**' tab:



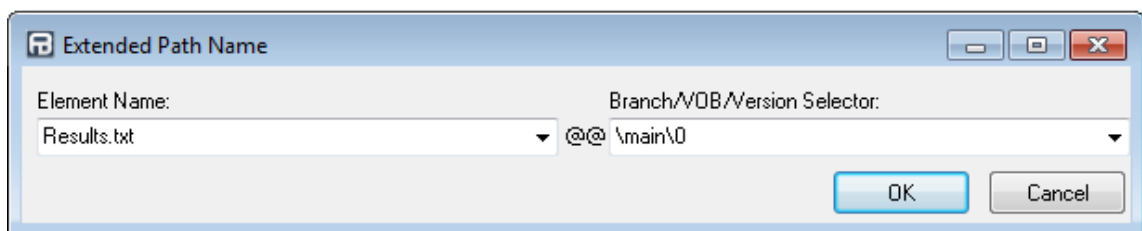
Hover the mouse over a pathname to see the full path.

Version- and VOB- Extended Pathnames can be found under the '**Extended Pathnames**' tab:

Note: Extended Pathnames are not available in ClearCase LT.



Click on Add to specify extended pathnames to act on:



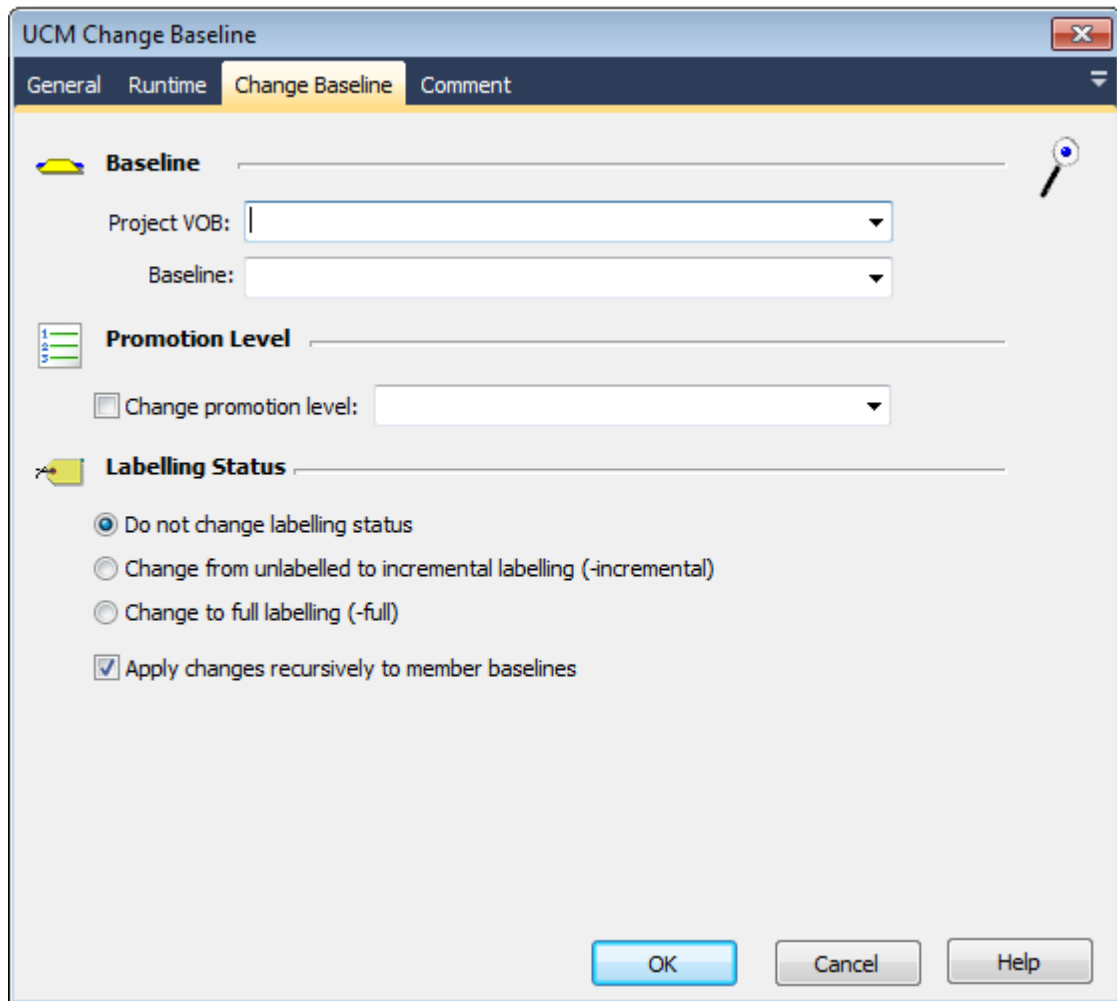
Click on the drop downs to see a list of elements for that view, or a list of available Branches, VOBs or Versions for each element.

6.34.3.4 UCM

6.34.3.4.1 UCM Change Baseline

Use this action to change or promote a UCM baseline.

For full details of the `chbl` command, see the ClearCase manual page (type *cleartool man chbl*.)



Baseline

Specify a UCM project VOB and a related baseline to change.

Promotion Level

Optionally, increase the promotion level of the baseline.

Labelling Status

You can update an unlabelled baseline to an incremental baseline, or a non-full labelled baseline to a full labelled baseline. If the baseline is already incremental (or fully)

labelled then this option is ignored.

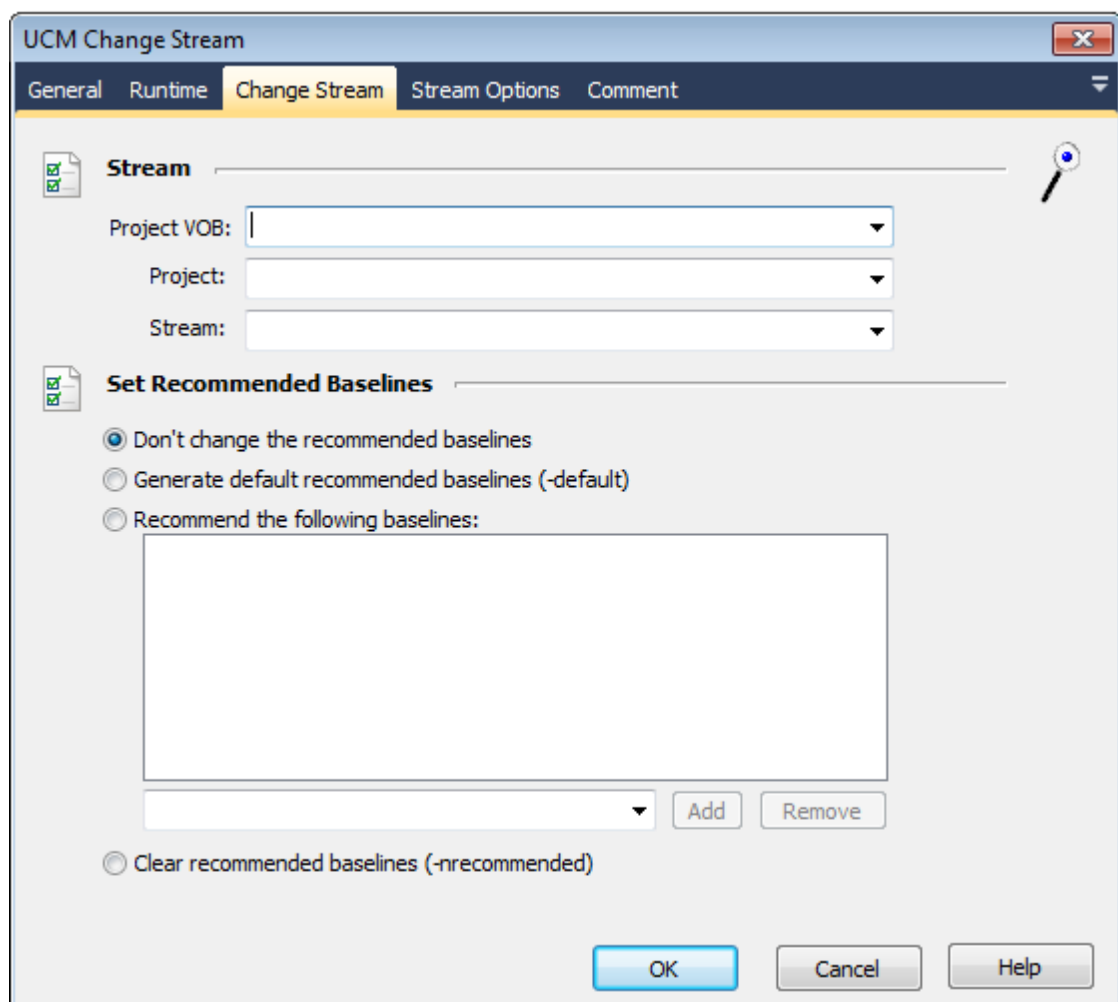
Apply Changes Recursively...

If this option is checked, the labelling status of member baselines will be updated as well.

6.34.3.4.2 UCM Change Stream

Use this action to change properties of a UCM stream.

For full details of the `chstream` command, see the ClearCase manual page (type *cleartool man chstream*.)



Stream

Specify the stream and the stream's base project VOB.

Generate Default recommended baselines

Optionally, you can set the recommended baselines to their recommended defaults (based on the promotion level criteria.)

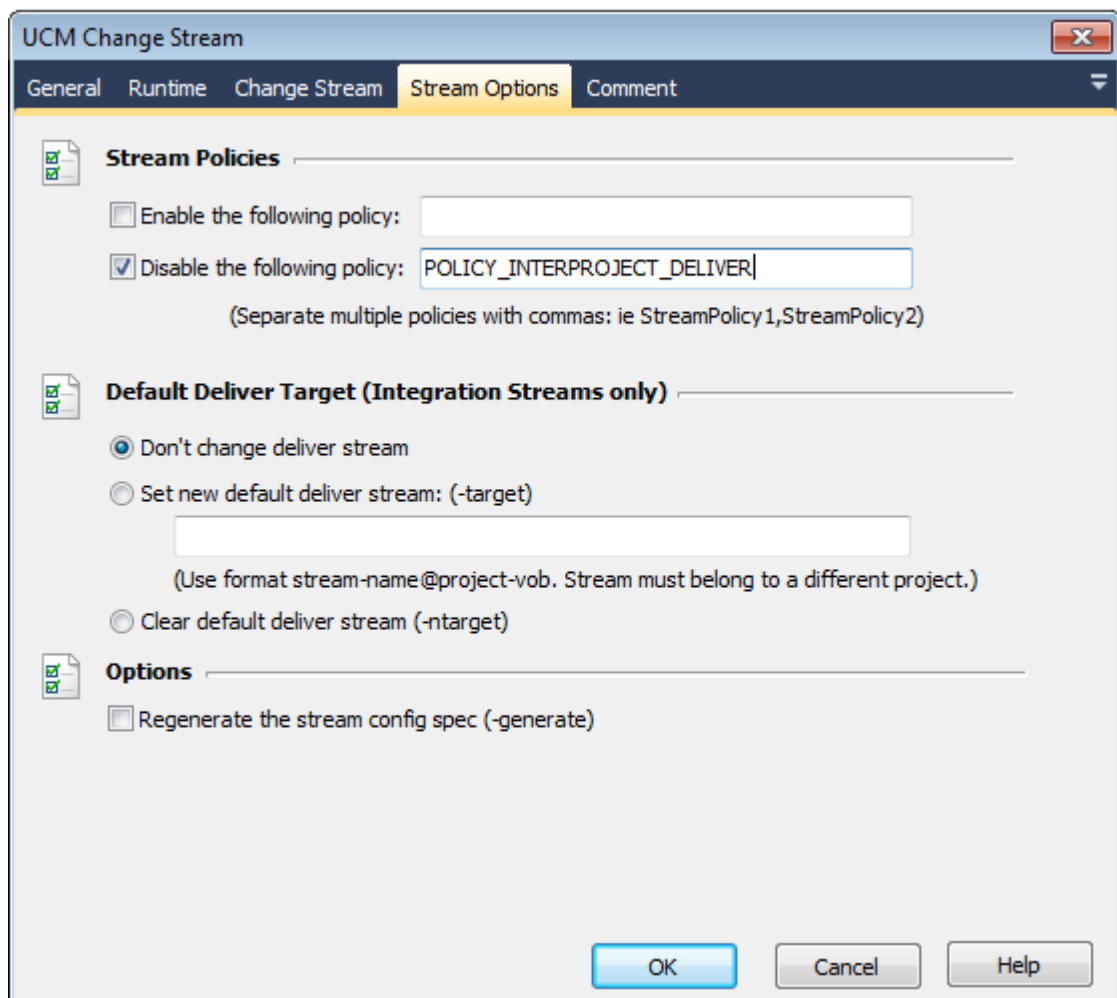
Recommend the following baselines

Specify a full list of recommended baselines for the stream, in the form *baseline@vob-selector*. Click on the dropdown menu to see a list of baselines created in the specified stream.

Click 'Add' to add the baseline name from the combo box, or 'Remove' to delete the currently selected baseline.

Clear Recommended Baselines

Remove all recommended baselines for this stream.



The image shows a screenshot of the 'UCM Change Stream' dialog box. The dialog has a title bar with a close button. Below the title bar is a tabbed interface with four tabs: 'General', 'Runtime', 'Change Stream', and 'Stream Options' (which is currently selected). The 'Stream Options' tab contains three sections:

- Stream Policies**: This section has two checkboxes. The first is 'Enable the following policy:' followed by an empty text box. The second is 'Disable the following policy:' followed by a text box containing 'POLICY_INTERPROJECT_DELIVER'. Below this is a note: '(Separate multiple policies with commas: ie StreamPolicy1,StreamPolicy2)'.
- Default Deliver Target (Integration Streams only)**: This section has three radio buttons. The first is 'Don't change deliver stream' (which is selected). The second is 'Set new default deliver stream: (-target)' followed by an empty text box. Below this is a note: '(Use format stream-name@project-vob. Stream must belong to a different project.)'. The third is 'Clear default deliver stream (-ntarget)'.
- Options**: This section has one checkbox: 'Regenerate the stream config spec (-generate)'.

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Stream Policies

Specify one or more policies to enable or disable. It is possible to both enable and disable different policies in the same comment.

If you want to specify more than one policy to enable or disable, separate each with a comma (no spaces.)

Default Deliver Target

This option is for integration streams only. You can specify a new deliver stream for the integration stream, or alternative clear the default deliver stream.

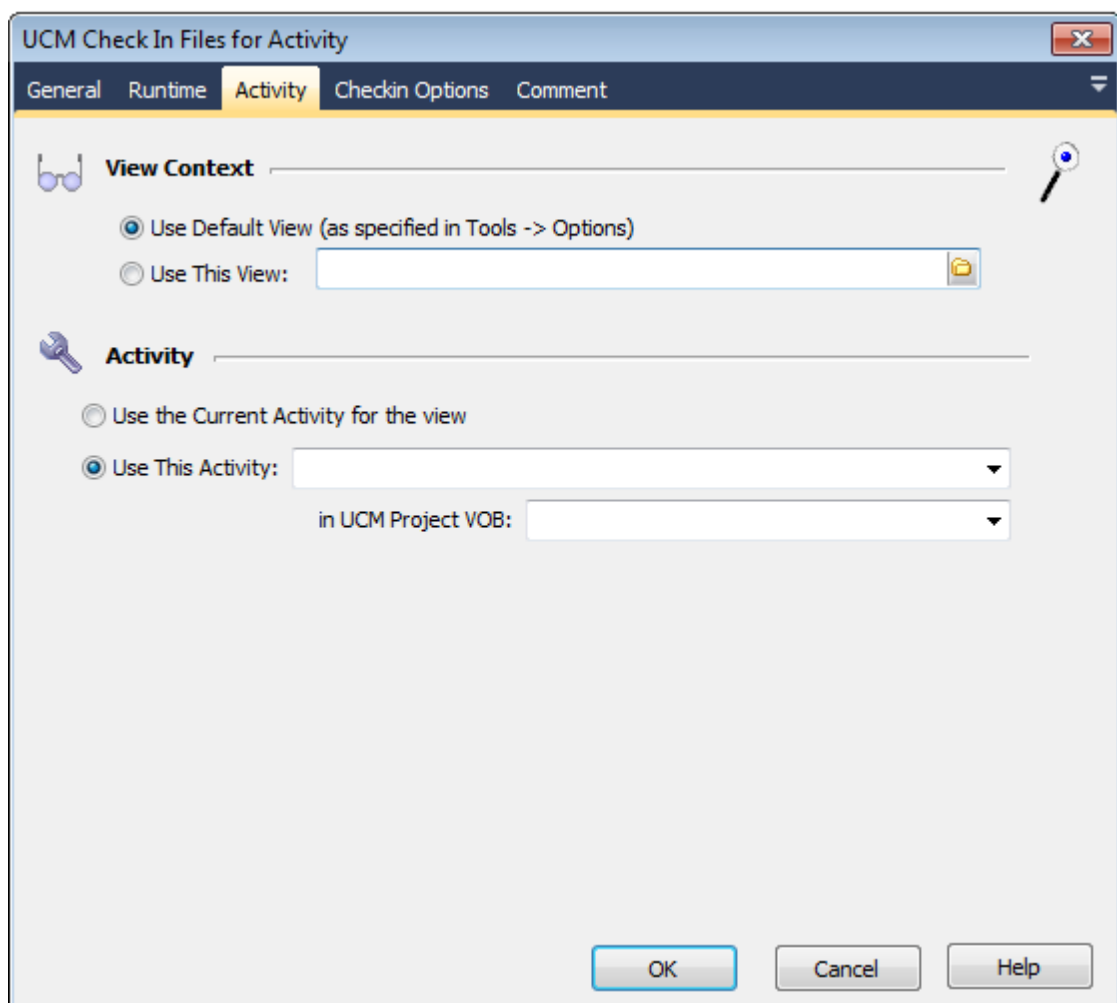
Regenerate the stream config spec

If this option is checked, it forces the stream to regenerate and reevaluate its config spec.

6.34.3.4.3 UCM Check In Files For Activity Action

Use this action to check in all files checked out to a UCM activity.

For full details of the checkin command, see the ClearCase manual page (type *cleartool man checkin*.)



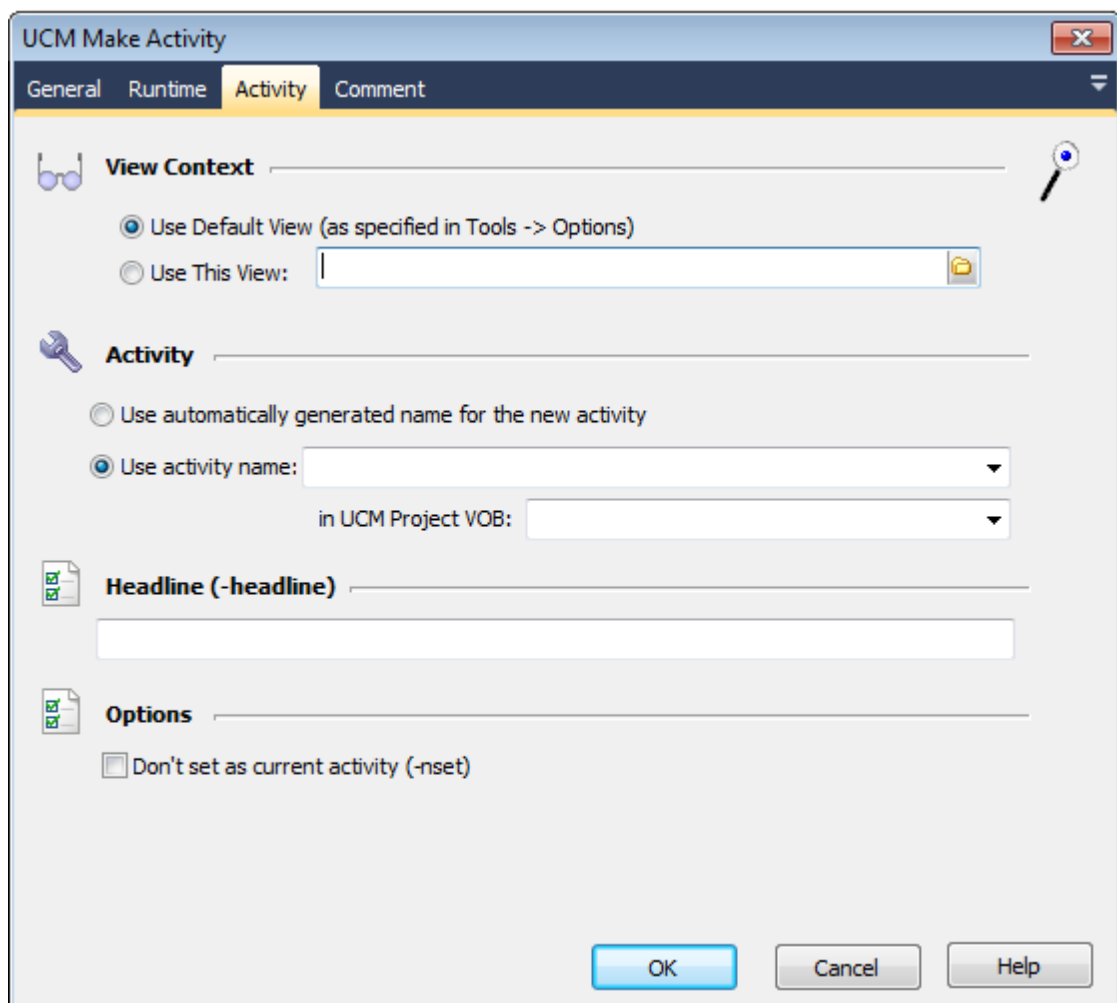
The stream to use can be taken from the default view (specified on the Options panel) or a view specific to that action.

The activity to check in files for can be either the current activity for that view, or a specific activity chosen from a UCM Project VOB.

6.34.3.4.4 UCM Make Activity Action

Use this action to create a new activity in a UCM stream. By default, the new activity is set as the new current activity for that stream.

For full details of the mkactivity command, see the ClearCase manual page (type *cleartool man mkactivity*.)



The stream to use can be taken from the default view (specified on the Options panel) or a view specific to that action.

Activity

ClearCase can automatically generate names for new activities. If a headline is specified, the name will be based on the headline text.

Headline

(Optional) a human-readable name for the activity.

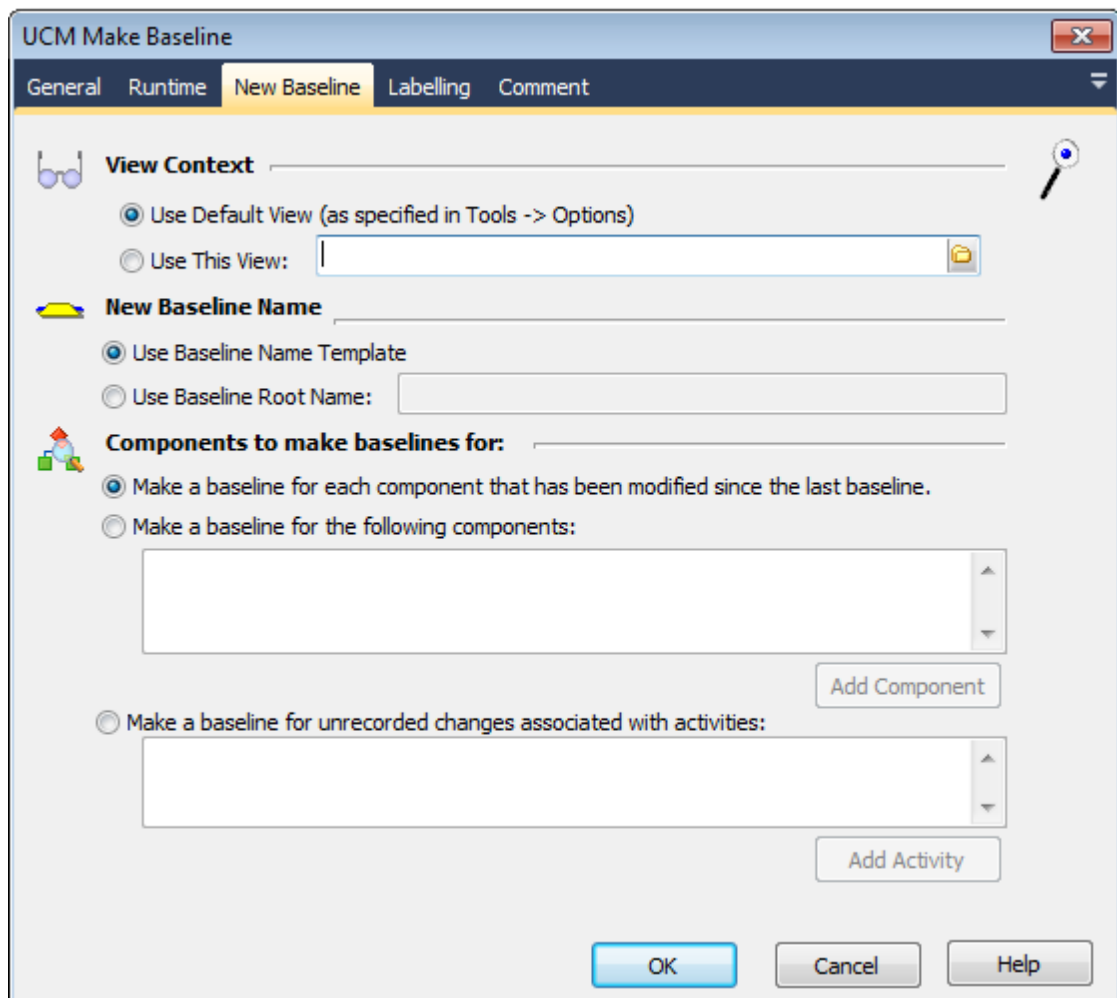
Options

By default, the newly created activity is set as the current activity for the stream. To avoid this, check the "Don't Set..." check box. The current activity can also be set with the UCM Set Activity Action.

6.34.3.4.5 UCM Make Baseline Action

Use this action to create a new baseline in a UCM stream.

For full details of the mkbl command, see the ClearCase manual page (type *cleartool man mkbl*.)



The baseline can be created in either the default view (specified on the Options panel) or a view specific to that action.

New Baseline Name

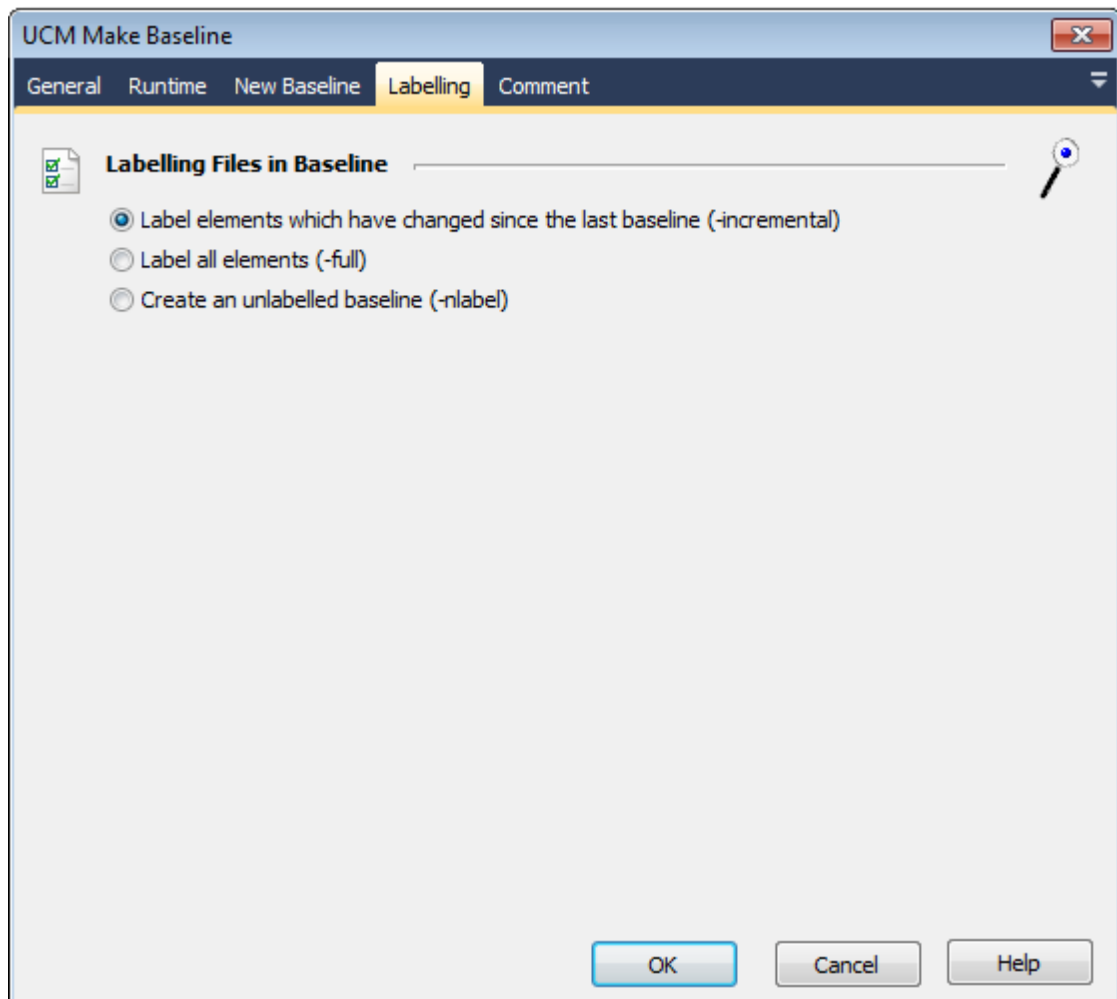
If a baseline name template is specified in the config spec, then the new baseline can be named from it.

Components to make baselines for

By default, a baseline contains all components that have been modified since the last baseline. However, baselines can be made to depend on components or activities. Baselines can also be made by importing a label type.

Both of the lists (components and activities) are updated with all of the components & activities for the current view. Check any which you wish to select.

If you cannot see the component or activity that you wish to select, change the View Context to the view that contains that component or activity.

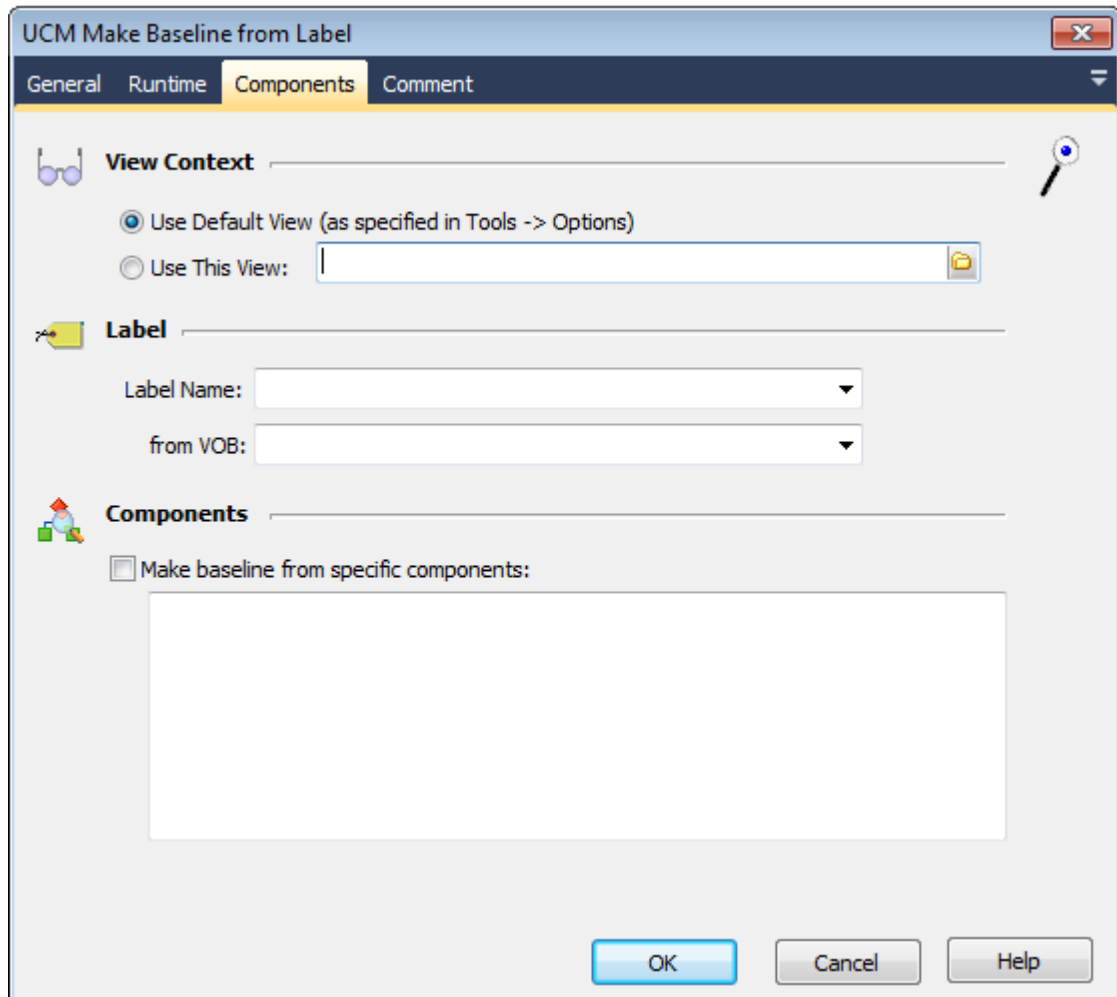


Elements in the baseline can be labelled with the name of the new baseline.

6.34.3.4.6 UCM Make Baseline from Label Action

Use this action to create a new baseline in a UCM stream. The baseline will correspond to a set of labelled versions in the VOB.

For full details of the mkbl command, see the ClearCase manual page (type *cleartool man mkbl.*)



The baseline can be created in either the default view (specified on the Options panel) or a view specific to that action.

Label

Choose a label name. All versions with that label name will be added to the new baseline.

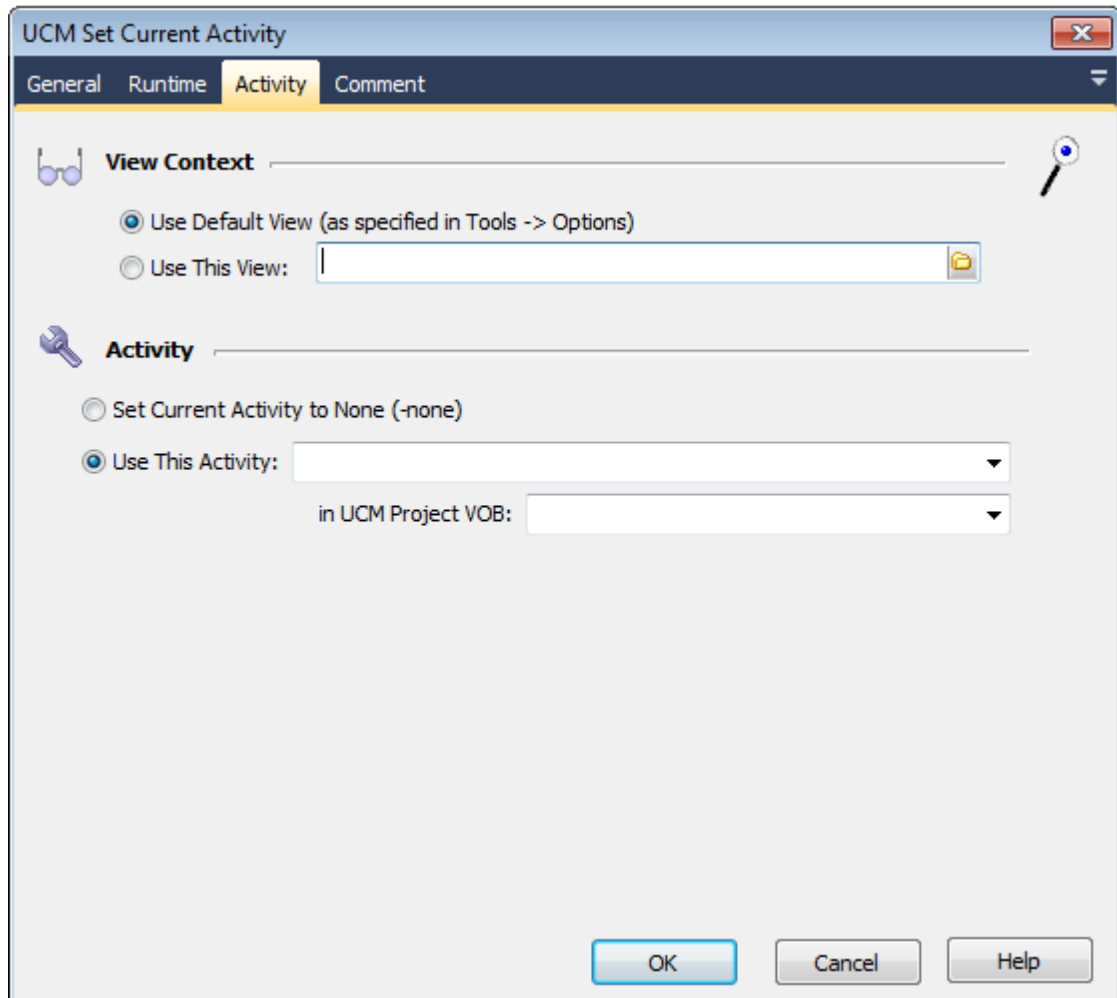
Components

If a list of components is specified, the new baseline will only contain labelled elements from those components.

6.34.3.4.7 UCM Set Current Activity Action

Use this action to set the current activity for a UCM stream.

For full details of the setactivity command, see the ClearCase manual page (type *cleartool man setactivity*.)



The stream to use can be taken from the default view (specified on the Options panel) or a view specific to that action.

Activity

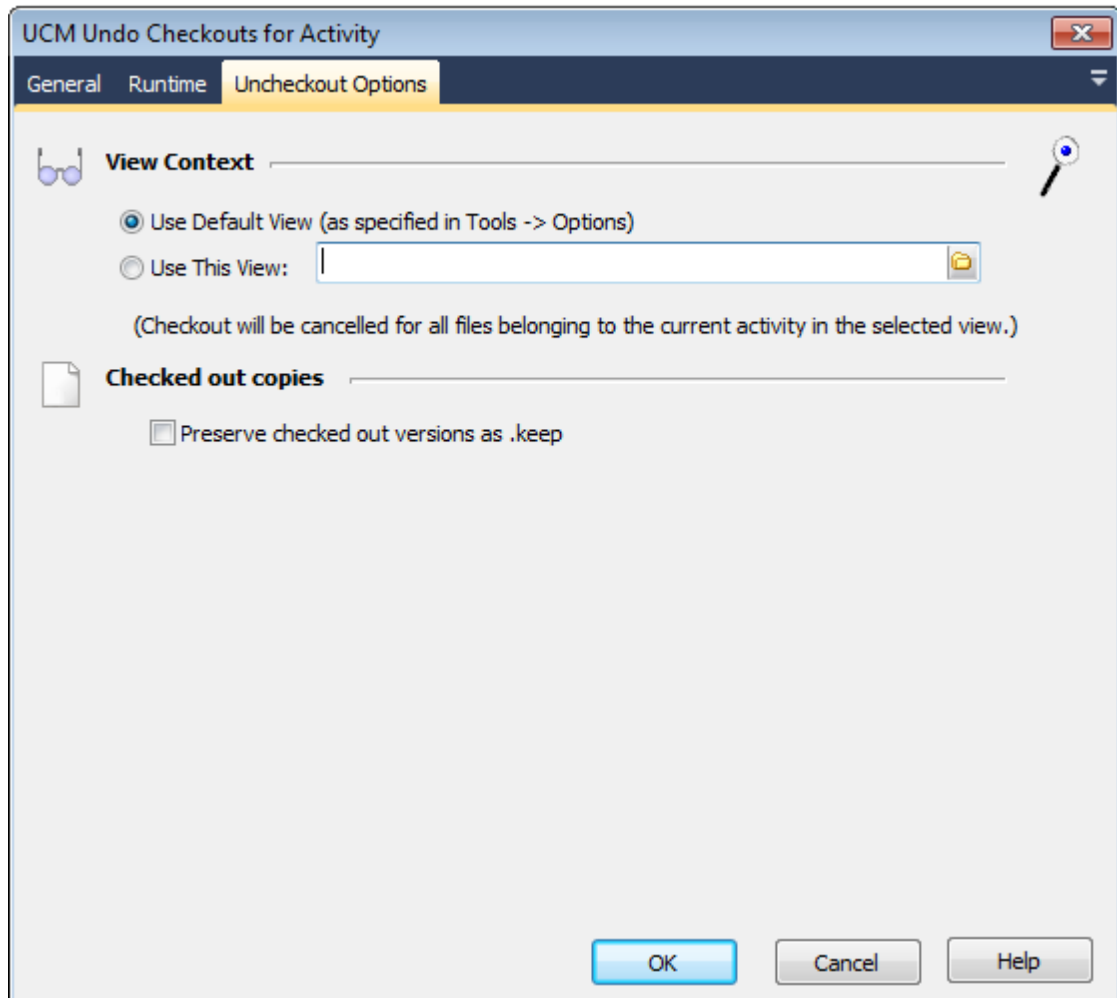
The current activity can be set to 'none', or to any activity from the Stream's UCM project VOB. Click on the VOB dropdown menu to see a list of VOBs, and the Activity dropdown menu to see a list of activities in that VOB.

6.34.3.4.8 UCM Undo Checkouts for Activity Action

Use this action to undo all checkouts registered to the current UCM activity for a view.

For full details of the uncheckout command, see the ClearCase manual page (type

cleartool man uncheckout.)



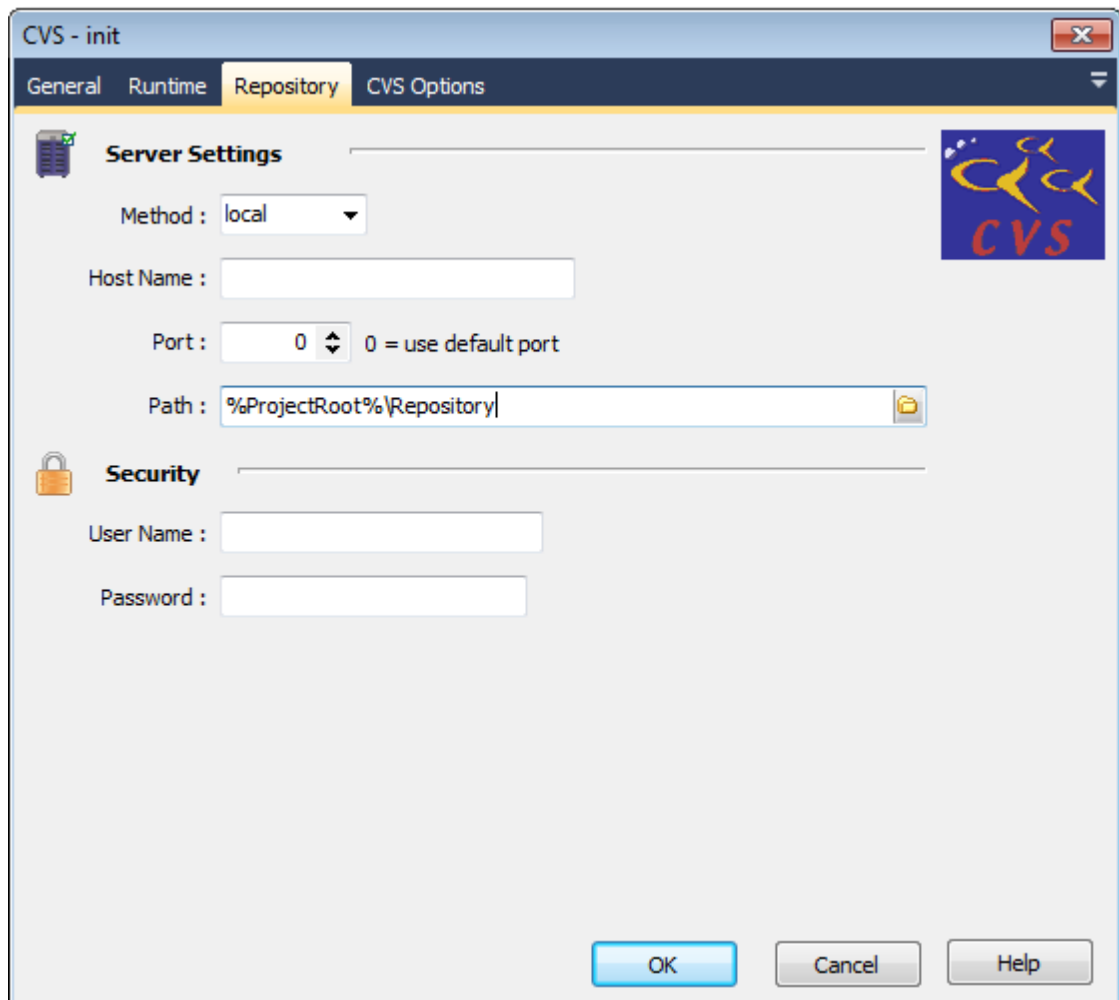
The stream to use can be taken from the default view (specified on the Options panel) or a view specific to that action.

6.34.4 CVS Actions

The CVS Actions provide a basic interface over the CVS command line tool, cvs.exe. Not all options are exposed in these action's property pages, however they do allow you to specify additional options to be passed to the command line.

All of the CVS actions have the Repository and Global Options property pages. These pages expose the options that control how you connect to CVS (either remotely or locally) and provide access to some properties that are common across most of the CVS actions.

For information on how to use CVS, see the CVS documentation available here : <http://www.cvshome.org>



There is an example Build script that demonstrates the usage of some of the CVS actions in the FinalBuilder\Examples\CVS Directory installed with FinalBuilder.

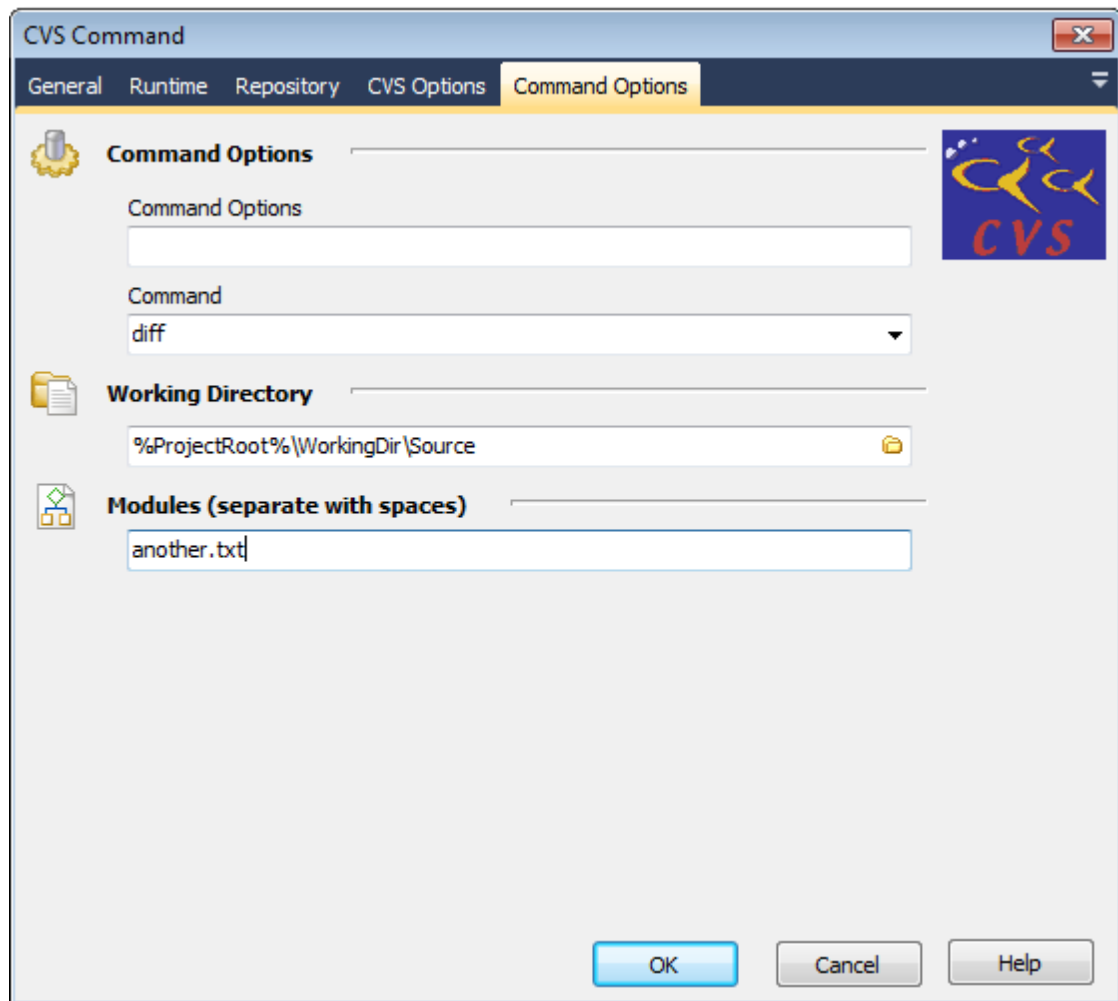
Note : CVS.exe requires that two environment variables HOMEDRIVE and HOMEPATH are set before running it. When running from the scheduler, make sure you run under a user that has these environment variables set, or add them as FinalBuilder project variables and check the "Make available as Environment variable" option when adding the variables. Typical values for these variables are :

HOMEDRIVE=E:

HOMEPATH=\Documents and Settings\Vincent

6.34.4.1 CVS Command Action

This action provides a generic CVS command interface. The list of commands in the Command drop down list is not the only commands that can be called, you can type the name of the command you wish to call into the combo box.



6.34.5 Git

Git actions provide interaction with Git Version Control System (msysgit).

To learn more about Git you can visit the [Git Homepage](#).

Before running Git actions you will need to provide the location of the Git installation directory via the Git Options page.

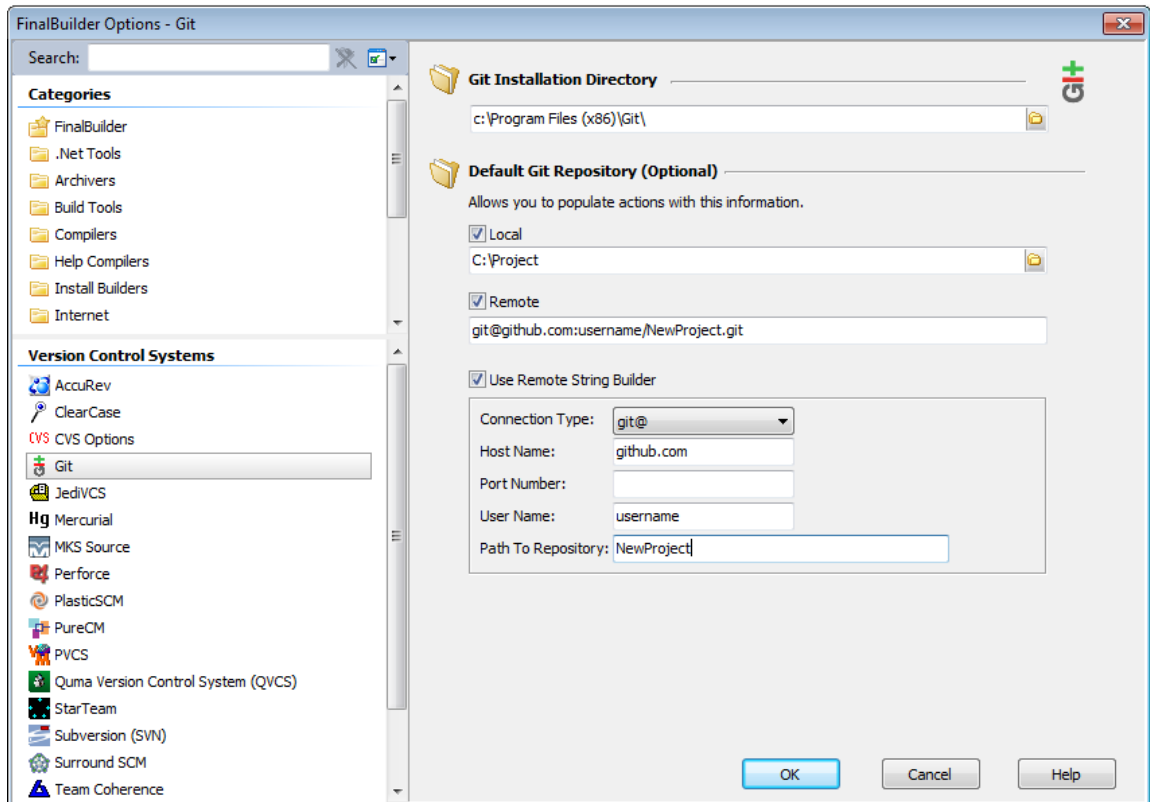
The actions available are:

- Git Add Files To Repository
- Git Checkout
- Git Clone Repository
- Git Commit
- Git Create Branch
- Git Create Repository
- Git Delete Branch
- Git Fetch
- Git Generic
- Git List Branches
- Git Merge
- Git Move
- Git Pull
- Git Push
- Git Remove Files
- Git Rename
- Git Repository Status
- Git Reset
- Git Show Commit Logs

6.34.5.1 Git Options

The Git Options page is available through Tools -> Options -> Version Control Systems -> Git. This is where you provide the location of the Git installation directory, which is required before running any Git actions.

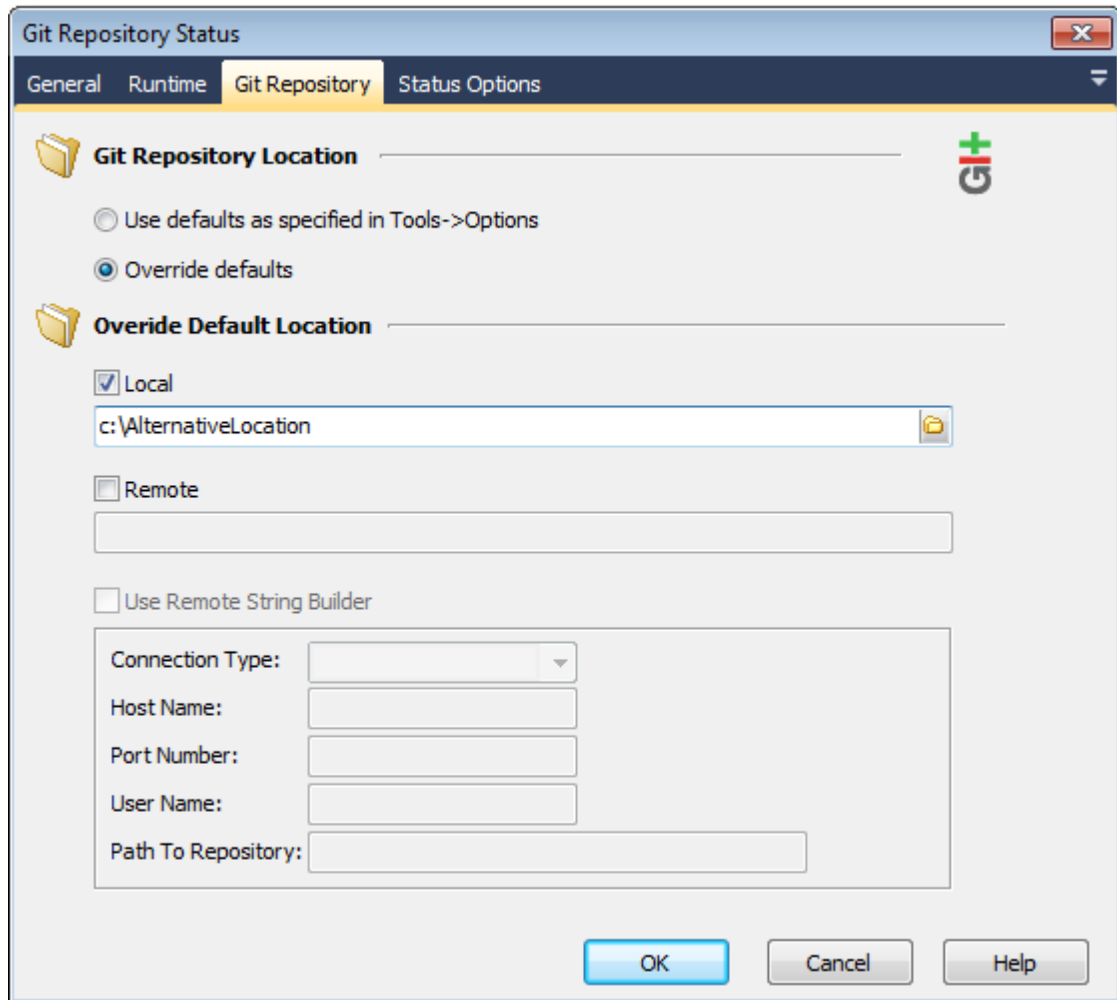
.



This page also provides the option to specify a default local and/or remote repository. Defining default repositories saves you from having to specify the repository values each time you want to run an action.

Once defined, you can add actions with the choice of using the defaults or overriding. Overriding allows you to specify an alternative values to defaults.

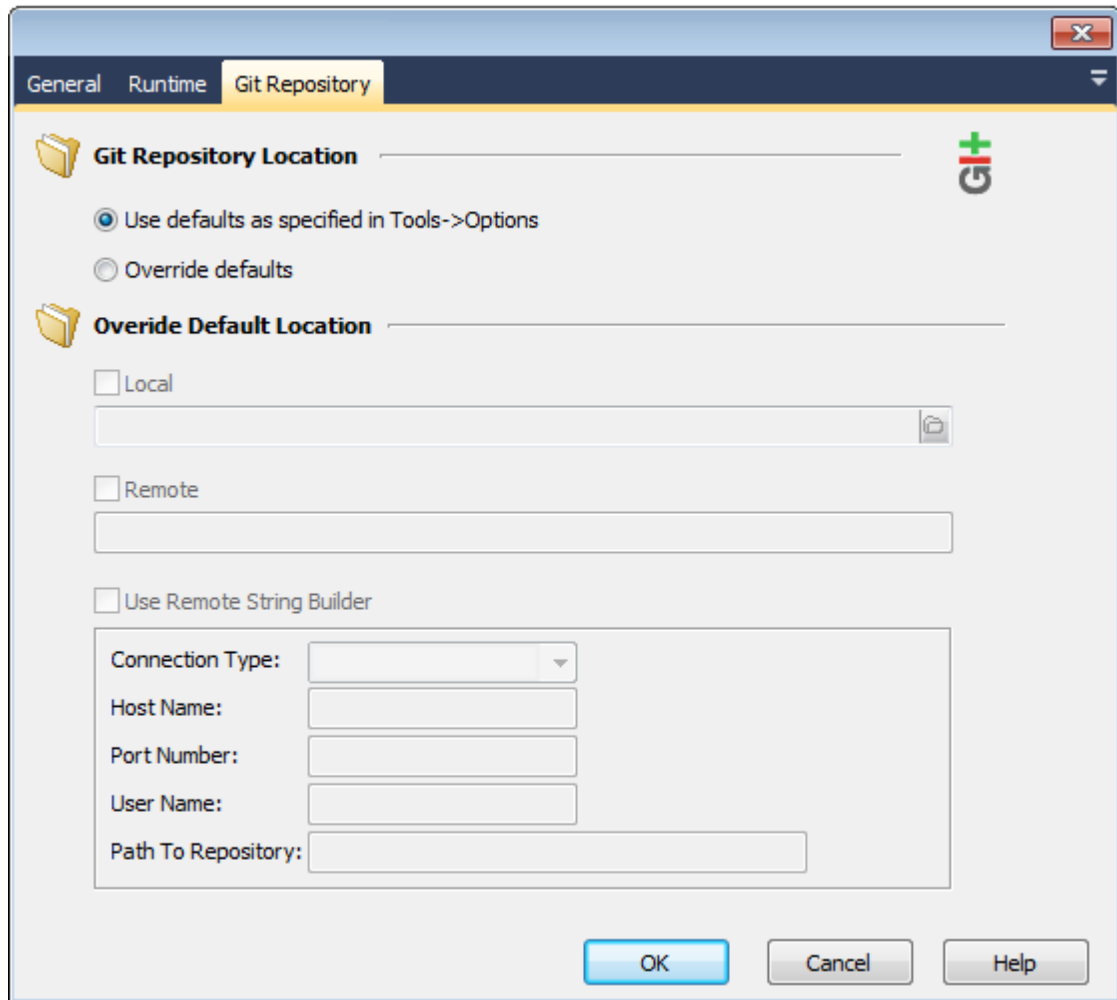
For example, if you have defined the defaults (as in the image above) but for one particular action you want to fetch from your default remote location to a repository other than your default local, you can simply override the local repository from the *Git Repository* tab (which is available for each action except the Create New Repository action) and the action will still use the default remote value as it has not been overridden.



6.34.5.2 Git Add Files To Repository

The Git Add Files action allows you to add the file contents of new or modified files to the index for inclusion of the next commit.

From the *Git Repository* tab select the local repository that you want to add the files to. If you have already specified a default local repository via Tools > Options > Git, then you can enable the *Use defaults* option to use this repository. Otherwise, you can override the default local repository by enabling the *Override defaults* option and specifying an alternative in the field provided.



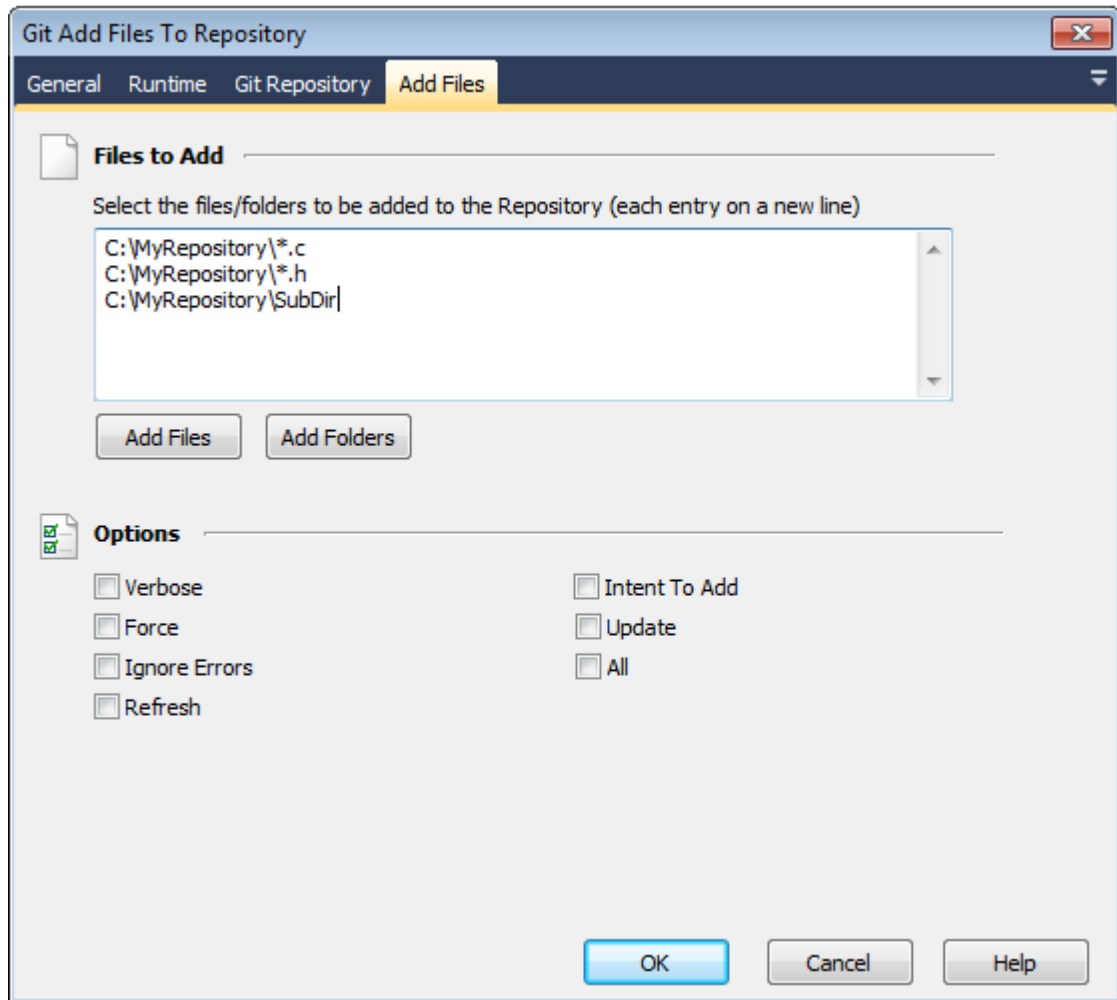
Once the local repository has been selected, switch to the *Add Files* tab and select the files and directories that you want to add. Ensure each entry is placed on a new line. The asterisk can also be used as a wildcard character, which saves having to manually add every file that you want to add.

For example:

*To add every file within the repository directory you can use *.**

*To add every file of a particular extension (in this example .txt) *.txt*

*To add files where names contain certain text (in this example "_v2") *_v2.**



There are also a number of options that can be executed with this action.

Verbose - Verbose output from Git.

Force - Add files that are otherwise ignored.

Ignore Errors - If an error occurs while indexing a file, ignore the error and continue to add the remainder of the files rather than ceasing on error. This will still result in the action failing.

Refresh - Only refresh the stat information in the index rather than adding the file.

Intent To Add - Add the path to the index without actually adding file content.

Update - This option allows you to only match against files listed in the index. This option also allows you to leave the file list blank in which case all tracked files will be updated in the index. Hence including a file in the file list which is currently untracked will have no effect as it will be ignored.

All - This option is similar to the *Update* except that it matches against both the working tree

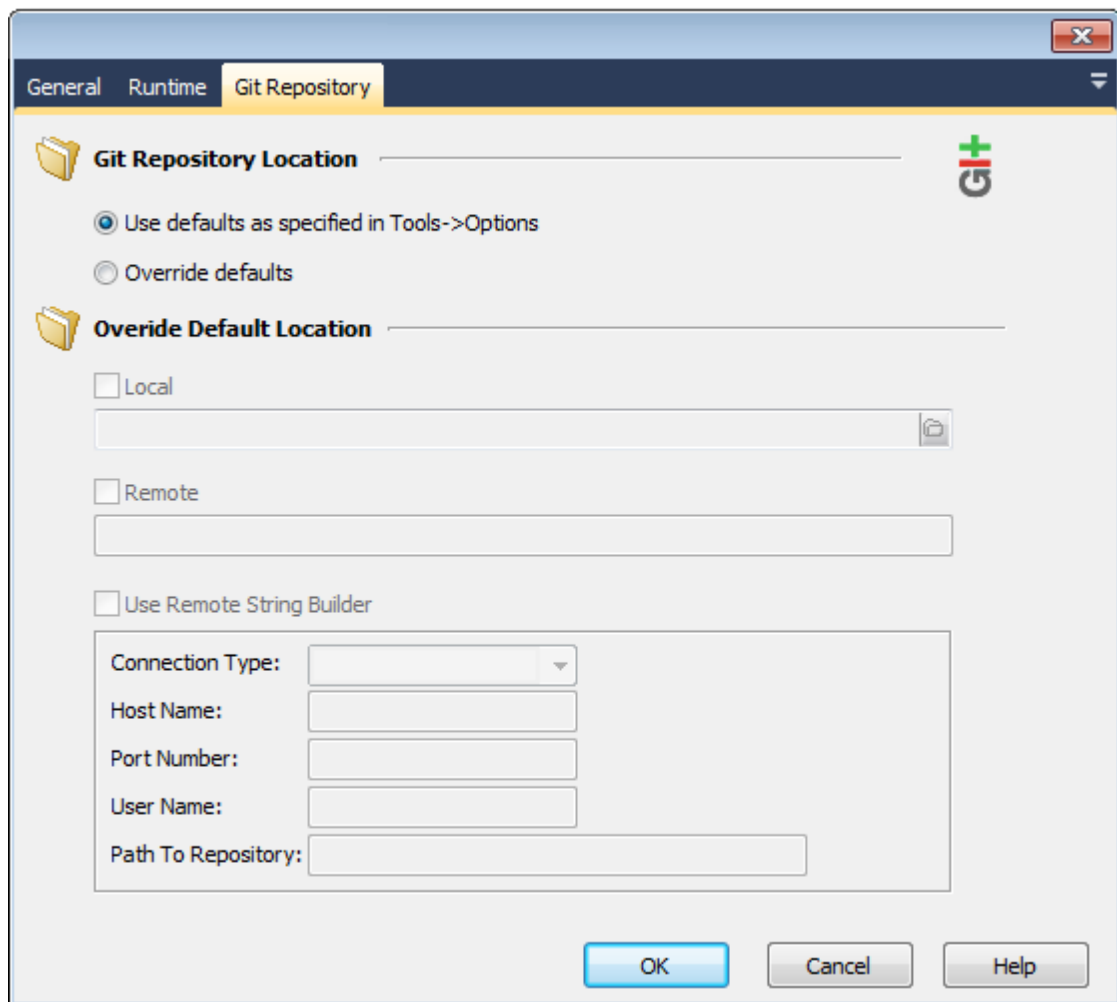
and the index. This option also allows you to leave the file list blank in which case all new files will be staged, modified files will be updated and files no longer in the working tree will be removed.

For more information on adding files to git, see: `git add`

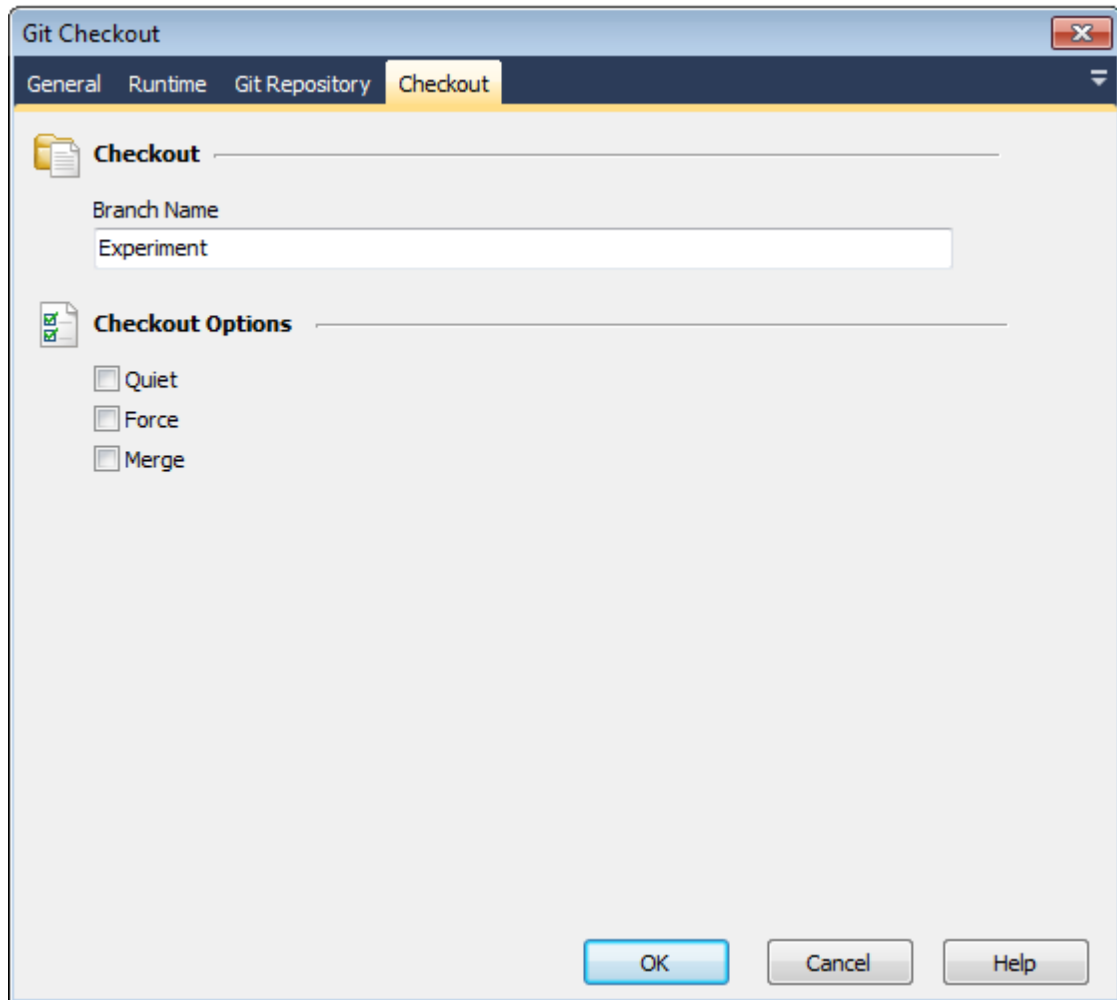
6.34.5.3 Git Checkout

The Git Checkout action allows you to checkout and switch to a different development branch.

From the *Git Repository* tab select the local repository where you want to checkout and switch branches. If you have already specified a default local repository via Tools > Options > Git, then you can enable the *Use defaults* option to use this repository. Otherwise, you can override the default local repository by enabling the *Override defaults* option and specifying an alternative in the field provided.



Once the local repository has been selected, switch to the *Checkout* tab and enter the name of the branch that you wish to switch to within the selected repository.



There are two options that can be executed with this action.

Quiet - Suppress output from Git.

Force - Allow switching branches when the index or the working tree is different to the head, which will throw away local changes.

Merge - Allows to you perform a three-way merge between the current branch, working tree contents and the new branch in order to preserve modifications. This occurs when attempting to switch branches where the current branch has local modifications that differ from the branch that is being switched to.

For more information on performing a checkout with git, see: [git checkout](#)

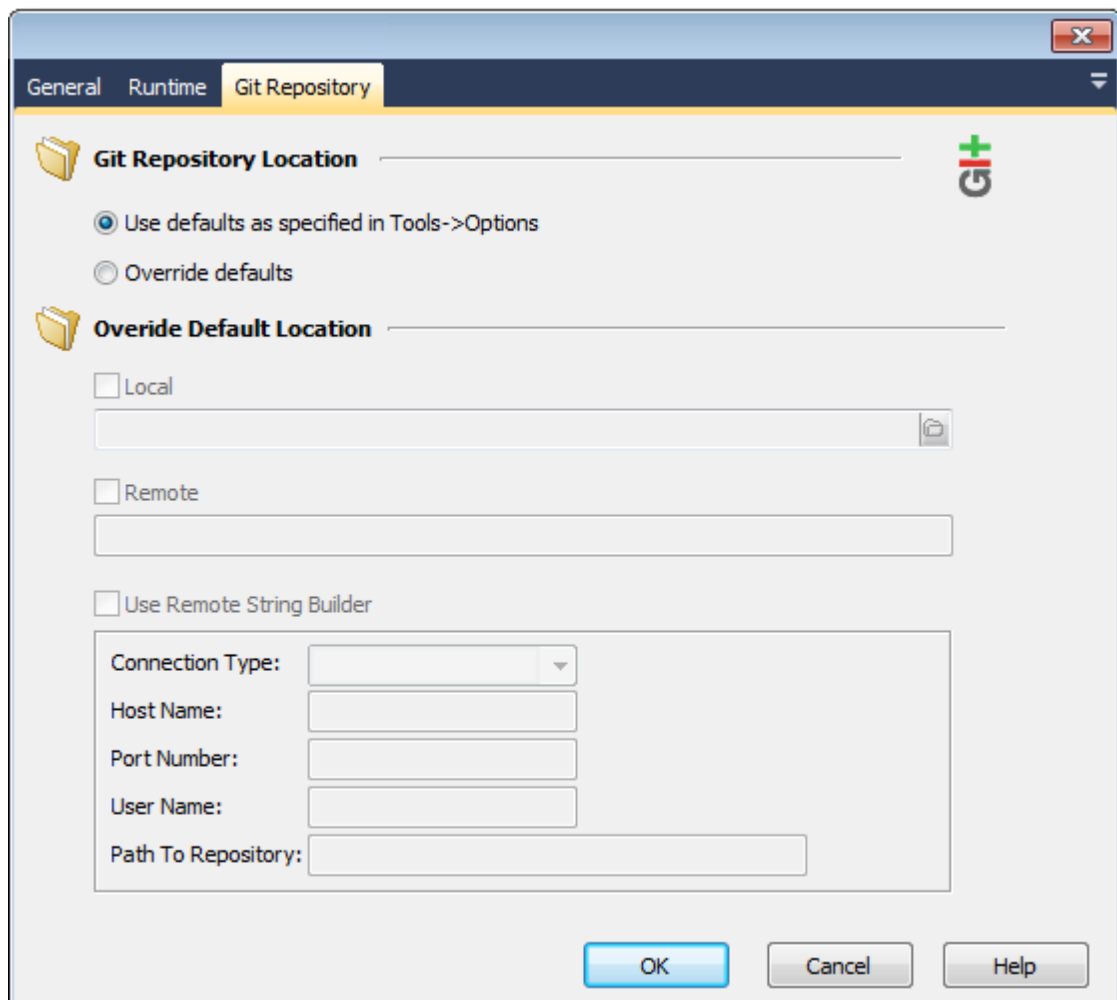
6.34.5.4 Git Clone Repository

The Git Clone Repository action allows you to clone an existing repository into a new directory.

From the *Git Repository* tab select the local repository where you want to clone the existing repository into. If you have already specified a default local repository via Tools > Options > Git, then you can enable the *Use defaults* option to use this repository. Otherwise, you can override the default local repository by enabling the *Override defaults* option and specifying an alternative in the field provided.

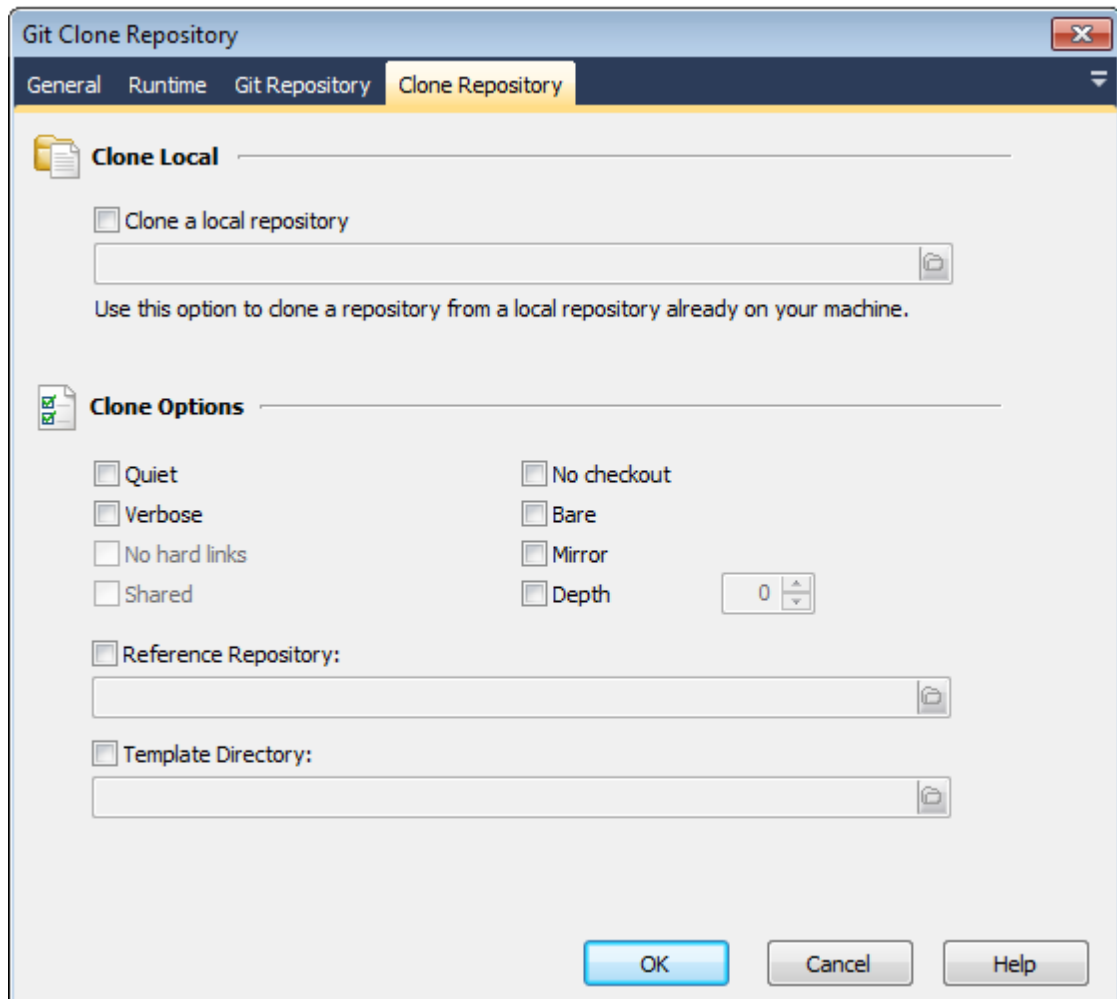
The destination directory cannot already contain a repository, otherwise the command will fail.

To clone a remote repository, you need to specify the location of the repository to clone from, either by using the default remote repository (if specified) or overriding the default value.



To clone a local repository switch to the *Clone Repository* tab and check the *Clone a local repository* option select the directory of the local repository to clone.

If both a local and a remote source repository are provided then the local repository will be cloned. Ensure that the *Clone a local repository* option is unchecked when attempting to clone a remote repository.



There are a number of options that can be executed with this action:

Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

No hard links - When cloning a local repository, optimise the process by copying the files under *.git/objects*. Only enabled when cloning a local repository.

Shared - When cloning from a local repository, don't setup hard links, instead setup *.git/objects/info/alternates* to share objects with the source repository. Only enabled when cloning a local repository.

No checkout - Do not checkout the HEAD once the clone has completed.

Bare - Rather than place the administrative files in *<Destination Directory>/.git* place all the files in the *<Destination Directory>*. (Implies No checkout option).

Mirror - Creates a mirror of the remote repository (implies Bare option).

Depth - Limit the resulting repository to the specified number of revisions to create a *shallow* repository. *Note: Shallow repositories have a number of limitations that users should be aware of before using this option.*

Reference Repository - This options allows you to copy *.git/objects/info/alternatives* from the repository located in the specified directory.

Template Directory - Specify the directory to copy the templates from (override the default of

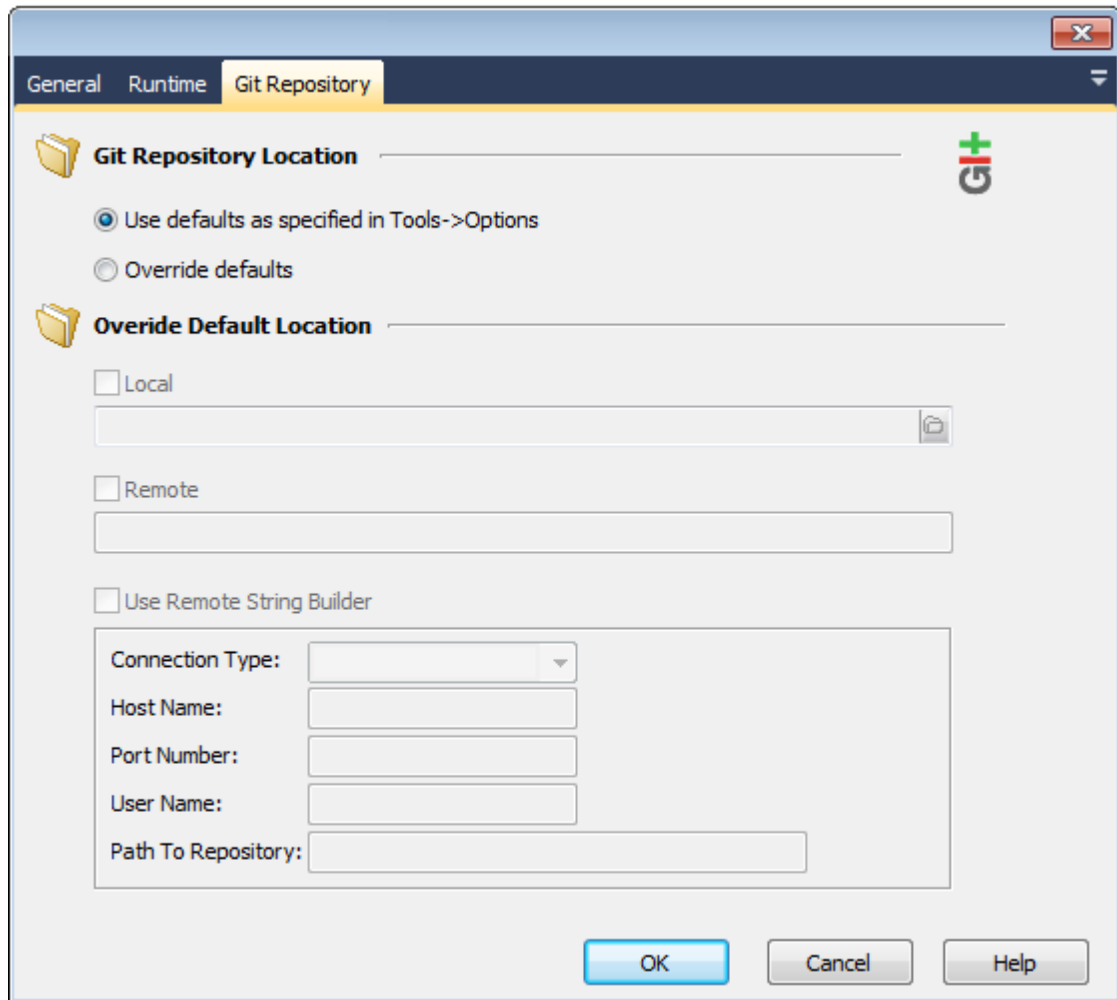
<Git Installation Directory>/share/git-core/templates/).

For more information on cloning a git repository, see: `git clone`

6.34.5.5 Git Commit

The Git Commit action allows you to record changes made to the Git Repository.

From the *Git Repository* tab select the local repository where you want to perform the commit. If you have already specified a default local repository via Tools > Options > Git, then you can enable the *Use defaults* option to use this repository. Otherwise, you can override the default local repository by enabling the *Override defaults* option and specifying an alternative in the field provided.



To perform a commit a message is required. Switch to the *Commit Message* tab to provide a message. There are three message options on this tab (you must select one):

- Commit Message: Simply type the commit message in the text field provided. The text entered here will be used as the commit message.
- Re-use commit message from this Commit ID: Provide a commit ID to re-use the commit message from a previous commit.
- Text from file as commit message: Select a file that contains the text that you want to use as your commit message.

The screenshot shows a 'Git Commit' dialog box with a dark blue title bar and a light gray body. The 'Commit Message' tab is selected, highlighted in yellow. The dialog contains a message box with a speech bubble icon, a text input field with a pre-filled commit message, and two optional checkboxes with their respective input fields. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Git Commit

General Runtime Git Repository **Commit Message** Commit Options

Commit Message

You must select one of these options.

☒ Commit Message:

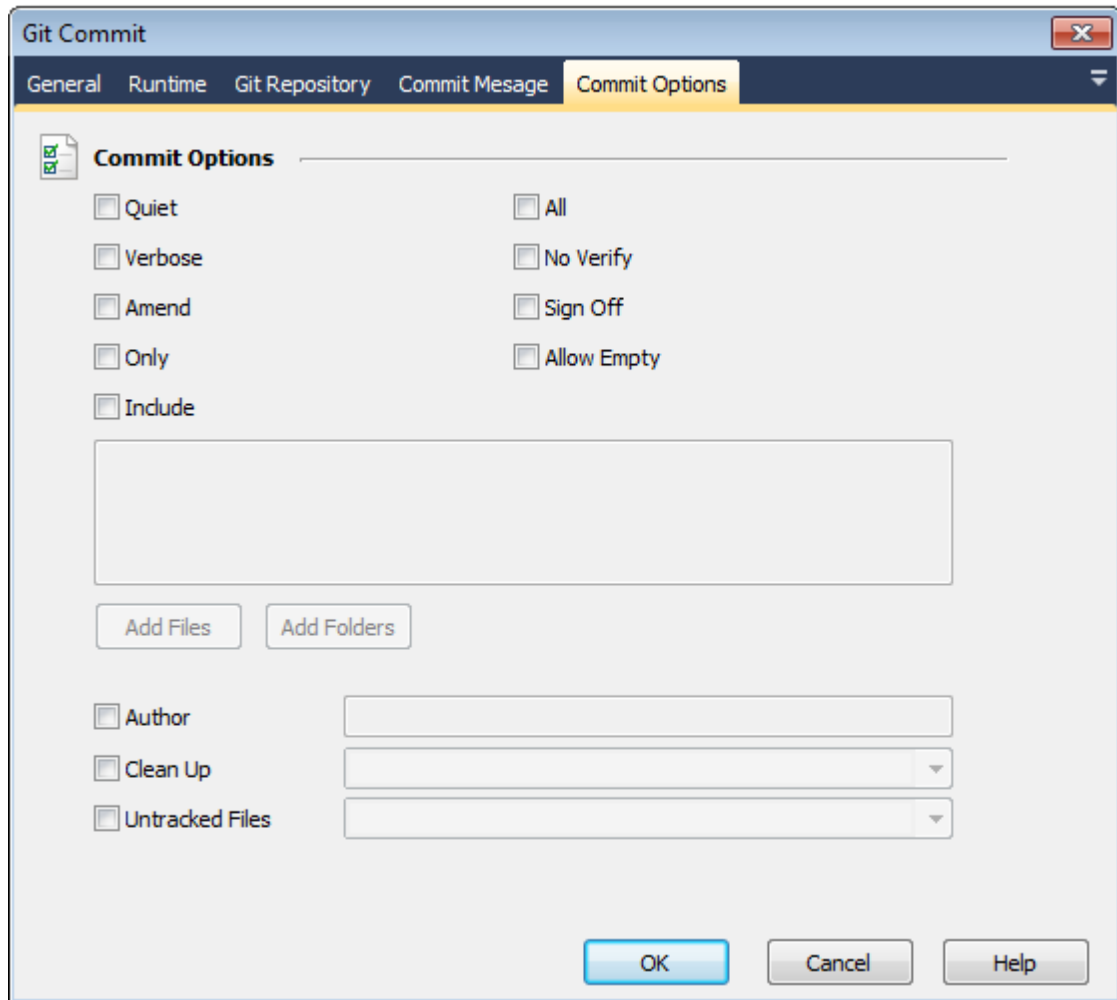
Add %Changes\$ to %repository%.

☐ Re-use commit message from this Commit ID:

☐ Text from file as Commit message:

OK Cancel Help

Switch to the *Commit Options* tab to see the list of available options that can be executed with this action.



Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

Amend - Amend the tip of the current branch.

Only - Commit only the files specified in the file list and ignore the content that is currently staged. If this option is used in conjunction with the *Amend* option then no paths need to be specified which can be used to amend the previous commit without committing changes that are currently staged.

Include - Before committing the contents that are already staged, stage the files that have been listed in the file list as well.

All - Automatically stage files that have changed and/or deleted but ignore new files that Git does not know about.

No Verify - Bypass the *pre-commit* and *commit-msg* hooks.

Sign Off - Add *Signed-off-by: <committer name>* at the end of the commit message.

Allow Empty - Allows you to override the safety mechanism that prevents you from making a commit where the tree is exactly the same as its sole parent commit.

Author - Override the author name used in the commit.

Clean Up - Determines how the commit message is cleaned up:

Verbatim - Do not change message at all.

Whitespace - Removes leading and trailing white space lines.

Strip - Removes both white space and commentary.

Default - Same as Strip option except only occurs if the message is to be edited.

Untracked Files - Specifies the handling of untracked files:

No - Do not show untracked files.

Normal - Show untracked files and directories.

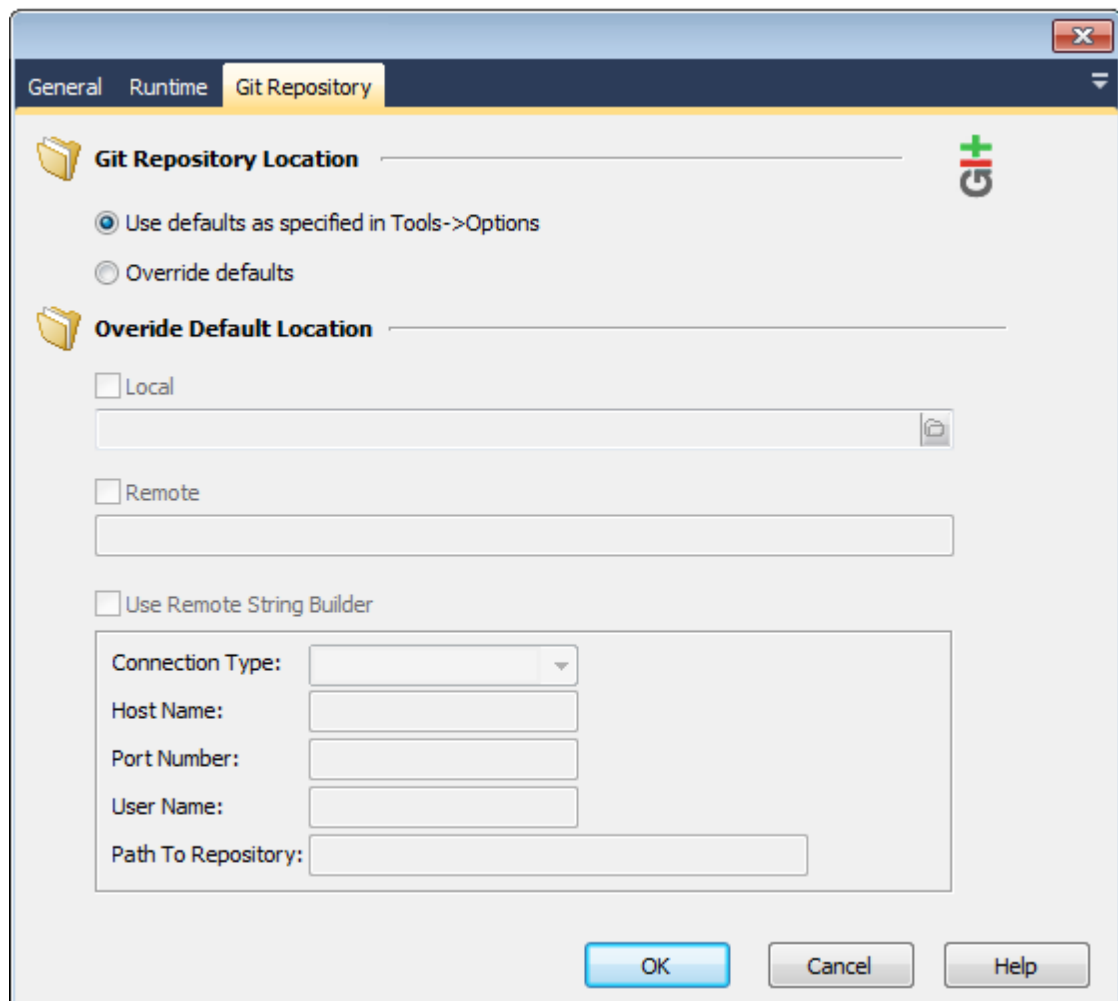
All - Show untracked files and individual files within untracked directories.

For more information on committing changes within git, see: `git commit`

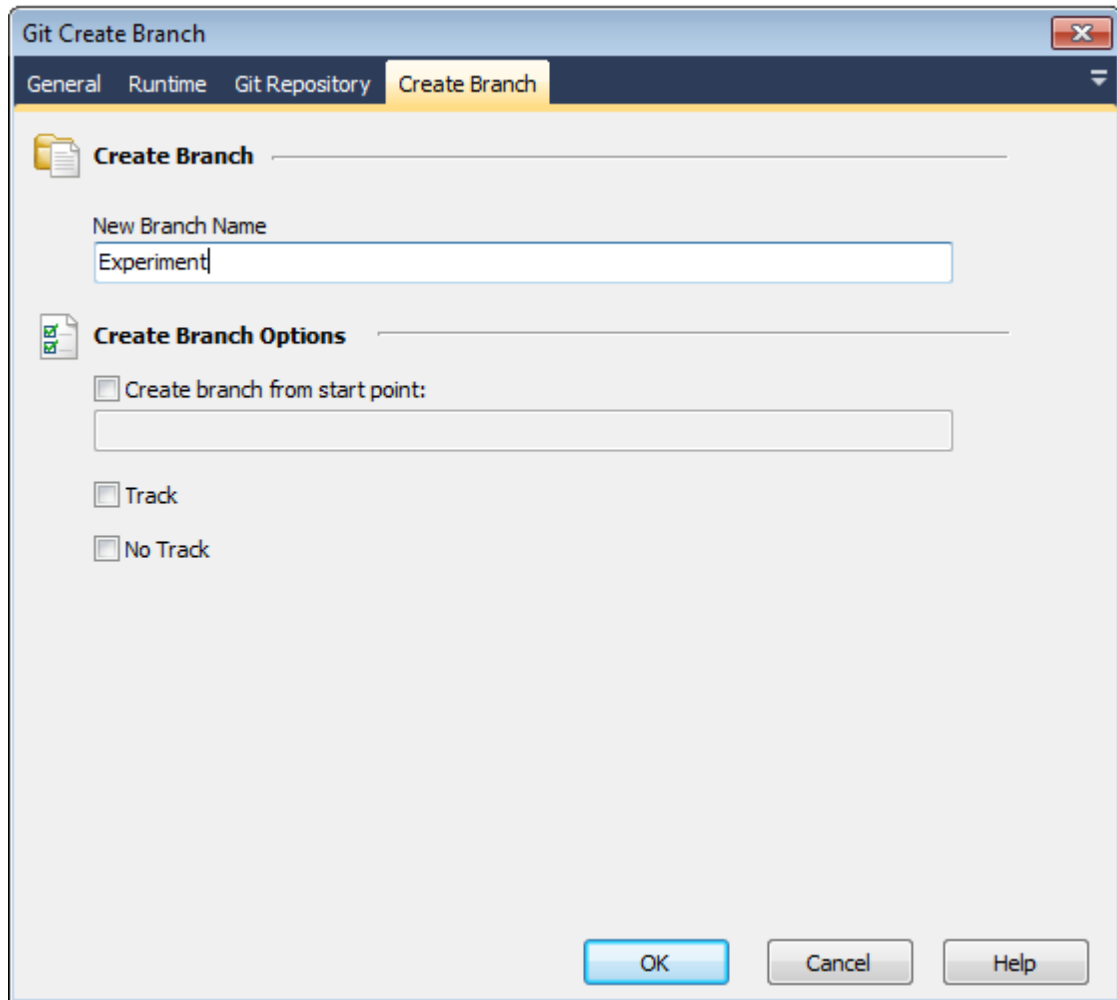
6.34.5.6 Git Create Branch

The Git Commit action allows you to create a new branch.

From the *Git Repository* tab select the local repository where you want to create a branch. If you have already specified a default local repository via Tools > Options > Git, then you can enable the *Use defaults* option to use this repository. Otherwise, you can override the default local repository by enabling the *Override defaults* option and specifying an alternative in the field provided.



On the *Create Branch* tab specify the name for the new branch. The name must comply with the git-check-ref-format rules.



The *Create Branch* tab contains a number of options that can be executed with this action:

Create Branch from start point - Set the new branch to point to a commit by providing a commit id or a tag.

Track - Setup the configuration to mark the start-point for the branch as *upstream* from the new branch.

No Track - Do not setup *upstream* configuration, even if the *branch.autosetupmerge* configuration variable is true.

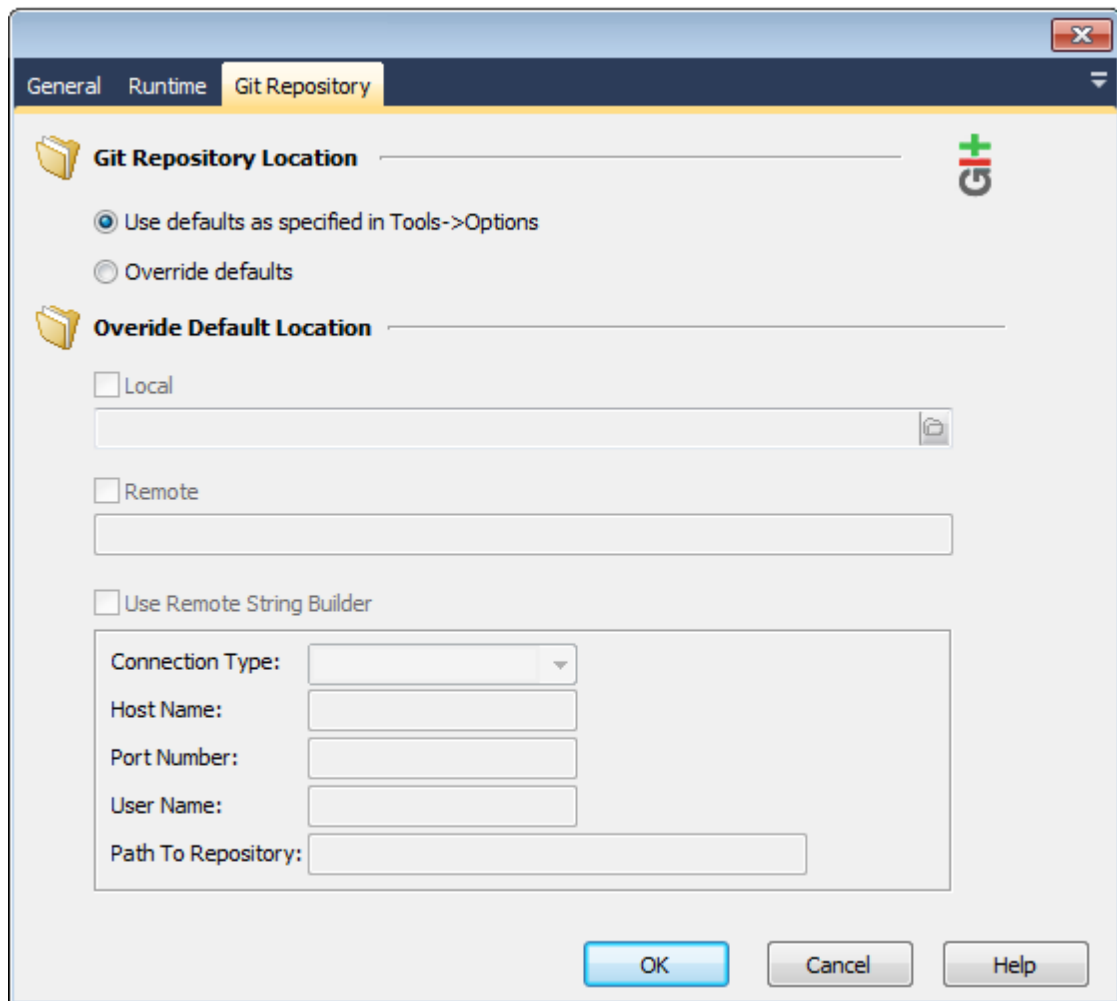
For more information on creating a branch in git, see: [git branch](#)

6.34.5.7 Git Create Repository

The Git Create Repository action allows you to initialize a new local repository on your machine.

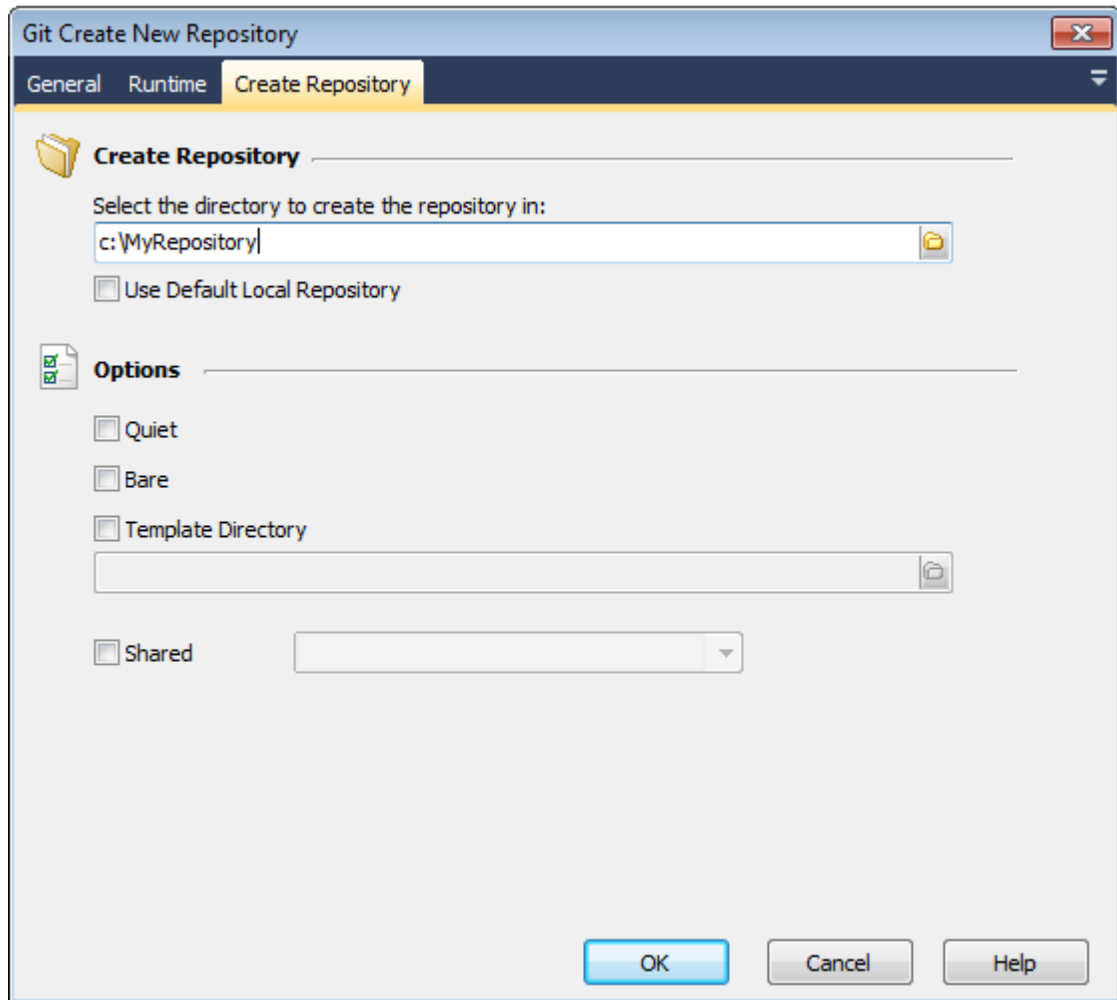
From the *Create Repository* tab select the directory where you want to initialize the new repository. The *Use Default Local Repository* option can be selected to initialize a repository in

the directory specified in Tools > Options > Git as the default local repository.



You need to select a directory that exists, otherwise the action will fail. If you run this action against a directory that already contains a repository, no information will be lost or re-written. It can be useful to do this in order to pick up recently added templates.

There are also a number options that can be executed from the *Create Repository* tab.



Quiet - Suppress output from Git.

Bare - Rather than place the administrative files in *<Destination Directory>/ .git* place all the files in the *<Destination Directory>*.

Template Directory - Specify the directory to copy the templates from (override the default of *<Git Installation Directory>/share/git-core/templates/*).

Shared - Specify that the repository will be shared by several users:

Umask/False - User permissions set as `umask(2)`. This is the default when the shared option is not used.

Group/True - Make the repository group writeable.

All/World/Everybody - Same as *Group/True* option except the repository is readable by all users.

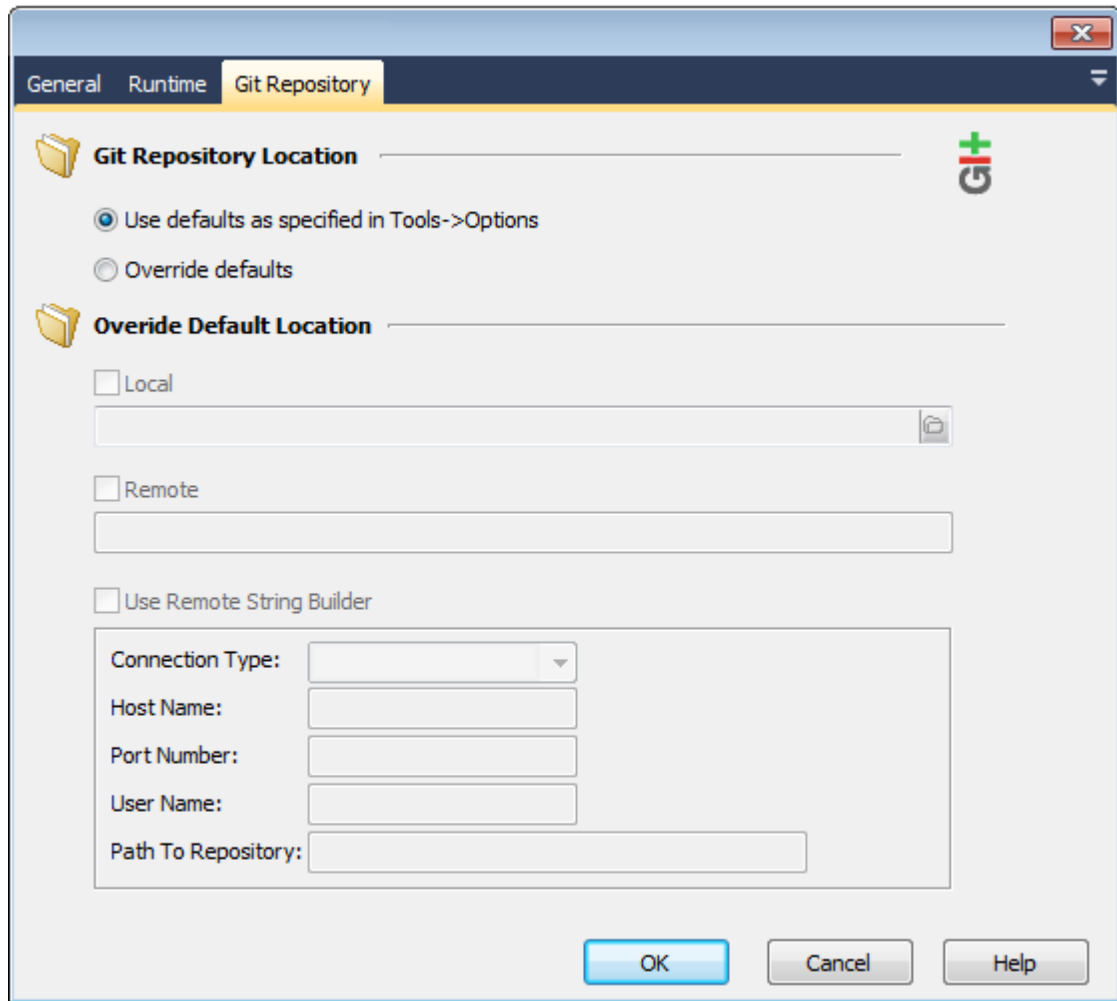
0xxx - This option allows you to specify the umask file mode permissions by specifying an octal number.

For more information on creating a git repository, see: `git init`

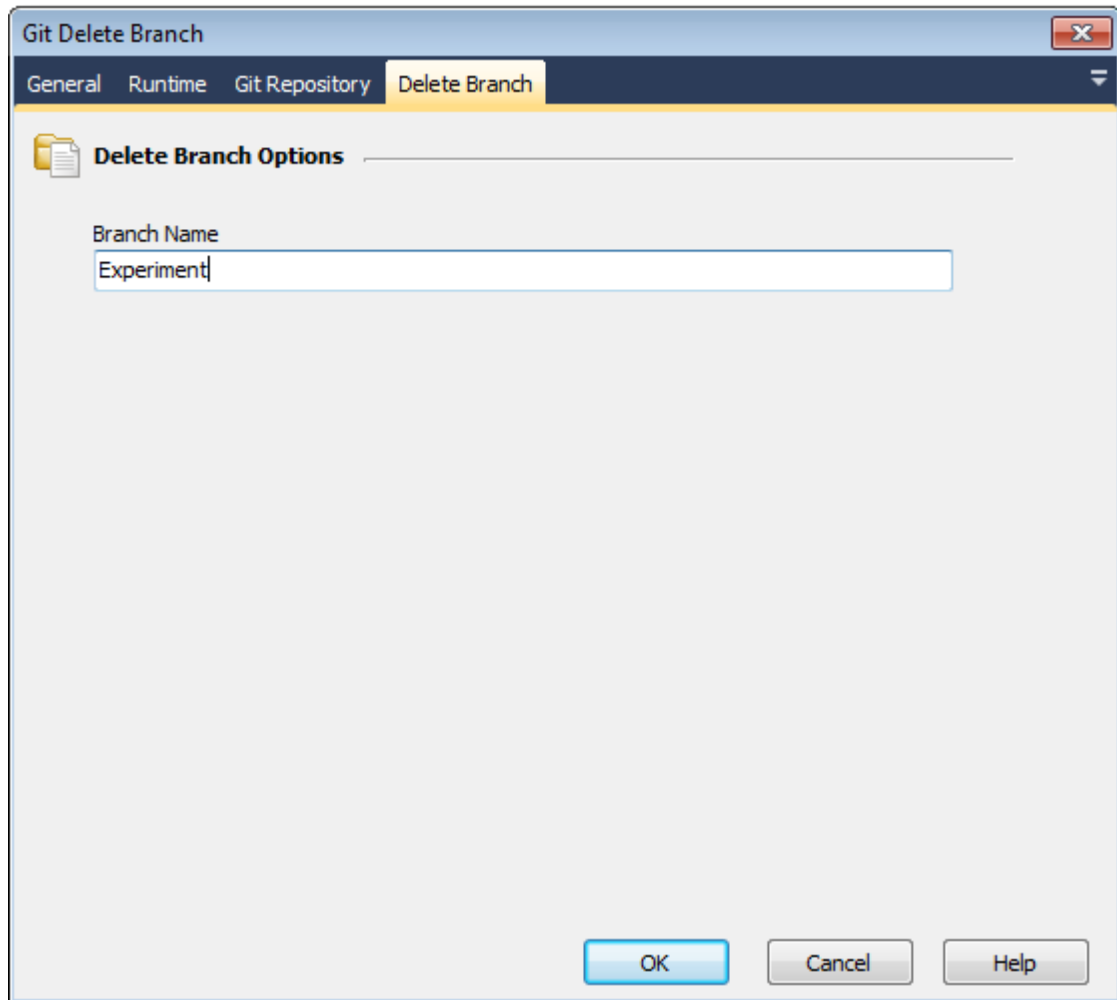
6.34.5.8 Git Delete Branch

The Git Delete Branch action allows you to delete a development branch from a local repository.

From the *Git Repository* tab select the directory that contains the repository from which you want to delete a branch.



Switch to the *Delete Branch* tab and enter the name of the branch to be deleted in the field provided.

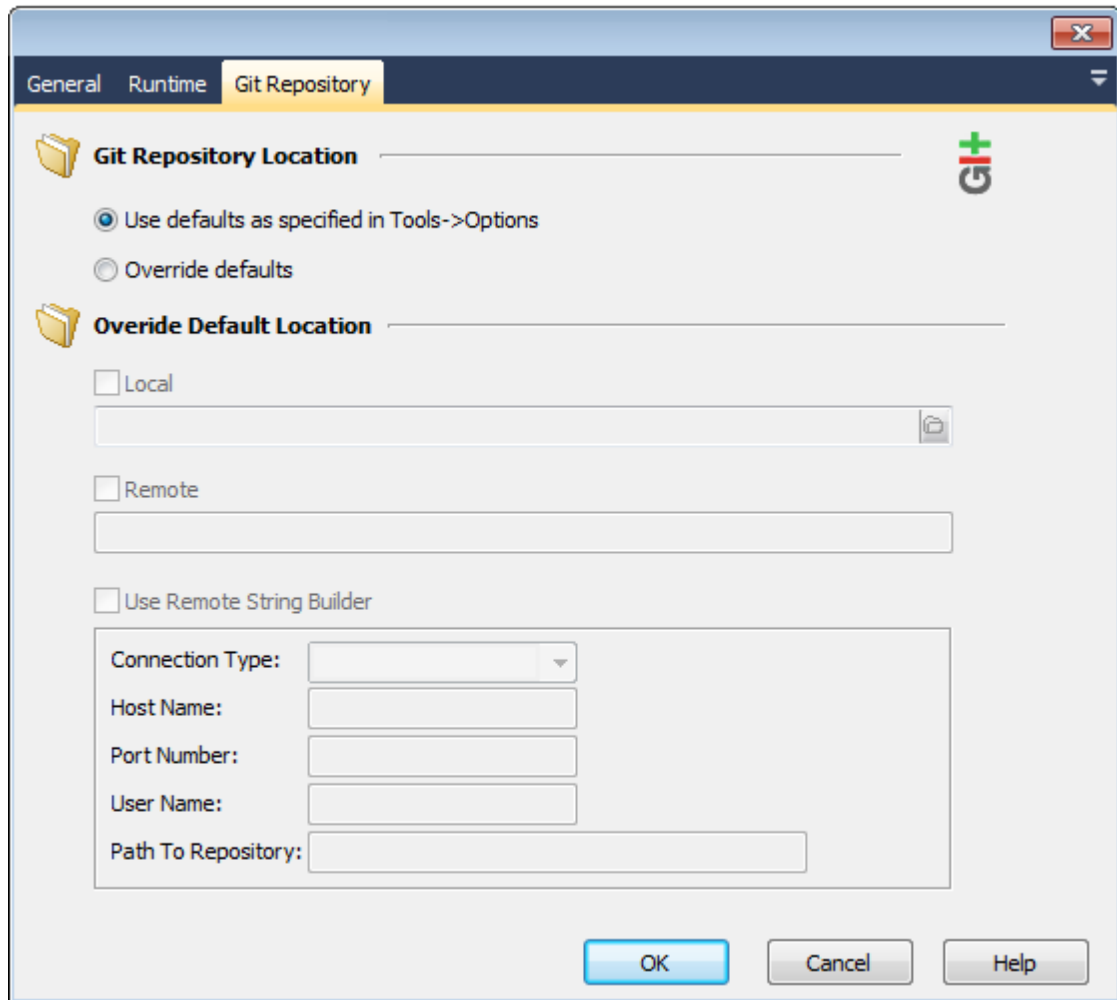


For more information on deleting a branch in git, see: [git branch](#)

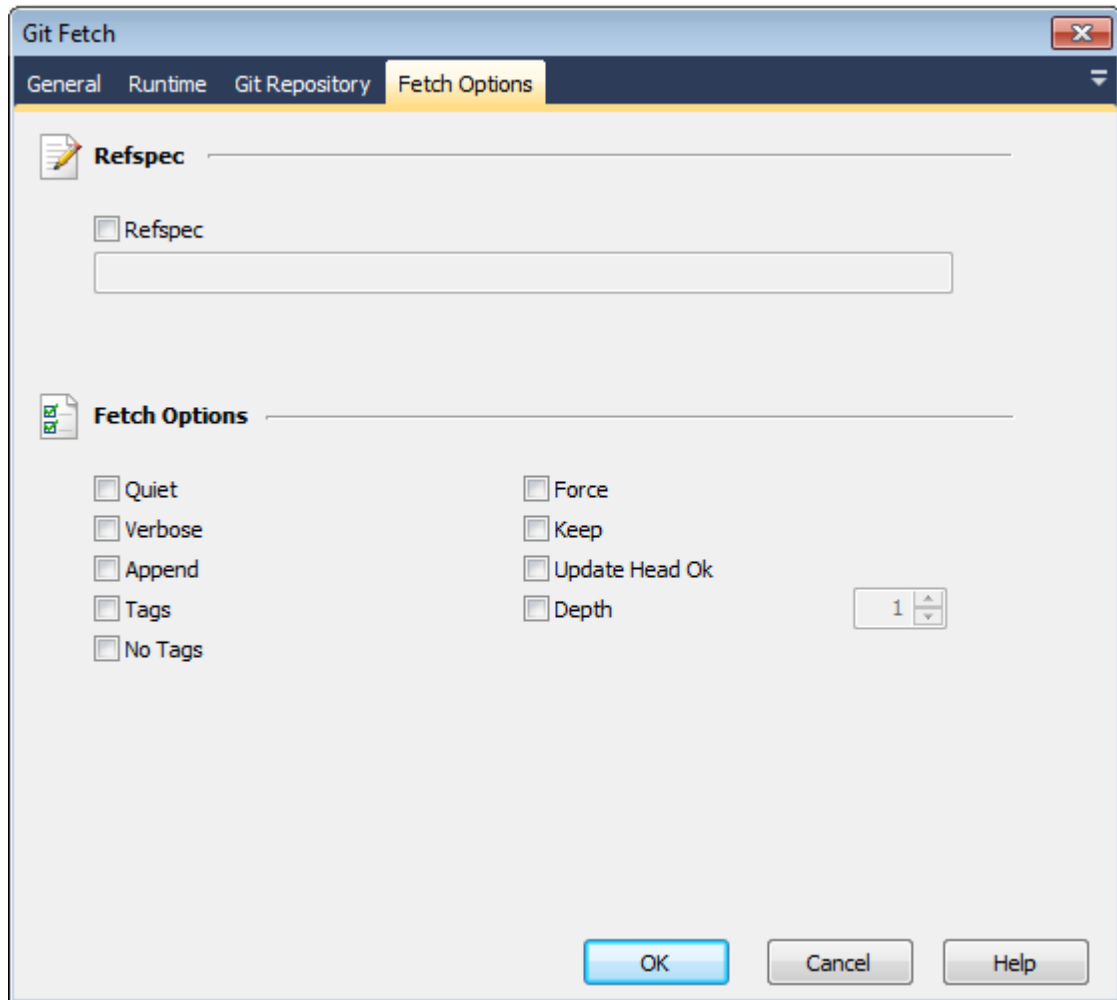
6.34.5.9 Git Fetch

The Git Fetch action allows you to download objects and refs from another repository.

From the *Git Repository* tab select the local repository location that you will be doing the fetch into and the remote repository that you will be doing the fetch from.



The Refspec option on the *Fetch Options* tab allows you to override the default behaviour of the fetch operation which will fetch all references under *refs/heads/* on the remote repository and write them to *refs/remotes/origin/* within the local repository. This option allows you to specify the source and destination references in the following way: *<source ref>:<destination ref>*. For more information about using the refspec option when fetching in git see *git fetch*.



The *Fetch Options* tab to select the options that you want to execute:

Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

Append - Append ref names and object names of the fetched material to the existing contents of `.git/FETCH_HEAD`. If this option is omitted the old data in this location will be overwritten.

Tags - This option specifies that all tags (and their associated objects) should be downloaded, even those that are not reachable from the branch heads that are being tracked (which are omitted by default).

No Tags - Tags that point to objects are downloaded from the remote repository by default. This option disables this from occurring.

Force - When using the refspec option to specify `<remote branch>:<local branch>` the fetch operation will not allow you update the local branch if the remote branch is not a descendant of the local branch. This option forces the fetch to occur regardless.

Keep - Keep downloaded pack.

Update Head Ok - The fetch command will not allow you to update the head which corresponds to the current branch by default. This option disables the check.

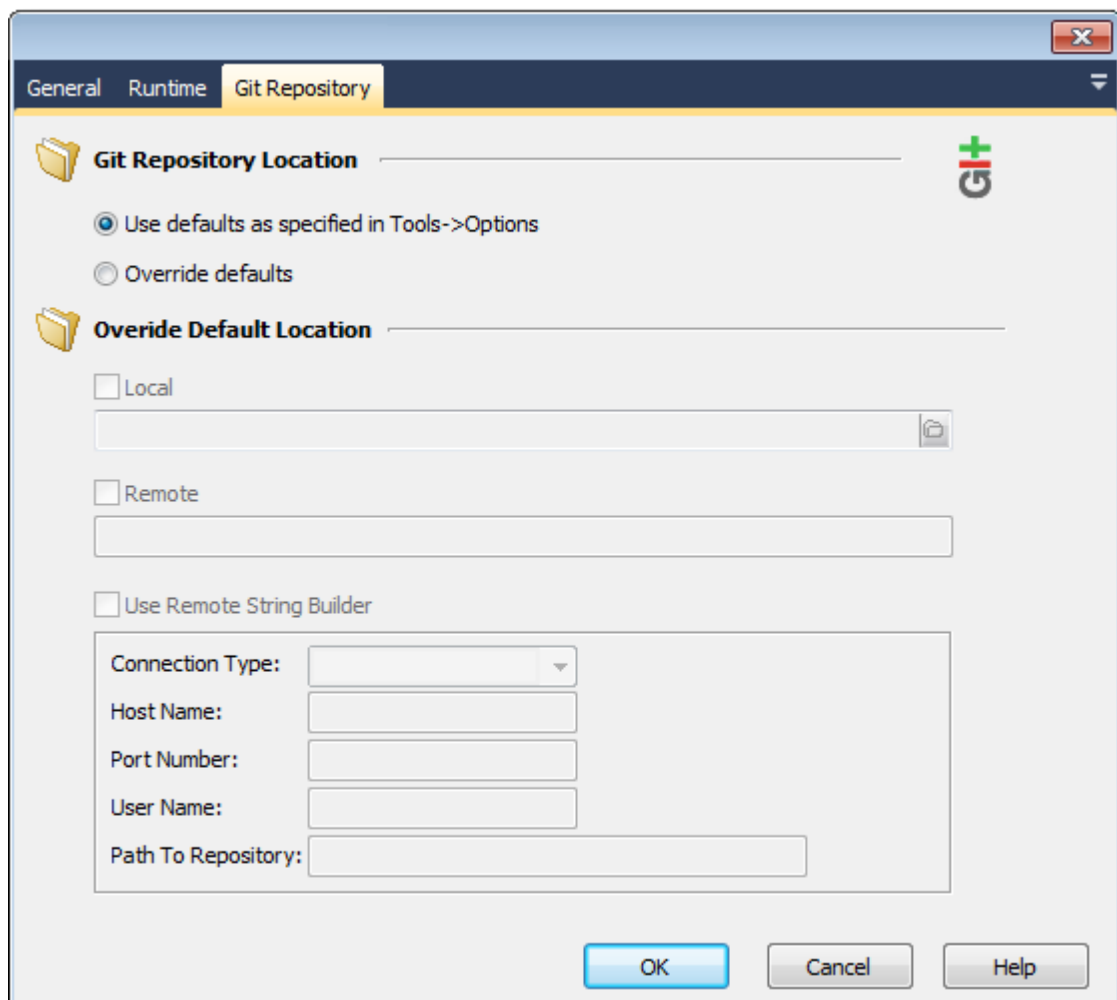
Depth - Increase the history depth of a *shallow* repository by the specified number of commits.

For more information on performing a fetch in git, see: [git fetch](#)

6.34.5.10 Git Generic

The Git Generic action allows you to run any Git command as you would via the command line.

From the *Git Repository* tab select the local repository location that you will be running the command against.

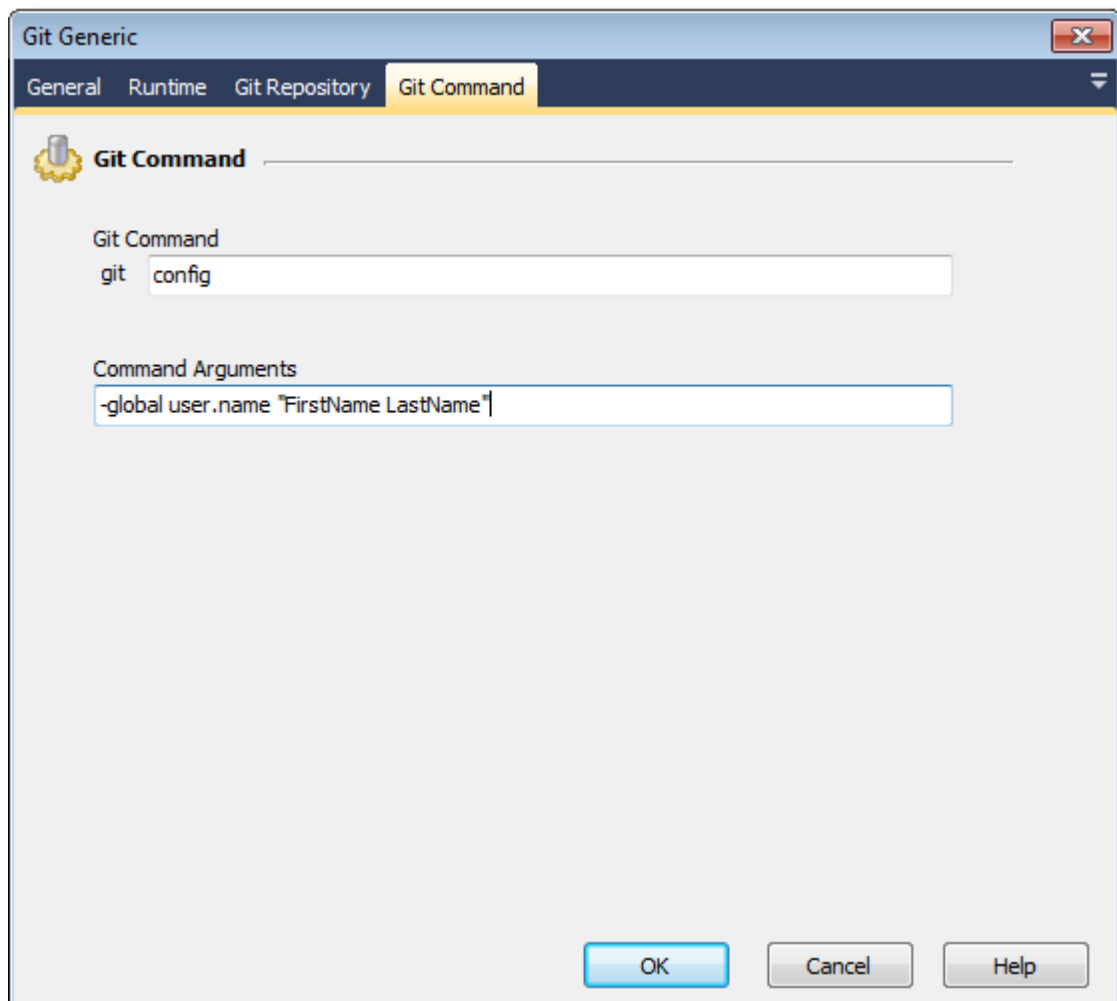


Switch to the *Git Command* tab to enter in a command.

In the *Git Command* field, enter a git command (do not type 'git' before the command as you

usually would from the command line).

Type any arguments into the *Command Arguments* field.



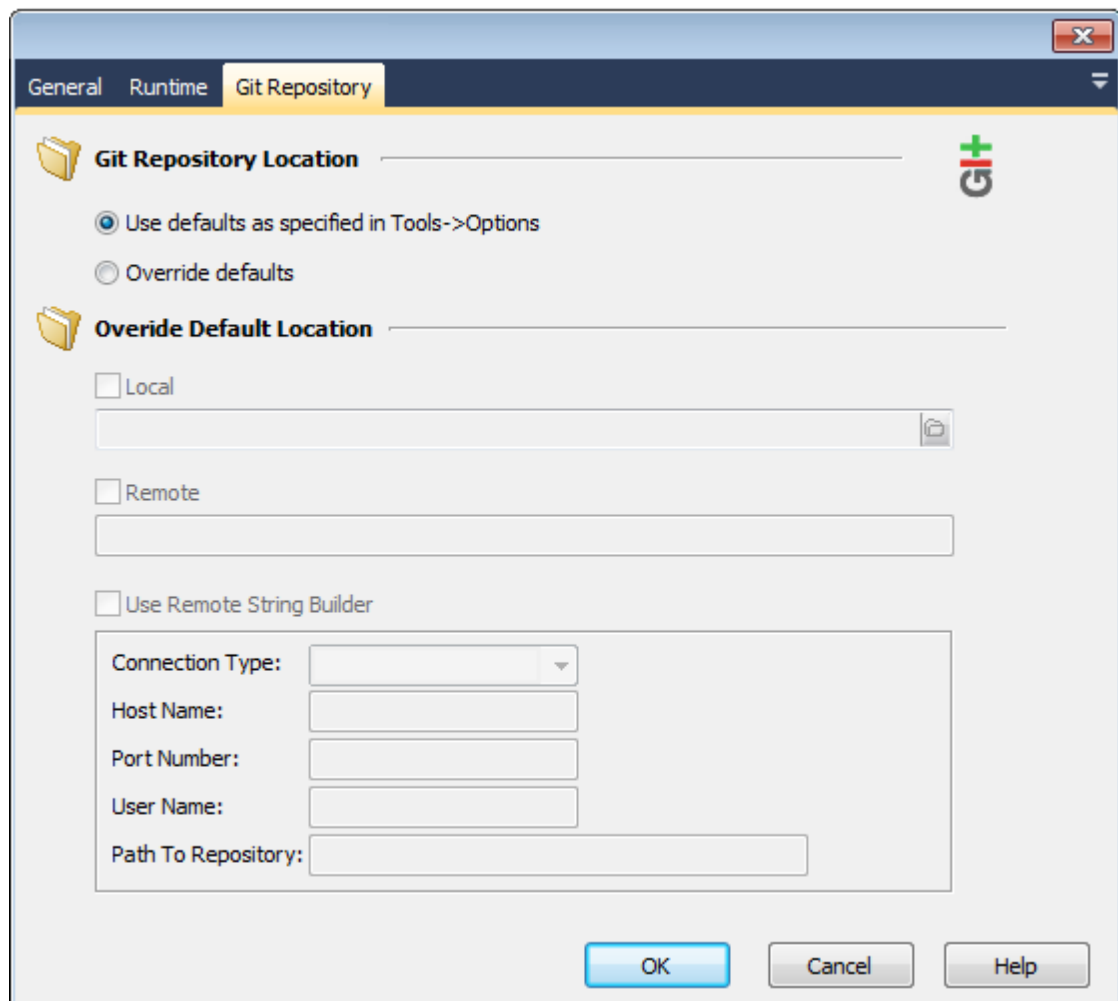
Note: Any syntax/command errors will result in a failed action.

For more information on git and the commands available, see: [git](#)

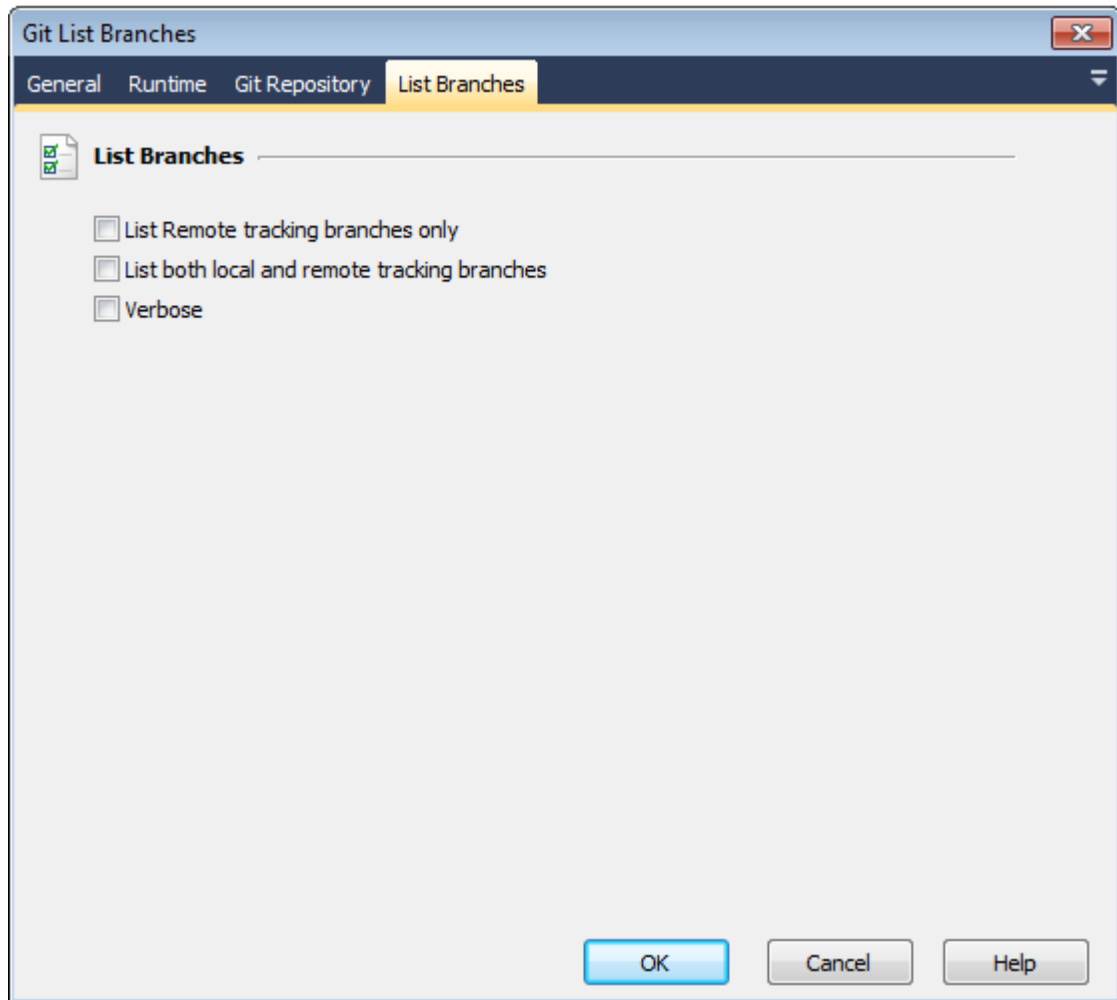
6.34.5.11 Git List Branches

The Git List Branches action allows you to list all the branches for a local repository.

From the *Git Repository* tab select the location of the local repository which you want to list the branches.



Switch to the *List Branches* tab to select any other options you would like to execute.



List Remote tracking branches only - self explanatory.

List both local and remote tracking branches - self explanatory.

Verbose - Verbose output from Git.

For more information on listing branches in git, see: [git branch](#)

6.34.5.12 Git Merge

The Git Merge action allows you to join two or more development histories together.

From the *Git Repository* tab select the location of the local repository where you want to merge the branches.

The screenshot shows the 'Git Repository' tab of a configuration window. The window has a title bar with a close button. Below the title bar are three tabs: 'General', 'Runtime', and 'Git Repository'. The 'Git Repository' tab is active. It contains a 'Git Repository Location' section with a folder icon and a text field. Below this are two radio buttons: 'Use defaults as specified in Tools->Options' (selected) and 'Override defaults'. The 'Override defaults' section has a folder icon and a text field. Below this are two checkboxes: 'Local' and 'Remote'. The 'Local' checkbox is selected. Below the 'Local' checkbox is a text field with a folder icon. Below the 'Remote' checkbox is a text field. Below these is a checkbox 'Use Remote String Builder'. Below this is a group box containing a 'Connection Type' dropdown, 'Host Name', 'Port Number', 'User Name', and 'Path To Repository' text fields. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

General Runtime **Git Repository**

Git Repository Location _____

☒ Use defaults as specified in Tools->Options
☐ Override defaults

Override Default Location _____

☐ Local

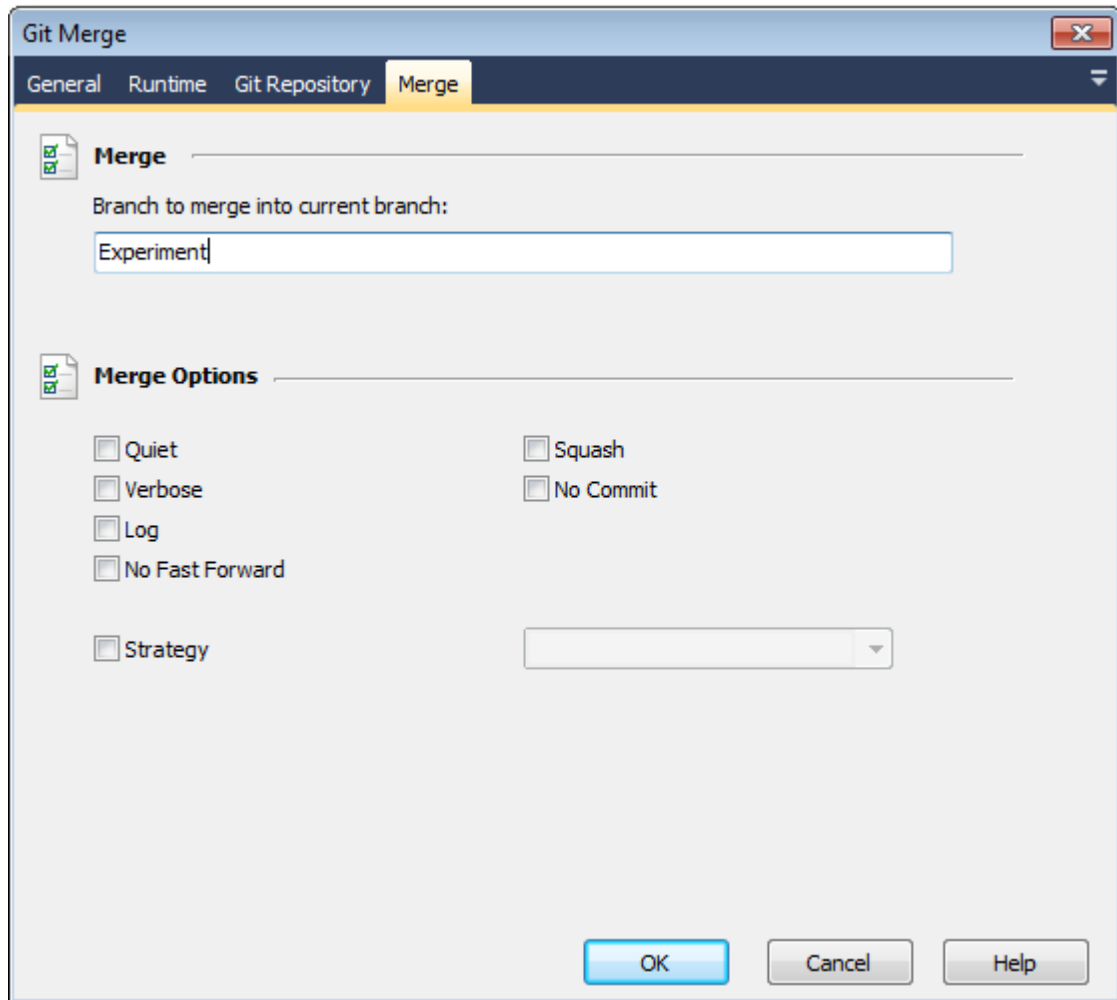
☐ Remote

☐ Use Remote String Builder

Connection Type: _____
Host Name: _____
Port Number: _____
User Name: _____
Path To Repository: _____

OK Cancel Help

Switch to the *Merge* tab to specify which branches you would like to merge. To merge more than two branches simply type the names into the text field separated by a space.



There are also a number of options available on the *Merge* tab:

Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

Log - Include single line descriptions from the actual commits that are being merged in message log.

No Fast Forward - Generate a merge commit even if the merge resolved as a fast-forward.

Squash - Produce the working tree and index state as if a real merge happened, but do not actually make the commit so that the next you run a commit it will create the merge commit.

No Commit - Perform the merge but pretend the merge failed and do not auto commit. This will allow user to make further changes before committing the results.

Strategy - Define the merge strategy to be used:

Resolve - Resolves two heads using a three-way merge algorithm. Considered safe and fast.

Recursive - Resolves two heads using a three-way merge algorithm. This option can detect and handle merges where renames are involved. This is the default strategy when pulling or merging a single branch.

Octopus - Resolves cases with more than two heads. Will not perform a complex merge where manual resolution is required. Intended for bundling topic branches together.

Ours - Resolves any number of branch heads. Result of the merge is the current branch head.

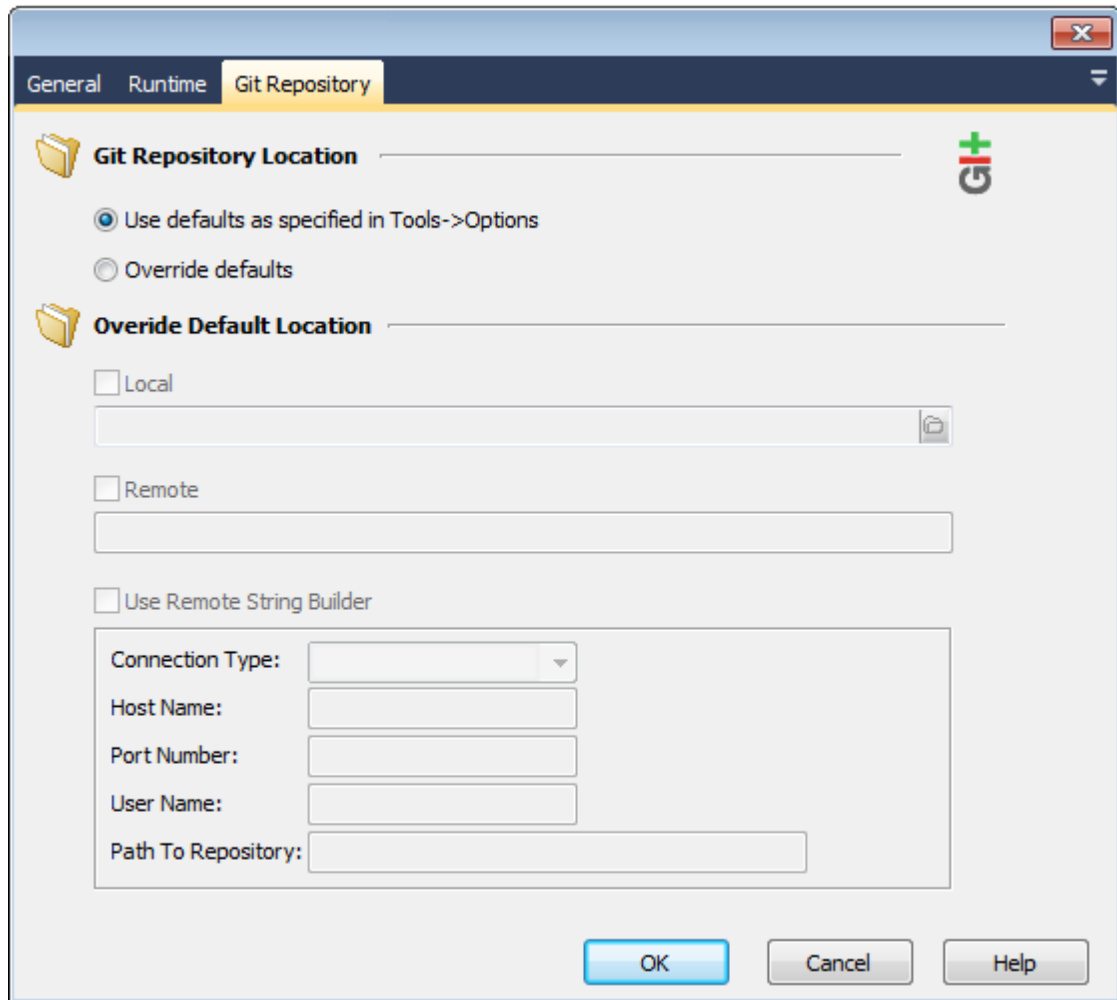
Subtree - Similar to the recursive strategy except this strategy adjusts the tree structure of the branches if one branch is a sub tree of another.

For more information on merging in git, see: [git merge](#)

6.34.5.13 Git Move

The Git Move action allows you to move files and directories within a repository.

From the *Git Repository* tab select the location of the local repository inside which you want to move files and /or directories.



Switch to the *Move Files/Folders* tab to select the destination of the files and directories that you intend to move.

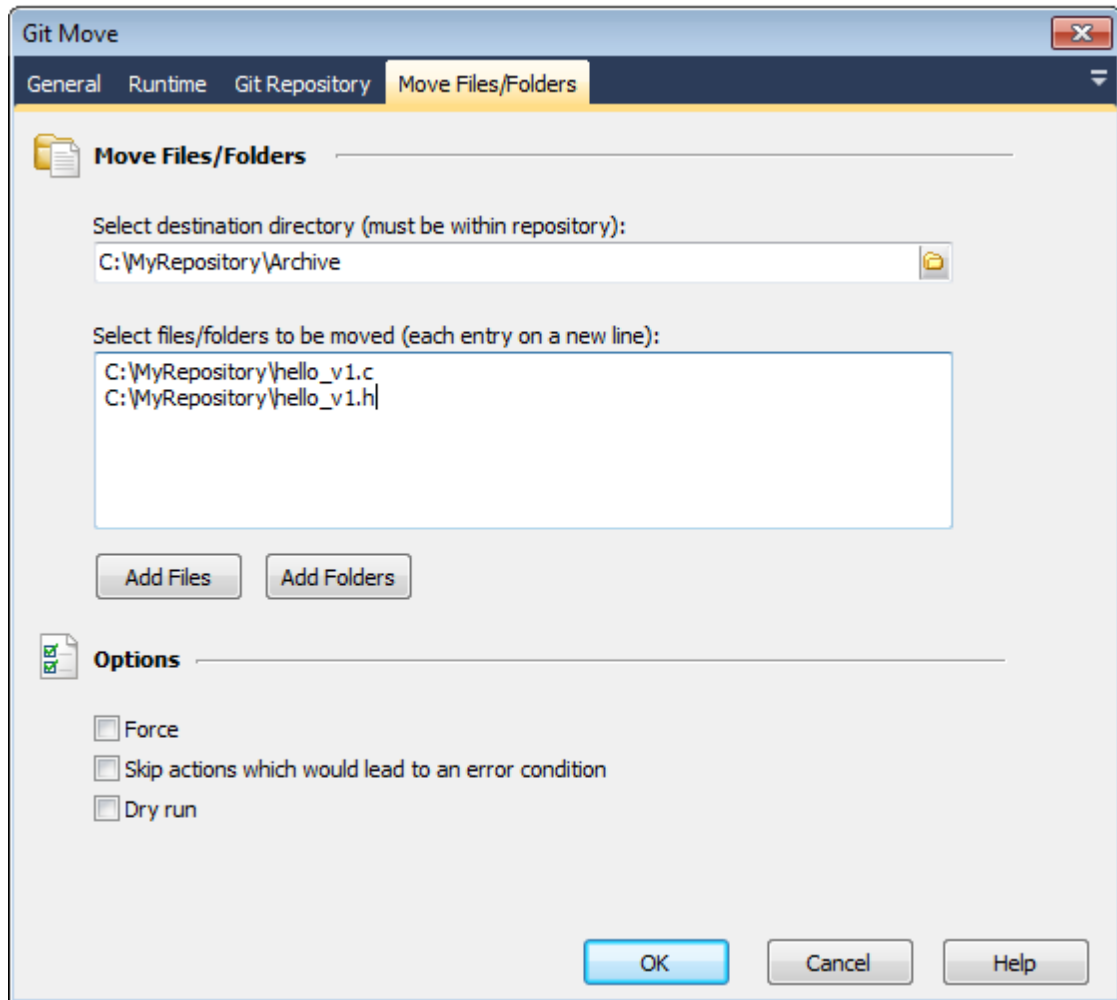
Use the *Add Files* button and *Add Folders* button to select content to be moved. You can also use the asterisk wildcard to move files:

For example:

*To move every file within the repository directory you can use *.**

*To move every file of a particular extension (in this example .txt) *.txt*

*To move files where names contain certain text (in this example "_v2") *_v2.**



There are also a number of options available on the *Move Files/Folders* tab:

Force - Force move even if the destination already exists (will result in overwrite).

Skip actions which would lead to an error condition - In the case that a file/directory doesn't exist or isn't controlled by Git an error occurs. This option allows you to skip files/directories that are going to result in an error.

Dry run - Do not actually move anything, just show what the result would be.

For more information on moving files in git, see: `git mv`

6.34.5.14 Git Pull

The Git Pull action allows you to fetch and merge with another repository into a local branch.

From the *Git Repository* tab select the location of the local repository that you want to *pull into*. To *pull from* a remote repository, provide the location on the same tab (not necessary if you have specified a default which is the one that you want to use).

Git Repository Location

☒ Use defaults as specified in Tools->Options

☐ Override defaults

Override Default Location

☒ Local

☐ Remote

☐ Use Remote String Builder

Connection Type:

Host Name:

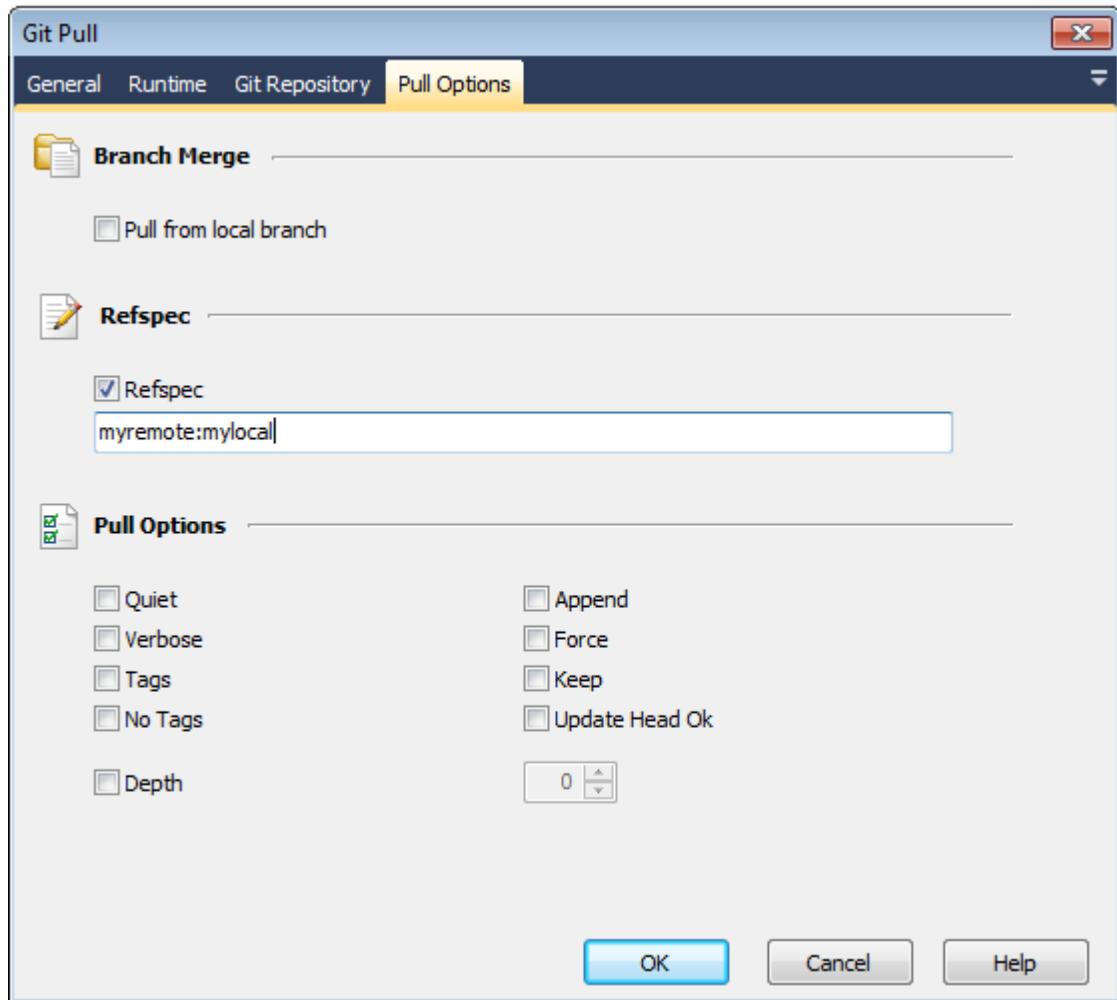
Port Number:

User Name:

Path To Repository:

OK Cancel Help

To *pull from* a local repository go to the *Pull Options* tab and select the *Pull from local branch* option. This will allow you to pull from the current repository.



The Respect option allows you to specify the source and destination refs for the fetch portion of the pull operation. The basic syntax for doing this is `<source ref>:<destination ref>`. For more information about using the refspec option see `git pull`.

There are also a number of options available from the *Pull Options* tab:

Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

Tags - This option specifies that all tags (and their associated objects) should be downloaded, even those that are not reachable from the branch heads that are being tracked (which are omitted by default).

No Tags - Tags that point to objects are downloaded from the remote repository by default. This option disables this from occurring.

Append - Append ref names and object names of the fetched material to the existing contents of `.git/FETCH_HEAD`. If this option is omitted the old data in this location will be overwritten.

Force - When using the refspec option to specify *<remote branch>:<local branch>* the fetch operation will not allow you update the local branch if the remote branch is not a descendant of the local branch. This option forces the fetch to occur regardless.

Keep - Keep downloaded pack.

Update Head Ok - The fetch command will not allow you to update the head which corresponds to the current branch by default. This option disables the check.

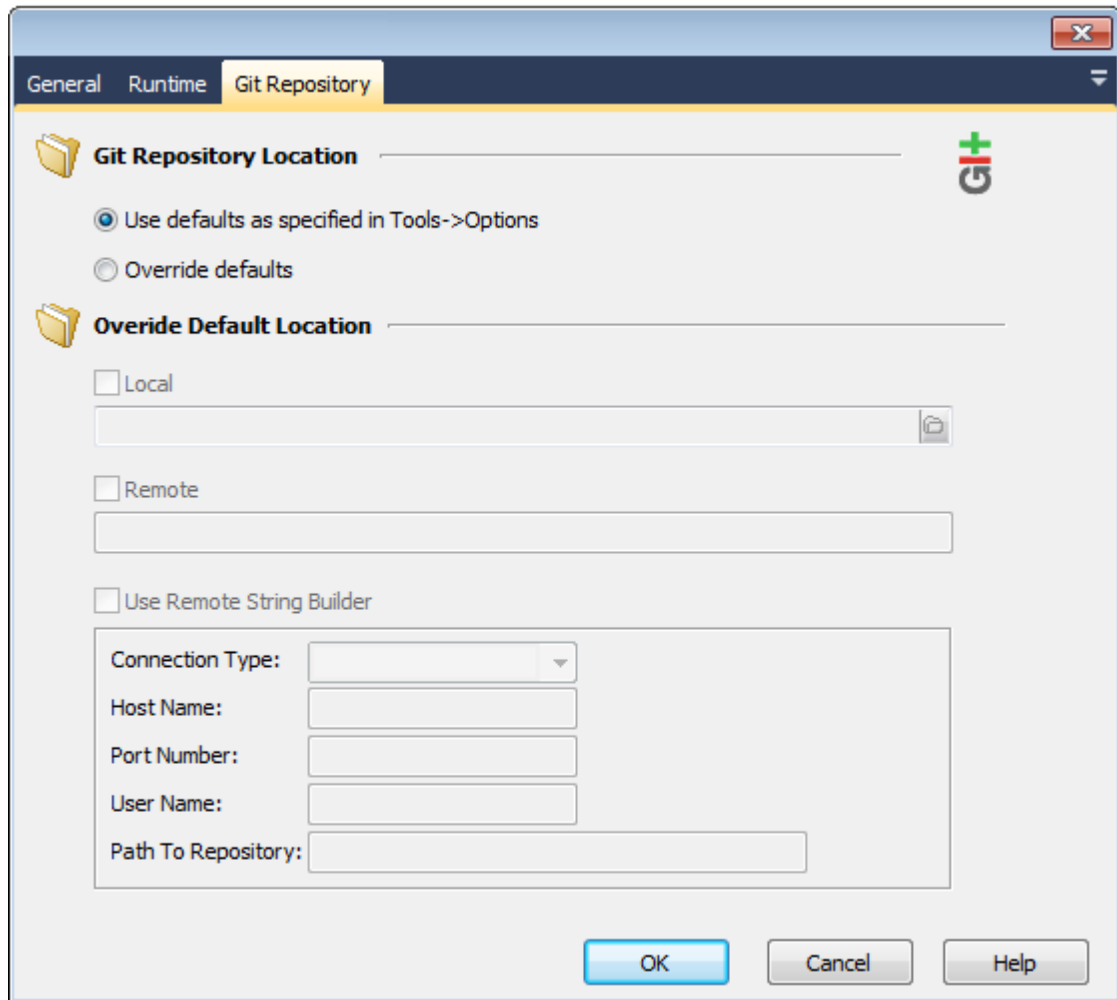
Depth - Increase the history depth of a *shallow* repository by the specified number of commits.

For more information on pulling in git, see: [git pull](#)

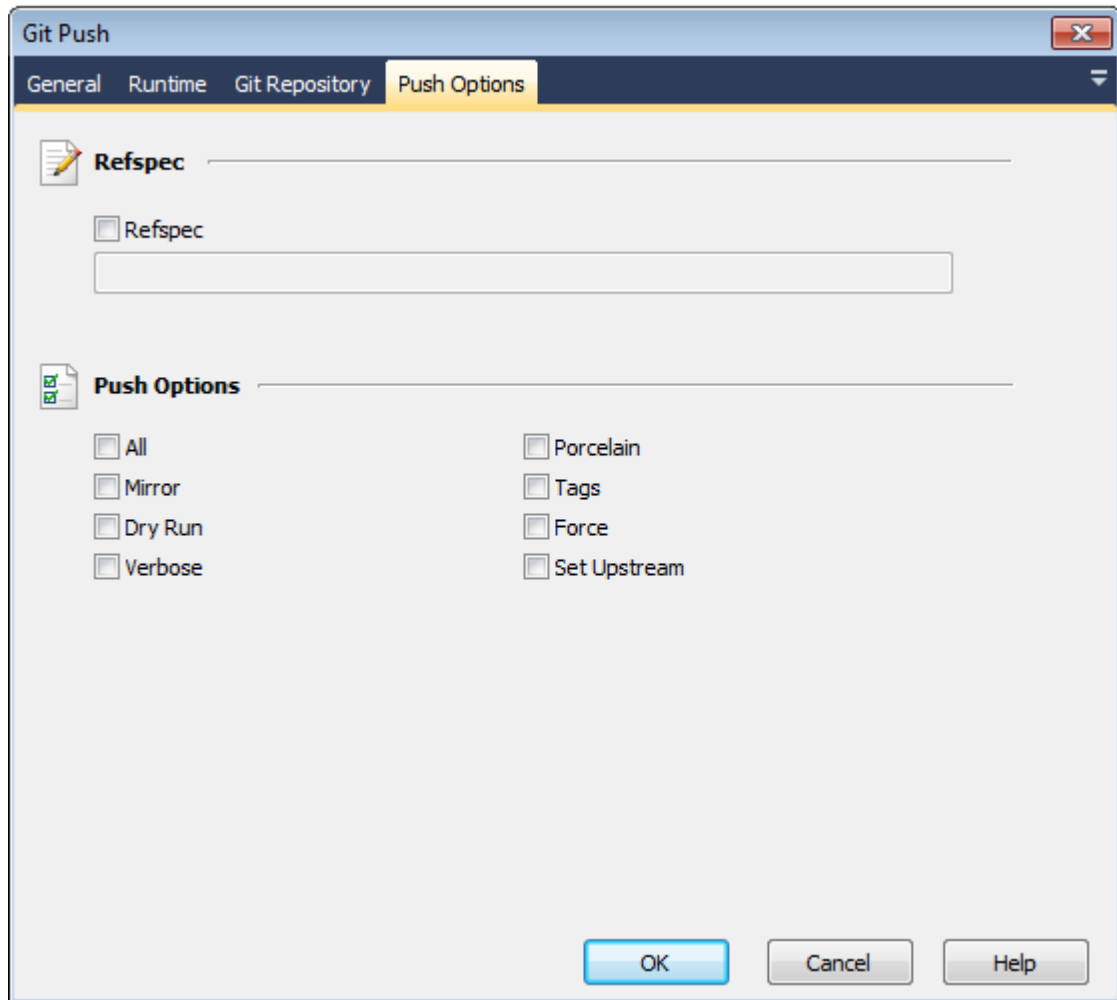
6.34.5.15 Git Push

The Git Push action allows you to push changes to a remote repository.

From the *Git Repository* tab select the location of the local repository that you want to push changes from. Also on this tab specify the remote repository that you want to push changes to.



By default the Push action will push local branches to matching branches that exist on the remote repository. To override this behaviour we can use the Refspec option from the *Push Options* tab.



The syntax for overriding this behaviour is `<source branch>:<destination branch>`. However if you want to update a remote branch which exists under the same name locally, you can just type the name of the branch into the field (omitting the source branch and the colon).

To delete a remote branch using this action, leave the source branch blank and enter the colon and the name of the remote branch that you wish to delete into the refspec field. For more information about using the refspec option with this action see `git push`.

There are a number of other options available from the *Push Options* tab:

All - Pushes all refs under *refs/heads*.

Mirror - Mirrors everything under *refs/* to the remote repository.

Dry Run - Do everything except actually send the updates.

Verbose - Verbose output from Git.

Porcelain - Produce machine readable output.

Tags - All refs under *refs/tags* are pushed in addition to the refs listed on the command line.

Force - Overrides the safety check that prevents you from pushing to a remote repository that is not an ancestor of the local repository that is attempting to push to it. Using this option can result in a loss of commits from the remote repository.

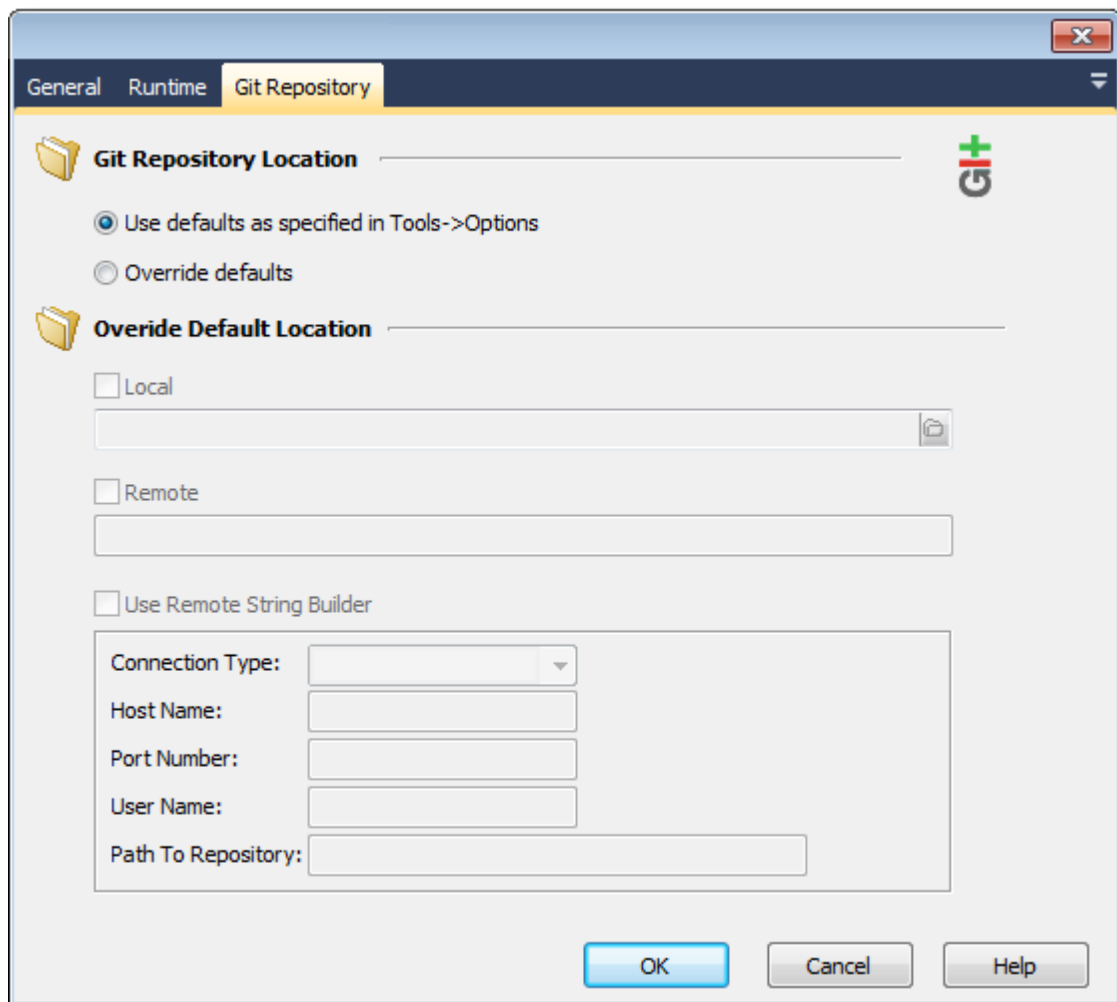
Set Upstream - When a push is successful add an upstream tracking reference.

For more information on pushing in git, see: [git push](#)

6.34.5.16 Git Remove Files

The Git Remove Files action allows you to remove files and directories from a repository.

From the *Git Repository* tab select the location of the local repository that you want to remove content from.



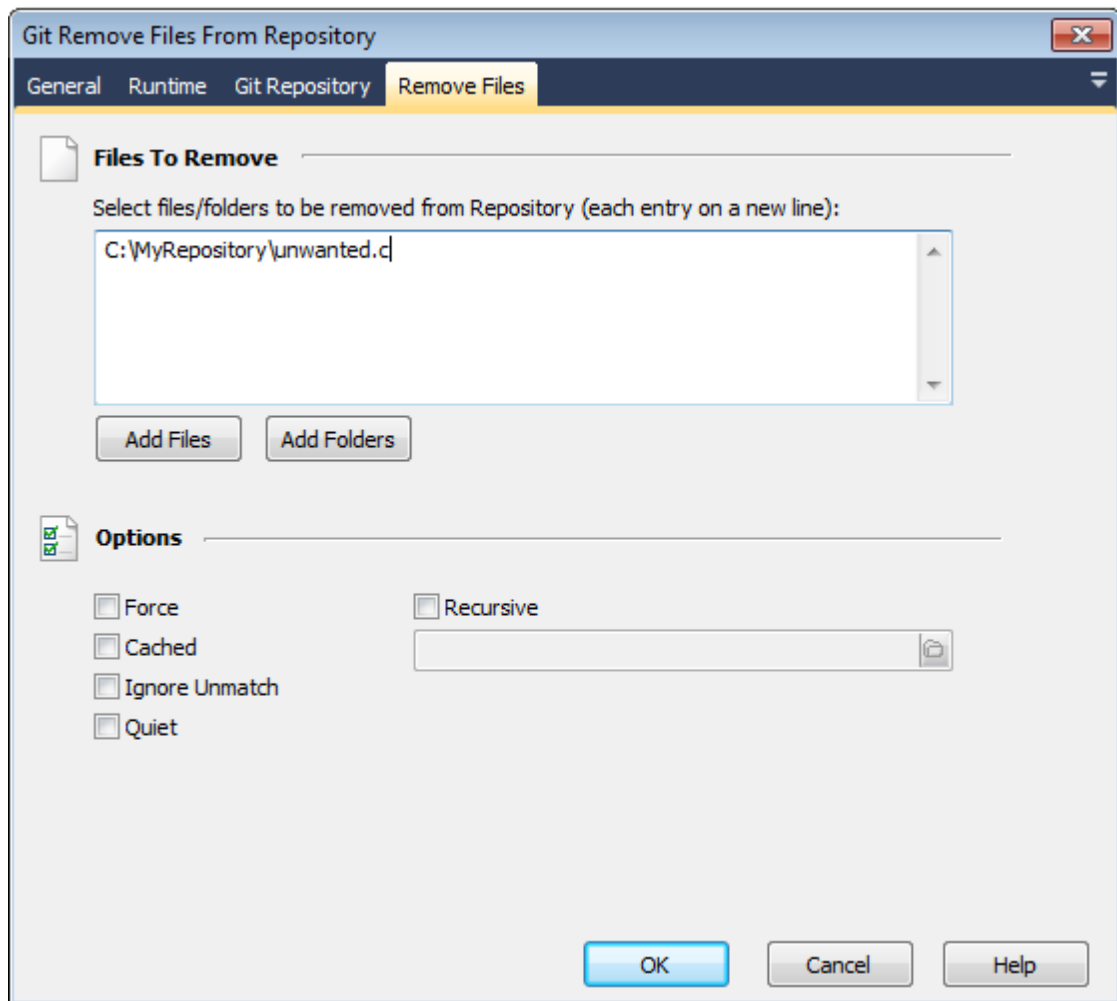
Switch to the *Remove Files* tab to specify the files and/or directories to remove from the repository. Use the *Add Files* button and the *Add Folders* button to select the content to be removed from the repository. Place each entry on a new line. You can also use the asterisk wildcard to remove files.

For example:

*To remove every file within the repository directory you can use *.**

*To remove every file of a particular extension (in this example .txt) *.txt*

*To remove files where names contain certain text (in this example "_v2") *_v2.**



The *Remove Files* tab also provides a number of options that can be executed:

Force - Override the up-to-date check.

Cached - Use this option to unstage files by removing them from the index, without actually removing files from the working tree.

Ignore Unmatch - This option means that the action will pass even if no files are matched for removal. This also disables the validity checking that each file/directory is subjected to before being flagged for removal.

Quiet - Suppress output from Git.

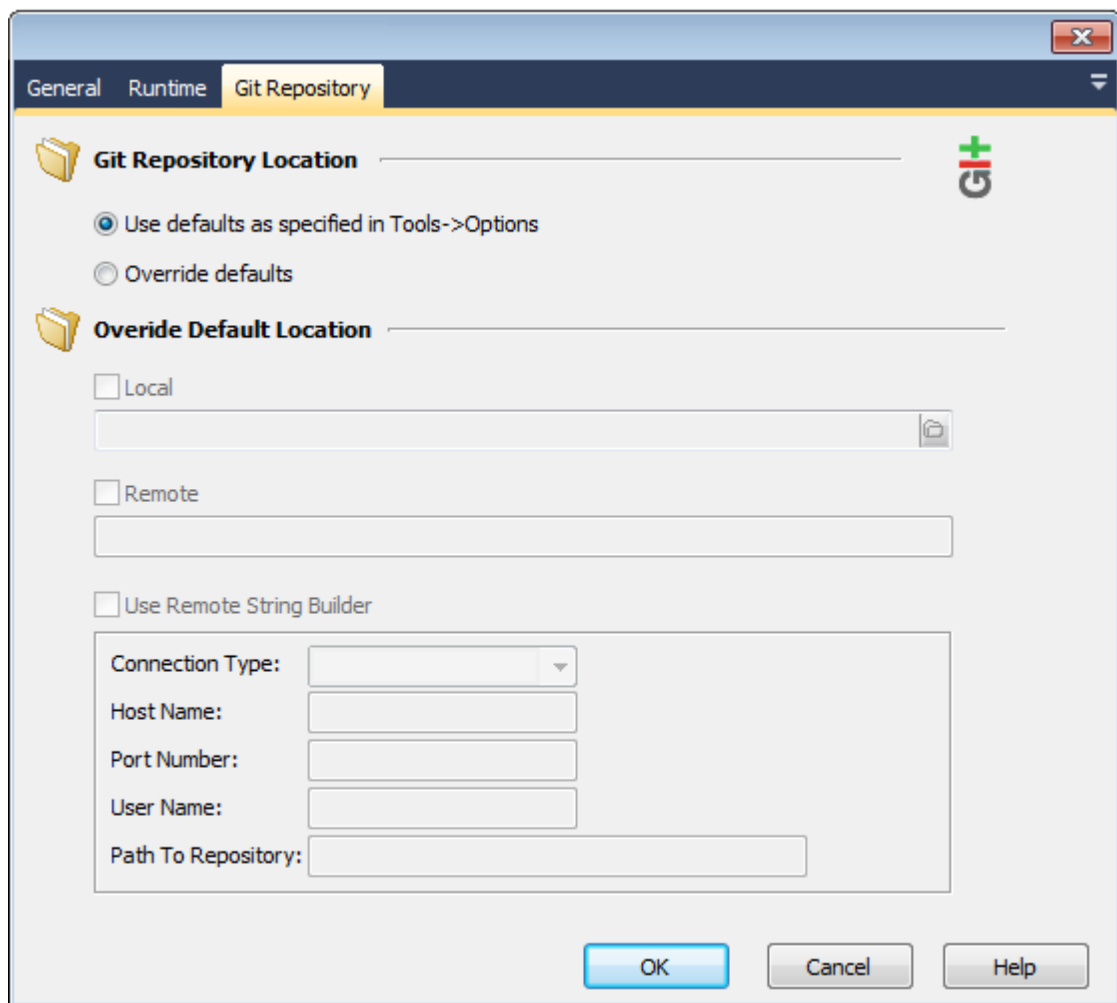
Recursive - Allow a recursive removal by providing a directory.

For more information on removing files in git, see: `git rm`

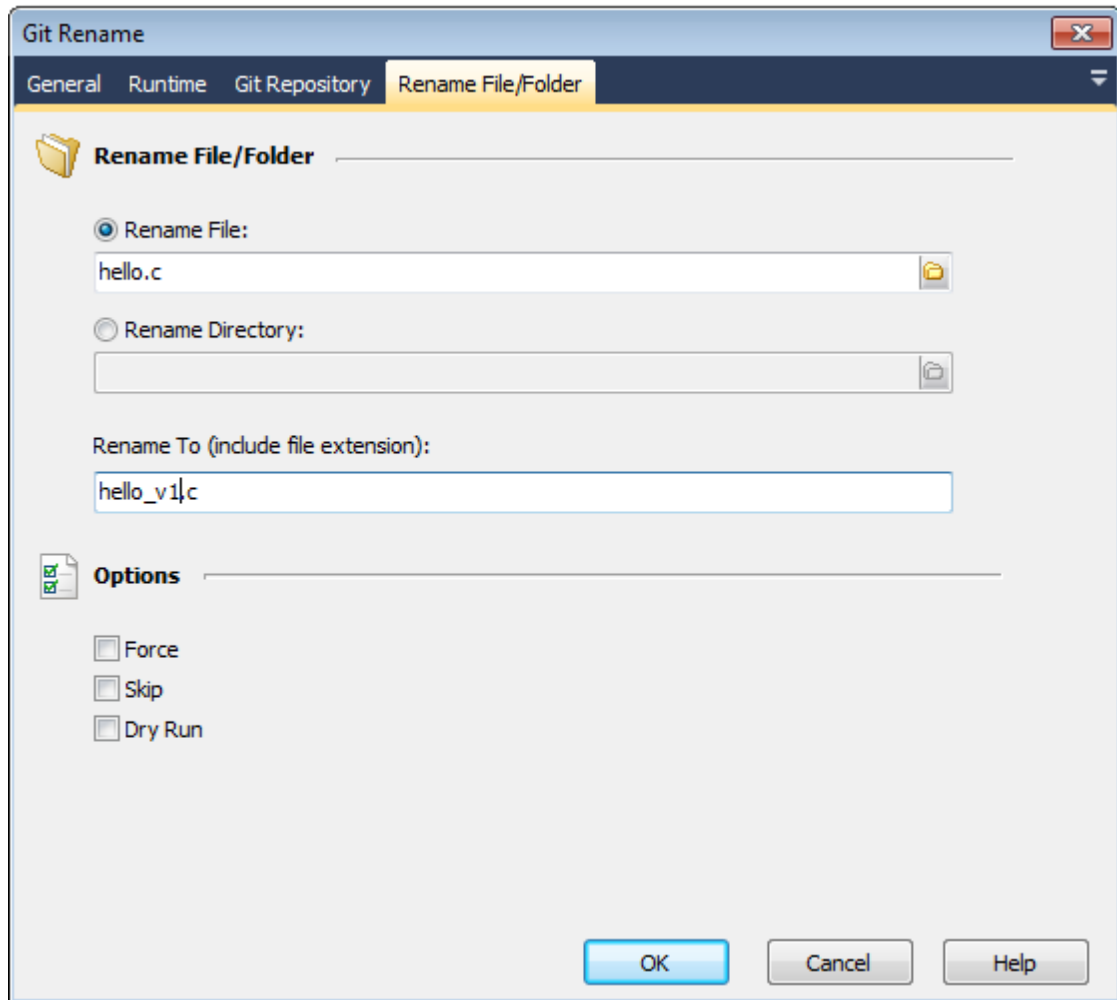
6.34.5.17 Git Rename

The Git Rename action allows you to rename files and directories within a repository.

From the *Git Repository* tab select the location of the local repository inside which you want to rename files and/or directories.



From the *Rename File/Folder* use the radio buttons to select whether you want to rename a file or a directory and select the relevant item to be renamed.



Enter a name in the *Rename To* field, ensuring that the name is valid (does not contain special characters). If renaming a file, remember to include the file extension in the *Rename To* field otherwise the file will be renamed without an extension.

There are also three options available from the *Rename File/Folder* tab:

Force - Force renaming the file/directory even if the target exists. This can result in data been overwritten.

Skip - In the case that a file/directory doesn't exist or isn't controlled by Git an error occurs. This option allows you to skip files/directories that are going to result in an error.

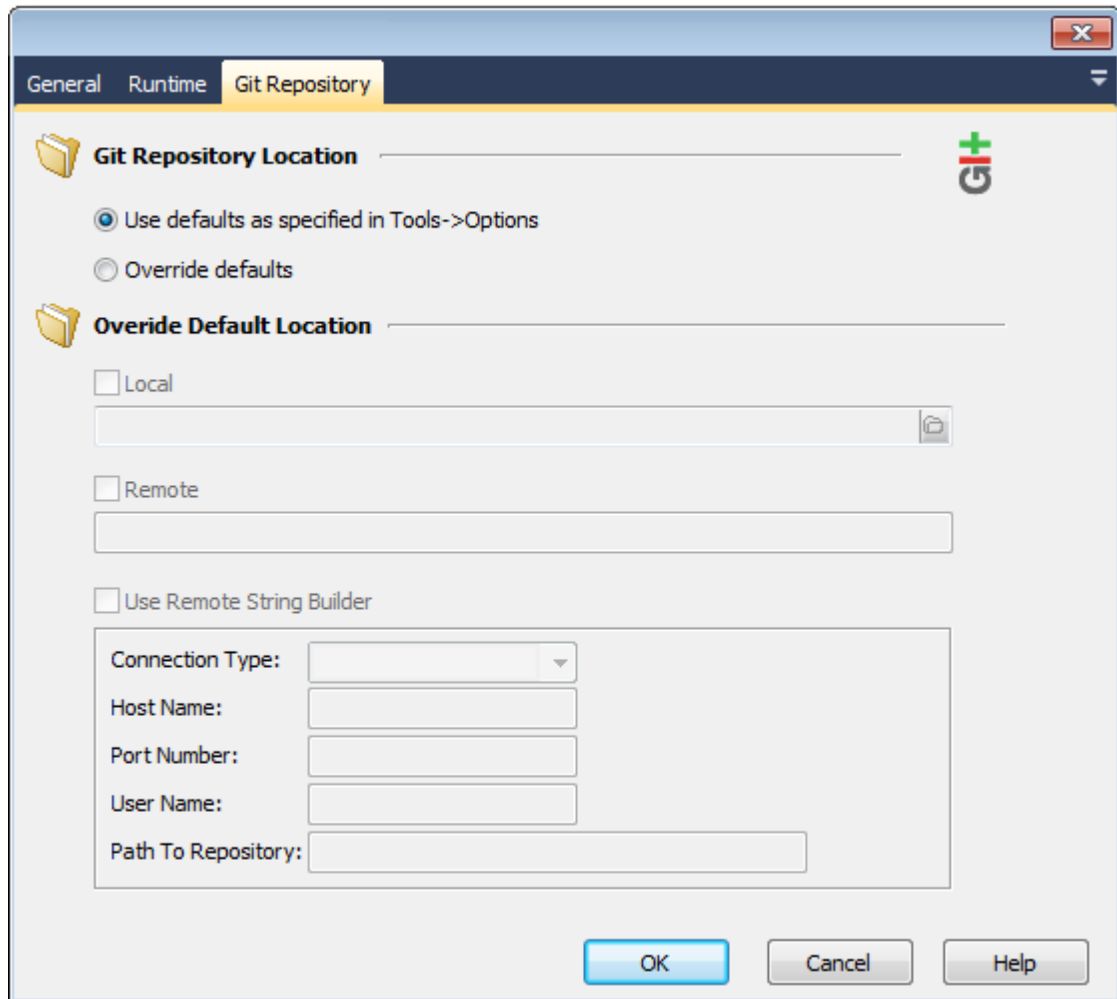
Dry Run - Do not actually rename anything, only show what would happen.

For more information on renaming files in git, see: `git mv`

6.34.5.18 Git Repository Status

The Git Repository Status action allows you to view the status of your local repository in terms of changes to be committed, changes to be updated and untracked files.

From the *Git Repository* tab select the location of the local repository which you want to display the status of.

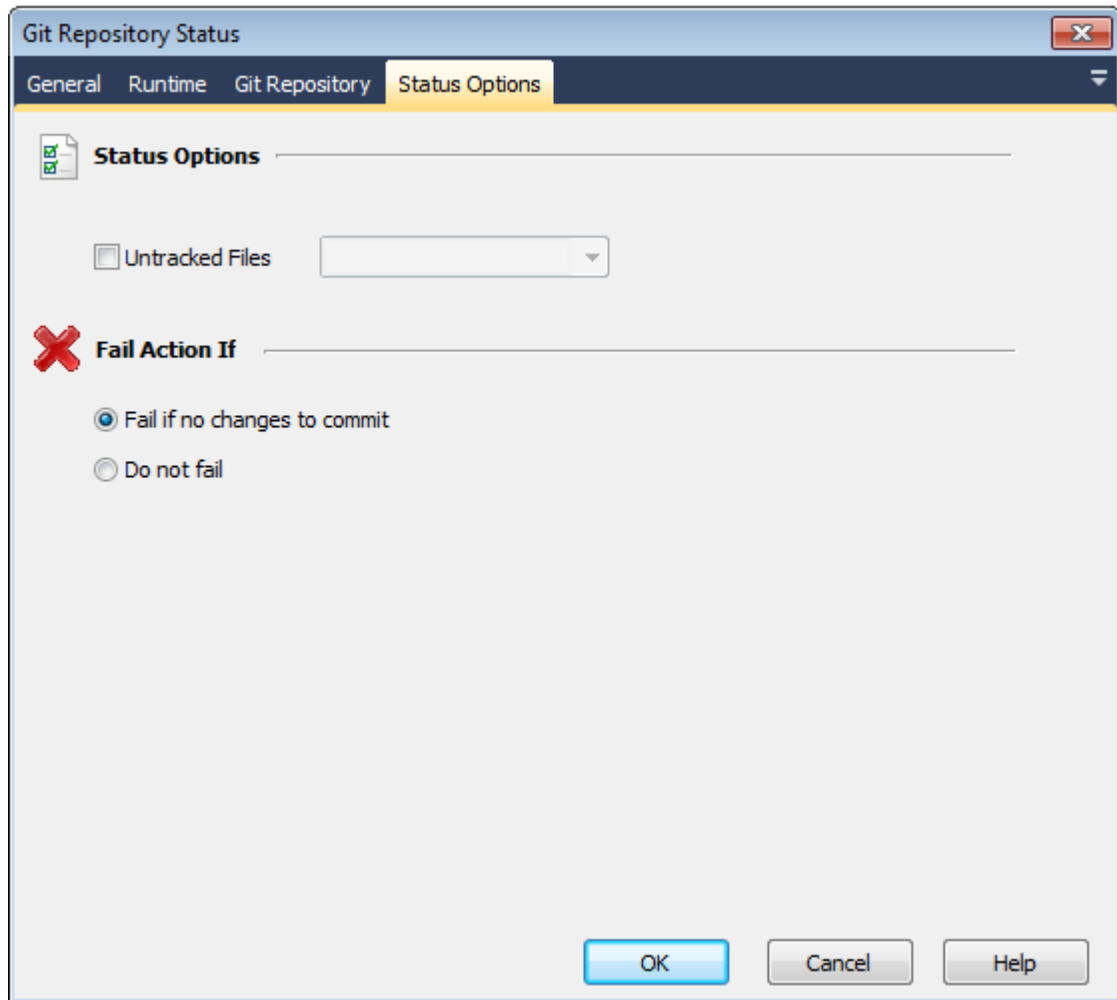


There is an option to specify how to handle untracked files on the *Status Options* tab. The *Untracked Files* option operates in three modes:

No- Do not show untracked files.

Normal - Show untracked files and directories.

All - Show untracked files and individual files within untracked directories.



By default, if there are no changes to commit the action will fail. To override this behaviour use the *Fail Action If* options.

Fail if no changes to commit - If there are no changes to commit the action will fail.

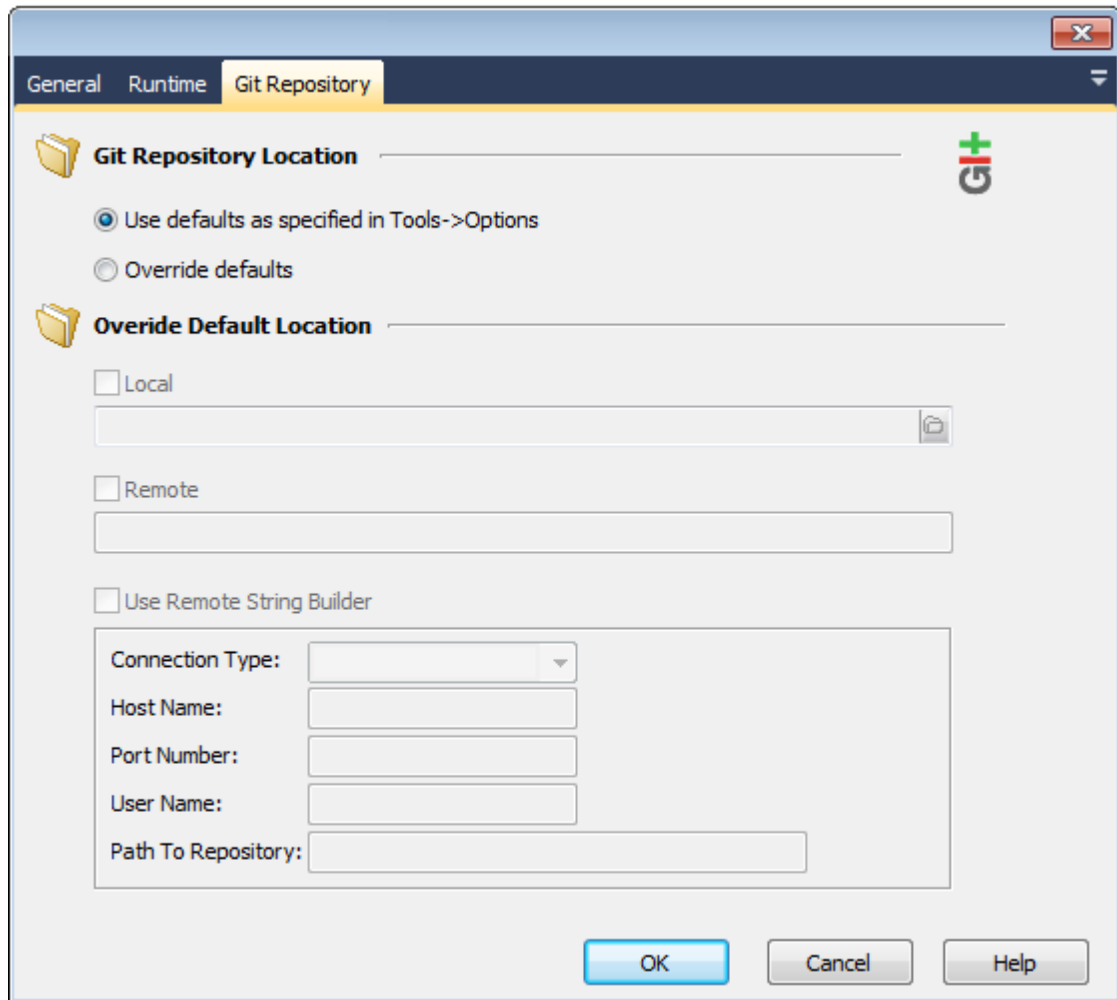
Do not fail - The action will pass regardless of whether or not there are changes to commit. If there are no changes to commit a warning message will be written to the log.

For more information on viewing repository status in git, see: `git status`

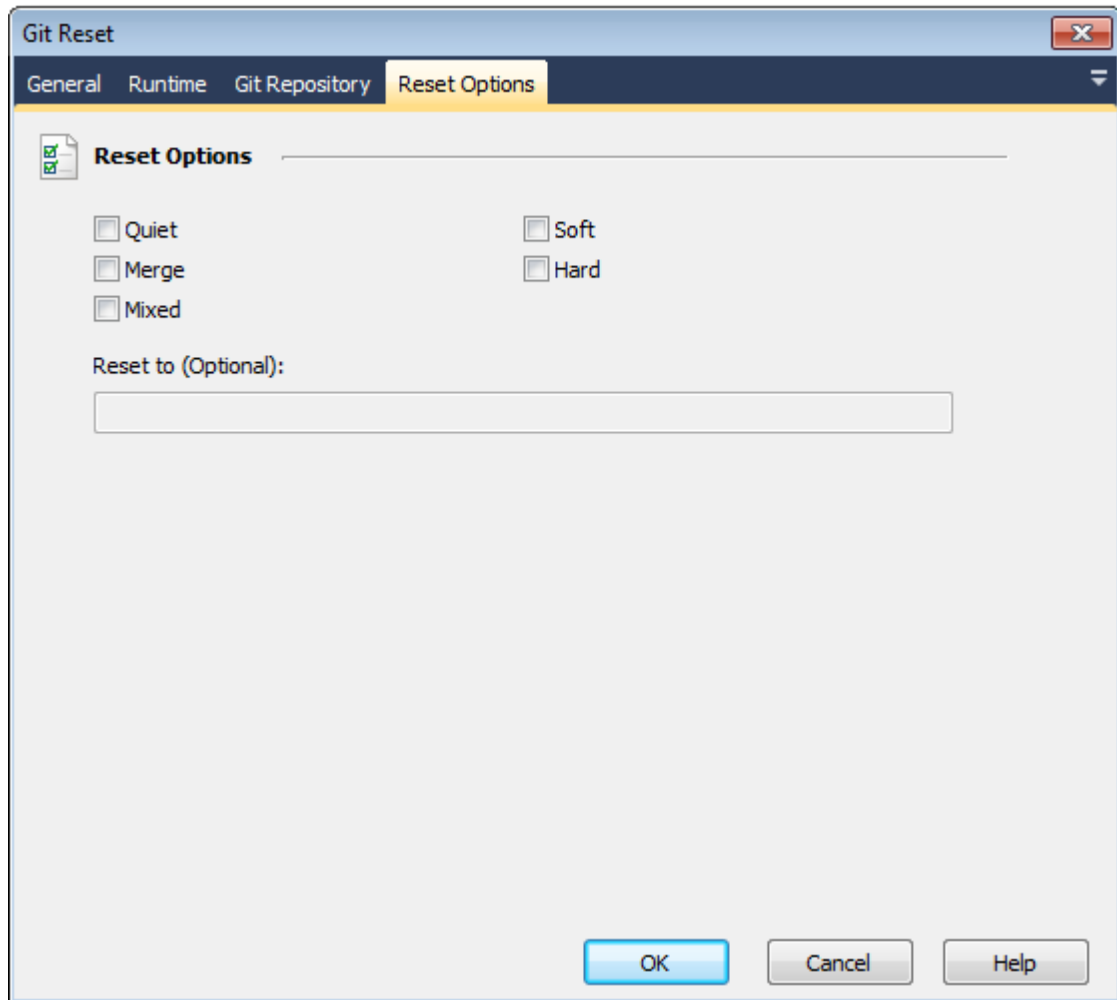
6.34.5.19 Git Reset

The Git Repository Status action allows you to reset the current HEAD to a specified state.

From the *Git Repository* tab select the location of the local repository of which you would like to reset.



There are a number of options available from the *Reset Options* tab:



Quiet - Suppress output from Git.

Merge - Reset the index to match to the tree recorded by the named commit and updates files where there are differences between the named commit and the commit in the current working tree.

Soft - Index and working tree are not touched. All files that have been changed remain in *Changes to be commit* status.

Hard - Matches the working tree and the index of the tree being switched to. Any changes to tracked files since the specified commit will be lost.

Mixed - Resets the index but not the working tree and reports what has not been updated.

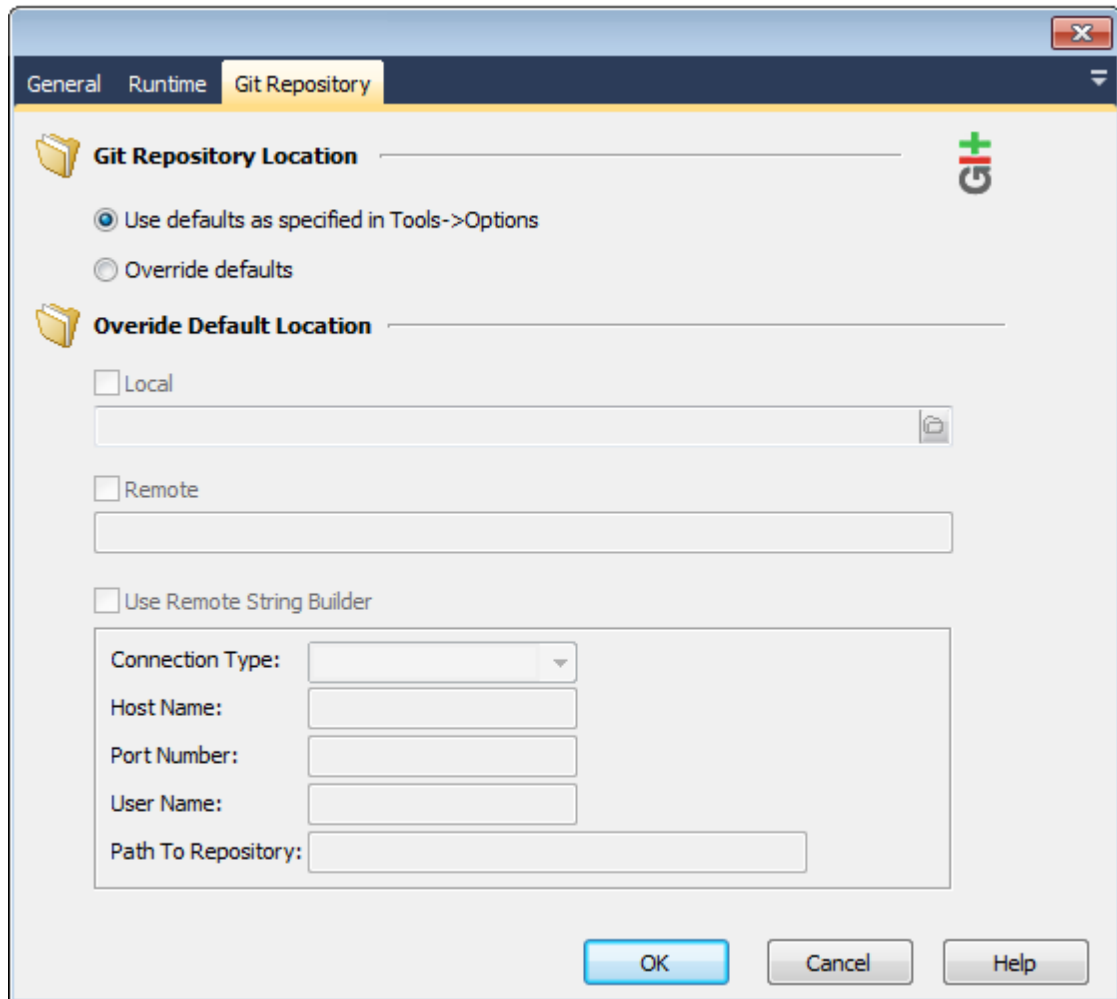
Reset To: This field allows you to provide a commit-id to reset the HEAD to. This value is optional, if left blank it defaults to HEAD.

For more information on performing a reset in git, see: `git reset`

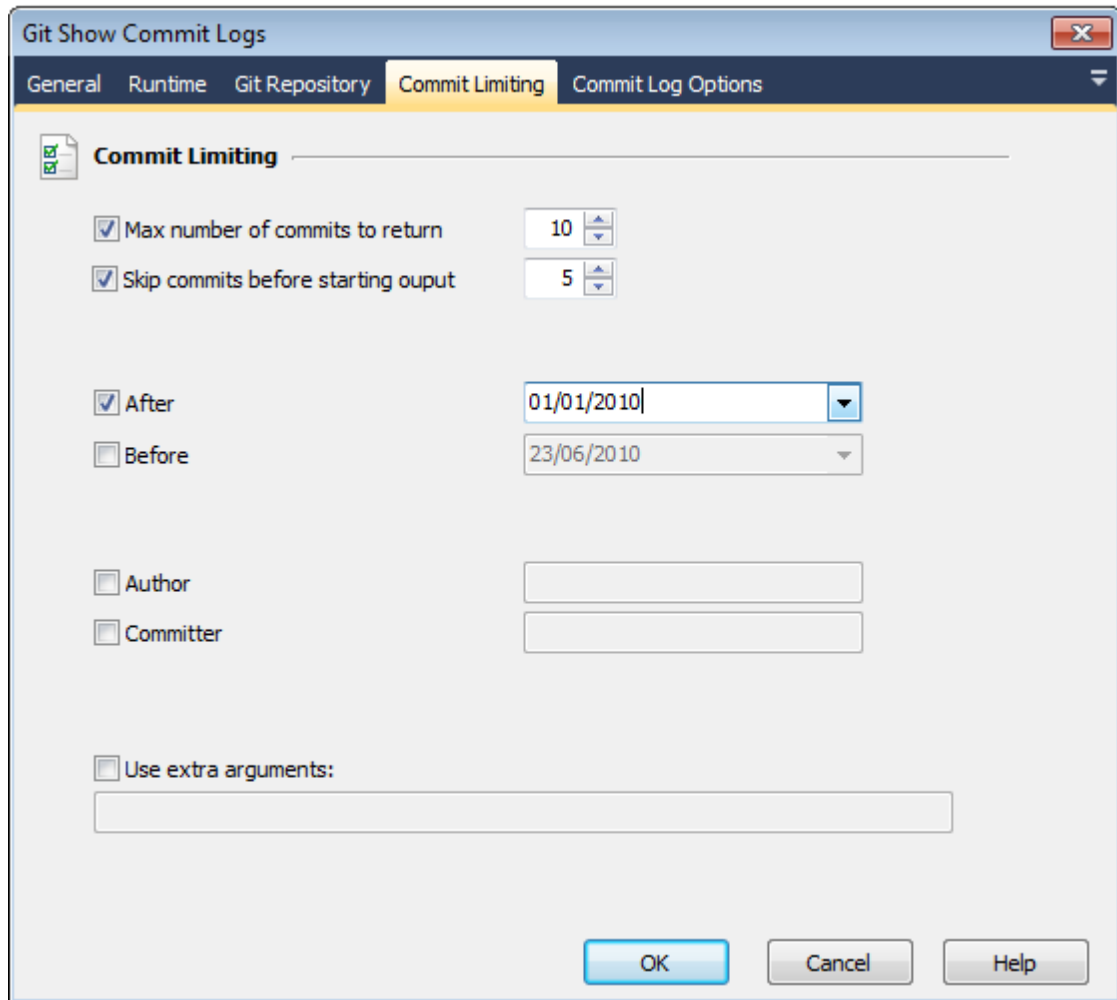
6.34.5.20 Git Show Commit Logs

The Git Show Commit Logs action allows you to display the commit history for a local repository.

From the *Git Repository* tab select the location of the local repository that you want to view the commit history of.



There are a number of options available to limit the amount of commits that are displayed on the *Commit Limiting* tab.



Max number of commits to return - Specify maximum number of commits to display.

Skip commits before starting output - Specify the number of commits to skip before starting to display commits.

After - Only display commits that occurred after the specified date.

Before - Only display commits that occurred before the specified date.

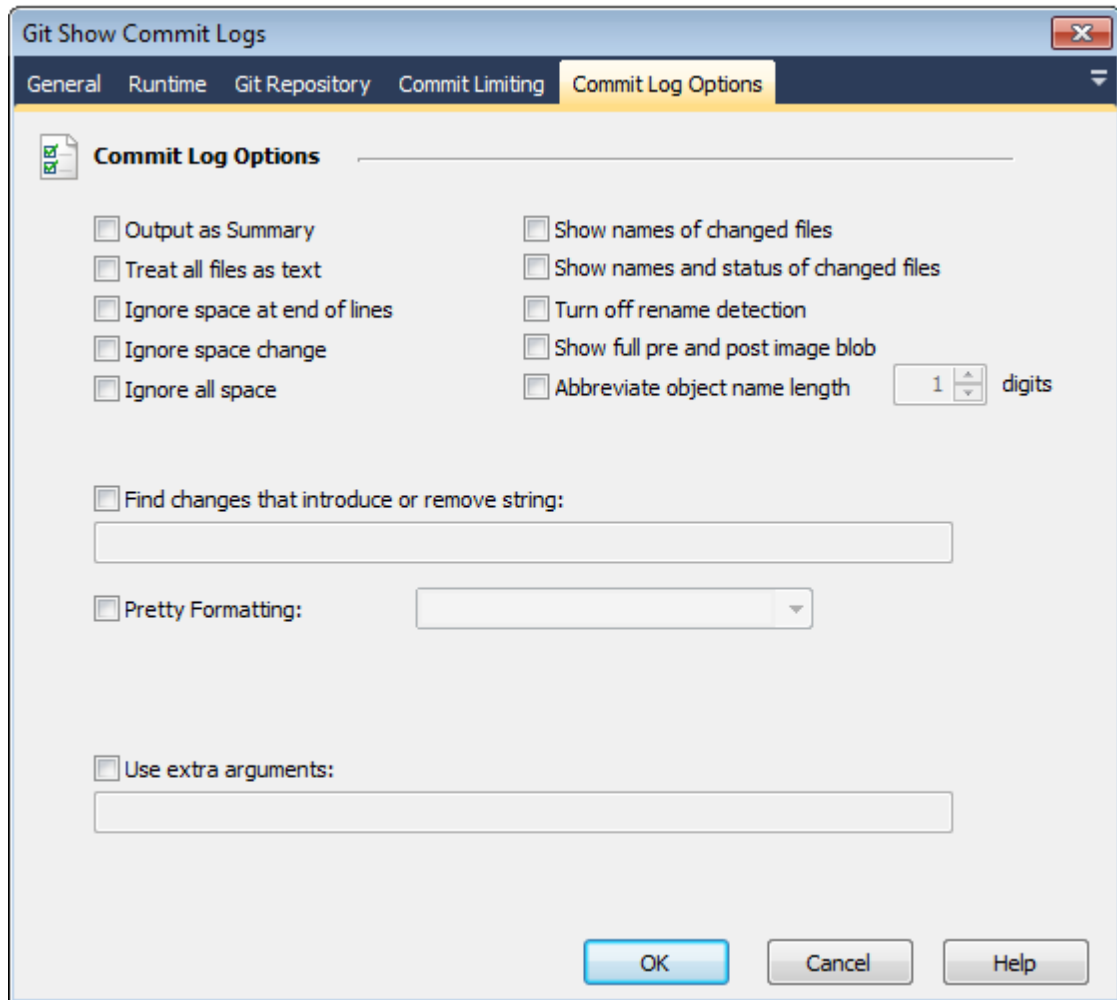
Author - Only display commits from the specified author.

Committer - Only display commits from the specified committer.

Use extra arguments - This option allows you to specify extra arguments to use for commit limiting.

Note: Any syntax errors in the Use Extra Arguments option will result in the action failing.

There are a number of general options available from the *Commit Log Options* tab.



Output as Summary - Output a summary of header information.

Treat all files as text - self explanatory.

Ignore space at end of lines - Ignore changes in white space at end of lines.

Ignore space change - Ignore changes in the amount of white space. Considers all sequences of one or more white space to be equivalent.

Ignore all space - Ignore all space when comparing lines.

Show names of changed files - Only the names of changed files will be displayed.

Show names and status of changed files - Only show the names and the status of changed files.

Turn off rename detection - Turn off rename detection.

Show full pre and post image blob - Rather than displaying the first few characters, show the

entire blob objects names.

Abbreviate object name length - Specify how many object name digits should be displayed.

Find changes that introduce or remove string - List changes that introduce or remove an instance of the string value provided.

Pretty Formatting - This option provides you with a number of modes for formatting the output of this action:

Online - Commit-Id and title line all on one line.

Short - Commit-Id, author and commit message title.

Medium - Commit-Id, author, date, title line and commit message. This is the format that is used by default when *Pretty Formatting* is not enabled.

Full - Commit-Id, author, committer, title line and commit message.

Fuller - Commit-Id, author, author date, committer, committer date, title line and commit message.

Email - Output the same information as the medium option except in the format of an email.

Raw - Displays the information exactly as it is stored in the commit object.

Format - Selecting this mode presents a text field that enables you to specify a custom format. To obtain a list of placeholders and see examples see `git log`.

Use extra arguments - This option allows you to specify extra arguments to use for commit log options.

Note: Any syntax errors in the Use Extra Arguments option will result in the action failing.

For more information on viewing the commit logs in git, see: `git log`

6.34.6 JediVCS

The JEDI VCS actions were kindly written and donated by Mathias Burbach (Maranatha Consulting).

JEDI VCS is an Open Source VCS which was based on FreeVCS. More information at <http://jedivcs.sourceforge.net/>

The available JediVCS actions are:

- JediVCS Synchronize Project
- JediVCS Label Project
- JediVCS Check Out
- JediVCS Undo Check Out
- JediVCS Check In
- JediVCS List Projects
- JediVCS List Project Modules
- JediVCS List Locks
- JediVCS Synchronize Project For Label

The JediVCS actions use the command line interface of jvcs. All the available options of jvcs are made available in the FinalBuilder plugins. For more information on the command line options, please refer to the "Jedi VCS Command Line Client.html" located in the JediVCS installation directory.

Note that the JediVCS actions output the used command line options to the FinalBuilder log, where they can be further scrutinised.

The JediVCS actions have global options, which can be set via the Tools Menu, Options. The global options allow you to set the following details:

- Location of JediVCS executable
- Default server name and port
- Default login/password

Each action can then either use the defaults, or override the defaults with specific values.

6.34.6.1 Scripting JediVCS actions

Scripting ActionStudio based actions is slightly different as you have to access the properties of the action via the PropertyStore, eg:

```
Action.PropertyStore.OverrideServerName = "localhost"
```

Below are the properties of the JediVCS actions:

JediVCS Synchronize Project

Properties

property: OverrideGlobalOptions	Type: boolean	Default: none
property: OverrideServerName	Type: string	Default: "localhost"
property: OverridePort	Type: integer	Default: 2106
property: OverrideUserName	Type: string	Default: none
property: OverridePassword	Type: string	Default: none
property: ProjectName	Type: string	Default: none

property: NoPrompt Type: boolean Default: True
property: CheckBy Type: string Default: "stamp"
property: CrossRef Type: boolean Default: False
property: IncludeHidden Type: boolean Default: False
property: ExtExitCode Type: boolean Default: False

JediVCS Label Project

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: Label Type: string Default: none

JediVCS Check Out

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: ModuleName Type: string Default: none

JediVCS Undo Check Out

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: ModuleName Type: string Default: none

JediVCS Check In

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: ModuleName Type: string Default: none
property: Comment Type: string Default: none

JediVCS List Projects

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"

property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectListVariable Type: string Default: none

JediVCS List Project Modules

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: FileName Type: string Default: none

JediVCS List Locks

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: CheckedOutBy Type: string Default: none
property: FileName Type: string Default: none

JediVCS Synchronize Project For Label

Properties

property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword Type: string Default: none
property: ProjectName Type: string Default: none
property: Label Type: string Default: none
property: NoPrompt Type: boolean Default: True
property: CheckBy Type: string Default: "stamp"
property: CrossRef Type: boolean Default: False
property: IncludeHidden Type: boolean Default: False
property: ExtExitCode Type: boolean Default: False

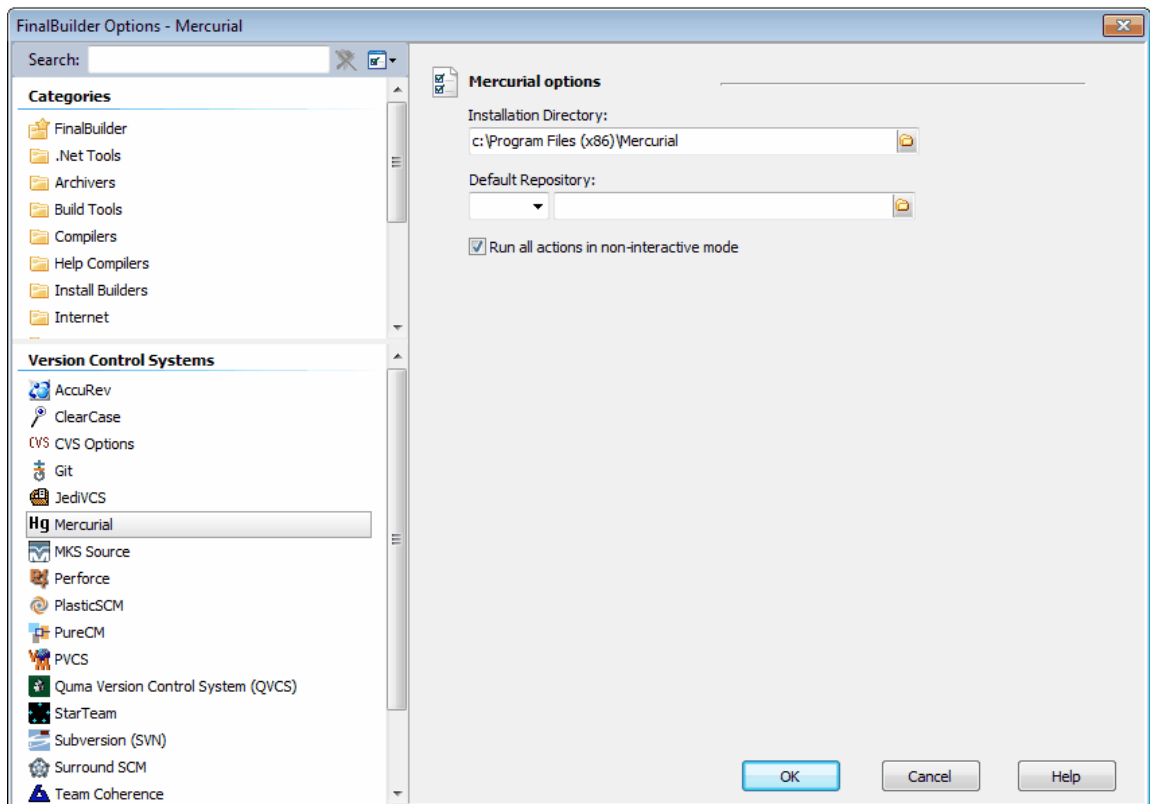
6.34.7 Mercurial

The Mercurial actions allow you to interact with the Mercurial distributed version control system.

For more information about Mercurial, see <http://mercurial.selenic.com>. For an introductory walk-through of using FinalBuilder and Mercurial, see Using Mercurial with FinalBuilder.

Global Options

Before you can use the Mercurial actions you need to set the global Mercurial options. You can do this by going to [Tools\Options], selecting [Version Control Systems] and then [Mercurial]. You will see the following screen:

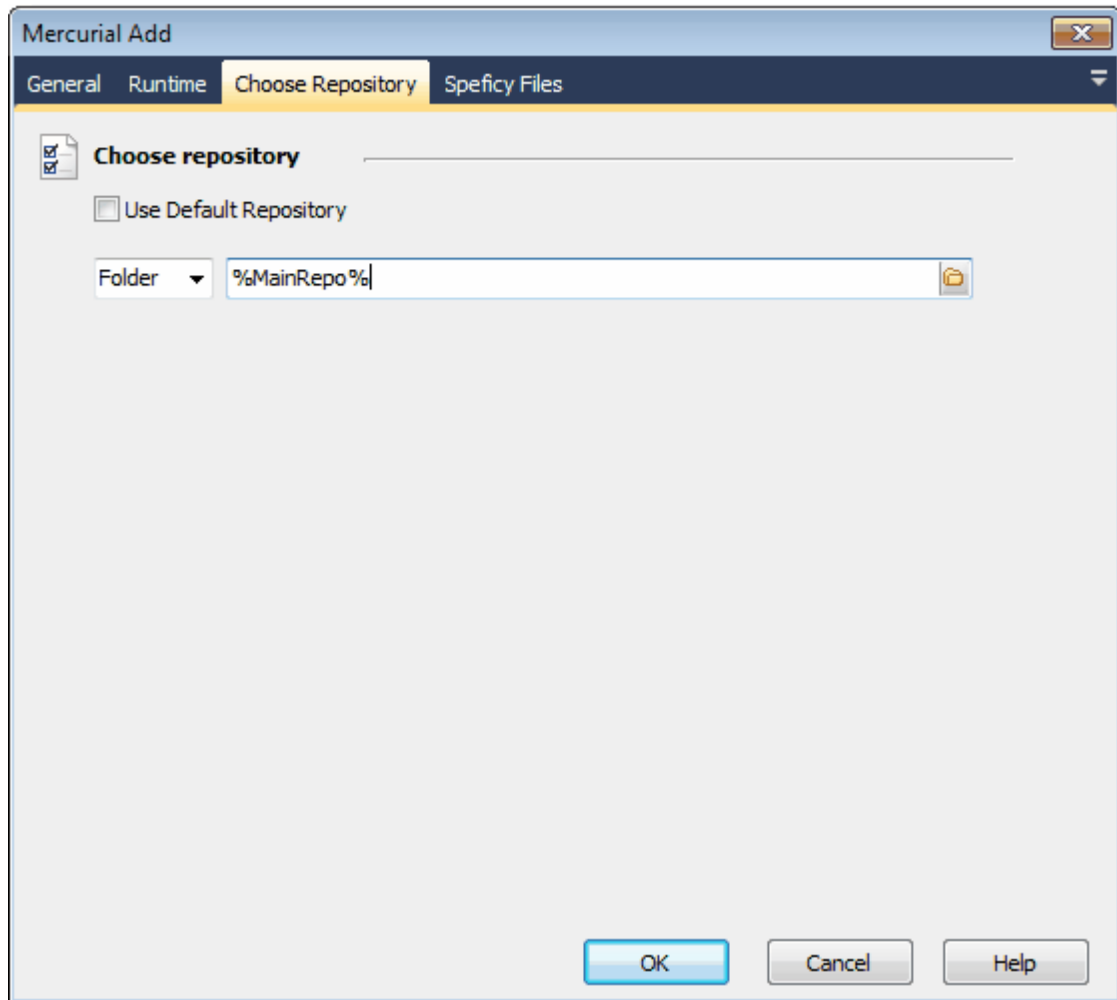


You must specify the directory where Mercurial is installed. Optionally, you can specify a default repository which most Mercurial actions will make use if specified. You can choose to override it in each action.

If you want to run Mercurial actions as part of an unattended build, you should tick the "Run all actions in non-interactive mode" option. This will, where possible, stop Mercurial from prompting for user input.

6.34.7.1 Choosing Repositories

All Mercurial actions require you to choose at least one repository to work on.



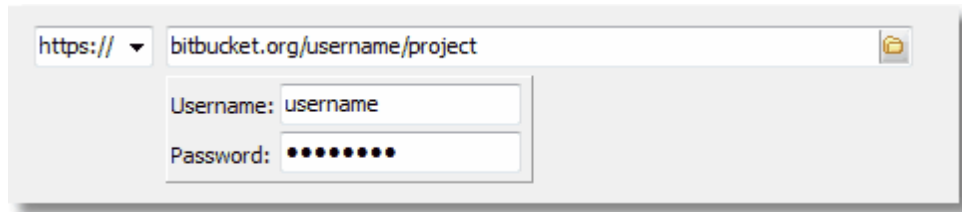
By default, the repository specified in the global options will be used. You can override this and specify a folder, HTTP, HTTPS or SSH repository.

Mercurial supports almost all actions for all repository types, although there are some unexpected behaviours. For example, if you use a remote repository as the target for a Clone, it will create the repository but not any of the files. For that reason, the general best practice is only use 'read' actions on remote repositories (such as the Clone source, Pull etc). All 'write' (Add, Update, Commit, Clone target etc) actions should be performed on local repositories.

The exception is Push, which is the recommended method for moving local changesets to a remote repository.

Access HTTP(S) Repositories

Choose the repository type, then specify the URL. You do not need to include the protocol in the URL, but you can if you want to. If you choose HTTPS you will be able to enter a username and password:



Your password is stored as an encrypted string in the FinalBuilder project file.

Accessing SSH Repositories

Mercurial provides SSH support by calling out to an external SSH application, which you need to configure. Getting Started with Mercurial has instructions for how to set up SSH and the same process needs to be followed to use Mercurial, SSH and FinalBuilder.

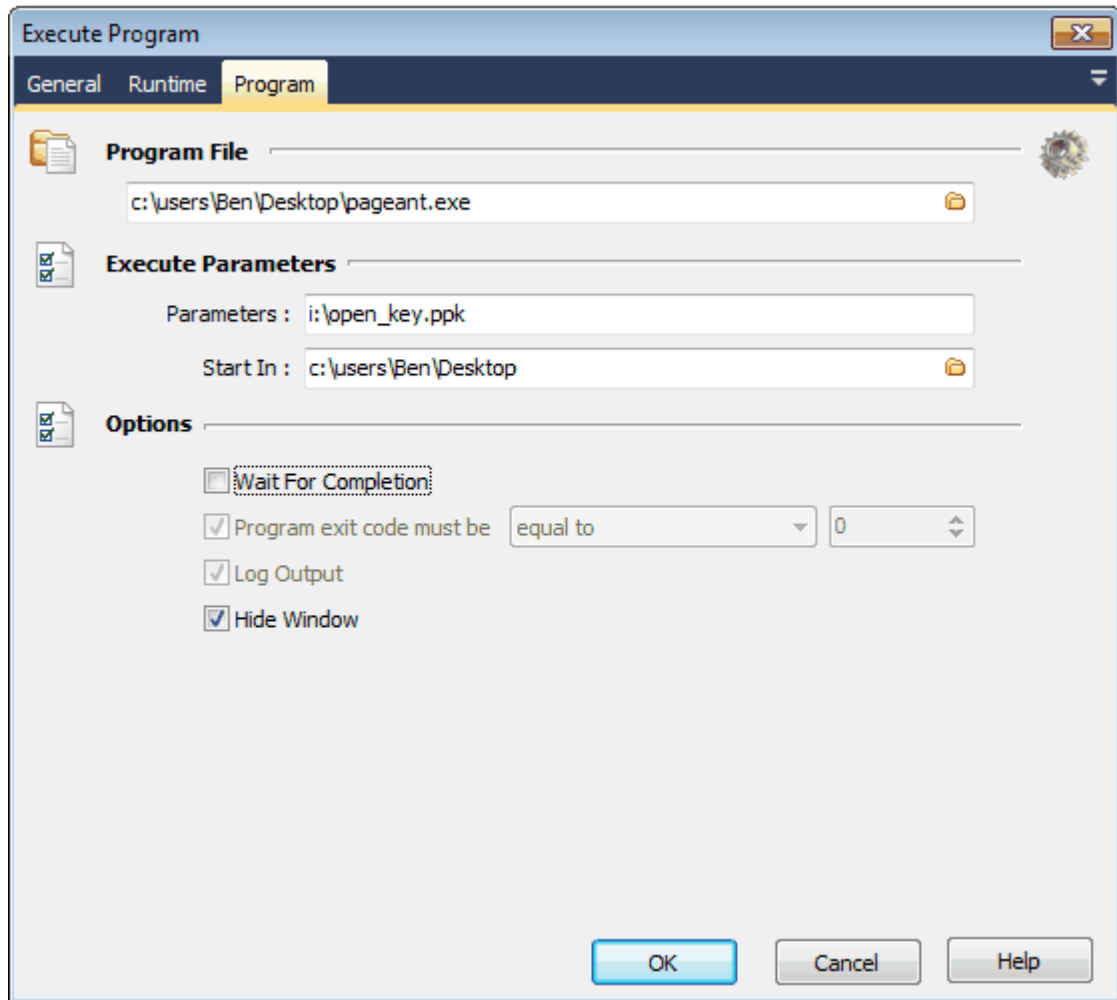
There is an additional complication if you want to run your builds from FinalBuilder Server. Pageant (the ssh key cache) requires you to interactively enter the password for your private key file, if you chose to protect it with one. As a result, if you want to use SSH in an unattended build, you need to use a private key file that is not password protected. Depending on your needs there's two ways to implement this.

If you want to use the same SSH key every time you use Mercurial the easiest thing to do is to modify Mercurial.ini to pass the location of the key as a parameter to your SSH client. For example:

```
[ui]
....
ssh = C:\Users\ben\Downloads\plink.exe -ssh -i i:\open_key.ppk
```

Now every time Mercurial calls out to plink it will use the specified key. This is the simplest approach and is recommended if at all possible.

If you want to use different keys in different circumstances you should make use of Pageant. To do this, at the start of your FinalBuilder script create an Execute Program action that launches Pageant, passing the location of the key file as a parameter:

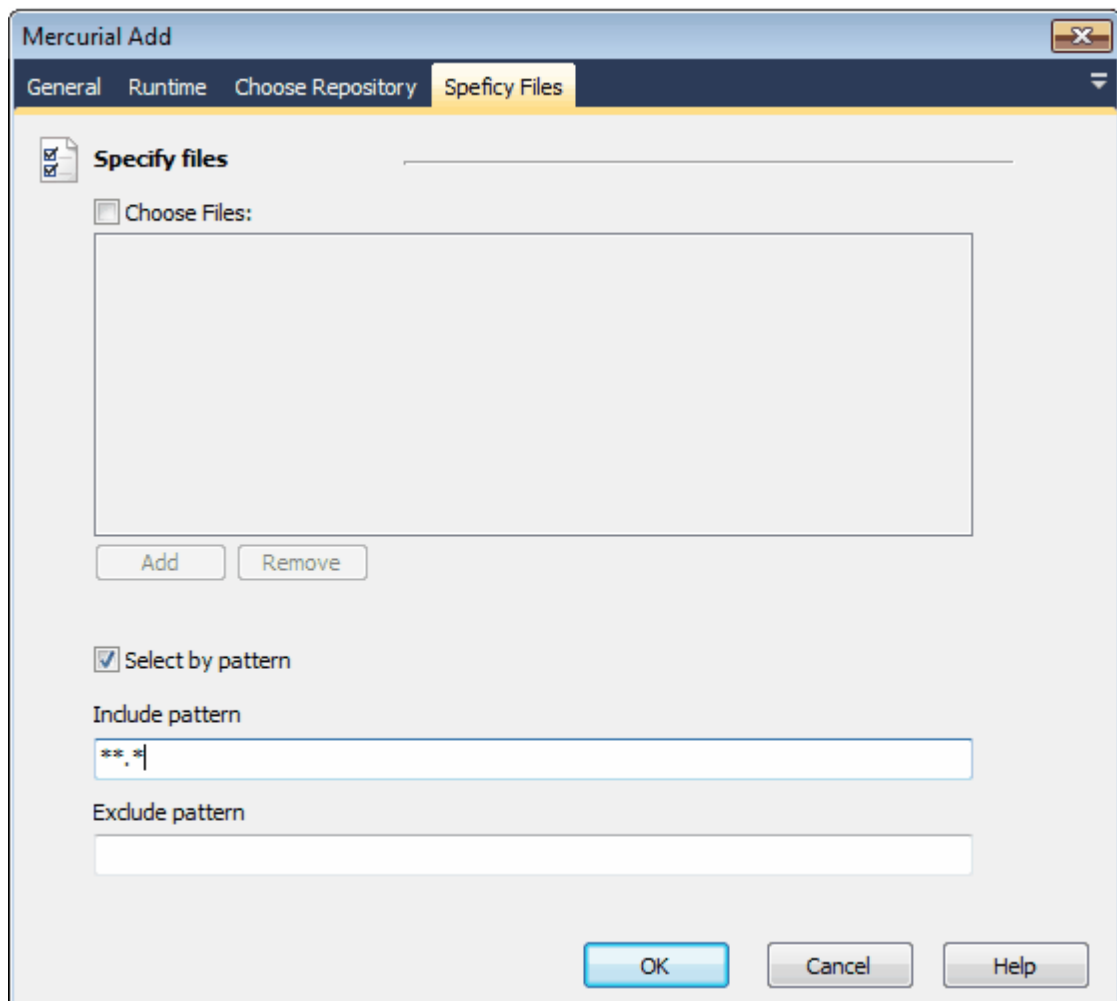


Be sure to deselect 'wait for completion'. If Pageant is already running it will add the specified key to the cache. If not, it will launch and add the key.

Recommended practice is to generate a key pair specifically for the build server/Mercurial server interaction and to not use the key for any other purpose.

6.34.7.2 Specifying Files

On the [Specify Files] tab you can pick specific files to use or you can use a pattern:



If no files or patterns are specified, all files in the working directory will be added. For more information on the supported patterns, see File Name Patterns.

6.34.7.3 Mercurial Add

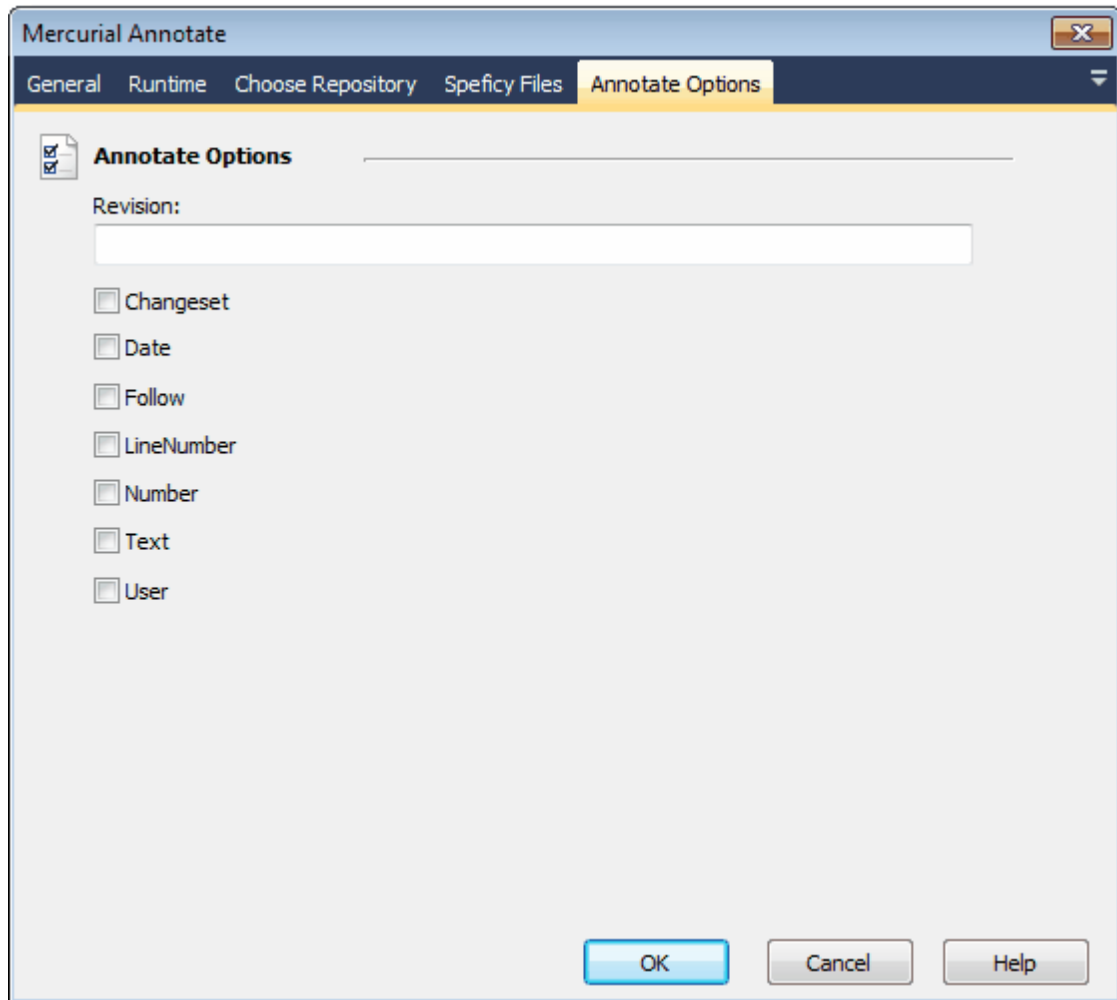
The Mercurial Add action adds the files you choose to the repository specified.

See Choosing Repositories for information on how to choose the repository to add files to.

See Specifying Files for information on how to select which files are added to the repository.

6.34.7.4 Mercurial Annotate

The Mercurial Annotate action lists changes in specified files, showing the revision id responsible for each line. It is useful for discovering when a change was made, and by whom.

**Revision**

List changes for the specified revision.

Changeset

List the changesets.

Date

List the date.

Follow

Follow file copies and renames.

LineNumber

Show the line number at the first appearance.

Number

List the revision number.

Text

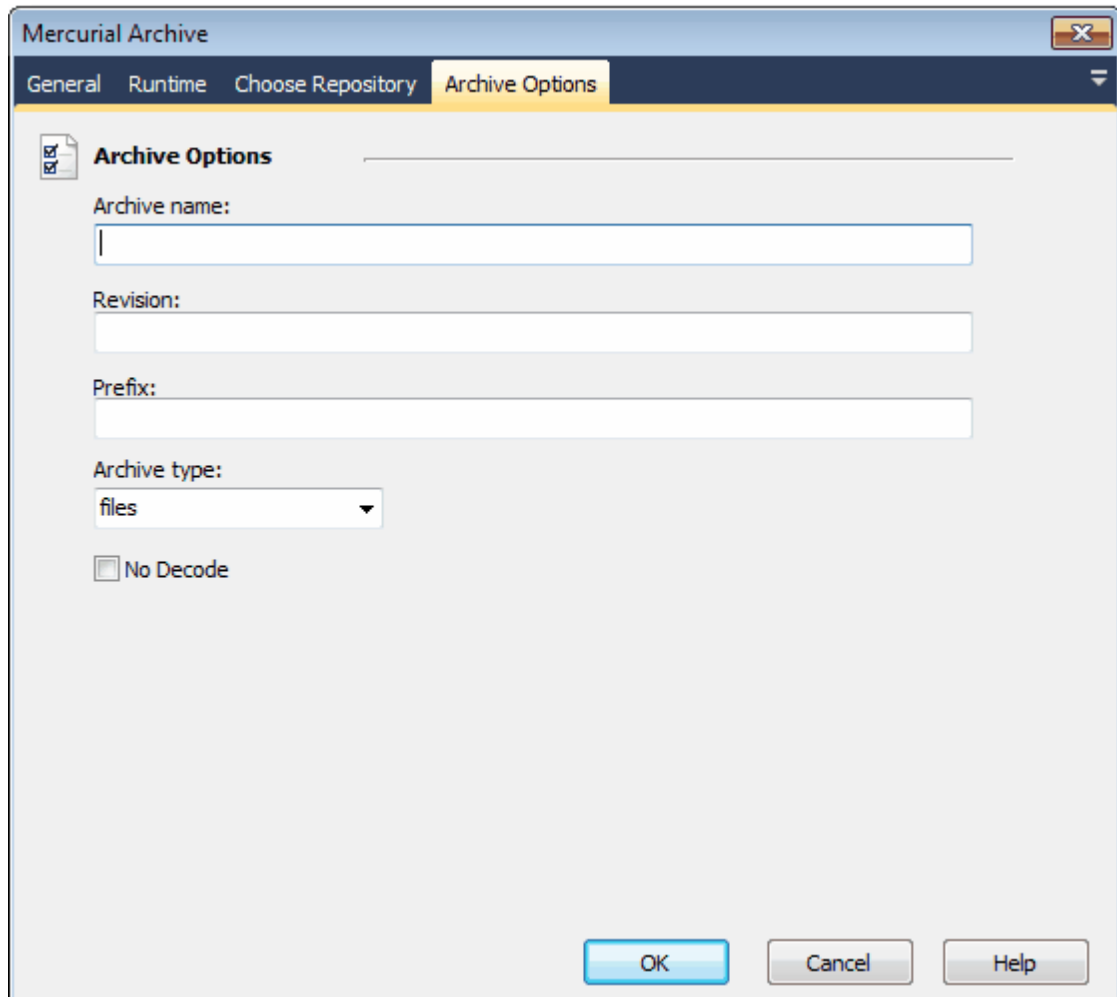
Treat all files as text.

User

List the author.

6.34.7.5 Mercurial Archive

The Mercurial Archive action creates an unversioned copy of the specified repository.



The screenshot shows the 'Mercurial Archive' dialog box with the 'Archive Options' tab selected. The dialog has a title bar with a close button. Below the title bar is a tabbed interface with four tabs: 'General', 'Runtime', 'Choose Repository', and 'Archive Options'. The 'Archive Options' tab is active and contains the following fields and controls:

- Archive name:** A text input field.
- Revision:** A text input field.
- Prefix:** A text input field.
- Archive type:** A dropdown menu currently showing 'files'.
- No Decode:** A checkbox that is currently unchecked.

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Archive name

Name of the archive file. The archive name follows the same convention as for Export.

Revision

The revision to create the archive from.

Prefix

Directory prefix for files in the archive.

Archive type

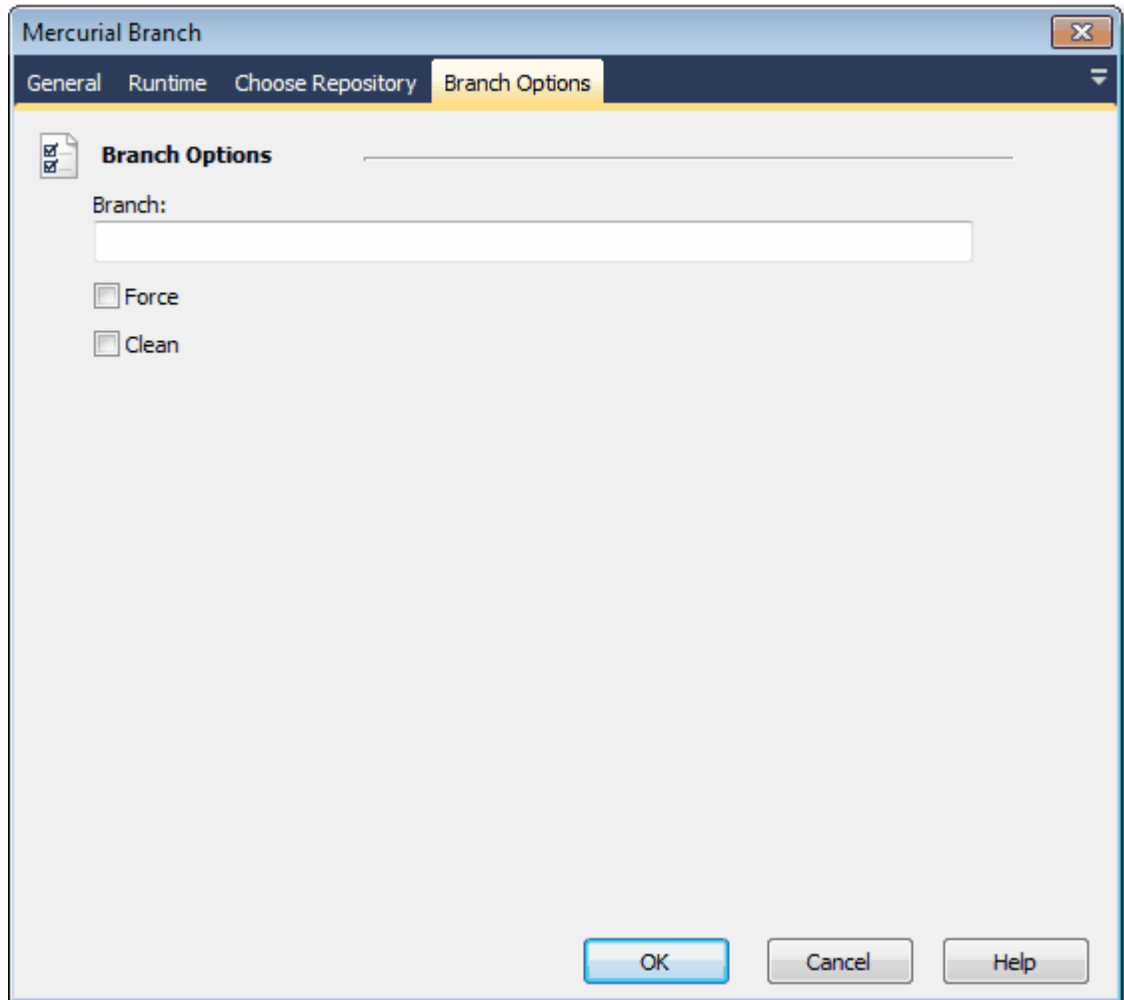
The type of archive to create.

No Decode

Do not pass files through decoders.

6.34.7.6 Mercurial Branch

The Mercurial Branch action allows you to get or set the current branch name. When no branch name is specified, the current branch name is shown. If a branch name is specified, the current branch has its name set to that specified name.

**Branch**

The name to give to the current branch.

Force

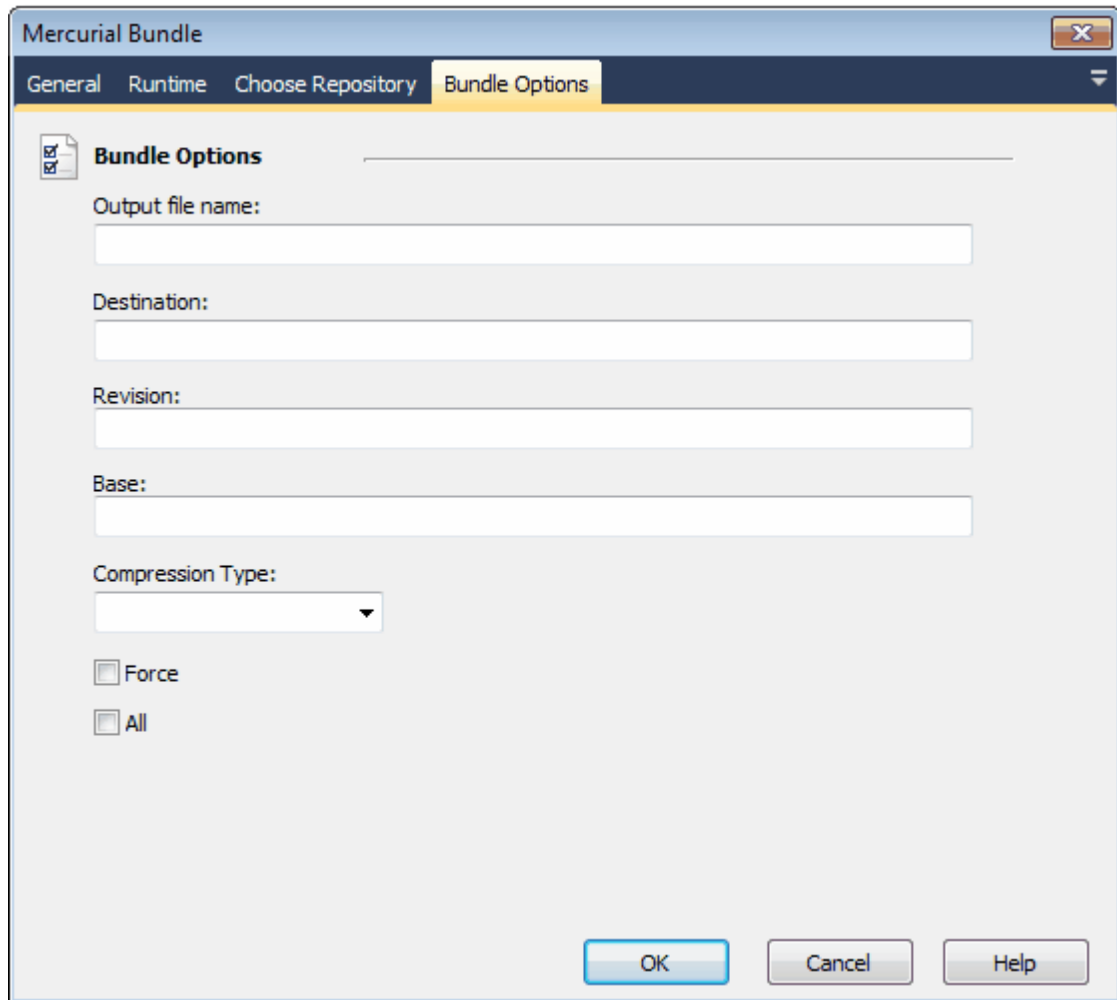
Set the branch name even if it shadows an existing branch.

Clean

Reset the branch name to the parent branch name.

6.34.7.7 Mercurial Bundle

The Mercurial Bundle action generates a compressed changegroup file which contains some or all of the changesets for the specified repository. It provides a way of moving changes between repositories when a direct pull/push is not possible.

**Output file name**

The name of the created bundle file.

Destination

The planned destination repository. If none is specified, the destination is assumed to have all the nodes specified by the Base parameter.

Revision

The revision up to which you would like to bundle.

Base

A base changeset to specify instead of a destination.

Compression Type

Type of compression to use on the bundle file.

Force

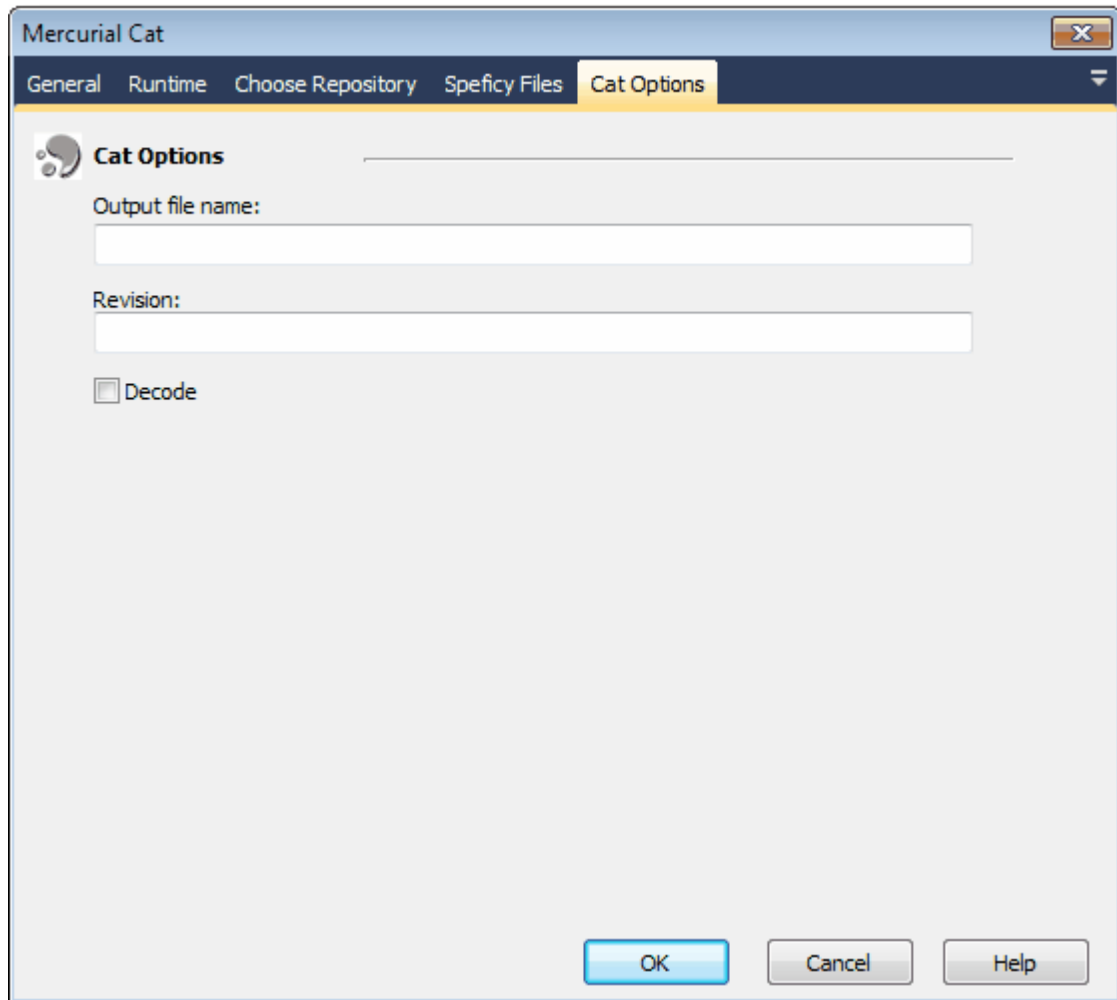
Run even when the destination repository is unrelated.

All

Bundle all changesets in the repository.

6.34.7.8 Mercurial Cat

The Mercurial Cat action prints the contents of the files specified.

**Output file name**

Output may be to file, in which case the output file name follows the same convention as for Export.

Revision

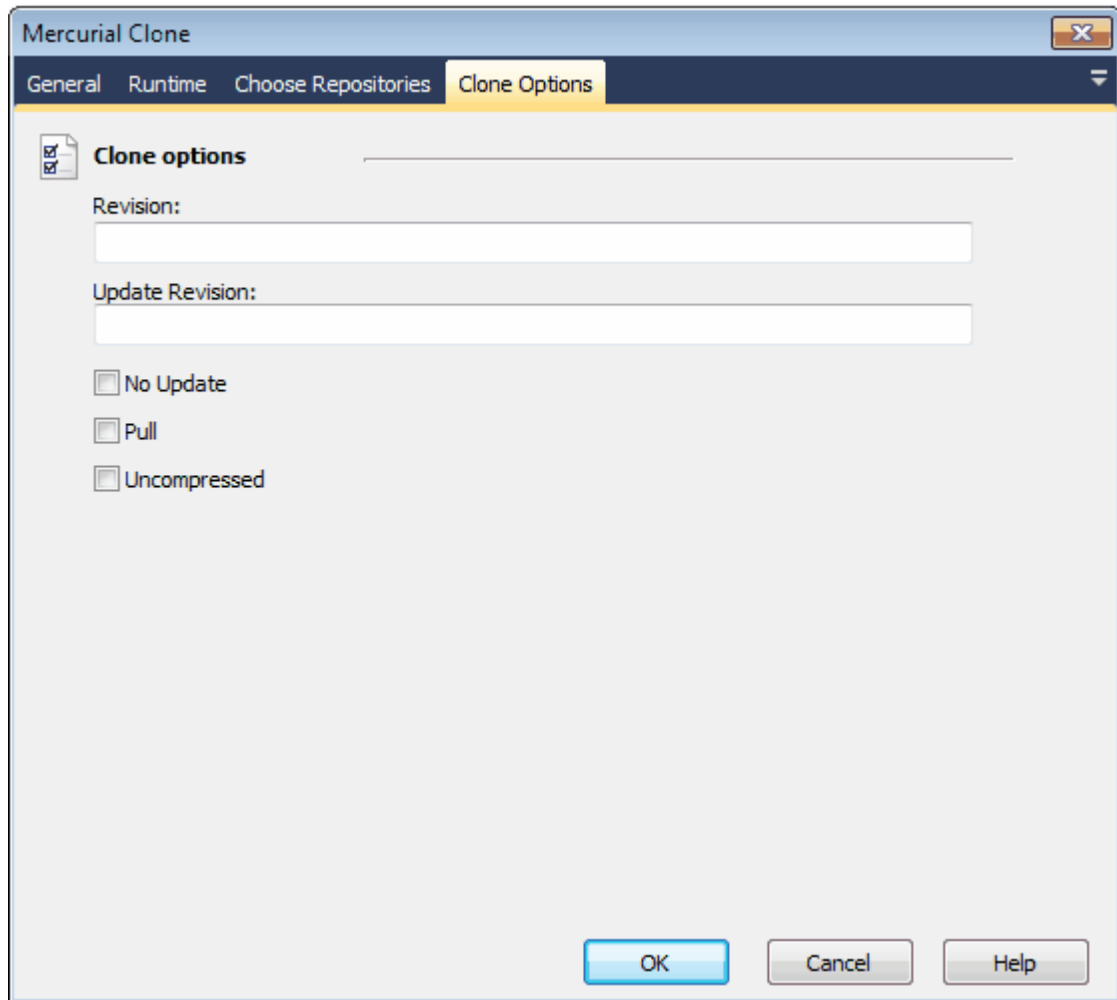
Print file contents as they were at the given revision.

Decode

Apply any matching decode filter.

6.34.7.9 Mercurial Clone

The Mercurial Clone action creates a copy of an existing repository in a new directory. The source repository can be local or network-based. The target repository should be a local directory. The target directory does not need to exist.

**Revision**

Clone on the specified revisions and ancestors.

Update Revision

Specify a revision in the current repository to tag, branch or check out

No Update

Does not populate the working directory. That is, only the repository metadata will be cloned.

Pull

Use the pull protocol to copy metadata.

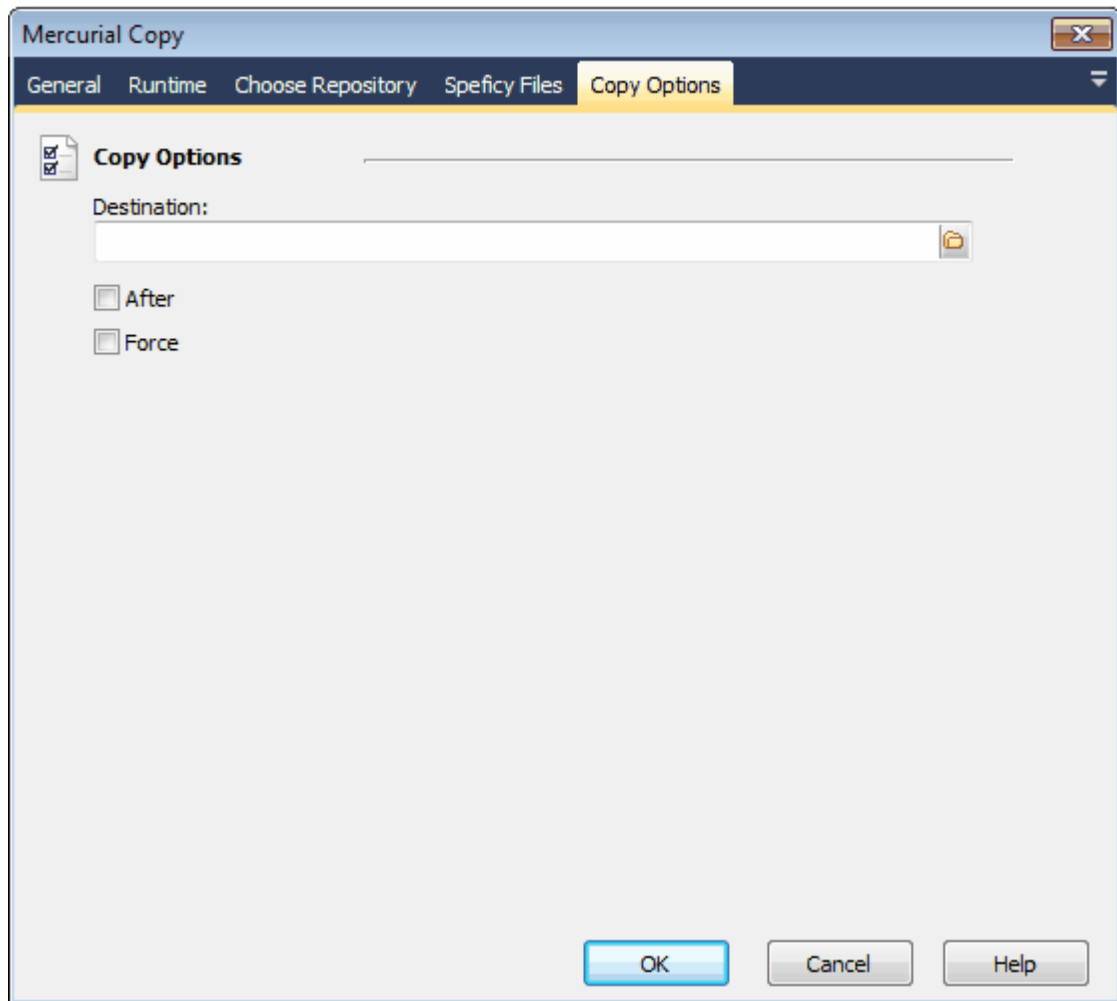
Uncompressed

Use uncompressed transfer. This is recommended when the source is local or on a LAN.

See also: Choosing Repositories

6.34.7.10 Mercurial Commit

The Mercurial Commit action commits changes to the specified files into the repository



Commit Message

The comment associated with the commit. If a commit message is not specified you will be interactively prompted to enter one when the action runs.

Log file location

You may store the commit message in a file. If you choose to do this, Log File Location should point to this file.

User

Set the specified user as the committer.

Date

Set the specified date as the commit date. See Date Formats for information on supported formats.

AddRemove

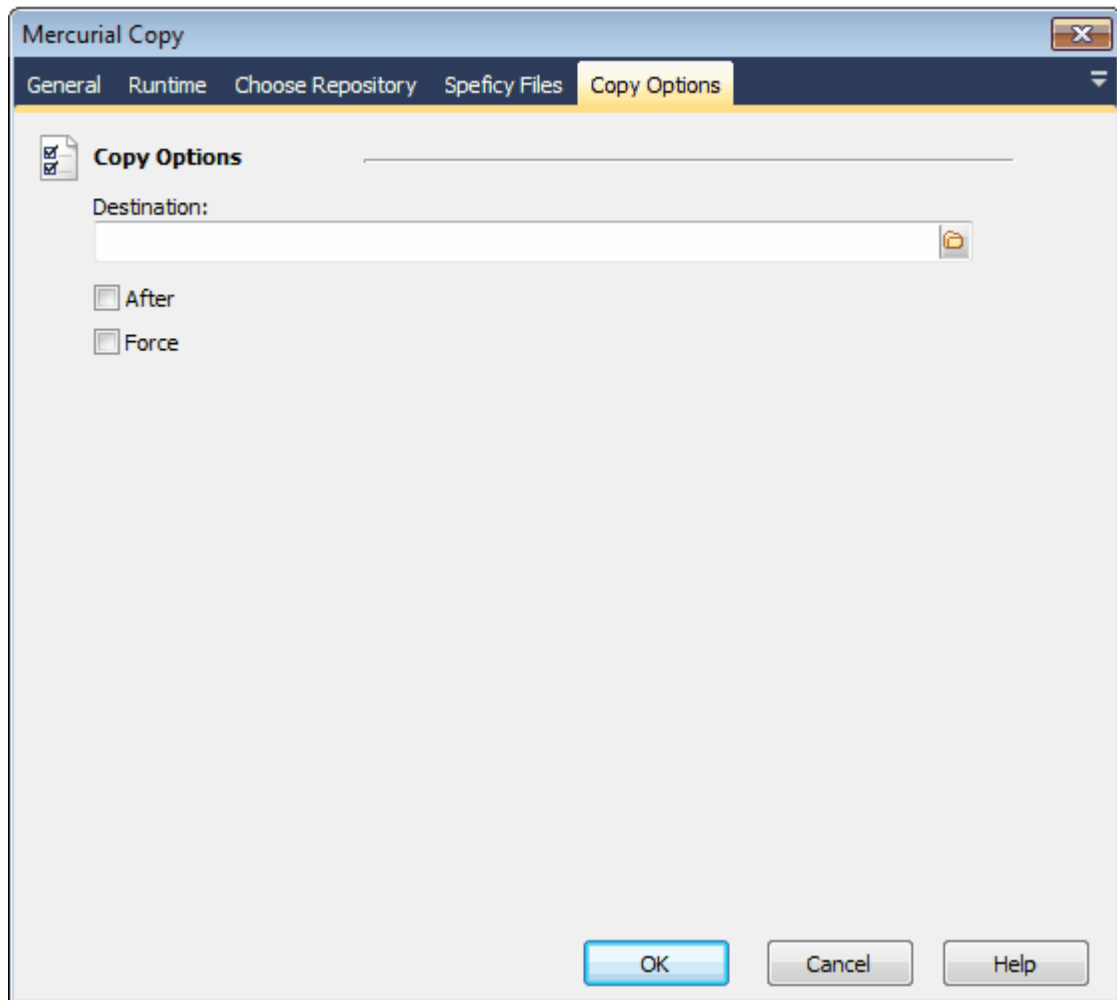
Marks new and/or missing files as added/removed before committing.

Close Branch

Marks the branch and closed, hiding it from the branch list

6.34.7.11 Mercurial Copy

The Mercurial Copy action marks the destination repository as having a copy of the specified files from the source repository. The destination may be a directory or a single file. Changes are not committed until the next explicit Commit.

**Destination**

Maybe a directory or a single file. If the destination is a file, the source must be a single file.

After

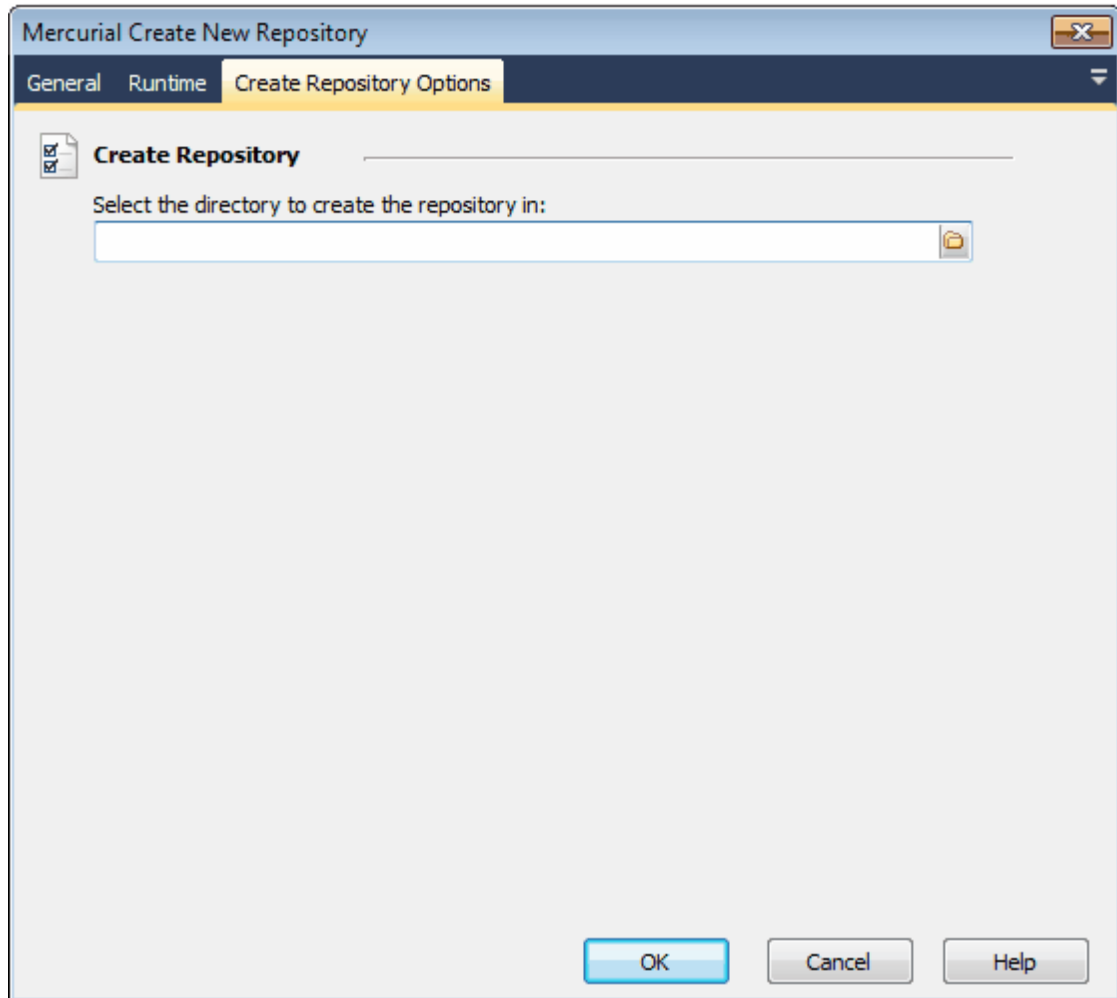
Record a copy that has already occurred.

Force

Forcibly copy over an existing managed file.

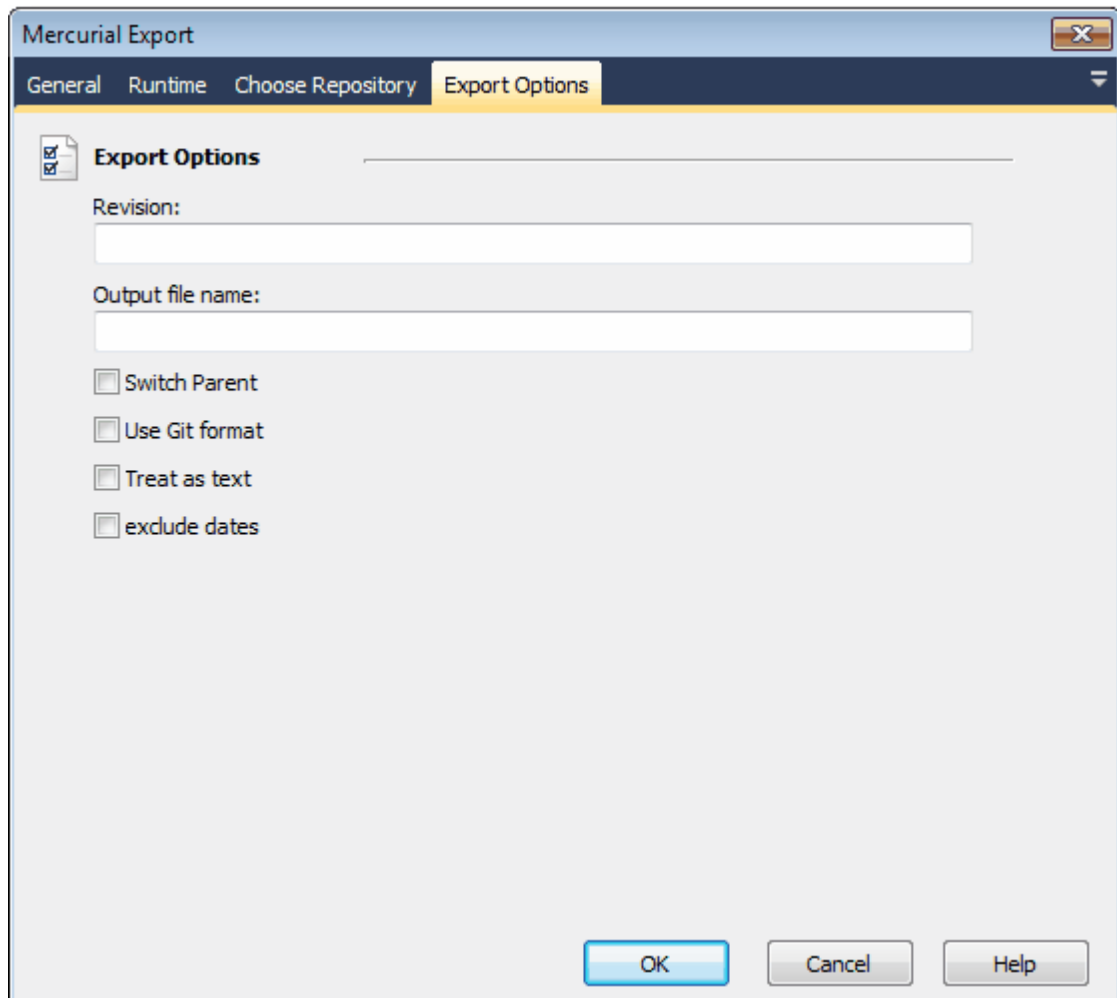
6.34.7.12 Mercurial Create Repository

The Mercurial Create New Repository action enables you create a new repository in a given local directory. The directory must exist.



6.34.7.13 Mercurial Export

The Mercurial Export action writes the header and diffs for one or more revisions to the output file specified.



Revision

The revision to export.

Output file name

The output file name is given using a format string. The formatting rules are as follows:

- %% literal "%" character
- %H changeset hash (40 bytes of hexadecimal)
- %N number of patches being generated
- %R changeset revision number
- %b basename of the exporting repository
- %h short-form changeset hash (12 bytes of hexadecimal)
- %n zero-padded sequence number, starting at 1
- %r zero-padded changeset revision number

NB: Mercurial uses % as a special character, so you cannot use FinalBuilder expressions in the output file name. You may receive warning about variables not

existing - these can be ignored

Switch Parent

Diff against the second parent.

Use Git format

Use the Git extended diff format.

Treat as text

Treat all files as text.

Exclude dates

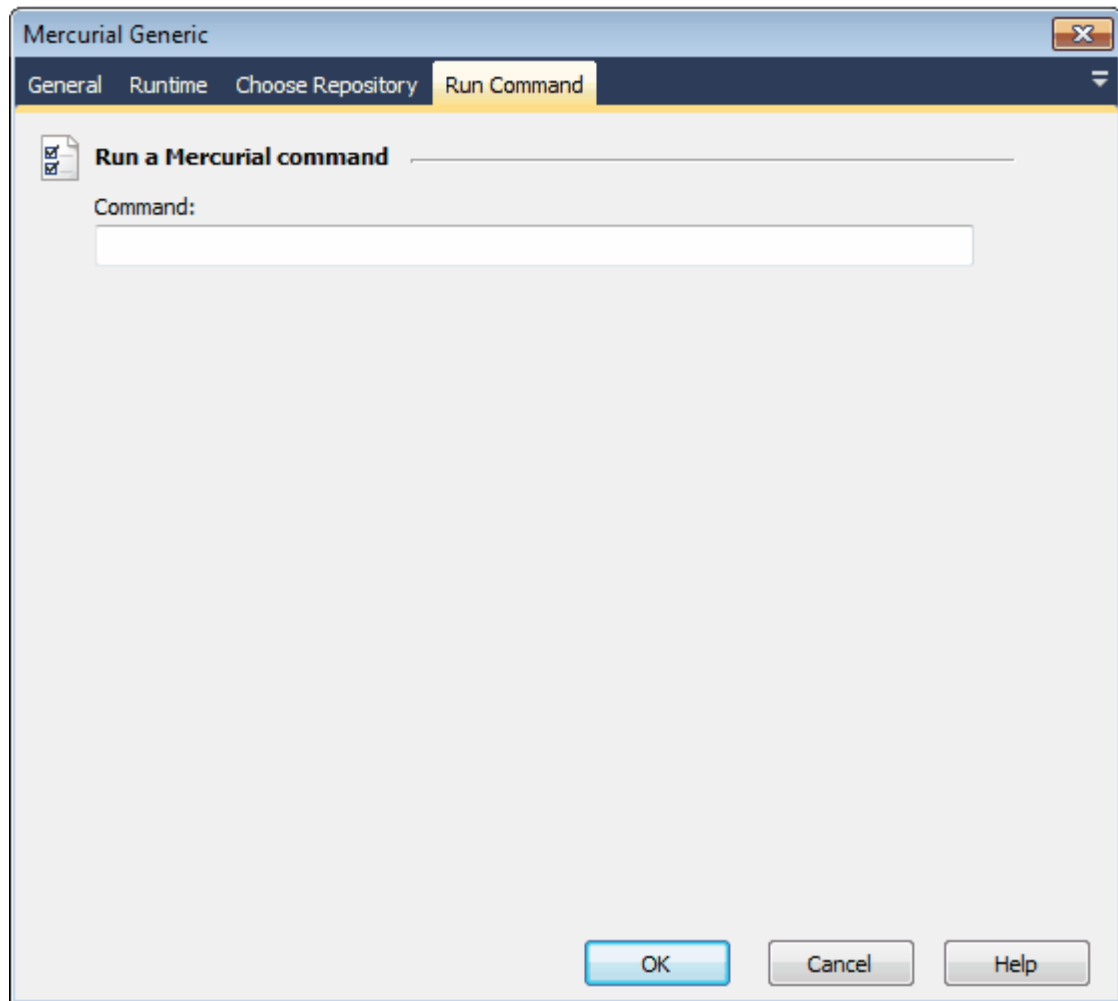
Don't include dates in diff headers.

6.34.7.14 Mercurial Forget

The Mercurial Forget action marks the specified files so they will no longer be tracked in the repository after the next commit. This only removes files from the current branch, not from the entire project history, and it does not delete them from the working directory.

6.34.7.15 Mercurial Generic

The Mercurial Generic action allows you to run any Mercurial command. The working directory is set to the repository specified.



Command

The command to send to Mercurial. FinalBuilder expressions will be expanded.

6.34.7.16 Mercurial Heads

The Mercurial Heads action prints a list of all head changesets for the specified repository. Heads are changesets with no child changesets, and are where development generally takes place.

6.34.7.17 Mercurial Import

The Mercurial Import action imports a patch file (created by Export) and commits the changes.

The screenshot shows the 'Mercurial Import' dialog box with the 'Import Options' tab selected. The dialog has a title bar with a close button. Below the title bar is a tabbed interface with four tabs: 'General', 'Runtime', 'Choose Repository', and 'Import Options'. The 'Import Options' tab is active and contains the following fields and options:

- Import Options**: A section header with a document icon and a checkmark.
- Patch file:** A text input field with a file selection icon on the right.
- Commit message:** A text input field.
- Commit file location:** A text input field with a file selection icon on the right.
- User:** A text input field.
- Date:** A text input field.
- Similarity:** A text input field.
- Force**: ☐ Force
- Exact**: ☐ Exact
- No Commit**: ☐ No Commit
- Import Branch**: ☐ Import Branch

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Patch file

File created by Export that contains the list of patches to apply.

Commit message

Comment associated with the commit. If a commit message is not specified you will be interactively prompted to enter one when the action runs.

Commit file location

To use a commit message from a file, specify the file in the Commit file location field.

User

Set the specified user as the committer.

Date

Set the commit date to the date specified. See Date Formats for more information.

Similarity

If you specify a 'similarity' between 0 and 100, Mercurial will attempt to discover file

renames and copies - the larger the number the looser match criteria.

Force

Skip check for outstanding uncommitted changes.

Exact

Apply patch to the notes from which it was generated.

No Commit

Don't commit, just update the working directory.

Import Branch

User any branch information in the patch (implied by Exact)

6.34.7.18 Mercurial Log

The Mercurial Log action prints the revision history of the specified repository.

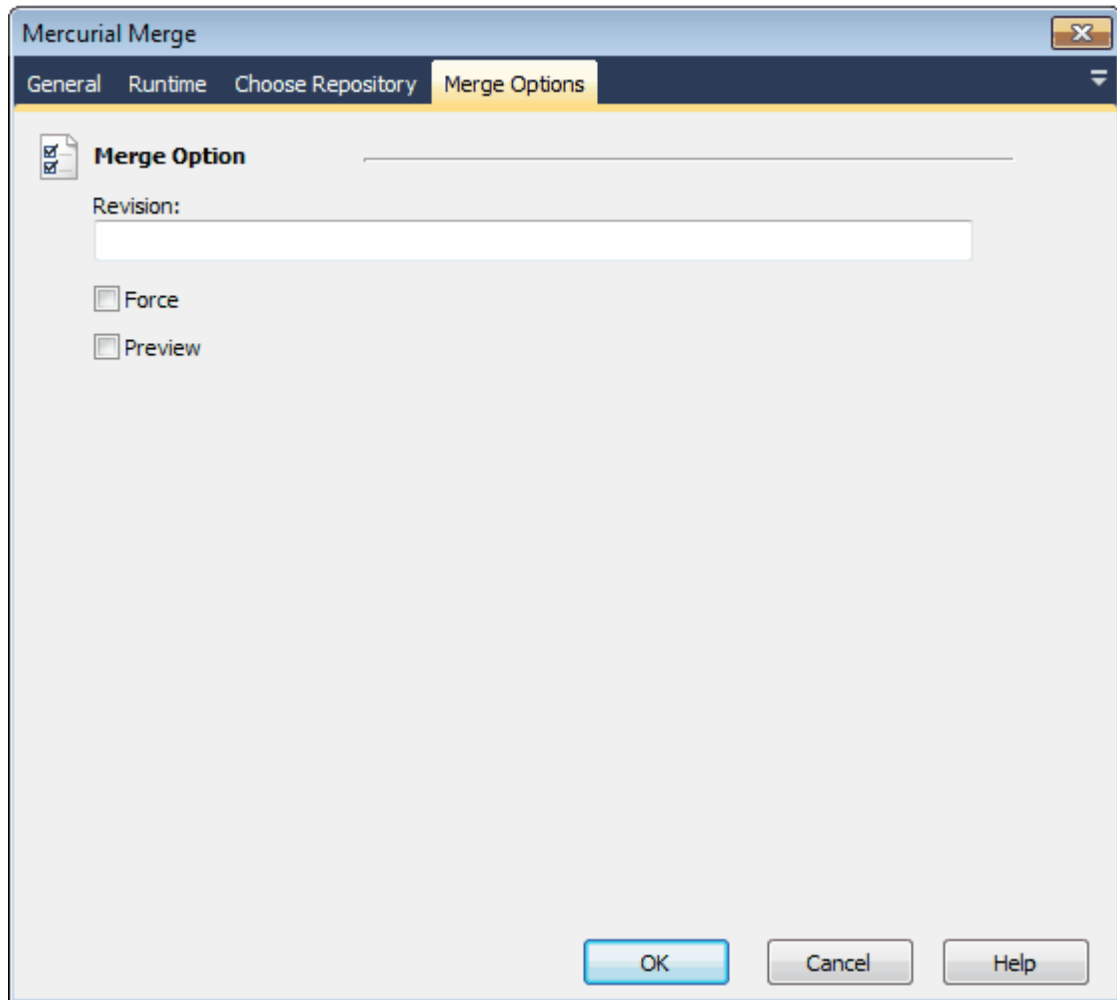
6.34.7.19 Mercurial Manifest

The Mercurial Manifest action prints a list of version controlled files for the specified repository.

6.34.7.20 Mercurial Merge

The Mercurial Merge action merges the working directory with another revision. Merge is often called after a set of changes have been pulled into the current repository.

NB The user will be prompted to interactively resolve conflicts in files that cannot be automatically merged.

**Revision**

The revision to merge. If not specified and the branch contains exactly one other head, merge will use that head. Otherwise, it needs to be specified.

Force

Merge even with outstanding (uncommitted) changes.

Preview

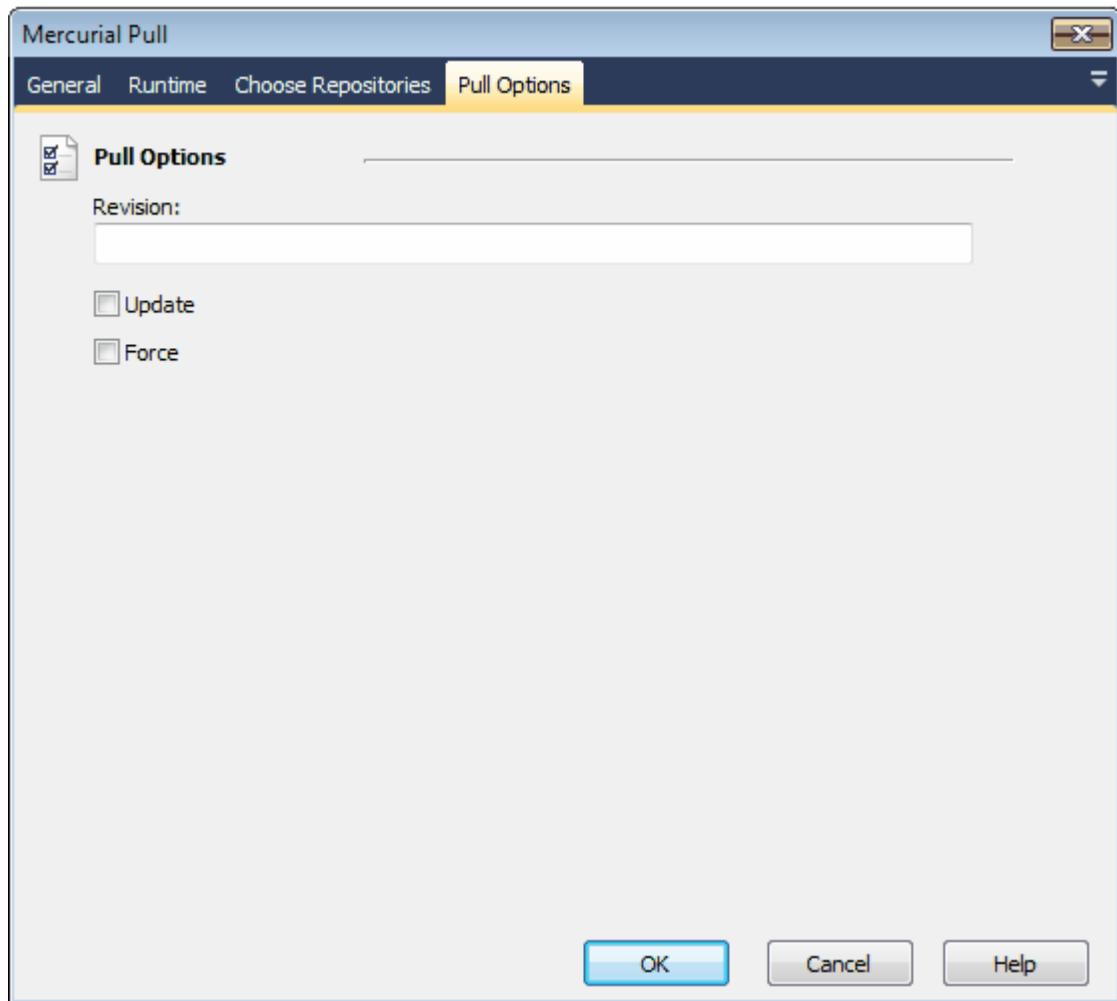
Review the revisions to merge (no merge is performed).

6.34.7.21 Mercurial Parents

The Mercurial Parents action prints the working directory's parent revisions.

6.34.7.22 Mercurial Pull

The Mercurial Pull action pulls changes from the source repository to the target repository.

**Revision**

Specify a revision in the source repository to pull from. This allows you to update the target repository to some point in the source repository's history.

Update

Run Update after the pull. If this is not selected, only the metadata (not the working directory) will be updated.

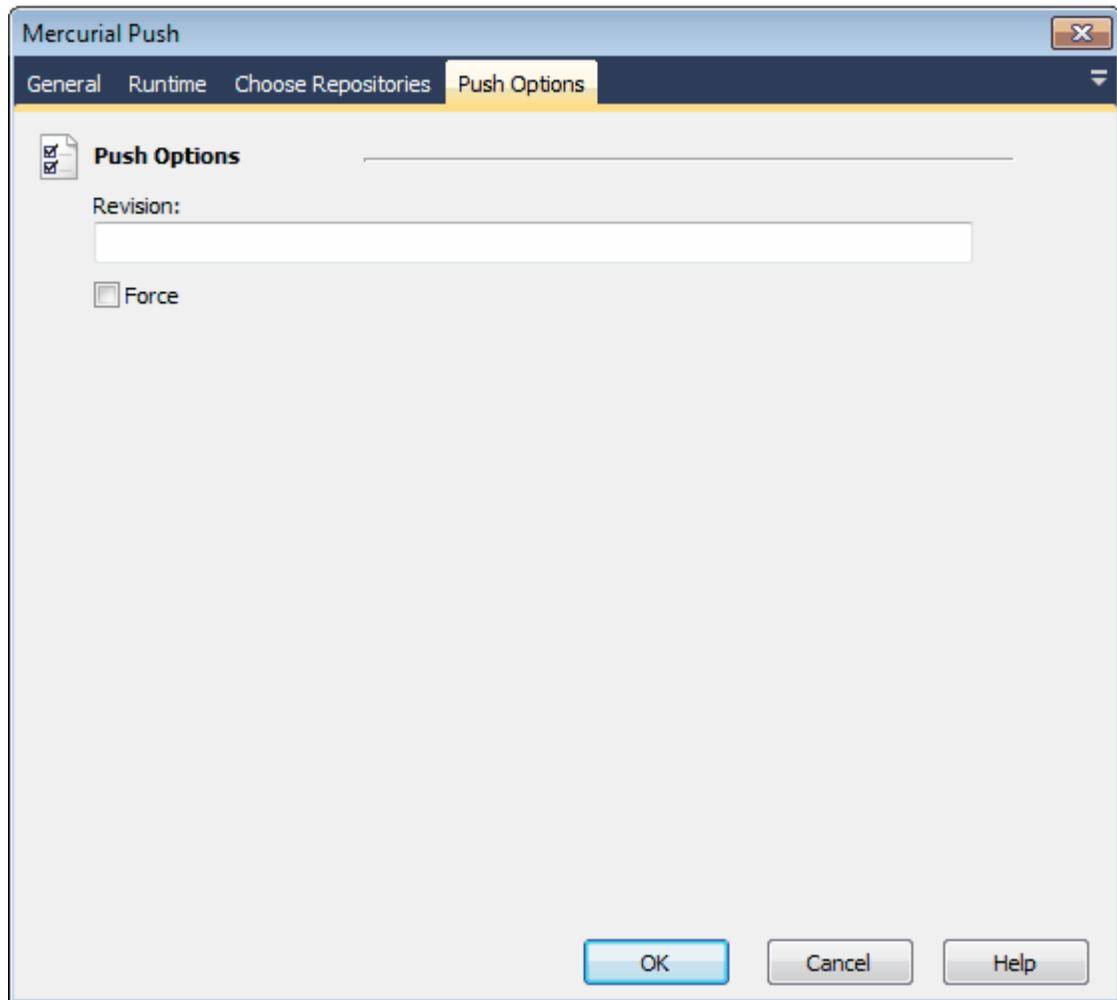
Force

Run even if the remote repository is unrelated.

6.34.7.23 Mercurial Push

The Mercurial Push action pushes changes from the source repository to the target repository. It is often used to move changes from a local to a remote repository.

Note that if you push to a local repository, you will need to run an Update on that repository. On remote repositories this is not required.

**Revision**

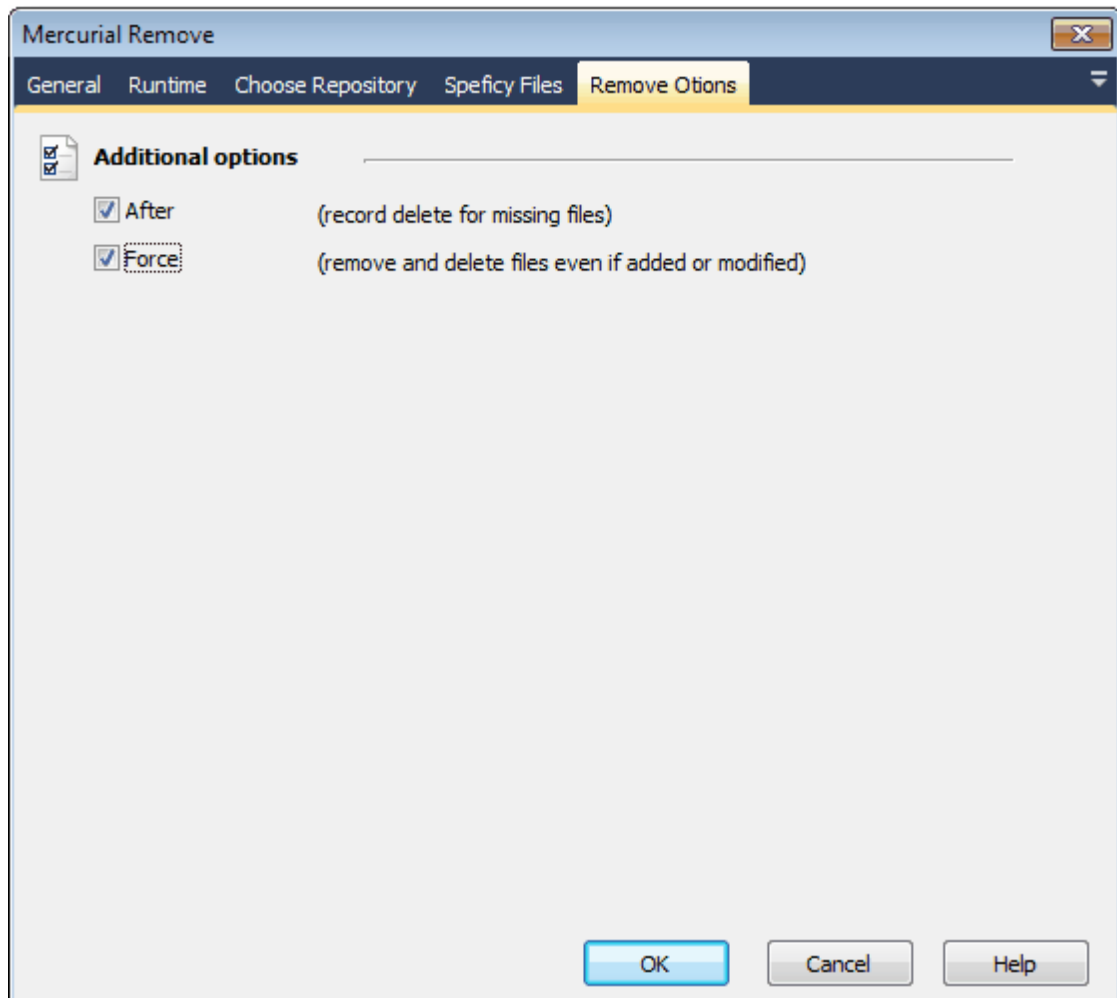
Push the specified revision.

Force

Force the push, even if the repositories are unrelated.

6.34.7.24 Mercurial Remove

The Mercurial Remove action marks the specified files for remove from the repository on the next commit.

**After**

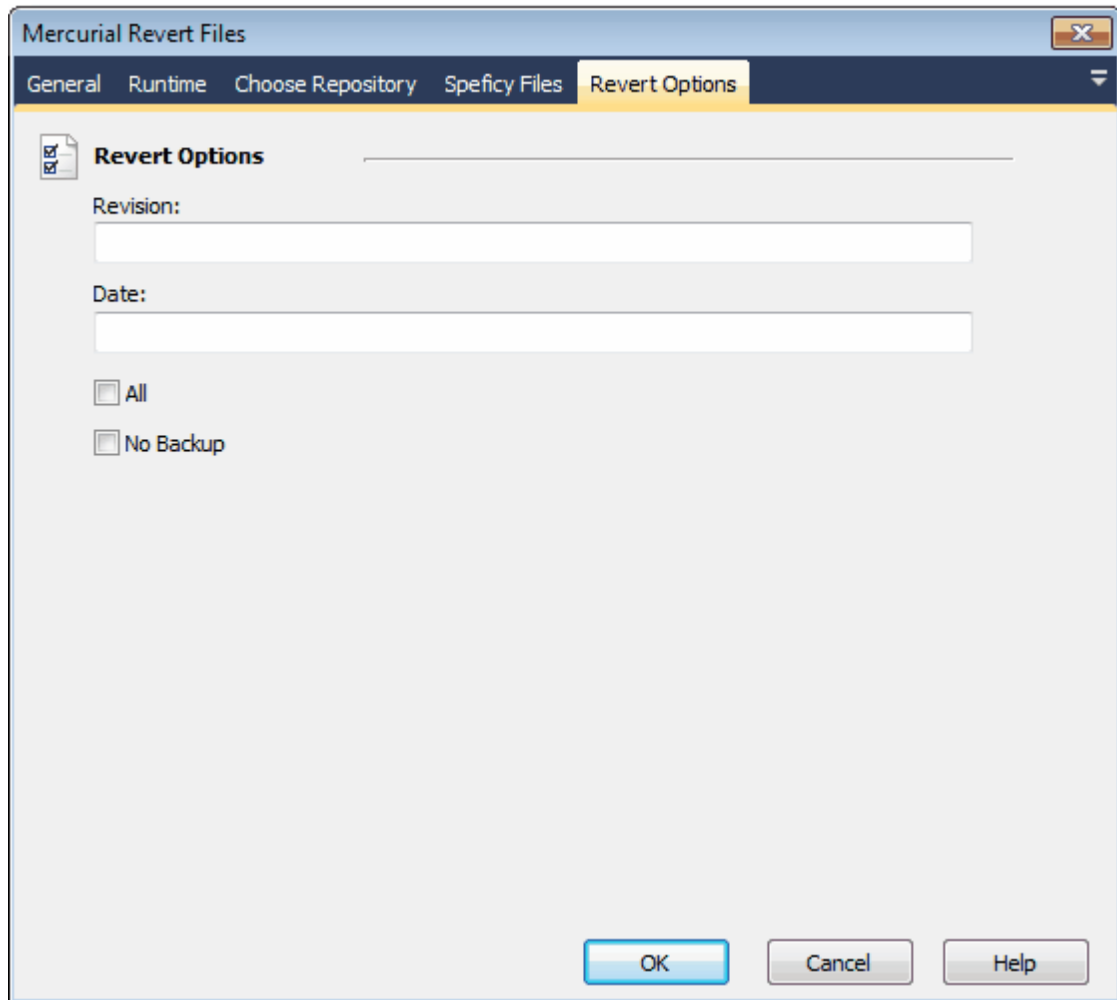
Marks files that are tracked in the repository but missing from the working directory as removed. That is, if you delete files outside of Mercurial, using the After option will also remove them from the repository.

Force

Remove and delete the file even if it is added or modified. Without this option, Remove will only work on files with no outstanding changes.

6.34.7.25 Mercurial Revert

The Mercurial Revert action reverts the specified files to the contents they had in the parent of the specified repository. Revert only works on changes that have not yet been committed.

**Revision**

Revision to revert to.

Date

Revert to the tipmost revision matching the date specified. See Date Formats for more information.

All

Revert all changes.

No Backup

Do not save backup copies of files.

6.34.7.26 Mercurial Rollback

The Mercurial Rollback action rolls back the last transaction in the repository. It should be used with extreme care - there is no way to undo a rollback.

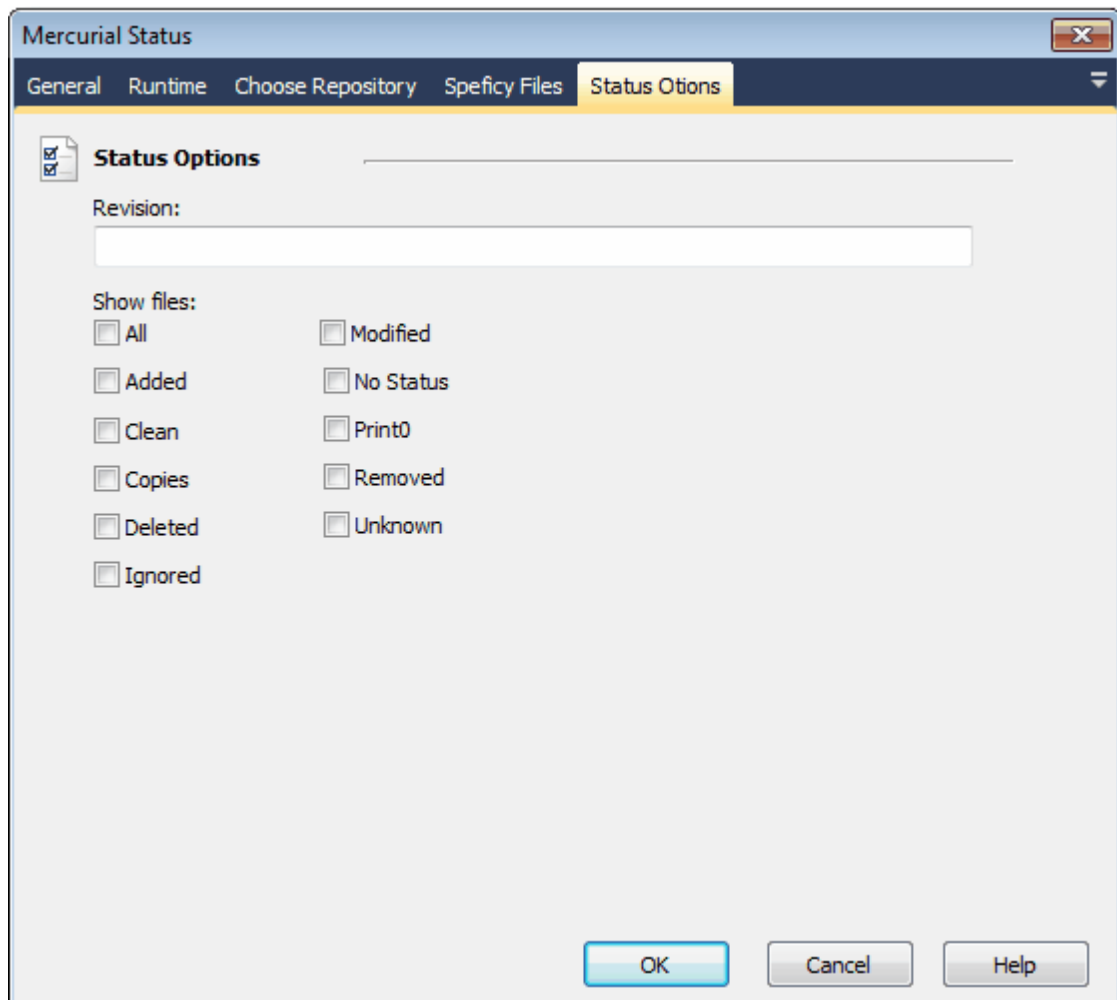
6.34.7.27 Mercurial Status

The Mercurial Status action prints the status of specified files in the repository.

The codes used to show the status of files are:

- M: modified

- A: added
- R: removed
- C: clean
- !: missing (deleted by non-hg command, but still tracked)
- ?: not tracked
- I: ignored



Revision

Use the specified revision as the base revision.

The other options are:

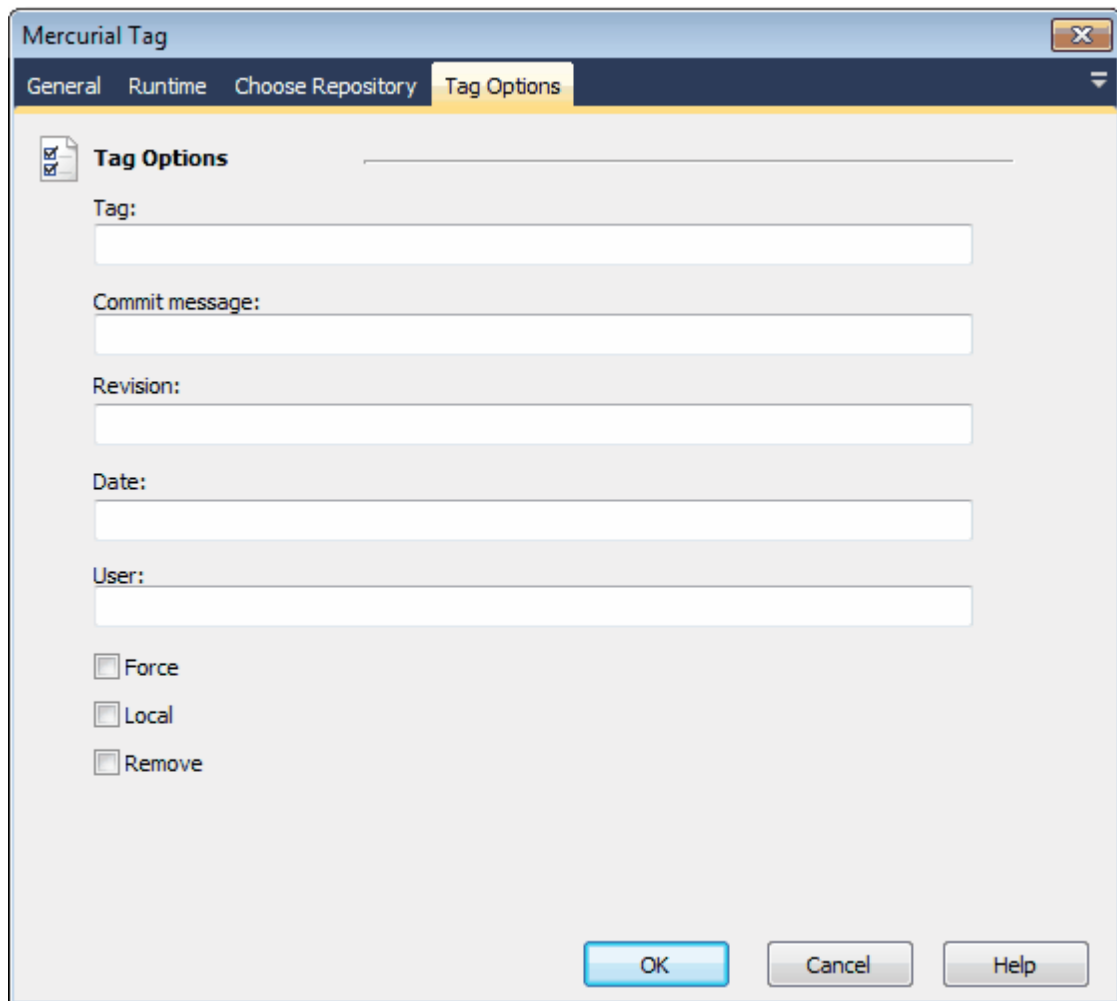
- All: show status of all files
- Modified: show only modified files
- Added: show only added files
- Removed: show only removed files
- Deleted: show only deleted (but tracked) files
- Clean: show only files without changes
- Unknown: show only unknown (not tracked) files
- Ignored: show only ignored files
- No Status: hide status prefix
- Copies: show source of copied files
- Print0: end filenames with NUL, for use with xargs

6.34.7.28 Mercurial Summary

The Mercurial Summary action prints a brief summary of the working directory state, including parents, branch, commit status and available updates.

6.34.7.29 Mercurial Tag

The Mercurial Tag action adds a tag to a particular revision in the specified repository



The screenshot shows the 'Mercurial Tag' dialog box with the 'Tag Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Choose Repository', and 'Tag Options'. The 'Tag Options' tab contains the following fields and options:

- Tag:** A text input field.
- Commit message:** A text input field.
- Revision:** A text input field.
- Date:** A text input field.
- User:** A text input field.
- Force:** A checkbox.
- Local:** A checkbox.
- Remove:** A checkbox.

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Tag

The tag to add.

Commit message

The comment to add to the commit. **NB** If a commit message is not specified you will be interactively prompted to enter one when the action runs.

Revision

The revision to tag. If none is specified, the parent of the working directory is used.

Date

Set the commit date as the date specified. See Date Formats for more information.

User

Set the specified user as the committer

Force

Replace an existing tag.

Local

Make a local tag. Local tags are stored only in the local repository and are not pushed, pulled or exported.

Remove

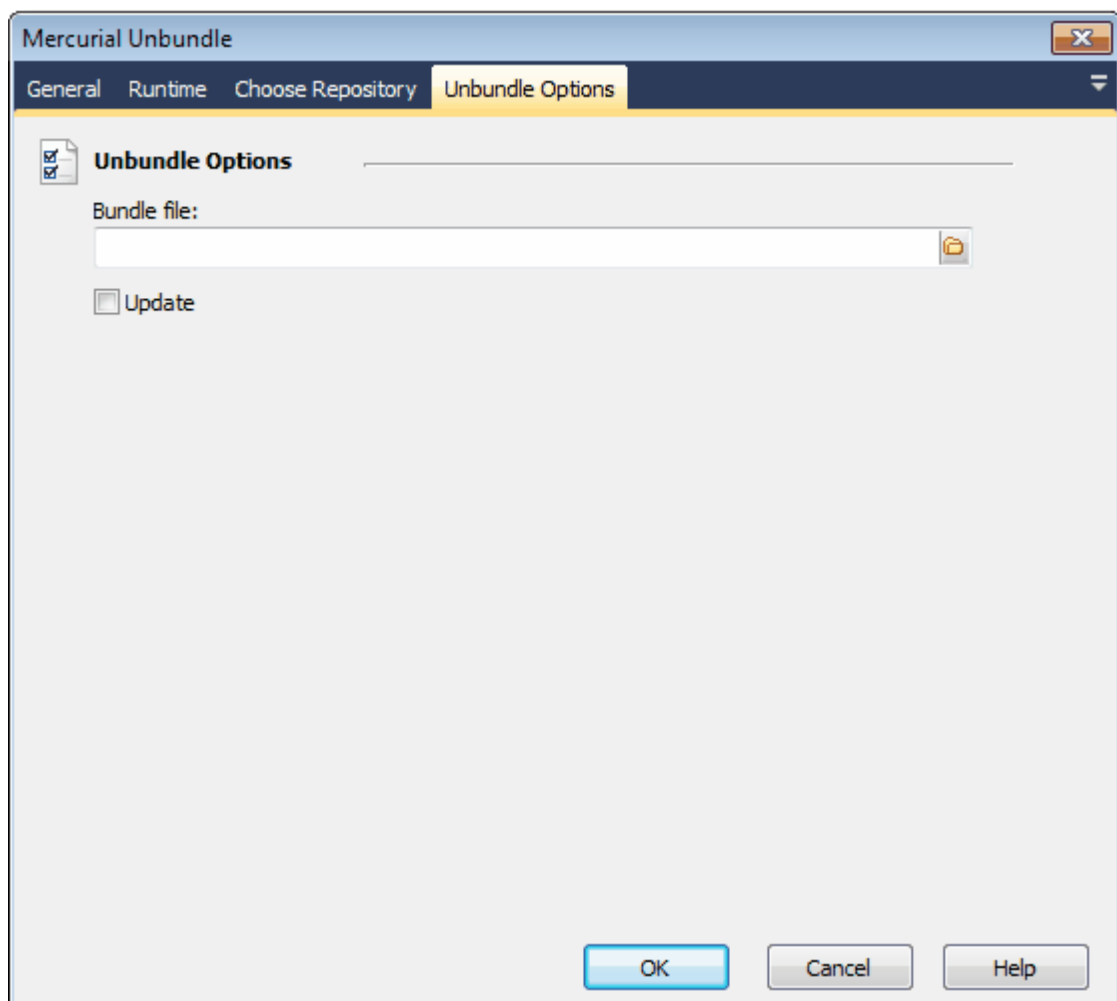
Remove the specified tag

6.34.7.30 Mercurial Tip

The Mercurial Tip action prints details about the current 'tip' revision, which is the changeset most recently added to the repository.

6.34.7.31 Mercurial Unbundle

The Mercurial Unbundle action applies the changesets contained in a bundle file to the specified repository.

**Bundle file**

The file containing the bundled changesets.

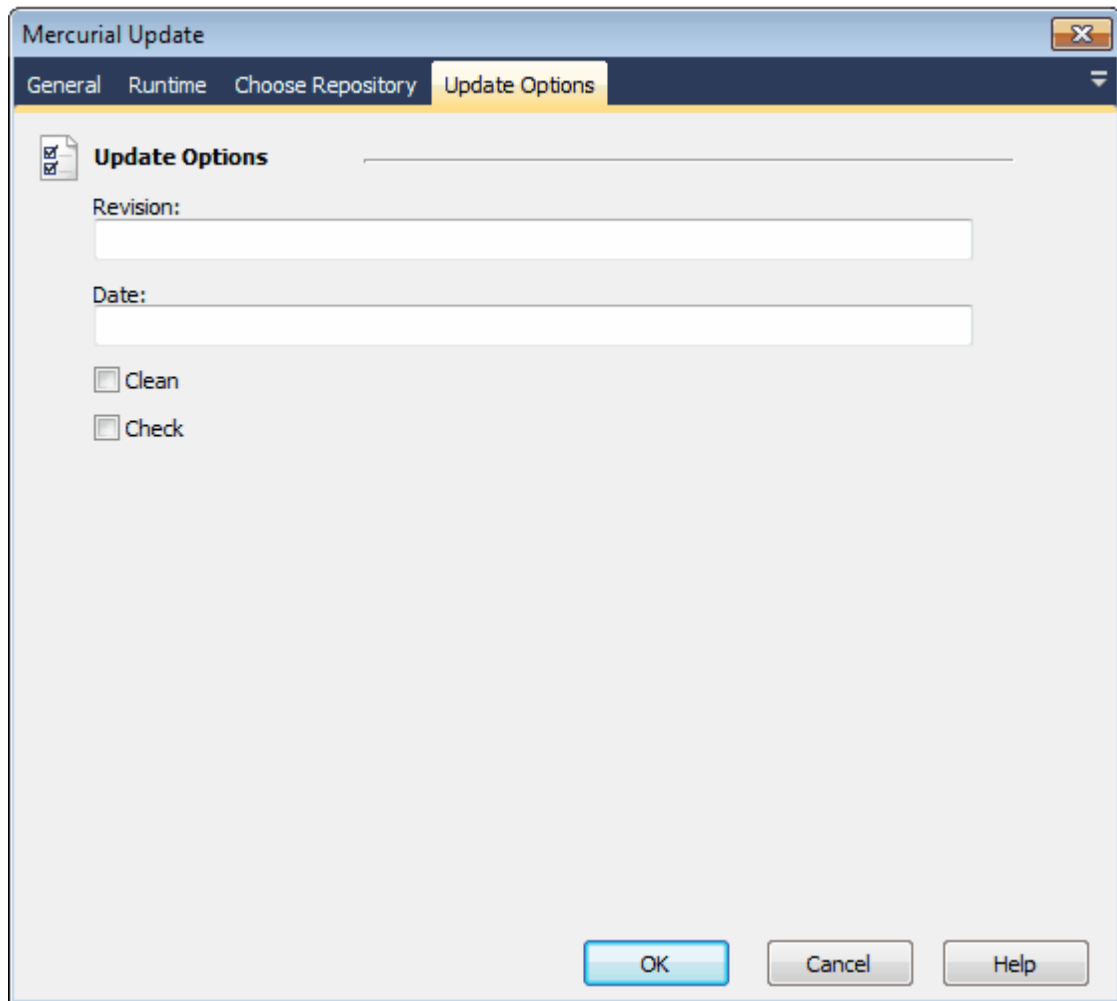
Update

Run Update after unbundling.

6.34.7.32 Mercurial Update

The Mercurial Update action updates the repository's working directory to the specified changeset. Update is usually run after actions such as Pull, which update the repository metadata but not the files in the working directory.

Update is also useful for reverting the working directory to a previous revision.

**Revision**

The revision to update to. If none is specified, the working directory is updated to the head of the current branch.

Date

Update to the tipmost revision matching the date. See Date Formats for more information.

Clean

Discard any uncommitted changes.

Check

Check for uncommitted changes.

6.34.7.33 Mercurial Verify

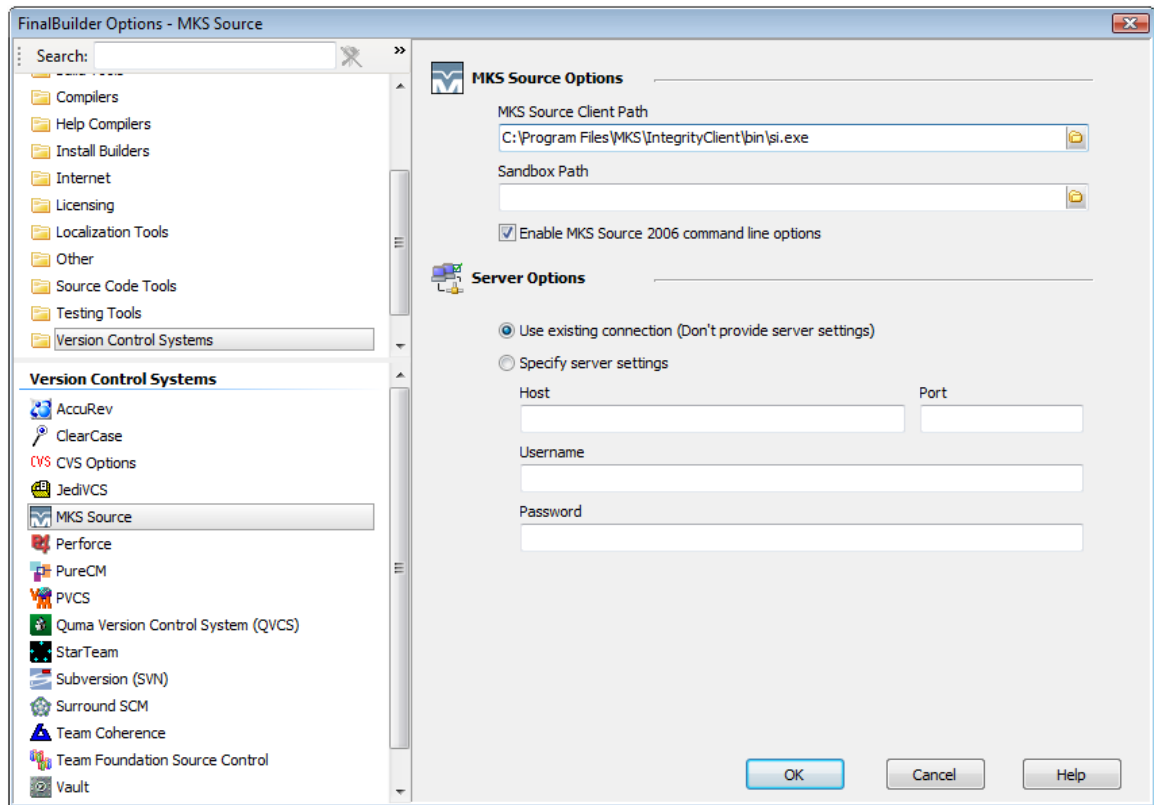
The Mercurial Verify action verifies the integrity of the specified repository.

6.34.8 MKS Source

The MKS Source actions allow interaction with a MKS Integrity Server.

For more information regarding MKS Integrity, visit www.mks.com.

Global Options



MKS Integrity Client Path

The location of the MKS Integrity command line client.

Sandbox Path

The local path to the Sandbox.

Server Options

Use Existing Connection

The actions will try to use an existing MKS Integrity connection.

Specify Server Settings

The actions will use the specified details to connect to a MKS Integrity server.

6.34.8.1 MKS Add

The MKS Add action enables you to add one or more nonmembers located in a Sandbox directory to a project.

The screenshot shows the 'MKS Add' dialog box with the 'Add Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Add Options' (which is highlighted), and 'Server Settings'. The 'Add Options' tab contains the following fields and controls:

- A green plus icon followed by the text 'Add Options'.
- A text field labeled 'Sandbox Location (Leave blank to use default Sandbox)' with a folder icon on the right.
- A text field labeled 'File or Directory' with a folder icon on the right.
- A checkbox labeled 'Recursive'.
- A dropdown menu labeled 'On Existing Archive...' with 'Cancel Operation' selected.
- A text field labeled 'Author'.
- A text field labeled 'Description'.
- A text field labeled 'Change Package ID'.
- A checkbox labeled 'Close Change Package'.
- At the bottom right, there are three buttons: 'OK', 'Cancel', and 'Help'.

Sandbox Location

The local path to the location of the sandbox.

File or Directory

Nonmembers that are to be added to the project.

On Existing Archive...

This options allows you to control the behaviour of MKS when it finds an existing archive for the nonmember.

Possible values are...

cancel - Cancel the Action

sharearchive - Share the archive.

newarchive - Create a new archive.

Change Package ID

The ID of the change package to be notified of this action.

Close Change Package

Controls whether to close the associated change package.

6.34.8.2 MKS Add Label

The MKS Add Label action enables you to assign a label to a revision of one or more members in a project.

The screenshot shows the 'MKS Add Label' dialog box. The 'Add Label Options' tab is active. The 'Sandbox Location' field is empty. The 'Member' field is empty. The 'Recursive' checkbox is unchecked. The 'Revision' field is empty. The 'Label' field is empty. The 'Allow moving between revisions' checkbox is unchecked. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

Sandbox Location

The local path to the location of the sandbox.

Member

The member to place the Label on to.

Revision

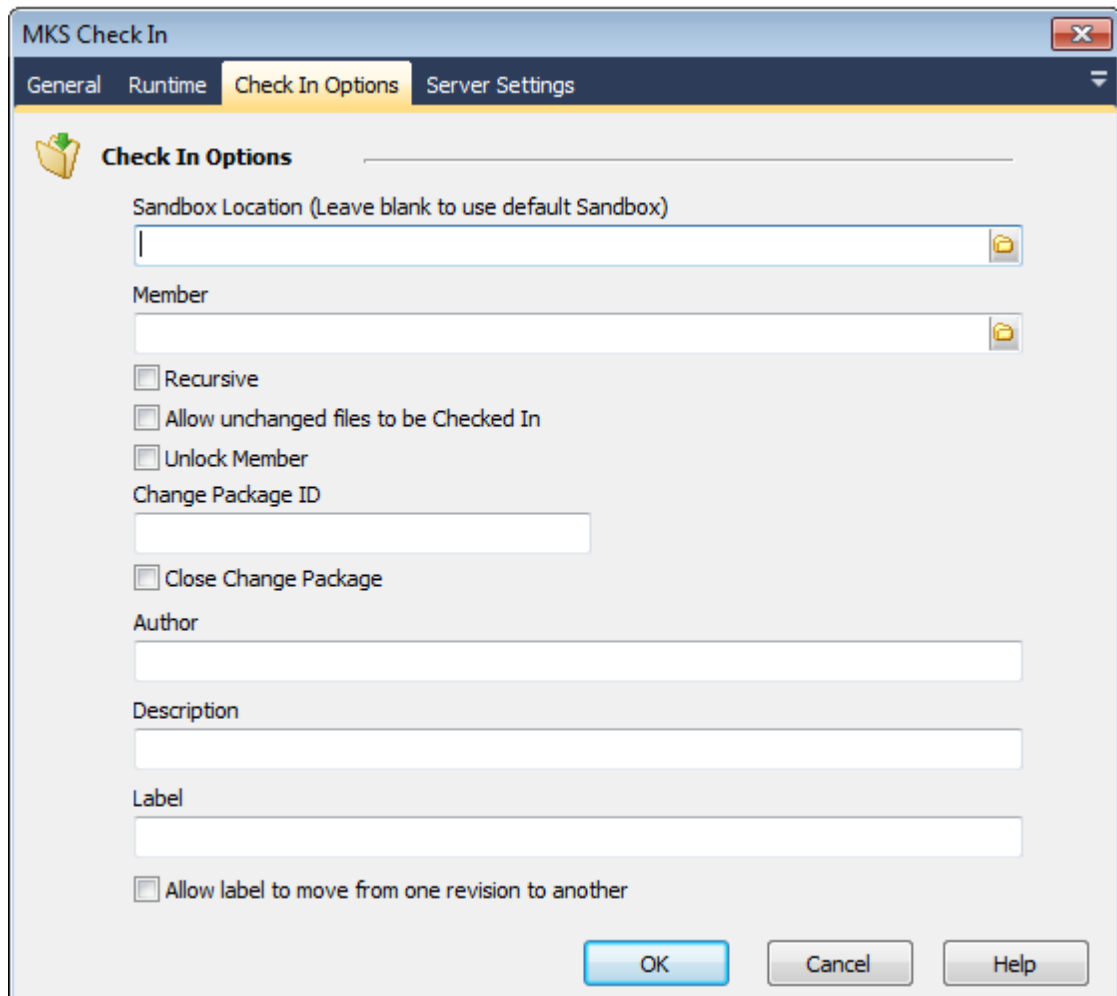
The revision of the member to add the label to.

Allow moving between revisions

Controls whether to allow the label to be moved between revisions.

6.34.8.3 MKS Check In

The MKS Check In action enables you to check in and save changed to Sandbox members.



The image shows the 'MKS Check In' dialog box with the 'Check In Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Check In Options' (active), and 'Server Settings'. The 'Check In Options' tab contains a folder icon and the title 'Check In Options'. Below this are several fields and checkboxes: 'Sandbox Location (Leave blank to use default Sandbox)' with a text box and a folder icon; 'Member' with a text box and a folder icon; three checkboxes: 'Recursive', 'Allow unchanged files to be Checked In', and 'Unlock Member'; 'Change Package ID' with a text box; a checkbox 'Close Change Package'; 'Author' with a text box; 'Description' with a text box; 'Label' with a text box; and a checkbox 'Allow label to move from one revision to another'. At the bottom right are three buttons: 'OK', 'Cancel', and 'Help'.

Sandbox Location

The local path to the location of the sandbox.

Member

The member to check in.

Unlock Member

Unlock the newly checked-in revision.

Change Package ID

The ID of the change package to be notified of this action.

Close Change Package

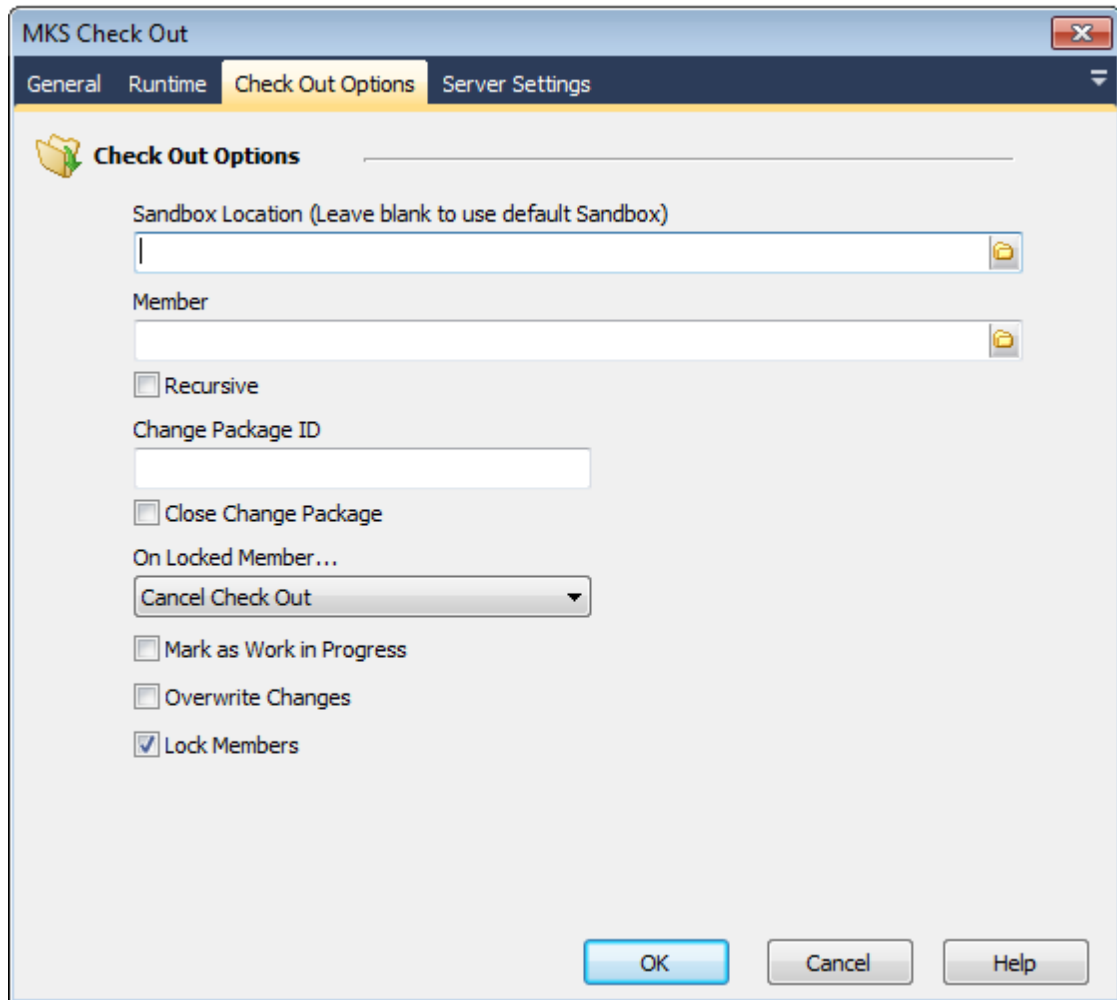
Controls whether to close the associated change package.

Label

The label to be applied to all checked-in members.

6.34.8.4 MKS Check Out

The MKS Check Out action enables you to check out members for modification.

**Sandbox Location**

The local path to the location of the sandbox.

Member

The member to check out.

Change Package ID

The ID of the change package to be notified of this action.

Close Change Package

Controls whether to close the associated change package.

On Locked Member...

This options allows you to control the behaviour of MKS when the member you are trying to check out is locked.

Possible values are...

cancel - Cancel the operation.

branch - Create a branch.

makewritable - Make the working file writable.

Overwrite Changes

Controls whether to overwrite the working file if it has changed.

Mark as Work-In-Progress

Controls whether to mark the member as Work-In-Progress.

Lock Members

Controls whether to put a lock on the members.

6.34.8.5 MKS Checkpoint

The MKS Checkpoint action enables you to create a new revision of an entire project.

The screenshot shows the 'MKS Checkpoint' dialog box with the 'Checkpoint Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Checkpoint Options' (active), and 'Server Settings'. The 'Checkpoint Options' tab contains the following fields and controls:

- Sandbox Location** (Leave blank to use default Sandbox): A text input field with a folder icon button to its right.
- ☐ **Recursive**: A checkbox.
- Author**: A text input field.
- Description**: A text input field.
- Label**: A text input field.
- ☐ **Label Members**: A checkbox.
- State**: A text input field.

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Sandbox Location

The local path to the location of the sandbox.

Label

The label to be applied to the checkpointed revision of the project.

Label Members

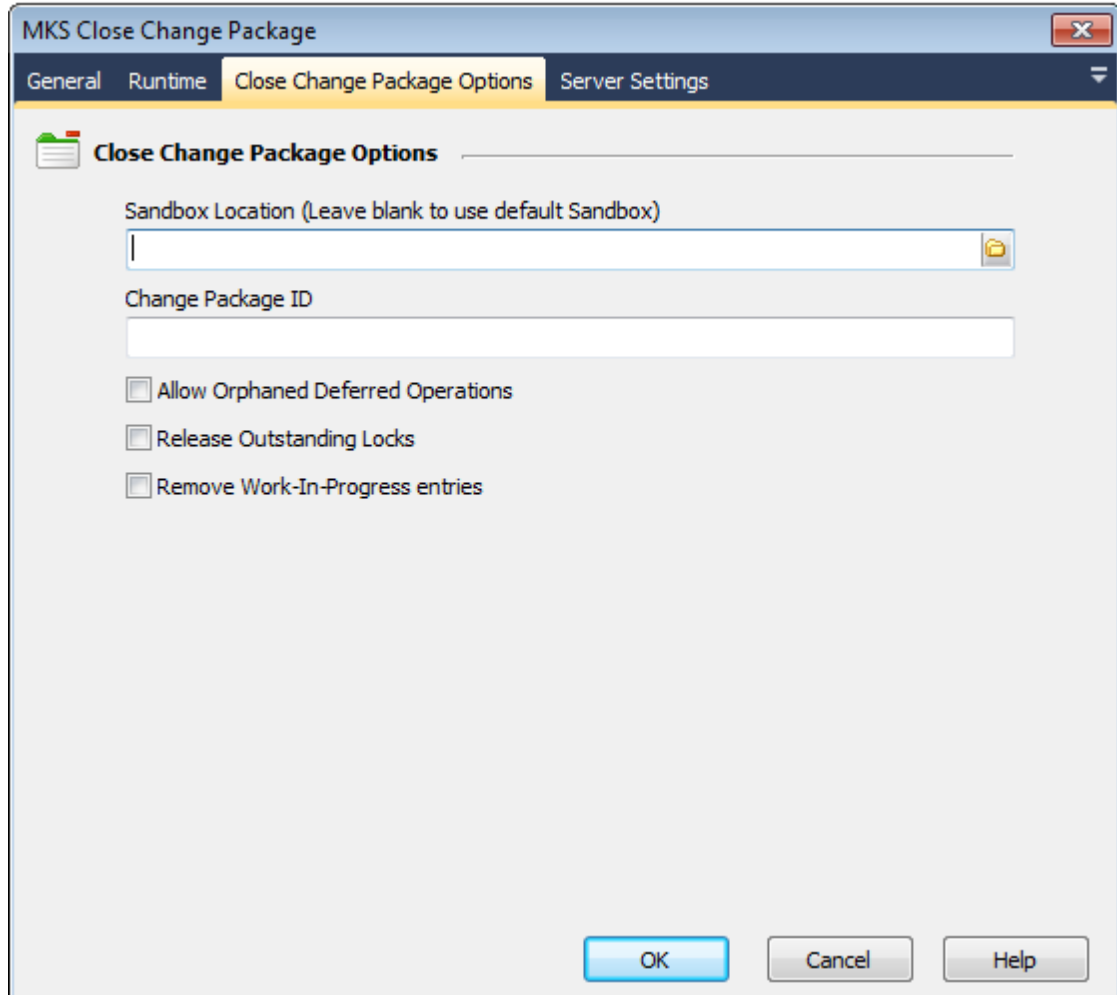
Controls whether to label the members as well as the project.

State

The state of the checkpointed revisions.

6.34.8.6 MKS Close Change Package

The MKS Close Change Package action enables you to close a change package.



Sandbox Location

The local path to the location of the sandbox.

Change Package ID

The ID of the change package that is to be closed.

Allow Orphaned Deferred Operations

Controls if orphaned deferred operations are permitted when the change package is closed.

Release Outstanding Locks

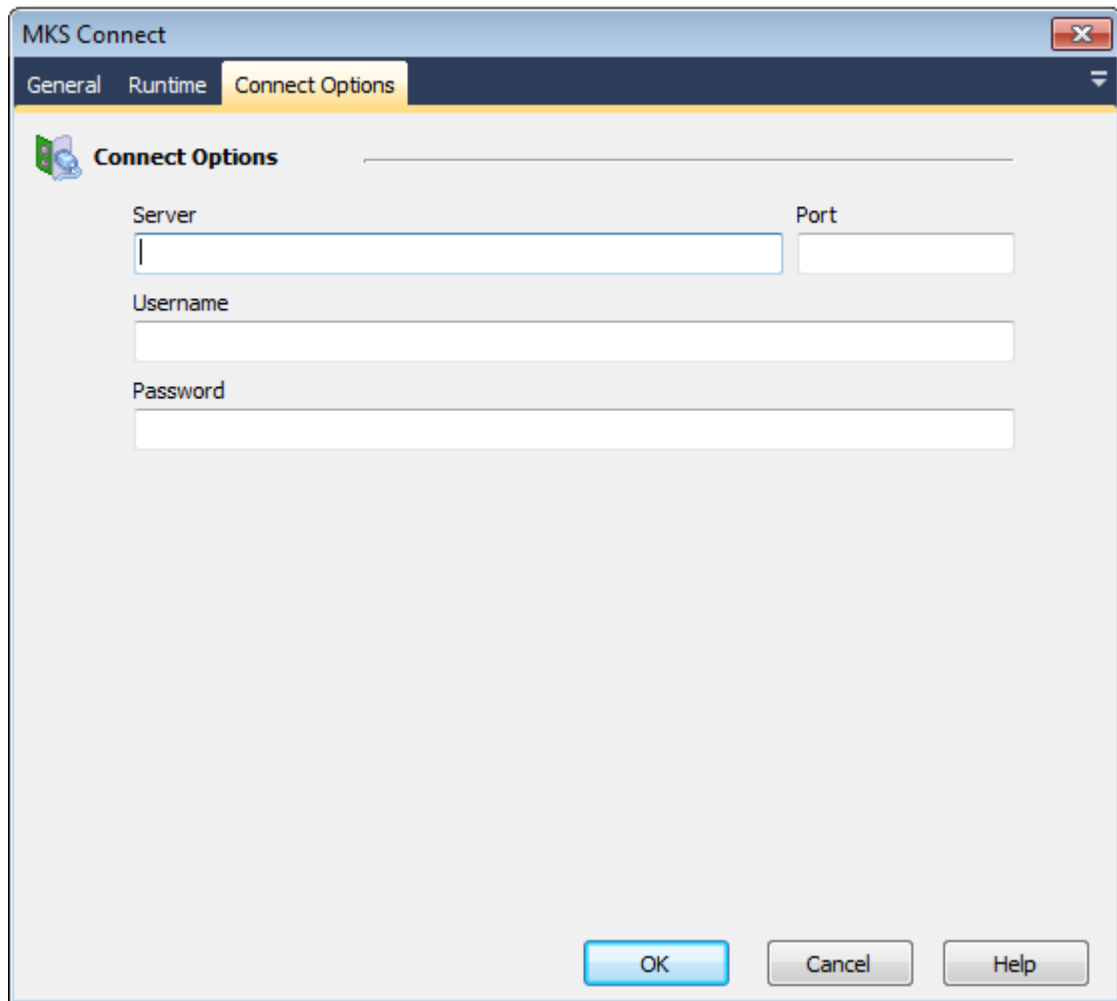
Controls whether outstanding locks are to be removed when the change package is closed.

Remove Work-In-Progress entries

Controls whether or not outstanding work-in-progress entries are to be removed from the change package.

6.34.8.7 MKS Connect

The MKS Connect action enables you to establish a connection with an Integrity Server.



The image shows a screenshot of the 'MKS Connect' dialog box. The dialog has a title bar with the text 'MKS Connect' and a close button. Below the title bar are three tabs: 'General', 'Runtime', and 'Connect Options'. The 'Connect Options' tab is selected and highlighted. The main area of the dialog is titled 'Connect Options' and contains four input fields: 'Server', 'Port', 'Username', and 'Password'. The 'Server' and 'Port' fields are on the same line, with 'Server' on the left and 'Port' on the right. Below them are the 'Username' and 'Password' fields. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Server

The name of the host server where the Integrity Server is located.

Port

The port on the host server where the Integrity Server is located.

Username

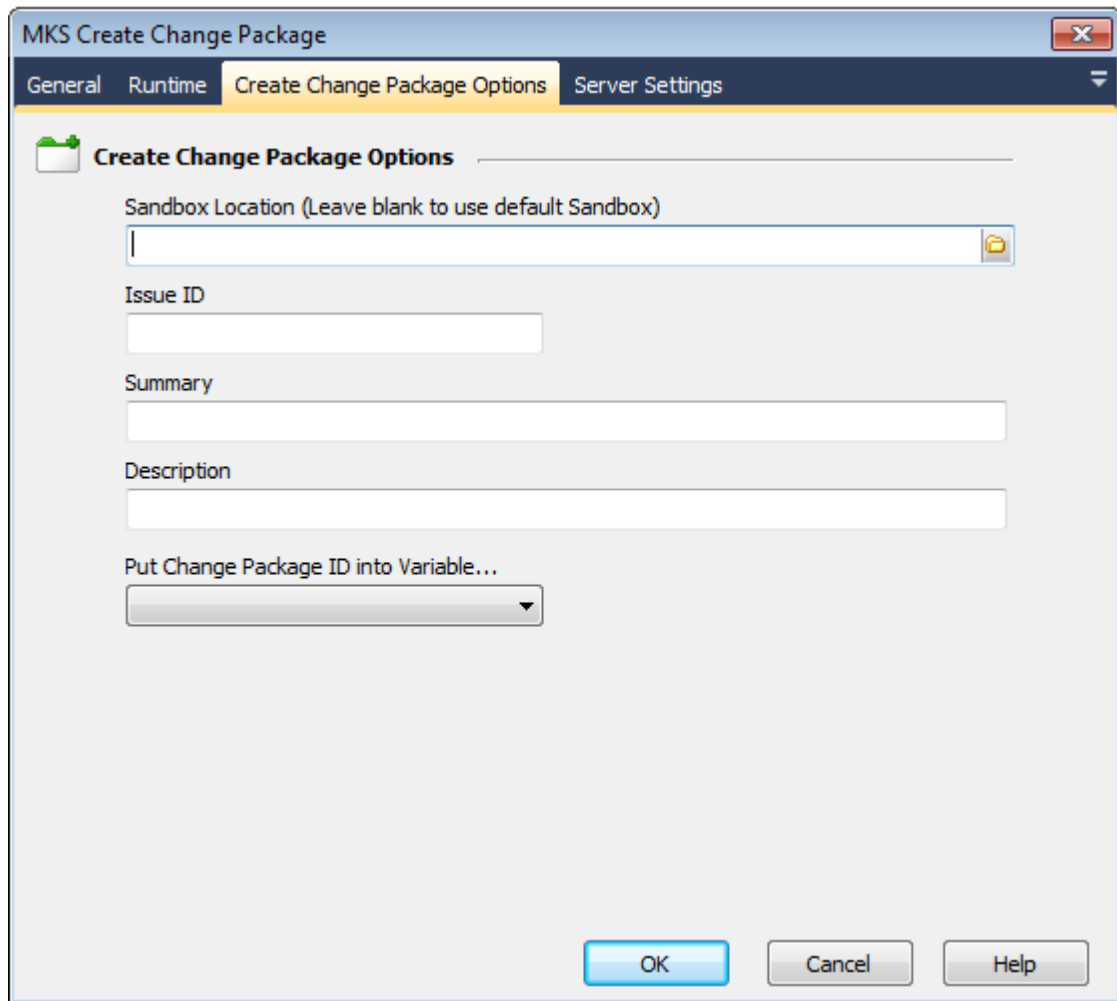
The user to use for connecting to the Integrity Server.

Password

The password to use for connecting to the Integrity Server.

6.34.8.8 MKS Create Change Package

The MKS Create Change Package action enables you to create a new change package.



The screenshot shows a dialog box titled "MKS Create Change Package" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog has four tabs: "General", "Runtime", "Create Change Package Options" (which is selected and highlighted in yellow), and "Server Settings". Below the tabs, there is a section titled "Create Change Package Options" with a folder icon. This section contains several input fields: "Sandbox Location (Leave blank to use default Sandbox)" with a text box and a folder selection icon; "Issue ID" with a text box; "Summary" with a text box; "Description" with a text box; and "Put Change Package ID into Variable..." with a dropdown menu. At the bottom right of the dialog are three buttons: "OK", "Cancel", and "Help".

Sandbox Location

The local path to the location of the sandbox.

Issue ID

The ID of the issue you are creating the change package for.

Summary

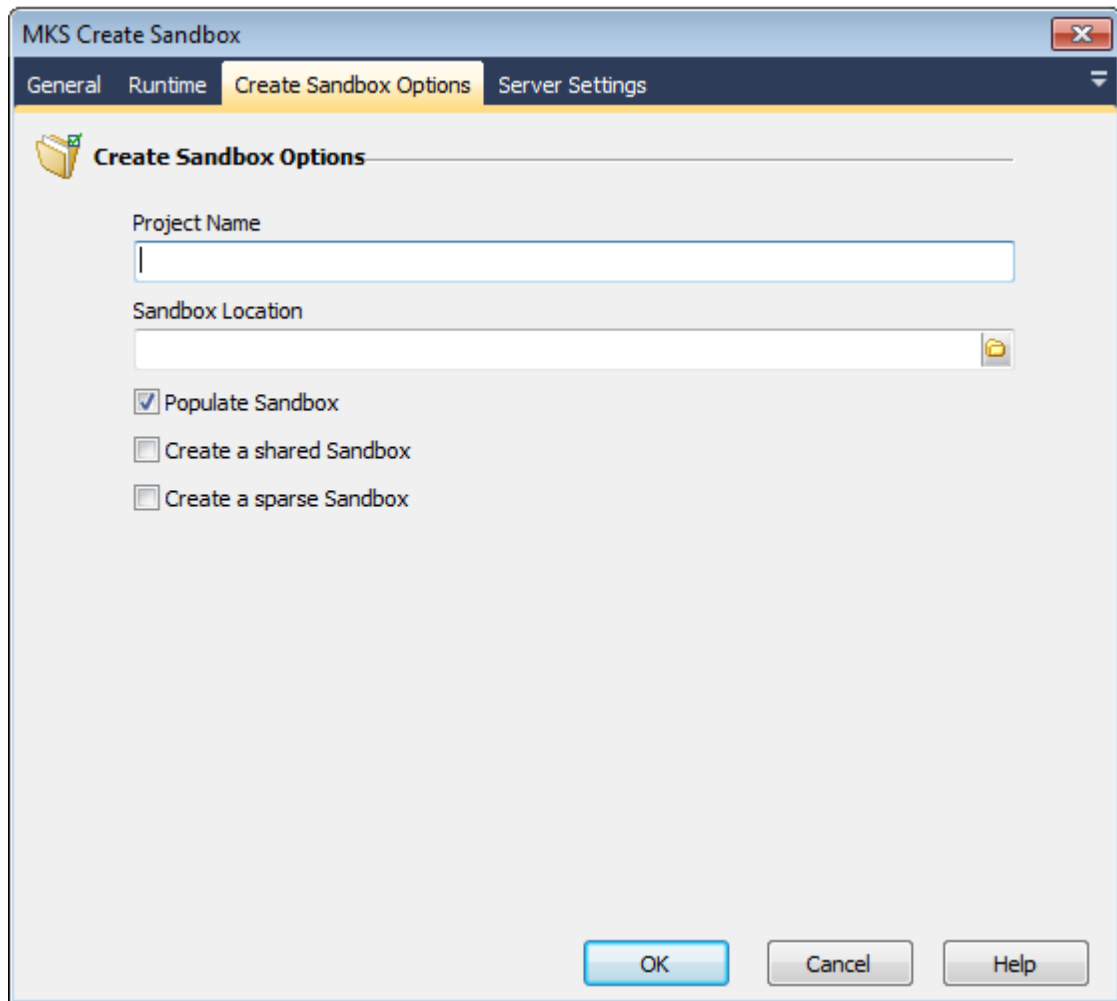
A brief summary of the change package being created.

Put Change Package ID into Variable...

The variable to hold the ID of the new Change Package.

6.34.8.9 MKS Create Sandbox

The MKS Create Sandbox action enables you to create new Sandbox on a client machine in the specified directory.

**Project Name**

The name of the project registered with the MKS Integrity server.

Sandbox Location

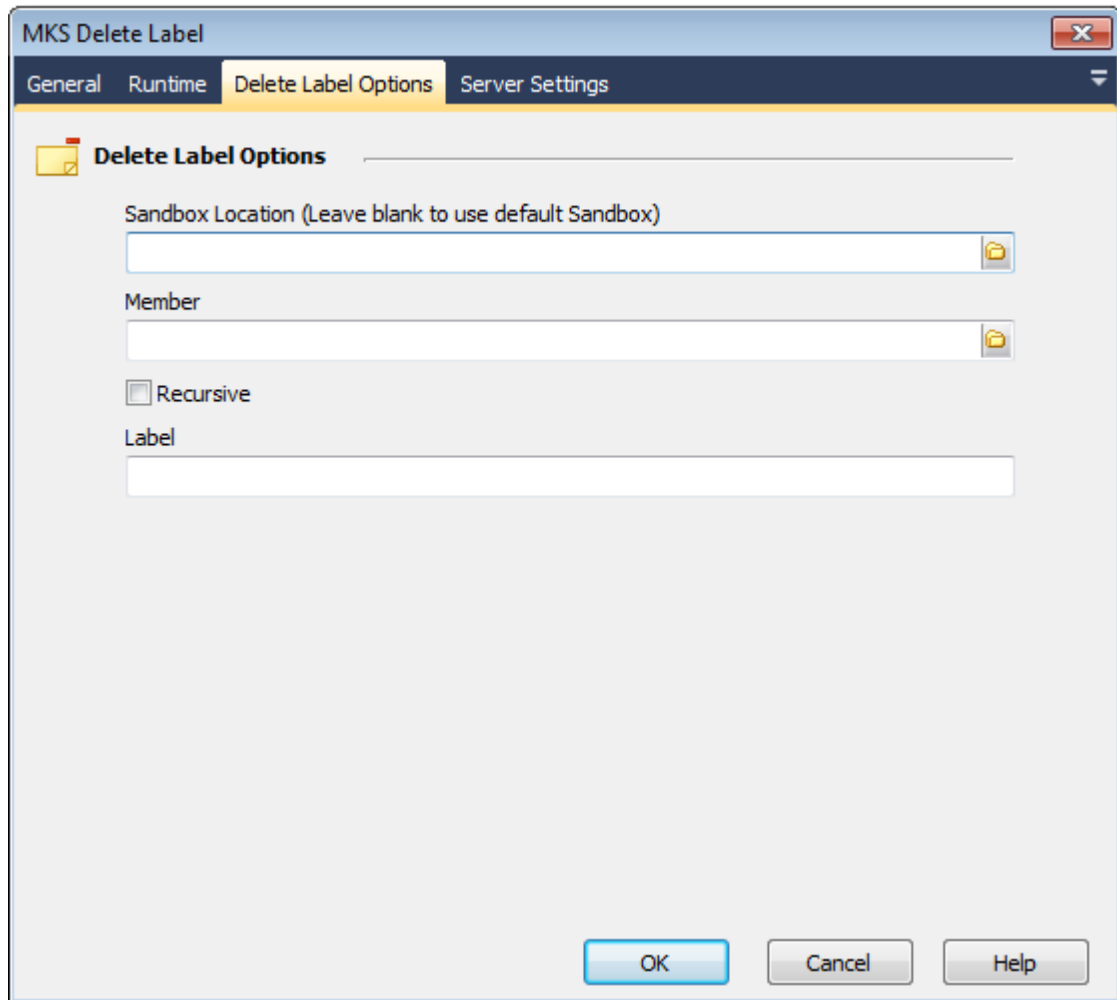
The local path to the location of the sandbox.

Populate Sandbox

Controls whether to populate the sandbox with read-only working files for all members.

6.34.8.10 MKS Delete Label

The MKS Delete Label action enables you to remove a label from one or more members.

**Sandbox Location**

The local path to the location of the sandbox.

Member

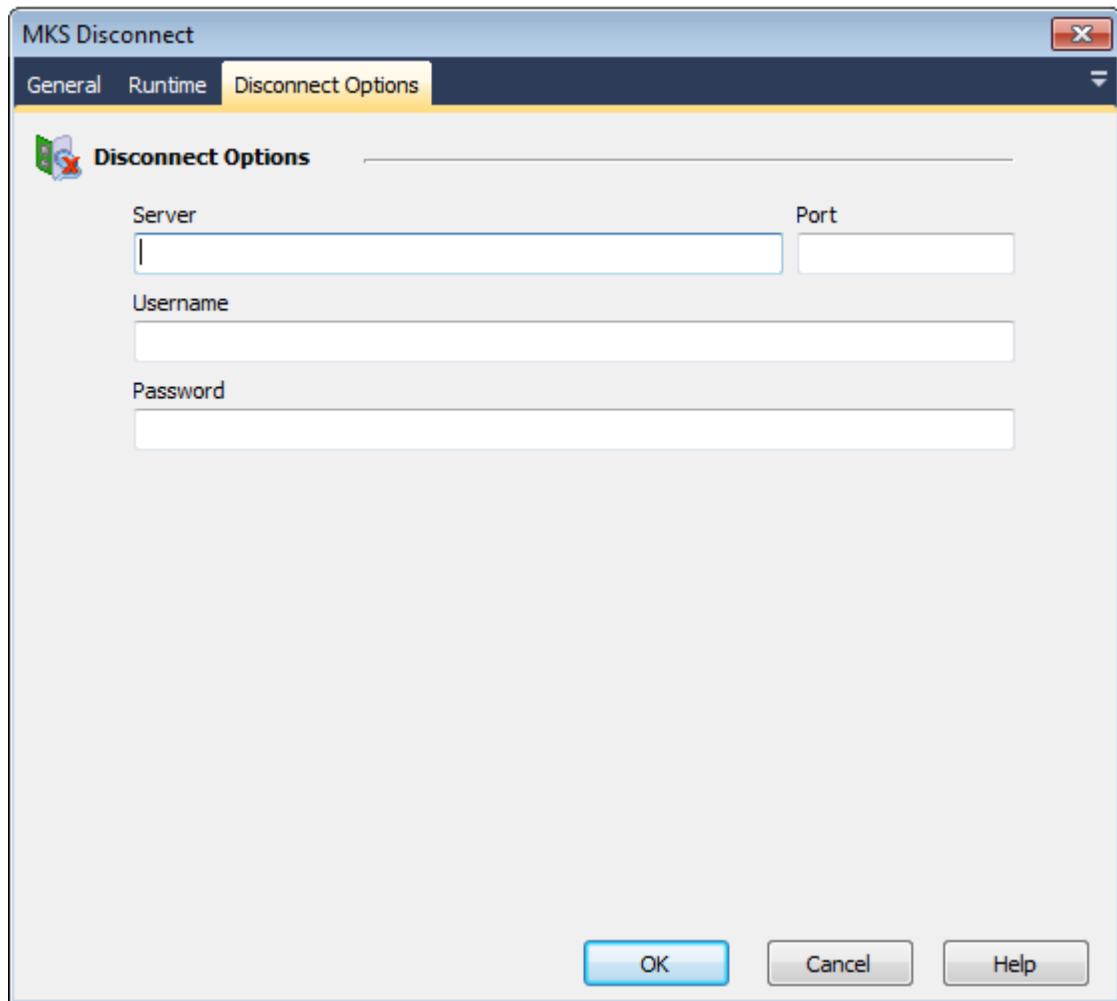
The member to remove the label from.

Label

The label to be removed from the members.

6.34.8.11 MKS Disconnect

The MKS Disconnect action enables you to disconnect the client connection to the Integrity Server.



The image shows a Windows-style dialog box titled "MKS Disconnect". It has three tabs: "General", "Runtime", and "Disconnect Options", with the latter being the active tab. Below the tabs, there is a section titled "Disconnect Options" with a small icon of a computer and a red 'X'. This section contains four input fields: "Server" and "Port" (smaller fields side-by-side), "Username" (a larger field), and "Password" (a larger field). At the bottom right of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Server

The name of the host server where the Integrity Server is located.

Port

The port on the host server where the Integrity Server is located.

Username

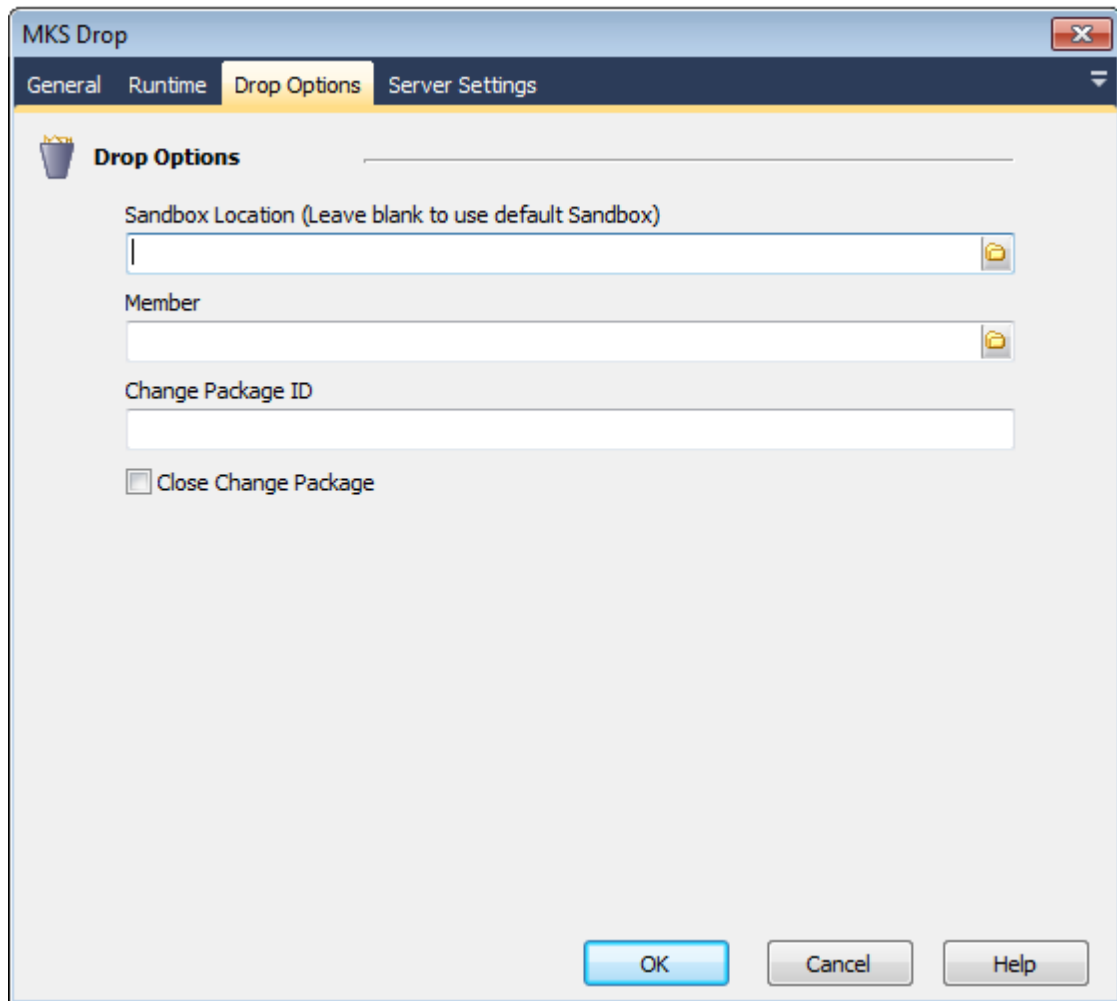
The user to use for connecting to the Integrity Server.

Password

The password to use for connecting to the Integrity Server.

6.34.8.12 MKS Drop

The MKS Drop action enables you to drop specified members or sub-projects from a project.

**Sandbox Location**

The local path to the location of the sandbox.

Member

The member to drop.

Change Package ID

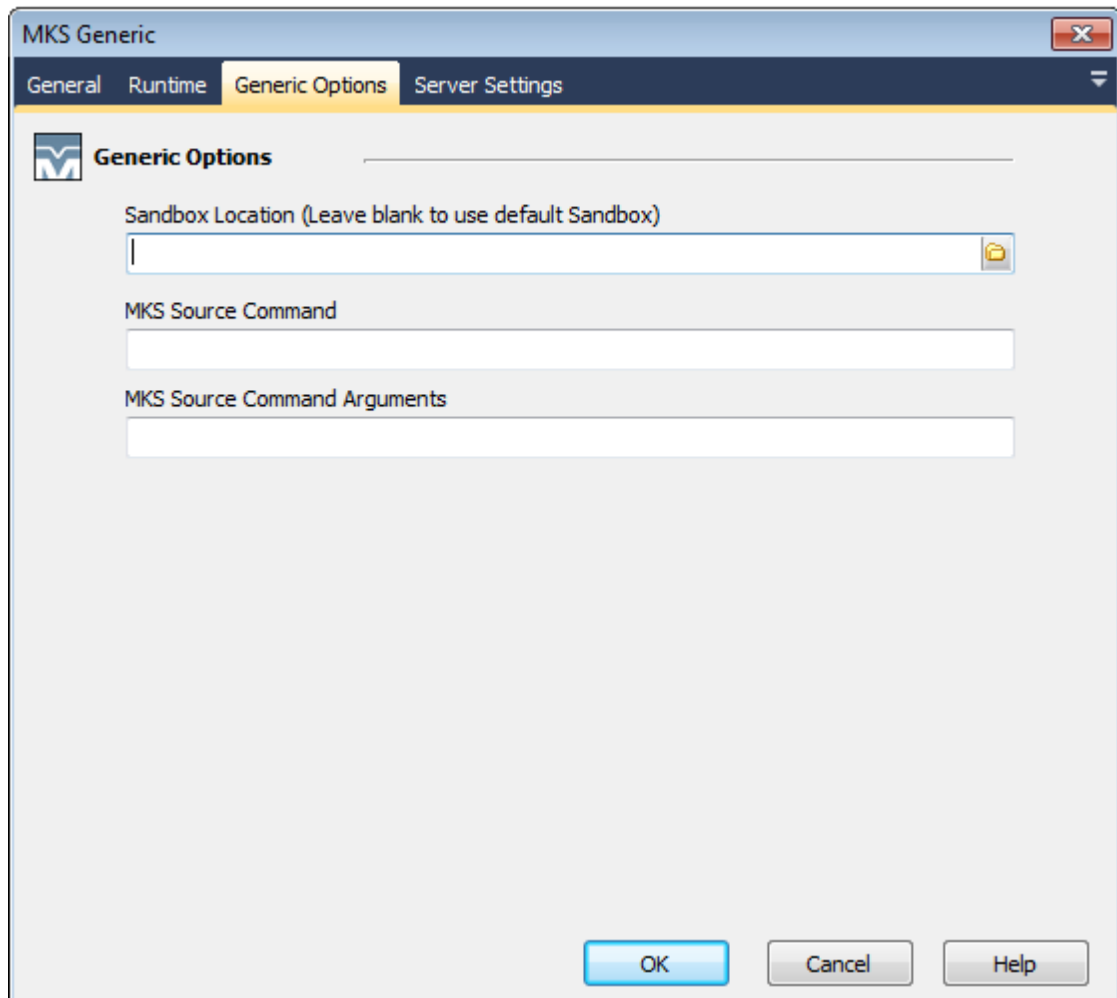
The ID of the change package to be notified of this action.

Close Change Package

Controls whether to close the associated change package.

6.34.8.13 MKS Generic

The MKS Generic action enables you to be able to call any MKS command with the specified arguments.

**Sandbox Location**

The local path to the location of the sandbox.

MKS Source Command

The command to be executed, for a complete list of possible commands refer to [SourceCLIRreference.pdf](#).

MKS Source Command Arguments

The arguments to be supplied to the command.

Examples

Show all active server connections, including their build version

Command: servers

Command Arguments: --showVersion

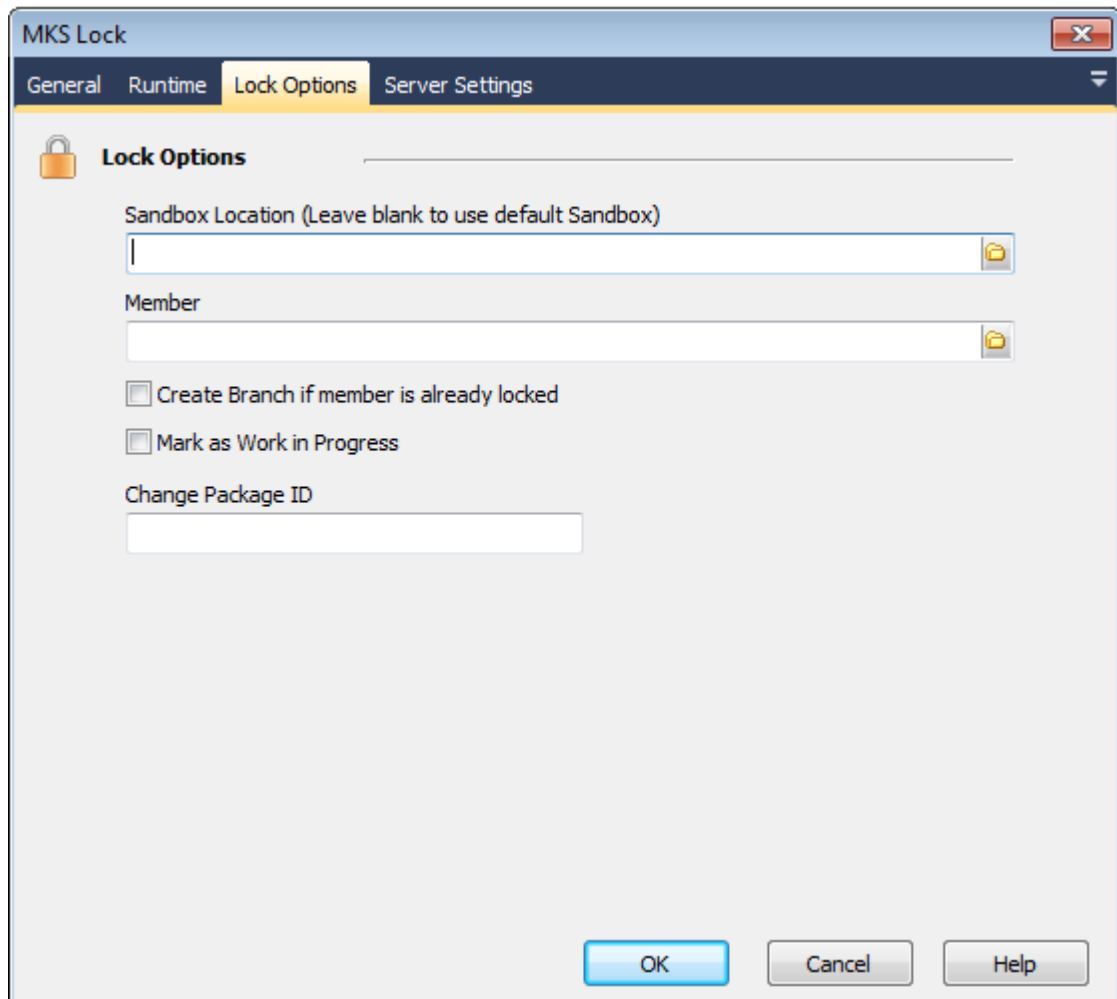
Freeze a project member, in this case the member is File.txt

Command: freeze

Command Arguments: File.txt

6.34.8.14 MKS Lock

The MKS Lock action enables you to lock one or more project members.



Sandbox Location

The local path to the location of the sandbox.

Member

The member to be locked.

Create Branch if member is already locked

Controls whether to create a branch if someone else has the member locked.

Mark as Work In Progress

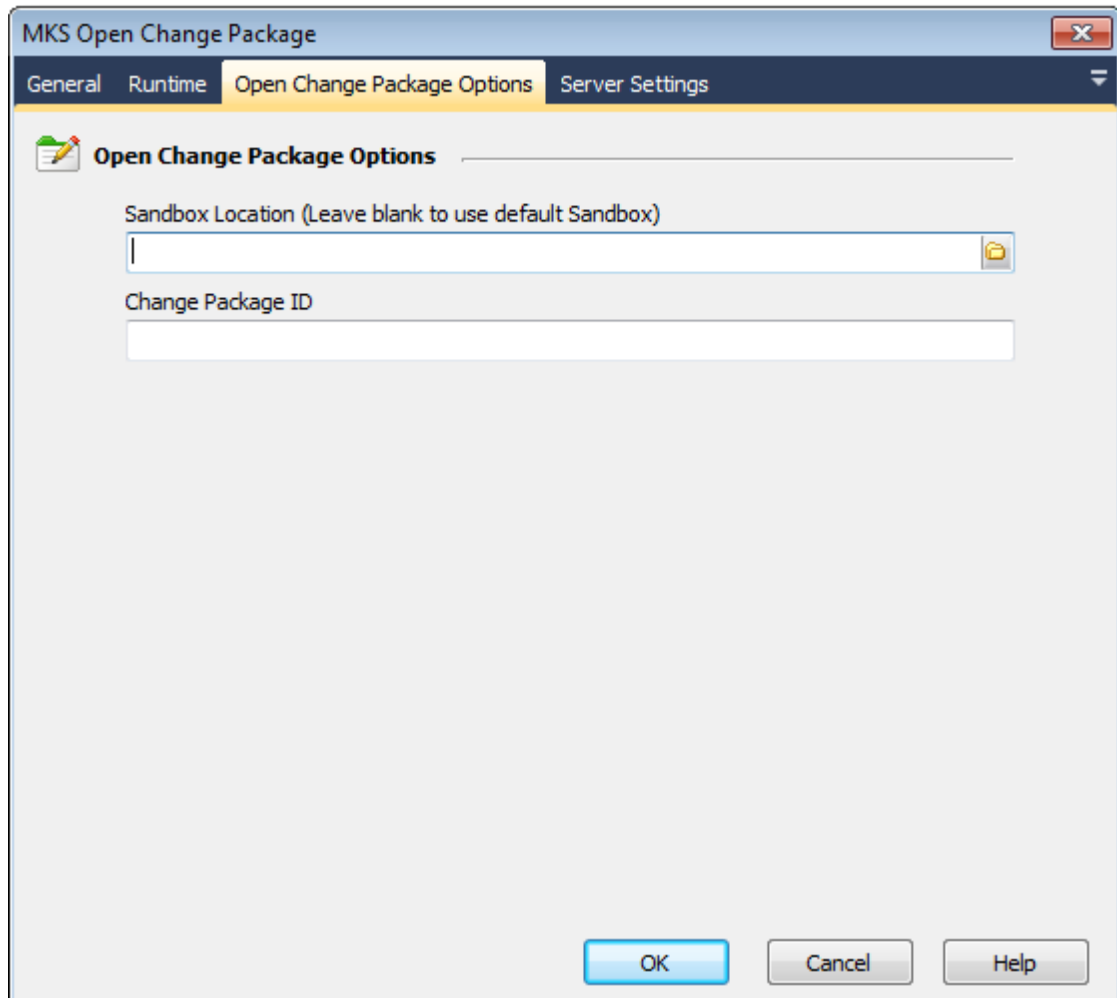
Controls whether to mark the member as Work-In-Progress.

Change Package ID

The ID of the change package to be notified of this action.

6.34.8.15 MKS Open Change Package

The MKS Open Change Package action enables you to reopen a change package.



Sandbox Location

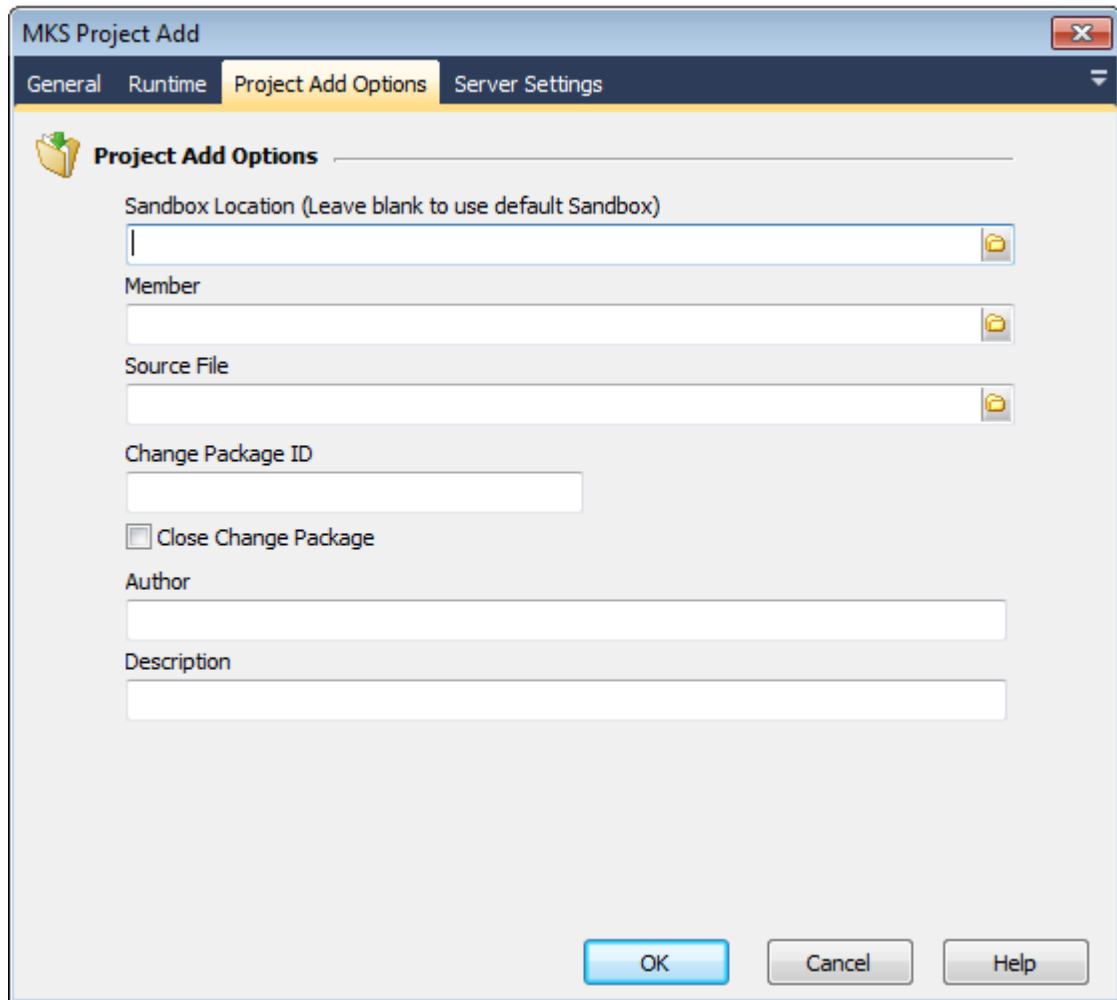
The local path to the location of the sandbox.

Change Package ID

The ID of the change package that is to be opened.

6.34.8.16 MKS Project Add

The MKS Project Add action enables you to add members to a project.



The screenshot shows the 'MKS Project Add' dialog box with the 'Project Add Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Project Add Options' (which is highlighted), and 'Server Settings'. The 'Project Add Options' tab contains a folder icon and the text 'Project Add Options'. Below this are several input fields: 'Sandbox Location (Leave blank to use default Sandbox)' with a folder selection button, 'Member' with a text input field and a folder selection button, 'Source File' with a text input field and a folder selection button, 'Change Package ID' with a text input field, a checkbox labeled 'Close Change Package', 'Author' with a text input field, and 'Description' with a text input field. At the bottom right are three buttons: 'OK', 'Cancel', and 'Help'.

Sandbox Location

The local path to the location of the sandbox.

Member

The name that the new member will have.

Source File

The file that will be checked in.

Change Package ID

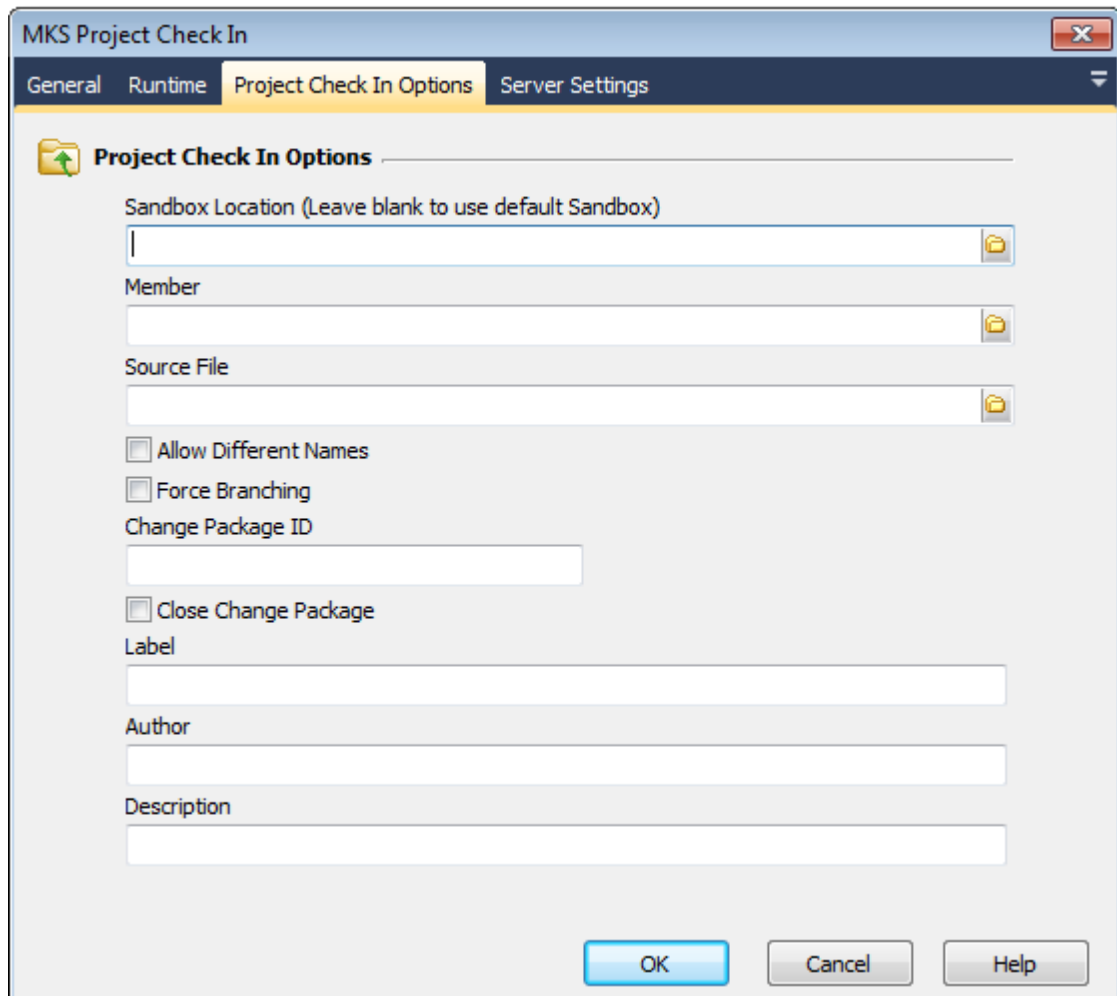
The ID of the change package to be notified of this action.

Close Change Package

Controls whether to close the associated change package.

6.34.8.17 MKS Project Check In

The MKS Project Add action enables you to check in and save changes to a project members through a project.



The screenshot shows the 'MKS Project Check In' dialog box with the 'Project Check In Options' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Project Check In Options' (active), and 'Server Settings'. The 'Project Check In Options' tab contains the following fields and controls:

- Sandbox Location** (Leave blank to use default Sandbox): A text field with a folder icon on the right.
- Member**: A text field with a folder icon on the right.
- Source File**: A text field with a folder icon on the right.
- ☐ **Allow Different Names**
- ☐ **Force Branching**
- Change Package ID**: A text field.
- ☐ **Close Change Package**
- Label**: A text field.
- Author**: A text field.
- Description**: A text field.

At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Sandbox Location

The local path to the location of the sandbox.

Member

The name of member to be check in.

Source File

The file that will be checked in.

Allow Different Names

Allows the source file to have a different name to the member.

Force Branching

Controls whether to force the creation of a branch. Turning this option off does not stop MKS from creating a branch if required.

Change Package ID

The ID of the change package to be notified of this action.

Close Change Package

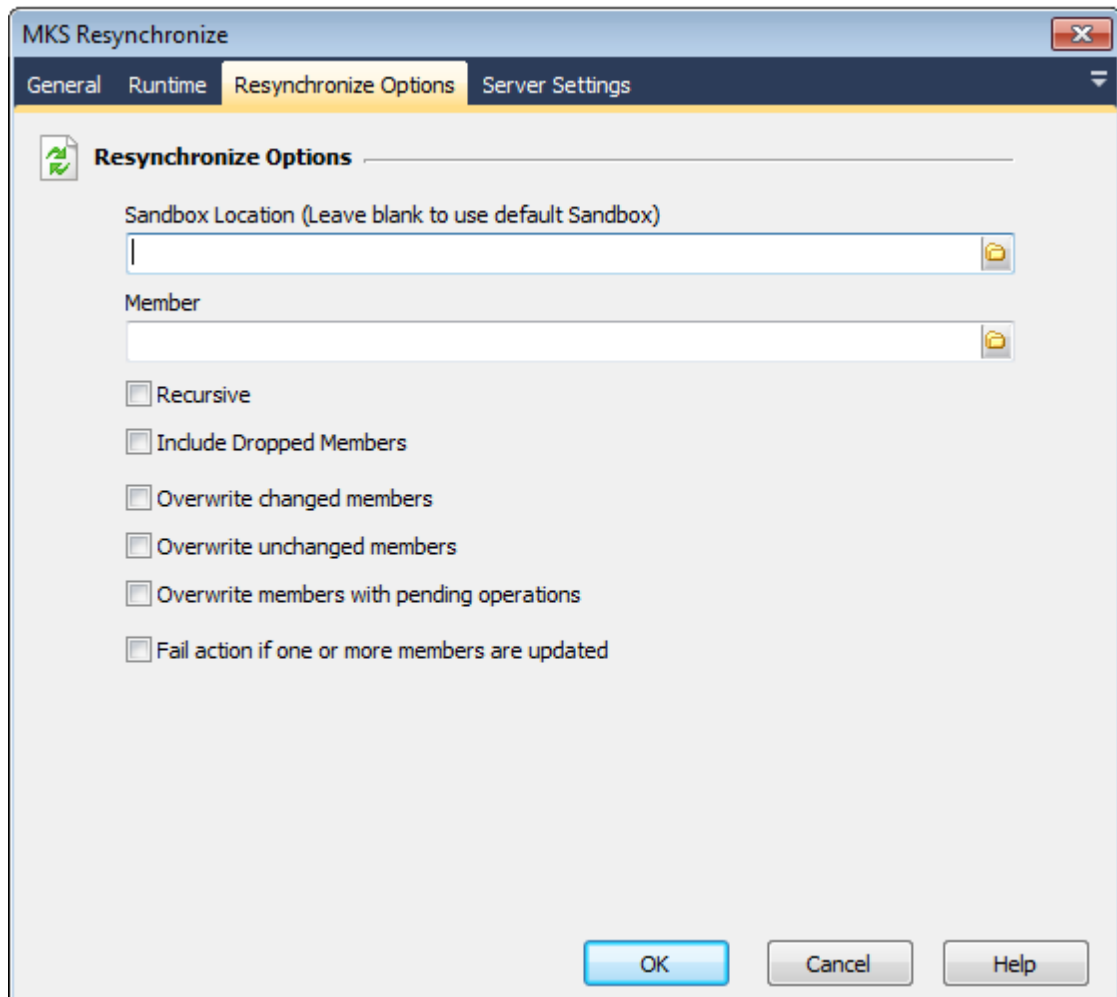
Controls whether to close the associated change package.

Label

The label to be applied to the member.

6.34.8.18 MKS Resynchronize

The MKS Resynchronize action enables you to update the working files in your Sandbox with the corresponding project members.

**Sandbox Location**

The local path to the location of the sandbox.

Member

The member to be resynchronized, leave blank for the entire project.

Include Dropped Members

Controls whether to delete dropped members from the sandbox.

Overwrite Changed Members

Controls whether to overwrite the working files if they have changed.

Overwrite Unchanged Members

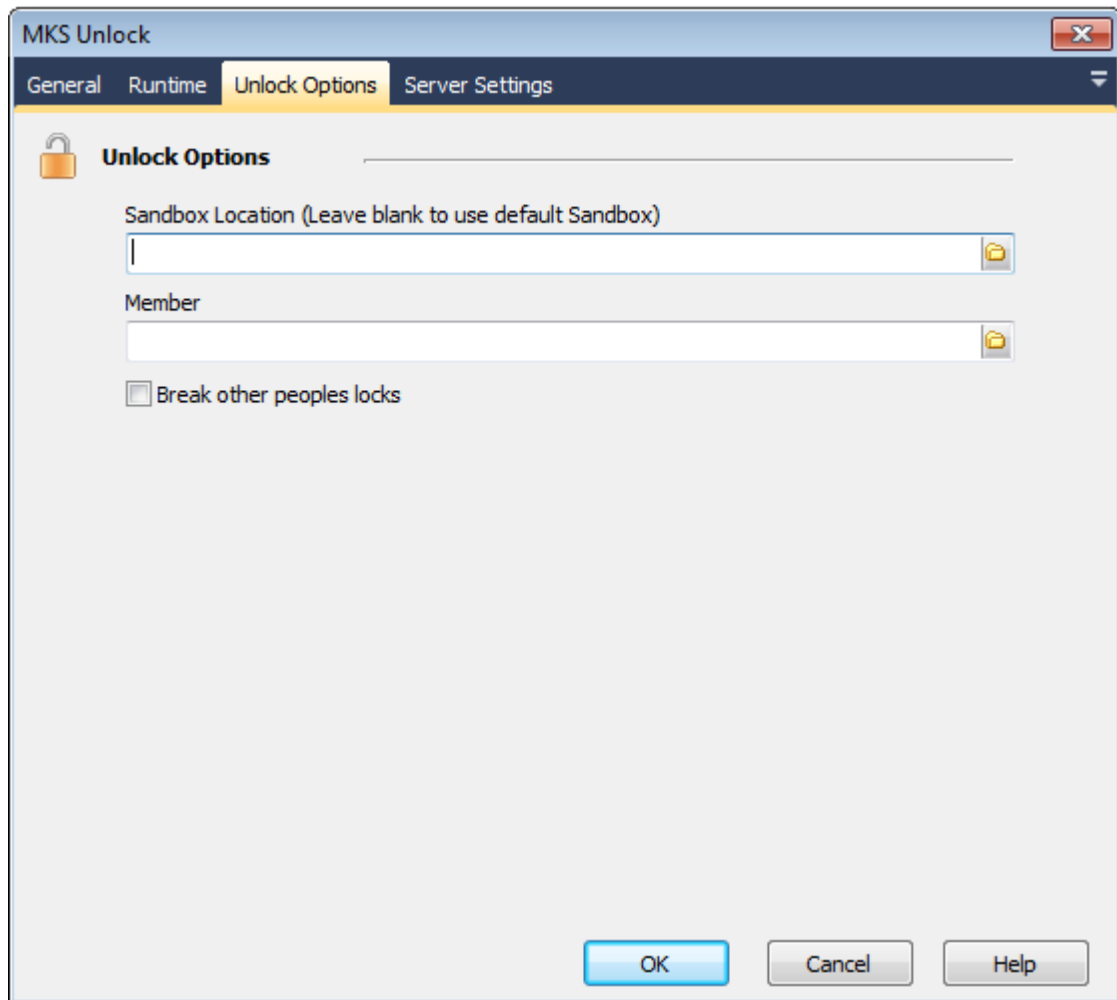
Controls whether to overwrite any unchanged members in the sandbox.

Overwrite members with pending operations

Controls whether to overwrite members that have pending operations on them.

6.34.8.19 MKS Unlock

The MKS Unlock action enables you to unlock one or more project members.



Sandbox Location

The local path to the location of the sandbox.

Member

The member to be unlocked.

Break other peoples locks

Controls whether MKS unlocks members that have locks on them owned by someone else.

6.34.9 Perforce

The Perforce Software Configuration Management System features comprehensive SCM capabilities built around a scalable client/server architecture. Users can access the Perforce Server through a range of Perforce client software. Perforce can be deployed quickly and easily, and requires minimal administration even for large or distributed sites.

For more information, see the Perforce website: <http://www.perforce.com/>

There is an old set of Perforce actions (Perforce Sync and Perforce Command) which are now deprecated, but are still available for backwards compatibility. We recommend that you disable or not install the old Perforce package if you don't need these actions.

6.34.9.1 Perforce Add Files

This action allows you to open files in a workspace for addition to the depot.

p4 command: add

"p4 add opens files within the client workspace for addition to the depot. The specified file(s) are linked to a changelist; the files are not actually added to the depot until the changelist is sent to the server with p4 submit. The added files must either not already exist in the depot, or exist in the depot but be marked as deleted at the head revision."

Add Items

Use the controls provided to specify the items to be added.

Options:

Add Files To Default Changelist: Files are linked to the default changelist.

Add Files To Specified Changelist: Files are linked to the specified changelist.

Add file(s) as: Specify the file type for the items being added. This overrides Perforce's default file type detection mechanism.

Force Inclusion of Wildcards in Filenames: Forces literal interpretation of characters that are generally used as wildcards (i.e. @, #, * and %).

Preview files to be added (files not actually added): Files are not actually added and metadata is not changed, simply a preview of which files would be added given the current parameters.

Override Defaults: Override Perforce Global Options

6.34.9.2 Perforce Create Branch

This action allows you to create a branch in a Perforce depot.

p4 command: branch

"p4 branch enables you to construct a mapping between two sets of files for use with

p4 integrate. A branch view defines the relationship between the files you're integrating from (the fromFiles) and the files you're integrating to (the toFiles). Both sides of the view are specified in depot syntax."

Options:

Branch Name: the name to give to the new branch

Description: a description of the new branch

Branch Mappings (branch view): define one or more mappings between a current source and the destination (for the new branch). Both the source and destination are in the format `//<depot>/<folder>/<file spec>`. To specify all files and recurse subdirectories use the syntax "...", eg. `//depot/my folder/...`

Locked Branch: If the branch is locked, only the Owner can modify the branch spec, and the spec can't be deleted until it is unlocked

Override Defaults: Override Perforce Global Options

6.34.9.3 Perforce Create Changelist

This action allows you to create a changelist in your Perforce depot.

p4 command: change

"When files are opened with p4 add, p4 delete, p4 edit, or p4 integrate, the files are listed in a changelist. Edits to the files are kept in the local client workspace until the changelist is sent to the depot with p4 submit. By default, files are opened within the default changelist, but multiple changelists can be created and edited with the p4 change command."

Options:

Create new changelist: specify the FinalBuilder variable which will be set to the new changelist number given by Perforce when the action finishes running.

Specify Changelist #: if you want to update the values/files/description of an existing changelist, then specify the changelist number

Changelist Description: Textual description of changelist. The description cannot be changed after submission, except by the Perforce superuser.

Use Files in default Changelist: Specify this option to add all files in the default changelist to the new changelist

Specify files for Changelist: The list of files being submitted in this changelist.

Jobs: A list of jobs that are fixed by this changelist. The list of jobs that appears when the form is first displayed is controlled by the p4 user form's JobView: setting.

Override Defaults: Override Perforce Global Options

6.34.9.4 Perforce Create Label

This action allows you to create a label in your Perforce depot.

p4 command: `label`

"Create or edit a label specification and its view. Running p4 label allows you to configure the mapping that controls the set of files that are allowed to be included in the label. After configuring the label, use p4 labelsync or p4 tag to tag files with the label."

Options:

Label name: The name of the label

Owner: The label's owner. By default, the user who created the label. Only the owner of a label may update what files are tagged with the label.

Description: An optional description of the label's purpose.

Views: A list of depot files that can be tagged with this label. No files are actually tagged until p4 labelsync is invoked. Unlike client views or branch views, which map one set of files to another, label views consist of a simple list of depot files.

Locked: If the label is locked, the list of files tagged with the label cannot be changed with p4 labelsync.

Override Defaults: Override Perforce Global Options

Notes:

If you specify an existing label name, then the label will be updated with all of the specified values. If you only want to update some of the fields, use the Perforce Update Label action.

6.34.9.5 Perforce Delete Branch

This action allows you to delete a branch in a Perforce depot.

p4 command: `branch -d`

"Delete the named branch specification. Files are not affected by this operation; only the stored mapping from one codeline to another is deleted. Normally, only the user who created the branch can use this flag."

Options:

Branch Name: the name of the branch to delete

Force (-f): force deletion even if the branch is locked. This allows Perforce administrators to delete branches they don't own.

Override Defaults: Override Perforce Global Options

6.34.9.6 Perforce Delete Changelist

This action allows you to delete a changelist.

p4 command: `change -d`

Options:

Changelist: the changelist number to delete

Force (-f) : Forcibly delete a previously submitted changelist. Only a Perforce administrator or superuser can use this command, and the changelist must have had all of its files removed from the system with p4 obliterate.

Override Defaults: Override Perforce Global Options

6.34.9.7 Perforce Delete Label

This action allows you to delete a label from your Perforce depot.

p4 command: `label -d`

Options:

Label name: The name of the label to delete

Force deletion (-f) : The -f flag forces the deletion even if the label is locked. (Deleting a locked label requires admin or super access.)

Override Defaults: Override Perforce Global Options

6.34.9.8 Perforce Generic

This action allows you to run other p4 command that do not have specific FinalBuilder actions.

Perforce Command: either specify or select the Perforce command to run (eg. add, client, have, jobs, etc). Press the "Load" button to load the combo box with the available Perforce commands. Press the "Get Help" button to load the Perforce help for the selected command.

Command Line Arguments: specify the various command line arguments to send to p4. This field accepts FinalBuilder variables, eg. -c %ChangeListNum%

Redirect file from STDIN: many Perforce command allow you to specify a -i parameter which means the input for the command can come from a file rather than Perforce launching a text editor with a text file that you need to fill out. Obviously launching a text editor in an automated build process is not ideal, so use this parameter to make your build process require no user interaction. The file needs to be in a specific format for each command.

Error Strings: specify errors strings which FinalBuilder will search for in the output of the Perforce command to determine if the action should succeed or fail.

Override Defaults: Override Perforce Global Options

6.34.9.9 Perforce Labelsync

This action allows you to sync a label in your local Perforce workspace.

p4 command: labelsync

"Synchronize a label with the contents of the current client workspace. p4 labelsync causes the named label to reflect the current contents of the client workspace by tagging the last revision of each file synced into the workspace with the label name. The label name can subsequently be used in a revision specification as @label to refer to the revision of the file that was tagged with the label."

Options:

Label name: The name of the label to sync

Label deleted depot files (-a) : Add the label to files that match the file pattern arguments, even if some of the files being labeled are deleted at their head revision.

Delete the label tag from the named files (-d) : Delete the label tag from the named files.

Do not capture output: Setting this option will prevent FinalBuilder from capturing the output of the p4 command. If you are doing a large labelsync, then this can speed up the process significantly.

Depot Files: specify files to sync using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Revision: specifies the revision of the files to sync. In each instance except the head revision, file revision modifiers are appended to the file specifiers

Head Revision - The latest version of the file

Revision Number (#<revision number>) - The nth revision of a file.

Label (@<labelname>) -The revision of file in the label labelname.

Change Number (@<change number>) - The revision of file immediately after changelist n was submitted.

Date/Time (@<datetime>) - The revision of file at the date and time specified.

Override Defaults: Override Perforce Global Options

Notes on usage:

- If you do not specify any depot files, then labelsync causes the label to reflect the contents of the client workspace by adding, deleting, and updating the set of files tagged with the label.
- If a file is given, labelsync updates the tag for only that named file. If the file argument includes a revision specification, then that revision is used instead of the revision existing in the workspace. If the file argument includes a revision range, then only the highest revision in that range is used.
- Only the Owner of an unlocked label may use labelsync to tag files with that label.
- A label that is set to locked cannot be updated with labelsync.

6.34.9.10 Perforce Lock

This action allows you to lock files in your Perforce depot.

p4 command: lock

"Lock an opened file against changelist submission. Locking files prevents all other users from submitting changes to those files. If the files are already locked by another user, p4 lock fails. When the user who locked a particular file submits the file, the lock is released."

Options:

Changelist (-c) : Lock only files included in the specified changelist

Depot Files: specify files to lock using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Override Defaults: Override Perforce Global Options

6.34.9.11 Perforce Old Actions

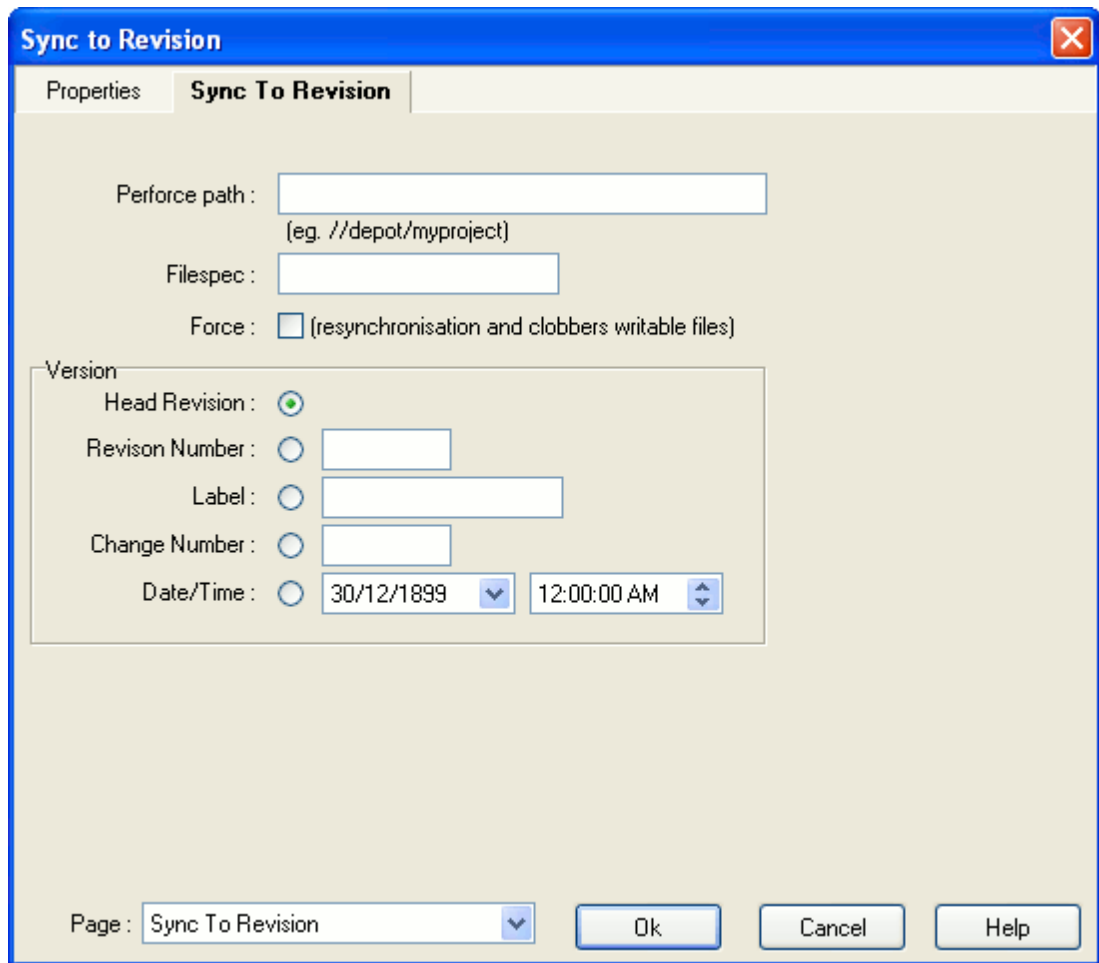
The old set of Perforce actions has been deprecated.

The help topics and actions are still available, but we encourage Perforce users to use the new set of actions.

6.34.9.11.1 Perforce Synchronise with View Action (Deprecated)

Copy files from the depot into the workspace.

The sync command can sync to either the Head Revision, Revision Number, Label, Change Number or a specific Date/Time.



The image shows a Windows-style dialog box titled "Sync to Revision". It has a blue title bar with a close button (X) in the top right corner. The dialog has two tabs: "Properties" and "Sync To Revision", with the latter being the active tab. The main area is light beige and contains the following controls:

- Perforce path :** A text input field with a placeholder example "(eg. //depot/myproject)".
- Filespec :** A text input field.
- Force :** A checkbox followed by the text "(resynchronisation and clobbers writable files)".
- Version** (grouped in a box):
 - Head Revision :** A radio button that is currently selected.
 - Revision Number :** A radio button followed by a text input field.
 - Label :** A radio button followed by a text input field.
 - Change Number :** A radio button followed by a text input field.
 - Date/Time :** A radio button followed by a date dropdown (showing "30/12/1899") and a time spinner (showing "12:00:00 AM").

At the bottom of the dialog, there is a "Page:" label followed by a dropdown menu showing "Sync To Revision", and three buttons: "Ok", "Cancel", and "Help".

For all other Perforce commands use the "Perforce Command" and enter the command line options manually.

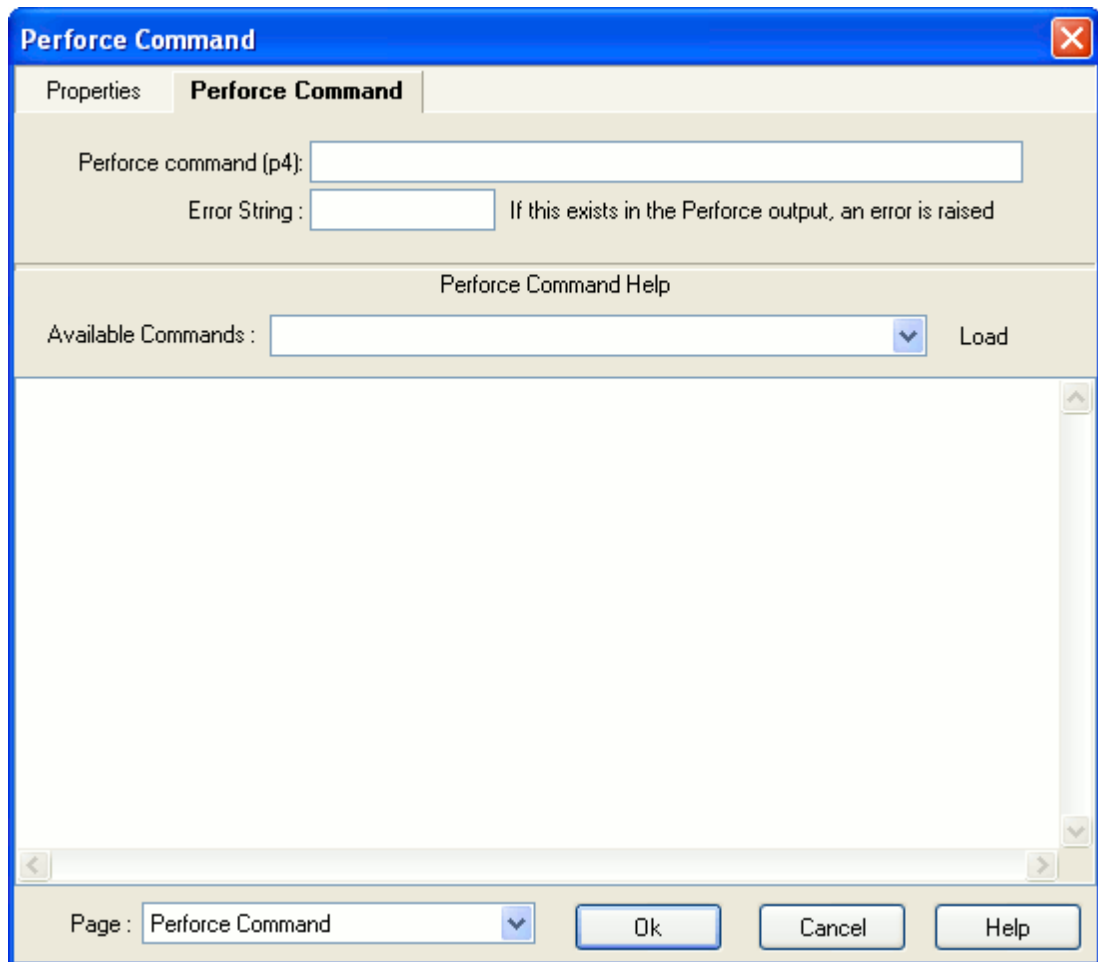
For more information on the sync command, see:

<http://www.perforce.com/perforce/doc.032/manuals/cmdref/sync.html>

This action was kindly provided by Tate Needham from t8software.com

6.34.9.11.2 Perforce Command Action (Deprecated)

This action allows you to run most Perforce commands. Note that command which prompt you for a response will not work and cause the action to hang.



The Command Reference can be found here:

<http://www.perforce.com/perforce/doc.032/manuals/cmdref/index.html>

This action was kindly provided by Tate Needham from t8software.com

6.34.9.12 Perforce open for Delete

This action allows you to prepare files for deletion from your Perforce depot.

p4 command: delete

"Open file(s) in a client workspace for deletion from the depot. The p4 delete command opens file(s) in a client workspace for deletion from the depot. The files are immediately removed from the client workspace, but are not deleted from the depot until the corresponding changelist is sent to the server with p4 submit."

Options:

Depot Files: specify files to delete using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Changelist (-c) : Opens the files for delete within the specified changelist. If this flag is not provided, the files are linked to the default changelist

Override Defaults: Override Perforce Global Options

6.34.9.13 Perforce open for Edit

This action allows you to open files for editing from your Perforce depot.

p4 command: edit

"Opens file(s) in a client workspace for edit. The specified file(s) are linked to a changelist, but the files are not actually changed in the depot until the changelist is sent to the server by p4 submit."

Options:

Depot Files: specify files to open using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Changelist number (-c) : Opens the files for edit within the specified changelist. If this flag is not provided, the files are linked to the default changelist.

Filetype (-t) : Stores the new file revision as the specified type, overriding the file type of the previous revision of the same file.

Modifiers: apply certain file type modifiers to the files

File always writeable (+w)

Old style keyword expansion (+ko) : Expands only the \$Id\$ and \$Header\$ keywords

Exclusive open (+l): If set, only one user at a time will be able to open a file for editing.

Store deltas (+D) : Default server storage mechanism for text files.

Store only head revision (+S) : Older revisions are purged from the depot upon submission of new revisions. Useful for executable or .obj files.

Executable file (+x) : Used for executable files.

RCS keyword expansion (+k) : Expands RCS (Revision Control System) keywords.

Store full version compressed (+C) : Default server storage mechanism for binary files.

Store full version uncompressed (+F) : Useful for large binaries, or for long ASCII files that aren't read by users as text, such as PostScript files.

Preserve local timestamp (+m) : The file's timestamp on the local file system is preserved upon submission and restored upon sync. Useful for third-party DLLs in Windows environments.

Override Defaults: Override Perforce Global Options**6.34.9.14 Perforce Opened**

This action allows you to list opened files.

p4 command: opened

"List files that are open in pending changelists. Use p4 opened to list files that are currently open via p4 add, p4 edit, p4 delete, or p4 integrate. By default, all open files in the current client workspace are listed. You can use command line arguments to list only those files in a particular pending changelist, or to show open files in all pending changelists, and to limit the number of files displayed."

Options:

List opened files in any client workspace (-a) : by default only files in the current workspace will be listed.

Specify changelist number (-c changelist#) : List the files in pending changelist changelist#. To list files in the default changelist, leave this field blank.

Perforce filespec: specify a filespec to limit the files listed which match the filespec

Override Defaults: Override Perforce Global Options**6.34.9.15 Perforce Options**

Specify the global options for all the Perforce actions.

P4.EXE Location: FinalBuilder attempts to detect where Perforce is installed, if this doesn't succeed then specify where FinalBuilder will find P4.exe

Client name (g-opts: -c) : Overrides any P4CLIENT setting with the specified client name.

Hostname (g-opts: -H) : Overrides any P4HOST setting and replaces it with the specified hostname.

Port (g-opts: -p) : Overrides any P4PORT setting with the specified port number.

Username (g-opts: -u) : Overrides any P4USER, USER, or USERNAME setting with the specified user name.

Password (g-opts: -P) : Overrides any P4PASSWD setting with the specified password.

Current Working Directory (g-opts: -d) : Overrides any PWD setting (i.e. current working directory) and replaces it with the specified directory.

Note:

- It is advised not to use Perforce environment variables in the various Perforce fields (in Perforce options or in the Perforce actions). Instead, define your own FinalBuilder variables and set them to the required values.
- These global options (except for the p4.exe location) can be overridden in each Perforce action

6.34.9.15.1 Override Perforce Global Options

Each Perforce action has the option to either:

- Use the global Perforce options (as specified in Tools menu | Options), or
- Use the Perforce defaults (ie. don't override the Perforce environment variables), or
- Override the Perforce global defaults

The options available when overriding the Perforce global defaults are:

Client name (g-opts: -c) : Overrides any P4CLIENT setting with the specified client name.

Hostname (g-opts: -H) : Overrides any P4HOST setting and replaces it with the specified hostname.

Port (g-opts: -p) : Overrides any P4PORT setting with the specified port number.

Username (g-opts: -u) : Overrides any P4USER, USER, or USERNAME setting with the specified user name.

Password (g-opts: -P) : Overrides any P4PASSWD setting with the specified password.

Current Working Directory (g-opts: -d) : Overrides any PWD setting (i.e. current working directory) and replaces it with the specified directory.

6.34.9.16 Perforce Revert

This action allows you to revert changes made to open files.

p4 command: revert

"Discard changes made to open files. Use p4 revert to discard changes made to open files, reverting them to the revisions last p4 synced from the depot. This command also removes the reverted files from the pending changelists with which they're associated."

Options:

Changelist (-c) : Reverts only those files in the specified changelist.

Only revert unchanged files (-a): Revert only those files that haven't changed (in terms of content or filetype) since they were opened.

Override Defaults: Override Perforce Global Options

Usage Notes:

- When you revert files you opened with p4 delete, the files are reinstated in the client workspace.
- When you revert files that have been opened by p4 add, Perforce leaves the client workspace files intact.
- When you revert files you've opened with p4 integrate, Perforce removes the files from the client workspace.

6.34.9.17 Perforce Submit

This action allows you to submit files to your Perforce depot.

p4 command: submit

"Send changes made to open files to the depot. When a file has been opened by p4 add, p4 edit, p4 delete, or p4 integrate, the file is listed in a changelist. The user's changes to the file are made only within in the client workspace copy until the changelist is sent to the depot with p4 submit."

Options:

Submit files in the default changelist: By default files are opened in the default changelist - this option will submit any files in the default changelist. Optionally specify a filespec to only submit files which match this filespec.

Submit specified changelist number: Does just this. Changelists are assigned numbers either manually by the user with p4 change, or automatically by Perforce when submission of the default changelist fails.

Reopen files for edit (-r) : Reopen files for edit in the default changelist after submission. Files opened for add or edit in will remain open after the submit has completed.

Description: Textual description of changelist (required)

Override Defaults: Override Perforce Global Options

6.34.9.18 Perforce Sync

This action allows you to sync files from your Perforce depot.

p4 command: sync

"Copy files from the depot into the workspace. p4 sync brings the client workspace into sync with the depot by copying files matching its file pattern arguments from the depot to the client workspace. When no file patterns are specified on the command line, p4 sync copies a particular depot file only if it is visible through the client view, not opened, and not already in the client workspace at it's latest revision"

Options:

Depot Files: specify files to sync using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Force Sync (-f) : Perforce performs the sync even if the client workspace already has the file at the specified revision. If the file is writable, it is overwritten.

Fail if no files synced: This option will fail the action if Perforce returns the message "file(s) up-to-date".

Revision: specifies the revision of the files to sync. In each instance except the head revision, file revision modifiers are appended to the file specifiers

Head Revision - The latest version of the file

Revision Number (#<revision number>) - The nth revision of a file.

Label (@<labelname>) -The revision of file in the label labelname.

Change Number (@<change number>) - The revision of file immediately after changelist n was submitted.

Date/Time (@<datetime>) - The revision of file at the date and time specified.

Override Defaults: Override Perforce Global Options

6.34.9.19 Perforce Tag

This action allows you to tag files with a label in your Perforce depot.

p4 command: tag

"Use p4 tag to tag specified file revisions with a label. A labelname is required. If a label named labelname does not exist, it is created automatically. If the label already exists, you must be the Owner: of the label and the label must be unlocked in order for you to tag or untag files with the label."

Options:

Label Name (-l) : the label to be applied to file revisions

Delete label tag (-d) : Delete the label tag from the named files.

Depot Files: specify files to tag using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Revision: specifies the revision of the files to sync. In each instance except the head revision, file revision modifiers are appended to the file specifiers

Head Revision - The latest version of the file

Revision Number (#<revision number>) - The nth revision of a file.

Label (@<labelname>) -The revision of file in the label labelname.

Change Number (@<change number>) - The revision of file immediately after changelist n was submitted.

Date/Time (@<datetime>) - The revision of file at the date and time specified.

Override Defaults: Override Perforce Global Options

6.34.9.20 Perforce Unlock

This action allows you to unlock files in your Perforce depot.

p4 command: unlock

"Release the lock on a file. The p4 unlock command releases locks created by p4 lock."

Options:

Changelist (-c) : Unlock files in pending changelist

Force Unlock (-f) : Superuser force flag; allows unlocking of files opened by other users.

Depot Files: specify files to unlock using the standard Perforce depot syntax (eg. //depot/folder/.../*.txt)

Override Start in Directory: use this option to specify a different directory from where p4 is run from

Override Defaults: Override Perforce Global Options

6.34.9.21 Perforce Update Label

This action allows you to update a label in your Perforce depot.

p4 command: label

"Create or edit a label specification and its view. Running p4 label allows you to configure the mapping that controls the set of files that are allowed to be included in the label. After configuring the label, use p4 labelsync or p4 tag to tag files with the label."

Options:

Label name: The name of the label

Owner: The label's owner. By default, the user who created the label. Only the owner of a label may update what files are tagged with the label.

Description: An optional description of the label's purpose.

Views: A list of depot files that can be tagged with this label. No files are actually tagged until p4 labelsync is invoked. Unlike client views or branch views, which map one set of files to another, label views consist of a simple list of depot files.

Locked: If the label is locked, the list of files tagged with the label cannot be changed with p4 labelsync.

Override Defaults: Override Perforce Global Options

Notes:

This action is designed to update certain fields of an existing label. If you want to create a new label then use the Perforce Create Label action.

6.34.9.22 Perforce Login Action

This action allows you to login to a Perforce server which is using Ticket Based Authentication.

p4 Command: login

"The p4 login command authenticates a user and creates a ticket that represents a

session with a Perforce server. Once authenticated, a user can access the Perforce server until either the ticket expires or until the user issues the p4 logout command."

Override Defaults: Override Perforce Global Options

Notes:

This action requires that the 'Use Ticket Based Authentication' is enabled in either the global options, or on the override defaults tab.

6.34.10 Plastic SCM

The Plastic SCM actions provide interaction with the Plastic SCM version control system.

For more information see the Plastic SCM webpage.

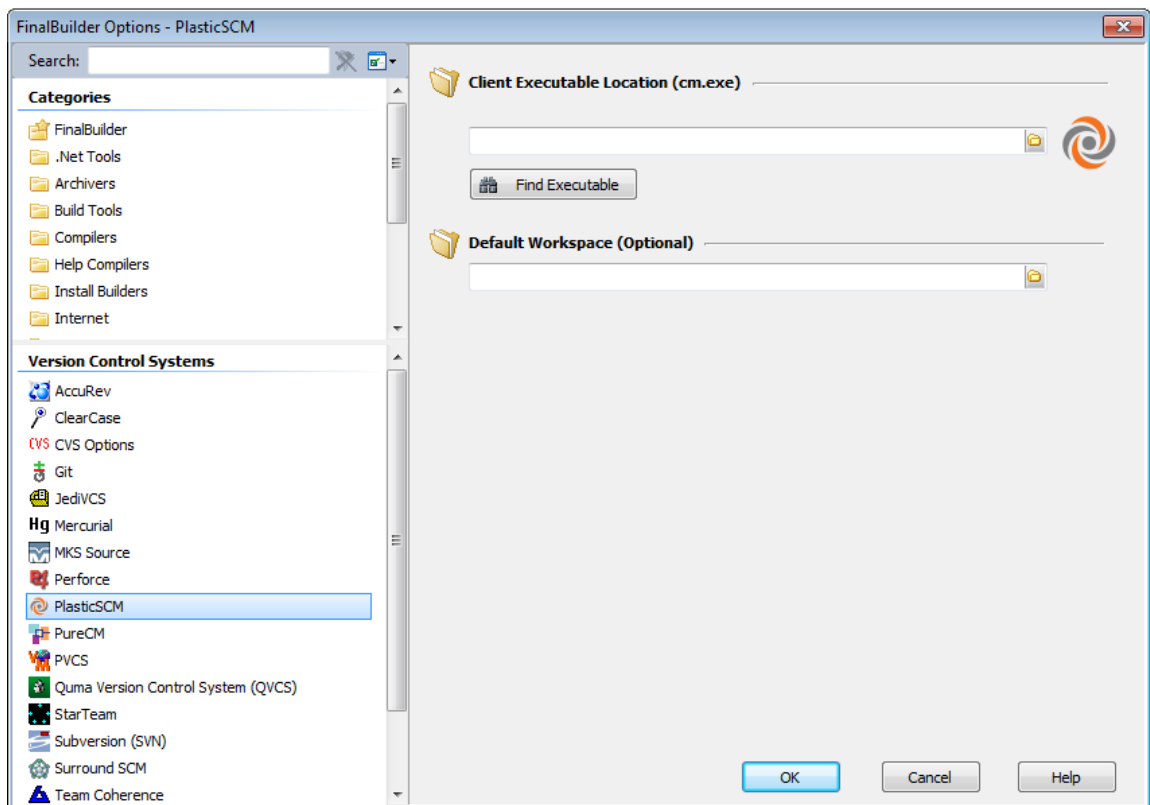
Before using the Plastic SCM actions you will need to specify the location of the client executable file (cm.exe) via the Options Page.

The actions available for Plastic SCM are:

- Plastic SCM Add New Items
- Plastic SCM Check In Items
- Plastic SCM Check Out Items
- Plastic SCM Configure Workspace
- Plastic SCM Create New Branch
- Plastic SCM Create Workspace
- Plastic SCM Generic
- Plastic SCM Label Workspace
- Plastic SCM Update

6.34.10.1 Plastic SCM Options

The Plastic SCM Options page is available via Tools -> Options -> Version Control Systems -> Plastic SCM.



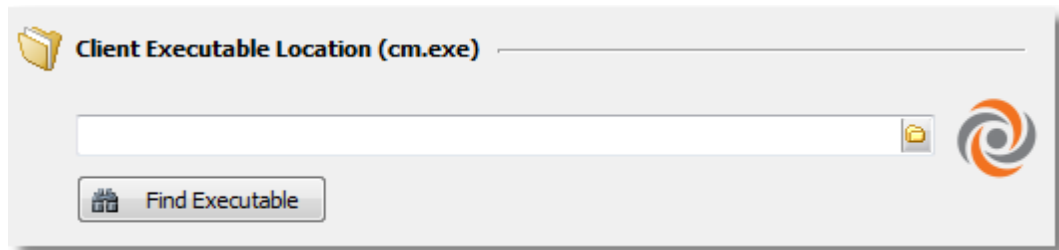
The Plastic SCM options page

Client Executable Location

The options page is where you provide the location of the the client executable (cm.

exe) which is required before running any Plastic SCM actions in Finalbuilder.

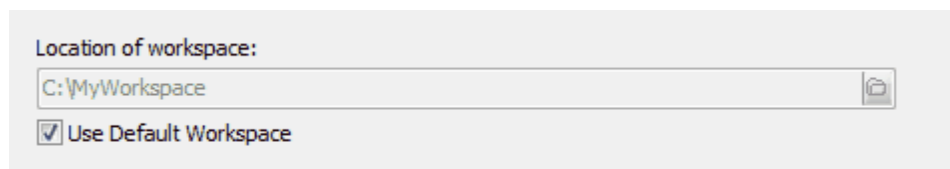
When you first load the options page if the executable field is blank the *Find Executable* button will be visible. When you click on this button it will attempt to find the location of the client executable. If the executable is found successfully, the field will be populated with the location of the executable and the button will be made invisible. If the attempt to find the executable failed a message will be displayed advising that the executable could not be found and will need to be specified manually, the button will also be disabled.



The Find Executable button is only visible when field is empty

Default Workspace (Optional)

The Plastic SCM options page also provides a means of specifying a default workspace. This is optional but can be used to save a significant amount of time when configuring and running actions. Each Plastic SCM action requires a workspace to be specified (for the action to run against) and provides an option to use the default workspace (the option is only available if a default has been specified). Once you have provided a default workspace on the options page you can simply select the *Use Default Workspace* option on the actions that you want to run against this workspace.



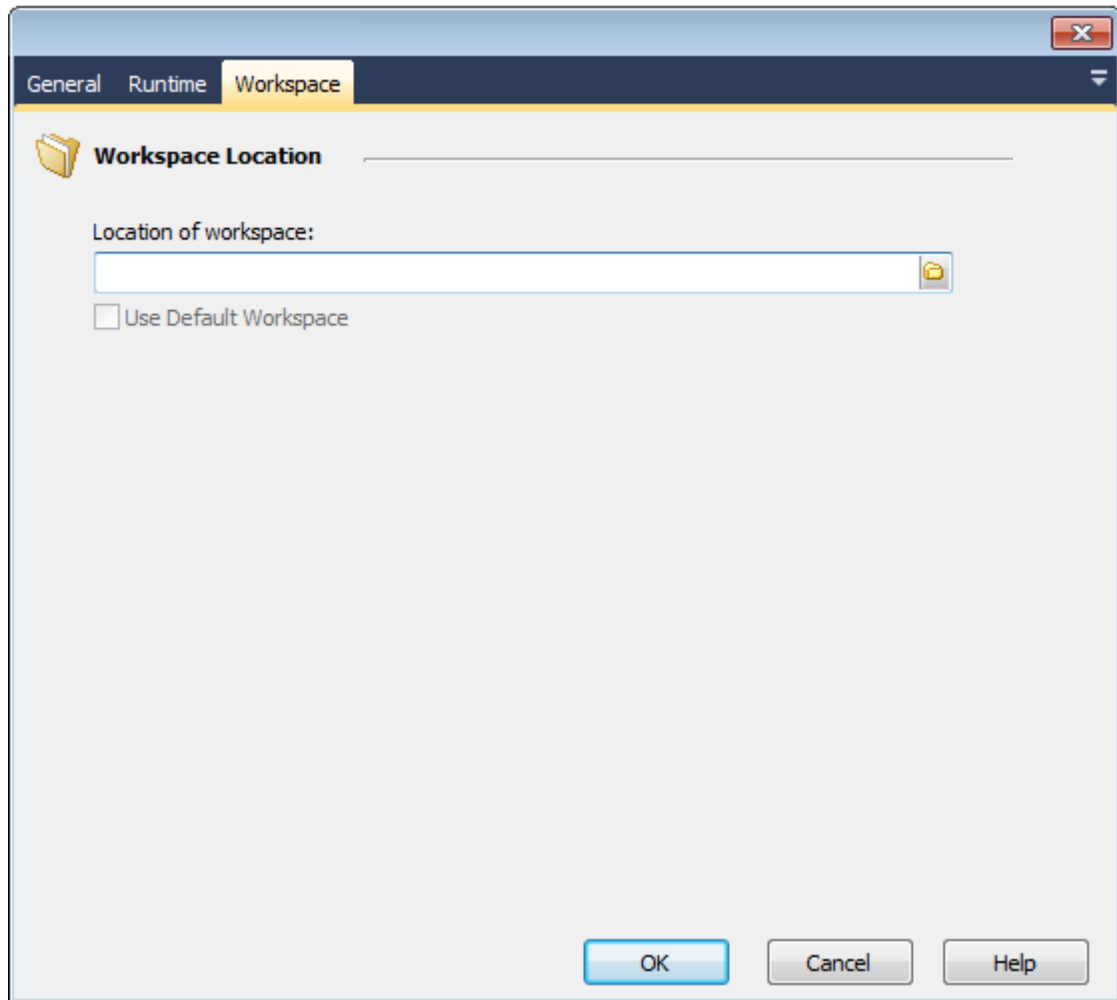
The Use Default Workspace option in use on a Plastic SCM action

The value from the default workspace is evaluated at run-time which means that you can specify a variable for the default workspace and update the value of the variable throughout the build process as need be.

6.34.10.2 Plastic SCM Add New Items

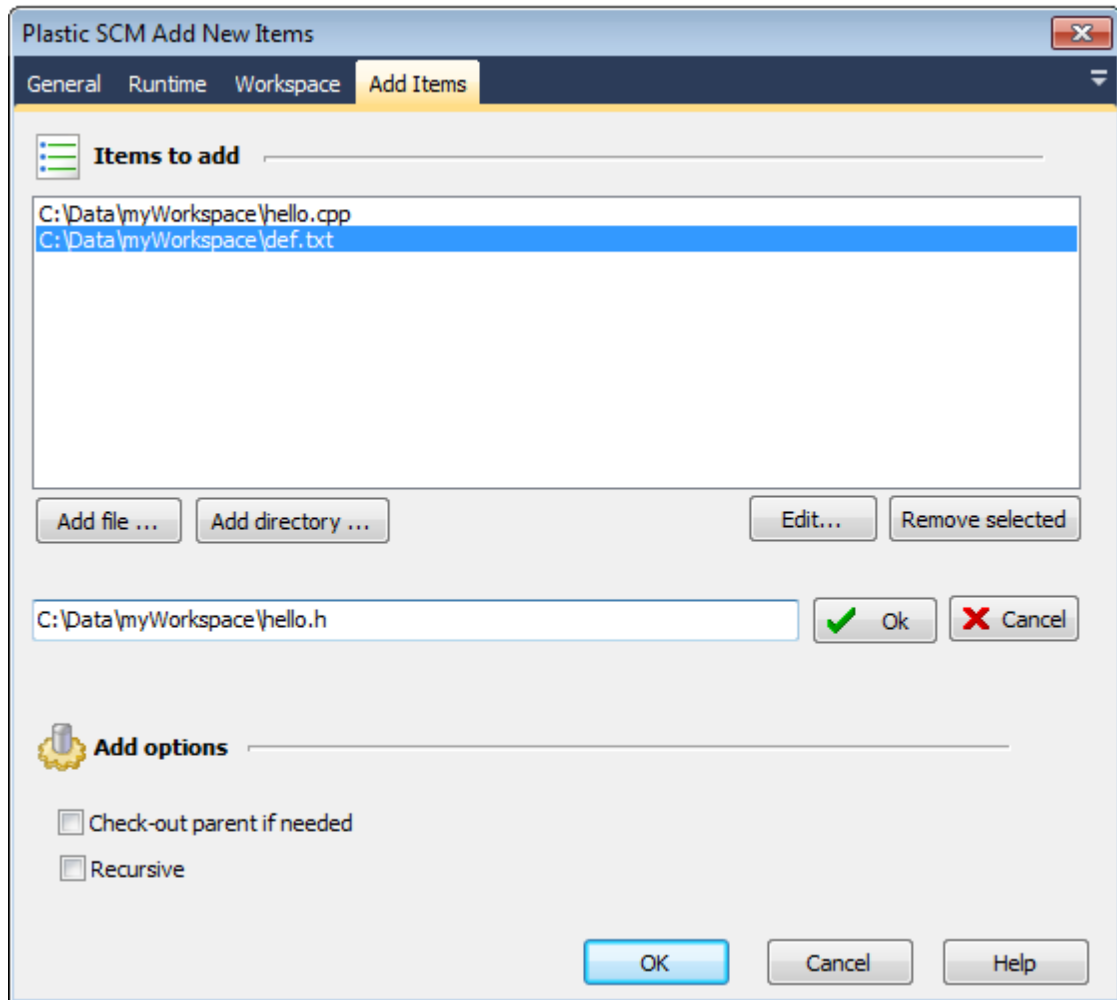
The Add New Items action allows you to add new items to a workspace.

First of all switch to the *Workspace* page and select the workspace which you want to add the items to (this field can be populated with the default workspace if you have specified one on the options page).



Next, choose the items to add to the selected workspace from the *Add Items* page. Add the files and directories via the controls provided.

Items that have been added to the list can be edited and removed with the corresponding controls.



There are also two options available for this action:

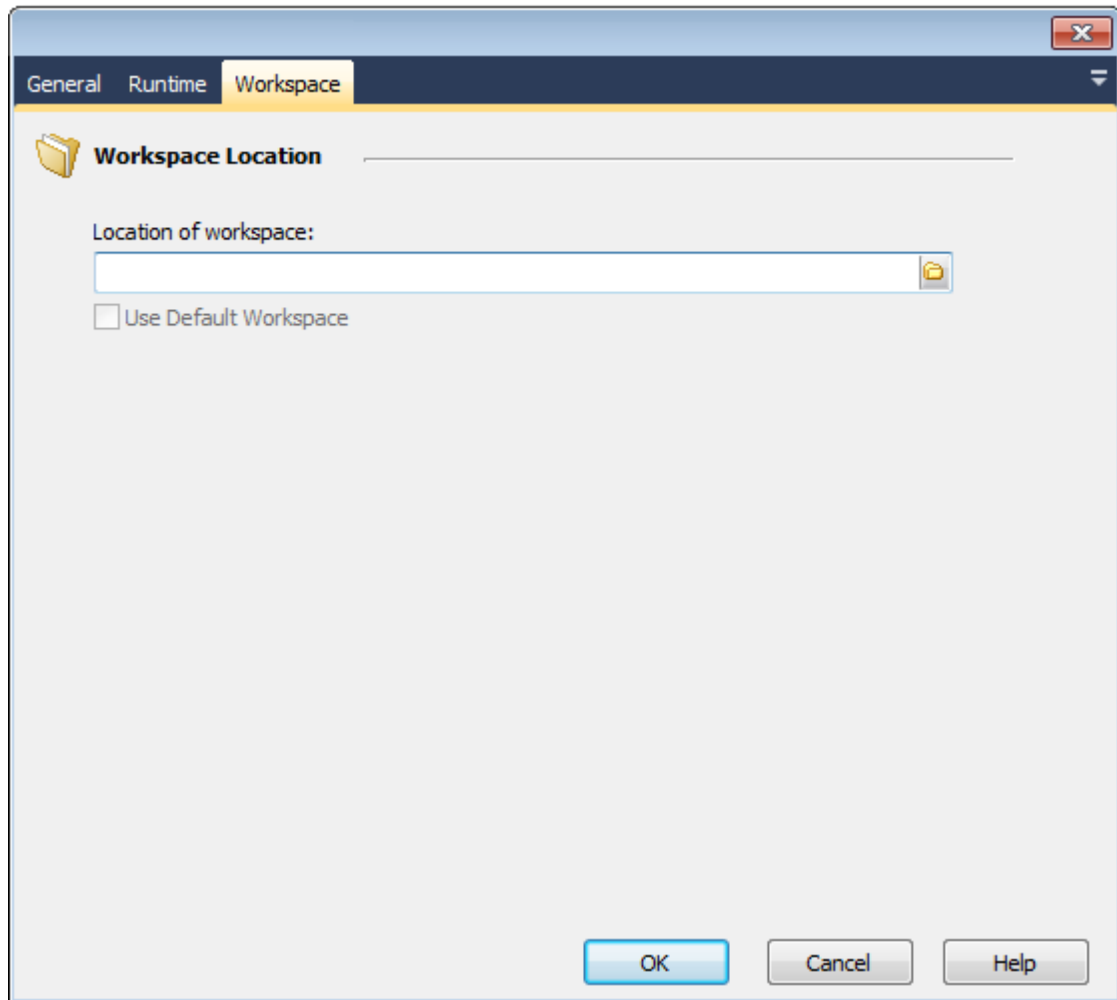
Check-out parent if needed - This automatically checks out the parent directory where required to add items.

Recursive - Recursively add item.

6.34.10.3 Plastic SCM Check In Items

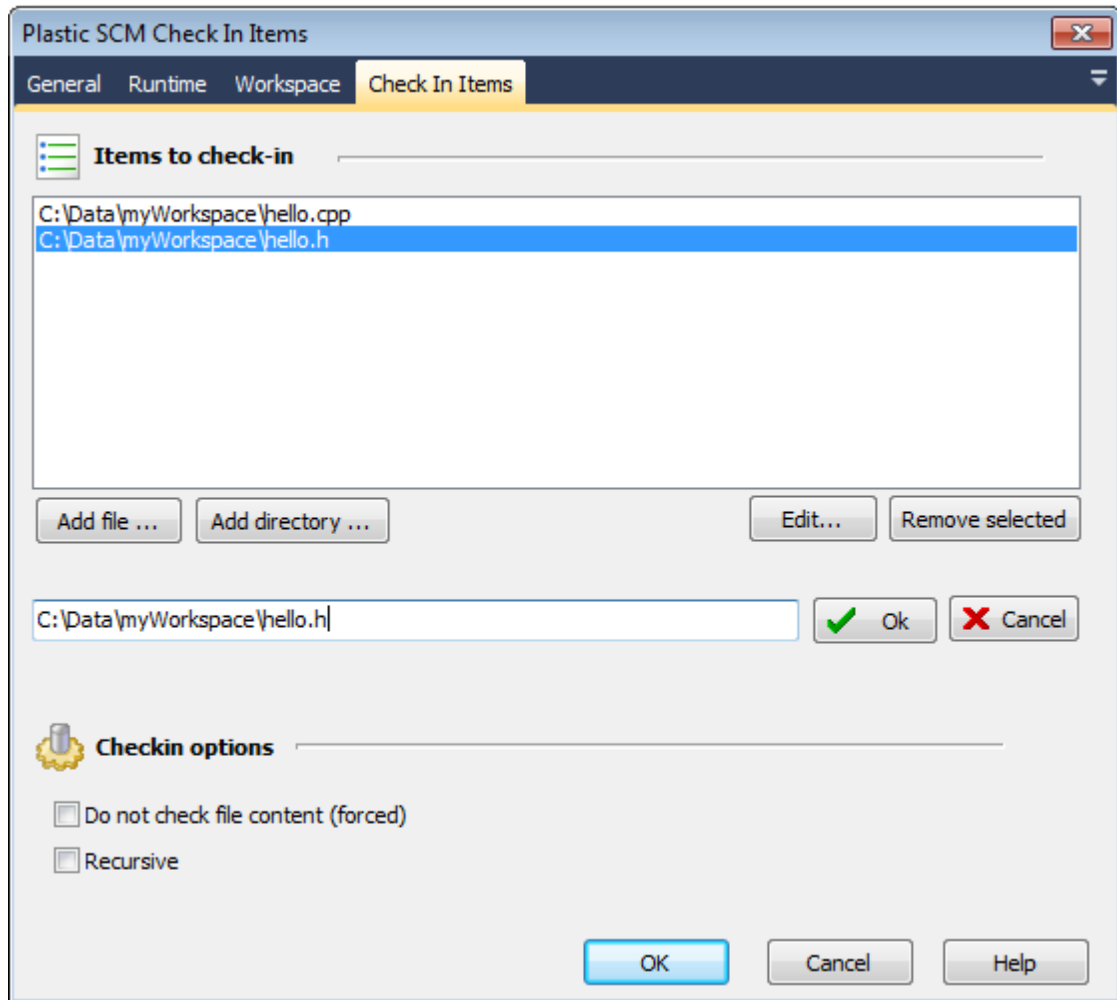
The Check In Item action allows you to create a new revision of changed items.

First of all switch to the *Workspace* page and select the workspace which you want to perform the check in on (this field can be populated with the default workspace if you have specified one on the options page).



Next, choose the items to check in to the selected workspace from the *Check In Items* page. Add the files and directories via the controls provided.

Items that have been added to the list can be edited and removed with the corresponding controls.



There are also two options available for this action:

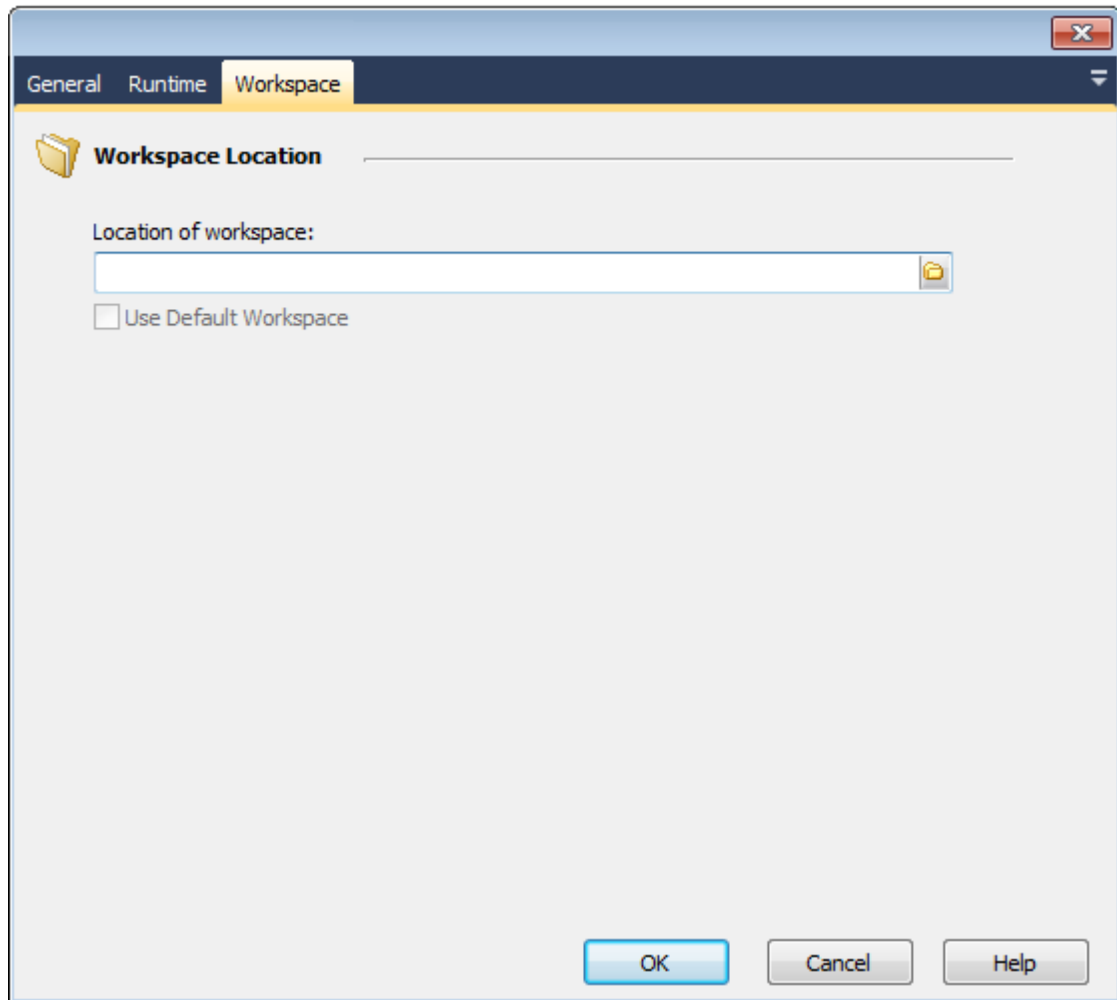
Do not check file content (forced) - Revision will be created even if changes have not been made since the last revision.

Recursive - Perform a recursive check in.

6.34.10.4 Plastic SCM Check Out Items

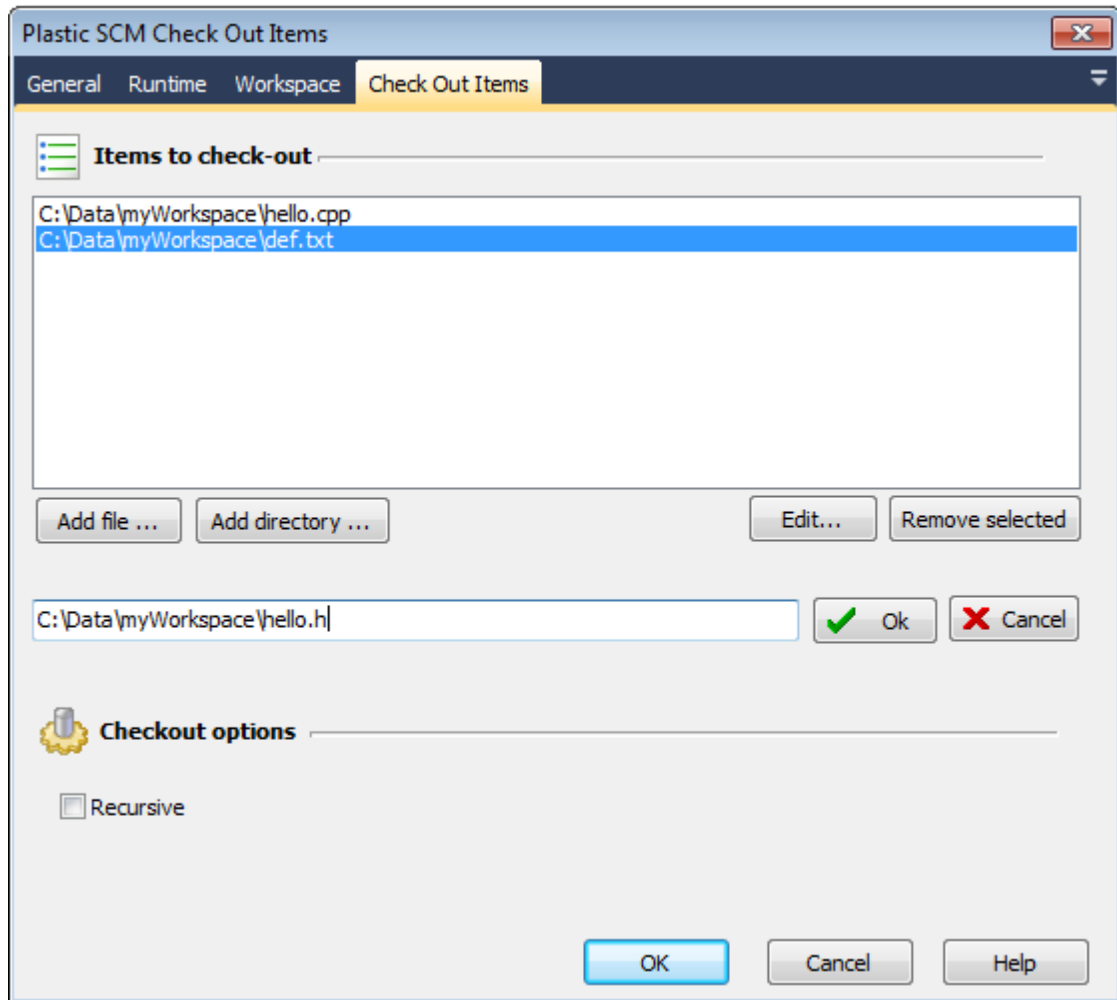
The Check In Item action allows you to check out items to allow modifications to be made.

First of all switch to the *Workspace* page and select the workspace which you want to perform the check out on (this field can be populated with the default workspace if you have specified one on the options page).



Next, choose the items to check out from the selected workspace via the *Check Out Items* page. Add the files and directories via the controls provided.

Items that have been added to the list can be edited and removed with the corresponding controls.



There is one option available for this action:

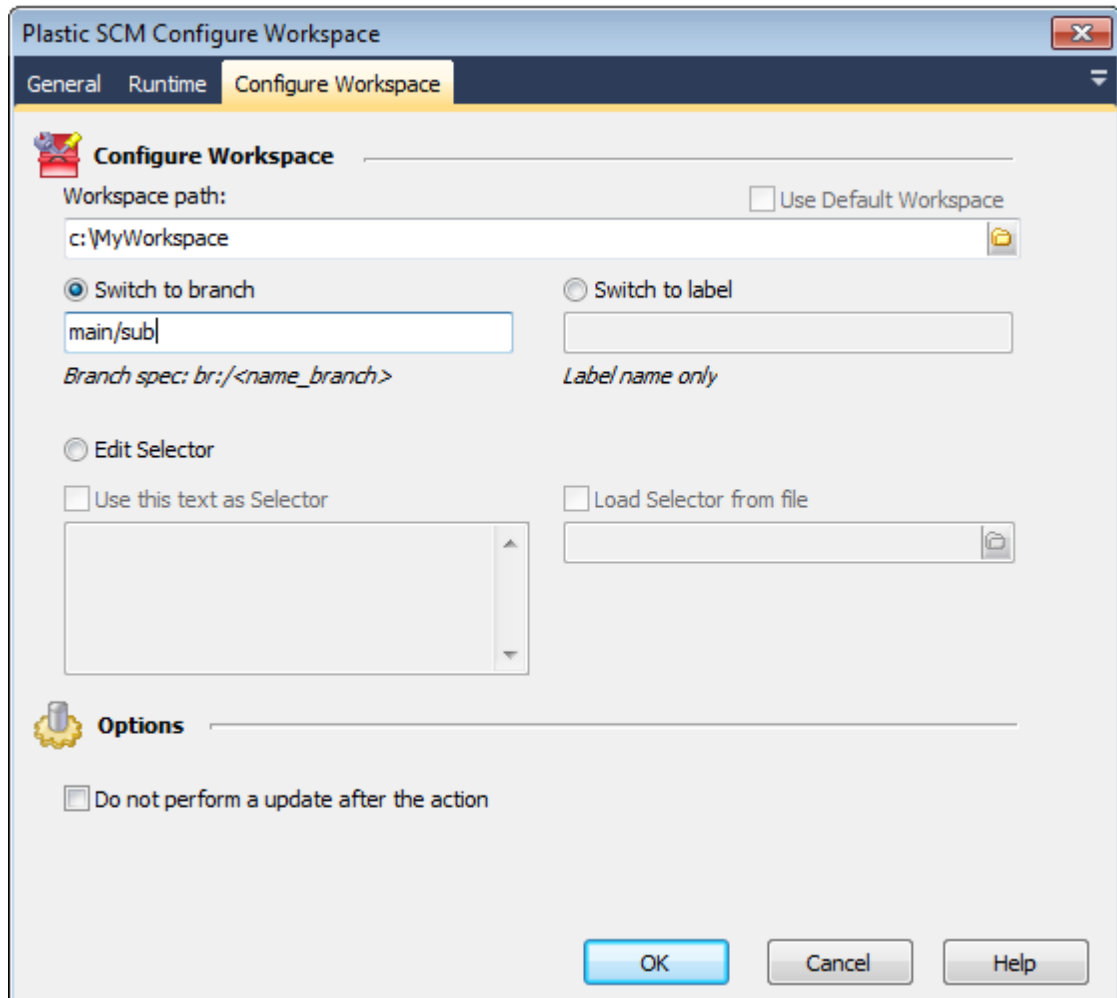
Recursive - Perform a recursive checkout.

6.34.10.5 Plastic SCM Configure Workspace

The Configure Workspace action allows you to:

- Switch to a specific branch within a workspace
- Switch to a specific label within a workspace
- Modify the workspace selector

First of all specify the workspace that you want to configure (this field can be populated with the default workspace if you have specified one on the options page).



Now choose which modification you would like to make to the workspace:

Switch to branch

1. Ensure the *Switch to branch* radio button is selected which will enable the corresponding text field.
2. Enter the name of the branch that you want to switch to within the workspace.

Switch to label

1. Ensure the *Switch to label* radio button is selected which will enable the corresponding text field.
2. Enter the name of the label that you want to switch to within the workspace.

Edit Selector

1. Select the *Edit Selector* radio button which will enable the checkboxes for the

modification options.

2. Select the modification option you wish to use (you can only select one):

- *Use this text as Selector* - allows you to type the selector text into the text field provided which will be used as the workspace selector.
- *Load Selector from file* - allows you to provide a file which contains the selector text.

Note: With either these options a mistake in the selector syntax will result in the action failing.

There is one option available for this action:

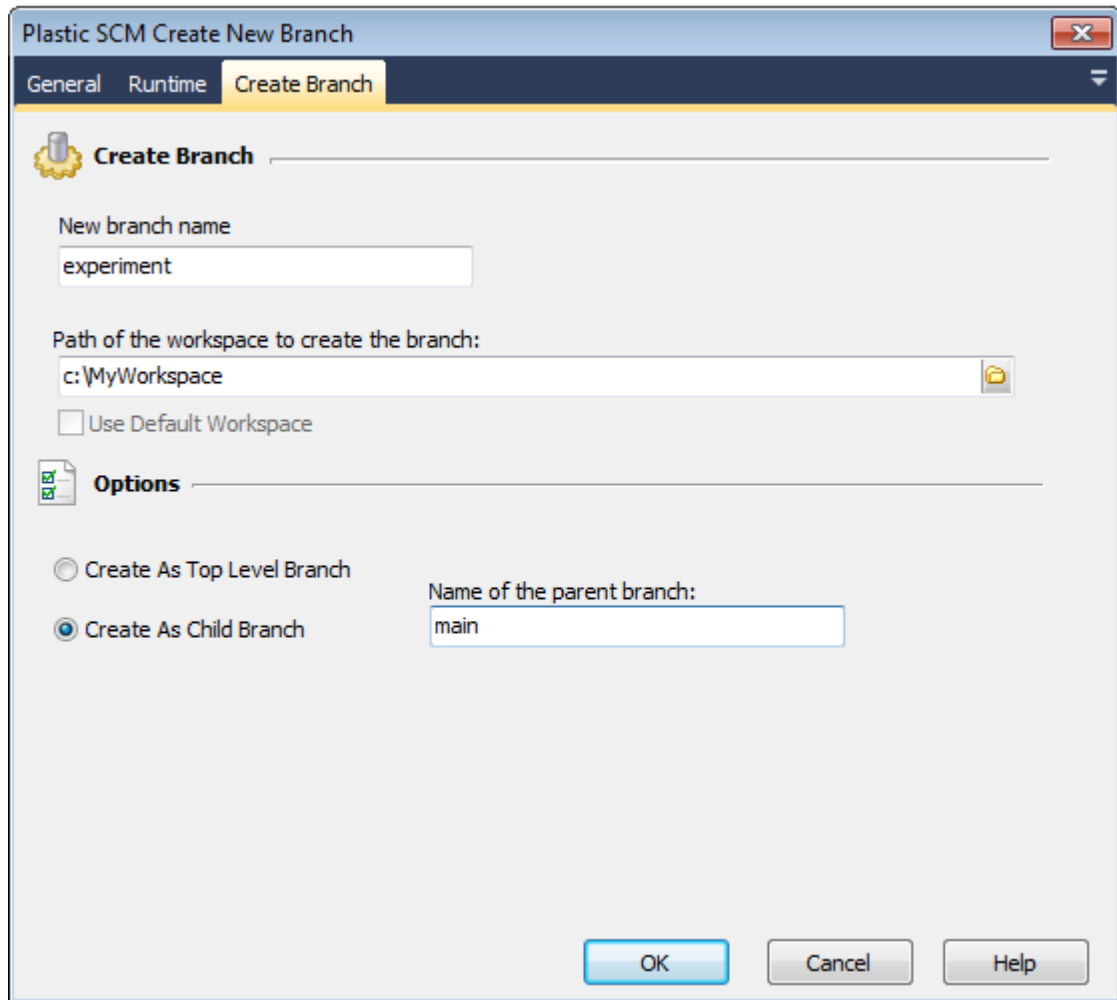
Do not perform update after the action - This prevents an automatic update of the workspace occurring after the action has run.

6.34.10.6 Plastic SCM Create New Branch

The Create New Branch action allows you to create a new branch in the current repository.

Enter a value for the New Branch Name in the field provided.

Specify the workspace that you are using (this field can be populated with the default workspace if you have specified one on the options page).



Under the options section you can specify whether to create the new branch as a top level branch or as a child branch of a branch that already exists. When creating the new branch as a child branch you must specify the name of the parent branch.

When creating a branch that is a child you must specify the complete reference of the parent branch.

For example:

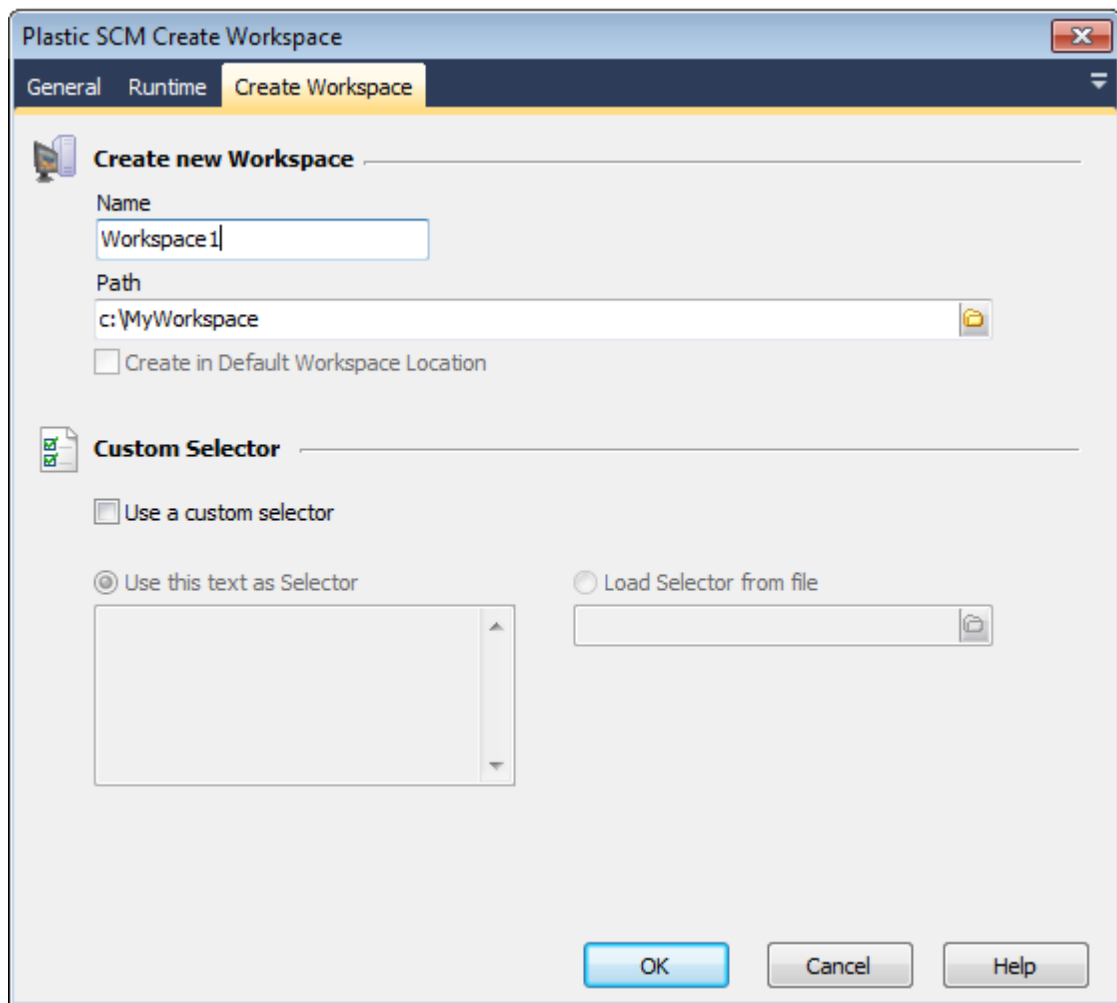
To create a child branch called *child2* which is a child of the *main/child1* branch you would need to specify *main/child1* (as opposed to simply *child1*) in the parent branch field. Without the complete parent branch reference the action will fail.

6.34.10.7 Plastic SCM Create Workspace

The Create Workspace action allows you to create a new local workspace.

First provide a name for the new local workspace.

Provide a location for the workspace on the local file system (this field can be populated with the default workspace if you have specified one on the options page).



The Custom Selector options allow you to specify a custom selector for the new workspace.

1. *Use this text as Selector* - allows you to type the selector text into the text field provided which will be used as the workspace selector.
2. *Load Selector from file* - allows you to provide a file which contains the selector text.

Note: With either of these options a mistake in the selector syntax will result in the action failing.

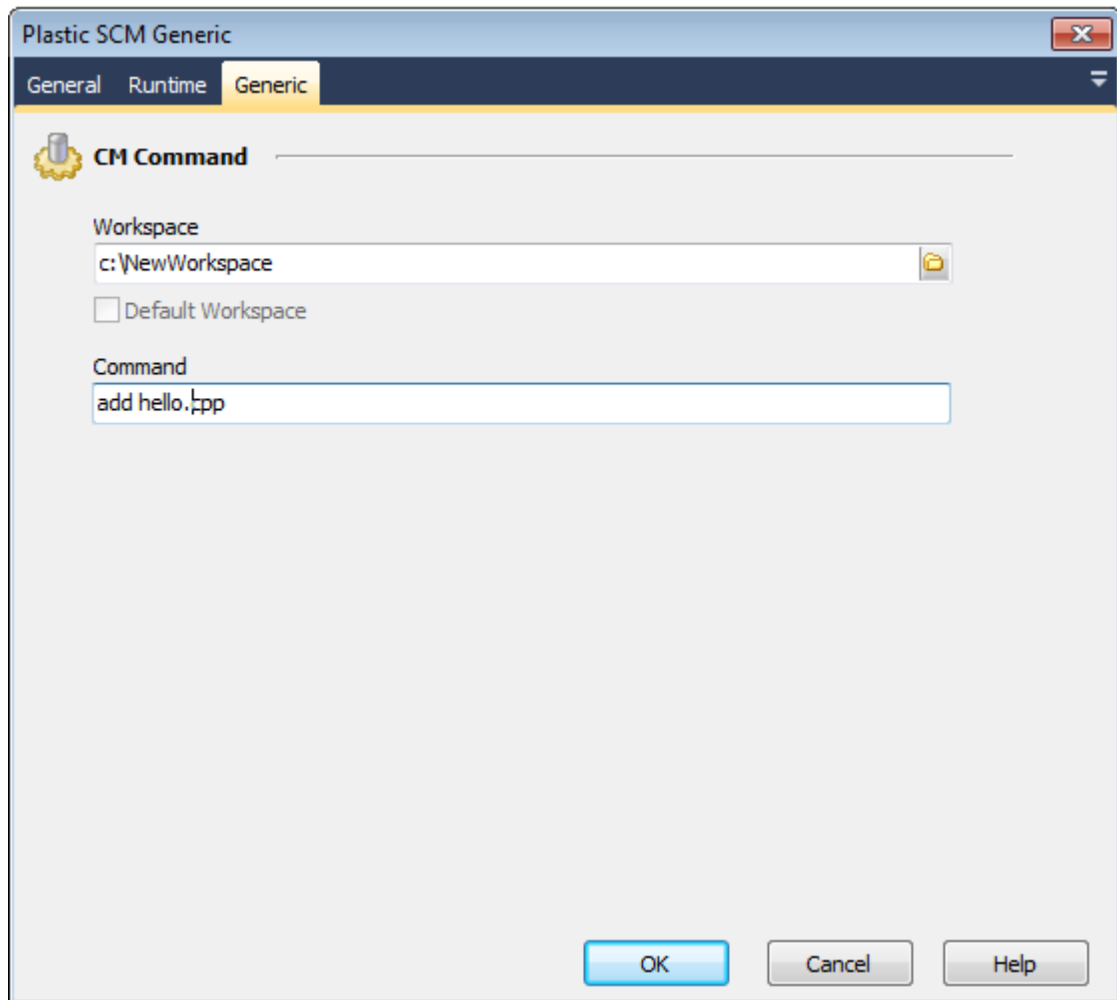
6.34.10.8 Plastic SCM Generic

The Generic action allows you to run a Plastic SCM command as you would from the command line.

Provide a value for the workspace location (this field can be populated with the default

workspace if you have specified one on the options page).

In the command field type a command as you would from the command line.

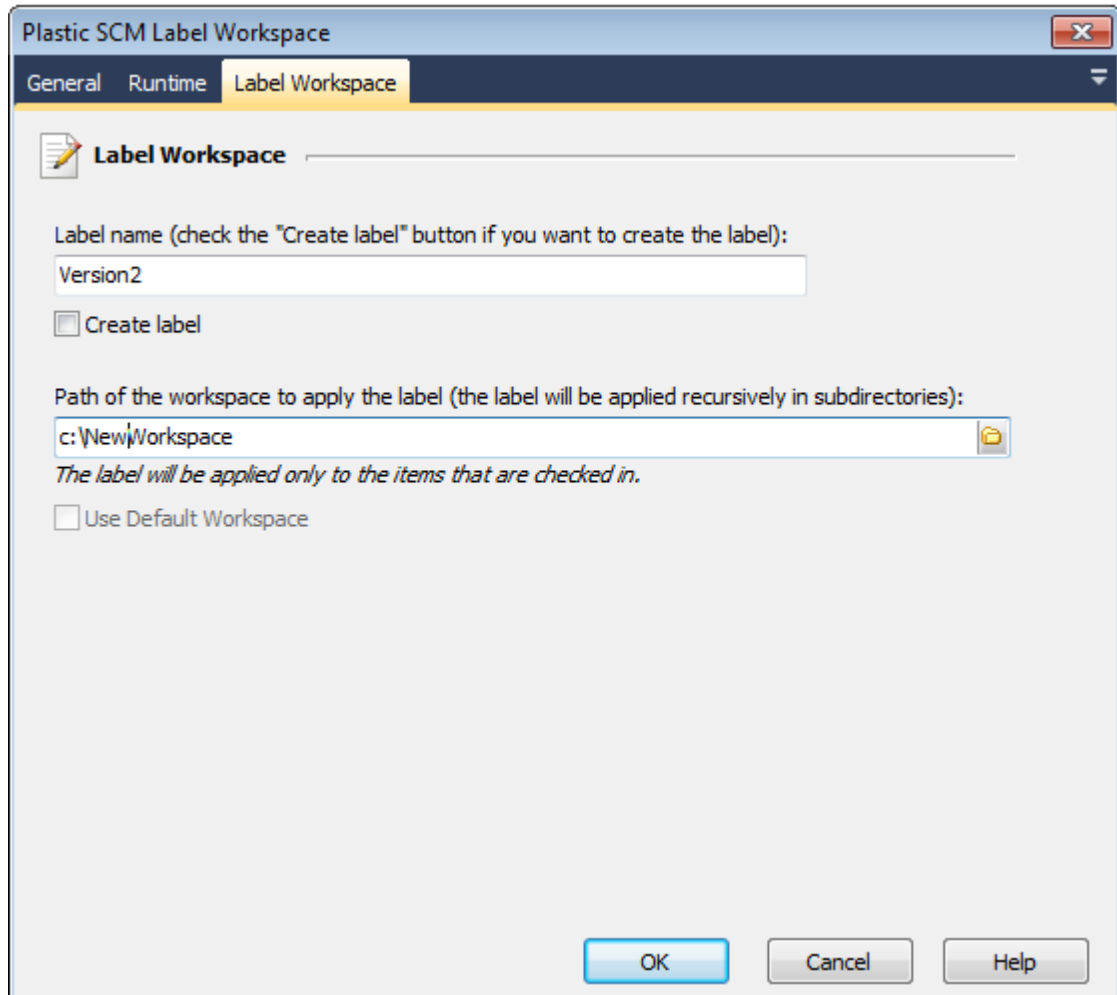


For example to run *cm status* you can either type *cm status* or simply *status* (either will be accepted). Provide any arguments to the command as you usually would.

Note: Any errors in the command syntax will result in the action failing.

6.34.10.9 Plastic SCM Label Workspace

The Label Workspace action allows you to create and apply labels to a workspace. The label is applied recursively to all items within the workspace.



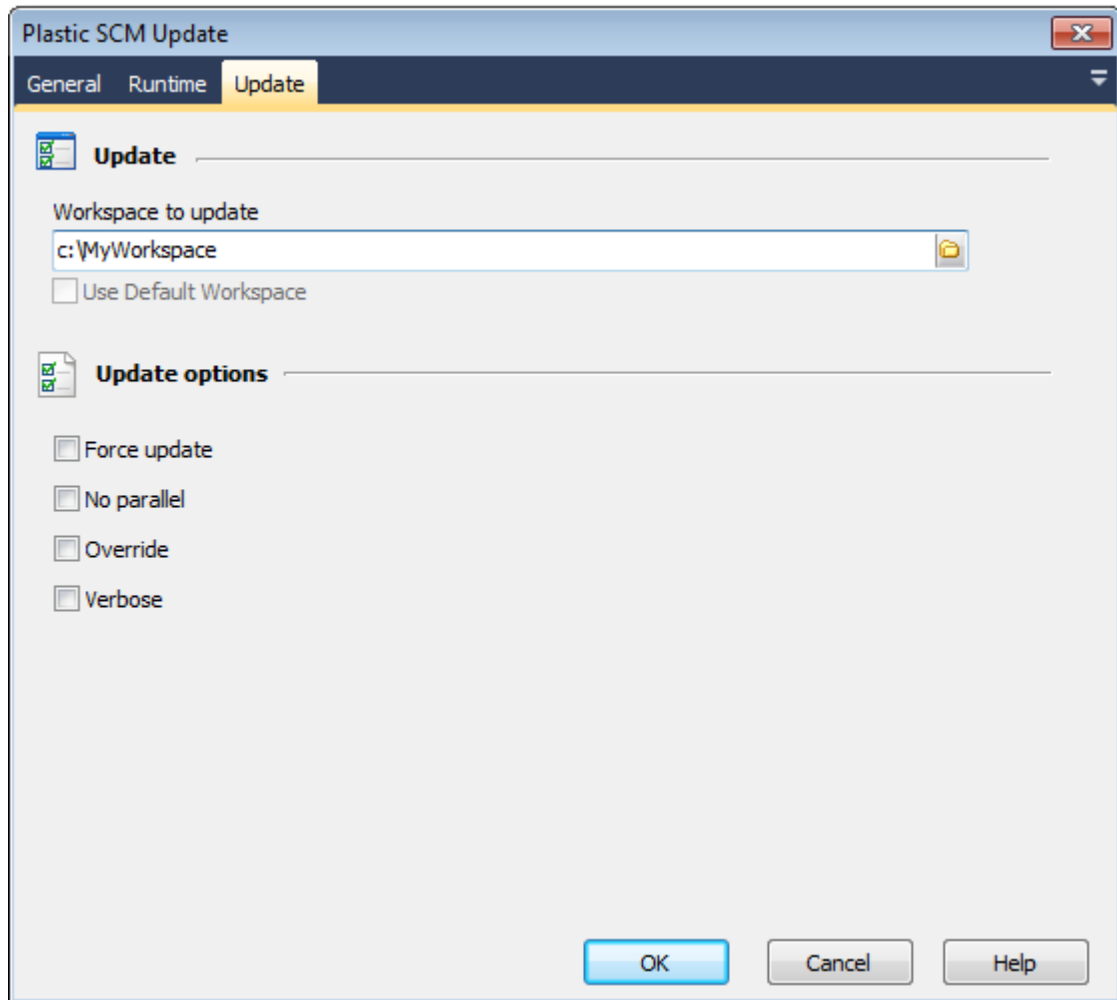
In the *Label name* field enter the name of the label to be applied to the workspace. If this label does not yet exist you will need to select the *Create label* option (this will create the label before applying it to the workspace).

Provide the path of the workspace to apply the label to (this field can be populated with the default workspace if you have specified one on the options page).

6.34.10.1(Plastic SCM Update

The Update action allows you to update items in a workspace.

Provide a value for the Workspace to update (this field can be populated with the default workspace if you have specified one on the options page).



The action provides the following options:

Force Update - Forces updating items to the revision specified in the selector.

No parallel - This option will use only a single thread to update the files. This can result in improved performance when updating over a slow network connection.

Override - Overrides changed files outside of Plastic SCM control.

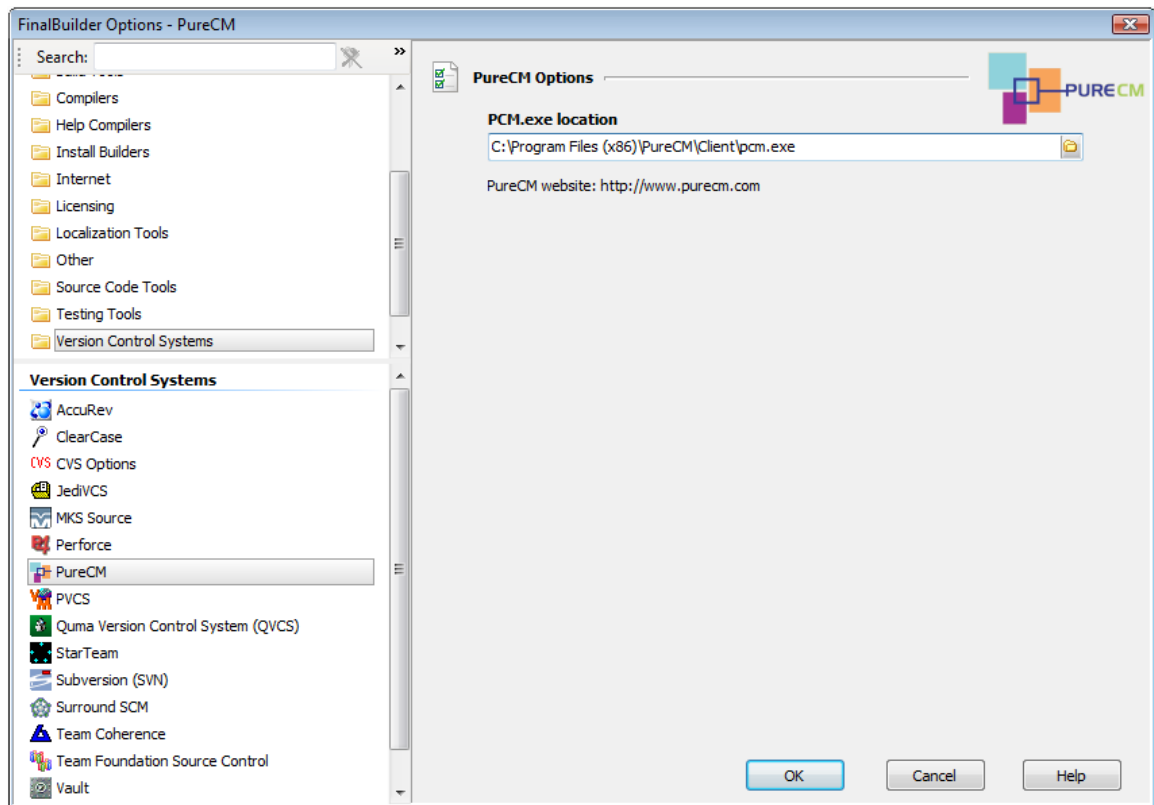
Verbose - Verbose output.

6.34.11 PureCM

The PureCM actions support PureCM 2007-2c and above.

These actions were written by PureCM.com Ltd, and are maintained by VSoft Technologies Pty Ltd. For more information on PureCM, please see <http://www.purecm.com>

6.34.11.1 PureCM Options

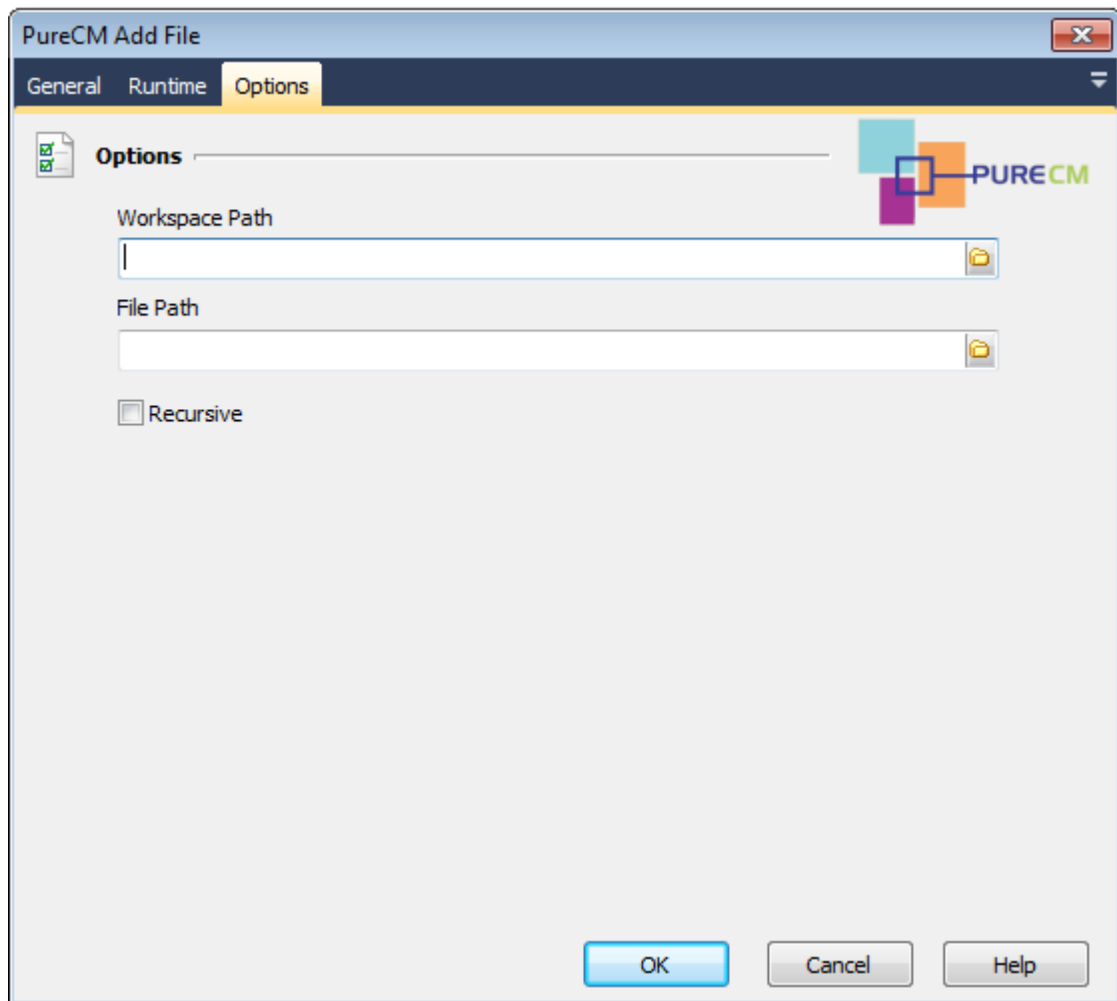


PCM.exe location

This is the path of the PureCM command line, that needs to be run to perform the PureCM actions. This will be found in the directory in which you installed the PureCM client.

6.34.11.2 PureCM Add File

The PureCM Add File action enables you to add a file or folder to your PureCM workspace.



Workspace Path

The local path to the workspace you wish to add the file to.

File Path

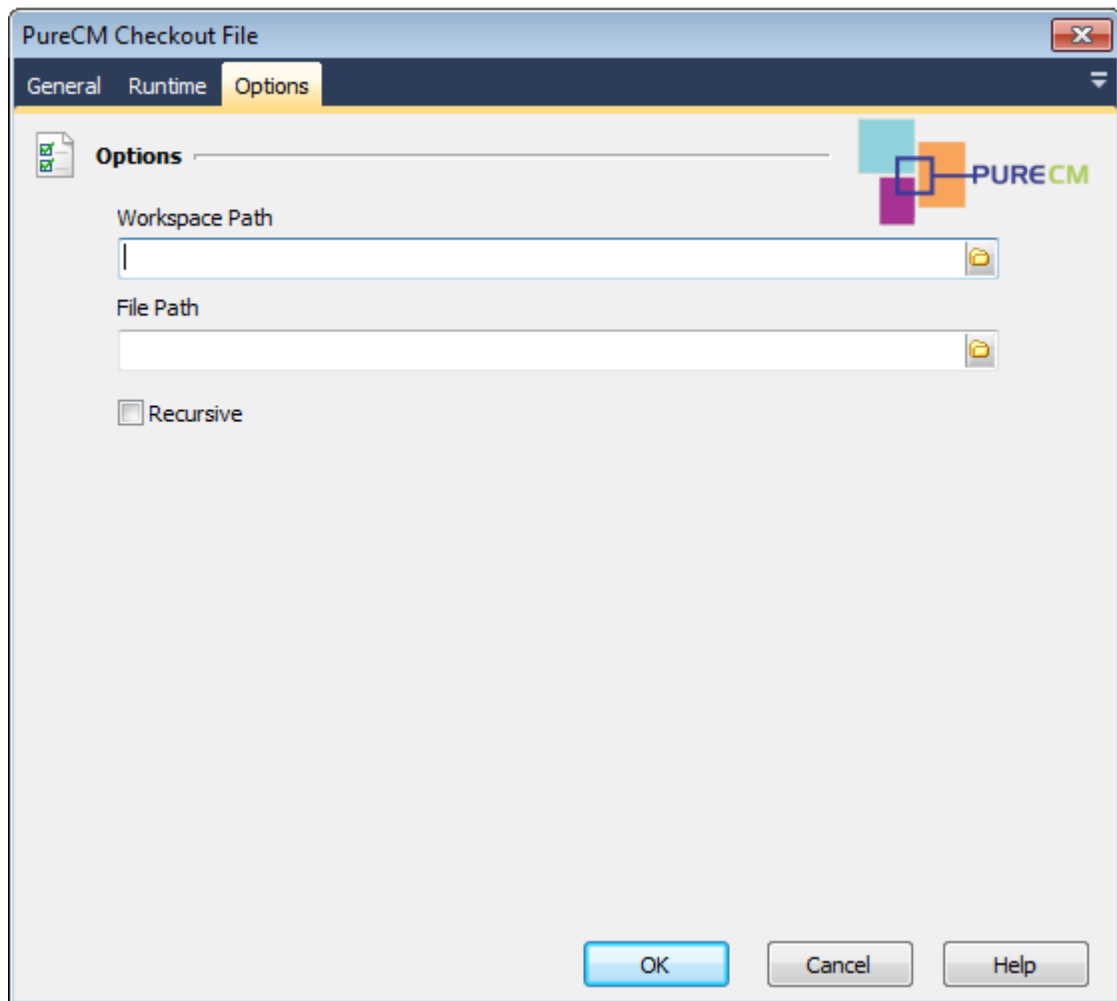
The path to the file or folder which you are adding. This file must be within your PureCM workspace.

Recursive

If you are adding a folder, the recursive option will also add all of the sub folders.

6.34.11.3 PureCM Checkout File

The PureCM Checkout File action enables you to edit a file within your Workspace.



Workspace Path

The local path to the workspace you wish to checkout the file in.

File Path

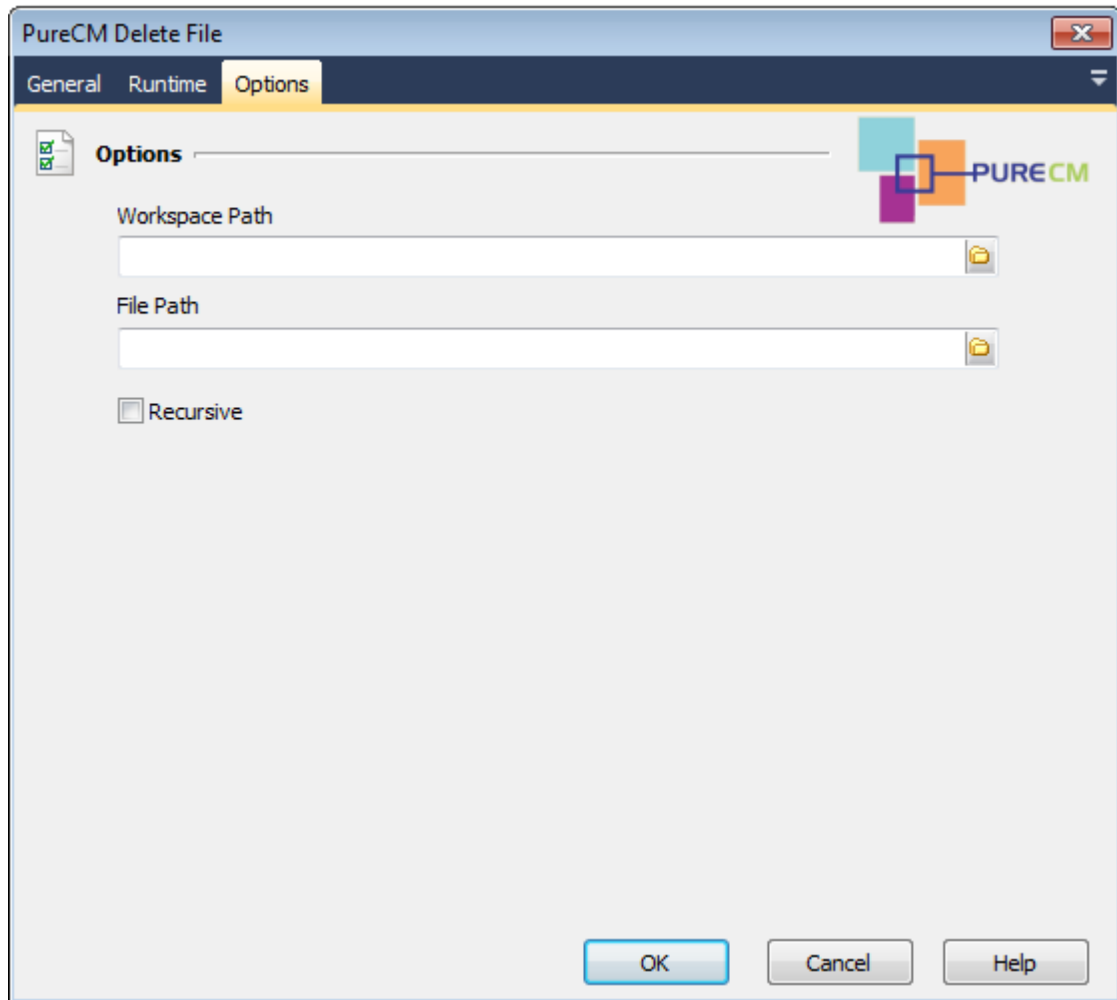
The path to the file or folder which you are checking out. This file must be within your PureCM workspace.

Recursive

If you are checking out a folder, the recursive option will also checkout all of it's sub folders.

6.34.11.4 PureCM Delete File

The PureCM Delete File action enables you to delete a file from your hard disk and remove it from your PureCM workspace.



Workspace Path

The local path to the workspace you wish to delete the file from.

File Path

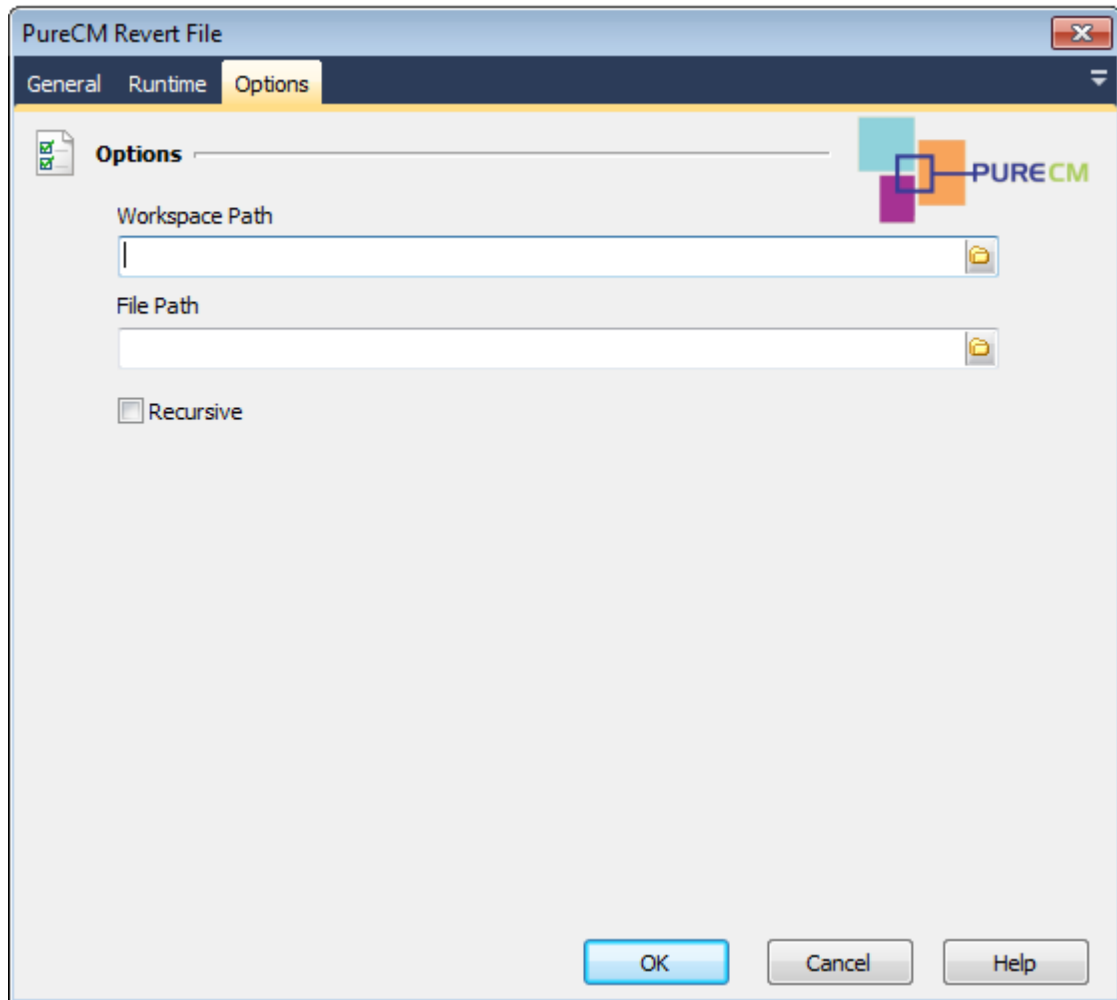
The path to the file or folder which you are deleting. This file must be within your PureCM workspace.

Recursive

If you are deleting a folder, the recursive option will also delete all of its sub folders. If the recursive option is not checked all files will be deleted from the selected folder. If the selected folder contains any sub folders it will not be deleted.

6.34.11.5 PureCM Revert File

The PureCM Revert File action enables you to revert all the changes made to a file within your workspace back to the server version and undo its checkout.

**Workspace Path**

The local path to the workspace you wish to revert the file in.

File Path

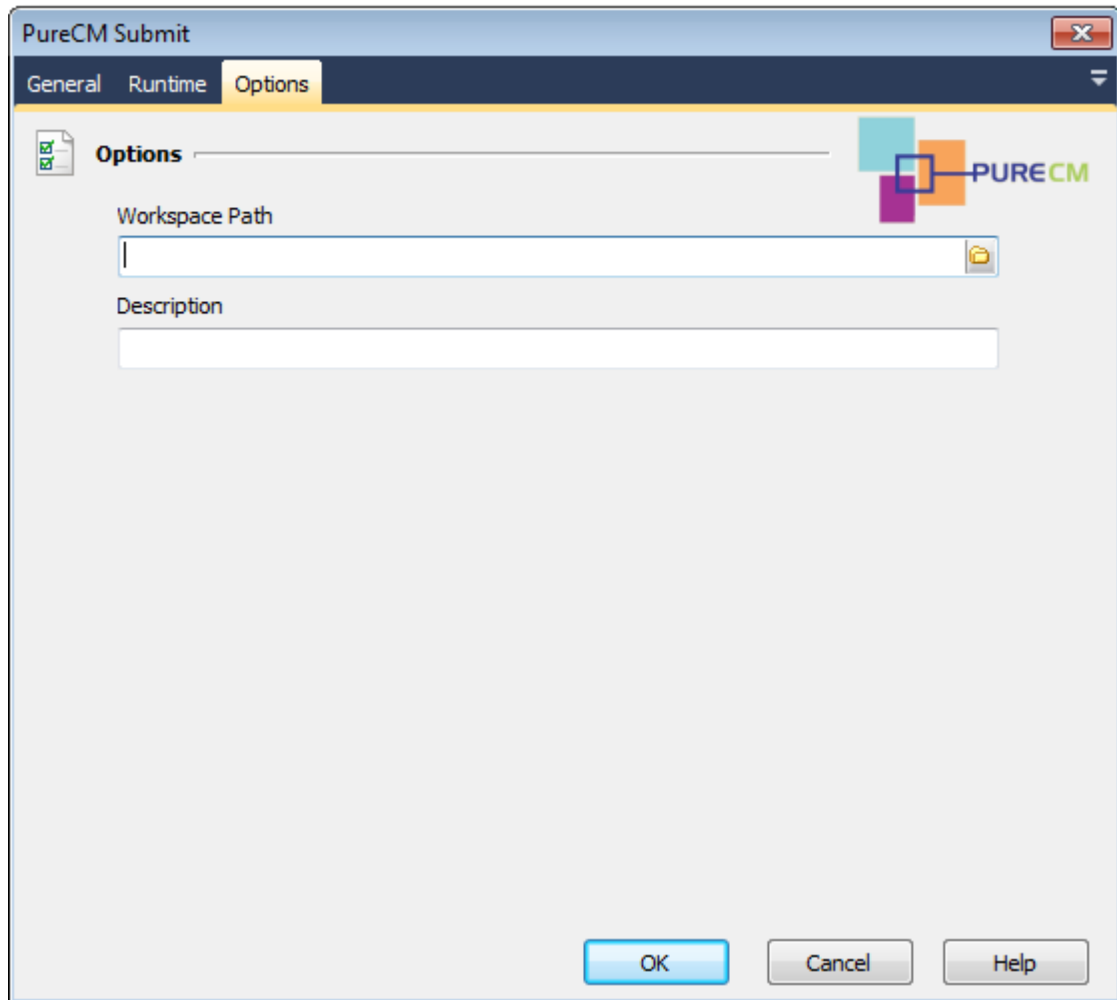
The path to the file or folder which you are reverting. This file must be within your PureCM workspace.

Recursive

If you are reverting a folder, the recursive option will also revert changes from it's sub folders.

6.34.11.6 PureCM Submit

The PureCM Submit action enables you to send all of the changes in your PureCM workspace to the server.

**Workspace Path**

The local path to the workspace you wish to submit your changes from.

Description

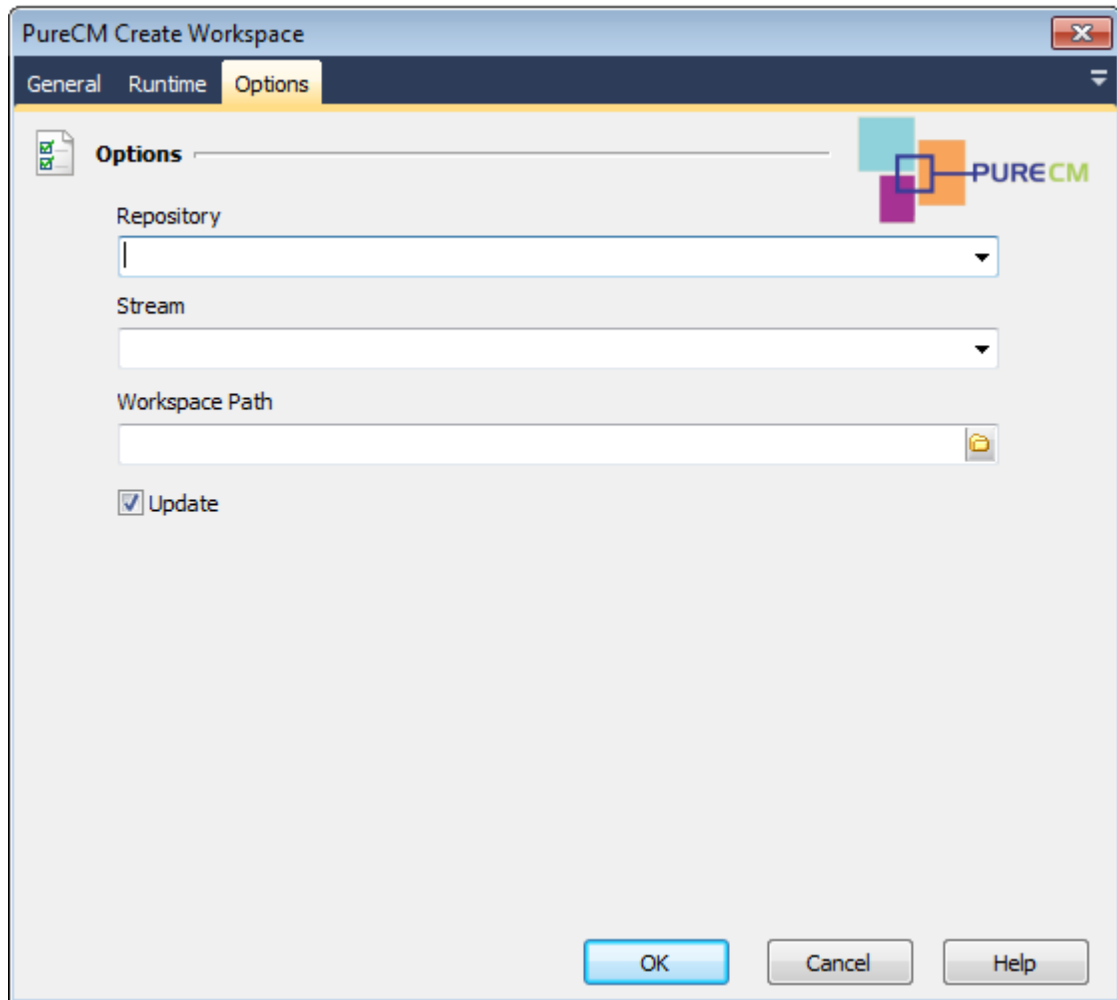
The description you wish to give to your changeset.

Note

If you wish to submit your changes with an associated issue from FinalBuilder you can achieve this by using the PureCM Generic action to call “submit” and creating an input file.

6.34.11.7 PureCM Create Workspace

The PureCM Create Workspace action enables you to make a new PureCM workspace to allow you to make changes to a stream.



Repository

The repository that you wish to create your workspace inside. You can either type in the repository name or select from a list of the current repositories.

Stream

The stream that you wish to create a workspace of. You can either type in the stream name or select from a list of all available streams inside your selected repository.

Workspace Path

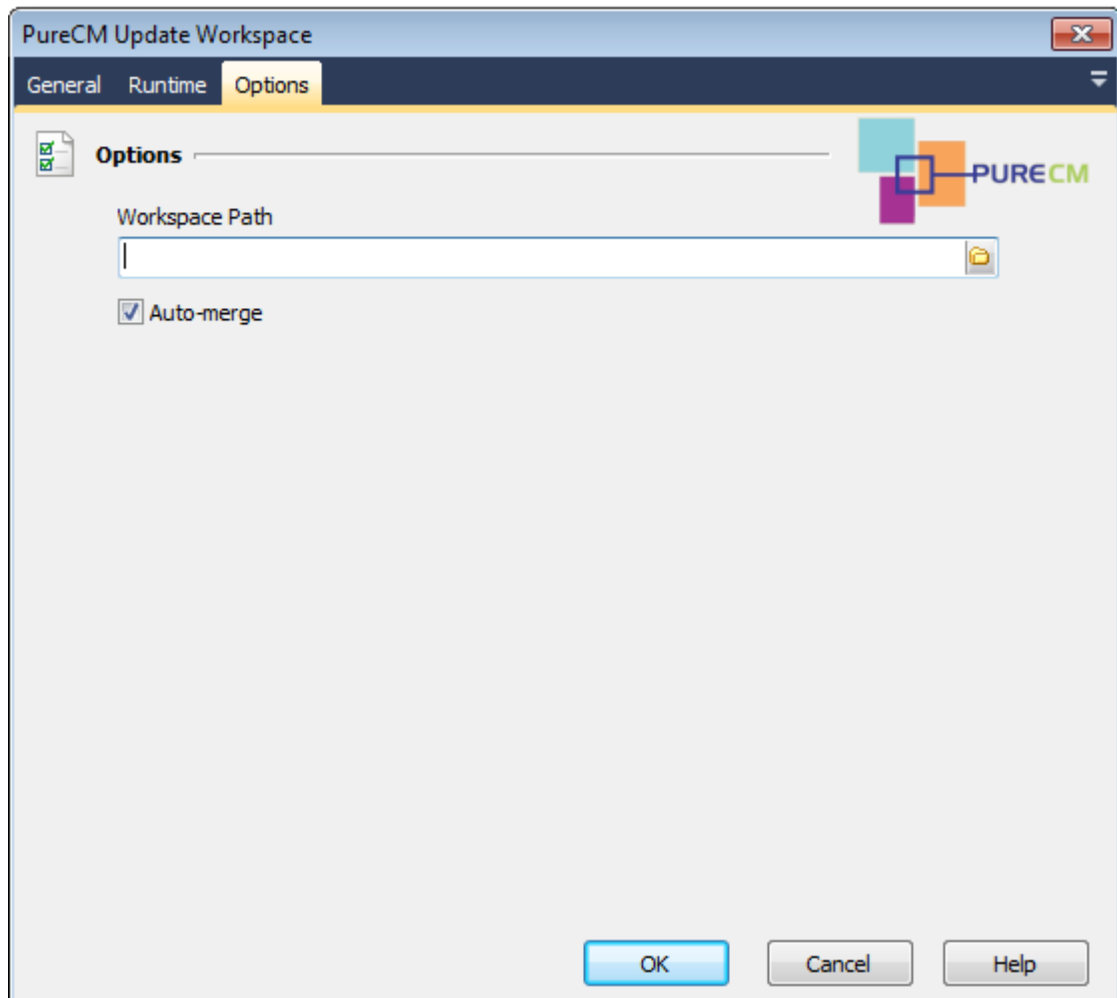
The local path to the workspace you are trying to create.

Update

The update option will download all of the data from the server. This is generally the default operation unless you wish to create the workspace and download the files at a later time. (Files can be downloaded later by using the 'PureCM Update Workspace' action.)

6.34.11.8 PureCM Update Workspace

The PureCM Update Workspace action enables you to download all the changes made to the server to your PureCM workspace.



Workspace Path

The local path to the workspace you are trying to update.

Auto-Merge

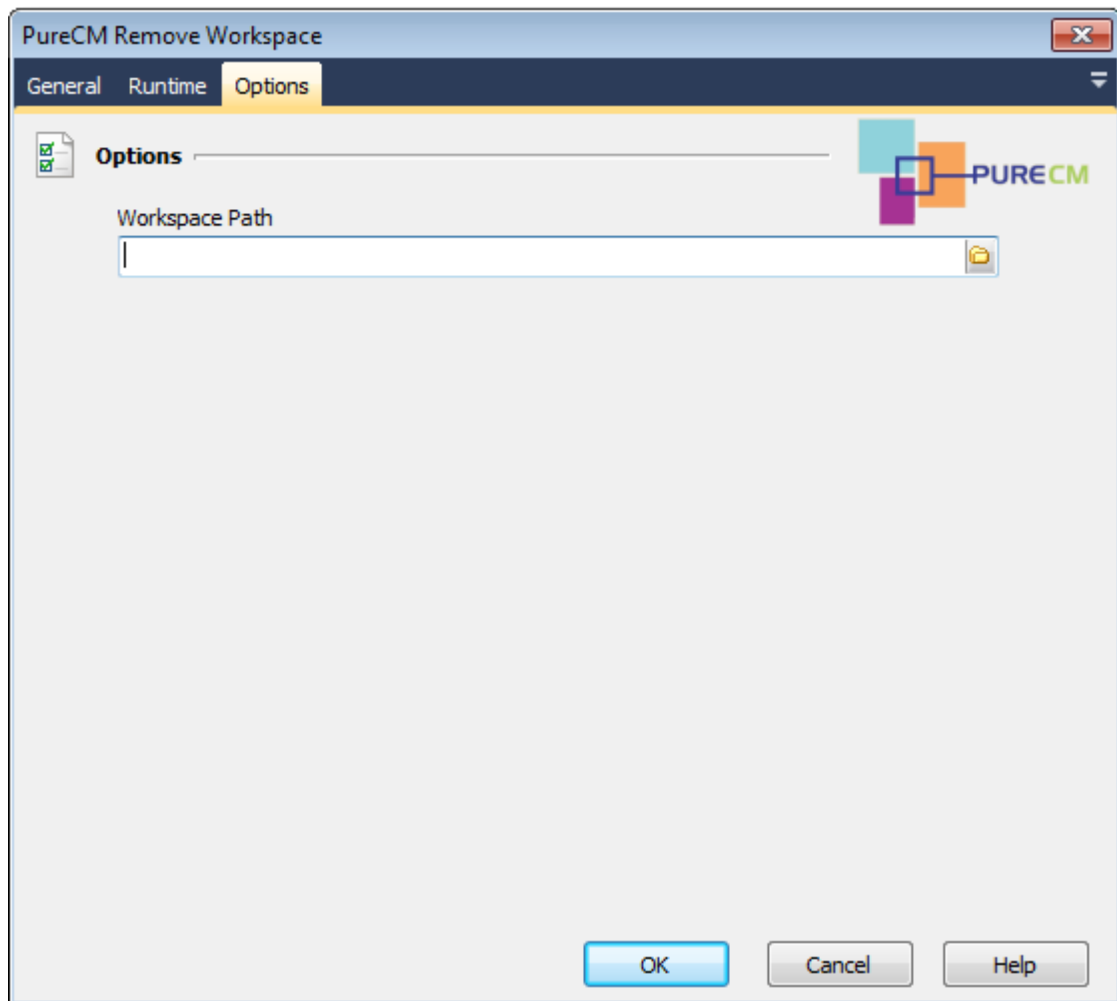
With the auto-merge option selected non-conflicting edits will be merged with the changes in your workspace.

Note

Any conflicting changesets will need to be resolved through either of the PureCM or pcm tools.

6.34.11.9 PureCM Remove Workspace

The PureCM Remove Workspace action enables you to remove a PureCM workspace.

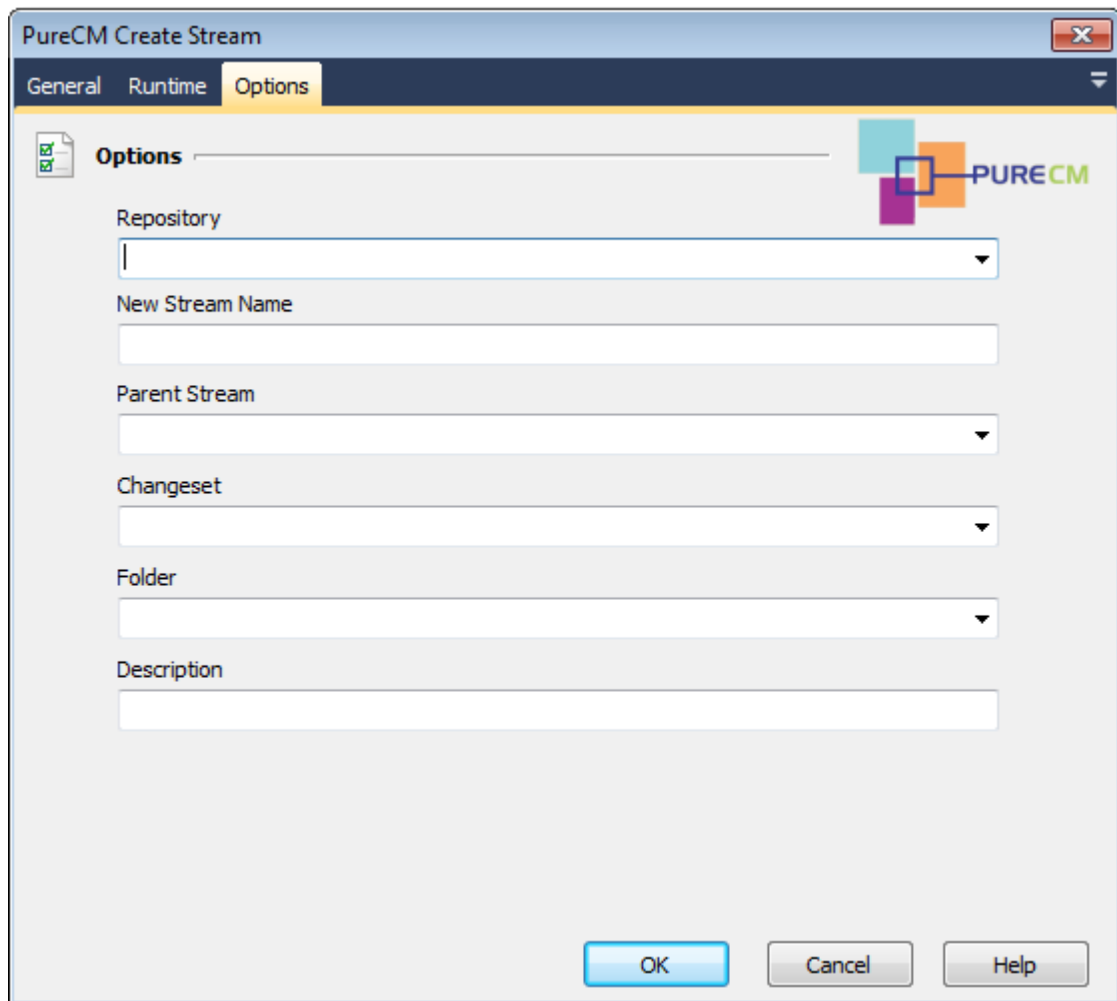


Workspace Path

The local path to the workspace you are trying to remove.

6.34.11.1 PureCM Create Stream

The PureCM Create Stream action enables you to make a new stream within PureCM.



The screenshot shows the 'PureCM Create Stream' dialog box with the 'Options' tab selected. The dialog has a title bar with a close button. Below the title bar are three tabs: 'General', 'Runtime', and 'Options'. The 'Options' tab is active and contains a 'PURECM' logo in the top right corner. The main area of the tab contains several input fields: 'Repository' (a dropdown menu), 'New Stream Name' (a text box), 'Parent Stream' (a dropdown menu), 'Changeset' (a dropdown menu), 'Folder' (a dropdown menu), and 'Description' (a text box). At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Repository

The repository that you wish to create your stream inside. You can either type in the repository name or select from a list of the current repositories.

New Stream Name

The name of the new stream you want to create.

Parent Stream

The stream that you wish to base the new stream on. You can either type in the stream name or select from a list of all available streams inside your selected repository.

Changeset

The changeset of the parent stream that you wish to base the new stream on. If this is left blank the latest changeset will be used. You can either type in the changeset id string, or you can select the changeset from the drop down list.

Folder

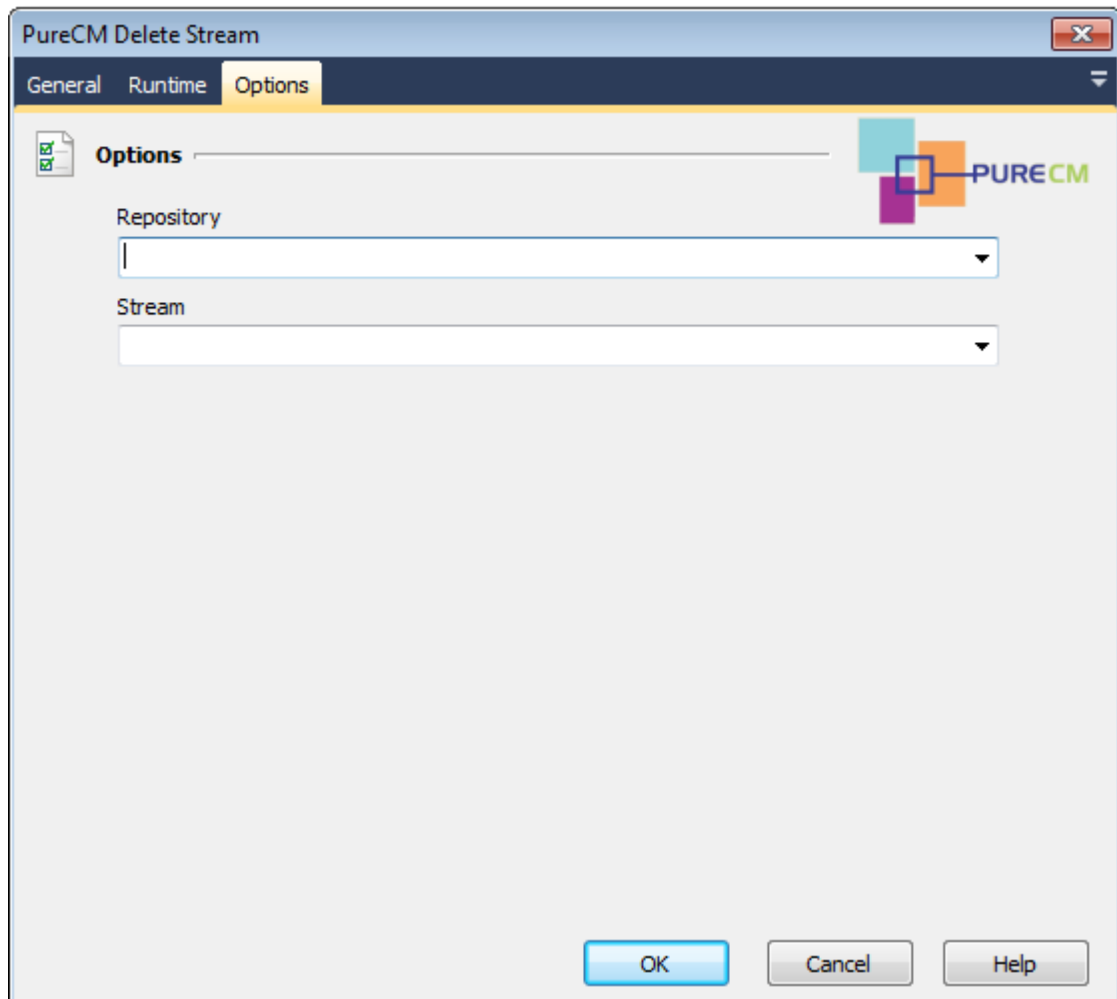
The folder the new stream will be placed in. You can either type in the folder path, or select one from the list of currently available folders.

Description

The description of the new stream you are creating. This can be left blank.

6.34.11.1 PureCM Delete Stream

The PureCM Delete Stream action enables you to delete a stream from PureCM.



Repository

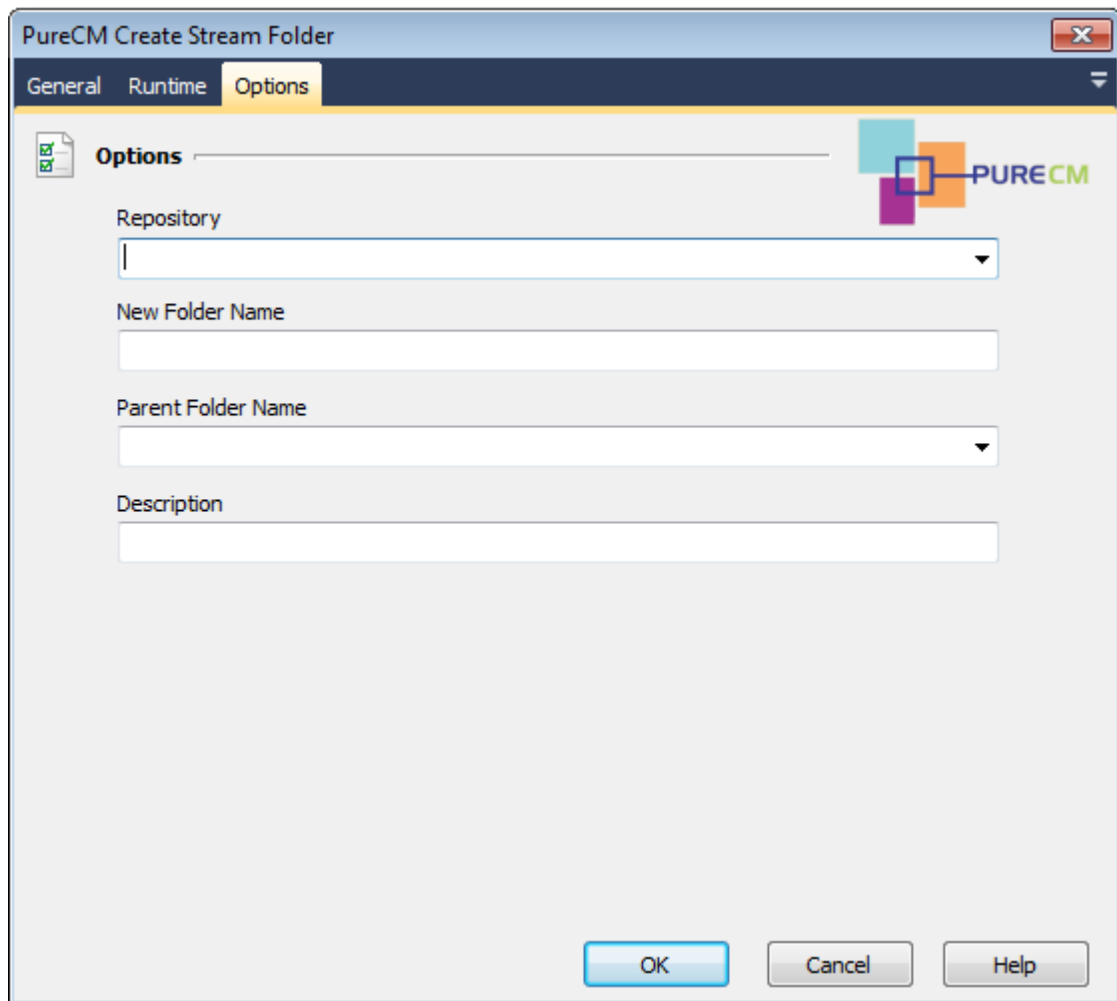
The repository that you wish to delete your stream from. You can either type in the repository name or select from a list of the current repositories.

Stream

The stream that you wish to delete. You can either type in the stream name or select from a list of all available streams inside your selected repository.

6.34.11.1 PureCM Create Stream Folder

The PureCM Create Stream action enables you to make a new stream within PureCM.



The screenshot shows the 'PureCM Create Stream Folder' dialog box with the 'Options' tab selected. The dialog has a title bar with a close button. Below the title bar are three tabs: 'General', 'Runtime', and 'Options'. The 'Options' tab is active and contains a 'PURECM' logo in the top right corner. The main area of the tab has a document icon and the label 'Options'. Below this are four input fields: 'Repository' (a dropdown menu), 'New Folder Name' (a text box), 'Parent Folder Name' (a dropdown menu), and 'Description' (a text box). At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Repository

The repository that you wish to create your stream folder inside. You can either type in the repository name or select from a list of the current repositories.

New Folder Name

The name of the new stream folder you want to create.

Folder

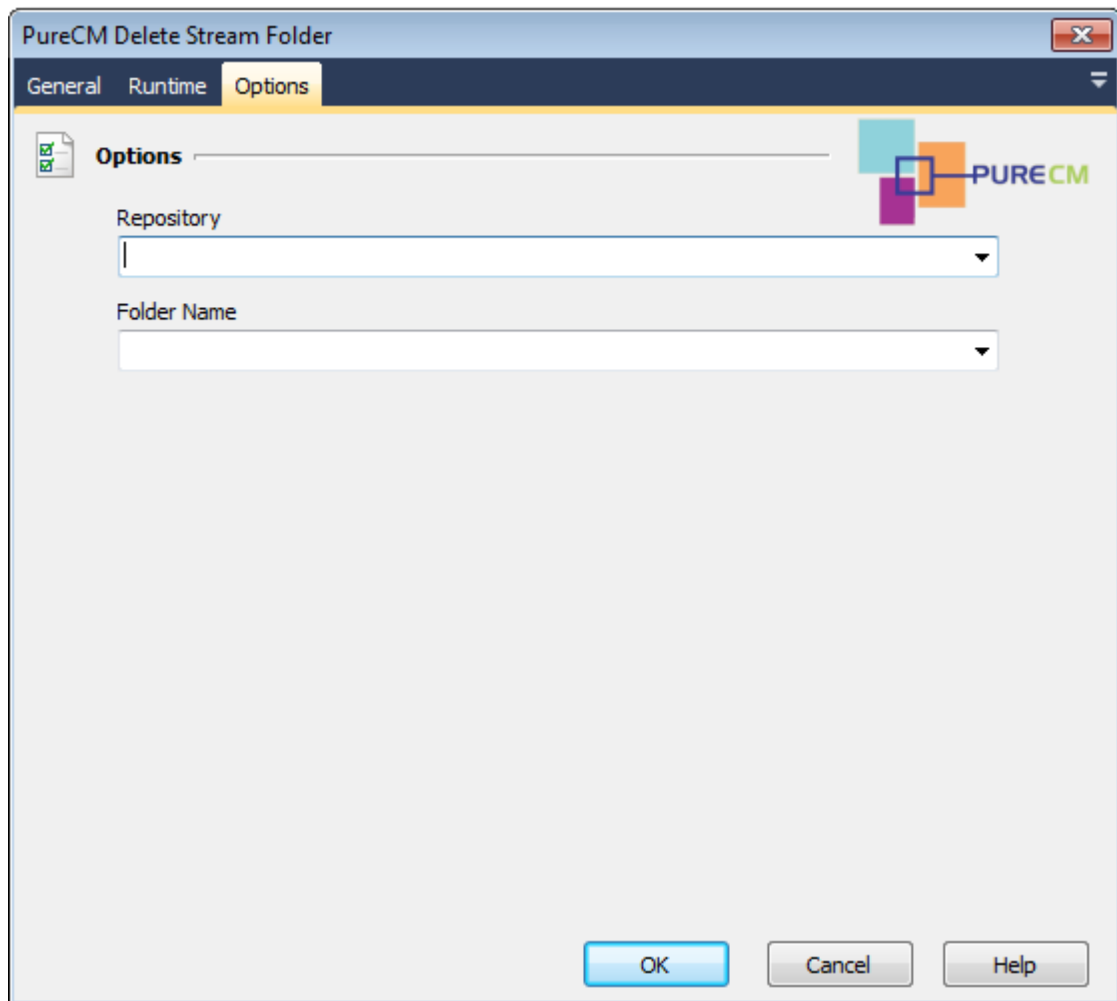
The folder the new stream folder will be placed in. You can either type in the folder path, or select one from the list of currently available folders.

Description

The description of the new stream folder you are creating. This can be left blank.

6.34.11.13 PureCM Delete Stream Folder

The PureCM Delete Stream action enables you to delete a stream from PureCM.



Repository

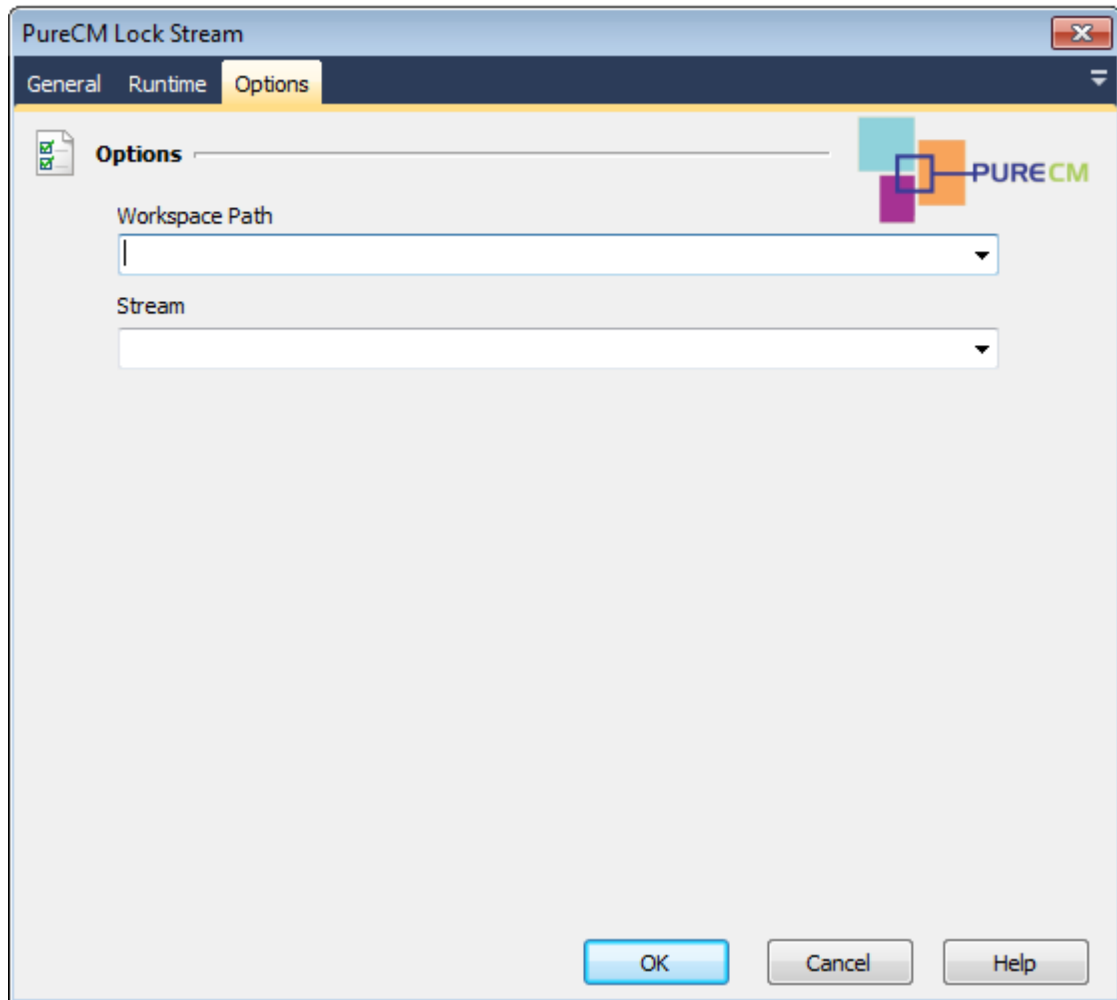
The repository that you wish to delete your stream from. You can either type in the repository name or select from a list of the current repositories.

Folder Name

The stream Folder that you wish to delete. You can either type in the Folder name or select from a list of all available streams folders inside your selected repository.

6.34.11.14 PureCM Lock Stream

The PureCM Lock Stream action enables you to lock a stream and prevent other users from making changes to it.



Repository

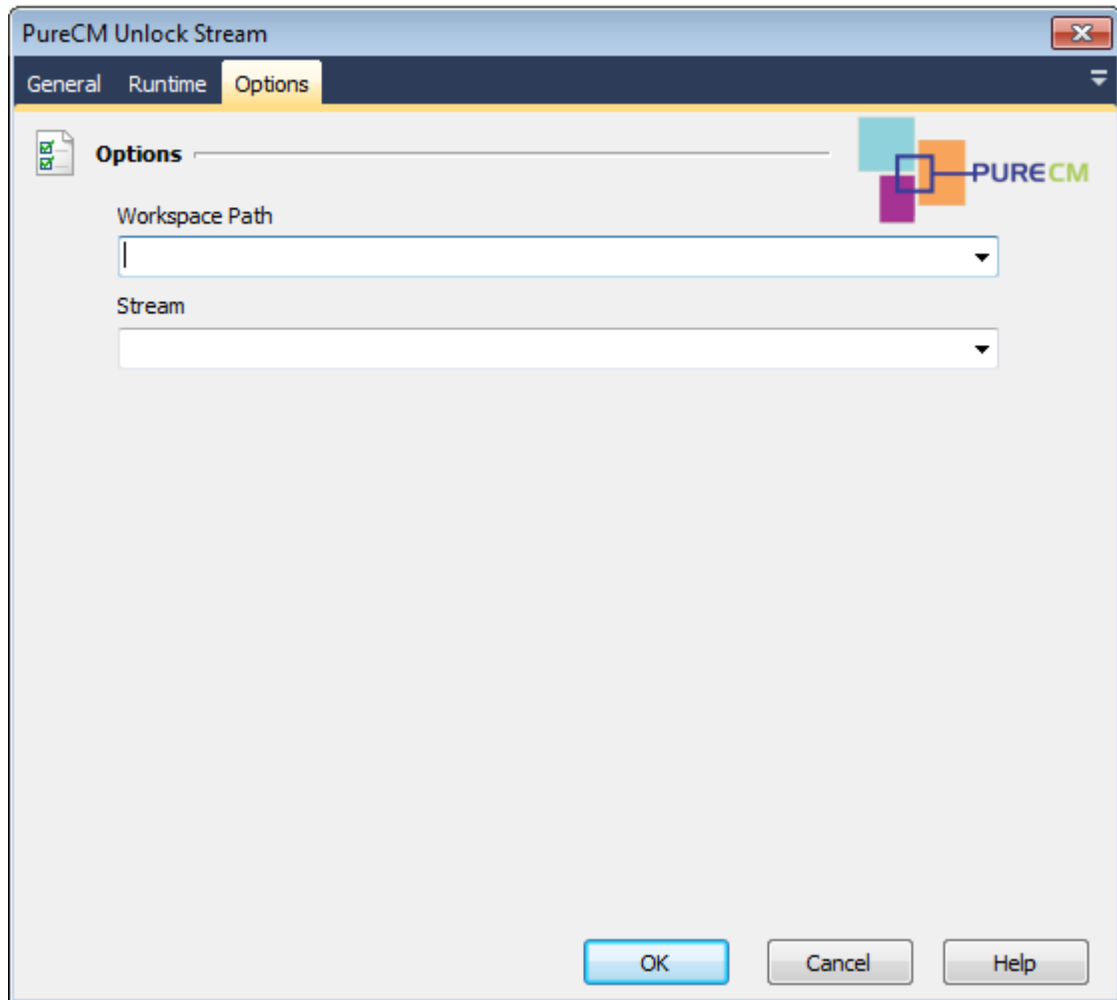
The repository that you wish to lock the stream in. You can either type in the repository name or select from a list of the current repositories.

Stream

The stream that you wish to lock. You can either type in the stream name or select from a list of all available streams inside your selected repository.

6.34.11.1!PureCM Unlock Stream

The PureCM Unlock Stream action enables you to unlock a previously locked stream and allow other users to submit changes.



Repository

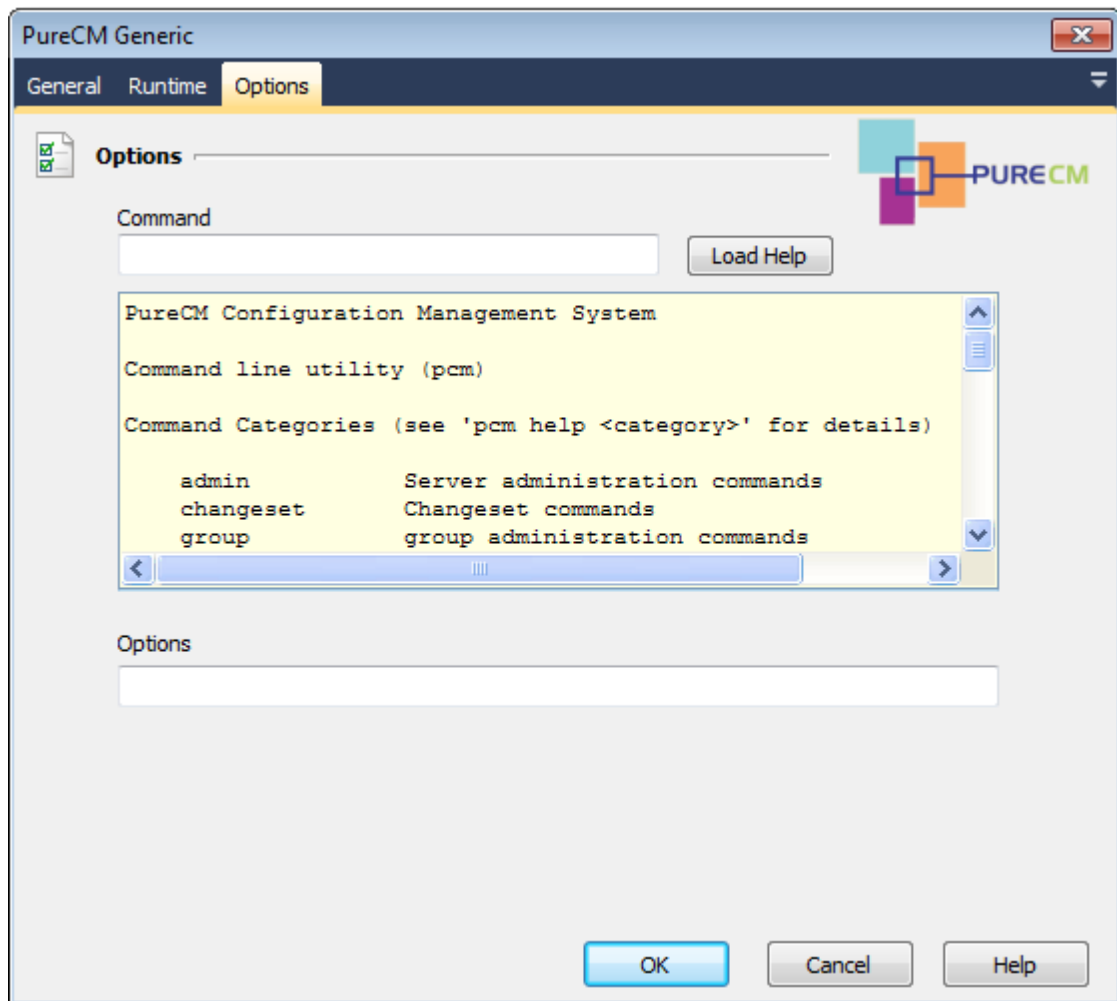
The repository that you wish to unlock the stream in. You can either type in the repository name or select from a list of the current repositories.

Stream

The stream that you wish to unlock. You can either type in the stream name or select from a list of all available streams inside your selected repository.

6.34.11.1 PureCM Generic

The PureCM Generic action enables you to call any command from the pcm command line tool.



Command

The pcm command you wish to call.

Options

The parameters of the pcm command.

Help

The help window will display the help for the pcm command entered into the command box. The help can be either refreshed by pressing the “Load Help” button, or by pressing the space bar.

6.34.12 PVCS

Use these actions to perform various operation on your PVCS repository.

The available actions are:

- PVCS Get
- PVCS Put
- PVCS Add Label
- PVCS Promote
- PVCS List Files

NOTE: The PVCS actions use the PCLI.exe command line interface for their various operations. You first need to set up the path to PCLI.exe in the Tools | Options dialog.

6.34.12.1 PVCS Add Label

PVCS Get will label files in your PVCS repository.

PVCS Add Label

General Runtime **Label**

Project

Project DB Directory : c:\PVCSDB

Project Path : /MyProject

Entity : *.*

Label

Build 1.0.1.53

☒ Floating (Label will float to the newest revision)

☐ Include versioned files in subprojects

Specify Revision

Revision, Promotion Group or Version Label

Leave blank to assign label to tip revision

Security

Username : Password :

Answer to any question

☐ Yes ☒ No

The PVCS command line (PCLI) sometimes prompts the user, FinalBuilder needs to answer this question automatically with either Yes or No

OK Cancel Help

Project DB Directory - specify the Database directory of your PVCS repository

Project Path - specify the path to the project to label

Entity - specify the file mask of the files you want to label

Label - the new label

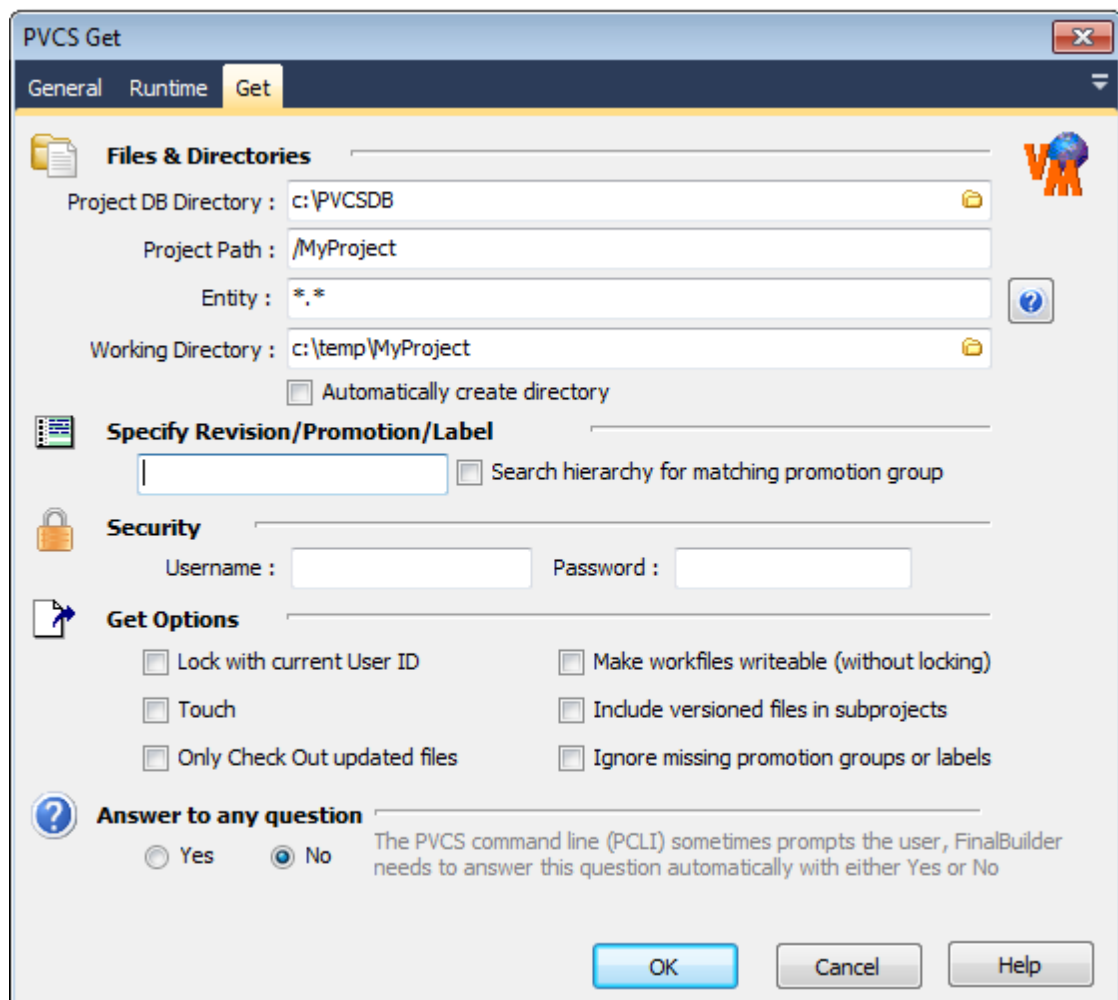
Floating - label will float to the newest revision

Revision - specify the revision, promotion group or version label of the files to label, if left blank the tip revision will be labeled

Security - override the default username/password

6.34.12.2 PVCS Get

PVCS Get will get files from your PVCS repository.



Project DB Directory - specify the Database directory of your PVCS repository

Project Path - specify the path to the project you want to get

Entity - specify the file mask of the files you want to retrieve.

Working Directory - specify the working directory to retrieve the files to

Automatically Create Directory - FinalBuilder will create the working directory if it doesn't exist

Revision/Promotion/Label - PVCS will only retrieve files that match the specified revision/promotion/label

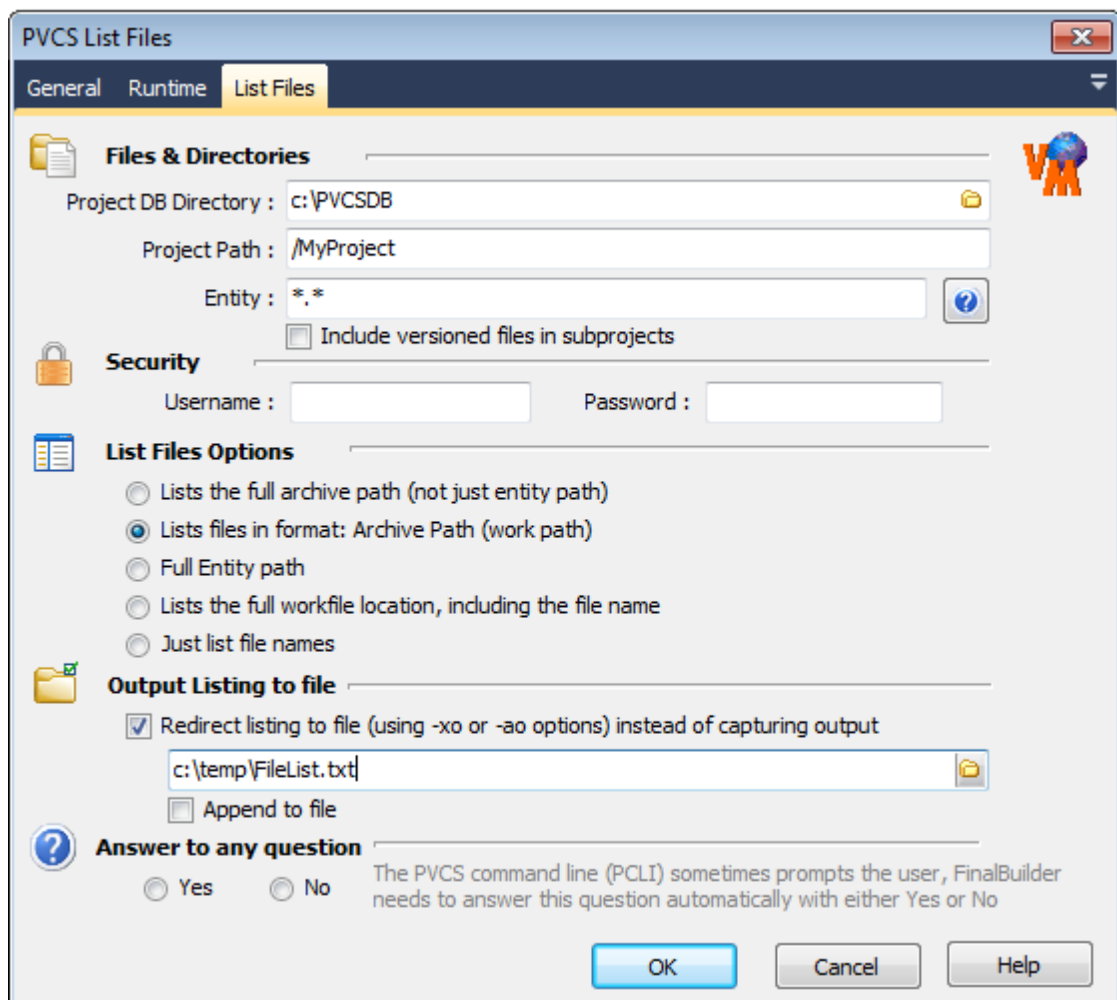
Security - specify an alternate username/password if you want to override the default security credentials

Lock with current User ID - lock the files retrieved

Touch - change the modified date on the retrieved files to the current date/time

6.34.12.3 PVCS List Files

PVCS Get will list files in your PVCS repository.



Project DB Directory - specify the Database directory of your PVCS repository

Project Path - specify the path to the project you want to list

Entity - specify the file mask of the files you want to list.

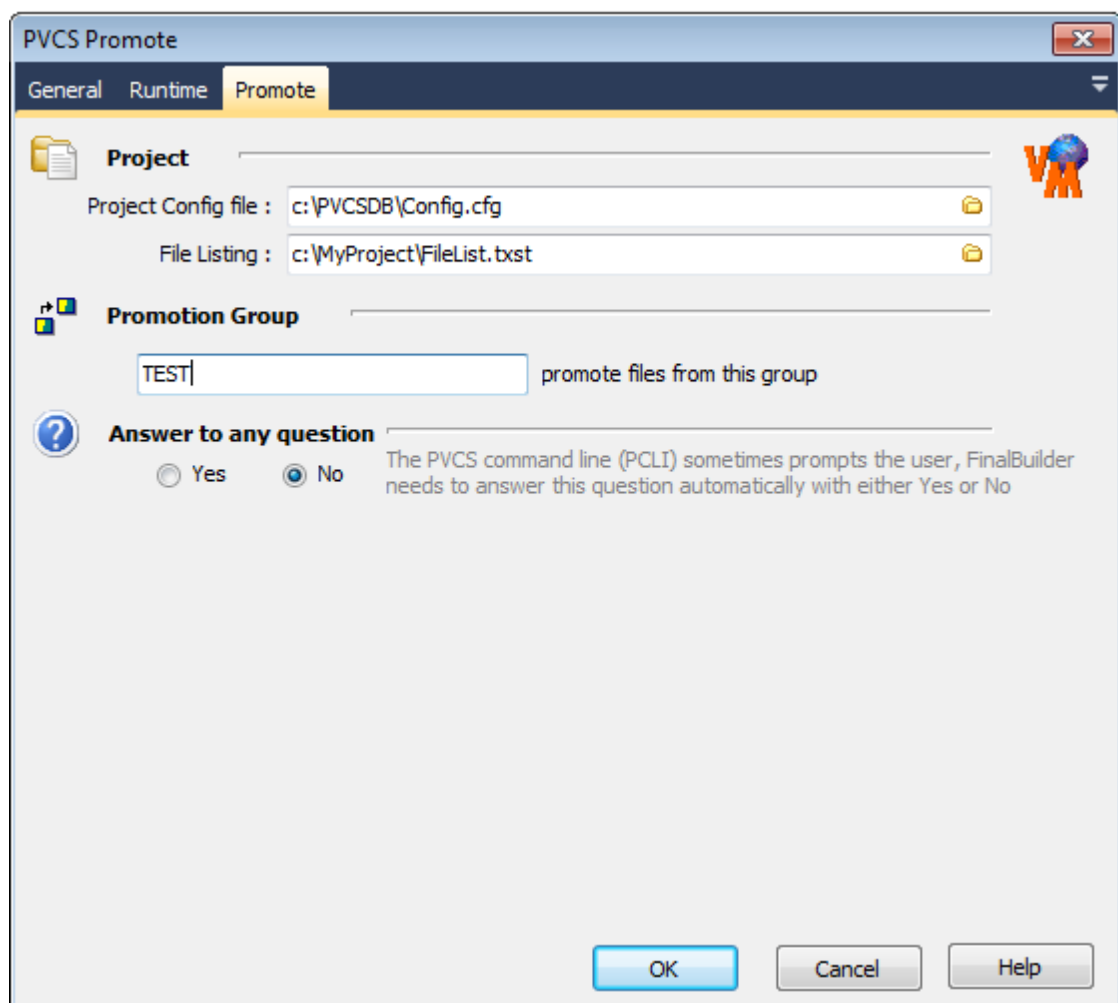
Security - specify an alternate username/password if you want to override the default security credentials

Output listing to file - if you want PVCS to redirect to a file, then select this option and enter the filename. Otherwise the listing will be captured in the FinalBuilder log.

Include Versioned Files in Subprojects - The "Include versioned files in subprojects" check box sets the -z flag when calling pcl.exe. In Version Manager 8.1 (and possibly other versions), the entity edit box must be set to "*" to make that work. Using "*", "*", or "*.txt" will only report the files in the listed VM project, and ignore any sub projects.

6.34.12.4 PVCS Promote

PVCS Get will promote files in your PVCS repository.



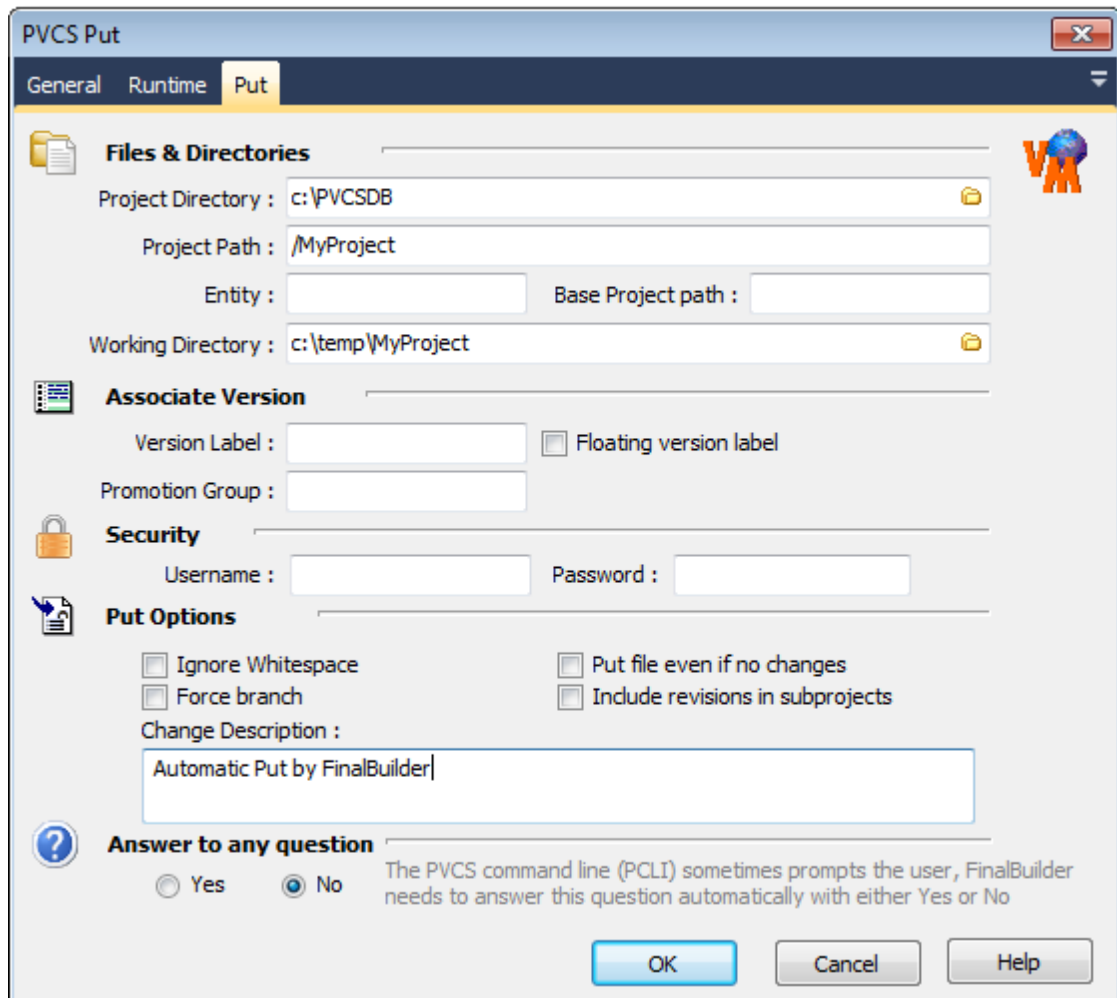
Project Config file - specify the project config file

File Listing - specify the file listing file

Promotion Group - specify the group to promote files from

6.34.12.5 PVCS Put

PVCS Get will put files to your PVCS repository.



Project DB Directory - specify the Database directory of your PVCS repository

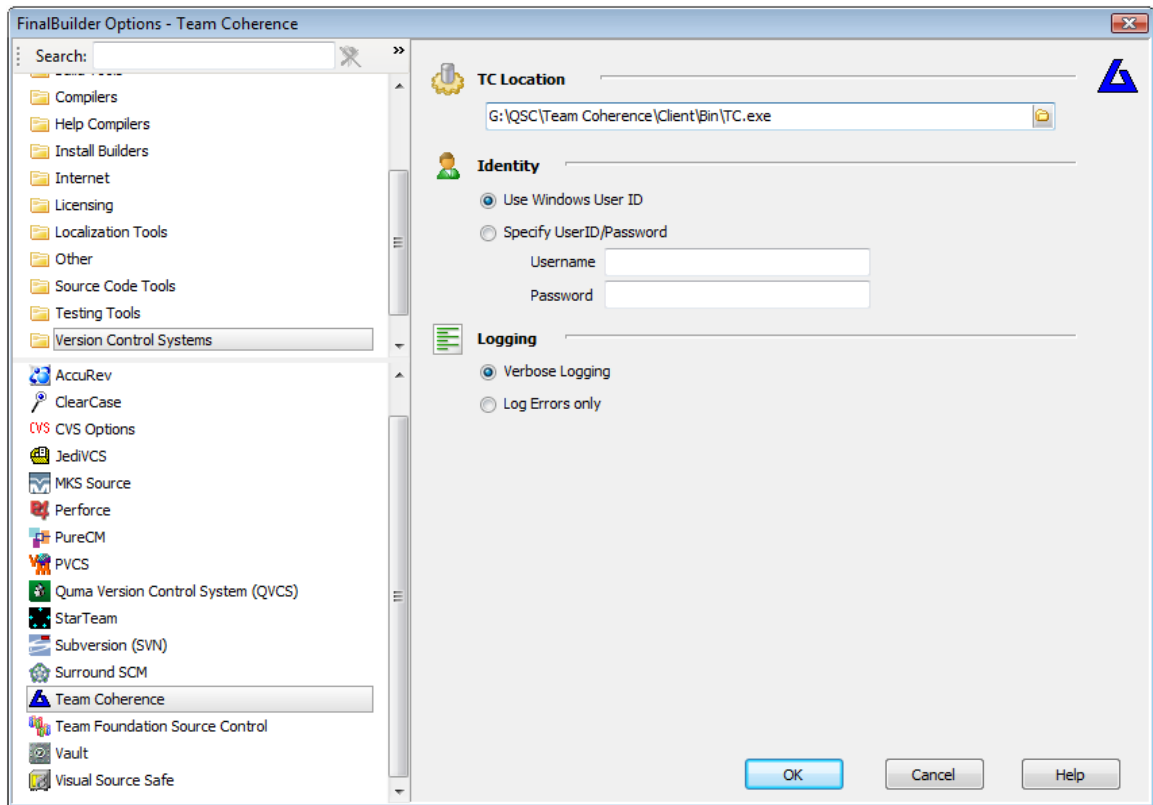
Project Path - specify the path to the project you want to put files to

Entity - specify the file mask of the files you want to put.

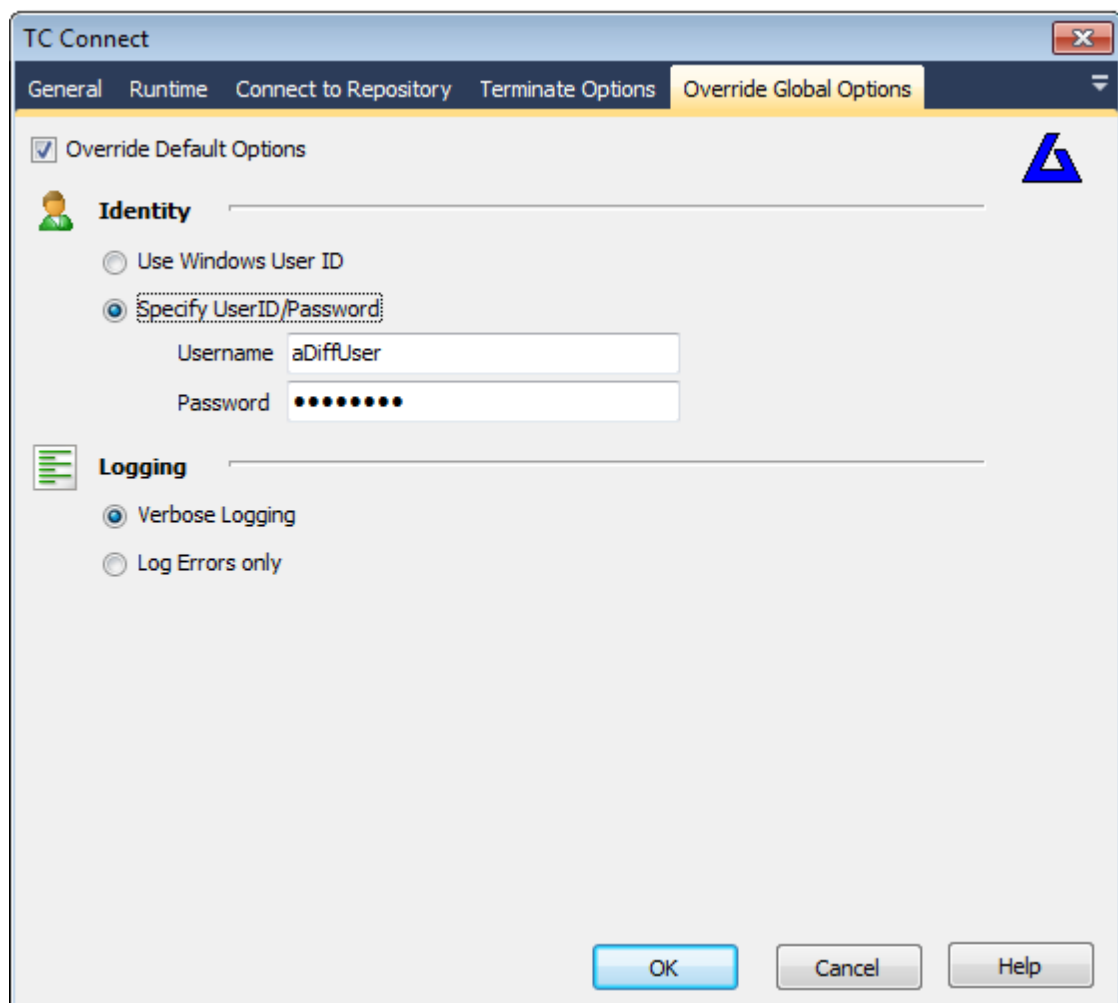
Working Directory - specify the working directory to put the files from

6.34.13 QSC Team Coherence Actions

These Actions support Team Coherence 7.1 or higher. They provide an easy to use wrapper around the TC command line tool. Before using these actions you need to check that the path to the TC command line tool is set in the FinalBuilder Options Dialog. FinalBuilder will attempt to find the path automatically however it is not always possible (depends on your installation).

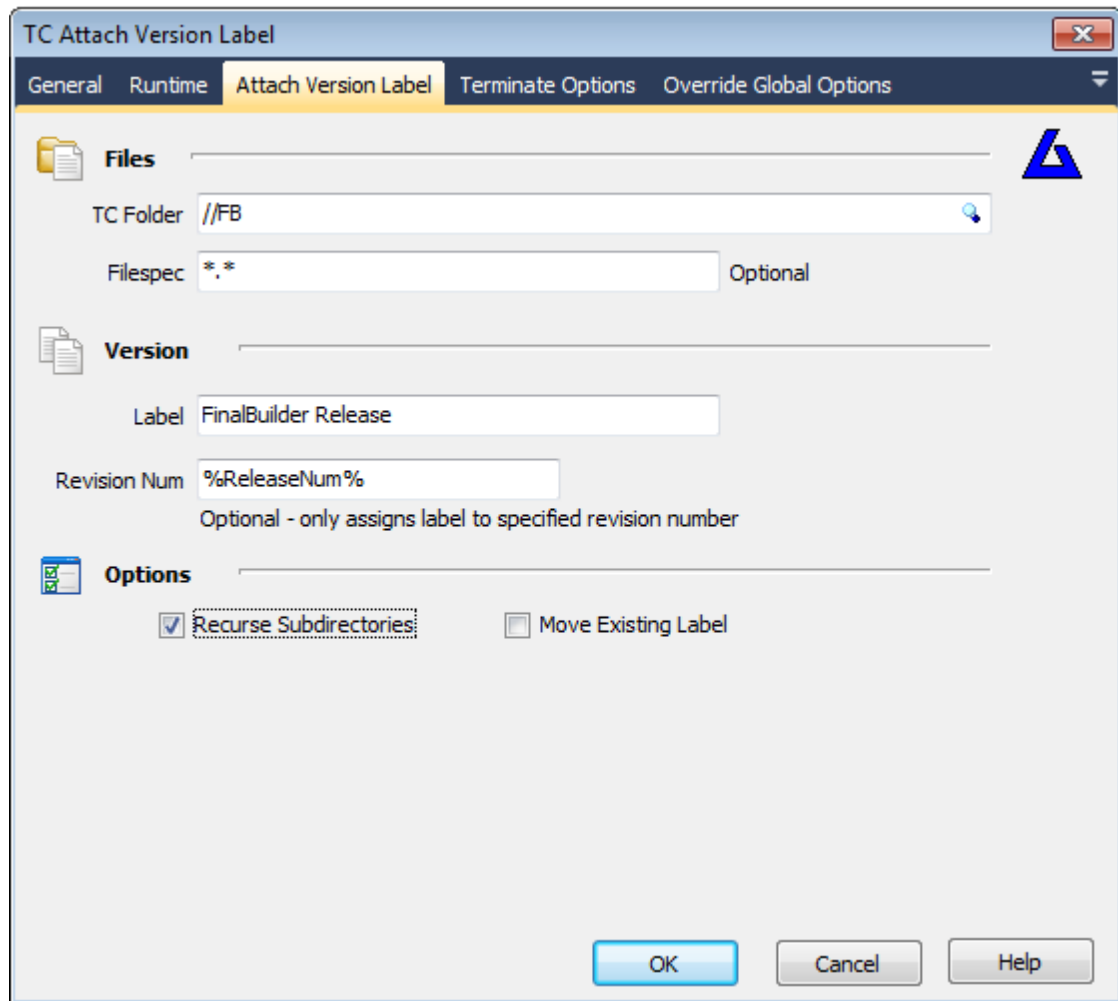


The default options can be overridden in the individual actions.



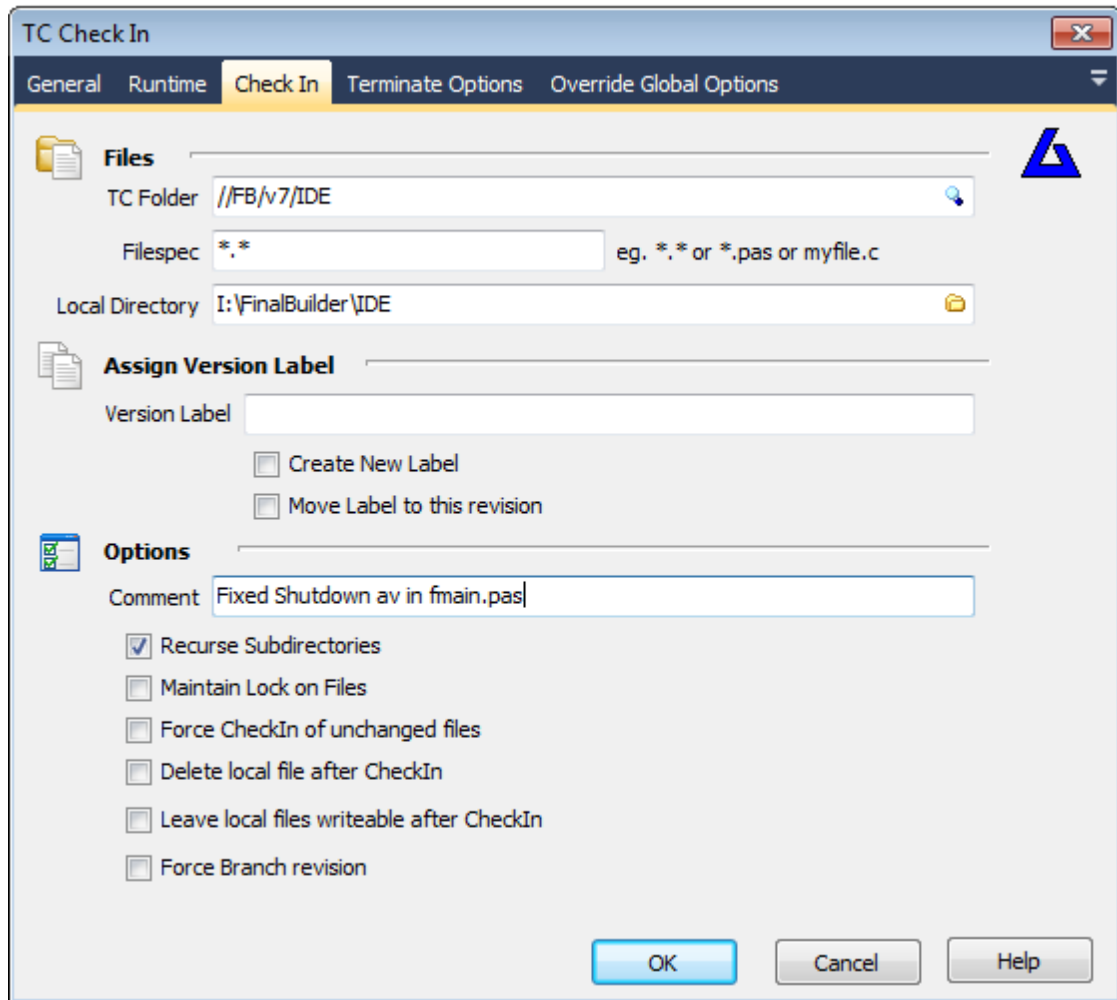
6.34.13.1 Team Coherence Attach Label Action

Attaches a Version Label to the latest revision of a file or to a specified revision.



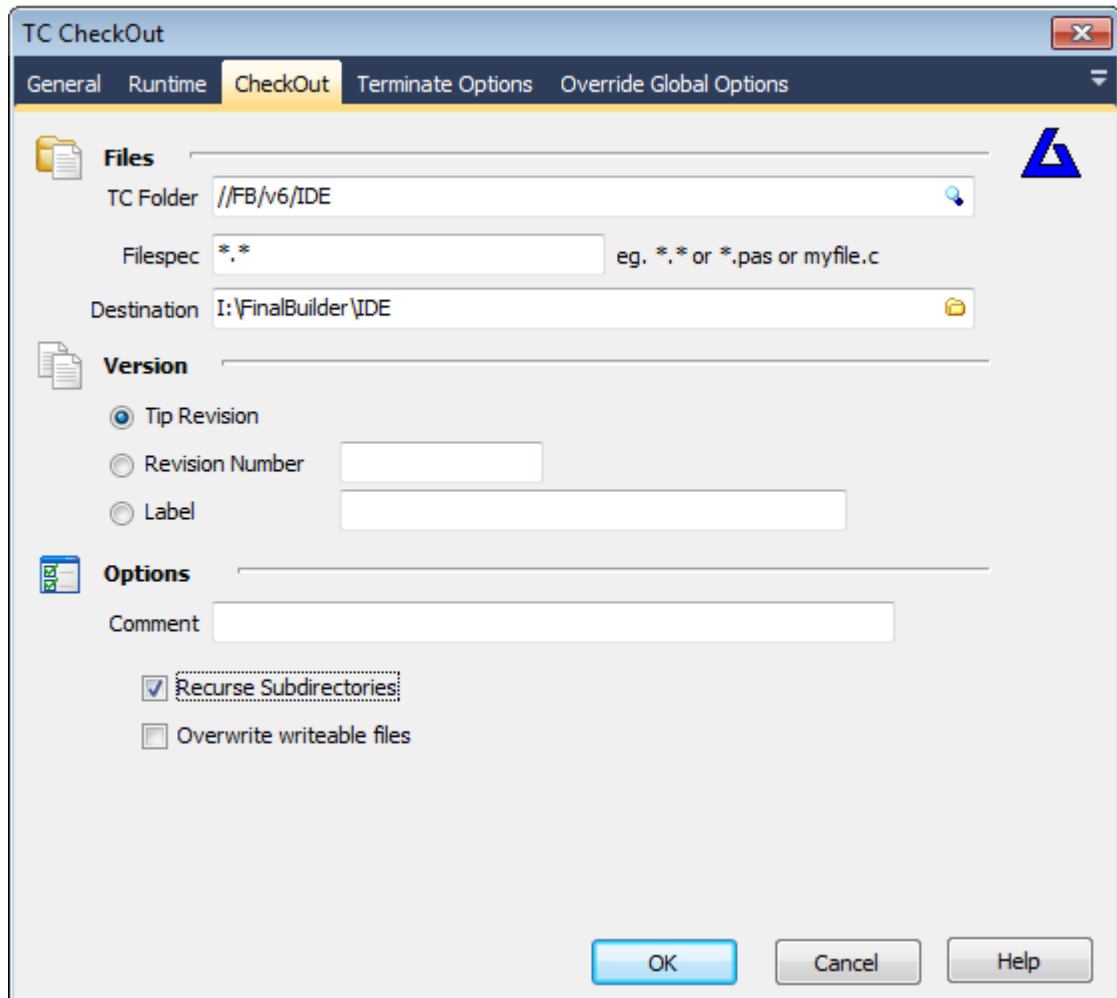
6.34.13.2 Team Coherence Check In Action

Updates Team Coherence with changes made to a checked out file and unlocks the archive.



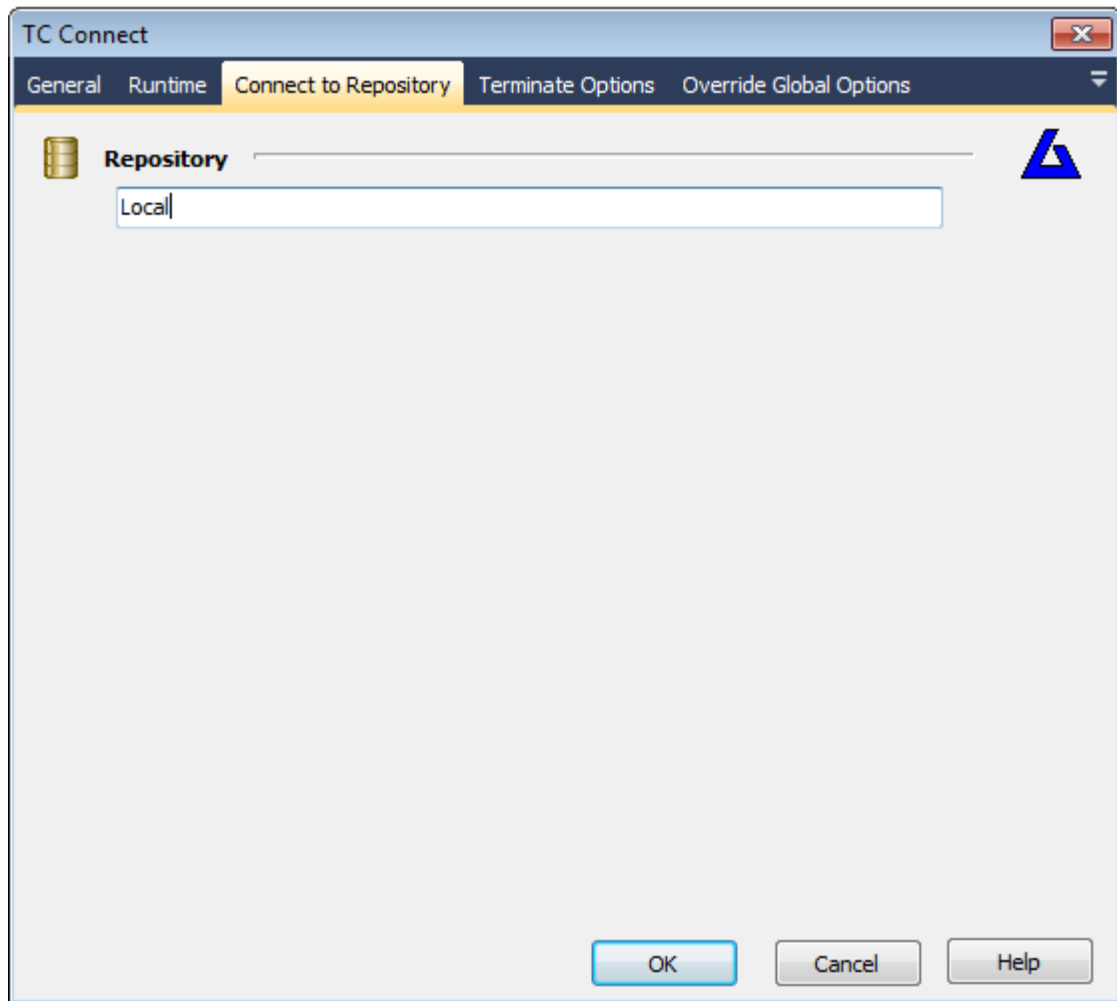
6.34.13.3 Team Coherence Check Out Action

Extracts a file from the current folder to the current directory for the purposes of editing.



6.34.13.4 Team Coherence Connect Action

Connects to a different repository.



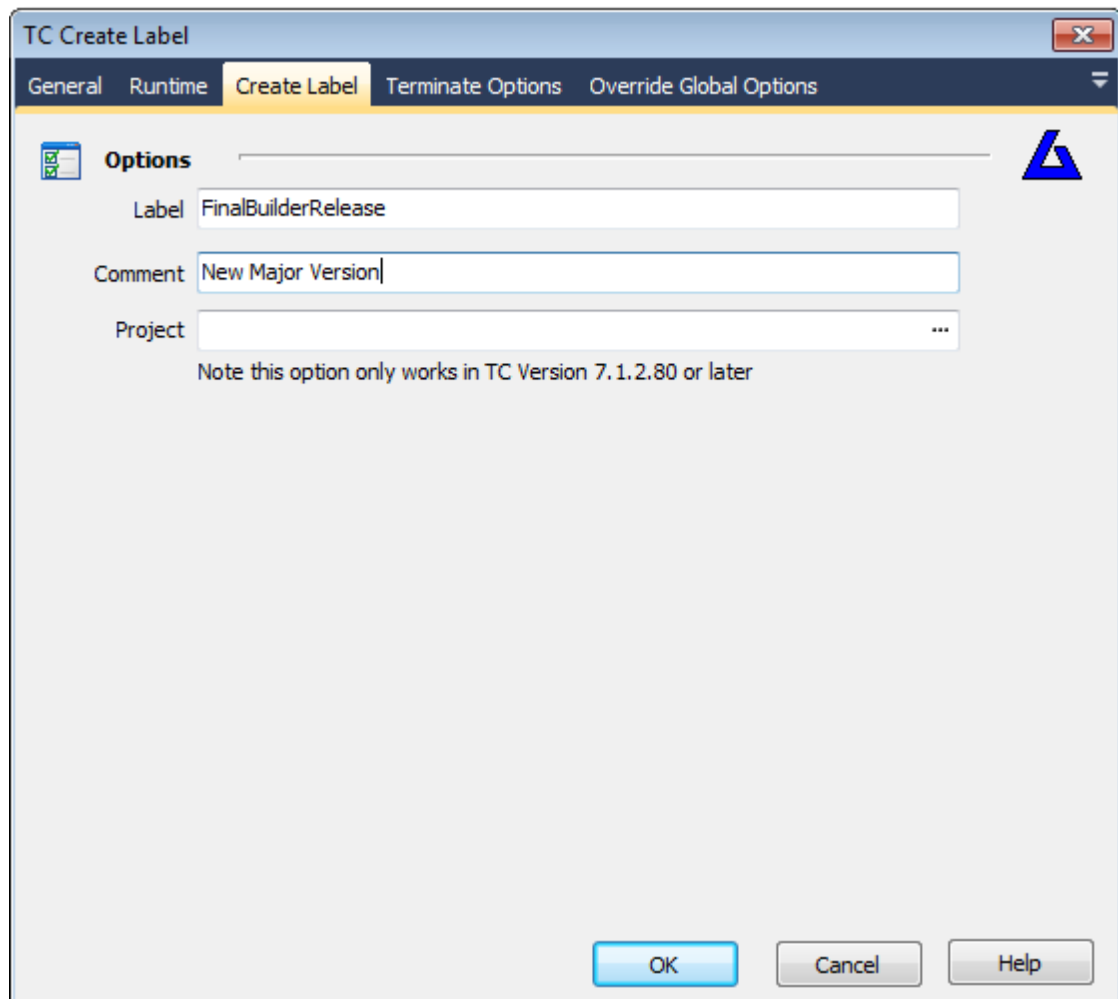
Remarks

When you connect to a different repository, it becomes the default for subsequent commands.

Note that you must be connecting to a previously created repository connection. You can define connections using the Team Coherence Version Manager.

6.34.13.5 Team Coherence Create Label Action

Creates a new Version Label.



6.34.13.6 Team Coherence Create View

Create a new View in the repository.

TC Create View

General Runtime **Create View** Terminate Options Override Global Options

New View Details

View Name: FBx

Description: FinalBuilder x.xx

Projects: //FinalBuilder

If no projects are specified then view will apply to all projects

One project per line

Options

☒ Shared View

☐ Base view on version

Version:

☐ Base view on promotion level

Promotion Level:

OK Cancel Help

If you want the view to only apply to certain projects, then you must specify them in the Projects section.

Shared View - Shared Views are available to all users connected to a repository, however shared views can only be created by Admin users.

Base view on version - Specify the version which the view will be based on.

Base view on promotion level - Specify the predefined promotion level to base the view on. Promotional views are used to view all files at a particular level in the promotional hierarchy. Users can get a read-only copy of all the files at a particular promotional level but cannot modify them.

More about TC Views, from the TC help file:

A View is perhaps the most powerful tool available in Team Coherence when it comes to managing multiple versions of a project. At their simplest, Views allow you to filter which projects, files, and revisions are displayed in Version Manager.

Working with previous versions of a project then simply becomes a case of selecting a View. Once defined and selected, as far as the user is concerned they will be working on the latest version of the project. All branching, synchronizing, reassignment of Version Labels, etc are handled automatically by Team Coherence whether working through Version Manager, or your selected IDE interface.

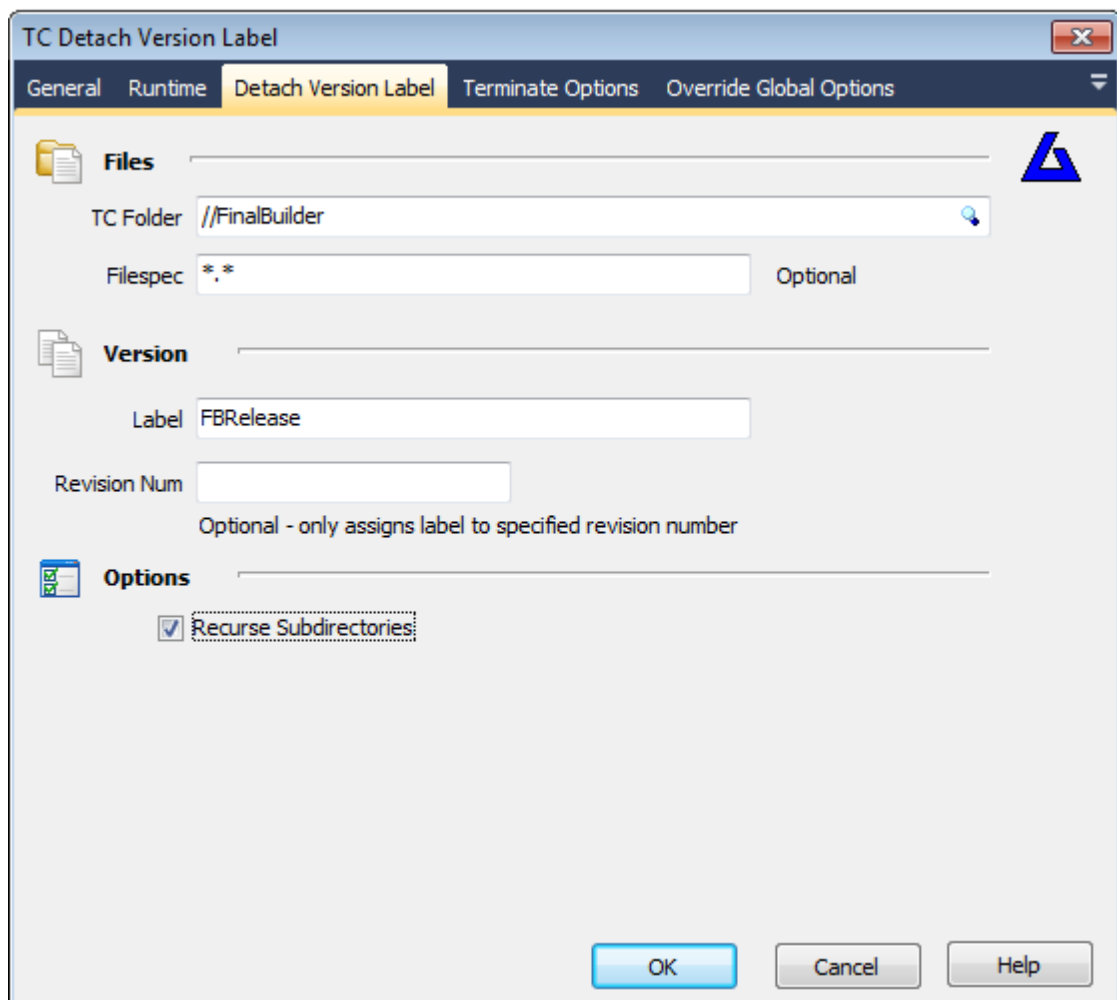
In addition, if required, the default working path for the files in any View can be different for each View, or can be based on the <default> View.

6.34.13.7 Team Coherence Delete View

Delete a view in the TC repository. See Team Coherence Create View for more information on views.

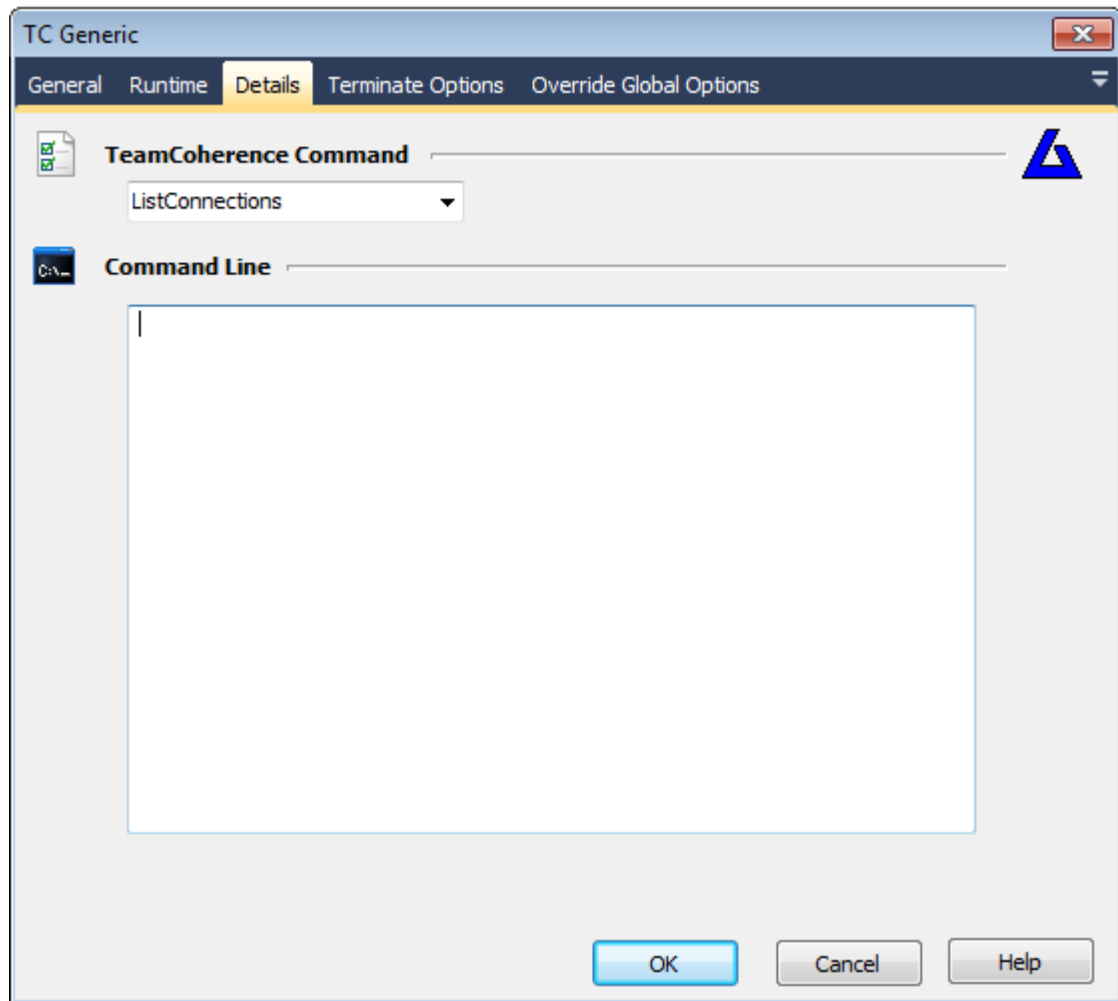
6.34.13.8 Team Coherence Detach Label Action

Detaches the specified label from the specified objects.



6.34.13.9 Team Coherence Generic

The TeamCoherence Generic action allows you to execute functionality which isn't natively supported by the other TC actions.

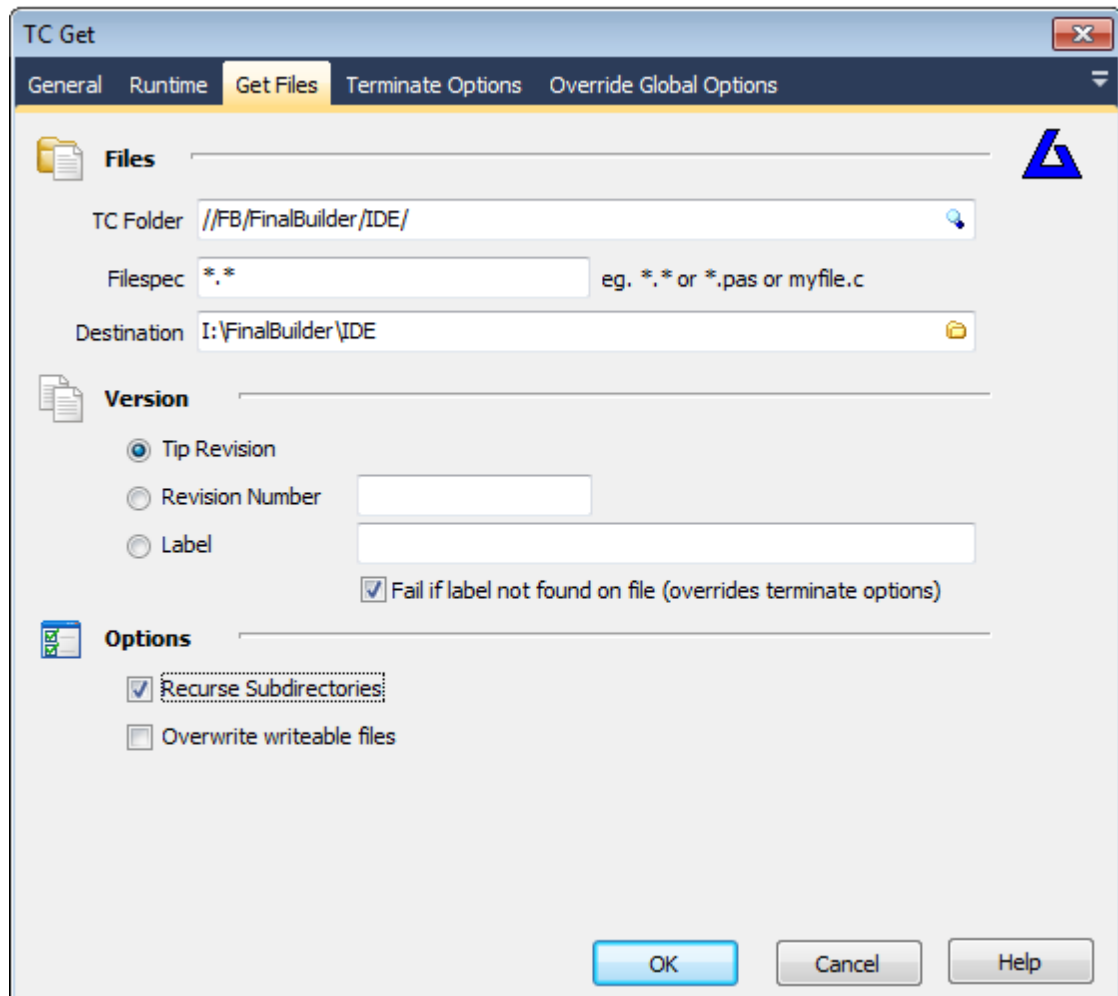


The TeamCoherence Command combobox specifies the available TC commands.

Command Line - specify the specific command line to send to tc.exe

6.34.13.1 Team Coherence Get Action

Gets a read-only copy of the specified files.

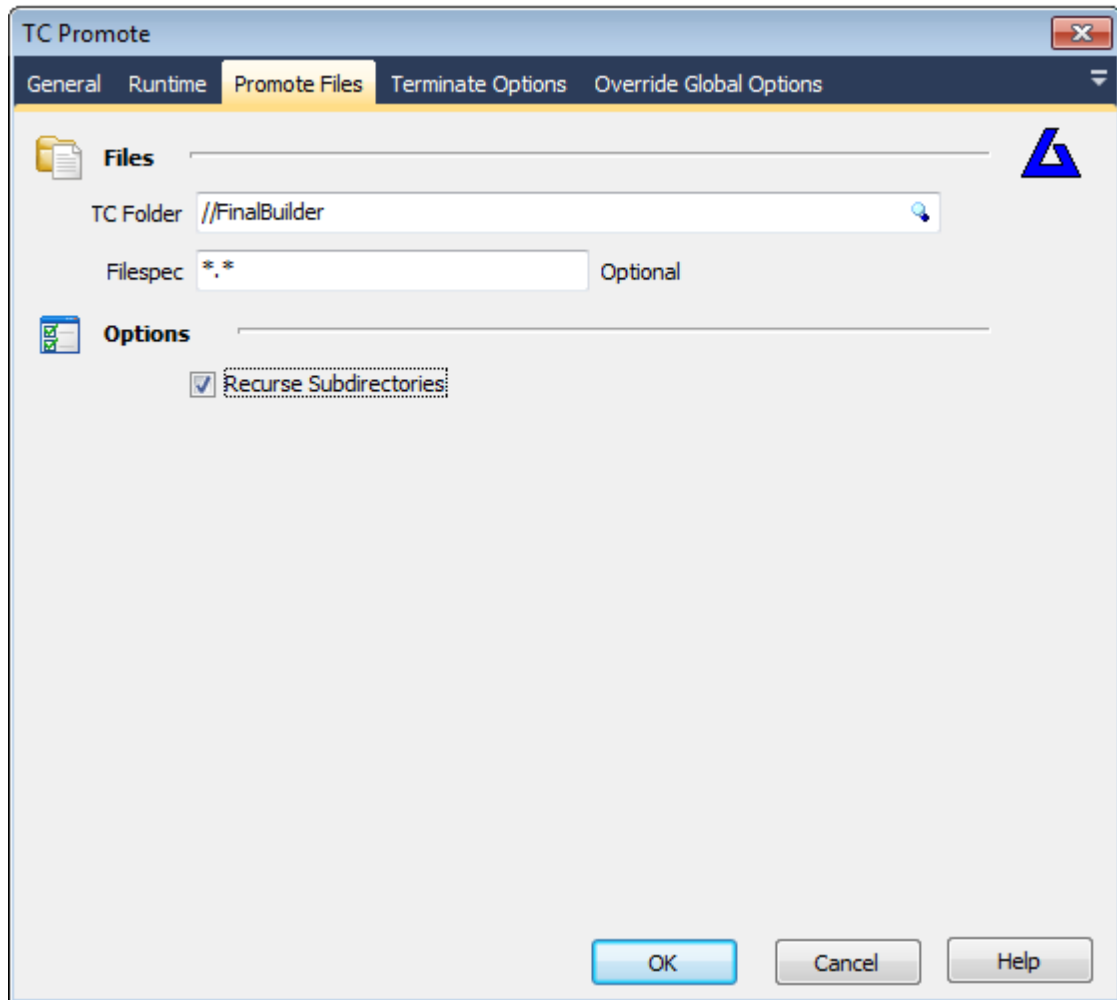


Remarks

If the local copy of the file is not read-only the action will fail for that file. To override this default and to cause the local file to be overwritten with the new read-only copy, use the -W command-line option.

6.34.13.1 Team Coherence Promote Action

Promotes the specified files to the next Promotion Level. This action will promote the tip revision (as defined by the current View) to the next higher Promotion Level.

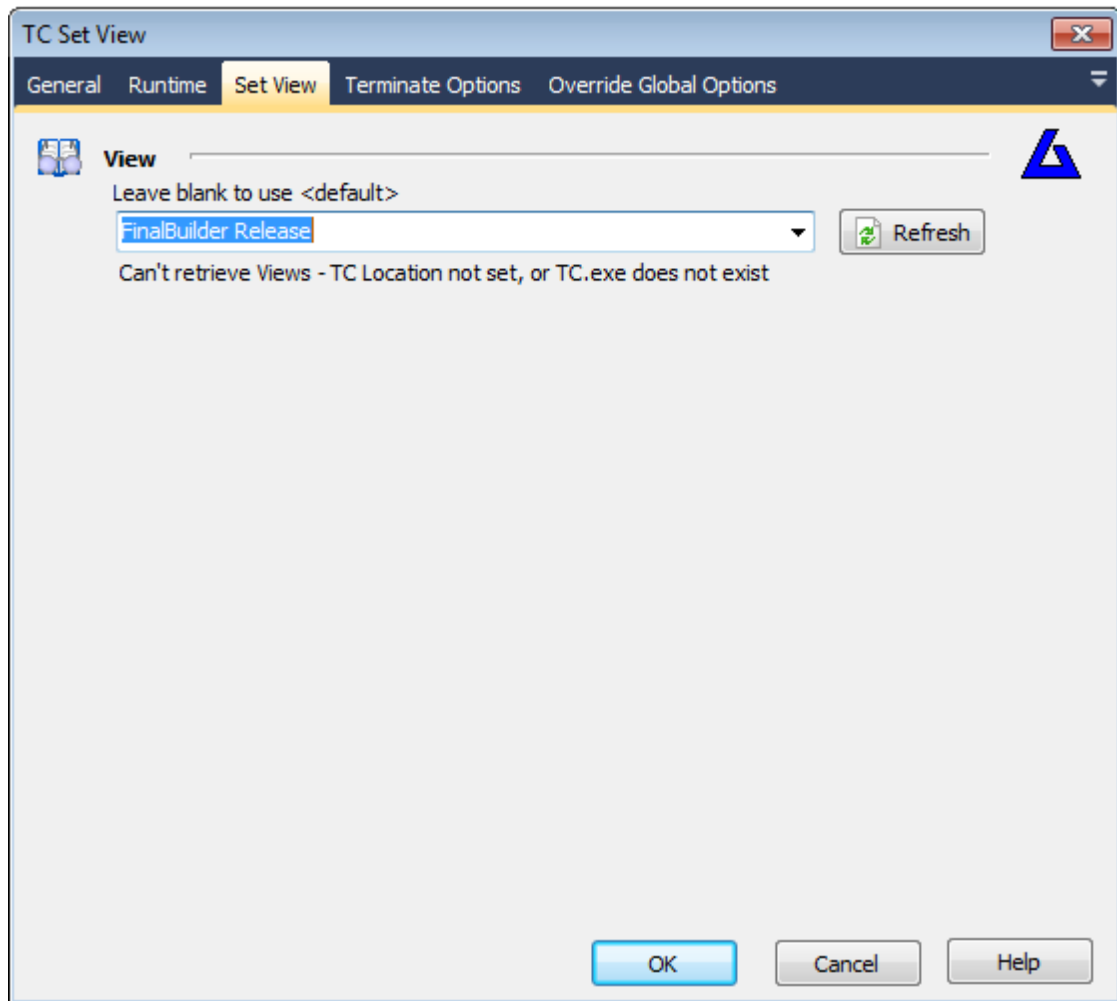


Both the Tip Revision and the next higher Promotion Level are defined by the currently selected View. If the current View is based on a Version Label, the revision that has the Version Label attached will be promoted to the first Promotion Level. If the current View is based on a Promotion Level, the revision that is currently at that level will be promoted to the next higher Promotion Level.

If the <default> View is current, or the current View is based on neither a Version Label or Promotion Level, the tip revision of each file will be promoted.

6.34.13.1 Team Coherence Set View Action

Selects a different view and makes it current.

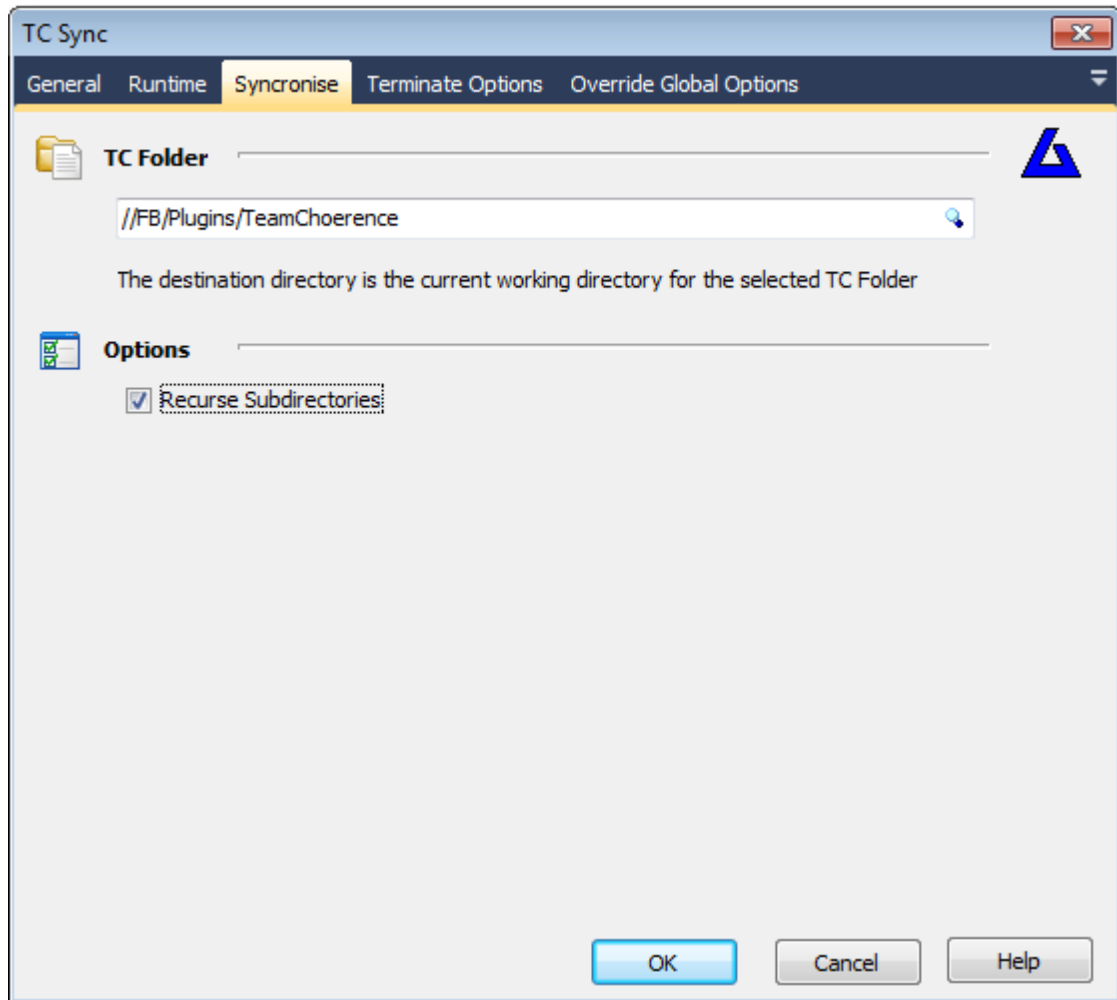


Remarks

For more information on Views, see the main Team Coherence help file.

6.34.13.1 Team Coherence Sync

The Sync command will synchronise your out of date or missing local files with the current files in the repository.

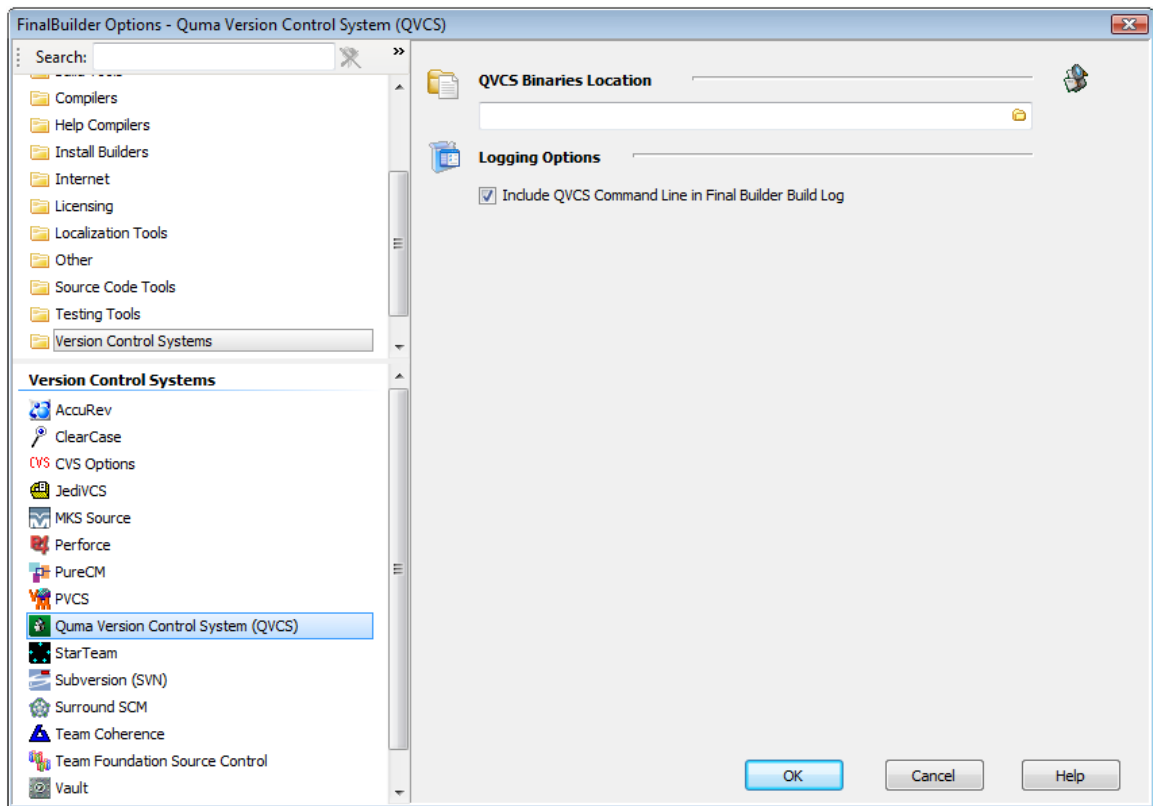


6.34.13.1 Team Coherence Update View

Update View enables you to change the definition of an existing view. See Team Coherence Create View for more information on the available properties.

6.34.14 QVCS Actions

The QVCS global options page contains settings that all the QVCS actions will use.



QVCS Binaries Location

The location to the folder which contains the QVCS binaries.

Logging Options

Determines whether the output from QVCS is shown in the FinalBuilder log.

6.34.14.1 QVCS Add File Action

The QVCS Add File action enables you to add files to your source code repository using QVCS.

For more information on QVCS, see <http://www.qumasoft.com/>

6.34.14.2 QVCS Check In File(s) Action

The QVCS Check In File(s) action enables you to check in files to your source code repository using QVCS.

For more information on QVCS, see <http://www.qumasoft.com/>

6.34.14.3 QVCS Check Out File(s) Action

The QVCS Check Out File(s) action enables you to check out files from your source code repository using QVCS.

For more information on QVCS, see <http://www.qumasoft.com/>

6.34.14.4 QVCS Get Latest Version Action

The QVCS Get Latest Version action enables you to synchronise your local files to the latest version from your source code repository using QVCS.

For more information on QVCS, see <http://www.qumasoft.com/>

6.34.14.5 QVCS Labels File(s) Action

The QVCS Label File(s) action enables you to assign version labels to your files in your source code repository using QVCS.

For more information on QVCS, see <http://www.qumasoft.com/>

6.34.14.6 QVCS Undo Check Out File(s) Action

The QVCS Undo Check Out File(s) action enables you to undo checkouts of specified files using QVCS.

For more information on QVCS, see <http://www.qumasoft.com/>

6.34.15 Seapine Surround SCM

The Surround SCM actions provide FinalBuilder integration for Seapine Surround SCM. The actions were developed with version 3.1.x of Surround SCM.

For more information on Surround SCM, see <http://www.seapine.com>

Before you start using the Surround SCM actions, it is suggested you first set up Global Surround SCM Options

The Surround SCM operations supported are: Get, CheckOut, CheckIn, Label, Create Branch, Freeze Branch, Unfreeze Branch, Checkout Report.

If the Surround SCM operation you want to perform is not supported by one of the FinalBuilder actions, you can use the Surround SCM Generic action.

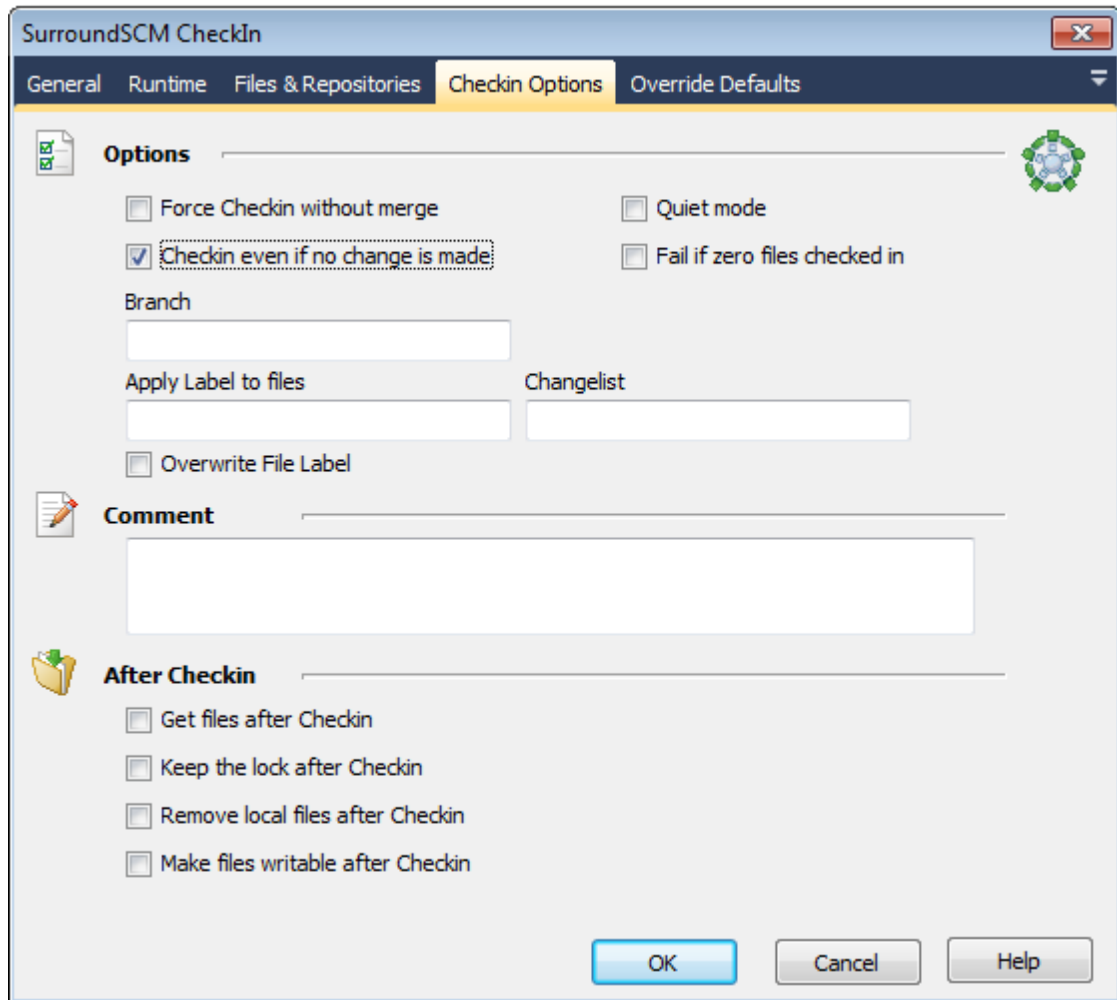
6.34.15.1 Surround SCM CheckIn

Check in updates Surround SCM files with changes, removes the lock on the files, and makes changes available to other users.

The Checkin action uses the same Files & Repositories Tab as the Surround SCM Get action.

Checkin Options Tab

Specify the checkin options - all values are optional.



Branch - Enter the branch name to check in the changes to. Default is set in the working directory.

Label - Enter a label for the check in code.

Changelist - Enter a new or existing changelist name to check in the file as part of a changelist.

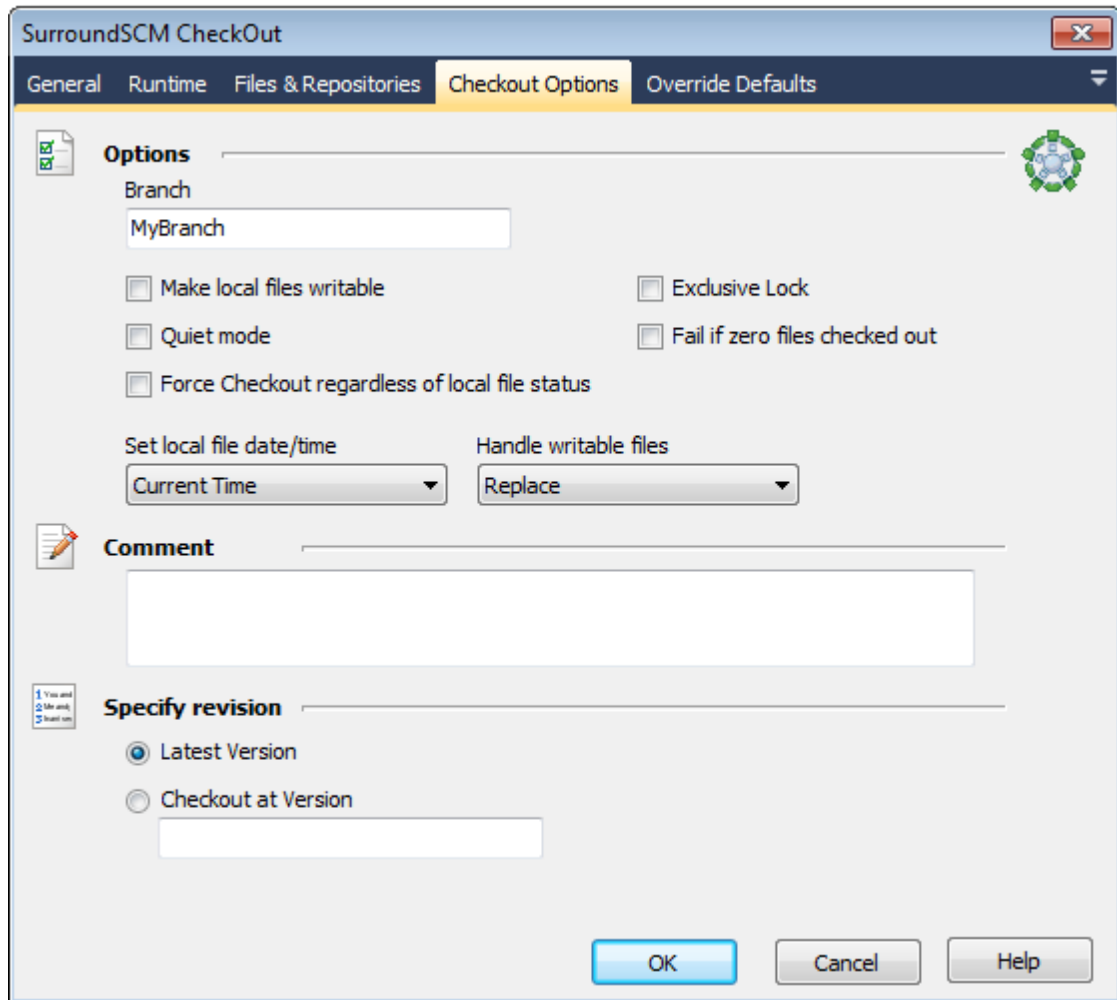
6.34.15.2 Surround SCM CheckOut

Check out files when you need to make changes. You can check out single files, multiple files, or a repository. Surround SCM creates a read-write copy of the file in the working directory.

The Checkout action uses the same Files & Repositories Tab as the Surround SCM Get action.

Checkout Options Tab

Specify the checkout options - all values are optional.



6.34.15.3 Surround SCM Checkout Report

The Checkout report action lists the files checked out in the repository. The action can also automatically fail if any files are checked out.

Checkout Report Options

There are five report types:

1. All checked out files in all repositories in the specified branch and baseline
2. All checked out files in the specified repository and branch
3. All checked out files in all repositories and branches in the specified baseline
4. All checked out files in the specified repository across all branches
5. All checked out files

Depending on which report type is chosen, the Branch, Repository, Baseline and Recursive options will be enabled/disabled to allow you to fill in the correct fields.

To further filter the results by user, enter one or more user names separated by a comma.

SurroundSCM Checkout Report

General Runtime **Checkout Report Options** Output Options Override Defaults

Report Type

1 - All checked out files in all repositories in specified branch and baseline

Options

Branch
Main

☐ Recursive

Repository

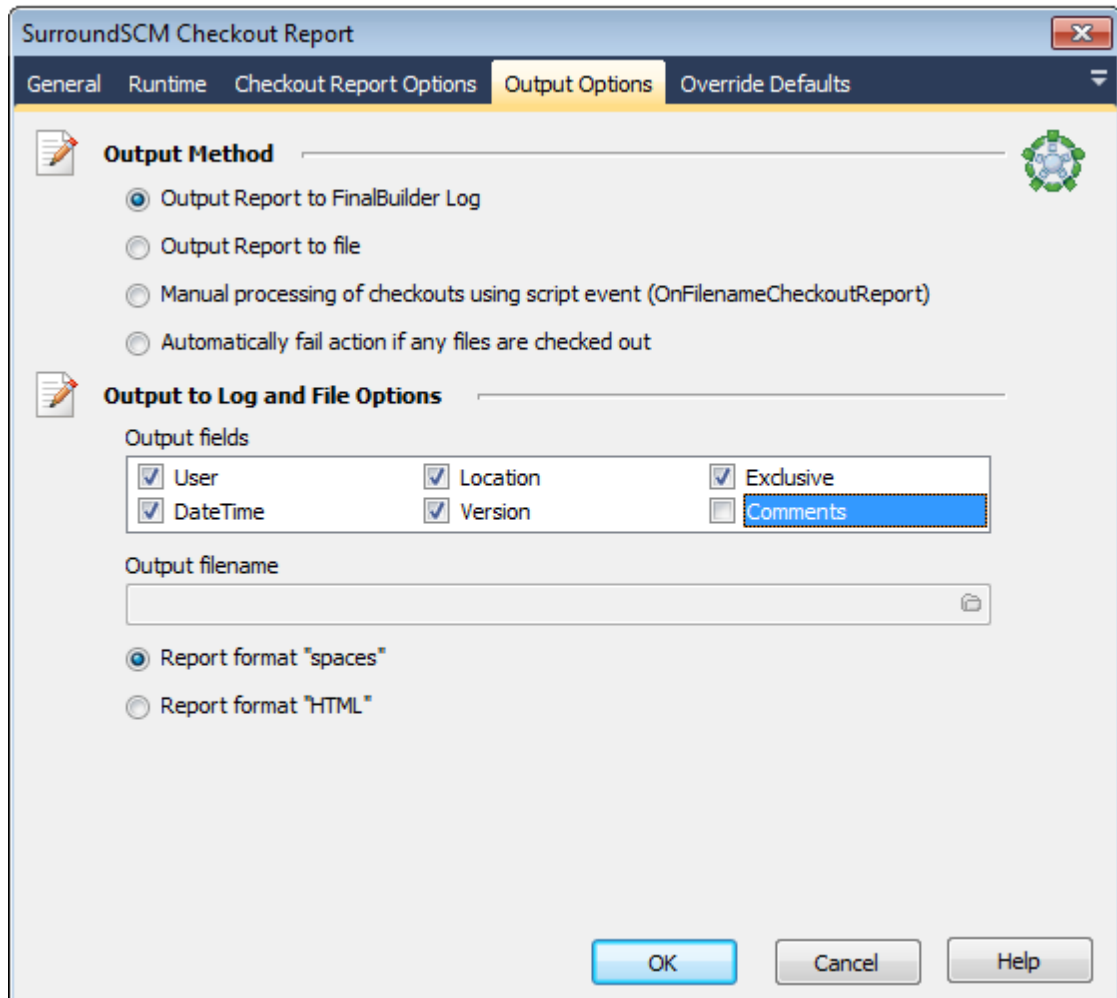
Baseline
Main

Users

Enter usernames to include in the report separated with a comma. Leave blank for all users

OK Cancel Help

Output Options



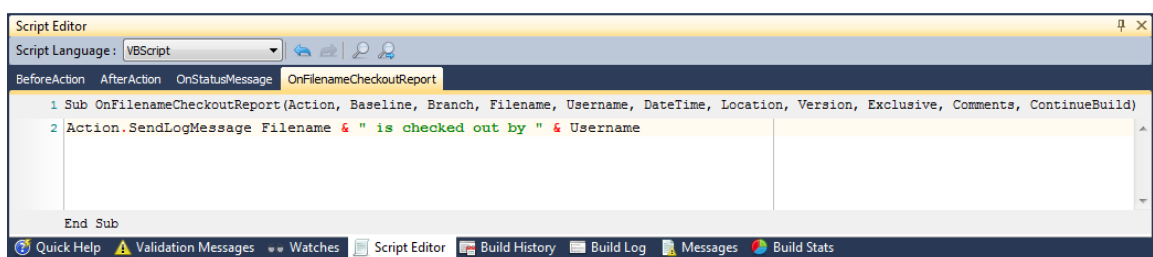
There are four possible modes when this action runs:

1. Output Report to FinalBuilder log
2. Output Report to file

Both these options allow you to specify which fields to output, eg. User, DateTime, Version etc. and also if the report should be formatted using spaces or in HTML.

3. Manual processing of checkouts using script event (OnFilenameCheckoutReport)
4. Automatically fail action if any files are checked out.

Option 3 allows the user to write script to process each checkout. The processing occurs in the OnFilenameCheckoutReport script event.



The above screen pic shows a very simple script which is executed for each file

checked out. The parameters are as follows:

Action - the action object.

Baseline - the name of the baseline where the checked out file is located

Branch - the name of the branch where the checked out file is located

Filename - the filename which is checked out

Username - the user who has the file checked out

DateTime - the date/time when the file was checked out

Location - the location of the checked out file

Version - the version of the file

Exclusive - specifies if the file is exclusively checked out

Comments - checked out comments of the file

ContinueBuild - out parameter, set this to False if you want to fail the action.

6.34.15.4 Surround SCM Create Branch

The Create Branch action allows creation of a new workspace, baseline or snapshot branch in Surround SCM.

Create Branch Tab

SurroundSCM Create Branch

General Runtime **Create Branch** Override Defaults

New branch

Branch
NewBranch

Repository Path
/Main

Options

Branch Type
workspace

Parent Branch

Comment

Make like a tree and branch.

Create branch at

☒ Latest Version

☐ Label

☐ Timestamp

format: yyyyymmddhh:mm:ss

☐ Include removed files

OK Cancel Help

Branch - enter the new branch name

Repository Path - the repository path for the new branch

Branch Type - specify workspace, baseline or snapshot

Parent Branch - specify the parent branch for the new branch

Create Branch At:

Latest Version - the most recent version of the files are used

Label - This option specifies which parent branch file versions are copied into the child branch, specified by the label.

Timestamp - Enter the timestamp (local time). This option specifies which parent branch file versions

are copied into the new child branch. Date/time format: yyyyymmddhh:mm:ss

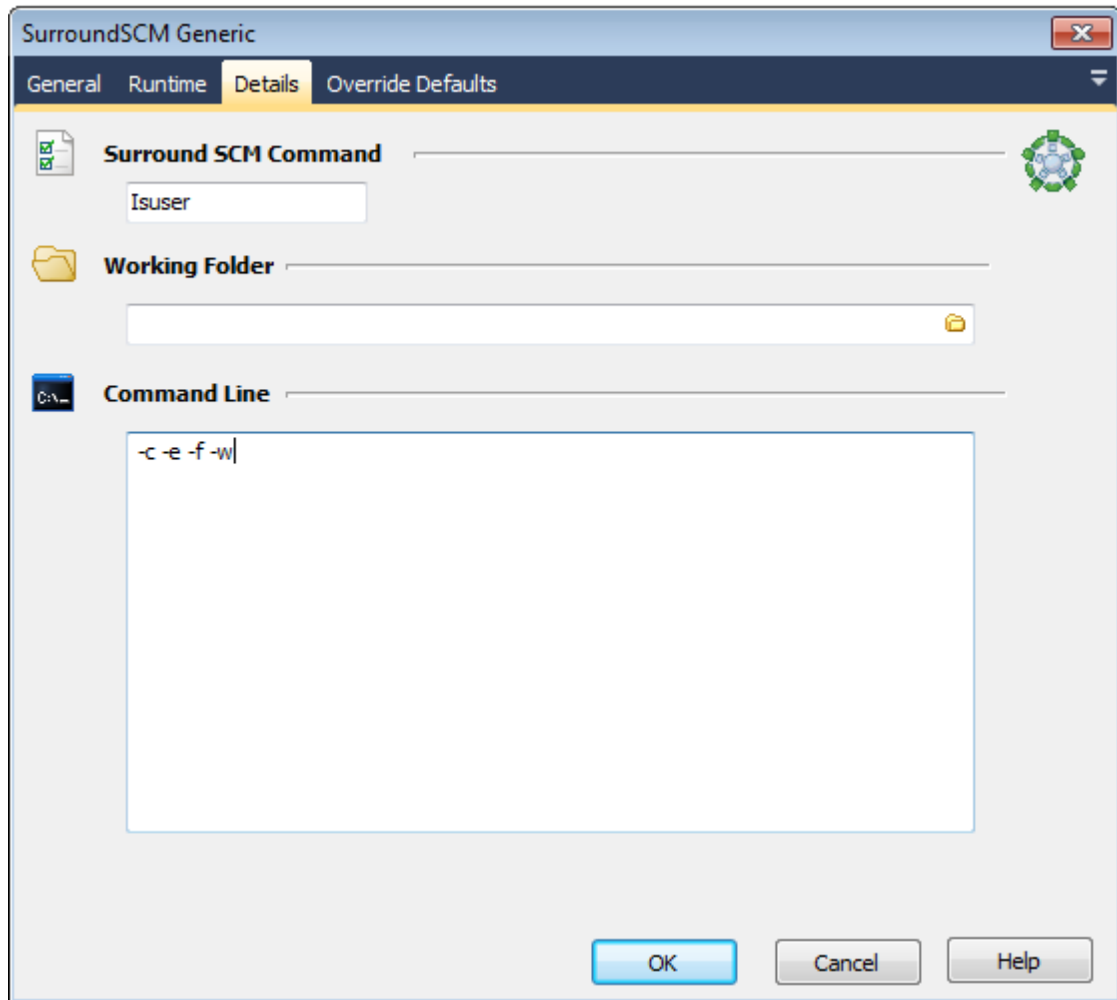
6.34.15.5 Surround SCM Freeze Branch

Freezing a branch prevents any code changes being made to files in the branch. When a branch is frozen, it is locked and no changes can be made to it.

6.34.15.6 Surround SCM Generic

If the Surround SCM operation you want to perform is not covered by the built in FinalBuilder Surround SCM actions, then you can use the Surround SCM Generic action.

The action requires you specify the SCM Command, the working folder to run the command from, and the command line to send to sscm.exe.



See the "Surround SCM CLI Guide.PDF" file for more information on the Surround SCM commands and command line options.

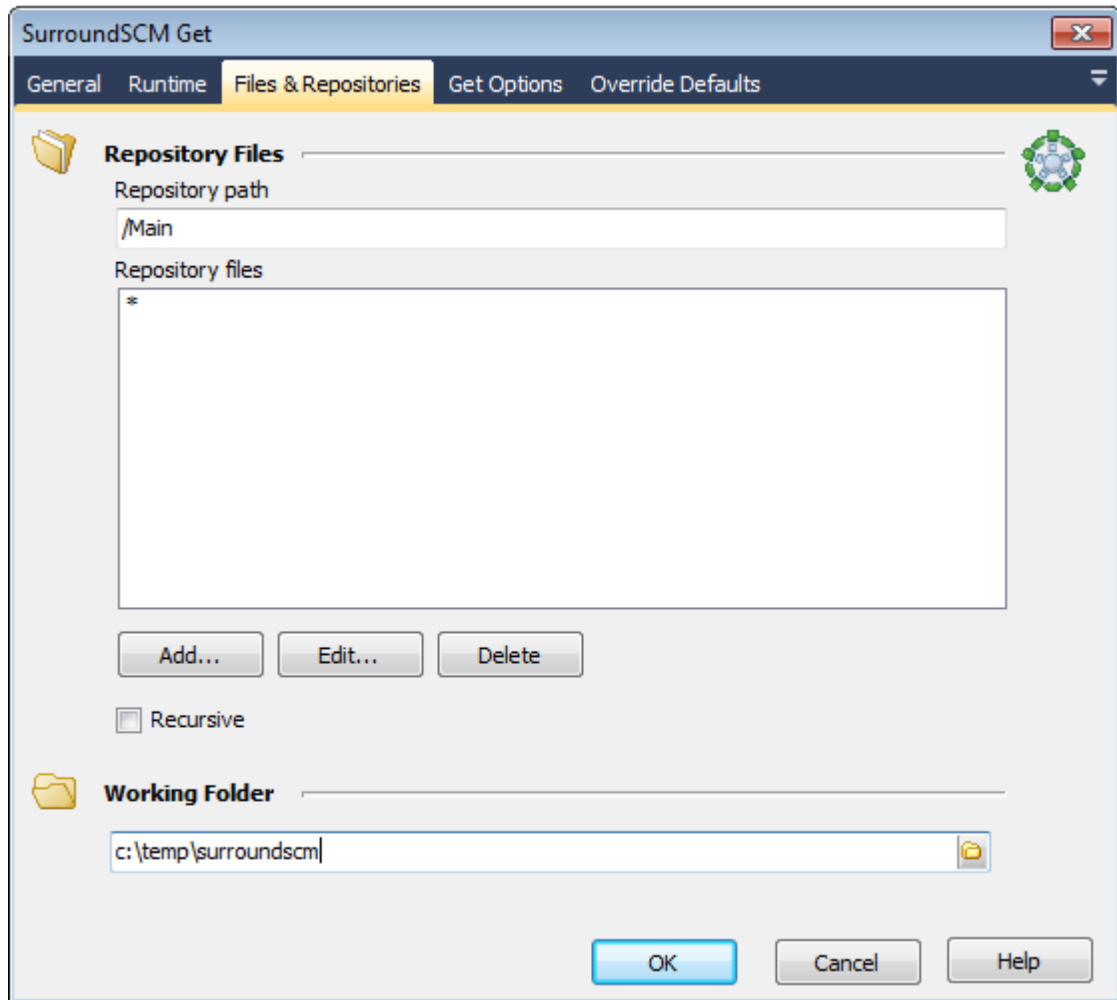
6.34.15.7 Surround SCM Get

Get files when you want to view a file but do not need to make any changes. You can get a single file, multiple files, or a repository. A read-only copy of the file is created in the specified directory.

Files & Repositories Tab

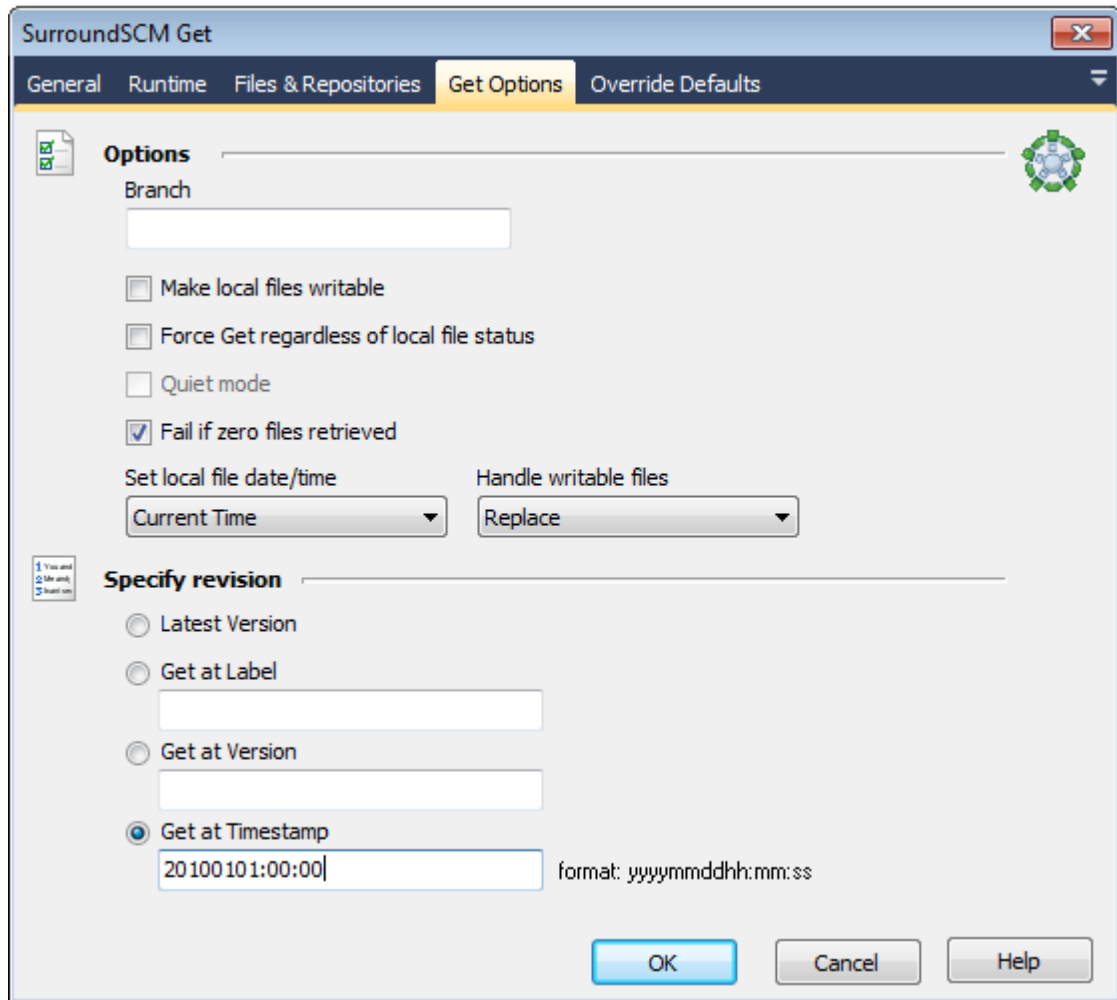
Specify the repository path and a list of files to retrieve. You can use * to specify all files.

Specify Recursive to Recursively get files and subrepositories.



Get Options Tab

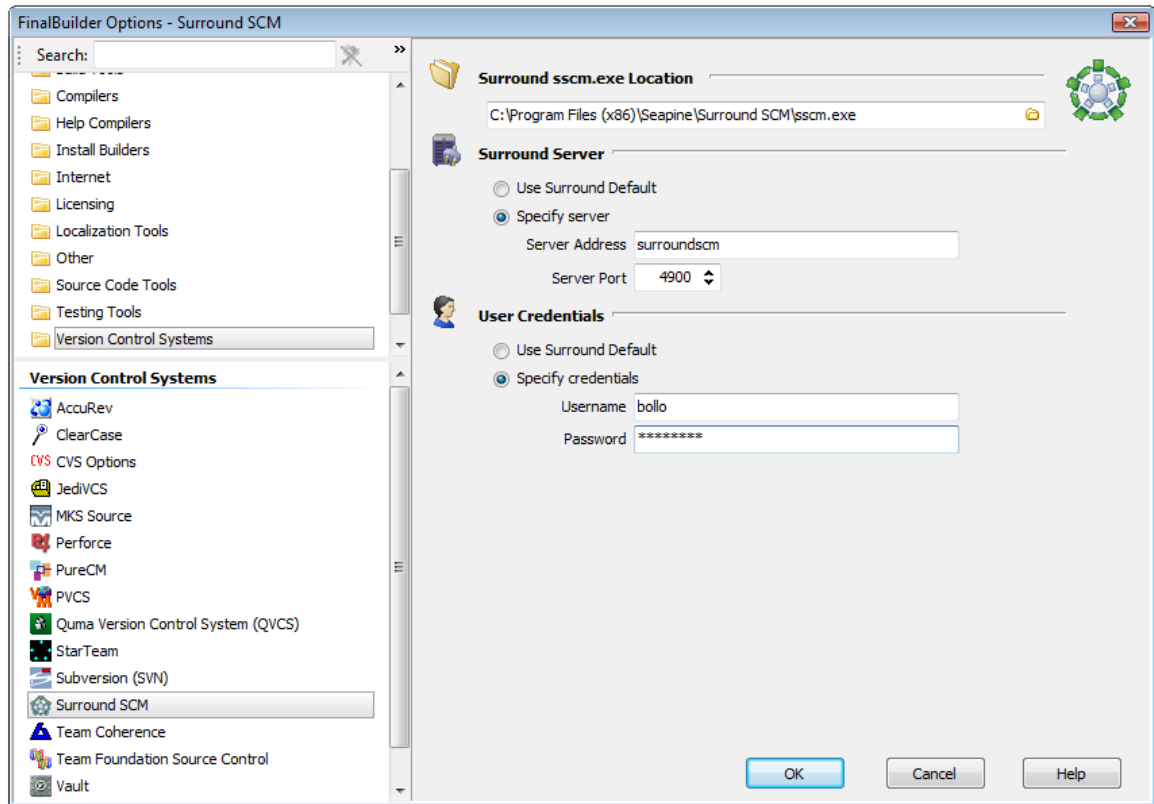
Specify the Get options - all values are optional.

**Tips:**

If you don't specify a working folder, then Surround will use the current working folder for the repository selected.

6.34.15.8 Surround SCM Global Options

The Surround SCM Global Options allow you to set up your FinalBuilder integration with default options for any Surround SCM actions used in any build, as well as the path to the Surround SCM command line interface (this is what FinalBuilder uses to perform the Surround SCM actions).



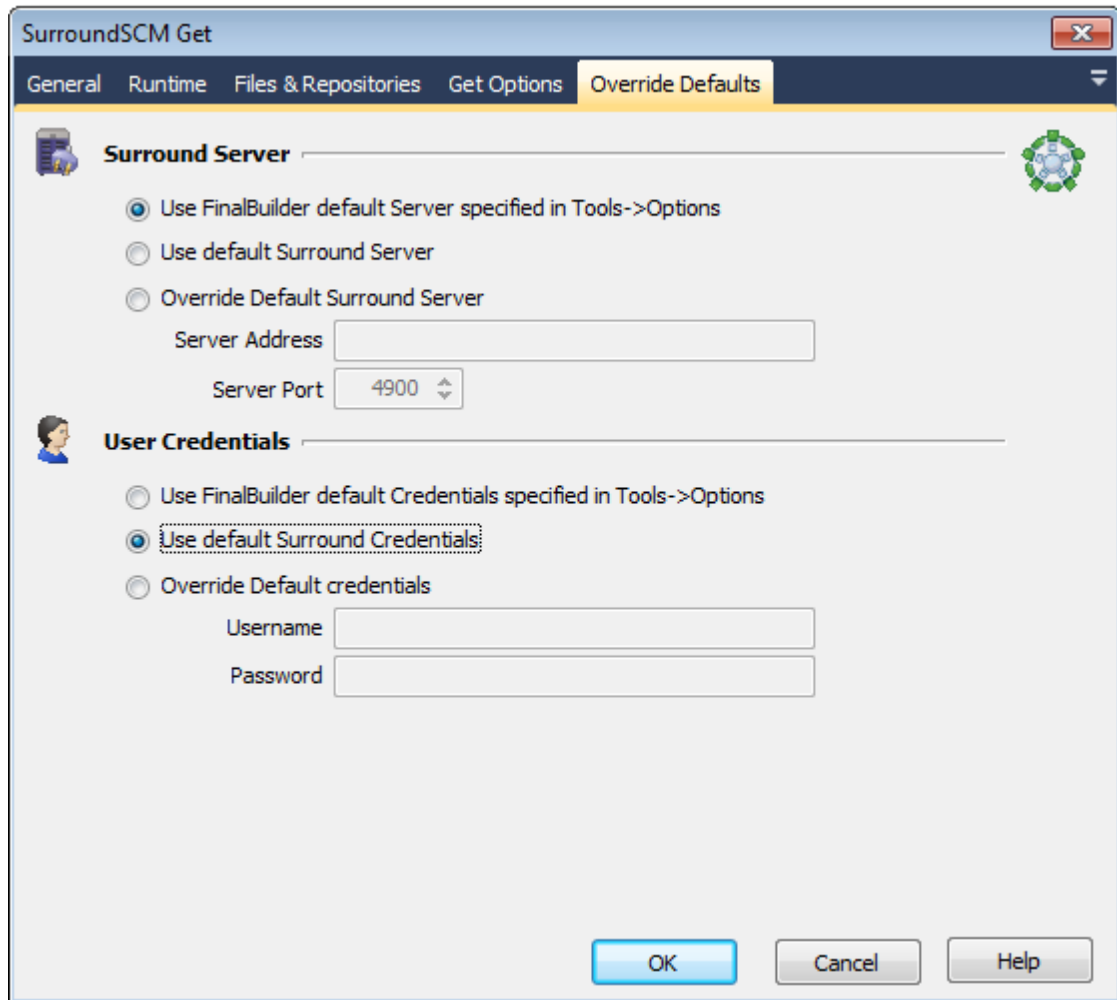
Access the Global Options for Surround via the Tools->Options menu in FinalBuilder. Then select the Version Control System category, and finally the Surround item.

FinalBuilder will attempt to auto detect the sscm.exe location - if the Surround sscm.exe Location is blank, then specify the location of sscm.exe.

You can specify both the default server and user credentials for all the Surround SCM actions. Either specify the specific values, or use the Surround SCM defaults (in this case no values will be specified on the command line). Note that in the any Surround SCM action, you can either use the global options specified here, or override the values.

6.34.15.8.1 Surround SCM Override Global Options

Each Surround SCM action in FinalBuilder allows you to override the Global Surround SCM options.



To specify the FinalBuilder global options, see Surround SCM Global Options

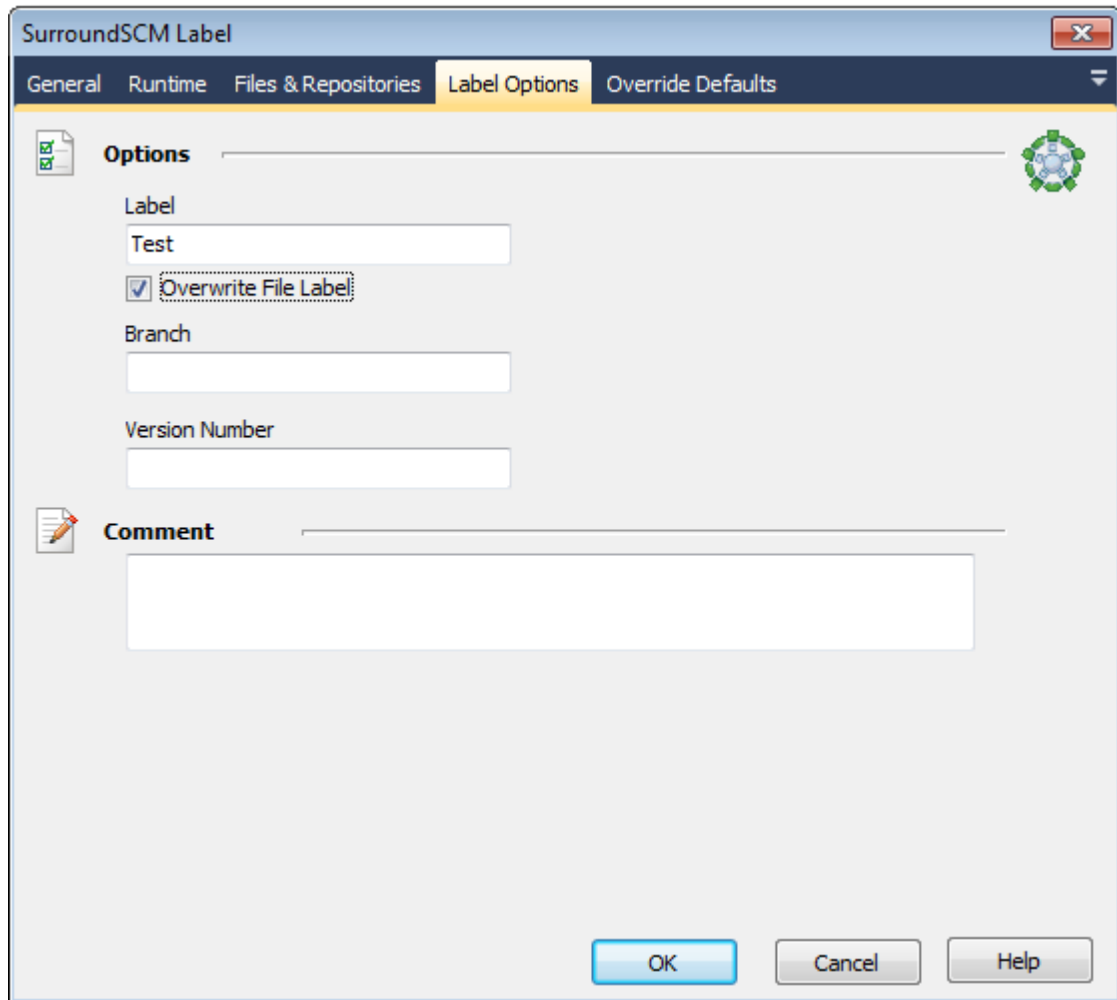
6.34.15.9 Surround SCM Label

Labels provide a way to mark a specific version of a file or repository. When you create a label, a new entry is created in the history. The file and the version number do not change.

The Label action uses the same Files & Repositories Tab as the Surround SCM Get action.

Label Options Tab

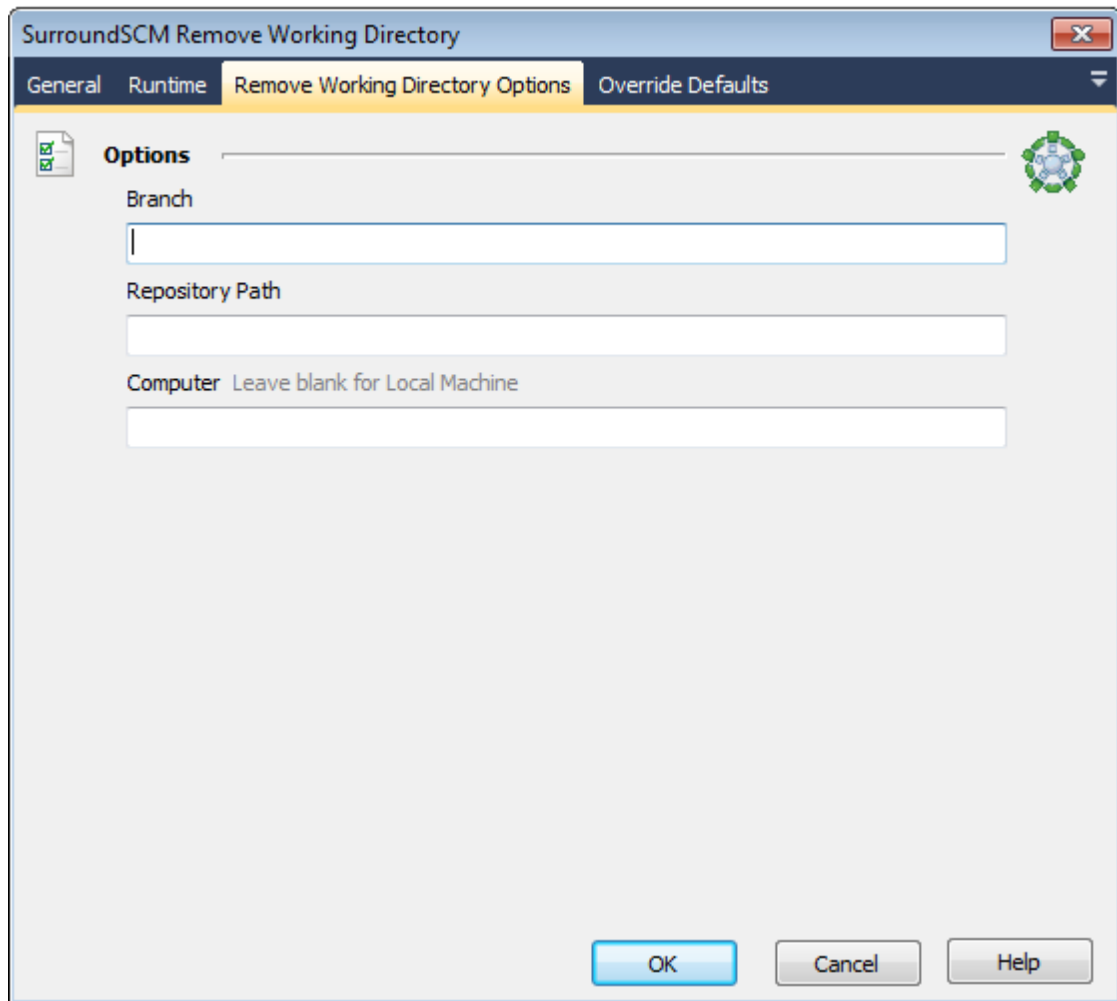
Specify the new label values - all values are optional except the Label name.



Tip: To label all files in a repository then use "/" as the repository file spec (without quotes).

6.34.15.1(Surround SCM Remove Working Directory

The Remove Working Directory action allows you to remove the working directory of a Repository.

**Branch**

The branch that contains the repository to remove the working directory from.

Repository Path

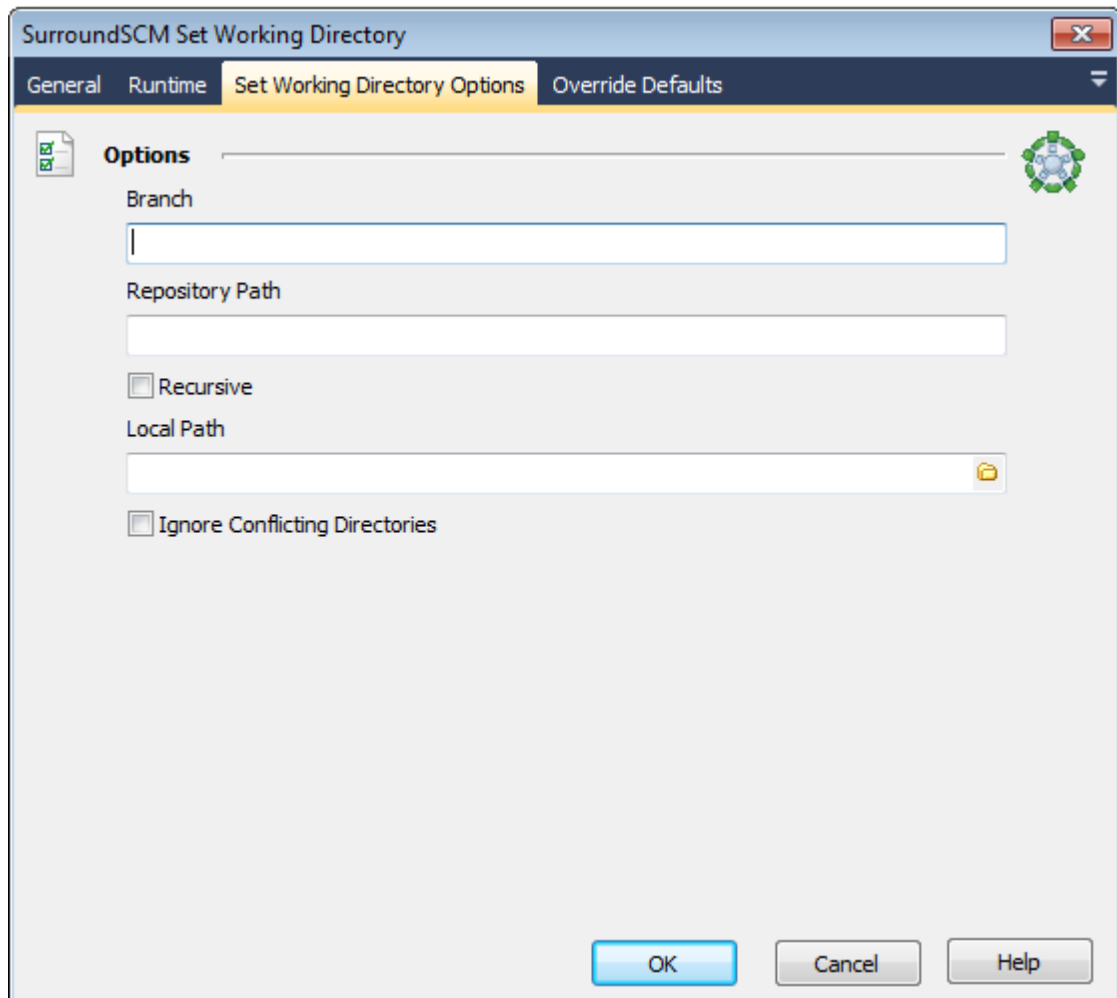
The full path to the repository, including parent repositories.

Computer

The computer that the working directory is on.

6.34.15.1 Surround SCM Set Working Directory

The Set Working Directory action allows you to set the working directory of a Repository.

**Branch**

The branch that contains the repository to set the working directory for.

Repository Path

The full path to the repository, including parent repositories.

Recursive

Child repositories will inherit this working directory.

Local Path

The full path to the directory where the repository will be set to use.

Ignore Conflicting Directories

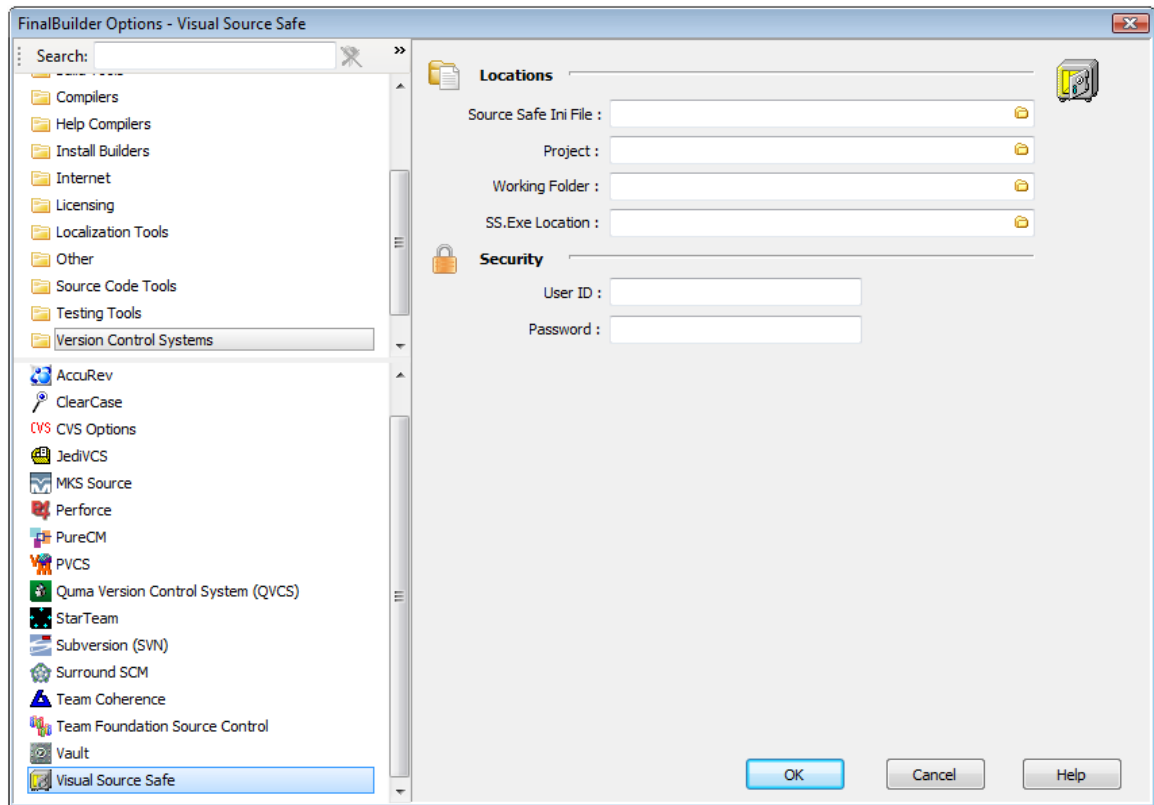
Will force the repository to use the specified directory, even though another repository is set to use it.

6.34.15.1: Surround SCM Unfreeze Branch

The Unfreeze Branch allows you to unlock a frozen branch during your build process.

6.34.16 Source Safe

The Source Safe global options page contains settings that apply to all the Source Safe actions.



Locations

The location of the files needed by the Source Safe actions.

Security

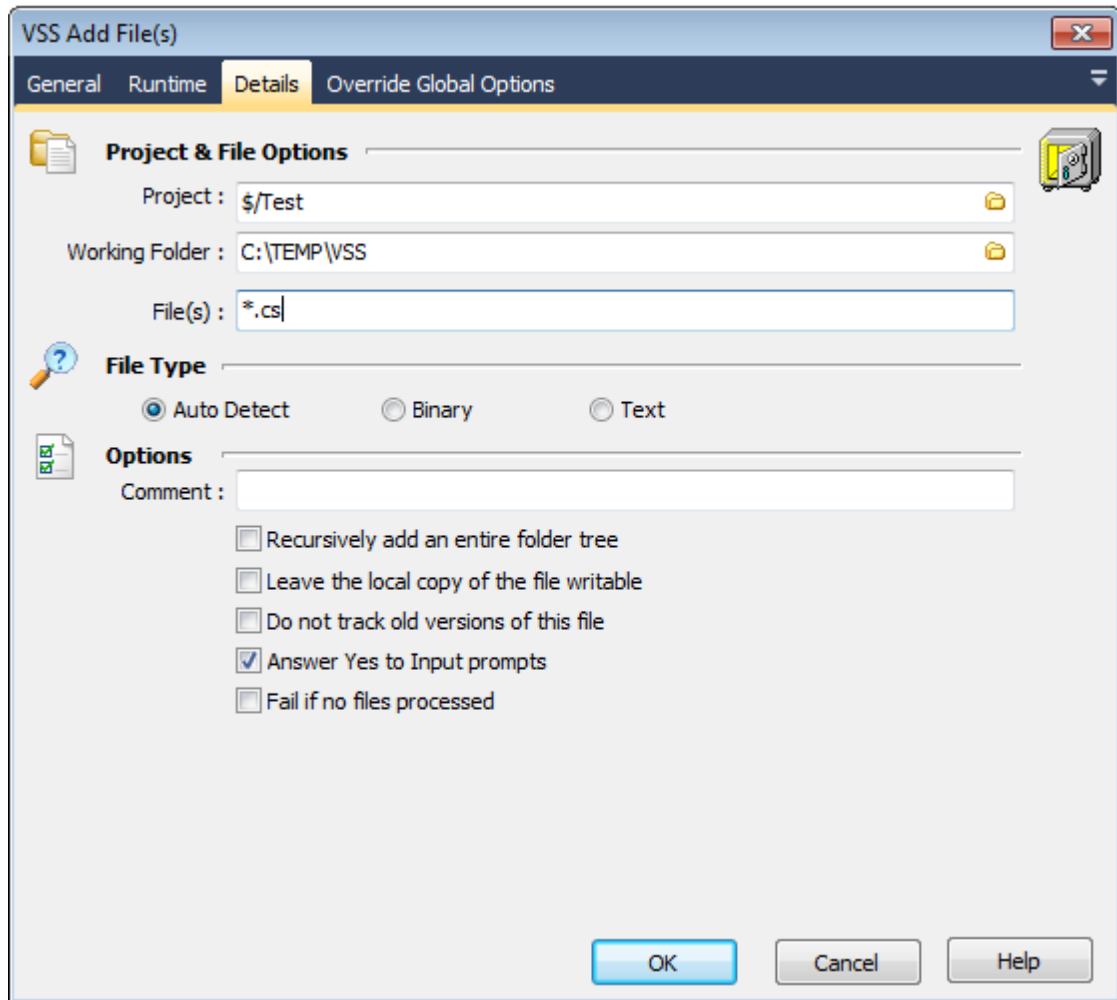
The security credentials to be used by the Source Safe actions.

NOTE: The Visual SourceSafe actions cannot be used under Async group as the actions temporarily modify the SourceSafe INI files.

6.34.16.1 Source Safe Add Files Action

Adds Files to a Source Safe Project.

How to override Global Options for the Source Safe actions.

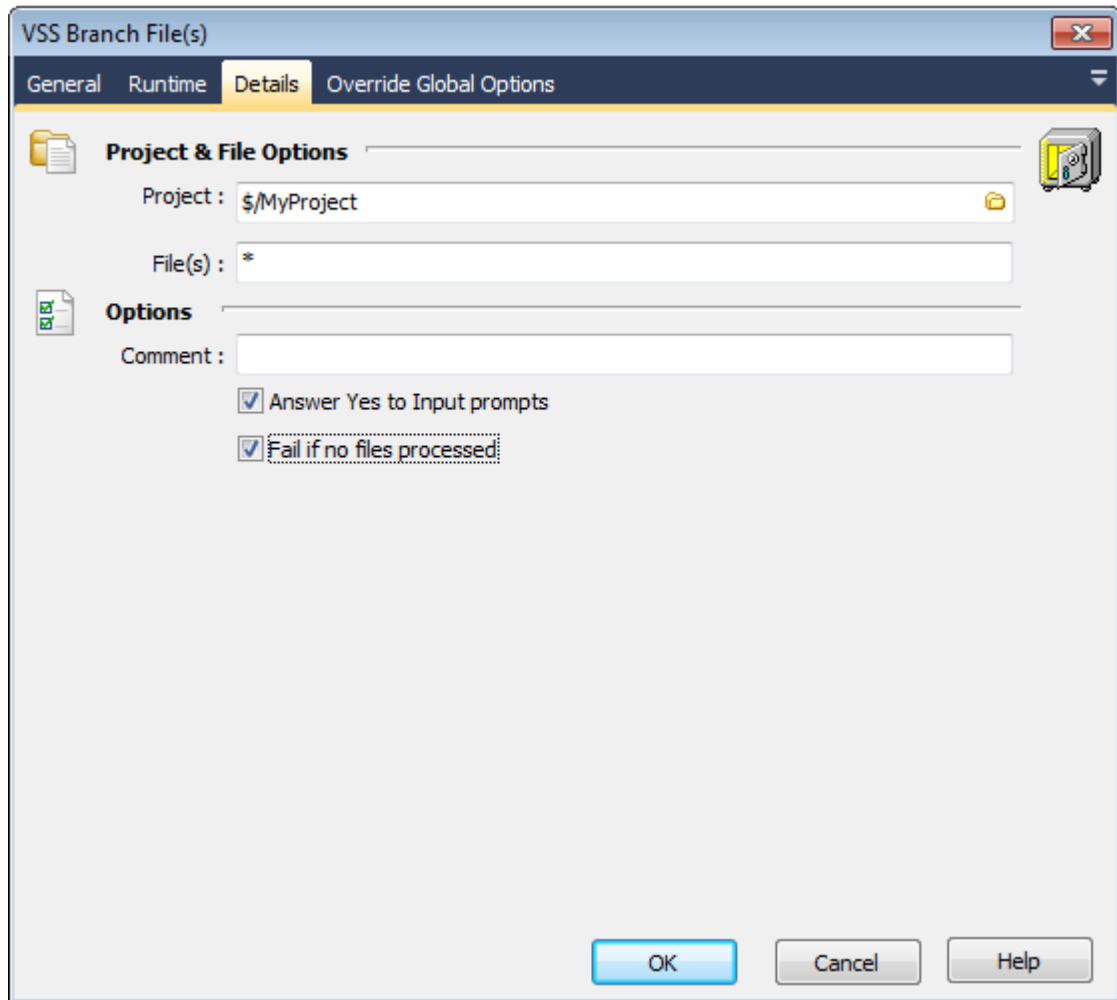


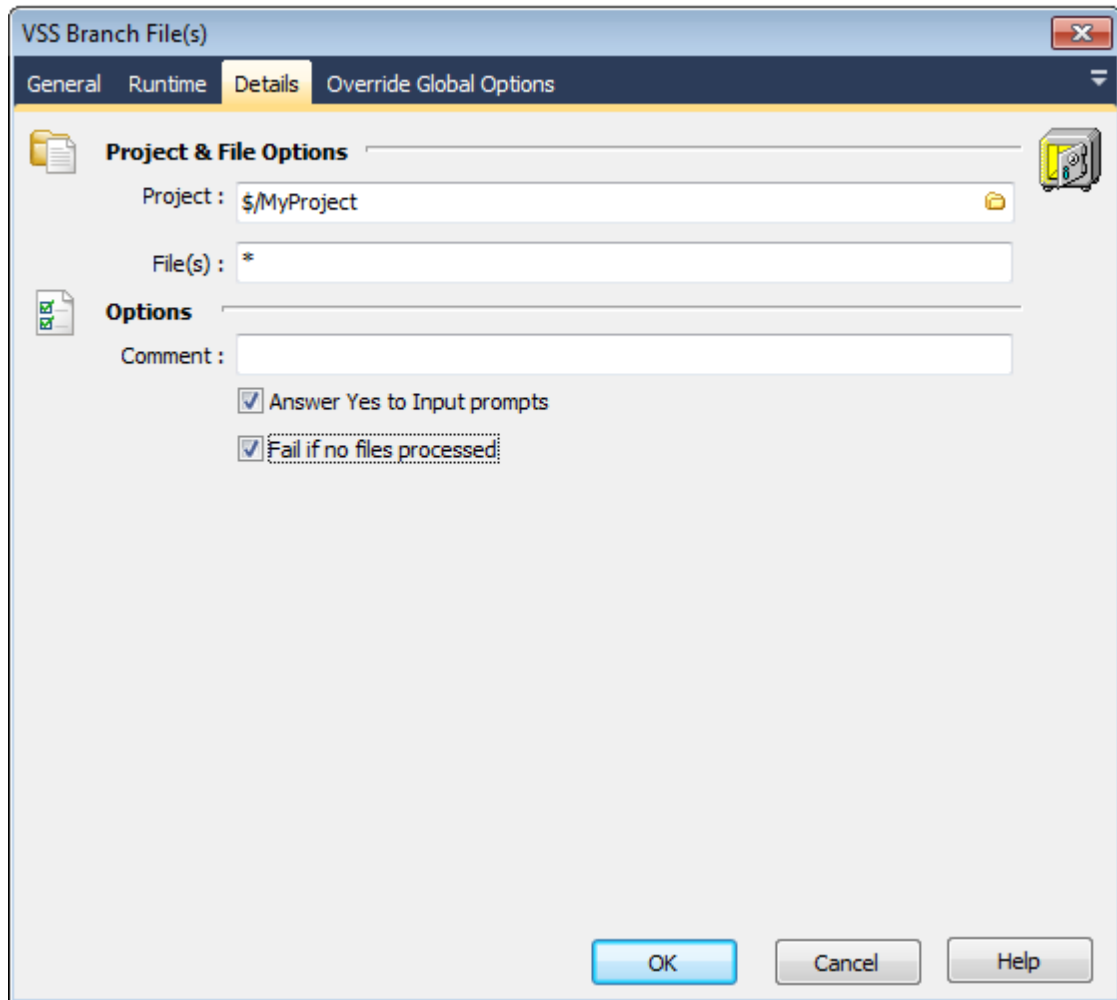
6.34.16.2 Source Safe Branch

The VSS Branch Files action will branch shared files in the specified project.

Branching severs the link between the original copy of the file and the shared file. See Source Safe Share.

How to override Global Options for the Source Safe actions.

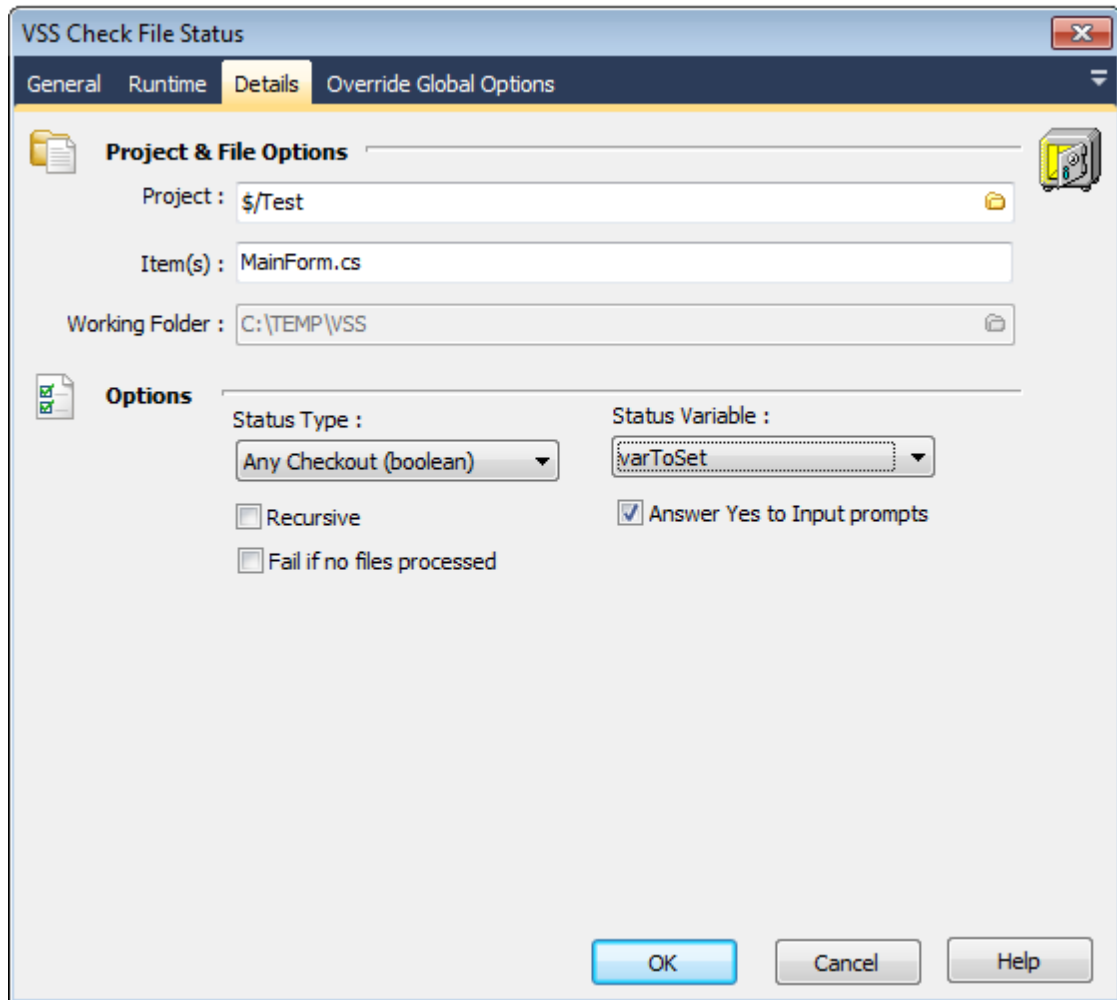




6.34.16.3 Source Safe Check File Status

This Action allows you to check the status of files in a source safe project. The status is read into a FinalBuilder variable which can then be used elsewhere in the build.

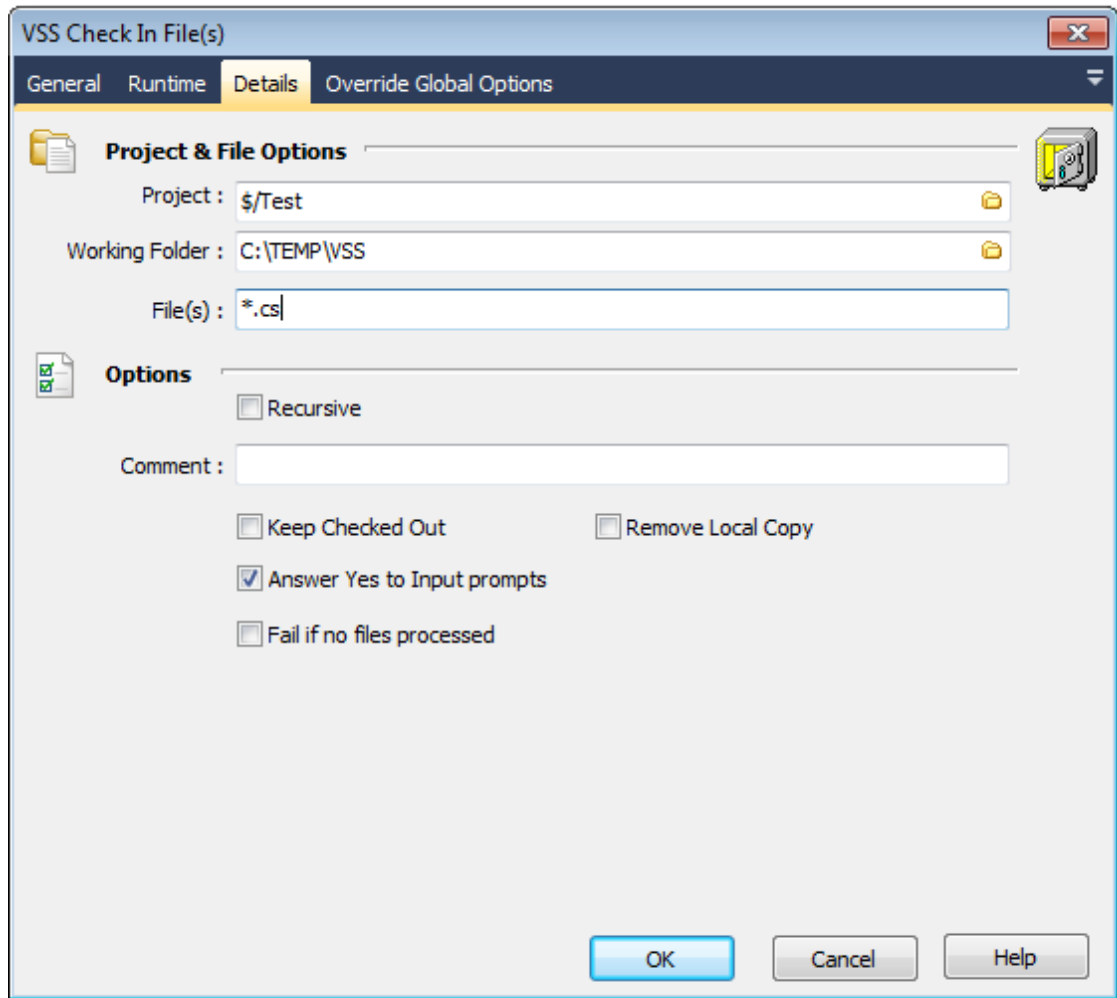
How to override Global Options for the Source Safe actions.



6.34.16.4 Source Safe Check In File(s) Action

This action allows you to check in files to a Visual Source Safe database. For detailed descriptions of the options available, see your Visual Source Safe 6.0 documentation or MSDN.

How to override Global Options for the Source Safe actions.



Scripting Info

The Action properties available are :

property SSProject : WideString;// The name of the source safe project, eg. \$/test/
subproject

property WorkingFolder : WideString;// The folder where the local copy will be placed.

property FileSpec : WideString;// The file specification for files to check out , you may use
wildcards in this

property KeepCheckedOut : WordBool;// Keep File Checked Out.

property RemoveLocalCopy : WordBool;// Remove the local copy after check in.

property Comment; //Add a comment when checking file in.

property Recursive : WordBool;// Recurse sub projects, checking in files/project which
match the filespec

property OverrideWorkingFolders; //override the working folders when recursively
checking in projects

property UserID : WideString;// Source Safe User ID, only needed if you do not want to
use the default

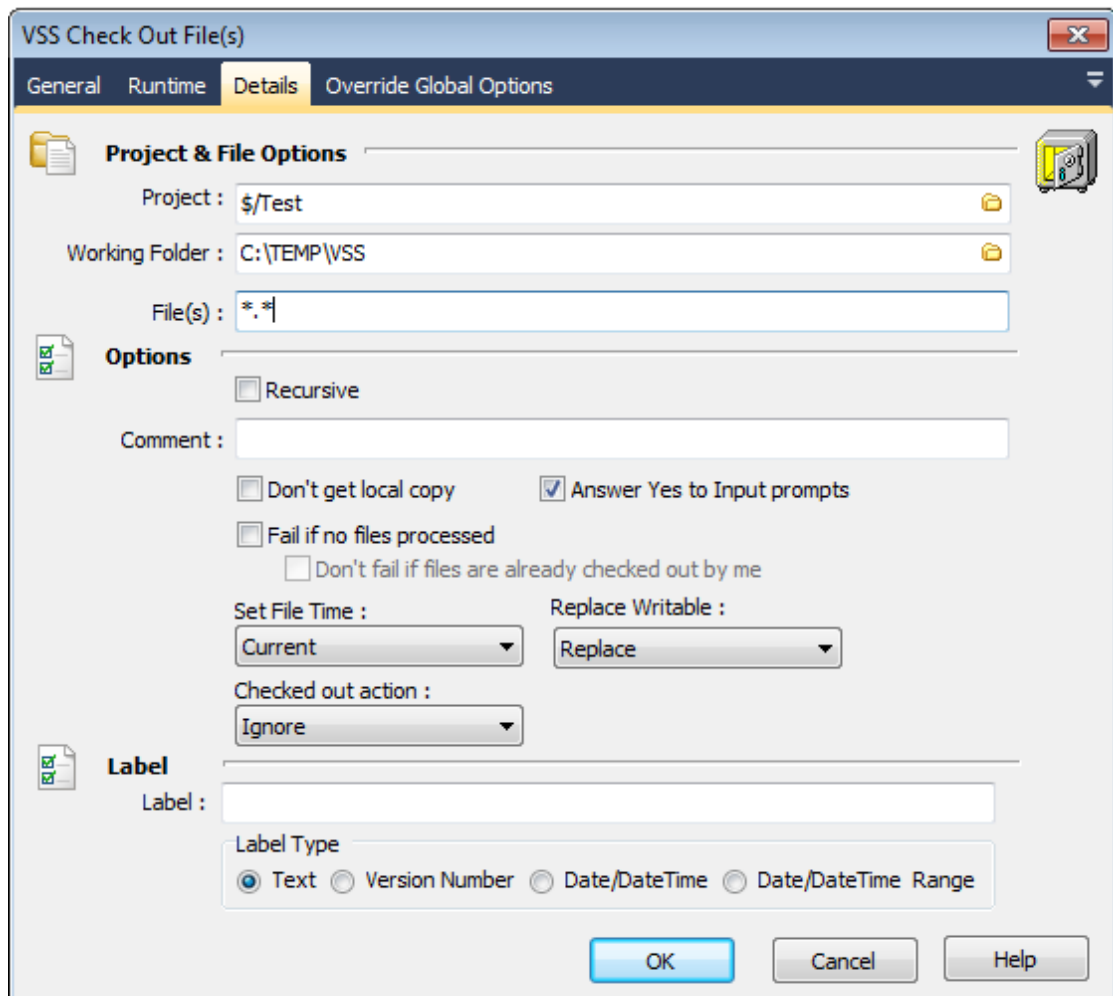
property Password : WideString;// Source Safe pwd for above user id.

property IniFile : WideString;// path to source safe ini file (if you do not want to use the
default

6.34.16.5 Source Safe Check Out File(s) Action

This action allows you to check out files from a Visual Source Safe database. For detailed descriptions of the options available, see your Visual Source Safe 6.0 documentation or MSDN.

How to override Global Options for the Source Safe actions.



Scripting Info

The Action properties available are :

- property** Recursive : WordBool;// Recurse sub projects, checking out files/project which match the filespec
- property** FileSpec : WideString;// The file specification for files to check out , you may use wildcards in this
- property** WorkingFolder : WideString;// The folder where the local copy will be placed.
- property** SSProject : WideString;// The name of the source safe project, eg. \$/test/subproject
- property** BuildTree : WordBool;// Build the local folders using the project names, overrides the working folders
- property** FileTime : integer; // Determines which time to set on files,
valid values are : ssDefault,ssCurrent,ssModification,
- ssCheckIn
- property** ReplaceWritable : integer;// Determines how to handle local files that are already writeable,

// valid values are : ssReplace, ssSkip, ssMerge

property UserID : WideString;// Source Safe User ID, only needed if you do not want to use the default

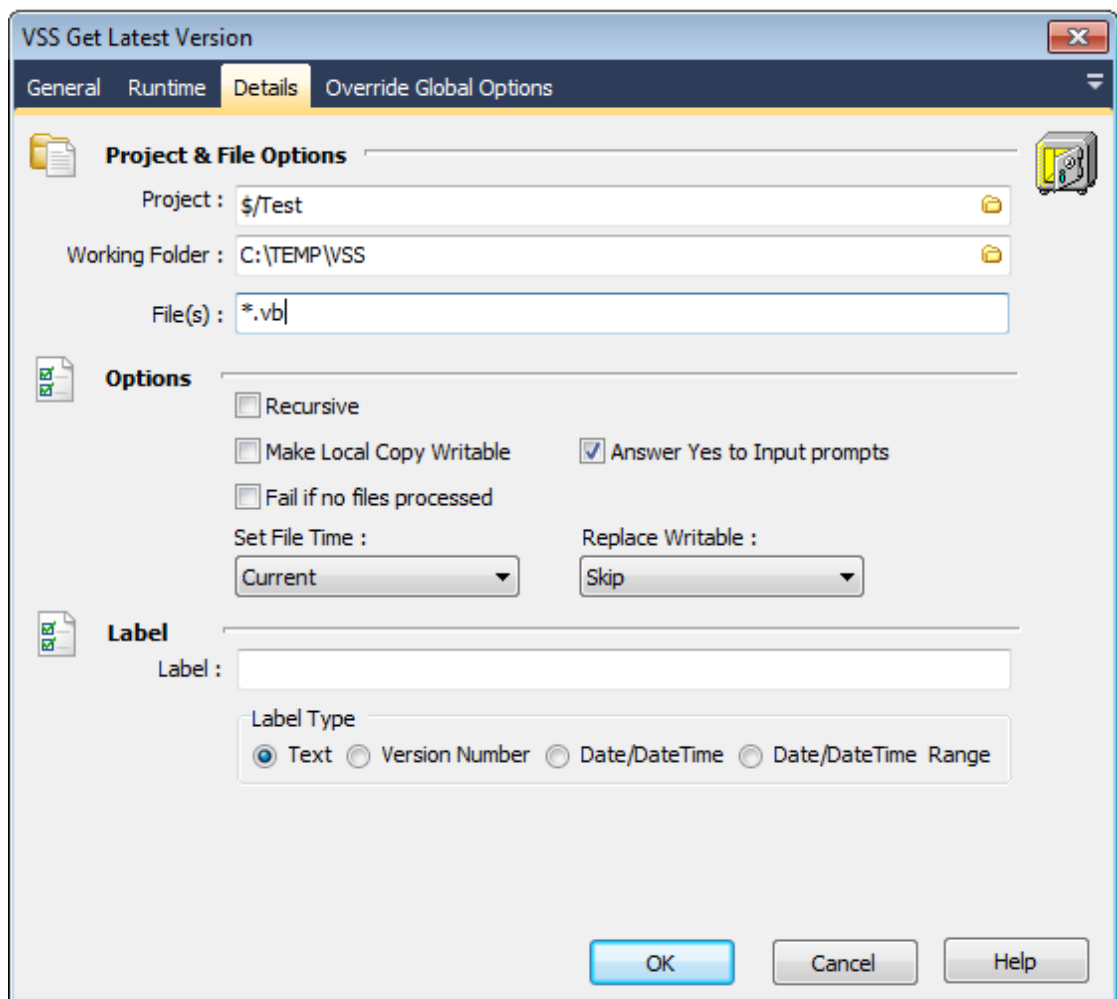
property Password : WideString;// Source Safe pwd for above user id.

property IniFile : WideString;// path to source safe ini file (if you do not want to use the default

6.34.16.6 Source Safe Get Latest Version Action

This action allows you to Get the Latest Version of files or projects from a Visual Source Safe database. For detailed descriptions of the options available, see your Visual Source Safe 6.0 documentation or MSDN.

How to override Global Options for the Source Safe actions.



You can specify a date string in the Label field. If a date and time is specified, then source safe will get the versions of the specified files with that exact date & time, if there is no version with that date & time then the file will not be retrieved. If you specify just a date, then source safe will get the latest versions of the specified files as at the specified date.

Scripting Info

The Action properties available are :

property SSProject : WideString;// The name of the source safe project, eg. \$/test/
subproject
property WorkingFolder : WideString;// The folder where the local copy will be placed.
property FileSpec : WideString;// The file specification for files to check out , you may use
wildcards in this
property KeepCheckedOut : WordBool;// Keep File Checked Out.
property RemoveLocalCopy : WordBool;// Remove the local copy after check in.
property Comment; //Add a comment when checking file in.
property Recursive : WordBool;// Recurse sub projects, checking in files/project which
match the filespec
property BuildTree : WordBool;// Build the local folders using the project names,
overrides the working folders
property FileTime : integer; // Determines which time to set on files,
valid values are : ssDefault,ssCurrent,ssModification,
ssCheckIn
property ReplaceWritable : integer;// Determines how to handle local files that are
already writeable,
// valid values are : ssReplace, ssSkip, ssMerge
property UserID : WideString;// Source Safe User ID, only needed if you do not want to
use the default
property Password : WideString;// Source Safe pwd for above user id.
property IniFile : WideString;// path to source safe ini file (if you do not want to use the
default

6.34.16.7 Source Safe Get Working Directory

The Get Working Directory action attempts to retrieve the working directory for the specified project using the source safe ini files.

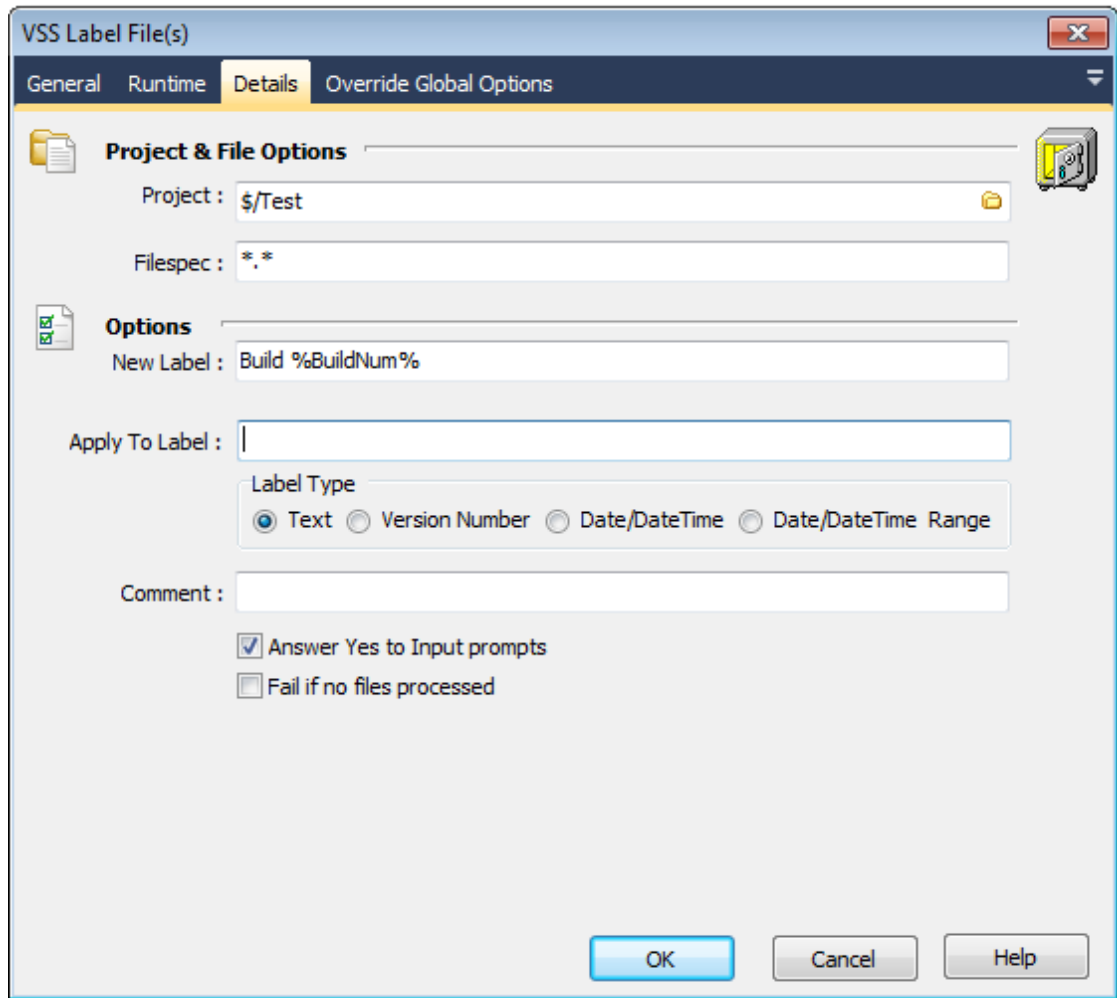
The working directory will be output to the FB Log and can be optionally written to a FinalBuilder variable.

Note: If you choose to get recursively (ie enable "Recursive") and are overriding the Working Directory (ie choosing a different working directory to the default), then you will need to check "Build Tree" in order to override all working directories in the Project (the Build Tree option appears when you select "Recursive".)

6.34.16.8 Source Safe Label File(s)

This action allows you to Label the latest version of a project or files in a Visual Source Safe database. For detailed descriptions of the options available, see your Visual Source Safe 6.0 documentation or MSDN.

How to override Global Options for the Source Safe actions.



Scripting Info

The Action properties available are :

property SSProject : WideString;// The name of the source safe project, eg. \$/test/subproject

property SSLabel : WideString;// The Label to apply

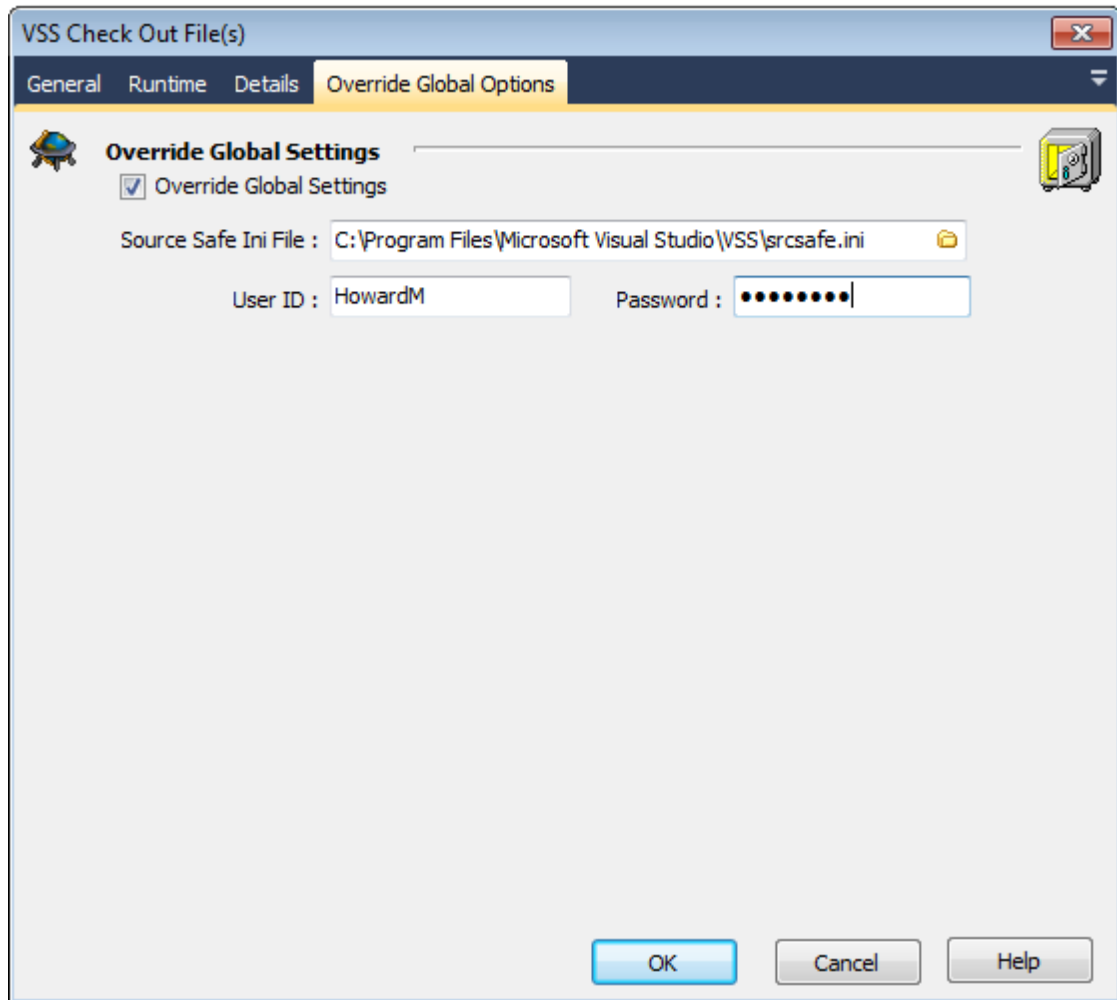
property FileSpec : WideString;// The file specification for files to label , you may use wildcards in this after check in.

property Comment; //Add a comment when Labelling a file or project.

6.34.16.9 Source Safe Override Global Options

Each of the source safe actions allow you to override the global options. This can be used to access more than one Source Safe database.

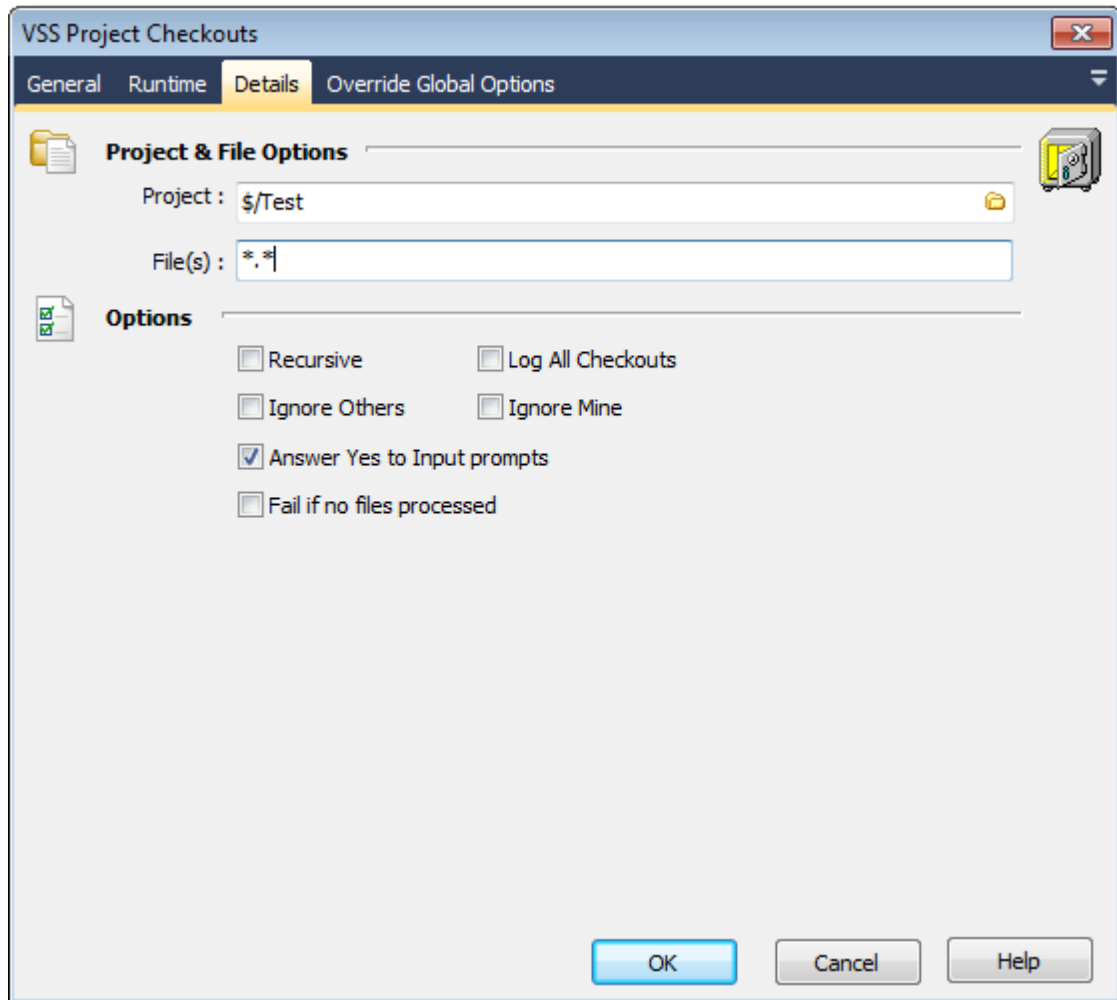
To set the global options, open the Options dialog (Tools menu) and then open the Version Control Systems category and select Visual Source Safe. The global options will then be used for any VSS action which doesn't have the "Override Global Settings" turned on.



6.34.16.1 Source Safe Project Checkouts

This action checks a Source Safe project to see if any files are checked out. If it finds any files checked out then it will fail. You can choose to ignore files checked out by yourself.

How to override Global Options for the Source Safe actions.

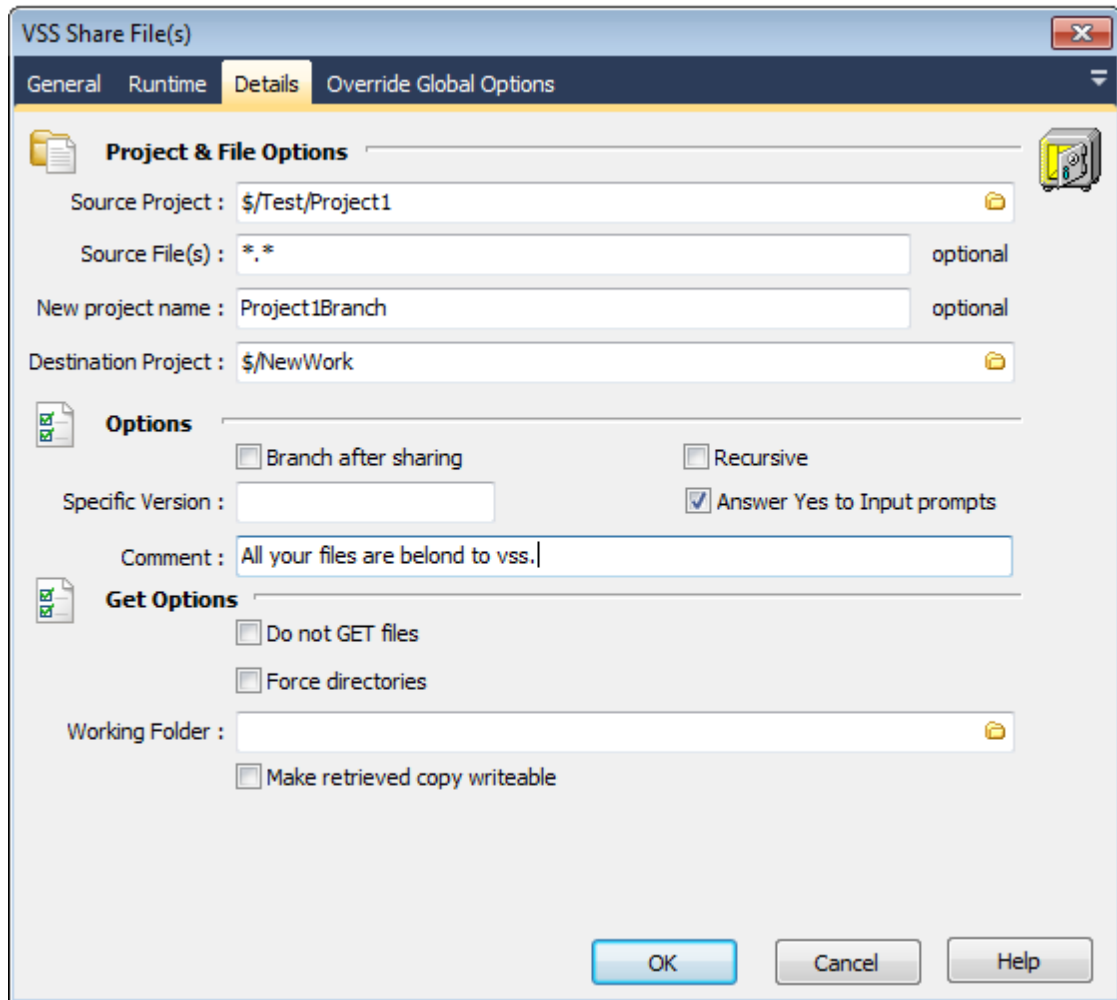


6.34.16.1 Source Safe Share

The VSS Share files action enables you to share a project or files to a new project in Source Safe.

Sharing files does not create a new file, but creates a link back to the original file. You can branch the shared files immediately by specifying the "Branch after sharing" option. Or you can branch later using the Source Safe Branch action.

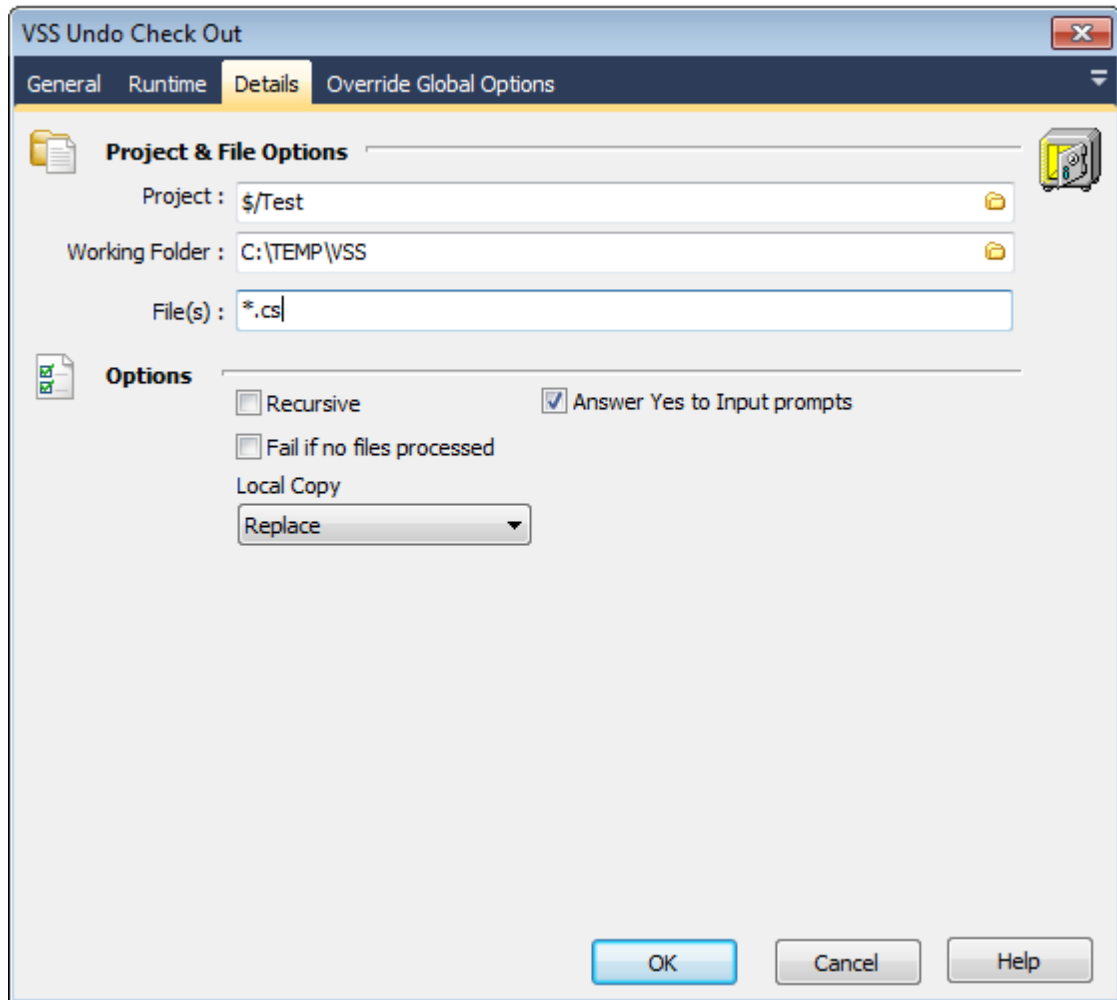
How to override Global Options for the Source Safe actions.



6.34.16.1 Source Safe Undo CheckOut Action

This action allows you to undo the check out of files from a Visual Source Safe database. For detailed descriptions of the options available, see your Visual Source Safe 6.0 documentation or MSDN.

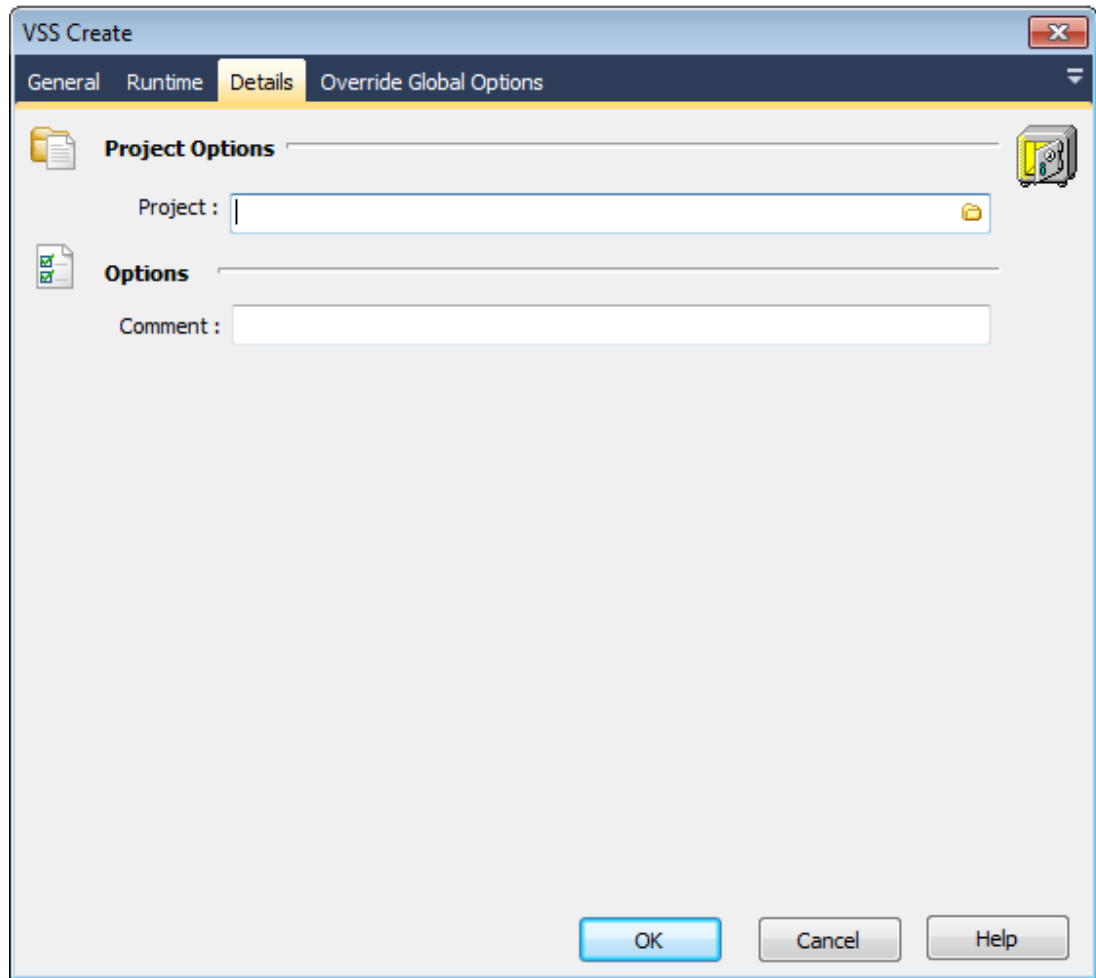
How to override Global Options for the Source Safe actions.



6.34.16.1:Source Safe Create Action

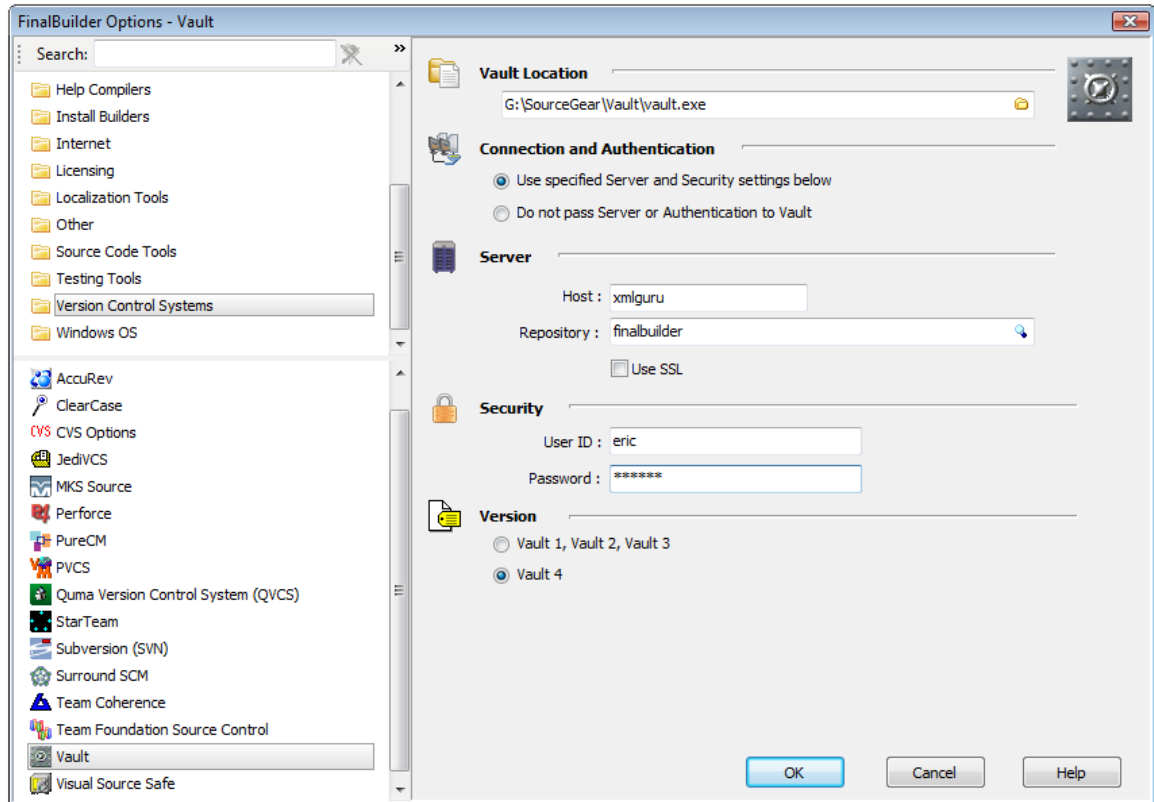
The VSS Create action will create a new sub-project within a repository.

How to override Global Options for the Source Safe actions.



6.34.17 SourceGear Vault Actions

These actions support SourceGear Vault 1.2 or higher. They provide an easy to use wrapper around the Vault command line tool too. Before using these actions you need to check that the path to the Vault command line tool is set in the FinalBuilder Options Dialog. FinalBuilder will attempt to find the path automatically however it is not always possible (depends on your installation).



The default server and security options can be overridden in the individual actions.

Vault Check Out

General Runtime CheckOut **Server Options**

Connection and Authentication

☐ Use Global Settings

☐ Do not pass Server or Authentication

☒ Override Global Settings

Server

Host : localhost

Repository : FinalBuilder

☐ Use SSL

Security

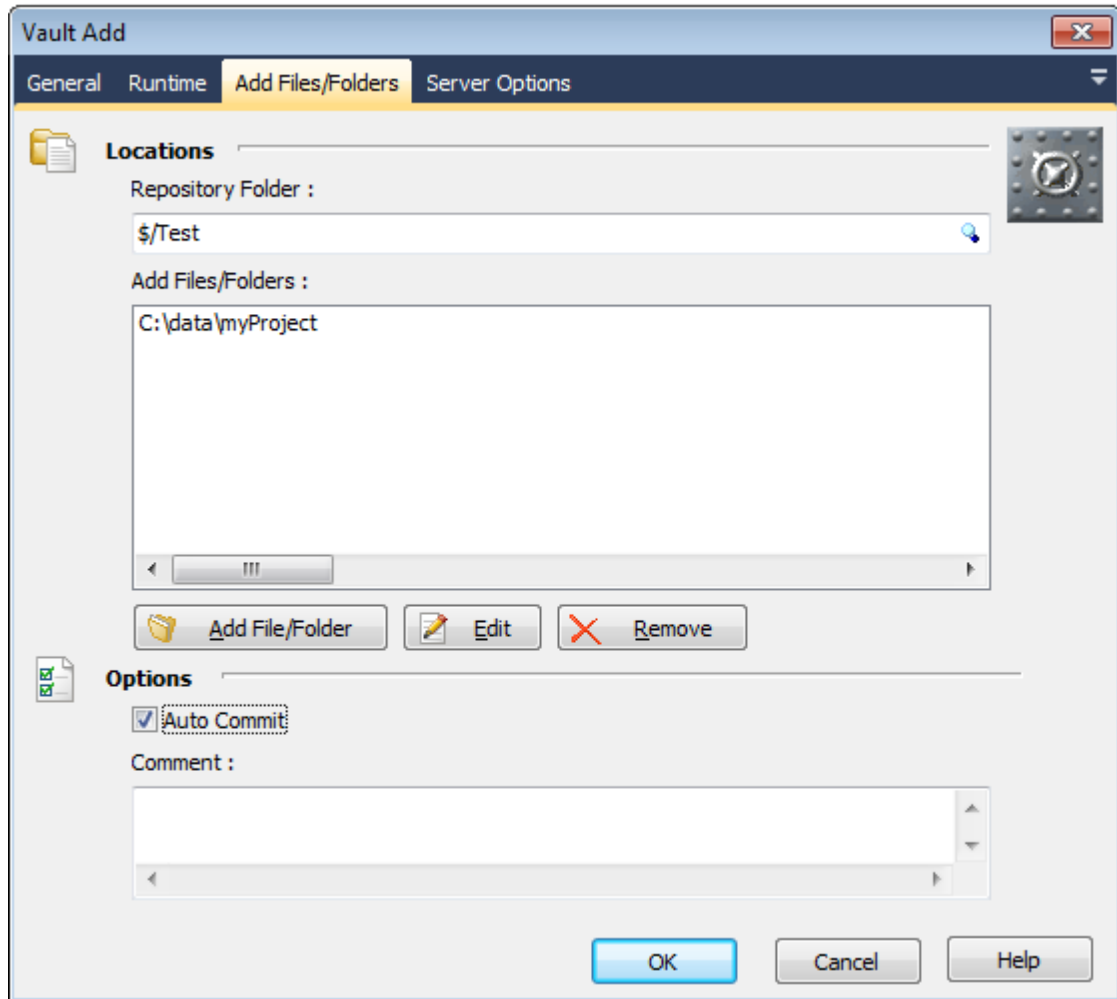
User ID : evilMastermind

Password : ●●●●●●●●

OK Cancel Help

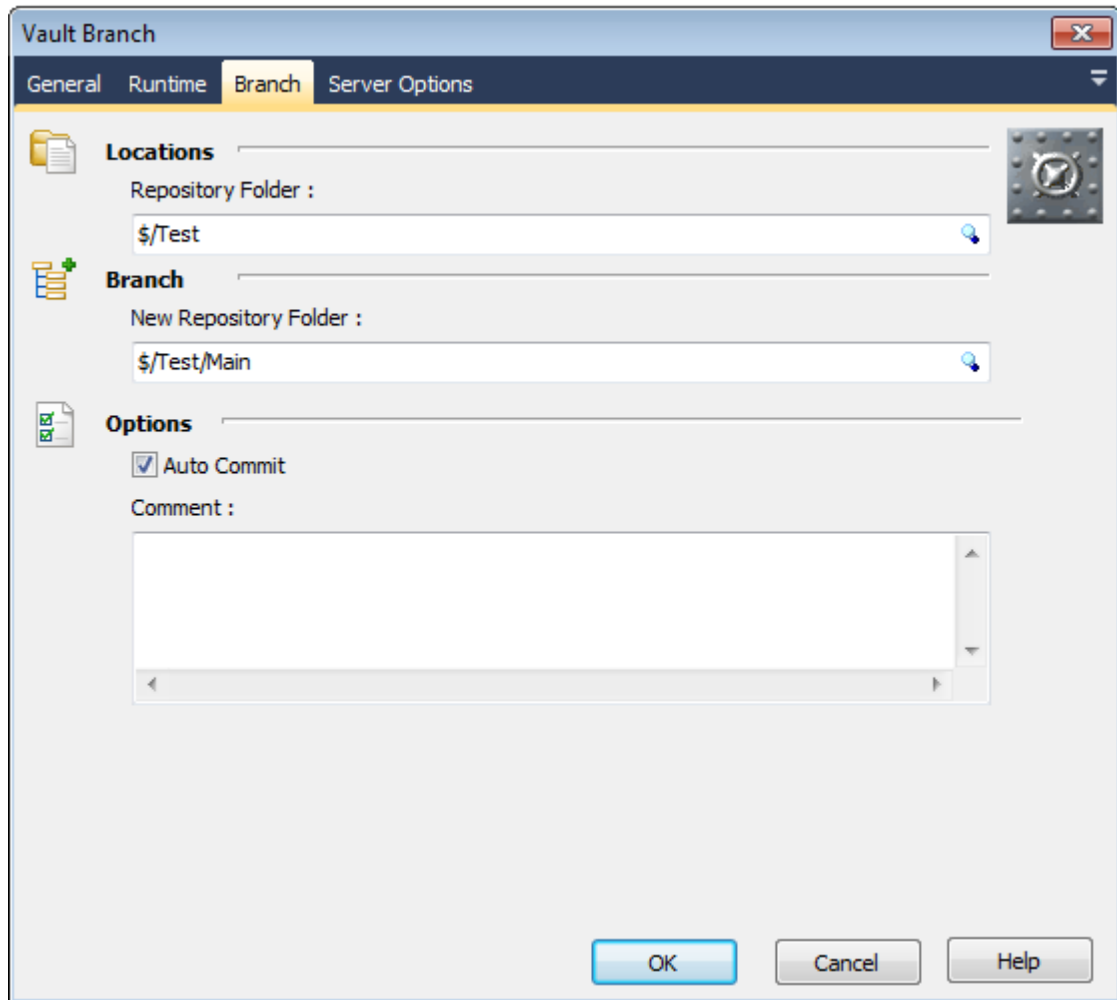
6.34.17.1 Vault Add Action

The Vault Add action will add the specified files and directories into the selected repository folder.



6.34.17.2 Vault Branch Action

BRANCH will create a branch for the repository folder specified by repository folder at New repository folder.





6.34.17.3 Vault Check In Action

CheckIn uses the COMMIT command, it will commit the items in the pending changeset list specified by files(s).




Vault Check In ✕


General Runtime **Check In** Server Options

 **Location**

Repository Folder : 

Repository Files : Leave file list blank to check in all files in the repository folder

 Add File  Edit  Remove

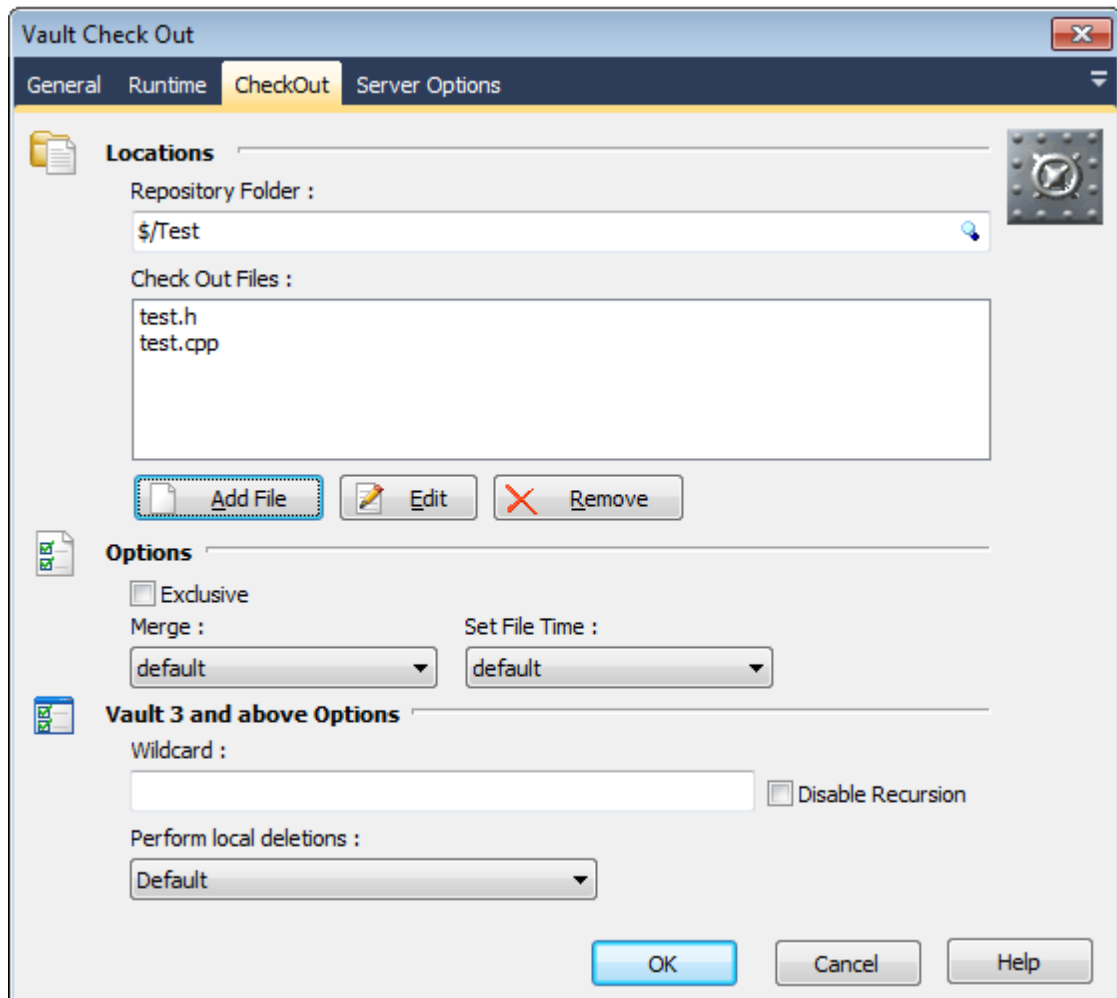
 **Options**

Comment :

Unchanged Files : ☐ Keep Checked Out

6.34.17.4 Vault Check Out Action

CHECKOUT will checkout files from the repository.

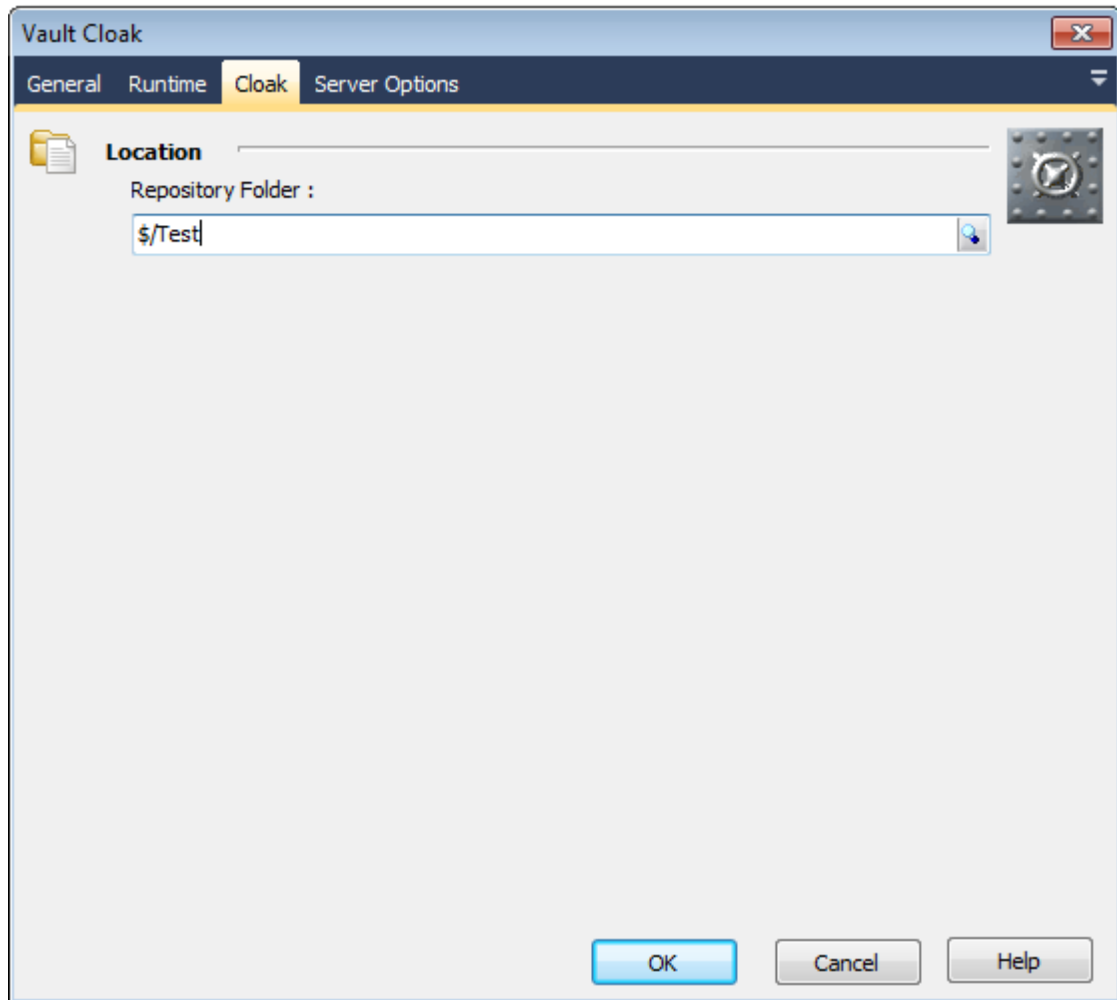


Server and authentication information is specified by:

- host host
Hostname of the vault server to connect to - also see SERVER
- ssl
Enables SSL for server connection
- user username
Username to use when connecting to server - also see USERNAME
- password password
Password to use when connecting to server
- repository repositoryname
Repository to connect to

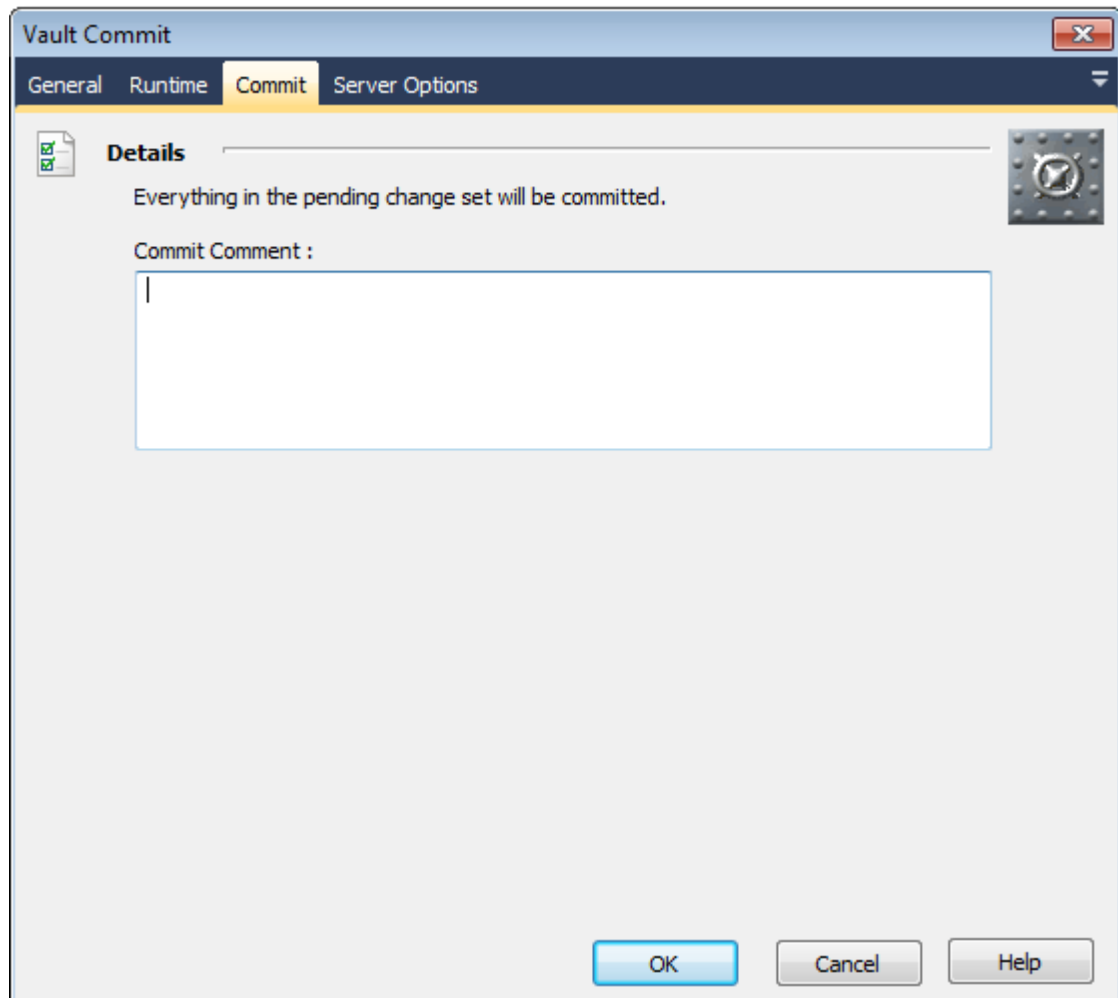
6.34.17.5 Vault Cloak Action

The Cloak action will cloak a repository folder.



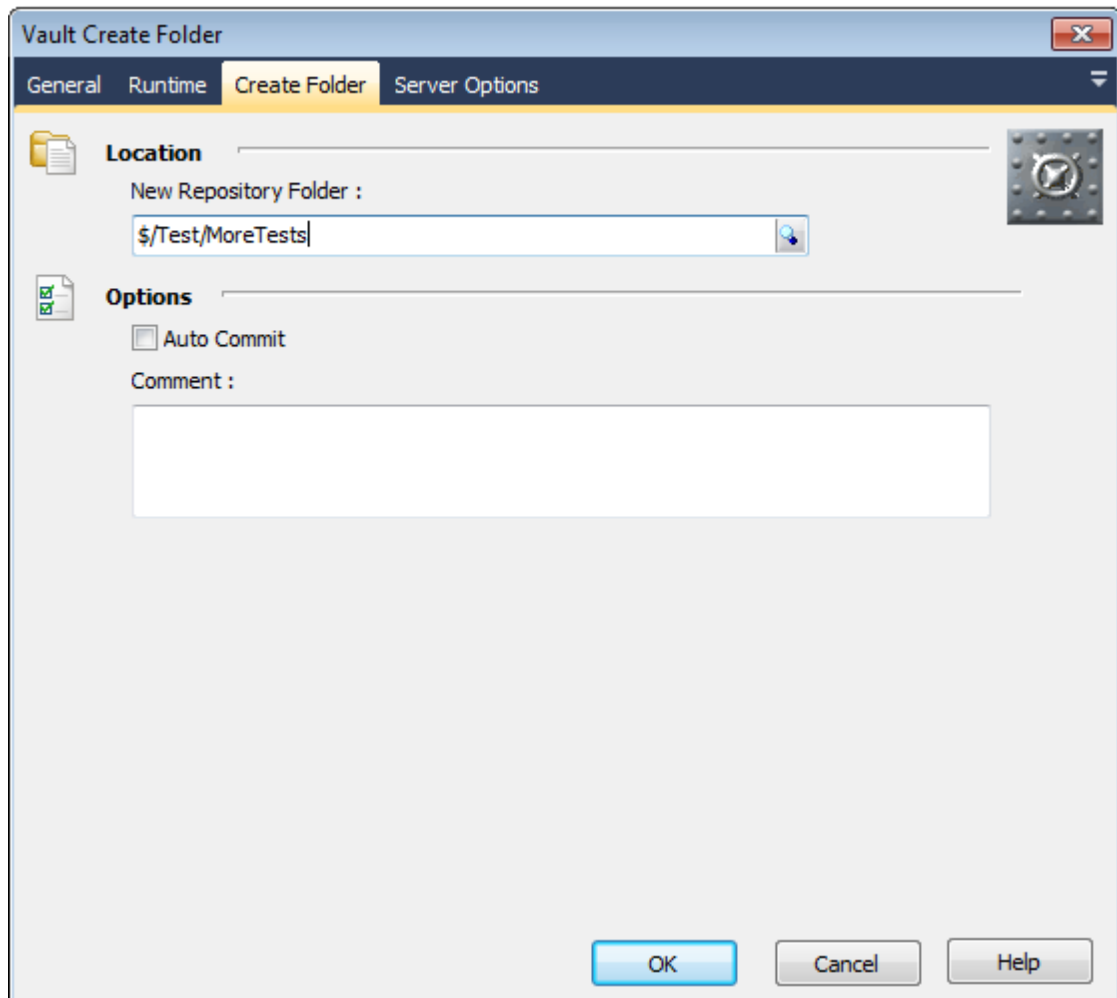
6.34.17.6 Vault Commit Action

Commits all the items in the pending changeset list.



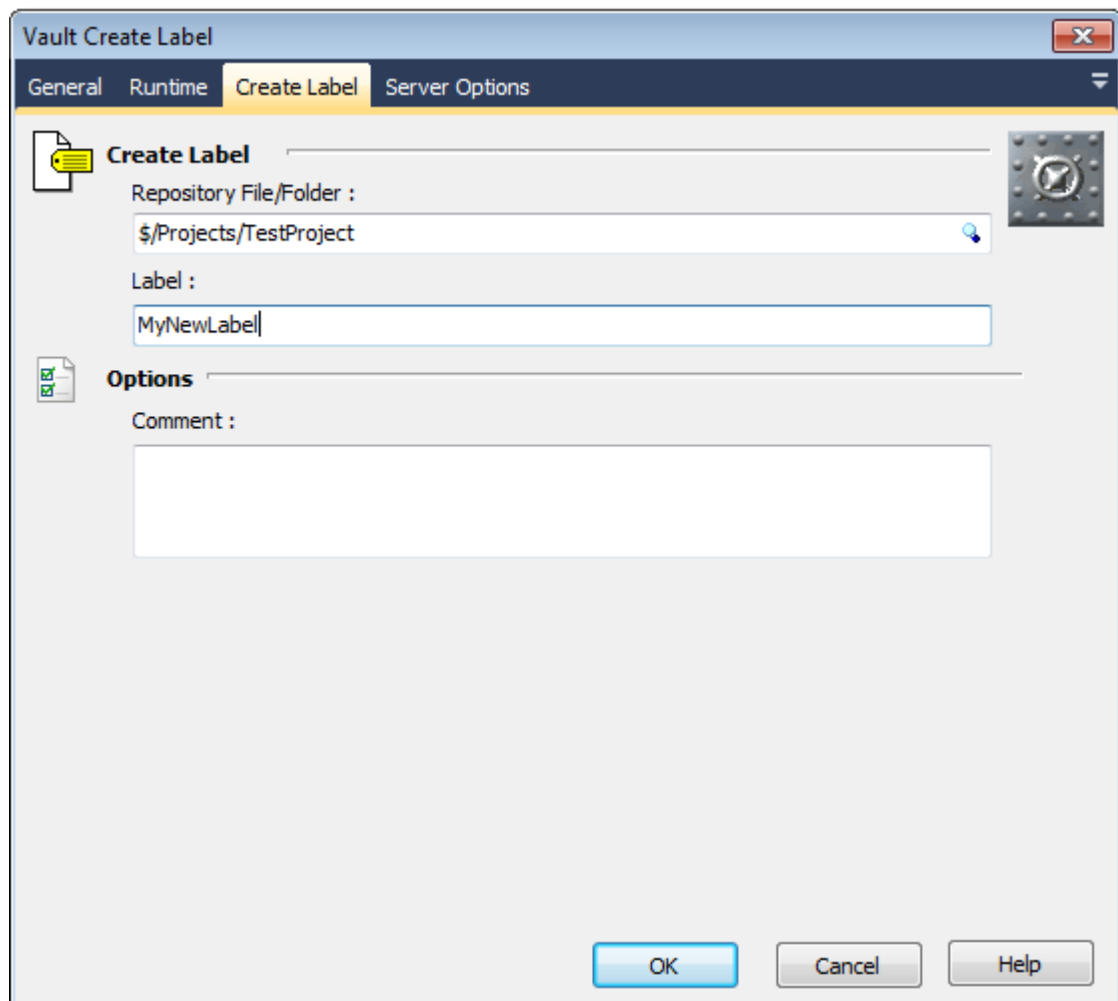
6.34.17.7 Vault Create Folder Action

The Vault Create Folder action will create a new vault folder at the specified location in the repository



6.34.17.8 Vault Create Label Action

The Vault Create Label action will create a label at the specified repository file/folder.



The screenshot shows the 'Vault Create Label' dialog box with the 'Create Label' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Create Label' (highlighted), and 'Server Options'. The 'Create Label' section contains a 'Repository File/Folder' field with the text '\$/Projects/TestProject' and a 'Label' field with the text 'MyNewLabel'. To the right of these fields is a small icon of a clock face. Below these fields is an 'Options' section with a 'Comment' field. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Vault Create Label

General Runtime **Create Label** Server Options

Create Label

Repository File/Folder :
\$/Projects/TestProject

Label :
MyNewLabel

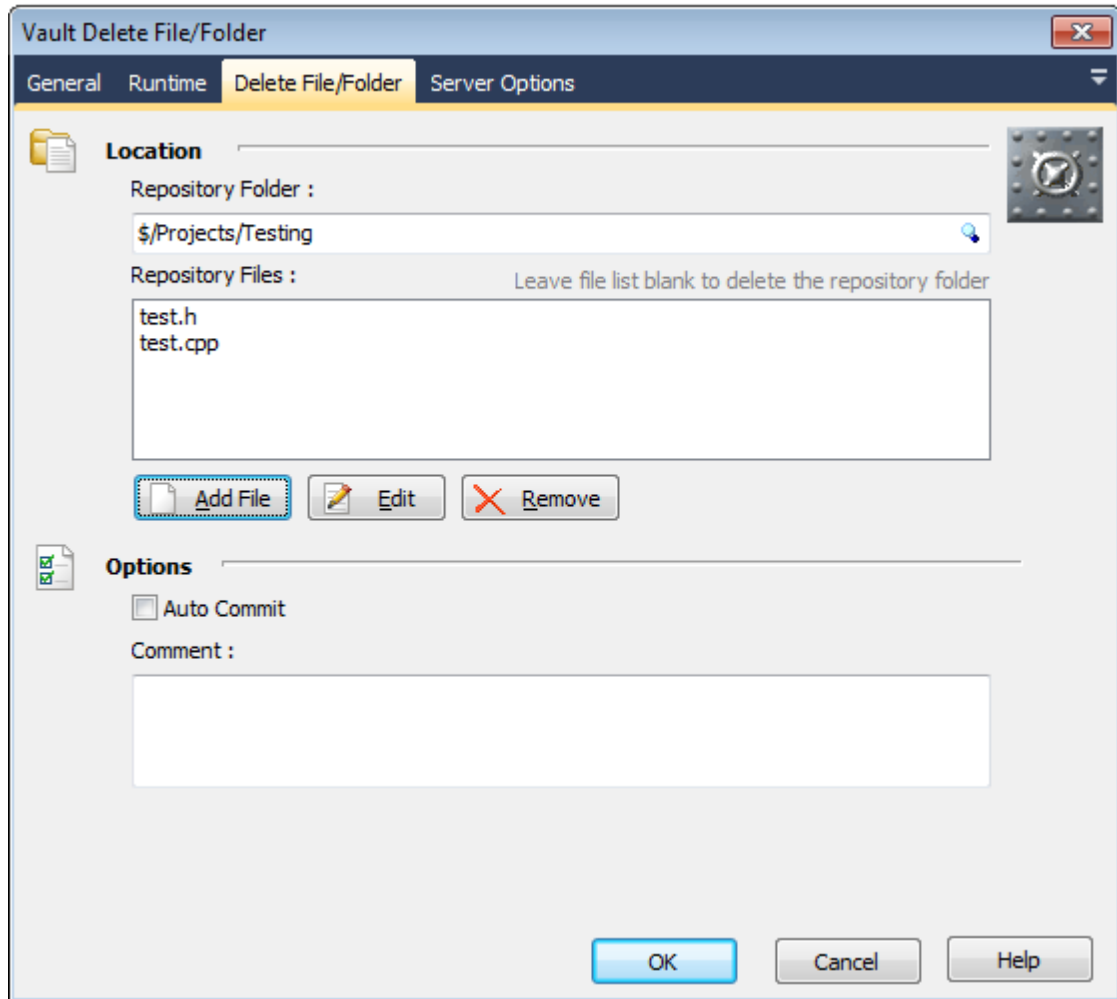
Options

Comment :

OK Cancel Help

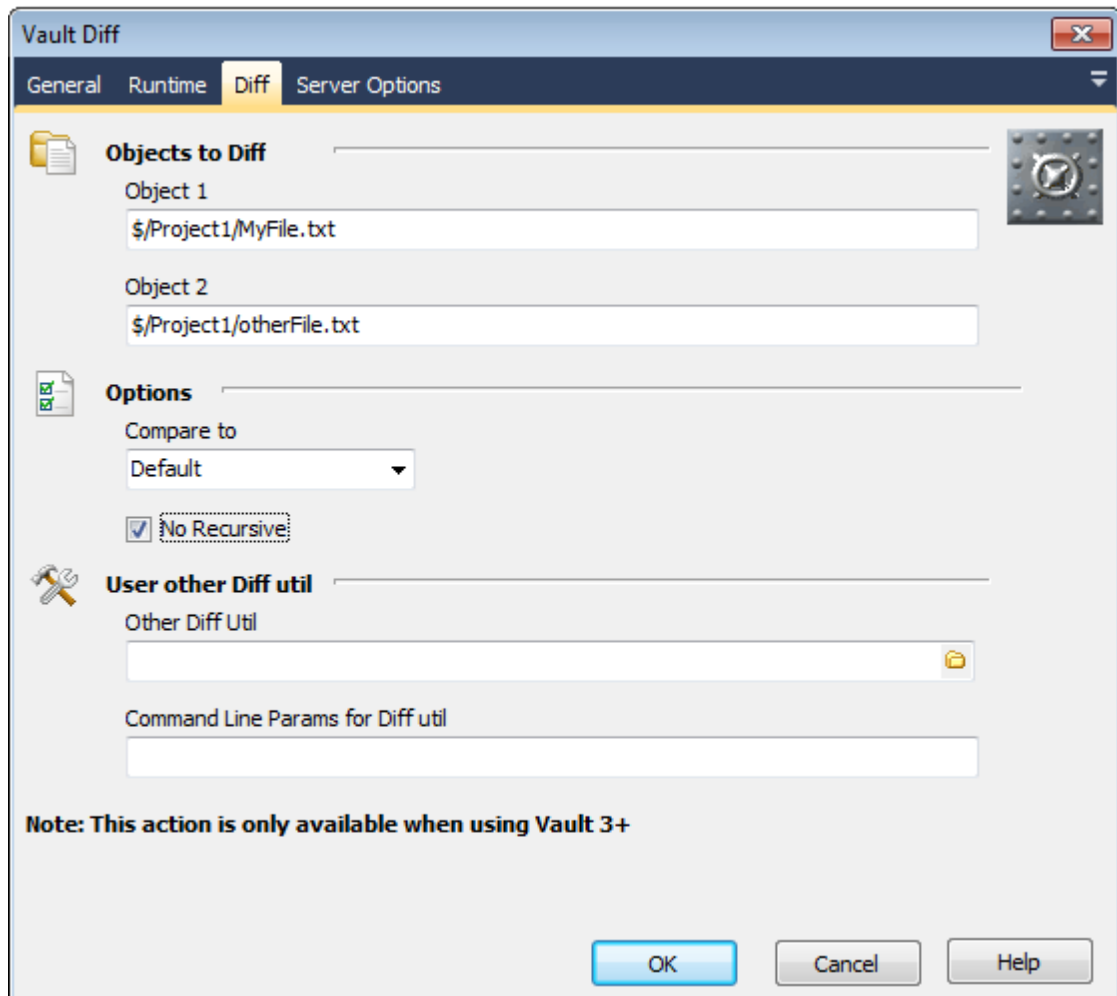
6.34.17.9 Vault Delete File/Folder Action

The Vault delete file/folder action will delete the specified files in the repository folder, or it will delete a repository folder if no files have been specified.



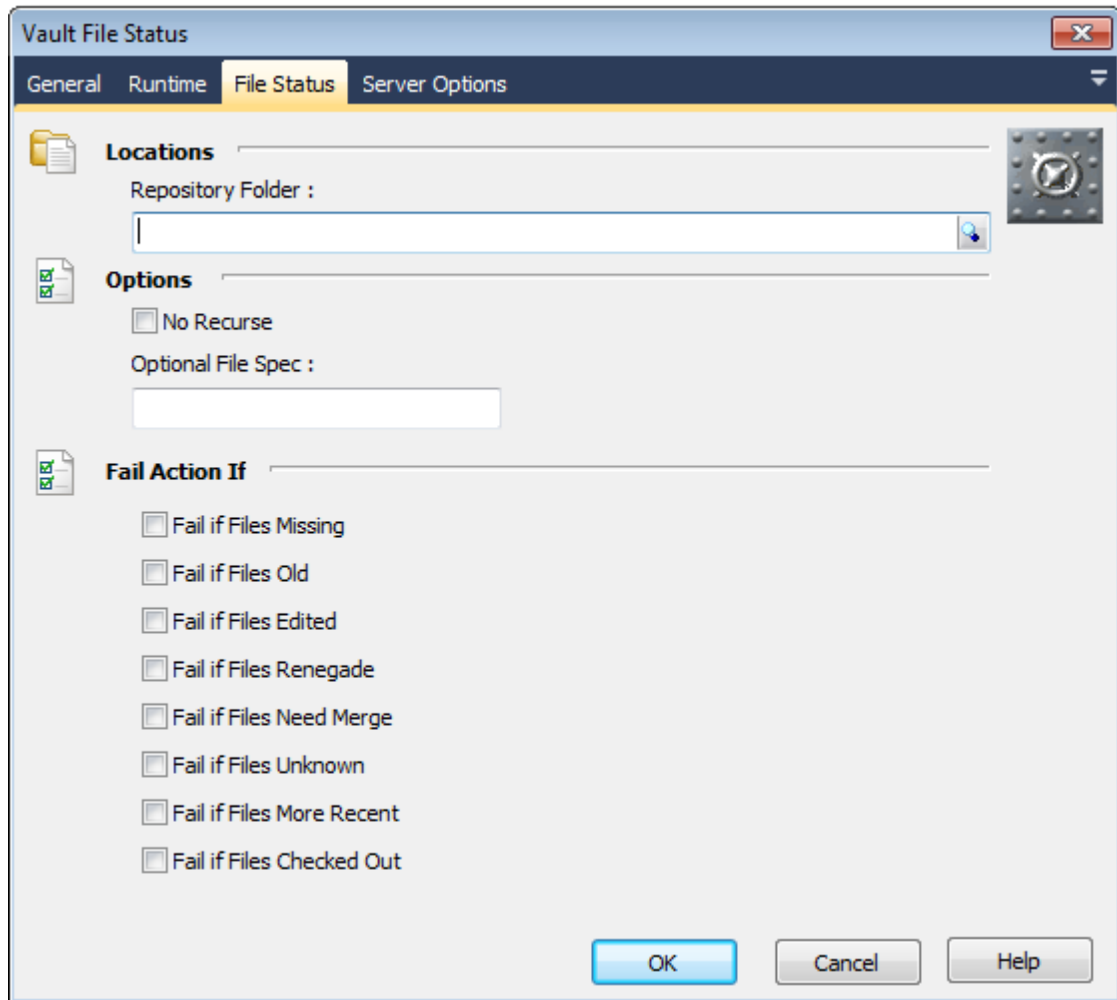
6.34.17.1 Vault Diff

The vault Diff action allows you to compare objects using various methods within and external to the vault repository.



6.34.17.1 Vault File Status

The Vault File Status action allows you to check a file, group of files, or an entire repository to see if all files are valid.

**Locations**

Specify the path to a Vault repository folder.

Options

No Recurse - do not recurse subdirectories of this folder.

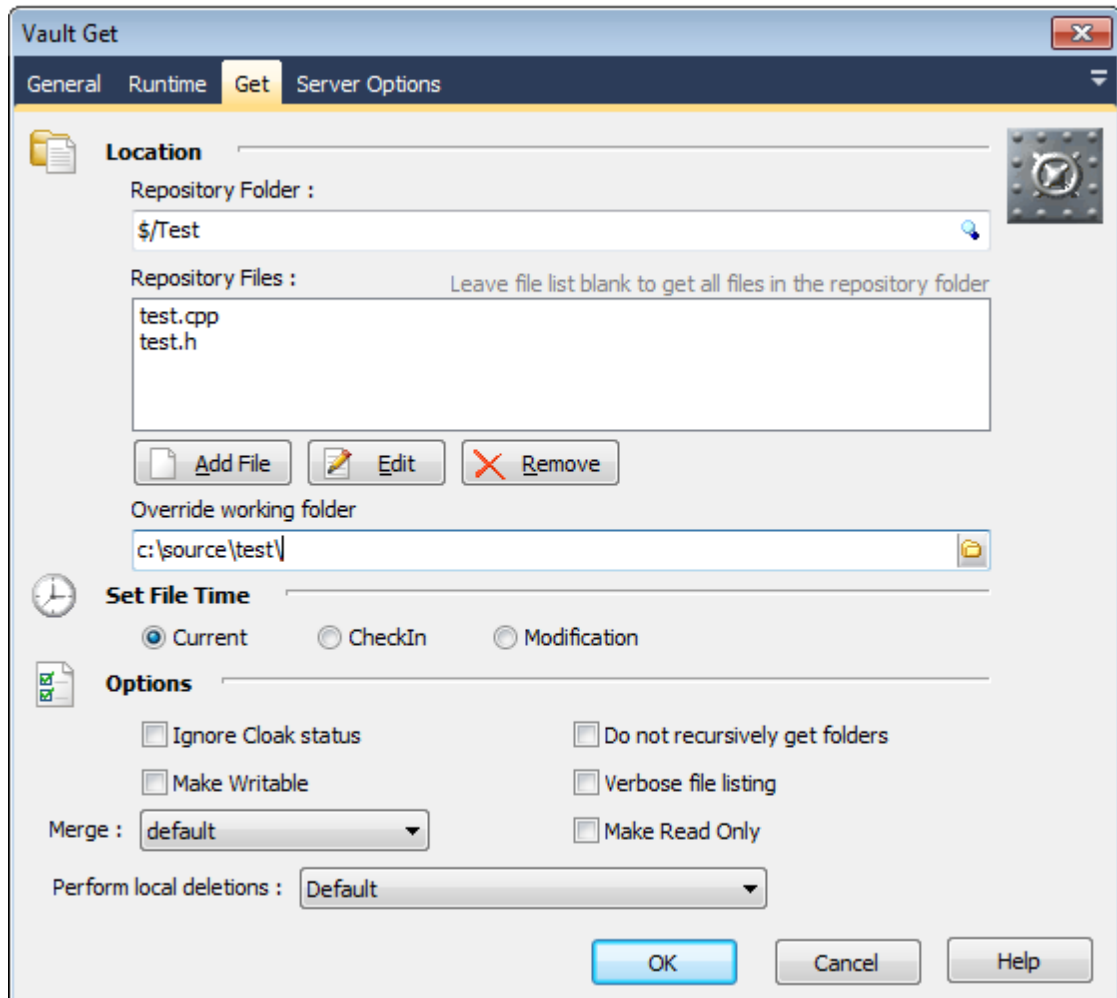
Optional File Spec - If blank, all files will be evaluated. Otherwise, specify a file or use wildcards to specify a group of files to check.

Fail Action If

Set which file conditions will cause the action to fail.

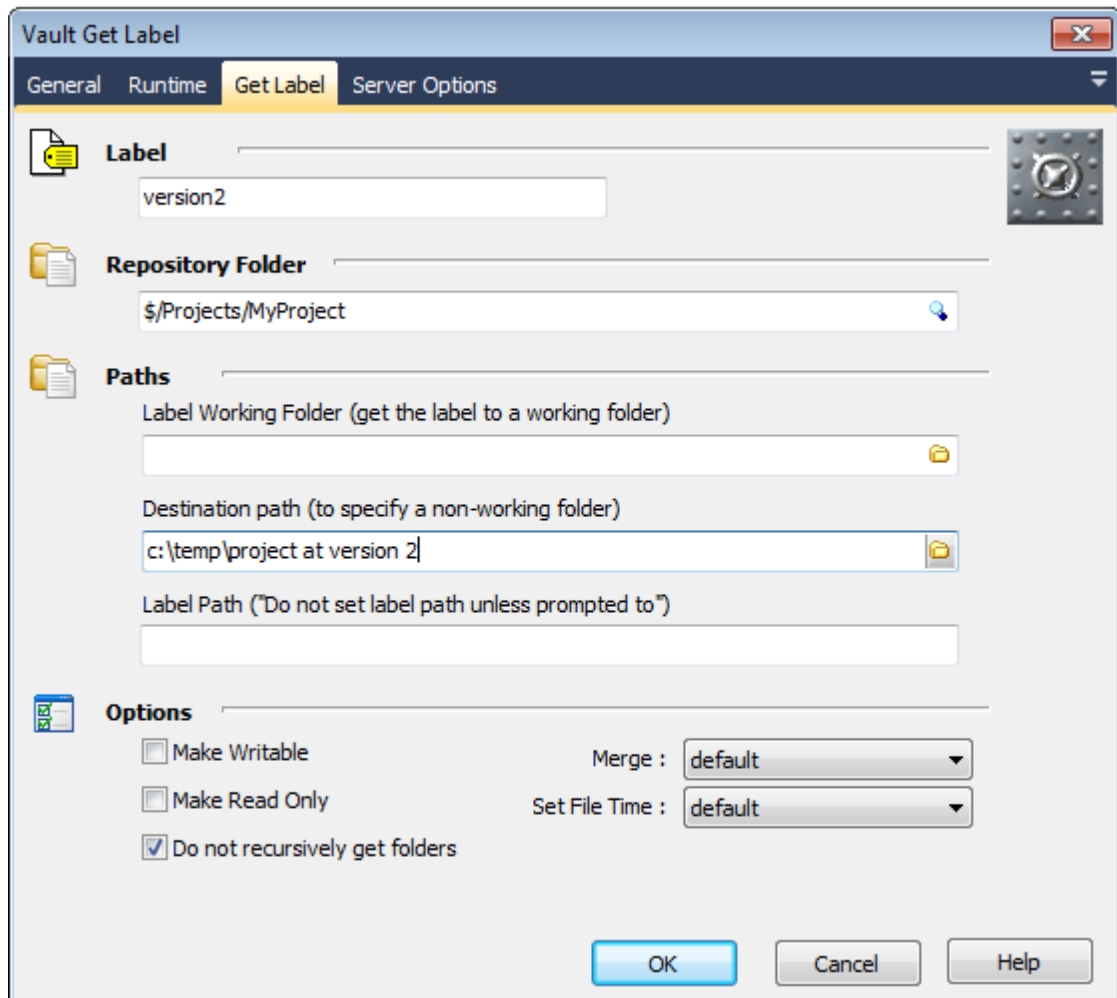
6.34.17.1 Vault Get Action

GET will retrieve the latest version of files or folders in the repository as specified by repository path(s). The files will be stored in their corresponding working folders on the local system.



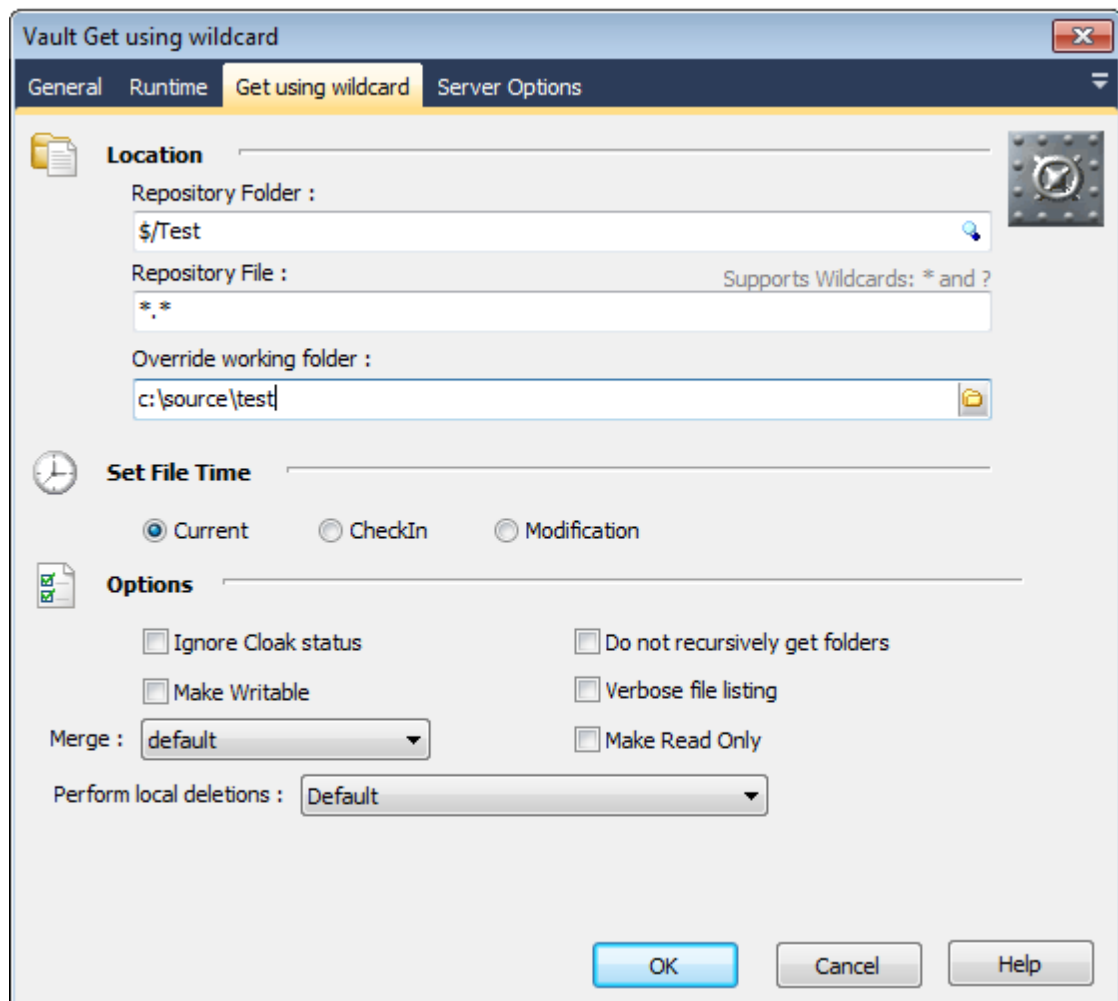
6.34.17.1 Vault Get Label

The Vault Get Label action allows you to get files from the repository at a particular label.



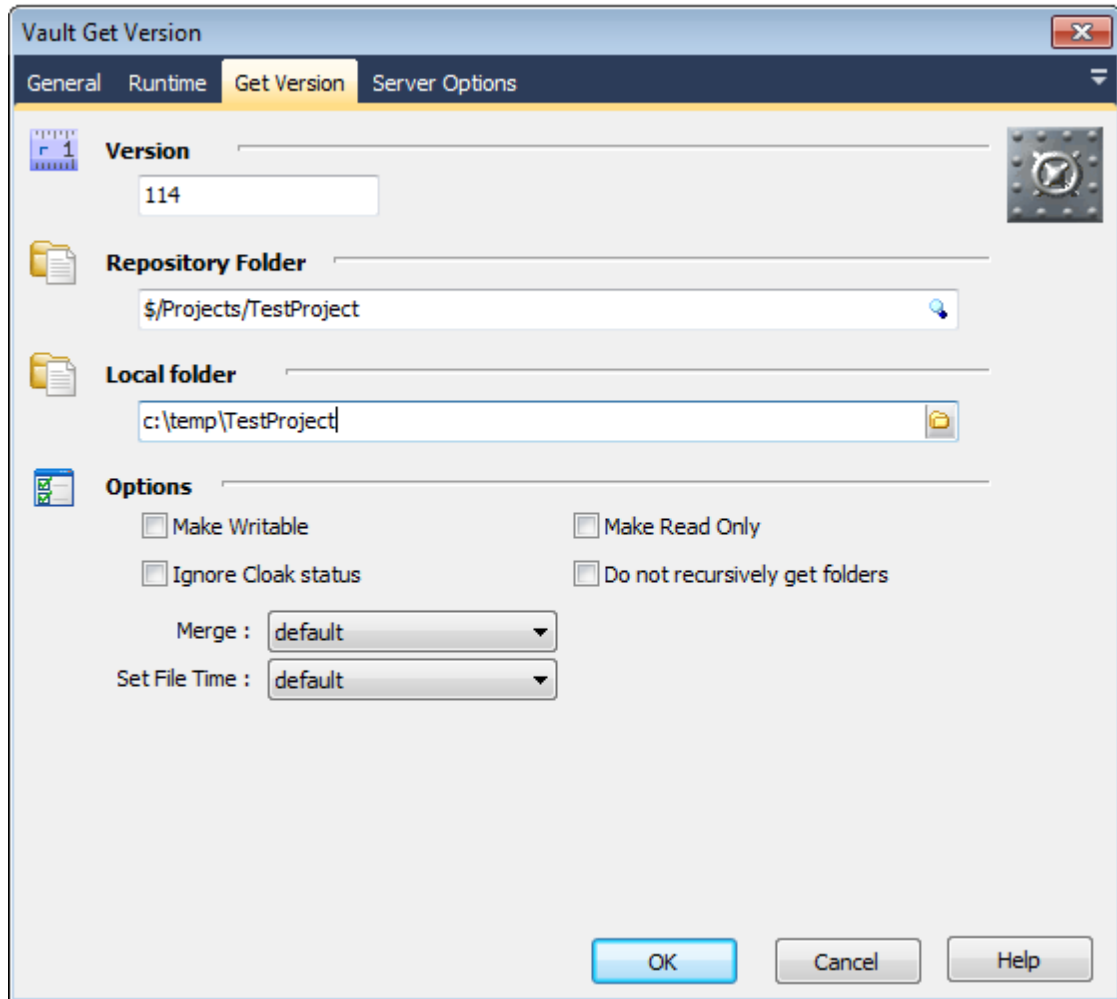
6.34.17.1 Vault Get using Wildcards Action

GETWILDCARD will retrieve all files within the folder specified by repositoryfolder whose name matches one of the wildcards specified. You may use '?' to match a single character or '*' to match a range of characters.



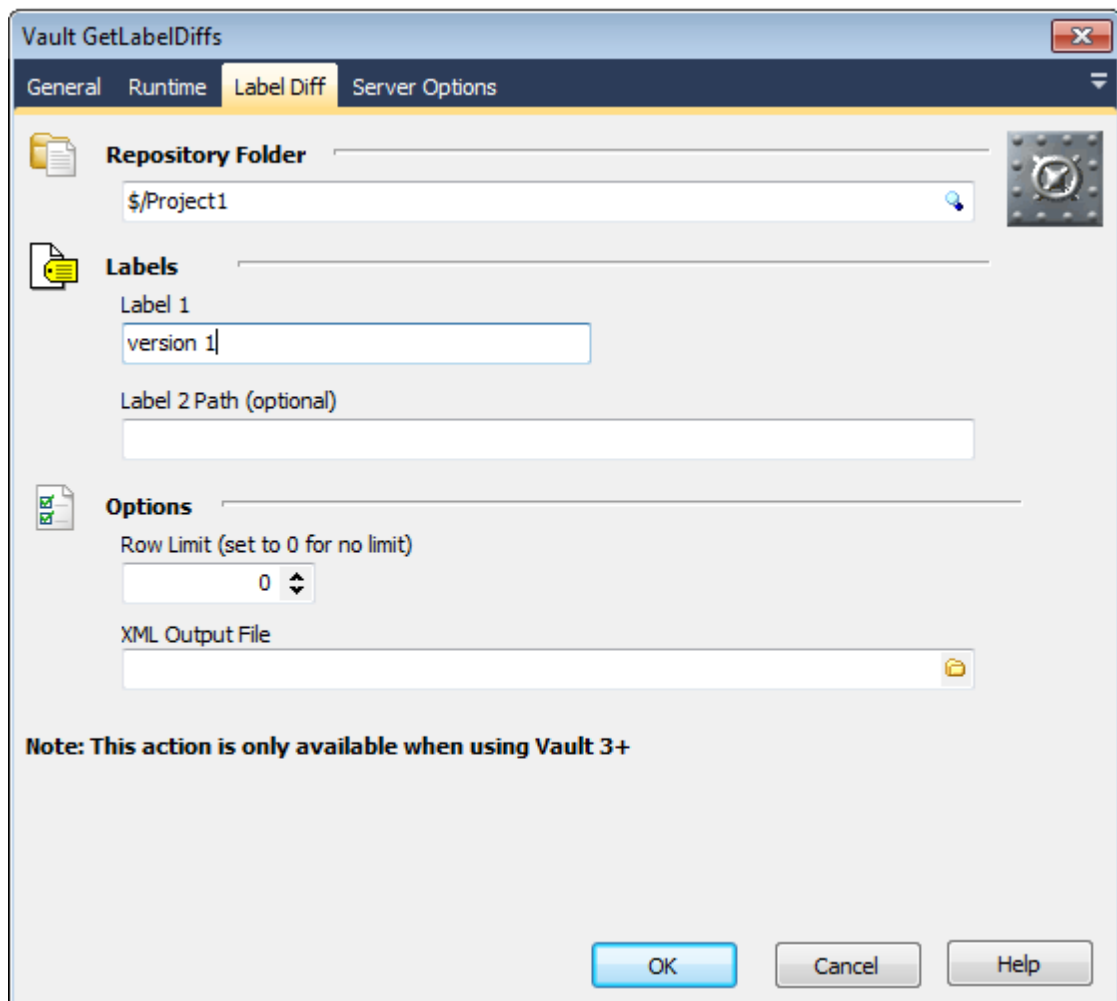
6.34.17.1!Vault Get Version Action

The Vault Get Version action will get files from the vault repository to the specified local folder at a specific version number.



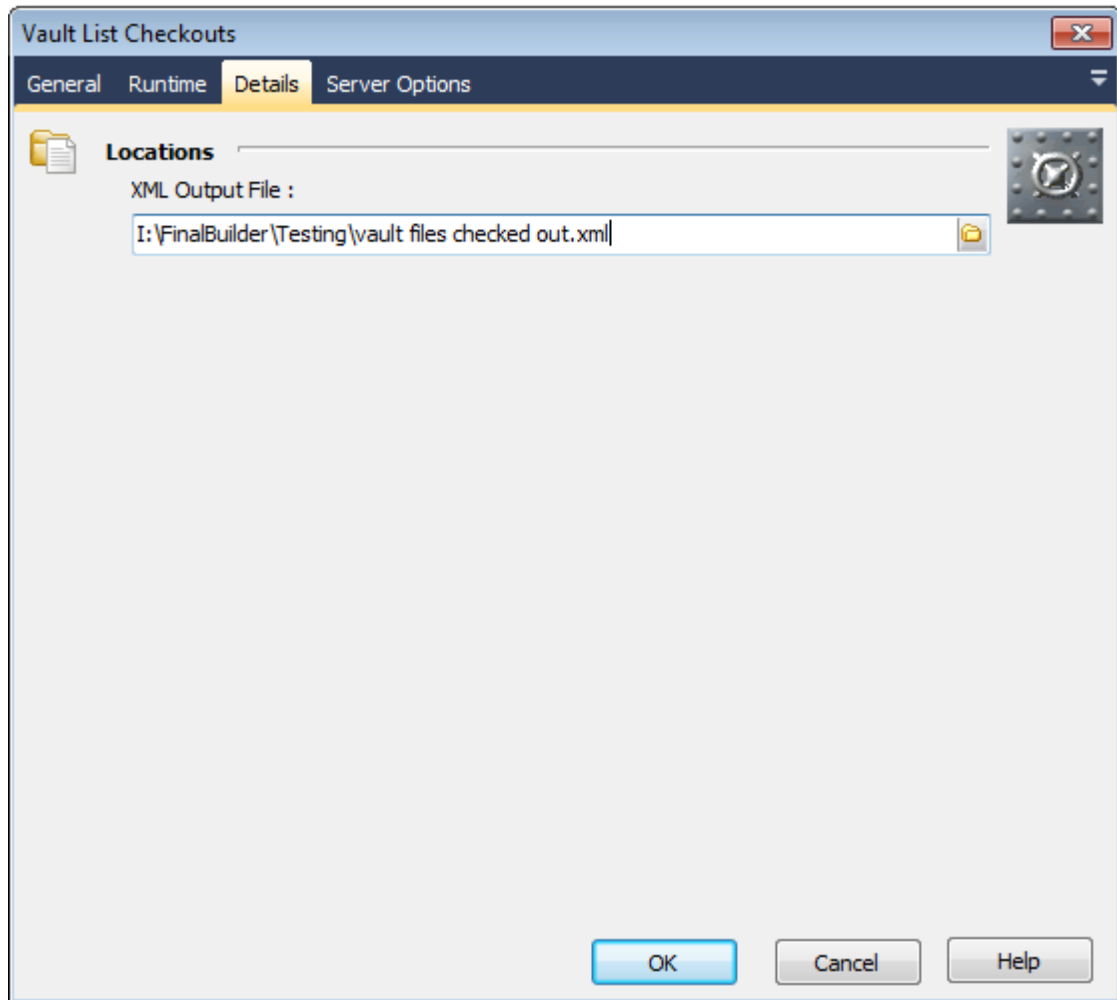
6.34.17.1 Vault GetLabelDiffs

Executes the GetLabelDiffs in Vault 3.0

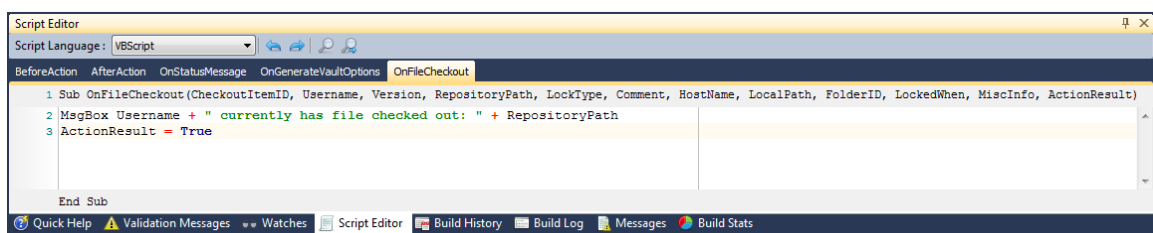


6.34.17.1 Vault ListCheckouts

The Vault ListCheckouts action enables you to query a vault repository for a report on all files currently checked out. The action can output the report as an xml file, or you can use the OnFileCheckout script event to process each file checked out by each user.



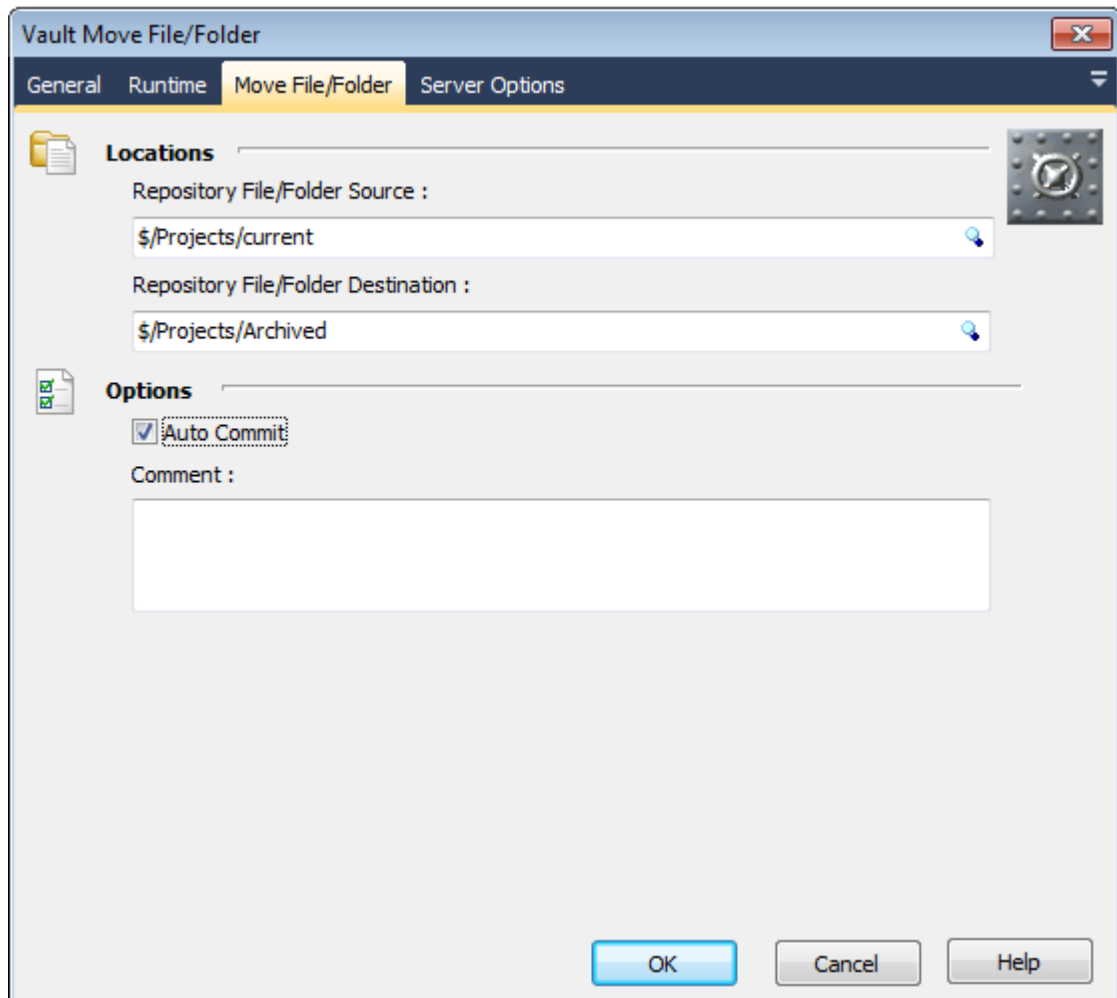
And to process each file checked out by each user, you can add code in the OnFileCheckout script event:



Setting the ActionResult to False will cause the action to fail after it has processed all of the files.

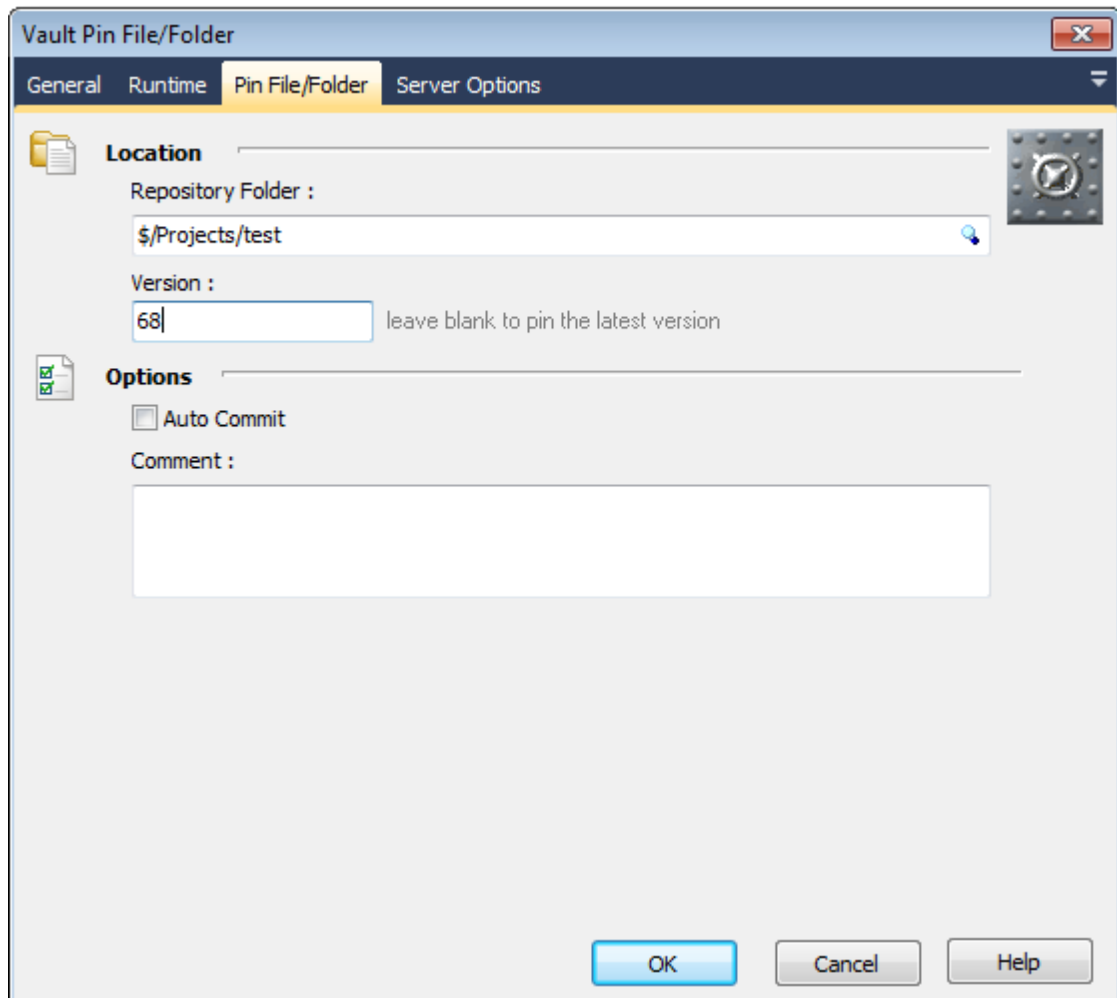
6.34.17.1 Vault Move File/Folder Action

The Vault Move File/Folder action allows you move a file or folder to a new repository path.



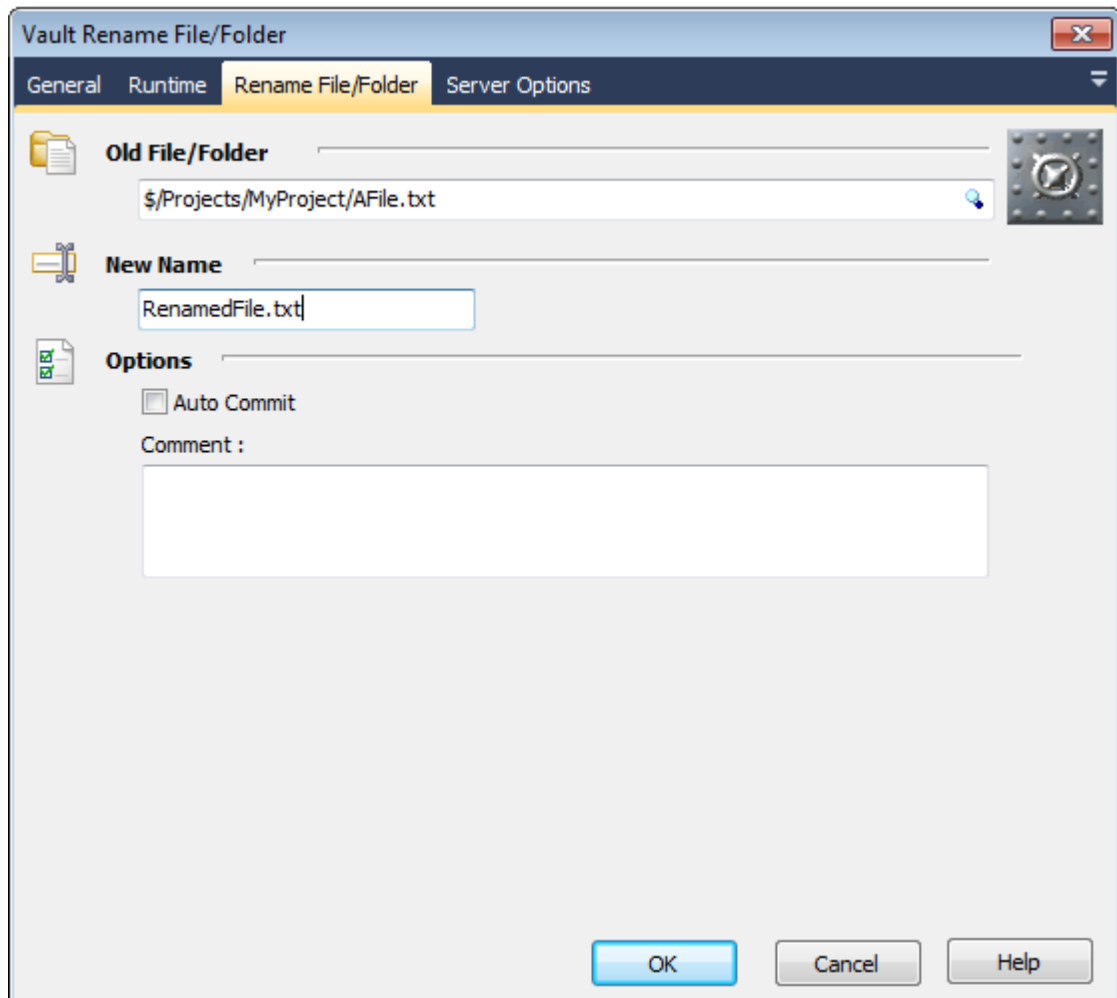
6.34.17.1 Vault Pin File/Folder Action

The vault Pin action allows you to Pin the specified file or folder in the repository to a particular version



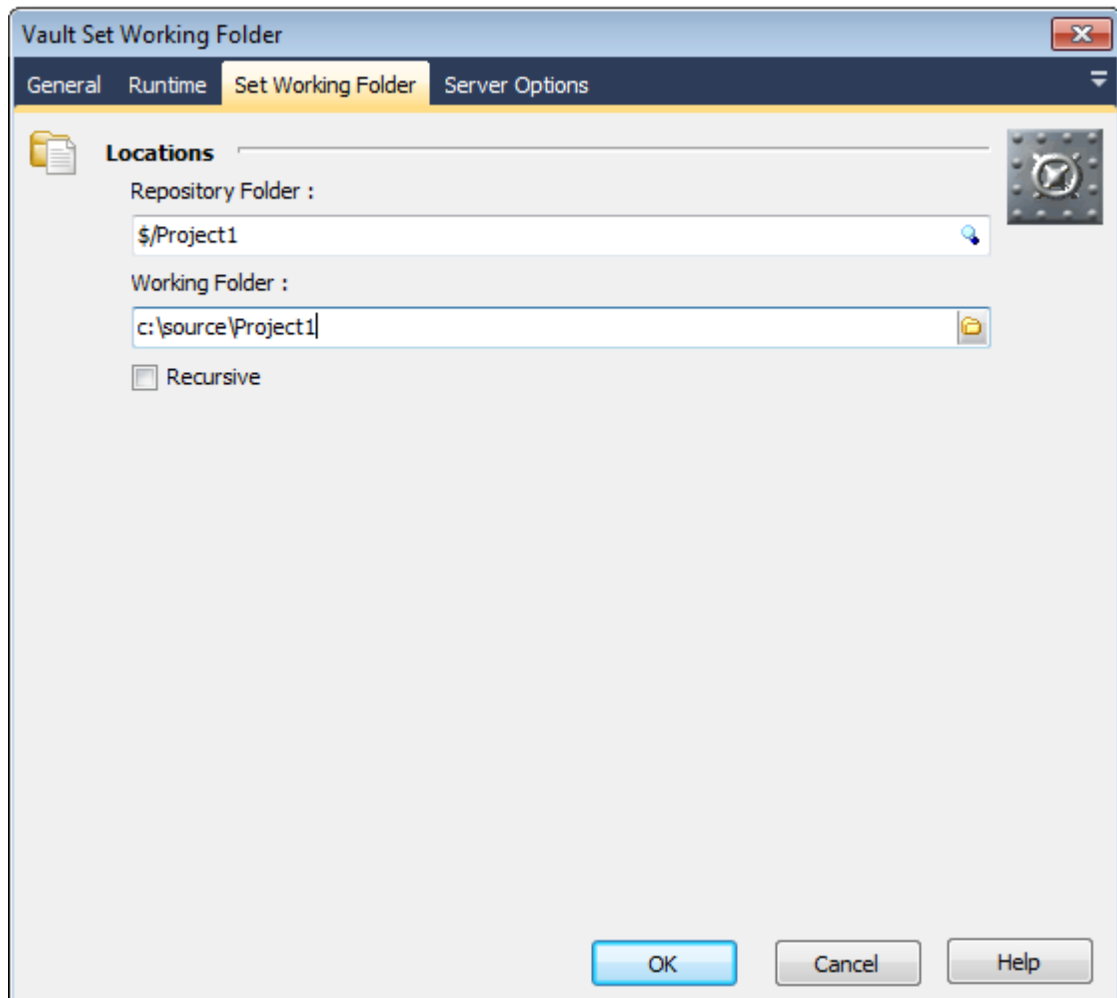
6.34.17.2 Vault Rename File/Folder Action

The vault rename file/folder action allows you to rename a file or folder in the vault repository.



6.34.17.2 Vault Set Working Folder

The vault set working folder action will set the working folder of a specified repository folder to a different location.



6.34.17.2 Vault Share File/Folder Action

Share a file/folder to a new location in the vault repository.

The screenshot shows the 'Vault Share File/Folder' dialog box with the 'Share File/Folder' tab selected. The dialog has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Share File/Folder' (active), and 'Server Options'. The main area contains three sections: 'Original File/Folder' with a text field containing '\$/Projects/version1' and a folder icon; 'Share to' with a text field containing '\$/Projects/version2' and a folder icon; and 'Options' with an 'Auto Commit' checkbox and a 'Comment' text area. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Vault Share File/Folder

General Runtime **Share File/Folder** Server Options

Original File/Folder

\$/Projects/version1

Share to

\$/Projects/version2

Options

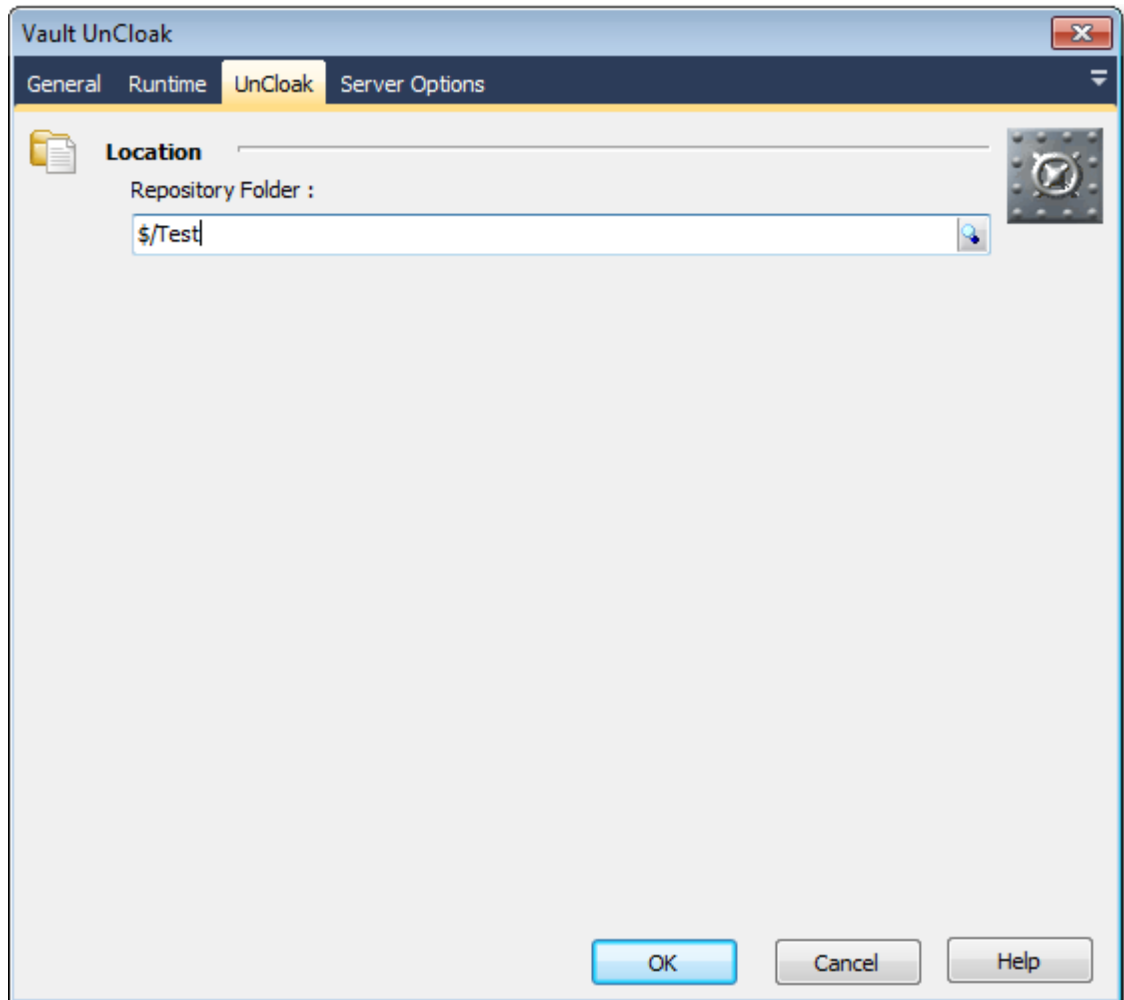
☐ Auto Commit

Comment :

OK Cancel Help

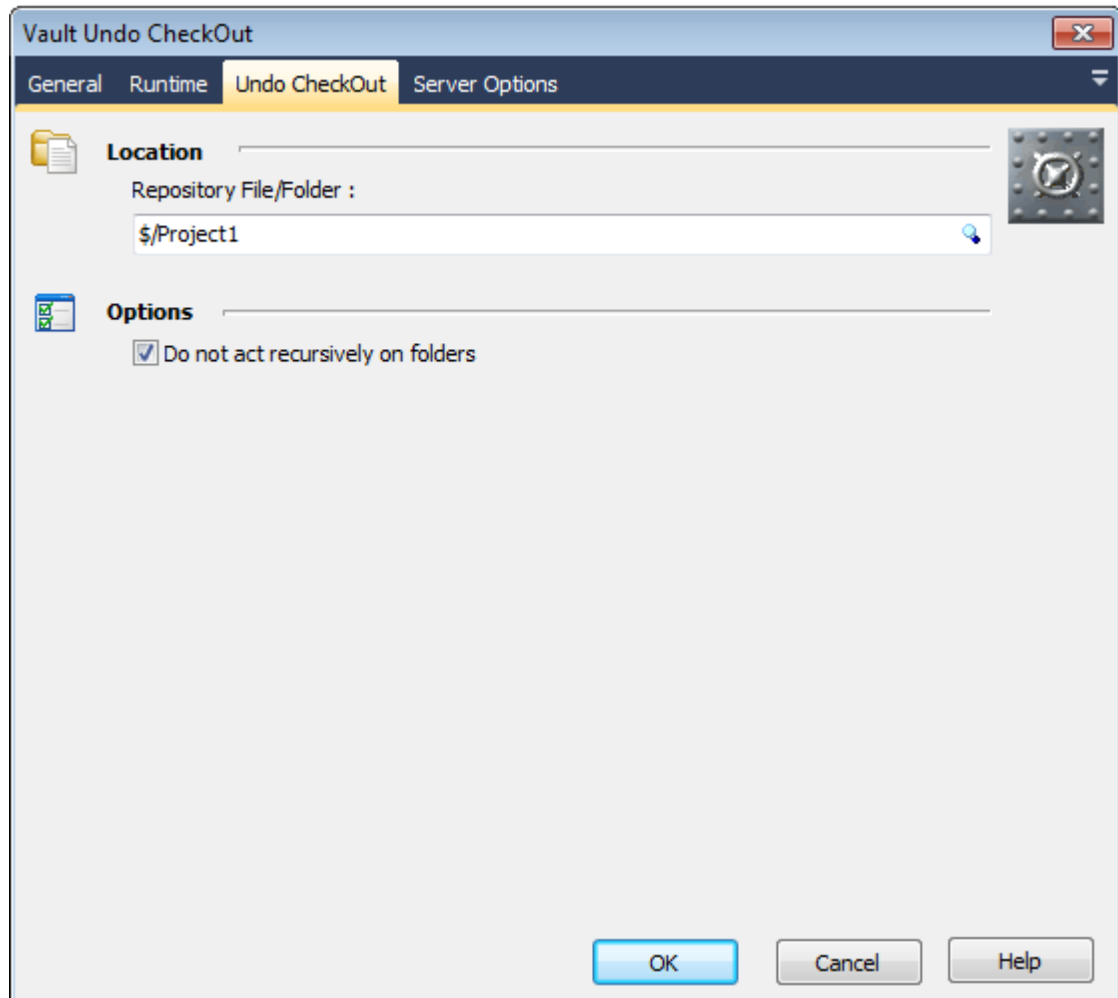
6.34.17.2 Vault UnCloak Action

The UnCloak action will uncloak a repository folder.



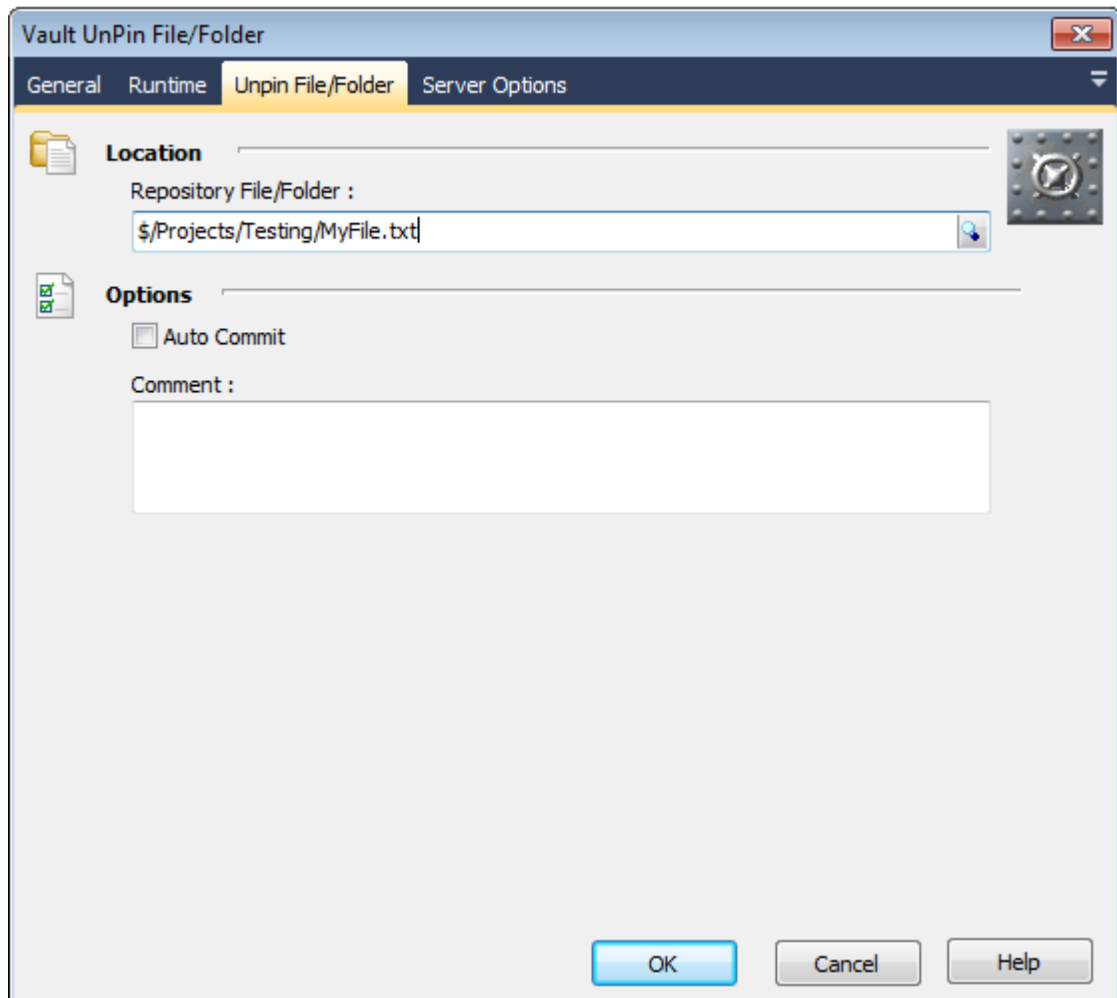
6.34.17.2 Vault Undo Checkout Action

The vault undo checkout action enables you to undo the checkout of any files in the specified repository folder, or will undo the checkout of a single file if a file in the repository is specified.



6.34.17.2 Vault UnPin File/Folder Action

The Vault UnPin action allows you remove any Pins on the specified file or folder in the repository



6.34.18 Subversion Actions

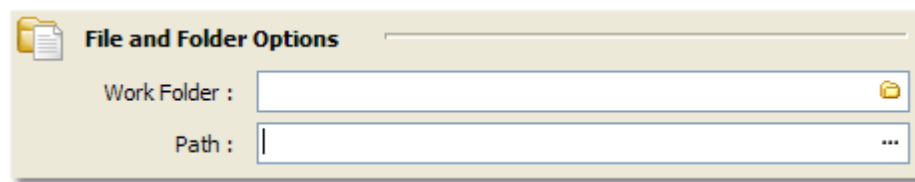
The Subversion actions support the Subversion version control system. They provide an easy way to use Subversion to check in or out files, or to modify a Subversion repository. Before using these actions you must set up the path to the Subversion command line tool in the FinalBuilder Options Dialog.

Important note: The command-line version of Subversion (svn.exe) must be installed, *not* just the TortoiseSVN GUI.

For more information about Subversion, see <http://subversion.tigris.org/>
For technical information about the Subversion commands, consult the Subversion documentation here: <http://svnbook.red-bean.com/nightly/en/svn-book.html>

Action-specific options

Work Folder and Path:



Most Subversion actions have fields for a "Work Folder" and "Path". These concepts are important to understand:

- **Work Folder:** The folder used as the current directory when the Subversion executable (svn.exe) is run. For example, if you specify "C:\Project\Docs" as the work folder, and "help.txt" as another argument, then Subversion will look for a file called "C:\Project\Docs\help.txt". Note that the Work Folder often corresponds to the Subversion "Working Copy", but does not have to.
- **Path:** A list of items for the Subversion action to operate on, separated by spaces. These items are generally either locations of files or folders on disk, or URLs that point to files or folders on a Subversion server. Some examples of valid path arguments (without quotes):
 - "c:\project\docs c:\project\src"
 - "file:///c:/project/readme.txt"
 - "gui.c gui.h *.txt"

General Options:

On each action, there is a General Options tab. The options for each action vary, but are from the following list:

Config Directory: Specifies a directory where user configuration files are found. You can specify a default in the Global Options.

Commit Message file: Specifies the name of a file which contains the commit message to use, rather than specifying the message directly.

Encoding: Specifies the name of the encoding.

Path to diff3.exe: Specifies the path to a diffing program to using, rather than the default. A default can be set in Global Options.

External Editor: Specifies the path to an external editor, such as Notepad, which will

be called by Subversion if a commit message is not provided.

The following flags are self-explanatory. See the Subversion documentation for more details.

Don't cache authentication information locally (--no-auth-cache)

Non Interactive (--non-interactive)

Output minimum text (--quiet)

Don't recurse into subdirectories (--non-recursive)

Set properties automatically (--auto-props)

Don't set properties automatically (--no-auto-props)

Override any warnings (--force)

Recursive (--recursive)

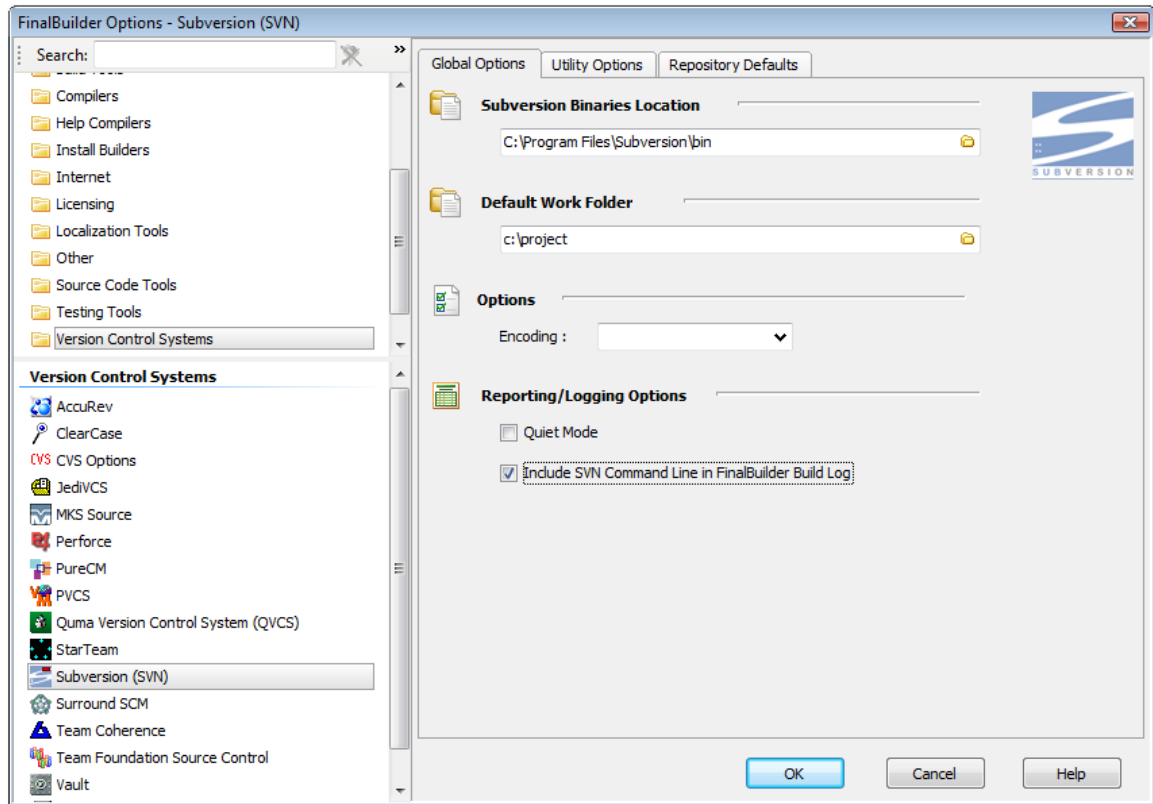
Revisions:

On certain actions, a revision specifier can be used. Revisions can take any of the following formats. If you click the ellipsis button (...) in the Revision editor field, then there is a custom dialog to choose a revision type.

Revision Type	Format	Example
Latest	HEAD	HEAD
Revision number	<Number>	7
Revision at date	{<Date>}	{2008-03-12}
Base revision of local working copy	BASE	BASE
Last commit (at or before base revision)	COMMITTED	COMMITTED
Previous revision, before last commit	PREV	PREV
A range between two revisions (inclusive)	<Revision 1>:<Revision 2>	7:12 {2008-01-01}:HEAD

Global Options:

Before you start using the Subversion actions in FinalBuilder, you should set up the following global options, utility options and repository defaults in the **Tools | Options** dialog:



"Subversion binaries location"

The path to the 'bin' directory of the Subversion installation, where svn.exe is found. This is usually C:\Program Files\Subversion\bin.

"Default Work Folder"

Optional. The directory suggested for the Work Folder field each time you create a new Subversion actions. If you mostly use one working copy directory, you should specify that here. You could also specify a variable reference, such as "%WorkFolder %".

"Encoding"

Optional. Used to specify a non-default encoding for commit messages, etc. If left blank, uses an encoding based on the operating system locale.

"Include SVN Command Line in FinalBuilder Log"

If this option is set, FinalBuilder will log the full command line as sent to svn.exe each time a Subversion action runs, allowing you to see diagnose any problems more easily.

Utility Options:

The screenshot shows a configuration window with a light beige background. It contains three main sections, each with a folder icon and a title: 'Configuration Directory', 'Global Arguments File', and 'Alternate Commands'. Each section has a text input field with a folder icon on the right. The 'Alternate Commands' section has two additional text input fields: 'Optional Diff3 Tool for Merging' and 'Editor Location if Using Interactive Mode', each also with a folder icon on the right.

"Configuration Directory"

Optional. Allows you to specify a non-default directory for Subversion configuration information. You can specify individual configuration directories for most actions, as well. (Corresponds to the "--config-dir" command line switch.)

"Global Arguments File"

Optional. If you specify a file, then its contents will be used as extra arguments to subversion. Alternatively, you can specify specific arguments files for each subversion action. (Corresponds to the "--file" switch.)

"Optional Diff3 Tool for Merging"

Optional. The location of a custom diff3 merging tool, if you prefer not to use the default tool. (Corresponds to the --diff3-cmd command line switch.)

"Editor Location if Using Interactive Mode"

Optional. The location of an external text editor application. Required if using interactive mode: Subversion will call this application to get the user to enter a commit message. (Corresponds to the --editor-cmd switch).

Repository Defaults:

The repository defaults are used as default settings when creating new Subversion actions. They specify which Subversion repository to use, and how to connect to it. Changing these options has no effect on existing Subversion actions.

"Method"

How to connect to the repository. For a local repository, use the "file" method. For network access to a Subversion server, use "http", "https" or "svn".

"Host Name"

Specifies the name of the server hosting the repository. Used when making network connections.

"Path"

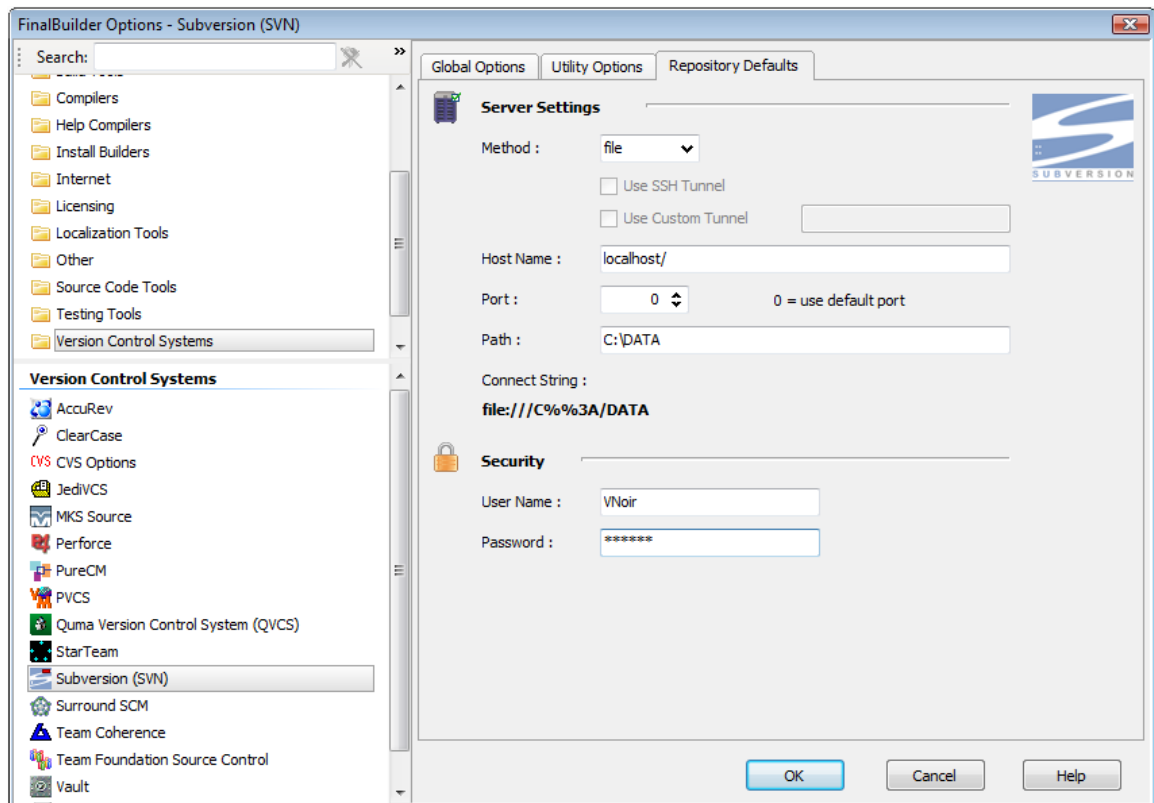
For a file-based repository, specify the local path (c:\svn\rep1). For a server, specify the path on the server (e.g., if your repository is found at svn://192.168.1.19/SVN/repository, use "SVN/repository").

"Connect String"

As you fill in the options, FinalBuilder gives you a preview of the complete identifier as it will be passed to Subversion.

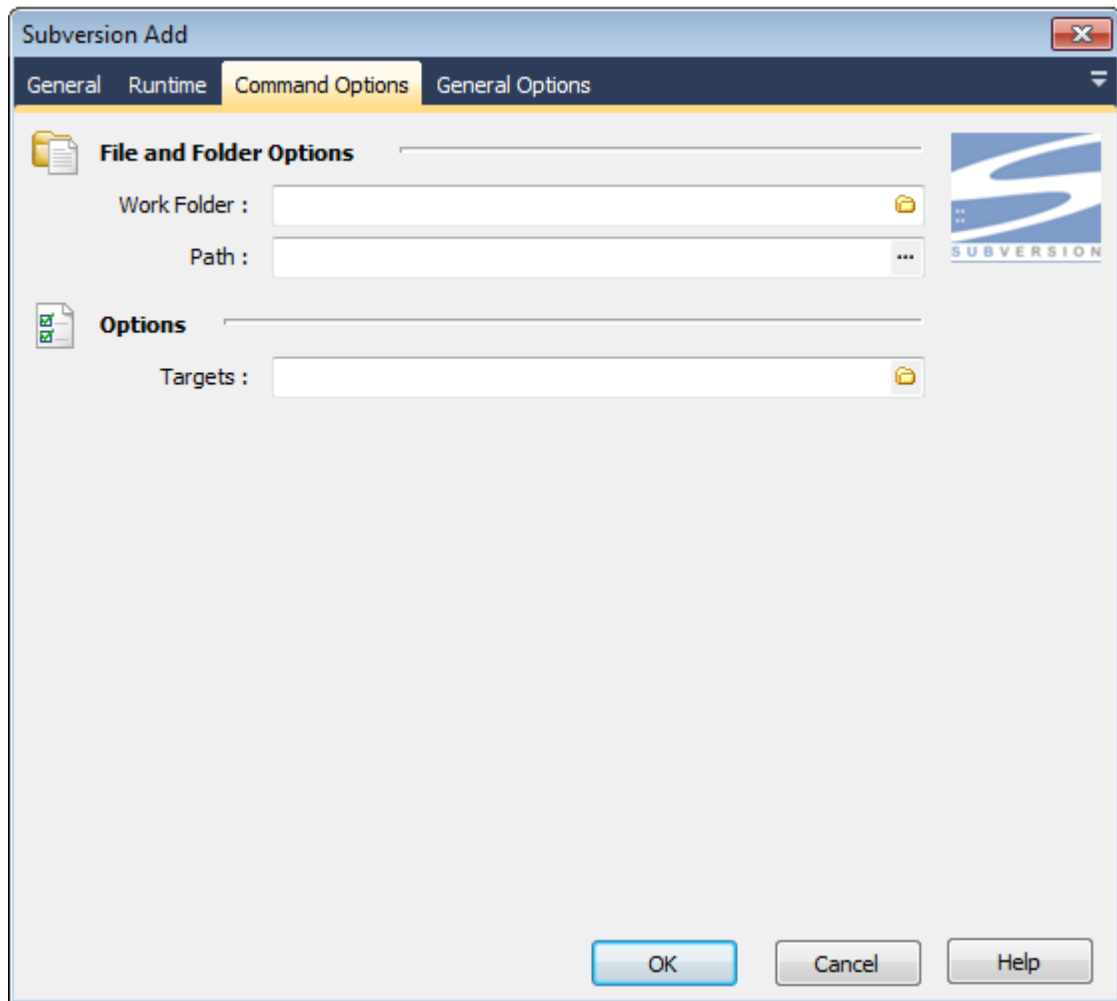
"Security"

Specify the user name/password with which to connect to the repository.



6.34.18.1 Subversion Add

The Subversion Add action places files and directories under version control. They will be added in the next commit.



Work Folder: The directory where the working copy will be created, and thus the files will be checked out to. By default, the path defined in the Options screen is used.

Path: Optional. Specifies a file or list of files and folders to be added.

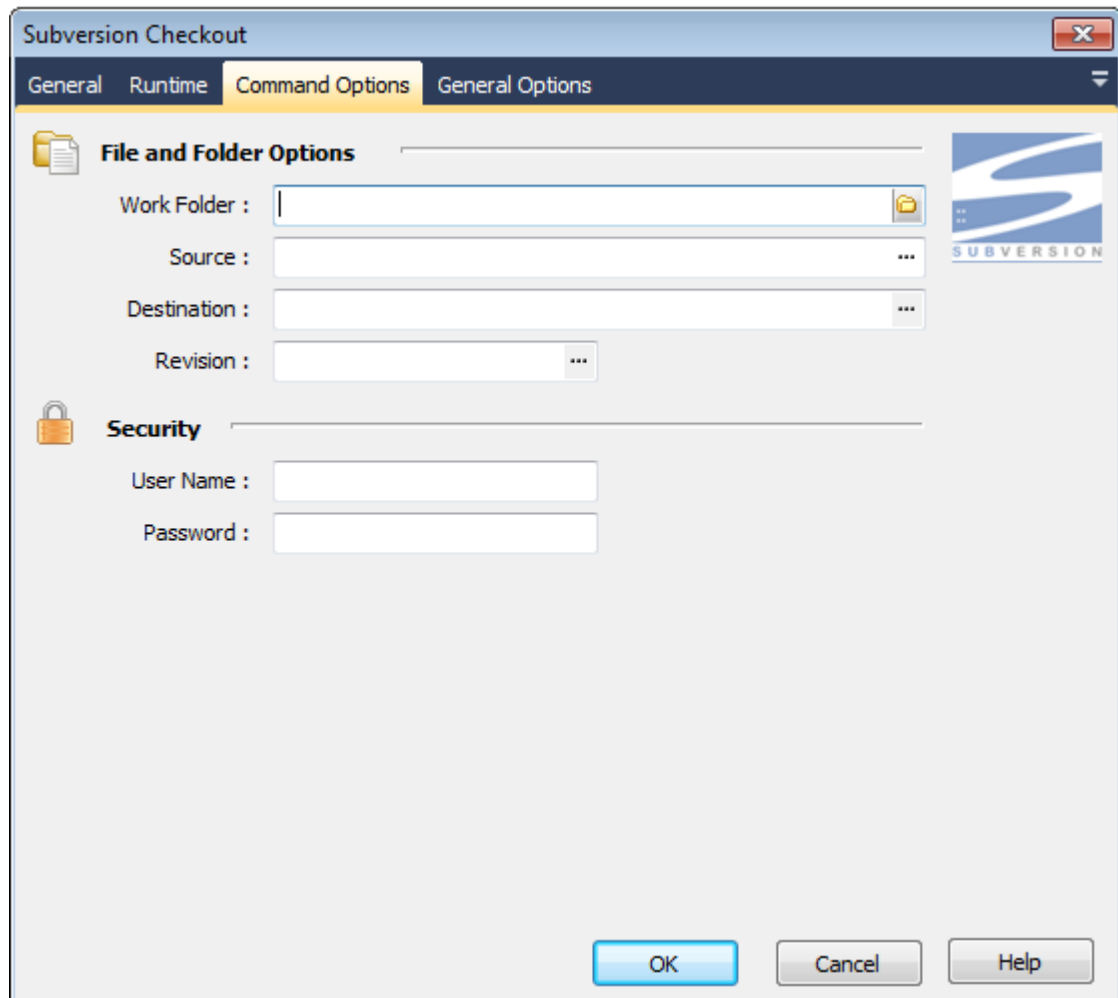
Note: See Subversion Actions for more information about the Work Folder and Path fields.

Targets: Gets the list of files to operate on from the file specified.

General Options: See Subversion Actions for more information.

6.34.18.2 Subversion Checkout

The Subversion Checkout action checks out one or more items from a repository into a working copy.



Work Folder: The folder where the action takes place.

Source: A space-separated list of items to check out.

Destination: The folder where the items will be checked out to. If not specified, the work folder is used.

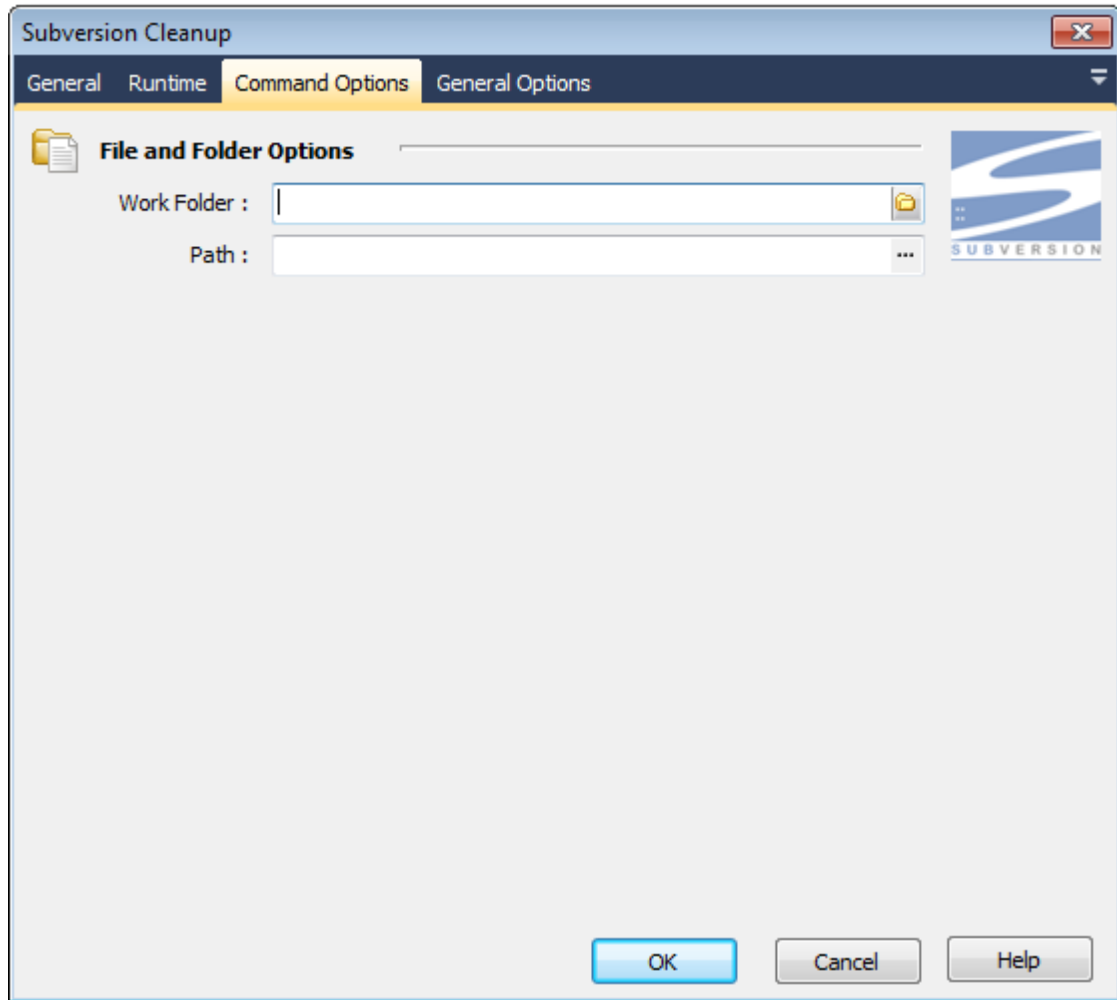
Revision: The revision of the item to check out.

General Options: See Subversion Actions for more information.

Note: See Subversion Actions for more information about the Work Folder and Path fields and the revision format specifier.

6.34.18.3 Subversion Cleanup

The Subversion Cleanup action recursively cleans up the working copy, removing locks and resuming unfinished operations. It is useful if a previous Subversion action has been interrupted, or if a lock has been detected.



Work Folder: The working copy to process.

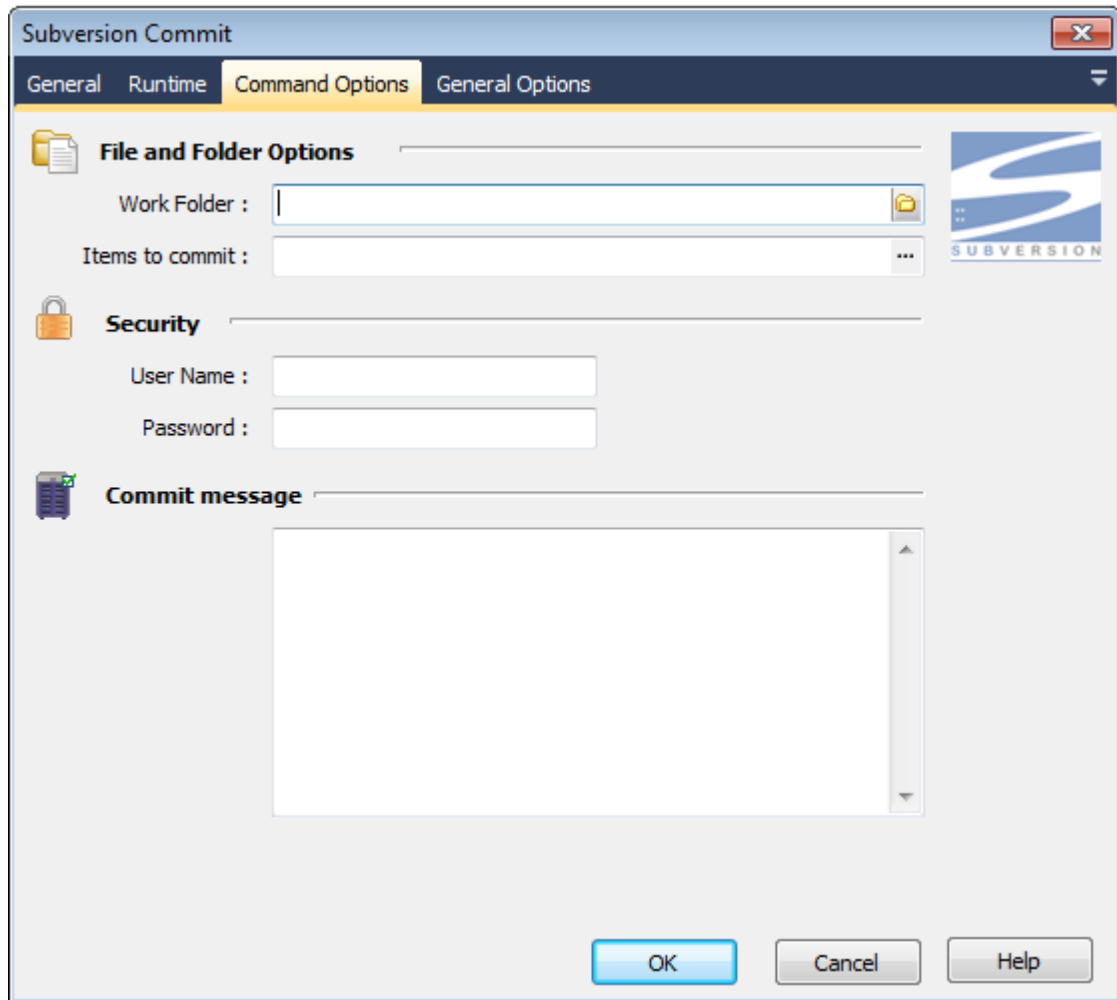
Path: Optional. To restrict the cleanup operation to certain files or folders, specify a file or list of files and folders for the action to operate on.

General Options: See Subversion Actions for more information.

Note: See Subversion Actions for more information about the Work Folder and Path fields, and the General Options tab.

6.34.18.4 Subversion Commit

The Subversion Commit action saves changes from a working copy to the repository. Only files and folders under version control are affected. To add files to version control, use the Subversion Add action.



Work Folder: The base working copy where items to commit are found.

Path: Optional. Specifies a space-separated list of folders and files within the working copy to commit. For example, "changes.txt src/source1.c" would commit only the changes.txt file in the root folder, and the source1.c file in the src folder. If left blank, all modifications found within the working copy are committed.

Note: See Subversion Actions for more information about the Work Folder and Path fields.

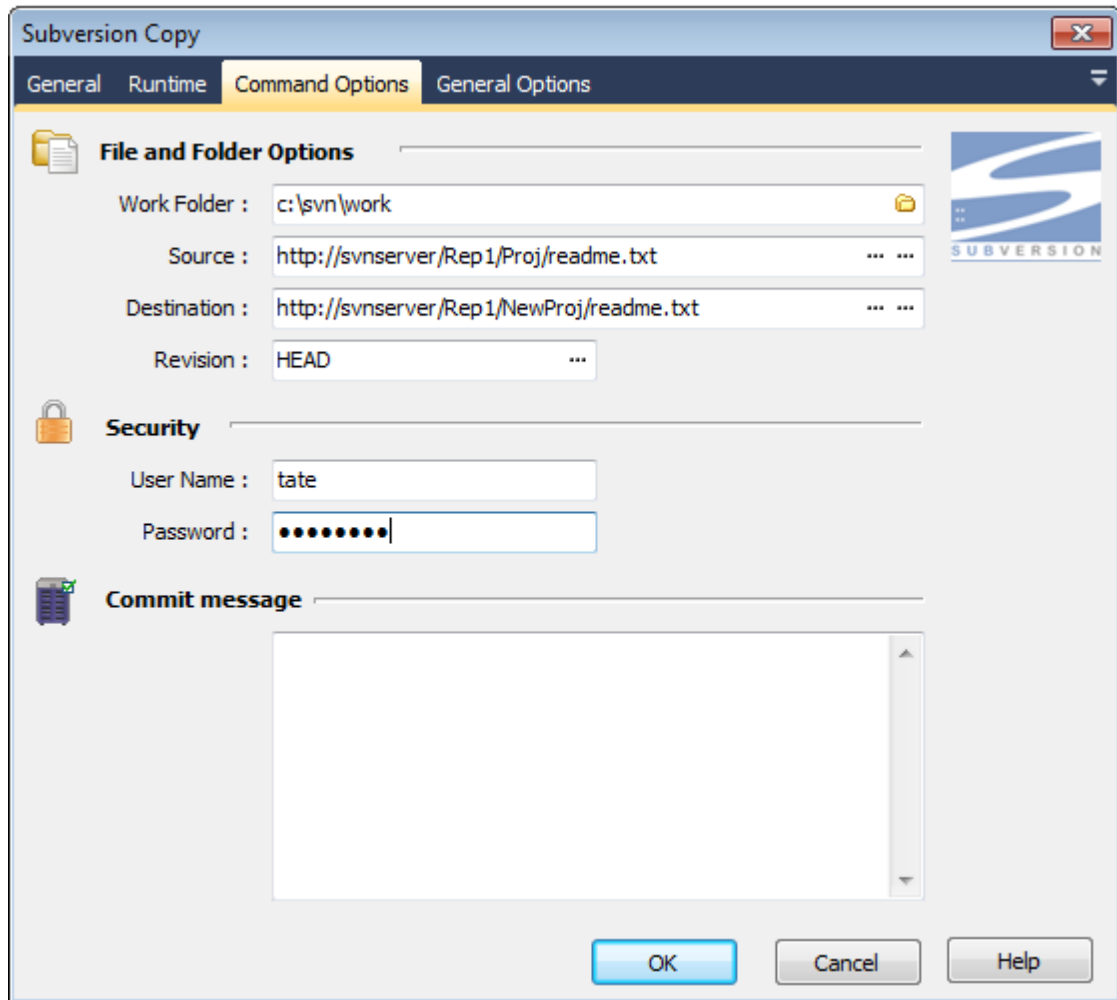
User Name/Password: Username and password, if using authentication.

Comment: Required. Message describing the changes, to be saved with commit in Subversion.

General Options: See Subversion Actions for more information.

6.34.18.5 Subversion Copy

The Subversion Copy action copies files or folders within a repository or working copy, or between a repository and a working copy. It is useful to recover deleted files, but can also be used to make duplicates of items, or to tag branches.



Work Folder: The location of the working copy. : See Subversion Actions for more information.

Work Folder

Specify a working folder for the action (this field is only necessary if relative paths are used for source or destination working folders.)

Source & Destination

Specify the source and destination for the copy. These entries can be Subversion repository URLs, or local working copy folders.

The source can be a file or a directory. Wildcard file names can be used. The destination must be a directory.

Behaviour

The behaviour of this action depends on whether you use local working copy folders or Subversion URLs for the copy operation. The behaviour is as follows:

Source	Destination	Behaviour
Local Working Copy	Local Working Copy	Copy and schedule for addition (with history)
Local Working Copy	Subversion URL	Immediately commit a copy of local directory to URL
Subversion URL	Local Working Copy	Check out URL into local directory, schedule for addition
Subversion URL	Subversion URL	Complete server-side copy; used to branch & tag.

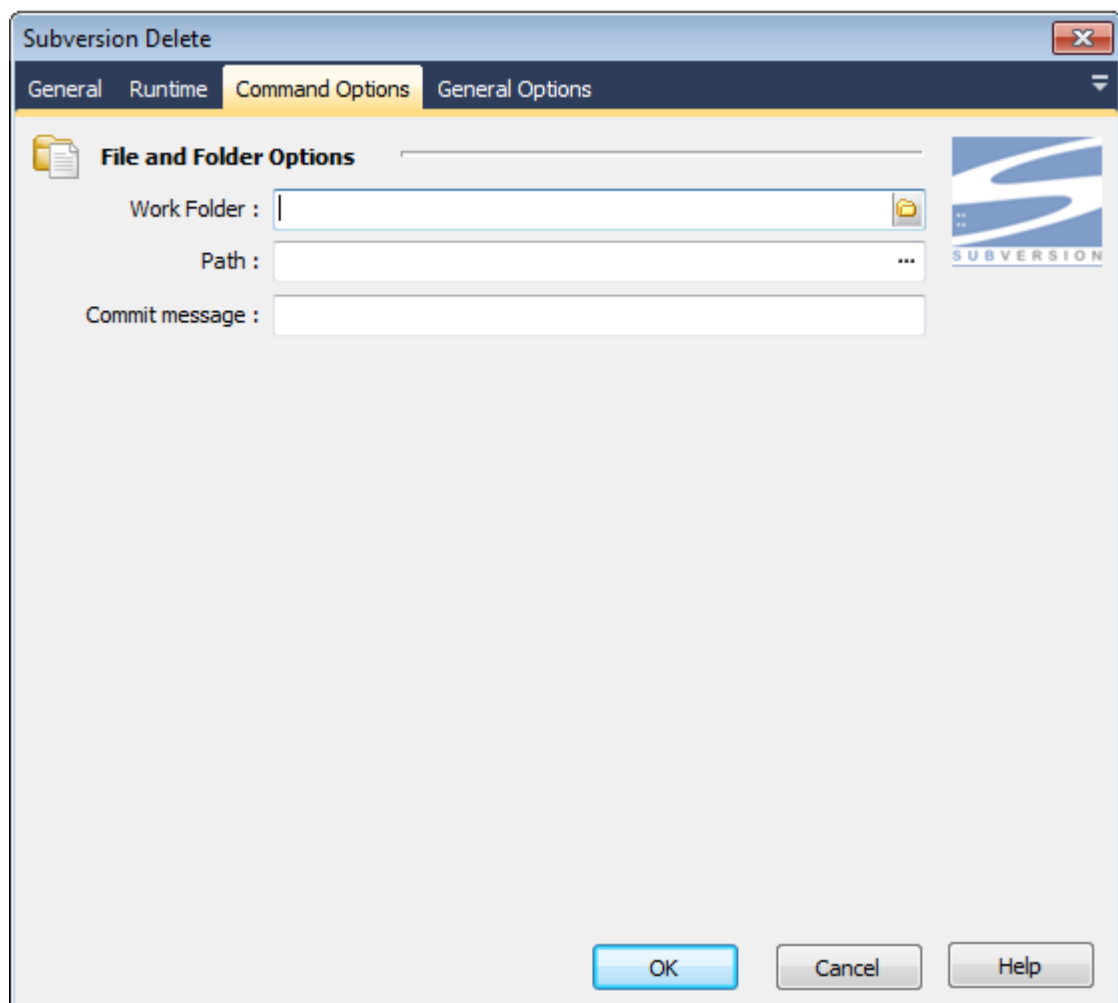
Revision: A specific revision of the source item. See Subversion Actions for more information on the format.

User Name/Password: Required if authentication is used.

General Options: See Subversion Actions for more information.

6.34.18.6 Subversion Delete

The Subversion Delete action removes an item from a Subversion repository or working copy. The deletion is not permanent, and remains in the repository's history.



Work Folder: The directory to work in.

Path: A file or space-separated list of files to delete.

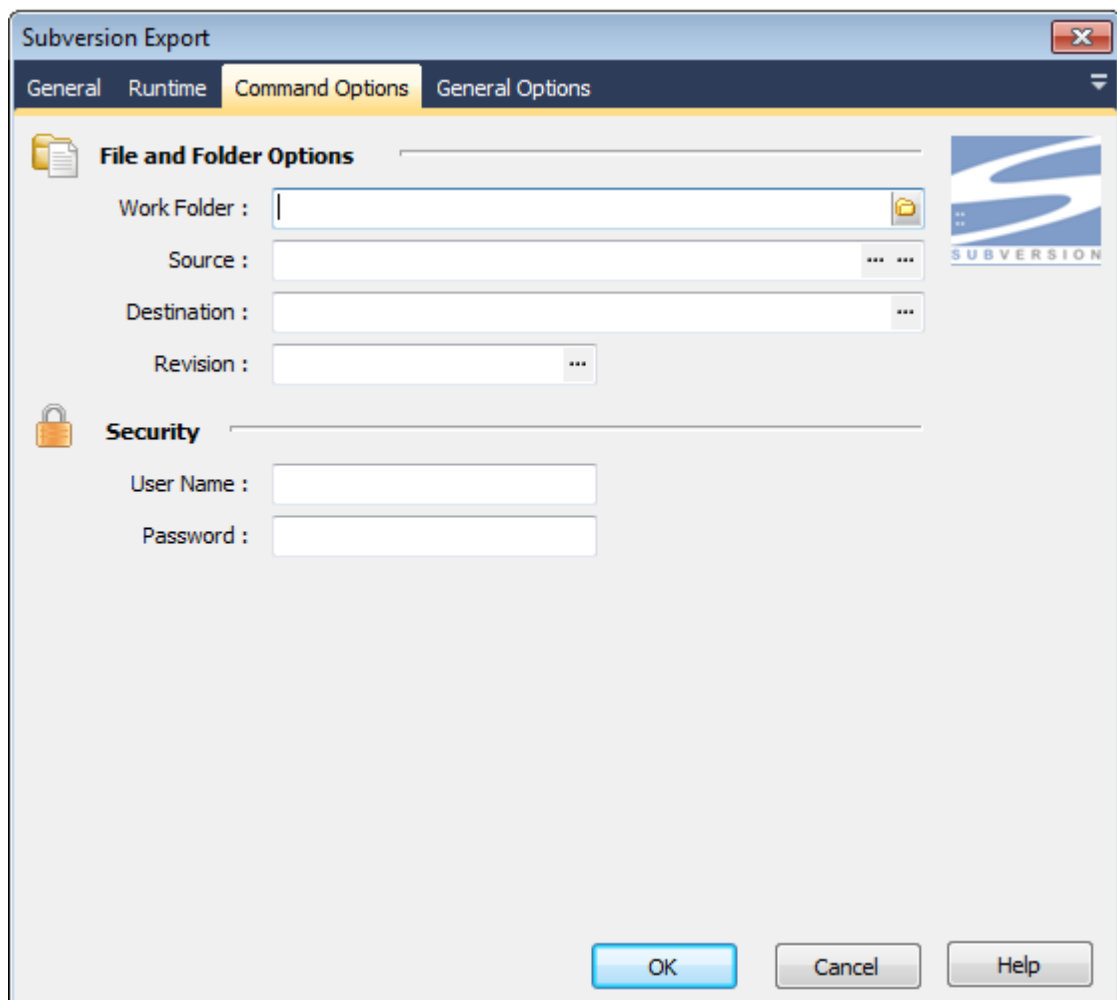
Note: See Subversion Actions for more information about the Work Folder and Path fields.

Message: A commit message for the deletion.

General Options: See Subversion Actions for more information.

6.34.18.7 Subversion Export

The Subversion Export action copies versioned files from a working copy or the repository, to another directory, allowing you to distribute the files without the .svn directory. Uncommitted changes in the working copy are by default retained in the export.



Work Folder: The work folder for the command. See Subversion Actions for more information.

Source: One or more folders to export, separated by spaces. If a URL is given, items will be exported from the repository at the given revision. If a path is given, all

versioned files from that folder are exported, for the given revision. See the Subversion documentation for more details.

Note: If PATH is omitted, the basename of the URL will be used as the destination. If multiple URLs are given each will be checked out into a sub-directory of PATH, with the name of the sub-directory being the basename of the URL.

Destination: The folder to export files to.

Revision: Optional. Specifies the revision being exported. See Subversion Actions for information on the revision format.

User Name/Password: Required if using authentication.

General Options: See Subversion Actions for more information.

6.34.18.8 Subversion Generic Action

The Subversion Generic Action allows you to run a Subversion command which is not otherwise supported by FinalBuilder.

The screenshot shows the 'Subversion Generic' dialog box with the 'Command Options' tab selected. The dialog has a title bar with a close button. Below the title bar are three tabs: 'General', 'Runtime', and 'Command Options'. The 'Command Options' tab contains four sections: 'File and Folder Options' with 'Work Folder' and 'Path' text boxes; 'Subversion Command' with a dropdown menu; 'Command Line' with a large text area; and 'Security' with 'User Name' and 'Password' text boxes. A Subversion logo is visible on the right side of the dialog. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Work Folder: The current directory from which the command is run. See Subversion Actions.

Path: A path to an item passed on the command line.

Subversion Command: The name of the Subversion command to run. You can choose from the drop down list, or enter a command manually.

Command Line: Any other arguments required by the command.

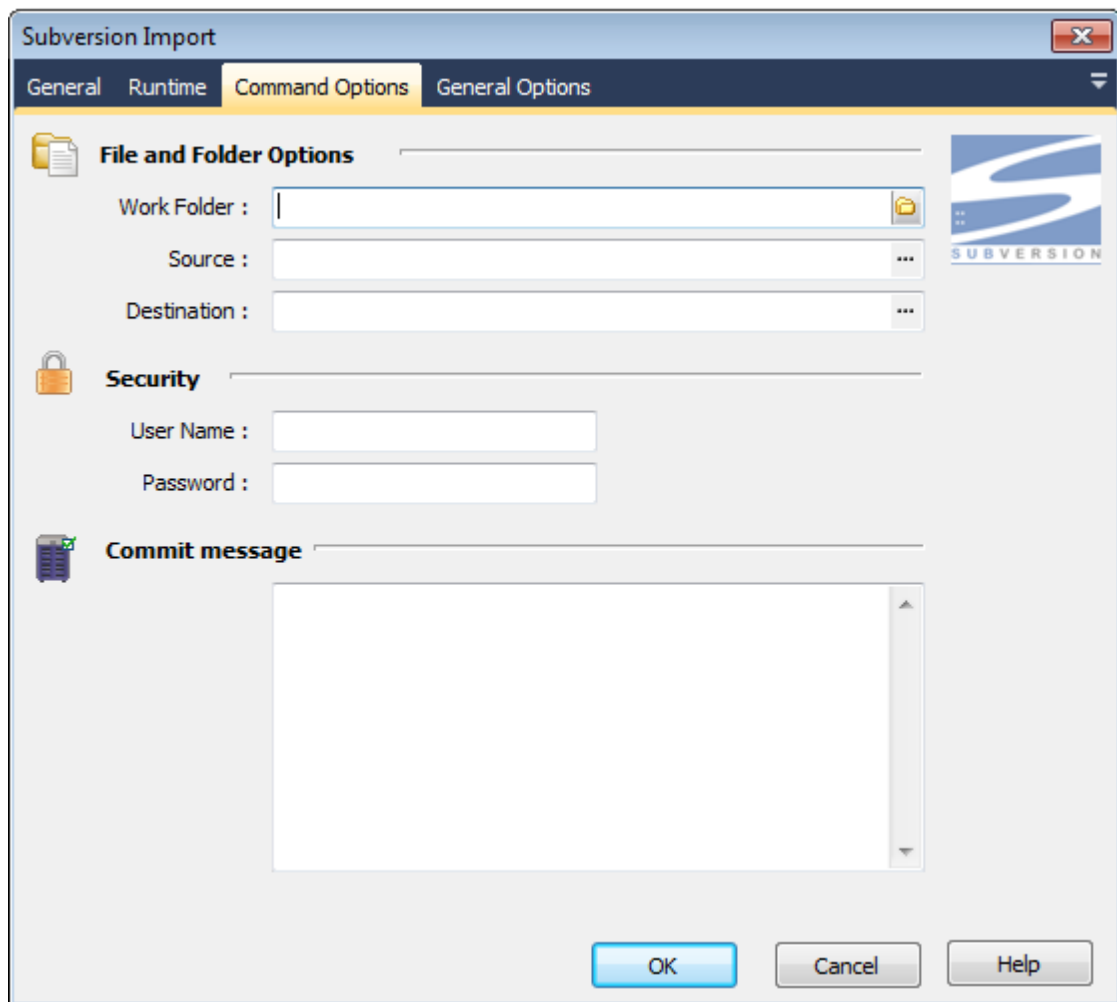
Note: The final command line is constructed as follows:

svn <Command> <Paths List> <Command Line>

General Options: See Subversion Actions for more information.

6.34.18.9 Subversion Import

The Subversion Import action recursively adds and commits an unversioned file or tree into the repository.



Work Folder: The base folder to work from. See Subversion Actions for more information.

Source: The file or folder to import.

Destination: The URL to import to.

User Name/Password: Required if using authentication.

General Options: See Subversion Actions for more information.

6.34.18.1 Subversion Info

The Subversion Info action retrieves information about a path in a repository or working copy, and stores it in one or more variables.

The screenshot shows the 'Subversion Info' dialog box with the 'Command Options' tab selected. The 'File and Folder Options' section includes 'Work Folder', 'Path', and 'Revision' fields. The 'Target Variables' section lists various Subversion metadata fields. The 'Security' section has 'User Name' and 'Password' fields. The dialog also features a Subversion logo and standard 'OK', 'Cancel', and 'Help' buttons.

Work Folder: The current directory from which the command is run. See Subversion Actions.

Path: A path to a file or folder about which you want to retrieve information. Should be specified as a file path for an item in a working copy, or as a URL for an item in the repository.

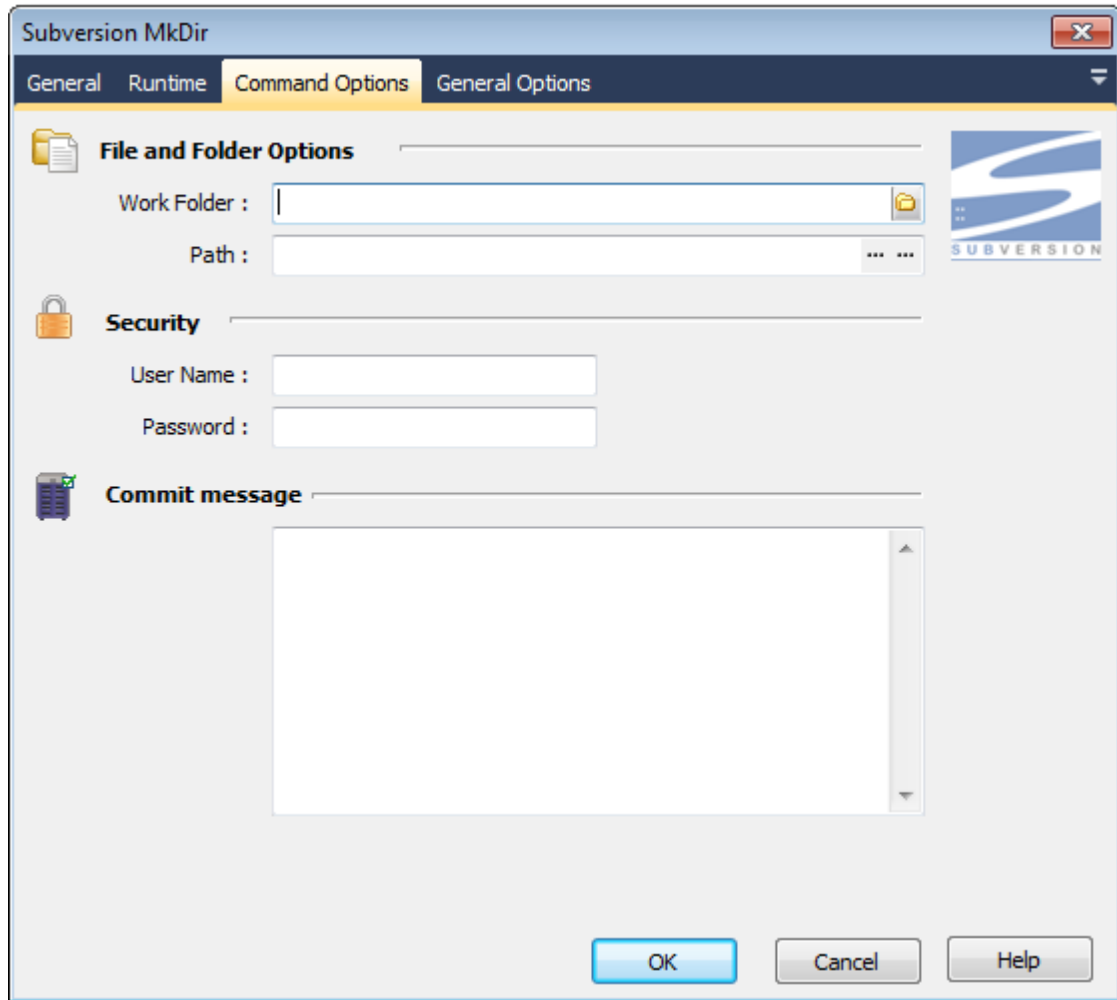
Revision: Optional. A revision specifier for the specified item. If omitted, the HEAD revision is used (for repository items), or either HEAD and uncommitted changes (for working copy items). See Subversion Actions for the format of revision specifiers.

User Name/Password: Optional. Username and password if authentication is required.

General Options: See Subversion Actions for more information.

6.34.18.1 Subversion Mkdir

The Subversion Mkdir command creates a new directory, either in a repository, or in a working copy. In the first case, it is immediately committed. In the latter case, it is added to version control but not committed.



Work Folder: The current directory from which the command is run. See Subversion Actions.

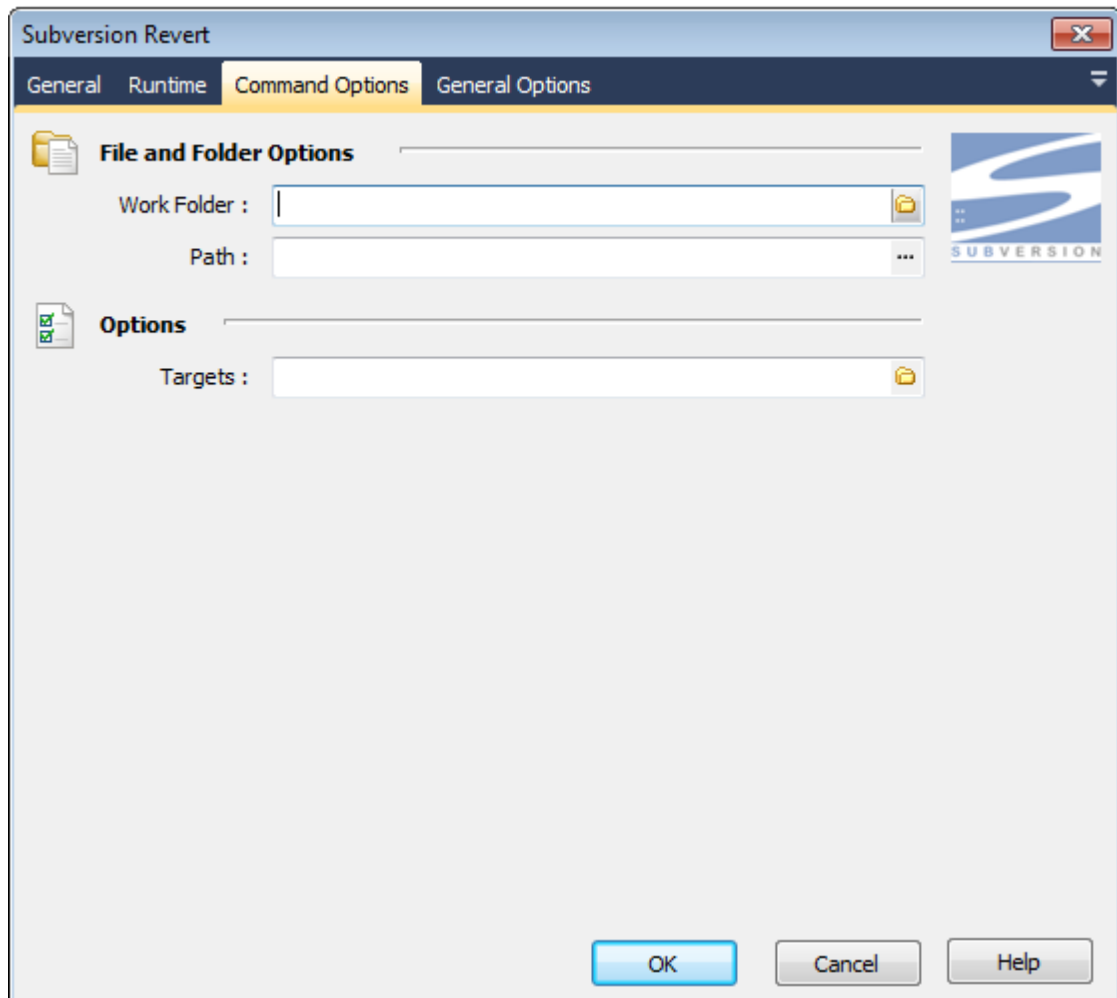
Path: The full path being created, either as a URL or a local directory. Note that all intermediate paths must exist. That is, if creating the path /docs/prod1/test, the path /docs/prod1 must already exist.

User Name/Password: Optional. Subversion username and password if authentication is used.

General Options: See Subversion Actions for more information.

6.34.18.1 Subversion Revert

The Subversion Revert action removes local modifications to items in a working copy, effectively "undoing" any uncommitted changes.



Work Folder: The current directory to run the action from.

Path: Optional if using a targets file. A space-separated list of files and folders to be reverted.

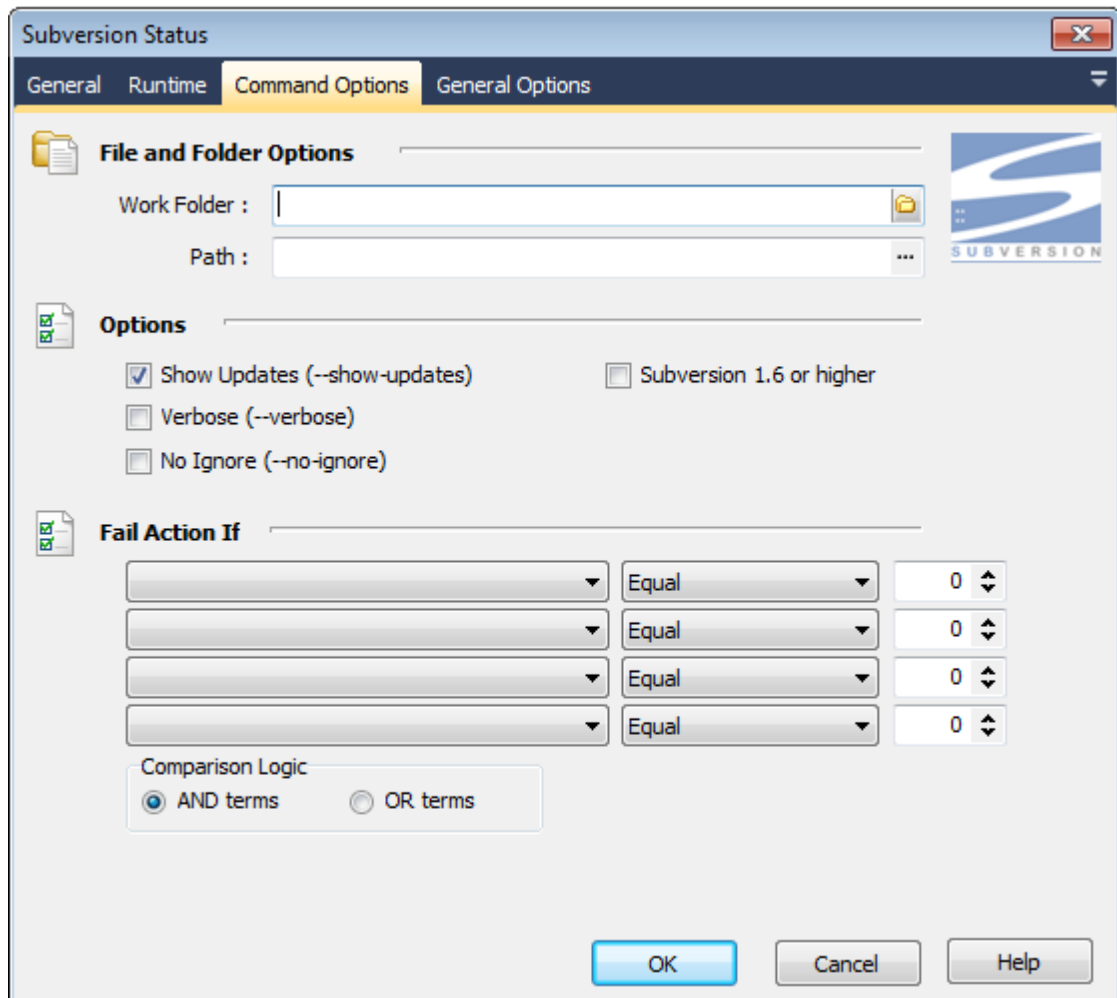
See Subversion Actions for more information on the Work Folder and Path.

Targets: Optional. Specifies a file containing a list of files and folders to be reverted.

General Options: See Subversion Actions for more information.

6.34.18.1 Subversion Status

The Subversion Status action allows you to control program flow based on the current state of items in a working copy.



This action acts like a conditional branch based on the total number of items in a given state, such as modified or locked. For example, you can choose to have the action fail if at least 1 of 3 specified items are locally modified.

Work Folder: The current directory to run the action from.

Path: Optional. A space-separated list of files and folders to be examined. File masks (e.g., *.txt) can be used. If no path is specified, then all files in the work folder are treated. If no files match a file specification, the action fails.

Show Updates: Compares items against the repository. You must check this box if using the "Total items with Newer Revisions" option.

Verbose: Outputs full revision information on all items into the log, including those that do not match file specifications on the path. This does not affect the behaviour of the action otherwise.

No Ignore: Tells Subversion to ignore its list of ignored files.

Fail Action If: You can define up to 4 conditions that are checked, and combined with either the AND or OR operator. FinalBuilder tallies up all the items that have each status, such as modified, missing, replaced and so forth. You then choose a status, a count and an operator which will cause the action to fail. For example, *Total Items Modified Equal 0* means the action fails if no items have been modified. The possible statuses are as follows:

- Total Items Modified
- Total Items Unmodified
- Total Items Added
- Total Items Deleted
- Total Items in Conflict
- Total Items related to Externals Definition
- Total Items Ignored
- Total Items Not Under Version Control
- Total Items Missing
- Total Items with Type Mismatch
- Total Properties Modified
- Total Properties UnModified
- Total Properties in Conflict
- Total Items Unlocked
- Total Items Locked
- Total Items with History with Conflict
- Total Items Switched
- Total Items with Newer Revisions (requires Show Updates checkbox)

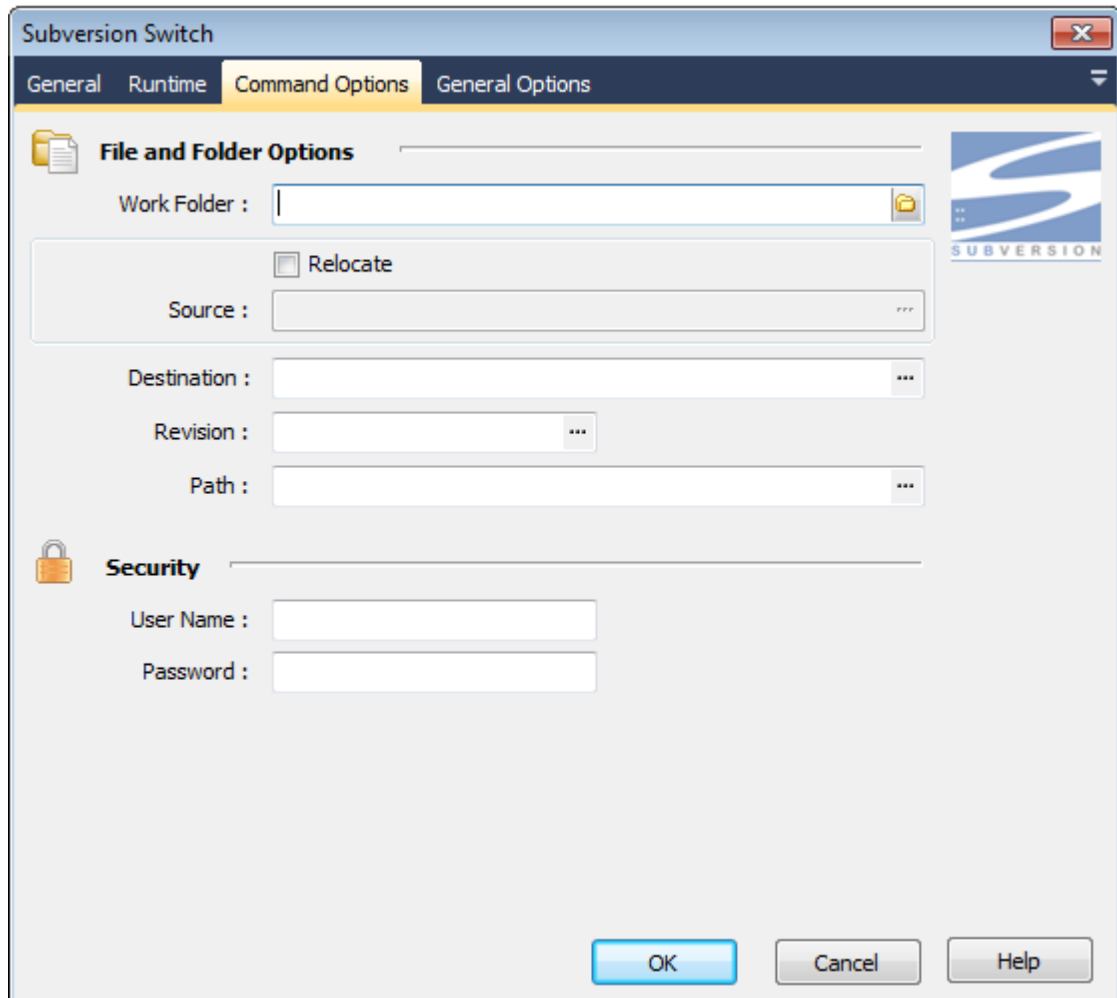
See the Subversion documentation for details on the meanings of these statuses.

Comparison Logic: "AND terms" indicates that the action fails only if all the conditions are met. "OR terms" indicates that the action fails if any of the conditions are met.

General Options: See Subversion Actions for more information.

6.34.18.1 Subversion Switch

The Subversion Switch action moves a working copy to a different URL.



The Switch action is complex and can have serious consequences. Read the Subversion documentation before using this action.

Work Folder: The location the command is run from.

Relocate: Used when the location of the repository itself has moved. Corresponds to the --relocate command line argument.

Source: The old location of the repository, if using the Relocate option.

Revision: The repository reversion to switch to. See Subversion Actions for information on the format.

Destination: The URL to switch to.

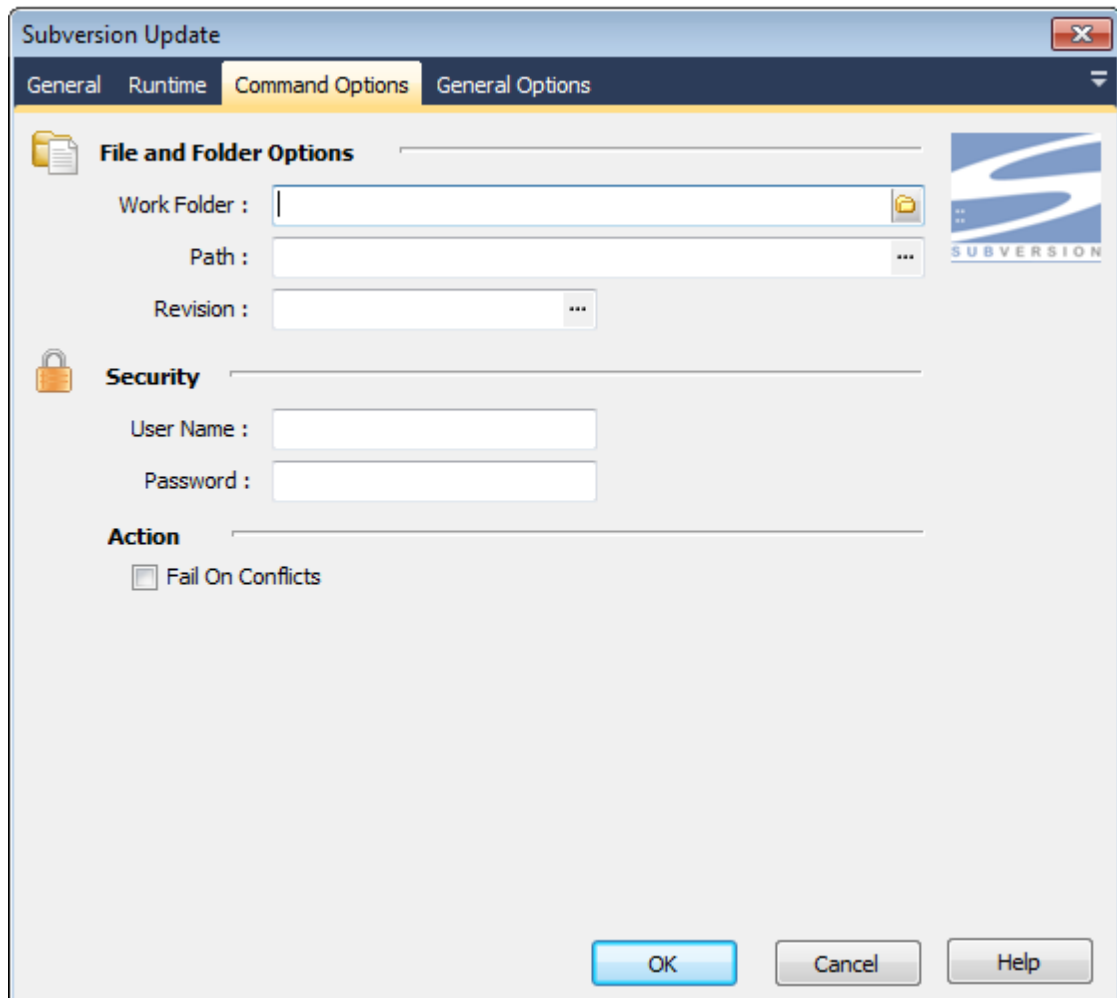
Path: The path to the working copy being switched. Can be specified as a relative path from the work folder. See Subversion Actions for more information about the Work Folder and Path fields.

User Name/Password: User name and password if using authentication

General Options: See Subversion Actions for more information.

6.34.18.1 Subversion Update

The Subversion Update action synchronises a working copy with the repository, by copying and merging files from the repository into it.



Work Folder: The current directory from which the command is run. See Subversion Actions.

Path: Optional. A space-separated list of paths to working copies to be updated.

Revision: Optional. The repository revision to update to. If not specified, the HEAD revision is used. See Subversion Actions for more information.

User Name/Password: User name and password if using authentication

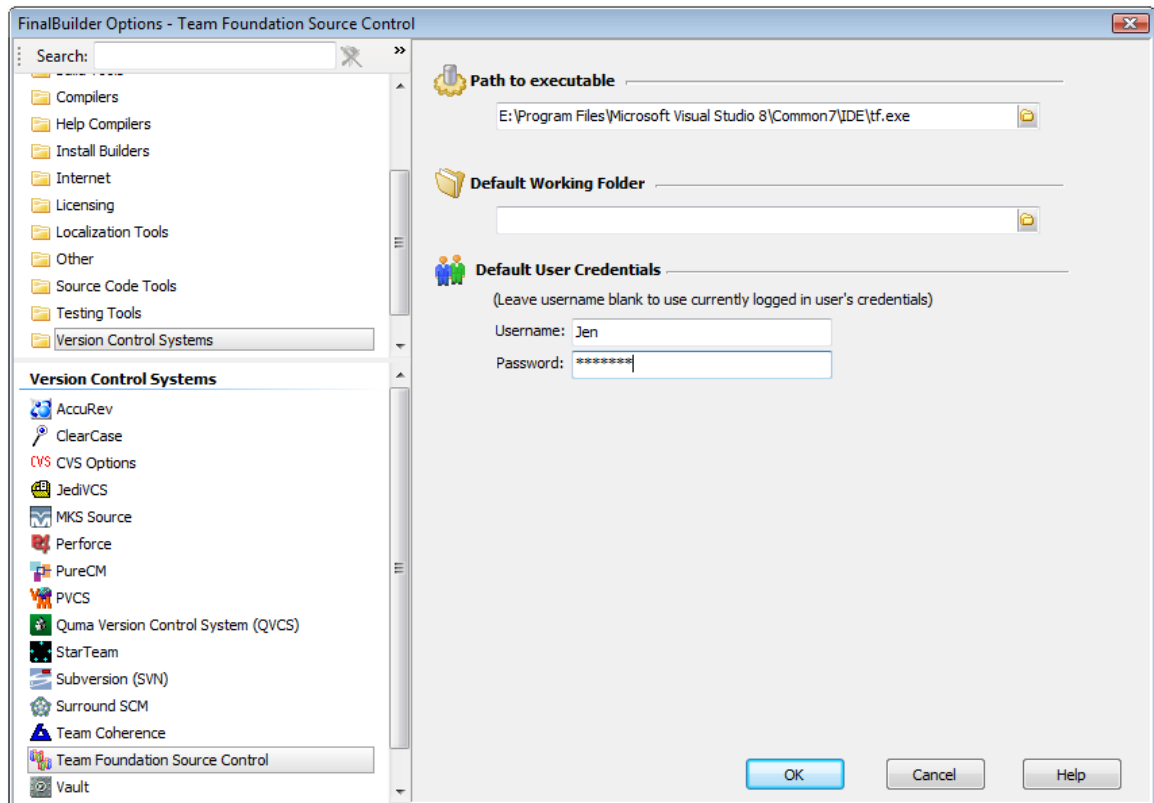
Fail On Conflicts: Causes the action to fail if any merge conflict is found during the update. That is, if SVN reports "C" for the status of any item, the action will fail, but the full update will still have taken place.

General Options: See Subversion Actions for more information.

6.34.19 Team Foundation Source Control

6.34.19.1 Options

To edit Team Foundation options, choose Options from the Tools menu then click on "Version Control Systems" -> "Team Foundation Source Control."



Path to executable

Specify the path to the "tf.exe" Team Foundation command line tool. This is the tool that FinalBuilder uses to access Team Foundation.

If this field is blank, the Team Foundation actions will attempt to autodetect the tool's location. Auto detection is normally successful, but if it fails then it is possible to enter the executable path here.

Default Working Folder

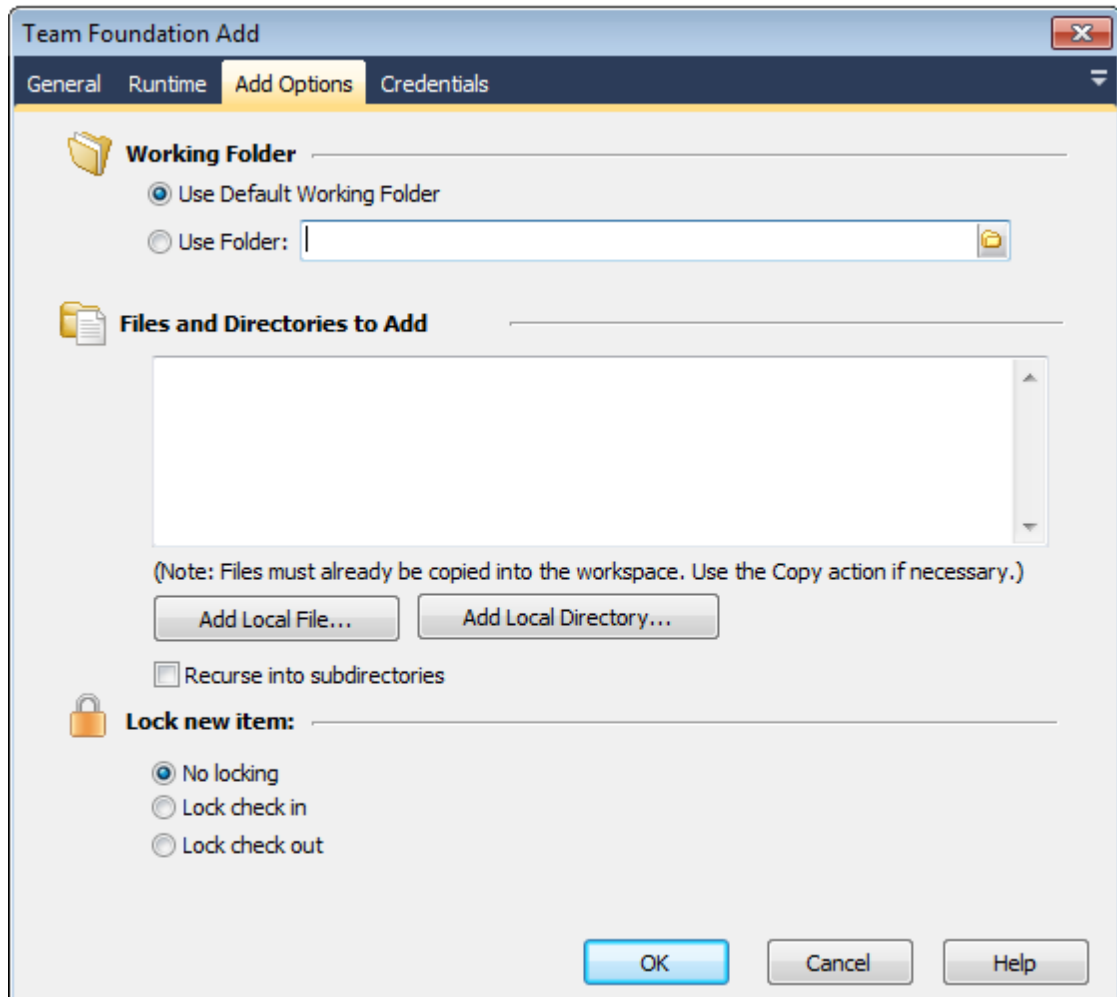
All of the Team Foundation Source Control actions determine the server, project and repository context from a local working folder. You will need to set up working folders in Visual Studio, or with Map Working Folder Action.

You can specify a default working folder for all actions, or specify individual working folders on a per-action basis.

6.34.19.2 Team Foundation Add

Use the Team Foundation Add action to add new file(s) to the repository.

The added files will appear as "pending changes" to the repository and **need to be checked in** before they become permanent.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and Directories to Add

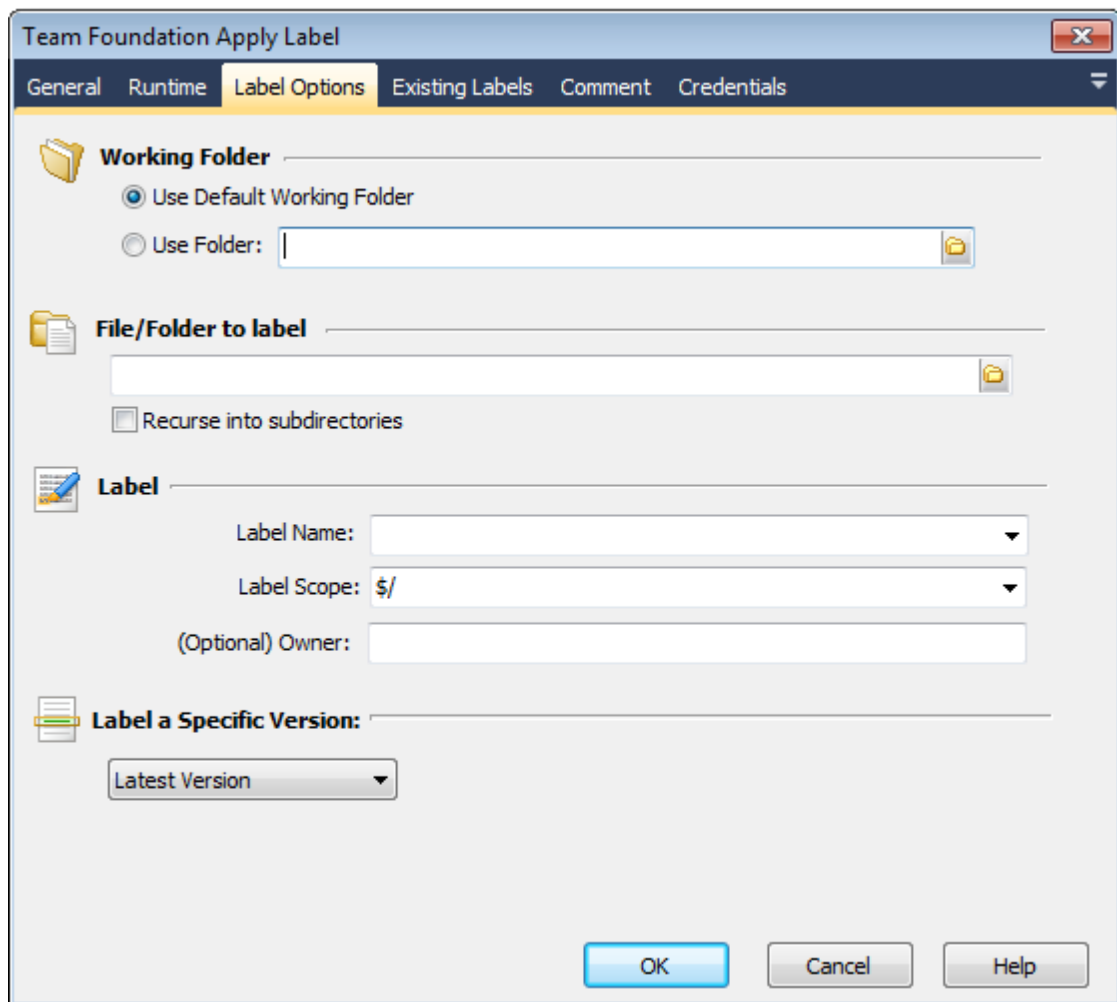
Specify the local paths of files and directories to add. The files will need to be already copied somewhere inside the working folder. Use the Copy File(s) (or Copy/Move File List) actions if necessary.

Lock new items

The new items can be automatically locked against check out (preventing others from checking them out) or check in (preventing others from checking in new versions) until you check in pending changes.

6.34.19.3 Team Foundation Apply Label

Use the Apply Label action to label a file or group of files. The label will not become permanent until a Check In Pending Changes operation takes place.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

File/Folder to label

Specify the name of the file or folder to label. Use wildcards (ie *) to select groups of files.

Label

Specify a name and a scope for the label. You can also specify an owner for the label (if the owner is someone other than the currently logged in user.)

Label a Specific Version

Instead of labelling the latest version of the file, you can choose to label a specific version. Versions can be selected by changeset number, label, a specific workspace or a specific date.

Label Options**"Replace Pre-Existing Labels"**

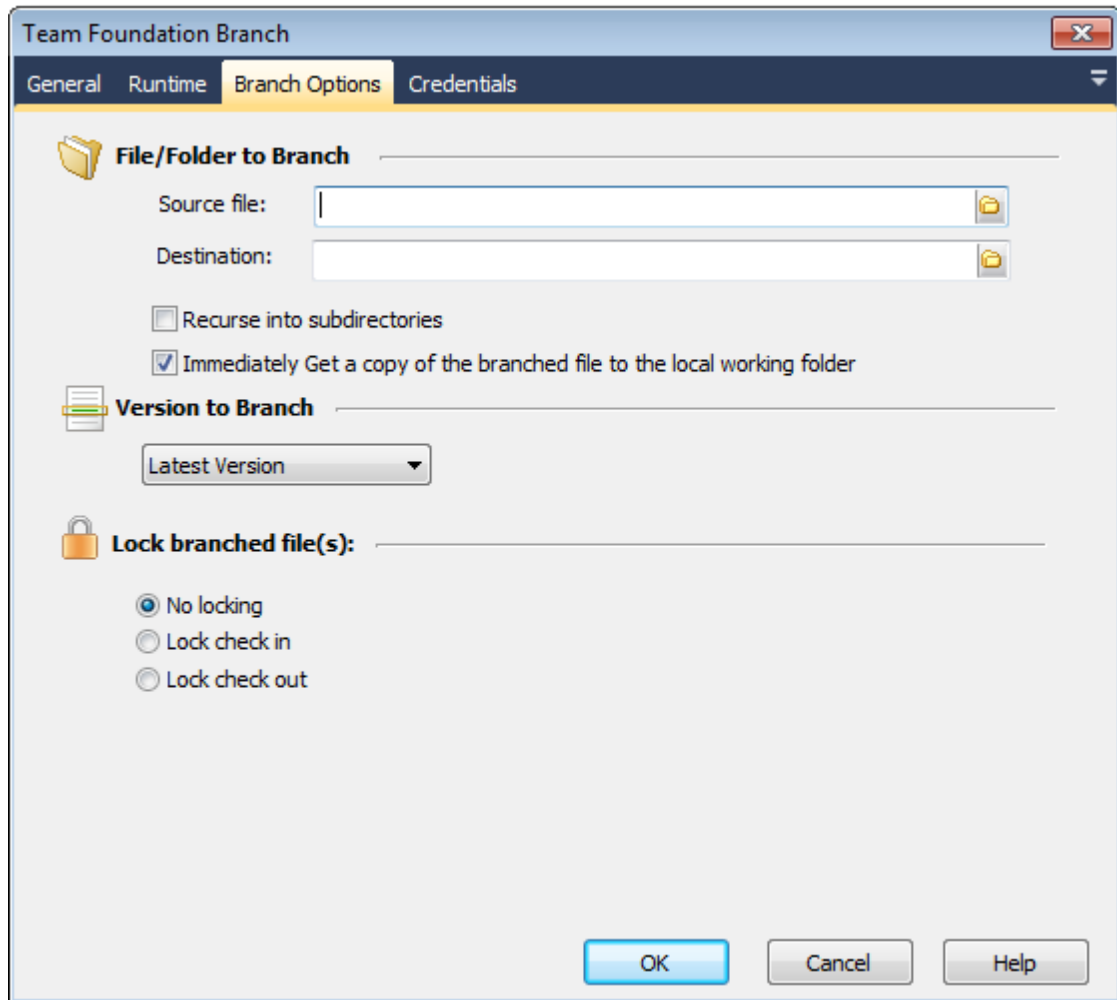
Other labels on the file(s)/directories will be removed. You must have sufficient privileges to remove labels belonging to other users.

"Ignore items with pre-existing labels"

If a file/directory is already labelled, it will not be labelled again.

6.34.19.4 Team Foundation Branch

Use the Team Foundation Branch action to create a new branch based on file(s) or directories. The branch will not become permanent until a Check In Pending Changes operation takes place.



File/Folder to Branch

Source file : Specify the file/directory to branch. Use wildcards (*) to specify multiple files.

Destination : Specify a file, or a folder, to branch to. If you specify a Team Foundation Source Control folder, the branched files will be created inside it.

"Immediately Get a copy of the branched file..."

Check this option to automatically retrieve a copy of the branched file.

Version to branch

Instead of branching from the latest version of the file, you can choose to label a specific version. Versions can be selected by changeset number, label, a specific workspace or a specific date.

Locked branched file(s)

Check this option to lock the branched files against check in or check out by other

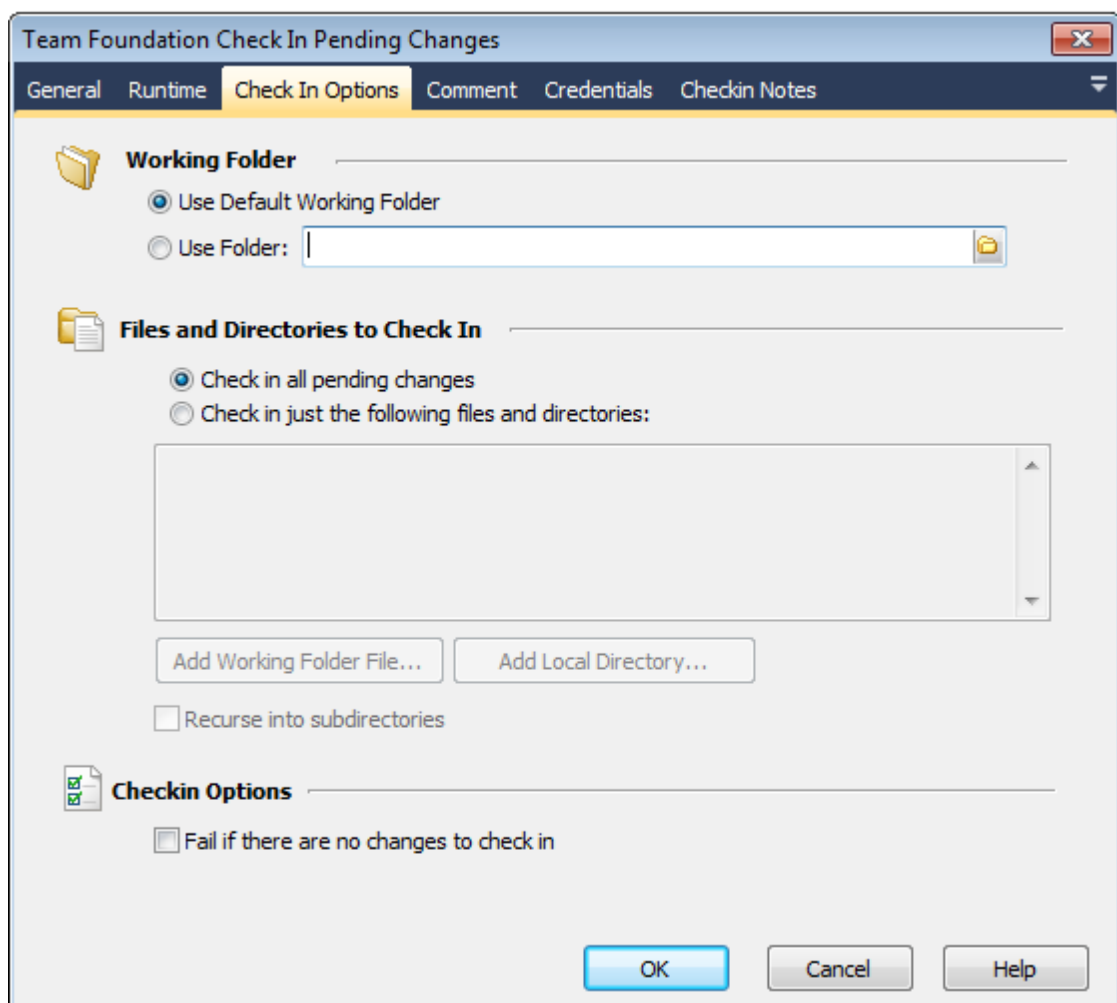
users, before the pending changes are checked in.

6.34.19.5 Team Foundation Check In Pending Changes

Use the Team Foundation Check In Pending Changes action to check in changes made to checked, new files, renamed, labelled, deleted or undeleted files.

Nearly all Team Foundation operations need to be checked in before they become permanent. Think of "Check In Pending Changes" as more of a "commit" operation than a "check in" operation (ie it needs to be done following most changes to a Team Foundation repository.)

(Instead of checking in, groups of pending changes can be Shelved or Undone.)



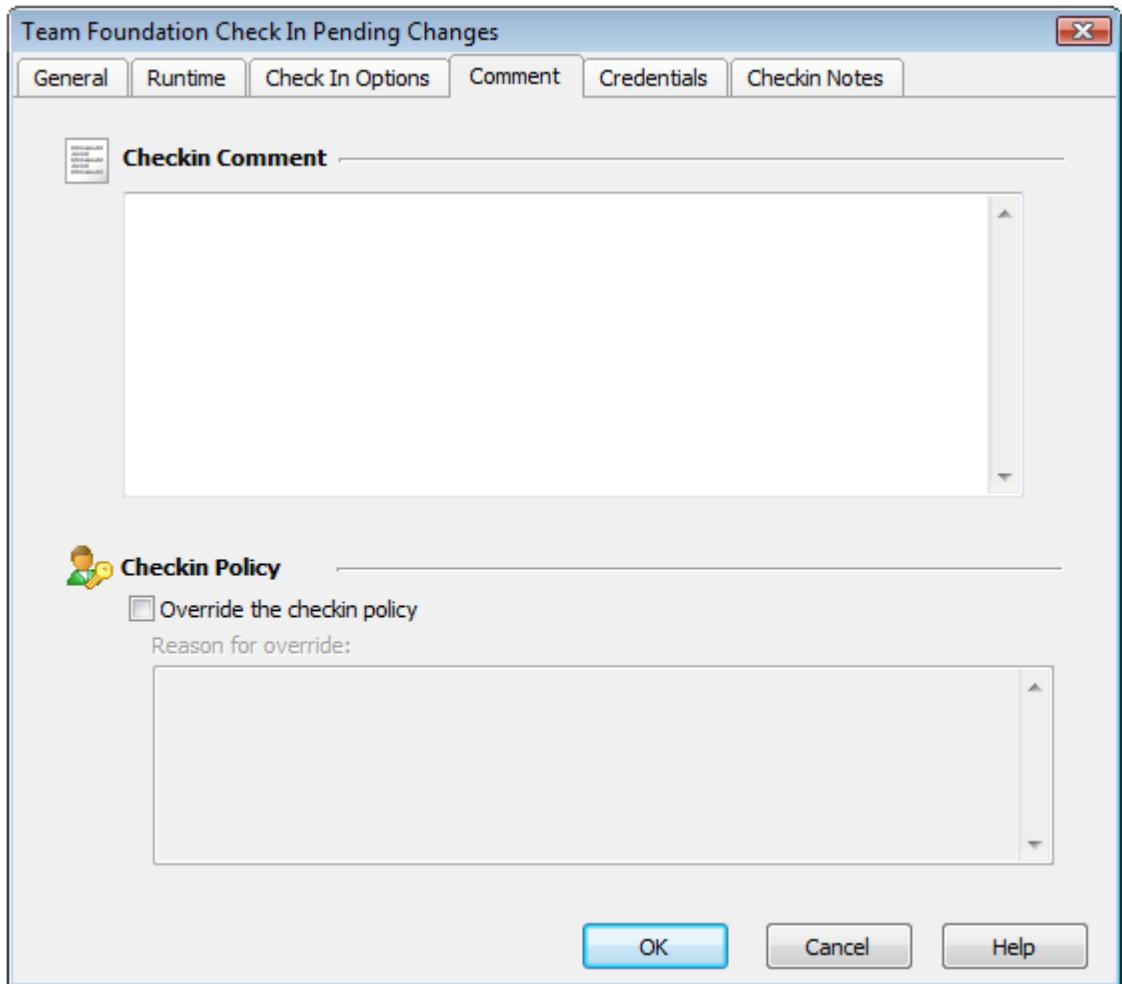
Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and Directories to Check In

You can check in all pending changes in a working folder, or specify specific files and directories to check in. Put each file/directory on a new line. Use wildcards (ie *) to select groups of files.

Comment

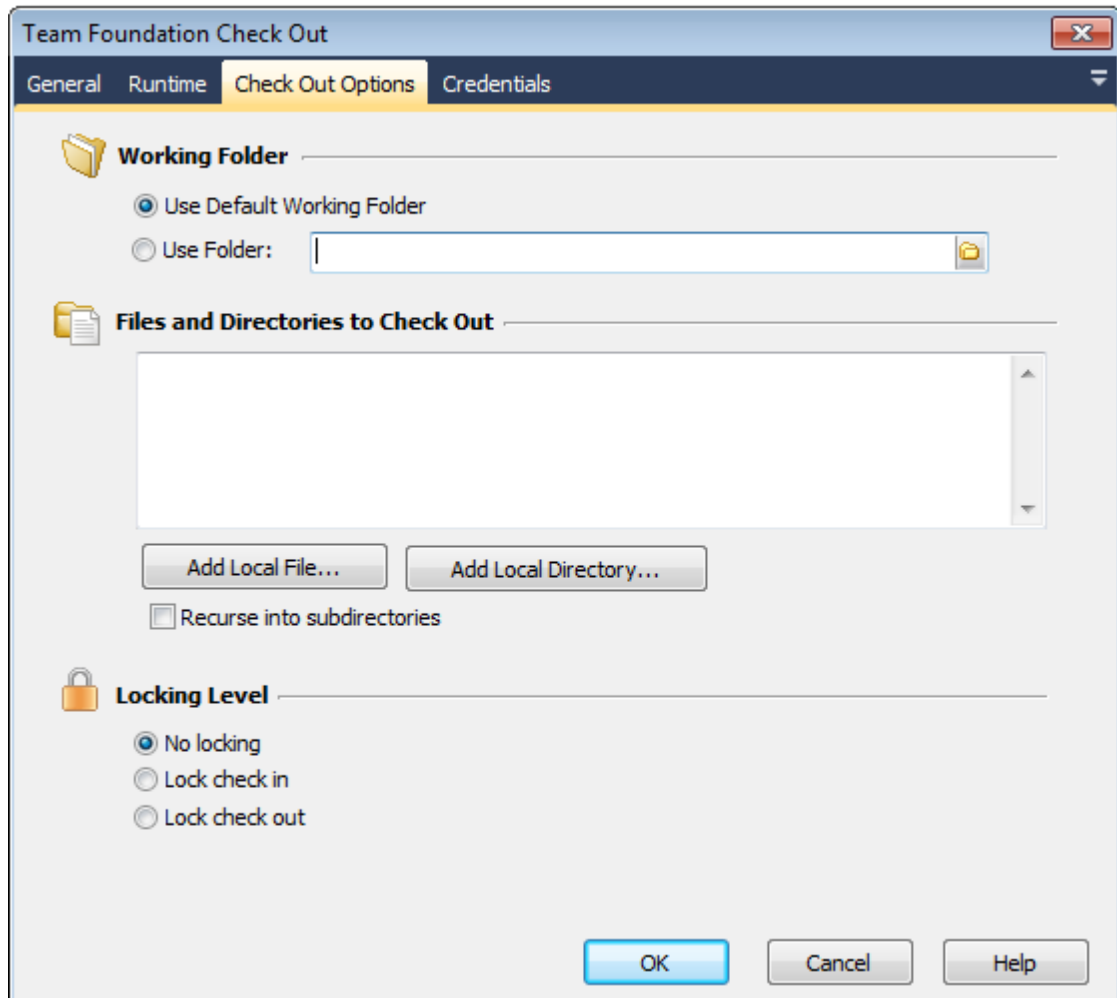


The screenshot shows the 'Team Foundation Check In Pending Changes' dialog box with the 'Comment' tab selected. The dialog has a title bar with a close button. Below the title bar are tabs: 'General', 'Runtime', 'Check In Options', 'Comment' (selected), 'Credentials', and 'Checkin Notes'. The 'Checkin Comment' section has a large text area for entering a comment. Below this is the 'Checkin Policy' section, which includes a checkbox labeled 'Override the checkin policy'. If this checkbox is checked, a 'Reason for override:' text area is visible. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Specify a checkin comment for your checkin. If the checkin is likely to violate the checkin policy, you can choose to override the checkin policy (with a reason.)

6.34.19.6 Team Foundation Check Out

Use the Team Foundation Check Out action to check out one or more files from the repository.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and directories to Check Out

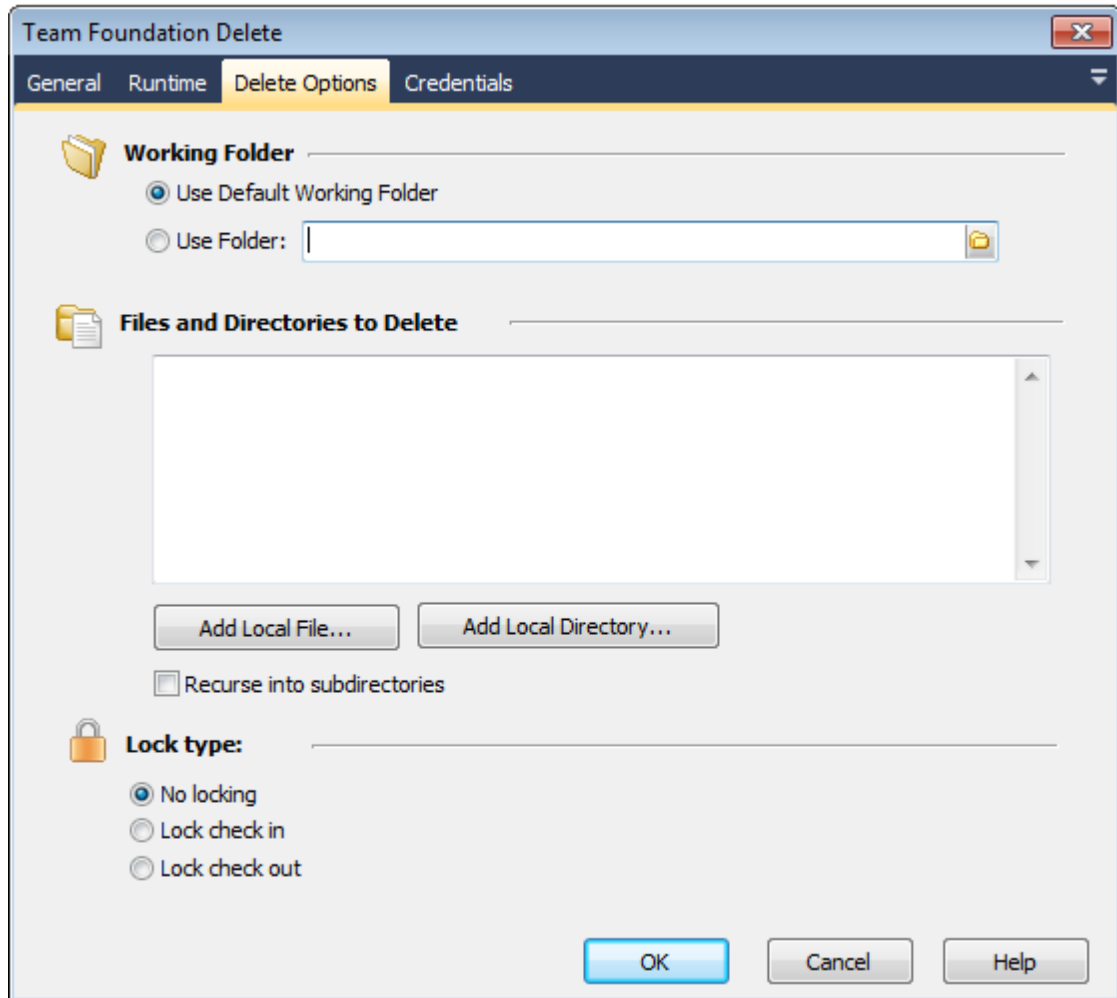
Specify the files and directories to check out. Put each new file/directory on a new line. Groups of files can be specified with wildcards (ie *.)

Locking Level

Specify the locking level for the checked out files. "Lock check in" prevents others from checking in changes to the same files. "Lock check out" prevents others from checking out the same files.

6.34.19.7 Team Foundation Delete

Use the Team Foundation Delete action to remove files from the repository. The delete operation will remain pending until Check In Pending Changes is invoked.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and Directories to Delete

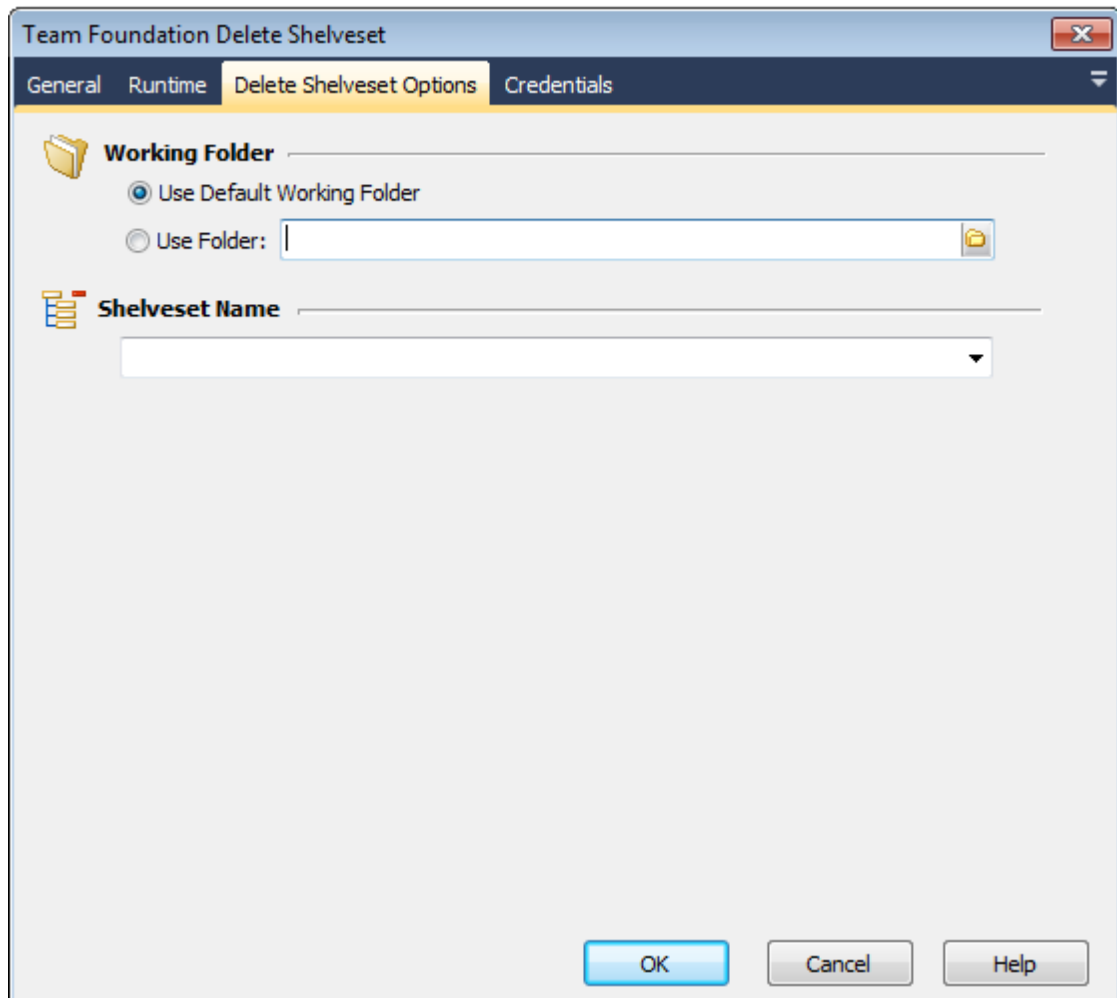
Specify the names of files and directories to delete. Put each new name on a new line. Use wildcards (ie *) (cautiously) to remove groups of files.

Lock Type

Select a lock type to lock the deleted files from check in/check out until the delete is checked in (the delete remains pending until Check In Pending Changes is used.)

6.34.19.8 Team Foundation Delete Shelveset

Use this action to delete a shelved changeset.



Working Folder

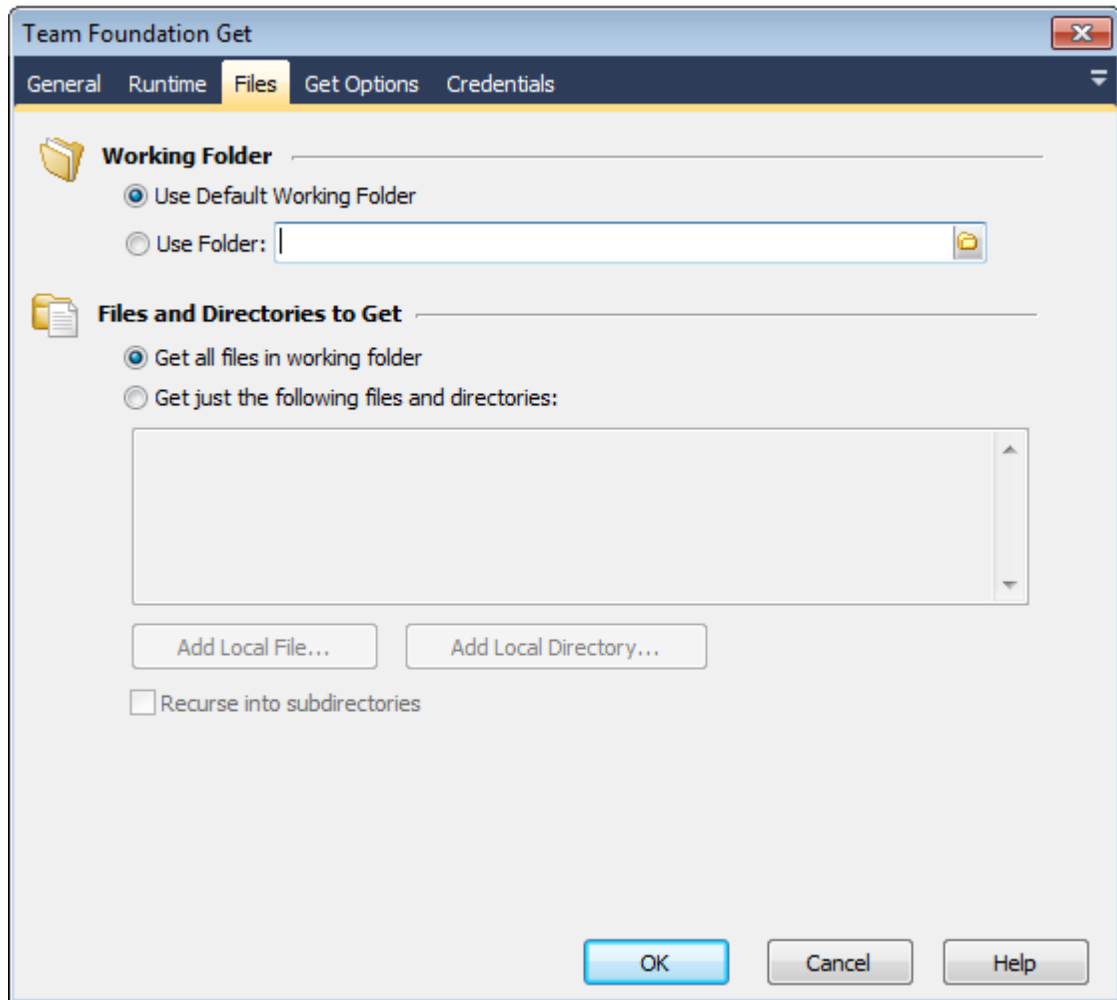
Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Shelveset Name

Specify the name of the shelveset to delete. Click on the dropdown to see a list of current shelvesets associated with the specified working folder.

6.34.19.9 Team Foundation Get

Use the Team Foundation Get action to update a local working folder with the contents of the repository.

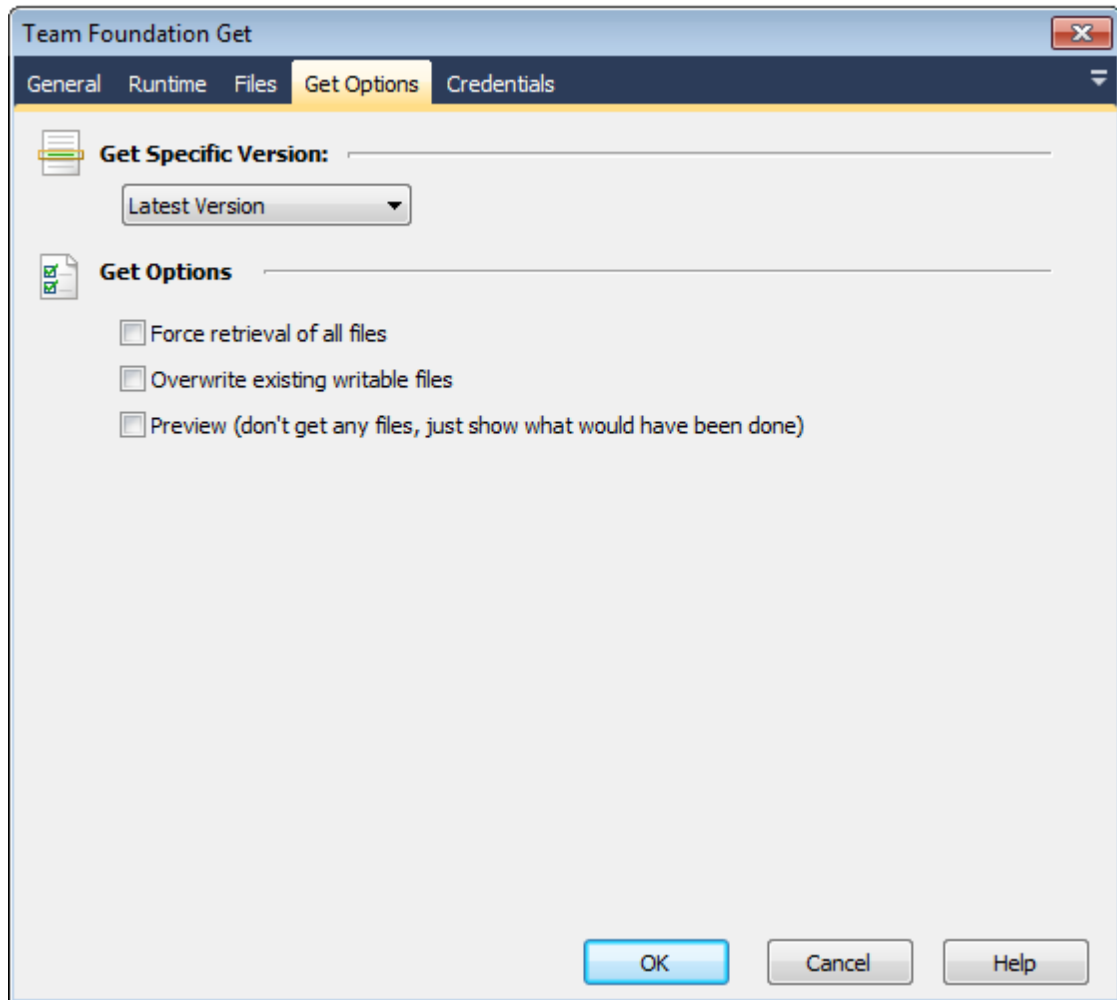


Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and Directories to Get

You can choose to get all the files in your working folder which will recursively get all files in the current workspace, or you can specify a list of files and directories to get. When '**Recurse into subdirectories**' is enabled then all files that match the file spec in the root directory and any subdirectory will be retrieved from the server. Use wildcards (ie *) to select groups of files. Put each new file/directory on a new line.



Get Specific Version

You can get the latest version of files from the repository, or a specific version. Versions can be selected by changeset number, label, a specific workspace or a specific date.

Get Options

"Force retrieval of all files"

If this option is set, all files will be retrieved from the repository regardless of whether or not they appear to have changed.

"Overwrite existing writable files"

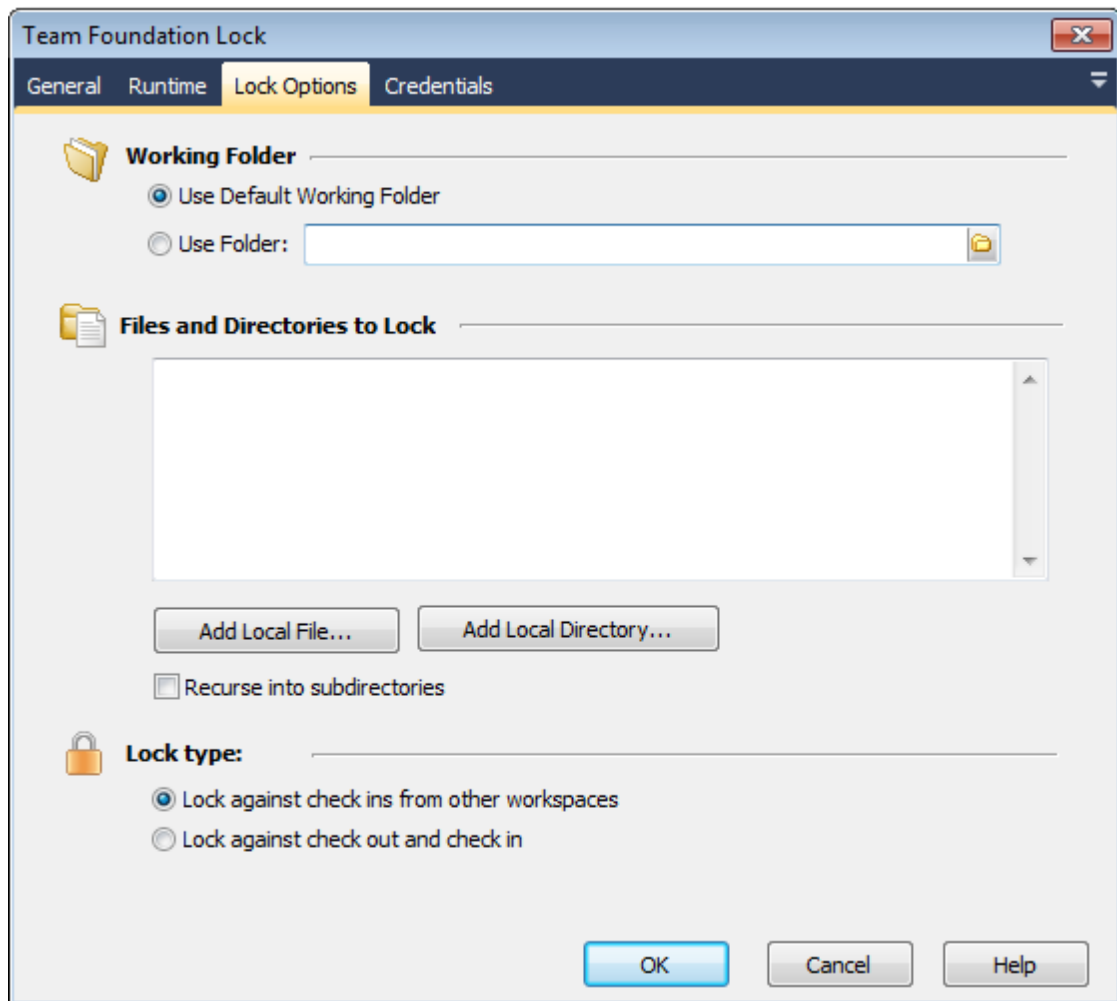
If this option is set, any writable files which are not checked out will be overwritten by the copies from the repository.

"Preview"

If this option is set, no files will be changed but the action will log the changes which would have been made.

6.34.19.1 Team Foundation Lock

Use the Team Foundation Lock Action to lock a set of files against check in or check out.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and Directories to Lock

Specify a list of files and directories to lock. Put each file/directory on a new line. Use wildcards (*) to specify groups of files.

Lock type

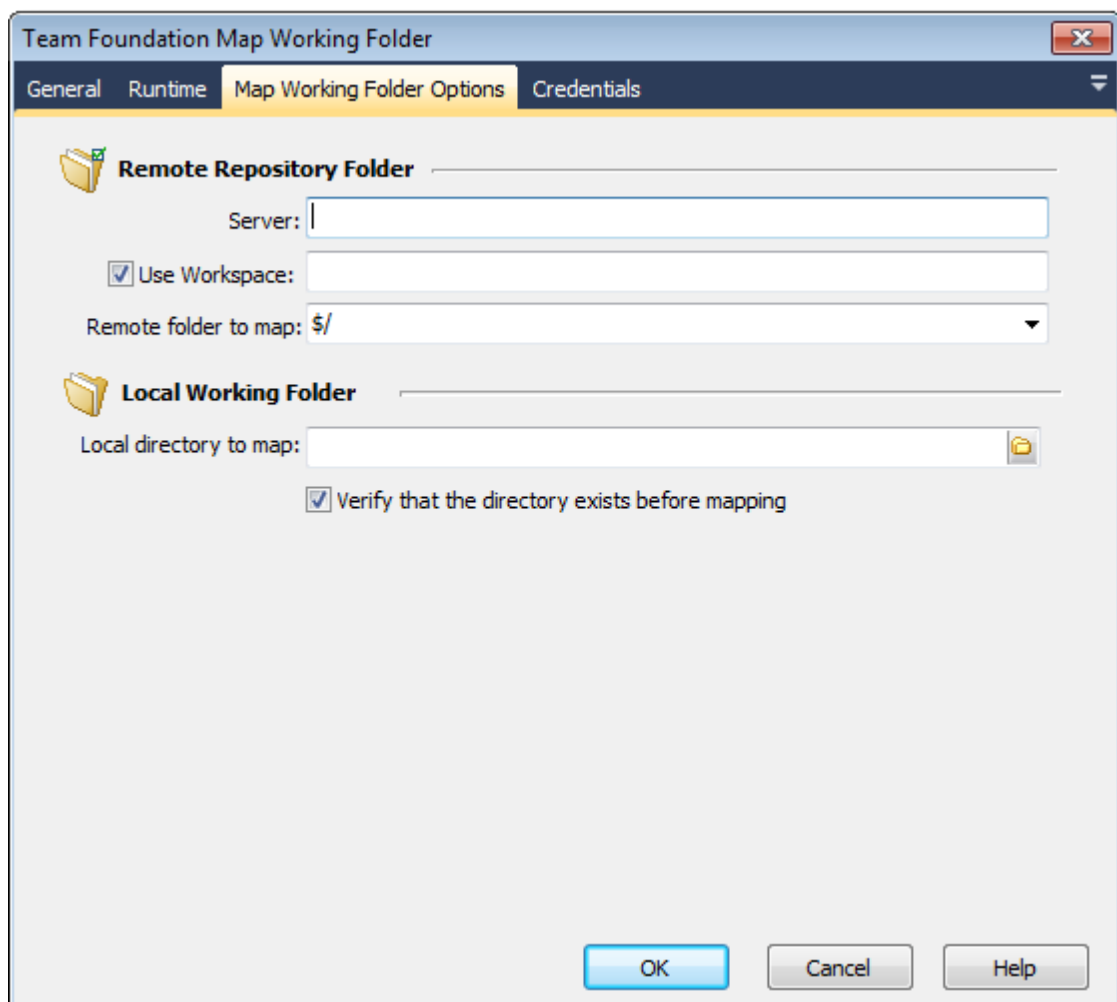
The action can be locked against checkins from other workspaces, or against both checkins and check outs.

6.34.19.1 Team Foundation Map Working Folder

Use the Map Working Folder action to define a new working folder. Working folders provide server, project, repository and workspace context to the other Team Foundation actions.

(Working folders can be configured in advance using Visual Studio, or as part of the build process with the Map Working Folder action.)

Note that you will need to use a Team Foundation Get Action to retrieve the contents of the working folder from the repository.



Remote Repository Folder

Specify the location of the repository to map. You can specify a server name, (optionally) a specific workspace, and also the folder inside that workspace.

Local Working Folder

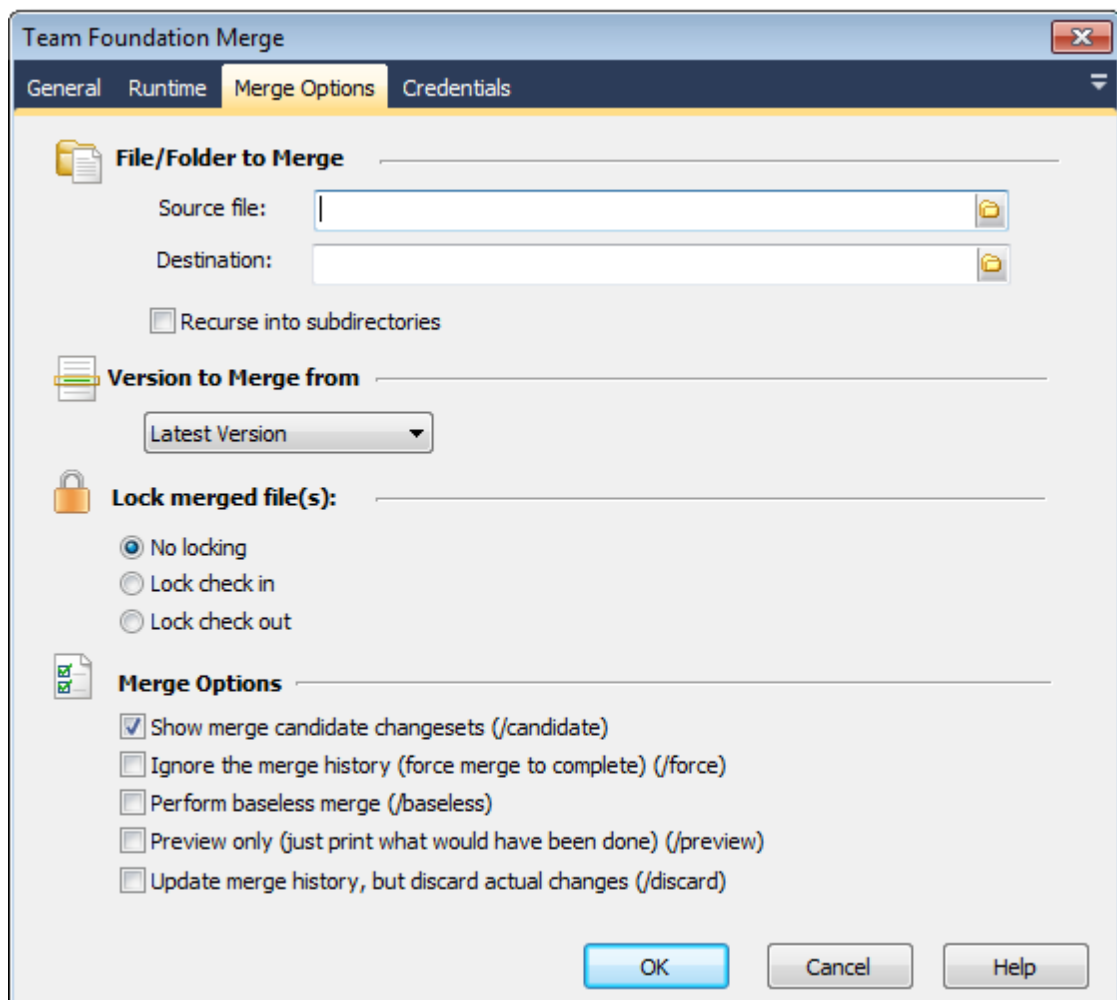
This is the local directory to map the working folder onto.

6.34.19.1 Team Foundation Merge

Use the Team Foundation Merge action to merge changes from another branch. The merge will not become permanent until a Check In Pending Changes operation takes place.

See the Merge Command reference in the Visual Studio Team Foundation Source Control Commands reference for more details on each option. Merges run by FinalBuilder always run with the "/noprompt" command line option set. Note that this can lead to unexpected behaviour.

Use the Team Foundation Merge action to merge changes from one branch into another.



File/Folder To Merge

Specify a branched file/folder to merge from, and a file/folder to merge to. Group multiple files with wildcards (ie *.*).

Version to Merge from

Instead of merging from the latest version of the file, you can choose to label a specific version. Versions can be selected by changeset number, label, a specific workspace, or a specific date.

Lock merged file(s)

Merged files can be locked against check in or check out by other users, until the changes are checked in.

Show merge candidate changesets (/candidate)

Use this option to show changesets in the source which have not been merged into the destination file/folder.

Ignore the merge history (/force)

Use this option to force the merge of all changes, even if some of them have been merged before.

Perform baseless merge (/baseless)

Use this option to perform a merge in absence of a base version.

Preview only (/preview)

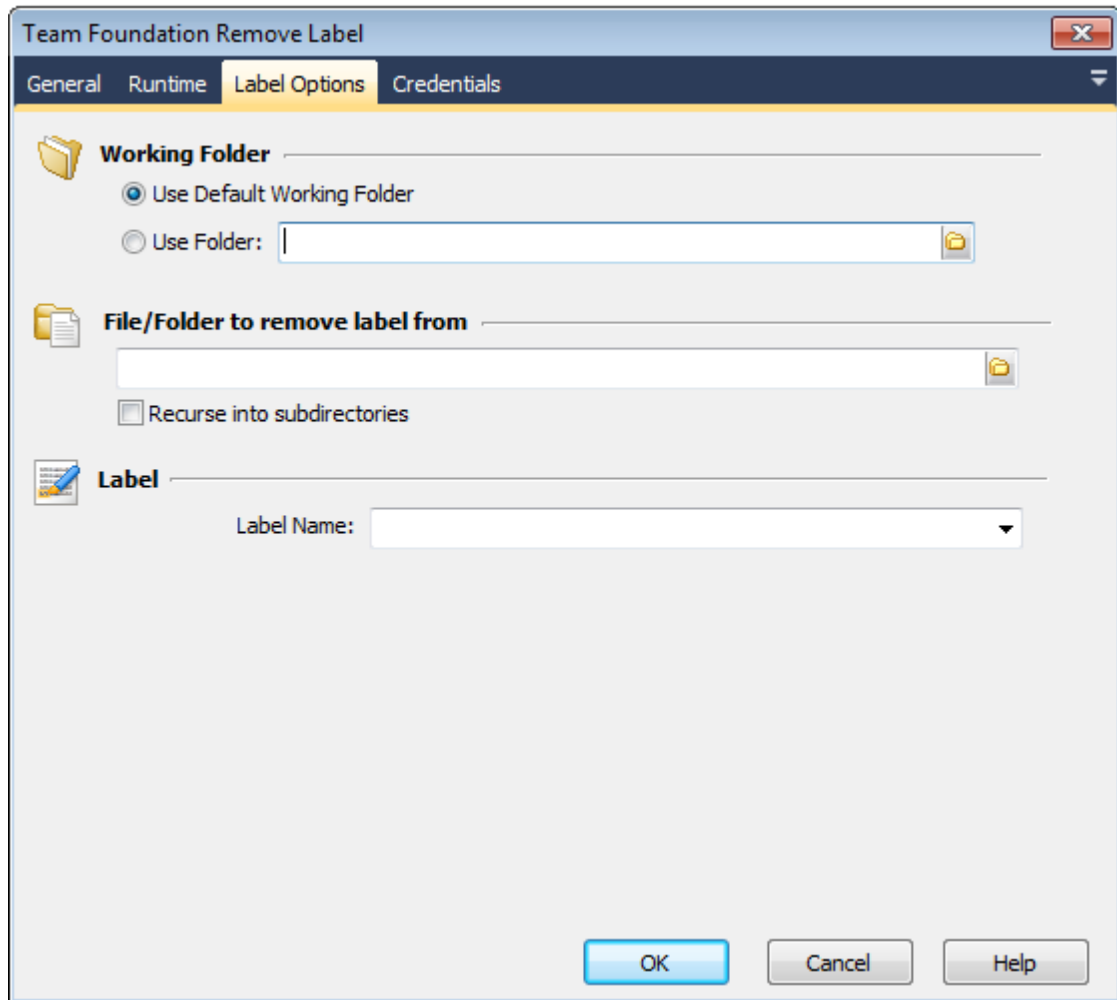
If this option is set, the merge will not actually happen but the action will log the changes which would have been made.

Update merge history, but discard actual changes (/discard)

If this option is set, Team Foundation will update the merge history but not invoke the actual changes.

6.34.19.1 Team Foundation Remove Label

Use the Team Foundation Remove Label action to remove a label from a file. You must have sufficient privileges to remove labels belonging to other users. The label will not be removed permanently until a Check In Pending Changes operation takes place.

**Working Folder**

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

File/Folder to remove label from

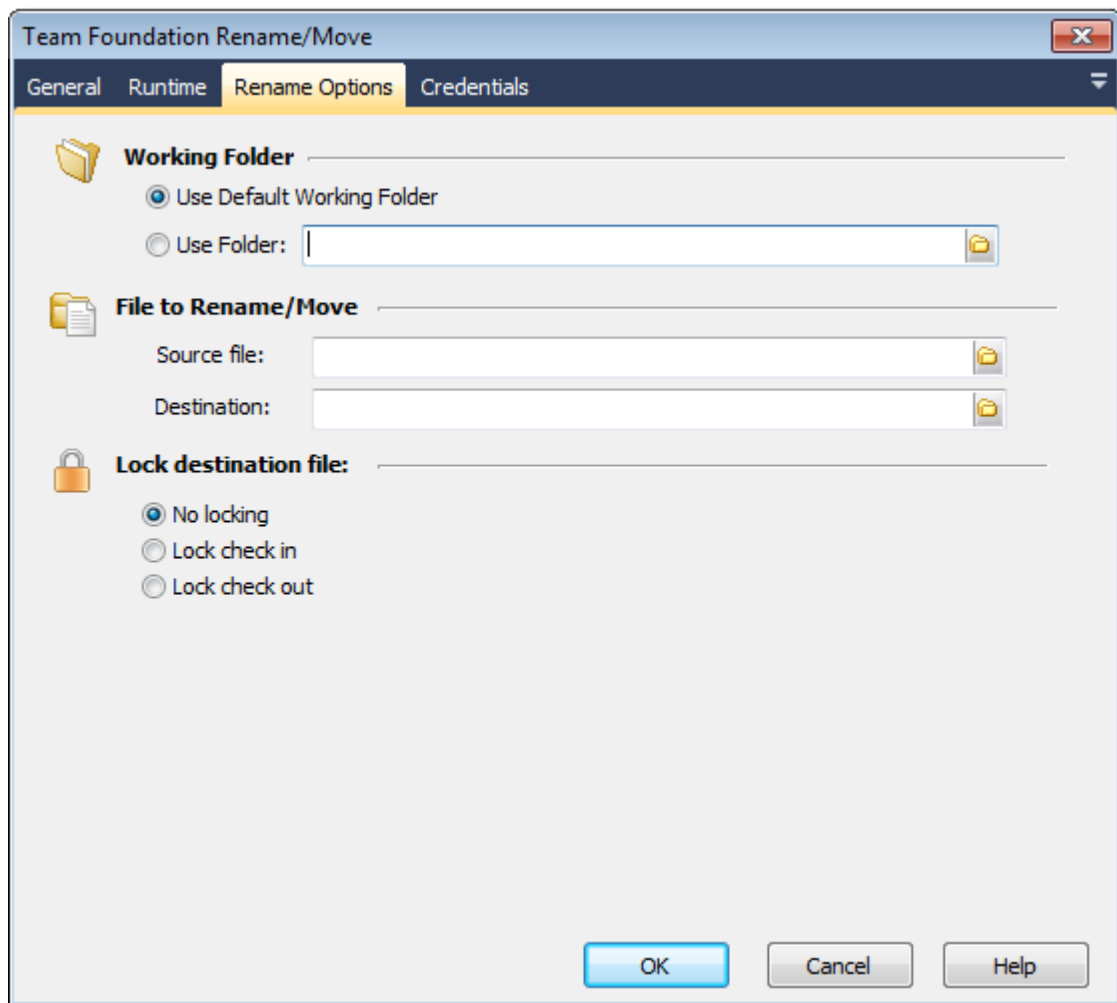
Use wildcards (ie *) to select groups of files.

Label

Give the name of the label to remove.

6.34.19.1 Team Foundation Rename/Move

Use the Team Foundation Rename/Move action to rename or move a file inside a repository. The new name/location will not become permanent until a Check In Pending Changes operation takes place.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

File to Rename/Move

Specify a file to rename/move, and a destination file. The destination can be a simple file name for rename, or a pathname for move.

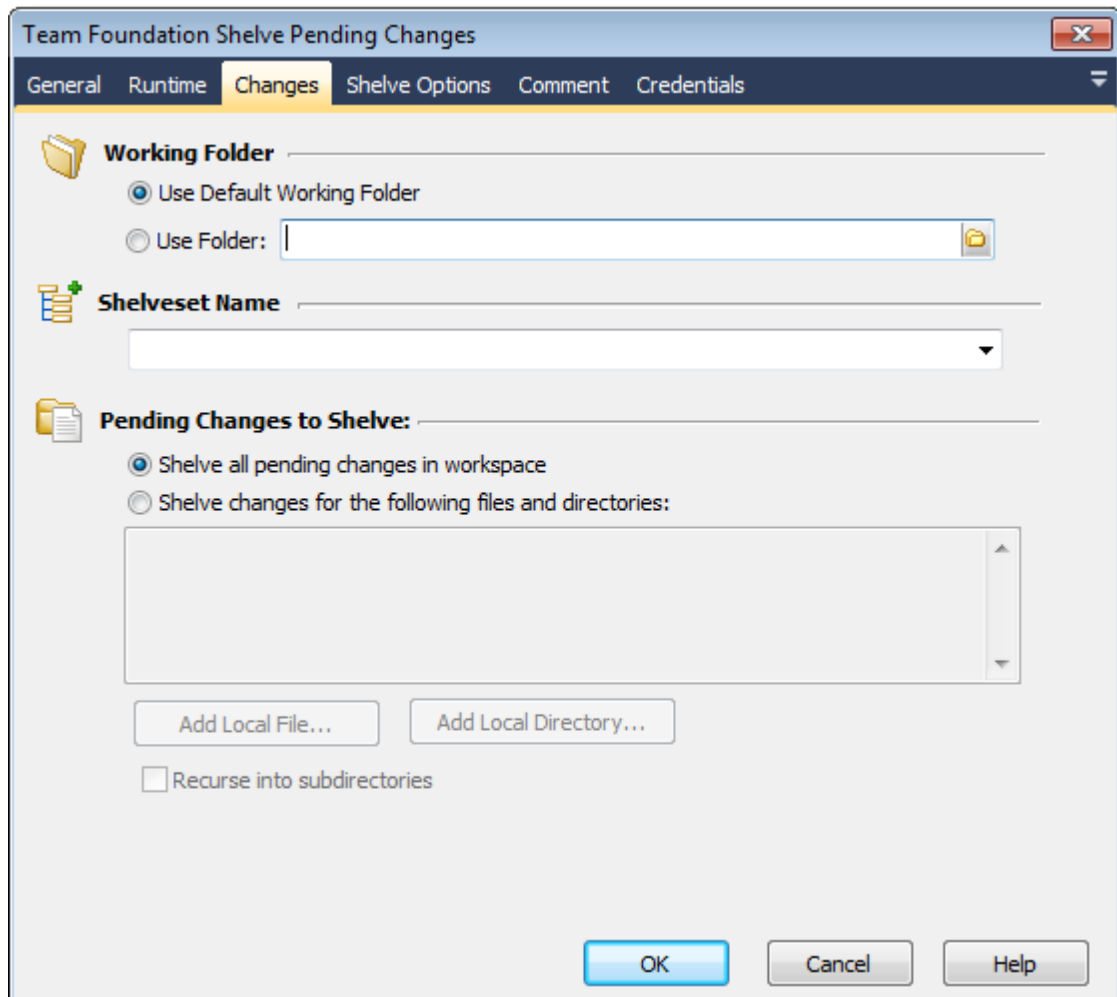
The destination must be inside the same working folder. To move a file between repositories, copy the file and then use the Team Foundation Add action.

Lock destination file

The destination file can be locked against check in or check out by other users.

6.34.19.1 Team Foundation Shelf Pending Changes

All changes made to a Team Foundation repository have to be checked in before they take effect. The "Shelve Pending Changes" action allows you to shelve a set of changes for later use. The shelved changes are called a Shelveset.



Working Folder

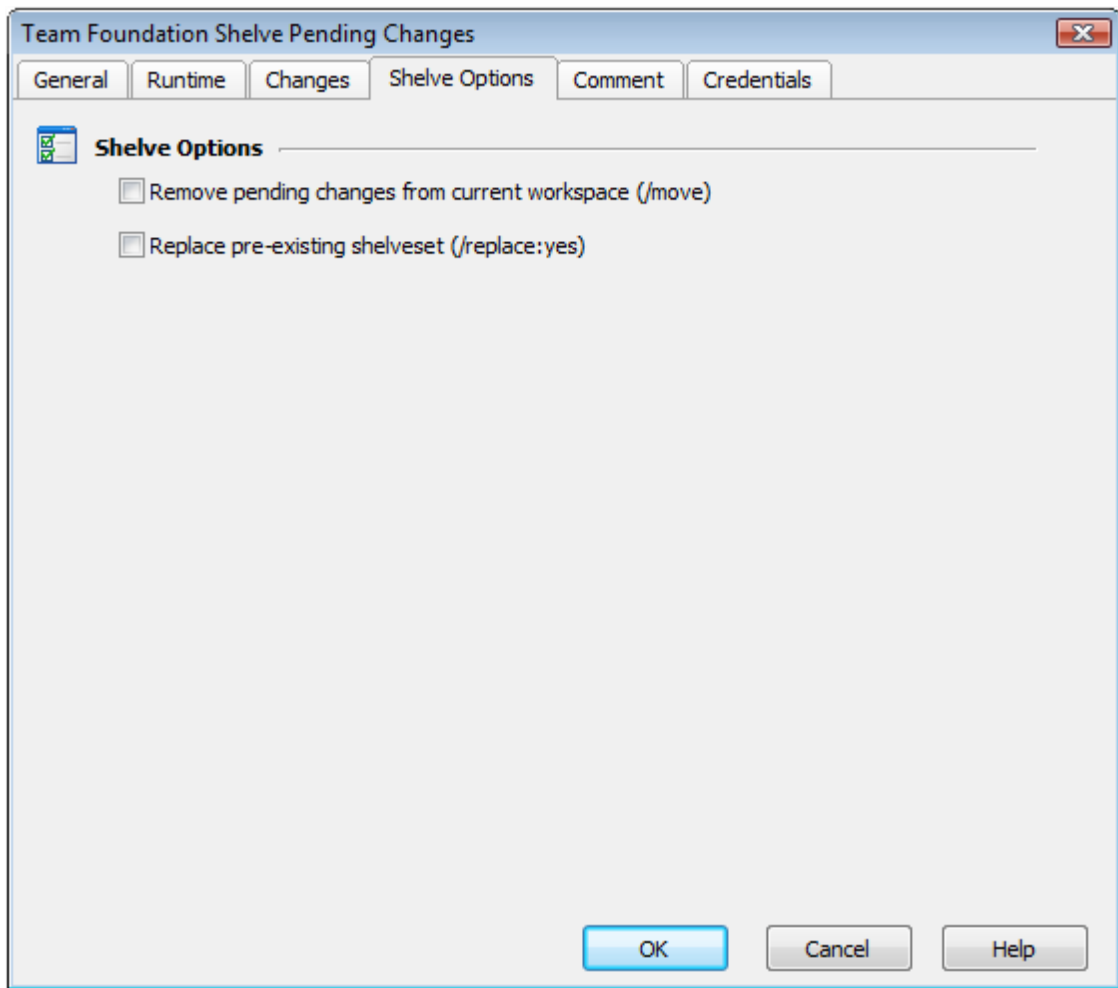
Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Shelveset Name

Specify a name for the shelveset. Click on the drop down to see a list of shelvesets which already exist in the working folder.

Pending Changes to Shelf

You can choose to shelve all pending changes for the workspace, or only changes that belong to certain files or directories. Put each file or directory on a new line.

**"Remove pending changes from current workspace"**

If this option is selected, the Shelve operation will remove the shelved changes from the current workspace. The files in the current workspace will be replaced with files from the repository.

"Replace pre-existing shelveset"

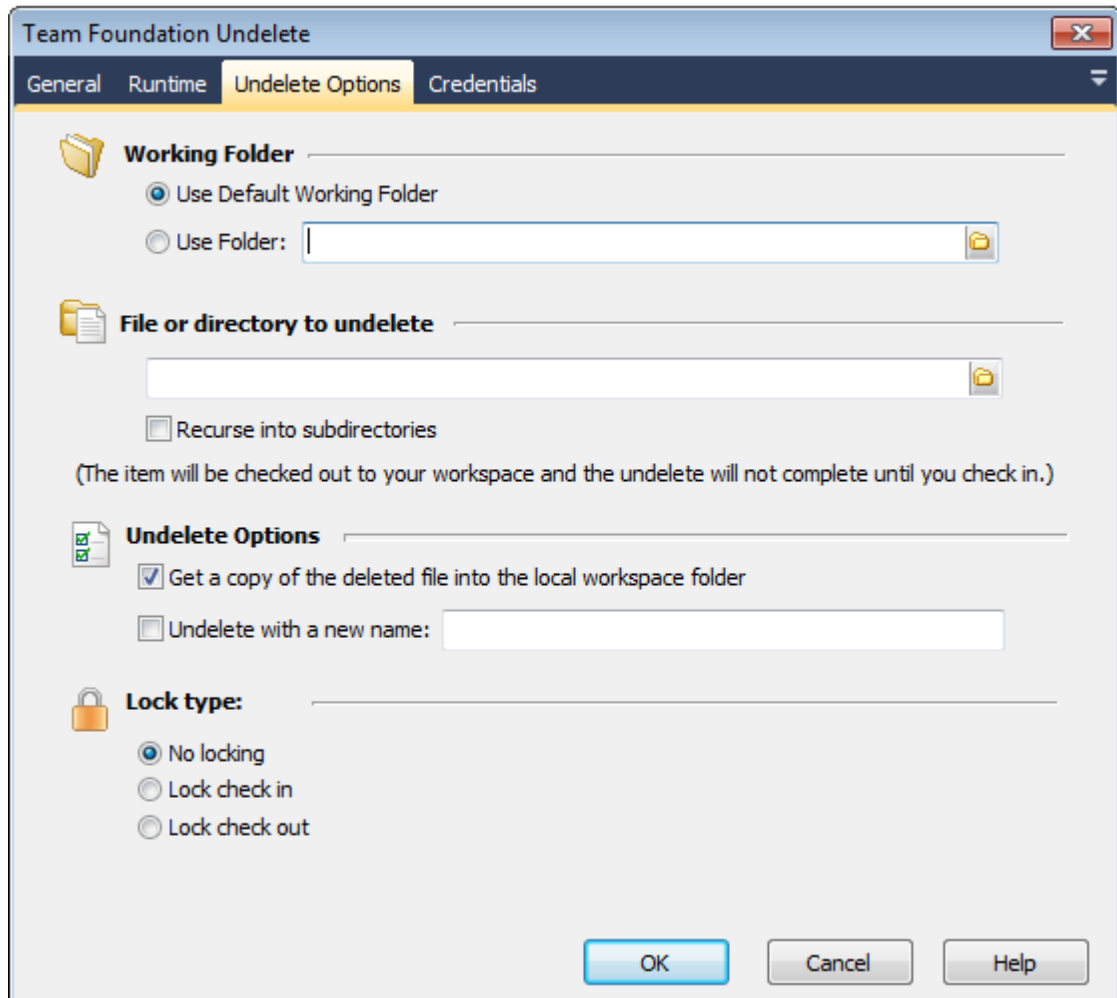
Check this option to replace a pre-existing shelveset with the same name. Note that the pre-existing shelveset will be lost.

Comment

Specify a comment for the shelved changes.

6.34.19.1 Team Foundation Undelete

Use the Undelete Action to undelete a previously deleted file. Note that, if the delete has not yet been checked in, you can undelete the file simply by using Undo Pending Changes followed by Get. The undelete will not become permanent until a Check In Pending Changes operation takes place.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

File or directory to undelete

Specify a file or directory to undelete. Specify multiple files with wildcards (ie *.*).

Undelete Options

"Get a copy of the deleted file into the local workspace folder" will automate a Get operation following the undelete.

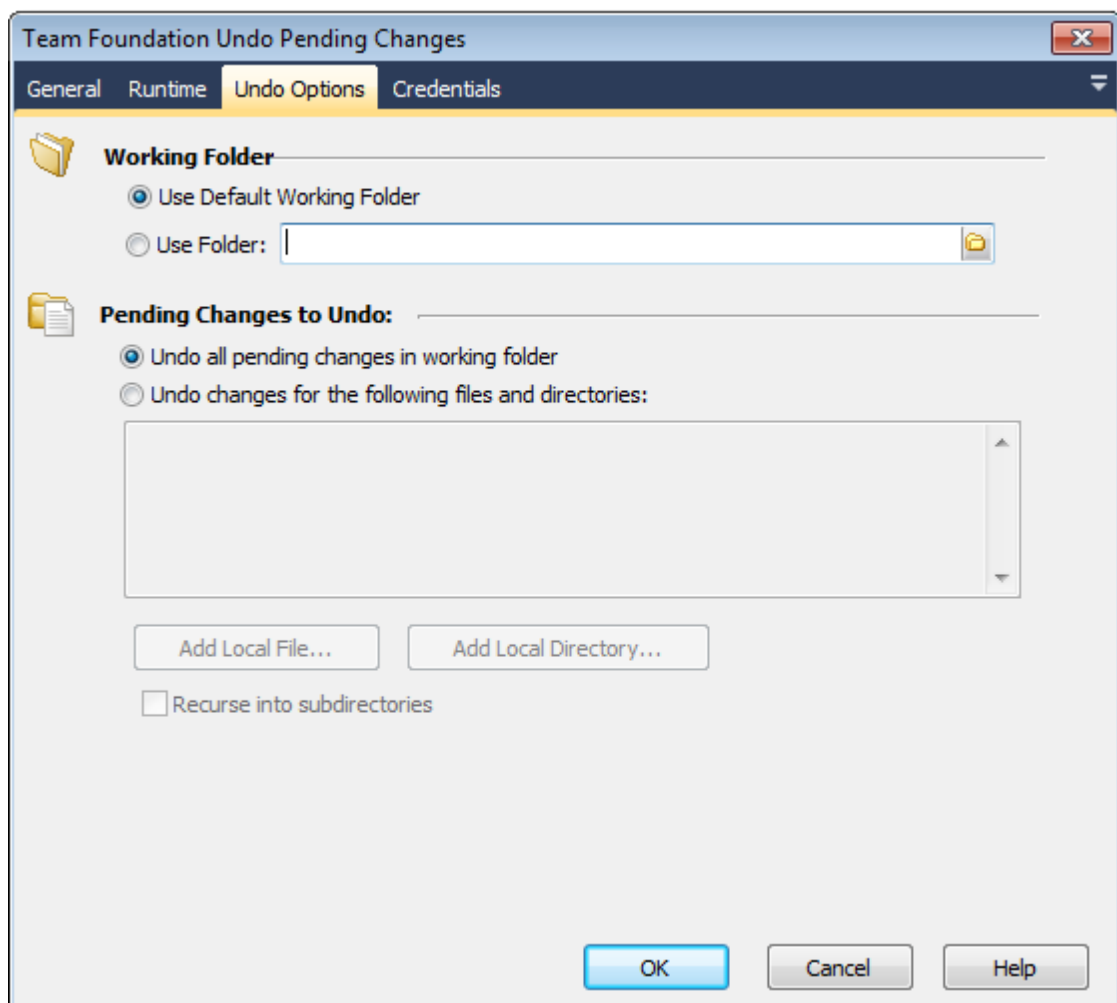
"Undelete with a new name" can give the newly undeleted file a new name.

Lock Type

The new file can be locked until a check in or unlock operation occurs.

6.34.19.17 Team Foundation Undo Pending Changes

The Undo Pending Changes action undoes a Team Foundation checkout and reverts all changed files to their pre-checkin state.



Working Folder

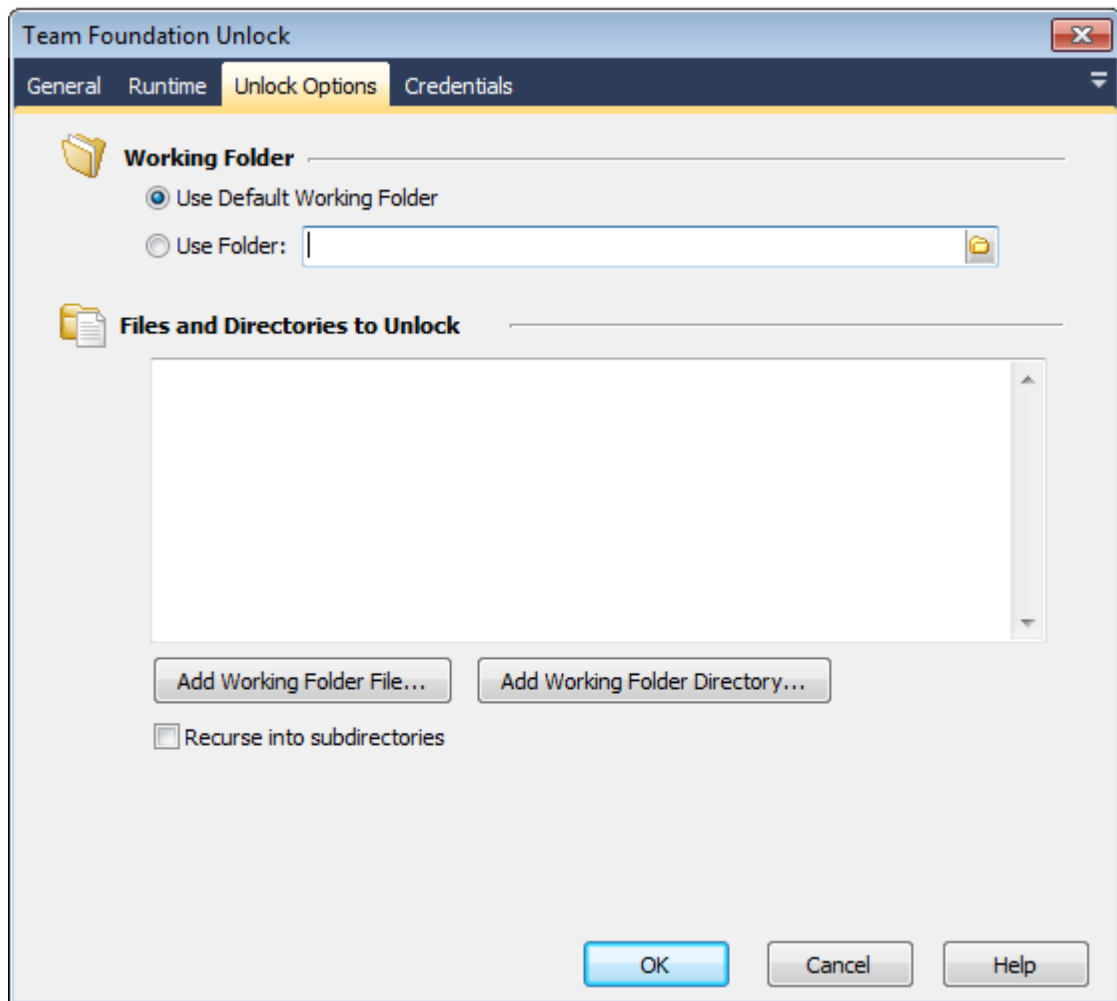
Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Pending Changes to Undo

You can choose to undo all pending changes in that working folder, or to undo changes made to specific files. If choosing specific files, put each new file/directory on a new line. Use wildcards (ie *) to select groups of files.

6.34.19.1 Team Foundation Unlock

Use the Team Foundation Unlock action to unlock previously locked files.



Working Folder

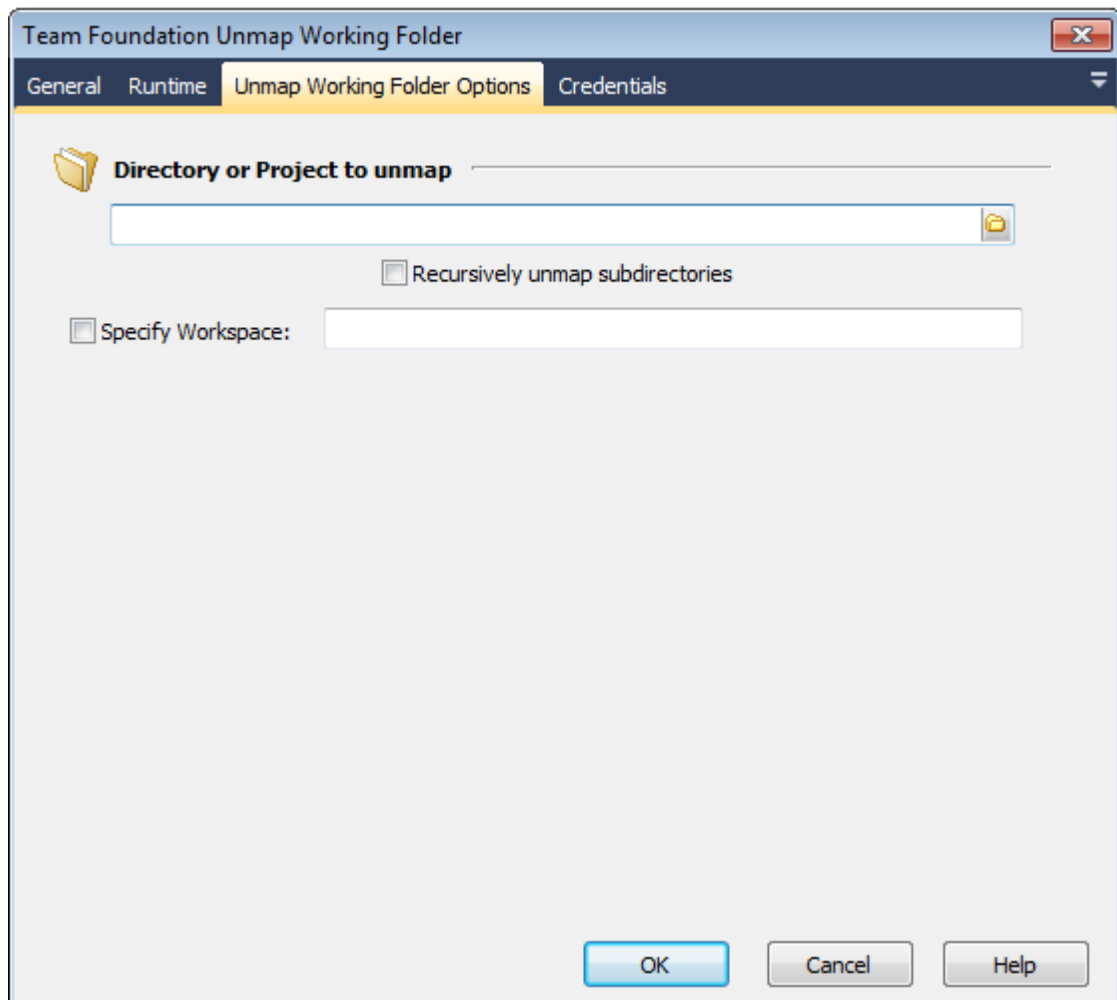
Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Files and Directories to Unlock

Specify a list of files and directories to unlock. Put each file/directory on a new line. Use wildcards (*) to specify groups of files.

6.34.19.1 Team Foundation Unmap Working Folder

Use the Unmap Working Folder action to unmap a previously mapped working folder (you can unmap folders which were mapped in either Visual Studio or via the Map Working Folder action.)

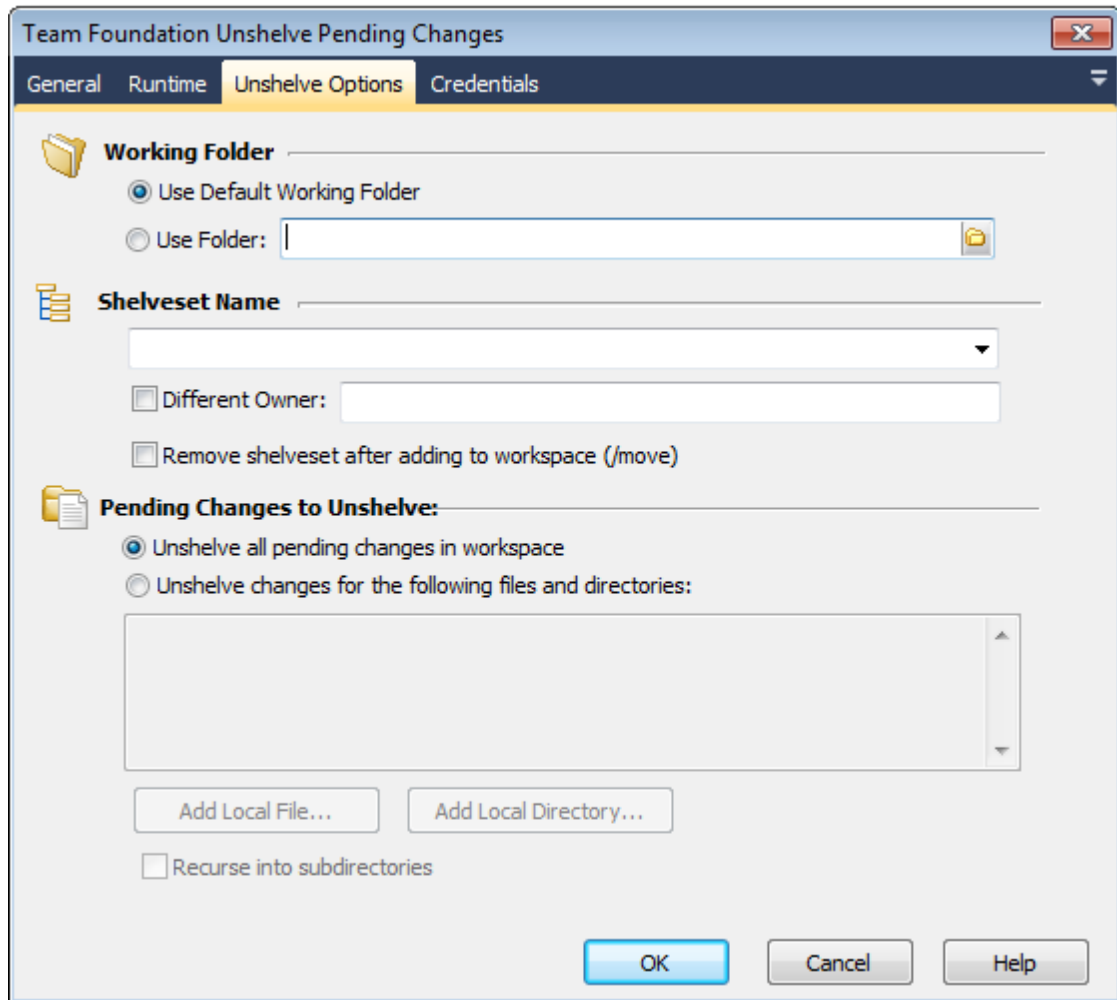


Local Working Folder

Specify the directory to unmap. Optionally, you can recursively unmap subdirectories which are also working folders.

6.34.19.2 Team Foundation Unshelve Pending Changes

Use the Unshelve Pending Changes action to unshelve a previously shelved changeset, bringing the changed files back into the current working folder. Note that unshelved changes still need to be checked in before the changes enter the repository.



Working Folder

Specify a working folder for the action. The working folder specifies the context for the Team Foundation server, project and repository (working folders can be configured in Visual Studio or with the Map Working Folder Action.) You can use the default working folder from the Options, or a specific folder for the action.

Shelveset Name

Specify the name of the shelveset to unshelve. Click on the dropdown to see a list of current shelveset names associated with the working folder.

"Different Owner"

Use this option to unshelve changes which were shelved by a different user.

"Remove shelveset after adding to workspace"

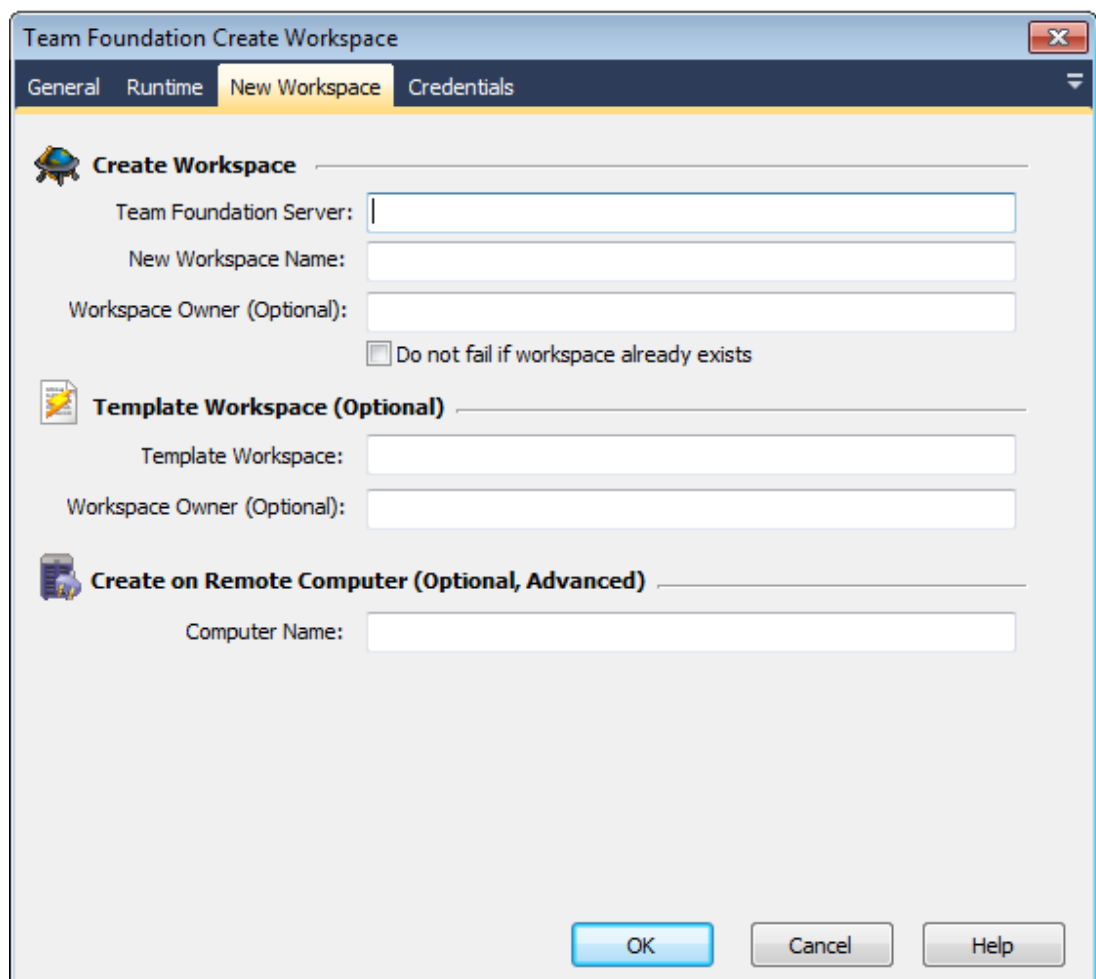
Check this option to remove the shelveset after the changes have been moved into the local workspace.

Pending Changes to Unshelve

You can choose to unshelve all changes associated with a shelve set, or only changes made to specific files. Specify each file or directory on a new line. Use wildcards to specify groups of files.

6.34.19.2 Team Foundation Create Workspace

The Team Foundation Create Workspace action allows you to create a new workspace for a repository on a specified Team Foundation Server. A workspace is a local copy of the files and folders on the server as well as the changes that you have made on locally.



The screenshot shows the 'Team Foundation Create Workspace' dialog box. It has four tabs: 'General', 'Runtime', 'New Workspace' (which is selected), and 'Credentials'. The 'New Workspace' tab contains three sections:

- Create Workspace**: Includes text boxes for 'Team Foundation Server:', 'New Workspace Name:', and 'Workspace Owner (Optional):'. There is a checkbox labeled 'Do not fail if workspace already exists'.
- Template Workspace (Optional)**: Includes text boxes for 'Template Workspace:' and 'Workspace Owner (Optional):'.
- Create on Remote Computer (Optional, Advanced)**: Includes a text box for 'Computer Name:'.

At the bottom right, there are three buttons: 'OK', 'Cancel', and 'Help'.

Create Workspace

Team Foundation Server

The Team Foundation Server which the workspace is being created for.

New Workspace Name

The name to give to the new workspace once it's created.

Workspace Owner

The username of the owner of the workspace, this field is only required when the workspace owner is not the person whose credentials are being used.

Template Workspace (Optional)

Template Workspace

The name of the existing workspace to use as a template, mappings of the existing workspace will be used in the new workspace.

Workspace Owner

The username of the owner of the template workspace.

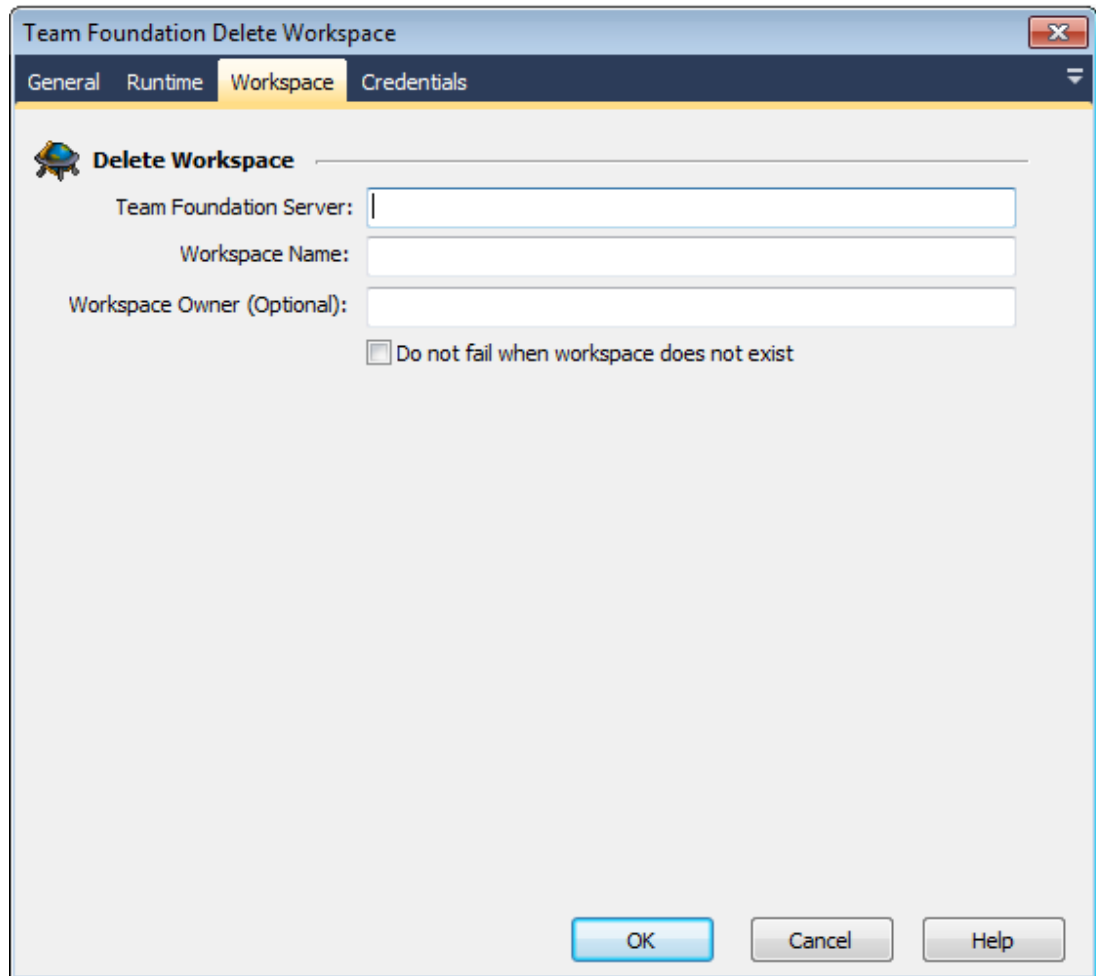
Create on Remote Computer

Computer Name

The name of the computer on which to create the workspace.

6.34.19.2 Team Foundation Delete Workspace

The Team Foundation Delete Workspace action allows you to delete an existing workspace on the local machine.



Delete Workspace

Team Foundation Server

The Team Foundation Server which the workspace belongs to.

New Workspace Name

The name of the workspace that is to be deleted.

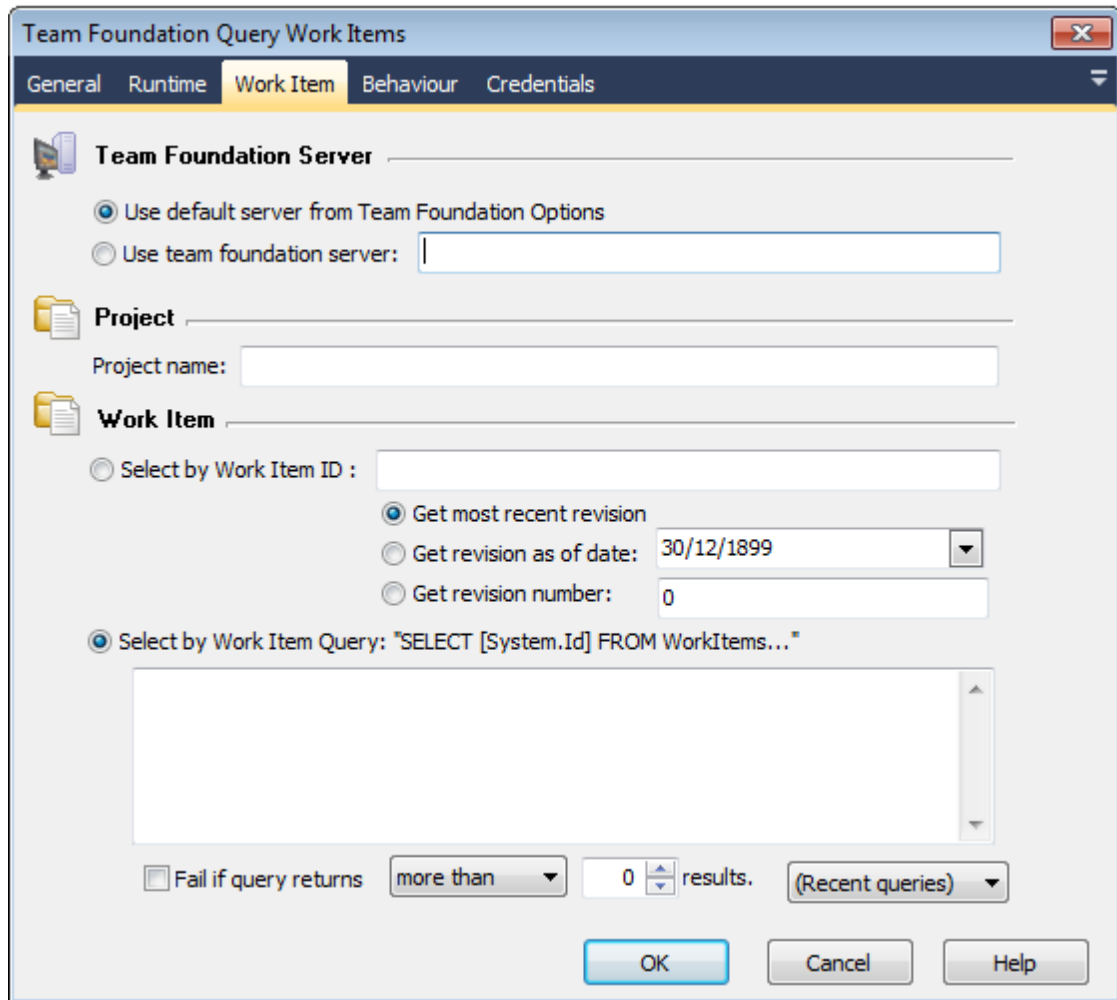
Workspace Owner

The username of the owner of the workspace, this field is only required when the workspace owner is not the person whose credentials are being used.

6.34.19.2 Team Foundation Work Items

The Team Foundation Work Item actions allow you to create, copy, edit and query work items from the Team Foundation work item database.

Work Item Property Page



Team Foundation Server

You can connect to the default Team Foundation server (specified under Options -> Other -> Team Foundation Work Items) or specify a specific server to connect to.

Project Name

Enter the name of the Team Foundation project to access.

Work Item

"Select by Work Item ID"

Choose this option to select a specific work item ID number. You can choose to select a specific revision of the work item, if you wish. Revisions can be chosen by date or revision number.

Note that if you are editing a Work Item, you can only select the most recent revision.

"Select by Work Item Query"

Select this option to enter a WQL Work Item Query. Queries can take one of two forms:

- Incomplete Queries. If you enter a query starting with "WHERE", then the text shown in the label (ie "SELECT [System.Id] FROM WorkItems") will be prepended to the full query string.
- Complete Queries. If you enter a complete, fully formed, query (ie "SELECT [System.Name] from Tasks WHERE [System.Description] contains 'Automated'") then no extra information will be added to the query string. Note that you may wish to SELECT specific properties if you are intending to edit or read them as part of the action (this will save both network traffic and time.)

"... Fail if query returns"

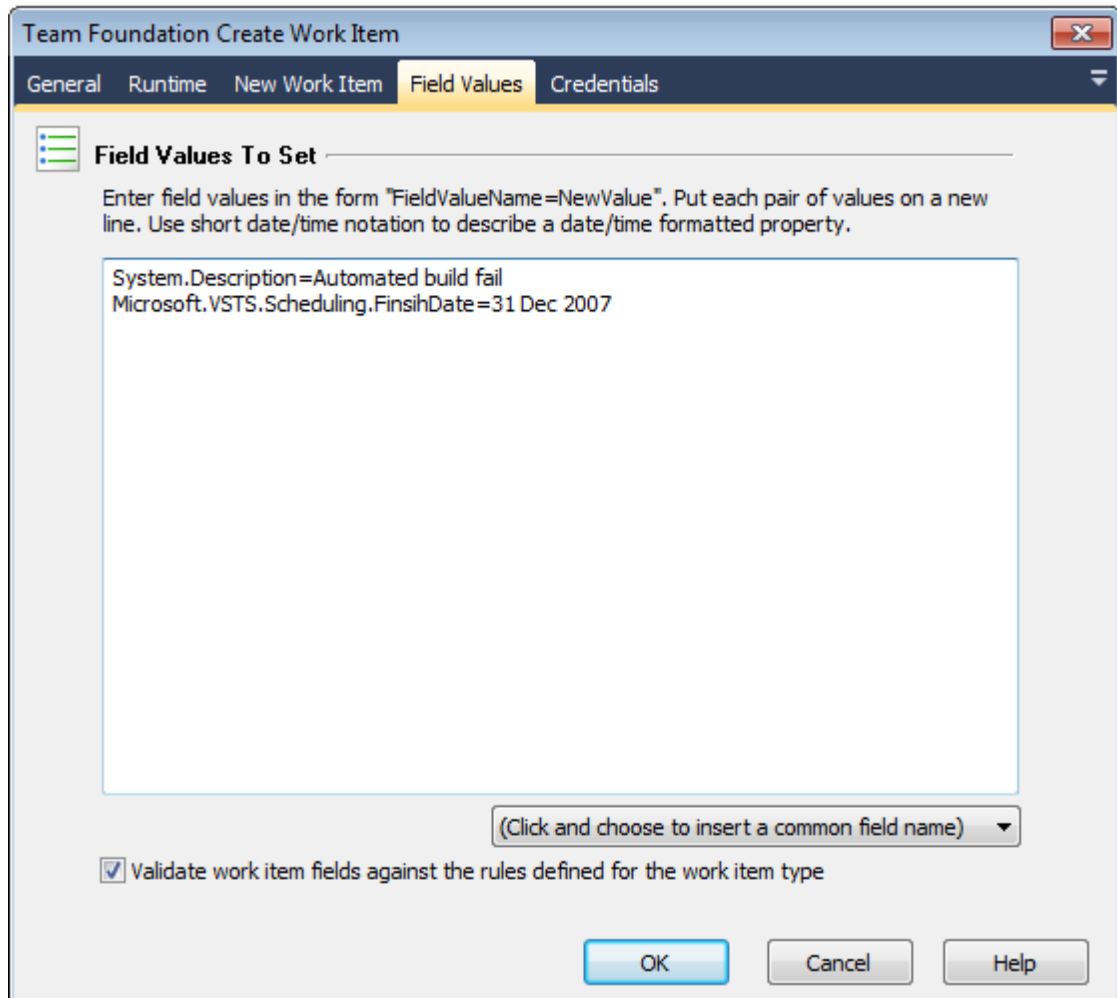
Optionally, you can choose to have the action fail if the Work Item Query returns the wrong number of results.

"(Recent queries)"

Click on this combo box to drop down a list of recent queries that have been used in FinalBuilder actions. Click on a query to insert it into the property page.

Field Values Property Page

The Field Values tab allows you to edit the values of one or more fields in selected work items:



Enter field=value pairs as shown above.

Properties must match the allowed formats for the field type (in the example above, the FinishDate property will be parsed to a .NET DateTime object before being assigned.)

"(Click and choose to insert a common field name)"

Click on this combo box to see a list of common field names. Select a field name to insert it into the field values list.

"Validate work item fields against the rules defined for the work item type"

If this option is not selected, field values will not be explicitly validated before they are committed to the database.

6.34.19.23.1 Team Foundation Query Work Items

The Team Foundation Query Work Items action allows you to get information for one or more Team Foundation work items. You can use the Work Item Query Language to make the query, if needed.

Work Item Property Page

See the Team Foundation Work Items overview topic for details.

Behaviour Property Page

Team Foundation Query Work Items

General Runtime Work Item **Behaviour** Credentials

Logging

☐ Don't log work item details

☐ Log summaries (ID, State, Title)

☐ Log full details

☐ Log values of these fields:

(Separate multiple field names with commas)

☒ Write formatted output to a variable:

Variable name:

Output format for each work item:

```
<BR>
<I>{System.Id}</I> <B>{System.Title}</B>({System.State})
<P>
{System.Description}
</P>
```

(Use {FieldName} to specify a field value. See Help topic for details.)

OK Cancel Help

"Don't log work item details"

Select this option if you do not want to log any details about the selected Work Items.

Log summaries (ID, State, Title)

The values of the **System.ID**, **System.State** and **System.Title** fields will be written to the log.

Log full details

The values of all Work Item fields will be written to the log. Note that this can be a slow process.

Log values of these fields

Enter the names of fields to log. Separate multiple fields with commas.

Write formatted output to a variable

Use this option to write out the work item data in a format which can then be included in a formatted document such as an HTML, XML or CSV file.

The example in the screenshot above will create a simple fragment of HTML describing each work item which is returned by the query, and write the entire fragment to the variable "workItemData". This HTML could then be inserted into an HTML file, for instance by the Write To Text File action.

To specify field values as part of the formatted output, use curly braces - ie {System.Description}.

To insert a regular opening curly brace into the formatted output, escape it as {{.

6.34.19.23.2 Team Foundation Create Work Item

Use the Team Foundation Create Work Item action to create a new work item in the Team Foundation server's database.

Team Foundation Create Work Item

General Runtime **New Work Item** Field Values Credentials

Team Foundation Server

☒ Use default server from Team Foundation Options

☐ Use team foundation server:

Project

Project name:

New Item

Work item type:

Work item title:

☐ Fail if a work item matching this title and type already exists

☐ Edit existing fields if a work item matching this title and type already exists

Work Item Id

☐ Save the new Work Item Id to a variable

OK Cancel Help

Team Foundation Server

You can connect to the default Team Foundation server (specified under Options -> Other -> Team Foundation Work Items) or specify a specific server to connect to.

Project

Specify the name of the project in which to create the new Work Item.

New Item

Specify a work item type, and the title of the new item. The drop-down menu displays Team Foundation built in types. To enter a custom work item type, type it into the combo box field.

"Fail if a work item matching this title and type already exists..."

The action can be set to fail if a work item matching it's description already exists.

"Edit existing fields if a work item matching this title and type already exists..."

If this checkbox is set then the action will edit the existing work item instead of creating a duplicate. Incompatible with the "Fail if a work item matching this title and type..." option.

"Save the new Work Item Id to a variable..."

Optionally, choose a FinalBuilder variable name to write the Work Item ID.

Field Values Property Page

See the Team Foundation Work Item overview topic.

6.34.19.23.3 Team Foundation Edit Work Item(s)

The Team Foundation Edit Work Item(s) action allows you to edit the field values of one or more work items.

Work Item Property Page

See the Team Foundation Work Items overview topic.

Field Values Property Page

See the Team Foundation Work Items overview topic.

6.34.19.23.4 Team Foundation Copy Work Item(s)

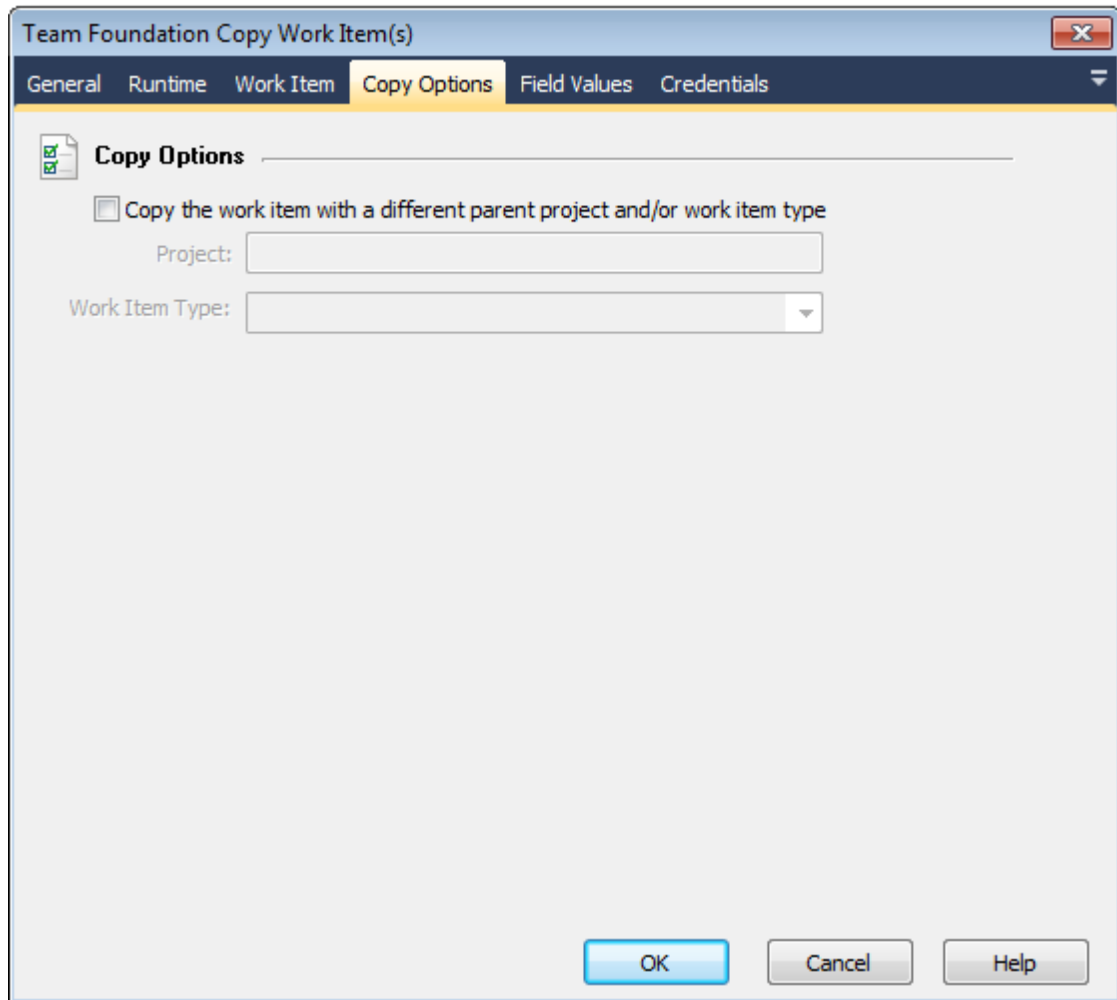
The Team Foundation Copy Work Item(s) action allows you to create duplicates of one or more Work Items. You can specify fields to edit in the copied item(s).

Work Item Property Page

Use this tab to select the Work Item(s) to copy. See the Team Foundation Work Items overview topic.

Field Values Property Page

Use this tab to select values to write into fields of the duplicate work item(s). See the Team Foundation Work Items overview topic.



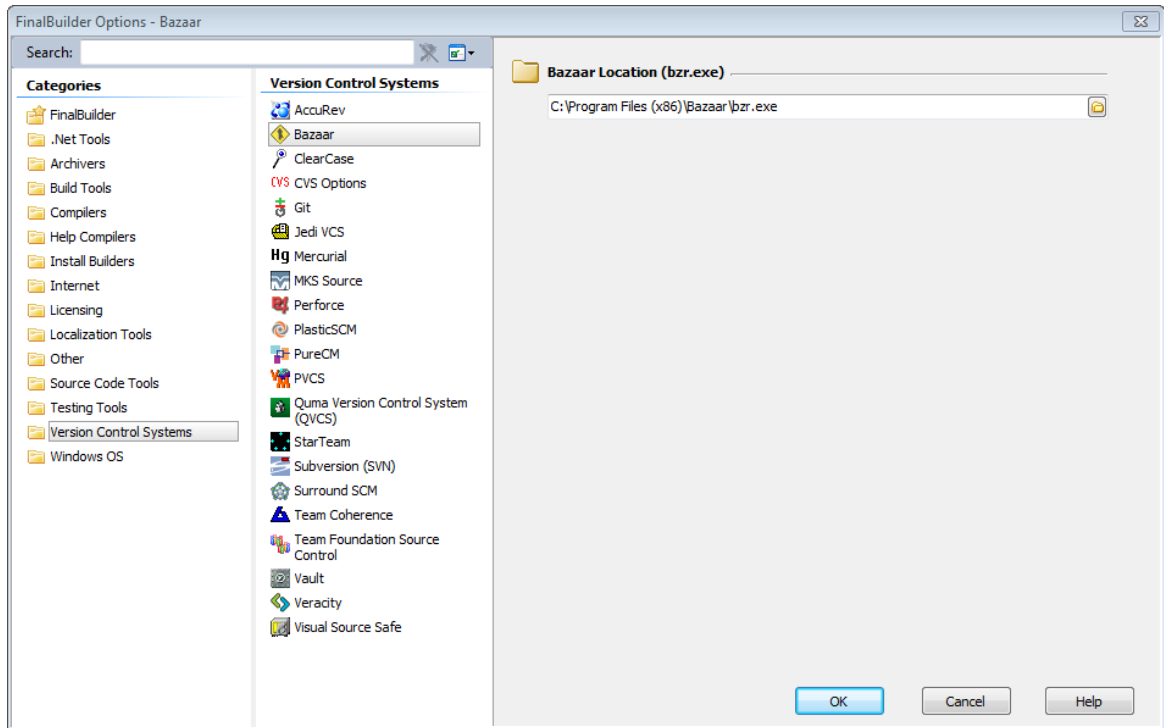
"Copy the work item with a different parent project and/or work item type"

Use this option to copy a Work Item between projects, and/or to convert a copied work item to a different type. If this box is checked, you will need to enter both the Project and Work Item Type into the fields below.

6.34.20 Bazaar

Bazaar is a version control system that helps you track project history over time and to collaborate easily with others. More information about Bazaar: <http://bazaar.canonical.com>

You must set the location of bzt.exe in the Bazaar options page (Tools > Options > Version Control Systems > Bazaar) before using any of the Bazaar actions.

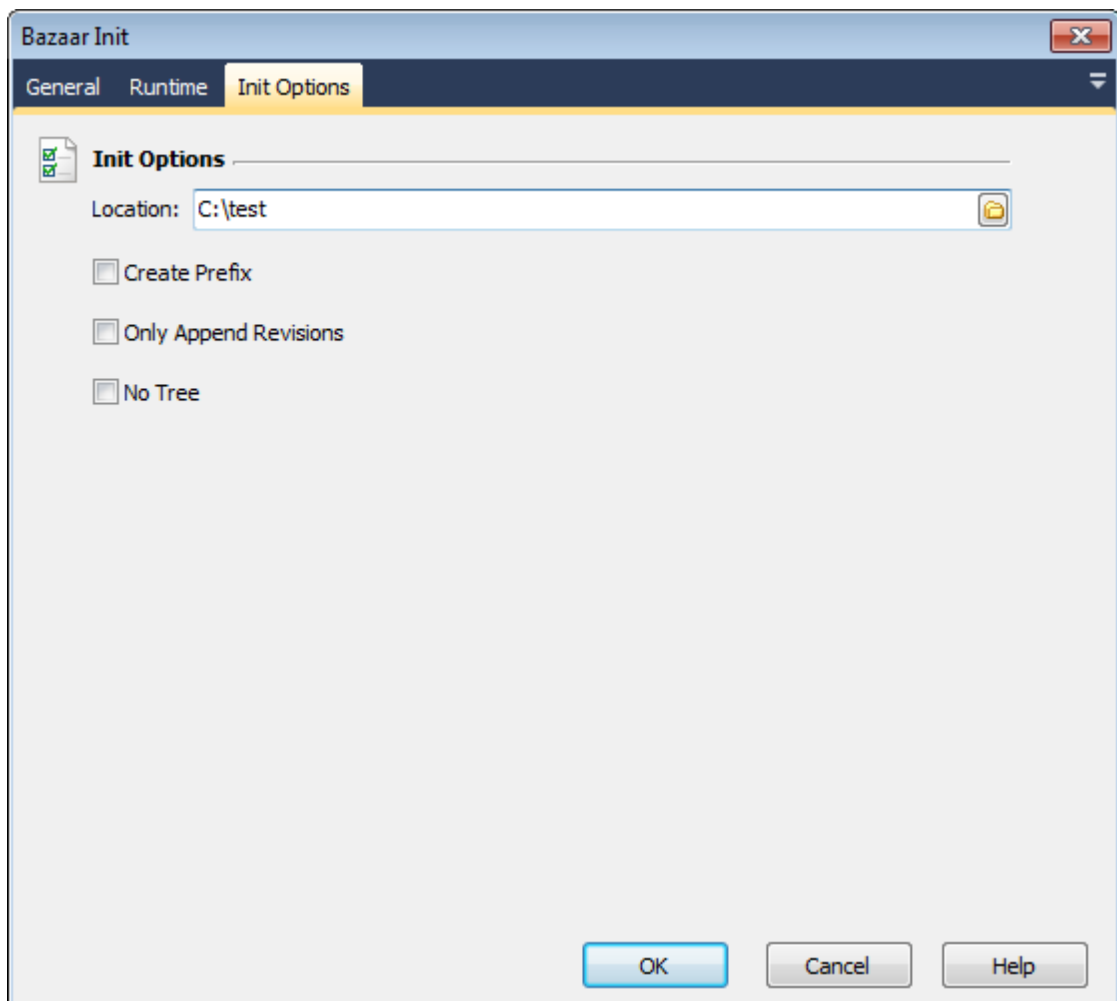


Bazaar Location

The location of the Bazaar command-line executable (Usually found in C:\Program Files\Bazaar\bzt.exe).

6.34.20.1 Bazaar Init

This action enables you to make a directory into a versioned branch.

**Location**

Location of the directory to initiate as a versioned branch.

Create Prefix

Create the path leading up to the branch, if it does not already exist.

Only Append Revisions

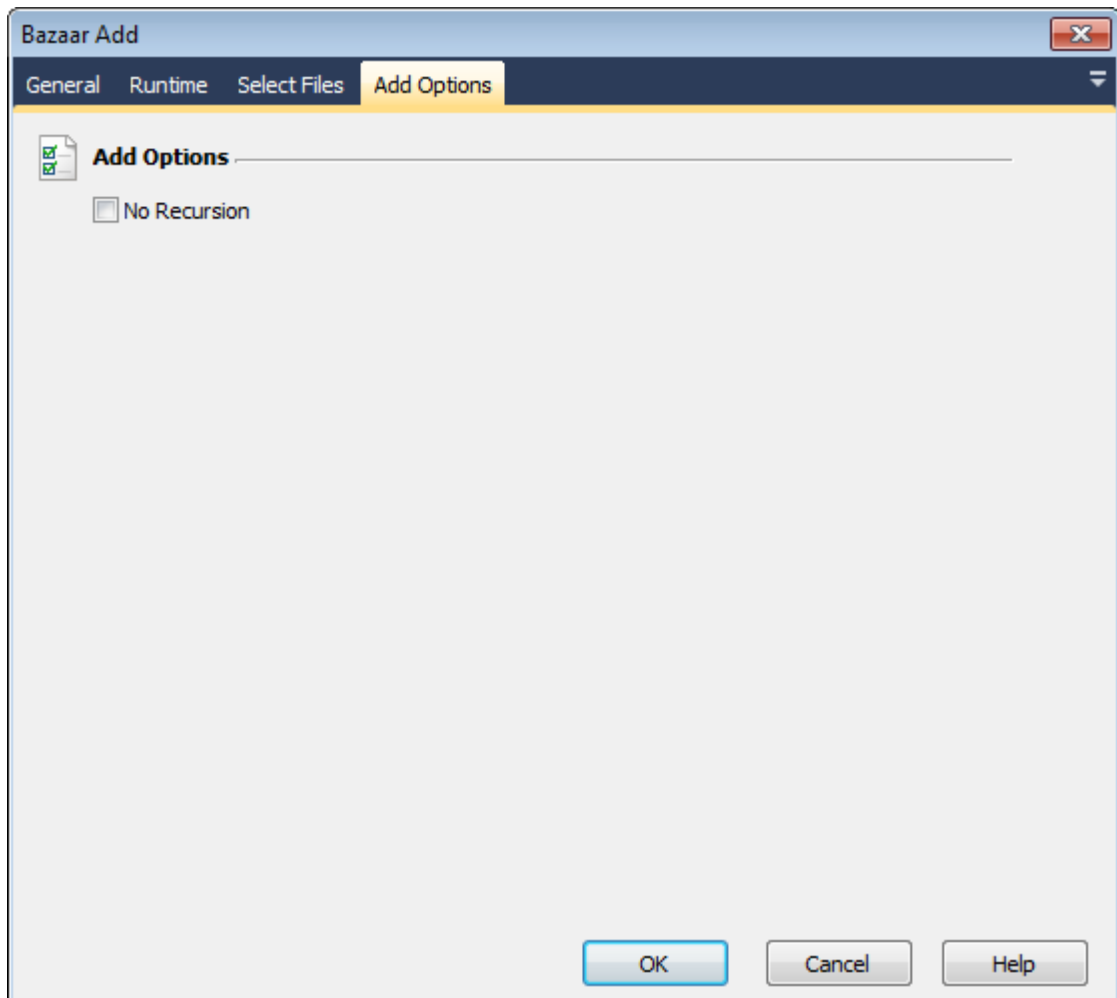
Never change revision numbers, or the existing log. Append revisions to it only.

No Tree

Create a branch without a working tree.

6.34.20.2 Bazaar Add

This action enables you to add specified files and/or directories to be tracked by the branch.



Select Files

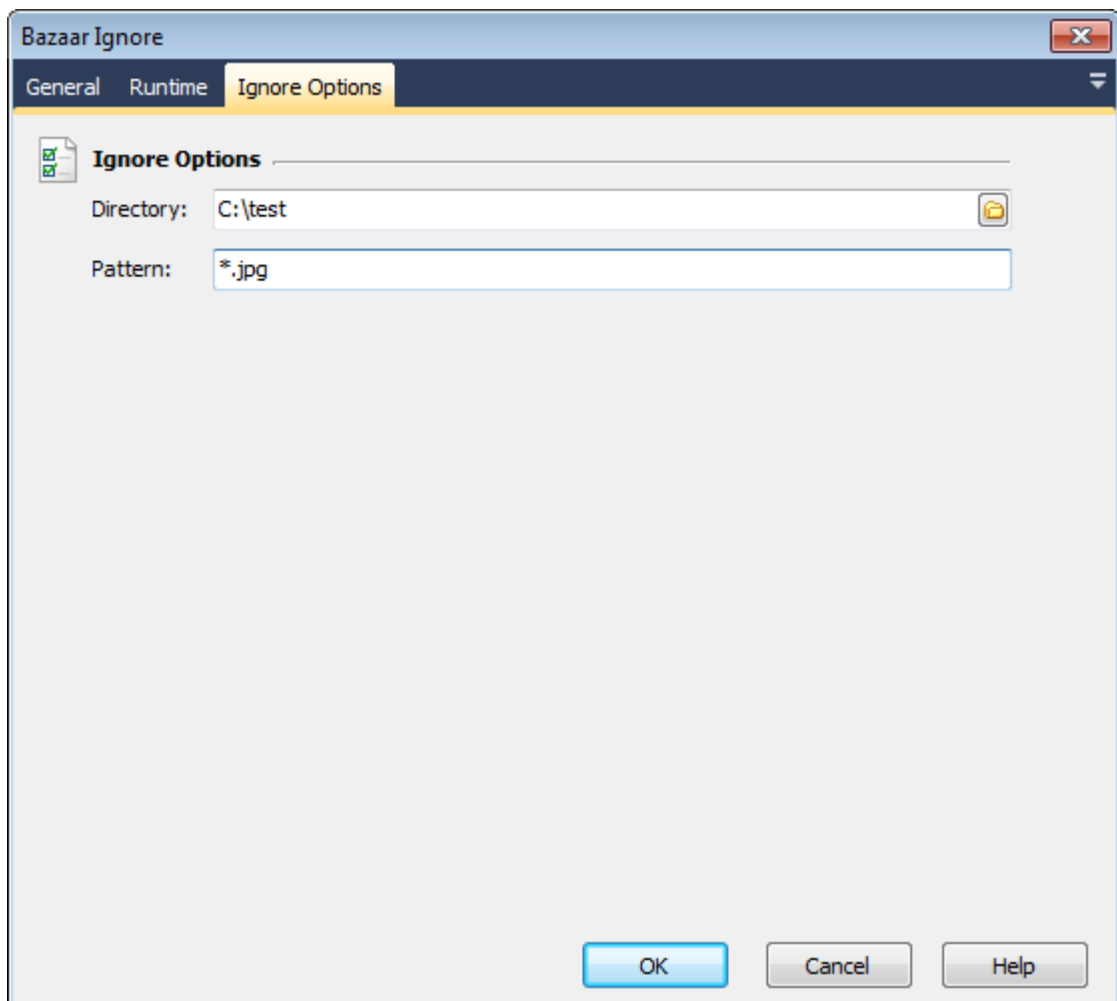
See Select Files.

No Recursion

Don't recursively add the contents of directories.

6.34.20.3 Bazaar Ignore

This action enables you to add pattern to the ignore list for the specified directory.

**Directory**

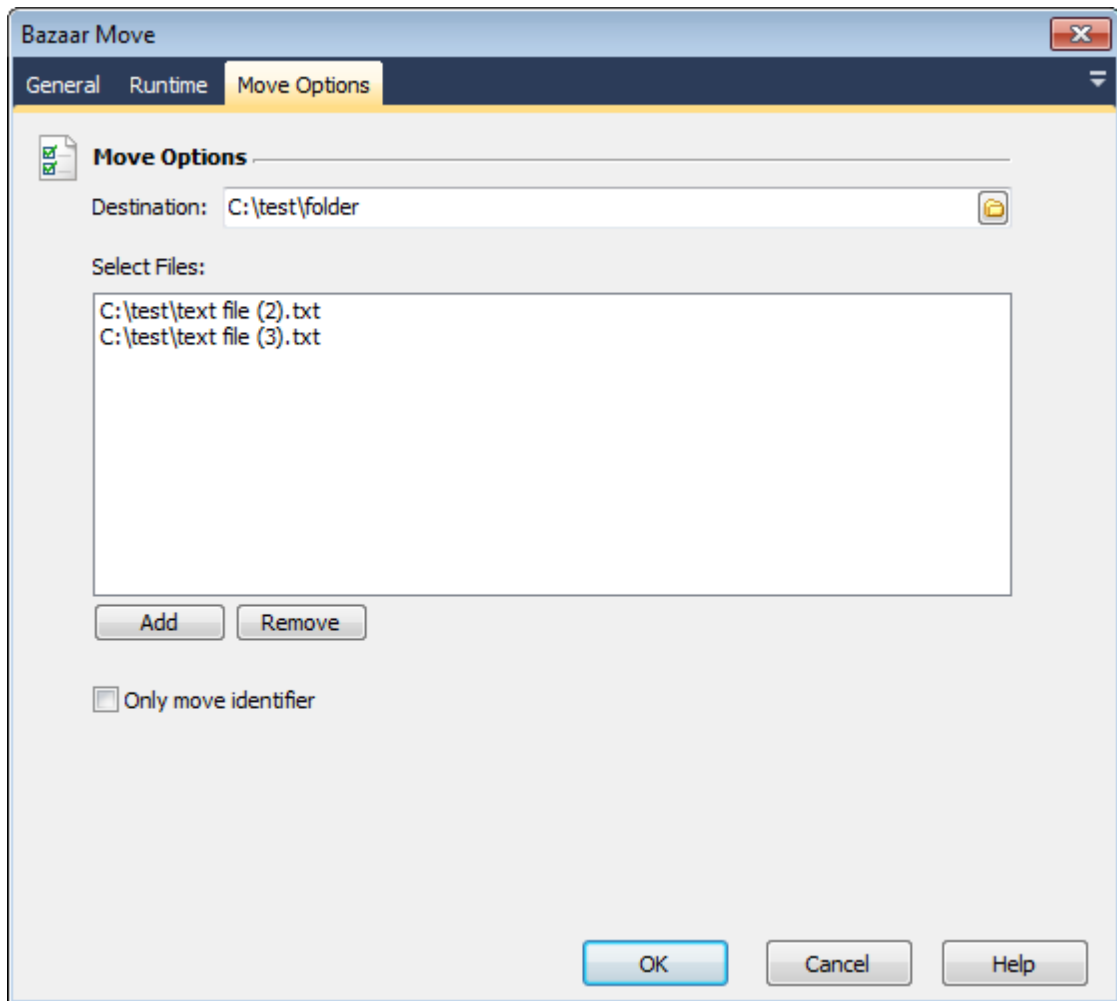
The directory to add the ignore pattern to.

Pattern

The pattern to add. Additional documentation on Bazaar patterns: <http://doc.bazaar.canonical.com/latest/en/user-reference/patterns-help.html>.

6.34.20.4 Bazaar Move

This action enables you to move a specific files or folders to another directory.

**Destination**

The directory to move the selected files to.

Select Files

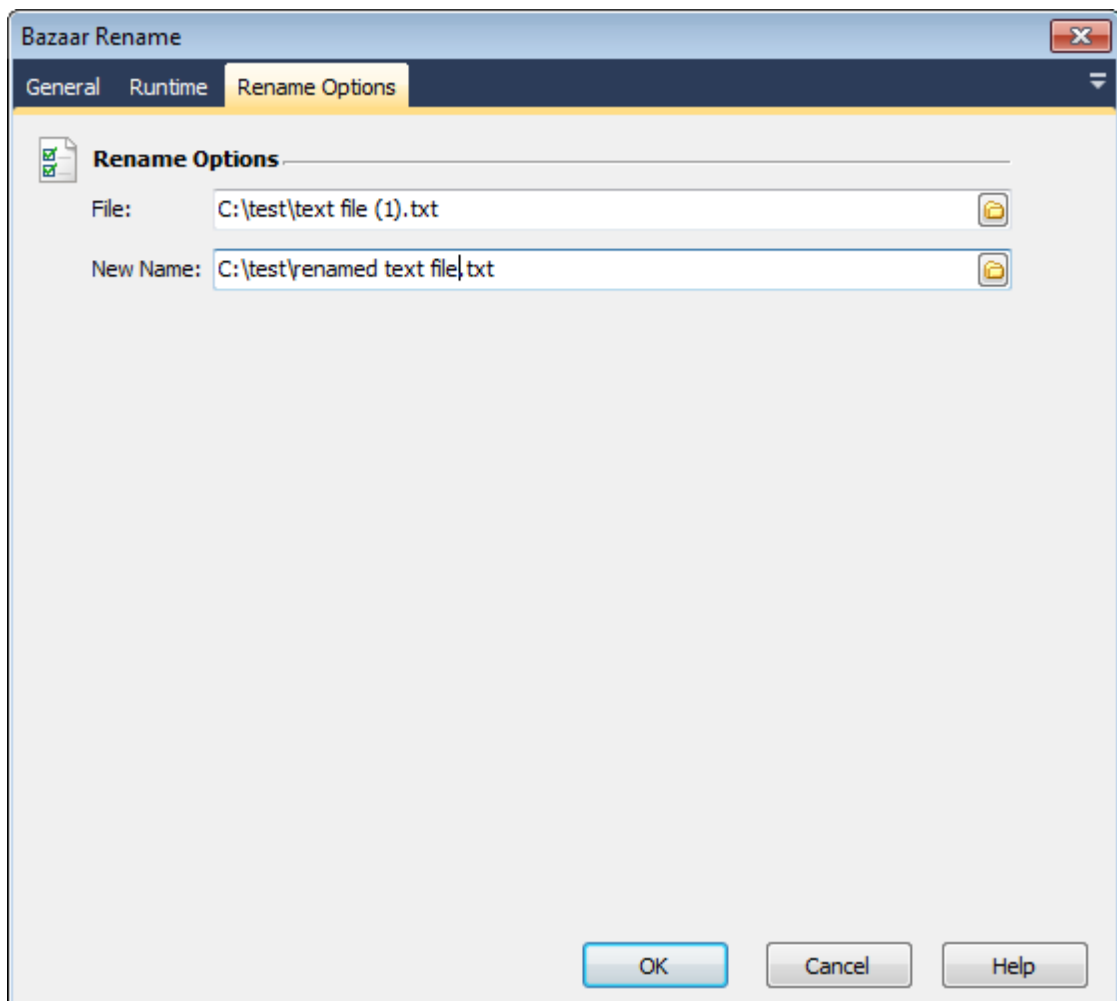
Select the files to be moved.

Only Move Identifier

Move only the bazaar identifier of the file, for if the file has already been moved.

6.34.20.5 Bazaar Rename

This action enables you to rename a specific file.

**File**

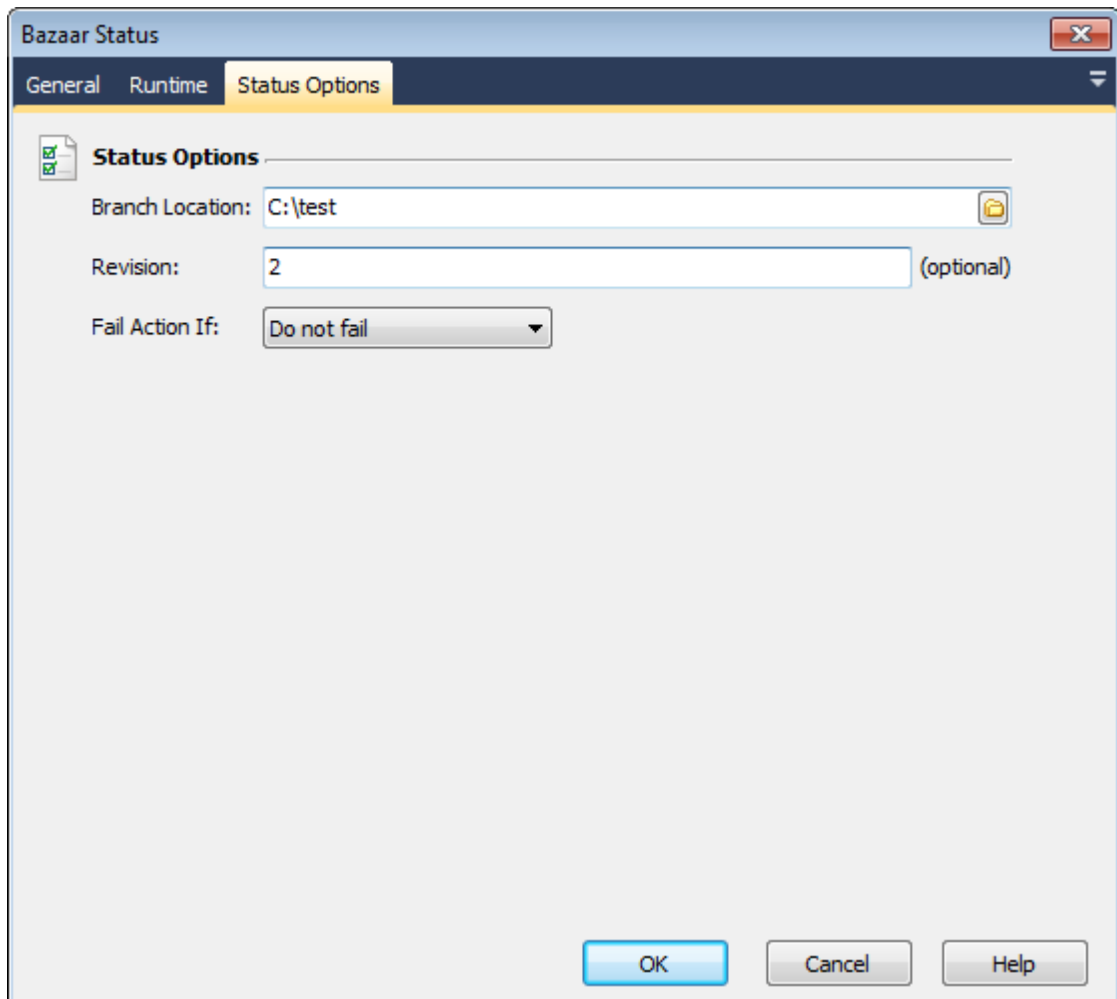
File to rename.

New Name

New name for the file.

6.34.20.6 Bazaar Status

This action enables you to check the status of the specified directory.



Branch Location

Location of the branch to get the status of.

Revision

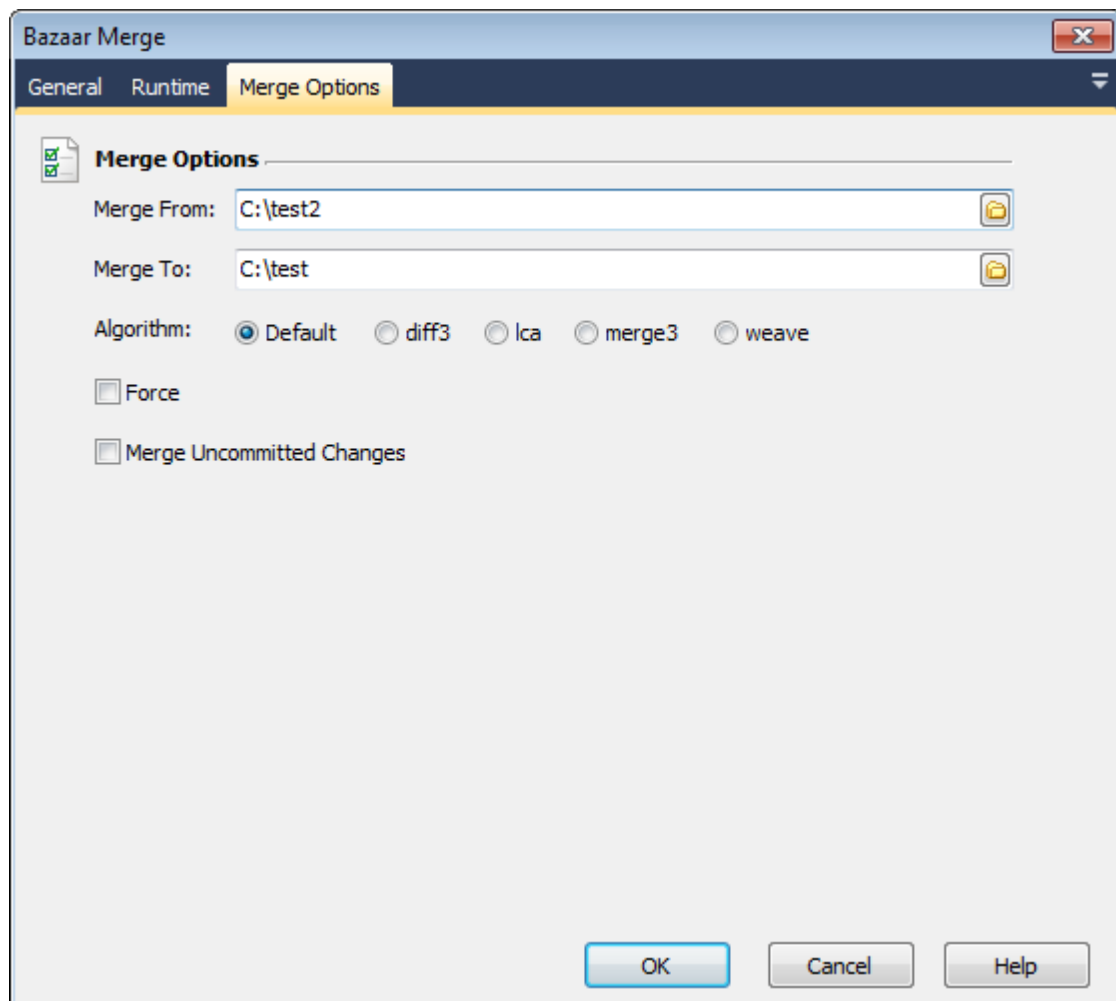
Revision to compare the current state with. If not specified, it will be compared against the latest checked in revision.

Fail Action If

Under what condition the action should fail. Useful for flow control within the project.

6.34.20.7 Bazaar Merge

This action enables you to merge one branch into another.

**Merge From**

The branch to merge from. This branch will not be modified.

Merge To

The branch to merge into. This branch will be changed to the result of the merge.

Algorithm

The algorithm to use for the merge. Default is usually merge3.

diff3: Merge using external diff3

lca: LCA-newness merge

merge3: Native diff3-style merge

weave: Weave-based merge

Force

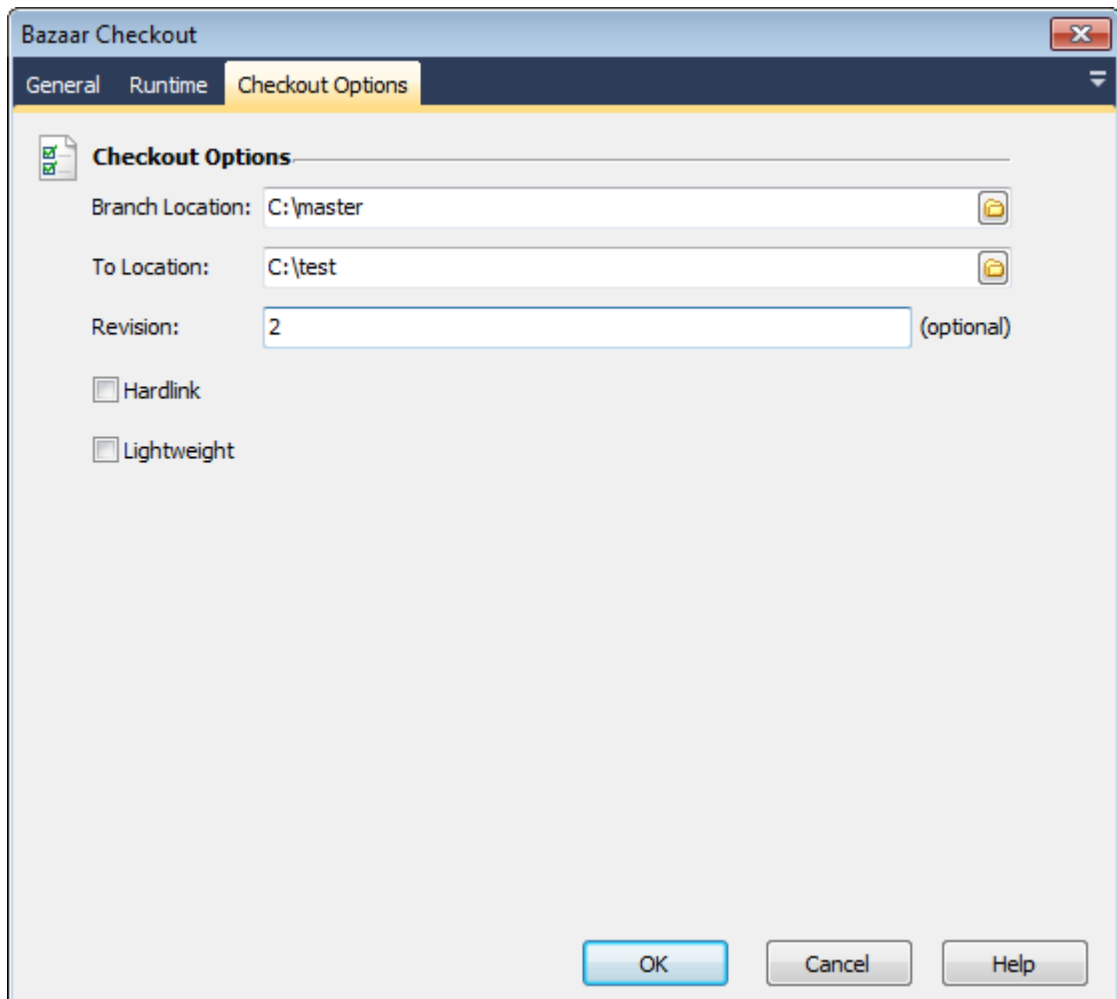
Merge even if the destination tree has uncommitted changes.

Merge Uncommitted Changes

Apply uncommitted changes from a working copy, instead of branch changes.

6.34.20.8 Bazaar Checkout

This action enables you to create a new checkout of an existing branch.

**Branch Location**

The location of the branch that is to be checked out.

To Location

The location to check the branch out into.

Revision

The revision of the branch to check out. If not specified, the latest checked in revision will be used.

Hardlink

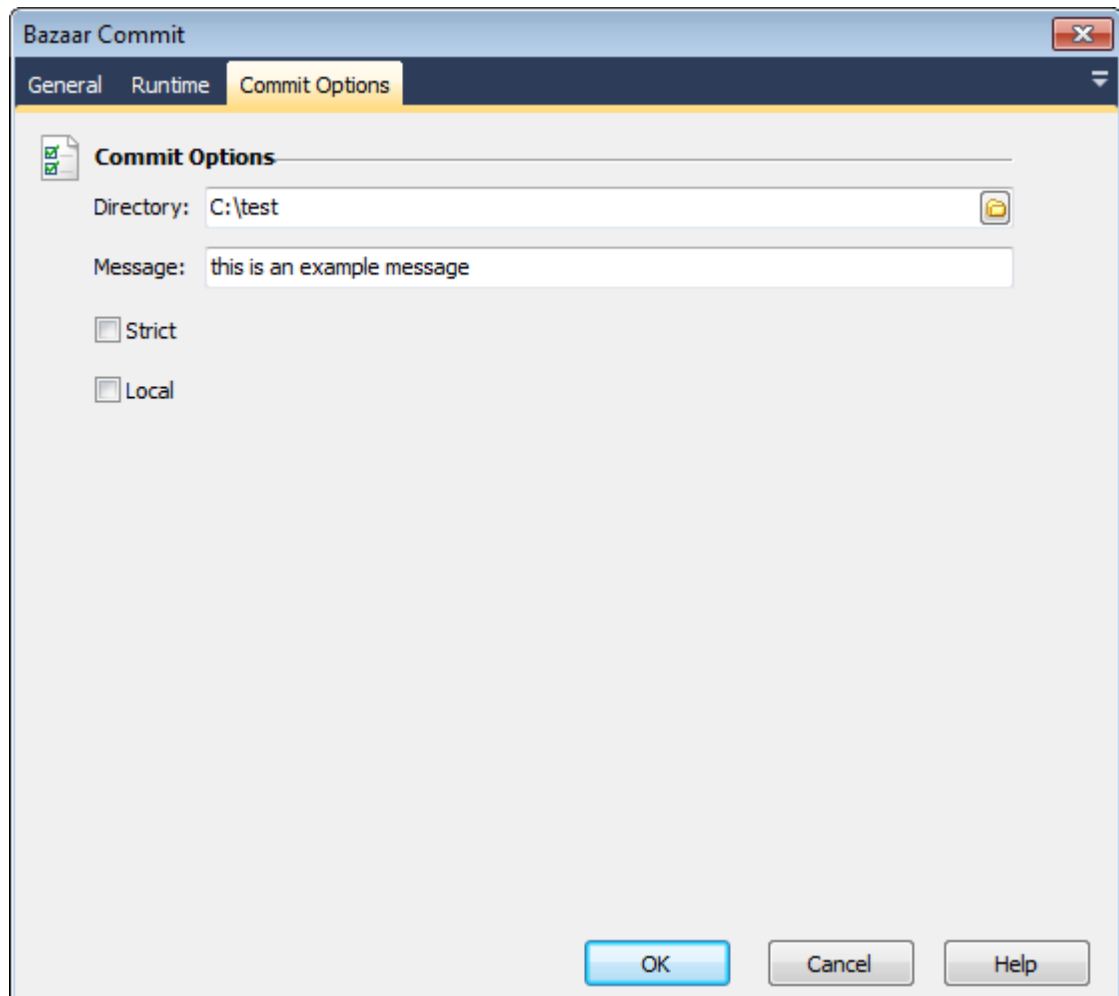
Hard-link working tree files where possible.

Lightweight

Perform a lightweight checkout. Lightweight checkouts depend on access to the branch for every operation. Normal checkouts can perform common operations like diff and status without such access, and also support local commits.

6.34.20.9 Bazaar Commit

This action enables you to commit changes in a directory into a new revision, and attach a message to the commit.

**Directory**

The directory to commit.

Message

The message to attach to the new commit

Strict

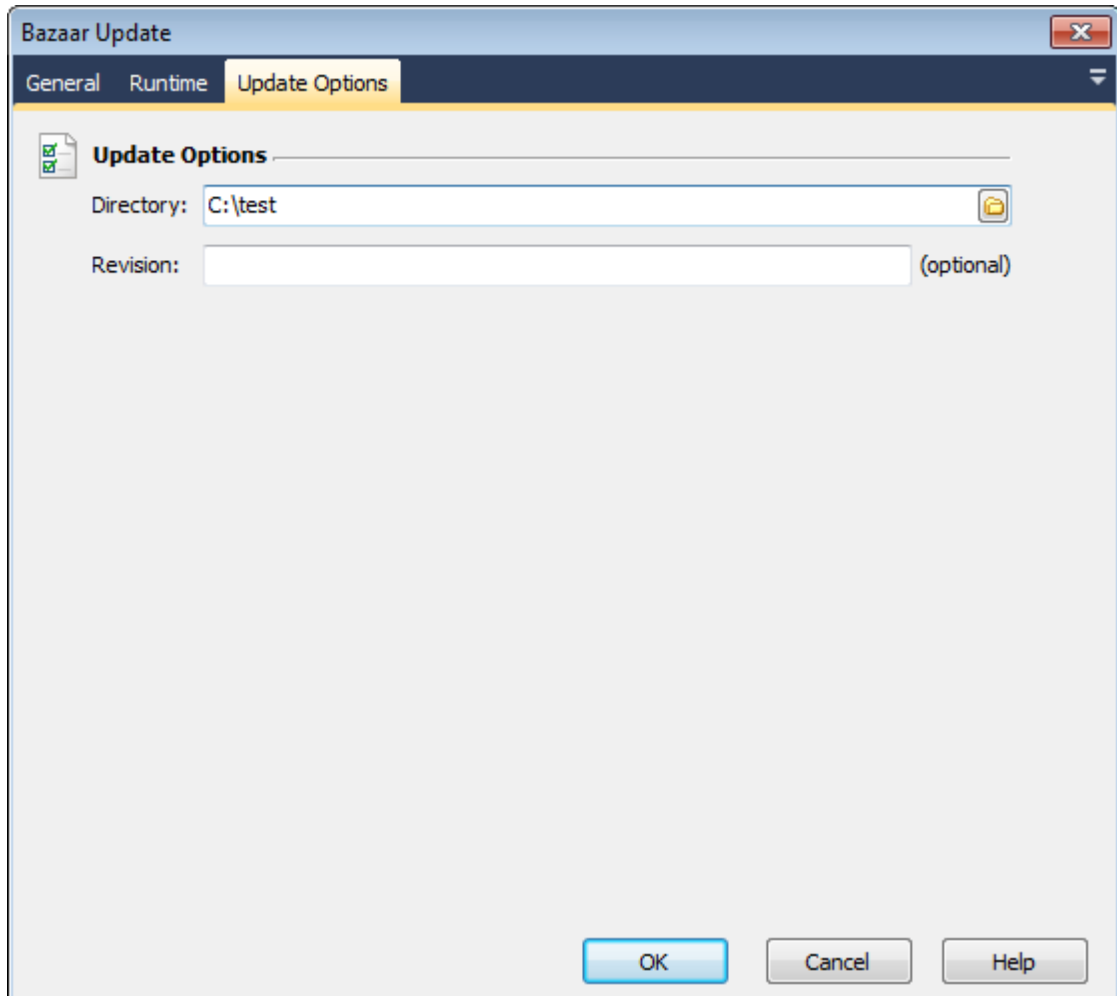
Refuse to commit if there are unknown files in the working tree.

Local

Perform a local commit in a bound branch. Local commits are not pushed to the master branch until a normal commit is performed.

6.34.20.1 Bazaar Update

This action enables you to update the working copy in the specified directory to be up to date with the branch.

**Directory**

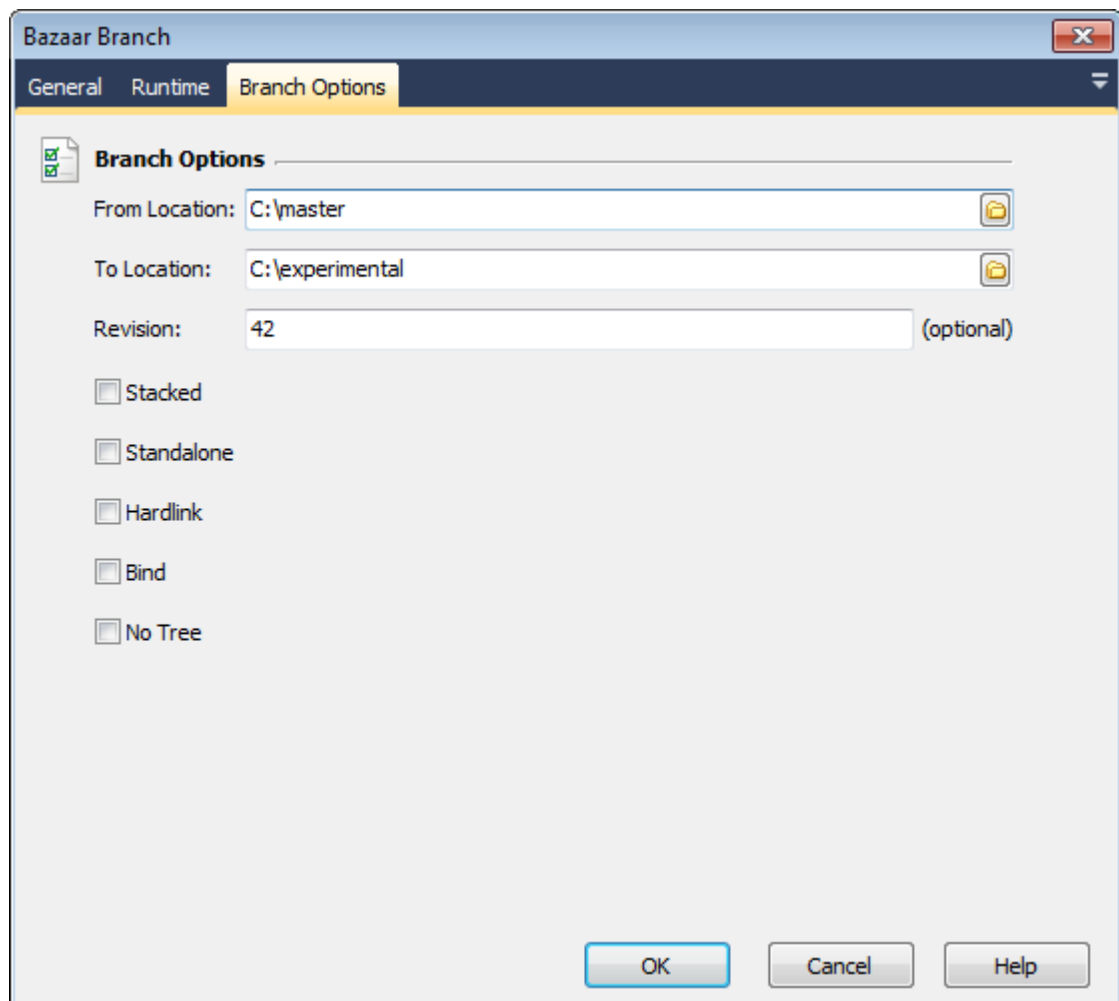
The directory containing the working copy.

Revision

The revision of the branch to update to. If not specified, the latest checked in revision will be used.

6.34.20.1 Bazaar Branch

This action enables you to create a new branch as a copy of an existing branch.

**From Location**

The branch to use as the source for the new branch.

To Location

The directory to create as the new branch.

Revision

The revision to use as the source. If no revision is specified, the latest committed revision will be used.

Stacked

Create a stacked branch referring to the source branch. The new branch will depend on the availability of the source branch for all operations.

Standalone

Do not use a shared repository, even if available.

Hardlink

Hard-link working tree files where possible.

Bind

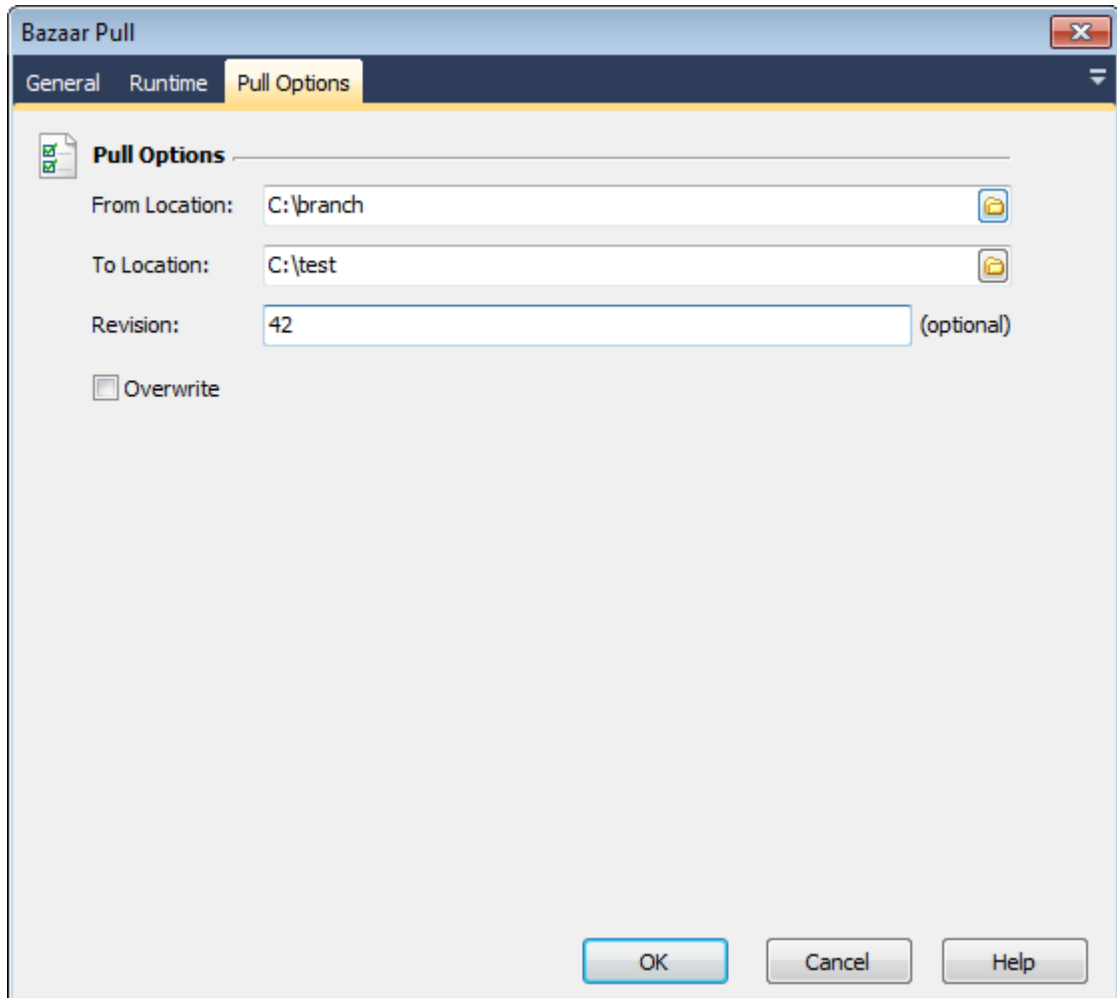
Bind new branch to the "from location".

No Tree

Create a branch without a working tree.

6.34.20.1 Bazaar Pull

This action enables you to turn an existing branch into a mirror of another branch.

**From Location**

The location to pull from.

To Location

The location to pull into.

Revision

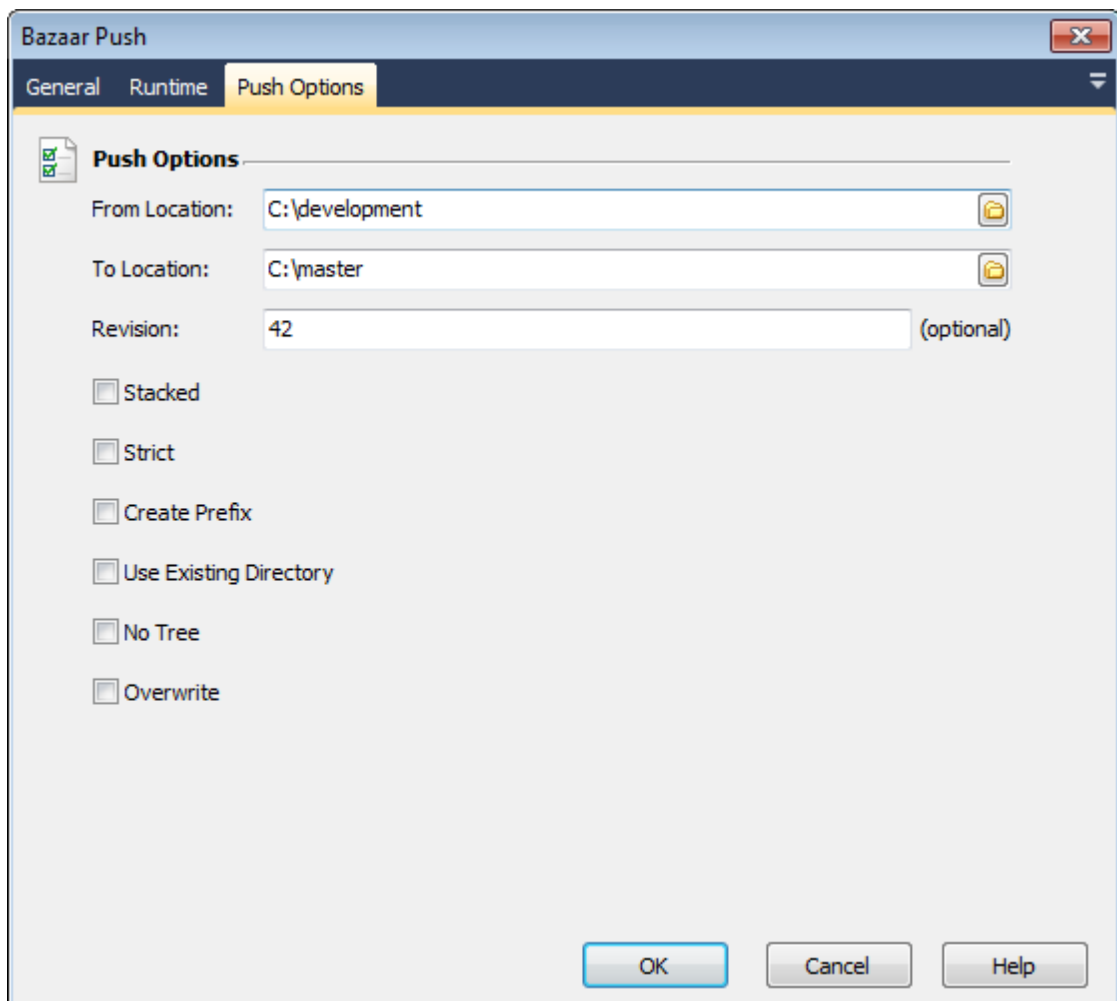
The revision to pull. If not specified, the latest checked in revision will be used.

Overwrite

Ignore differences between branches and overwrite unconditionally.

6.34.20.1 Bazaar Push

This action enables you to update a mirror of a branch.

**From Location**

Location to push from.

To Location

Location to push to.

Revision

Revision to push. If not specified, the latest checked in revision will be used.

Stacked

Create a stacked branch that references the public location of the parent branch.

Strict

Refuse to push if there are uncommitted changes in the working tree.

Create Prefix

Create the path leading up to the branch if it does not already exist.

Use Existing Directory

By default, push will fail if the target directory exists, but does not already have a control directory. This option will allow push to proceed.

No Tree

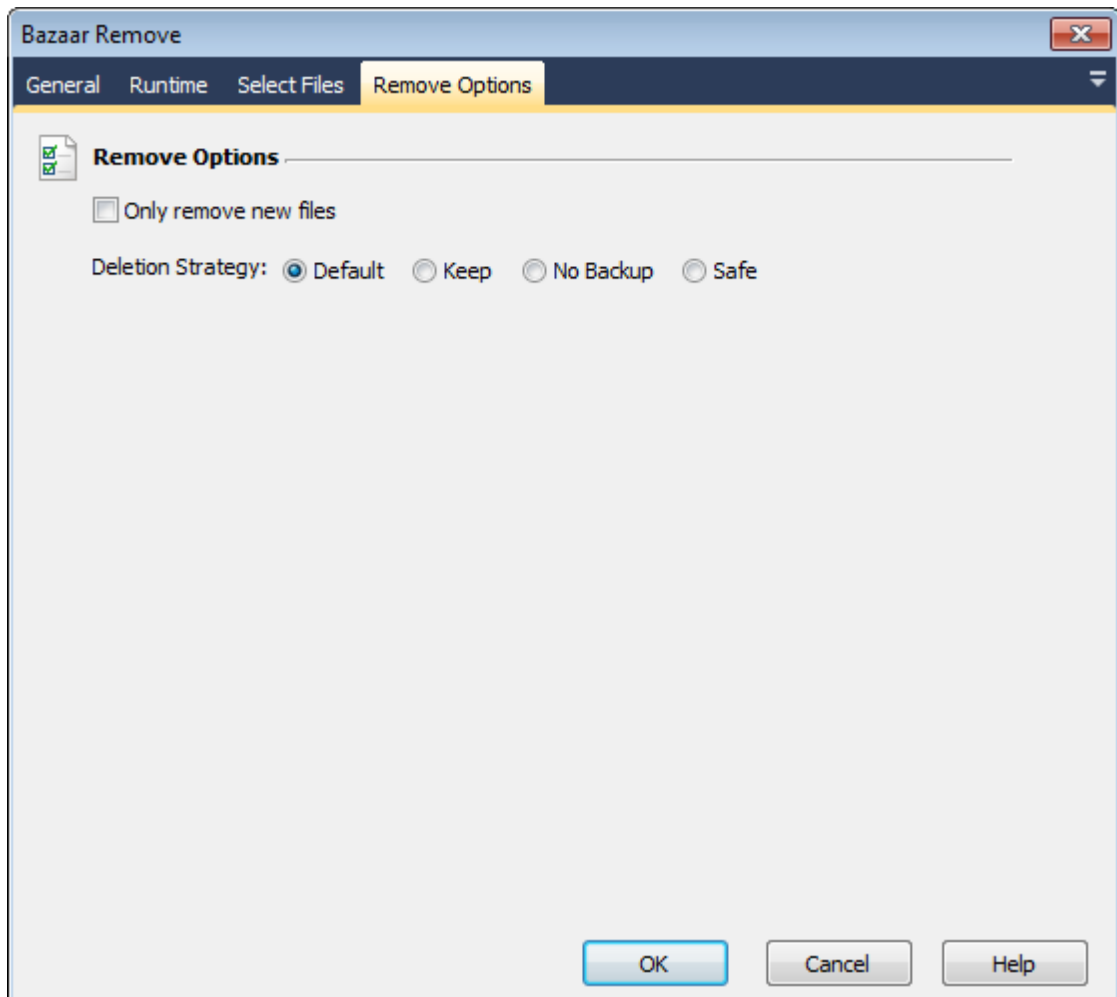
Don't populate the working tree, even for protocols that support it.

Overwrite

Ignore differences between branches and overwrite unconditionally.

6.34.20.1 Bazaar Remove

This action enables you to remove the specified files and/or folders from a branch.

**Select Files**

See Select Files.

Only remove new files

Only remove files that have never been committed.

Deletion Strategy

Default: Use the default deletion strategy.

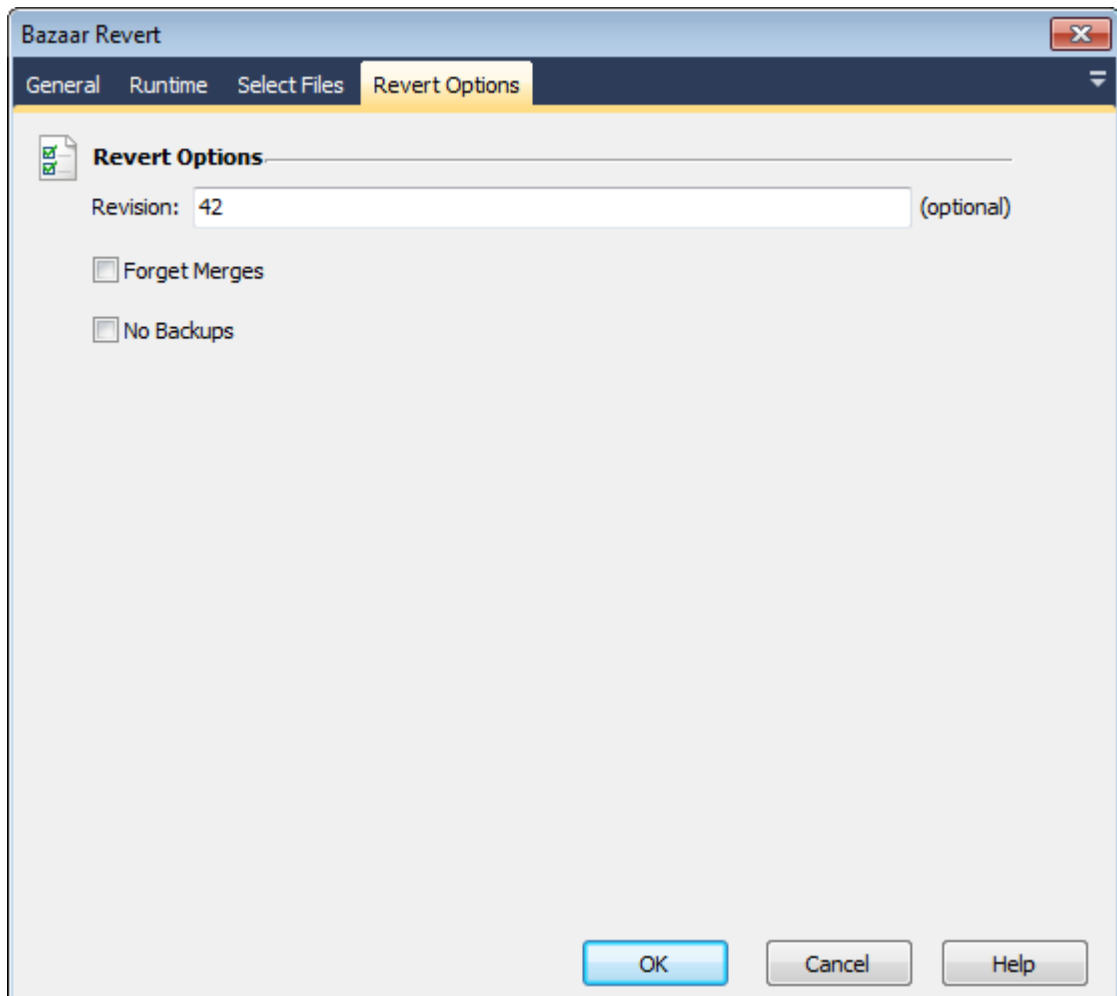
Keep: Delete from Bazaar but leave the working copy.

No Backup: Don't backup changed files.

Safe: Backup changed files.

6.34.20.1 Bazaar Revert

This action enables you to revert the specified files and/or folders to a previous committed revision.



Select Files

See Select Files.

Revision

The revision to revert to. If not specified, the latest checked in revision will be used instead.

Forget Merges

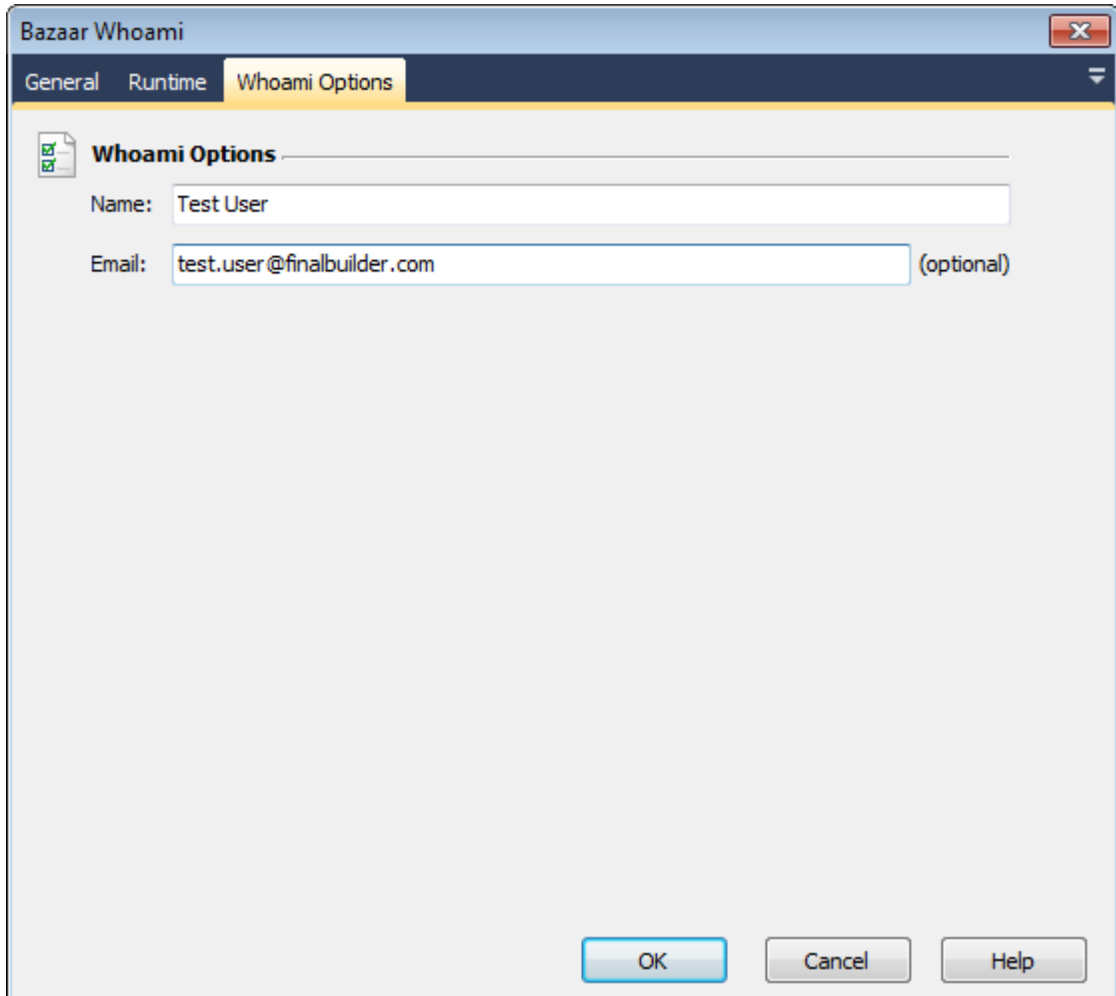
Remove pending merge marker, without changing any files.

No Backups

Do not save backups of reverted files.

6.34.20.1 Bazaar Whoami

This action enables you to set the user ID that Bazaar using.



The screenshot shows a dialog box titled "Bazaar Whoami" with a close button (X) in the top right corner. The dialog has three tabs: "General", "Runtime", and "Whoami Options", with the "Whoami Options" tab selected. Below the tabs, there is a section titled "Whoami Options" with a small icon of a document with checkmarks. This section contains two text input fields: "Name:" with the value "Test User" and "Email:" with the value "test.user@finalbuilder.com". The "Email:" field is followed by the text "(optional)". At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help".

Name

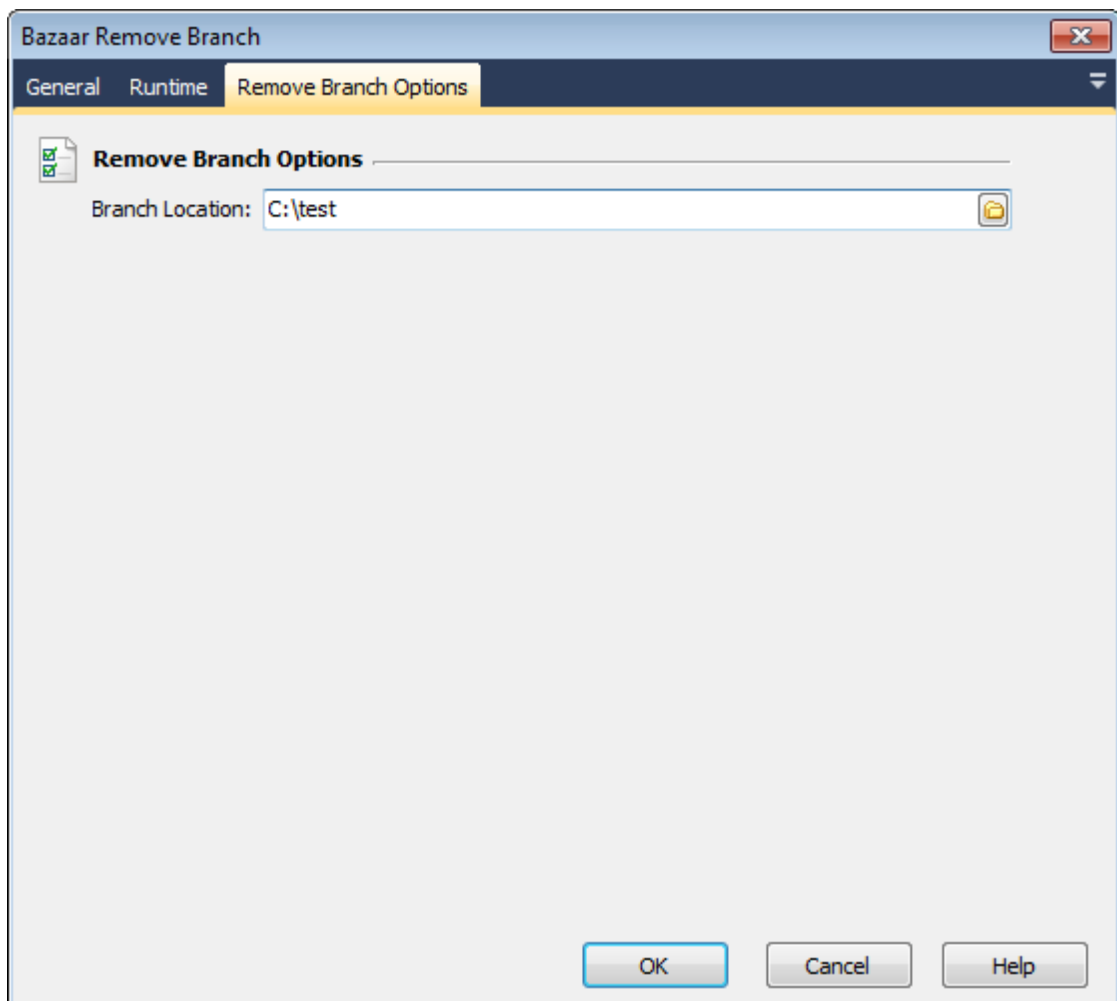
Name of the user.

Email

Email of the user. Optional, but recommended.

6.34.20.1 Bazaar Remove Branch

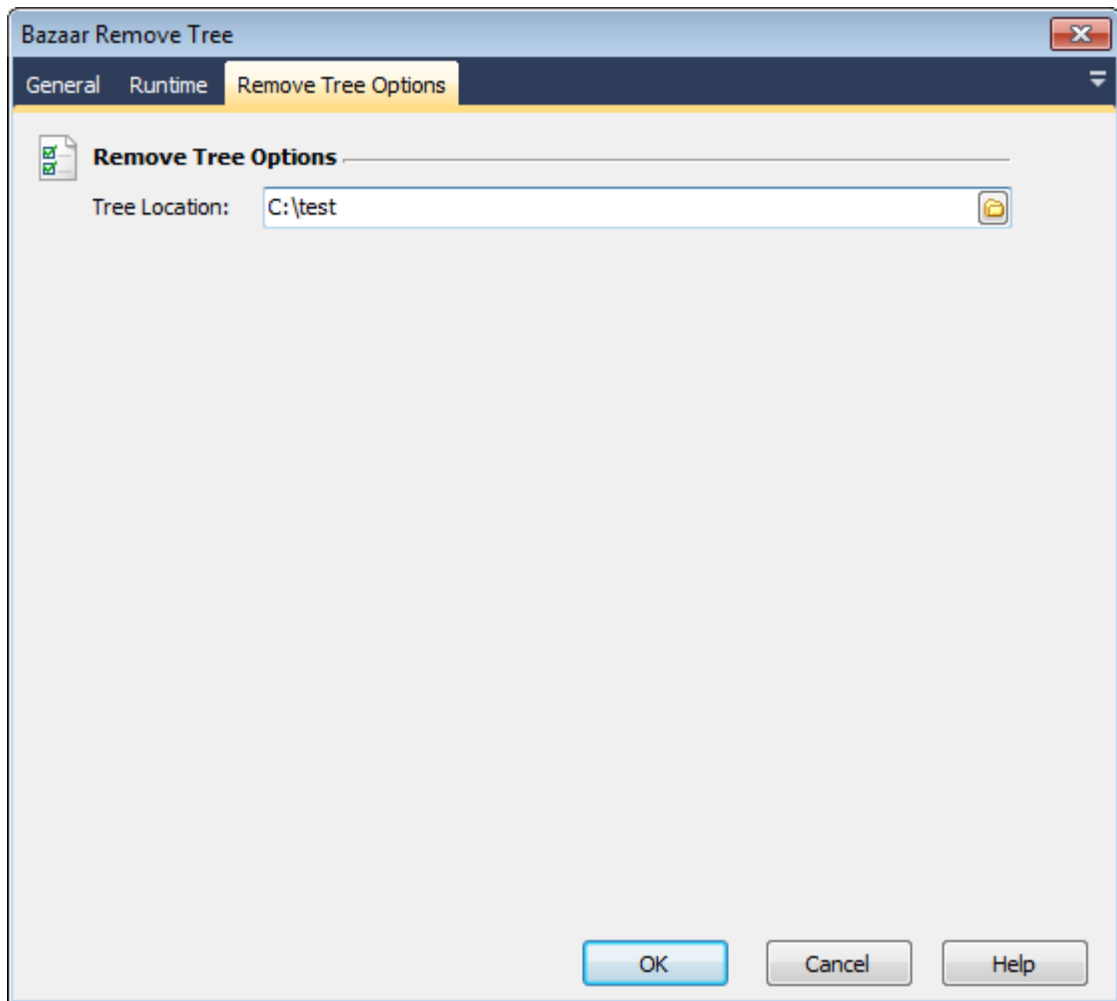
This action enables you to remove a branch from the specified location.

**Branch Location**

Location of the branch to remove.

6.34.20.1 Bazaar Remove Tree

This action enables you to remove the working tree from the given directory.

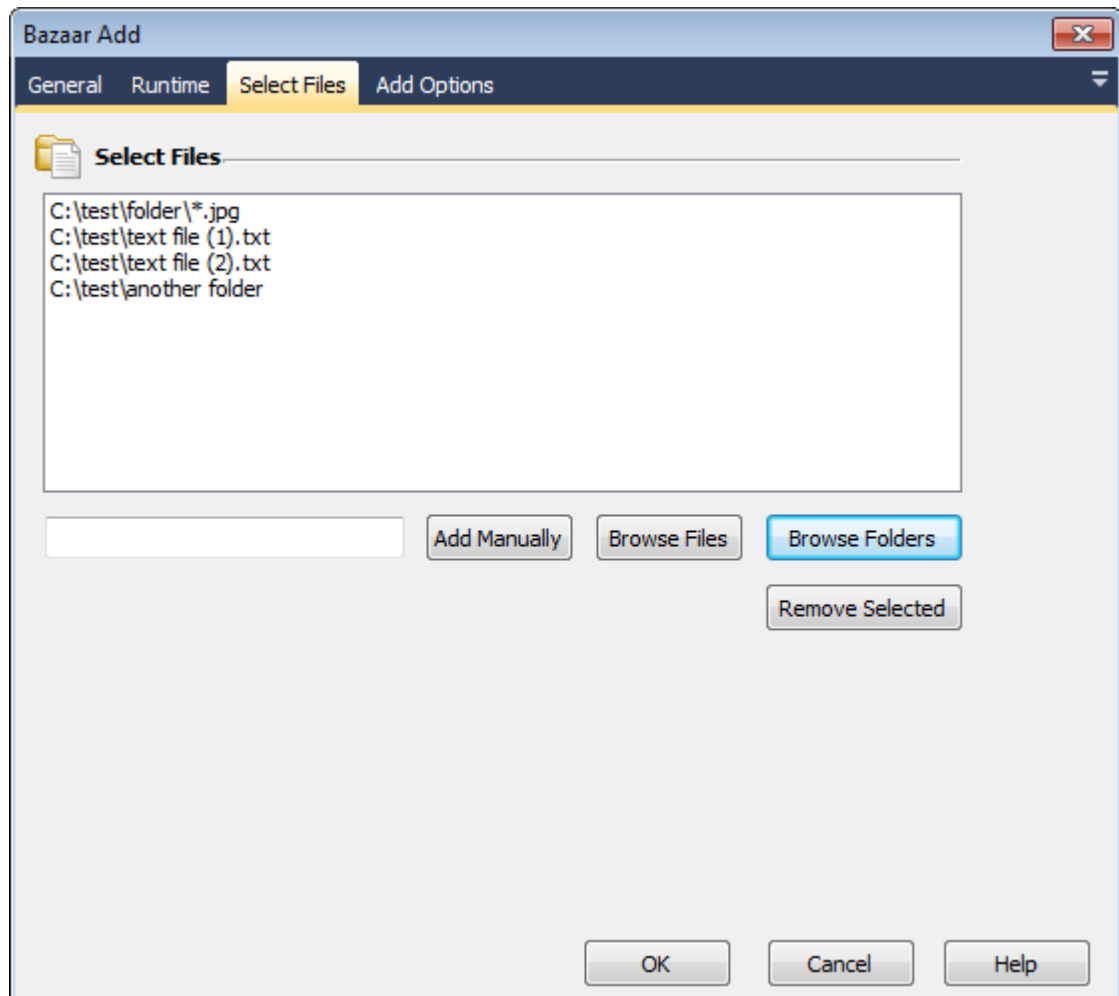


Tree Location

The directory that contains the working tree.

6.34.20.1 Select Files

This page allows you to select which files and/or folders that the action should operate on.



Add Manually

Add the text in the text box to the list of files and folders. Useful for adding wildcards and other patterns to the list.

Browse Files

Add the selected file to the list of files and folders.

Browse Folders

Add the selected folder to the list of files and folders.

Remove Selected

Remove the selected item from the list.

6.35 Virtualization Systems

6.35.1 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

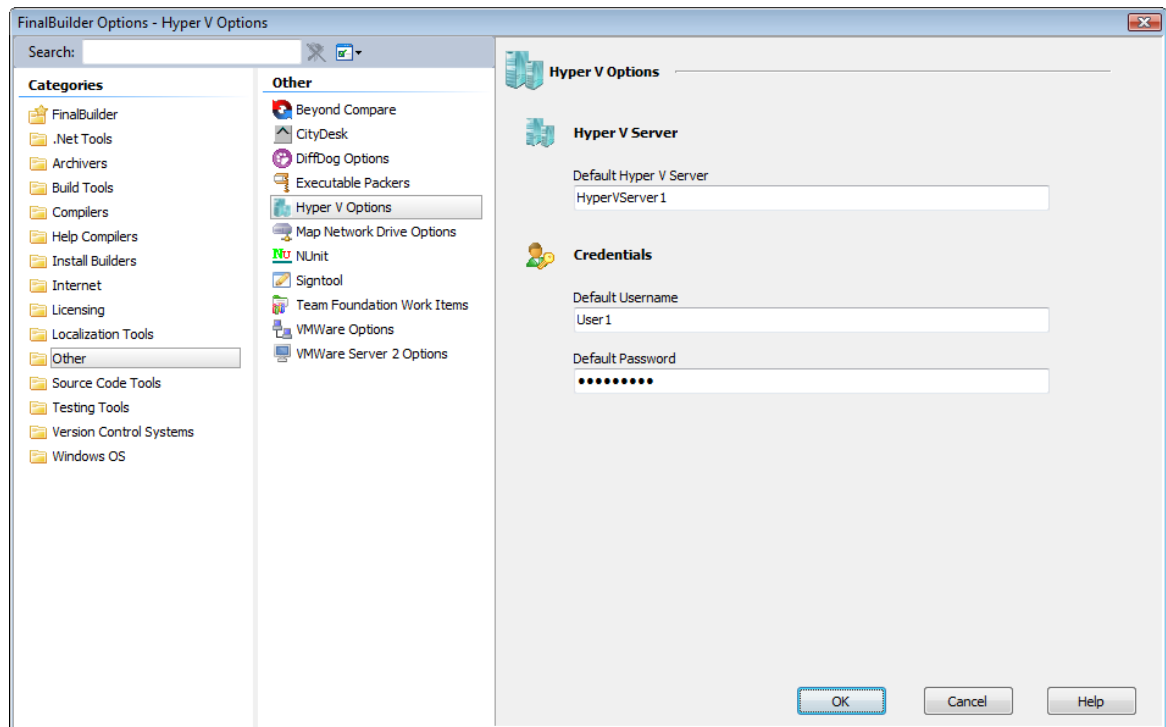
The following actions are available:

- 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.



6.35.1.1 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.

The screenshot shows a Windows-style dialog box titled "Hyper V List Virtual Machines". It has four tabs: "General", "Runtime", "Hyper V Server" (which is selected and highlighted in yellow), and "List Virtual Machines". The "Hyper V Server" tab contains two main sections. The first section, "Hyper V Server", has two radio button options: "Use Server Specified in Hyper V Options" (unselected) and "Provide the Hyper V Server Name" (selected). Below these is a text field labeled "Hyper V Server:" containing the text "HypervServer2". The second section, "Credentials", has a checked checkbox "Use Credentials". Below it are two more radio button options: "Use Credentials Specified in Hyper V Options" (unselected) and "Provide Credentials" (selected). Below these are two text fields: "Username:" containing "User2" and "Password:" containing a series of dots. At the bottom right are three buttons: "OK", "Cancel", and "Help".

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:

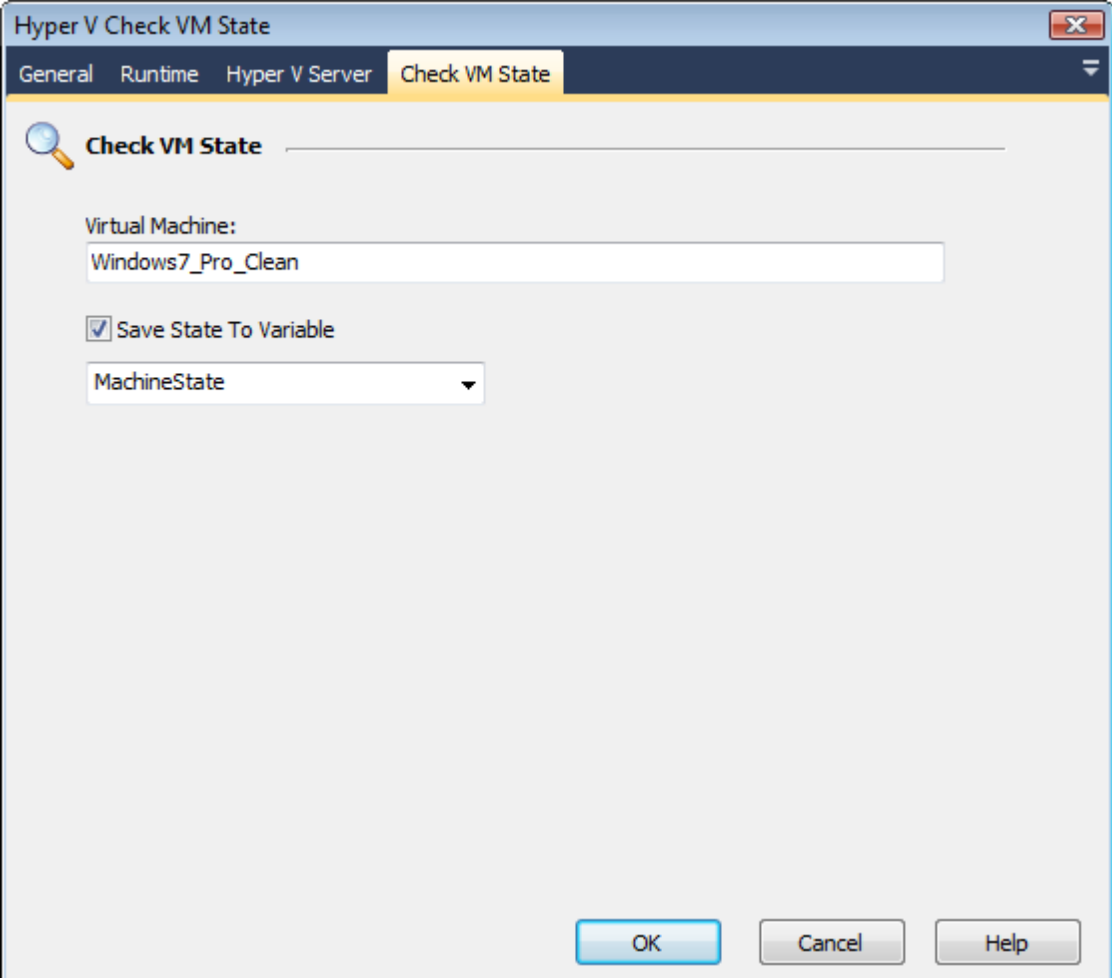
- **Use Credentials Specified in Hyper V Options** - If you have specified default credentials on the Hyper V Options page that you wish to use, select this option.
- **Provide Credentials** - This option allows you to specify the credentials to be used.

6.35.1.2 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.



The screenshot shows a dialog box titled "Hyper V Check VM State". It has four tabs: "General", "Runtime", "Hyper V Server", and "Check VM State", with the last tab being active. The main area contains a search icon and the text "Check VM State". Below this, there is a label "Virtual Machine:" followed by a text input field containing "Windows7_Pro_Clean". Underneath is a checked checkbox labeled "Save State To Variable" and a dropdown menu currently showing "MachineState". At the bottom right, there are three buttons: "OK", "Cancel", and "Help".

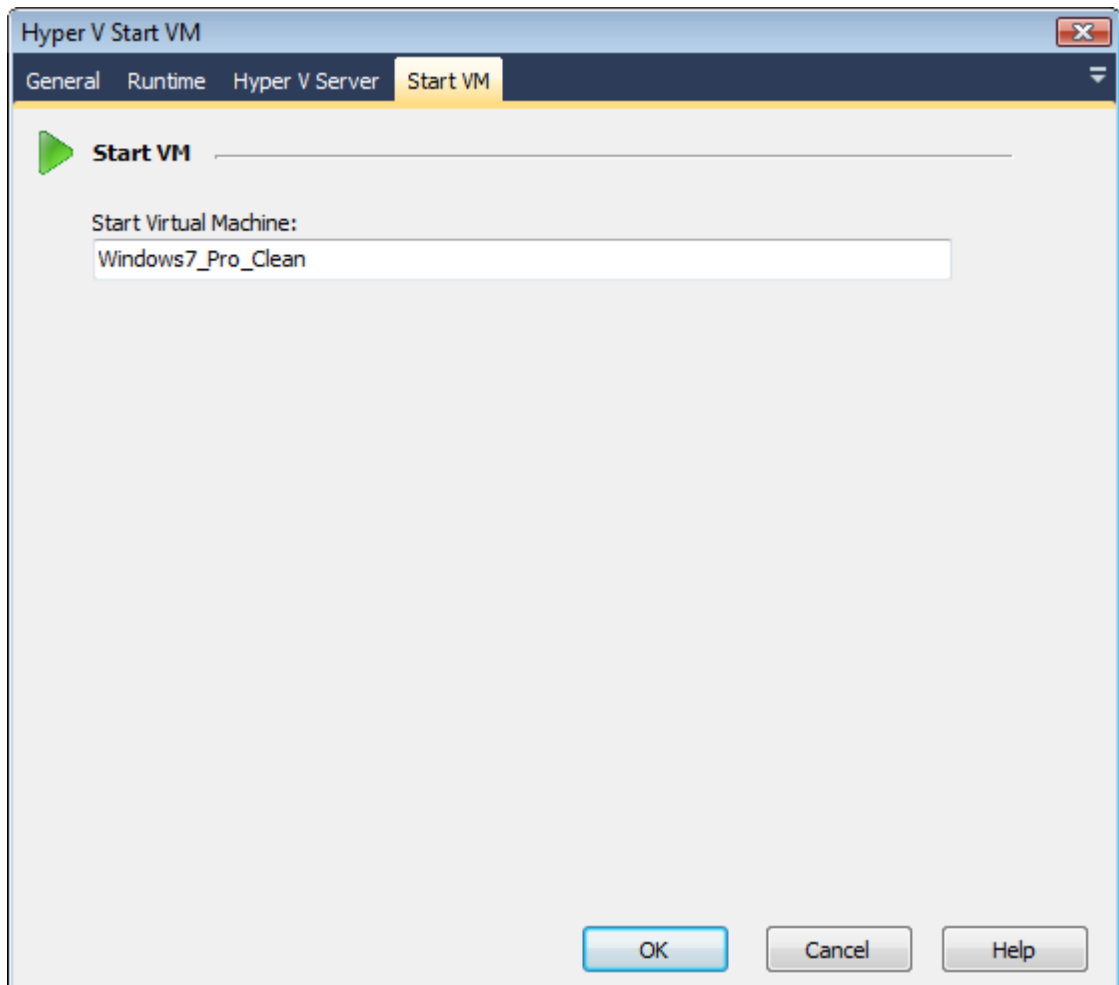
You also have the option saving the machine state into a variable for later use.

6.35.1.3 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.

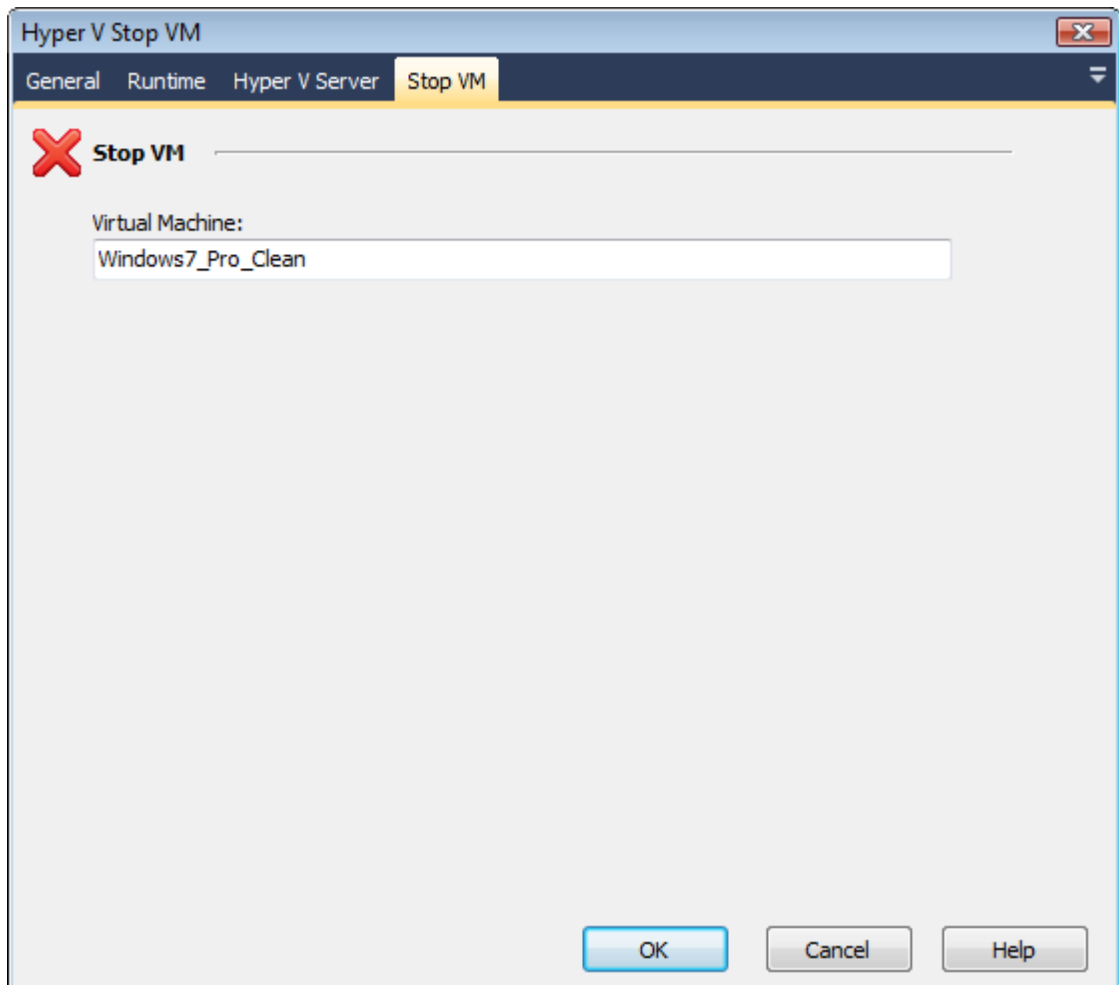


6.35.1.4 Hyper V Stop VM

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.

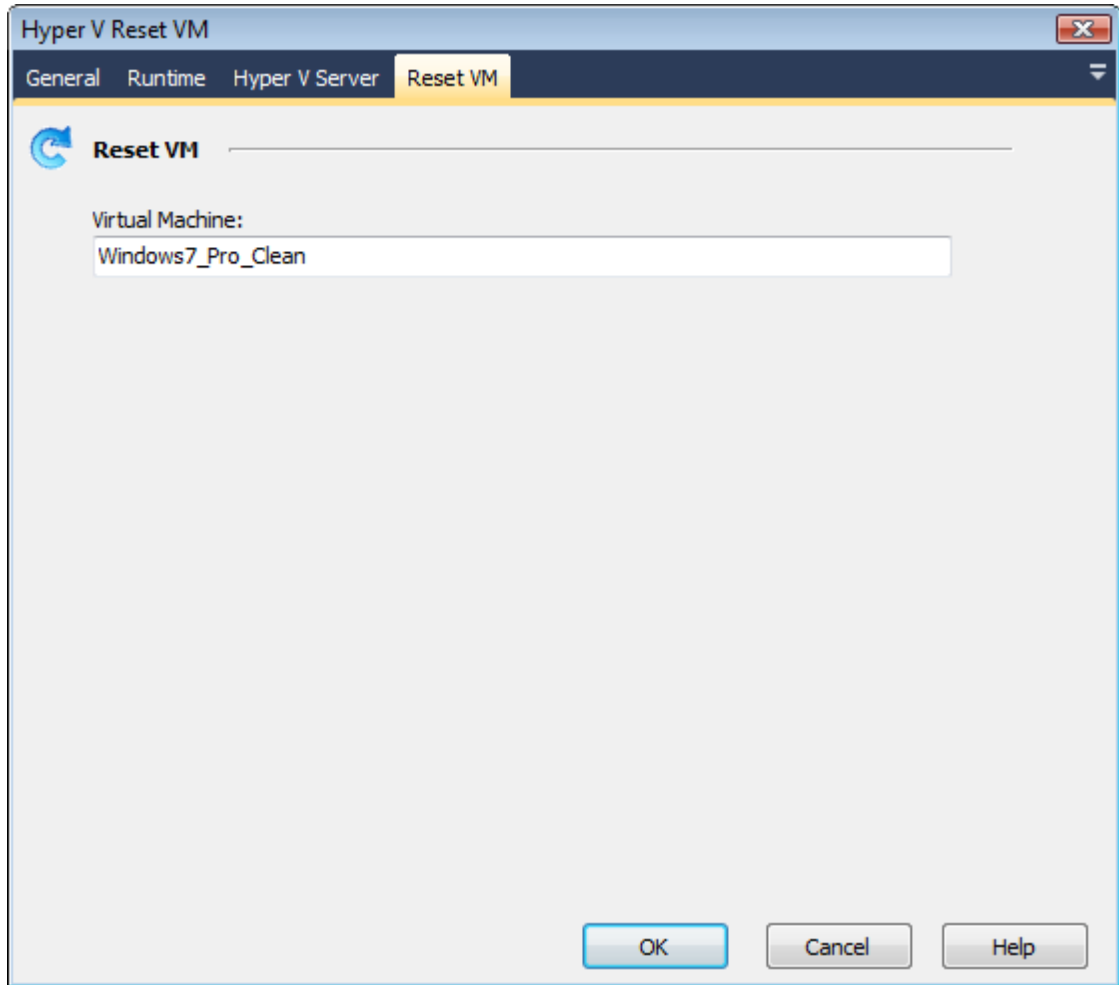


6.35.1.5 Hyper V Reset VM

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.

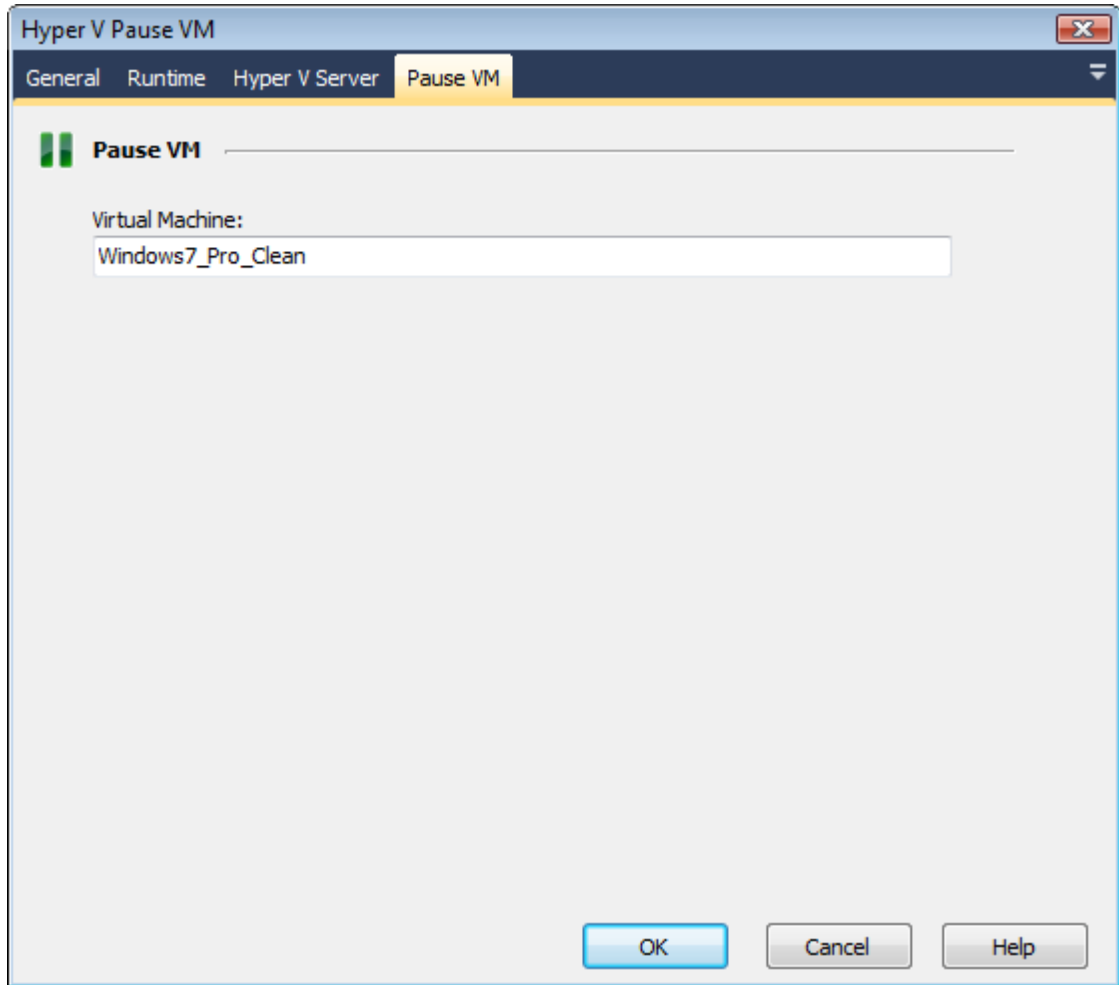


6.35.1.6 Hyper V Pause VM

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.

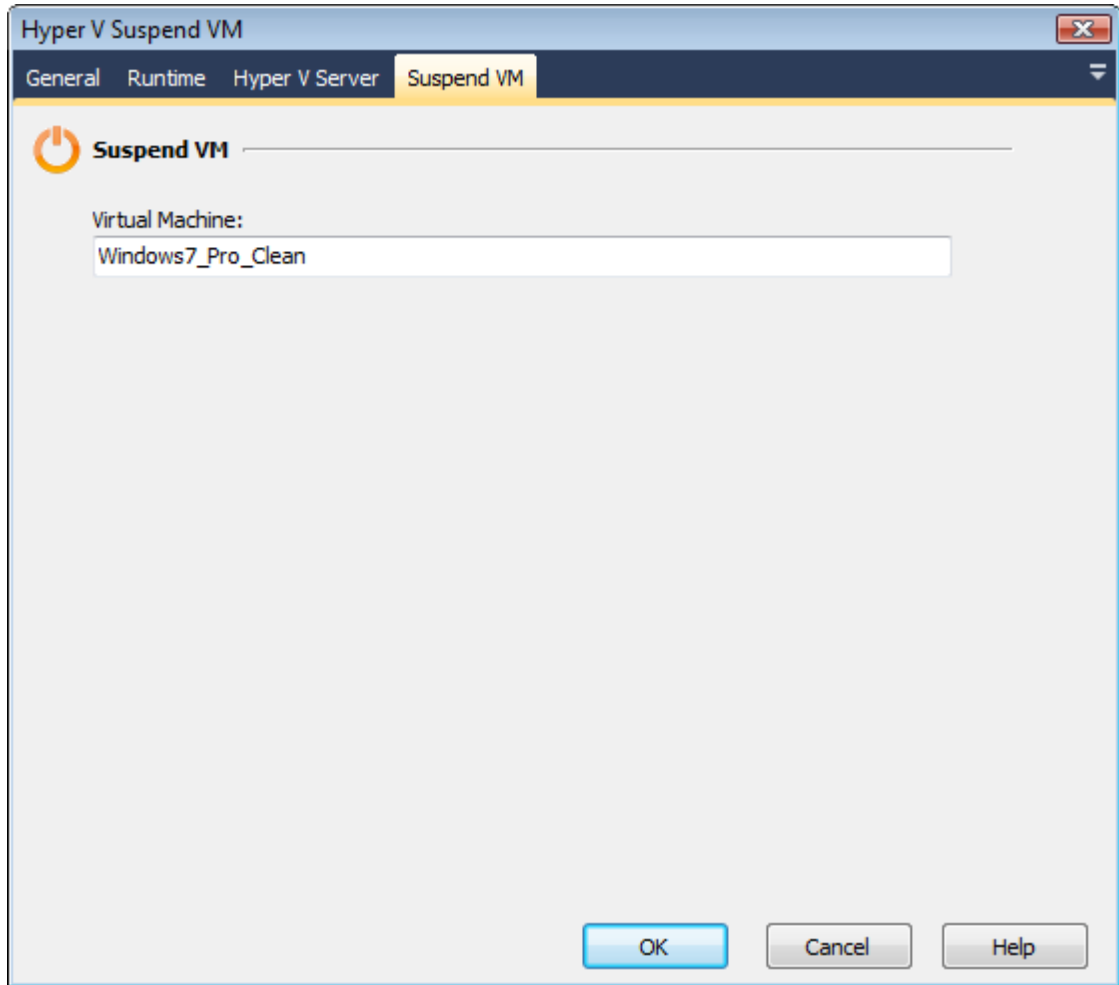


6.35.1.7 Hyper V Suspend VM

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.

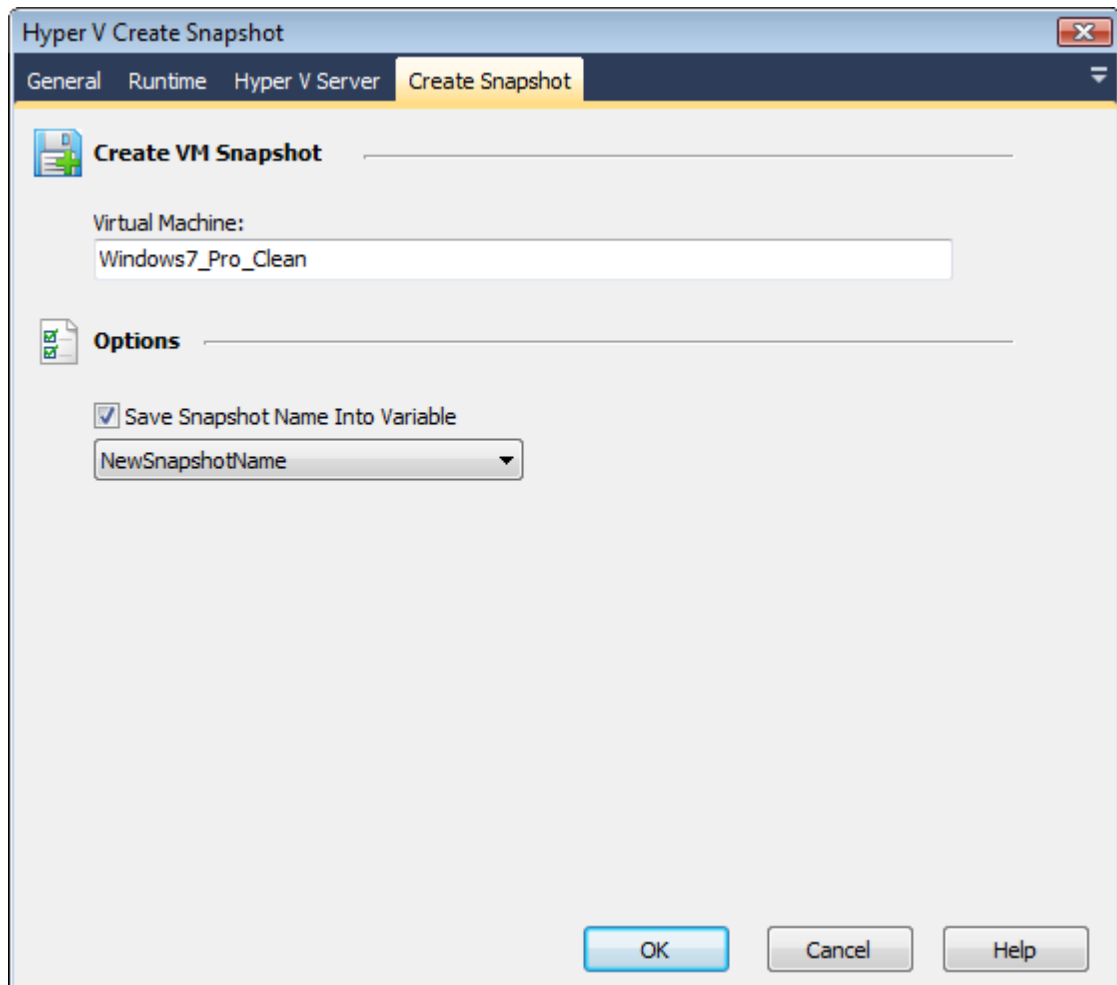


6.35.1.8 Hyper V Create VM Snapshot

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.



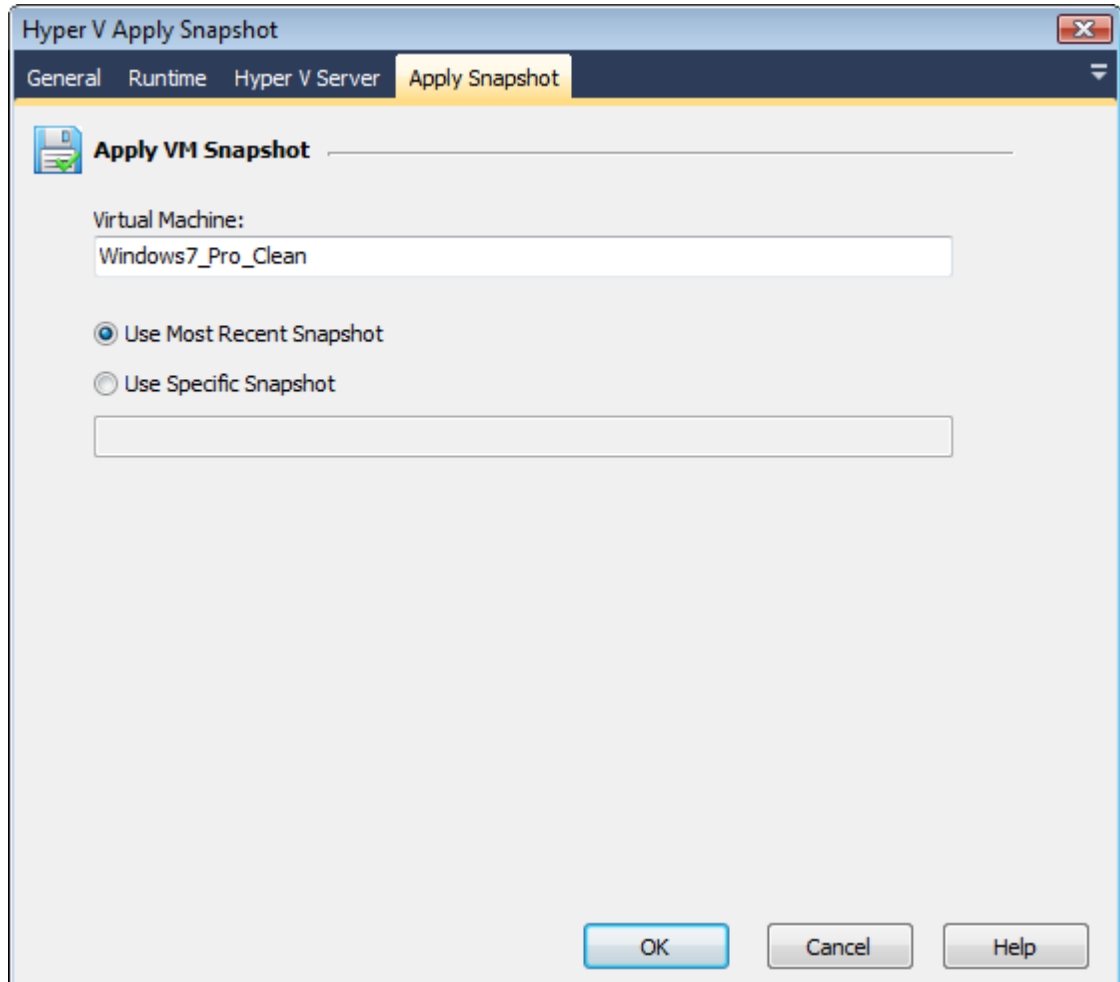
On successful creation of a system snapshot, the name of the new snapshot will be written to the log. By using the **Save Snapshot Name Into Variable** option, the name of the newly created snapshot can be saved into a specified variable for later use.

6.35.1.9 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.



Select from the following two options:

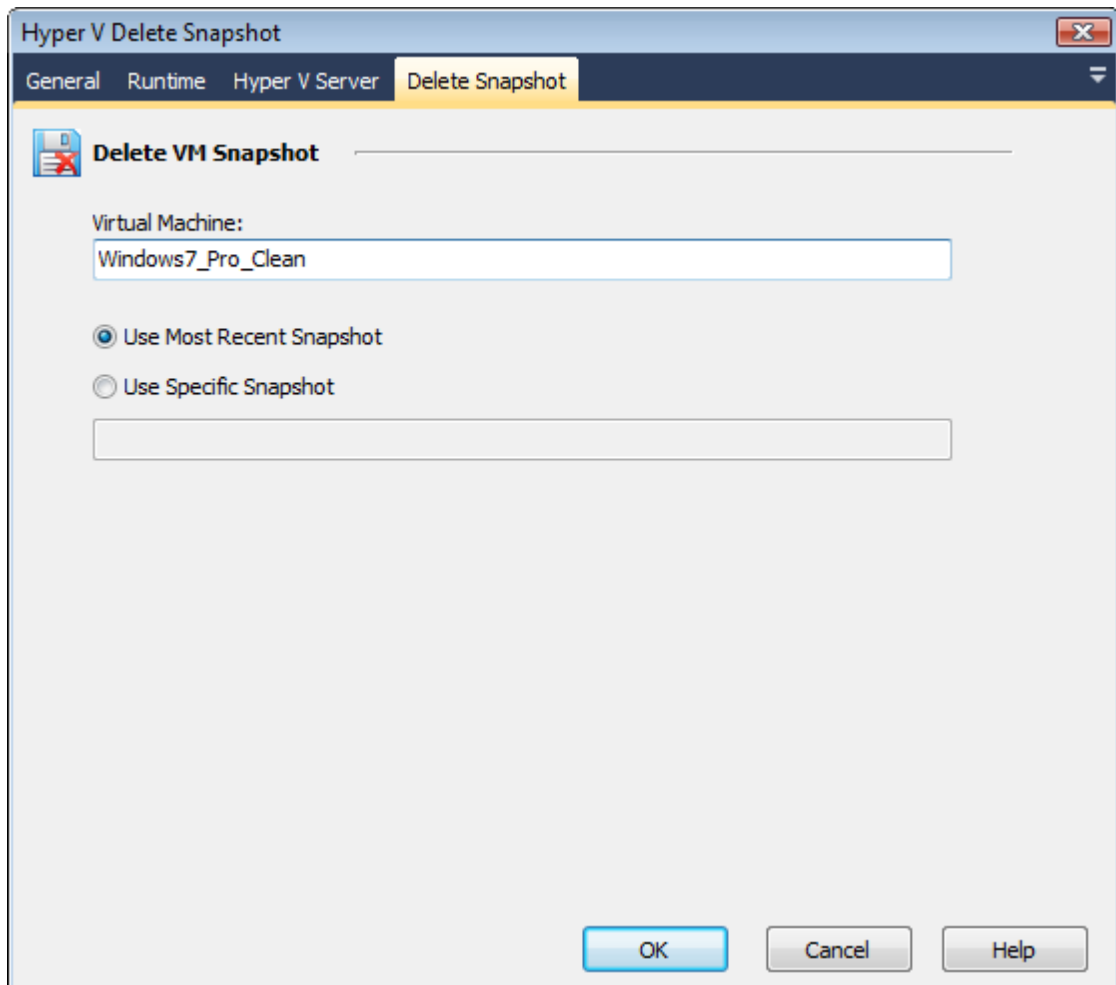
- **Use Most Recent Snapshot** - Use the snapshot that was created most recently.
- **Use Specific Snapshot** - Provide the name of a snapshot to restore the virtual machine to.

6.35.1.10 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.



Select from the following two options:

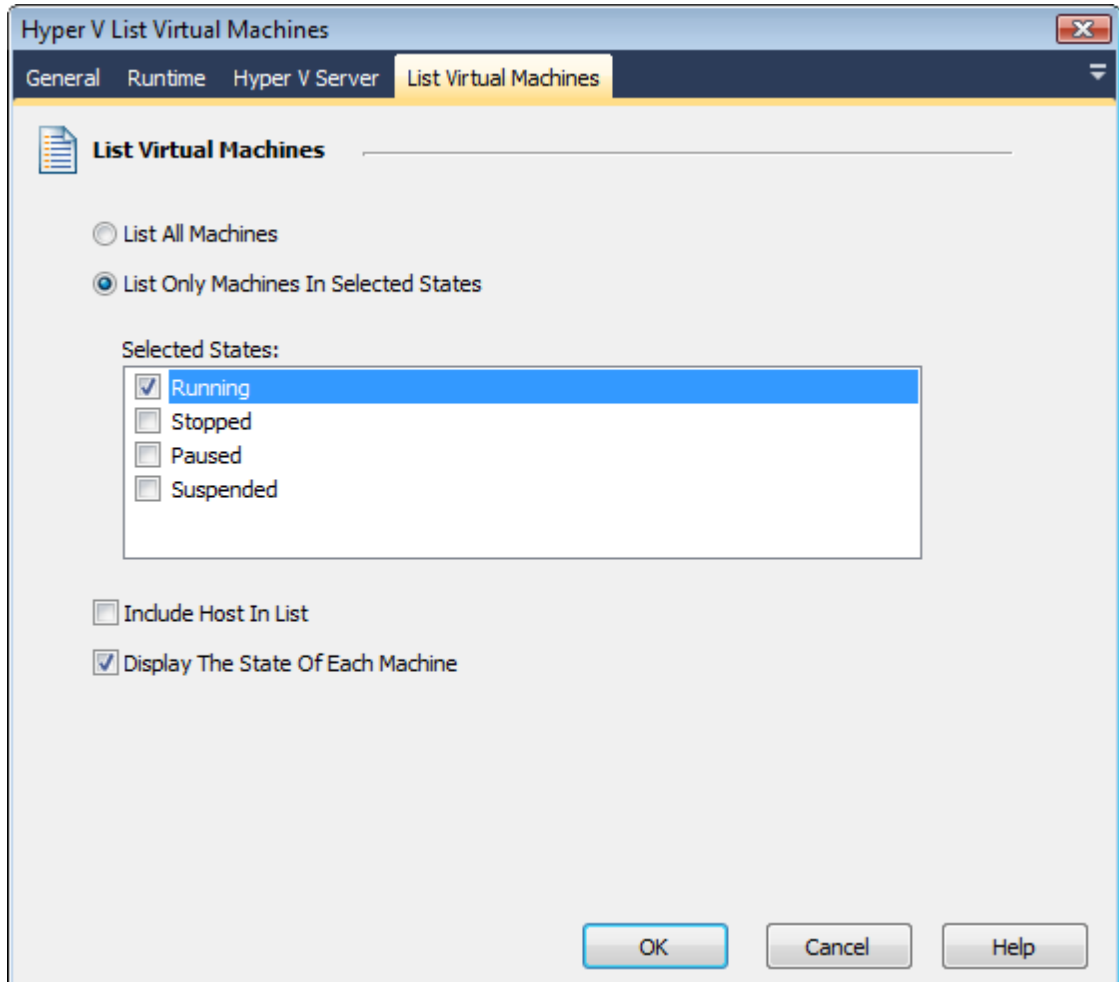
- **Use Most Recent Snapshot** - Delete the snapshot that was created most recently.
- **Use Specific Snapshot** - Provide the name of a snapshot to delete.

6.35.1.11 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 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.

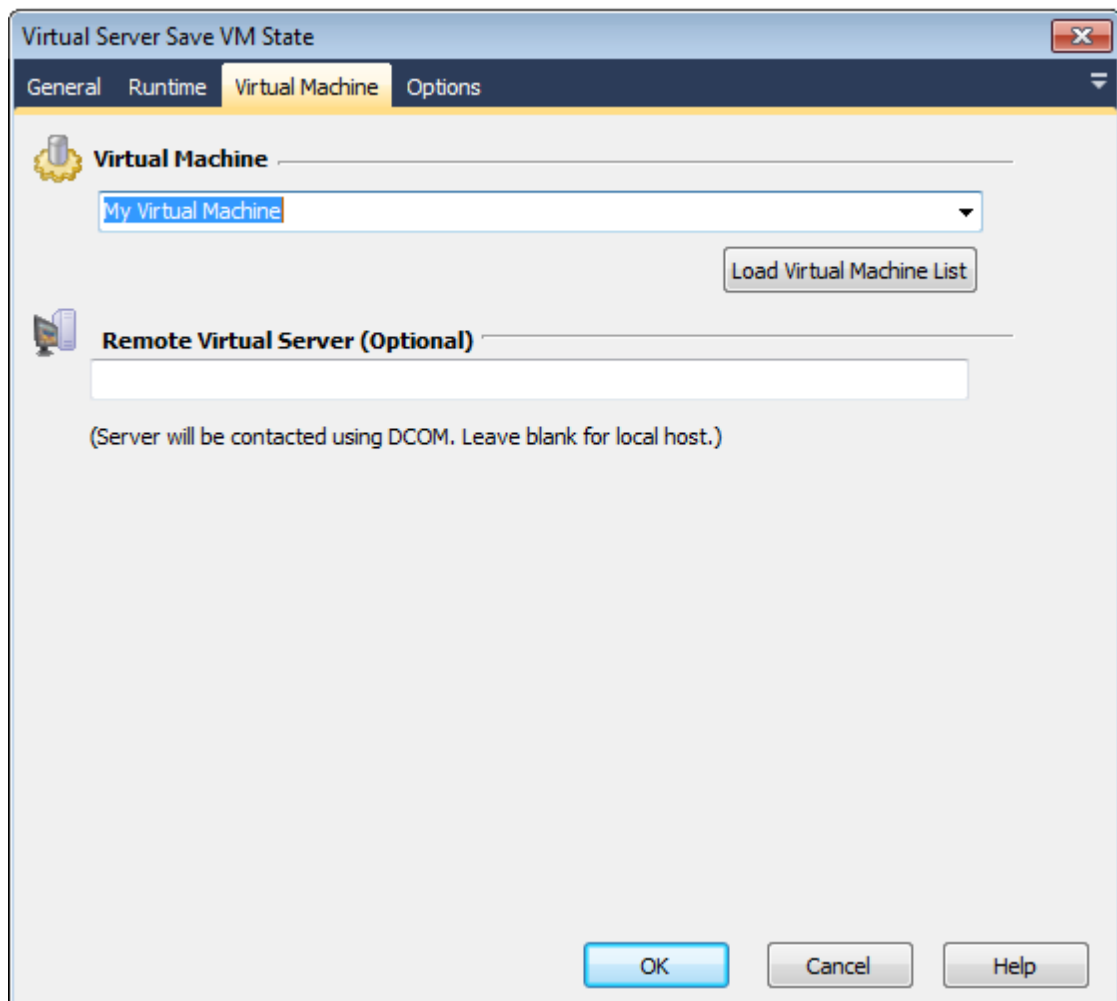
6.35.2 Microsoft Virtual Server

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.)

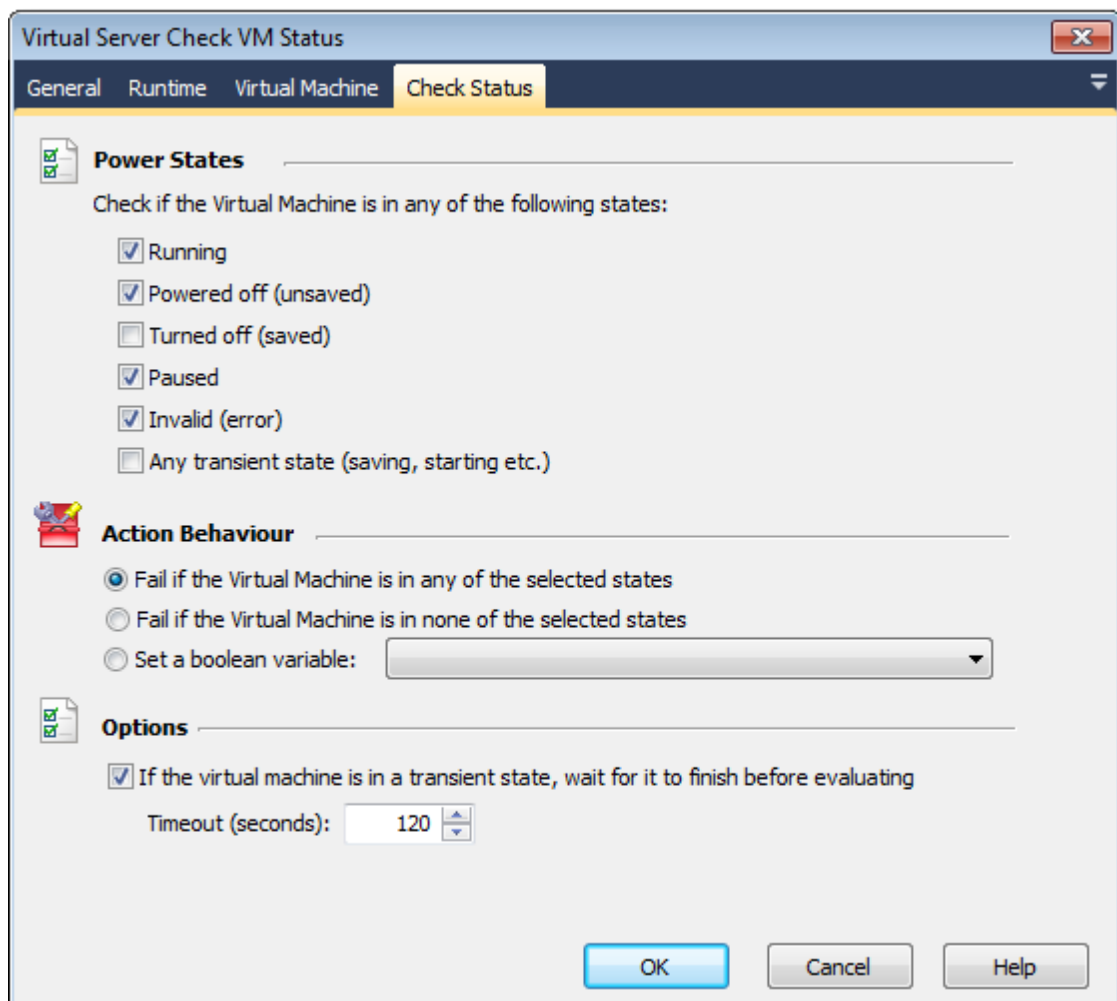
If the Virtual Server is running on the local machine, leave this field blank.

6.35.2.1 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.



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"

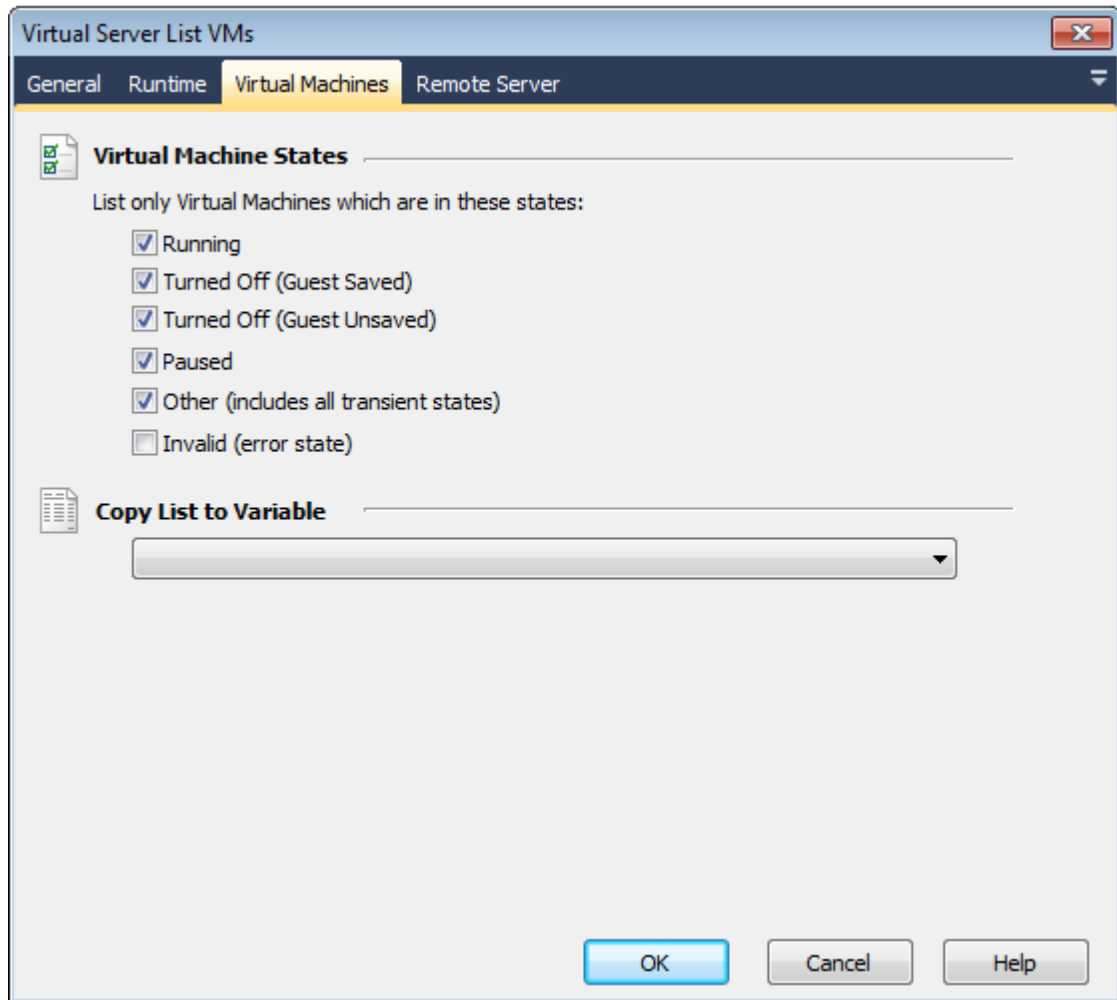
If this option is selected, the action will wait for any "transient" state to have completed before it goes ahead to check the state of the VM. If the action is still in the transient state when the timeout is reached, the action will fail.

(Note that this option does not apply if the "Any transient state" option is selected above.)

6.35.2.2 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.

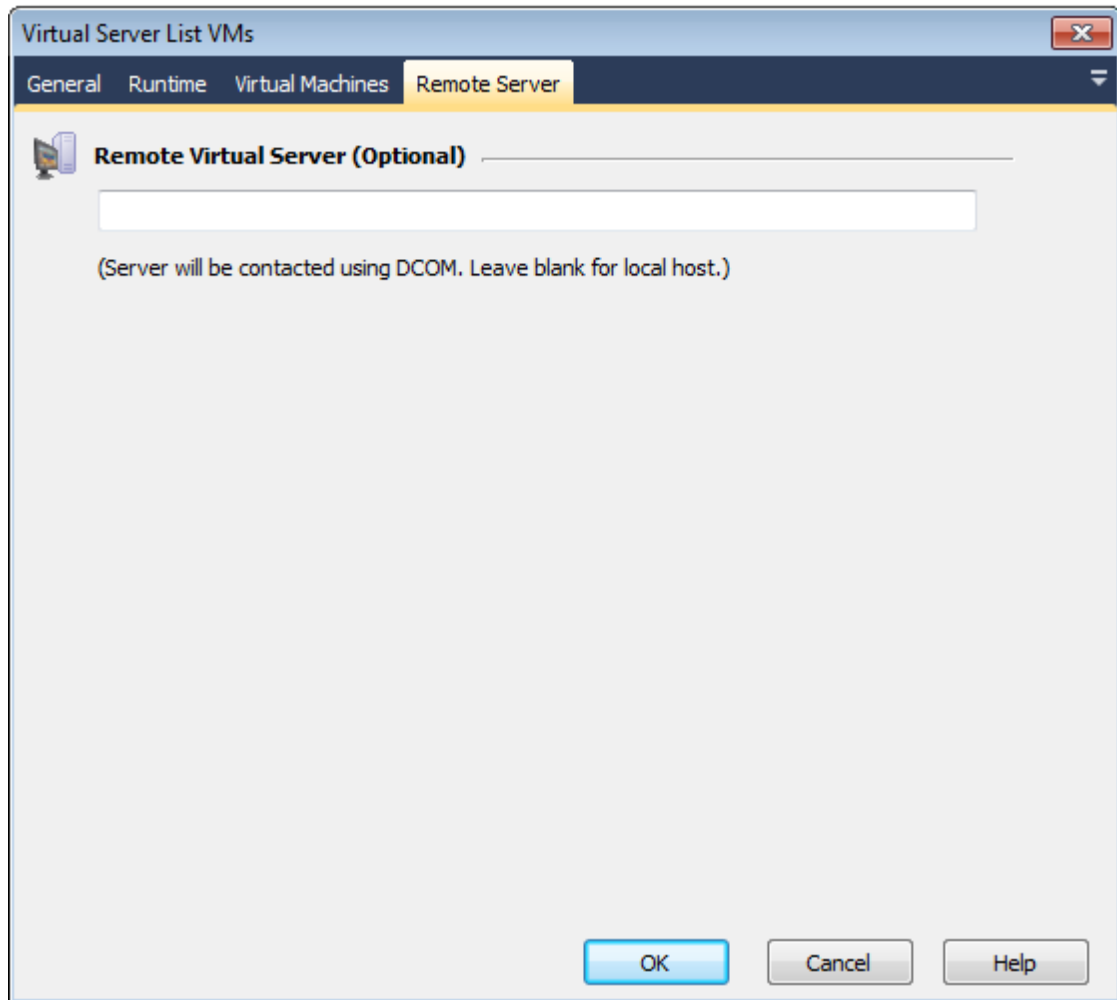


Virtual Machine States

Specify the states that you want included in the list.

Copy List to Variable

Optionally, specify a FinalBuilder variable to write the list of Virtual Machine names to. This list could then be used in a List Iterator.



If the virtual server is running on a remote machine, specify the host name here.

FinalBuilder will attempt to connect via DCOM, using the current running user's credentials.

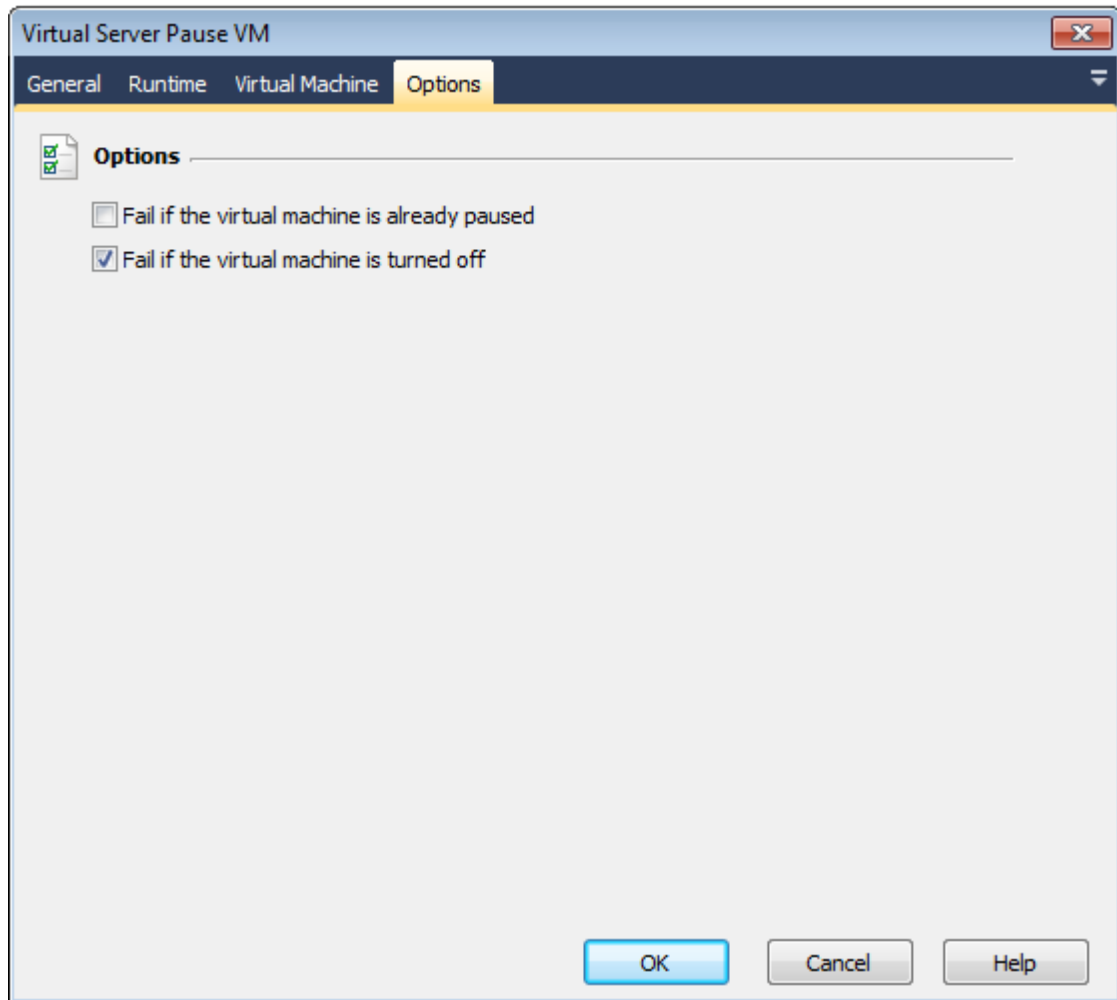
6.35.2.3 Virtual Server Pause VM

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.

**"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"

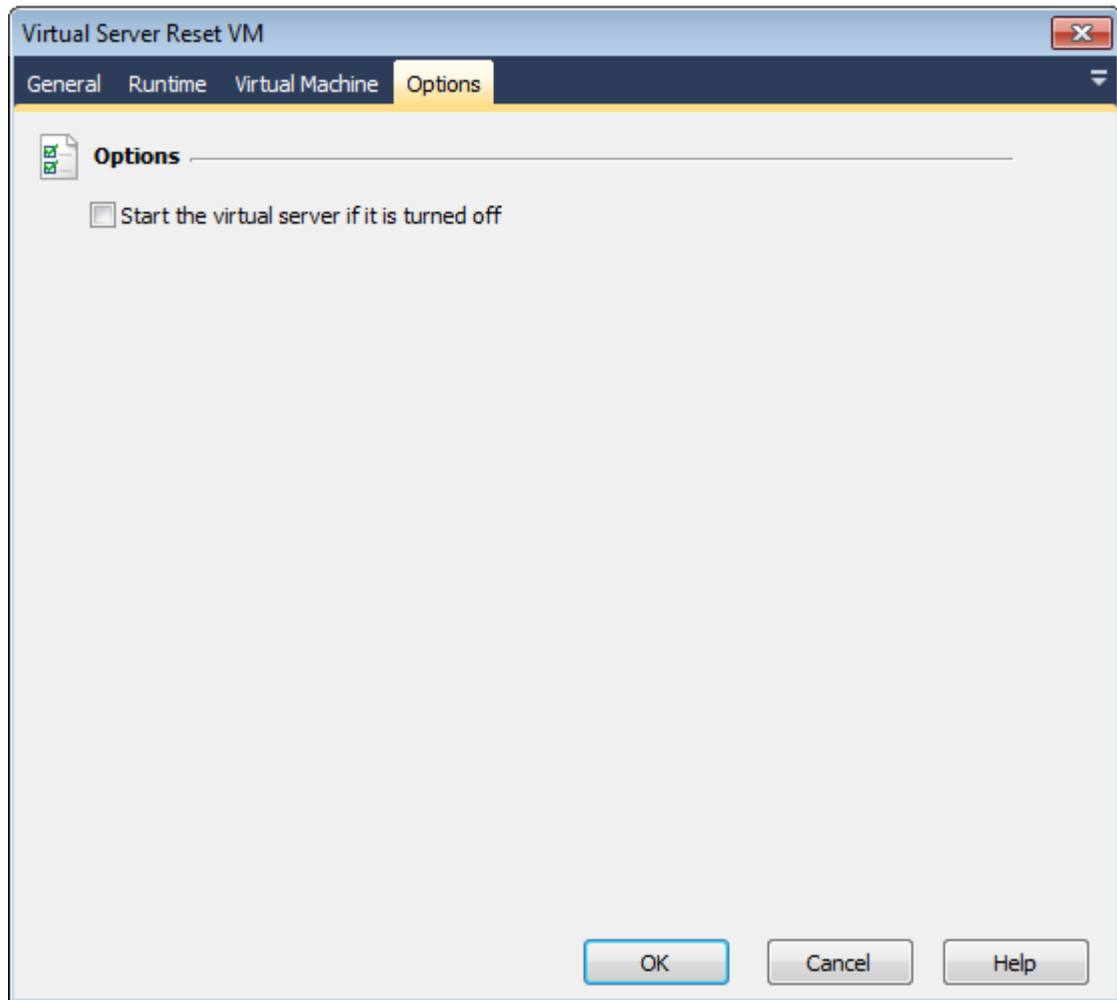
If this option is selected and the virtual machine is powered down, the action will fail. Otherwise, it will continue.

6.35.2.4 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.

**"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.

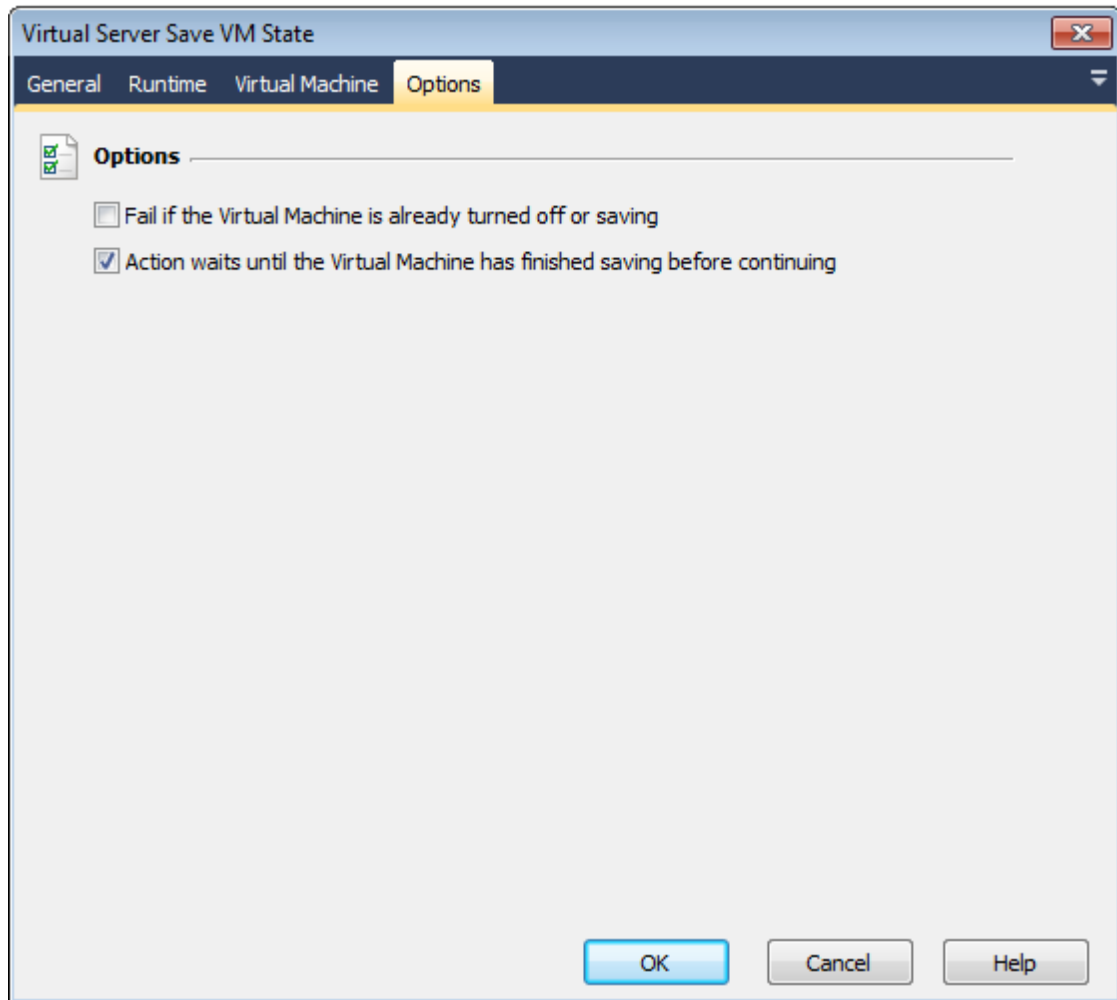
If this option is not selected and the Virtual Machine is turned off, the action will fail.

6.35.2.5 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.

**"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"

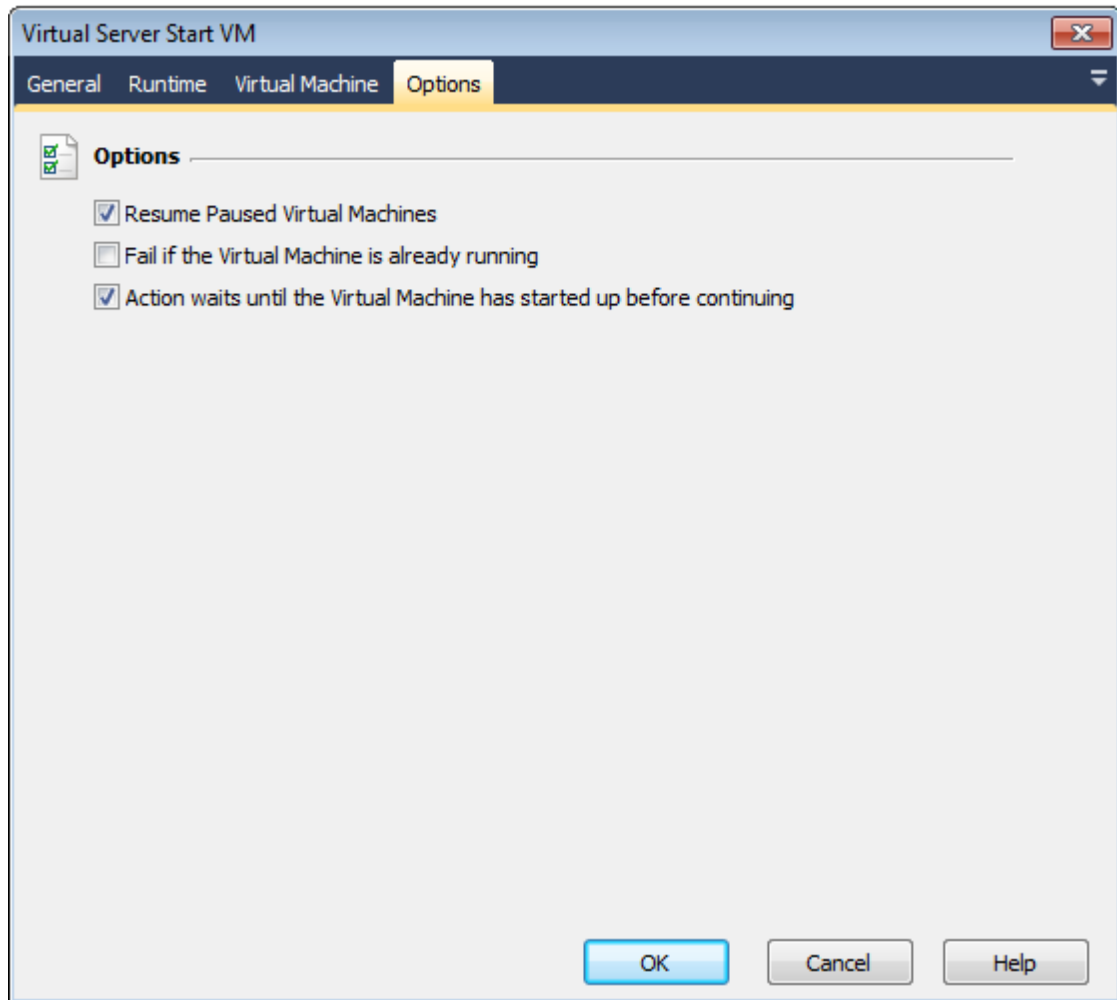
If this option is selected, the action will wait until the Virtual Machine has completed saving and powered down before continuing.

6.35.2.6 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.

**"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.)

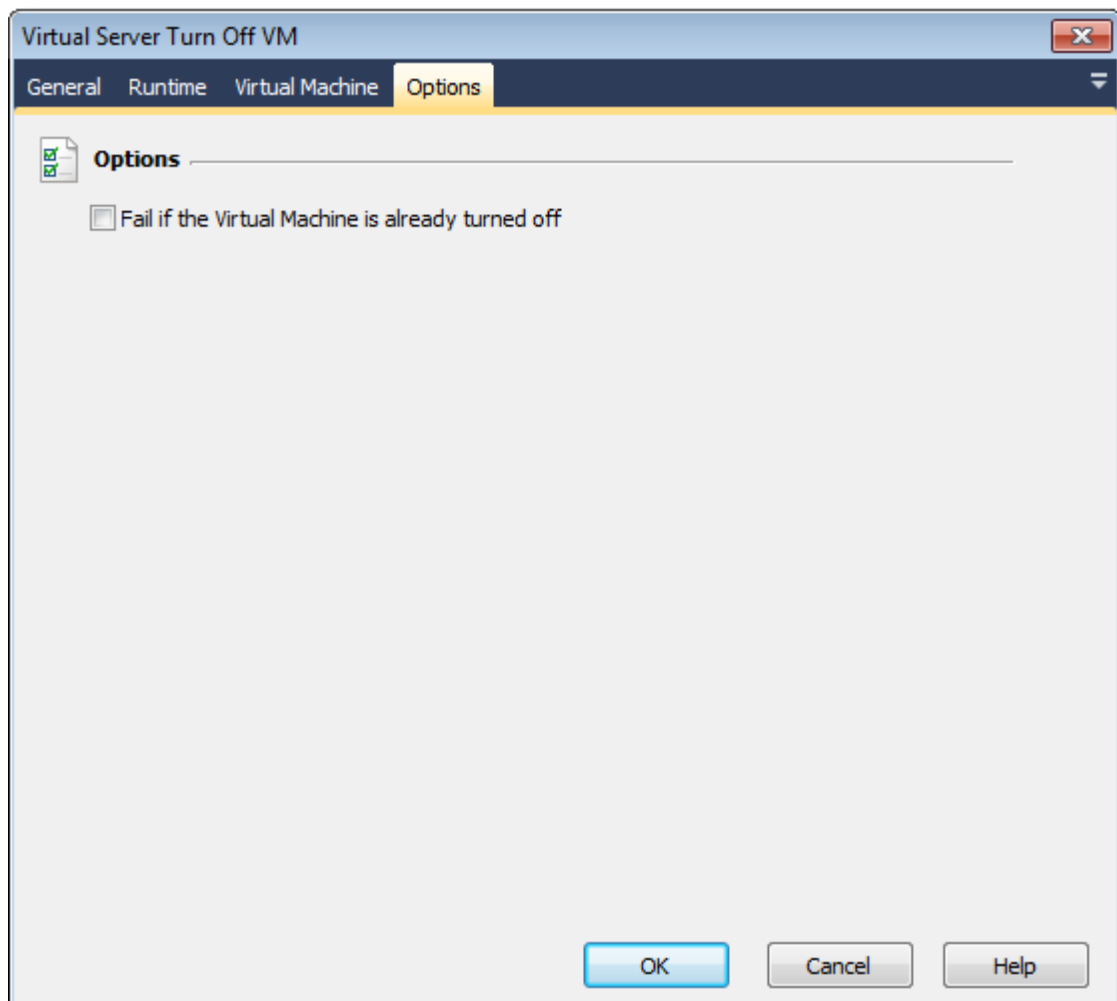
If this option is selected, the action will not continue until the Virtual Machine is up and running. If you are not accessing the Virtual Machine immediately, you can disable this option.

6.35.2.7 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"

If this option is selected and the VM is already powered down, the action will fail (otherwise it will continue.)

6.35.3 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

The screenshot shows a dialog box titled "VMWare Server Start VM" with a close button in the top right corner. The dialog has four tabs: "General", "Runtime", "Server" (which is selected and highlighted in yellow), and "Virtual Machine". The "Server" tab contains the following fields:

- Server Address (leave blank for local):** A text input field.
- Port (zero=default):** A numeric spinner box set to 0.
- Authentication:** A section header with a small icon of two people.
- Username:** A text input field.
- Password:** A text input field.

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

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.

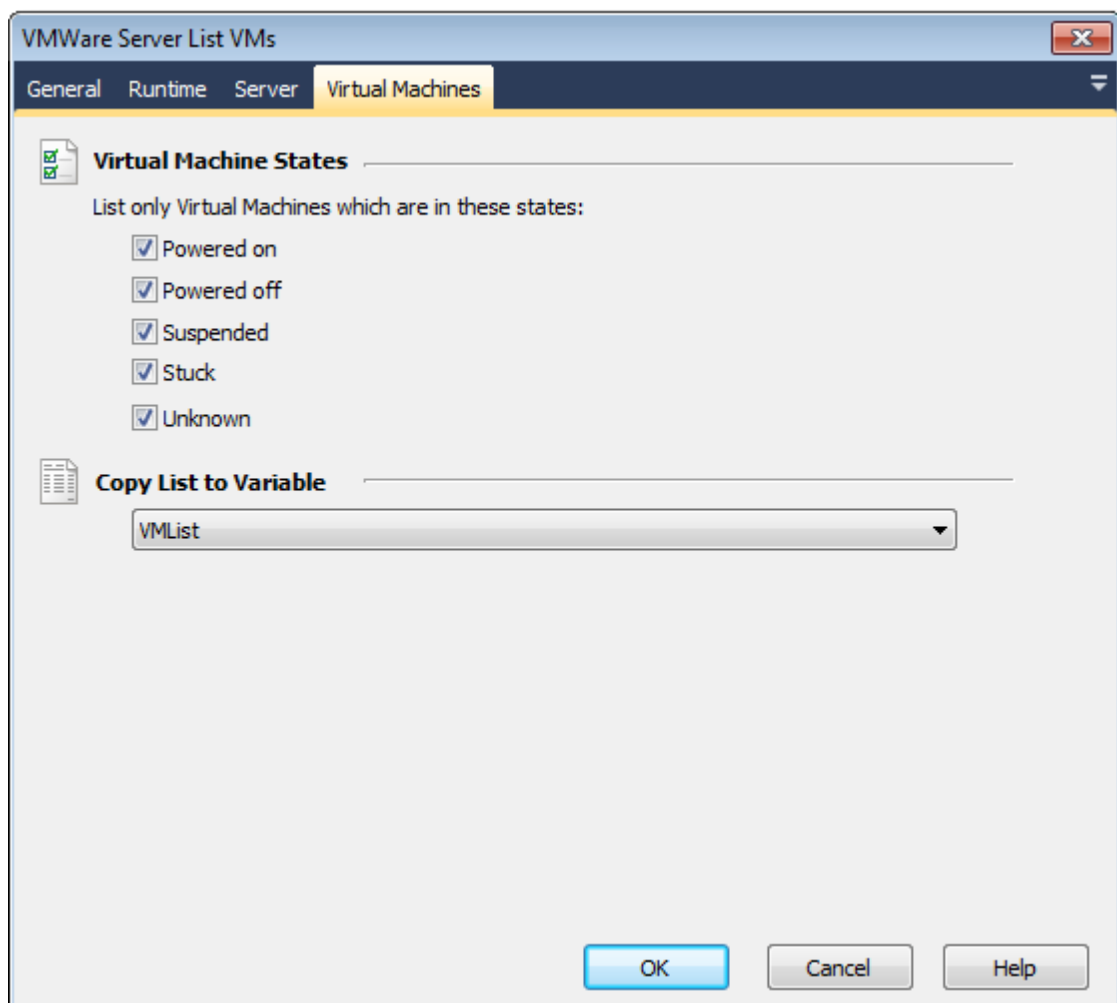
If left blank, the connection will use the credentials of the currently logged in user (recommended.)

6.35.3.1 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

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.

6.35.3.2 VMWare Server Start / Stop / Reset / Suspend VM

This group of VMWare actions modifies the state of a running or stopped Virtual Machine:

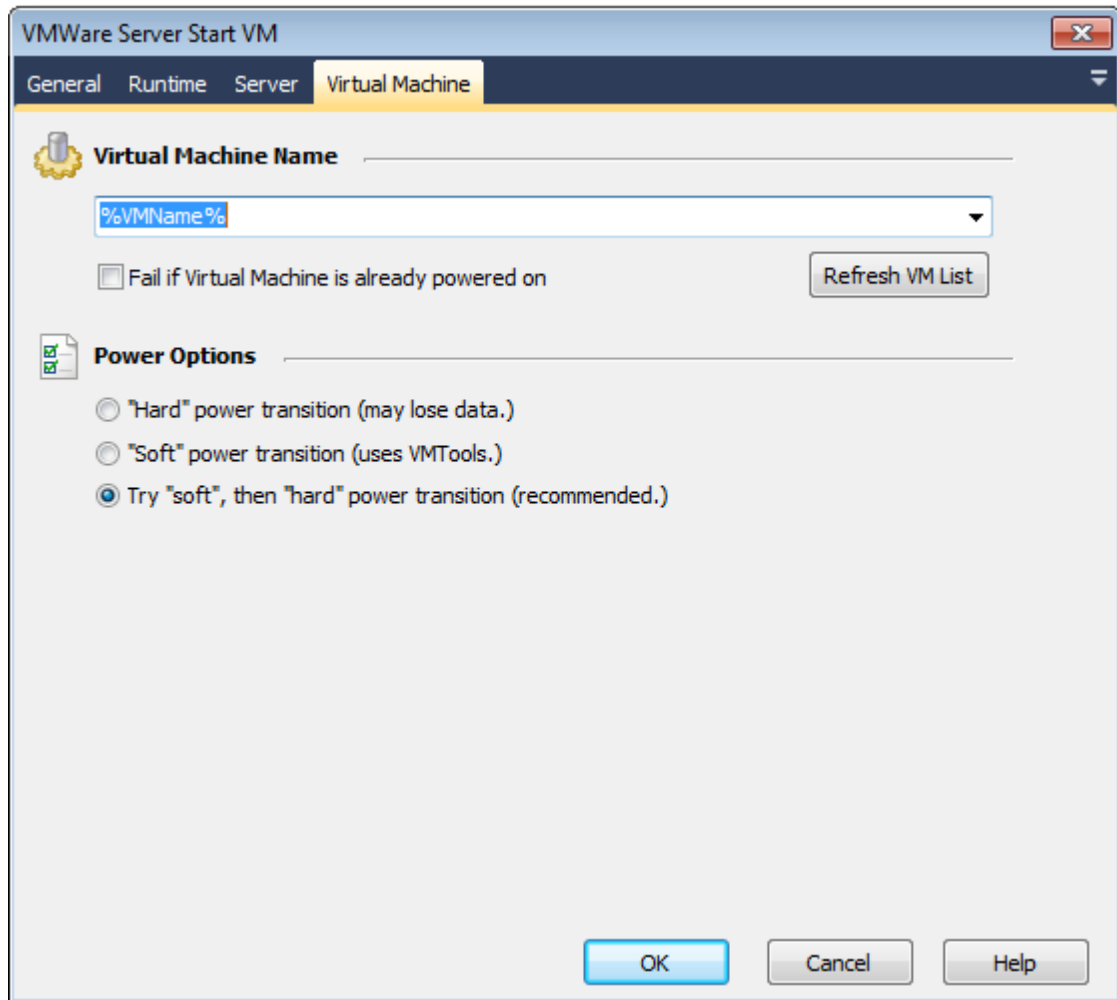
- **VMWare Server Start** starts a virtual machine which is suspended or powered off.
- **VMWare Server Stop** powers off a running virtual machine.
- **VMWare Server Reset** resets a running virtual machine.
- **VMWare Server Suspend** suspends a running virtual machine.

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

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.)

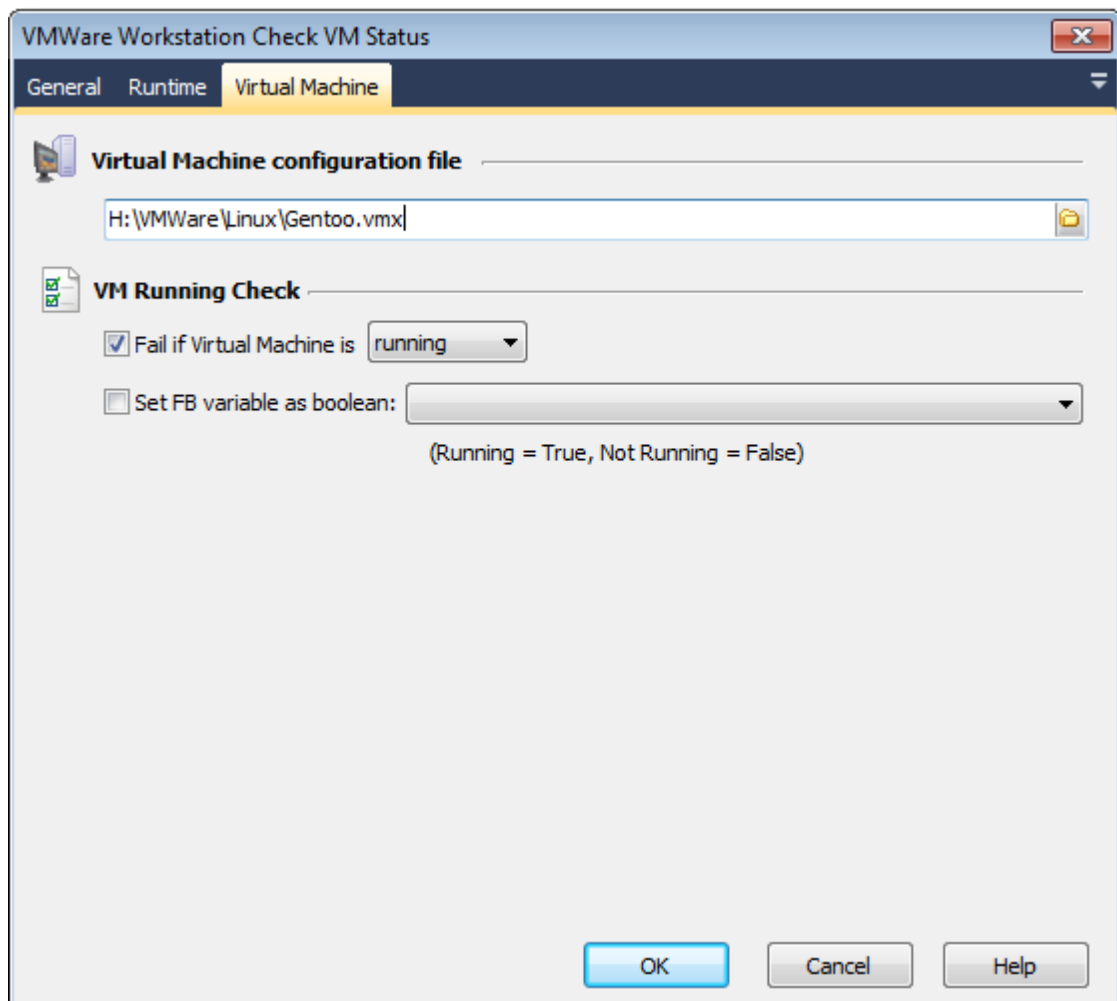
By default, the VMWare Server actions will try a safer soft operation, but revert to a hard operation if the soft operation fails.

6.35.4 VMWare Workstation

6.35.4.1 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.



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"

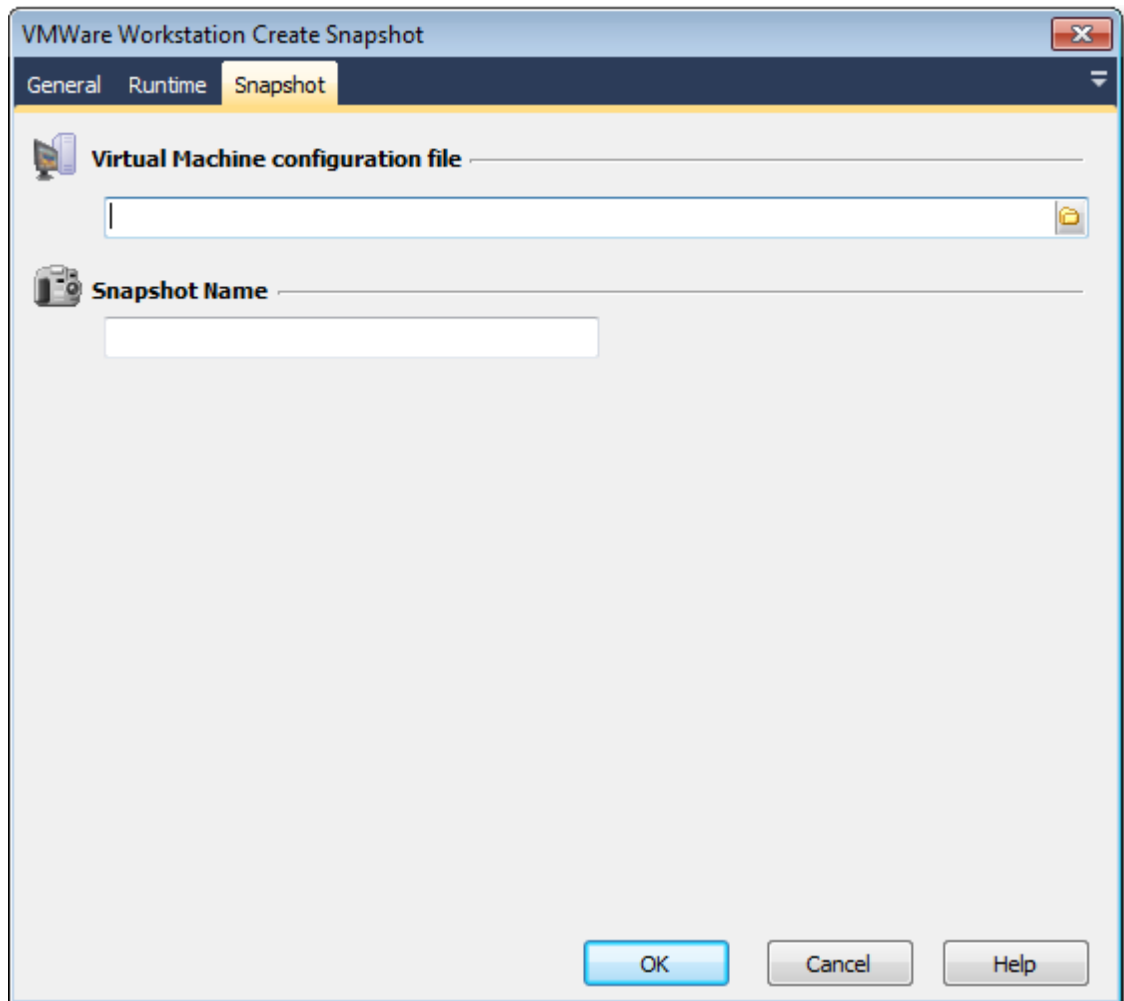
If this box is checked, the action will fail if the VM is either running or not running.

"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.

6.35.4.2 Create Snapshot

The VMWare Create Snapshot action creates a snapshot of a currently running or stopped Virtual Machine.



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.

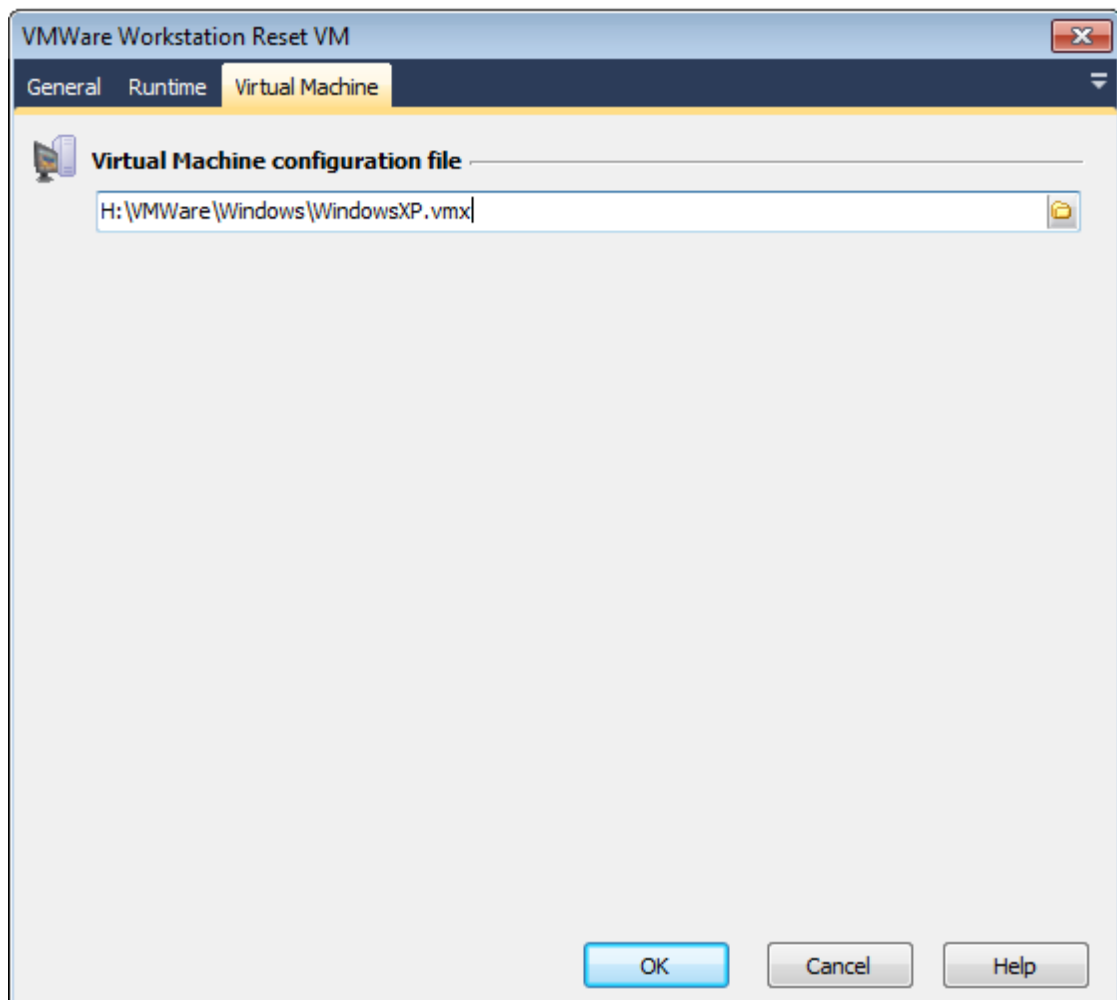
NOTE: In recent versions of VMWare, a hint is displayed when creating background snapshots. If the hint dialog is displayed, it will delay an automated snapshot until OK is pressed. If the build is running unattended, it is advisable to check if this hint is

displayed and check the "Do Not Display Again" option if it is so.

6.35.4.3 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.



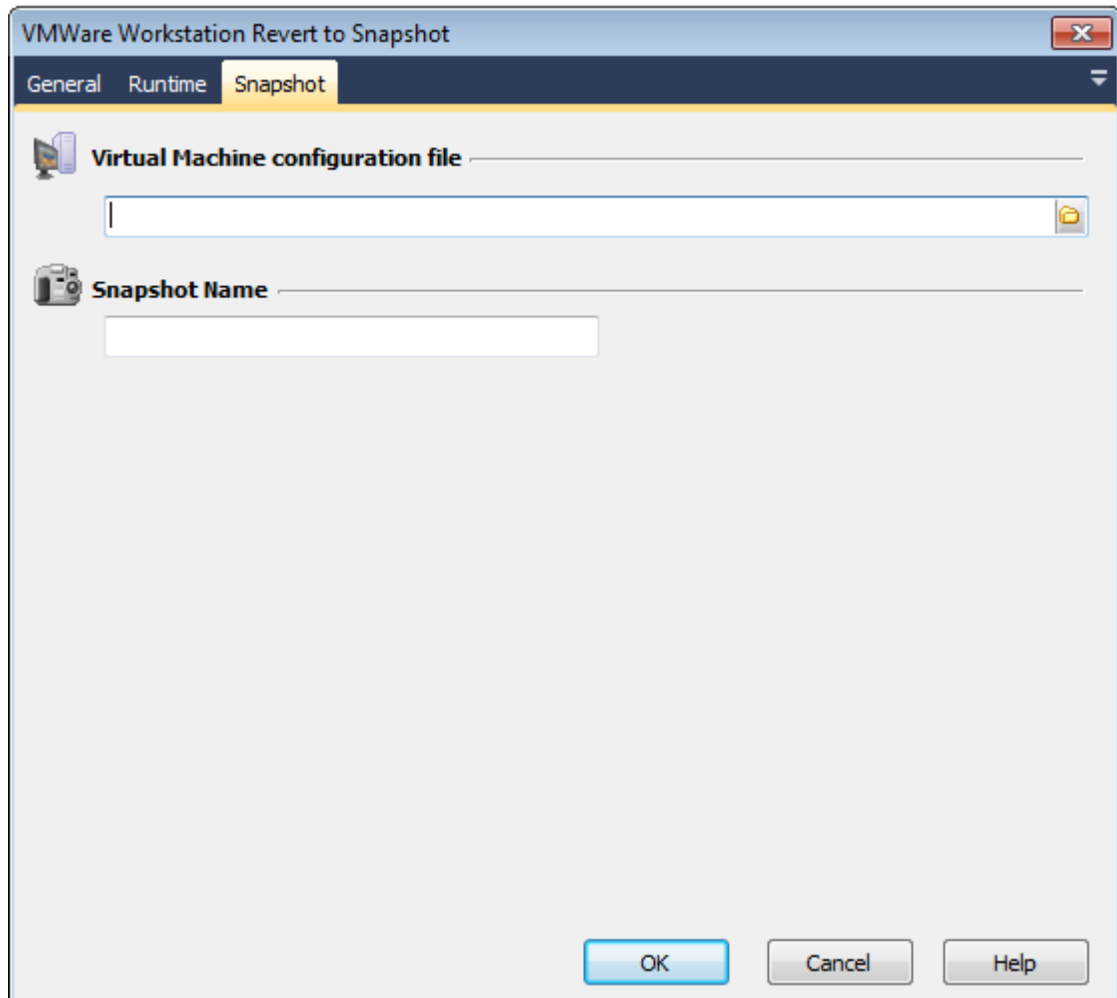
Virtual Machine Configuration File

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).

6.35.4.4 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

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

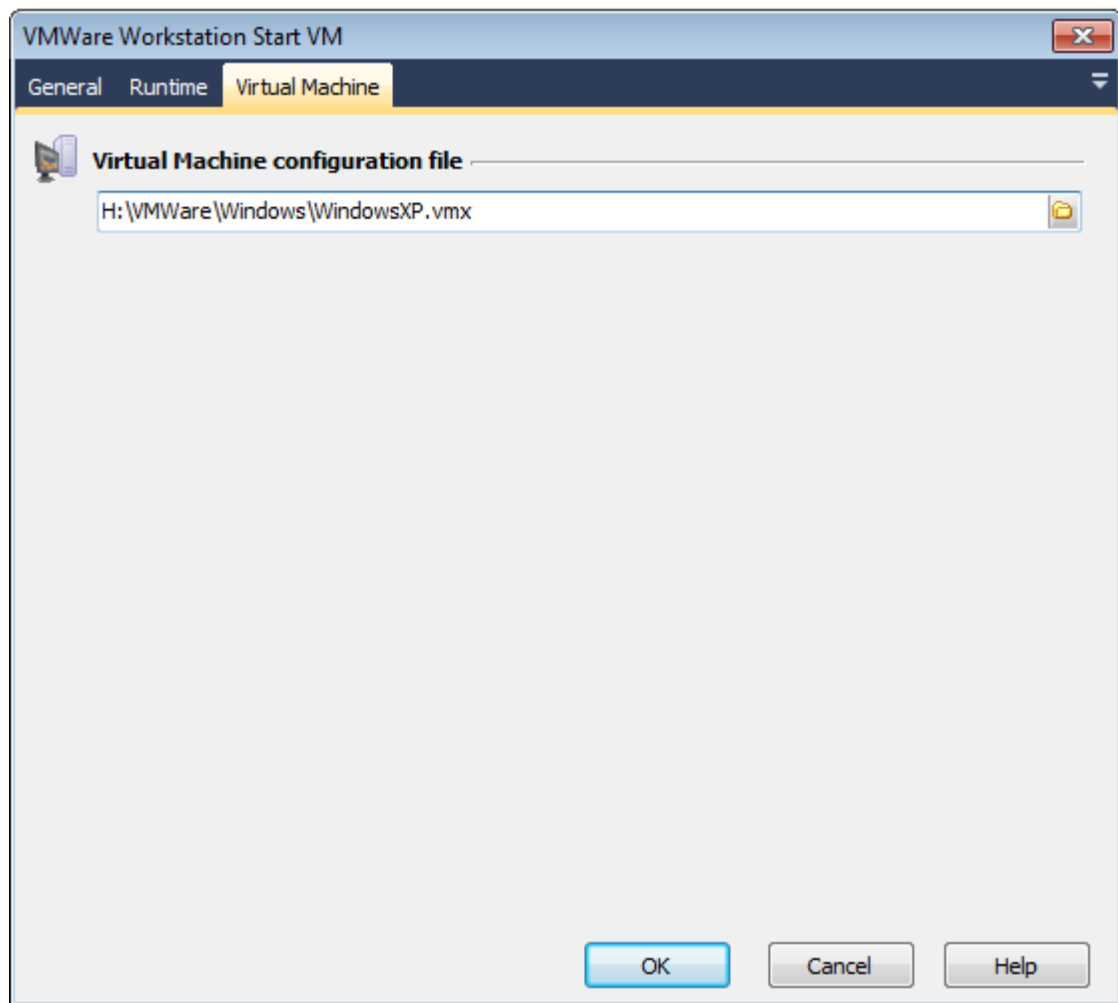
The identifying name for the snapshot to restore to.

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.

6.35.4.5 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

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.

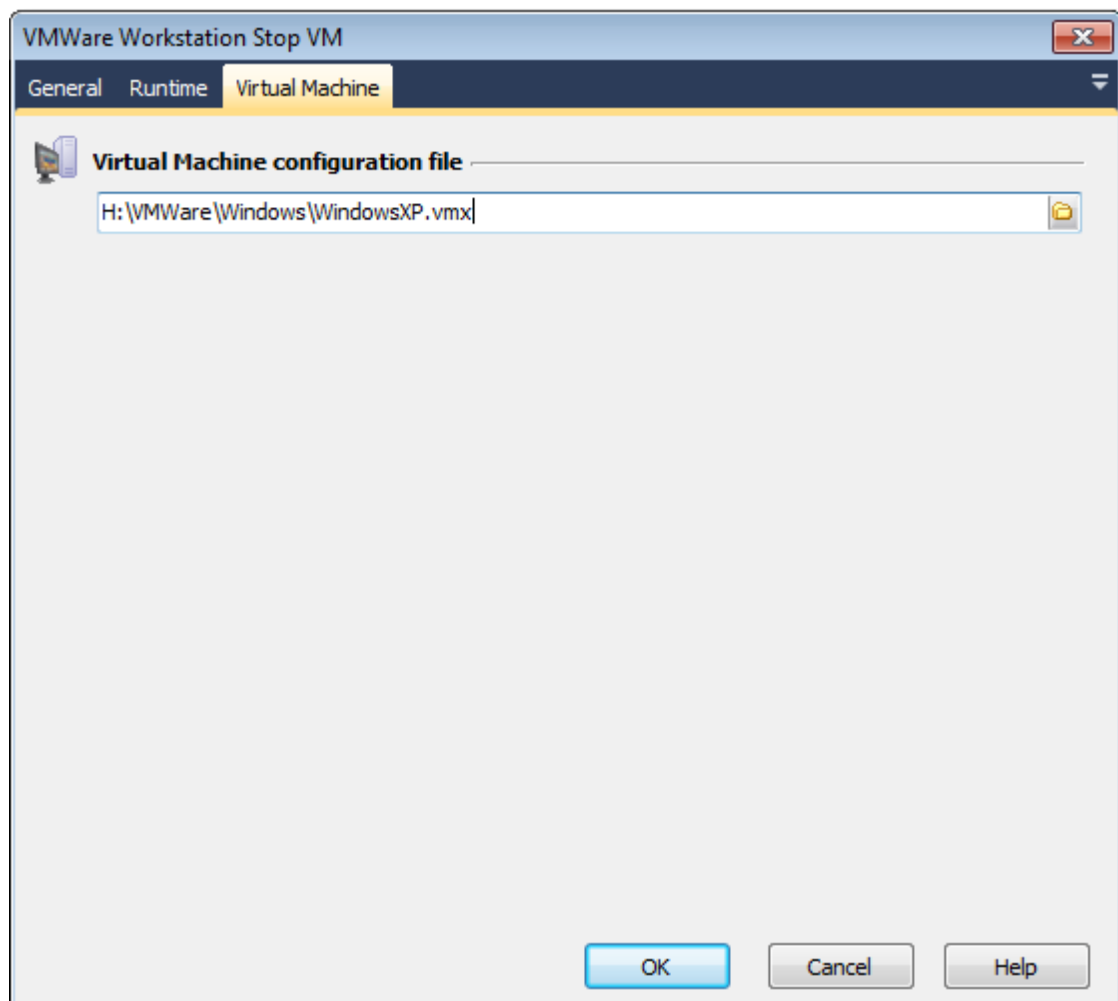
The Virtual Machine must not be already running in a VMWare workstation instance.

6.35.4.6 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.

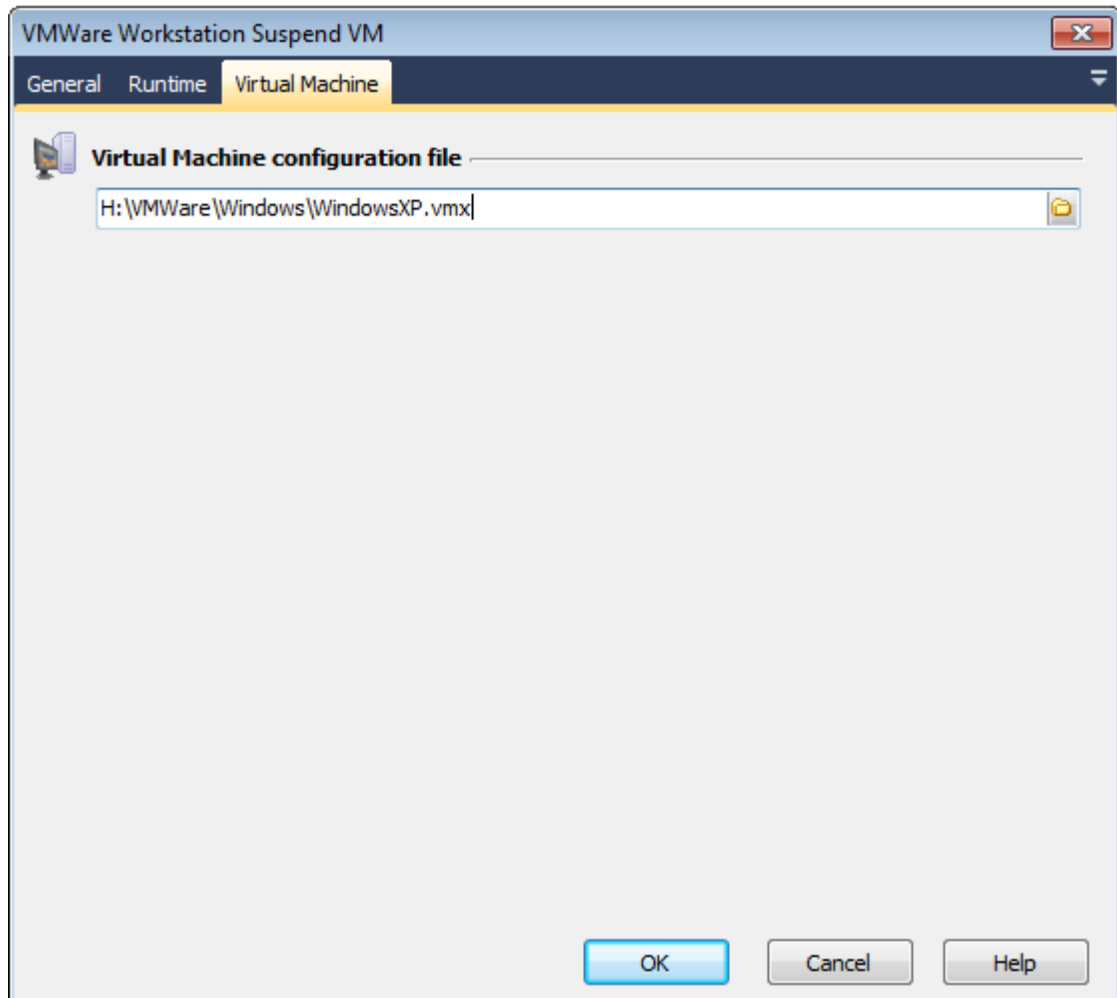


Virtual Machine Configuration File

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).

6.35.4.7 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.

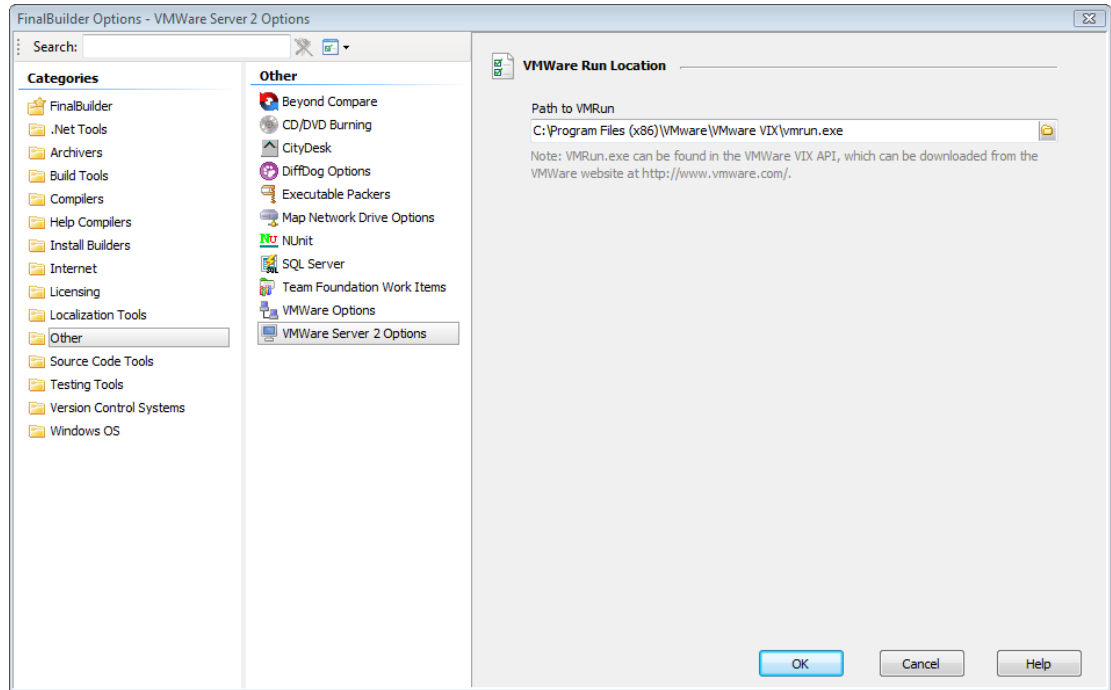


Virtual Machine Configuration File

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).

6.35.5 VMWare

The VMWare actions use the VMRun command line utility to control virtual machines either on a VMWare Server installation or VMWare Workstation.



Connection Options

The screenshot shows a dialog box titled "VMWare - List Virtual Machines" with a close button in the top right corner. It has four tabs: "General", "Runtime", "Connection" (which is selected and highlighted in yellow), and "List Options".

Under the "Connection" tab, there are two main sections:

- Connection Options**: Indicated by a checkmark icon. It contains:
 - Host Type**: A dropdown menu currently showing "VMWare Server 2".
 - Host Name**: A text input field. Below it is a note: "Leave blank for localhost, use format 'https://<IP Address>:<port>/sdk' for VMWare Server 2."
 - Host Port**: A spinner box showing the value "0". Below it is a note: "Default port is 902, use 0 for localhost; ignored when connecting to VMWare Server 2."
- Authentication**: Indicated by a person icon. It contains:
 - Username**: A text input field. Below it is a note: "Leave blank to authenticate as current user when connecting to the local machine."
 - Password**: A text input field.

At the bottom right of the dialog are three buttons: "OK", "Cancel", and "Help".

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

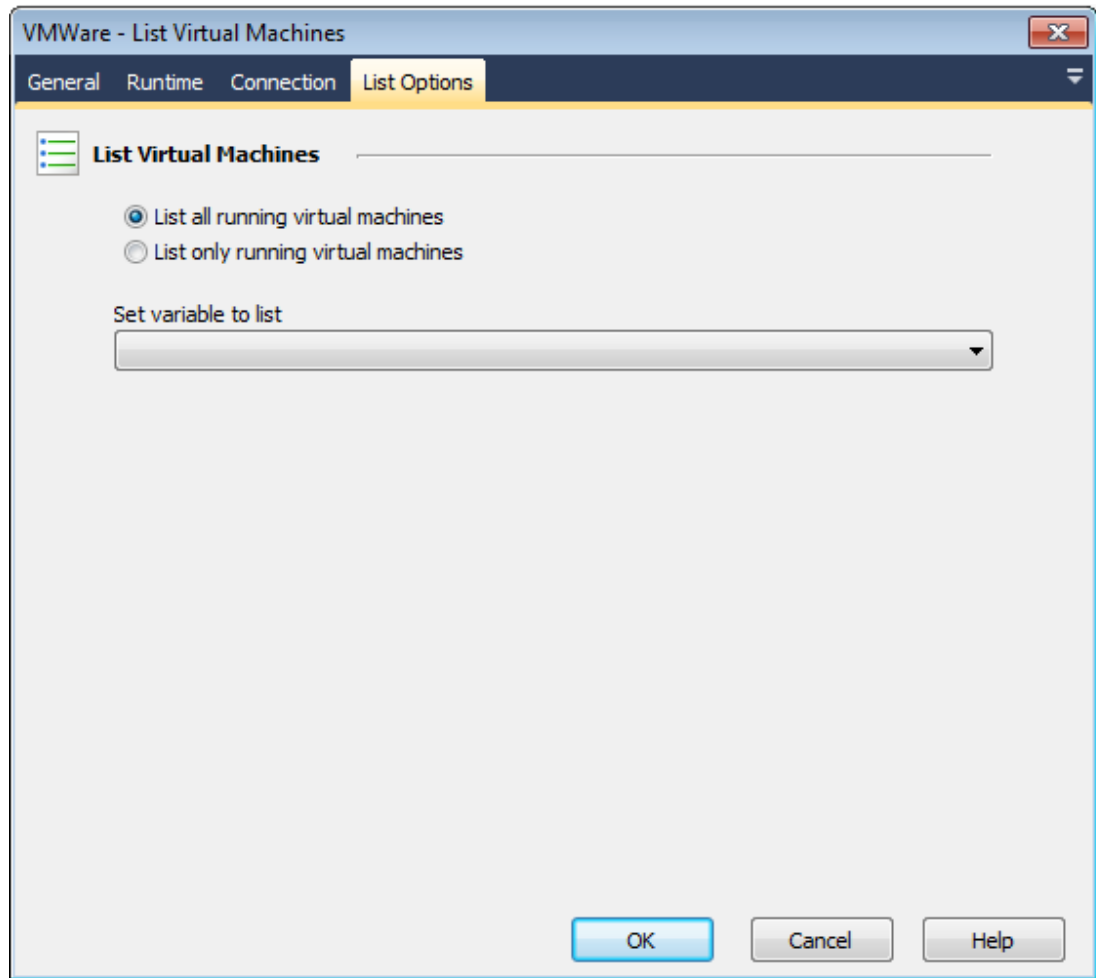
The name of the machine where the VMWare host is located, use *localhost* if you are connecting to an instance on the local machine.

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.

6.35.5.1 List Virtual Machine

VMWare - List Virtual Machine action enables you to list the virtual machines registered on the host.

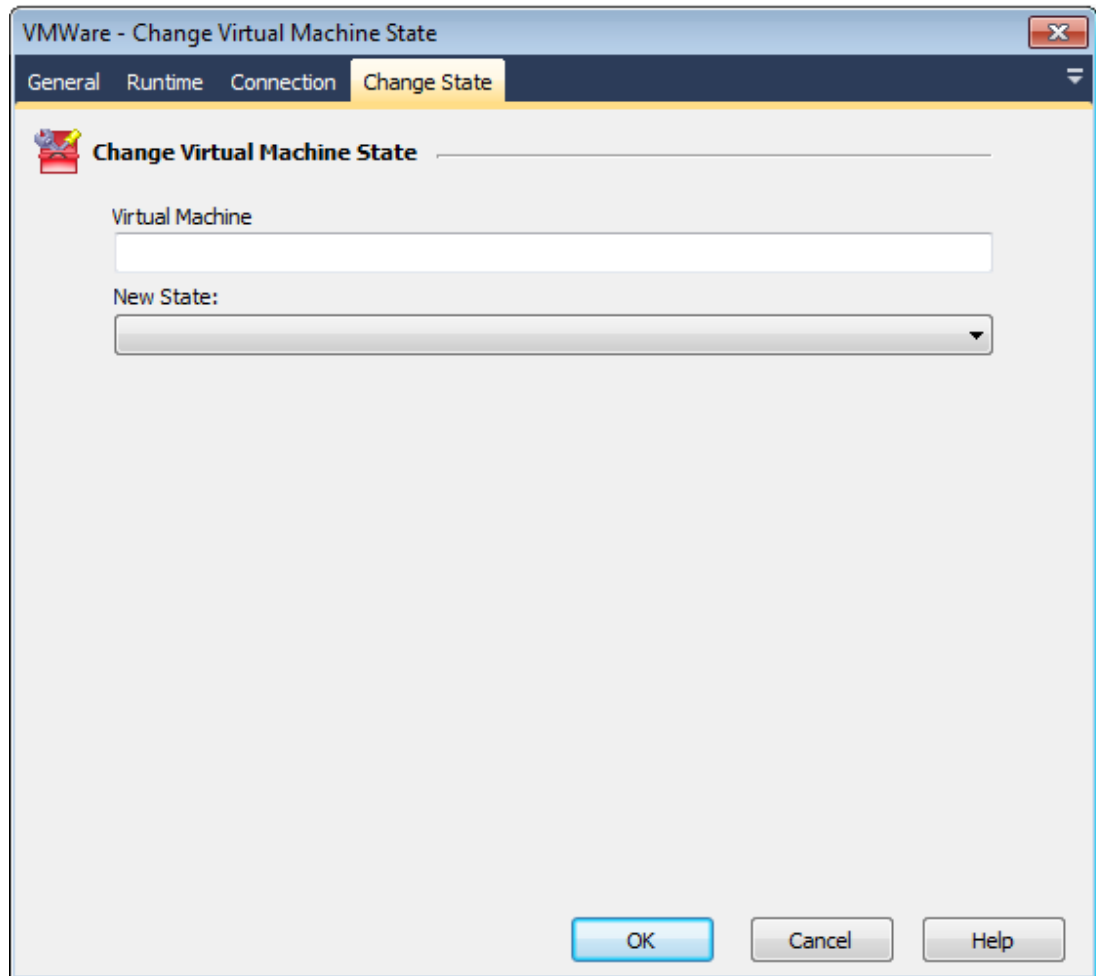


Options

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.

6.35.5.2 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.

**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.

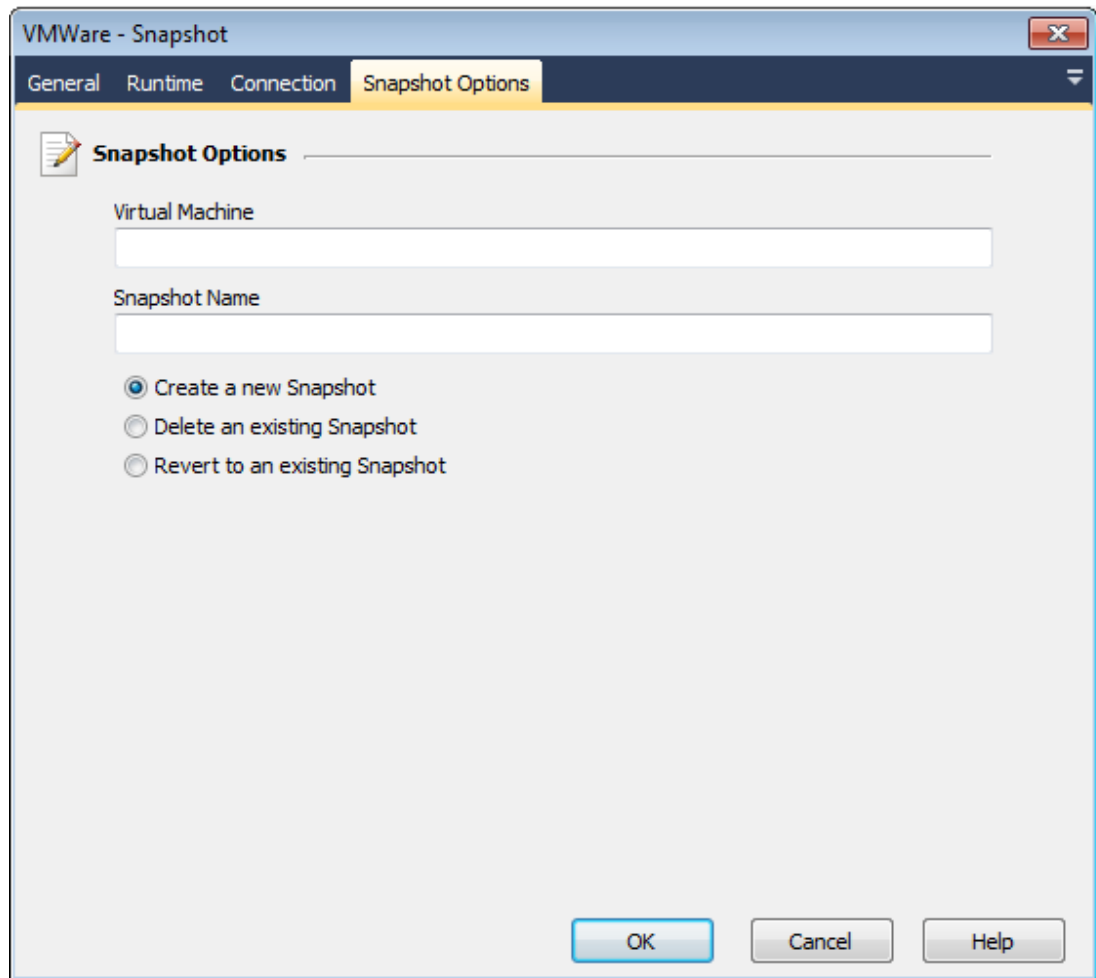
New State

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.

6.35.5.3 Snapshot

Allows control of the snapshots of a Virtual Machine.



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.

6.35.5.4 Execute Guest Command

Enables a guest command to be executed on the virtual machine using the VMWare Guest Utility.

The screenshot shows the 'VMWare - Execute Guest Command' dialog box. It has a title bar with a close button. Below the title bar are four tabs: 'General', 'Runtime', 'Connection', and 'Guest Options' (which is selected). The 'Guest Options' tab contains two main sections: 'Guest Credentials' and 'Guest Command'. The 'Guest Credentials' section has three input fields: 'Virtual Machine', 'Username', and 'Password'. The 'Guest Command' section has a 'Command' dropdown menu, an 'Arguments' table with columns 'Argument Name' and 'Value', and a 'Set Variable to Output' dropdown menu. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

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

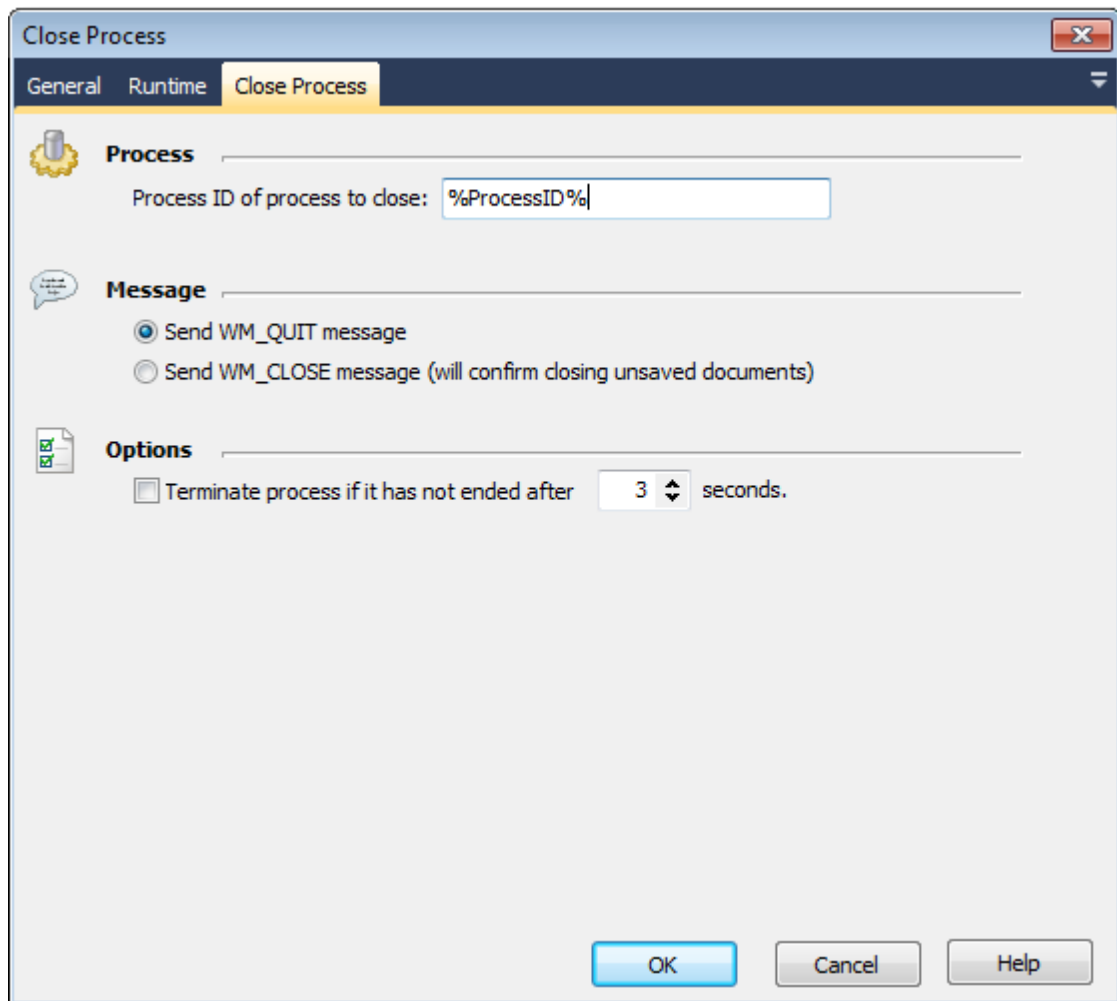
The variable to store the output of the guest command.

6.36 Windows OS

6.36.1 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

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..."

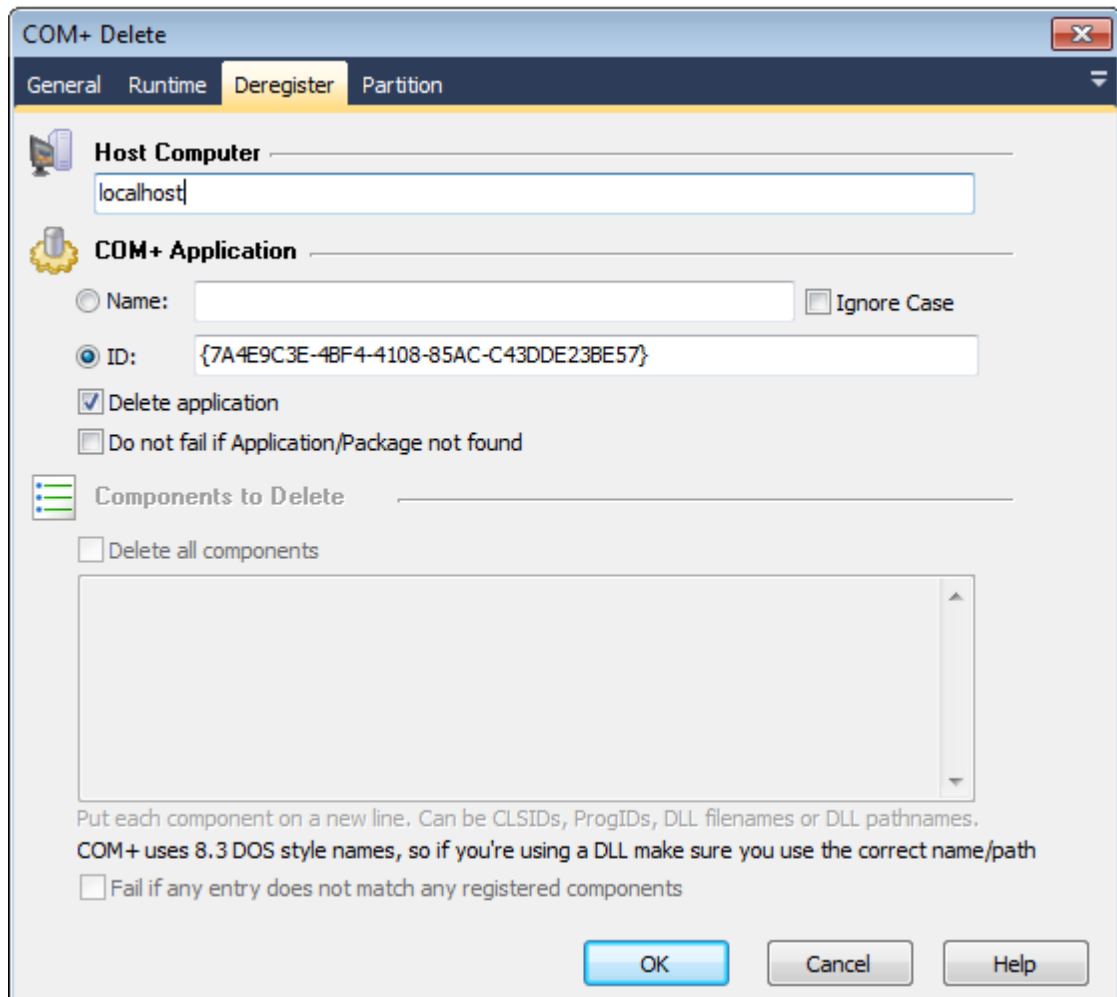
This option sets a timeout after which a process is forcibly terminated. This is useful if you would like to terminate a possibly unstable process, or a process which has paused waiting for confirmation of unsaved data (see WM_CLOSE in the Message section.)

If this option is not checked, the action will not pause to confirm whether or not the process has ended.

6.36.2 COM+ Administration

6.36.2.1 COM+ Delete

The COM+ Delete Action allows you to delete registered COM+ applications or COM+ 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.

6.36.2.2 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.

COM+ Edit Application

General Runtime **COM+ Application** Application Type Options

Host Computer _____
localhost

COM+ Application _____
☒ Name: My Application|
☐ ID: _____

Partition _____
(Windows 2003 only. Can be specified as an ID or a Name. Leave blank for the base partition.)

OK Cancel Help

Host Computer

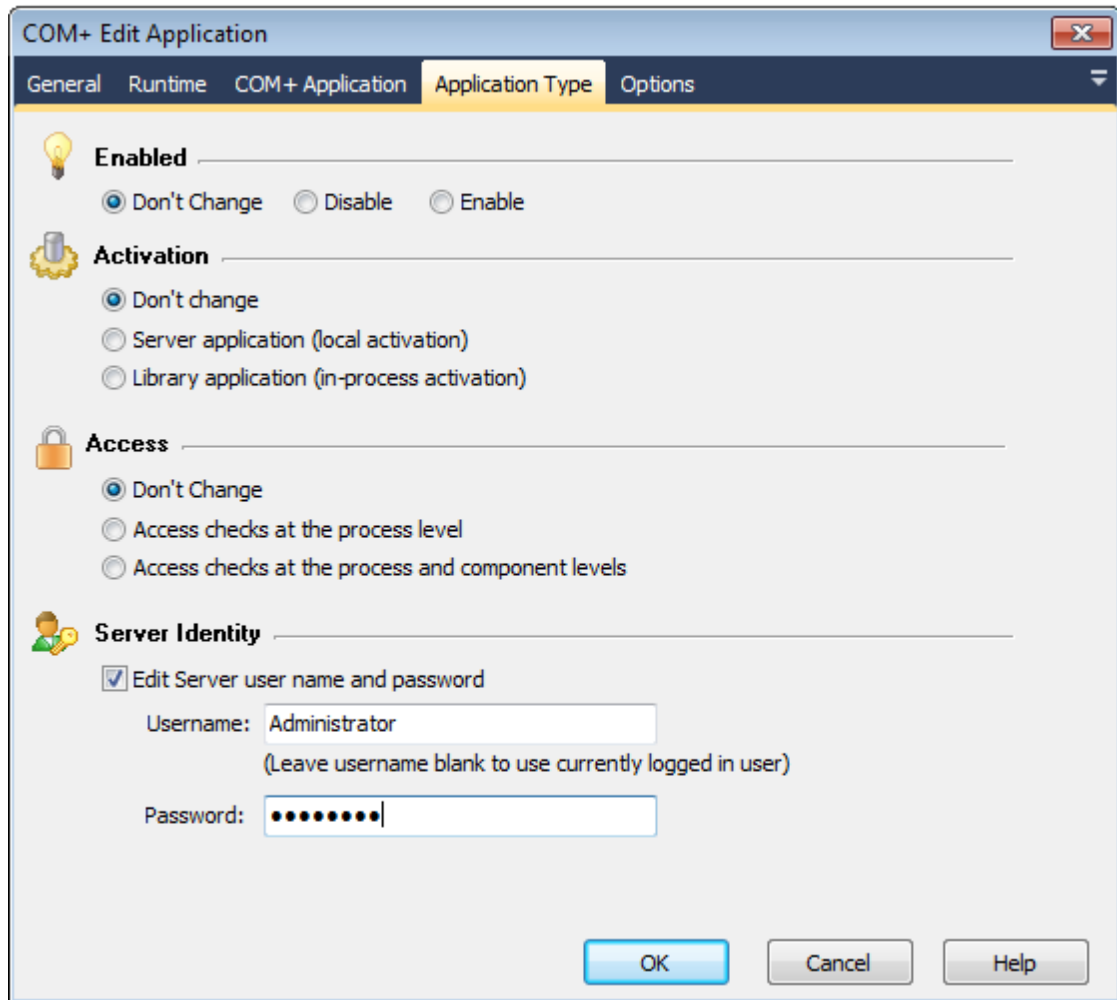
The computer you want to edit the Application on. You will need Administrator privileges on this computer. Blank is equivalent to localhost.

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.



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.

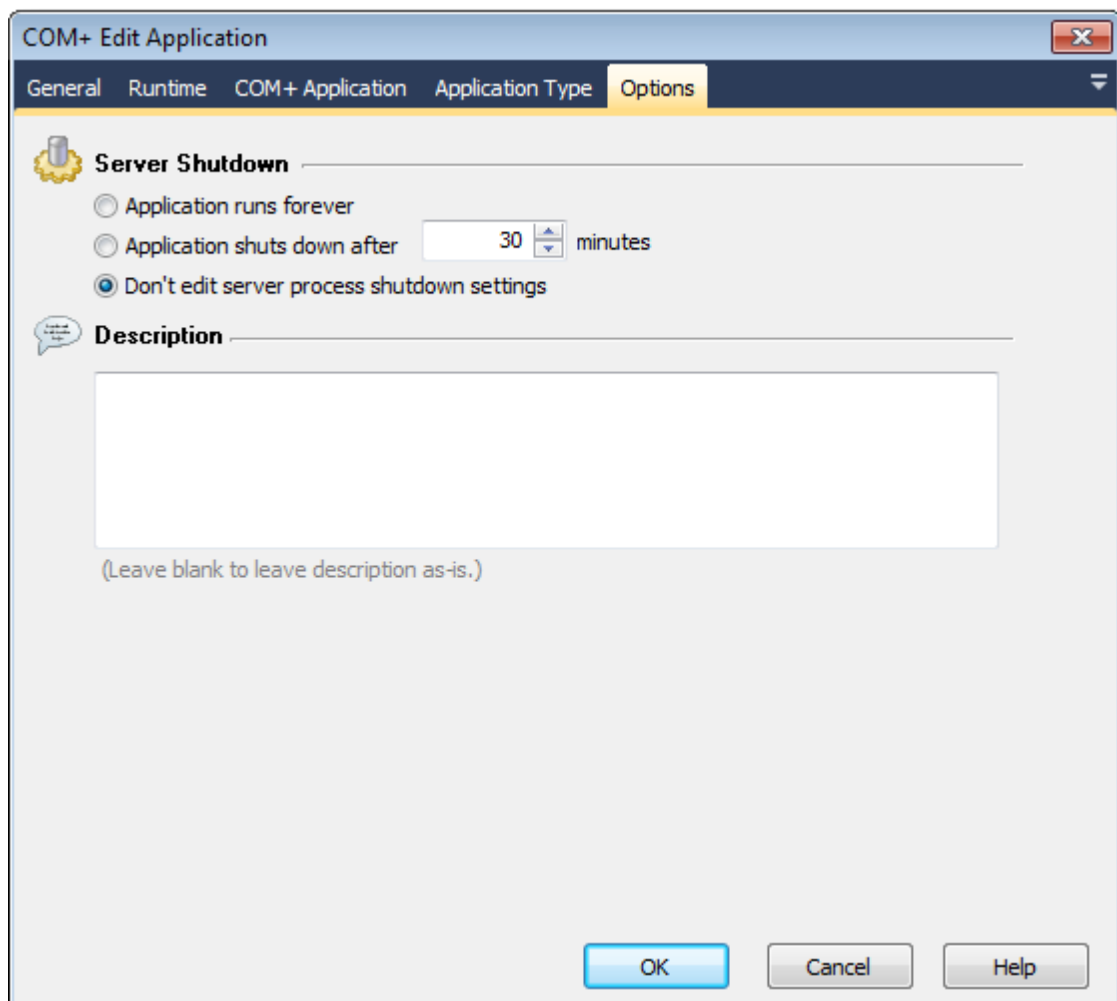
Access

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.



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.

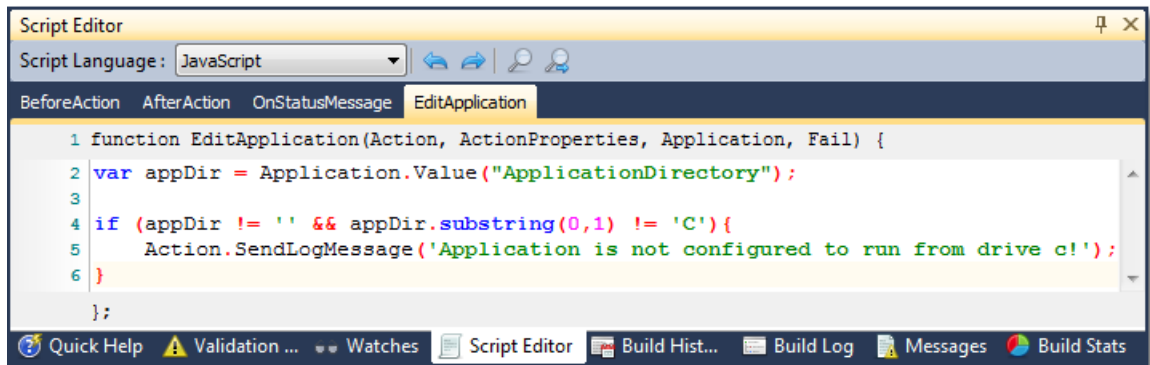
Description

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:



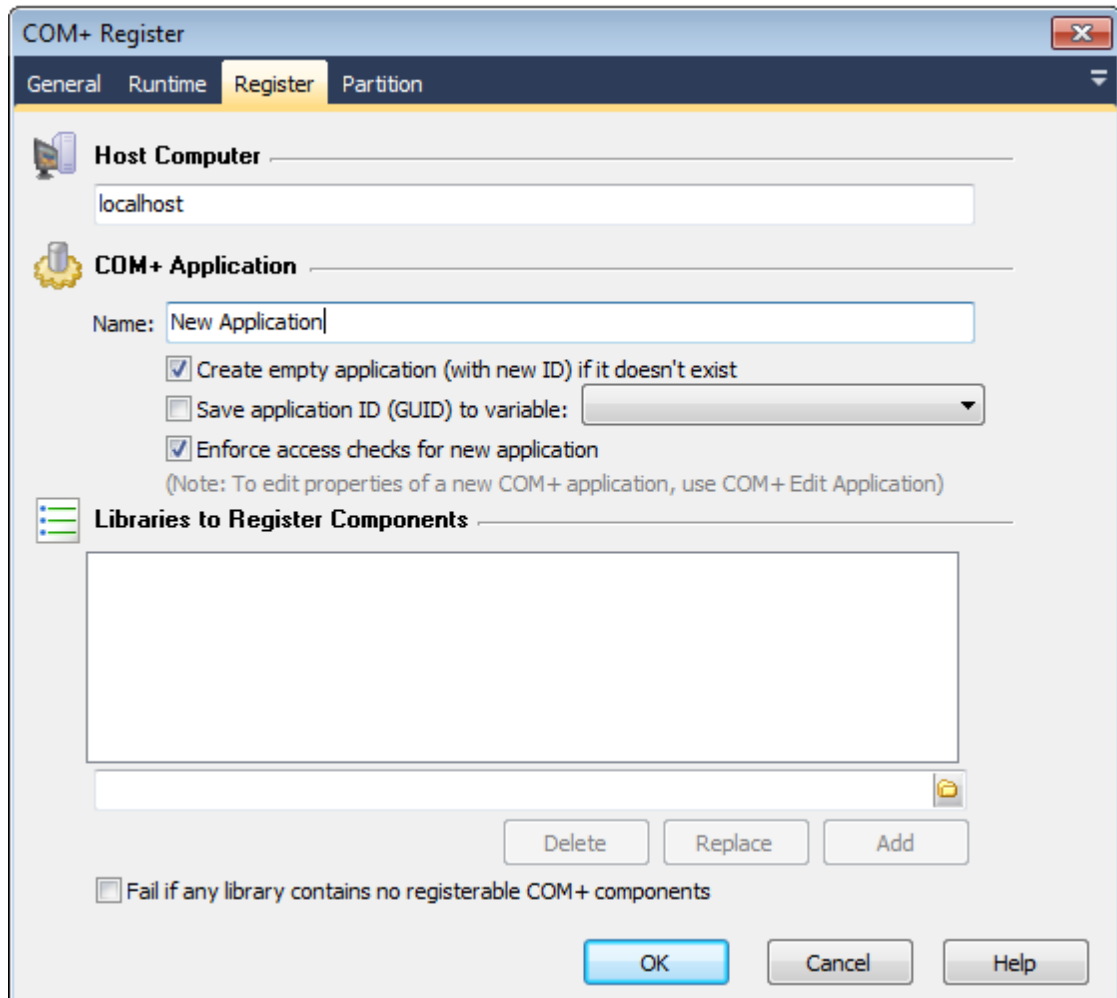
To read and write Application properties, use the syntax `Application.Value("PropertyName")`. Set the Fail parameter to true to force the action to fail programmatically.

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.

6.36.2.3 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

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 FinalBuilder 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

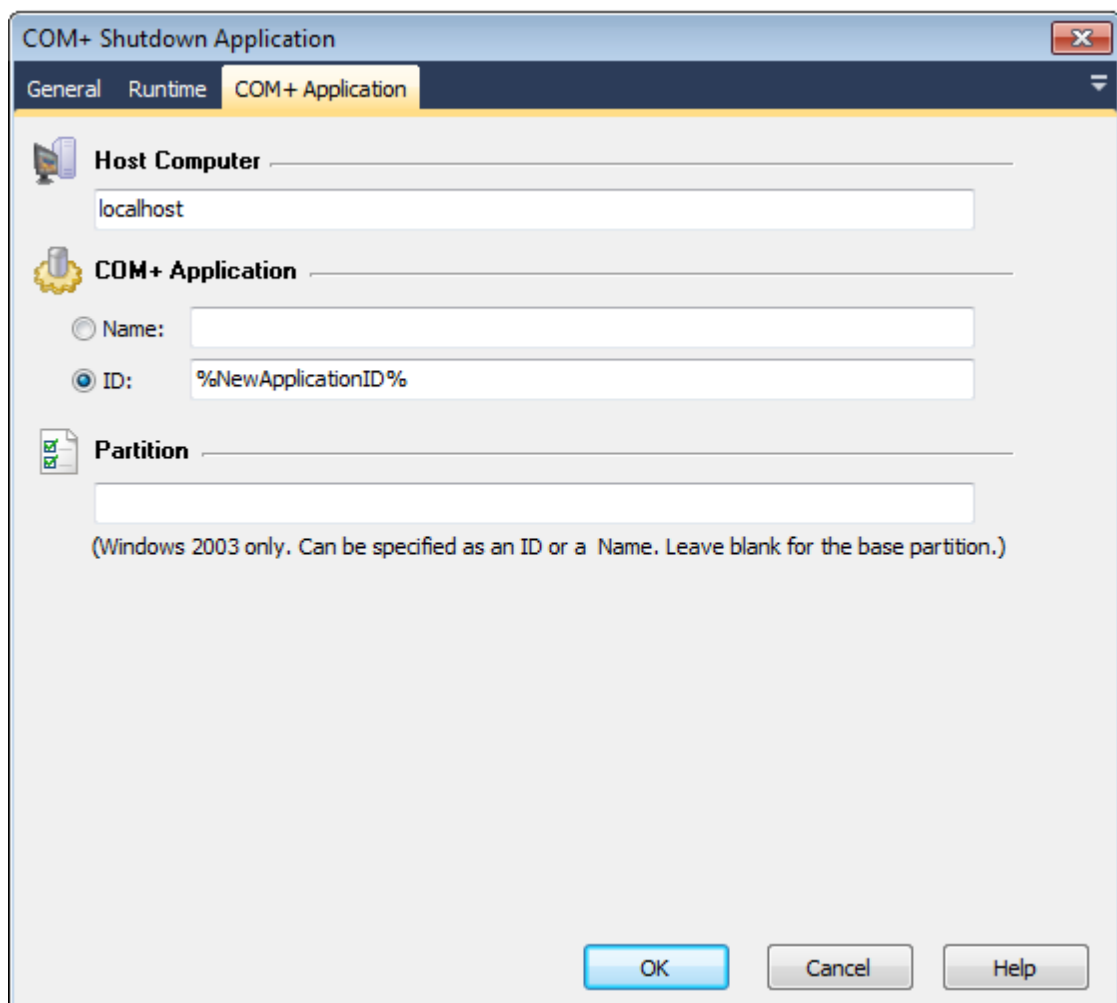
Add the shared libraries that you wish to load COM+ components from. Check the **"Fail if any library contains no registerable COM+ components"** to have the action fail if any DLL does not contain any components.

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.

6.36.2.4 COM+ Shutdown Application

The COM+ Shutdown Application action allows you to shut down a running COM+ Application.



The screenshot shows the "COM+ Shutdown Application" dialog box with three tabs: "General", "Runtime", and "COM+ Application". The "COM+ Application" tab is selected. It contains three sections: "Host Computer" with a text field containing "localhost"; "COM+ Application" with radio buttons for "Name:" and "ID:", where "ID:" is selected and the text field contains "%NewApplicationID%"; and "Partition" with an empty text field. A note below the Partition field states: "(Windows 2003 only. Can be specified as an ID or a Name. Leave blank for the base partition.)". At the bottom are "OK", "Cancel", and "Help" buttons.

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 FinalBuilder 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.

6.36.2.5 COM+ Start Application

The COM+ Start Application action allows you to start a shut down or inactive COM+ Application.

The screenshot shows the 'COM+ Start Application' dialog box with the 'COM+ Application' tab selected. The 'Host Computer' field contains 'localhost'. The 'COM+ Application' section has two radio buttons: 'Name' (selected) with the text 'My Application' in the adjacent field, and 'ID' (unselected) with an empty field. The 'Partition' section has an empty text field. A note below the partition field states: '(Windows 2003 only. Can be specified as an ID or a Name. Leave blank for the base partition.)'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Field	Value
Host Computer	localhost
COM+ Application Name	My Application
COM+ Application ID	
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.)

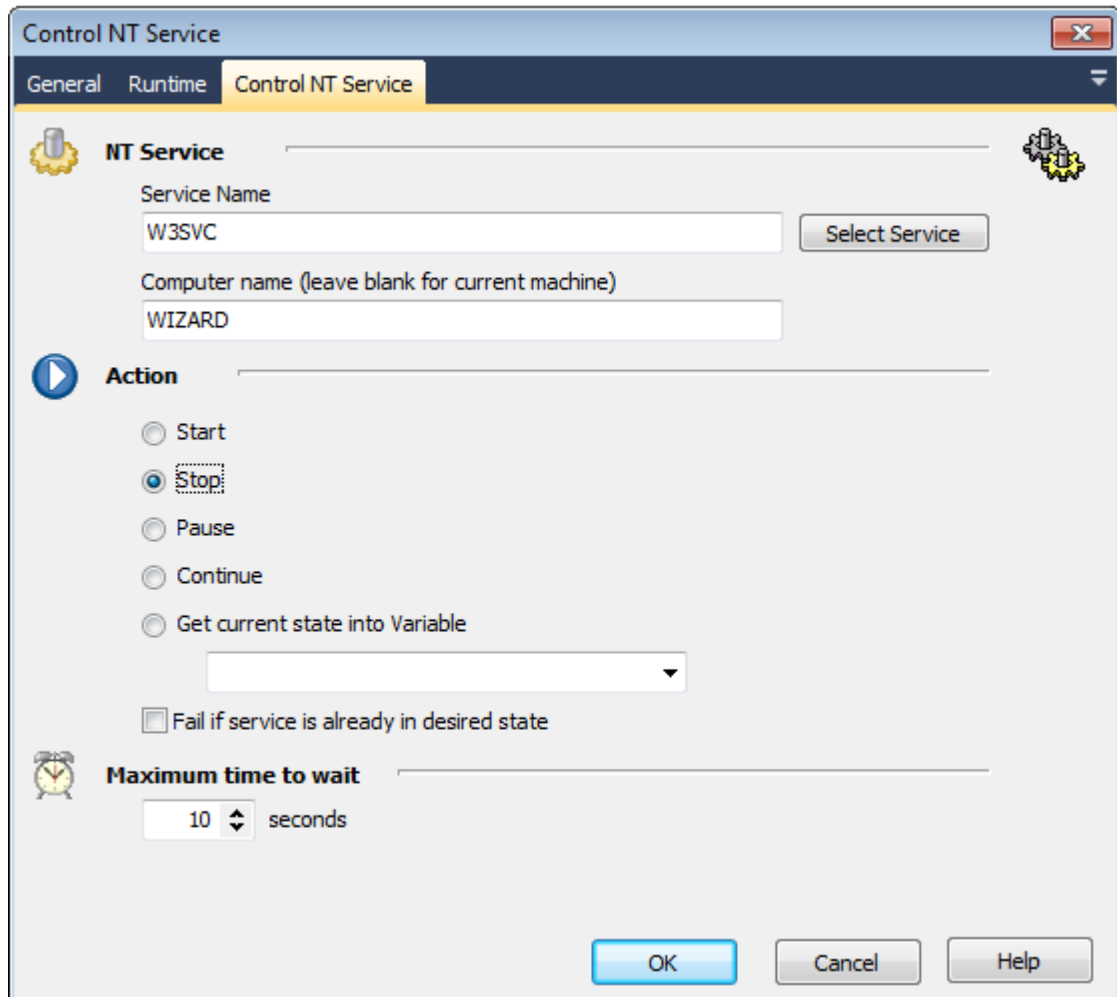
Partition

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.

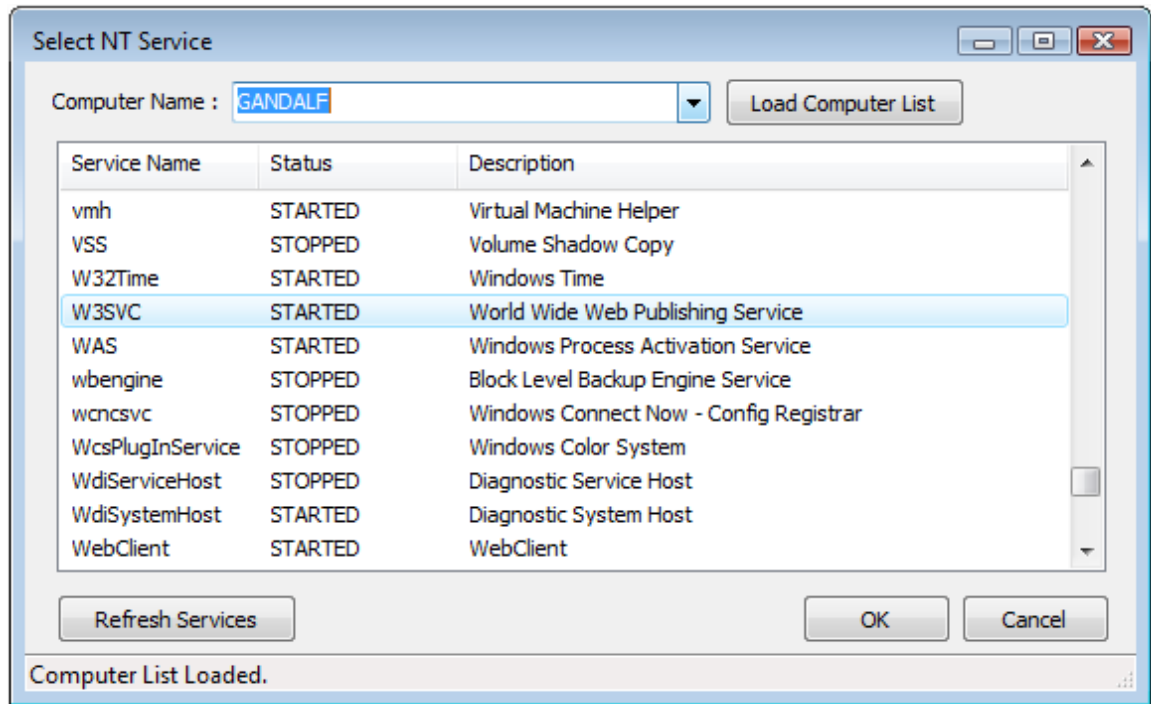
6.36.3 Control Service Action

[FinalBuilder Professional Edition]

This action allows you to control the state of a windows service on the local machine or a remote machine.

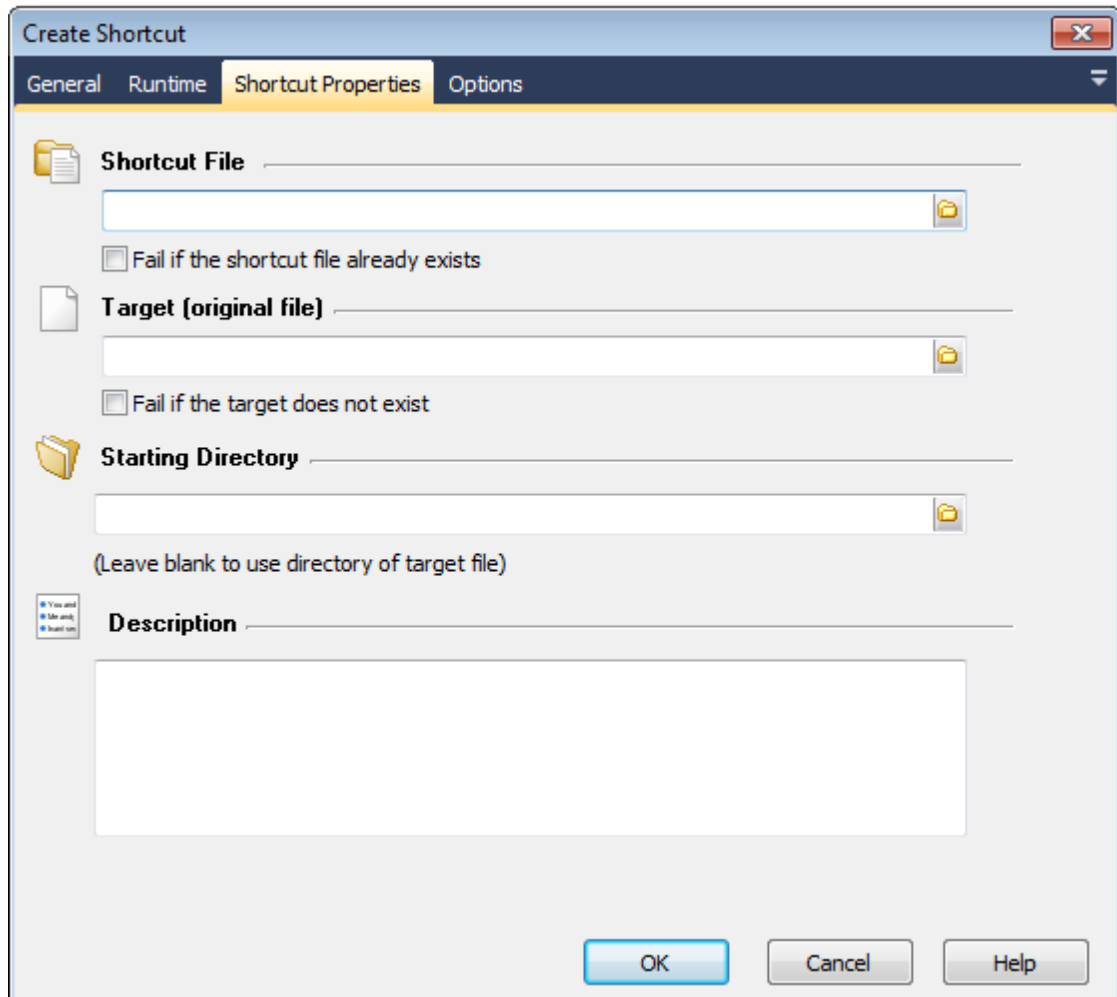


Click on the Select service to choose which service to control. By default the registered services of the local machine are listed, to change this either type the name of the computer or select from the drop down list (click on load to get a list, this is not done automatically as it can be slow in a large network) and the click on the Refresh Services button.



6.36.4 Create Shortcut

The Create Shortcut action allows you to create an Explorer shortcut (.lnk file) from one file to another.



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

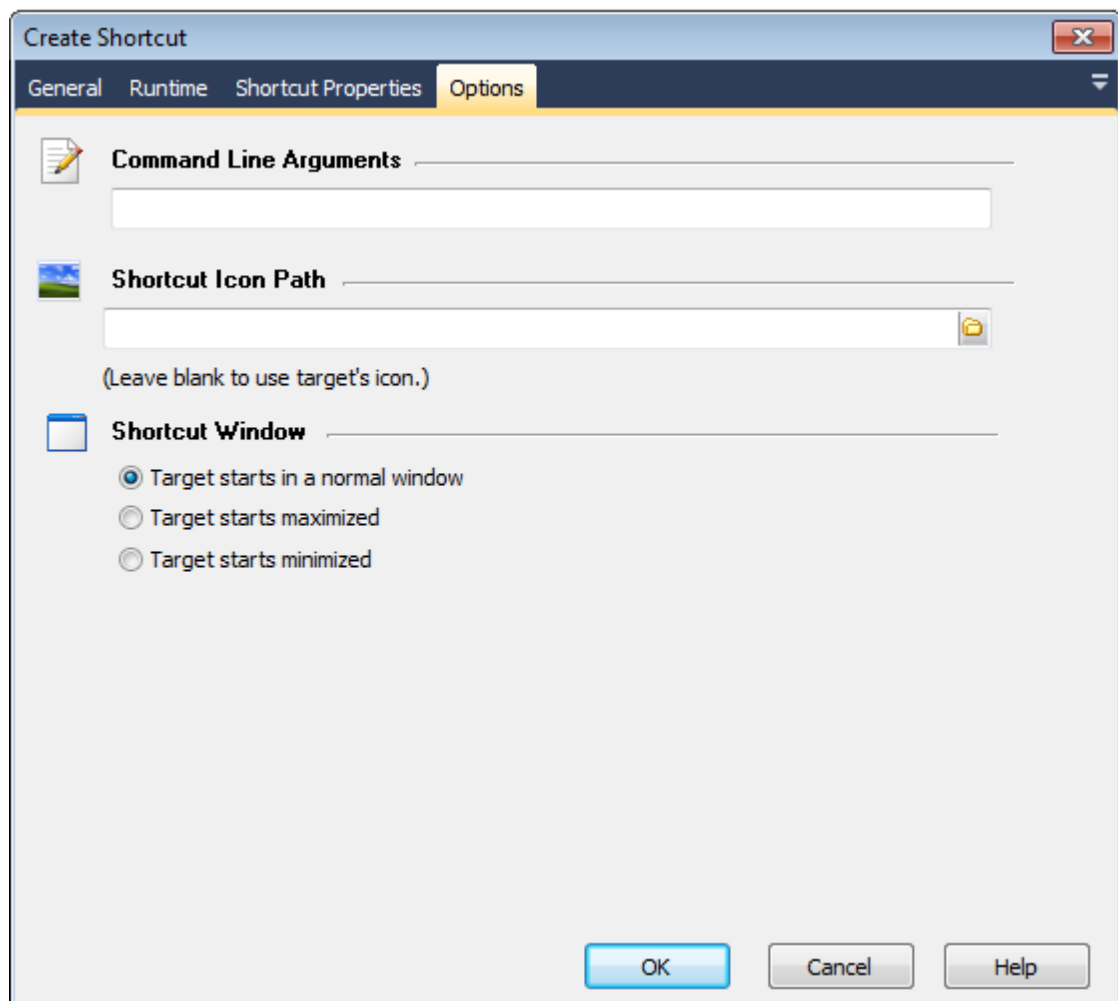
existing file. If this option is not set then the shortcut will be created, regardless.

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

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.

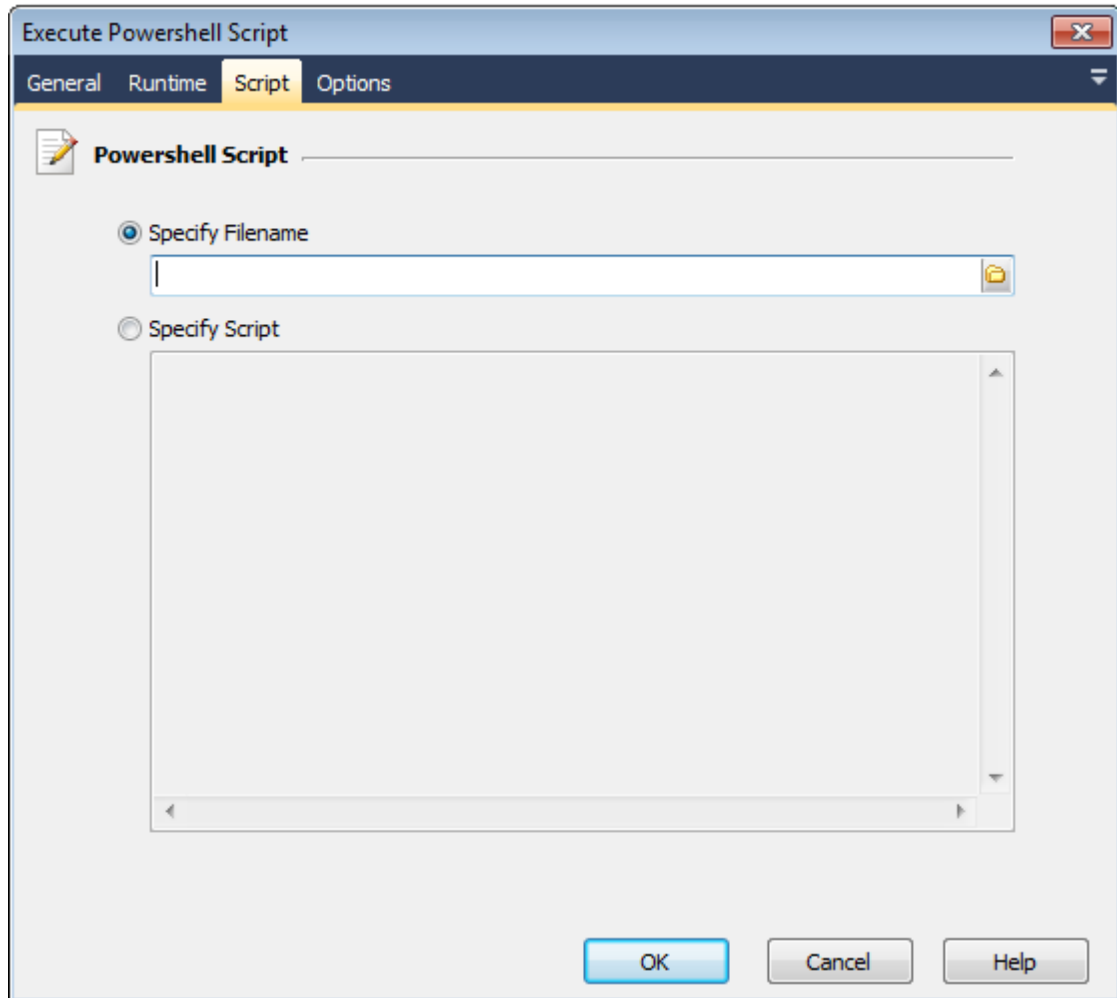
Shortcut Window

This option determines whether the target starts normally, in a maximized window, or in a minimized window in the background.

6.36.5 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.

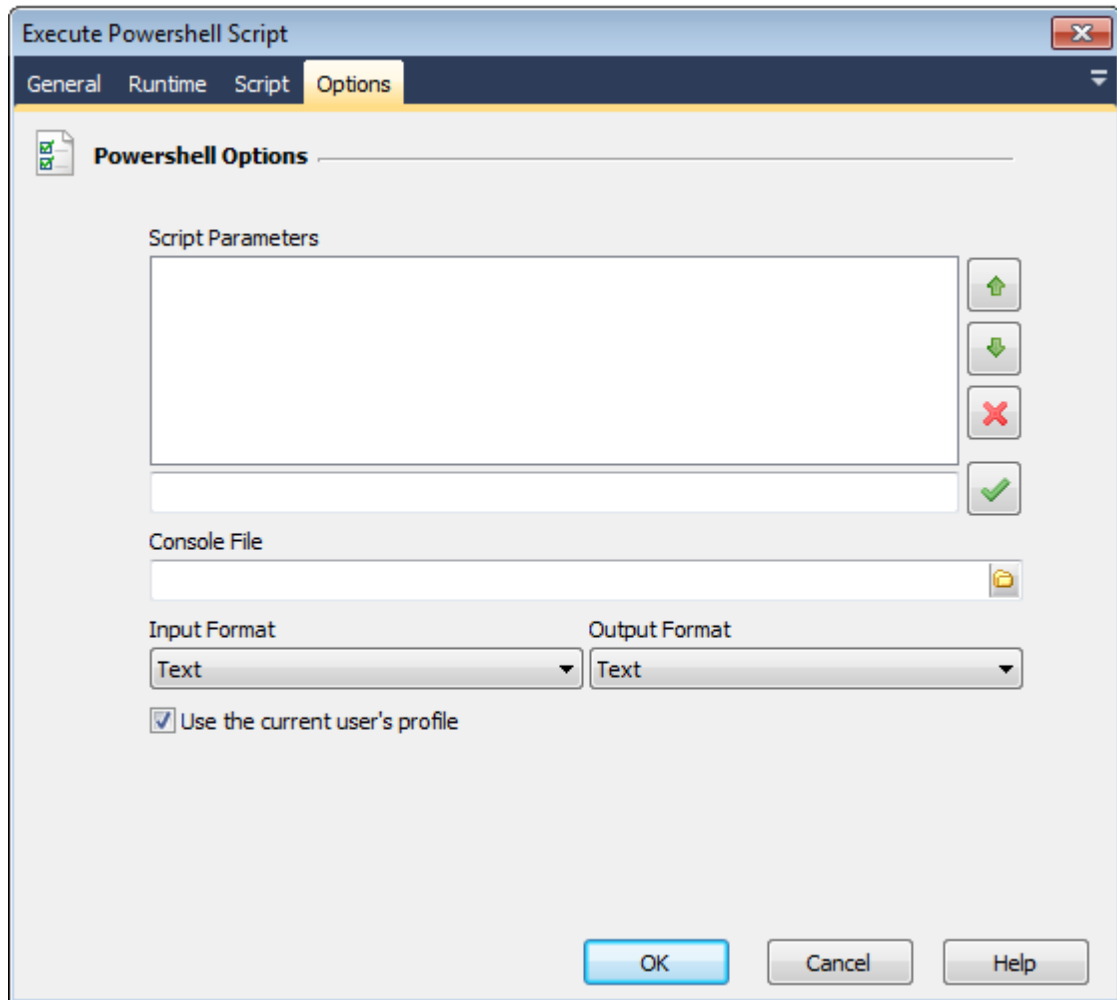
For more information see: <http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx>

**Filename**

The full path to the script to be executed.

Script

The Powershell script block to be executed.

**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 data being received from Windows Powershell.

Use the current user's profile

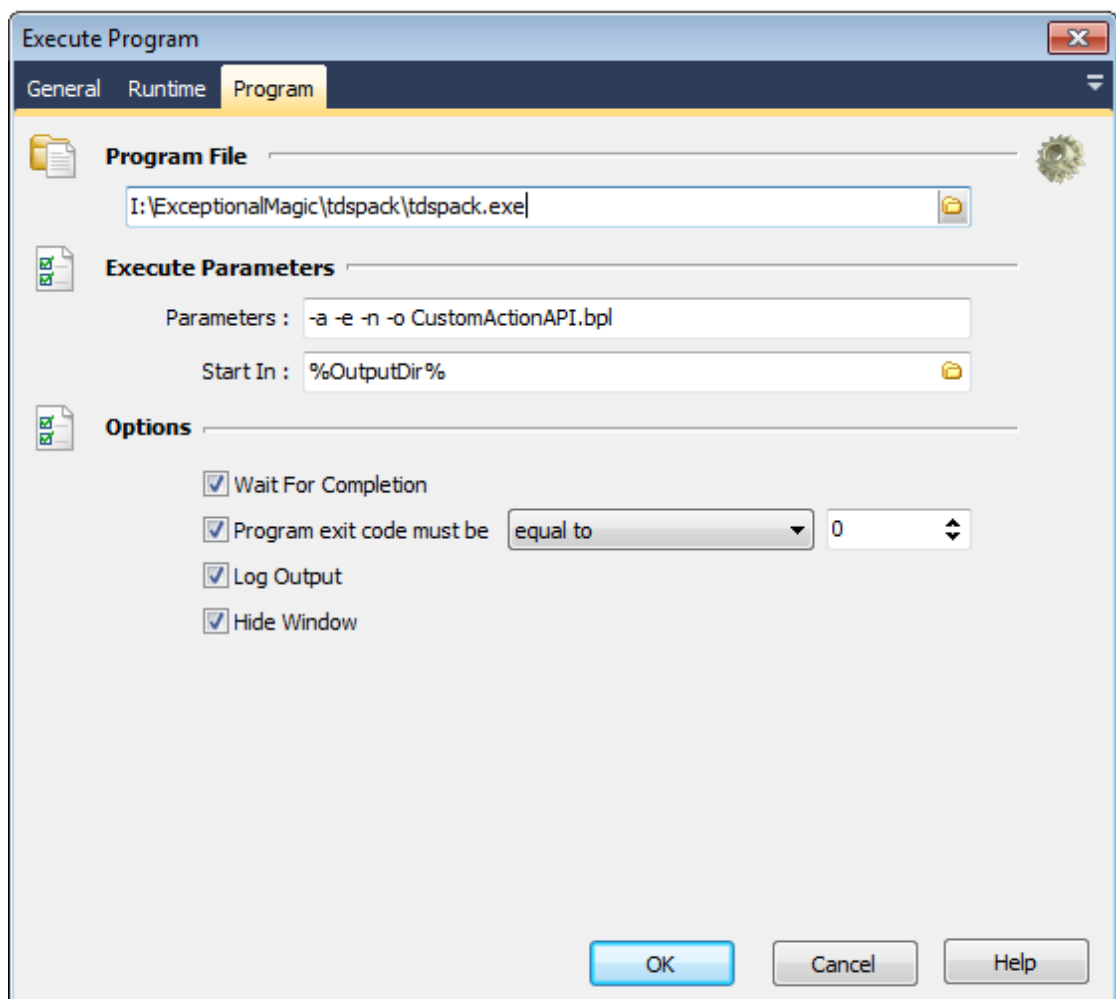
Determines whether the user's profile will be loaded into the Powershell instance.

6.36.6 Execute Program Action

The Execute Program Action allows you to execute nearly any program from a FinalBuilder build.

FinalBuilder can capture the output of console applications (such as command line compilers) and display the captured output in the FinalBuilder 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.



Wait For Completion

By default, FinalBuilder 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.

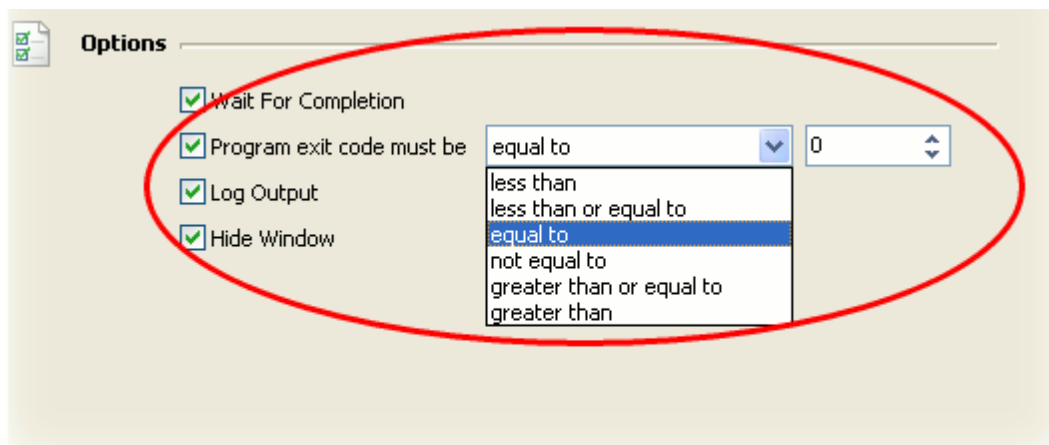
Log Output

If this option is set, FinalBuilder will send all console output to the FinalBuilder 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 FinalBuilder. Unfortunately, this is a problem within the third party programs themselves and it is not possible for FinalBuilder 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..."

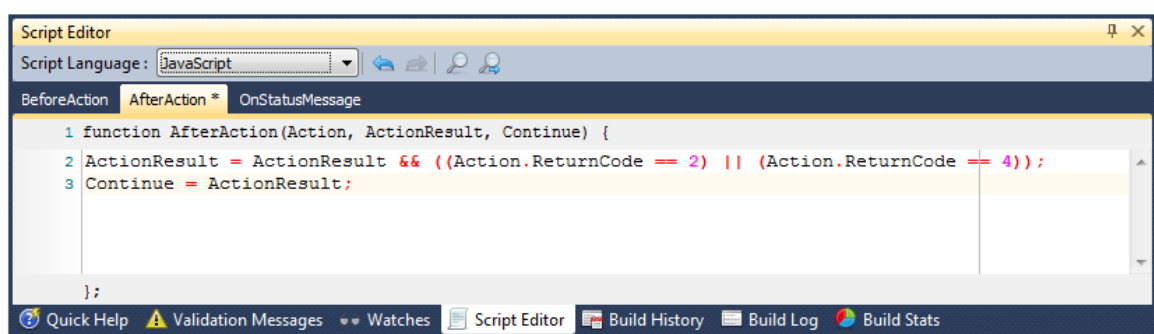
If you enable this option, FinalBuilder will require the program's exit code to be within a certain range:



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:



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

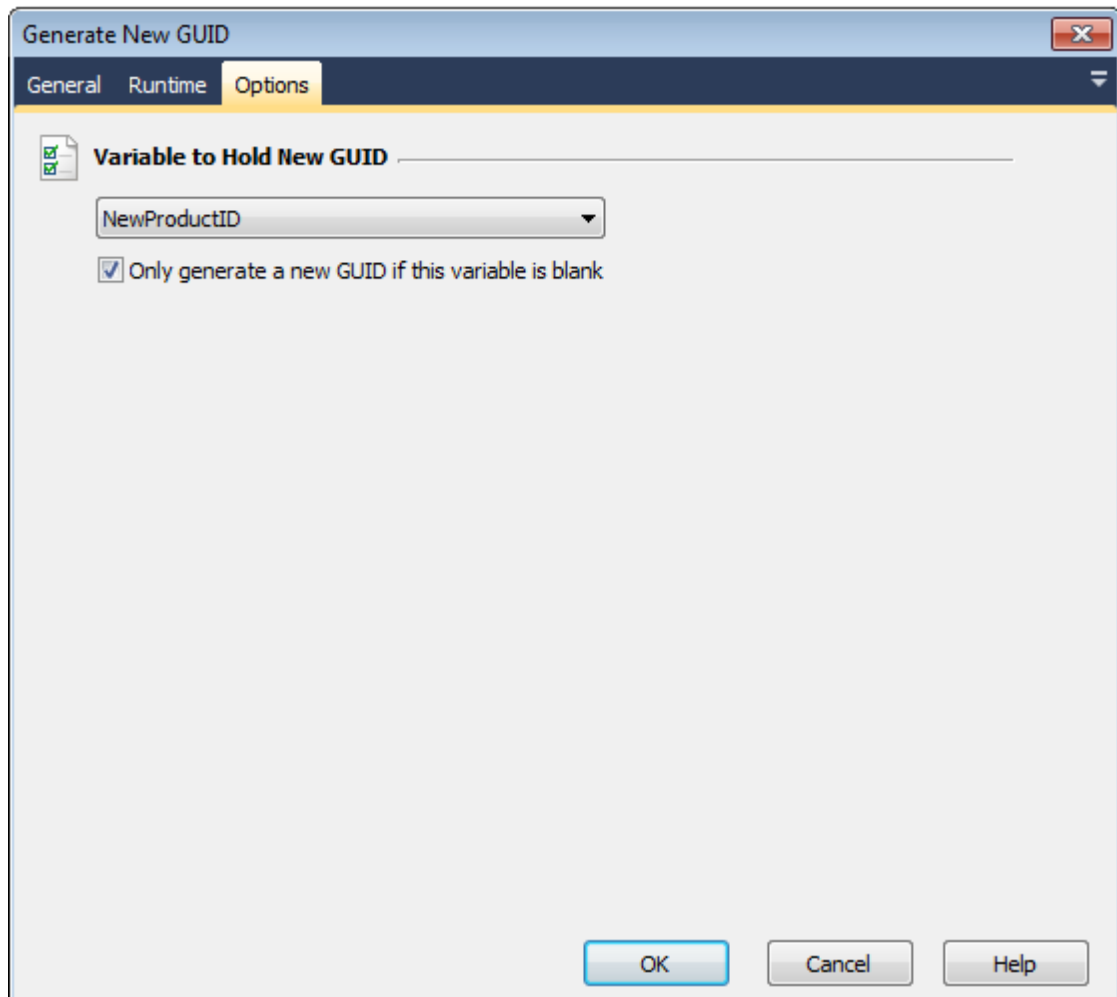
The Action properties available are :

```
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.

6.36.7 Generate New GUID

The Generate New GUID action generates a new GUID string and assigns it to a FinalBuilder variable.



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

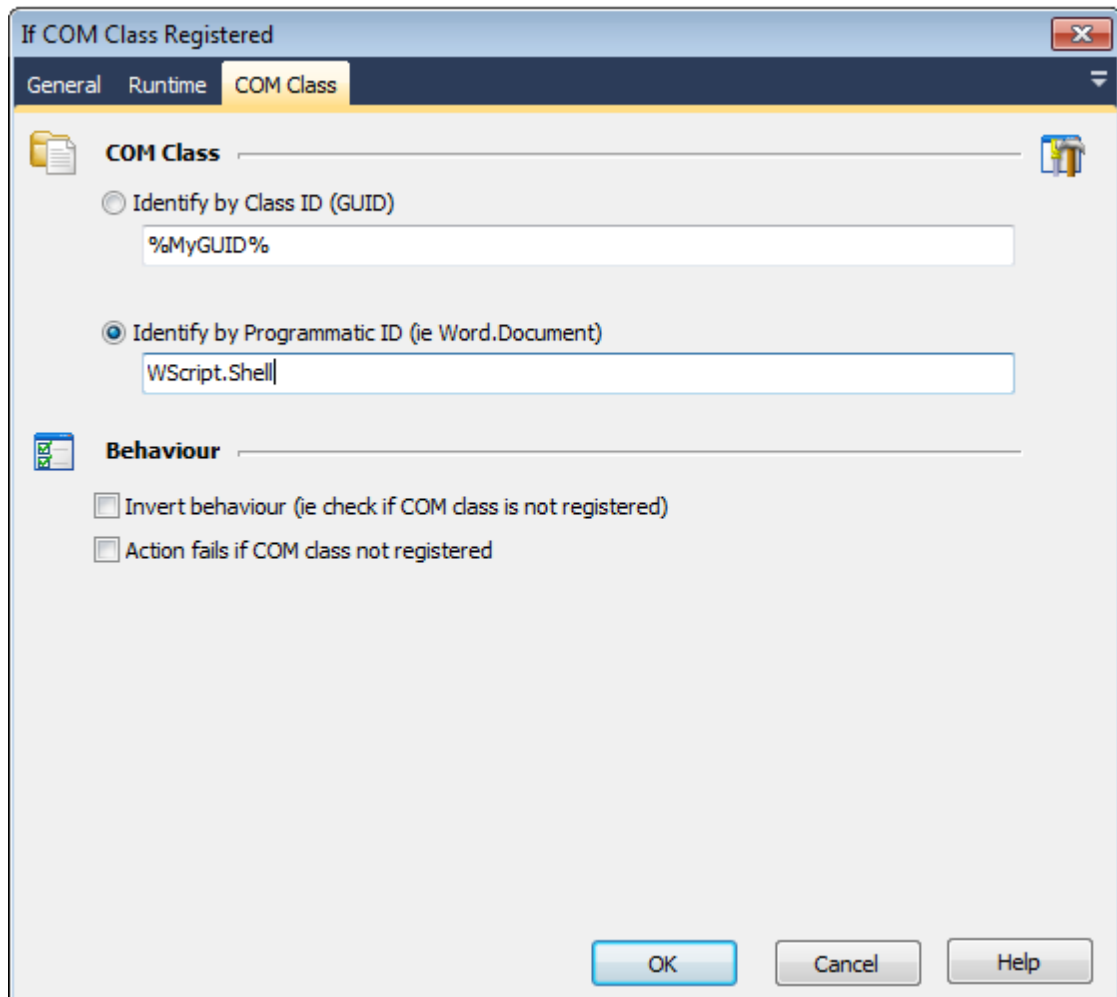
The OnGenerateGUID script event allows you to do some processing when the new GUID is generated. The GUID parameter holds the new GUID value (as a string.)

See also : NewGUIDString() script function.

6.36.8 If COM Class Registered

The If COM Class Registered action enables you to control the flow of your build 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.



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.

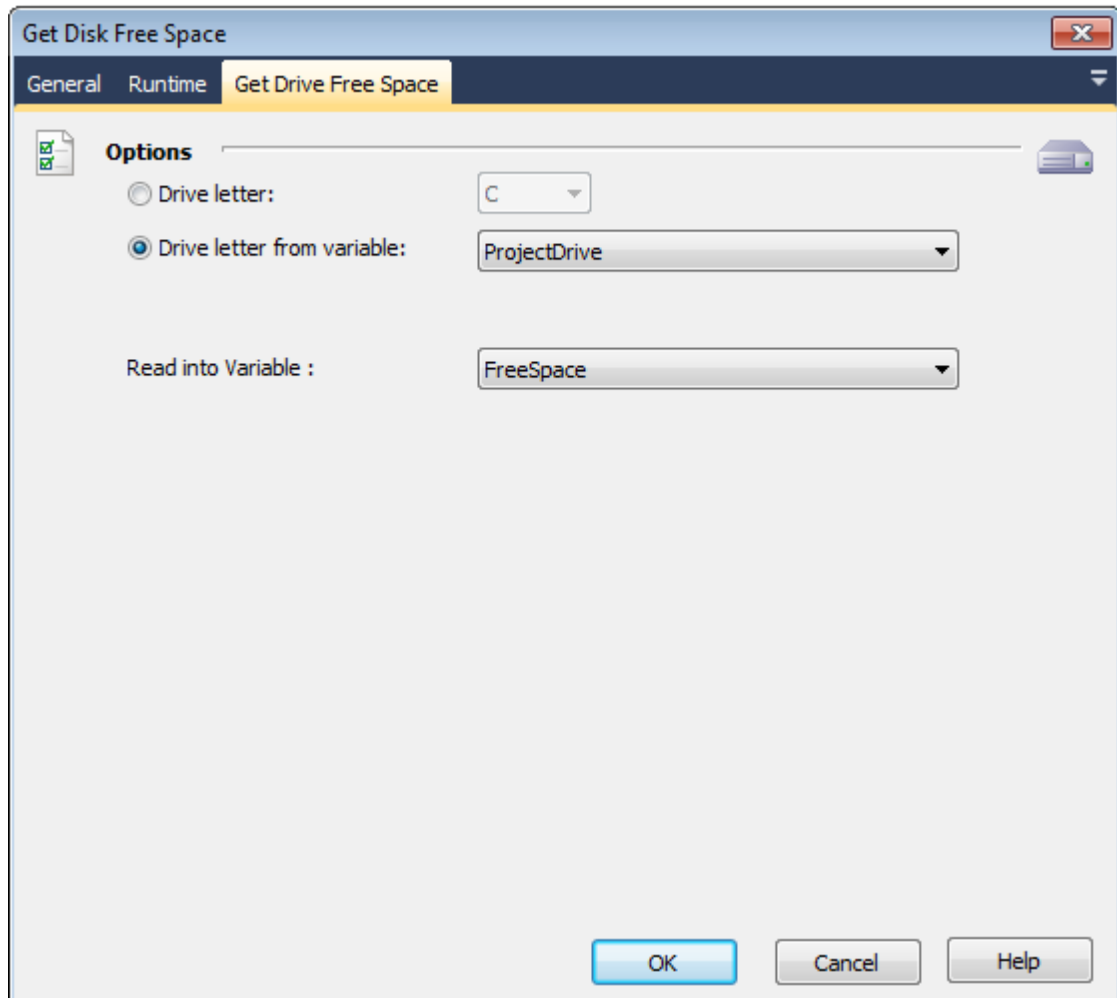
Action fails if COM class not registered

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.'

6.36.9 Logical and Network Drives

6.36.9.1 Get Disk Free Space

Reads the free disk space of a drive into a FinalBuilder variable.

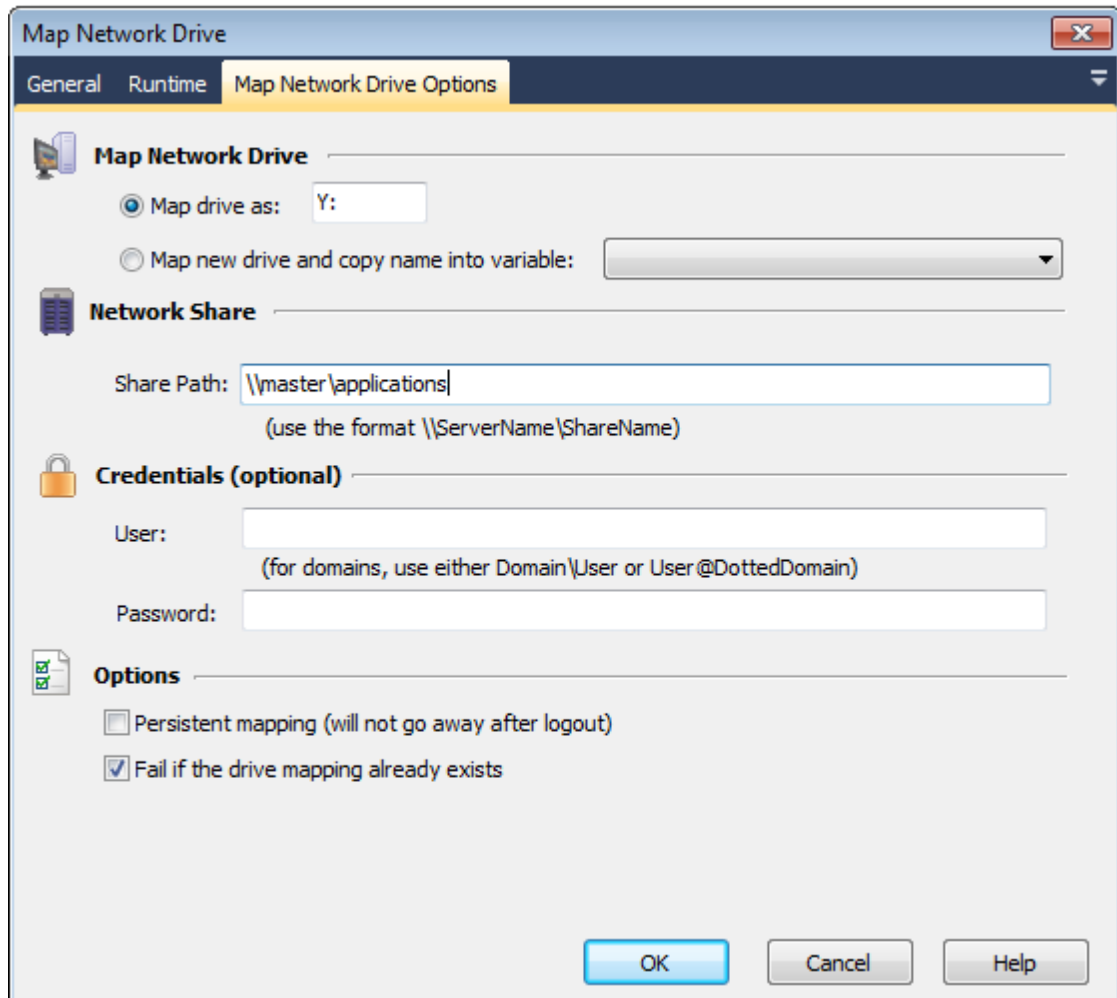


The drive letter can be fixed, or loaded from a variable value at runtime. If loaded from a variable, the variable value must be either a single letter (ie "D"), or a single letter plus a colon (ie "D:").

The disk space available on the drive is returned in bytes. If the drive specified is not available, then the action will fail.

6.36.9.2 Map Network Drive Action

The map network drive action creates a network drive mapping to a remote network share.



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 build.

Save Credentials

Saves the login and password for use with future connection attempts.

Fail if the drive mapping already exists

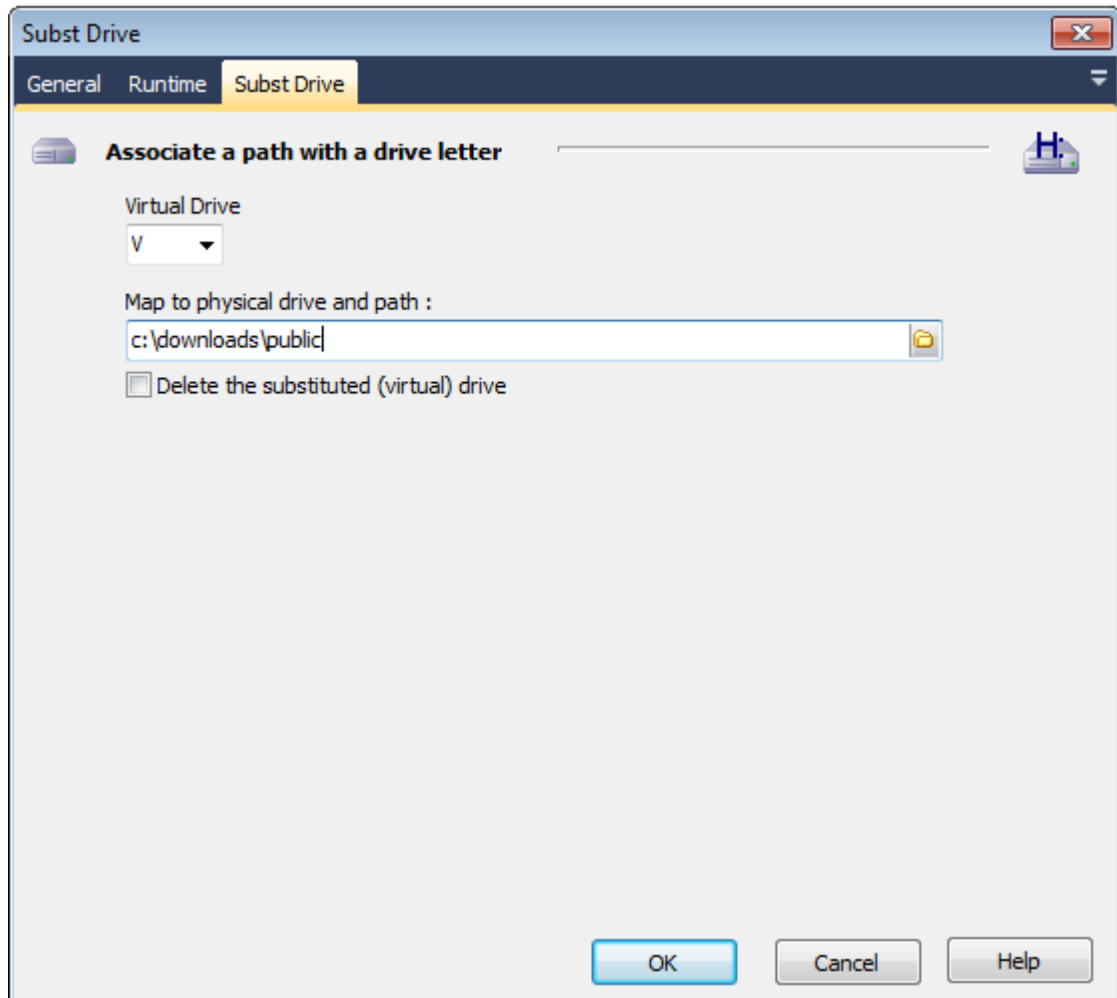
If this option is checked, the action will fail if a drive is already mapped to that drive name. If unchecked, the action will skip the mapping if the drive already exists (note that the action does not check which share is mapped to the drive, nor that the drive is in fact a drive mapping and not a real disk drive!)

This option has no effect if you are mapping a new drive name.

6.36.9.3 Subst Drive Action

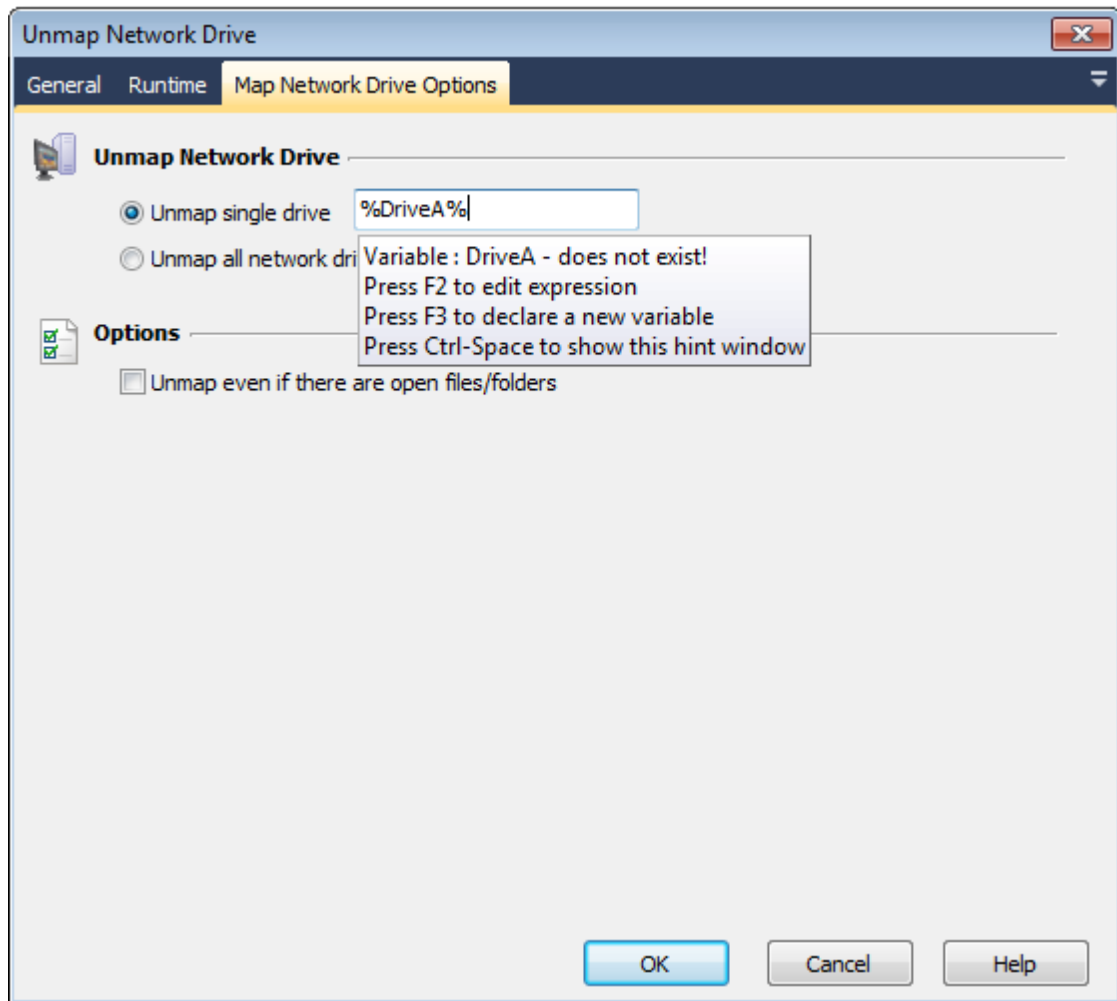
This action calls the DOS subst.exe program, to map a folder to a drive letter.

To map a network share to a virtual drive, use the Map Network Drive action.



6.36.9.4 Unmap Network Drive Action

The unmap network drive action deletes a previously mapped network drive.



Unmap Single Drive

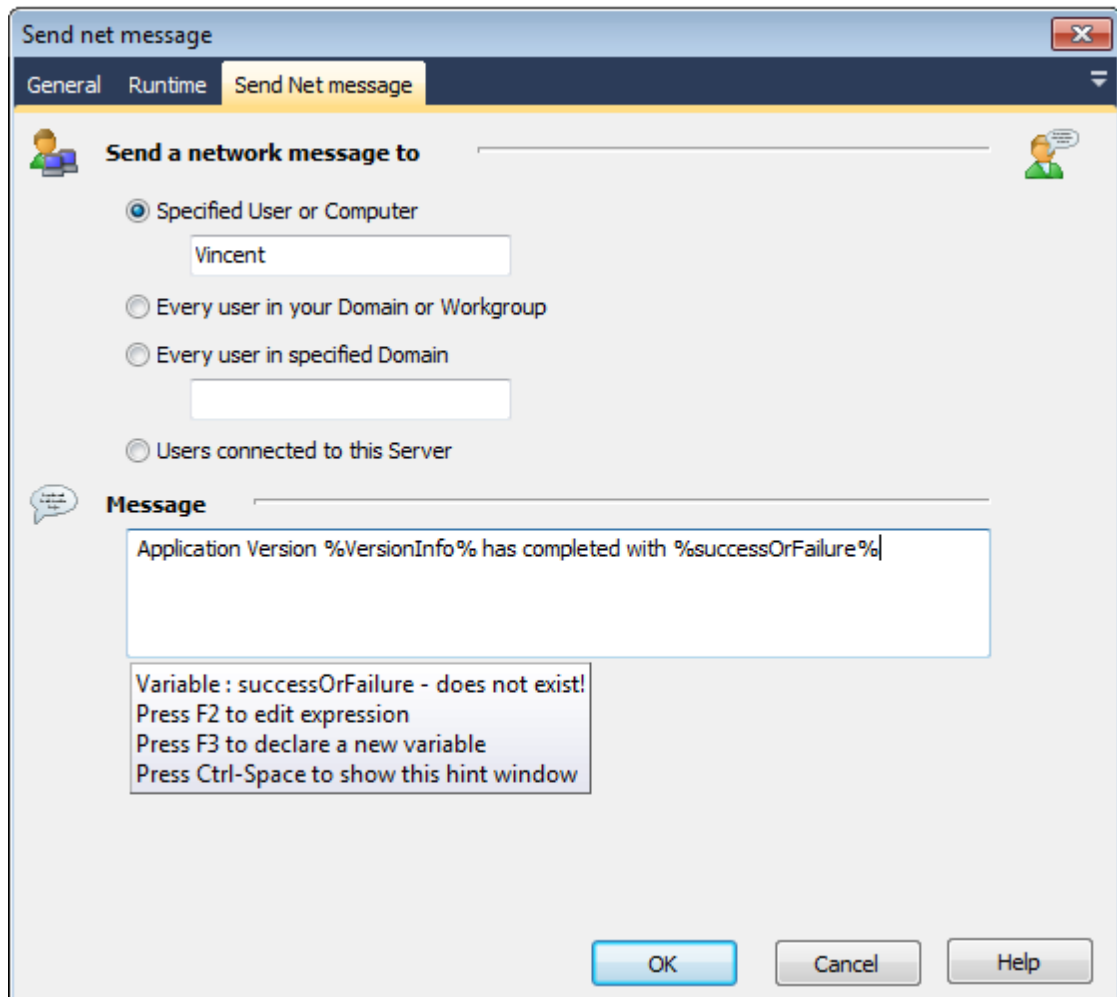
Enter the drive name you wish to unmap, or (optionally, as shown in the picture above) a variable name.

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 FinalBuilder and drives mapped by other processes.

6.36.10 Net Send Message Action

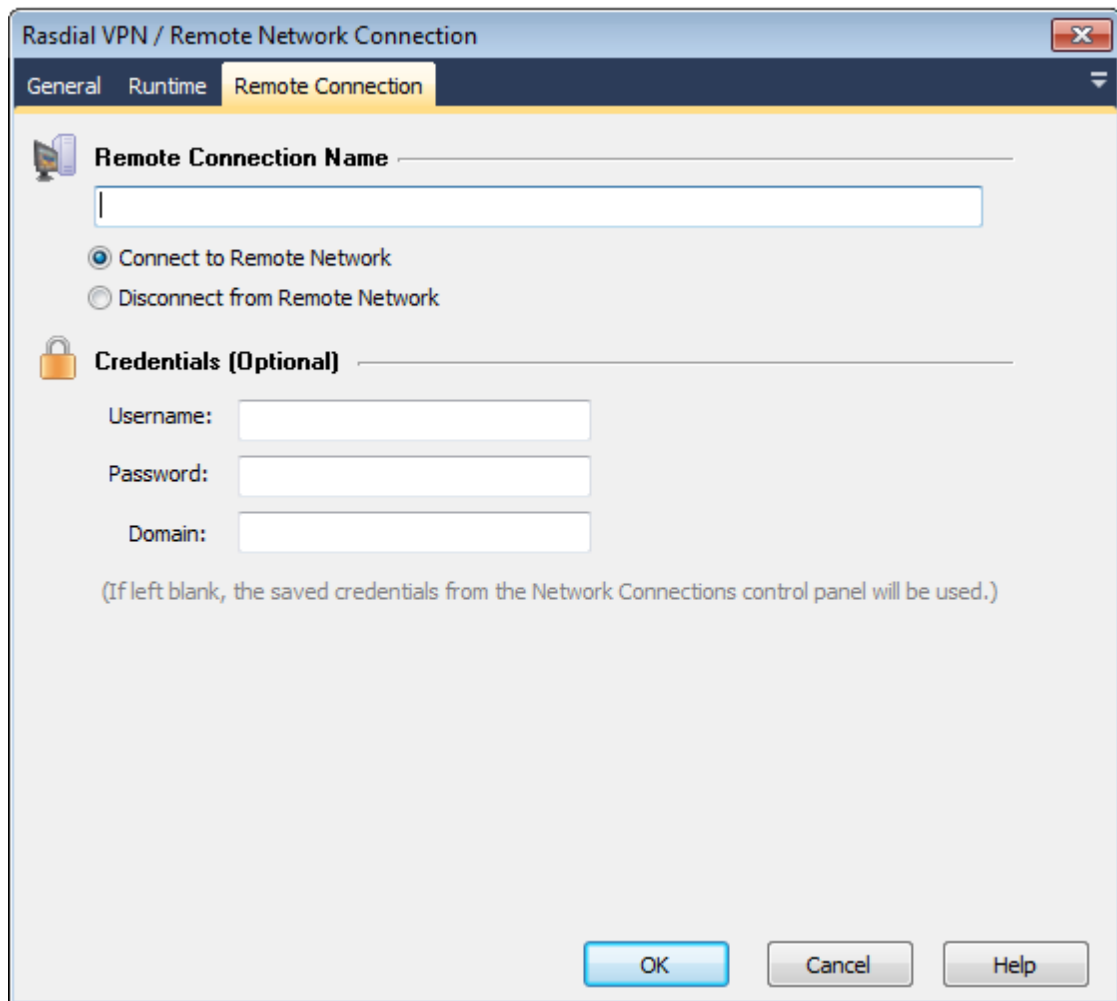
Send a Message using windows NET SEND



6.36.11 Networking Configuration and Diagnostics

6.36.11.1 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

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.

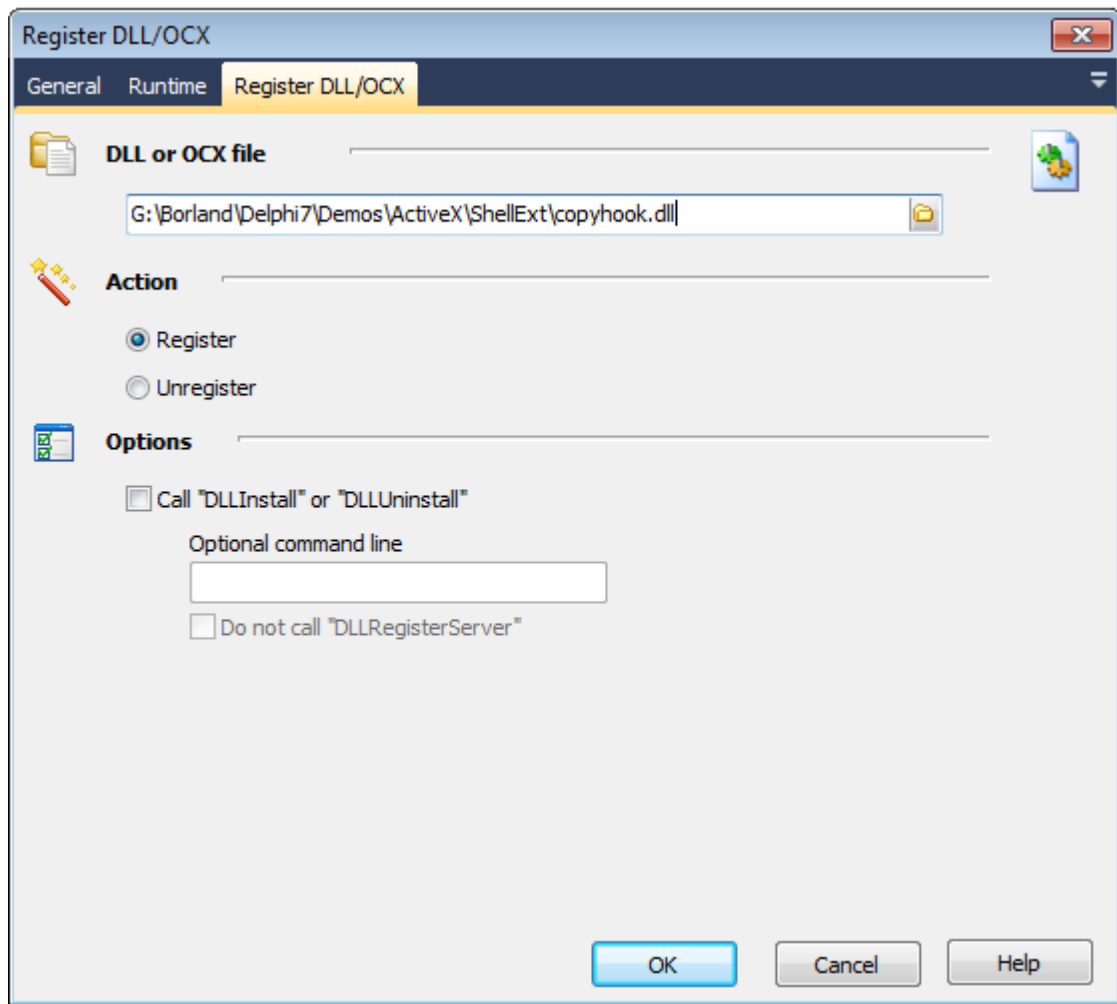
Credentials (Optional)

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.

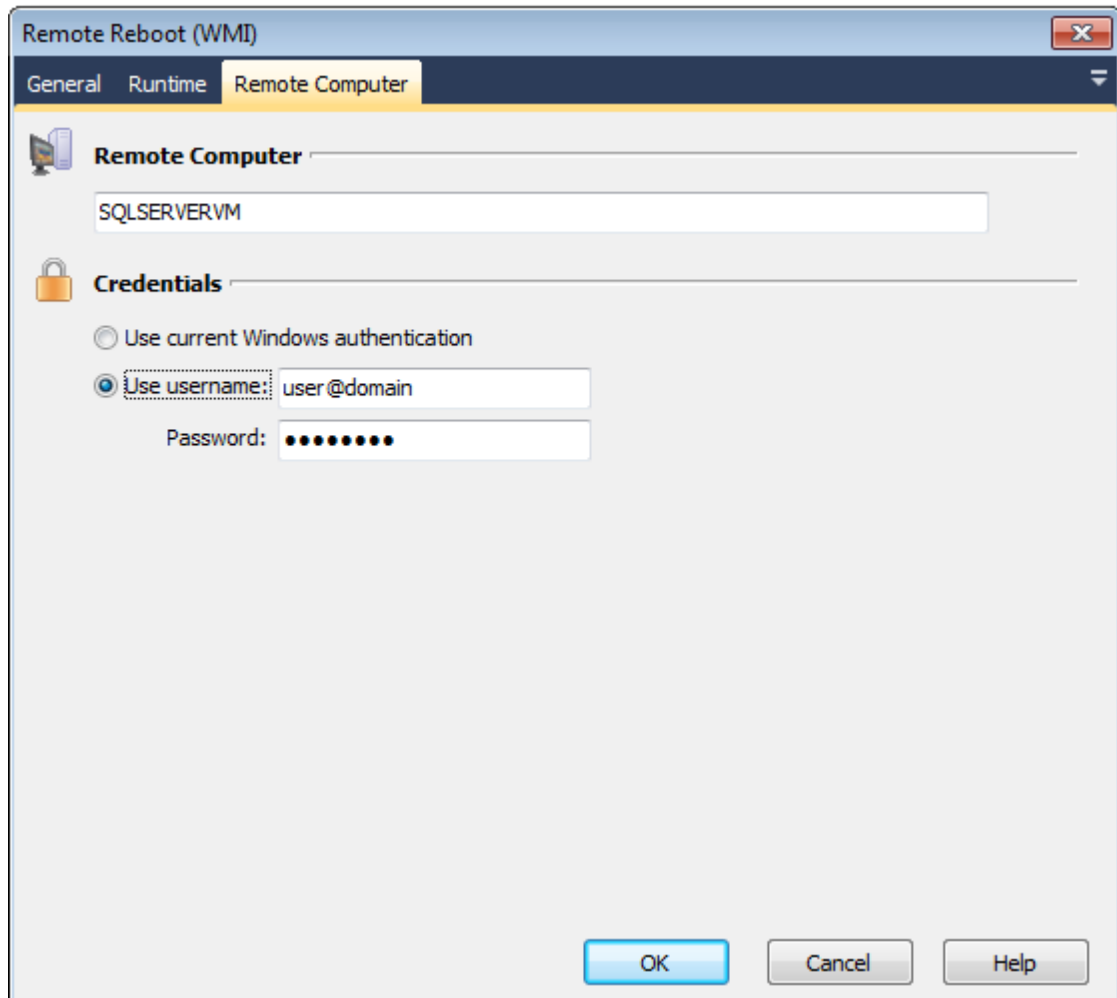
6.36.12 Register DLL/OCX Action

Register an ActiveX control (OCX) or COM DLL.



6.36.13 Remote Reboot Action

The Remote Reboot action allows you to reboot a Windows computer using WMI.



Remote Computer

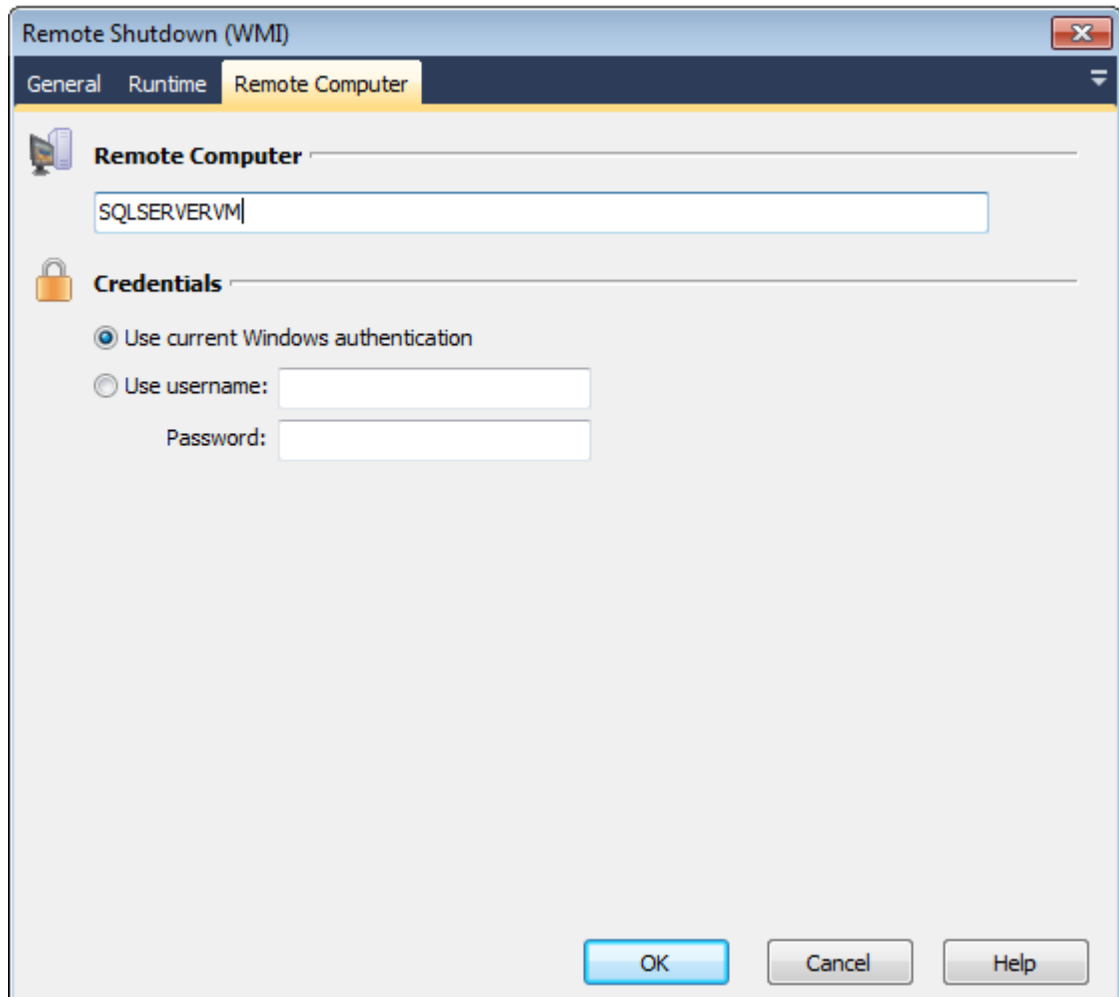
Enter the address of the computer to reboot. Note that the Remote Reboot action cannot be used on the local machine.

Credentials

WMI can either use the current Windows account, or a specific user account, for authentication.

6.36.14 Remote Shutdown Action

The Remote Shutdown action allows you to shut down a Windows computer using WMI.



Remote Computer

Enter the address of the computer to shut down. Note that the Remote Shutdown action cannot be used on the local machine.

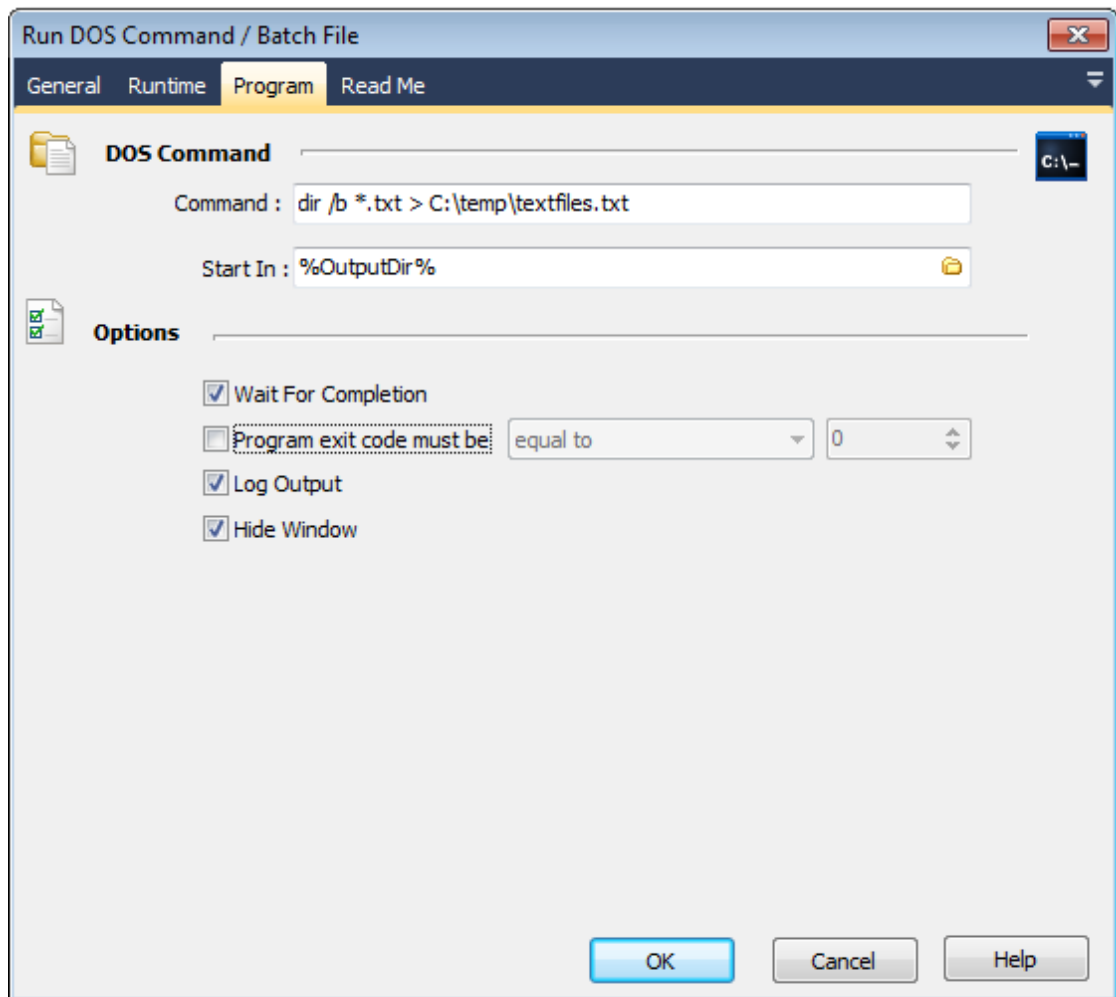
Credentials

WMI can either use the current Windows account, or a specific account, for authentication.

6.36.15 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

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 :

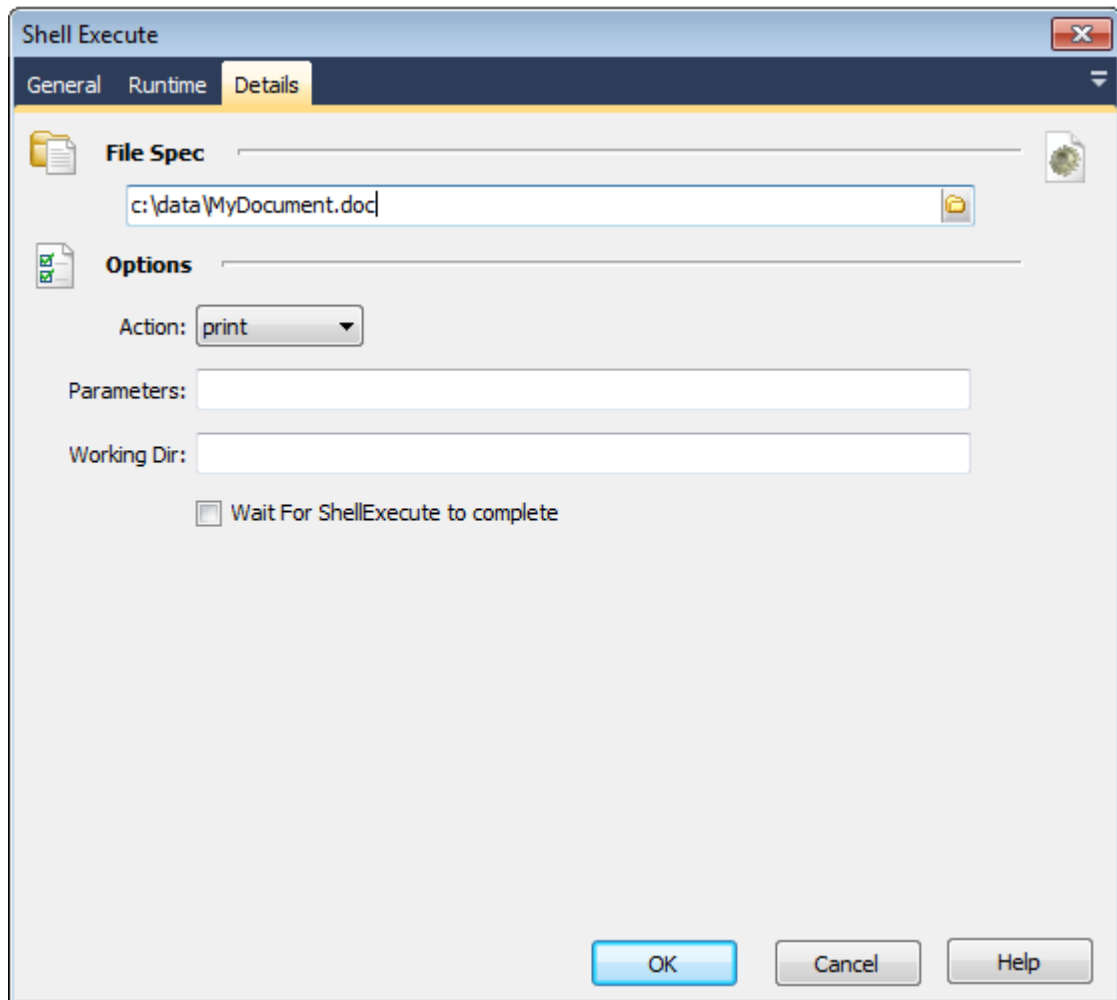
property StartInDir : WideString;
property LogOutput : WordBool;
property WaitForCompletion : WordBool;
property ReturnCode : integer; //read only
property HideWindow : WordBool;
property Command : WideString;
property EnableReturnCodeCheck : Boolean;
property ReturnCodeComparator : TFBRunReturnCodeComparator;
property ReturnCodeToCheck : Boolean;

These properties may be set in the BeforeAction and AfterAction script events.

6.36.16 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:



6.36.17 Sysinternals PsTools Suite

PSTools is a free suite of Windows administration utilities provided by SysInternals.Com .

FinalBuilder provides full command-line automation for the PsTools suite

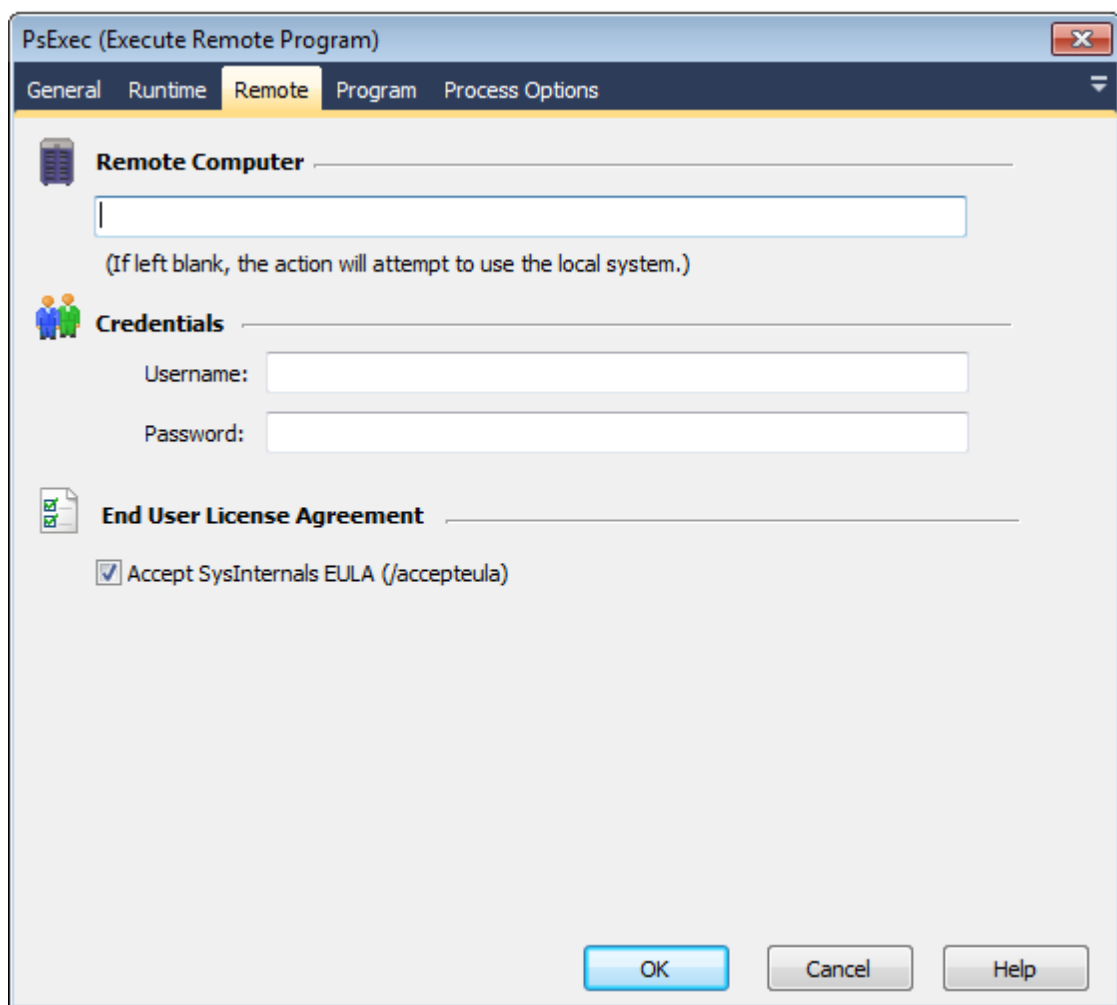
Global Options

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

Enter the name of a remote computer on the network. If no name is supplied, the action will operate on the local machine.

Credentials

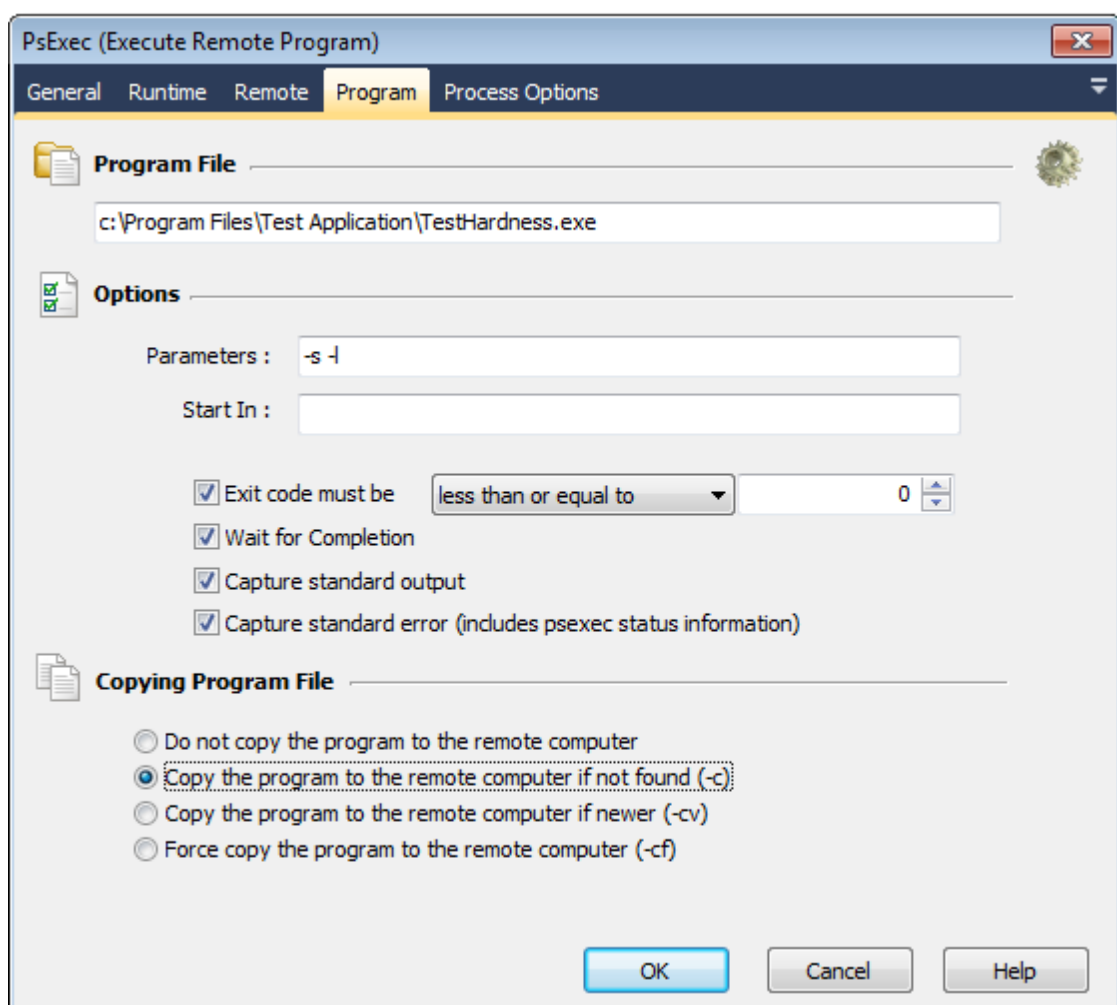
Enter a username and password combination. If no credentials are supplied, the action will attempt to connect using the credentials of the currently logged in user.

6.36.17.1 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.



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.

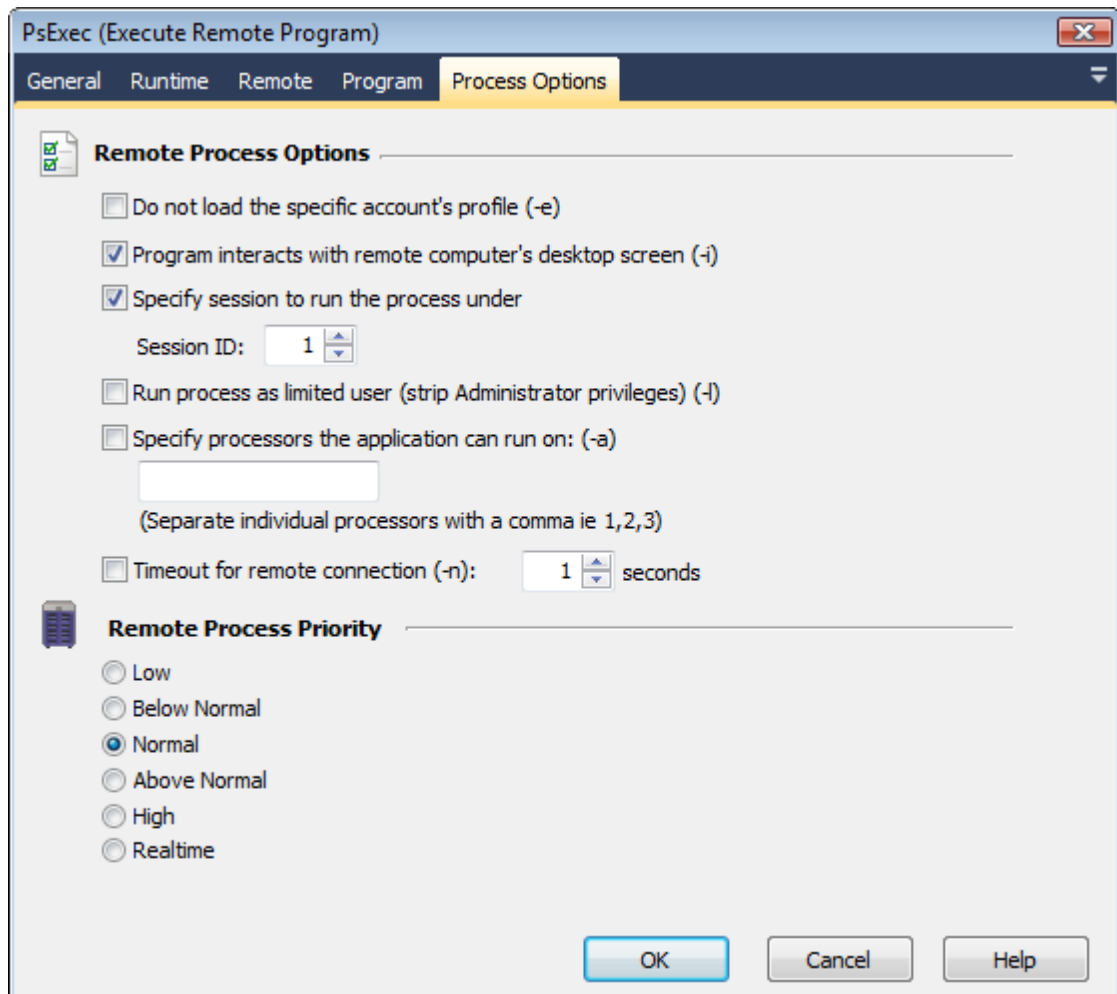
Capture standard output / Standard error

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".

**"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"

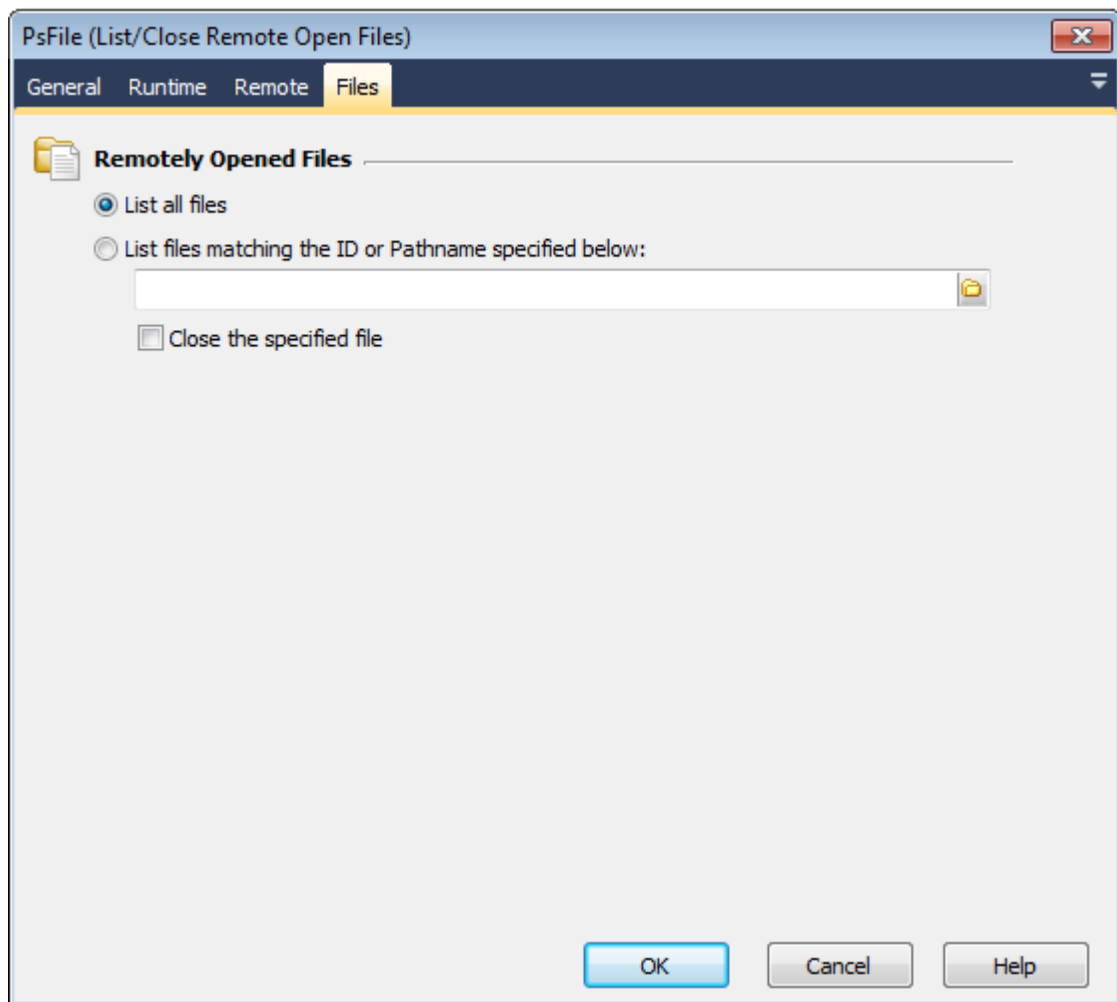
Specify a timeout length for network communications.

"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.

6.36.17.2 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.



"List all files"

Select this option to have PsFile list all open files on the specified computer.

"List files matching..."

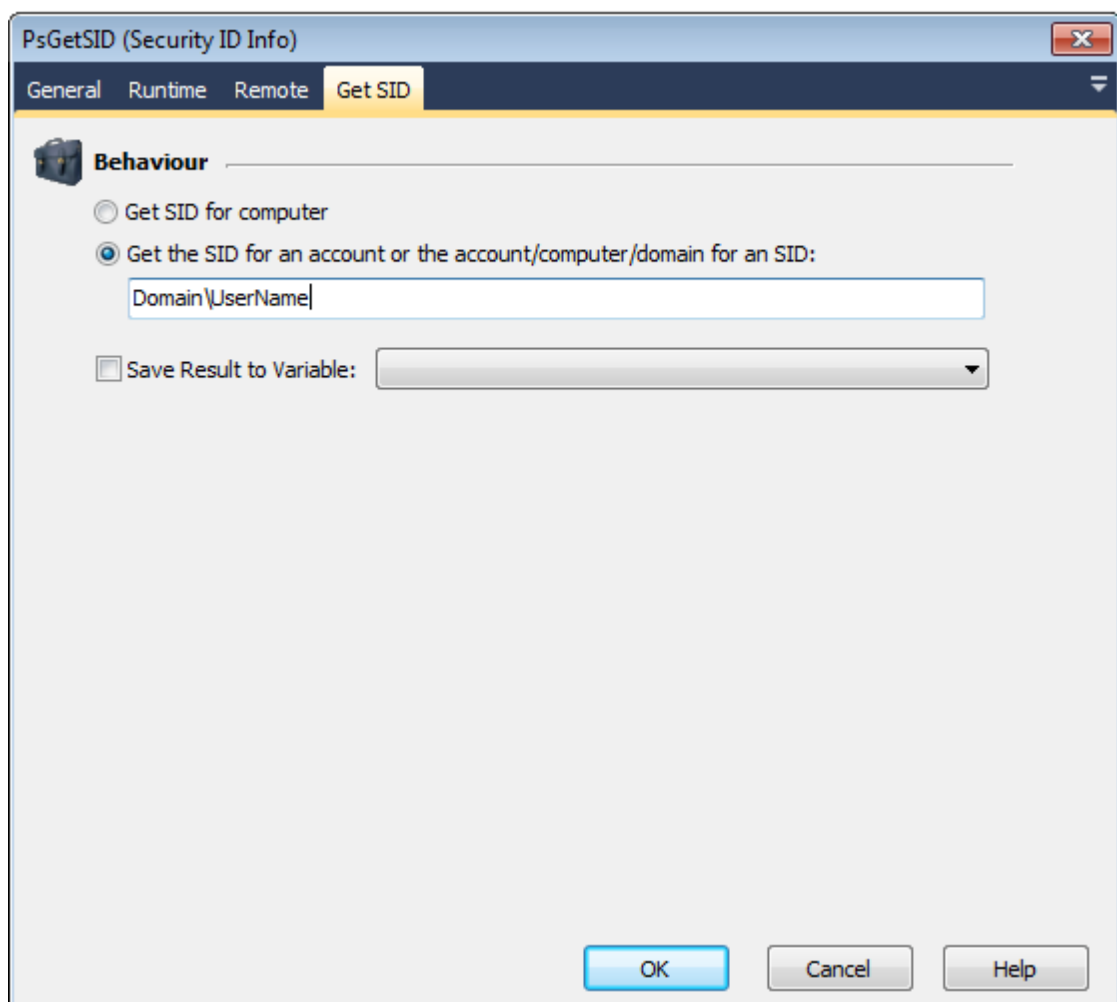
Enter a PsFile ID, or a (partial or complete) pathname, in order to just list certain files. Check the **"Close the specified file"** box to force-close any files which match.

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.3 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.



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

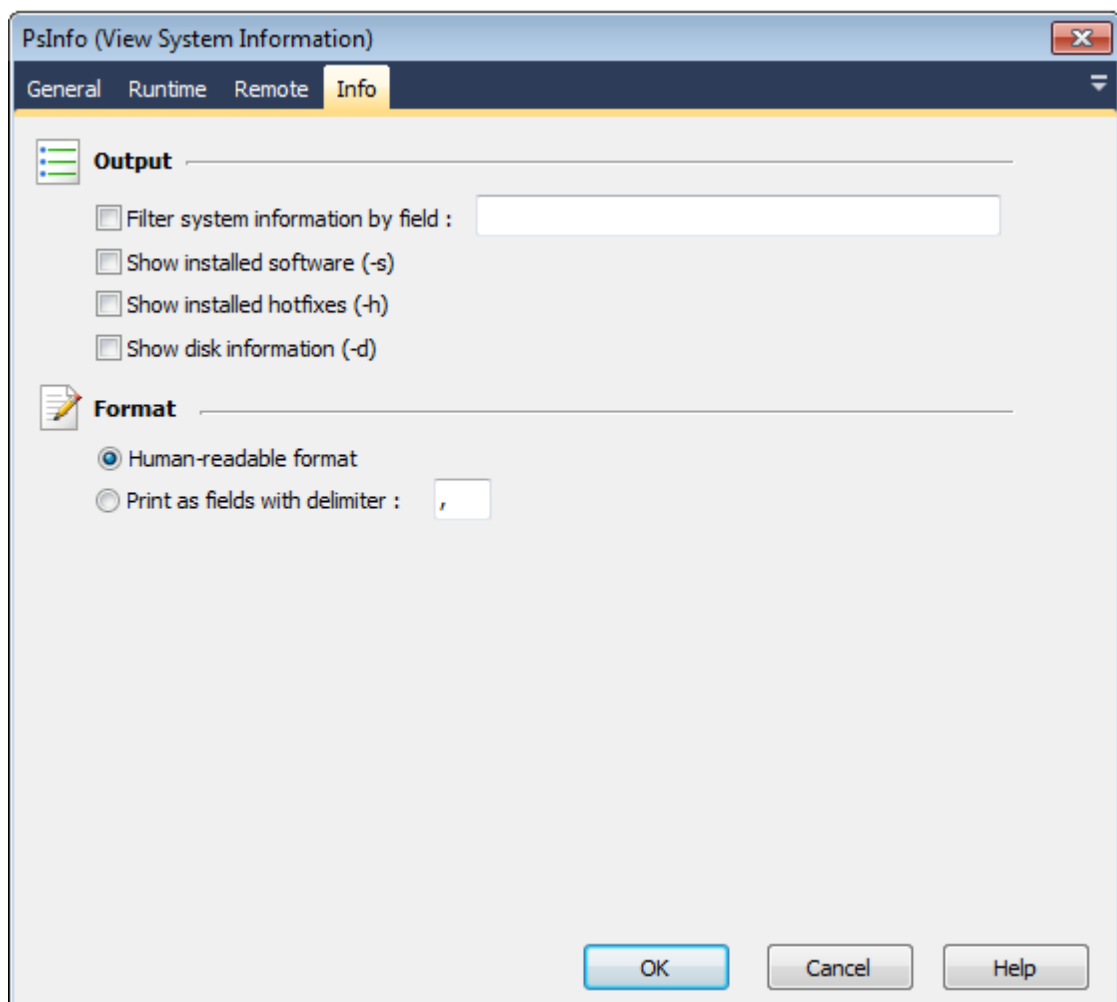
Check this option in order to save the SID (or account/domain/computer name) to a variable.

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.4 PsInfo (View System Information)

PsInfo logs information about the local (or remote) system. See the PsInfo help topic of the PsTools help for details.



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

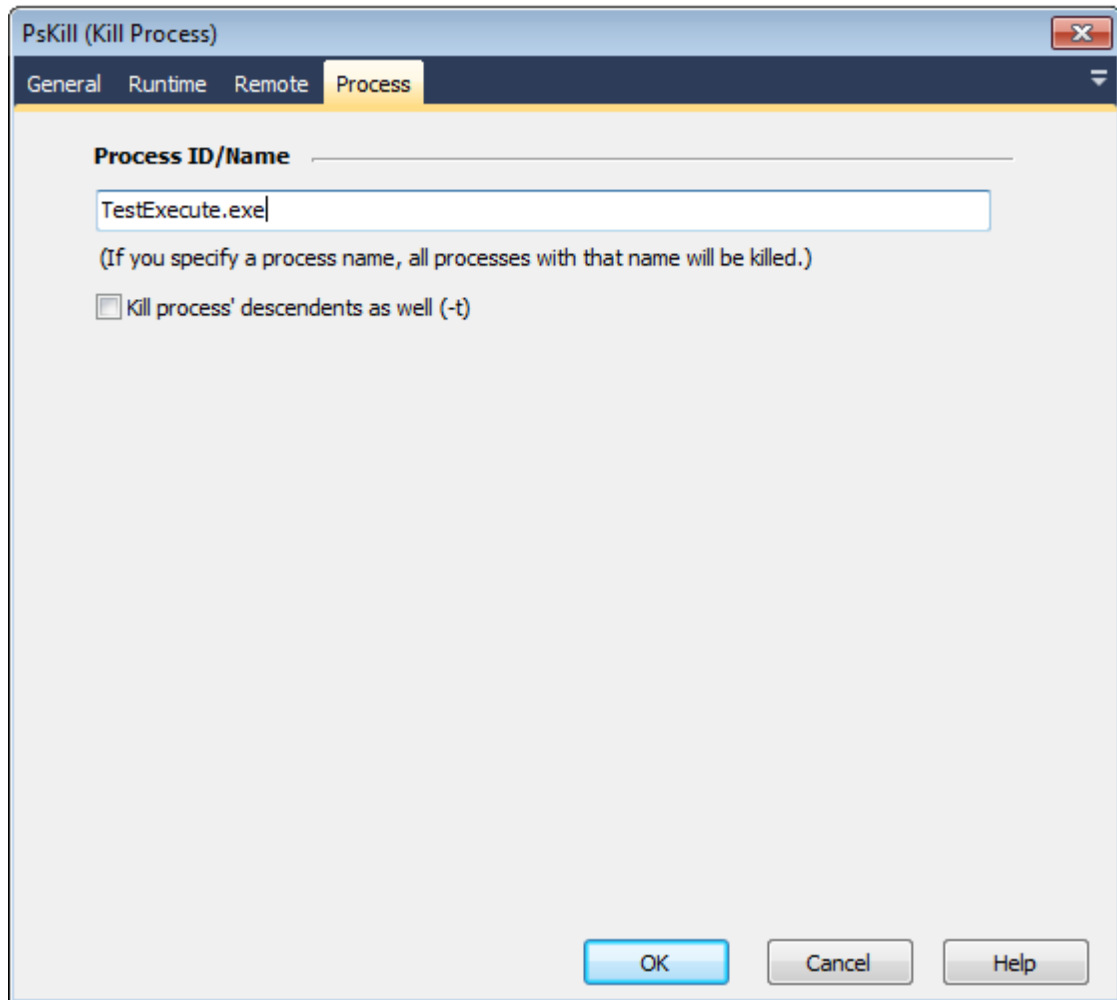
PsInfo output can be printed in human-readable format, or a delimited machine-readable fields (the default is comma-delimited.)

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.5 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**

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' descendants as well (-t)

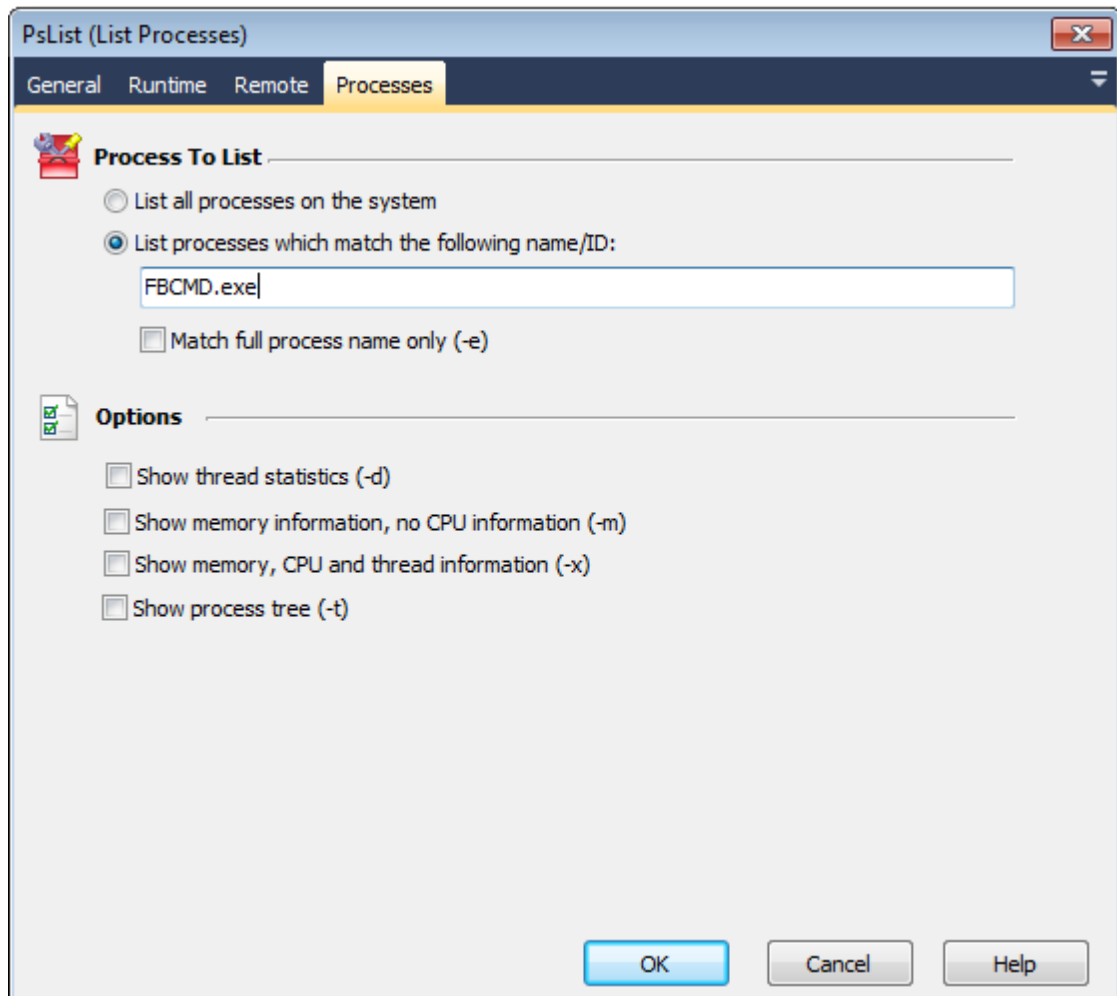
If this option is checked, PsKill will also kill the process' descendants in the process tree.

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.6 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.



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)

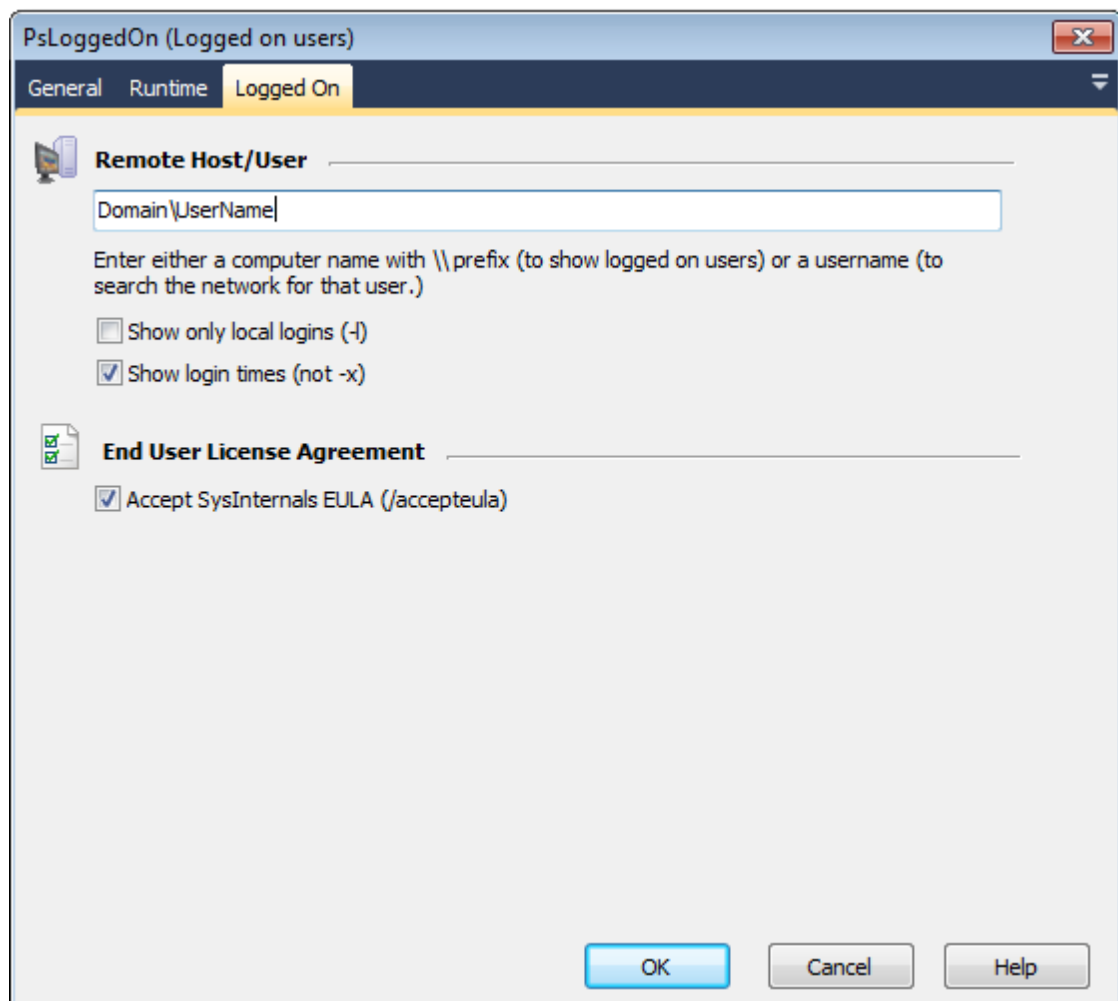
If you choose this option, the processes will be listed in a tree structure showing parent and child processes.

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.7 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

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)"

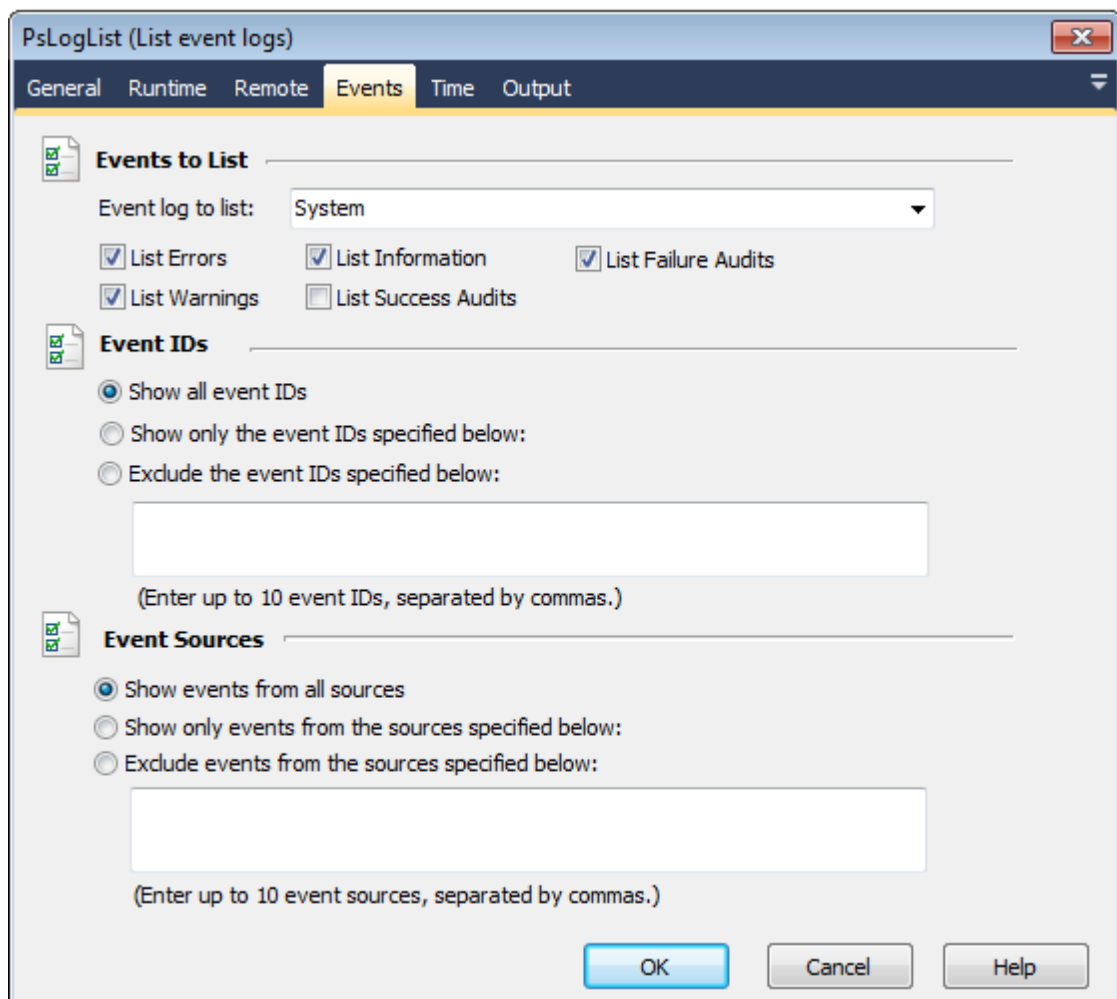
If this option is selected, PsLoggedOn only lists users that are logged in locally.

"Show login times (not -x)"

If this option is selected, PsLoggedOn shows login times as well as usernames.

6.36.17.8 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.



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

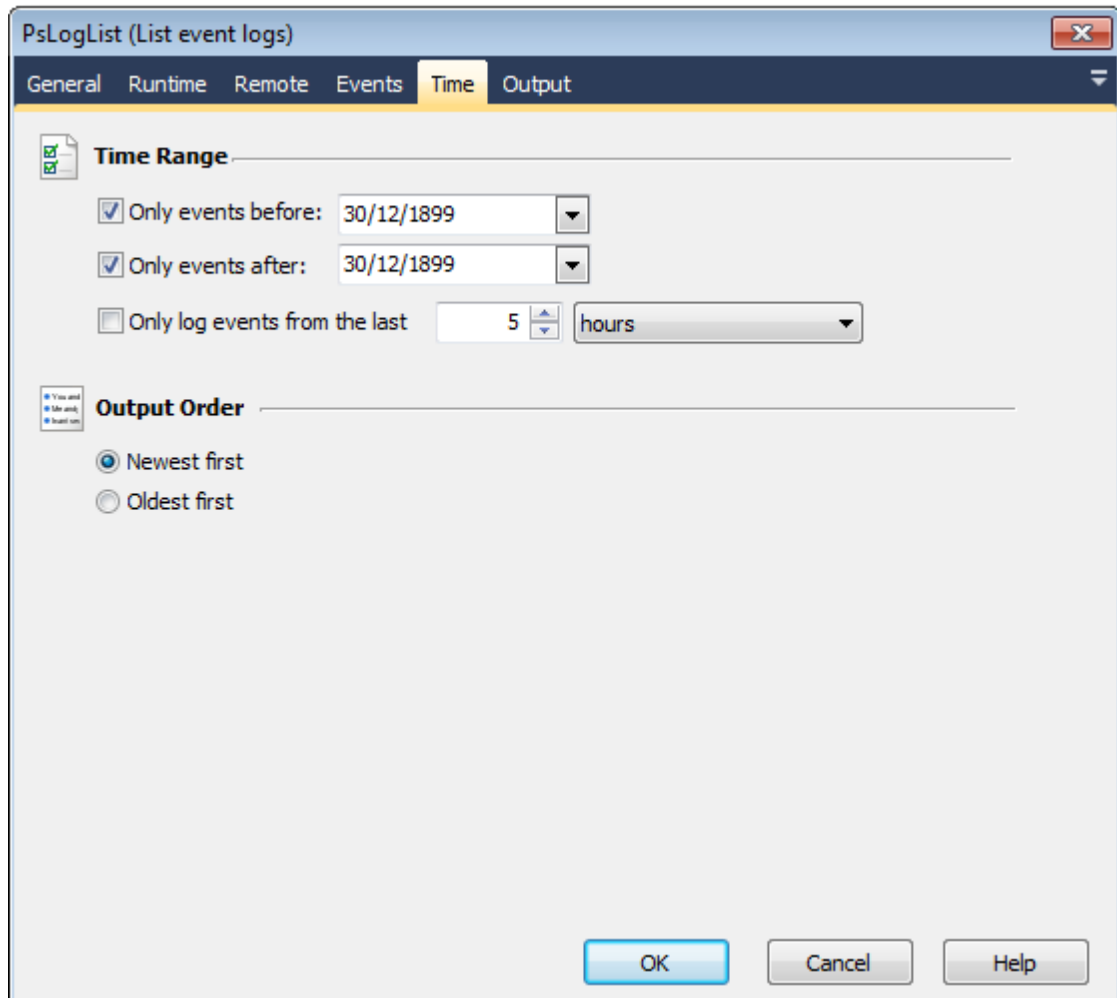
Specify the types of log messages you wish to be printed.

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.

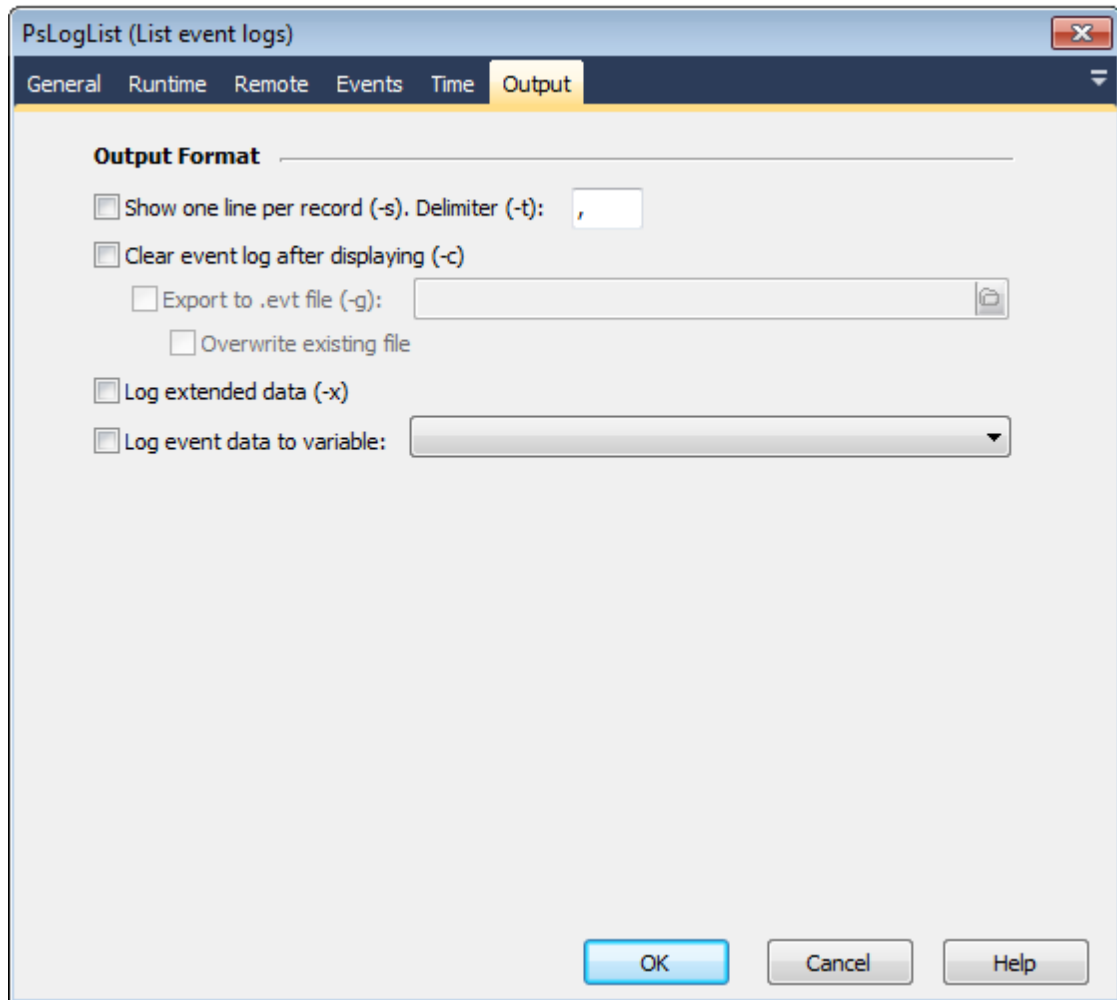


Time Range

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.

**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

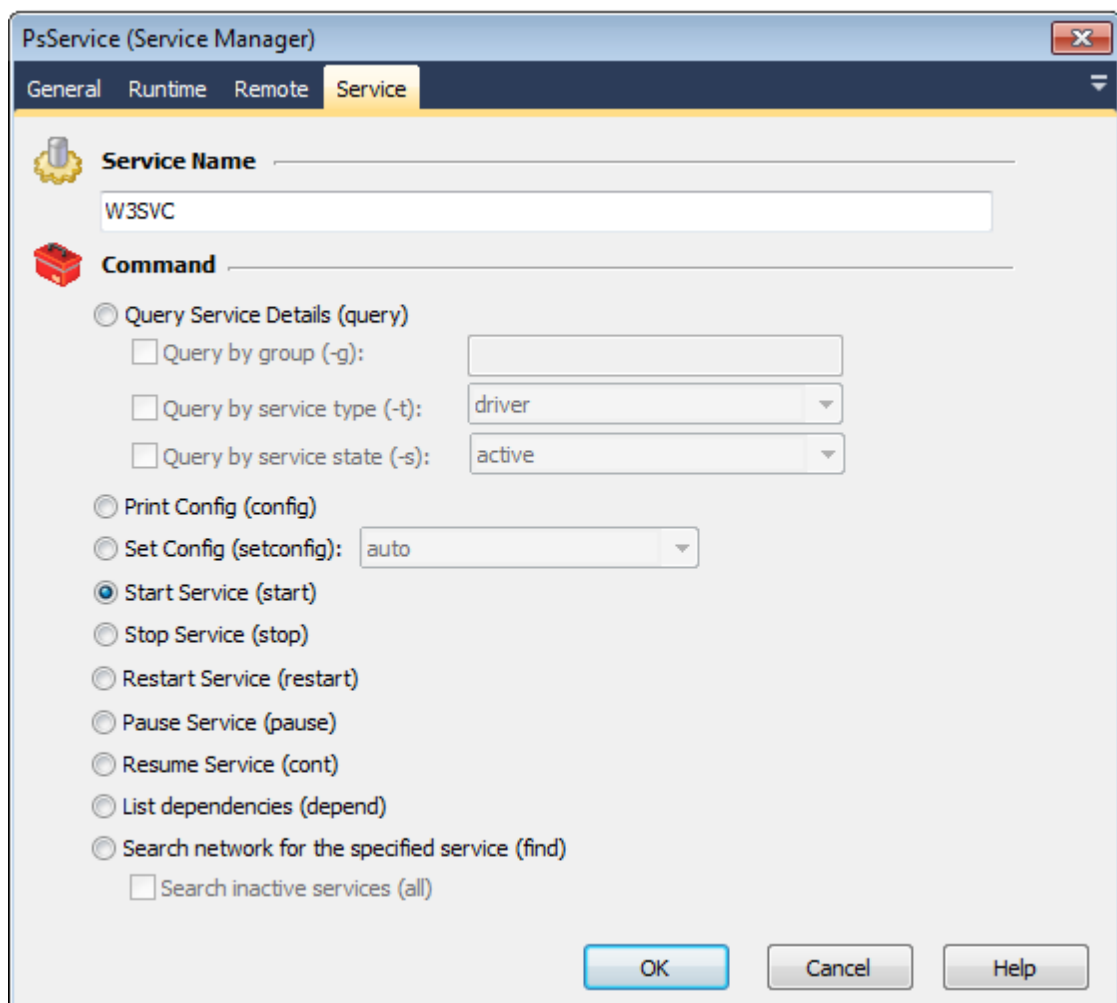
See the PsTools Common Options topic.

6.36.17.9 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, FinalBuilder 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.



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.

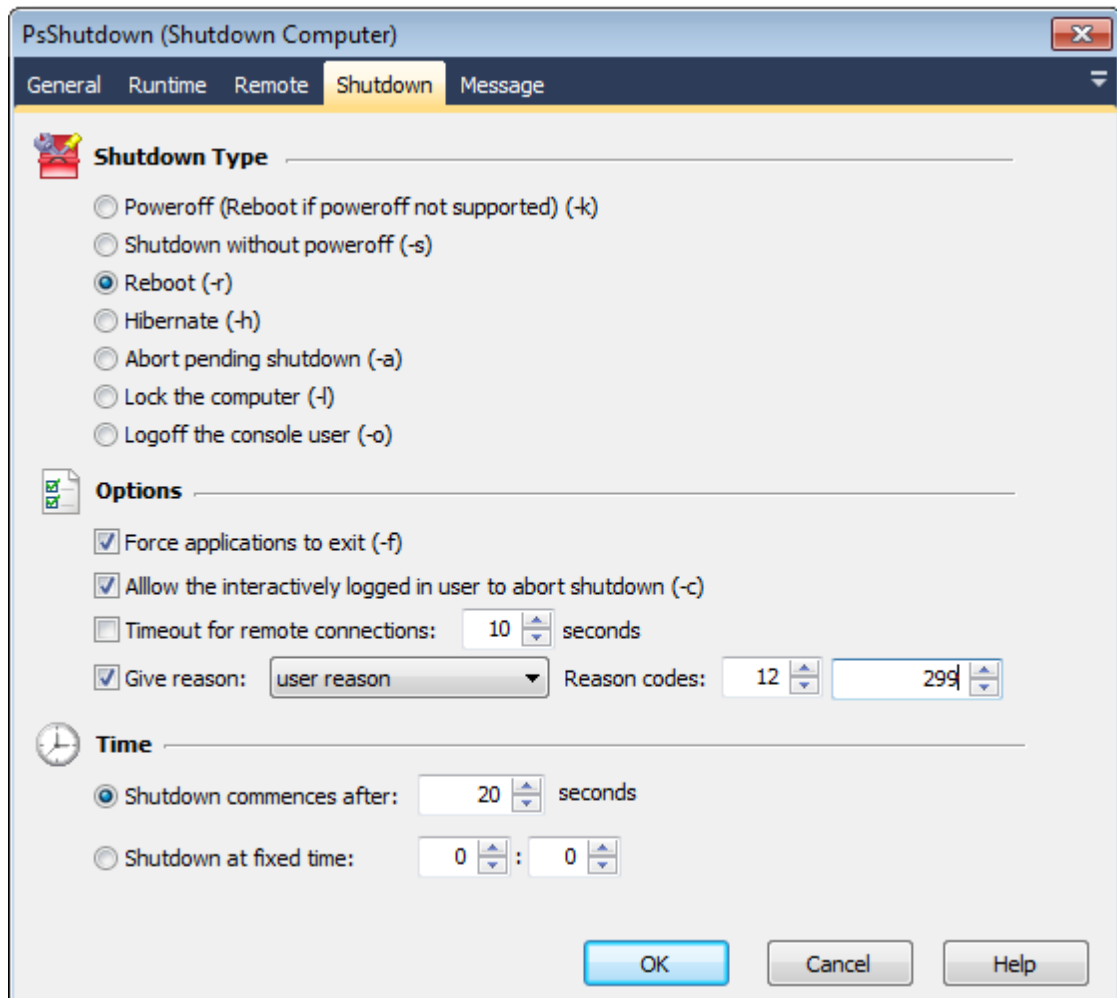
Check the "**Search inactive services (All)**" box to search for both running and inactive services (otherwise, PsService only searches for running instances of the service.)

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.1(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.



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

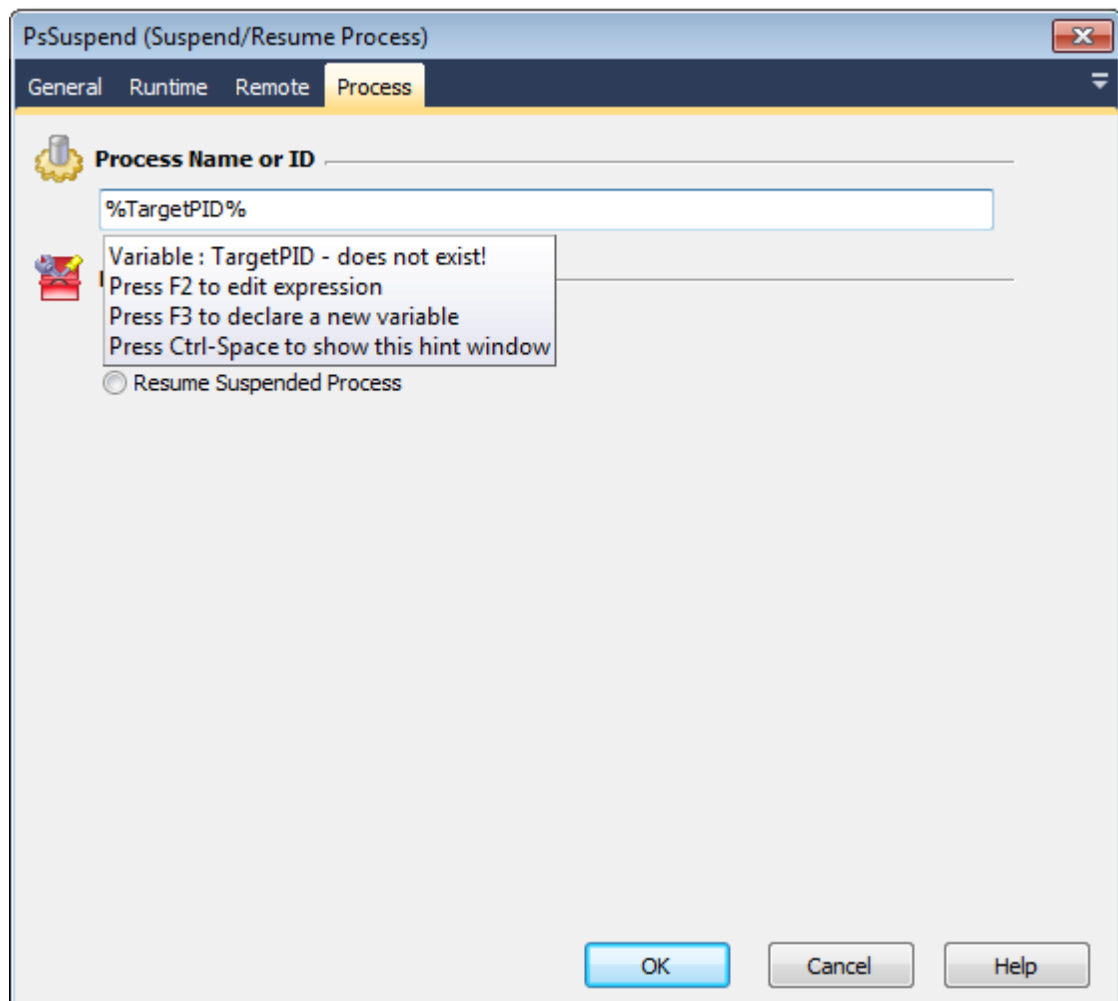
You can specify a delay (in seconds) before shutdown, or an absolute time (hours and minutes) at which to shutdown. Only applicable to those shutdown types marked (*) above.

"Remote" Tab

See the PsTools Common Options topic.

6.36.17.1 PsSuspend (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**

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).

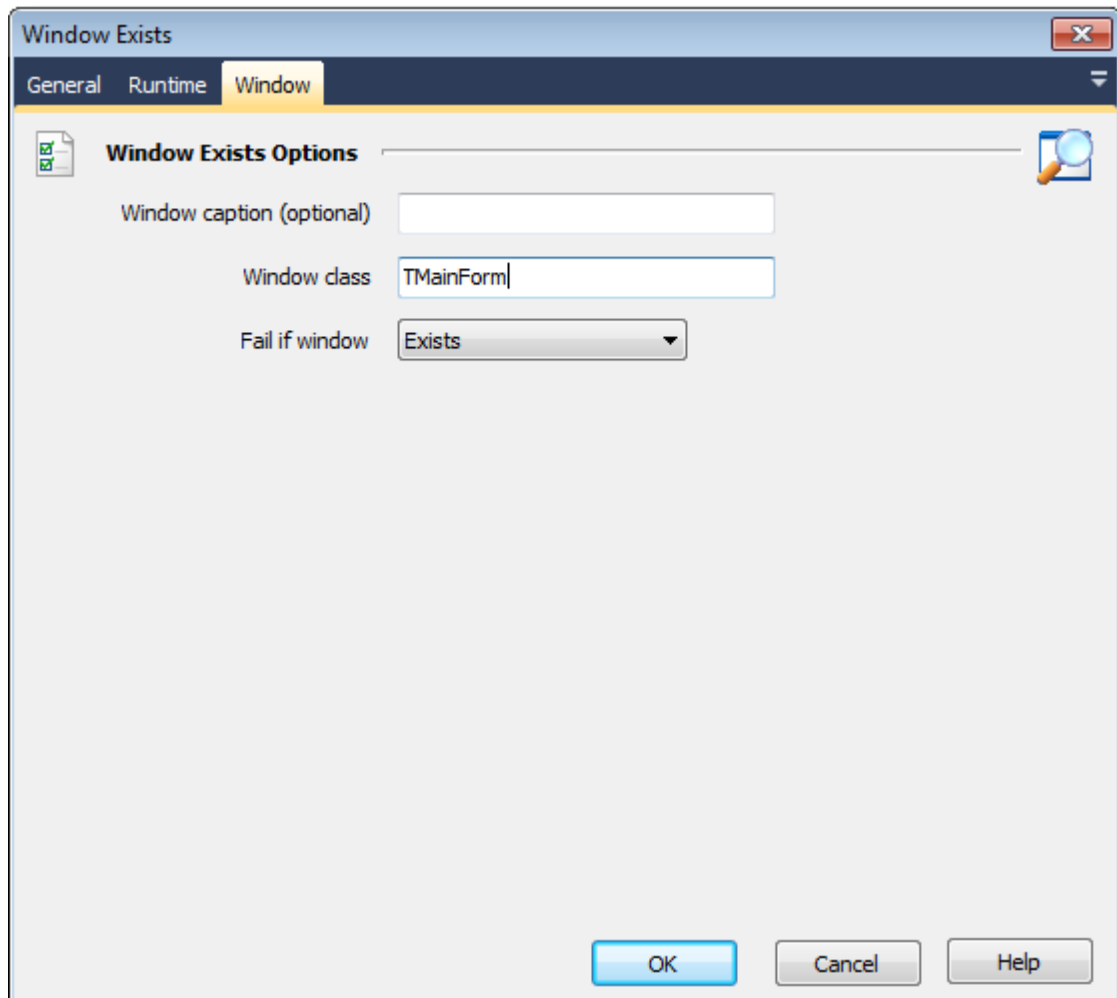
"Remote" Tab

See the PsTools Common Options topic.

6.36.18 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](#)

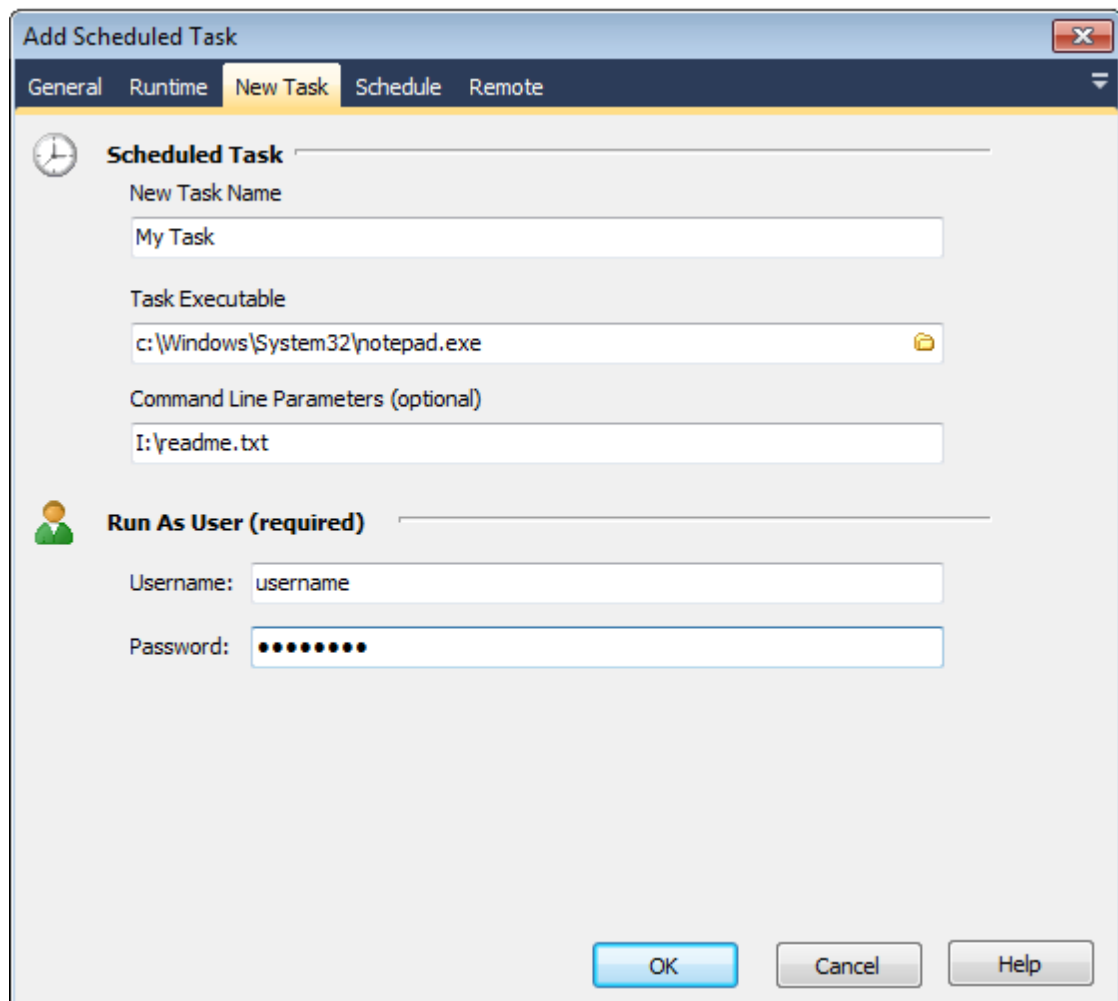


6.36.19 Windows Scheduler

6.36.19.1 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 FinalBuilder builds with the FinalBuilder Scheduler.



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.

The screenshot shows the 'Add Scheduled Task' dialog box with the 'Schedule' tab selected. The 'Start Time' section has three options: 'Absolute start time' (set to 2:04 PM), 'Start relative to time action is run' (selected, set to 24 hours and 0 minutes), and 'Specify start date' (set to 24/06/2010). The 'Specify end date' is also set to 24/06/2010. The 'Schedule Type' section has five options: 'Run Once Only' (selected), 'Run On Startup', 'Run On Login', 'Run On Idle', and 'Run Every' (set to 1 hour). The 'Schedule Options' section has an 'Idle time' of 1 minute and a grid of checkboxes for 'Run each day' (Mon-Sun) and 'Specify months' (Jan-Dec). The 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

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")

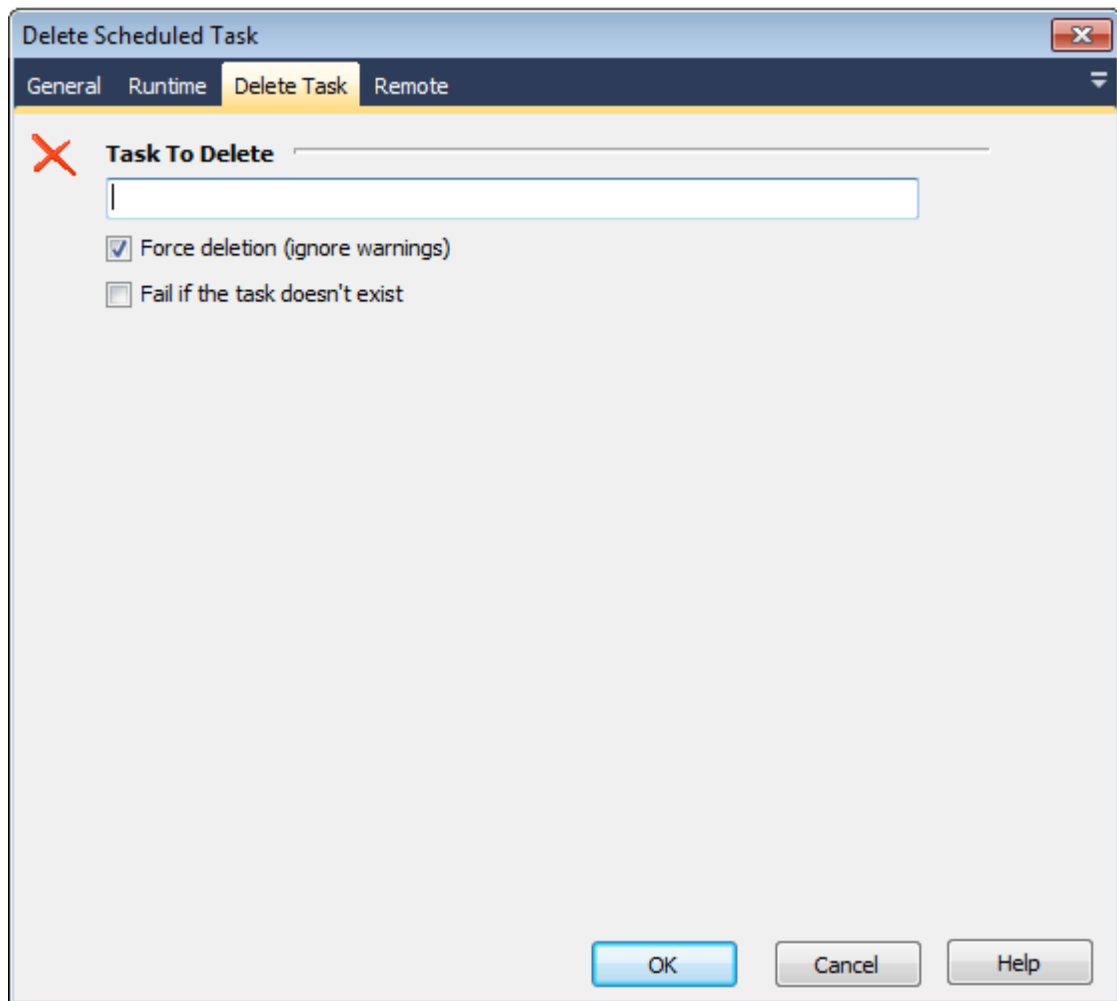
This option changes the behaviour of the monthly schedule so that the task will only run on the first day of certain months.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.36.19.2 Delete Scheduled Task

The Delete Scheduled Task action allows you to remove a scheduled task from the Windows Scheduler.



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

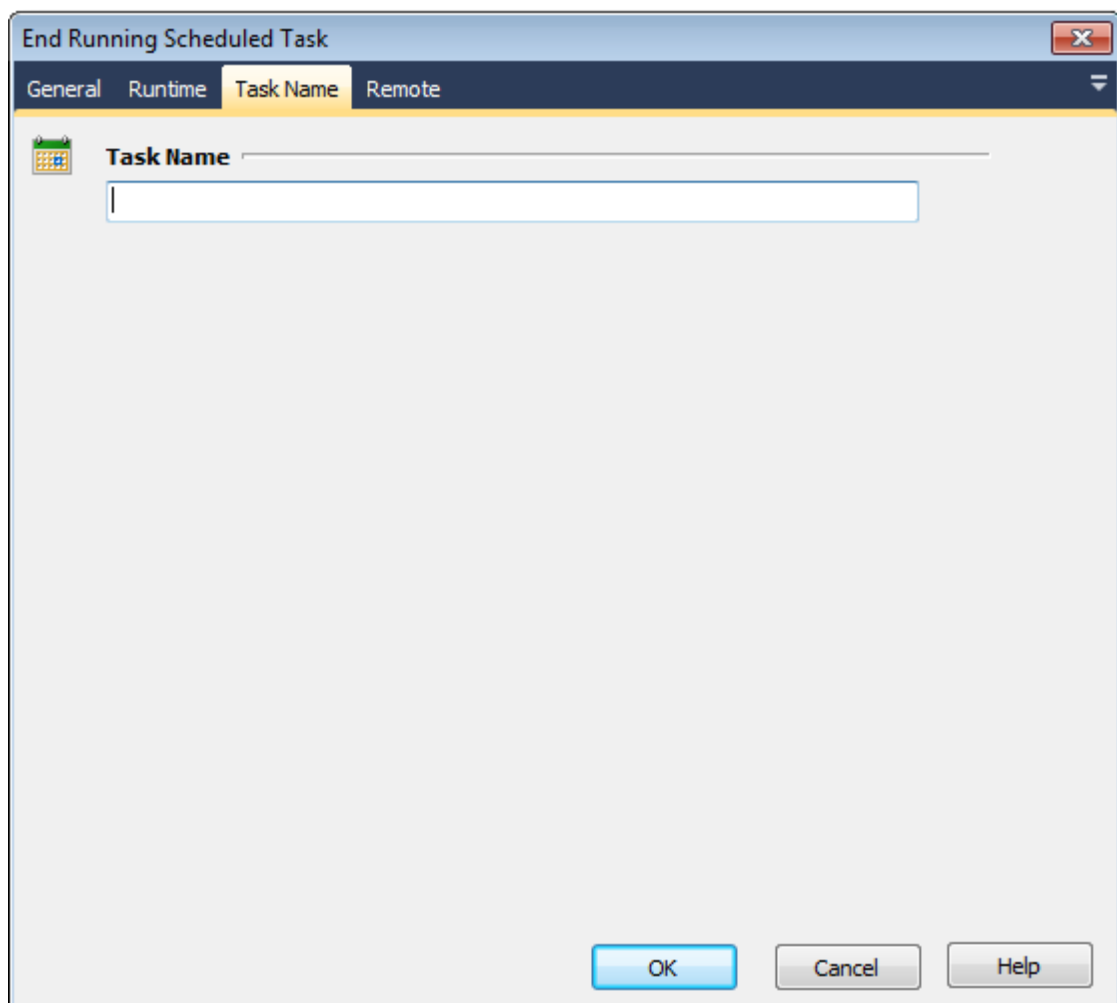
will be ignored and the action will succeed.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.36.19.3 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

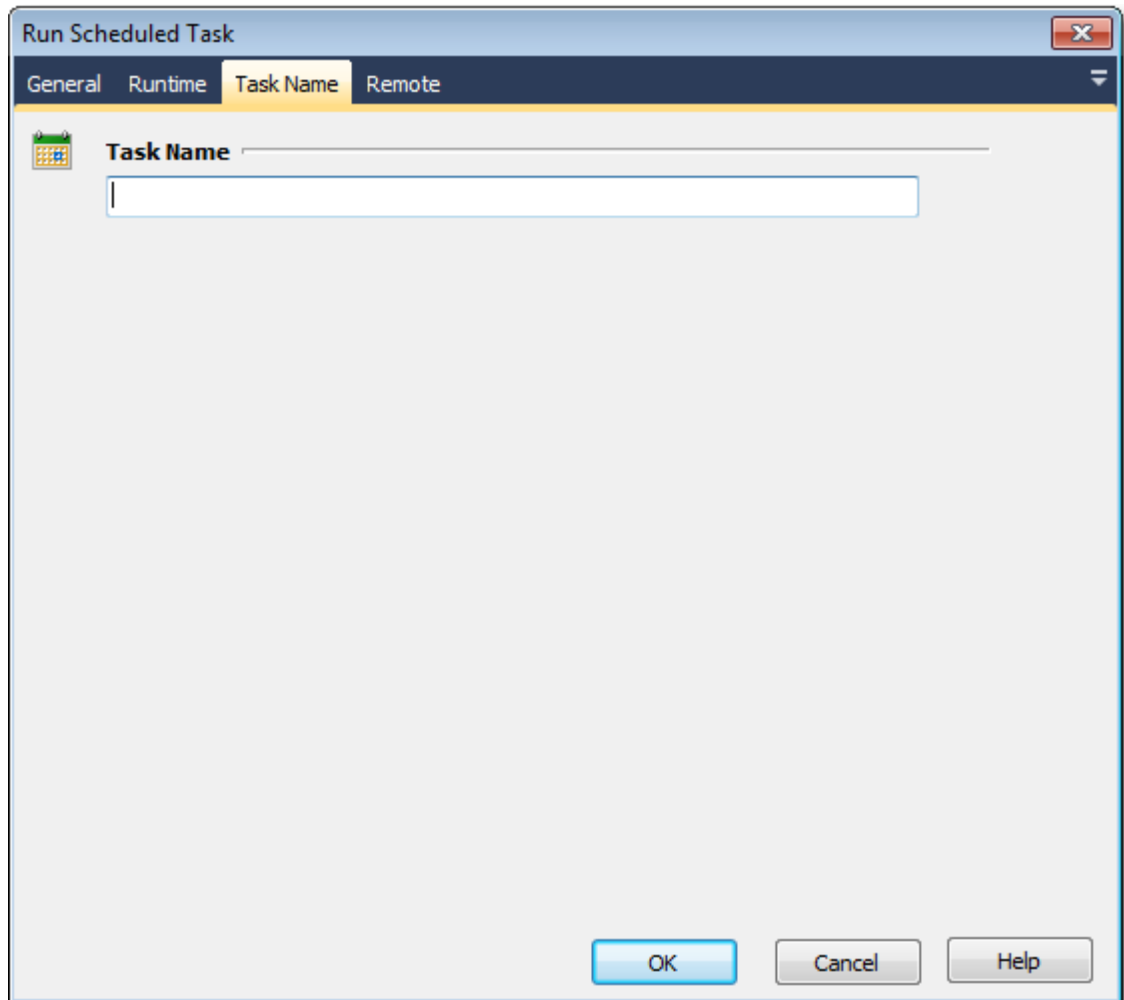
Enter the name of the running task to stop.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.36.19.4 Run Scheduled Task

The Run Scheduled Task action explicitly launches a scheduled task immediately.



Task Name

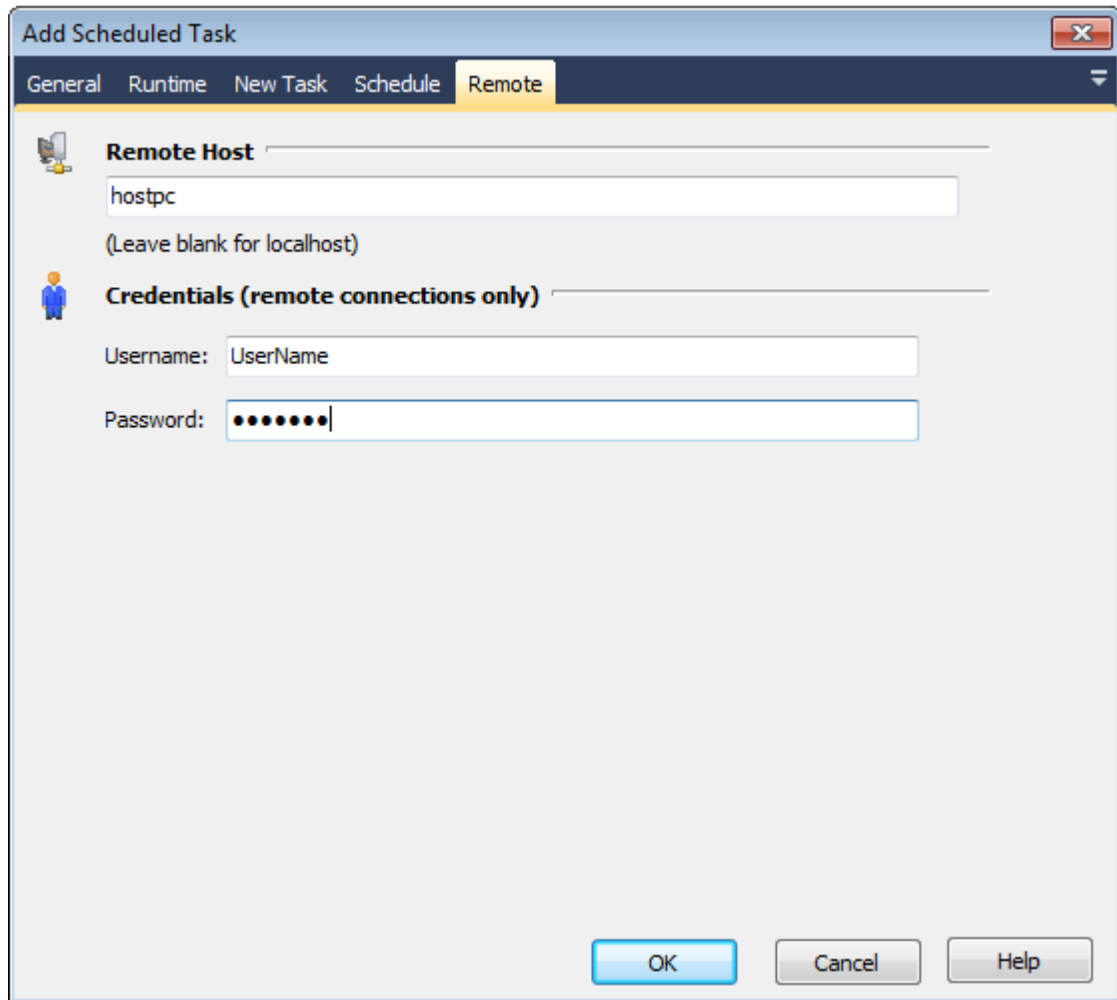
Enter the name of a task to run.

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

6.36.19.5 Scheduling Remote Tasks

All of the Windows Scheduler actions contain a Remote tab, which allows you to schedule tasks on remote machines:



The screenshot shows the 'Add Scheduled Task' dialog box with the 'Remote' tab selected. The 'Remote Host' field contains 'hostpc' and has a tooltip that says '(Leave blank for localhost)'. The 'Credentials (remote connections only)' section has a 'Username' field with 'UserName' and a 'Password' field with masked characters. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Add Scheduled Task

General Runtime New Task Schedule **Remote**

Remote Host
(Leave blank for localhost)

Credentials (remote connections only)

Username:

Password:

OK Cancel Help

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.

6.36.20 WMI Actions

6.36.20.1 WMI Kill Process Action

[FinalBuilder Professional Edition]

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.

The screenshot shows the 'WMI Kill Process' dialog box. It has a title bar with a close button. Below the title bar are three tabs: 'General', 'Runtime', and 'WMI Kill Process'. The 'WMI Kill Process' tab is selected. The dialog is divided into three main sections. The first section, 'Computer Name', has a computer icon, a text box, and a 'Load' button. The second section, 'Process', has a gear icon, two radio buttons ('Process ID' is selected), and two checkboxes ('Fail if more than one process found' and 'Fail if no processes are found' are both checked). The third section, 'Credentials', has a lock icon, a 'Username' text box (containing 'UserName'), and a 'Password' text box (containing dots). At the bottom are 'OK', 'Cancel', and 'Help' buttons.

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.

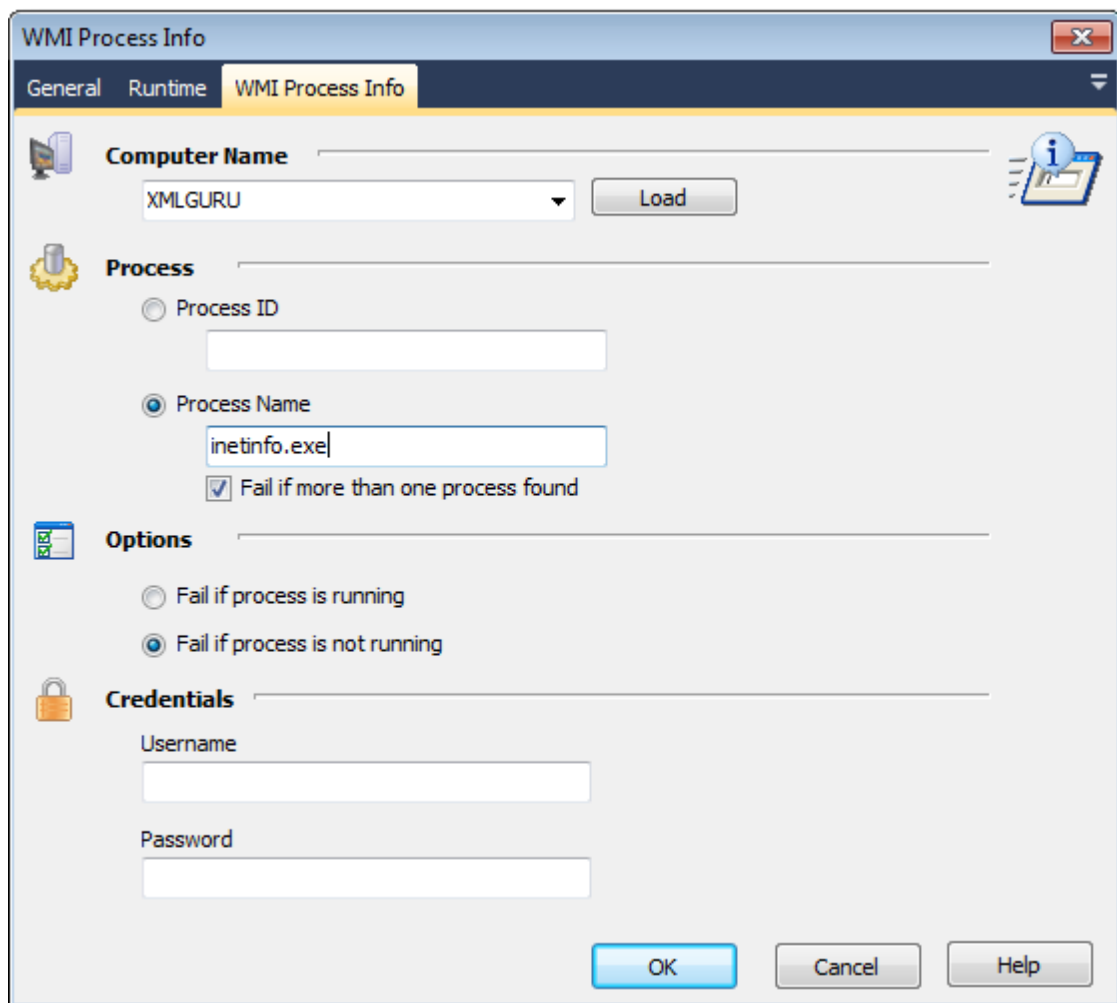
Credentials

Specify credentials for the WMI connection. If no credentials are specified, WMI uses the currently logged in user's Windows Authentication credentials.

6.36.20.2 WMI Process Info Action

[FinalBuilder Professional Edition]

This action allows you to interrogate a process for information, or just check if the process is running or not.



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;

```

6.36.20.3 WMI Process Iterator

[FinalBuilder Professional Edition]

The WMI process iterator allows you to iterate through a list of processes running on any Windows machine. Each process can be identified by its 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

General Runtime **WMI Process Iterator**

Computer Name _____

Load

Credentials _____
Username

Password

Variable to set _____

☐ 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

OK Cancel Help

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 FinalBuilder 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.

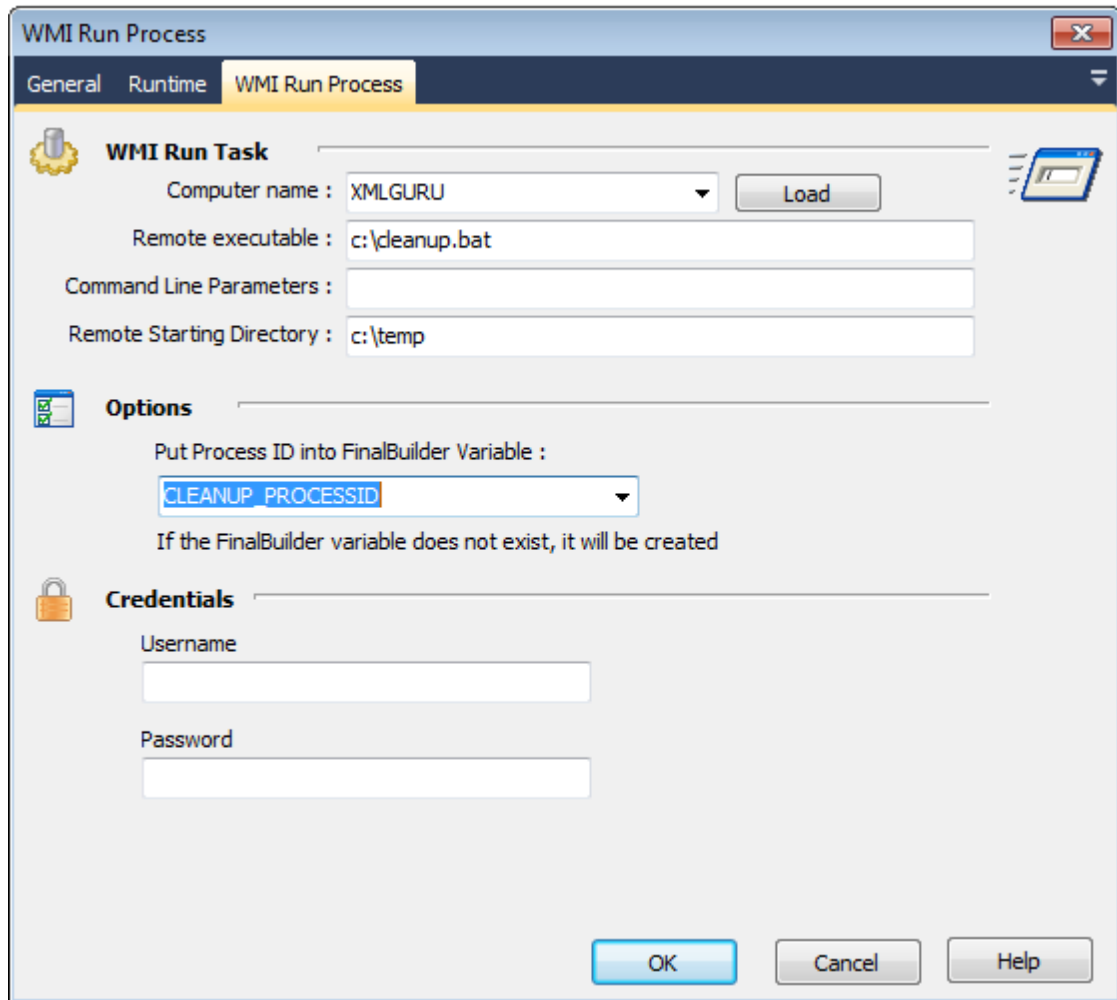
Fail if no processes are found

If this option is selected, the action will fail outright if no processes are found which match the chosen criteria.

6.36.20.4 WMI Run Process Action

[FinalBuilder Professional Edition]

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.



The image shows a 'WMI Run Process' dialog box with three tabs: 'General', 'Runtime', and 'WMI Run Process'. The 'WMI Run Process' tab is active. It contains three sections: 'WMI Run Task', 'Options', and 'Credentials'. The 'WMI Run Task' section has fields for 'Computer name' (set to 'XMLGURU'), 'Remote executable' (set to 'c:\cleanup.bat'), 'Command Line Parameters' (empty), and 'Remote Starting Directory' (set to 'c:\temp'). There is a 'Load' button next to the 'Computer name' field. The 'Options' section has a checkbox 'Put Process ID into FinalBuilder Variable' which is checked, and a dropdown menu showing 'CLEANUP_PROCESSID'. Below this is a note: 'If the FinalBuilder variable does not exist, it will be created'. The 'Credentials' section has fields for 'Username' and 'Password'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

WMI Run Process

General Runtime **WMI Run Process**

WMI Run Task

Computer name : XMLGURU Load

Remote executable : c:\cleanup.bat

Command Line Parameters :

Remote Starting Directory : c:\temp

Options

Put Process ID into FinalBuilder Variable :

CLEANUP_PROCESSID

If the FinalBuilder variable does not exist, it will be created

Credentials

Username

Password

OK Cancel Help

You can choose to save the process ID in a FinalBuilder variable so that it may be used in later actions such as the WMI Kill Process Action.

6.37 XML Actions

XML Namespaces

A common issue that occurs when using the FinalBuilder XML actions is the following error:

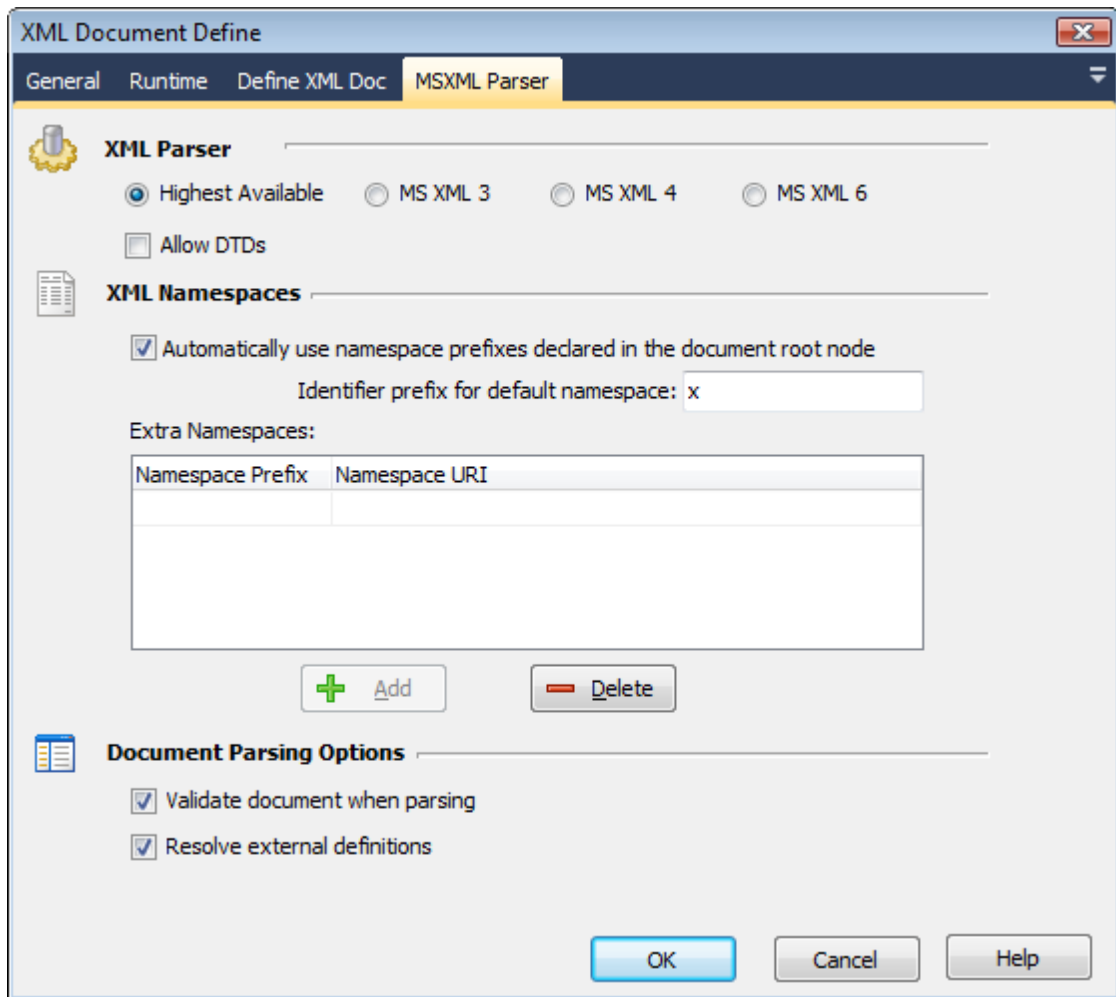
```
XPath returned no node : /Project/Import
```

When using an XML document that 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.



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.

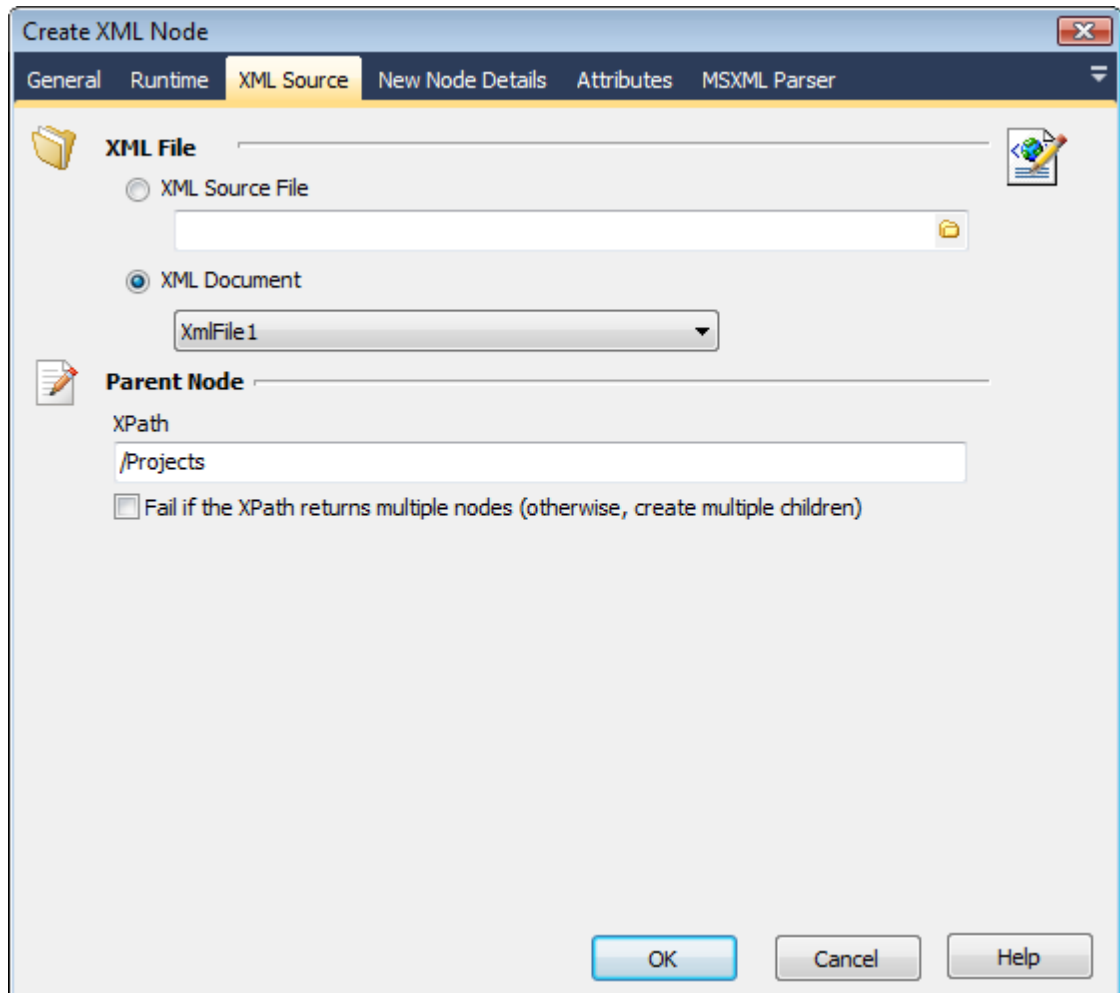
6.37.1 Altova DiffDog Action

The Altova DiffDog Action allows you to automate XML file comparisons using Altova DiffDog as part of your project.

For more information on Altova DiffDog, see here: http://www.altova.com/products/diffdog/diff_merge_tool.html

6.37.2 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.

Create XML Node

General Runtime XML Source **New Node Details** Attributes MSXML Parser

New Node

Node Name:

☒ Node Text: ☐ Child xml:

☐ Append to child nodes ☐ Insert as first child node

☐ Insert Before Child At XPath ☒ Insert After Child At XPath

Insert At XPath:

☐ 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

OK Cancel Help

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.

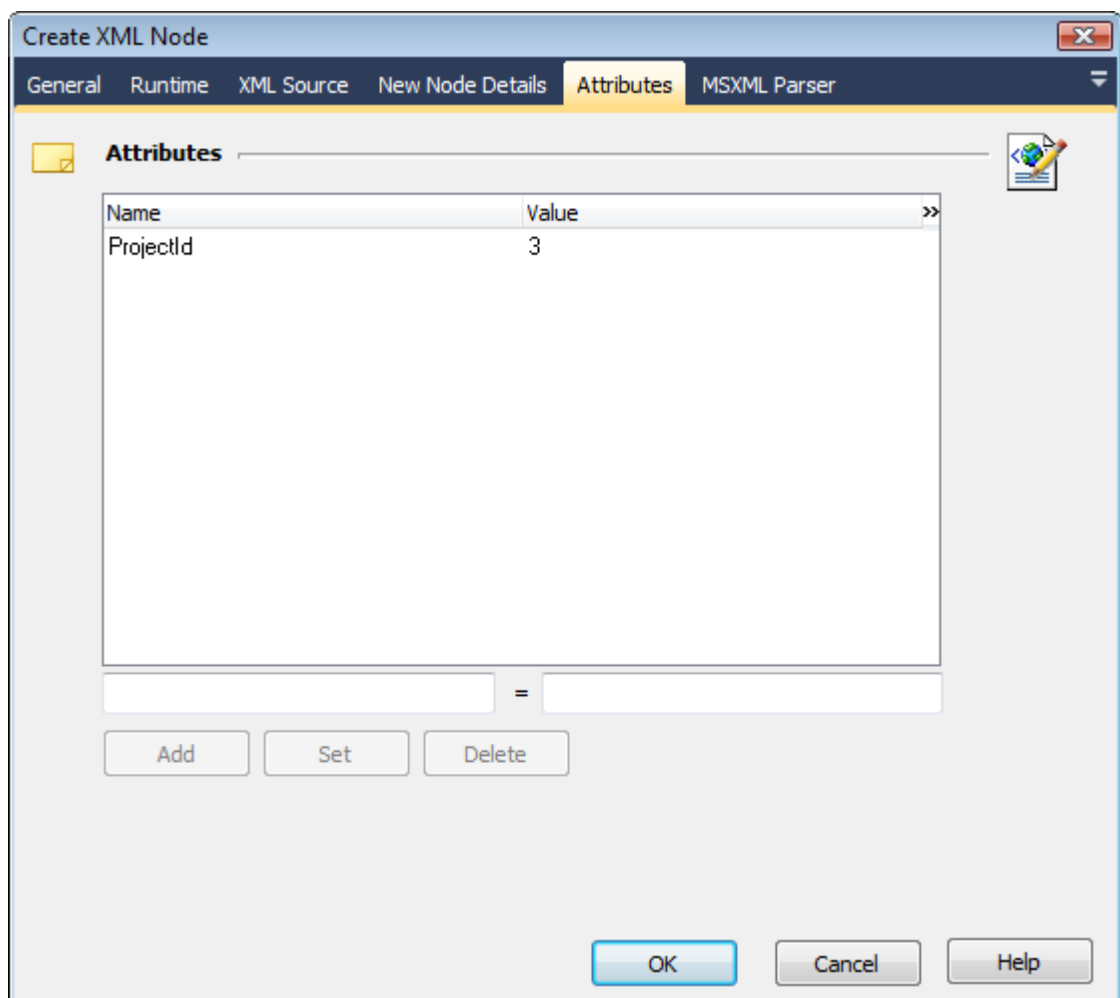
Note that if multiple child nodes with the same name exist, all of them will be erased but only one will be created.

"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.



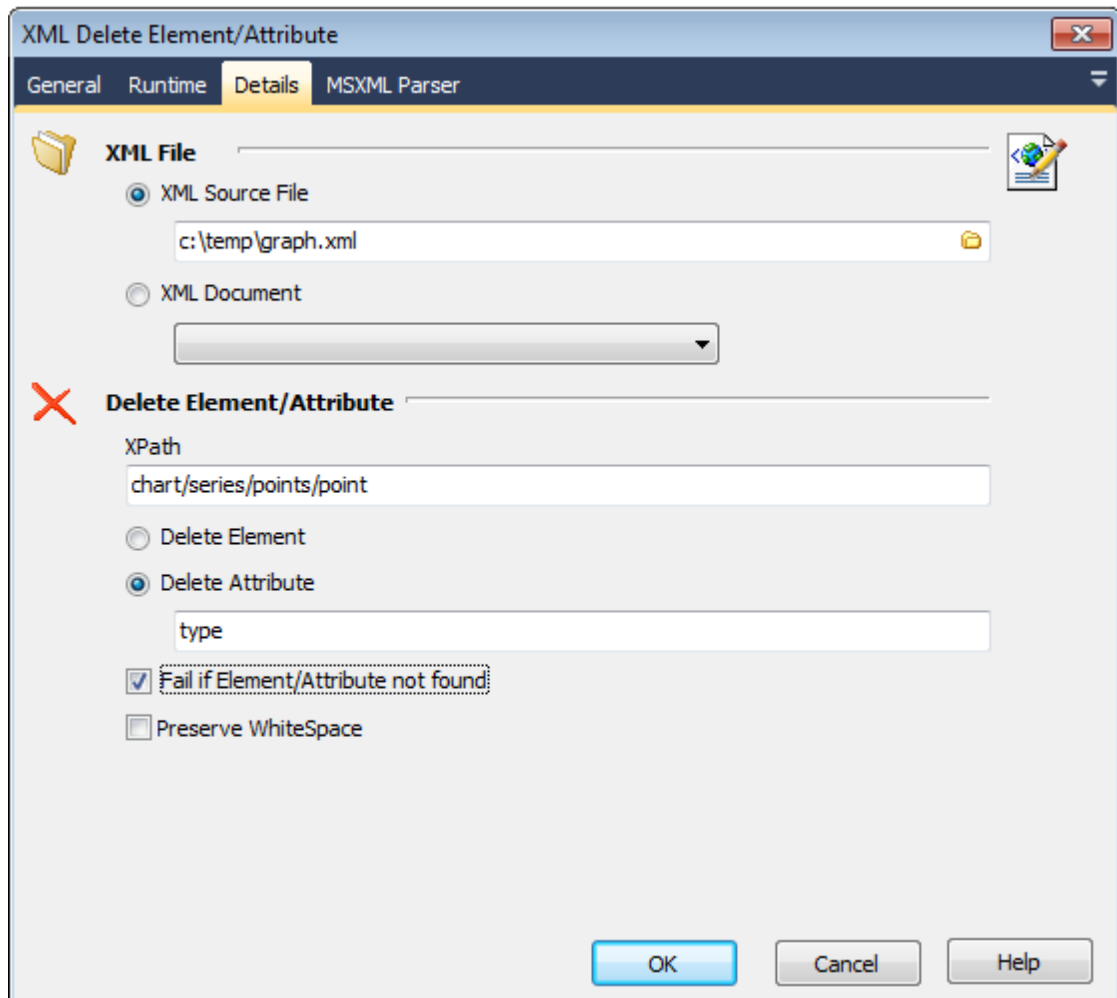
The attributes list shows attribute and value names for the new node. To add a new attribute, type its name and value into the edit fields and then click the Add button.

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.

6.37.3 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 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

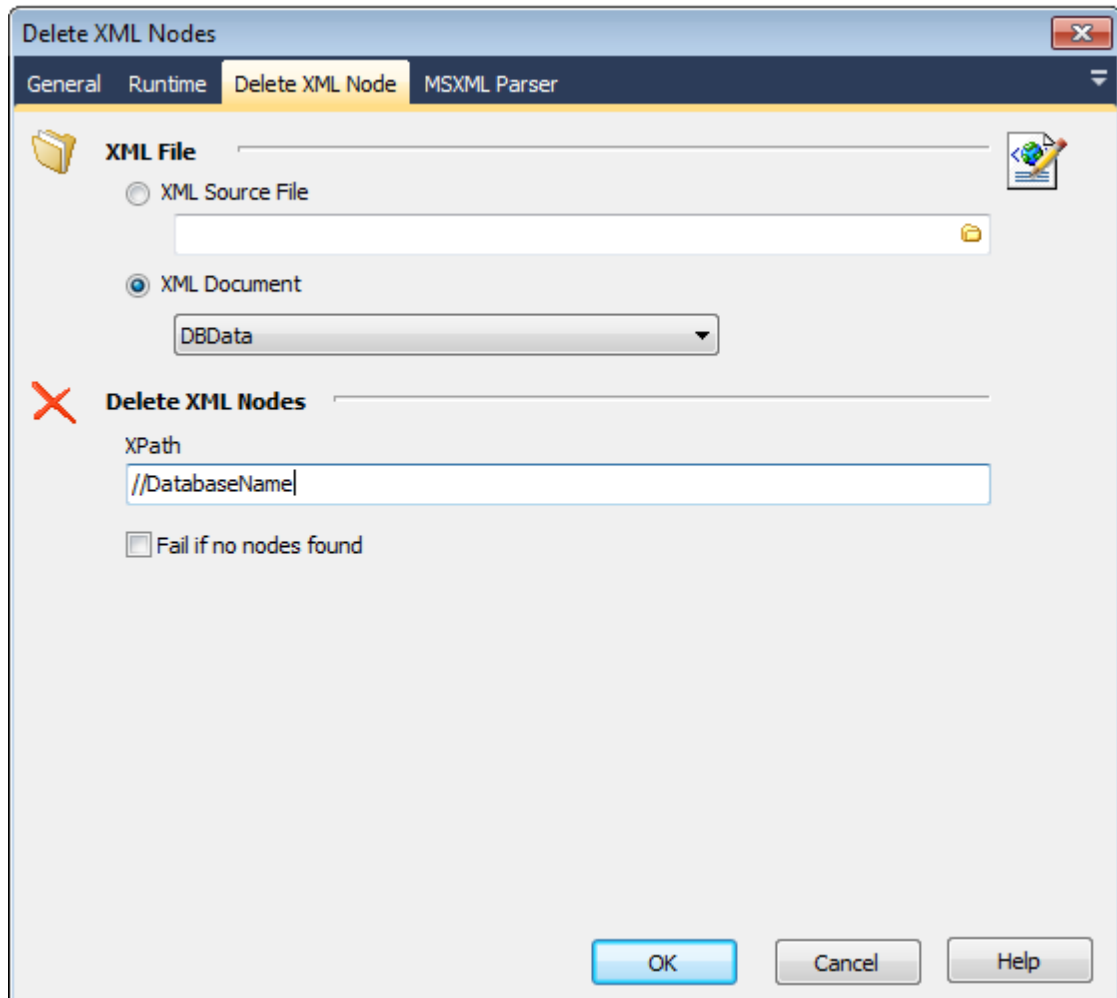
Preserve WhiteSpace - whitespace in the specified xml source file is preserved when the file is written to disk

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.

6.37.4 Delete XML Nodes

This action allows you to delete a node or set of nodes in an XML document.



XML File

Specify an XML file or document to to remove nodes from.

Delete XML Nodes

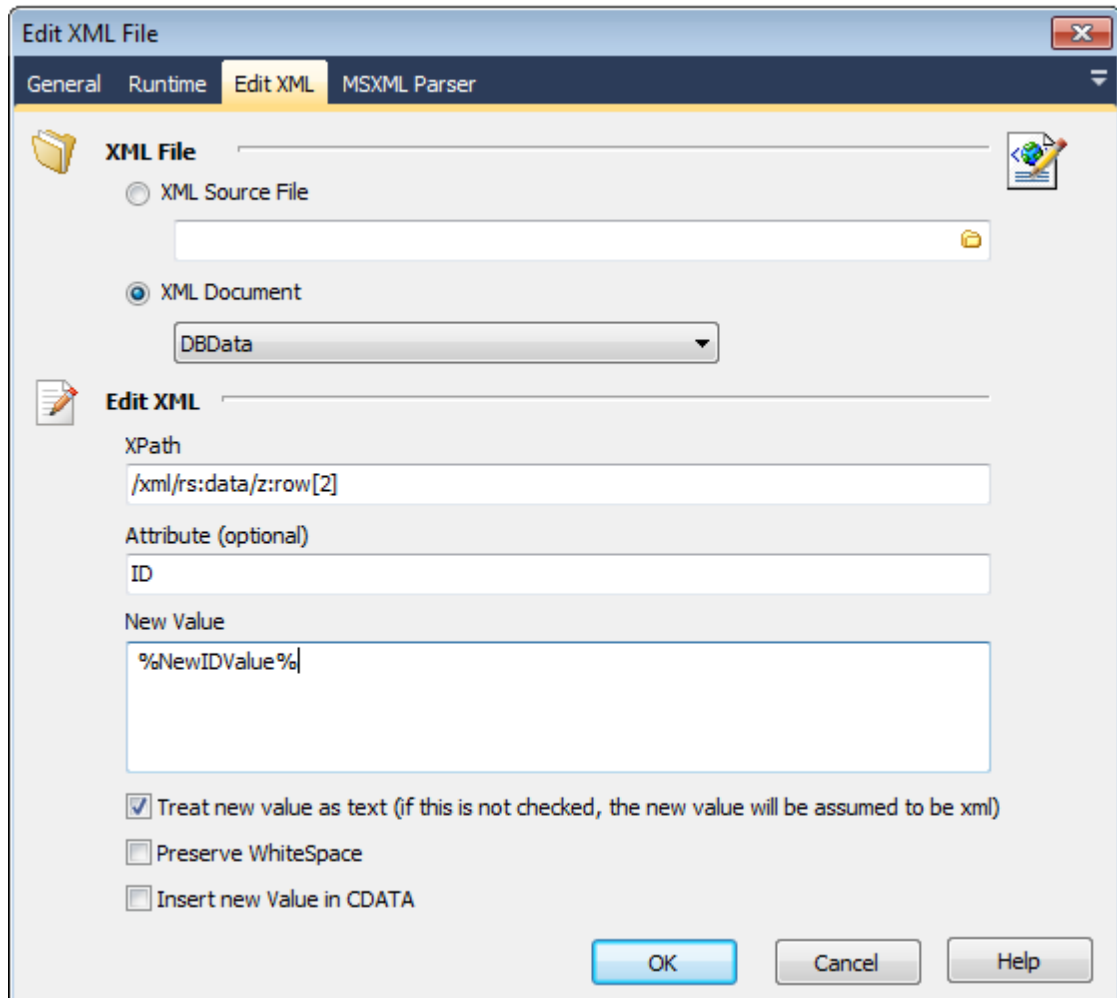
Specify the XPath to the node or nodes you wish to delete. Check the 'Fail if no nodes found' checkbox if you want the action to fail if no nodes match the XPath.

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.

6.37.5 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 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

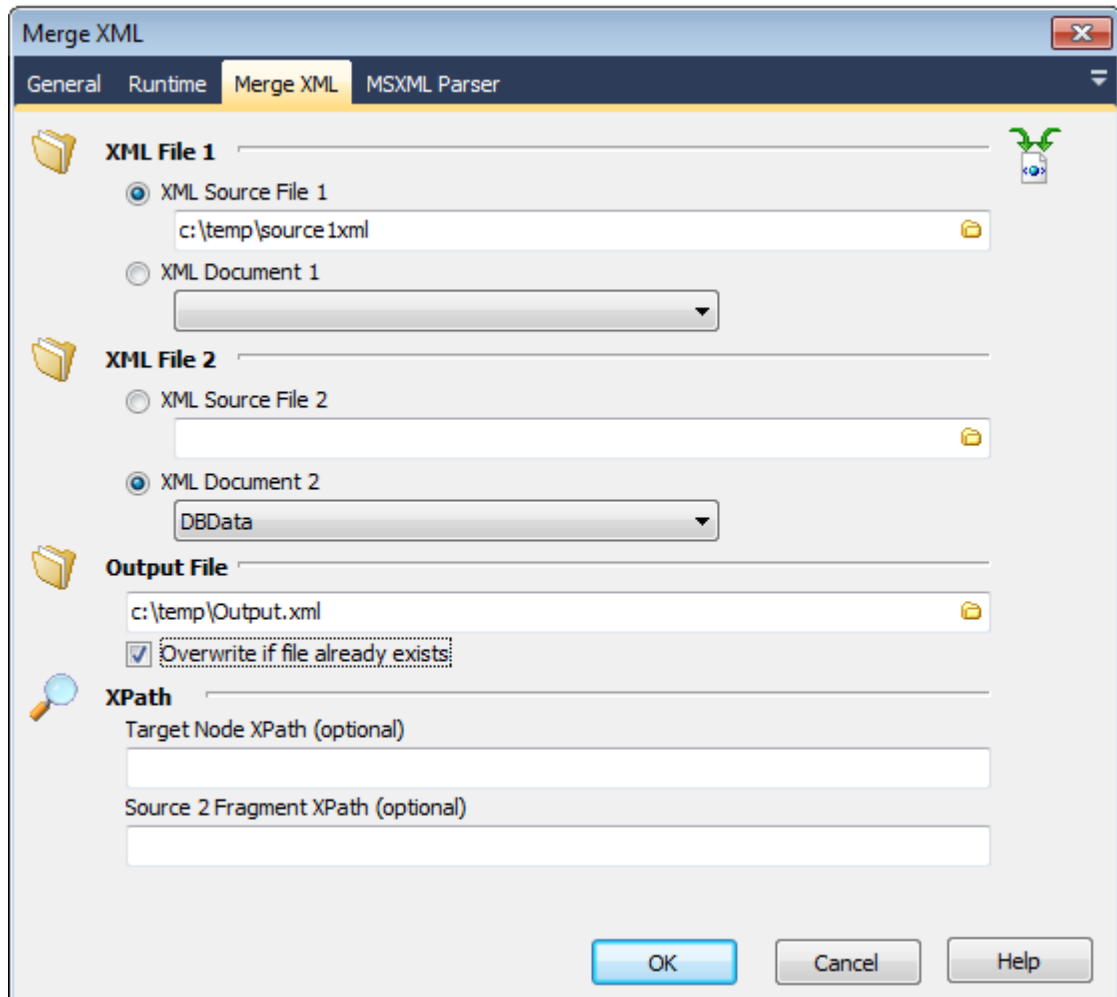
Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

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.

6.37.6 Merge XML Action

This action merges two XML documents. Source 2 will be merged into Source 1 and saved as a new document.



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

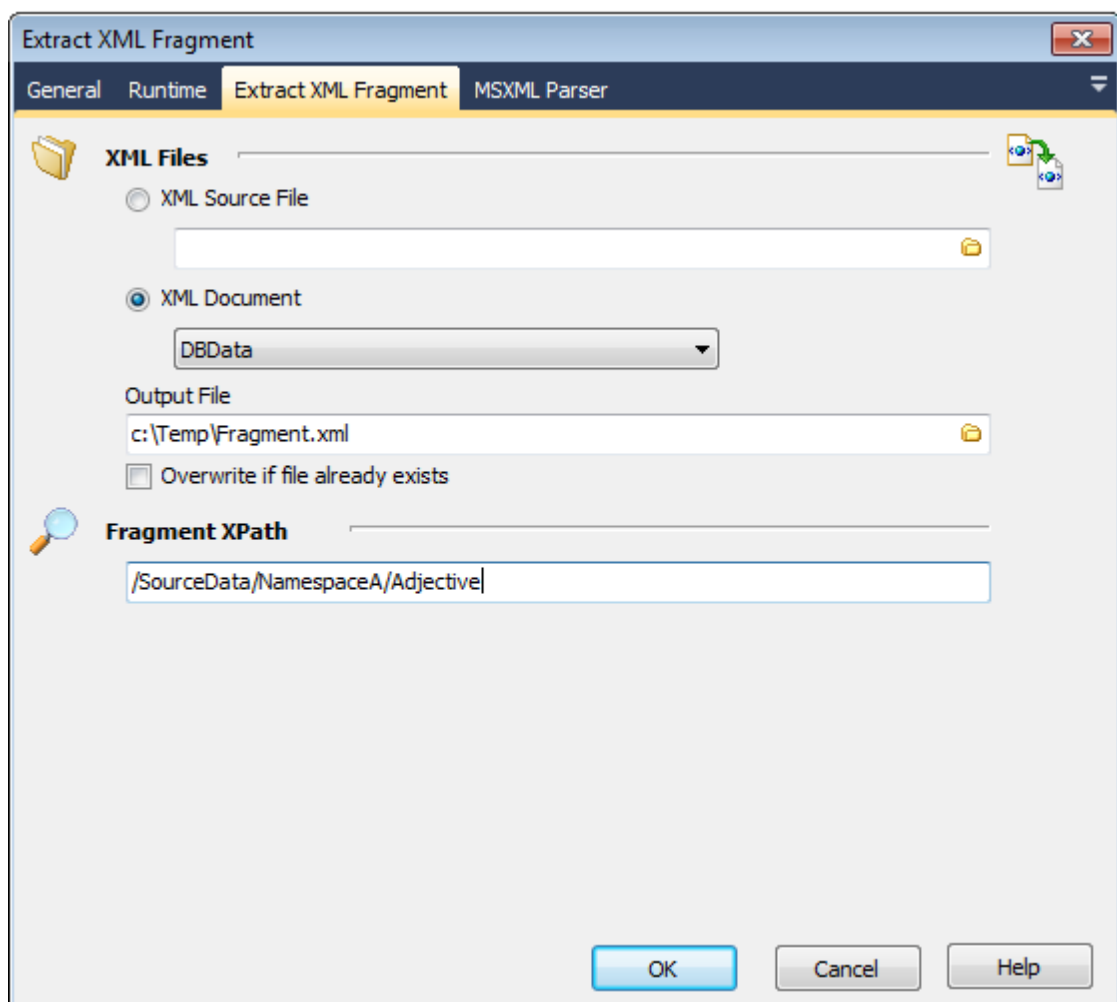
merge rather than the whole document.

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.

6.37.6.1 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

Specify a source file (or document), and a destination file to write the fragment to.

Fragment XPath

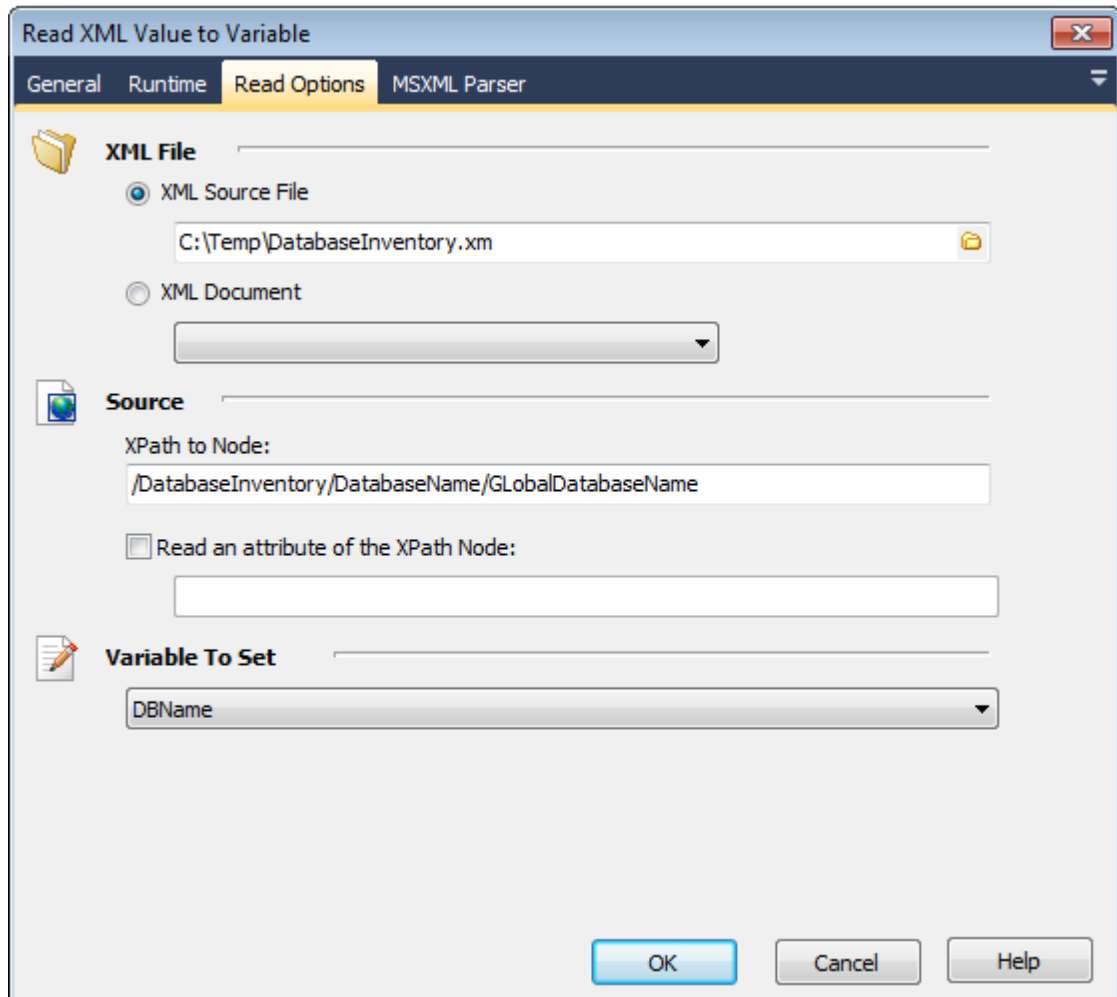
Specify the XPath to the nodes you wish to extract.

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.

6.37.7 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 FinalBuilder variable.



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, FinalBuilder 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 FinalBuilder 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.

6.37.8 Transform XML

This action performs an XSL Transform using the Microsoft XML parser.

Transform XML

General Runtime **Transform XML** Transform Options MSXML Parser

XML File

☒ XML Source File

☐ XML Document

XSLT and Output file

XSLT File

Output File

☐ Overwrite if file already exists

XSLT Parameters

Parameter	Value

'Insert' or 'Down' to add a new row

OK Cancel Help

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.

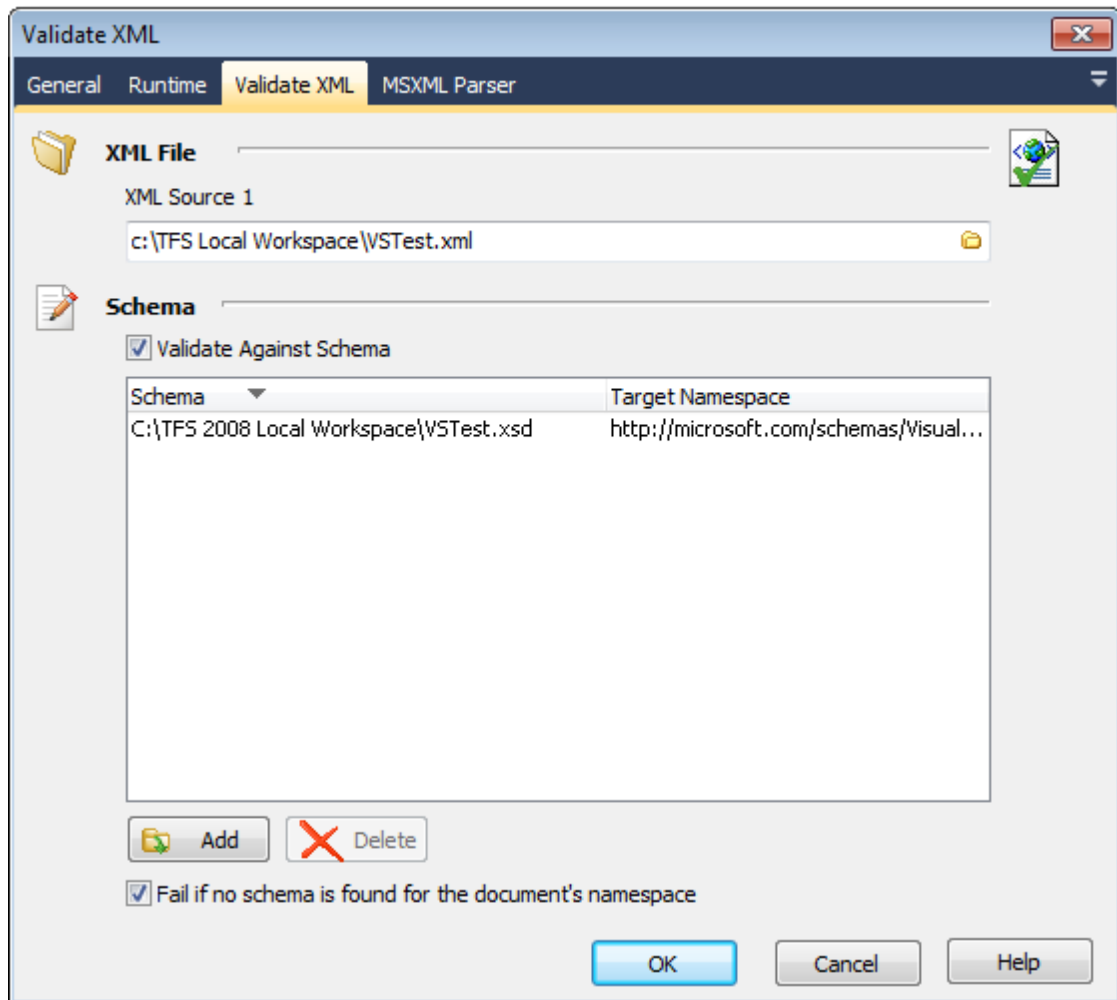
MSXML Parser Page

Allows you to set some options to be used when the parser loads the XML. See the XML Parser Options topic for details.

6.37.9 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.



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.

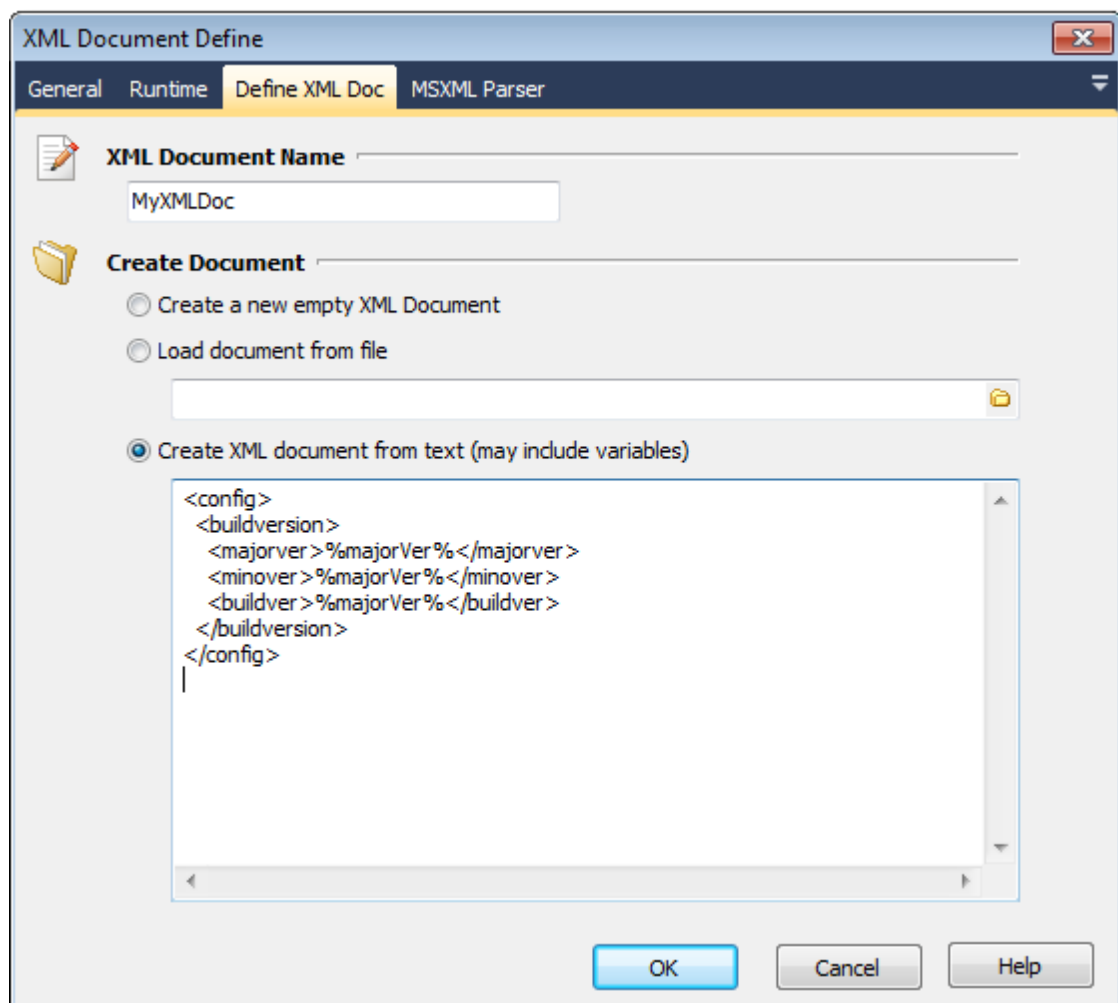
6.37.10 XML Document Define

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 - 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

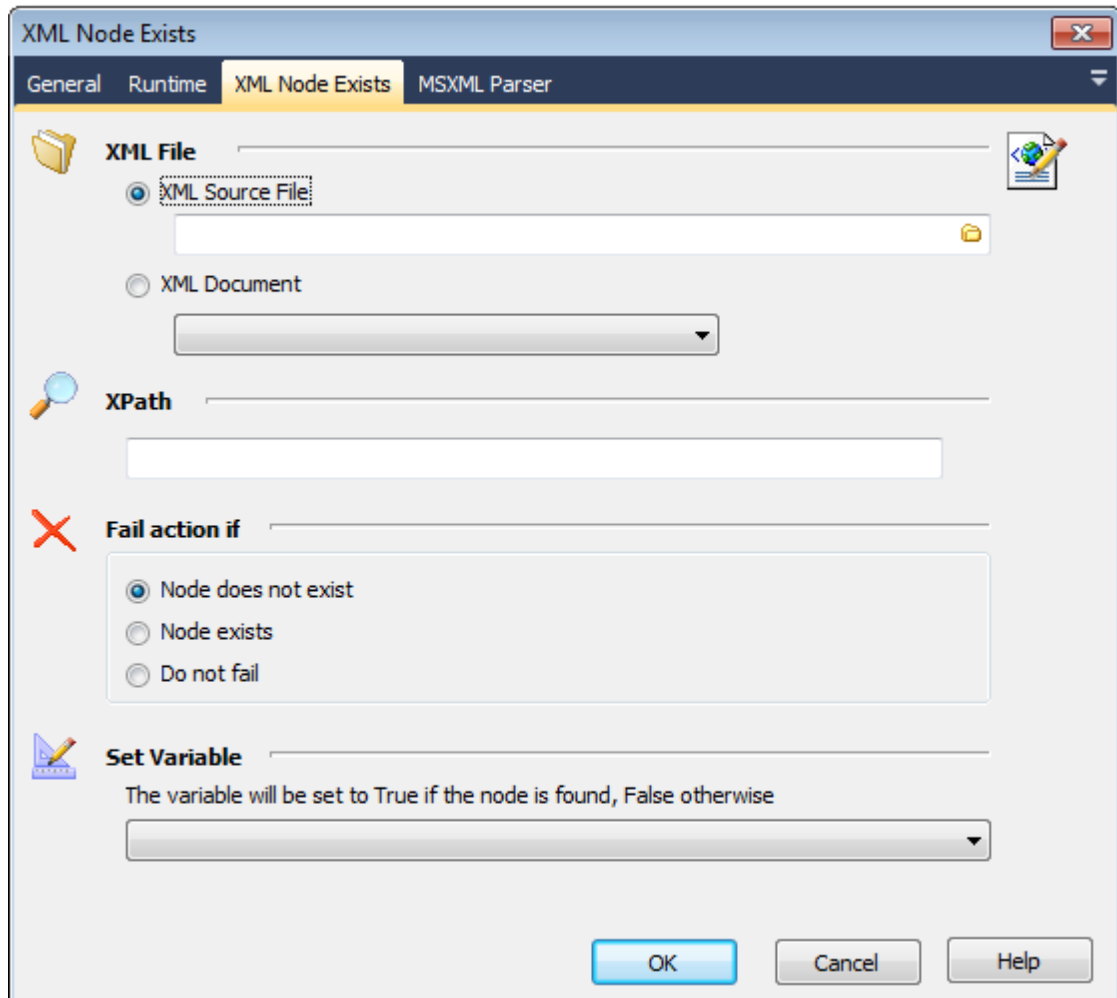
MSXML Parser Page

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.

6.37.11 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

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](http://www.w3schools.com/xpath/).

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

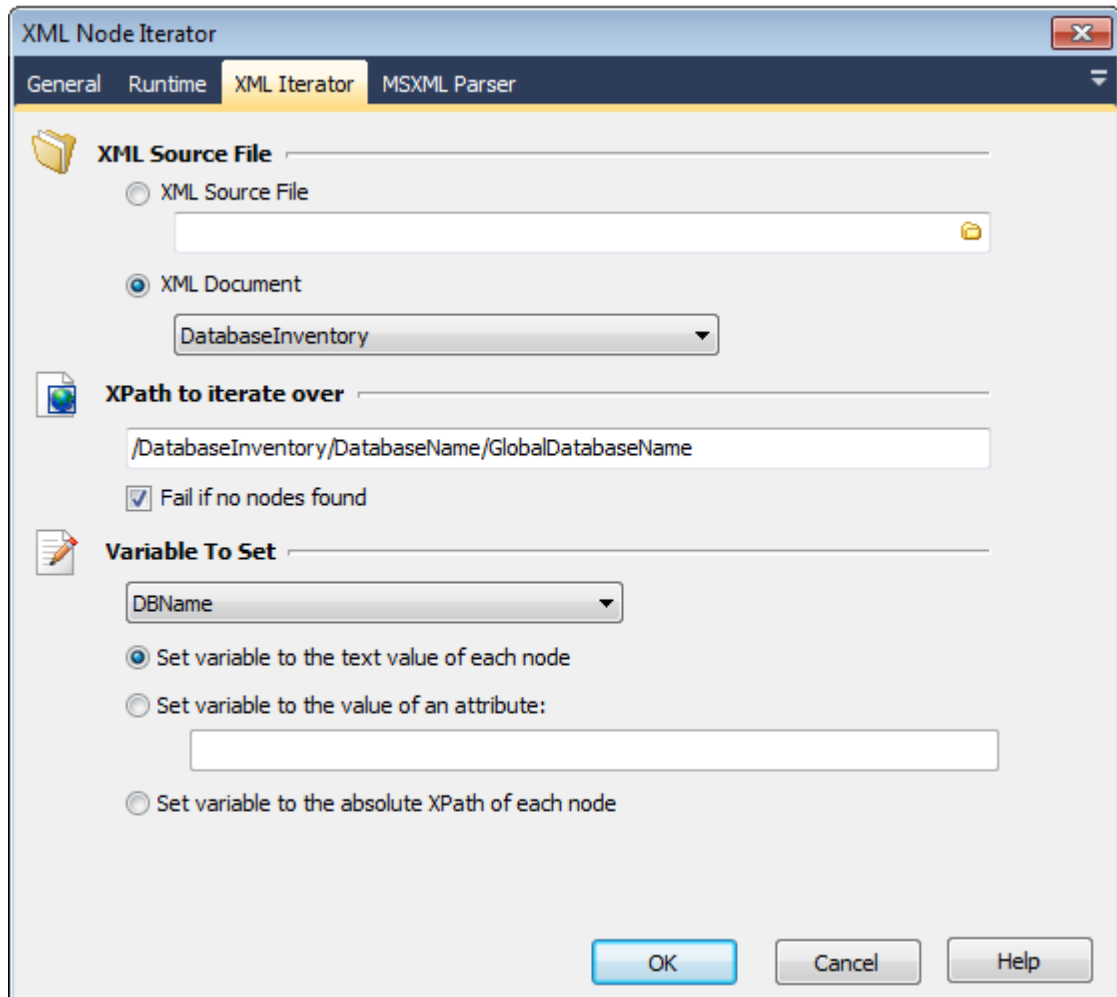
The result of the search (True or False) can be optionally written to a variable.

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.

6.37.12 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

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**

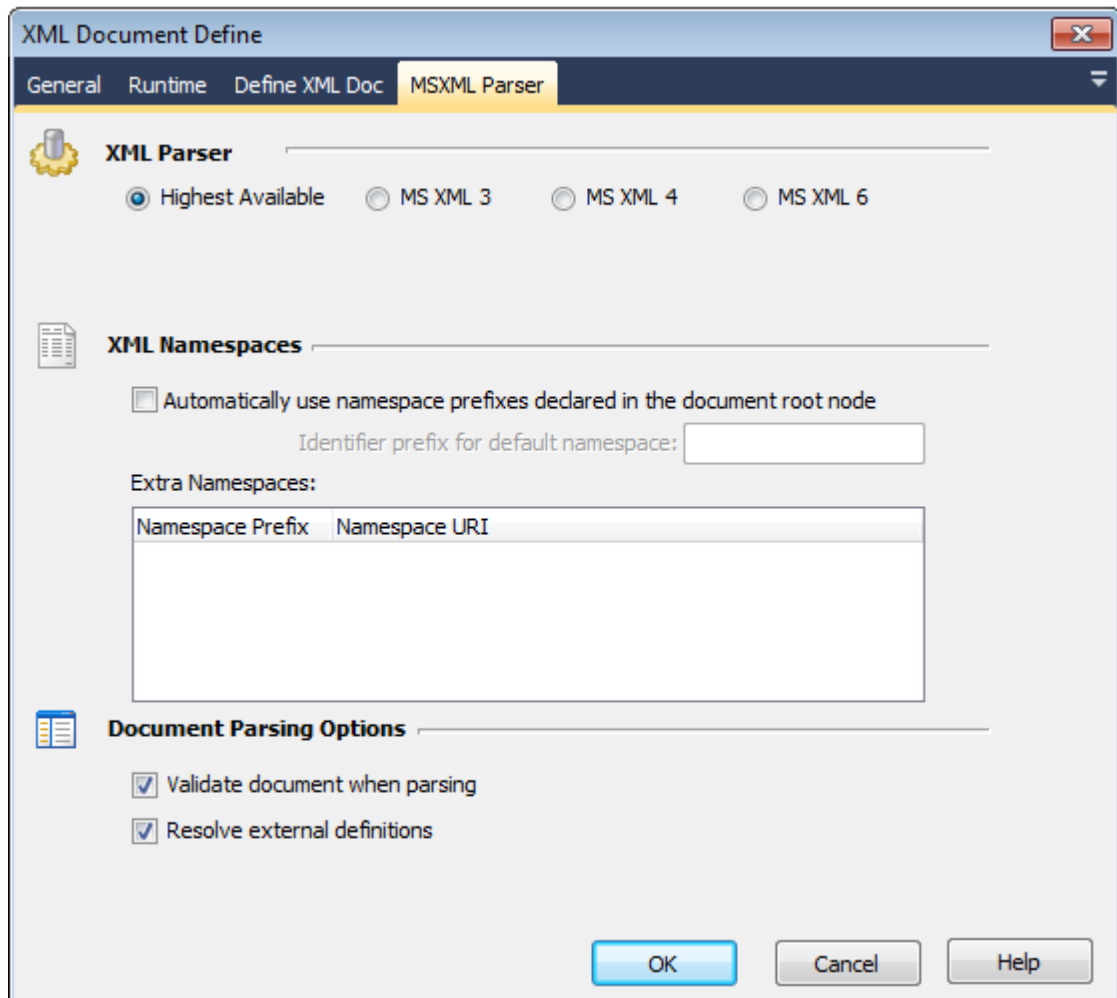
For each iteration, the variable will be set to the absolute XPath of the selected node. The XPath variable can then be used to provide XPaths to the other XML Actions.

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.

6.37.13 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 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 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 XPath's 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.

6.37.14 XML Save Document

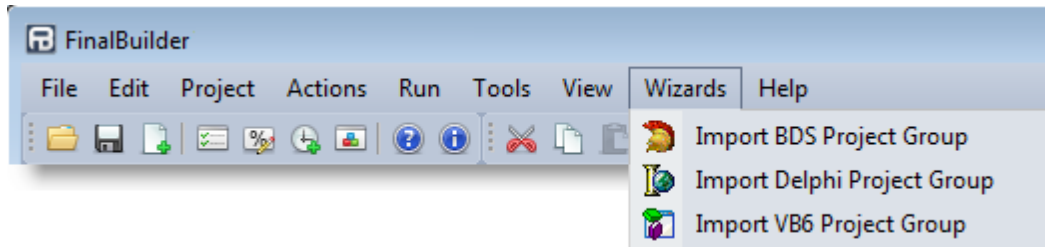
Save an XML Document Object created with the XML Document Define action to disk.

Simply select the XML Document Object and a filename to save the xml file to.

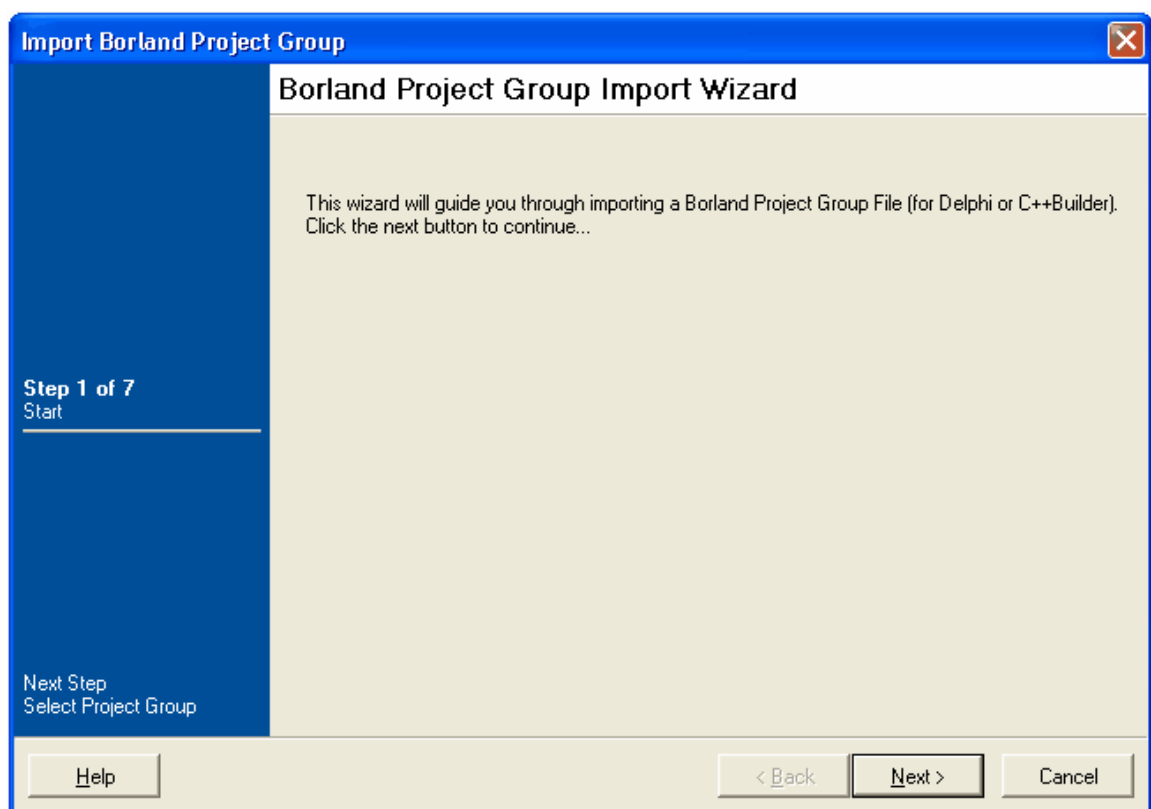
7 Wizards

7.1 Import Embarcadero Project Wizard

The Import Embarcadero Project Group Wizard imports Delphi project Group (.bpg) files.



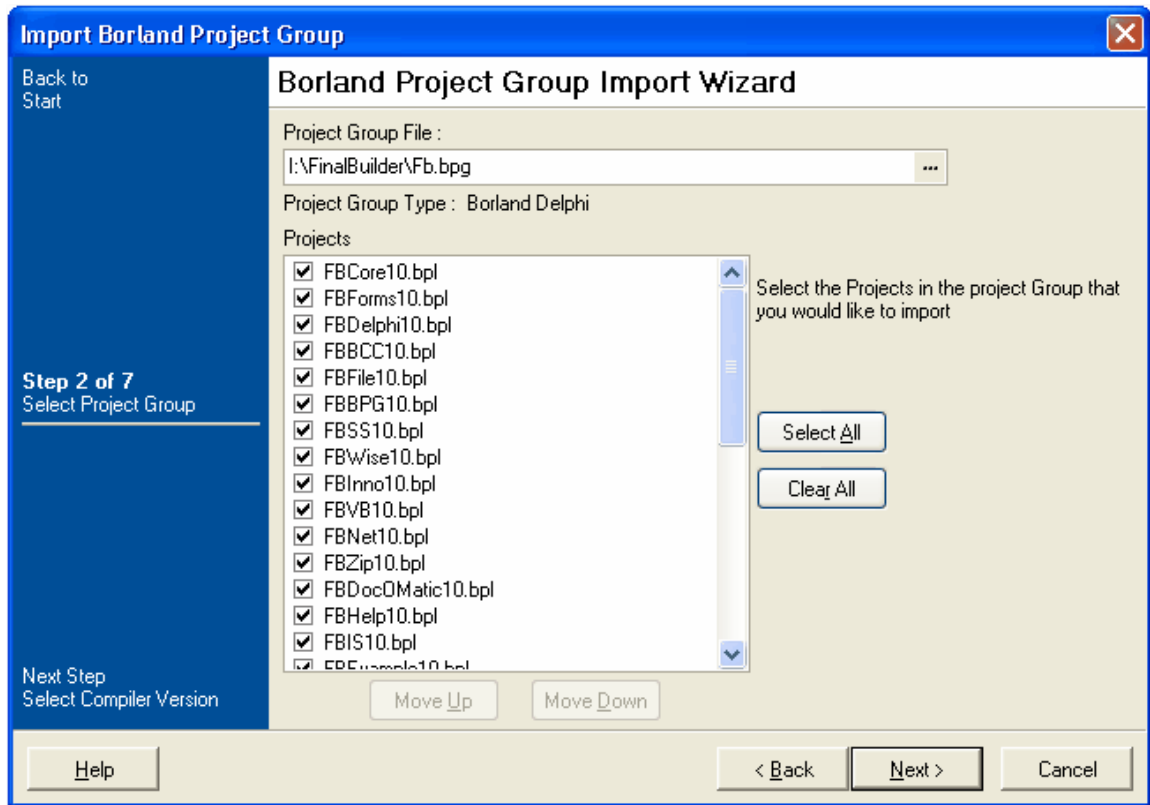
Note: To use the wizard, the *Delphi Project Group Import Wizard Package* must be selected in the PackageManager. FinalBuilder must then be restarted.



Next - Select Project Group Page

7.1.1 Select Project Group Page

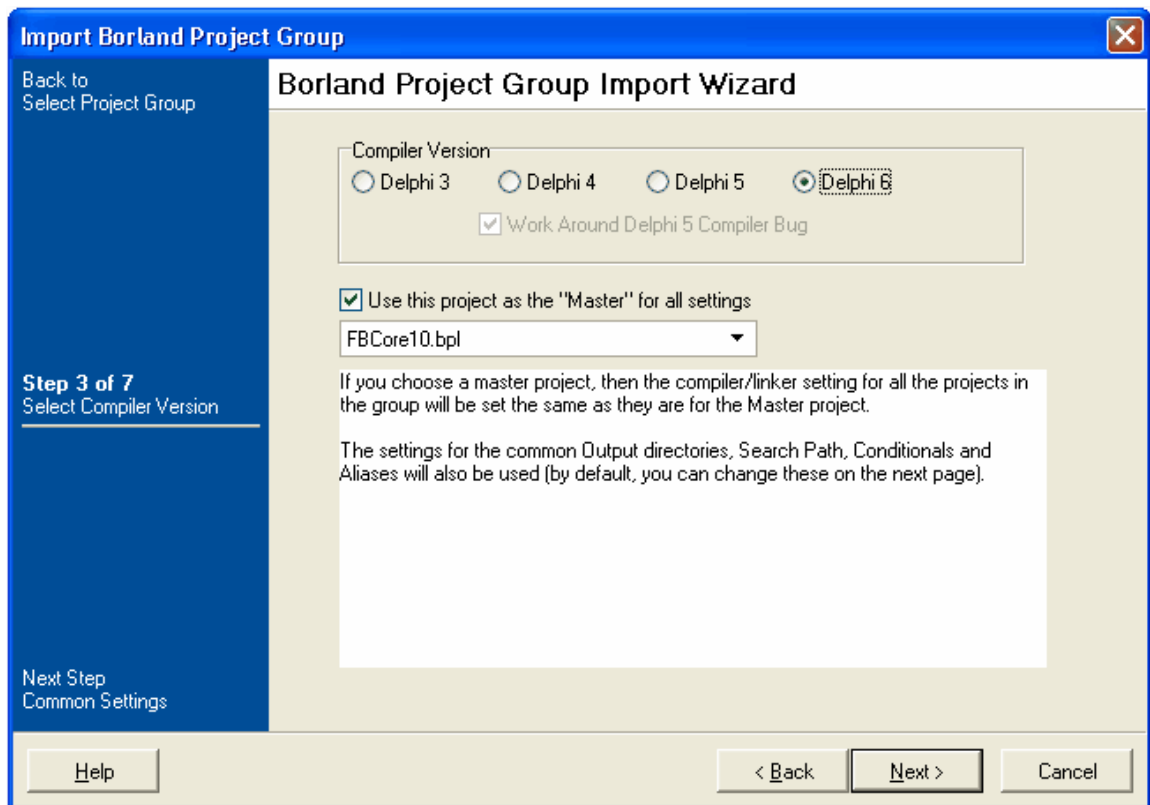
This page allows you to select the Project Group File (.bpg) and select the projects with the project group that you wish to import.



Next - Select Compiler Version Page

7.1.2 Select Compiler Version Page

There is no way for FinalBuilder to determine the correct version of Delphi to use for the project, so you must select the version of Delphi you wish to use. You can also select a project to act as the "Master" project, such that the settings from the master project will be applied to all projects. Some of the common settings can be overridden in the next page.



Next - Common Settings Page

7.1.3 Common Settings Page

This page allows you to set common settings for Output directories, Library Path, Search Path, Conditionals and Aliases. If you selected a Master project on the previous page, then the default settings will be those of the master project.

Import Borland Project Group

Back to
Select Compiler Version

Borland Project Group Import Wizard

☒ Apply Common Output Directories ☐ Apply Common Library Path
☒ Apply Common Search Path

Directories

Output directory : I:\FinalBuilder\Build ...

Unit output directory : .\ ...

Search path : inalBuilder\CorePackage;I:\FinalBuilder\FormsPackage ...

Library Path : \$(DELPHI)\Lib;\$(DELPHI)\Bin;\$(DELPHI)\Imports;... ..

BPL output directory : I:\FinalBuilder\Build ...

DCP output directory : I:\FinalBuilder\Build ...

☐ Apply Common Conditionals

Conditionals

Conditional defines :

☐ Apply Common Aliases

Aliases

Unit aliases : WinTypes=Windows;WinProcs=Windows;DbiTypes=BDE;

Help < Back Next > Cancel

Step 4 of 7
Common Settings

Next Step
Version Info

Next - Version Info Page

7.1.4 Version Info Page

This page makes it possible to apply common version info to all projects in the project group. It also makes it possible (by use of some scripting and FinalBuilder variables, to link the projects so that they share the same version numbering.

The screenshot shows the 'Borland Project Group Import Wizard' window, specifically 'Step 5 of 7: Version Info'. The window has a blue title bar and a sidebar on the left with navigation links: 'Back to Common Settings', 'Step 5 of 7: Version Info' (highlighted), and 'Next Step: Import Options'. The main area contains several sections with checkboxes and input fields:

- Include Version Info** (checked)
- Link Version Numbers for all projects** (checked)
 - FinalBuilder Project Variables**: Major (VERSION_MAJOR), Minor (VERSION_MINOR), Release (VERSION_RELEASE), Build (VERSION_BUILD)
 - Initial Values**: Major (1), Minor (2), Release (9), Build (676)
 - Auto Increment Build Number** (checked)
- Apply Common Locale** (checked)
 - Locale ID: \$0C09
 - English (Australia) (dropdown)
 - Code Page: 1252
- Apply Common Product Info** (checked)
 - Company Name: VSoft Technologies Pty Ltd
 - Internal Name: FinalBuilder
 - Legal Copyright: 2001 - 2002 VSoft Technologies Pty Ltd
 - Legal Trademarks: FinalBuilder
 - Product Name: FinalBuilder
 - Product Version: 1.0.0.0
 - Include Compiled Date in version info** (checked)

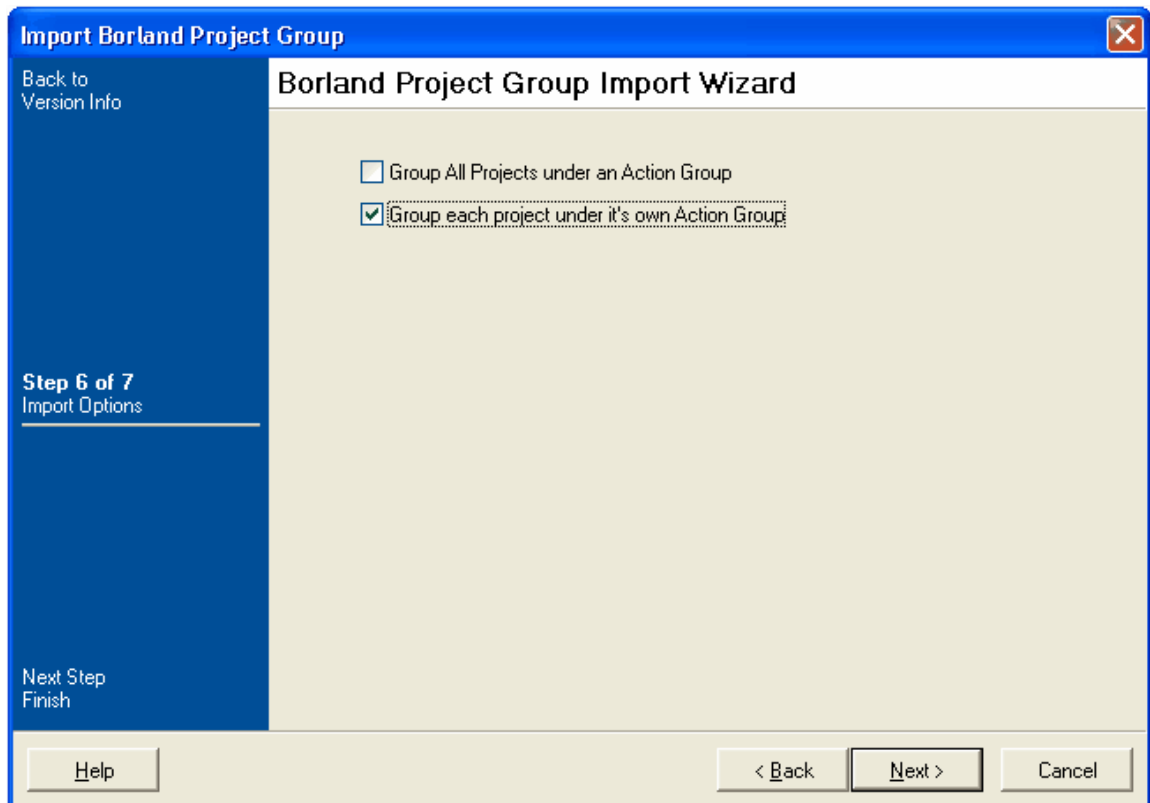
At the bottom are buttons for 'Help', '< Back', 'Next >', and 'Cancel'.

Next - Import Options Page

7.1.5 Import Options Page

Group All Projects under an Action Group - This option will make all projects children of an action group. This is useful when the project group is being imported into a FinalBuilder project that already has other Delphi compiler actions.

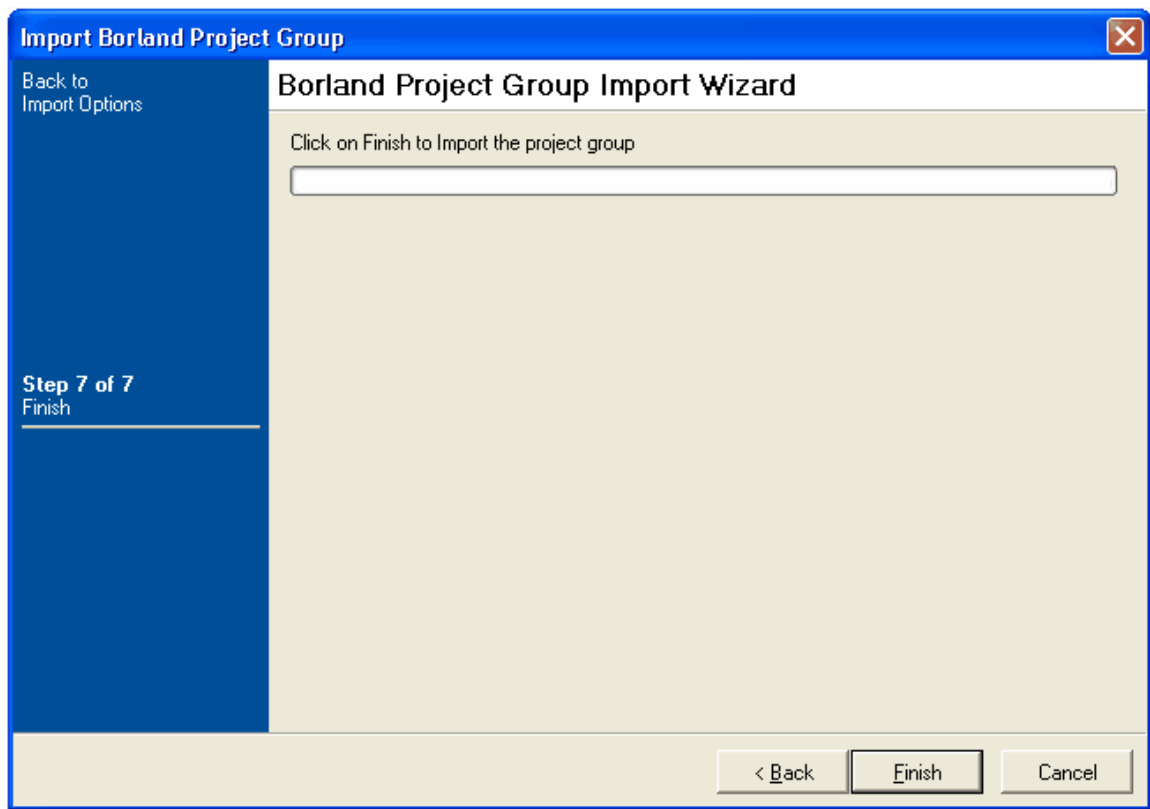
Group each Project under it's own Action Group - This option will group each action under an Action Group. This is advisable if you intend to add further action which are specific to each Delphi project.



Next - Finish Page

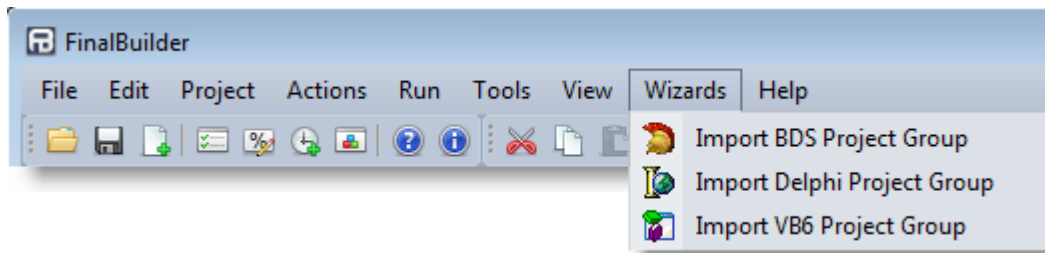
7.1.6 Finish Page

Click on the Finish button to import the project group.

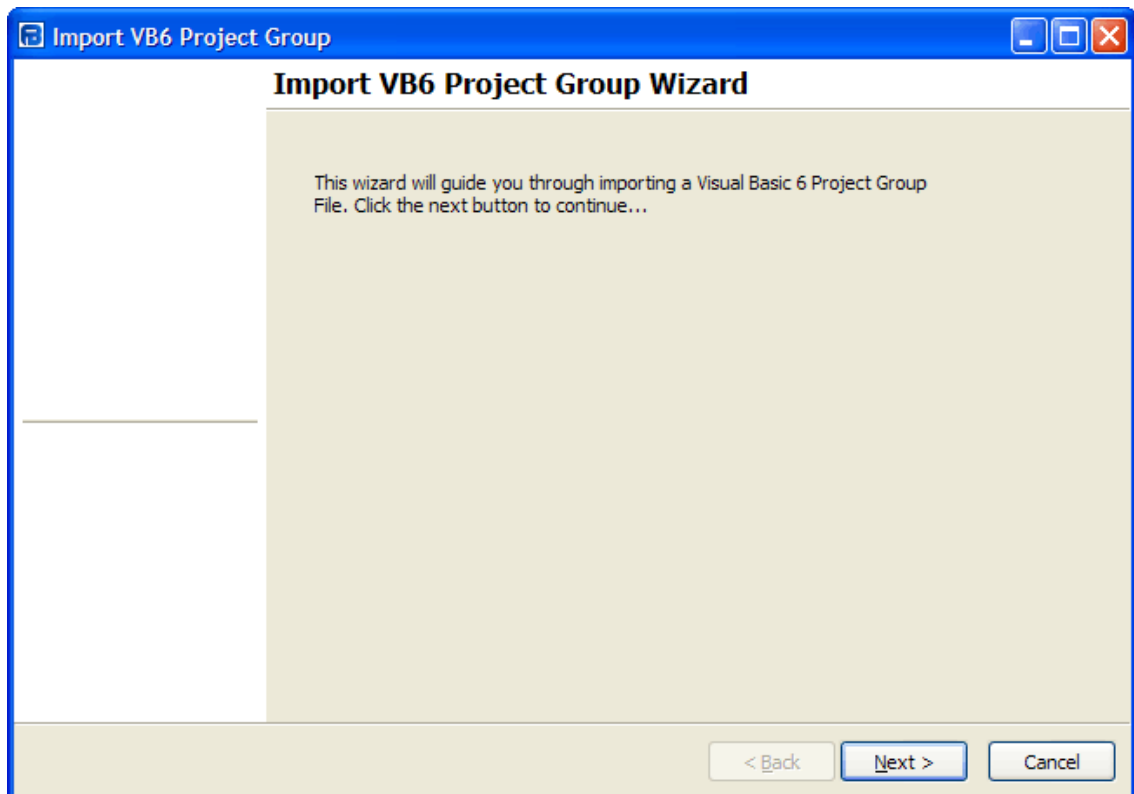


7.2 Import VB6 Project Group Wizard

This wizard will guide you through importing a VB6 project group.



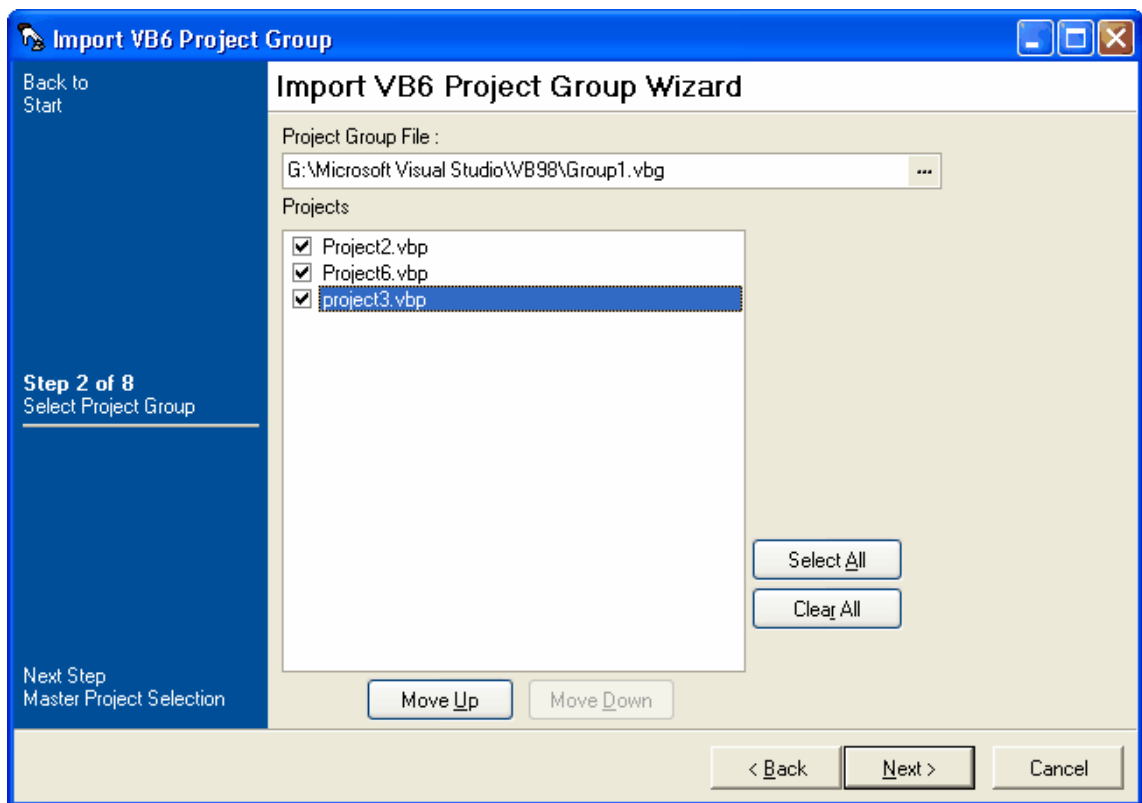
Note: To use the wizard, the *Visual Basic 6 Wizard Package* must be selected in the PackageManager. FinalBuilder must then be restarted.



Next: Select Project Group.

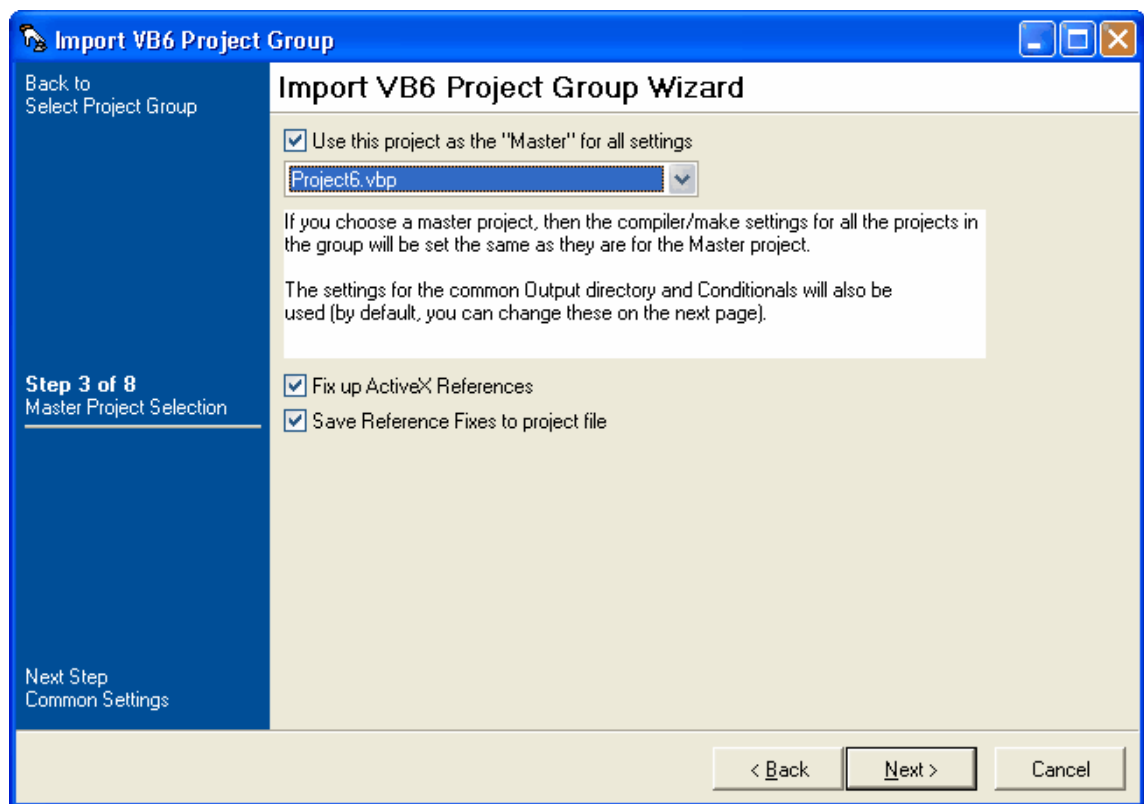
7.2.1 Select Project Group

This page allows you to select the Project Group File (.vbg) and select the projects with the project group that you wish to import.



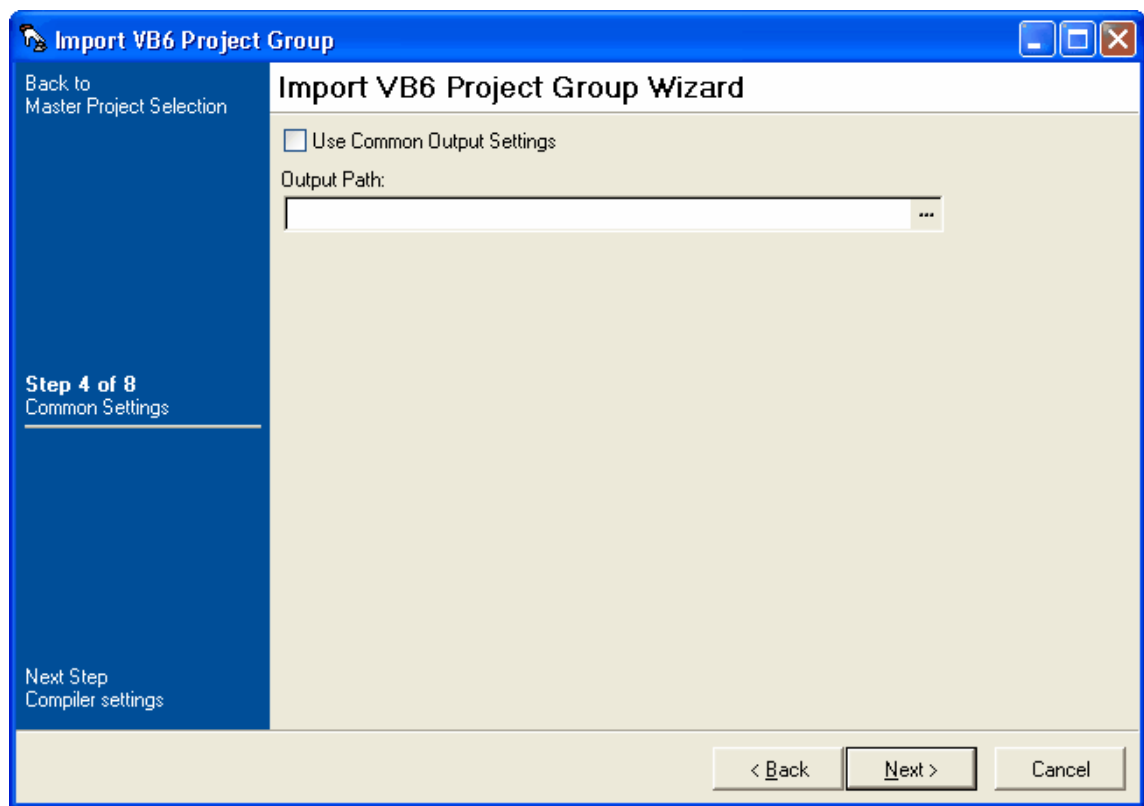
Next: Master Project Selection

7.2.2 Master Project Selection



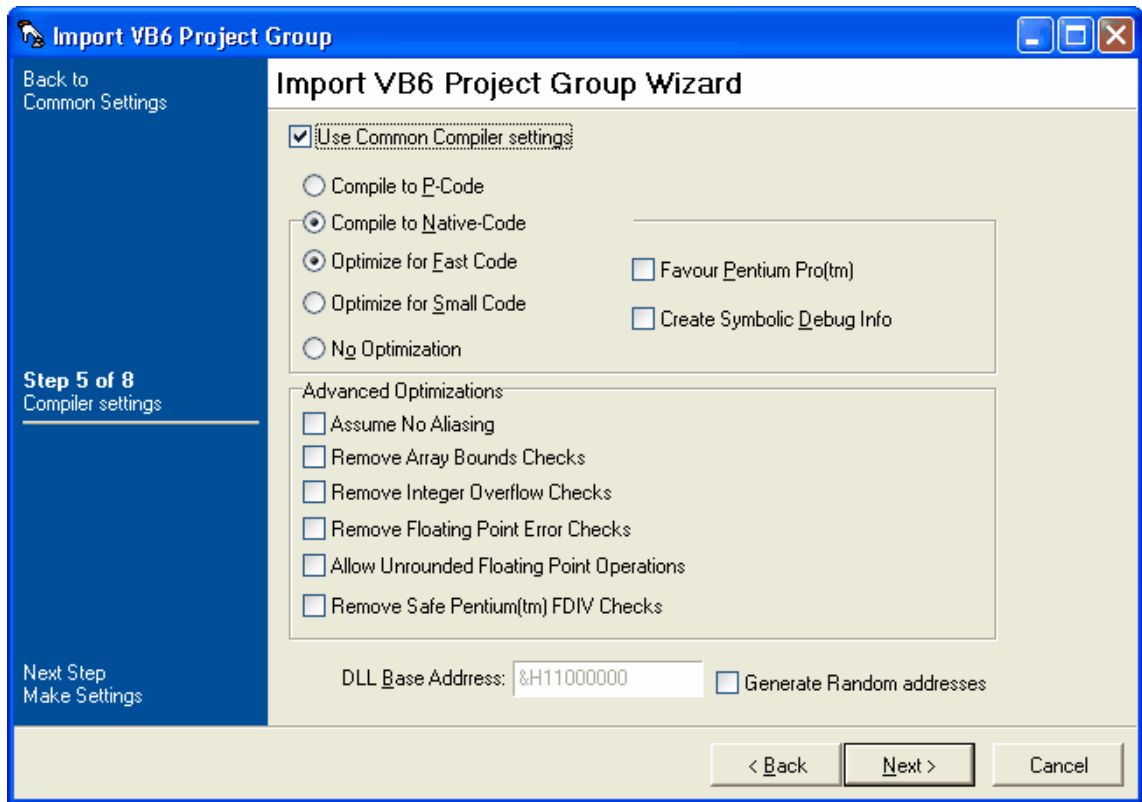
Next: Common Output Path Setting

7.2.3 Common Output Path Setting



Next: Compiler Settings

7.2.4 Compiler Settings



Next: Make Settings

7.2.5 Make Settings

The screenshot shows the 'Import VB6 Project Group Wizard' dialog box, specifically Step 6 of 8: Make Settings. The dialog has a blue title bar and a light beige main area. On the left, a blue sidebar contains navigation links: 'Back to Compiler settings', 'Step 6 of 8 Make Settings' (highlighted), and 'Next Step Grouping'. The main area is titled 'Import VB6 Project Group Wizard' and contains several sections:

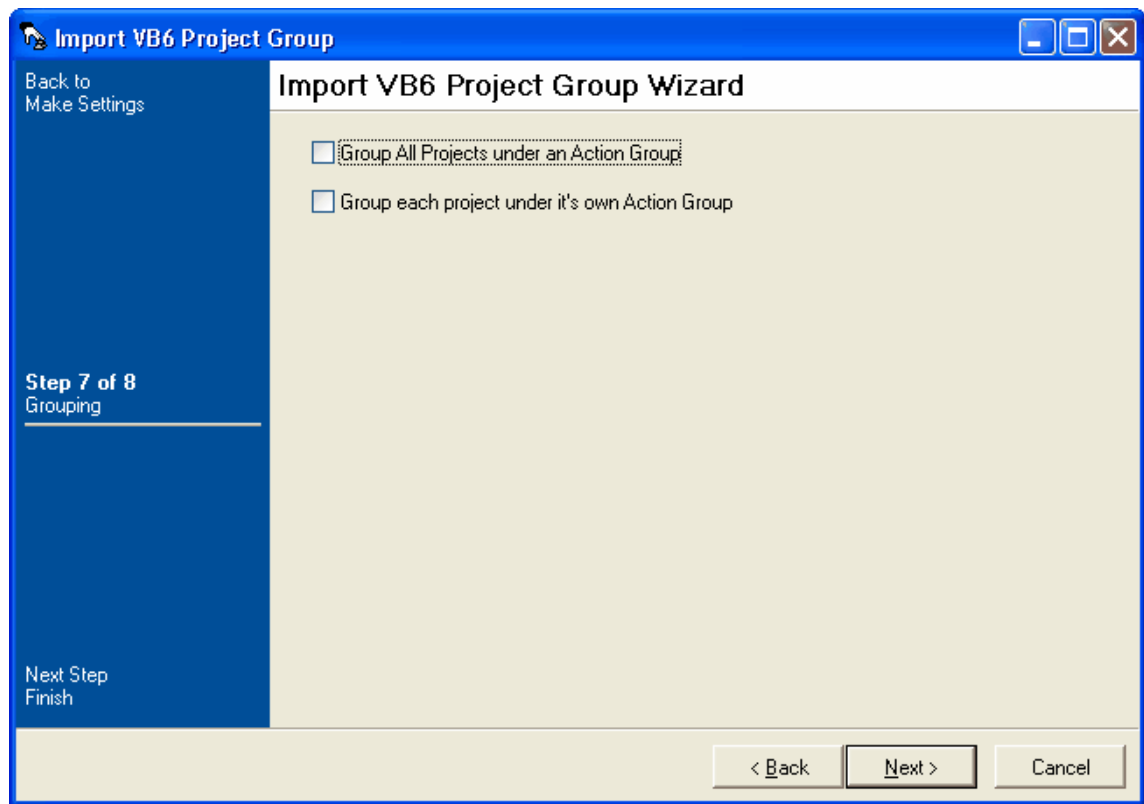
- Version Info:** A checkbox labeled 'Use Common Make settings' is checked. Below it is a table with two columns: 'Key' and 'Value'.

Key	Value
Comments	
Company Name	VSoft Technologies Pty Ltd
File Description	
Legal Copyright	
Legal TradeMarks	
Product Name	
- Include Compile Date:** A checkbox labeled 'Include Compile Date in Comment Version Info Field' is checked.
- Link Version Numbers:** A checkbox labeled 'Link Version Numbers for all projects' is checked. Below it, a section titled 'FinalBuilder Project Variables' contains three rows: 'Major' with 'VERSION_MAJOR' and a spinner set to 1, 'Minor' with 'VERSION_MINOR' and a spinner set to 0, and 'Revision' with 'VERSION_REVISION' and a spinner set to 0. Below these are two unchecked checkboxes: 'Auto Increment' and 'Use VBP Version Numbers'.
- Conditional Compilation Arguments:** A text box for entering arguments.

At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

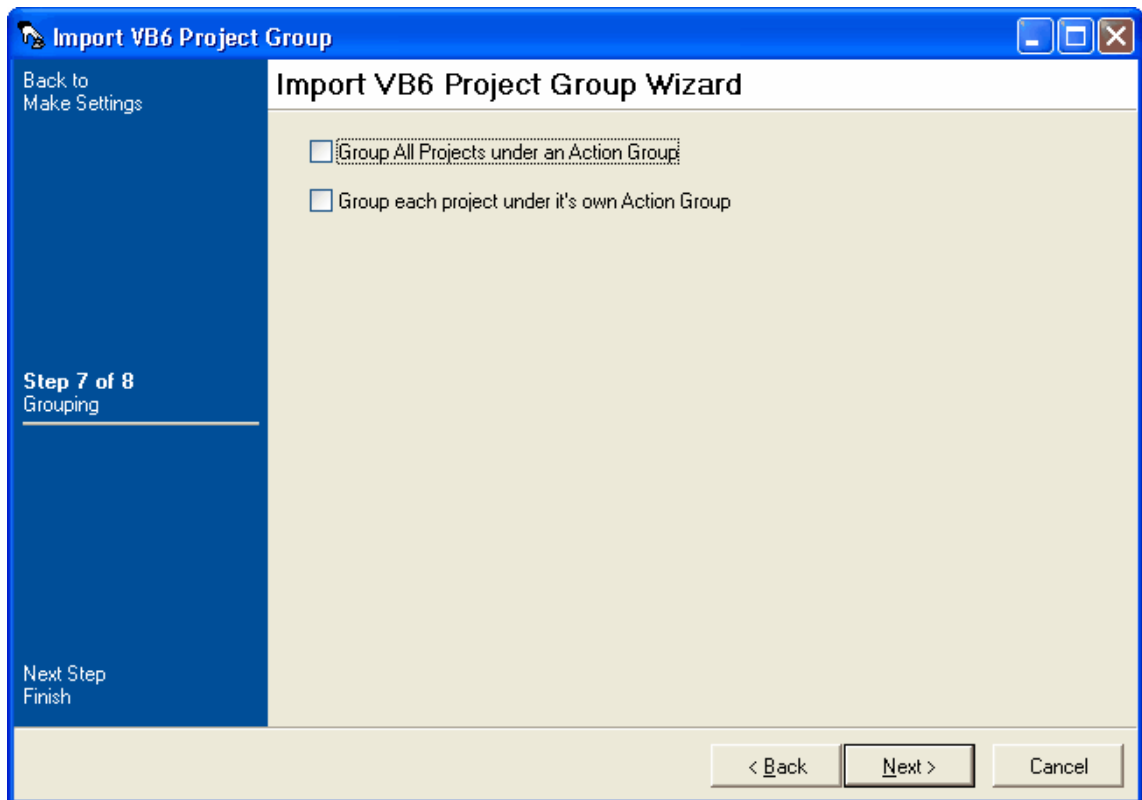
Next: Grouping

7.2.6 Grouping



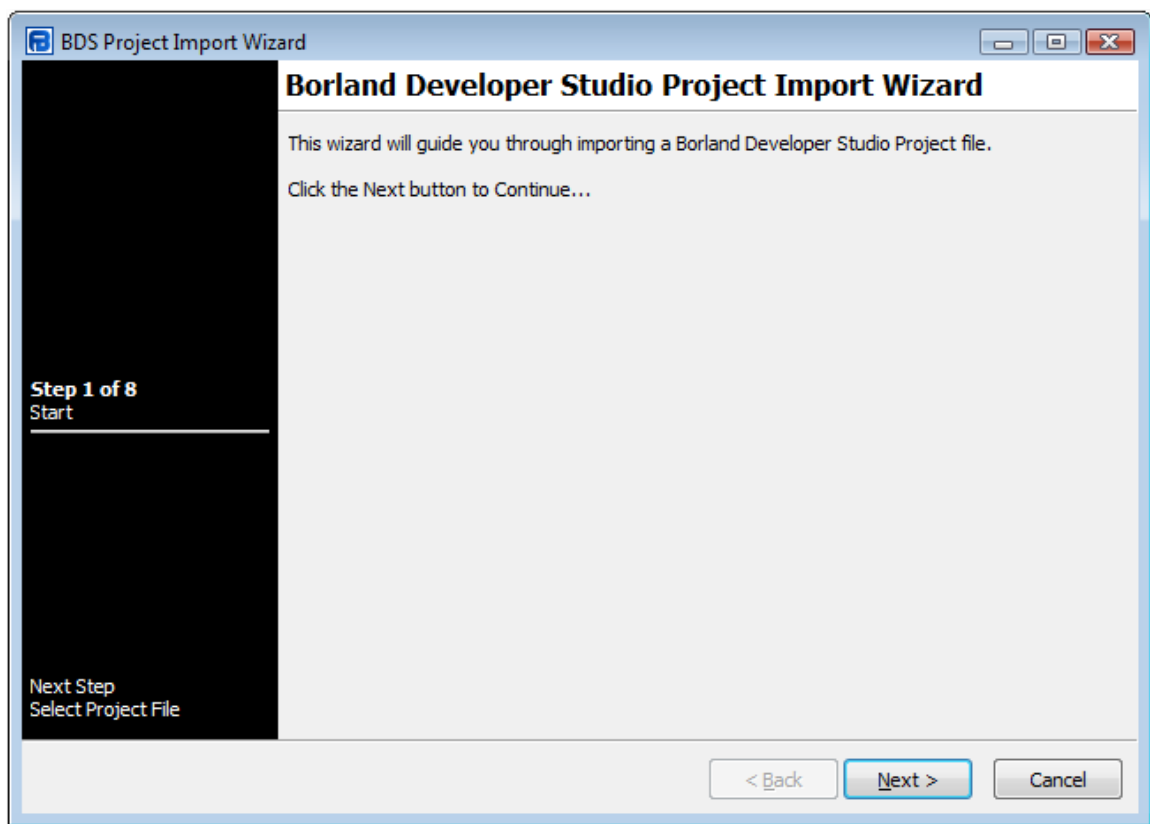
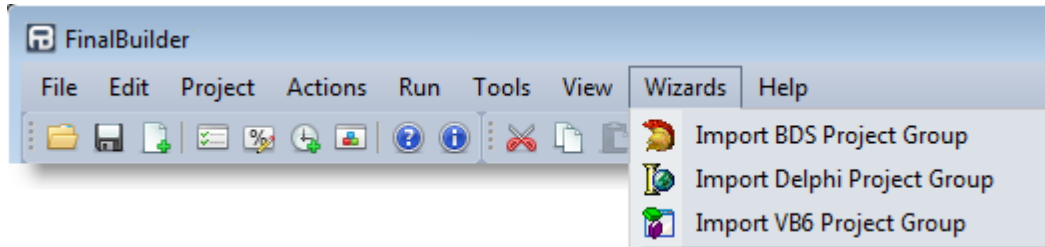
Next: Finish

7.2.7 Finish



7.3 Embarcadero Developer Studio Project Import Wizard

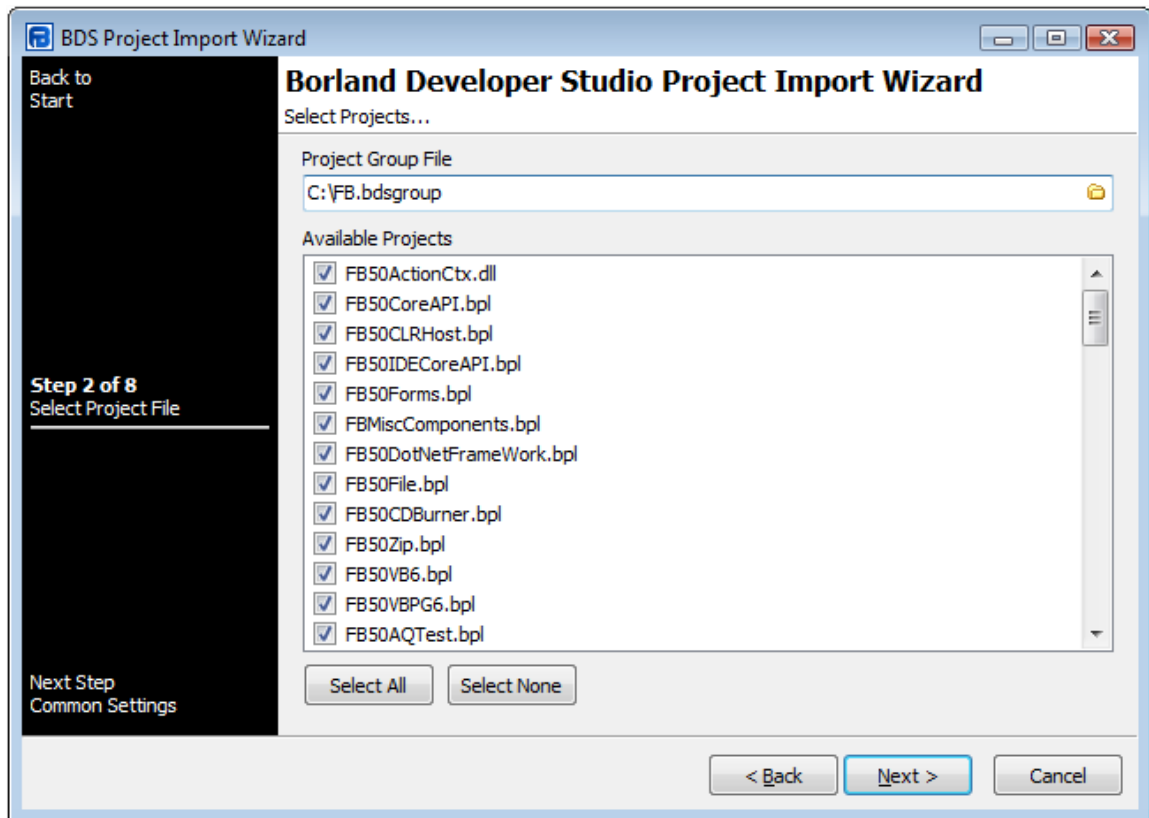
The Import BDS Project Group Wizard imports Embarcadero Developer Studio project (.bds) files.



Next: Select Projects

7.3.1 Select Projects

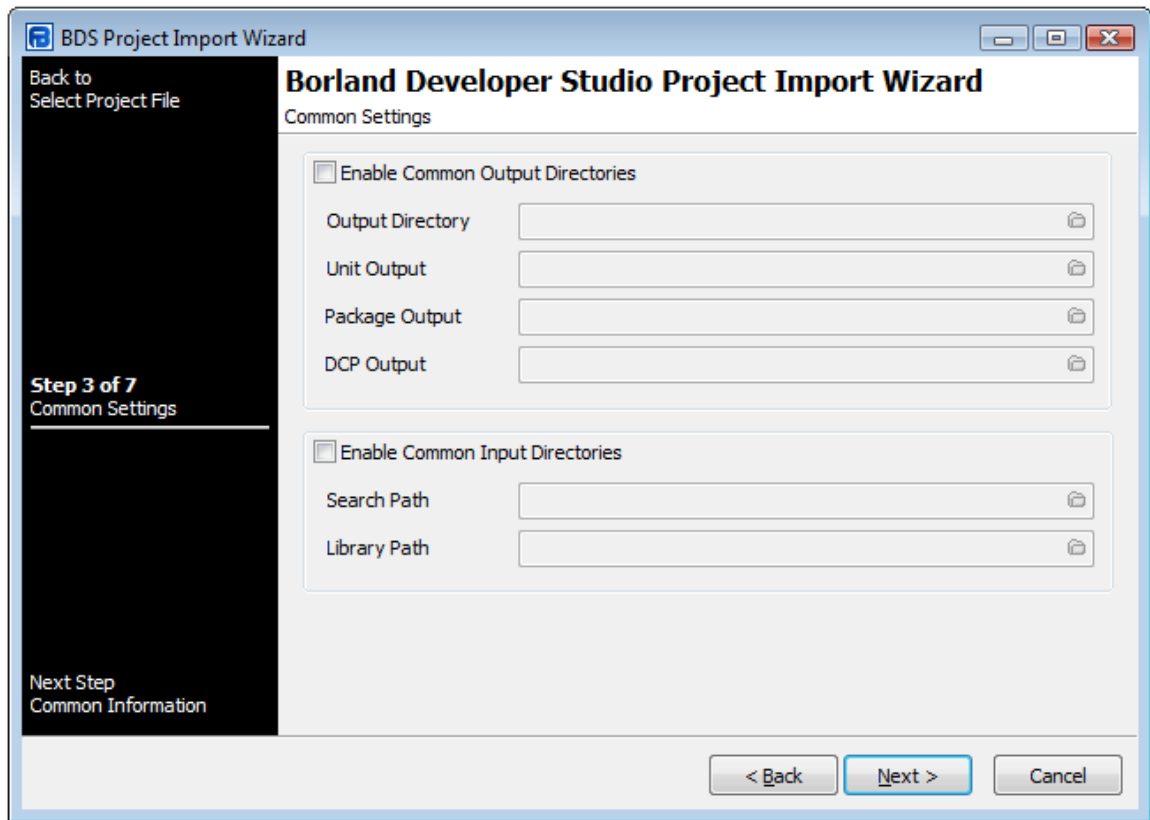
Select the project group file that contains the projects you wish to import.



Next: Common Settings

7.3.2 Common Settings

Set common output and input directories you would like the projects to share.



Next: Common Information

7.3.3 Common Information

Set the product, version and language settings you would like the projects to share.

The screenshot shows the 'Borland Developer Studio Project Import Wizard' window, specifically 'Step 4 of 7: Common Information'. The window has a title bar with the text 'BDS Project Import Wizard'. On the left side, there is a dark vertical panel with the following text: 'Back to Common Settings' at the top, 'Step 4 of 7 Common Information' in the middle, and 'Next Step Win32 Property Set' at the bottom. The main area of the wizard is titled 'Common Information' and contains three sections, each with a checkbox and several input fields:

- Enable Common Product Information:** This section includes input fields for 'Company Name', 'Internal Name', 'Legal Copyright', 'Legal Trademarks', 'Product Name', and 'Product Version'.
- Enable Common Version:** This section includes input fields for 'Major', 'Minor', 'Rel.', and 'Build'.
- Enable Common Language Settings:** This section includes a 'Language' dropdown menu, an 'ID:' label, and a 'Code Page' input field with the value '1252'.

At the bottom right of the wizard, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Next: Win32 Property Set

7.3.4 Win32 Property Set

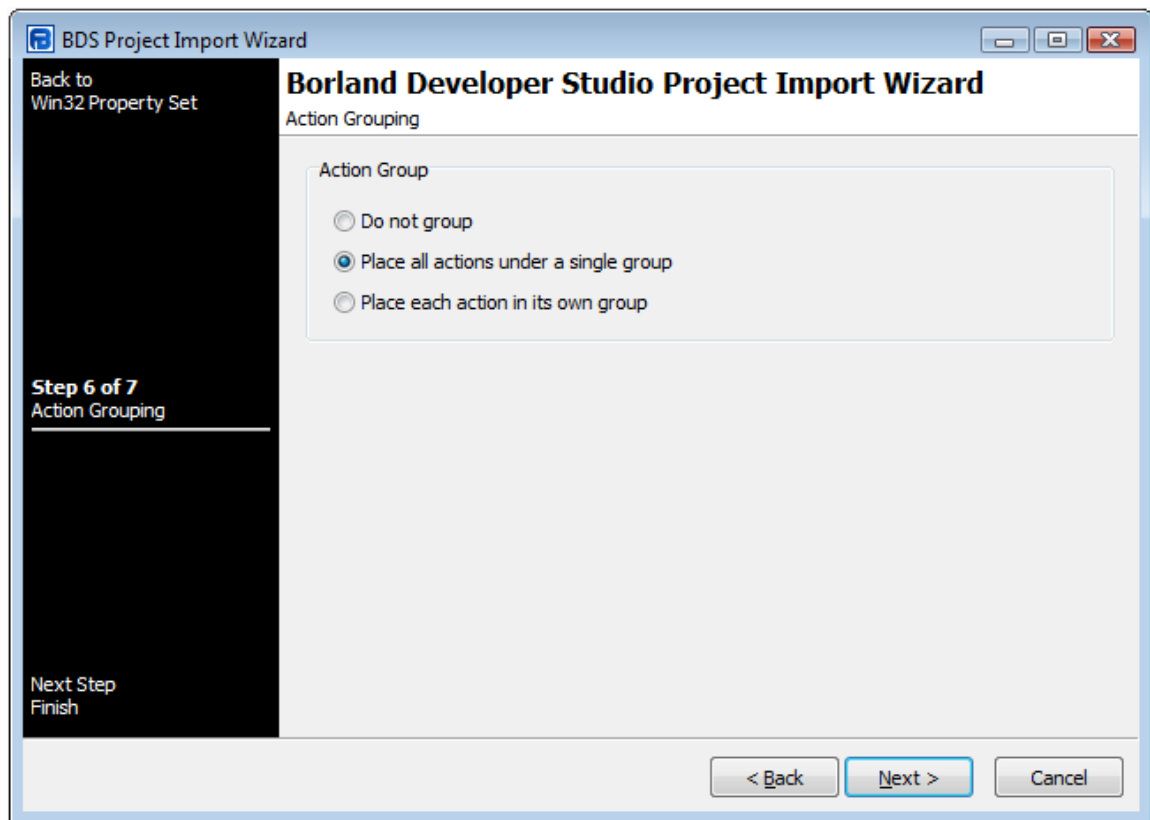
You can optionally choose to create a Win32 Property Set from the projects.

The screenshot shows the 'Borland Developer Studio Project Import Wizard' window, specifically 'Step 5 of 7: Win32 Property Set'. The window has a title bar with 'BDS Project Import Wizard' and standard Windows window controls. On the left, a dark sidebar contains the text 'Back to Common Information', 'Step 5 of 7 Win32 Property Set', and 'Next Step Action Grouping'. The main area is titled 'Win32 Property Set' and contains a checkbox labeled 'Create a Win32 Info Property Set'. Below this checkbox are several input fields: 'Property Set Name', 'Title', 'Description', 'Company', 'Product', 'Copyright', 'Trademark', and a 'Version' section with sub-fields for 'Major', 'Minor', 'Rel.', and 'Build'. At the bottom right, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Next: Action Grouping

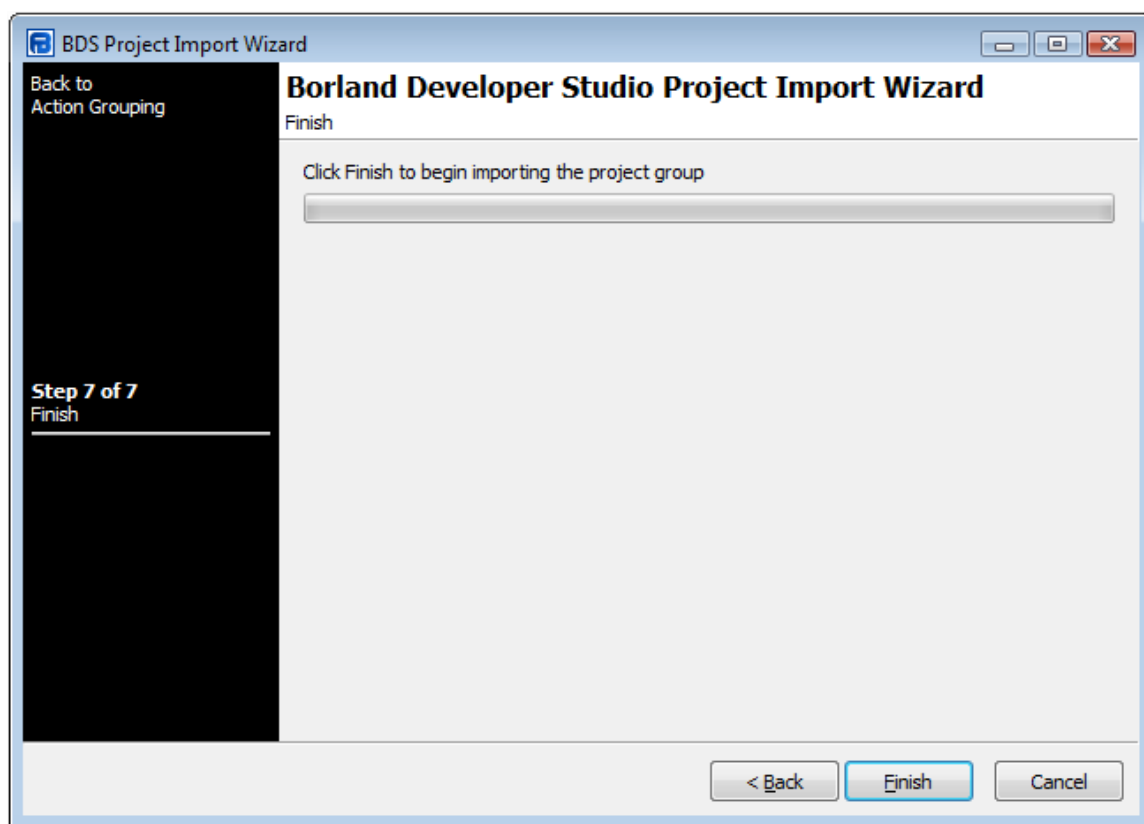
7.3.5 Action Grouping

Define how you would like the actions to be grouped.



Next: Finish

7.3.6 Finish



8 Automating FinalBuilder

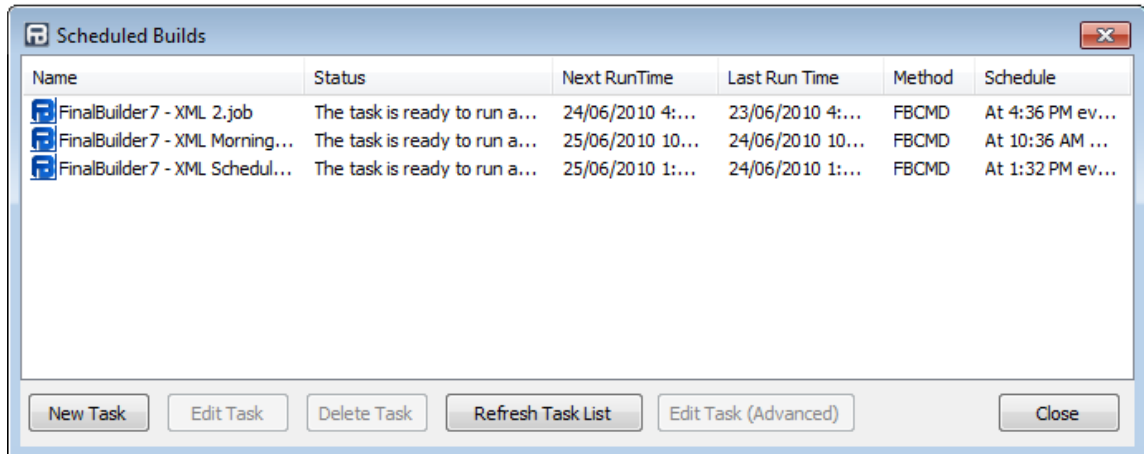
There are a number of ways to automate FinalBuilder:

- Executing FinalBuilder (the IDE) from the command line
- Using the built in scheduler
- Executing FBCMD on the command line
- Using FinalBuilder Server for centralized build management and continuous integration

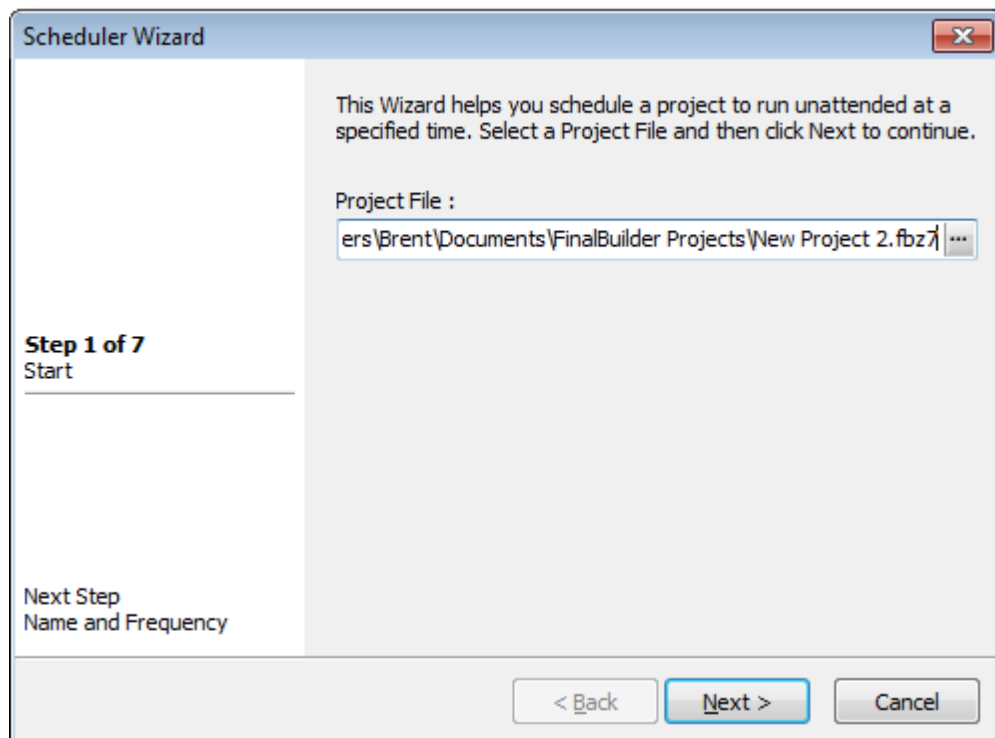
8.1 FinalBuilder IDE

8.1.1 Scheduling builds

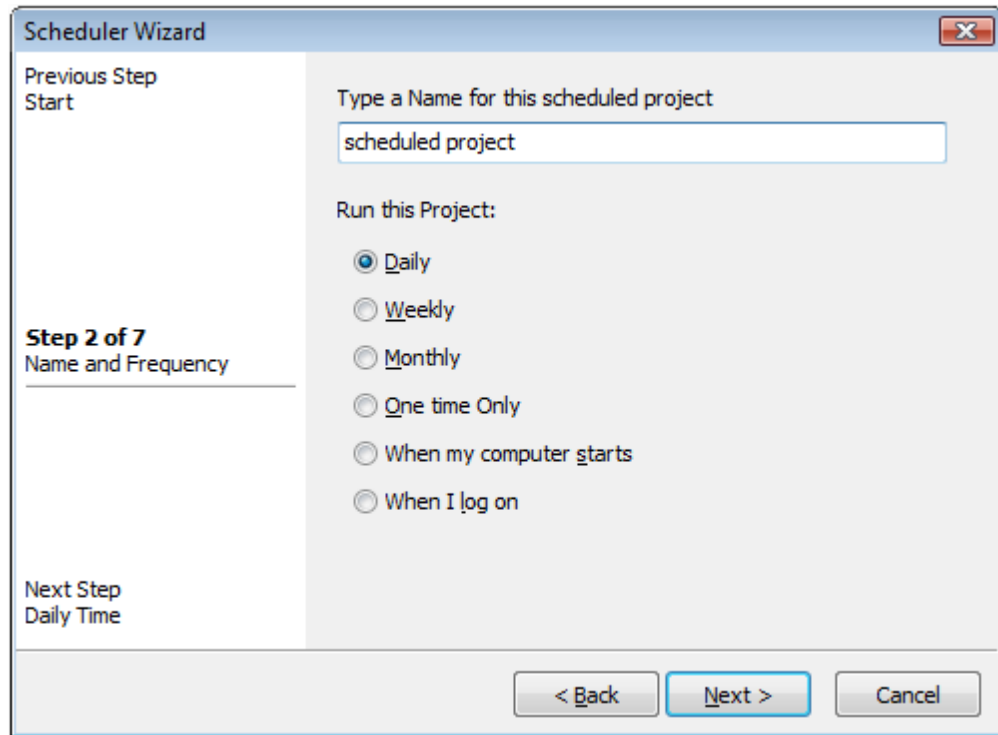
FinalBuilder can schedule builds to be run by the Windows Scheduling service. To access the scheduling functions, choose Scheduled builds from the Tools menu.



To Create a new scheduled build, click on the New Task button. This will display the wizard that will step you through scheduling a build.



Select a FinalBuilder Project File and then click Next to continue.

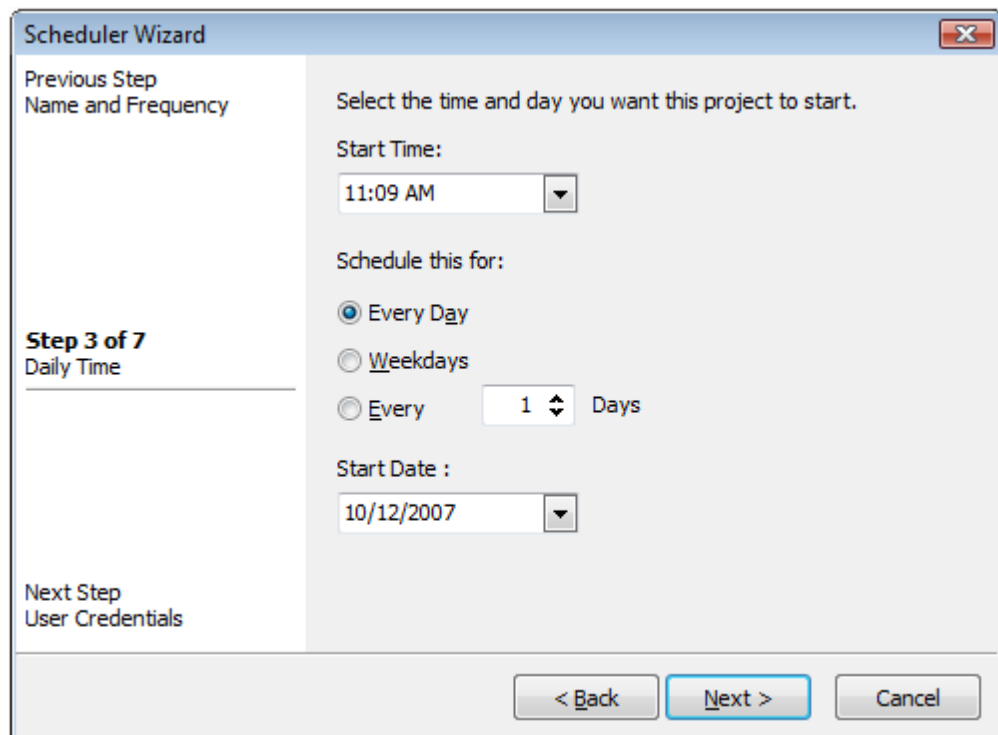


The Scheduler Wizard window is titled "Scheduler Wizard". On the left, a sidebar shows the progress: "Previous Step Start", "Step 2 of 7 Name and Frequency" (highlighted), and "Next Step Daily Time". The main area contains the following fields and options:

- "Type a Name for this scheduled project": A text box containing "scheduled project".
- "Run this Project:": A group of radio buttons with the following options:
 - ☒ Daily
 - ☐ Weekly
 - ☐ Monthly
 - ☐ One time Only
 - ☐ When my computer starts
 - ☐ When I log on

At the bottom right are three buttons: "< Back", "Next >" (highlighted), and "Cancel".

Enter a Title for the scheduled build and then specify the frequency of the build, then click on next to continue.

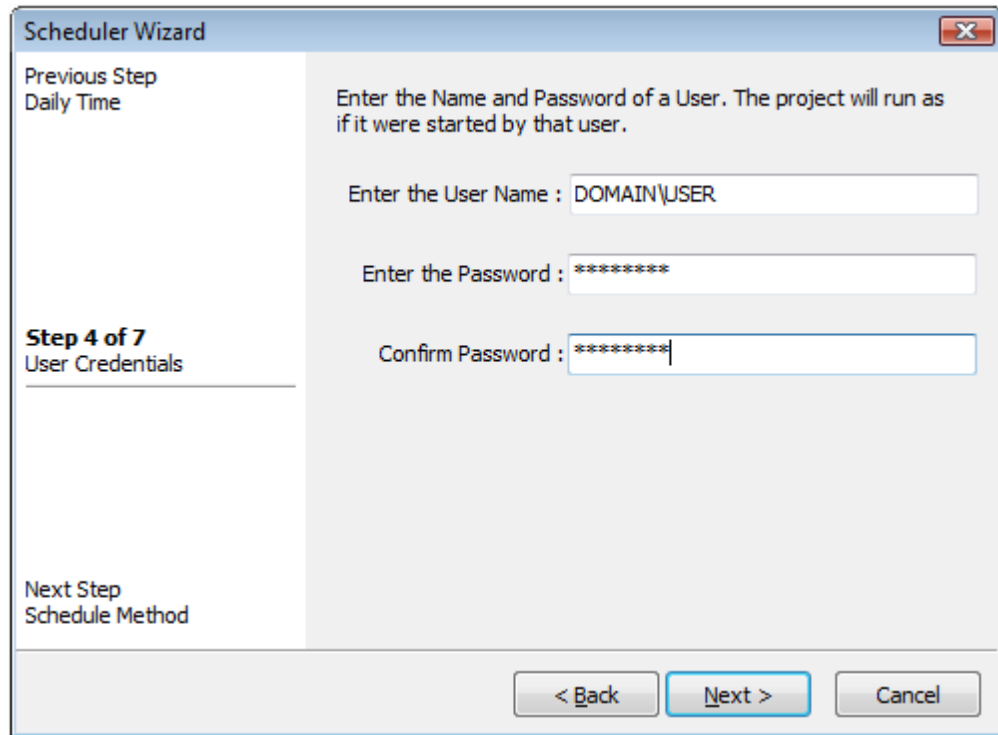


The Scheduler Wizard window is titled "Scheduler Wizard". On the left, a sidebar shows the progress: "Previous Step Name and Frequency", "Step 3 of 7 Daily Time" (highlighted), and "Next Step User Credentials". The main area contains the following fields and options:

- "Select the time and day you want this project to start.":
- "Start Time:": A dropdown menu showing "11:09 AM".
- "Schedule this for:": A group of radio buttons with the following options:
 - ☒ Every Day
 - ☐ Weekdays
 - ☐ Every 1 Days (with a spinner box around the number 1)
- "Start Date :": A dropdown menu showing "10/12/2007".

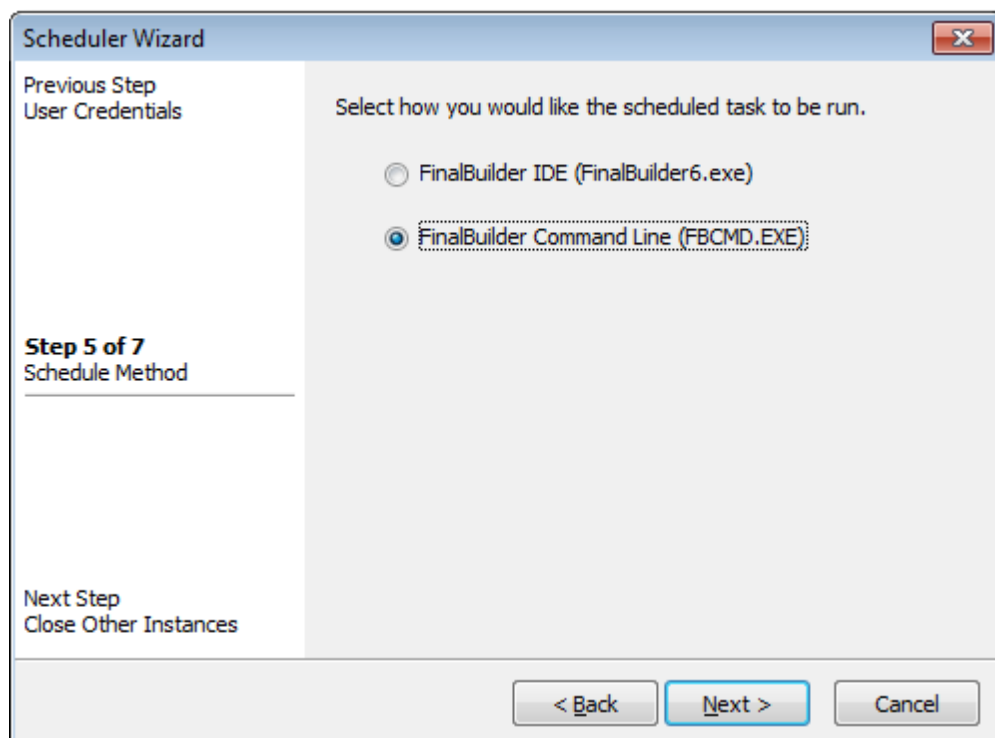
At the bottom right are three buttons: "< Back", "Next >" (highlighted), and "Cancel".

Specify the Time and starting date for the schedule, then click next to continue.



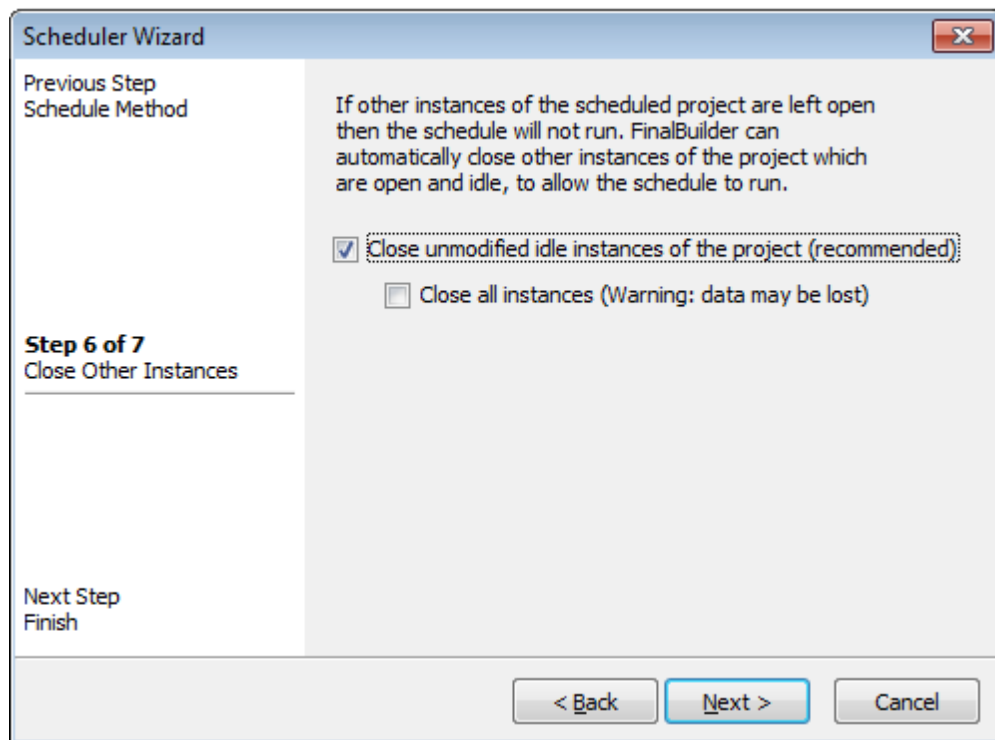
The image shows the 'Scheduler Wizard' dialog box at Step 4 of 7, titled 'User Credentials'. The left sidebar shows 'Previous Step: Daily Time' and 'Next Step: Schedule Method'. The main area contains the instruction: 'Enter the Name and Password of a User. The project will run as if it were started by that user.' There are three input fields: 'Enter the User Name:' with the text 'DOMAIN\USER', 'Enter the Password:' with '*****', and 'Confirm Password:' with '*****'. At the bottom are buttons for '< Back', 'Next >', and 'Cancel'.

Provide the user name that the build will run as and the password, then click continue.



The image shows the 'Scheduler Wizard' dialog box at Step 5 of 7, titled 'Schedule Method'. The left sidebar shows 'Previous Step: User Credentials' and 'Next Step: Close Other Instances'. The main area contains the instruction: 'Select how you would like the scheduled task to be run.' There are two radio button options: 'FinalBuilder IDE (FinalBuilder6.exe)' and 'FinalBuilder Command Line (FBCMD.EXE)'. The second option is selected. At the bottom are buttons for '< Back', 'Next >', and 'Cancel'.

You can choose to run a scheduled build in the IDE or from the command line tool (FBCMD). FBCMD is recommended for builds that will run unattended.

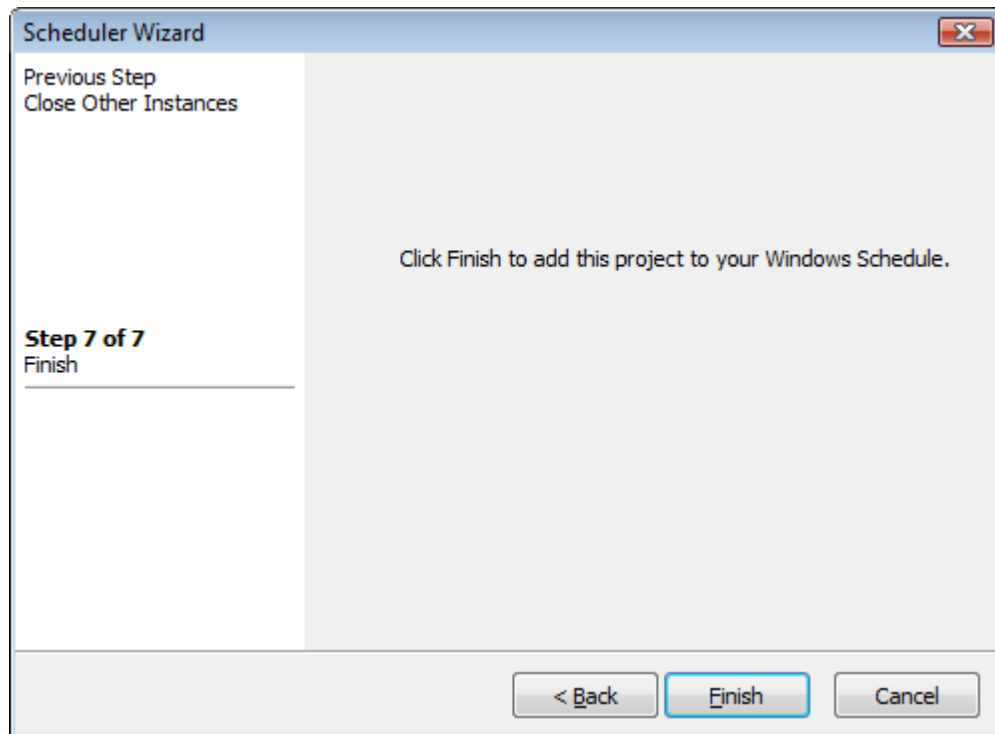


FinalBuilder can optionally close other instances of itself which are using the scheduled project. This is useful because only one FinalBuilder instance can have a single project open at a time.

The **Close unmodified idle instances of the project** option will close any other FinalBuilder 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 FinalBuilder 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 builds 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 FBCMD reference topics..



Click Finish to add this build to your Scheduled Tasks.

8.1.2 IDE Command Line Interface

(To run builds from the command line in a console window, use the Command line executable.)

FinalBuilder Command Line Options

Usage : FinalBuilder.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 /v	Set FinalBuilder 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.	/vOutputDIR="d:\Output\My Output"; DCUDIR=d:\temp\dcu
-a or /a	Allow interactive actions (prompt for variables, message boxes, etc.) in an automatic build.*	/r /a
-o or /o	Close any other instances of the FinalBuilder IDE which are idle with unmodified copies of the same project file.*	/r /o
-of or /of	Forcibly close any other instances of the FinalBuilder 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	-tl

	be open multiple times concurrently. Log history will be lost when the FinalBuilder IDE closes.	
-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 FinalBuilder as a scheduled task, you should always use the /r and /e switches.

See Also

FBCMD Command Line Tool | Scheduling builds

8.1.2.1 Exit Codes

The FinalBuilder IDE uses the following Windows exit codes to indicate the outcome of the build :

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	FinalBuilder was force terminated following a remote request from another FinalBuilder instance (started using the -of option.)
999	The user elected to force terminate FinalBuilder after the running project failed to terminate.

See the Command Line Exit Codes topic to see the extended range of exit codes available from FBCMD.exe.

See Also

IDE Command Line Interface | Scheduling builds

8.2 Command Line version

FBCMD is a text mode version of FinalBuilder which can run projects from the command line. It is installed in the FinalBuilder directory.

FBCMD can be faster at executing than the IDE, as the IDE has more visual overhead.

Command Line Syntax

The command line syntax for FBCMD is as follows:

FBCMD.exe [options] [/P]<projectfile>

Options :

Switch	Behaviour	Example
/P	FinalBuilder 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.fbz7" or "C:\Build\MyProject.fbz7"
/I	Ignore all system message (eg. Ctrl-C, Shutdown.)	/I
/A	Allow interactive actions (message dialogs, prompt for variables, etc.)	/A
/V	Set FinalBuilder 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 build runs many actions very quickly. Console output is still enabled (see below.) With this option (or /TL) enabled, the same project can be opened multiple times concurrently.	/S
/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.	/TL

	With this option (or /S) enabled, the same project can be opened multiple times concurrently.	
@	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 FinalBuilder 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 builds | IDE Command Line Interface

8.2.1 Exit Codes

FBCMD 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 FinalBuilder version.

See Also

IDE Exit Codes

8.3 MSBuild Task

FinalBuilder comes with a built-in MSBuild task, allowing FinalBuilder projects to be execution as part of an MSBuild Project.

(If you want to use FinalBuilder as part of a Team Foundation Build Project, see the topic on FinalBuilder's graphical Team Build Integration features.)

To include a FinalBuilder project inside an MSBuild project, you will need to include exactly one `UsingTask` element for the `FinalBuildertask` (see below for the full syntax), and then one or more `FBCmd` tasks to execute FinalBuilder projects.

Here is a minimal example which can be copy-and-pasted into an MSBuild Project file. The FinalBuilder project will run for the "AfterBuild" target.

```
<UsingTask TaskName="FinalBuilder.FBCmd" AssemblyName="FinalBuilder.
MSBuild.Tasks, Version=7.0.0.0, Culture=neutral,
PublicKeyToken=300e5b79bda2b697" />

<Target Name="AfterBuild">
  <FBCmd ProjectFile="C:\My Build\My Build.fbp7" />
</Target>
```

Here is an example task showing all available options:

```
<FBCmd ProjectFile=". \Included Project.fbp7"
  Variables="VariableA='Hello World';VariableB=66"
  StopIfFBFails="true"
  AllowInteractive="true"
  DontWriteToLog="false"
  Timeout="30000"
  ShowBanner="true"
  FBVersion="7"
/>
```

Project File

Path to the FinalBuilder project to execute. Required.

Variables

Optional. Specify one or more variables to set in the target project. Use the syntax "Name=Value". Separate multiple variables with semicolons. Optional.

StopIfFBFails

Should the MSBuild Target fail if FB fails? Defaults to True.

AllowInteractive

Should interactive actions be allowed in the project? Defaults to False.

DontWriteToLog

Should output be not written to the FinalBuilder log? Defaults to False.

Timeout

Should the action time out after X milliseconds? Defaults to no timeout.

ShowBanner

Should the FinalBuilder banner be sent to the MSBuild log?

FBVersion

What version of FinalBuilder should be used to execute the project? By default, this is determined by the project file name.

Other Options

There are some other, undocumented, options which are used when working with Team Foundation Server. It is recommended that you use the FinalBuilder Visual Studio Addin to configure these options.

8.4 Team Foundation Server Integration

FinalBuilder can be used to integrate with and extend Microsoft Team Foundation Server. If you are using TFS 2008, please see the Team Build section. If you are using TFS 2010, please see the Windows Workflow section.

8.4.1 Team Build (TFS 2008)

Team Build is Microsoft's build framework, part of Microsoft Team Foundation Server. FinalBuilder can be used to extend the Team Build engine, in place of MSBuild.

8.4.1.1 Visual Studio Addin

The Team Build Addin allows you to configure Team Build for use with FinalBuilder. The Addin requires Visual Studio 2008 with Team Explorer installed.

The Addin can also be used when connected to Team Foundation 2005 servers. Some functionality is not available when connecting to TFS 2005 servers.

Installation

The Addin is automatically installed by the FinalBuilder installer, but only if Visual Studio 2008 is already installed. The Addin file is installed to the All Users\Application Data directory, under Microsoft\MSEnvShared\Addins.

To enable or disable the Addin, go to Tools -> Addin Manager in Visual Studio, and check or uncheck the box for "FinalBuilder Team Build Addin".

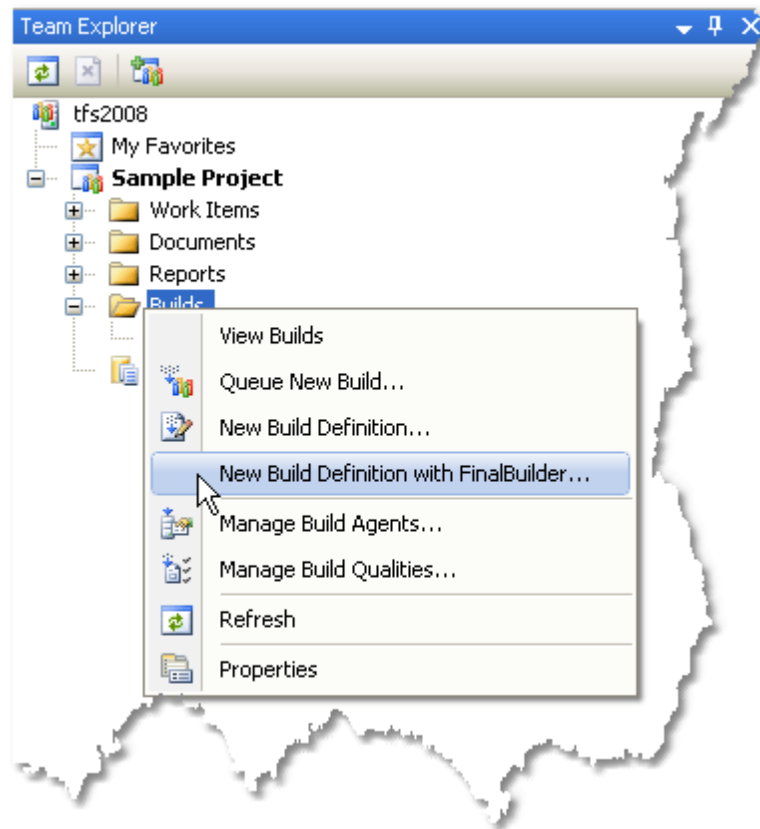
Commands

The Addin provides two new commands from Team Explorer - "New Build Definition With FinalBuilder" and "Edit FinalBuilder Project(s)"

8.4.1.1.1 New Build Definition With FinalBuilder

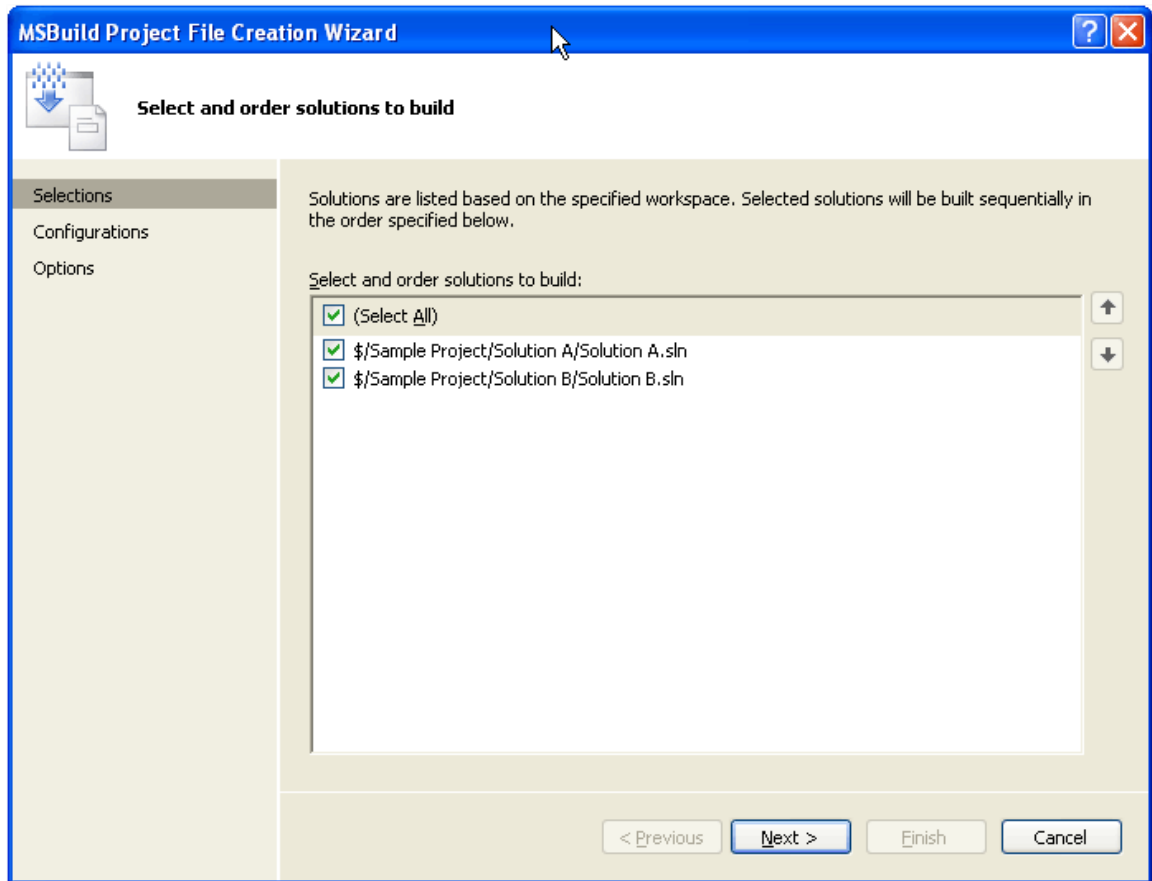
Use this command to add a new Build Definition to a Team Project, using FinalBuilder as the build engine instead of MSBuild. A new FinalBuilder project is automatically created, added to the build definition, and checked in to source control.

To use this command, right-click on the "Builds" node in Team Explorer and choose "New Build Definition with FinalBuilder...":

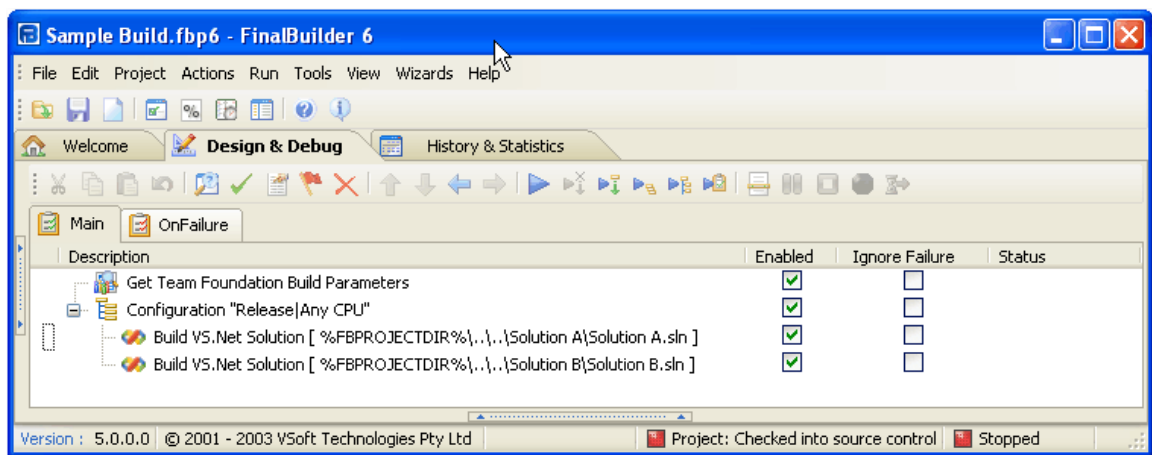


Before choosing this command, it is recommended that you create a local working directory for your Team Project's source control repository. If you do not have one set, go to Source Control Explorer, click on the "\$/<Team Project>" node, and choose "Get Latest".

When you choose the "New Build Definition with FinalBuilder..." command, the normal Team Foundation "New Build Definition" dialog will appear. Fill in the fields for this dialog normally. When creating a new project and choosing solutions to build, select any solutions which you intend to build with FinalBuilder as part of your project.



When you are done, click OK to close the Build Definition dialog. A new FinalBuilder project will be created to run during the Team Build. The FinalBuilder IDE will automatically load it.



The project contains actions for all of the solutions built in the Build Definition. The project is automatically bound to source control.

You can extend the FinalBuilder project in any way you like, to add extra functionality. You can even remove the Build VS.Net Solution actions and replace them with something else (like MSBuild actions or a VS.NET Solution Iterator.) All of the functionality described in the topic Team Build Functionality is automatically available

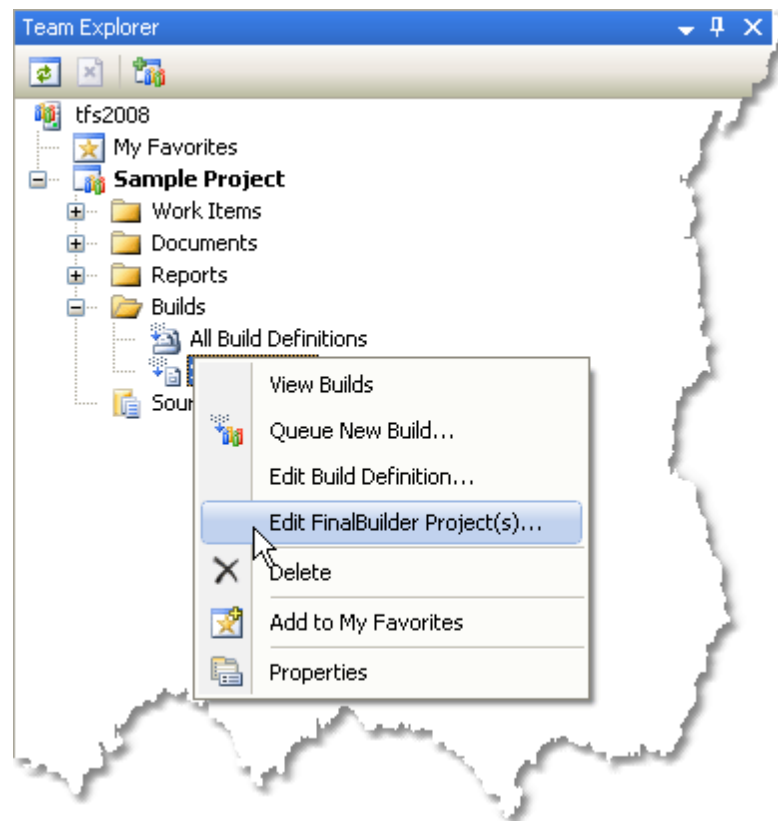
when this project runs under Team Build.

To further edit the Team Build specific functionality, go back to Team Explorer, right-click on the build type, and choose Edit FinalBuilder Project(s)

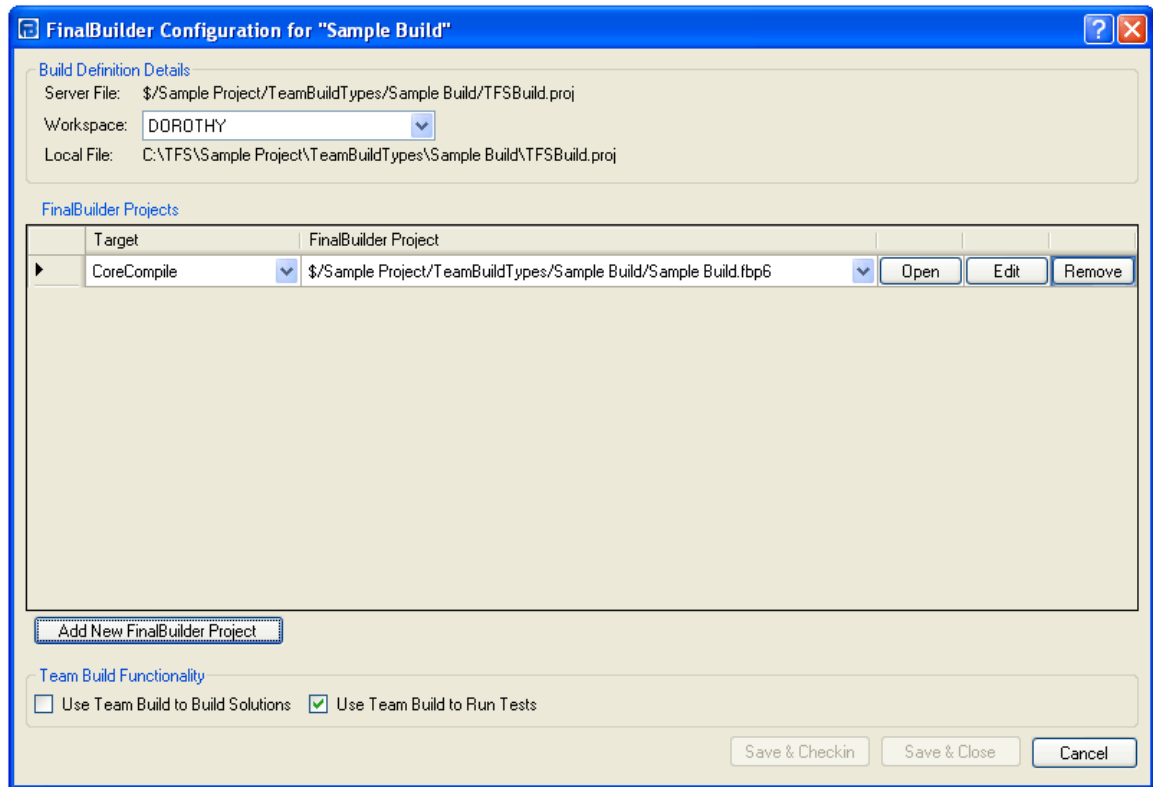
8.4.1.1.2 Edit FinalBuilder Project(s)

The Edit FinalBuilder Project(s) command lets you edit the FinalBuilder projects integrated into any Team Build Definition. A graphical editor automatically manages the Team Build project file, inside Visual Studio.

To use this command, right-click on any build definition in Team Explorer and choose "Edit FinalBuilder Project(s)...":



The "FinalBuilder Configuration" dialog for your build type will appear. This dialog shows all FinalBuilder projects built as part of the Team Build file.



Build Definition Details

This section contains information about the Build Definition file that the editor is working with

Server File - The path to the build definition on the server.

Workspace - The local workspace currently in use for editing the file. You can set this to any workspace that maps the server path.

Local File - The local path to the build definition file, in the current workspace.

FinalBuilder Projects

This grid shows all FinalBuilder projects currently defined in the build definition. Each of row corresponds to an FBCMD task in the project file.

Target - The FinalBuilder project runs as part of this MSBuild Target. All available Team Build targets are shown. The recommended target to use is *CoreCompile*, if FinalBuilder is being used to build the solution files; or *BeforeCompile/AfterCompile*, if Team Build is being used to build the solution files.

FinalBuilder Project - The path to the FinalBuilder project on the server. The combo box dropdown shows all FinalBuilder projects available on the server. There is also a dropdown entry called (*Create new FinalBuilder project file...*), which allows you to create a brand-new project file (with source control bindings), and add it to the Team Project repository.

Open - Press this button to open a local copy of the FinalBuilder project in the FinalBuilder IDE.

Edit - Press this button to open the FinalBuilder Task dialog, in order to set more advanced properties for this task.

Remove - Press this button to remove this FinalBuilder project from the build definition file.

Add New FinalBuilder Project

Click this button to add a new entry to the "FinalBuilder Projects" table. This adds a new FBCmd task to the team build project file.

Team Build Functionality

(This feature is not available when connected to a TFS 2005 server.)

Team Build comes with built-in functionality for building Visual Studio solution files and running Visual Studio tests. If you need more flexibility, you can override this built-in functionality and use FinalBuilder to perform these tasks.

If you uncheck the "Use Team Build to Build Solutions" checkbox, the CoreCompile target becomes available in the Targets column. Assign a FinalBuilder project to this target to use it instead of MSBuild to build your solutions.

If you uncheck the "Use Team Build to Run Tests" checkbox, the RunTest target becomes available in the Targets column. Assign a FinalBuilderproject to this target to use it for test execution.

Saving

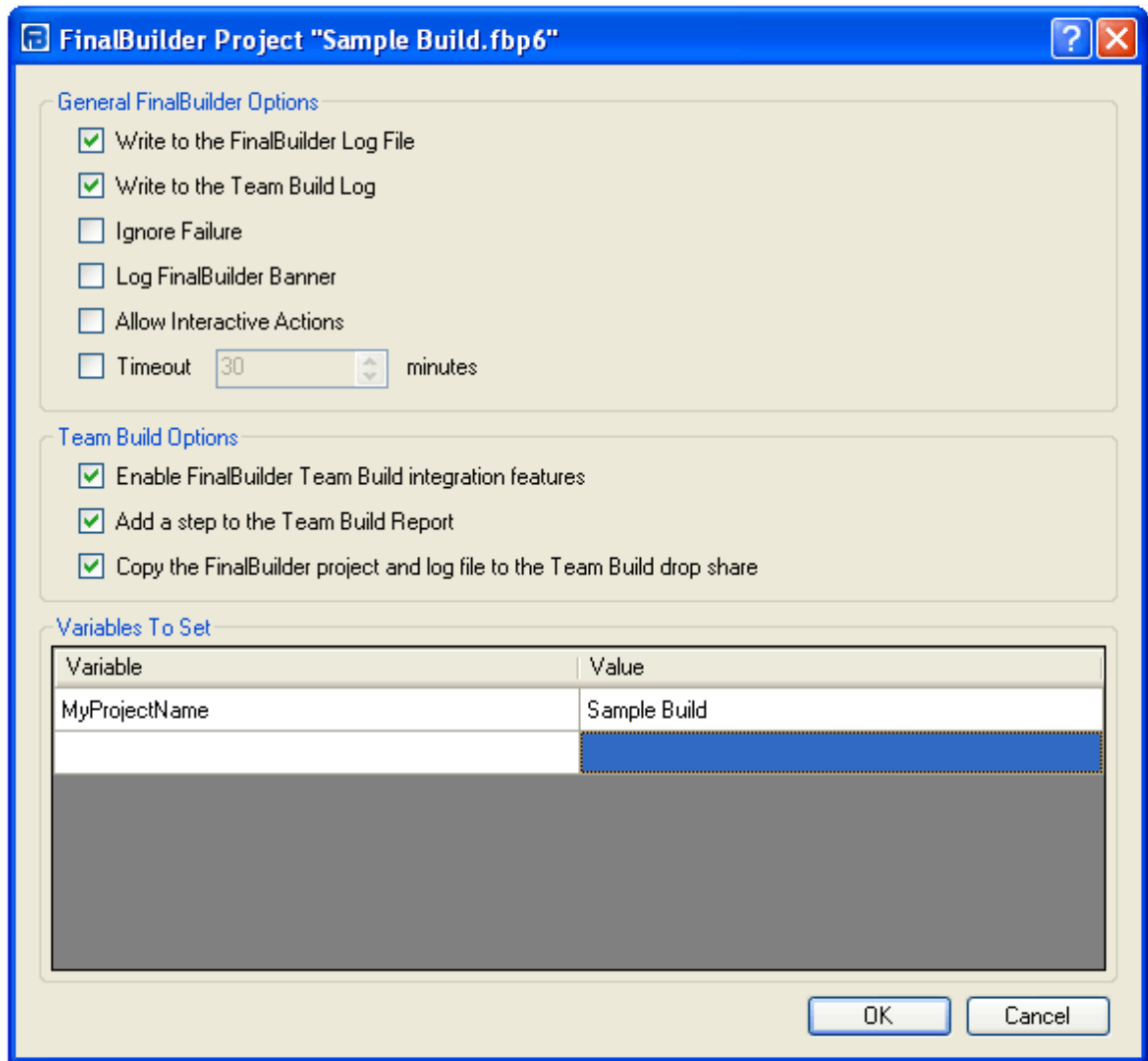
If the build definition project is modified, the "Save & Checkin" and "Save & Close" checkboxes become available.

Save & Close will save a local copy of the modified team build file, but will not check it in to the server. This means Team Build will still use the old project file when running builds.

Save & Checkin will save a local copy of the modified team build file, and then check it in to the server. This means Team Build will use the new project file when running builds.

8.4.1.1.2.1 FinalBuilder Task dialog

This dialog appears when you click the Edit button for a FinalBuilder project, in the Edit FinalBuilder Project(s) dialog. It shows custom options for a single FBCmd task to run a FinalBuilder project.



General FinalBuilder Options

Each option corresponds to a property of the FBCmd MSBuild task.

Write to the FinalBuilder Log File - If this option disabled, no output is written to FinalBuilder's internal log file.

Write to the Team Build Log - If this option is disabled, no output from the project is written to the Team Build log.

Ignore Failure - If this option is enabled, the build will not fail even if this FinalBuilder project fails.

Log FinalBuilder Banner - If this option is enabled, the FinalBuilder FBCMD "Banner" (version number, etc.) will appear in the Team Build log.

Allow Interactive Actions - If this option is enabled, interactive actions (like Prompt For Variables (Enhanced)) will be allowed to run. You almost certainly do not want to enable this option.

Timeout <> minutes - If this option is enabled, the FinalBuilder process will be timed out after this many minutes of execution.

Team Build FinalBuilder Options

Enable FinalBuilder Team Build integration features - Enable this option so that FinalBuilder "knows" it is running under Team Build and can support extra Team Build Functionality.

Add a step to the Team Build Report - Enable this option to add a step called "FinalBuilder Project <ProjectName>" to the Team Build report.

Copy the FinalBuilder project and log file to the Team Build drop share - This option is useful because Team Build creates a new "sandbox" workspace for each build. This option will deploy the just-run FinalBuilder project and log file to the drop share. You can open the project to view the full FinalBuilder log file for that build.

Variables To Set

Select any variables to set in the project. You can reference MSBuild properties like \$(MyPropertyName).

It is not necessary to use this field for common Team Build parameters. These can be retrieved using the Get Team Foundation Build Parameters action, inside FinalBuilder.

8.4.1.1.3 Team Build Functionality

The following extra functionality is automatically made available when a FinalBuilder project runs as part of Team Build.

Published MSTest (Visual Studio Test) Results

By default, the MSTest action will publish results to the Team Foundation Data Warehouse, so they are available from Build Explorer and Team Foundation Reports.

Test Statistics

Other testing tools actions (such as NUnit, MbUnit, etc.) will automatically publish test statistics to the Team Foundation Data Warehouse. These statistics (tests run, tests failed, etc.) are available from Build Explorer and Team Foundation Reports.

If there are failed tests but the FinalBuilder project completed successfully, the build result is automatically updated from "Success" to "Partial Success".

Compiler Results

Results from compiler actions also publish metrics (errors, warnings, etc.) to the Team Foundation Data Warehouse.

8.4.1.2 Get Team Foundation Build Parameters Action

This action allows you to retrieve some properties of the Team Build instance which can then be used in your FinalBuilder project, when running under Team Build. It can also be used to simply check if a project is running under Team Build, or natively inside FinalBuilder.

Note: In order for this action to work properly, the Team Build type must be set up using the Team Build Addin for Visual Studio.

Get Team Foundation Build Parameters

General Runtime **Team Build Parameters**

Behaviour

☐ Fail the action if not run as part of Team Build

% Set Variables to Team Build Properties

Team Server URL: TeamServer

Team Project:

Build Id:

Platform:

Flavor:

% Set Variables to Team Build Paths

Deployment Folder: DeploymentDirectory

Solution Root Folder: SourceRoot

Note: This action needs to be run as part of a Team Build under Microsoft Team Foundation Server.

OK Cancel Help

Fail the action if not run as part of Team Build

Enable this property to have the action fail if not running under Team Build. If this property is not enabled, the action will run but does not do anything.

Set Variables to Team Build Properties / Paths

Optionally, set any FinalBuilder variables that you would like set to the values of Team Build properties.

Team Server URL - The URL of the team server instance, ie *http://myteamserver:8080/*. In the example above, the Team Server URL will be saved to the FinalBuilderproject variable "TeamServerURL".

Team Project - The name of the current team project.

Build Id - The Id of the current build. For Team Foundation 2008, this normally looks like <BuildTypeName>_<Date>.<No>

Platform - The platform name for the current build. Defaults to "Any CPU".

Flavor - The flavor name for the current build. Defaults to "FinalBuilder".

Deployment Folder - The target drop folder path for the current running build. Useful if you want to deploy any extra files to the drop share.

Solution Root Folder - The source folder for the current build's "sandbox" environment. Equivalent of the \$(SolutionRoot) MSBuild property in Team Build, or the \$(SourceDir) property for workspace mappings.

8.4.2 Windows Workflow (TFS 2010)

In order to support the Workflow Foundation build technology used by Team Foundation Server 2010, FinalBuilder 7 ships with a custom TFS Workflow activity and two XAML build templates, which are located in <FinalBuilderInstallDir>\TFS Templates:

- FinalBuilderBuild.xaml : replaces the build and test processes in the default TFS template with a call out to FinalBuilder. This template uses TFS to create the workspace, get the code out of source control (including the FinalBuilder project) and otherwise set up the Build Agent. It then calls FinalBuilder and runs the project you have specified.
- FinalBuilderPostBuild.xaml : runs FinalBuilder at the end of the default TFS template. TFS is used to build and test the solution and the FinalBuilder project is run afterwards to do any post-build operations. It allows you to easily extend the TFS build with FinalBuilder, taking advantage of all of its built-in actions

In both cases the output from FinalBuilder will appear in the TFS build log. If the FinalBuilder build fails, the TFS build will also fail.

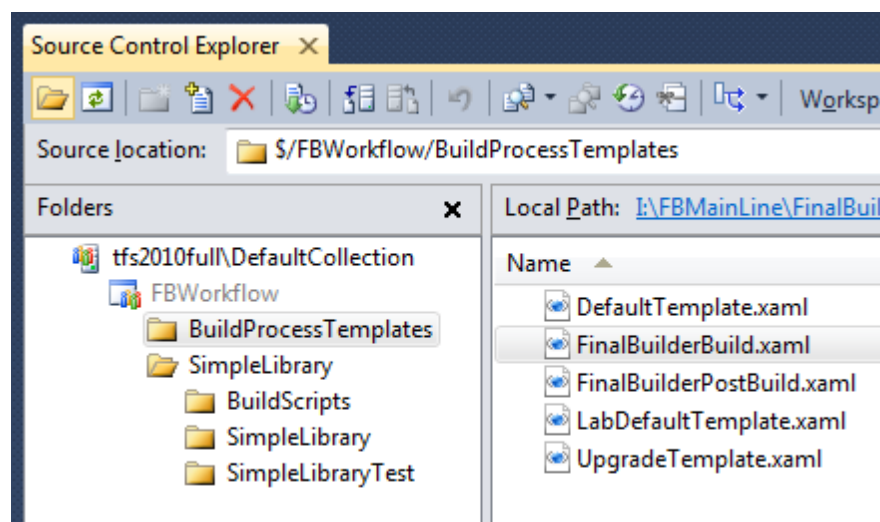
To get started, you need to

- Add the FinalBuilder templates to TFS
- Create a FinalBuilder project that uses TFS data
- Make your project accessible to TFS, and
- Configure and Queue a Build

It is assumed that you have TFS installed and configured, and are comfortable using it.

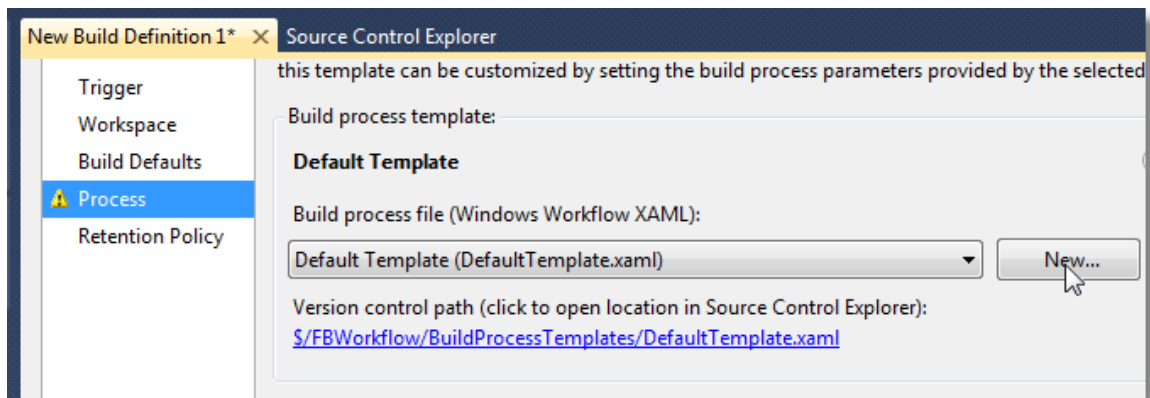
8.4.2.1 Adding FinalBuilder templates to TFS

Before you can use the FinalBuilder templates, they must first be added to TFS. The first step is to add the FinalBuilder templates (from <FinalBuilderInstallDir>\TFS Templates) to your Team Project's source control. I suggest adding it to <Your Team Project>\BuildProcessTemplates, alongside the TFS-provided templates.

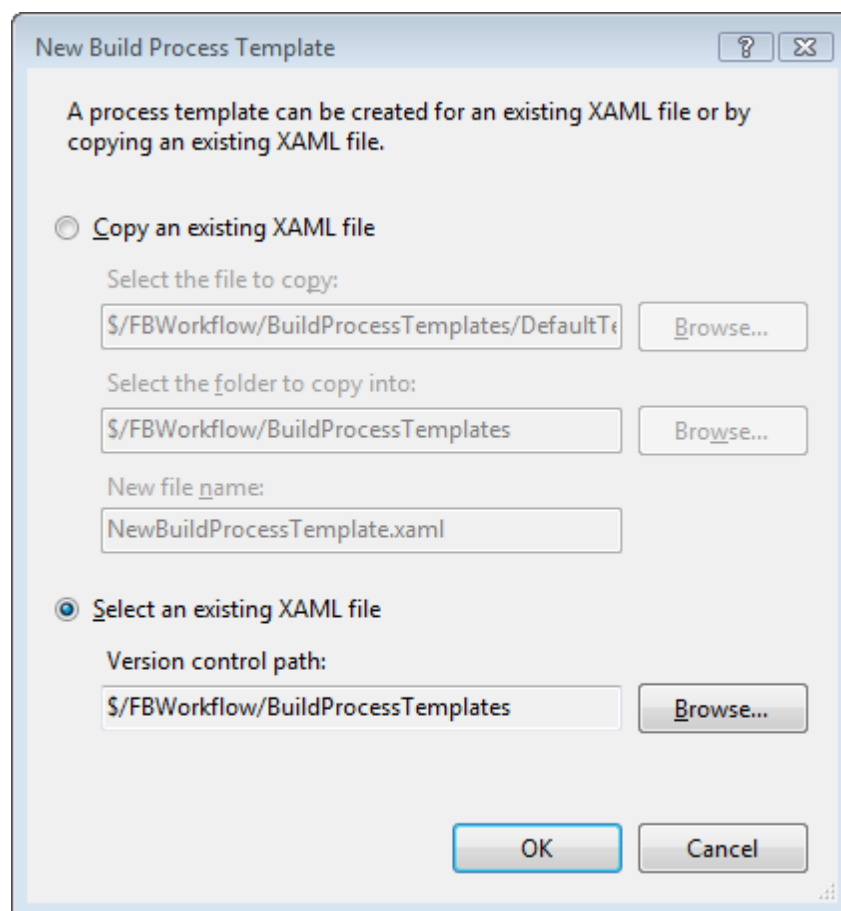


Now, in Team Explorer, under your team project right-click on [Builds] and choose [New Build Definition...]. Go to the Process section. Under "Build process template" choose "Show details". The drop-down list allows you to choose from any of the templates that have been added. Because this is the first time you've used the

FinalBuilder template it will not be in the list. To add it, click [New...]



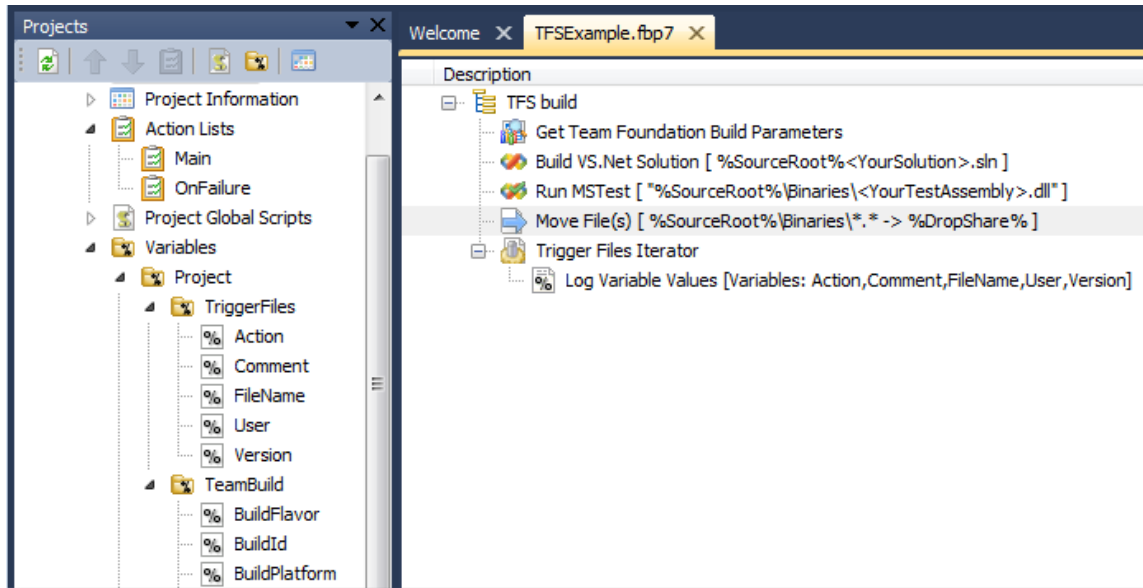
Choose [Select an existing XAML file] then browse to the location in source control where you added the FinalBuilder template. Press [OK] until you are back at the Process section.



The FinalBuilder project template should now be selected in the drop down list. The next time you create a new build definition in this team project the template will be in the list.

8.4.2.2 Using TFS data in FinalBuilder

To use TFS data in FinalBuilder you need to make use of some TFS-specific actions. To get you started, there is an example project in <FinalBuilderInstallDir>\TFS Templates called TFSExample.fbp7. This project contains actions to perform a basic TFS-integrated build.



The project contains 6 actions and some related Project Variables:

Get Team Foundataion Build Parameters assigns TFS data to the specified project variables. There is no need for you to modify this action.

Build VS.NET Solution builds a Visual Studio.NET solution. On the [Solution] tab you will see that the Solution File is set to %SourceRoot%\<YourSolution>.sln. Replace <YourSolution> with the name of the solution that you wish to build. Also note that Configuration is set based on the values coming from TFS.

On the [Paths] tab you will see that the Output Directory is set to %SourceRoot%\Binaries. You may change this if you wish, but it's it not necessary.

Run MSTest uses MSTest to run your test assemblies. On the [Settings] tab you will see that the Test Container is set to %SourceRoot%\Binaries\<YourTestAssembly>.dll. Replace <YourTestAssembly> with the name of your test assembly, or add other assemblies to the list. On the [Publish Results] tab the action is set to automatically publish the results to the TFS server, you can change this if you like.

Move Files moves the built and tested assemblies to the drop location.

Trigger Files Iterator is a special action that iterates through the list of files that triggered the build. In TFS-lingo, it is iterating through the changesets associated with the build. Like [Get Team Foundation Build Parameters], this action populates project variables with data from TFS.

Log Variable Values simply logs the trigger files data. It's provided as an example of how you access the data, should you wish to use it for another purpose.

Important Notes

The [Ignore Failure] option is checked for the *MSTest* action. If any unit tests fail, the *MSTest* action will fail, and setting [Ignore Failure] allows the FinalBuilder and TFS builds to continue. Un-check [Ignore Failure] if you would prefer the build to halt on failed tests. In either case, test failures will appear in the TFS build log and in TFS reports.

Because the drop location may be on a different server to the build agent, it is important that MSTest runs on files located on the build agent. Unless you explicitly set up the trust relationship, .NET will not allow executing of assemblies on remote machines. This is why we build and test in a directory under % SourcesRoot% and then move the files to the drop location after testing.

Removing MSTest

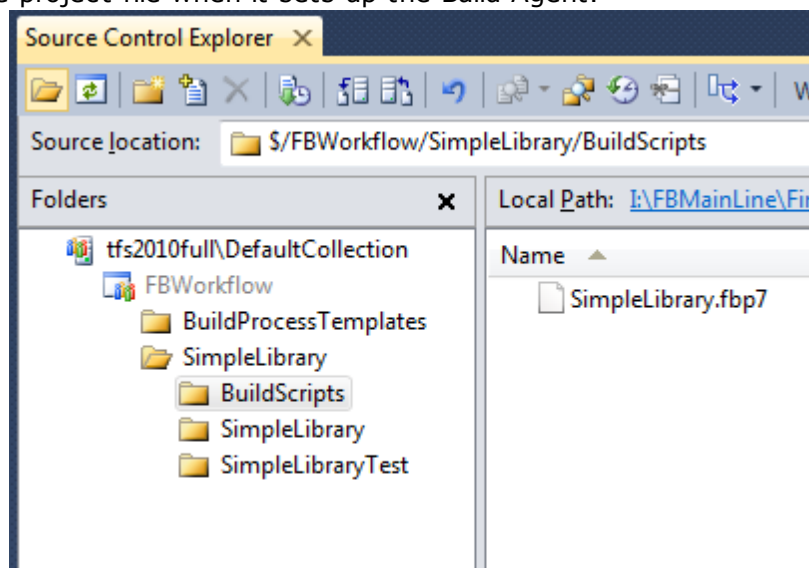
If you do not want to use MSTest, you can delete the [MSTest] and [Move Files] actions. Then, edit the *Build with VS.NET* action and change the Output Directory (on the [Paths] tab) to be %DropShare%.

Customising the Project

The FinalBuilder project running via TFS should be able to execute actions as per usual. Just keep in mind that you are executing on a Build Agent using credentials that the Agent runs under.

8.4.2.3 Making your project accessible to TFS

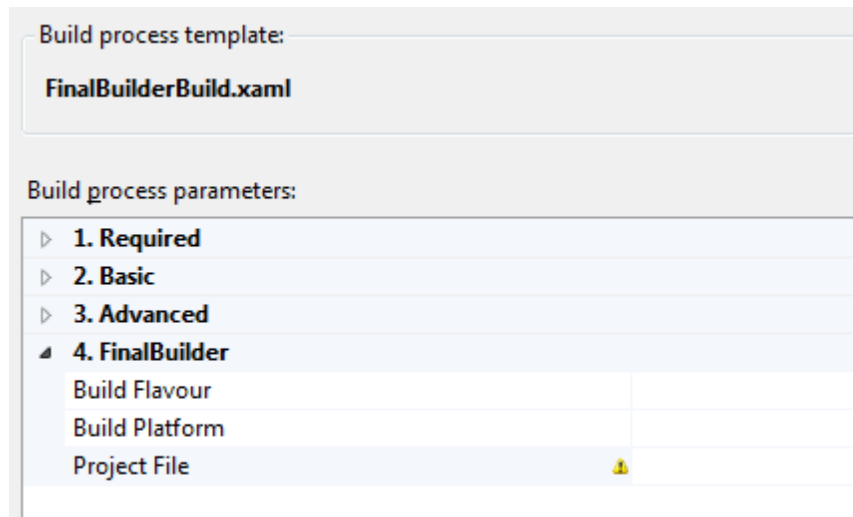
Add your project file into source control as part of the same team project that as the .NET solution you wish to build. If you add it to a different team project, TFS will not 'get' the project file when it sets up the Build Agent.



8.4.2.4 Configure and Queue a Build

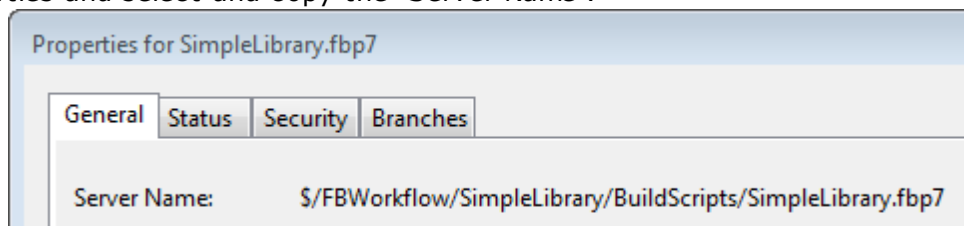
Create a new build definition, and on the Process tab, hit "Details" and select one of the FinalBuilder templates from the drop down list.

Below the "Build process template" area is an area called "Build process parameters". This has now been updated to contain parameters required by the FinalBuilder template.



You will see a yellow warning symbol in the FinalBuilder area of the parameters. This is because Project File is a mandatory parameter and it has not yet been supplied.

You need to specify the location of the FinalBuilder project in source control. To find the location, open the source control explorer, locate project file, right-click, choose properties and select and copy the "Server Name".

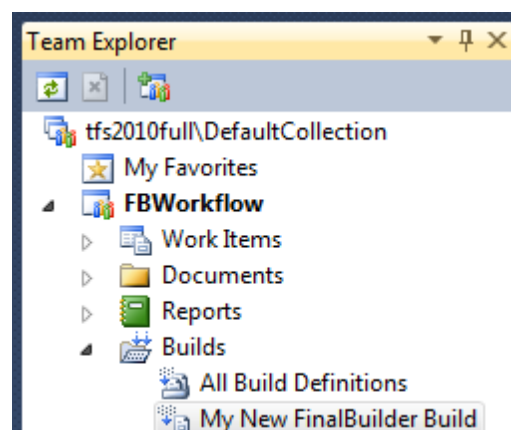


Go back to the Build Definition window and paste this value into the Project File parameter.

There are 2 optional parameters:

- Build Flavour: An arbitrary string that you label the build with, for example Debug or Release. If unspecified it will default to "Debug".
- Build Platform: The target platform for the build. It should be set to Any CPU, x86 or x64. The default is Any CPU.

You can now save your build definition and it will appear under "Builds" in Team Explorer. Right-click on this and [Queue New Build...]



9 Tips and Tricks

9.1 Two Environments

A common situation is developing a build script 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 and easier version control, you may wish to use a single shared INI file as follows.

Let's assume that whenever you do a build using the development machine (named *John*), you want it to take place in C:\Builds\Testing. When you do a build on the production machine (named *Build*), it takes place on J:\Builds\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:

```
[Build]
BuildHome=J:\Builds\Latest

[John]
BuildHome=C:\Builds\Testing
```

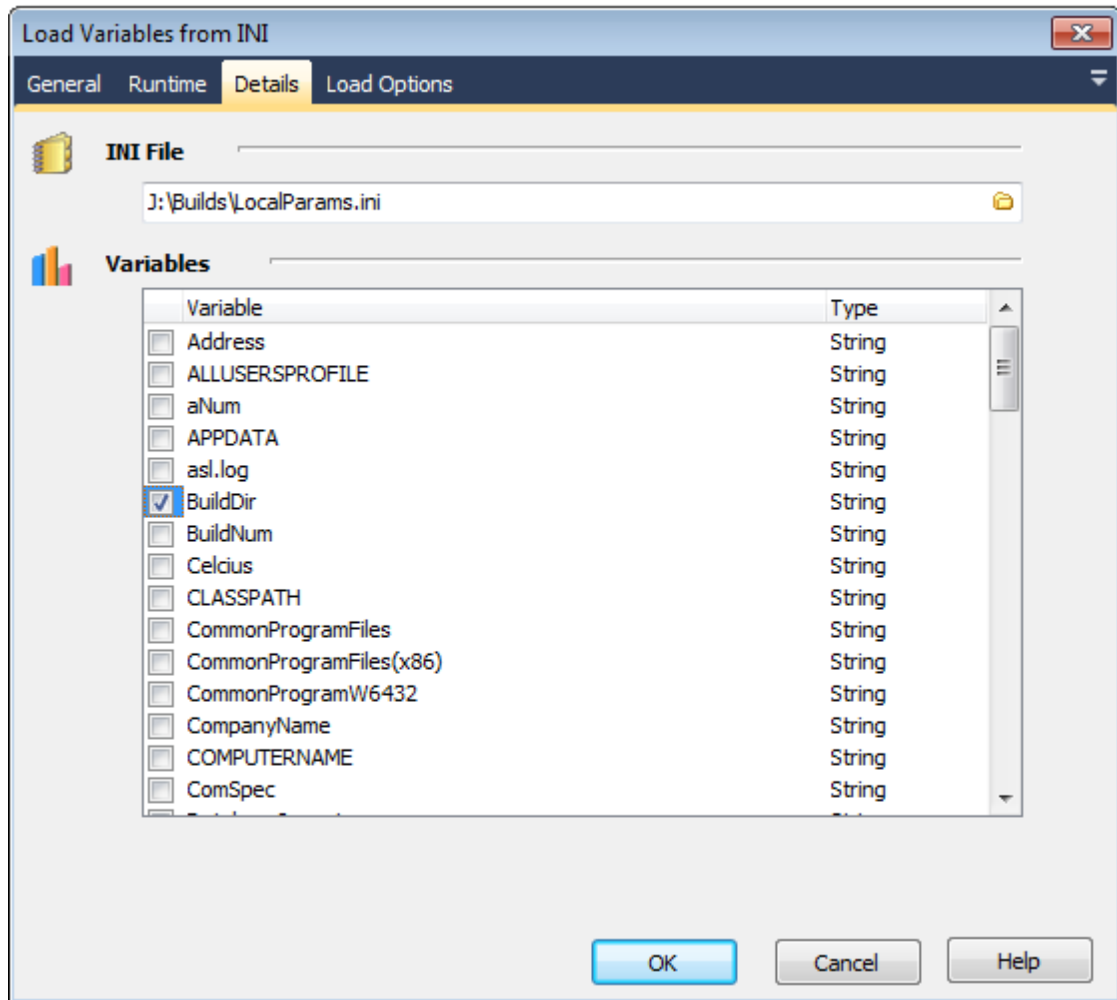
Name the file something like "LocalParams.ini".

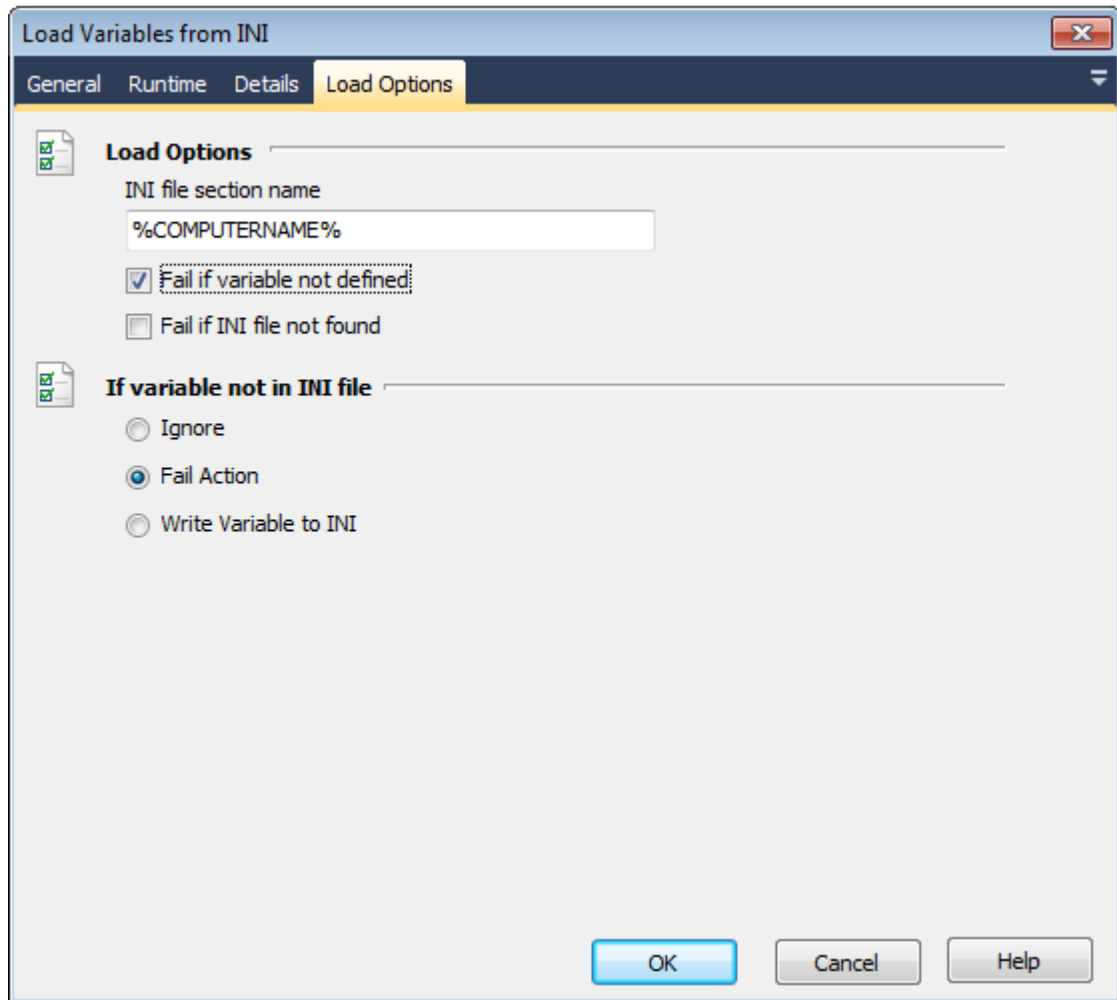
2. Make your build depend on these parameters

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 build

At the start of the build, 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:





4. The finished product

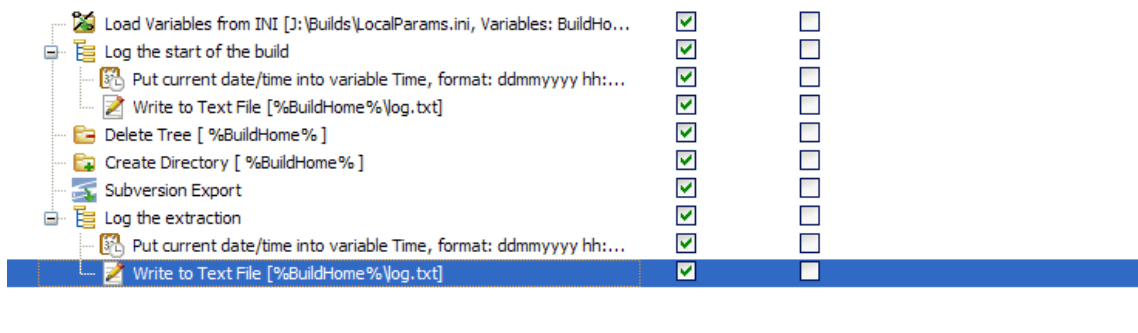
Description	Enabled	Ignore Failure	Status
Load Variables from INI [J:\Builds\LocalParams.ini, Variables: BuildHome]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Delete Tree [%BuildHome%]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Create Directory [%BuildHome%]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Subversion Export	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

9.2 Reusing Logic

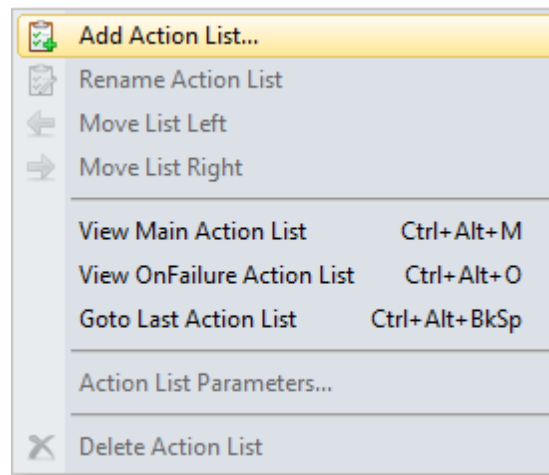
When you have come up with a useful sequence of actions, you may want to reuse it. Here are several methods, in increasing levels of sophistication.

Action lists

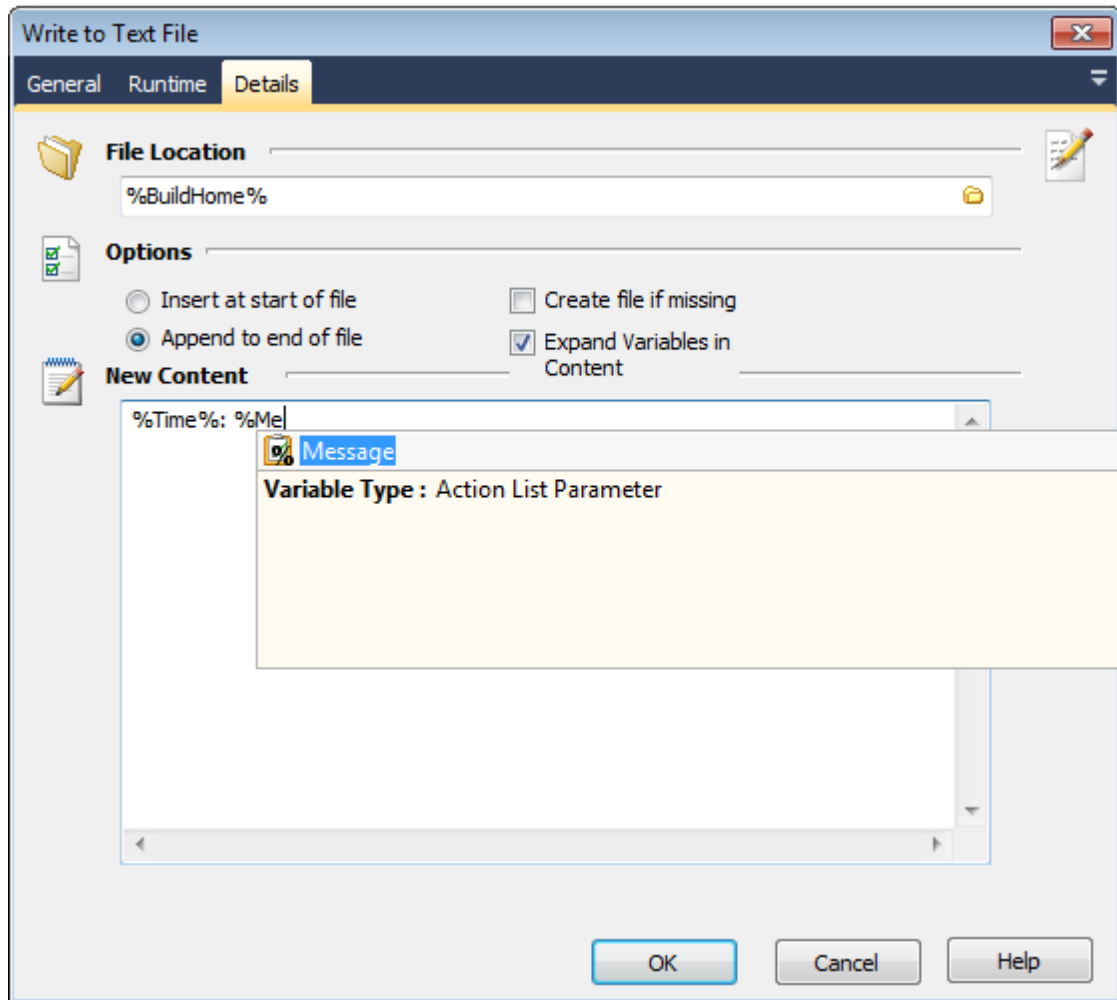
Let's take the example of a couple of actions used to log to an external log file with a date and time stamp. Before writing to the log with the Write Text File action, the Time variable is updated using the Get Date Time action.



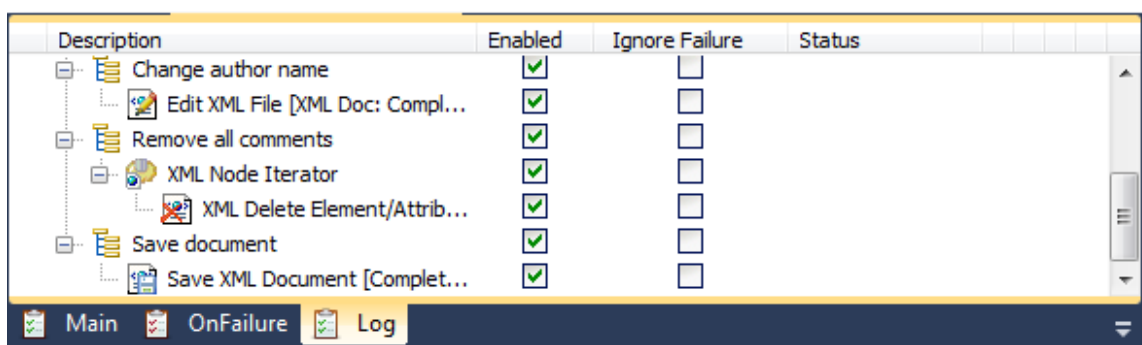
Here, the log actions have been copied. This is not ideal, as if we improve the logic later on, we'll have to edit all those actions. Instead, create a new action list, called Log:



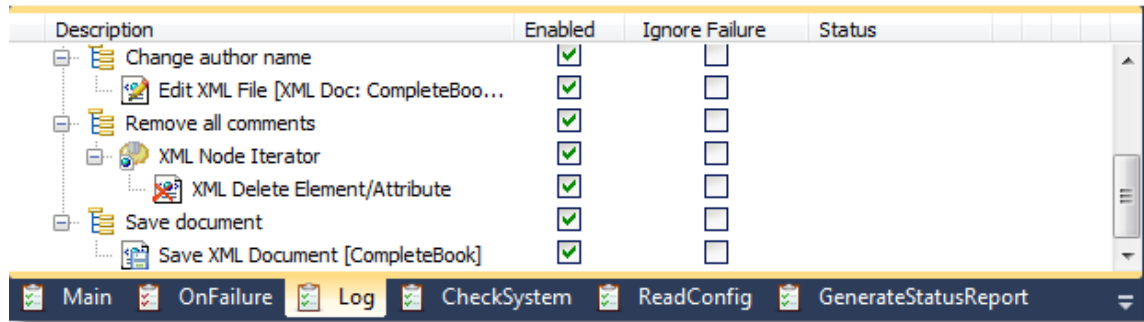
Move the actions to the new action list. Now, create a parameter to contain the log message, by right clicking on the action list title.



The action list now looks like this:



The final step is to replace the original calls with Run Action List actions.



You can now call these action lists from any project.

9.3 A Configurable build

Frequently, when you set up a FinalBuilder project to build a product, you actually have several closely related products to build. There are several ways to solve this problem. This tutorial demonstrates one approach, using INI files, prompts and switch statements to make one build process configurable.

First, define a short code for each distinct product. Let's say we have two products, ABC and DEF. However, sometimes ABC needs to be built with an extra sub-product. We'll call that combination ABC2.

INI File

Create an INI file with some parameters for each build:

```
[ABC]
SolutionFile=ABC.sln
Description=A brilliant calculator
SVNBranch=Prod/ABC

[DEF]
SolutionFile=DEF.sln
Description=Data encryption filter
SVNBranch=Prod/DEF

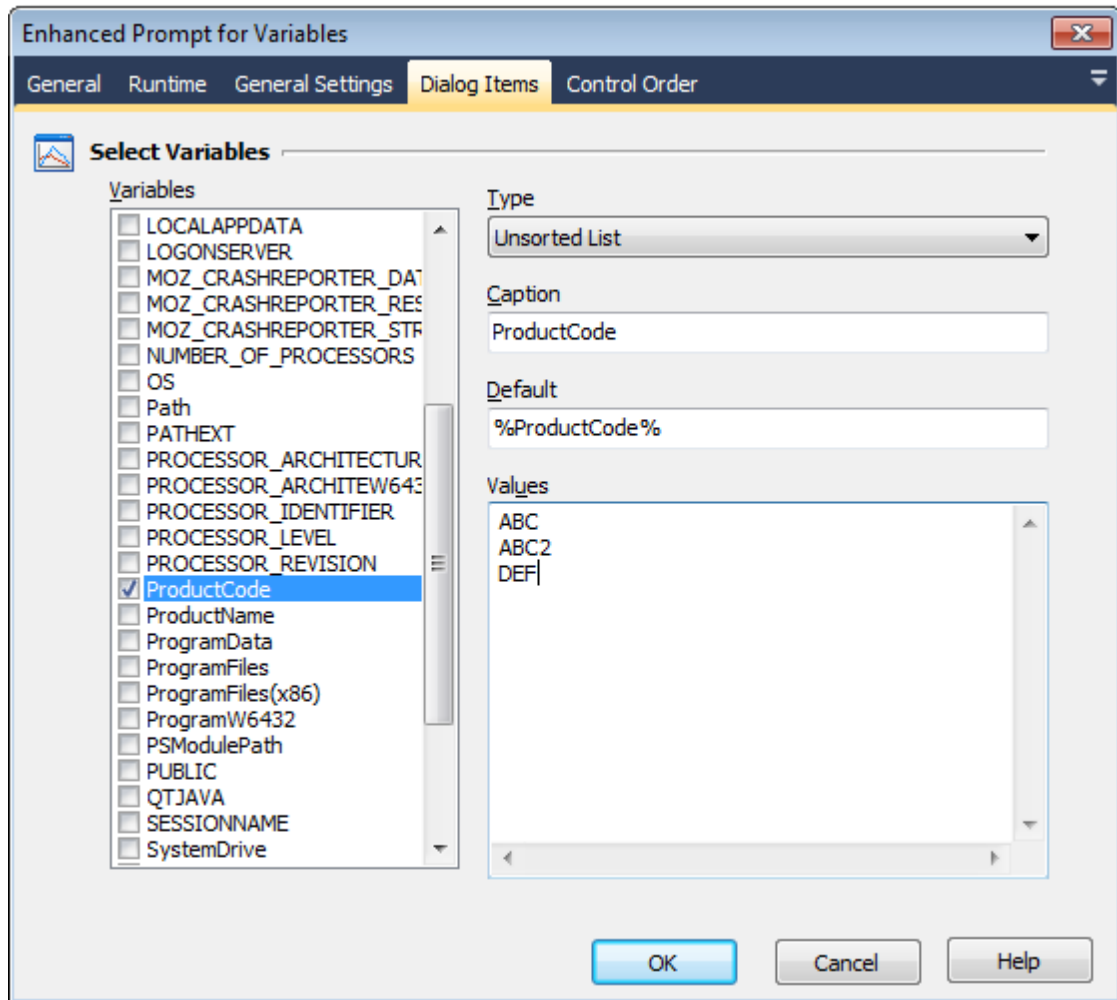
[ABC2]
SolutionFile=ABC.sln
Description=A brilliant calculator PLUS
SVNBranch=Prod/ABC
BuildABC2=True
```

Variables

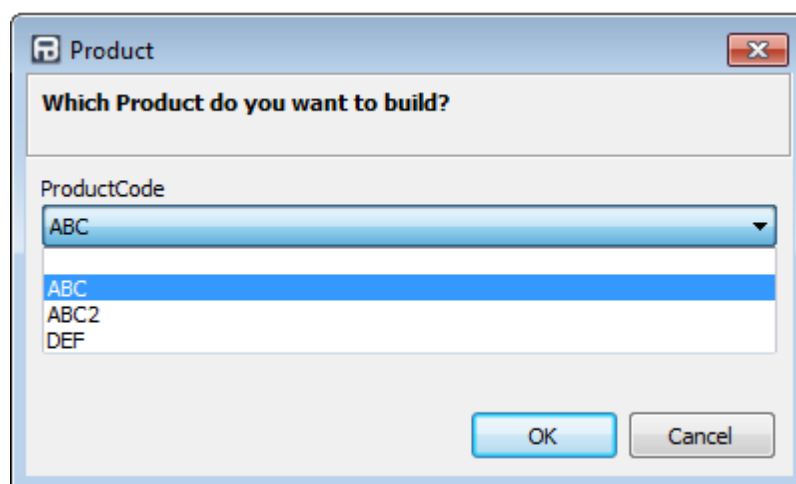
Create a "productcode" variable. Make it persistent, so that each build can default to the same type as the previous one.

Prompt

Create a "Enhanced prompt for variables" action. Here you will give the user the choice of which build to create. 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.



At runtime, this will look as follows:



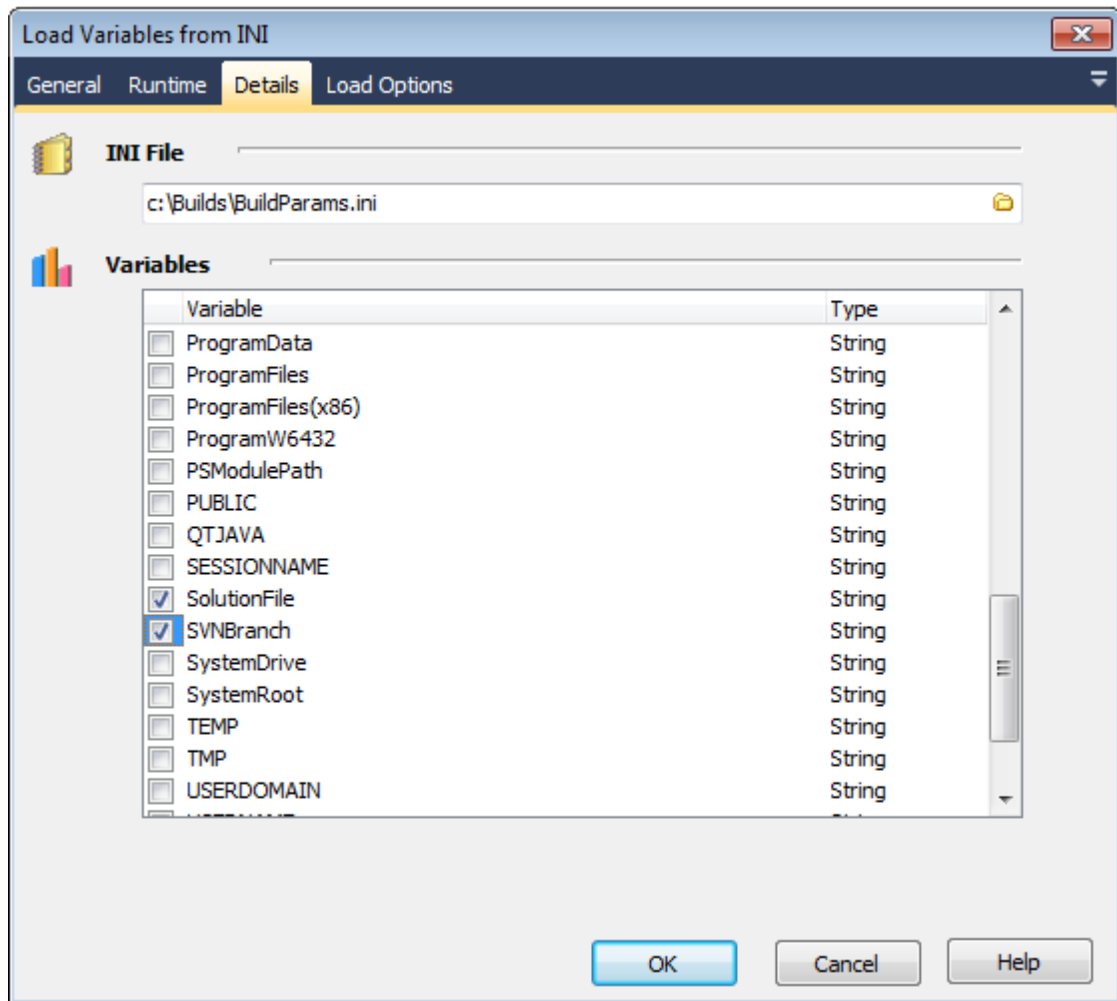
Load INI file

Next, we need to load the settings for the chosen build. We first load the mandatory

settings, then the optional ones.

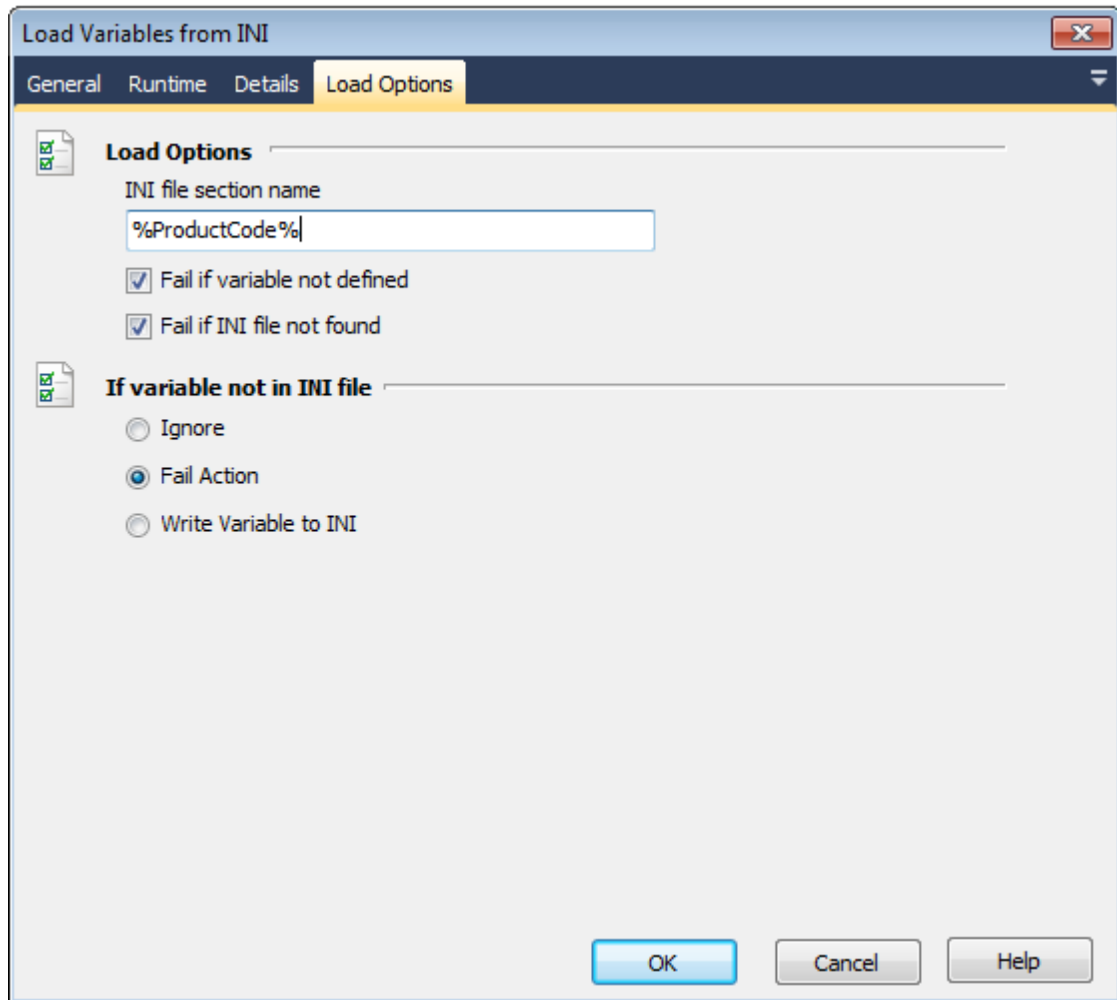
We create one "Load Variables from INI" action with these settings:

- INI file: C:\Builds\BuildParams.ini
- Variables: Description, SolutionFile, SVNBranch
- INI file section name: %ProductCode%
- If variable not in INI file: Fail Action



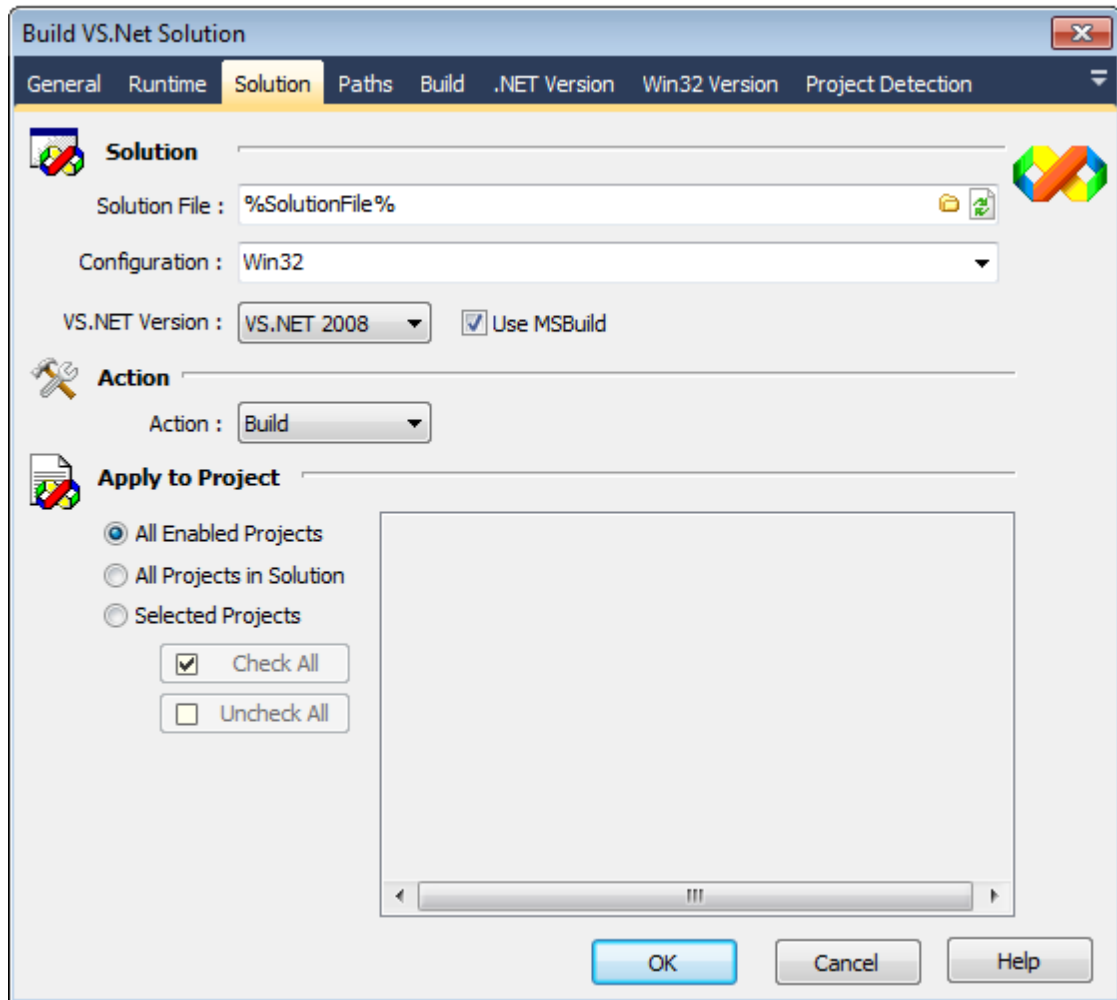
Next, we create one for the optional settings:

- INI file: C:\Builds\BuildParams.ini
- Variables: Description, SolutionFile, SVNBranch
- INI file section name: %ProductCode%
- If variable not in INI file: Fail Action



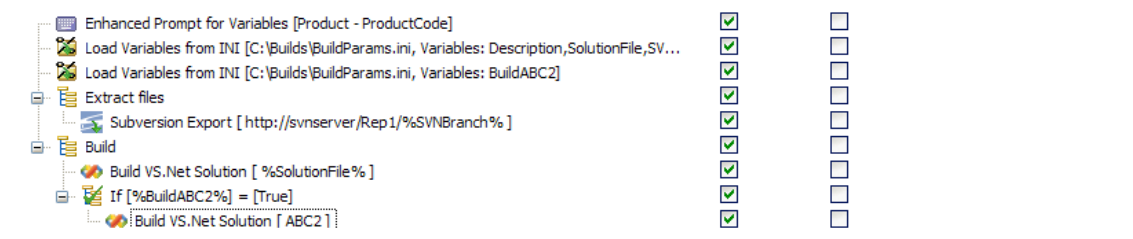
Use the variables

Now construct your build process, using these variables everywhere:



Conclusion

The overall result looks like this:



You now have a single build which is capable of building different products. All the same core logic - loading variables, extracting files from your version control system, building - is stored in only one place. This is much better than having a separate FinalBuilder project for each product. In that situation, if you found a problem in one build, you would have to fix it in every project individually, making your build process more error prone and labour intensive.

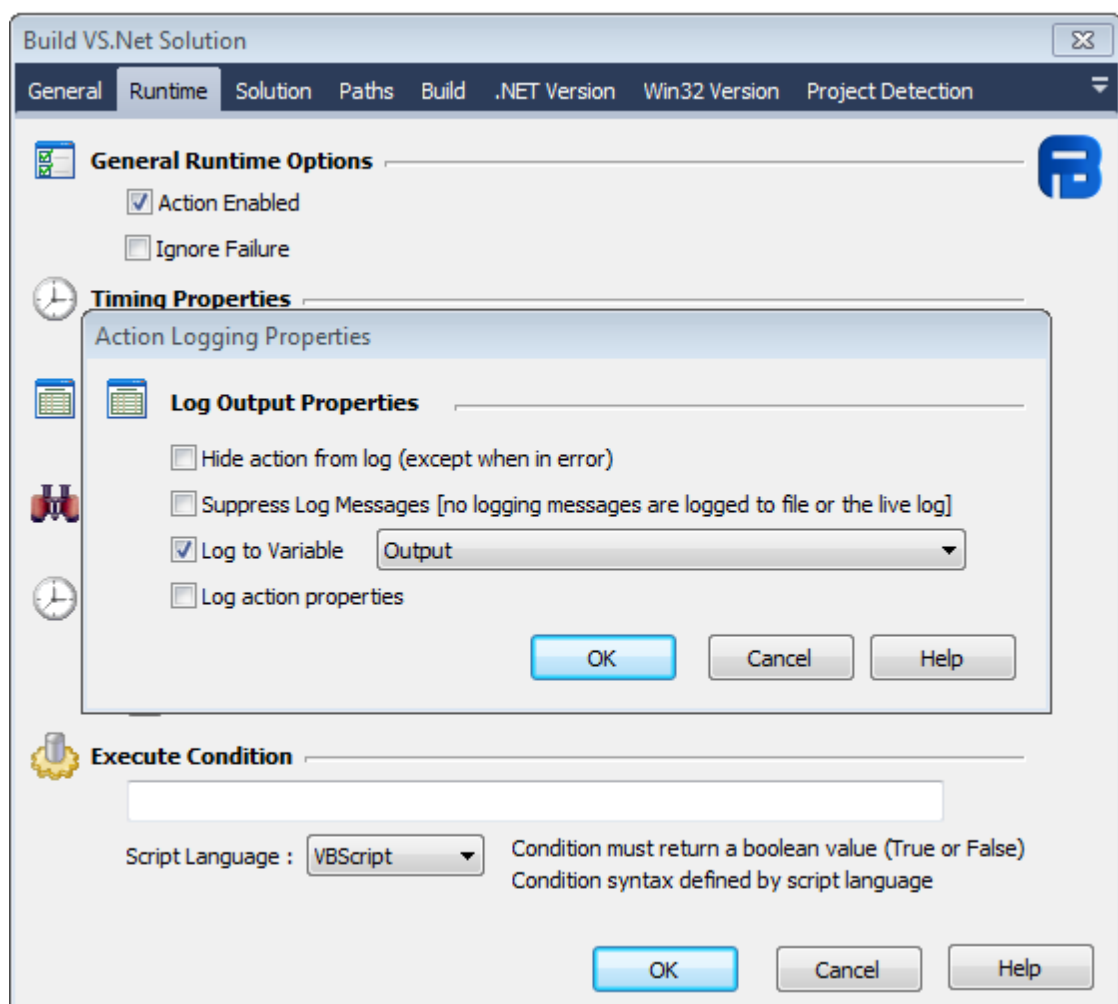
9.4 Analysing Output

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.

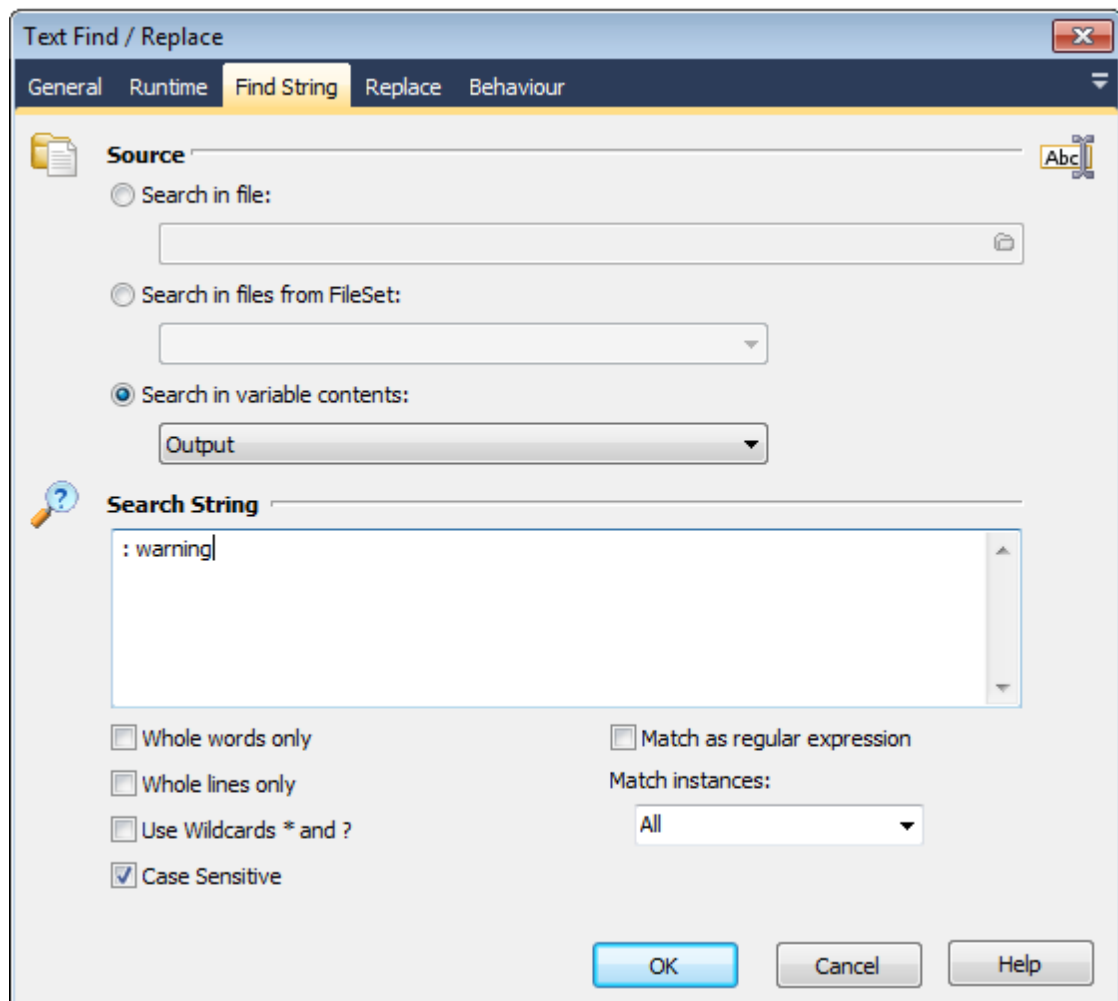
Counting warnings

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"

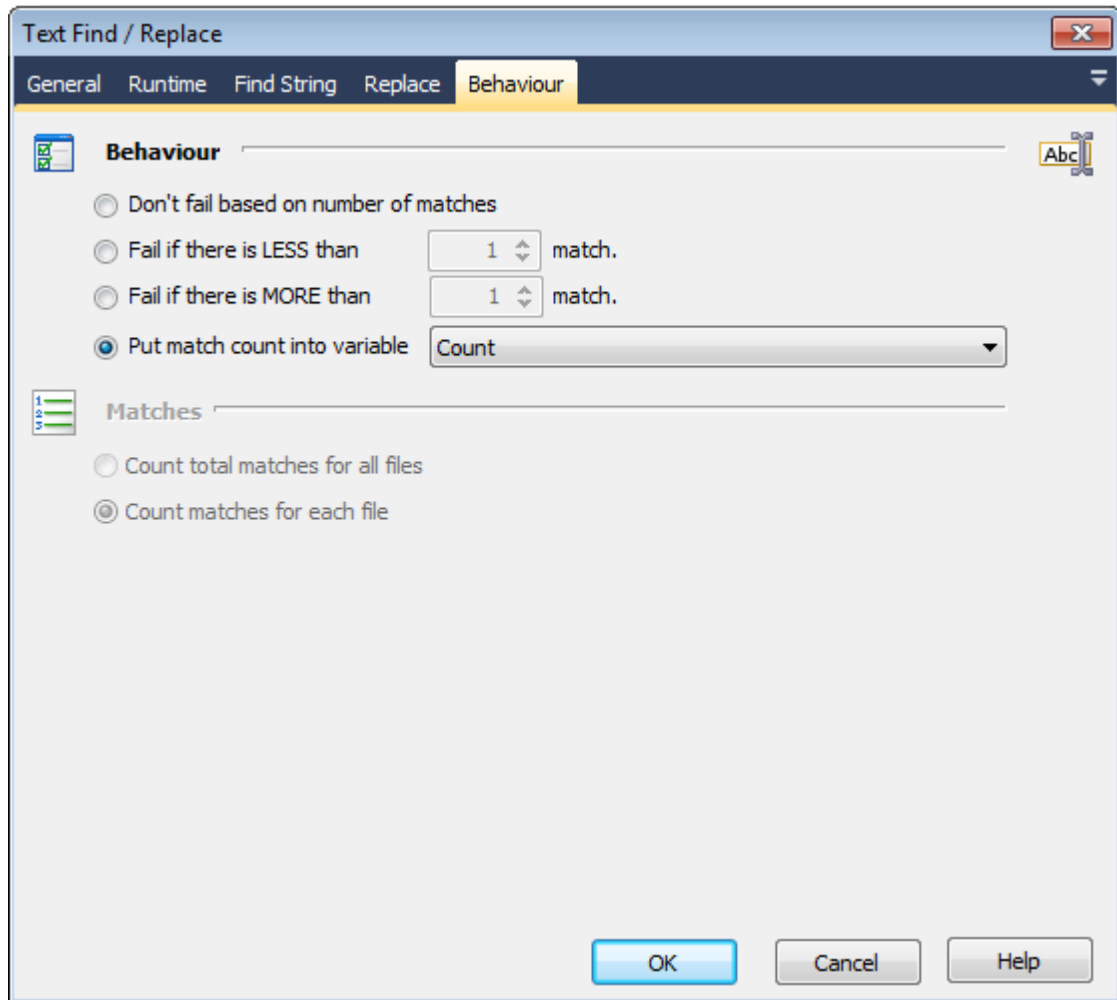
First, on the **Runtime** tab of the relevant action, go to "Logging Properties" and select "Log to Variable".



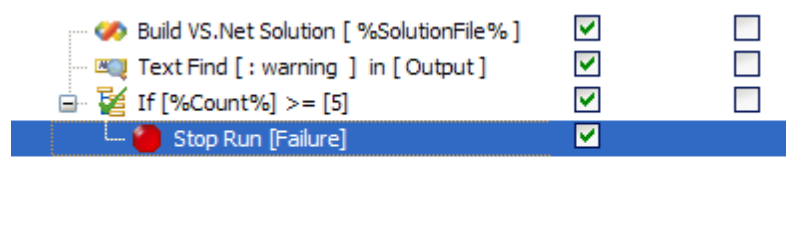
Next, use a "Text Find / Replace" action to count the number of times the string appears:



On the **Behaviour** tab, set the variable to hold the number of matches:



Now you're all set to use the variable however you like:

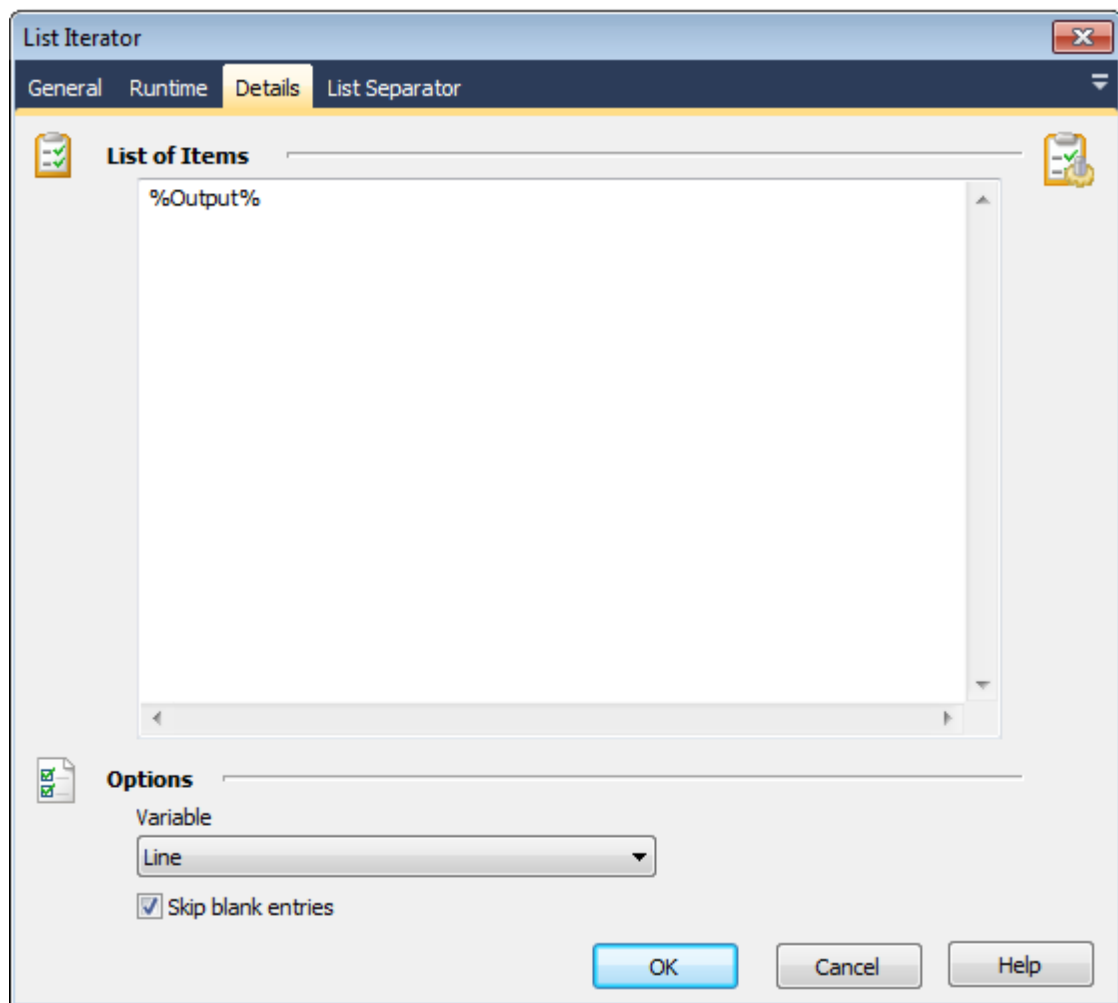


Processing a log line by line

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.

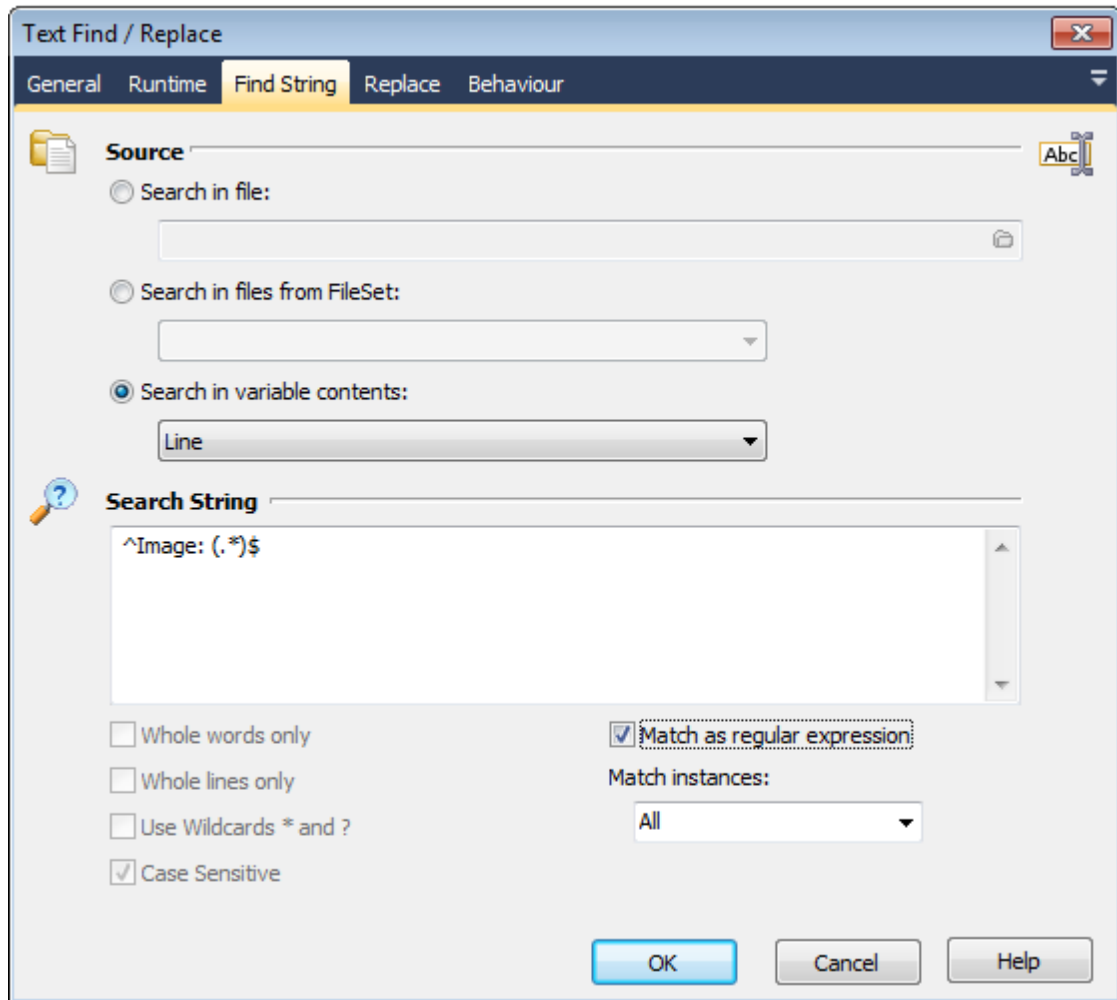
Start by logging the output of action to the Output variable. You will also need a variable to hold each line of output. Call it "Line".

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, FinalBuilder has a very large upper limit on variable size.



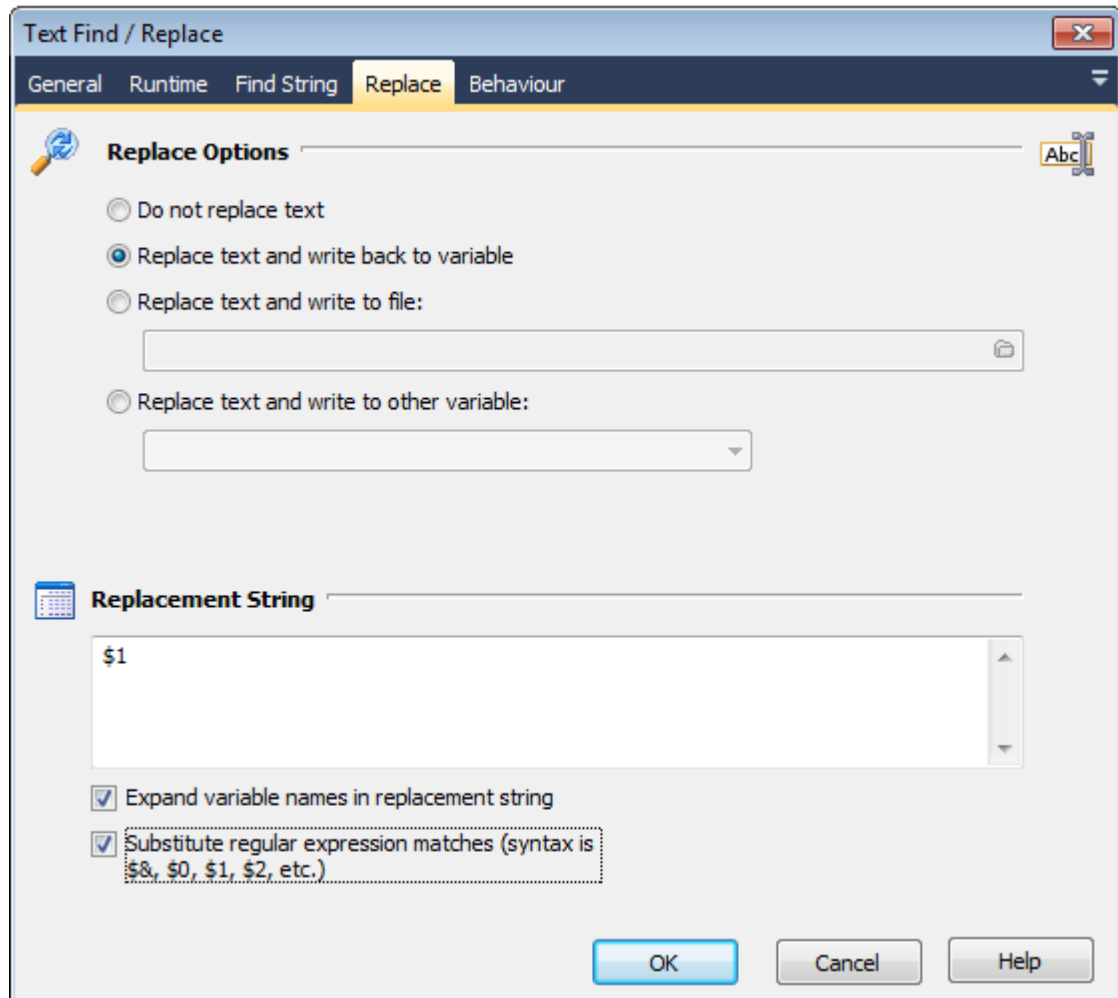
Leave the "List Separator" settings as the default: a carriage return/line feed.

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:

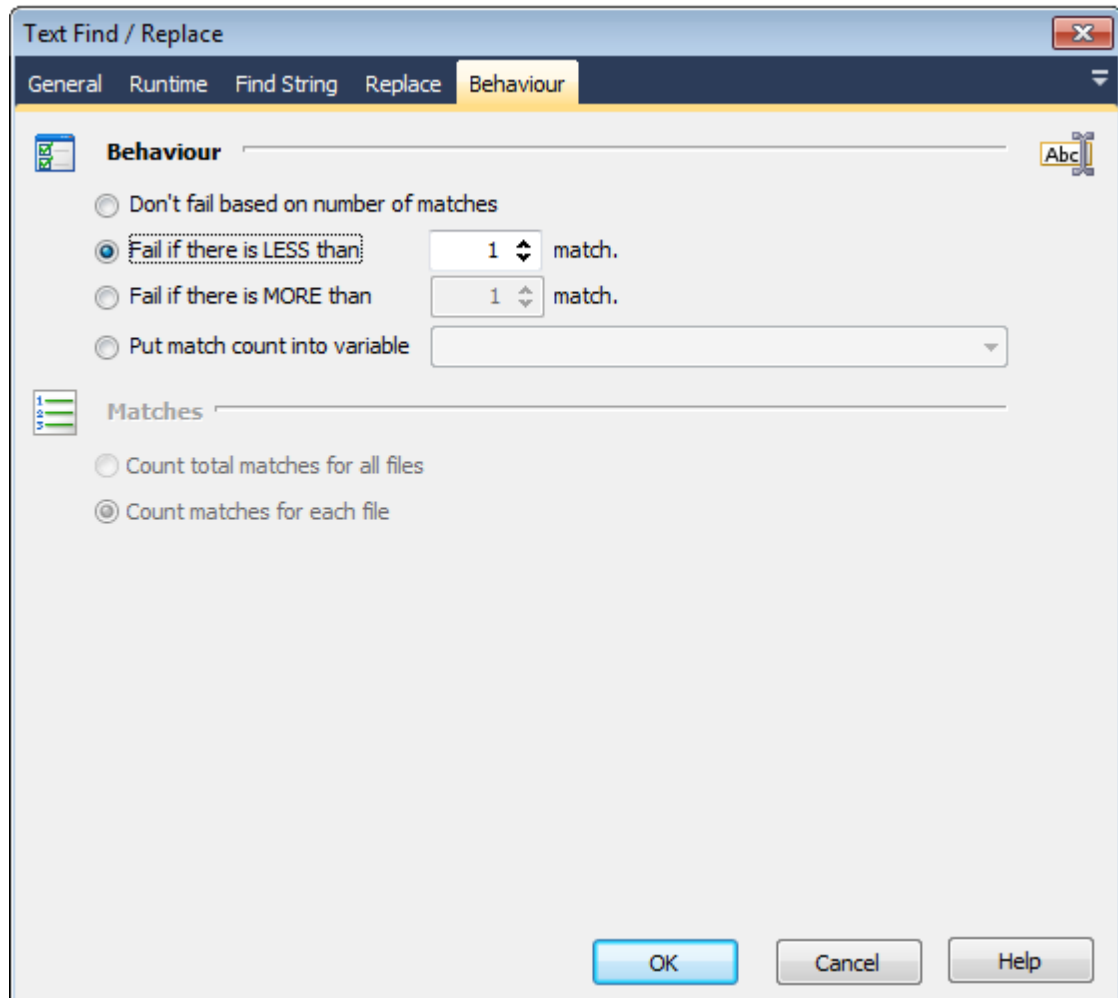


This regular expression means *"beginning at the start of the line, match the word Image, a colon, a space, then store the whole rest of the line as subexpression 1"*.

On the **Replace** tab, we write that subexpression back to the same variable:



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.



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.

Description	Enabled	Ignore Failure	Status
Execute Program [ProcessFiles.exe]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
List Iterator [Iterator variable: Line]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Text Replace [^Image: (.*)\$...] with [\$1] in [Line]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Copy File(s) [%Line% -> J:\Backup\]	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

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.

9.5 Documenting your Project

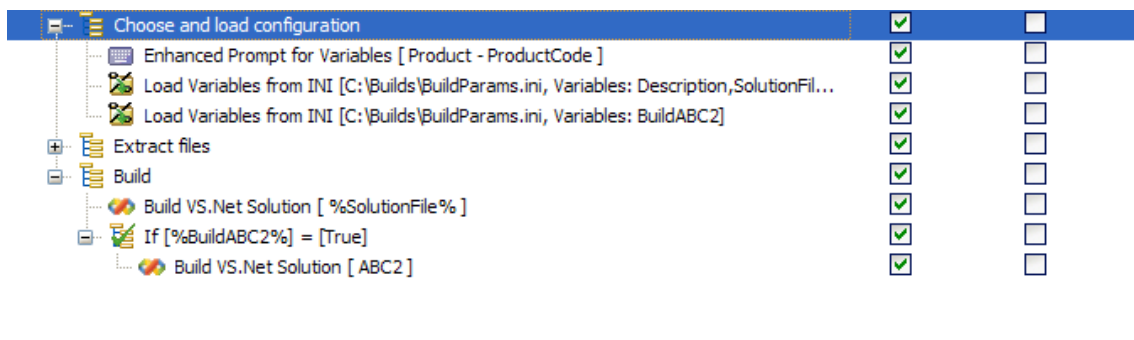
As your FinalBuilder project gets larger and more complex, you should document it, to help you and others maintain it in the future. Here are some of the ways FinalBuilder lets you do that.

Topics covered:

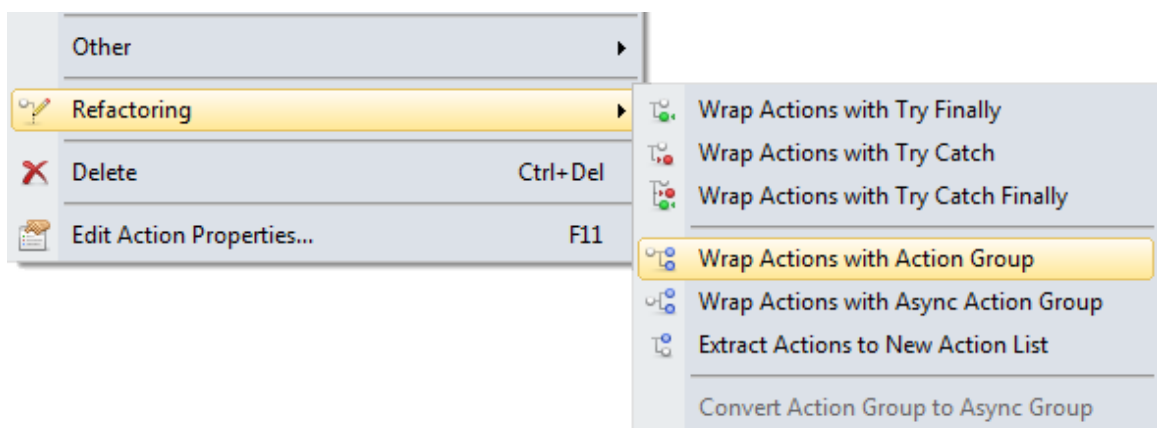
- Action groups
- Comment actions
- Action comments
- Variable comments
- Project information notes

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.



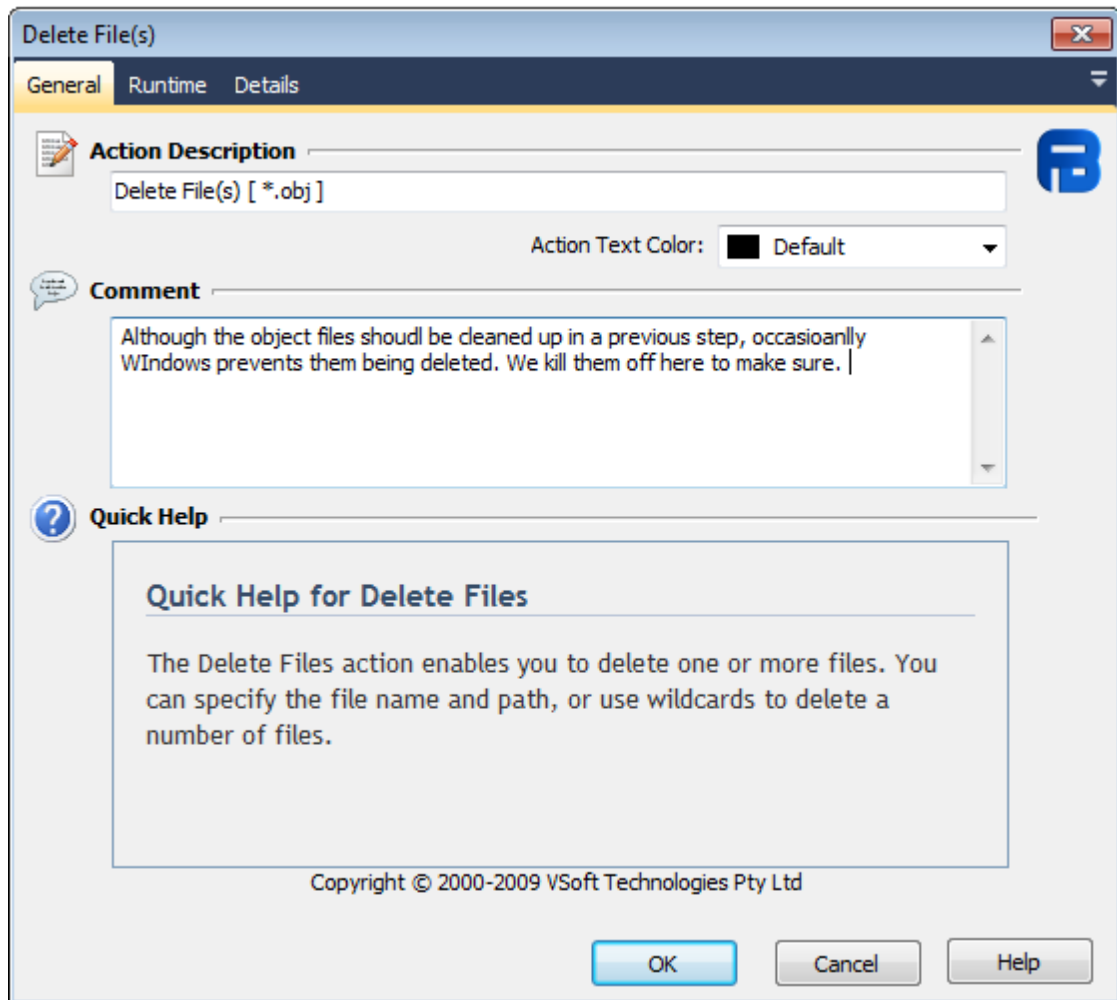
To quickly create an action group, select a group of actions, right click, and choose "Refactoring > Wrap Actions with Action Group". Then press F2 to give the action group a meaningful name.



Comment actions are useful for providing further information about a part of your build. They draw the maintainer's attention to some important information about a part of the build. 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.



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.



Variables

While system variables come with predefined comments, it's up to you 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.

Edit Variable

Variable Name :

Group Name :

Specify a group hierarchy in the form Group.SubGroup

BasicType

☒ Variant ☐ Integer ☐ DateTime

☐ String ☐ Float ☐ Boolean

Format String :

Default Value :

☐ Make Available as Environment Variable

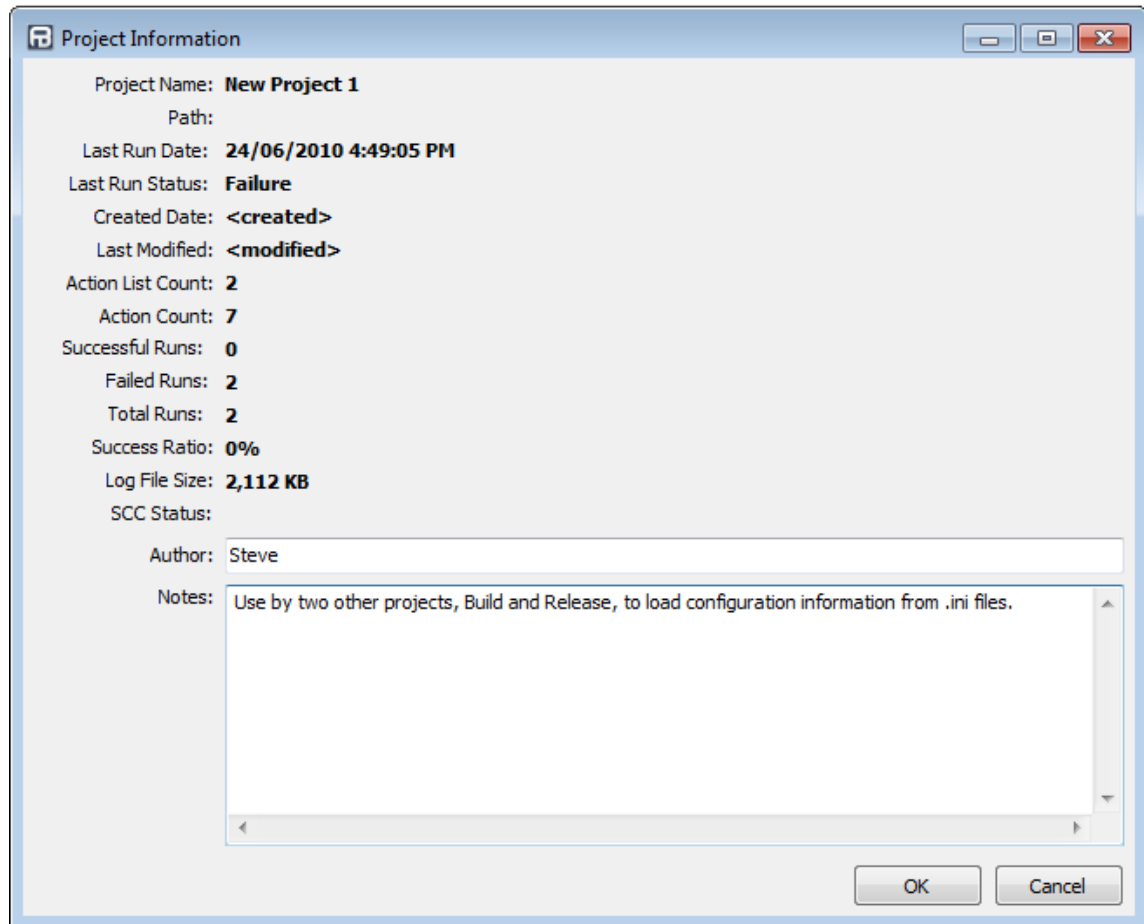
☐ Macro (expand variable references on each access)

☐ Persistent

Comment :

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.



9.6 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 build. There are several advantages to doing this:

- The build doesn't abort at the first error, so you get more information if several projects fail to compile.
- You can recover from minor errors
- You can treat different errors differently, for example, by emailing different responsible people.
- You can record your own statistics and logs.

Here's a simple example which builds all the projects listed in a text file. The variable ErrorCount records the total number of errors, while ErrorProjects builds a list of the projects which have failed.

Description	Enabled	Ignore Failure	Status
Set Variable ErrorCount to [0]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
Set Variable ErrorProjects to []	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
File Contents Iterator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Iteration 1 of 3
Try	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
Build VS.Net Solution [c:\builds\%Project%.sln]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Error
Catch	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
Set Variable [Increment variable ErrorCount]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Paused
Append To Variable [ErrorProjects]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Paused
End	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
If [%ErrorCount%] > [0]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
MessageBox %ErrorCount% errors.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
Stop Run [Failure]	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
Else	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed
MessageBox Build succeeded.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Completed

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 Build VS.Net Solution action. If the compilation succeeds, the Catch part is not run.
4. If the compilation fails, the Catch part is run: the ErrorCount variable is incremented, and the ErrorProjects variable is appended to. The build then continues on the next loop of the iterator.
5. After all the projects are built, a message is shown if there was at least one error. We then use a Stop Run action to signal that the build as a whole has failed.
6. If there was no error, a different message is shown. By default, builds 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 projects 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:

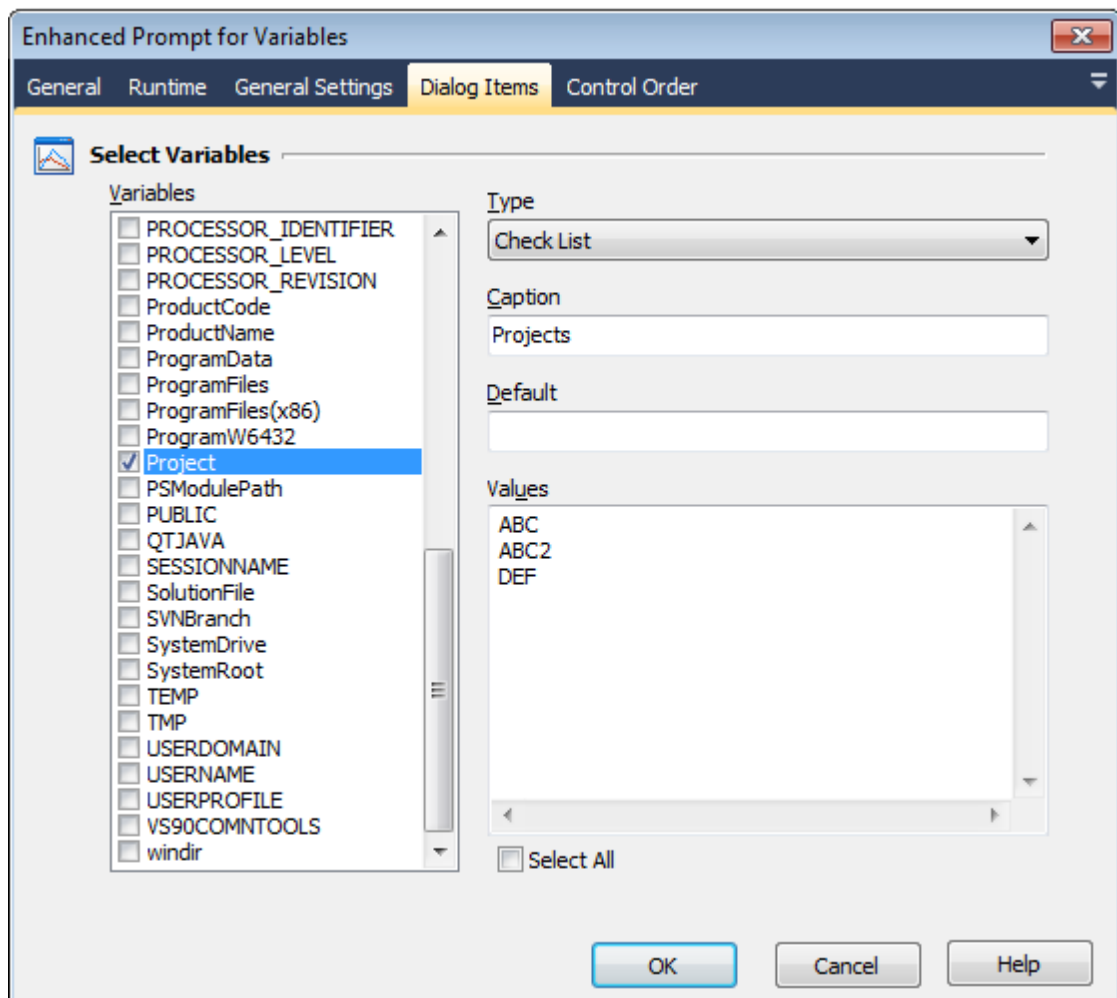
Properties	
Behaviour	
Enabled	<input checked="" type="checkbox"/>
Ignore Failure	<input type="checkbox"/>
Pause Interval	0
Retry Attempts	0
Retry Pause Interv	1000
Delete Files	
Delete hidden files	False
Delete read only files	False
Fail if no file	True
File specification	Delete File(s) [*.obj]
Description	
Comment	Although the object files should be cleaned up
Description	Delete File(s) [*.obj]
Identity	
Action Name	Delete File(s)
Package	C:\Program Files (x86)\FinalBuilder 7\FBFile.b
Logging	
Action Log Title	Failing due to %errorcount% errors. These p
Expand Action Log	<input checked="" type="checkbox"/>
Hide Action From Log	<input type="checkbox"/>
Log Action Properties	<input type="checkbox"/>
Log To Variable	
Action Log Title Sets a different title for the action for logging purposes. This title is only ever used in the Log. Clear this property to revert to the Description. Scripting: Action.ActionLogTitle	

Projects Actions Properties Run

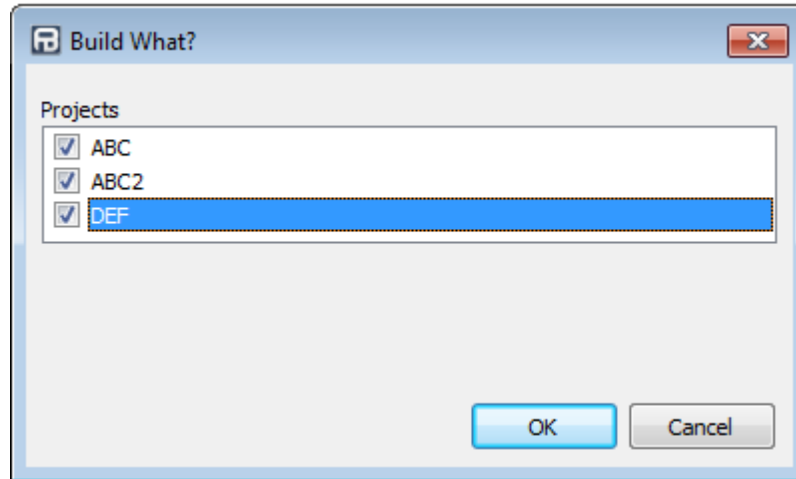
9.7 Running Builds in Sequence

After you set up a A configurable build that can build any one of the products in your suite, the logical next step is to build several or all of them at once, in order. This tutorial shows the general approach.

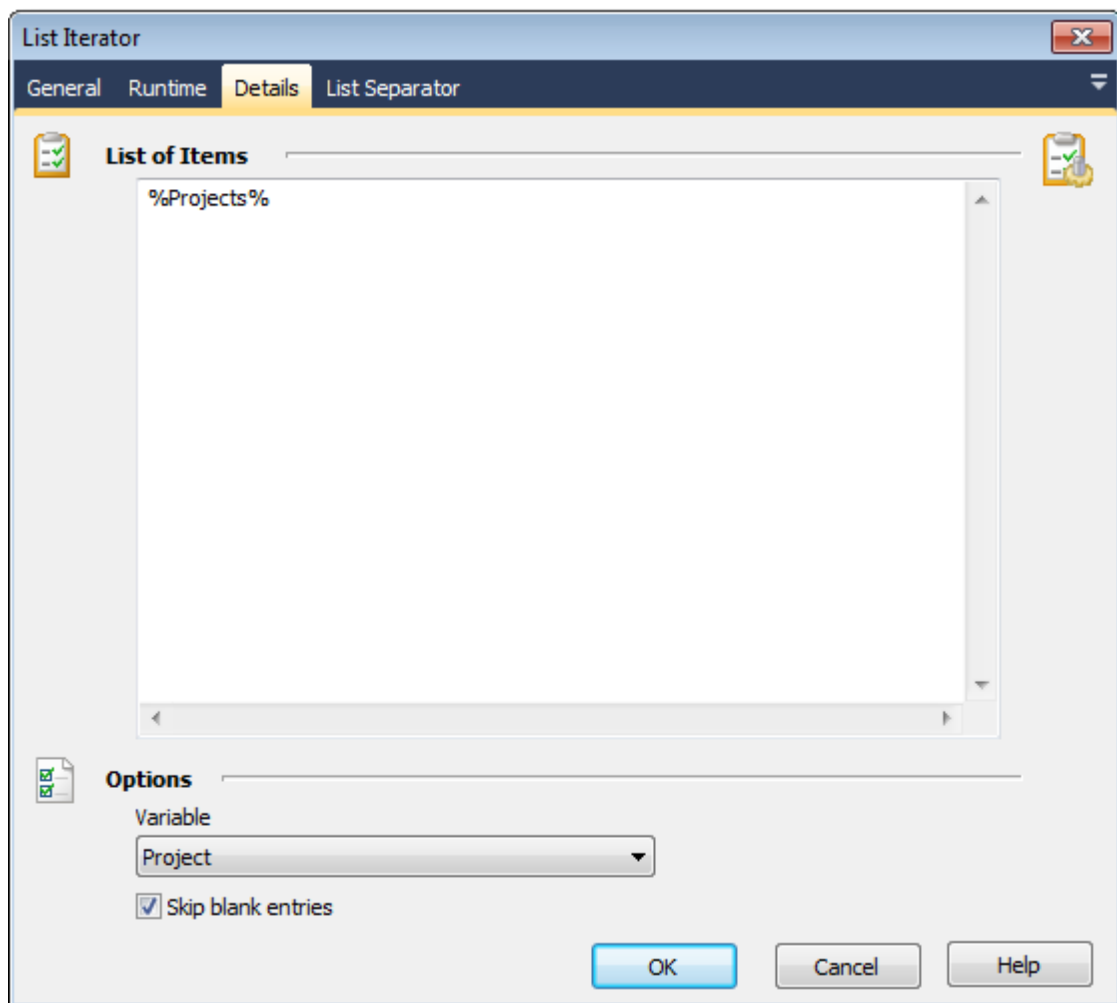
First, create a new project called something like "MultiBuild". You want to collect input from the user about which builds to run, then iterate over the answer.



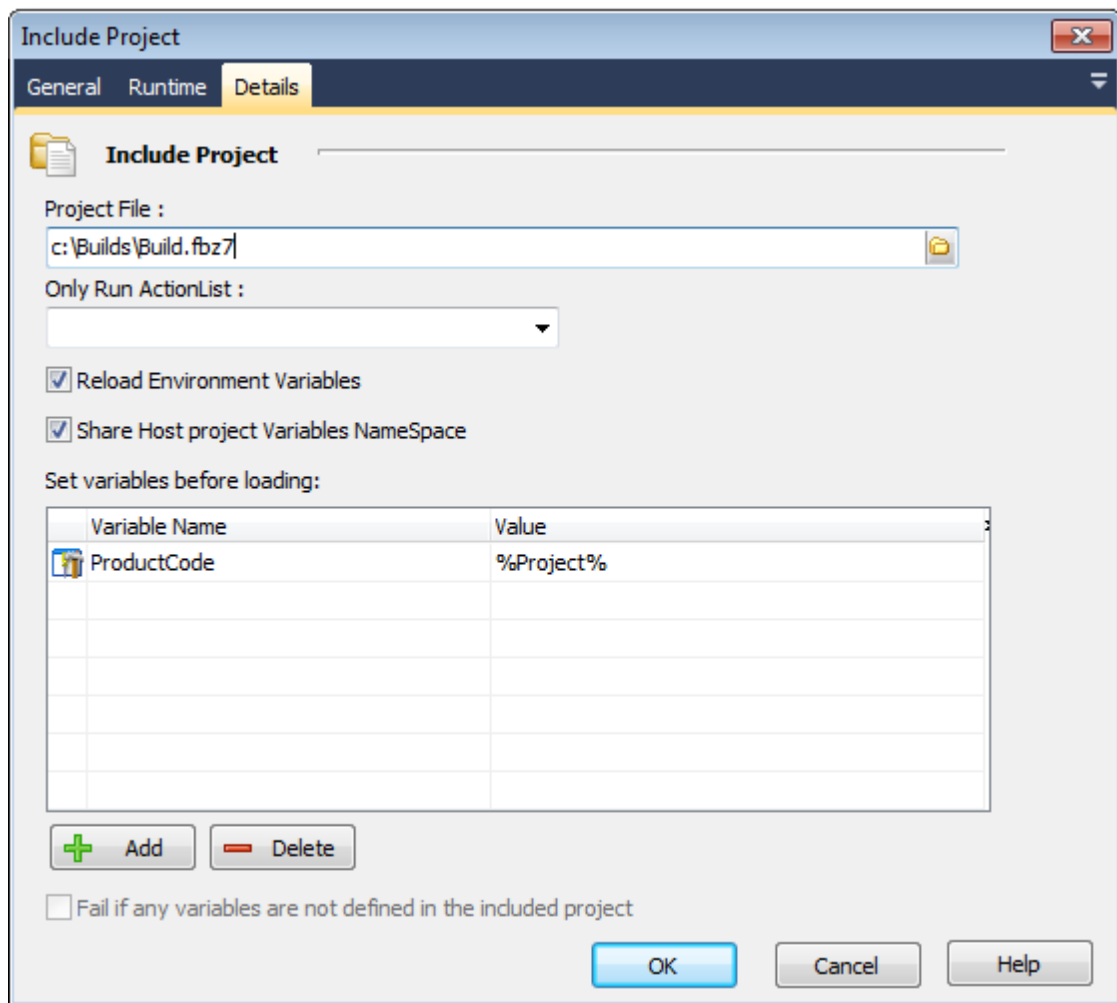
We use an Enhanced Prompt for Variables with type Check List here. At runtime this appears as follows:



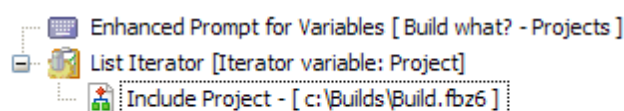
The results of the dialog are saved to the Projects variable, separated by a carriage return/linefeed sequence. That means you can iterate over them with a List Iterator.



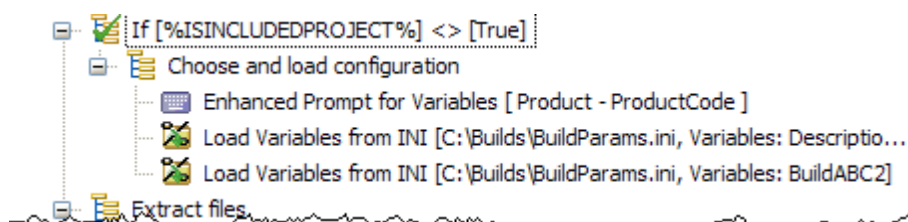
Inside the list iterator, include the project that actually performs the build, passing the build code as parameter:



The outline of your project now looks like this:



The final step is to modify your build project (C:\Builds\Build.fbz7 in this example) to accept incoming parameters. You can use the Application Variable `ISINCLUDEDPROJECT` to determine whether the project should ask the user for input or not.



Now, if the build project is run from the multibuild project, it won't ask the user which product to build - the correct value is already in the ProductCode variable.

9.8 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:\builds\source directory. This DOS command lists the files in order from oldest to newest: `dir /b /od /a-d c:\builds\source*.*`

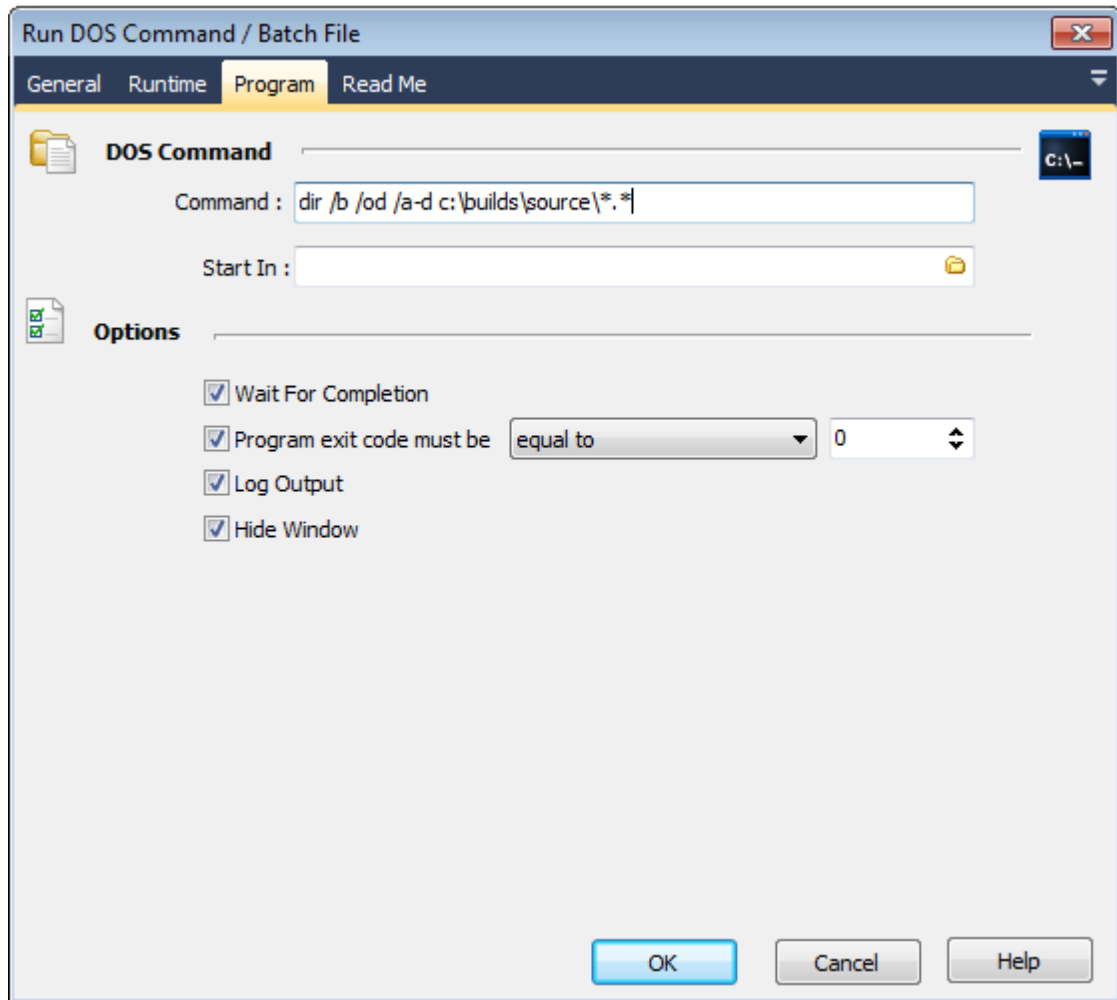
The output is as follows:

```
LOGOSTRS.PAS
LOGOMAIN.PAS
ABOUT.PAS
hello.pas
BBXAbt.pas
bbxmain.pas
ttree.pas
logging.pas
servtest.pas
```

We thus simply want to retrieve the last line of that output. An Output Monitor do this.

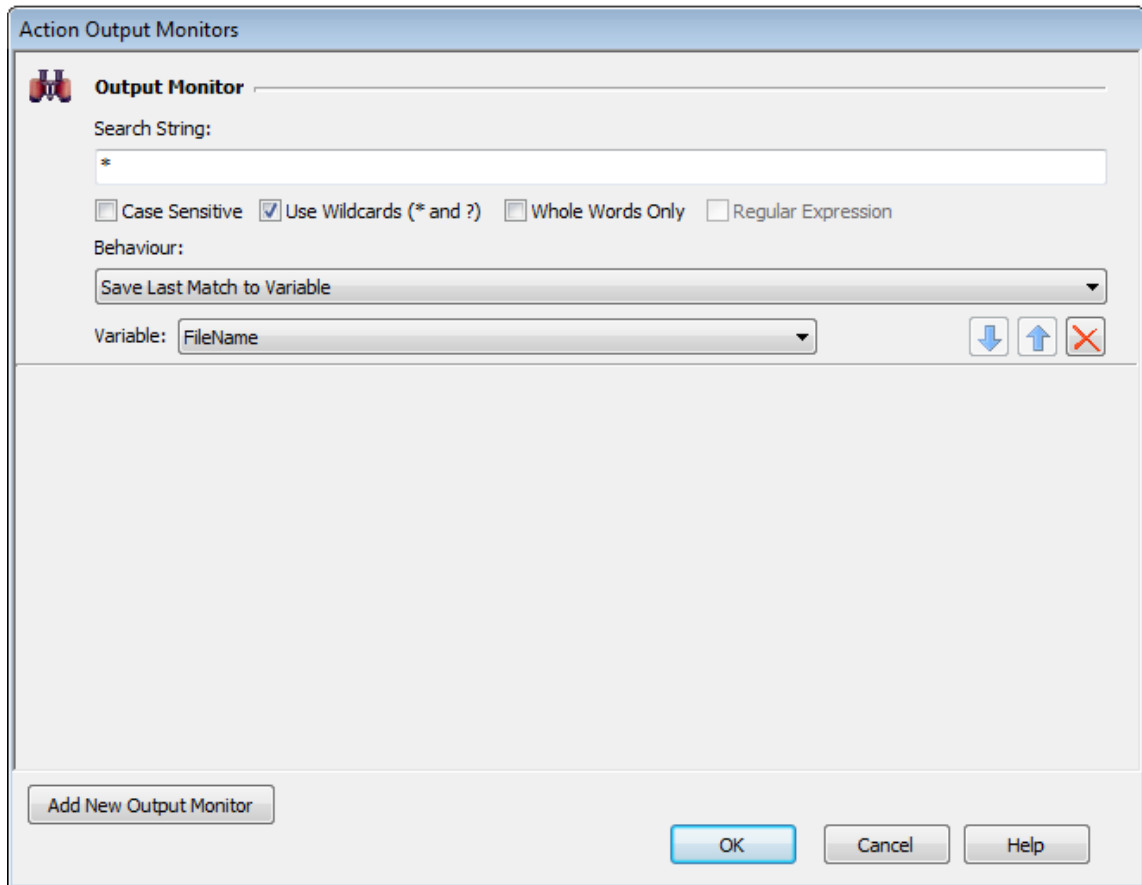
First, create a variable to hold the name of the most recent file. Call it "filename".

Next, create a Run DOS Command / Batch File action.



On the **Runtime** tab, click the **Output Monitors** ellipsis button (...).

Click "Add a New Output Monitor" then set the options as shown:

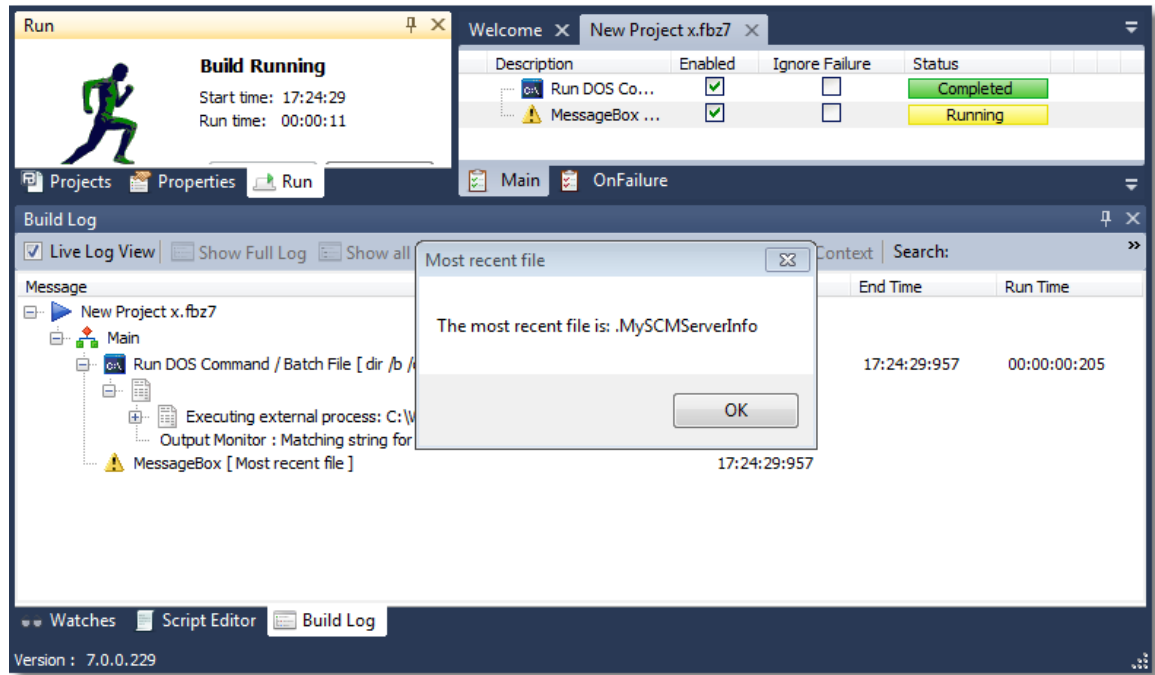


The **Search String** defines what lines of text are selected. In this case, we don't care what the text is - we just want the last one. So the *, combined with **Use Wildcards** matches any text.

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:



10 Reference

10.1 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 FinalBuilder: 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', 'foobalkjdfllkj9r' and 'foobr'

 foob.+r matches strings like 'foobar', 'foobalkjdfllkj9r' 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', 'foobaaaaar' etc.
 fooba{2,3}r matches strings like 'foobaar', or 'foobaaar' but not 'foobaaaaar'

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 'bbbbb', '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.

10.2 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

11 Support

11.1 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 build 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.)

Embarcadero Delphi Compiler Action

Problem: Compiling Projects with UNC Paths in the project file property will result in the resource file not being linked into the executable. This is appears to be a bug in the Delphi command line compiler (not yet confirmed).

Workaround : Build locally, not on a network drive, or use the Map Network Drive action to create network file paths without UNC pathnames.

Wise InstallMaster/InstallBuilder Actions

Problem: The Wise compiler (versions earlier than 9.02) do not return a non zero return code when an error occurs during compilation. This means that FinalBuilder has no way of detecting if the compile of the installer failed or not. Usually Wise displays a message box when an error occurred.

Workaround : Stop the build manually after the message box is displayed.

Problem: The Wise compiler will hang if you attempt to run it as part of a scheduled build. This is a known problem with wise, not with FinalBuilder! The issue is that the Wise Compiler will display a message box or dialog and of course there is no way to respond to it while it is being run unattended.

Workaround : Enable the "Monitor for Error Dialogs" option (only available on the Properties view for an action). This option will close the dialog that is shown and fail the action. This problem may be solved with Wise 9.02, however we have not confirmed this.

Embarcadero C++ Builder Action

Problem: Some static Lib projects will not link correctly when compiled from FinalBuilder. You might see something like this :

```
J:\Embarcadero\CBUILD~1\bin\..\BIN\TLIB /u debug\jpegD.lib @MAKE0000.@@@  
DOS-reported error: Bad file number  
TLIB 4.5 Copyright (c) 1987, 1999 Inprise Corporation  
opening 'MAKE0000.@@@'
```

** error 1 ** deleting debug\jpegD.lib
MAKE failed, returned : 1

Workaround : In some cases (where the "Bad file number" error is seen) it may be possible to work around this by specifying -tDEFLIB.BMK in the BPR2MAKE Options field, and Turning off the "Capture Make Output" option.

Problem: Some projects (especially those migrated from an earlier version of BCB) may not compile in FinalBuilder, even though they compile fine in the BCB IDE. Typical symptoms include Unresolved External symbols errors.

Workaround : This work around works in some cases, where the package/libraries/sparelibs options in the bpr/bpk file are not correct. Open the project in the BCB IDE, and change the Use Packages option and save the project. Then change the Use Packages option back to what it was before, and save the project. Close BCB and attempt to build the project from FinalBuilder

Problem: Make fails with the following error :

Error E2266: No file names given
MAKE failed, returned : 1

The most common cause of this a space in the intermediate output path. Make does not like this, even the mak files generated from the IDE will fail to build correctly.

Workaround : The only way around this is to not use intermediate output paths with spaces in them.

11.2 FAQ

Q. I'm getting this error when running my Delphi action : Error expanding variables in xxx : Variable : DELPHI - does not exist!

A. You are probably using the DELPHI variable outside of the Library Path or SearchPath settings. For example if you have your library path set to %MYLIBRARYPATH% and the default value of %MYLIBRARYPATH% is \$(DELPHI)\lib, then you will see this error. The DELPHI variable is a special variable, at runtime the Delphi action replaces the variable with a compiler version specific variable before the variable expansion is called (eg. DELPHI5DIR for Delphi 5). If you look in the variables dialog you will not find a DELPHI variable declared.

Q. Why is no executable is being produced when compiling my Delphi 5 project with FinalBuilder

A. This is a bug with the Delphi 5 command line compiler (dcc32.exe), it is fixed in Delphi 6 & 7. To work around this problem, select the "Work around Delphi 5 Compiler bug Check box".

Q. Personal Firewalls & FinalBuilder

I run ZoneAlarm Pro as a firewall, and every time I update FinalBuilder, it warns that (the changed) FinalBuilder is attempting to access the internet, and that it is attempting to obtain server access (i.e., it listens for connections).

A. FinalBuilder checks & listens for other copies of FinalBuilder running on the local network with the same license key. If it detects another copy, and for example the license key is a single user license key, then the second copy will fail to start. For instance, if you have a 2 user license key then the third copy will fail to start.

11.3 FinalBuilder Support

VSoft Technologies provide support for FinalBuilder™ on our forums, and via email.

Web forum interface : <http://www.FinalBuilder.com/forums.aspx>

Support Email : support@FinalBuilder.com

Index

- . -

- .Net 3rd Party Tools
 - Demeanor Action 777
 - Dotfuscator Action 777
 - FxCop Action 778
 - ILMerge 798
 - NCover 780
 - NCoverExplorer 783
 - XenoCode Action 778
- .Net Framework Tools 748
 - NGen Install 748
 - NGen Uninstall 750
 - Ngen Update 752
 - Precompile ASP.NET 2.0 753
 - Register Assembly in COM 756
 - Run AL.EXE Action 757
 - Run ASPNET_REGIIS.EXE Action 758
 - Serviced Components Installation Tool 759
 - VS.NET Setup & Deployment Project Updater 800
- .Net SDK Tools 762
 - Extract Public Key Action 762
 - GAC Download Cache Action 763
 - GAC Install Action 763
 - GAC Uninstall Action 764
 - Generate Key Pair Action 766
 - Install Key in Container Action 766
 - Manifest Generation and Editing 767
 - Re-sign Assembly Action 771
 - Run ResGen.exe Action 771
 - Run SN.EXE Action 772
 - Type Library Export Action 773
 - Type Library Import Action 774
 - Verify Strong Name Action 775
- .Net Solution Iterator 244

- A -

- Accessing Property Sets 816
- AccuRev
 - Add 913
 - Add a Change Package Entry 914
 - Change User 915
 - Check Out 916
 - Create Snapshot 917
 - Create Stream 918
 - Defunct 919
 - Generic 920
 - Keep 921
 - Lock 922
 - Promote 923
 - Purge 924
 - Remove 925
 - Remove a Change Package Entry 926
 - Unlock 927
 - Update 928
- Action Behaviour 134
- Action Group Action 374
- Action Inspector 17
- Action List
 - Columns 14
 - Exit Action List 379
 - Run Action List 390
 - Working with Actions 14
- Action List Parameters 121
- Action Lists 7, 8, 18
- Action Logging Properties 138
- Action Object 100
- Action output monitors 139, 1475
- Action Properties 100, 134
 - Logging 138
 - Output Monitors 139
 - Processor Affinity and Process Priority 145
 - Run Action as User 143
 - Timing 137
- Action Types 8, 20
- Action Usage Information 67
- ActionComment Property 100
- ActionLogTitle Property 100
- ActionName Property 100
- ActionPackage 721
- Actions 7
 - Adding Actions 14
 - Copying Actions 14
 - Deleting Actions 14
 - Moving Actions 14
 - Selecting Actions 14
- Actions Summary 130
- Actions Tab 20
- ActionStudio 133
- Active Script 86
- Add Action List 18
- Add File 1065
- Add Scheduled Task 1368
- Adding & Editing Variables 109

- Adding Actions 14
- ADO Dataset Iterator 267
- ADO Execute SQL Action 269
- ADO Execute Stored Procedure Action 270
- AfterAction Event 98
- Agile Builds 2
- AL.EXE 757
- Altova DiffDog 1383
- Ant Project Action 177
- Application Variables 118
- AQTest Action 878
- Archiving 163, 166, 167, 169, 170, 173, 174
 - 7Zip 149
 - Create Archive 149
 - Create Zip File Action 156
 - Delete from Archive 153
 - Extract Archive 153
 - Extract Zip File Action 160
 - List Archive 154
 - Test Archive 155
 - Update Archive 155
 - WinRAR Action 176
- Armadillo 713
- Ask Question Action 505
- ASP.NET Compiler 753
- ASPack 272
- ASPNET_REGIIS 758
- ASProtect 714
- Assembly Info Updater Action 192
- Assembly Linker 757
- Assign Properites of Property Set 819
- Async Action Group 391
- Authenticode Action 299
- AutomatedQA Test Complete 4 Actions 881
- AutomatedQA Test Complete Actions 880
- Automatic Usage Submission 67
- Automating FinalBuilder 1432
 - Command Line Interface 1438
 - Console Build Tool 1440
 - Console Exit Codes 1443
 - IDE Exit Codes 1439
 - MSBuild Task 1444
 - Scheduling Builds 1433
- Automating GUI Applications 421
- Auxillary project files 77

- B -

- Backup Database Action 258
- Backup IIS5 Metabase Action 652

- Backup IIS6 Metabase Action 666
- Backup Registry Key 451
- Basic Concepts 7
- Beep Action 507
- BeforeAction Event 98
- BMP Image Manipulation 725
- BPG File Iterator 213
- BR 856
- Build Delphi Project Action 201
- Build History 35
- Build Log 8, 47
- Build Statistics 37
 - Chart Data Options 39
 - Chart Types 38
 - Chart Visual Options 40, 55
 - Exporting Build Statistics 41
 - Printing Statistics Charts 43
 - Success Ratio Chart 37
- Build Status Message 282
- Build Tools
 - Ant Project Action 177
 - MSBuild Project Action 180
 - NAnt Project Action 183
- Build VS.NET Solution 232
- Builds by Day 38
- Builds by Day of Week 38
- Builds by Month 38
- Builds by Time of Day 38
- Builds by Week 38
- Burn CD Action 185
- Burn ISO Action 188

- C -

- C# Builder Compiler Action 208
- C# Compiler Action 246
- C# Project Compiler Action 247
- C++ Builder Compiler Action 194
- Calculate File CRC32 Action 301
- Calculate File MD5 Action 303
- Calculate Size of Files 332
- Carriage Return 856
- Case Action 376
- Case Statement
 - Case Action 376
 - Switch Action 396
- Catch Action 397
- Category 20
- CD/DVD Burning
 - Burn CD Action 185

- CD/DVD Burning
 - Burn ISO Action 188
 - Check Ready Action 189
 - Create ISO Action 190
 - Erase CDRW Action 191
- Change String Case 858
- Chart Data Options 39
- Chart Types 38
- Chart Visual Options 40, 55
- Check Catalogue Action 261
- Check Database Action 261
- Check File Exists Action 305
- Check for Updates 69
- Check If Host Exists 804
- Check Ready Action 189
- Check Spelling 738
- Check Virtual Machine Status 1294
- Check VM Status (Virtual Server) 1281
- ChildActionCount Property 100
- ChildActions Function 100
- Choose One Action 508
- Chrome Compiler Action 218
- CityDesk Action 719
- Clear Build History 35
- ClickOnce
 - Manifest Generation and Editing 767
- Close Process Action 1308
- CodeHealer 832
- COM Class Registered 1331
- COM+ Delete 1310
- COM+ Edit Application 1311
- COM+ Register 1315
- COM+ Shutdown Application 1317
- COM+ Start Application 1318
- Command Line Interface
 - Console Build Tool 1440
 - Console Build Tool Exit Codes 1443
 - IDE Command Line Arguments 1438
- Comment Action 720
- Compile Delphi for .Net Project Action 208
- Compile Embarcadero Resource Script Action 215
- Compile Visual Basic Project 223
- Compilers 201
 - Chrome Action 218
 - Embarcadero C# Compiler Action 208
 - Embarcadero C++ Builder 194
 - Embarcadero Delphi for .Net 208
 - Embarcadero Resource 215
 - IncrediBuild Actions 219
 - Java 220
 - Microsoft C# Compiler Action 246
 - Microsoft C# Project Compiler Action 247
 - Microsoft J# Compiler Action 251
 - Microsoft J# Project Compiler Action 252
 - Microsoft VB.NET Compiler Action 249
 - Microsoft VB.NET Project Compiler Action 249
 - Visual Basic 223
 - Visual C++ 6 228
 - Visual Studio .NET 232
 - VisualStudio .Net Solution Iterator 244
- Concatenate Files Action 307
- Concatenation of Strings 860
- Continuous Integration 2
- Control Service Action 1320
- Convert file to PDF 807
- Convert HTML to MHT File 524
- Convert Images 725
- Convert PDF to another format 806
- Copy Files Action 308
- Copy/Move File List Action 310
- Copy/Move FileSet 293
- Copying Actions 14
- Create ActionStudio Documentation Action 284
- Create Archive 149
- Create Branch 1065
- Create Changelog 1066
- Create Directory Action 312
- Create IIS5 Virtual Directory Action 653
- Create IIS5 Web Site Action 656
- Create IIS6 Application Pool 668
- Create IIS6 FTP Directory Action 681
- Create IIS6 FTP Site Action 684
- Create IIS6 Virtual Directory Action 669
- Create IIS6 Web Site Action 671
- Create ISO Action 190
- Create Label 1067
- Create Shortcut 1322
- Create Snapshot of Virtual Machine 1295
- Create Text File Action 313
- Create XML Node 1384
- Create Zip File Action 156
- Creating Custom Actions 133
- CRLF 856
- CSV Iterator
 - Comma 852
 - Delimited 852
- Cursor Position 429
- Custom Actions 133
- CVS Actions 963
- CVS Command 964

- D -

Database

ADO Dataset Iterator	267
ADO Execute SQL Action	269
ADO Execute Stored Procedure Action	270
SQL Server Backup Database Action	258
SQL Server Best Practices Analyzer	259
SQL Server Check Catalogue Action	261
SQL Server Check Database Action	261
SQL Server DTSRun Action	255
SQL Server Execute SQL Action	256
SQL Server Rebuild Indexes Action	262
SQL Server Remove Unused Space Action	263
SQL Server Restore Database Action	264
SQL Server Set Access Mode Action	265
SQL Server Update DB Statistics Action	266

Date of File/Folder 333

DateTime 94

DateTime Action 723

Debugger (Script) Options 90

Debugging 33

Decrypt String 862

Define Property Set 817

Define XML Document Object 1401

Delay Action 377

Delete Branch 1067

Delete Changelist 1068

Delete Directory Action 314, 316

Delete Files Action 316

Delete from Archive 153

Delete IIS5 Virtual Directory Action 659

Delete IIS5 Web Site Action 660

Delete IIS6 Application Pool 674

Delete IIS6 FTP Directory Action 689

Delete IIS6 FTP Site Action 690

Delete IIS6 Virtual Directory Action 675

Delete IIS6 Web Site Action 676

Delete Label 1068

Delete Log Entry 35

Delete Scheduled Task 1371

Delete XML Element/Attribute 1388

Delete XML Nodes 1390

Deleting Actions 14

Delphi BPG Iterator 213

Delphi Compiler - EurekaLog 207

Delphi Compiler Action 201

Delphi for .Net Compiler Action 208

Delphi Options 205

Demeanor Action 777

Depend on a file 326

Deployment Project 800

Description Property 100

Design Time Options 52

DiffDog 1383

Disk Free Space 1333

Doc-O-Matic 439

DOS Command Action 1344

Dotfuscator Action 777

Double Quotes 866

Drag & Dropping Actions 14

Drive Free Space 1333

DTSRun Action 255

DTSRun Location 261

DVD Burning 185

DVD Check Ready 189

- E -

EC2

EC2 Attach Volume	639
EC2 Change Instance States	629
EC2 Create Image	635
EC2 Create Snapshot	645
EC2 Create Volume	641
EC2 Delete Snapshot	646
EC2 Delete Volume	642
EC2 Deregister Image	636
EC2 Detach Volume	640
EC2 Get Availability Zones	647
EC2 Get Image Properties	633
EC2 Get Instance Properties	628
EC2 Get Snapshot Properties	643
EC2 Get Volume Properties	637
EC2 Run Instances	630
EC2 Terminate Instance	632

Echo Procedure 100

Edit Field 148

Edit Password 81

Edit XML File Action 1391

Editor (Script) Options 88

Else Action 378

Email Action 549

Embarcadero C# Builder Compiler Action 208

Embarcadero C++ Builder Action 194

Embarcadero Delphi 201

Embarcadero Java Compiler 222

Embarcadero Project Group 1410

Embarcadero Resource Compiler Action 215

- Encrypt ActionPackage 721
 - Encrypt String 862
 - End Action 397
 - End Running Scheduled Task 1372
 - Environment Variables 117
 - Erase CDRW Action 191
 - Error Codes
 - Console Error Codes 1443
 - IDE Error Codes 1439
 - Error Handling 32
 - Errors 32
 - Escaping Variable References 114
 - Estimated Progress 24, 281
 - EurekaLog 207
 - Event 98
 - Exceptions 32
 - Executable Packers
 - ASPack 272
 - UPX Pack 274
 - Execute Condition 98, 102
 - Execute Program Action 1327
 - Execute SQL Action 256
 - Exit Action List Action 379
 - Exit Codes
 - Console Exit Codes 1443
 - IDE Exit Codes 1439
 - ExpandExpression Function 100
 - Export Log 35, 278
 - Export Log Fragment 278
 - Export Registry Key 451
 - Exporting Build Statistics 41
 - Extension of Files 854
 - Extract Archive 153
 - Extract File Version Action 318
 - Extract Public Key Action 762
 - Extract String 874
 - Extract XML Fragment Action 1394
 - Extract Zip File Action 160
- F -**
- FAQ 1506
 - FBCmd 1440
 - FBCmd MSBuild Task 1444
 - Field Editor 148
 - File Contents Iterator Action 706
 - File Dependency 326
 - File Iterator Action 707
 - File Size Action 332
 - File Types 77
 - File/Folder Date 333
 - Filename Manipulation 854
 - Filepath Manipulation 854
 - Files & Directories 356, 359, 360
 - Authenticode Action 299
 - Calculate File CRC32 Action 301
 - Calculate File MD5 Action 303
 - Check File Exists Action 305
 - Concatenate Files Action 307
 - Copy Files Action 308
 - Copy/Move File List Action 310
 - Create Directory Action 312
 - Create Text File Action 313
 - Delete Directory Action 314
 - Delete Files Action 316
 - Extract File Version Action 318
 - File Dependency 326
 - Find File(s) 330
 - Get File Size 332
 - Get File/Folder Date 333
 - Move Directory Action 335
 - Move Files Action 336
 - Read Text File Action 338
 - Rename File or Directory Action 340
 - RoboCopy Actions 341
 - Set File Attributes Action 355
 - Touch Files Action 365
 - Write Text File Action 367
 - XCOPY Action 368
 - FileSet Copy/Move 293
 - FileSet Define 288
 - FileSet Delete Files 295
 - FileSet Filters 291
 - FileSet Log Files 296
 - FileSet Patterns 289
 - FileSet Refresh 297
 - FileSet to Variable 298
 - FileSets 286
 - Copy/Move FileSet 293
 - FileSet Accessing 286
 - FileSet Define 288
 - FileSet Delete Files 295
 - FileSet Filters 291
 - FileSet Log Files 296
 - FileSet Refresh 297
 - FileSet Specifying Patterns 289
 - FileSet to Variable 298
 - Filter 20
 - Filter Log By Action 47, 48
 - FinalBuilder Server 2

FinalBuilder System Variables 118
 Finally Action 397
 Find / Replace
 In IDE 30
 Text Find / Replace Action 741
 Fix TLBImp Project References 797
 Flow Control 373
 Async Action Group 391
 Case Action 376
 Catch Action 397
 Conditional Execution 102
 Delay Action 377
 Else Action 378
 End Action 397
 Exit Action List 379
 Finally Action 397
 If .. Then Action 382
 If Prev Action Failed Action 386
 Include FinalBuilder Project 387
 Iterators 704
 Parallelization 391
 Raise Exception Action 389
 Run Action List 390
 Stop Run Action 395
 Switch Action 396
 Try Action 397
 Waiting For Events 400
 While Loop 420
 Folder Iterator Action 709
 Folder name Manipulation 854
 Format DateTime 94
 Free Space of Disk 1333
 FTP Actions 525
 FTP Client 541
 FTP Mirror
 FTP 528
 Mirror 528
 Sync 528
 FTP Upload 535
 FTPS Change Directory
 Change Directory 604
 FTPS 604
 FTPS Change Up Directory
 Change Up Directory 605
 FTPS 605
 FTPS Check If Connected
 Check If Connected 606
 FTPS 606
 FTPS Connect
 Connect 607
 FTPS 607

FTPS Create Directory
 Create Directory 611
 FTPS 611
 FTPS Delete File
 Delete File 612
 FTPS 612
 FTPS Disconnect
 Disconnect 613
 FTPS 613
 FTPS Download Files
 Download Files 615
 FTPS 615
 FTPS Download File
 Download File 614
 FTPS 614
 FTPS Get File Date
 FTPS 616
 Get File Date 616
 FTPS Get File Size
 FTPS 617
 Get File Size 617
 FTPS Get Remote Checksum
 FTPS 618
 Get Remote Checksum 618
 FTPS Get Remote Directory
 FTPS 619
 Get Remote Directory 619
 FTPS List Directory
 FTPS 620
 List Directory 620
 FTPS No-op
 FTPS 622
 No-op 622
 FTPS Options 604
 FTPS Remove Directory
 FTPS 623
 Remove Directory 623
 FTPS Rename File
 FTPS 624
 Rename File 624
 FTPS Send Command
 FTPS 625
 Send Command 625
 FTPS Upload Files
 FTPS 626
 Upload Files 626
 FxCop Action 778

- G -

GAC Download Cache Action 763

GAC Install Action 763
 GAC Uninstall Action 764
 Generate Key Pair Action 766
 Generate New GUID 1330
 Generate Random Number 722
 Get DateTime Action 723
 Get Disk Free Space Action 1333
 Get File Size Action 332
 Get File/Folder Date 333
 Getting Started 26
 Basic Concepts 7
 IDE Modes 8
 GIF Image Manipulation 725
 Git
 Add Files 967
 Checkout 970
 Clone Repository 972
 Commit 974
 Create Branch 978
 Create Repository 980
 Delete Branch 983
 Fetch 984
 Generic 987
 List Branches 988
 Merge 990
 Move 993
 Options 965
 Pull 995
 Push 998
 Remove Files 1001
 Rename 1003
 Repository Status 1005
 Reset 1006
 Show Commit Logs 1009
 Global Script 79
 GPInstall Action 458
 Graph 37
 GUI Automation 421
 Perform Mouse Click 428
 Perform Mouse Move 429
 Send Keyboard Input 430
 Set Scrollbar Position 433
 Specifying target controls 422, 423
 Wait for Window or Control 435
 GUID (Generate new) 1330

- H -

Help & Manual 441
 Help Compilers
 Doc-O-Matic 439

Help & Manual 441
 HTML Help 443
 NDoc 444
 Sandcastle - BuildAssembler 445
 Sandcastle - MRefBuilder 446
 Sandcastle - XslTransform 448
 WinHelp 449
 HTML Help 443
 HTML to MHT File 524
 HTTP Get Action 545
 Hyper V 1267
 Apply VM Snapshot 1276
 Check VM State 1270
 Create VM Snapshot 1275
 Delete VM Snapshot 1277
 List Virtual Machines 1278
 Pause VM 1273
 Reset VM 1272
 Server 1268
 Start VM 1270
 Stop VM 1271
 Suspend VM 1274
 Hyper V Options 1267

- I -

ICQ Action 555
 IDE Debug Mode 33
 Watches 34
 IDE Design Mode 13
 Action Lists 18
 Actions Tab 20
 Project Tree 22
 Properties Tab 17
 Quick Help 26
 Script Editor 27
 Working with Actions 14
 IDE Modes 8
 IDE Running Mode 44
 Build History 35
 Estimated Progress 24
 Run History 35
 Run Log 47
 Running a project 45
 Validation 50
 If .. Then Action 382
 If COM Class Registered 1331
 If Prev Action Failed Action 386
 Ignore Error 32
 IgnoreFailure Property 100
 IIS 649

- IIS5 650
 - Backup IIS Metabase Action 652
 - Create Virtual Directory Action 653
 - Create Web Site Action 656
 - Delete Virtual Directory Action 659
 - Delete Web Site Action 660
 - Restore IIS Metabase Action 660
 - Start/Stop/Pause Website Actions 661
 - Start/Stop/Restart WWW Service Actions 661
 - Unload IIS Application Action 661
- IIS6 663
 - Backup IIS Metabase Action 666
 - Create Application Pool 668
 - Create Virtual Directory Action 669
 - Create Web Site Action 671
 - Delete Application Pool 674
 - Delete Virtual Directory Action 675
 - Delete Web Site Action 676
 - Restore IIS Metabase Action 676
 - Start/Stop/Pause Website Actions 677
 - Start/Stop/Restart WWW Service Actions 677
 - Unload IIS Application Action 677
- IIS6 FTP 679
 - Create FTP Directory Action 681
 - Create FTP Site Action 684
 - Delete FTP Directory Action 689
 - Delete FTP Site Action 690
 - Start/Stop/Pause FTP Site Actions 690
 - Start/Stop/Restart FTP Service Actions 690
- ILASM 253
- ILMerge 798
- Image Manipulation 725
- Images
 - Convert, Resize, Rotate 725
 - EXIF 729
 - Lossless Transform 729
 - Rotate 729
- Include FinalBuilder Project Action 387
- IncrediBuild Actions 219
- Increment Value of Property Set 820
- INI File Iterator Action 710
- Ini Files
 - Load Variables to INI Action 899
 - Read 452
 - Save Variables to INI Action 907
 - Write 457
- INI Files Reference 1503
- InnoSetup Action 459
- Input Box 510
- Insert Text in PDF file 808
- Install Application 493
- Install Key in Container Action 766
- InstallAnywhere .Net Action 460
- InstallAnywhere Enterprise Action 461
- InstallAware Action 462, 465
- Installers
 - GPInstall Action 458
 - Group Policy 465
 - InnoSetup Action 459
 - InstallAnywhere .Net Action 460
 - InstallAnywhere Enterprise Action 461
 - InstallAware Action 462, 465
 - InstallShield 2008 Action 467
 - InstallShield Developer Action 468
 - InstallShield Pro Action 471
 - InstallShield Pro Windows Installer Edition Action 474
 - InstallShield Universal Action 475
 - NSIS Action 477
 - Setup Factory 486
 - SetupBuilder Compile Project 478
 - SetupBuilder Run Installer 483
 - Wise for Windows Installer Action 496
 - Wise InstallBuilder/InstallMaster Action 497
 - WiX 499
- InstallShield 2008 Action 467
- InstallShield Developer Action 468
- InstallShield Options 470
- InstallShield Pro Action 471
- InstallShield Pro Windows Installer Edition Action 474
- InstallShield Universal Action 475
- Interactive
 - Ask Question Action 505
 - Choose One Action 508
 - Input Box 510
 - MessageBox 511
 - Multi Question Action 512
 - Prompt for File or Directory 514
 - Prompt for Password 516
 - Prompt for Variables 518
 - Prompt for Variables (Enhanced) 520
- Internet
 - Convert HTML to MHT File 524
 - Download 566
 - FTP Actions 525
 - FTP Client 541
 - FTP Upload 535
 - HTTP Get Action 545
 - ICQ Action 555
 - NNTP News Post Action 548
 - Send Email Action 549

- Internet
 - Send MSN Message 556
 - SSH Batch Execute Action 583
 - SSH Close Action 581
 - SSH Execute Command Action 582
 - SSH Open Action 580
 - Telnet Action 560
 - Internet Information Services 649
 - IIS5 650
 - IIS6 663
 - IIS6 FTP 679
 - Internet Information Services 5
 - Backup IIS Metabase Action 652
 - Create Virtual Directory Action 653
 - Create Web Site Action 656
 - Delete Virtual Directory Action 659
 - Delete Web Site Action 660
 - Restore IIS Metabase Action 660
 - Start/Stop/Pause Website Actions 661
 - Start/Stop/Restart WWW Service Actions 661
 - Unload IIS Application Action 661
 - Internet Information Services 6
 - Backup IIS Metabase Action 666
 - Create Application Pool 668
 - Create Virtual Directory Action 669
 - Create Web Site Action 671
 - Delete Application Pool 674
 - Delete Virtual Directory Action 675
 - Delete Web Site Action 676
 - Restore IIS Metabase Action 676
 - Start/Stop/Pause Website Actions 677
 - Start/Stop/Restart WWW Service Actions 677
 - Unload IIS Application Action 677
 - Internet Information Services 6 FTP
 - Create FTP Directory Action 681
 - Create FTP Site Action 684
 - Delete FTP Directory Action 689
 - Delete FTP Site Action 690
 - Start/Stop/Pause FTP Site Actions 690
 - Start/Stop/Restart FTP Service Actions 690
 - Interop References Fixup 797
 - ISO Buring 188
 - ISO Creation 190
 - ISQL Location 261
 - Iterators
 - ADO Dataset Iterator 267
 - File Contents Iterator 706
 - File Iterator 707
 - Folder Iterator 709
 - For Loop 380
 - General Information 704
 - INI File Iterator 710
 - List Iterator 711
 - VisualStudio .Net Solution Iterator 244
 - While Loop 420
 - XML Node Iterator 1405
- ## - J -
- J# Compiler Action 251
 - J# Project Compiler Action 252
 - Java Compiler Action 220
 - Java JDK Configuration 222
 - JavaScript 27, 86
 - JDK Configuration 222
 - JPEG Image Manipulation 725
 - JPEG Lossless Transform 729
 - JPG Image Manipulation 725
 - JScript 27, 86
- ## - K -
- Kill Process (WMI) 1375
 - Known Problems 1504
- ## - L -
- Labelsyntax 1069
 - Language Tools
 - Visualizer 717
 - License 4
 - Licensing
 - Amradillo 713
 - ASProtect 714
 - ProActivate 716
 - Line Feed 856
 - List Archive 154
 - List Iterator Action 711
 - List Sort 736
 - List VMs (Virtual Server) 1282
 - List VMs in (VMWare Server) 1291
 - Live Logging 47
 - Load Property Set from FBD 823
 - Load Property Set from INI file 824
 - Load Property Set from Registry 825
 - Load Variables From INI Action 899
 - Localization Tools
 - Visualizer 717
 - Lock 1070
 - Lock file 732
 - Log Action 278

Log Values of Property Set 826
 Logging Options 57
 Login 1078
 LogToVariable Property 100
 Loop 706, 707, 709, 710, 711
 Lossless Transform 729
 Lowercase 858

- M -

MadExcept Action 216
 Mage 767
 Main 7, 18
 Manifest Generation and Editing 767
 Map Network Drive 1333
 Mercurial
 Add 1020
 Annotate 1020
 Archive 1022
 Branch 1023
 Bundle 1023
 Cat 1025
 Choosing Repositories 1017
 Clone 1025
 Commit 1027
 Copy 1028
 Create Repository 1029
 Export 1030
 Forget 1031
 Generic 1032
 Heads 1032
 Import 1033
 Log 1034
 Manifest 1034
 Merge 1034
 Parents 1035
 Pull 1036
 Push 1036
 Remove 1038
 Revert 1038
 Rollback 1039
 Specifying Files 1020
 Status 1039
 Summary 1041
 Tag 1041
 Tip 1042
 Unbundle 1042
 Update 1043
 Verify 1044
 Merge PDF Files 810
 Merge XML Action 1393

MessageBox 511
 MessageBox Constants 97
 Messages 70
 MHT File Conversion 524
 Microsoft C# Compiler Action 246
 Microsoft C# Project Compiler Action 247
 Microsoft Deployment Project 800
 Microsoft J# Compiler Action 251
 Microsoft J# Project Compiler Action 252
 Microsoft Powershell 1325
 Microsoft Resource Compiler 205
 Microsoft VB.NET Compiler Action 249
 Microsoft VB.NET Project Compiler Action 249
 MKS
 Add 1046
 Add Label 1047
 Check In 1048
 Check Out 1049
 Checkpoint 1050
 Close Change Package 1051
 Connect 1052
 Create Change Package 1053
 Create Sandbox 1054
 Delete Label 1055
 Disconnect 1056
 Drop 1057
 Generic 1058
 Lock 1059
 Open Change Package 1060
 Project Add 1061
 Project Check In 1062
 Resynchronize 1063
 Unlock 1064
 Modified date 333
 Mouse Clicks 428
 Mouse Movement 429
 Move Directory Action 335
 Move Files Action 336
 Moving Actions 14
 MSBuild Project Action 180
 MSDeploy 568, 569, 575, 576, 578
 MSI 488
 Install Application 493
 MSI Generic 488
 MSI Merge 489
 MSI Transform 490
 MSI Update Properties 491
 MSI Validation 492
 Reinstall Application 494
 Uninstall Application 495

MSIL Assembler 253
 MSN Send Message 556
 MSTest Action 883
 Multi Question Action 512
 Mutex 732

- N -

NAnt Project Action 183
 Natural sort 736
 NCover 780
 NCover Reporting 785
 NCoverExplorer 783
 NDepend 836
 NDoc 444
 Net Send Message Action 1338
 Network 804
 Network Drive Mapping 1333
 Network Drive Unmapping 1337
 New Line 856
 News Post Action 548
 NGen Install 748
 NGen Uninstall 750
 Ngen Update 752
 NNTP News Post Action 548
 NuGet 791
 NuGet Delete 791
 NuGet Install 792
 NuGet Pack 794
 NuGet Push 795
 NuGet Spec 796
 NullSoft NSIS Action 477
 Number (Random) 722
 NUnit Action 887

- O -

OnFailure 7, 18
 Open for Delete 1072
 Open for Edit 1073
 Opened 1074
 Options 1074
 Design Time 52
 InstallShield Options 470
 Logging 57
 Run Time 54
 Script Debugger 90
 Script Editor 88
 Setting via scripting 104
 Source Control Integration 65

SQL Server Options 261
 Validation 60
 Options Dialog 51
 Options screen 35
 Order 736
 Orientation 20
 OSQL Location 261
 Output Monitors 139
 Output Tree 47
 Override progress 281
 Overview 1

- P -

Package Manager 71
 Pad String 864
 Parallel Build Process 391
 Parent Function 100
 Pascal Analyzer Action 834
 Password for Project 81
 Password Prompt 516
 Path Manipulation 854
 Pause (Delay Action) 377
 Pause VM (Virtual Server) 1284
 PauseInterval Property 100
 PDF
 Convert PDF to another format 806
 Convert to PDF 807
 Insert Text 808
 Merge PDF files 810
 PDF Convert to another format 806
 PDF Insert Text 808
 PDF Merge files 810
 Perforce 1065, 1066, 1067, 1068, 1069, 1070, 1072,
 1073, 1074, 1075, 1076, 1077, 1078
 Perforce Command [deprecated] 1072
 Perforce Generic 1068
 Perforce Sync with View [deprecated] 1070
 Perform Mouse Click 428
 Perform Mouse Move 429
 Plastic SCM
 Add Items 1081
 Check In 1083
 Check Out 1085
 Configure Workspace 1087
 Create New Branch 1089
 Create Workspace 1090
 Generic 1091
 Label Workspace 1093
 Options 1080

- Plastic SCM
 - Update 1093
- Plugins 71
- Powershell 86, 1325
- Precompile ASP.NET 2.0 753
- Printing Statistics Charts 43
- ProActivate 716
- Process Close Action 1308
- Process Info Action 1376
- Process Iterator (WMI) 1377
- Process Priority 145
- Processor Affinity 145
- Progress 282
- Progress bar 24, 281
- Project file types 77
- Project Global Script 79
- Project Password 81
- Project Summary 83
- Project Tree 22
- Project Variables 115
- Prompt for File or Directory Action 514
- Prompt for Password 516
- Prompt for Variables (Enhanced) Action 520
- Prompt for Variables Action 518
- Properties Property 100
- Properties Tab 17
- Property Set
 - Assign Properties 819
 - Define 817
 - Increment Value 820
 - Load from FBD 823
 - Load from INI 824
 - Load From Registry 825
 - Log Values 826
 - Save to FBD 827
 - Save to INI 828
 - Save to Registry 829
 - to Variable 830
- Property Set Definitions 813
 - .Net Property Sets 814
 - VB property Sets 814
 - Win32 Property Sets 813
- Property Set to Variable 830
- Property Sets
 - Accessing 816
 - Delphi Compiler 206
- PsTools 1347
 - PsExec 1348
 - PsFile 1351
 - PsGetSID 1352
 - PsInfo 1353
 - PsKill 1354
 - PsList 1356
 - PsLoggedIn 1357
 - PsLogList 1358
 - PsService 1362
 - PsShutdown 1364
 - PsSuspend 1365
- PureCM
 - About 1095
 - Add File 1096
 - Checkout File 1097
 - Create Stream 1104
 - Create Stream Folder 1106
 - Create Workspace 1100
 - Delete File 1097
 - Delete Stream 1105
 - Delete Stream Folder 1107
 - Generic 1110
 - Lock Stream 1107
 - Options 1095
 - Remove Workspace 1103
 - Revert File 1098
 - Submit 1099
 - Unlock Stream 1108
 - Update Workspace 1102
- PVCS 1111
 - PVCS Add Label 1111
 - PVCS Get 1112
 - PVCS List Files 1113
 - PVCS Promote 1114
 - PVCS Put 1115

- Q -

- QA Wizard 889
- Queue 838
 - Queue Clear 841
 - Queue Define 839
 - Queue get count 842
 - Queue Insert 843
 - Queue is empty 844
 - Queue Iterator 845
 - Queue log items 847
 - Queue Peek 849
 - Queue Pop 850
 - Queue Push 851
- Quick help 26
- Quotes in Strings 866
- QVCS
 - Add File 1130

QVCS

- Check In 1130
- Check Out 1130
- Get Latest Version 1131
- Label Files 1131
- Options 1130
- Undo Checkout 1131

- R -

- Raise Exception Action 389
- Random 736
- Random Number 722
- RAR Files 176
- Rasdial VPN 1339
- Read Ini 452
- Read Text File Action 338
- Read XML Value to Variable 1396
- Reboot Action 1342
- Rebuild Indexes Action 262
- REGASM 756
- Register Assembly in COM 756
- Register COM+ 1315
- Register DLL/OCX Action 1341
- Registry
 - Delete Registry Value 453
 - Read Registry Value 453
 - Set Registry Value 453
- Registry Export 451
- Registry Key Save 451
- REGSVCS 759
- Regular Expression Reference 1496
- Reinstall Application 494
- Remote Network Connection 1339
- Remote Reboot Action 1342
- Remote Shutdown 1343
- Remote Task Scheduling 1373
- Remove Unused Space Action 263
- Remove whitespace 876
- Rename File or Directory Action 340
- Replace String 870
- Reset Virtual Machine 1296
- Reset VM 1292
- Reset VM (Virtual Server) 1285
- Re-sign Assembly Action 771
- Resize Images 725
- Resource Compiler 205
- Resource Script Compiler Action 215
- Restore Database Action 264
- Restore IIS5 Metabase Action 660

- Restore IIS6 Metabase Action 676
- Reverse String 872
- Revert 1075
- Revert to Snapshot of Virtual Machine 1296
- Robocopy 347, 348, 349, 351, 352
- RoboCopy Actions 341
- Rotate Images 725
- RTF convert to PDF 807
- Run Action as User 143
- Run Action List Action 390
- Run AL.EXE Action 757
- Run ASPNET_REGIIS.EXE Action 758
- Run Command 45
- Run Process (WMI) 1379
- Run ResGen.exe Action 771
- Run Scheduled Task 1373
- Run Script Action 735
- Run SN.EXE Action 772
- Run Status Message 282
- Run Time Options 54
- Running 282

- S -

- Sandcastle
 - BuildAssembler 445
 - MRefBuilder 446
 - XslTransform 448
- Save Log to File 278
- Save Property Set to FBD 827
- Save Property Set to INI 828
- Save Property Set to Registry 829
- Save Variables To INI Action 907
- Save VM State (Virtual Server) 1286
- Save XML Document Object 1409
- SCC Provider 62
- Scheduled Task Add 1368
- Scheduled Task Delete 1371
- Scheduled Task End Running 1372
- Scheduled Task Run 1373
- Scheduling Builds 1433
- Script
 - Global Script 79
- Script Debugger Options 90
- Script Editor 27
- Script Editor Options 88
- Script Functions 91
- Scripting 86
 - Accessing Options 104

Scripting	86	Verify Digital Signatures	360
Accessing TStrings	105	Single Quotes	866
Action Methods	100	Sisualizer	717
Action Properties	100	Size of Files Action	332
Events	98	SMTP	549
Execute Condition	102	SN Tool	
Global Functions	91	Extract Public Key	762
Scrollbar Position	433	Generate Key Pair	766
Seapine QA Wizard	889	Install Key	766
Search	20	Re-sign Assembly	771
Search / Replace		Run SN.EXE	772
In IDE	30	Verify Strong Name	775
Text Find / Replace Action	741	Solution Iterator	244
SecureZIP	163, 166, 167, 169, 170, 173, 174	Sort Text List	736
SecureZIP Options	162	Sound	507
SecureZIP Add Files To Archive	163	Source Code Tools	
SecureZIP Common Pages	170	CodeHealer	832
SecureZIP Exclude List	174	NDepend	836
SecureZIP Extract Archive	166	Pascal Analyzer Action	834
SecureZIP Filters	173	Source Control Integration	62
SecureZIP List Archive Contents	167	Source Control Integration Options	65
SecureZIP Security	170	Source Safe	
SecureZIP Test Archive Integrity	169	Add Files	1148
Selecting Actions	14	Branch	1149
Send Email	549	Check File Status	1151
Send Keyboard Input	430	Check In	1152
Send Message Action	1338	Check Out	1160
Send MSN Message	556	Checkout	1154
SendInput	428	Create	1161
SendKeys	430	Get Working Directory	1156
SendLogMessage Procedure	100	GetLatest Version	1155
SendProgress Procedure	100	Label Files	1156
Sentence	858	Options	1148
Server	2	Override Global Options	1157
Serviced Components Installation Tool	759	Project Checkouts	1158
Set Build Status Message	282	Share	1159
Set File Attributes Action	355	Space	876
Set Run Status Message	282	Spell Check	738
Set Scrollbar Position	433	SQL Server	
Set Variable Action	909	Backup Database Action	258
SetLogTitle Procedure	100	Best Practices Analyzer	259
Setup Factory	486	Check Catalogue Action	261
SetupBuilder Compile Project	478	Check Database Action	261
SetupBuilder Run Installer	483	DTSRun Action	255
Shell Execute Action	1346	Execute SQL Action	256
Shortcut create	1322	Rebuild Indexes Action	262
Shutdown Action	1343	Remove Unused Space Action	263
Signtool		Restore Database Action	264
Sign Files	356	Set Access Mode Action	265
Timestamp Files	359	Update DB Statistics Action	266
		SQL Server Best Practices Analyzer	259

- SQL Server Options 261
- SQLCmd Location 261
- SQLMaint Location 261
- SSH Batch Execute Action 583
- SSH Close Action 581
- SSH Execute Command Action 582
- SSH Open Action 580
- Stack 838
- Stack Clear 841
- Stack Define 839
- Stack get count 842
- Stack Insert 843
- Stack is empty 844
- Stack Iterator 845
- Stack log items 847
- Stack Peek 849
- Stack Pop 850
- Stack Push 851
- Start Virtual Machine 1298
- Start VM 1292
- Start VMs (Virtual Server) 1287
- Start/Stop/Pause IIS5 Website Actions 661
- Start/Stop/Pause IIS6 FTP Site Actions 690
- Start/Stop/Pause IIS6 Website Actions 677
- Start/Stop/Restart IIS5 WWW Service Actions 661
- Start/Stop/Restart IIS6 FTP Service Actions 690
- Start/Stop/Restart IIS6 WWW Service Actions 677
- StarTeam
 - Add Files 931
 - Apply Label 931
 - Check In 931
 - Check Lock/Unlock Files 932
 - Check Out 931
 - Create Label 931
 - Delete Files 932
 - Generic 932
 - List Files 932
 - Update Status 932
- Statistics 37
- Status 282
- Stepping 33
- Stop Command 45
- Stop Run Action 395
- Stop Virtual Machine 1299
- Stop VM 1292
- Stopping a build 45
- String Manipulation
 - Breaks 856
 - Case 858
 - Concatenation 860
 - Encryption 862
 - Find / Replace 741
 - Index 868
 - Pad 864
 - Path 854
 - Pos 868
 - Position 868
 - Quotes 866
 - Replace 870
 - Reverse 872
 - Substring 874
 - Trim 876
- Submit 1076
- Subst Drive Action 1335
- Substring 874
- Subversion
 - Add 1195
 - Checkout 1196
 - Cleanup 1197
 - Commit 1197
 - Copy 1198
 - Delete 1200
 - Export 1201
 - Generic 1202
 - Import 1203
 - Info 1204
 - MkDir 1205
 - Revert 1206
 - Status 1207
 - Switch 1209
 - Update 1210
- Subversion Add 1197
- Success Ratio Chart 37
- Support 1507
- SuppressStatusMessages Property 100
- Surround SCM
 - CheckIn 1132
 - Checkout 1133
 - Checkout Report 1134
 - Create Branch 1137
 - Freeze Branch 1138
 - Generic 1138
 - Get 1139
 - Label 1143
 - Remove Working Directory 1145
 - Set Working Directory 1146
 - Unfreeze Branch 1147
- Suspend Virtual Machine 1300
- Suspend VM 1292
- Switch Action 396
- Sync 1076

Sysinternals 1347
 PsExec 1348
 PsFile 1351
 PsGetSID 1352
 PsInfo 1353
 PsKill 1354
 PsList 1356
 PsLoggedOn 1357
 PsLogList 1358
 PsService 1362
 PsShutdown 1364
 PsSuspend 1365
 System Tray 73
 System Variables 118

- T -

Tag 1077
 Target Property Page 422, 423
 Task Scheduling
 Add Scheduled Task 1368
 Delete Scheduled Task 1371
 End Running Scheduled Task 1372
 Run Scheduled Task 1373
 Task Scheduling Remote Tasks 1373
 Team Build Integration 1447, 1450, 1452, 1454, 1455
 Team Coherence
 Attach Label 1118
 Check In 1118
 Check Out 1119
 Connect 1121
 Create Label 1122
 Create View 1123
 Delete View 1124
 Detach Label 1124
 Generic 1125
 Get 1126
 Promote 1126
 Set View 1128
 Sync 1128
 Update View 1129
 Telnet Action 560
 Terminate Command 45
 Test Archive 155
 TestComplete 2/3 880
 TestComplete 4 881
 Testing Tools
 AQTest Action 878
 AutomatedQA Test Complete 4Actions 881
 AutomatedQA Test Complete Actions 880

MStest Action 883
 NUnit Action 887
 Seapine QA Wizard 889
 Text Find / Replace 741
 Text Sort 736
 TIFF Image Manipulation 725
 Timeout (Delay Action) 377
 Timing 137
 TLBImp Fixup references 797
 Touch Files Action 365
 Transform Image 729
 Transform XML Action 1398
 Tray Icon 73
 Trim String 876
 Try Action 397
 TStrings 105
 Turn off VM (Virtual Server) 1289
 Type Library Export Action 773
 Type Library Import Action 774

- U -

Uninstall Application 495
 Unload IIS5 Application Action 661
 Unload IIS6 Application Action 677
 Unlock 1077
 Unmap Network Drive 1337
 Update Archive 155
 Update DB Statistics Action 266
 Update Label 1078
 Updates 26, 69
 Uppercase 858
 UPX Pack 274
 User Action Lists 7
 User Variables 116
 Using PropertySets with Delphi Compiler 206
 Using Variables 112

- V -

Validate XML Action 1400
 Validation 50
 Validation Options 60
 Variable Options 123
 Variable Sense 123
 Variables
 Action List Parameters 121
 Adding & Editing Variables 109, 123
 Application Variables 118
 Environment Variables 117

Variables

- Escaping References 114
- Load from INI Action 899
- Overview 106
- Predefined FinalBuilder Variables 118
- Project Variables 115
- Save To INI Action 907
- Set Variable Action 909
- User Variables 116
- Using Variables 112

Vault

- Add 1165
- Branch 1165
- Check In 1166
- Check Out 1168
- Cloak 1169
- Commit 1170
- Create Folder 1171
- Create Label 1172
- Delete File/Folder 1173
- Diff 1174
- File Status 1174
- Get 1175
- Get Label 1177
- Get using Wildcards 1177
- Get Version 1179
- GetLabelDiffs 1180
- List Checkouts 1180
- Move File/Folder 1182
- Pin 1183
- Rename File/Folder 1184
- Set Working Folder 1185
- Share File/Folder 1186
- UnCloak 1187
- Undo Checkout 1187
- UnPin 1189

- VB.NET Compiler Action 249
- VB.NET Project Compiler Action 249
- VB6 Project Group 1417
- VBScript 27, 86
- VDProj File Updater 800
- Verify Strong Name Action 775
- Version Control Integration 62
- Version Control Systems
 - AccuRef 911
 - ClearCase 933
 - CVS Actions 963
 - CVS Command 964
 - JediVCS 1013
 - MKS Source 1045
 - Perforce 1065

- PVCS 1111
- QVCS 1130, 1131
- Source Safe 1152, 1157
- StarTeam 929, 931, 932
- Subversion 1190, 1195, 1196, 1197, 1198, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1209, 1210
- Surround SCM 1132, 1133, 1134, 1137, 1138, 1139, 1143, 1145, 1146, 1147
- Team Coherence 1118, 1119, 1121, 1122, 1123, 1124, 1125, 1126, 1128, 1129
- Team Foundation Source Control 1211
- TeamCoherence 1116
- Vault 1163, 1165, 1166, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1177, 1179, 1180, 1182, 1183, 1184, 1185, 1186, 1187, 1189

Version History 3

View Log 35

Virtual Server 1280

- Check VM Status 1281
- List VMs 1282
- Pause VM 1284
- Reset VM 1285
- Save VM State 1286
- Start VMs 1287
- Turn off VM 1289

Visual C++ 6 228

Visual SourceSafe 1152, 1157

Visual Studio .NET 232

Visual Studio Addin - Team Build 1447, 1450, 1452

VisualStudio .Net Solution Iterator 244

VMWare 1301, 1302, 1303, 1305, 1306

VMWare Server 1290, 1301, 1303, 1305, 1306

- List Virtual Machines 1302
- List VMs 1291
- Reset VM 1292
- Start VM 1292
- Stop VM 1292
- Suspend VM 1292

VMWare Workstation

- Check Virtual Machine Status 1294
- Create Snapshot 1295
- Reset Virtual Machine 1296
- Revert to Snapshot 1296
- Start Virtual Machine 1298
- Stop Virtual Machine 1299
- Suspend Virtual Machine 1300

VPN (Rasdial) 1339

- W -

Wait (Delay Action) 377

Wait for Window or Control 435

Waiting For Events

- Certain Date or Time 417
- Command Output 401
- Files 403
- FinalBuilder Variables 416
- Ini File Change 405
- Overview 400
- Process 407
- Registry Change 409
- Remote Computer 411
- Script Result 413

Watches 33, 34

Web Deploy 568

- Delete 575
- Deploy 569
- Dump 576
- Get Dependencies 576
- Get System Info 578
- Package 569
- Sync 569

WGet Download Action 566

While Loop Action 420

Whitespace 876

Window Exists Action 1367

Windows Installer 488

- Install Application 493
- MSI Generic 488
- MSI Merge 489
- MSI Transform 490
- MSI Update Properties 491
- MSI Validation 492
- Reinstall Application 494
- Uninstall Application 495

Windows Powershell 1325

Windows Scheduler Actions

- Add Scheduled Task 1368
- Delete Scheduled Task 1371
- End Running Scheduled Task 1372
- Run Scheduled Task 1373

Windows Scheduling Remote Tasks 1373

WinHelp 449

WinRAR Action 176

Wise for Windows Installer Action 496

Wise Installer Action 497

WiX (Windows Installer XML Toolkit) 499

WiX Compiler 499

WiX Linker 501

Wizards

- Embarcadero Project Group 1410
- VB6 Project Group 1417

WMI Kill Process Action 1375

WMI Process Info Action 1376

WMI Process Iterator 1377

WMI Run Process Action 1379

Working with Actions 14

Write Ini 457

Write Text File Action 367

- X -

XCopy Action 368

XenoCode Action 778

XML

- Altova DiffDog 1383
- Create XML Node 1384
- Define XML Document Object 1401
- Delete XML Element/Attribute 1388
- Delete XML Nodes 1390
- Edit XML File Action 1391
- Extract XML Fragment Action 1394
- Merge XML Action 1393
- Node Iterator 1405
- Read XML Value to Variable 1396
- Save XML Document Object 1409
- Transform XML Action 1398
- Validate XML Action 1400

XML Define Document Object 1401

XML Node Iterator 1405

XML Save Document Object 1409

- Z -

Zip Files 149, 153, 154, 155, 156, 160