FinalBuilder

Automating the Build Process

Table of Contents

Part I FinalBuilder 1

1 FinalBuilder Overview 1
2 Version History 2
3 License 3

Part II Getting Started 6

1 FinalBuilder Basic Concepts 6
2 FinalBuilder IDE Modes 7
IDE Design Mode 8
Project Tree 8
Action Types 10
Action Inspector 11
Action Lists 11
Working with Actions 13
Script Editor 15
Quick Help 15
IDE Debug Mode 17
Watches 18
IDE Running Mode 19
Running a Project 19
Run Log 21
Run History 22
Validation Errors 22
Estimated Progress 23
3 Options Dialog 25
4 Project Edit Password 26
5 Project Summary 28
6 Messages 29
7 Tray Icon 30
8 Check for Updates 32
9 Project Files & Other Files 33
10 FinalBuilder Plugins 34
11 Project Global Script 35
12 Package Manager 36

Part III Scripting 38

1 Scripting in FinalBuilder 38
2 Global Script functions 40
Format DateTime Formatting Options 43
MessageBox Constants 45
3 Action Script Events 46
Action Properties and Methods 48
Execute Condition 50
4 Accessing the Options settings via scripting 52

© 2001-2006 VSoft Technologies Pty Ltd

5 Accessing TStrings based properties 55

Part IV Variables 56

1 Variables Overview 56
2 Adding Variables 58
3 Using Variables 59
4 Project Variables 60
5 User Variables 61
6 Environment Variables 62
7 System Variables 63
8 Action List Parameters 64

Part V Actions Reference 66

1 Common Action Properties 69
2 Flow Control 71

Delay Action 71
Include FinalBuilder Project 72
Run Action List Action 74
Exit Action List Action 75
If .. Then Action 76
If Prev Action Failed Action 78
Else Action 79
Stop Build Action 80
Raise Exception 81
Switch Action 82
Case Action 83
Try/Catch/Finally/End Actions 84
For Loop Action 87
While Loop Action 89
Running actions in Parallel 90
Waiting For Events (Wait For actions) 94
Wait For File 95
Wait For Variable 97
Wait For Process 99
Wait For Script Result 100
Wait For Remote Computer 101
Wait For Ini File 102
Wait For Registry 104
Wait For Command 106
Wait Until 108

3 Property Sets 111

Property Set Definitions 113
Win32 Property Sets 113
.Net Property Sets 114
VB Property Sets 114
Accessing PropertySets 116
PropertySet define 117
PropertySet save to INI file 118
PropertySet save to Registry 119
PropertySet save to FBD 120
PropertySet load from INI file 121
PropertySet load from Registry 122
PropertySet load from FBD 123

© 2001-2006 VSoft Technologies Pty Ltd

PropertySet increment value ... **124**
PropertySet log values ... **125**
PropertySet assign properties ... **126**
PropertySet to Variable ... **127**
4 File Sets ... **129**
Accessing FileSets ... **130**
FileSet Define ... **132**
FileSet Patterns ... 132
FileSet Filters ... 134
FileSet to Variable ... **136**
FileSet log files ... **137**
FileSet Refresh ... **138**
Copy/Move FileSet Action ... **139**
FileSet Delete ... **141**
5 Version Control Actions ... **142**
ClearCase ... **142**
ClearCase Pathnames ... 142
ClearCase Object Selector ... 144
Base ClearCase ... 146
ClearCase Update Snapshot View Action ... 146
ClearCase Check Out Action ... 148
ClearCase Check In Action ... 149
ClearCase Undo Checkouts Action ... 150
ClearCase Find Checkouts Action ... 150
ClearCase Make Element Action ... 151
ClearCase Make Label Type Action ... 152
ClearCase Apply Label Action ... 154
ClearCase Make Attribute Type Action ... 154
ClearCase Apply Attribute Action ... 156
ClearCase Lock Action ... 157
ClearCase Unlock Action ... 158
ClearCase Run Cleartool (Generic) Action ... 159
ClearCase Get Config Spec Action ... 160
ClearCase Set Config Spec Action ... 161
UCM ... 162
UCM Make Activity Action ... 162
UCM Set Current Activity Action ... 163
UCM Check In Files For Activity Action ... 164
UCM Undo Checkouts for Activity Action ... 164
UCM Make Baseline Action ... 165
UCM Make Baseline from Label Action ... 167
UCM Change Baseline ... 168
UCM Change Stream ... 170
Source Safe ... **173**
Source Safe Check In File(s) Action ... 173
Source Safe Check Out File(s) Action ... 174
Source Safe Get Latest Version Action ... 175
Source Safe Label File(s) ... 176
Source Safe Project Checkouts ... 177
Source Safe Check File Status ... 178
Source Safe Undo CheckOut Action ... 179
Source Safe Add Files Action ... 180
Source Safe Branch ... 181
Source Safe Share ... 182
Source Safe Get Working Directory ... 183
Source Safe Override Global Options ... 183
Perforce ... **185**
Perforce Options ... 185

© 2001-2006 VSoft Technologies Pty Ltd

Perforce Create Branch ... 185
Perforce Delete Branch ... 185
Perforce Sync ... 185
Perforce Submit ... 185
Perforce open for Edit ... 185
Perforce Opened ... 185
Perforce Create Label ... 185
Perforce Update Label ... 185
Perforce Delete Label ... 185
Perforce Labelsync ... 185
Perforce open for Delete ... 185
Perforce Tag ... 186
Perforce Lock ... 186
Perforce Unlock ... 186
Perforce Revert ... 186
Perforce Create Changelist ... 186
Perforce Delete Changelist ... 186
Perforce Generic ... 186
Perforce Old Actions ... 186
Peforce Synchronise with View Action ... 186
Perforce Command Action ... 188
CVS Actions ... 189
CVS Command Action ... 190
Borland StarTeam Actions ... 191
StarTeam Check In Action ... 192
StarTeam Check Out Action ... 192
StarTeam Lock/Unlock Files Action ... 192
StarTeam Create Label Action ... 192
StarTeam Apply Label Action ... 192
StarTeam Update Status Action ... 193
StarTeam Delete Files Action ... 193
StarTeam Add Files Action ... 193
StarTeam List Files Action ... 193
QVCS Actions ... 194
QVCS Add File Action ... 194
QVCS Check In File(s) Action ... 194
QVCS Check Out File(s) Action ... 194
QVCS Get Latest Version Action ... 194
QVCS Labels File(s) Action ... 194
QVCS Undo Check Out File(s) Action ... 194
QSC Team Coherence Actions ... 195
Team Coherence Connect Action ... 197
Team Coherence Set View Action ... 198
Team Coherence Get Action ... 199
Team Coherence Check In Action ... 199
Team Coherence Check Out Action ... 200
Team Coherence Attach Label Action ... 202
Team Coherence Detach Label Action ... 203
Team Coherence Create Label Action ... 204
Team Coherence Promote Action ... 204
Team Coherence Sync ... 205
Team Coherence Create View ... 207
Team Coherence Update View ... 208
Team Coherence Delete View ... 208
SourceGear Vault Actions ... 209
Vault Check Out Action ... 211
Vault Get Action ... 211
Vault Get using Wildcards Action ... 212

© 2001-2006 VSoft Technologies Pty Ltd

Vault Branch Action ... 213
Vault Commit Action ... 215
Vault Check In Action ... 215
Vault Cloak Action ... 217
Vault UnCloak Action ... 218
Vault Add Action ... 219
Vault Create Folder Action ... 220
Vault Delete File/Folder Action ... 221
Vault Get Version Action ... 222
Vault Create Label Action ... 223
Vault Move File/Folder Action ... 224
Vault Rename File/Folder Action ... 225
Vault Share File/Folder Action ... 226
Vault Undo Checkout Action ... 226
Vault Pin File/Folder Action ... 228
Vault UnPin File/Folder Action ... 229
Vault Get Label ... 230
Vault Set Working Folder ... 231
Vault Diff ... 232
Vault GetLabelDiffs ... 233
Vault ListCheckouts ... 233
Vault File Status ... 234
Subversion Actions ... 236
Subversion Add ... 238
Subversion Checkout ... 238
Subversion Cleanup ... 238
Subversion Copy ... 238
Subversion Commit ... 238
Subversion Export ... 238
Subversion Import ... 238
Subversion MkDir ... 238
Subversion Update ... 238
Subversion Status ... 238
Subversion Generic Action ... 238
Seapine Surround SCM ... 240
Surround SCM Global Options ... 240
Surround SCM Override Global Options ... 241
Surround SCM Get ... 241
Surround SCM CheckOut ... 243
Surround SCM CheckIn ... 244
Surround SCM Label ... 245
Surround SCM Create Branch ... 246
Surround SCM Freeze Branch ... 247
Surround SCM Unfreeze Branch ... 248
Surround SCM Checkout Report ... 248
Surround SCM Generic ... 251
Team Foundation Source Control ... 252
Options ... 252
Team Foundation Get ... 253
Team Foundation Check Out ... 255
Team Foundation Check In Pending Changes ... 256
Team Foundation Undo Pending Changes ... 257
Team Foundation Add ... 258
Team Foundation Lock ... 260
Team Foundation Unlock ... 261
Team Foundation Rename/Move ... 262
Team Foundation Map Working Folder ... 263
Team Foundation Unmap Working Folder ... 264

© 2001-2006 VSoft Technologies Pty Ltd

Team Foundation Apply Label ... 265
Team Foundation Remove Label ... 266
Team Foundation Delete ... 267
Team Foundation Undelete ... 268
Team Foundation Branch ... 270
Team Foundation Merge ... 271
Team Foundation Shelve Pending Changes ... 273
Team Foundation Unshelve Pending Changes ... 275
Team Foundation Delete Shelveset ... 277
Team Foundation Work Items ... 278
Team Foundation Query Work Items ... 280
Team Foundation Create Work Item ... 282
Team Foundation Edit Work Item(s) ... 283
Team Foundation Copy Work Item(s) ... 283
JediVCS ... 285
Scripting JediVCS actions ... 285
6 Iterators ... 288
File Iterator ... 290
List Iterator ... 292
Folder Iterator ... 293
File Contents Iterator ... 294
INI File Iterator ... 295
7 Interactive ... 296
Prompt for Variables Action ... 297
Prompt for Variables Action (Enhanced) ... 299
Ask Question Action ... 303
Multi Question ... 304
Prompt for File or Directory ... 306
Message Box Action ... 307
Input Box Action ... 308
8 Misc Actions ... 309
Action Group ... 309
Beep Action ... 310
Export Log Action ... 311
Get DateTime Action ... 312
Comment Action ... 314
Run Script Action ... 315
CityDesk Action ... 316
Text Find Action ... 317
Save FinalBuilder Project Action ... 319
Log Variable Values ... 320
9 Variables ... 321
Set Variable Action ... 321
If Variable Defined Action ... 323
Save Variables To Ini ... 325
Load Variables From Ini ... 327
Define Variable Action ... 330
Reset Variable to Default ... 332
Append to Variable Action ... 333
10 Files & Directories ... 334
File Dependency Action ... 334
Check File Exists Actoin ... 338
Delete File(s) Action ... 339
Create Directory Action ... 341
Delete Directory Action ... 342
Move Directory Action ... 344

© 2001-2006 VSoft Technologies Pty Ltd

Copy File(s) Action ... **345**
Move File(s) Action ... **347**
Copy/Move File List ... **349**
Set File Attributes Action ... **351**
Touch File(s) ... **352**
Text Replace Action ... **353**
Concatenate Files Action ... **355**
Rename File or Directory Action ... **356**
Authenticode ... **357**
Create Text File ... **359**
Write Text File ... **360**
Read Text File ... **361**
XCopy ... **362**
RoboCopy ... **363**
Extract File Version ... **364**
Calculate File CRC32 ... **365**
Calculate File MD5 ... **367**
File Comparison Tools ... **369**
File Compare Action ... 369
Beyond Compare Action ... 370
Araxis Merge Compare Actions ... 371

11 Windows OS ... **374**

Execute Program Action ... **374**
Run DOS Command Action ... **376**
Subst Drive Action ... **378**
Window Exists Action ... **379**
Net Send Message Action ... **380**
Get Disk Free Space ... **381**
Register DLL/OCX Action ... **382**
Control Service Action ... **383**
WMI Run Process Action ... **385**
WMI Kill Process Action ... **386**
WMI Process Iterator ... **387**
WMI Process Info Action ... **389**
Shell Execute ... **391**
If COM Class Registered ... **392**
Remote Reboot Action ... **394**
Remote Shutdown Action ... **395**
Map Network Drive Action ... **396**
Unmap Network Drive Action ... **398**
Close Process Action ... **399**
COM+ Administration ... **401**
COM+ Register ... 401
COM+ Delete ... 402
COM+ Edit Application ... 403
COM+ Shutdown Application ... 407
COM+ Start Application ... 408
Create Shortcut ... **410**
Rasdial VPN / Remote Network Connection ... **413**
Windows Scheduler ... **415**
Add Scheduled Task ... 415
Delete Scheduled Task ... 418
Run Scheduled Task ... 419
End Running Scheduled Task ... 420
Scheduling Remote Tasks ... 420
"AT" Scheduler ... **422**
Schedule Add ... 422
Schedule Delete ... 424

© 2001-2006 VSoft Technologies Pty Ltd

Schedule List 425
12 Compilers 427
Java Compiler Action 427
Borland Compiler Options 429
JDK Configurations 429
Microsoft 430
Compile Visual Basic Project 430
VB6 Version Compatibility 433
Using PropertySets with VB6 434
Visual C++ 6 Action 435
Using PropertySets with Visual C++ 438
Visual Studio .NET Action 439
Using PropertySets with VisualStudio.Net 441
Microsoft C# Compiler Action 444
Microsoft C# Project Compiler Action 444
Microsoft VB.NET Compiler Action 446
Microsoft VB.NET Project Compiler Action 446
Microsoft J# Compiler Action 448
Microsoft J# Project Compiler Action 448
Borland 450
Compile Delphi Project 450
Using PropertySets with Delphi 453
Compile Borland Resource Script 454
Borland C++ Builder Action 455
Borland C++Builder Scripting Reference 456
Borland C++Builder Compiler Options 458
Borland C++Builder Advanced Compiler Options 458
Borland C++Builder Linker Options 459
Borland C++Builder Advanced Linker Options 459
Borland C++Builder CPP Options 459
Borland C++Builder TASM Options 460
Borland C++Builder TLib Options 460
Borland C++Builder Pascal Options 461
Borland C++Builder Corba Options 461
Borland C++Builder CodeGuard Options 462
Using PropertySets with Borland C++ Builder 462
Borland C# Builder Project Compiler Action 463
Borland Delphi 8 for .NET Action 463
MadExcept Compiler Action 464
AssemblyInfo Updater Action 465
Chrome 467
Incredibuild 468
13 Installers 469
Wise InstallBuilder/InstallMaster 469
Wise For Windows Installer 471
InstallShield 472
InstallShield Global Options 472
InstallShield Pro - Std Edition 473
InstallShield Pro - Windows Installer Edition 476
InstallShield Developer 477
InstallShield Universal Installer 479
Inno Setup 480
GPInstall Action 481
InstallAnywhere Enterprise 482
InstallAnywhere .Net 483
InstallAware 484
WiX (Windows Installer XML Toolkit) 486
WiX Compiler (candle.exe) 486

© 2001-2006 VSoft Technologies Pty Ltd

WiX Linker (light.exe) 488
Nullsoft NSIS 491
14 Internet 492
Send Email (SMTP) 492
FTP Actions 498
FTP Upload 501
FTP Client (deprecated) 505
Telnet Client Action 509
HTTP Get Action 510
Send ICQ Message Action 511
Send MSN Message Action 512
NNTP News Post Action 516
15 Help Compilers 517
Build Doc-O-Matic Project 517
WinHelp Compiler 519
HTML Help Compiler 521
Help & Manual Action 522
NDoc Action 524
16 Ini Files & Registry 525
Read Ini File 525
Write Ini File 526
Read/Set/Delete Registry Value 527
17 Archiving 530
Create Zip File 530
Extract Zip File Action 533
WinRAR Action 534
7Zip 535
Create Archive 535
Test Archive 538
Extract Archive 538
Update Archive 539
List Archive 540
Delete from Archive 540
18 Testing Tools 541
AQTest 541
AutomatedQA TestComplete 2.x/3.x Actions 543
AutomatedQA TestComplete 4 Action 544
NUnit Action 546
MSTest 547
Run QA Wizard Batch File 548
19 Licensing Tools 550
ASProtect Action 550
ProActivate Action 551
Armadillo Action 552
20 CD/DVD Burner Actions 553
Burn CD/DVD Action 554
Create ISO Action 556
Burn ISO Action 557
Check Ready Action 558
Erase CD/DVD RW 559
21 .NET Actions 560
.Net Framework Tools 560
Precompile ASP.NET 2.0 Application 560
Run AL.EXE 563
Register Assembly in COM [REGASM] 564

© 2001-2006 VSoft Technologies Pty Ltd

Run ASPNET_REGIIS.EXE ... 565
NGen 2.0 Native Image Install Action ... 566
NGen 2.0 Native Image Uninstall Action ... 569
NGen 2.0 Native Images Update Action ... 570
.Net SDK Tools ... 572
Generate Key Pair [SN] ... 572
Verify Strong Name [SN] ... 572
Install Key in Container [SN] ... 573
Extract Public Key [SN] ... 575
Re-sign Assembly [SN] ... 576
Run SN.EXE ... 576
GAC Install [GACUTIL] ... 577
GAC Uninstall [GACUTIL] ... 578
GAC Download Cache [GACUTIL] ... 580
Type Library Import [TLBIMP] ... 580
Type Library Export [TLBEXP] ... 581
Run ResGen.exe ... 582
3rd Party Tools ... 584
FxCop ... 584
Dotfuscator ... 584
Demeanor ... 585
XenoCode ... 585
Other ... 587
Fix TLBImp Project Reference ... 587
22 XML Actions ... 588
Read XML Value to Variable ... 588
XML Node Iterator ... 590
Transform XML ... 592
Merge XML Action ... 593
Extract XML Fragment Action ... 594
Edit XML File Action ... 596
Validate XML File ... 598
Delete XML Nodes ... 599
Create XML Node ... 601
23 Build Tools ... 604
Ant Project Action ... 604
Nant Project Action ... 605
MSBuild Project Action ... 606
24 Database ... 607
SQL Server Actions ... 607
Execute SQL Action ... 607
DTSRun Action ... 609
SQL Server Backup Database ... 610
SQL Server Remove Unused Space ... 611
SQL Server Check Database ... 612
SQL Server Check Catalogue ... 612
SQL Server Update DB Statistics ... 613
SQL Server Rebuild Indexes ... 613
SQL Server Best Practices Analyzer ... 613
ADO Execute SQL ... 616
ADO Execute Stored Procedure ... 617
25 Source Code Tools ... 619
CodeHealer ... 619
Pascal Analyzer ... 621
26 Virtualization Systems ... 623
VMWare Workstation ... 623

© 2001-2006 VSoft Technologies Pty Ltd

Start Virtual Machine ... 623
Stop Virtual Machine ... 624
Suspend Virtual Machine ... 625
Reset Virtual Machine ... 625
Check Virtual Machine Status ... 626
Create Snapshot ... 627
Revert To Snapshot ... 628
VMWare Server ... 630
VMWare Server List VMs ... 631
VMWare Server Start / Stop / Reset / Suspend VM ... 632
Microsoft Virtual Server ... 634
Virtual Server List VMs ... 635
Virtual Server Start VM ... 636
Virtual Server Save VM State ... 638
Virtual Server Turn Off VM ... 639
Virtual Server Reset VM ... 640
Virtual Server Pause VM ... 641
Virtual Server Check VM Status ... 642
27 Internet Information Services (IIS) ... 644
IIS 5 ... 645
Server Selection ... 645
Website Selection ... 646
Backup IIS Metabase Action ... 647
Restore IIS Metabase Action ... 648
Start / Stop / Restart WWW Service Actions ... 649
Start / Stop / Pause Website Actions ... 650
Unload IIS Application Action ... 650
Create Virtual Directory Action ... 651
Delete Virtual Directory Action ... 654
Create Web Site Action ... 655
Delete Web Site Action ... 656
IIS 6 ... 657
Server Selection ... 657
Website Selection ... 658
Backup IIS Metabase Action ... 659
Restore IIS Metabase Action ... 661
Start / Stop / Restart WWW Service Action ... 662
Start / Stop / Pause Website Actions ... 663
Unload IIS Application Action ... 663
Create Virtual Directory Action ... 664
Delete Virtual Directory Action ... 668
Create Web Site Action ... 669
Delete Web Site Action ... 670
28 Executable Packers ... 671
UPX Pack Executable ... 671
Part VI Wizards 674
1 Import Borland Project Wizard ... 674
Select Project Group Page ... 675
Select Compiler Version Page ... 676
Common Settings Page ... 677
Version Info Page ... 678
Import Options Page ... 679
Finish Page ... 680
2 Import VB6 Project Group Wizard ... 681
Select Project Group ... 682
Master Project Selection ... 683

© 2001-2006 VSoft Technologies Pty Ltd

Common Output Path Setting 684
Compiler Settings 685
Make Settings 686
Grouping 687
Finish 688

Part VII Automating FinalBuilder 689

1 FinalBuilder IDE 690
Scheduling Builds 690
IDE Command Line Interface 694
Exit Codes 694
2 Command Line version 695

Part VIII Reference 697

1 Regular Expression Reference 697
2 INI Files 704

Part IX Support 705

1 Known Problems 705
2 FAQ 707
3 FinalBuilder Support 708

© 2001-2006 VSoft Technologies Pty Ltd

1 FinalBuilder

1.1 FinalBuilder Overview



Point. Click. Build. Deliver!

Automated Builds with FinalBuilder

What is FinalBuilder™?

FinalBuilder is a powerful Automated Build Tool that allows developers to automate their software build process in a reliable and repeatable manner. FinalBuilder enables you to define your build process easily, allowing you to focus on more interesting and important tasks! Anyone can run the build, with one click (or keystroke) in FinalBuilder. Builds can be scheduled via the Windows scheduler, so automated nightly builds are easy to setup. FinalBuilder saves time, often running tasks in seconds that would take minutes or hours if done manually.

Throw out those complicated Batch Files!

Many developers use dos batch files to automate their builds. These batch files are typically difficult to maintain, have poor error handling, little or no error logging. FinalBuilder is easy to use, so the build process doesn't become the domain of one key person!

How does FinalBuilder Work?

Most development tools such as compilers, install builders, version controls systems etc, 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. FinalBuilder defines Actions, where each action provides an interface to some third party tool or to an internal function. Actions are chained together to create the build process. FinalBuilder also supports Active Scripting, each action exposes events which can be coded in VBScript or JavaScript.

© 2001-2006 VSoft Technologies Pty Ltd

1.2 Version History

Version history is available at:
http://www.FinalBuilder.com/FinalBuilder/versionhistory.html

© 2001-2006 VSoft Technologies Pty Ltd

1.3 License

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

FinalBuilder End User Licence Agreement
Copyright © 2006 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 4 Standard", "FinalBuilder 4 Professional", and "FinalBuilder 4 ActionStudio", herein referred to as the Software.

1. Copyright and Ownership

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

2. Definitions

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

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

3. Grant of License

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

4. License Types

4.1 Named User License: Under the terms of a Named User License, you may install a reasonable number of copies of the Software provided that the specific Named User is the only individual permitted to use the Software. A Named User is a specific individual designated by you to use the Software. A non-human operated device is counted as a Named User if the device utilizes the Software. A Named User License may not be used concurrently on different computers.

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 provided the number of users accessing or using the Software simultaneously does not exceed the Concurrent User Licenses purchased for such use.

4.3 Site License: User 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.

4.4 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.

4.5 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

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: 02 6282 7488
Facsimile: 02 6282 7488
Email: salesinfo@vsoft-tech.com.au

© 2001-2006 VSoft Technologies Pty Ltd

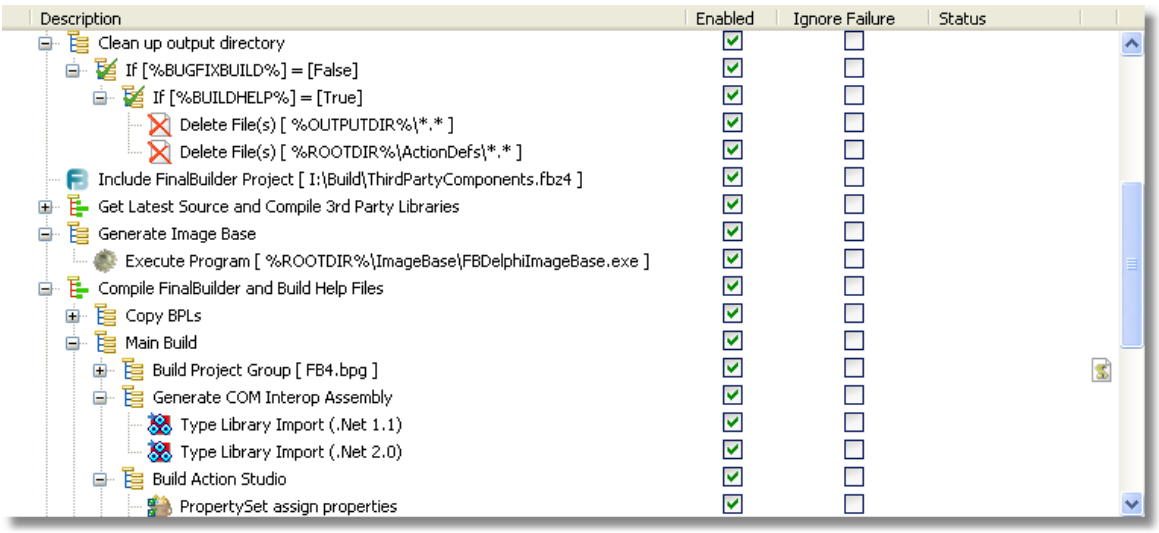
2 Getting Started

2.1 FinalBuilder Basic Concepts

- Use the FinalBuilder IDE to design your build process using Actions (see below)
- Test and Debug your build process in the IDE
- Run your build in the IDE or using FBCMD

Actions and Action Lists

FinalBuilder uses one or more Action Lists which each contain Actions to perform the build process. An Action List contains a sequence of discreet steps called Actions which are executed when the build process runs. An Action can perform tasks such as calling a compiler, copying files, creating directories, etc.



By default each FinalBuilder project has a Main Action List and an OnFailure Action List. The build 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. Action Lists are covered later in this help file.

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

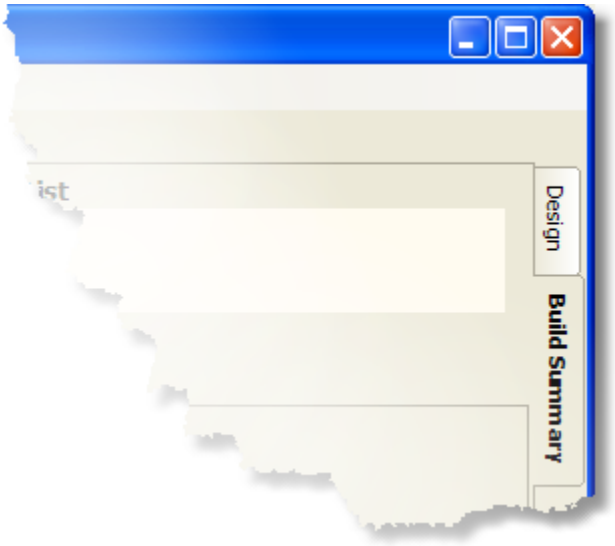
© 2001-2006 VSoft Technologies Pty Ltd

2.2 FinalBuilder IDE Modes

The FinalBuilder IDE supports three main modes of operation:

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

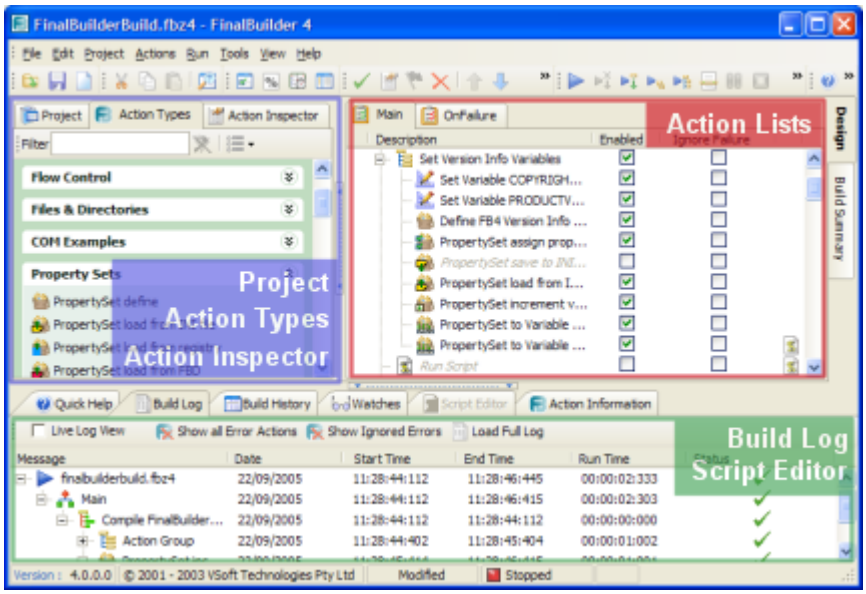
The way these modes are achieved is by various tabs in the FinalBuilder user interface. The tab on the right hand side (see below) changes between "Design" (and Debug) and "Running Status". The tabs at the bottom of the user interface displays Quick Help, Build Log, Build History, Watchs, Script Editor and Messages. Both these sets of tabs are automatically changed depending on the mode (eg. when a build starts), and can be manually overriden.



© 2001-2006 VSoft Technologies Pty Ltd

2.2.1 IDE Design Mode

Design mode consists of three main sections:



Project/Action Types/Action Inspector

This section contains three tabs:

- Project Tree (Contains a tree of the action lists, global script and variables in your project)
- Action Types (Contains the list of available action types that you can use in your build process)
- Action Inspector (A quick way to view and change some of the properties of selected actions, more complex properties are set from the properties dialog available for each action)

Action Lists

This is where you create and edit the actions of your build process. 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.

Build Log/Script Editor

There are many tabs in the bottom section, but the most relevent ones during design of your build process are the Script Editor and the Quick Help.

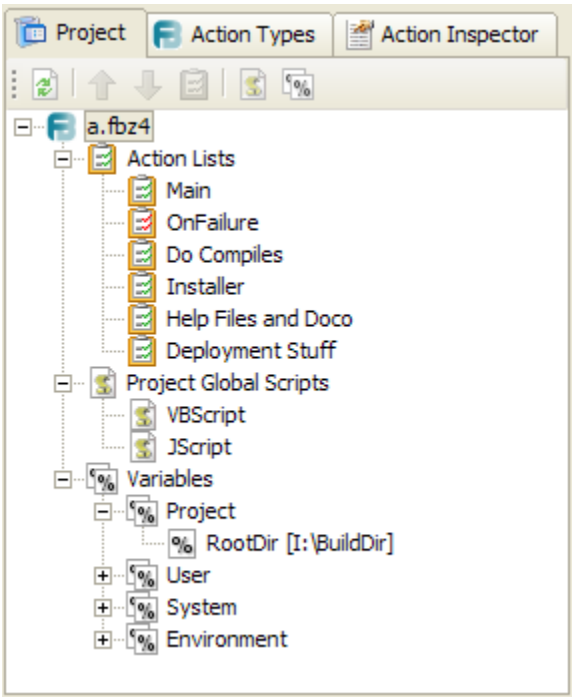
2.2.1.1 Project Tree

The project tree displays the following information:

- The name of your project file
- All the action lists in your project
- Nodes for the Project Global Script

© 2001-2006 VSoft Technologies Pty Ltd

- All the project, user, system and environment variables for your project



The project tab also has the following functionality:

Action Lists
- Add Action List
- Reordering Action Lists (using Ctrl Up, Ctrl Down, toolbar, context menu, and drag and drop)
- Selecting Action List (using context menu, toolbar and double click)

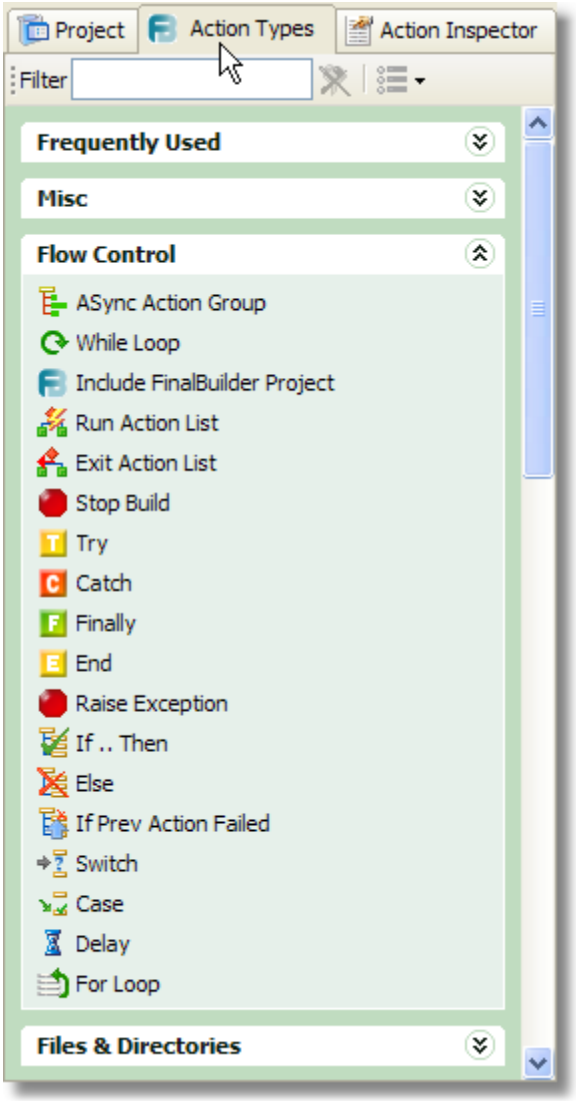
Project Global Scripts
- Open Project Global Script editor (using context menu, toolbar, and double click)

Variables
- Add Variable (context menu)
- Edit Variable (context menu, double click)
- Open Edit Variables dialog (context menu, toolbar)

© 2001-2006 VSoft Technologies Pty Ltd

2.2.1.2 Action Types

The Action Types Panel is what you use to search for actions to add to your build process.



The available Action Types are grouped together 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 Filter edit box at the top of the Action Types Panel. You can use the keyboard shortcut Ctrl+I to set the focus to the filter edit control :



To Cancel the Filter click on the "Clear Filter" button or press Escape :



To Select an Action Type Category quickly, use the Action Categories  drop down menu button and select the category from the menu.

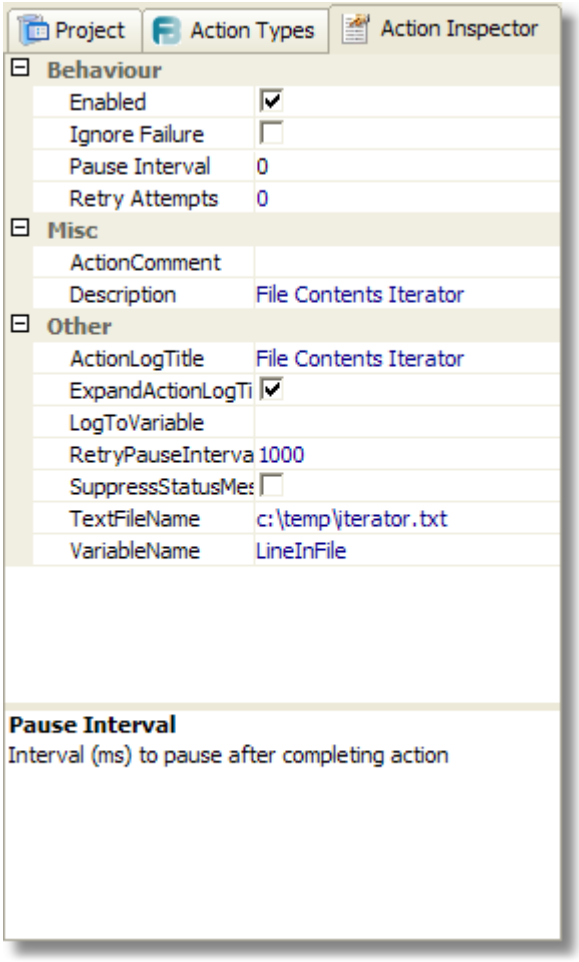
© 2001-2006 VSoft Technologies Pty Ltd

See Also

Action Lists

2.2.1.3 Action Inspector

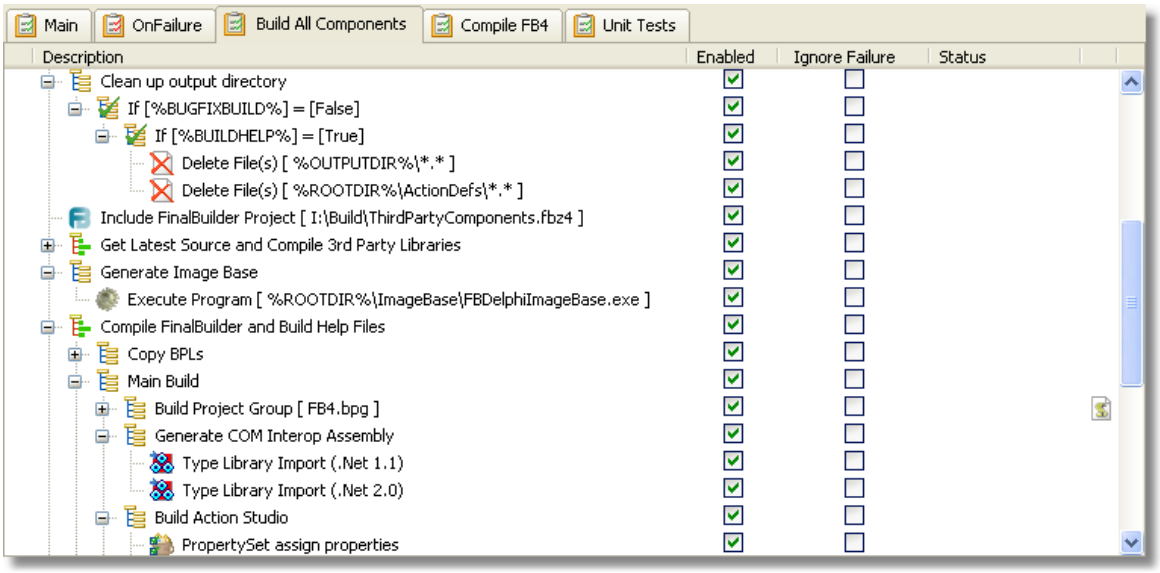
The Action Inspector provides an easy way to view and change some properties of a selected action or multiple selected actions. When an action is selected in the Action List, then the Action Inspector updates with the properties of that action. Any property displayed in the action inspector can be edited. Some actions provide a context sensitive description at the bottom of the Action Inspector.



2.2.1.4 Action Lists

By Default, a new FinalBuilder project has 2 Actions Lists, Main and OnFailure. These default Action Lists cannot be deleted or moved. When a build starts, the first enabled action in the Main Action List is executed, after that if any action fails, the execution switches to the OnFailure Action List (provided OnFailure contains any actions.) The OnFailure Action List allows you to perform cleanup tasks when the build fails for any reason.

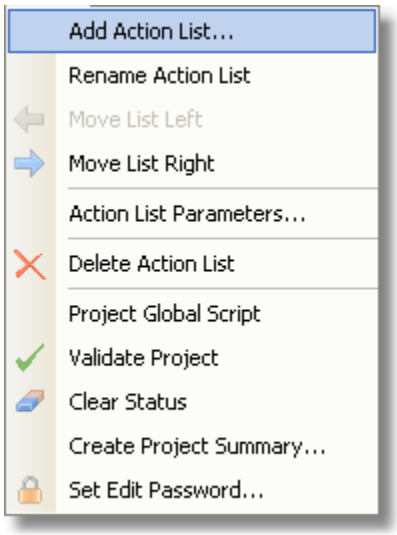
© 2001-2006 VSoft Technologies Pty Ltd



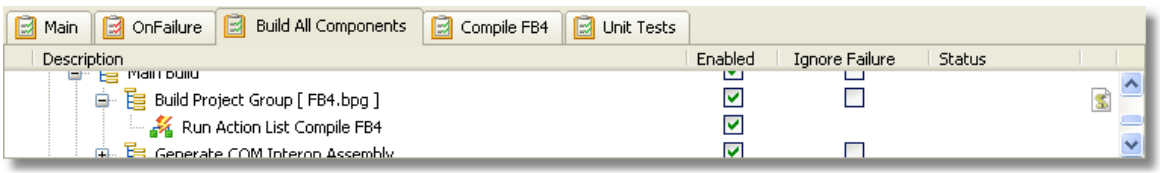
(Tip : You can drag actions from one action list to another by dragging them over the action list tab, then dropping them on the other action list.)

Adding Action Lists

You canAdd, 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 example, the Run Action List in the "Build All Components" action list calls the "Compile FB4" action list.)

© 2001-2006 VSoft Technologies Pty Ltd

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

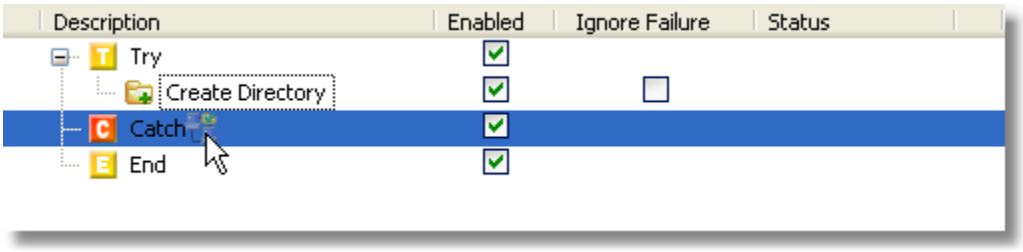
FinalBuilder Variables | Action List Parameters

2.2.1.5 Working with Actions

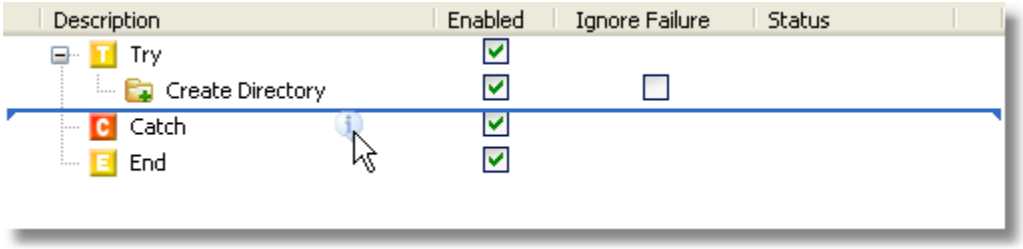
Adding Actions to an Action List

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

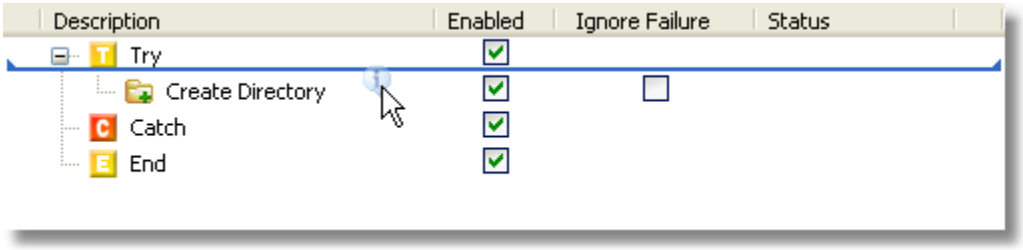
The Action List Tree view provides guide lines to indicate where the action will be dropped. In this example the action will be dropped as a child to the highlighted action, because the mouse is over the icon or name of the action,



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

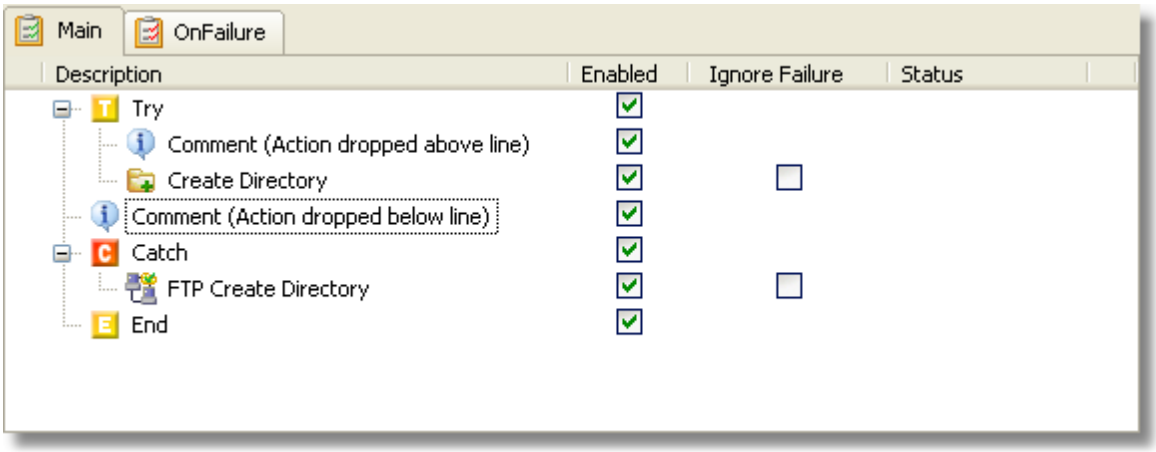


In this example the Action will be dropped After the action above the line, because the guide line indicators point up.



© 2001-2006 VSoft Technologies Pty Ltd

After the above drag n drop operations our action list looks like this :



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.

© 2001-2006 VSoft Technologies Pty Ltd

2.2.1.6 Script Editor

The script editor is where you can write VBScript or JScript 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

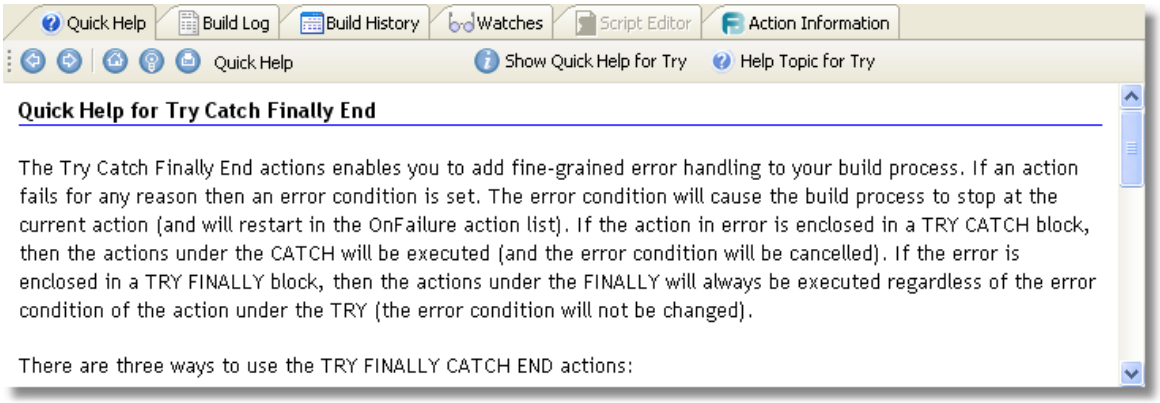
2.2.1.7 Quick Help

The Quick Help page provides Quick Help, Getting Started, Tip of the Day, and FinalBuilder Update information.

Quick Help

Primarily, the Quick Help page provides summary help information for an action. To view the quick help for an action you can either:

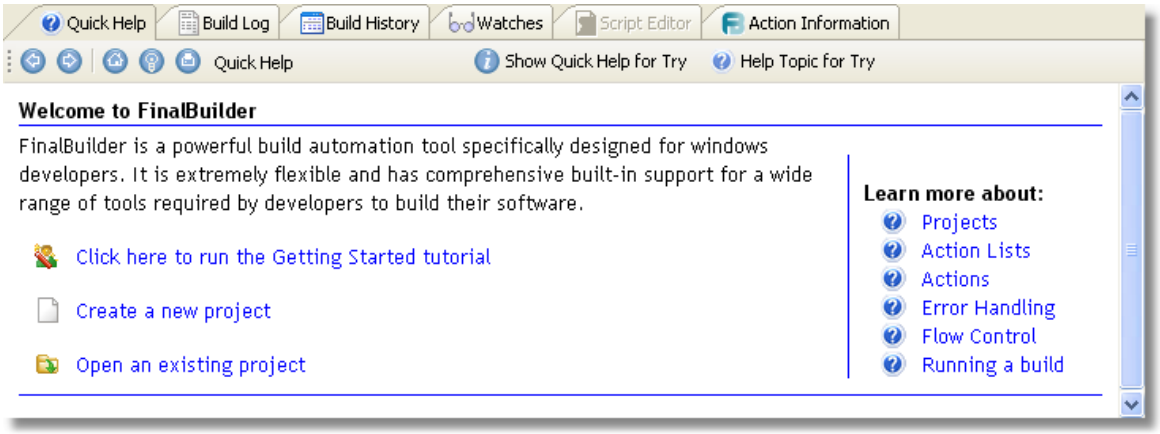
- Select the action from your Action List and:
  - press Alt F1, or
  - right-click and select Quick Help, or
  - click the "Show Quick Help for ..." button on the quick help tab
- Right-Click on an action in Action Types and select "Show Quick Help"



Getting Started

It's second function is to provide a "Getting Started" tutorial for people to learn the basics of FinalBuilder:

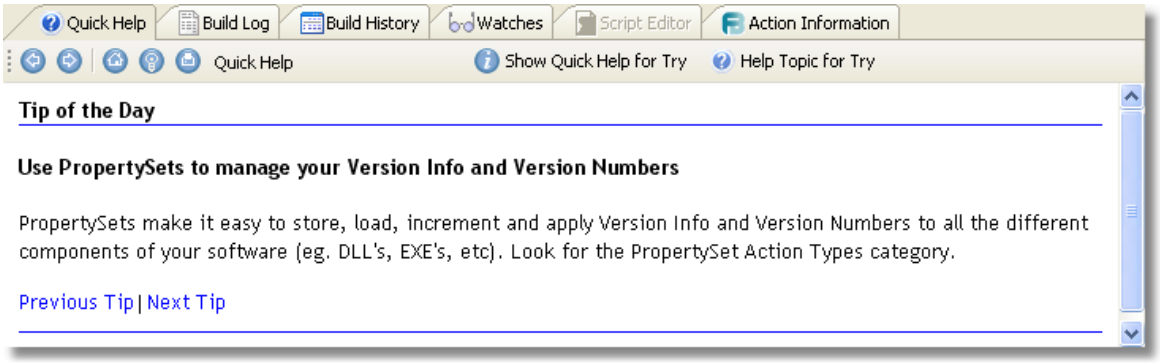
© 2001-2006 VSoft Technologies Pty Ltd



To show the Getting Started / Welcome to FinalBuilder page, click the blue Home button (3rd from the left) in the quick help toolbar.

Tip Of The Day

A new tip of the day loads automatically on each new day. Alternatively, click the blue "Light Bulb" icon in the quick help toolbar.



Check For Updates

The fourth function of Quick Help is to display information about a new version of FinalBuilder. When FinalBuilder loads, it automatically checks for any updates and displays the information in the Quick Help window. You can manually force a check for update by choosing the "Check for Updates" option in the Help menu. More information in topic Check for Updates

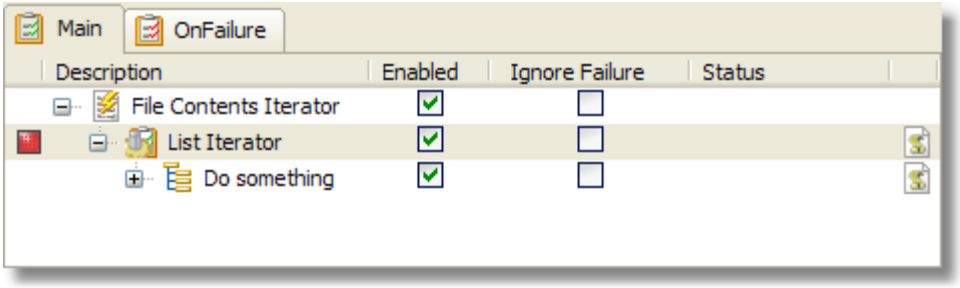
© 2001-2006 VSoft Technologies Pty Ltd

2.2.2 IDE Debug Mode

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

How to Add and Remove Breakpoints

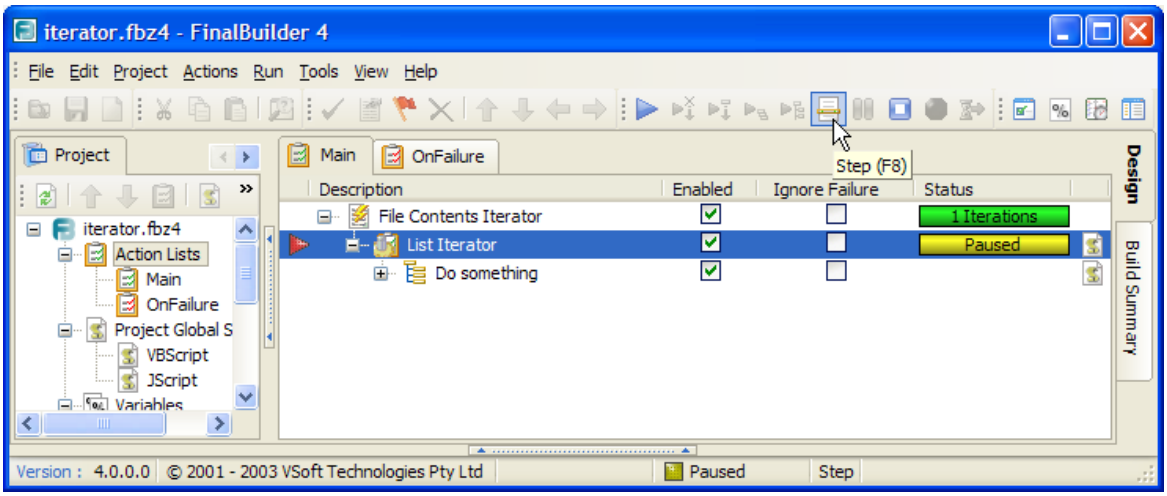
- Right click on an action and choose Breakpoint.
- Press F5
- Choose Breakpoint from the action menu.
- Click in the gutter next to the action



Notice that the List Iterator action has a breakpoint - the build will pause when it reaches the List Iterator action.

How to step through your build process

The following screen shot shows a paused build process - you can step using the toolbar icon, the F8 key or the Run Menu, Step item.



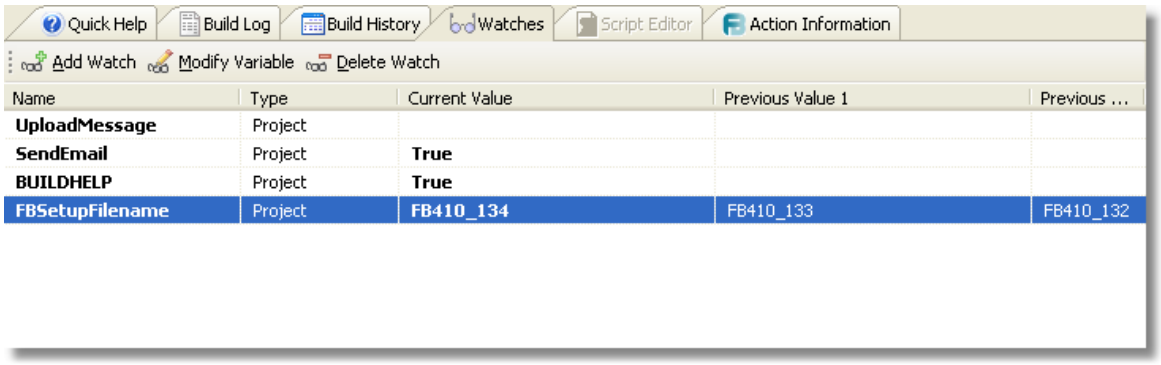
See Watches and Live Logging for more information on debugging projects at runtime.

© 2001-2006 VSoft Technologies Pty Ltd

2.2.2.1 Watches

Watches enables you to watch and modify variable values during a build process.

It is designed as a debugging aid so that you can step over actions and see the current and previous values for the specified variables.



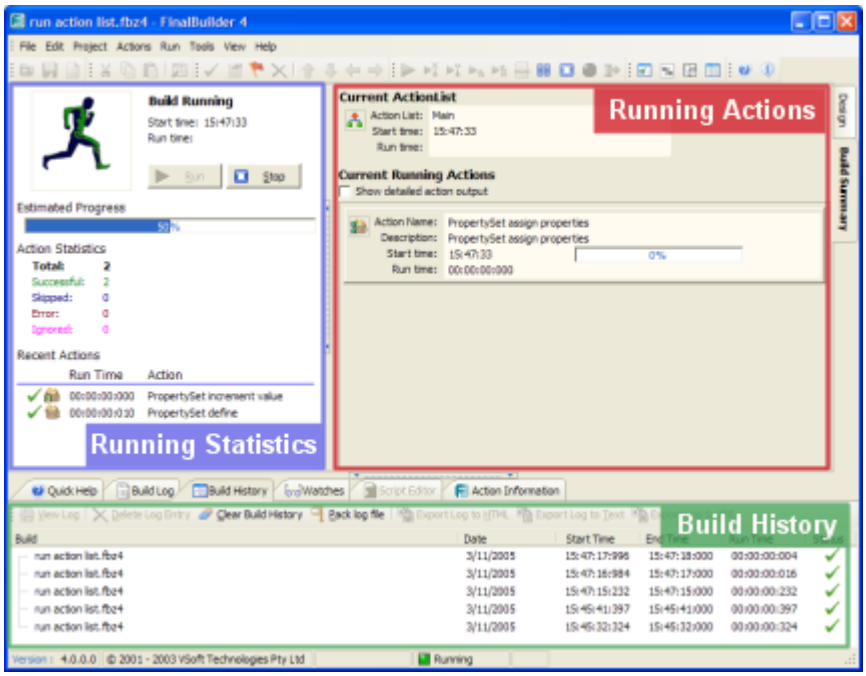
To add a new watch, click on the Add Watch button or drag a variable from the Project tab.

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 build is paused!

© 2001-2006 VSoft Technologies Pty Ltd

2.2.3 IDE Running Mode

Running mode consists of the following sections:



Running Statistics

This section displays the following information:

- A "running man" animation to indicate the status of the build. When the build complete this changes to either a tick or cross depending on the build result.
- Buttons to Run or Stop the build
- Estimated progress of the entire build process.
- 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)

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)

Build History

This section show all the builds in the build archive file.

2.2.3.1 Running a Project

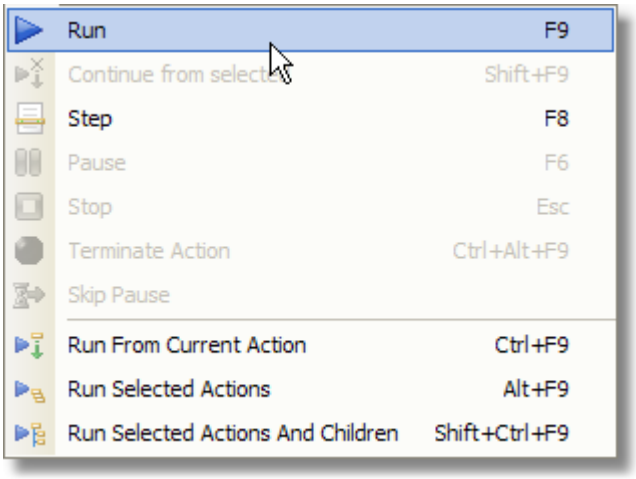
When you have created your build process (and maybe debugged it), you can then start the build in a number of ways:

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

© 2001-2006 VSoft Technologies Pty Ltd

- Restart Build 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 any selected actions and also the 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:

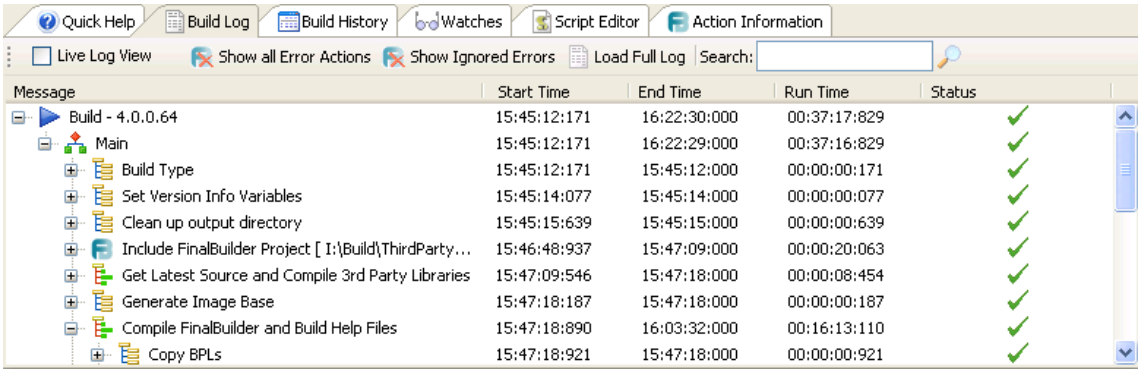
- **Step** - If the build is paused, it will run the next action and pause. If the build hasn't started, it will run the first action in Main and then pause.
- **Pause** - The build will pause after the current action completes.
- **Stop** - The build will stop after the current action completes. Some 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.

© 2001-2006 VSoft Technologies Pty Ltd

- **Continue** - The build will continue running if it's in the paused state

2.2.3.2 Run 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 and 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 Run usually ends when an action has this status.

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.

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, updating the log can be very CPU intensive. You can use the "Live Log View" option in the Build Log tab to disable live logging if the CPU overhead of the live logging is effecting your build process.

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 process. To see the complete hierarchy, it is necessary to wait until the build has completed.

All the log information is still written to the log file, and when the build completes you can open

© 2001-2006 VSoft Technologies Pty Ltd

the most recent build to examine the log.

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.

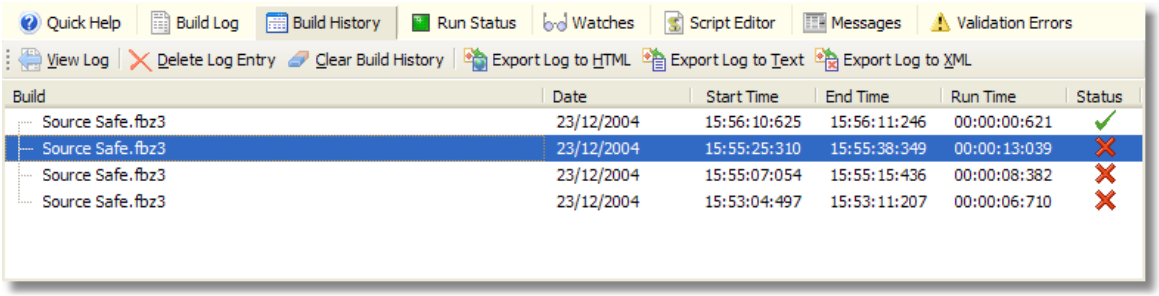
Show Ignored Errors

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

2.2.3.3 Run History

The Build History displays a summary of the previous builds of the current project.

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



This shows the last four builds of the Source Safe project.

The available options are:

View Log - load build details in the Build Log.

Delete Log Entry - removes this build log from the Log History

Clear Build History - removes all the build logs from the Build History

Export Log - Exports the selected build log to HTML, Text, or XML

The Build History only maintains a certain number of build logs, you can change the setting in Tools->Options.

2.2.3.4 Validation Errors

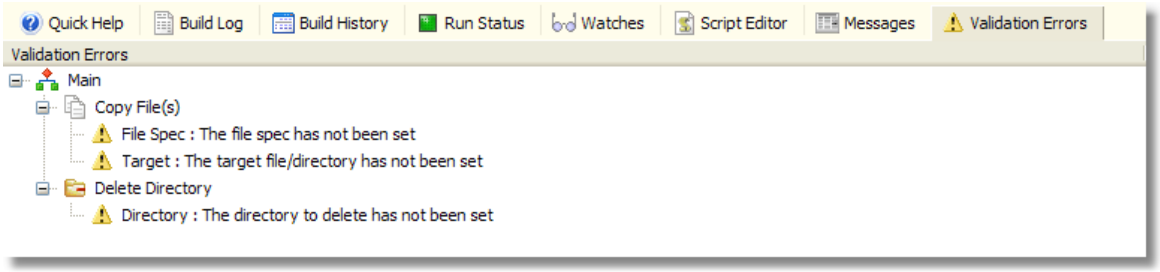
Project Validation checks that all actions in the project appear to have been set up properly. The validation tab displays any validation errors in the current project or the action just edited.

Validation can be triggered in the following ways:

- When a build starts (see Tools->Options->General Options->Variables and Validation)

© 2001-2006 VSoft Technologies Pty Ltd

- After an action has been edited with the property dialog
- Manually (Project->Validate Project)

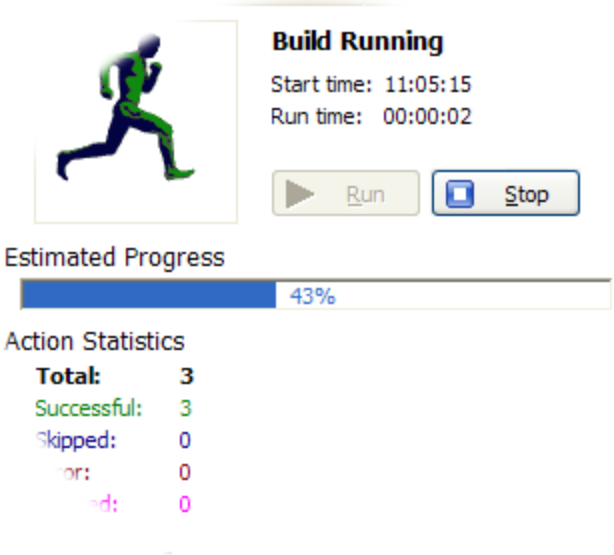


Double-Clicking on the validation errors will highlight the action in your project with the validation error.

The validation does not prevent the project from being saved, but will prevent the project from being Run. You can turn off validation before a project being run in Tools->Options->General Options->Variables and Validation

2.2.3.5 Estimated Progress

Estimated Progress is a progress bar visible in the Build Summary view that increments by one every time an action completes while your build runs, and the total is dynamically calculated. This is what the progress bar looks like during a build:



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

1. When a build 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 build. If there was a last successful build, then the amount of actions that executed in that build is used to set the total.
3. If there was no last successful build, then FinalBuilder calculates how many actions are in your build process and uses that as the total.

© 2001-2006 VSoft Technologies Pty Ltd

For some build process, the above methods to calculate the progress may not be adequate. For example, a build process may have two modes (eg. Full and BugFix). The two different modes may have very different progress totals, and using the last successful build action count method will only work some of the time. The build process itself can set the estimated total using a script method "SetEstimatedProgressTotal" which takes a single integer parameter which allows the progress total to be set to any value at any time during the build process.

An example how to use the "SetEstimatedProgressTotal" script method.

Using the same example above - your build process has Full and BugFix modes. You know (because you've run a Full build enough times) that a Full build will run 469 actions, and that a BugFix build runs only 210 actions. Your build process will have some logic at the start which figures out if a Full build or BugFix build is required, and using that same logic it can call the script function for the full build (in JScript):

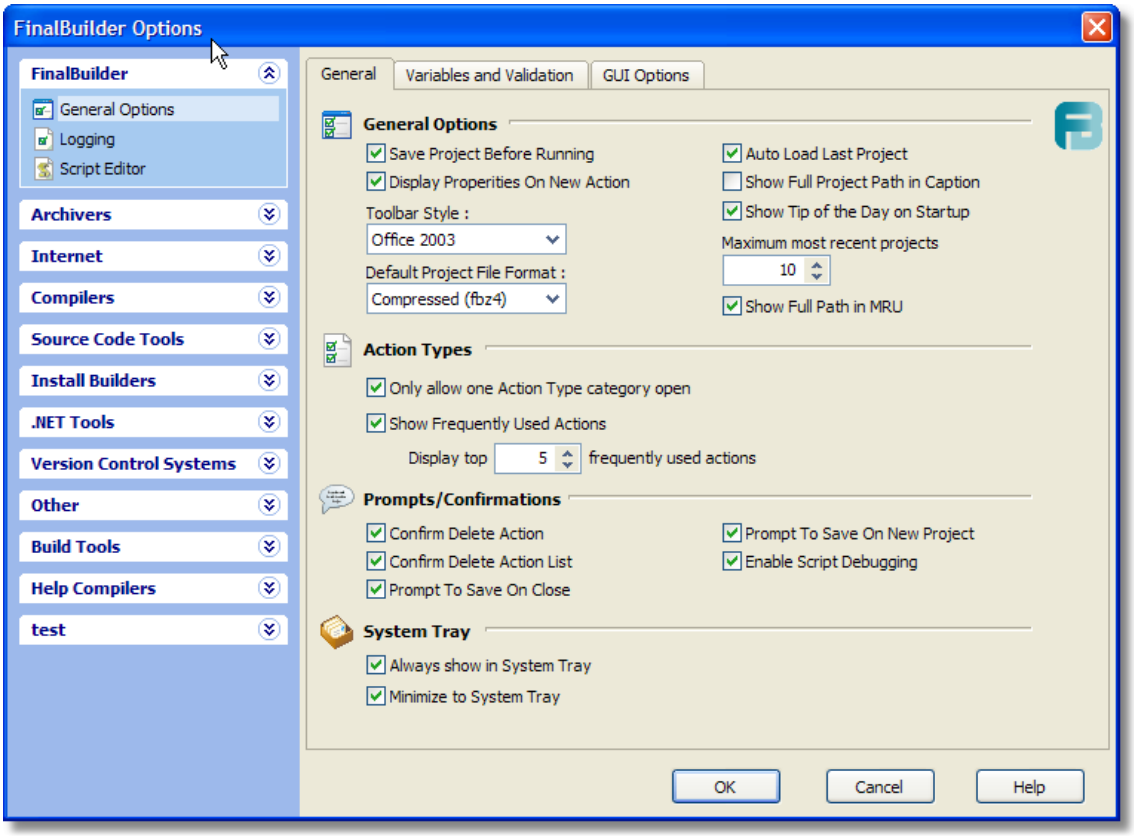
```
if (FullBuild){
  SetEstimatedProgressTotal(469)
} else {
  SetEstimatedProgressTotal(210)
}
```

This gives you complete control over the estimated progress, and it can be set at any time during your build.

© 2001-2006 VSoft Technologies Pty Ltd

2.3 Options Dialog

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

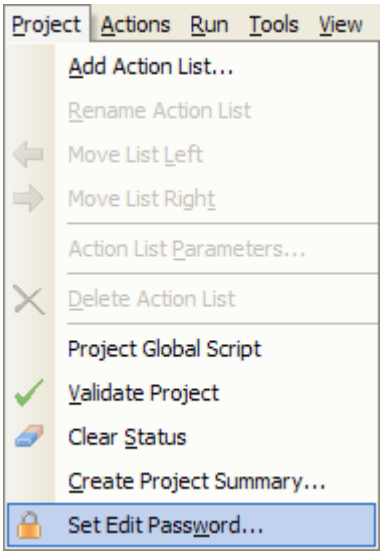


© 2001-2006 VSoft Technologies Pty Ltd

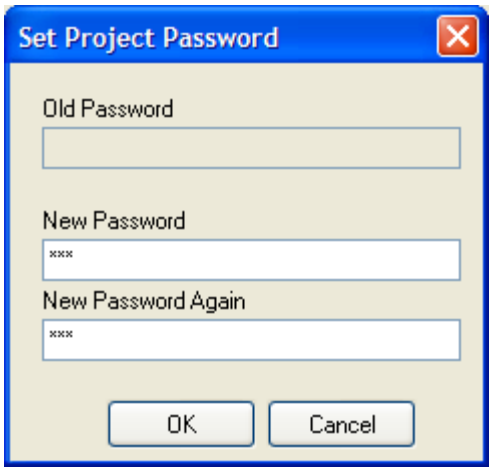
2.4 Project Edit Password

The Edit Password for a project allows the owner of the project to set a password which is required to view and edit the steps of the build process 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:

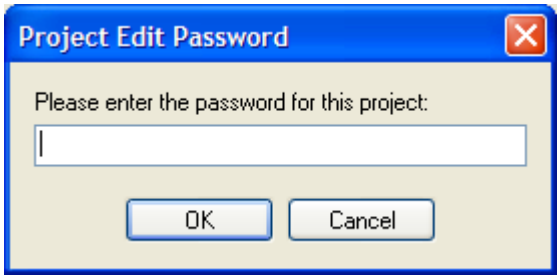


This will bring up the Project Edit Password dialog - type in the password required to edit the project. To set no password for the project, simply leave the new password fields blank.



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

© 2001-2006 VSoft Technologies Pty Ltd



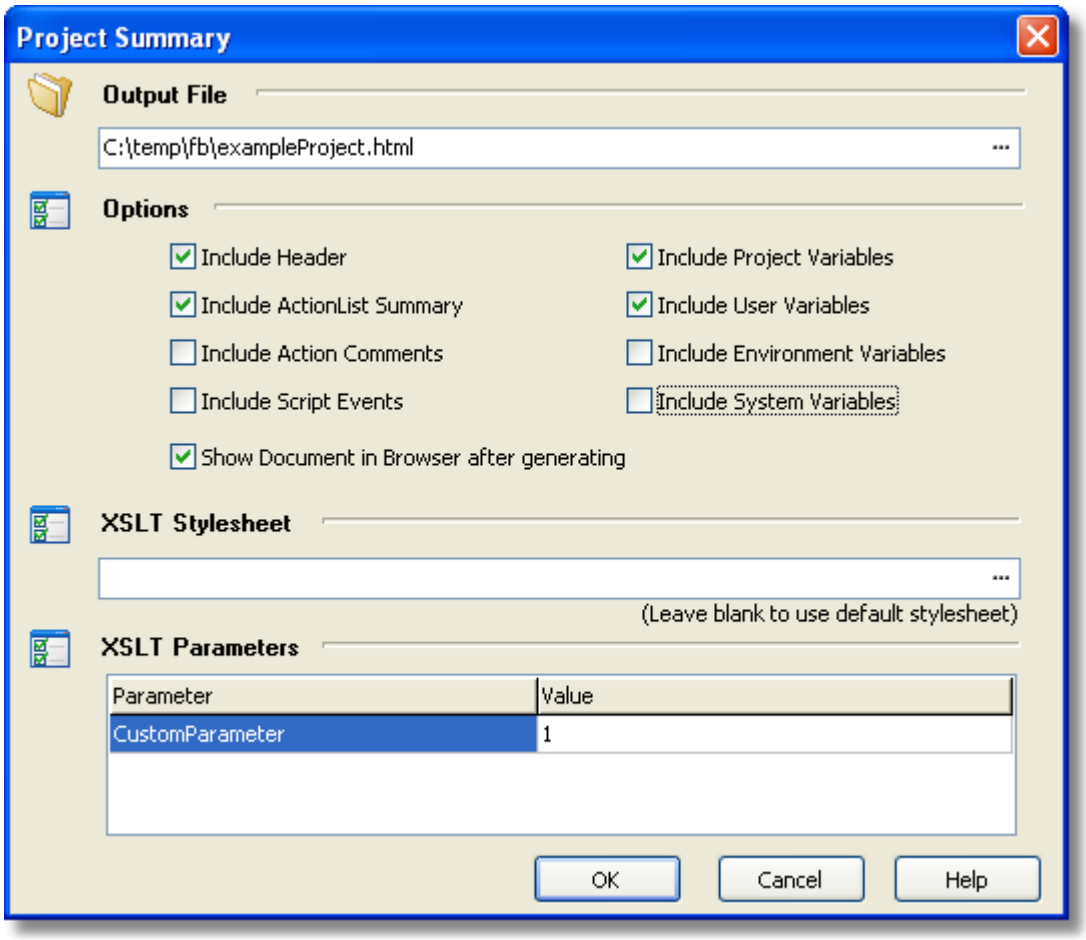
One of the limitations with project edit passwords is that the project file must be saved as a compressed project file (.fbz4) - as this prevents the encrypted password in the project file being removed easily.

© 2001-2006 VSoft Technologies Pty Ltd

2.5 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 (fbp2_2_html.xsl) will be used.

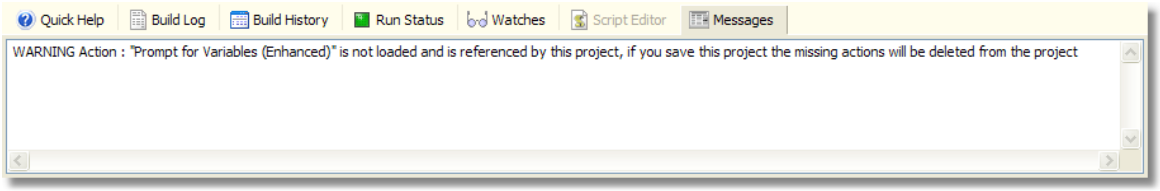
The XSLT Parameters allow you to pass custom parameters to your stylesheet.

© 2001-2006 VSoft Technologies Pty Ltd

2.6 Messages

The Messages tab displays 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:



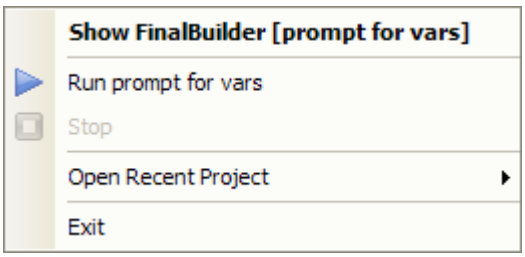
© 2001-2006 VSoft Technologies Pty Ltd

2.7 Tray Icon

FinalBuilder can optionally show in the System Tray:



The FB 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 process.

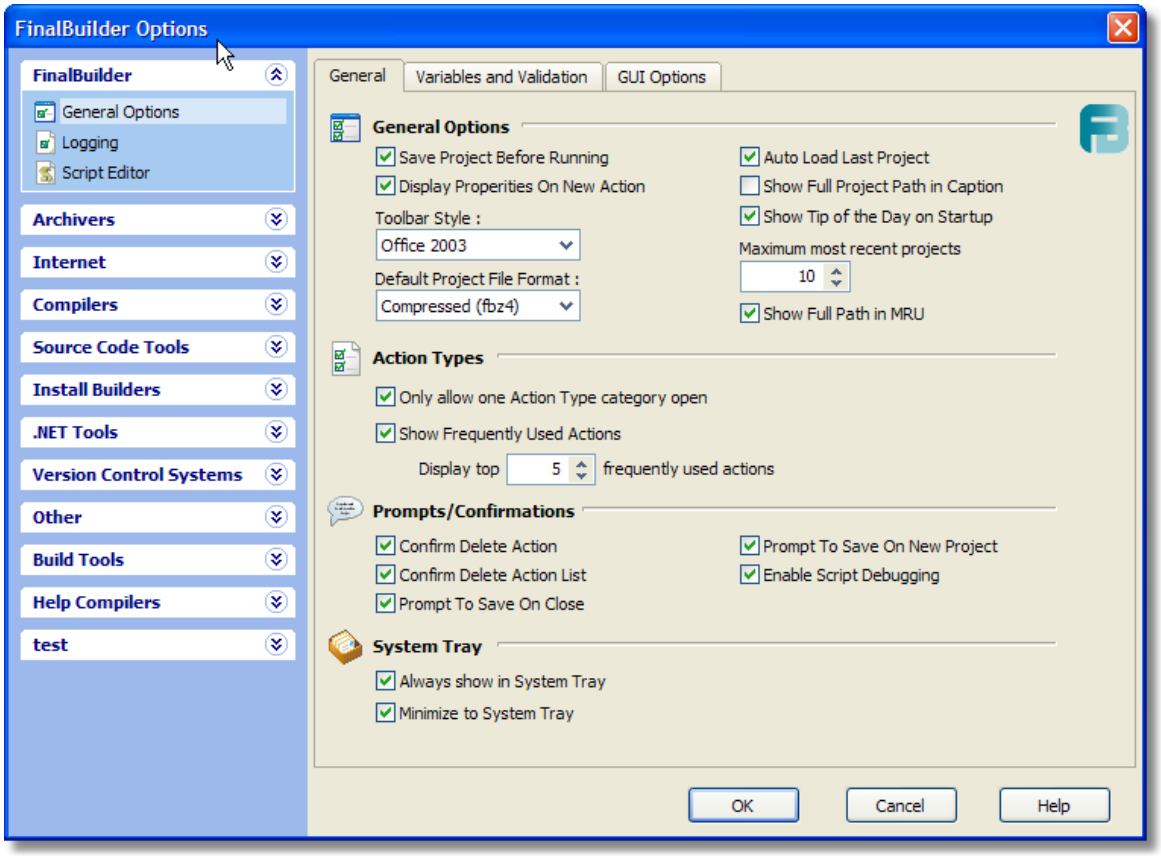
Stop - Stops the build process

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 FB System Tray settings:

© 2001-2006 VSoft Technologies Pty Ltd



System Tray:

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

Minimize to System Tray - When FB is minimised FB will not show on the task bar or the task manager, but will show on the system tray.

The Tray Icon will also display the state of the build process:

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

© 2001-2006 VSoft Technologies Pty Ltd

2.8 Check for Updates

Once a month FinalBuilder will prompt the user if they would like 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.

© 2001-2006 VSoft Technologies Pty Ltd

2.9 Project Files & Other Files

FinalBuilder can use one of two different file formats for the project file.

<project>.fbz4

Compressed FinalBuilder project file. This is the default format for a project file. The FBZ4 file is a zipped FBP4 file, so you can use any standard zip tool to uncompress it.

<project>.fbp4

Uncompressed FinalBuilder project file. Project files are standard XML format and so are quite verbose. It is recommended to use the compressed project file format for large projects.

Auxillary 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>.fblz

The FBLZ file is the log archive file. It records the logs of any previous builds, up to the "Log History Count" option (default is 4). If you put the log archive file under version control make sure it is not read only when you run your builds, otherwise the log file won't be written to.

<project>.fbw

The FBW file is the watches file (ie. the variable watches you have defined), you do not really need to put this file into version control as it's only needed when debugging projects.

<project>.fbd

The FBD file contains information specific to the project but may change during a build process. For example, if a compiler action supports auto incrementing the version information, then the version information will be stored in the FBD file so that the project file doesn't need to be writable during the build process. Please ensure this file is not read only.

<project>.lck

The lck 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 lck file is deleted when a project is closed, or FinalBuilder closes.

© 2001-2006 VSoft Technologies Pty Ltd

2.10 FinalBuilder Plugins

FinalBuilder Plugins extend the functionality of FinalBuilder. Plugins are created in action packages, which can be enabled and disabled from the Package Manager.

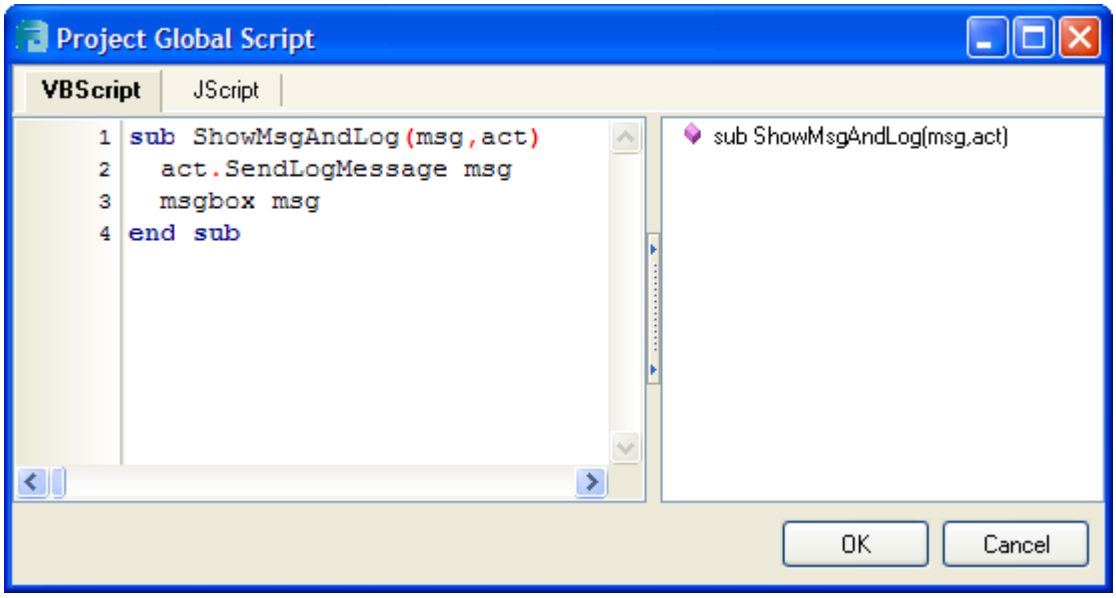
For more information on plugins see the integrated help inside FinalBuilder ActionStudio and also the ActionStudioManual.pdf file, which can be found in the FinalBuilder directory.

© 2001-2006 VSoft Technologies Pty Ltd

2.11 Project Global Script

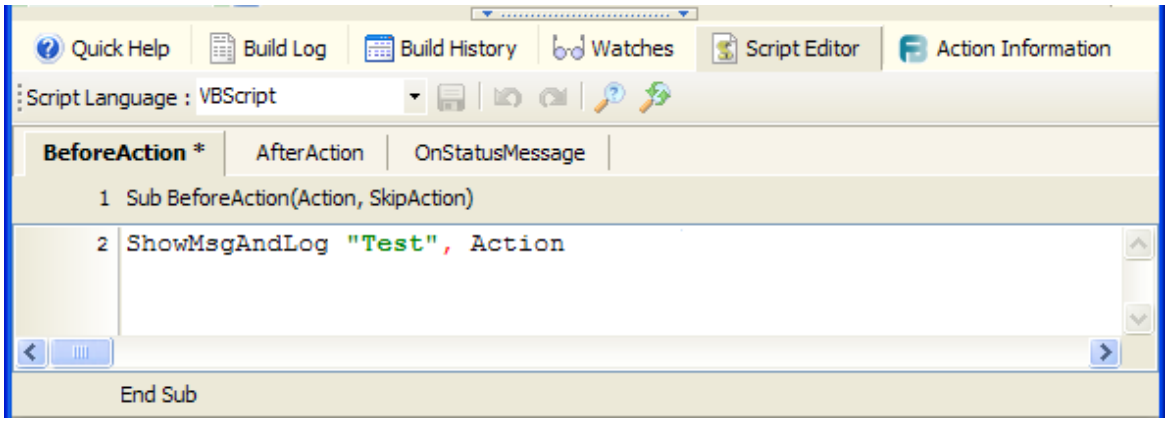
FinalBuilder allows you to have global VBScript and JScript stored in your project file and the script code is available to all actions in the project (in Action Events).

For example, define some global VBScript (access this dialog from the Project menu):



This dialog supports syntax highlighting, code completion, parameter hints and also will display subs, functions, constants, etc 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).

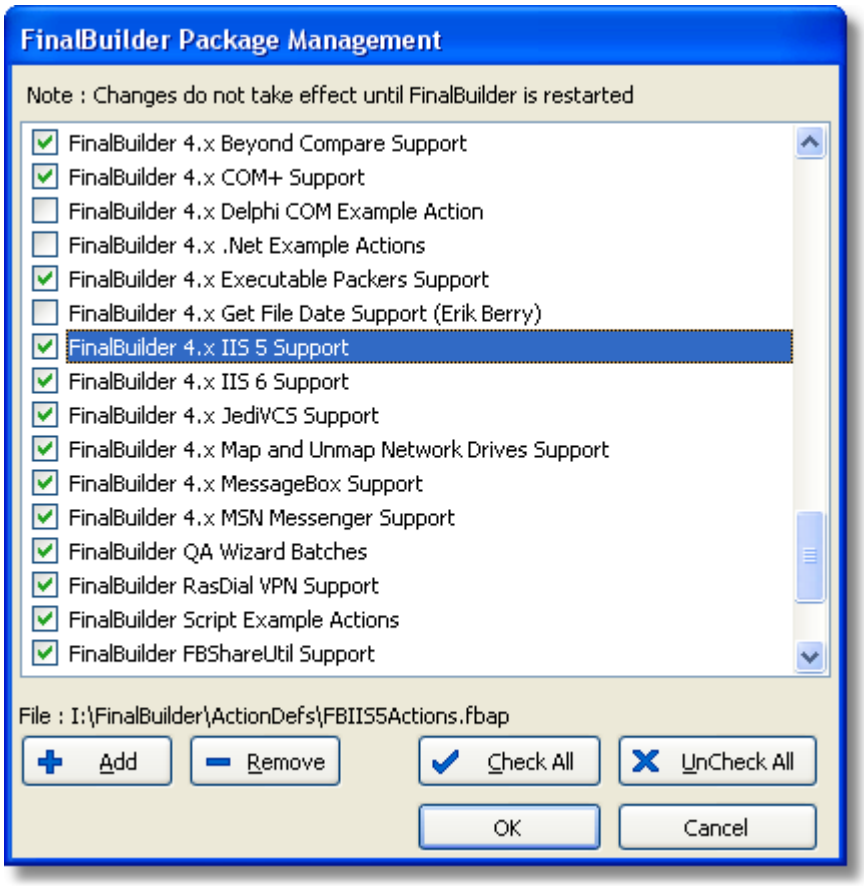
And then you can use the script as you would a normal function, sub, etc.



© 2001-2006 VSoft Technologies Pty Ltd

2.12 Package Manager

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



Only checked packages are loaded at startup. 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 plugin packages created with ActionStudio or downloaded from the contributed actions 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.)

© 2001-2006 VSoft Technologies Pty Ltd

© 2001-2006 VSoft Technologies Pty Ltd

3 Scripting

3.1 Scripting in FinalBuilder

Active Scripting

FinalBuilder Supports Active Scripting using the VBScript and JScript languages that is installed on your machine.

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).

What Can I do in the Scripts

FinalBuilder scripts can make use of almost all the functionality of Active Scripting, including external COM/ActiveX calls.

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.

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 Action scripts, by inserting a comment with "USEUNIT scriptfilename"

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"

© 2001-2006 VSoft Technologies Pty Ltd

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/

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

© 2001-2006 VSoft Technologies Pty Ltd

3.2 Global Script functions

Global Scripting Functions

Apart from standard VBScript/JavaScript ActiveScript 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 is a no-op when FinalBuilder is run from the scheduler.

function GetClipBoardText : string;
Gets the text currently on the clipboard
GetClipBoardText is a no-op when FinalBuilder is run from the scheduler.

procedure CopyToClipBoard(const value : string)
Copies the string to the clipboard
CopyToClipBoard is a no-op 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.

© 2001-2006 VSoft Technologies Pty Ltd

function ExcludeTrailingPathDelimiter(value : string) : string;
Removes a trailing path delimiter from the specifed directory if it exists.

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

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

function GetCurrentDir : string;
Returns the current working directory

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

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

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

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

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

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

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

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

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

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

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

function RunProcess...

Deprecated. Allows the user to launch an external process and obtain a result code when it exits.

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

3.2.1 Format DateTime Formatting Options

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

Specifier Displays

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

© 2001-2006 VSoft Technologies Pty Ltd

contents of the TimePMString global variable for any hour after noon.
/ Displays the date separator character given by the DateSeparator global variable.
: Displays the time separator character given by the TimeSeparator global variable.
'xx'/"xx" Characters enclosed in single or double quotes are displayed as-is, and do not affect formatting.

Example usages of the FBFormatDateTime function (in VBScript)

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

© 2001-2006 VSoft Technologies Pty Ltd

3.2.2 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.

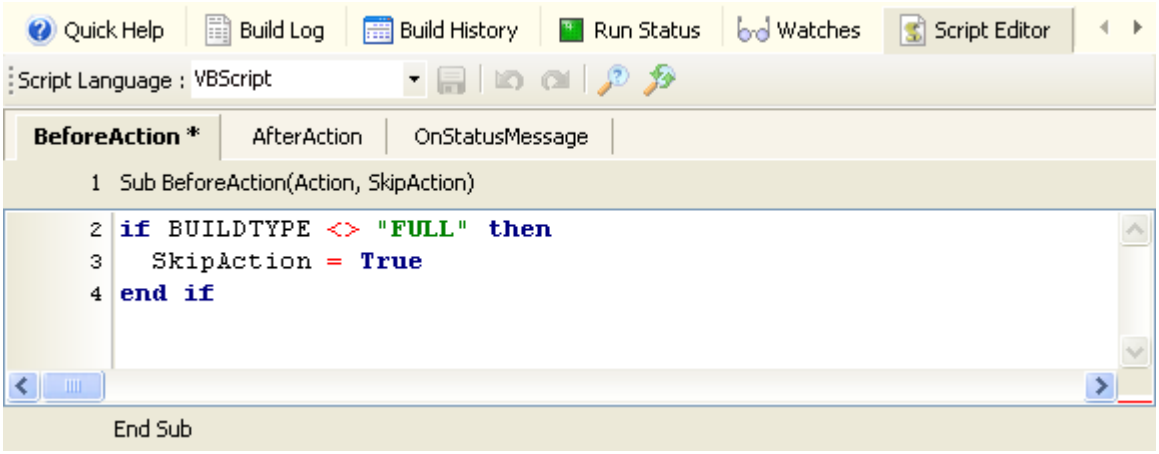
```
if MessageBox("Do you want to cancel the build", "Cancel Build", mbOKCANCEL) =
mrOK then
  // build needs to stop
end if
```

© 2001-2006 VSoft Technologies Pty Ltd

3.3 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 process.

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).

© 2001-2006 VSoft Technologies Pty Ltd

© 2001-2006 VSoft Technologies Pty Ltd

3.3.1 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 OnStatusMessge event!

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

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

function **ExpandExpression**(expr As String) As String
This expands the string passed in by substituting 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
```

© 2001-2006 VSoft Technologies Pty Ltd

```
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

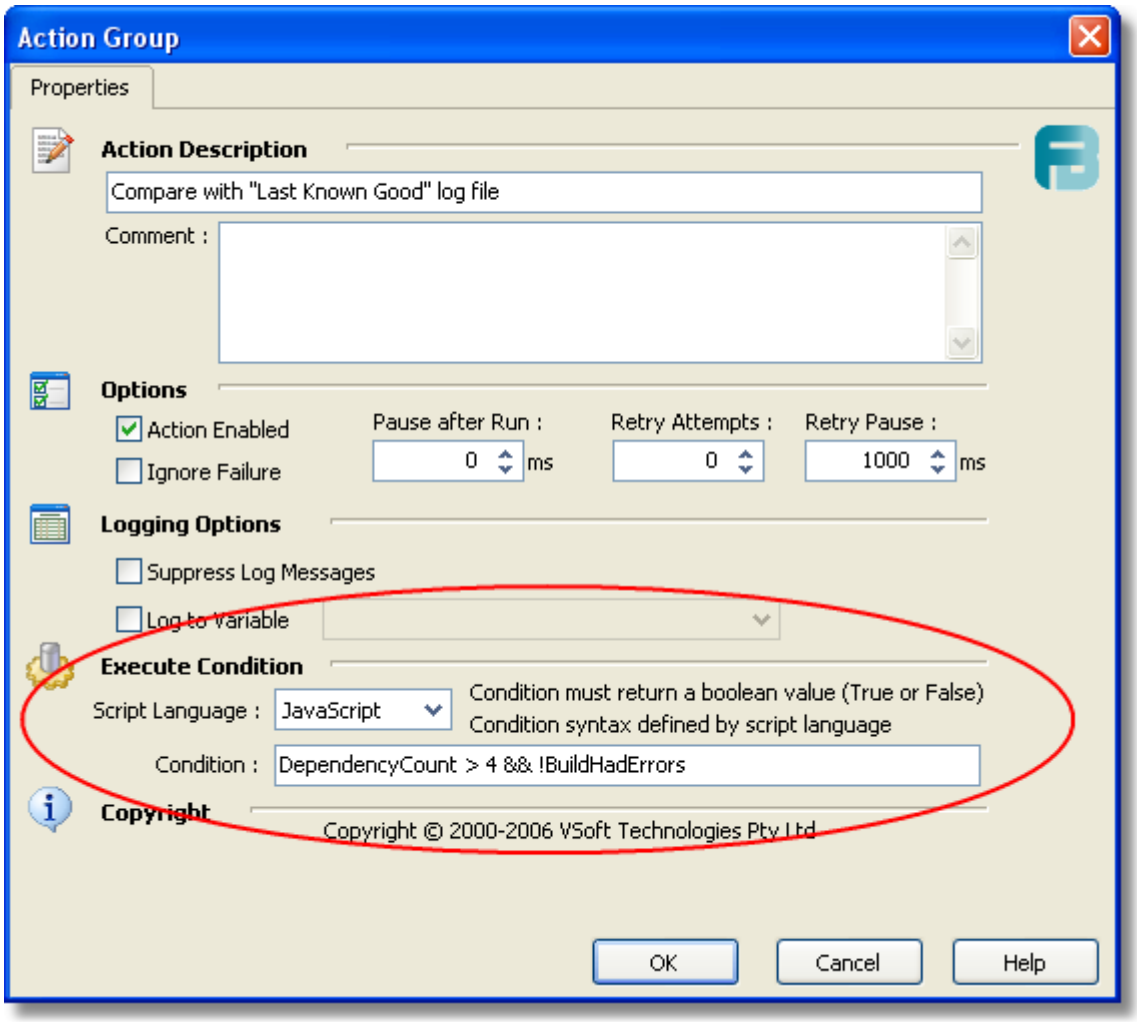
property: **PauseInterval** As Cardinal
Provides access to the Action's PauseInterval property

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

© 2001-2006 VSoft Technologies Pty Ltd

3.3.2 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

When using FinalBuilder variables in an Execute Condition condition, you do not need to specify the %'s. eg. if you have a FB variable "BuildHadErrors" then you can simply specify BuildHadErrors and not %BuildHadErrors%. In the example above, previous actions should have set the BuildHadErrors variable to either True or False, and the DependenyCount to an integer value, all depending on certain conditions.

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

(VBScript)
BuildType = "Full"

(JScript)
BuildType == "Full"

© 2001-2006 VSoft Technologies Pty Ltd

where BuildType is a FinalBuilder variable.

© 2001-2006 VSoft Technologies Pty Ltd

3.4 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. Note that these modified settings are not saved and will be lost when you close FinalBuilder, unless you choose Tools -> Options and click OK before closing.

The options objects are accessible using the GetOptionsObject script function. This function takes the name of the options object as a parameter. The name is the same as appears in the Options dialog. For example to access the Delphi options you would use this code (VBScript example) :

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

Listed below are the available properties on the 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
```

Delphi Options

Name : Borland Delphi

© 2001-2006 VSoft Technologies Pty Ltd

property D3LibraryPath : string
property D4LibraryPath : string
property D5LibraryPath : string
property D6LibraryPath : string
property BCB3LibraryPath : string
property BCB4LibraryPath : string
property BCB5LibraryPath : string
property BCB6LibraryPath : string
property UseRegForLibraryPath : Boolean

Inno Setup Options

Name : Inno Setup

property InnoDLL : string

Help & Manual 3 Options

Name : Help & Manual

property HM3Location: string;

Wise Installer Options

Name : Wise Installer

property WiseInstaller : string
property WiseWindowsInstaller : string

Doc-O-Matic Options

Name : Doc-O-Matic

property DocomaticExe: string

Help Compiler Options

Name : Microsoft Help

property HCRTFLocation : string
property HHCLocation : string

InstallShield Options

Category : Install Builders
Name : Installshield Pro

property CompilerLocation : string
property IFXLocation : string
property ISRTLocation : string
property IncludePath : string
property ISBuildLocation : string
property ISCmdBldLocation : string

GPInstall Options

© 2001-2006 VSoft Technologies Pty Ltd

Name : GP-Install

property GPBuilderLocation: string

ASProtect Options

Name : ASProtect

property ASProtectLocation: string

Armadillo Options

Name : Armadillo

property ArmadilloLocation:string

InternetOptions

Name : Internet

```
property FTPServer : string;
property FTPUser : string;
property FTPPass : string;
property FTPPort : integer ;

property SMTPHost : string;
property SMTPPort : integer;
property SMTPAuth : boolean;
property SMTPEHLO : boolean;
property SMTPUser : string;
property SMTPPass : string;
```

© 2001-2006 VSoft Technologies Pty Ltd

3.5 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>
```

© 2001-2006 VSoft Technologies Pty Ltd

4 Variables

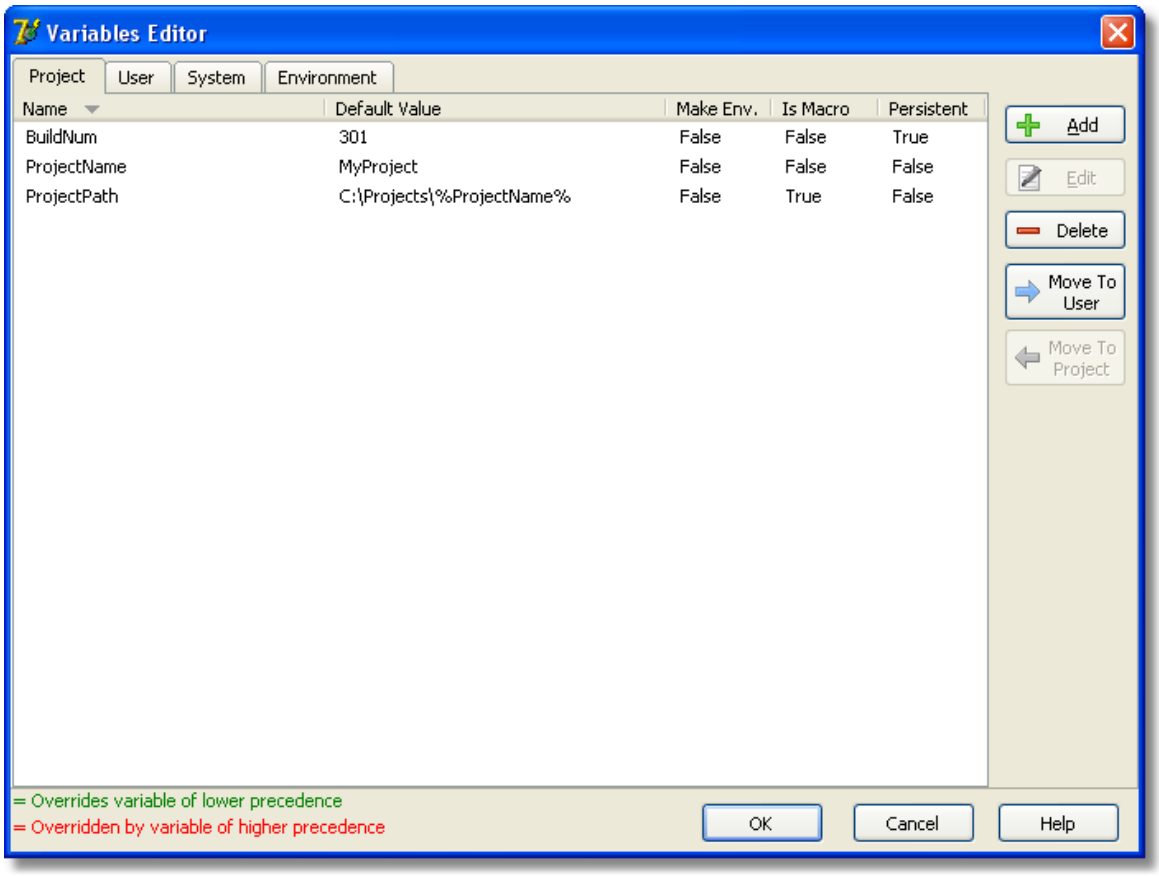
4.1 Variables Overview

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 four types of variables available: Project, User, System and Environment. System and Environment variables cannot be edited in any way from FinalBuilder. Project variables are specific to the current project, and User variables are available to the user in any project, but are specific to the current user.

Variables can be modified at run time either by using 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 scripts in the same way as normal (eg. VBScript or JScript) variables. Only Project and User variables can be modified.



Make Env.
Project and User Variables can be made available as environment variables to applications that are executed by FinalBuilder.

Is 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 - it's value cannot be set during the build process using the Set Variable action or any other means.

© 2001-2006 VSoft Technologies Pty Ltd

In the above example, the ProjectPath variable will evaluate the ProjectName variable whenever it is accessed.

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 .fbv files. Each fbv 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 .fbv file.)

See Also

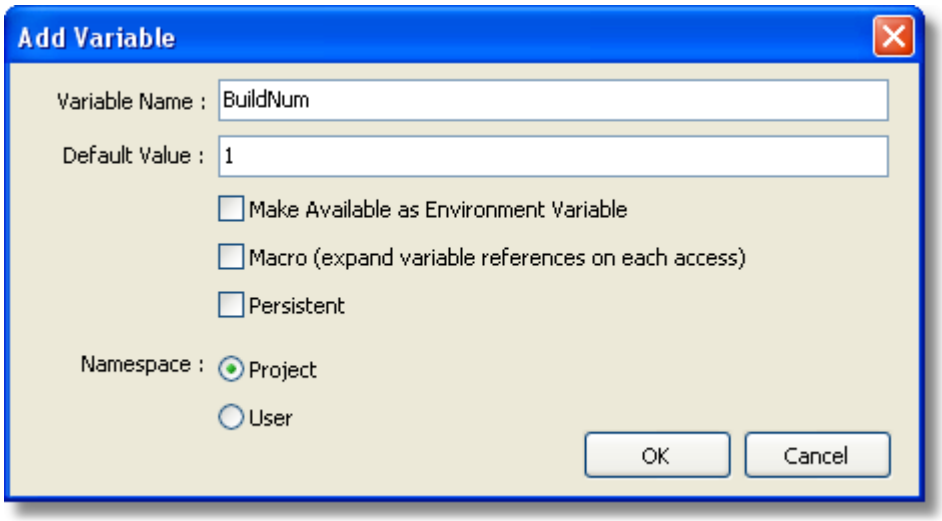
Using Variables
Project Variables
User Variables
Environment Variables
System Variables
Action List Parameters

© 2001-2006 VSoft Technologies Pty Ltd

4.2 Adding Variables

Variables can be added from either the Variables Dialog (Tools Menu -> Edit Variables), or from the Project Tree.

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



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

Default Value
The variable will be set to this default value when a build is started

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 varable 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.

© 2001-2006 VSoft Technologies Pty Ltd

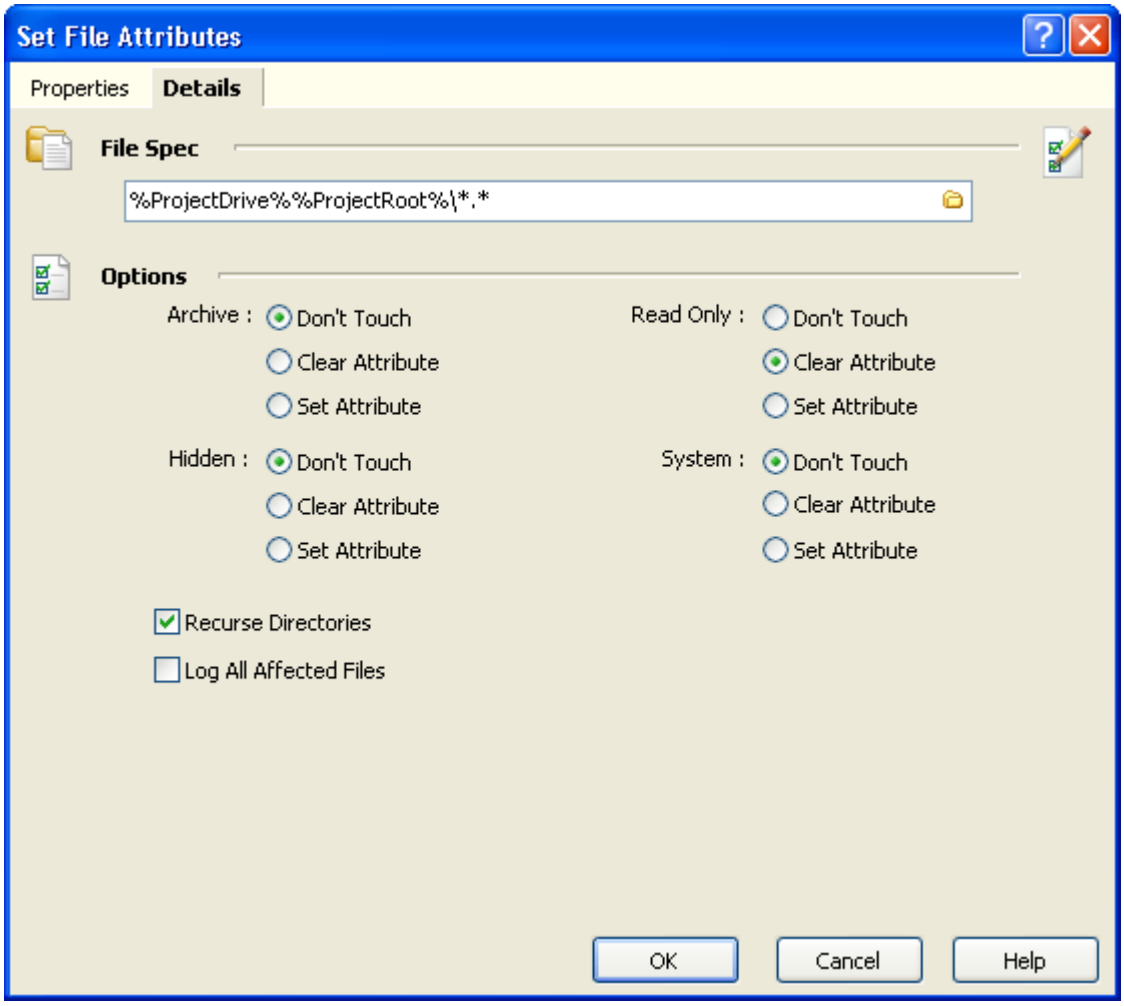
4.3 Using Variables

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

eg. : %BUILDOUTPUTDIR%

If you need to use a % symbol in a property, make sure you escape it with another one, eg. %%

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.

© 2001-2006 VSoft Technologies Pty Ltd

4.4 Project Variables

Project Variables are stored in the project file. This makes them available on any machine where the project is run.

© 2001-2006 VSoft Technologies Pty Ltd

4.5 User Variables

User Variables are stored in an INI file in the following location:

\Documents and Settings\<username>\Application Data\FinalBuilder4\FBUserVariables.ini

They are useful for defining variables that will be used in multiple projects but are specific to a user.

© 2001-2006 VSoft Technologies Pty Ltd

4.6 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.

© 2001-2006 VSoft Technologies Pty Ltd

4.7 System Variables

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

These System 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 System 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.

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
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)

© 2001-2006 VSoft Technologies Pty Ltd

4.8 Action List Parameters

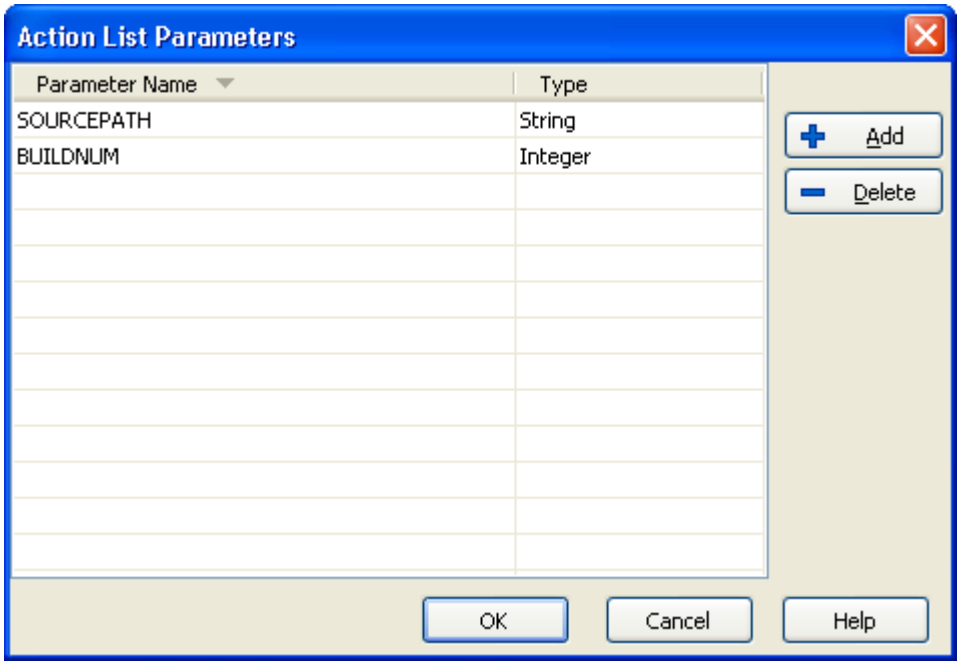
Action List Parameters are designed to allow you to call an action list with a different set parameters.

The scope of an action list parameter is confined to the actions within the Action List that defines the parameter. 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.

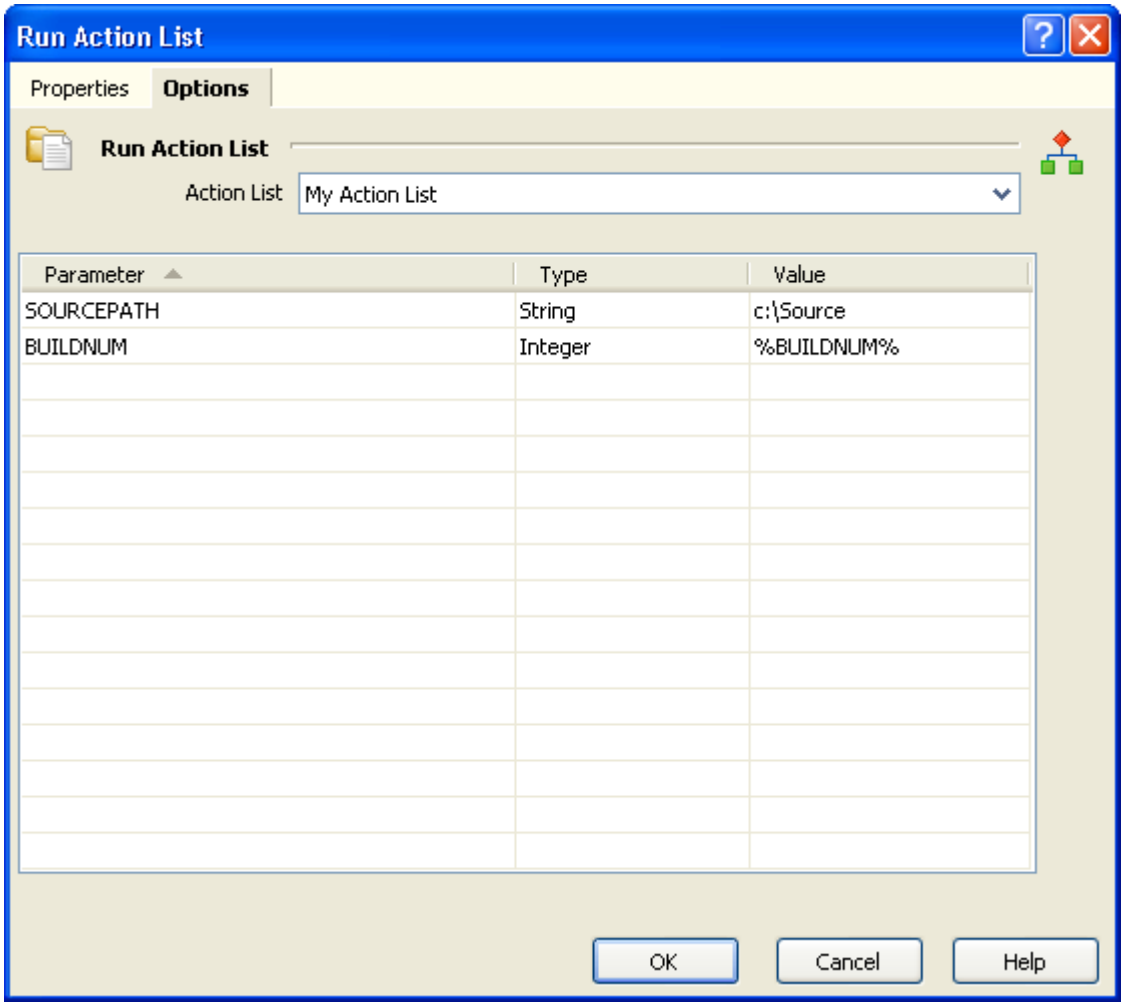
To view or modify action list parameters you can either:
1. Right-click on the action list tab and select "Action List Parameters...", or
2. Select the "Action List Parameters..." menu item from the Project menu.



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:

© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5 Actions Reference

What are Actions?

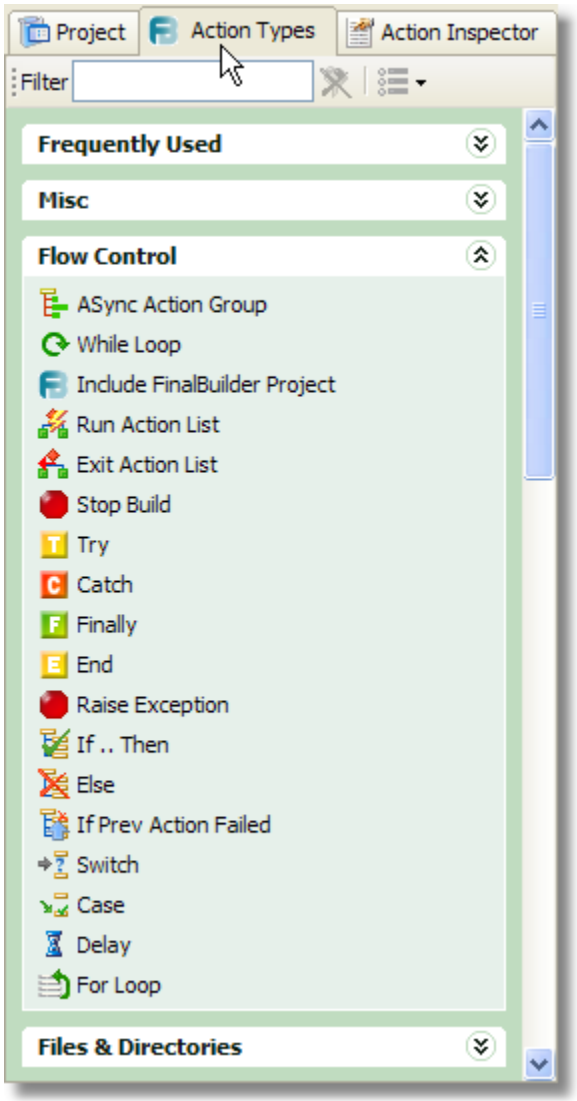
Actions are the cornerstone of FinalBuilder. FinalBuilder chains together Actions into Action Lists to create a repeatable build process. FinalBuilder has built in actions to do things such as Copy Files, Move files, run programs etc. In addition, FinalBuilder has actions to interface with a number of third party products such as Borland Delphi, Microsoft Visual Basic, Microsoft Source Safe, Wise Installer 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 process, you will start off by adding actions to the Main Action list. See the Action Lists topic for more information.

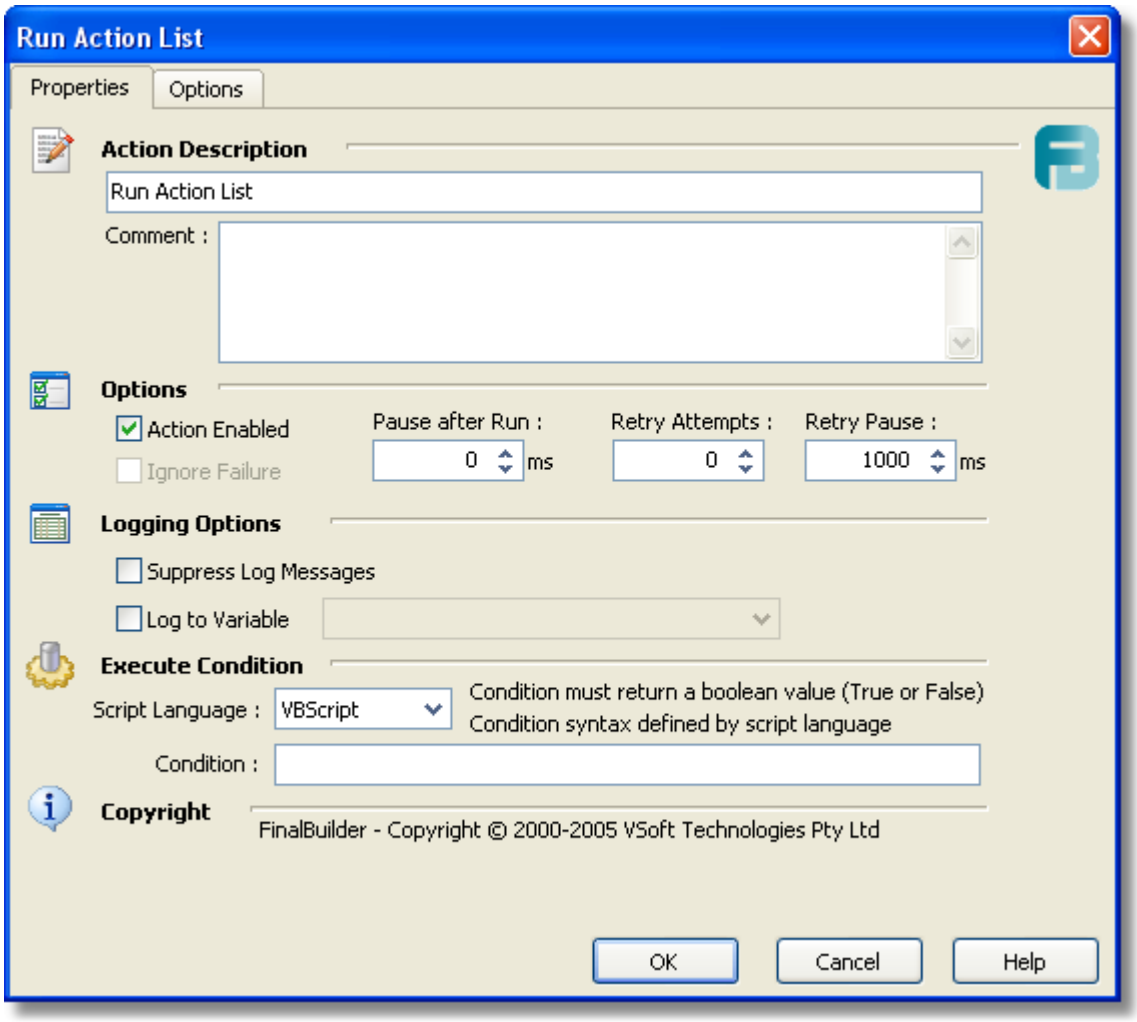
Using Actions

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:

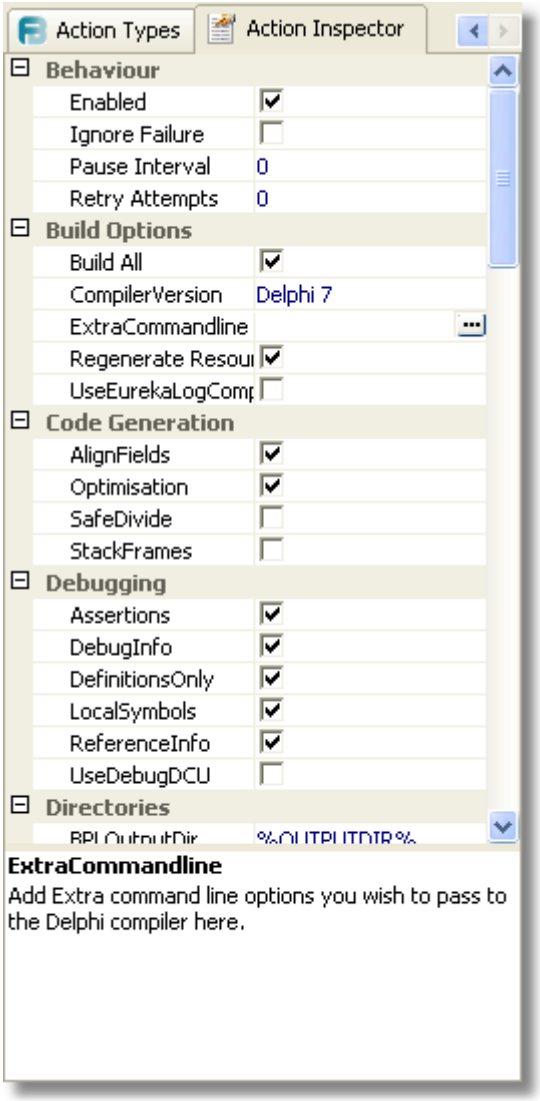
© 2001-2006 VSoft Technologies Pty Ltd



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

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

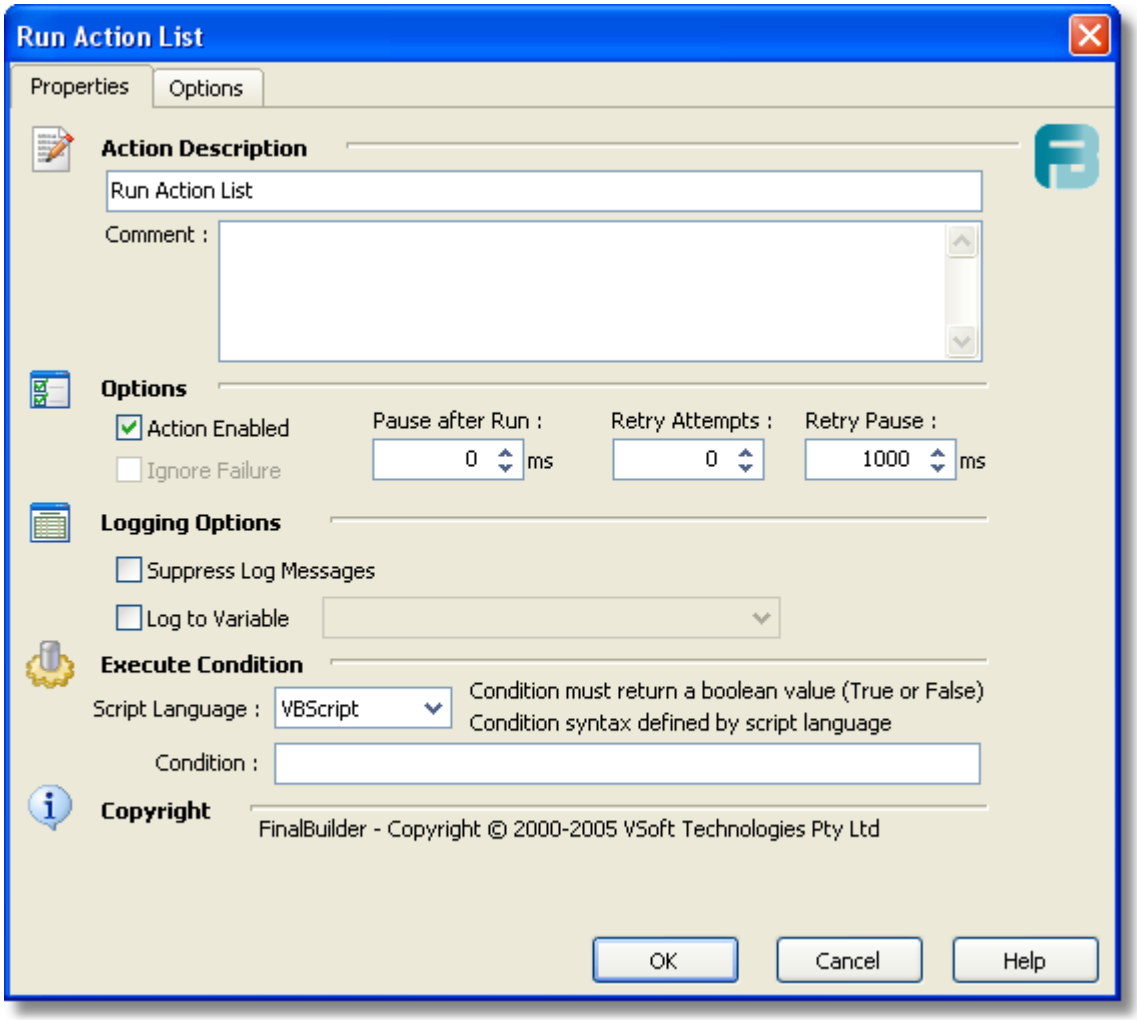
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.1 Common Action Properties

All FinalBuilder actions have the following properties:



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.)

Action enabled

Disable an action to have it ignored during the build process. Disabled actions are shown greyed out in italics.

Ignore failure

Even if this action fails, the build will continue. Some actions (such as Run Action List, shown above) 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.

© 2001-2006 VSoft Technologies Pty Ltd

Pause after Run

After the action completes, the build process 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 the Try action topic for more details.

Logging Options

Check "Suppress log messages" to have none of the output written to the log.

Check "Log to variable" to have the contents of the action output written to a FinalBuilder variable.

Execute Condtion

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

© 2001-2006 VSoft Technologies Pty Ltd

5.2 Flow Control

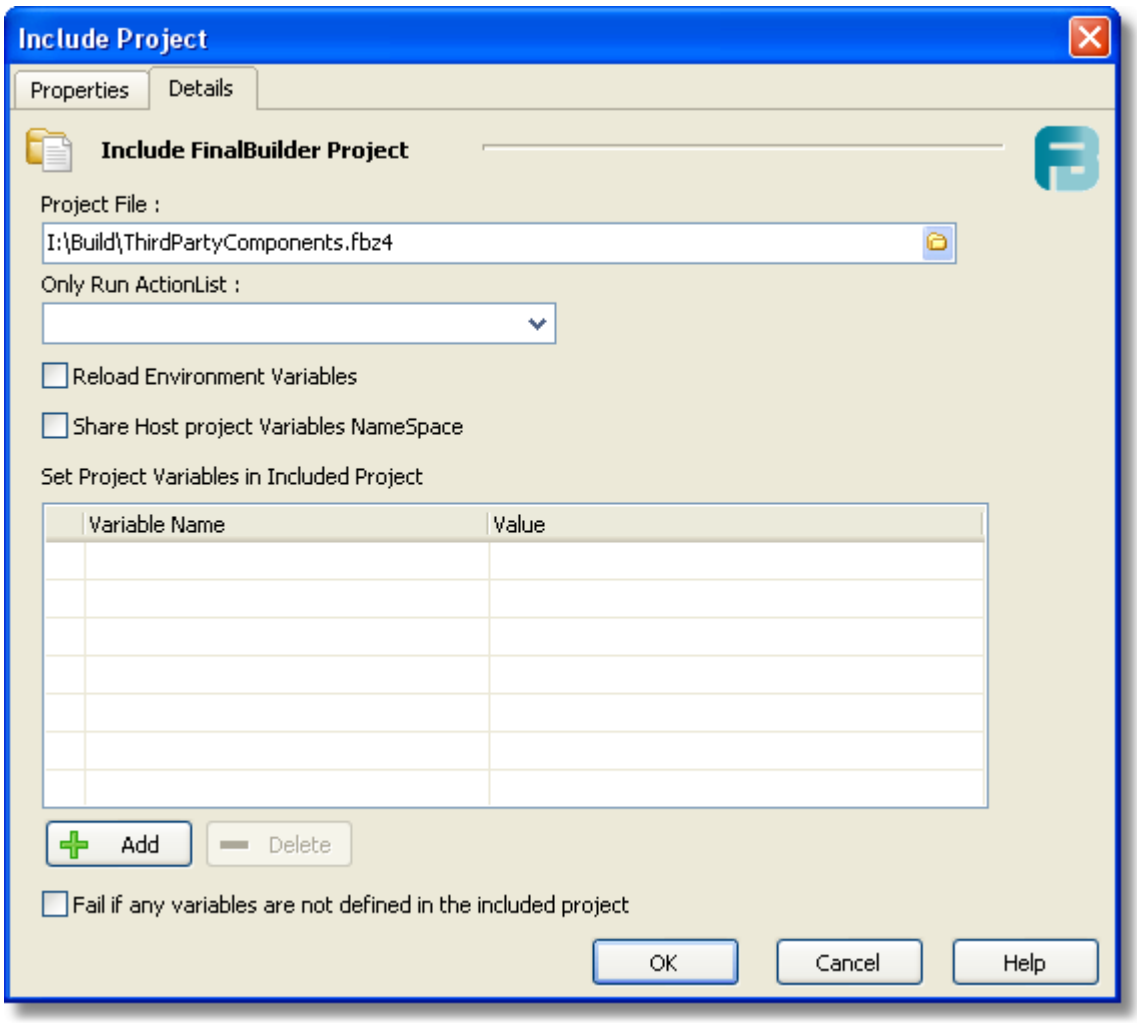
5.2.1 Delay Action

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

© 2001-2006 VSoft Technologies Pty Ltd

5.2.2 Include FinalBuilder Project

This action type allows you to include other FinalBuilder projects in your build process. This allows you to modularise your build process 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 cannnot 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.

© 2001-2006 VSoft Technologies Pty Ltd

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 are loaded at that time. If there are variable name clashes then the existing project variable 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.)

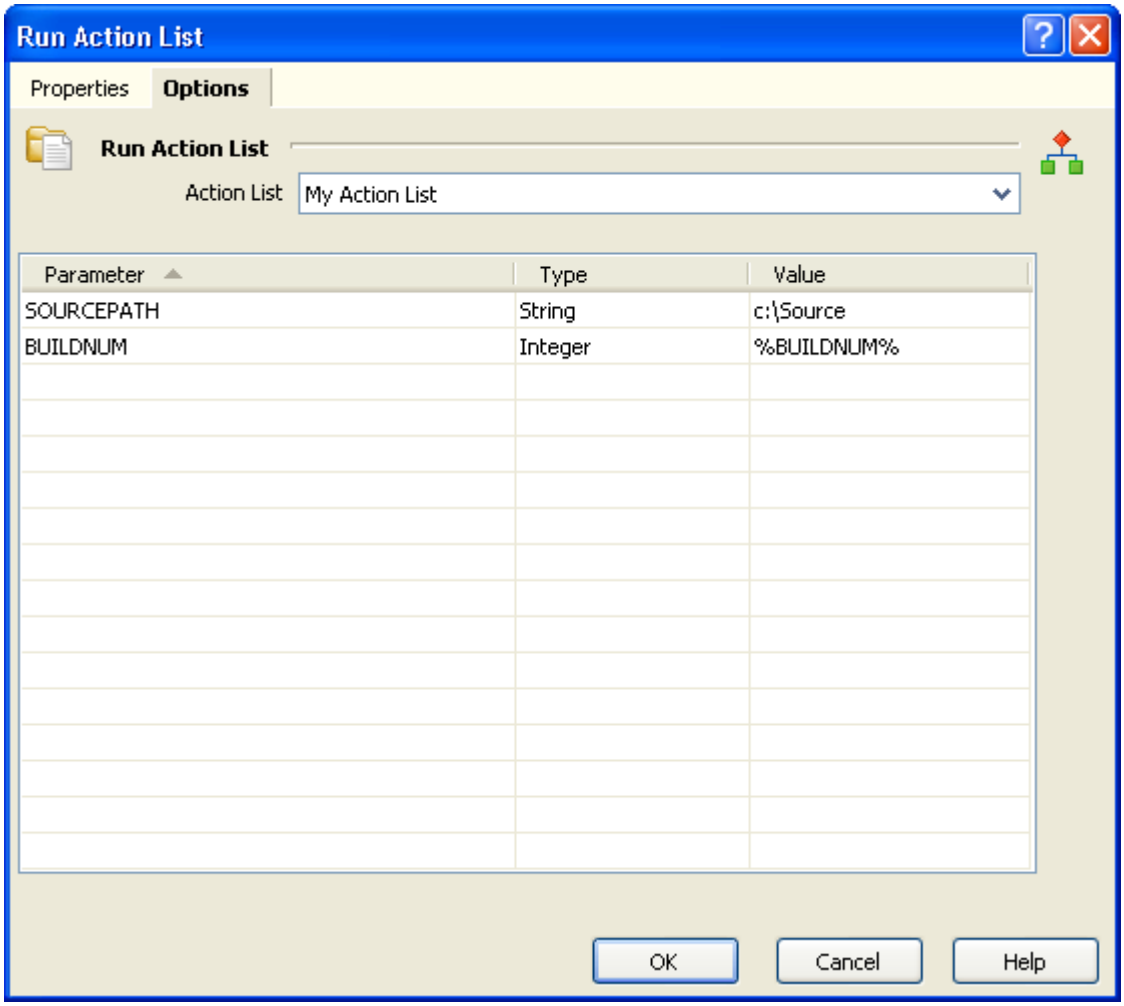
Fail if Any Variables Do Not Exist in the Included Project

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.3 Run Action List Action

This action enables you to run a different Action List, you can add new Actions Lists from the Project Menu. Note you should take care to avoid circular references between action lists - FinalBuilder does not check for recursion, so Action Lists may be used recursively, but doing so may cause your build to infinitely recurse.



See Also

Action Lists
Action List Parameters

© 2001-2006 VSoft Technologies Pty Ltd

5.2.4 Exit Action List Action

The action causes the current action list to terminate and control flow returns to the calling action list.

Note that all Finally sections will run first before the action list is exited.

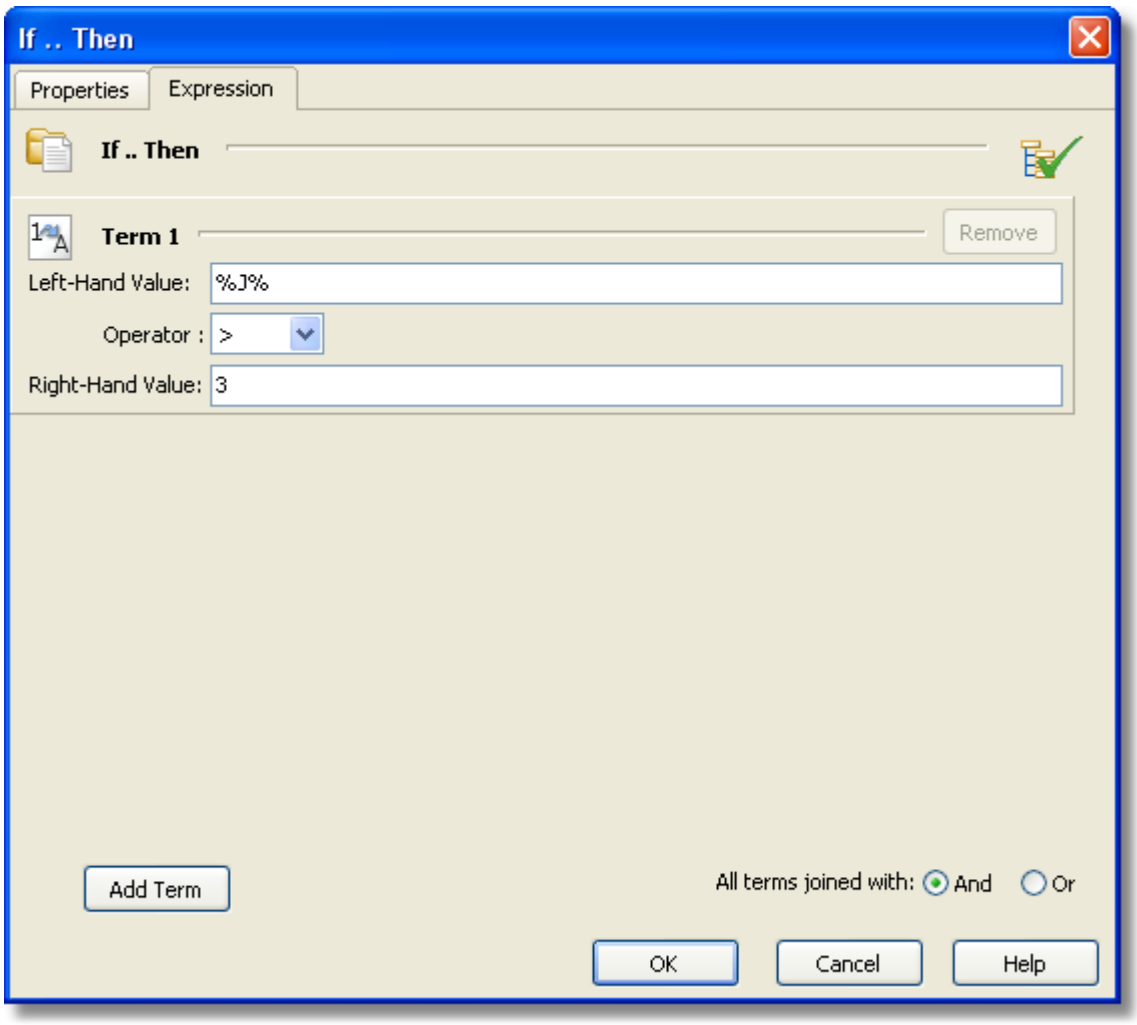
See Also

Stop Build Action

© 2001-2006 VSoft Technologies Pty Ltd

5.2.5 If .. Then Action

This action uses a boolean expression to determine if its child actions should run or not.



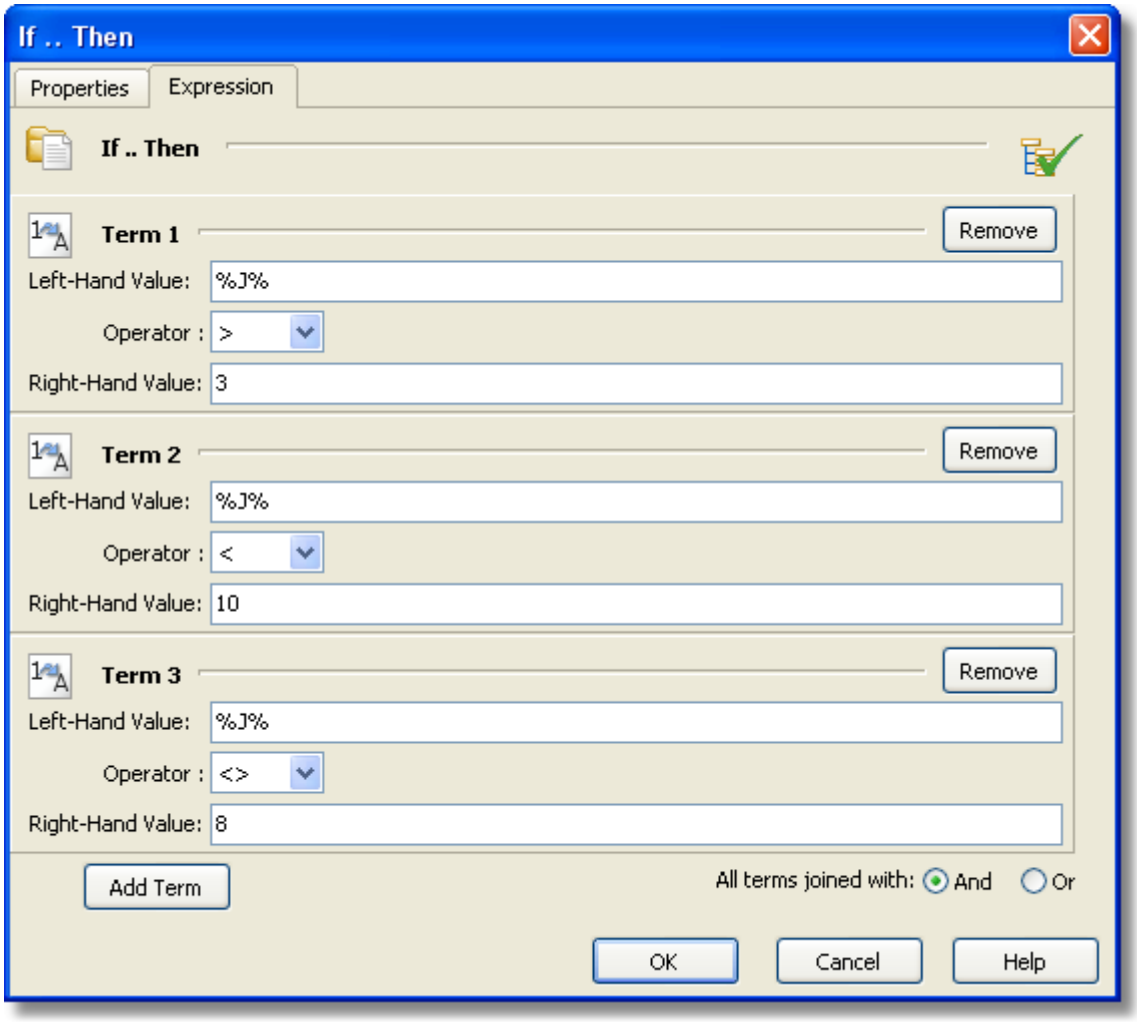
Each term is comprised of a left hand side, a right hand size, and a comparison operator. Use %VariableName% syntax to compare the values of variables. String, integer or floating point values can all be compared.

Each expression can have an unlimited number of terms. Click on the "Add Term" button to add a new term. Terms can be joined together with boolean "And" or "Or" directives.

Example

The following If... Then action will execute its children when variable J is equal to 4, 5, 6, 7 or 9:

© 2001-2006 VSoft Technologies Pty Ltd

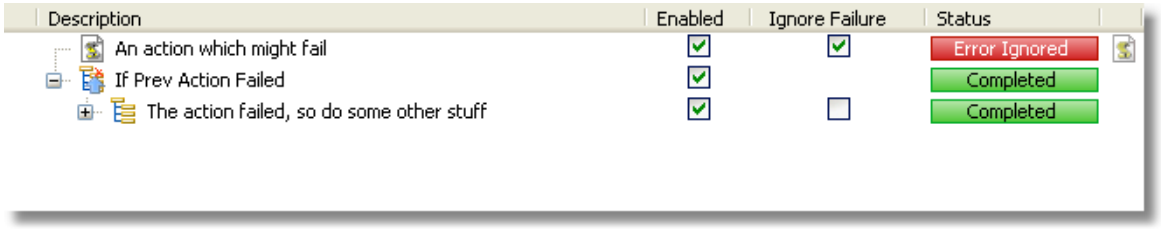


You can also use If..Then to execute a set of actions if the expression evaluates to false. See the Else Action for details.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.6 If Prev Action Failed Action

This action will execute it's child actions only if the previous sibling action failed. For this to happen the previous sibling action needs Ignore Failure enabled:



You can also use execute a set of actions if the previous action succeeds (see the Else Action.)

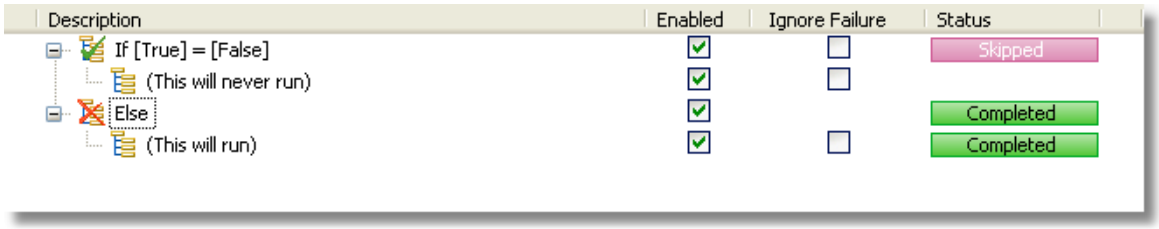
For more sophisticated error handling and control, you may wish to look at the Try and Catch actions.

© 2001-2006 VSoft Technologies Pty Ltd

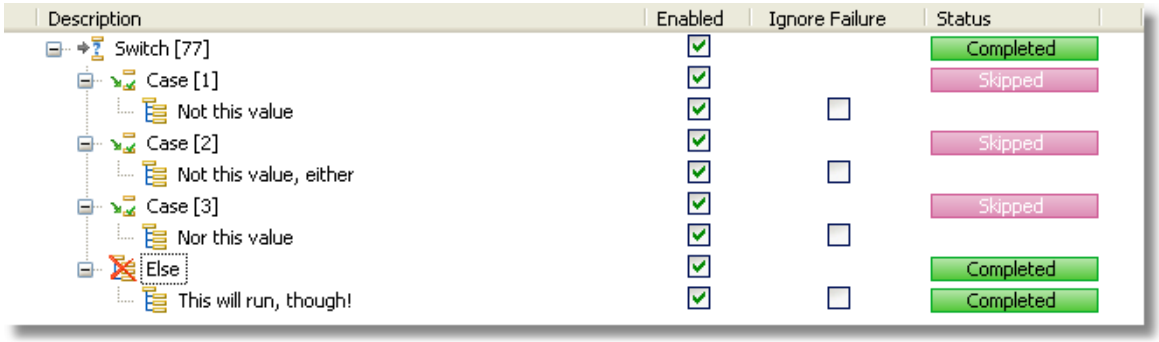
5.2.7 Else Action

The else action can be used with the "If..Then" action, the "If Prev Action Failed" action or the Switch/Case Action:

For if actions:



For switch actions:



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).

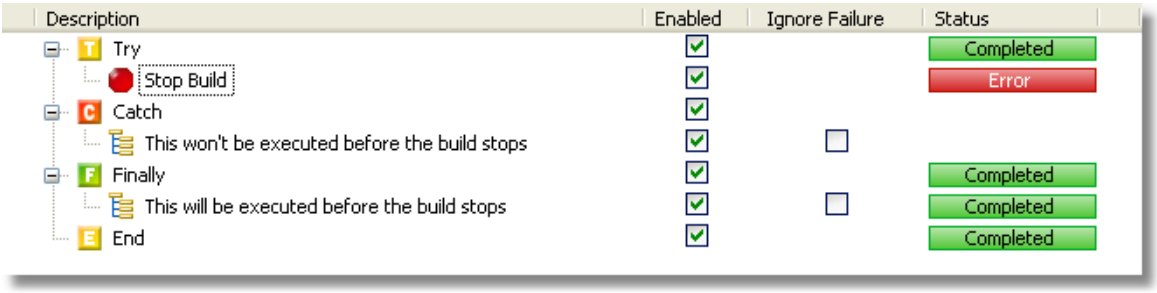
See Also

If .. Then Action | If Prev Action Failed Action | Switch Action | Case Action

© 2001-2006 VSoft Technologies Pty Ltd

5.2.8 Stop Build Action

This action will cause the build to stop.



The BuildResult property determines if the build stops with success or failure. If BuildResult is False the OnFailure List will execute.

Note that if the Stop Action is inside a Try/Finally block the Finally block will still execute.

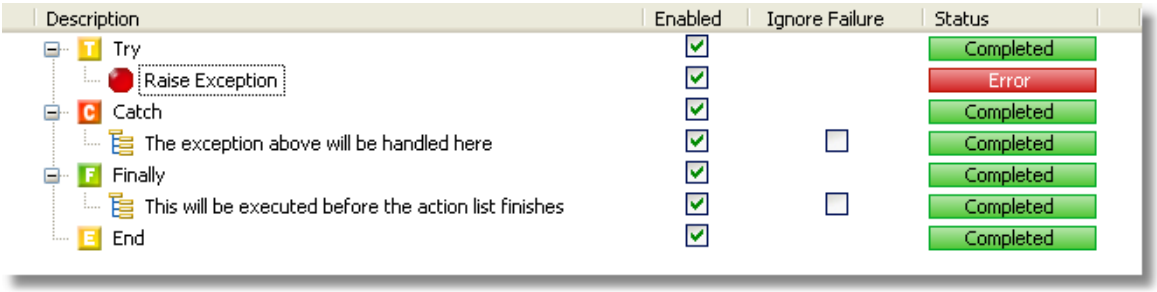
See Also

Exit Action List Action

© 2001-2006 VSoft Technologies Pty Ltd

5.2.9 Raise Exception

The raise exception action will cause an error condition to be set when it is executed. The flow of the build will then change as if an error had occurred.



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.

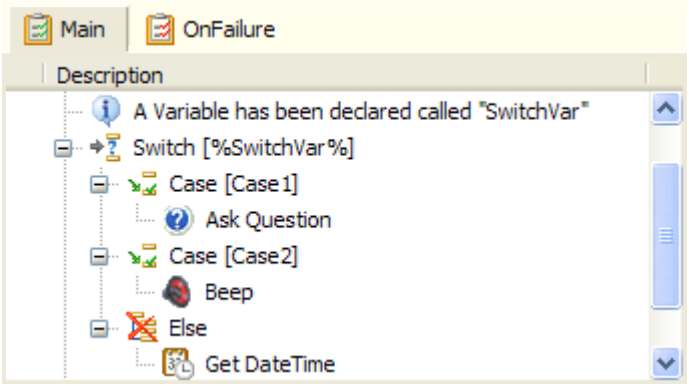
In the example above, the build stop with status "Succeeded" because the exception was handled in the Try/Catch block.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.10 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 should 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.



See Also

Else Action

© 2001-2006 VSoft Technologies Pty Ltd

5.2.11 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

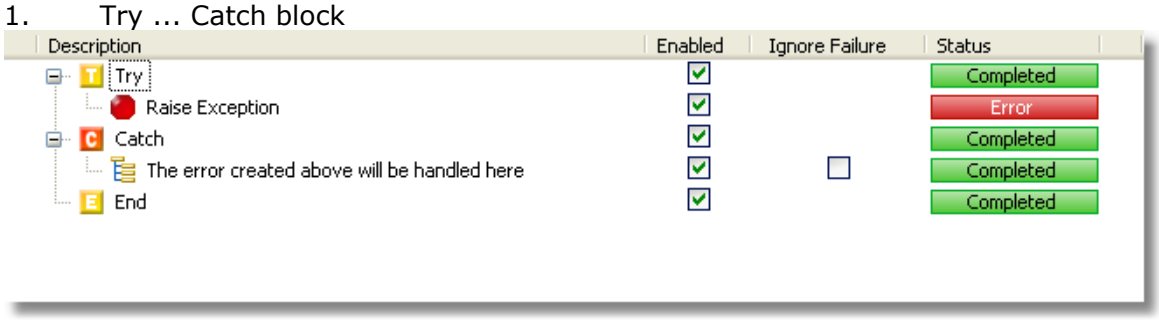
© 2001-2006 VSoft Technologies Pty Ltd

5.2.12 Try/Catch/Finally/End Actions

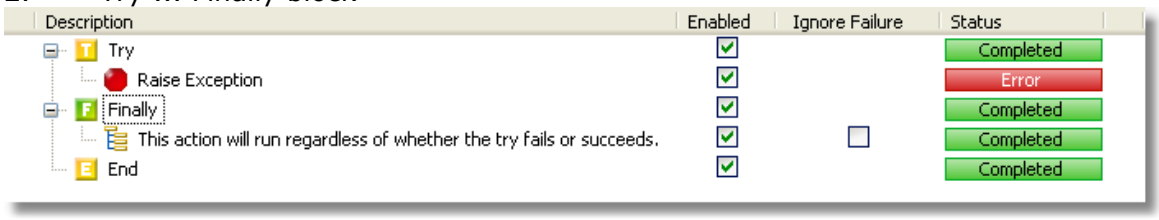
The Try action, along with the Catch, Finally and End actions provide structured exception handling. They allow you to create localised error handling and resource protection, just as you do in programming languages such as C++, C#, Delphi etc.

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

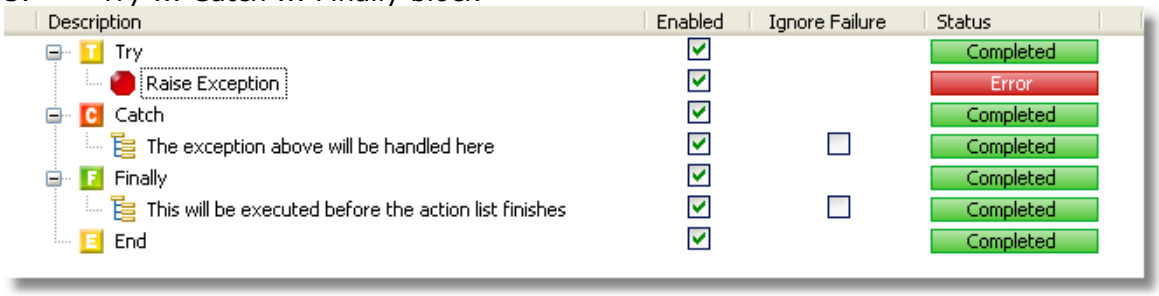


2. Try ... Finally block



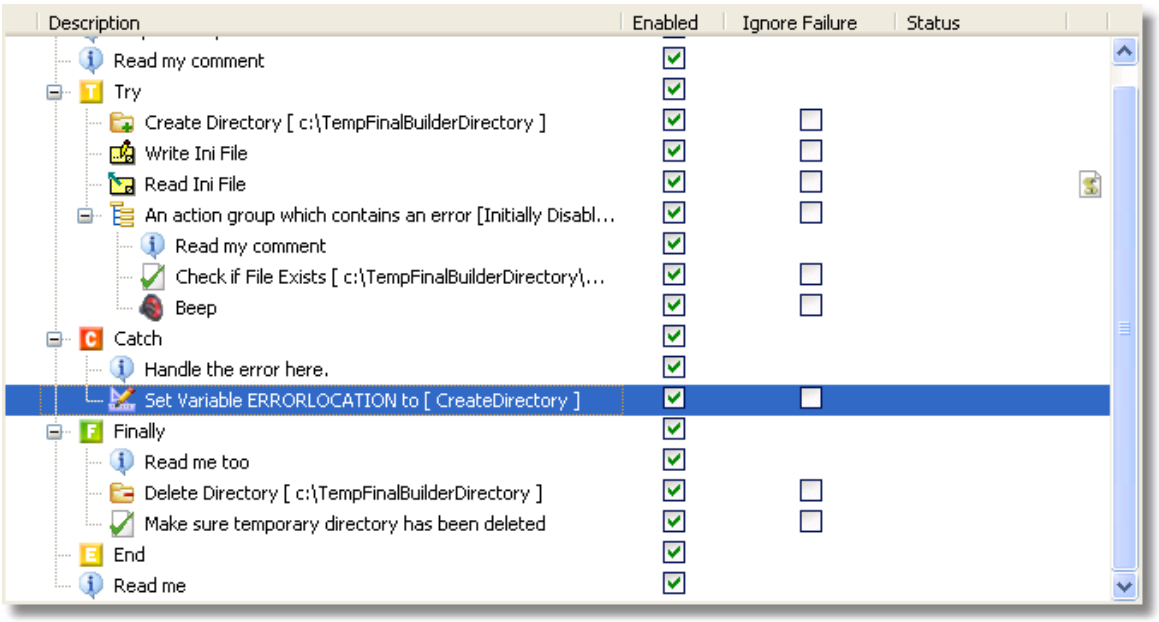
A Finally Action will almost always execute it's child actions, the exception being if there is a structural error with the try or catch (ie. a missing end, or finally before catch etc).

3. Try ... Catch ... Finally block



A more complete example:

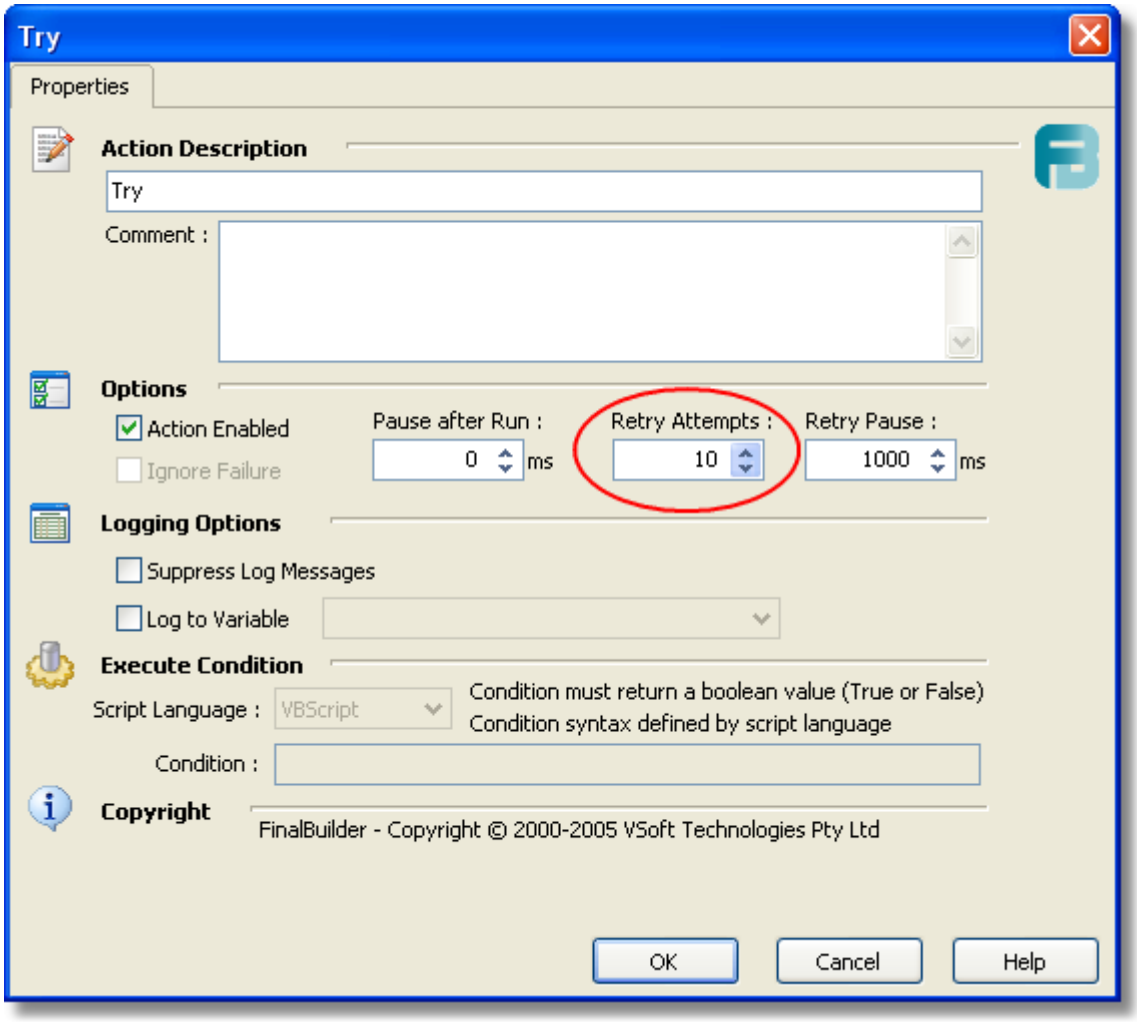
© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

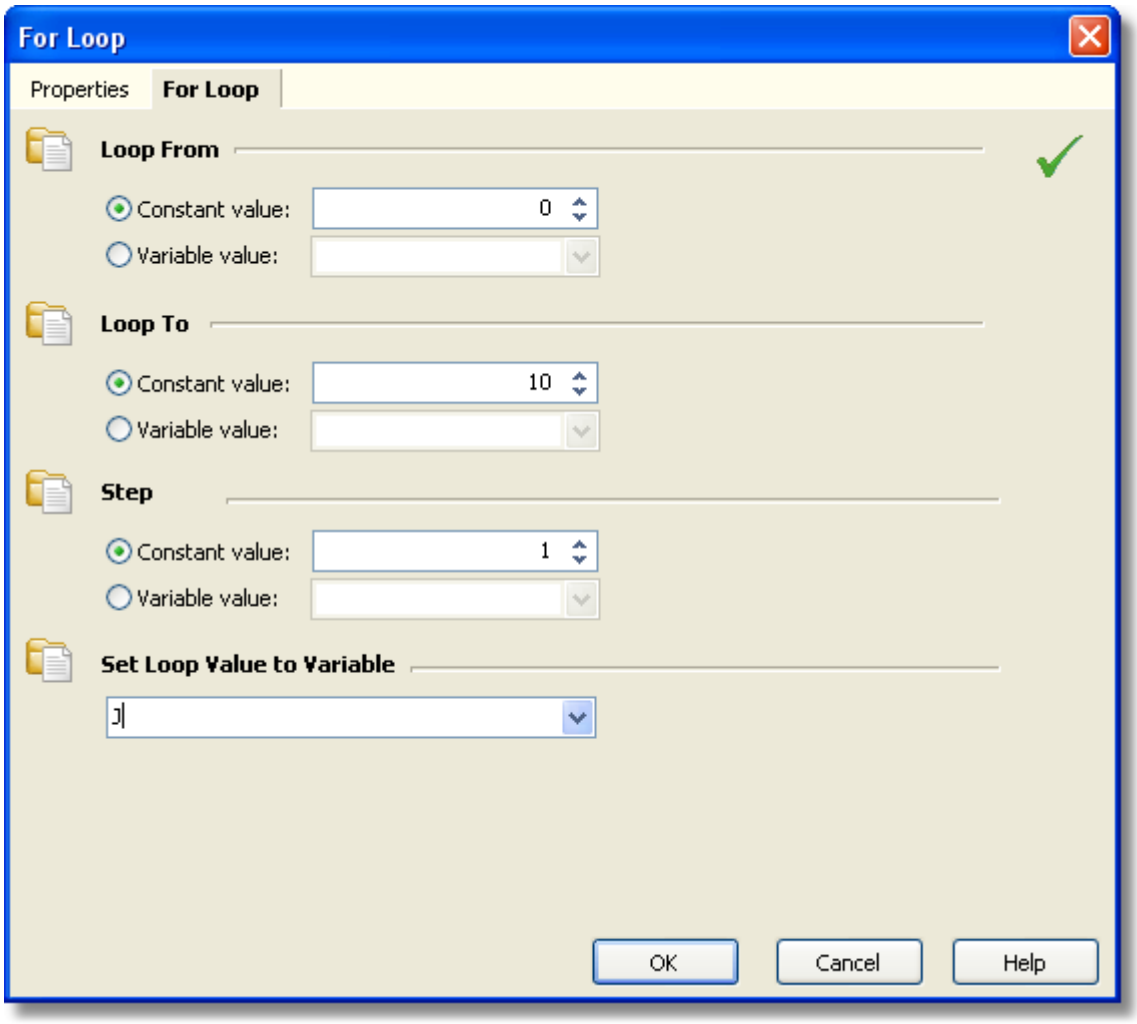
© 2001-2006 VSoft Technologies Pty Ltd

5.2.13 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 J = 0 To 10 Step 1 (BASIC)
or
for(j = 0; j <= 10; j++) (C/Java)



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.)

© 2001-2006 VSoft Technologies Pty Ltd

The 'to' value can be either a constant or a FinalBuilder variable.

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

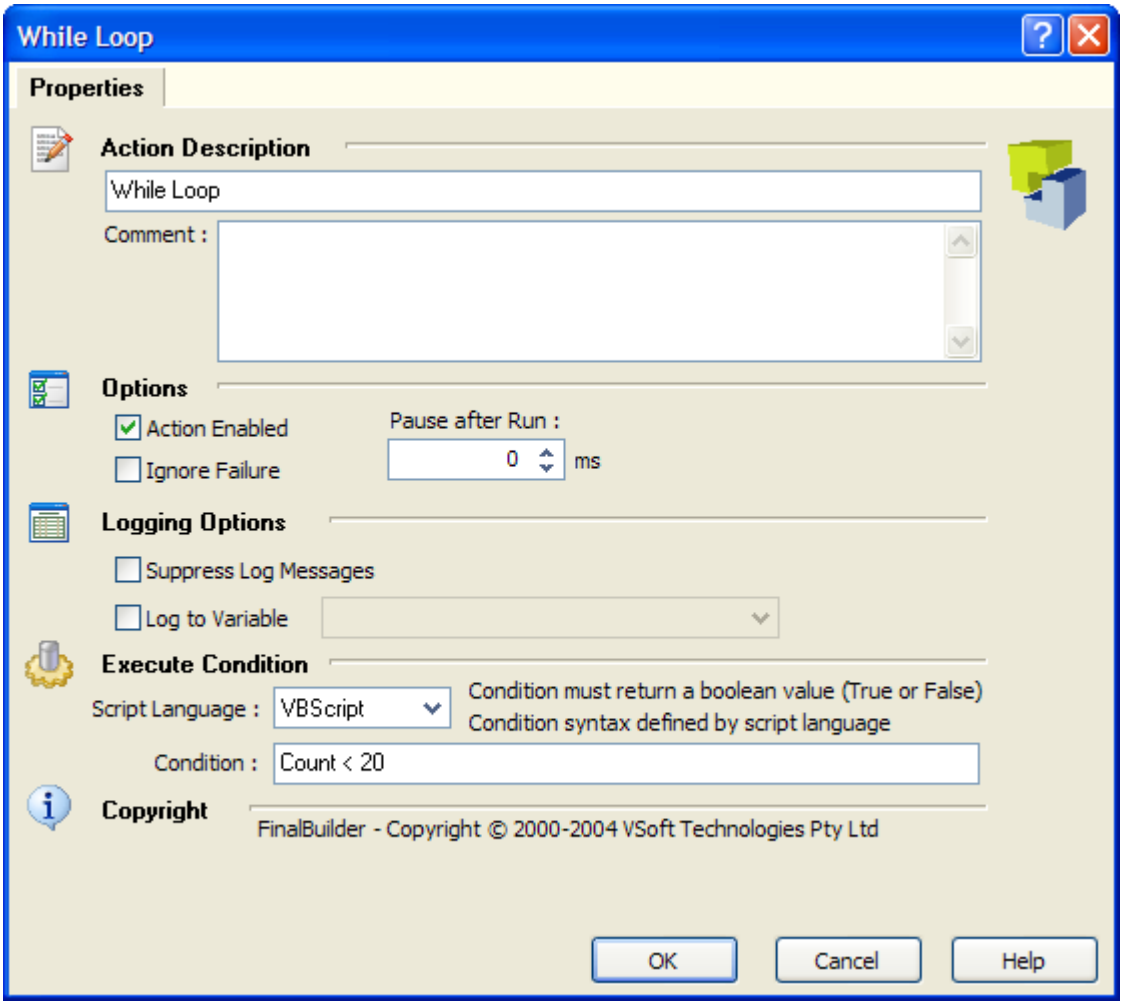
Specify a FinalBuilder variable to use for the loop value. If no variable is specified, the loop value will be stored internally.

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.14 While Loop Action

The While Loop will execute while it's Condition evaluates to True. The While action has no specific functionality other it's looping capability (specifically it tells the stepping engine which action to run next, overriding the default behaviour).



NOTE: Note that it will execute its script events (like any other action) first before executing it's child actions. What this means is that the While Loop Action will execute it's AfterAction Event Before any of the Child Actions are executed. The reason for this is that the stepping engine does not have any knowledge of which type of action it is executing and does not treat the while loop action any differently.

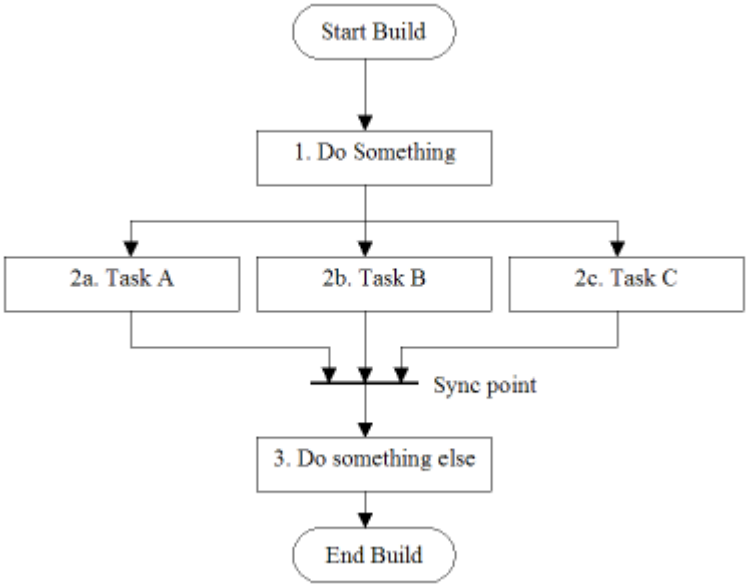
While a While Loop is an iterator, it does not share any of the common iterator script events and properties.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.15 Running actions in Parallel

Motivation

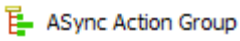
FinalBuilder allows you to structure your build process so that multiple actions can be run simultaneously. A graphical representation of multiple actions running in parallel is represented by this flow diagram:



Task 1 needs to finish before actions 2a, 2b and 2c can run. Actions 2a, b and c all need to finish before action 3 can run.

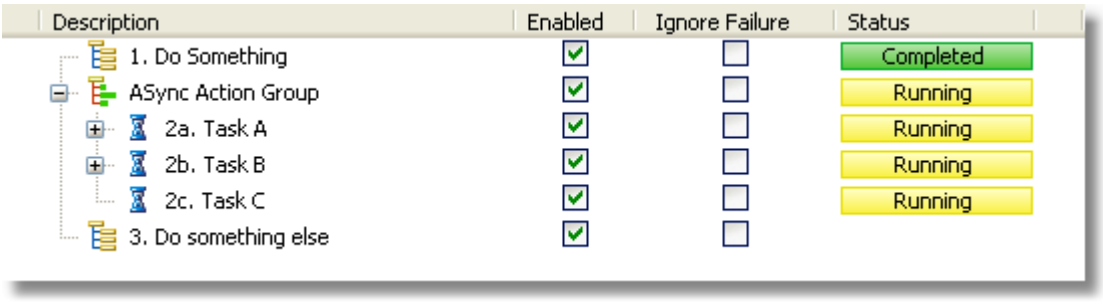
Async Action Groups

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:

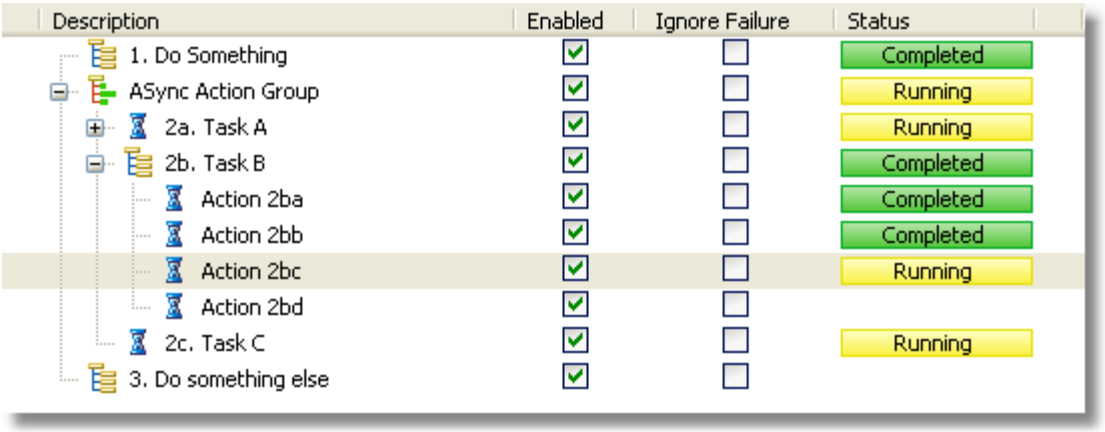


© 2001-2006 VSoft Technologies Pty Ltd

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

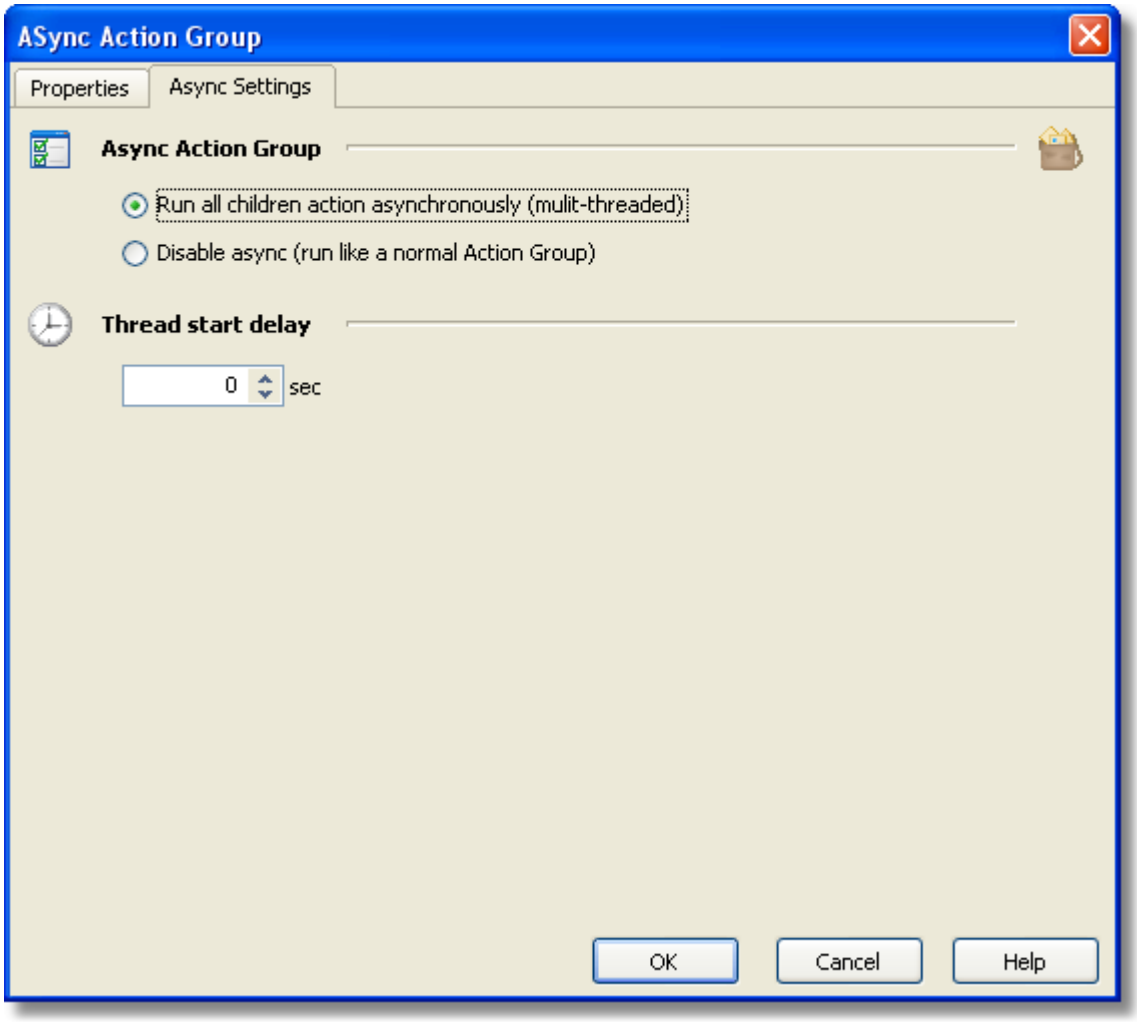
If actions under an ASync group have child actions, then these actions will run sequentially in the same thread as the parent (ie not asynchronously) and follow the normal processing rules:



(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

© 2001-2006 VSoft Technologies Pty Ltd



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.

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.

Getting the Most from Async Action Groups

The purpose of ASync Action Group is to allow the build process to run multiple actions simultaneously to speed up the build 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 build process 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.16 Waiting For Events (Wait For actions)

The "Wait For..." actions allow you to pause your build process 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

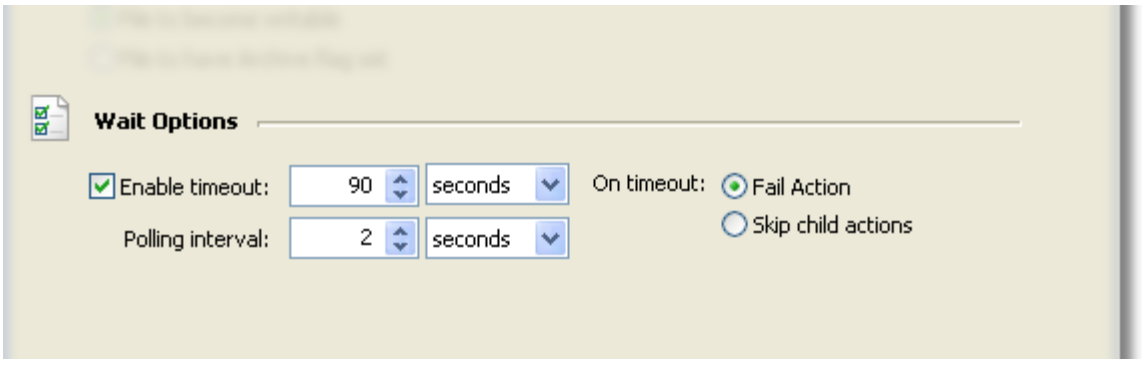
Note that the "Wait For..." action is a "busy-waiting" or "polling" wait - a particular wait condition is checked every so many seconds to see if it has changed.

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 build/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:



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.

© 2001-2006 VSoft Technologies Pty Ltd

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 build.)

Timing Limitations

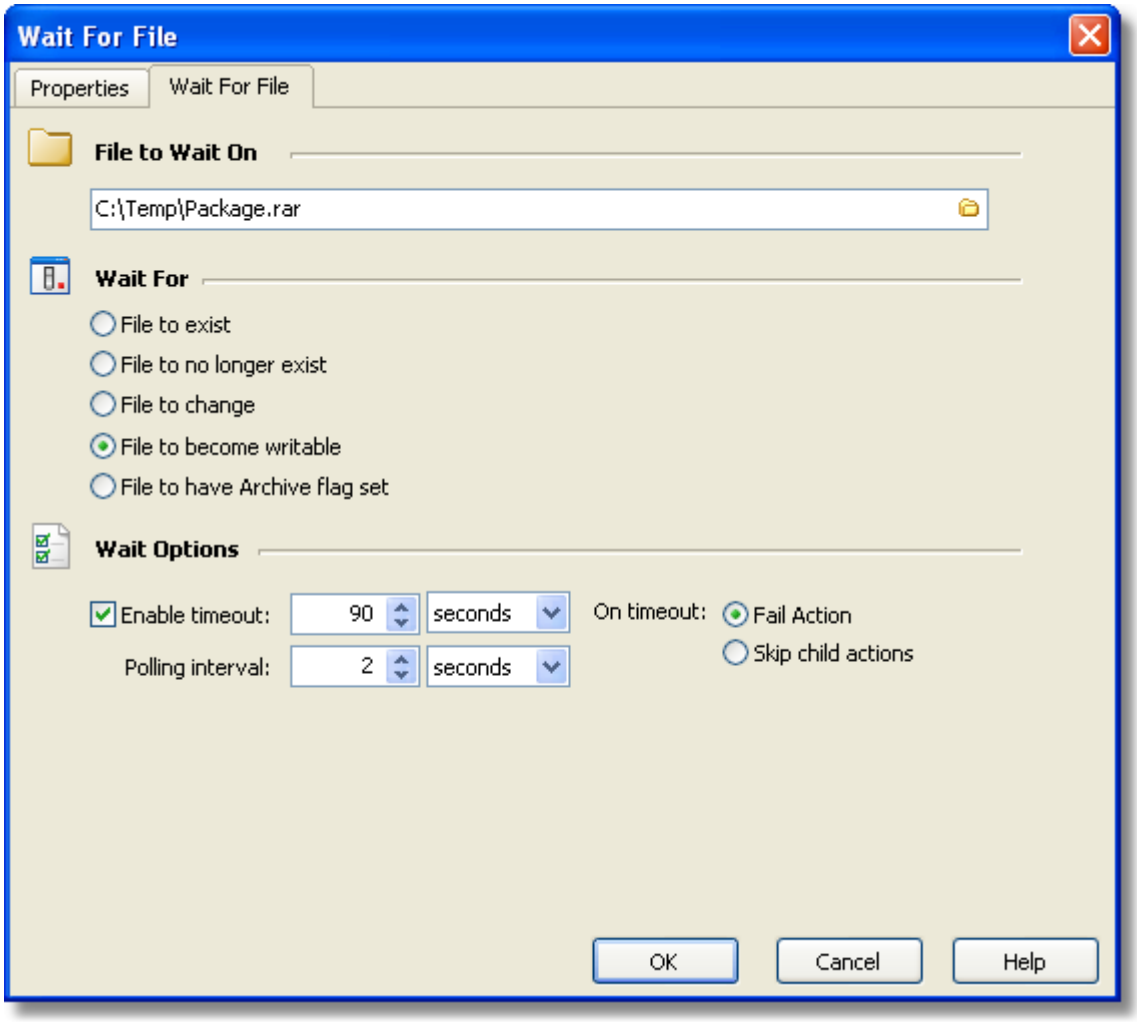
Note that the Polling Interval is not a polling frequency - it does not guarantee that the action will poll every X seconds, it only guarantees that the action will pause for X seconds in between each poll. In reality, the action will poll every X+(Length of last polling event) seconds, and in some actions (ie Wait For Remote Computer) the polling event itself may take several seconds.

5.2.16.1 Wait For File

The Wait For File event pauses the build process until something happens to a file.

For an overview of "Wait For..." events, including specifics of the Wait Options, see this topic.

© 2001-2006 VSoft Technologies Pty Ltd



You can choose to wait for one of the following conditions:

File to exist

The build will continue when the given file has been created.

File to no longer exist

The build will continue when the given file has been deleted or moved.

File to change

The build 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 build will continue when the file has its Read Only flag cleared, and is not locked for write access.

© 2001-2006 VSoft Technologies Pty Ltd

In the example screenshot above, if Package.rar is being created when the action runs then the build 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 build will continue when the file has its Archive (A) bit set.

Wait Options
See the Waiting For Events topic.

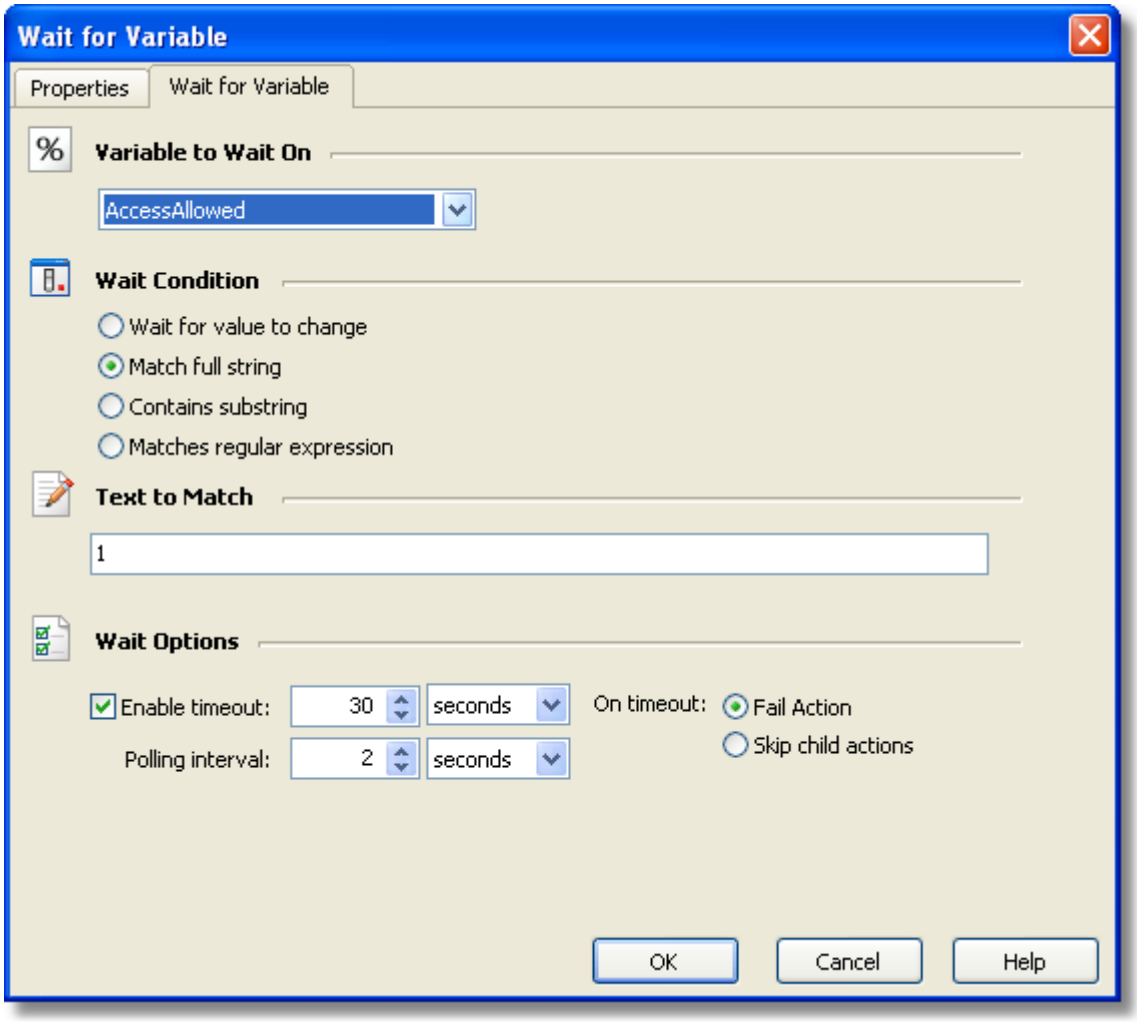
5.2.16.2 Wait For Variable

Wait for Variable pauses the build process 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.)

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

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."

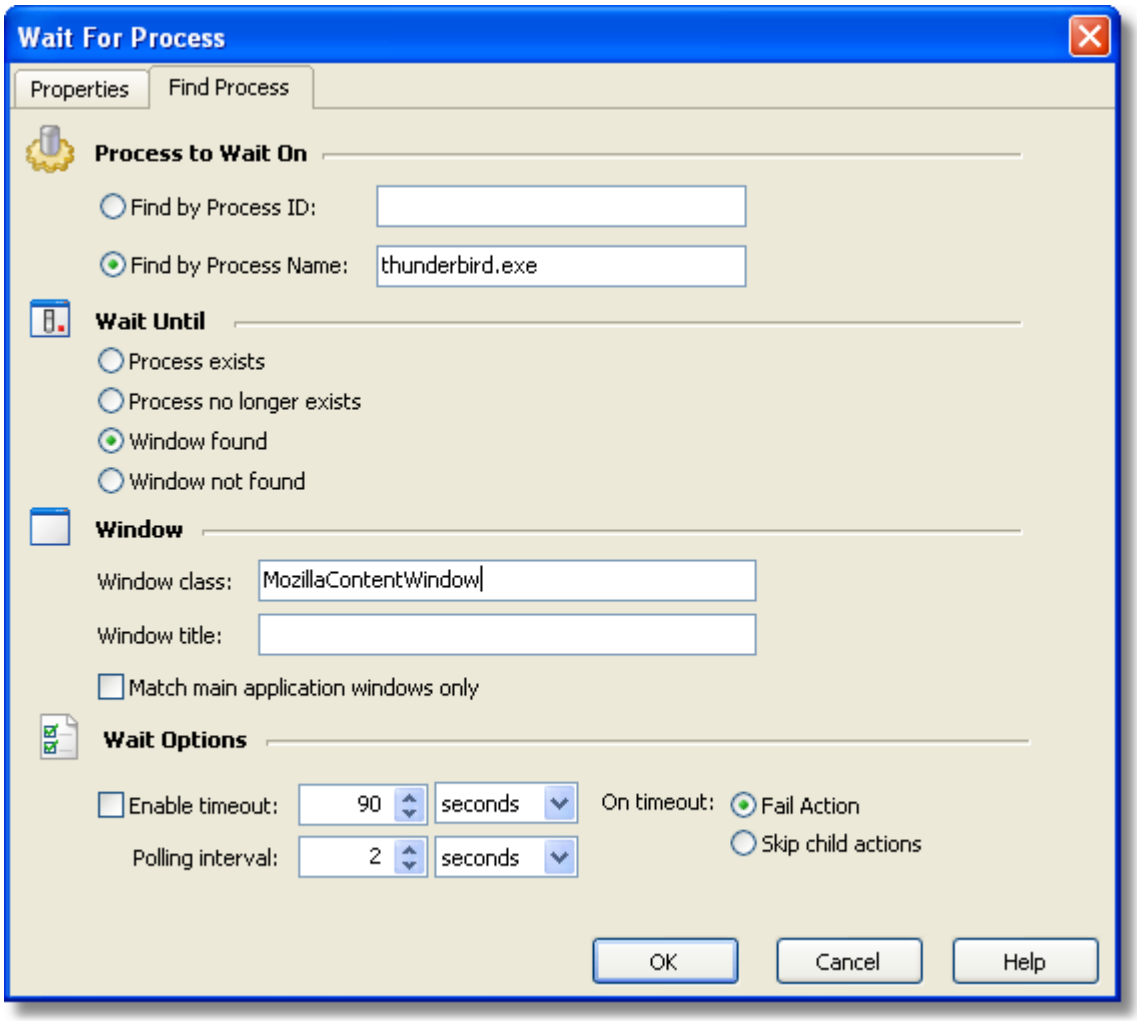
Wait Options

See the Waiting For Events topic.

5.2.16.3 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

© 2001-2006 VSoft Technologies Pty Ltd

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 Borland'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.

5.2.16.4 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

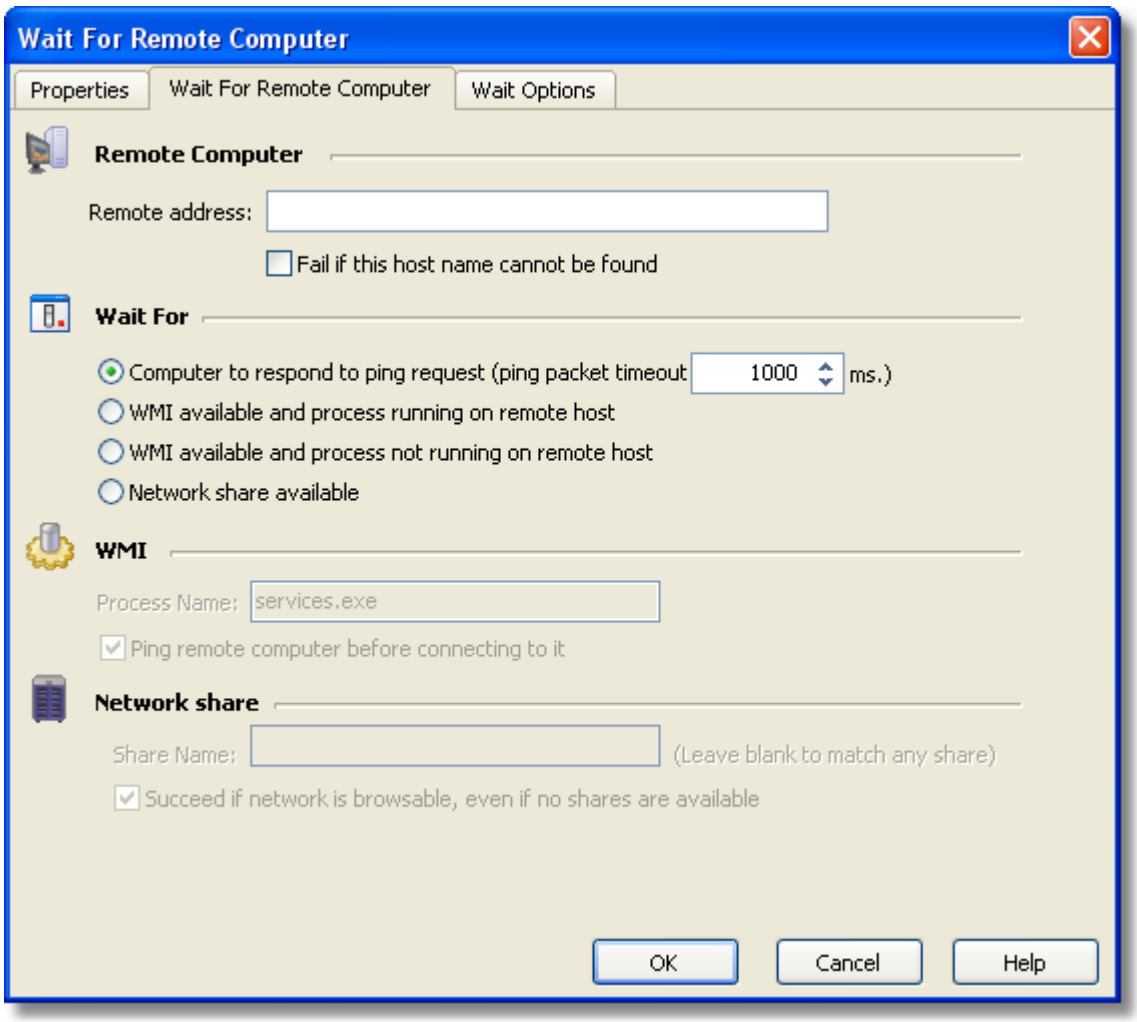
See the Waiting For Events topic for more information on Wait For actions.

© 2001-2006 VSoft Technologies Pty Ltd

5.2.16.5 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.



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,

© 2001-2006 VSoft Technologies Pty Ltd

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.

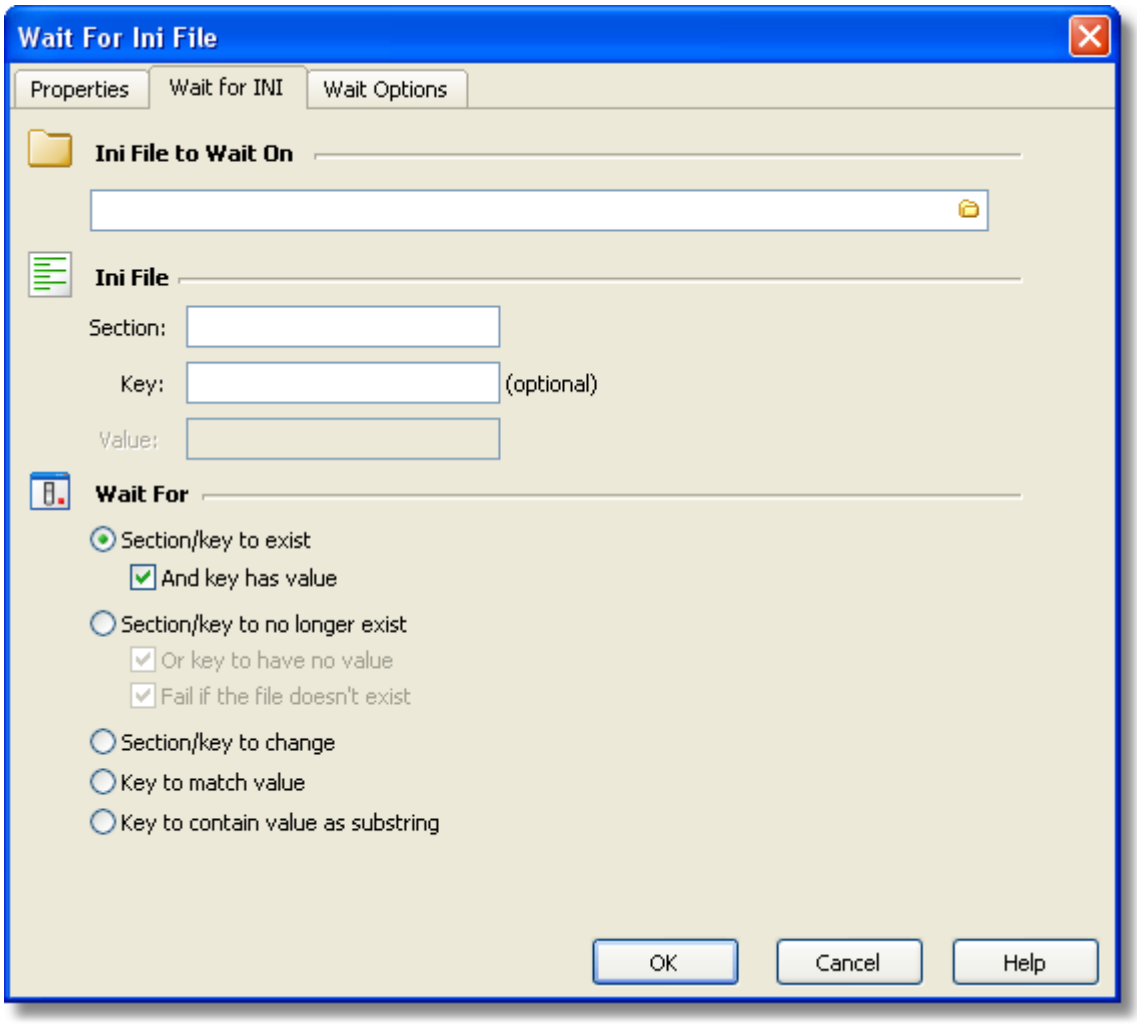
Wait Options

See this topic.

5.2.16.6 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.

© 2001-2006 VSoft Technologies Pty Ltd



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), specifyi t 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**"

© 2001-2006 VSoft Technologies Pty Ltd

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**"

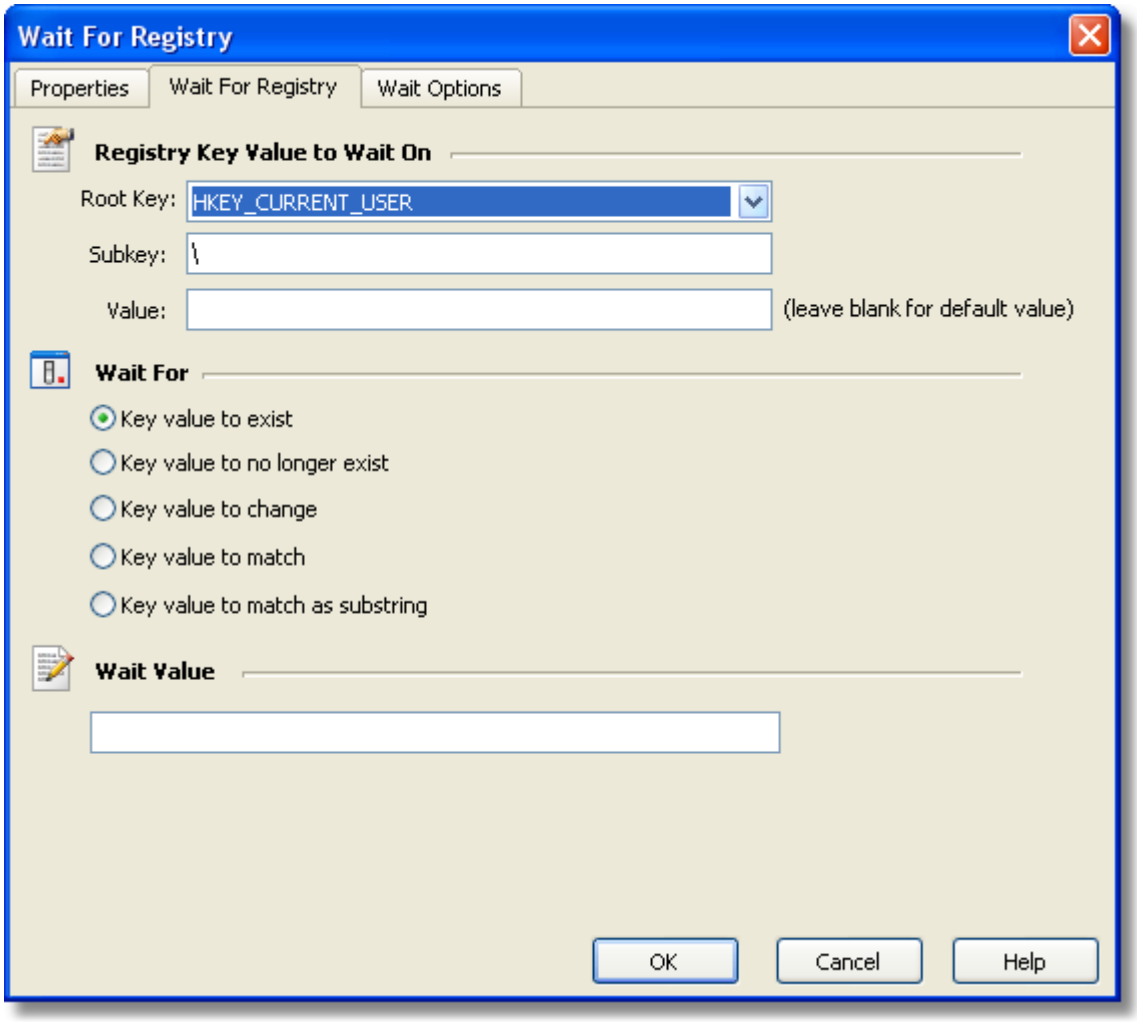
You need to specify a key name and a value to wait for.

The action will continue when the value for the key contains the substring you specify in the "Value" field.

5.2.16.7 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.

© 2001-2006 VSoft Technologies Pty Ltd



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**"

© 2001-2006 VSoft Technologies Pty Ltd

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.

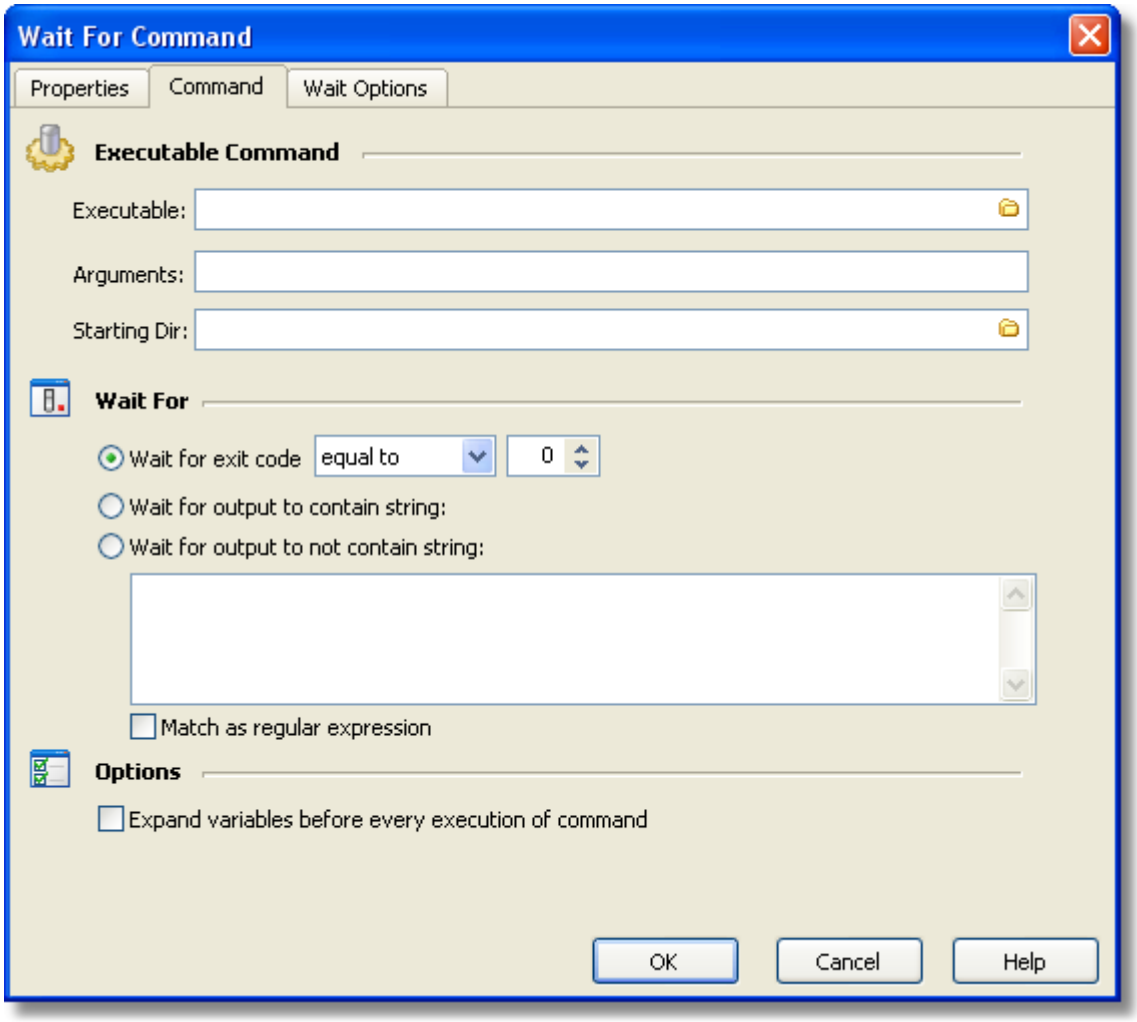
String values are treated literally. DWORD values are treated as their integer values. Binary data blocks are extracted as byte strings. No other data types are supported.

5.2.16.8 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.)

© 2001-2006 VSoft Technologies Pty Ltd



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:**"

© 2001-2006 VSoft Technologies Pty Ltd

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 build process.

(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.)

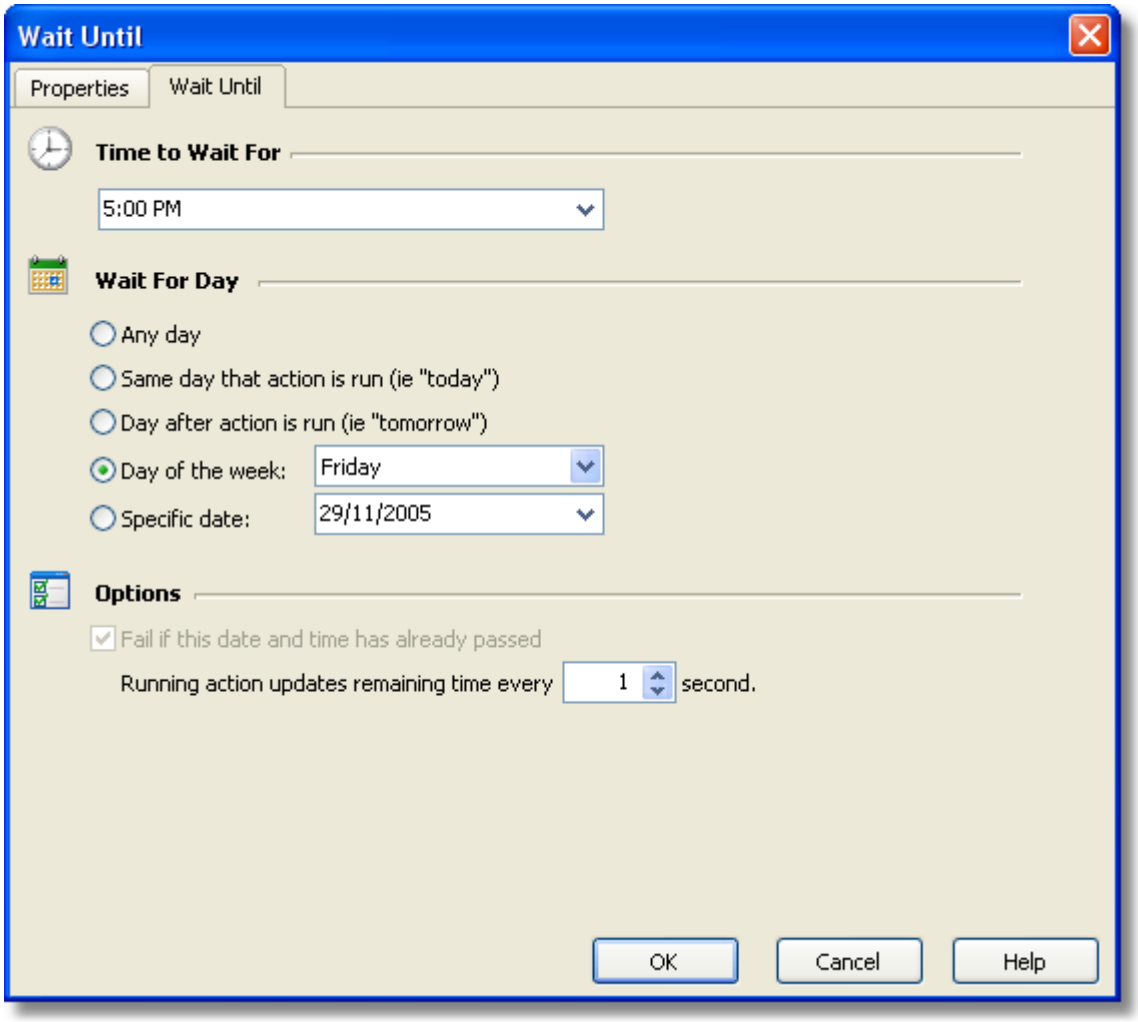
5.2.16.9 Wait Until

The Wait Until action allows you wait for a specific amount of time before continuing your build process.

(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.

© 2001-2006 VSoft Technologies Pty Ltd



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 occurence 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

© 2001-2006 VSoft Technologies Pty Ltd

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 Build Summary 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.

© 2001-2006 VSoft Technologies Pty Ltd

5.3 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 process. 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 successfull. 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.1 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

5.3.1.1 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)

© 2001-2006 VSoft Technologies Pty Ltd

5.3.1.2 .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)

5.3.1.3 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)

© 2001-2006 VSoft Technologies Pty Ltd

- LegalTradeMarks (string)
- ProductName (string)

© 2001-2006 VSoft Technologies Pty Ltd

5.3.2 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>
```

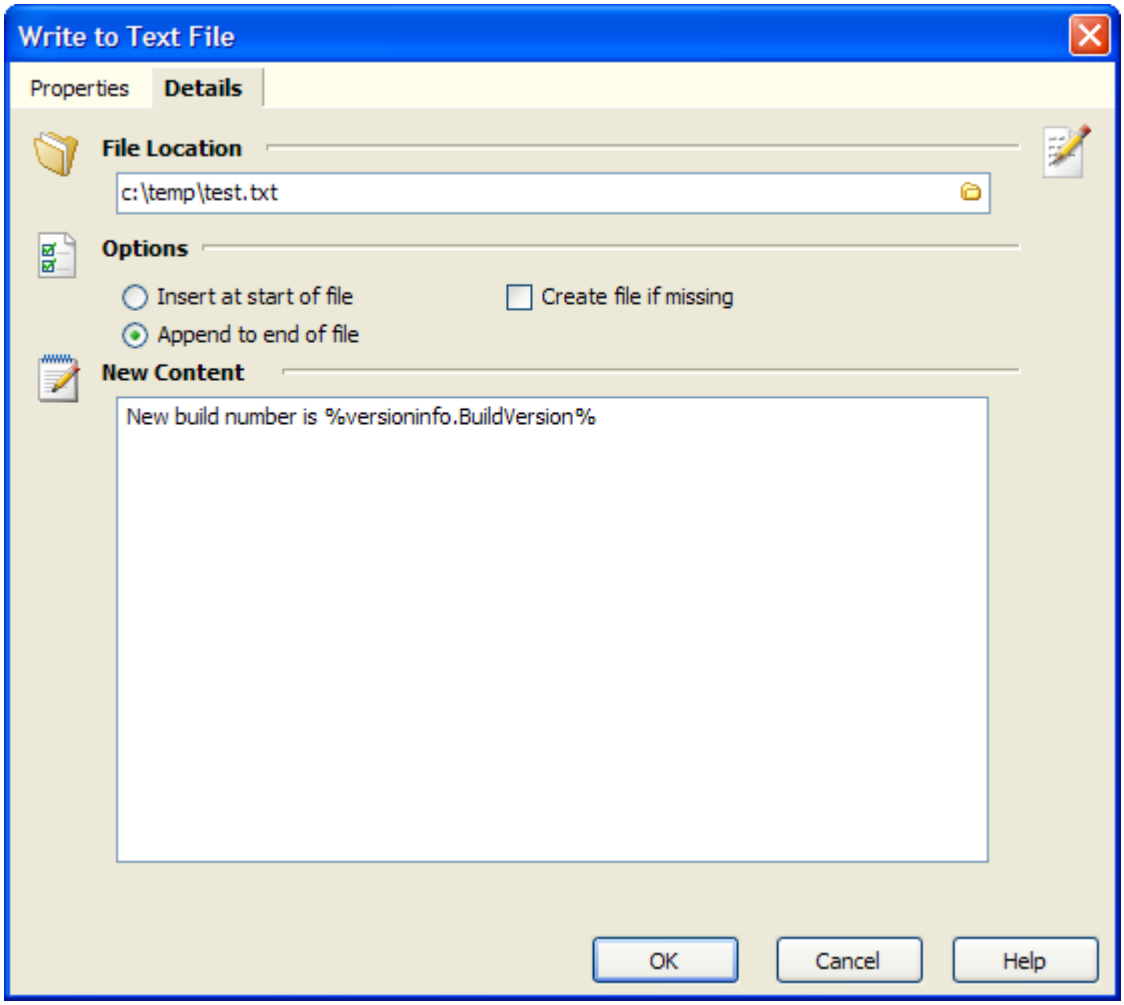
eg. (VBScript)

```
msgbox MyVersionNumbers.MajorVersion
```

Accessing PropertySet values in Action fields

The syntax to access a PropertySet value in an action field is:
%<PropertySetName>.<PropertySetValue>%

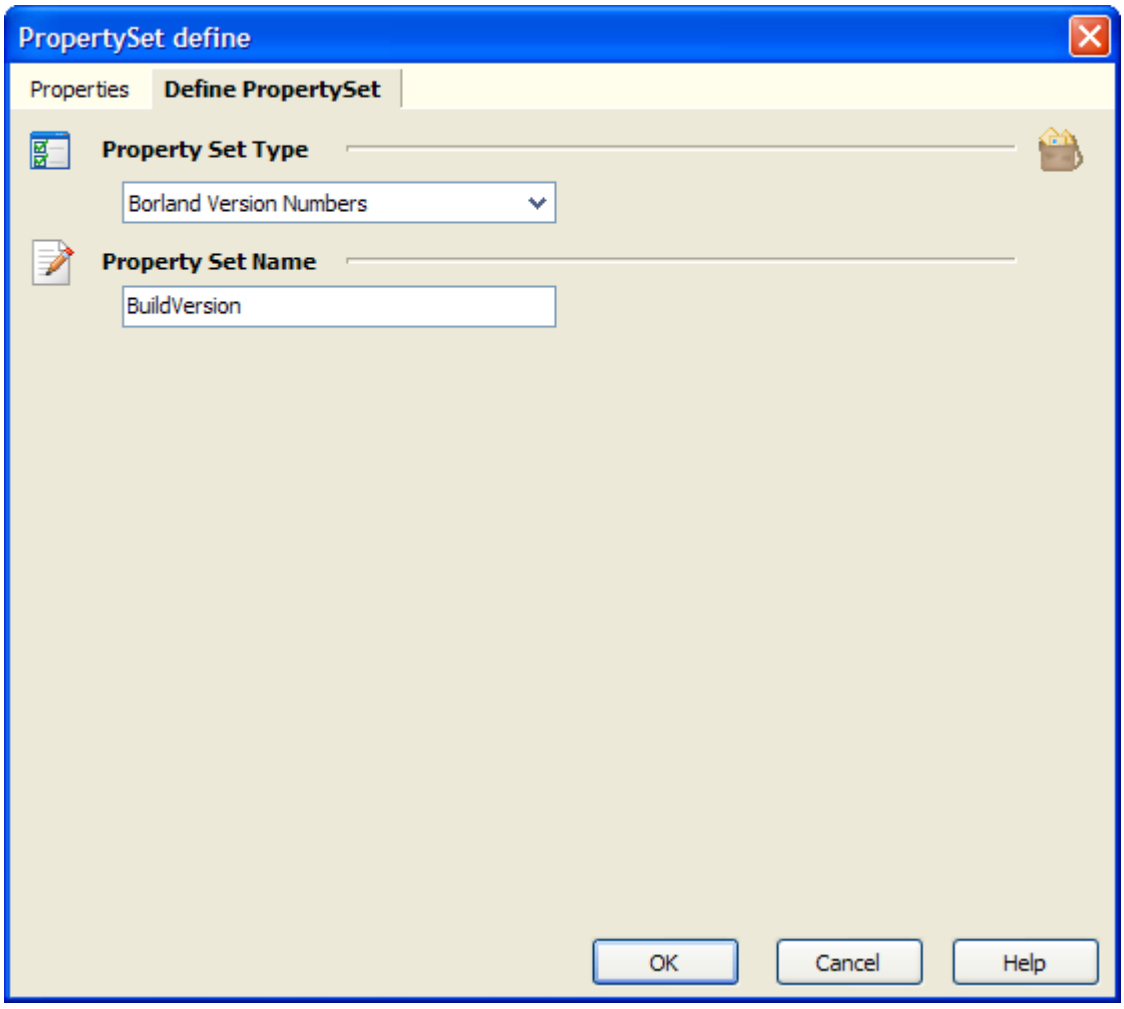
The example below shows how to access the BuildVersion property in an action:



© 2001-2006 VSoft Technologies Pty Ltd

5.3.3 PropertySet define

Use the PropertySet define action to create a new property set for your build project.



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).

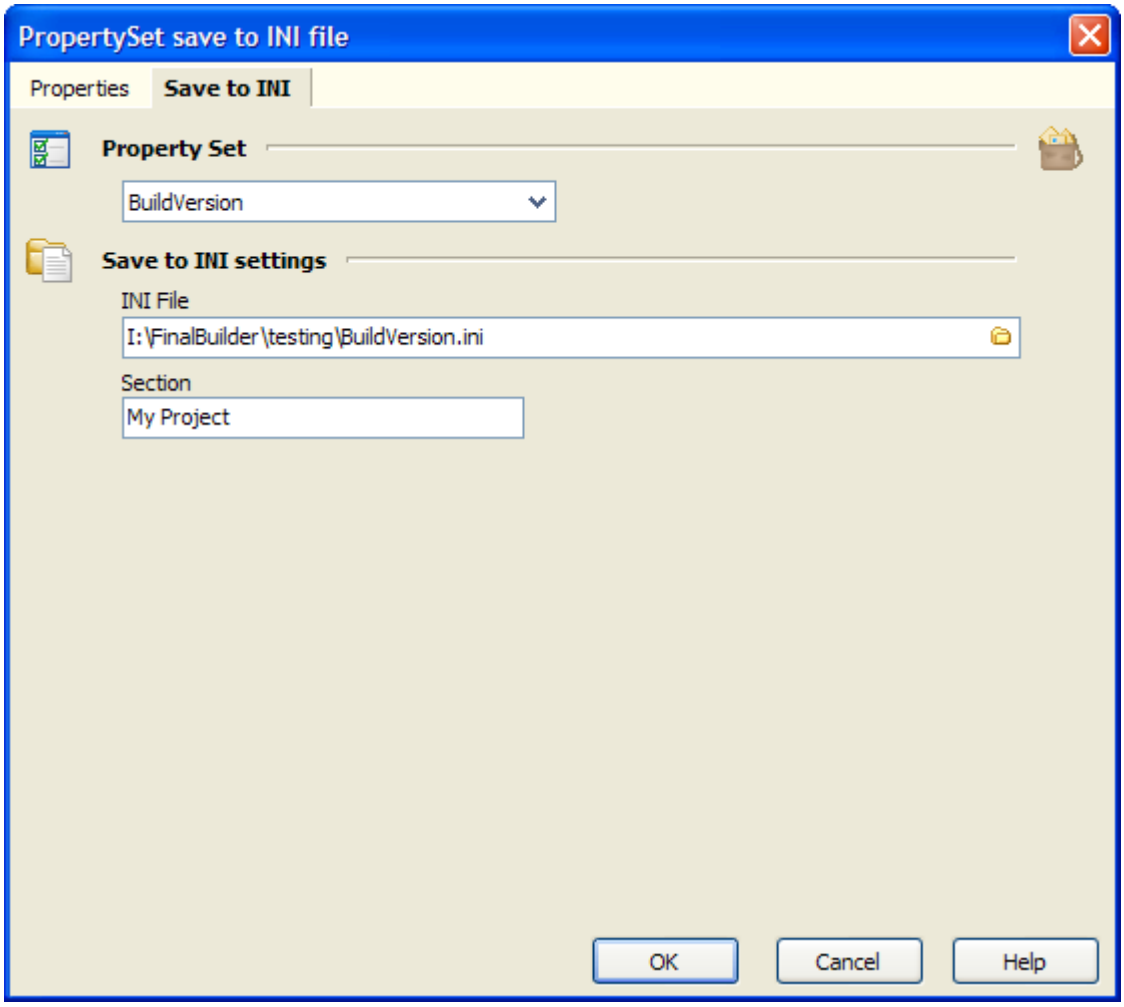
NOTE: If you change the name of a property set, you'll need to update any actions in your project which reference the PropertySet, as the reference is done via the PropertySet's name.

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.4 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.

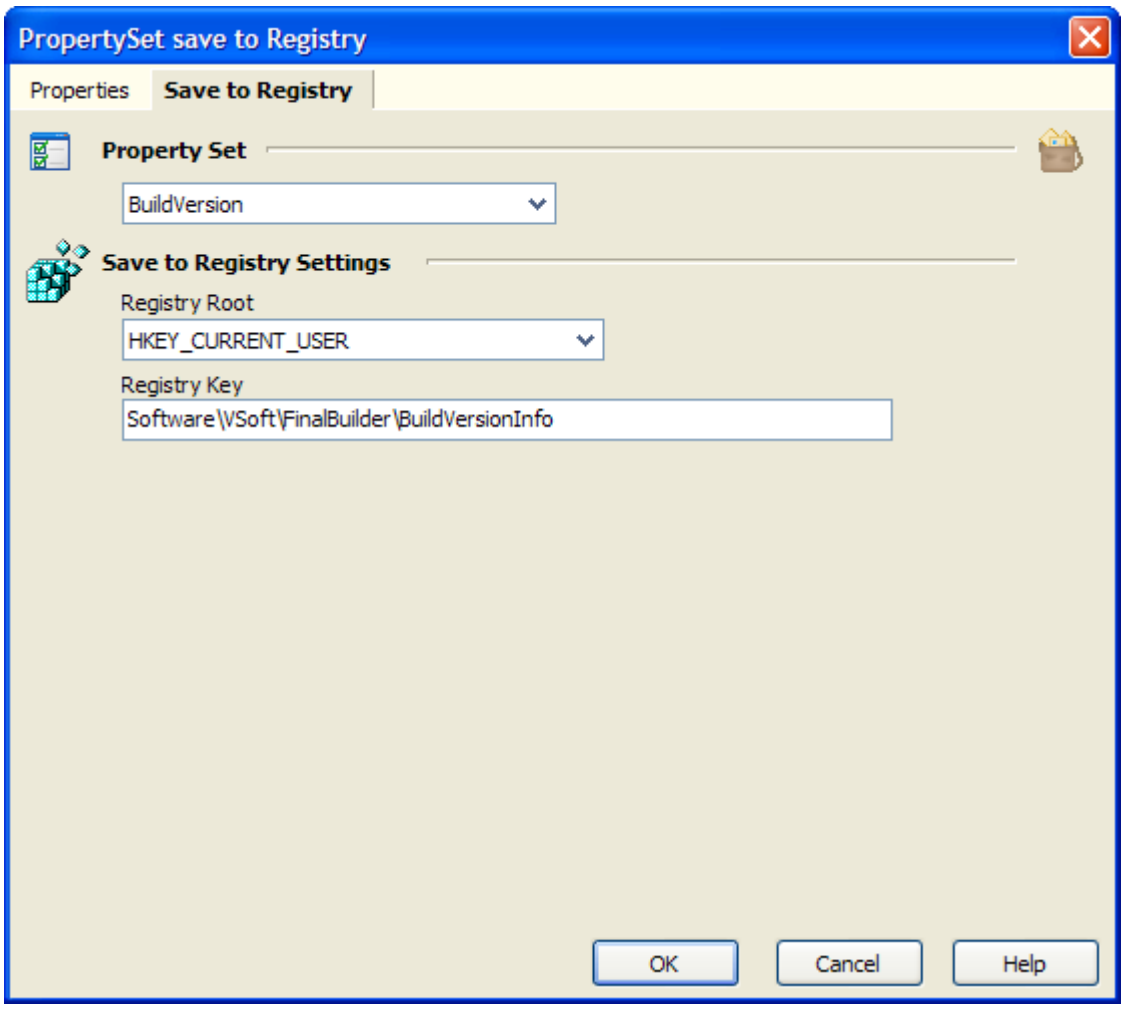
Section - The INI file section in which to store the PropertySet values

The INI file will automatically be created if it doesn't exists.
The Section in the INI file will also be created if it doesn't exist.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.5 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.

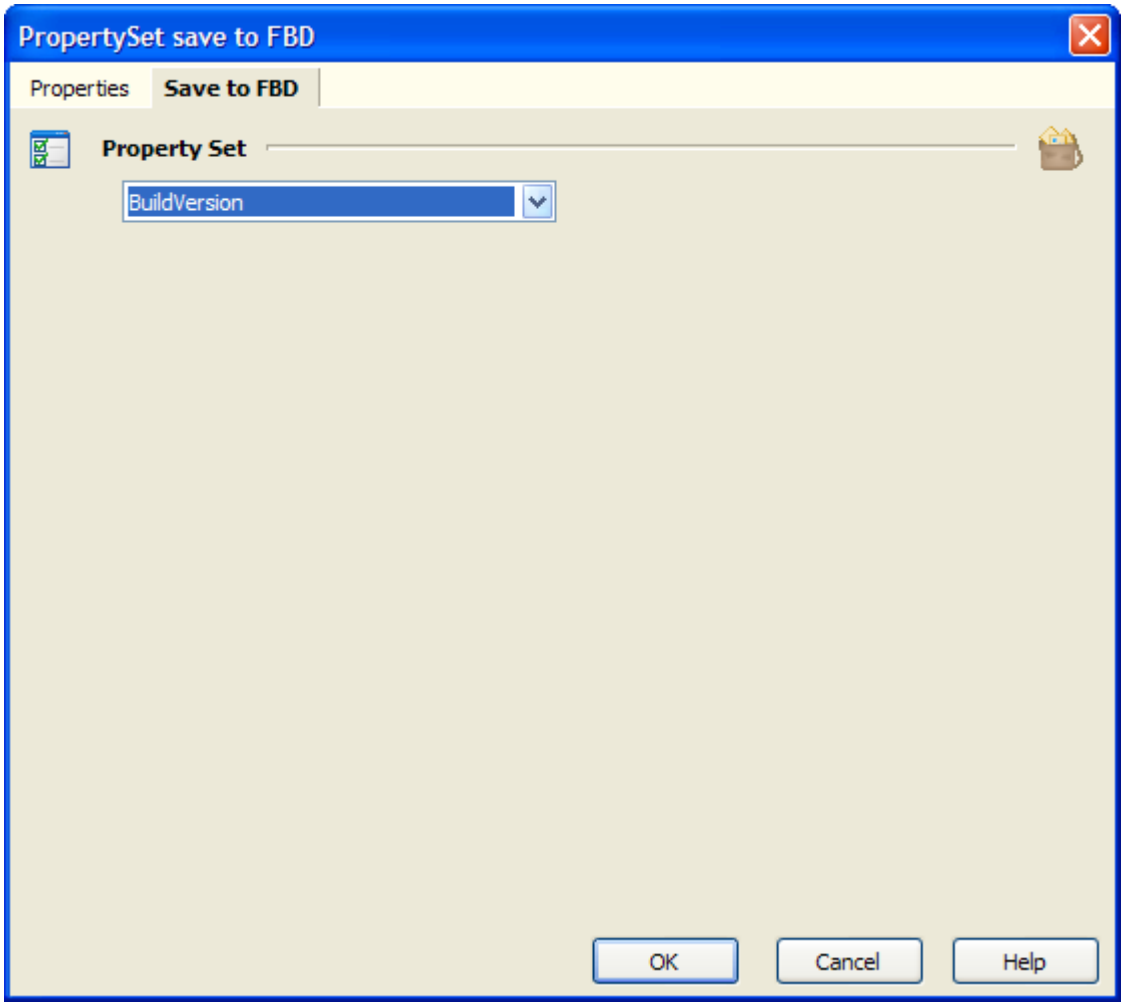
Registry Root - Select the Registry Root, usually Current_User or Local_Machine

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.6 PropertySet save to FBD

The PropertySet save to FBD action will persist the values of your property set to your project's FBD file.



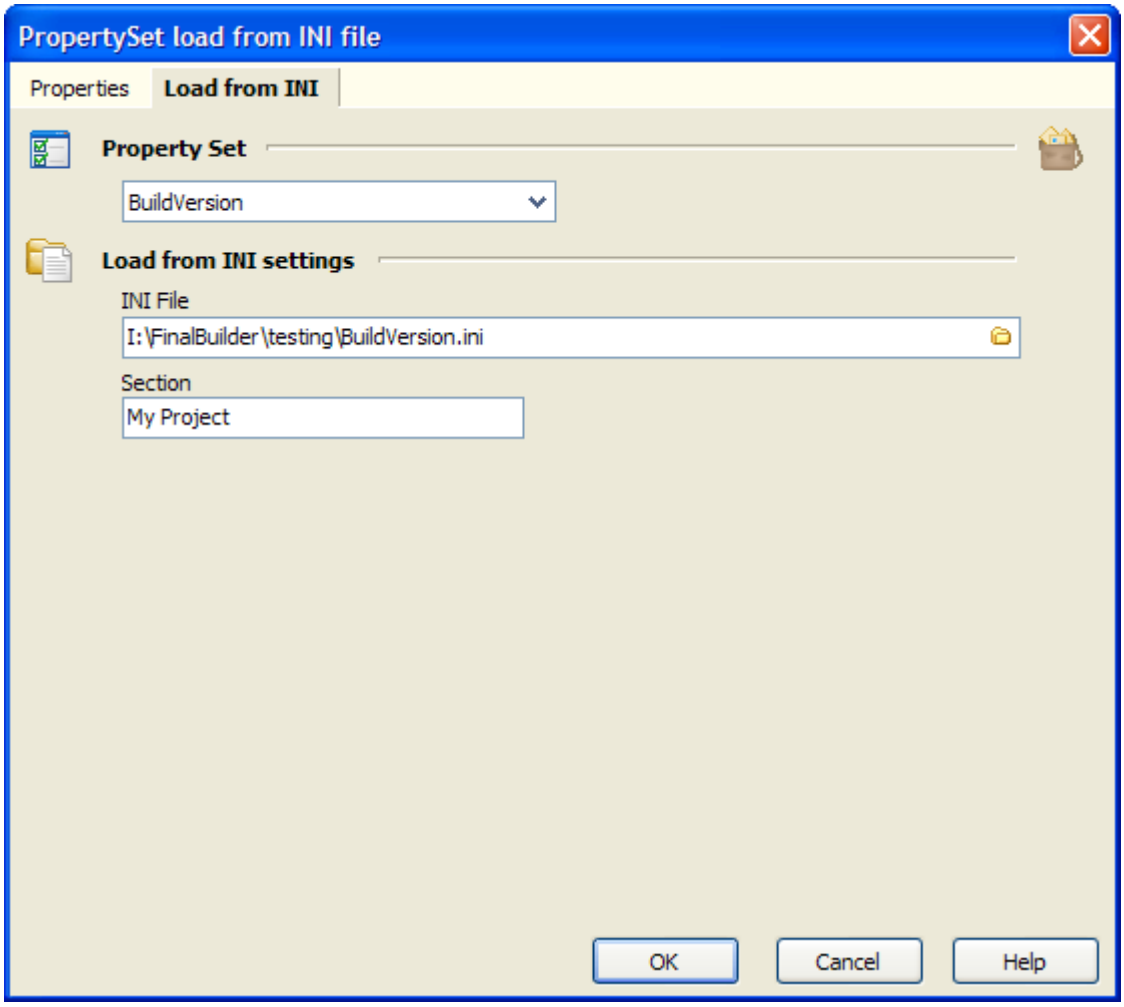
For more information on the FBD file, see Project Files & Other Files.

The FBD file will be automatically created if it doesn't already exist.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.7 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.

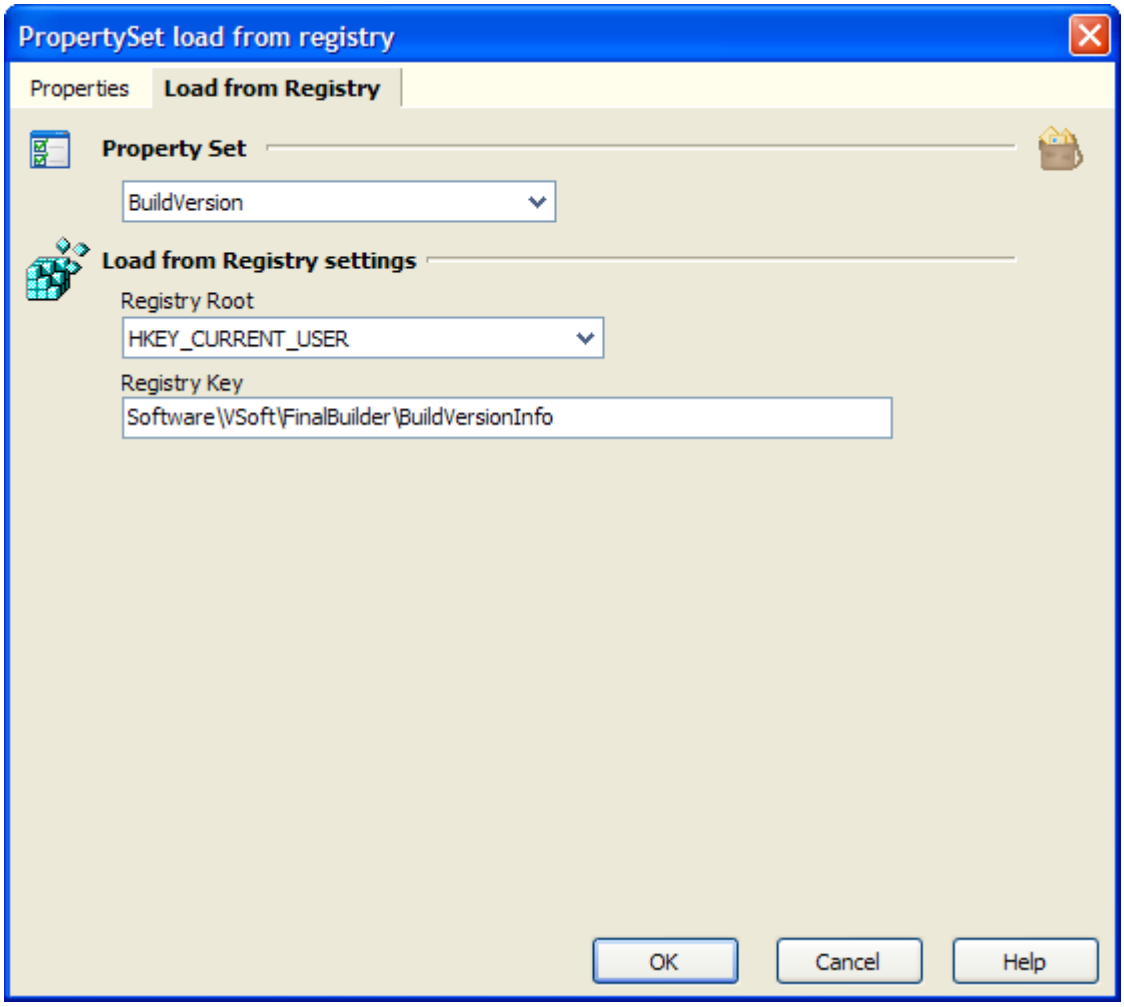
Section - The INI file section in which to load the PropertySet values

The action will fail if the INI file doesn't exists or if the Section doesn't exist in the INI file.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.8 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.

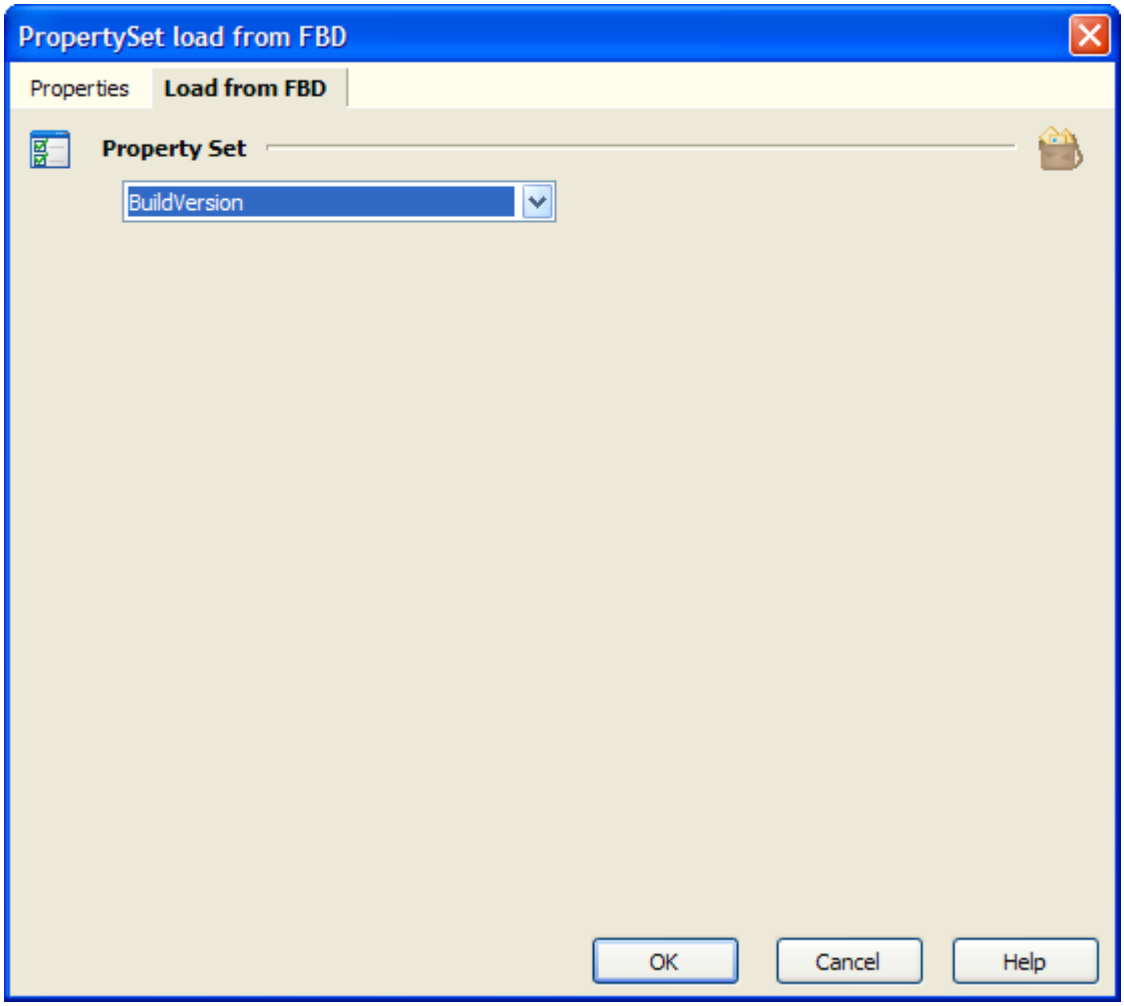
Registry Root - Select the Registry Root, usually Current_User or Local_Machine

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.9 PropertySet load from FBD

The PropertySet save to FBD action will persist the values of your property set to your project's FBD file.



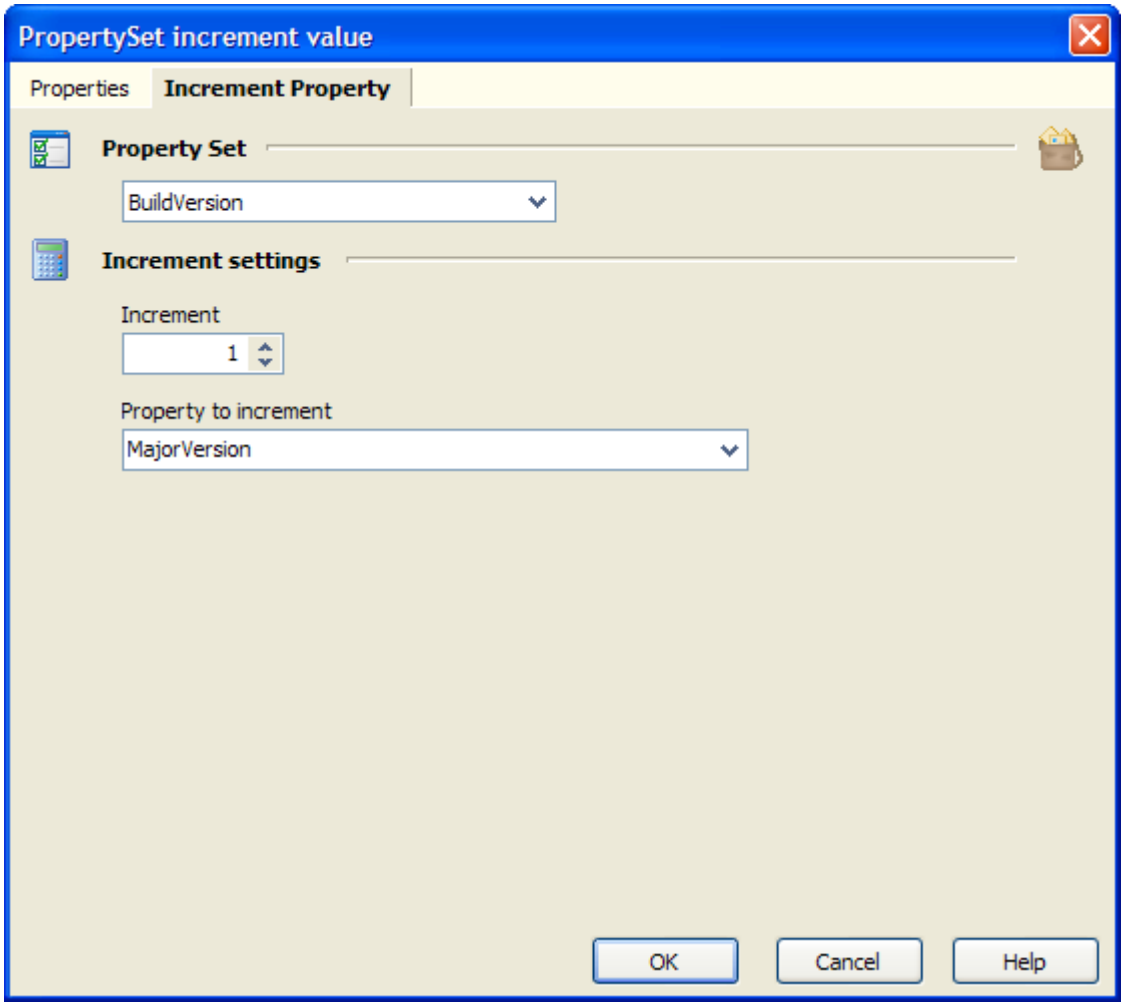
For more information on the FBD file, see Project Files & Other Files.

The action will fail if the FBD file doesn't exist.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.10 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.

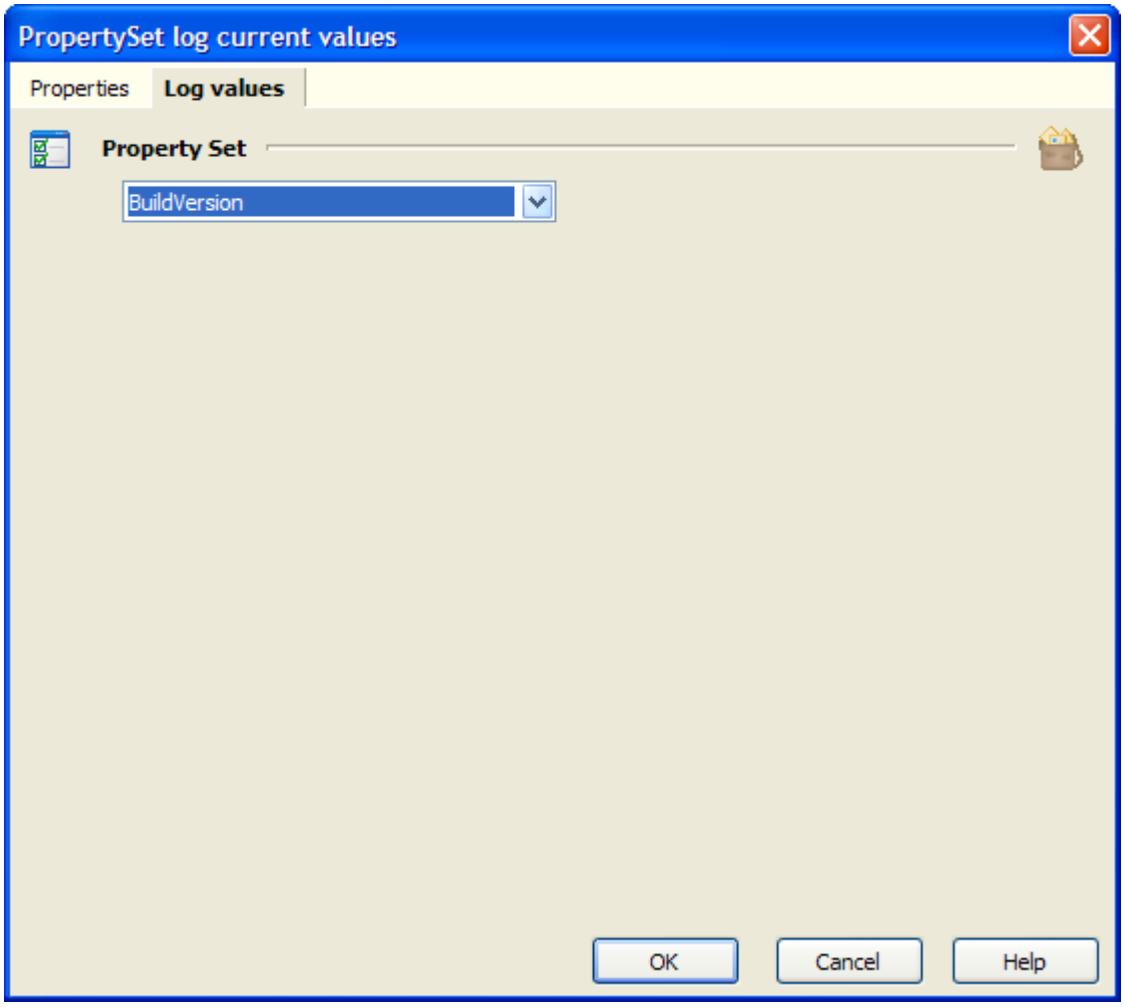
Increment - Select a positive or negative value which will be added to the integer property.

Property to increment - Select the property to increment. If the property is not an integer type the action will fail

© 2001-2006 VSoft Technologies Pty Ltd

5.3.11 PropertySet log values

The PropertySet log values action outputs the current values of the property set to the FinalBuilder log.

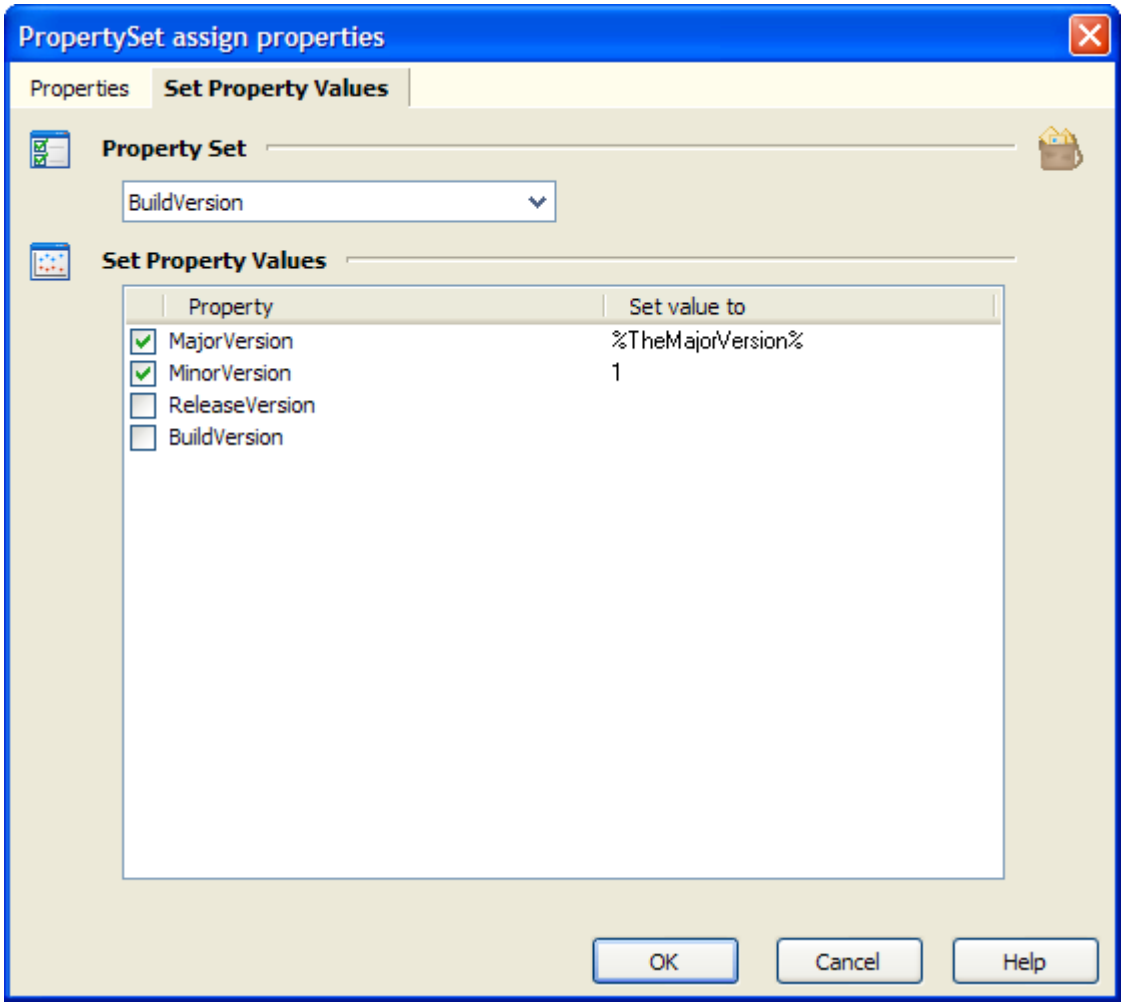


Simply select the Property Set, and during your build process the action will log each property name along with it's current value. This is useful during debugging, as well as to record the values of the property set in your build log.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.12 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.

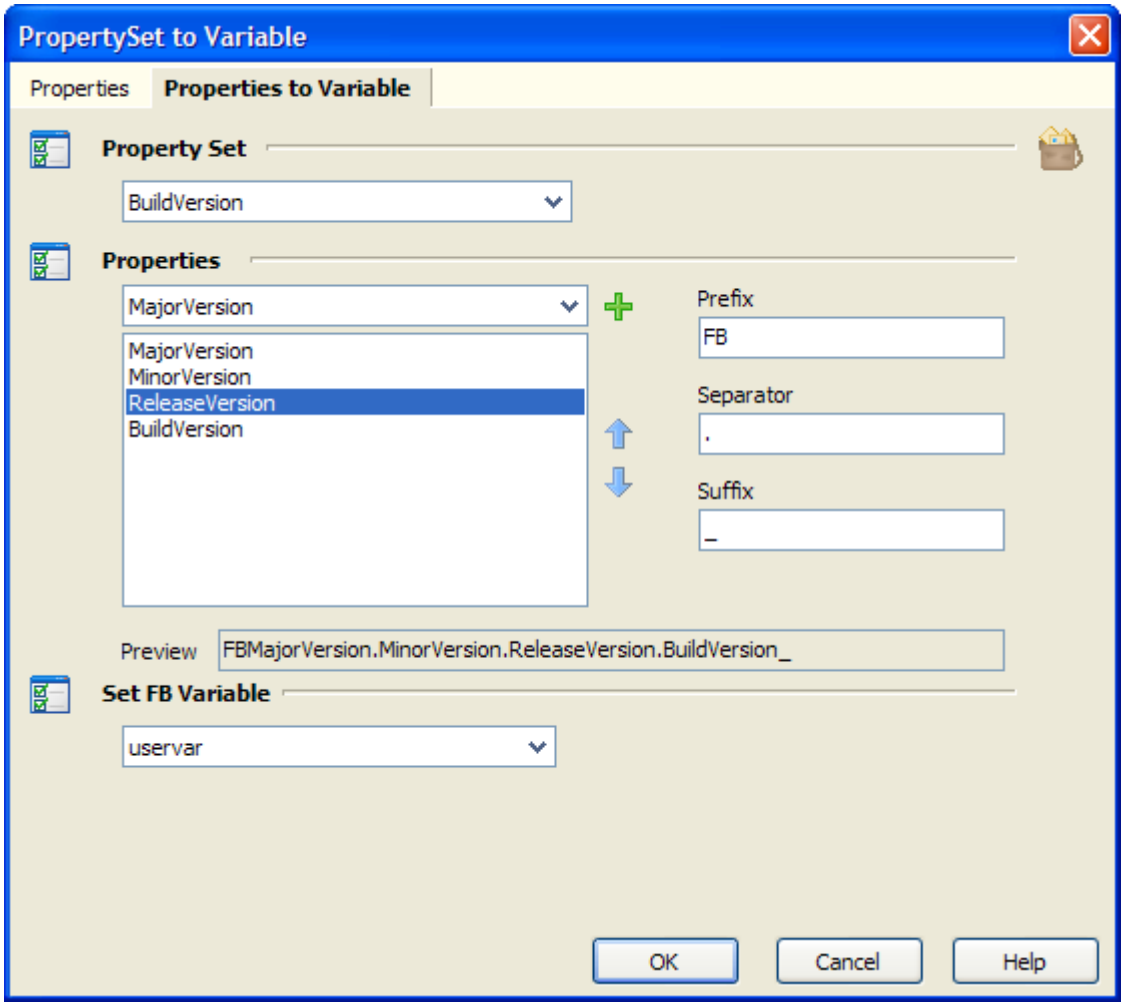


The checkbox denotes which properties will be assigned to new values when the action executes. Enter the new value to set the properties to in the "Set value to" column. The expression will be evaluated, therefore allowing FinalBuilder variables.

© 2001-2006 VSoft Technologies Pty Ltd

5.3.13 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.

Preview - the preview displays what the resultant text string will be according to the properties selected, prefix, separator, and sufix values.

Set FB Variable - the resultant text string will be saved in the selected FB variable.

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.4 File Sets

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 Exlude 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 process. See Accessing FileSets to see how other actions in the build process can use the list of files selected.

© 2001-2006 VSoft Technologies Pty Ltd

5.4.1 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

© 2001-2006 VSoft Technologies Pty Ltd

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

© 2001-2006 VSoft Technologies Pty Ltd

5.4.2 FileSet Define

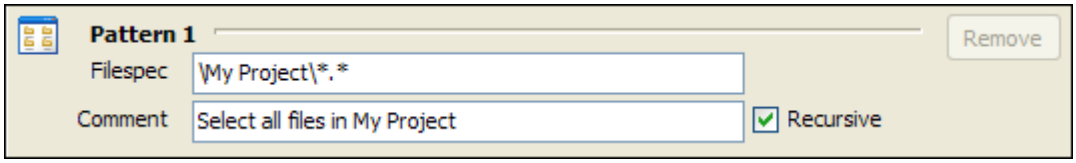
The FileSet Define action is used to create a new FileSet. The FileSet Define action has the following properties:

File Set 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 used
Default Separator - the default separator style used when outputting more than one file
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

5.4.2.1 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 asterix "*" 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

© 2001-2006 VSoft Technologies Pty Ltd

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 filestructure:

```
<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. Exlude 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\

© 2001-2006 VSoft Technologies Pty Ltd

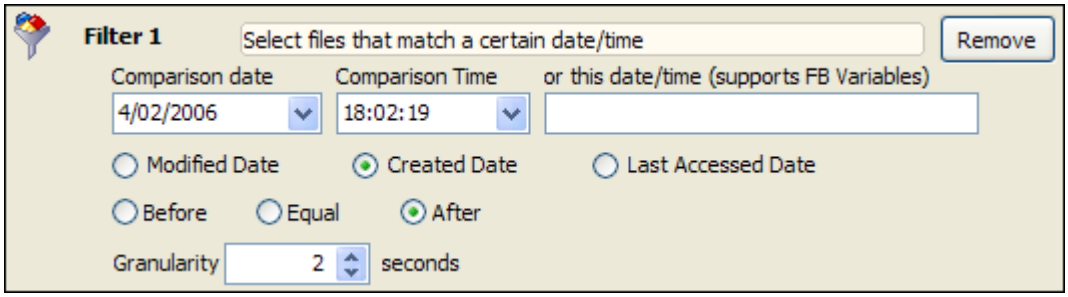
5.4.2.2 FileSet Filters

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:

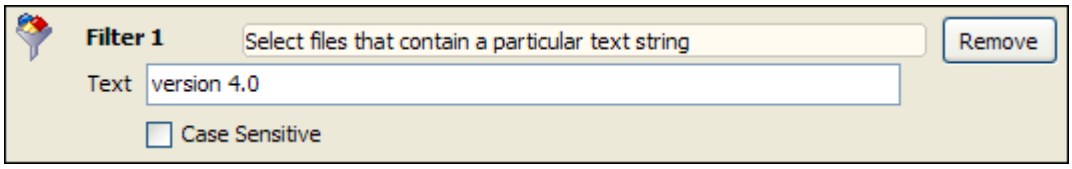
- Date/Time
- Contents
- Attributes
- Size

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



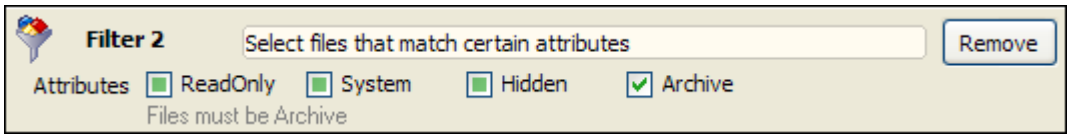
This example shows that only files created after 4/2/2006 12:02:19 will be added to the FileSet. The granularity allows slight differences in the time you specify and the time retrieved from the Windows file system - a granularity of 2-5 seconds is recommended.

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.

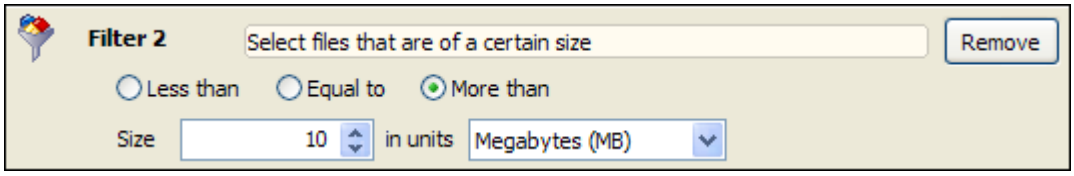
Attributes Filter
The Attributes filter allows you to filter the FileSet files depending on the file system attributes of the files.



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

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.4.3 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

© 2001-2006 VSoft Technologies Pty Ltd

5.4.4 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.

© 2001-2006 VSoft Technologies Pty Ltd

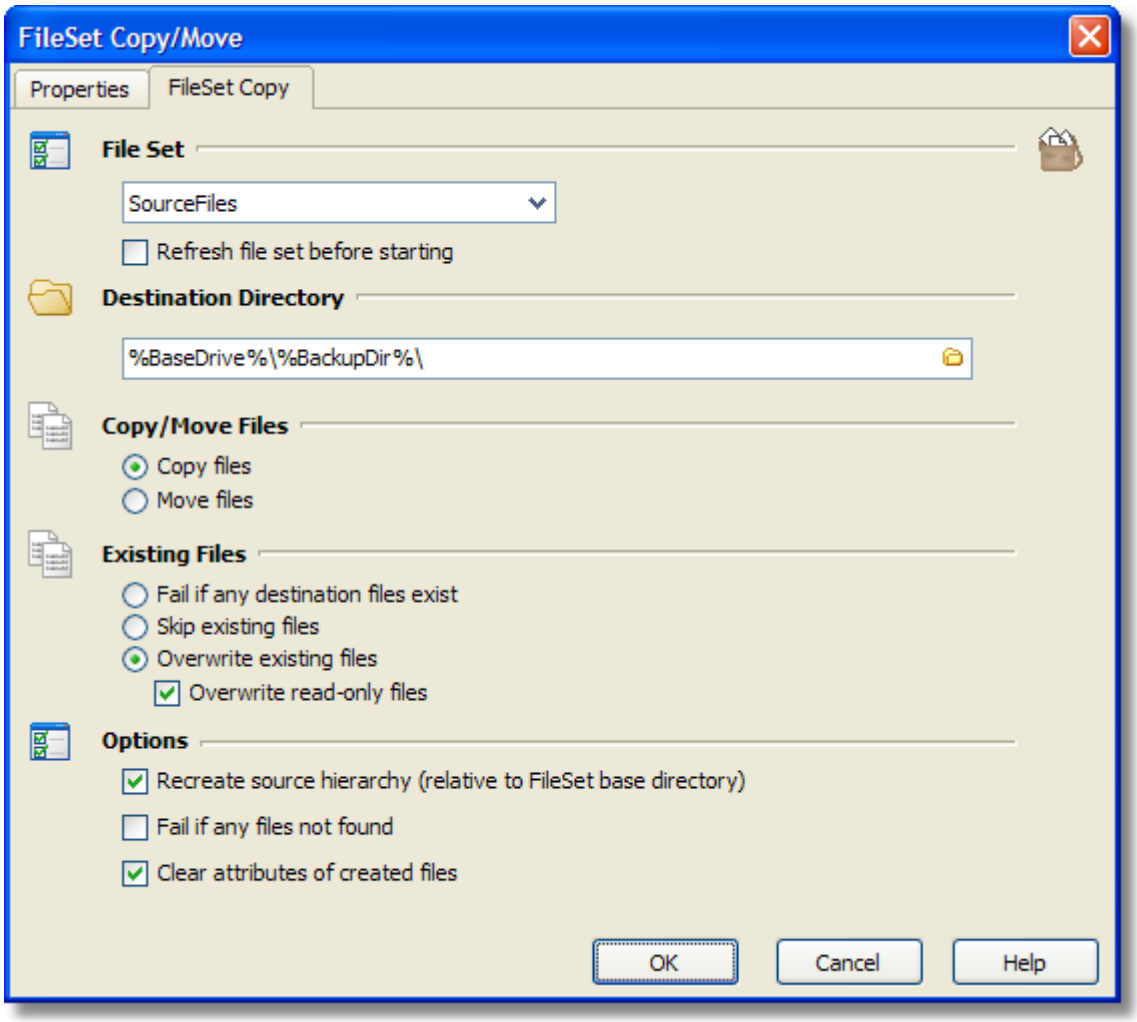
5.4.5 FileSet Refresh

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

© 2001-2006 VSoft Technologies Pty Ltd

5.4.6 Copy/Move FileSet 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 FileSet to use

Refresh FileSet before starting - if this option is set the FileSet will be refreshed. 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

© 2001-2006 VSoft Technologies Pty Ltd

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

© 2001-2006 VSoft Technologies Pty Ltd

5.4.7 FileSet Delete

The FileSet Delete action will delete all files selected in the FileSet from your disk drive.

Log Each File Deleted - record each file deleted in the log

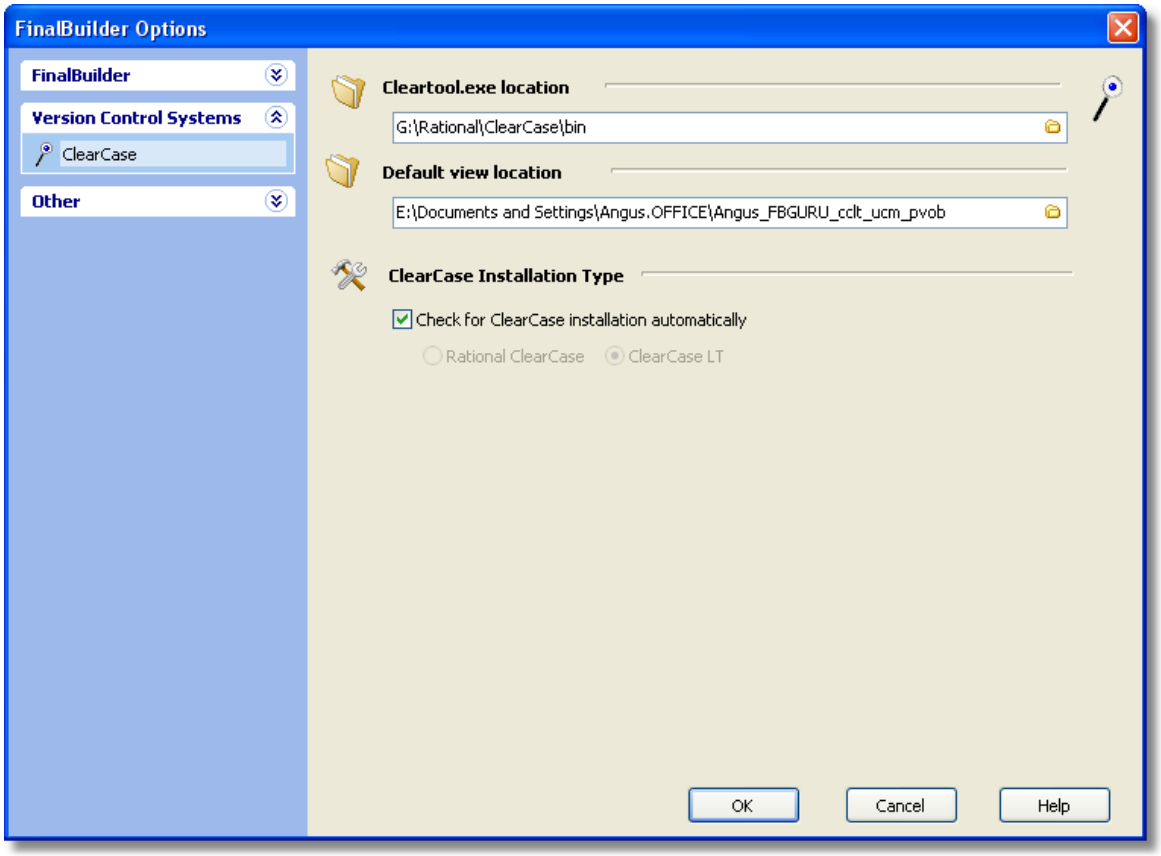
Ignore Delete Failures - continue deleting the rest of the files in the fileset if there was an error deleting any file, otherwise the action will stop and fail as soon as an error occurs deleting a file.

© 2001-2006 VSoft Technologies Pty Ltd

5.5 Version Control Actions

5.5.1 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 overriden manually if desired. ClearCase Installation Type enables and disables some options which are only available in ClearCase or ClearCase LT, respectively.

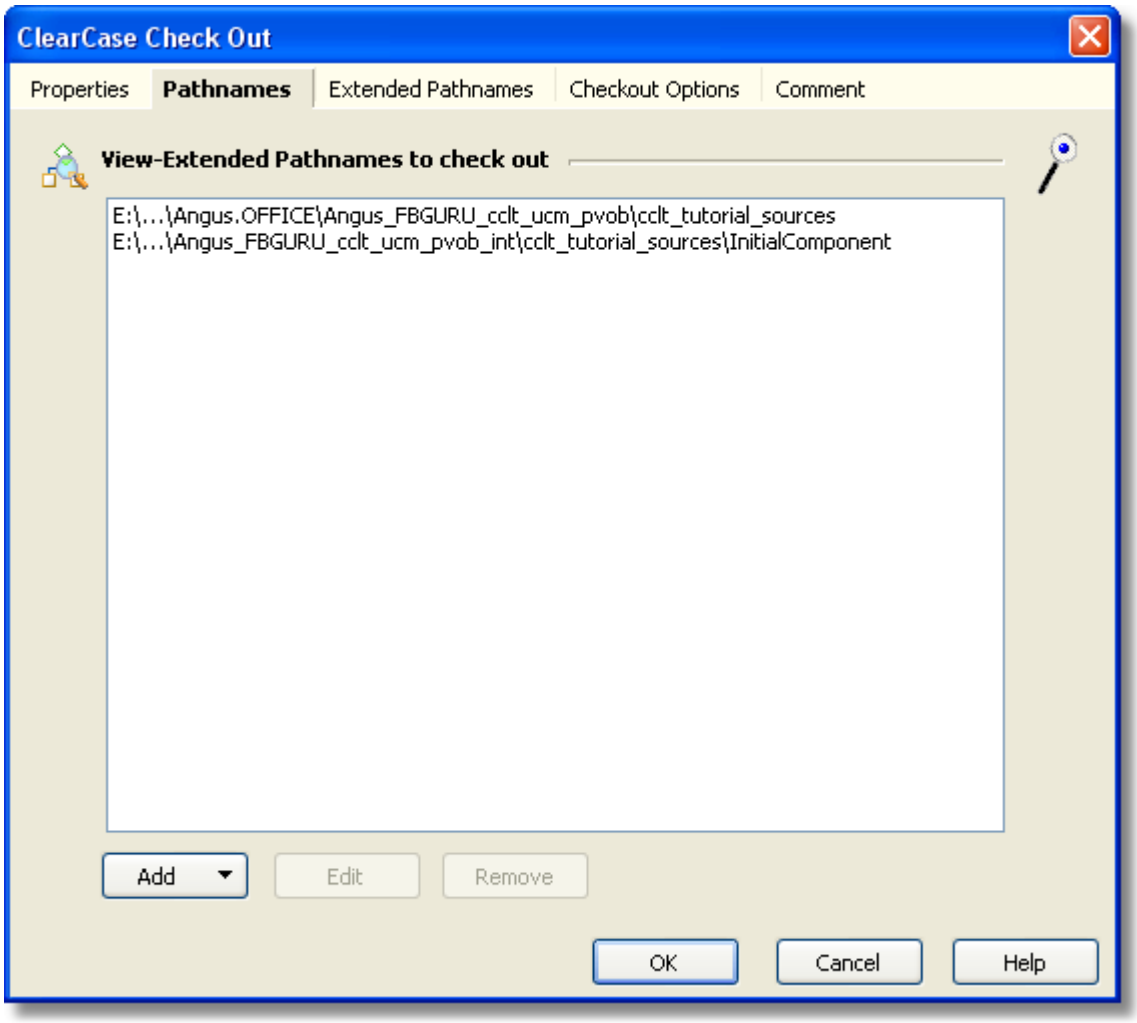
5.5.1.1 ClearCase Pathnames

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:

© 2001-2006 VSoft Technologies Pty Ltd

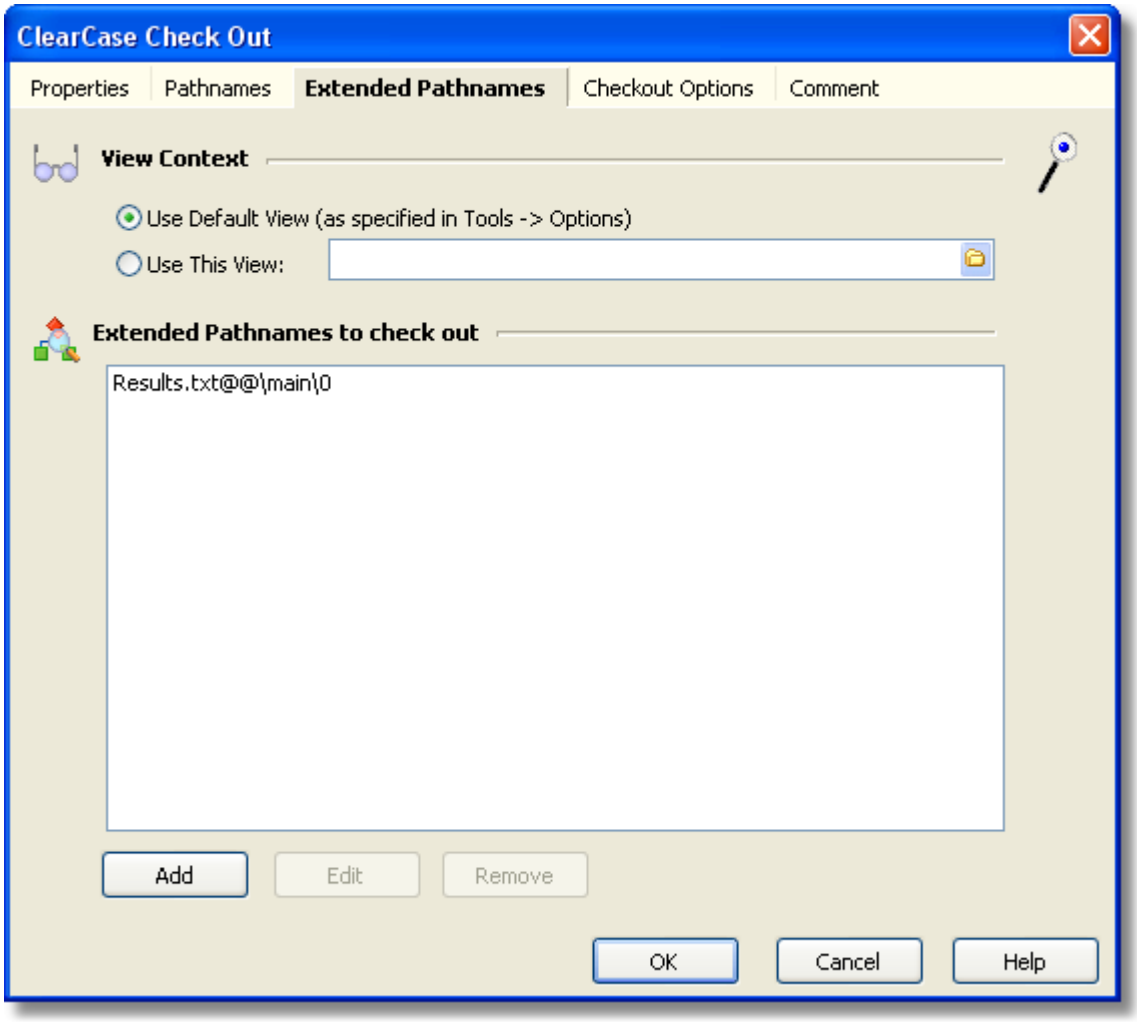


Hover the mouse over a pathname to see the full path.

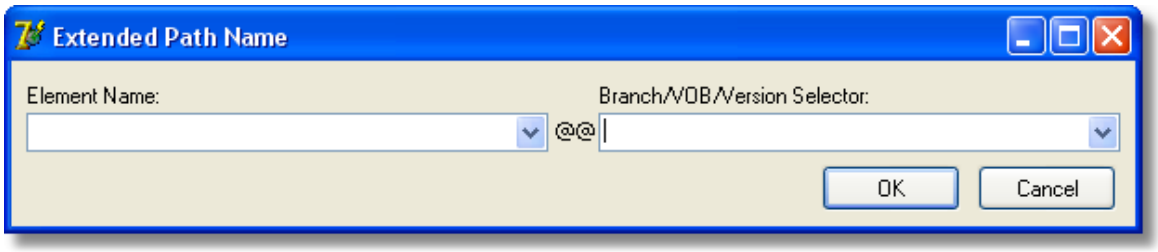
Version- and VOB- Extended Pathnames can be found under the '**Extended Pathnames**' tab:

Note: Extended Pathnames are not available in ClearCase LT.

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.5.1.2 ClearCase Object Selector

The Object Selector allows some ClearCase actions to target abstract ClearCase objects.

Objects that can be targeted include:

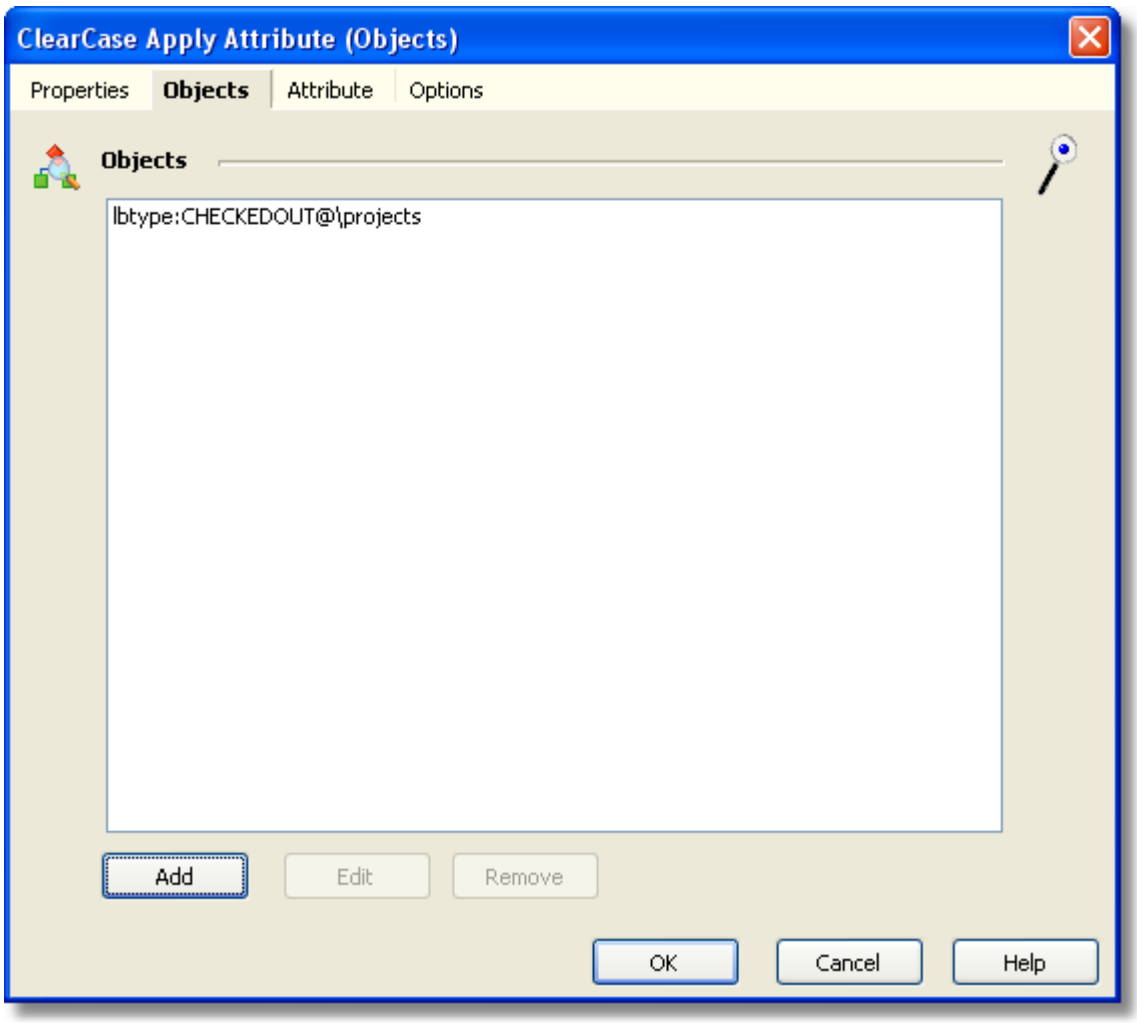
- VOBs
- Attribute Types

© 2001-2006 VSoft Technologies Pty Ltd

- Branches
- Elements
- Hyperlinks
- Labels
- Triggers
- Pools
- Object IDs

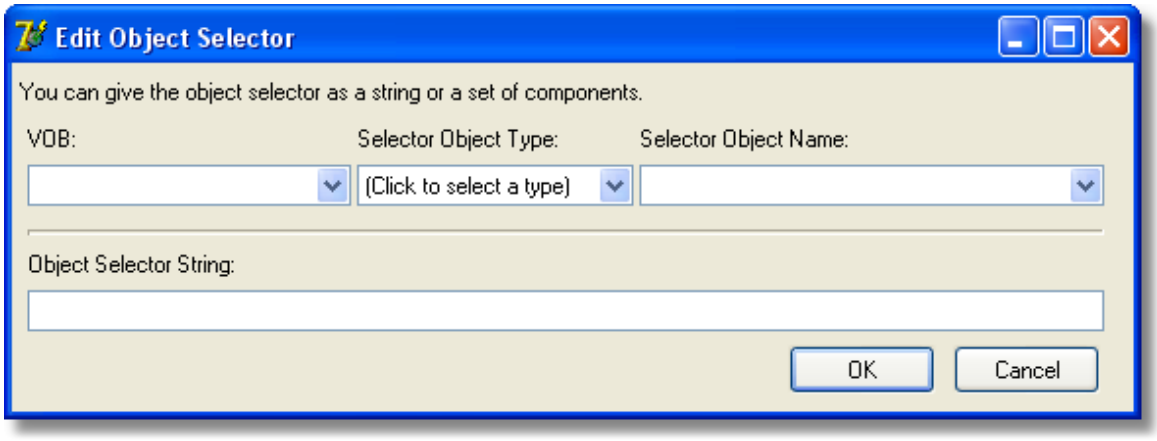
Also,
- UCM Activities
- UCM Baselines
- UCM Components
- UCM Folders
- UCM Projects
- UCM Streams

In the screenshot below, a label type has been selected to apply an attribute:



Clicking the Add button brings up the object selection dialog:

© 2001-2006 VSoft Technologies Pty Ltd



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.

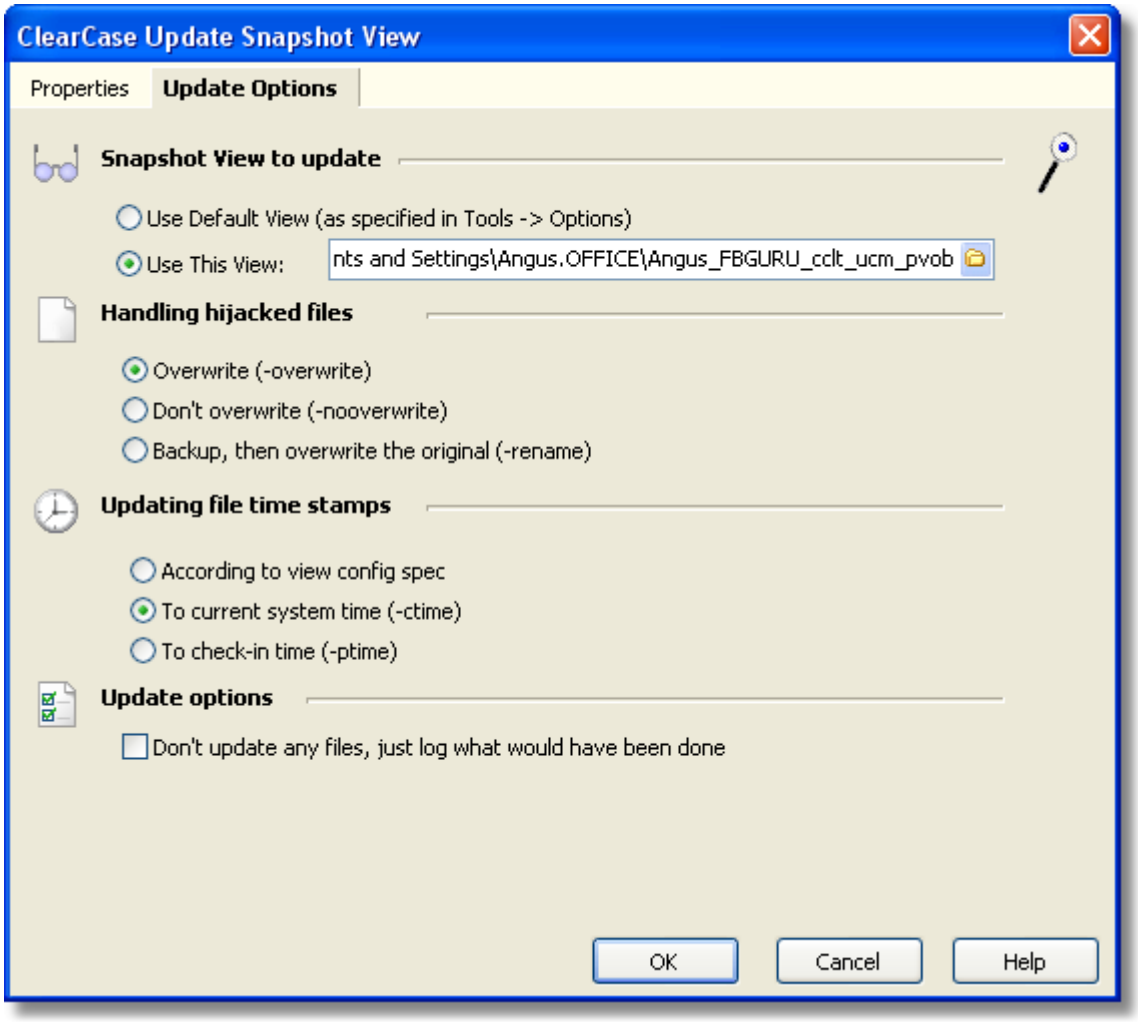
5.5.1.3 Base ClearCase

5.5.1.3.1 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*.)

© 2001-2006 VSoft Technologies Pty Ltd



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.

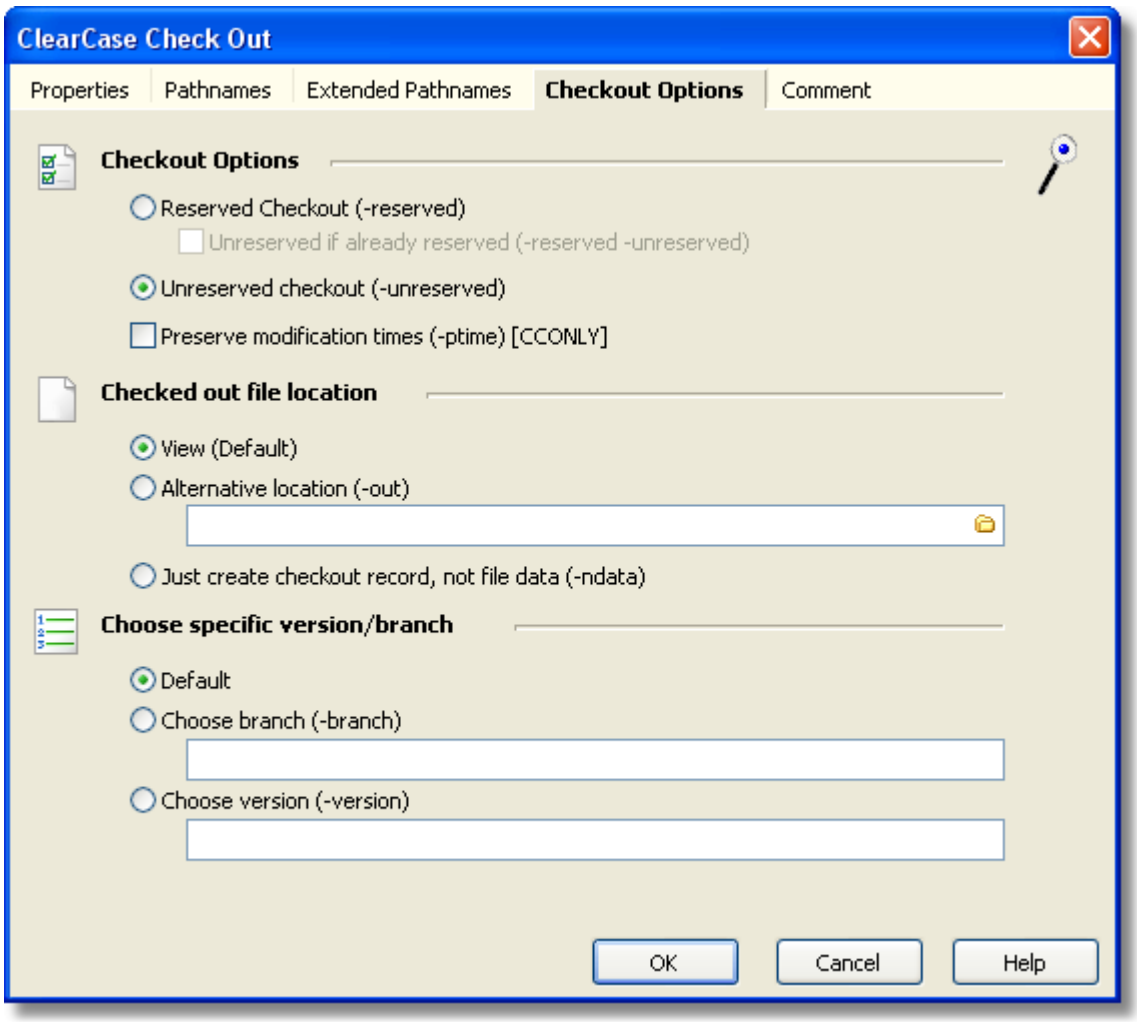
© 2001-2006 VSoft Technologies Pty Ltd

5.5.1.3.2 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

© 2001-2006 VSoft Technologies Pty Ltd

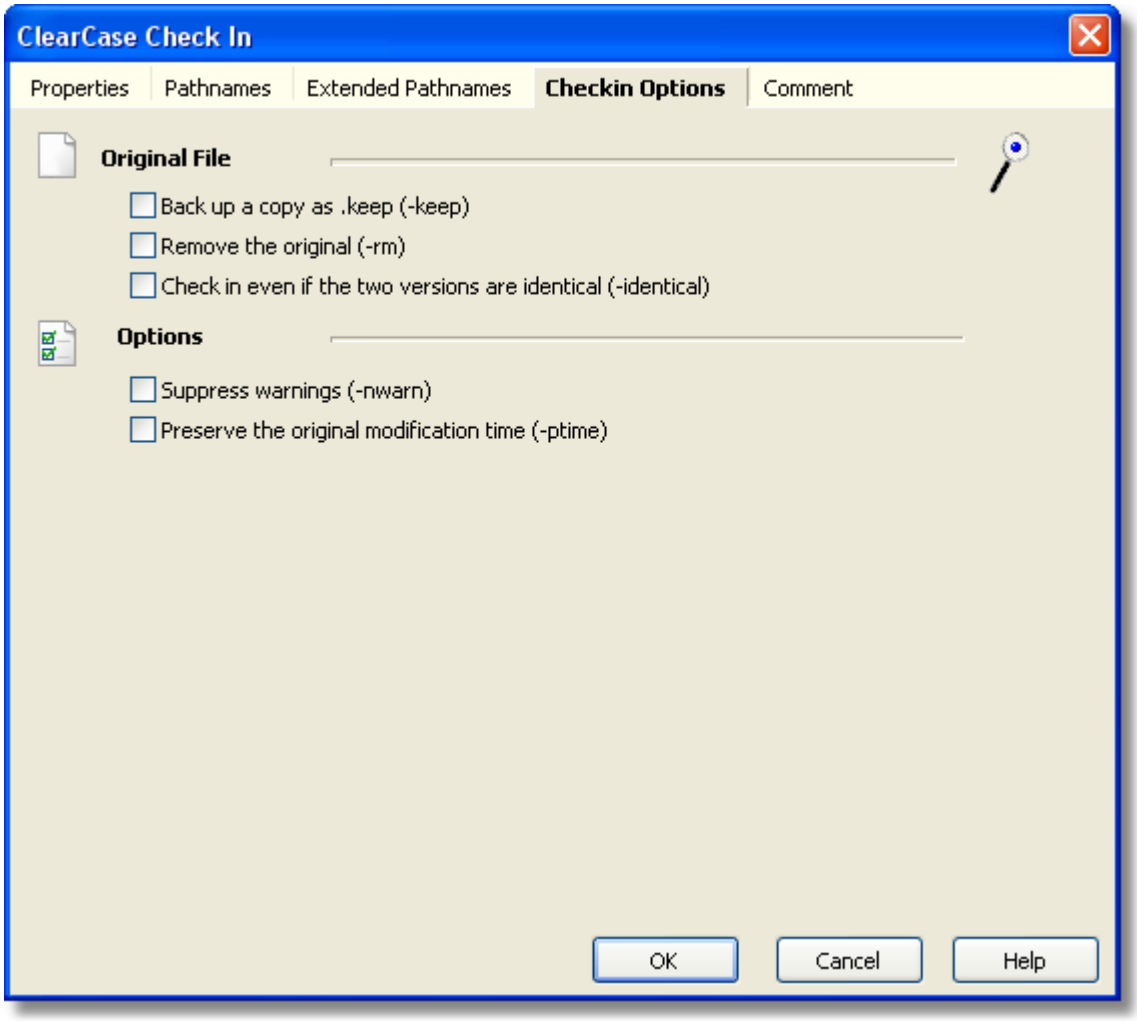
checked out from dynamic views. This option is not available on ClearCase LT.

5.5.1.3.3 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.

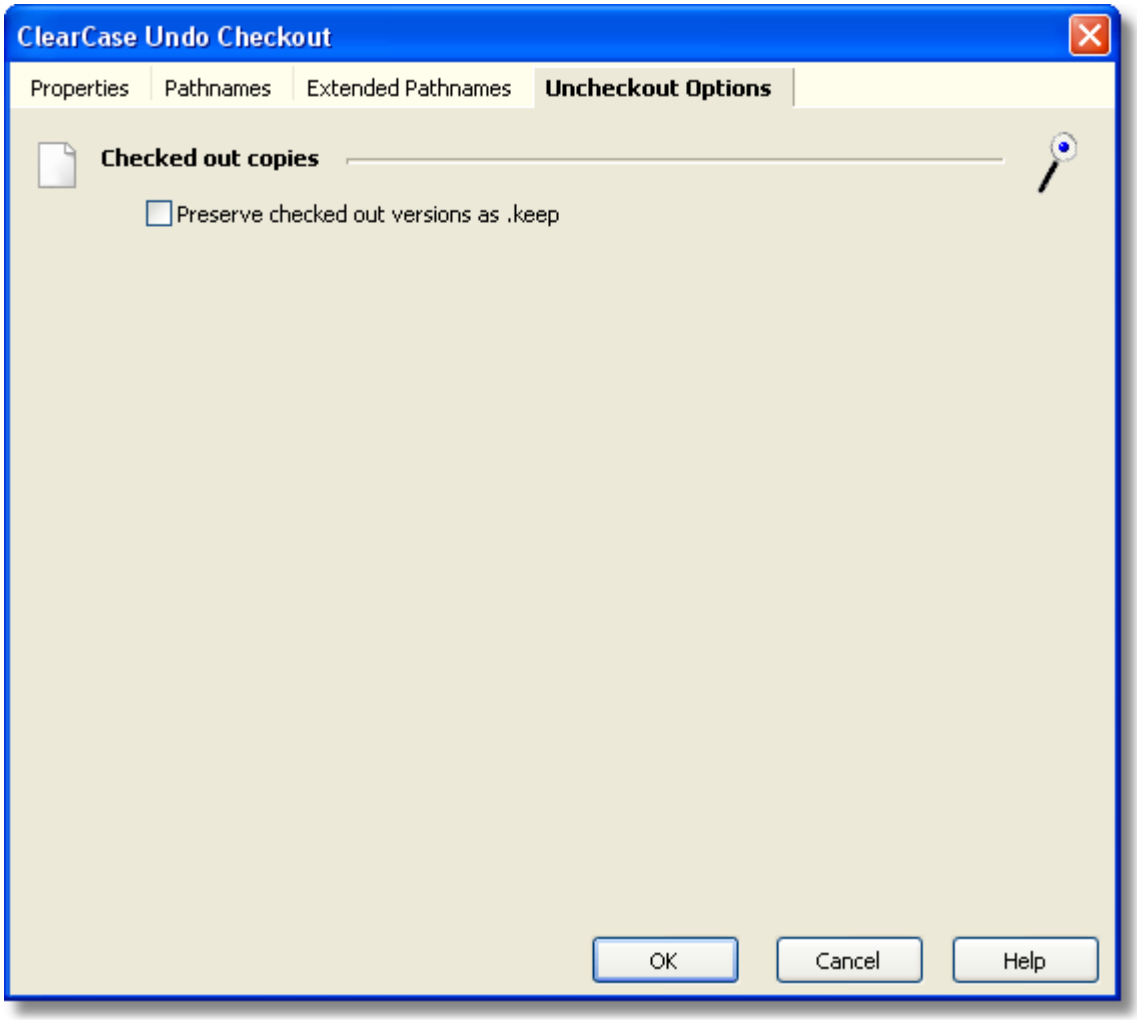
© 2001-2006 VSoft Technologies Pty Ltd

5.5.1.3.4 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.

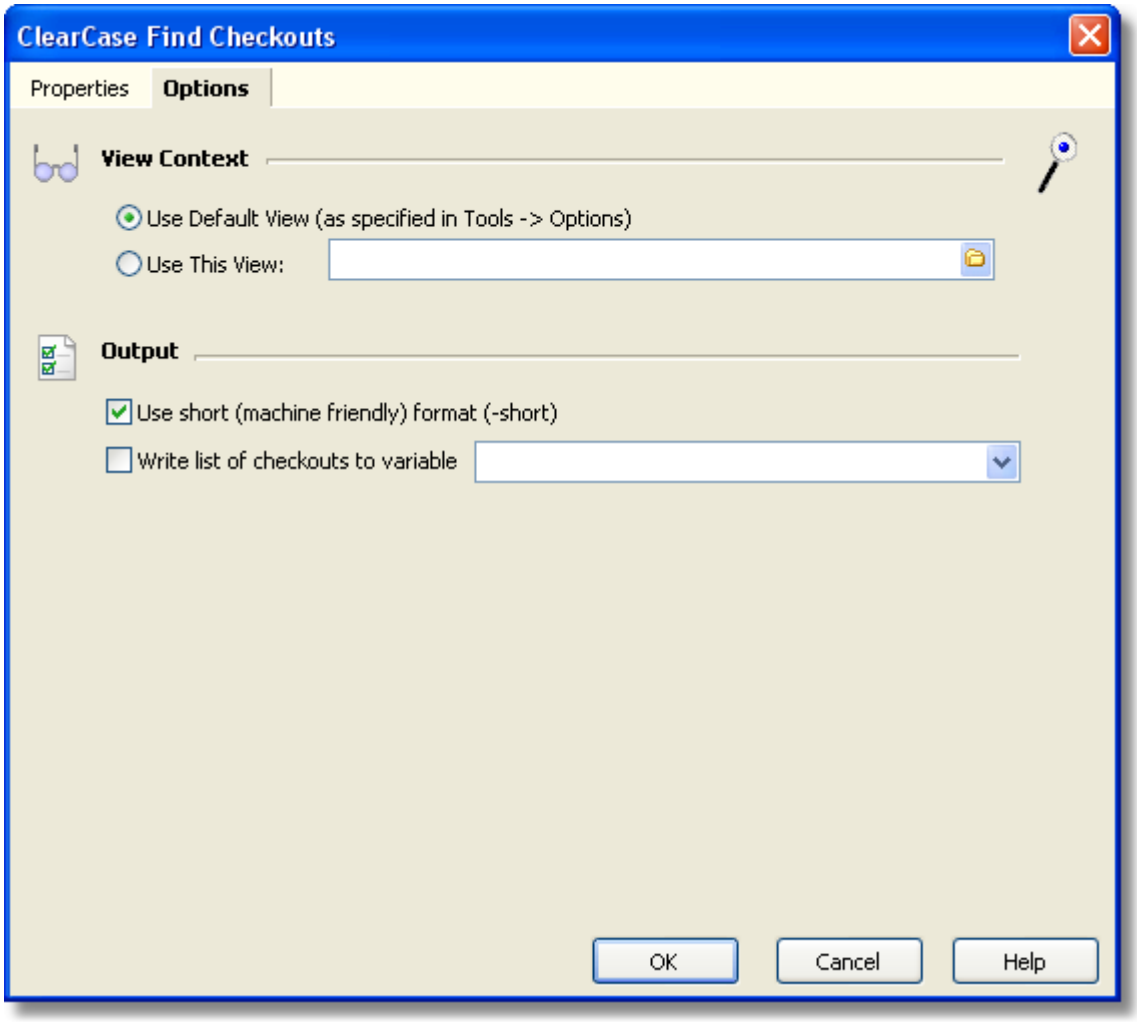
5.5.1.3.5 ClearCase Find Checkouts Action

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

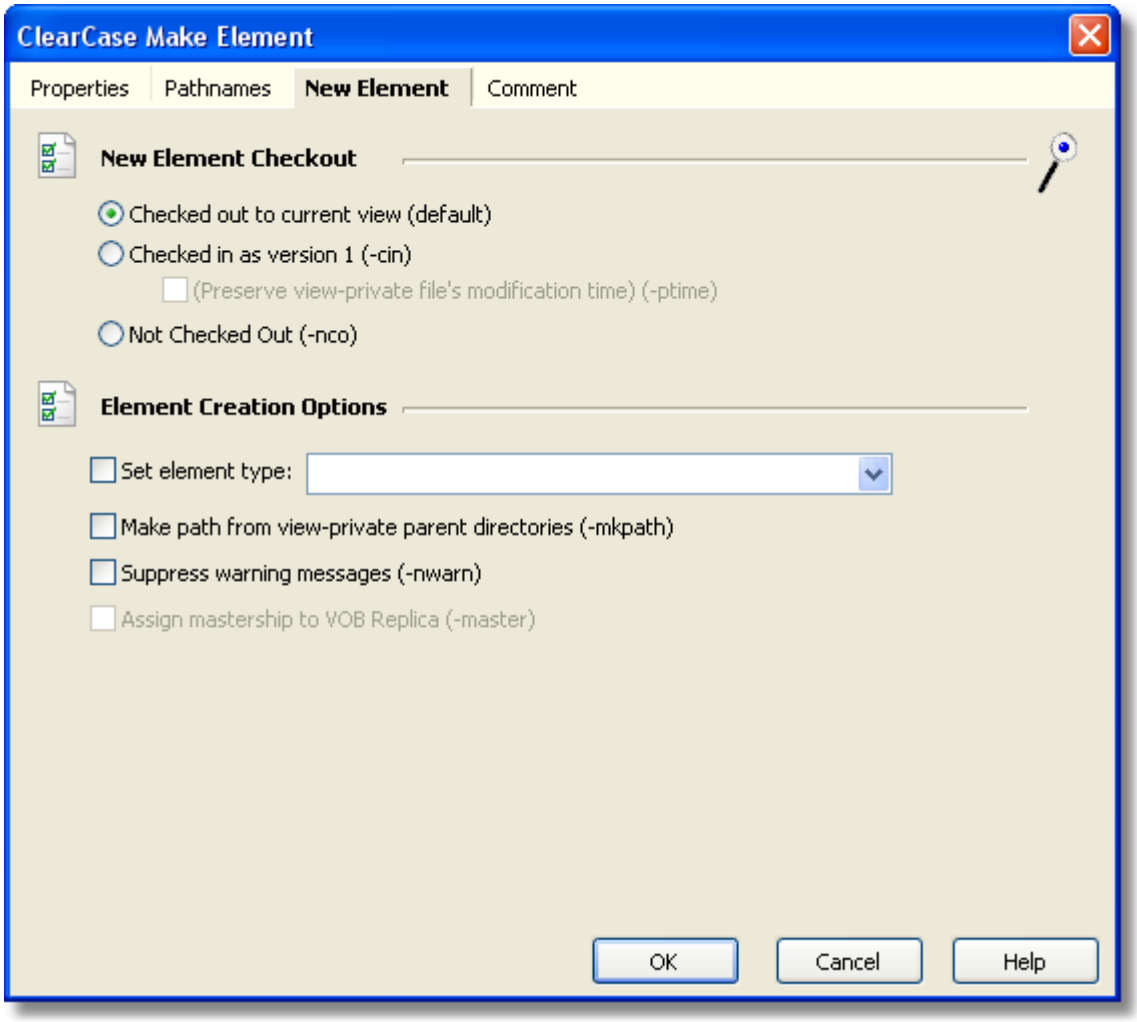
5.5.1.3.6 ClearCase Make Element Action

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

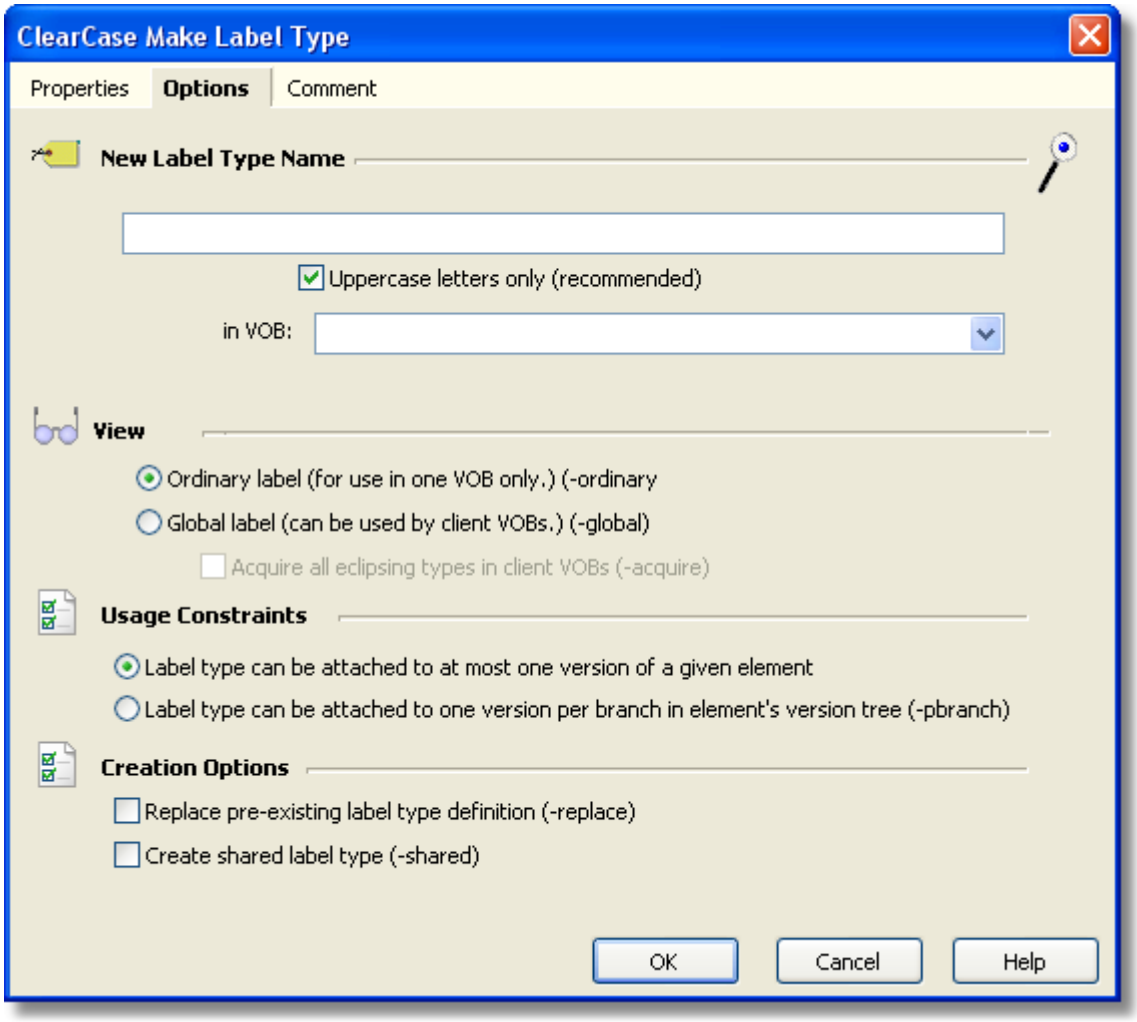
Assign Mastership to VOB Replica
This option is not available in ClearCase LT.

5.5.1.3.7 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*.)

© 2001-2006 VSoft Technologies Pty Ltd



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.

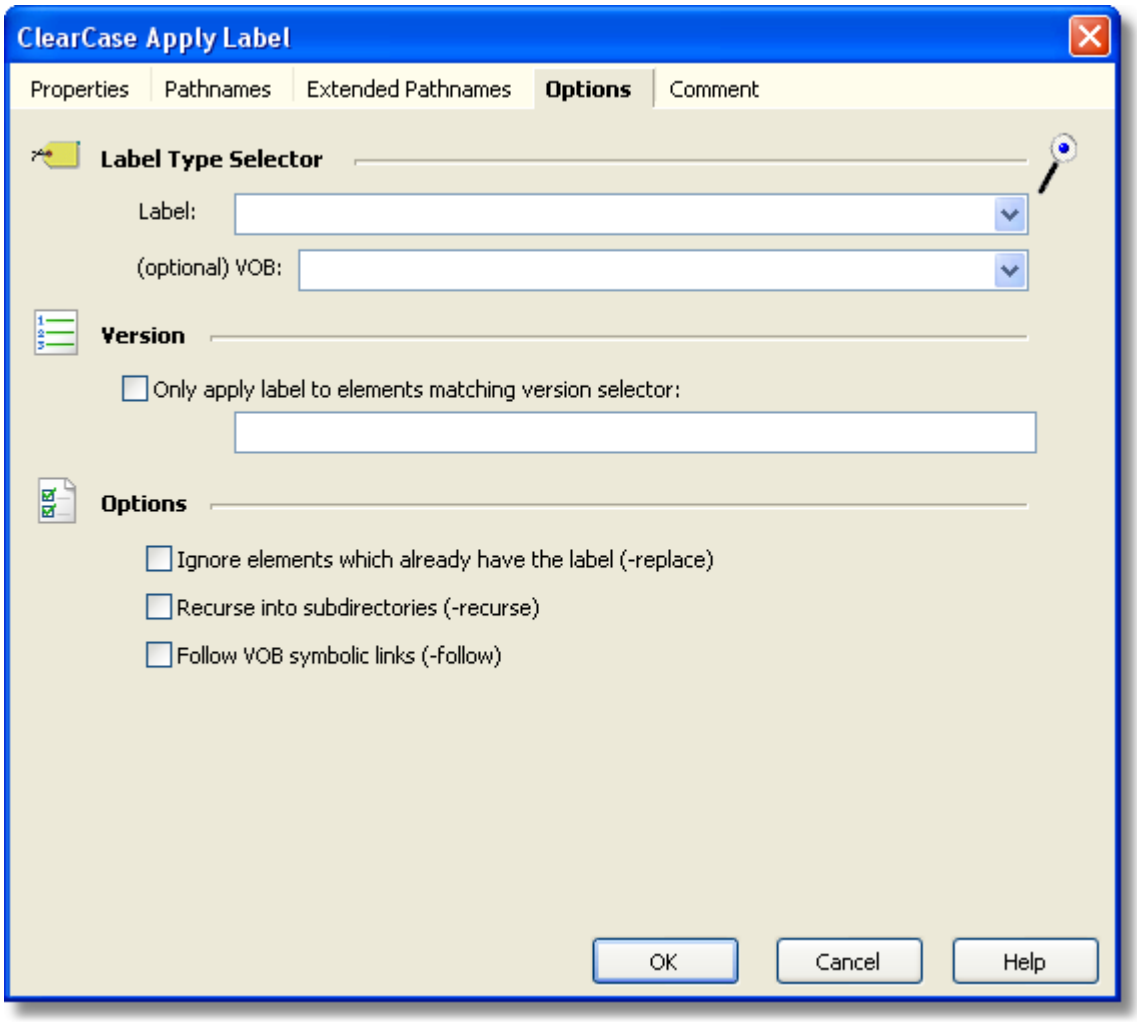
© 2001-2006 VSoft Technologies Pty Ltd

5.5.1.3.8 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.

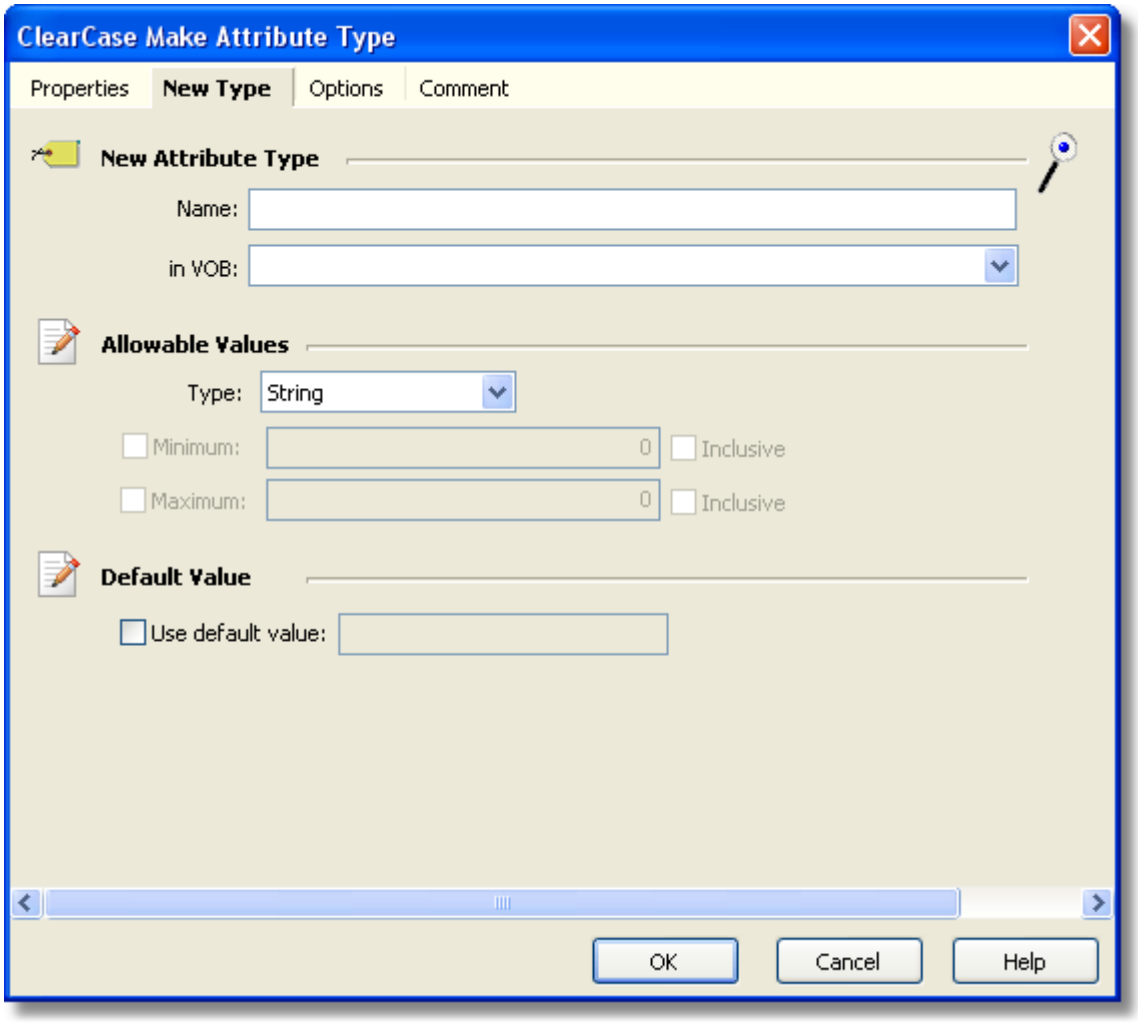
5.5.1.3.9 ClearCase Make Attribute Type Action

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 mkattype command, see the ClearCase manual page (type

© 2001-2006 VSoft Technologies Pty Ltd

cleartool man mkattype.)



New Attribute Type

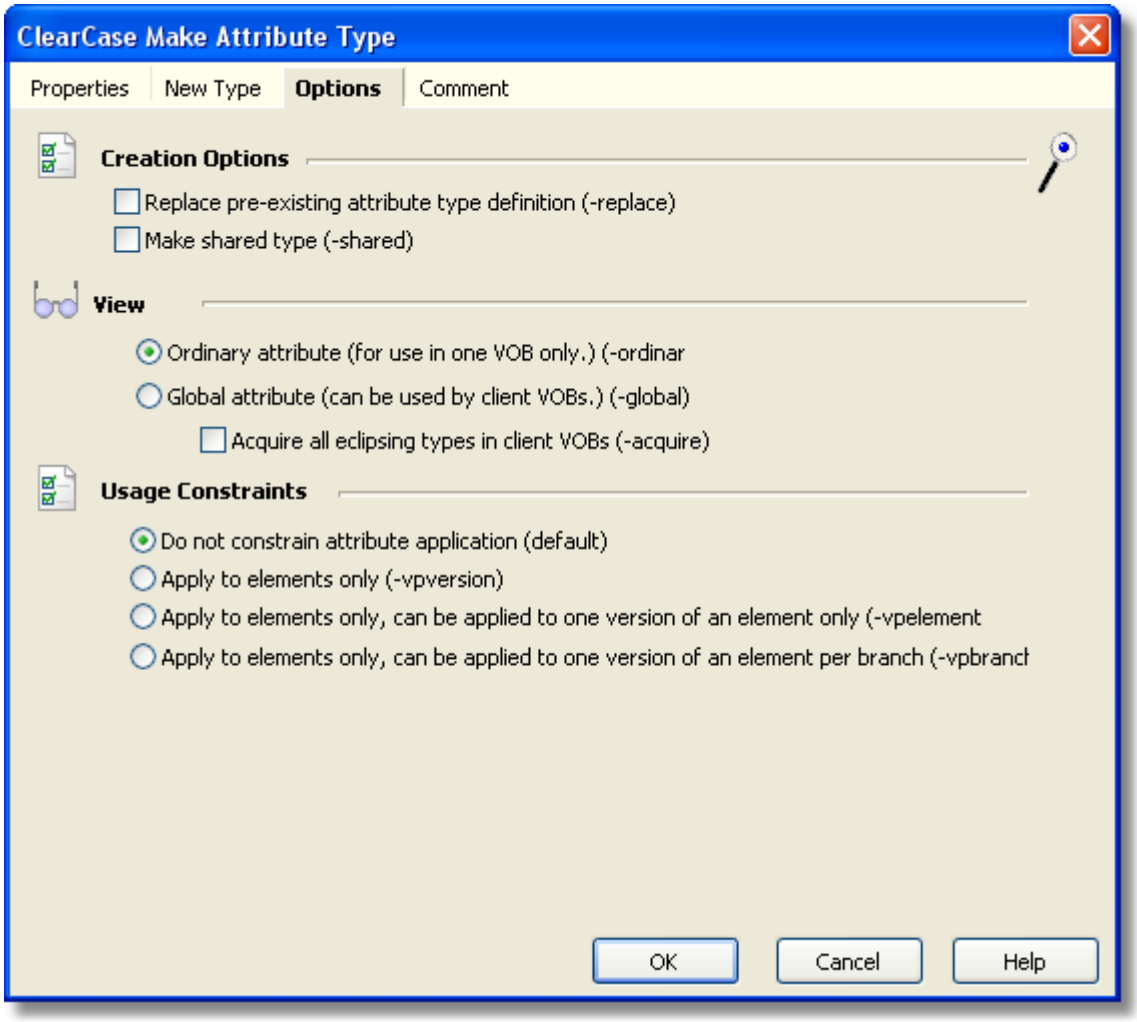
Click on the dropdown menu to see a list of VOBs.

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.5.1.3.10 ClearCase Apply Attribute Action

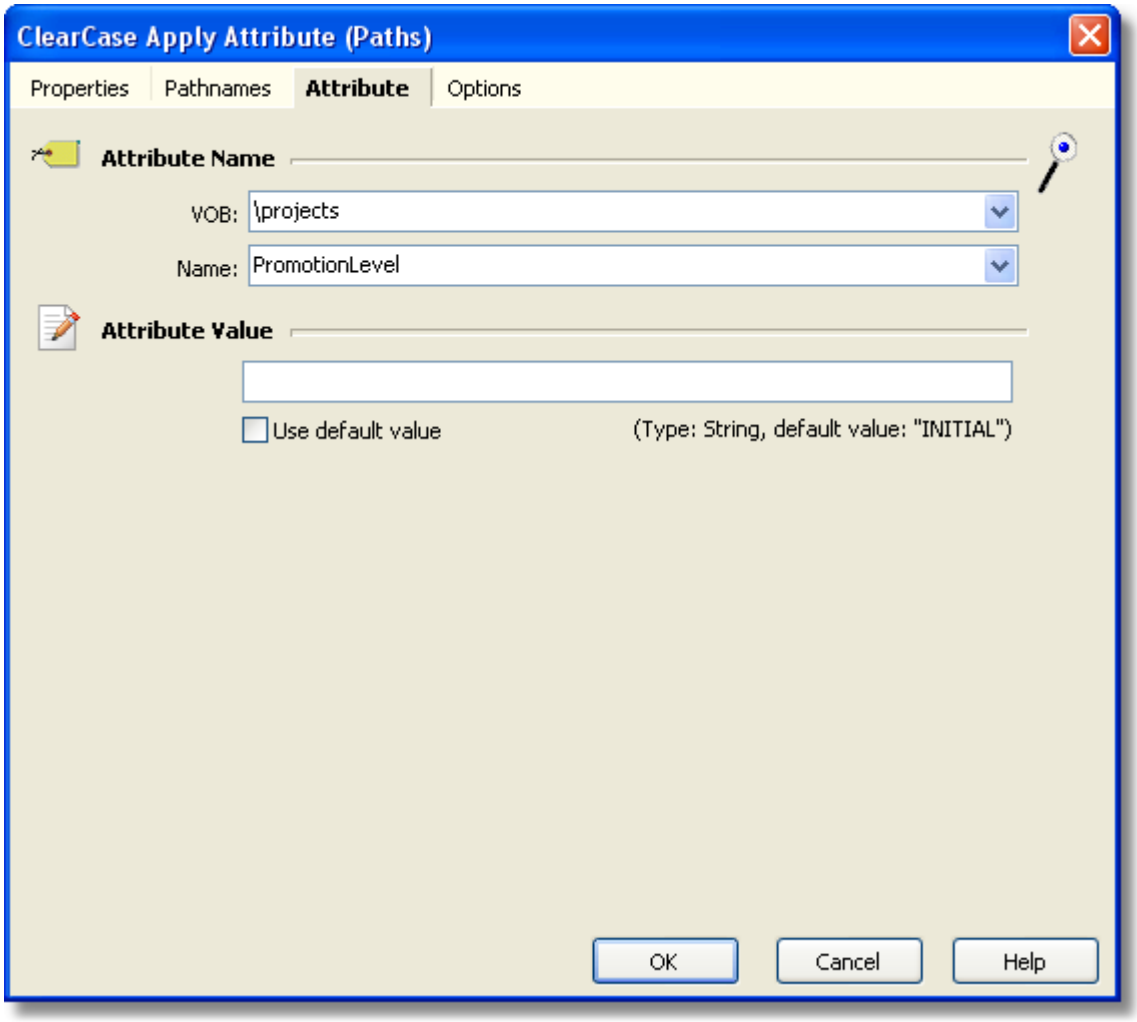
Use this action to apply an attribute (with value) to one or more elements in a VOB.

For full details of the mkattr command, see the ClearCase manual page (type *cleartool man mkattr.*)

The Apply Attribute Action comes in to forms: Apply Attribute (Paths) and Apply

© 2001-2006 VSoft Technologies Pty Ltd

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.)

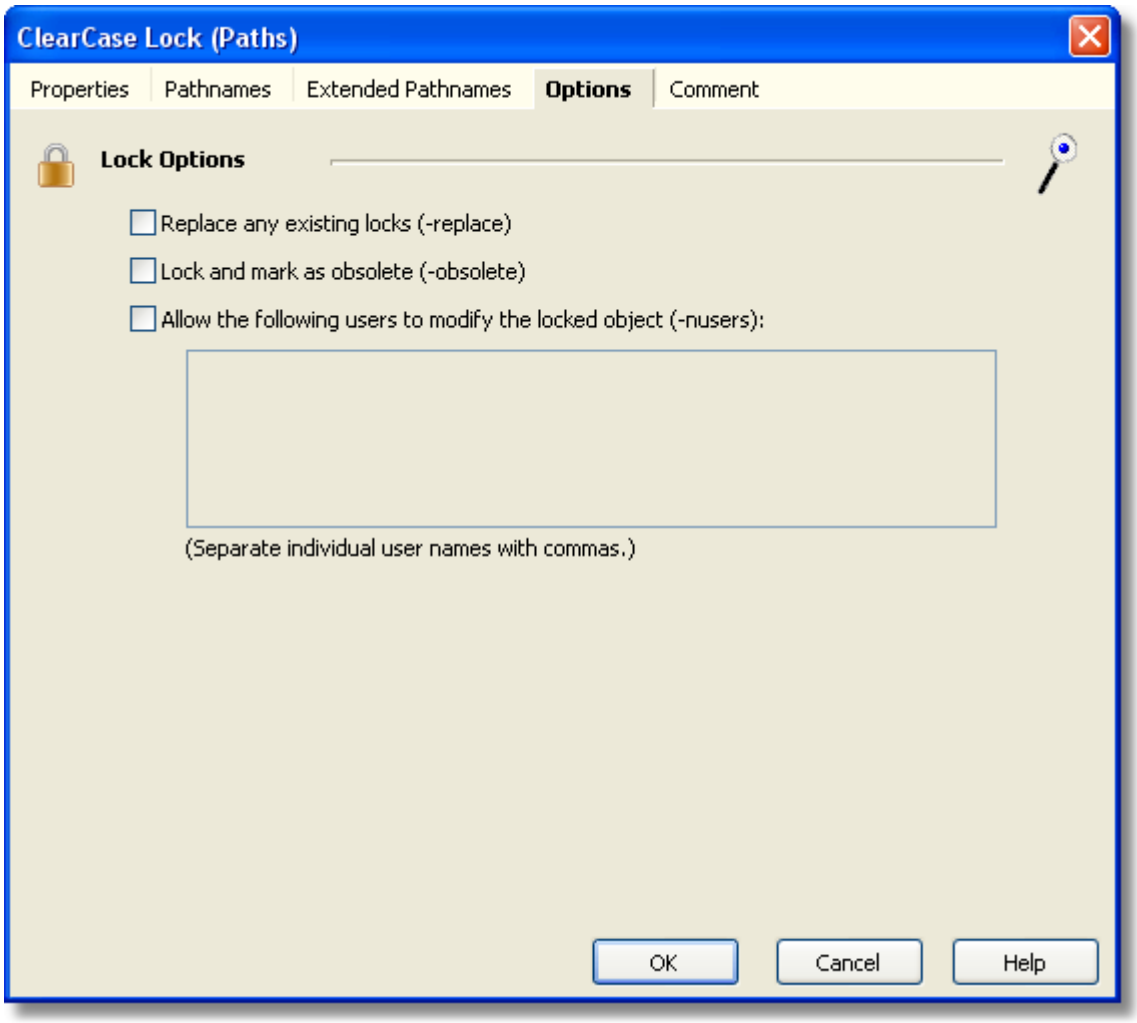
5.5.1.3.11 ClearCase Lock Action

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 to 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.5.1.3.12 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 to 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.

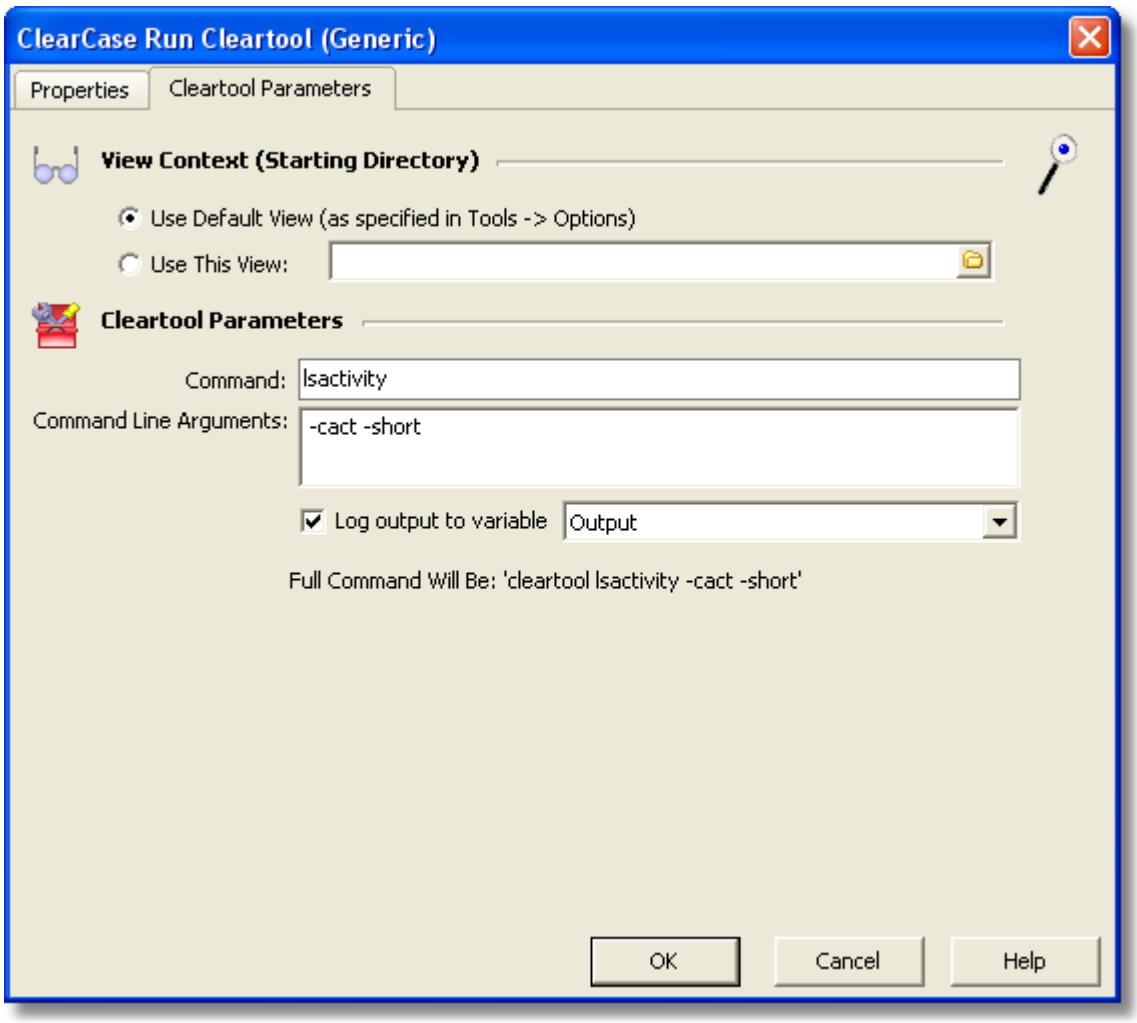
© 2001-2006 VSoft Technologies Pty Ltd

There are no specific options for this action.

5.5.1.3.13 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

© 2001-2006 VSoft Technologies Pty Ltd

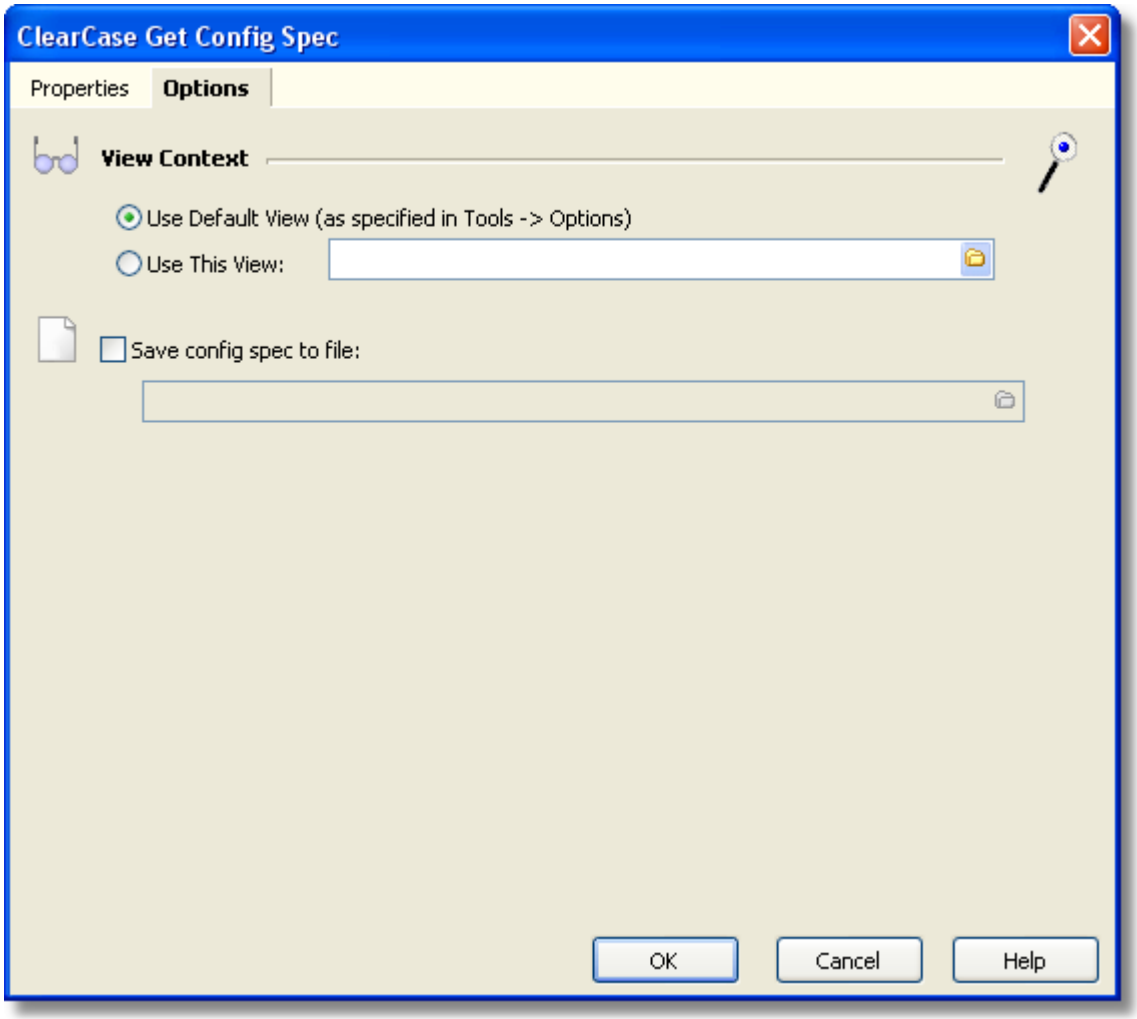
Log output to variable works in a similar way to the 'Log output to Variable' on the Properties tab, except the Cleartool-specific "Log output" only saves the cleartool output and trims any leading and trailing whitespace. This means that, in the example above, the variable 'Output' will be written with 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!

5.5.1.3.14 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.

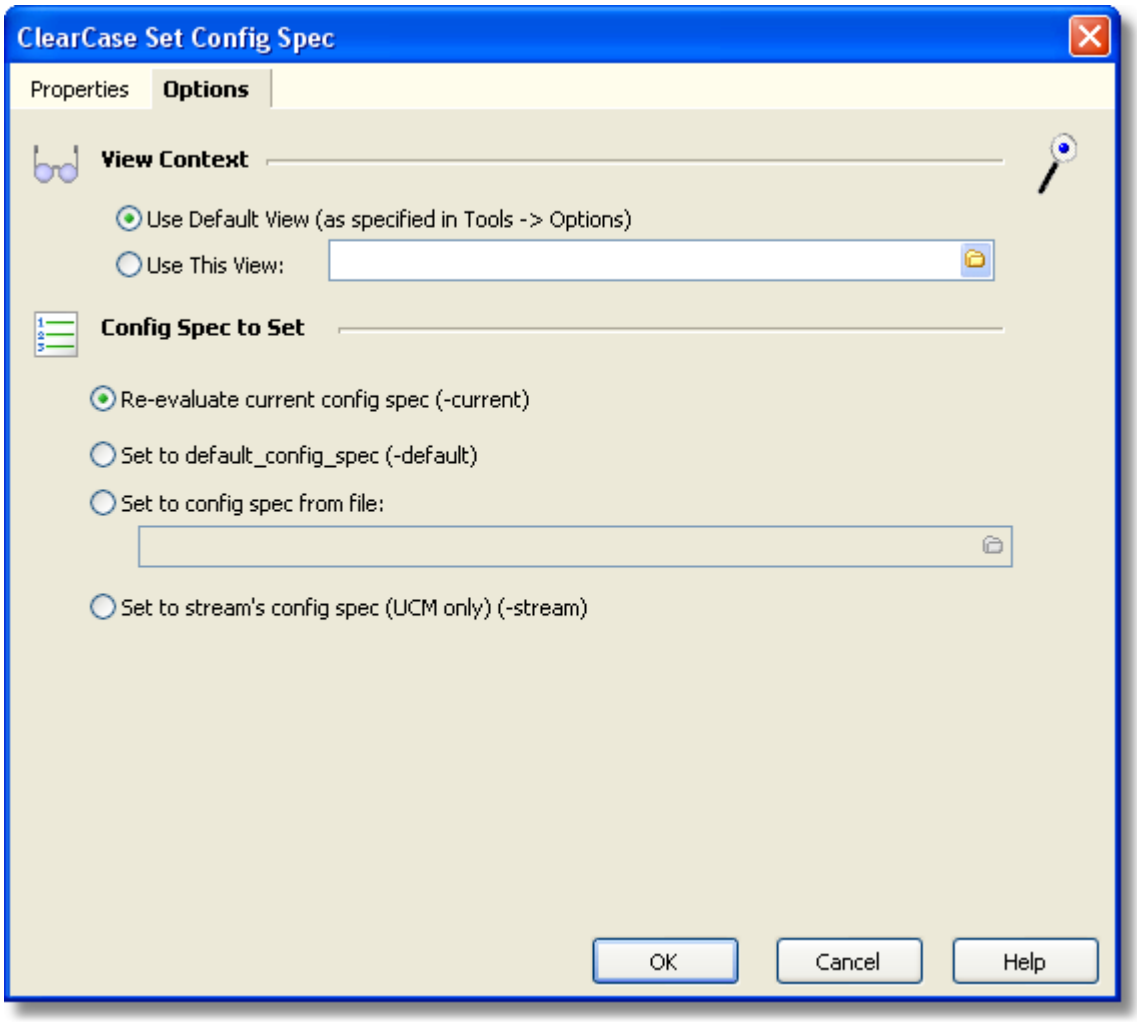
© 2001-2006 VSoft Technologies Pty Ltd

Config specs which have been saved to a file can be set with the Set Config Spec Action.

5.5.1.3.15 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

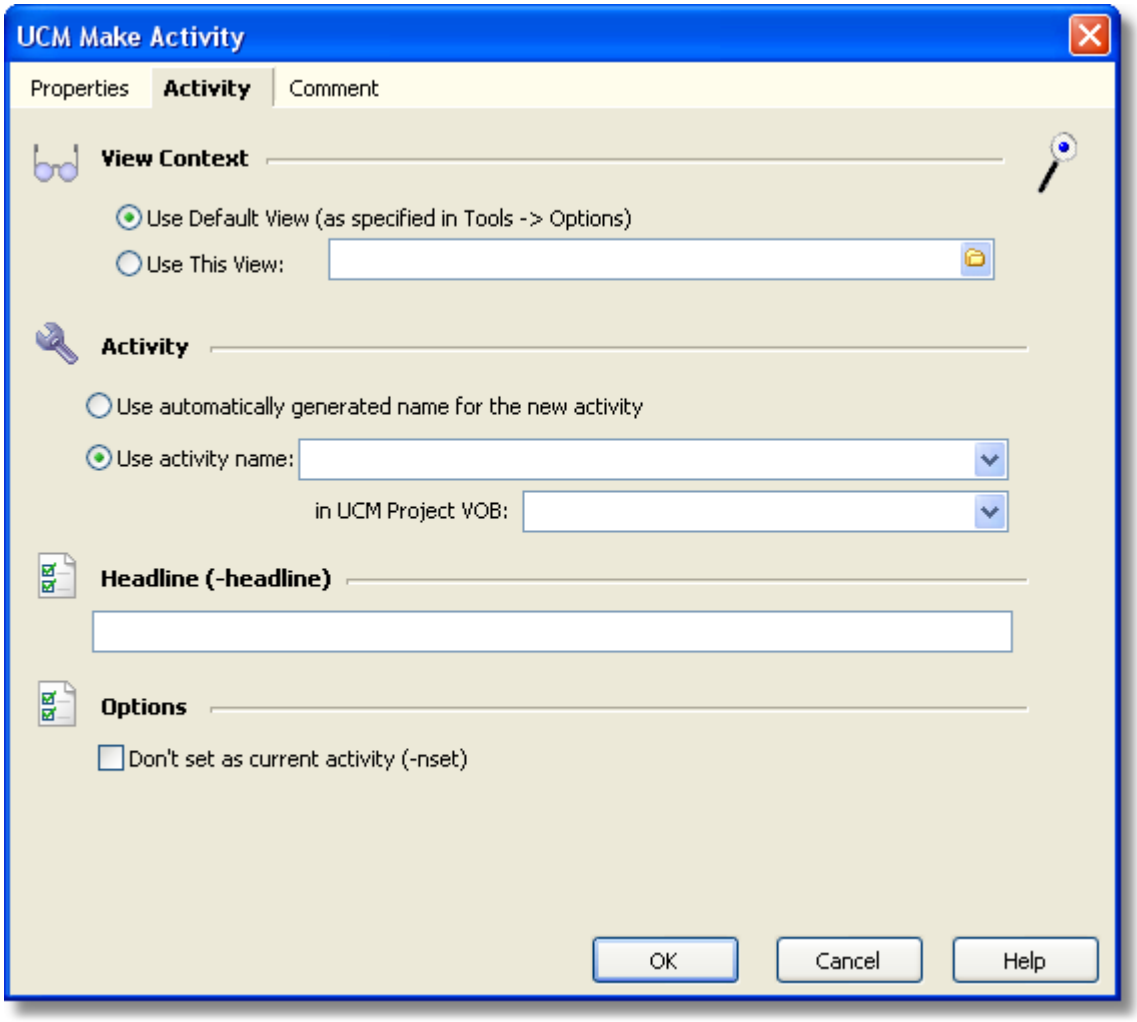
© 2001-2006 VSoft Technologies Pty Ltd

5.5.1.4 UCM

5.5.1.4.1 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.

© 2001-2006 VSoft Technologies Pty Ltd

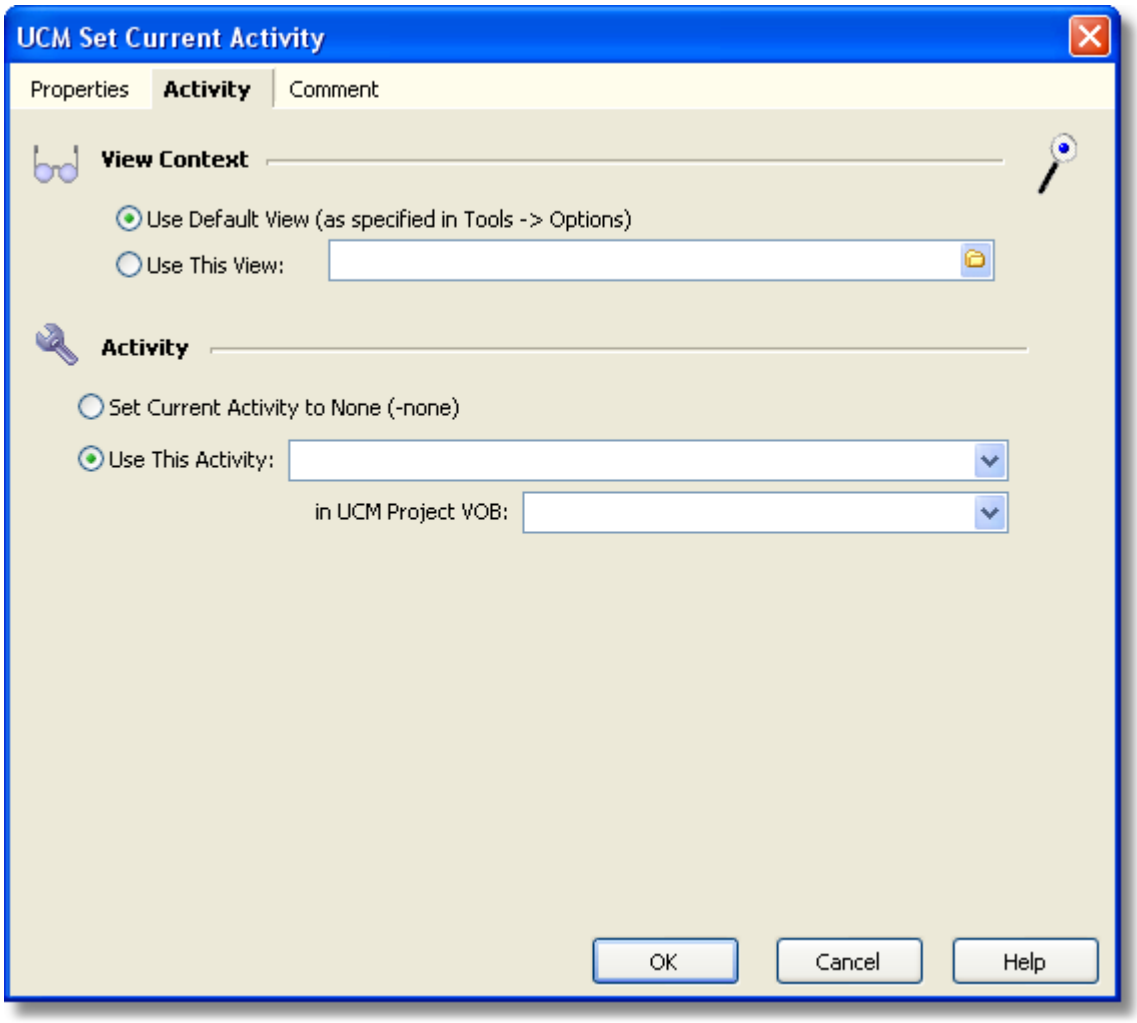
Options

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.

5.5.1.4.2 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

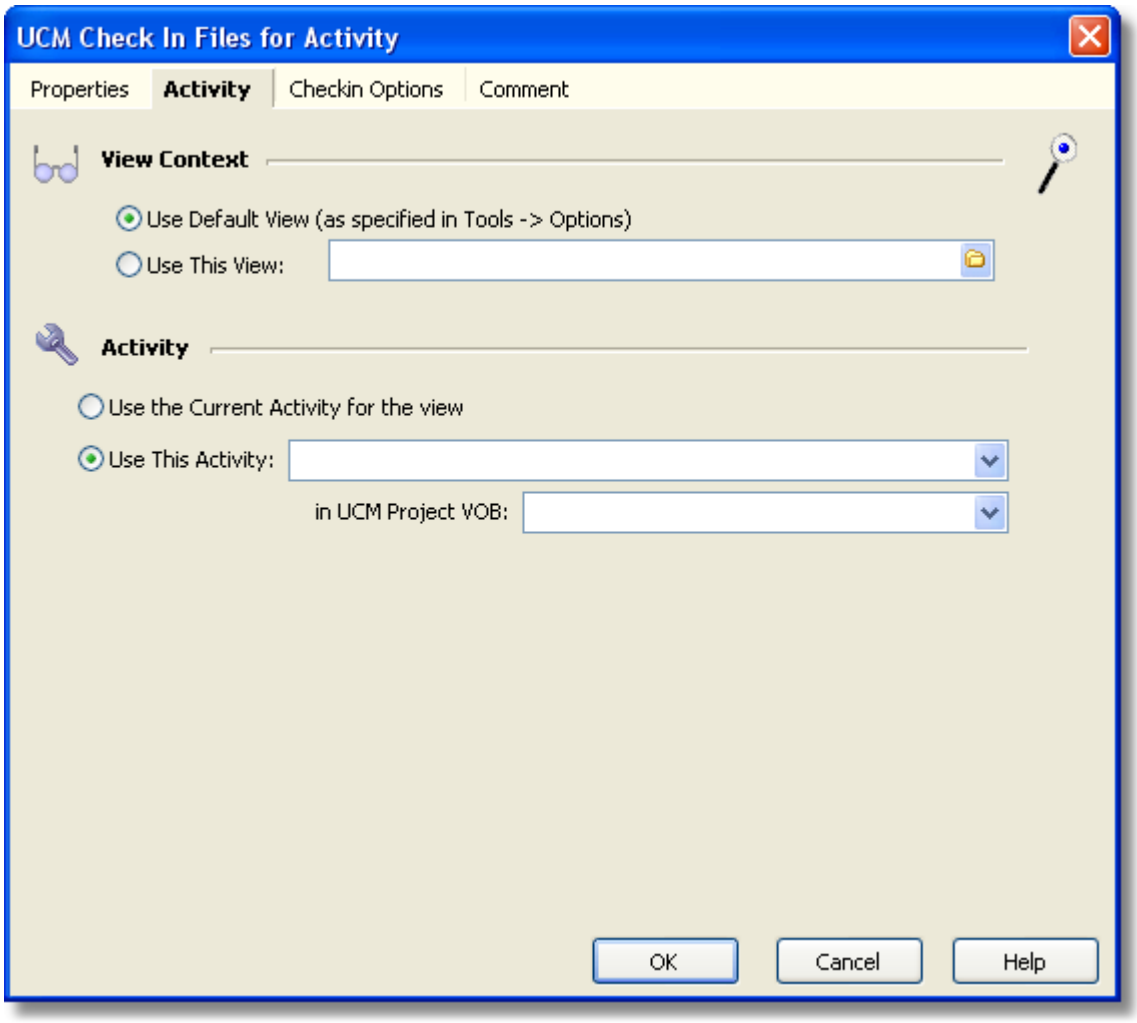
The current activity can be set to 'none', or to any activity from the Stream's UCM project VOB. Click on the VOB dropdown menu to see a list of VOBs, and the Activity dropdown menu to see a list of activities in that VOB.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.1.4.3 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.

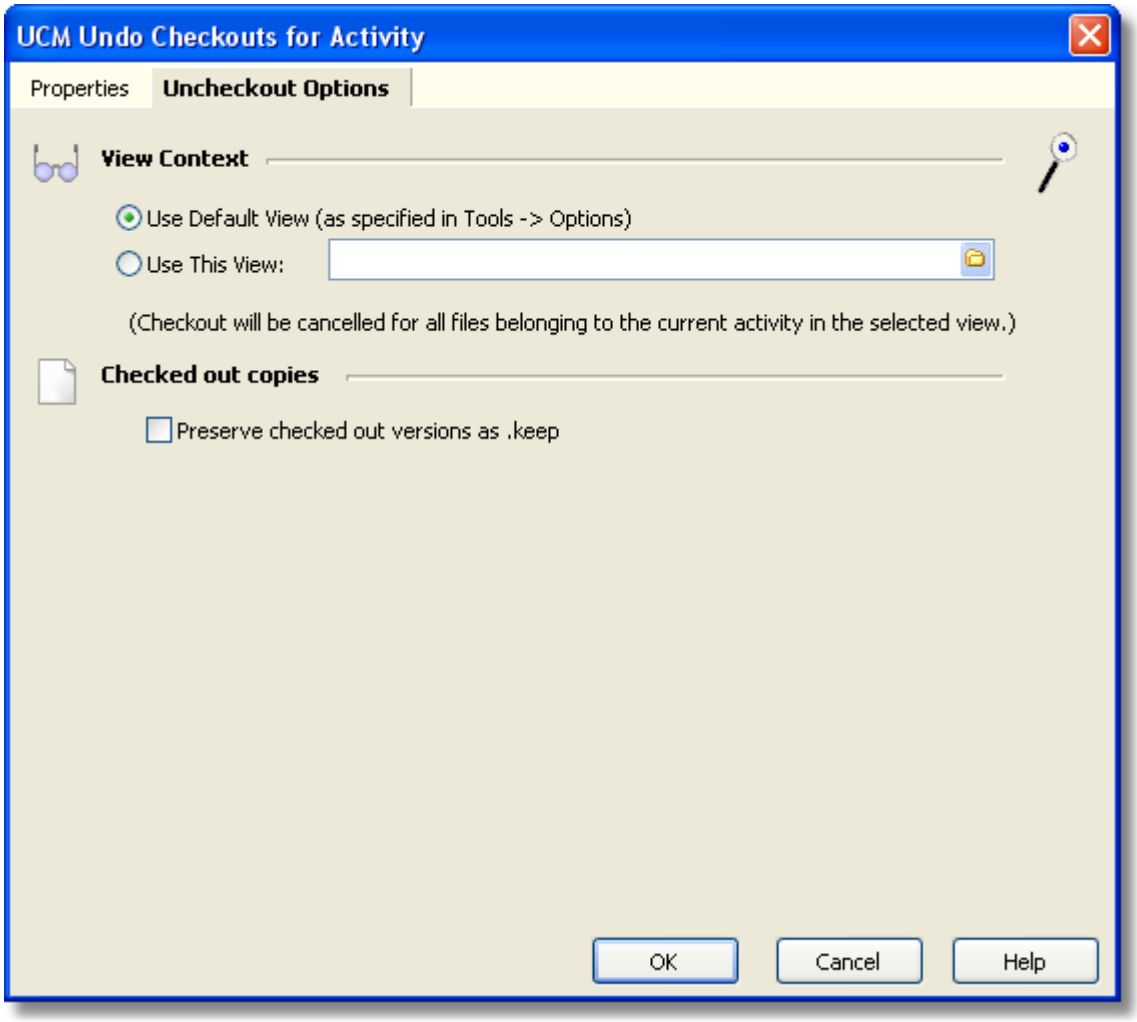
The activity to check in files for can be either the current activity for that view, or a specific activity chosen from a UCM Project VOB.

5.5.1.4.4 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*.)

© 2001-2006 VSoft Technologies Pty Ltd



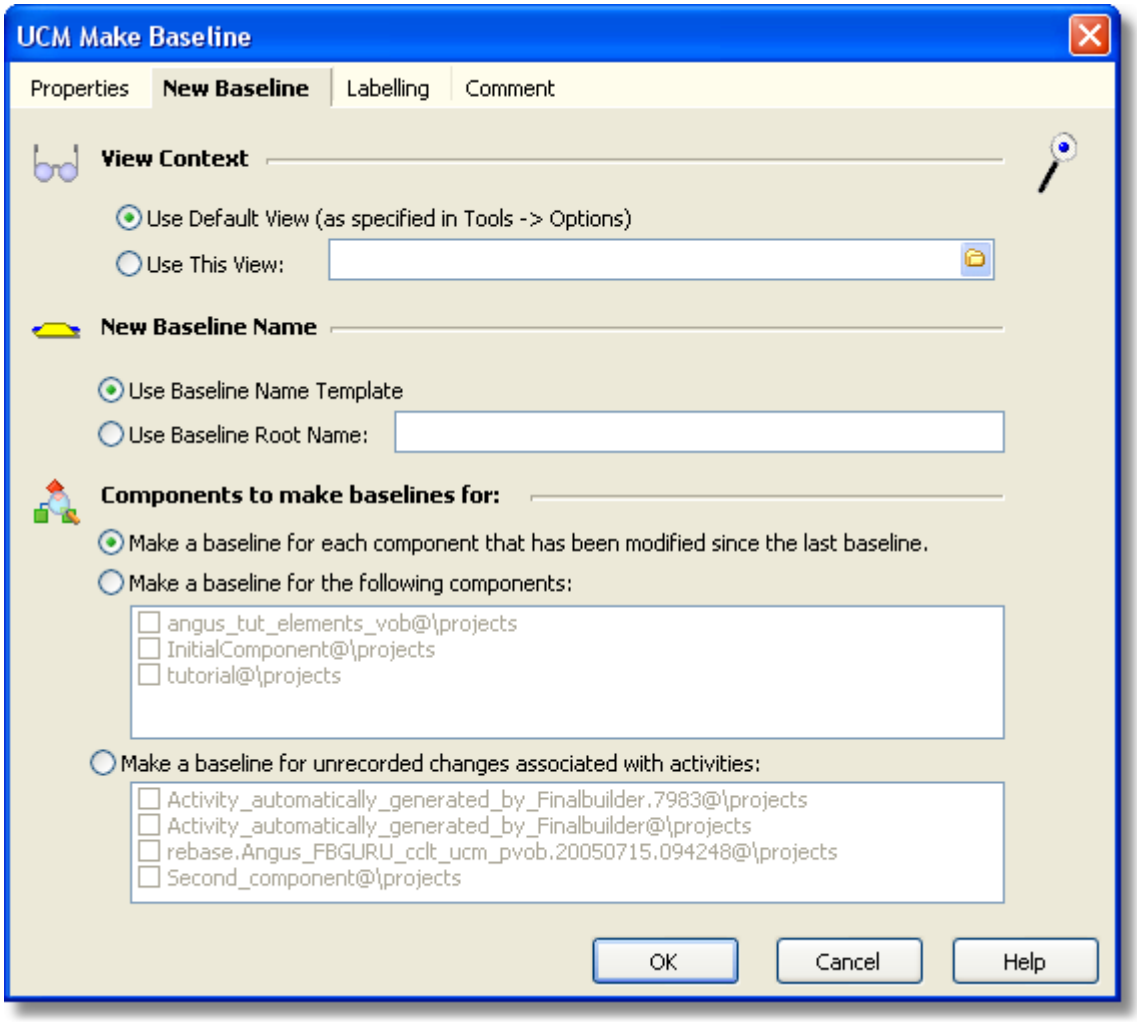
The stream to use can be taken from the default view (specified on the Options panel) or a view specific to that action.

5.5.1.4.5 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*.)

© 2001-2006 VSoft Technologies Pty Ltd



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.

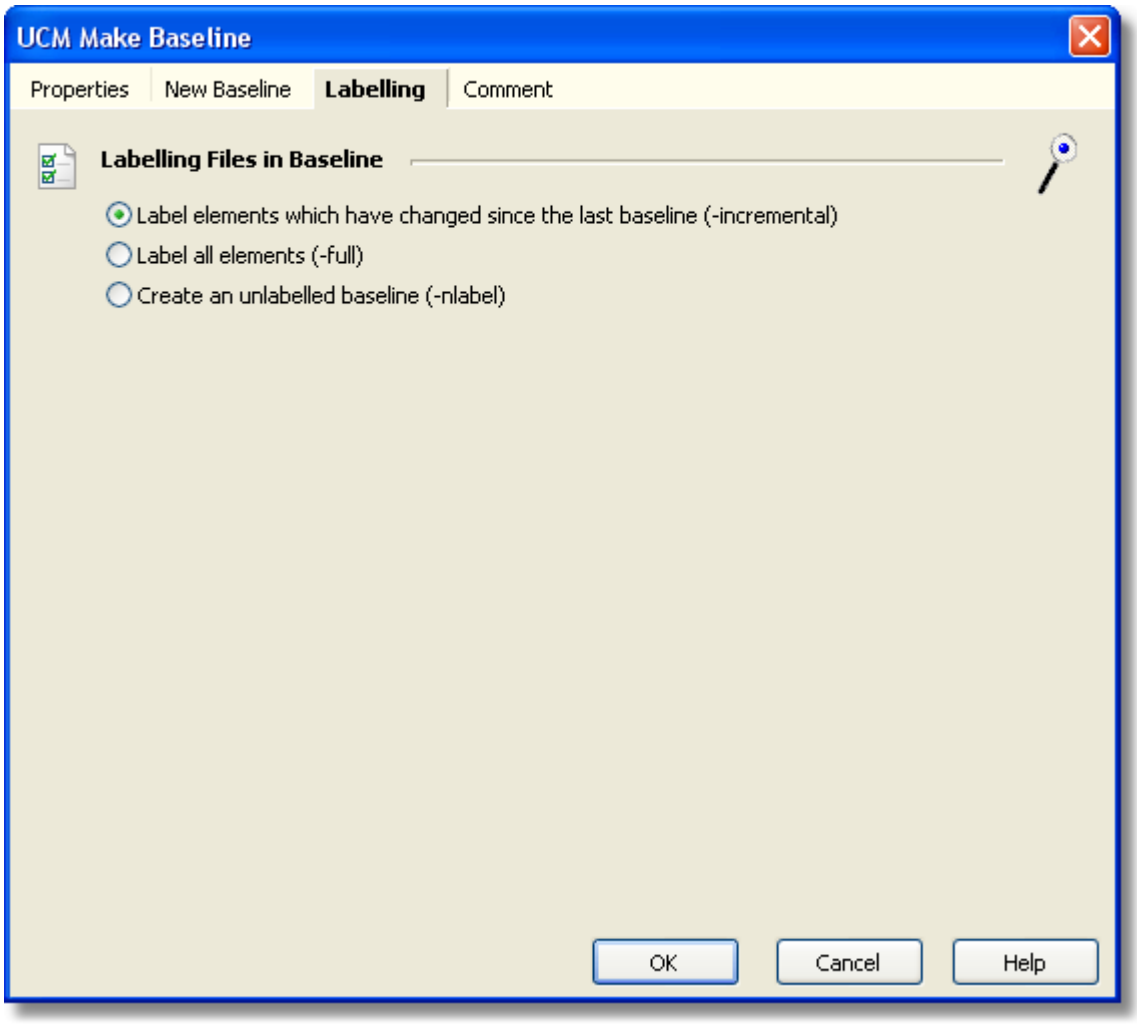
Components to make baselines for

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.

© 2001-2006 VSoft Technologies Pty Ltd



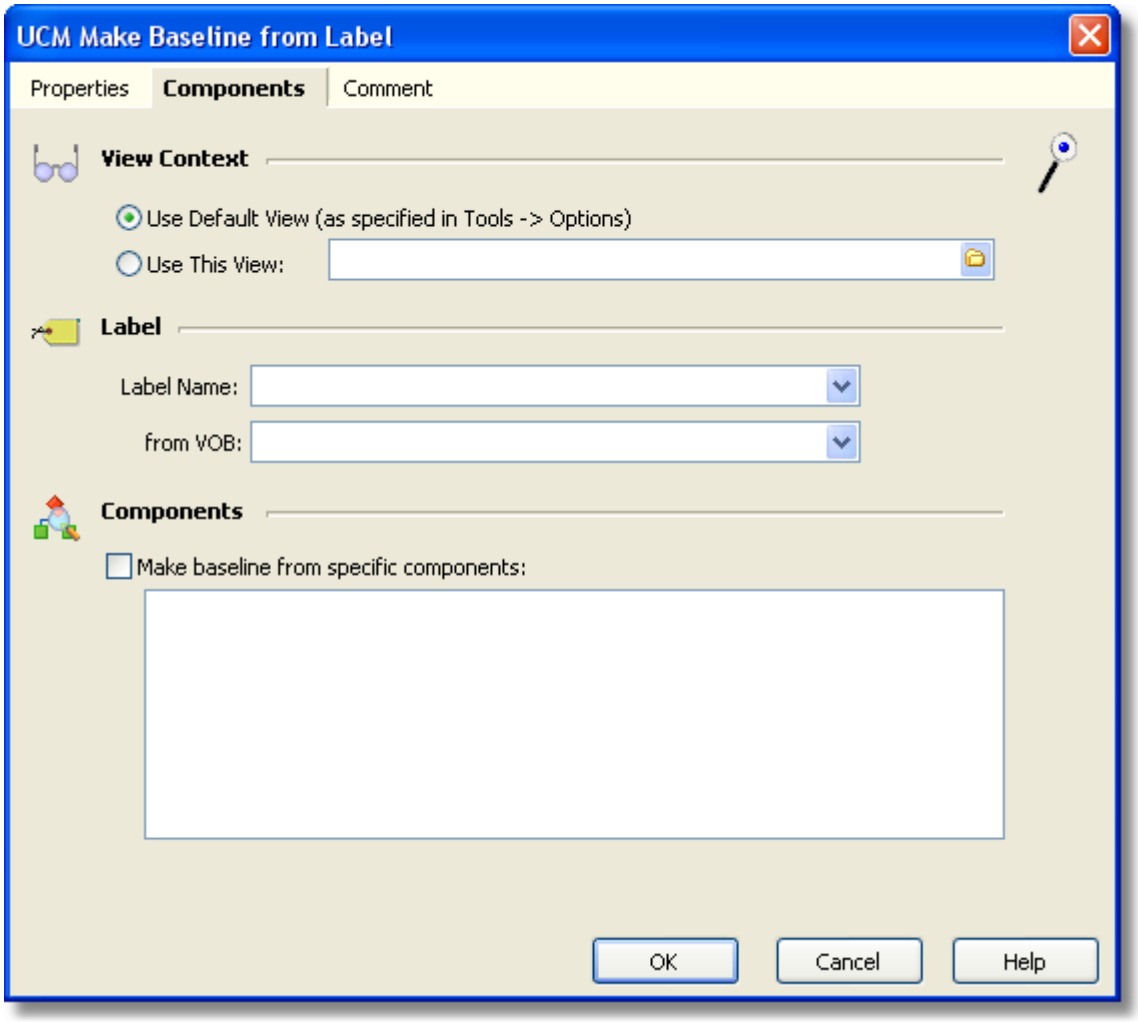
Elements in the baseline can be labelled with the name of the new baseline.

5.5.1.4.6 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*.)

© 2001-2006 VSoft Technologies Pty Ltd



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

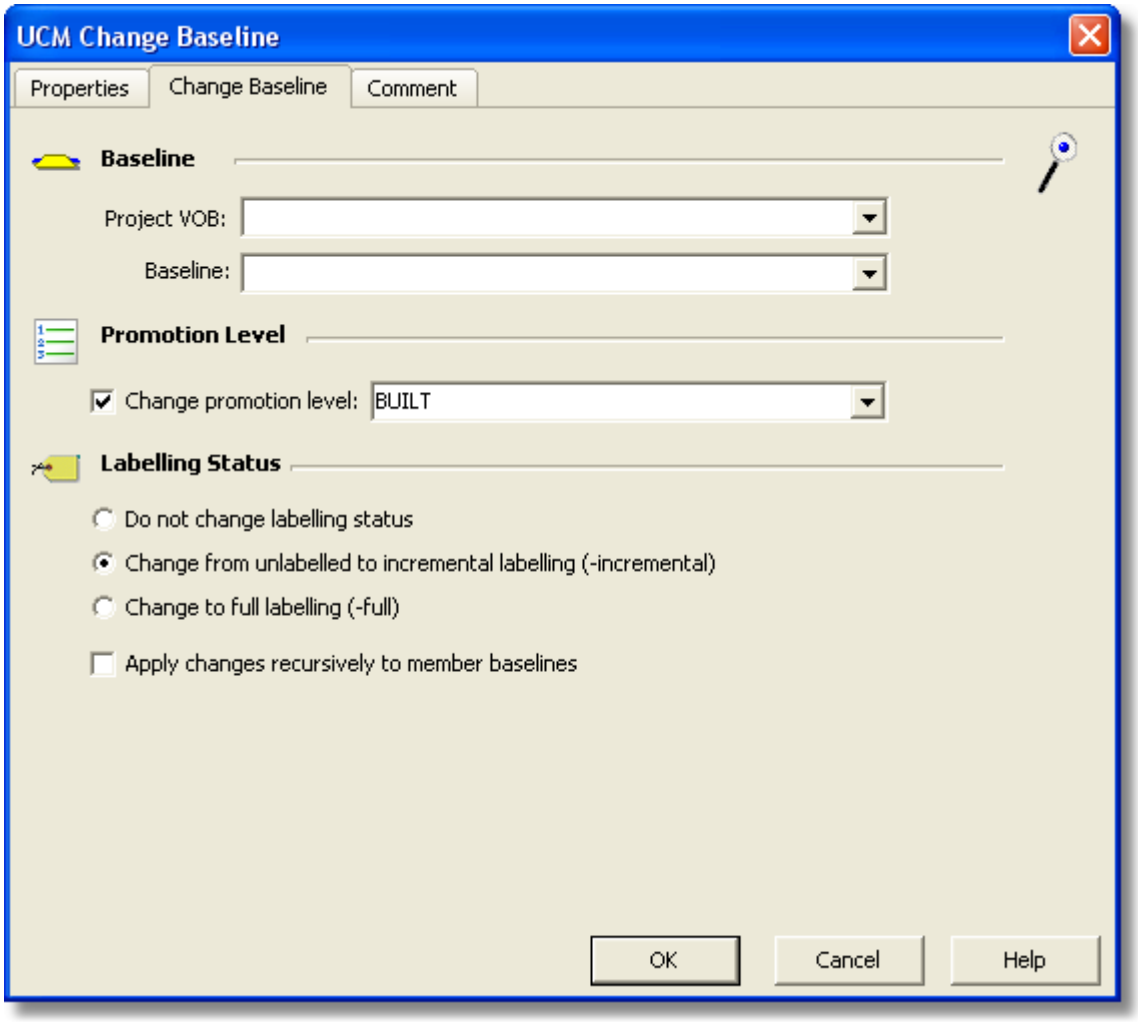
If a list of components is specified, the new baseline will only contain labelled elements from those components.

5.5.1.4.7 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*.)

© 2001-2006 VSoft Technologies Pty Ltd



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...

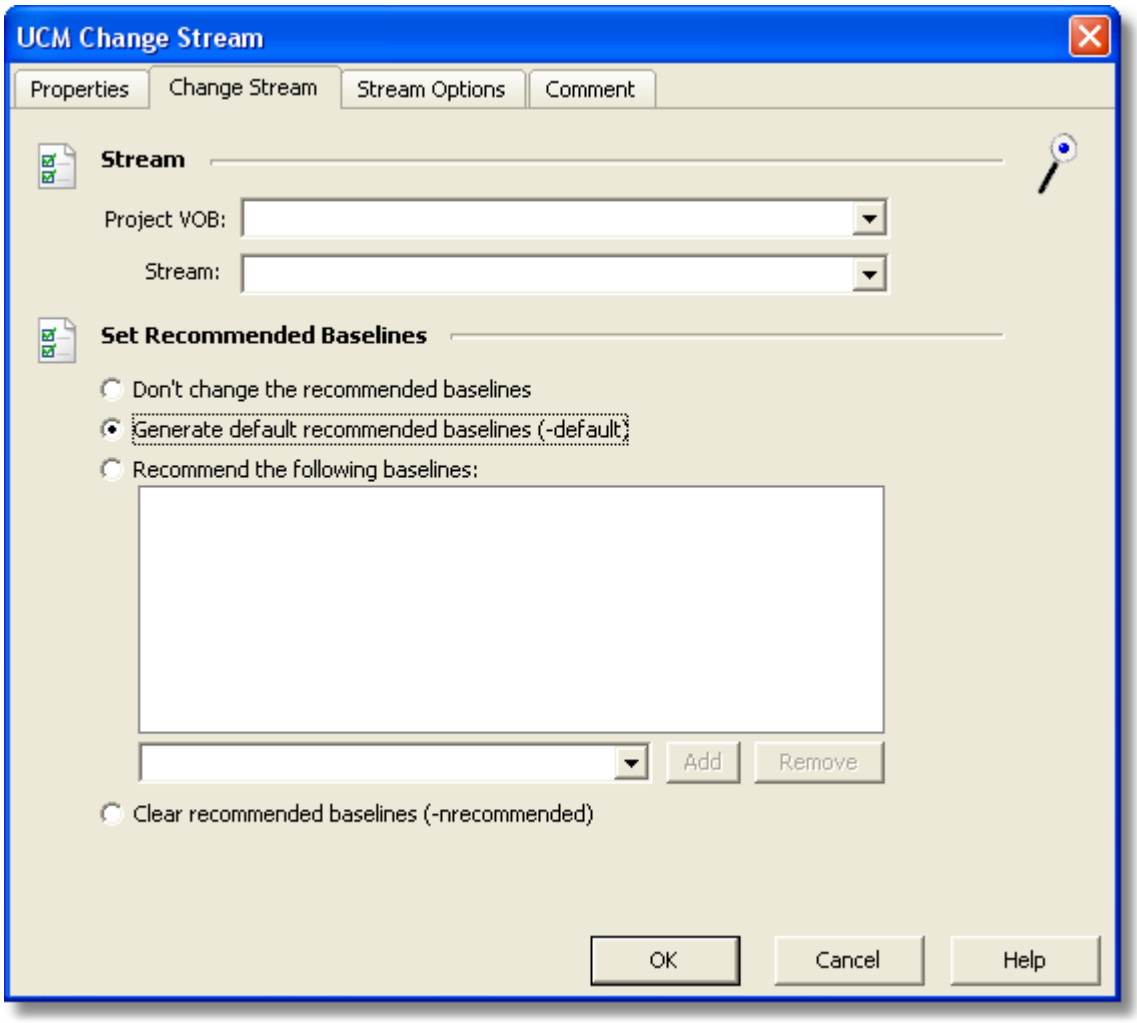
If this option is checked, the labelling status of member baselines will be updated as well.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.1.4.8 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

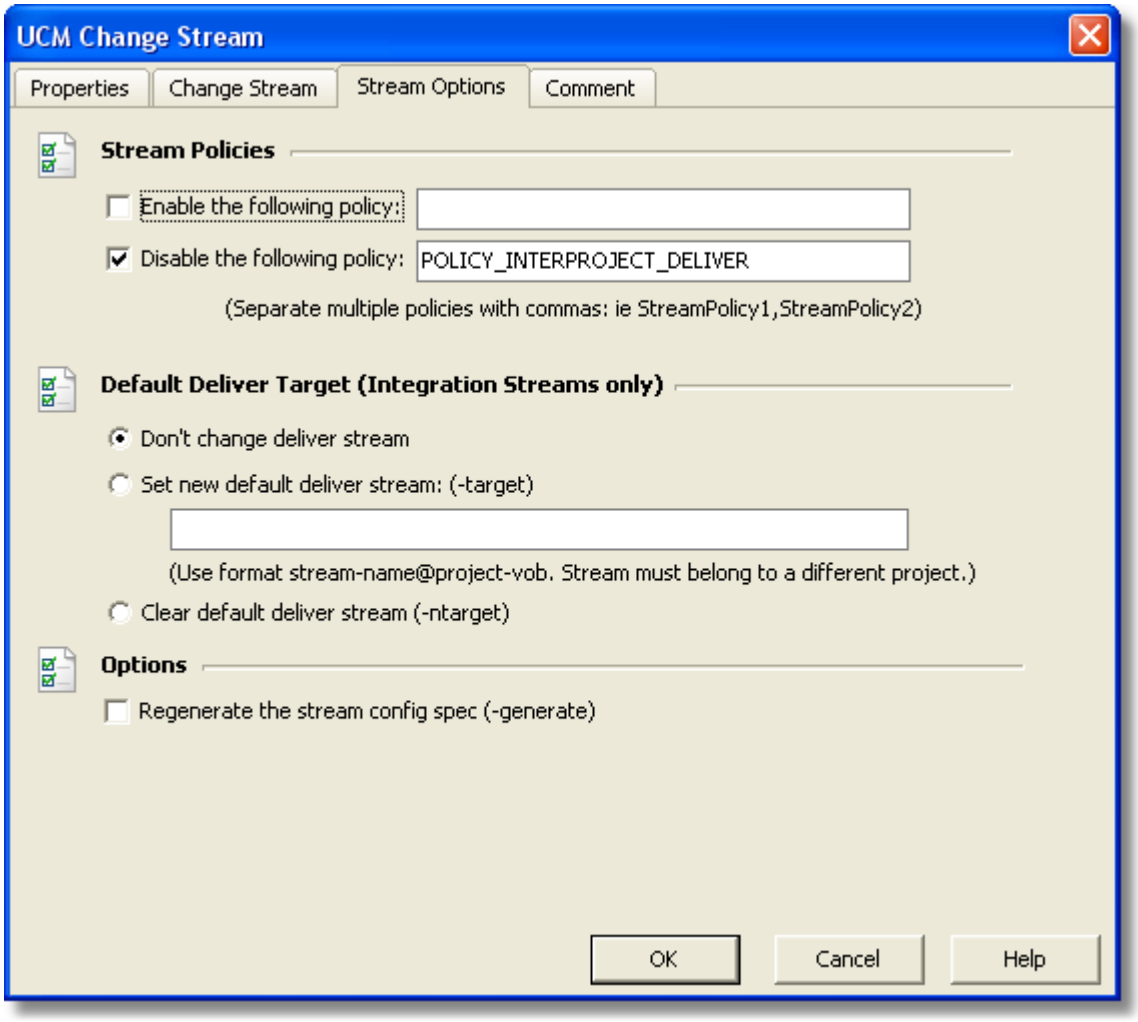
Specify a full list of recommended baselines for the stream, in the form *baseline@vob-selector*. Click on the dropdown menu to see a list of baselines created in the specified stream.

© 2001-2006 VSoft Technologies Pty Ltd

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.



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.

© 2001-2006 VSoft Technologies Pty Ltd

Regenerate the stream config spec

If this option is checked, it forces the stream to regenerate and reevaluate its config spec.

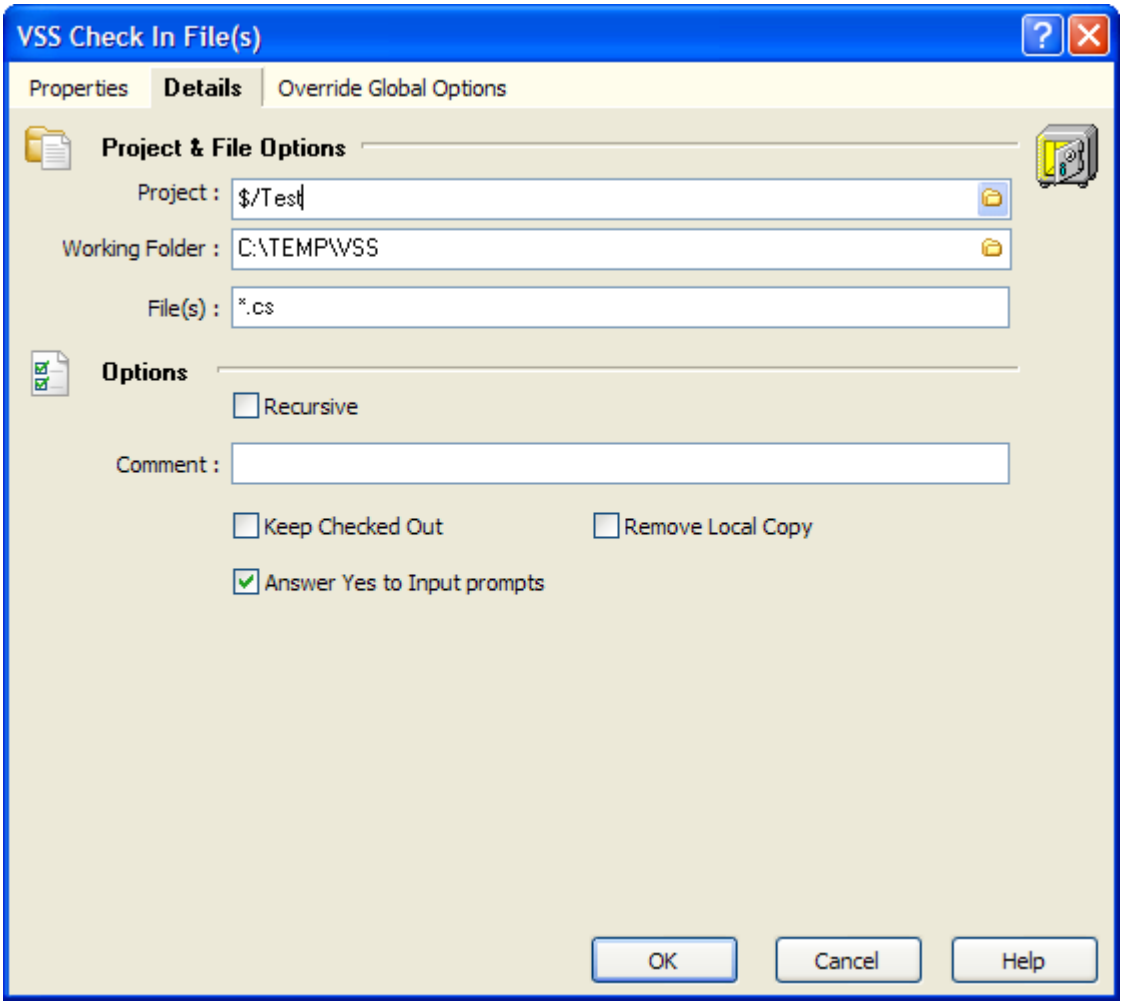
© 2001-2006 VSoft Technologies Pty Ltd

5.5.2 Source Safe

5.5.2.1 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

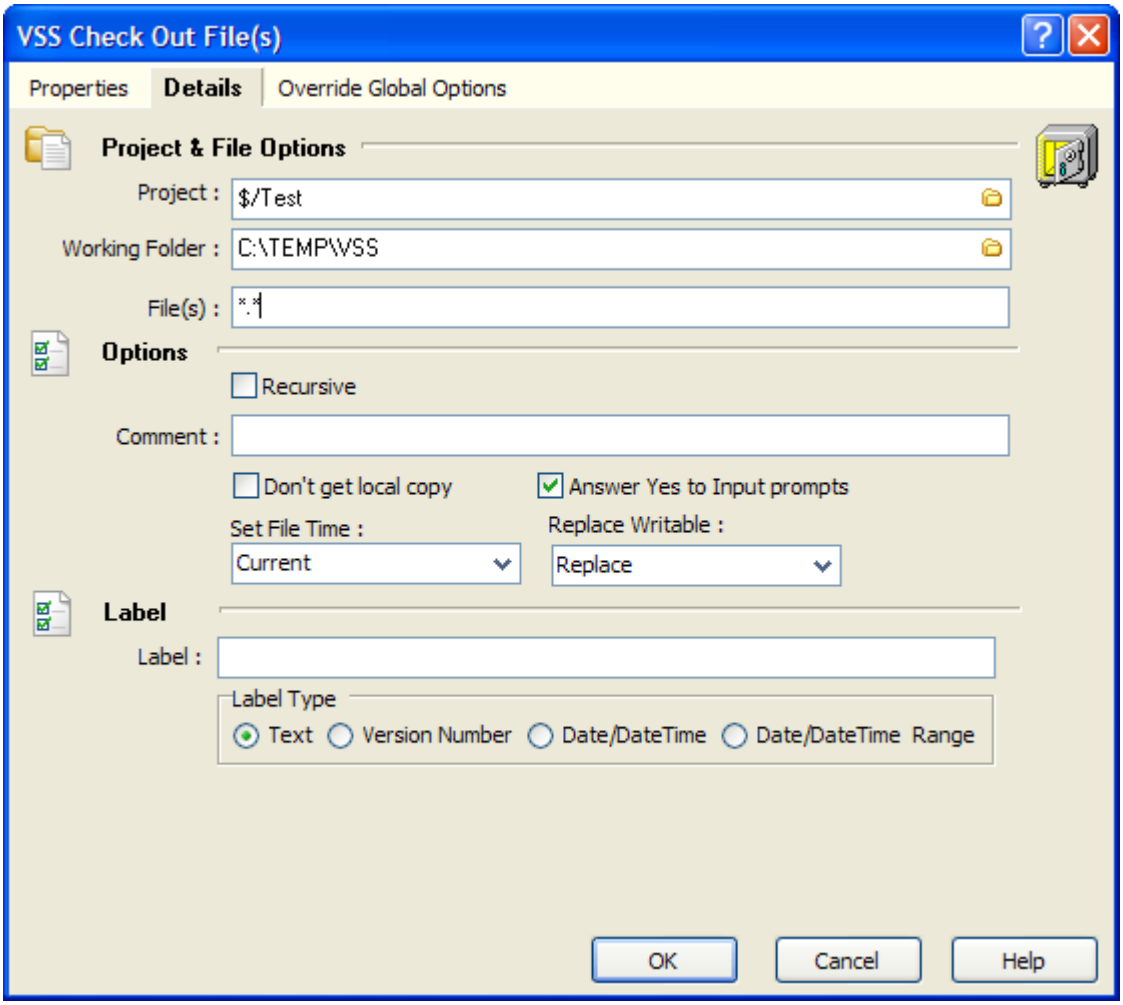
© 2001-2006 VSoft Technologies Pty Ltd

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

5.5.2.2 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/subpoject
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,

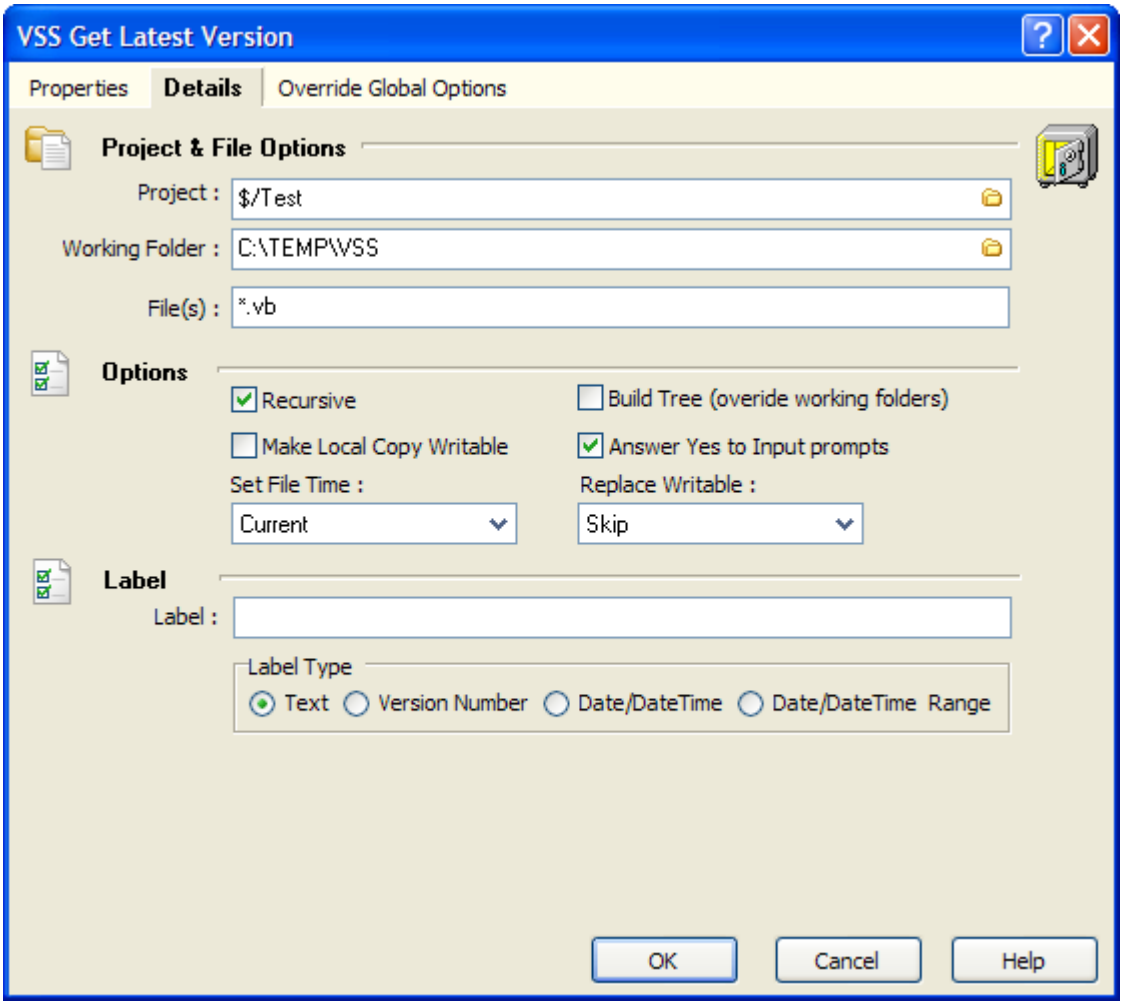
© 2001-2006 VSoft Technologies Pty Ltd

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

5.5.2.3 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.

© 2001-2006 VSoft Technologies Pty Ltd

Scripting Info

The Action properties available are :

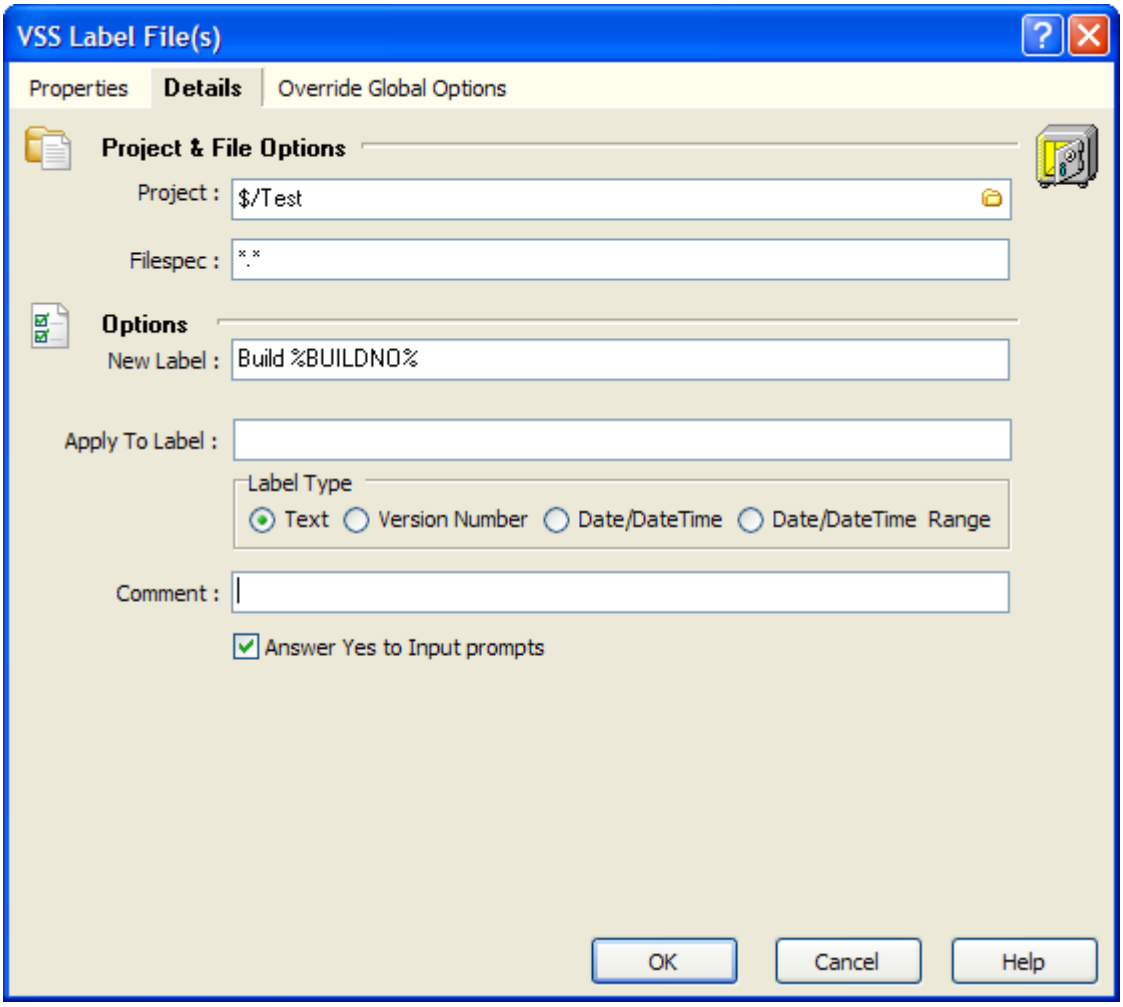
property SSProject : WideString;// The name of the source safe project, eg. $/test/subpoject
property WorkingFolder : WideString;// The folder where the local copy will be placed.
property FileSpec : WideString;// The file specification for files to check out , you may use wildcards in this
property 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

5.5.2.4 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.

© 2001-2006 VSoft Technologies Pty Ltd



Scripting Info

The Action properties available are :

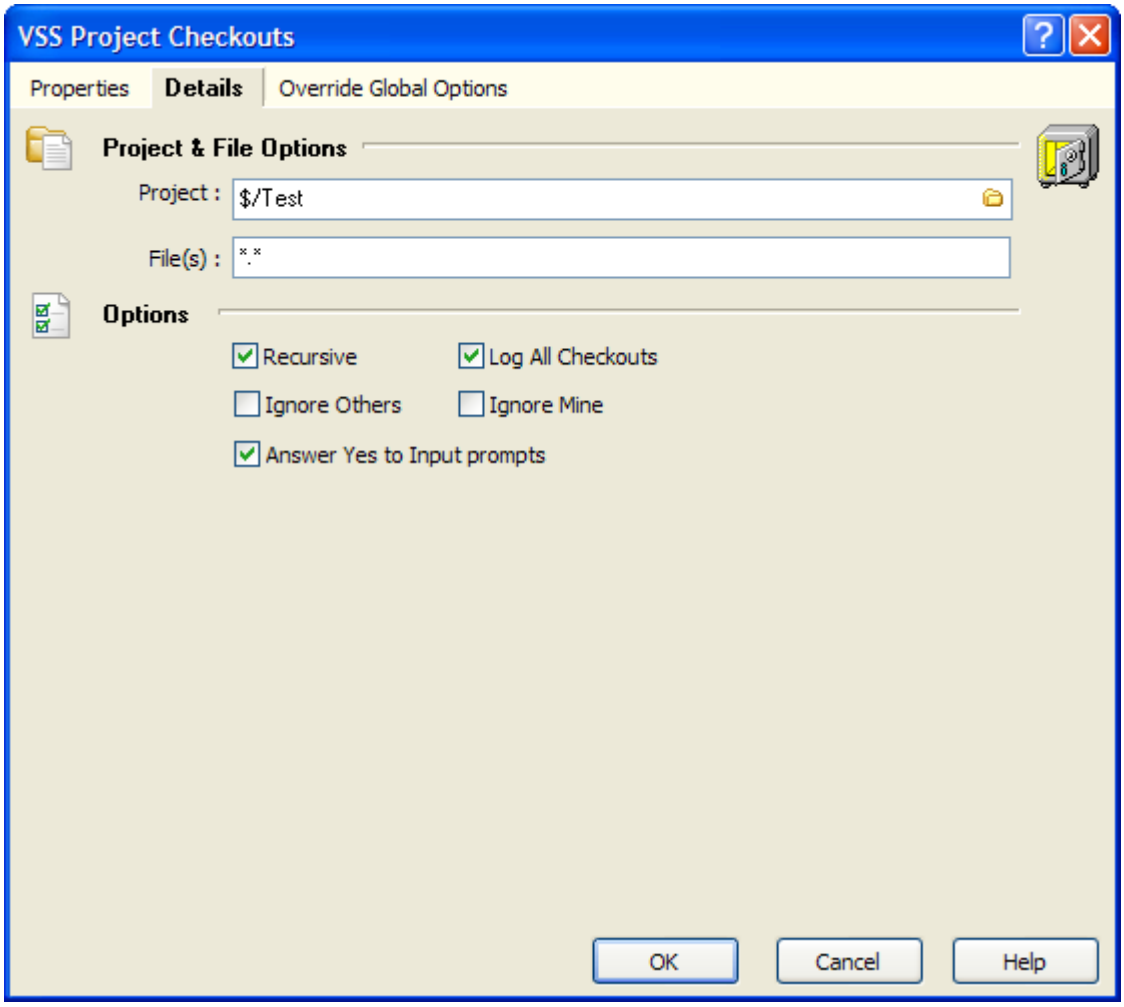
property SSProject : WideString;// The name of the source safe project, eg. $/test/subproject
property SSLabel : WideString;// The Label to apply
property FileSpec : WideString;// The file specification for files to label , you may use wildcards in this after check in.
property Comment; //Add a comment when Labelling a file or project.

5.5.2.5 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.

© 2001-2006 VSoft Technologies Pty Ltd

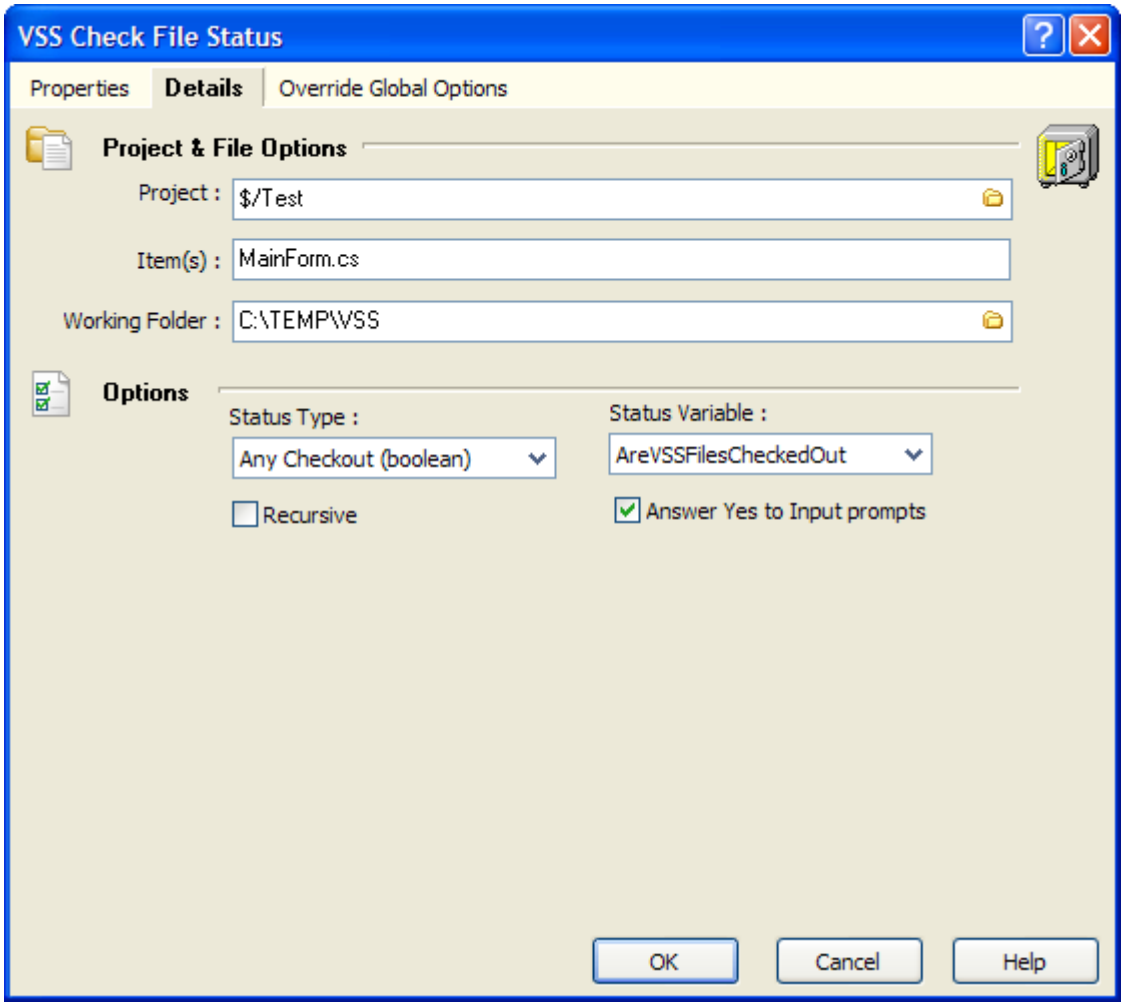


5.5.2.6 Source Safe Check File Status

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.

© 2001-2006 VSoft Technologies Pty Ltd

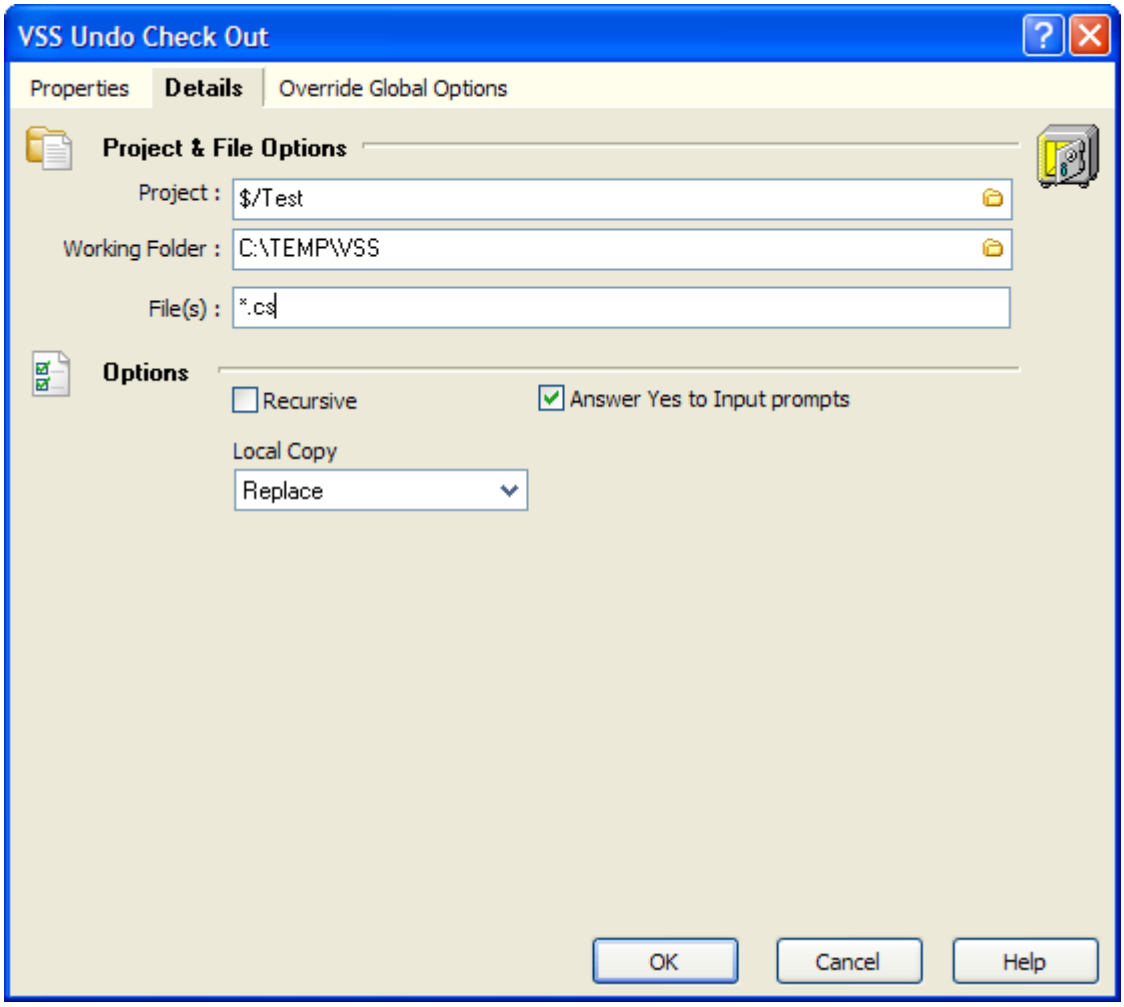


5.5.2.7 Source Safe Undo CheckOut Action

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.

© 2001-2006 VSoft Technologies Pty Ltd

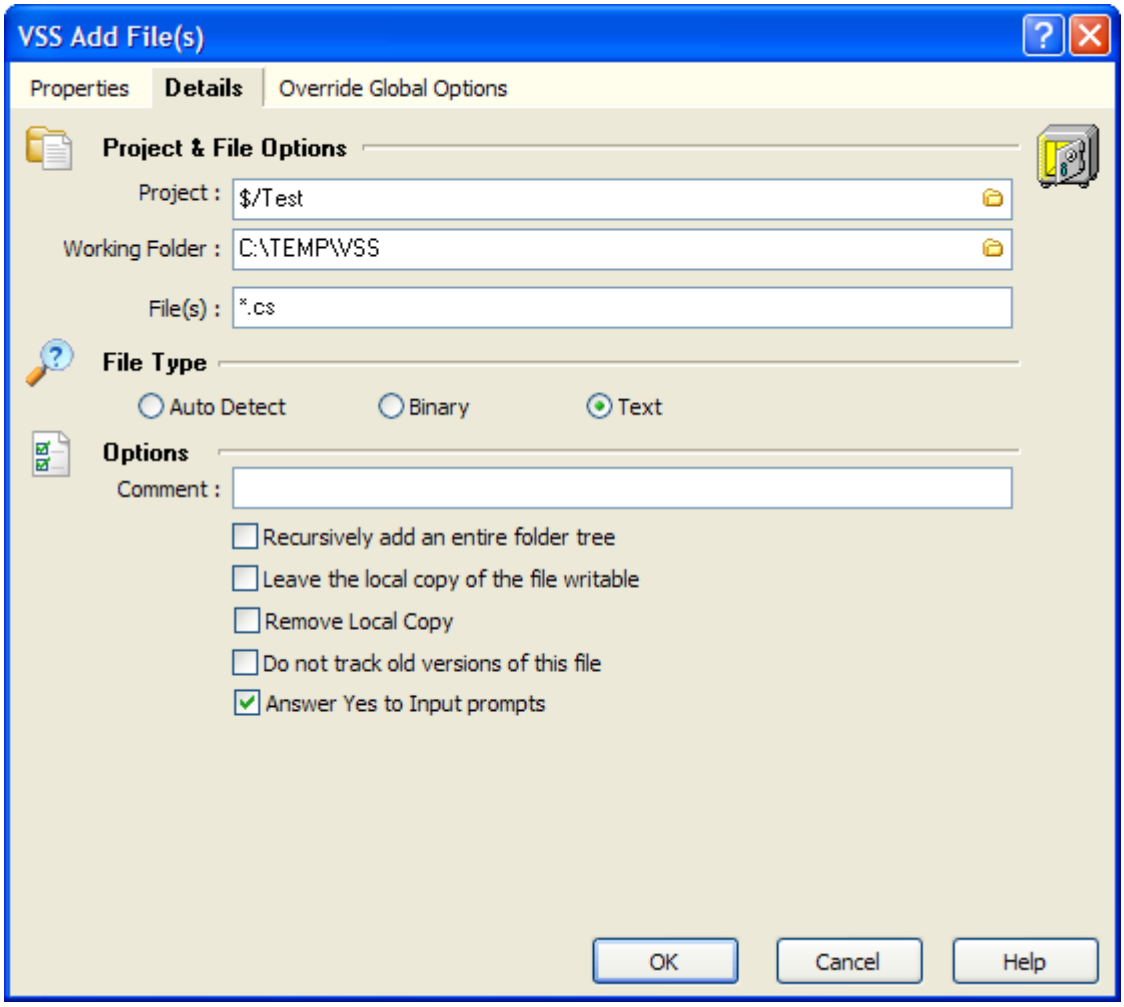


5.5.2.8 Source Safe Add Files Action

Adds Files to a Source Safe Project.

How to override Global Options for the Source Safe actions.

© 2001-2006 VSoft Technologies Pty Ltd



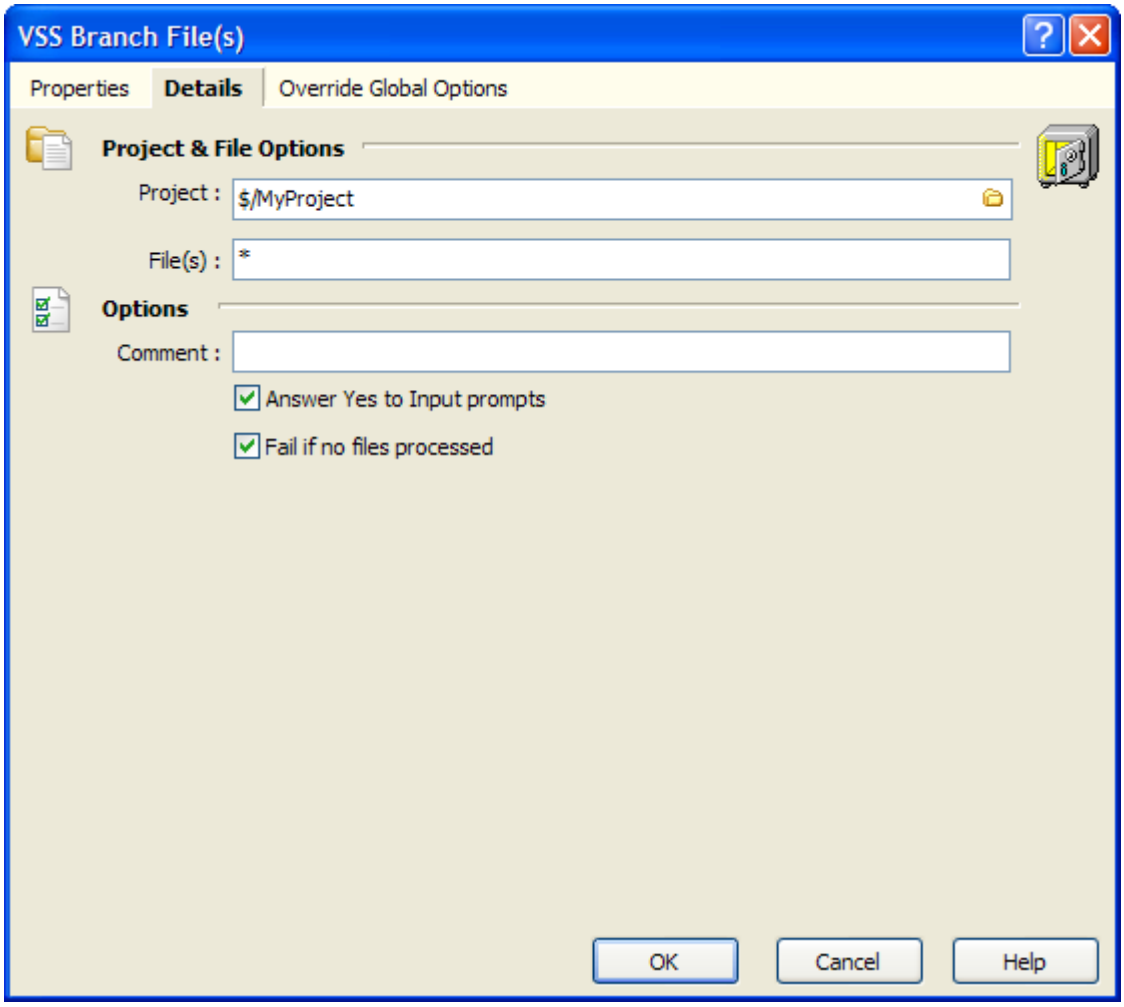
5.5.2.9 Source Safe Branch

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.

© 2001-2006 VSoft Technologies Pty Ltd



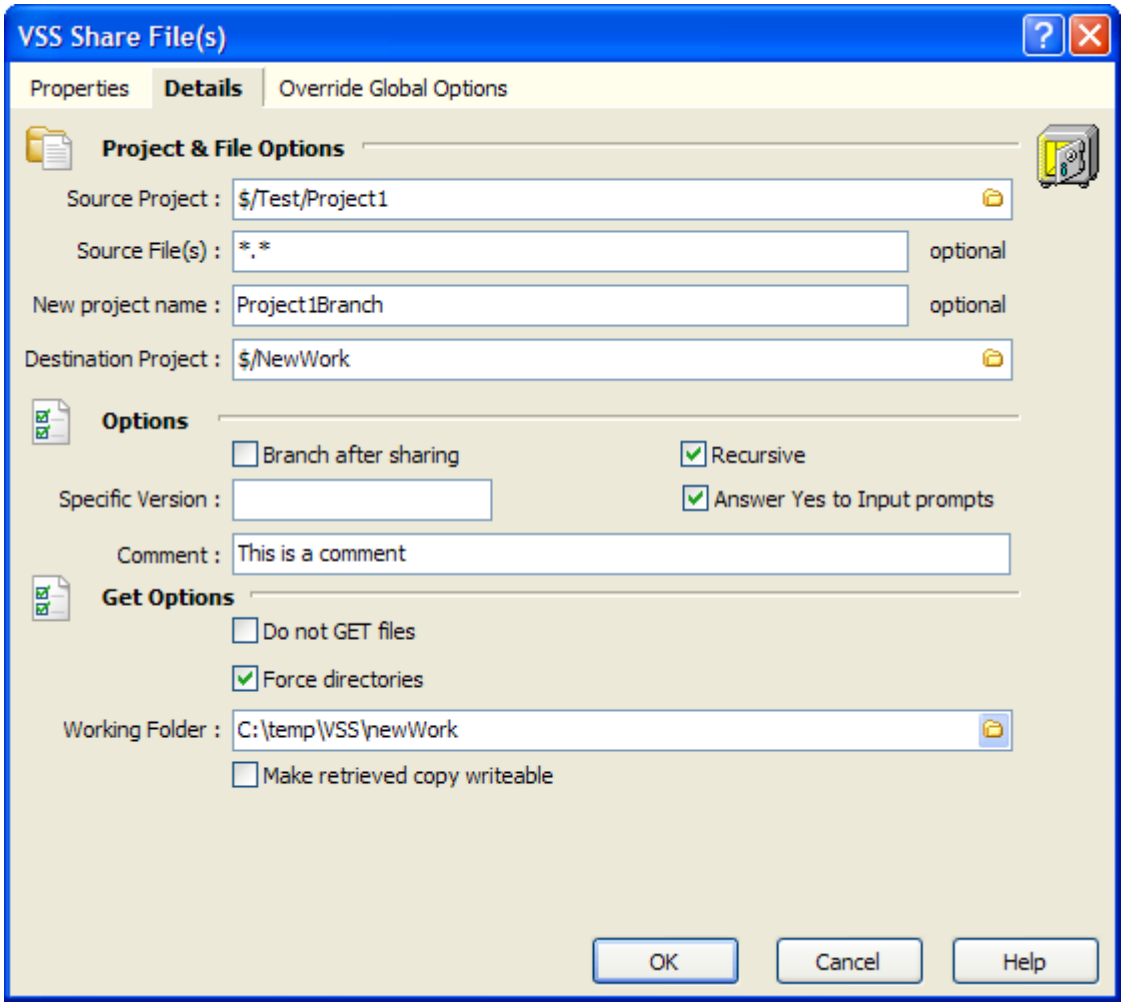
5.5.2.10 Source Safe Share

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.

© 2001-2006 VSoft Technologies Pty Ltd



5.5.2.11 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.

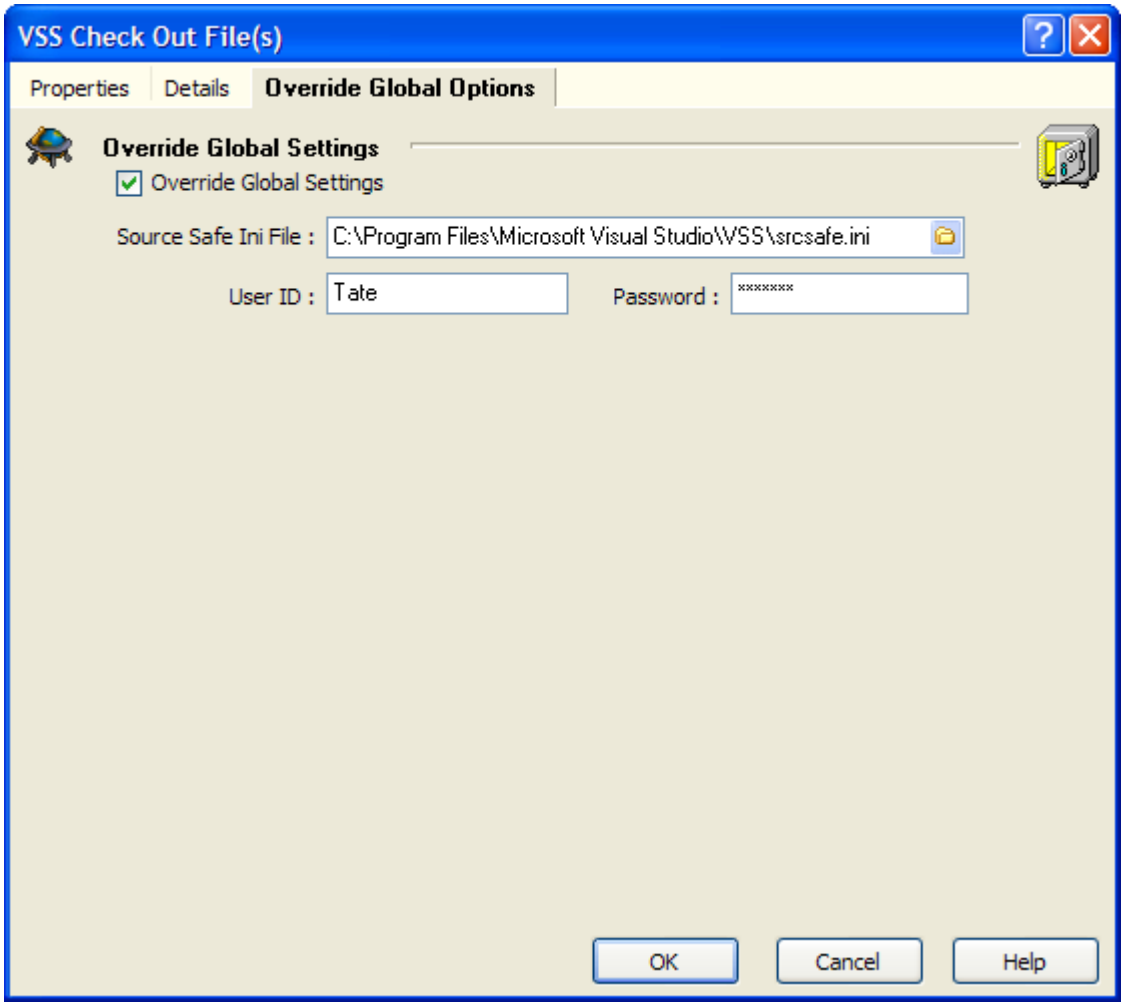
The working directory will be output to the FB Log and can be optionally written to a FinalBuilder variable.

5.5.2.12 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.

© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.3 Perforce

5.5.3.1 Perforce Options

Perforce help coming soon.

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.

5.5.3.2 Perforce Create Branch

Perforce help coming soon.

5.5.3.3 Perforce Delete Branch

Perforce help coming soon.

5.5.3.4 Perforce Sync

Perforce help coming soon.

5.5.3.5 Perforce Submit

Perforce help coming soon.

5.5.3.6 Perforce open for Edit

Perforce help coming soon.

5.5.3.7 Perforce Opened

Perforce help coming soon.

5.5.3.8 Perforce Create Label

Perforce help coming soon.

5.5.3.9 Perforce Update Label

Perforce help coming soon.

5.5.3.10 Perforce Delete Label

Perforce help coming soon.

5.5.3.11 Perforce Labelsync

Perforce help coming soon.

5.5.3.12 Perforce open for Delete

Perforce help coming soon.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.3.13 Perforce Tag

Perforce help coming soon.

5.5.3.14 Perforce Lock

Perforce help coming soon.

5.5.3.15 Perforce Unlock

Perforce help coming soon.

5.5.3.16 Perforce Revert

Perforce help coming soon.

5.5.3.17 Perforce Create Changelist

Perforce help coming soon.

5.5.3.18 Perforce Delete Changelist

Perforce help coming soon.

5.5.3.19 Perforce Generic

Perforce help coming soon.

5.5.3.20 Perforce Old Actions

The old set of Perforce actions has been deprecated.

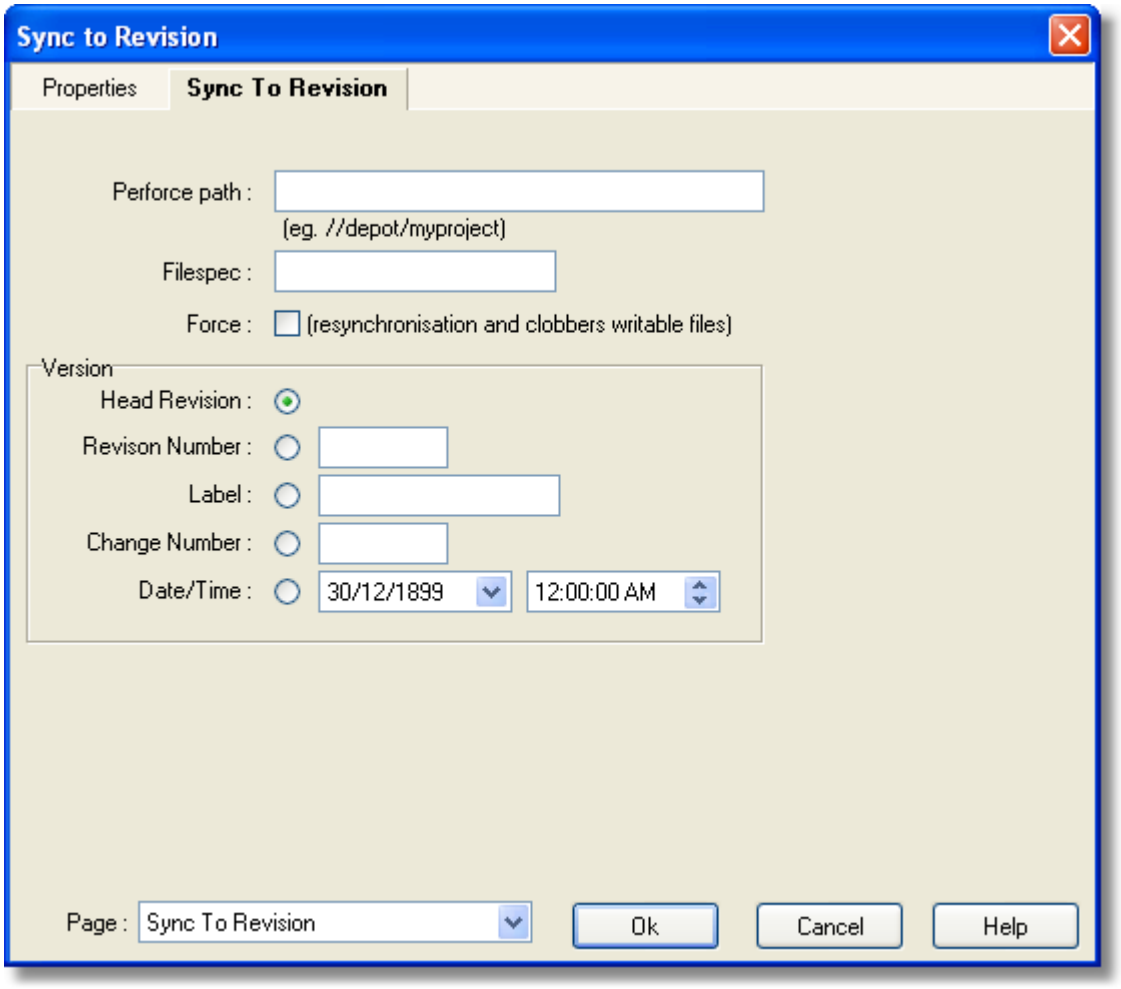
The help topics and actions are still available, but we encourage Perforce users to use the new set of actions.

5.5.3.20.1 Peforce Synchronise with View Action

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.

© 2001-2006 VSoft Technologies Pty Ltd



For all other Perforce commands use the "Perforce Command" and enter the command line options manually.

For more information on the sync command, see:

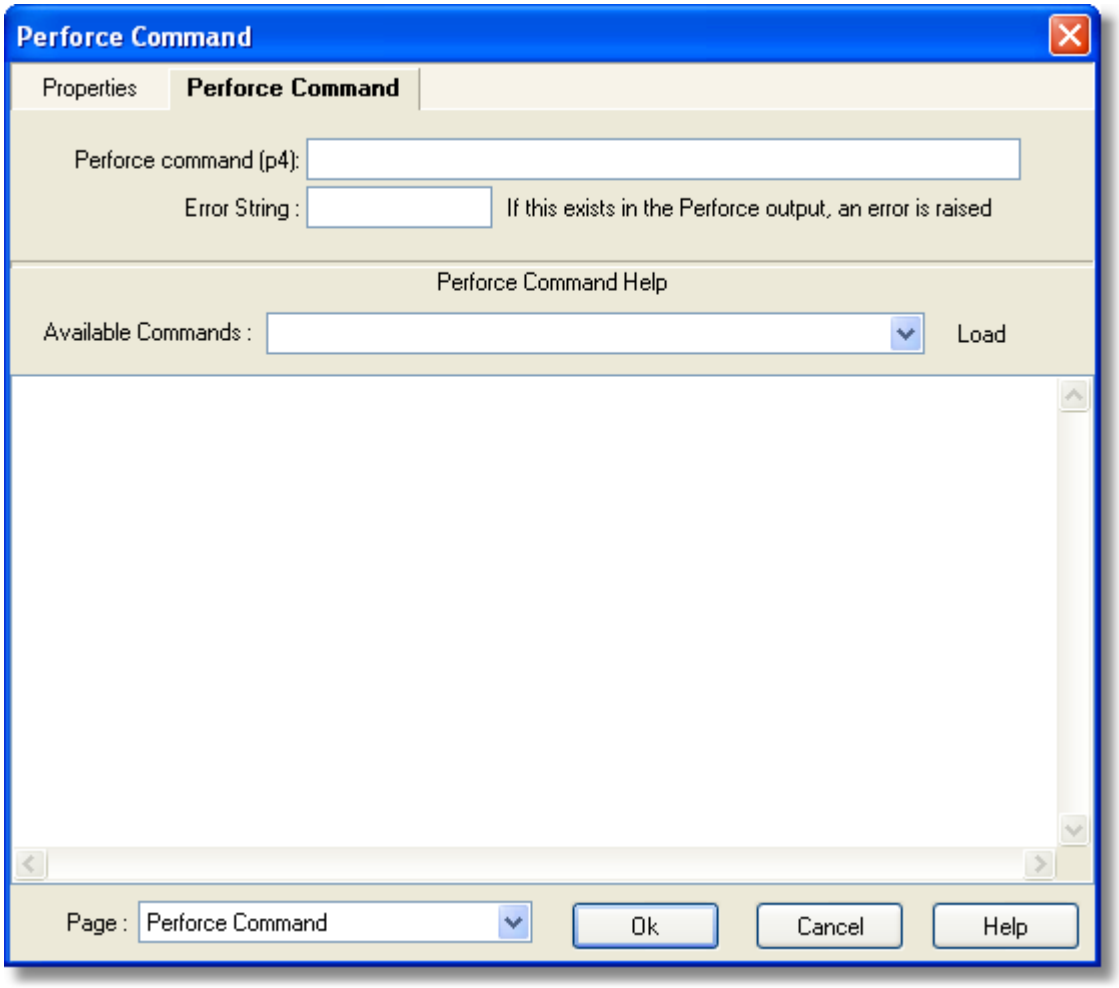
http://www.perforce.com/perforce/doc.032/manuals/cmdref/sync.html

This action was kindly provided by Tate Needham from t8software.com

© 2001-2006 VSoft Technologies Pty Ltd

5.5.3.20.2 Perforce Command Action

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

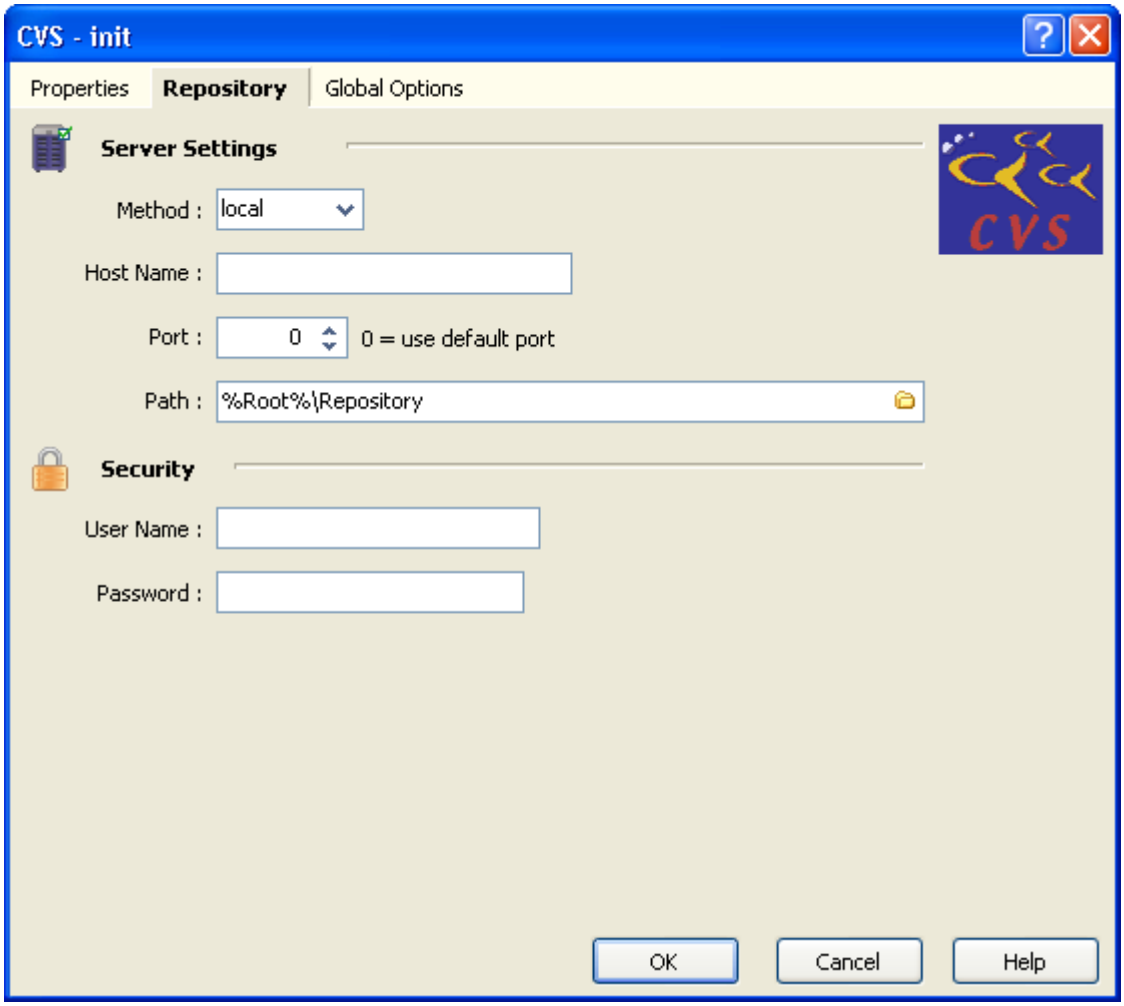
© 2001-2006 VSoft Technologies Pty Ltd

5.5.4 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 :

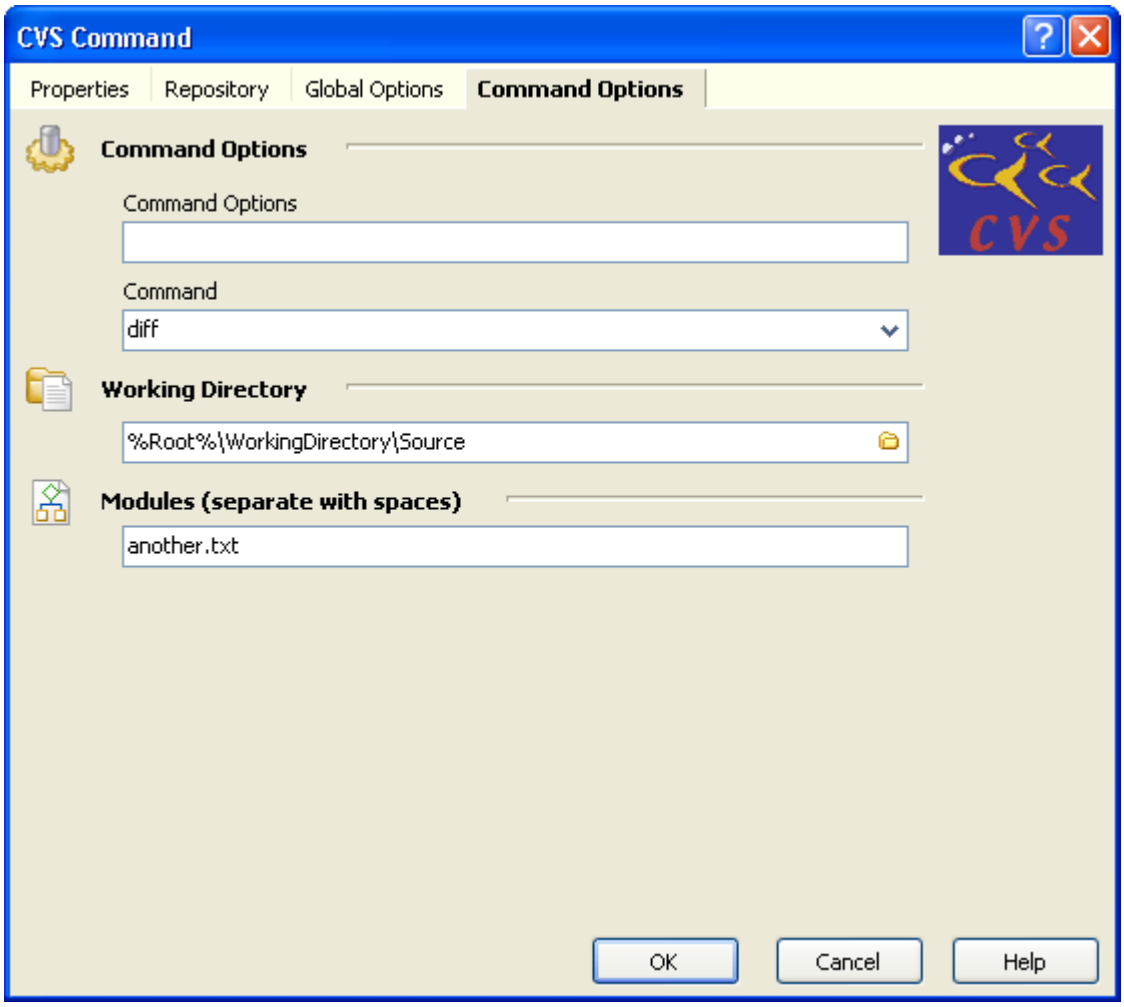
HOMEDRIVE=E:

© 2001-2006 VSoft Technologies Pty Ltd

HOMEPATH=\Documents and Settings\Vincent

5.5.4.1 CVS Command Action

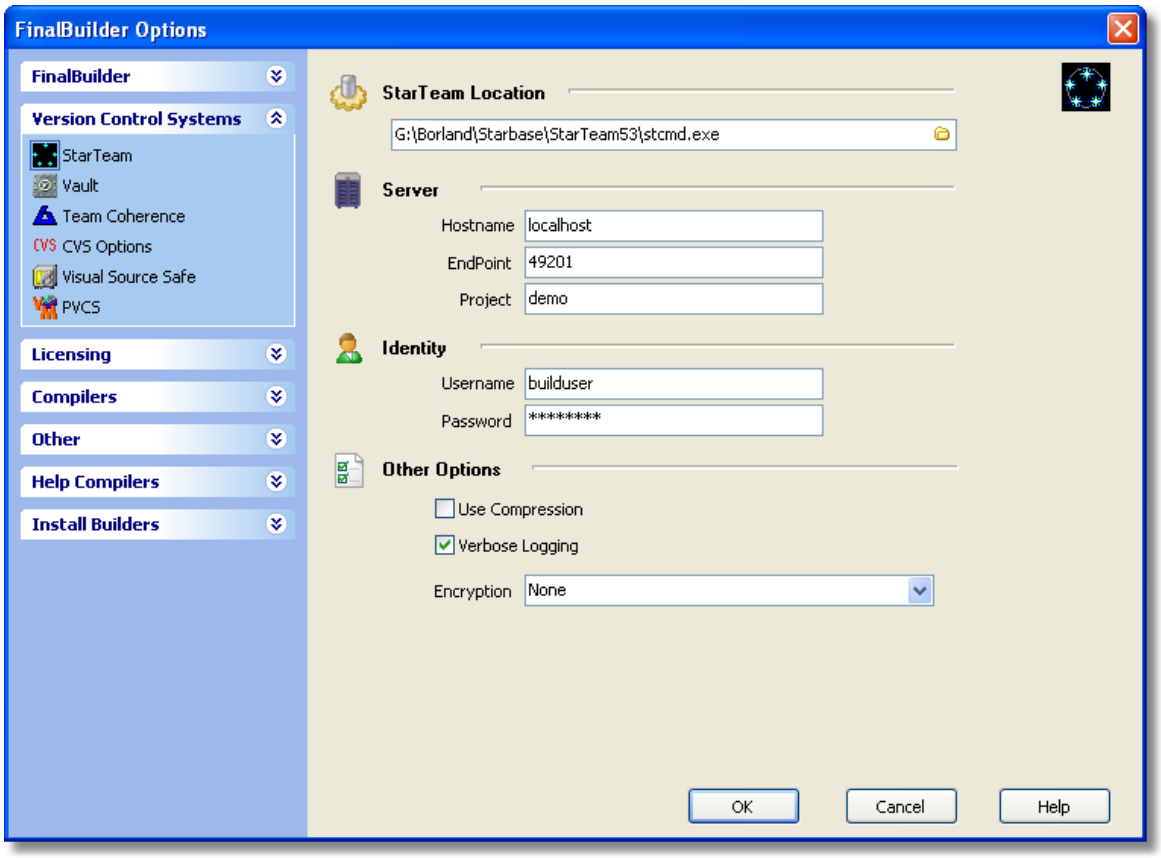
This action provides a generic CVS command interface. The list of commands in the Command drop down list is not the only commands that can be called, you can type the name of the command you wish to call into the combo box.



© 2001-2006 VSoft Technologies Pty Ltd

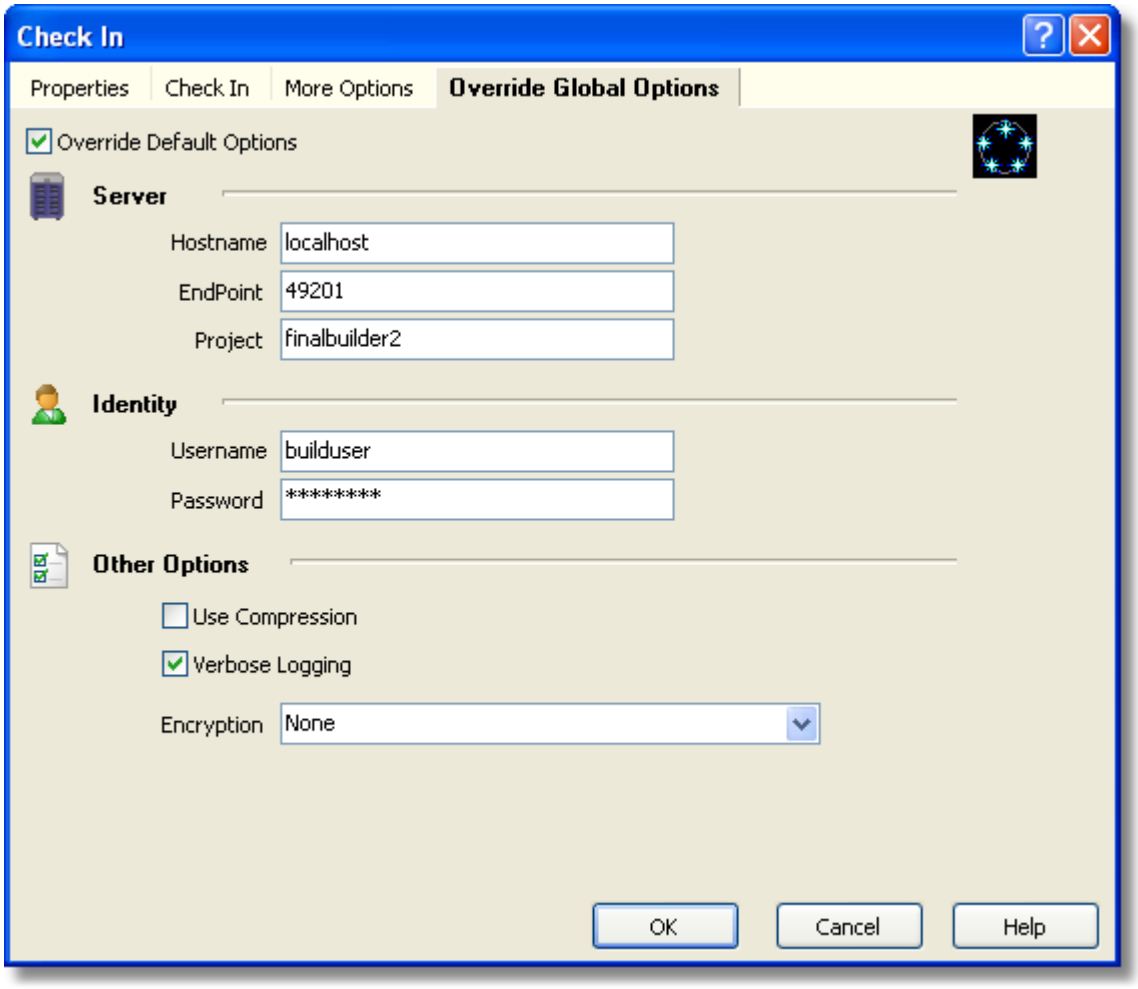
5.5.5 Borland StarTeam Actions

These Actions support Borland StarTeam 5.2.x or higher. They provide an easy to use wrapper around the StarTeam command line 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.

© 2001-2006 VSoft Technologies Pty Ltd



5.5.5.1 StarTeam Check In Action

This action calls the StarTeam command line tool to check in files.

5.5.5.2 StarTeam Check Out Action

This action calls the StarTeam command line tool to check out files.

5.5.5.3 StarTeam Lock/Unlock Files Action

This action calls the StarTeam command line tool to Lock/Unlock files.

5.5.5.4 StarTeam Create Label Action

This action calls the StarTeam command line tool to create a Label

5.5.5.5 StarTeam Apply Label Action

This action calls the StarTeam command line tool to Apply a Label

© 2001-2006 VSoft Technologies Pty Ltd

5.5.5.6 StarTeam Update Status Action

This action calls the StarTeam command line tool to Update the status of files.

5.5.5.7 StarTeam Delete Files Action

This action calls the StarTeam command line tool to delete files.

5.5.5.8 StarTeam Add Files Action

This action calls the StarTeam command line tool to Add files.

5.5.5.9 StarTeam List Files Action

This action calls the StarTeam command line tool to List files.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.6 QVCS Actions

5.5.6.1 QVCS Add File Action

The QVCS Add File action enables you to add files to your source code repository using QVCS as part of your build process.

For more information on QVCS, see http://www.qumasoft.com/

5.5.6.2 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 as part of your build process.

For more information on QVCS, see http://www.qumasoft.com/

5.5.6.3 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 as part of your build process.

For more information on QVCS, see http://www.qumasoft.com/

5.5.6.4 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 as part of your build process.

For more information on QVCS, see http://www.qumasoft.com/

5.5.6.5 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 as part of your build process.

For more information on QVCS, see http://www.qumasoft.com/

5.5.6.6 QVCS Undo Check Out File(s) Action

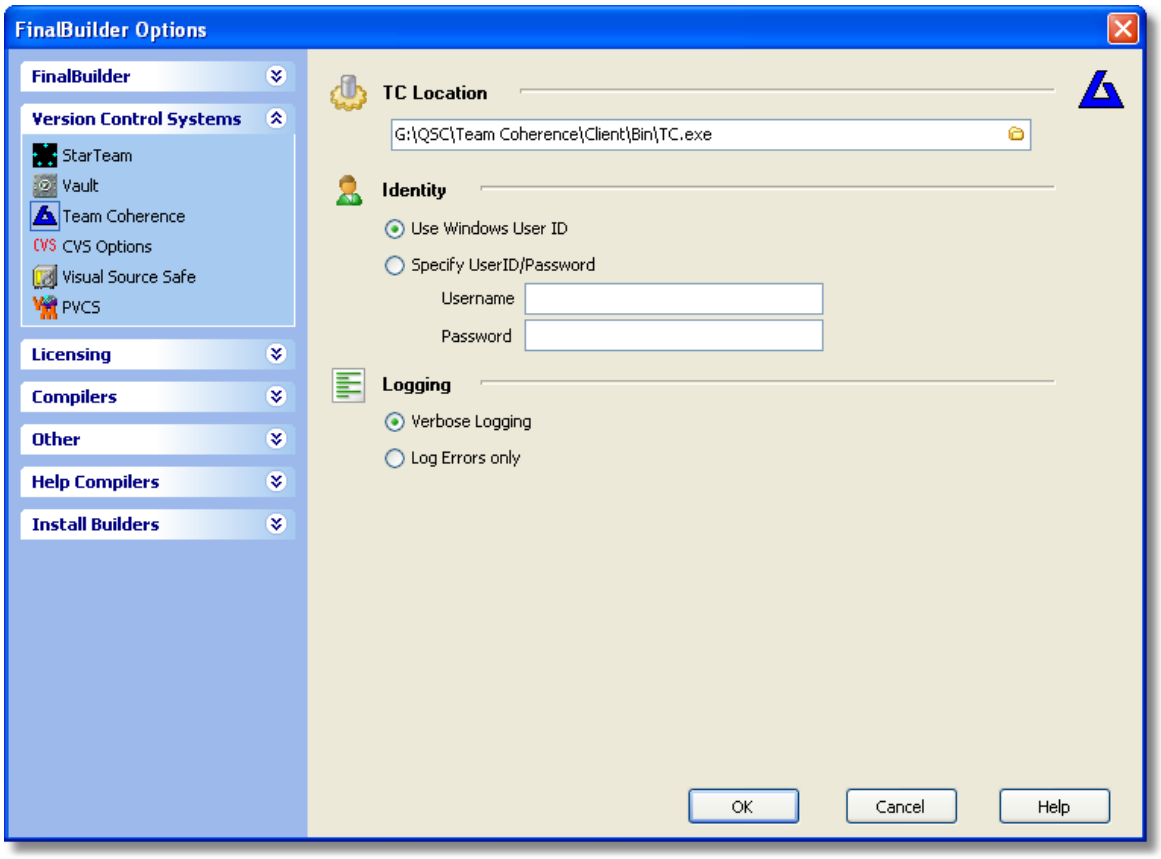
The QVCS Undo Check Out File(s) action enables you to undo checkouts of specified files using QVCS as part of your build process.

For more information on QVCS, see http://www.qumasoft.com/

© 2001-2006 VSoft Technologies Pty Ltd

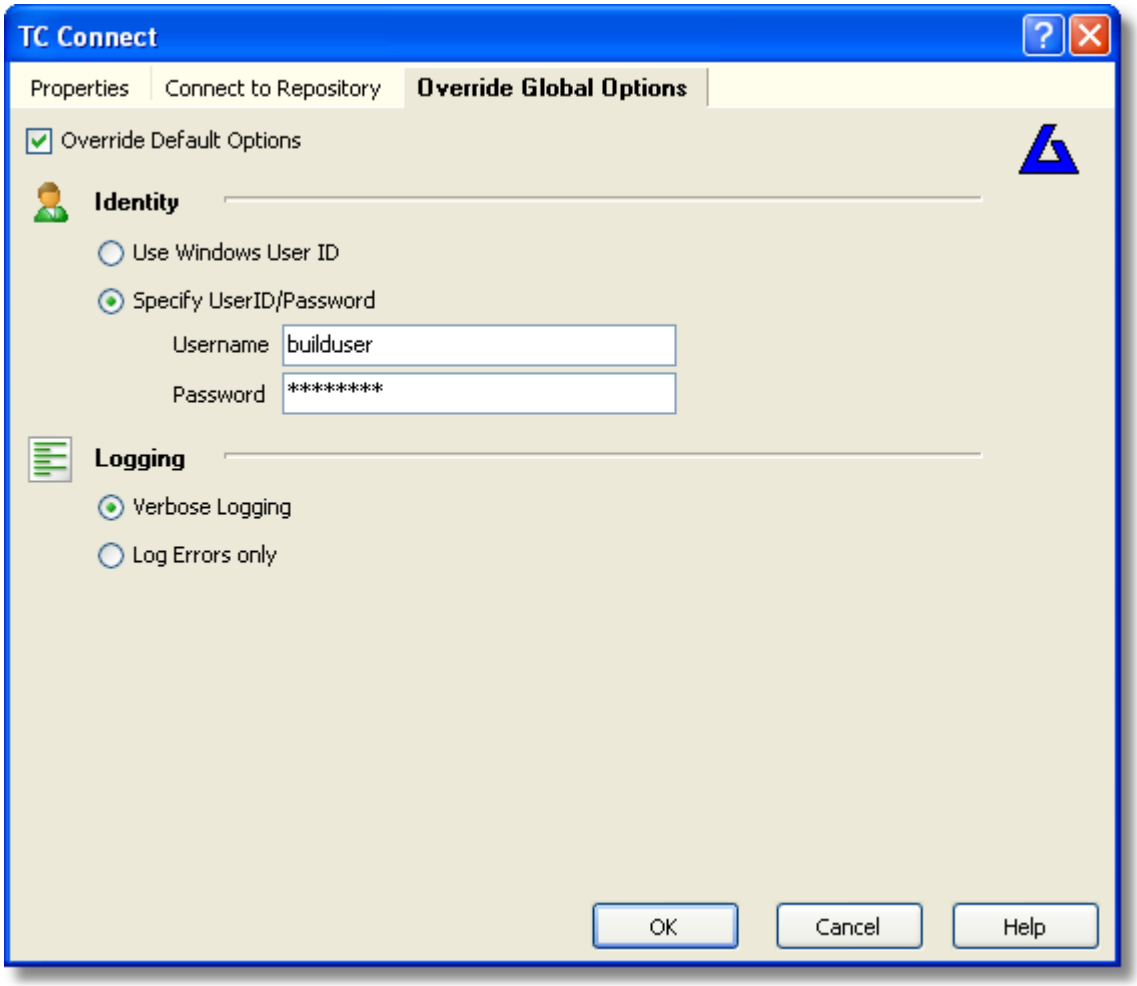
5.5.7 QSC Team Coherence Actions

These Actions support Team Coherence 7.1 or higher. They provide an easy to use wrapper around the TC command line too. 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.

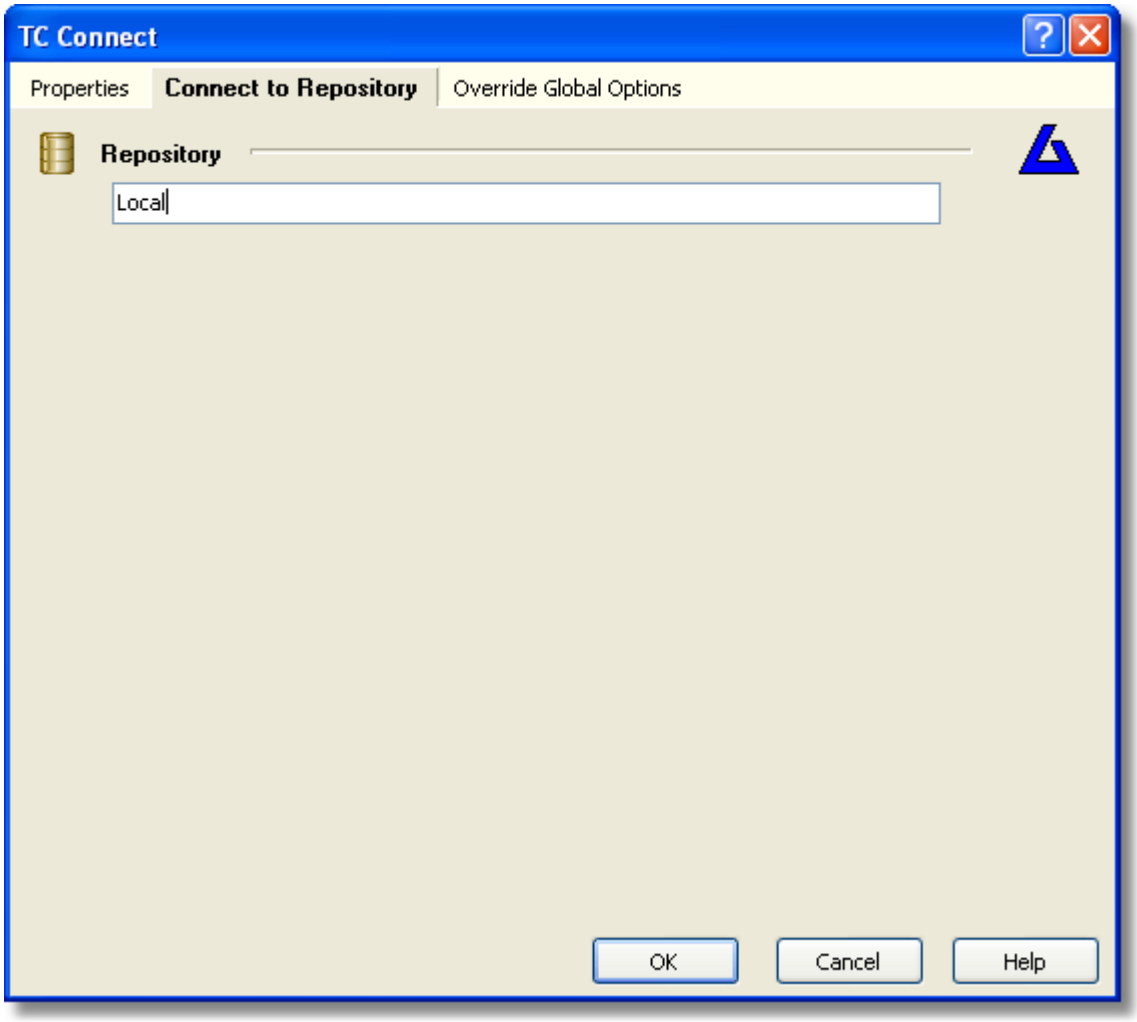
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.1 Team Coherence Connect Action

Connects to a different repository.



Remarks

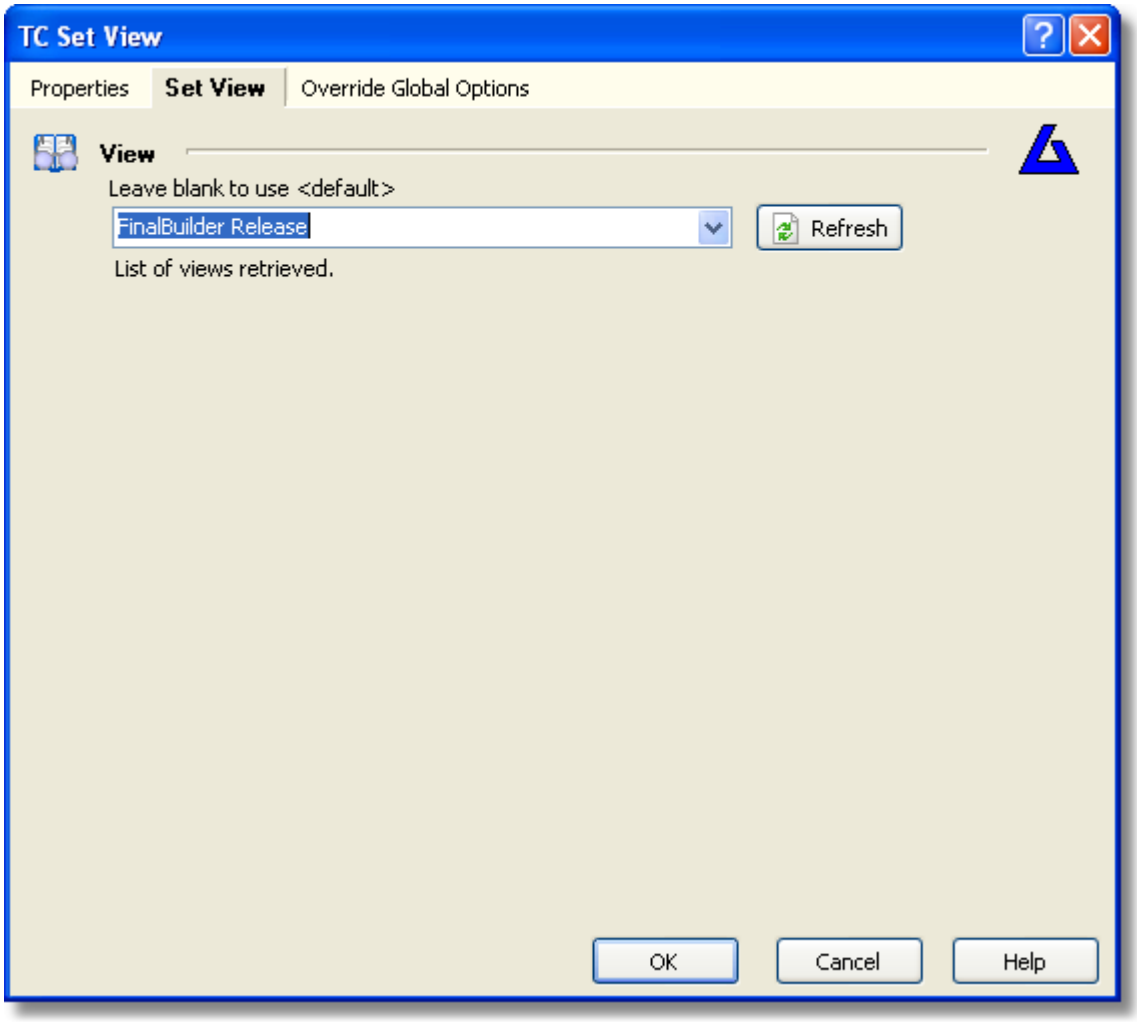
When you connect to a different repository, it becomes the default for subsequent commands.

Note that you must be connecting to a previously created repository connection. You can define connections using the Team Coherence Version Manager.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.2 Team Coherence Set View Action

Selects a different view and makes it current.



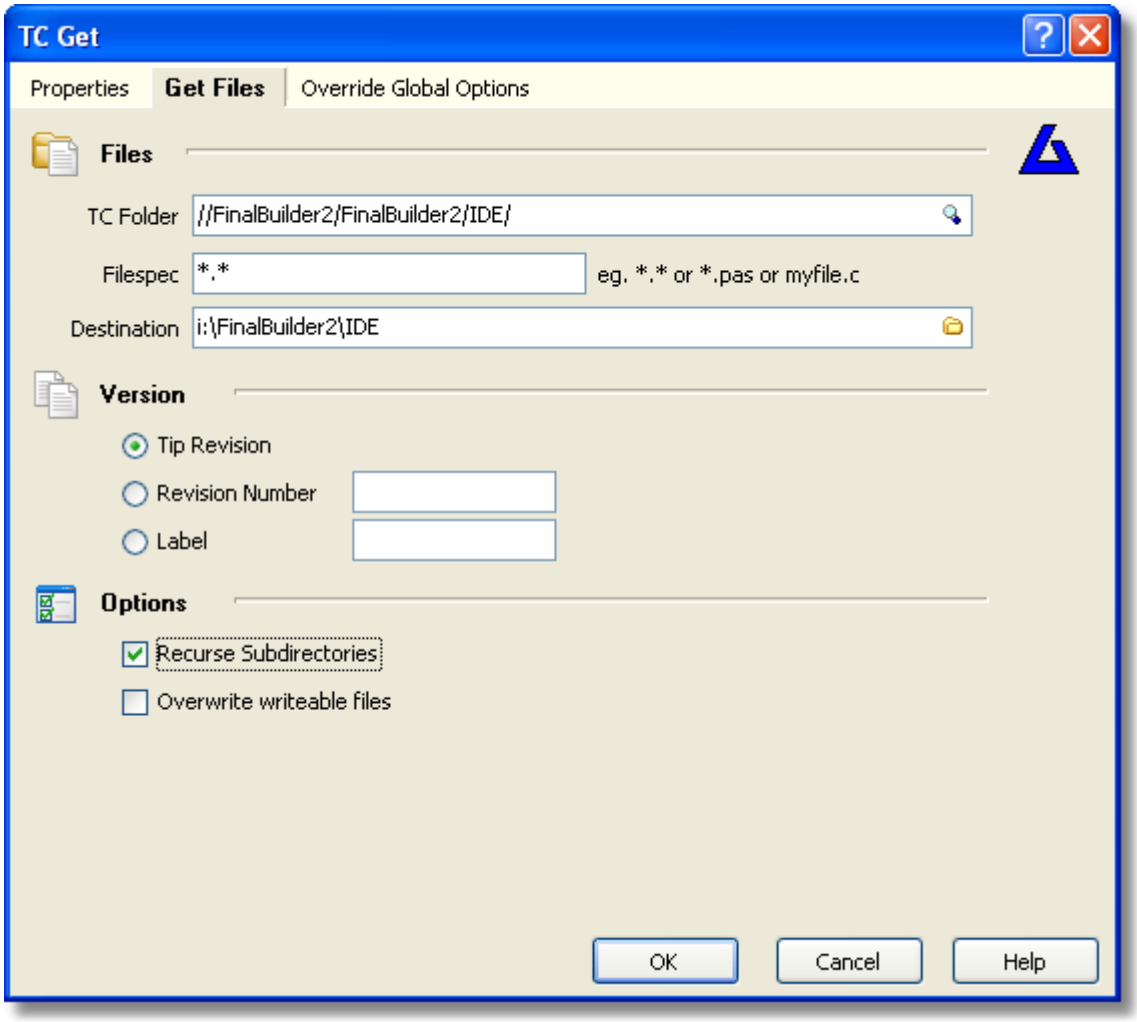
Remarks

For more information on Views, see the main Team Coherence help file.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.3 Team Coherence Get Action

Gets a read-only copy of the specified files.



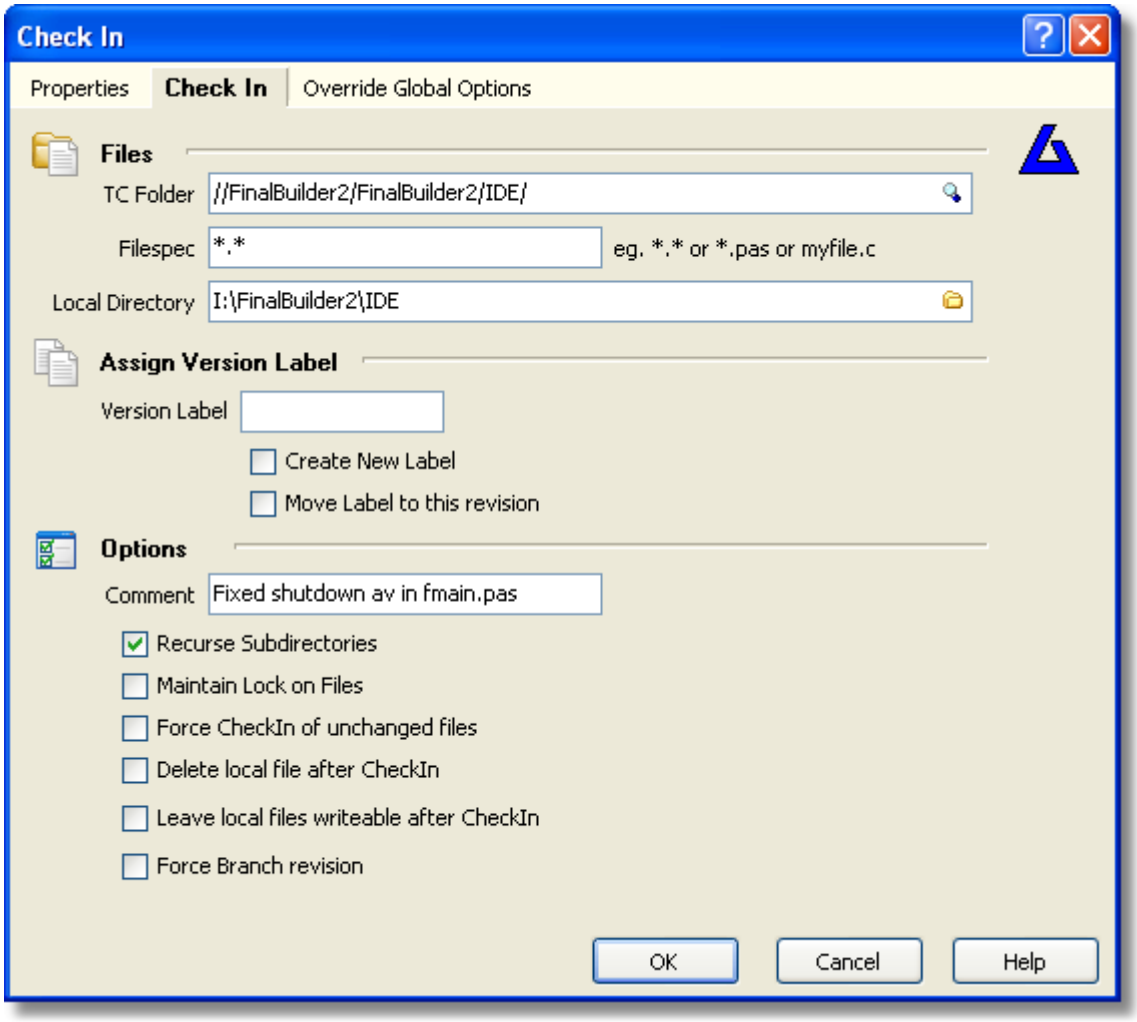
Remarks

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.

5.5.7.4 Team Coherence Check In Action

Updates Team Coherence with changes made to a checked out file and unlocks the archive.

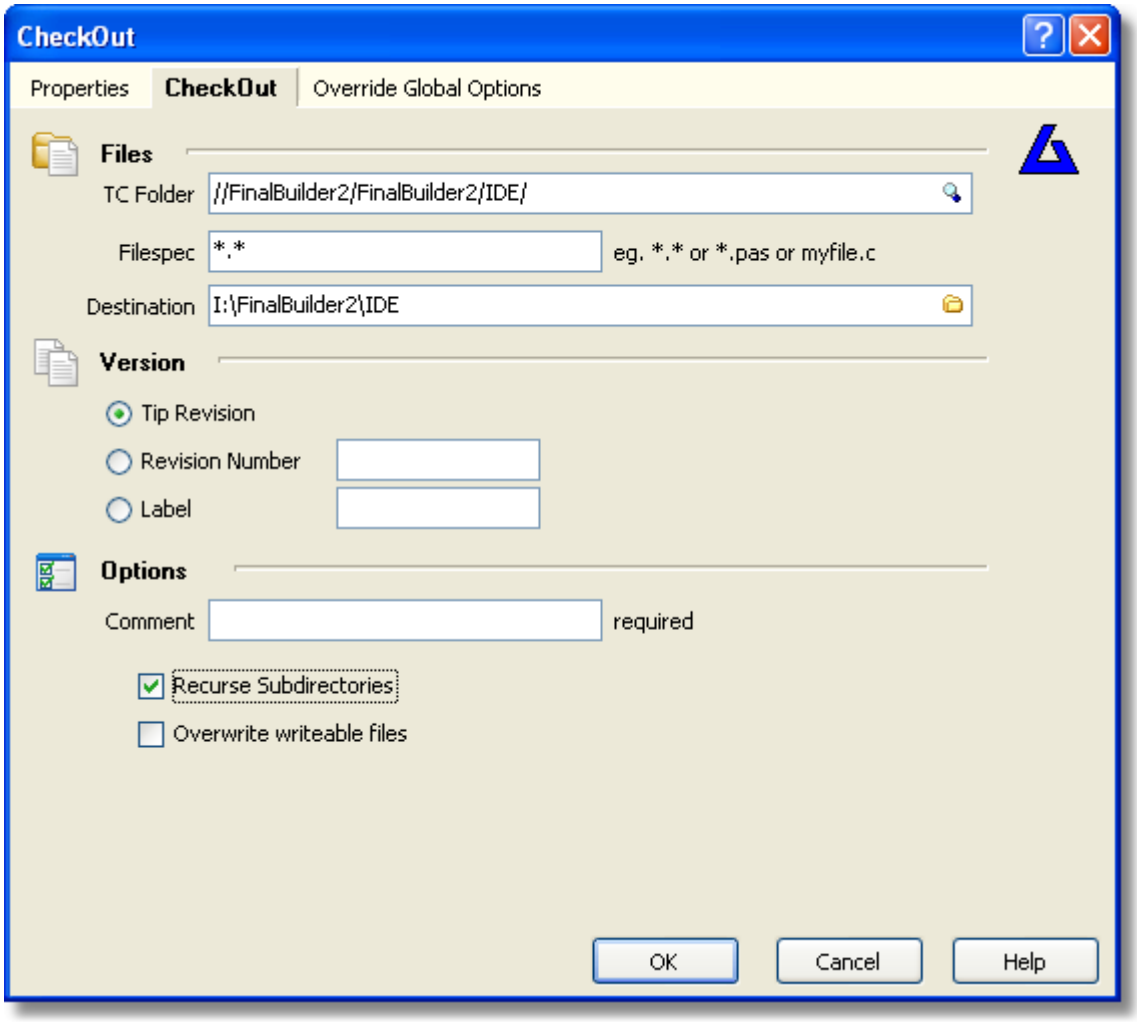
© 2001-2006 VSoft Technologies Pty Ltd



5.5.7.5 Team Coherence Check Out Action

Extracts a file from the current folder to the current directory for the purposes of editing.

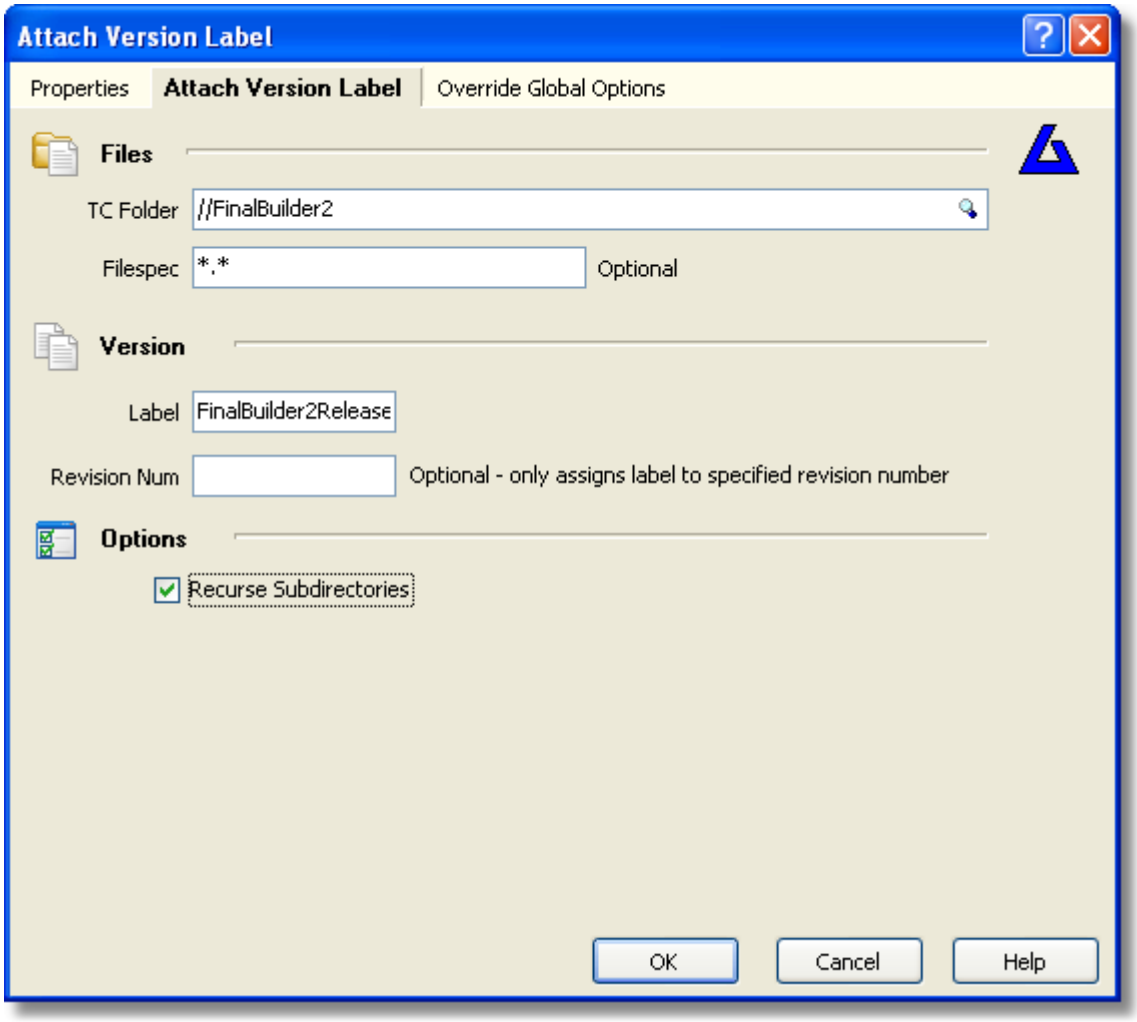
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.6 Team Coherence Attach Label Action

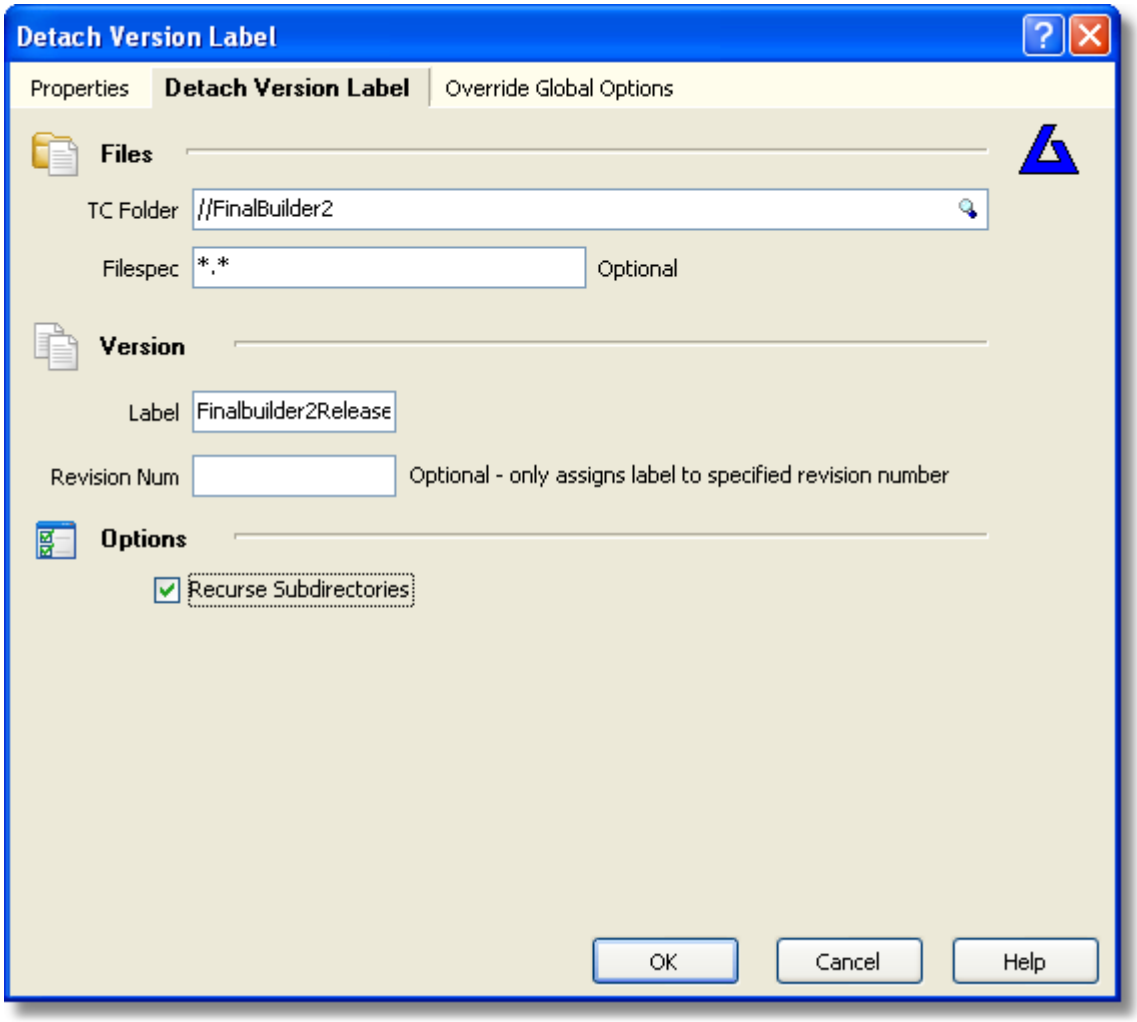
Attaches a Version Label to the latest revision of a file or to a specified revision.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.7 Team Coherence Detach Label Action

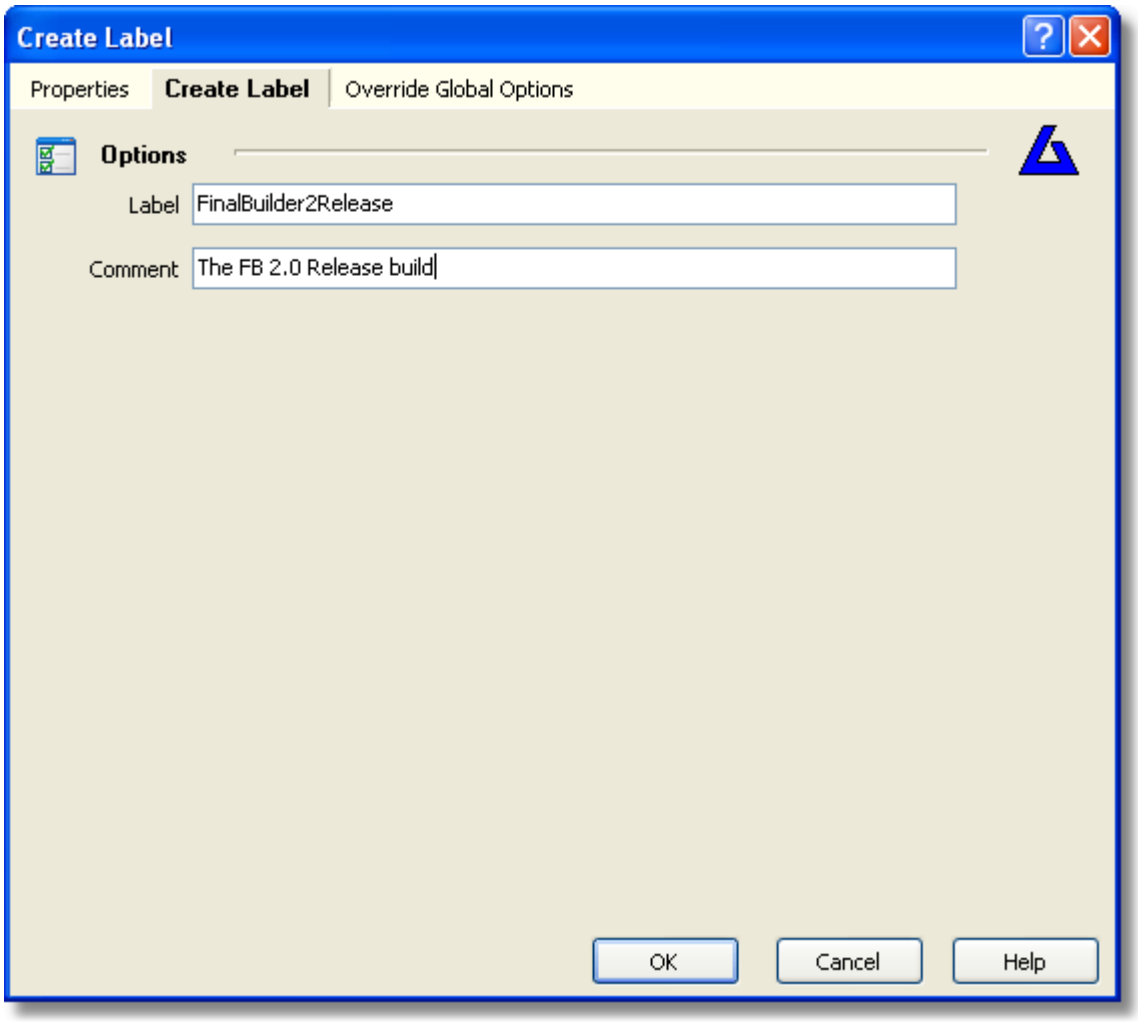
Detaches the specified label from the specified objects.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.8 Team Coherence Create Label Action

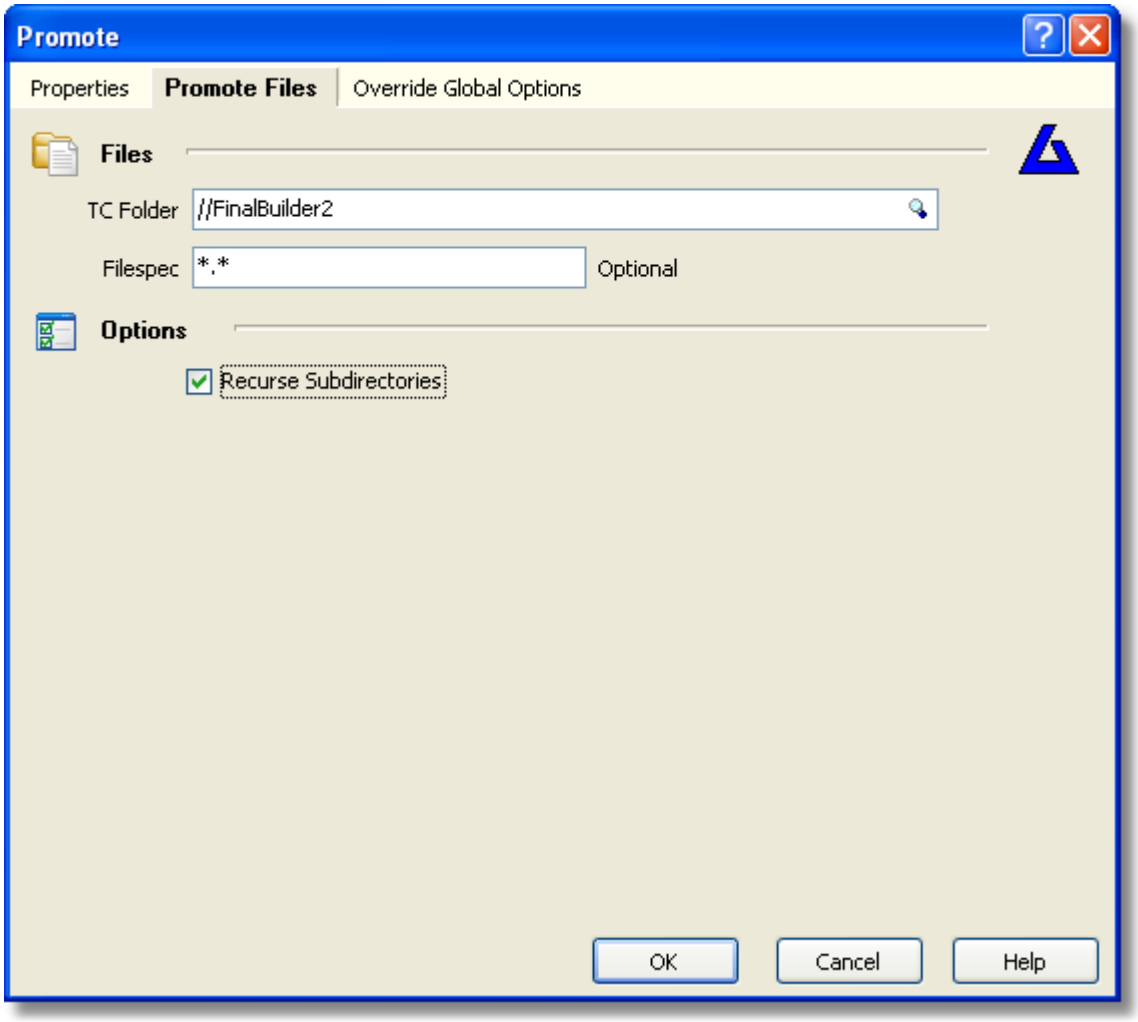
Creates a new Version Label.



5.5.7.9 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.

© 2001-2006 VSoft Technologies Pty Ltd



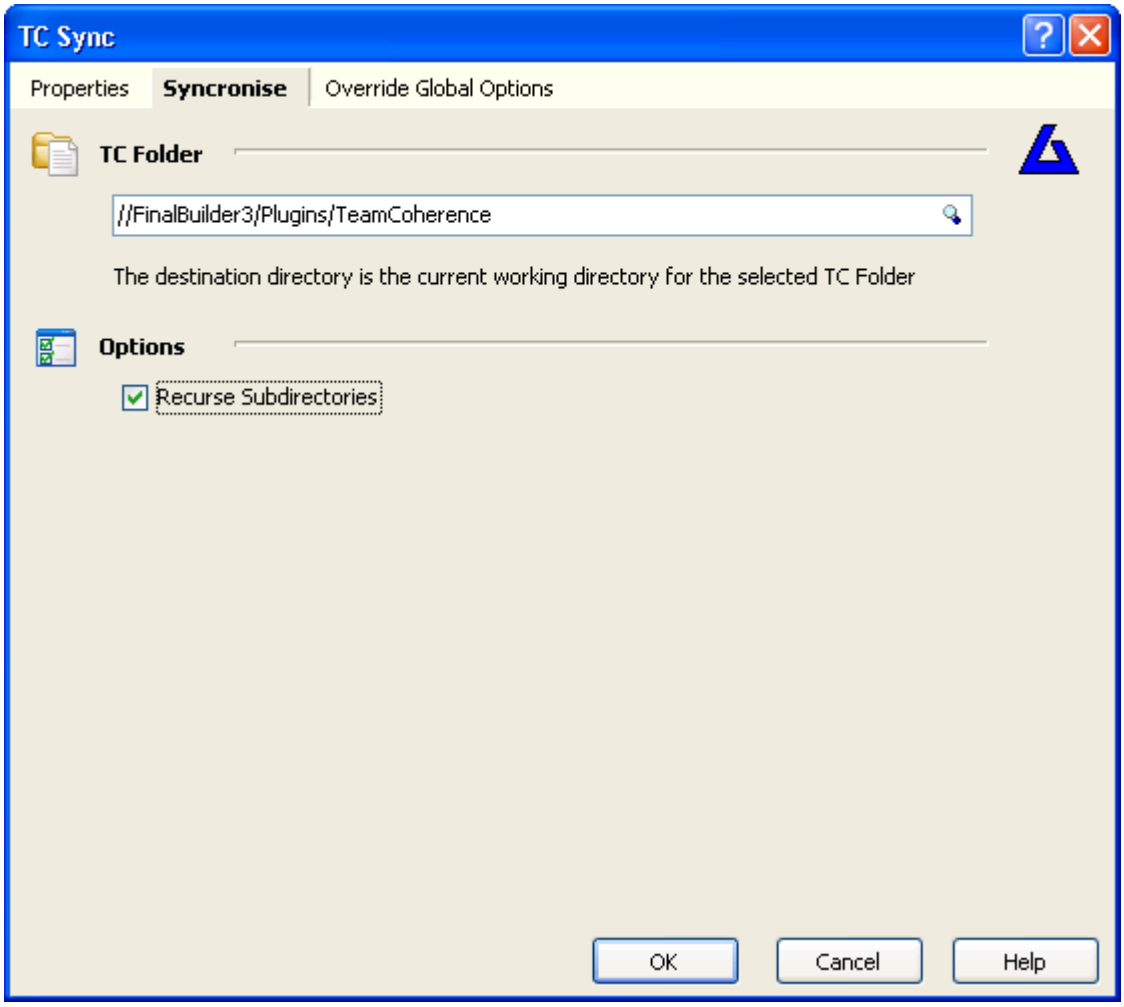
Both the Tip Revision and the next higher Promotion Level are defined by the currently selected View. If the current View is based on a Version Label, the revision that has the Version Label attached will be promoted to the first Promotion Level. If the current View is based on a Promotion Level, the 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.

5.5.7.10 Team Coherence Sync

The Sync command will synchronise your out of date or missing local files with the current files in the repository.

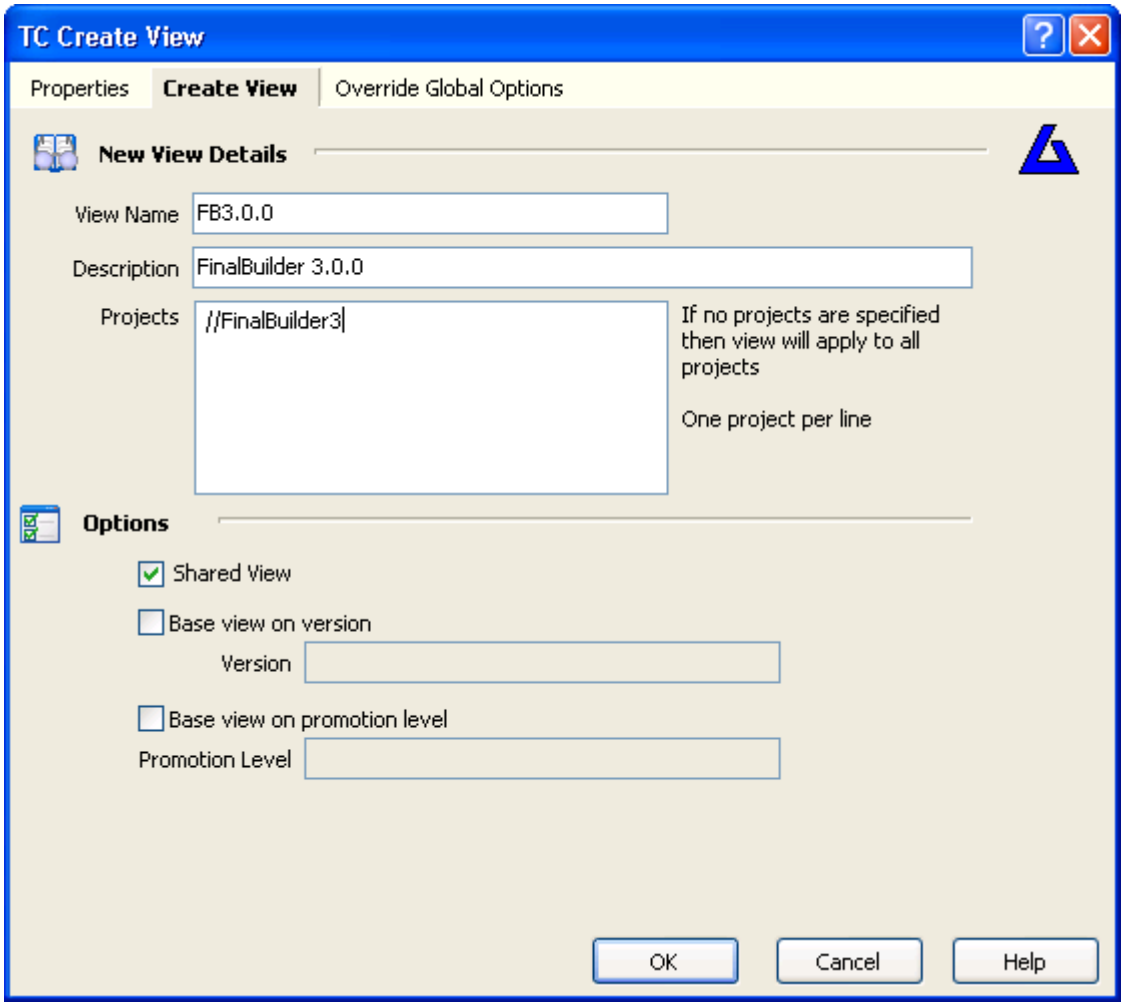
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.7.11 Team Coherence Create View

Create a new View in the repository.



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

© 2001-2006 VSoft Technologies Pty Ltd

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.

5.5.7.12 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.

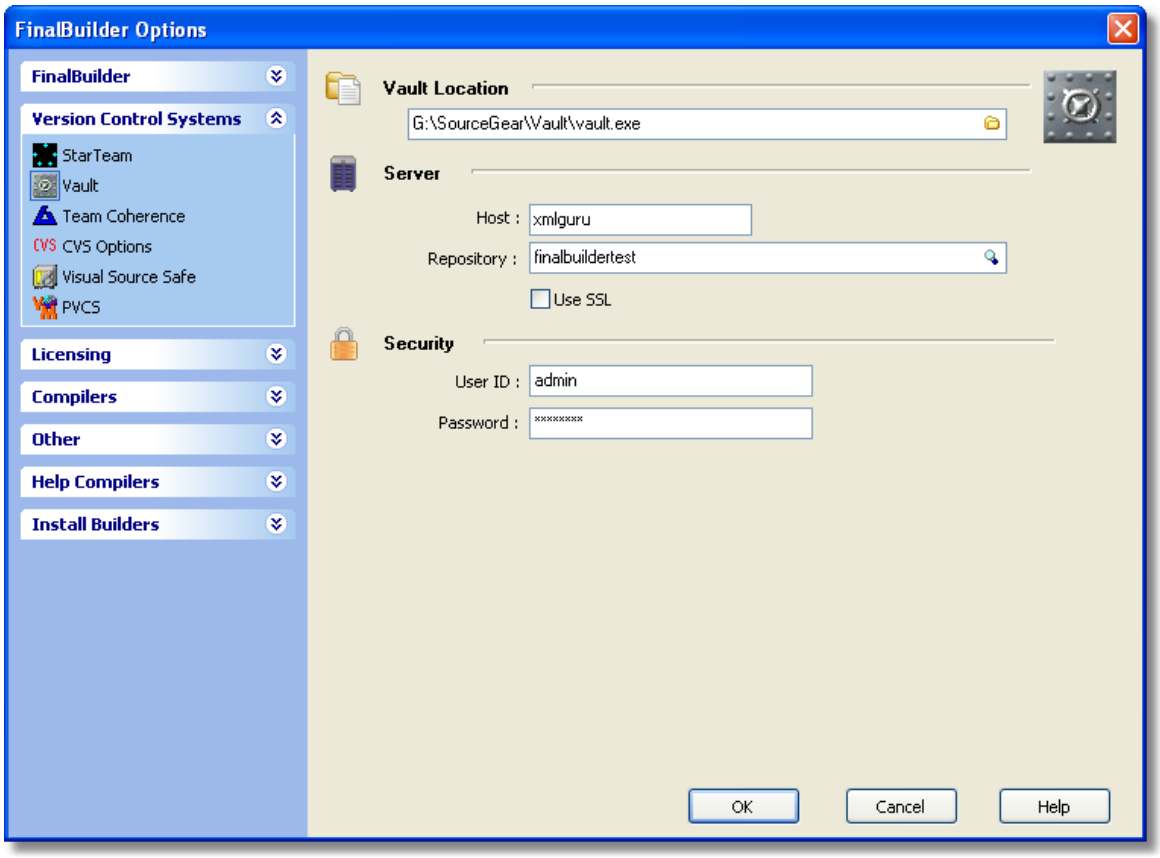
5.5.7.13 Team Coherence Delete View

Delete a view in the TC repository. See Team Coherence Create View for more information on views.

© 2001-2006 VSoft Technologies Pty Ltd

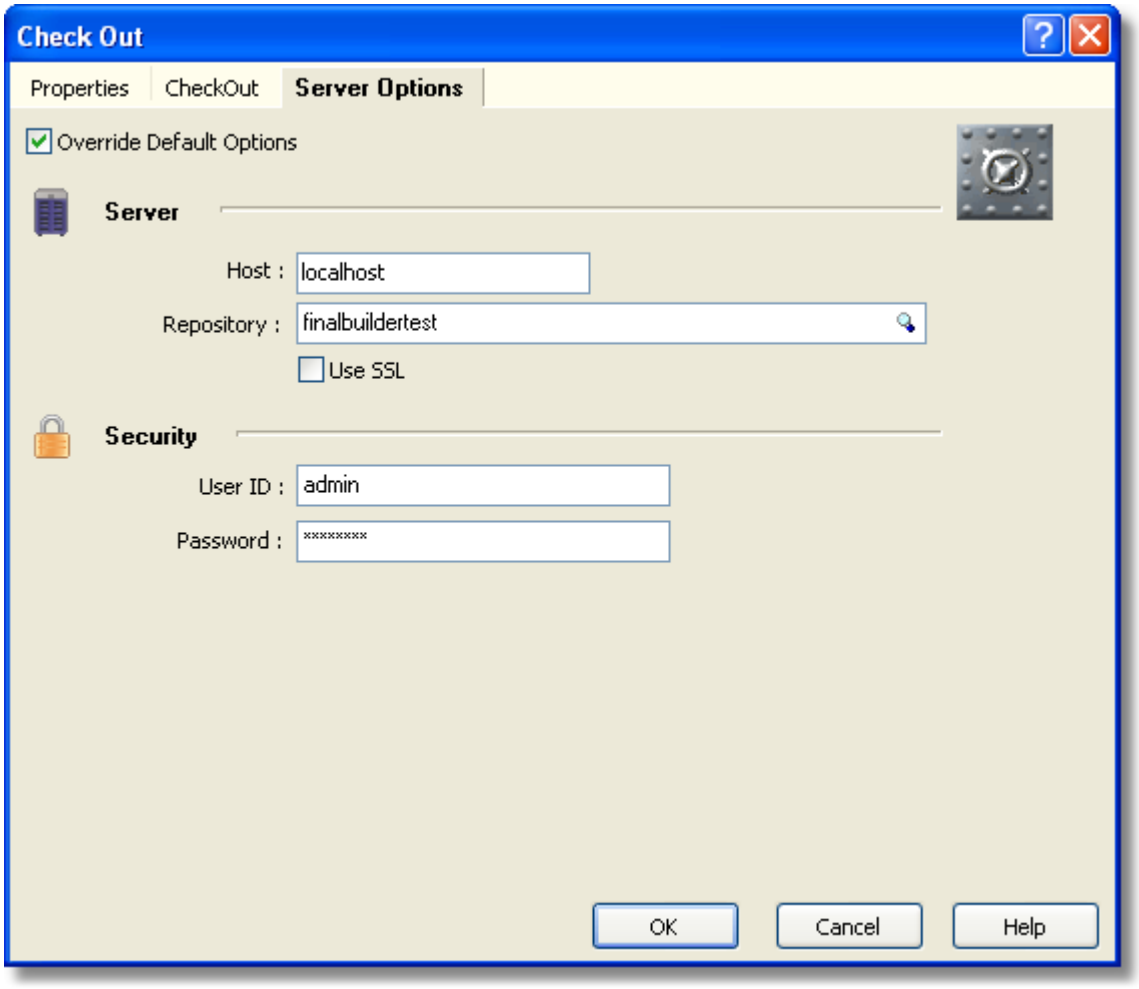
5.5.8 SourceGear Vault Actions

These Actions support SourceGear Vault 1.2 or higher. They provide an easy to use wrapper around the Vault command line 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.

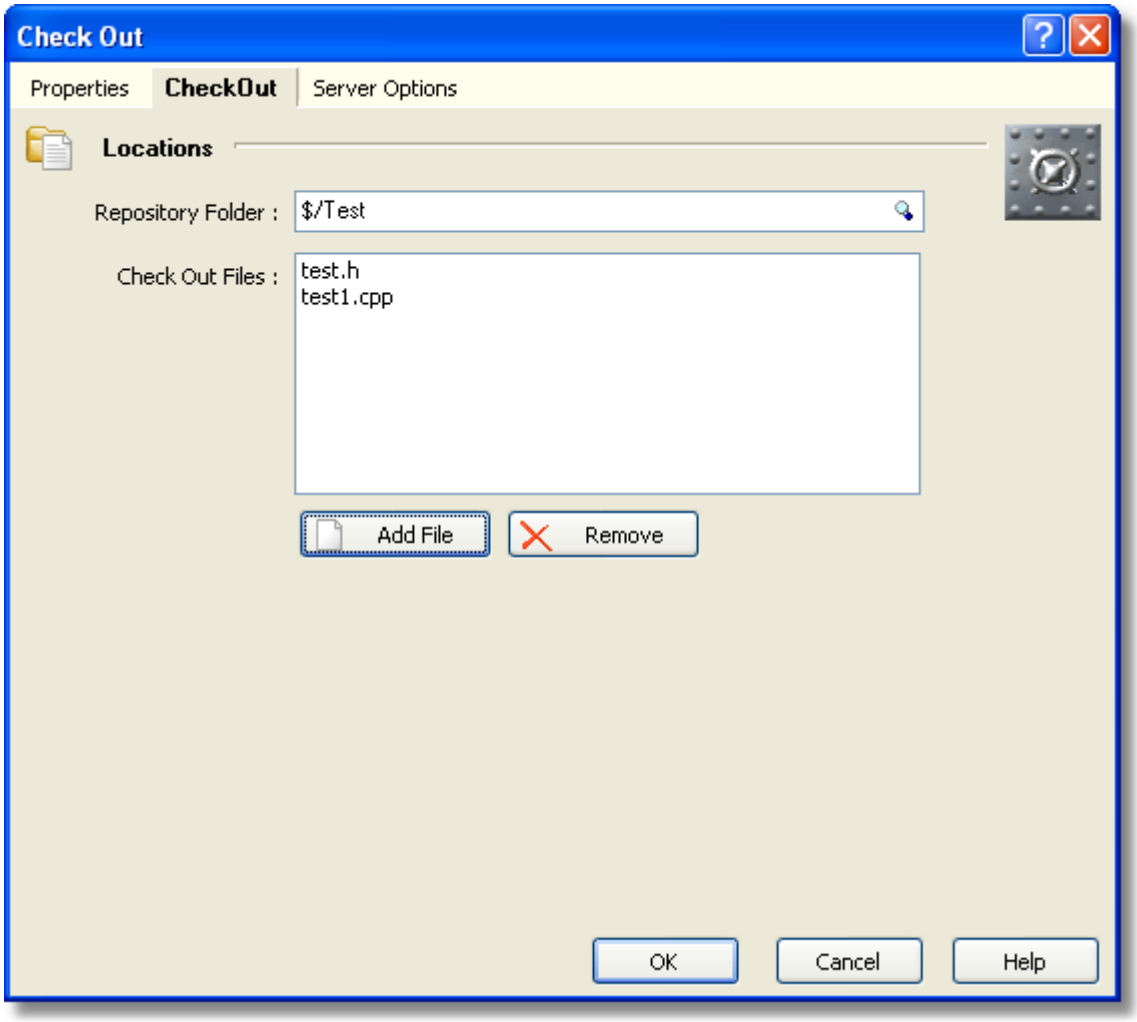
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.1 Vault Check Out Action

CHECKOUT will checkout files from the repository.

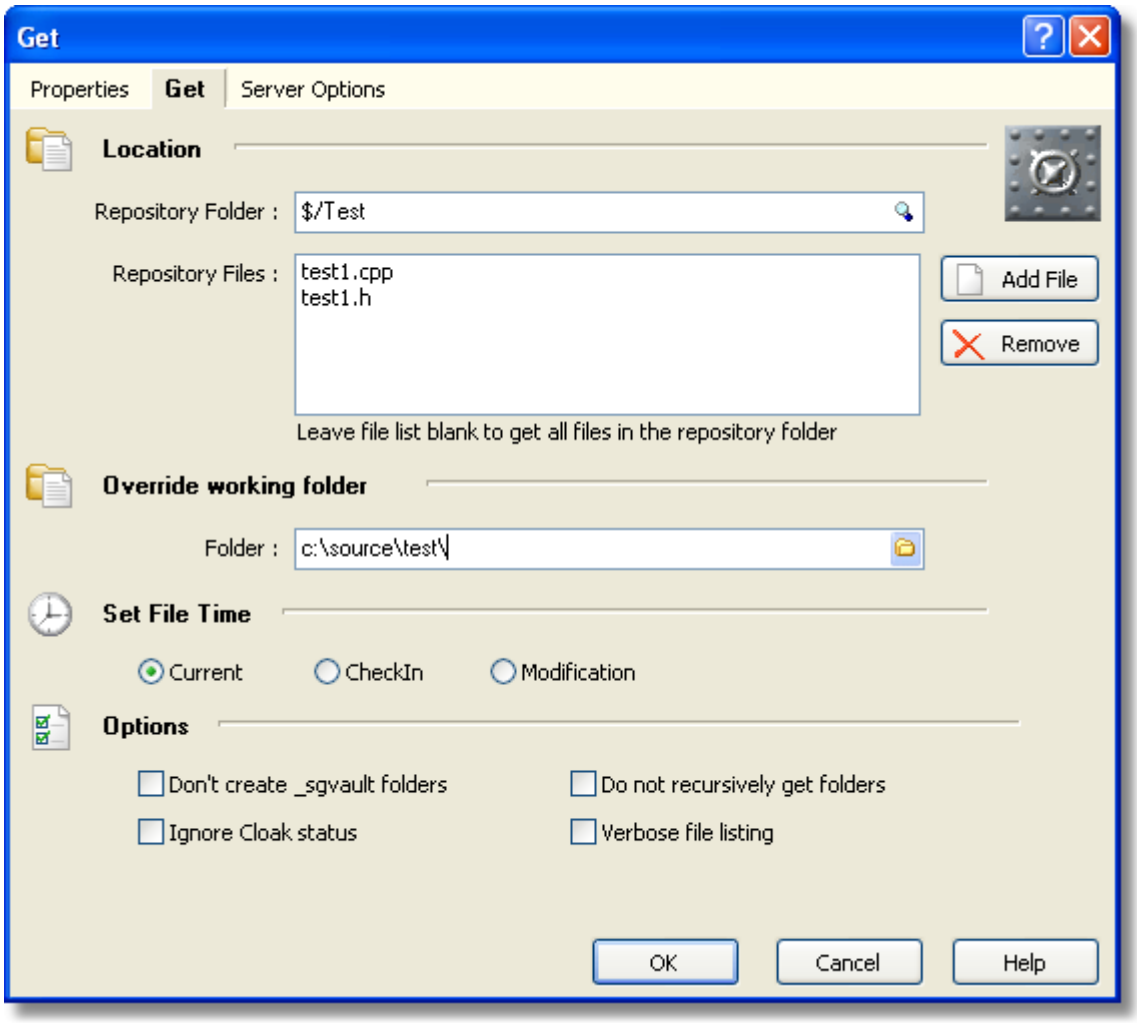


Server and authentication information is specified by:
-host host
Hostname of the vault server to connect to - also see SERVER
-ssl
Enables SSL for server connection
-user username
Username to use when connecting to server - also see USERNAME
-password password
Password to use when connecting to server
-repository repositoryname
Repository to connect to

5.5.8.2 Vault Get Action

GET will retrieve the latest version of files or folders in the repository as specified by repositorypath(s). The files will be stored in their corresponding working folders on the local system.

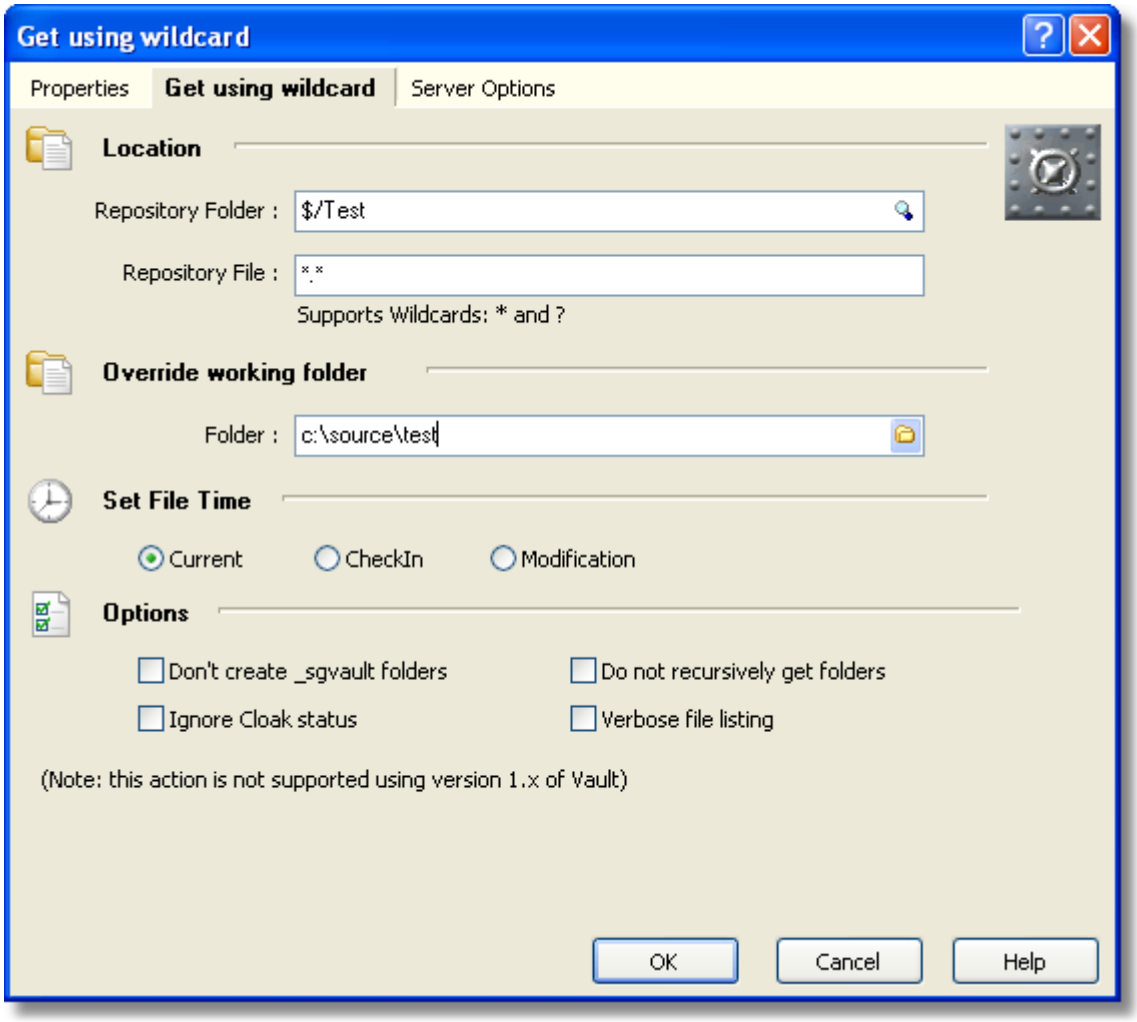
© 2001-2006 VSoft Technologies Pty Ltd



5.5.8.3 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.

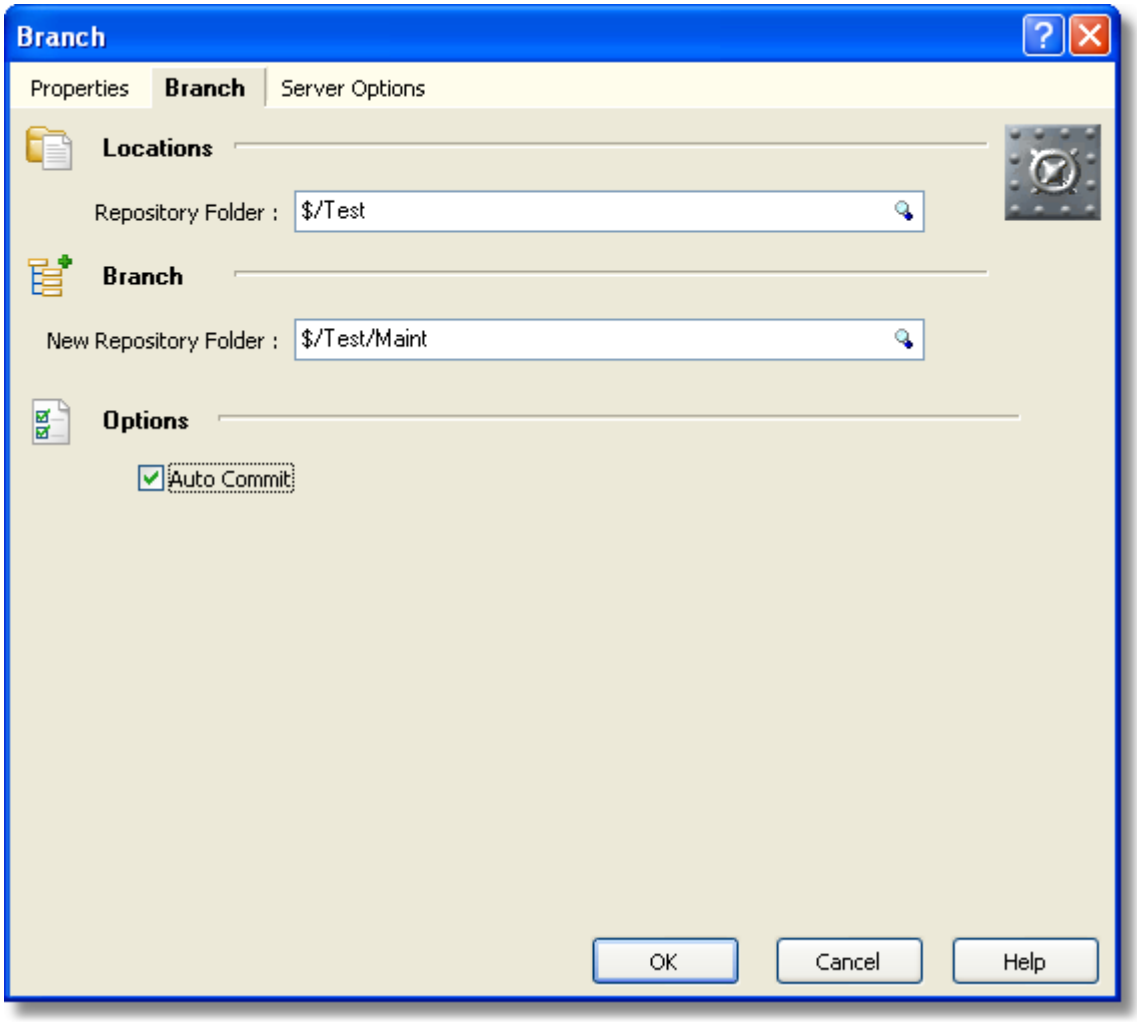
© 2001-2006 VSoft Technologies Pty Ltd



5.5.8.4 Vault Branch Action

BRANCH will create a branch for the repository folder specified by repository folder at New repository folder.

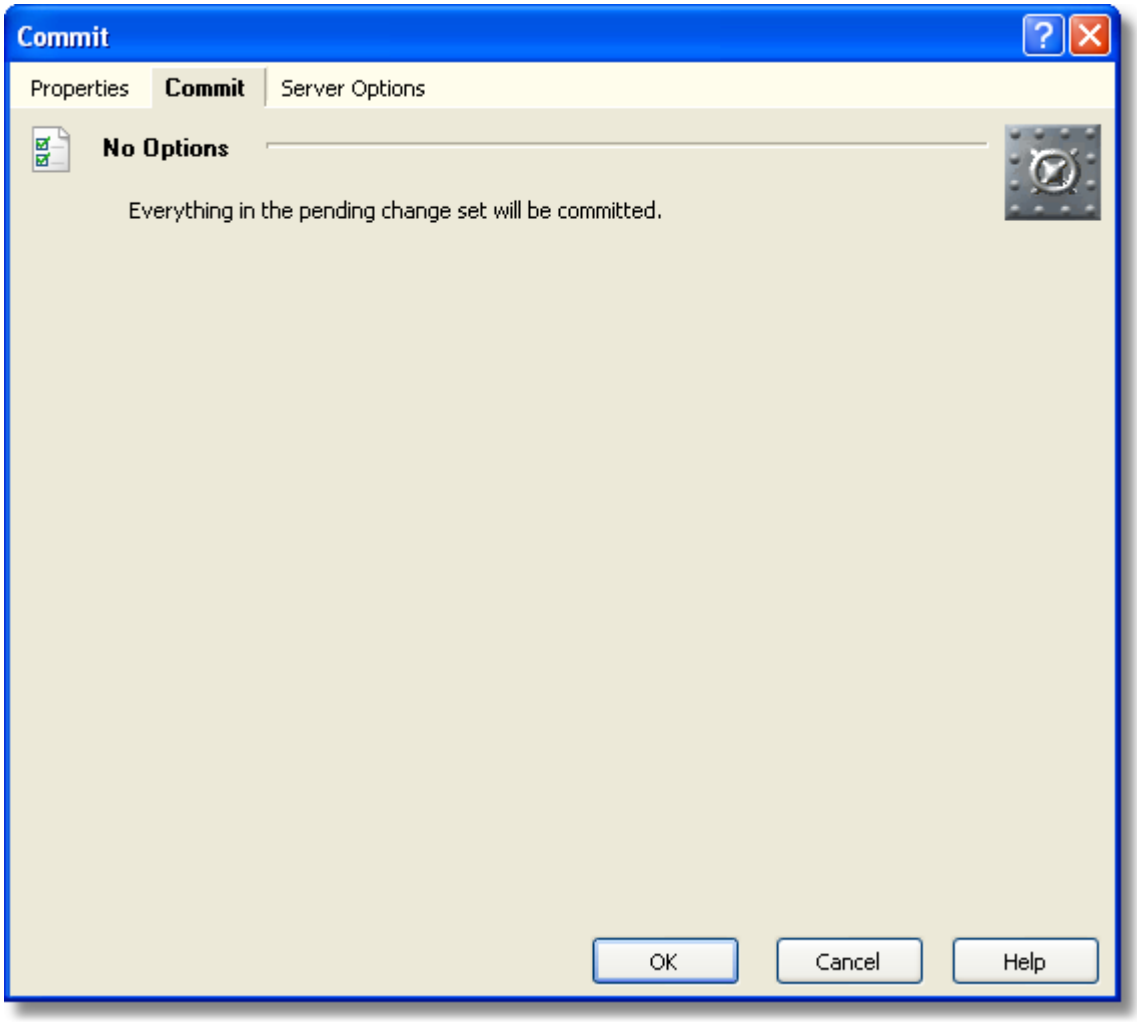
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.5 Vault Commit Action

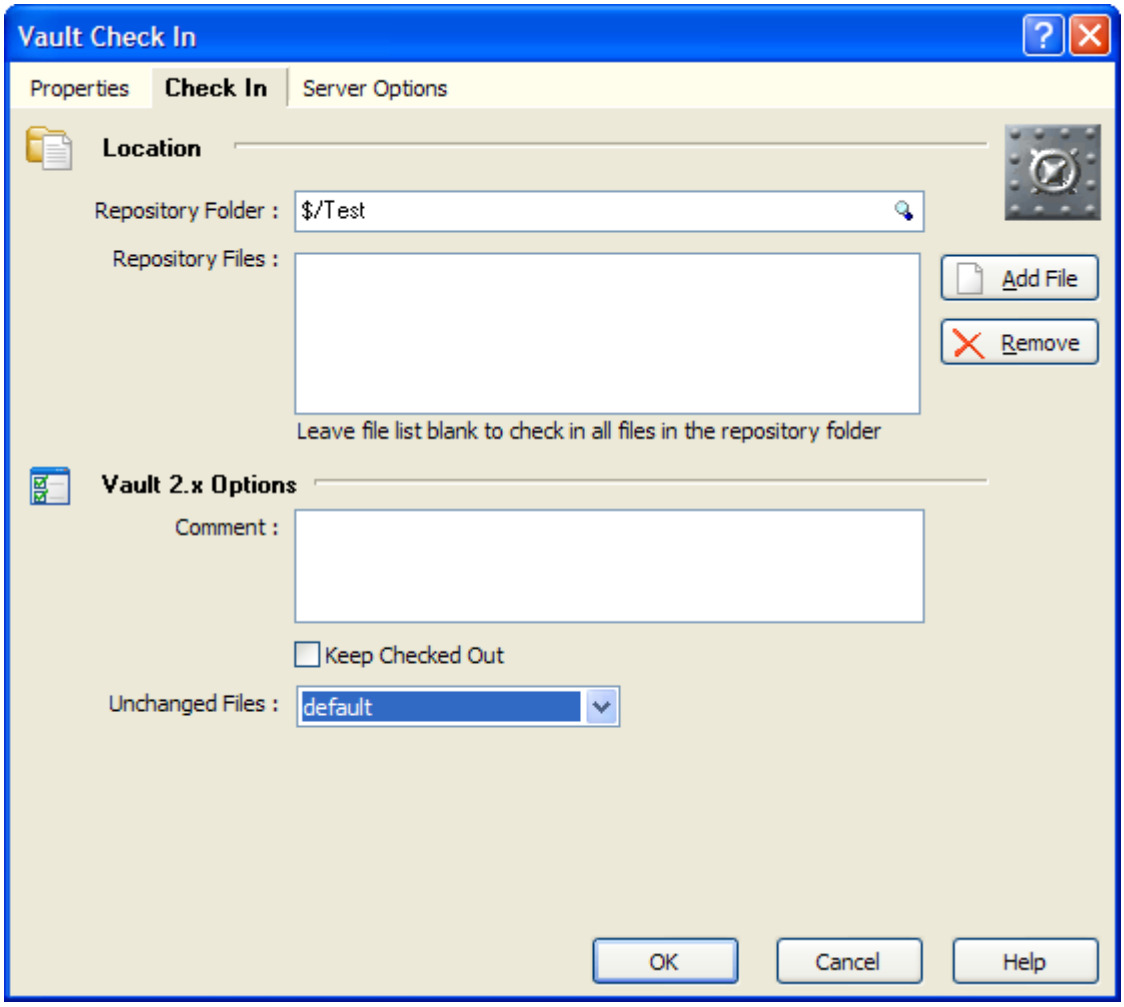
Commits all the items in the pending changeset list.



5.5.8.6 Vault Check In Action

CheckIn uses the COMMIT command, it will commit the items in the pending changeset list specified by files(s).

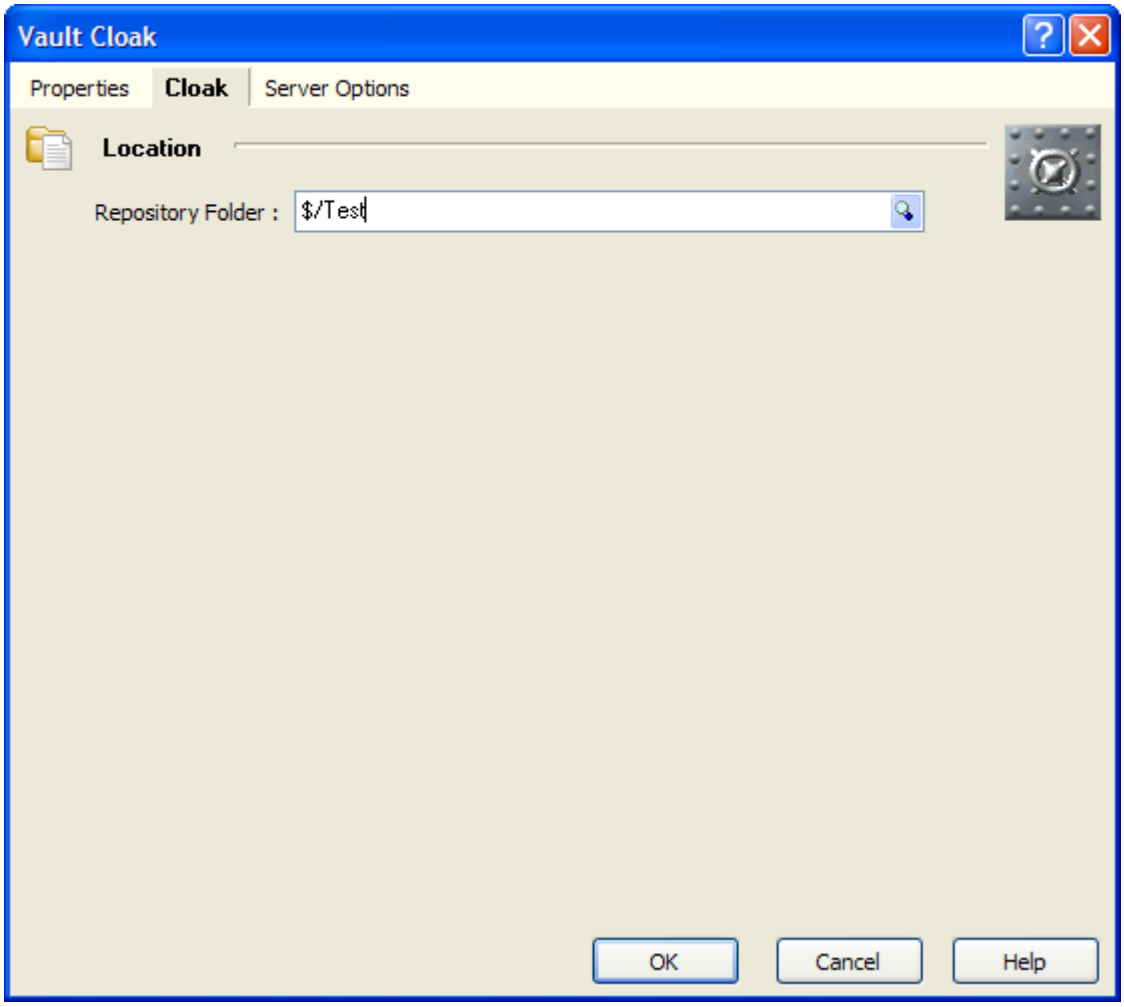
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.7 Vault Cloak Action

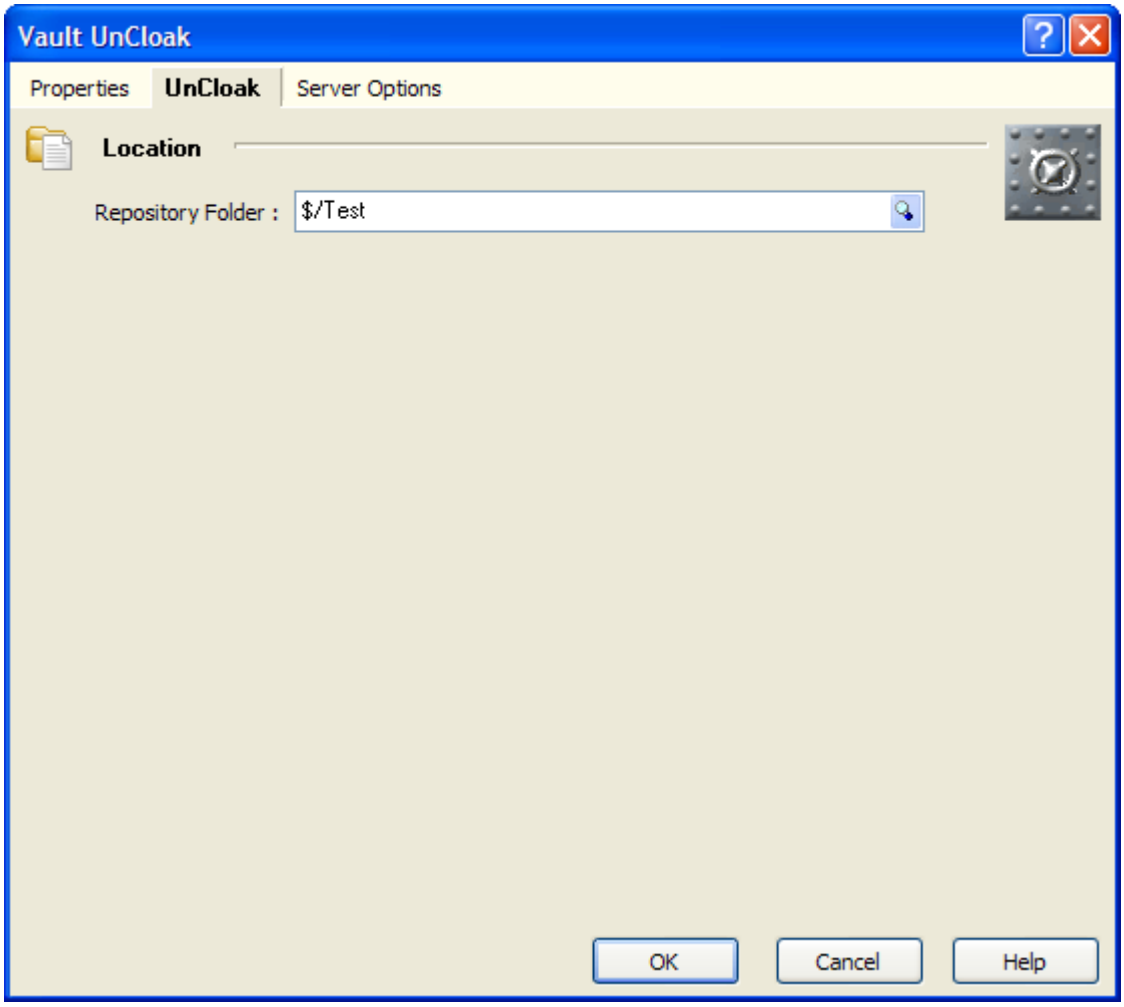
The Cloak action will cloak a repository folder.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.8 Vault UnCloak Action

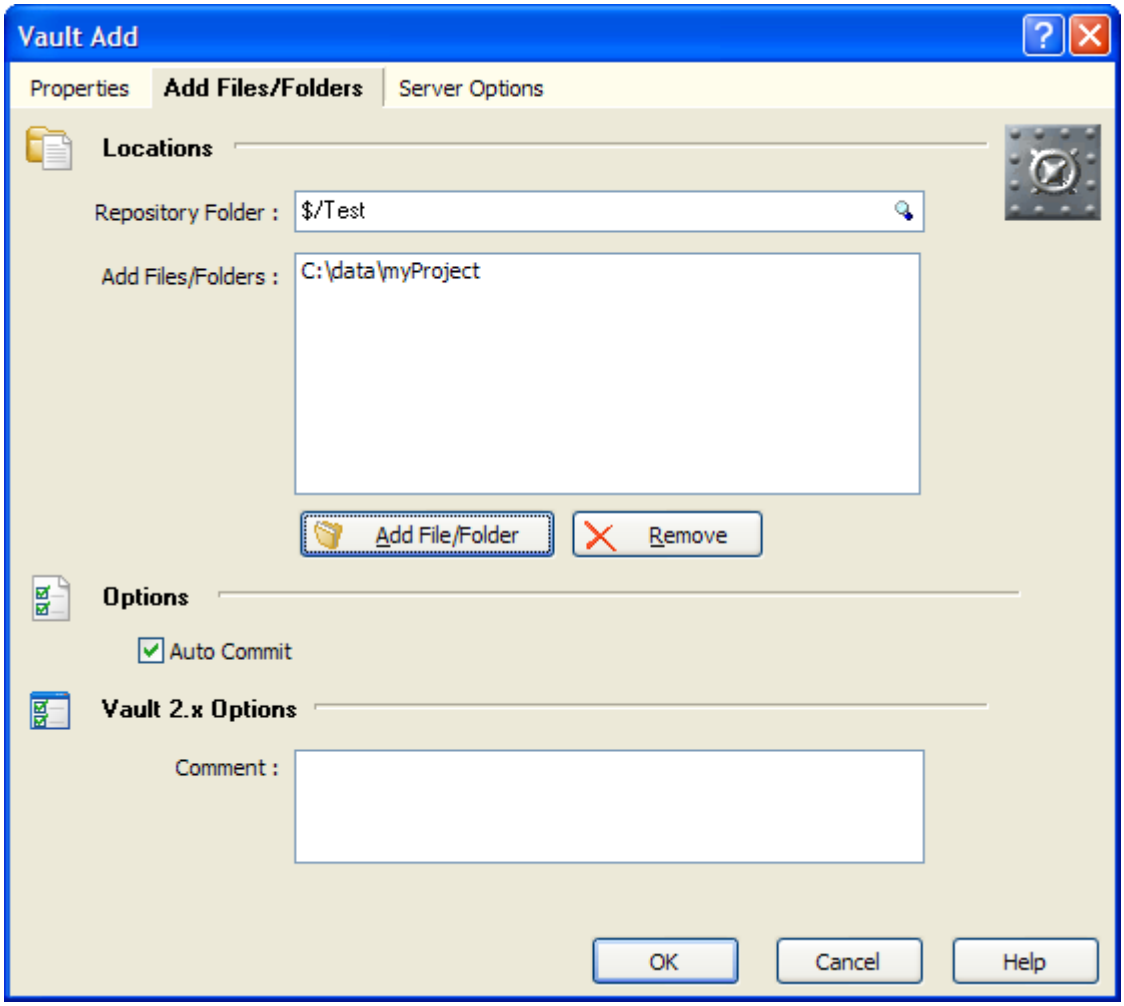
The UnCloak action will uncloak a repository folder.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.9 Vault Add Action

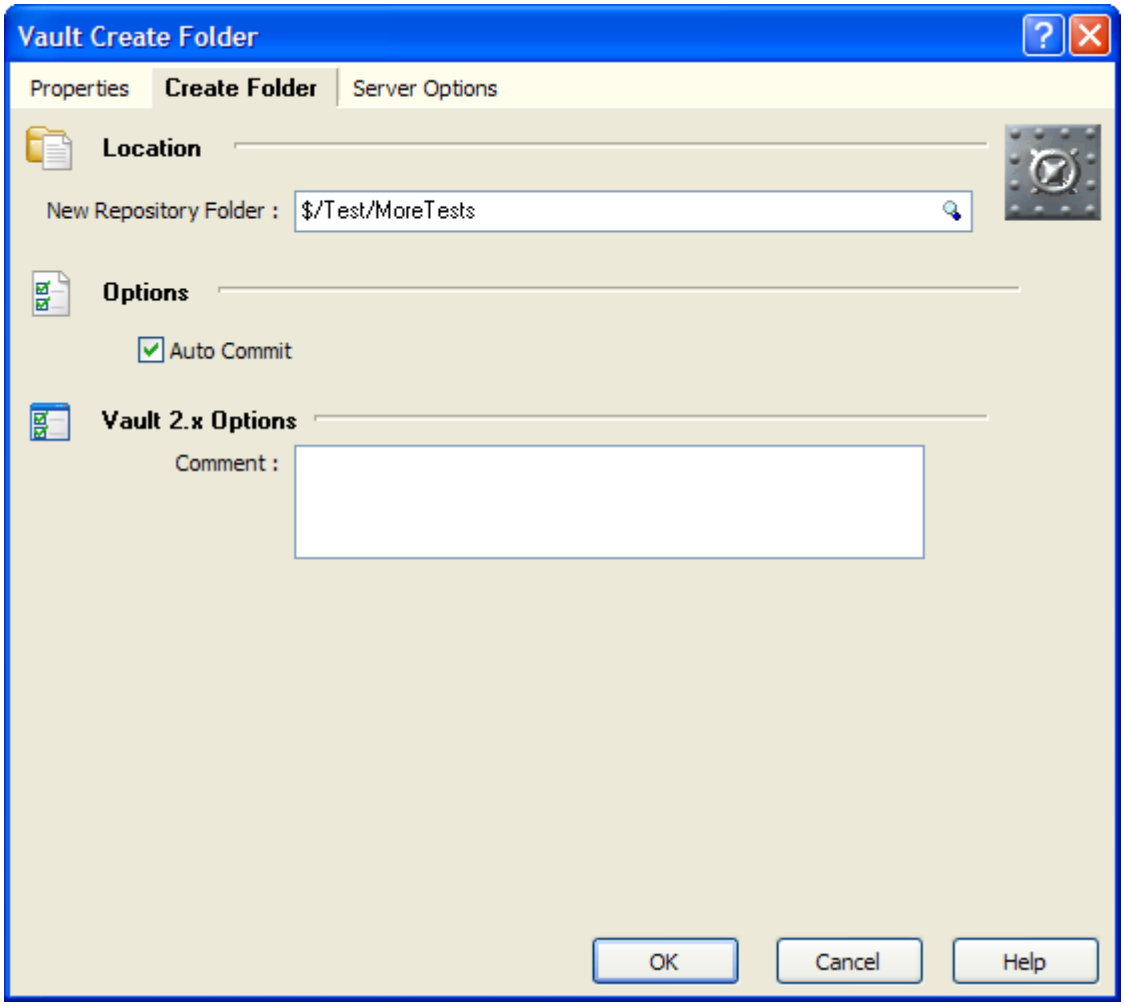
The Vault Add action will add the specified files and directories into the selected repository folder.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.10 Vault Create Folder Action

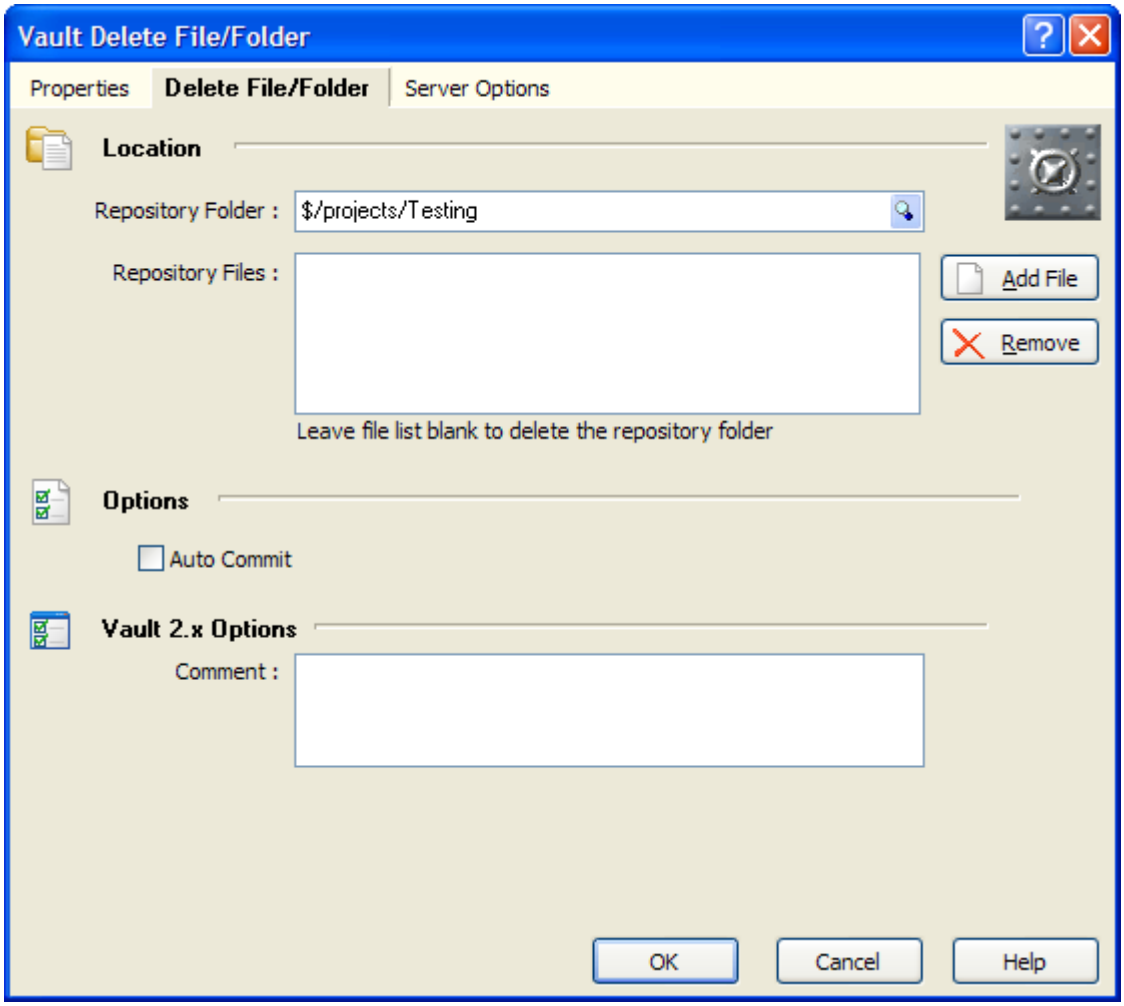
The Vault Create Folder action will create a new vault folder at the specified location in the repository



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.11 Vault Delete File/Folder Action

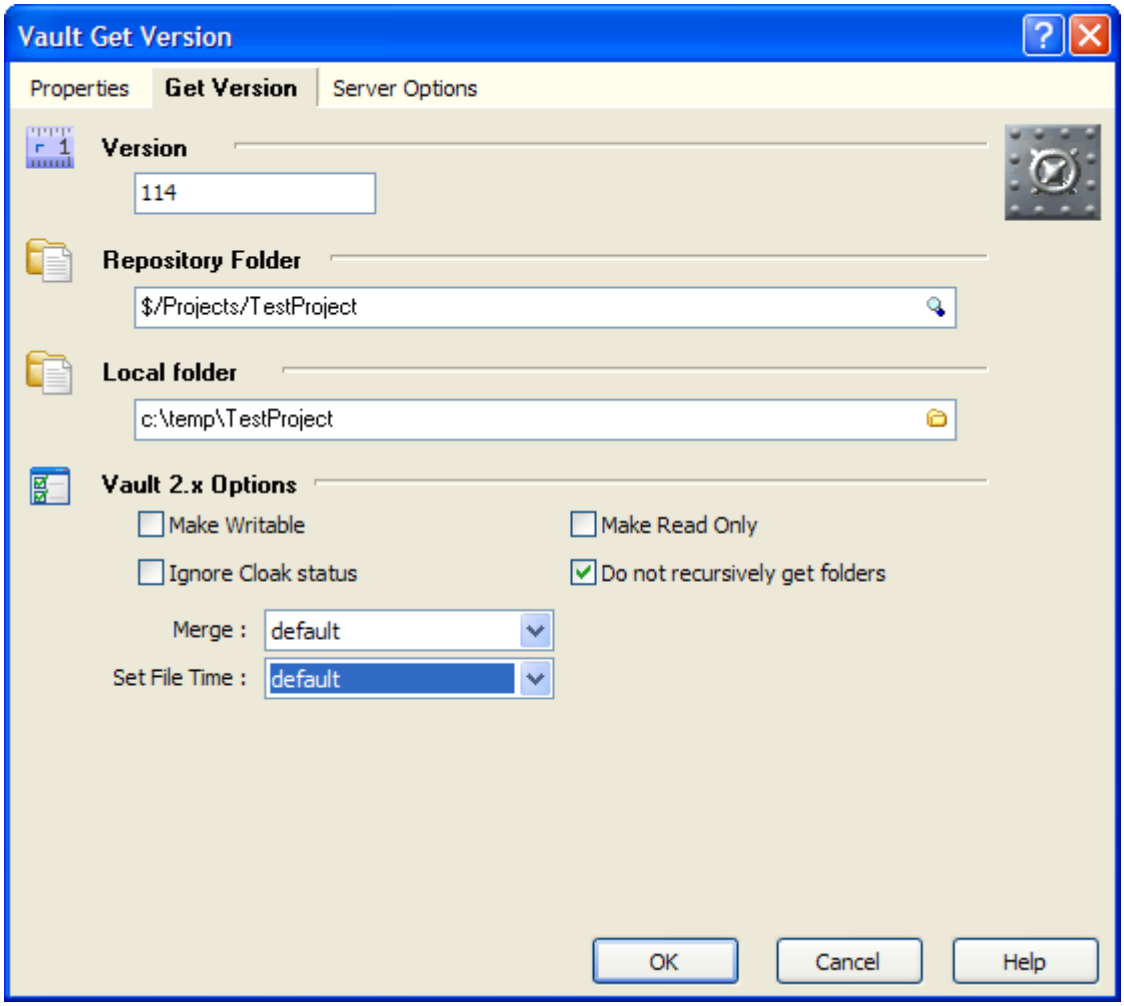
The Vault delete file/folder action will delete the specified files in the repository folder, or it will delete a repository folder if no files have been specified.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.12 Vault Get Version Action

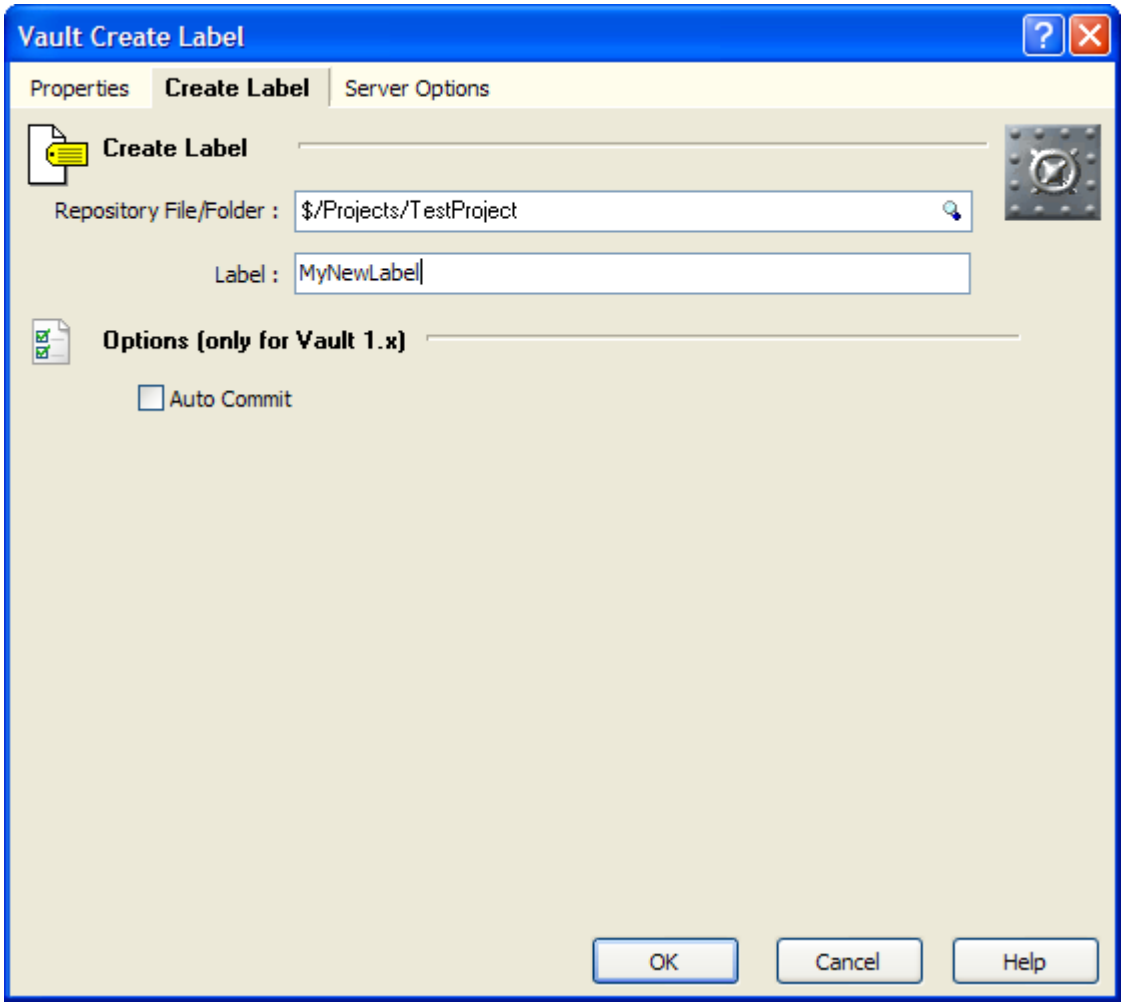
The Vault Get Version action will get files from the vault repository to the specified local folder at a specific version number.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.13 Vault Create Label Action

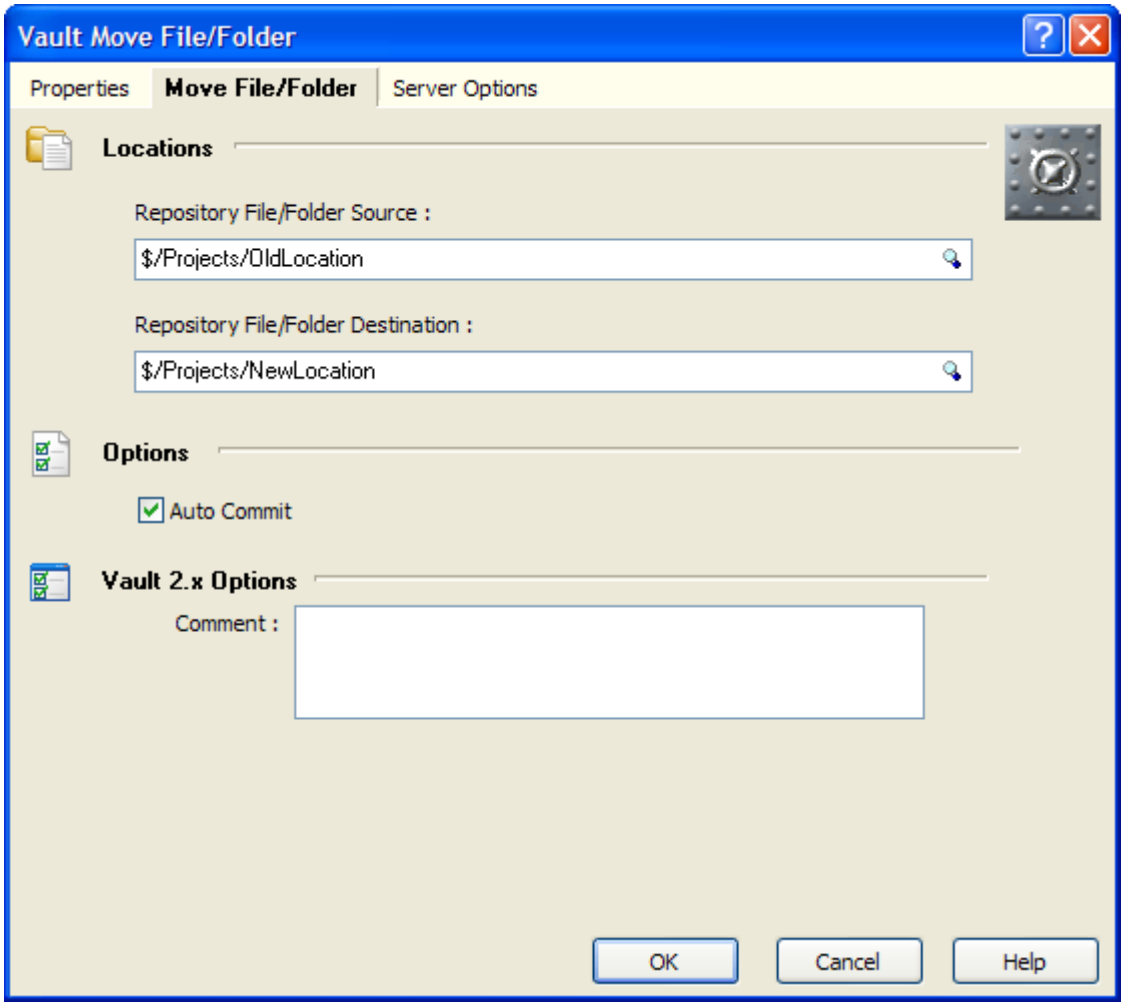
The Vault Create Label action will create a label at the specified repository file/folder.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.14 Vault Move File/Folder Action

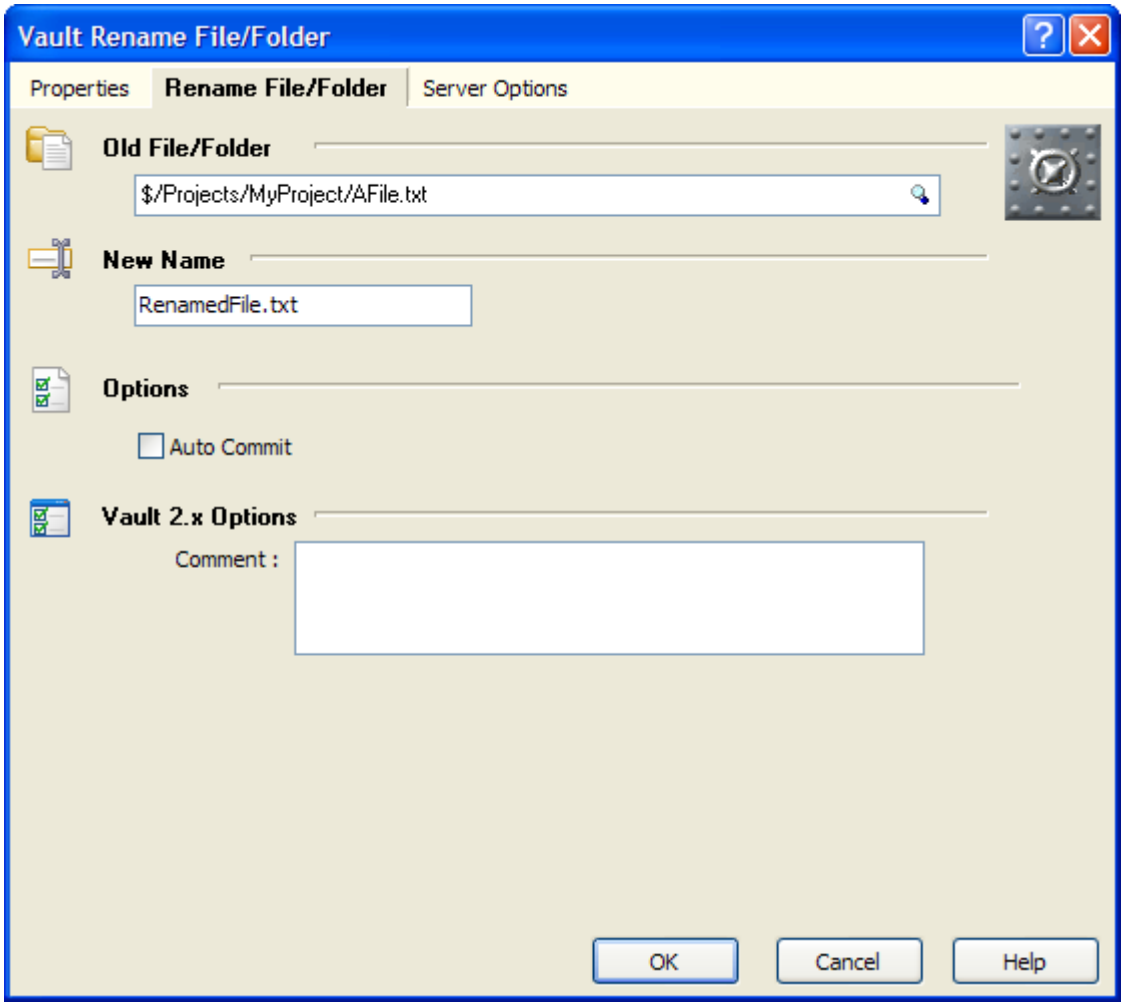
The Vault Move File/Folder action allows you move a file or folder to a new repository path.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.15 Vault Rename File/Folder Action

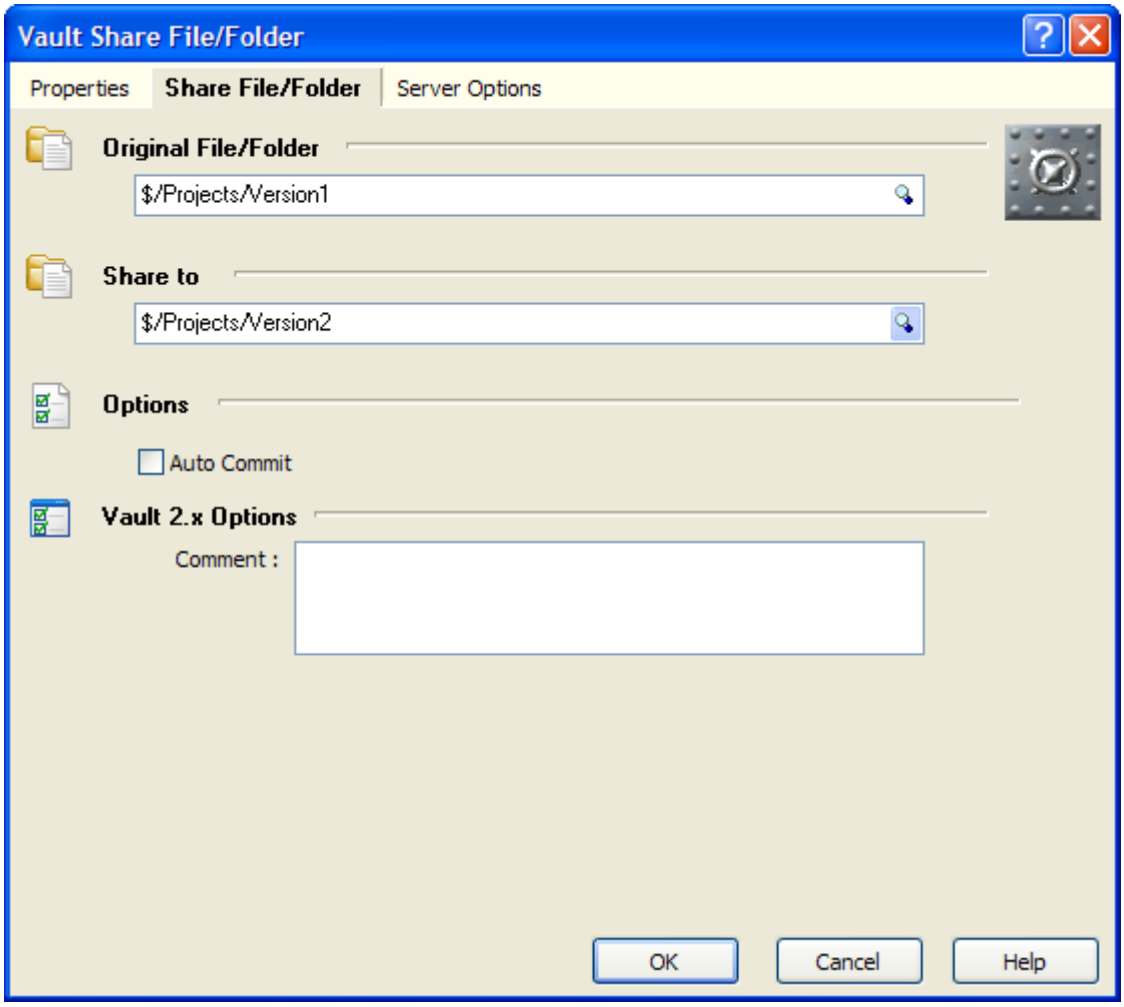
The vault rename file/folder action allows you to rename a file or folder in the vault repository.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.16 Vault Share File/Folder Action

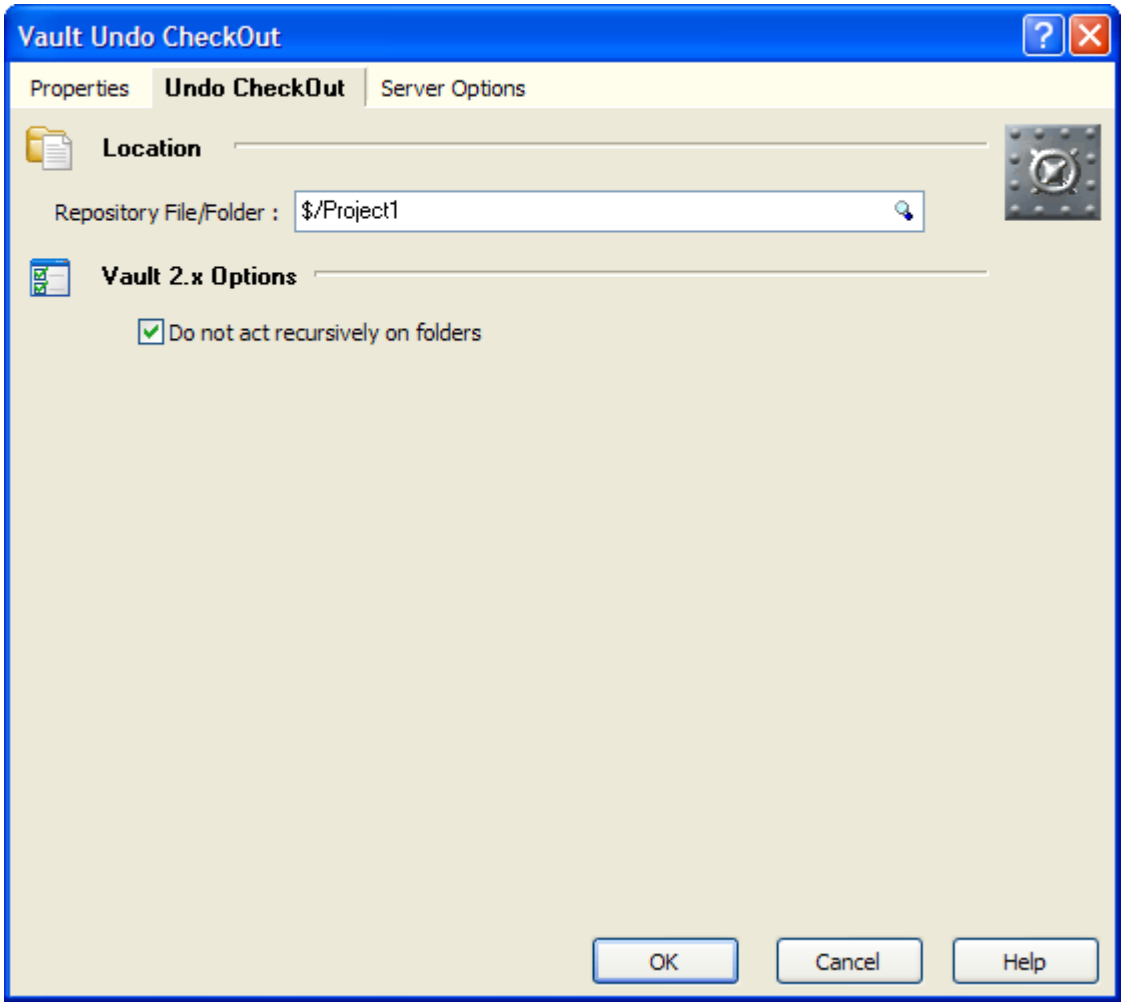
Share a file/folder to a new location in the vault repository.



5.5.8.17 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.

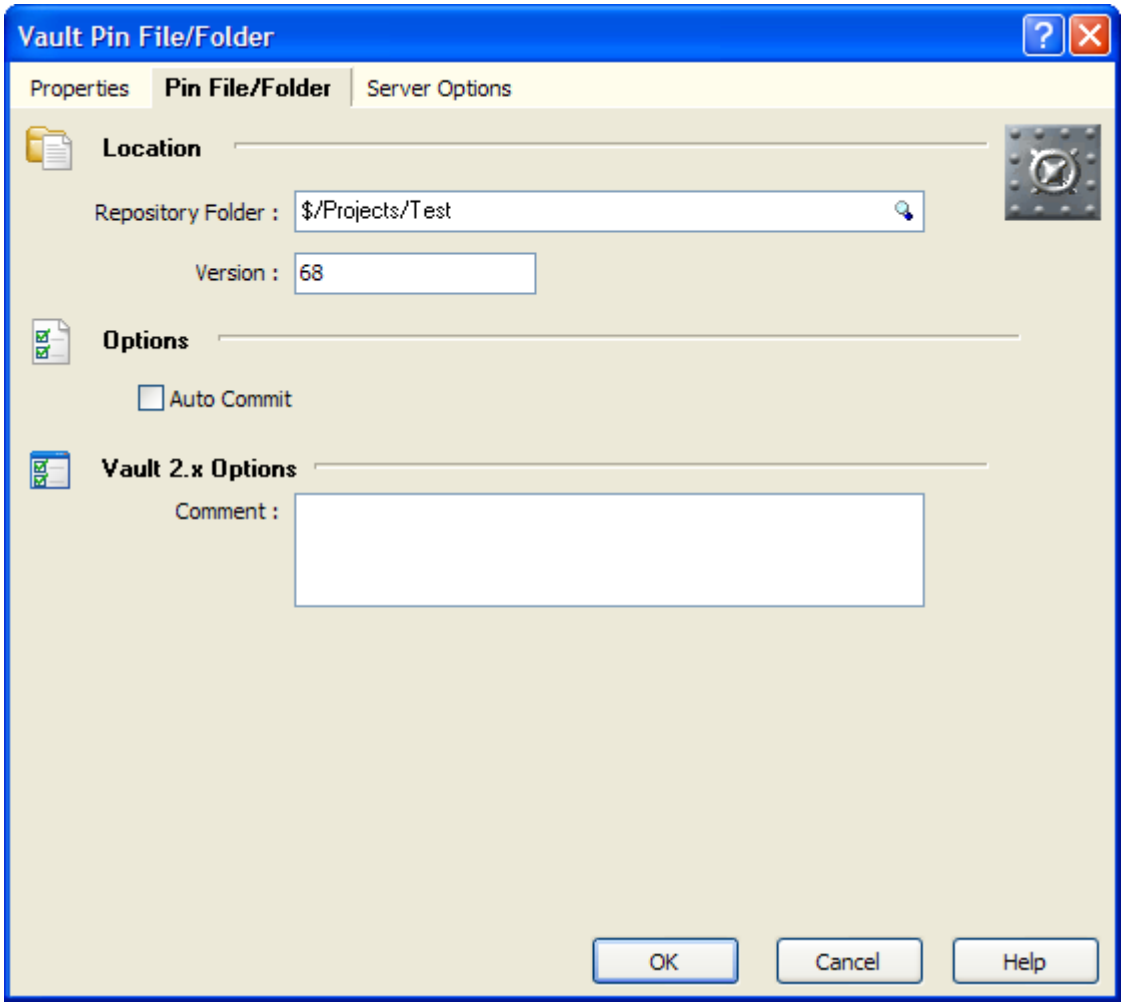
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.18 Vault Pin File/Folder Action

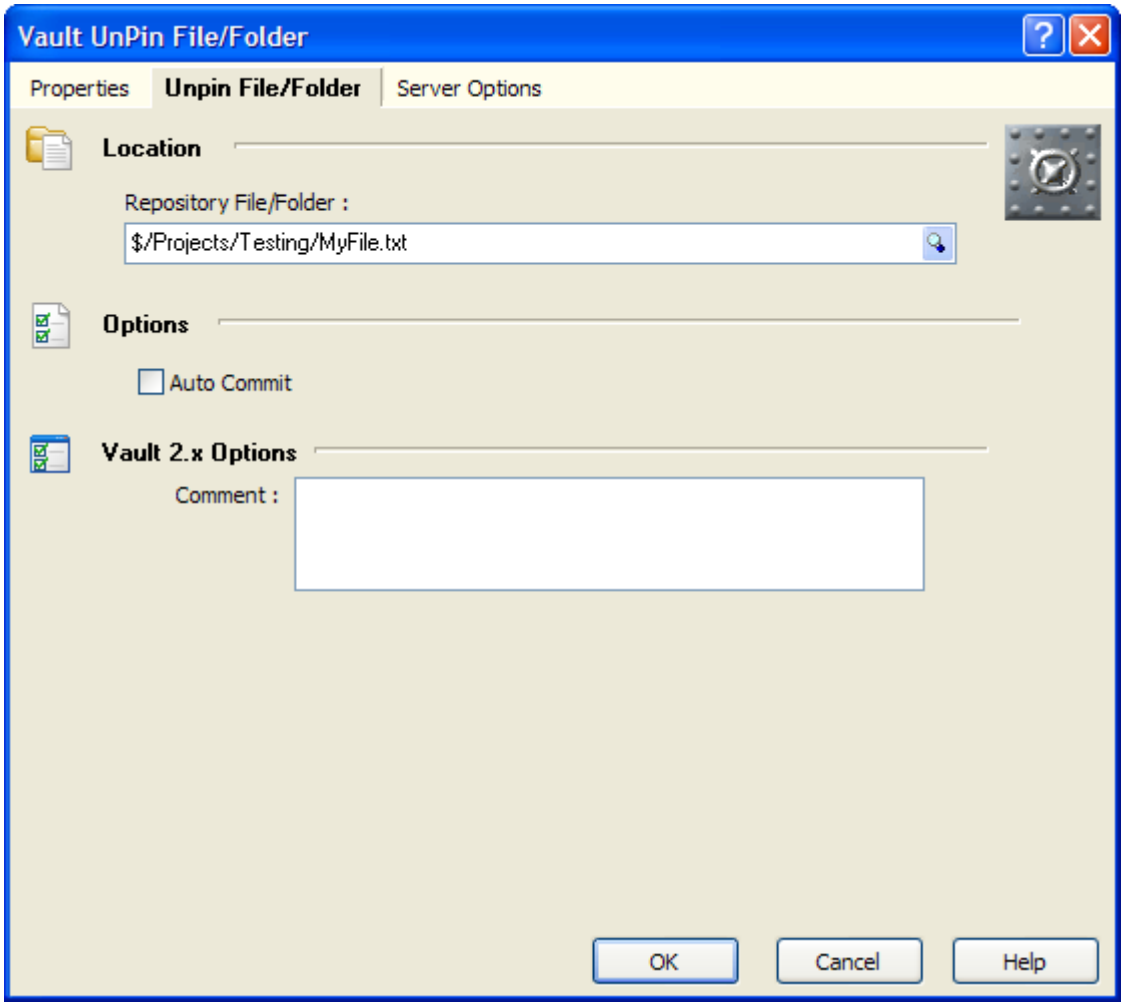
The vault Pin action allows you to Pin the specified file or folder in the repository to a particular version



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.19 Vault UnPin File/Folder Action

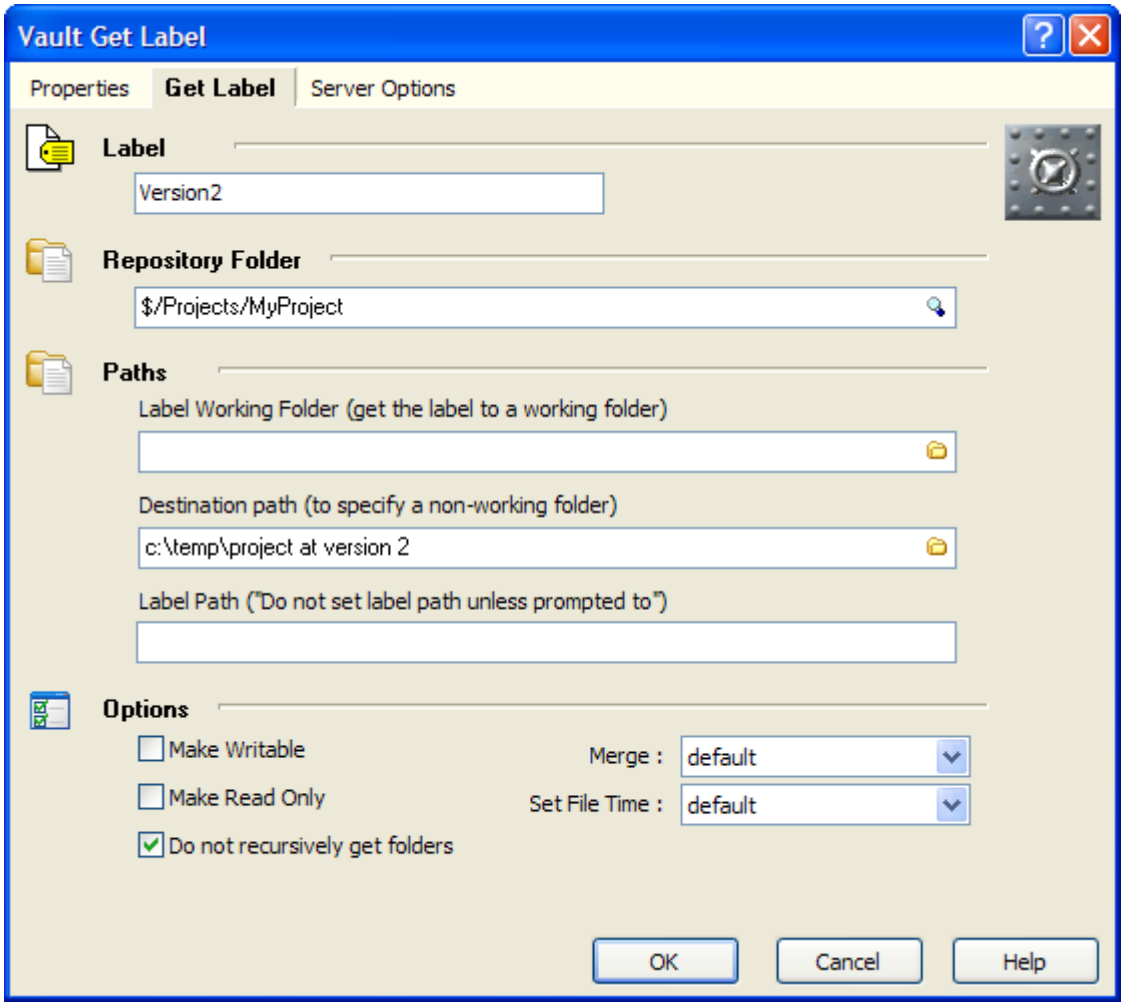
The Vault UnPin action allows you remove any Pins on the specified file or folder in the repository



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.20 Vault Get Label

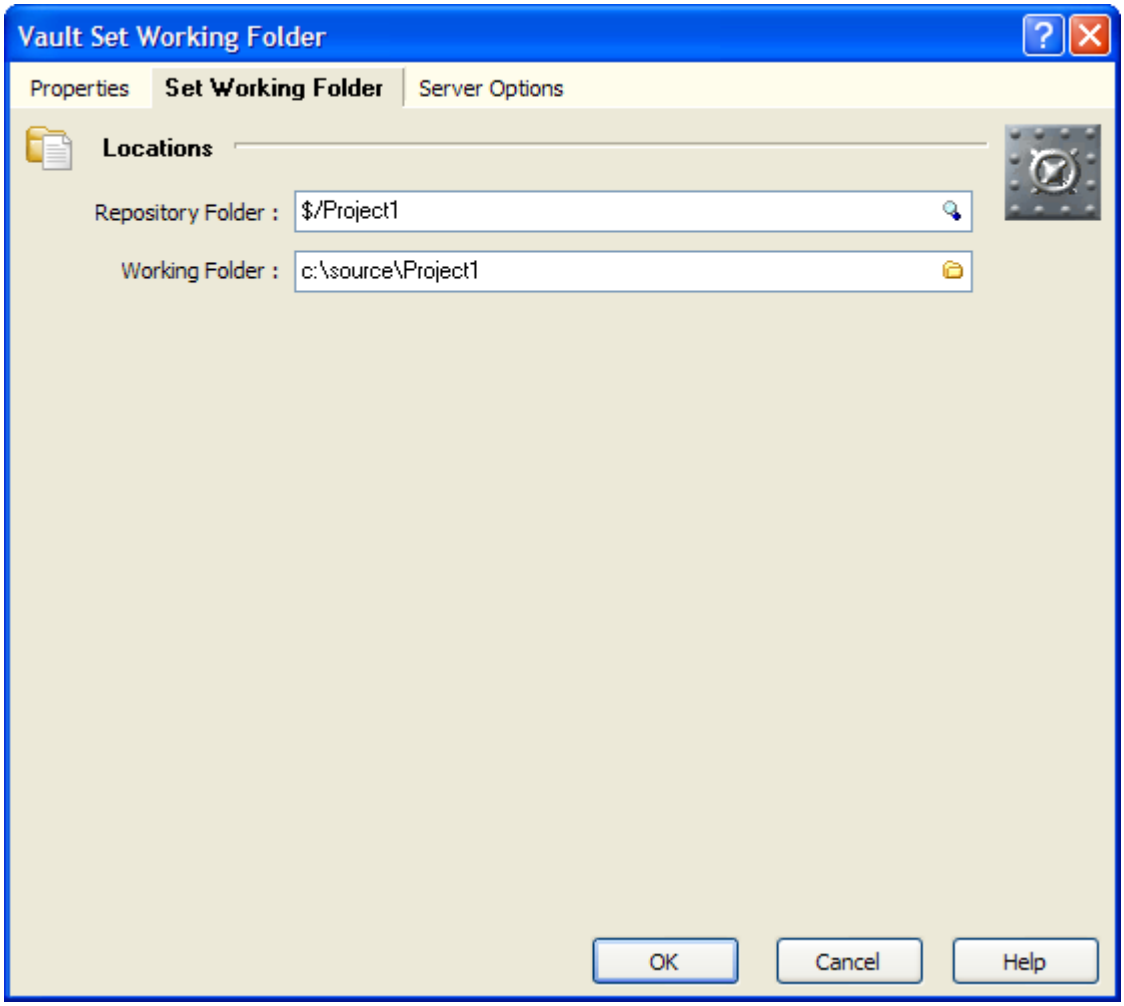
The Vault Get Label action allows you to get files from the repository at a particular label.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.21 Vault Set Working Folder

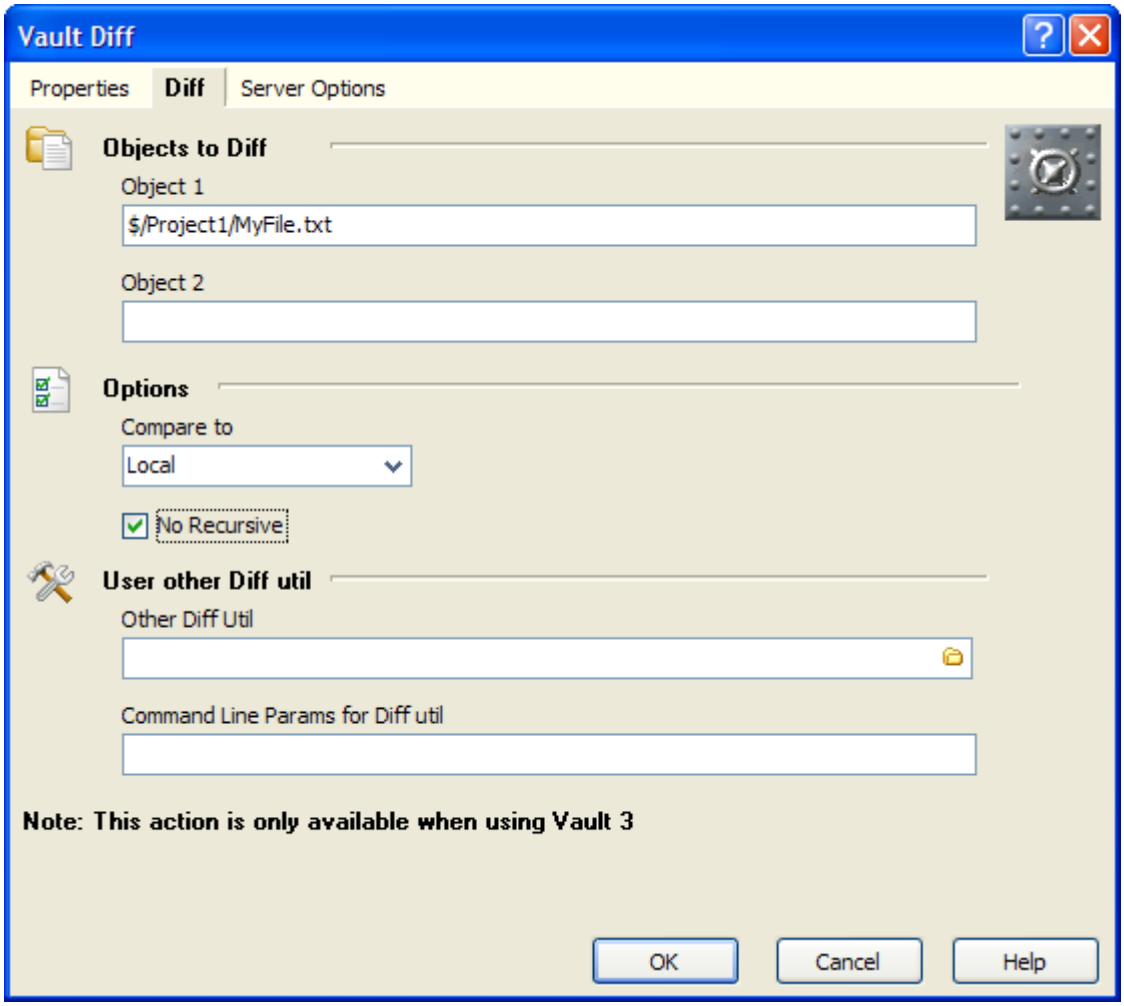
The vault set working folder action will set the working folder of a specified repository folder to a different location.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.22 Vault Diff

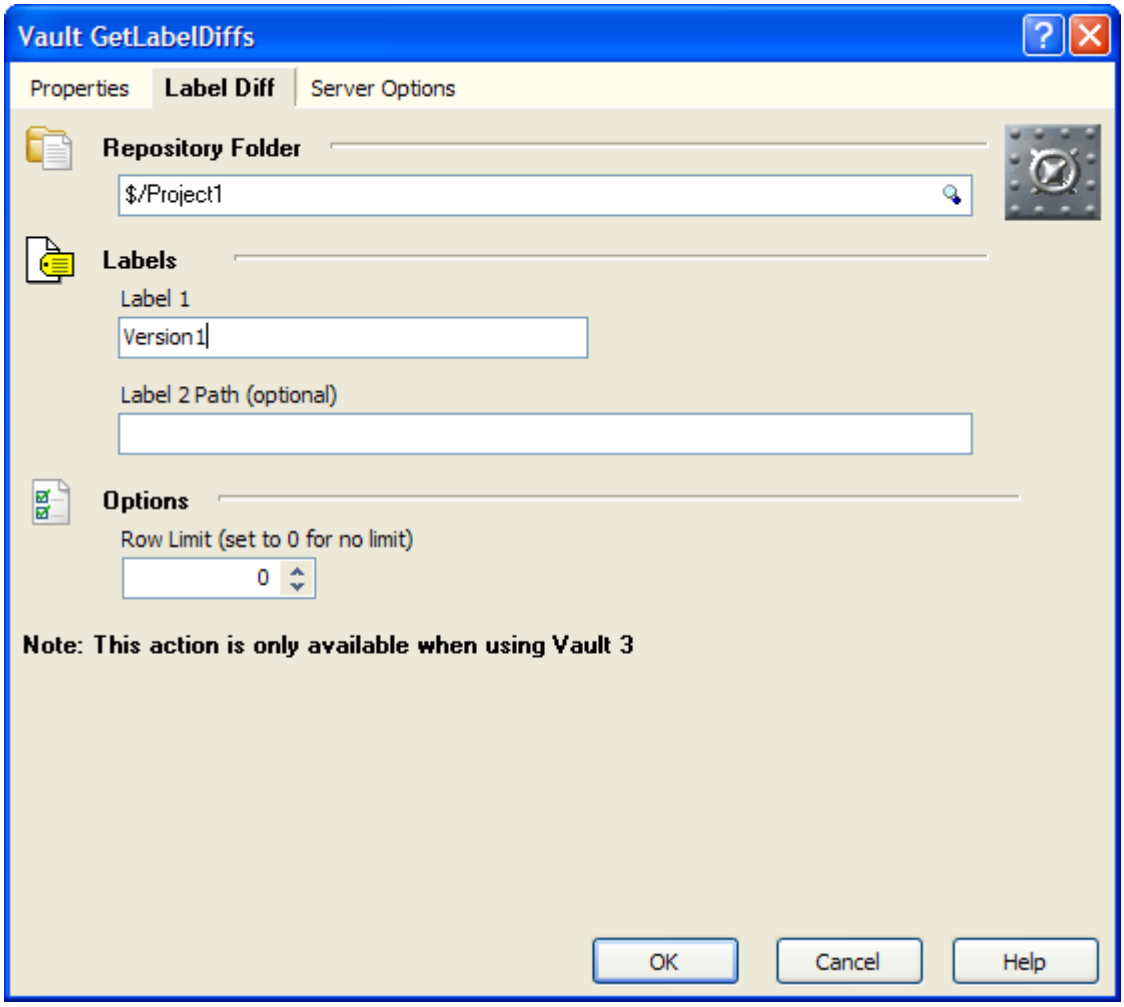
The vault Diff action allows you to compare objects using various methods within and external to the vault repository.



© 2001-2006 VSoft Technologies Pty Ltd

5.5.8.23 Vault GetLabelDiffs

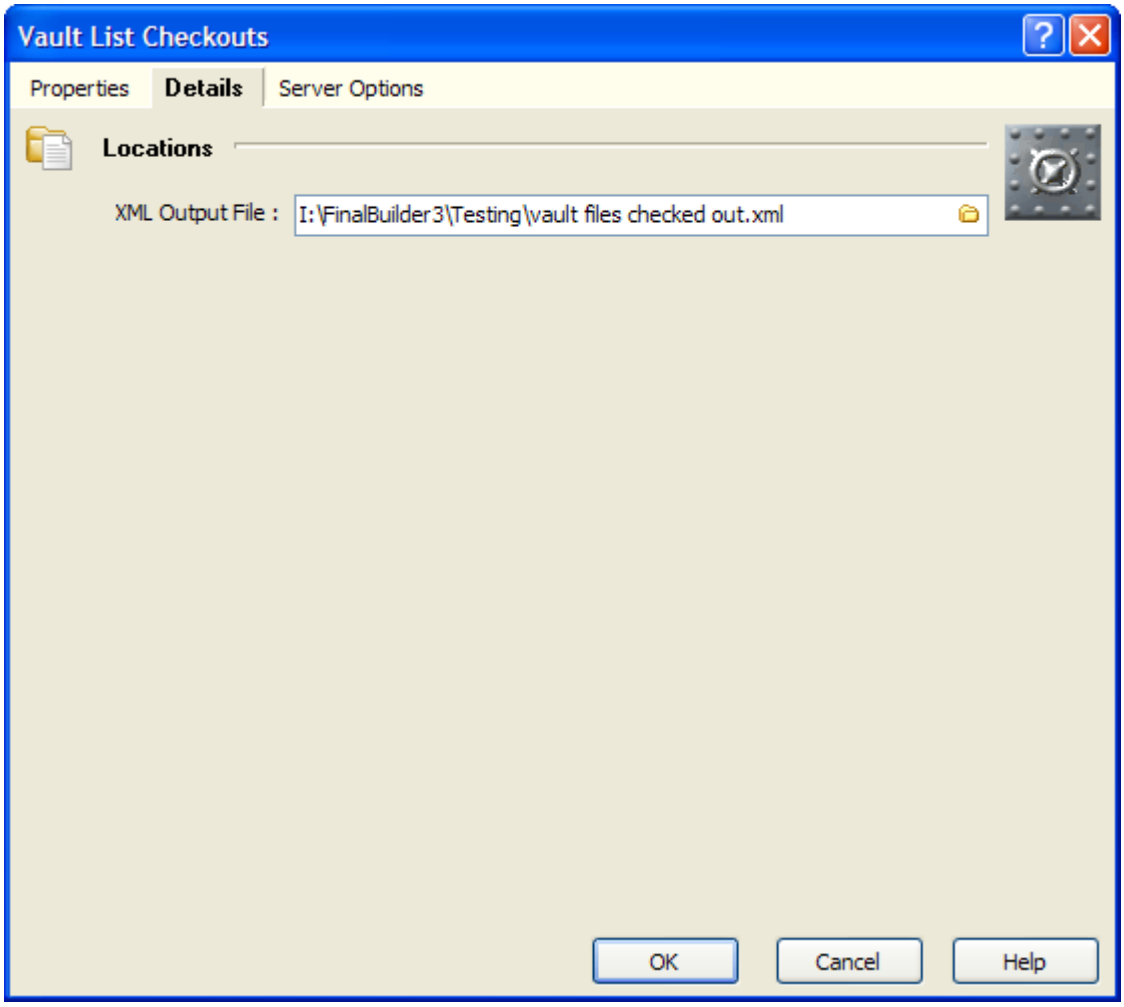
Executes the GetLabelDiffs in Vault 3.0



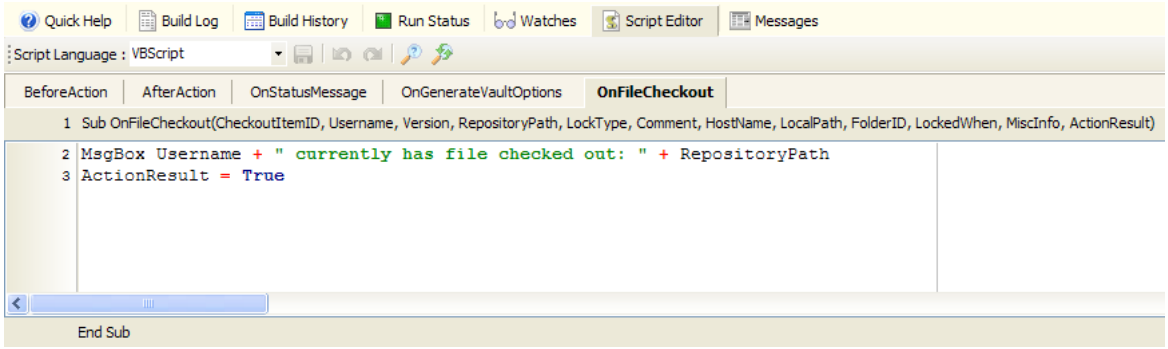
5.5.8.24 Vault ListCheckouts

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.

© 2001-2006 VSoft Technologies Pty Ltd



And to process each file checked out by each user, you can add code in the OnFileCheckout script event:

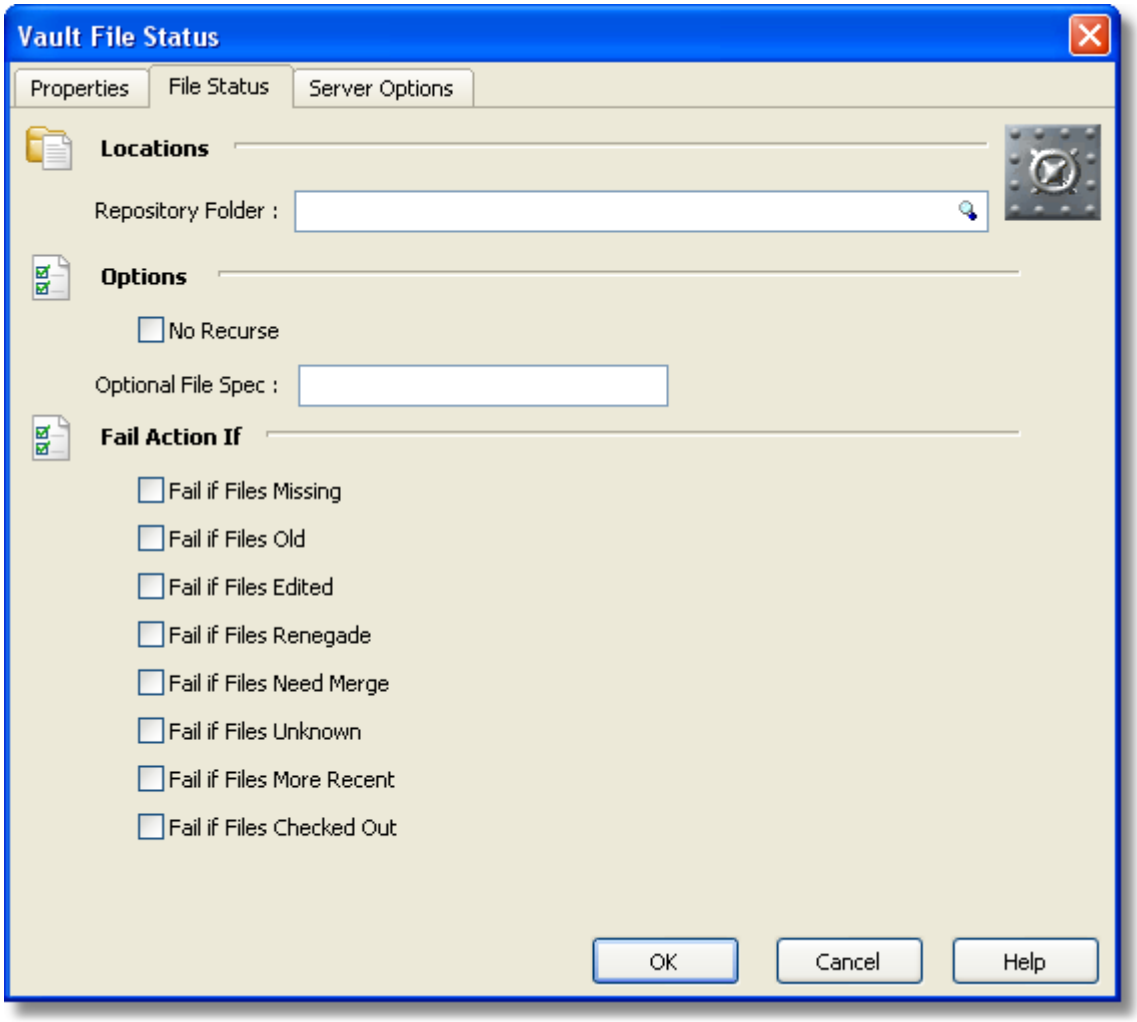


Setting the ActionResult to False will cause the action to fail after it has processed all of the files.

5.5.8.25 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.

© 2001-2006 VSoft Technologies Pty Ltd



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

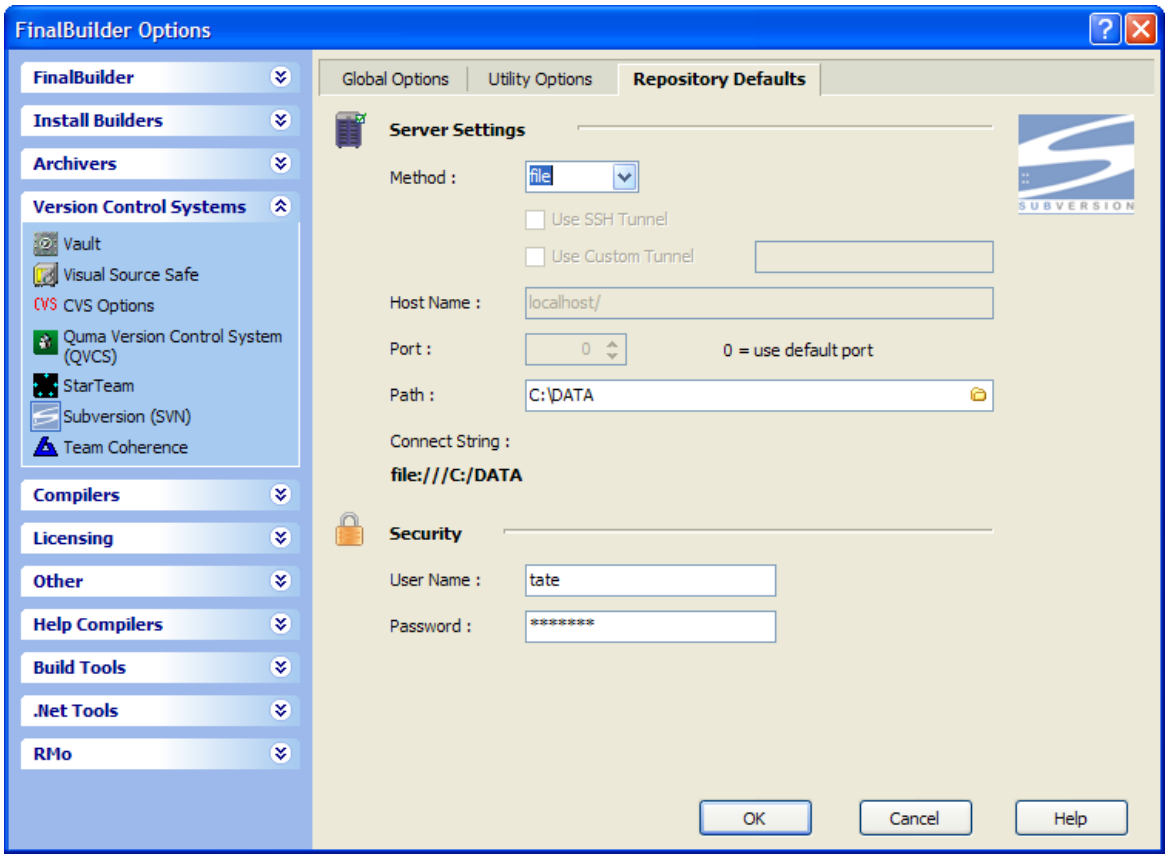
Set which file conditions will cause the action to fail.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.9 Subversion Actions

The Subversion actions support the Subversion version control system.

For more information about Subversion, see the homepage at http://subversion.tigris.org/



Before you start using the Subversion actions in FinalBuilder, you should set up the global options, utility options and repository defaults in the Tools->Options dialog:

Global Options:

"*Subversion binaries location*"

Specify the path to the 'bin' directory of the Subversion installation. FinalBuilder will use this directory to run svn.exe.

"*Default Work Folder*"

Optional. Specify a default local working folder (mapped to a repository) for subversion actions.

"*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.

© 2001-2006 VSoft Technologies Pty Ltd

"*Include SVN Command Line in FinalBuilder Log*"

If this option is set, FinalBuilder will log the full command line to svn.exe each time a Subversion action runs, allowing you to see which options are enabled.

Utility Options:

"*Configuration Directory*"

Optional. Allows you to specify a non-default directory for Subversion configuration information. Equivalent to the "--config-dir" command line switch. You can specify individual configuration directories for most actions, as well.

"*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. Equivalent to the "--file" switch.

"Diff3/Editor Command"

Optional. Allows you to specify a custom diff3 or editor application for merging or text file editing. (Equivalent of the --diff3-cmd and --editor-cmd switches.)

Repository Defaults:

Specify the details of how to connect to the Subversion server:

"*Method*"

Specify how you would like to connect to the repository. Connections can be made by file, via HTTP or via the SVN protocol.

"*Host Name*"

The server hosting the repository. Used when making network connections.

"*Path*"

The local path, or the server path, to the repository.

"*Connect String*"

As you fill in the options, FinalBuilder gives you a preview of the complete identifier as it will be passed to svn.

© 2001-2006 VSoft Technologies Pty Ltd

"*Security*"

Specify the user name/password with which to connect to the repository.

5.5.9.1 Subversion Add

The Add command will put files and directories under version control, scheduling them for addition to repository. They will be added in next commit.

5.5.9.2 Subversion Checkout

The Checkout command will check out a working copy from a repository.

5.5.9.3 Subversion Cleanup

The Cleanup command will recursively clean up the working copy, removing locks, resuming unfinished operations, etc.

5.5.9.4 Subversion Copy

The Copy command will duplicate something in working copy or repository, remembering history.

5.5.9.5 Subversion Commit

The Cleanup command will send changes from your working copy to the repository.

5.5.9.6 Subversion Export

The Export command will create an unversioned copy of a tree.

5.5.9.7 Subversion Import

The Import command will commit an unversioned file or tree into the repository.

5.5.9.8 Subversion MkDir

The MkDir command will create a new directory under version control.

5.5.9.9 Subversion Update

The Update command will bring changes from the repository into the working copy.

5.5.9.10 Subversion Status

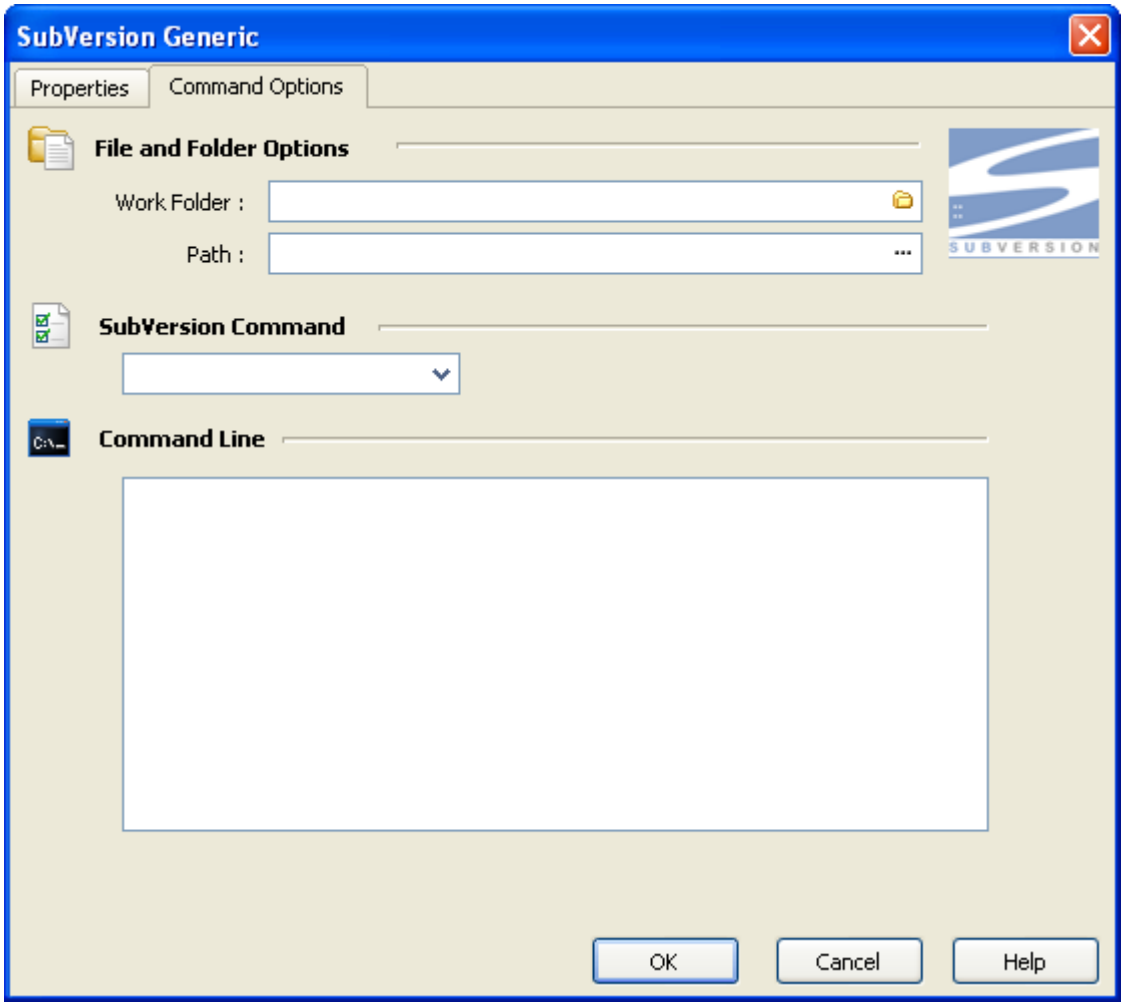
The status command gets the status of working copy files and directories.

5.5.9.11 Subversion Generic Action

The Subversion Generic Action allows you to run a Subversion command which is not otherwise supported by FinalBuilder.

See the Subversion Actions topic to see the Subversion option settings.

© 2001-2006 VSoft Technologies Pty Ltd



File & folder Options

Specify a working folder and (optionally) a file or list of files and folders for the action to operate on.

Subversion Command

Enter the command to run with svn.

Command Line

Enter any other arguments to be inserted after the command name.

The command line is built as follows:

svn *<Command> <Paths List> <Command Line>*

© 2001-2006 VSoft Technologies Pty Ltd

5.5.10 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

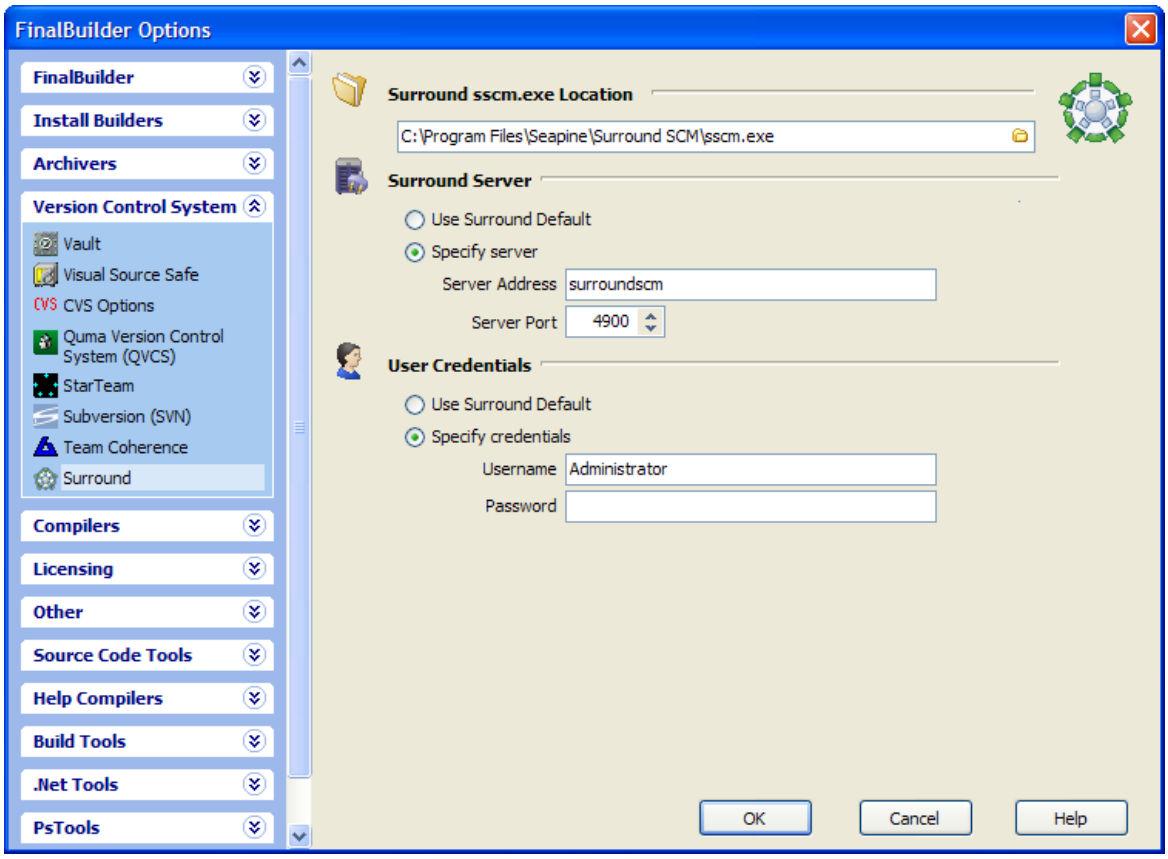
Before you start using the Surround SCM actions, it is suggested you first set up Global Surround SCM Options

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.

5.5.10.1 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 autodetect the sscm.exe location - if the Surround

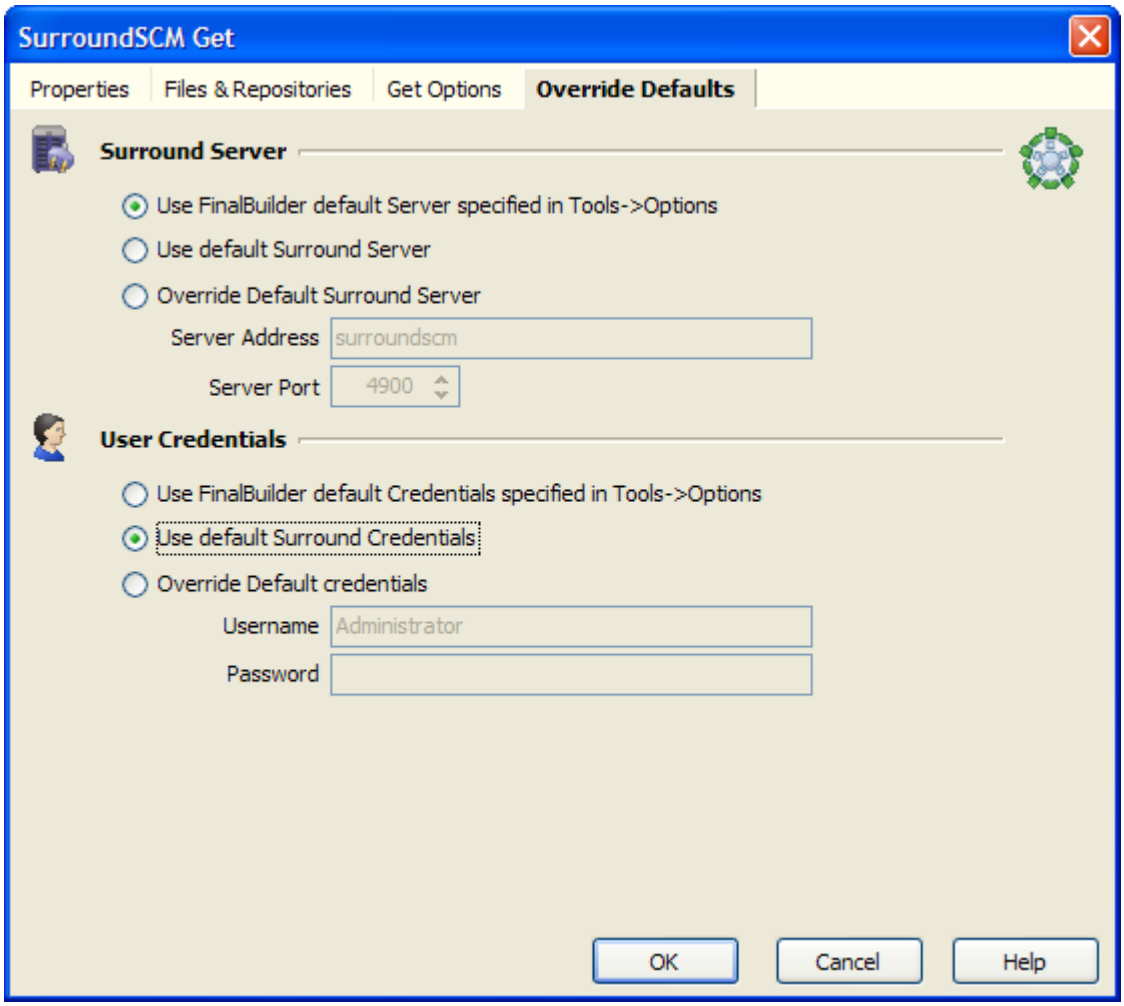
© 2001-2006 VSoft Technologies Pty Ltd

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.

5.5.10.1.1 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

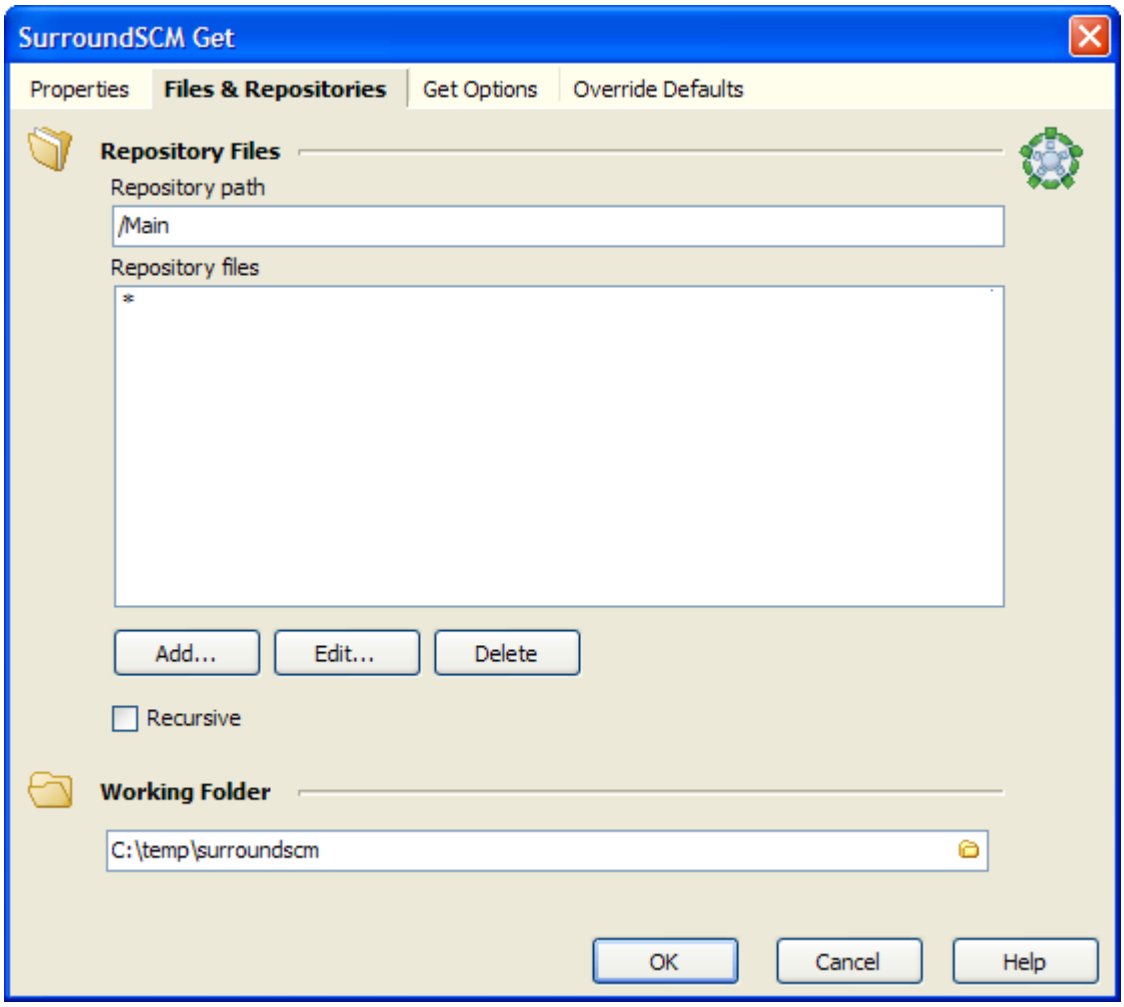
5.5.10.2 Surround SCM Get

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.

© 2001-2006 VSoft Technologies Pty Ltd

Files & Repositories Tab

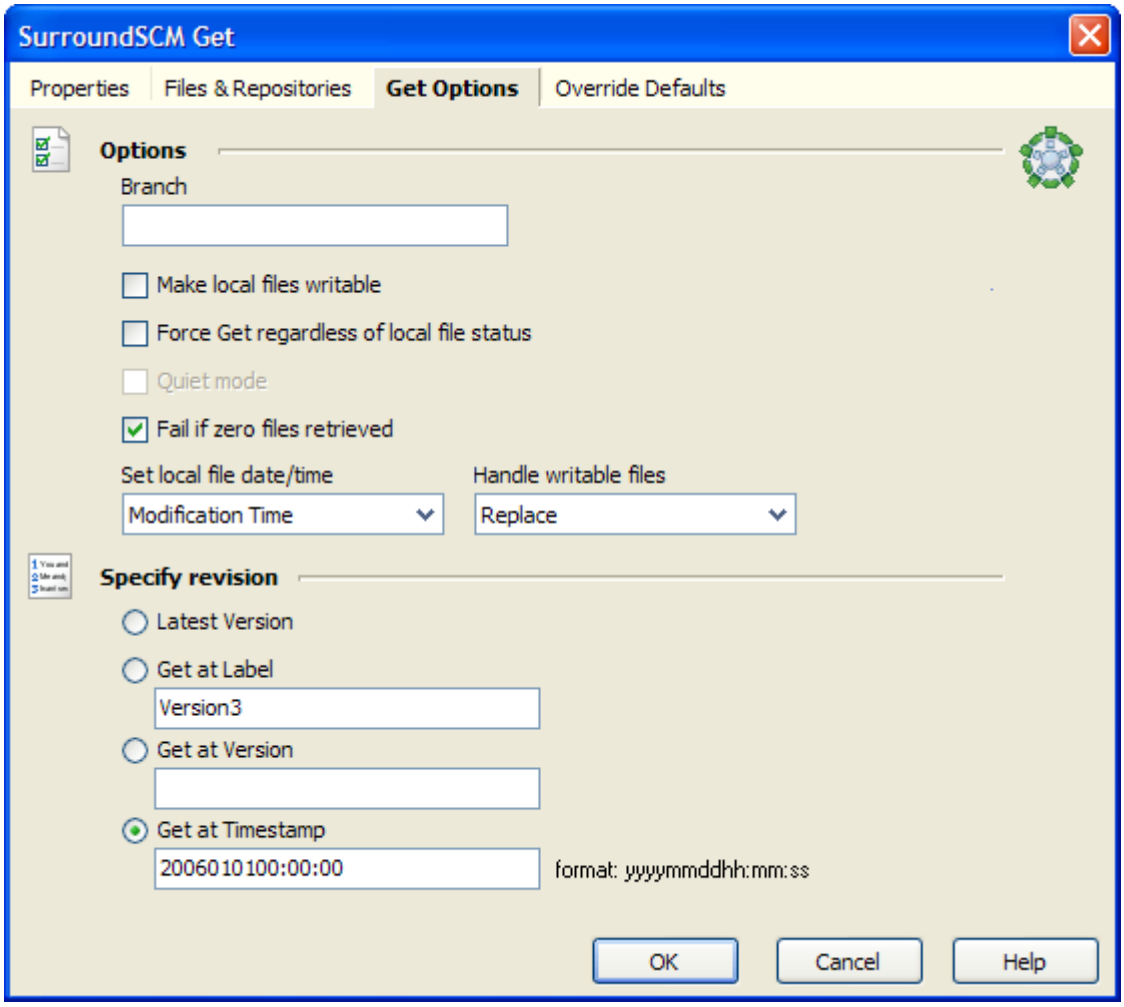
Specify the repository path and a list of files to retrieve. You can use * to specify all files.
Specify Recursive to Recursively get files and subrepositories.



Get Options Tab

Specify the Get options - all values are optional.

© 2001-2006 VSoft Technologies Pty Ltd



Tips:
If you don't specify a working folder, then Surround will use the current working folder for the repository selected.

5.5.10.3 Surround SCM CheckOut

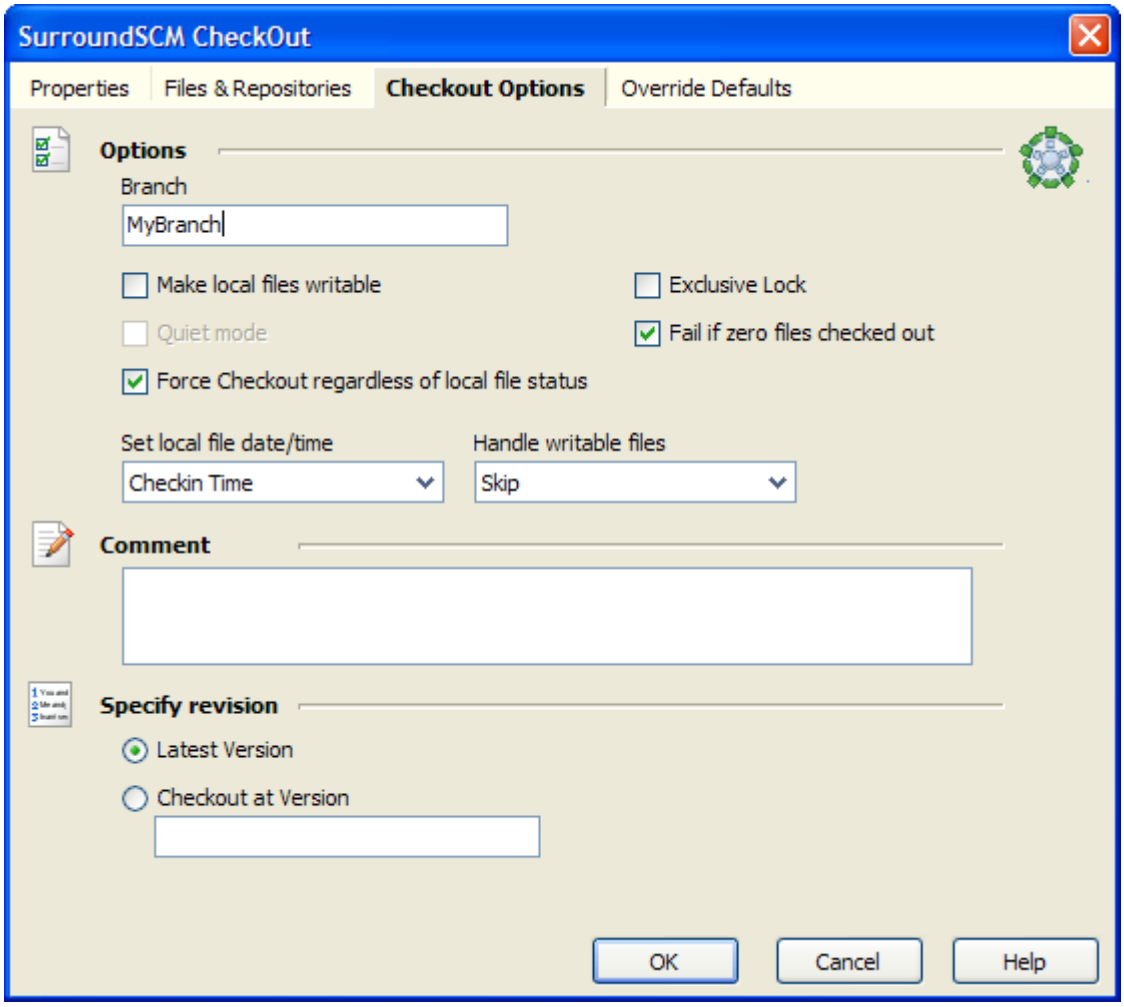
Check out files when you need to make changes. You can check out single files, multiple files, or a
repository. Surround SCM creates a read-write copy of the file in the working directory.

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.

© 2001-2006 VSoft Technologies Pty Ltd



5.5.10.4 Surround SCM CheckIn

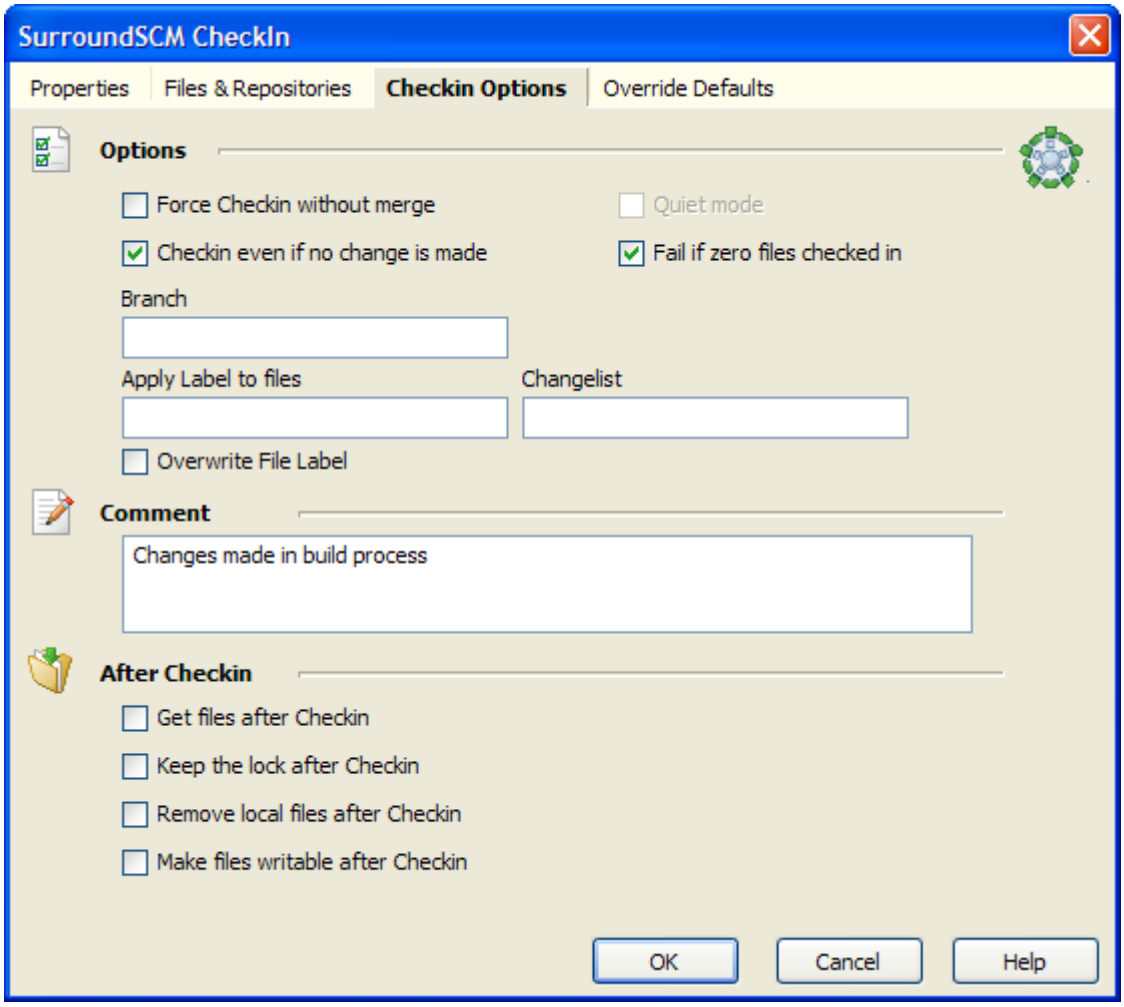
Check in updates Surround SCM files with changes, removes the lock on the files, and makes changes
available to other users.

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.5.10.5 Surround SCM Label

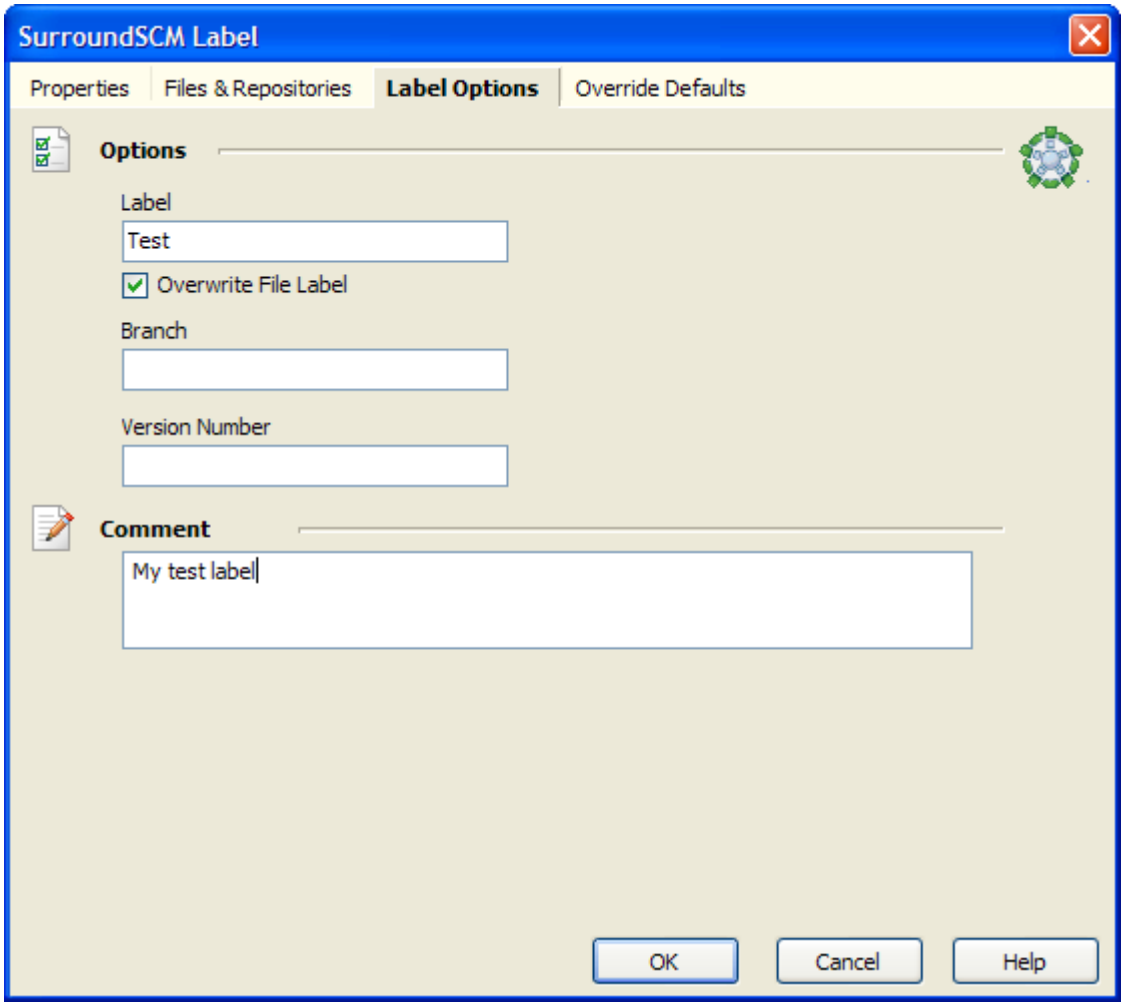
Labels provide a way to mark a specific version of a file or repository. When you create a label, a new
entry is created in the history. The file and the version number do not change.

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.

© 2001-2006 VSoft Technologies Pty Ltd



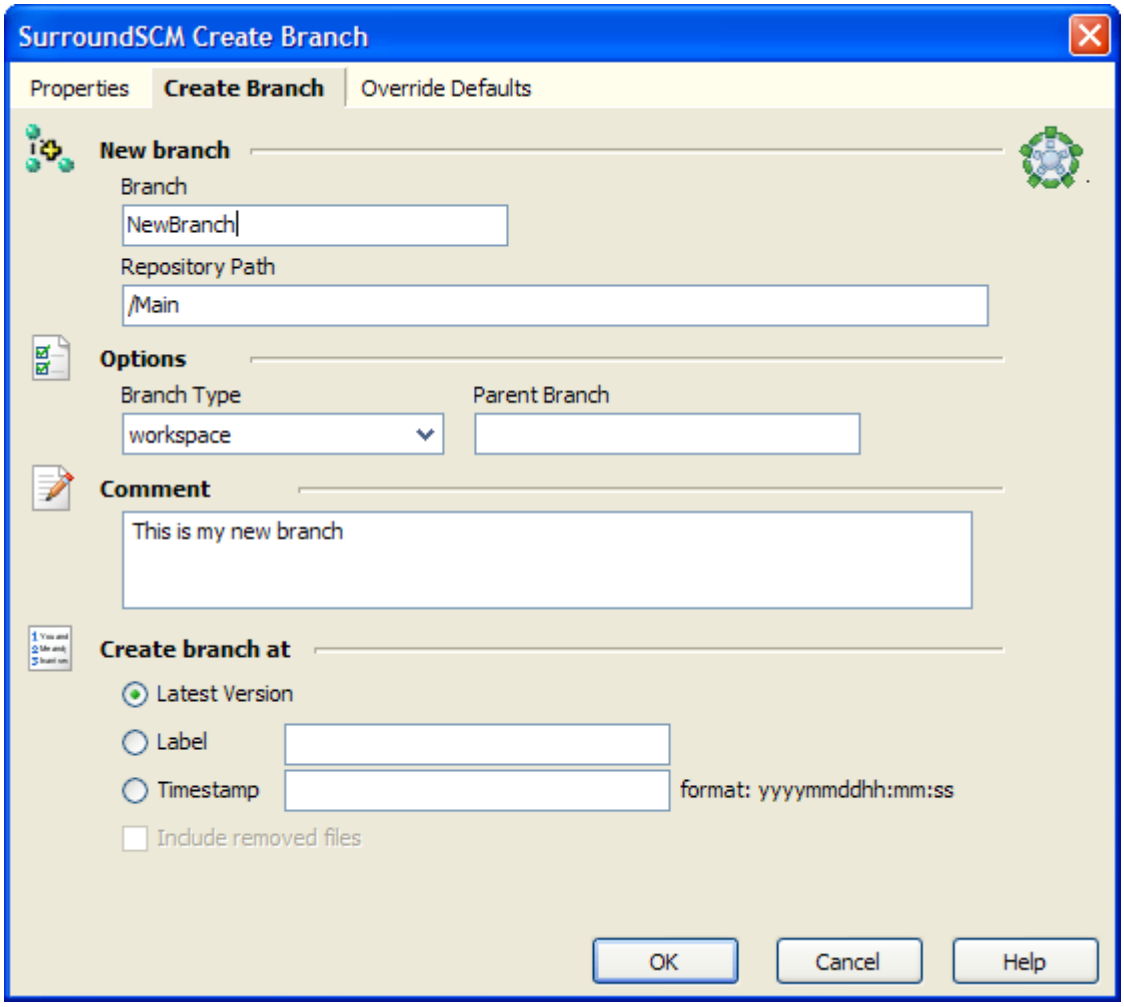
Tip: To label all files in a repository then use "/" as the repository file spec (without quotes).

5.5.10.6 Surround SCM Create Branch

The Create Branch action allows creation of a new workspace, baseline or snapshot branch in Surround SCM.

Create Branch Tab

© 2001-2006 VSoft Technologies Pty Ltd



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: yyyymmddhh:mm:ss

5.5.10.7 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.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.10.8 Surround SCM Unfreeze Branch

Unlock a frozen branch.

5.5.10.9 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.

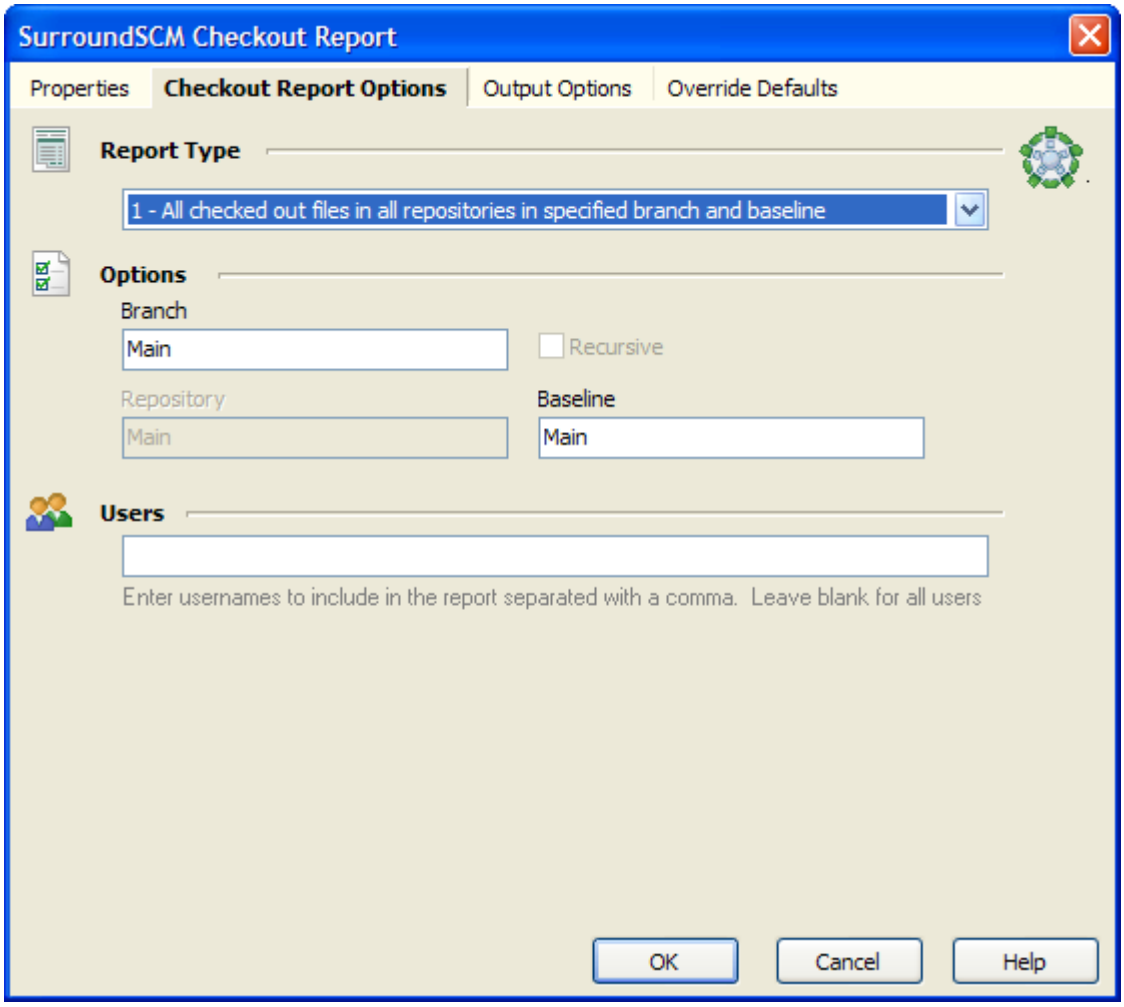
Checkout Report Options

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.

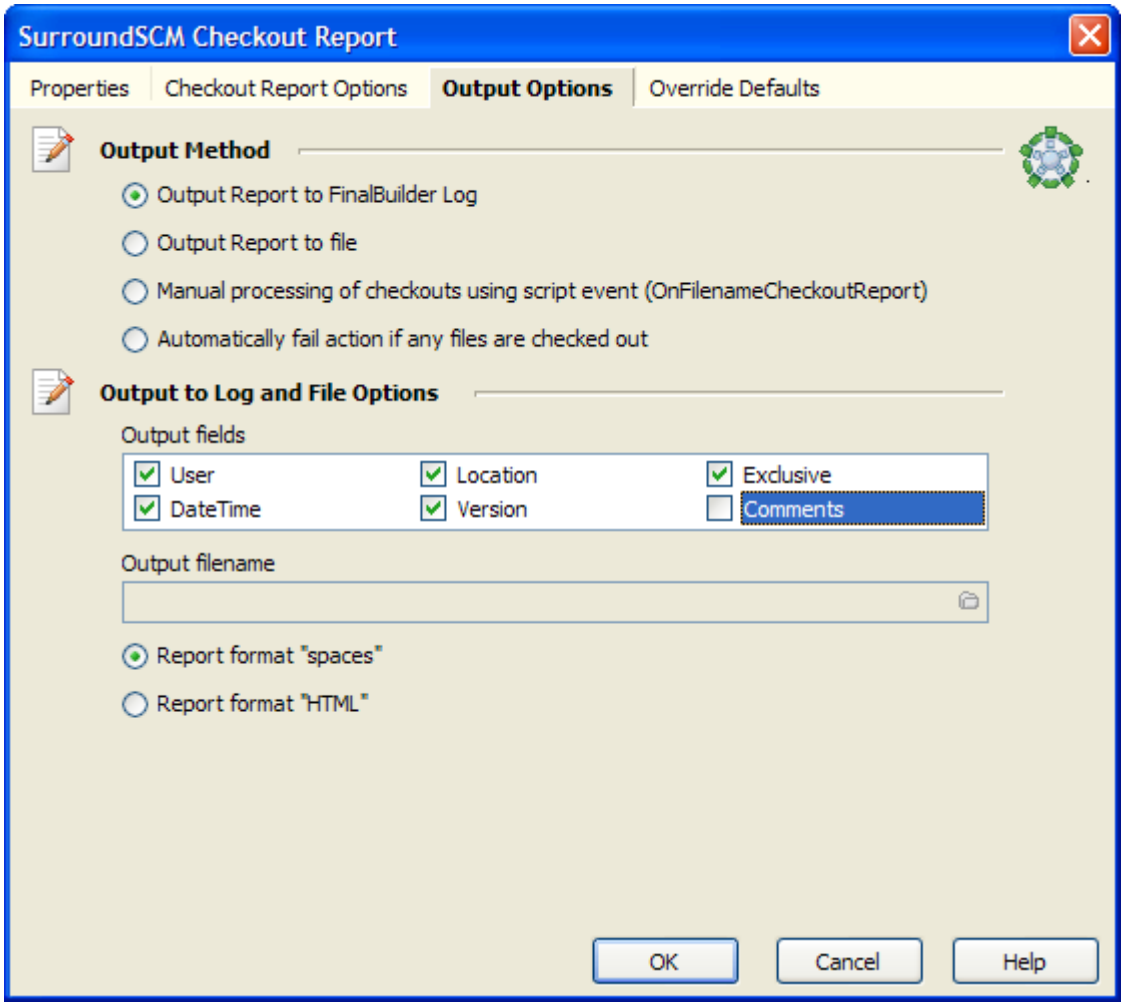
To further filter the results by user, enter one or more user names separated by a comma.

© 2001-2006 VSoft Technologies Pty Ltd



Output Options

© 2001-2006 VSoft Technologies Pty Ltd

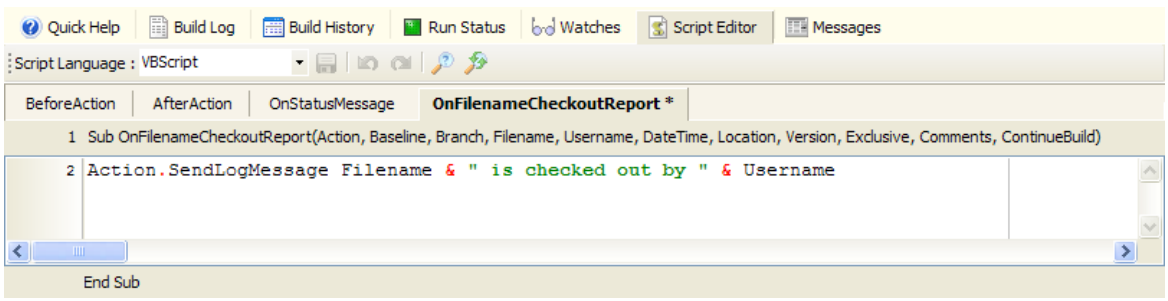


There are four possible modes when this action runs:
1. Output Report to FinalBuilder log
2. Output Report to file

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 (VBScript or JScript) 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

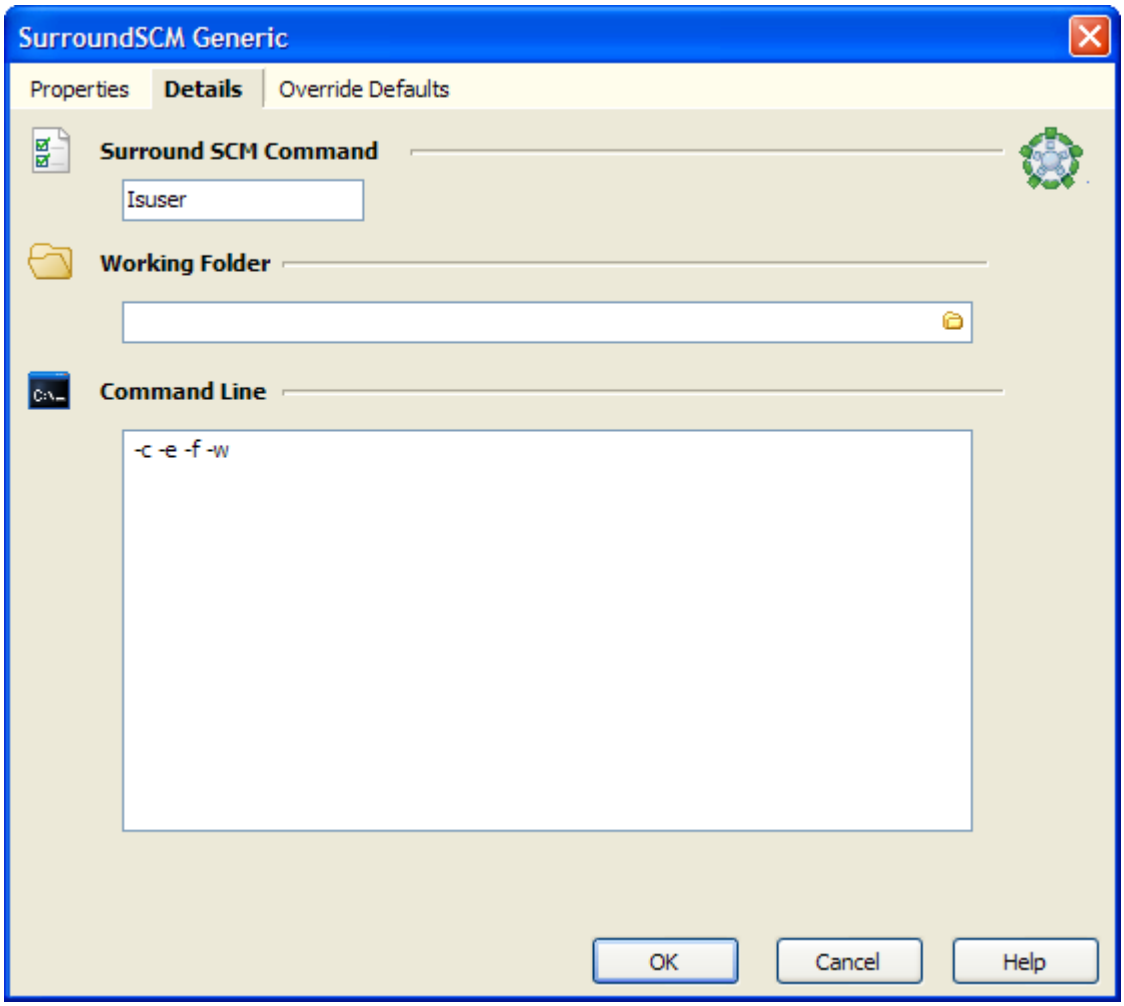
© 2001-2006 VSoft Technologies Pty Ltd

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.

5.5.10.10 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.



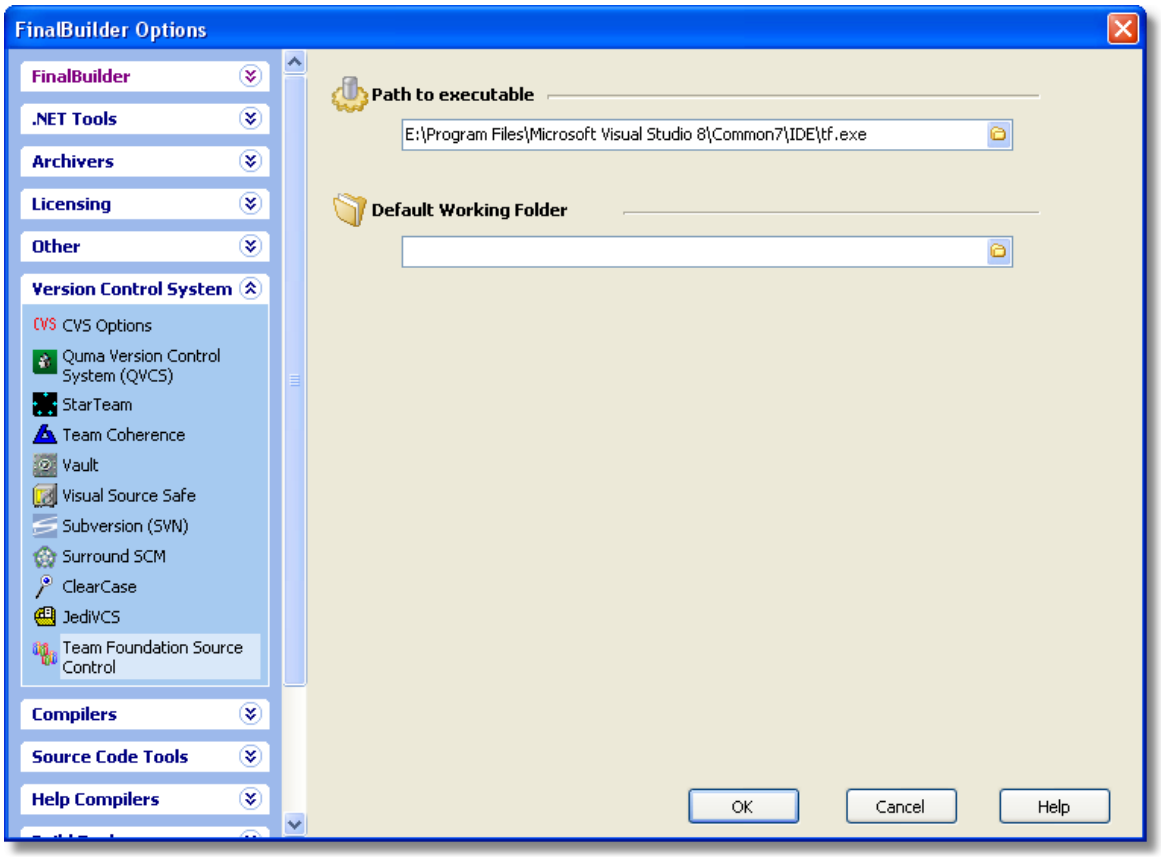
See the "Surround SCM CLI Guide.PDF" file for more information on the Surround SCM commands and command line options.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11 Team Foundation Source Control

5.5.11.1 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. Autodetection is normally successful, but if it fails then it is possible to enter the executable path here.

Default Working Folder

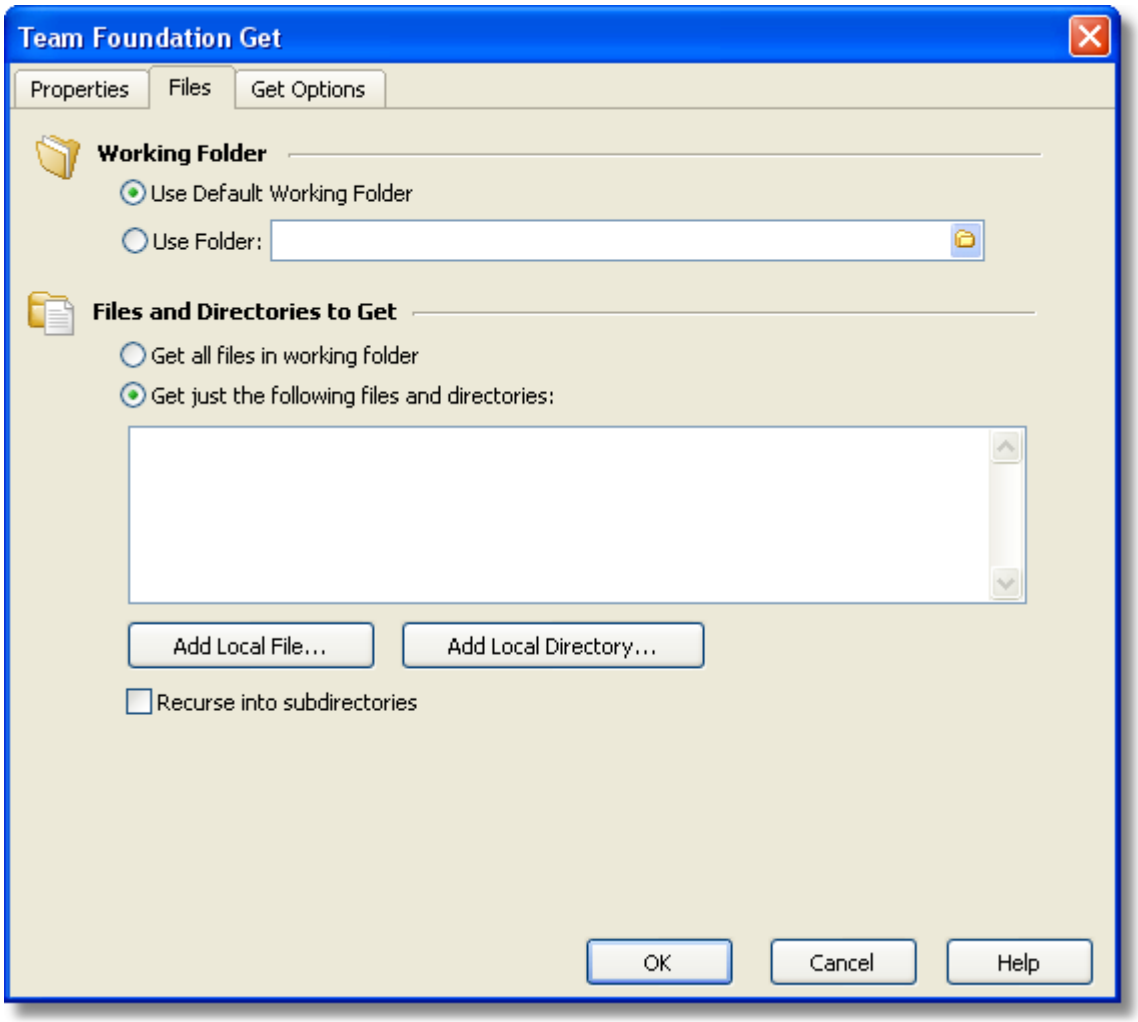
All of the Team Foundation Source Control actions determine the server, project and repository context from a local working folder. You will need to set up working folders in Visual Studio, or with Map Working Folder Action.

You can specify a default working folder for all actions, or specify individual working folders on a per-action basis.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11.2 Team Foundation Get

Use the Team Foundation Get action to update a local working folder with the contents of the repository.



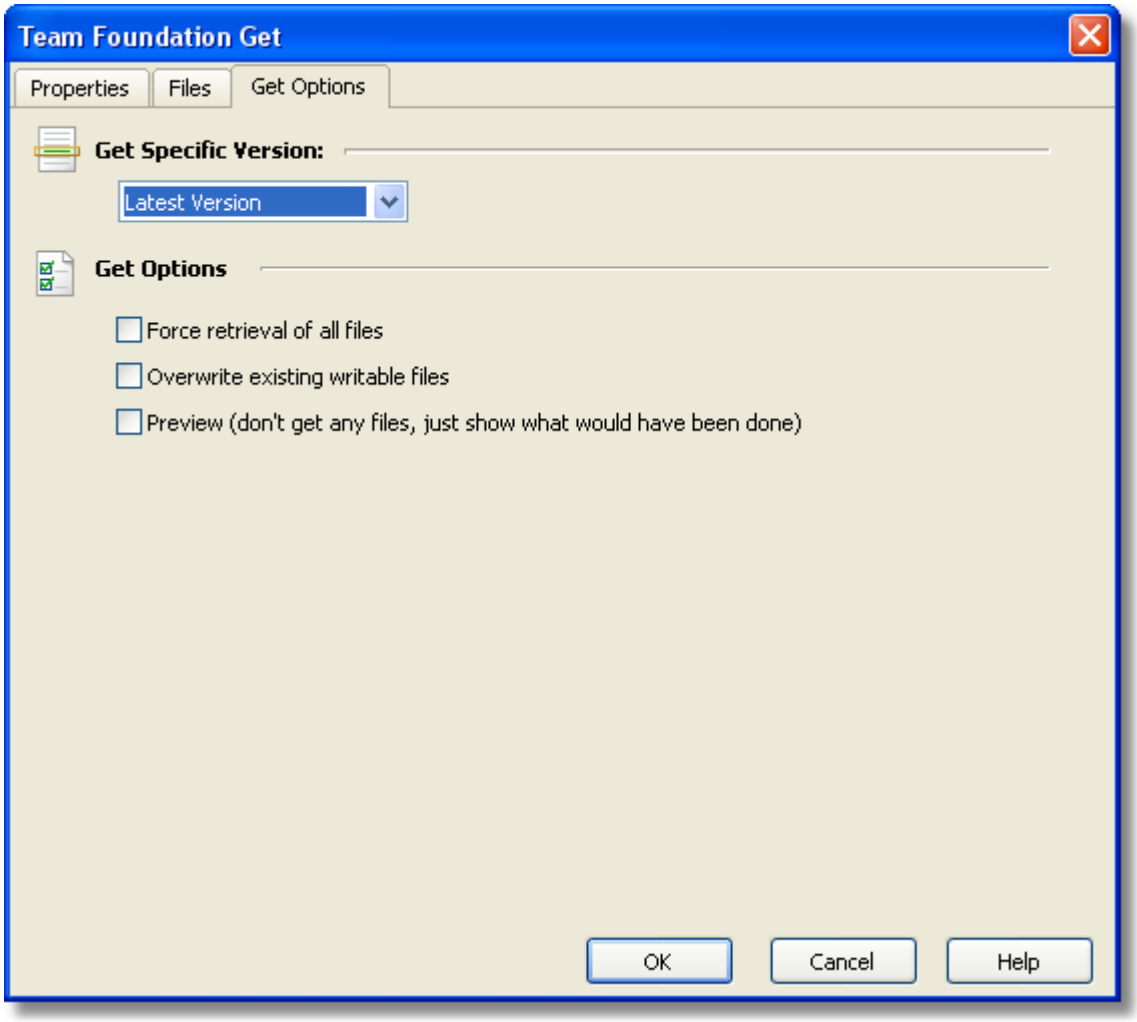
Working Folder

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

Specify the files and directories you wish to update. You can Get all files, or just a specific set of files and directories. Use wildcards (ie *) to select groups of files. Put each new file/directory on a new line.

© 2001-2006 VSoft Technologies Pty Ltd



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.

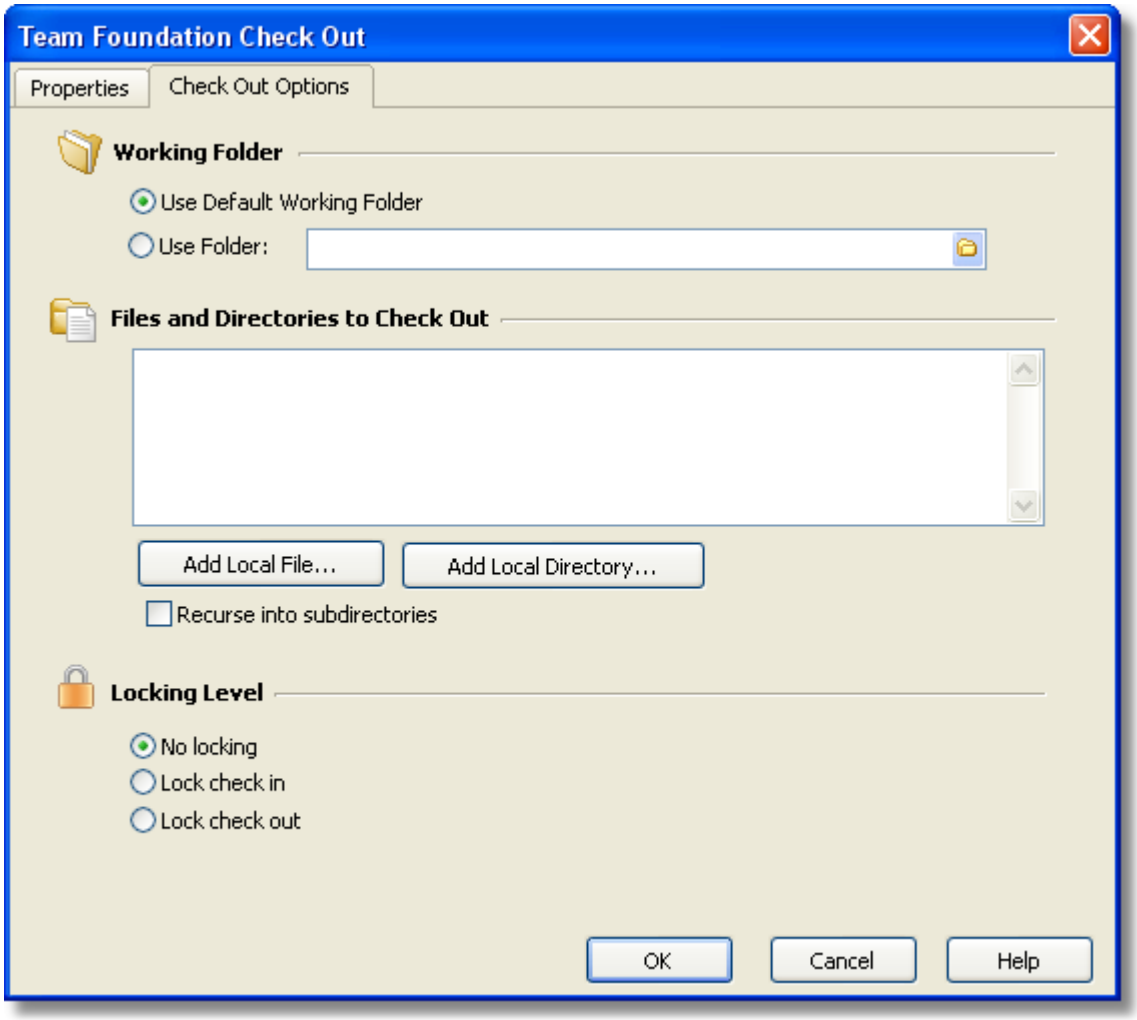
"**Preview**"

© 2001-2006 VSoft Technologies Pty Ltd

If this option is set, no files will be changed but the action will log the changes which would have been made.

5.5.11.3 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

© 2001-2006 VSoft Technologies Pty Ltd

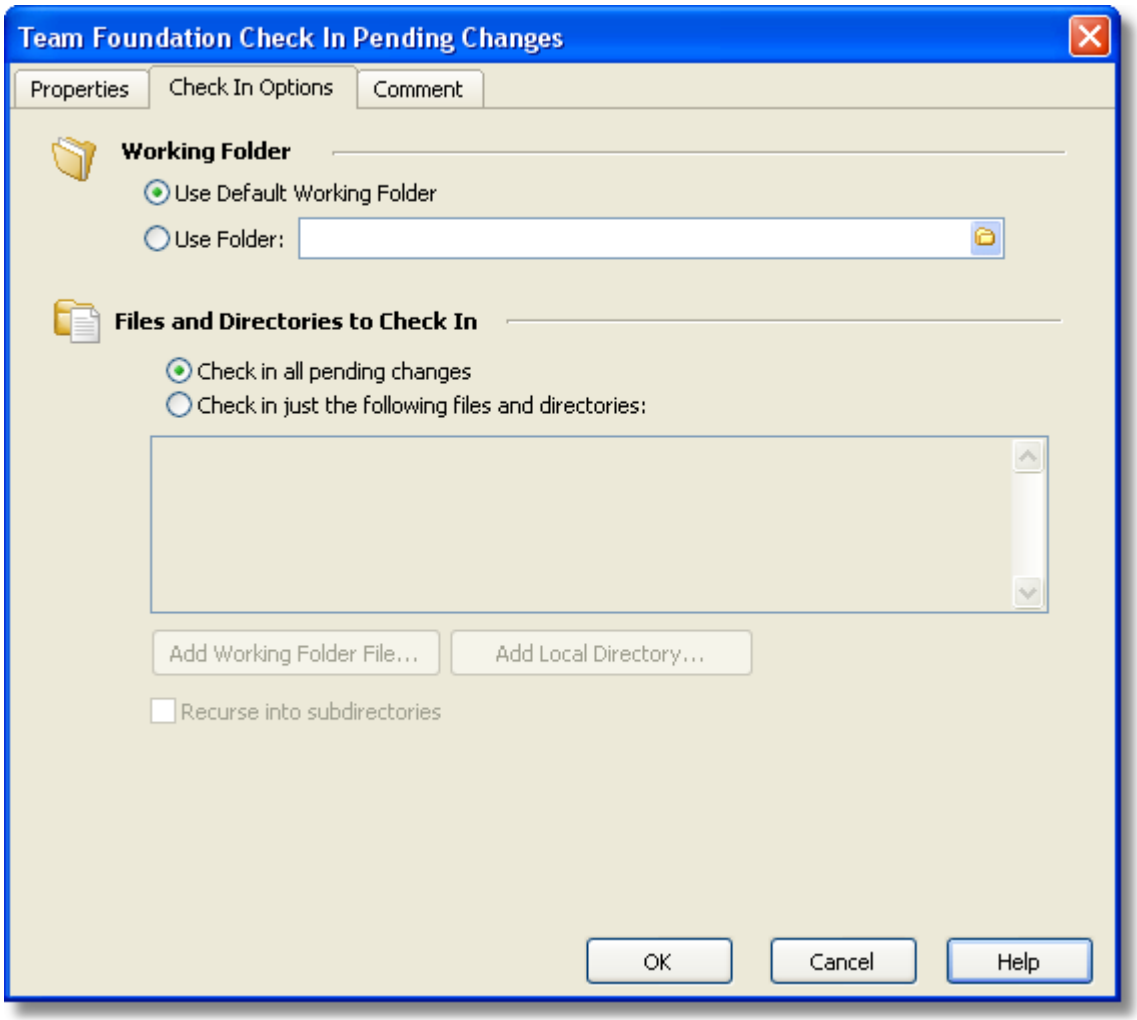
Specify the locking level for the checked out files. "Lock check in" prevents others from checking in changes to the same files. "Lock check out" prevents others from checking out the same files.

5.5.11.4 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

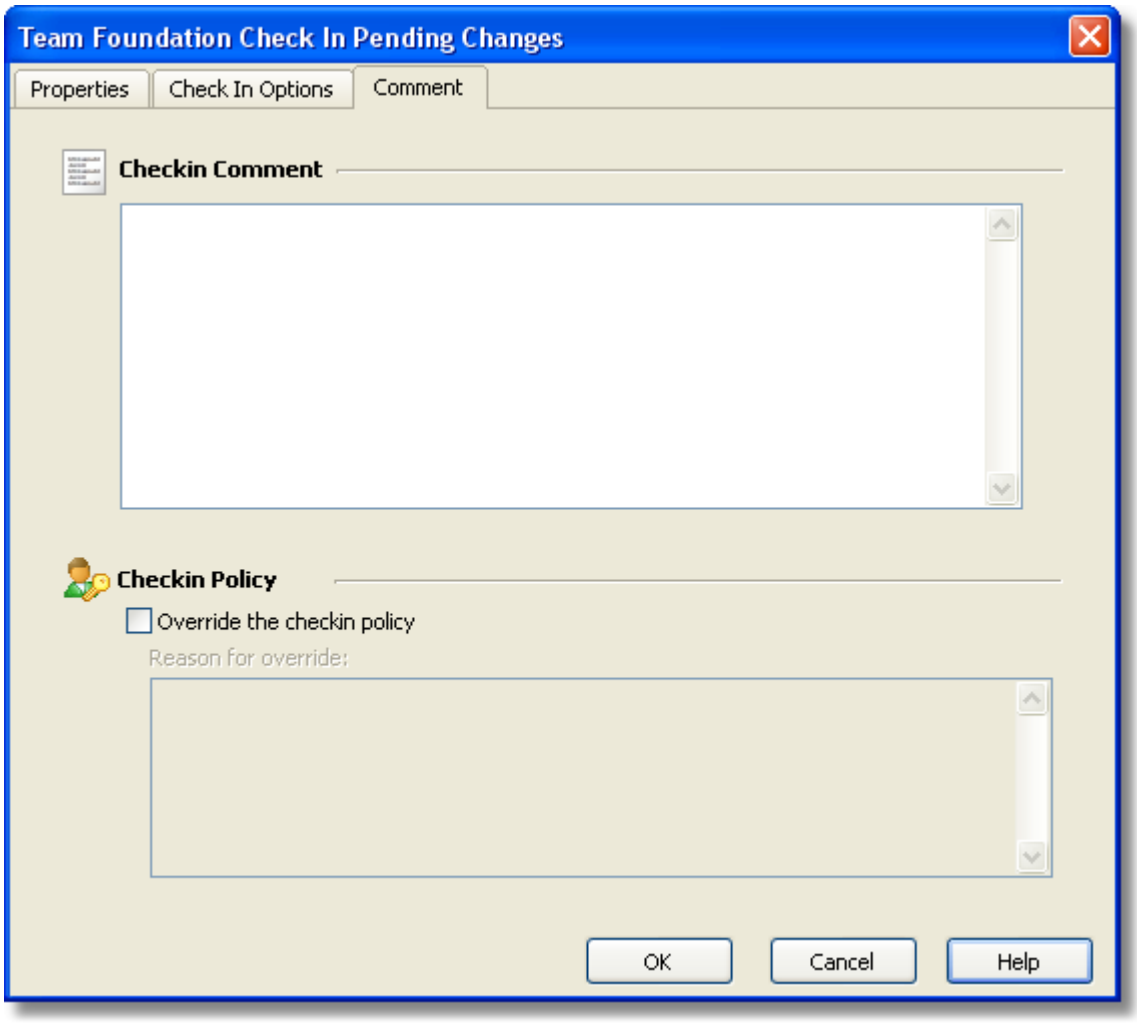
© 2001-2006 VSoft Technologies Pty Ltd

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

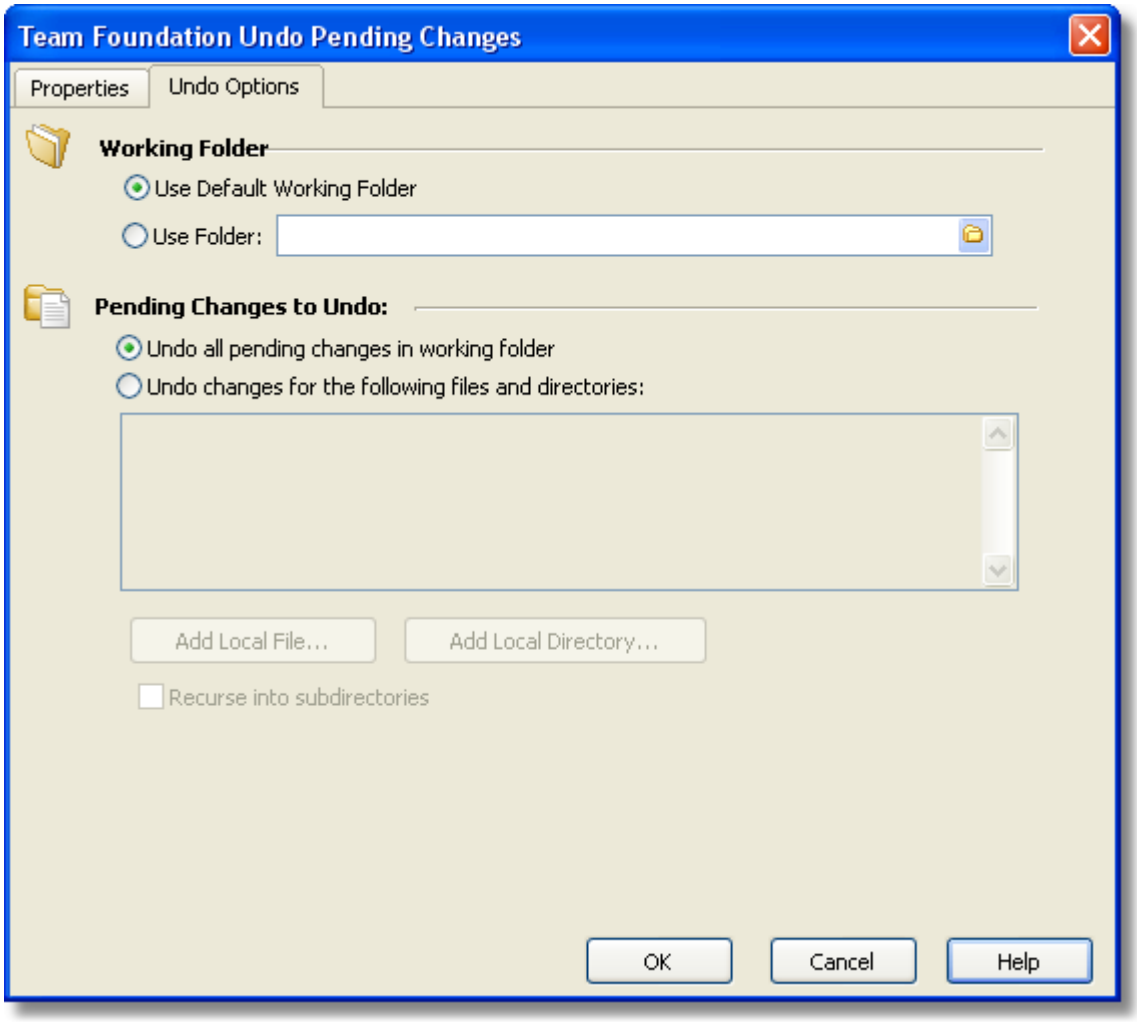


Specify a checkin comment for your checkin. If the checkin is likely to violate the checkin policy, you can choose to override the checkin policy (with a reason.)

5.5.11.5 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.

© 2001-2006 VSoft Technologies Pty Ltd



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

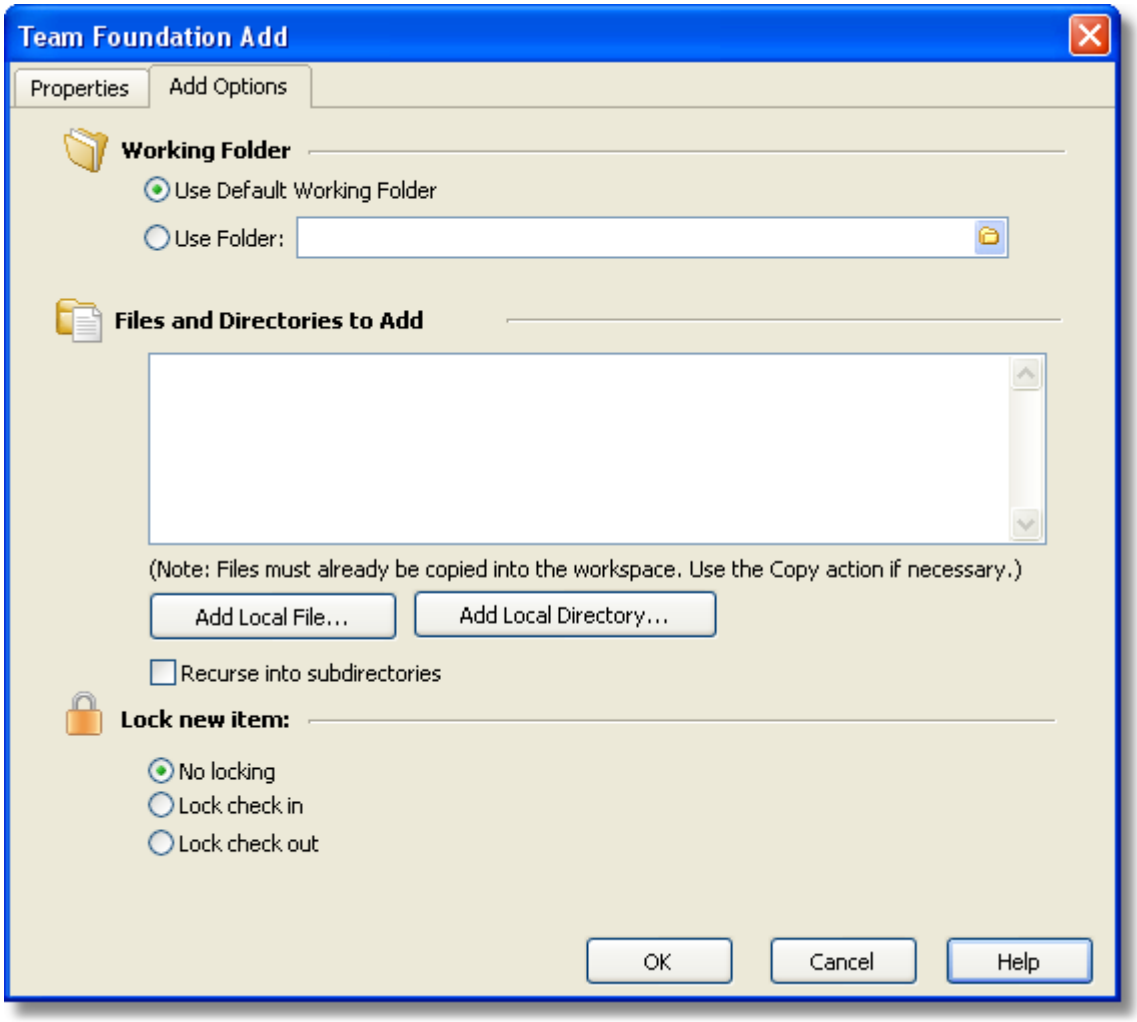
You can choose to undo all pending changes in that working folder, or to undo changes made to specific files. If choosing specific files, put eac new file/directory on a new line. Use wildcards (ie *) to select groups of files.

5.5.11.6 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.

© 2001-2006 VSoft Technologies Pty Ltd



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 Files(s) (or Copy/Move File List) actions if necessary.

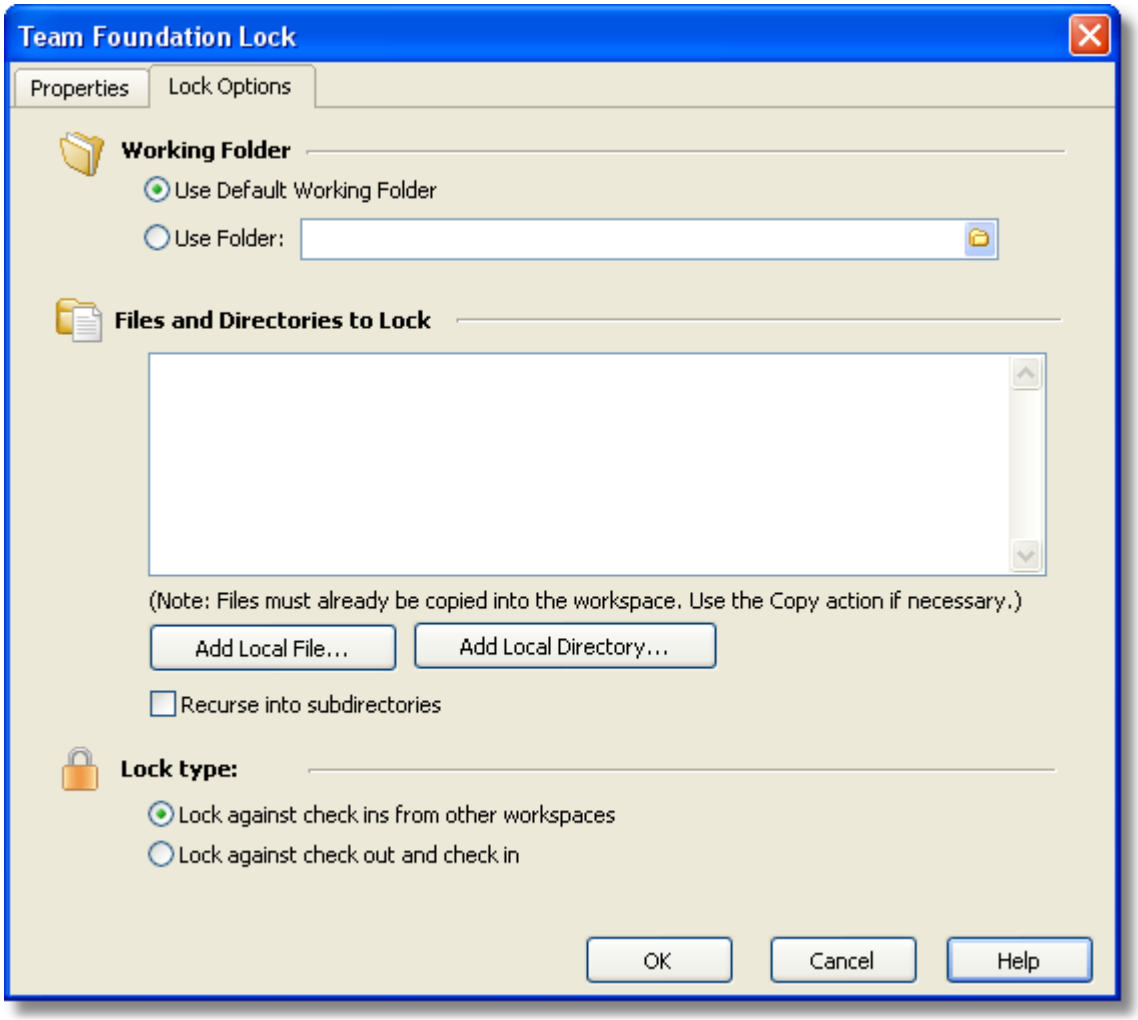
Lock new items

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11.7 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.

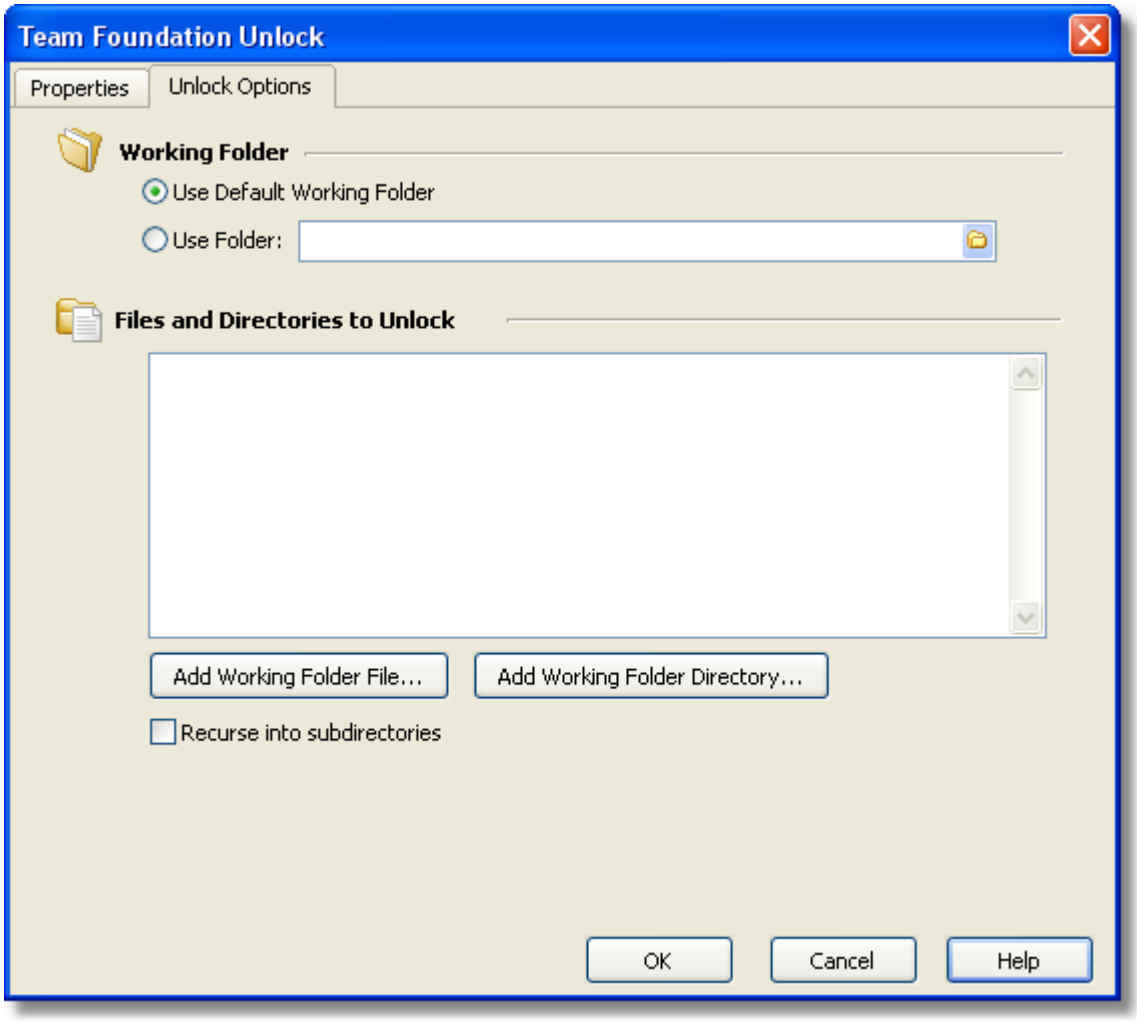
Lock type

The action can be locked against checkins from other workspaces, or against both checkins and check outs.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11.8 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.

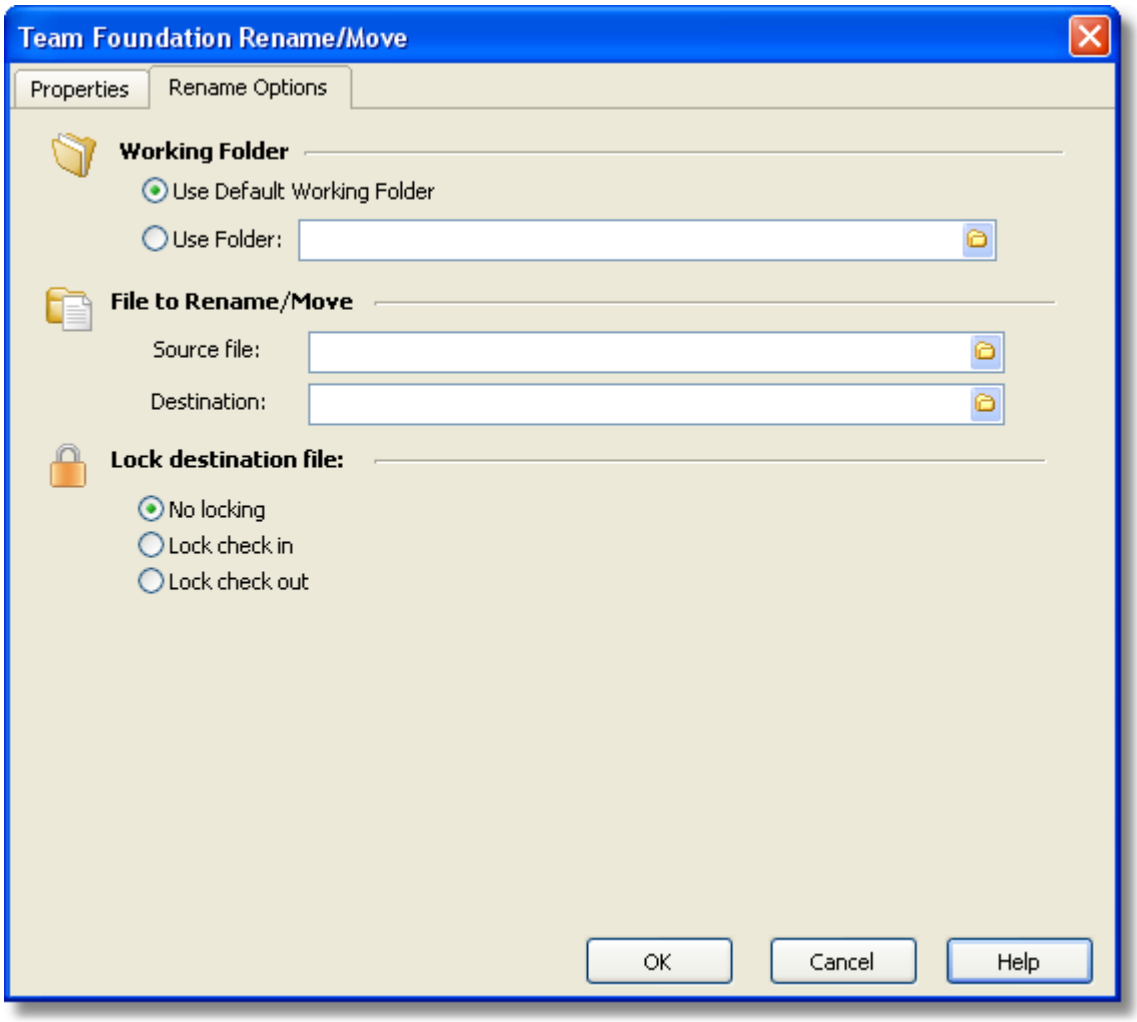
Files and Directories to Unlock

Specify a list of files and directories to unlock. Put each file/directory on a new line. Use wildcards (*) to specify groups of files.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11.9 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.

© 2001-2006 VSoft Technologies Pty Ltd

Lock destination file

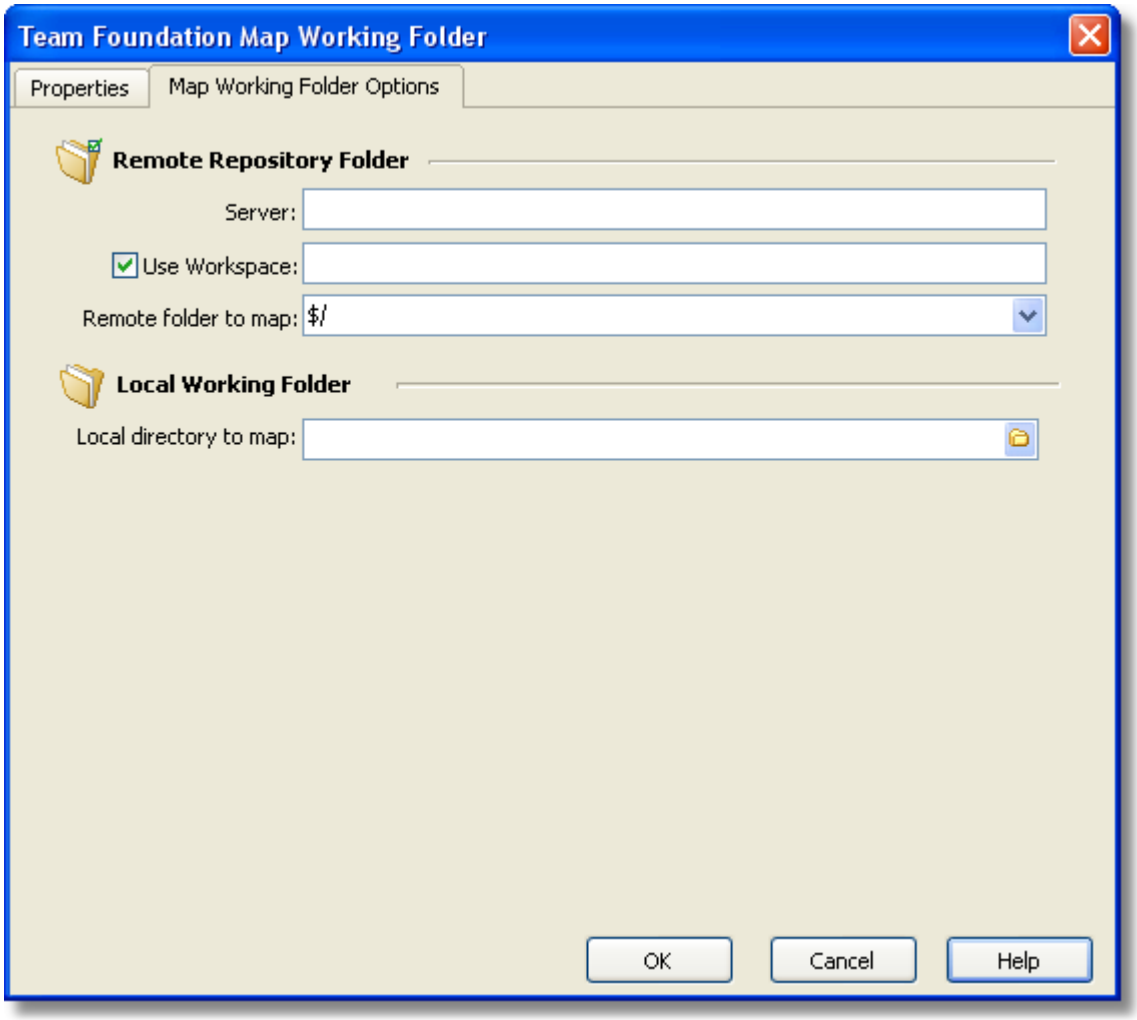
The destination file can be locked against check in or check out by other users.

5.5.11.10 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,

© 2001-2006 VSoft Technologies Pty Ltd

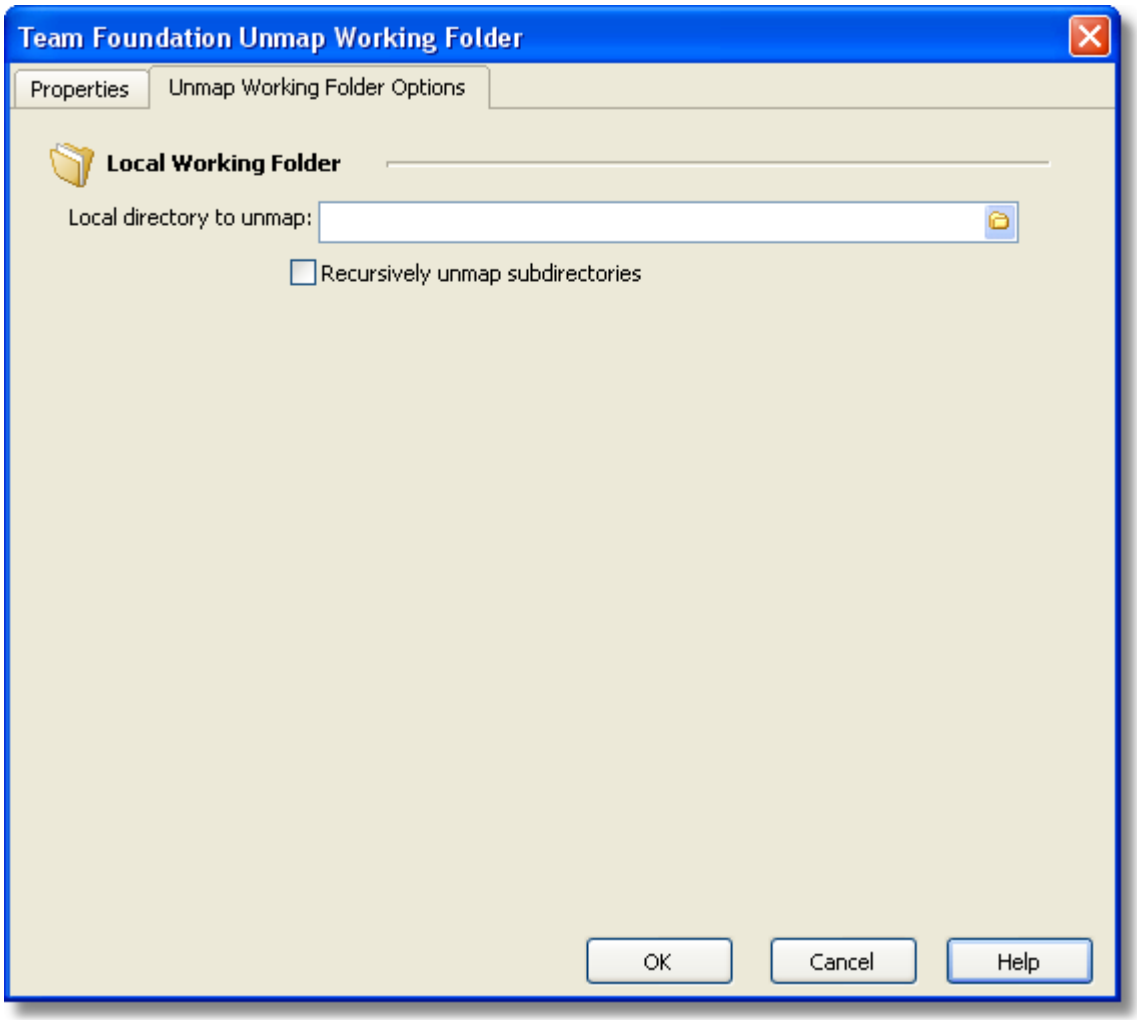
(optionally) a specific workspace, and also the folder inside that workspace.

Local Working Folder

This is the local directory to map the working folder onto.

5.5.11.11 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.)



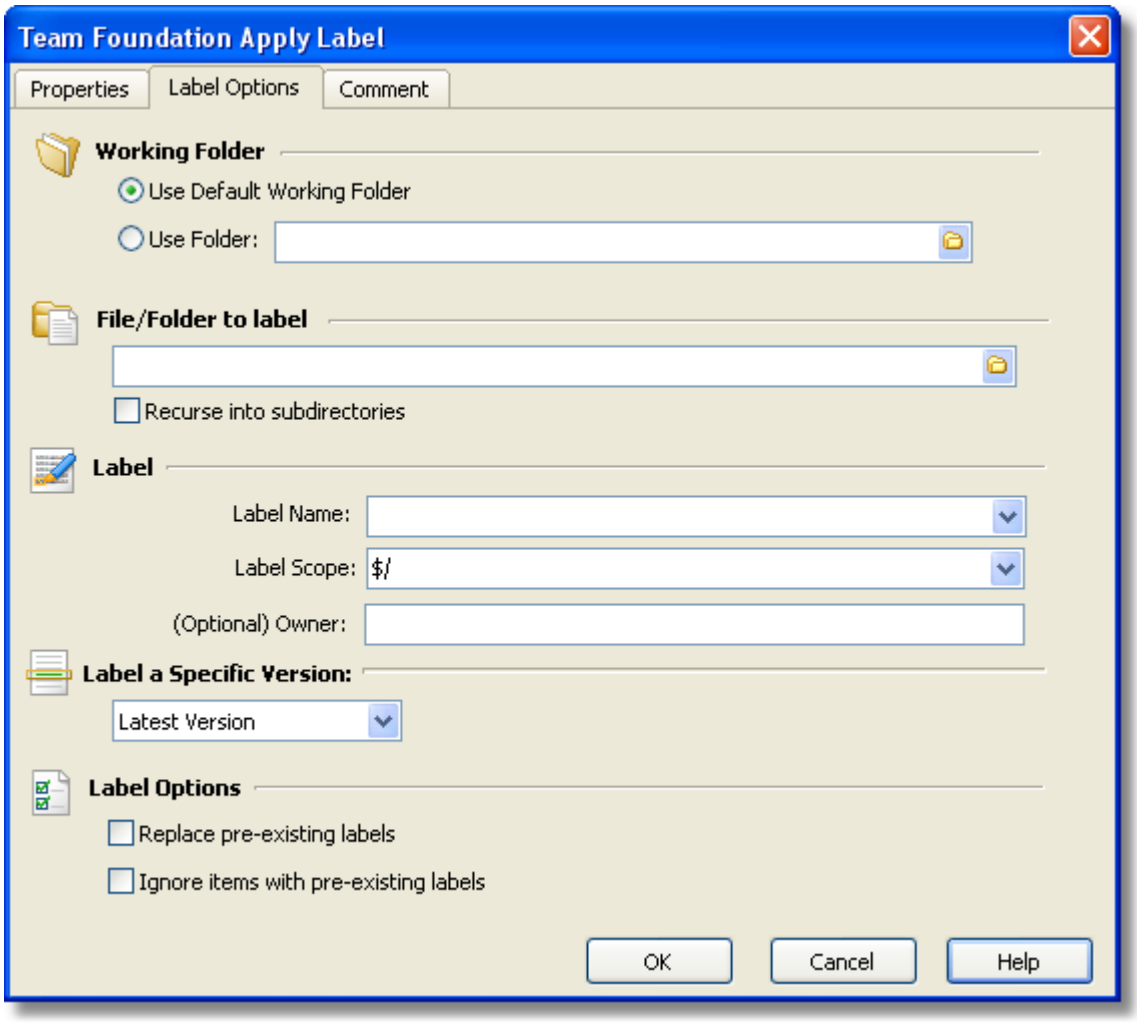
Local Working Folder

Specify the directory to unmap. Optionally, you can recursively unmap subdirectories which are also working folders.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11.12 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

© 2001-2006 VSoft Technologies Pty Ltd

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)/directorie(s) will be removed. You must have sufficient privileges to remove labels belonging to other users.

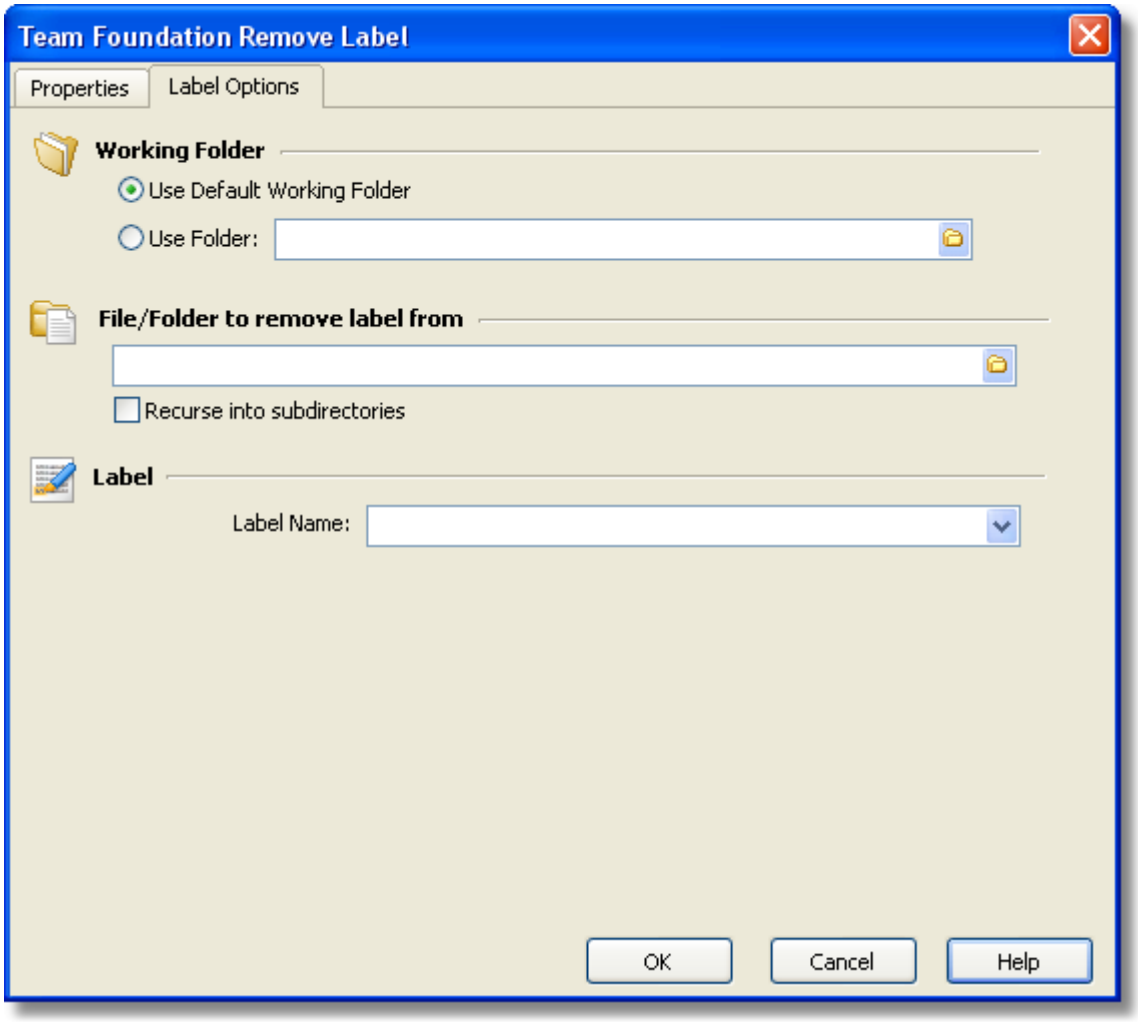
"**Ignore items with pre-existing labels**"

If a file/directory is already labelled, it will not be labelled again.

5.5.11.13 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

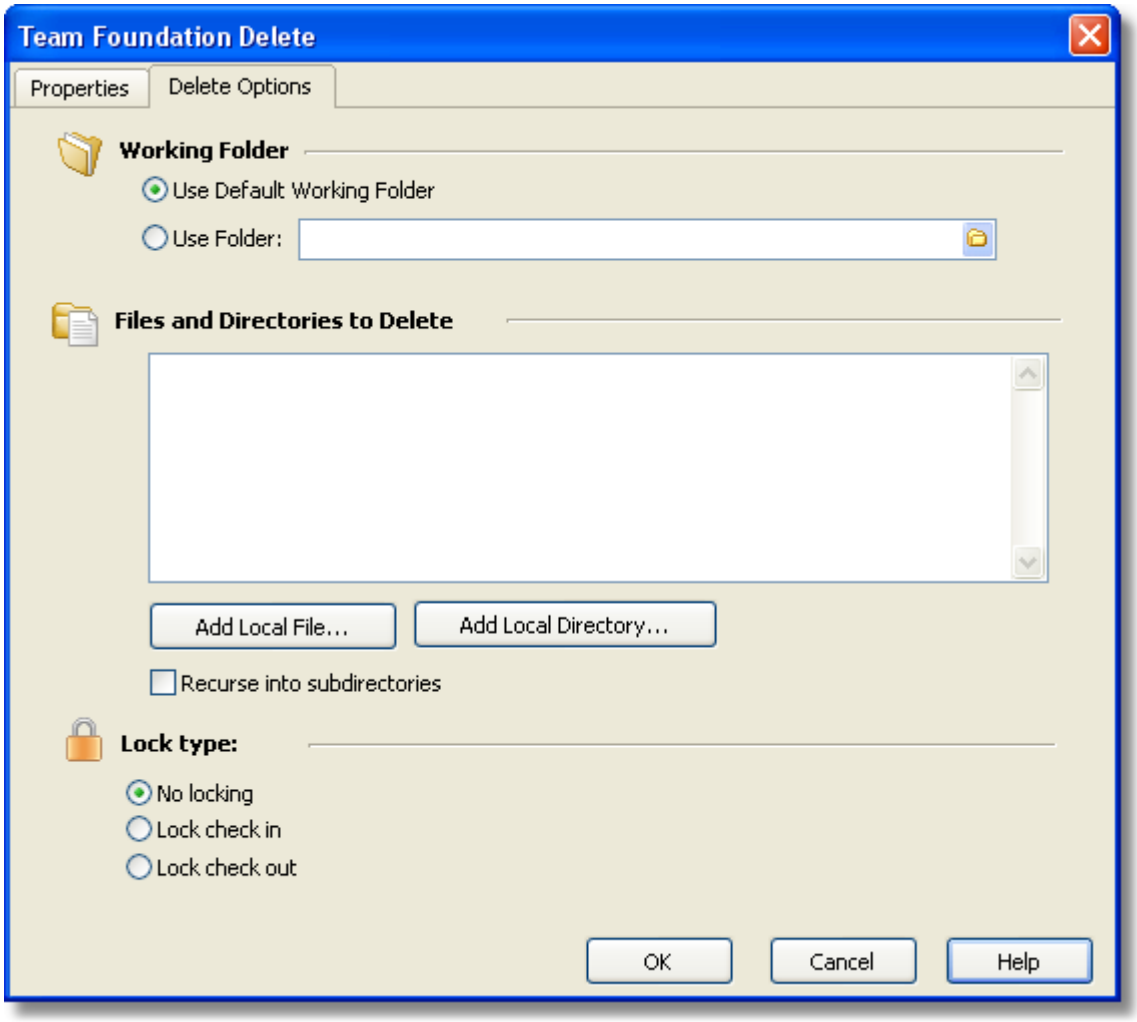
Label

Give the name of the label to remove.

5.5.11.14 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.

© 2001-2006 VSoft Technologies Pty Ltd



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

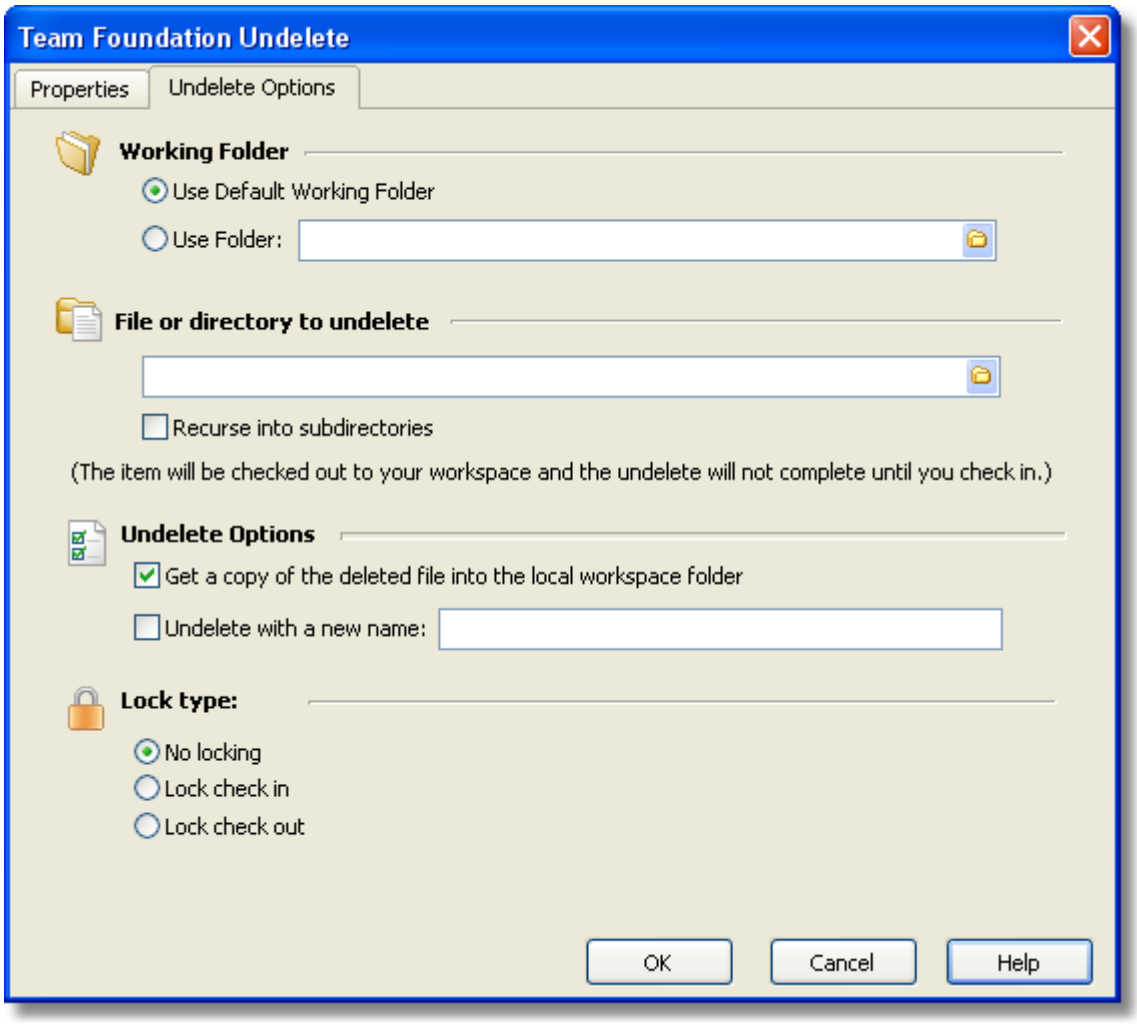
Select a lock type to lock the deleted files from check in/check out until the delete is checked in (the delete remains pending until Check In Pending Changes is used.)

5.5.11.15 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

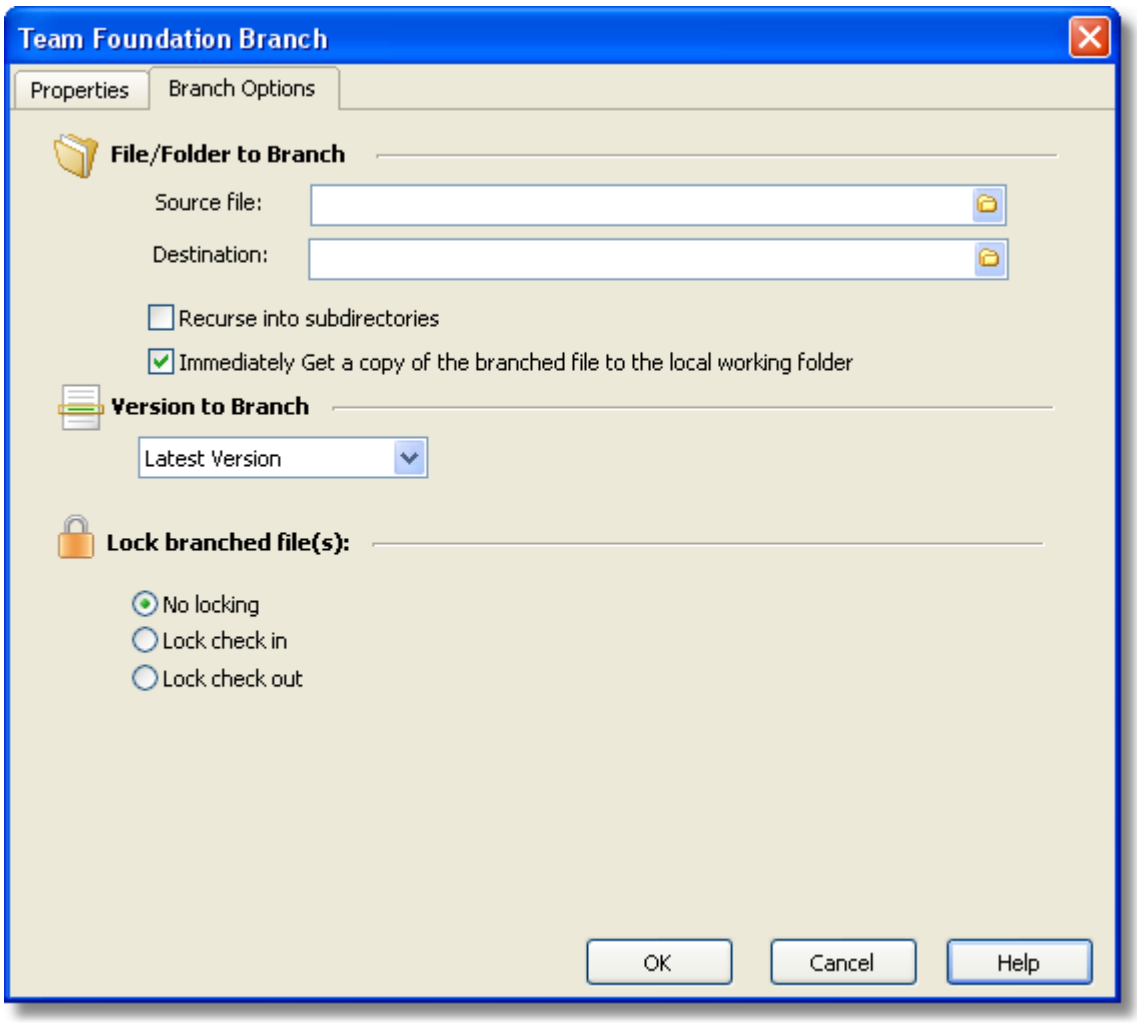
Lock Type

© 2001-2006 VSoft Technologies Pty Ltd

The new file can be locked until a check in or unlock operation occurs.

5.5.11.16 Team Foundation Branch

Use the Team Foundation Branch action to create a new branch based on file(s) or directorie(s). 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.

Desintation : 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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

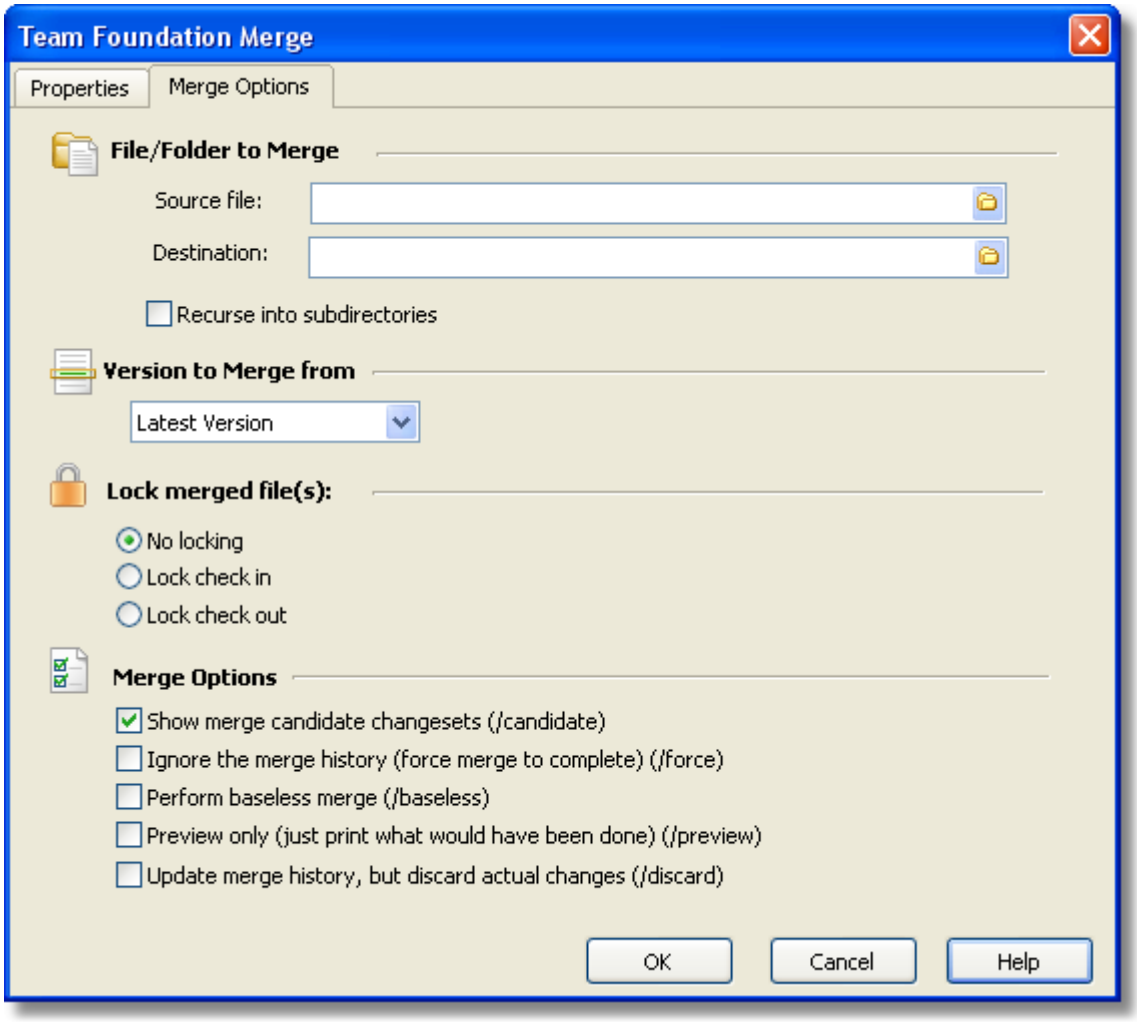
5.5.11.17 Team Foundation Merge

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

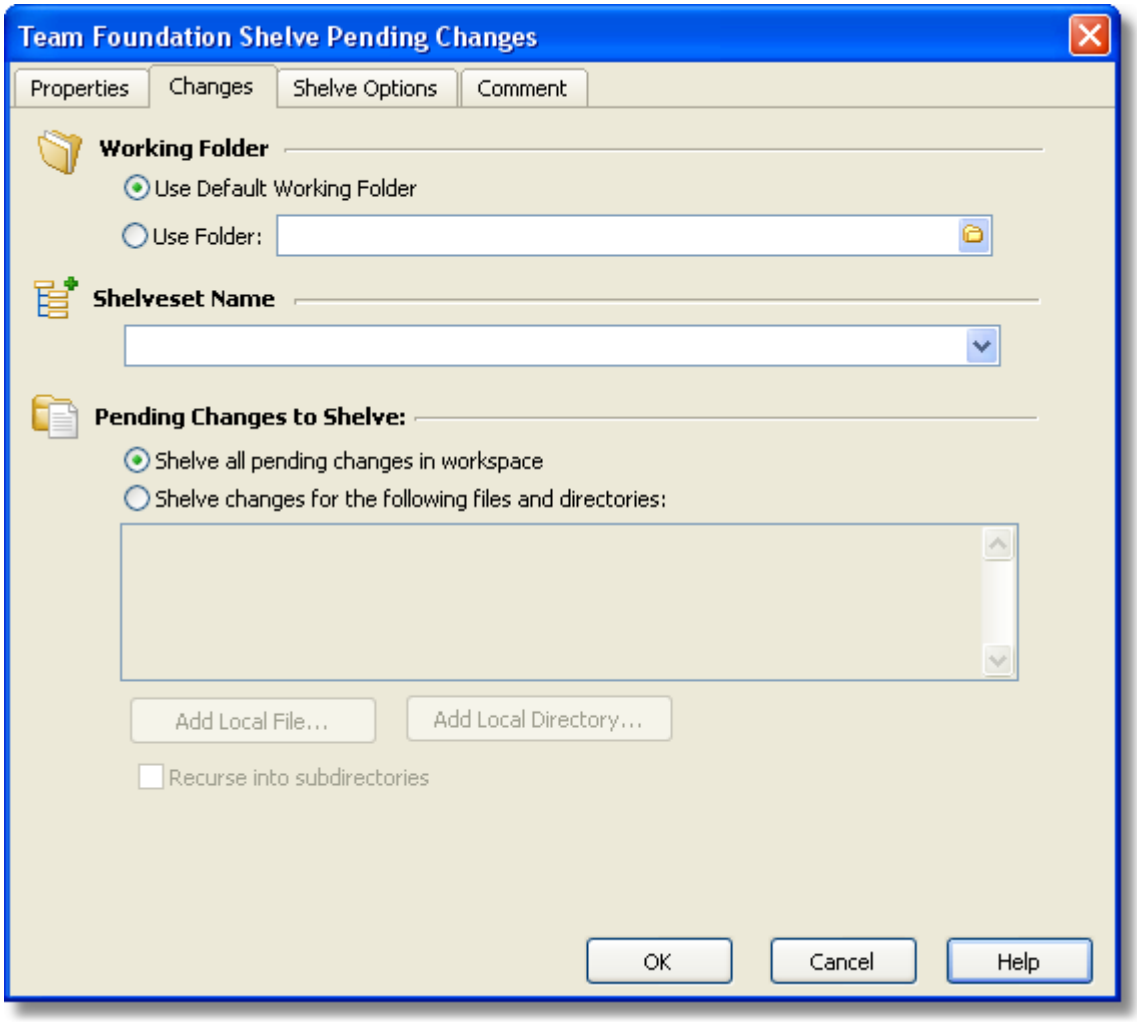
Update merge history, but discard actual changes (/discard)

If this option is set, Team Foundation will update the merge history but not invoke the actual changes.

5.5.11.18 Team Foundation Shelve 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

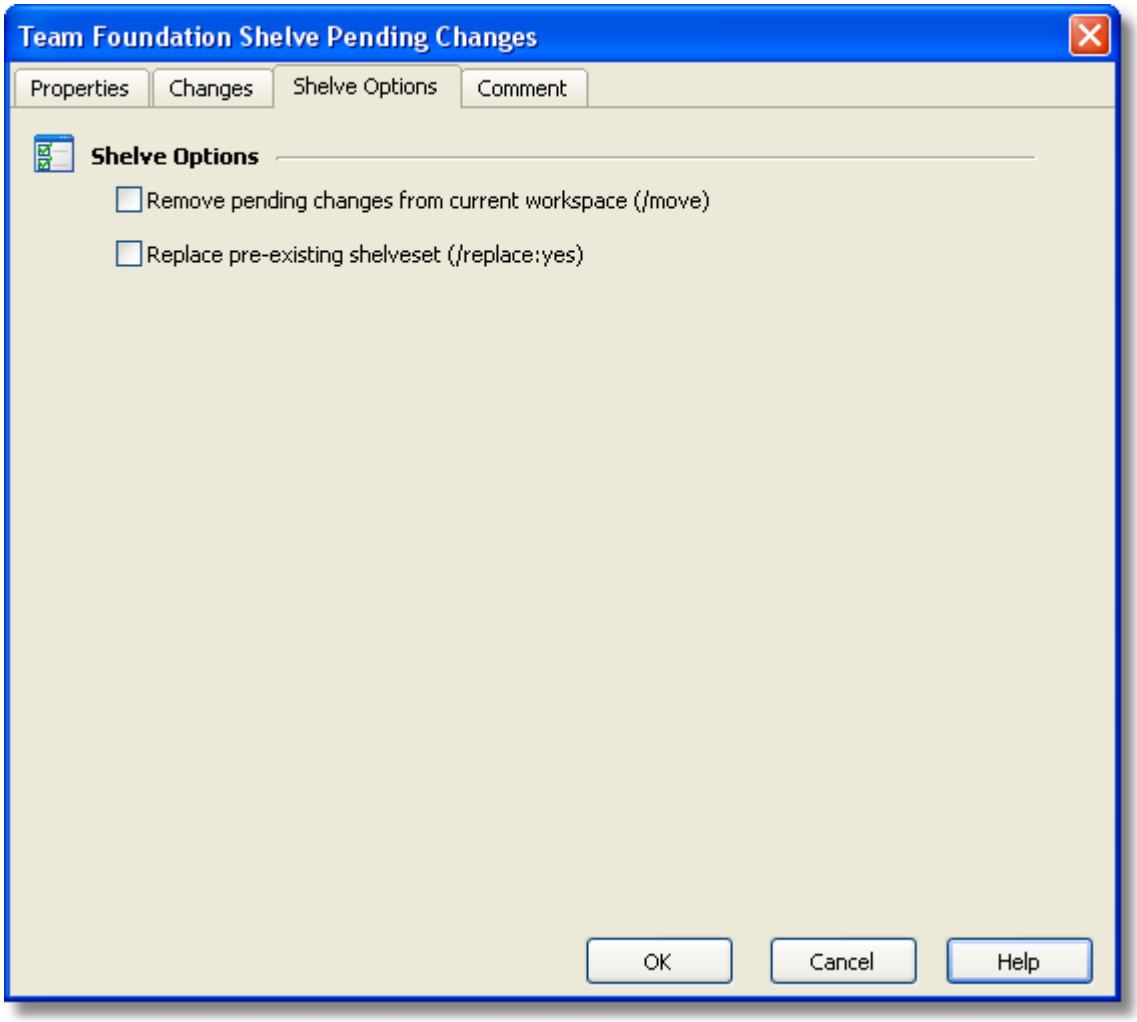
Shelveset Name

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 Shelve

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.

© 2001-2006 VSoft Technologies Pty Ltd



"**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.

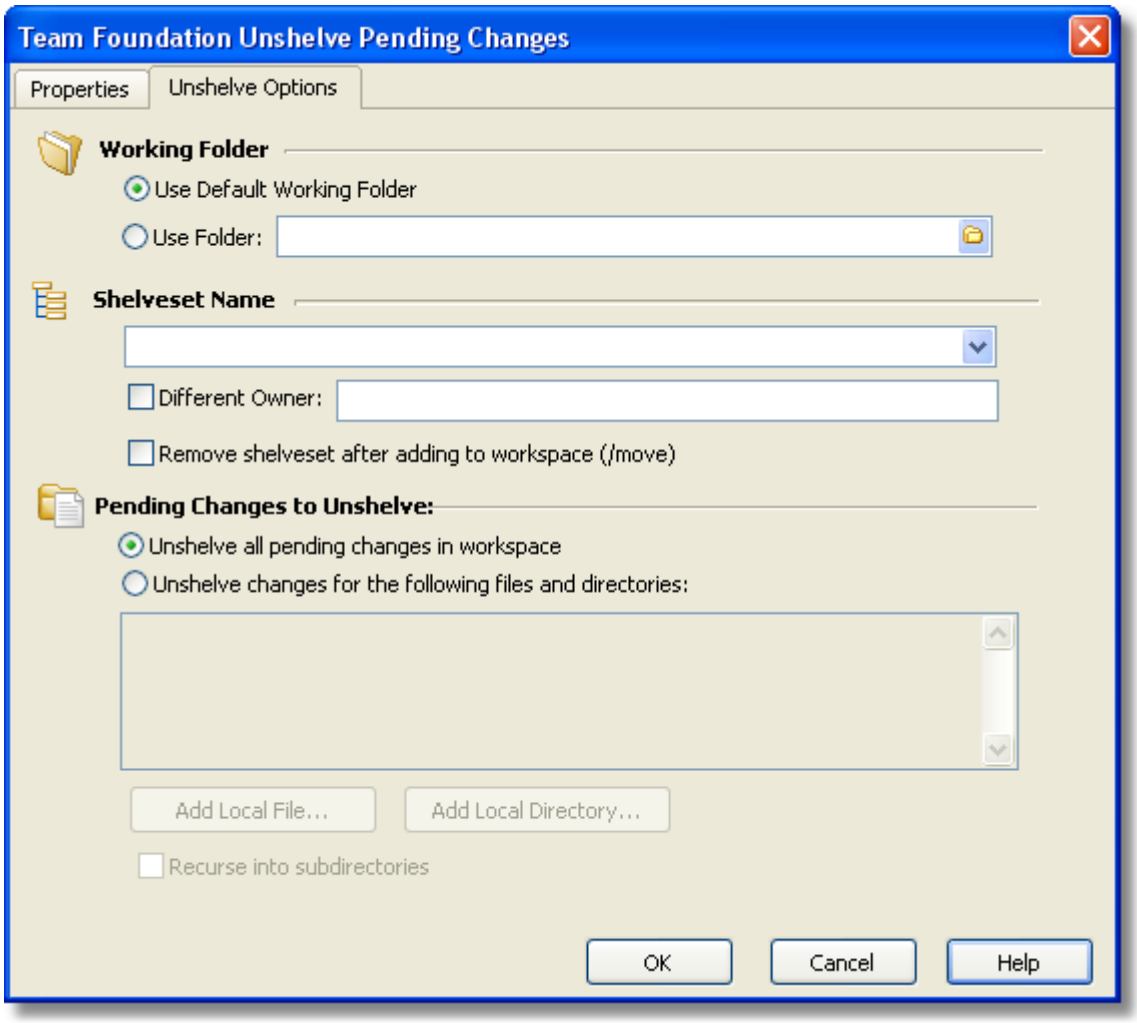
Comment

Specify a comment for the shelved changes.

5.5.11.19 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

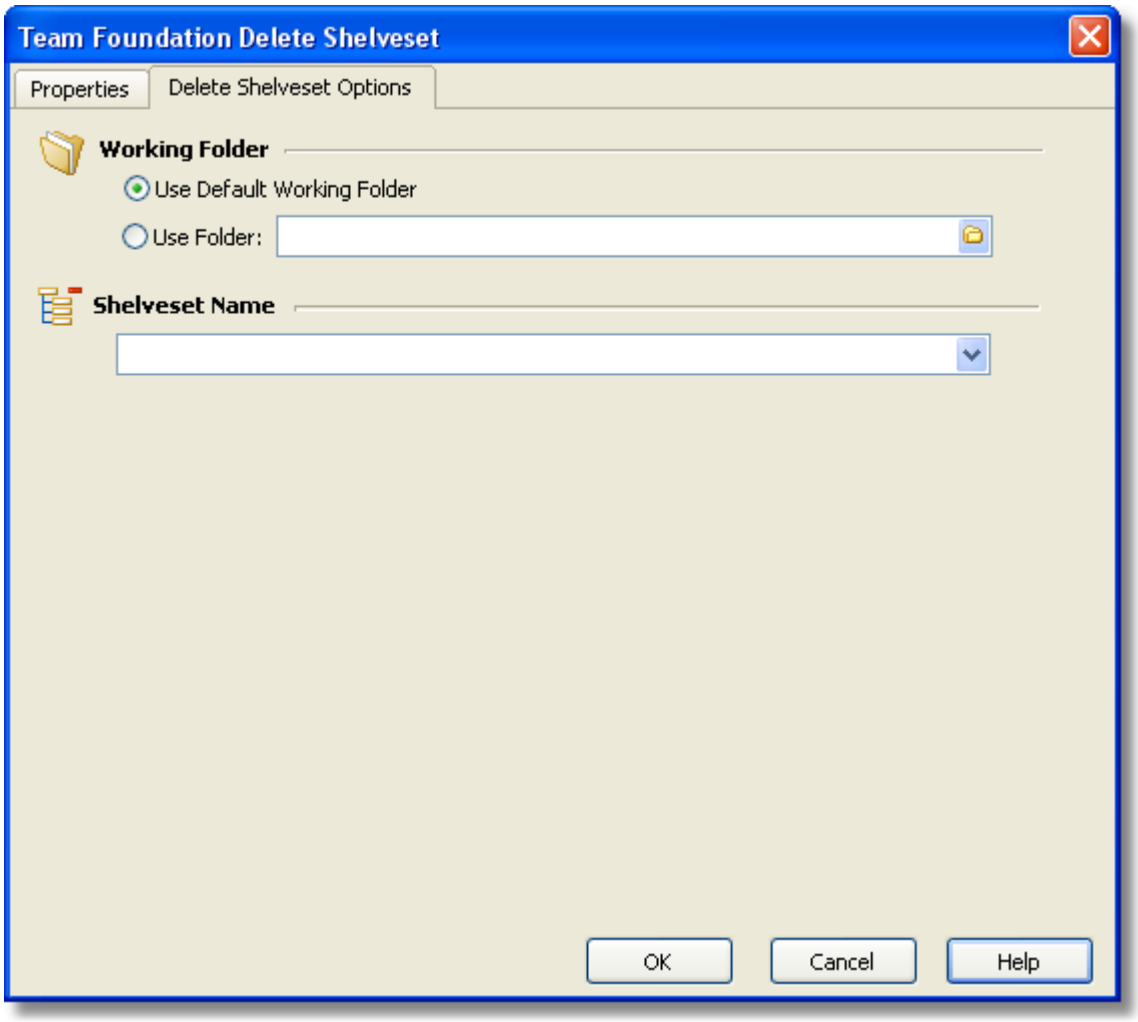
© 2001-2006 VSoft Technologies Pty Ltd

Pending Changes to Unshelve

You can choose to unshelve all changes associated with a shelveset, or only changes made to specific files. Specify each file or directory on a new line. Use wildcards to specify groups of files.

5.5.11.20 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

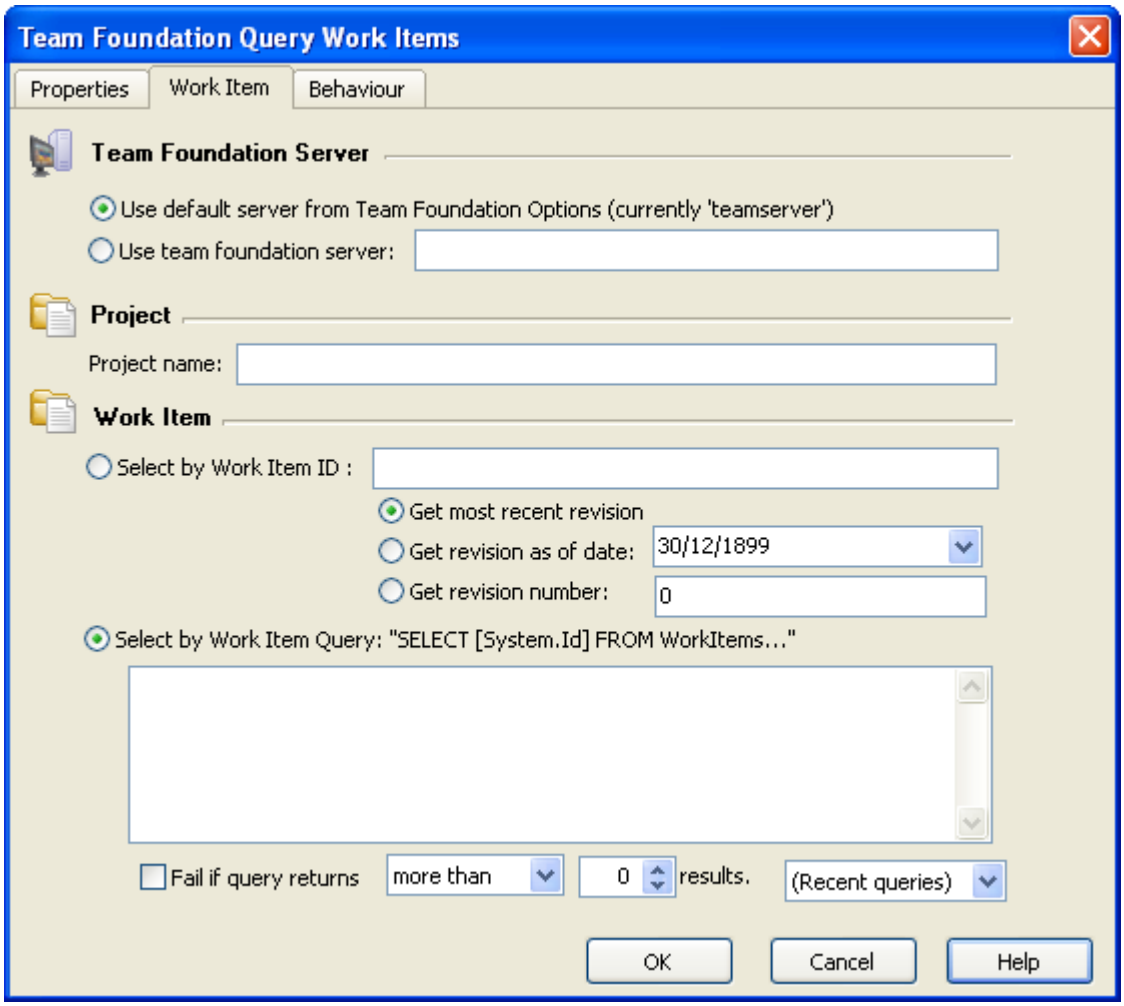
Specify the name of the shelveset to delete. Click on the dropdown to see a list of current shelvesets associated with the specified working folder.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.11.21 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

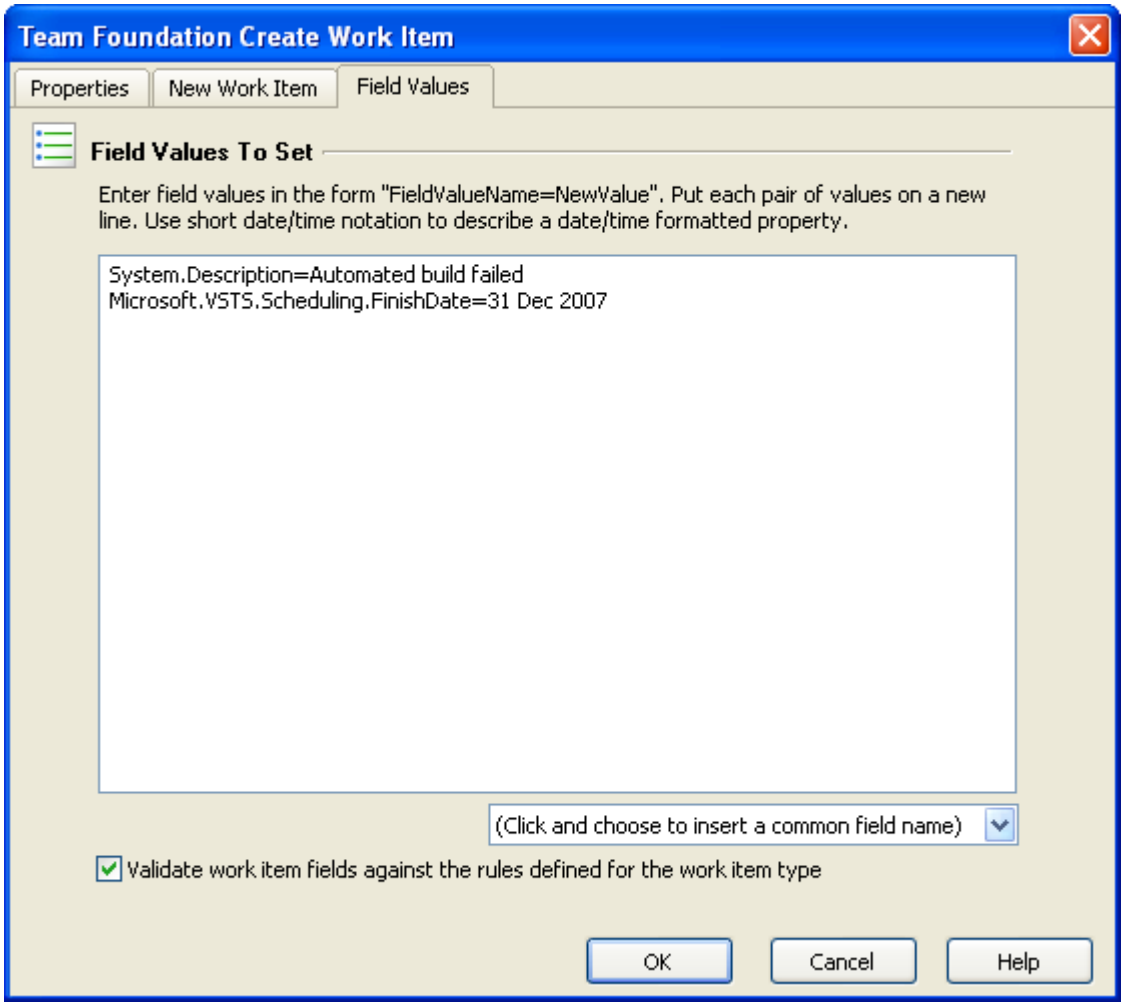
"**(Recent queries)**"

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:

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.5.11.21.1 Team Foundation Query Work Items

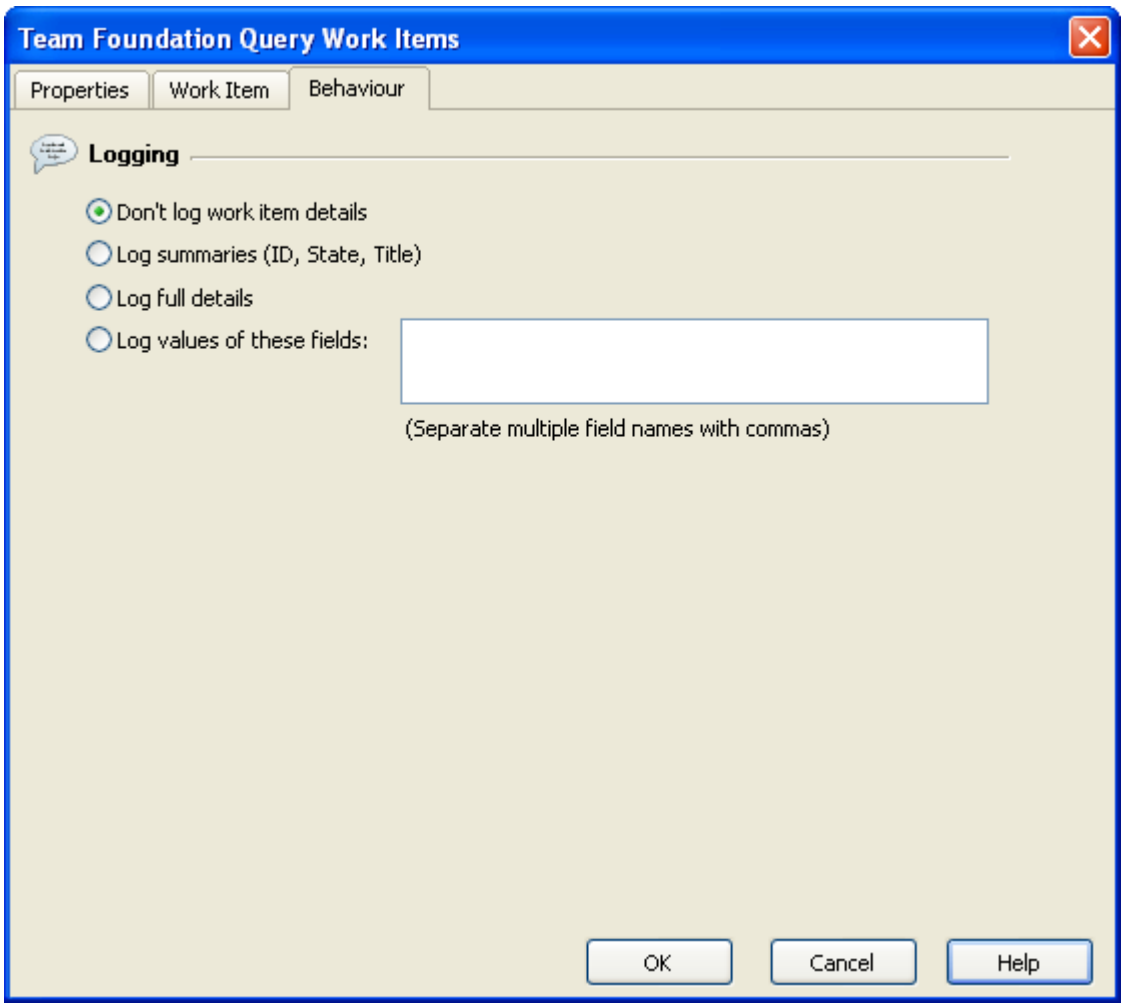
The Team Foundation Query Work Items action allows you to get information for one or more Team Foundation work items. You can use the Work Item Query Language to make the query, if needed.

© 2001-2006 VSoft Technologies Pty Ltd

Work Item Property Page

See the Team Foundation Work Items overview topic for details.

Behaviour Property Page



"**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.

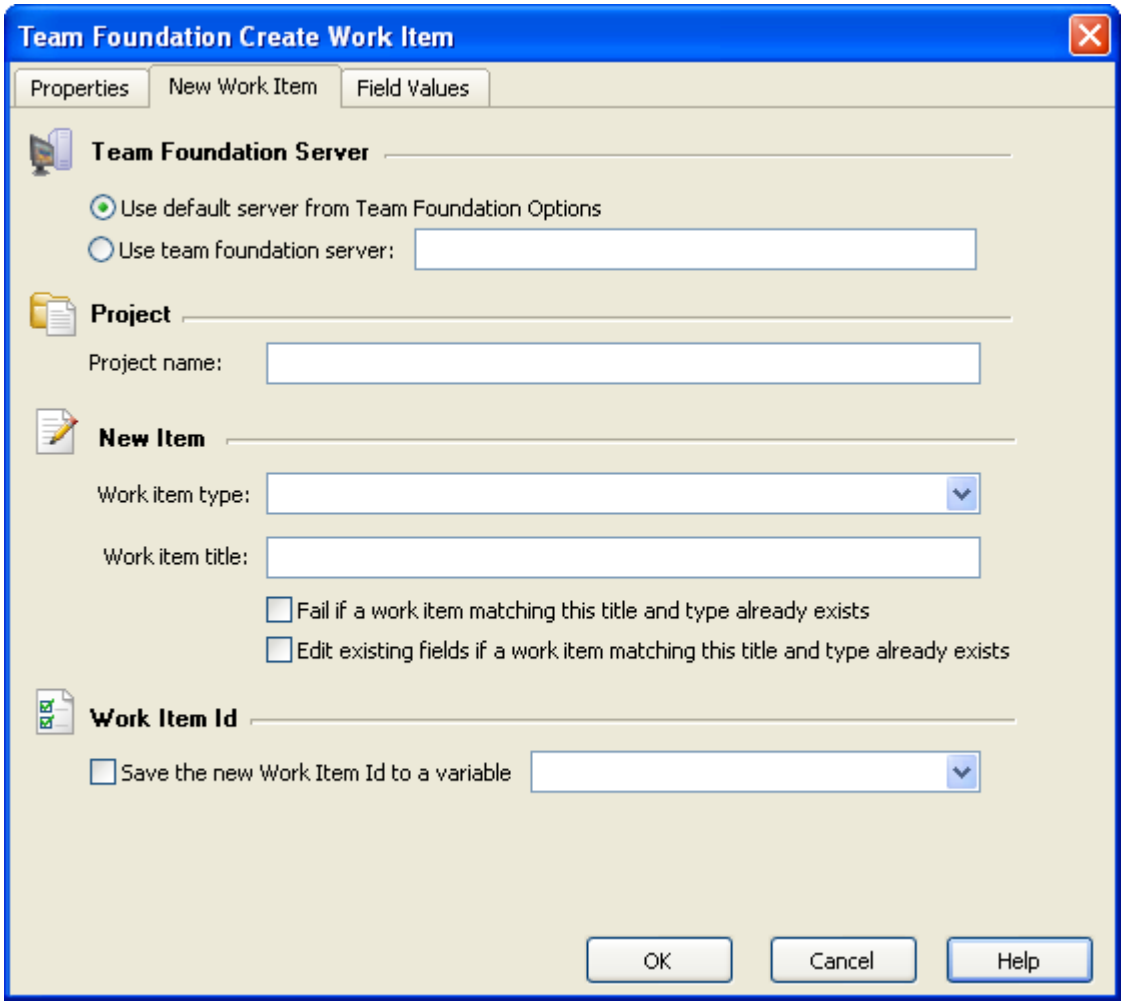
© 2001-2006 VSoft Technologies Pty Ltd

Log values of these fields

Enter the names of fields to log. Separate multiple fields with commas.

5.5.11.21.2 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 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

5.5.11.21.3 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.

5.5.11.21.4 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).

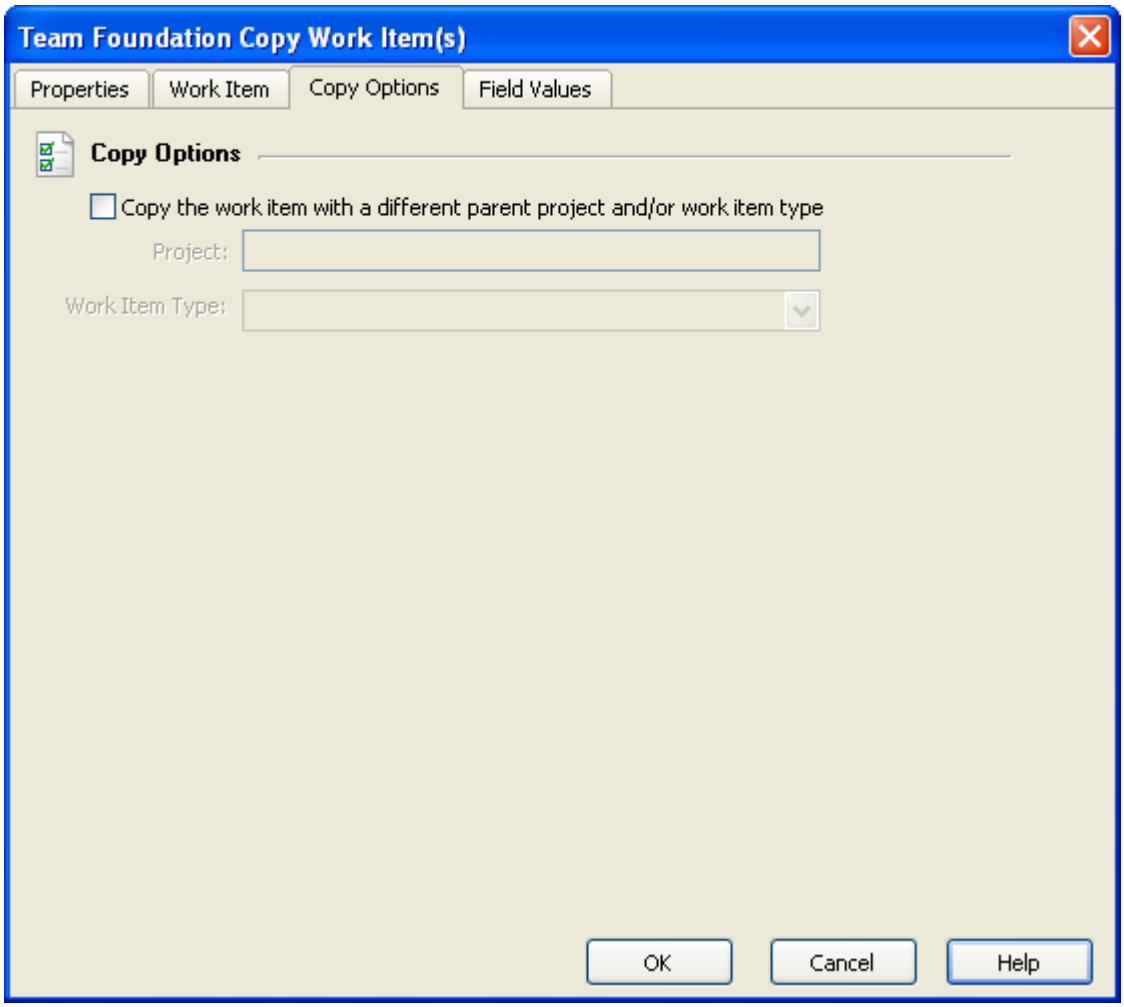
Work Item Property Page

Use this tab to select the Work Item(s) to copy. See the Team Foundation Work Items overview topic.

Field Values Property Page

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.5.12 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.

5.5.12.1 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

© 2001-2006 VSoft Technologies Pty Ltd

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

© 2001-2006 VSoft Technologies Pty Ltd

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

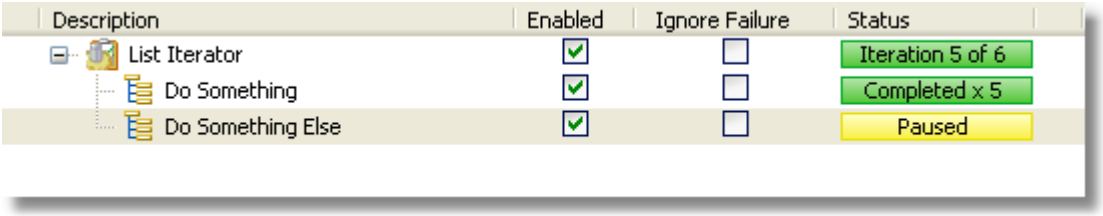
JediVCS Synchronize Project For Label
Properties
property: OverrideGlobalOptions Type: boolean Default: none
property: OverrideServerName Type: string Default: "localhost"
property: OverridePort Type: integer Default: 2106
property: OverrideUserName Type: string Default: none
property: OverridePassword 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

© 2001-2006 VSoft Technologies Pty Ltd

5.6 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". When the action executes the first time, it first figures out how many items there are. Typically the action will then set a variable of your choice to the first item and then run the child actions. So, for example, the variable "MyLetter" would be set to "A" and then the child actions are executed. Then it will set the variable to the next item in the list and run the children again - this continues for as many items in the list there are. Below is a screen shot of an iterator after the build has finished.



(The list iterator in the above screenshot has been paused during the fifth of six iterations.)

Iterator Actions

- 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.)

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 events these with Iterators.

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. If you wish to run some script before each iteration begins, the best way is to add a Run Script Action as the first child of the iterator.

© 2001-2006 VSoft Technologies Pty Ltd

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.

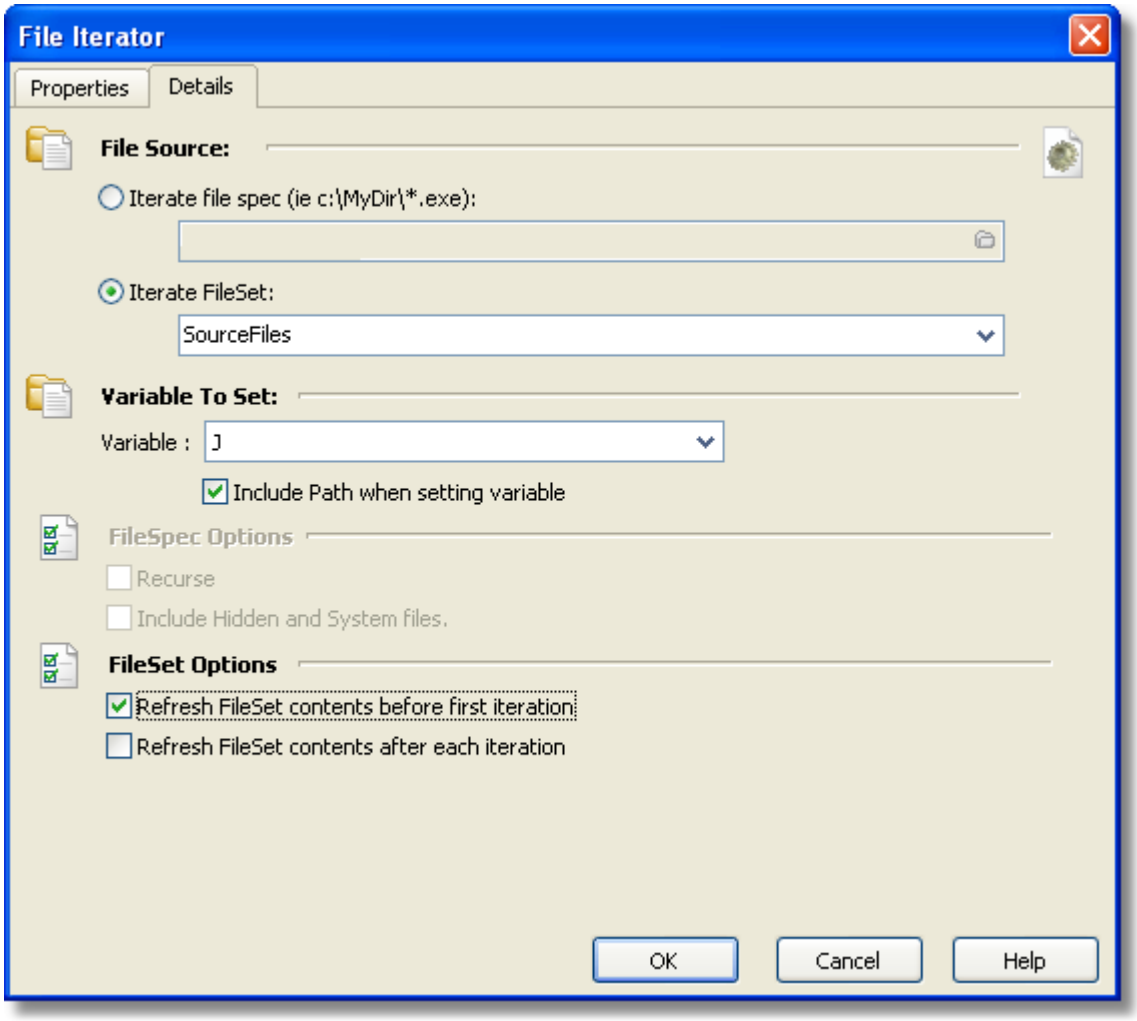
This is equivalent to having a Run Script action as the sibling immediately before the iterator.

© 2001-2006 VSoft Technologies Pty Ltd

5.6.1 File Iterator

The File Iterator Action allows you to iterate through 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 File Set (previously defined with the File Set Define action.) to iterate. Each file in the list will be iterated once.

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.

FileSpec Options

© 2001-2006 VSoft Technologies Pty Ltd

(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.

FileSet Options

(The 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.

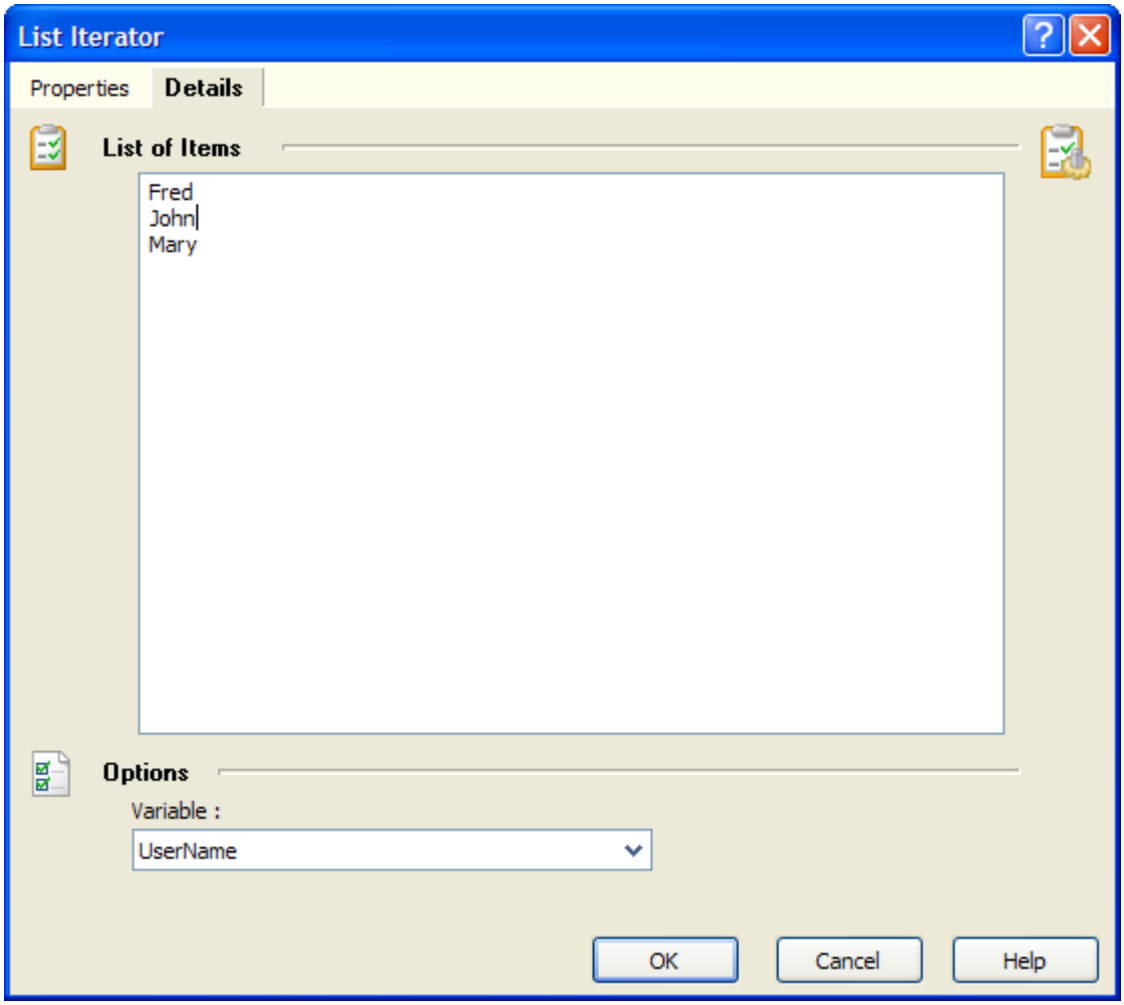
Warning: Refreshing the FileSet contents after each iteration may have unexpected results, if you are not extremely sure what is happening. Use with caution.

© 2001-2006 VSoft Technologies Pty Ltd

5.6.2 List Iterator

The List Iterator Action allows you to perform actions based on items in a list.

For general information on iterators, see the Iterators overview topic.



List of Items : This is the list of items to perform the action on. Each new line represents the new value.

FinalBuilder Variables (ie %ListContents%) can be used as part (or all) of the list.

Variable : The name of the FinalBuilder Project/User Variable to place each item in.

Scripting Info

You can set the list of items in script from the "**OnFirstRun**" event. Example script text:

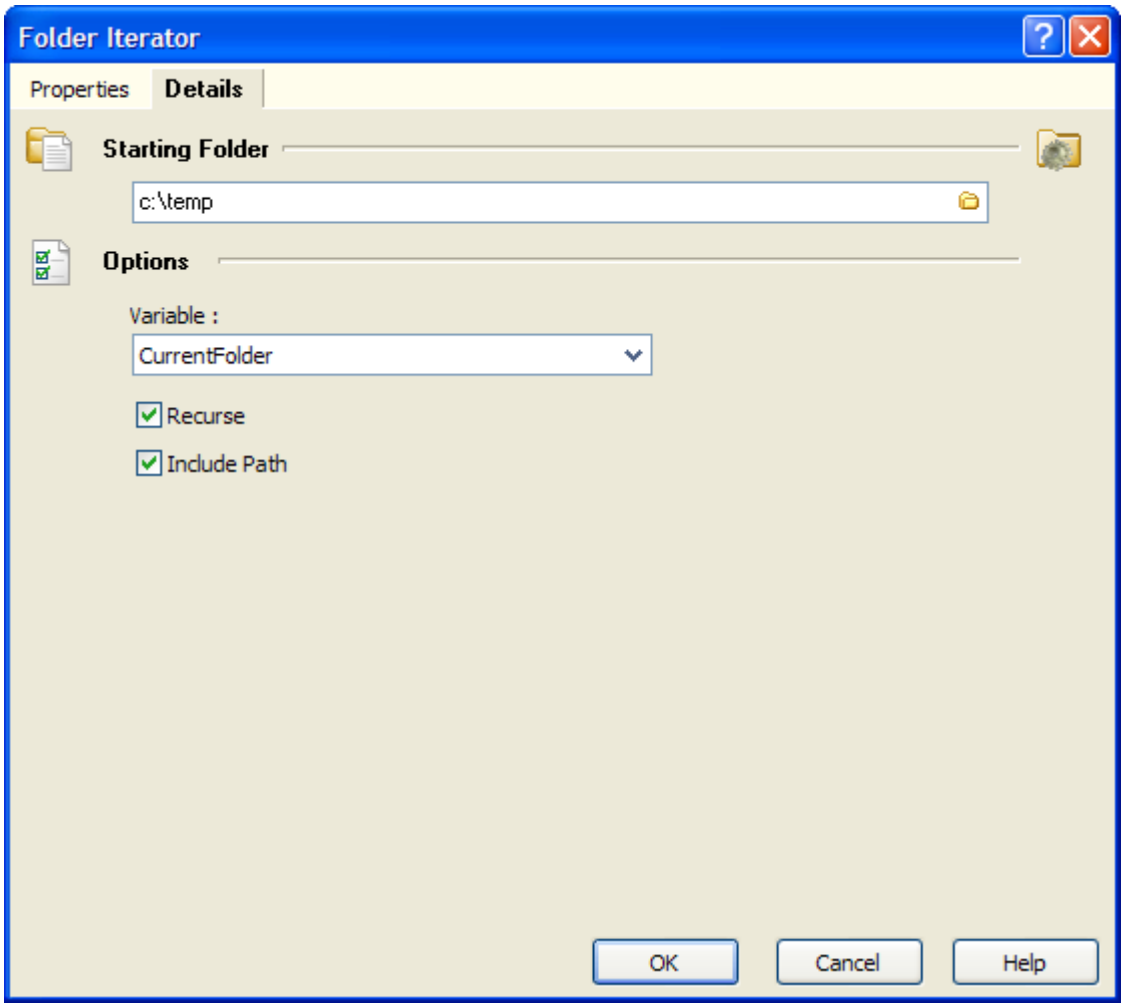
```
Action.ListOfItems.clear
Action.ListOfItems.Add("Item 1")
Action.ListOfItems.Add("Item 2")
```

© 2001-2006 VSoft Technologies Pty Ltd

5.6.3 Folder Iterator

The Folder Iterator action enables you to repeat a set of build steps 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 multiple 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"

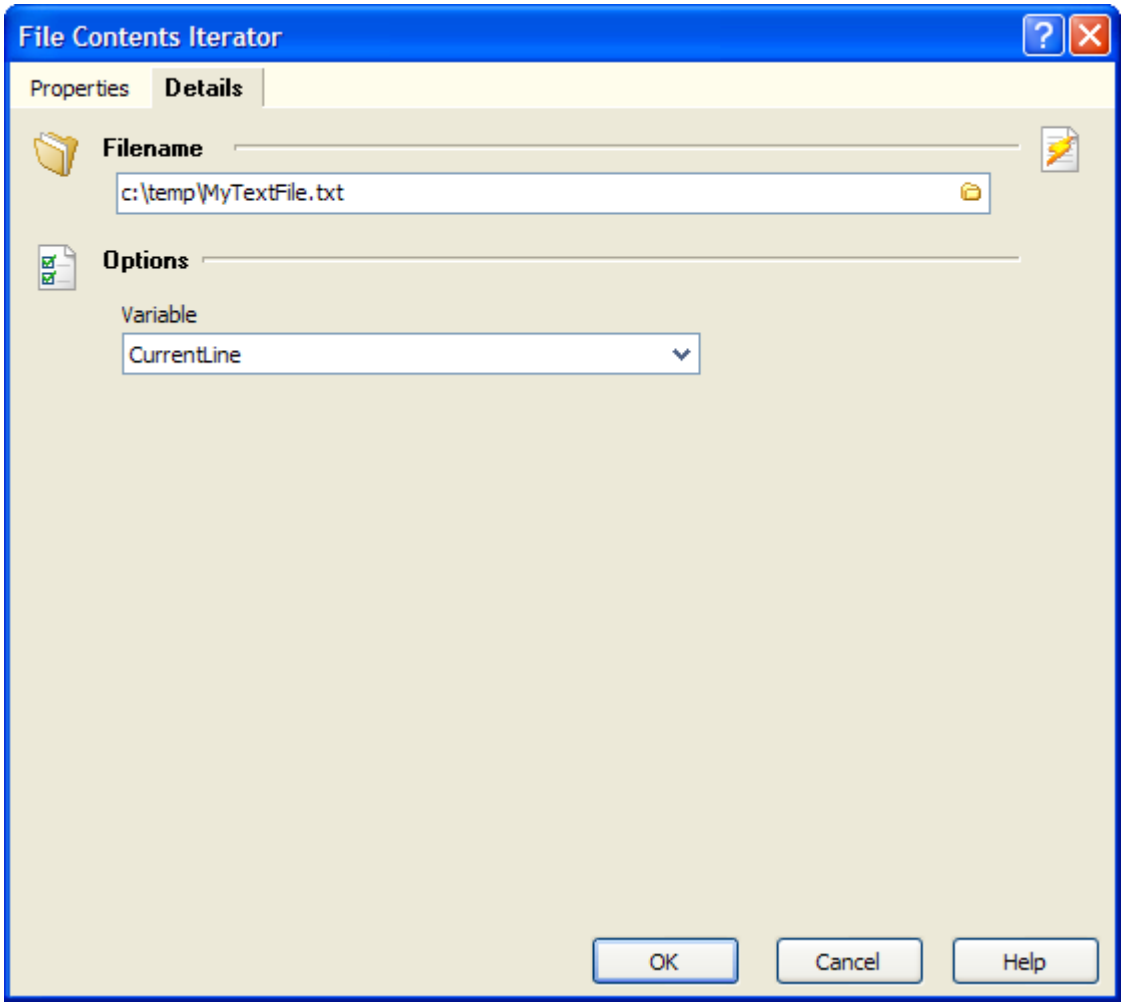
© 2001-2006 VSoft Technologies Pty Ltd

5.6.4 File Contents Iterator

The File Contents Iterator action enables you to perform a group of actions for each line of text read from a file

An example might be to read a file containing a list of project files; for each project file execute the compiler action.

(For general information on iterators, see the Iterators overview topic.)



Filename - the name of the text file to read

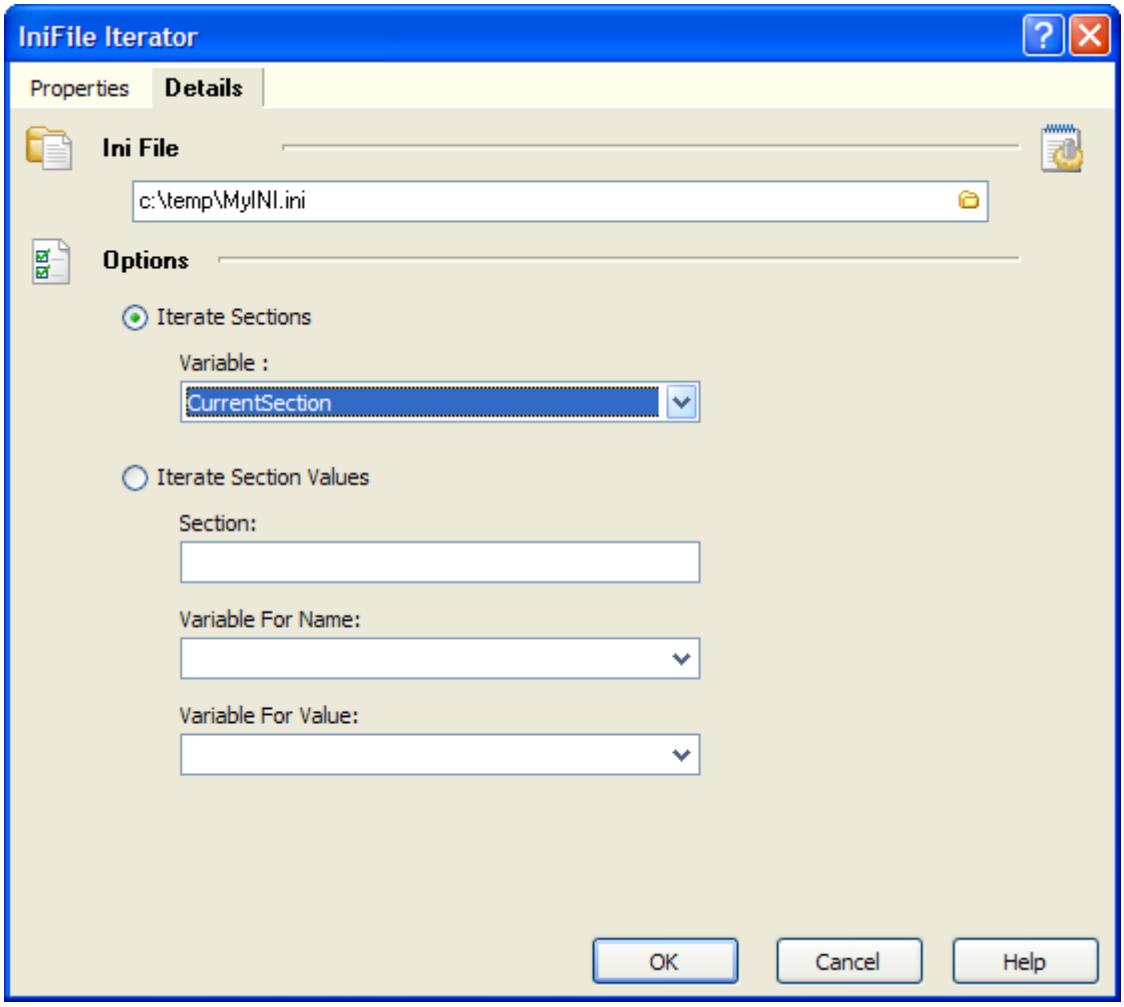
Variable - the FinalBuilder variable to place the current line of the file in for each iteration

© 2001-2006 VSoft Technologies Pty Ltd

5.6.5 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 as part of your build process.

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

© 2001-2006 VSoft Technologies Pty Ltd

5.7 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.

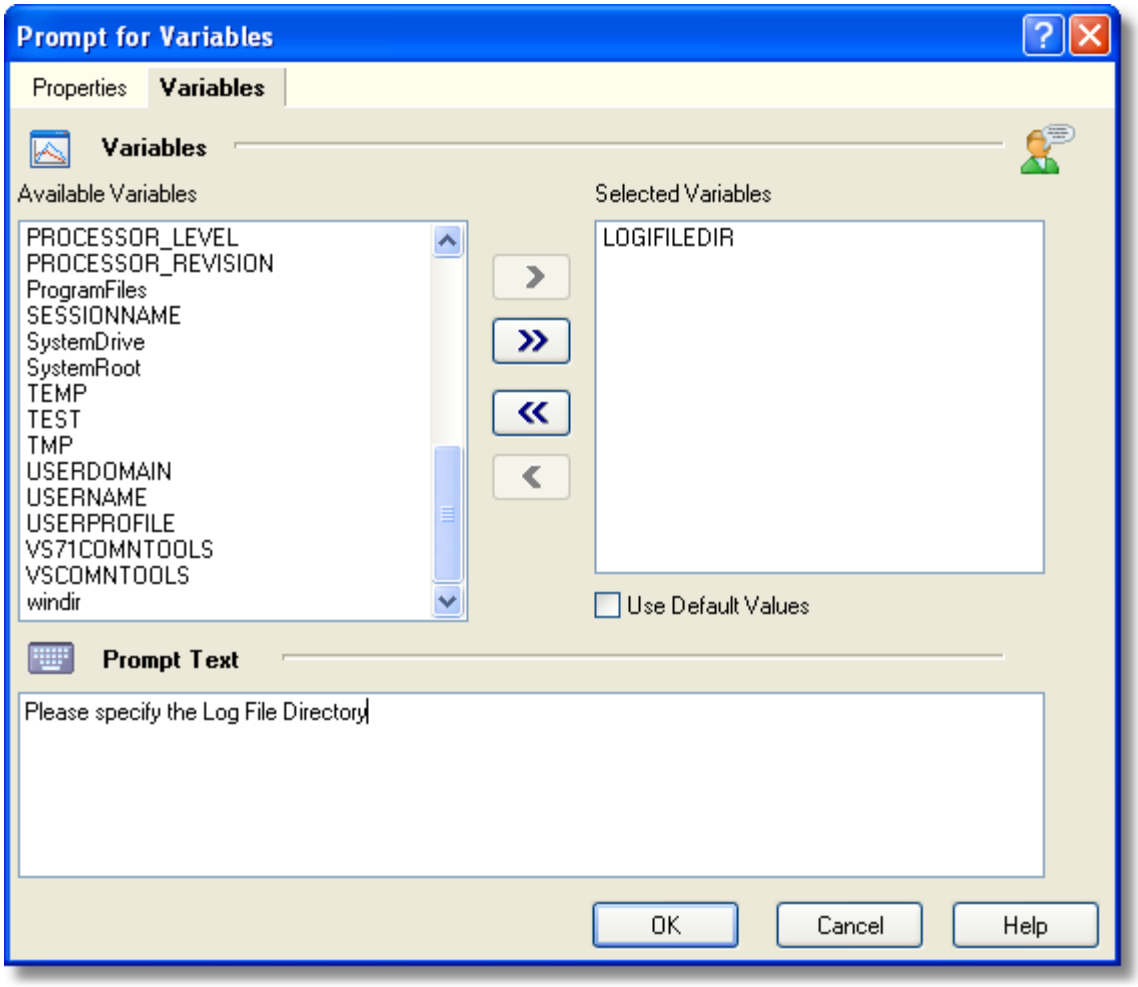
© 2001-2006 VSoft Technologies Pty Ltd

5.7.1 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.

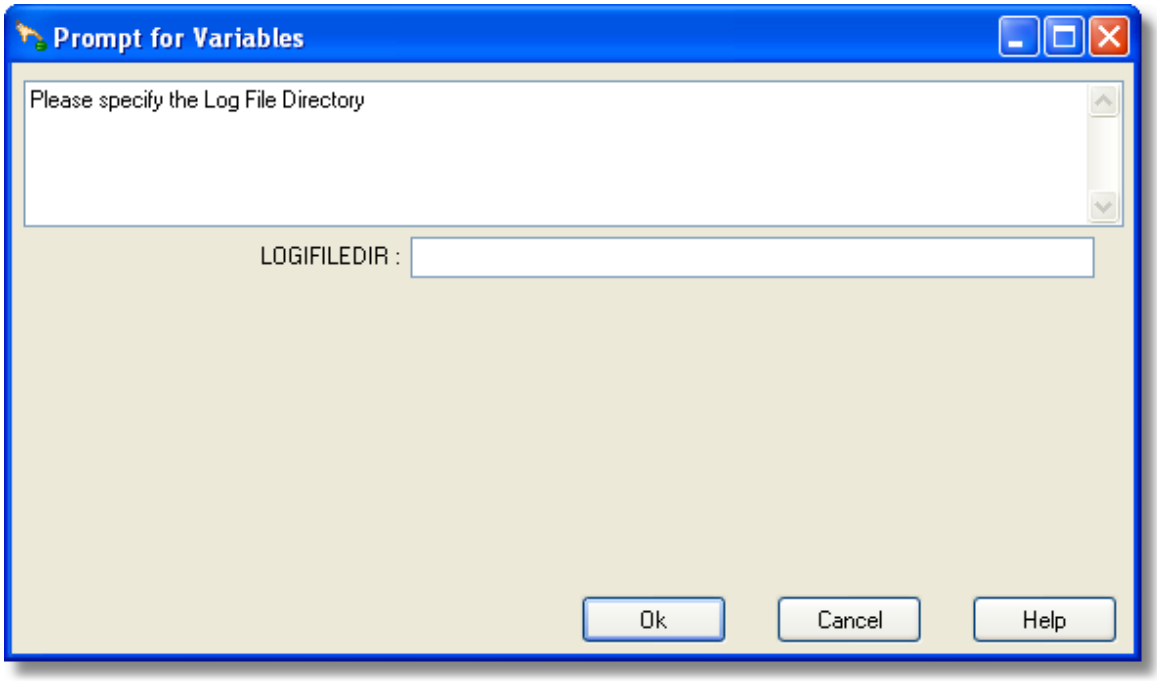
Note: This action should not be used in unattended builds as it will cause them to hang waiting for user input. By default, interactive actions are not available from command line console builds. Interactive actions cannot be used inside Async Action Groups.

For more sophisticated prompts requirements, use the Prompt For Variables (Enhanced) action.



At runtime the above configuration would cause this form to be displayed when the action executes.

© 2001-2006 VSoft Technologies Pty Ltd



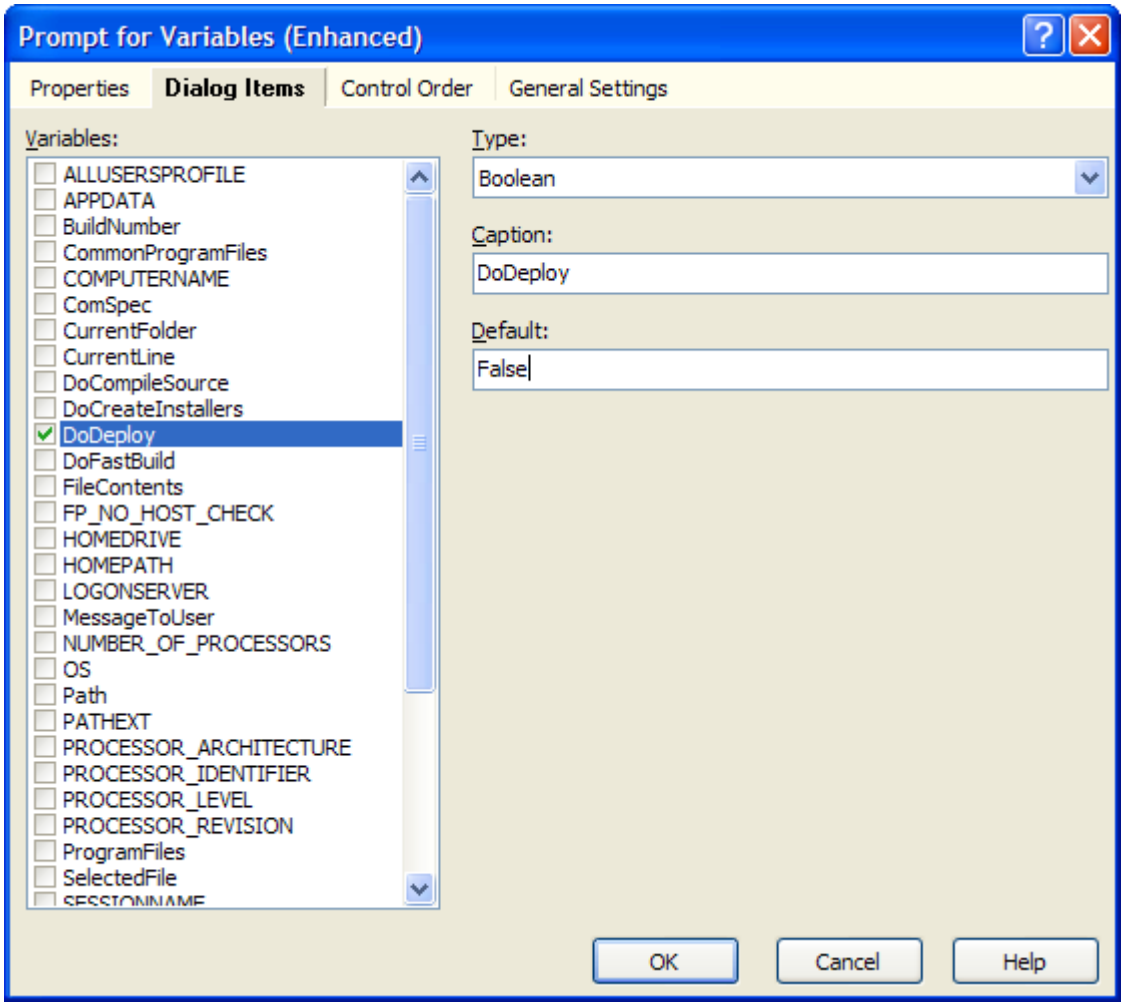
© 2001-2006 VSoft Technologies Pty Ltd

5.7.2 Prompt for Variables Action (Enhanced)

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.

For more information on variables, including creating your own project and user variables, see here.

Note: By default, interactive actions are not available from command line console builds. Interactive actions cannot be used inside Async Action Groups.

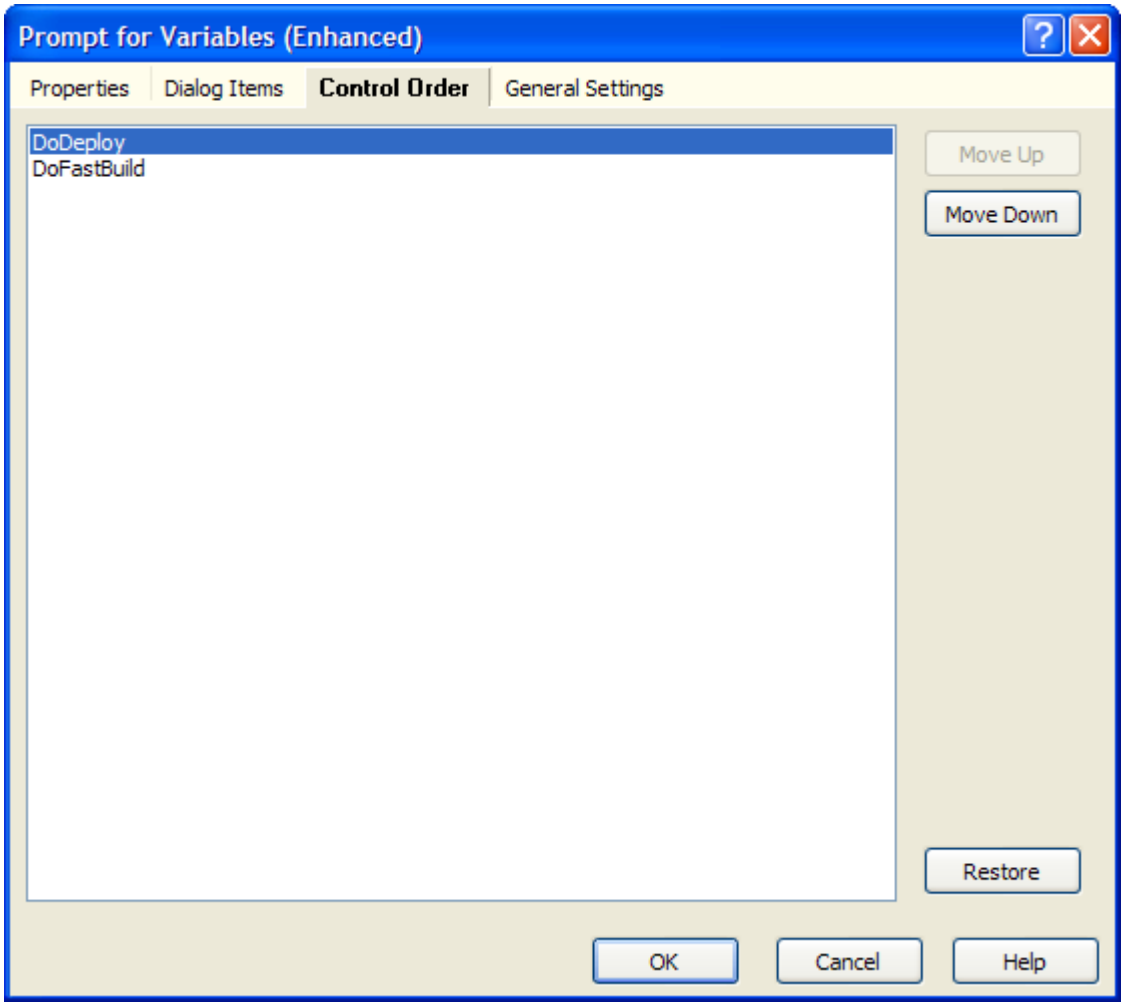


Values can have initial defaults (which can be variable values or fixed values.)

For "Check List" type prompts, specify default values as a comma-delimited list.

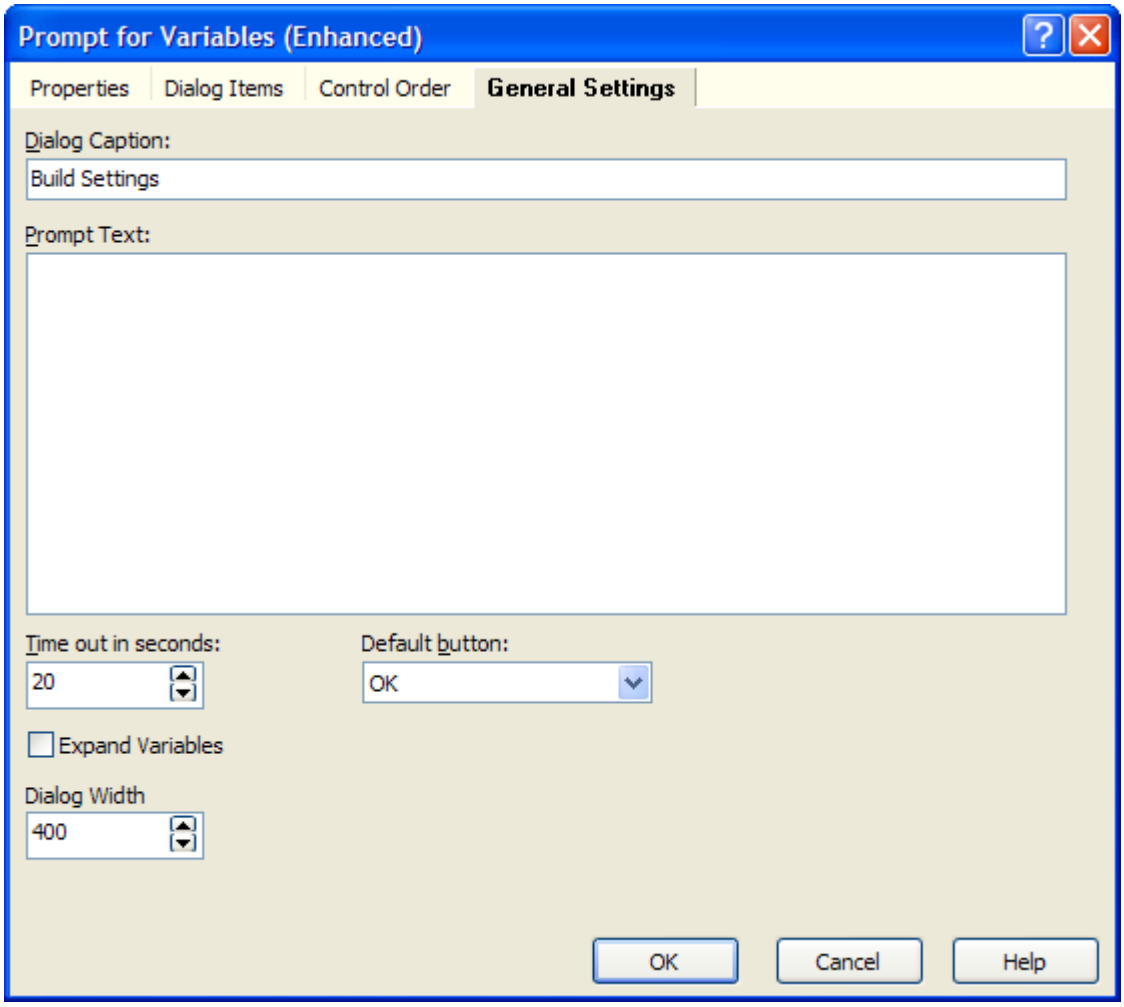
The action allows you to set the order of the various values:

© 2001-2006 VSoft Technologies Pty Ltd



The general settings lets you further customise the dialog:

© 2001-2006 VSoft Technologies Pty Ltd

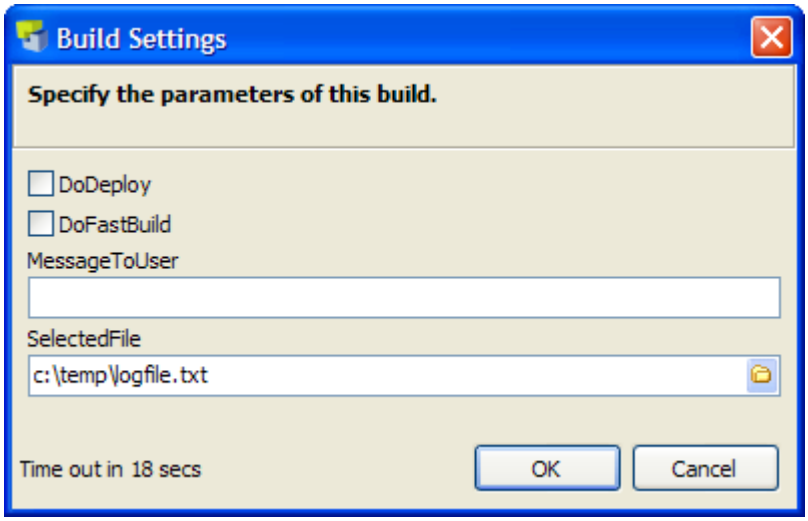


Time Out In Seconds - The dialog will automatically close if the user hasn't interacted with the dialog for the specified time

Default Button - This used for the timeout to choose a suitable action when the timeout is reached

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 build.

© 2001-2006 VSoft Technologies Pty Ltd

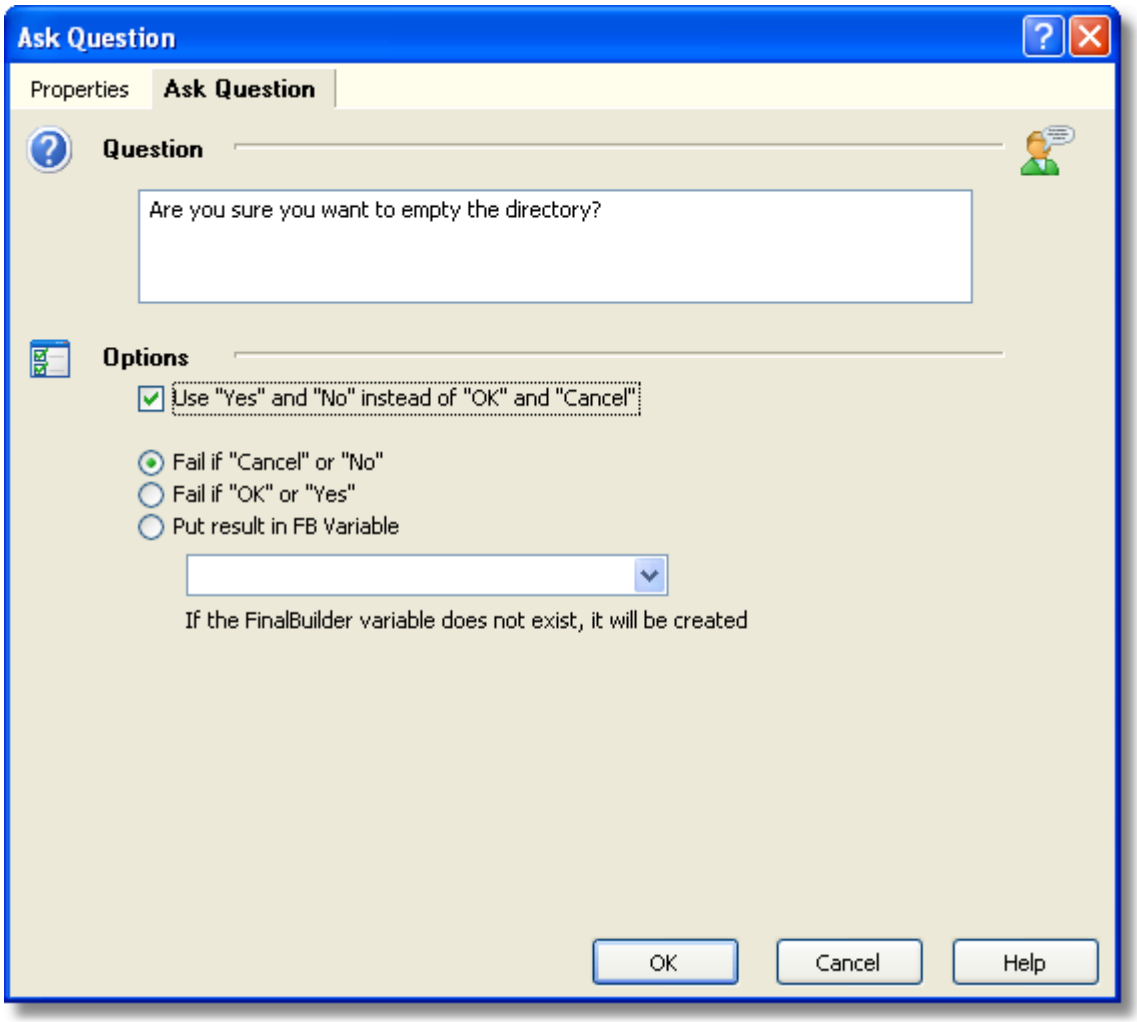


© 2001-2006 VSoft Technologies Pty Ltd

5.7.3 Ask Question Action

This action allows you to prompt the person running the build process. 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 in unattended builds as it will cause them to hang waiting for user input. By default, interactive actions are not available from command line console builds. Interactive actions cannot be used inside Async Action Groups.



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

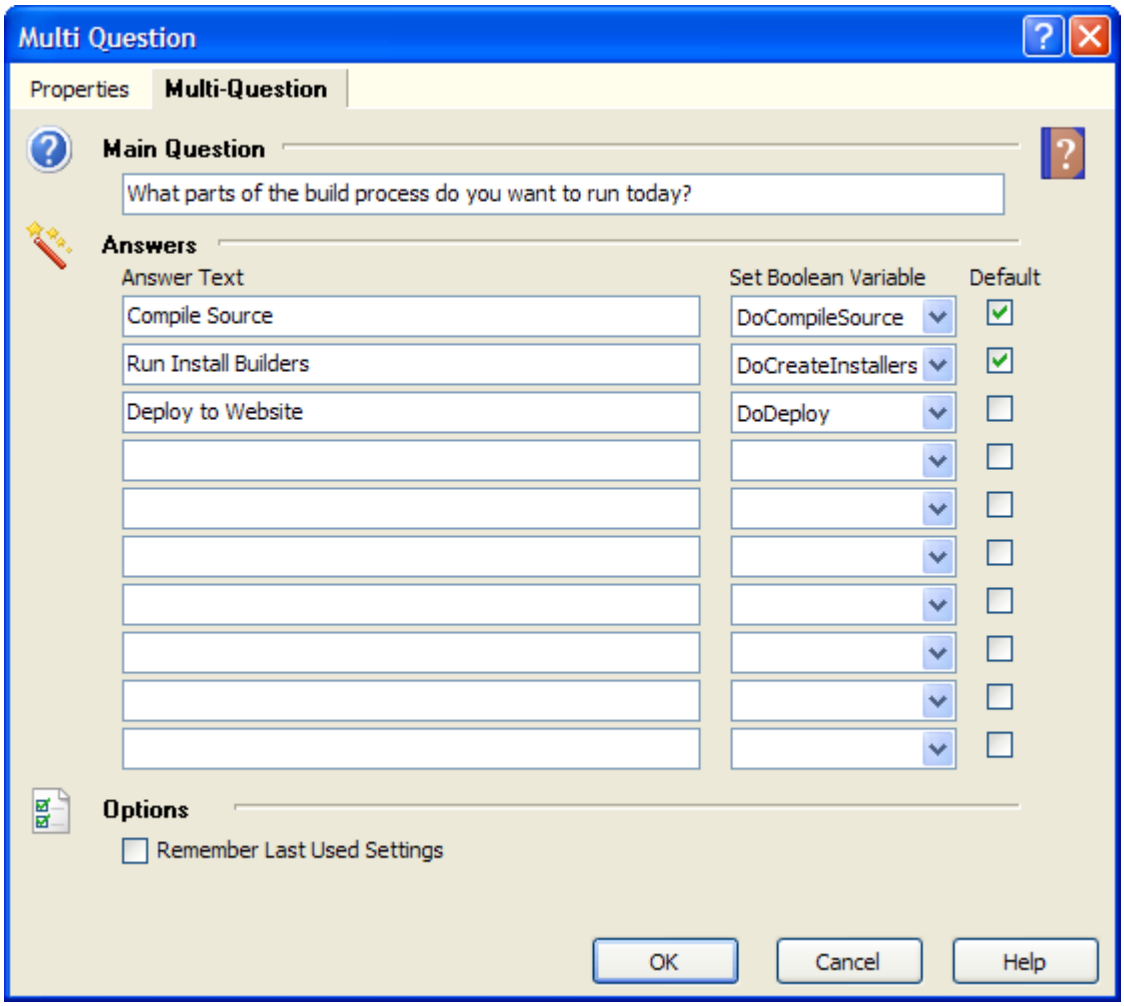
© 2001-2006 VSoft Technologies Pty Ltd

5.7.4 Multi Question

The MultiQuestion action enables you to interact with the user during your build process. 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 build process.

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 command line console builds. Interactive actions cannot be used inside Async Action Groups.

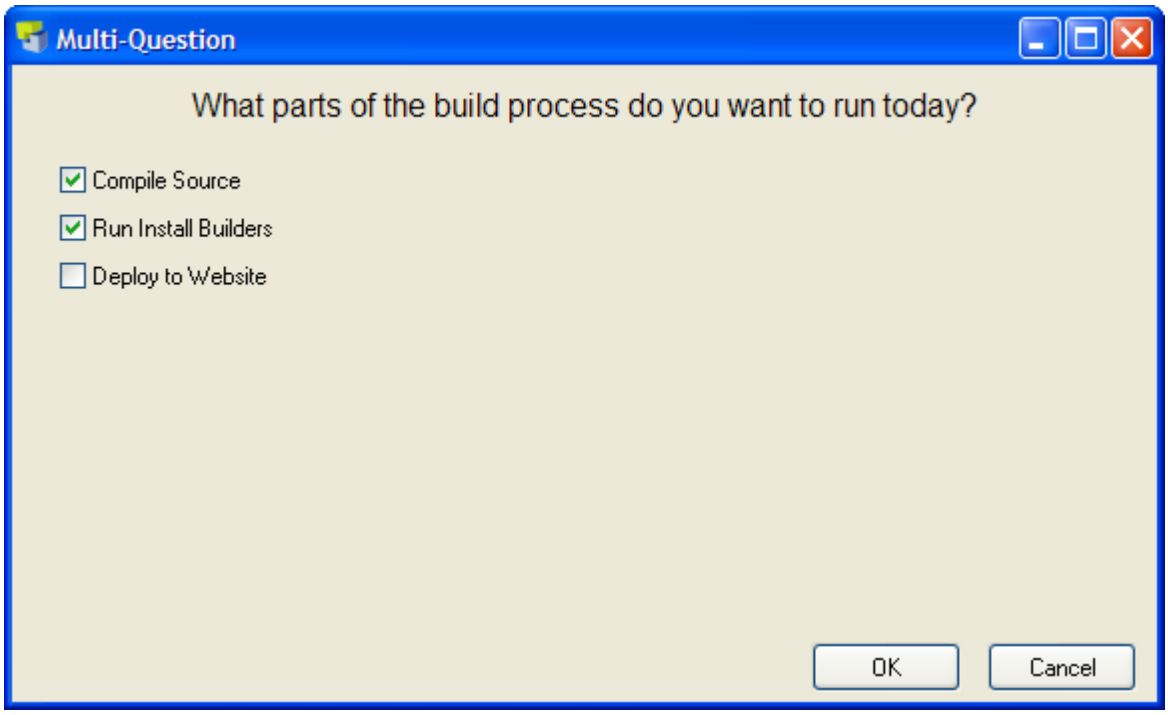


For Example, the main question could be "What parts of the build process do you want to run today". Answer 1: "Compile Source", Answer 2: "Run Install Builders", Answer 3: "Deploy to Website". The boolean answers can then be used in the Condition property of other actions to control whether they are 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:

© 2001-2006 VSoft Technologies Pty Ltd

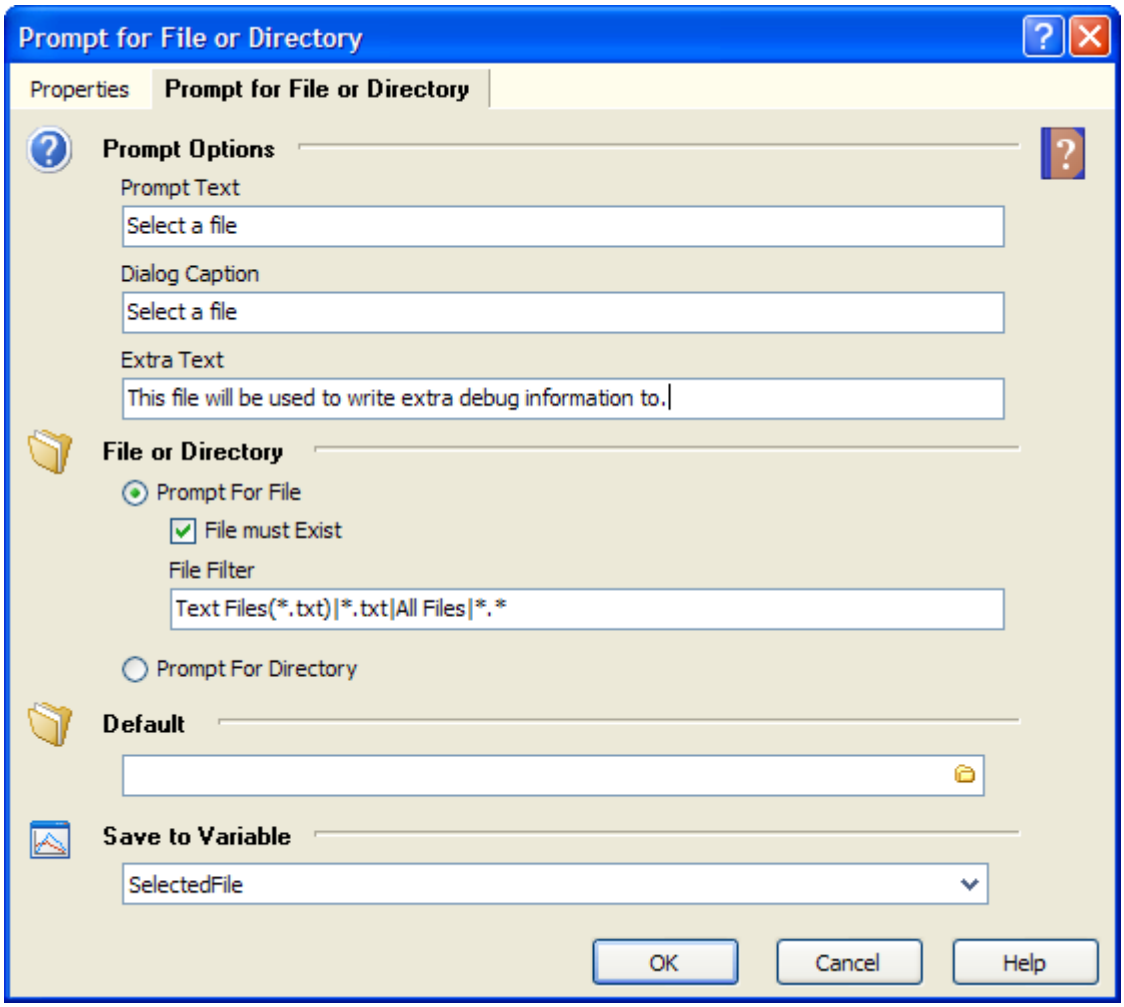


© 2001-2006 VSoft Technologies Pty Ltd

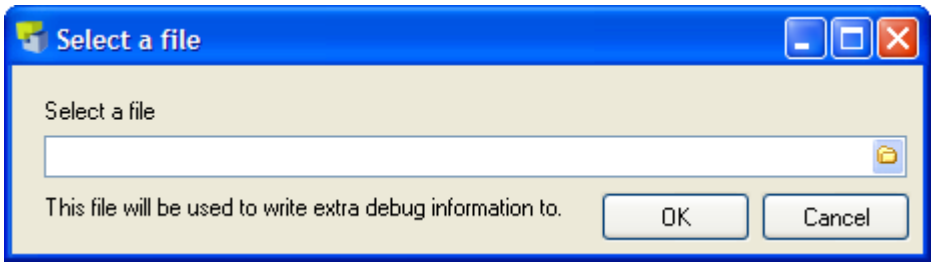
5.7.5 Prompt for File or Directory

The Prompt for File or Directory action enables you to ask the user to specify a file or directory during your build process. 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 command line console builds. Interactive actions cannot be used inside Async Action Groups.



When the action is executed it looks something like this:



© 2001-2006 VSoft Technologies Pty Ltd

5.7.6 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 command line console builds. Interactive actions cannot be used inside Async Action Groups.

This action was donated by Peter Thornqvist.

© 2001-2006 VSoft Technologies Pty Ltd

5.7.7 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 command line console builds. Interactive actions cannot be used inside Async Action Groups.

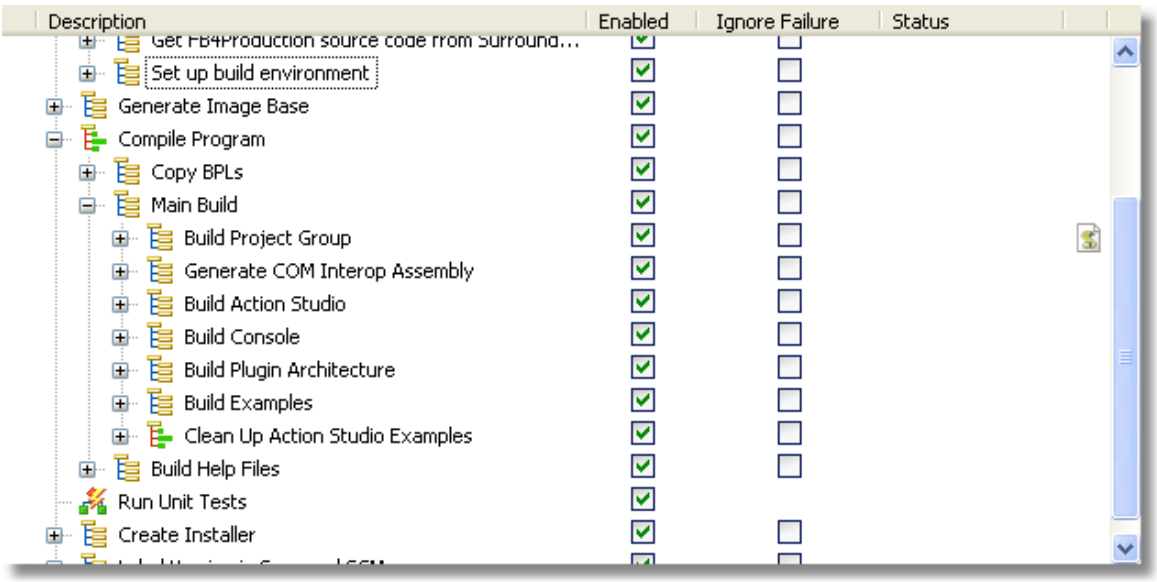
This action was donated by Peter Thornqvist.

© 2001-2006 VSoft Technologies Pty Ltd

5.8 Misc Actions

5.8.1 Action Group

The Action Group action type has no specific functionality, but is extremely useful for structuring build processes:



This action's BeforeAction and AfterAction 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.

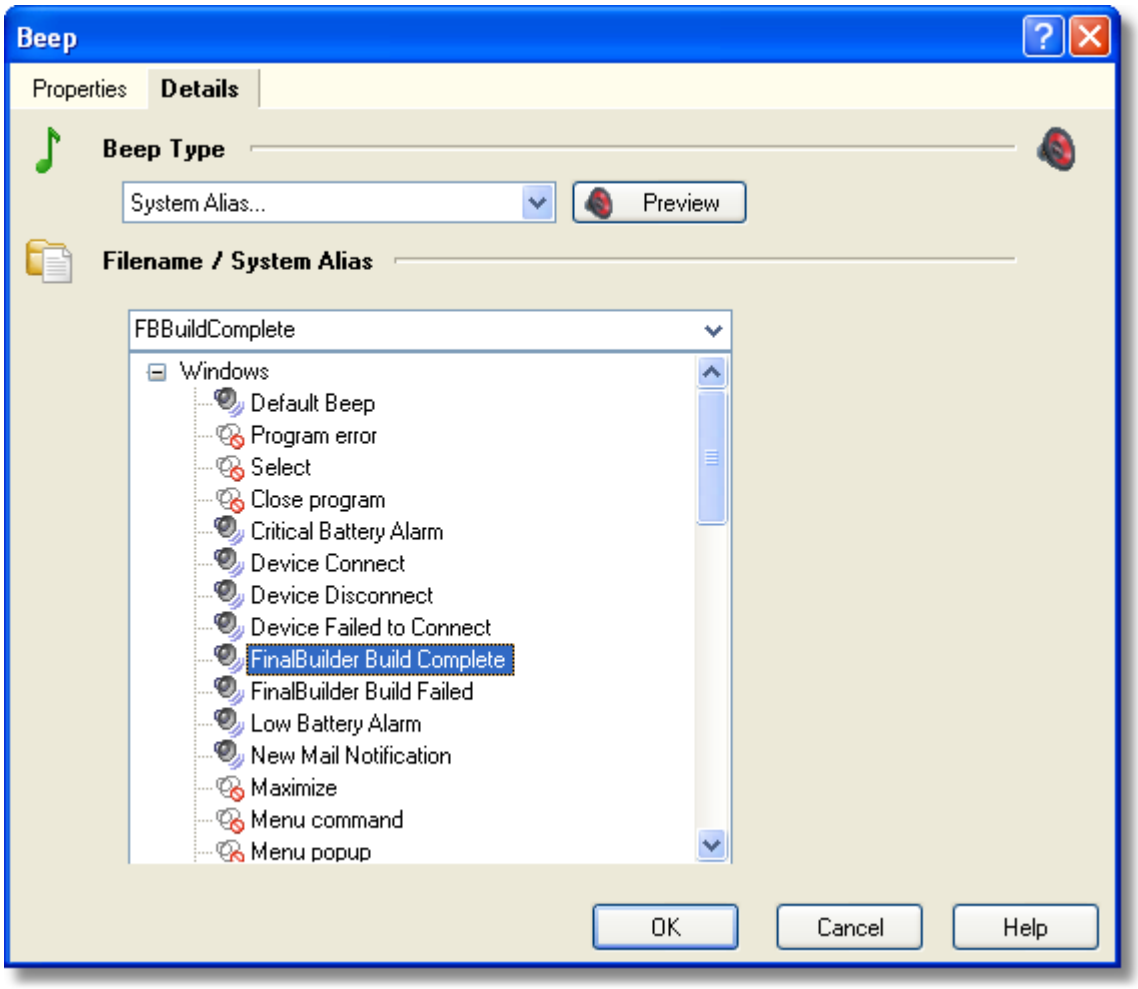
See also

Action Lists | Async Action Group | Include FinalBuilder Project Action

© 2001-2006 VSoft Technologies Pty Ltd

5.8.2 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.



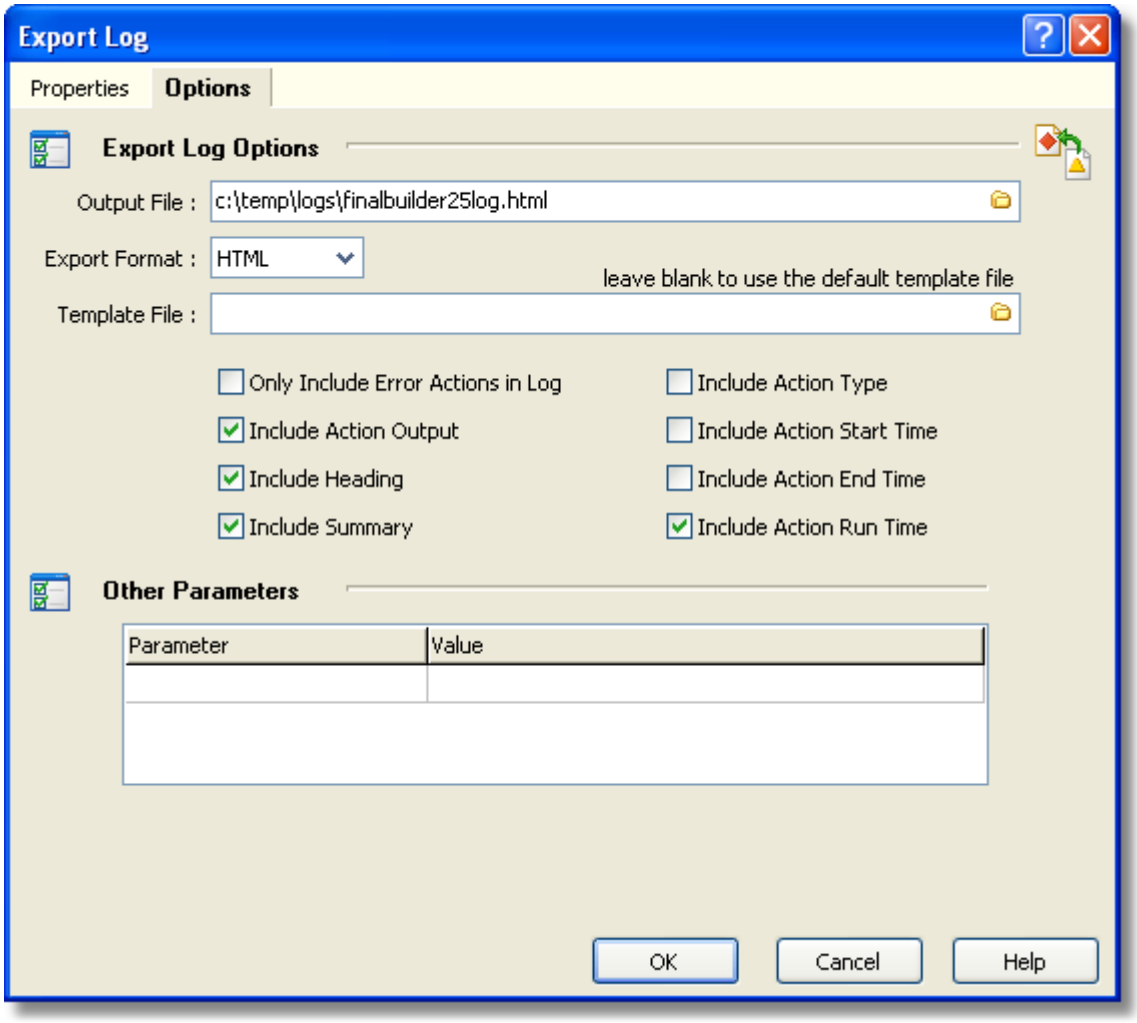
© 2001-2006 VSoft Technologies Pty Ltd

5.8.3 Export Log Action

This action allows you to export the FinalBuilder log file as part of your build process.

This enables the log to be sent as an attachment via email, stored, further processed, etc.

You can choose to export the current build, in Text, HTML or XML formats.



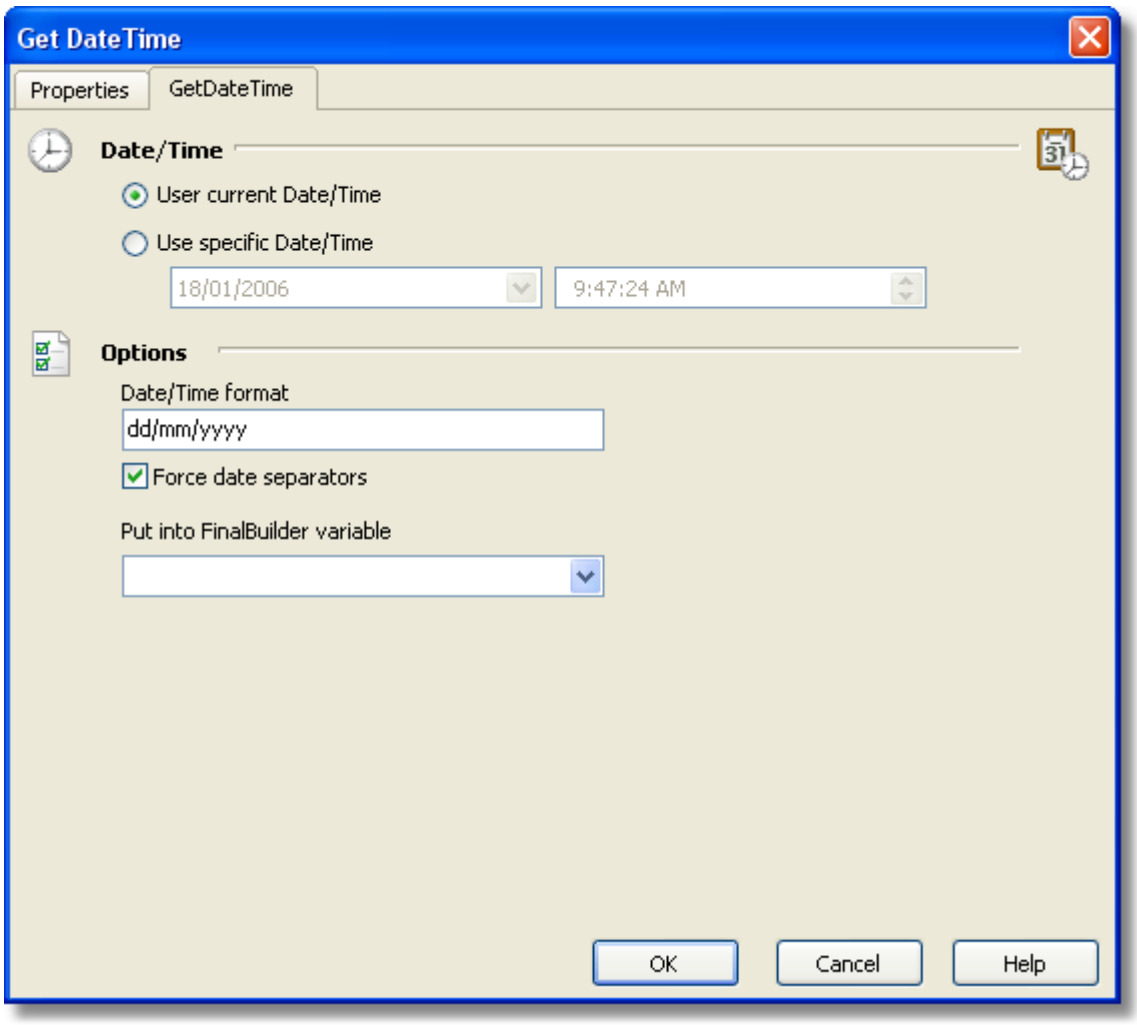
You can also specify a different XSL Stylesheet template file to alter the output format of the log file. The "Other Parameters" allow you to pass variables to your XSL Stylesheet.

Not that the default options for the log can be set in the FinalBuilder options dialog.

© 2001-2006 VSoft Technologies Pty Ltd

5.8.4 Get DateTime Action

This action copies a specific Date or Date/Time value into a FinalBuilder Variable.



Valid date format specifiers for the Format property :

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

d Displays the day as a number without a leading zero (1-31).
dd Displays the day as a number with a leading zero (01-31).
ddd Displays the day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd Displays the day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
ddddd Displays the date using the format given by the ShortDateFormat global variable.
dddddd Displays the date using the format given by the LongDateFormat global variable.
e (Windows only) Displays the year in the current period/era as a number

© 2001-2006 VSoft Technologies Pty Ltd

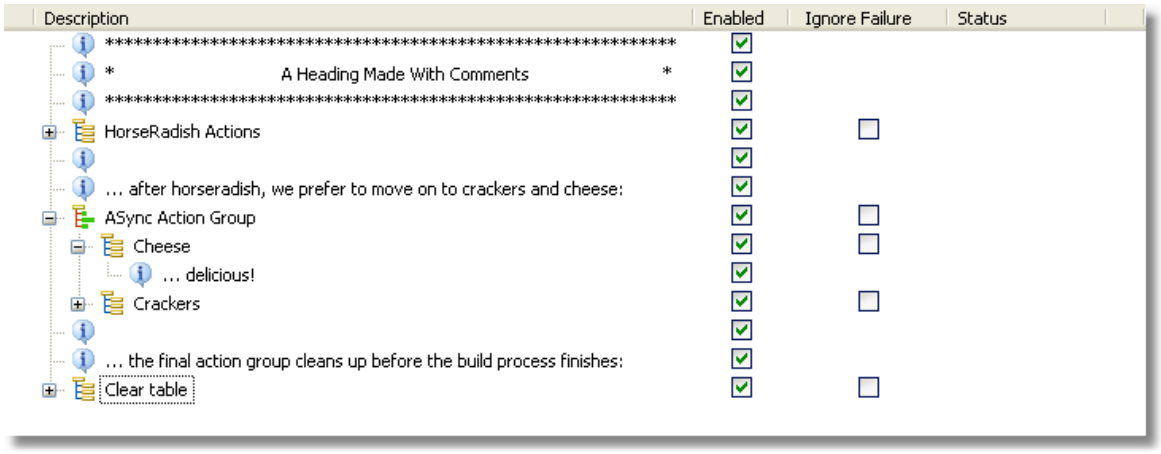
without a leading zero (Japanese, Korean and Taiwanese locales only).
ee (Windows only) Displays the year in the current period/era as a number with a leading zero (Japanese, Korean and Taiwanese locales only).
g (Windows only) Displays the period/era as an abbreviation (Japanese and Taiwanese locales 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.

The "**Force Date Separators**" option forces the format string to use literal '/' and ':' instead of the separators specified in the Regional Settings Control Panel.

© 2001-2006 VSoft Technologies Pty Ltd

5.8.5 Comment Action

Comment Actions provide a way to place descriptive text inside action lists.



Comments do not get executed at run time and do not have script events.

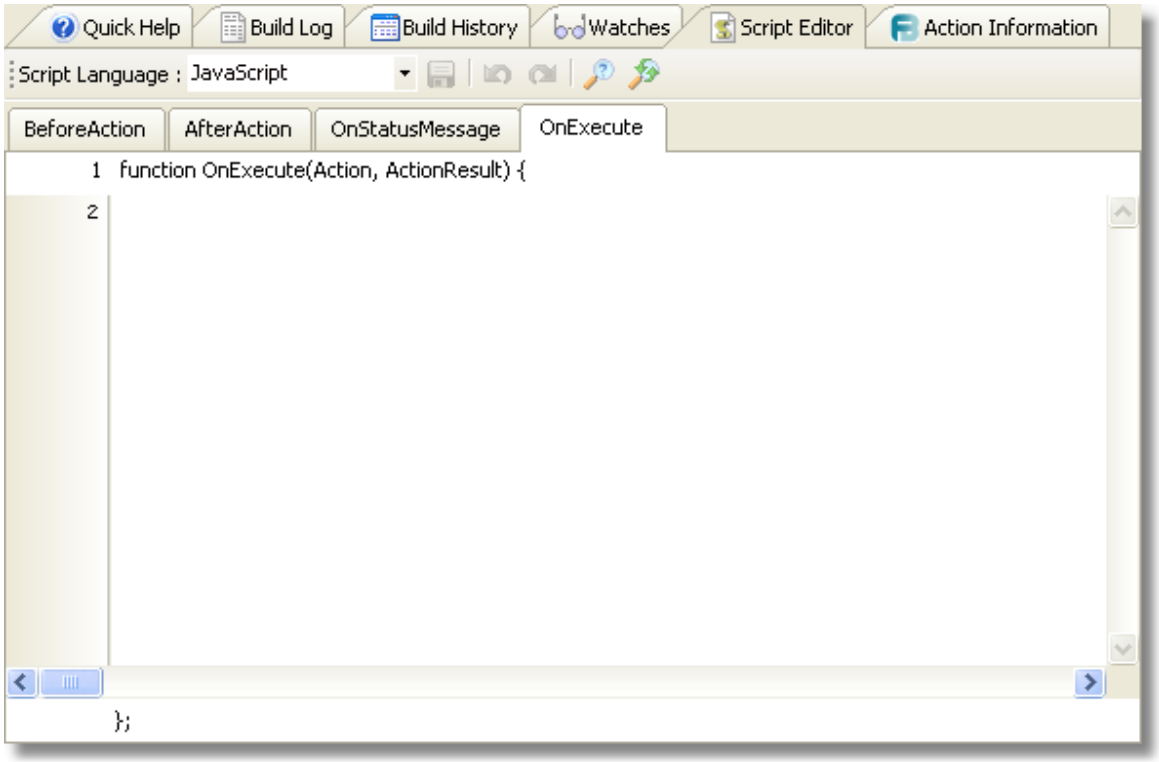
Action Groups can also be used to make your build easy to understand.

© 2001-2006 VSoft Technologies Pty Ltd

5.8.6 Run Script Action

The Run Script Action is a simple action that provides an extra script event, OnExecute, where you can place active script code:

(For more information on FinalBuilder scripting and action script events, see the Scripting in FinalBuilder overview topic.)

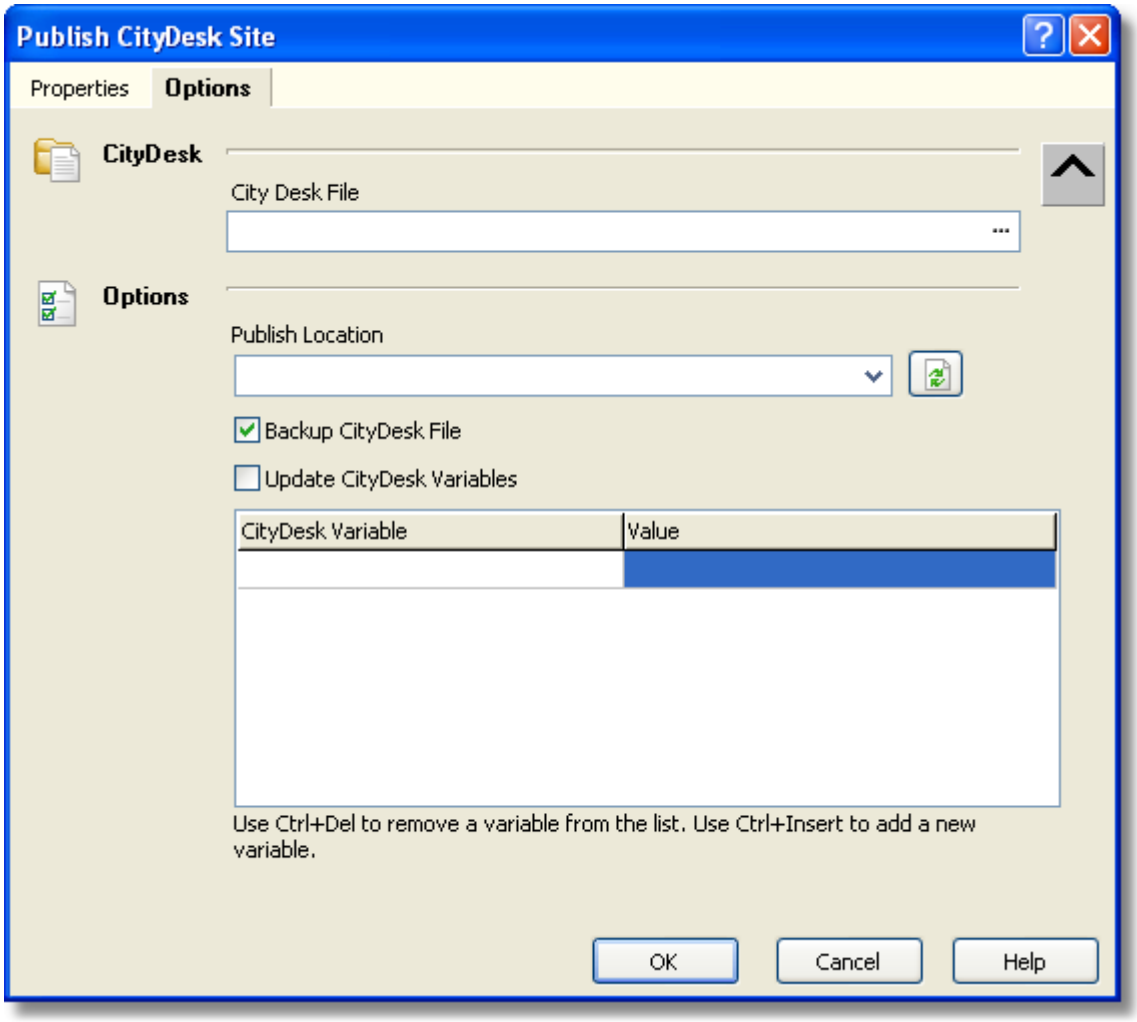


The OnExecute event has an extra boolean parameter, ActionResult, which determines the outcome of the action. ActionResult starts set to true. To fail the action, set ActionResult to false.

© 2001-2006 VSoft Technologies Pty Ltd

5.8.7 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!)

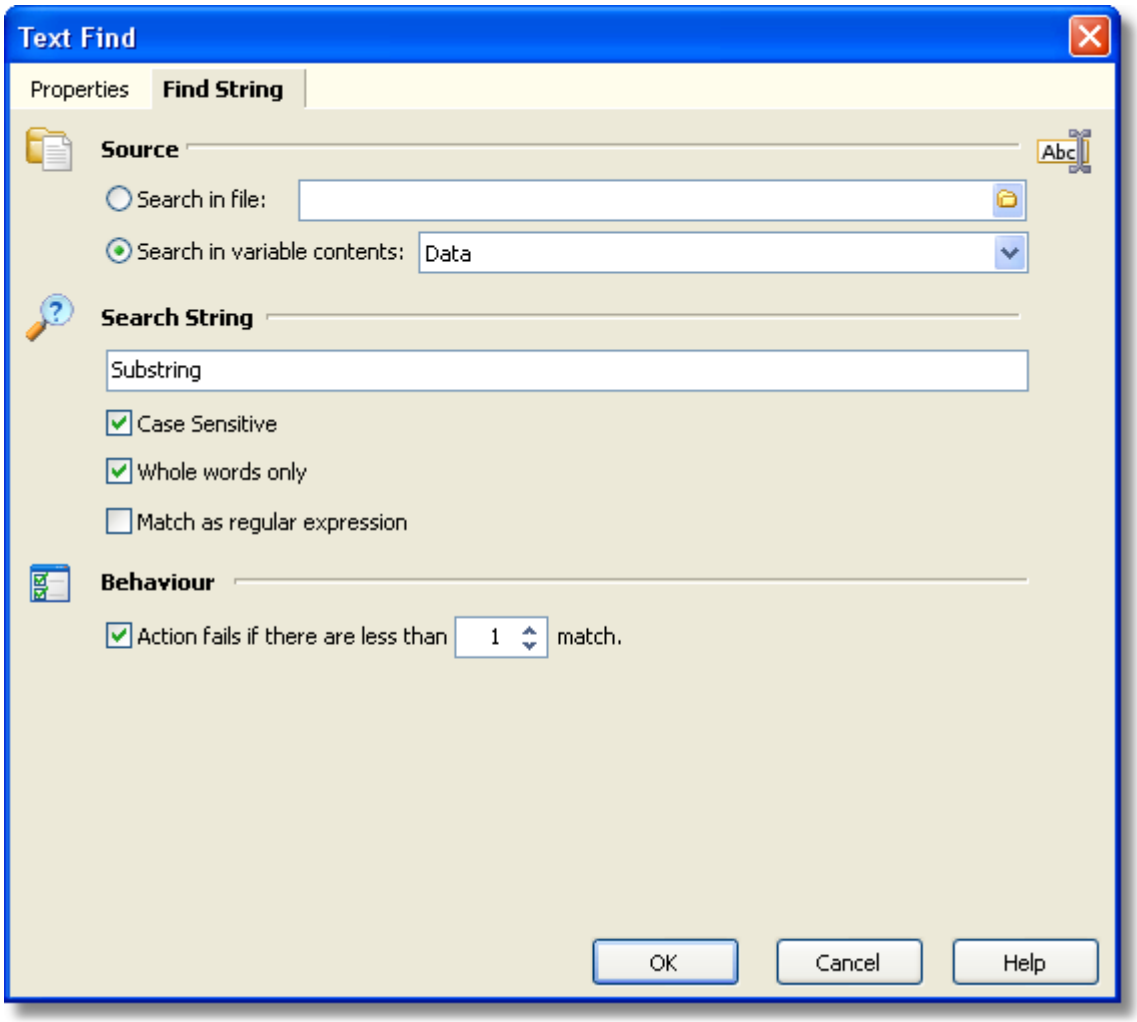
Note : CityDesk 2 Files are Microsoft Access databases, and this action uses the Microsoft Jet 4.x OLEDB provider. This provided is not installed in Windows by default. If you have Office XP/2003 or Access XP/2003 you should already have it, otherwise you can download it from the Microsoft web site.

© 2001-2006 VSoft Technologies Pty Ltd

5.8.8 Text Find Action

The Text Find action allows you search for a string (or regular expression) in a file or variable contents.

To replace string content, see the Text Replace Action.



Source

The text to search can be sourced from either a file or the contents of a variable.

Search String

Enter a string to search for. If searching for a plain string, there are options for case sensitivity and to match whole words only. Alternatively, search for a regular expression.

Case Sensitive

If checked, the search will be case sensitive.

© 2001-2006 VSoft Technologies Pty Ltd

Whole words Only

If checked, the search will only include whole words.

Match as Regular Expression

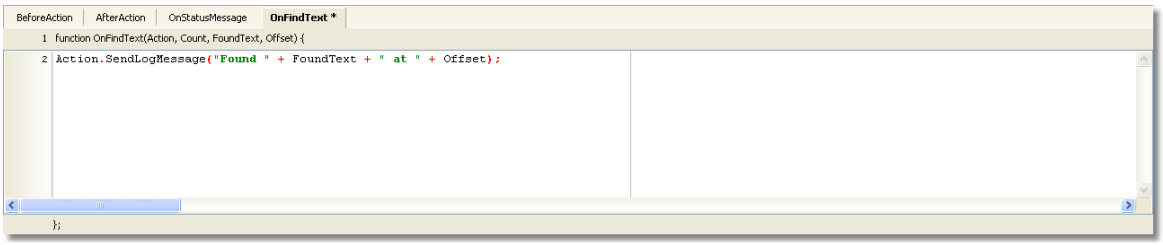
If checked, the search string will be treated as a regular expression. See the Regular Expression reference for details. If this option is set, the Case Sensitive and Whole Words Only options do not apply.

Behaviour

The action can be set to fail if there are less than a predefined number of matches.

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.

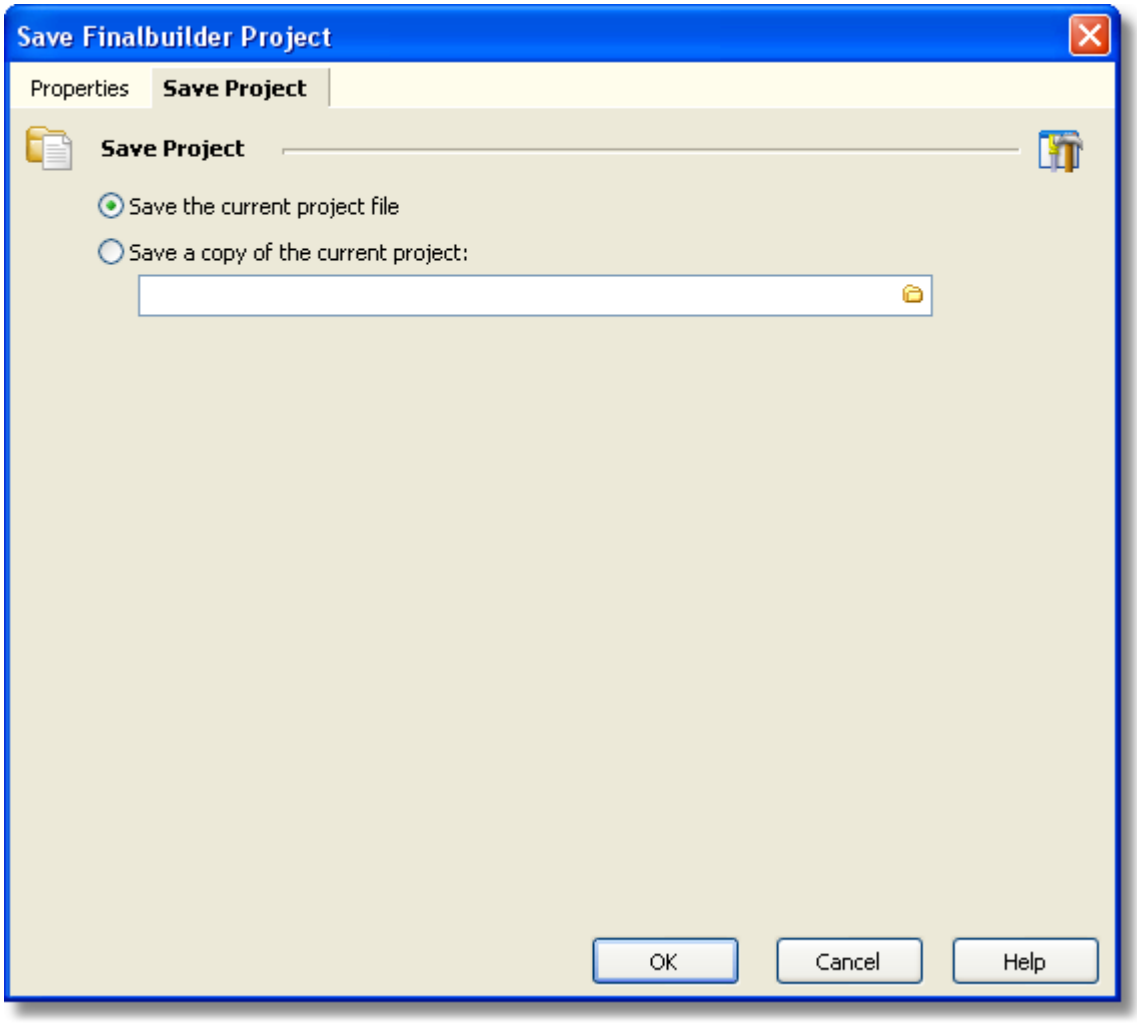
Offset
The offset in the source at which the text was found. *Offset* is zero-indexed.

© 2001-2006 VSoft Technologies Pty Ltd

5.8.9 Save FinalBuilder Project Action

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

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.

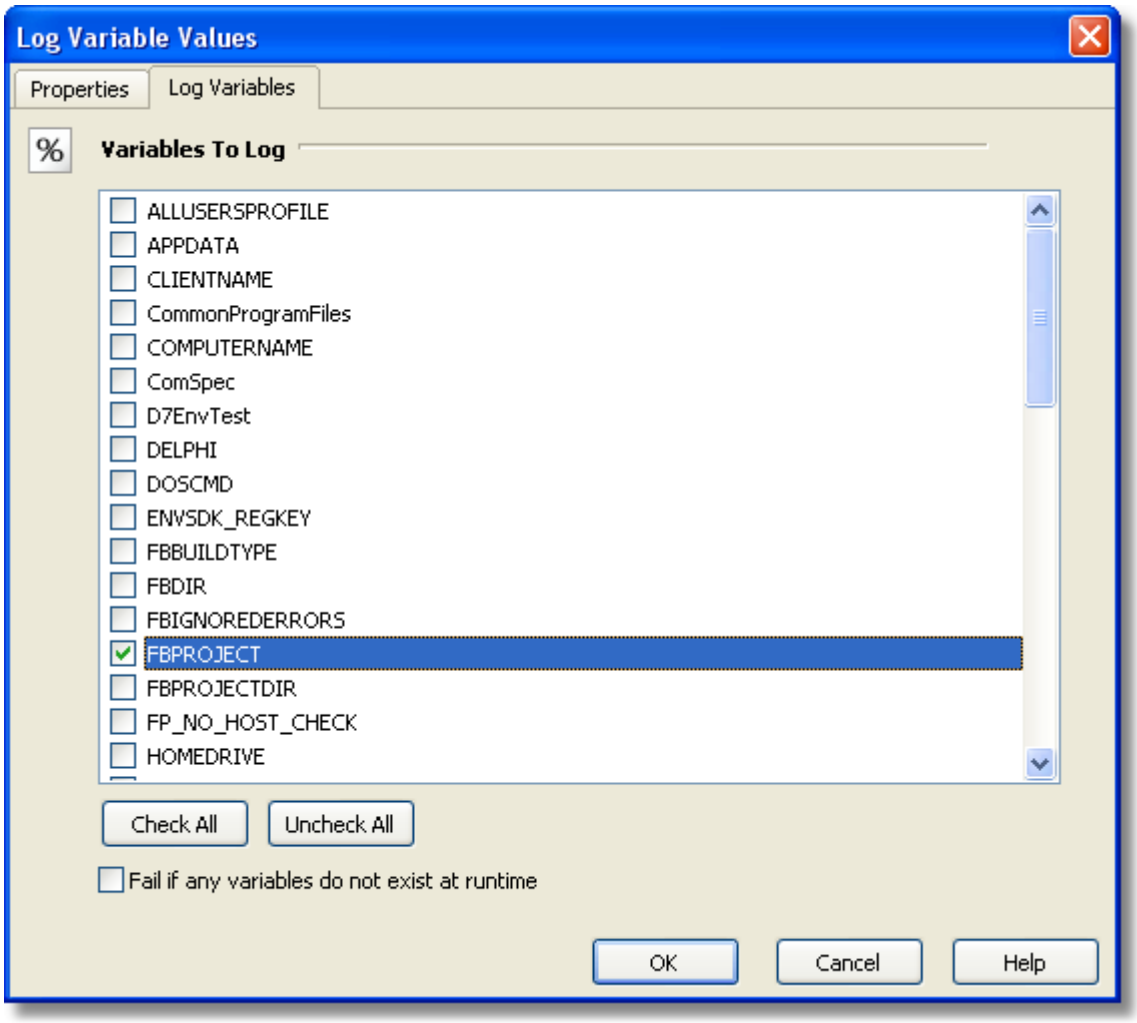
Save a copy of the current project

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

© 2001-2006 VSoft Technologies Pty Ltd

5.8.10 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 FinalBuilderprojects (see also Watches) or can be used to make log files easier to follow.



Variables To Log

Check the names of the variables for the action to log. Click the Check/Uncheck All buttons to quickly select or unselect 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 merely cause warnings to be displayed.

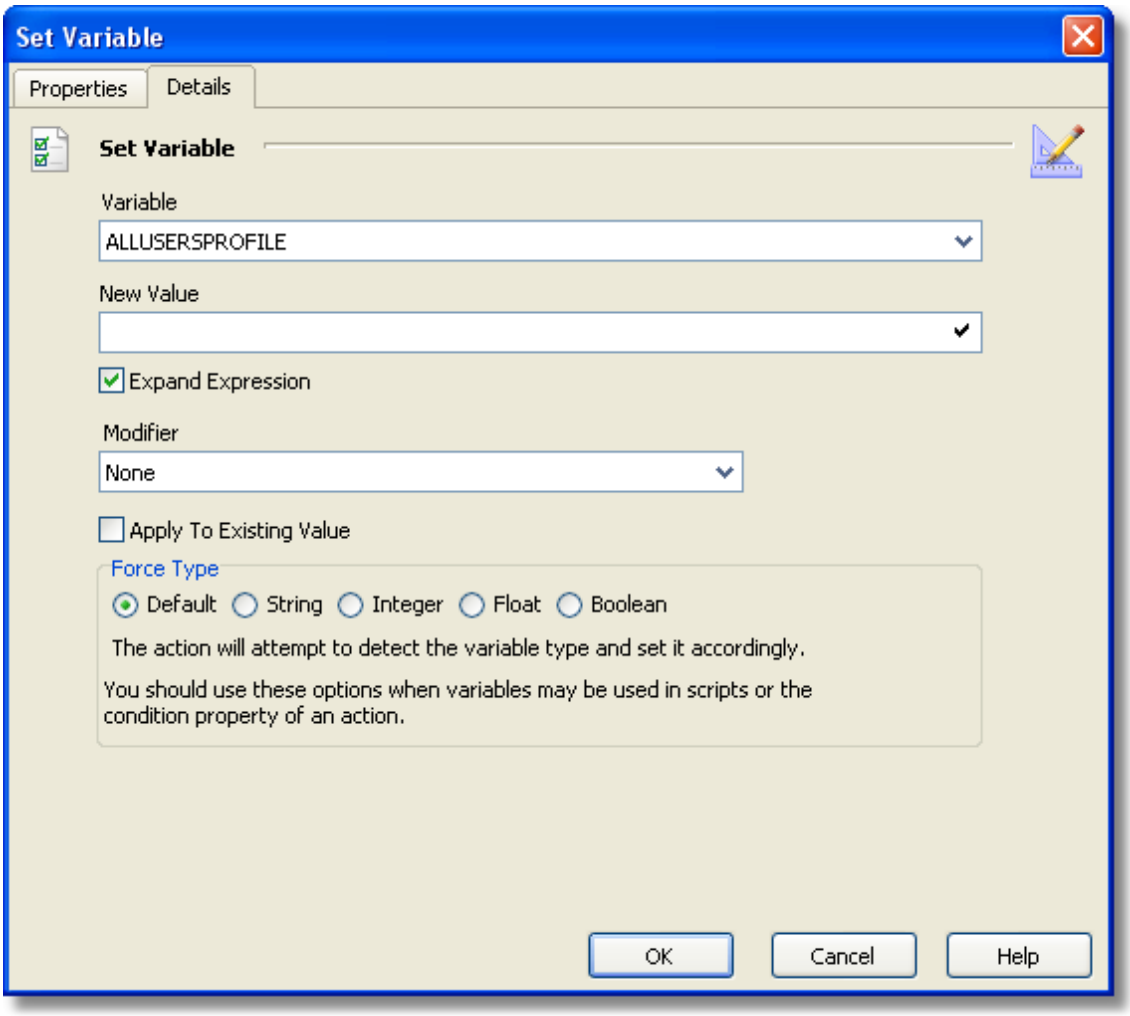
© 2001-2006 VSoft Technologies Pty Ltd

5.9 Variables

5.9.1 Set Variable Action

The Set Variable Action provides a means to set the value of a Project Variable or a User Variable without resorting to writing script. The value can include other variables. To append to the existing variable, simply prefix the new value with %VARIABLENAME% where variablename is the name of the variable who's value you are setting.

Note: Although Action List Parameters can be changed, they must be changed from script - not via the Set Variable action.



Variable

The name of a FinalBuilder variable to set.

New Value

The new value for the variable. Check the Expand Expression box to expand variable and other references in the value.

Modifier

A modifier is a function that can be applied to the new value (after any variables have been replaced.) The available modifiers are :

© 2001-2006 VSoft Technologies Pty Ltd

None - the default.
Trim
TrimLeft
TrimRight
IncludeTrailingBackSlash
ExcludeTrailingBackSlash
ExtractFileName,
ExtractFilePath
ExtractFileDrive
ExtractFileExt
ShortFileName
AddQuotes
AddDoubleQuotes
StripQuotes
Increment
Deincrement
LowerCase
UpperCase
Encrypt - Encrypts using blowfish with a hard wired key
Decrypt - Decrypts values previously encrypted with the Encrypt modifier.

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.

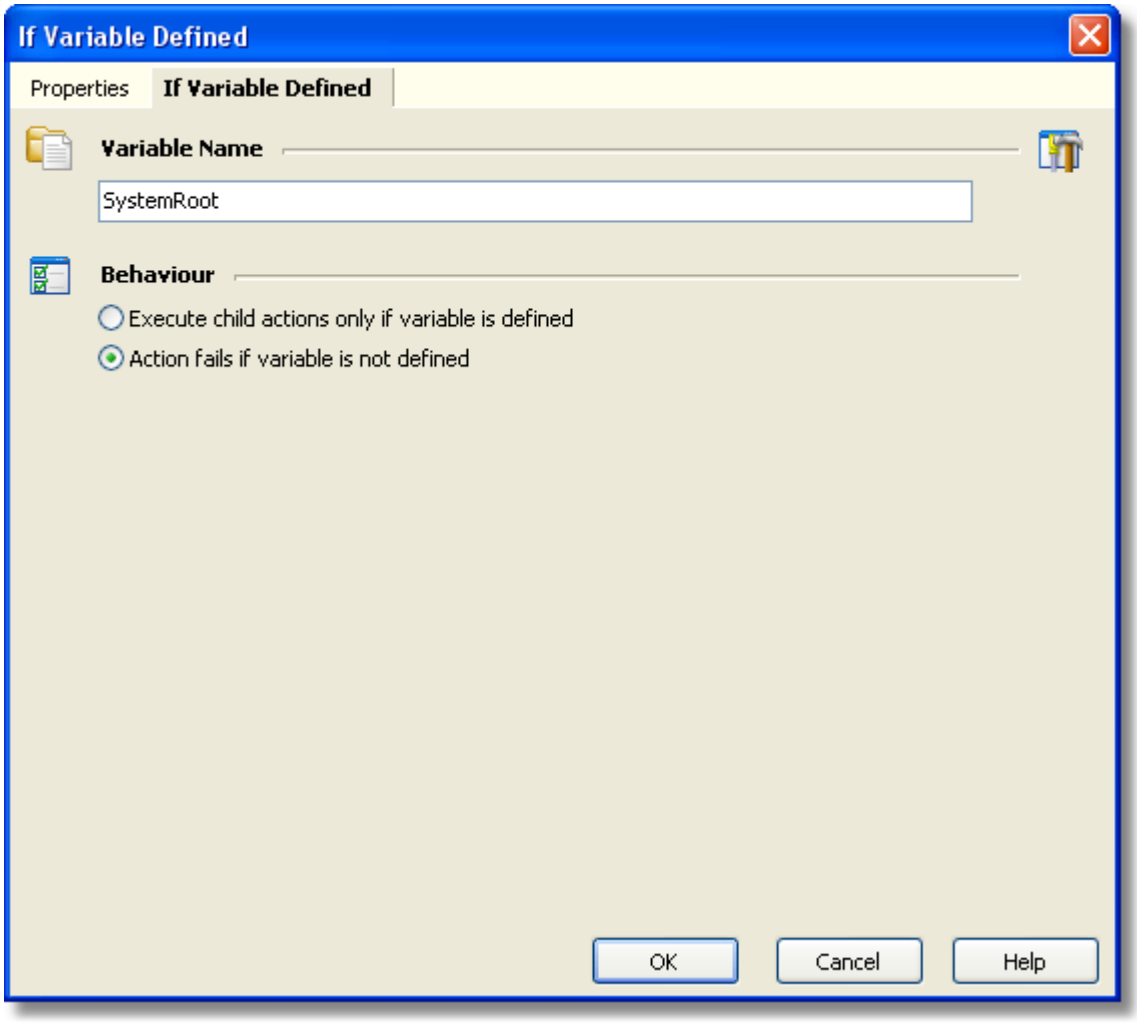
© 2001-2006 VSoft Technologies Pty Ltd

5.9.2 If Variable Defined Action

The If Variable Defined action enables you to control the flow of your build process depending
on whether a FinalBuilder variable is defined. If the variable is defined, then the child actions of the action will be executed.

Note: The action also tests for Windows Environment variables.

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.



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.

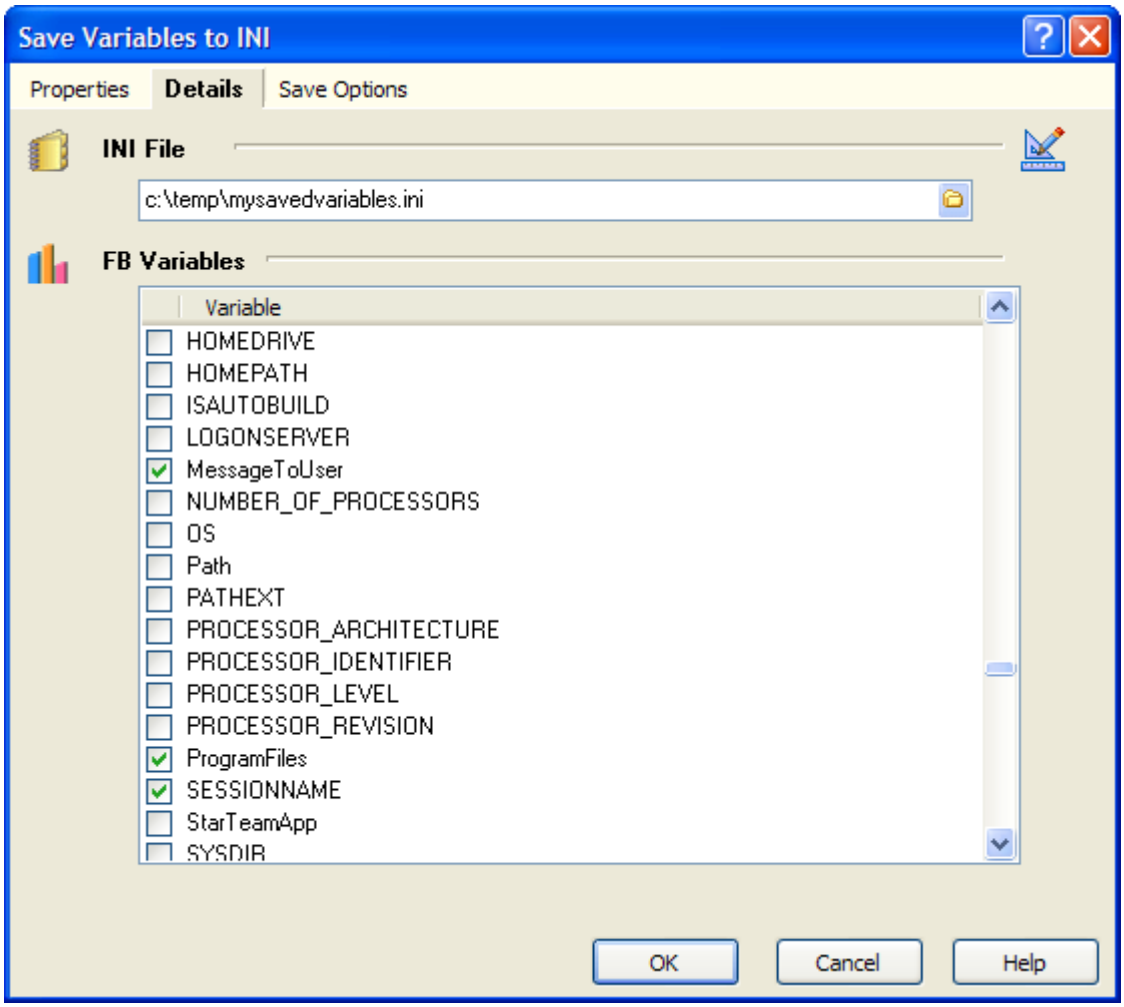
© 2001-2006 VSoft Technologies Pty Ltd

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.)

© 2001-2006 VSoft Technologies Pty Ltd

5.9.3 Save Variables To Ini

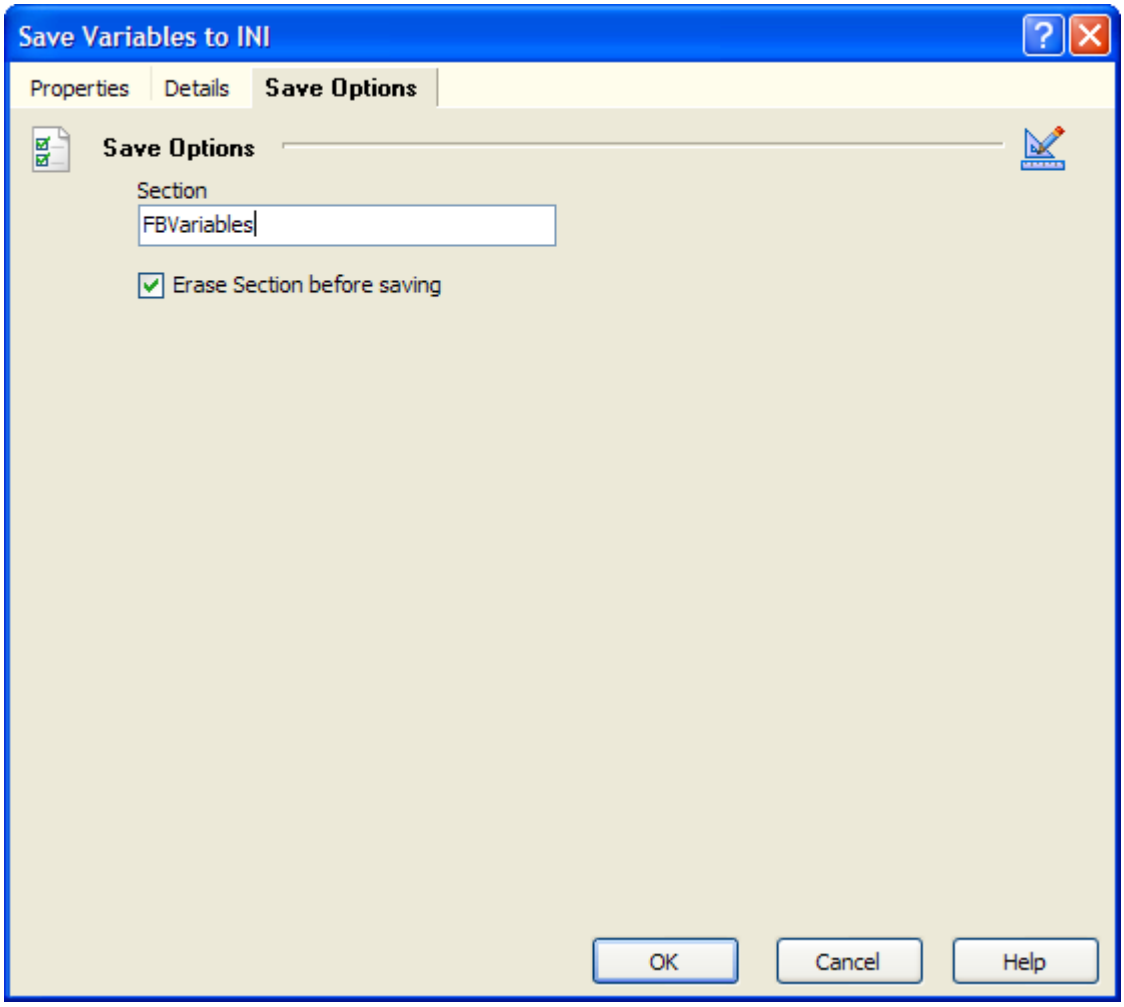
The Save Variables from INI file action enables you to save a set of FinalBuilder variables to an INI file.



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>

© 2001-2006 VSoft Technologies Pty Ltd



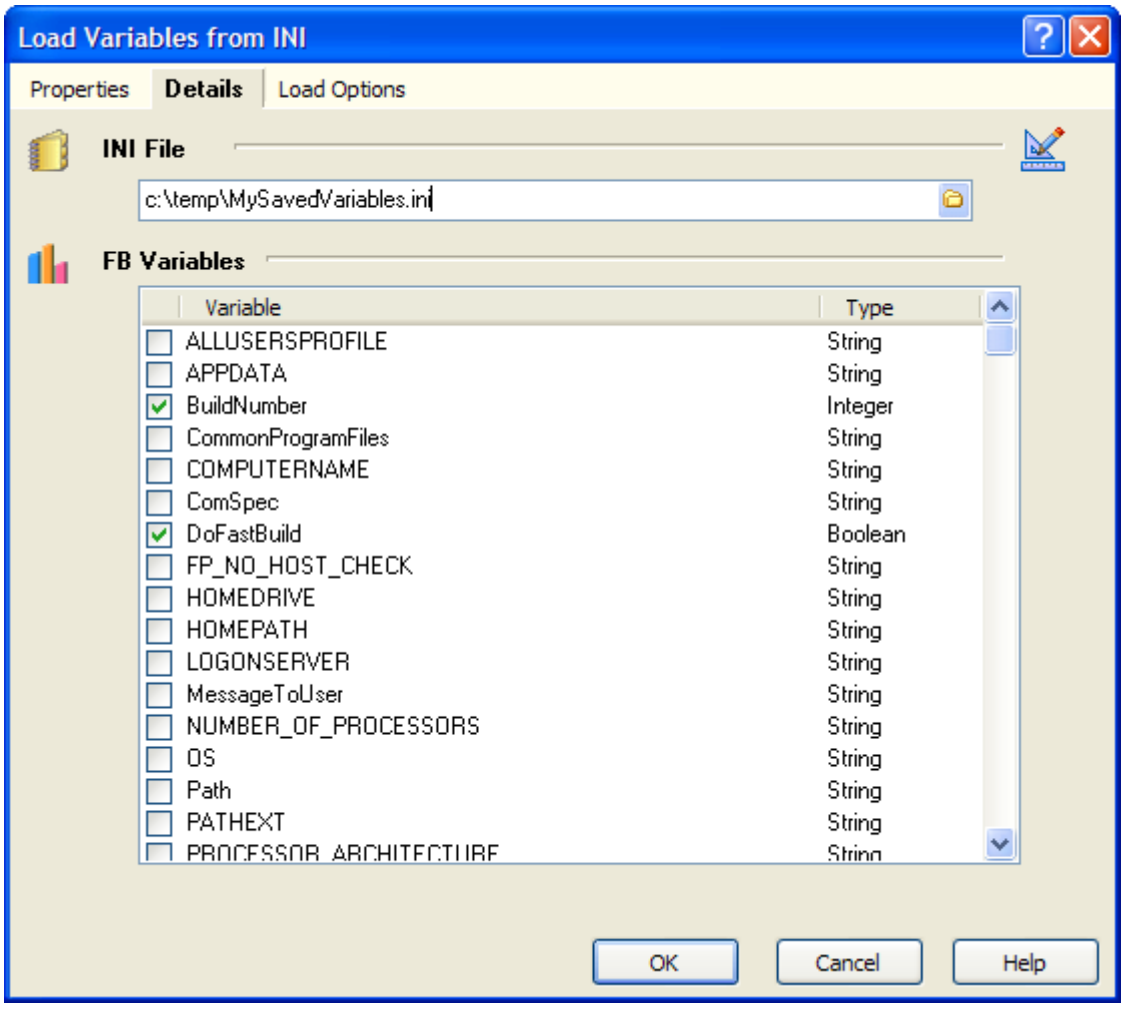
Section - Specify the section in the INI file to write the variables to

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.9.4 Load Variables From Ini

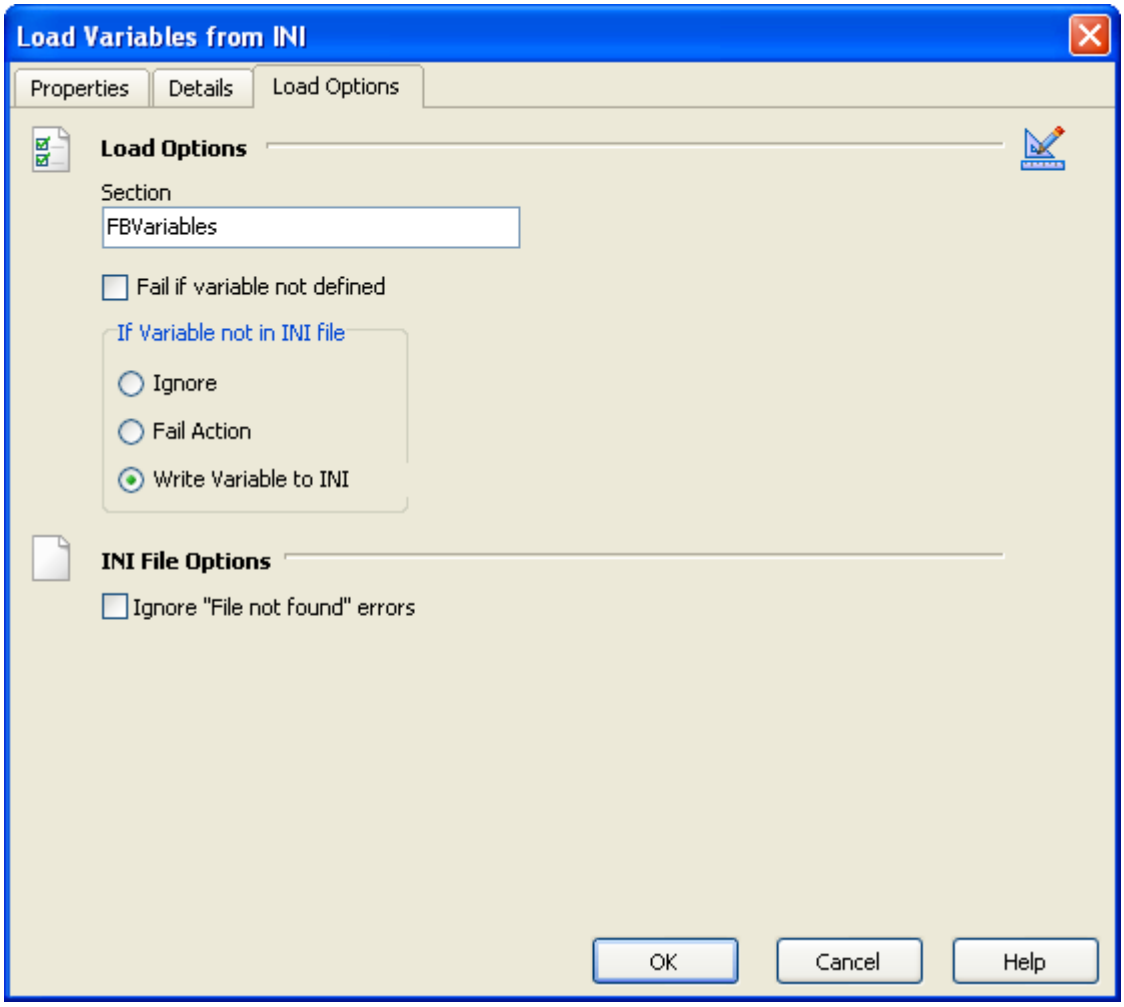
The Load Variables from INI file action enables you to set FinalBuilder variables to the values specified in an INI file.



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.

© 2001-2006 VSoft Technologies Pty Ltd



Section - specify the section in the INI where the variables are located.

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.

Fail if variable not defined

If this option is set and you attempt to load an undefined variable, the action will fail.

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.

© 2001-2006 VSoft Technologies Pty Ltd

Write Variable to INI File - If any variable is not found in the INI file, the current value of that variable will be written to the INI file.

Ignore "File not found" errors

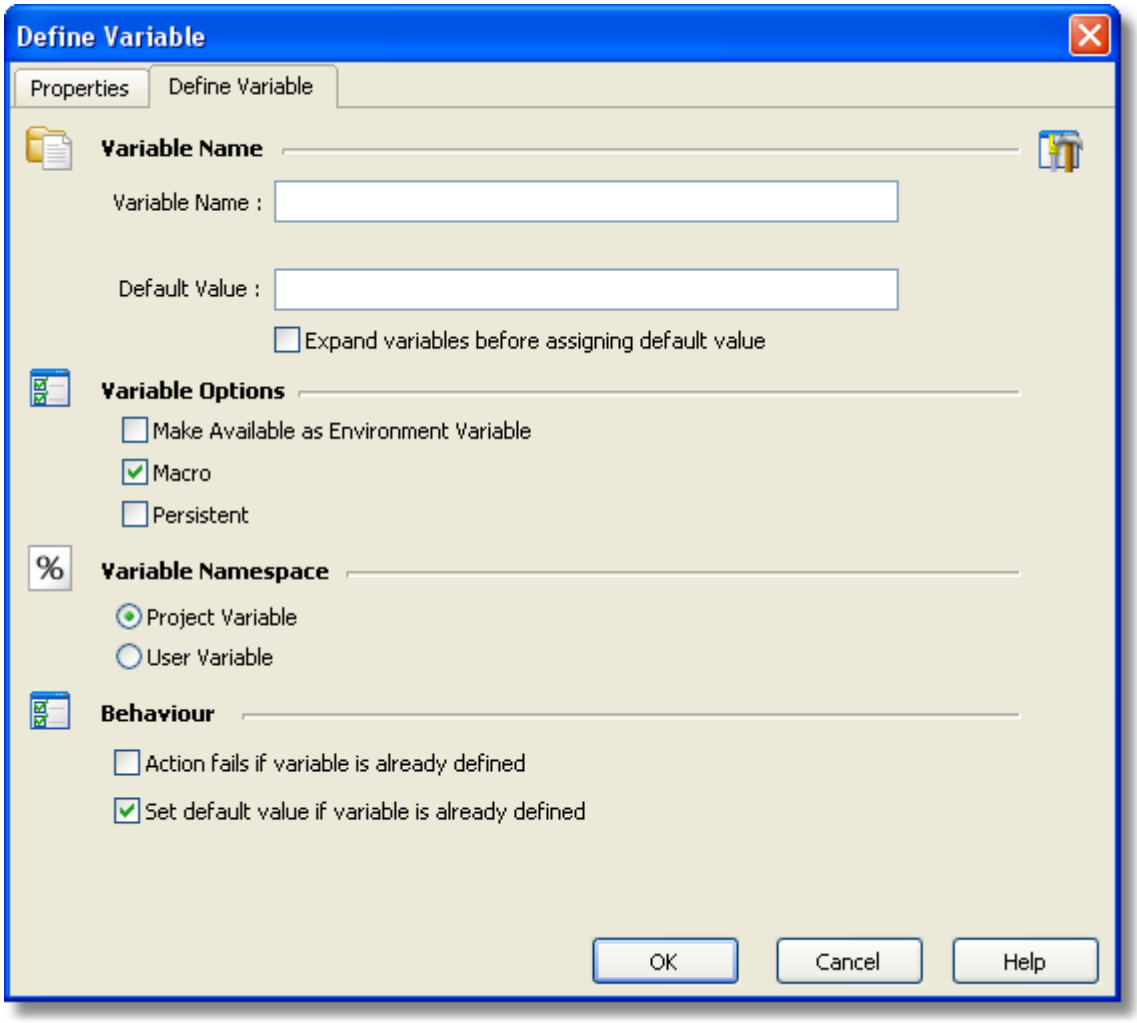
Set this option to true if you want the action to succeed even if the INI File does not exist at all. You will need to set "If Variable not in INI File" to 'Ignore' or 'Write Variable to INI'. If you use 'Write Variable to INI', a new INI file will be created.

© 2001-2006 VSoft Technologies Pty Ltd

5.9.5 Define Variable Action

The define variable action allows you to define a new FinalBuilder variable during a build process. 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.



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

© 2001-2006 VSoft Technologies Pty Ltd

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 - it's value cannot be set during the build process using the Set Variable action or any other means.

Peristent

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 process *are* persistent, so if you save a project file after defining a new variable, the new variable will remained defined in the Variables window.)

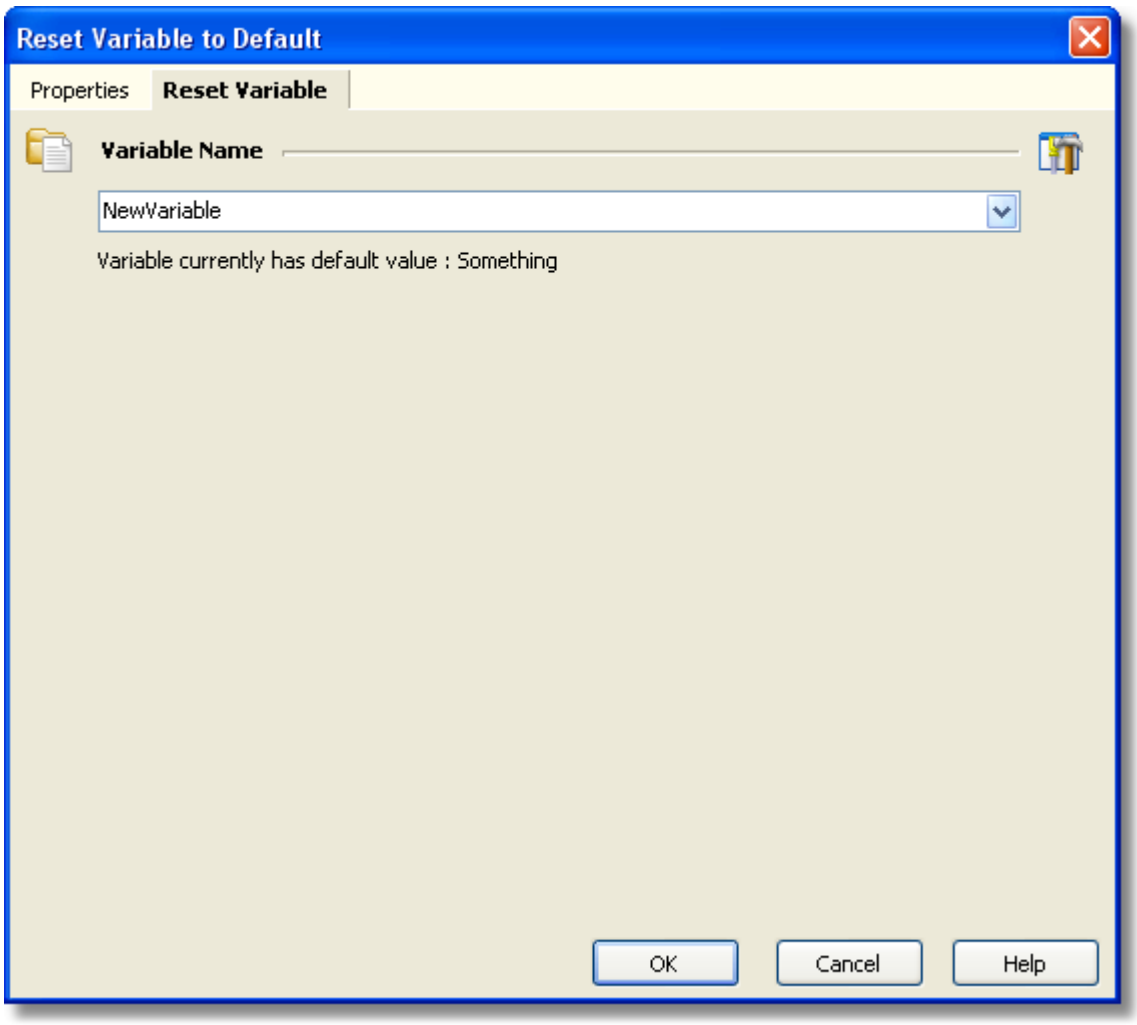
If the action is set to ignore preexisting variables, it can be set to update the default value. Note that this is the only part of the preexisting variable which can be updated by the action - it is impossible to change any other properties of the variable.

If the "update default value" box is not checked, the action will do nothing if the variable already exists.

© 2001-2006 VSoft Technologies Pty Ltd

5.9.6 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.

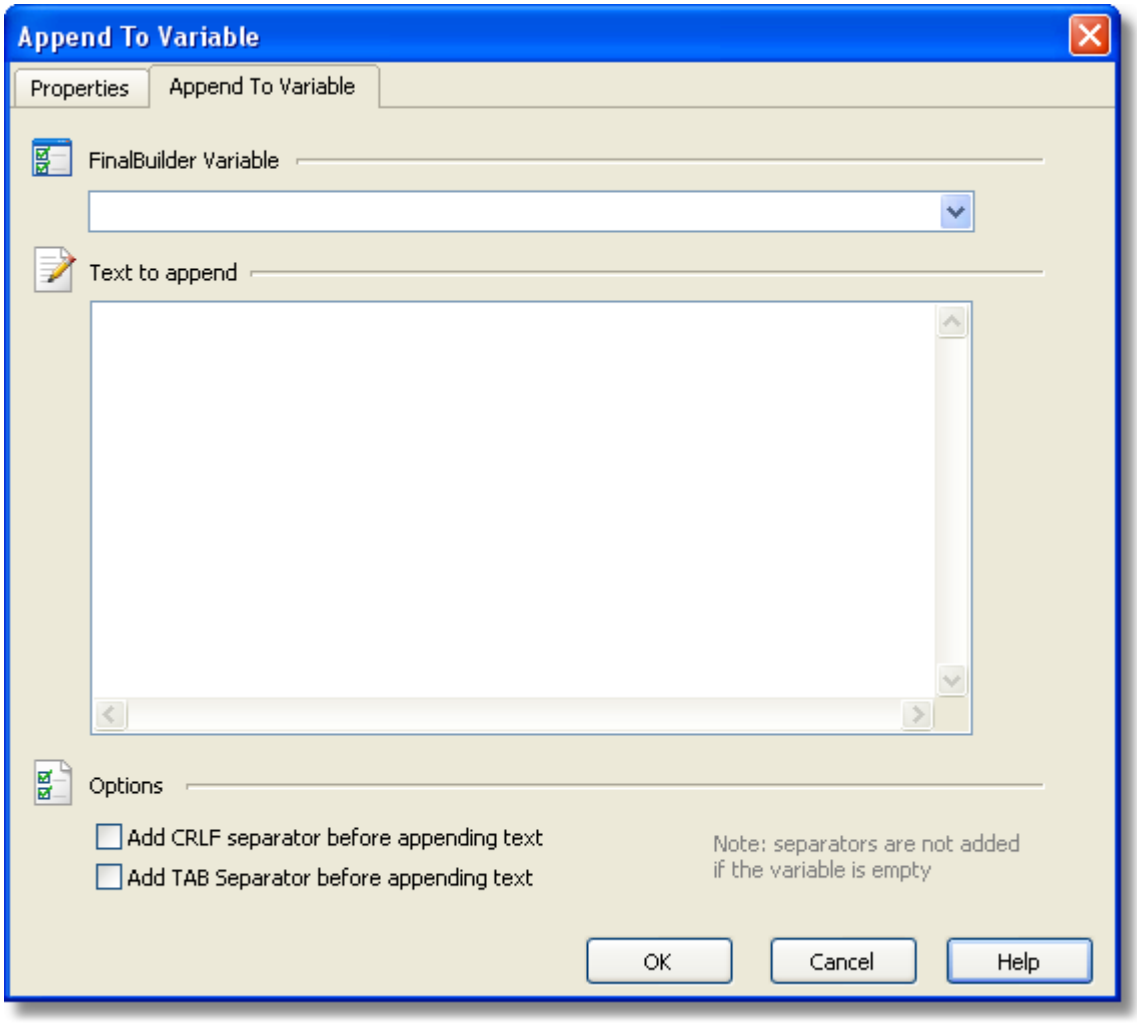
Note that you can reset variables which do not have default values - the variable will be set to blank.

© 2001-2006 VSoft Technologies Pty Ltd

5.9.7 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.



FinalBuilder 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.)

© 2001-2006 VSoft Technologies Pty Ltd

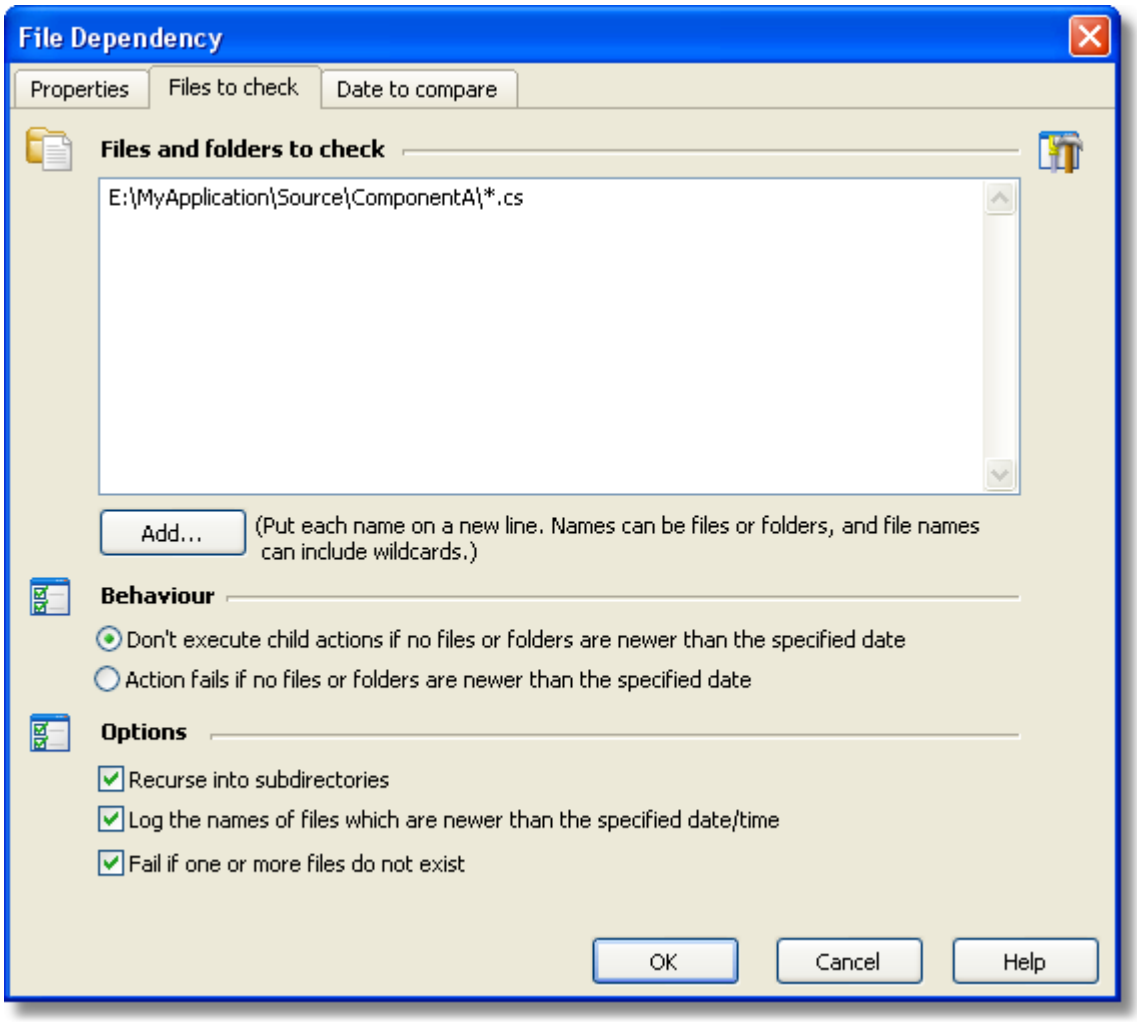
5.10 Files & Directories

5.10.1 File Dependency Action

The file dependency action allows you to control the flow of your build process 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:

© 2001-2006 VSoft Technologies Pty Ltd

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"

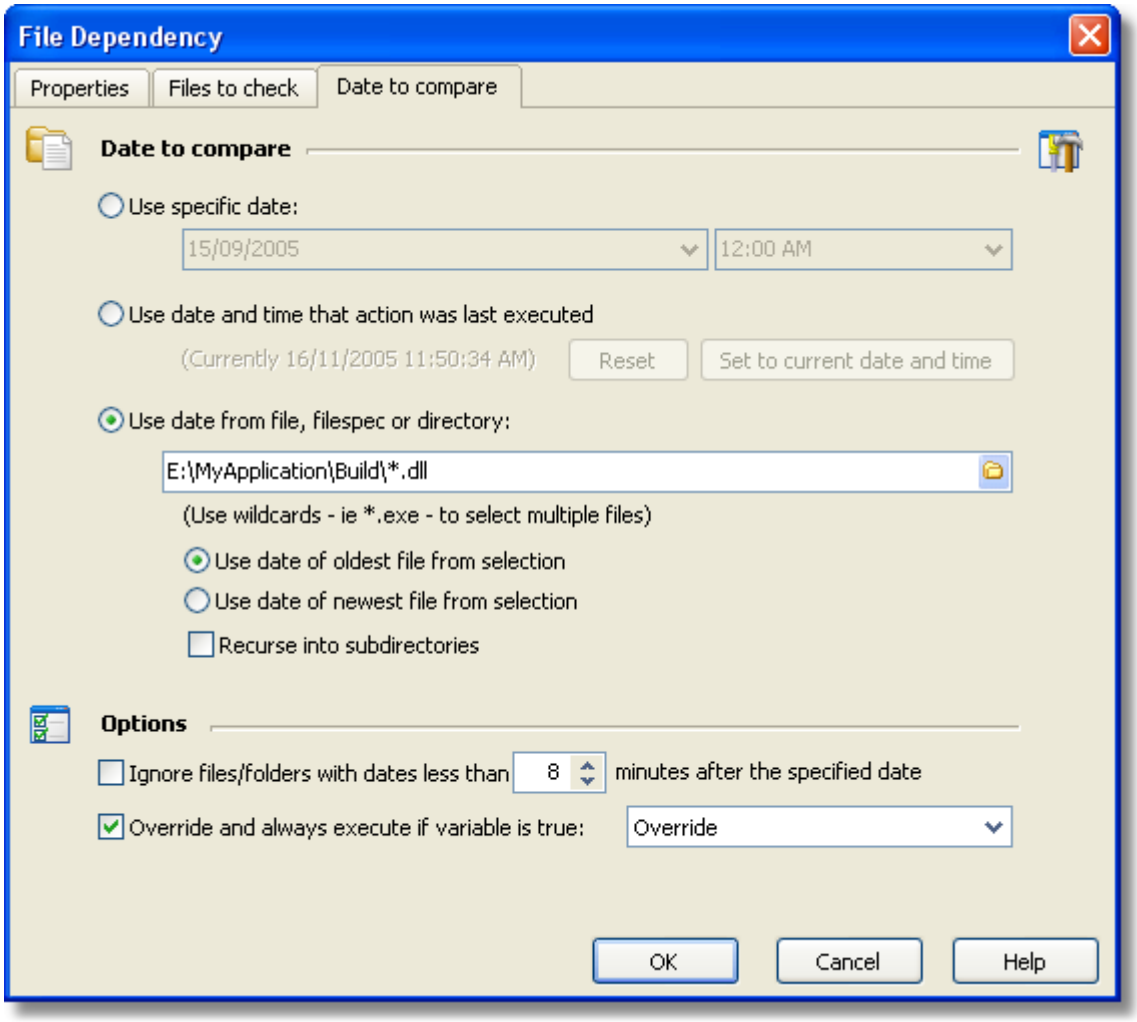
If you choose this option, then FinalBuilder will log the names of files which are newer than the timestamp.

"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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

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..."

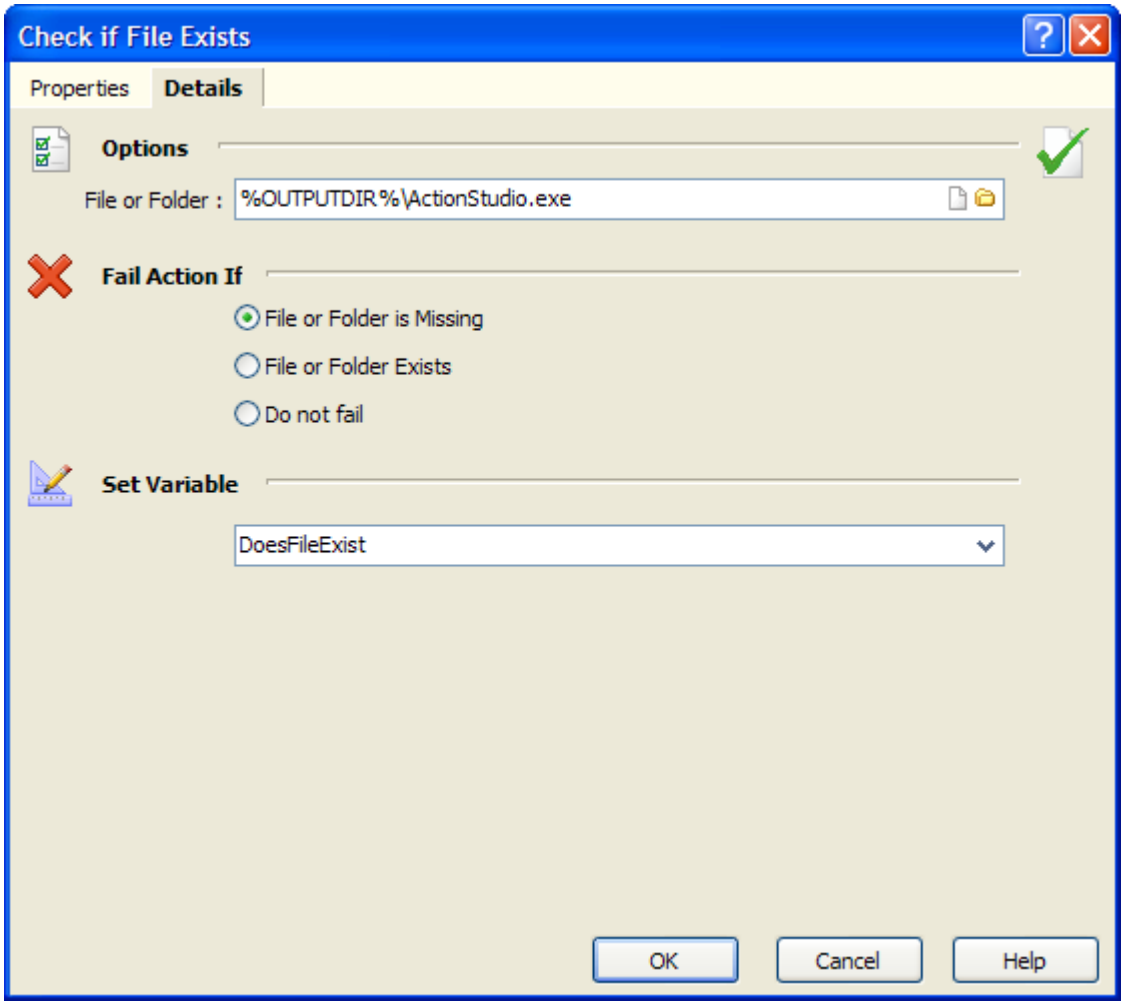
If this option is selected, then the chosen FinalBuilder variable (which needs to be a boolean variable) is evaluated before file dependencies are calculated. If the variable evaluates to true then the dependency evaluation is skipped and the child actions are run automatically.

(This option is useful if you want to sometimes force your build to go ahead, regardless of timestamps.)

© 2001-2006 VSoft Technologies Pty Ltd

5.10.2 Check File Exists Actoin

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



You can choose to fail the action if the file or folder exists, or if the file or folder is missing.

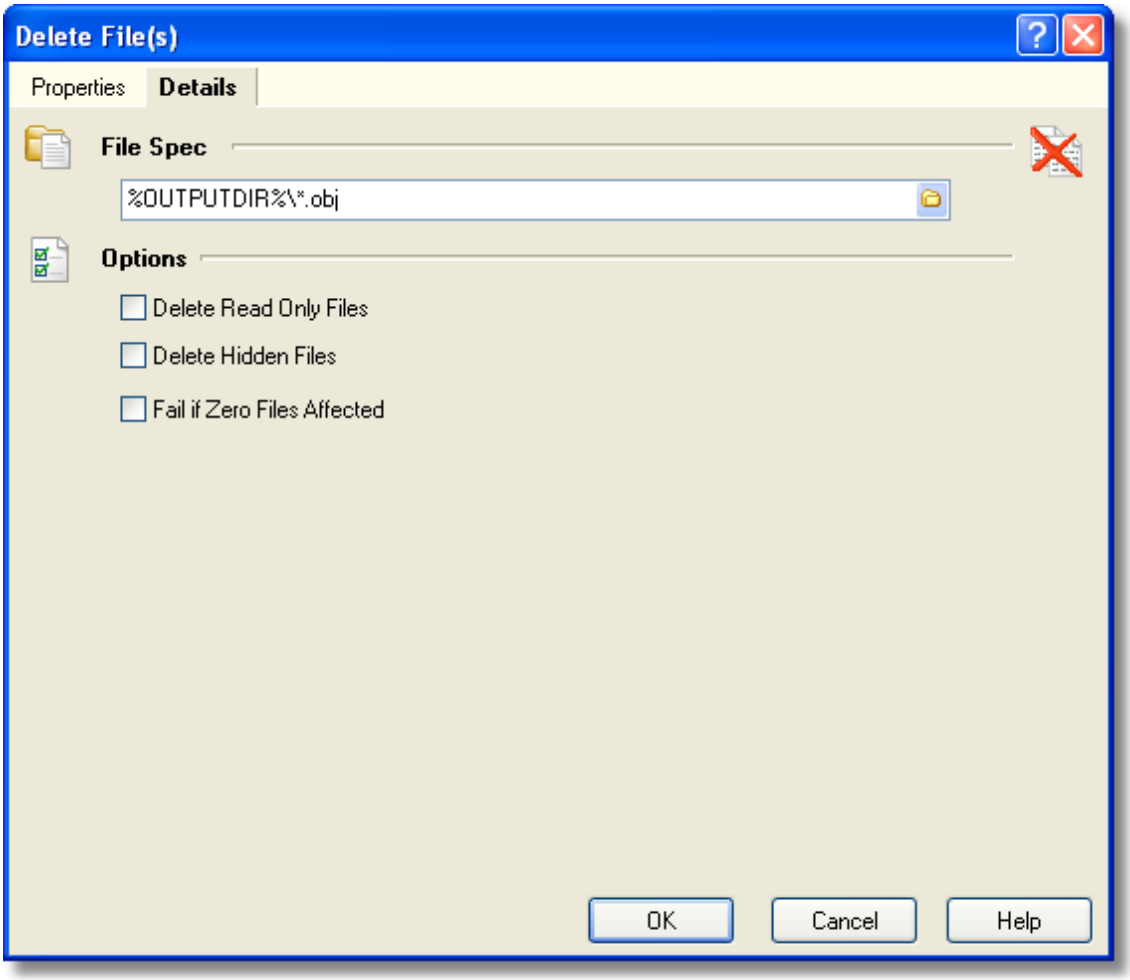
You can also choose not to fail the action at all; in this case you would normally set a variable so that you can use the condition in a subsequent action (for example an If...Then action).

© 2001-2006 VSoft Technologies Pty Ltd

5.10.3 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
```

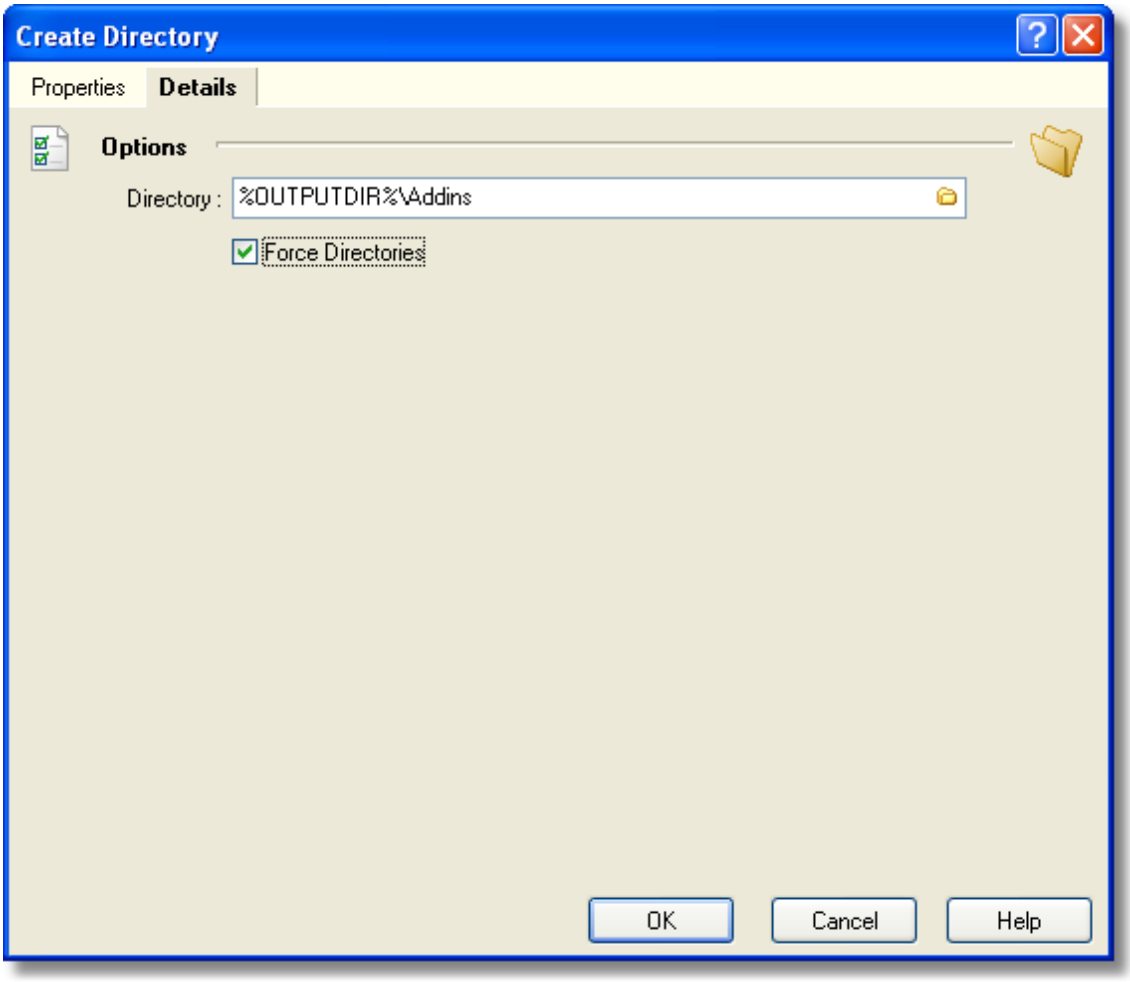
© 2001-2006 VSoft Technologies Pty Ltd

```
property  FailIfNoFile : WordBool; // fail if no files affected.
property  DeleteReadOnly : WordBool; // delete read only files.
property  DeleteHidden : WordBool; //delete hidden files.
```

© 2001-2006 VSoft Technologies Pty Ltd

5.10.4 Create Directory Action

This action will create the specified Directory.



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

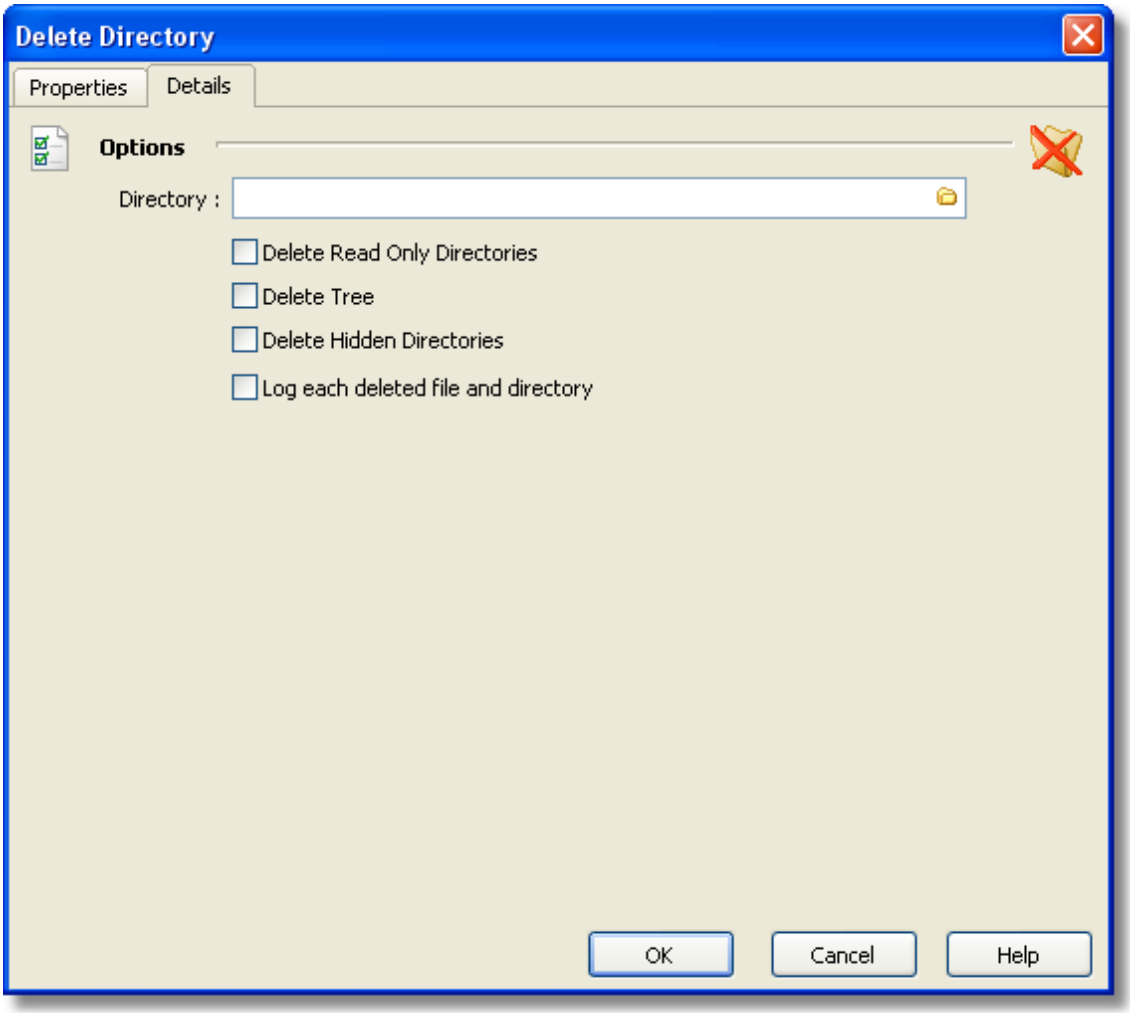
The Action properties available are :

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

© 2001-2006 VSoft Technologies Pty Ltd

5.10.5 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 :

© 2001-2006 VSoft Technologies Pty Ltd

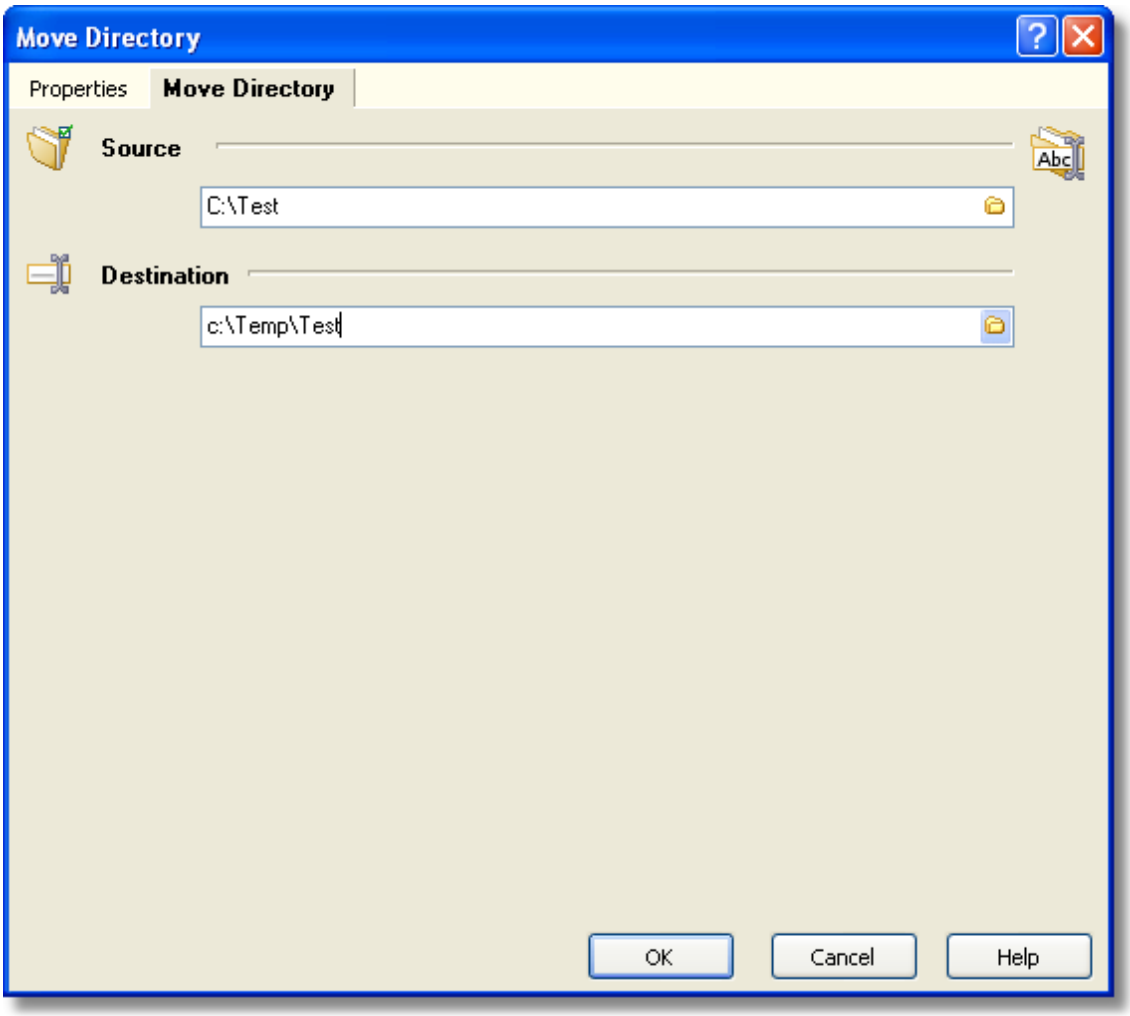
```
property FileOrDirectory  : WideString;// The Directory to Delete
property DeleteReadOnly : WordBool;
property DeleteHidden : WordBool;
property DeleteTree : WordBool;
```

© 2001-2006 VSoft Technologies Pty Ltd

5.10.6 Move Directory Action

This action will move the specified Directory to the Destination Directory.

Note that the Destination directory is the new name for the folder, so for example if you want to move c:\test to c:\temp\test then you should specify c:\temp\test as the destination directory.



Scripting Info

The Action properties available are :

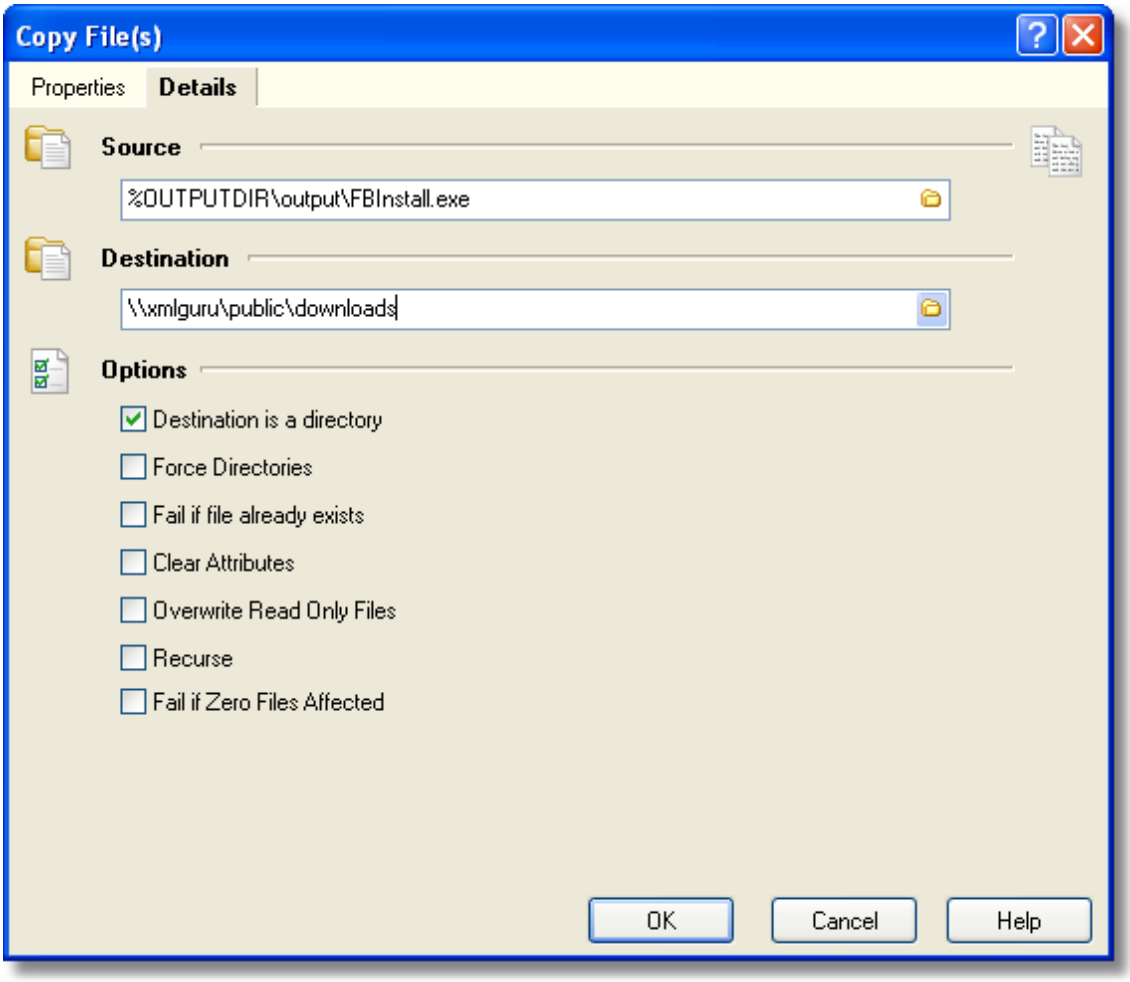
```
property  Directory  :String; // the directory you wish to move.
property  DestinationDirectory : String ; // the new path for the directory.
```

© 2001-2006 VSoft Technologies Pty Ltd

5.10.7 Copy File(s) Action

This action will copy a File or Files to the destination file or directory. You can use wildcards in the source setting to copy more than one file, in the Destination is a directory 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.

If you need more sophisticated support for choosing files to copy, use the Copy/Move FileSet action.



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 parent directories of the destination do not exist, they will be created.

Fail if file already exists - If this option is set and the destination file already exists, the action will fail. If it is not set, the destination file will be overwritten.

Clear Attributes - If this option is set, the destination file will be created with all attributes (Archive, Read Only, Hidden, System) cleared.

Overwrite Read Only Files - Set this option if you want to overwrite read only destination files.

© 2001-2006 VSoft Technologies Pty Ltd

Recurse - Set this option to search for file spec matches in subdirectories of the source directory.

Fail if Zero Files Affected - If this option is set and no files are copied, the action will fail.

Scripting Info

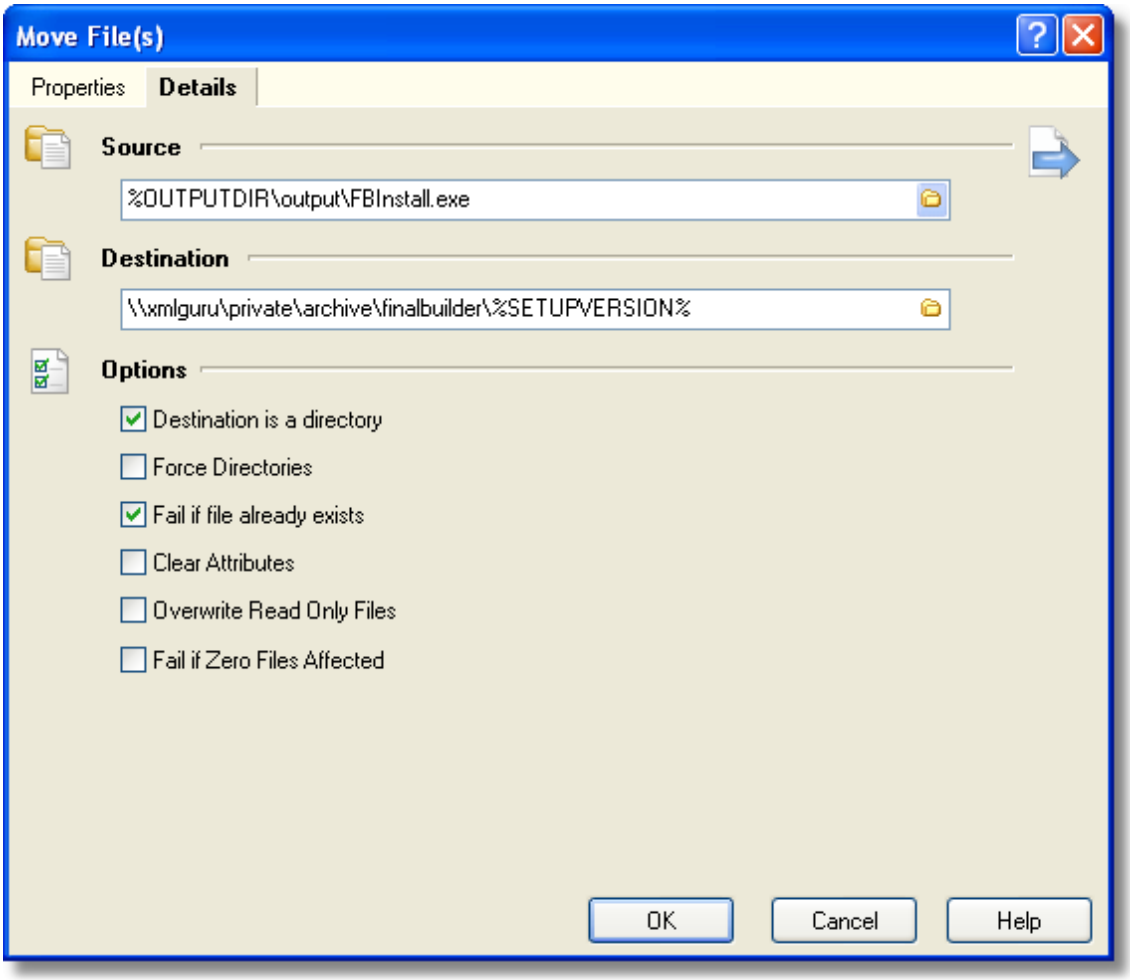
The Action properties available are :

```
        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  OverwriteReadOnly : WordBool;//Overwrite Read Only Files
        property  Recurse : WordBool;// Recurse subdirectories, only valid if the target is a directory
        property  ClearAttributes : WordBool // clear the moved file's attributes.
        property  FailIfZeroFiles : WordBool;
```

© 2001-2006 VSoft Technologies Pty Ltd

5.10.8 Move File(s) Action

This action will Move a File or Files to the destination file or directory. You can use wildcards in the source setting to Move more than one file, in the Destination is a directory 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.



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 parent directories of the destination do not exist, they will be created.

Fail if file already exists - If this option is set and the destination file already exists, the action will fail. If it is not set, the destination file will be overwritten.

Clear Attributes - If this option is set, the destination file will be moved with all attributes (Archive, Read Only, Hidden, System) cleared.

Overwrite Read Only Files - Set this option if you want to overwrite read only destination files.

Recurse - Set this option to search for file spec matches in subdirectories of the source directory.

Fail if Zero Files Affected - If this option is set and no files are moved, the action will fail.

© 2001-2006 VSoft Technologies Pty Ltd

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 OverwriteReadOnly : WordBool;//Overwrite Read Only Files
property Recurse : WordBool;// Recurse subdirectories, only valid if the target is a directory
property ClearAttributes : WordBool // clear the moved file's attributes.
property FailIfZeroFiles : WordBool;

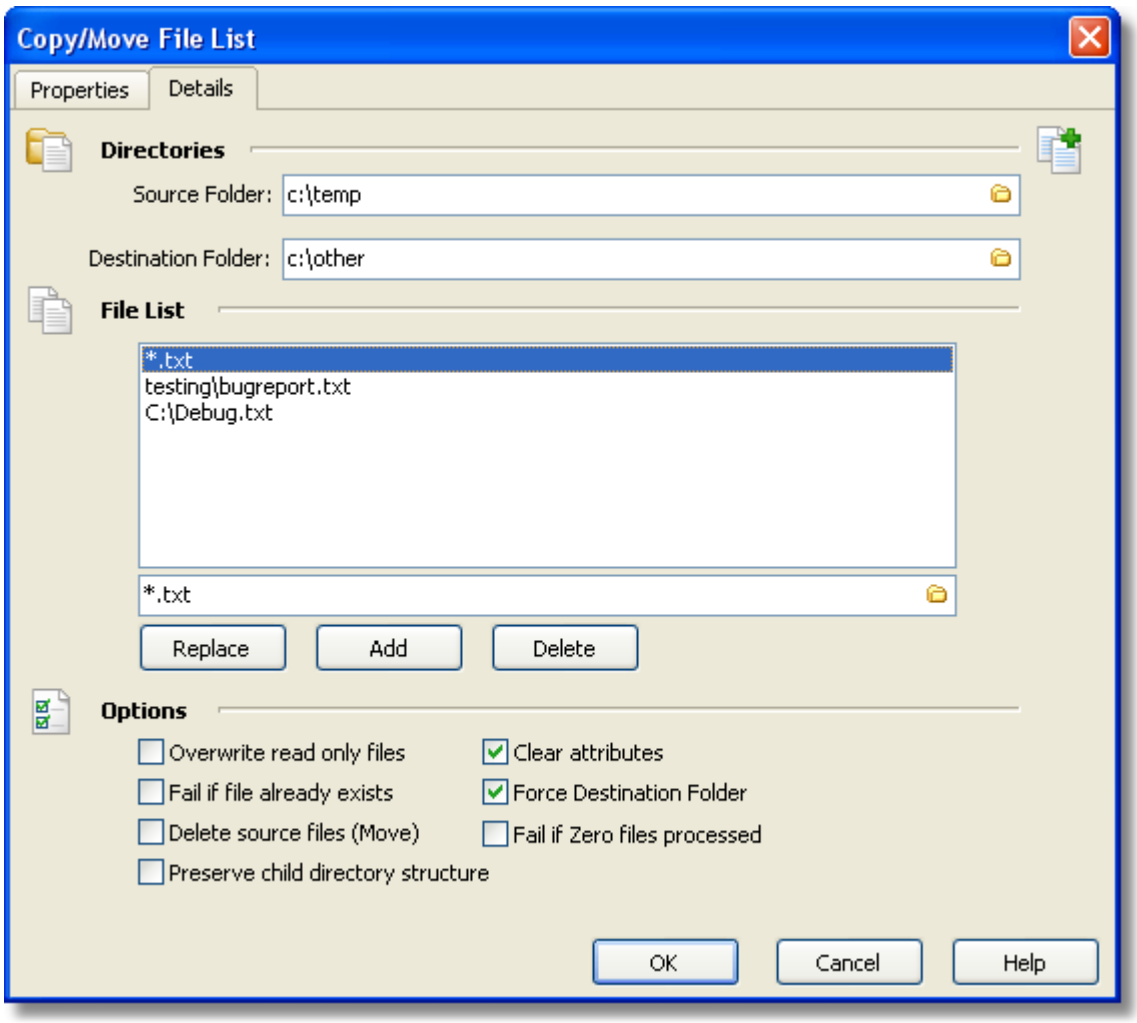
© 2001-2006 VSoft Technologies Pty Ltd

5.10.9 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. This action is now supported by VSoft Technologies as part of FinalBuilder.



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

© 2001-2006 VSoft Technologies Pty Ltd

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

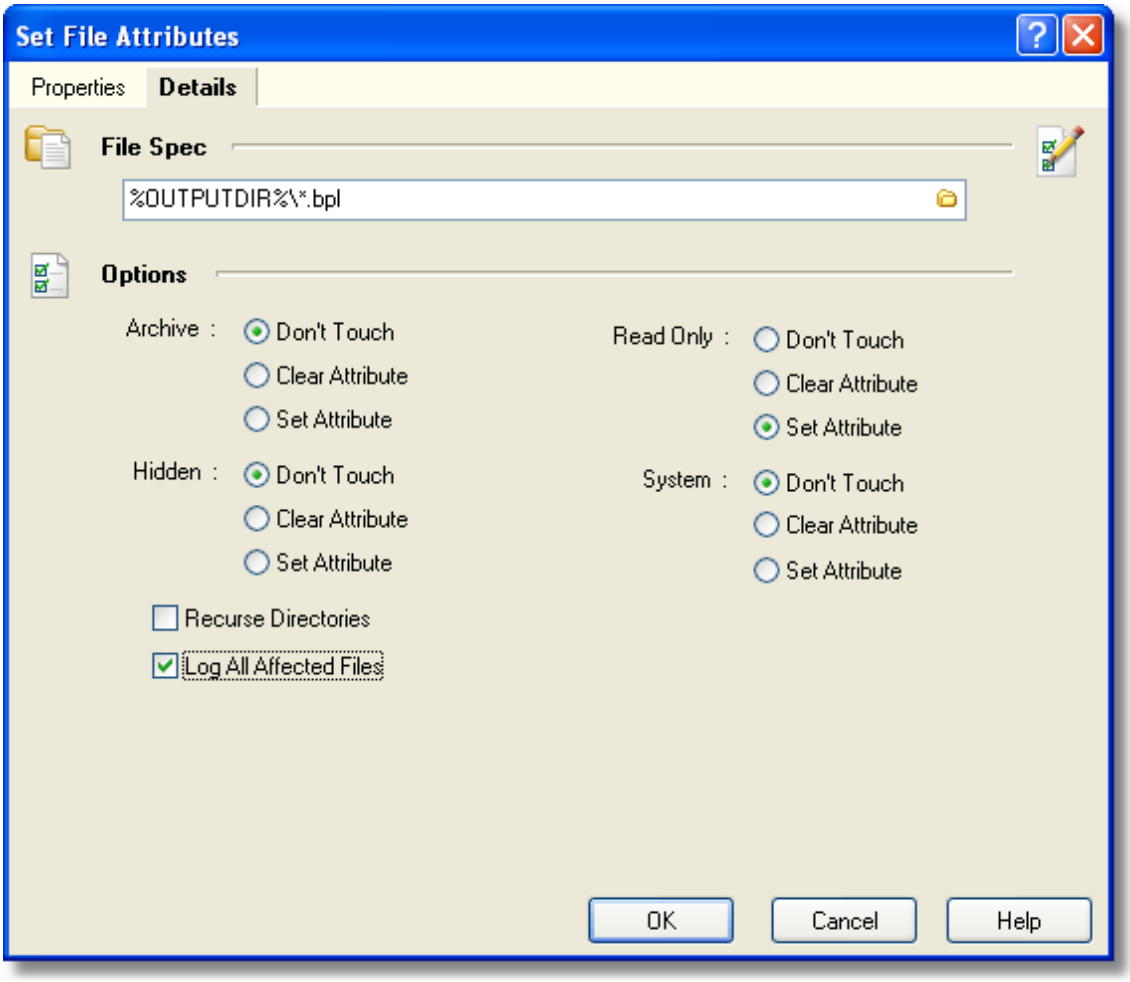
The list of files to be moved/copied from the Source Folder. This can include wildcards. Paths can be local (relative to the source directory) or absolute.

(Note that wildcards do not recurse subdirectories.)

© 2001-2006 VSoft Technologies Pty Ltd

5.10.10 Set File Attributes Action

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



Scripting Info

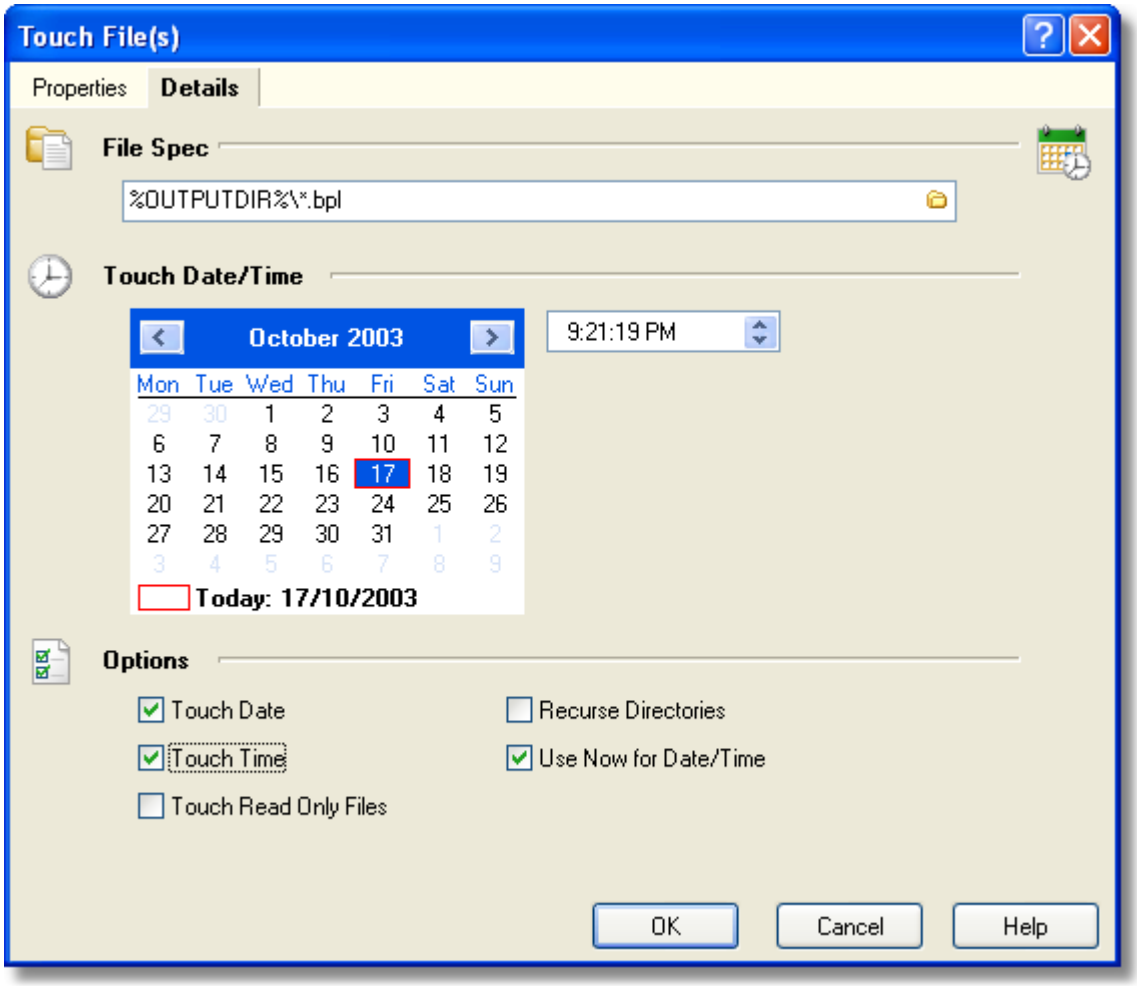
The Action properties available are :

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.10.11 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 File Date and/or Time for a file or Files.



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

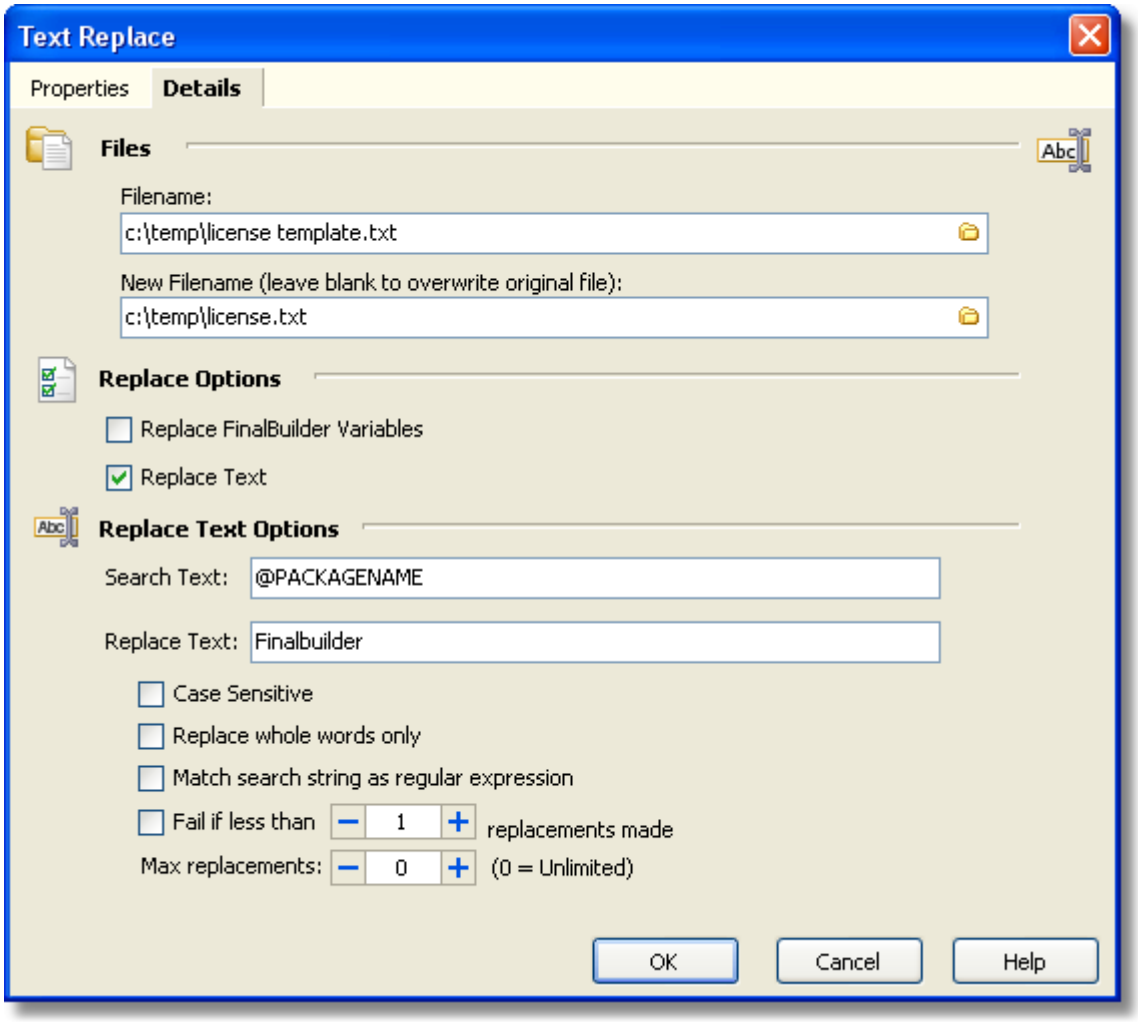
© 2001-2006 VSoft Technologies Pty Ltd

5.10.12 Text Replace Action

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

© 2001-2006 VSoft Technologies Pty Ltd

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.

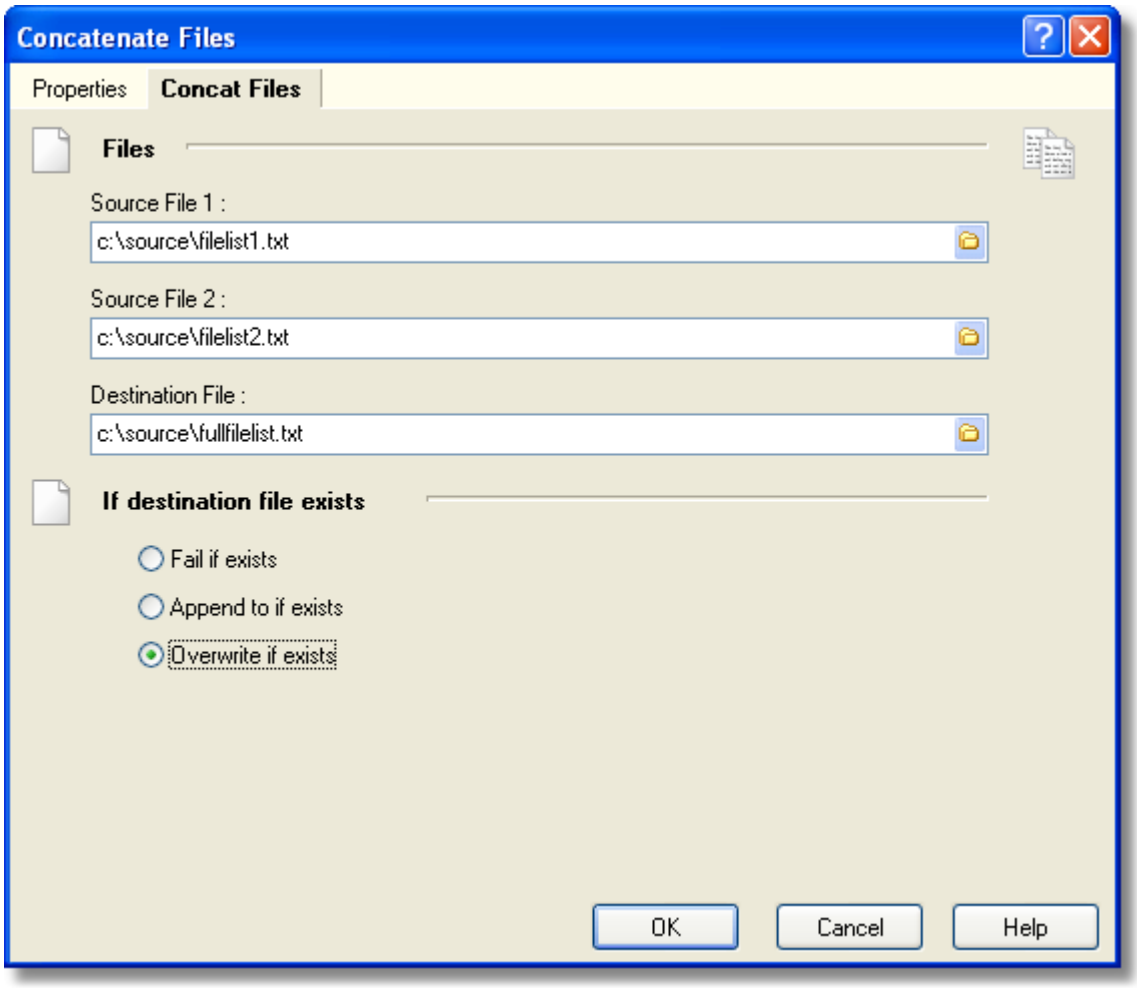
Max Replacements: N

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

© 2001-2006 VSoft Technologies Pty Ltd

5.10.13 Concatenate Files Action

This action concatenates two text files into a single new file.

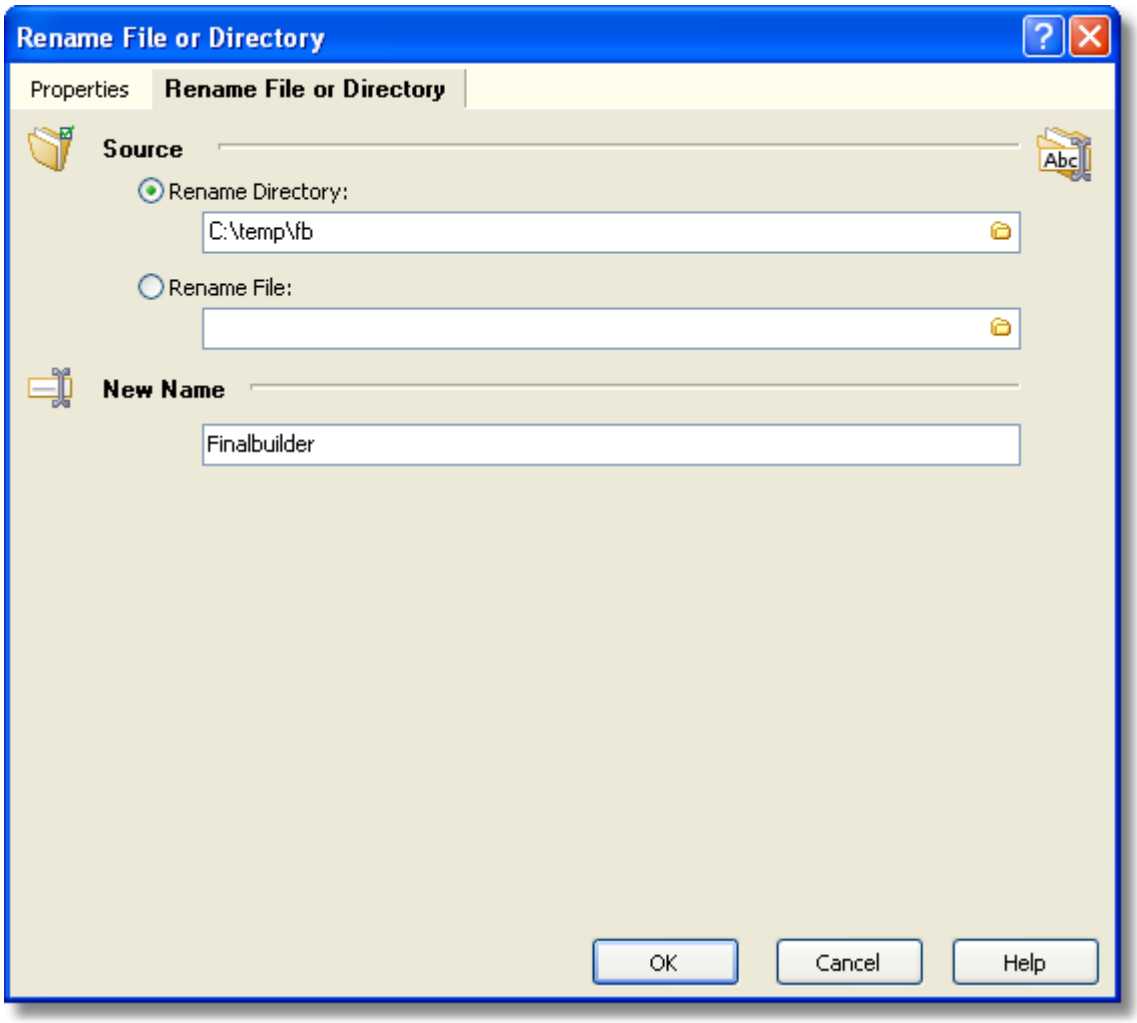


Note that Source Files 1 & 2 cannot be the same file, and they cannot be the same as the destination file. Source 2 may be blank if appending to an existing file.

© 2001-2006 VSoft Technologies Pty Ltd

5.10.14 Rename File or Directory Action

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



The NewName property is the new name of the file or directory (do not specify the path).

© 2001-2006 VSoft Technologies Pty Ltd

5.10.15 Authenticode

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

This action requires your Authenticode Signing Certificate to be in a PFX file. Most CA's are provided your certificate in two files: an SPC and a PVK.

To make the PFX, you need to use a Microsoft tool called PVKIMPRT [more info: http://msdn.microsoft.com/vba/technical/pvk.asp] or [download: http://download.microsoft.com/download/vba50/Utility/1.0/NT5/EN-US/pvkimprt.exe].

Then, open a console window (dos box), switch to the directory that contains your certificates 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), and then you get 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 -- suggestion: pick a GOOD password. Finally, you get prompted as to where you would like the PFX created.

This page on the VeriSign website provides some useful information also: http://www.verisign.com/resources/gd/authenticode/index.html

---
This action includes source code from StreamSec. It's 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

© 2001-2006 VSoft Technologies Pty Ltd

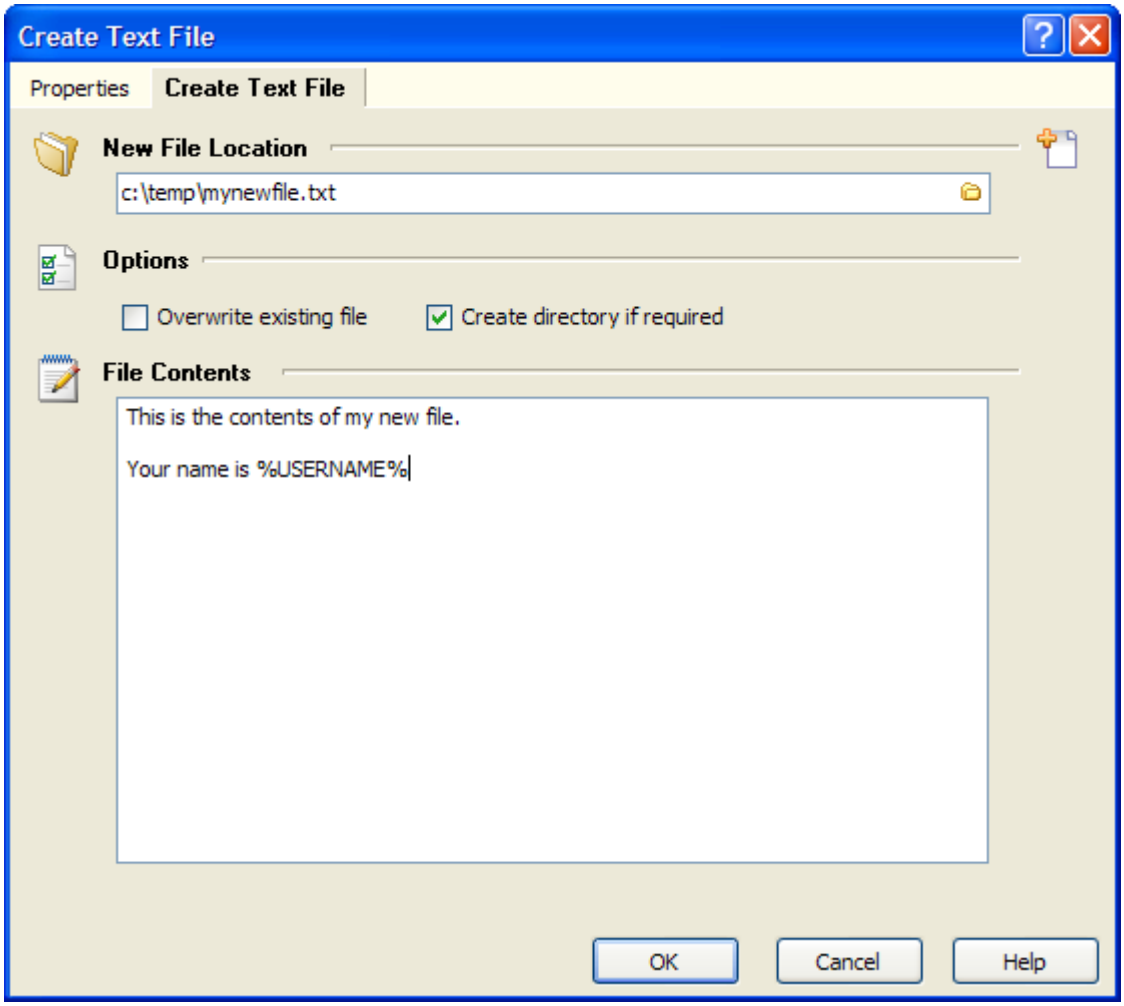
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

© 2001-2006 VSoft Technologies Pty Ltd

5.10.16 Create Text File

The Create Text File action enables you to create of text files as part of your build process.

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.

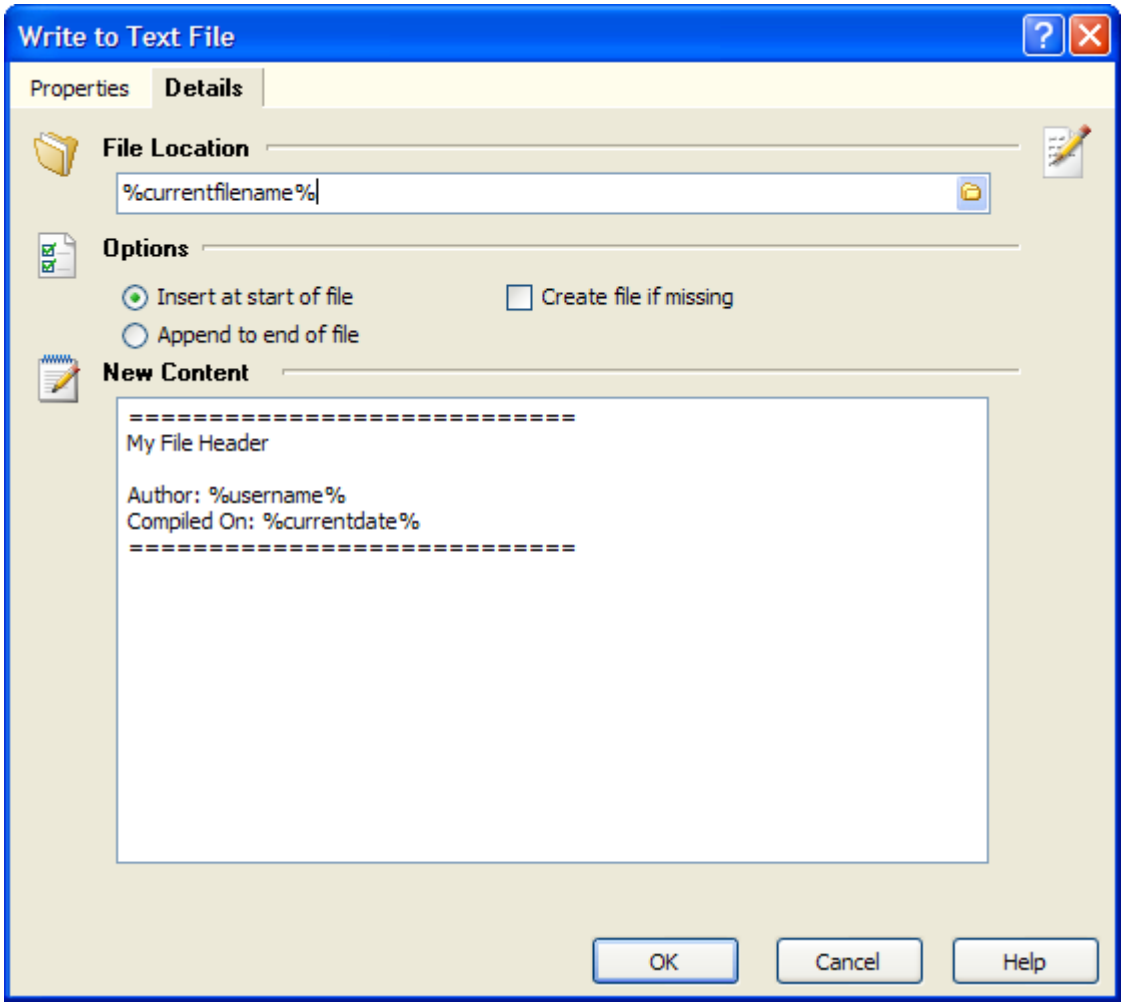
File Contents
Specify the text to be written to the file. You can use FinalBuilder variables in the File Contents field to customise the file based on the build.

© 2001-2006 VSoft Technologies Pty Ltd

5.10.17 Write Text File

The Write to Text File action enables you to automate appending to text files during your build process.

You can use FinalBuilder variables in the File Contents field to customise the content based on the build. You can either append to the end of the file, or insert text at the start of the file.



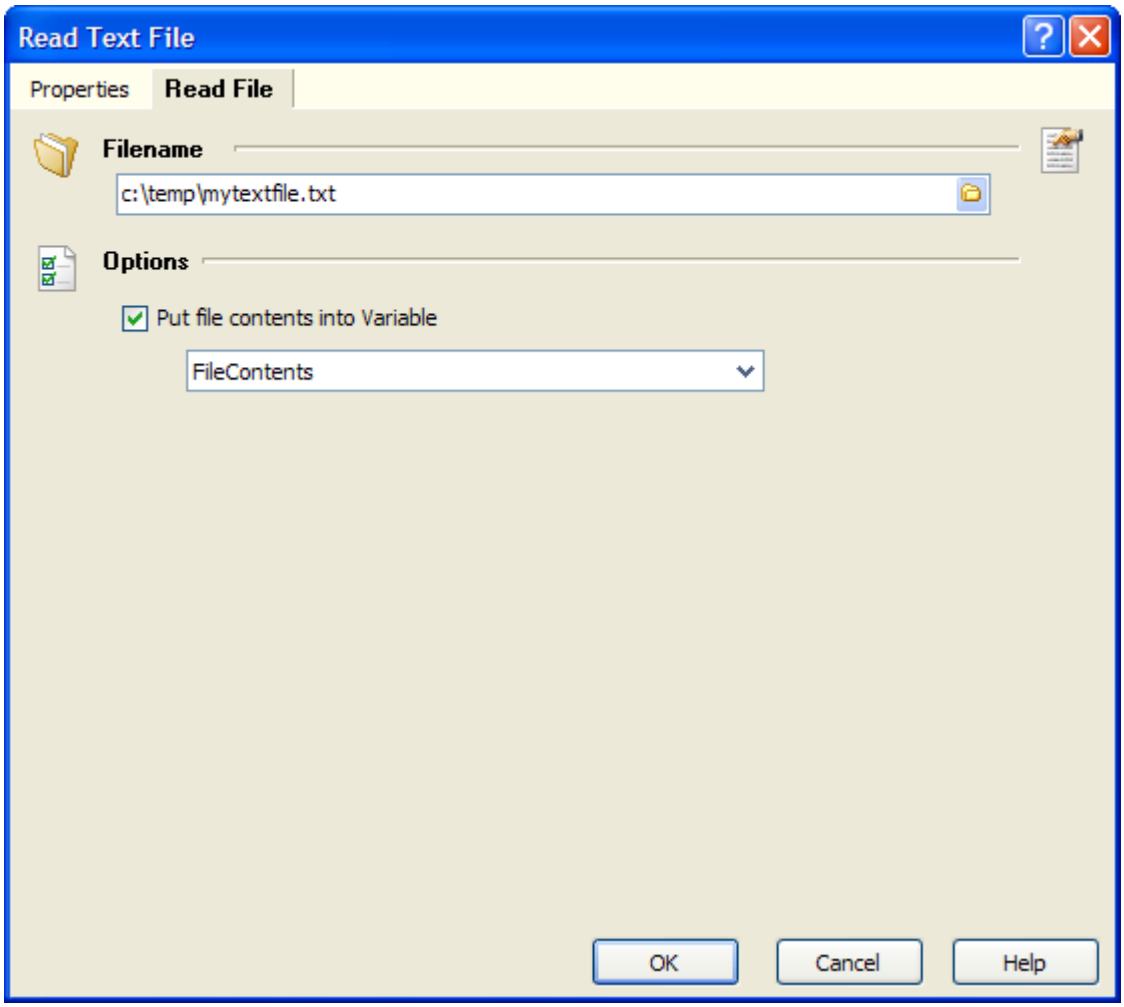
File Location - specify the filename to write to (the example shows a variable which will contain a filename during execution - provided by a file iterator action)
Insert location - Specify to append to end of file, or insert at beginning
Create file if missing - the action will normally fail if the file doesn't already exist.
New Content - specify the content to write to the text file

© 2001-2006 VSoft Technologies Pty Ltd

5.10.18 Read Text File

The Read Text File action enables you to automate the reading of a text file as part of your build process

You can read the text file into a FinalBuilder variable, and also process each line in the OnReadLine event.



Filename

Specify 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 during your build process.

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.10.19 XCopy

The XCopy action enables you to automate file and directory copying using XCopy as part of your build process. XCopy is a powerful utility included in Windows which is faster and more capable than standard file and copy tools.

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

© 2001-2006 VSoft Technologies Pty Ltd

5.10.20 RoboCopy

The RoboCopy action enables you to automate file and directory copying using RoboCopy as part of your build process. 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.)

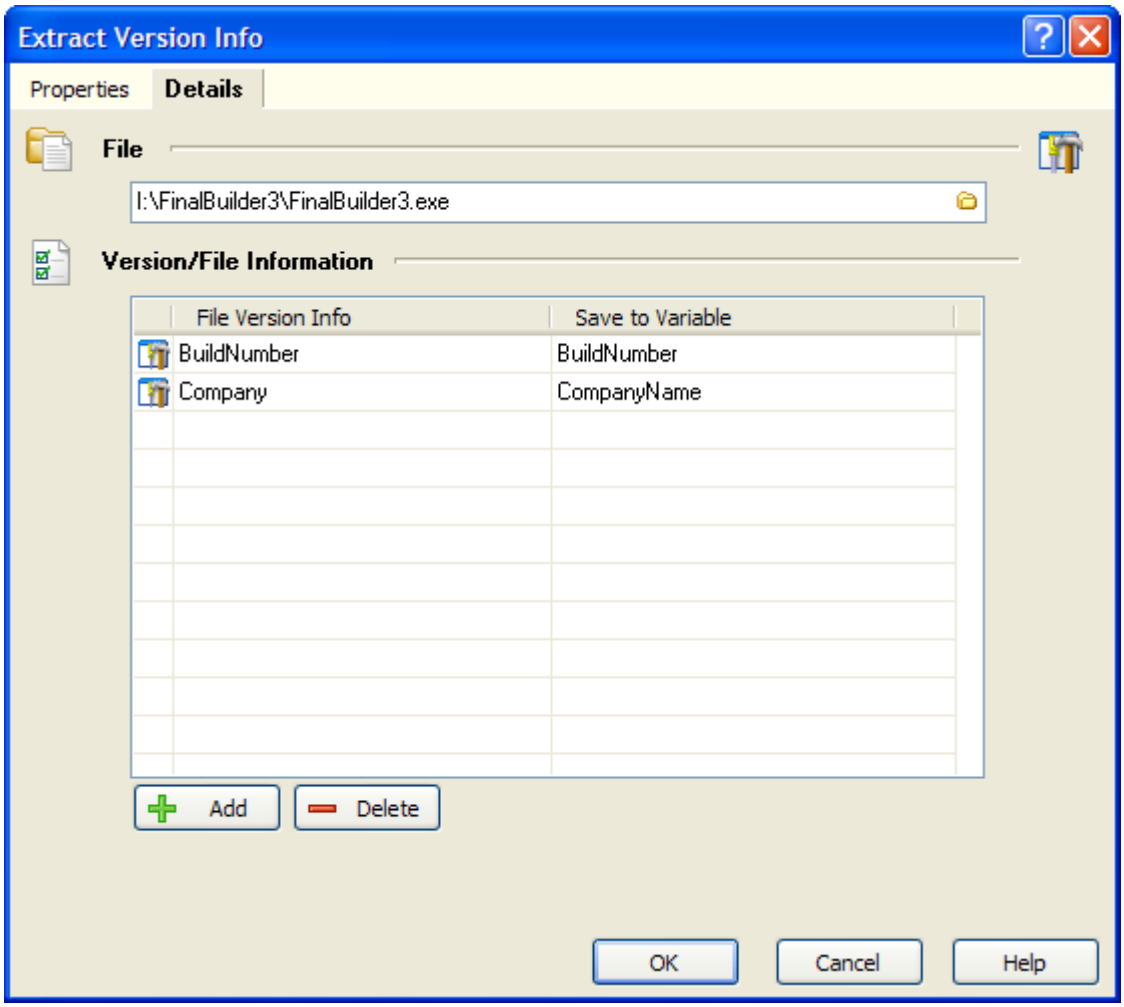
The RoboCopy actions are:
Robocopy - perform file and directory copy operations
Robocopy Job - run Robocopy Jobs
Robocopy Move - perform file and directory move operations
Robocopy Mirror - synchronise source directory to destination directory

© 2001-2006 VSoft Technologies Pty Ltd

5.10.21 Extract File Version

The Extract File/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.

Save To Variable
Specify the FinalBuilder variable to save the value to.

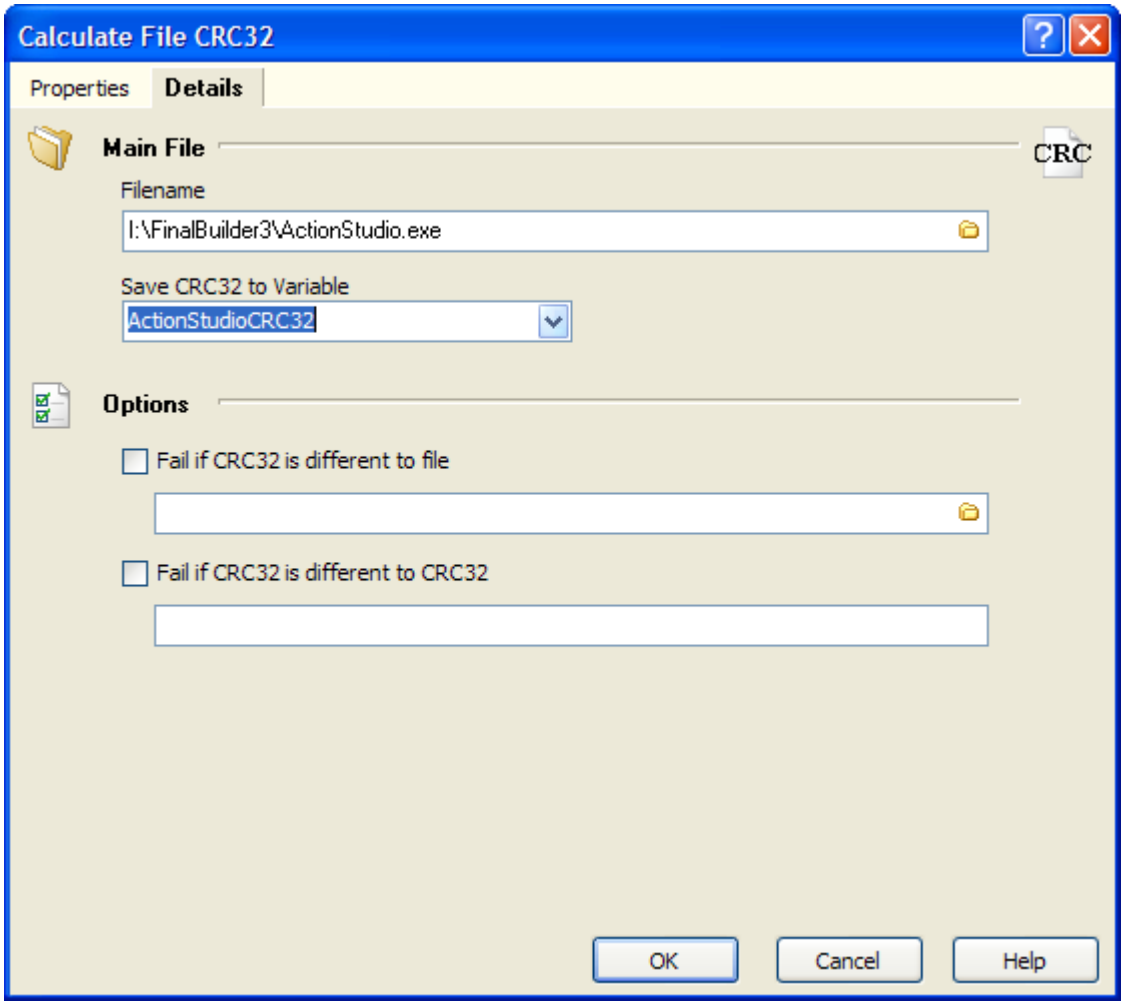
© 2001-2006 VSoft Technologies Pty Ltd

5.10.22 Calculate File CRC32

The File CRC32 action enables you to automate the calculation of a file's CRC32 as part of your build process. The action can also fail if the CRC differs from another file or a specified CRC32.

What is CRC32?

The CRC is acronym for "Cyclic Redundancy Code" and 32 represent the length of checksum in bits. The "CRC" term is reserved for algorithms that are based on the "polynomial" division idea. The idea to compute the checksum is equal for all CRC algorithms: Take the data as a VERY long binary number and divide it by a constant divisor. If you do this with integer values you get a rest; this rest is the CRC checksum (for example 7 / 3 = 2 + rest 1 => 1 is the checksum of 8. CRC is a family of algorithms and CRC32 is one certain member of this family (other members are CRC16, XMODEM,...); CRC32 produces a checksum with a length of 32 Bit (= 4Byte).



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.

© 2001-2006 VSoft Technologies Pty Ltd

Fail if CRC32 is different to CRC32 - specify a CRC32 to compare to the calculated CRC32 of the main file

© 2001-2006 VSoft Technologies Pty Ltd

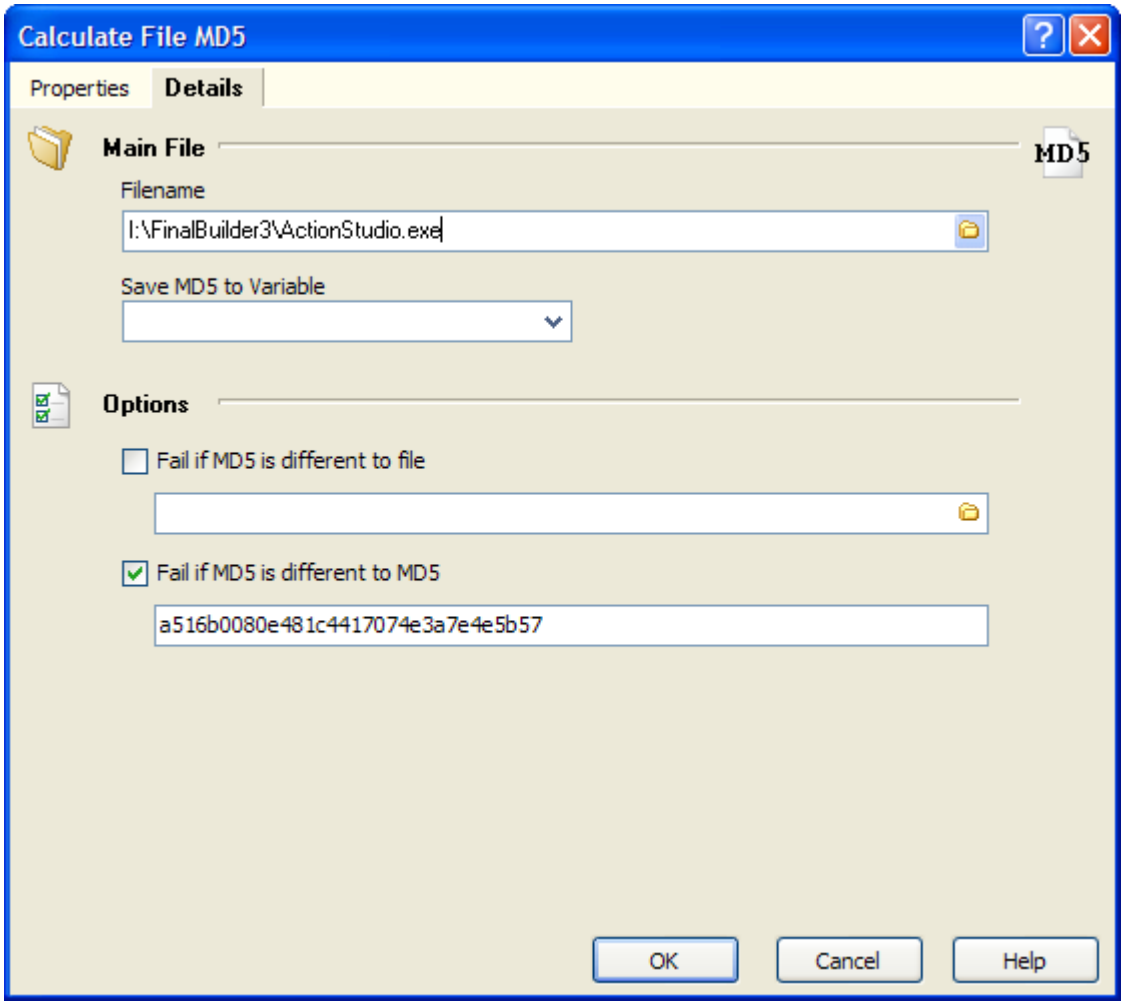
5.10.23 Calculate File MD5

The File MD5 action enables you to automate the calculation of a file's MD5 as part of your build process. 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 cryptosystem 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.

© 2001-2006 VSoft Technologies Pty Ltd

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

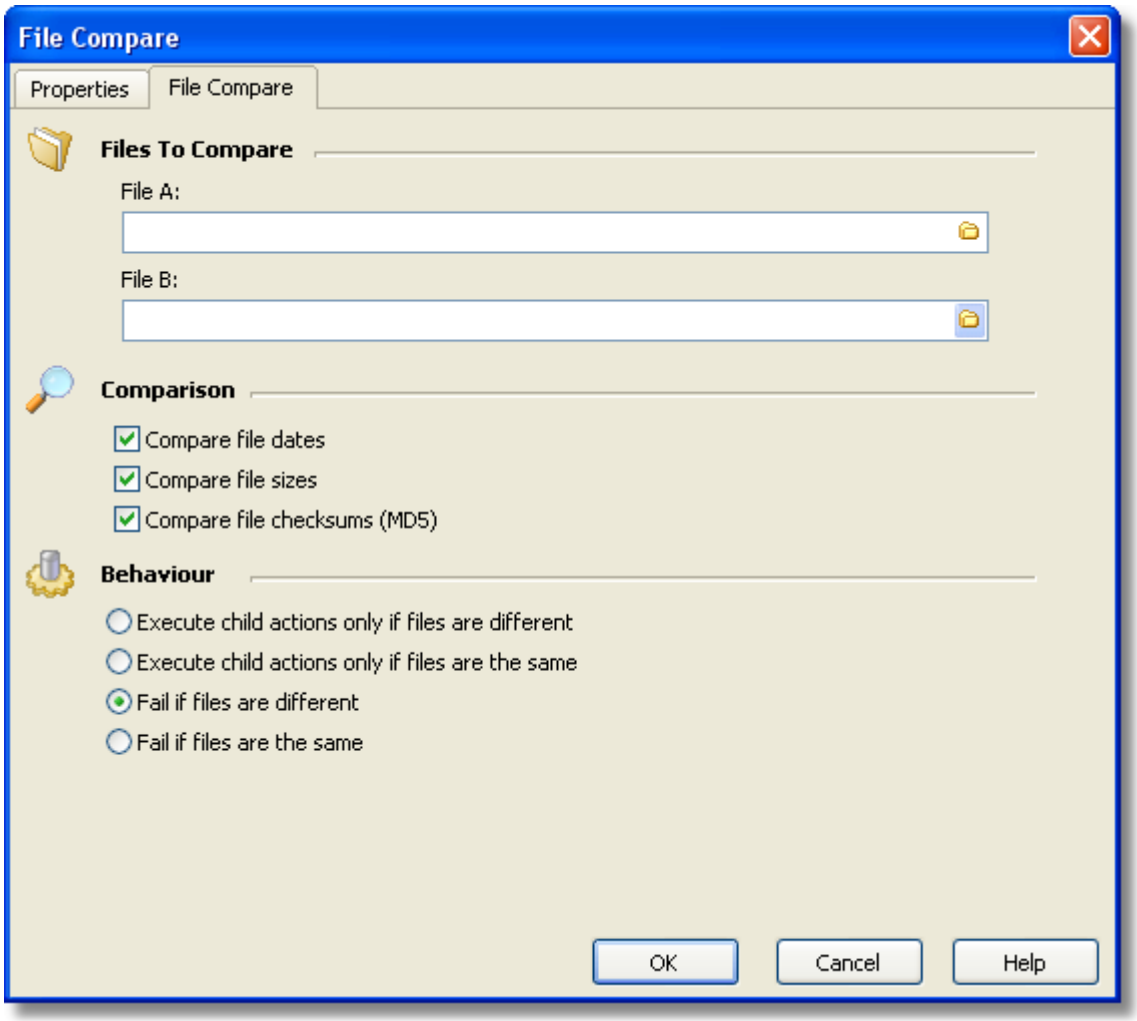
© 2001-2006 VSoft Technologies Pty Ltd

5.10.24 File Comparison Tools

5.10.24.1 File Compare Action

The File Compare action allows you to compare two files and use the result to influence your build process.

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.

© 2001-2006 VSoft Technologies Pty Ltd

"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.

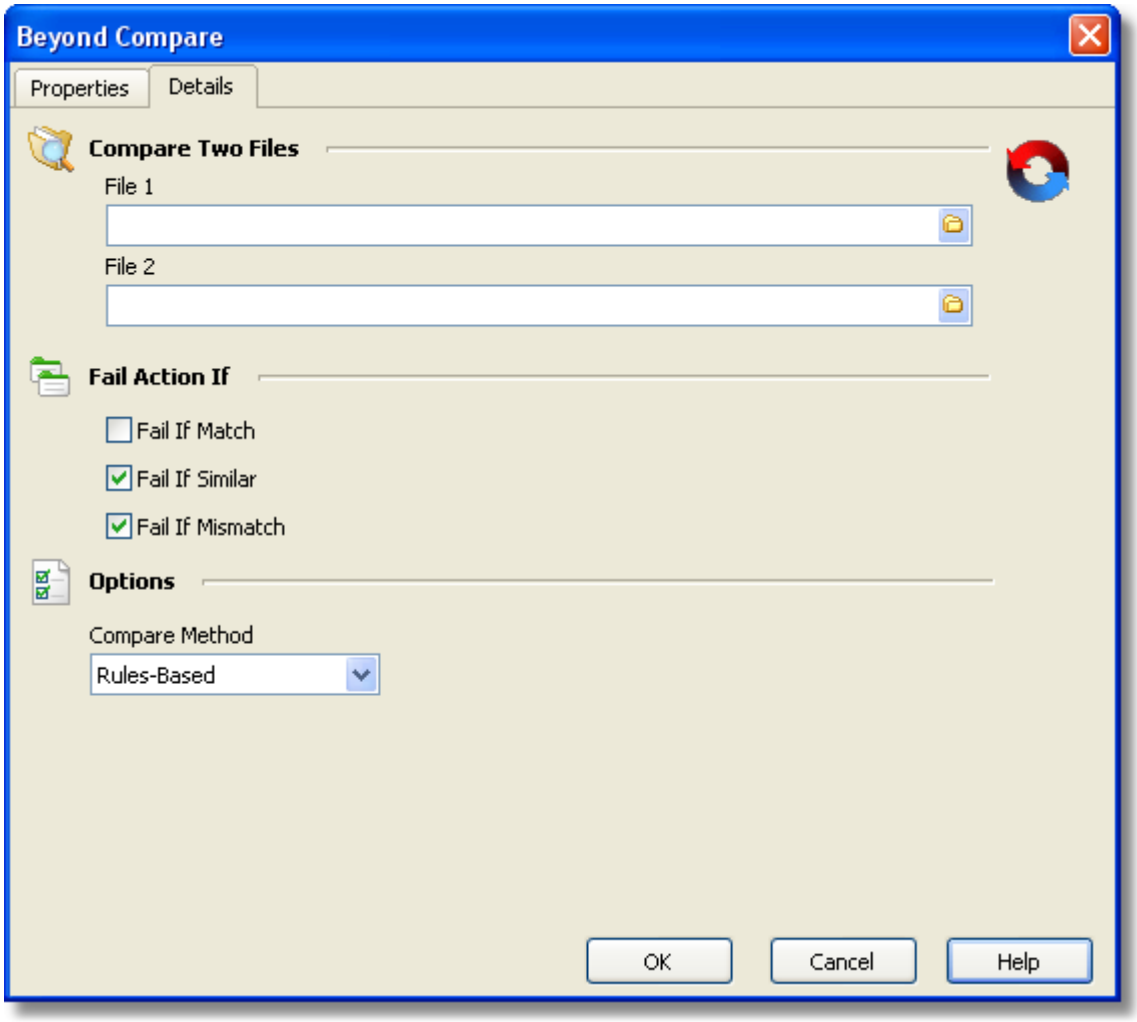
Behaviour

Choose how the result of the comparison will influence the build process. 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.

5.10.24.2 Beyond Compare Action

The Beyond Compare uses the QuickCompare mode of Scooter Software's Beyond Compare to compare the contents of two files.

© 2001-2006 VSoft Technologies Pty Ltd



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.

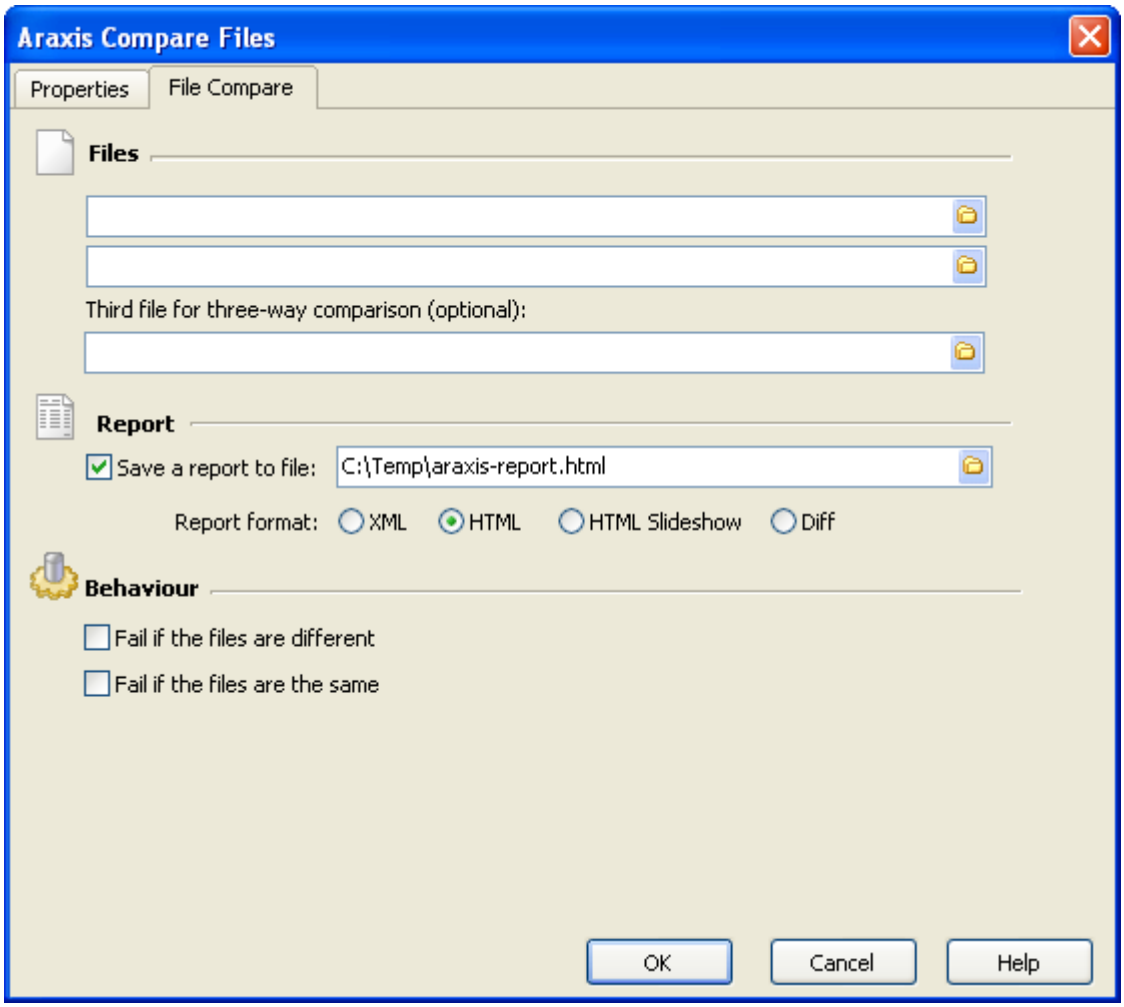
5.10.24.3 Araxis Merge Compare Actions

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

© 2001-2006 VSoft Technologies Pty Ltd

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.

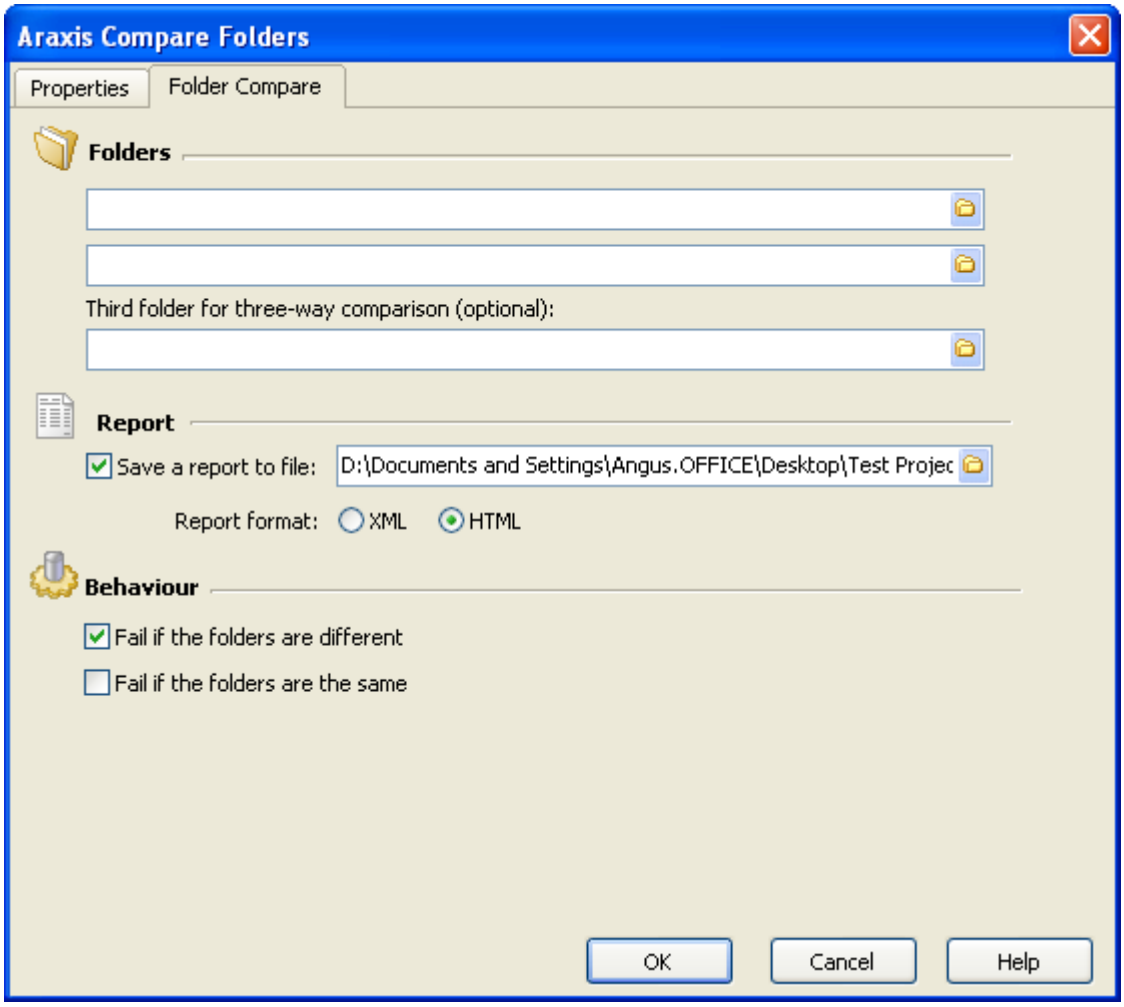
Report

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.)

© 2001-2006 VSoft Technologies Pty Ltd



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.)

© 2001-2006 VSoft Technologies Pty Ltd

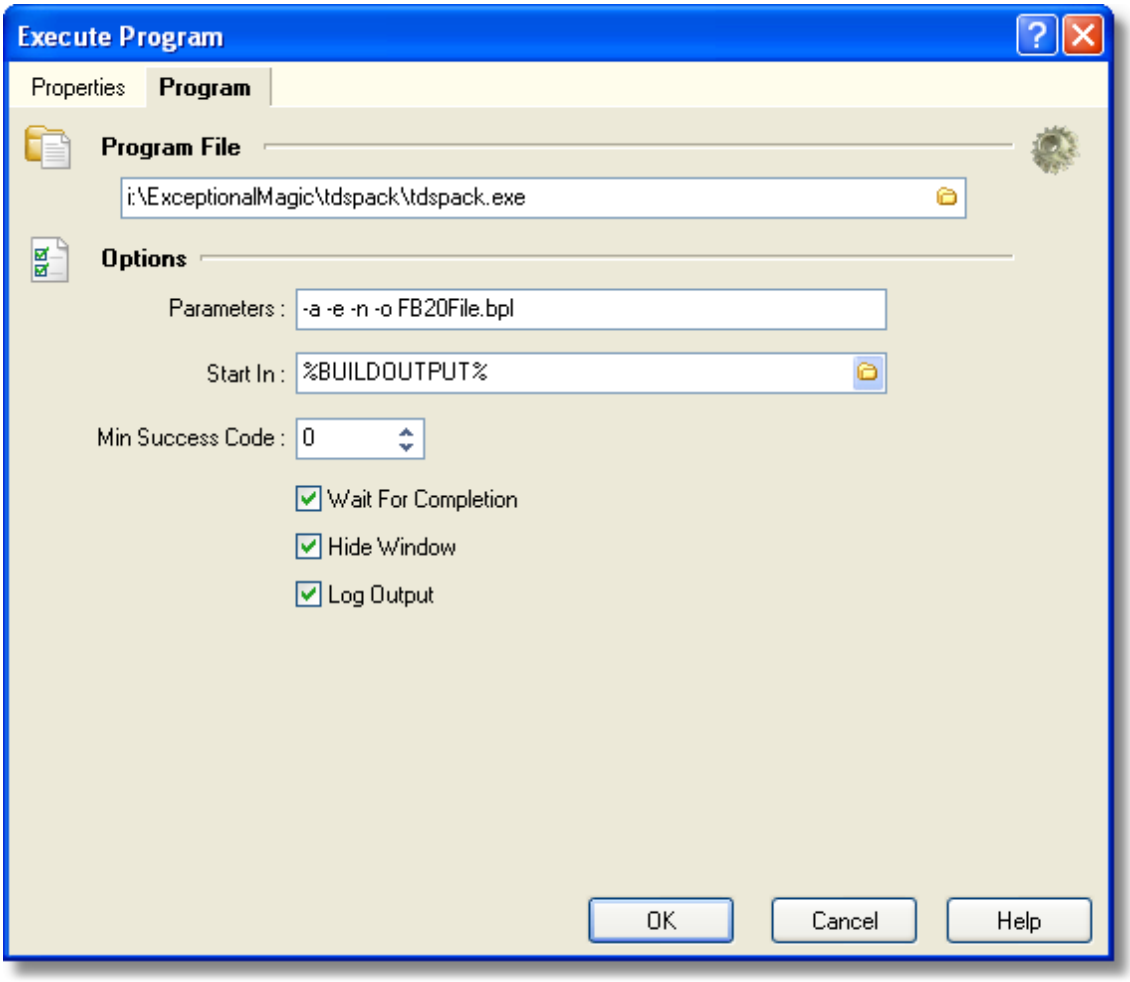
5.11 Windows OS

5.11.1 Execute Program Action

The Execute Program Action allows you to execute nearly any program from a FinalBuilder Build process.

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, use the Run DOS Command action instead.



Min Success Code

Some applications may not return a zero exit code when closing, even though they may have completed successfully. In these rare cases you can set the min success code to an appropriate value so that FinalBuilder can detect if the application executed successfully.

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

© 2001-2006 VSoft Technologies Pty Ltd

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.

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 SuccessRC : integer;
```

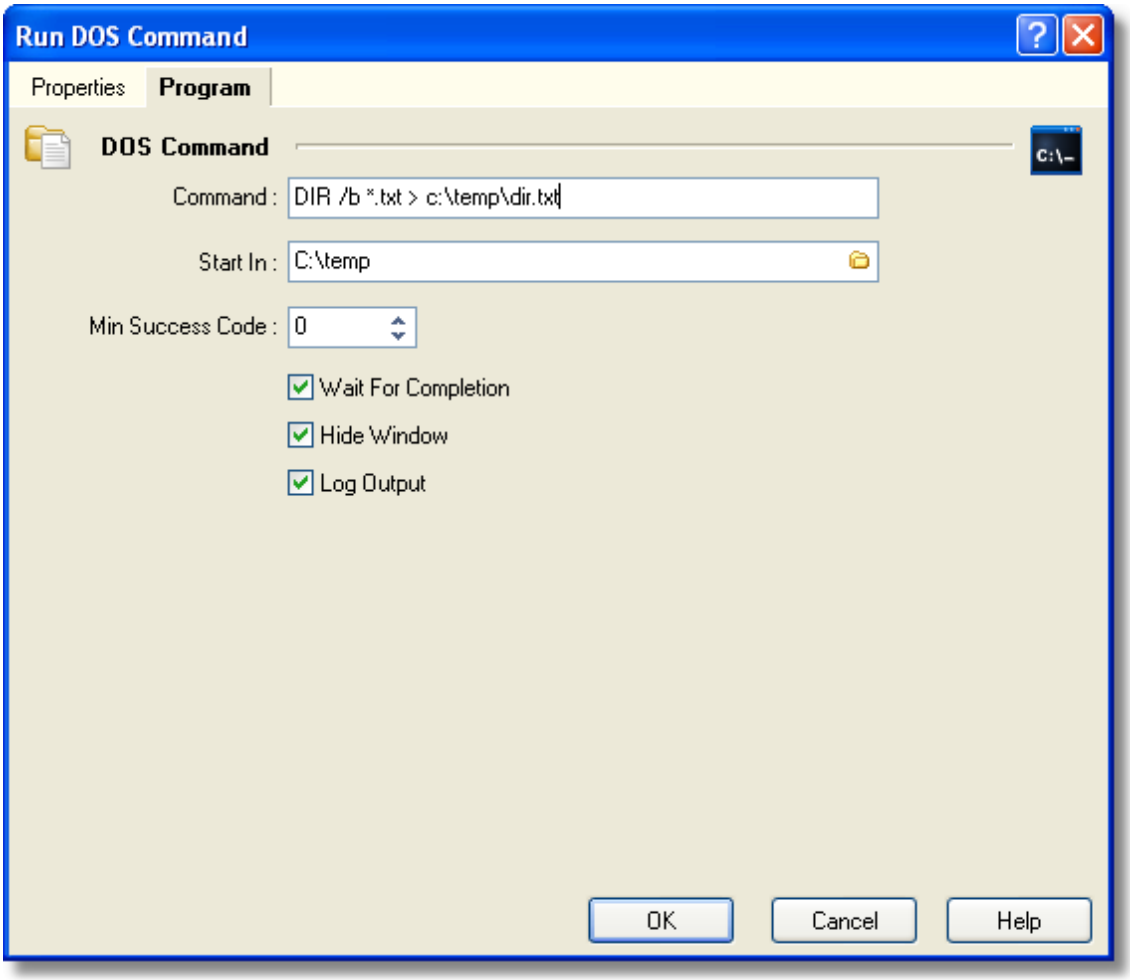
These properties may be set in the BeforeAction and AfterAction Script events.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.2 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.



Min Success Code

Some commands or batch files may not return a zero exit code when closing, even though they may have completed successfully. In these rare cases you can set the min success code to an appropriate value so that FinalBuilder can detect if the application executed successfully.

Wait For Completion

By default, FinalBuilder will wait for the command 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 command begins executing. When Wait For Completion is turned off, command output is not available for capture.

Hide Window

If this option is set, the command will not run in a visible console window.

© 2001-2006 VSoft Technologies Pty Ltd

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.

Scripting Info

The Action properties available are :

property StartInDir : WideString; // the starting directory for Dos
property LogOutput : WordBool; // capture the output of the command. Note that the output cannot always be captured (for example XCopy)
property WaitForCompletion : WordBool; //wait for this command to complete for continuing to the next action
property ErrorCode : integer; //read only
property HideWindow : WordBool; // Hide the window while executing the dos command
property SuccessRC : integer; // this is the lowest return code that should be deemed a successful execution. Some applications return a non-zero code even though they do actually execute ok.
property Command : WideString; // the dos command!

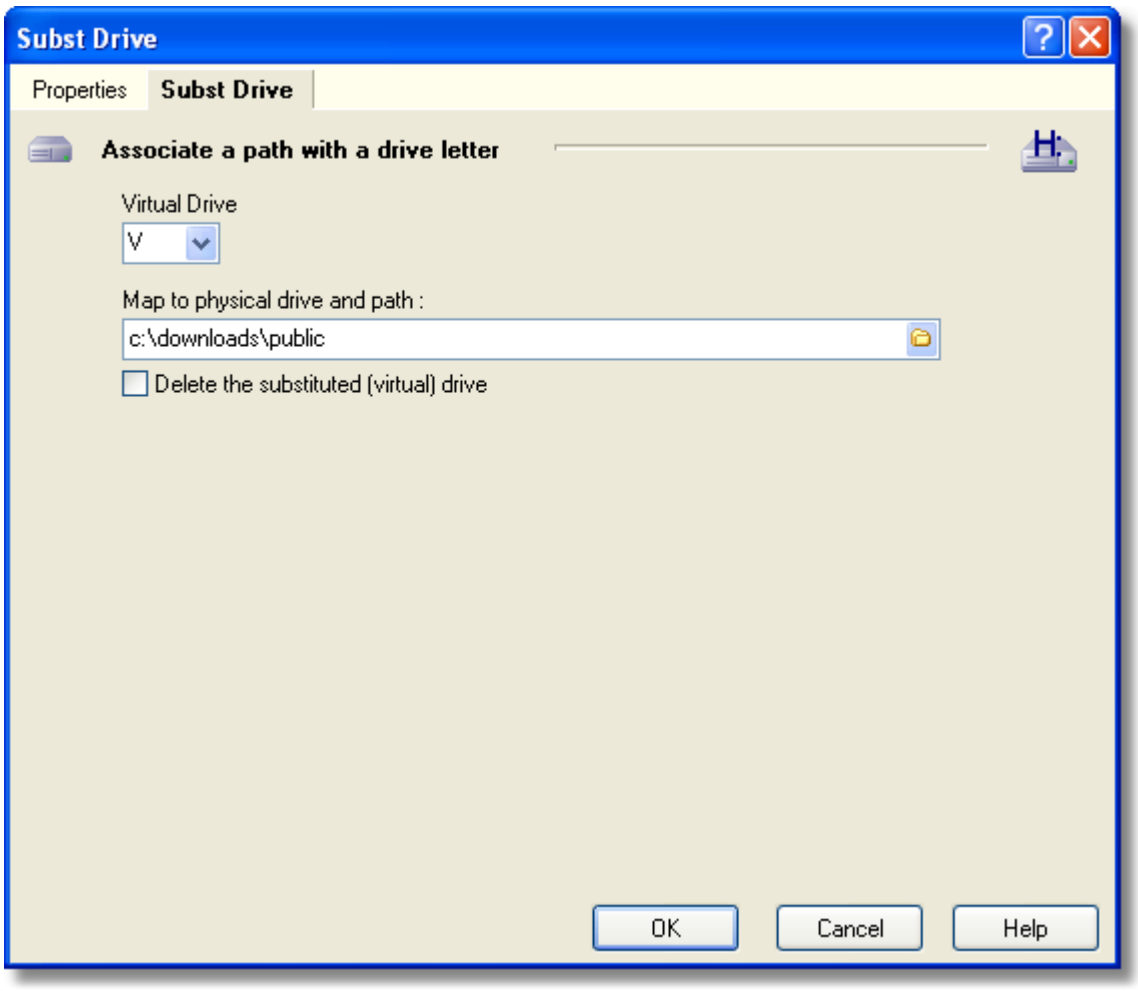
These properties may be set in the BeforeAction and AfterAction Script events.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.3 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.

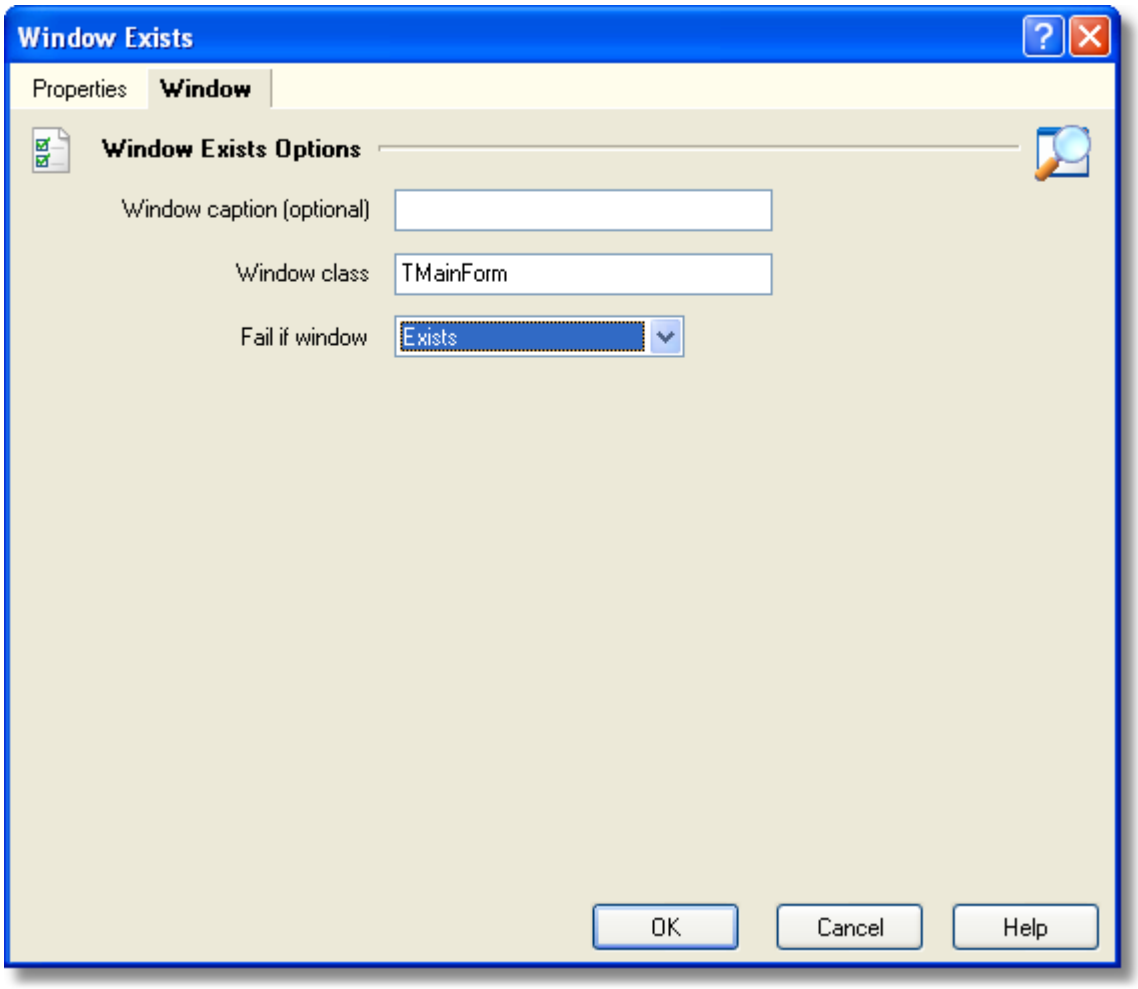


© 2001-2006 VSoft Technologies Pty Ltd

5.11.4 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.

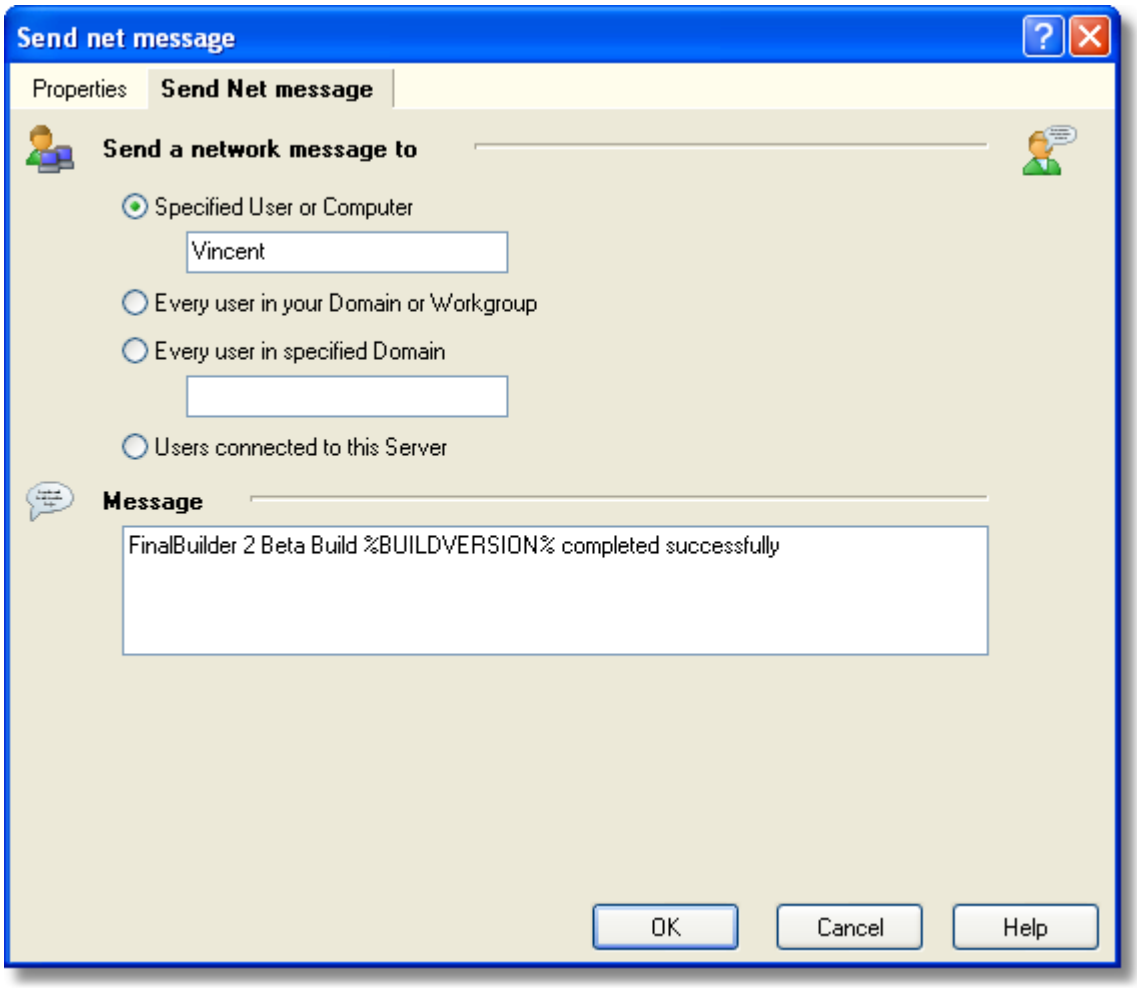
This action was written and kindly donated by Erik Berry, maintainer of GExperts



© 2001-2006 VSoft Technologies Pty Ltd

5.11.5 Net Send Message Action

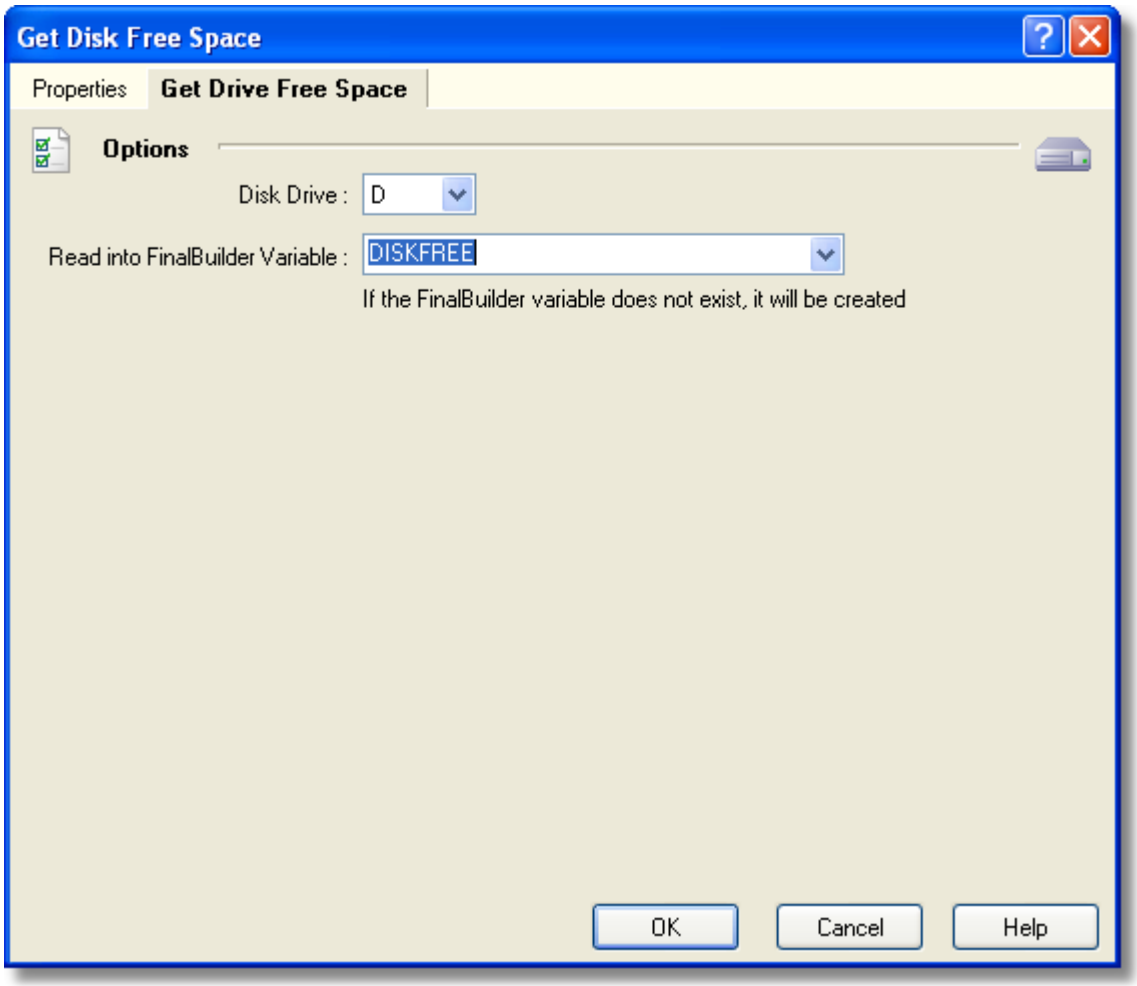
Send a Message using windows NET SEND



© 2001-2006 VSoft Technologies Pty Ltd

5.11.6 Get Disk Free Space

Reads the free disk space of a drive into a FinalBuilder variable.



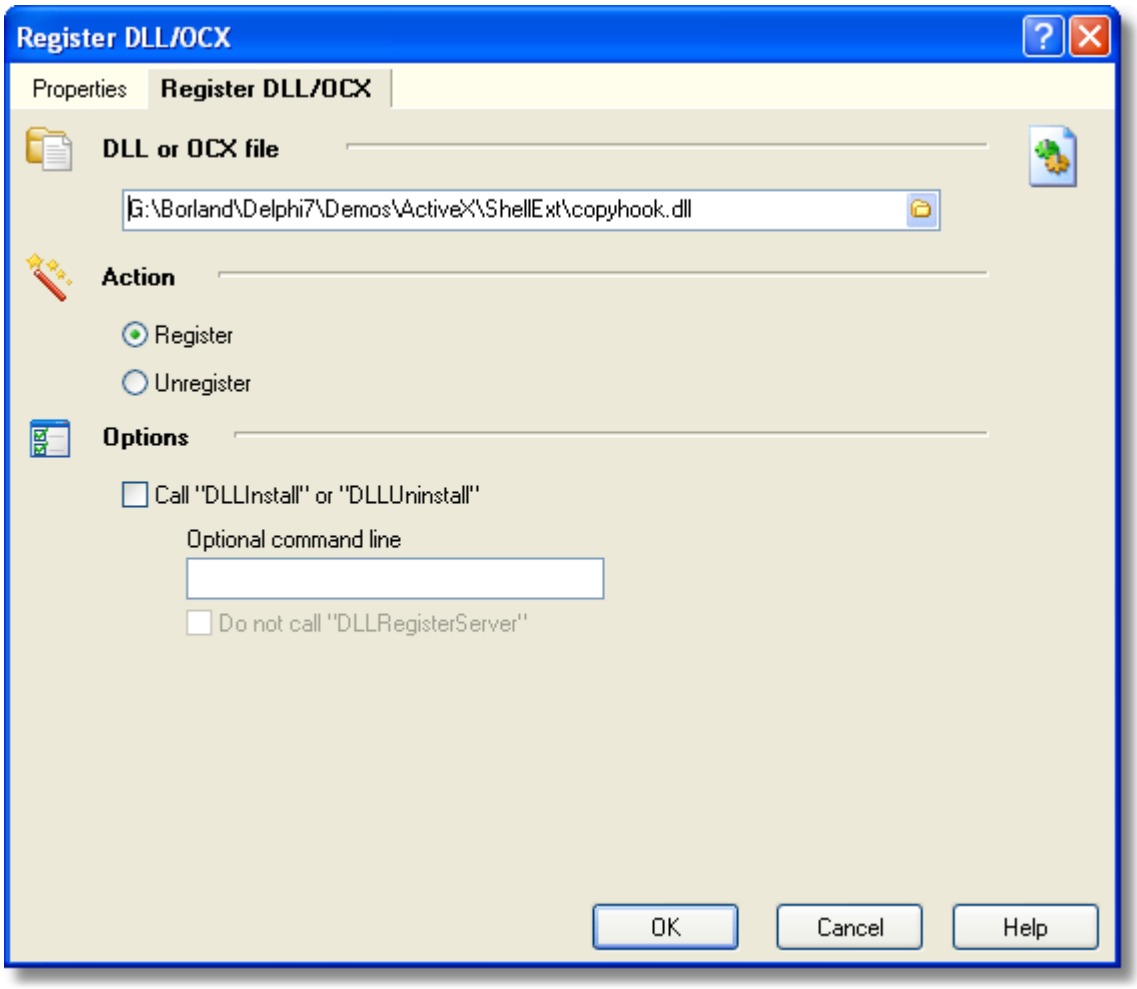
The disk space available on the drive is returned in BYTES.
If the drive specified is not available, then the action will fail.

If you need to dynamically set the Disk Drive during your build, then you can access the Drive property in script. eg:
BeforeAction script:
Action.Drive = 'R'
or
Action.Drive = MyDynamicDriveLetter

© 2001-2006 VSoft Technologies Pty Ltd

5.11.7 Register DLL/OCX Action

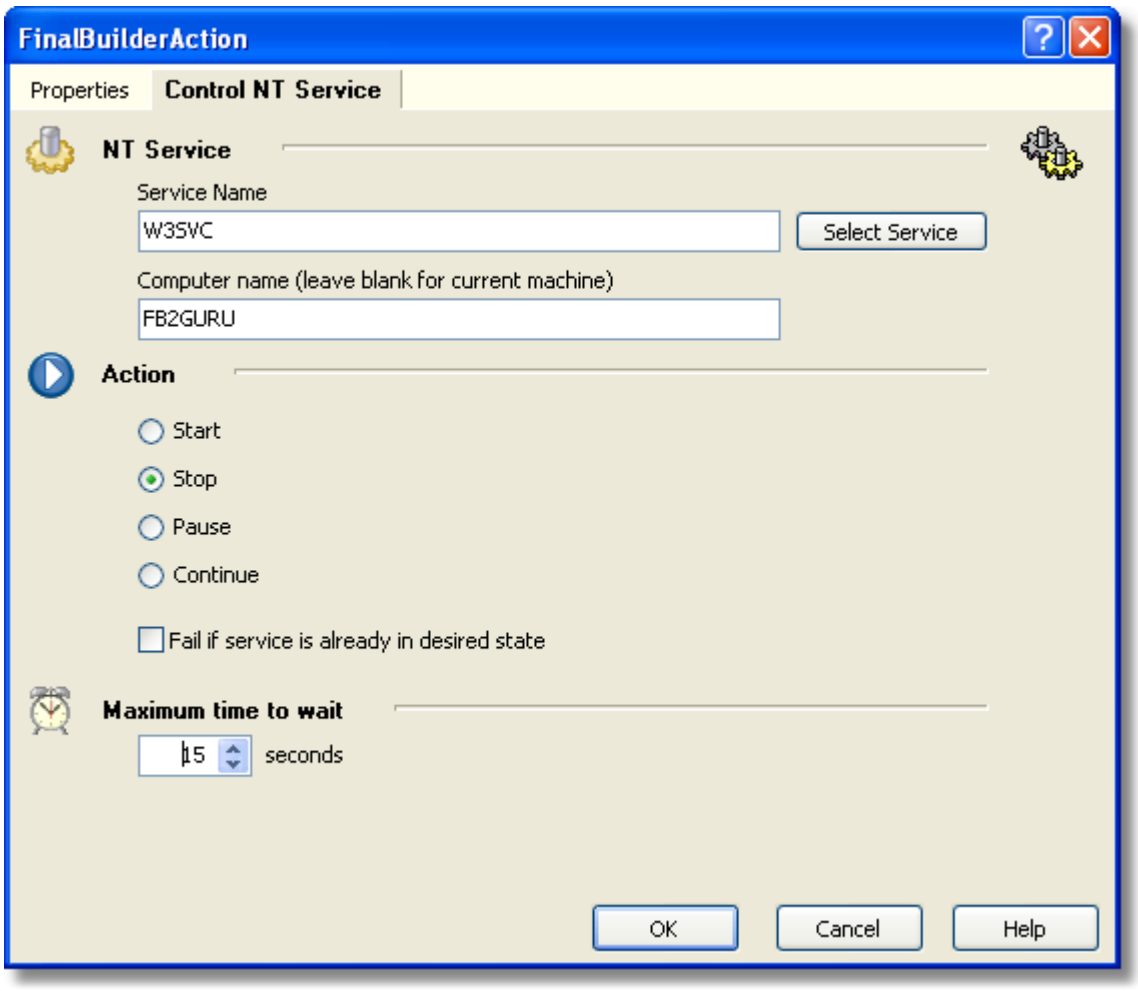
Register an activeX control (OCX) or COM DLL.



© 2001-2006 VSoft Technologies Pty Ltd

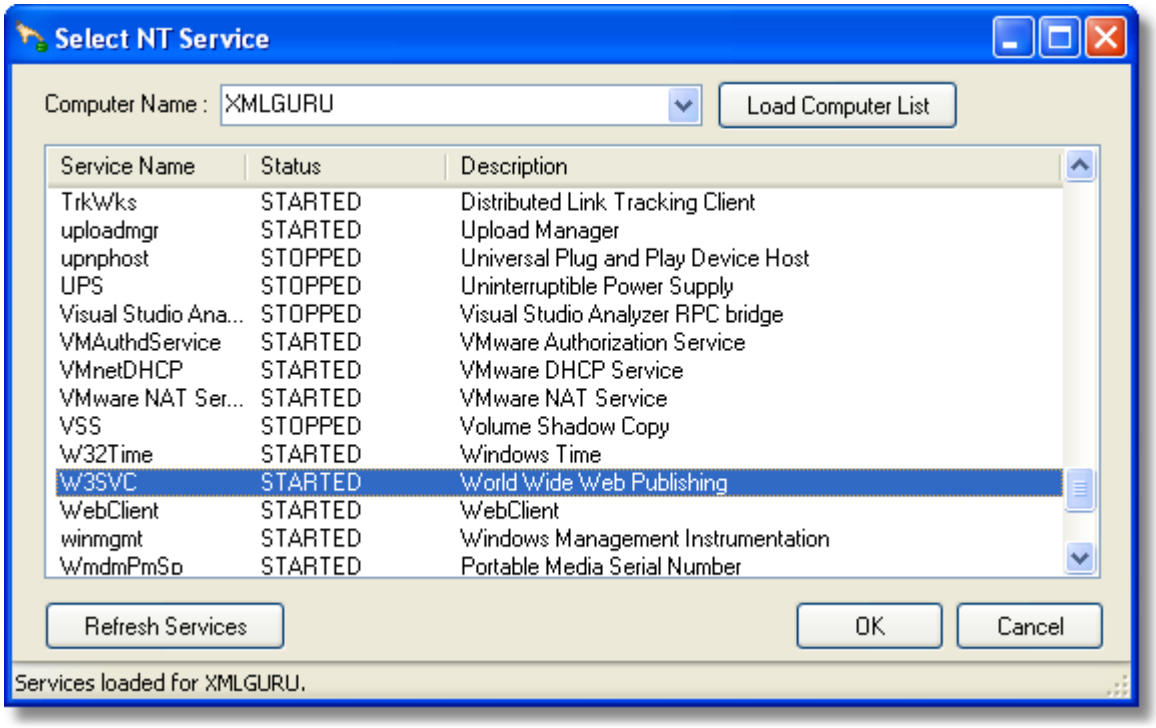
5.11.8 Control Service Action

[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.

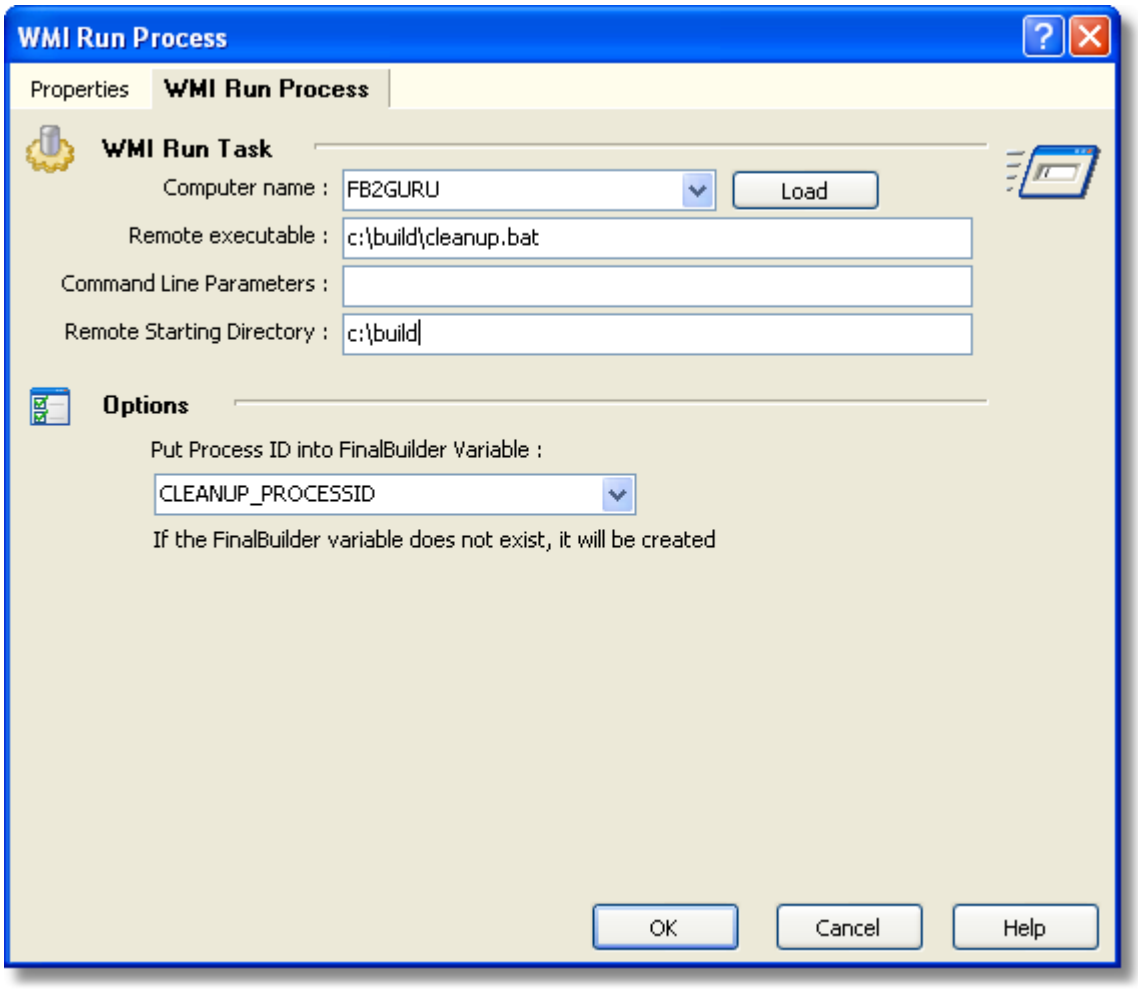
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.11.9 WMI Run Process Action

[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.

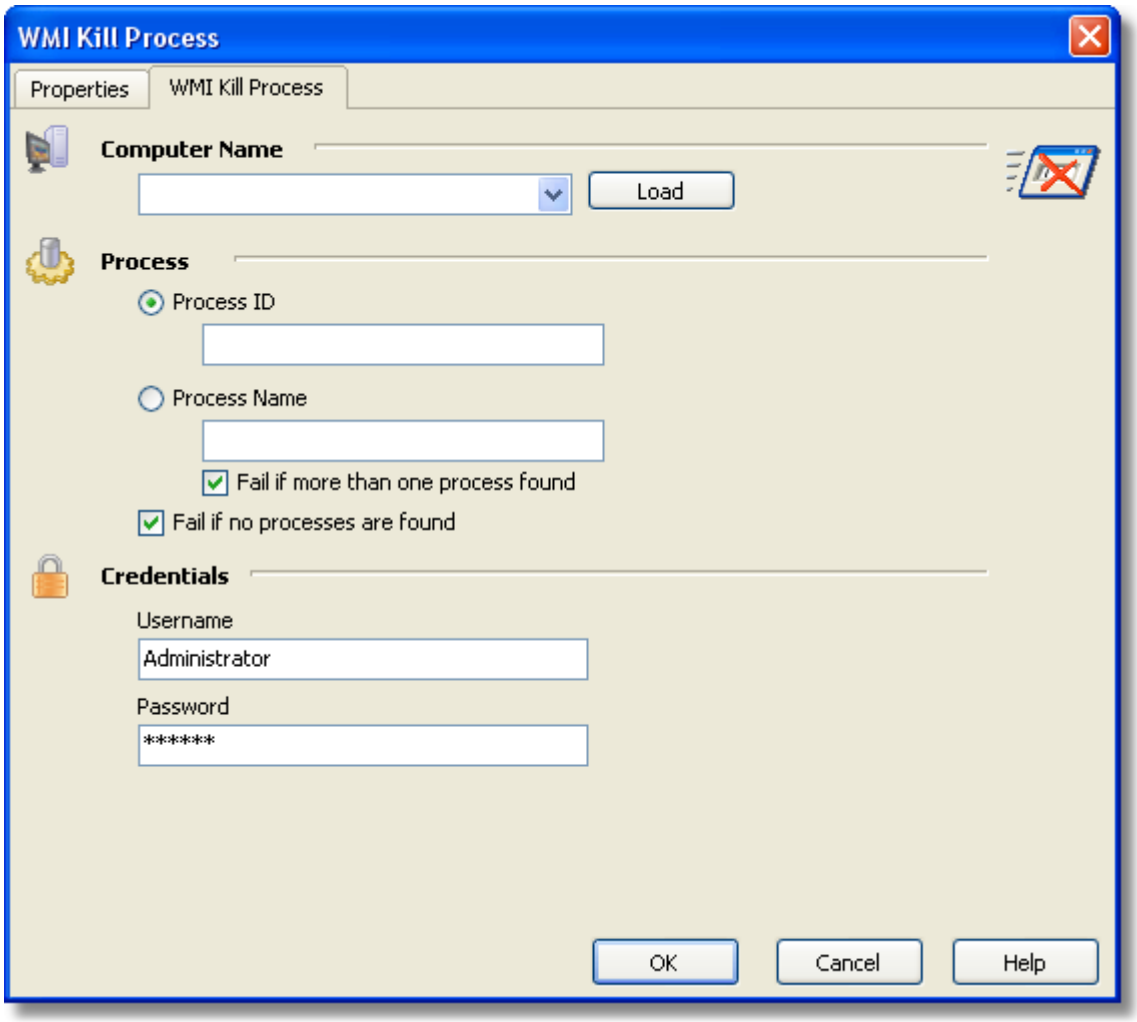


You can choose to save the process ID in a FinalBuilder variable so that it may be used in later actions such as the WMI Kill Process Action.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.10 WMI Kill Process Action

[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.



Process

You can specify the remot 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

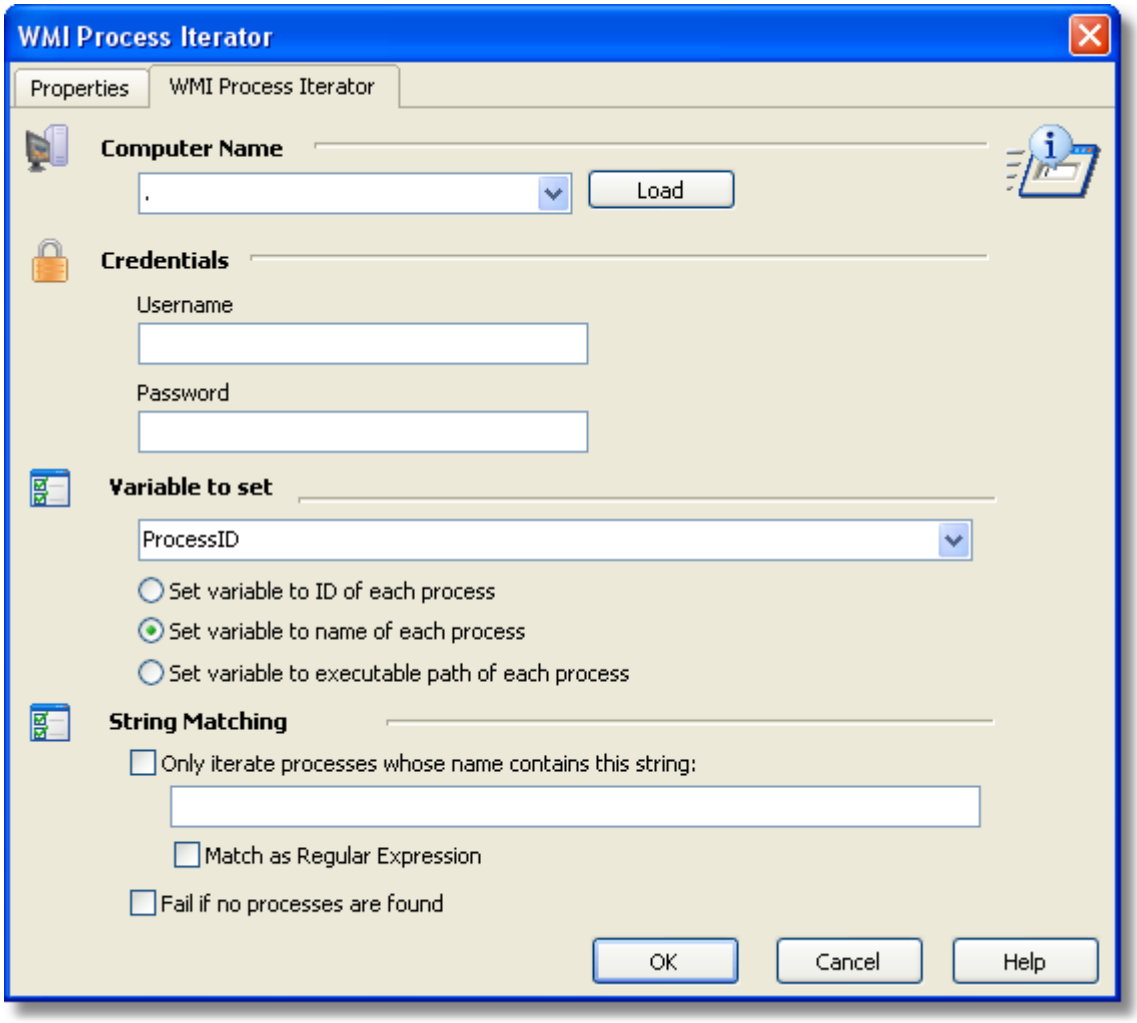
Specify credentials for the WMI connection. If no credentials are specified, WMI uses the currently logged in user's Windows Authentication credentials.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.11 WMI Process Iterator

The WMI process iterator allows you to iterate through a list of processes running on any Windows machine. Each process can be identified by it's process ID or name. The list of processes can be filtered to match an executable name or regular expression.

For general information about iterators, click here.



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

© 2001-2006 VSoft Technologies Pty Ltd

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.

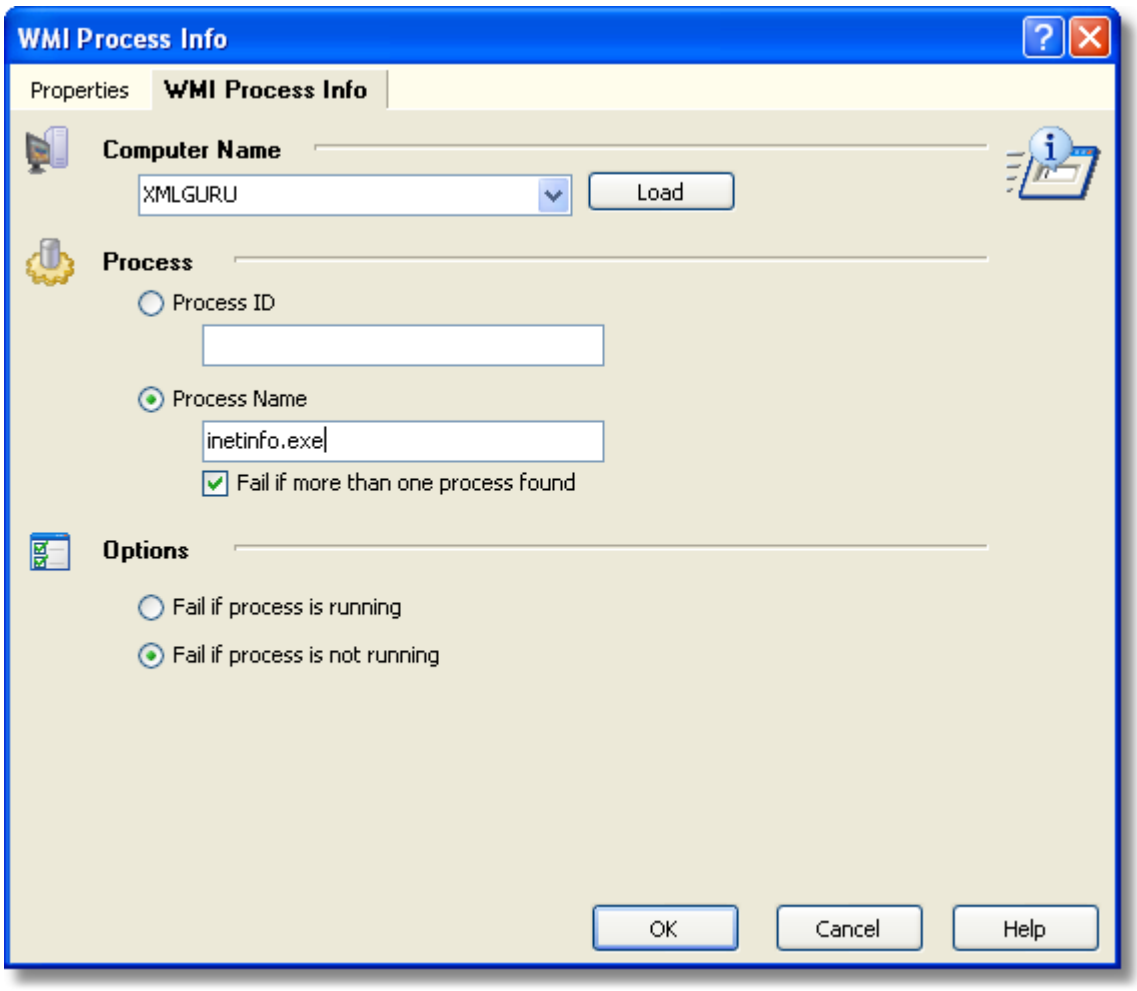
Fail if no processes are found

If this option is selected, the action will fail outright if no processes are found which match the chosen criteria.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.12 WMI Process Info Action

[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;
```

© 2001-2006 VSoft Technologies Pty Ltd

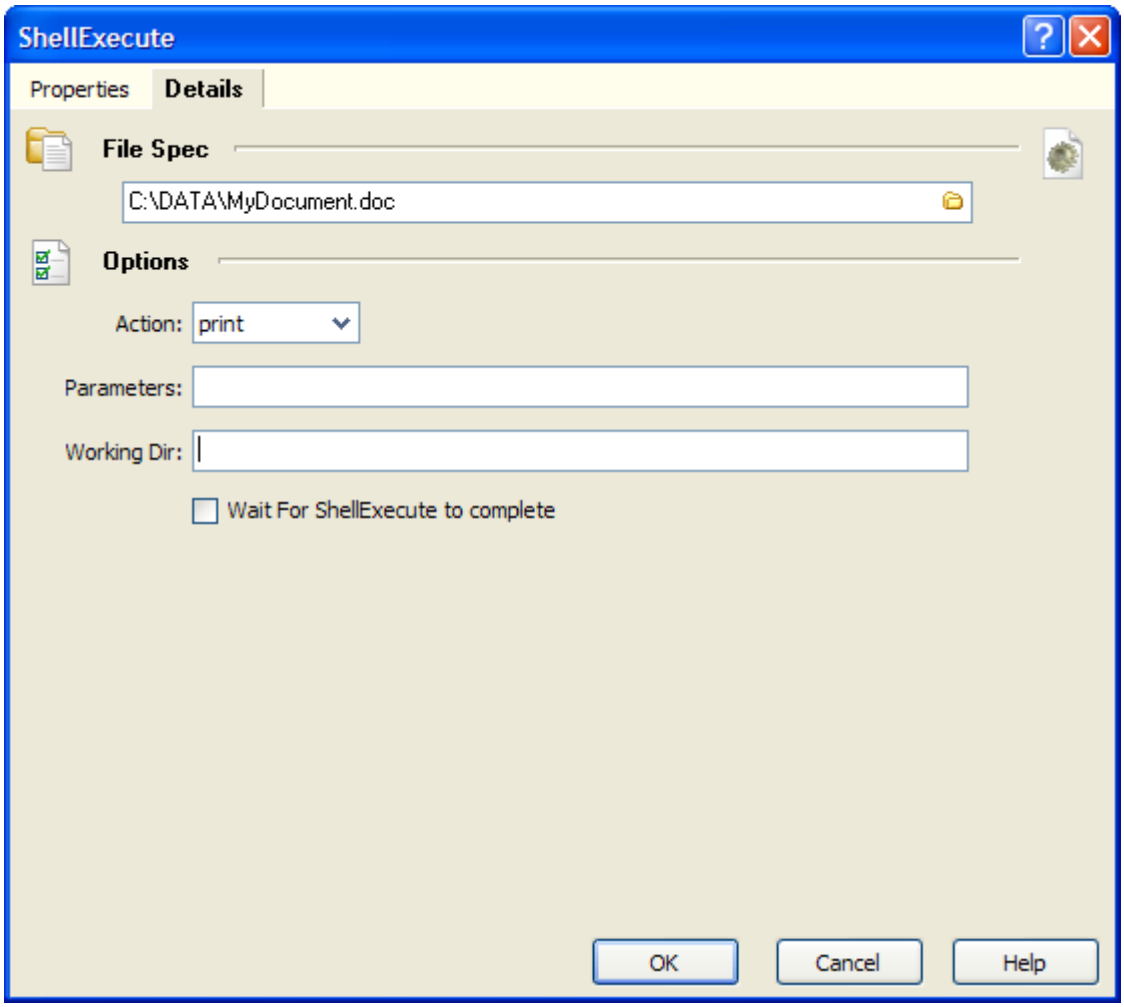
```
property ProcessPeakPageFileUsage        : LongWord;
property ProcessPeakVirtualSize          : LongWord;
property ProcessCreationDate             : TDateTime;
property ProcessKernelModeTime           : LongWord;
property ProcessMaxWorkingSetSize        : LongWord;
property ProcessMinWorkingSetSize        : 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;
```

© 2001-2006 VSoft Technologies Pty Ltd

5.11.13 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:

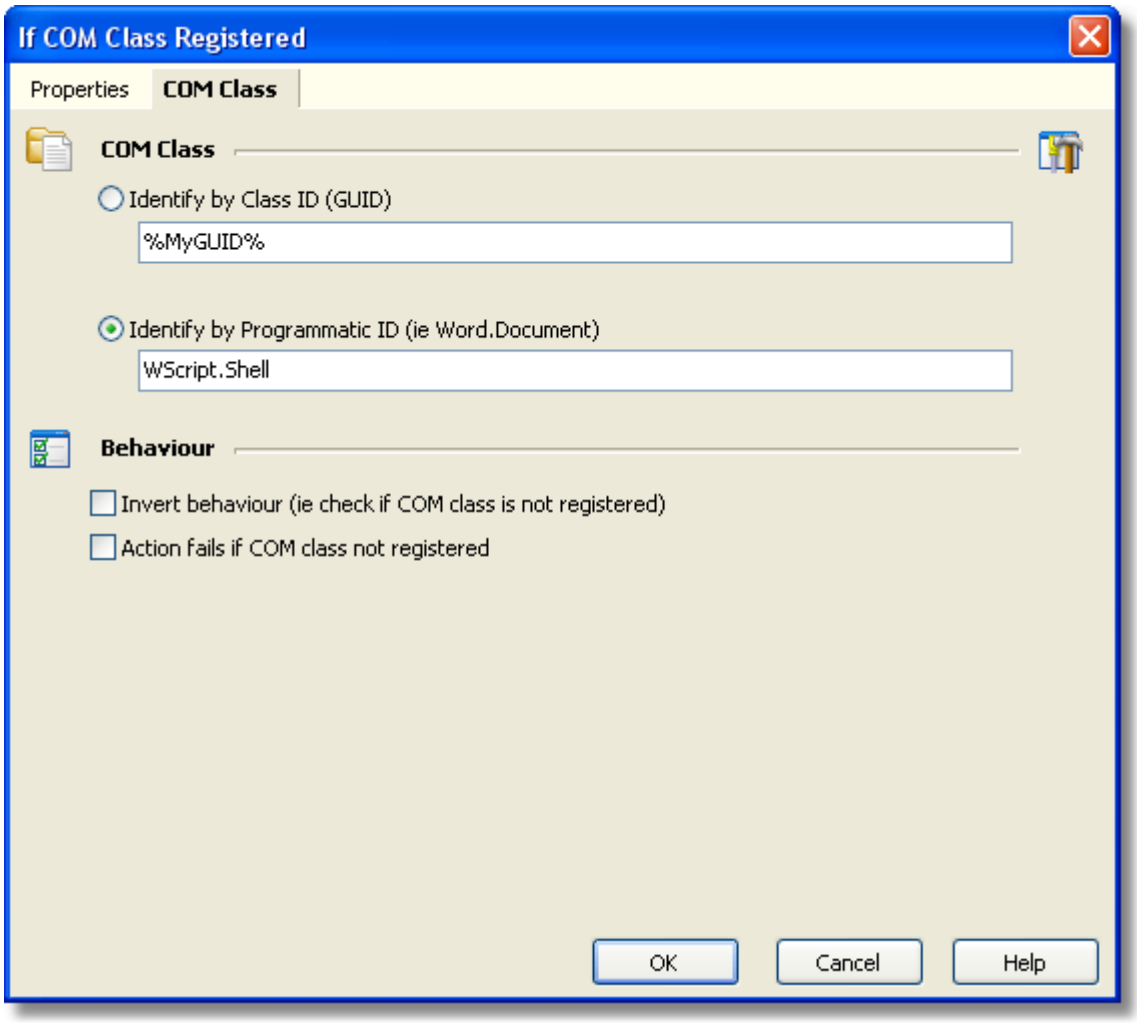


© 2001-2006 VSoft Technologies Pty Ltd

5.11.14 If COM Class Registered

The If COM Class Registered action enables you to control the flow of your build process 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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

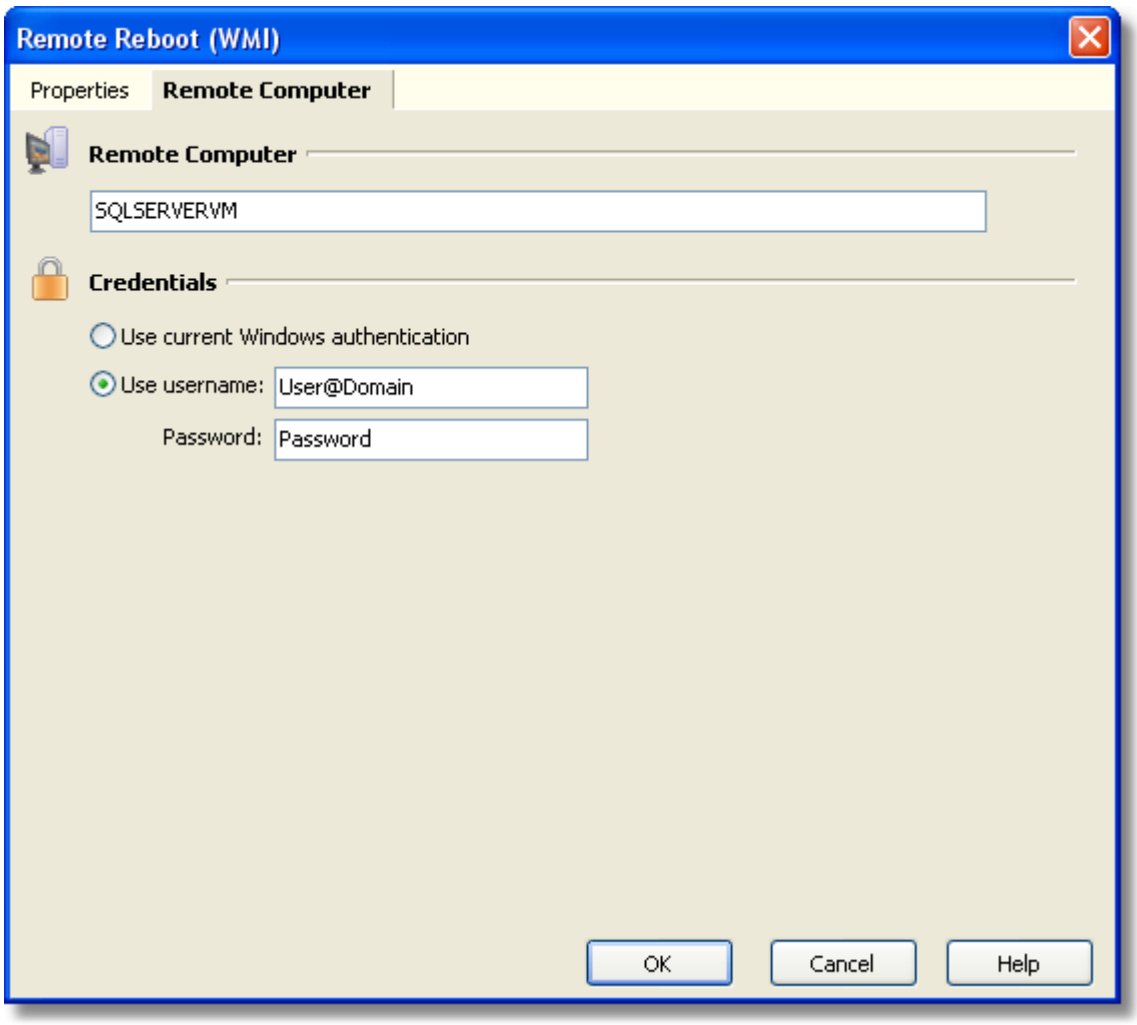
Action fails if COM class not registered

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.'

© 2001-2006 VSoft Technologies Pty Ltd

5.11.15 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.

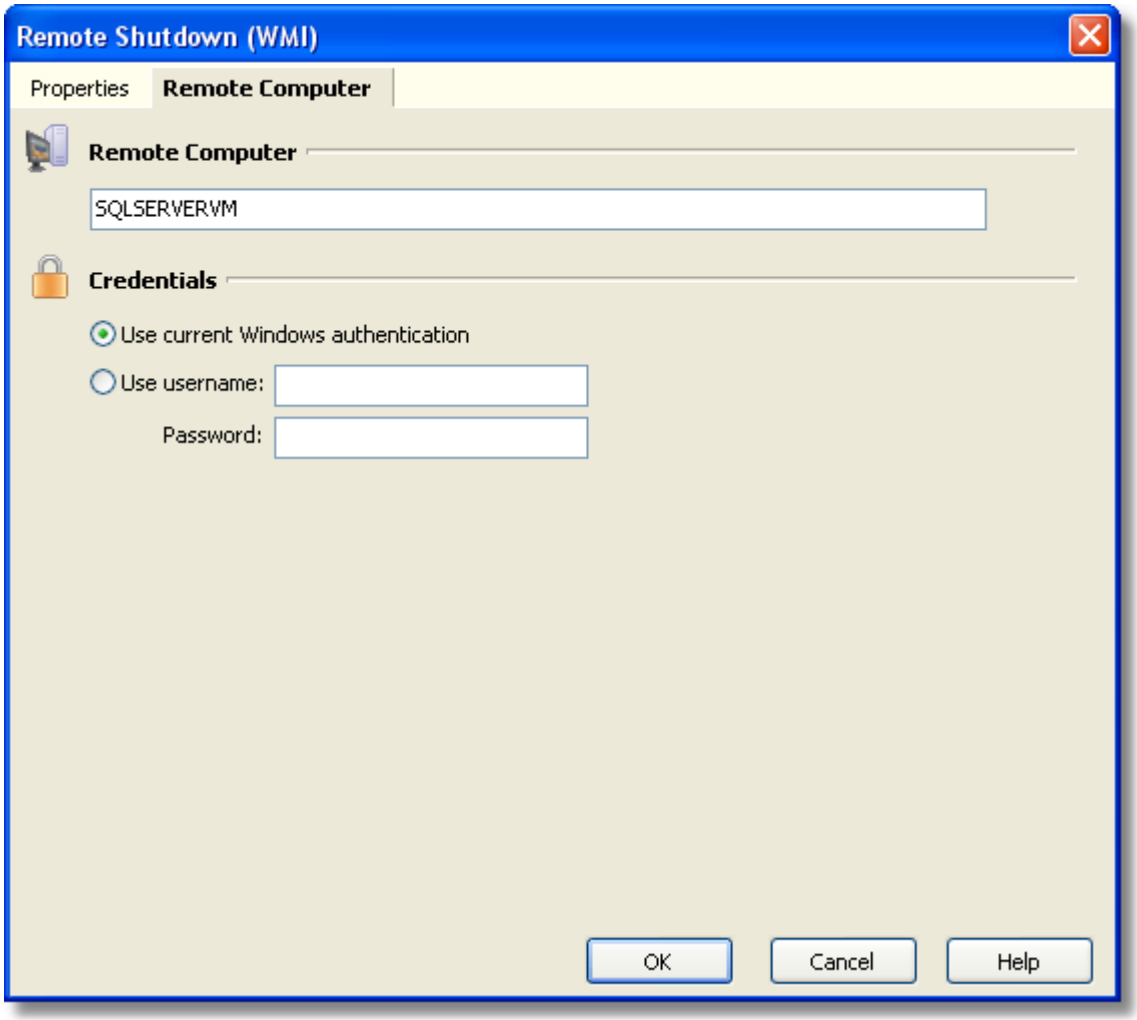
Credentials

WMI can either use the current Windows account, or a specific user account, for authentication.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.16 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.

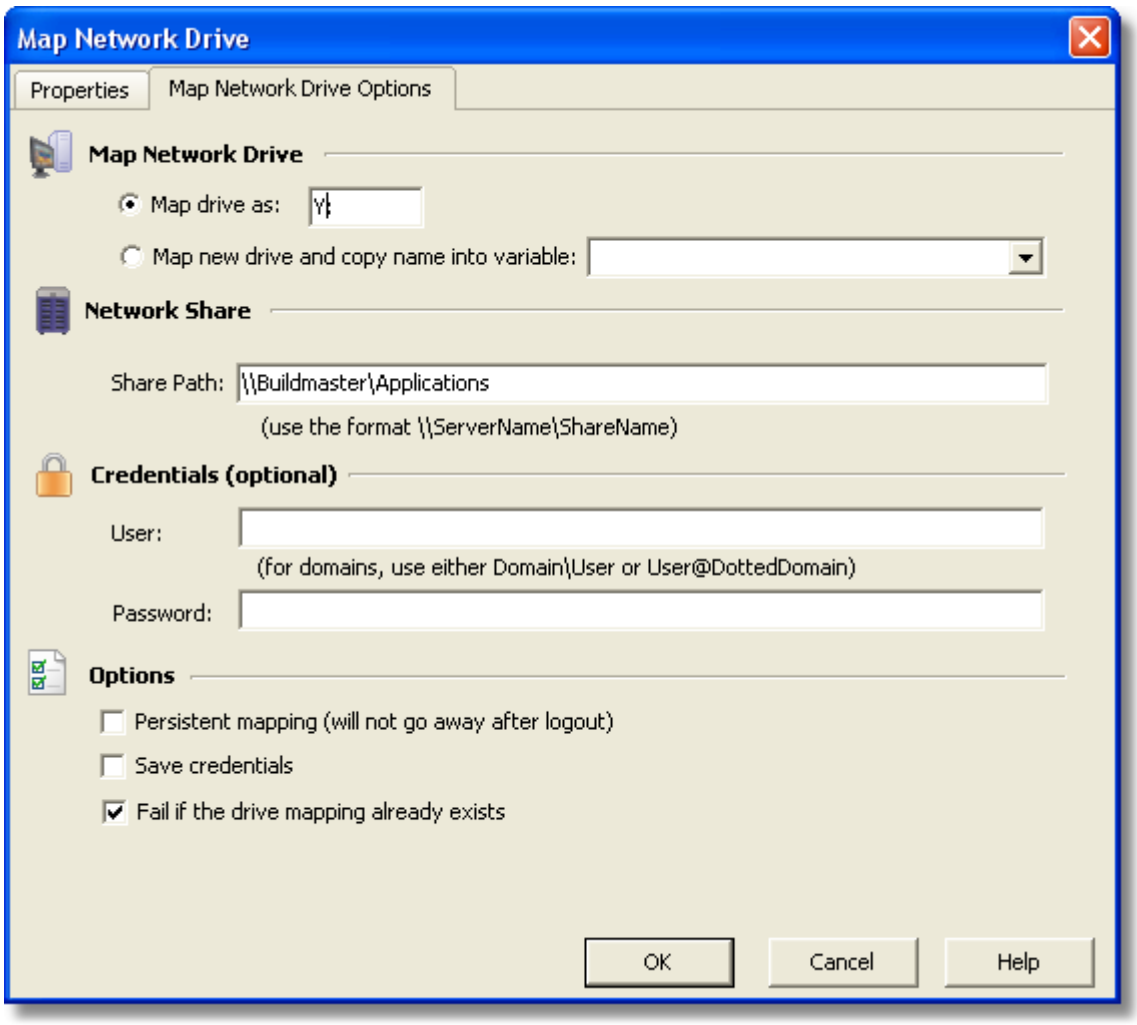
Credentials

WMI can either use the current Windows account, or a specific account, for authentication.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.17 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

© 2001-2006 VSoft Technologies Pty Ltd

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 process.

Save Credentials

Saves the login and password for use with future connection attempts.

Fail if the drive mapping already exists

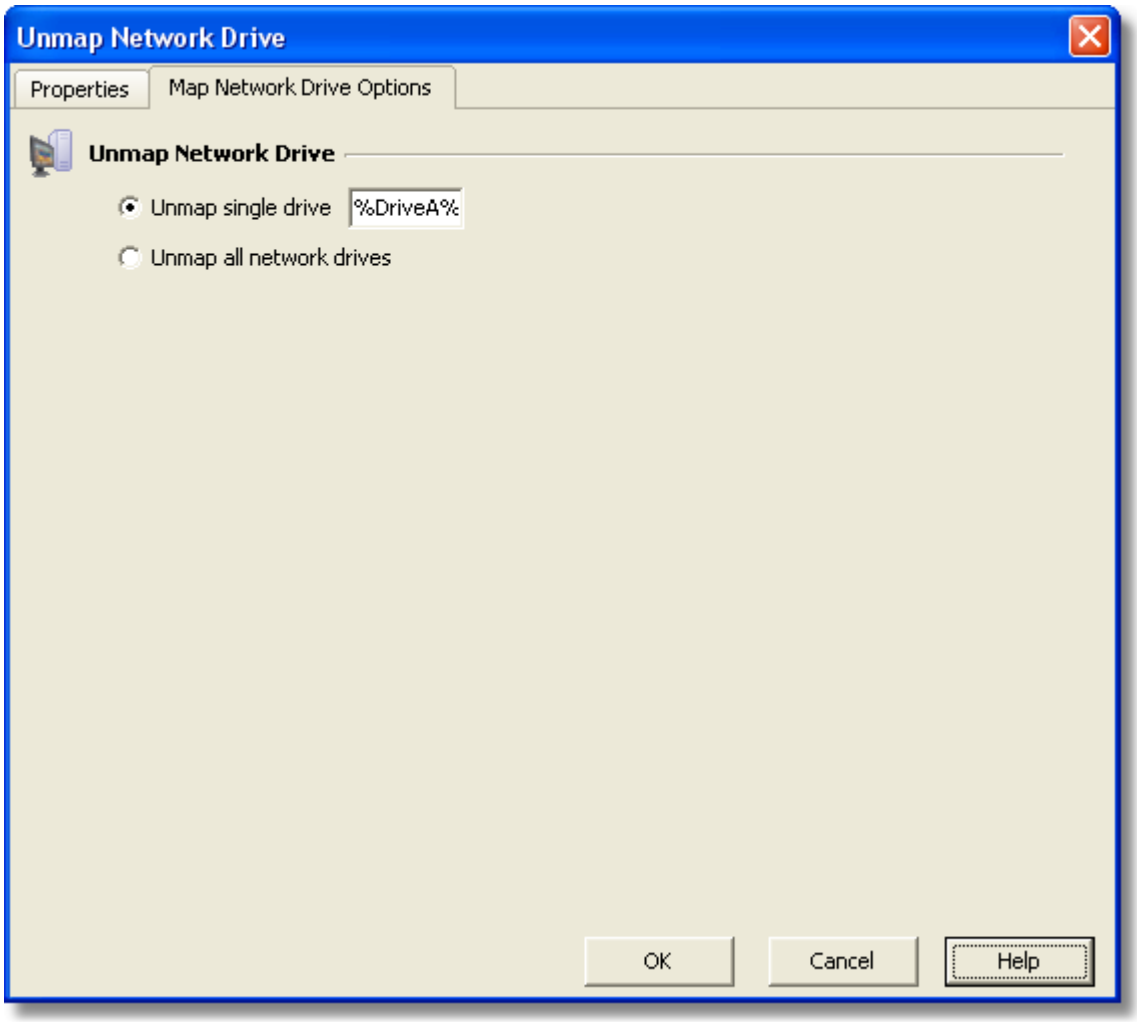
If this option is checked, the action will fail if a drive is already mapped to that drive name. If unchecked, the action will skip the mapping if the drive already exists (note that the action does not check which share is mapped to the drive, nor that the drive is in fact a drive mapping and not a real disk drive!)

This option has no effect if you are mapping a new drive name.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.18 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

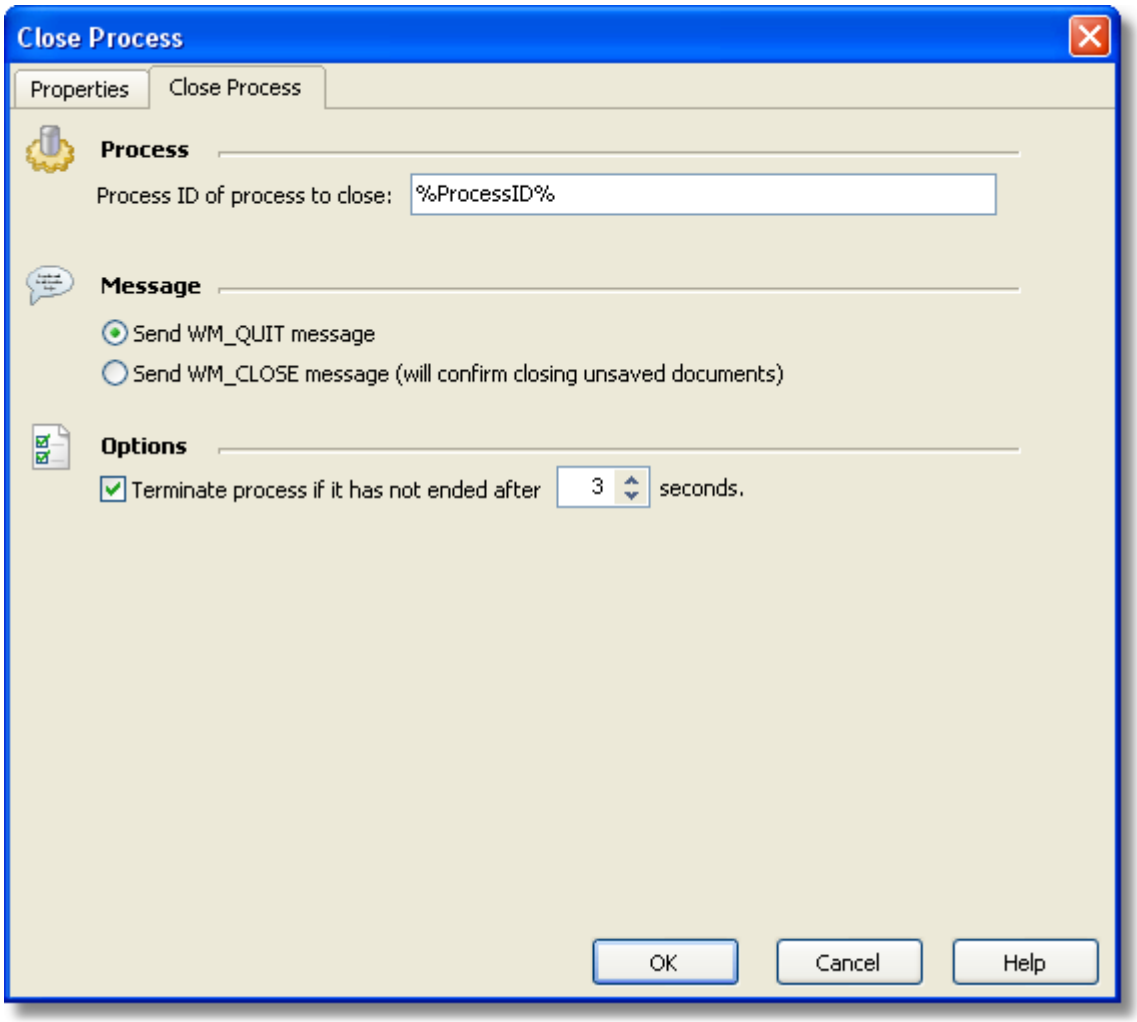
This option will attempt to remove all mapped network drives from the system. Note that the action does not distinguish between drives mapped by FinalBuilder and drives mapped by other processes.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.19 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.)

© 2001-2006 VSoft Technologies Pty Ltd

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.)

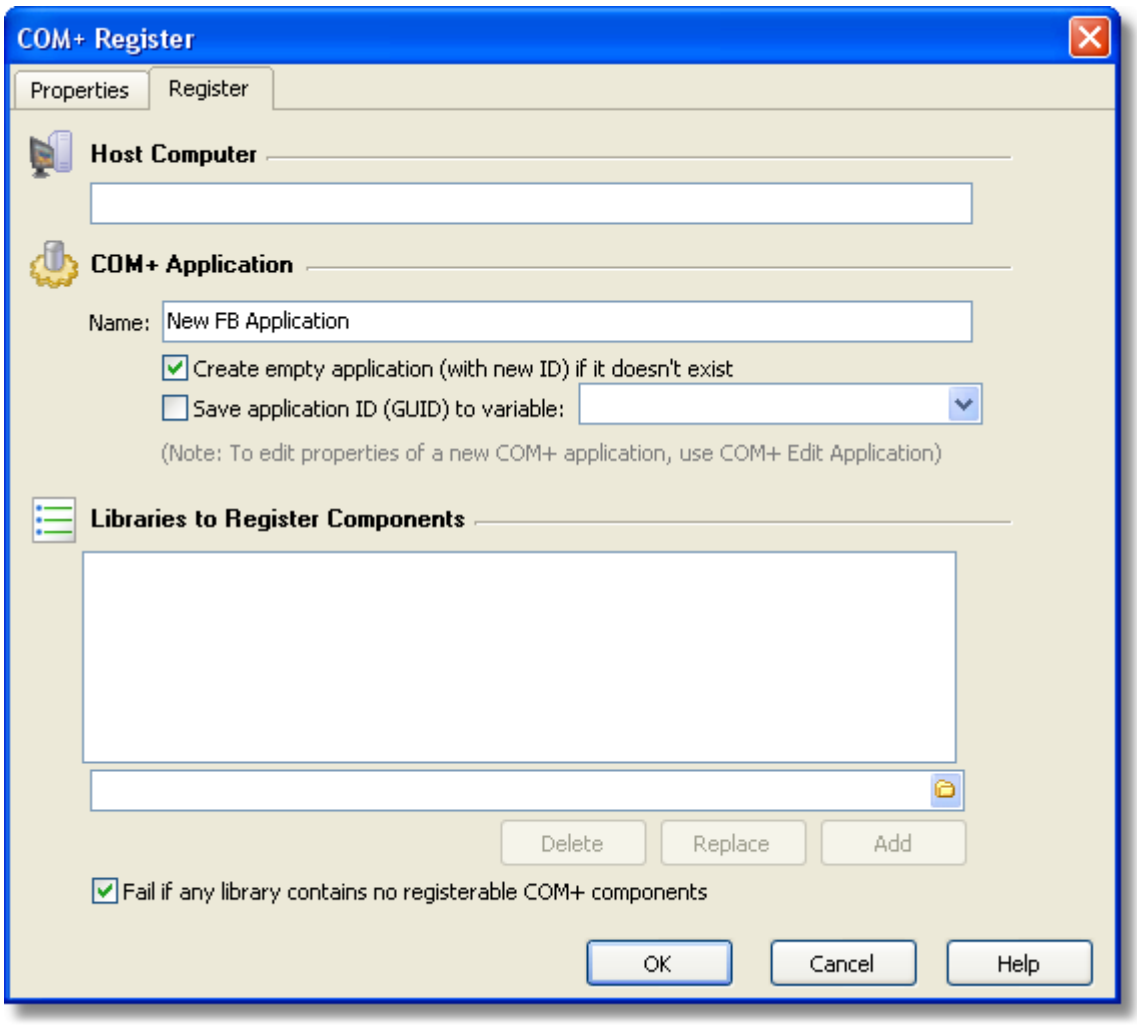
If this option is not checked, the action will not pause to confirm whether or not the process has ended.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.20 COM+ Administration

5.11.20.1 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.

© 2001-2006 VSoft Technologies Pty Ltd

Save application ID (GUID) to variable

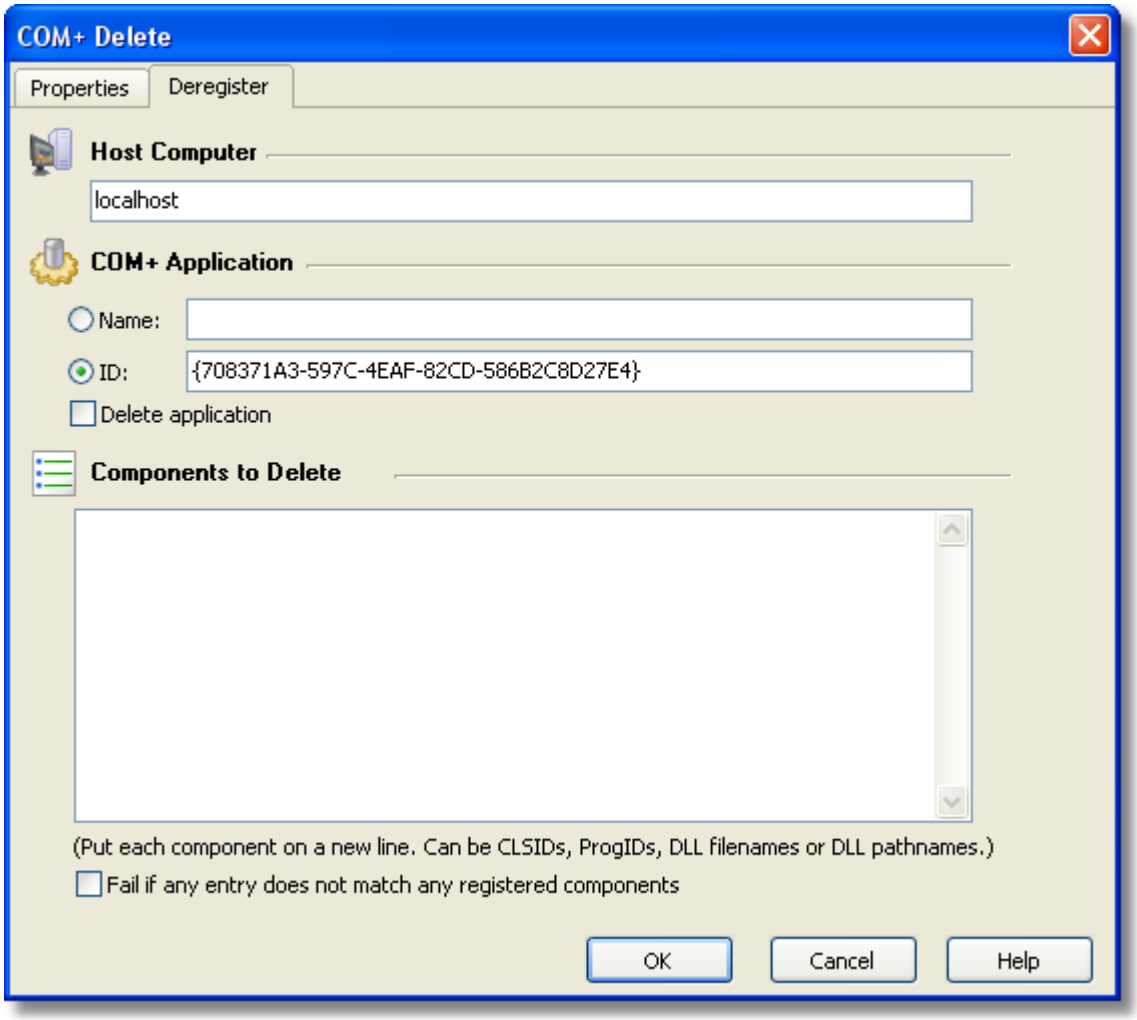
If this option is checked, then the unique ID string of the COM+ Application (existing or new) will be saved to a FinalBuilder variable.

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.

5.11.20.2 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

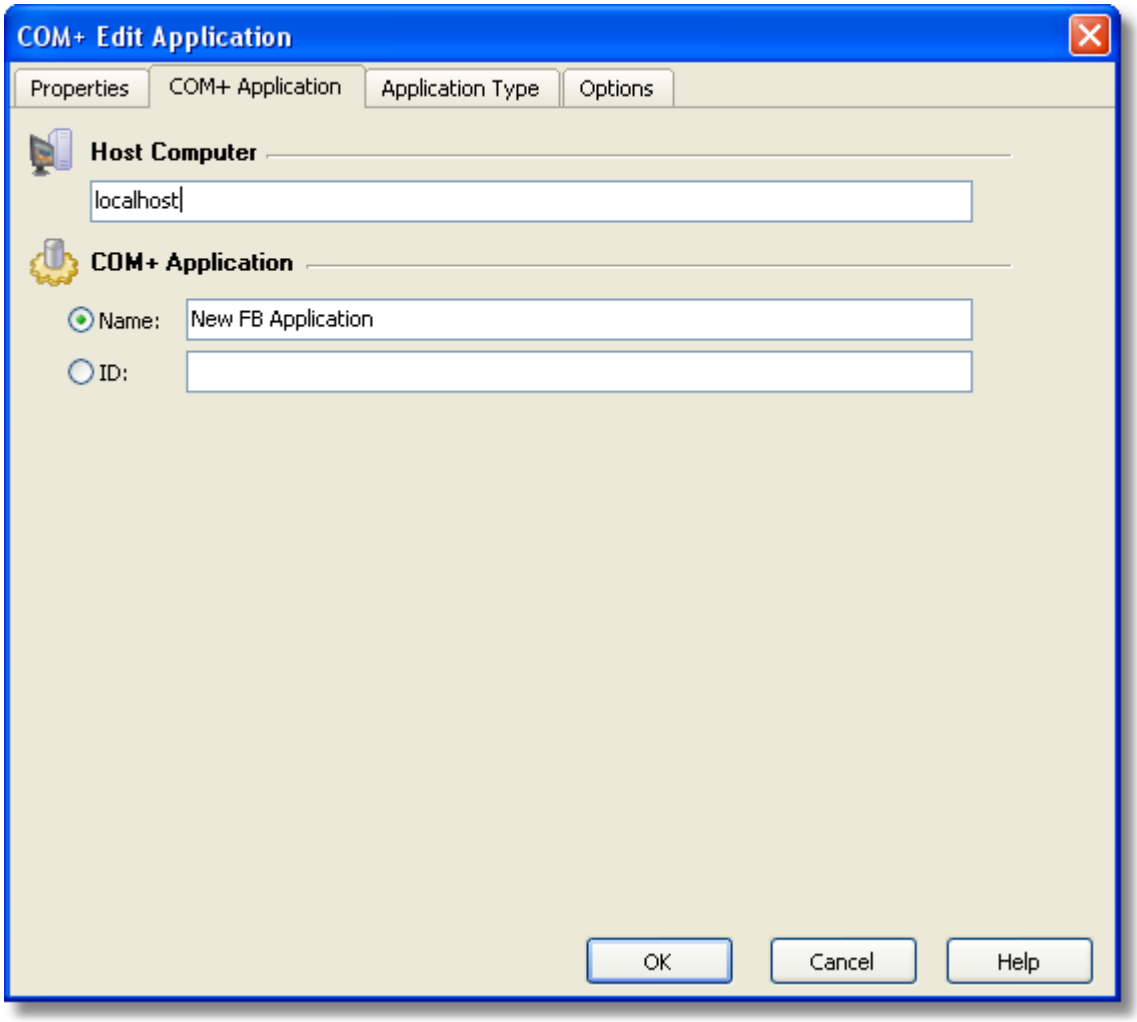
Components can be specified as component names (ProgIDs, as shown in the Component Services Components list), CLSIDs (GUIDs) or DLL filenames or pathnames (partial pathnames are also acceptable.) If DLL names are specified, all components belonging to that DLL will be deleted.

5.11.20.3 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.

© 2001-2006 VSoft Technologies Pty Ltd



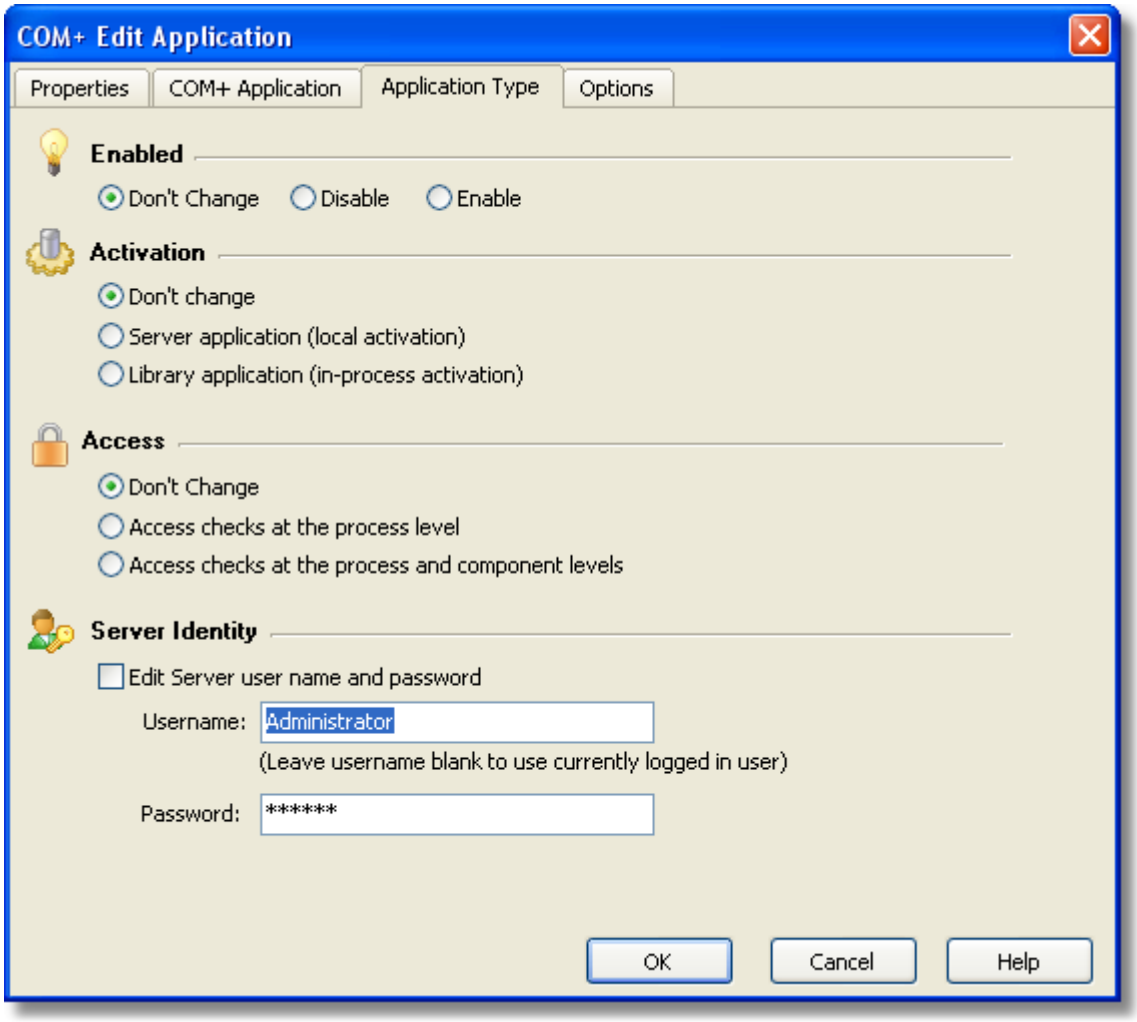
Host Computer

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.)

© 2001-2006 VSoft Technologies Pty Ltd



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

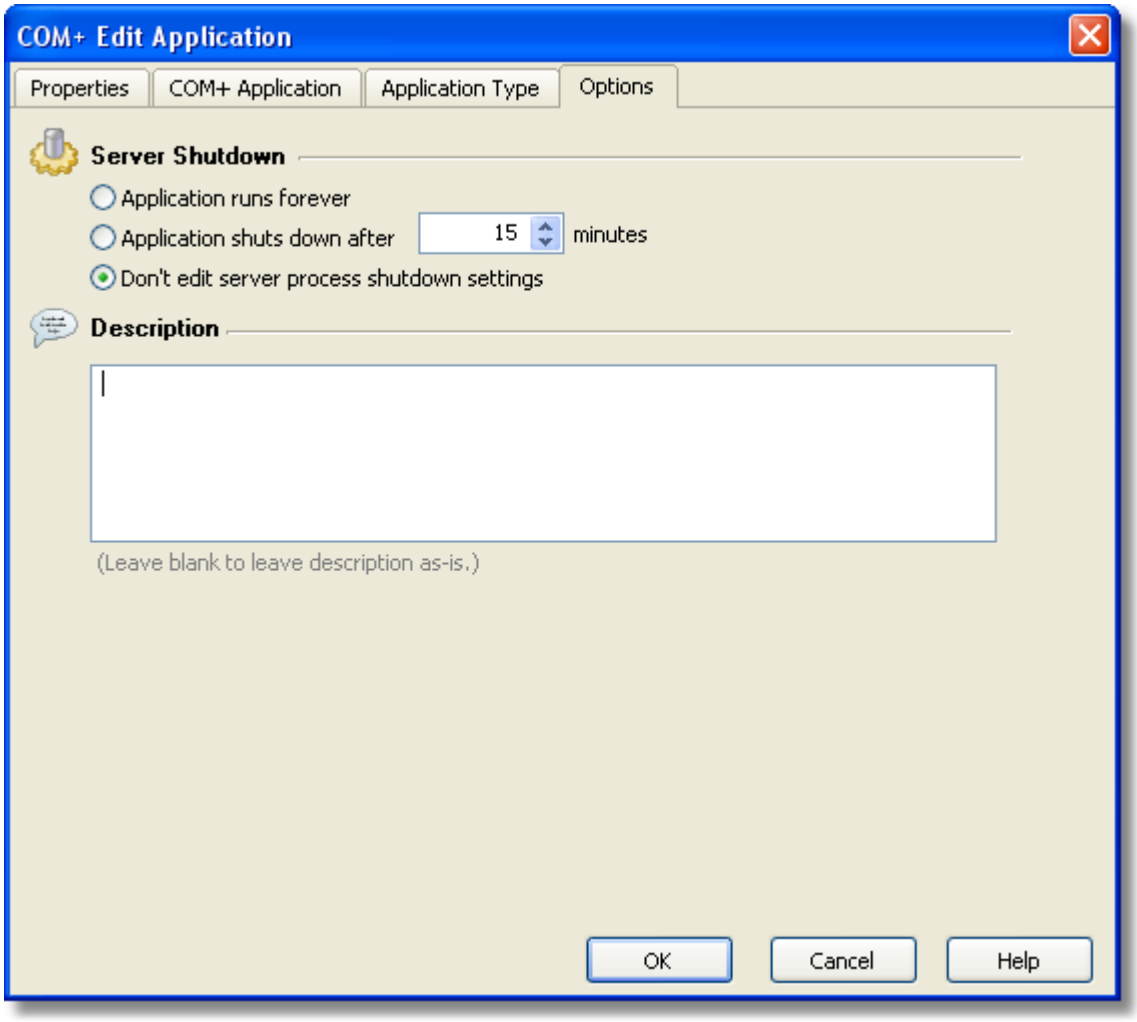
You can choose to set the Application to enable access checks at the process level, or

© 2001-2006 VSoft Technologies Pty Ltd

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

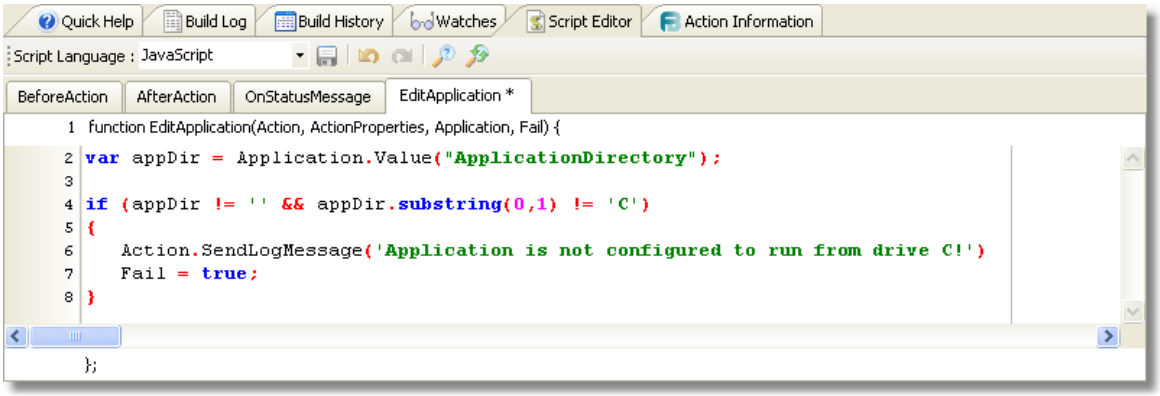
Enter a description to replace the existing description of the application. If left blank, the description will not be updated.

© 2001-2006 VSoft Technologies Pty Ltd

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:



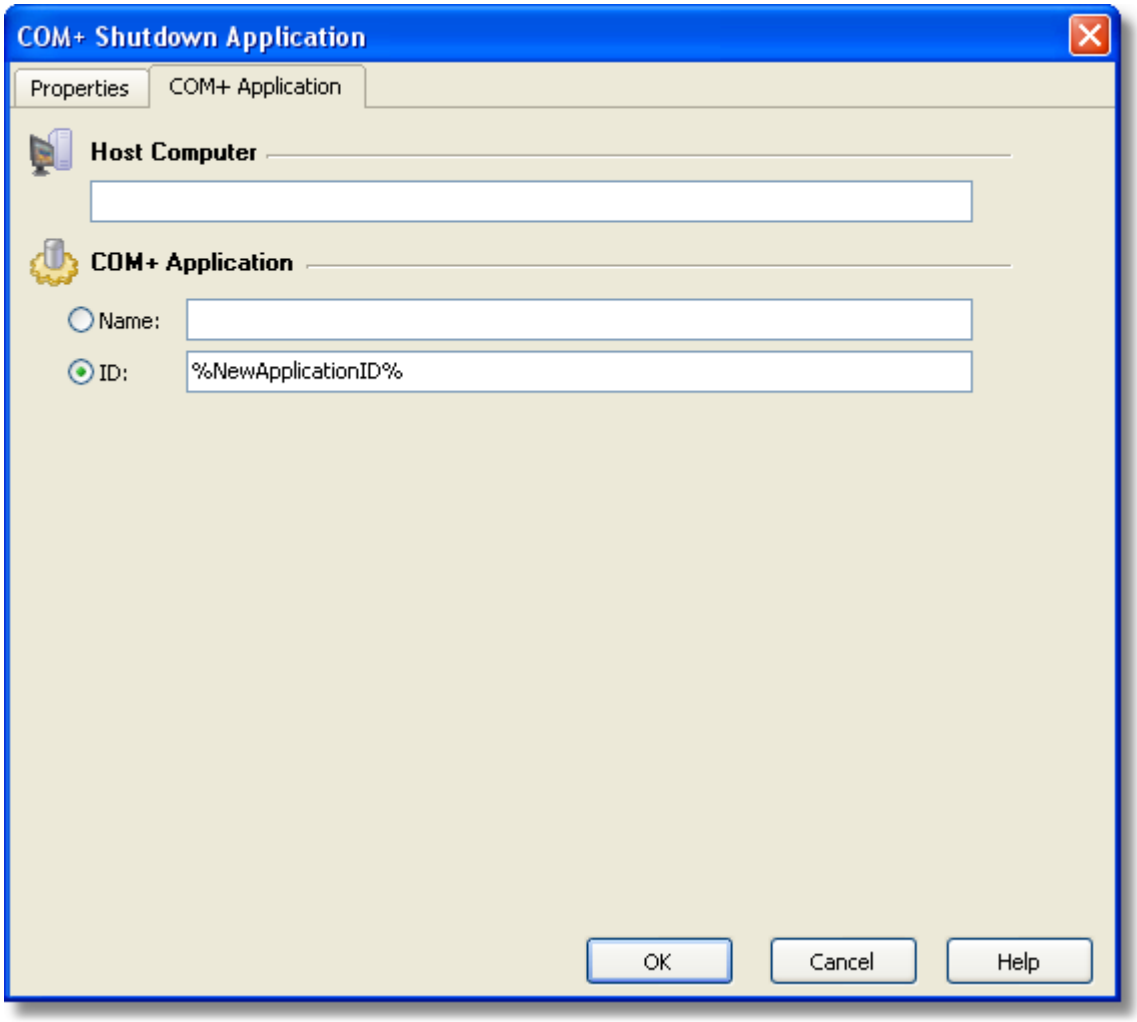
To read and write Application properties, use the syntax Application.Value("PropertyName"). Set the Fail parameter to true to force the action to fail programmatically.

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.

5.11.20.4 COM+ Shutdown Application

The COM+ Shutdown Application action allows you to shut down a running COM+ Application.

© 2001-2006 VSoft Technologies Pty Ltd



Host Computer

The computer you want to shut down the application on. You will need Administrator privileges on this computer. Leave blank for localhost.

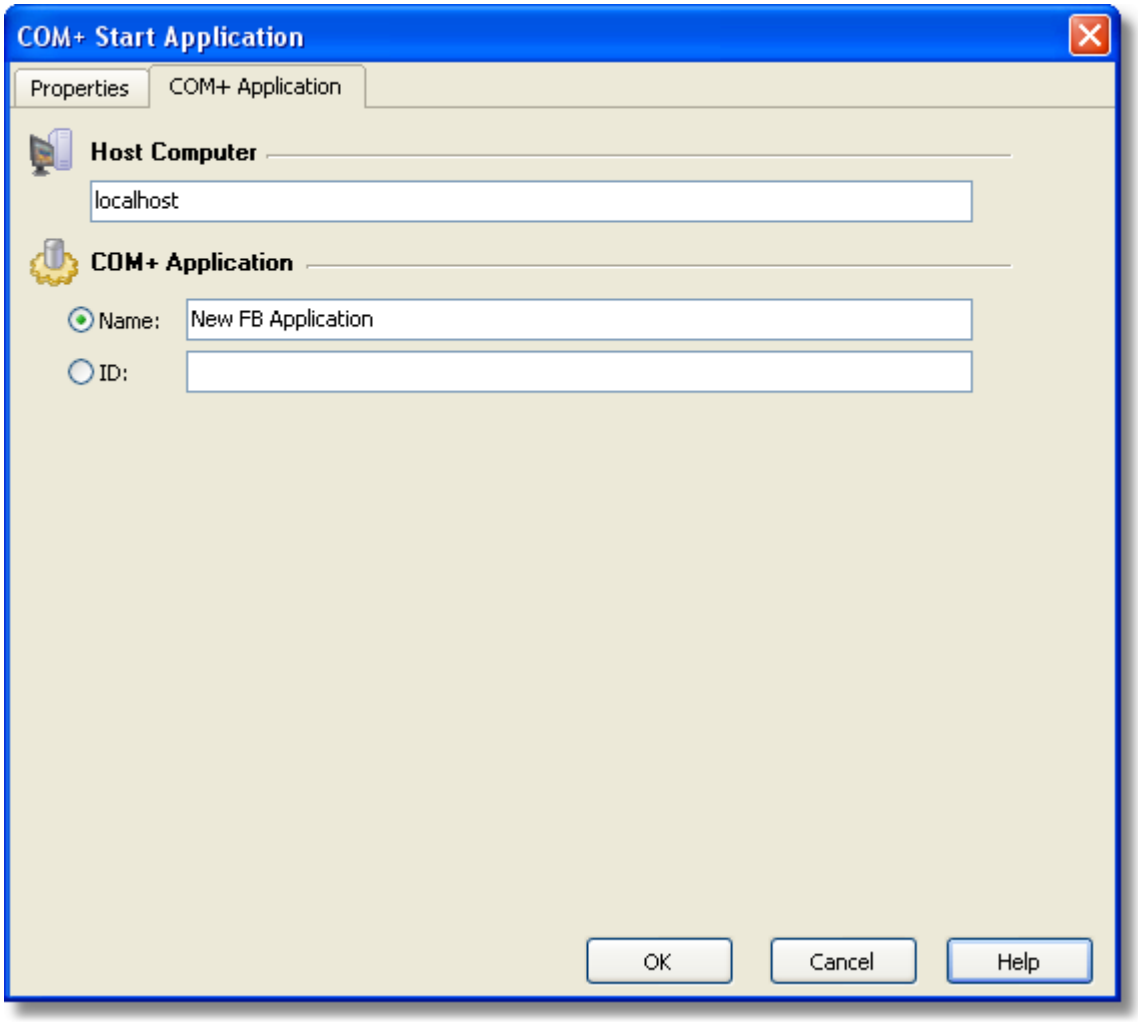
COM+ Application

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.

5.11.20.5 COM+ Start Application

The COM+ Start Application action allows you to start a shut down or inactive COM+ Application.

© 2001-2006 VSoft Technologies Pty Ltd



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.

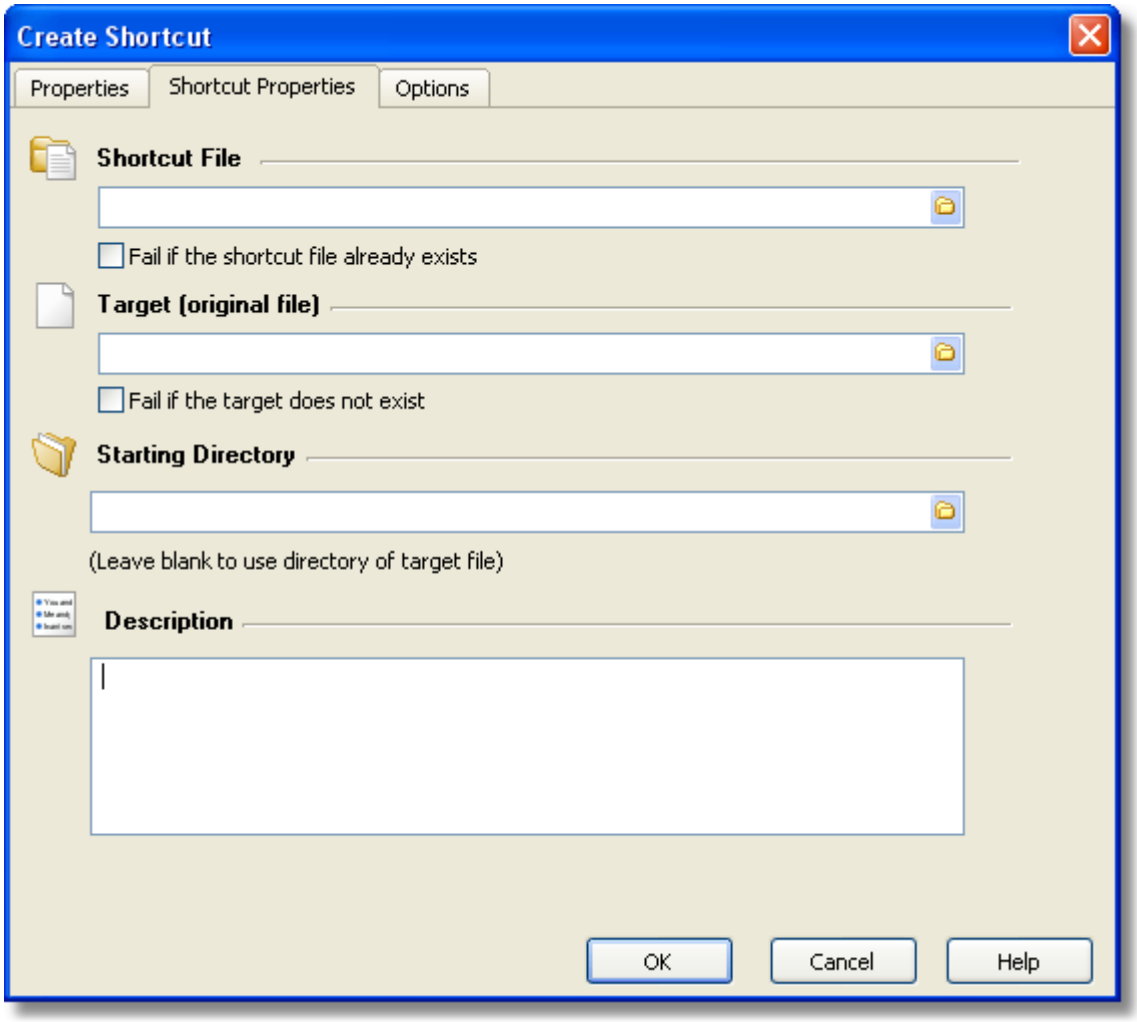
COM+ Application

You can specify the application name by its name or its ID (GUID.)

© 2001-2006 VSoft Technologies Pty Ltd

5.11.21 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

© 2001-2006 VSoft Technologies Pty Ltd

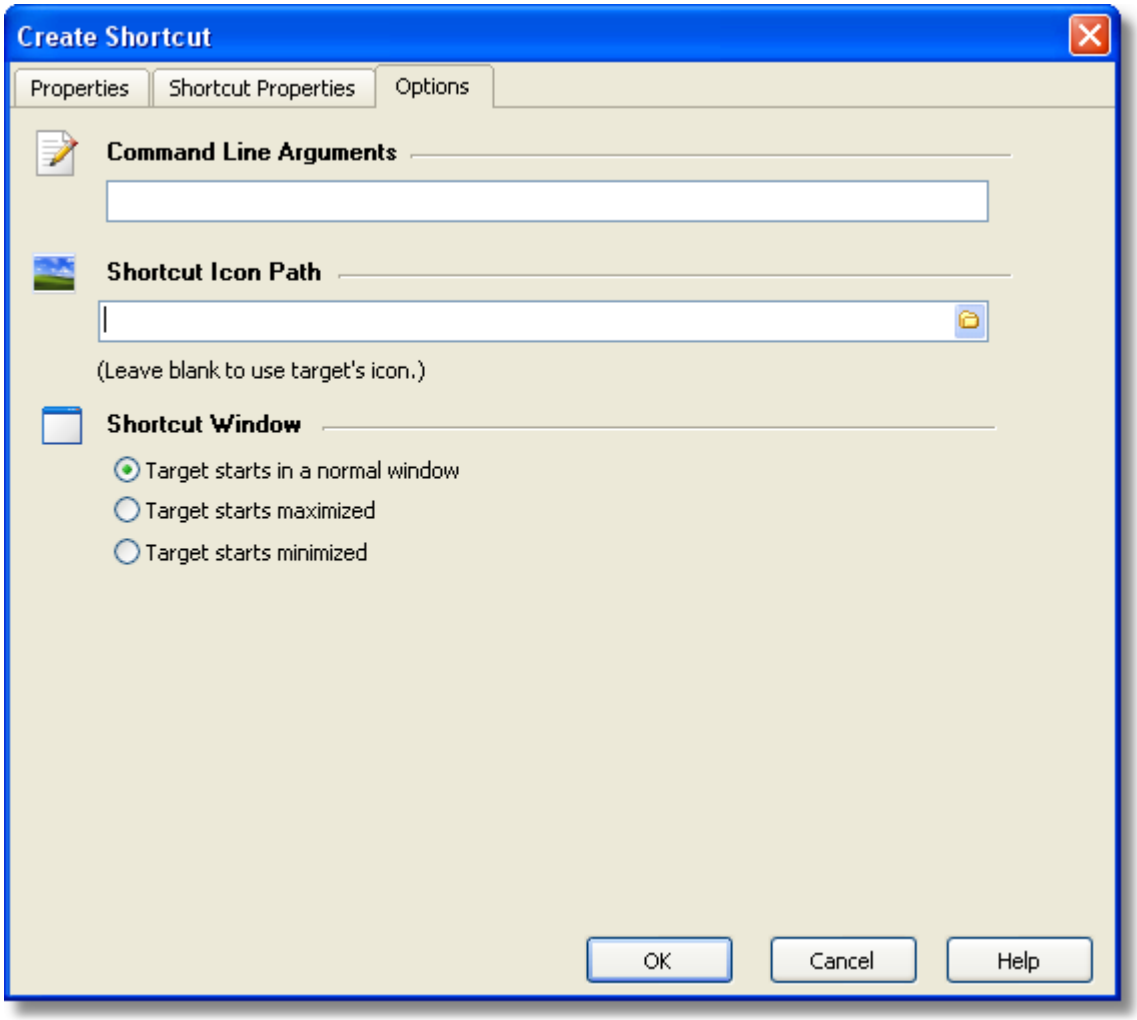
existing file. If this option is not set then the shortcut will be created, regardless.

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.

© 2001-2006 VSoft Technologies Pty Ltd

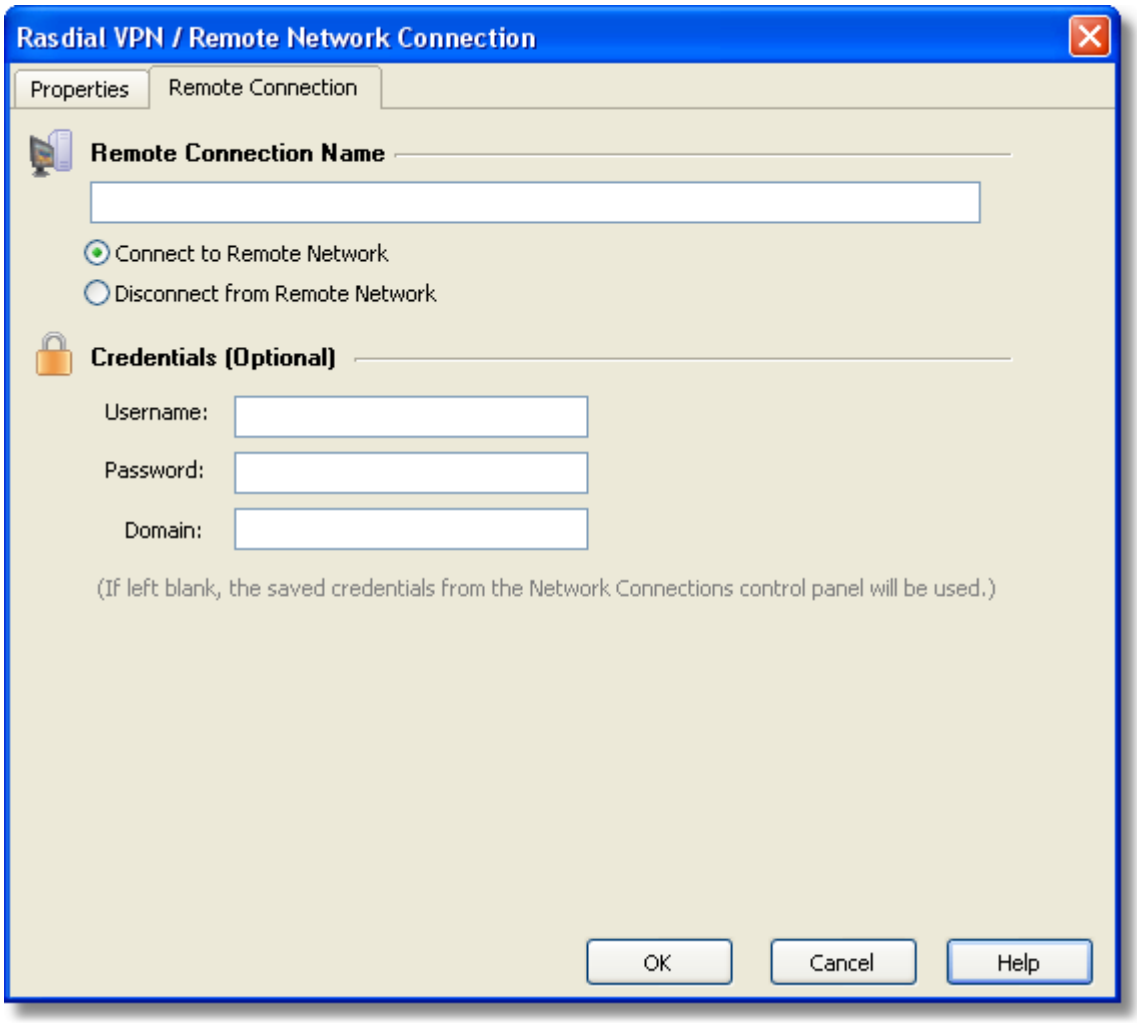
Shortcut Window

This option determines whether the target starts normally, in a maximized window, or in a minimized window in the background.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.22 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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

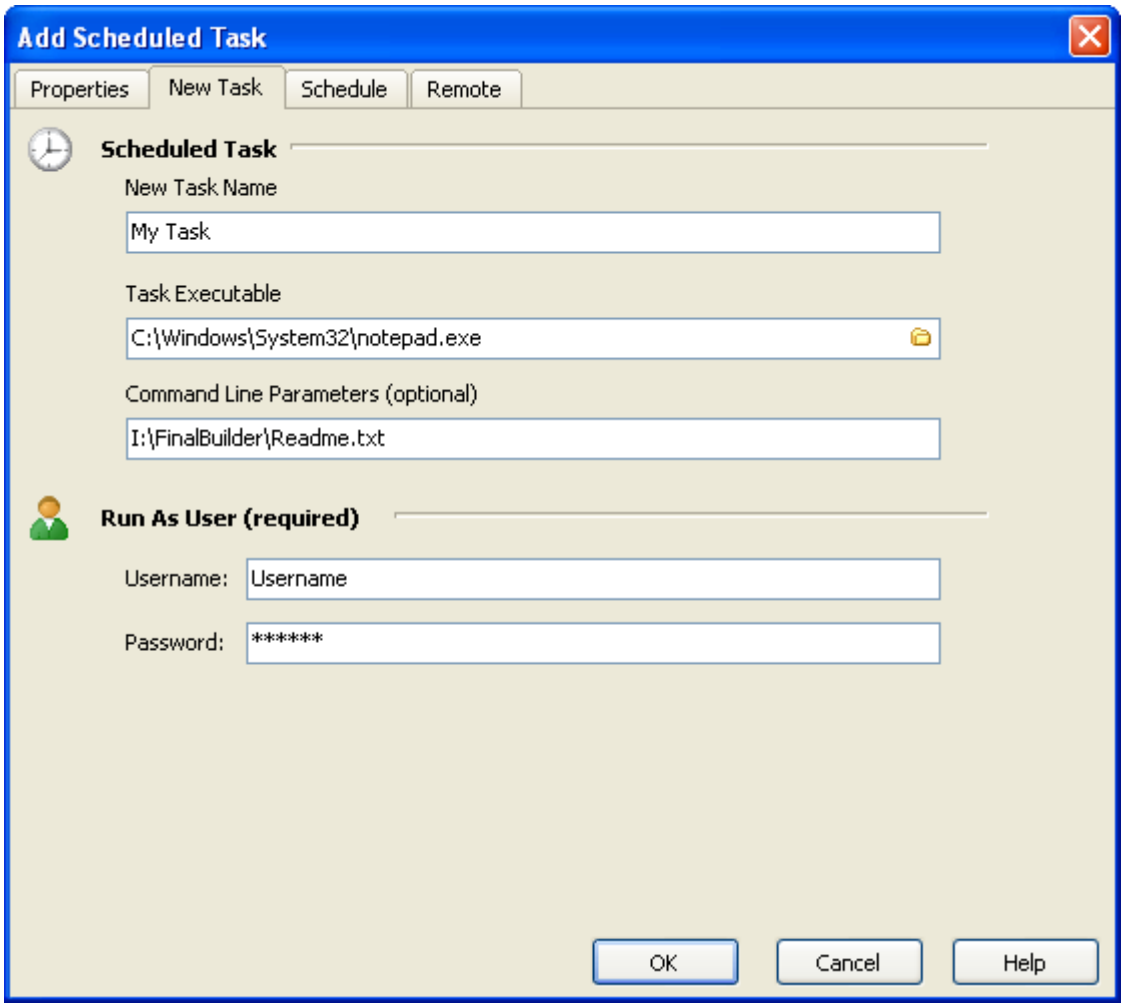
© 2001-2006 VSoft Technologies Pty Ltd

5.11.23 Windows Scheduler

5.11.23.1 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.

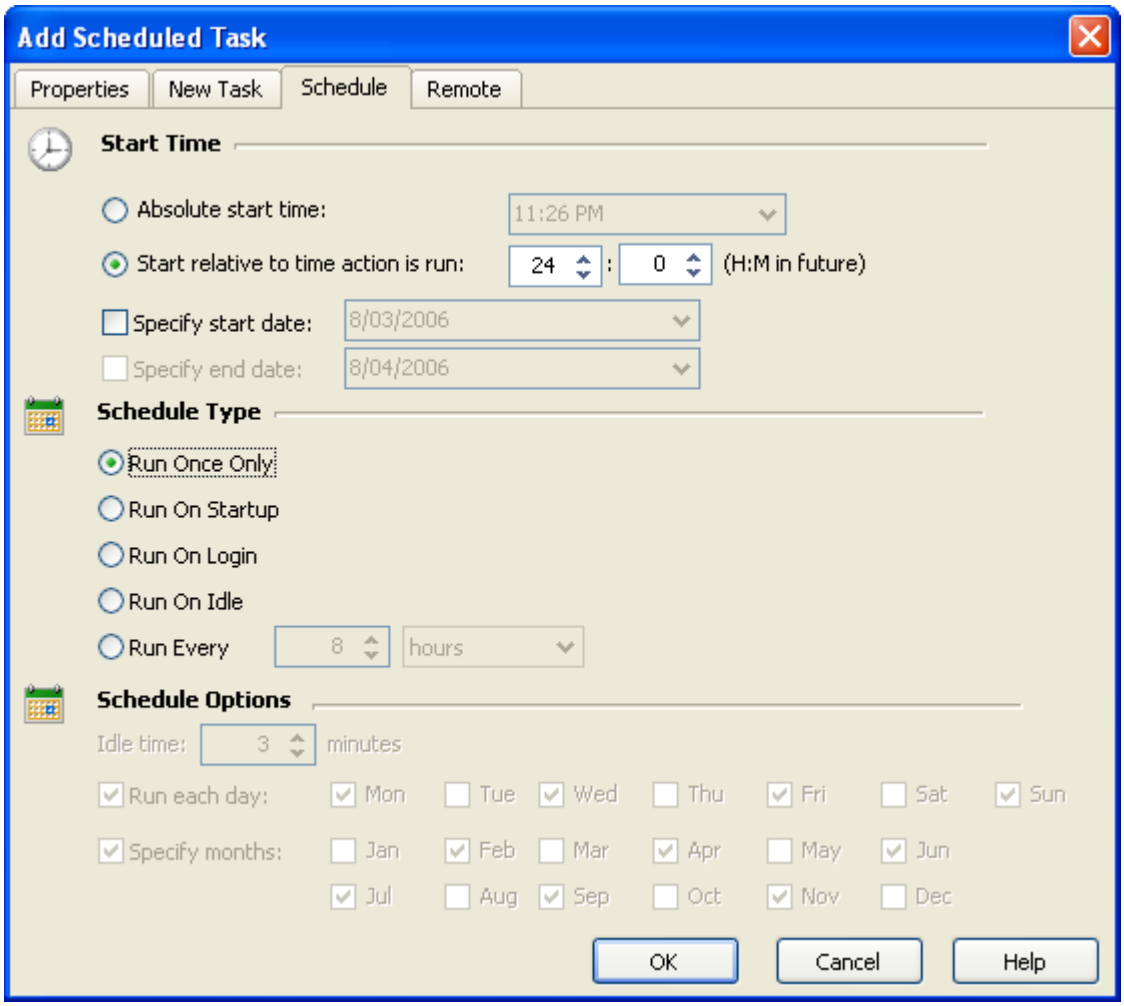
© 2001-2006 VSoft Technologies Pty Ltd

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.



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.)

© 2001-2006 VSoft Technologies Pty Ltd

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.

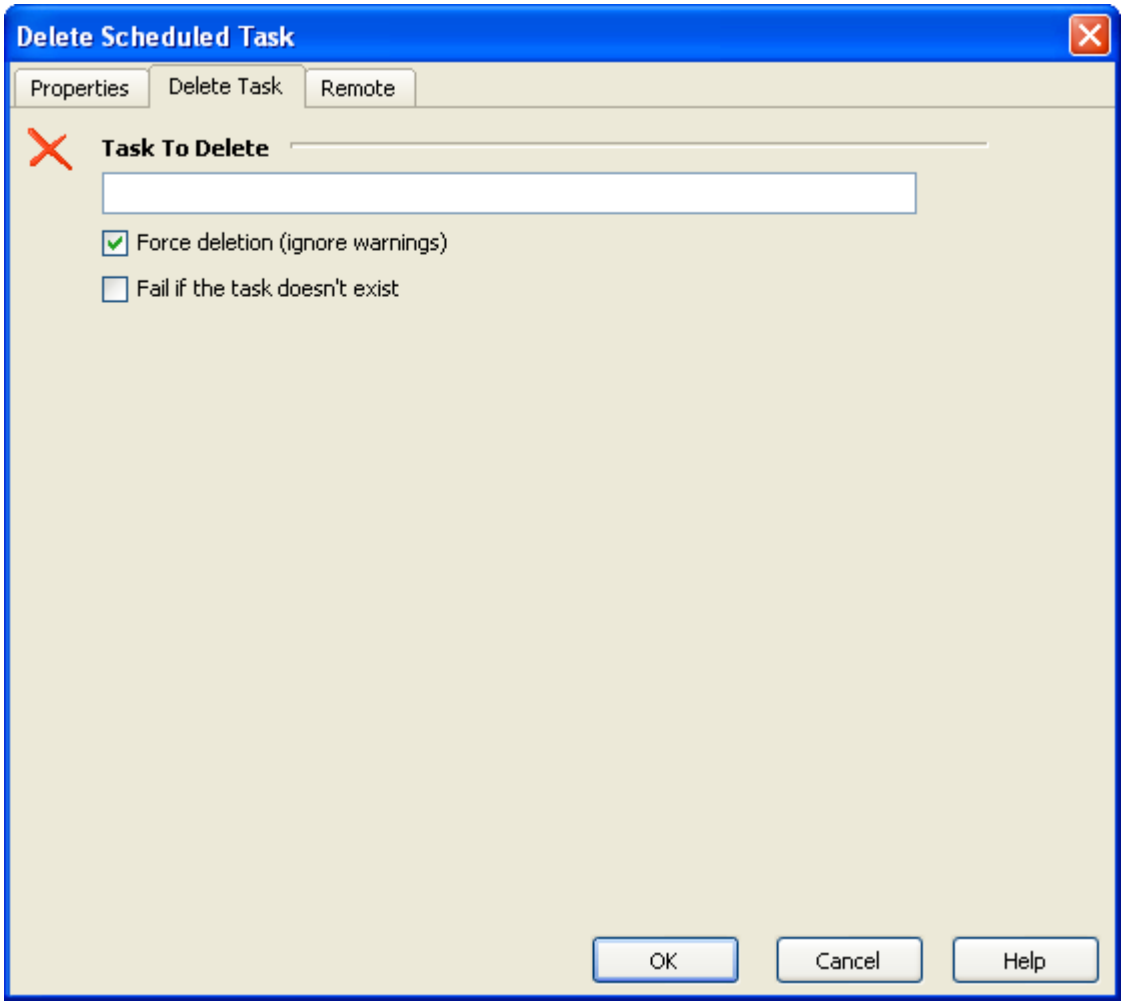
Remote Tab

© 2001-2006 VSoft Technologies Pty Ltd

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

5.11.23.2 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.

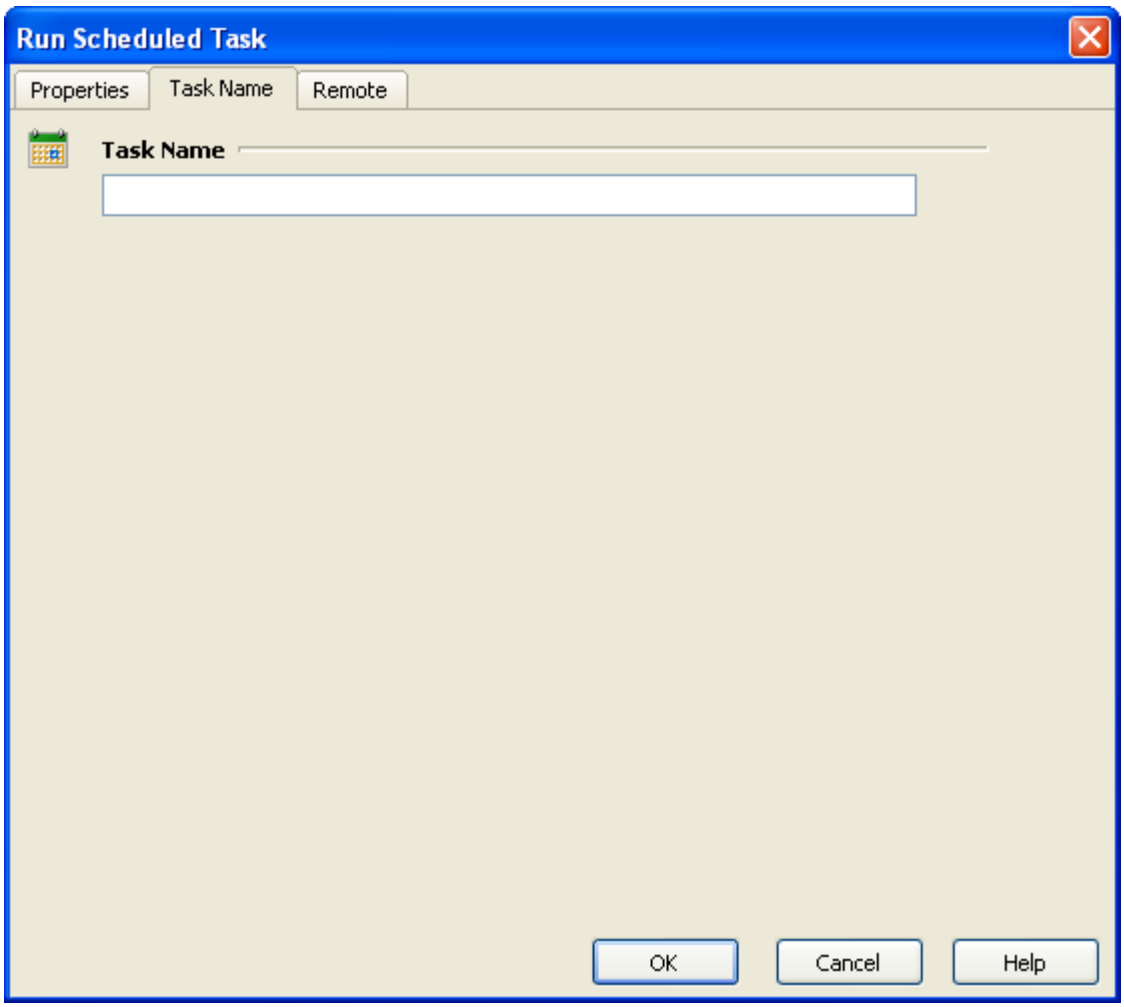
© 2001-2006 VSoft Technologies Pty Ltd

Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

5.11.23.3 Run Scheduled Task

The Run Scheduled Task action explicitly launches a scheduled task immediately.



Task Name

Enter the name of a task to run.

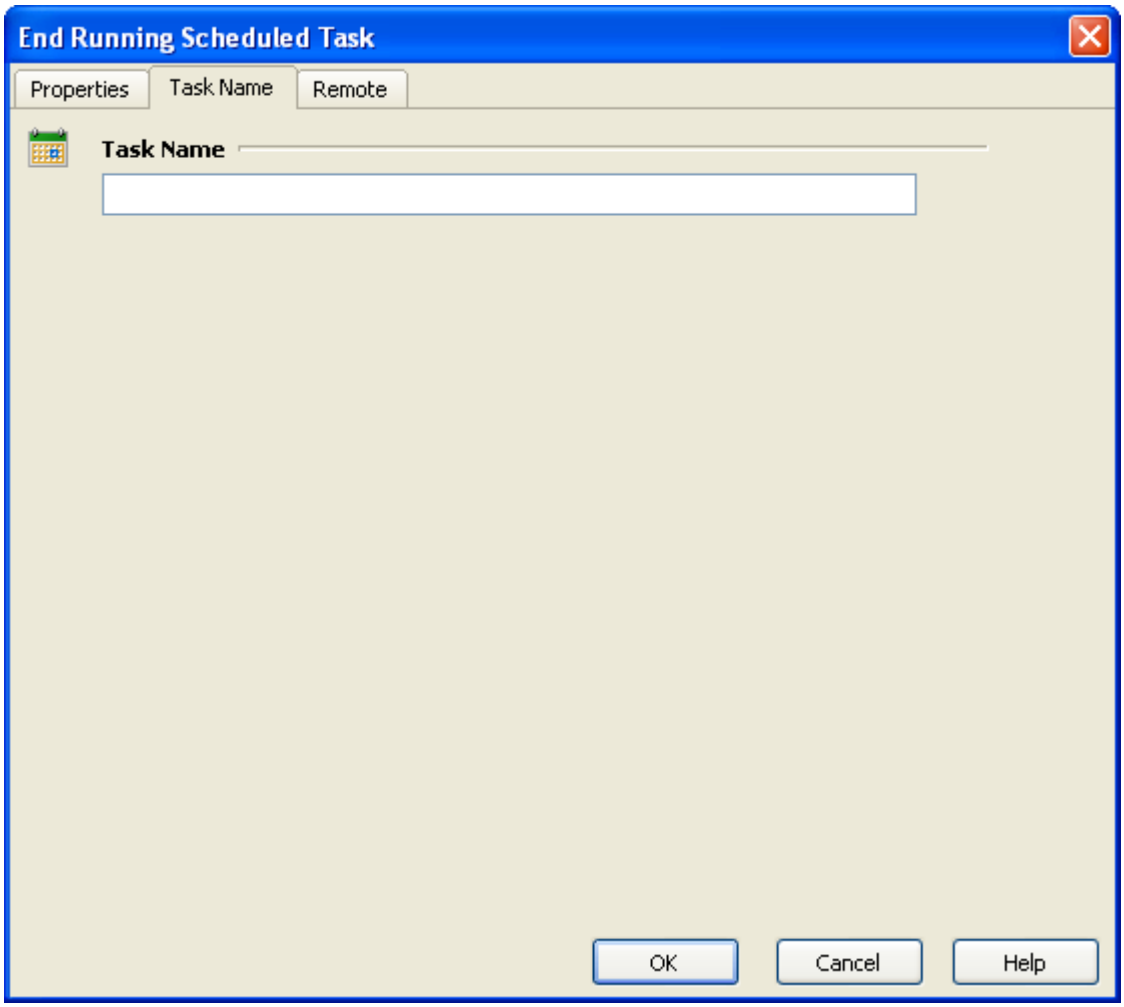
Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

© 2001-2006 VSoft Technologies Pty Ltd

5.11.23.4 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.

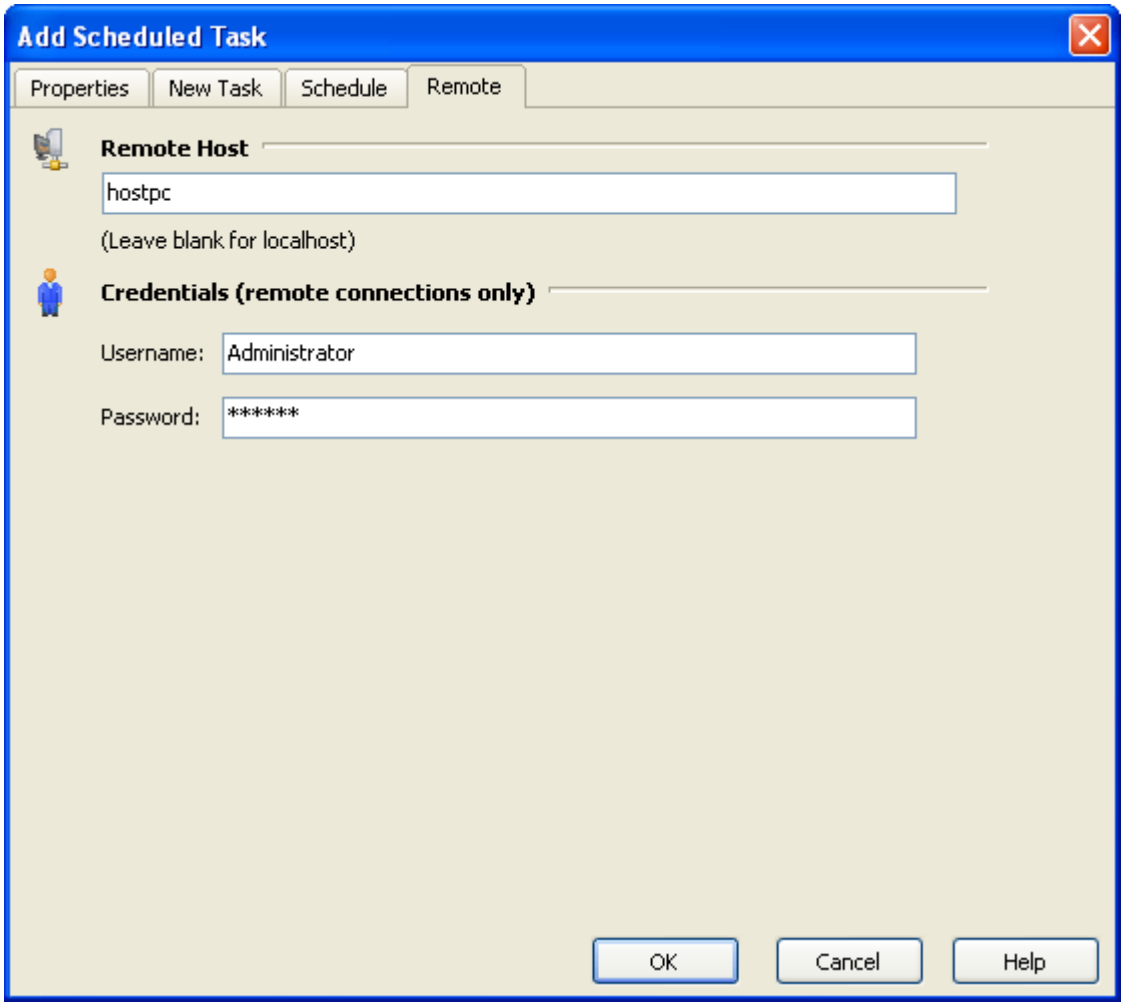
Remote Tab

See the Scheduling Remote Tasks topic for details on working with the Windows scheduler on remote computers.

5.11.23.5 Scheduling Remote Tasks

All of the Windows Scheduler actions contain a Remote tab, which allows you to schedule tasks on remote machines:

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

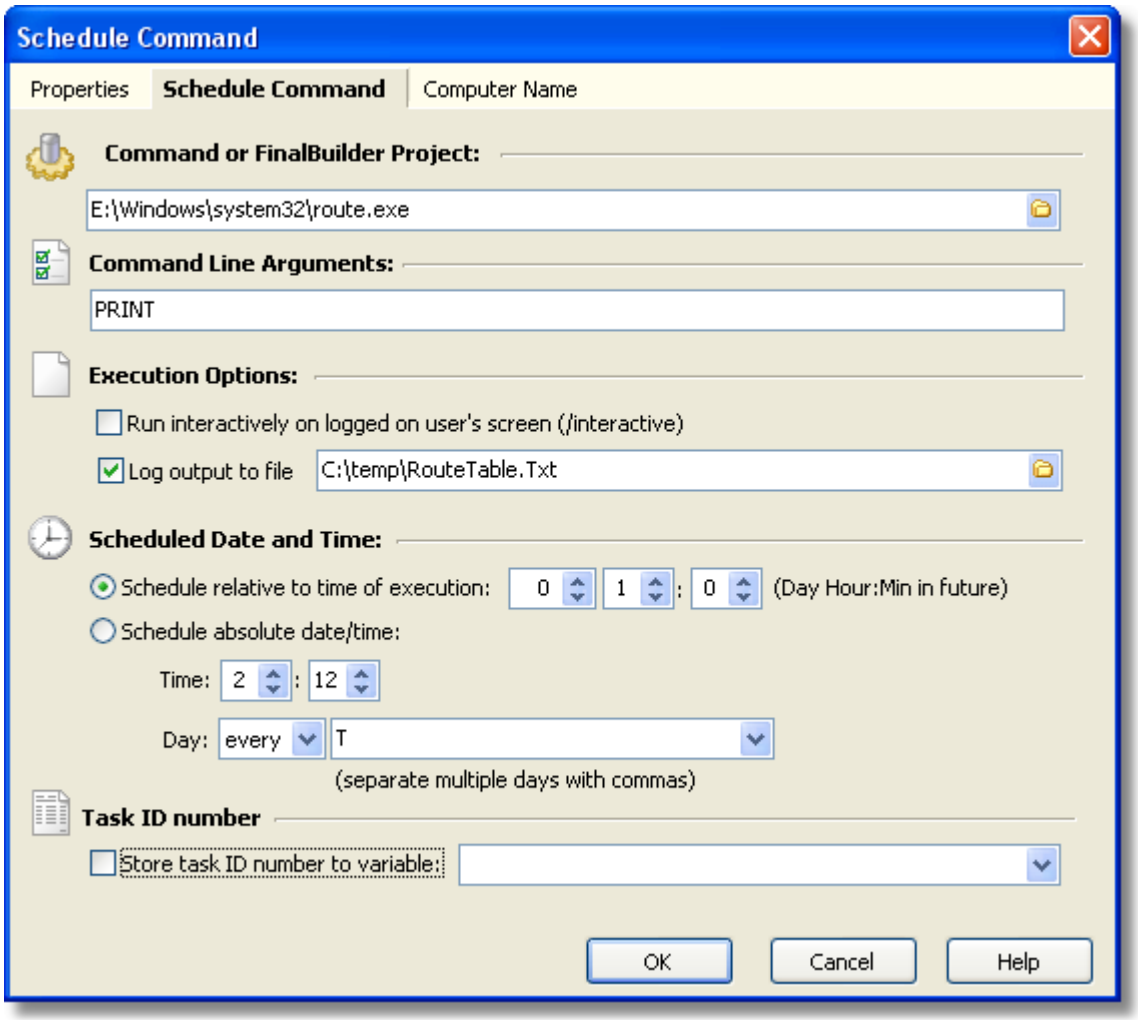
5.11.24 "AT" Scheduler

5.11.24.1 Schedule Add

Note: FinalBuilder supports both the DOS 'AT' Scheduler and the Windows Task Scheduler. It is recommended you use the Windows Scheduler actions in most cases.

The Schedule Add Action acts as a front-end to the DOS 'AT' command, and allows you to schedule a program for execution at a later time.

You can schedule an executable command or a FinalBuilder project file to be executed.



Command or FinalBuilder Project

Specify the name of an executable, or a FinalBuilder project, you would like to execute.

Command Line Arguments

Specify the name of any command line arguments to pass to the executable.

© 2001-2006 VSoft Technologies Pty Ltd

Execution Options

Run interactively on logged on user's screen...

If selected, then the program will display on the screen of the logged on user at time of execution. Otherwise, it will execute in the background.

Note: If running a FinalBuilder project file, "Interactive" running means the project will be loaded in FinalBuilder, "non-interactive" means it will be loaded "behind the scenes" with FBCMD.

Log output to file

If checked, the "standard output" of the executable will be sent to a file.

Note: This option is not available when using a FinalBuilder project file in Interactive mode.

Scheduled Date & Time

The execution time can either be scheduled statically, or relative to the time at which the action is executed.

Schedule relative to time of execution...

Specify the duration after which the program will be loaded. In the above example, route.exe will be scheduled to execute one hour after the action is run.

Schedule absolute date / time

Specify a time and (optionally) a date for execution. Execution can be "every" day or only on the "next" day which matches the parameters.

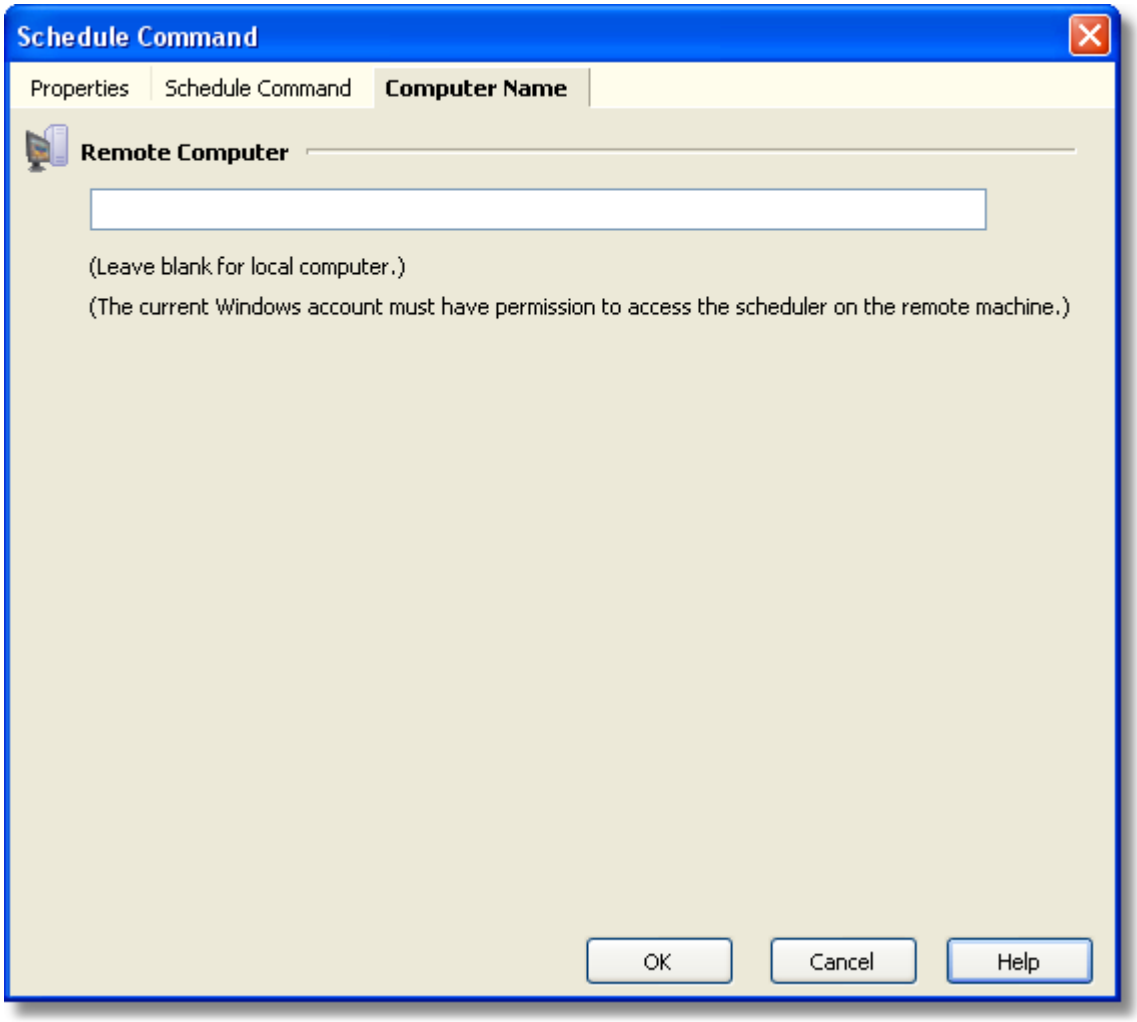
The days specified can be a combination of either the letters M,T,W,Th,F,S,Su (Monday-Friday), or the numbers 1 through 31.

Task ID Number

Store Task ID Number to Variable

If checked, the task ID of the new task will be stored to a variable. It can be used in other actions, for example Deschedule Command.

© 2001-2006 VSoft Technologies Pty Ltd



Remote Computer

The command can be scheduled on a remote computer instead of the local one. Note that the current Windows account must have adequate permissions on the remote machine (a Windows NT Domain may be necessary in order to achieve this.)

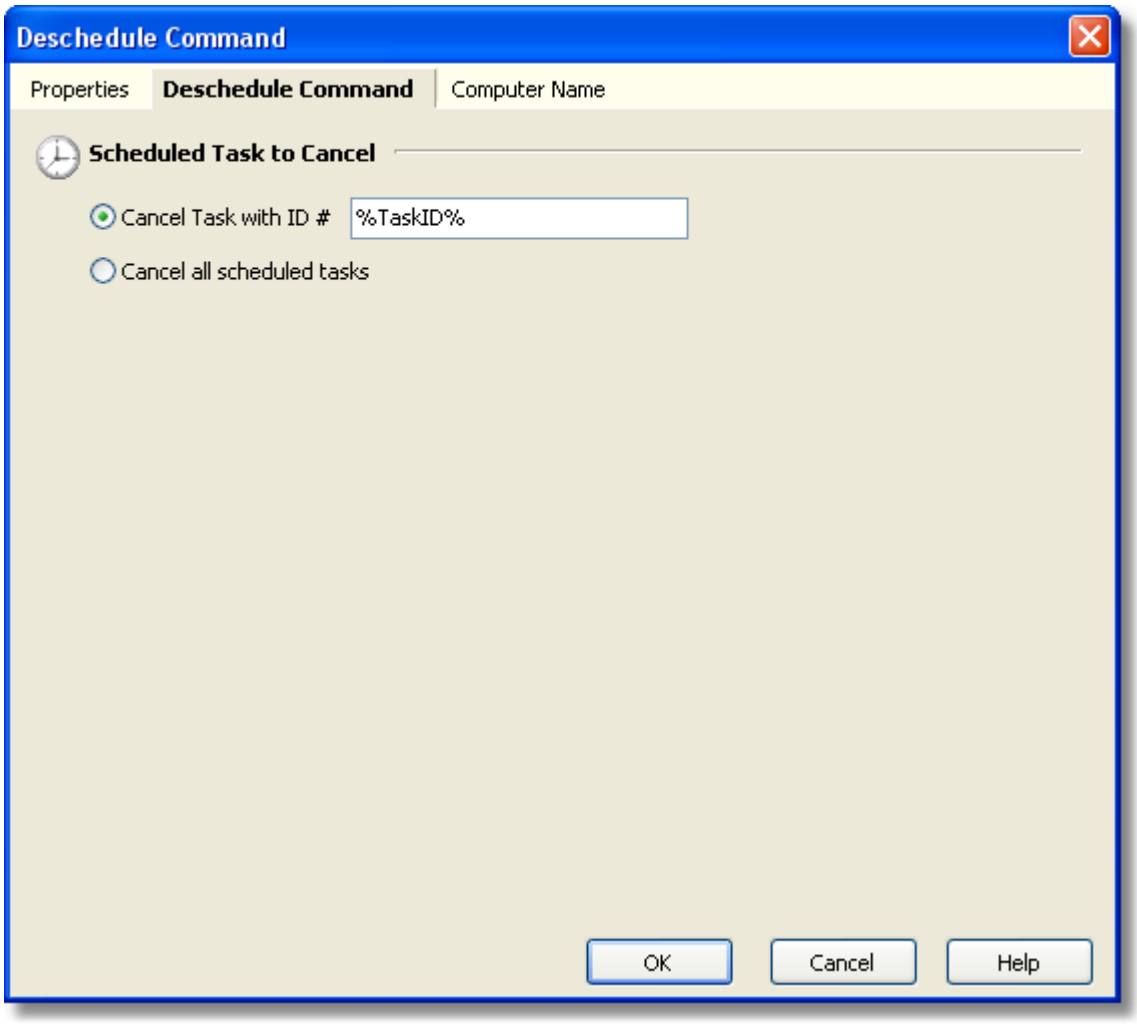
Note: Automatically executing FinalBuilder project files on a remote machine will only succeed if the FinalBuilder installation directory path & FinalBuilder project file paths are the same on both computers. If not, you will need to specify the execution path manually and pass in the FinalBuilder project file as a command line argument.

5.11.24.2 Schedule Delete

Note: FinalBuilder supports both the DOS 'AT' Scheduler and the Windows Task Scheduler. It is recommended you use the Windows Scheduler actions in most cases.

The Schedule Delete Action acts as a front-end to the DOS 'AT' command, and allows you to remove commands which have already been scheduled. Tasks can be de-scheduled individually (by passing a numeric task ID), or all pending tasks can be de-scheduled at once.

© 2001-2006 VSoft Technologies Pty Ltd



Cancel Task with ID#

The specified task will be cancelled. Note that in the example, a FinalBuilder variable is being used to specify the Task ID.

Cancel all Scheduled Tasks

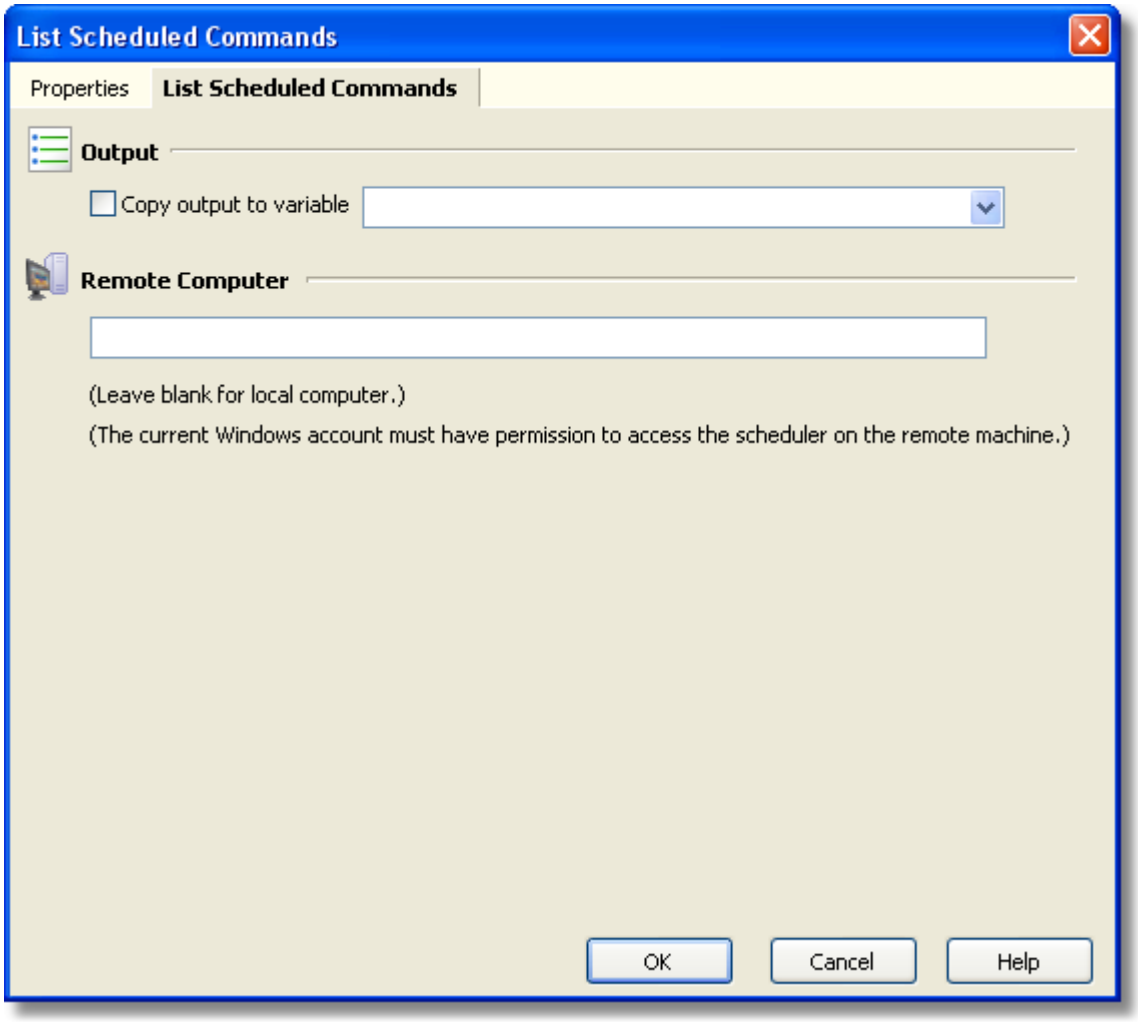
All tasks scheduled for execution will be cancelled.

5.11.24.3 Schedule List

Note: FinalBuilder supports both the DOS 'AT' Scheduler and the Windows Task Scheduler. It is recommended you use the Windows Scheduler actions in most cases.

The Schedule List Action acts as a front-end to the DOS 'AT' command, and lists all processes scheduled by AT for execution on the local, or a remote, computer.

© 2001-2006 VSoft Technologies Pty Ltd



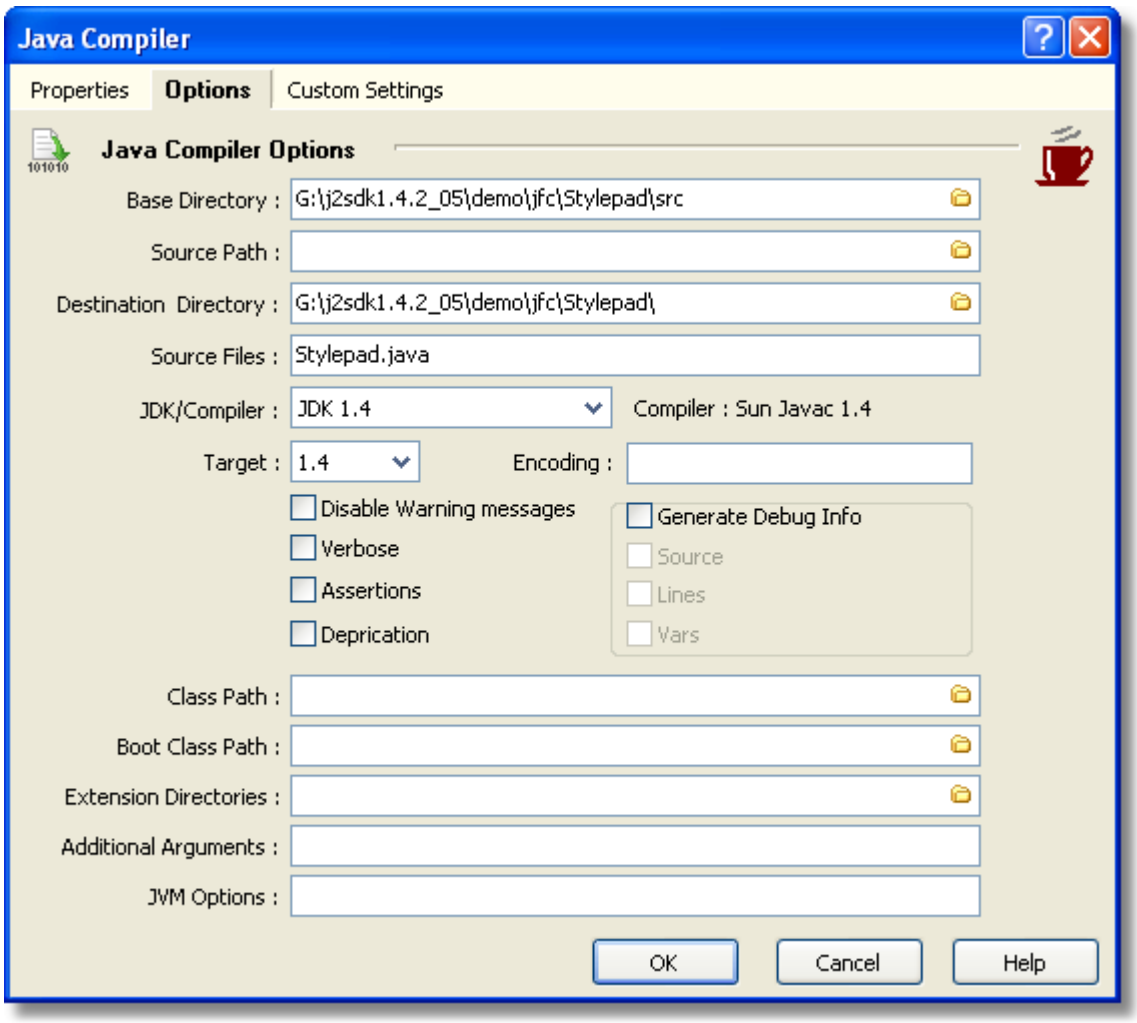
Copy output to variable

The list of tasks can be copied to a FinalBuilder variable (otherwise, it is only written to the log.)

© 2001-2006 VSoft Technologies Pty Ltd

5.12 Compilers

5.12.1 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 Borland 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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.12.1.1 Borland Compiler Options

The Borland Compiler does not have any extra settings.

5.12.1.2 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.

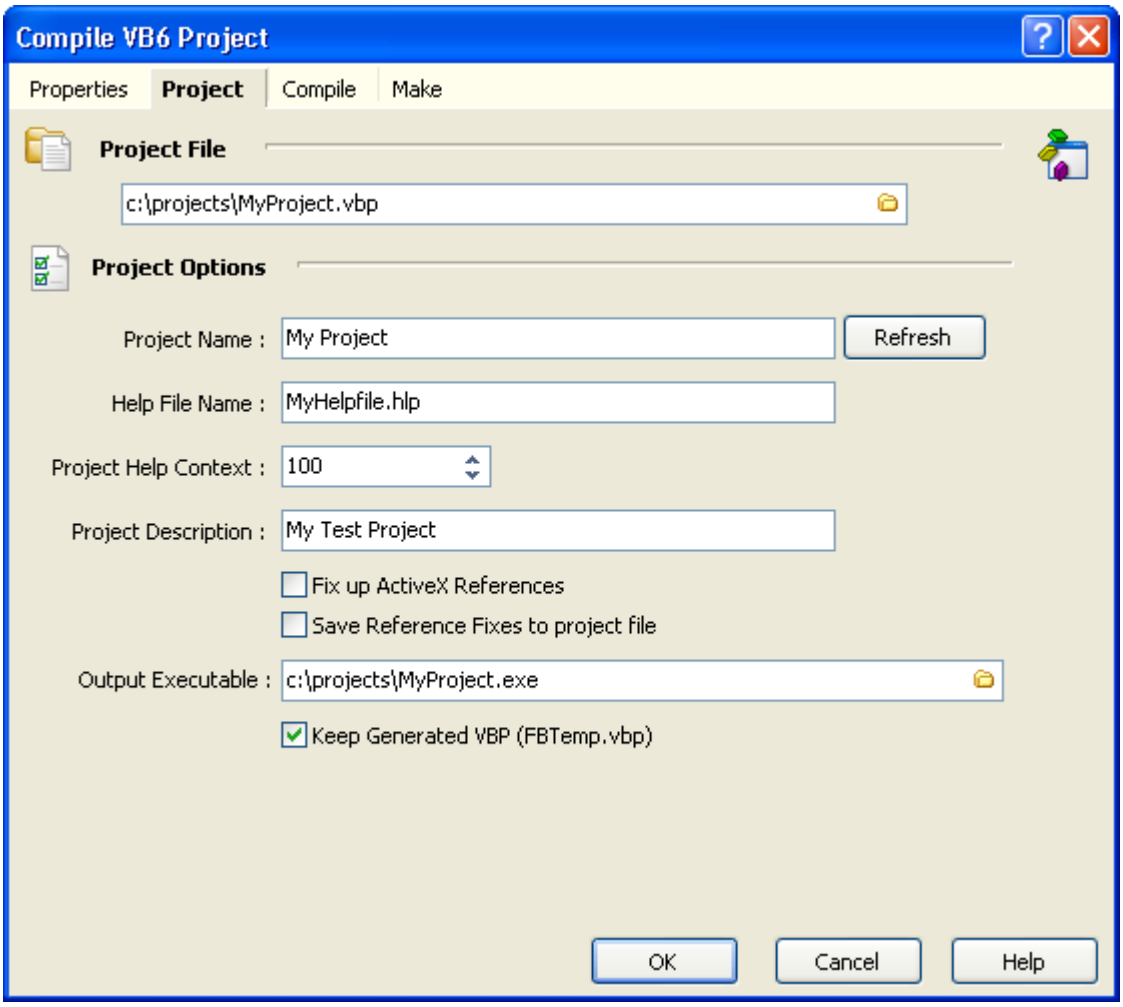
You should now see the jdk config you just created in the dropdown list on the Java Compiler action's property page. You can create as many configurations as you like. Note that these configs are stored in the registry, if you move your project to another machine you should first define those configurations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

© 2001-2006 VSoft Technologies Pty Ltd

5.12.2 Microsoft

5.12.2.1 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.

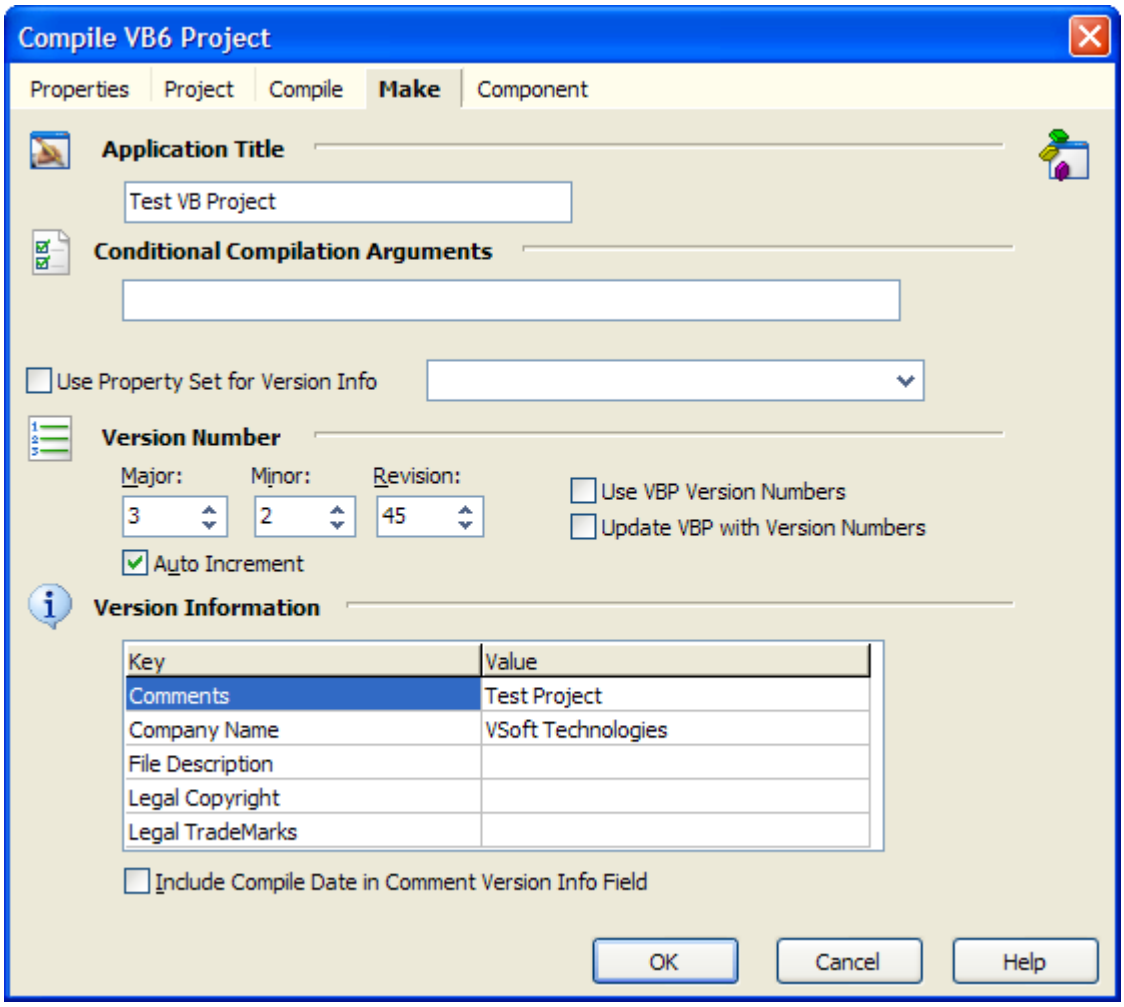
Help File Name : The name of the Help file associated with the project.

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.

© 2001-2006 VSoft Technologies Pty Ltd

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

© 2001-2006 VSoft Technologies Pty Ltd

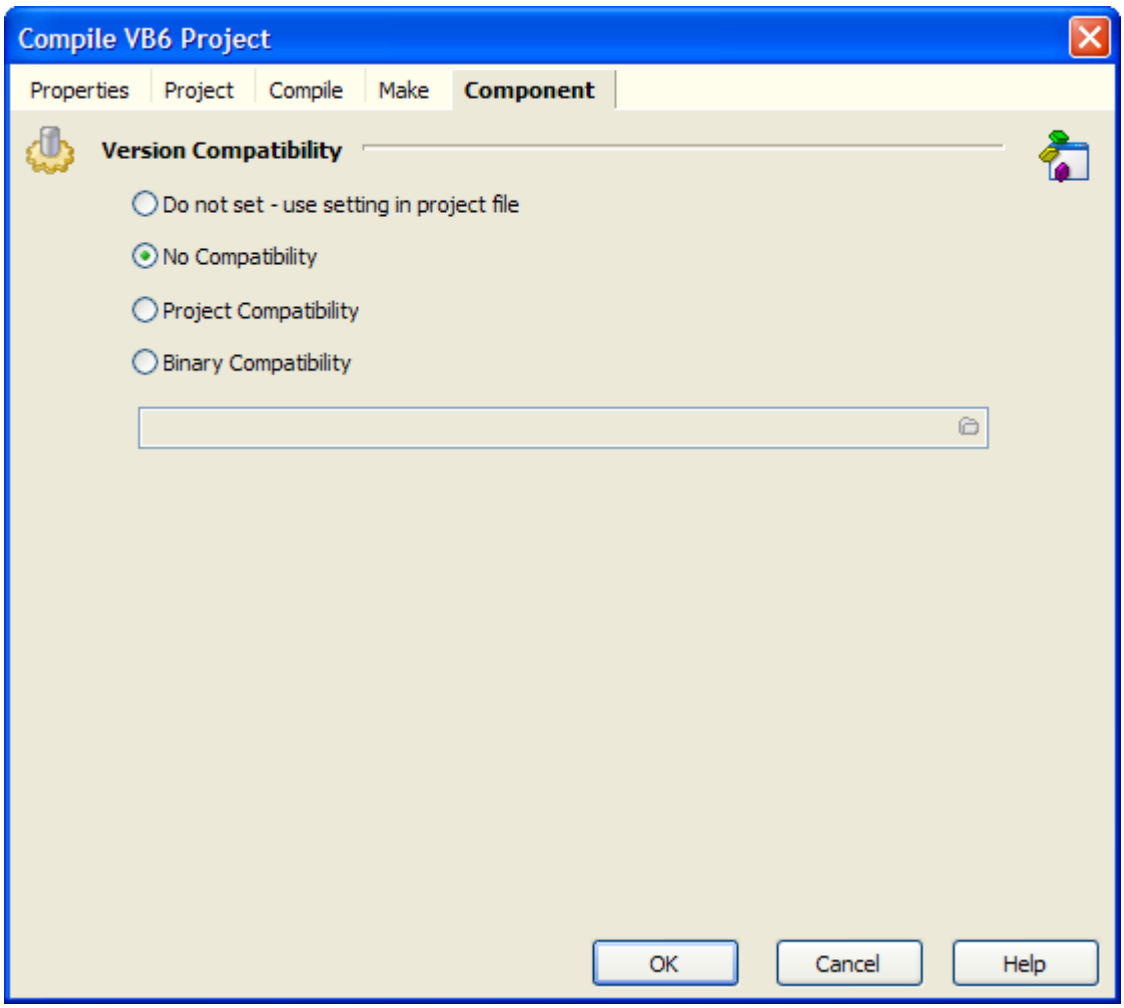
property ProjectName: WideString
property ProjectDescription: WideString
property ProjectHelpFile: WideString
property ProjectHelpContextID: integer

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

© 2001-2006 VSoft Technologies Pty Ltd

5.12.2.1.1 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

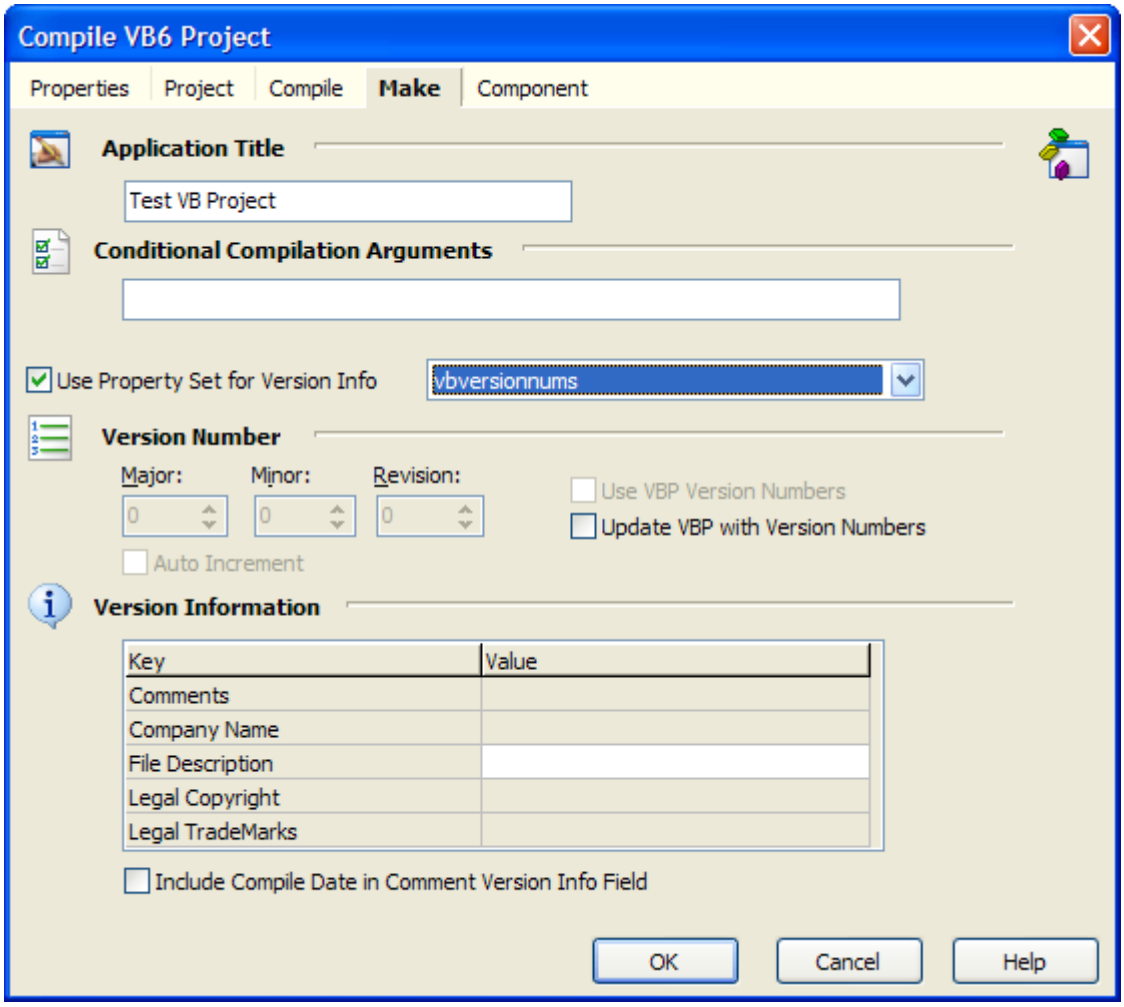
TIP: Use this setting following the release of a component to other developers.

5.12.2.1.2 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:

© 2001-2006 VSoft Technologies Pty Ltd

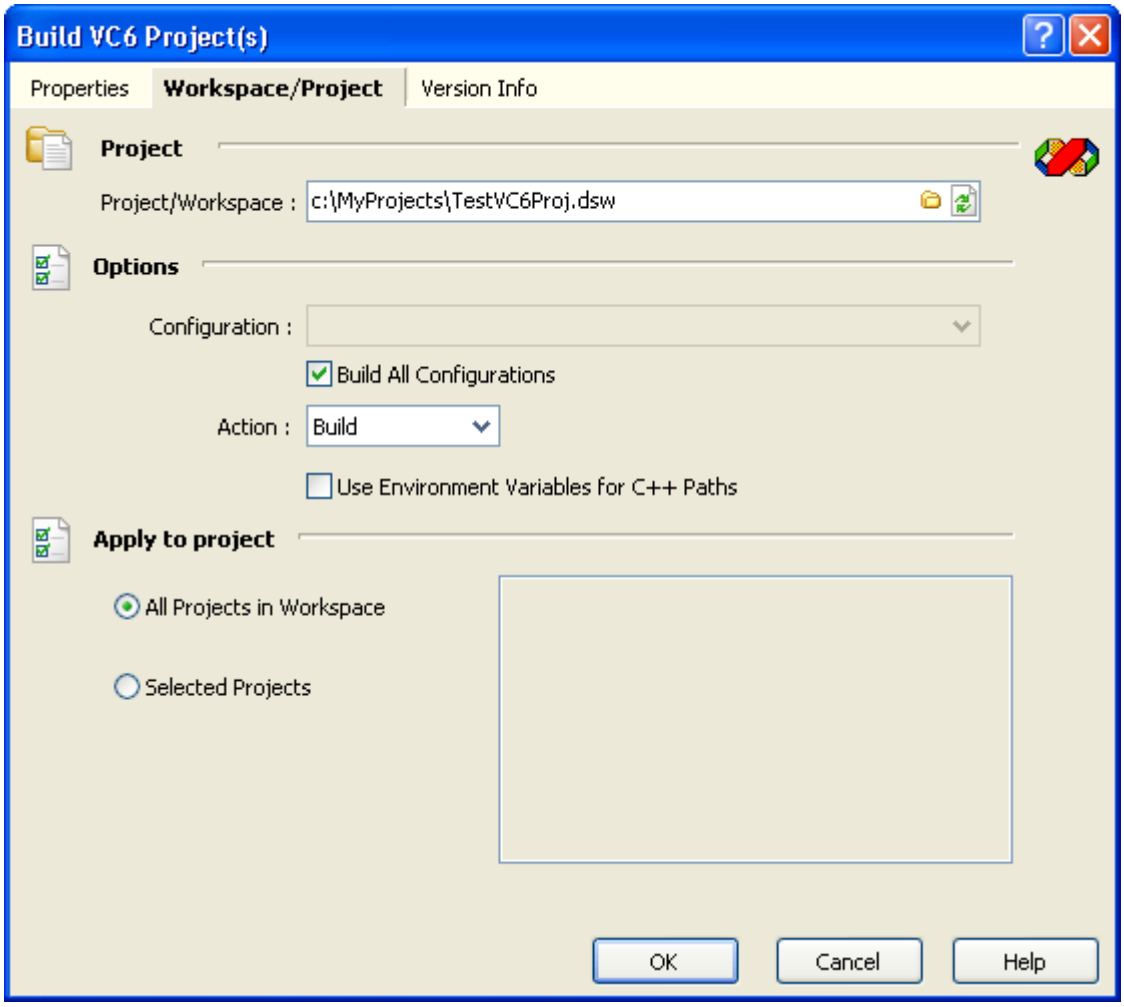


When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

5.12.2.2 Visual C++ 6 Action

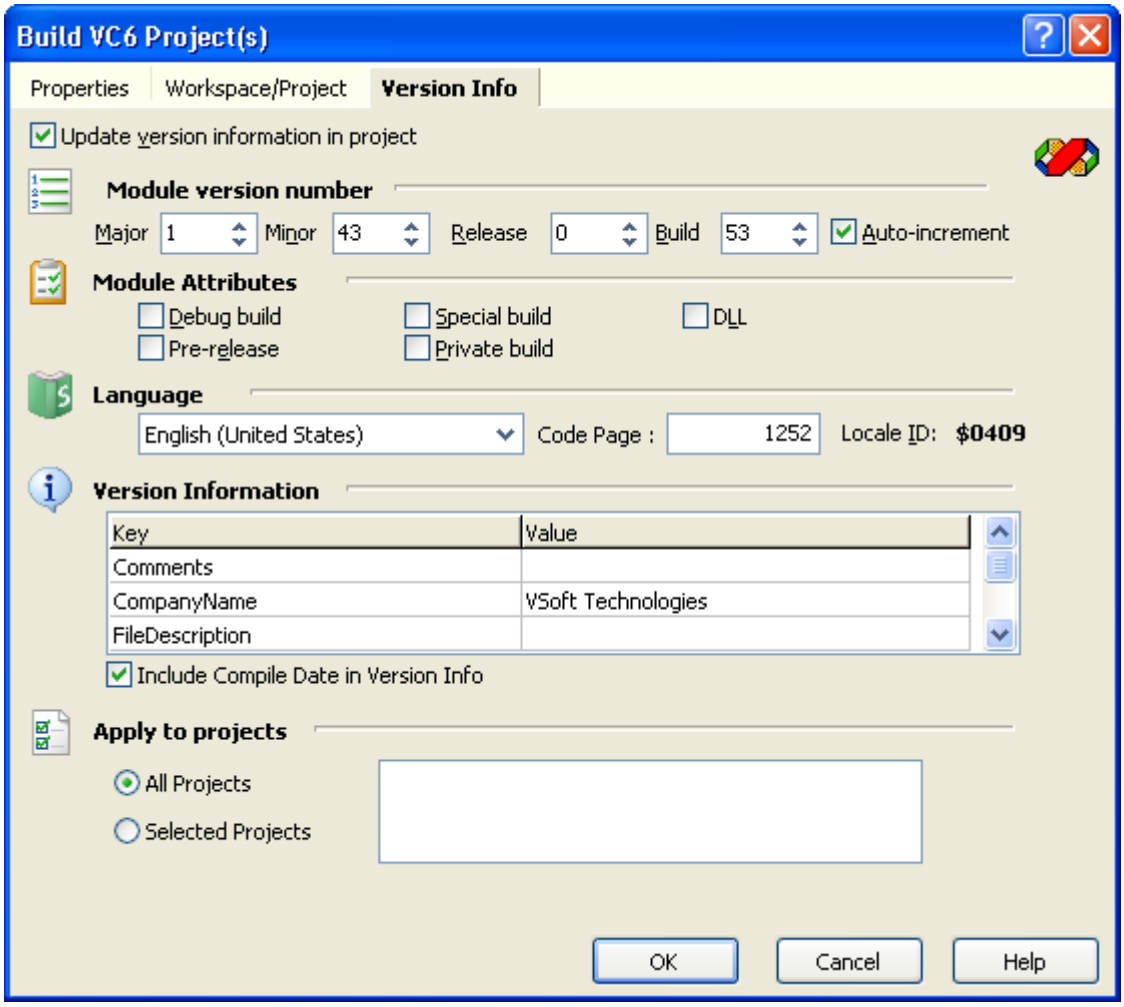
This action provides the ability to build Visual C++ 6 projects and Workspaces. You can choose to build/rebuild/clean all projects in a workspace or selected projects. You can also choose the Configuration to be used (this configuration must exist in all projects in the workspace).

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

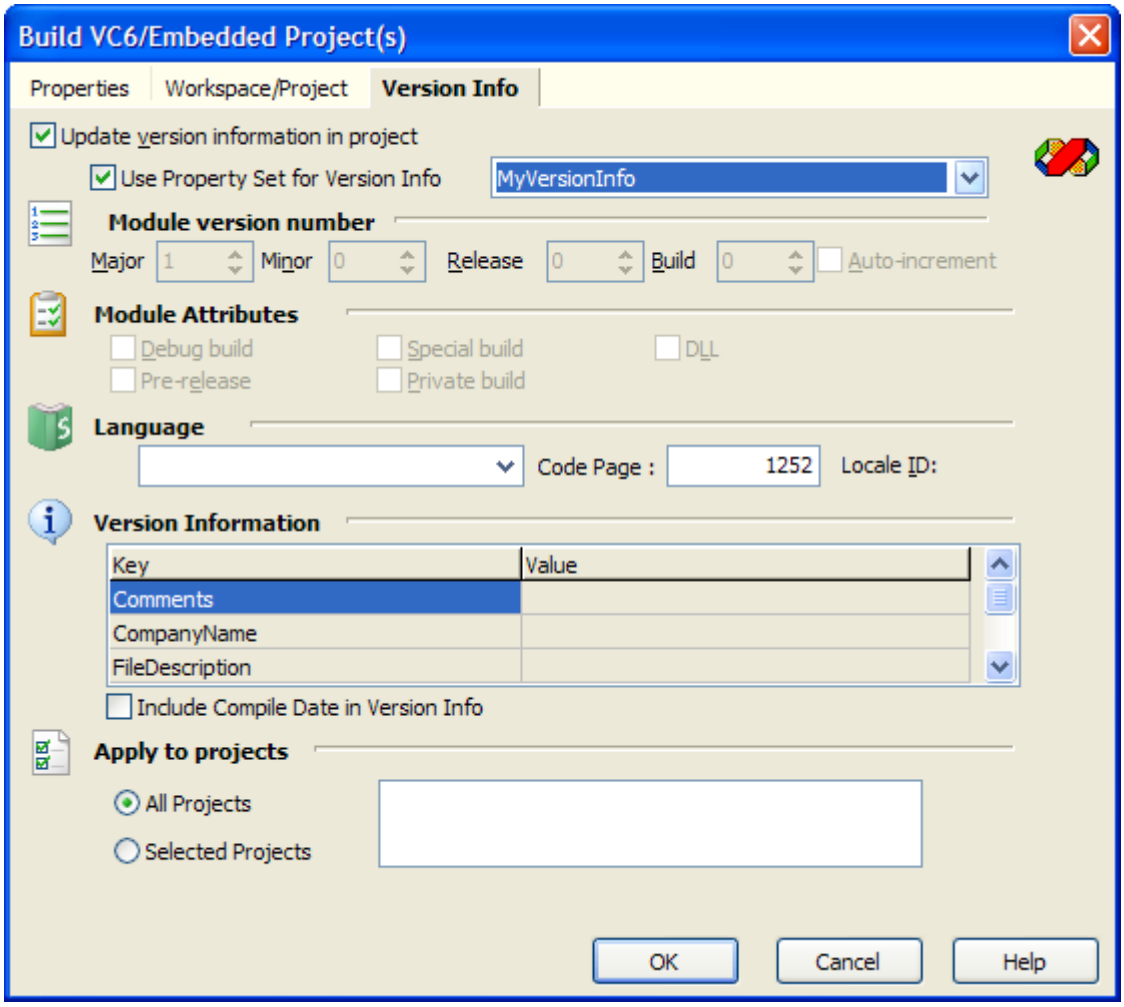
```
property InvokeMacro      : string;
property IncludeVerInfo   : boolean;
property AutoIncBuild     : boolean;
property MajorVersion     : integer;
property MinorVersion     : integer;
property ReleaseVersion   : integer;
property BuildVersion     : integer;
property IsDebug          : boolean;
property 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=
```

5.12.2.2.1 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:

© 2001-2006 VSoft Technologies Pty Ltd

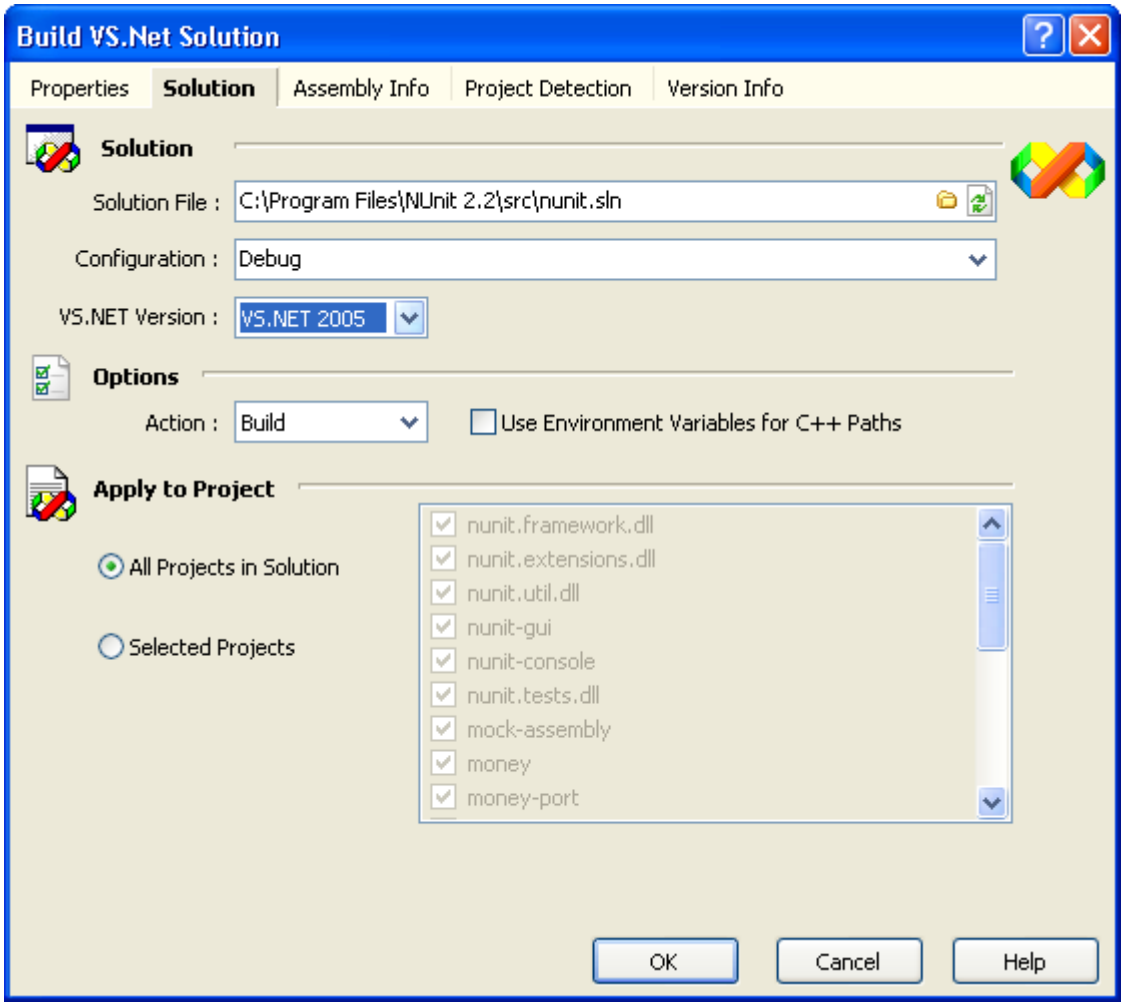


When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

5.12.2.3 Visual Studio .NET Action

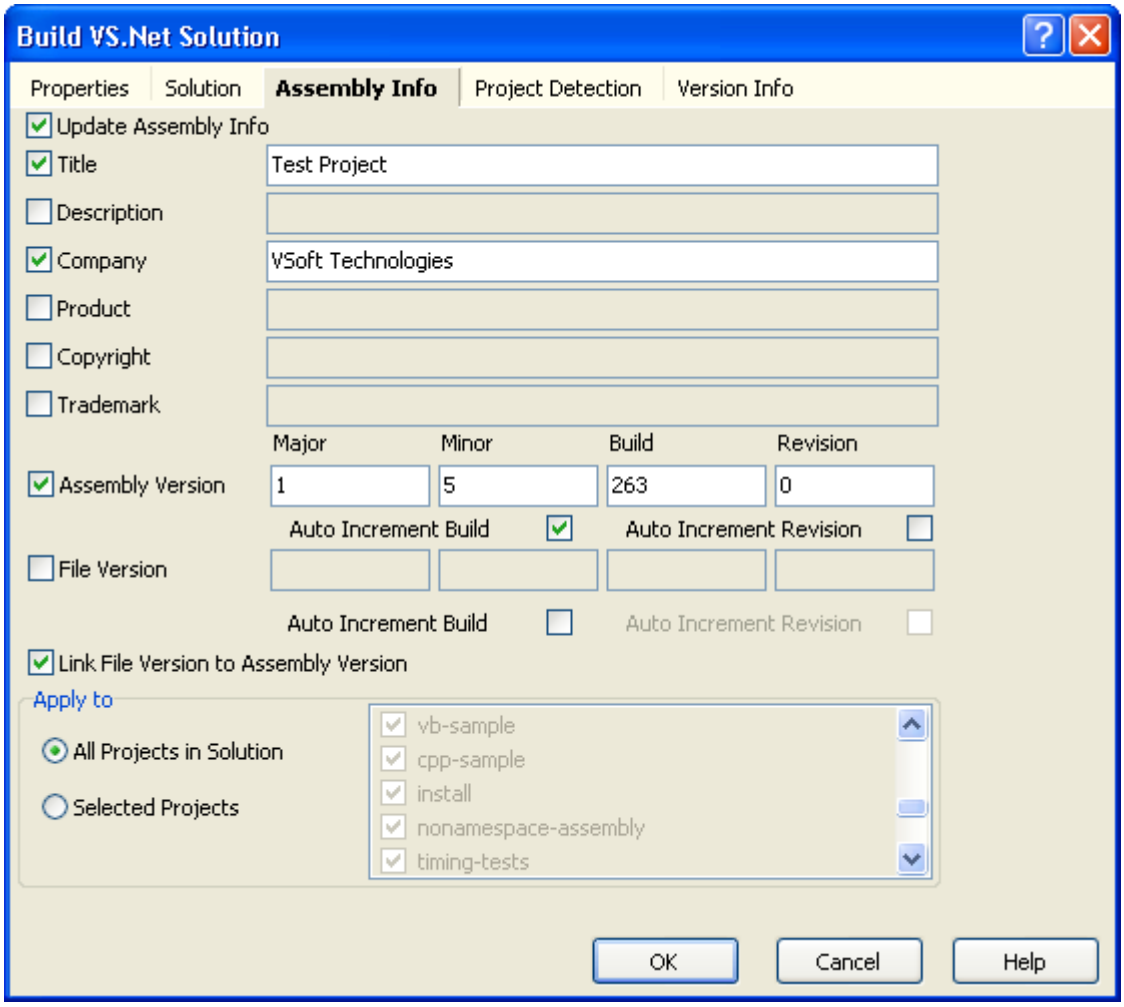
This action provides the ability to build Visual Studio .NET solutions using FinalBuilder. You can choose to build, rebuild or clean the whole solution or selected projects in the solution. You can also choose which Solution Configuration to use.

© 2001-2006 VSoft Technologies Pty Ltd



The Assembly Info page provides the facility to update Assembly info in all or selected projects in the solution. Note that this will not work for ASP.NET projects in the solution. Support for updating win32 version info in unmanaged C++ projects will be added in the next update.

© 2001-2006 VSoft Technologies Pty Ltd



5.12.2.3.1 Using PropertySets with VisualStudio.Net

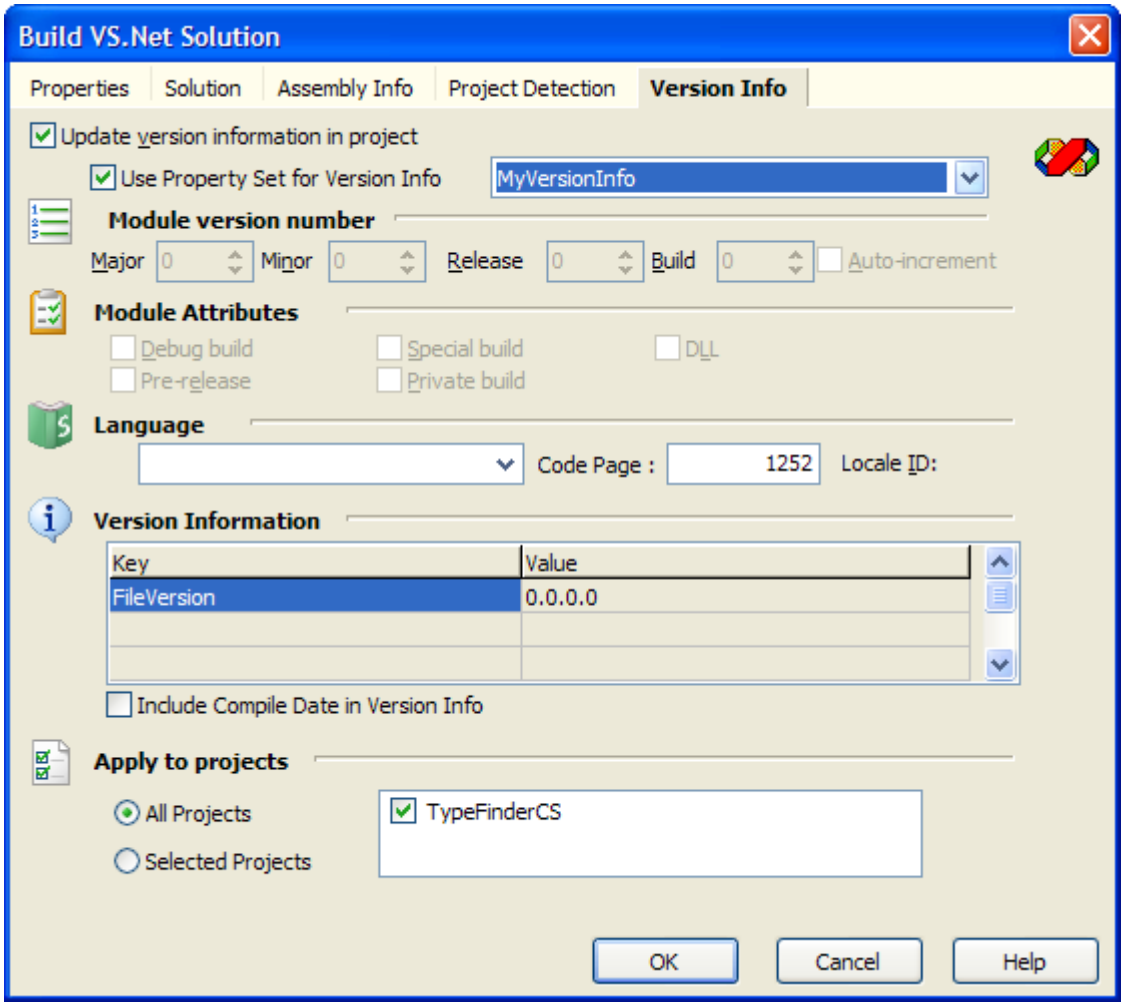
To use PropertySets with VisualStudio.Net, first you need to define a property set using the PropertySet define action and choose either Win32 Version Numbers or Win32 Version Info if you want to update the Version Info, and/or .Net Assembly Numbers or .Net Assembly Info if you want to update the Assembly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

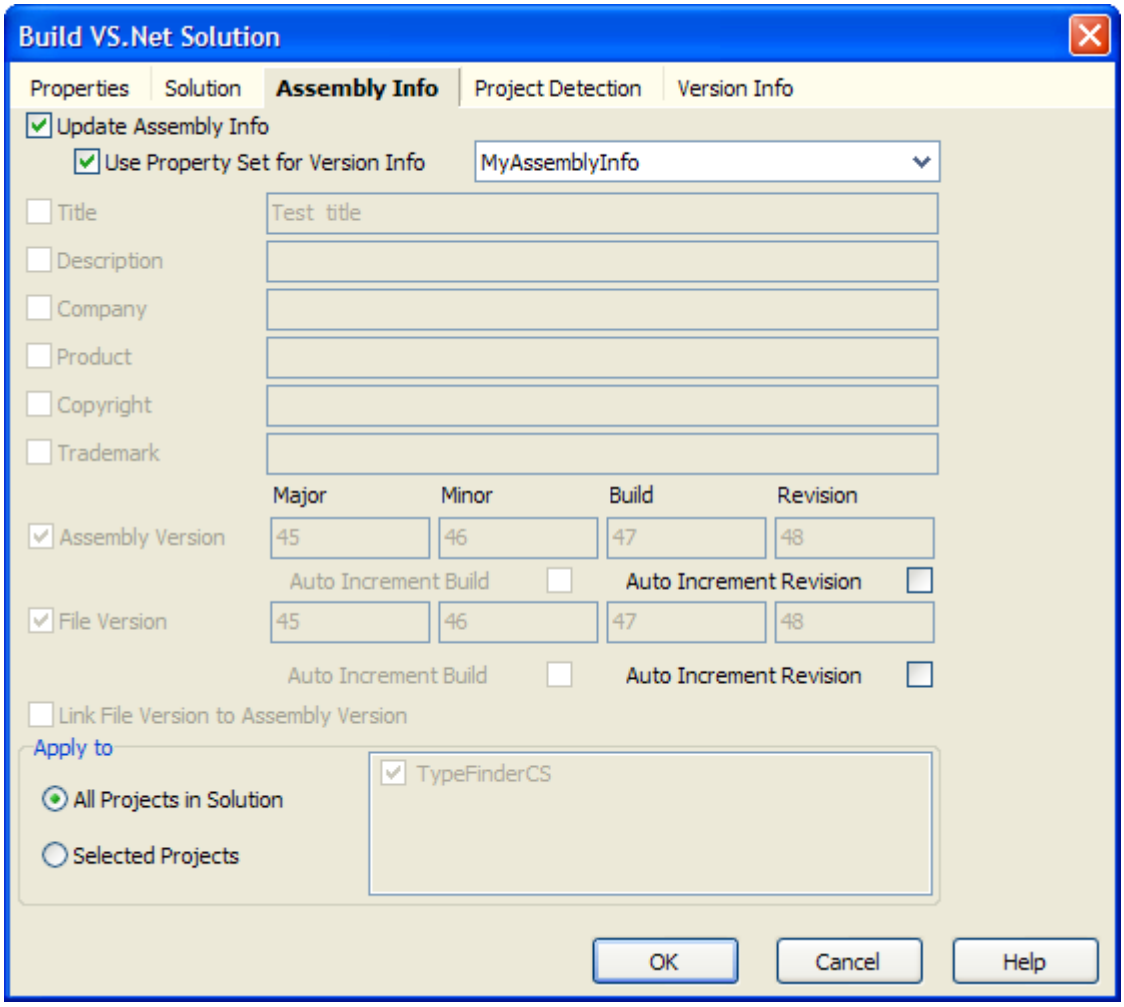
eg. For Version Info (must use a Win32 PropertySet)

© 2001-2006 VSoft Technologies Pty Ltd



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

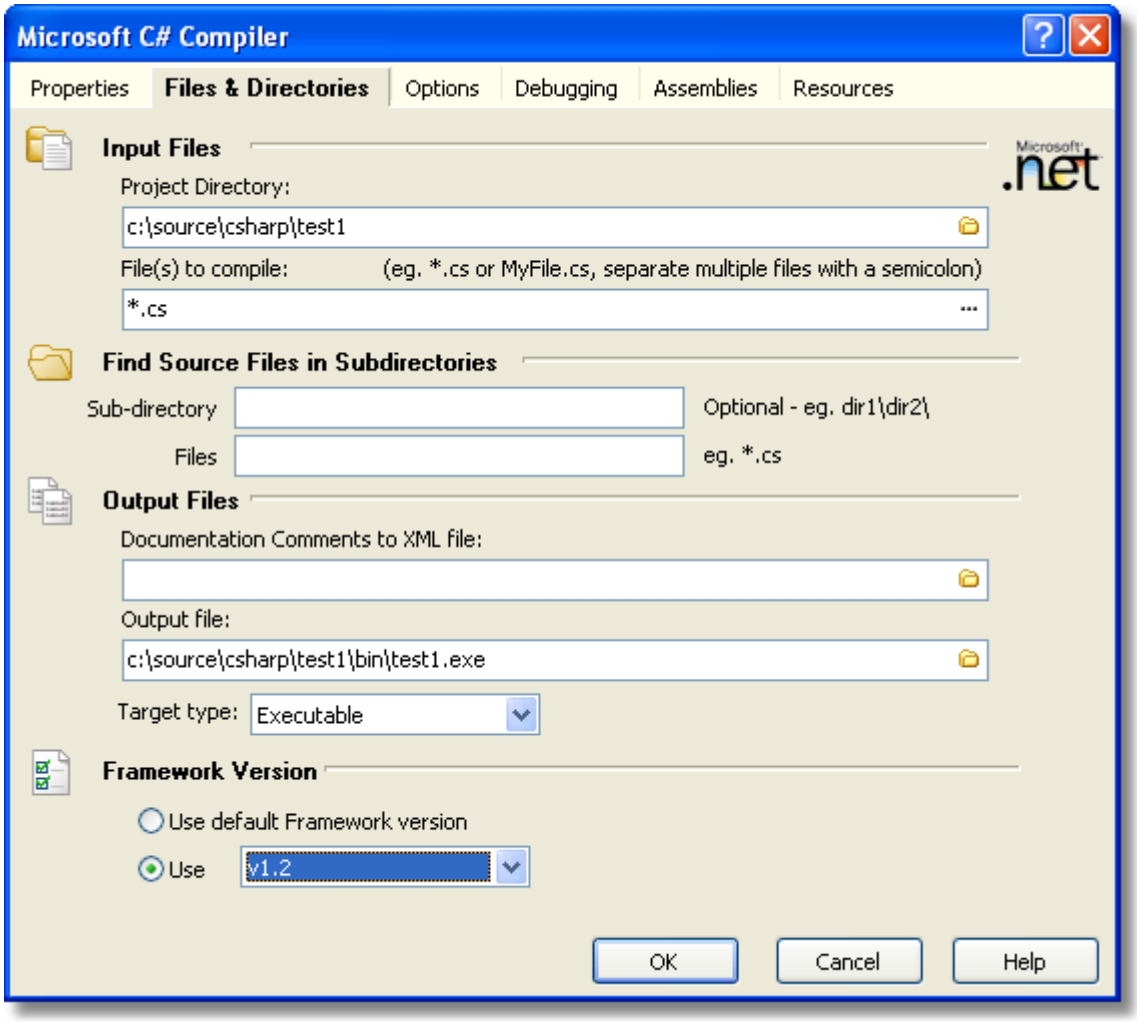
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.12.2.4 Microsoft C# Compiler Action

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



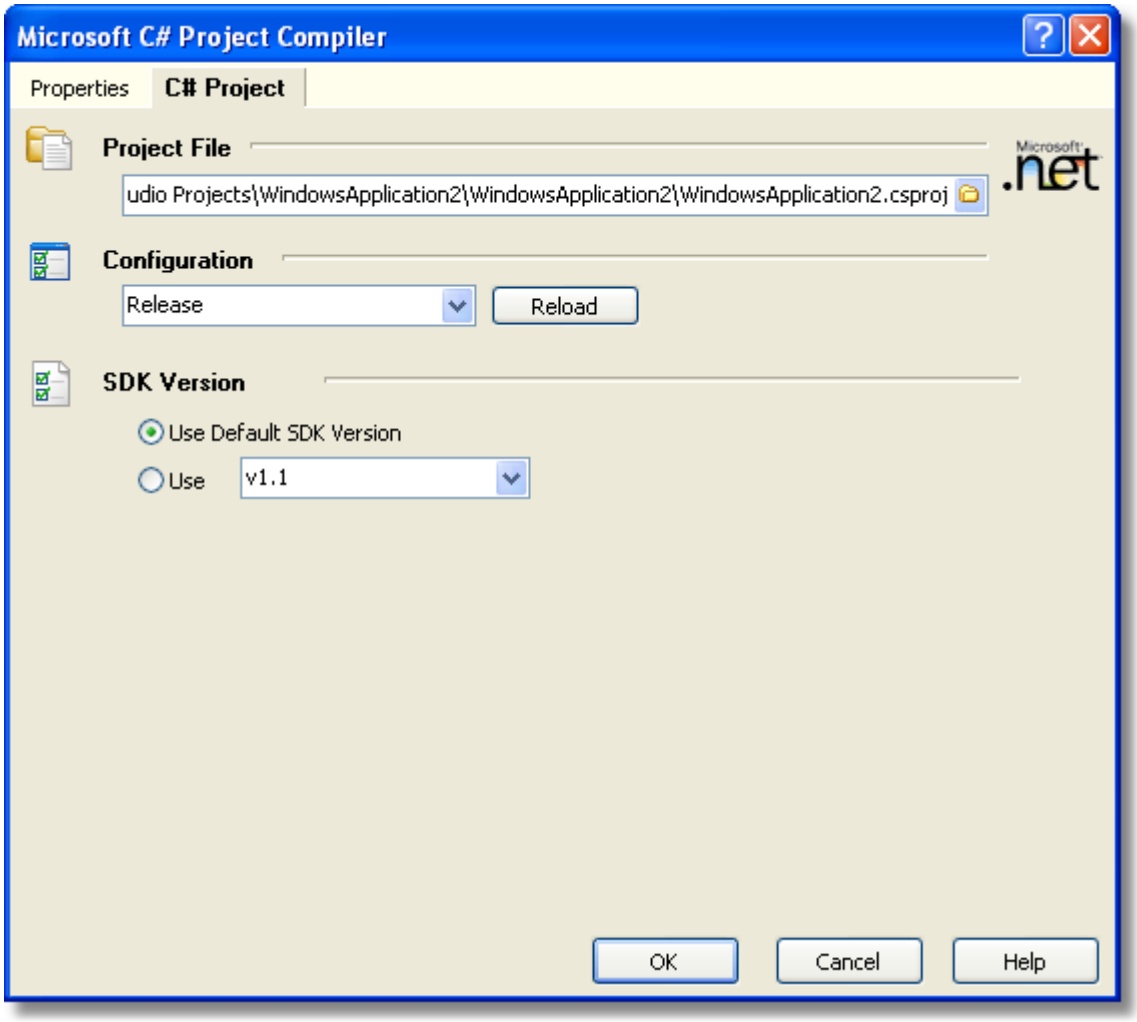
Information on the compiler options can be found in the .NET Framework documentation on MSDN :

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

5.12.2.5 Microsoft C# Project Compiler Action

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

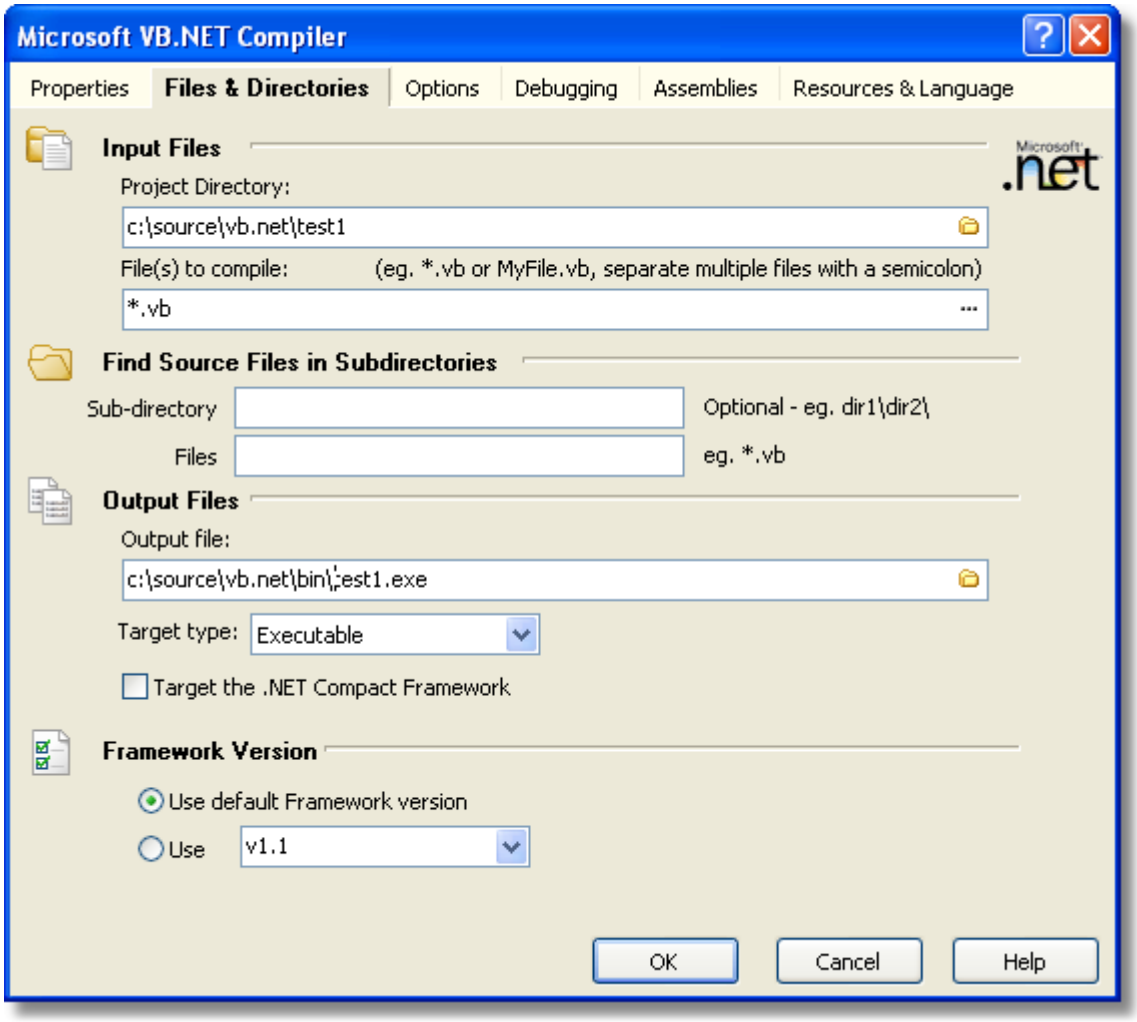
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.12.2.6 Microsoft VB.NET Compiler Action

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



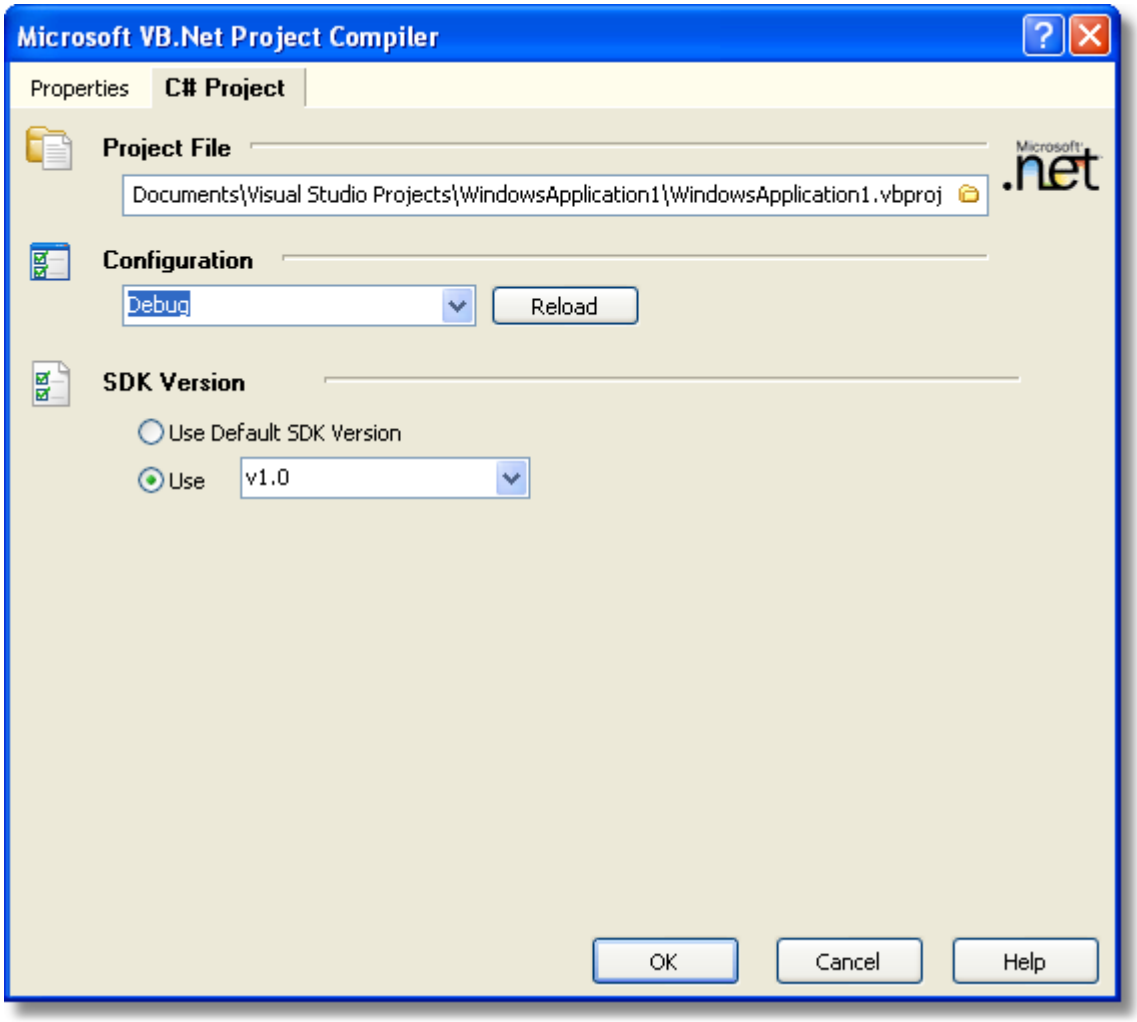
Information on the compiler options can be found in the .NET Framework documentation on MSDN :

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vblr7/html/vaconbuildingfromcommandline.asp

5.12.2.7 Microsoft VB.NET Project Compiler Action

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

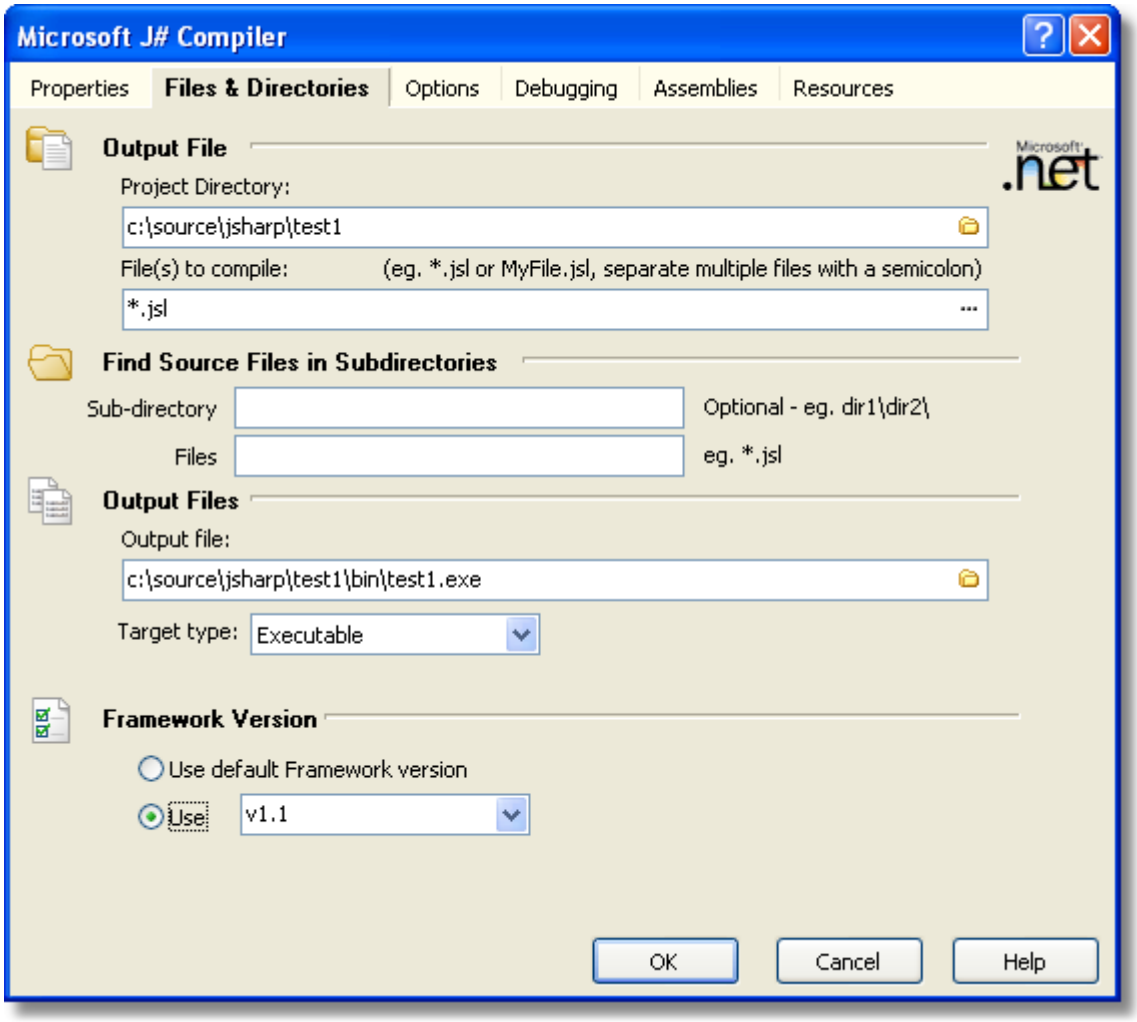
© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd

5.12.2.8 Microsoft J# Compiler Action

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



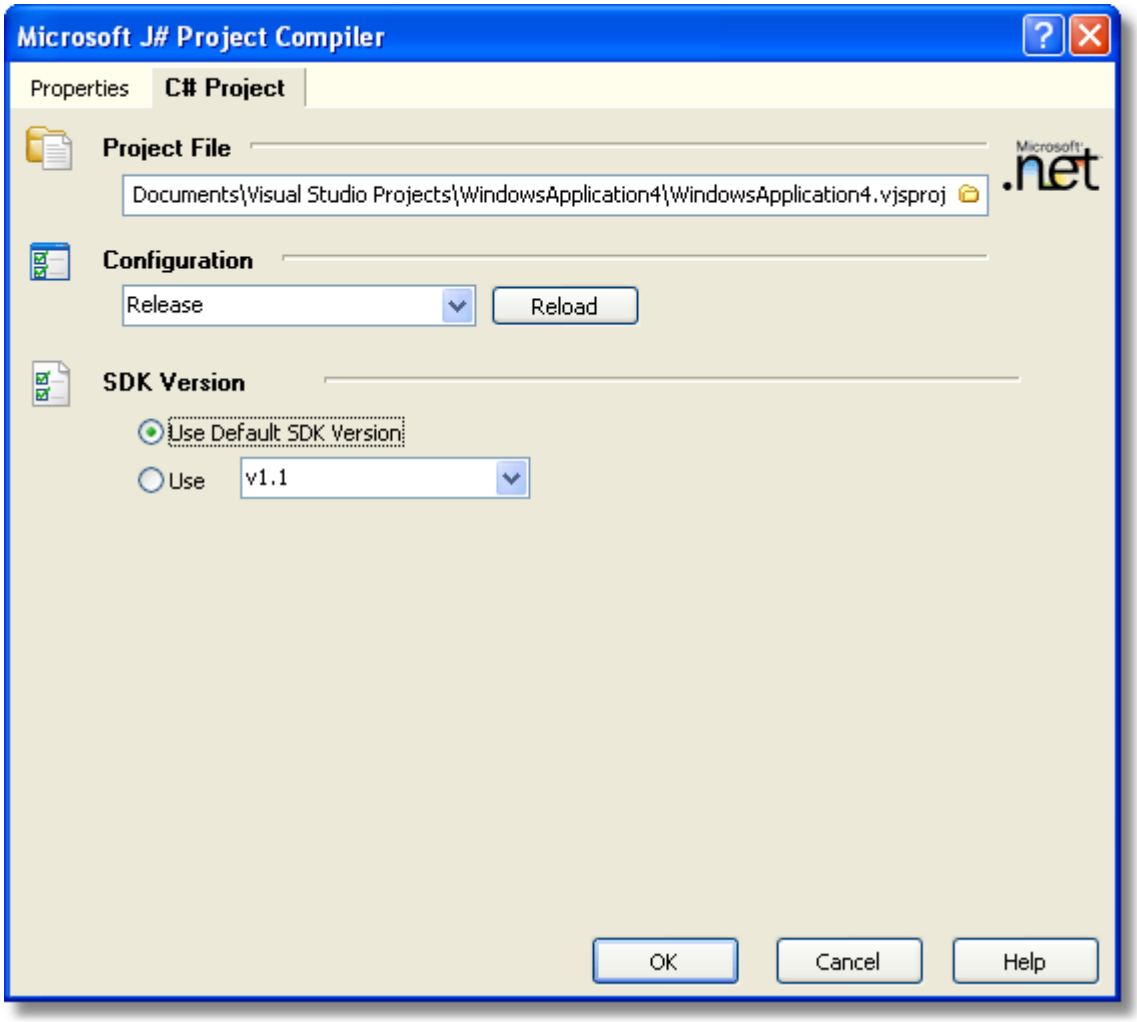
Information on the compiler options can be found in the .NET Framework documentation on MSDN :

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

5.12.2.9 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.

© 2001-2006 VSoft Technologies Pty Ltd



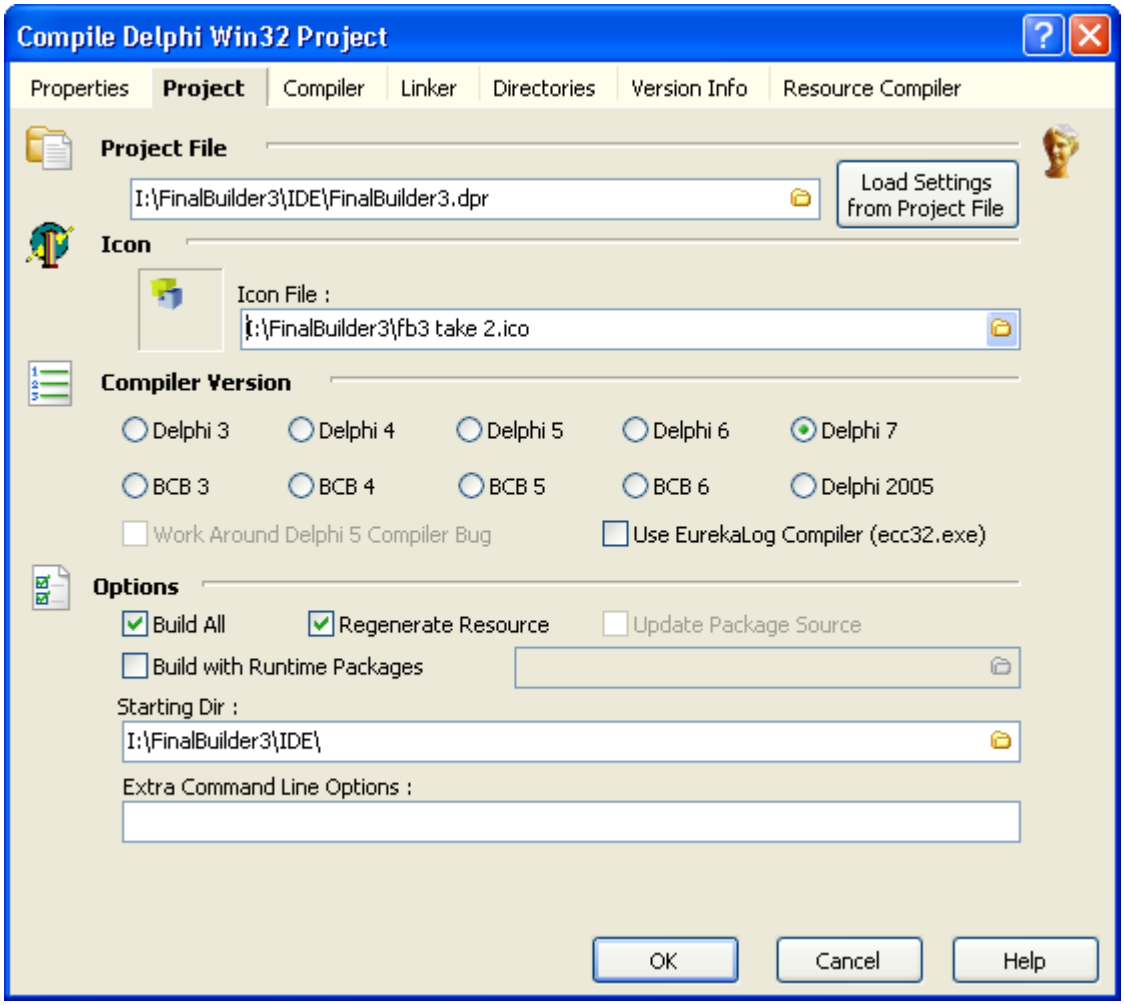
© 2001-2006 VSoft Technologies Pty Ltd

5.12.3 Borland

5.12.3.1 Compile Delphi Project

This action provides a familiar interface to the Delphi command line compiler. The options on the Compiler, Linker, Directories and Version Info are described in the Delphi Help File. They provide the same options that are available when compiling a Delphi project from within the Delphi IDE. Note that you may use FinalBuilder Variables in the Directories dialog but no in the version info dialog.

This action can also 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 in the Project Tab pictured above :

Project File : This is the path to the Delphi project, Delphi package or unit file you wish to compile.

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 Borland Resource Compiler to do this. If you use

© 2001-2006 VSoft Technologies Pty Ltd

version info in your project and increment any version numbers then you should leave this checked.

Load Settings from DOF File : The Delphi IDE maintains a file with the extension .dof with the project compiler/linker settings, version info, search path and output directories. Clicking this button will cause FinalBuilder to read this file and apply those settings to this action. You should do this when you first create the action. After that, the settings are maintained separately from the Delphi IDE. If you make changes to your project in the IDE that affect the operation of the program then you should reload these settings in FinalBuilder so that it compiles correctly from FinalBuilder.

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 (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.

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.

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 are :

```
  property CompilerOpt : IDelphiCompilerOptions; // see 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 Py 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
```

© 2001-2006 VSoft Technologies Pty Ltd

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

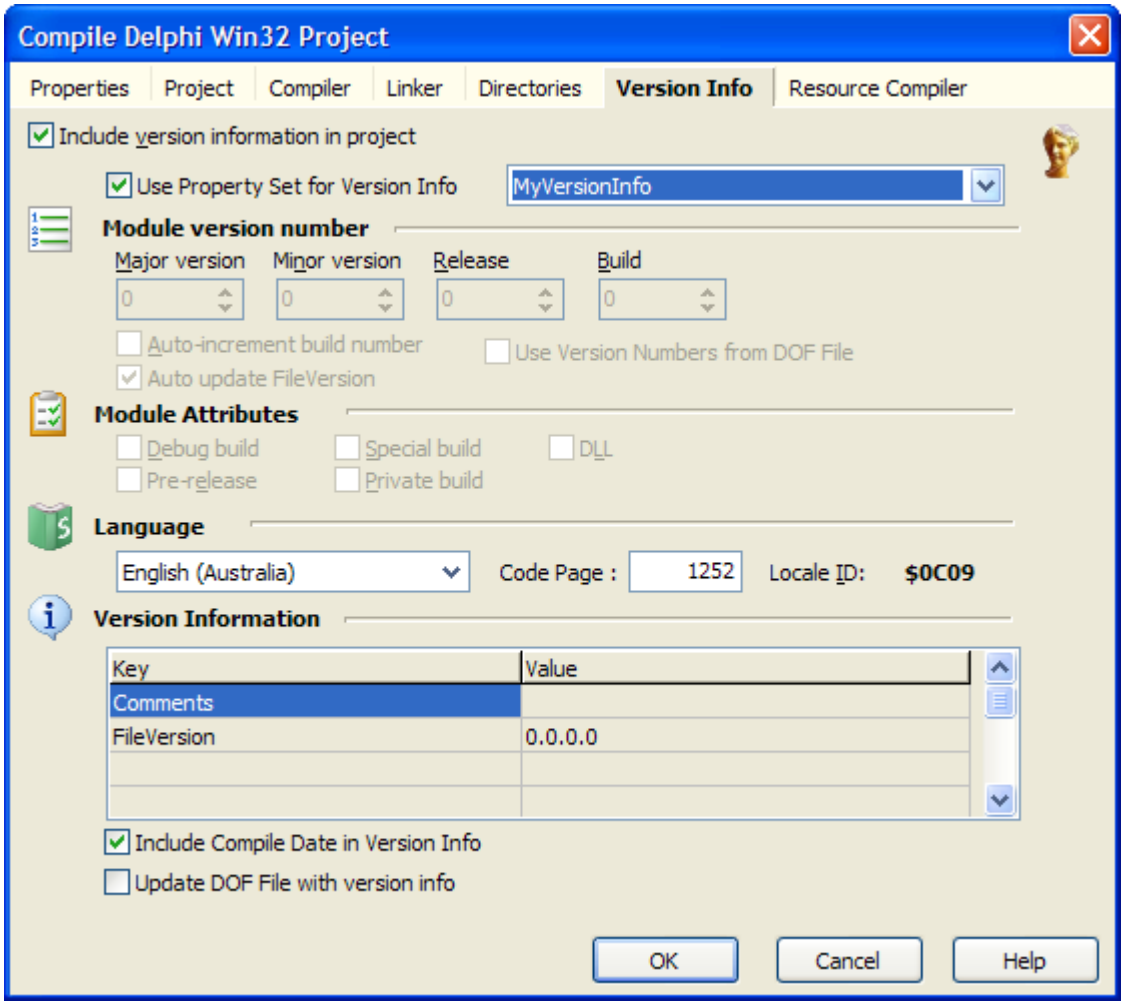
Action methods :
procedure SetVersionInfoKey(name : WideString; Value : WideString);
function GetVersionInfoKey(name : WideString) : WideString;

© 2001-2006 VSoft Technologies Pty Ltd

5.12.3.1.1 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:

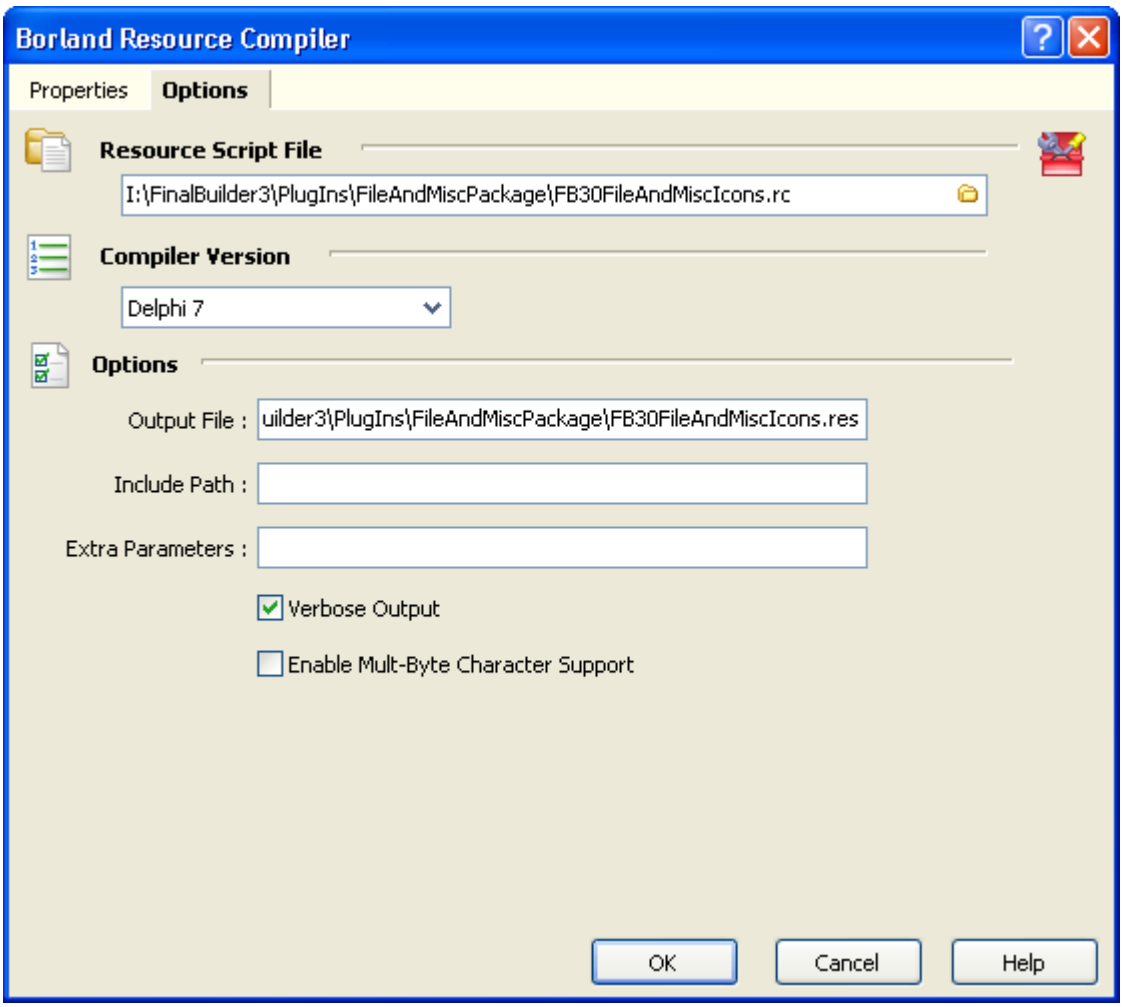


When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

© 2001-2006 VSoft Technologies Pty Ltd

5.12.3.2 Compile Borland Resource Script

This action provides an easy to use interface for the Borland 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 Borland Resource compiler is provided as part of Delphi And C++Builder. This option allows you to choose which version of dcc32 to use. Note that it includes Delphi 7 and C++Builder 6 & 7 which do not yet exist! This is for future use only!

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;
```

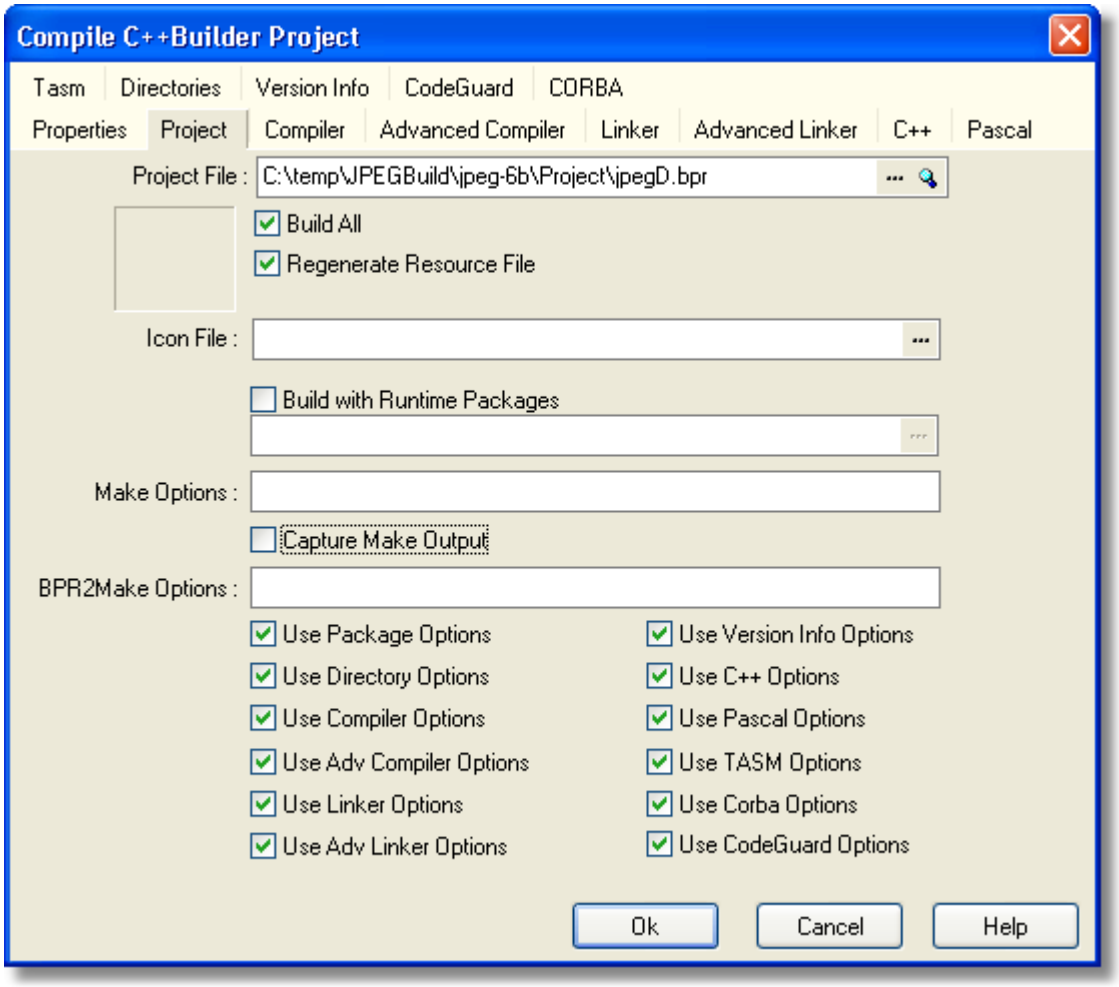
© 2001-2006 VSoft Technologies Pty Ltd

```
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;
```

5.12.3.3 Borland C++ Builder Action

This action provides support for Borland'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 :

© 2001-2006 VSoft Technologies Pty Ltd

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 chose 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.

Borland C++Builder Scripting Reference

5.12.3.3.1 Borland 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;
```

© 2001-2006 VSoft Technologies Pty Ltd

```
    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 CaptureMakeOuput : 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;
```

Borland C++ Builder Compiler Options Scripting Reference

Borland C++ Builder Linker Options Scripting Reference

Borland C++ Builder Advanced Compiler Options Scripting Reference

Borland C++ Builder Advanced Linker Options Scripting Reference

Borland C++ Builder TLib Options Scripting Reference

Borland C++ Builder CPP Options Scripting Reference

Borland C++ Builder Pascal Options Scripting Reference

Borland C++ Builder TASM Options Scripting Reference

Borland C++ Builder Corba Options Scripting Reference

© 2001-2006 VSoft Technologies Pty Ltd

Borland C++ Builder CodeGuard Options Scripting Reference

5.12.3.3.2 Borland 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;

5.12.3.3.3 Borland 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;

© 2001-2006 VSoft Technologies Pty Ltd

Valid values are : lcBorland,lcANSI,lcUnixV,lcKR

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

5.12.3.3.4 Borland 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: intege;
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;

5.12.3.3.5 Borland 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;

5.12.3.3.6 Borland C++Builder CPP Options

Scripting Info

The Action.CppOpt properties available are :

property MemberPointers : TCBuilderMemberPointers;
Valid values are :

© 2001-2006 VSoft Technologies Pty Ltd

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 FastExceptionProlog: boolean;
property VirtualTables : TCBuilderVirtualTables;
Valid values are : vtSmart,vtLocal,vtExternal,vtPublic

property ForceCPPCompile: boolean

5.12.3.3.7 Borland C++Builder TASM Options

Scripting Info

The Action.TASMOpt properties available are :

property DebugInfo : TCBuilderTASMDebugInfo;
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 rea;
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;

5.12.3.3.8 Borland 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;

© 2001-2006 VSoft Technologies Pty Ltd

property UseRTL : boolean;

5.12.3.3.9 Borland 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;

5.12.3.3.10 Borland 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;

© 2001-2006 VSoft Technologies Pty Ltd

5.12.3.3.11 Borland C++Builder CodeGuard Options

Scripting Info

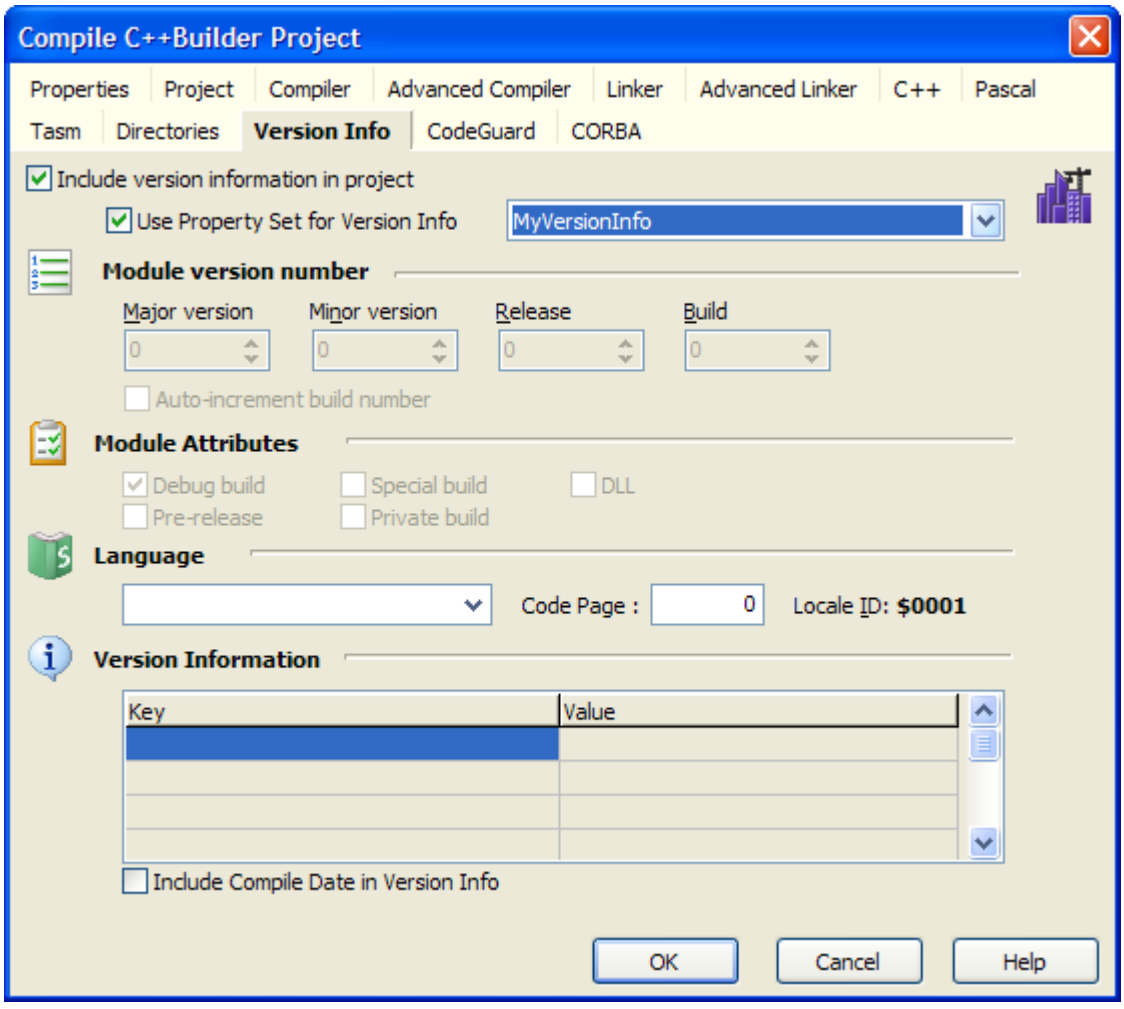
The Action.CodeGuardOpt properties available are :

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

5.12.3.3.12 Using PropertySets with Borland C++ Builder

To use PropertySets with Borland 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 Borland C++ Builder action on the Version Info property page, select "Use property set for version info" and select the PropertySet to use:

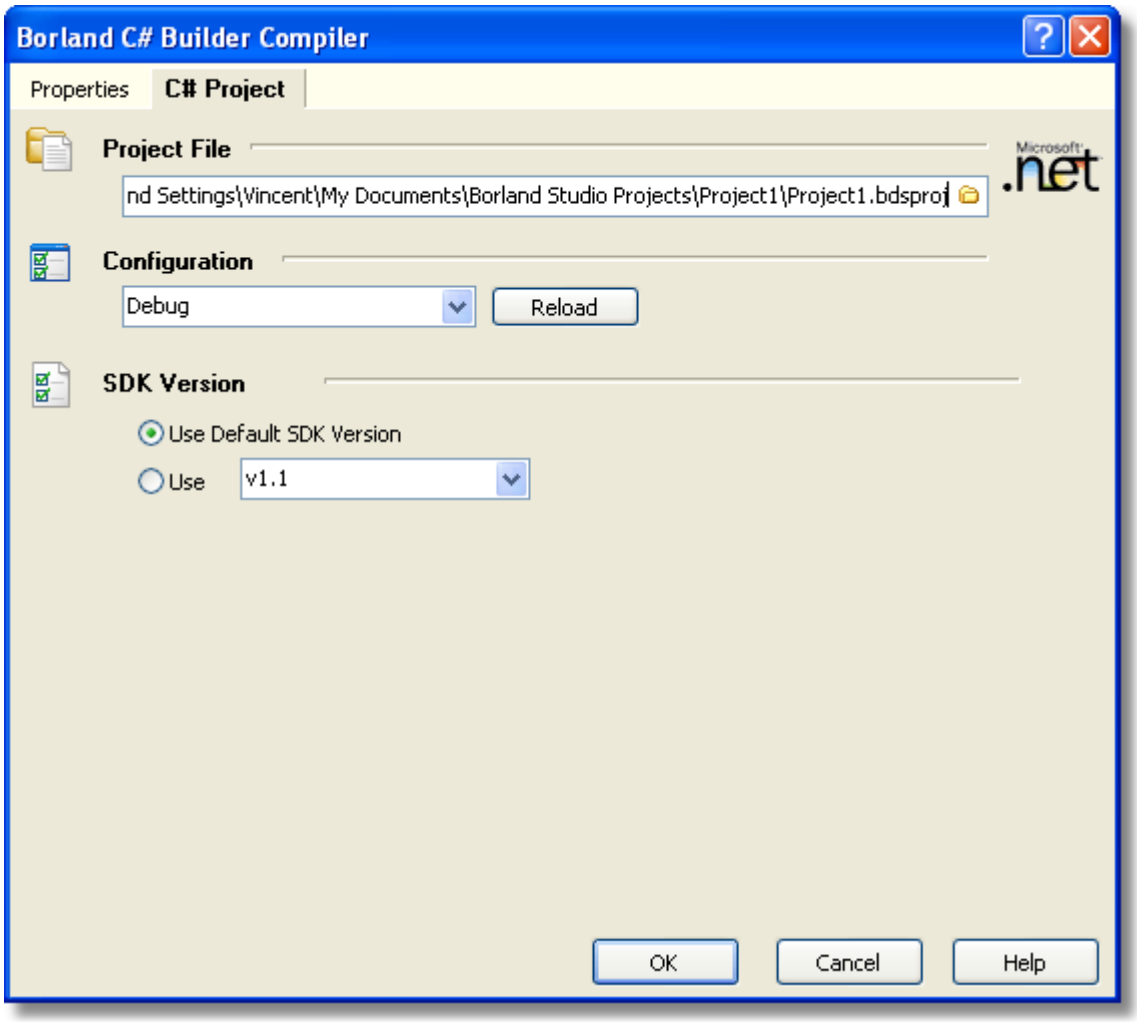


When the PropertySet is chosen, certain UI controls will be disabled as these values will be set automatically using the PropertySet.

© 2001-2006 VSoft Technologies Pty Ltd

5.12.3.4 Borland C# Builder Project Compiler Action

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



5.12.3.5 Borland Delphi 8 for .NET Action

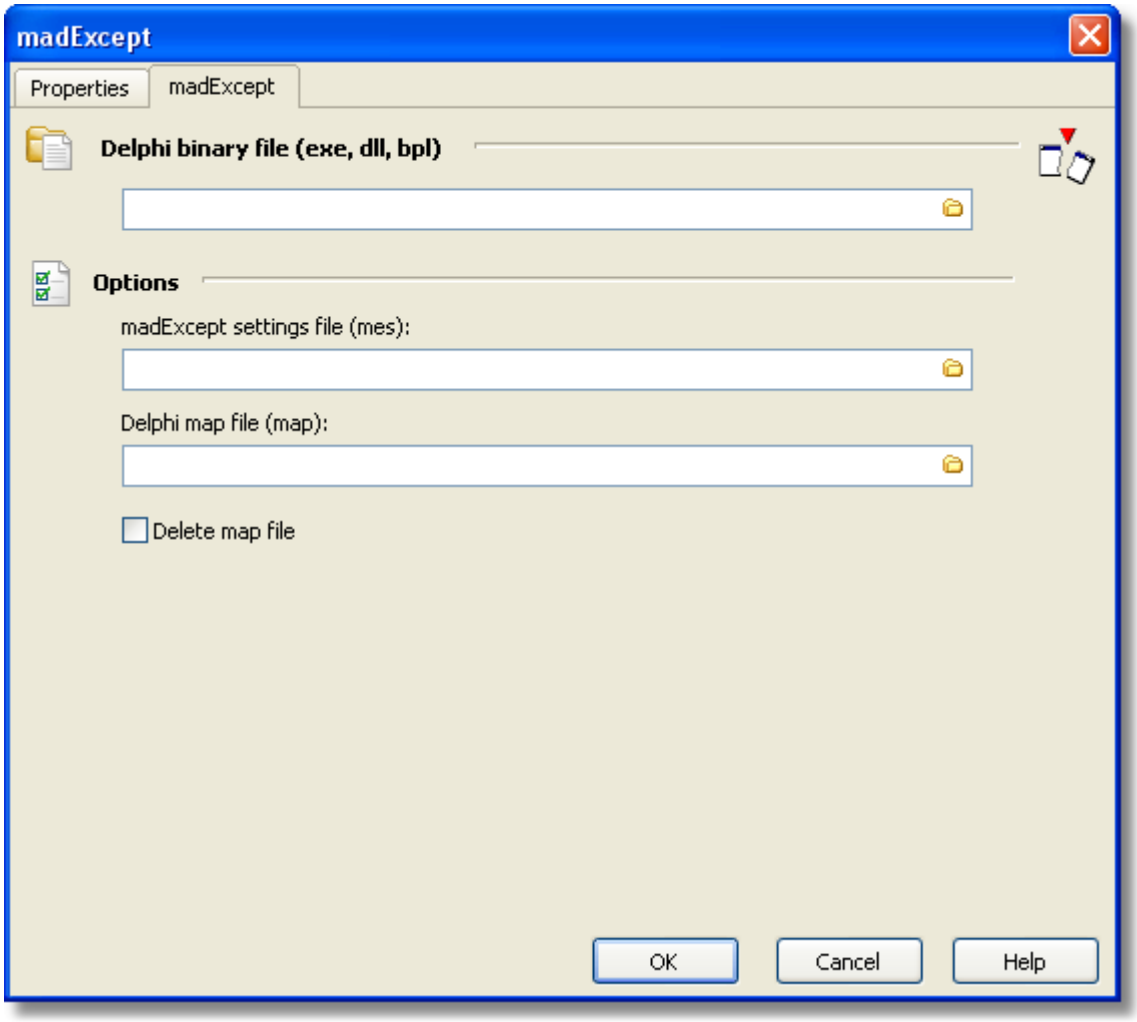
This action will build Borland Delphi™ 8 for .NET projects (.bdsproj). Delphi 8 is significantly different from earlier versions of Delphi, thus requiring a separate action.

© 2001-2006 VSoft Technologies Pty Ltd

5.12.4 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 Borland Delphi compiler which is why it's in the Compiler category.)



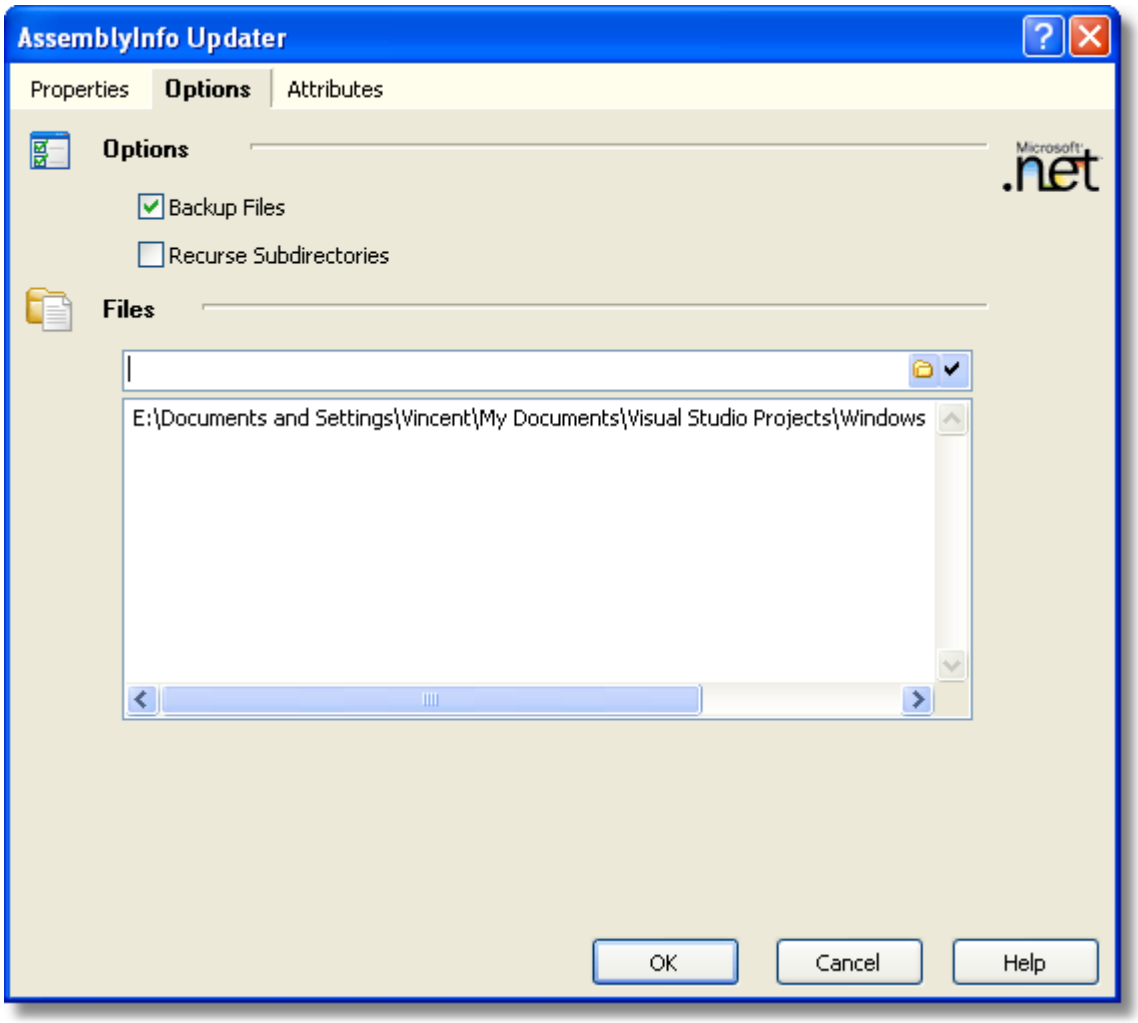
Specify a Delphi binary file and corresponding .Map and MadExcept settings (.Mes) file.

© 2001-2006 VSoft Technologies Pty Ltd

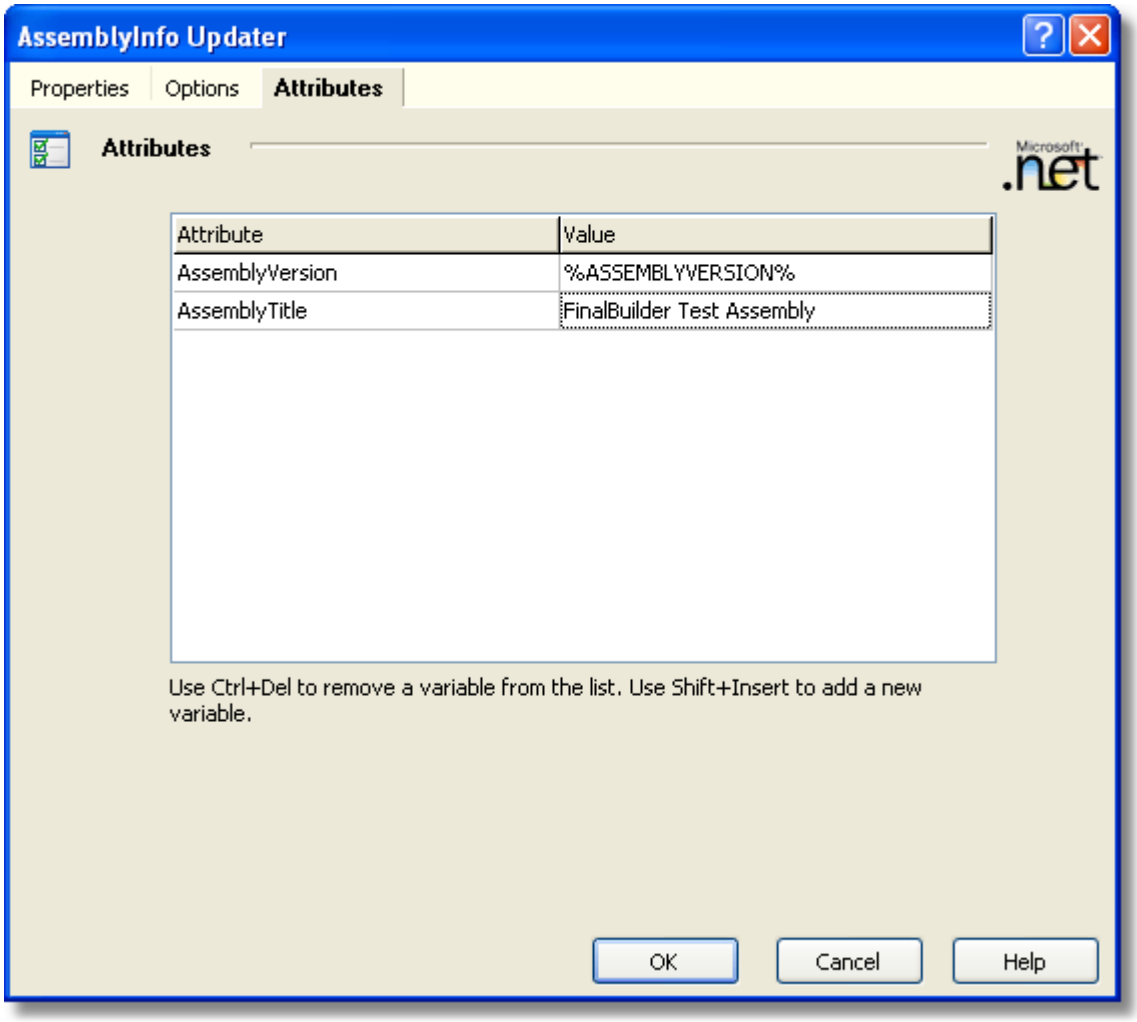
5.12.5 AssemblyInfo Updater Action

This action allows you to update the Assembly info in one or more source files. Languages supported are :

C#, VB.NET, J#, C++ and Delphi for .NET.



© 2001-2006 VSoft Technologies Pty Ltd



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. FB 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 FB will look to see if
the value is quoted and preserve the quotes.

© 2001-2006 VSoft Technologies Pty Ltd

5.12.6 Chrome

The Chrome action enables you to automate compilation of your Chrome projects as part of your build process

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/

© 2001-2006 VSoft Technologies Pty Ltd

5.12.7 Incredibuild

The IncrediBuild actions enable you to automate IncrediBuild distributed compiles as part of your build process.

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/

© 2001-2006 VSoft Technologies Pty Ltd

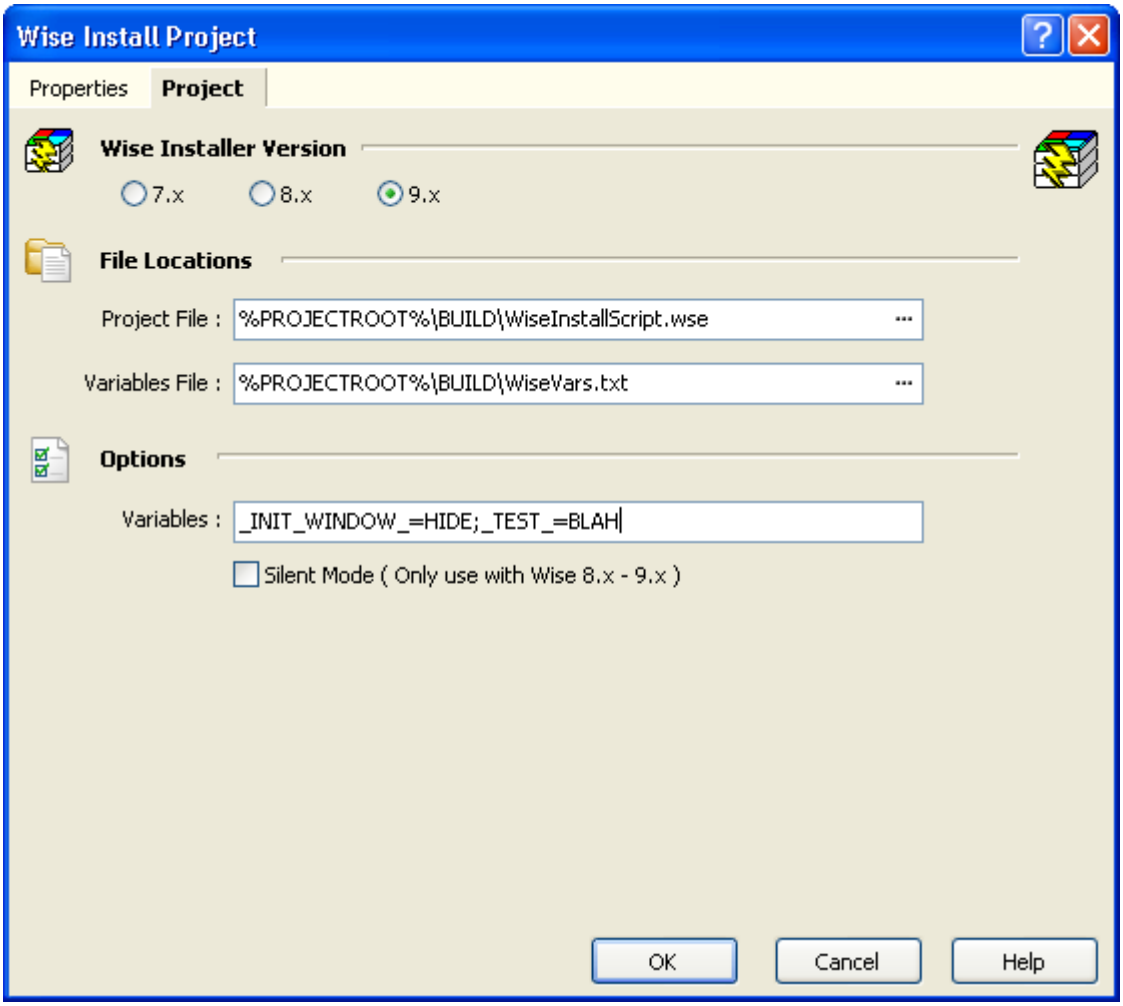
5.13 Installers

5.13.1 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

© 2001-2006 VSoft Technologies Pty Ltd

The Action properties available are :

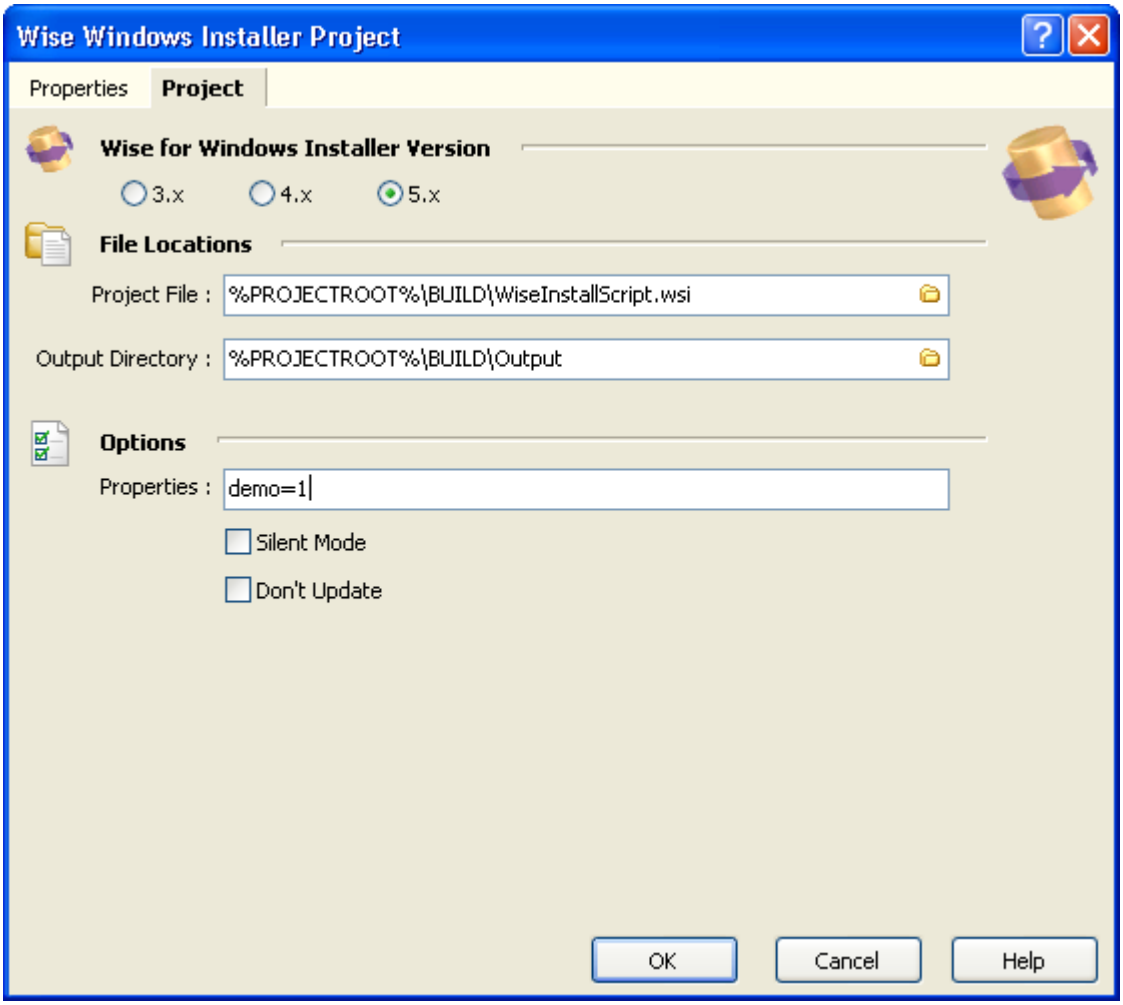
property ProjectFile : WideString
property Variables : WideString
property VariablesFile : WideString

© 2001-2006 VSoft Technologies Pty Ltd

5.13.2 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

© 2001-2006 VSoft Technologies Pty Ltd

5.13.3 InstallShield

5.13.3.1 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"
```

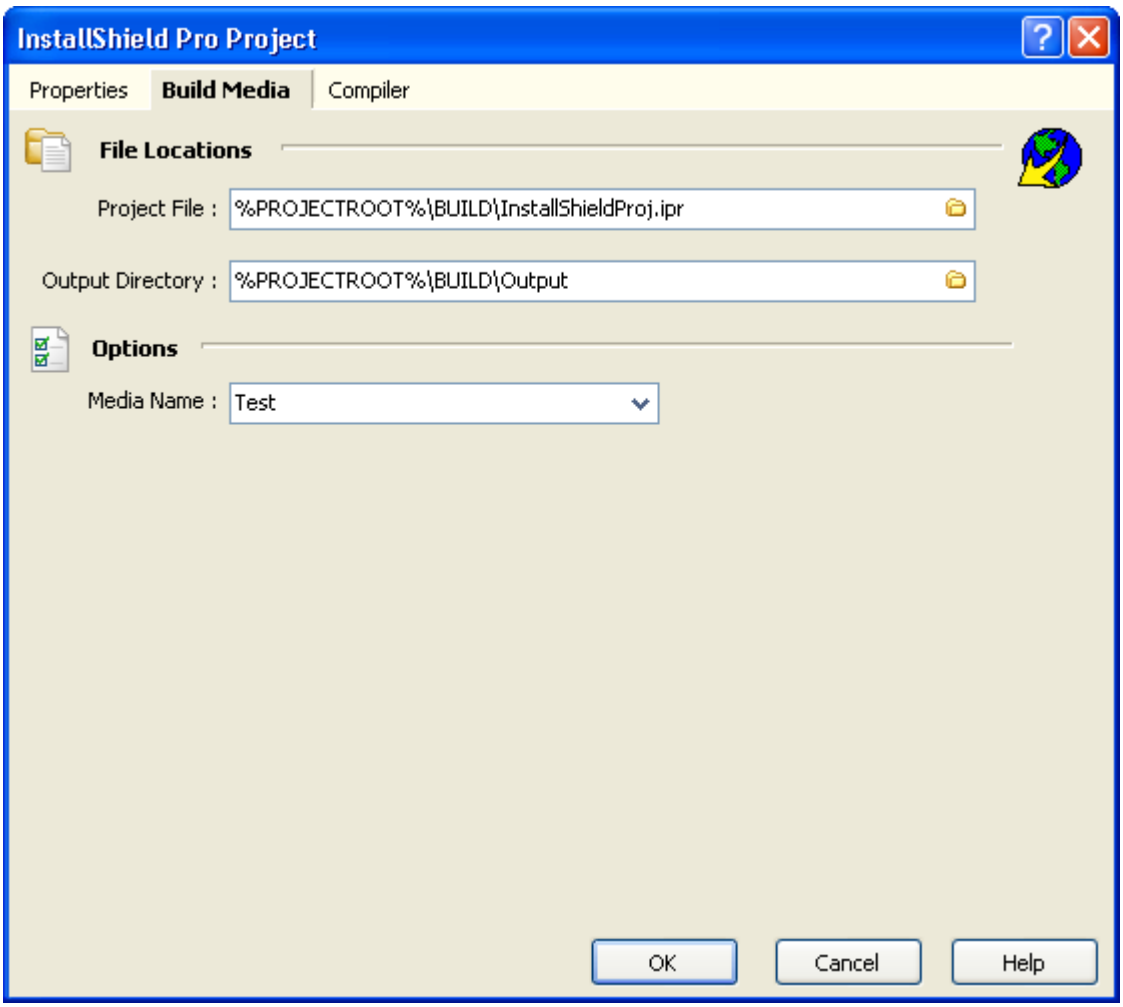
The available properties of the InstallShield object are:

- CompilerLocation // required for InstallShield Professional 6.x to 7.x
- IFXLocation // required for InstallShield Professional 6.x to 7.x
- ISRTLocation // 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

© 2001-2006 VSoft Technologies Pty Ltd

5.13.3.2 InstallShield Pro - Std Edition

This Action provides an interface to InstallShield Pro - Standard Edition version 6.2.x - 7.x



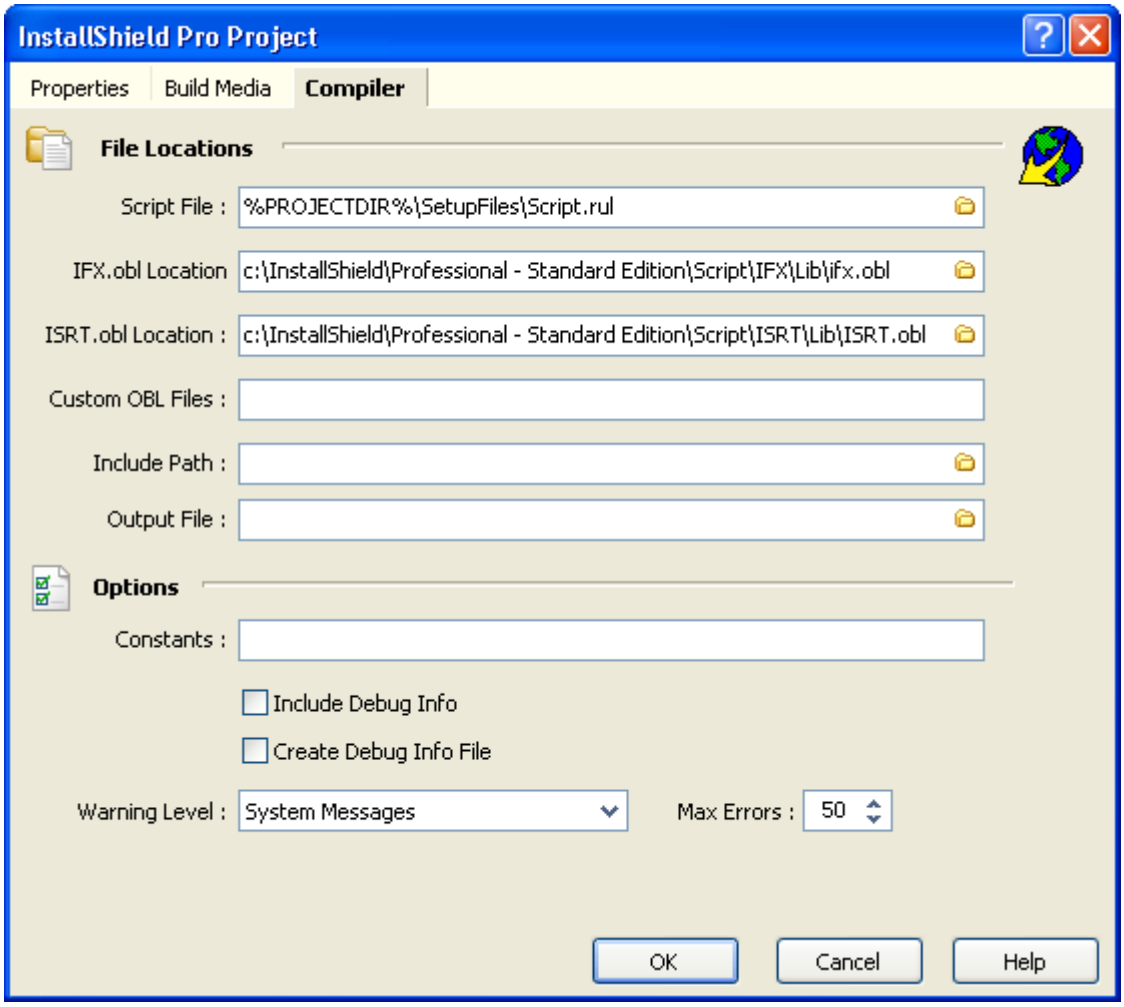
Project File : The fully qualified path to the setup project file (*.ipr).

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

Scripting Info

The Action properties available are :

property BuildMedia : WordBool // Build the Media
property CompileSetup : WordBool //CompileSetup // compile the setup program

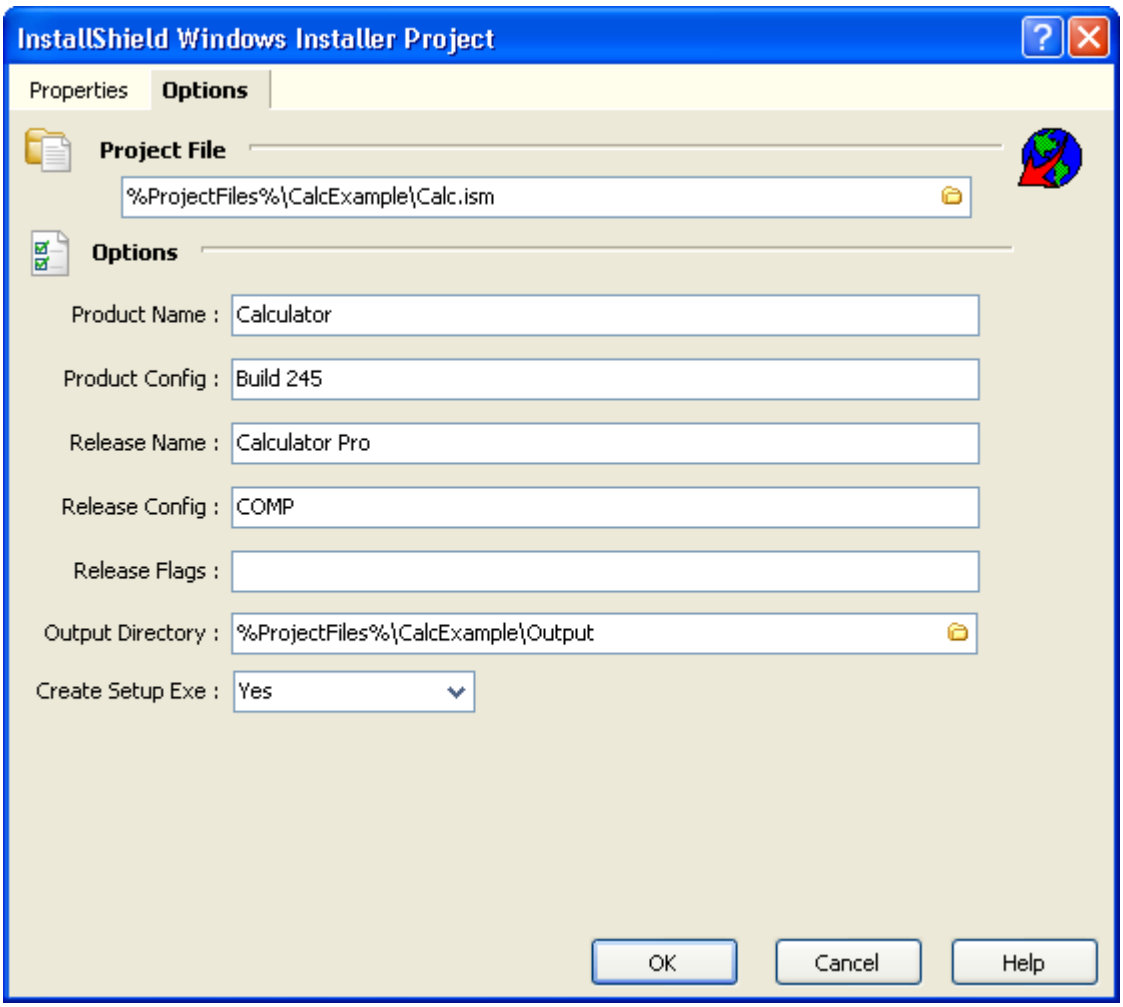
property ProjectFile : WideString
property OutputDirectory : WideString
property MediaName : WideString;

property SetupScriptFile : WideString
property IFXLocation : WideString
property ISRTLocation : 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

© 2001-2006 VSoft Technologies Pty Ltd

5.13.3.3 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.

© 2001-2006 VSoft Technologies Pty Ltd

Create Setup.Exe : Specify whether or not you would like to create a setup.exe along with your setup project.

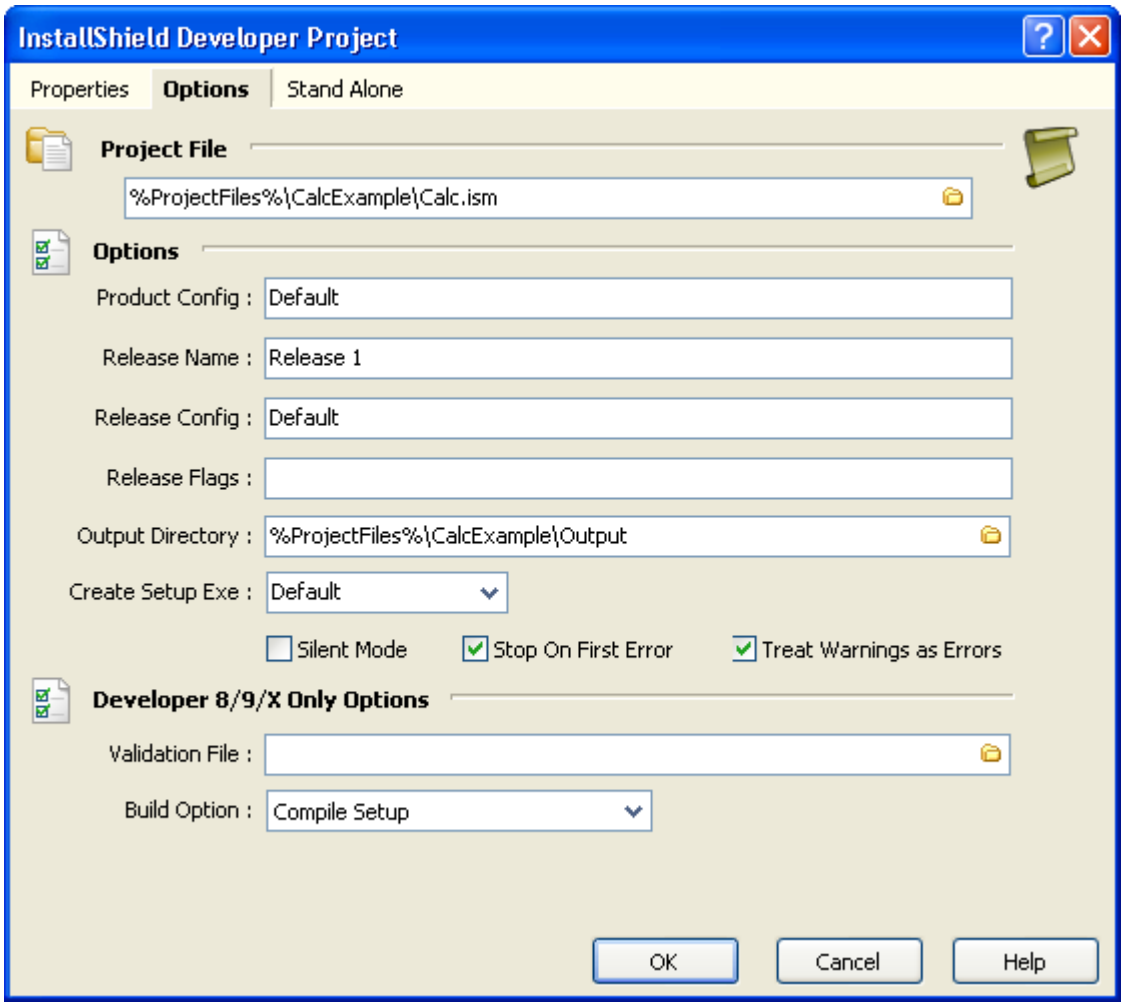
Scripting Info

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

5.13.3.4 InstallShield Developer

This Action provides an interface to InstallShield Developer 7.x - 11.5



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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

Silent Mode : Run the InstallShield compiler in silent mode, this will disable the output of compiler messages.

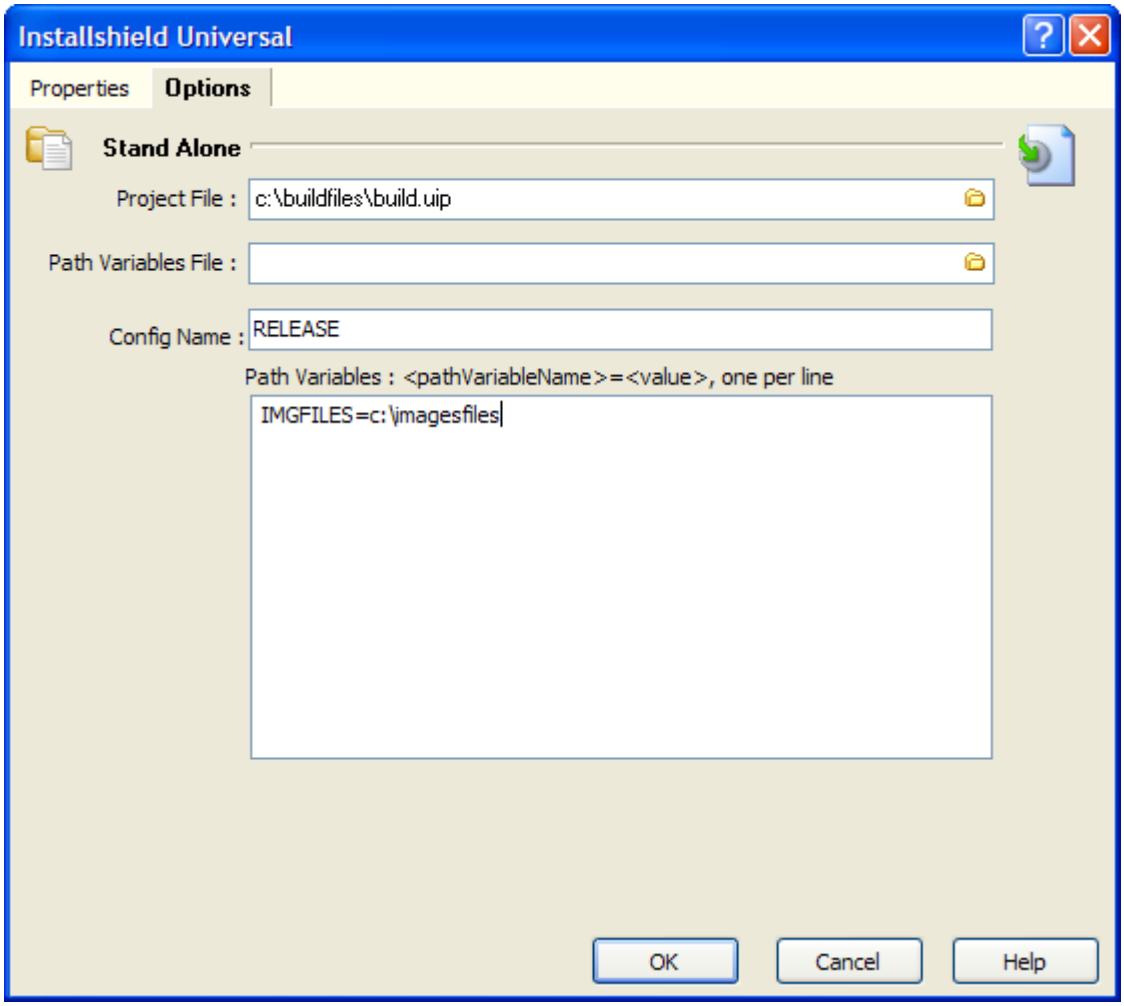
Treat Warnings as Errors: Any warnings issued by the Installshield compiler will be treated as errors.

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.13.3.5 InstallShield Universal Installer

The InstallShield Universal action enables you to automate the process of building InstallShield Universal installations as part of your build process.



Project File - specify your InstallShield Universal project file

Path Variables File - specify the fully qualified path to your variables file

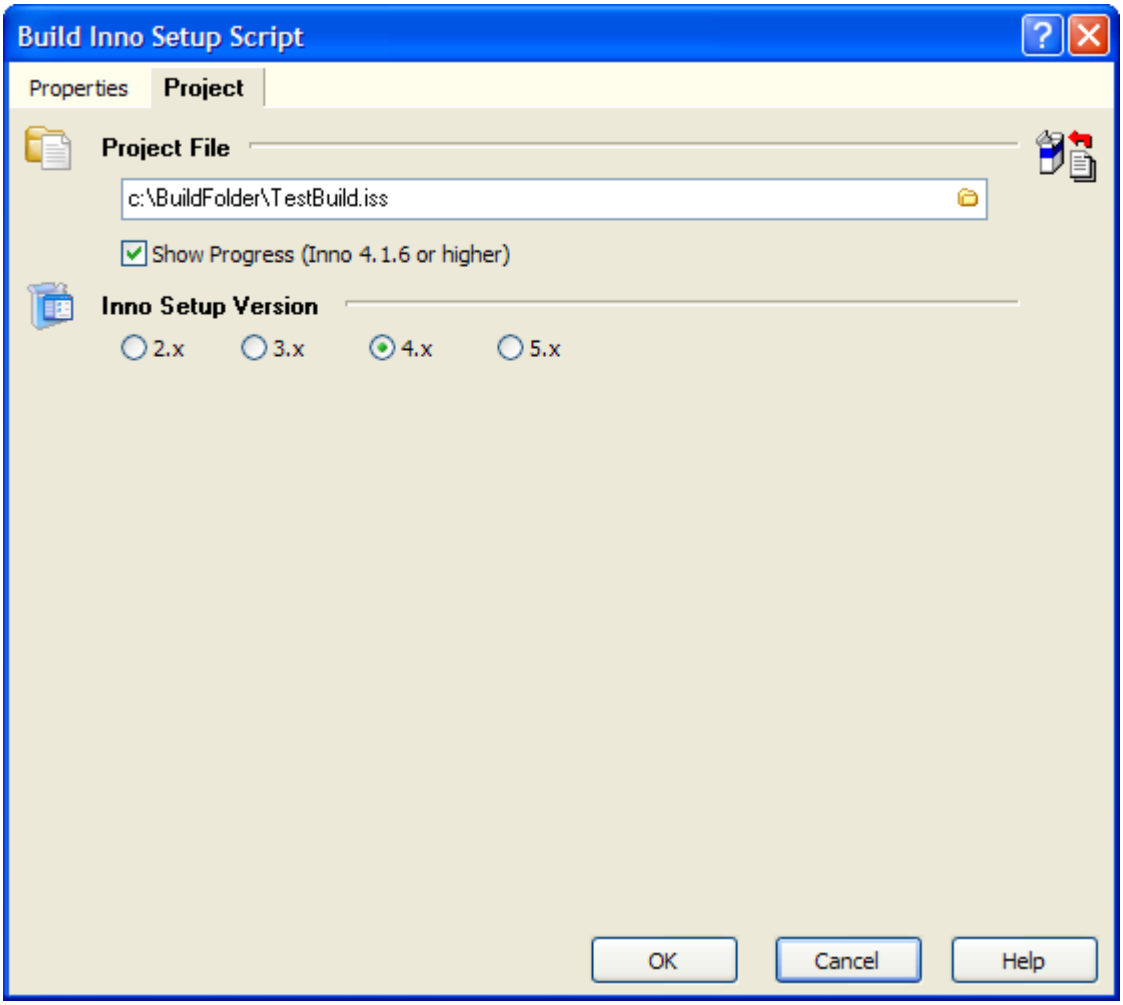
Config Name - enter the configuration you want to run. The configurations are defined in your project file.

Path Variables - enter name/value pairs defining the values for directory variables

© 2001-2006 VSoft Technologies Pty Ltd

5.13.4 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 (ISCmplr.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 :

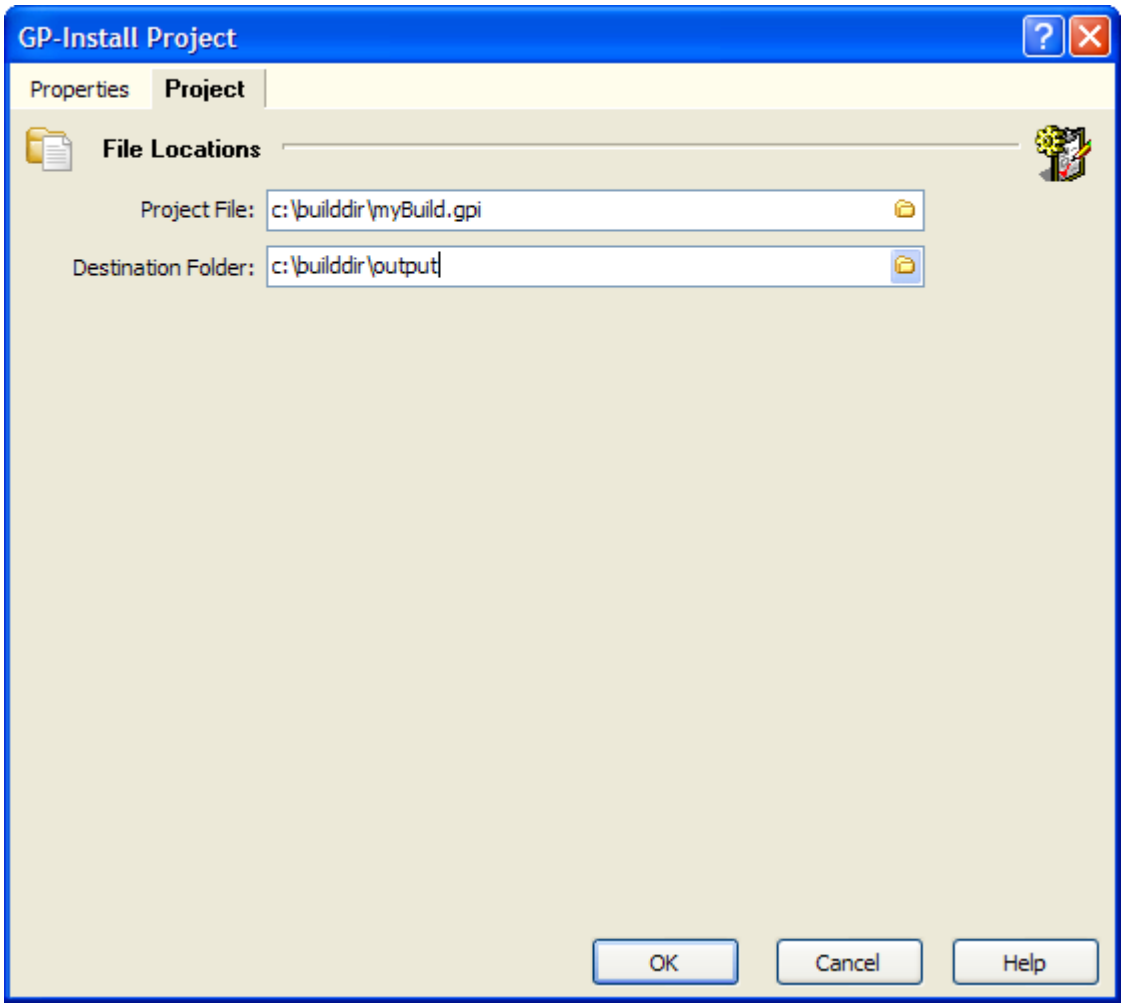
property ScriptFile : WideString
property InnoVersion : integer // 0 = 2.x, 1 = 3.x, 2 = 4.x, 3 = 5.x

Inno Setup is a freeware setup compiler which can be downloaded from :
http://www.jrsoftware.org/isinfo.htm

© 2001-2006 VSoft Technologies Pty Ltd

5.13.5 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;
```

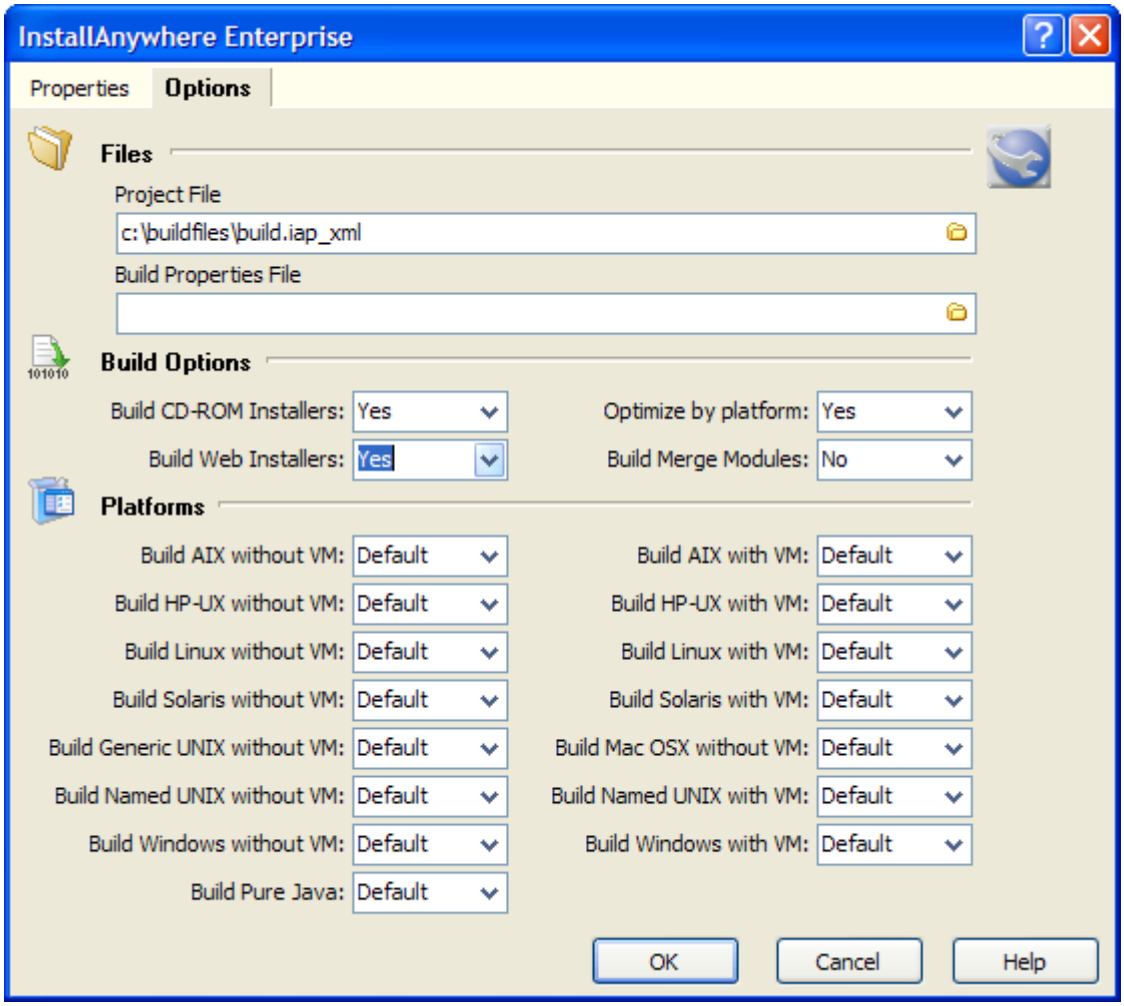
© 2001-2006 VSoft Technologies Pty Ltd

5.13.6 InstallAnywhere Enterprise

The InstallAnywhere Enterprise action enables you to automate the process of building ZeroG InstallAnywhere Enterprise installations as part of your build process.

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



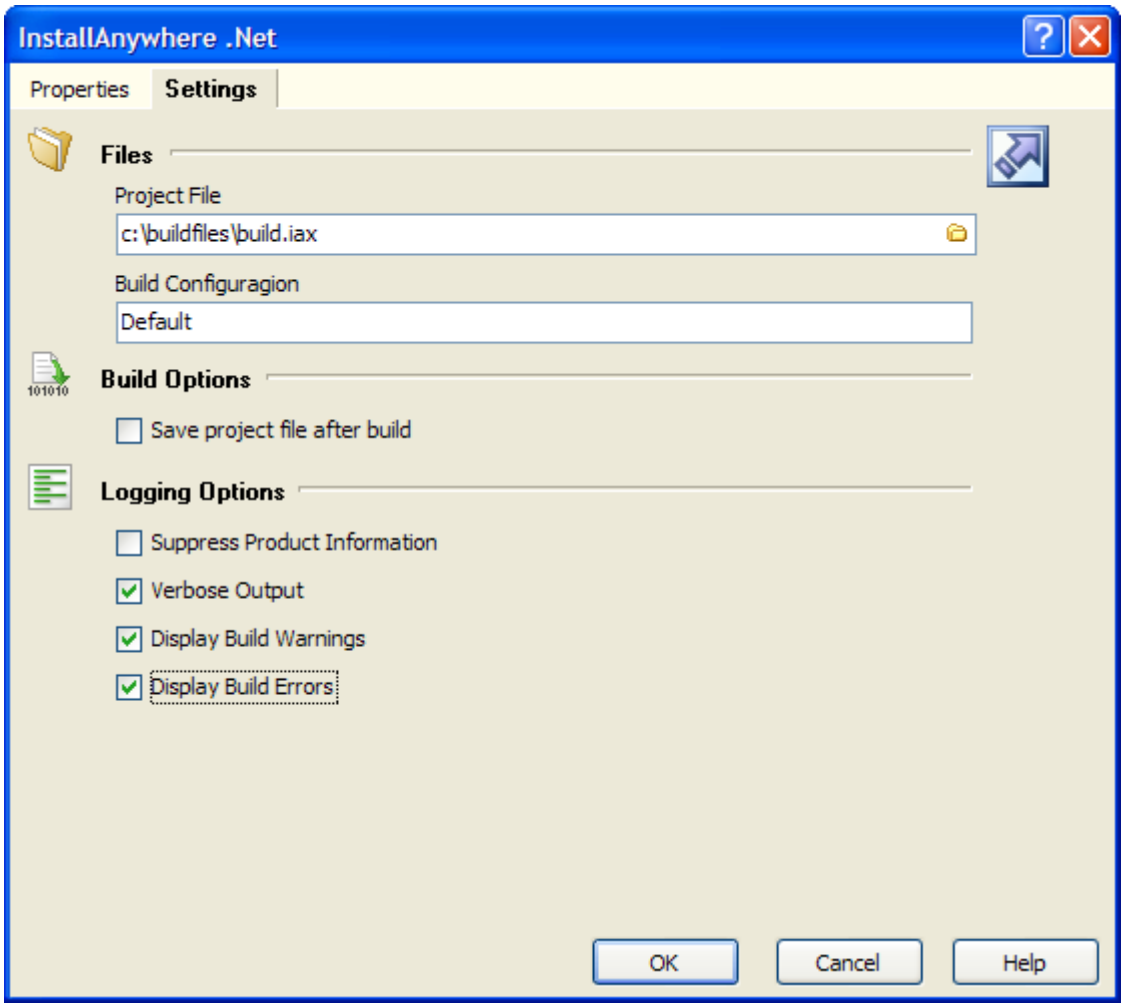
© 2001-2006 VSoft Technologies Pty Ltd

5.13.7 InstallAnywhere .Net

The InstallAnywhere.Net action enables you to automate the process of building ZeroG InstallAnywhere.Net installations as part of your build process.

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



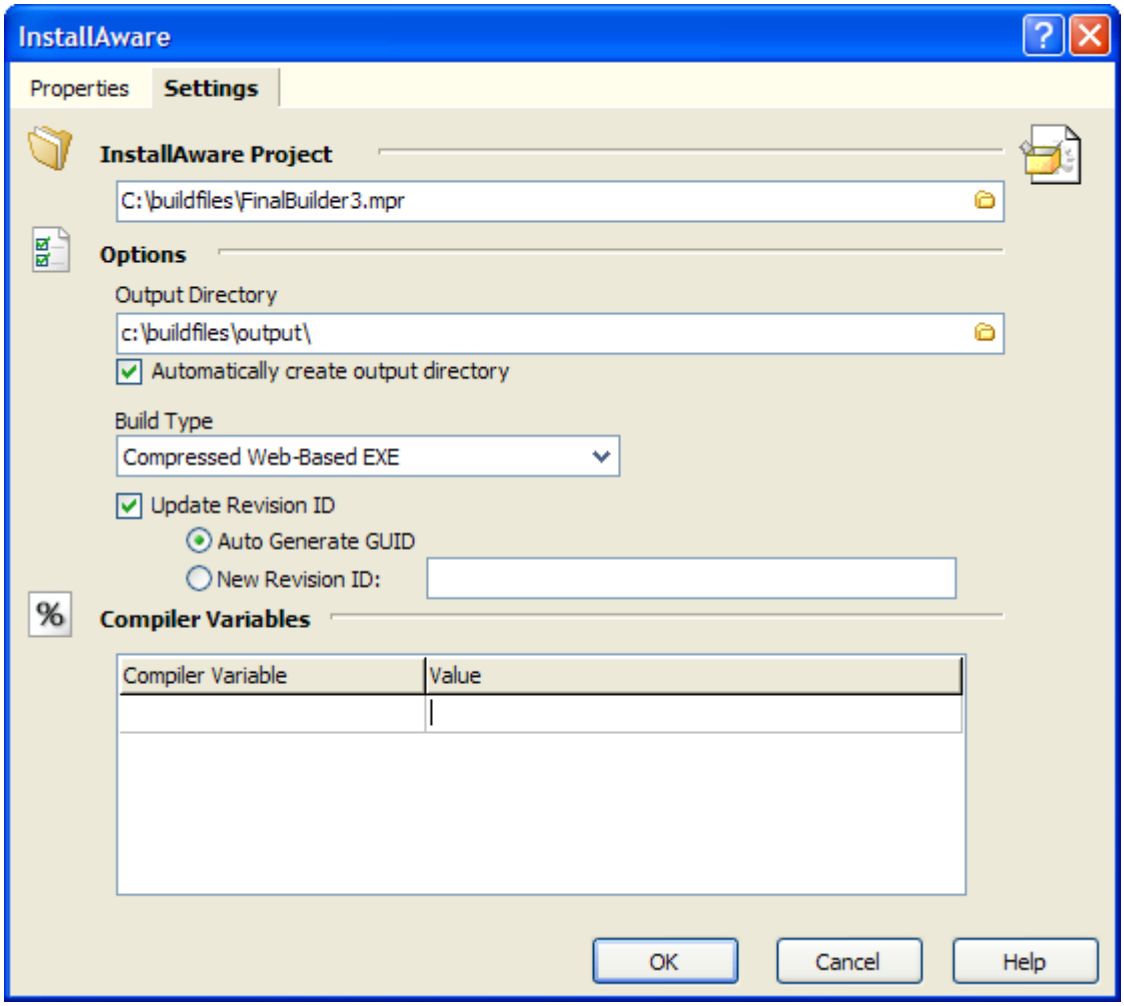
© 2001-2006 VSoft Technologies Pty Ltd

5.13.8 InstallAware

The InstallAware action enables you to automate the process of building InstallAware installations as part of your build process.

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

© 2001-2006 VSoft Technologies Pty Ltd

Specify the new Revision ID that you want this version to have.

© 2001-2006 VSoft Technologies Pty Ltd

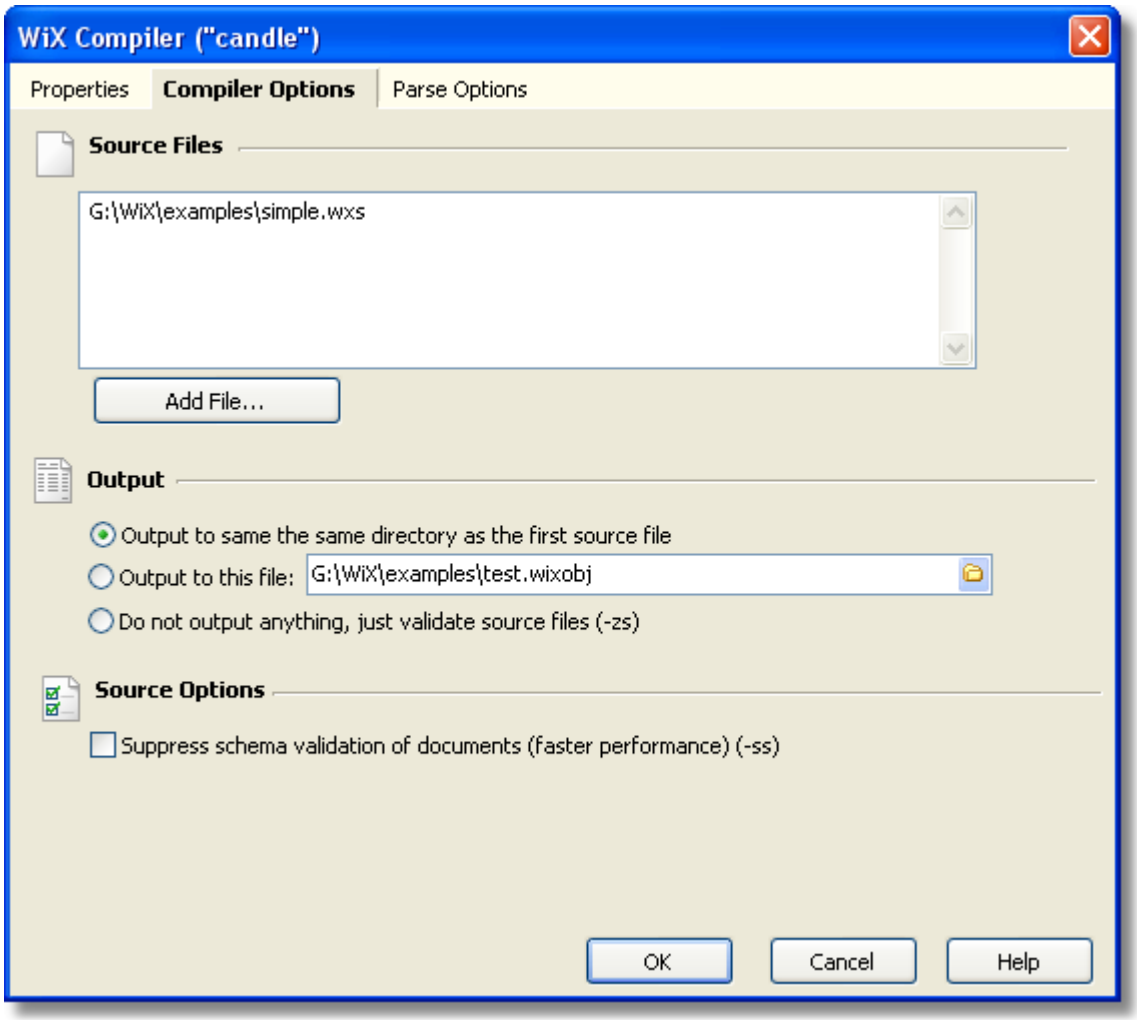
5.13.9 WiX (Windows Installer XML Toolkit)

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.

5.13.9.1 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

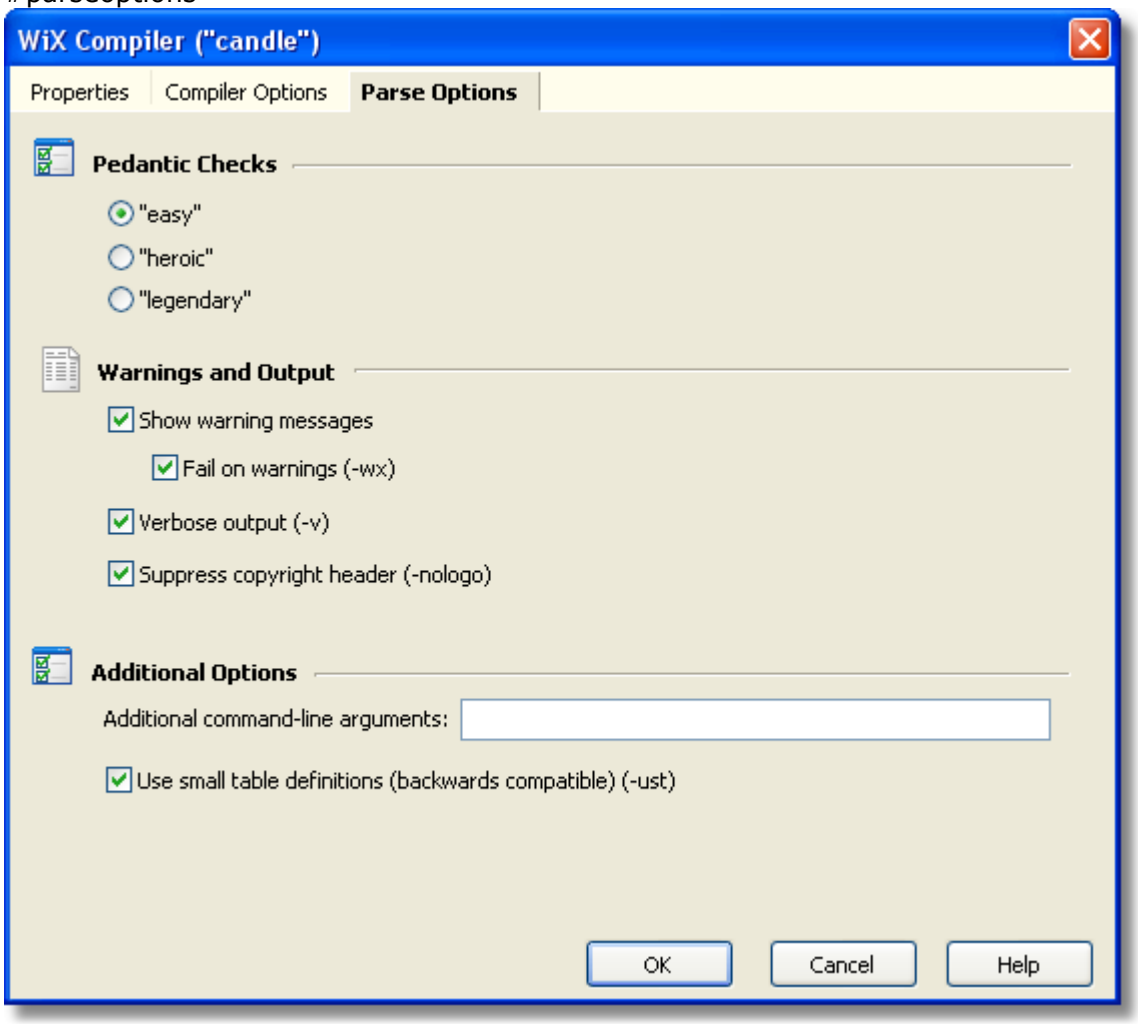
Specify one or more .wxs files to compile. For information on the .wxs format, see the 'doc' directory of your WiX installation.

Output

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

Verbose Output

More progress data will be logged (useful for debugging.)

Suppress copyright header

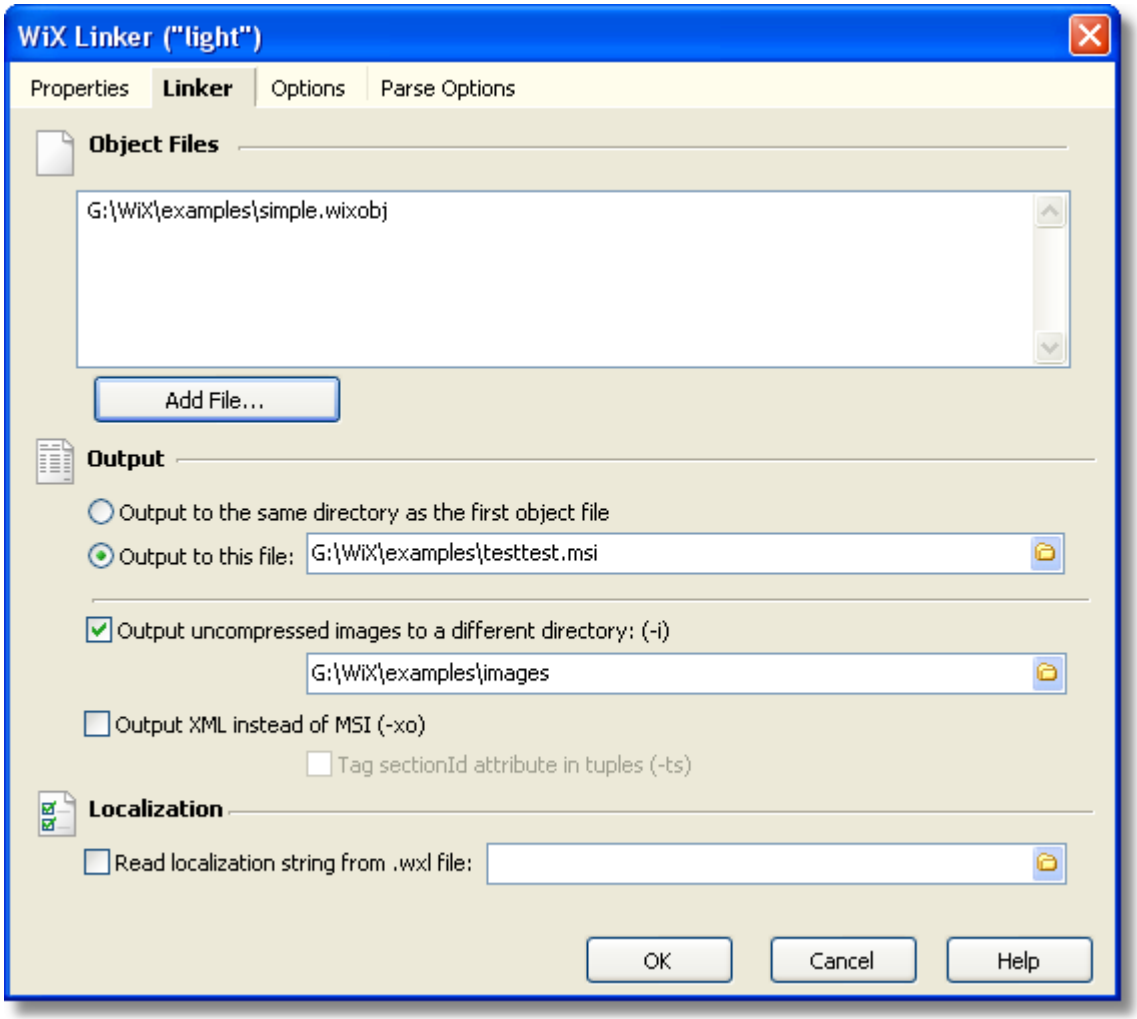
The WiX copyright header will not be displayed. Useful for a smaller log size.

Additional Command-Line Arguments

Type any additional command line arguments that you want to pass to WiX.

5.13.9.2 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.

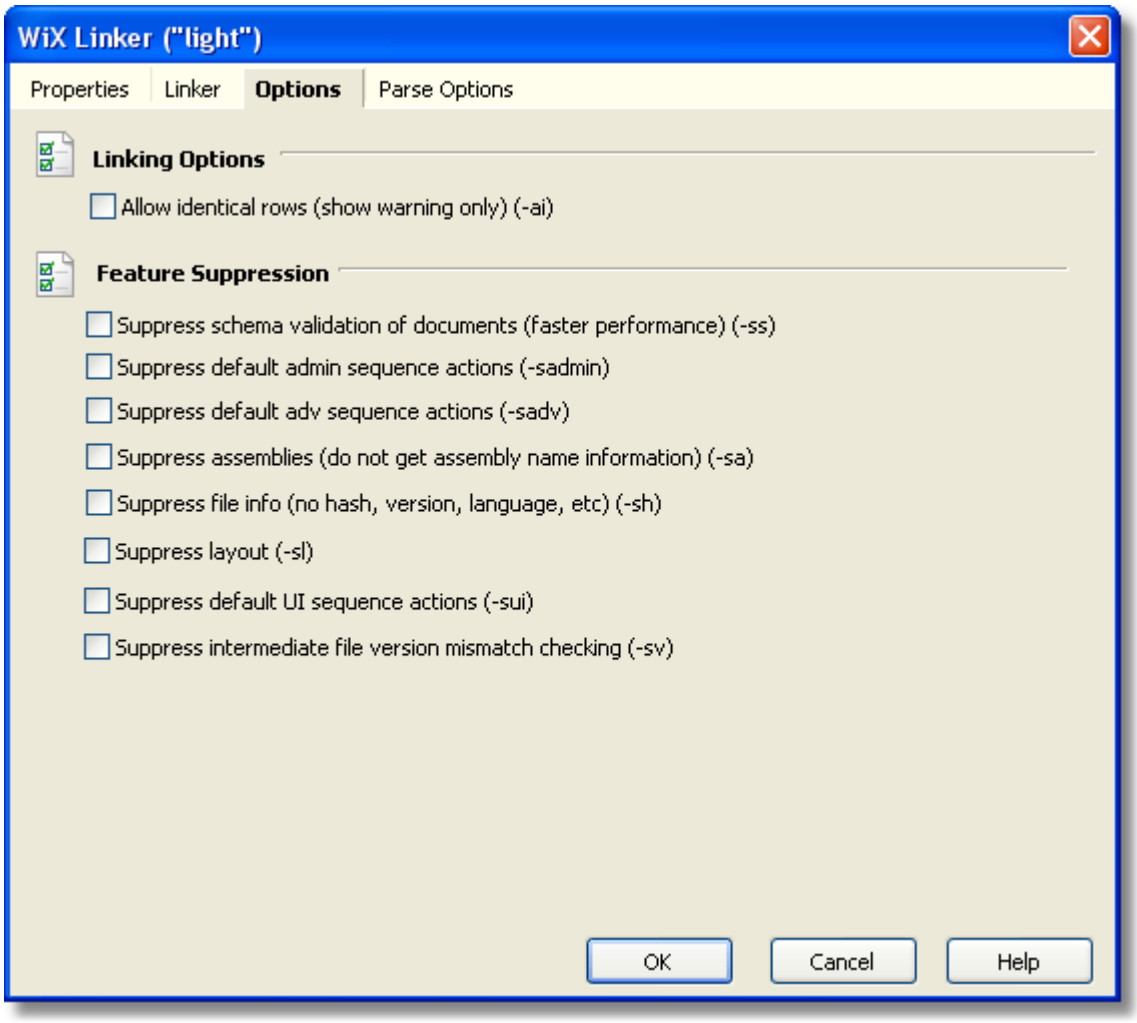
© 2001-2006 VSoft Technologies Pty Ltd

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.



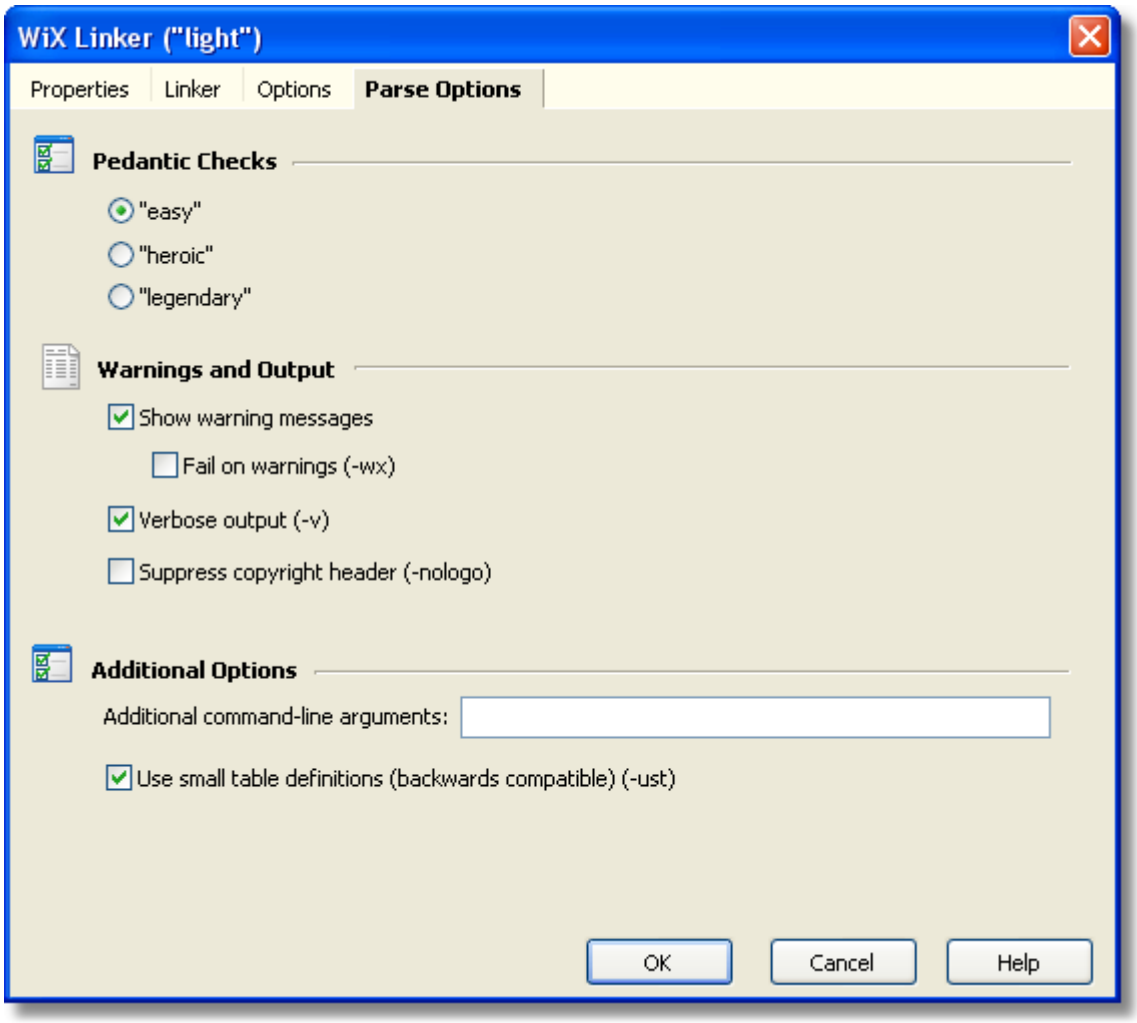
Allow Identical Rows

Normally the WiX Linker will fail if it detects identical rows. With this option checked it will only emit a warning.

Feature Suppression

© 2001-2006 VSoft Technologies Pty Ltd

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.

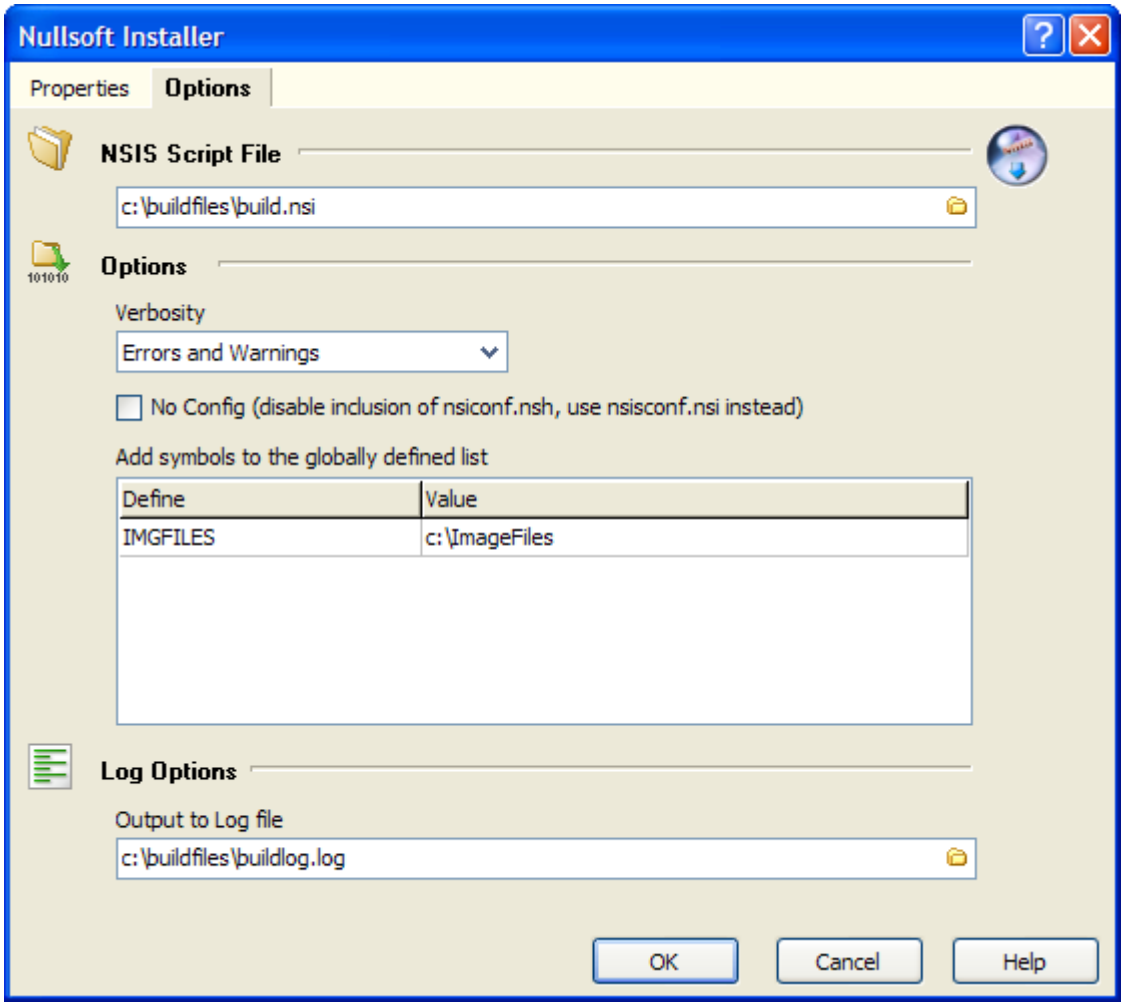
© 2001-2006 VSoft Technologies Pty Ltd

5.13.10 Nullsoft NSIS

The Nullsoft Installer action enables you to automate building Nullsoft Installer projects as part of your build process.

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/

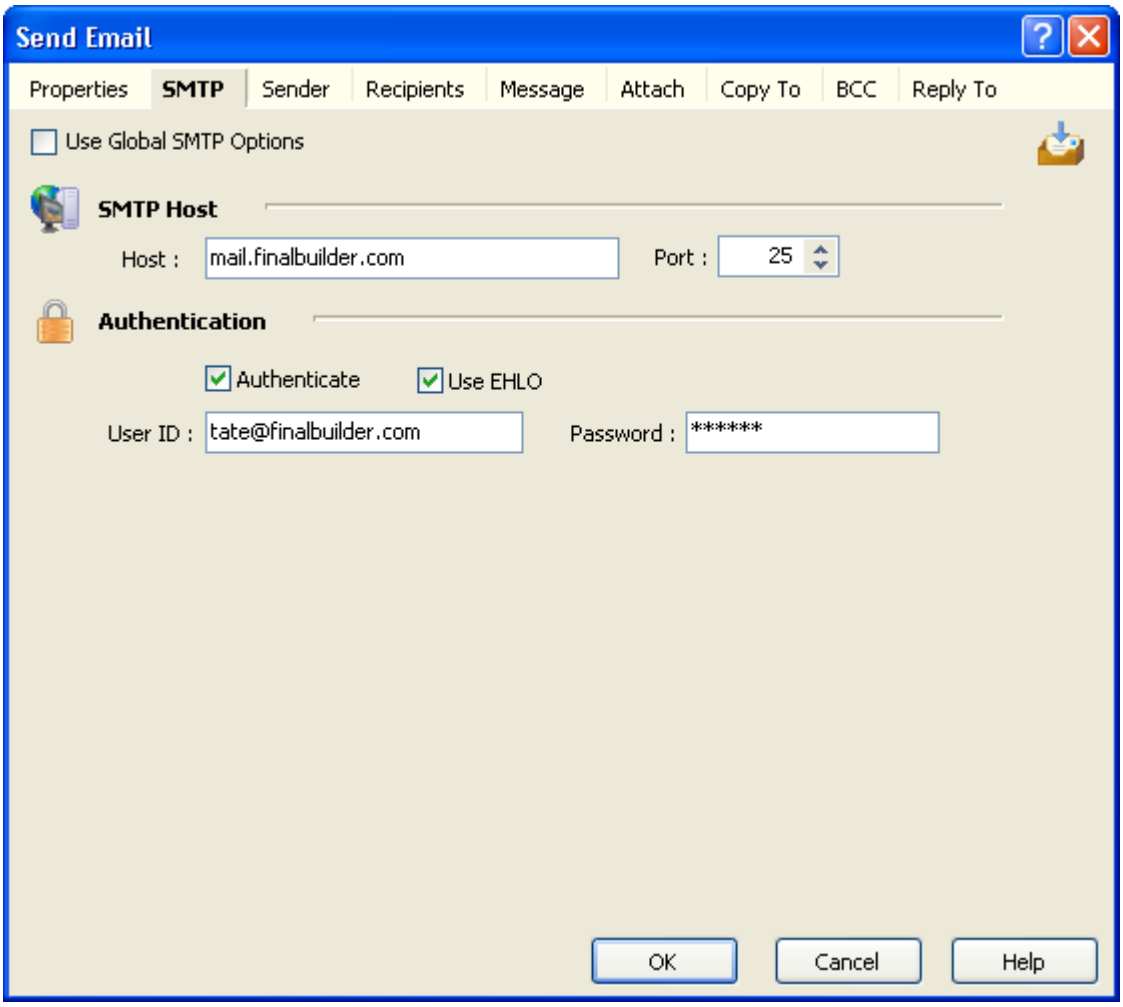


© 2001-2006 VSoft Technologies Pty Ltd

5.14 Internet

5.14.1 Send Email (SMTP)

This action allows your FinalBuilder projects to send email via an SMTP server.



Host : The host name (or IP address) of the SMTP server

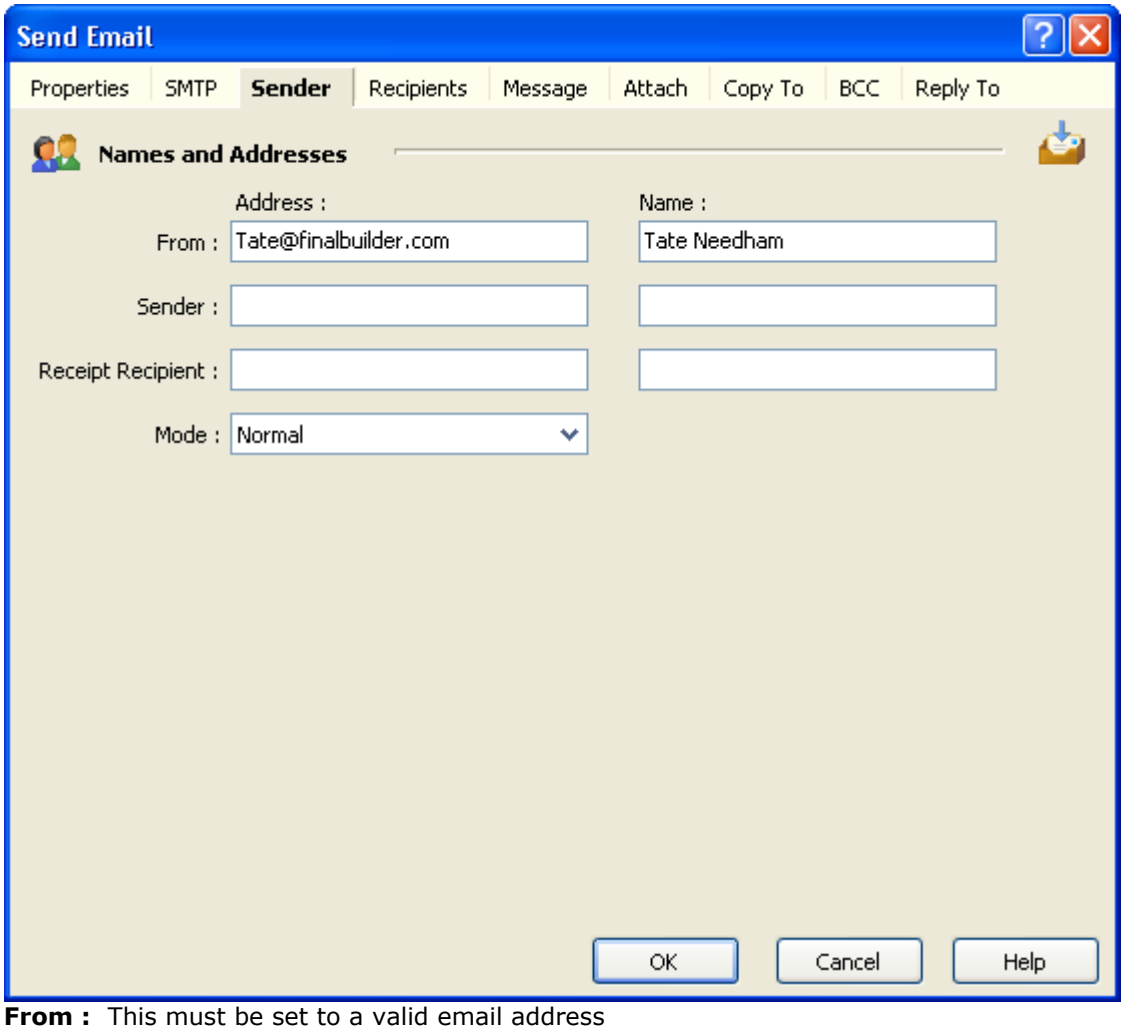
Port : The port on which to connect, defaults to 25

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.

© 2001-2006 VSoft Technologies Pty Ltd



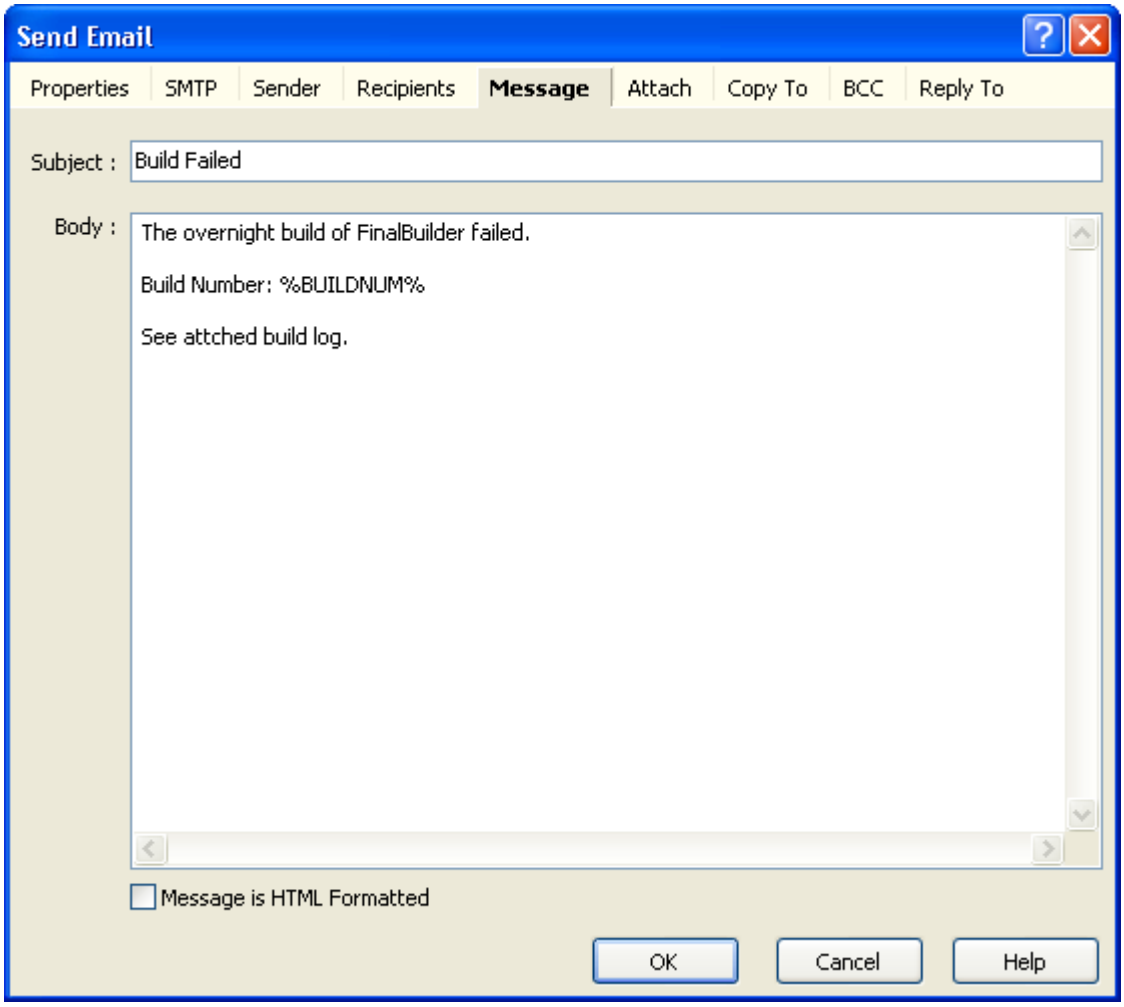
From : This must be set to a valid email address

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.

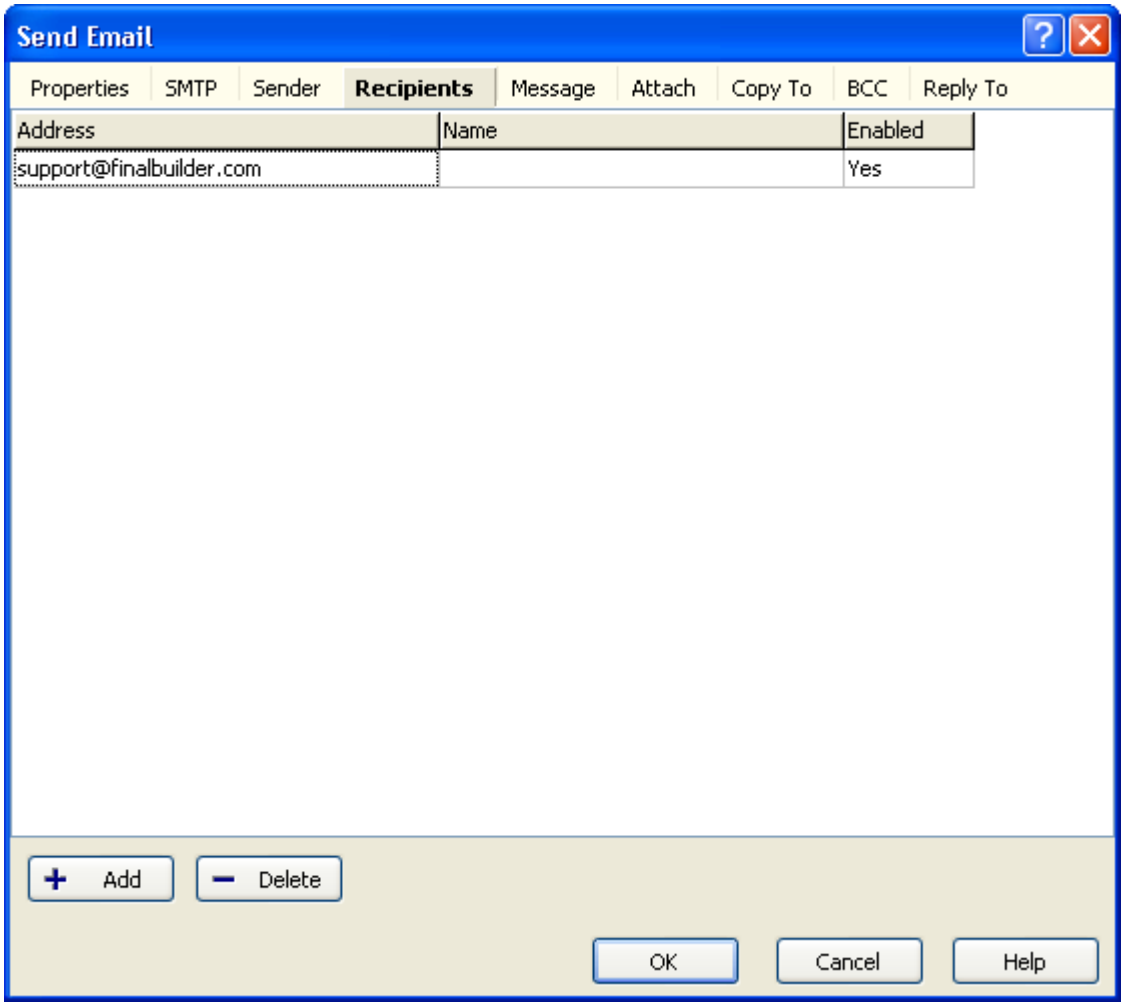
© 2001-2006 VSoft Technologies Pty Ltd



Subject : The email message's subject

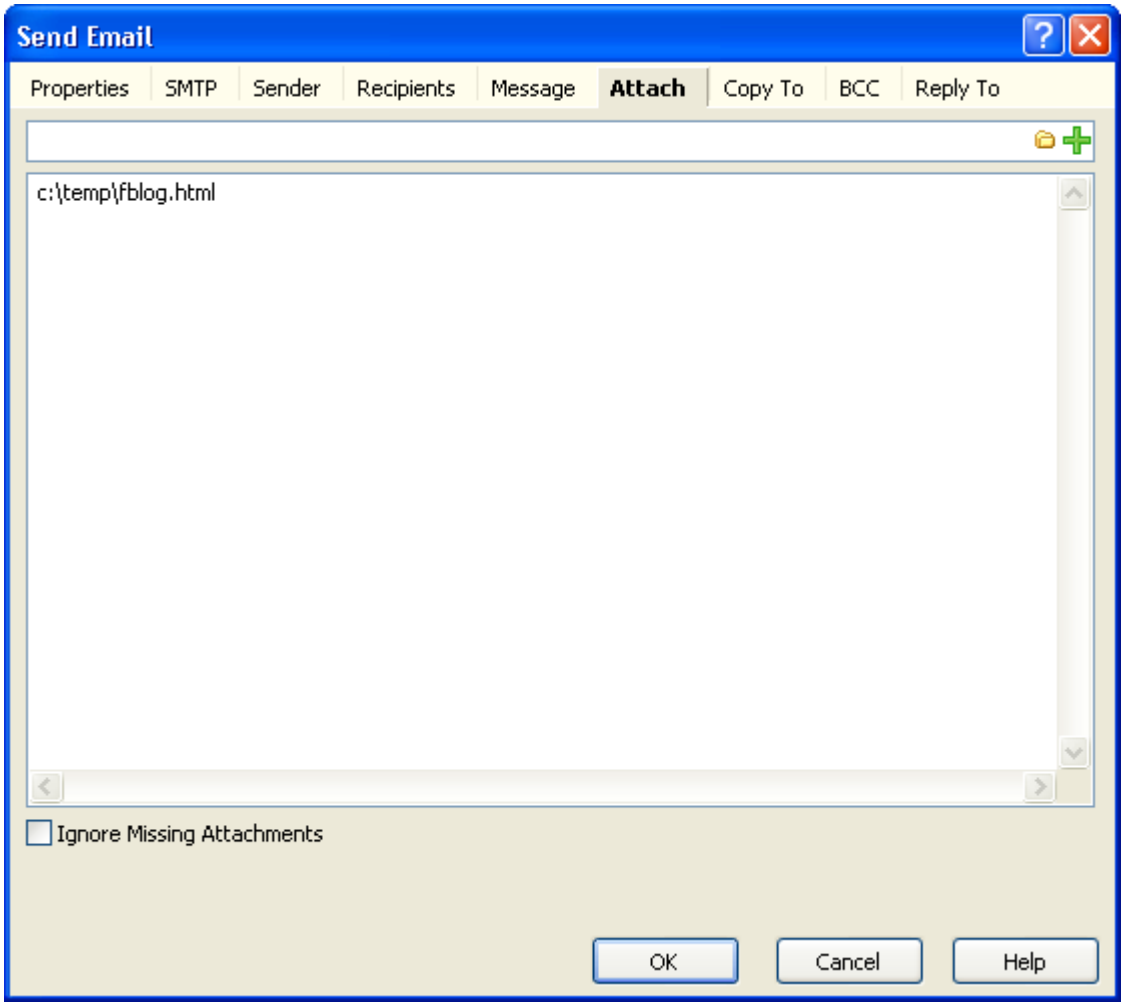
Body : The actual message.

© 2001-2006 VSoft Technologies Pty Ltd



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

© 2001-2006 VSoft Technologies Pty Ltd



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)

© 2001-2006 VSoft Technologies Pty Ltd

```
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

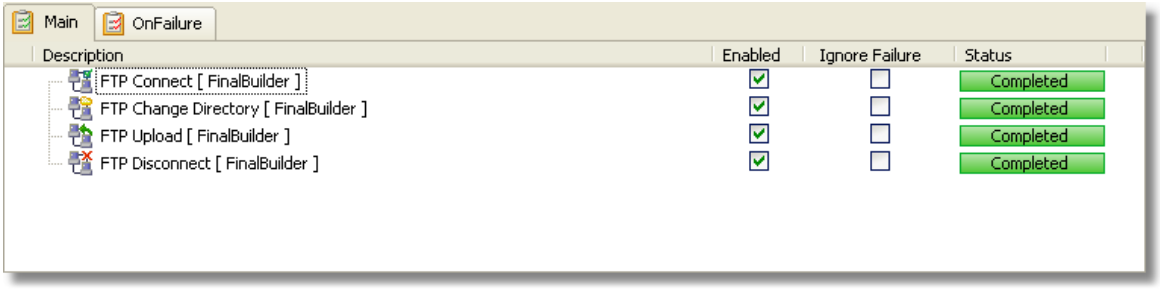
© 2001-2006 VSoft Technologies Pty Ltd

5.14.2 FTP Actions

The FTP Actions allow you to add various FTP commands to your build process.

Use the FTP Connection to define a connection, and then specify this connection in the other FTP actions. Your build process should include an FTP Disconnect action to close the FTP connection.

Example:



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

© 2001-2006 VSoft Technologies Pty Ltd

- 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

© 2001-2006 VSoft Technologies Pty Ltd

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

© 2001-2006 VSoft Technologies Pty Ltd

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 process, 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 amout 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

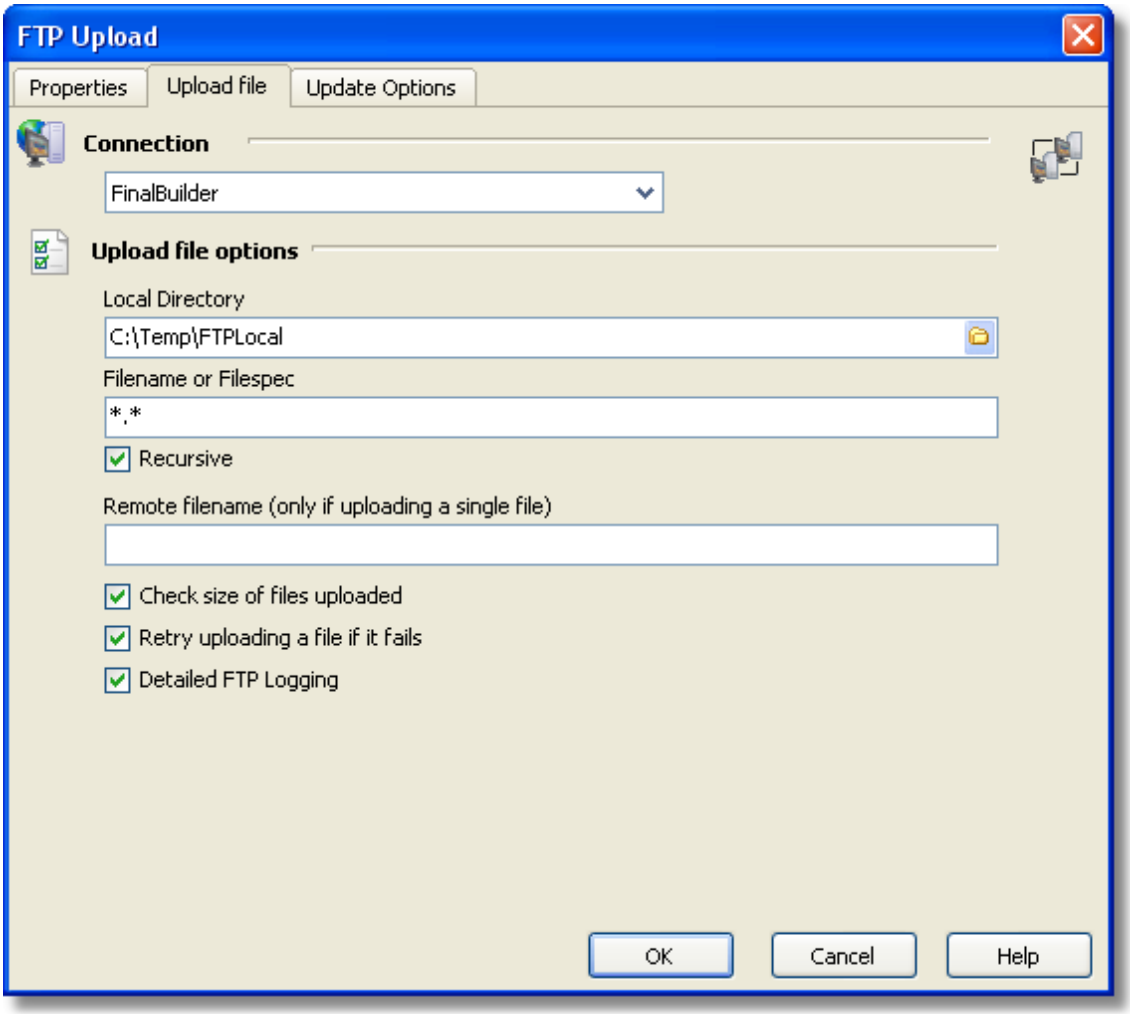
This action is used to disconnect the specified FTP connection. This action should be called after all FTP operations have completed so that the connection is not left open when the build ends.

- Connection Name - select the connection (see FTP Connect action)

5.14.2.1 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

Remote Filename

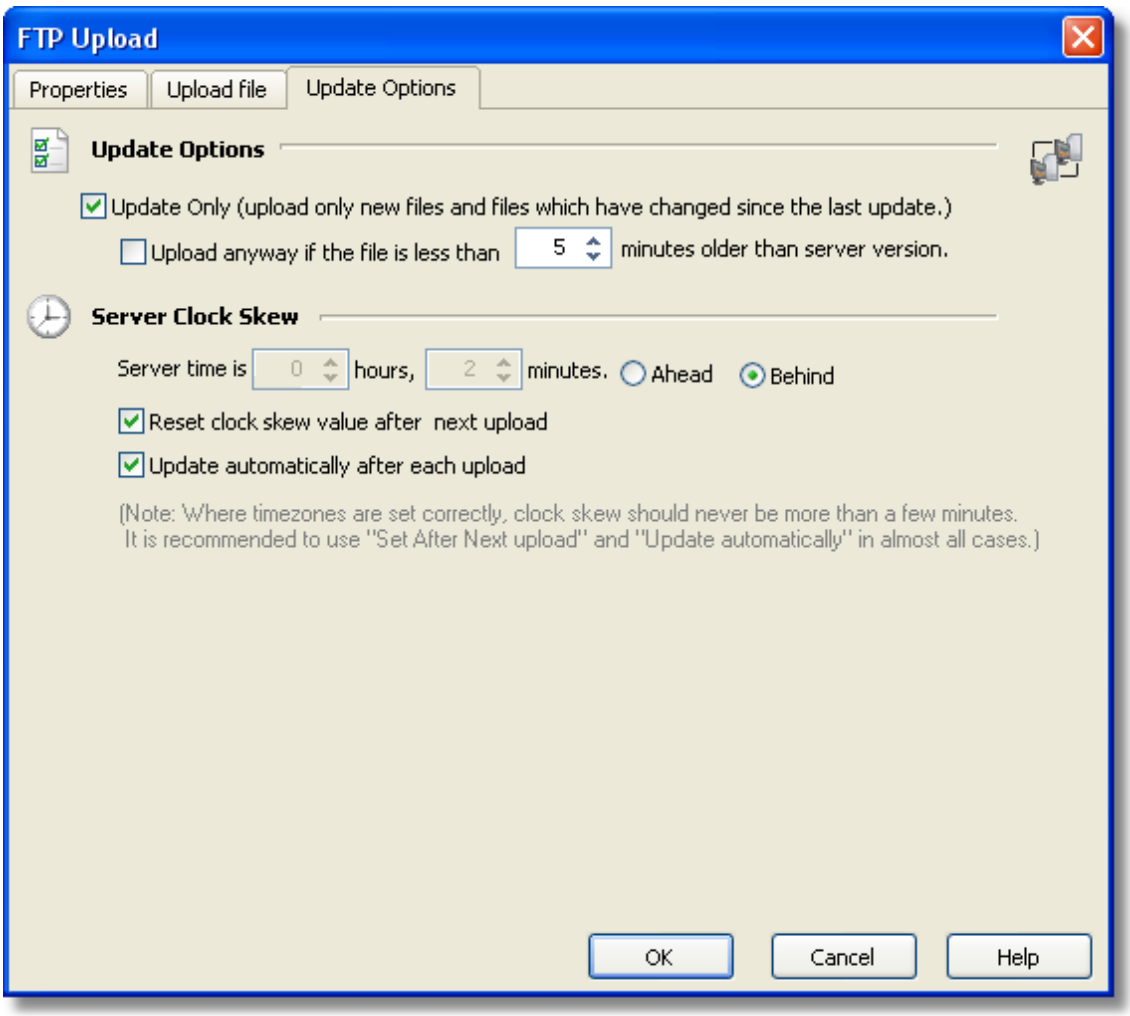
If you are uploading a single file, then you can change the name of the uploaded file on the server.

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.



Update Only

If you check this box, only files which differ from the versions on the server will be

© 2001-2006 VSoft Technologies Pty Ltd

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 (ie 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.

Most FTP Servers report the time in UTC, which is automatically converted to the local timezone. Therefore, clock skew values are normally at most one or two minutes. In cases where either the FTP Server or the local computer have an incorrect or unspecified timezones, or where the FTP server reports non-UTC times, then the skew value 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 reset.)

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.

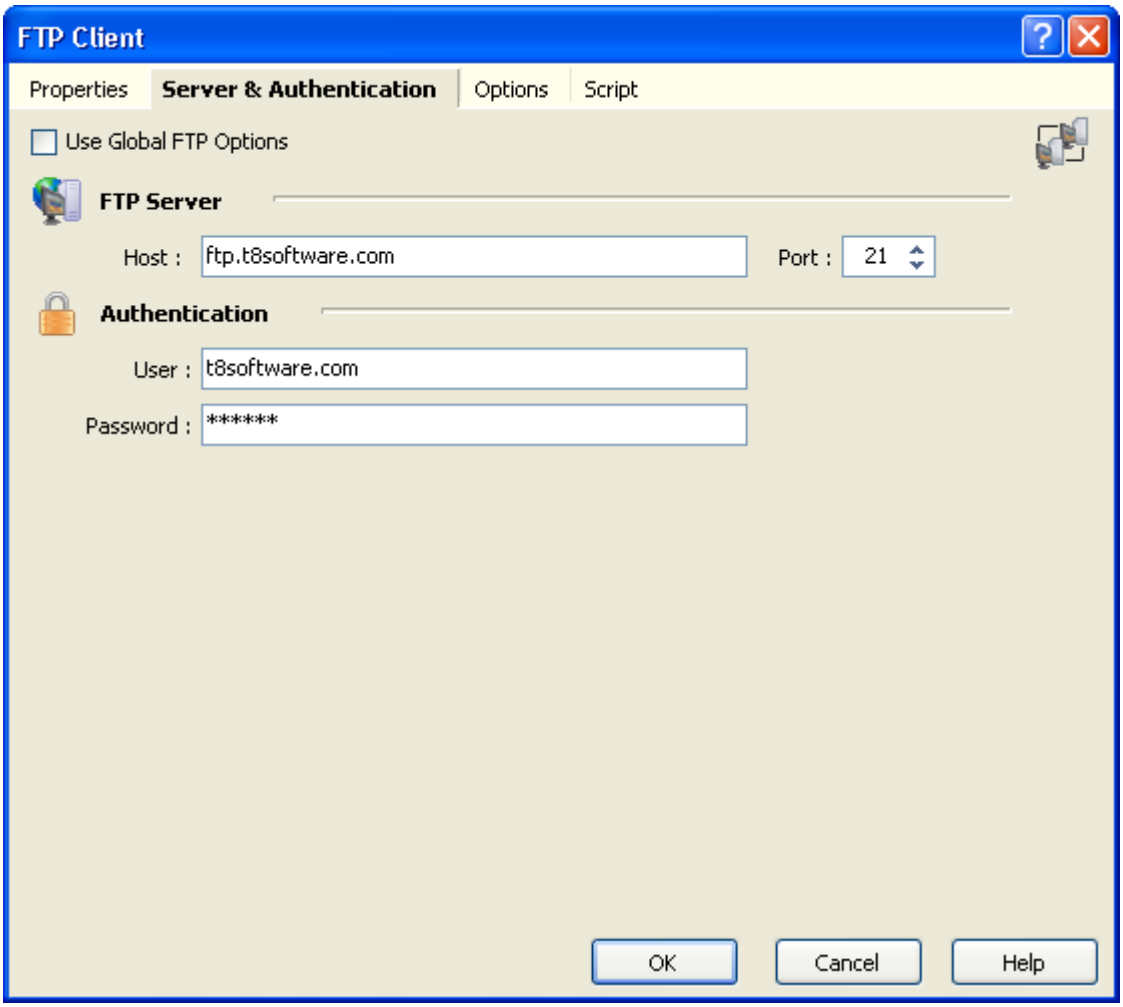
© 2001-2006 VSoft Technologies Pty Ltd

5.14.3 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

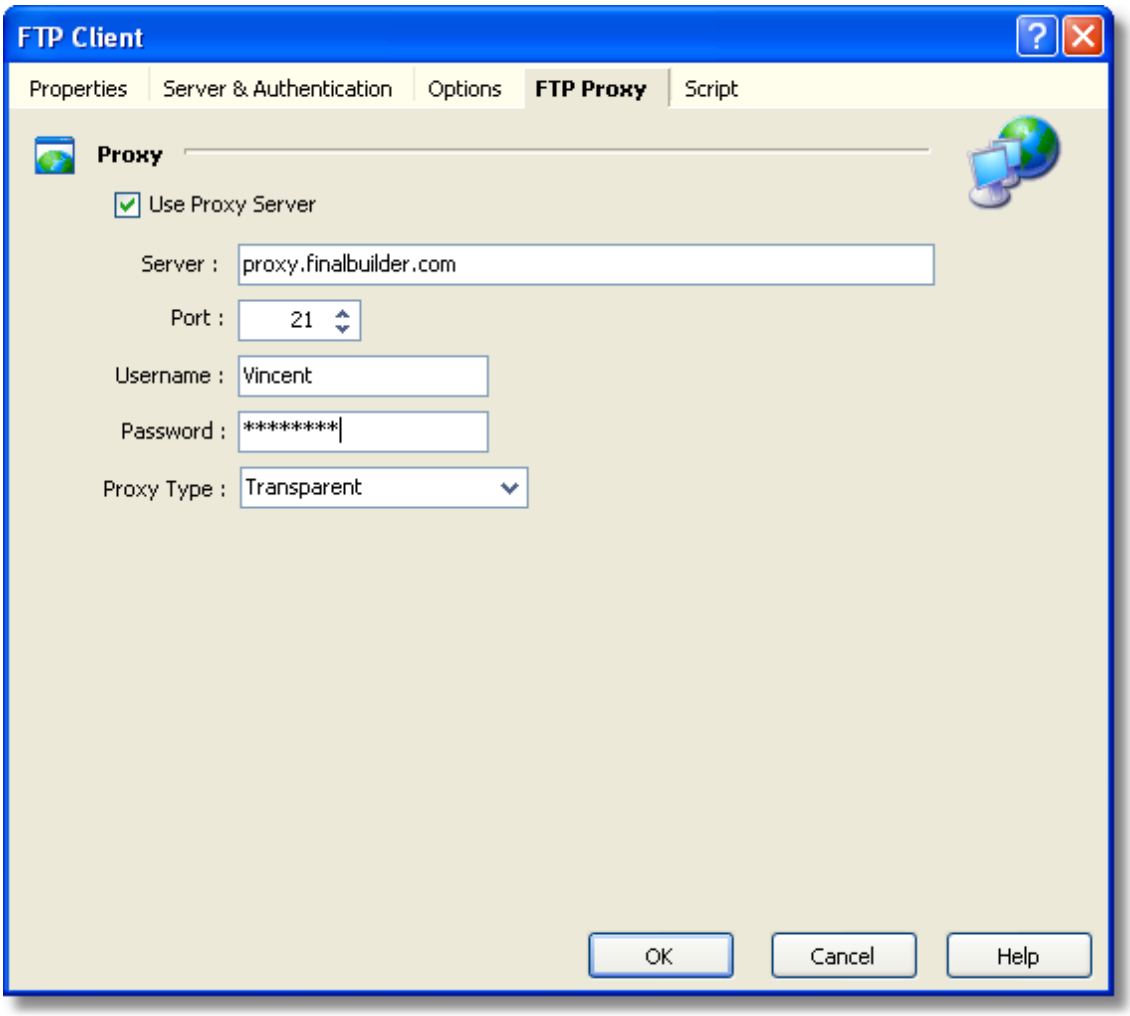
Detailed Logging : Enables Detailed logging.

Binary Transfer Mode : Sets the transfer mode to Binary (the default)

ASCII Transfer Mode : Sets the transfer mode to ASCII

© 2001-2006 VSoft Technologies Pty Ltd

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

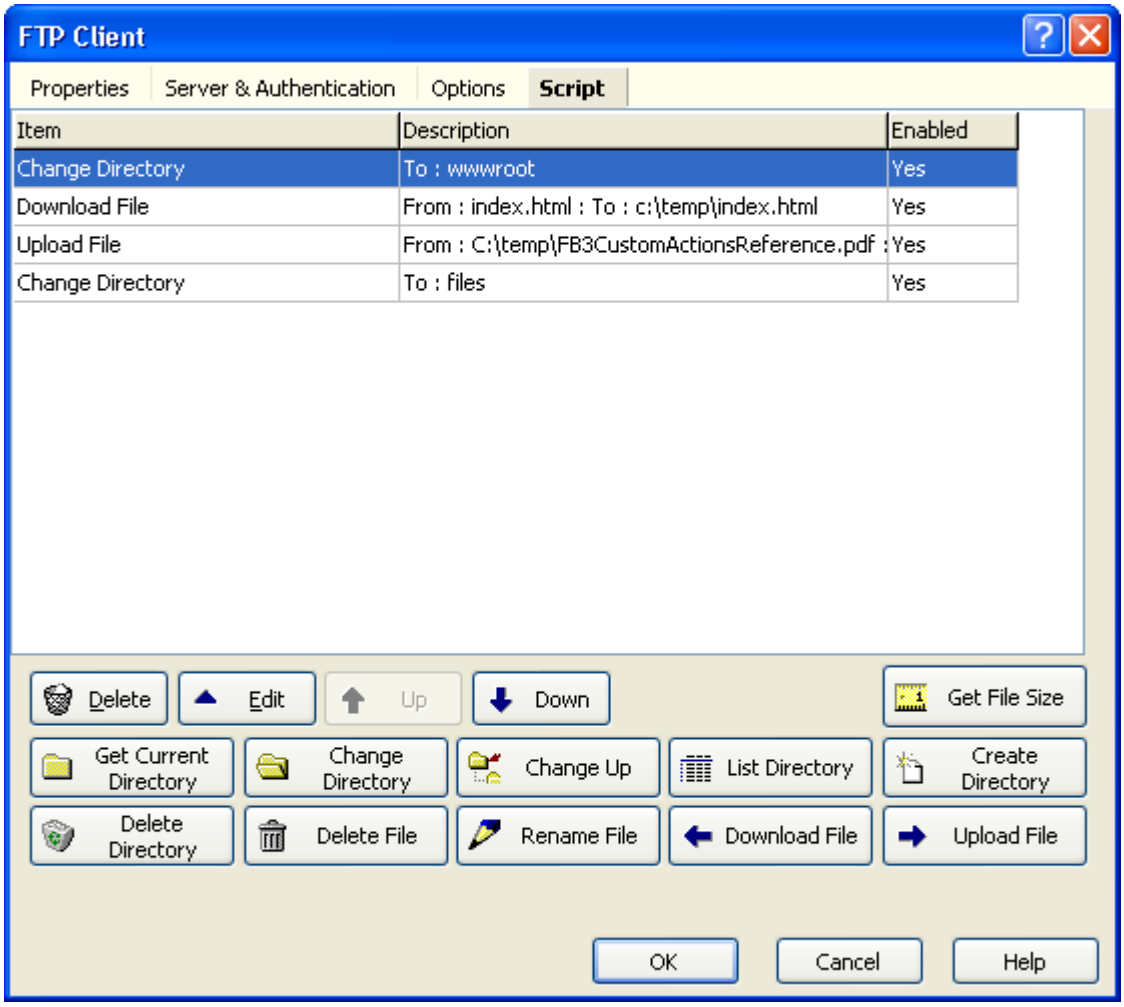
Password : The password for authentication

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

© 2001-2006 VSoft Technologies Pty Ltd



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.

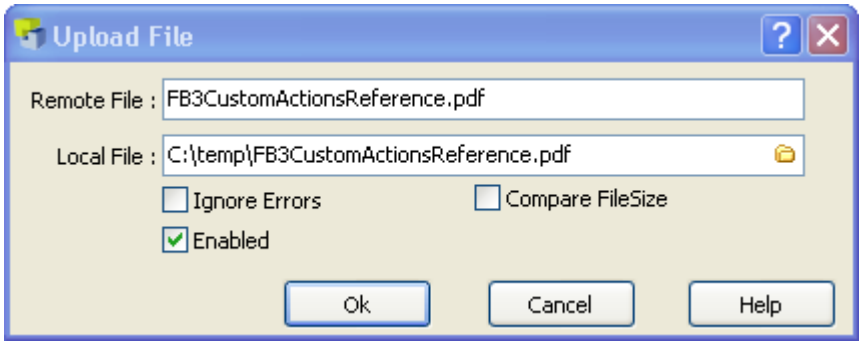
Delete File : Deletes the specified remote File.

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.

© 2001-2006 VSoft Technologies Pty Ltd



Scripting Info

The Action properties available are :

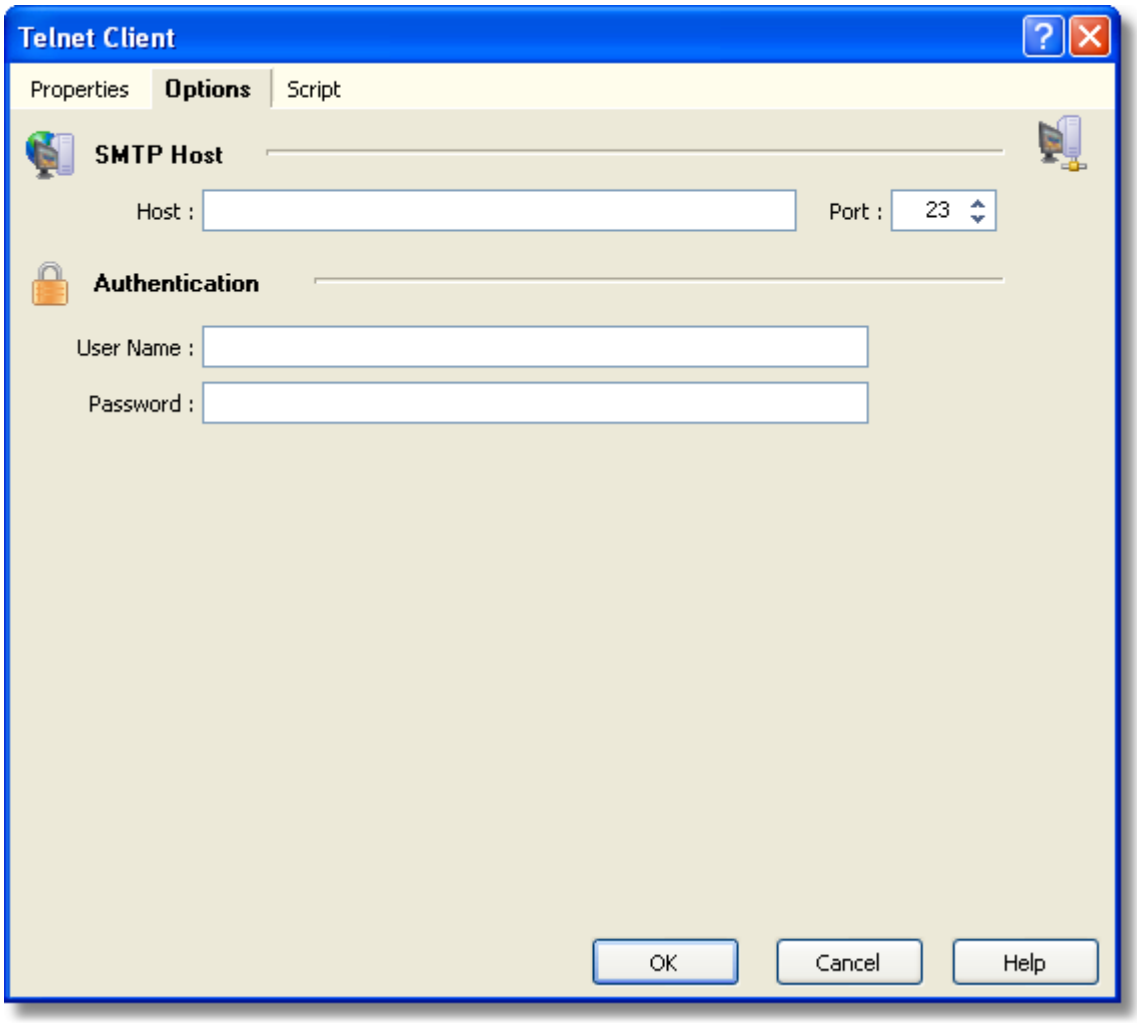
property Host : WideString
property Port : integer
property UserID : WideString
property Password : WideString
property Passive : WordBool
property CurrentDir : WideString
property DetailedLogging : WordBool

This action uses the Open Source Indy Components, for more information see the Indy web site : http://www.nevrona.com/Indy

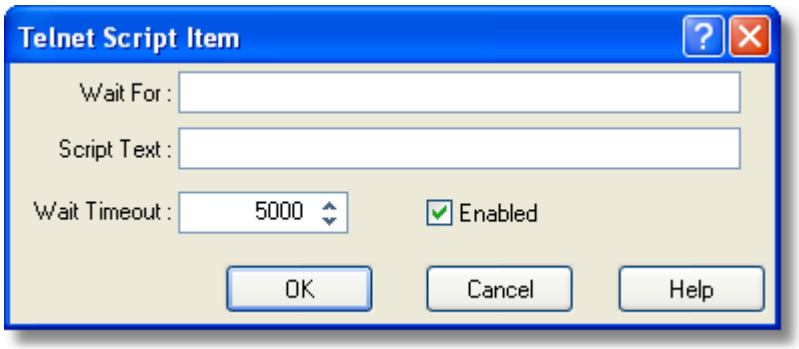
© 2001-2006 VSoft Technologies Pty Ltd

5.14.4 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.

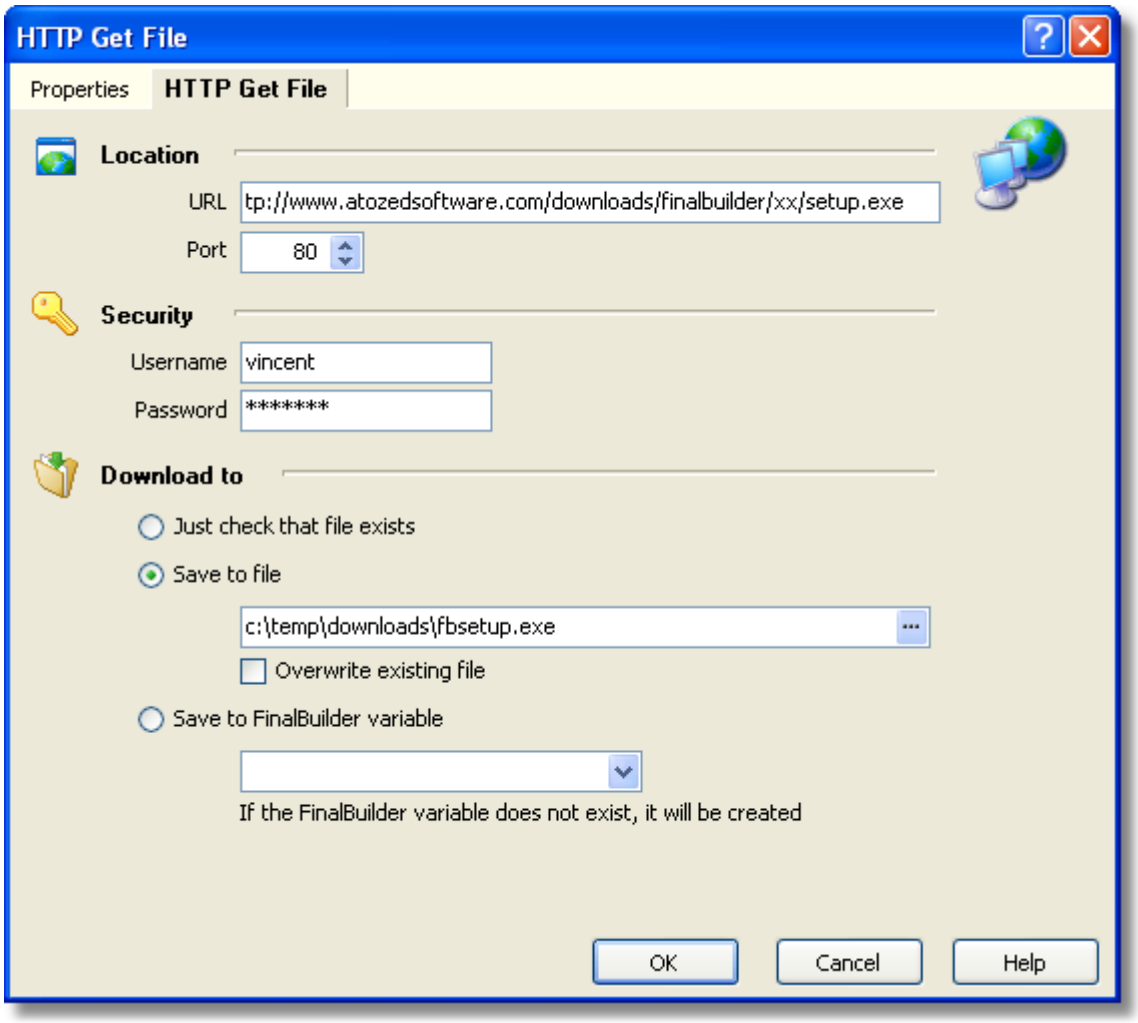


To send the UserName set the ScriptText field to @USERNAME
To send the Password set the ScriptText field to @PASSWORD

© 2001-2006 VSoft Technologies Pty Ltd

5.14.5 HTTP Get Action

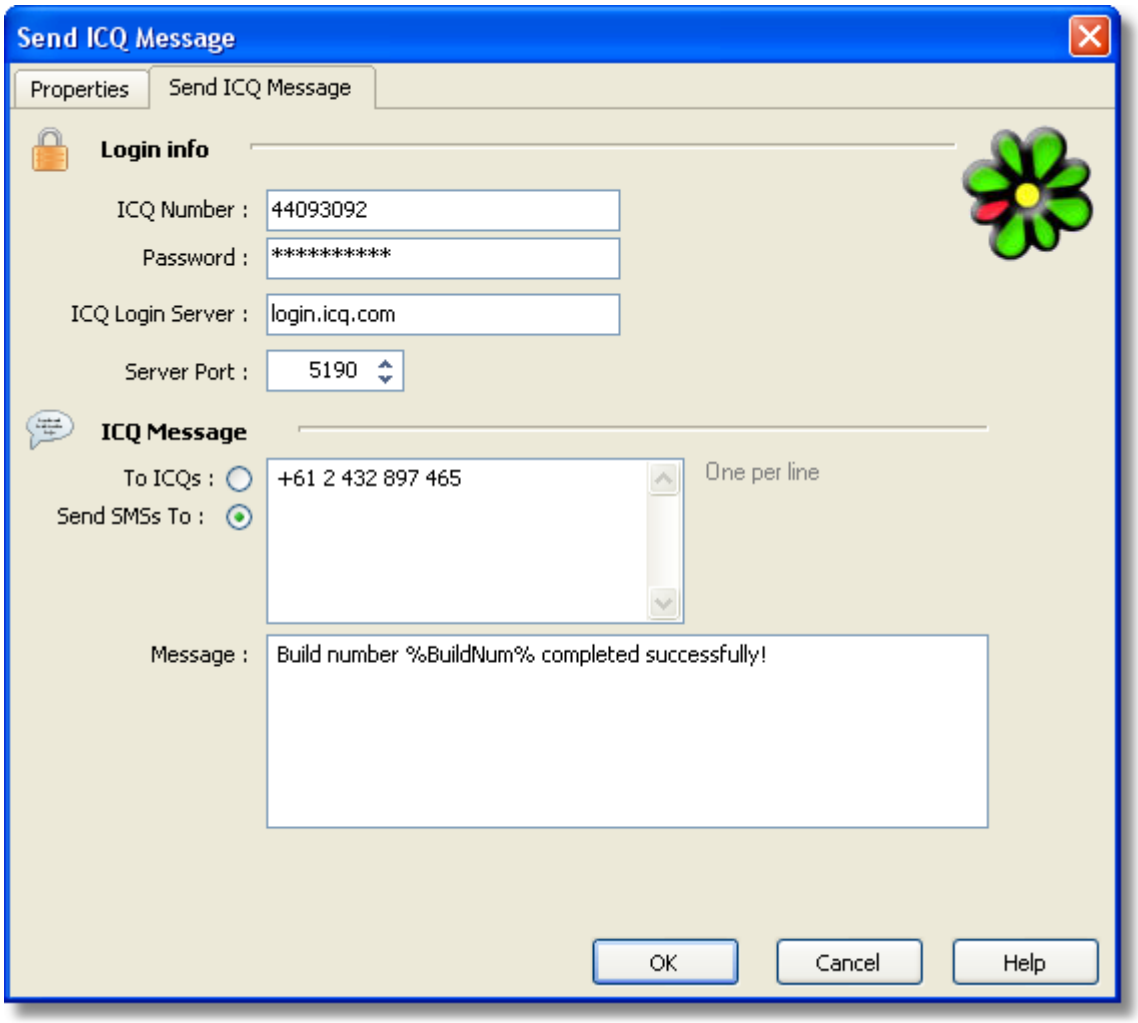
This action enables you to download a file using the http protocol.



© 2001-2006 VSoft Technologies Pty Ltd

5.14.6 Send ICQ Message Action

This action enables you to send ICQ messages.



Login info

Specify an existing ICQ number, password, an ICQ server to log in to and a port to connect to.

ICQ Message

Messages can be sent to either ICQ numbers of SMS phone numbers (for SMS, the target phone number will need to be connected to an ICQ-supported network.) Put each new phone or ICQ number on a new line.

Message

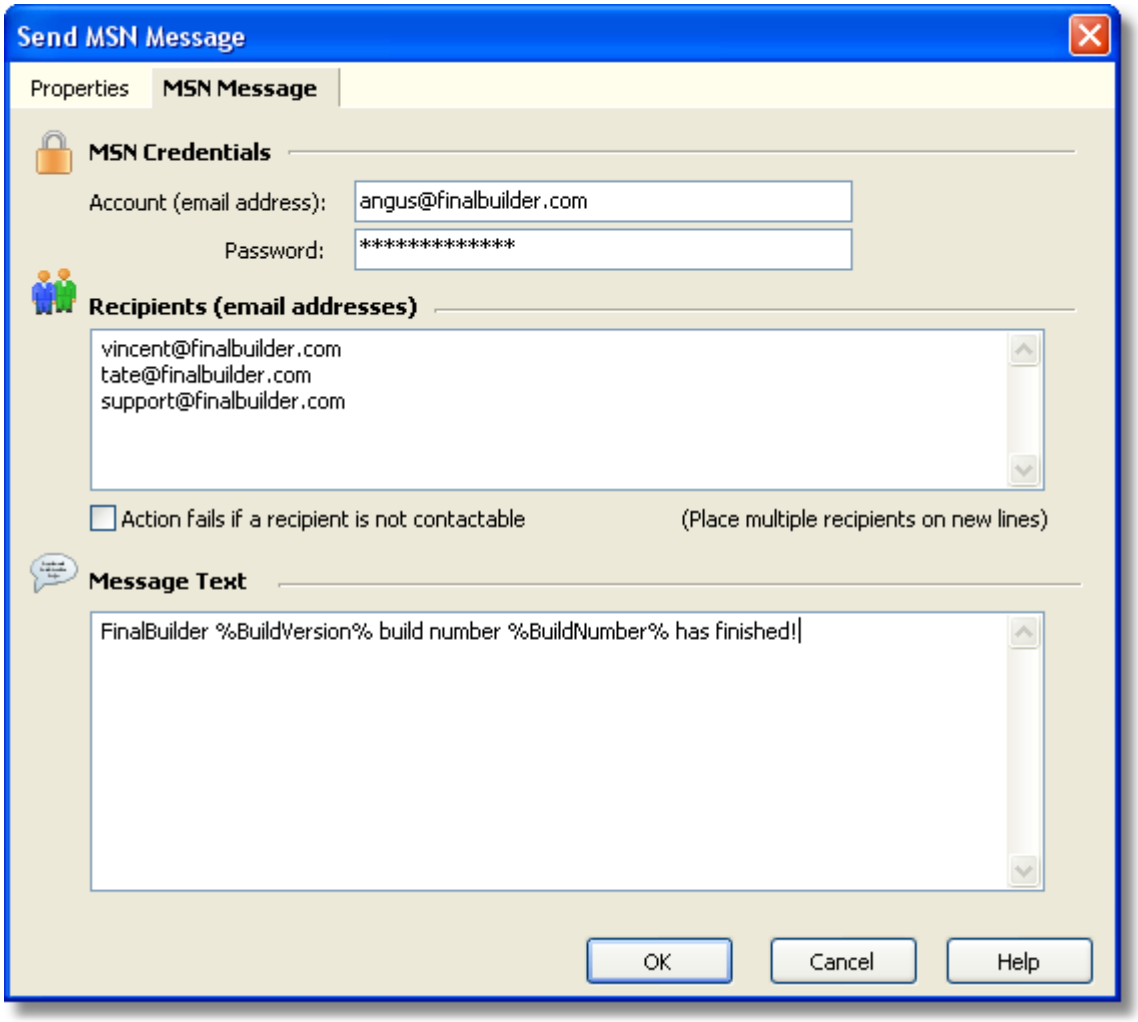
The text of the ICQ message.

© 2001-2006 VSoft Technologies Pty Ltd

5.14.7 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 process.

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.

© 2001-2006 VSoft Technologies Pty Ltd

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

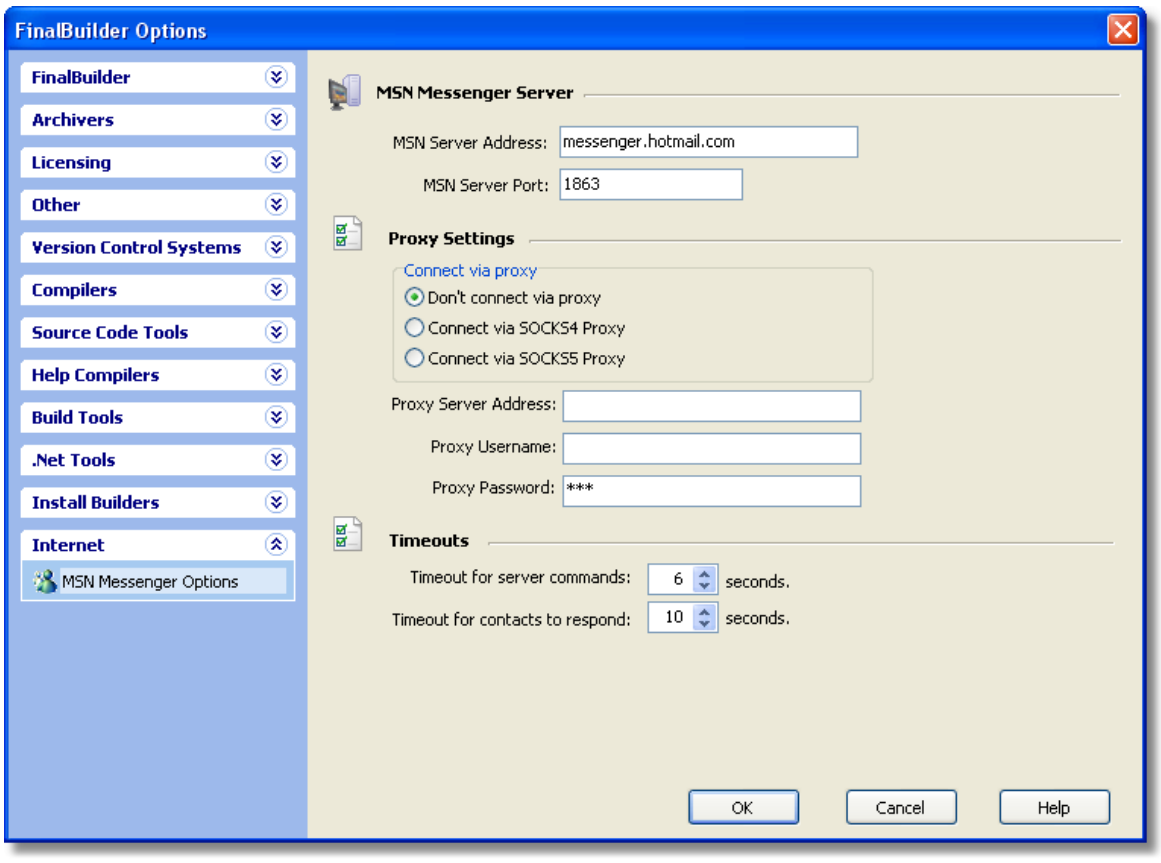
The action can be set to fail if one or more contacts could not be reached (either because they are offline or because that email address is not registered with MSN.)

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



© 2001-2006 VSoft Technologies Pty Ltd

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
```

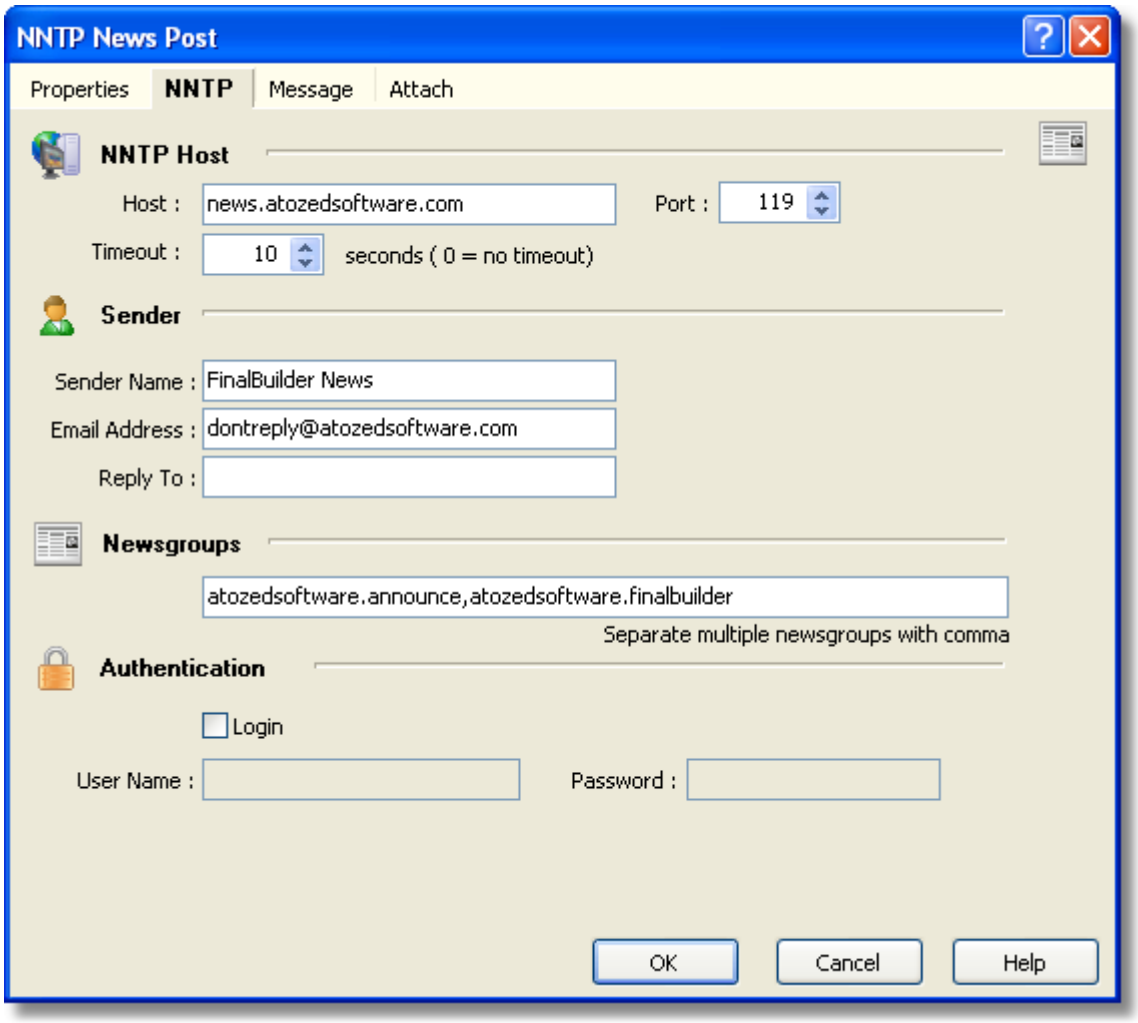
```
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.
```

© 2001-2006 VSoft Technologies Pty Ltd

5.14.8 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.

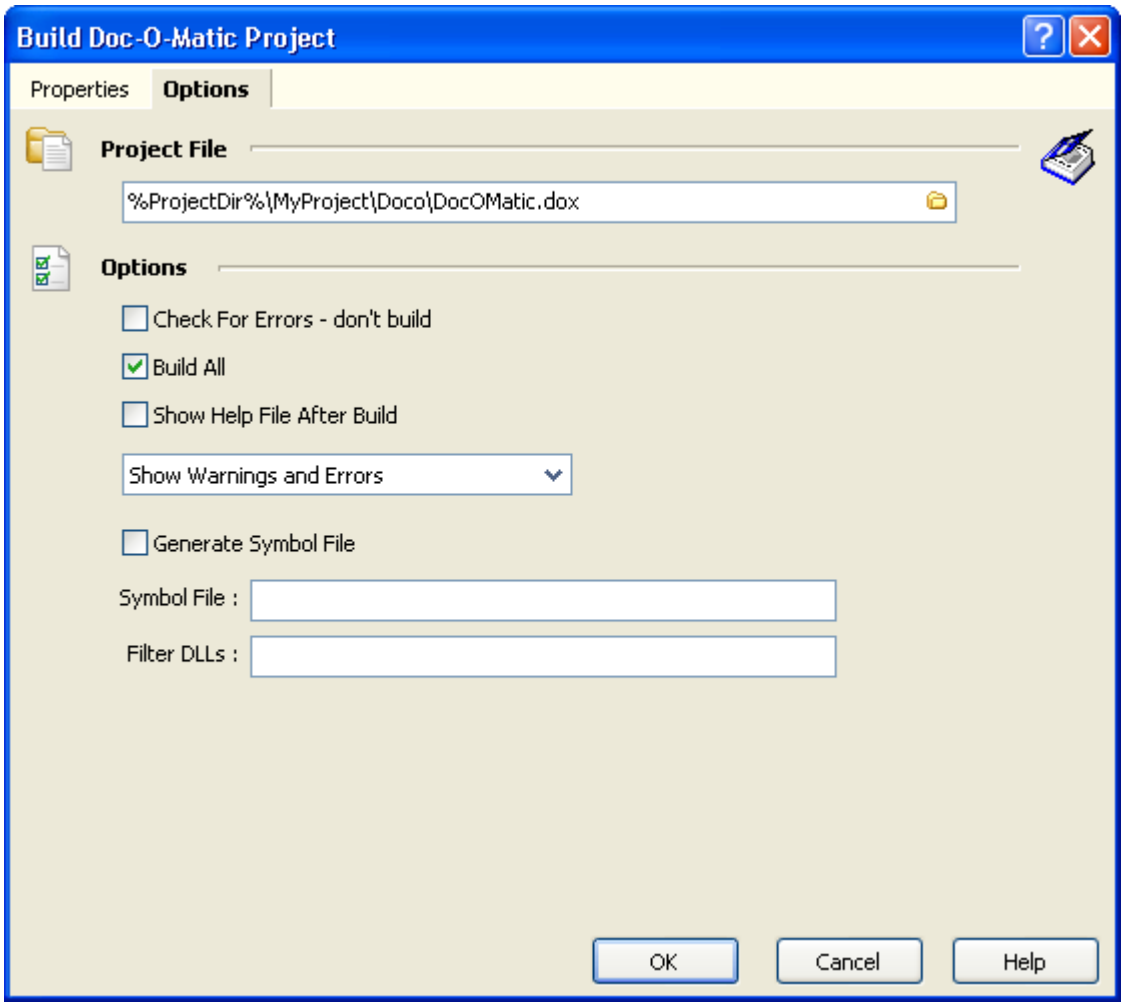


© 2001-2006 VSoft Technologies Pty Ltd

5.15 Help Compilers

5.15.1 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

© 2001-2006 VSoft Technologies Pty Ltd

property WarningLevel : integer // valid values are : wlShowAll, wlShowHintsWarningsAndErrors, wlShowWarningsAndErrors , wlShowErrorsOnly

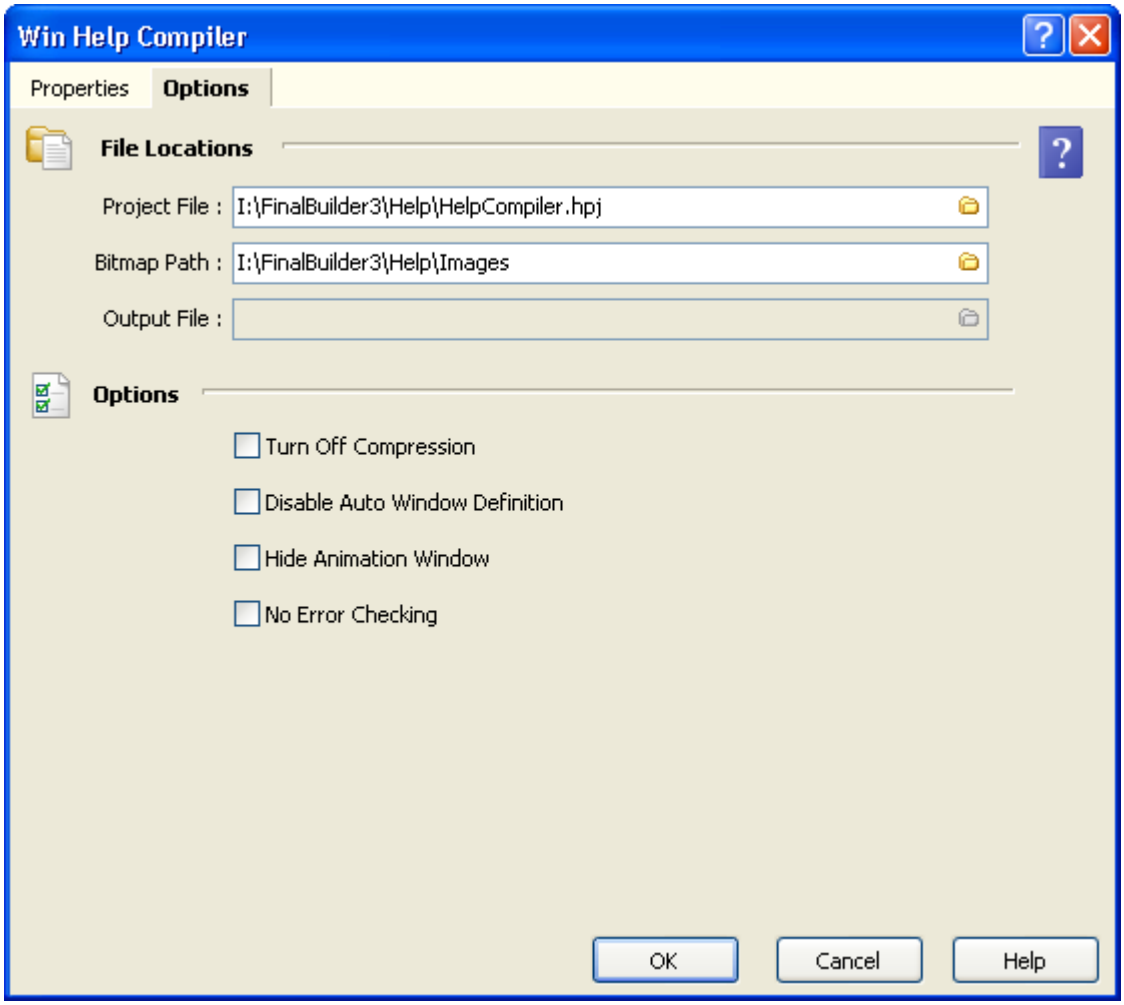
property ProjectFile : WideString
property ShowAfterBuild : WordBool
property SymbolFile : WideString
property GenerateSymbolFile : WordBool

Doc-O-Matic can be found at http://www.toolsfactory.com/

© 2001-2006 VSoft Technologies Pty Ltd

5.15.2 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 (*.hpj) 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

© 2001-2006 VSoft Technologies Pty Ltd

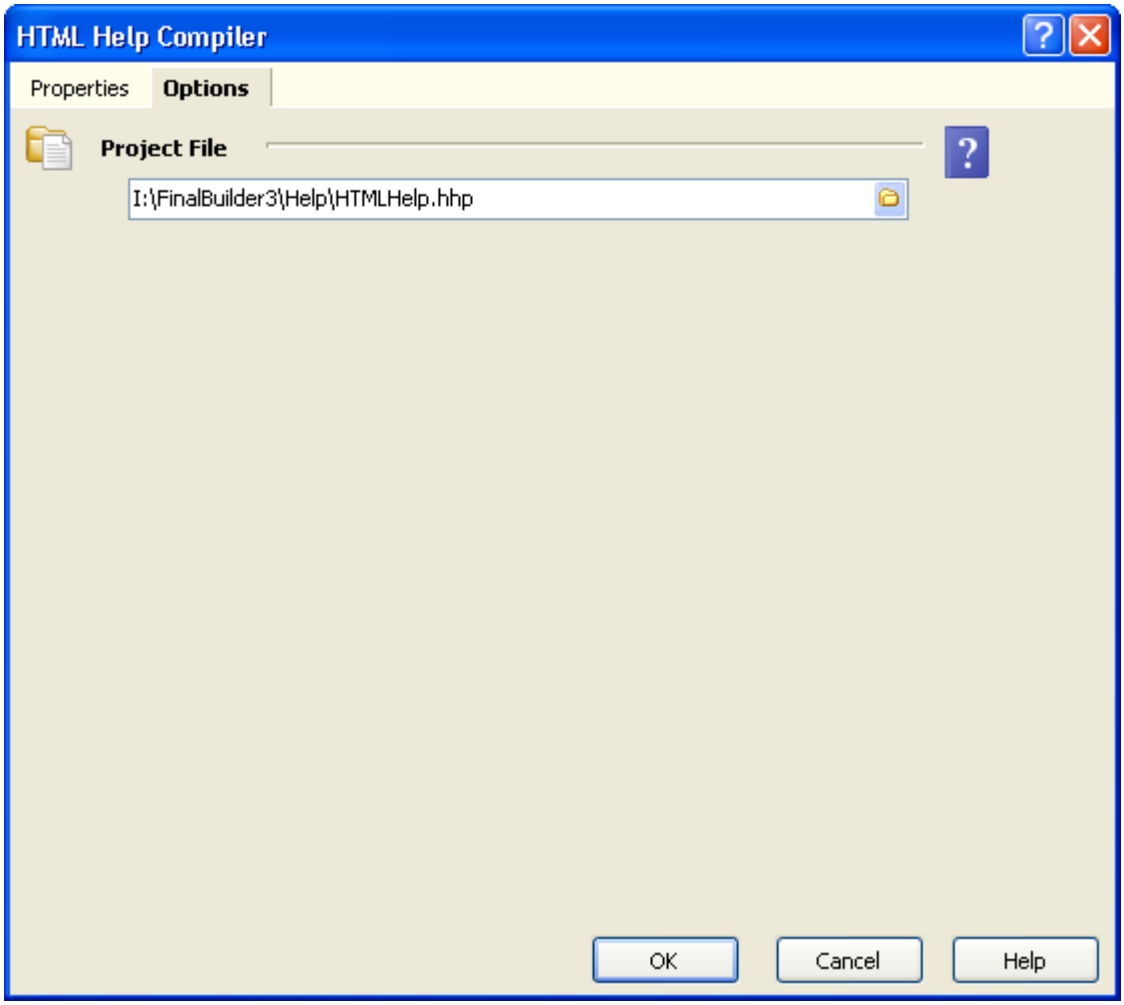
The Action properties available are :

property ProjectFile : WideString
property BitmapPath : WideString
property TurnOffCompression : WordBool
property HideAnimationWin : WordBool
property DisableAutoWinDef : WordBool
property NoErrorCheck : WordBool
property OutputFile : WideString

© 2001-2006 VSoft Technologies Pty Ltd

5.15.3 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 :

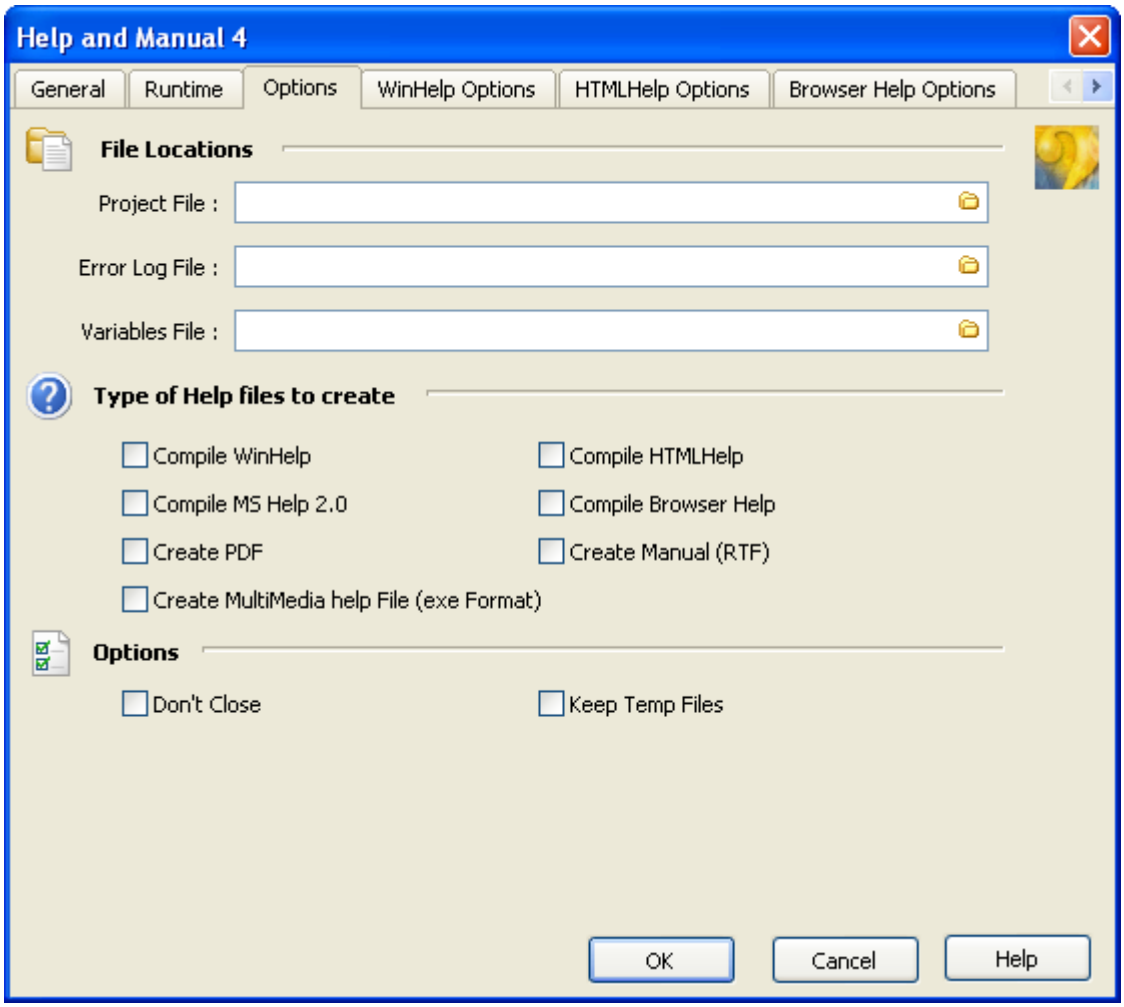
property ProjectFile : WideString

© 2001-2006 VSoft Technologies Pty Ltd

5.15.4 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 then one type of help file can be built. The specific options for each Help file type can be found under the relevant options tab.

© 2001-2006 VSoft Technologies Pty Ltd

Don't Close
The Help & Manual program will be left running after the action finishes.

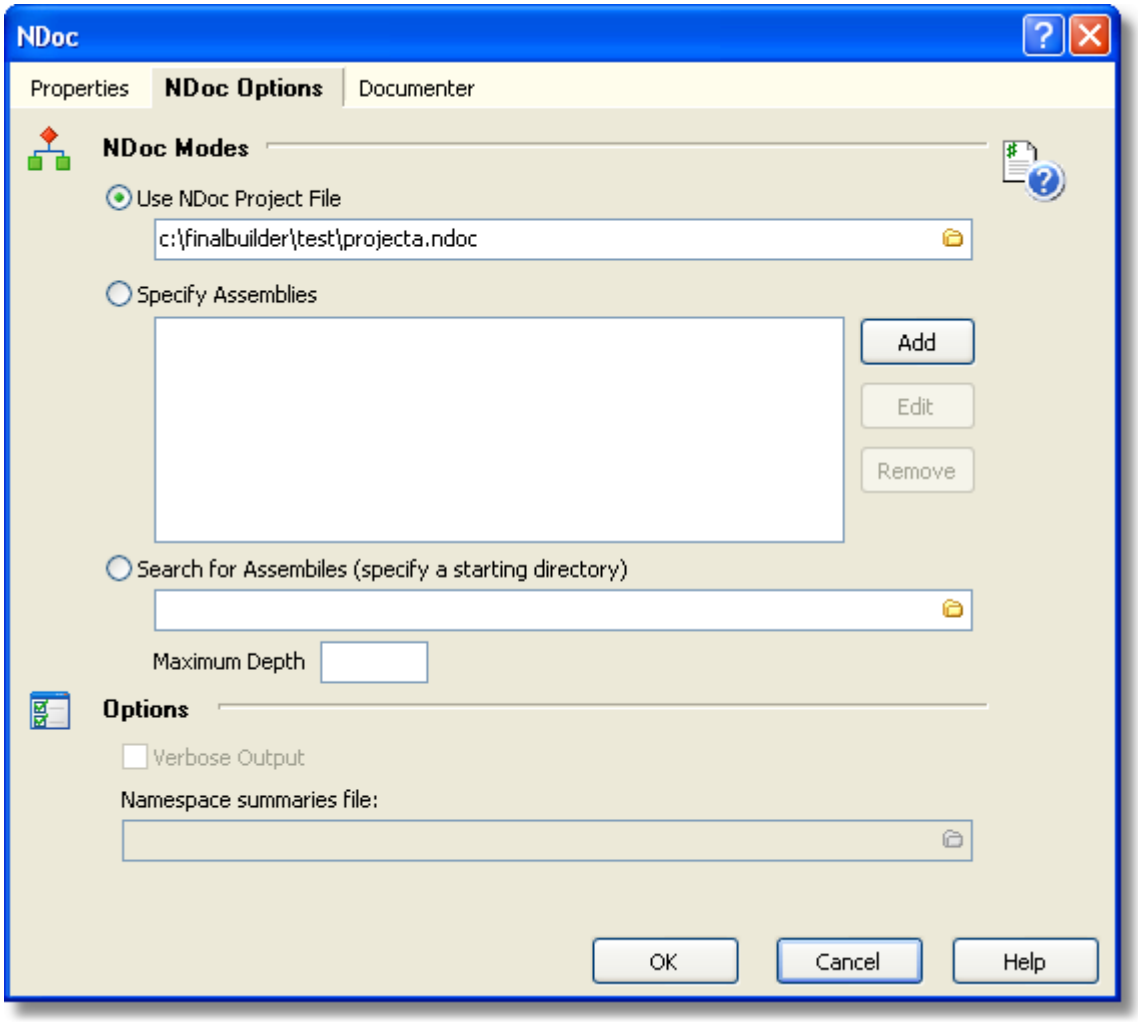
Keep Temp Files
Ensures the files generated by the 'HTML Help', 'WinHelp' and 'Visual Studio Help' help file compilers are not deleted.

© 2001-2006 VSoft Technologies Pty Ltd

5.15.5 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:

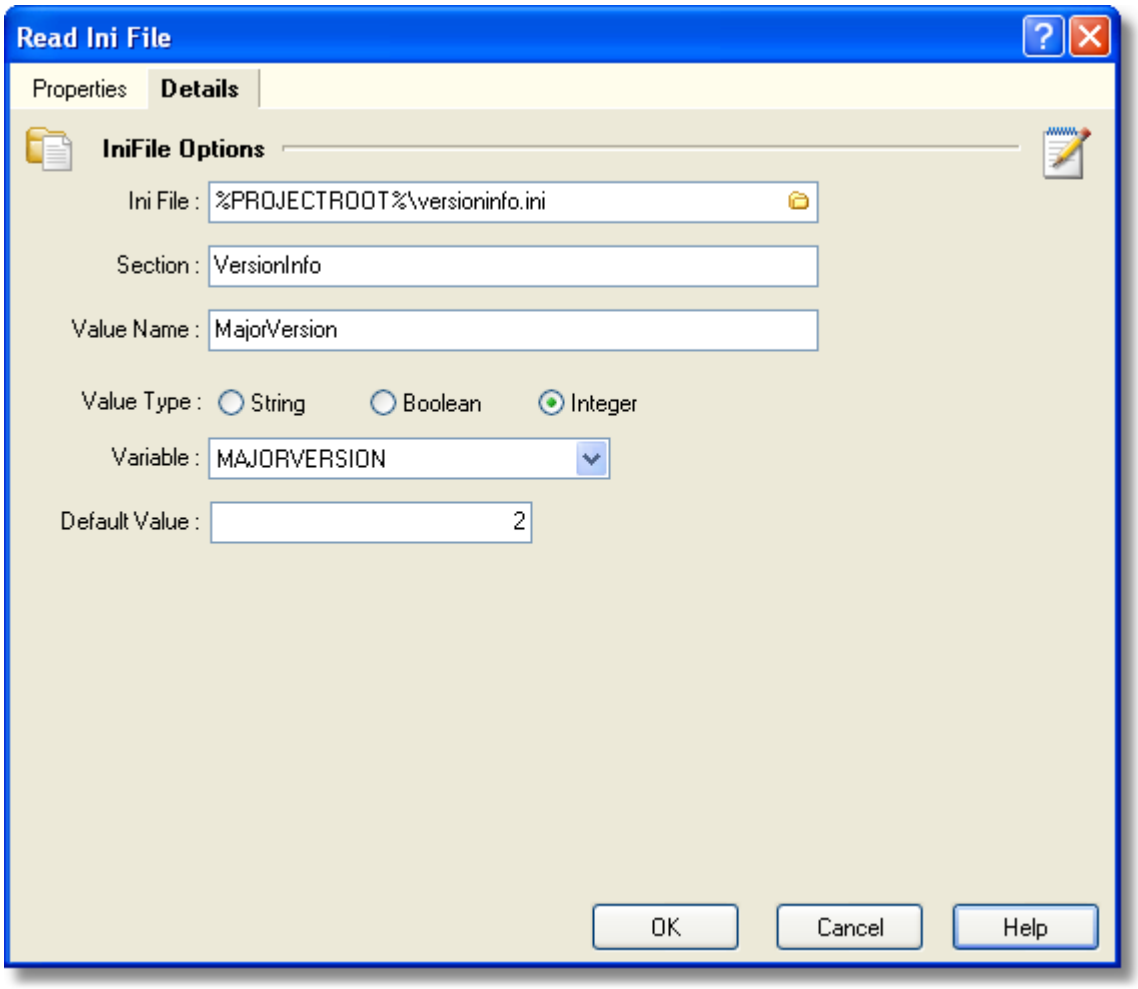
http://ndoc.sourceforge.net/wiki

© 2001-2006 VSoft Technologies Pty Ltd

5.16 Ini Files & Registry

5.16.1 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 :

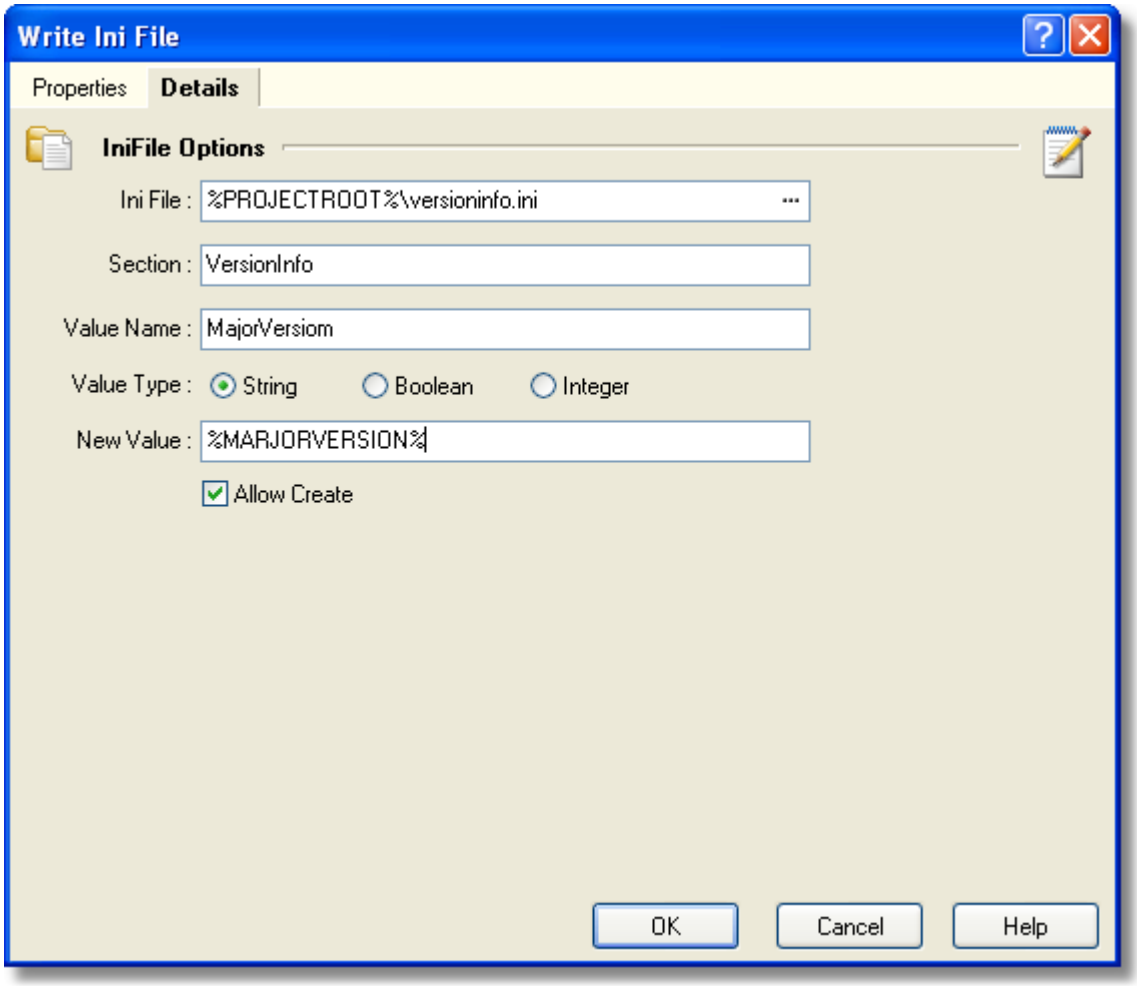
property IniFile : WideString
property Section : WideString
property ValueName : WideString
property VariableName : WideString
property DefaultValue : WideString

See also INI file format information

© 2001-2006 VSoft Technologies Pty Ltd

5.16.2 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 :

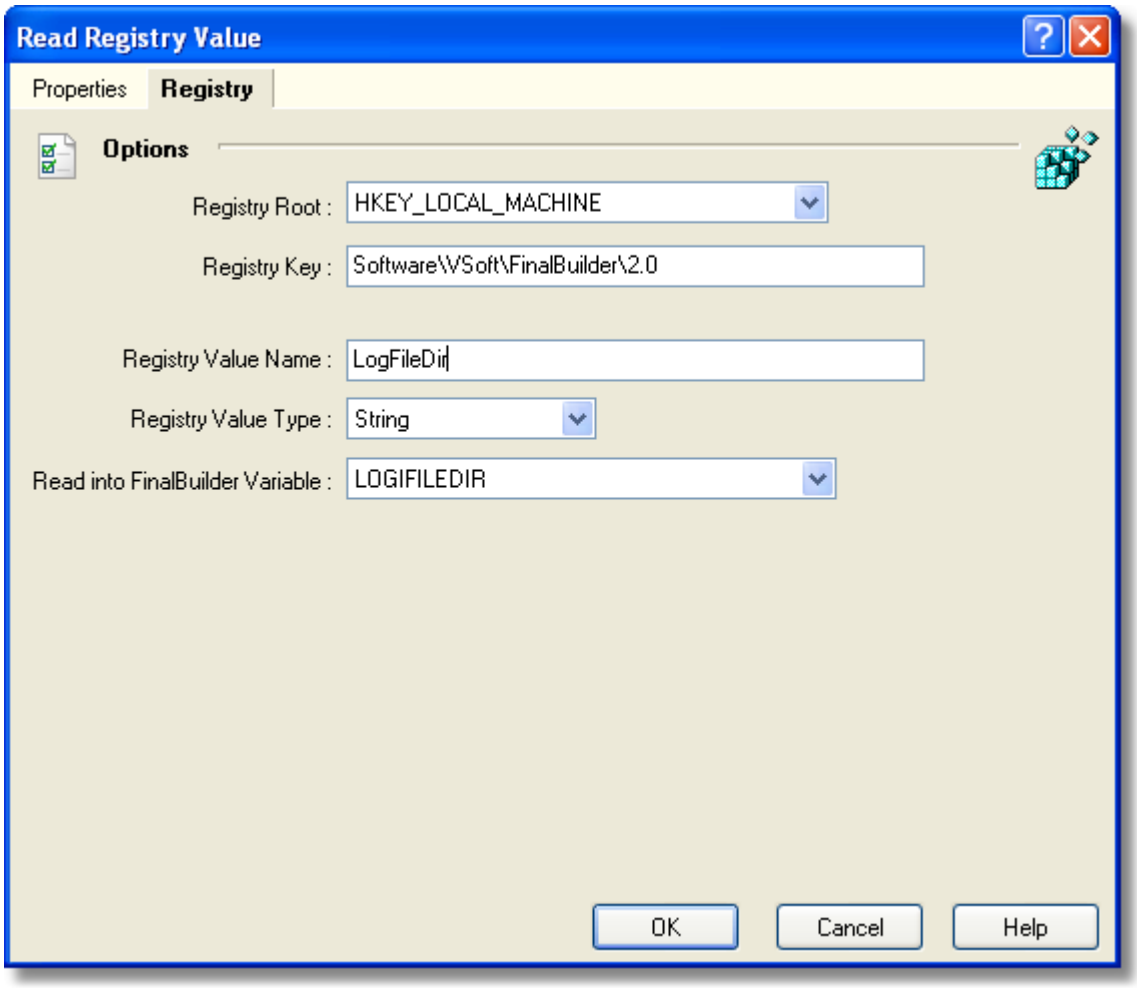
property IniFile : WideString
property Section : WideString
property ValueName : WideString
property NewValue : WideString
property AllowCreate : WordBool

See also INI file format information

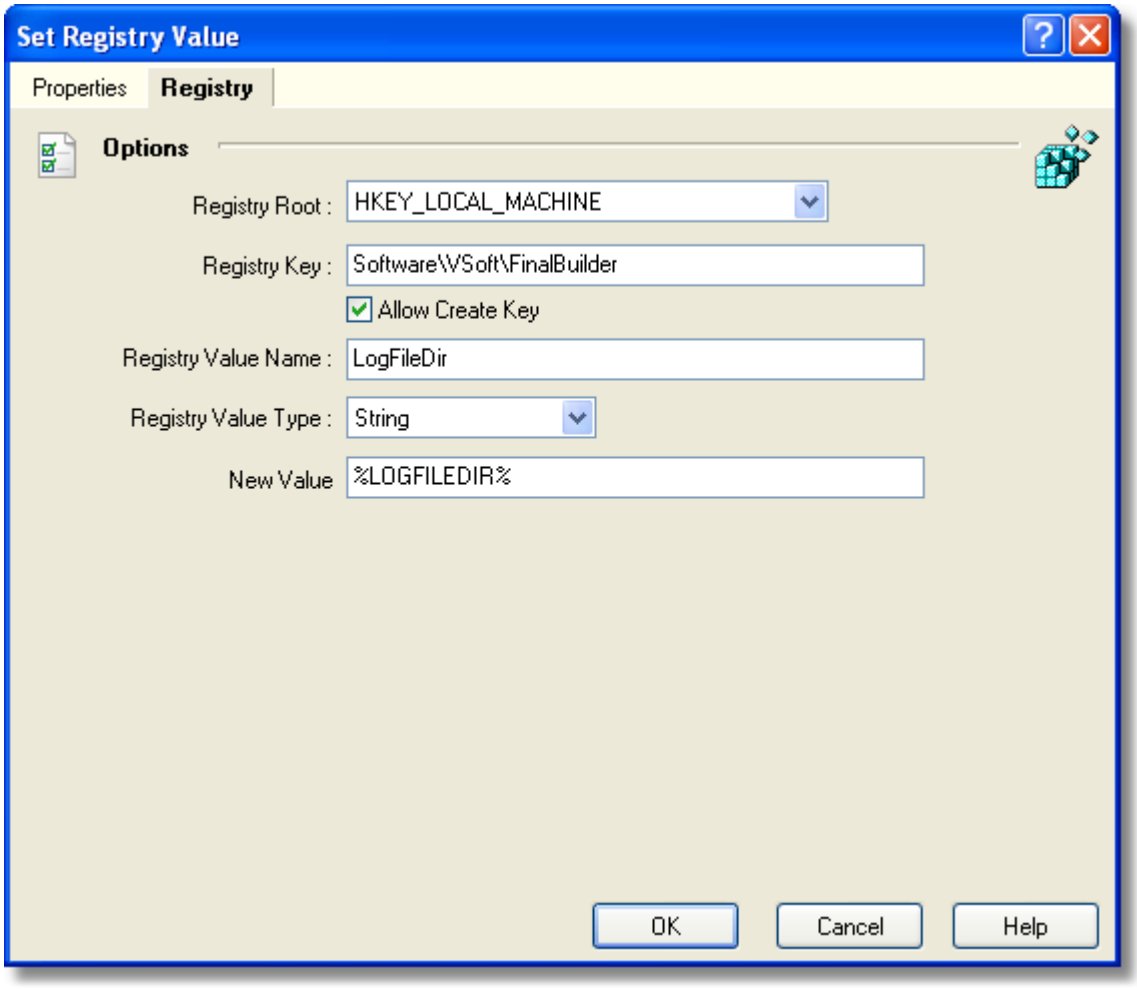
© 2001-2006 VSoft Technologies Pty Ltd

5.16.3 Read/Set/Delete Registry Value

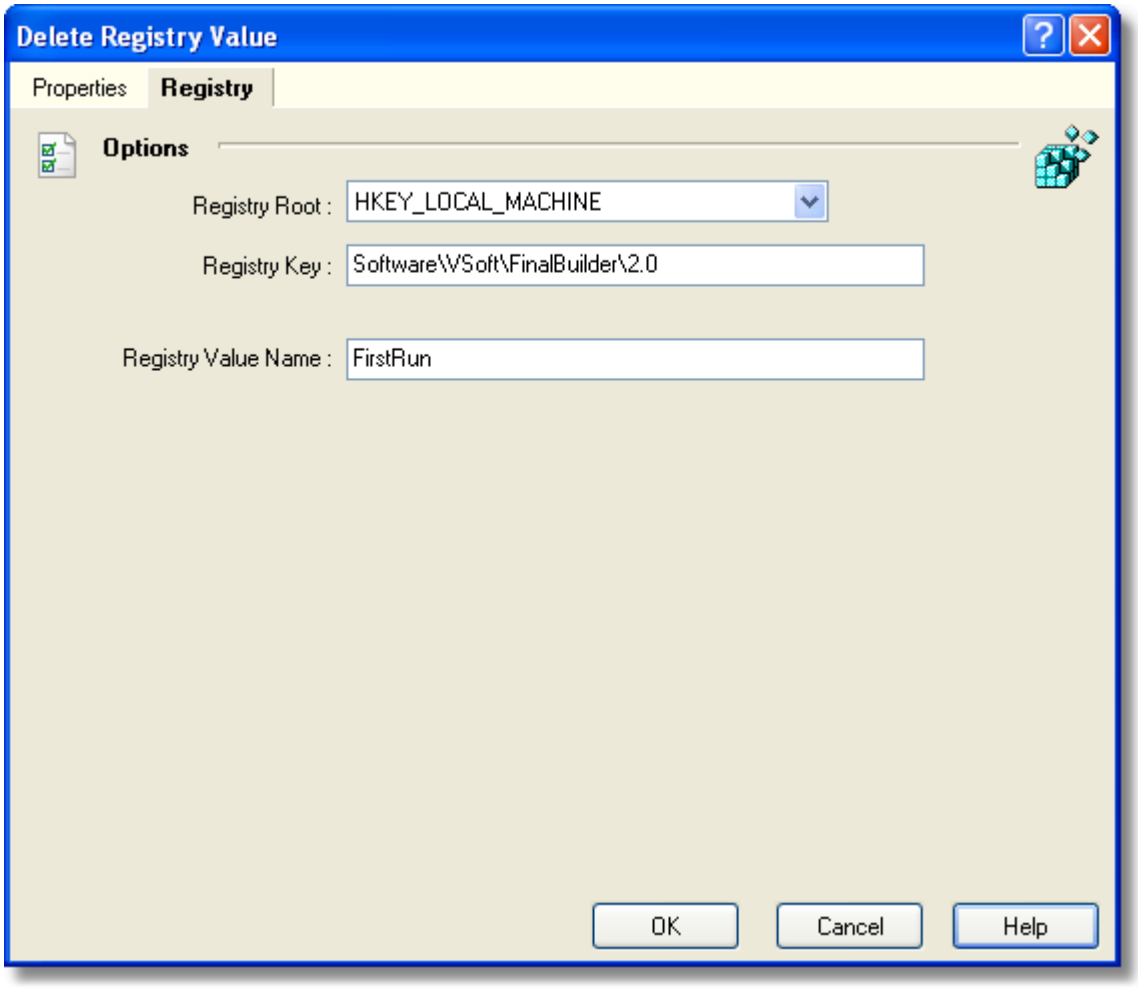
These actions allow you to Read a Registry Value into a FinalBuilder variable, Set a Registry Value or Delete a Registry Value.



© 2001-2006 VSoft Technologies Pty Ltd



© 2001-2006 VSoft Technologies Pty Ltd



Scripting Info

The Action properties available for all 3 actions are :

```
property RegRoot      : HKEY;  valid values are :
                          HKEY_CLASSES_ROOT,HKEY_CLASSES_ROOT,
                          HKEY_CURRENT_USER,HKEY_LOCAL_MACHINE,
                          HKEY_USERS,HKEY_CURRENT_CONFIG
property RegKey       : String;
property RegValType   : TRegValueType - valid values are rtInteger and rtString
property RegValueName : String;
```

For Read Value :

```
property FBVariable : string;
```

For Set Value:

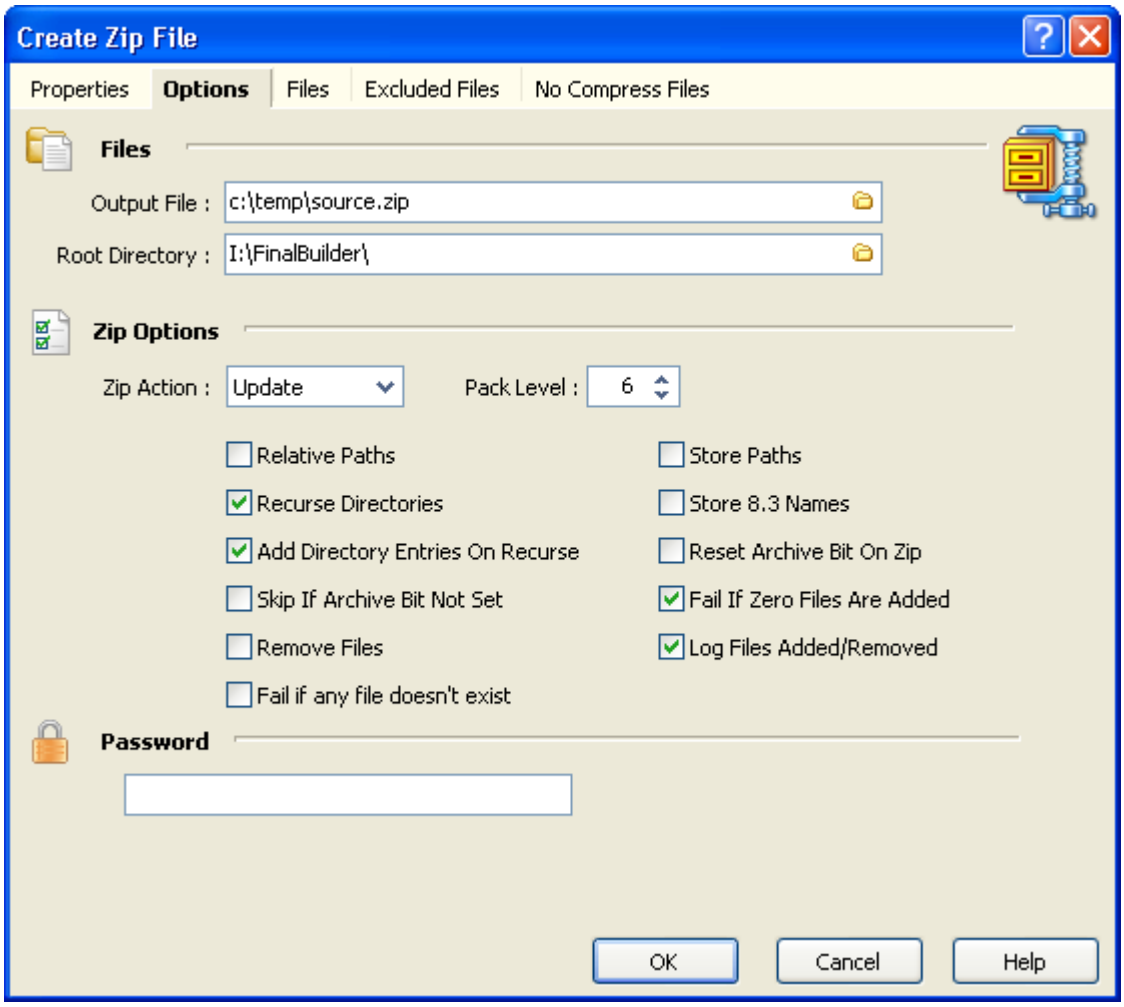
```
property AllowCreateKey;
```

© 2001-2006 VSoft Technologies Pty Ltd

5.17 Archiving

5.17.1 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

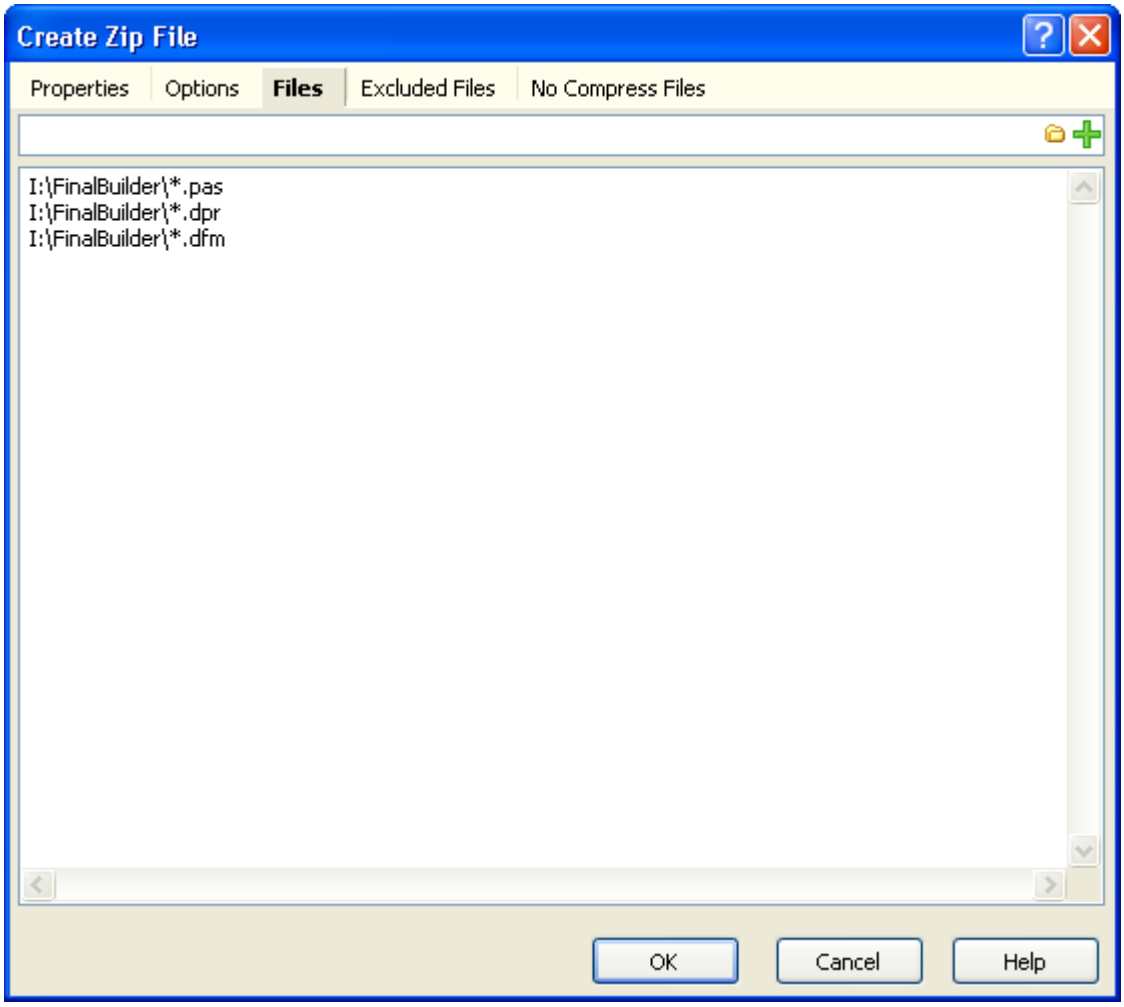
Skip if Archive bit Not Set : Setting this to True will cause files that do not have their Archive Bit set (turned on) to be skipped during zip operations. Therefore, while this is set to True, only files with their Archive attribute turned on will be zipped.

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.

© 2001-2006 VSoft Technologies Pty Ltd



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

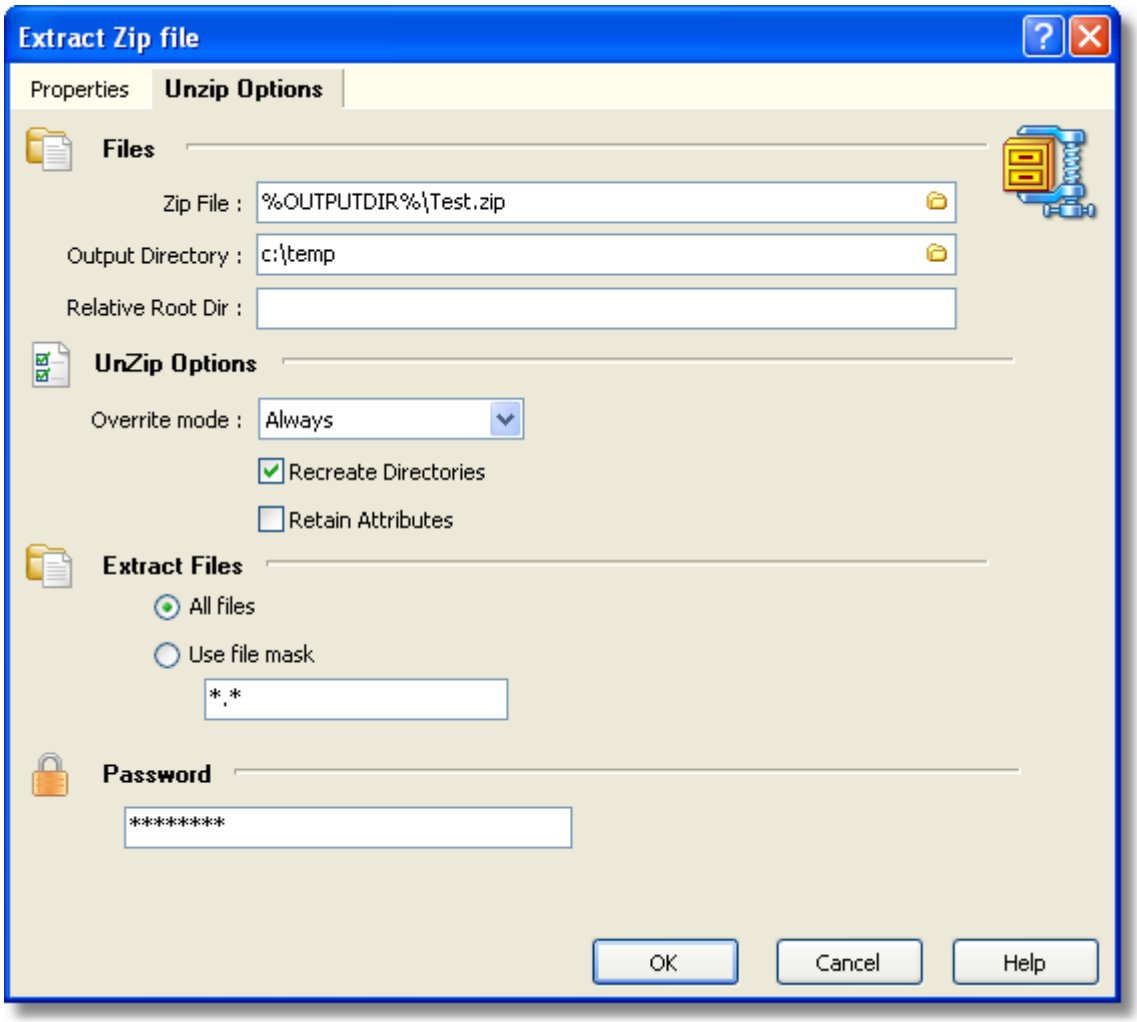
The Action properties available are :

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

© 2001-2006 VSoft Technologies Pty Ltd

5.17.2 Extract Zip File Action

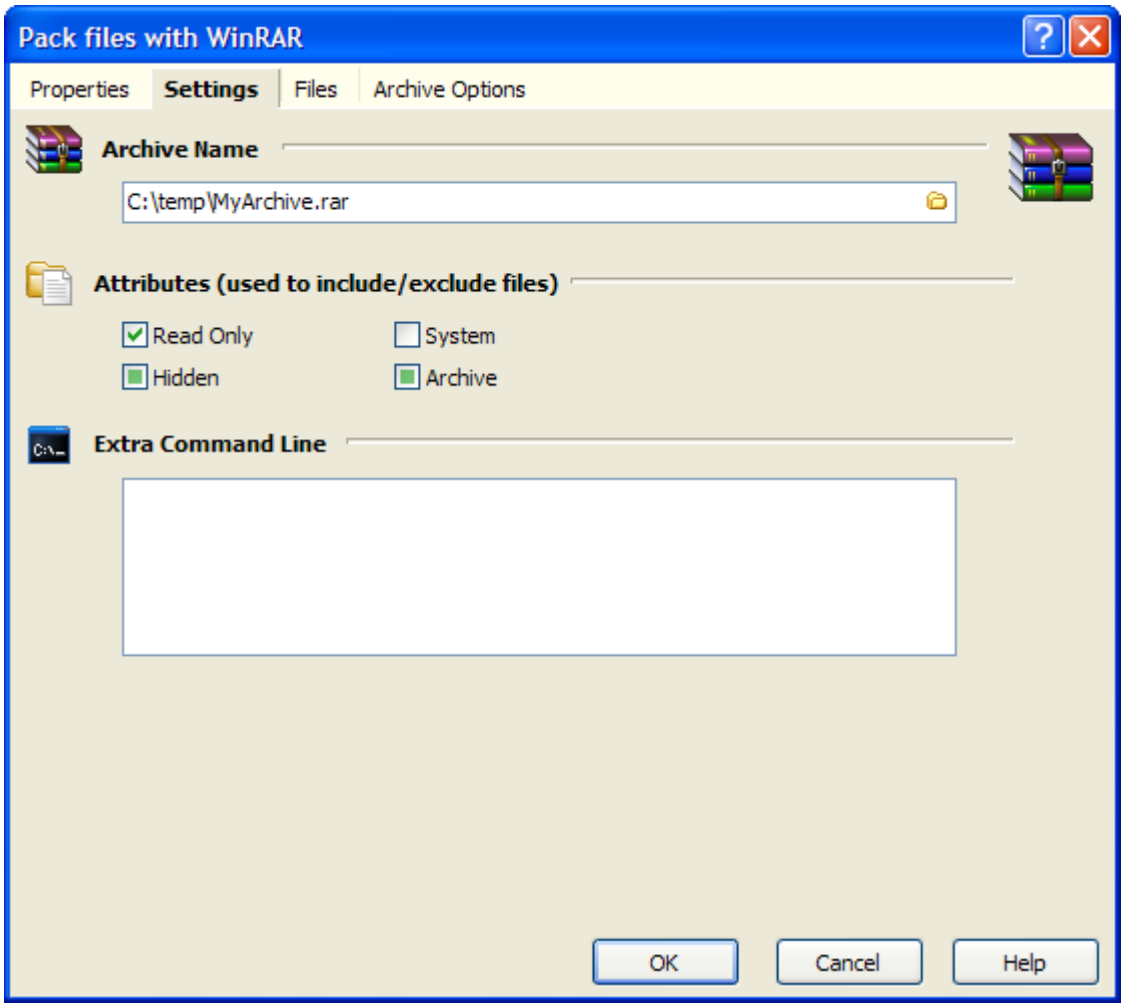
This action allows you to extract files from a zip archive file. You can choose which files to extract using file masks or just extract all files.



© 2001-2006 VSoft Technologies Pty Ltd

5.17.3 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

© 2001-2006 VSoft Technologies Pty Ltd

5.17.4 7Zip

The 7Zip actions enable you to automate archive operations as part of your build process.

The archive operations available are:

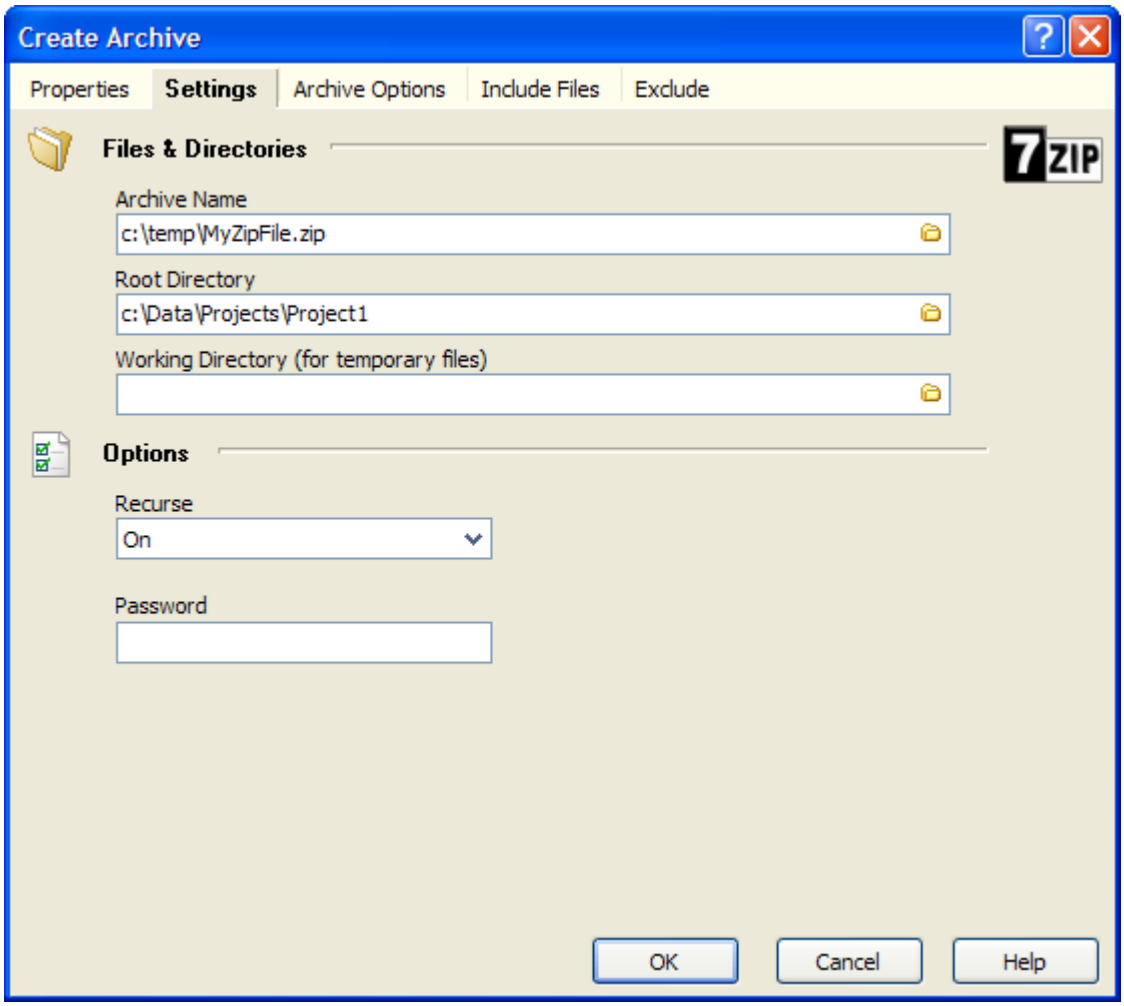
Create Archive (supports Zip, 7z, GZip, BZip2, TAR)
Test Archive (supports Zip, 7z, GZip, BZip2, TAR)
List files in Archive (supports Zip, 7z, GZip, BZip2, TAR, RAR, ARJ, CAB, CPIO, RPM, DEB, SPLIT)
Extract Archive (supports Zip, 7z, GZip, BZip2, TAR, RAR, ARJ, CAB, CPIO, RPM, DEB, SPLIT)
Update Archive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

5.17.4.1 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



© 2001-2006 VSoft Technologies Pty Ltd

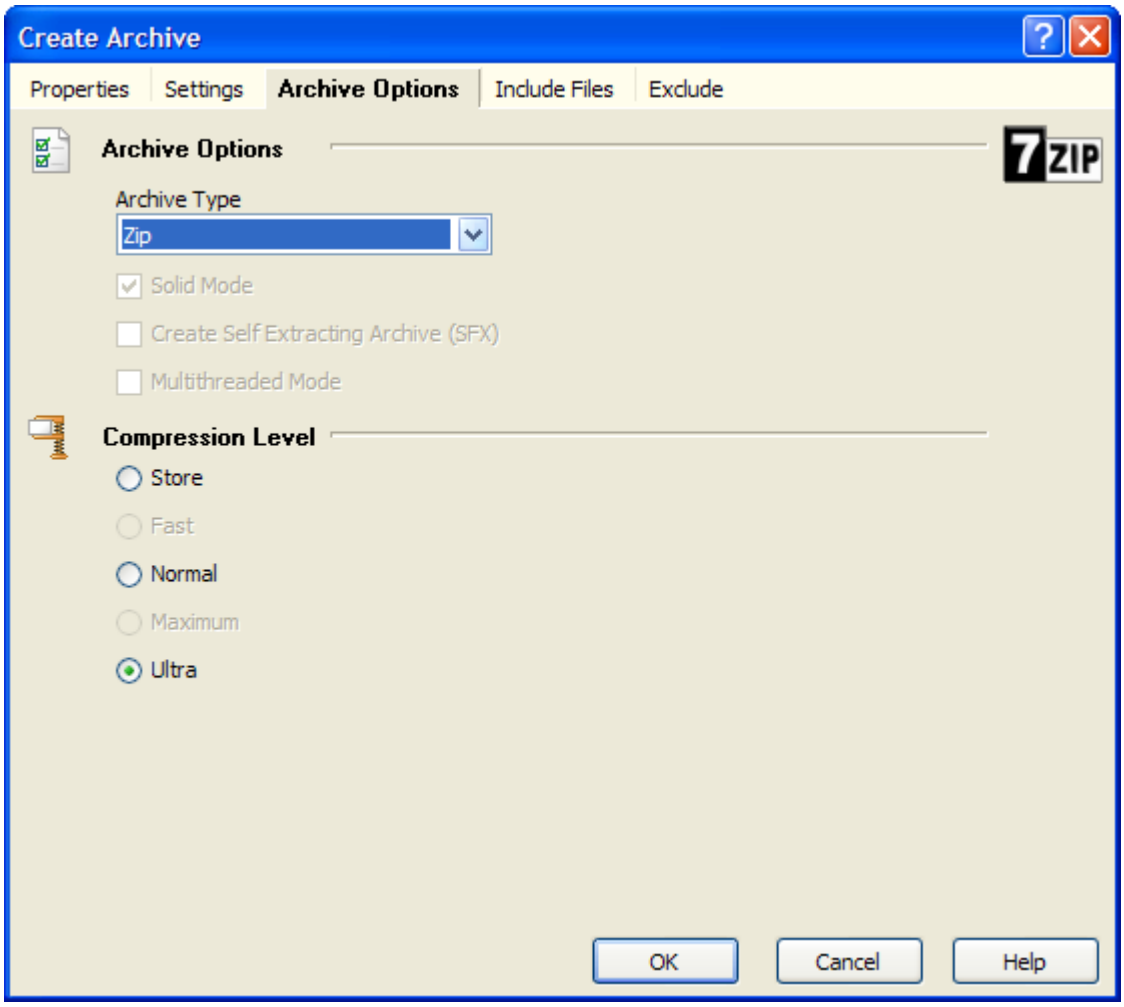
Archive Name - specify the name of the new archive. The file extension should match the type of archive you are creating (eg. ZIP)

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

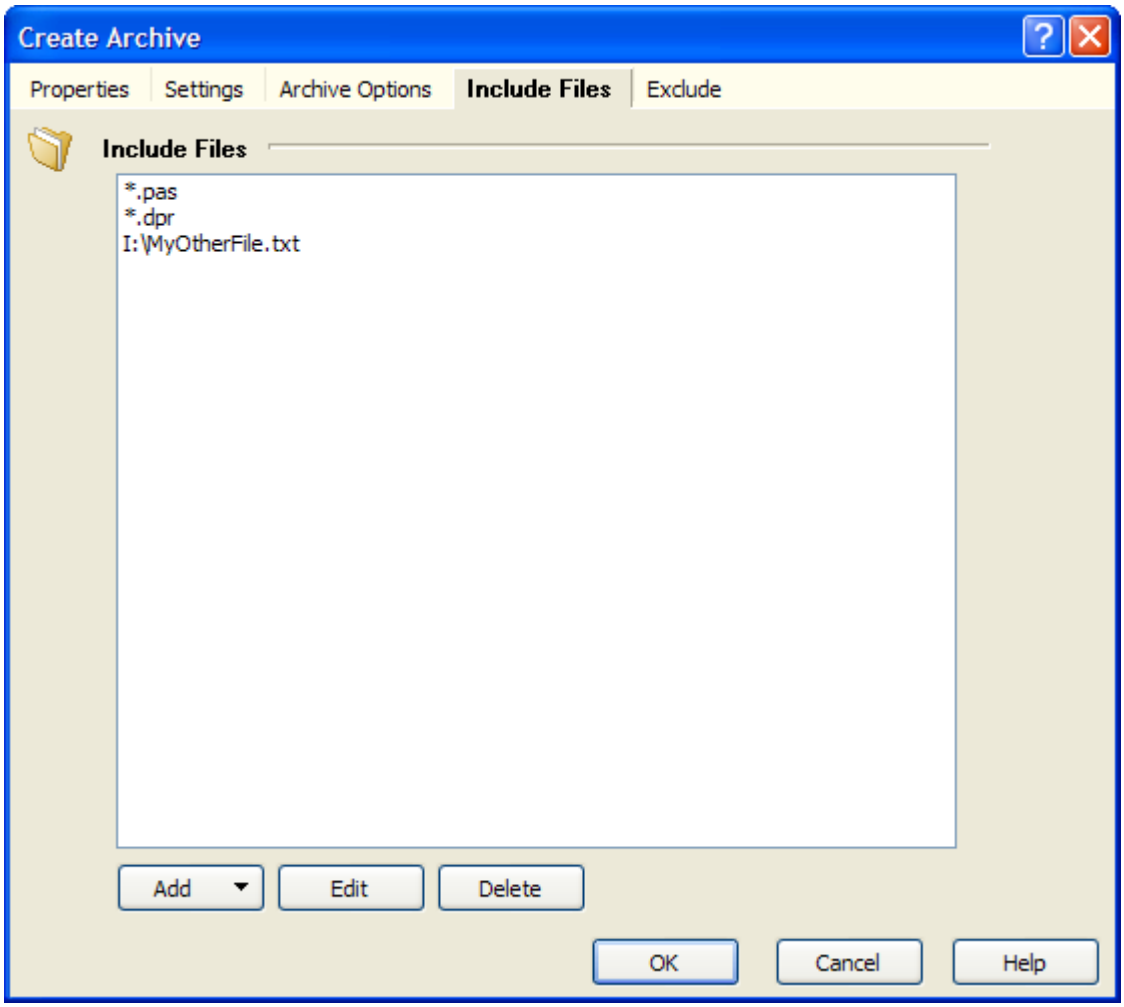


Archive Type - specify the archive type you want to create. It should match the file extension of the archive name.

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.

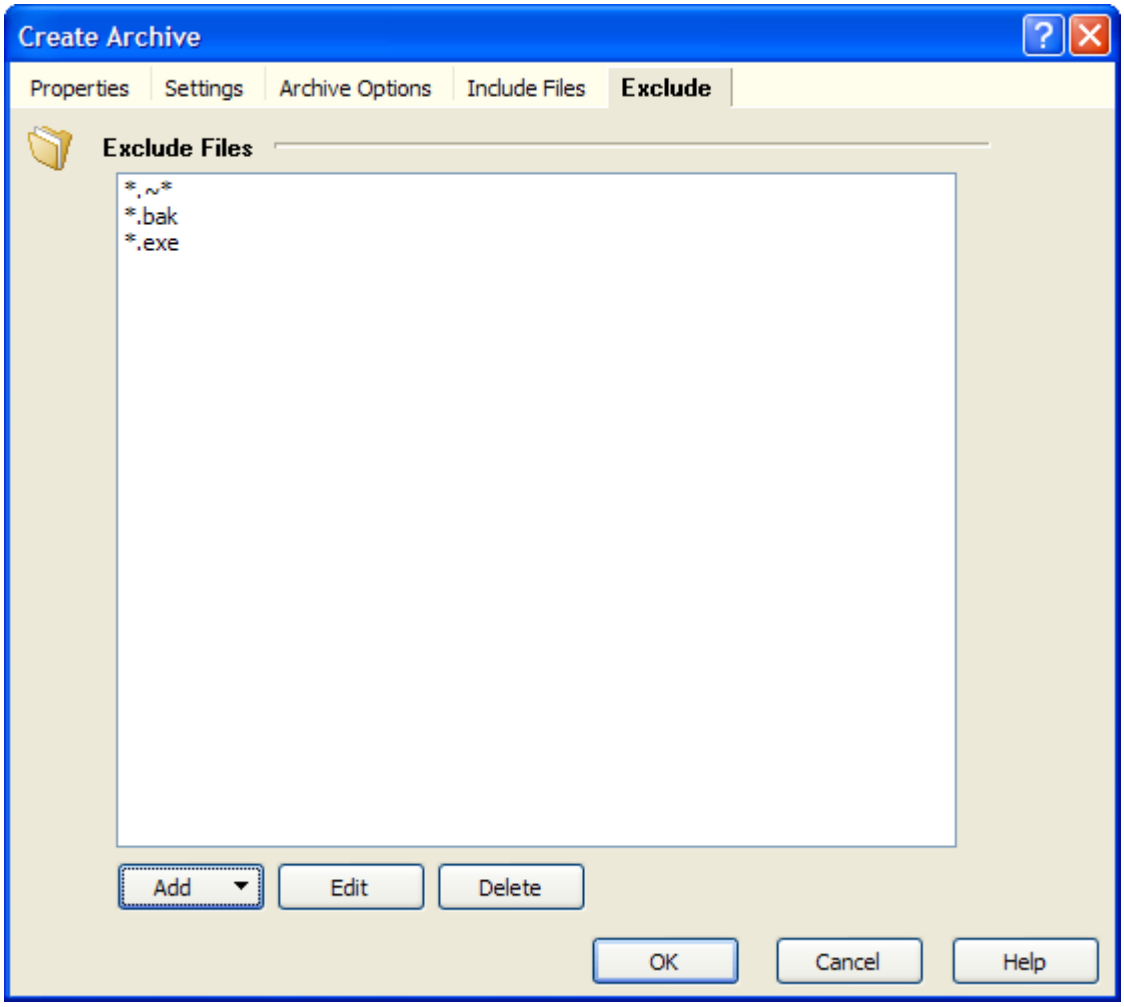
© 2001-2006 VSoft Technologies Pty Ltd



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

© 2001-2006 VSoft Technologies Pty Ltd



Specify any files or filespecs to exclude from the archive.

5.17.4.2 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

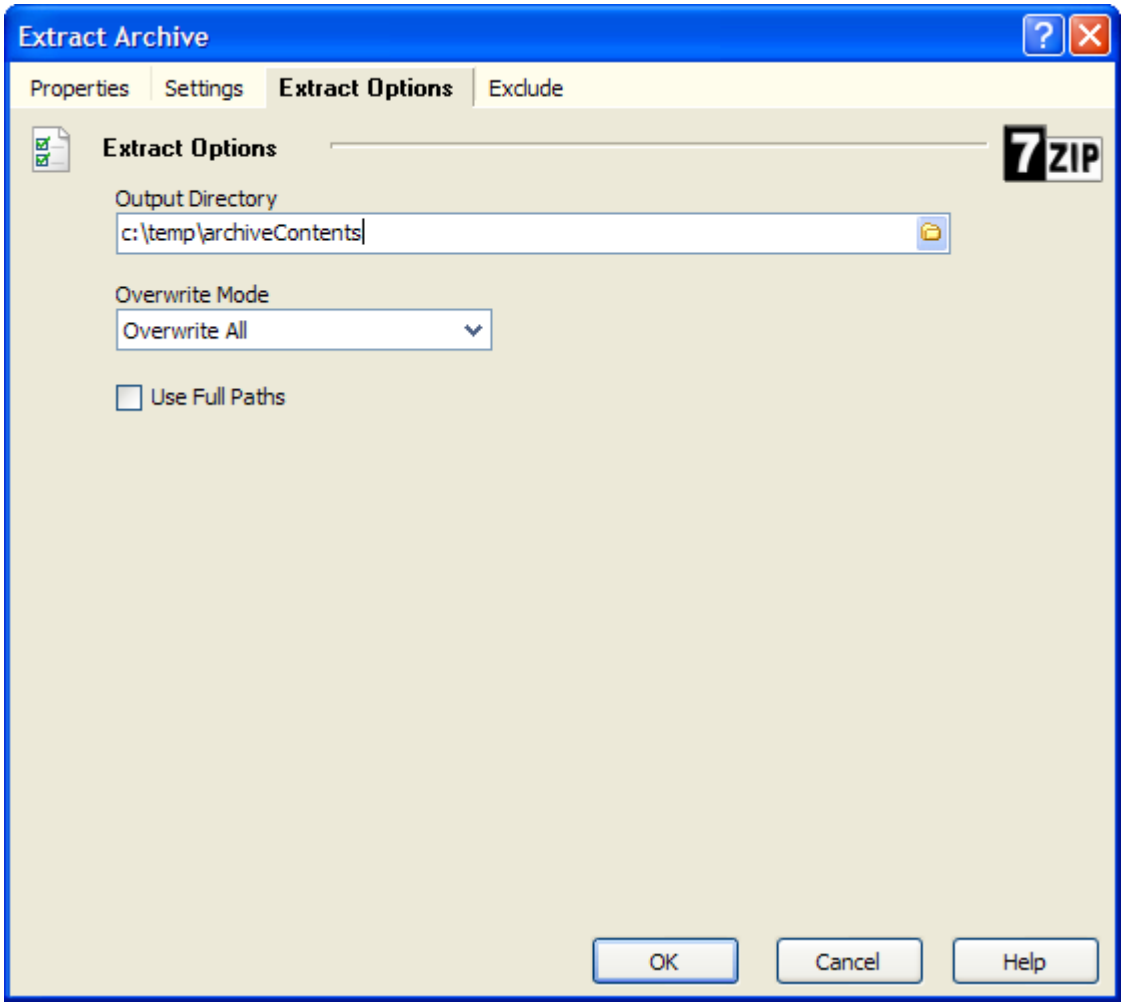
For more detailed descriptions of the options, see the Create Archive action.

5.17.4.3 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

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.17.4.4 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.

© 2001-2006 VSoft Technologies Pty Ltd

5.17.4.5 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.

5.17.4.6 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

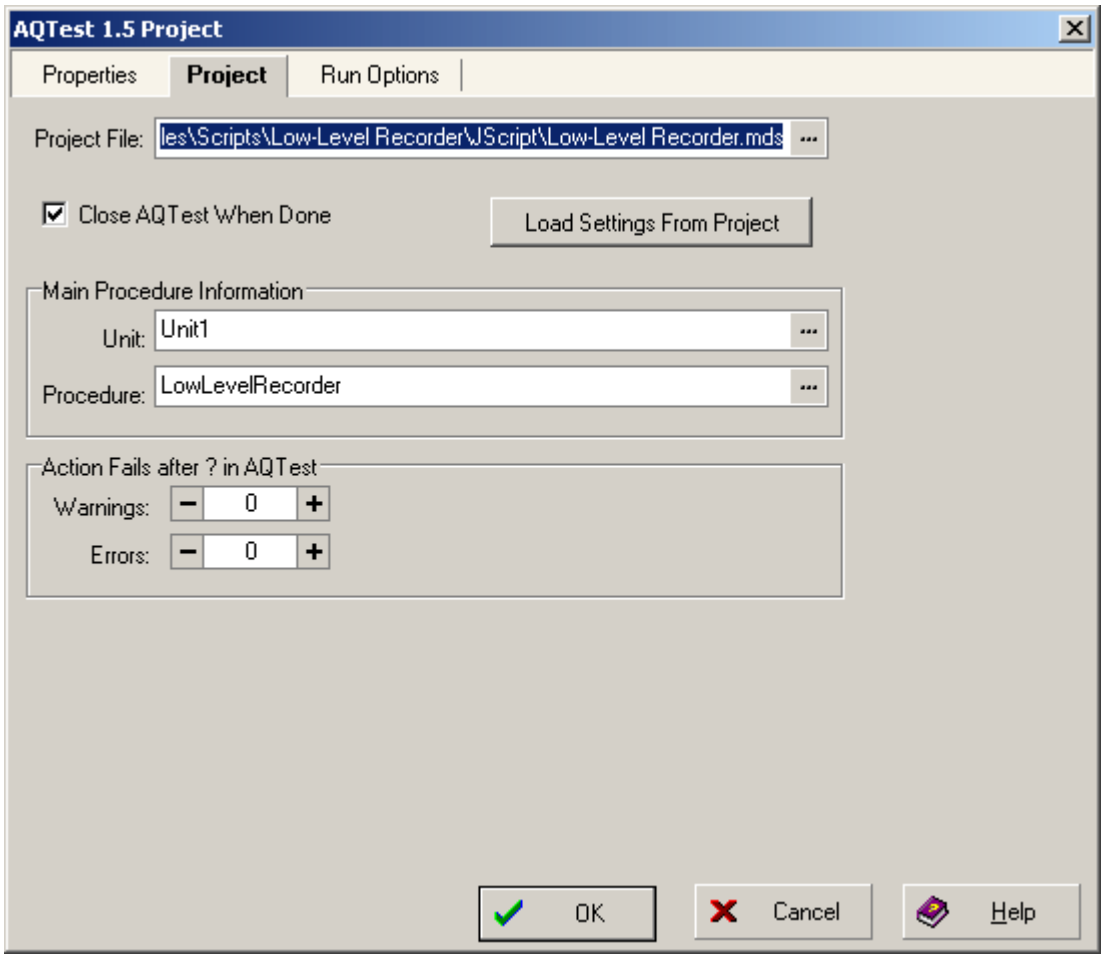
For more detailed descriptions of the options, see the Create Archive action.

© 2001-2006 VSoft Technologies Pty Ltd

5.18 Testing Tools

5.18.1 AQTest

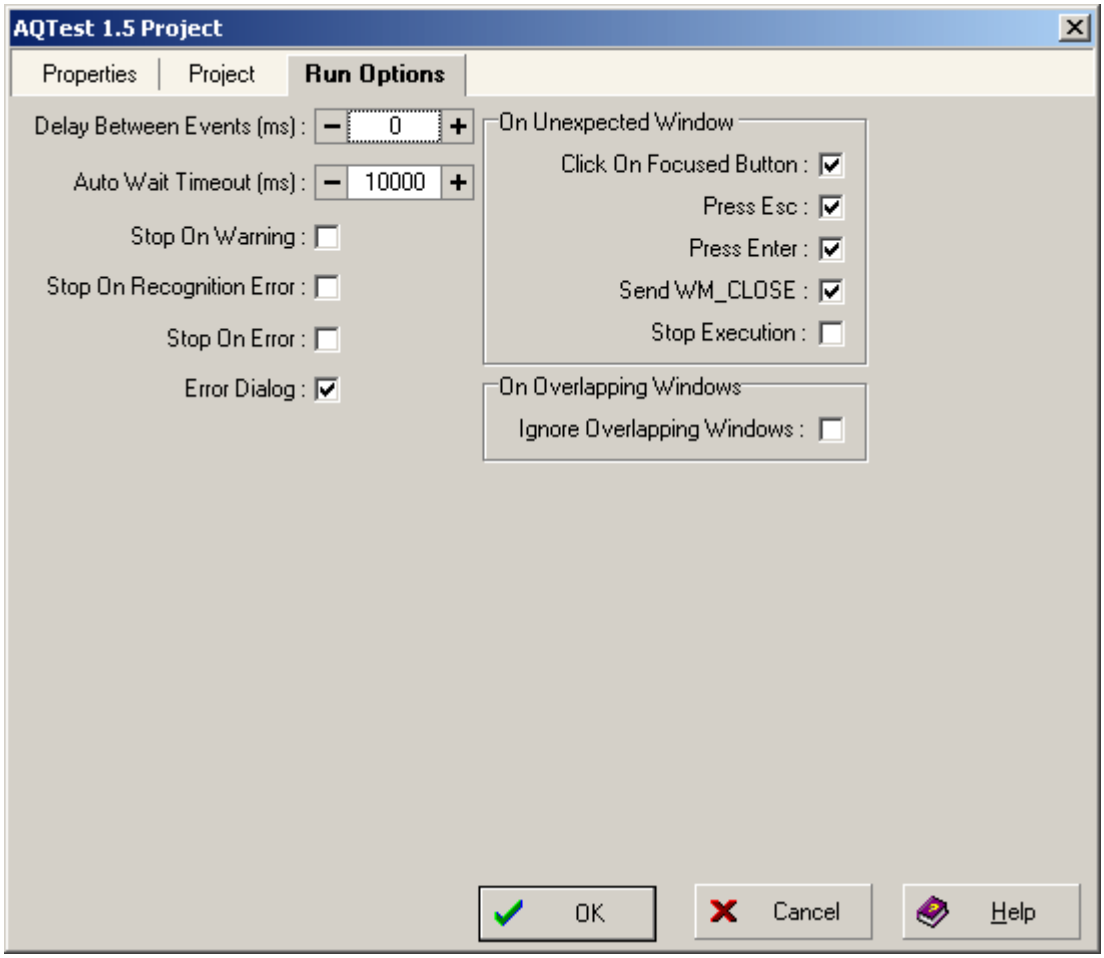
This action integrates Automated QA's AQTest automated testing tool with FinalBuilder. This makes it possible to perform Regression testing as part of the build process.



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.

© 2001-2006 VSoft Technologies Pty Ltd



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

© 2001-2006 VSoft Technologies Pty Ltd

5.18.2 AutomatedQA TestComplete 2.x/3.x Actions

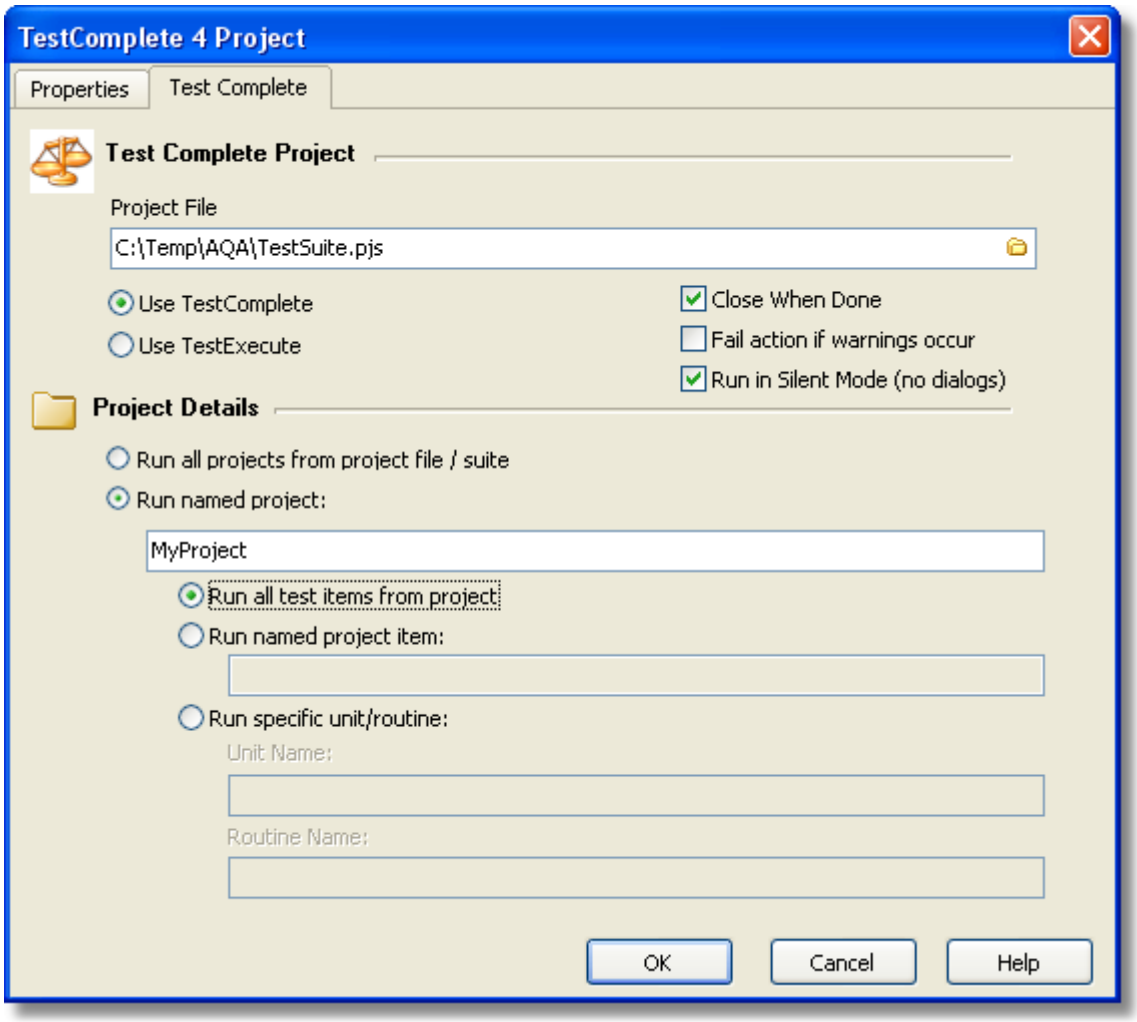
These Actions provide integration with AutomatedQA's TestComplete 2.x & 3.x products. They were kindly provided by Eric Holton - http://www.holtonsystems.com

© 2001-2006 VSoft Technologies Pty Ltd

5.18.3 AutomatedQA TestComplete 4 Action

The Test Complete 4 Project action provides support for execution of Automated QA TestComplete 4 & TestExecute 4 projects.

You will need to set the path to TestComplete 4 under Tools menu -> Options -> Testing Tools.



TestComplete 4 Project

The full path to a TestComplete 4 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 4.

Close When Done

Select to close TestComplete 4 when the test is completed. Note that TestComplete

© 2001-2006 VSoft Technologies Pty Ltd

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.

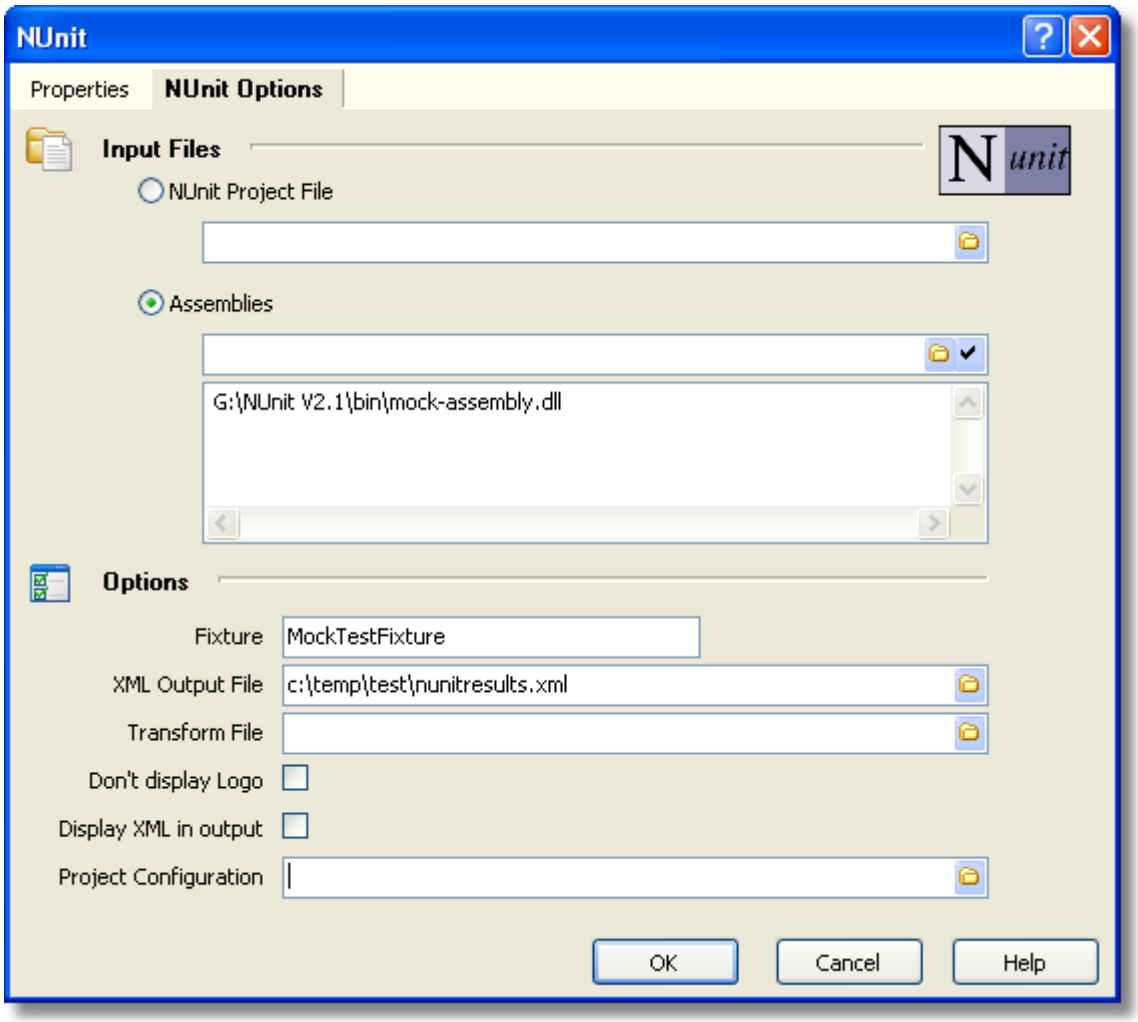
Run all test items / Run named project item / Run specific unit/routine

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.18.4 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/

© 2001-2006 VSoft Technologies Pty Ltd

5.18.5 MSTest

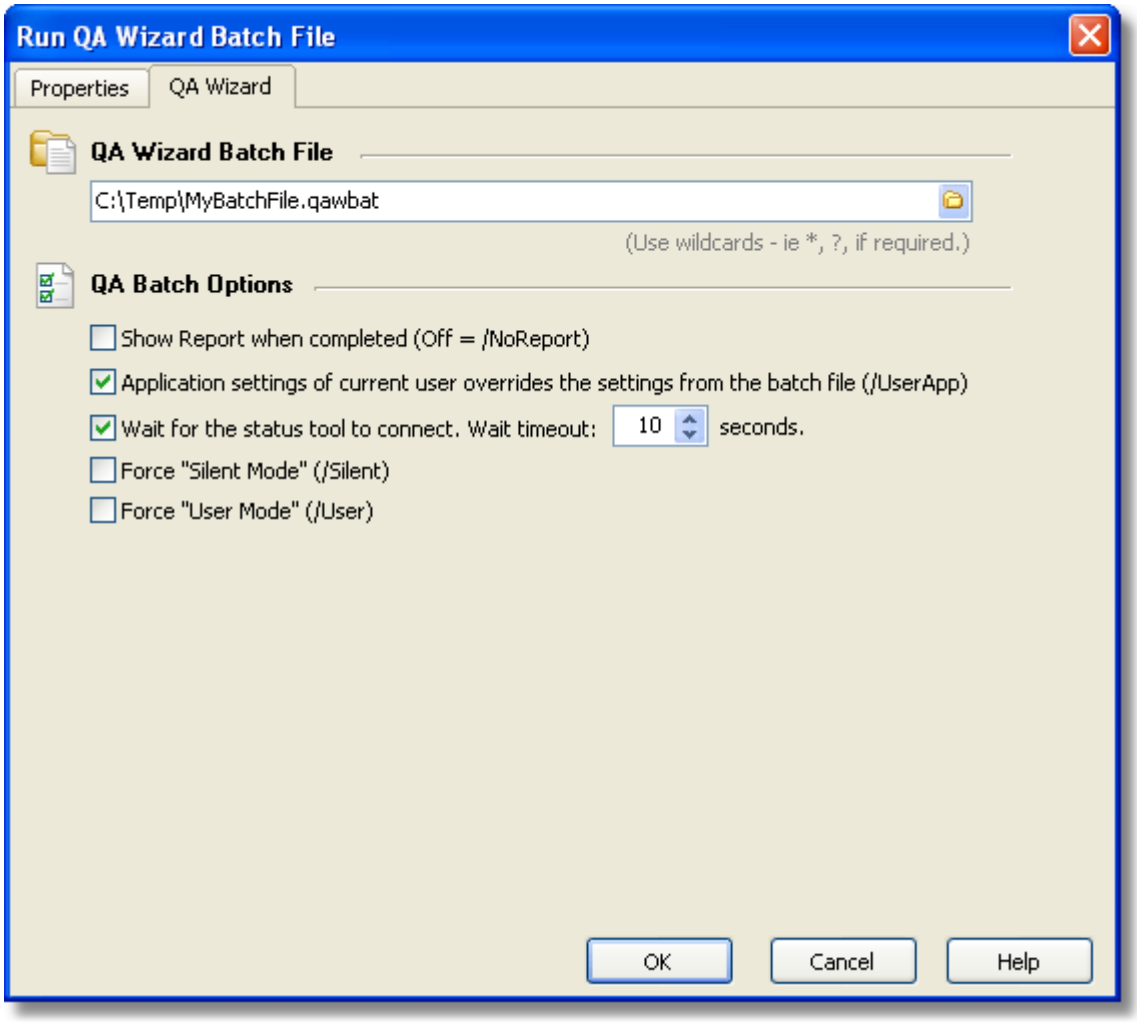
The MS Test action enables you to automate the testing of your .Net assemblies using MSTest.

© 2001-2006 VSoft Technologies Pty Ltd

5.18.6 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

© 2001-2006 VSoft Technologies Pty Ltd

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"

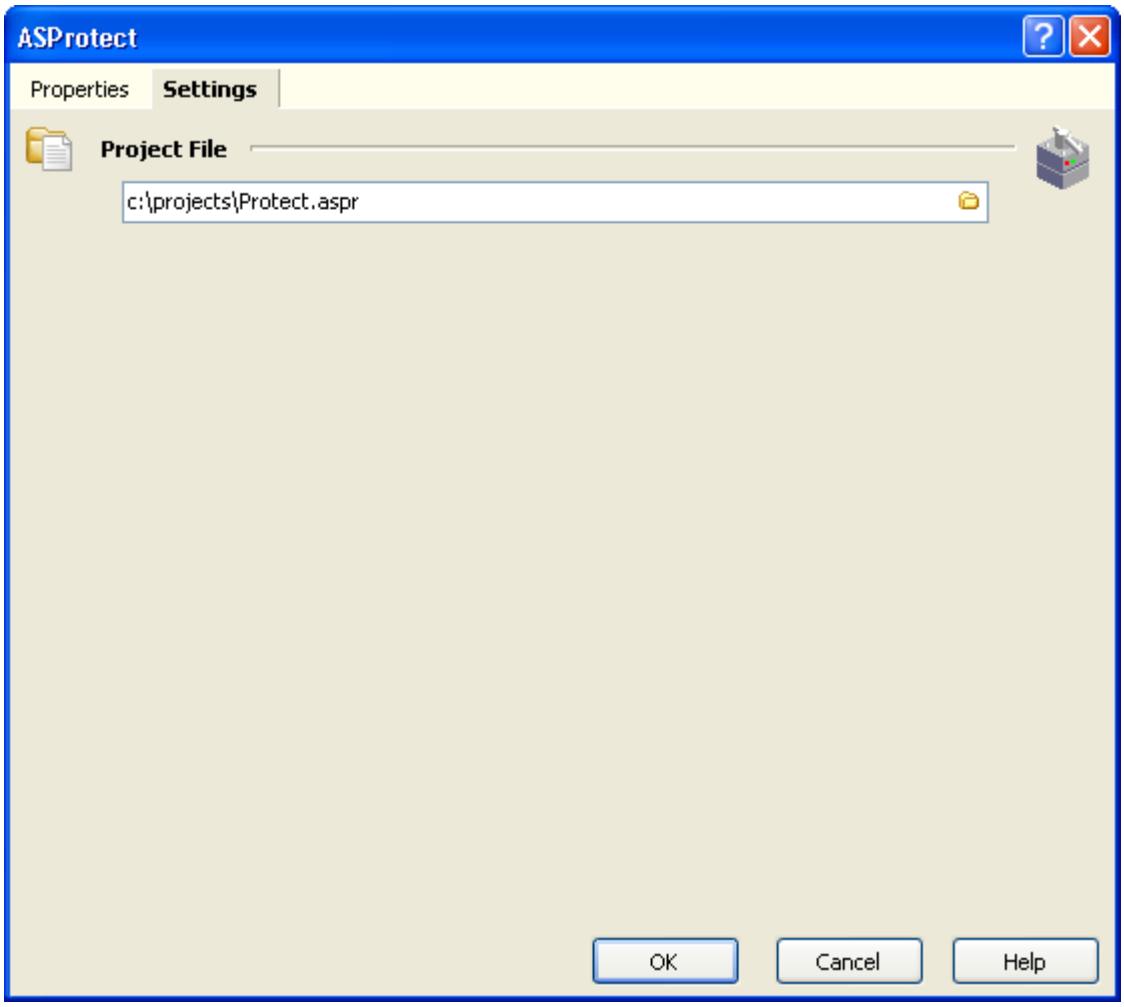
Forces the batch script to start in "user" mode. Equivalent to the /User switch. Mutually exclusive with "Silent Mode".

© 2001-2006 VSoft Technologies Pty Ltd

5.19 Licensing Tools

5.19.1 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;

The Options Object is exposed as :

function ASProtectOptions : IFBASProtectOptions

It has one property:

property ASProtectLocation: string

© 2001-2006 VSoft Technologies Pty Ltd

5.19.2 ProActivate Action

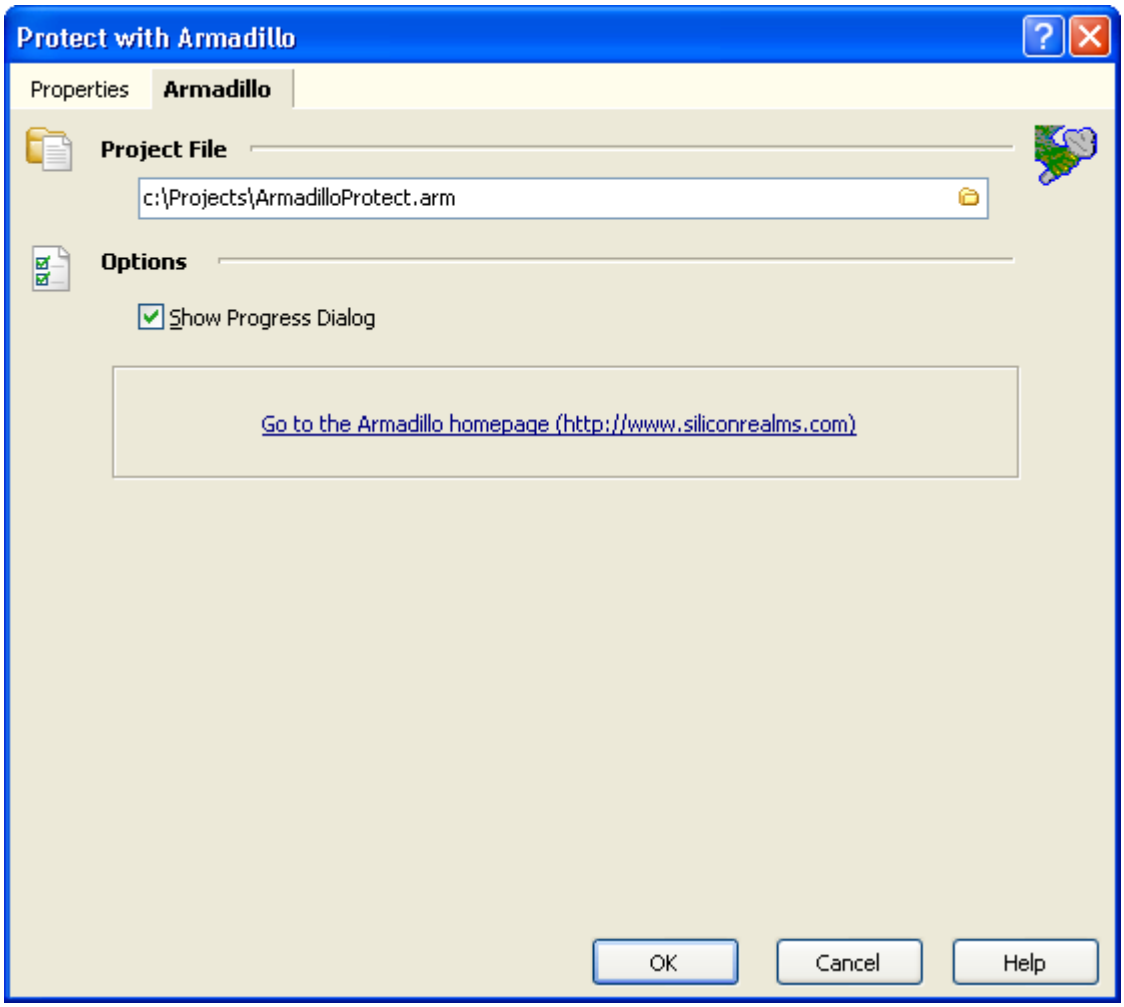
This Action provides support for TurboPower's ProActivate software Licensing product. For more information on ProActivate please visit the TurboPower website http://www.turbopower.com

© 2001-2006 VSoft Technologies Pty Ltd

5.19.3 Armadillo Action

This action provides support for the Armadillio Software protection system. For more info on Armadillio, 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.

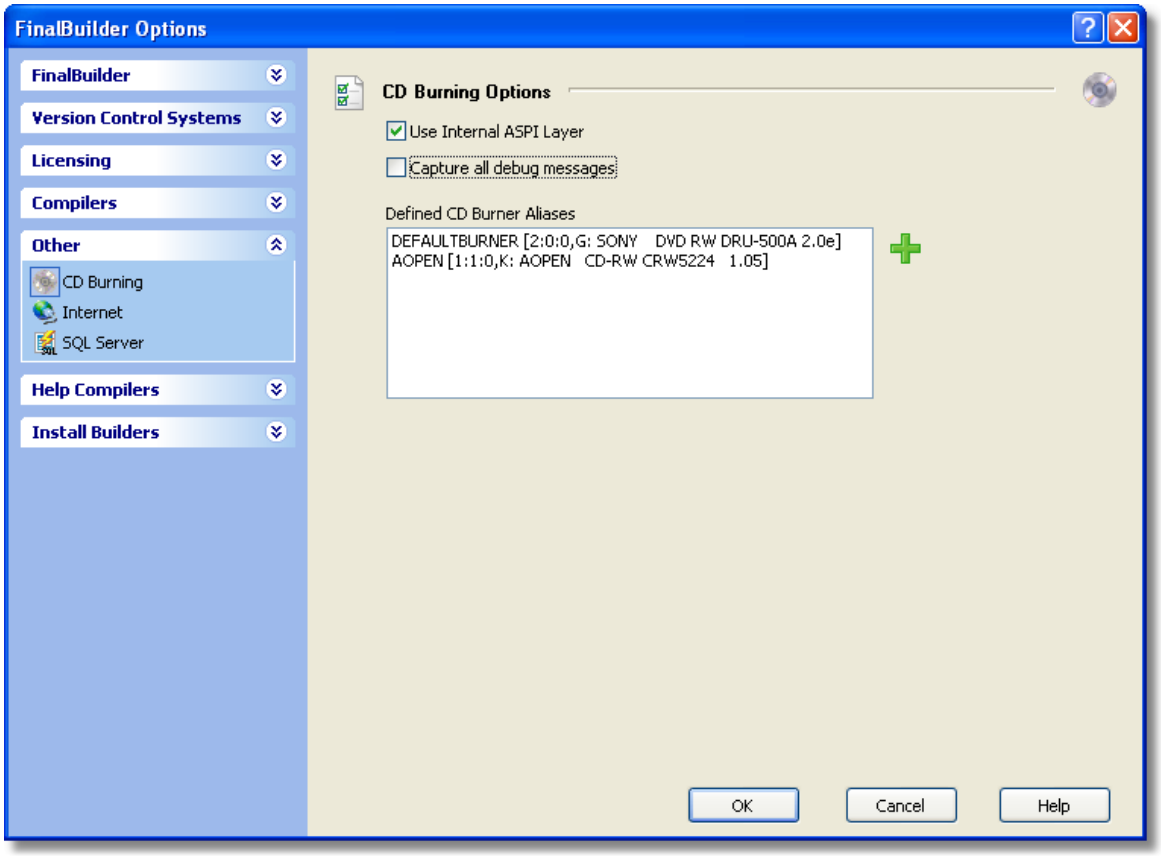
This action was written and kindly provided by Peter Thörnqvist.



© 2001-2006 VSoft Technologies Pty Ltd

5.20 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.

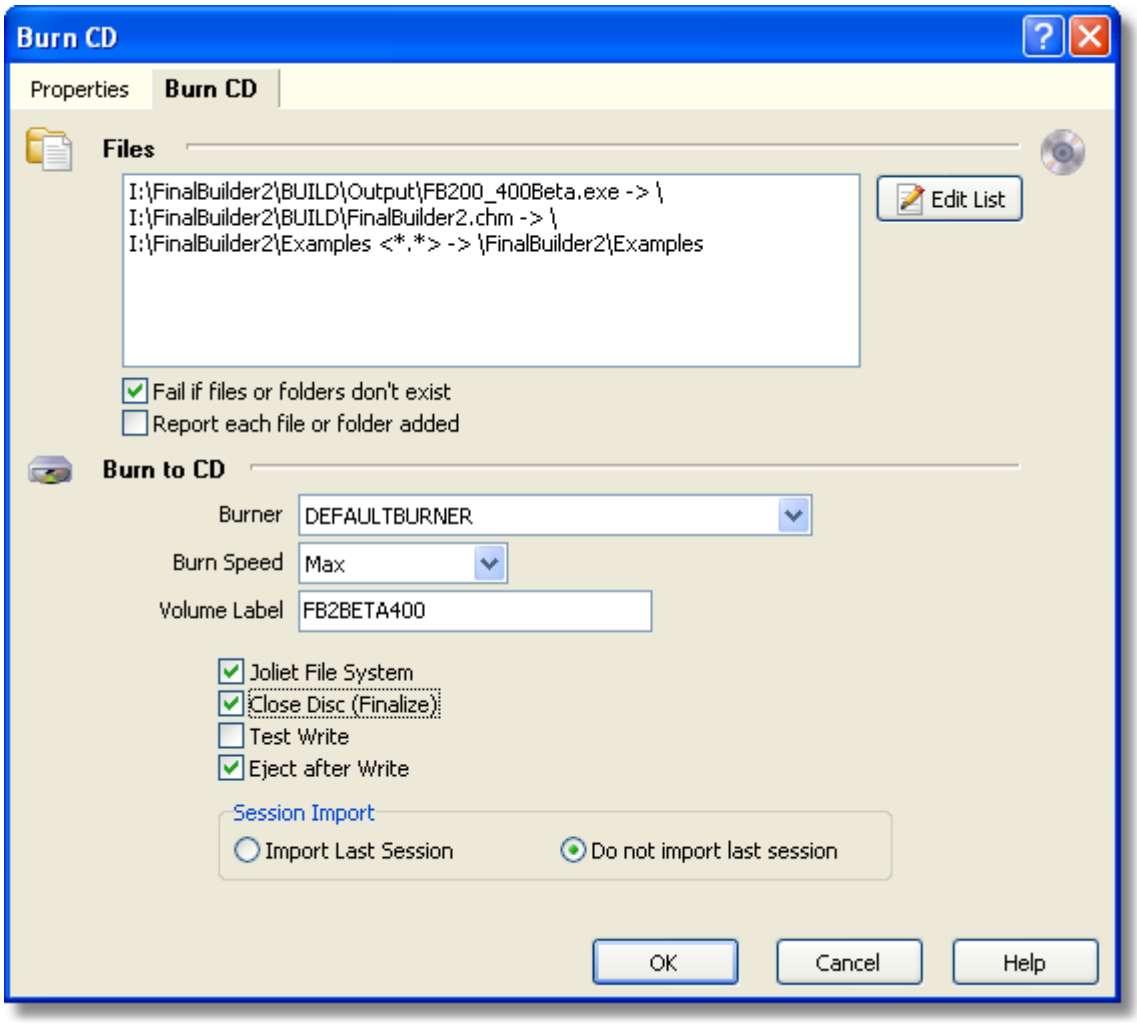


By default, FinalBuilder uses it's own internal ASPI layer (CD/DVD Burner API), which supports most types of drives including IDE, SCSI, USB & Firewire. If you find that FinalBuilder does not find your drive, or it has problems with it, try turning off this option and it will use an external aspi layer in WNASPI32.DLL. Note that the Adaptec layer commonly found on many machines does not support Firewire or USB devices, we recommend using the WNASPI32.DLL from Nero into your system32 directory if you have it and want to try an external aspi layer.

© 2001-2006 VSoft Technologies Pty Ltd

5.20.1 Burn CD/DVD Action

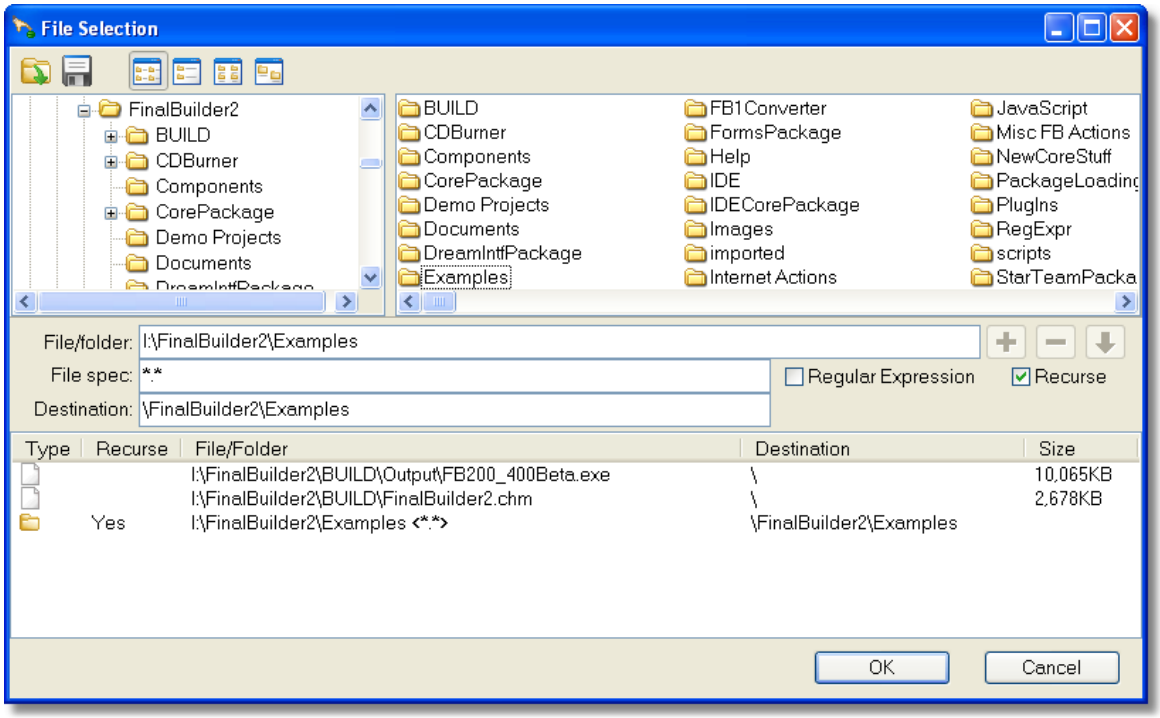
[Professional Edition]
This action supports burning files direct to CDR/CDRW/DVDR & DVDRW Media (depending on your burner hardware). The files burned directly to the drive, ie an image is not created first. Note that to use this action you need to define an alias for your burner first (see here for more 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

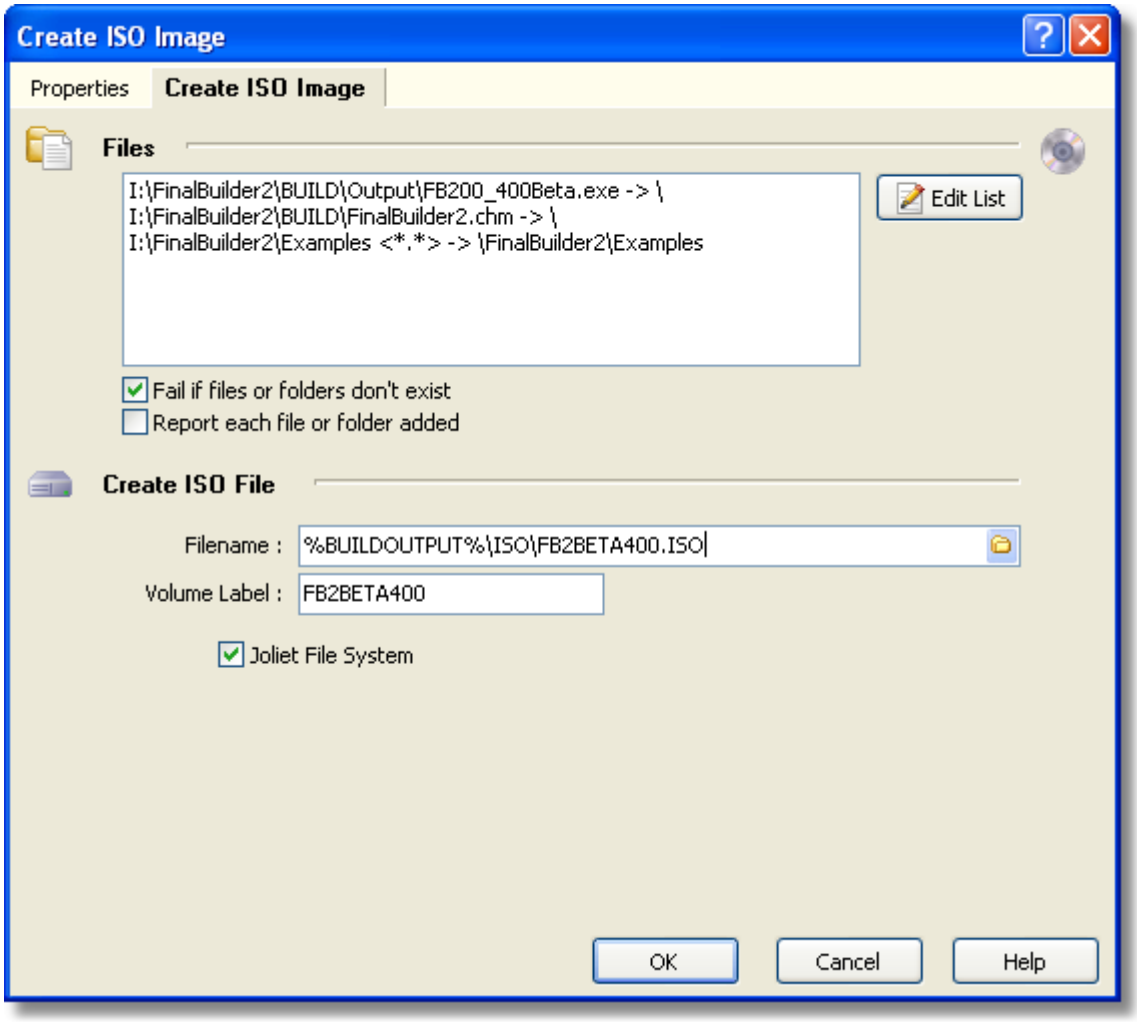
"Eject after write"

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.20.2 Create ISO Action

[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.



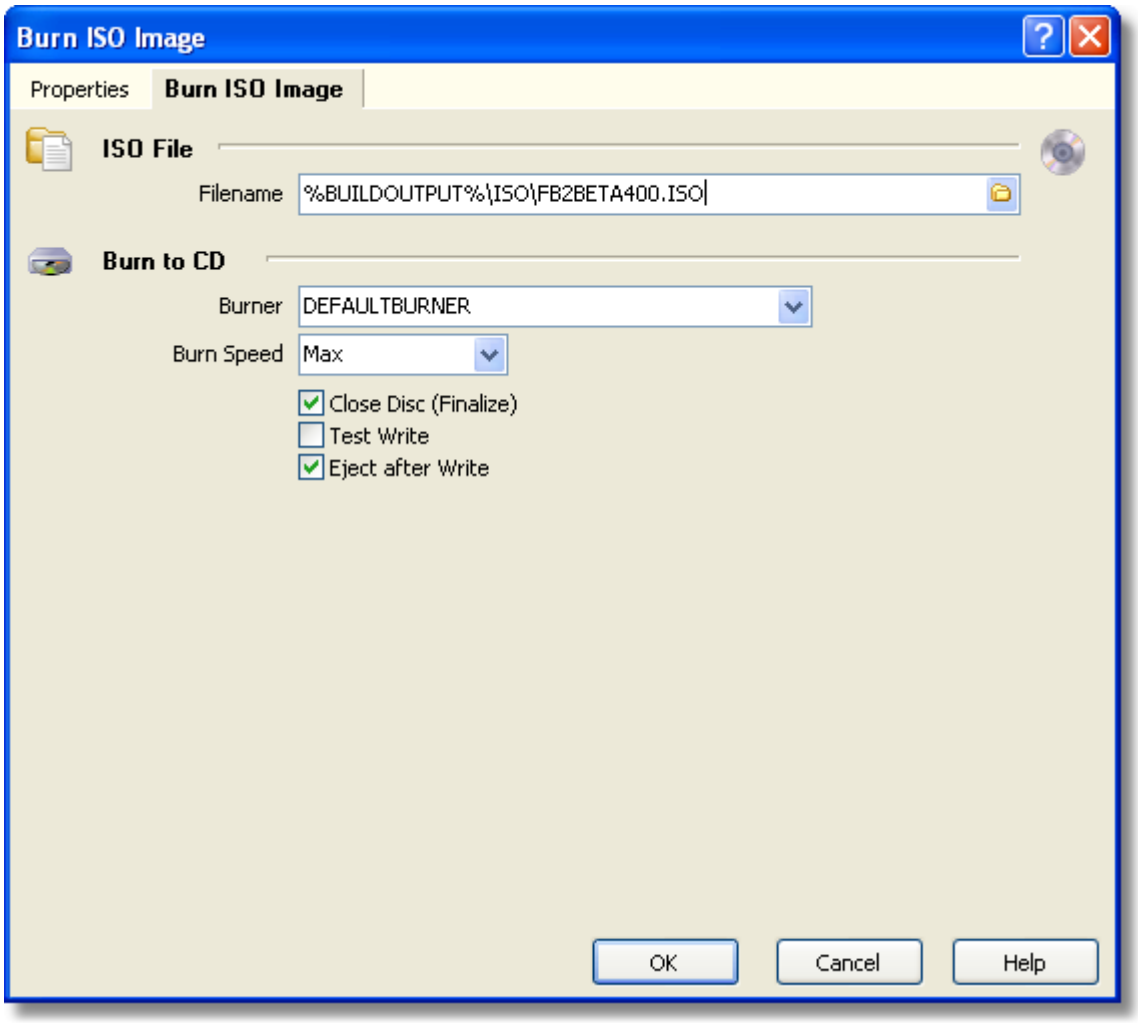
"Joliet File System"

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.20.3 Burn ISO Action

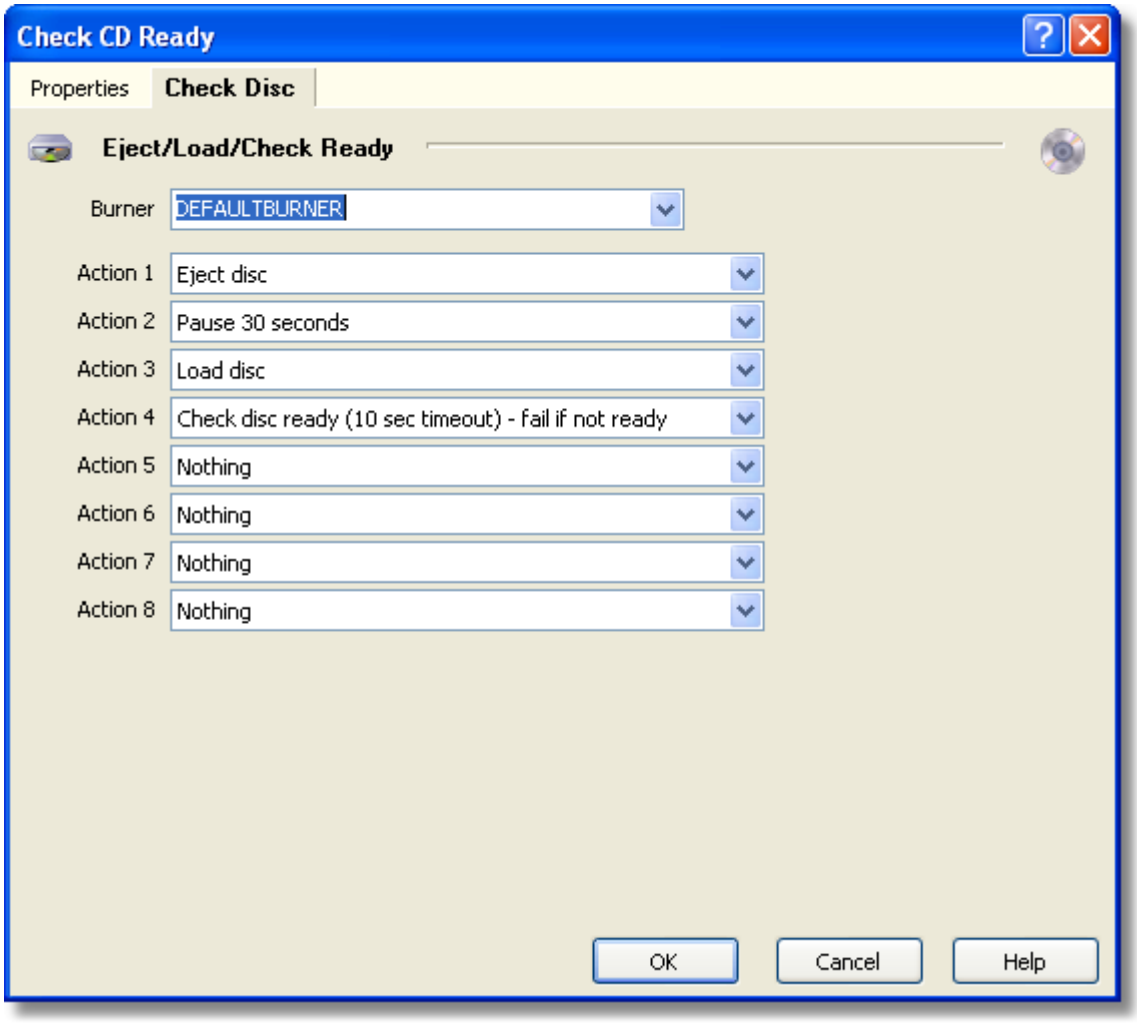
[Professional Edition]
This action supports burning an ISO file generated by FinalBuilder(or any other tool that generates valid ISO Files) to CDR/CDRW/DVDR & DVDRW Media (depending on your burner hardware). Note that to use this action you need to define an alias for your burner first (see here for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



© 2001-2006 VSoft Technologies Pty Ltd

5.20.4 Check Ready Action

[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).



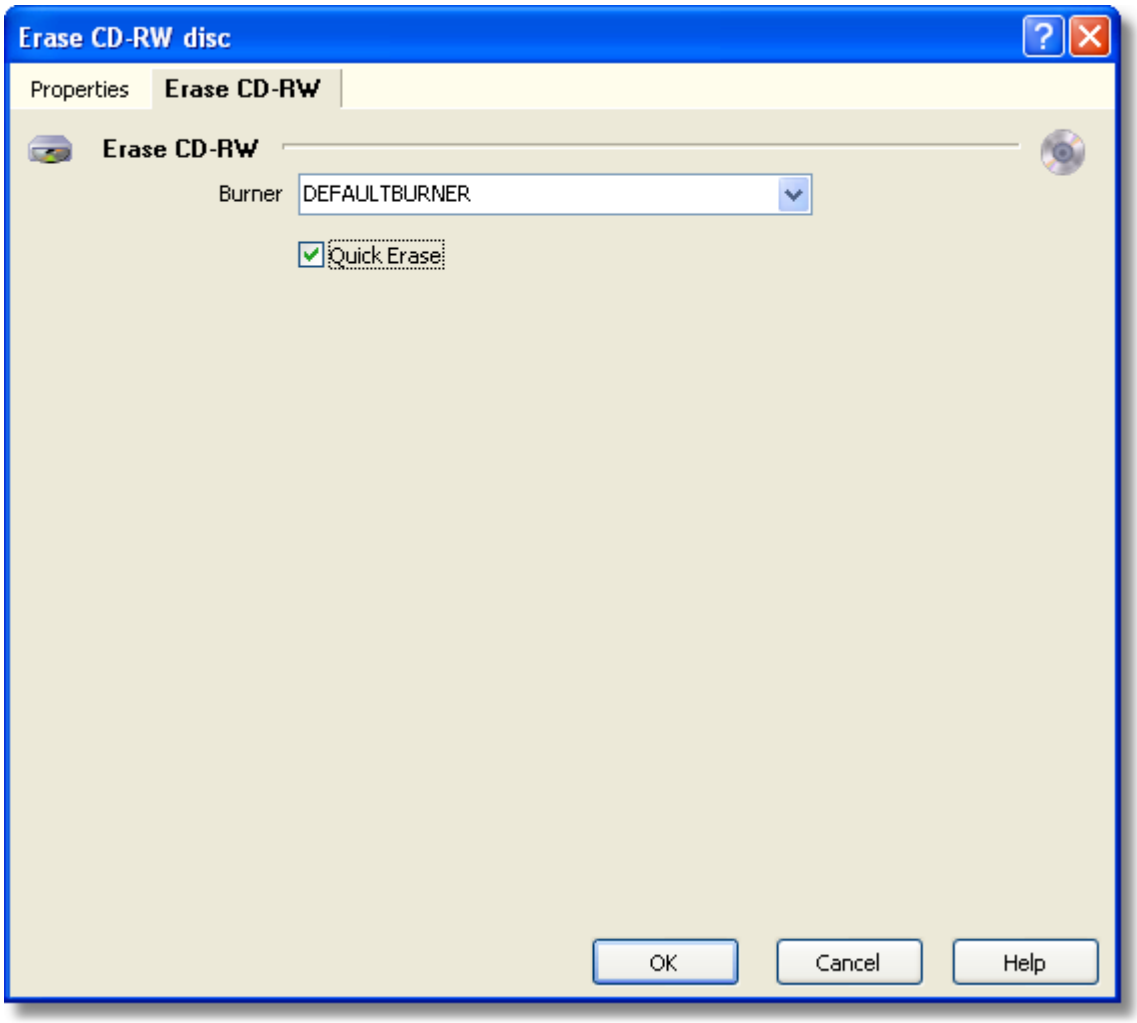
© 2001-2006 VSoft Technologies Pty Ltd

5.20.5 Erase CD/DVD RW

[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!).



© 2001-2006 VSoft Technologies Pty Ltd

5.21 .NET Actions

5.21.1 .Net Framework Tools

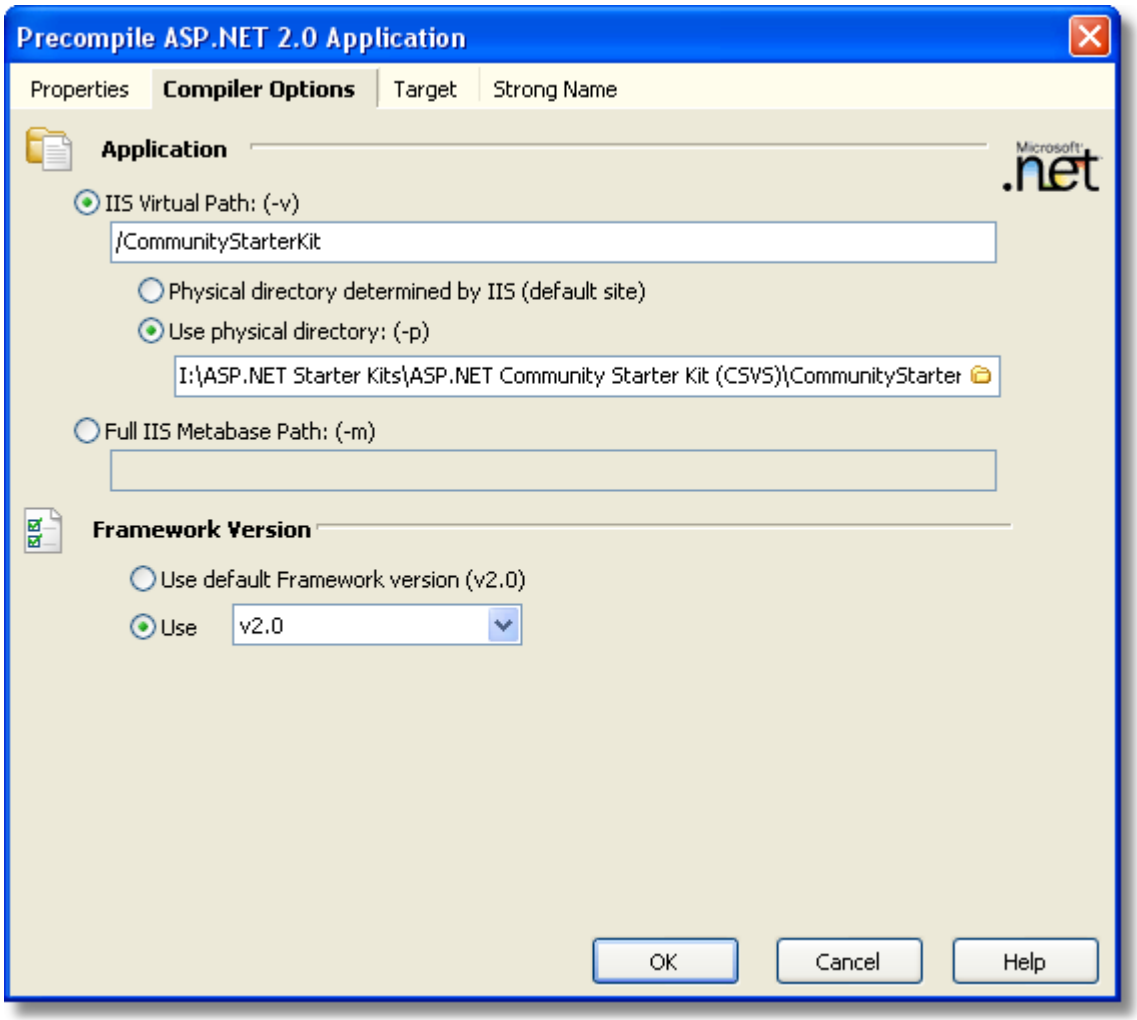
The .Net Framework tools are installed as part of the .Net framework (ie. if you have .Net installed, you have the framework tools)

5.21.1.1 Precompile ASP.NET 2.0 Application

The ASP.NET 2.0 compiler action allows you to precompile ASP.NET 2.0 applications before deployment.

Precompiling has two main advantages:

- The ASP application does not need to be compiled when first accessed.
- By deploying only compiled ASP applications, the ASPX source files do not need to be in the IIS virtual directories.



Application Path

There are three ways to specify the path to your ASP.NET Application:

© 2001-2006 VSoft Technologies Pty Ltd

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.

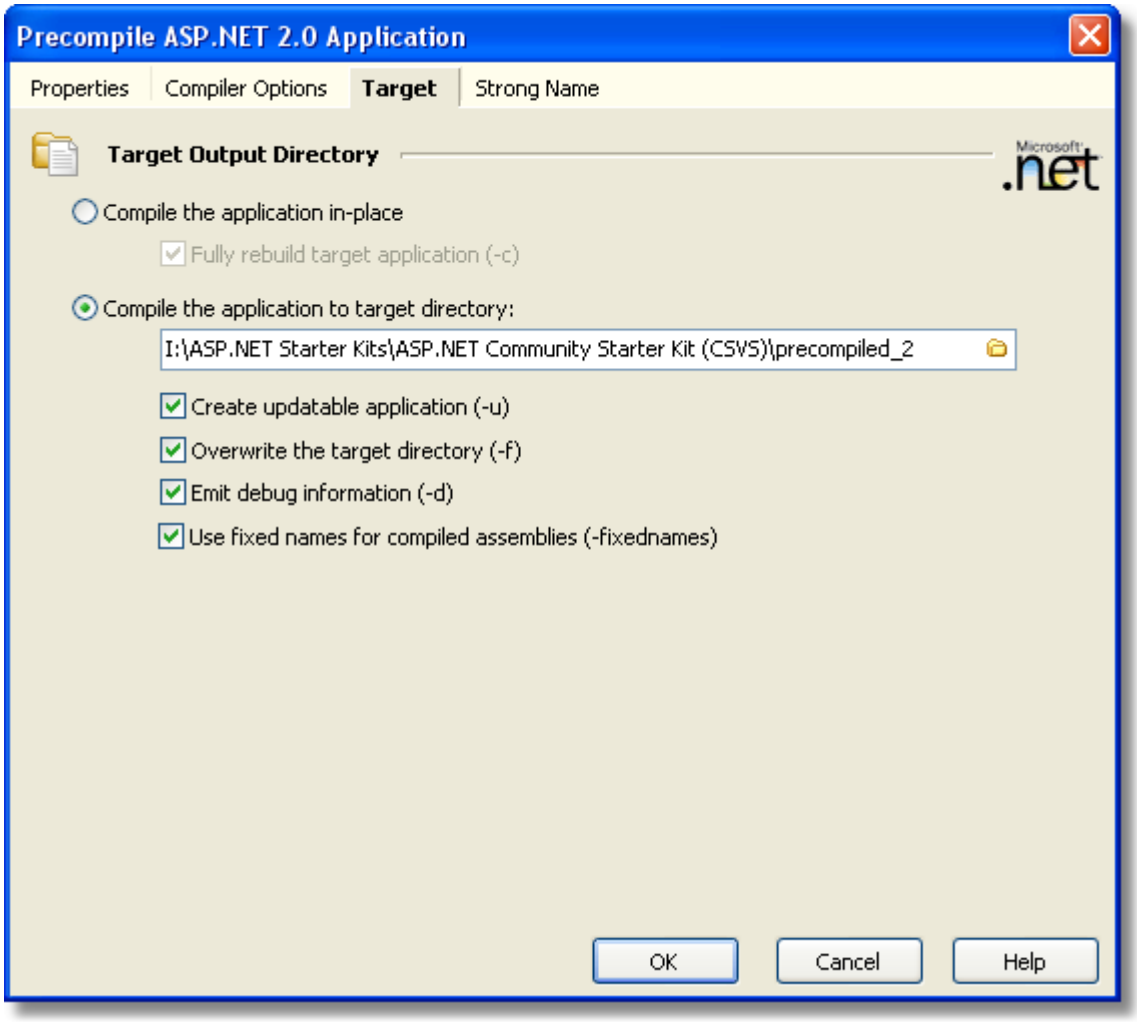
Full IIS Metabase Path

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

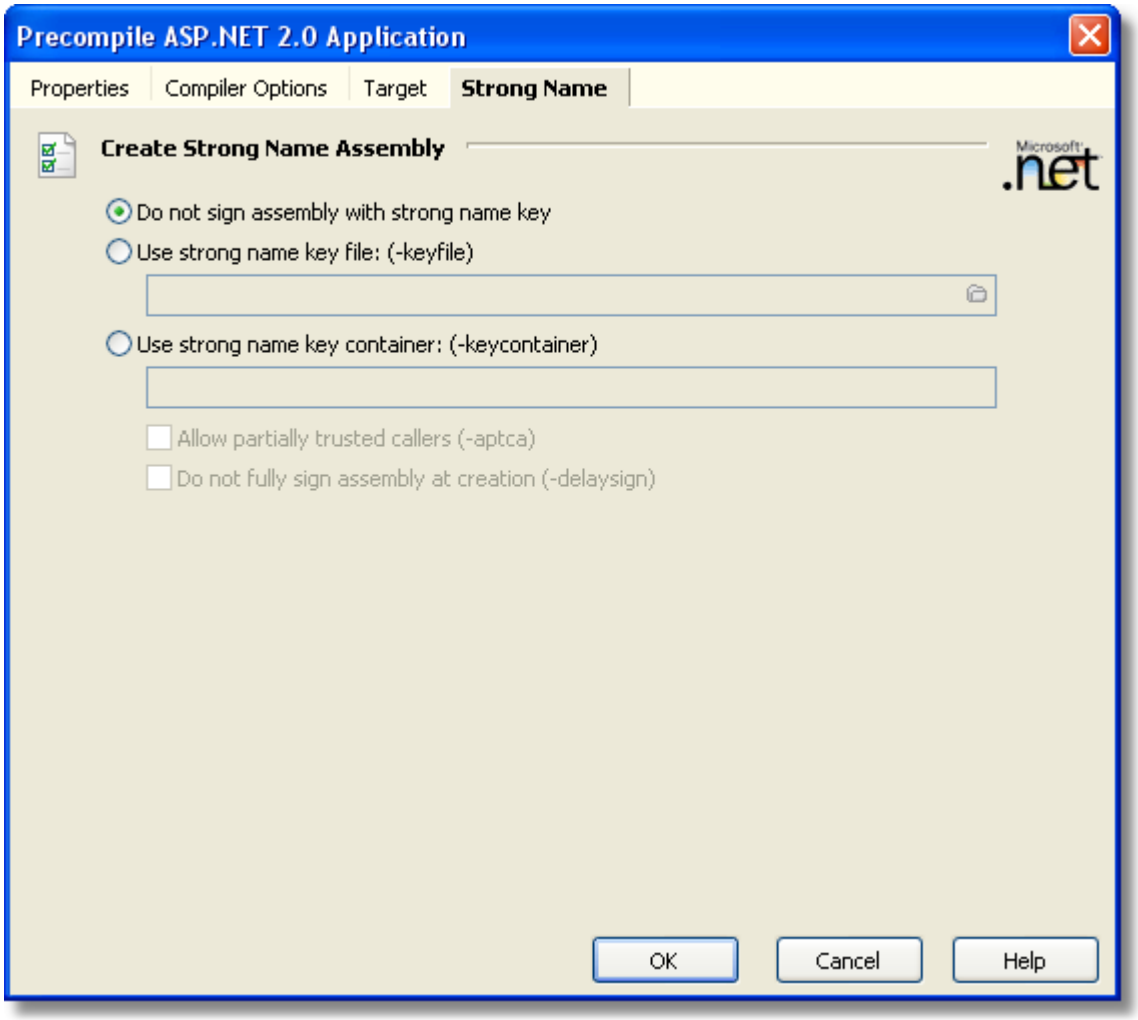
Emit debug information : The compiler will emit debug information to the log during the compile process.

Use fixed names for compiled assemblies : The compiled assemblies will be

© 2001-2006 VSoft Technologies Pty Ltd

given fixed names.

(If this option is not set, the names will be autogenerated.)

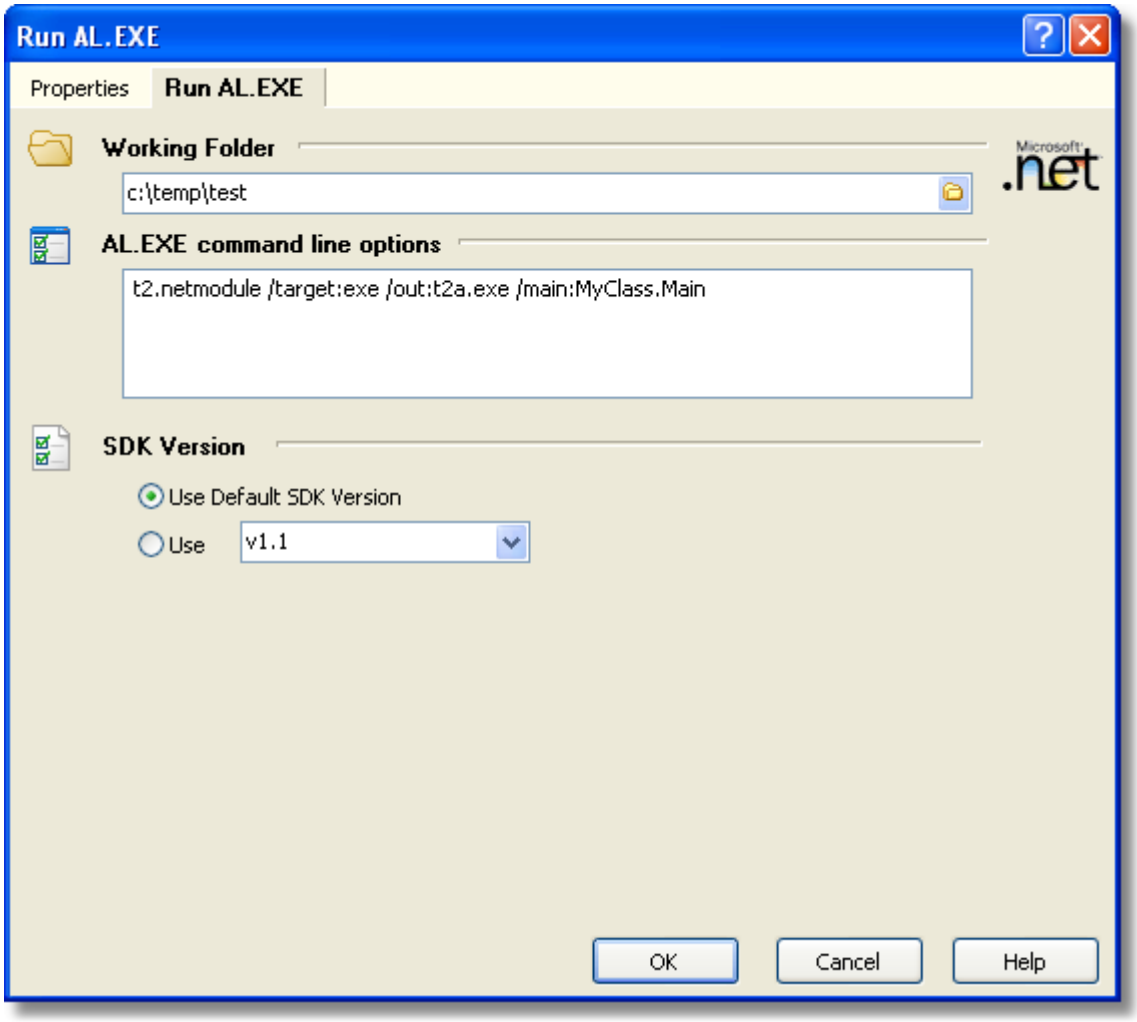


The compiled application assembly can be signed with a strong name from a key file or key container. If using a key container, the application can be set to allow partially trusted callers or to not fully sign the compiled assembly.

5.21.1.2 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.

© 2001-2006 VSoft Technologies Pty Ltd



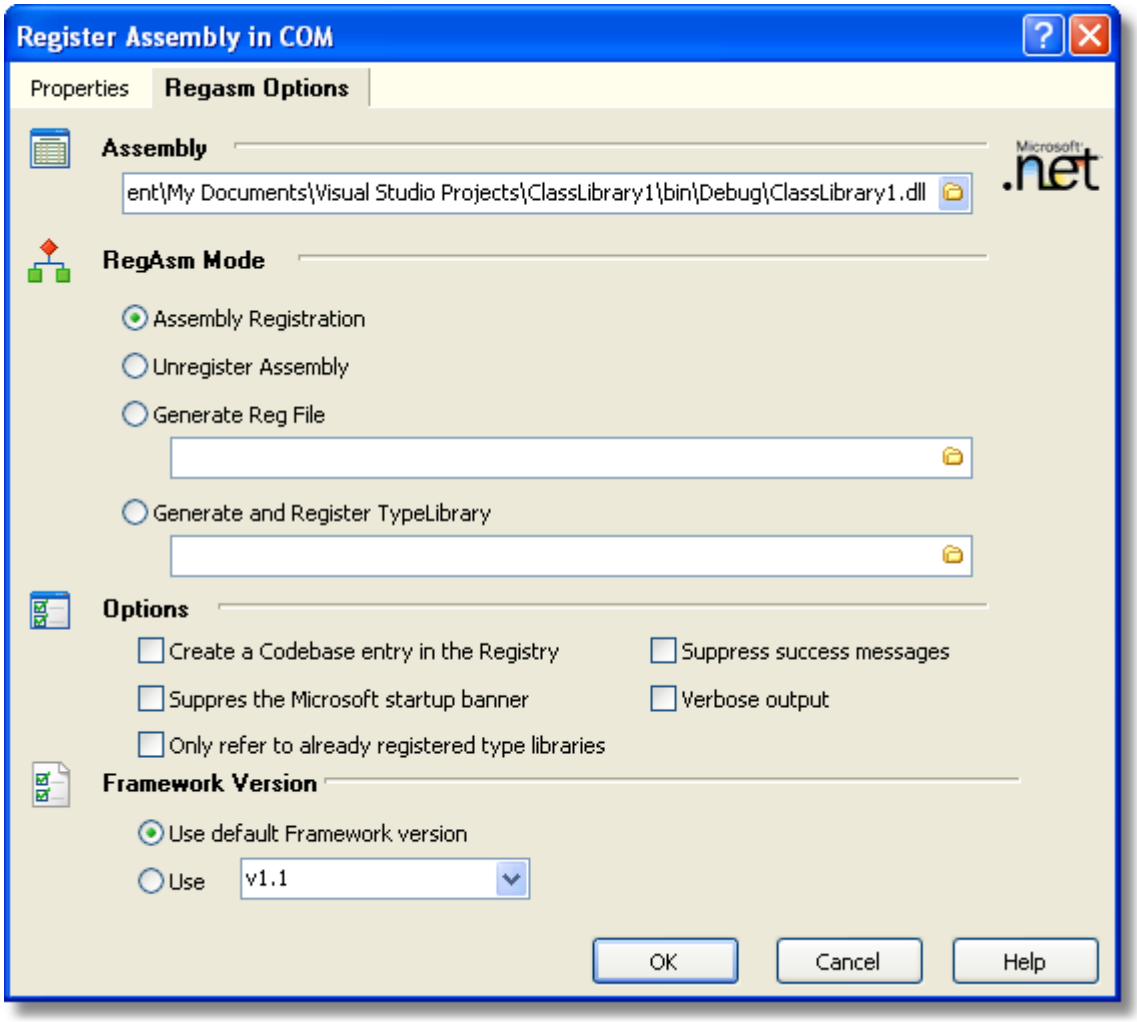
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfAssemblyGenerationUtilityAlexe.asp

5.21.1.3 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.

© 2001-2006 VSoft Technologies Pty Ltd



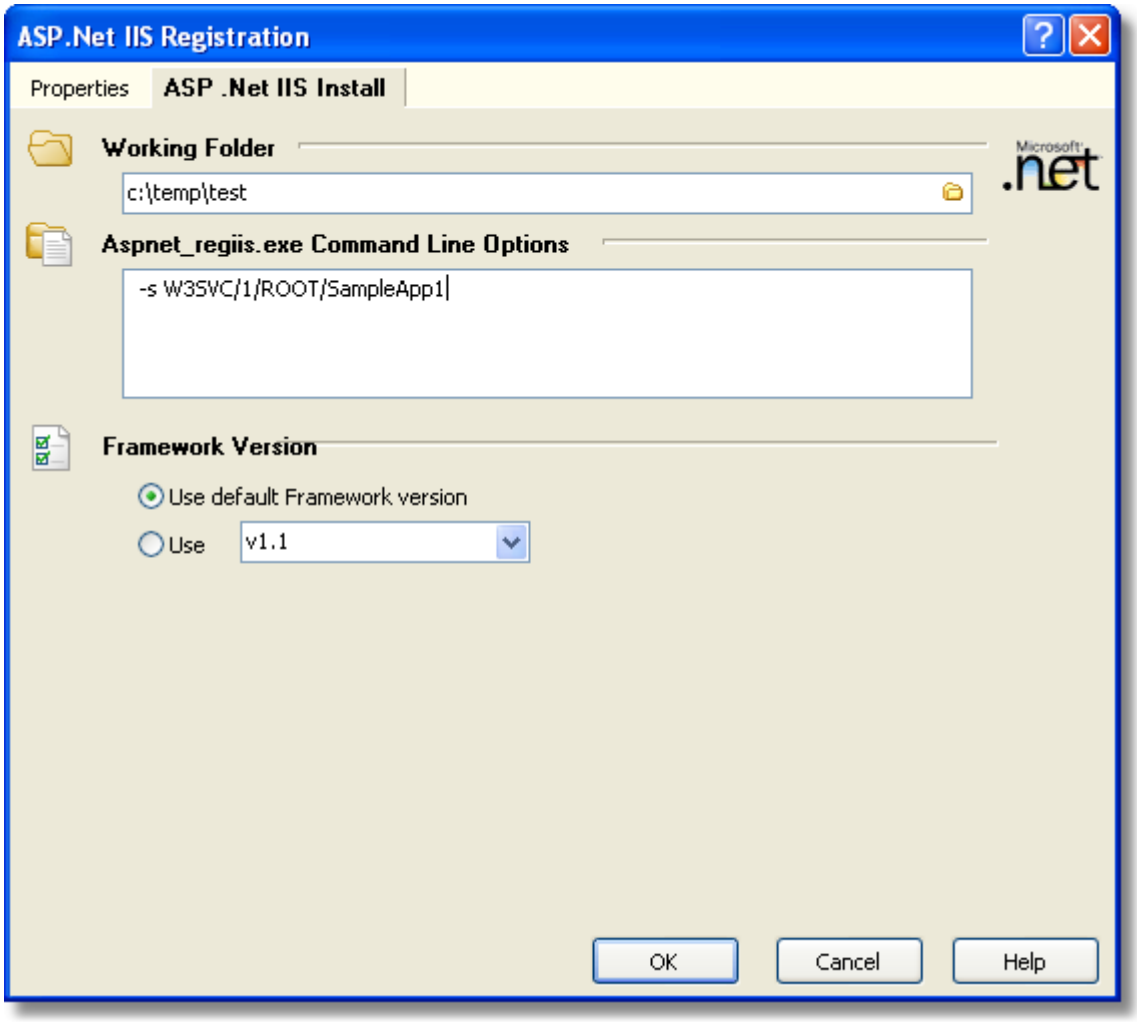
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfAssemblyRegistrationToolRegasmexe.asp

5.21.1.4 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.

© 2001-2006 VSoft Technologies Pty Ltd



For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfASPNETIISRegistrationToolAspnet_regiisexe.asp

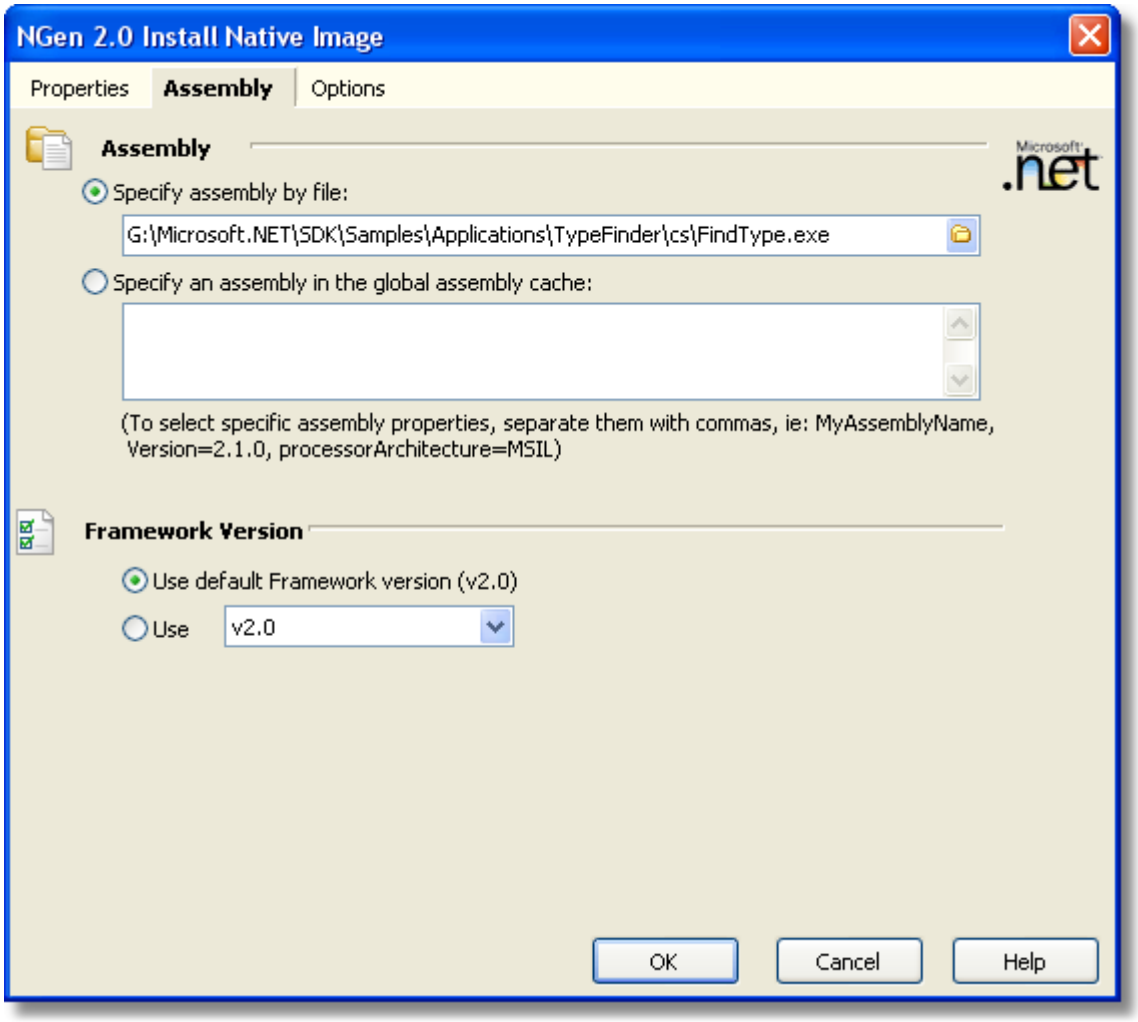
5.21.1.5 NGen 2.0 Native Image Install Action

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

© 2001-2006 VSoft Technologies Pty Ltd



Specify Assembly By File

Enter the path of an assembly to precompile.

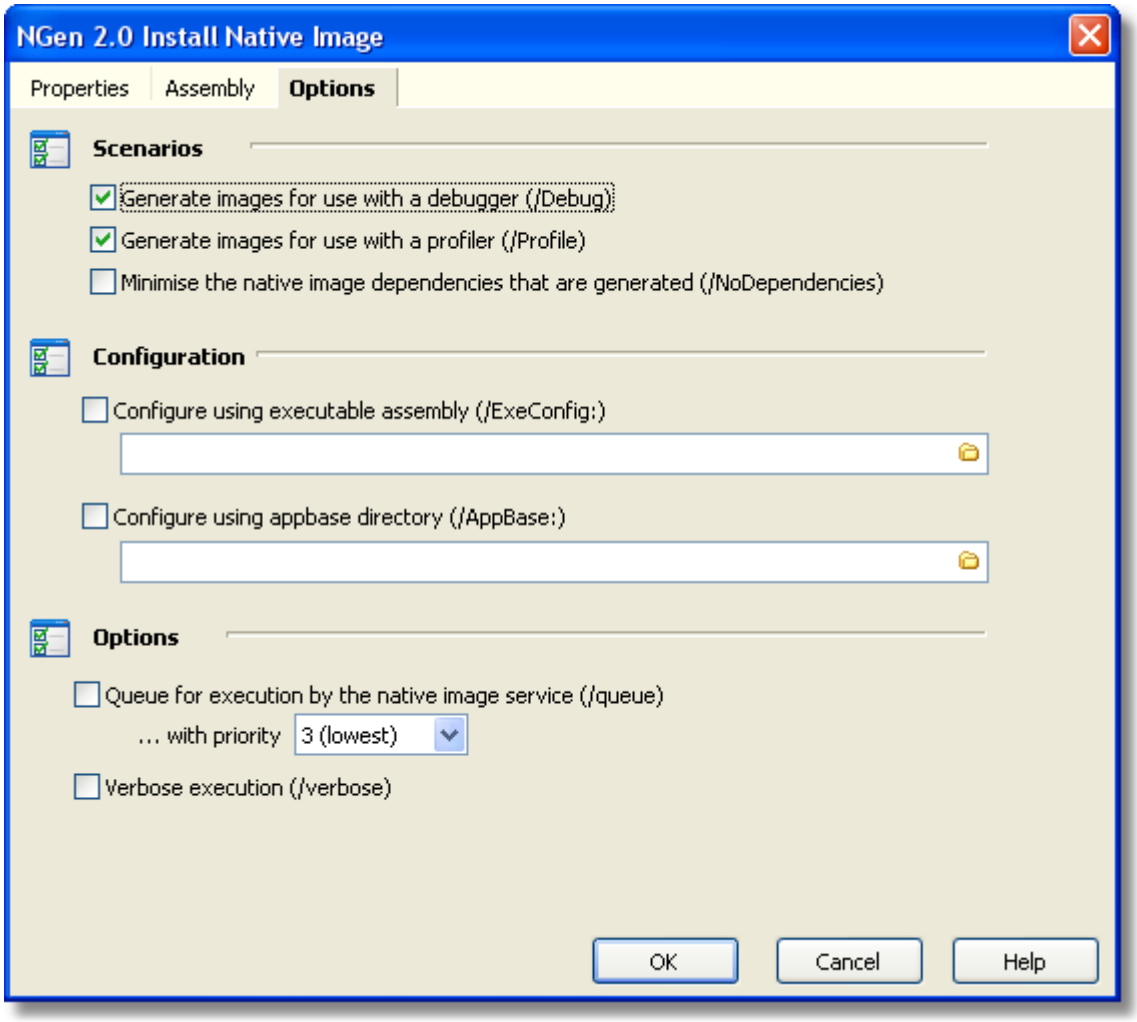
Specify an assembly in the global assembly cache

Specify the name of an assembly in the global assembly cache. Seperate extra properties with commas.

Framework Version

Specify the version of the .NET framework to use for NGen. The minimum is 2.0.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

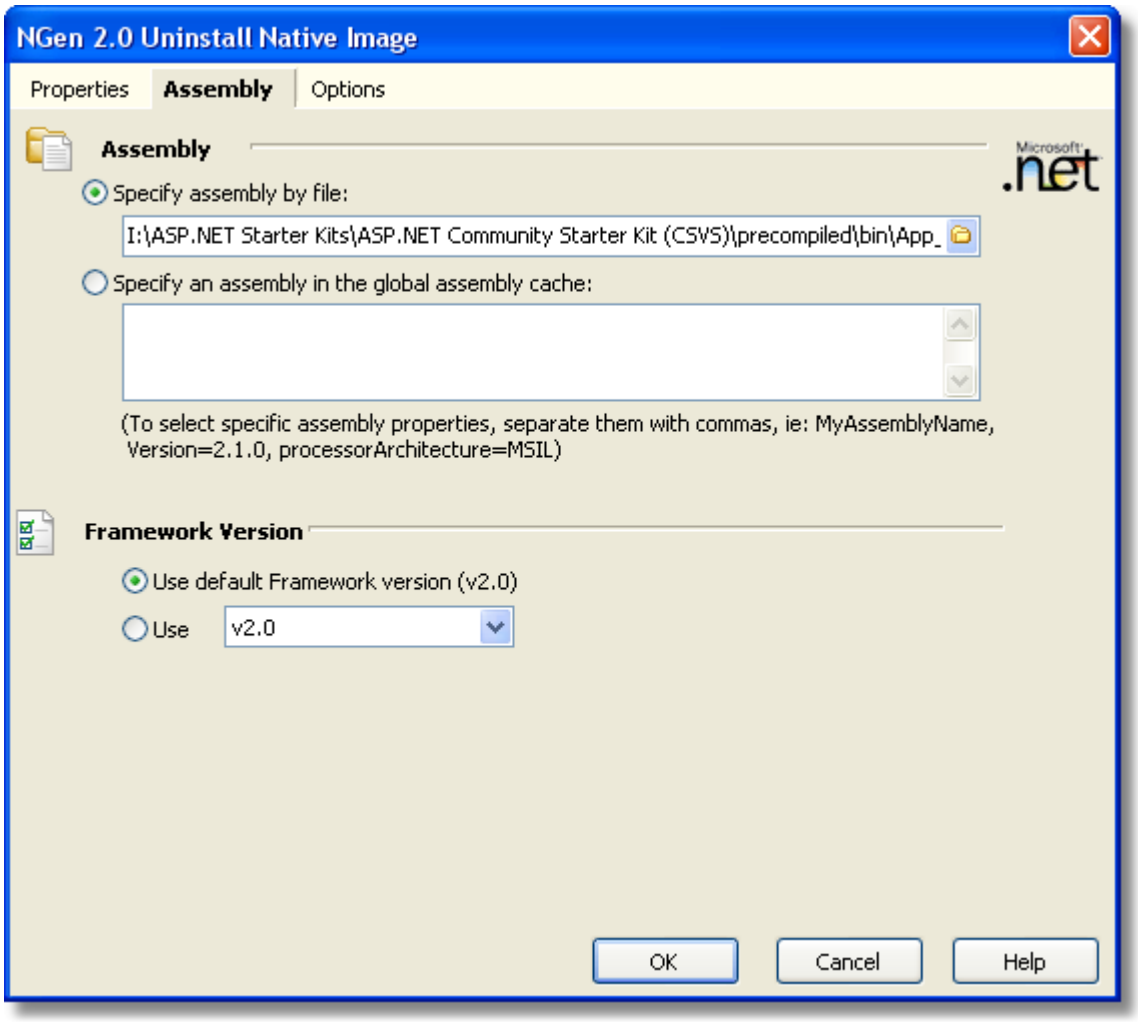
5.21.1.6 NGen 2.0 Native Image Uninstall Action

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



Specify Assembly By File

Enter the path of an assembly to remove.

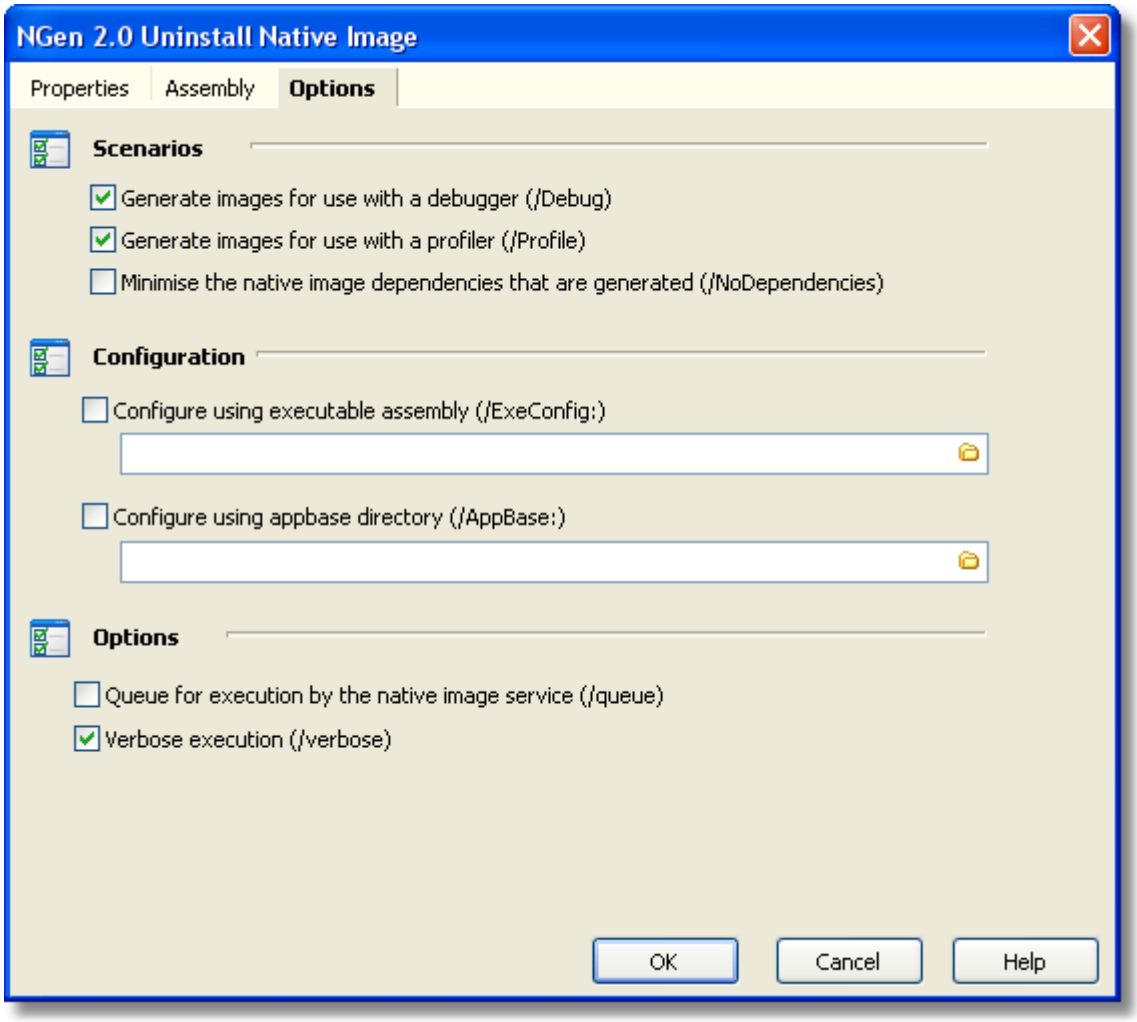
Specify an assembly in the global assembly cache

Specify the name of an assembly in the global assembly cache. Seperate extra properties with commas.

Framework Version

© 2001-2006 VSoft Technologies Pty Ltd

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.

5.21.1.7 NGen 2.0 Native Images Update Action

The NGen 2.0 Native Images Update Action updates any invalid images from the Native Images Cache.

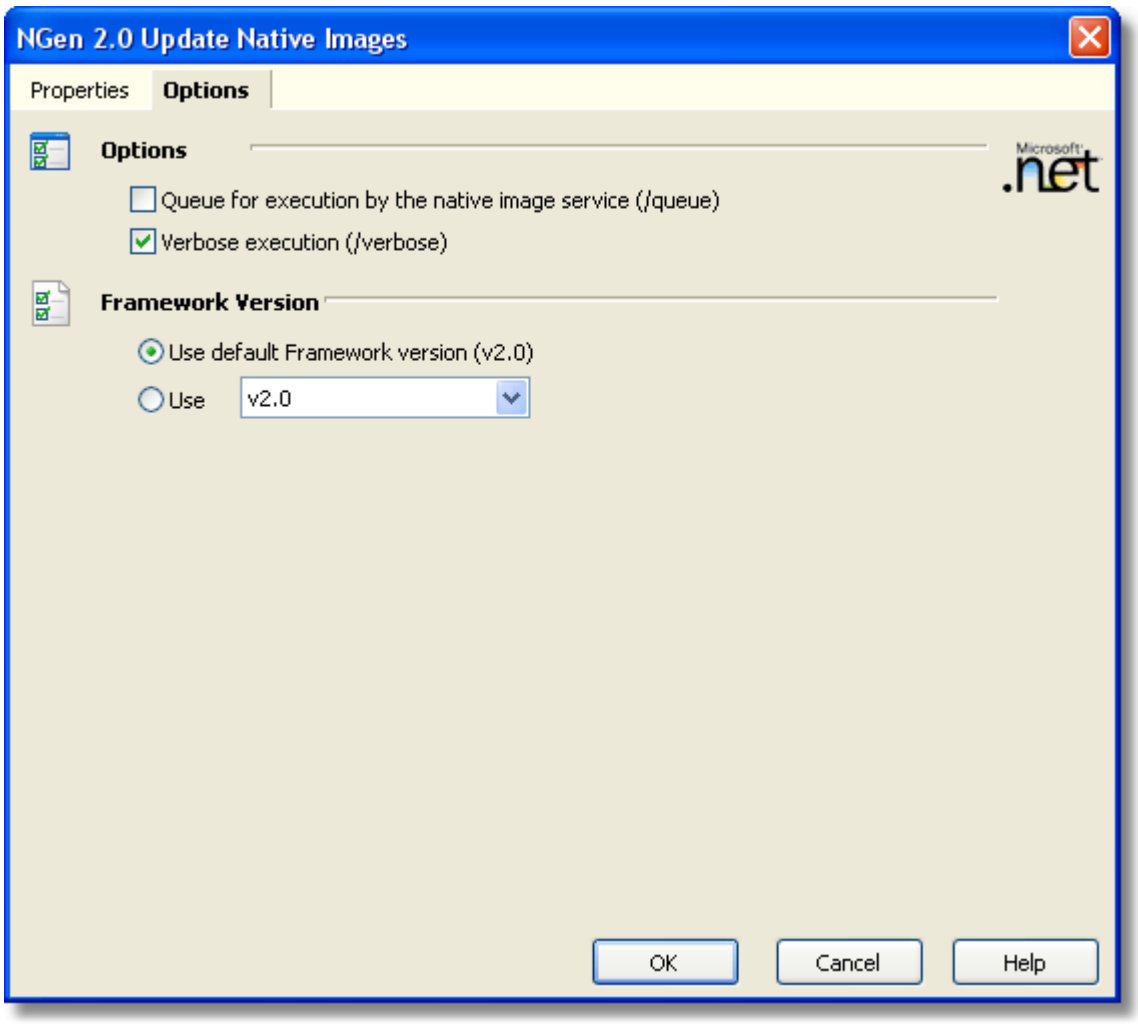
Updating invalid images is required because if an assembly image is out of date, the

© 2001-2006 VSoft Technologies Pty Ltd

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



Queue for execution...

Rather than being updated immediately, the updates will be queued for execution by the Native Image Service.

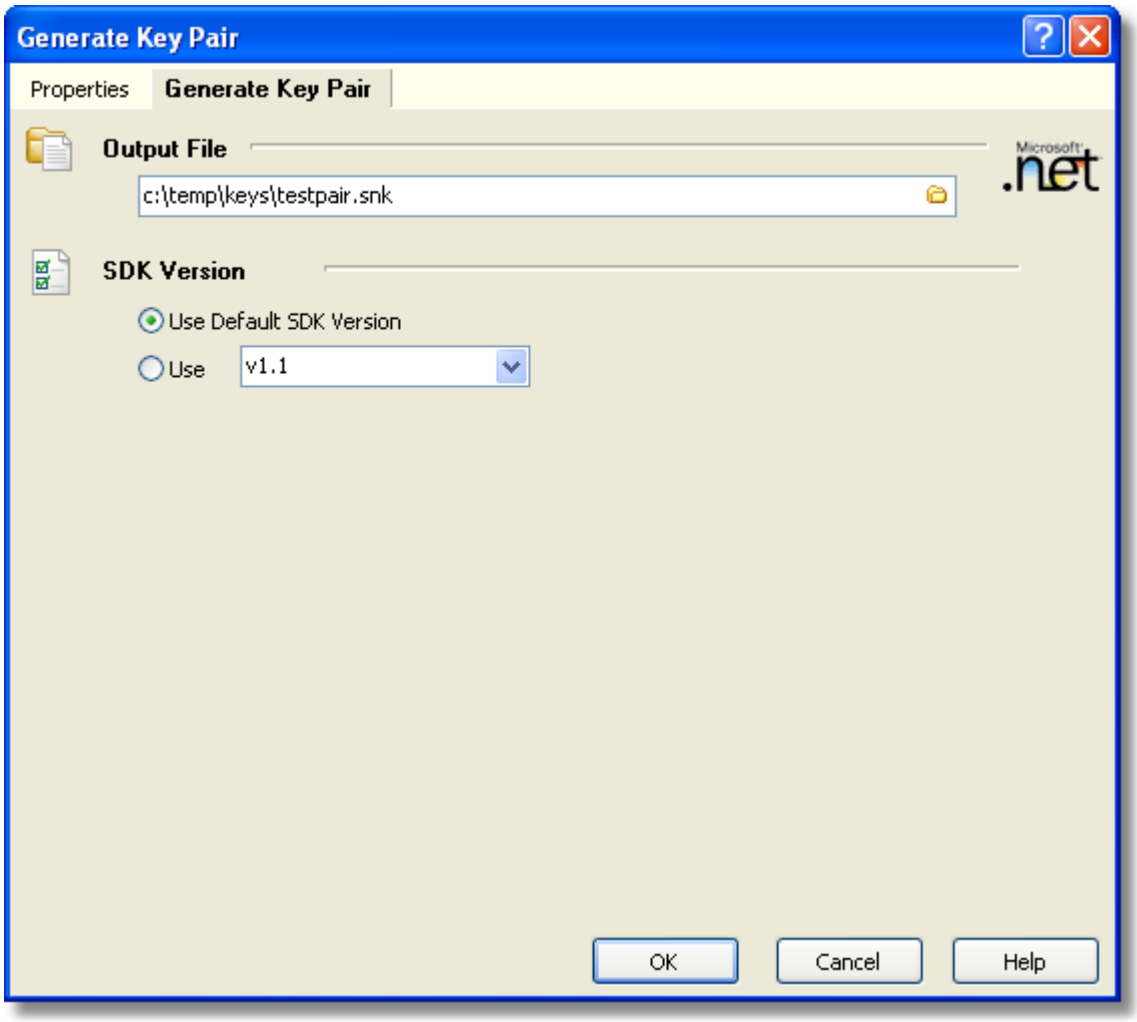
© 2001-2006 VSoft Technologies Pty Ltd

5.21.2 .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

5.21.2.1 Generate Key Pair [SN]

Generates a new key pair and writes it to the specified file.



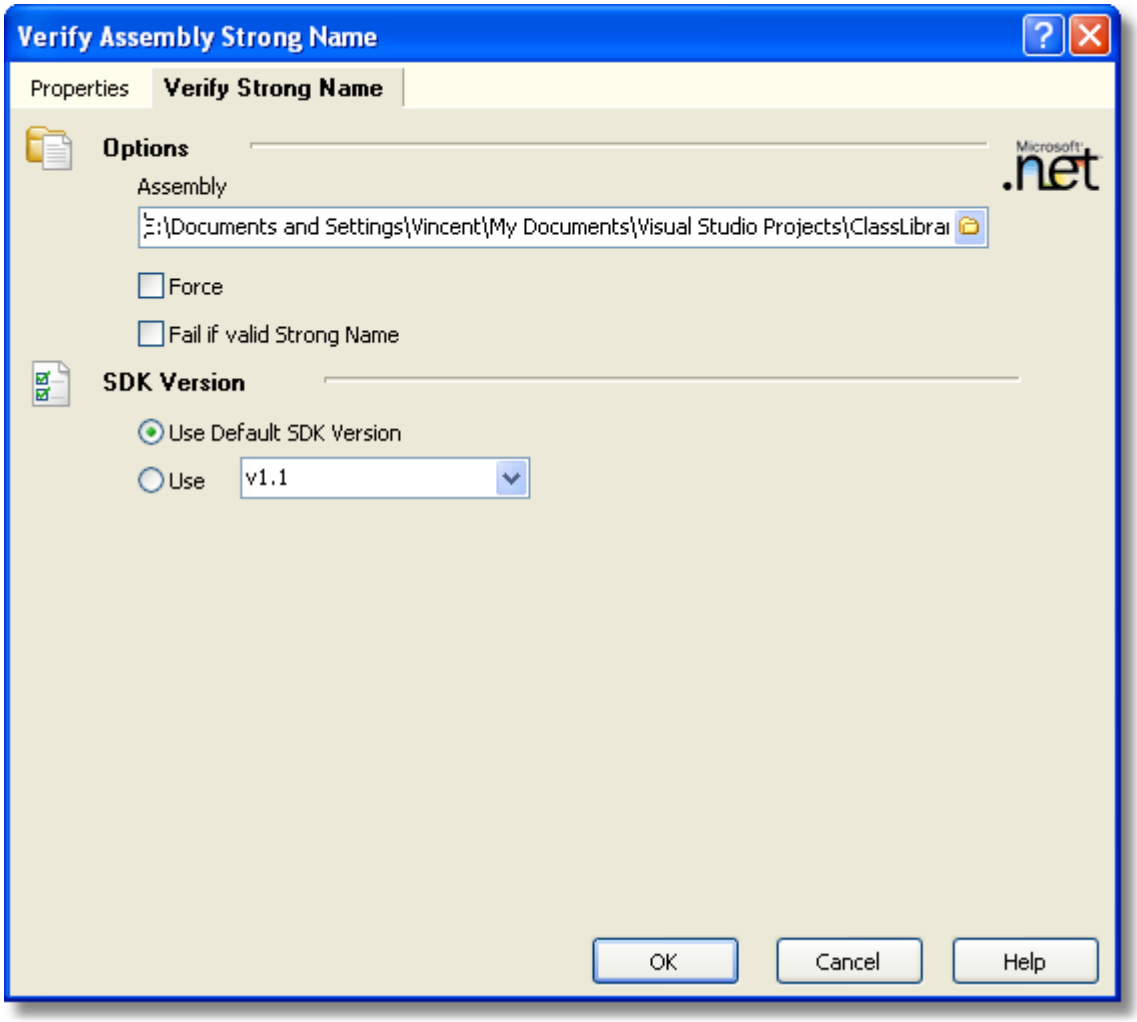
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp

5.21.2.2 Verify Strong Name [SN]

Verifies the strong name in assembly, where assembly is the name of a file that contains an assembly manifest.

© 2001-2006 VSoft Technologies Pty Ltd



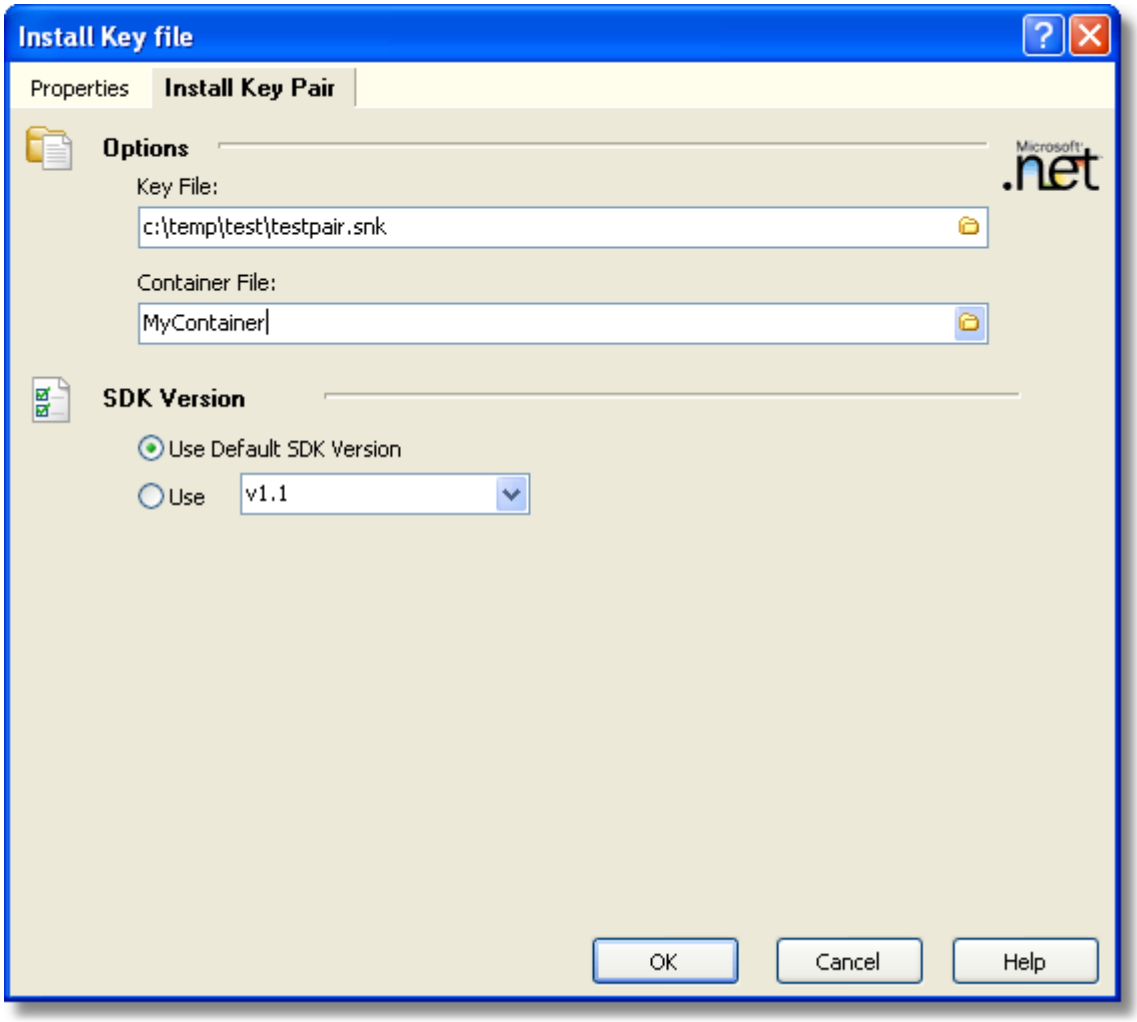
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp

5.21.2.3 Install Key in Container [SN]

Installs the key pair from infile in the specified key container. The key container resides in the strong name CSP.

© 2001-2006 VSoft Technologies Pty Ltd



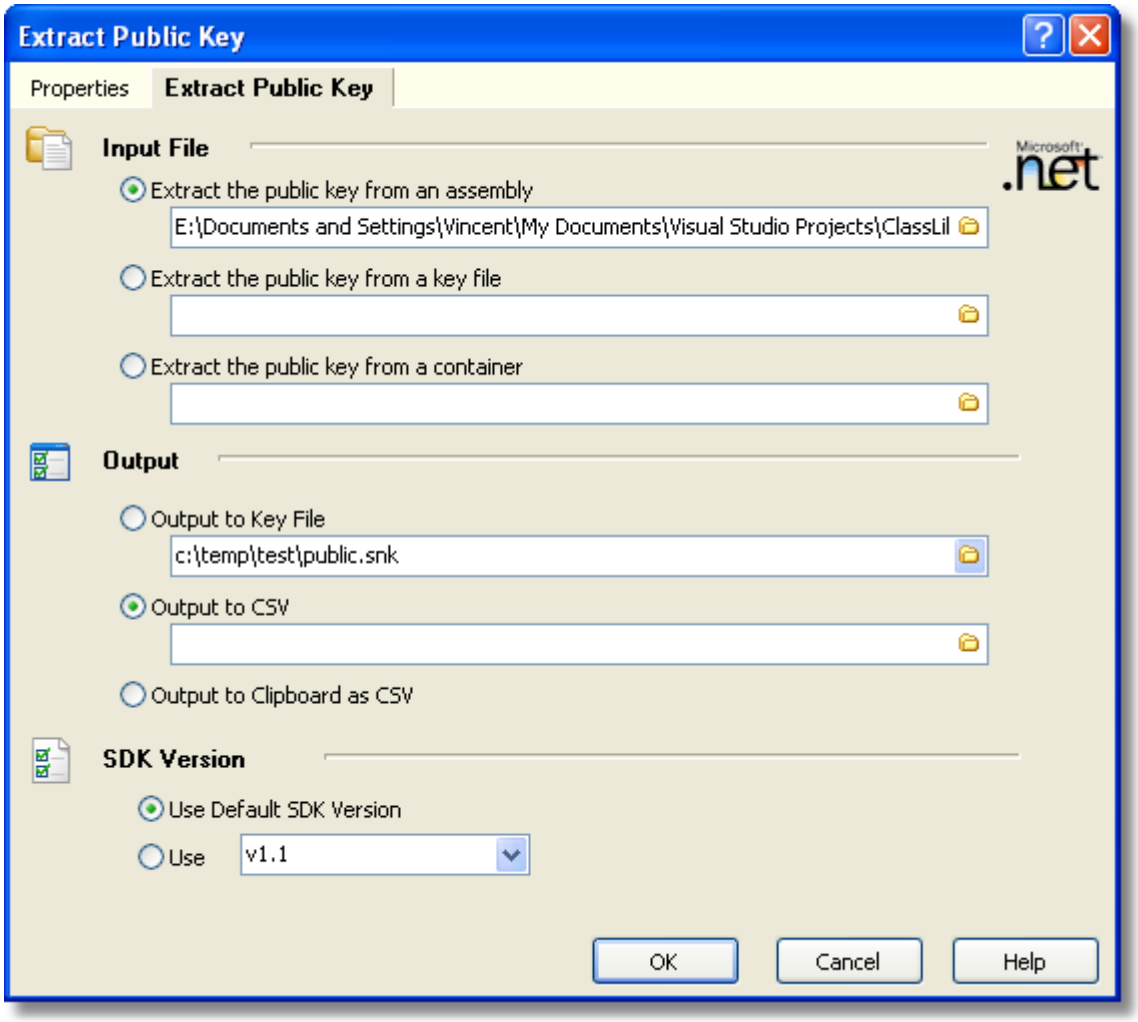
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp

© 2001-2006 VSoft Technologies Pty Ltd

5.21.2.4 Extract Public Key [SN]

Extracts the public key from assembly and stores it in outfile.



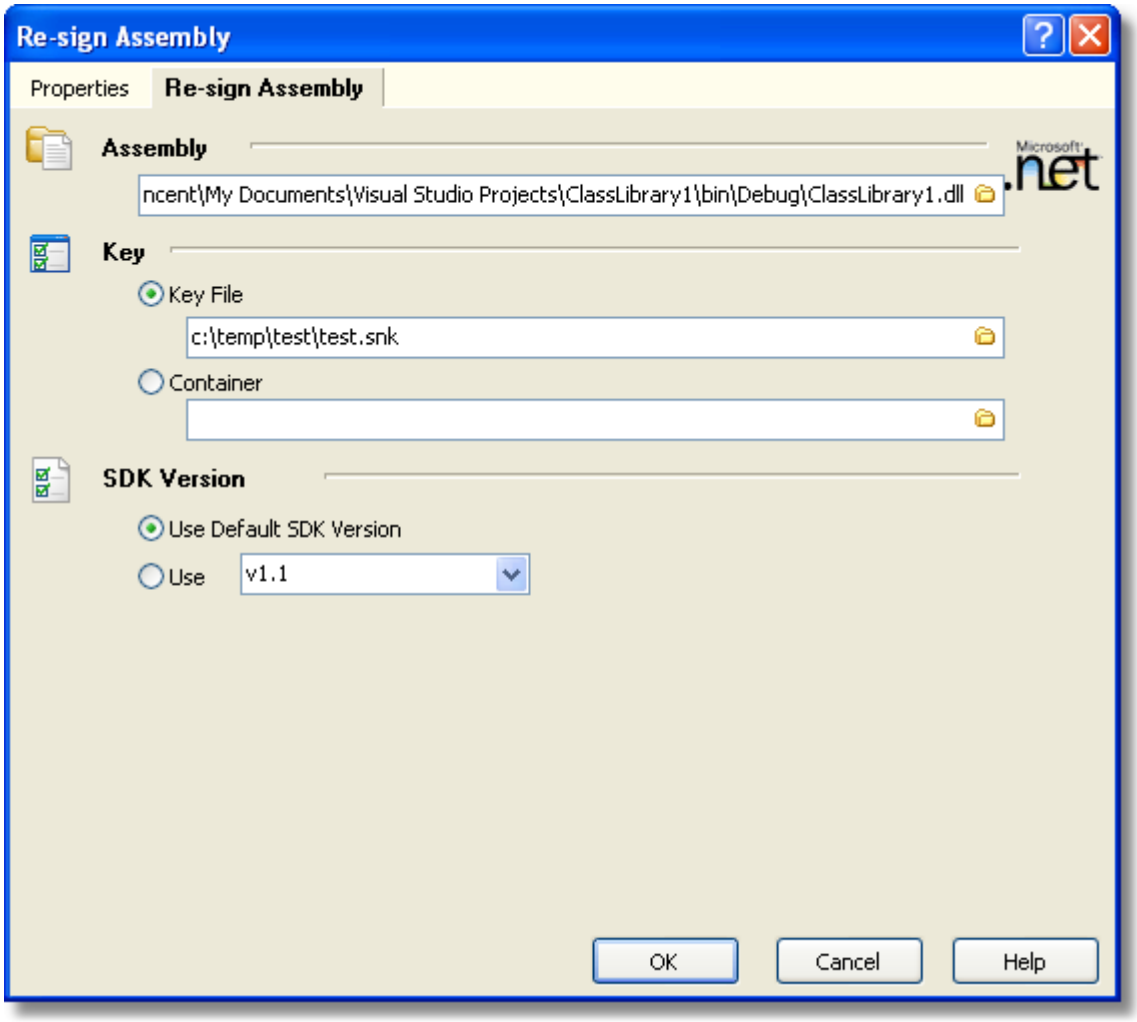
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp

© 2001-2006 VSoft Technologies Pty Ltd

5.21.2.5 Re-sign Assembly [SN]

Re-signs a previously signed or delay signed assembly.



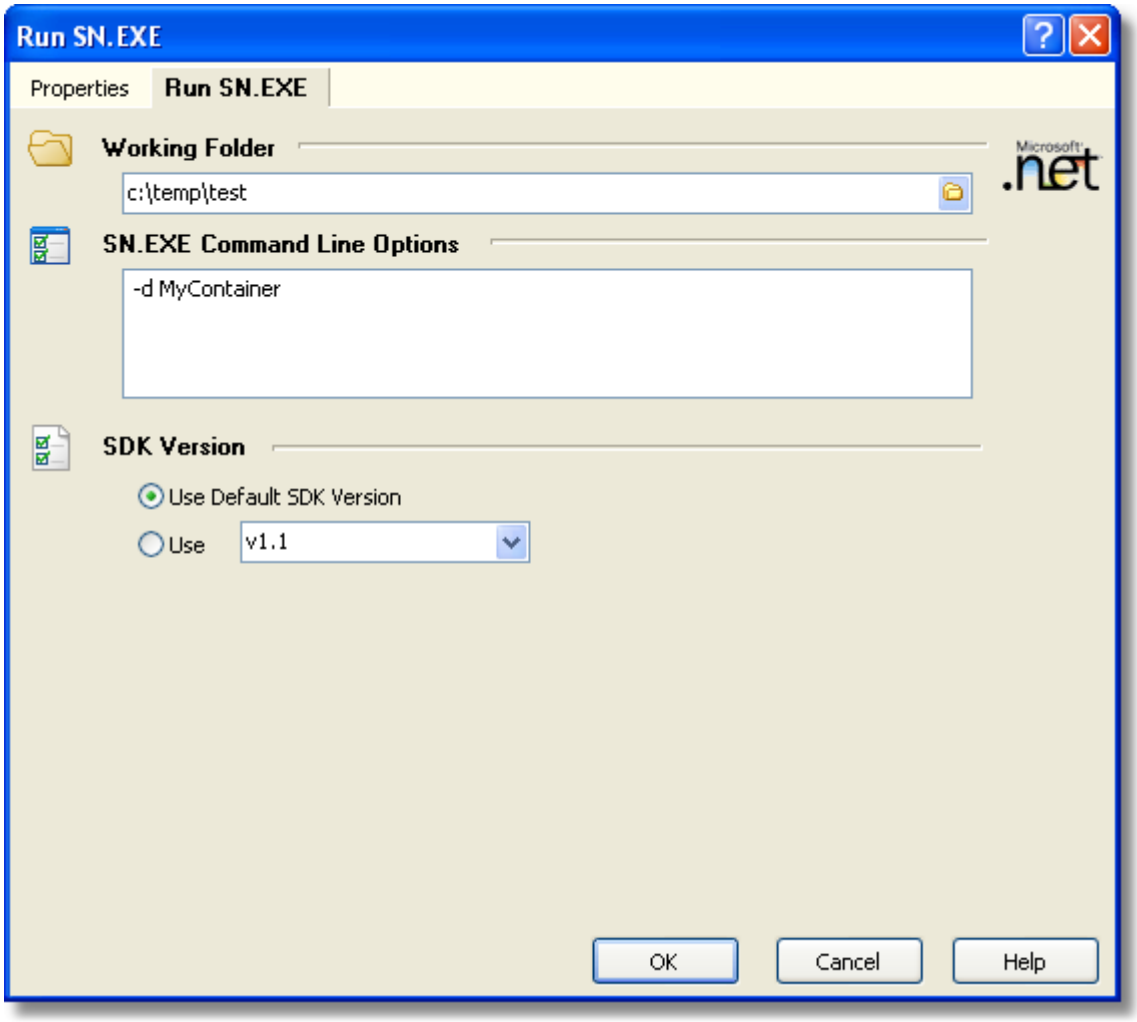
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp

5.21.2.6 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.

© 2001-2006 VSoft Technologies Pty Ltd



For more information see:

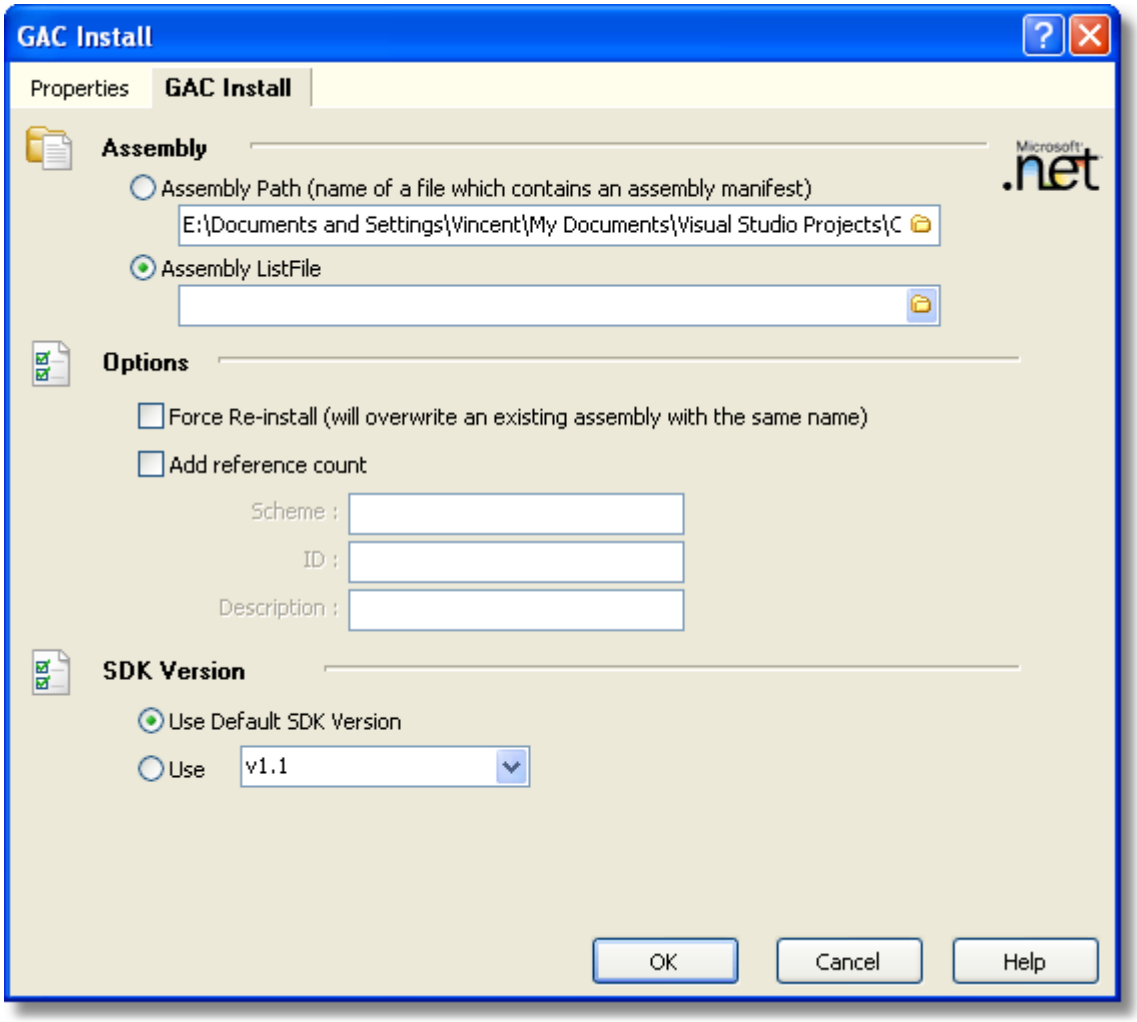
http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfStrongNameUtilitySNexe.asp

5.21.2.7 GAC Install [GACUTIL]

Installs an assembly into the global assembly cache.

This action automates the following options:
/i
/if
/il
/ir

© 2001-2006 VSoft Technologies Pty Ltd



For more information see:

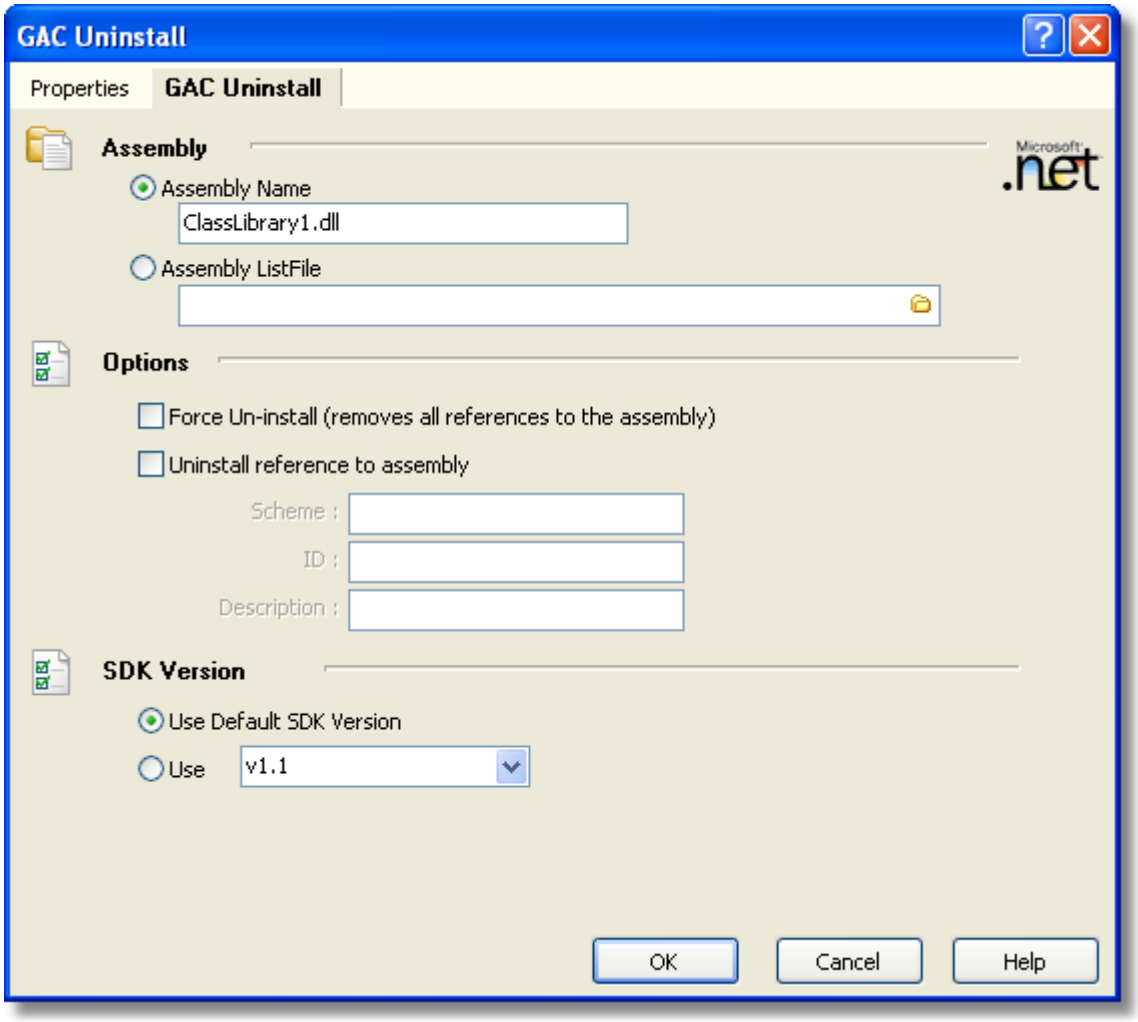
http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutilexe.asp

5.21.2.8 GAC Uninstall [GACUTIL]

Uninstalls an assembly from the global assembly cache.

This action automates the following options:
/u
/uf
/ul
/ur

© 2001-2006 VSoft Technologies Pty Ltd



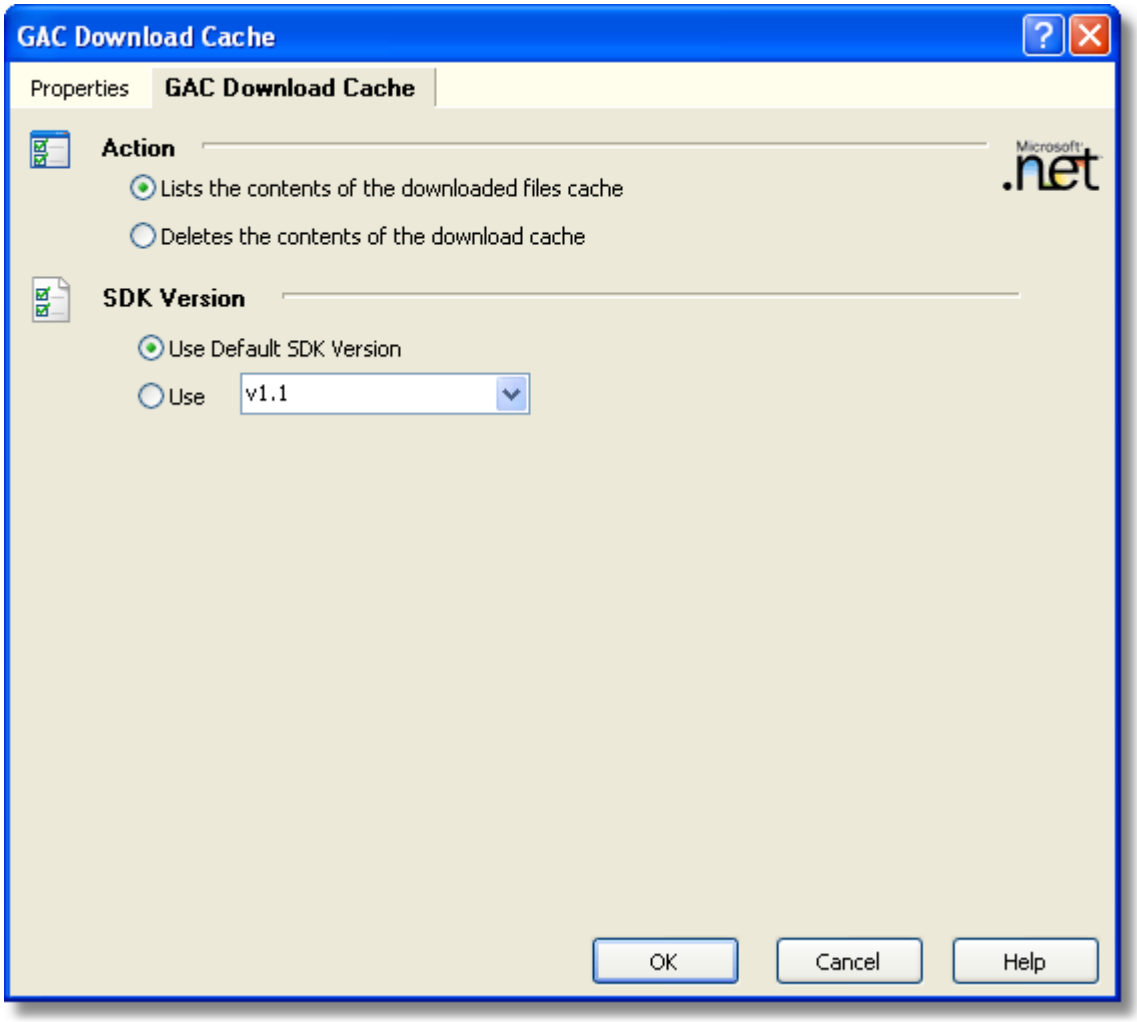
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutilexe.asp

© 2001-2006 VSoft Technologies Pty Ltd

5.21.2.9 GAC Download Cache [GACUTIL]

Lists or deletes the contents of the downloaded files cache.



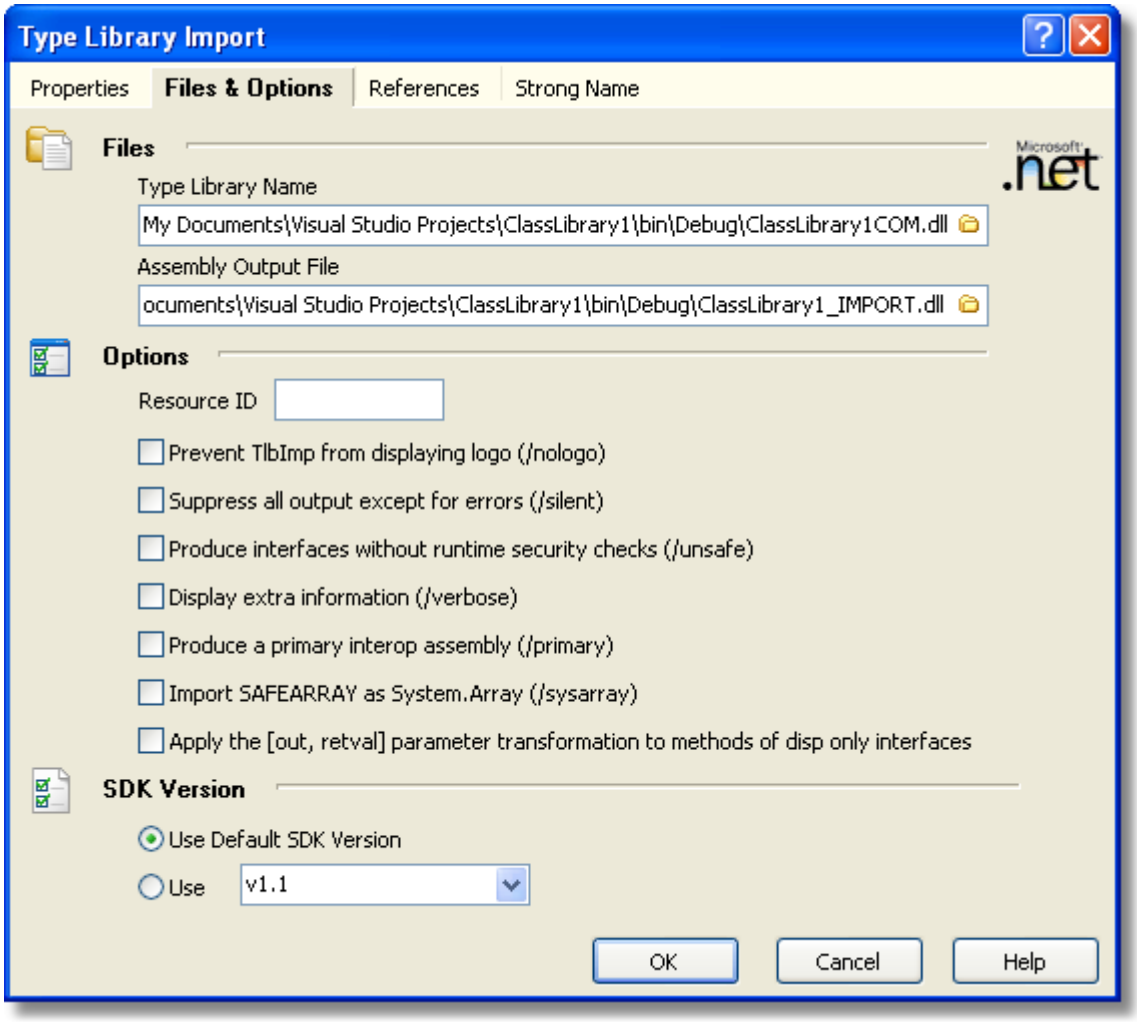
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfGlobalAssemblyCacheUtilityGacutilexe.asp

5.21.2.10 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.

© 2001-2006 VSoft Technologies Pty Ltd



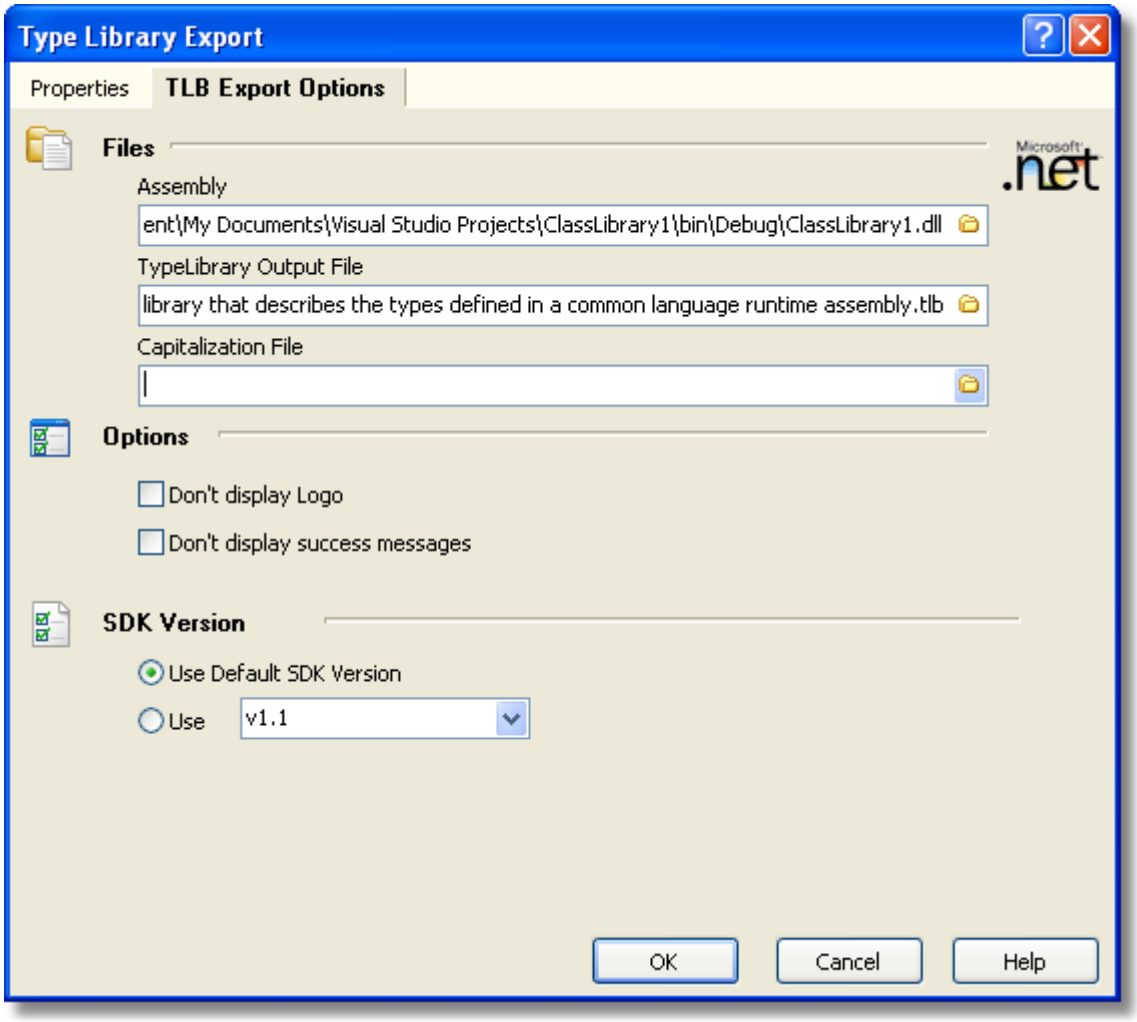
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfTypeLibraryImporterTlbimpexe.asp

5.21.2.11 Type Library Export [TLBEXP]

The Type Library Exporter generates a type library that describes the types defined in a common language runtime assembly.

© 2001-2006 VSoft Technologies Pty Ltd



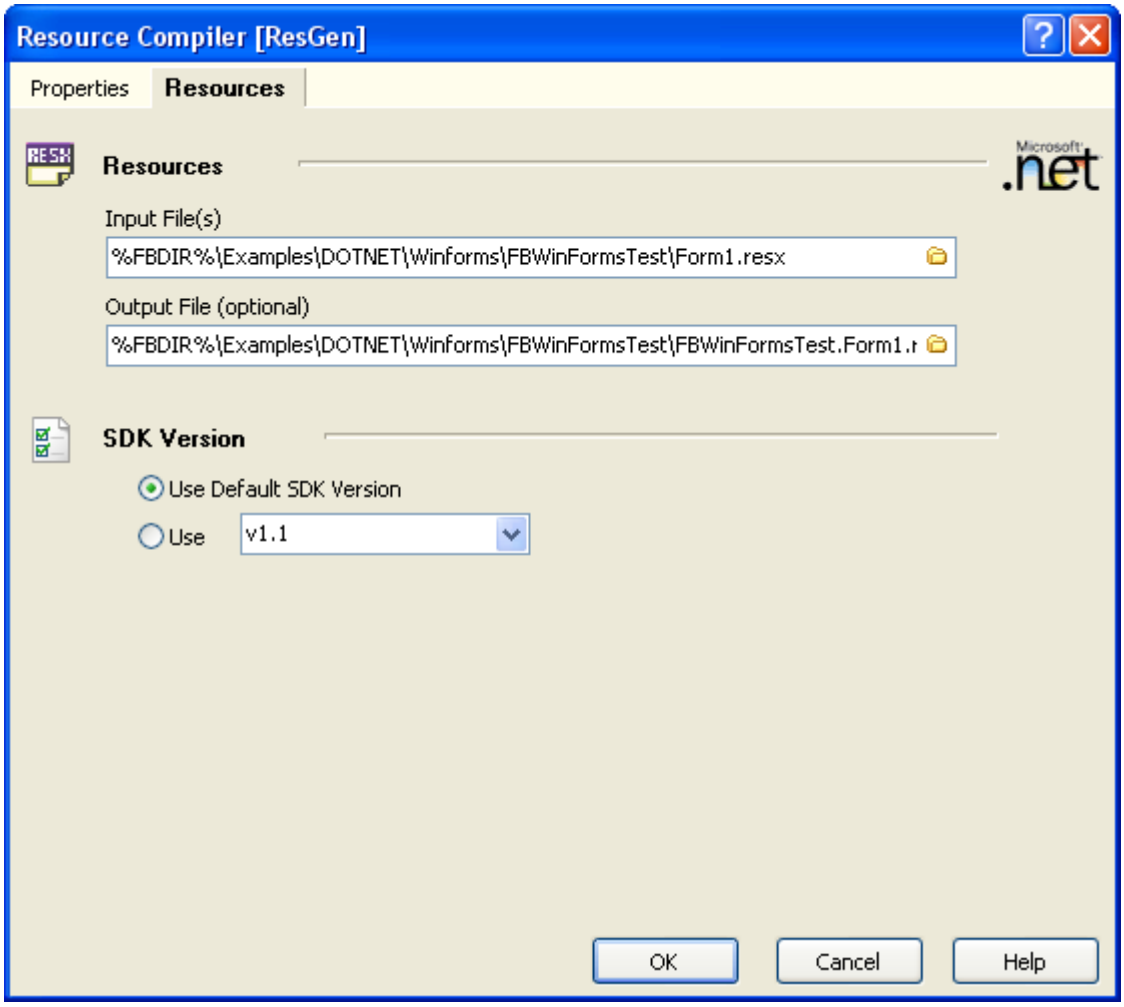
For more information see:

http://msdn.microsoft.com/library/en-us/cptools/html/cpgrfTypeLibraryExporterTlbExpexe.asp

5.21.2.12 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.

© 2001-2006 VSoft Technologies Pty Ltd



For more information see :

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cptools/html/cpgrfresourcefilegeneratorutilityresgenexe.asp

© 2001-2006 VSoft Technologies Pty Ltd

5.21.3 3rd Party Tools

Some .Net tools have been placed in other categories:

NDoc Action

NUnit Action

MSTest

Nant Project Action

MSBuild Project Action

5.21.3.1 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.gotdotnet.com/team/fxcop/

5.21.3.2 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 indefinately 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/

© 2001-2006 VSoft Technologies Pty Ltd

5.21.3.3 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

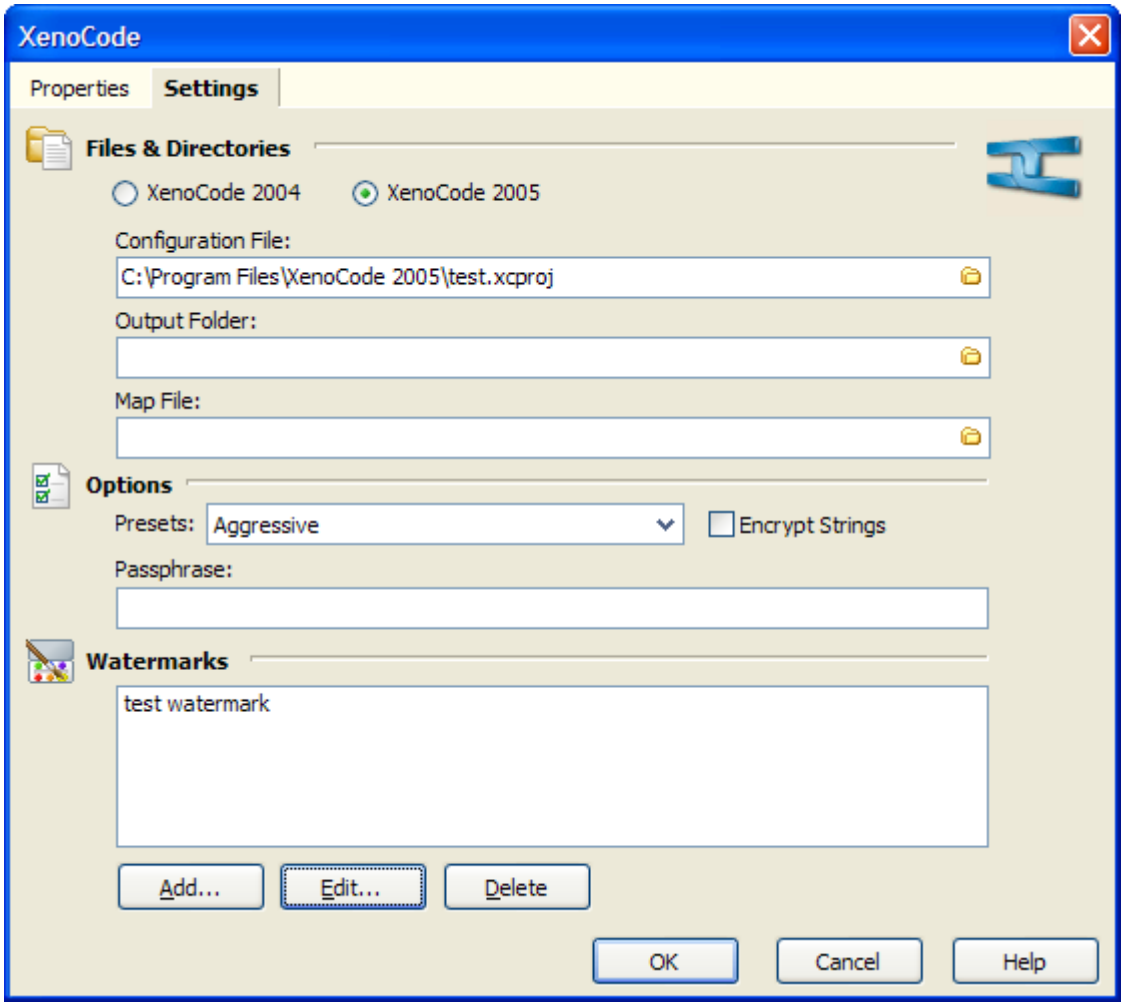
5.21.3.4 XenoCode

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/

© 2001-2006 VSoft Technologies Pty Ltd



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.

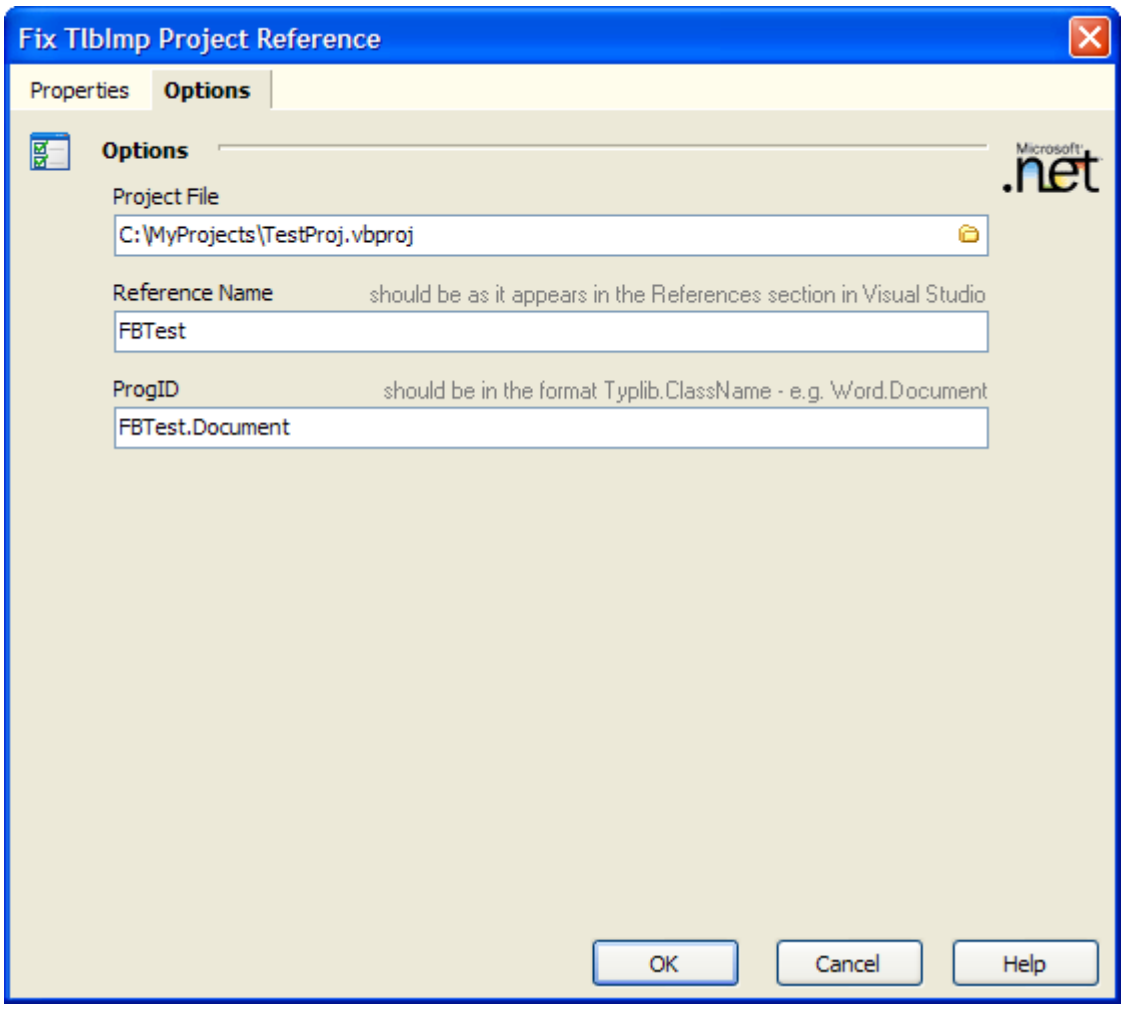
If no output folder is specified, then the output folder is set to a XenoCode subdirectory of the configuration folder.

© 2001-2006 VSoft Technologies Pty Ltd

5.21.4 Other

5.21.4.1 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 Typlibrary 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 typelibrary 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 TyplibName.ClassName, for example Word.Application. This prog id enables the action to find the typelibrary 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.

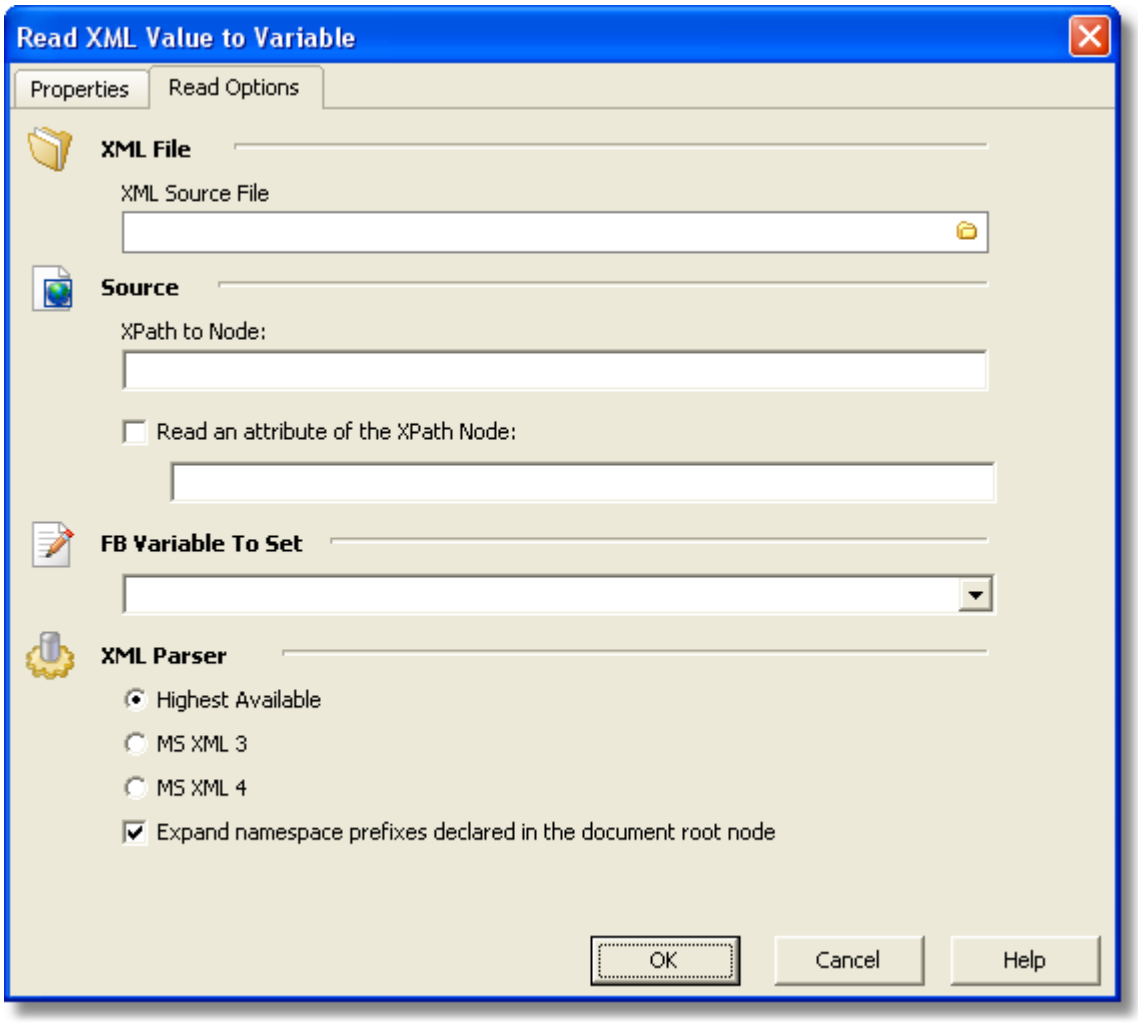
© 2001-2006 VSoft Technologies Pty Ltd

5.22 XML Actions

5.22.1 Read XML Value to Variable

[Professional Edition]

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

© 2001-2006 VSoft Technologies Pty Ltd

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.

XML Parser

Specify a version of the MS XML Parser to use. "Highest Available" should be fine.

Expand namespace prefixes declared in document root node

If this option is checked then the XML Parser will expand any xmlns: attributes that it finds in the document root node. This enables the use of XPaths which reference nodes with XML Namespace prefixes.

This option is included to overcome a shortcoming in MS XML, whereby XPaths do not automatically take into consideration namespaces or namespace prefixed nodes within a document.

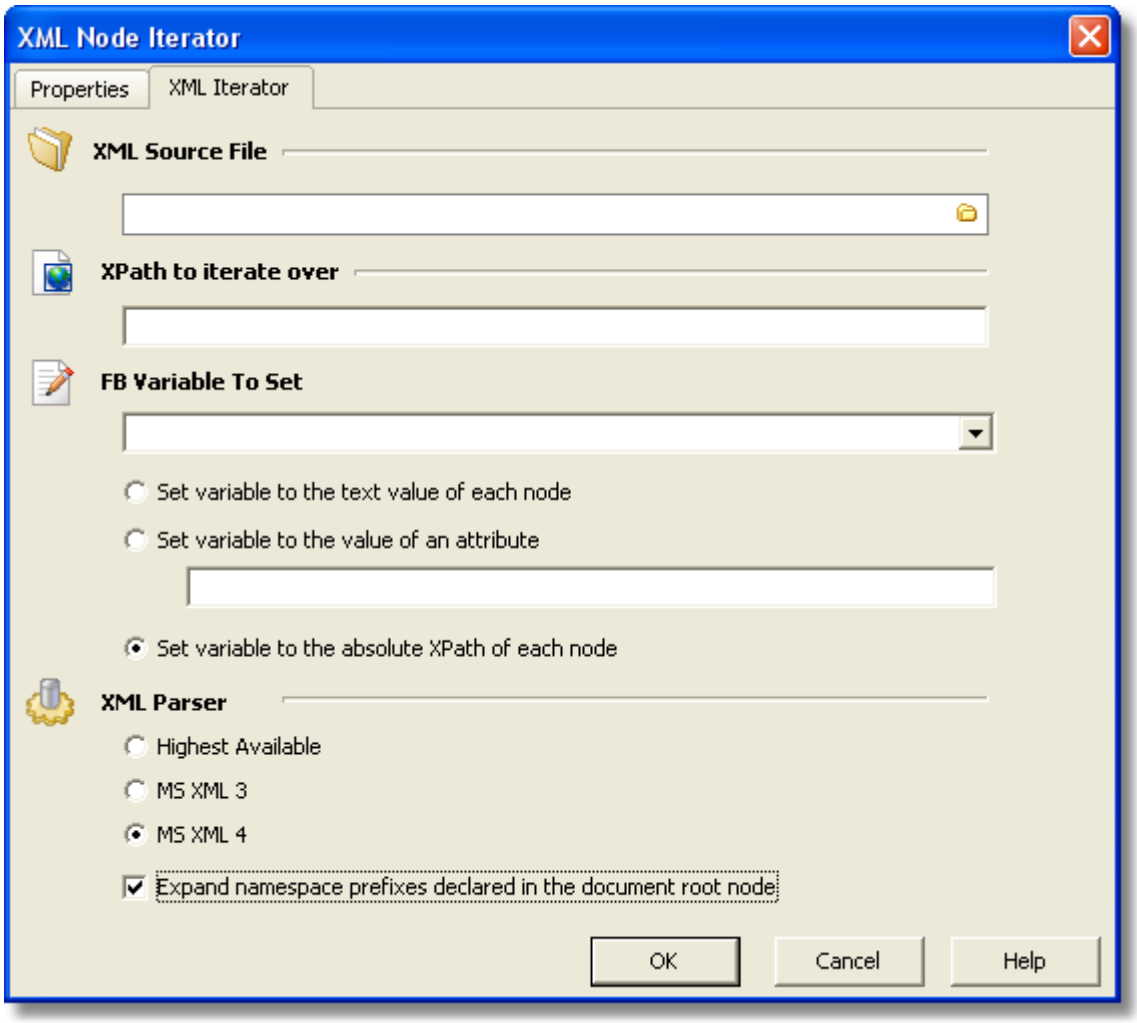
(Note that if a namespace prefix is declared somewhere other than the document root node, you will not be able to specify XPaths which reference that namespace! In order to reference these nodes then you will need to add the relevant declaration to the document root node.)

© 2001-2006 VSoft Technologies Pty Ltd

5.22.2 XML Node Iterator

[Professional Edition]

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 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.

© 2001-2006 VSoft Technologies Pty Ltd

- **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.

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.

Expand namespace prefixes declared in document root node

If this option is checked then the XML Parser will expand any xmlns: attributes that it finds in the document root node. This enables the use of XPaths which reference nodes with XML Namespace prefixes.

This option is included to overcome a shortcoming in MS XML, whereby XPaths do not automatically take into consideration namespaces or namespace prefixed nodes within a document.

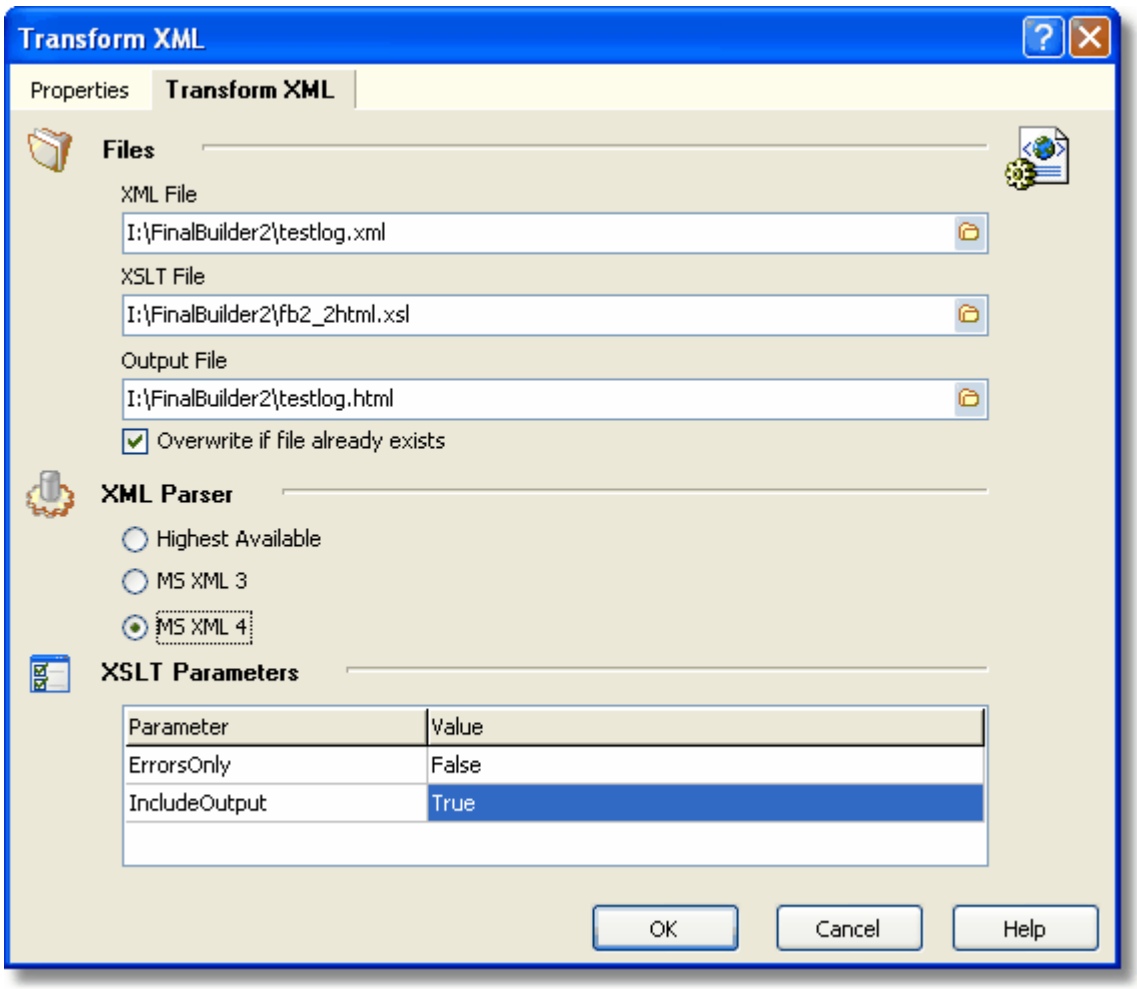
(Note that if a namespace prefix is declared somewhere other than the document root node, you will not be able to specify XPaths which reference that namespace! In order to reference these nodes then you will need to add the relevant declaration to the document root node.)

© 2001-2006 VSoft Technologies Pty Ltd

5.22.3 Transform XML

[Professional Edition]

This action performs an XSL Transform using the Microsoft XML parser.



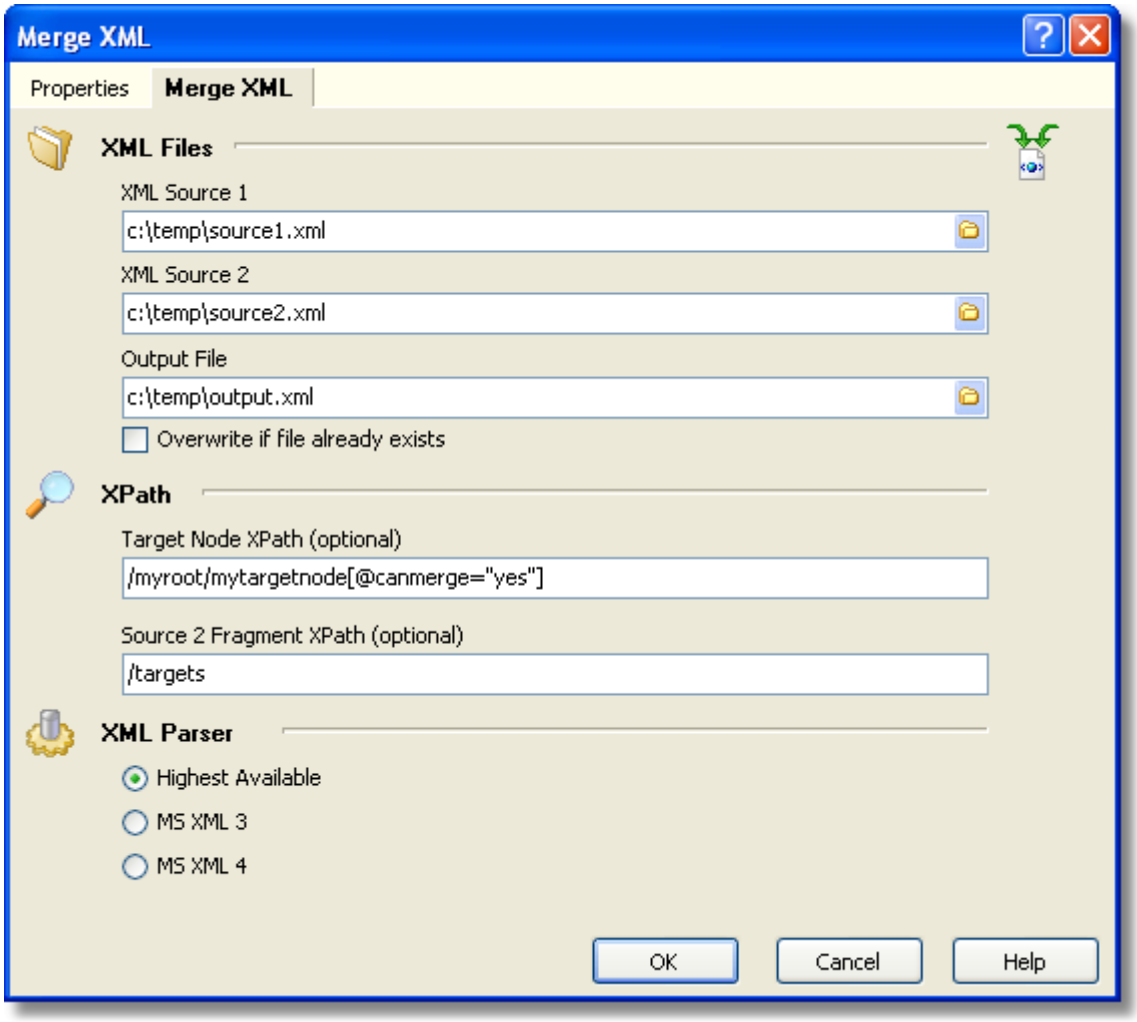
The XSLT Parameters allows you to provide dynamic values (such as build numbers etc) to your stylesheet, which can be used to alter the output of the transform.

© 2001-2006 VSoft Technologies Pty Ltd

5.22.4 Merge XML Action

[Professional Edition]

This action merges two XML documents. Source 2 will be merged into Source 1 and saved as a new document.



The TargetNode XPath option allows you to specify the node where the Source2 document will be inserted.

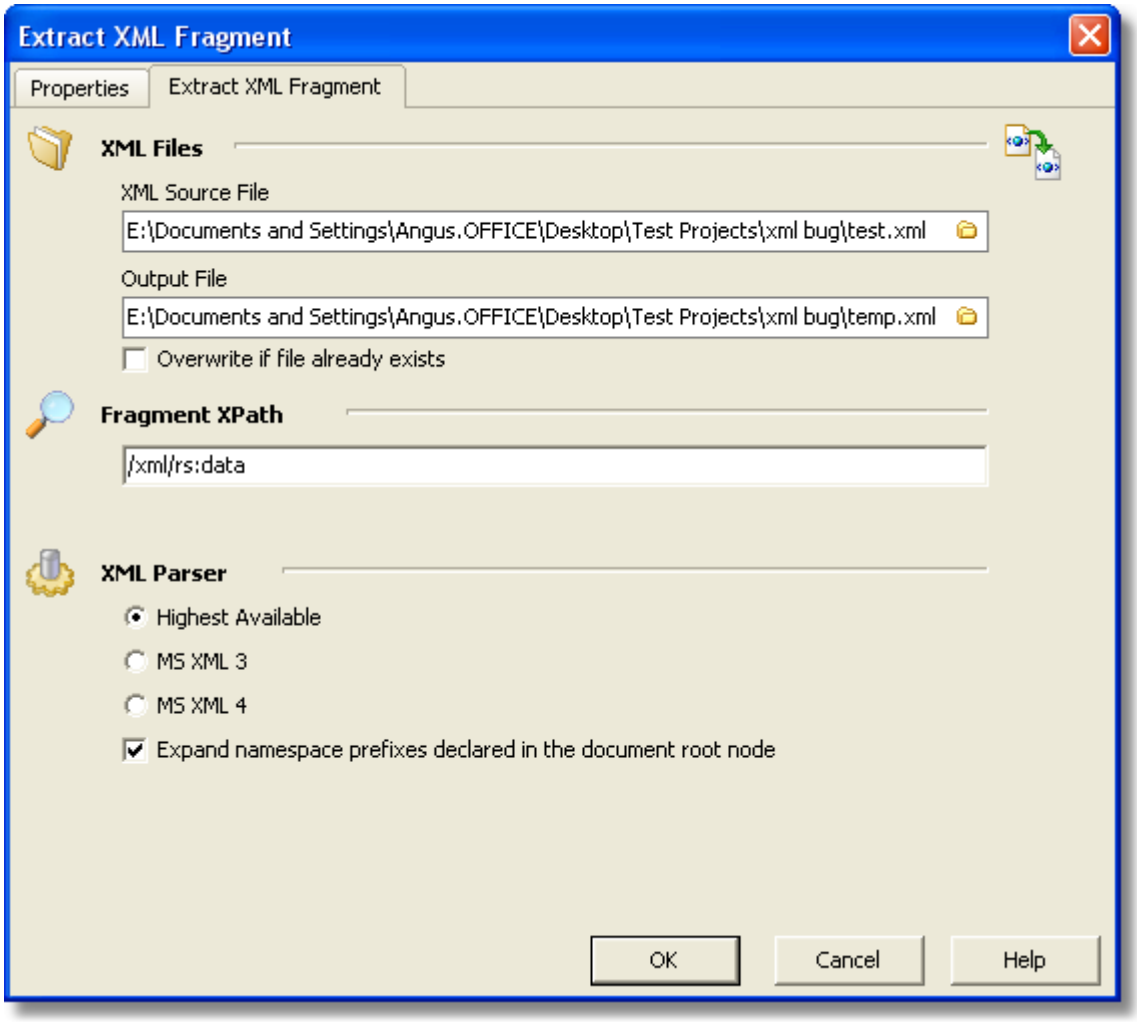
The Source 3 Fragment XPath option allows you to specify a document fragment to merge rather than the whole document.

© 2001-2006 VSoft Technologies Pty Ltd

5.22.5 Extract XML Fragment Action

[Professional Edition]

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 and a destination file to write the fragment to.

Fragment XPAth

Specify the XPath to the nodes you wish to extract.

XML Parser

Specify which version of the MS XML Parser you wish to use to parse the document. In most cases, 'Highest Available' will be fine.

Expand namespace prefixes declared in document root node

© 2001-2006 VSoft Technologies Pty Ltd

If this option is checked then the XML Parser will expand any xmlns: attributes that it finds in the document root node. This enables the use of XPaths which reference nodes with XML Namespace prefixes.

This option is included to overcome a shortcoming in MS XML, whereby XPaths do not automatically take into consideration namespaces or namespace prefixed nodes within a document.

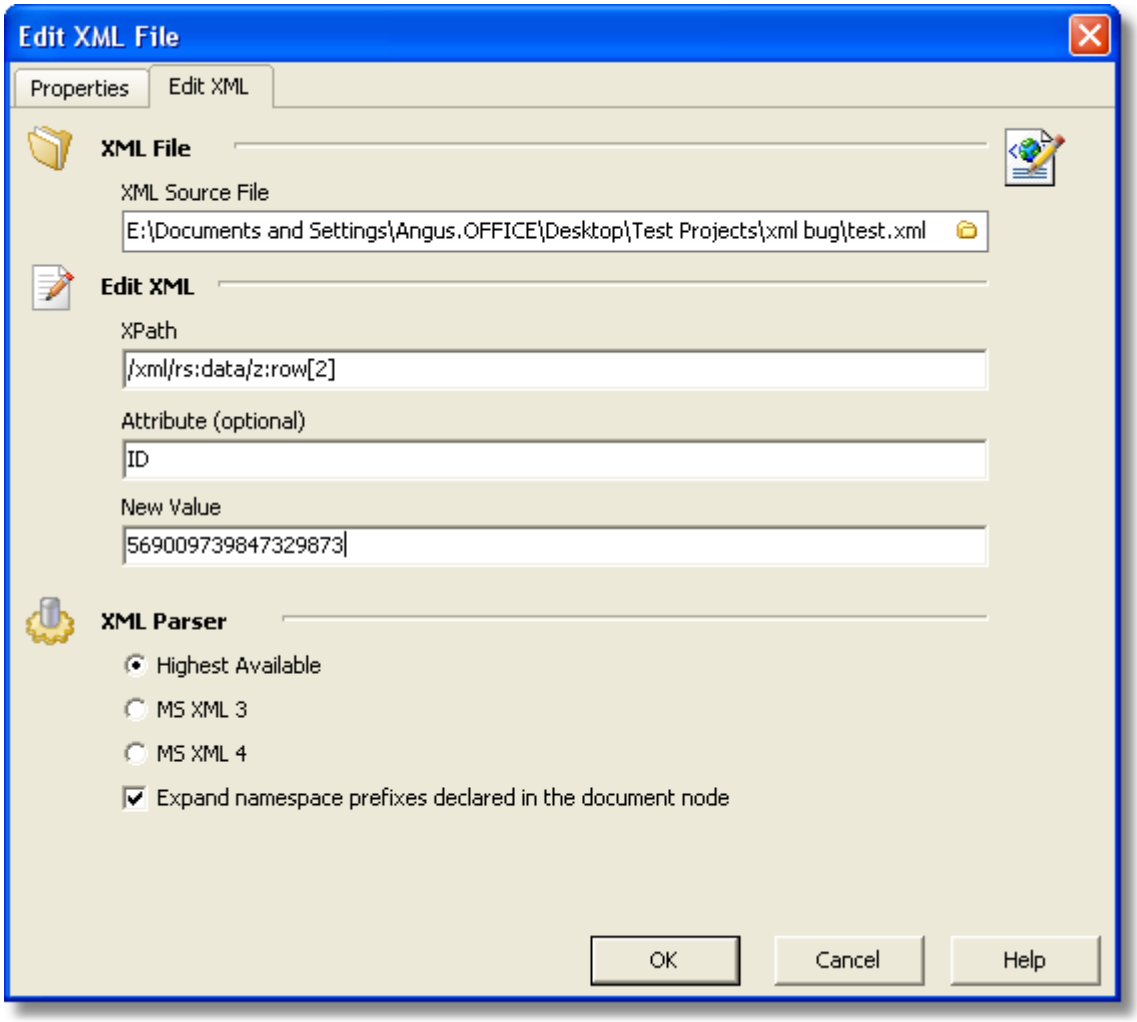
(Note that if a namespace prefix is declared somewhere other than the document root node, you will not be able to specify XPaths which reference that namespace! In order to reference these nodes then you will need to add the relevant declaration to the document root node.)

© 2001-2006 VSoft Technologies Pty Ltd

5.22.6 Edit XML File Action

[Professional Edition]

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 the XML file 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.

© 2001-2006 VSoft Technologies Pty Ltd

New Value

Specify the value to write to the XML file.

XML Parser

Specify which version of the MS XML Parser to use to parse the file. In most circumstances, "Highest Available" will work fine.

Expand namespace prefixes declared in document root node

If this option is checked then the XML Parser will expand any xmlns: attributes that it finds in the document root node. This enables the use of XPaths which reference nodes with XML Namespace prefixes.

This option is included to overcome a shortcoming in MS XML, whereby XPaths do not automatically take into consideration namespaces or namespace prefixed nodes within a document.

(Note that if a namespace prefix is declared somewhere other than the document root node, you will not be able to specify XPaths which reference that namespace! In order to reference these nodes then you will need to add the relevant declaration to the document root node.)

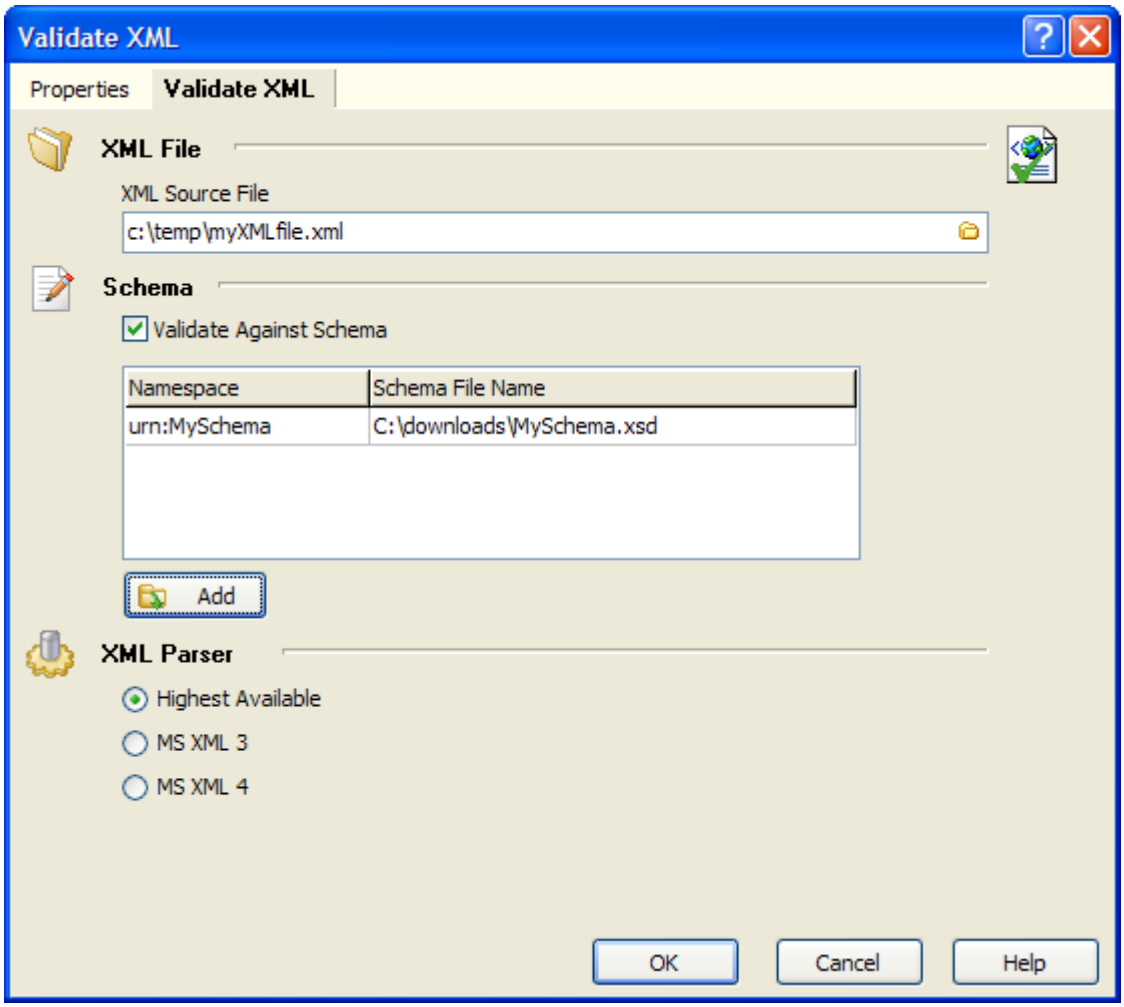
© 2001-2006 VSoft Technologies Pty Ltd

5.22.7 Validate XML File

[Professional Edition]

This action validates an XML file using the Microsoft XML parser.

You can validate against schemas, or if no schemas are specifies then the action parses the xml file and reports any errors found.

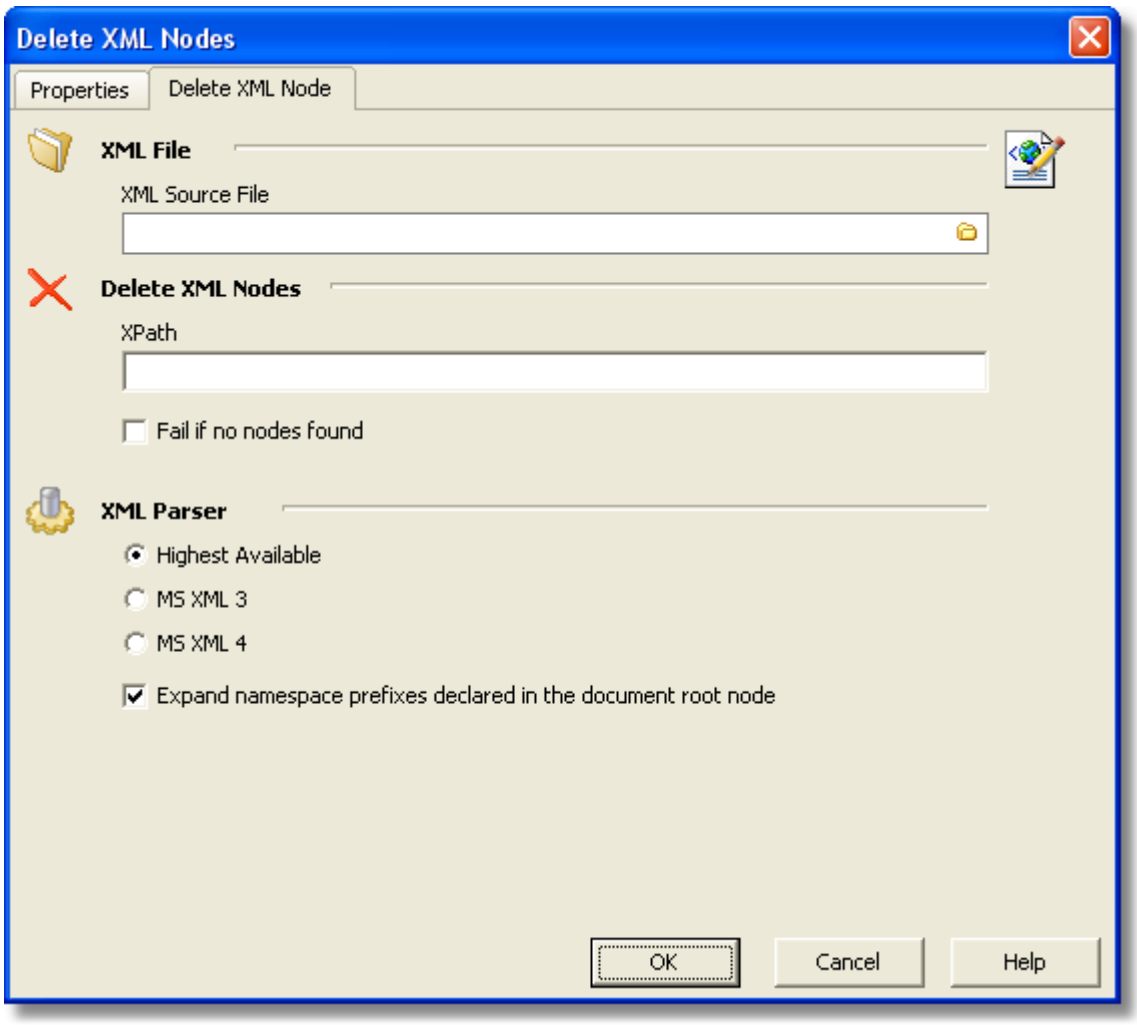


© 2001-2006 VSoft Technologies Pty Ltd

5.22.8 Delete XML Nodes

[Professional Edition]

This action allows you to delete a node or set of nodes in an XML document.



XML File

Choose the XML File you wish 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.

XML Parser

Specify a version of the MS XML Parser to use. In most situations, 'Highest Available' will be fine.

Expand namespace prefixes declared in document root node

© 2001-2006 VSoft Technologies Pty Ltd

If this option is checked then the XML Parser will expand any xmlns: attributes that it finds in the document root node. This enables the use of XPaths which reference nodes with XML Namespace prefixes.

This option is included to overcome a shortcoming in MS XML, whereby XPaths do not automatically take into consideration namespaces or namespace prefixed nodes within a document.

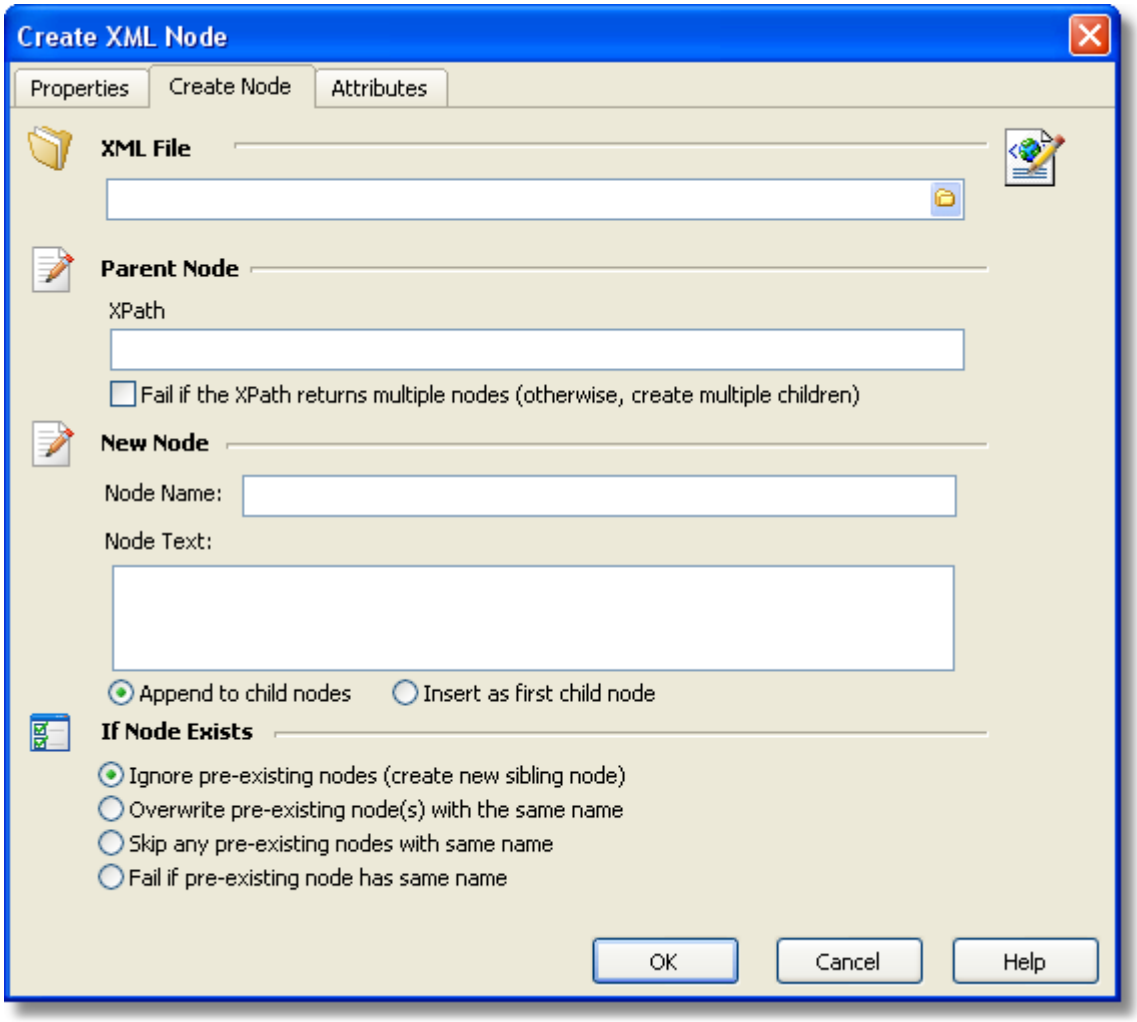
(Note that if a namespace prefix is declared somewhere other than the document root node, you will not be able to specify XPaths which reference that namespace! In order to reference these nodes then you will need to add the relevant declaration to the document root node.)

© 2001-2006 VSoft Technologies Pty Ltd

5.22.9 Create XML Node

[Professional Edition]

The Create XML Node action allows you to insert an extra node into an existing XML file.



XML File

Enter the path to the XML file you wish to modify. The file must be writable.

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.

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

© 2001-2006 VSoft Technologies Pty Ltd

any text content.

Append to child nodes / Insert as first child node

Choose "Append to child nodes" to have the new node created last - after any sibling nodes.

Choose "Insert as first child node" to have the new node created before any siblings.

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.

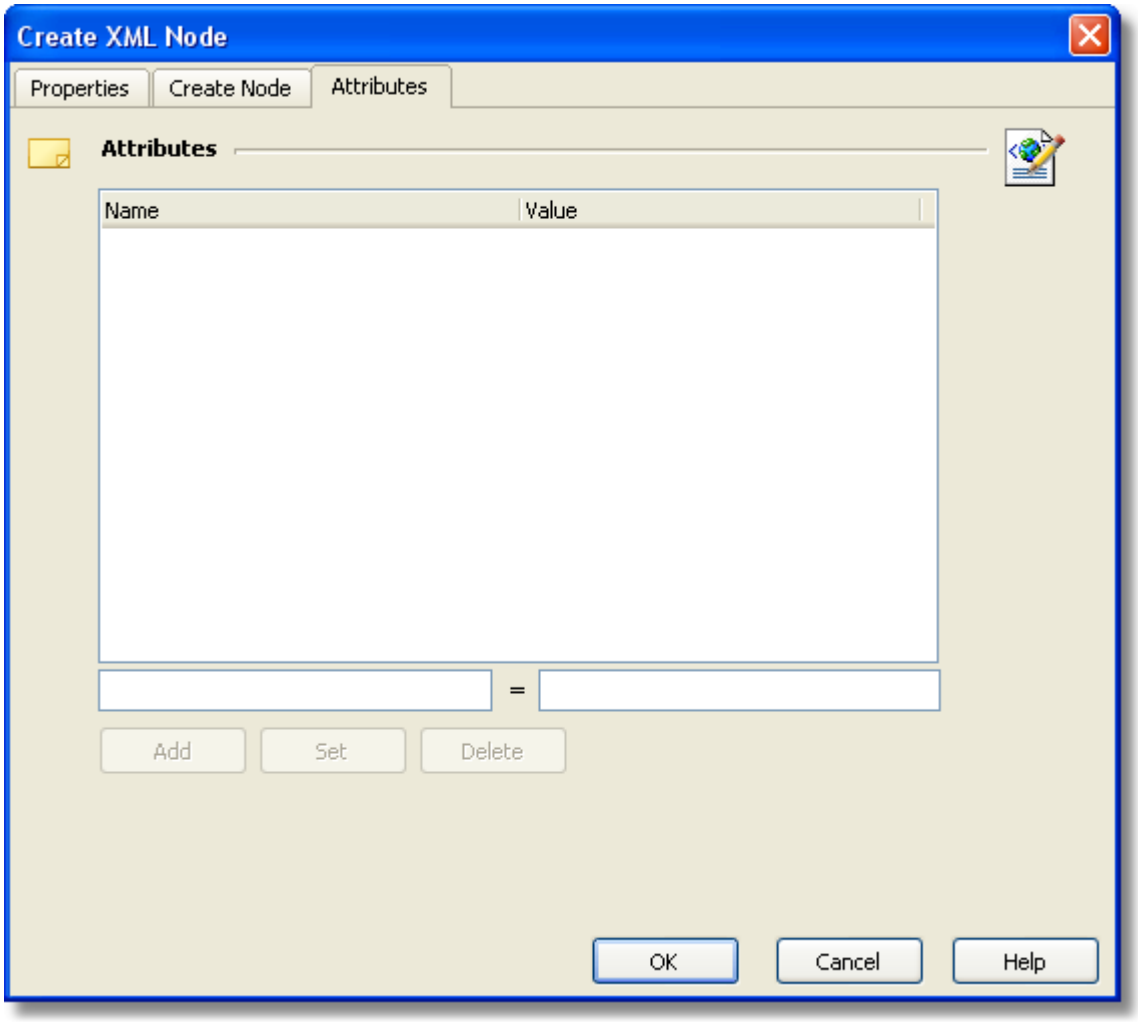
"Skip any pre-existing nodes with same name"

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.

© 2001-2006 VSoft Technologies Pty Ltd



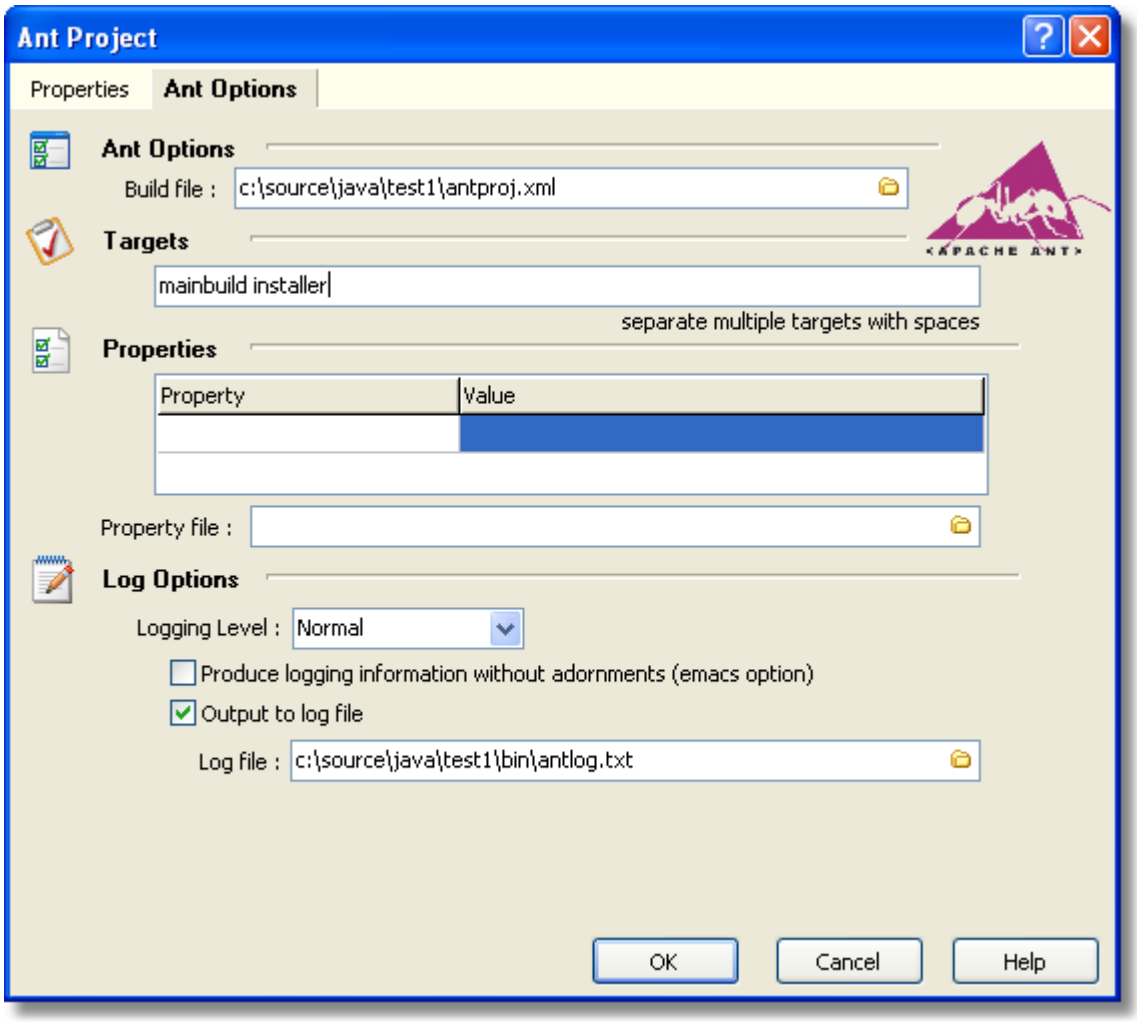
The attributes list shows attribute and value names for the new node. To add a new attribute, type its name and value into the edit fields and then click the Add button.

© 2001-2006 VSoft Technologies Pty Ltd

5.23 Build Tools

5.23.1 Ant Project Action

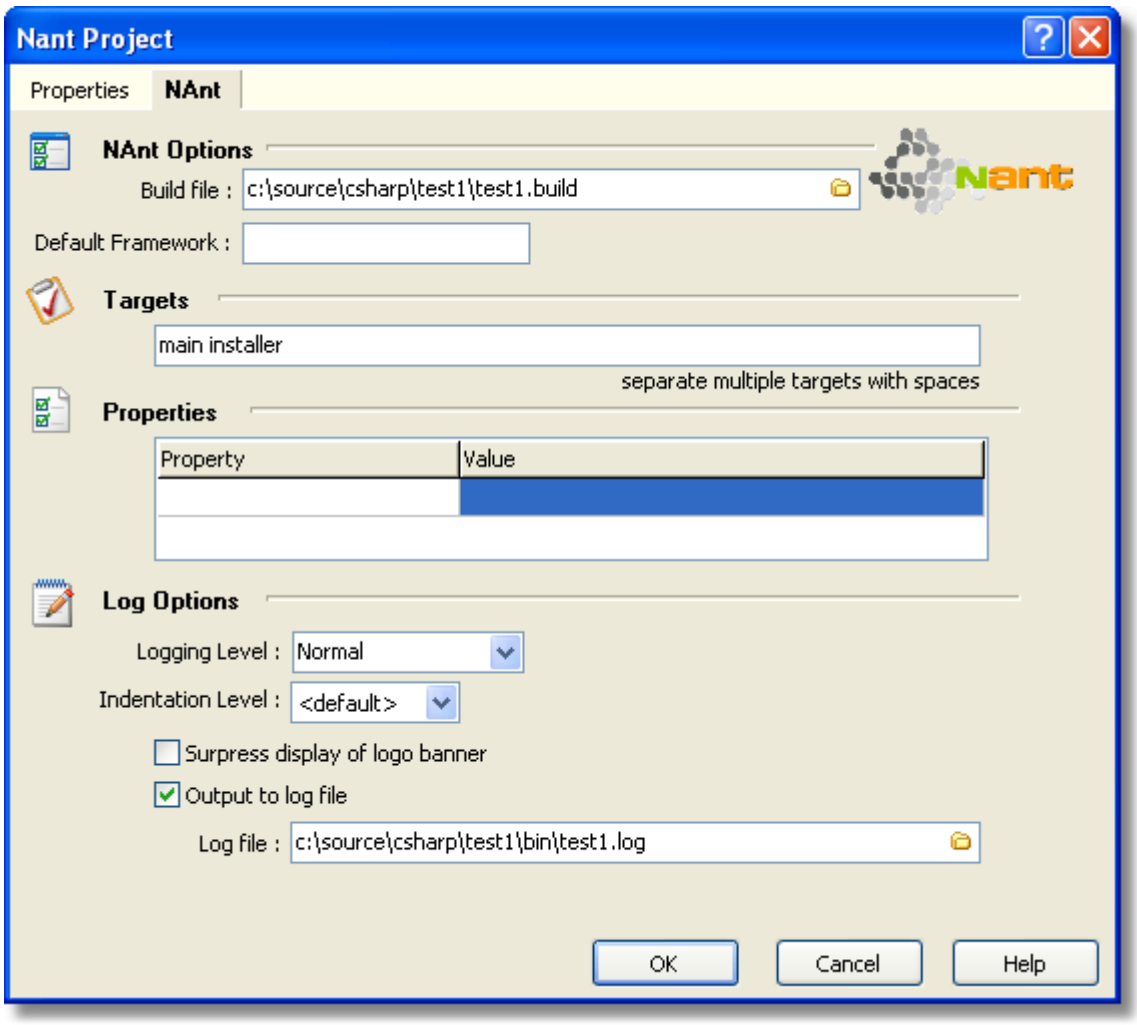
This action executes the Apache Ant command line build tool, making it easy to integrate legacy build scripts into your FinalBuilder Build process.



© 2001-2006 VSoft Technologies Pty Ltd

5.23.2 Nant Project Action

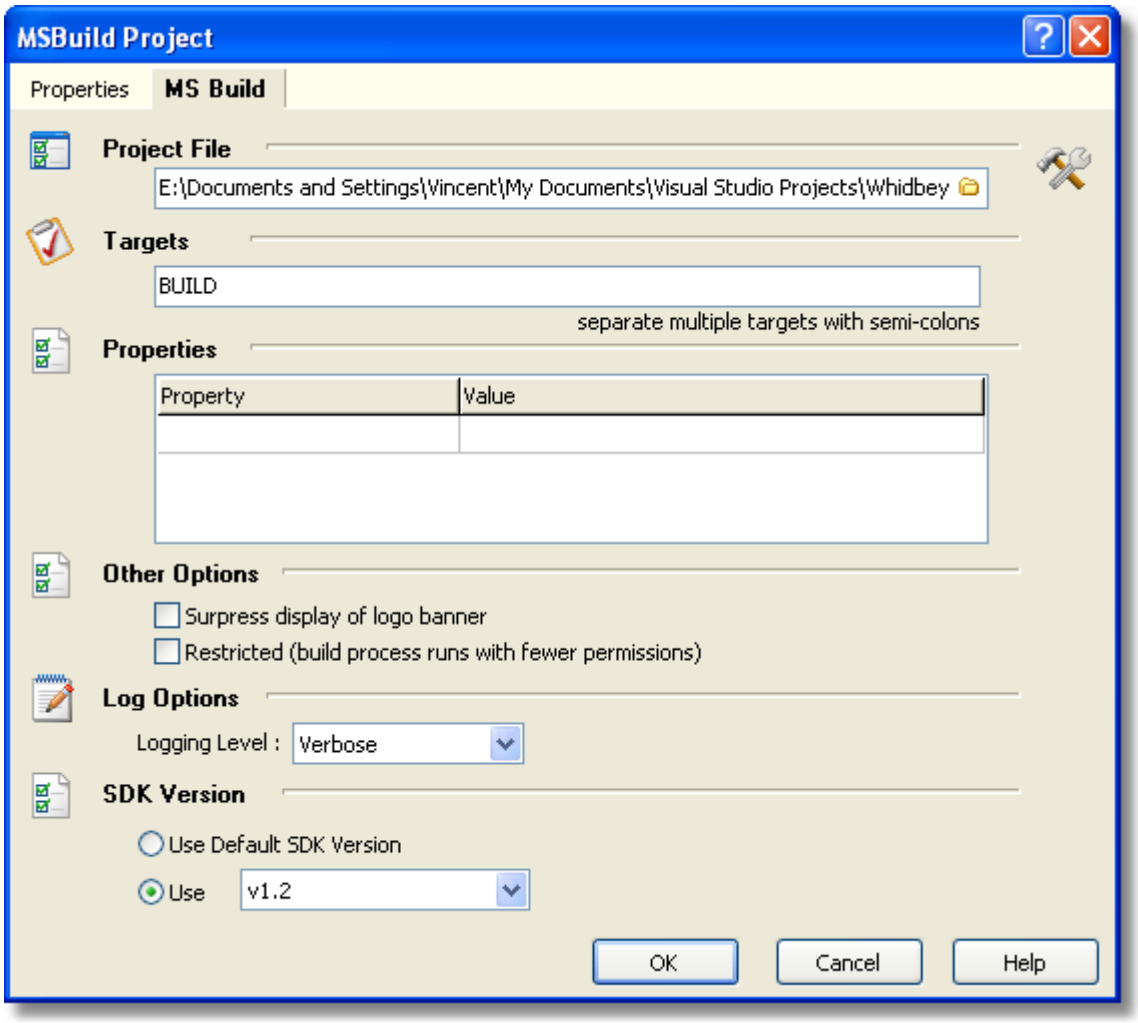
This action executes the Nant command line build tool, making it easy to integrate legacy build scripts into your FinalBuilder Build process.



© 2001-2006 VSoft Technologies Pty Ltd

5.23.3 MSBuild Project Action

This action runs the Microsoft MSBuild command line tool that is part of the .NET Framework 1.2. MSBuild projects are the format used by Visual Studio.NET 2004/Whidbey. Whidbey projects have the same file extension as earlier versions of visual studio (ie csproj, vbproj) but are actually MSBuild scripts.



© 2001-2006 VSoft Technologies Pty Ltd

5.24 Database

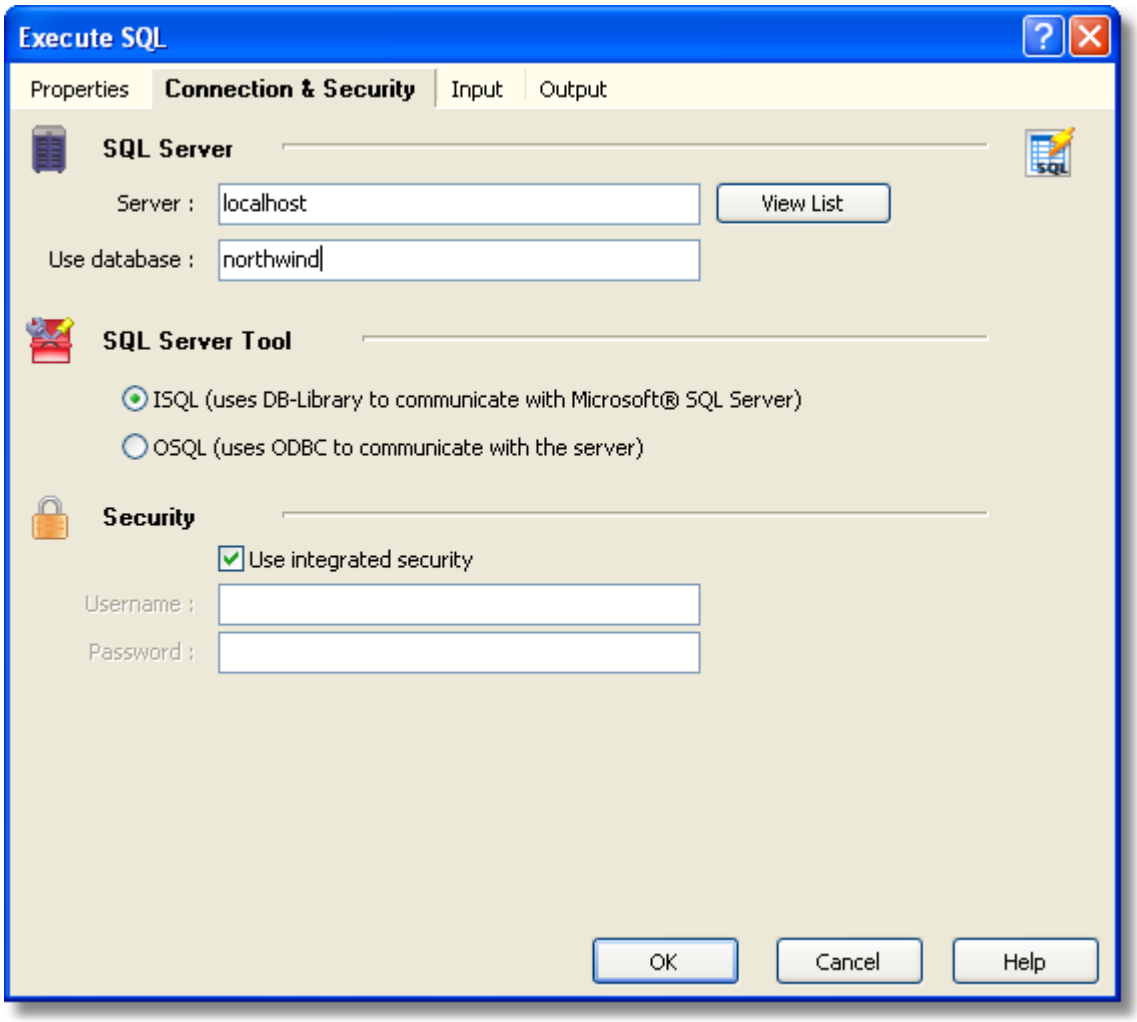
5.24.1 SQL Server Actions

5.24.1.1 Execute SQL Action

[Professional Edition]

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

Connection & Security Tab



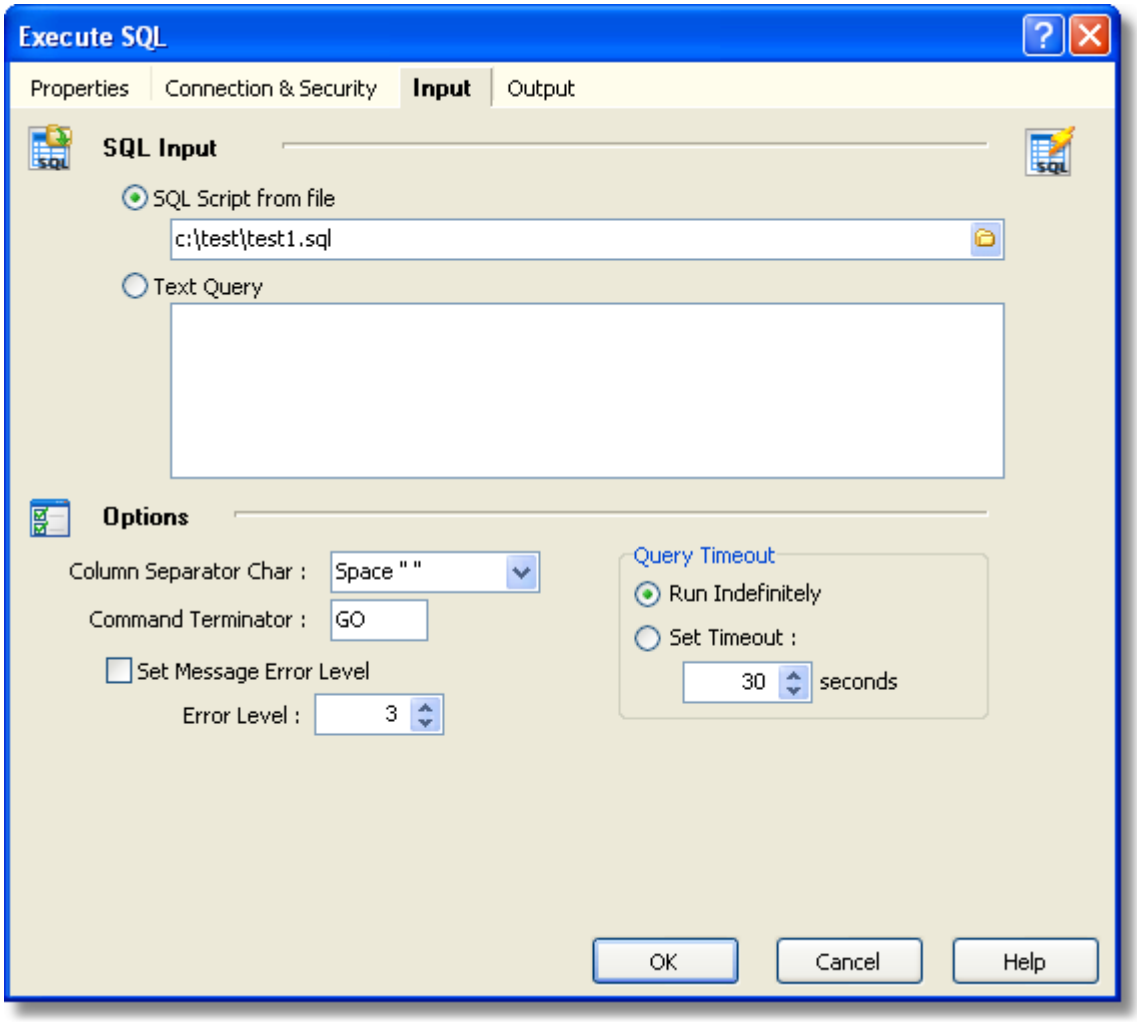
Server - Specify the MS SQL Server to use. Clicking "View List" will attempt to locate any SQL Servers on the network.
Use Database - Specify a database to run the SQL against.

SQL Server Tool - choose the method (isql or osql) to use to connect with.

Security - If not using integrated (ie. Windows) security, you should specify the username and password to use.

Input Tab

© 2001-2006 VSoft Technologies Pty Ltd



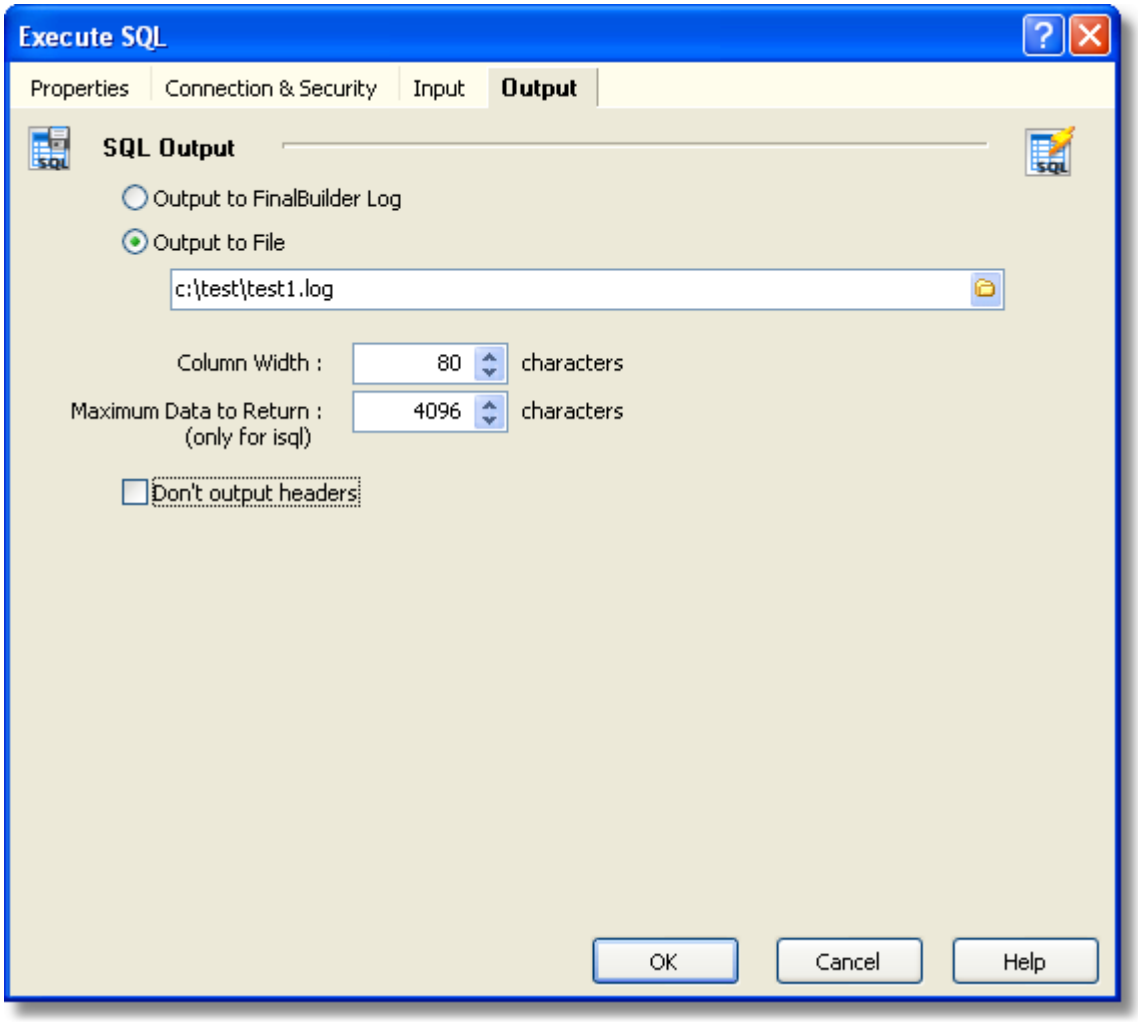
SQL Input - SQL Script from file - specify a file which contains an SQL Script.
- Text Query - Specify the SQL Query to execute.

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

Output Tab

© 2001-2006 VSoft Technologies Pty Ltd



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

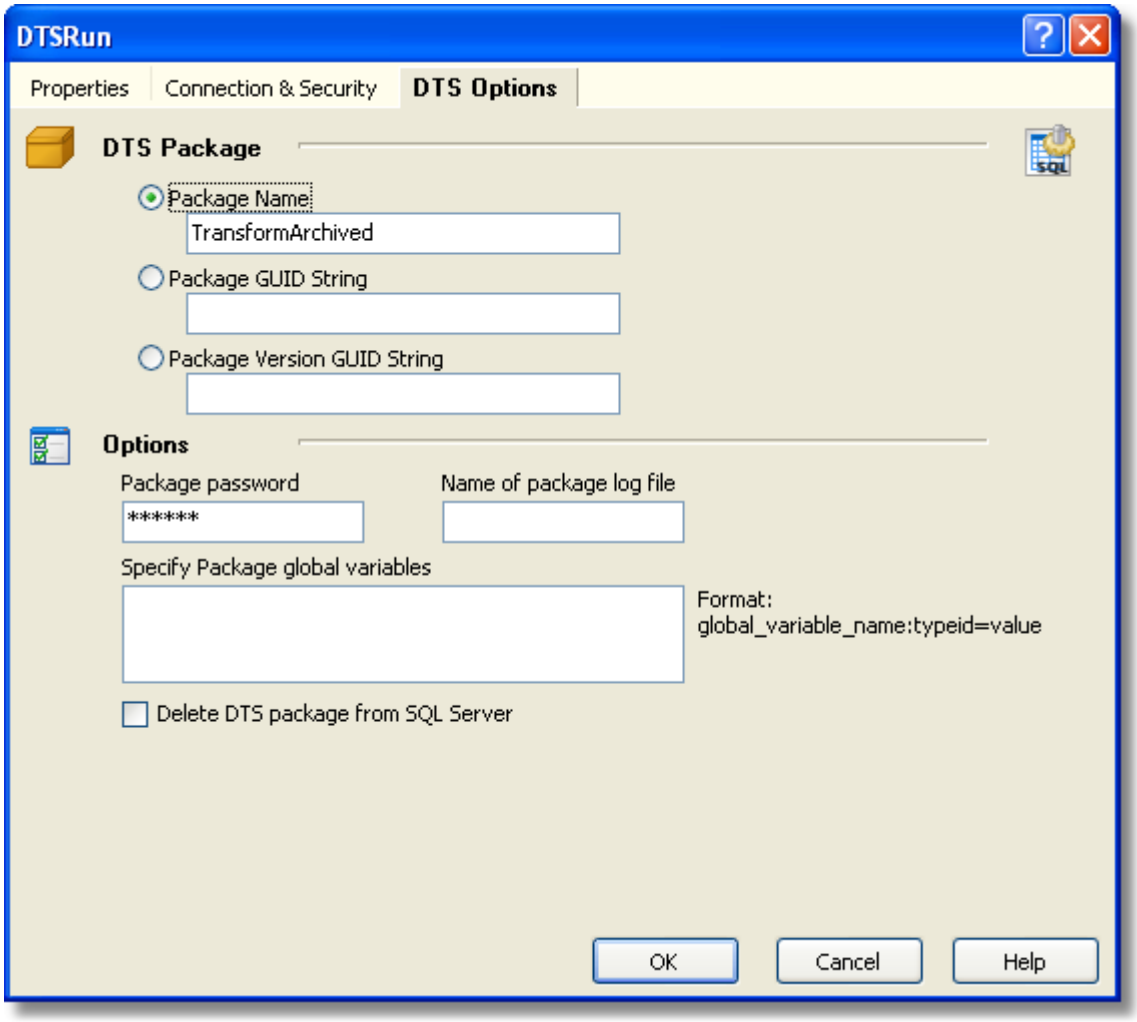
isql : http://msdn.microsoft.com/library/en-us/coprompt/cp_isql_8r39.asp
osql : http://msdn.microsoft.com/library/en-us/coprompt/cp_osql_1wxl.asp

5.24.1.2 DTSRun Action

[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.

© 2001-2006 VSoft Technologies Pty Ltd



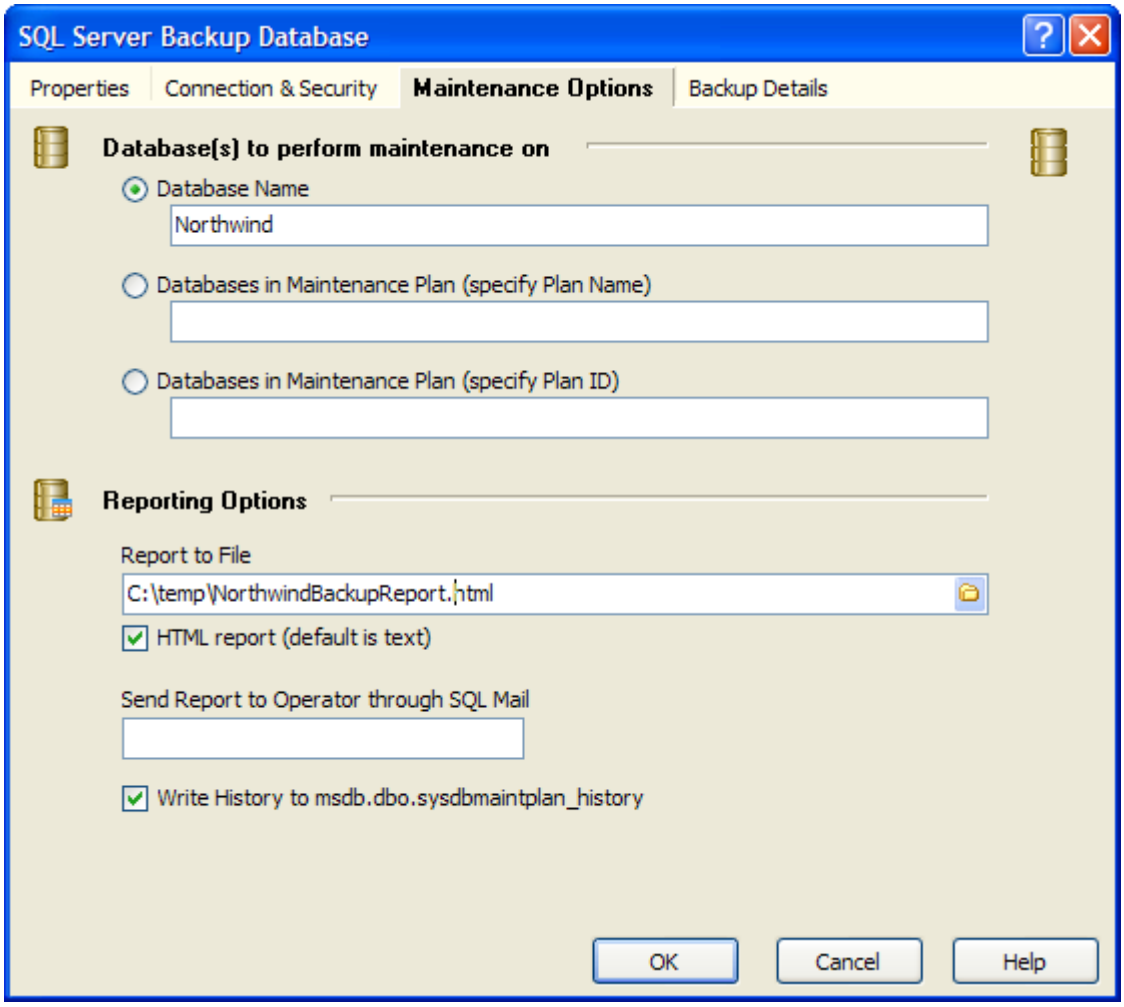
For information on the options for this see:

http://msdn.microsoft.com/library/en-us/coprompt/cp_dtsrun_95kp.asp

5.24.1.3 SQL Server Backup Database

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

© 2001-2006 VSoft Technologies Pty Ltd

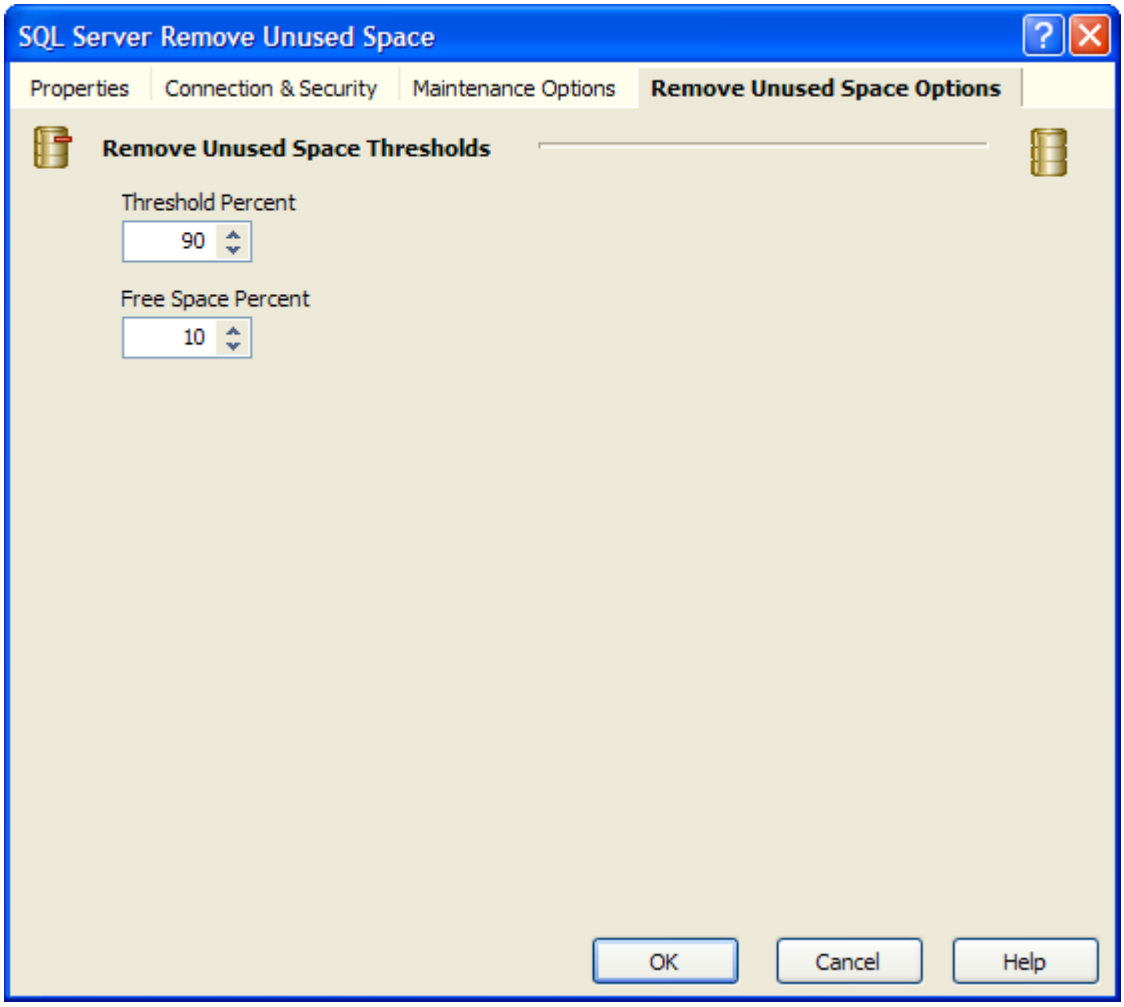


For more information on the sqlmaint tool see:
http://msdn.microsoft.com/library/en-us/coprompt/cp_sqlmaint_19ix.asp

5.24.1.4 SQL Server Remove Unused Space

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.

© 2001-2006 VSoft Technologies Pty Ltd



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

5.24.1.5 SQL Server Check Database

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

5.24.1.6 SQL Server Check Catalogue

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.

© 2001-2006 VSoft Technologies Pty Ltd

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

5.24.1.7 SQL Server Update DB Statistics

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

5.24.1.8 SQL Server Rebuild Indexes

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

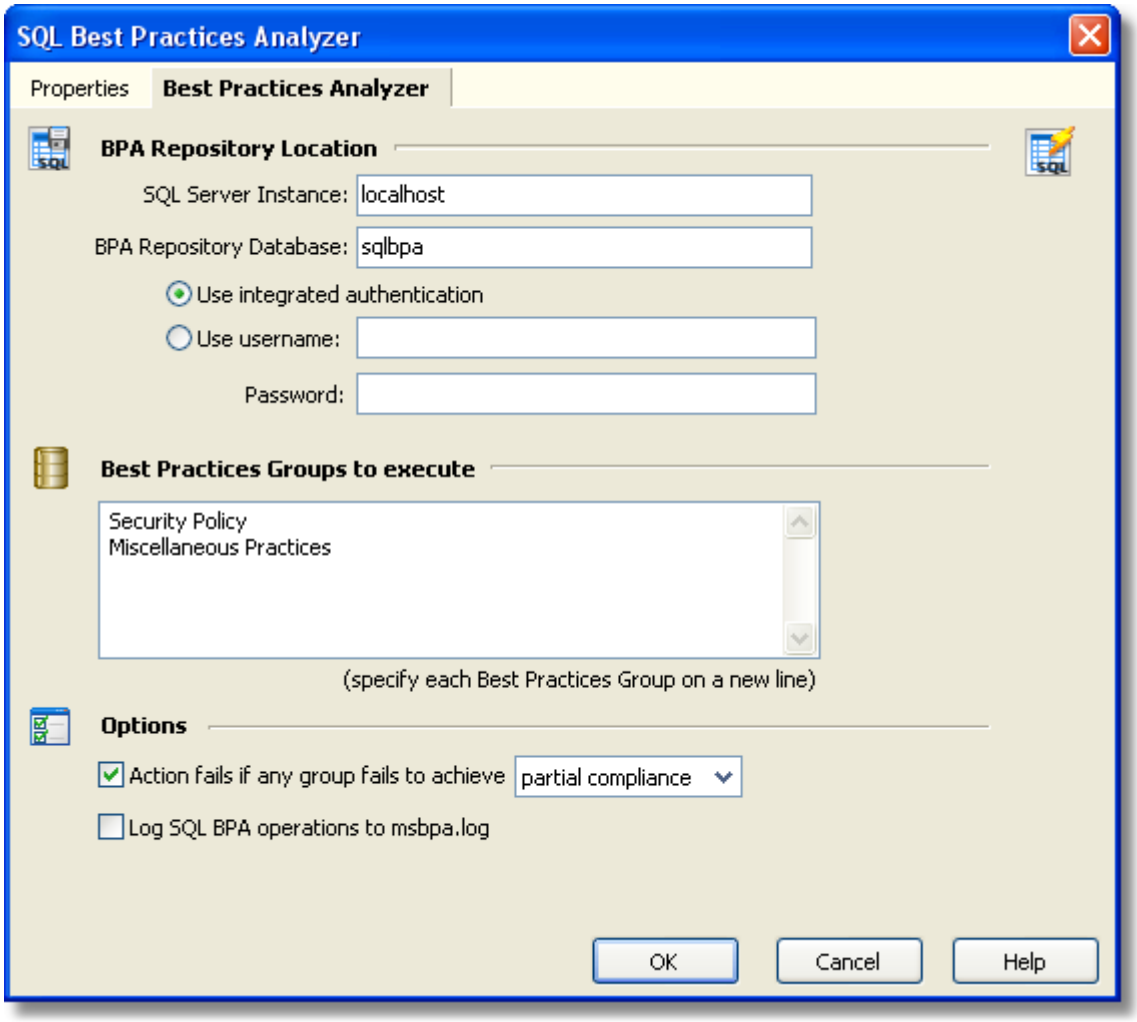
5.24.1.9 SQL Server Best Practices Analyzer

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.

© 2001-2006 VSoft Technologies Pty Ltd



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)

© 2001-2006 VSoft Technologies Pty Ltd

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.

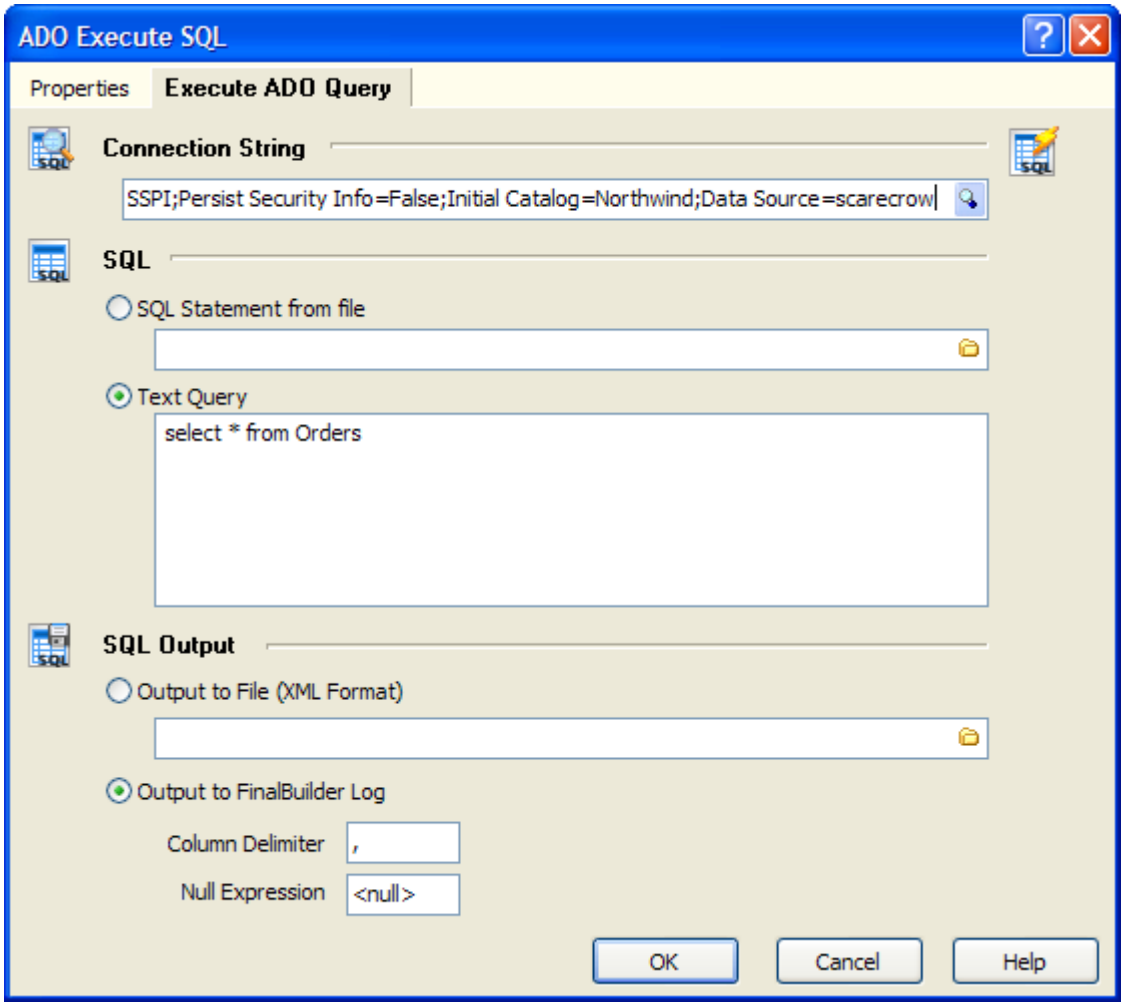
Note that each Best Practices Group also generates a log file in this directory, regardless of whether or not this option is checked.

© 2001-2006 VSoft Technologies Pty Ltd

5.24.2 ADO Execute SQL

[Professional Edition]

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.

© 2001-2006 VSoft Technologies Pty Ltd

5.24.3 ADO Execute Stored Procedure

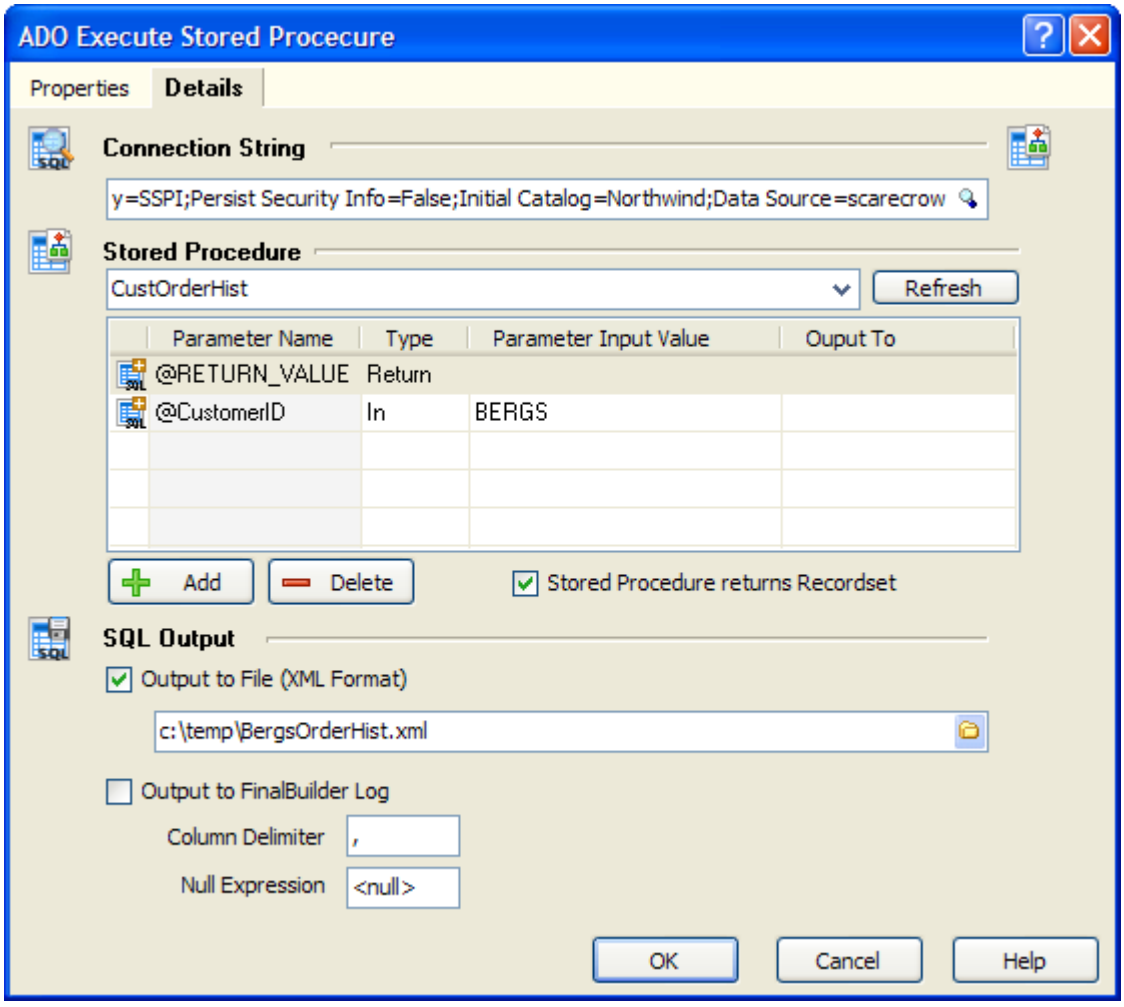
[Professional Edition]

The ADO Stored Procedure action enables you to automate the execution of stored procedures using ADO as part of your build process.

The action property pages helps you build your connection string, displays a list of stored procedures in the target database, and will retrieve the parameters for the selected stored procedure. The value of each input parameter can be set, and the out values and return values can be saved to a variable.

It is important to set "Stored Procedure Returns Recordset" correctly, as this controls how the action internally calls the stored procedure.

NOTE: This action is only available in the Professional Edition of FinalBuilder



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.

© 2001-2006 VSoft Technologies Pty Ltd

Parameters - The Name and Type of the parameters should retrieved automatically, if not then you need set them to the correct values. For any IN or INOUT parameters you should set a value (either a hardcoded value, or use an FB Variable). For any Return, OUT, or INOUT parameters you can optionally set an FB Variable for the value to be saved to when the stored procedure executes.

Use the **Add** and **Delete** buttons to manually define the parameters if the stored procedure is yet to be defined.

Stored Procedure returns Recordset - It is very important that you set the "Stored Procedure returns Recordset" option correctly, as this determines how the stored procedure is called internally.

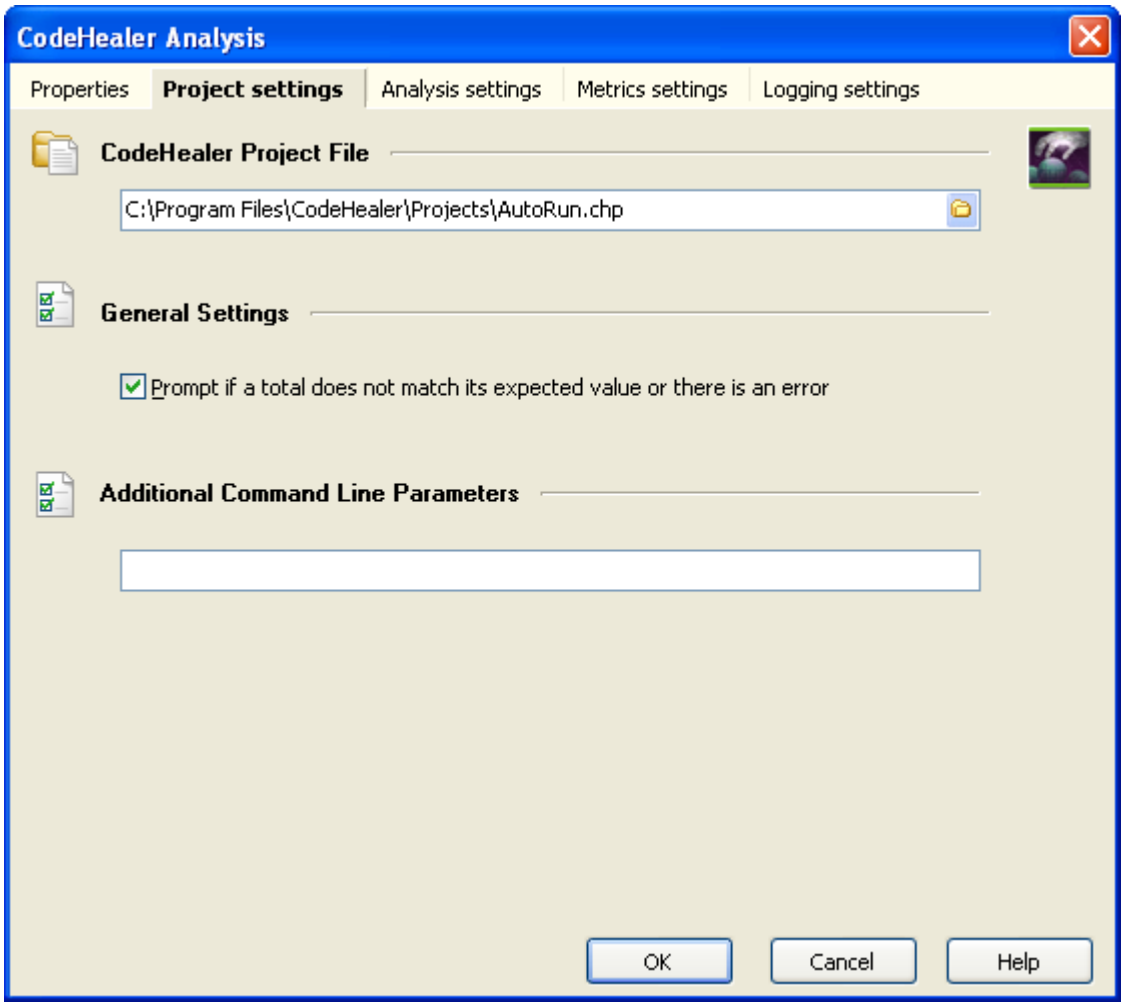
SQL Output - You can output a result recordset to the FB log, and/or to and XML file using the ADO XML file format.

© 2001-2006 VSoft Technologies Pty Ltd

5.25 Source Code Tools

5.25.1 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.



CodeHealer Project Settings

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

© 2001-2006 VSoft Technologies Pty Ltd

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.

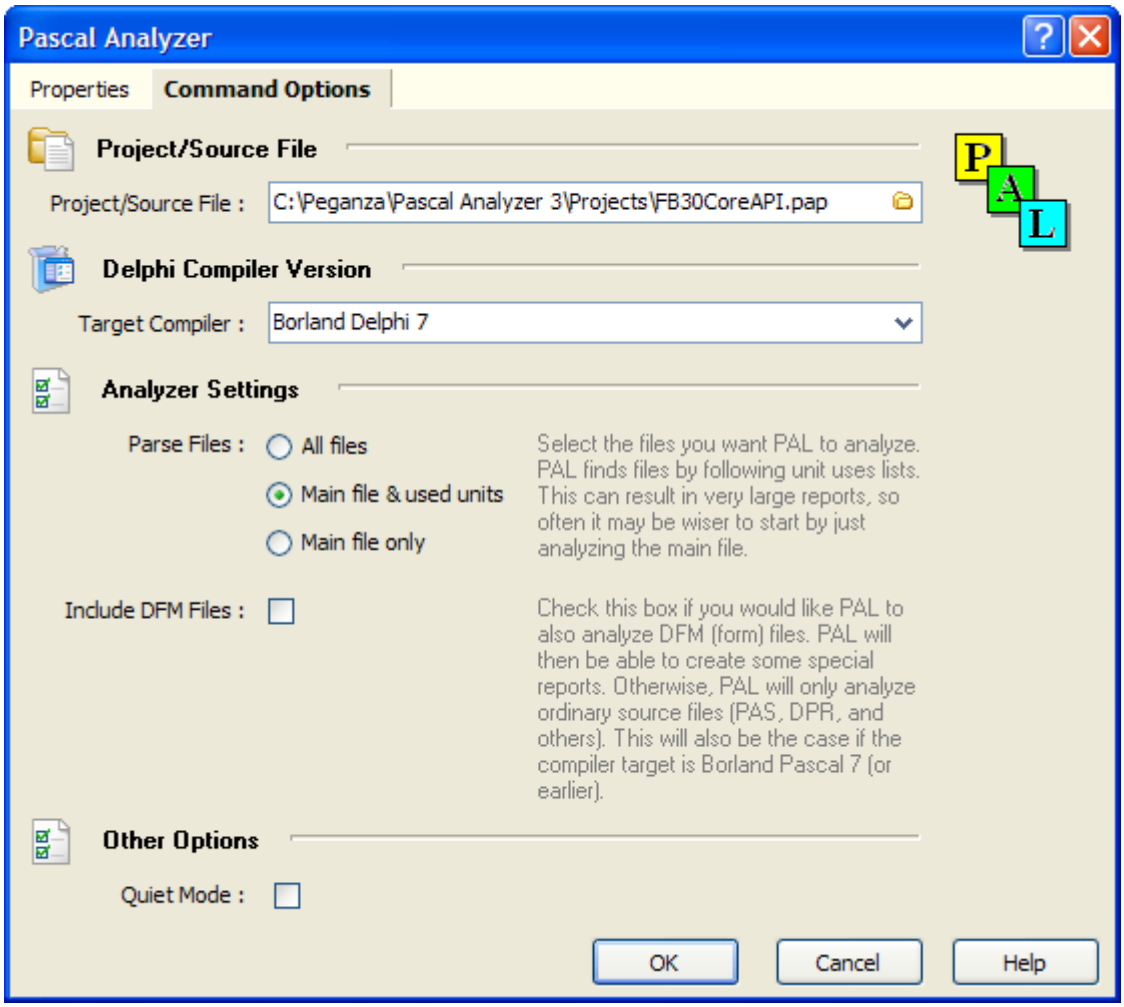
The information provided by CodeHealer will also help developers understand and improve existing code that they have not written. This is especially helpful for corporate developers who take over the management of legacy applications.

For more information on CodeHealer, and to download a trial version, please visit http://www.socksoftware.com.

© 2001-2006 VSoft Technologies Pty Ltd

5.25.2 Pascal Analyzer

The Pascal Analyzer action enables you to analyze your Pascal source code as part of your build process 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 Borland 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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

© 2001-2006 VSoft Technologies Pty Ltd

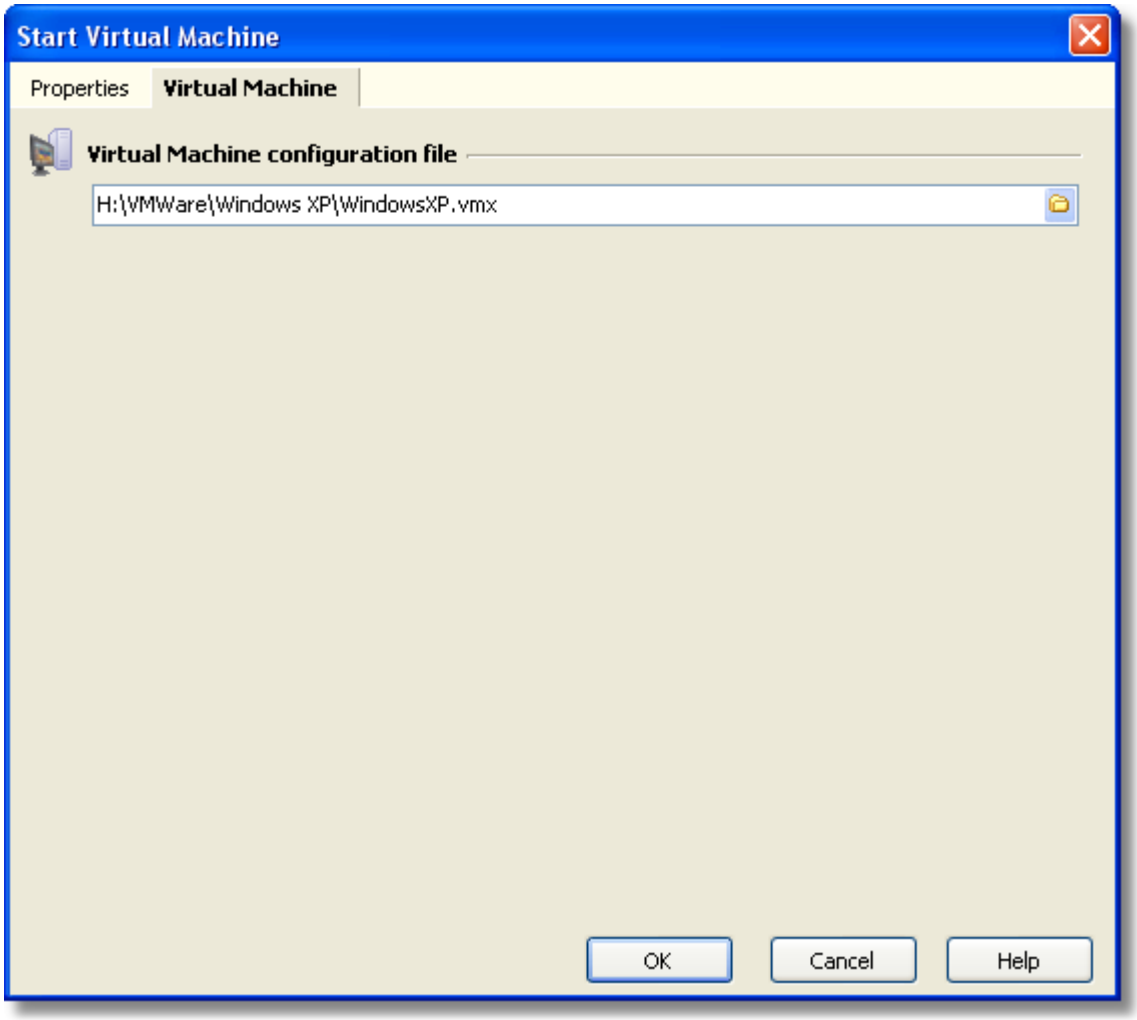
5.26 Virtualization Systems

5.26.1 VMWare Workstation

5.26.1.1 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.

© 2001-2006 VSoft Technologies Pty Ltd

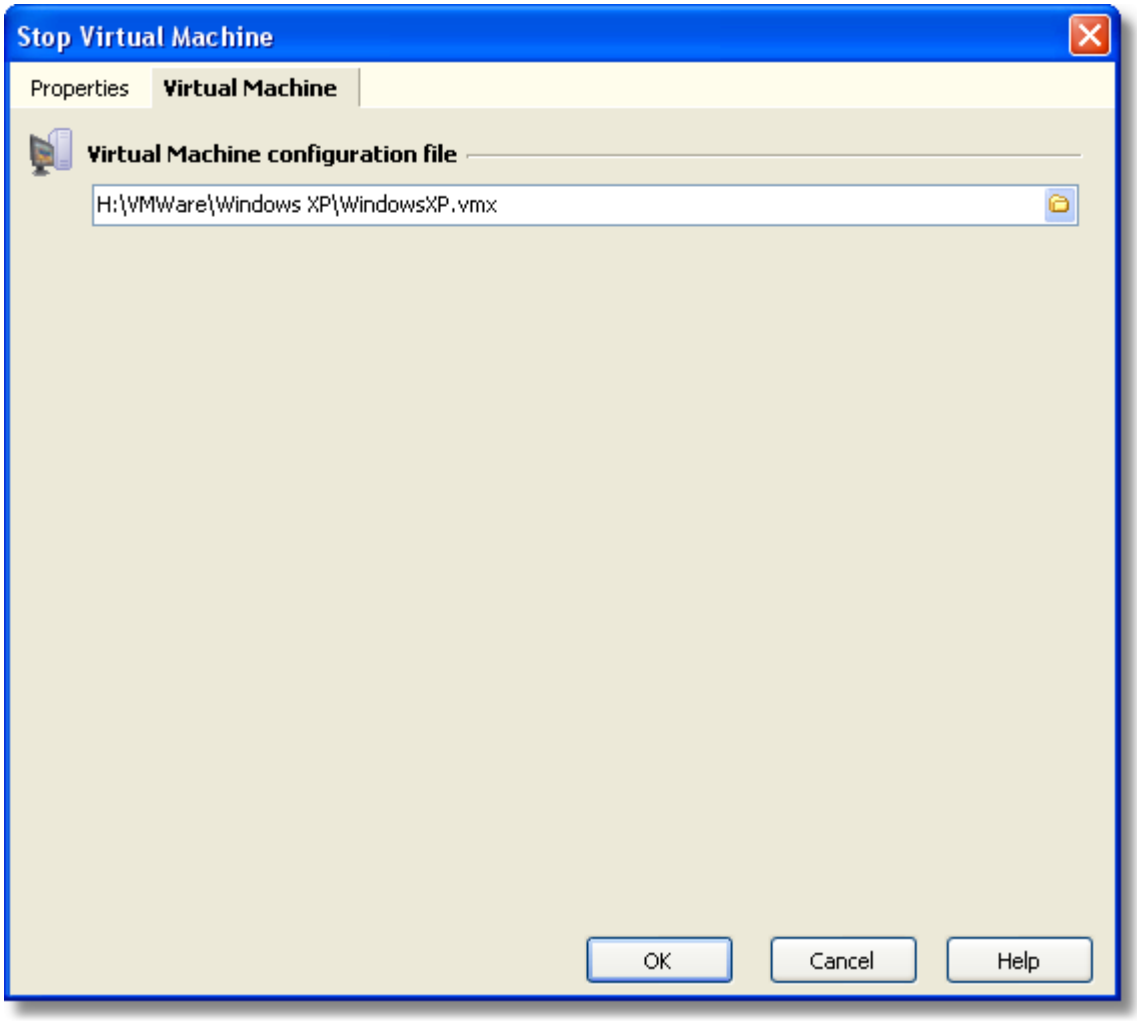
The Virtual Machine must not be already running in a VMWare workstation instance.

5.26.1.2 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.



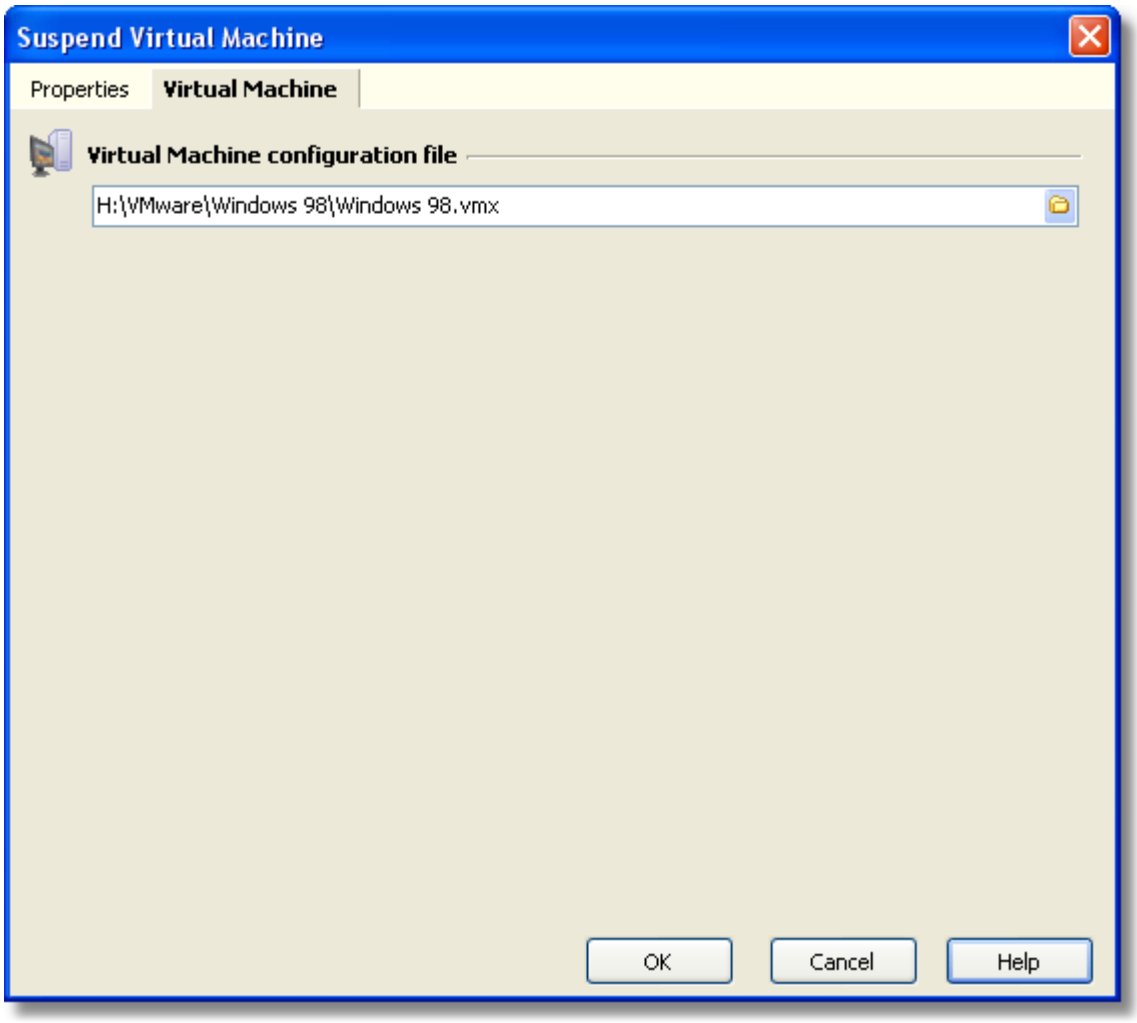
Virtual Machine Configuration File

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).

© 2001-2006 VSoft Technologies Pty Ltd

5.26.1.3 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

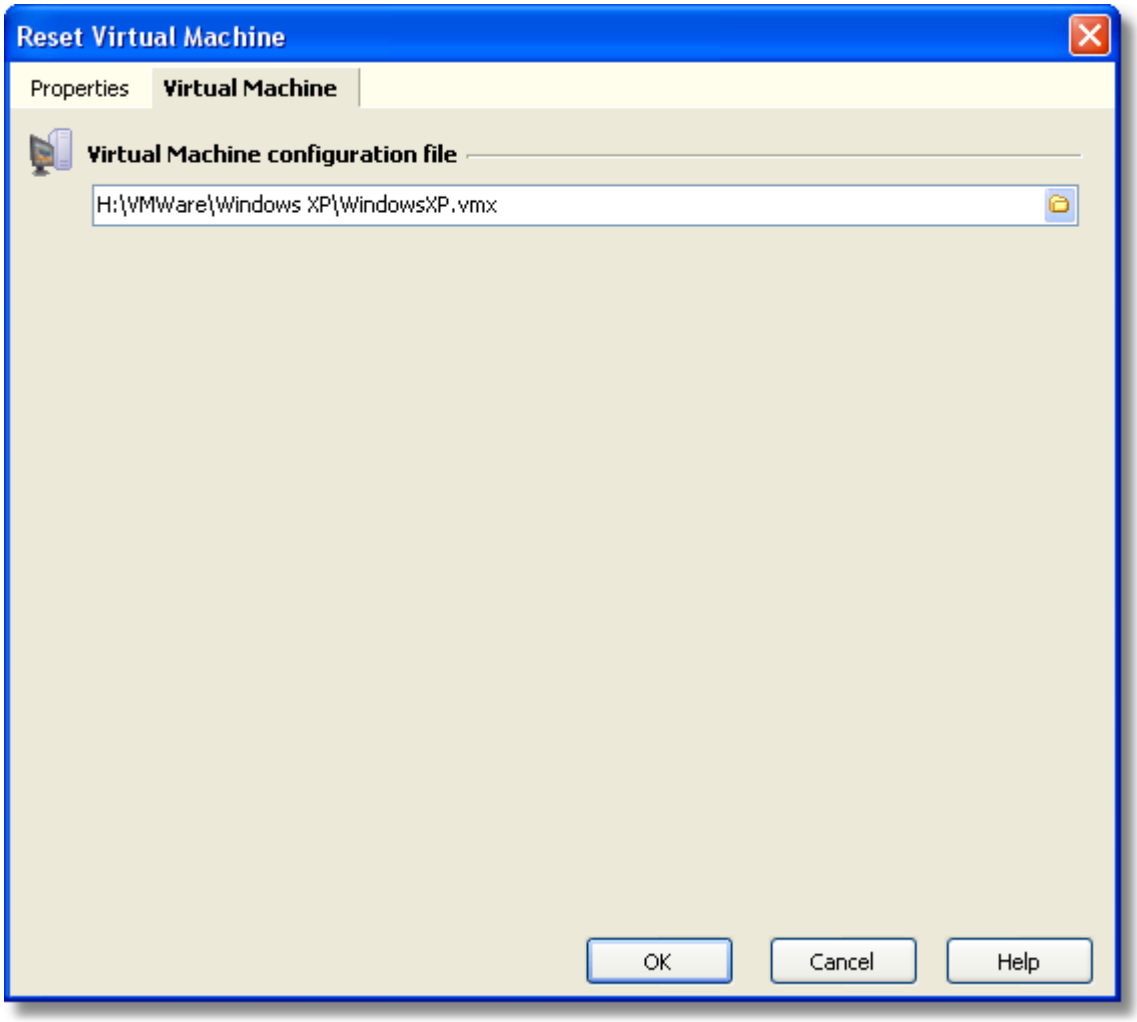
Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team file (.vmtm).

5.26.1.4 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.

© 2001-2006 VSoft Technologies Pty Ltd



Virtual Machine Configuration File

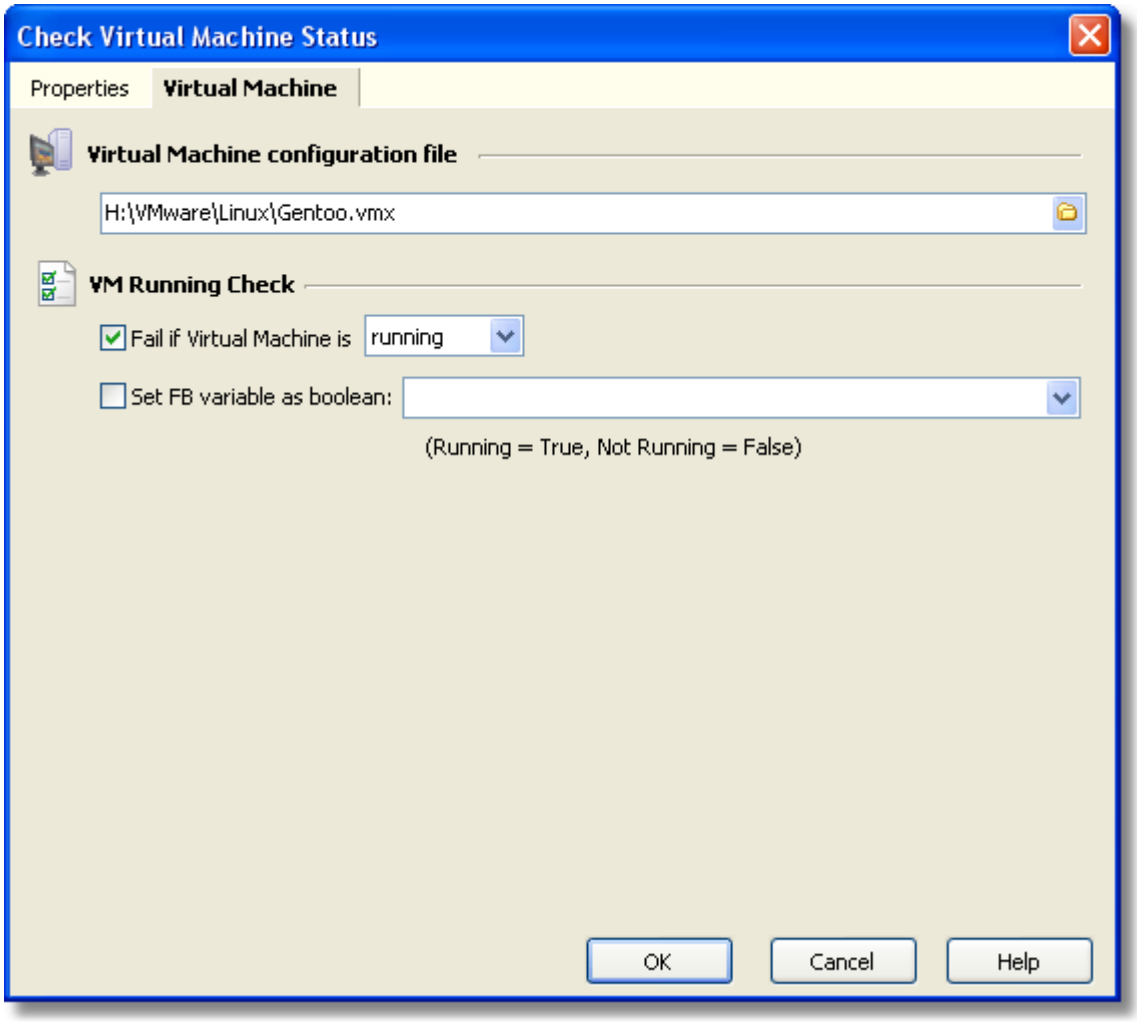
Enter the path to the Virtual Machine's .vmx configuration file, or alternatively (for VMWare Workstation 5.0 users) a Virtual Machine Team file (.vmtm).

5.26.1.5 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

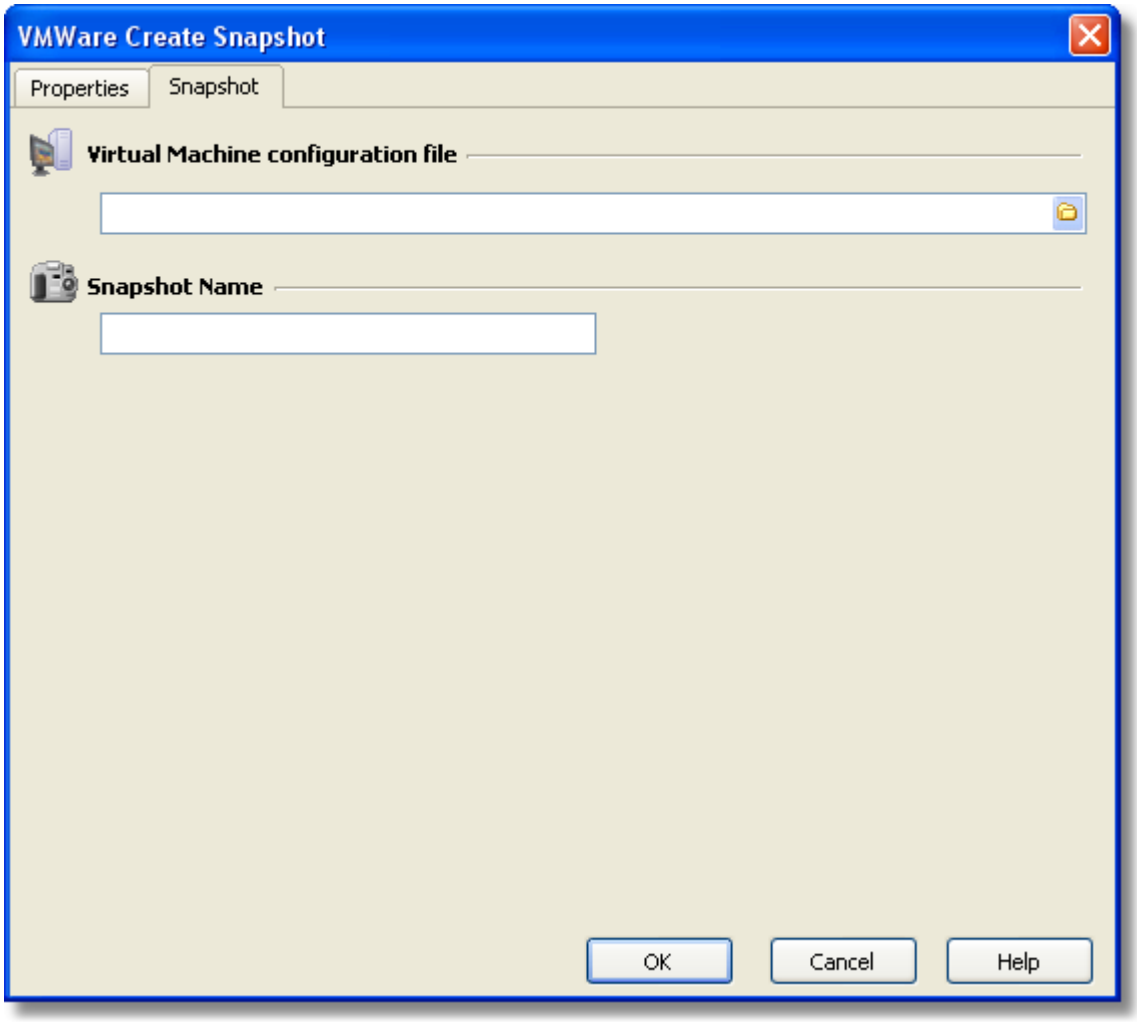
"*Set FB Variable as boolean*"

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.

5.26.1.6 Create Snapshot

The VMWare Create Snapshot action creates a snapshot of a currently running or stopped Virtual Machine.

© 2001-2006 VSoft Technologies Pty Ltd



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.

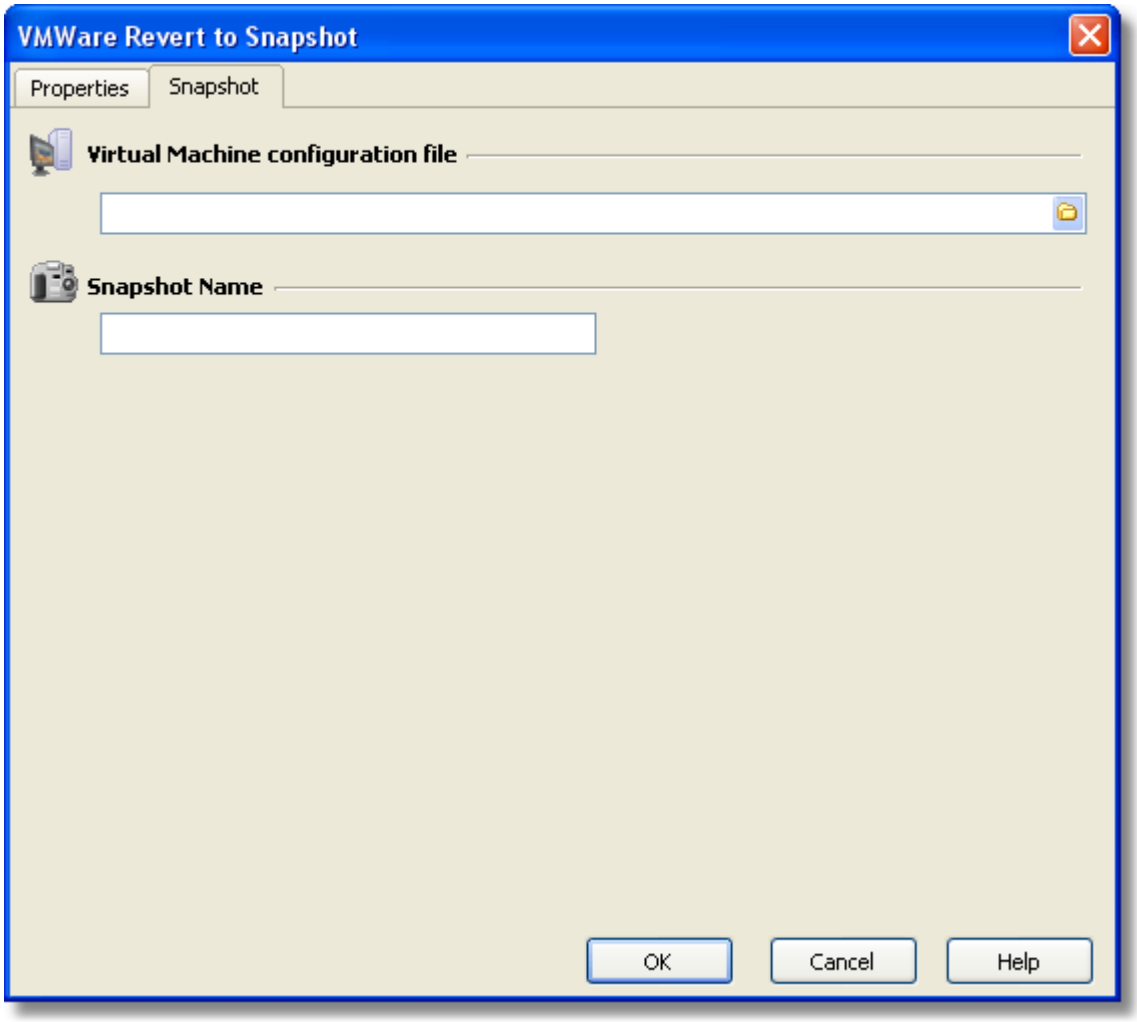
NOTE: In recent versions of VMWare, a hint is displayed when creating background snapshots. If the hint dialog is displayed, it will delay an automated snapshot until OK is pressed. If the build is running unattended, it is advisable to check if this hint is displayed and check the "Do Not Display Again" option if it is so.

5.26.1.7 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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

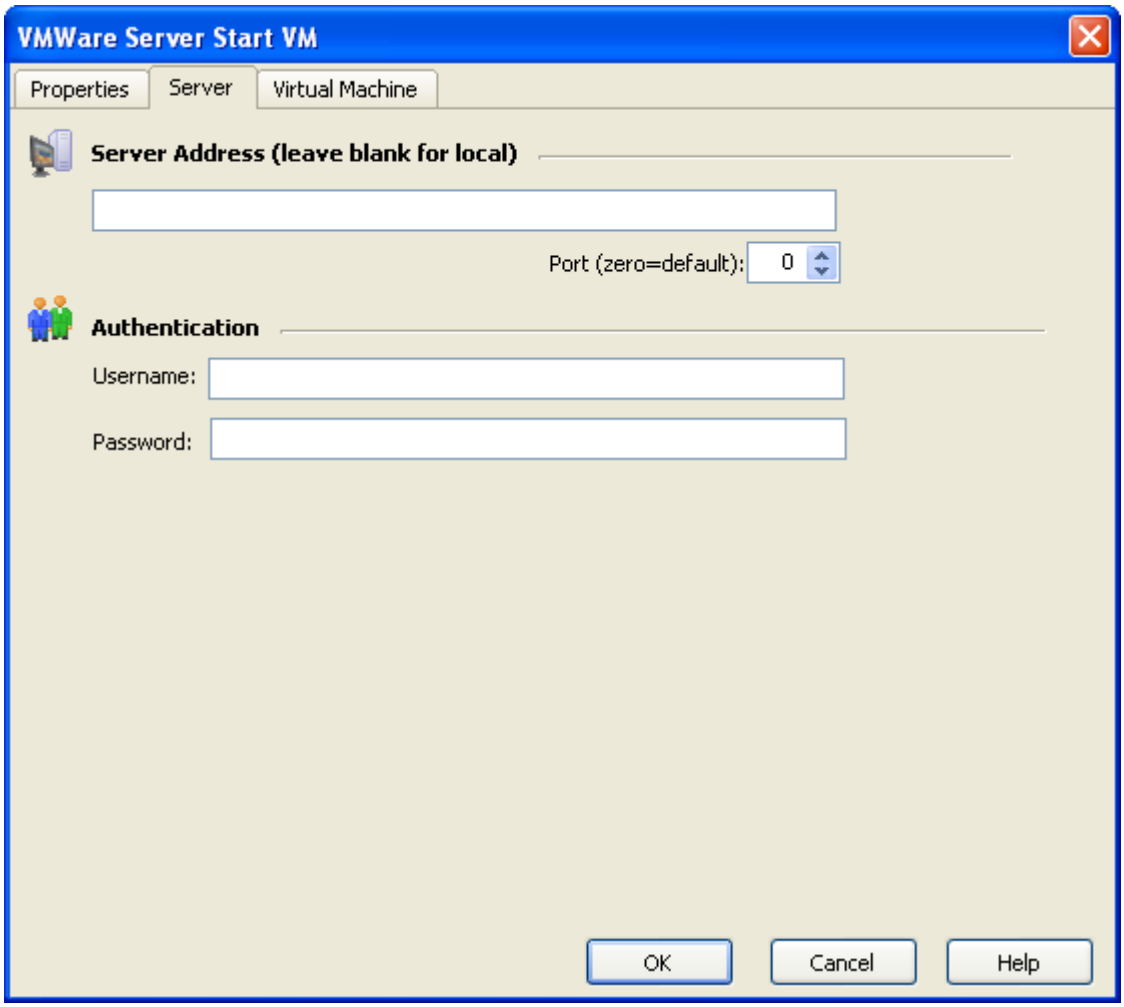
© 2001-2006 VSoft Technologies Pty Ltd

5.26.2 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



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

© 2001-2006 VSoft Technologies Pty Ltd

If you wish to VMWare Server using non-standard credentials, enter them here.

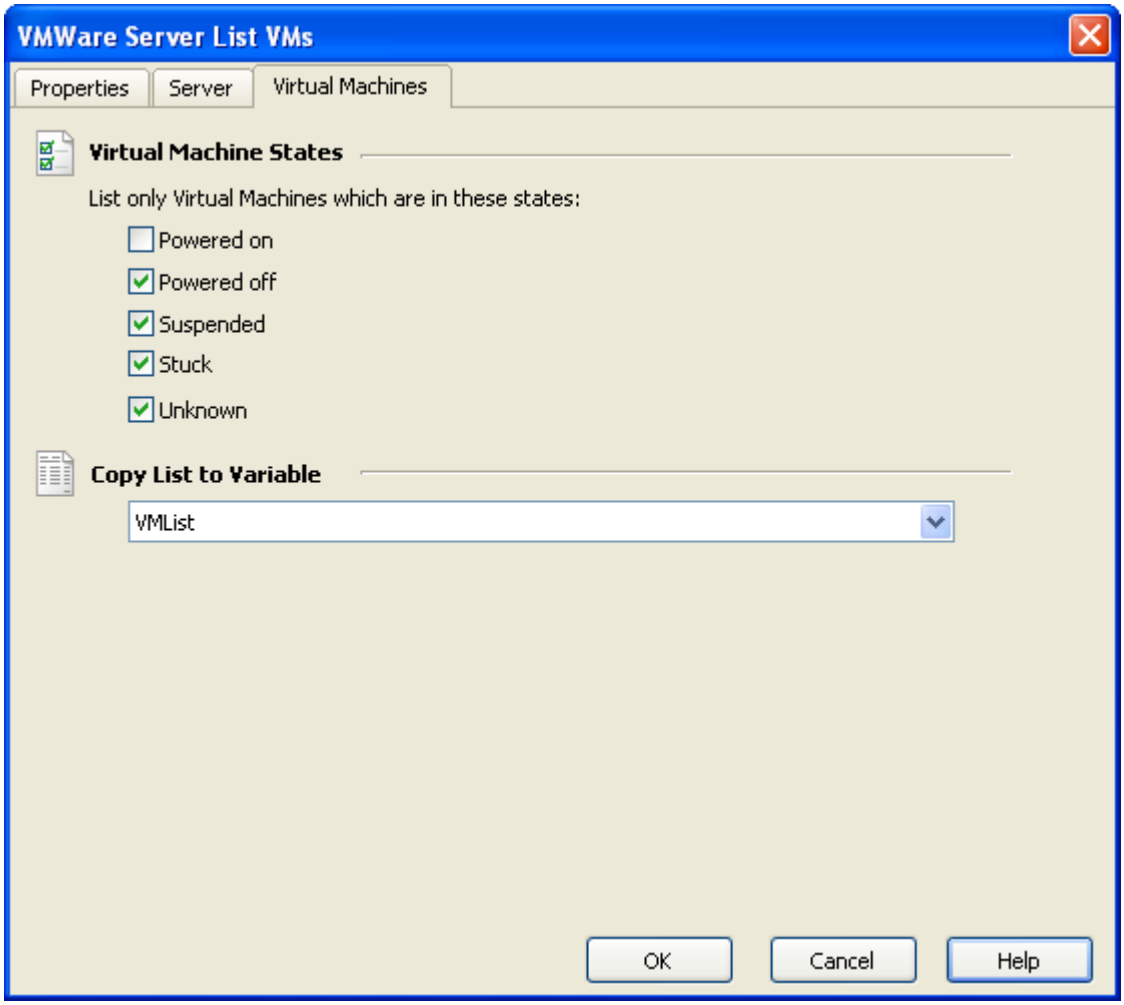
If left blank, the connection will use the credentials of the currently logged in user (recommended.)

5.26.2.1 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

© 2001-2006 VSoft Technologies Pty Ltd

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.

5.26.2.2 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.

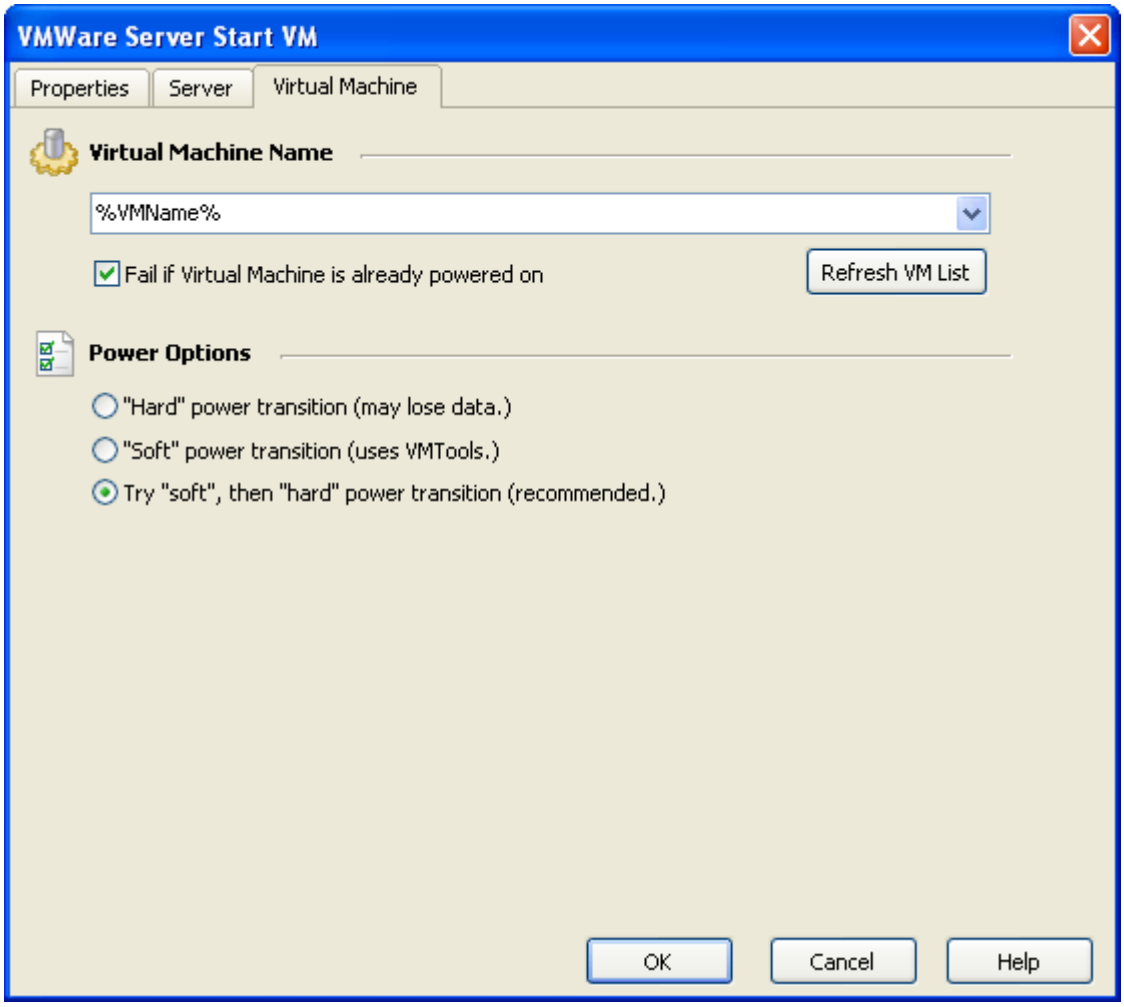
All of these VMWare actions contain the following property pages:

Server Property Page:

See the VMWare Server overview topic.

Virtual Machine Property Page:

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

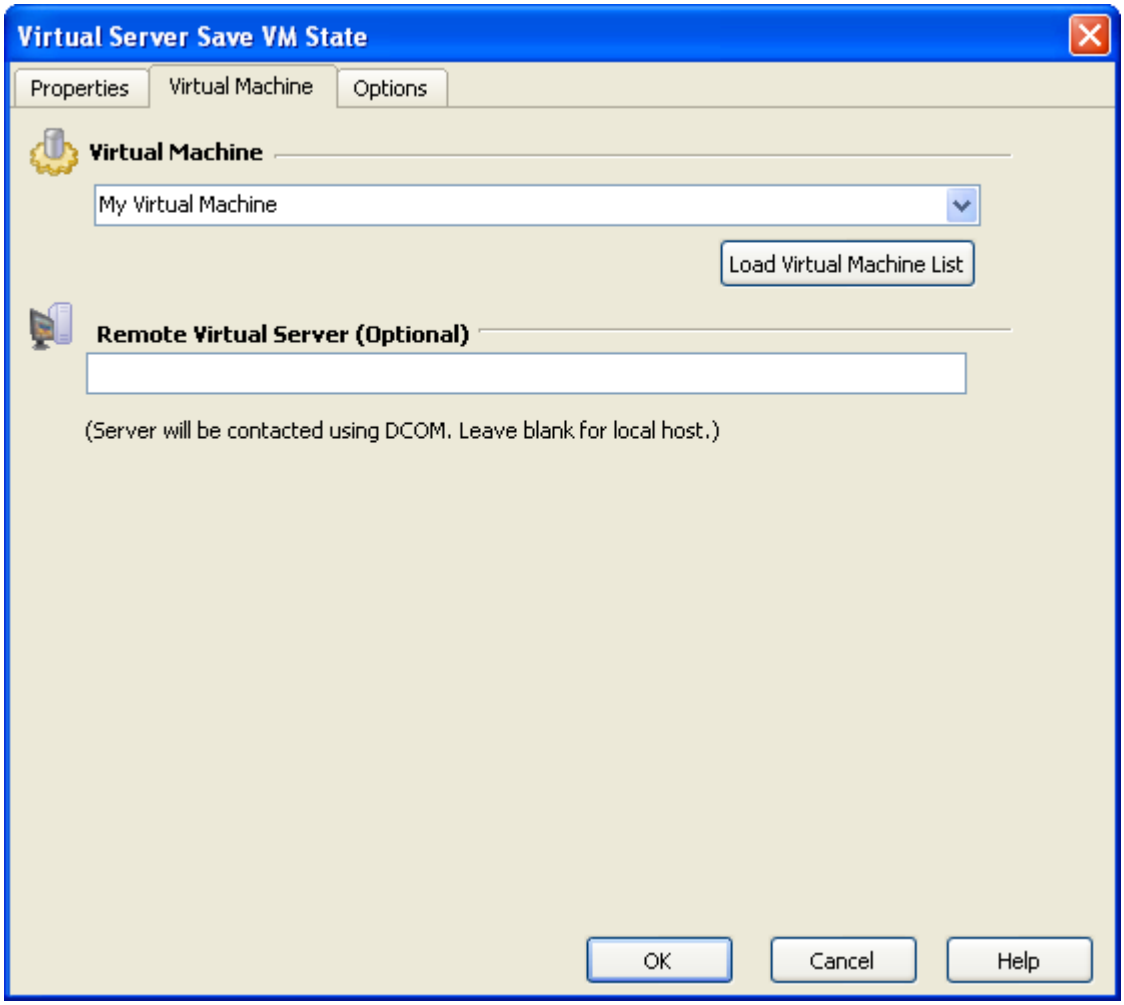
5.26.3 Microsoft Virtual Server

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)

© 2001-2006 VSoft Technologies Pty Ltd

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.

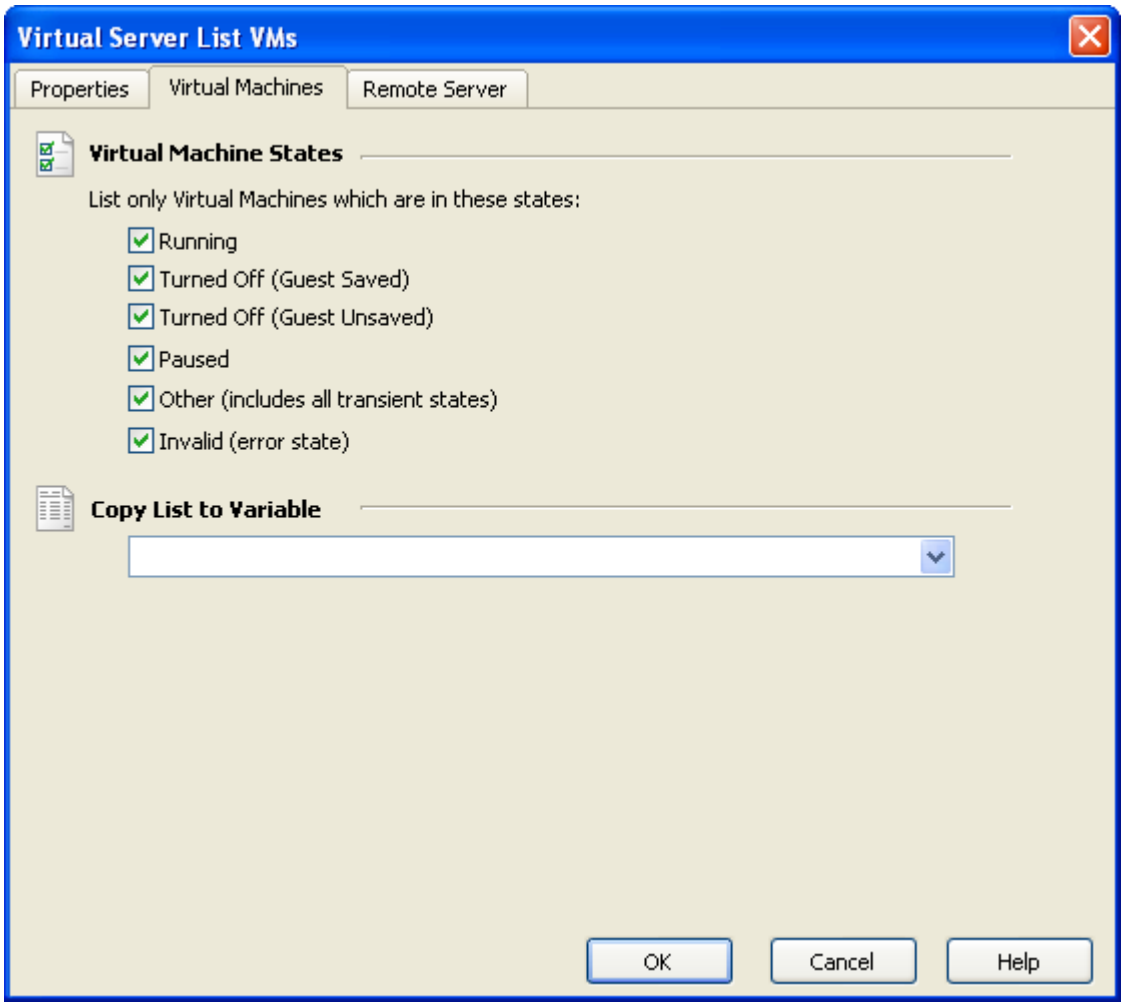
You will still need the Virtual Server COM Client API installed on the local machine (although you do not need a full Virtual Server installation.)

If the Virtual Server is running on the local machine, leave this field blank.

5.26.3.1 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.



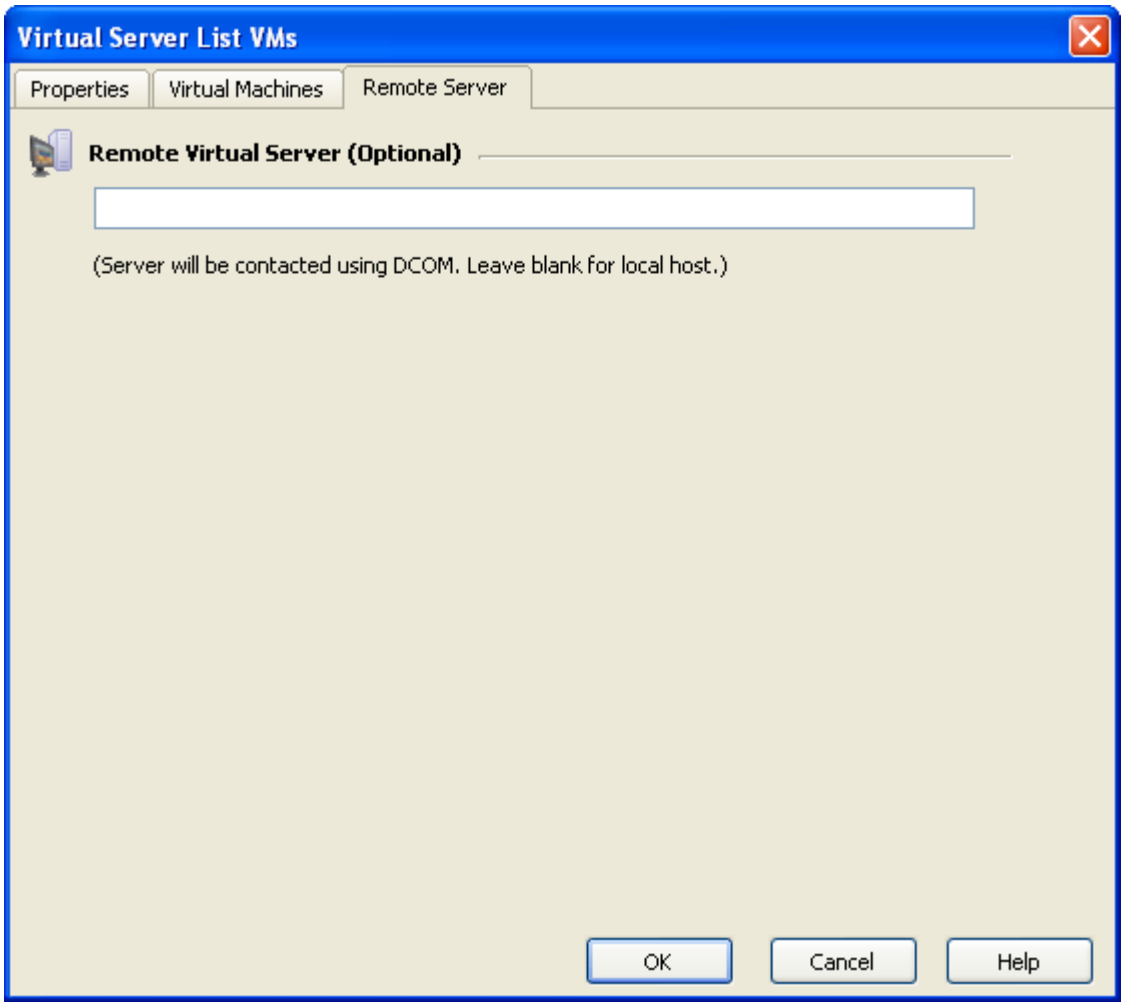
Virtual Machine States

Specify the states that you want included in the list.

© 2001-2006 VSoft Technologies Pty Ltd

Copy List to Variable

Optionally, specify a %ProductName% 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. You will still need the Virtual Server COM API installed on your computer.

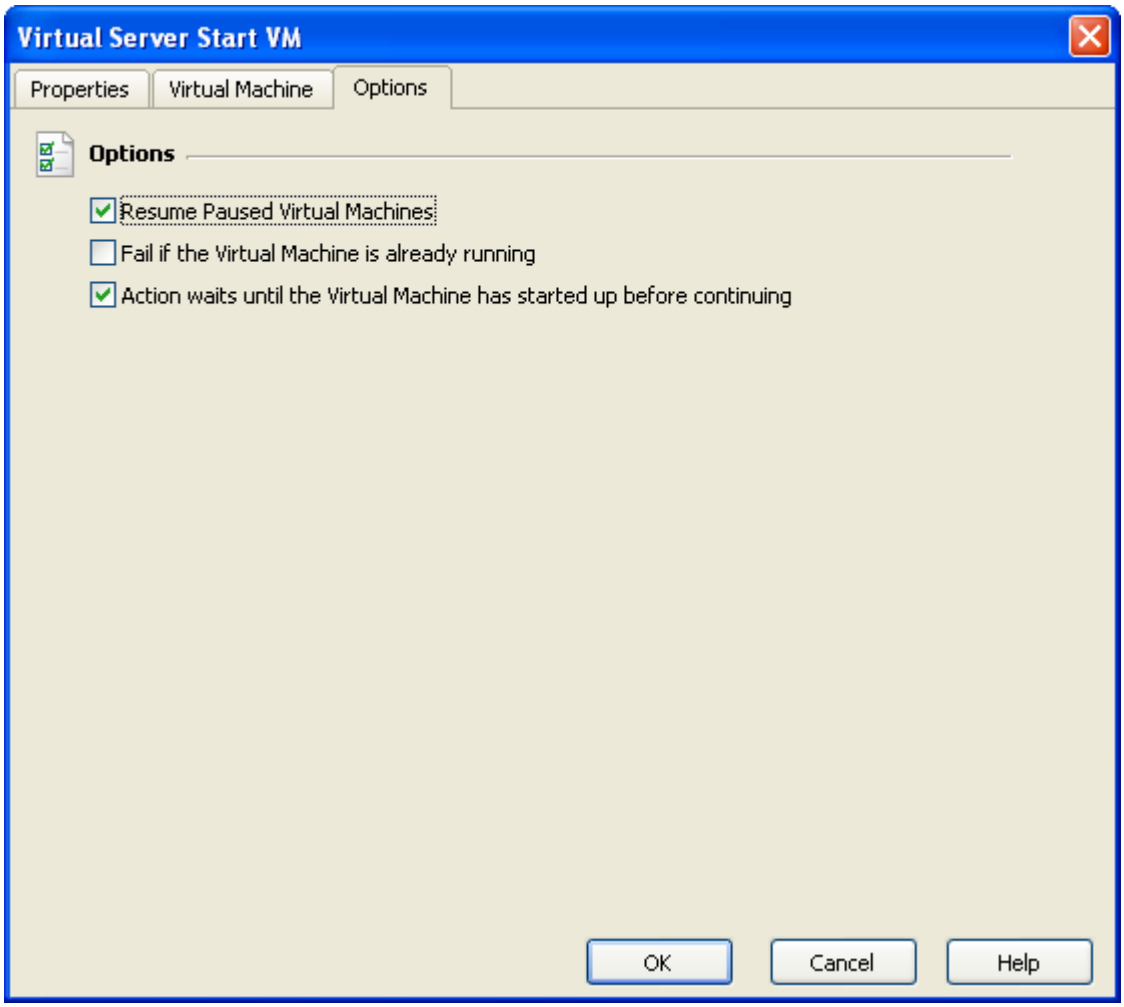
5.26.3.2 Virtual Server Start VM

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.

© 2001-2006 VSoft Technologies Pty Ltd



"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.

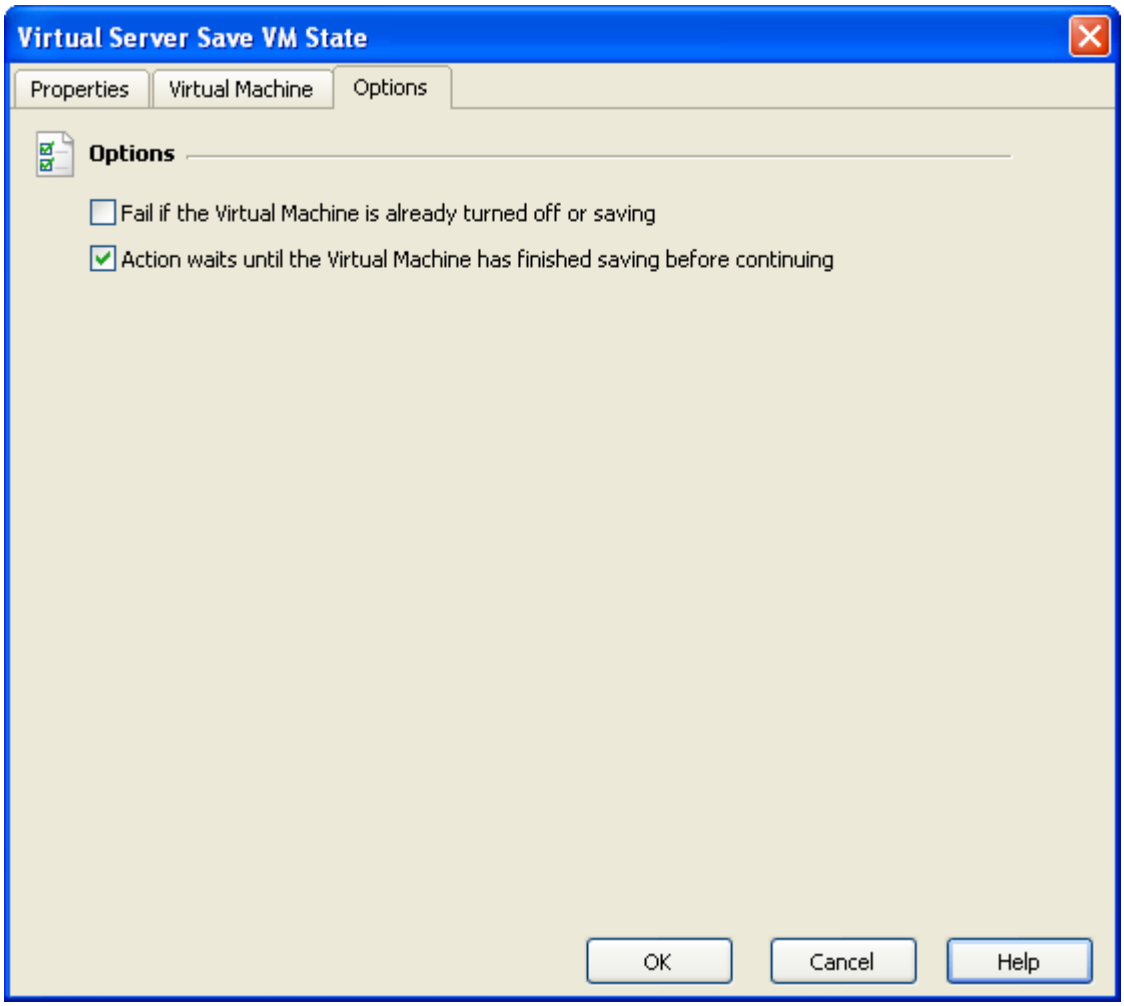
© 2001-2006 VSoft Technologies Pty Ltd

5.26.3.3 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.

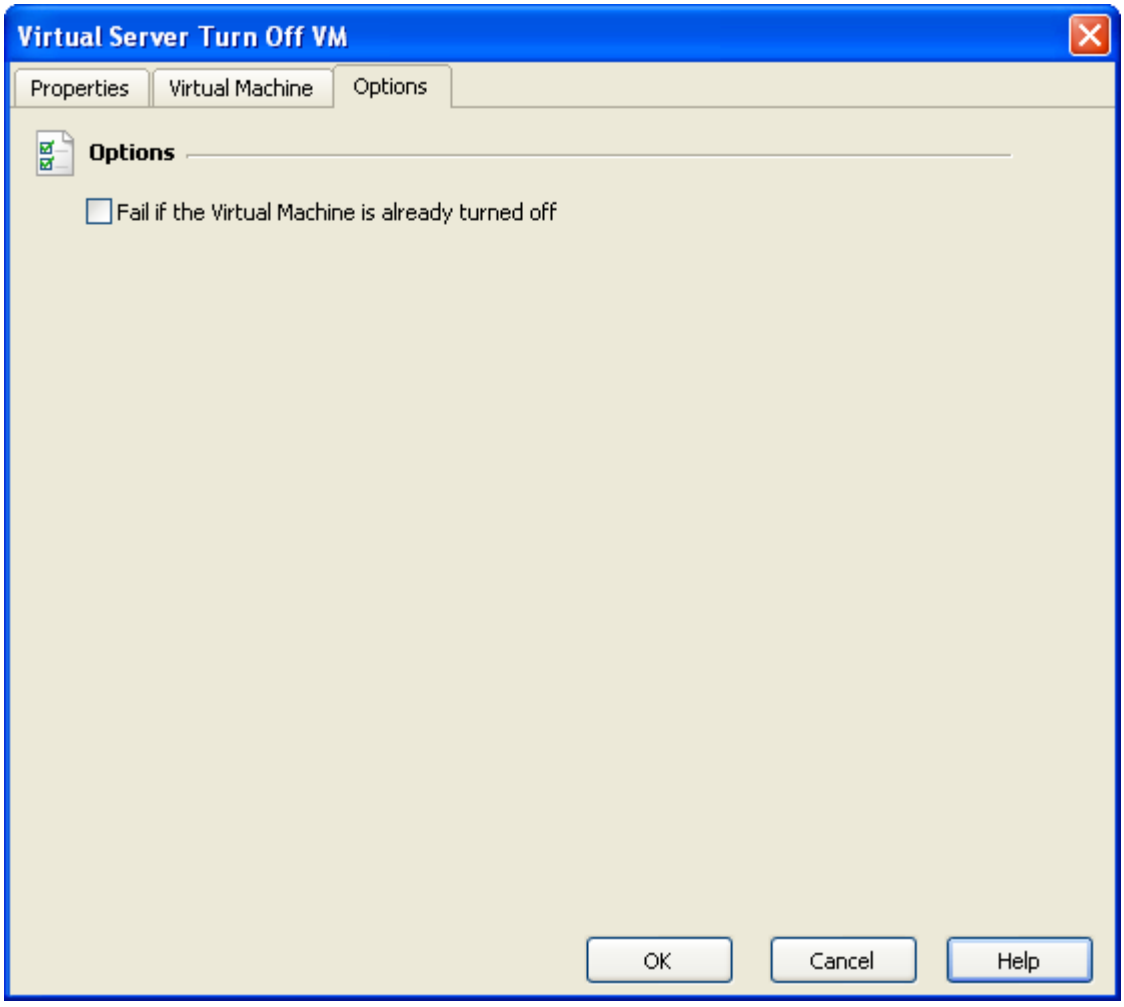
© 2001-2006 VSoft Technologies Pty Ltd

5.26.3.4 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.)

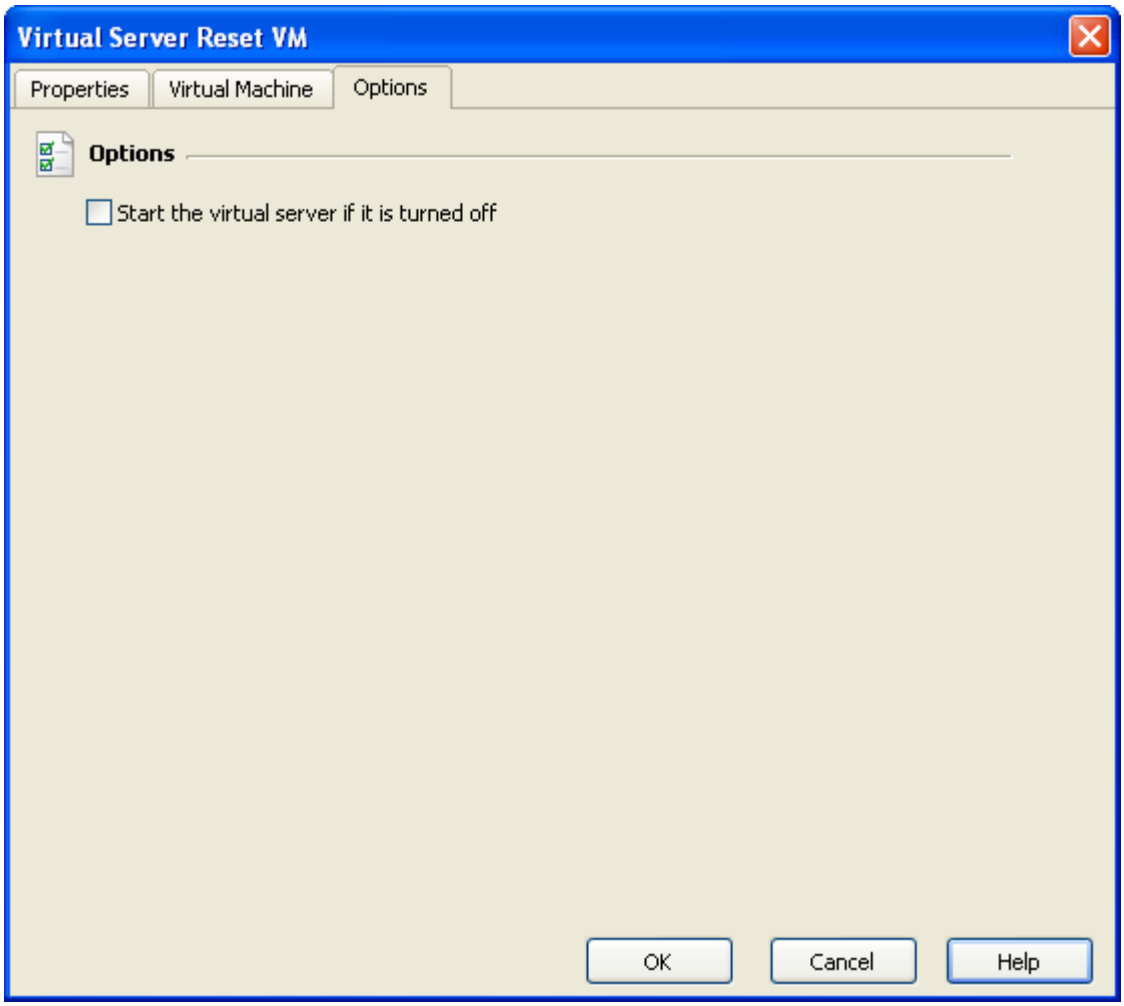
© 2001-2006 VSoft Technologies Pty Ltd

5.26.3.5 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.

© 2001-2006 VSoft Technologies Pty Ltd

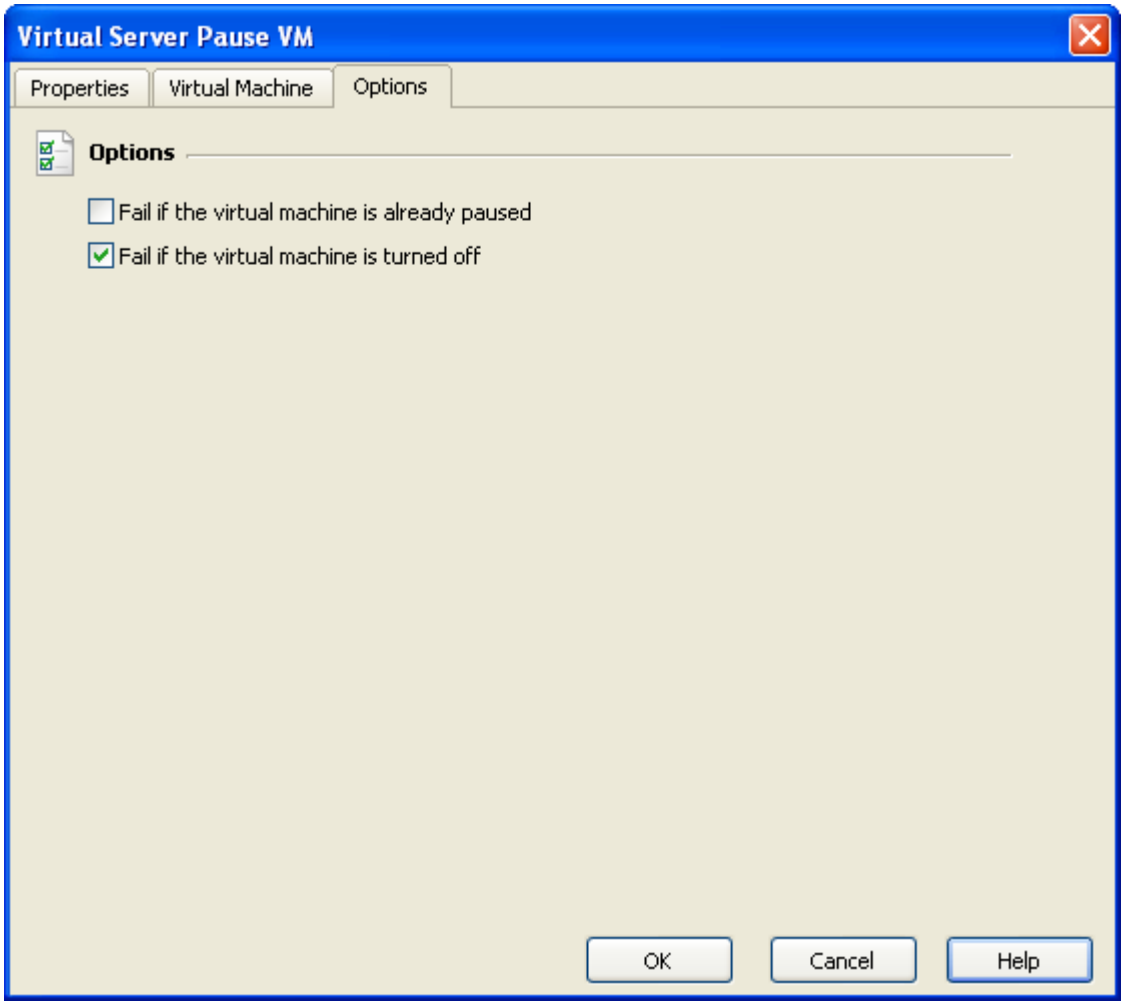
5.26.3.6 Virtual Server Pause VM

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.

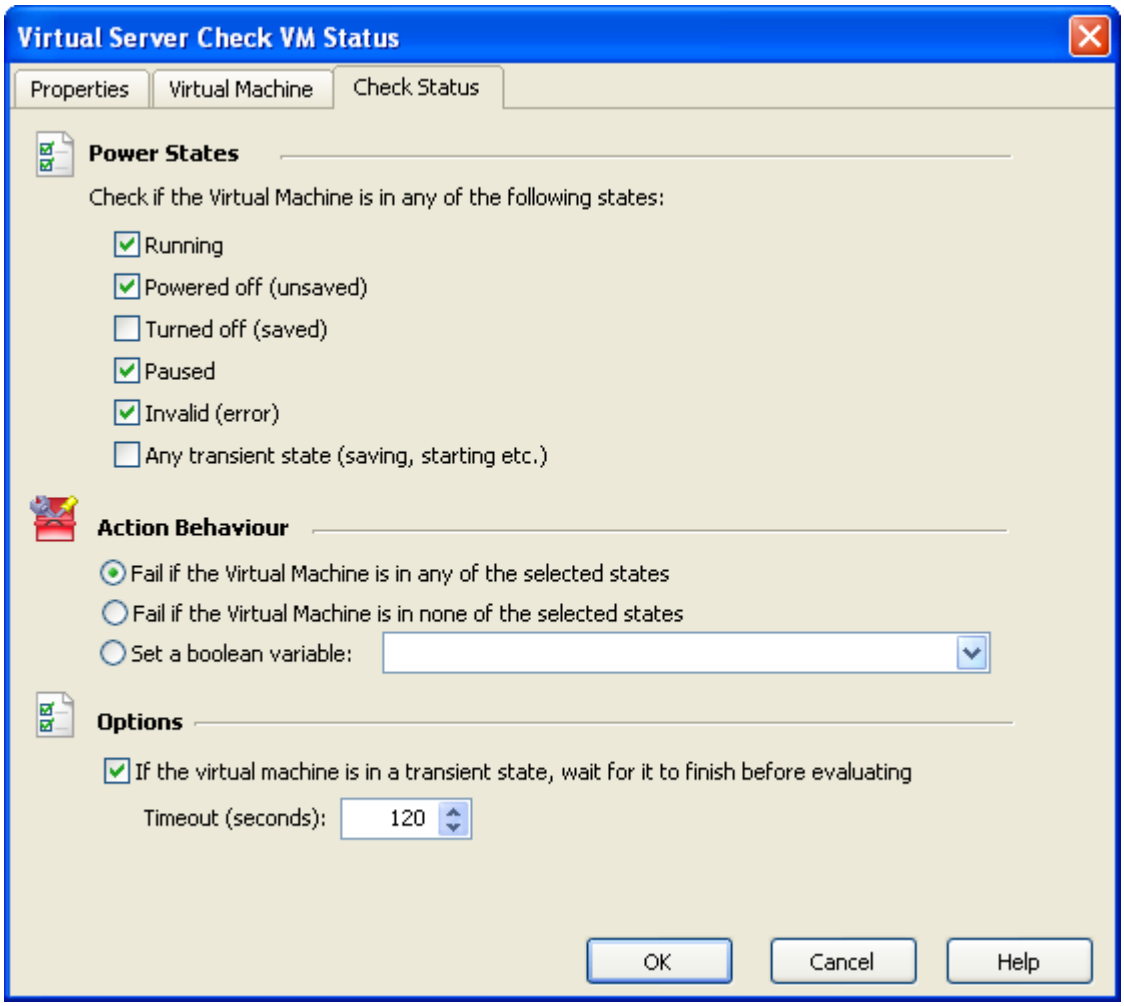
© 2001-2006 VSoft Technologies Pty Ltd

5.26.3.7 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

© 2001-2006 VSoft Technologies Pty Ltd

"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.)

© 2001-2006 VSoft Technologies Pty Ltd

5.27 Internet Information Services (IIS)

The IIS actions allow you to administer Microsoft IIS 5 and IIS 6 servers from your build process. 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.

© 2001-2006 VSoft Technologies Pty Ltd

5.27.1 IIS 5

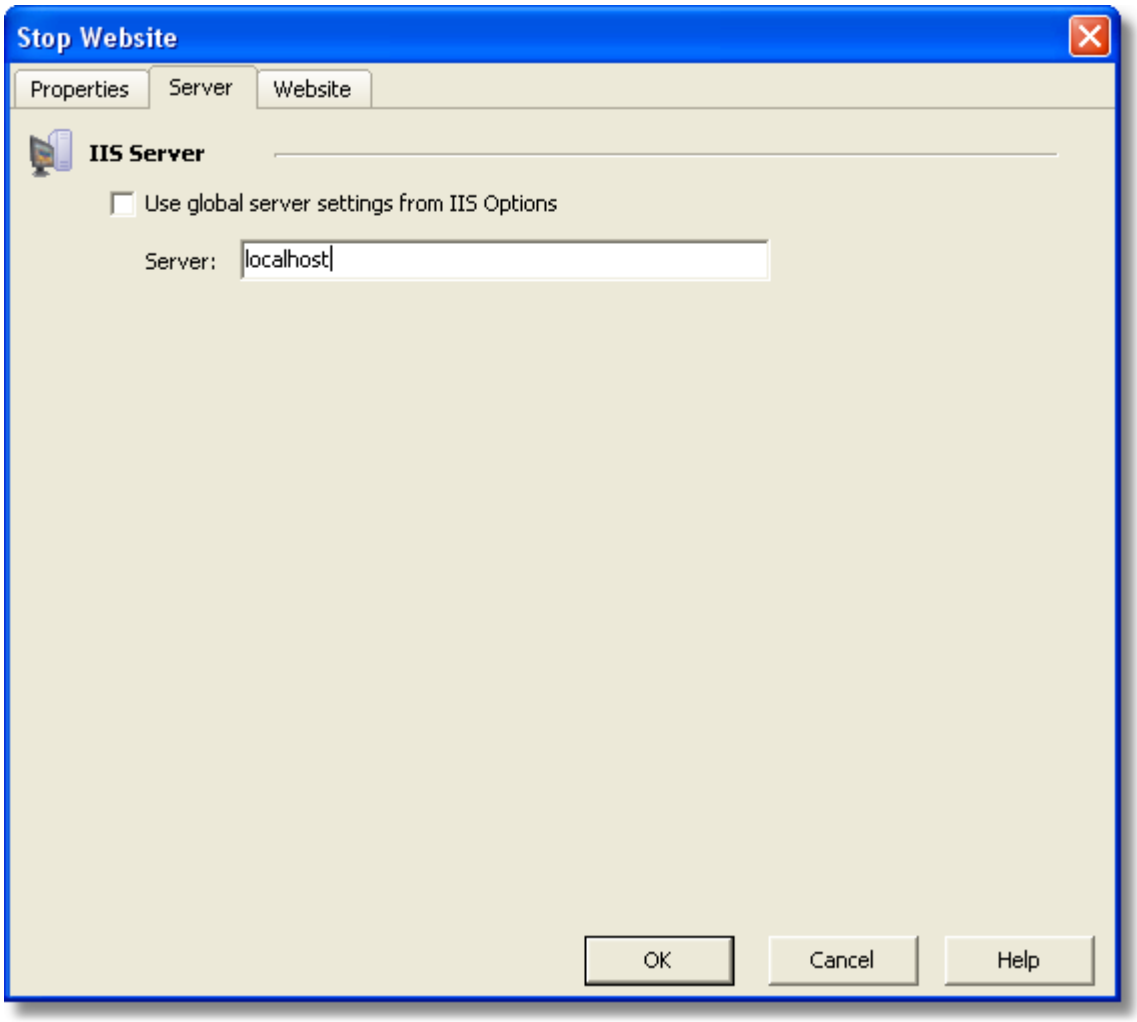
The IIS 5 actions use the ADSI interface to administer features of IIS.

These actions also work with IIS 6.0, but some IIS 6.0 administration features (such as user/password authentication) are not available.

5.27.1.1 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.

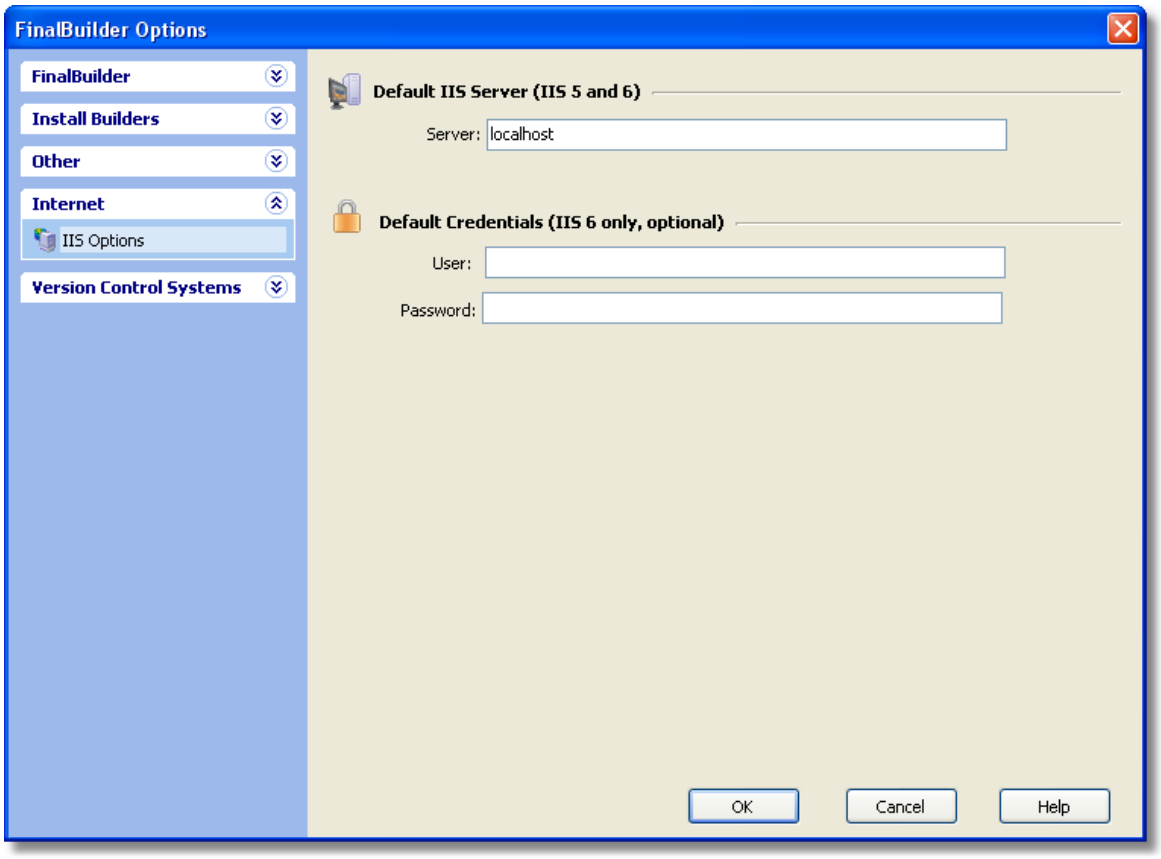
Note that the currently logged in user must have access privileges for the IIS server you wish to administer.



To select the global server name, choose Options from the Tools menu and then click on the IIS Options tab under 'Internet'.

Note that although the Options tab contains options for both a default IIS Server & default credentials, only the "Default IIS Server" option is used by IIS 5.

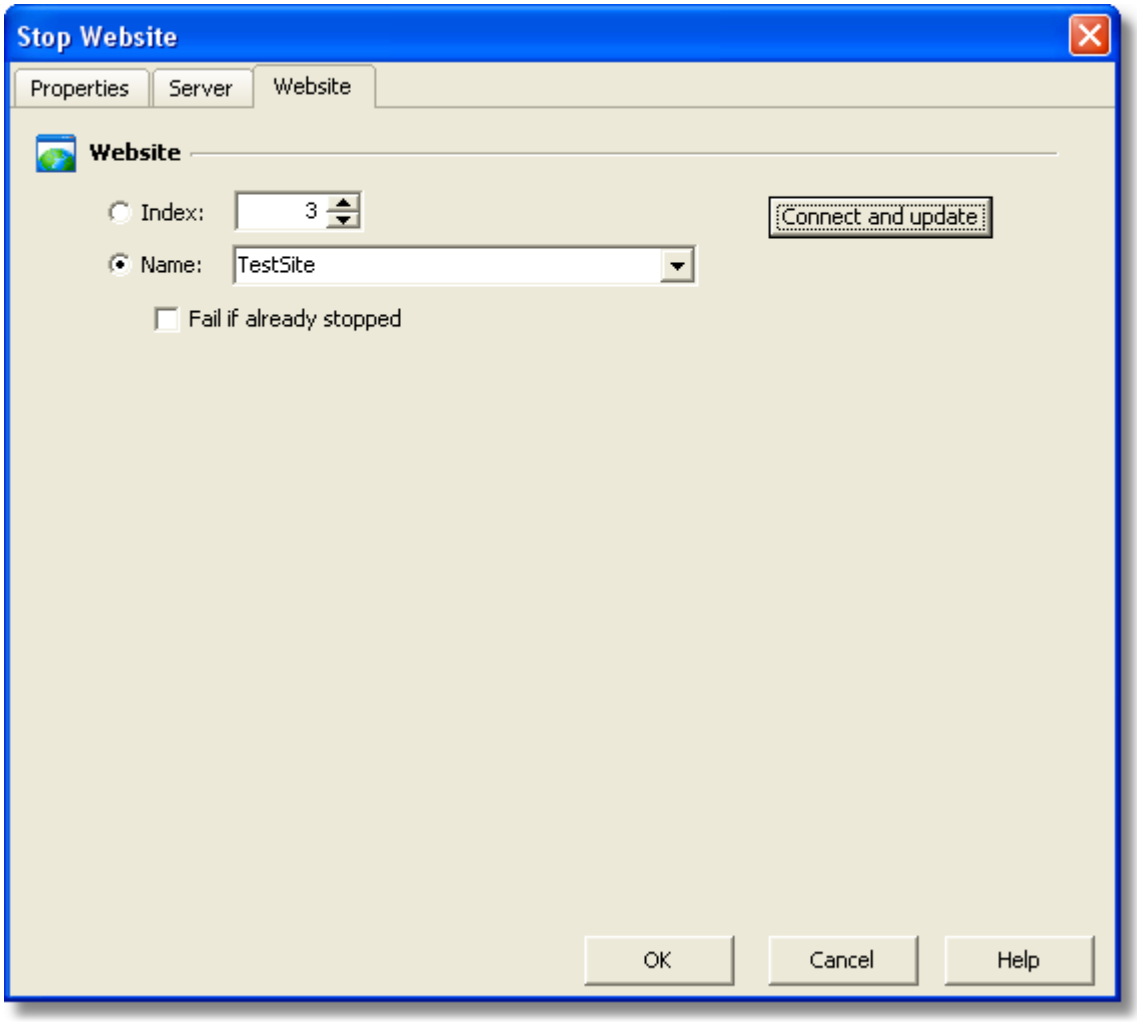
© 2001-2006 VSoft Technologies Pty Ltd



5.27.1.2 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.)

© 2001-2006 VSoft Technologies Pty Ltd



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.

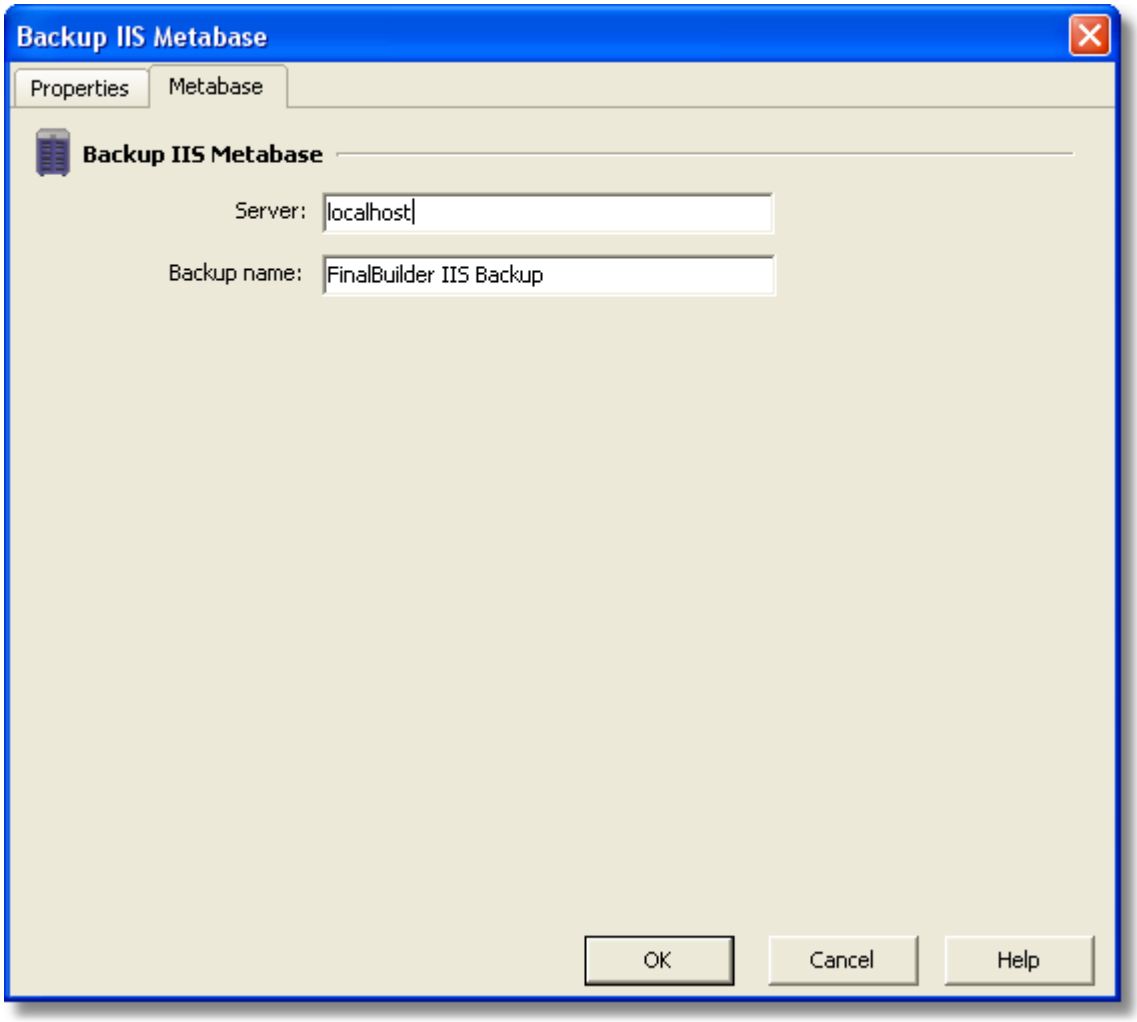
The "**Fail if already...**" checkbox allows you to specify whether some actions will abort if a given action (ie stopping or starting a web site) has already been taken.

5.27.1.3 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".

© 2001-2006 VSoft Technologies Pty Ltd



Server

Specify the name of the server you wish to back up.

Backup name

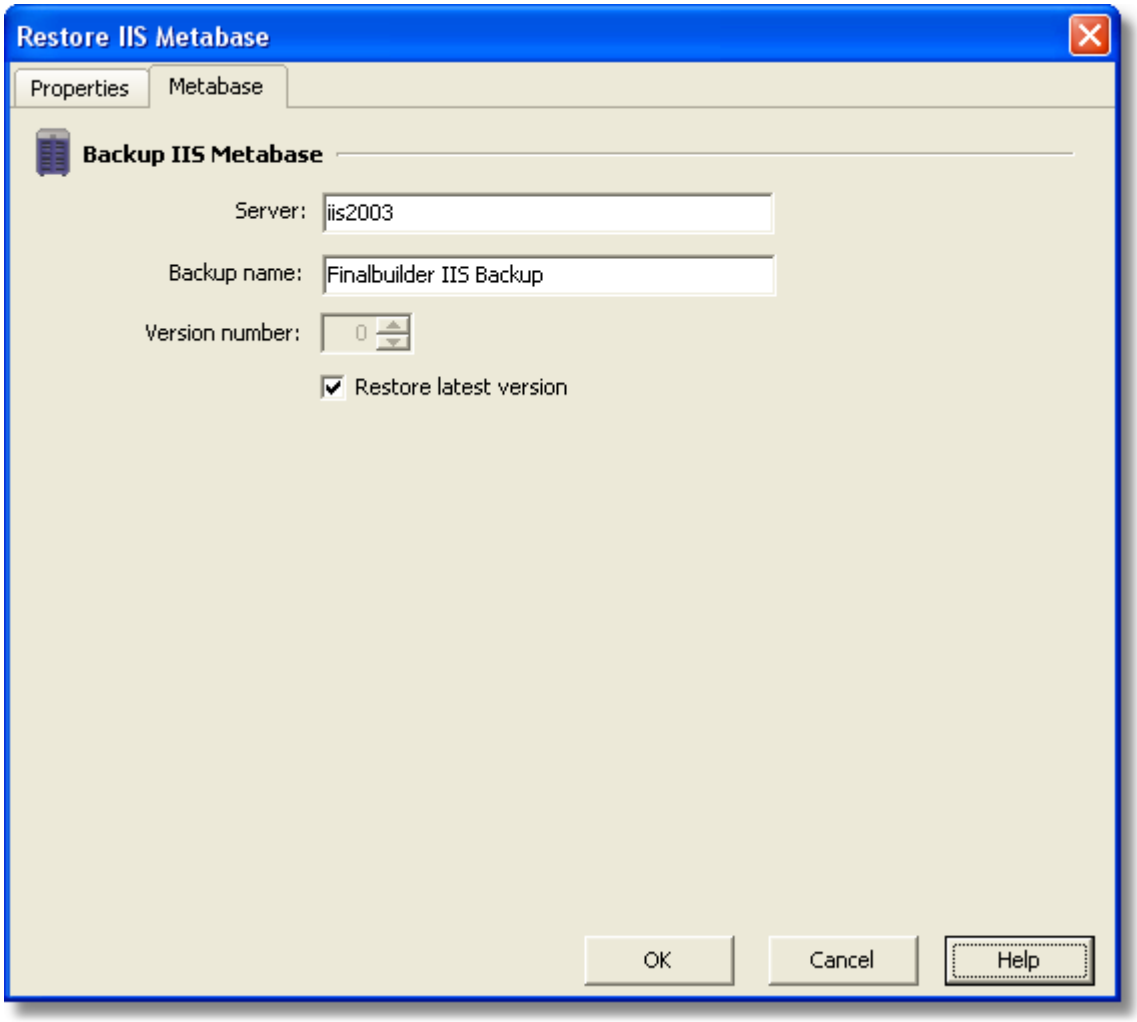
Specify the name you wish to give the backup. Multiple backups with the same name will be given incremental backup version numbers (ie FinalBuilder IIS Backup 0, FinalBuilder IIS Backup 1, FinalBuilder IIS Backup 2...)

5.27.1.4 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".

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.27.1.5 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

© 2001-2006 VSoft Technologies Pty Ltd

the case of the Restart action, the service will be started even if it is initially stopped.)

5.27.1.6 Start / Stop / Pause Website Actions

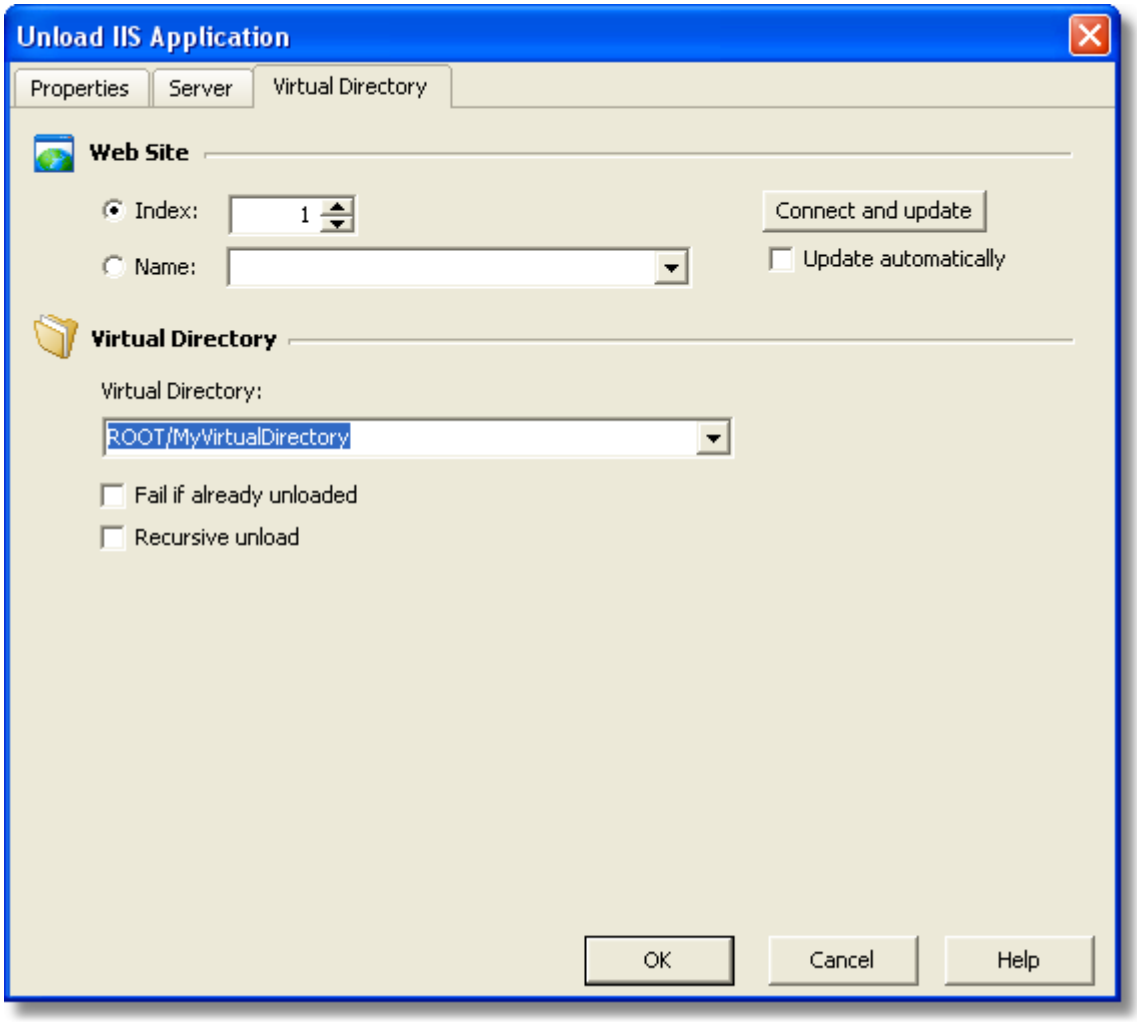
The Start, Stop & Pause Website actions allow you to take individual IIS websites on- and off-line.

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.)

5.27.1.7 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.

© 2001-2006 VSoft Technologies Pty Ltd

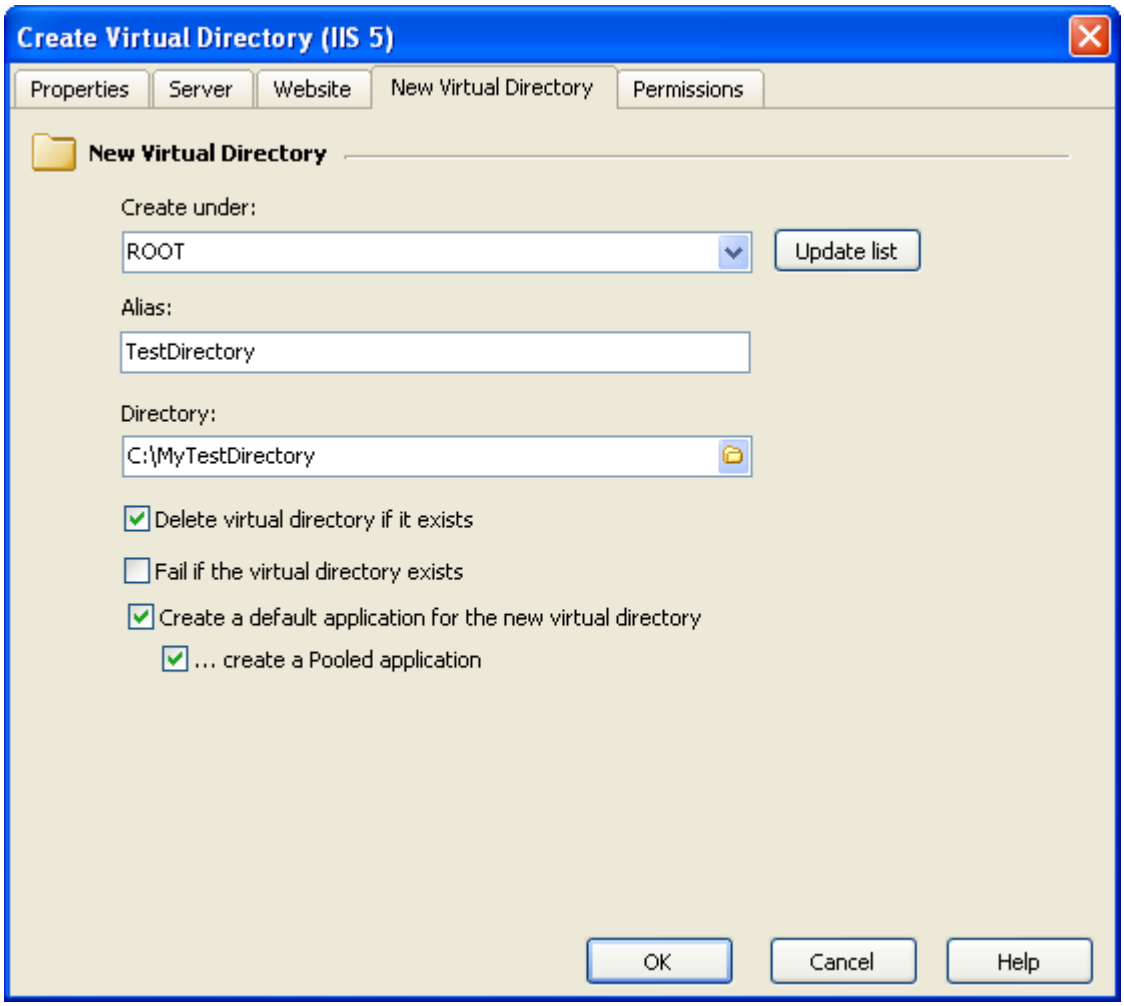
Virtual Directory

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.

5.27.1.8 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

© 2001-2006 VSoft Technologies Pty Ltd

of Virtual Directories from the specified IIS Server.

Alias

Specify the name by which you would like the new directory to be known.

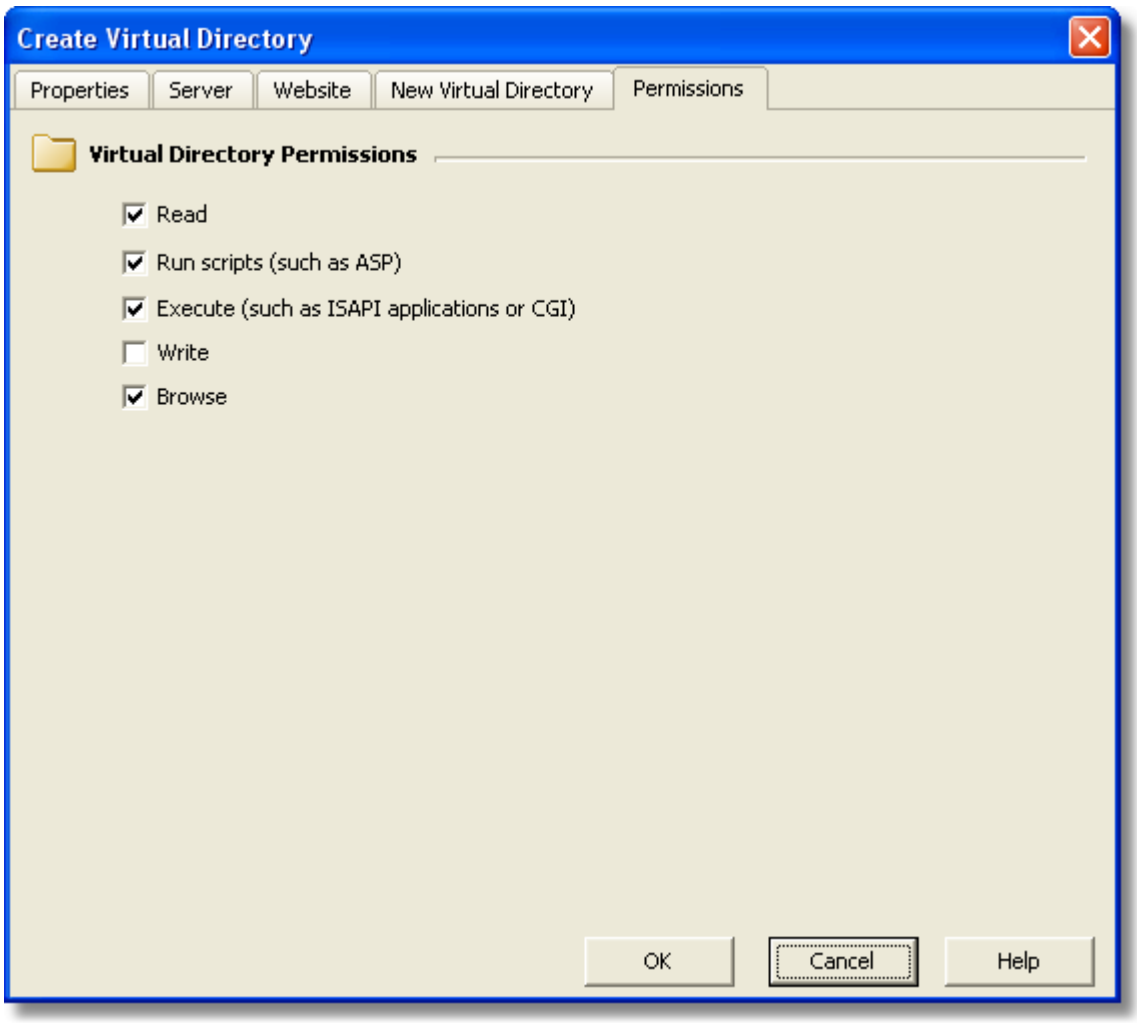
Directory

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

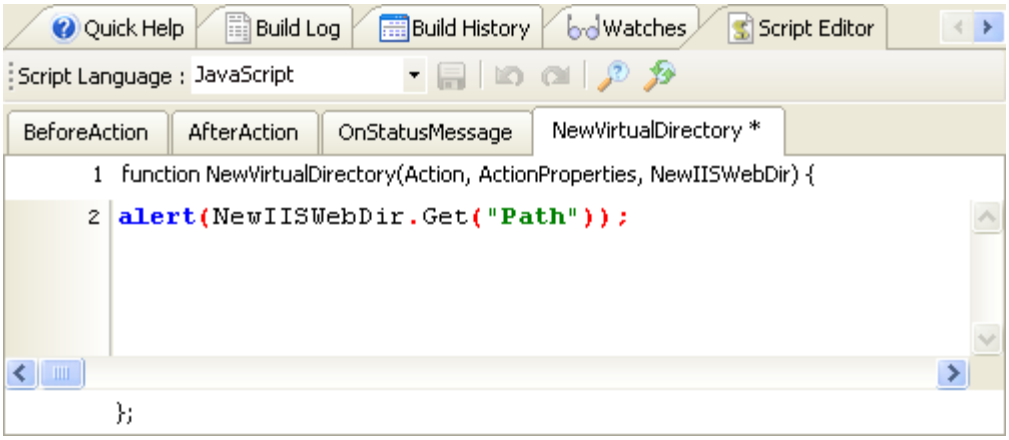


Set the permissions you want IIS users to have on the new Virtual Directory.

© 2001-2006 VSoft Technologies Pty Ltd

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.

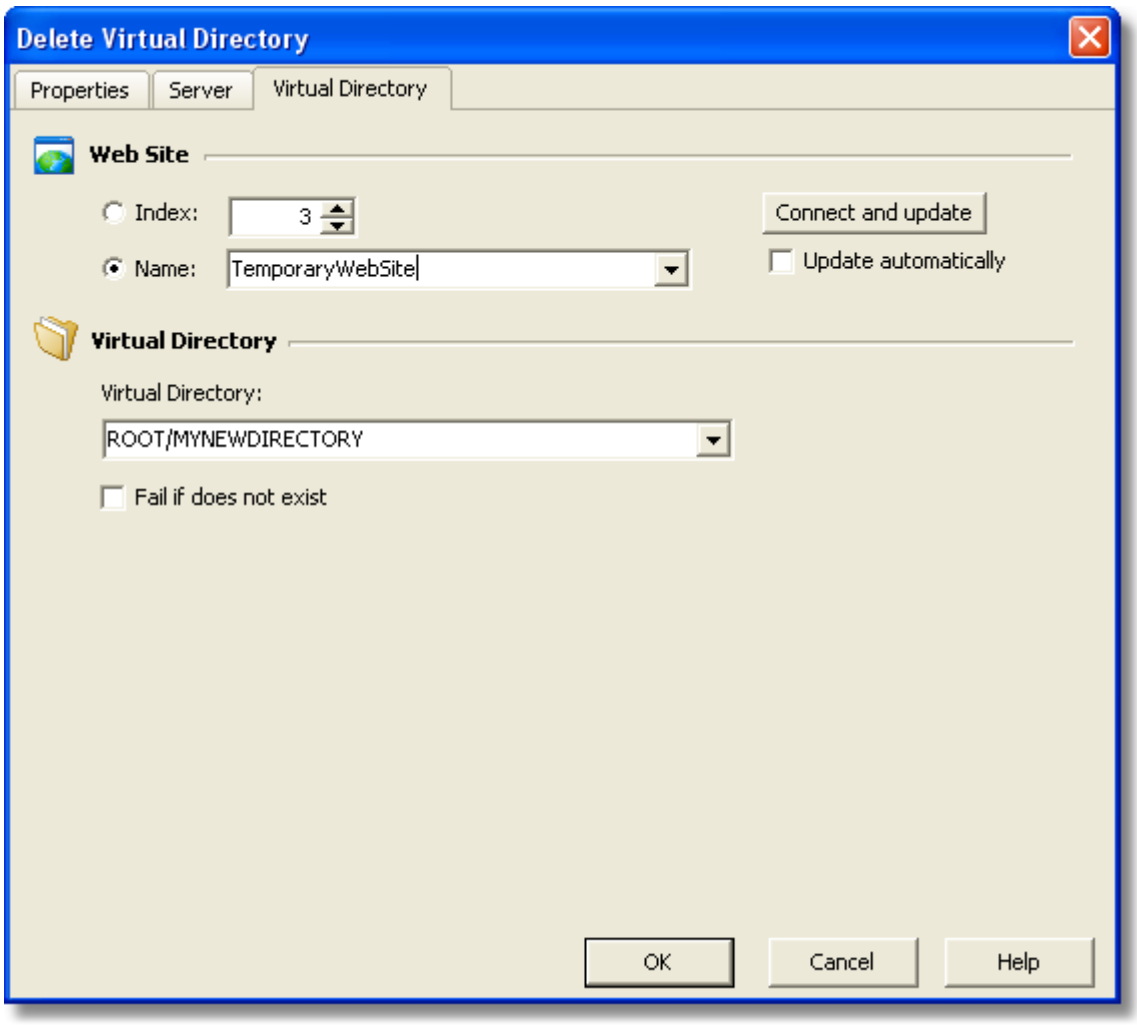


The **NewIISWebDir parameter** is an ADSI container object is of type IIsWebVirtualDir (see the MSDN Documentation for details.) You will need to use the .Get() and .Put() methods to retrieve and set properties on the object. See the MSDN documentation for a full list of property names.

It is not necessary to call the .SetInfo() method to save any changes, this is called for you when the event exits.

© 2001-2006 VSoft Technologies Pty Ltd

5.27.1.9 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.

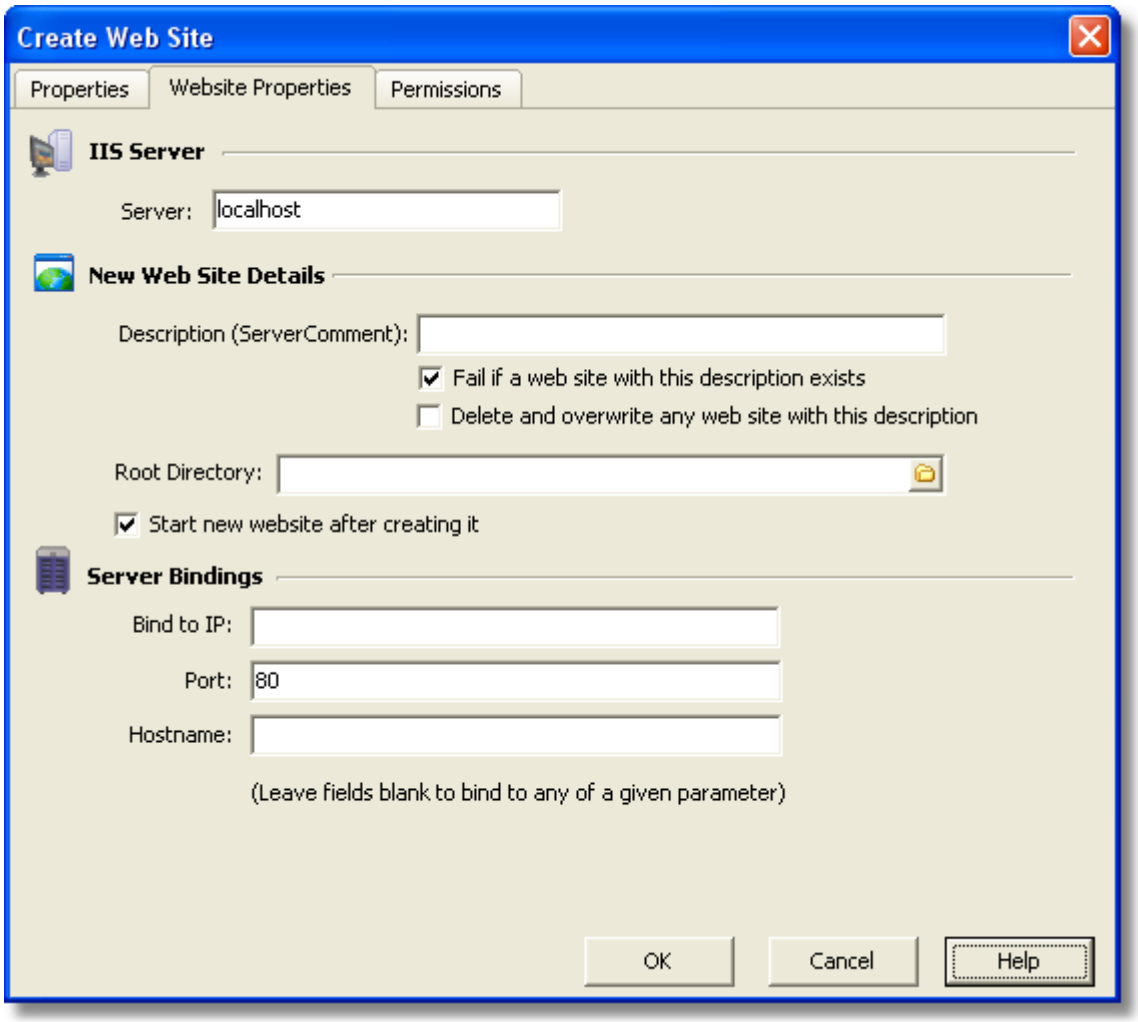
Fail if does not exist

Check this box if you want the action to fail should the specified virtual directory not exist.

© 2001-2006 VSoft Technologies Pty Ltd

5.27.1.10 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.



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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

5.27.1.11 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.

© 2001-2006 VSoft Technologies Pty Ltd

5.27.2 IIS 6

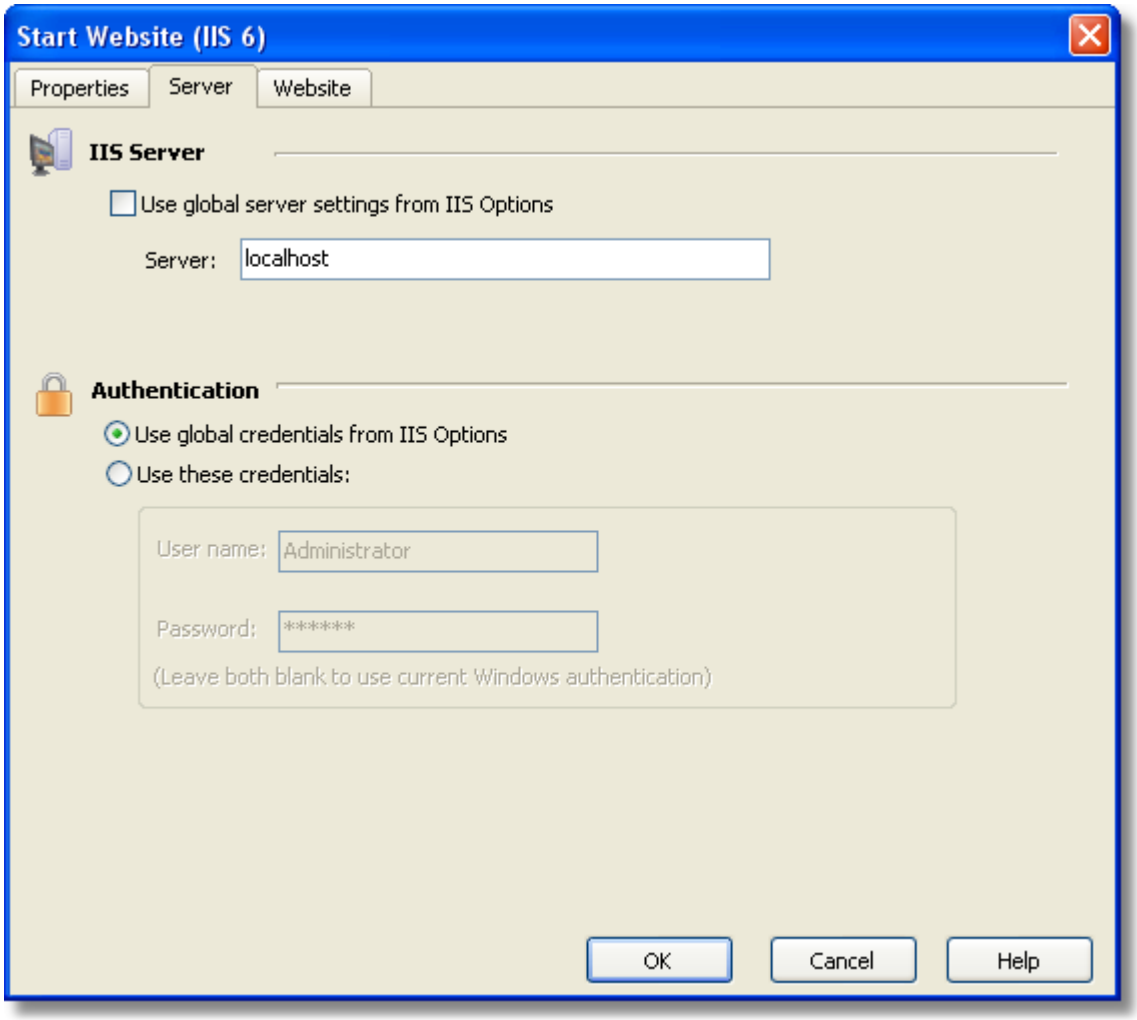
The IIS 6 actions use the WMI interface to administer features of IIS.

(The WMI interface is not available in IIS versions below 6.0.)

5.27.2.1 Server Selection

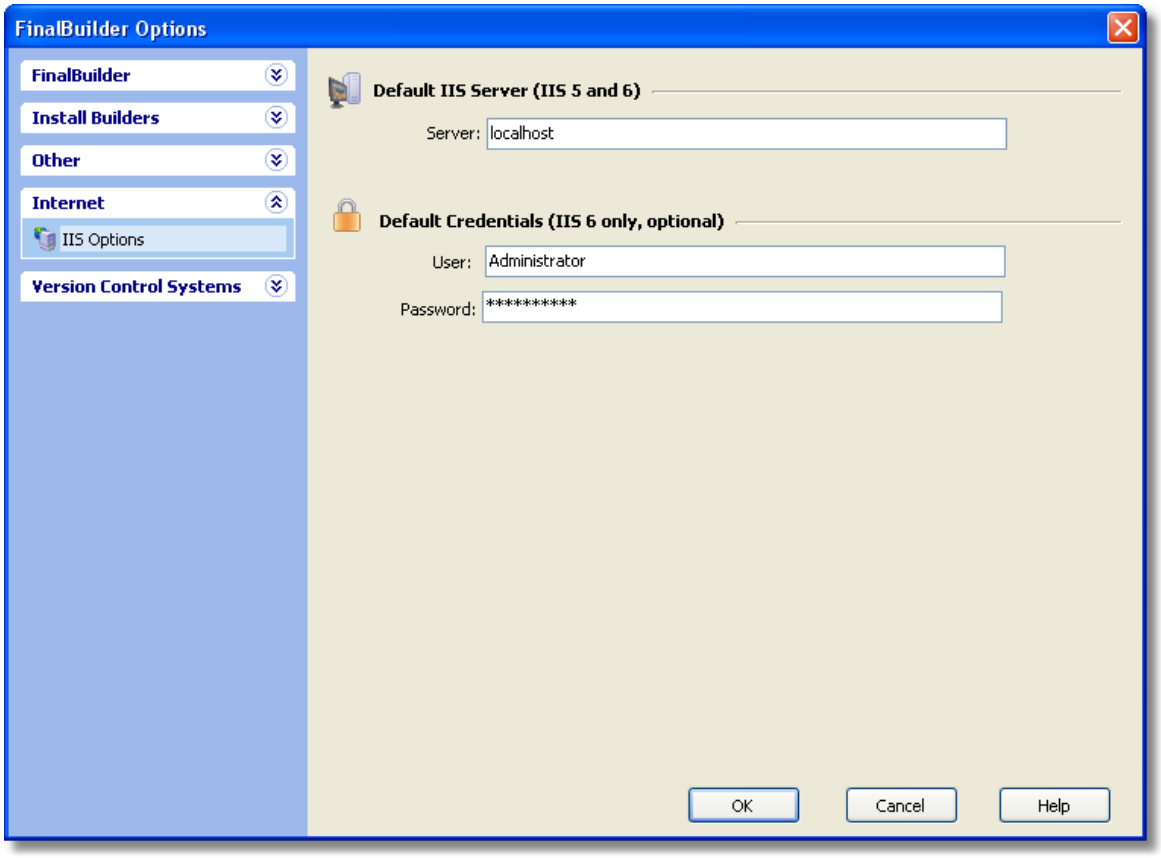
For each IIS action, you can specify the IIS Server as either a global server (specified in the options) or a particular server name. You can also specify credentials as either the credentials specified in the Options, or a particular set of credentials.

If the credentials are left blank, Windows authentication with the credentials of the currently logged-in user will be used.



To select the global server name and credentials, choose Options from the Tools menu and then click on the IIS Options tab under 'Internet':

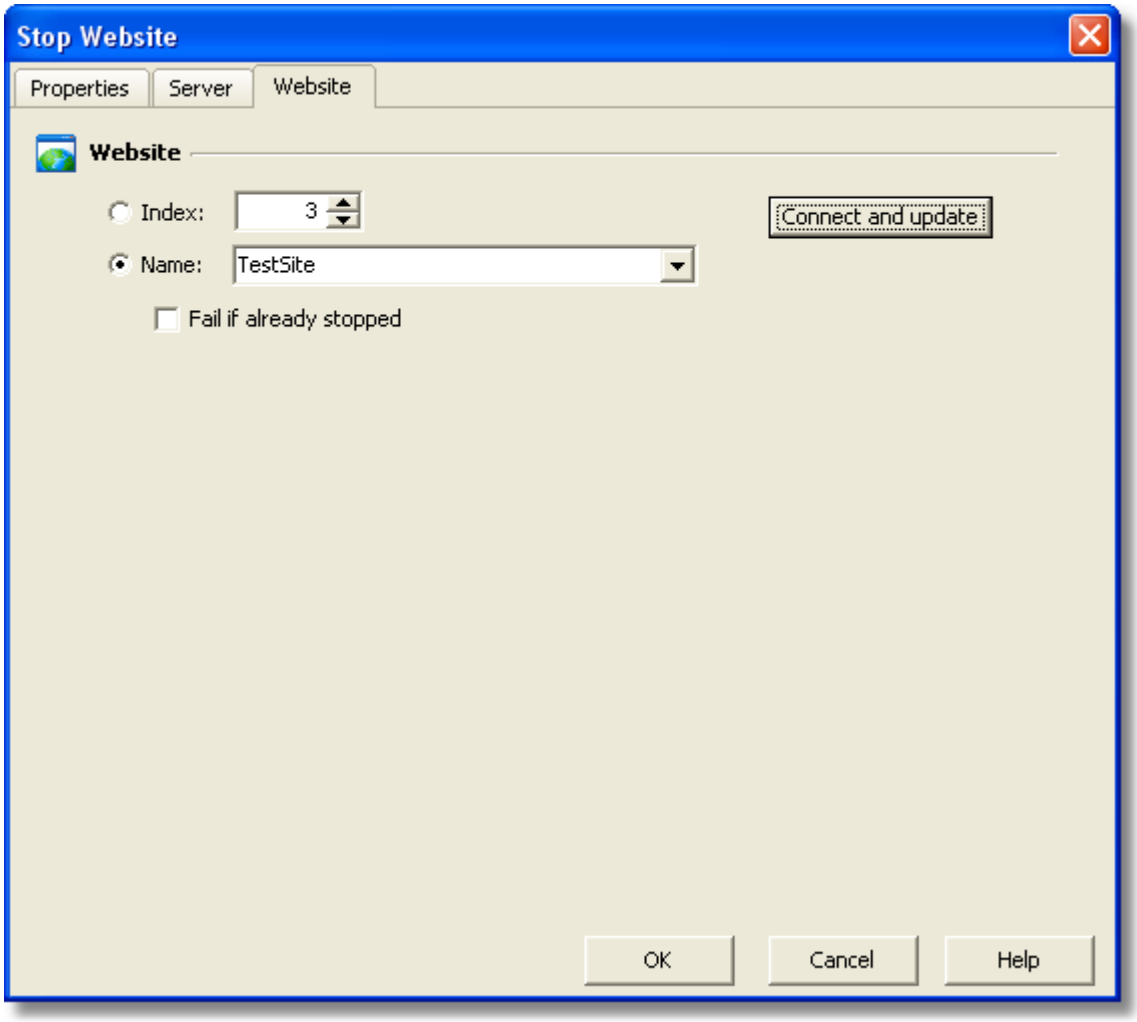
© 2001-2006 VSoft Technologies Pty Ltd



5.27.2.2 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.)

© 2001-2006 VSoft Technologies Pty Ltd



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.

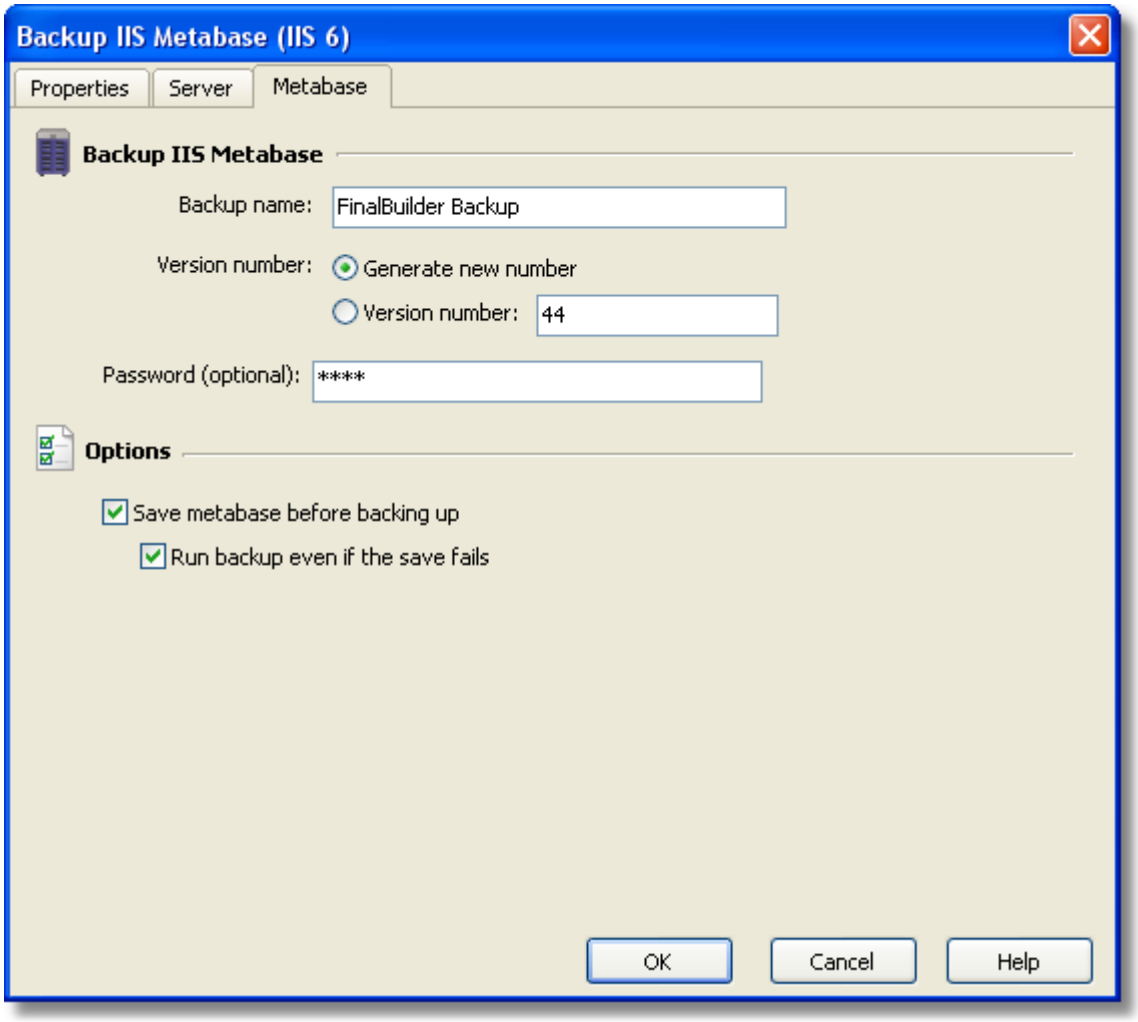
5.27.2.3 Backup IIS Metabase Action

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

© 2001-2006 VSoft Technologies Pty Ltd

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."

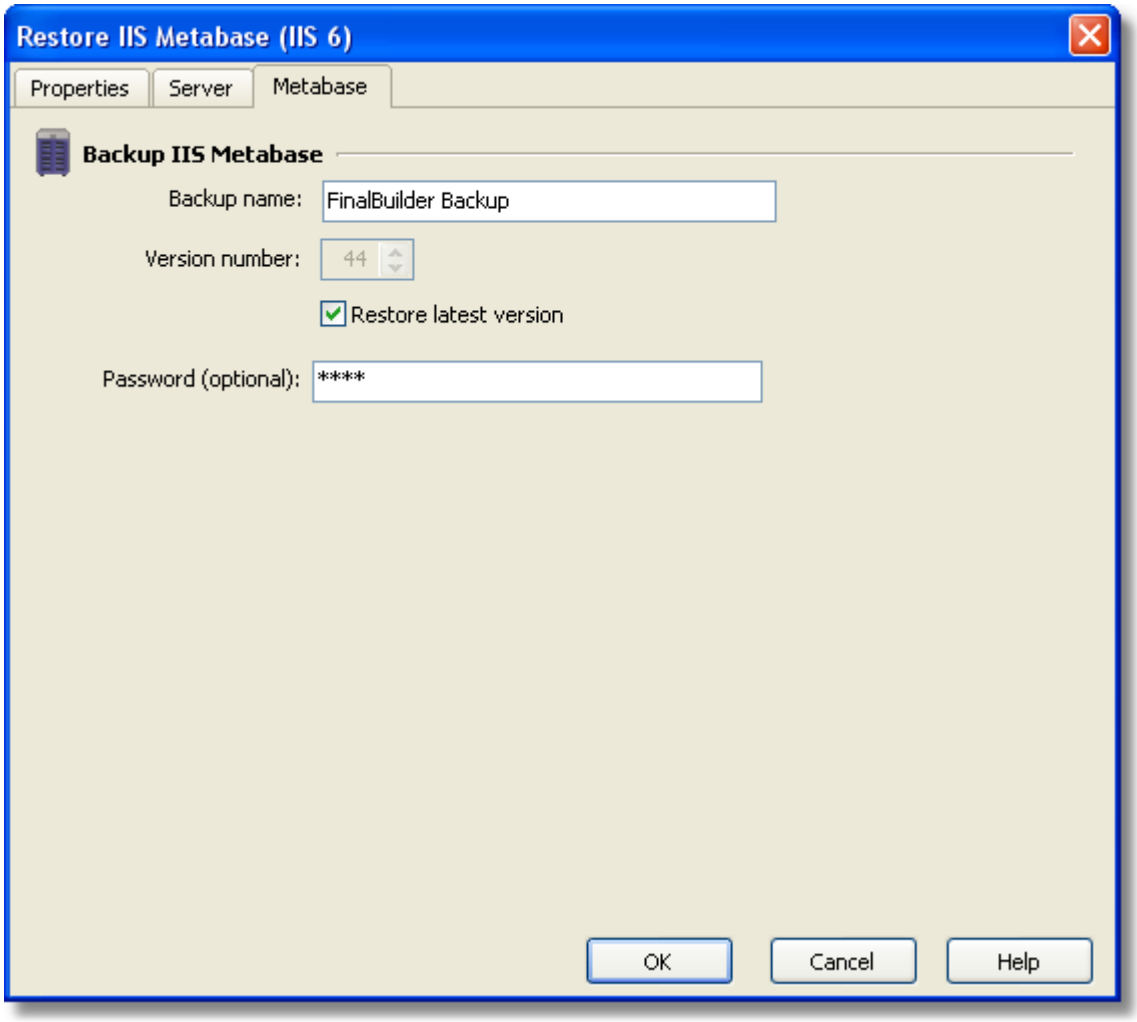
5.27.2.4 Restore IIS Metabase Action

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.

© 2001-2006 VSoft Technologies Pty Ltd



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.

5.27.2.5 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 administrate, and to specify credentials for connecting to the server.

© 2001-2006 VSoft Technologies Pty Ltd

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.)

5.27.2.6 Start / Stop / Pause Website Actions

The Start, Stop & Pause Website actions allow you to take individual IIS websites on- and off-line.

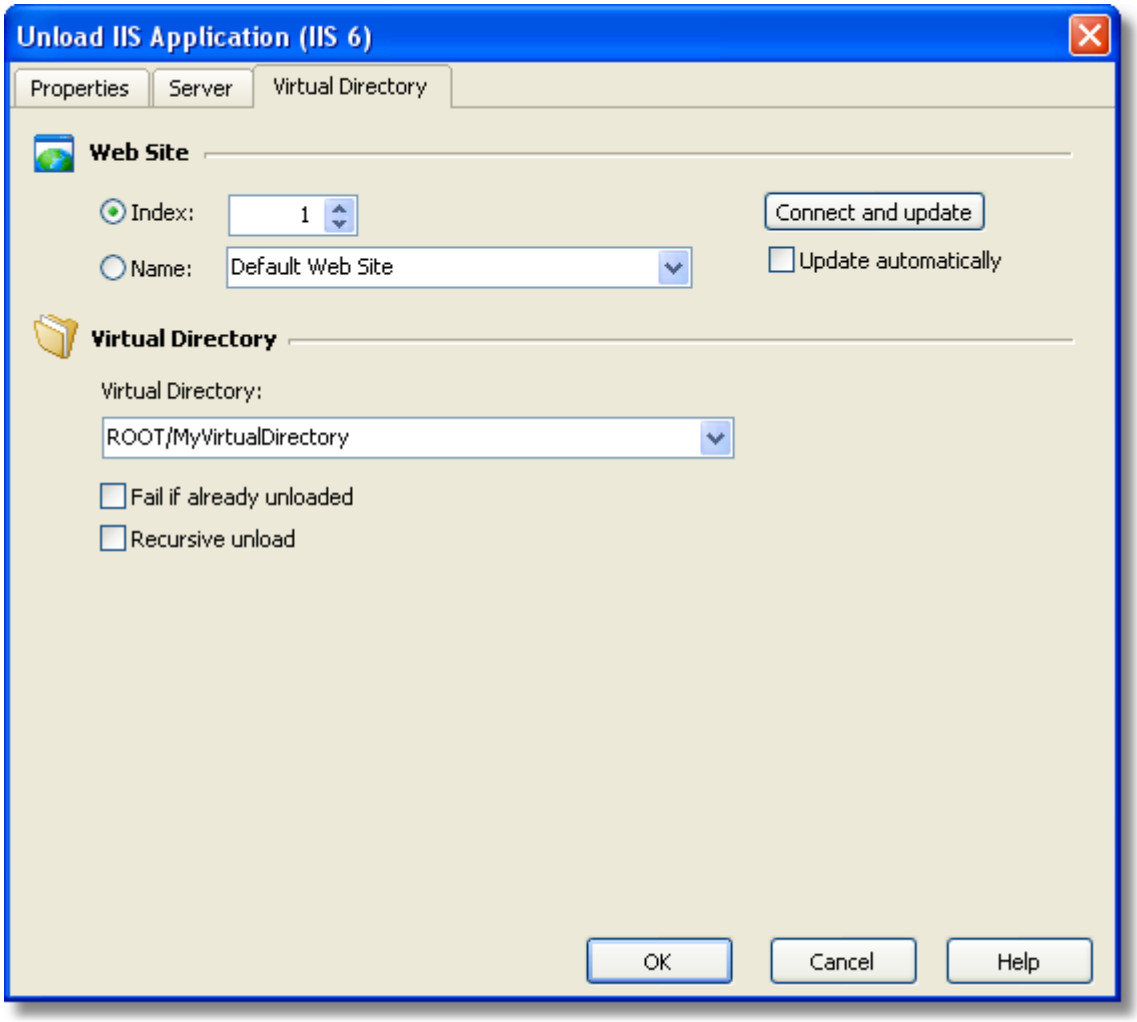
Use the Server Selection tab to choose the server you wish to administrate, and credentials for the connection.
Then use the Website Selection tab to choose the web site you wish to modify.

The actions can be set not to fail if the selected web site is already in the desired state (running/stopped.)

5.27.2.7 Unload IIS Application Action

The Unload IIS Application actions allows you unload a given IIS web application from memory.

© 2001-2006 VSoft Technologies Pty Ltd



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

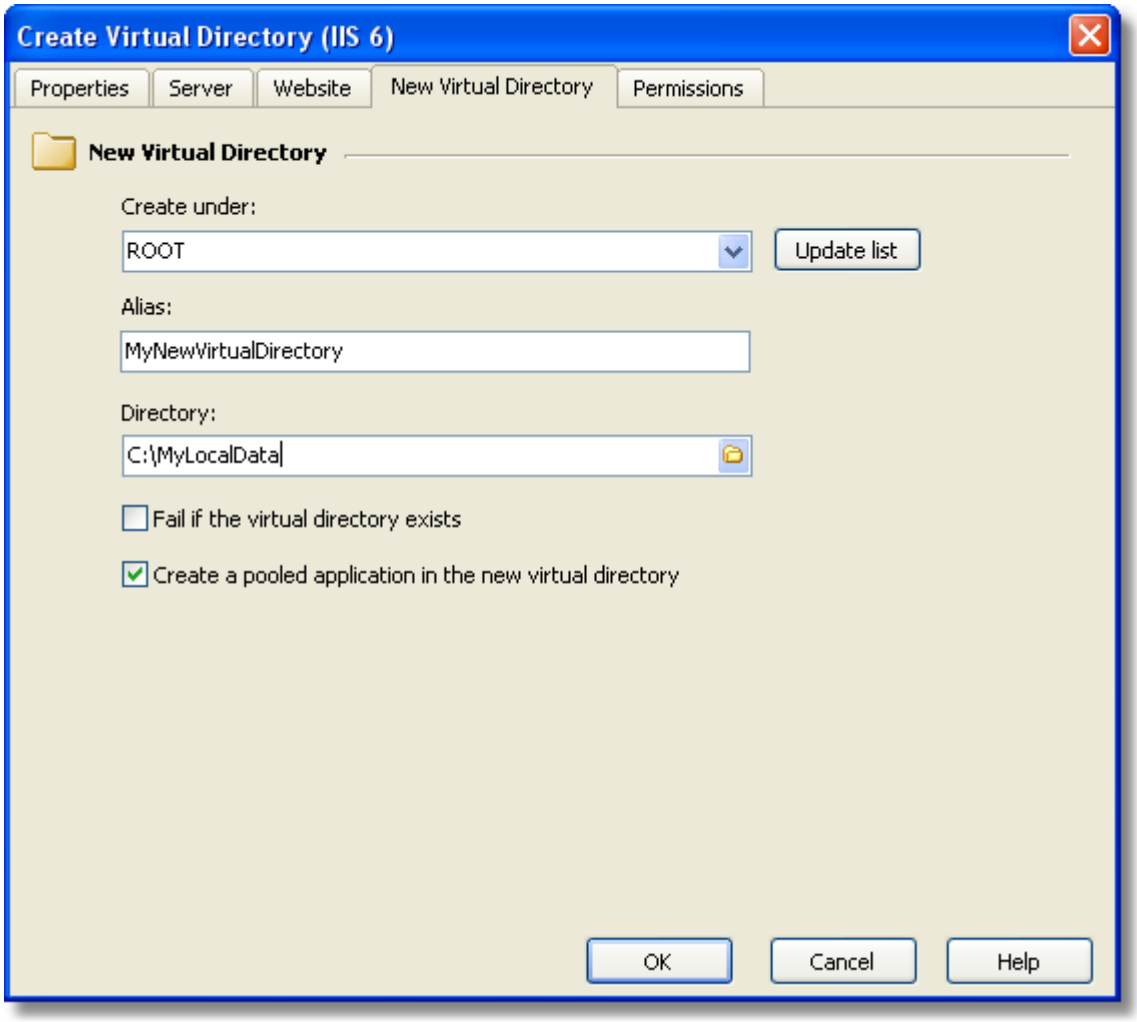
Choose the virtual directory which hosts the application that you wish to unload. You can set the action not to fail if the application is already unloaded.

Select "Recursive Unload" to recursively unload applications in all child directories.

5.27.2.8 Create Virtual Directory Action

The Create Virtual Directory action allows you to add a new virtual directory to an existing IIS web site.

© 2001-2006 VSoft Technologies Pty Ltd



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

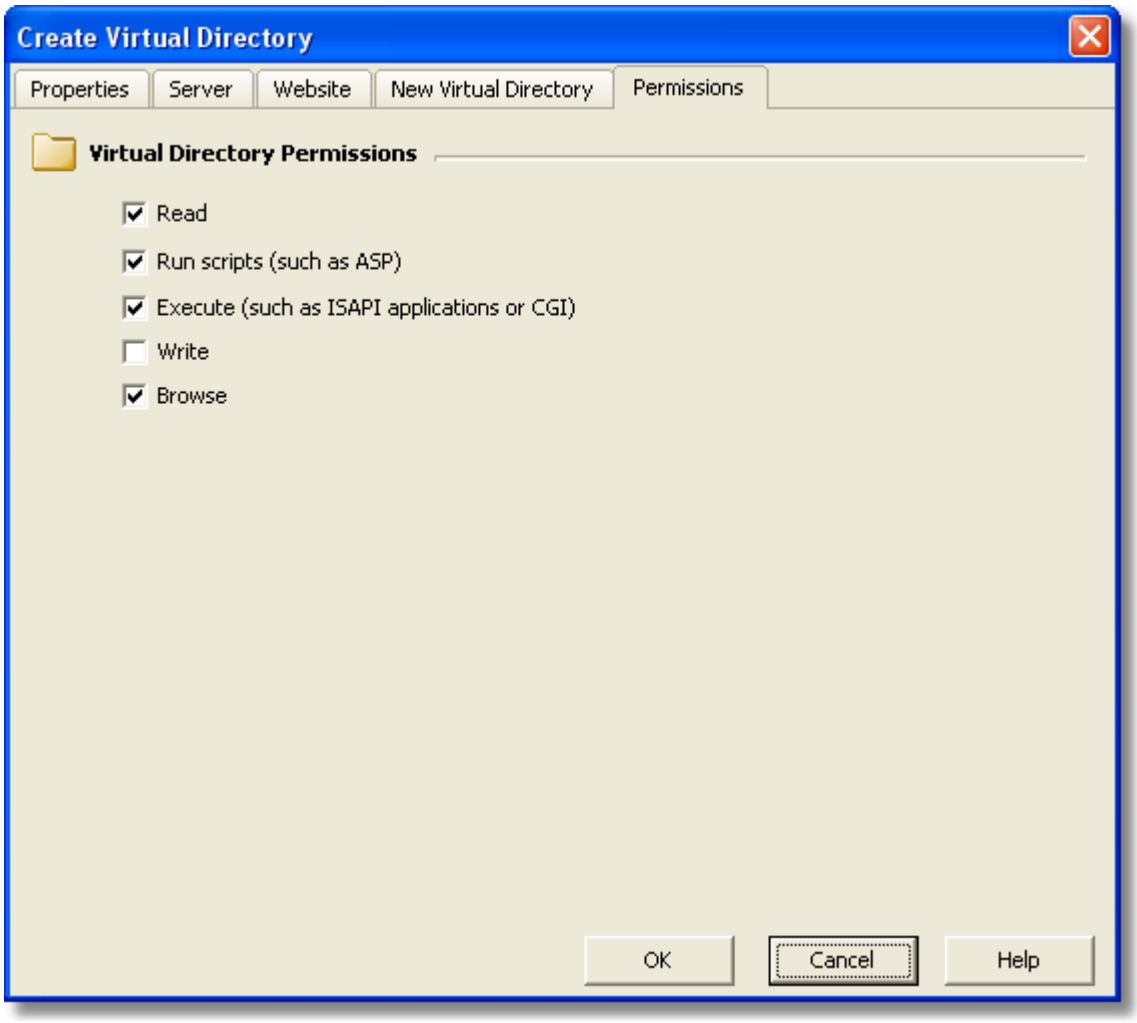
Specify the local directory for the contents of the new virtual directory.

Create a pooled application...

© 2001-2006 VSoft Technologies Pty Ltd

Check this box if you would like to create a pooled web application running in the new virtual directory.

Permissions Tab

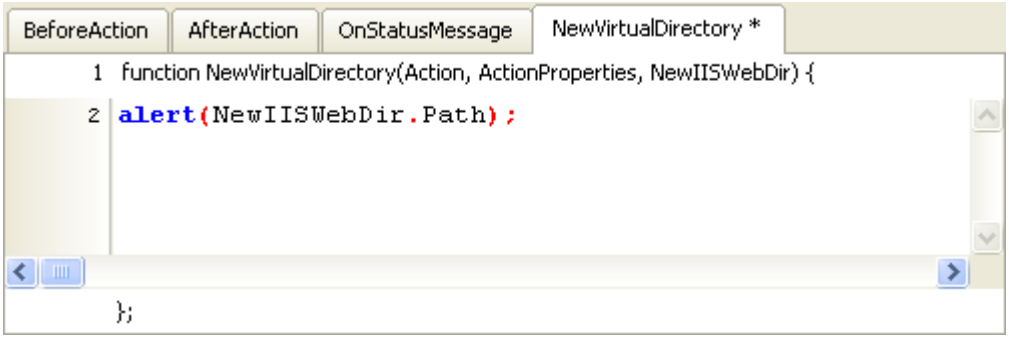


Set the permissions you want IIS users to have on the new Virtual Directory.

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.

© 2001-2006 VSoft Technologies Pty Ltd



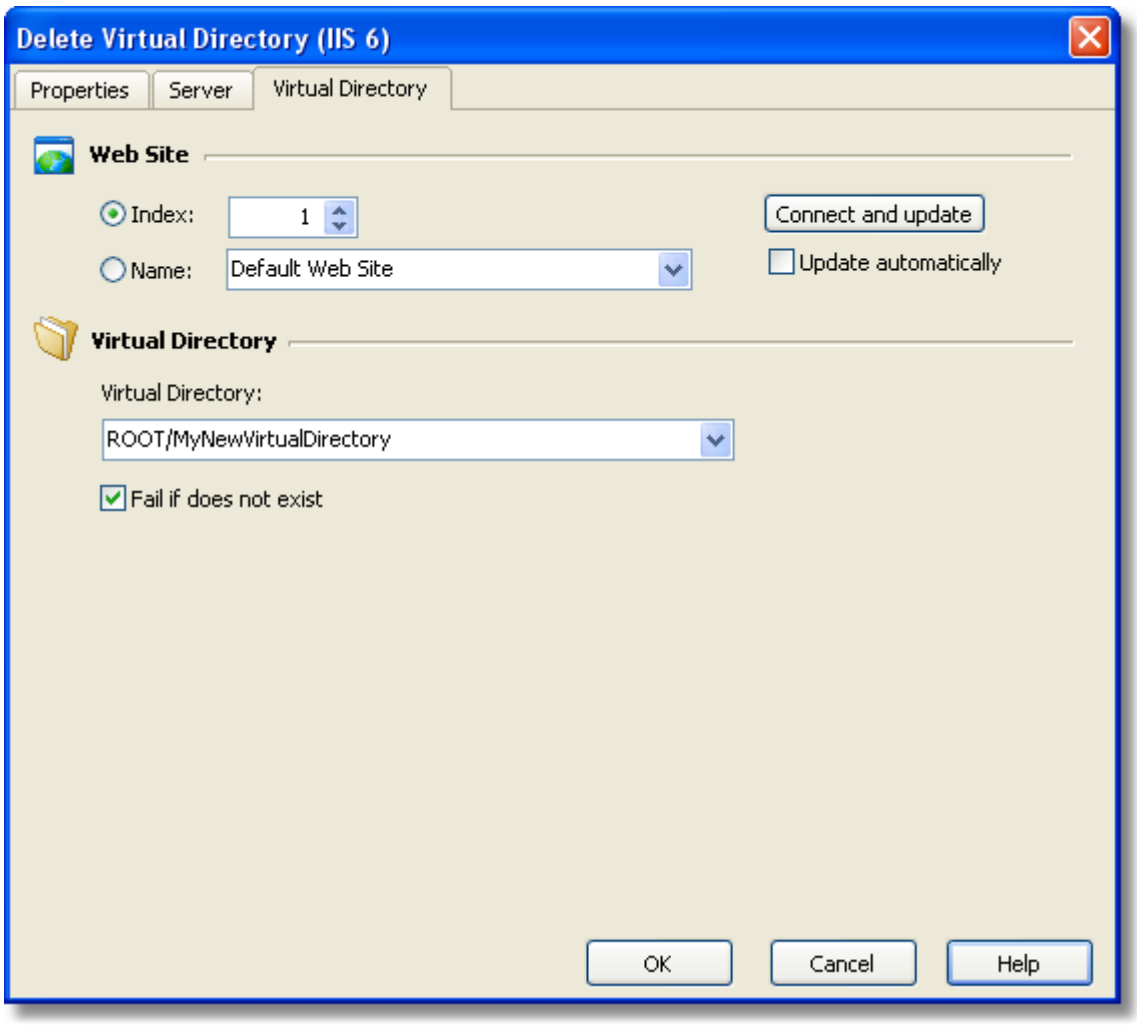
The **NewIISWebDir parameter** is an WMI settings object of type IIsWebVirtualDirSetting (see the MSDN Documentation.)

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.)

© 2001-2006 VSoft Technologies Pty Ltd

5.27.2.9 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.

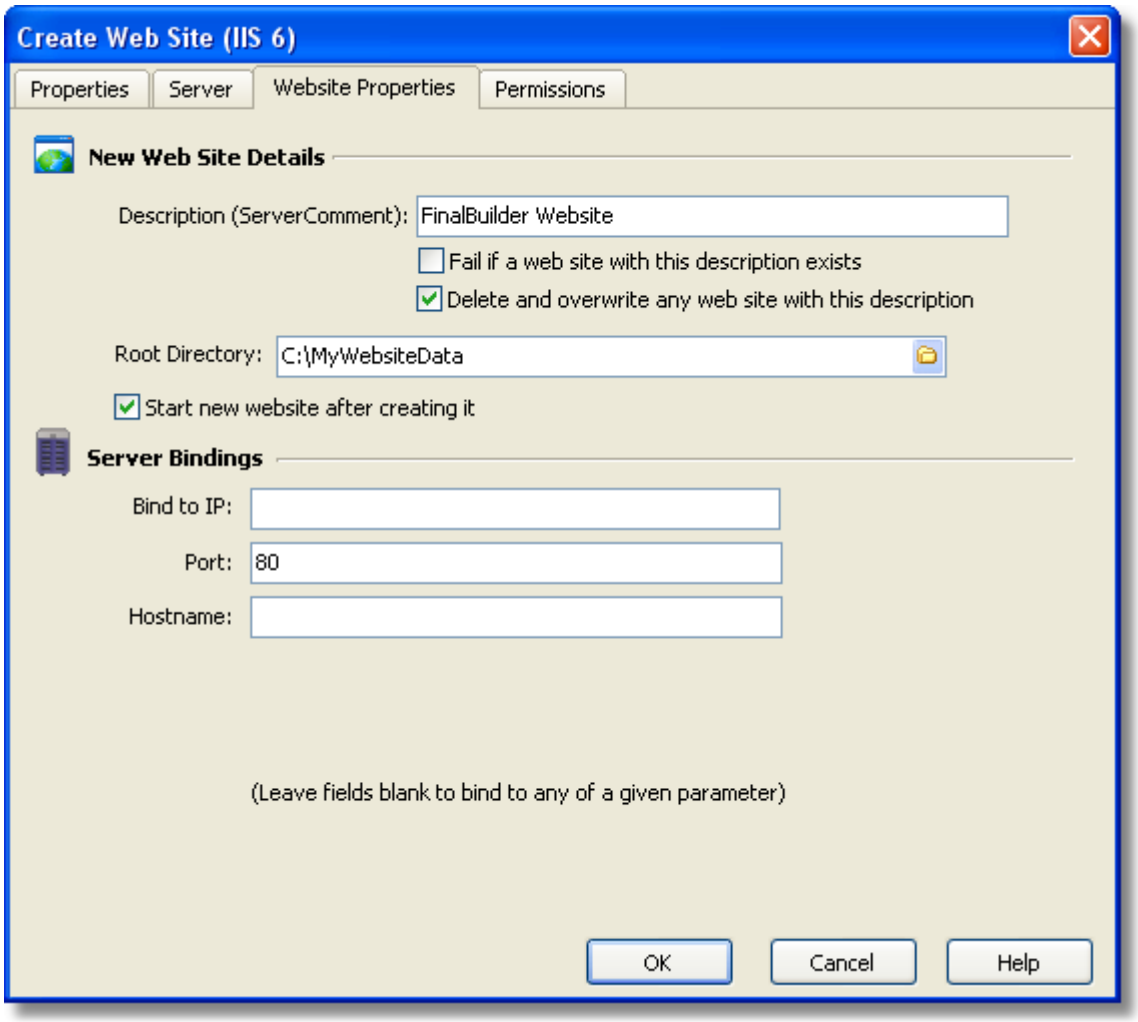
Fail if does not exist

Check this box if you want the action to fail should the specified virtual directory not exist.

© 2001-2006 VSoft Technologies Pty Ltd

5.27.2.10 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.



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.

© 2001-2006 VSoft Technologies Pty Ltd

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.

5.27.2.11 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.

© 2001-2006 VSoft Technologies Pty Ltd

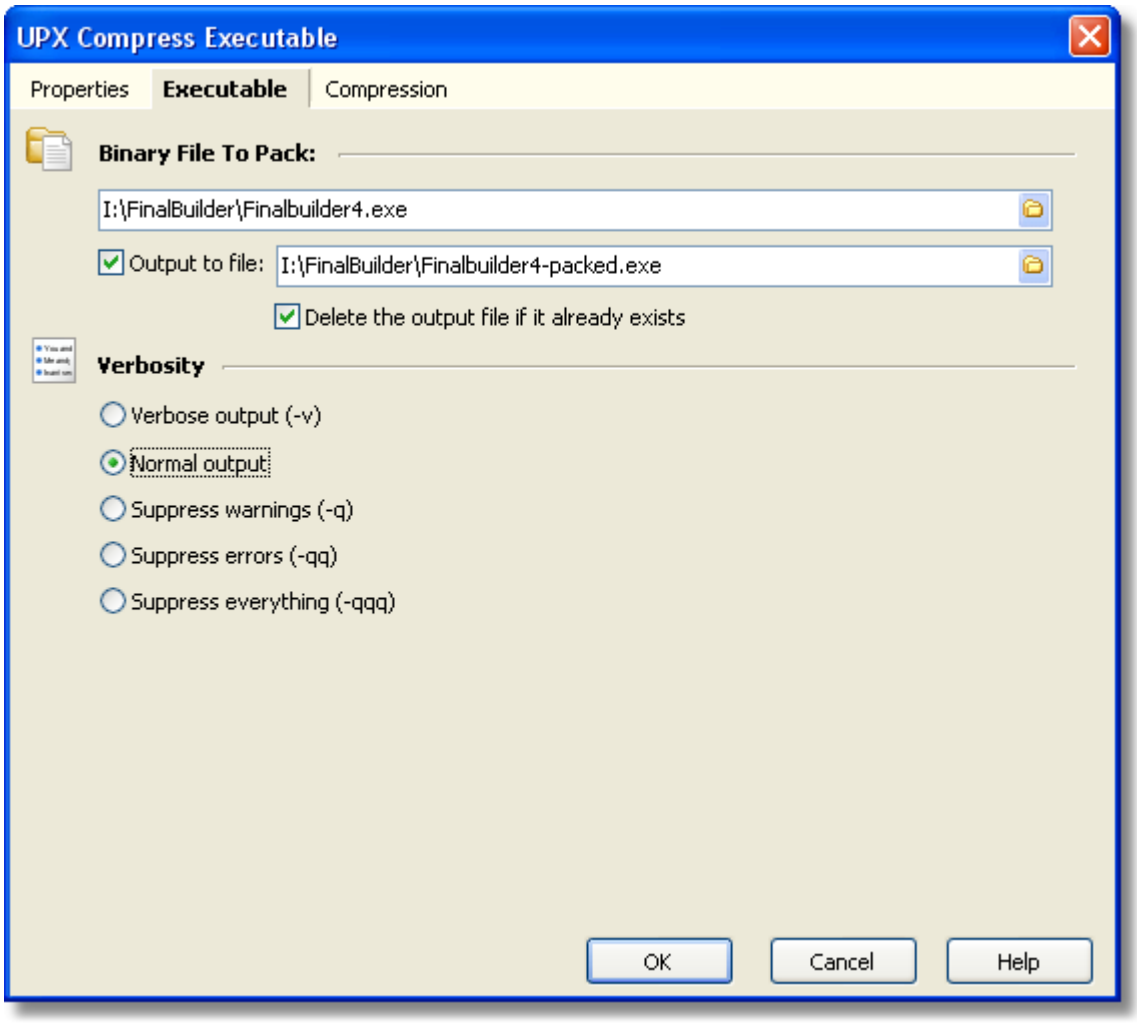
5.28 Executable Packers

5.28.1 UPX Pack Executable

UPX (The Ultimate Packer for Executables) 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

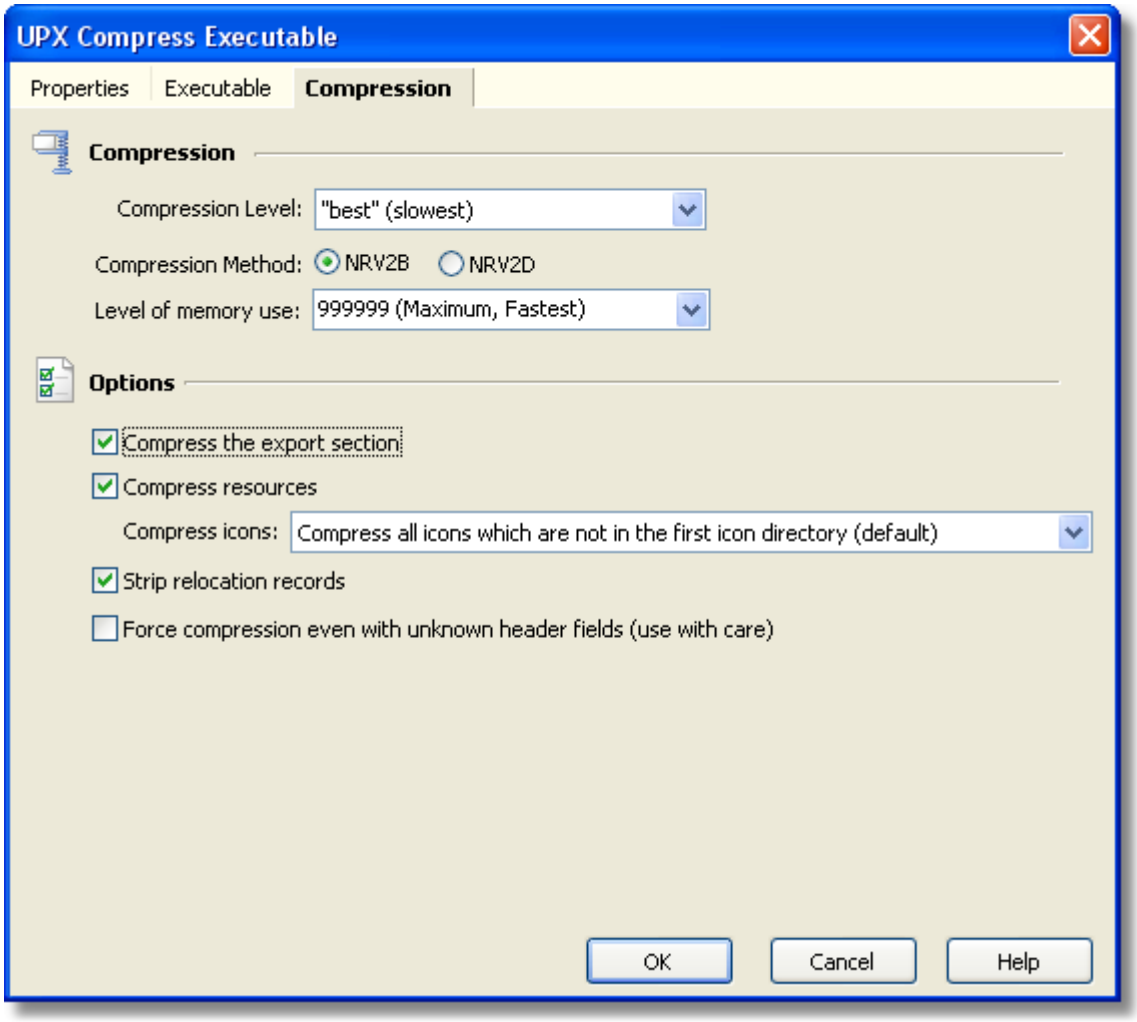
UPX can pack executable files, Win32 DLLs and screensavers.

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

© 2001-2006 VSoft Technologies Pty Ltd

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

© 2001-2006 VSoft Technologies Pty Ltd

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)

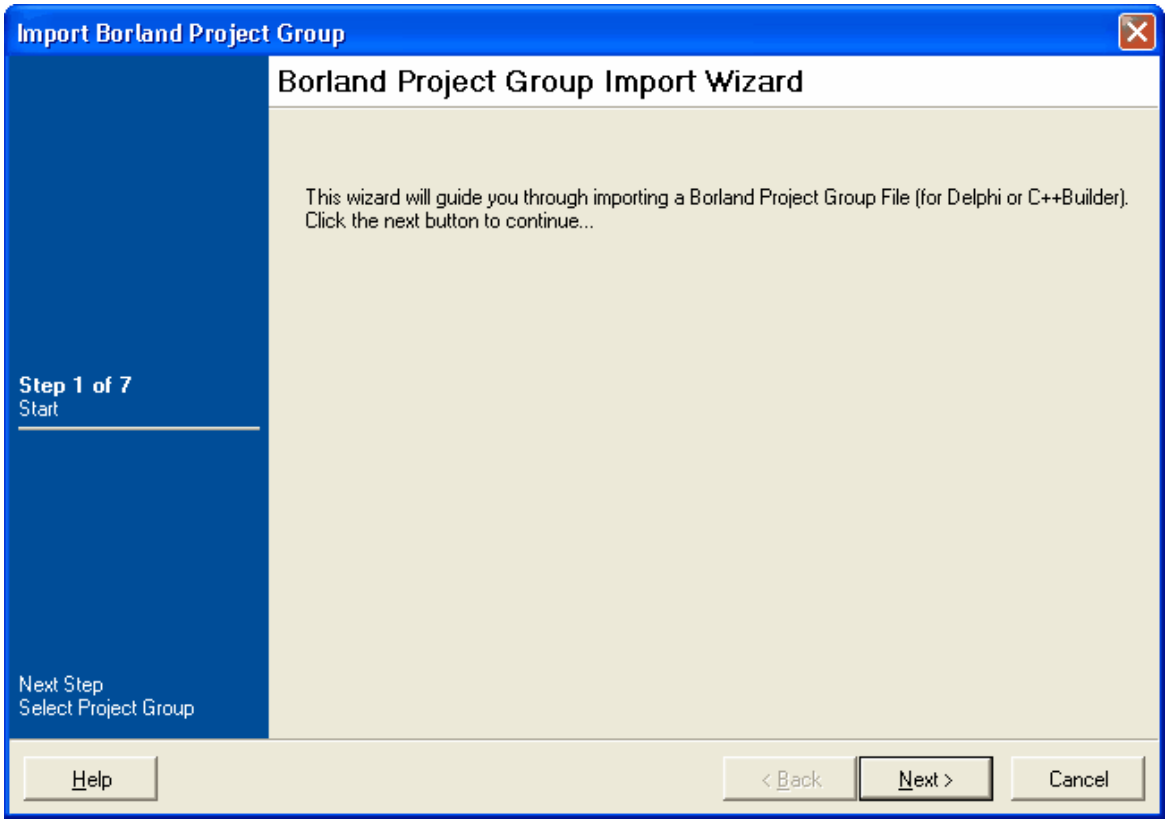
If this option is checked, it forces UPX to compress the executable even if it finds unknown data in the executable header fields. This may render your application unusable. See the UPX documentation for more details.

© 2001-2006 VSoft Technologies Pty Ltd

6 Wizards

6.1 Import Borland Project Wizard

The Import Borland Project Group Wizard imports Delphi project Group (.bpg) files.

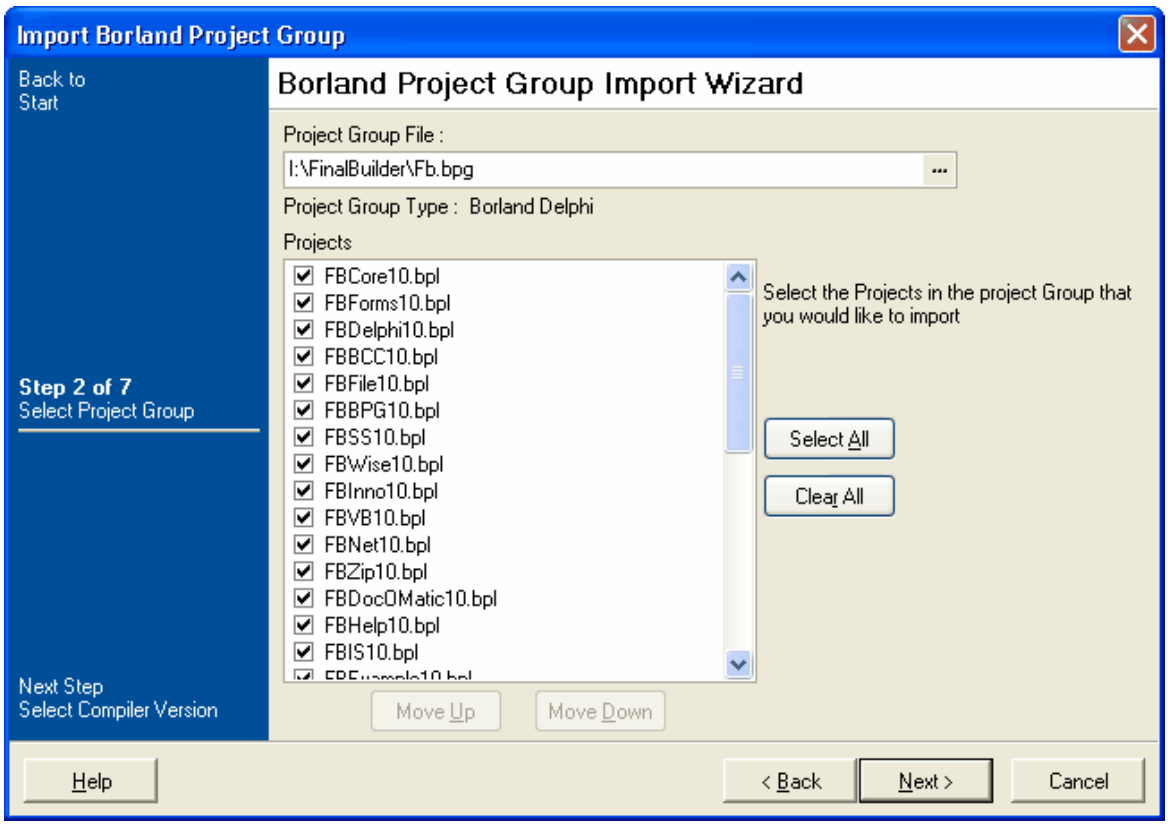


Next - Select Project Group Page

© 2001-2006 VSoft Technologies Pty Ltd

6.1.1 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.

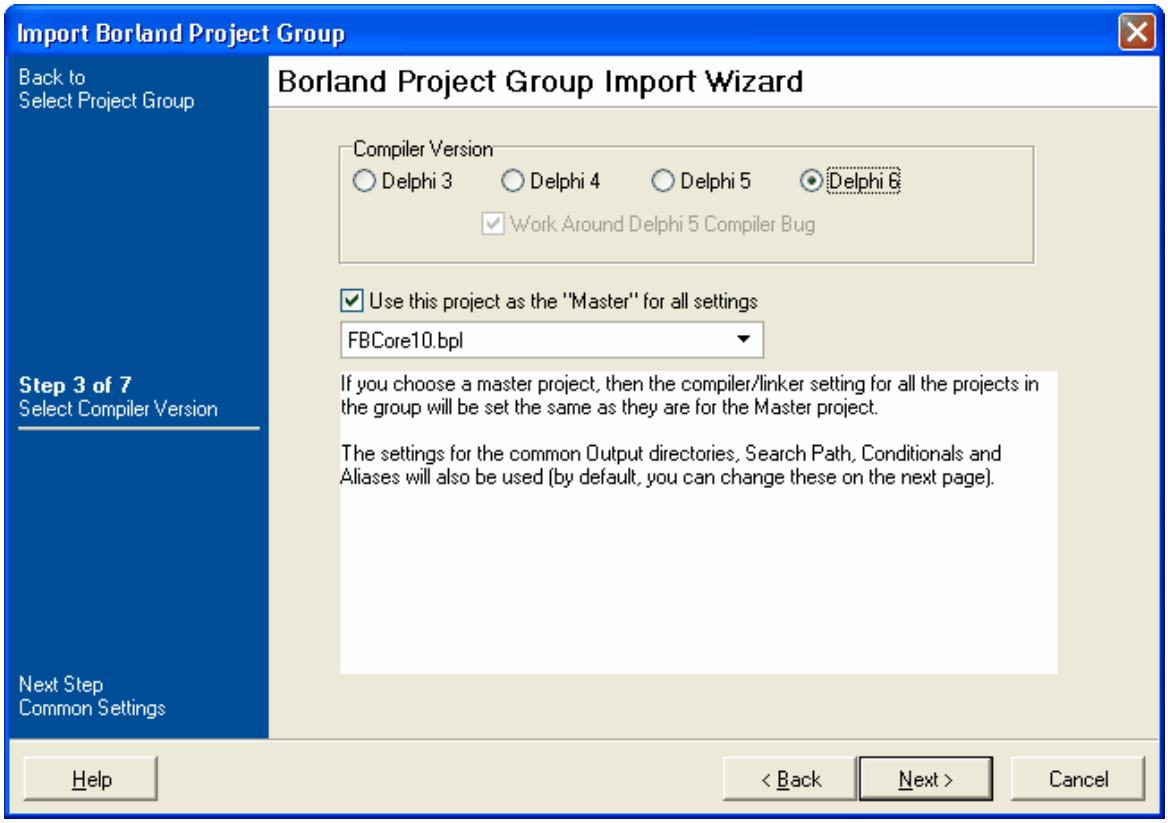


Next - Select Compiler Version Page

© 2001-2006 VSoft Technologies Pty Ltd

6.1.2 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.

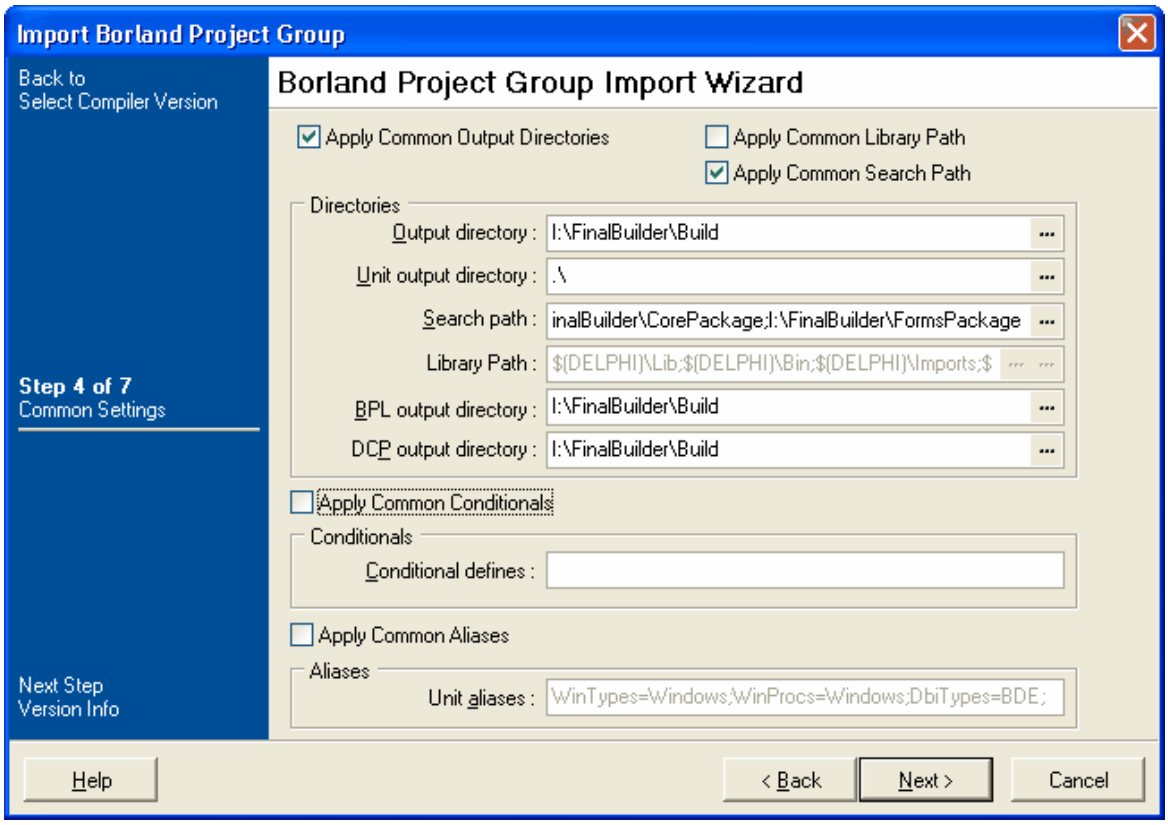


Next - Common Settings Page

© 2001-2006 VSoft Technologies Pty Ltd

6.1.3 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.

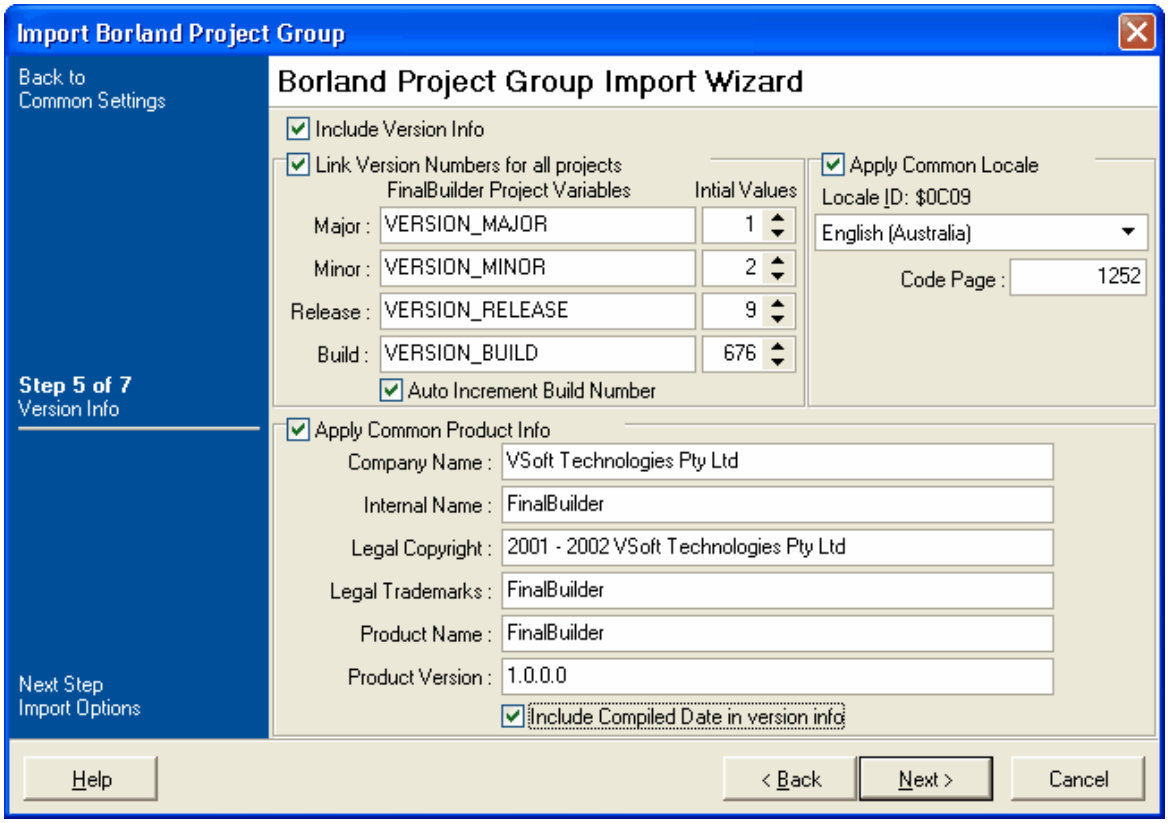


Next - Version Info Page

© 2001-2006 VSoft Technologies Pty Ltd

6.1.4 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.



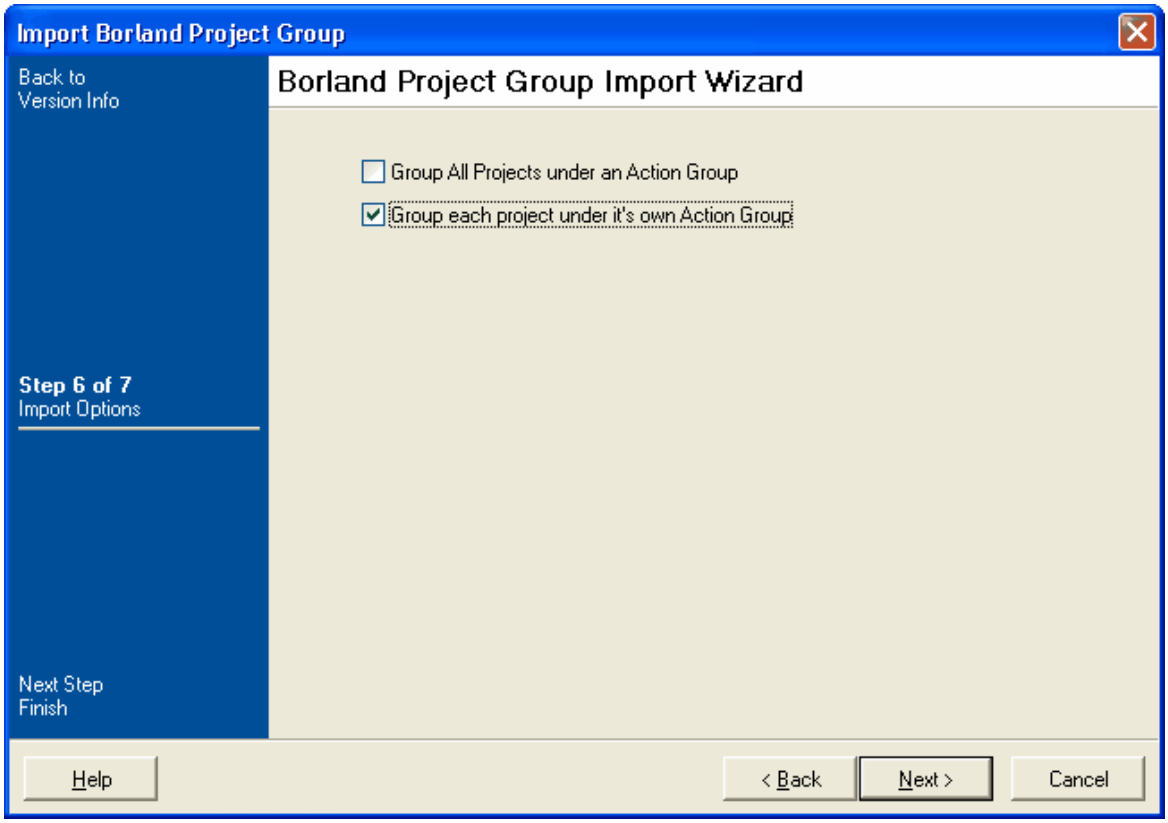
Next - Import Options Page

© 2001-2006 VSoft Technologies Pty Ltd

6.1.5 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 ad further action which are specific to each Delphi project.

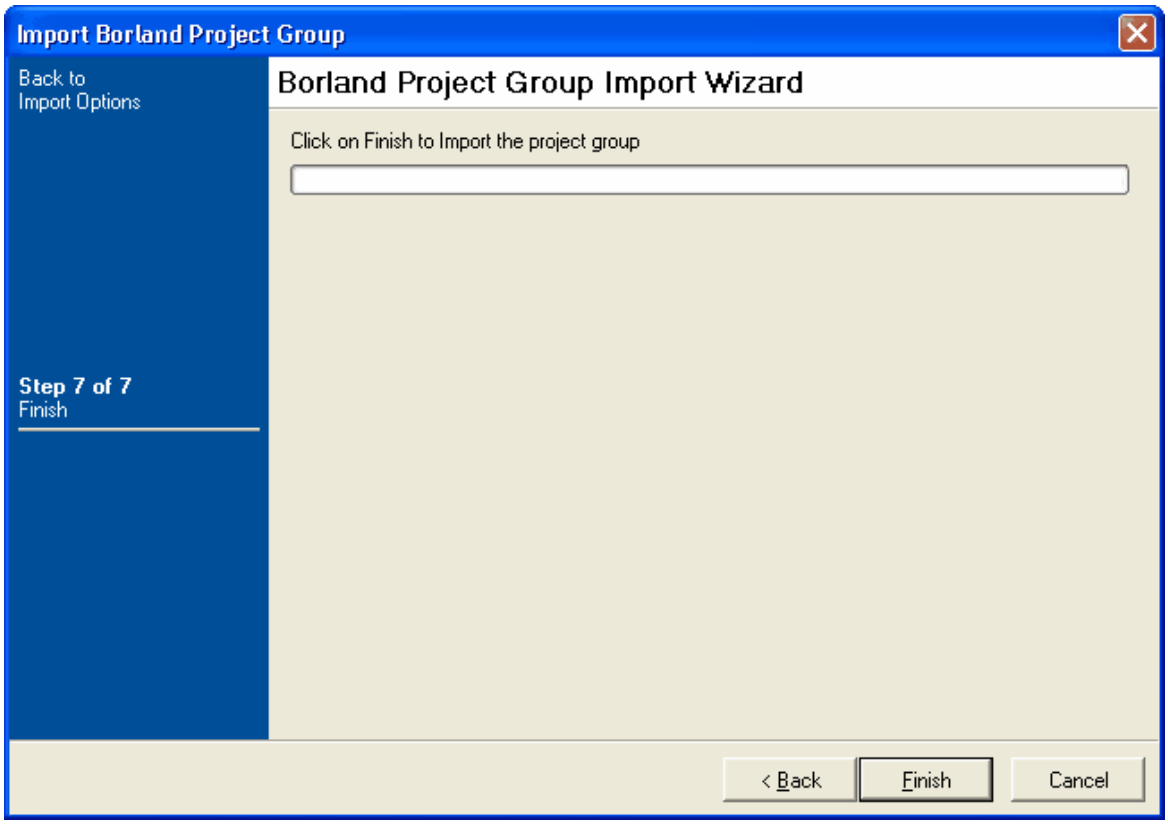


Next - Finish Page

© 2001-2006 VSoft Technologies Pty Ltd

6.1.6 Finish Page

Click on the Finish button to import the project group.



© 2001-2006 VSoft Technologies Pty Ltd

6.2 Import VB6 Project Group Wizard

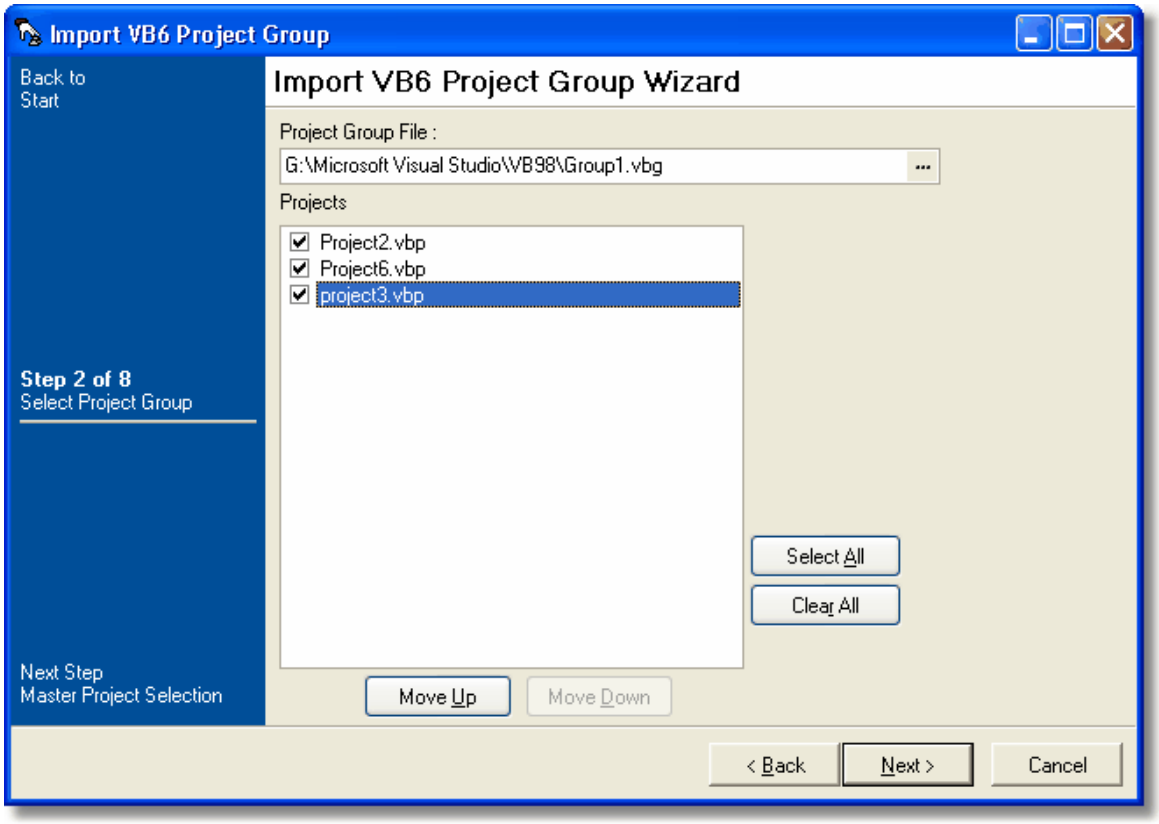
This wizard will guide you through importing a VB6 project group.

Move on to the first page.

© 2001-2006 VSoft Technologies Pty Ltd

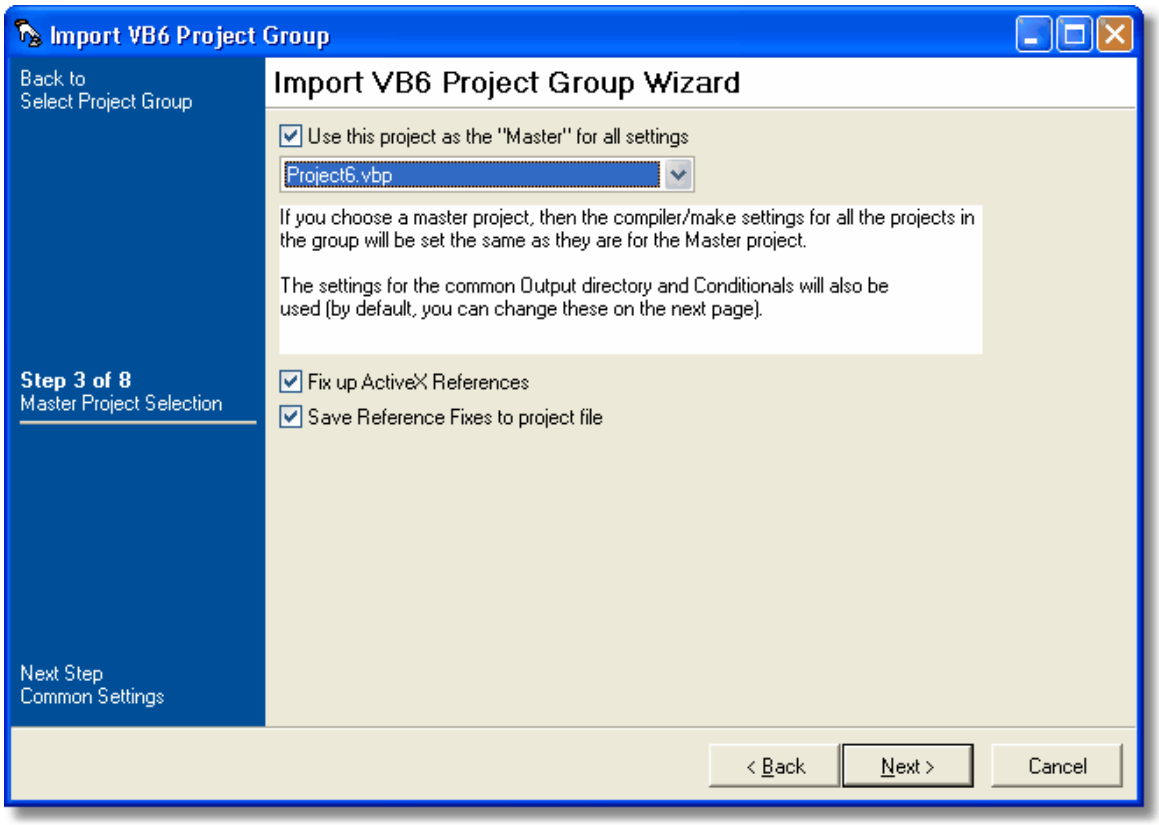
6.2.1 Select Project Group

This page allows you to select the Project Group File (.vbg) and select the projects with the project group that you wish to import.



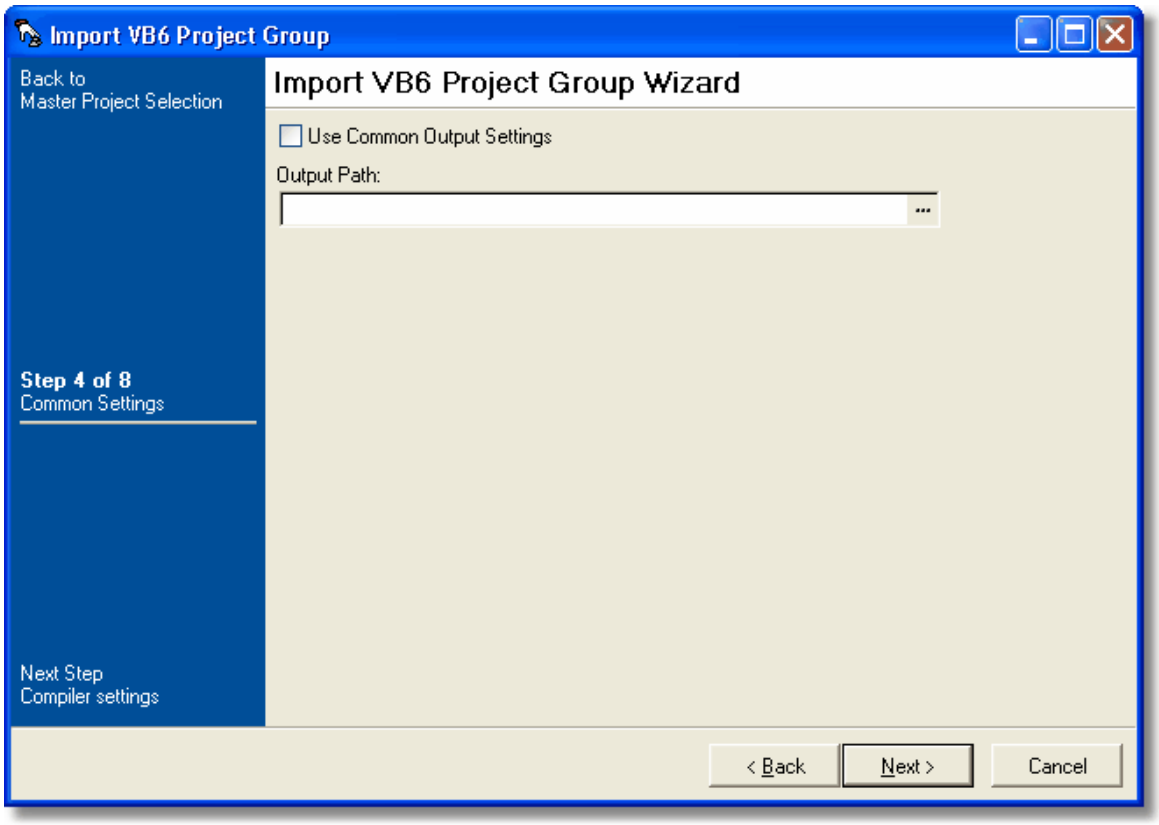
© 2001-2006 VSoft Technologies Pty Ltd

6.2.2 Master Project Selection



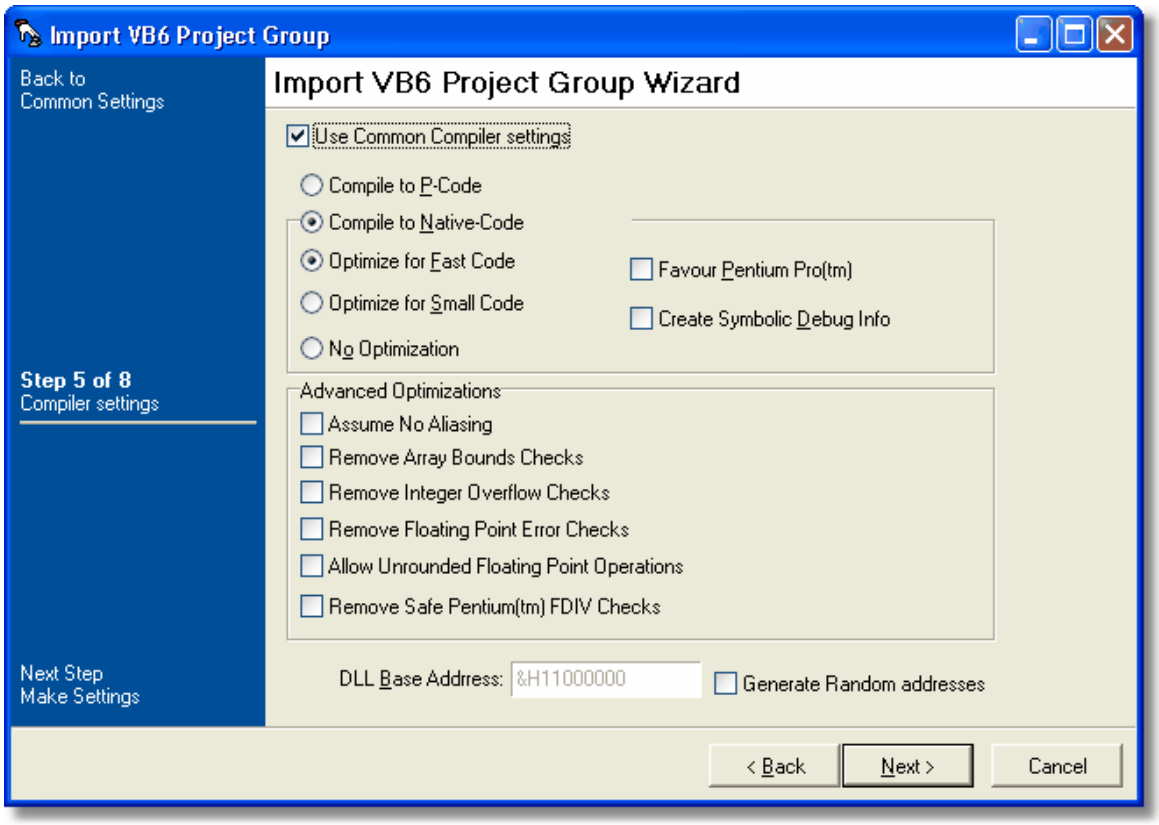
© 2001-2006 VSoft Technologies Pty Ltd

6.2.3 Common Output Path Setting



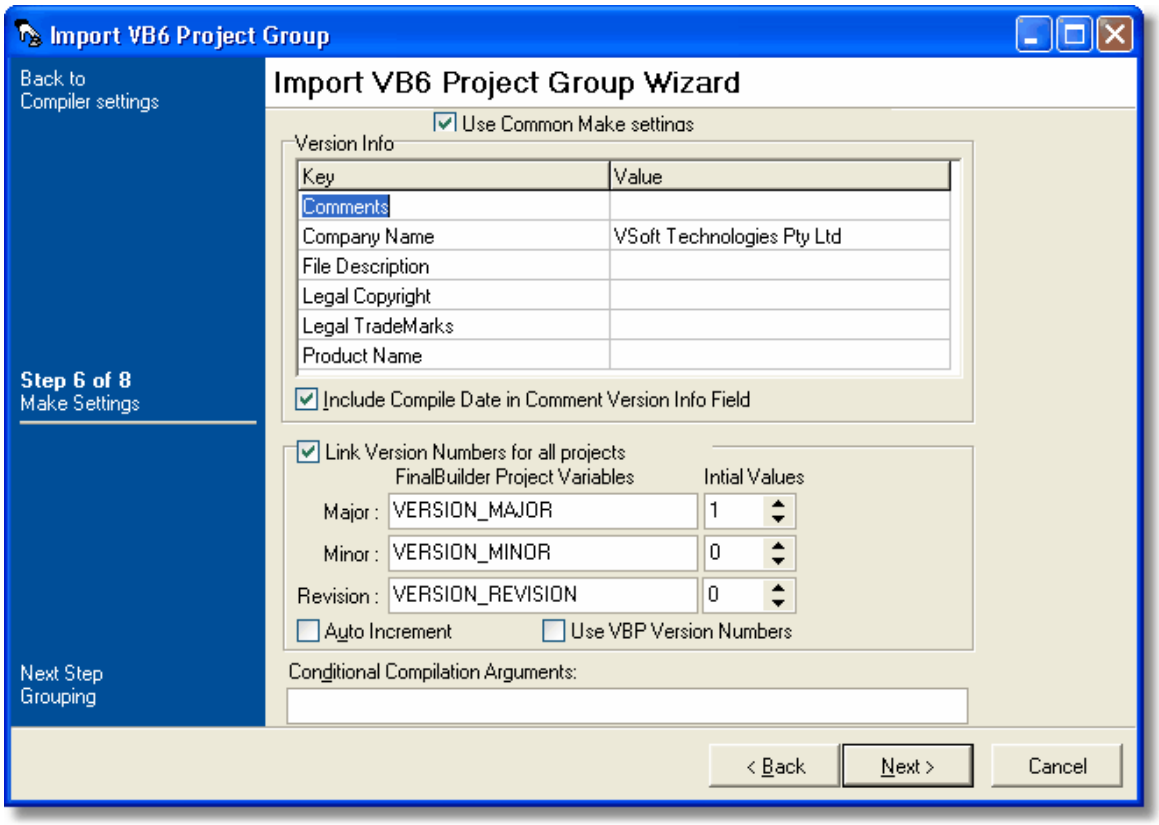
© 2001-2006 VSoft Technologies Pty Ltd

6.2.4 Compiler Settings



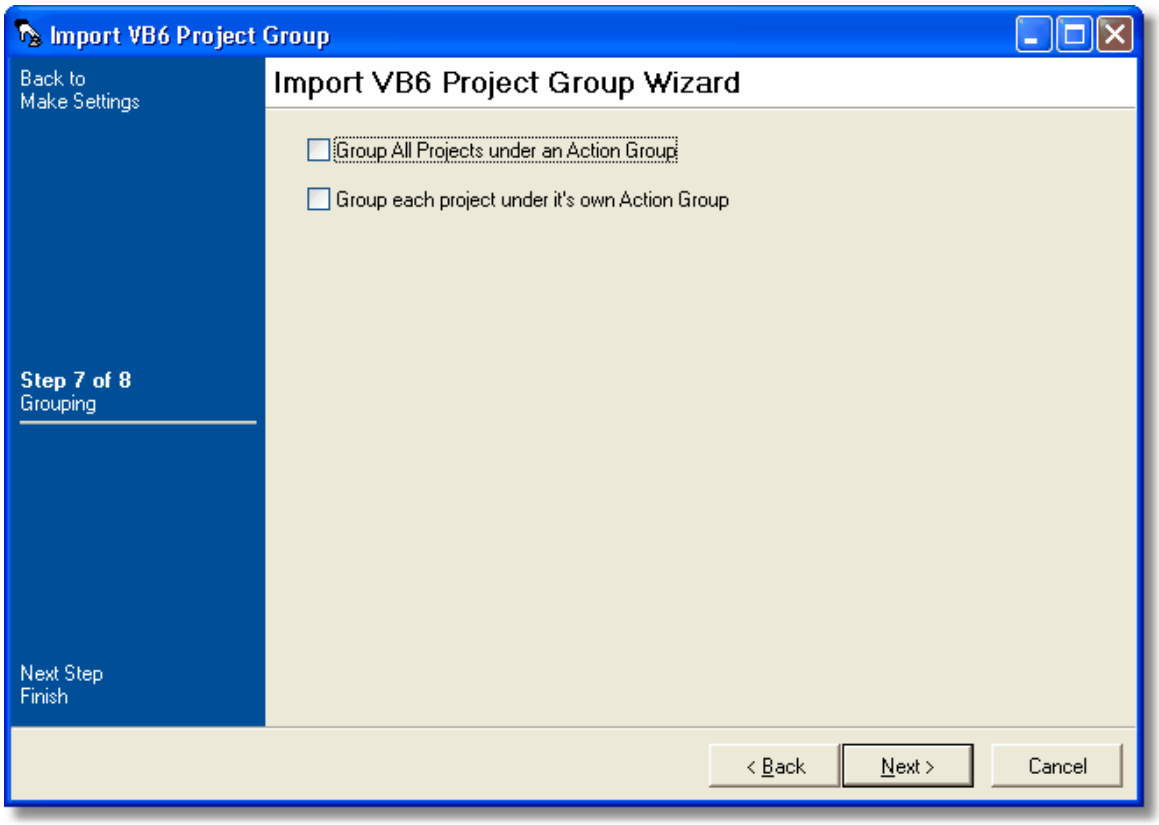
© 2001-2006 VSoft Technologies Pty Ltd

6.2.5 Make Settings



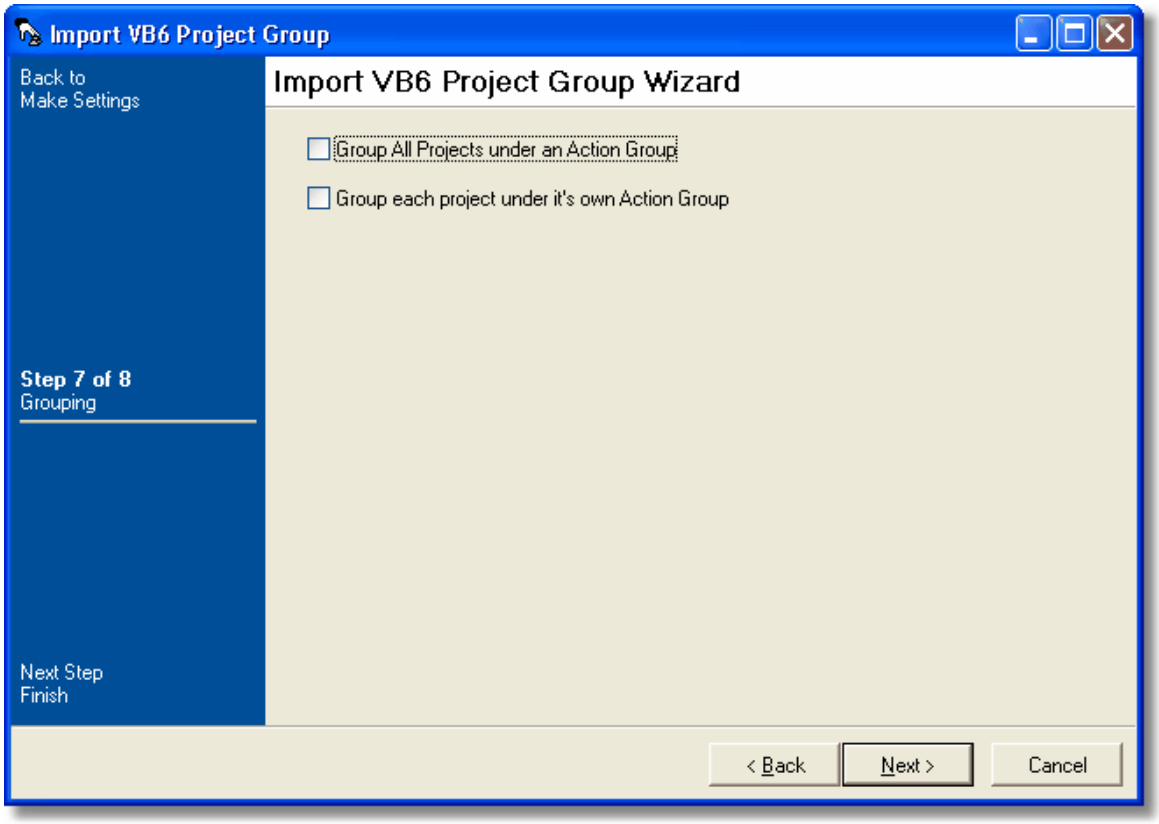
© 2001-2006 VSoft Technologies Pty Ltd

6.2.6 Grouping



© 2001-2006 VSoft Technologies Pty Ltd

6.2.7 Finish



© 2001-2006 VSoft Technologies Pty Ltd

7 Automating FinalBuilder

There are three ways to automate FinalBuilder:

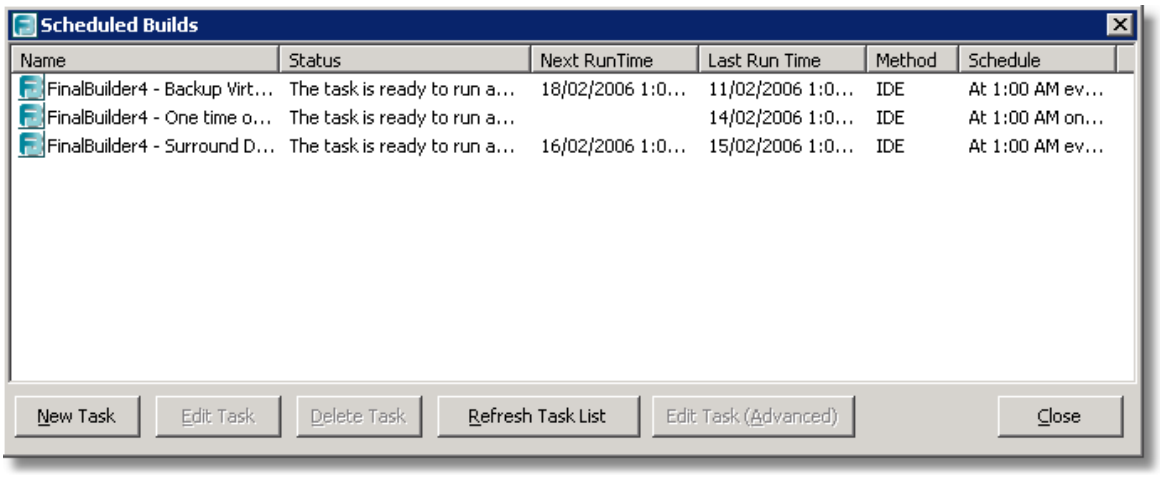
- Executing FinalBuilder (the IDE) from the command line
- Using the built in scheduler
- Executing FBCMD.exe on the command line

© 2001-2006 VSoft Technologies Pty Ltd

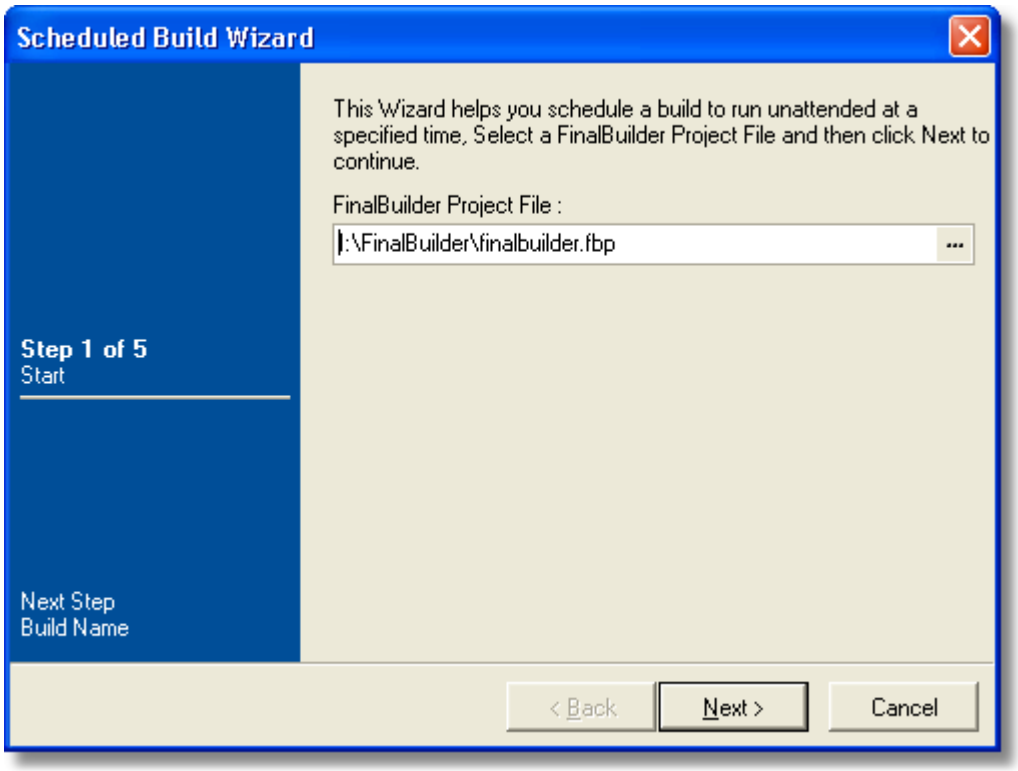
7.1 FinalBuilder IDE

7.1.1 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

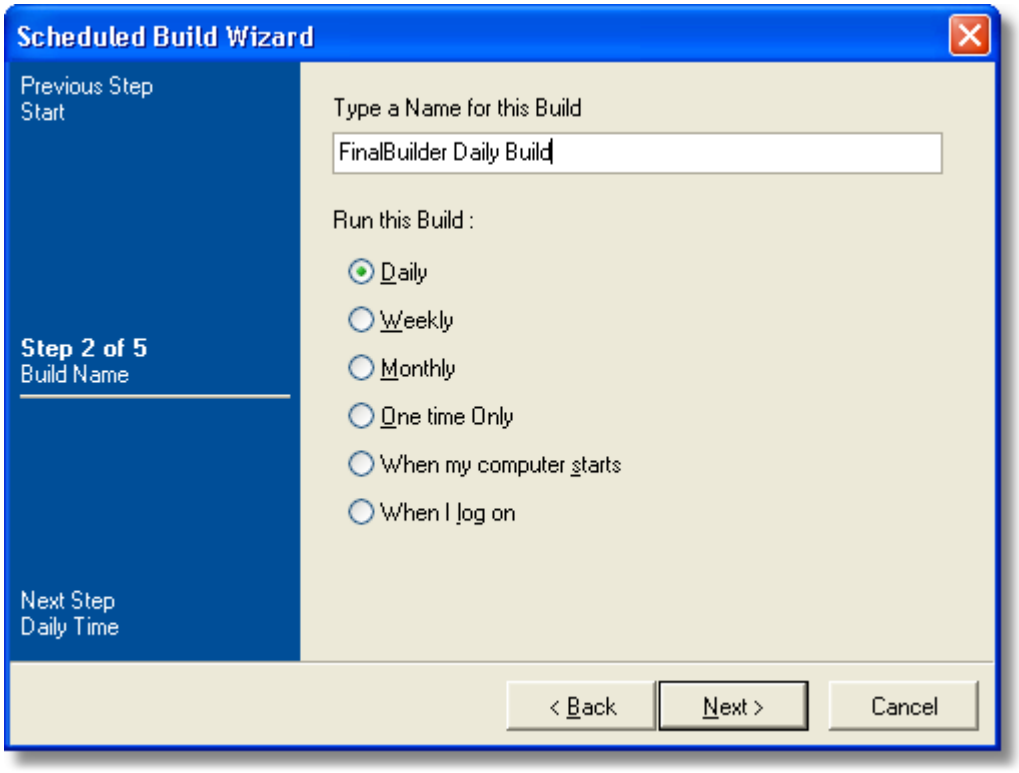


To Create a new new scheduled build, click on the New Task button. This will display the wizard that will step you through scheduling a build.

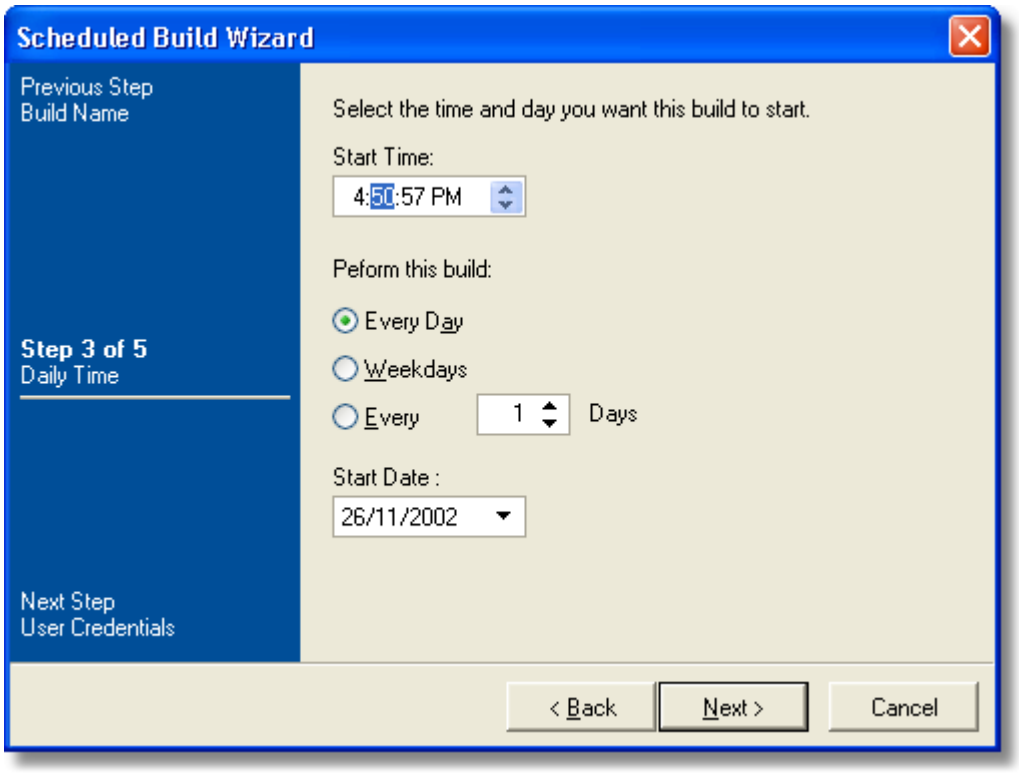


Select a FinalBuilder Project File and then click Next to continue..

© 2001-2006 VSoft Technologies Pty Ltd

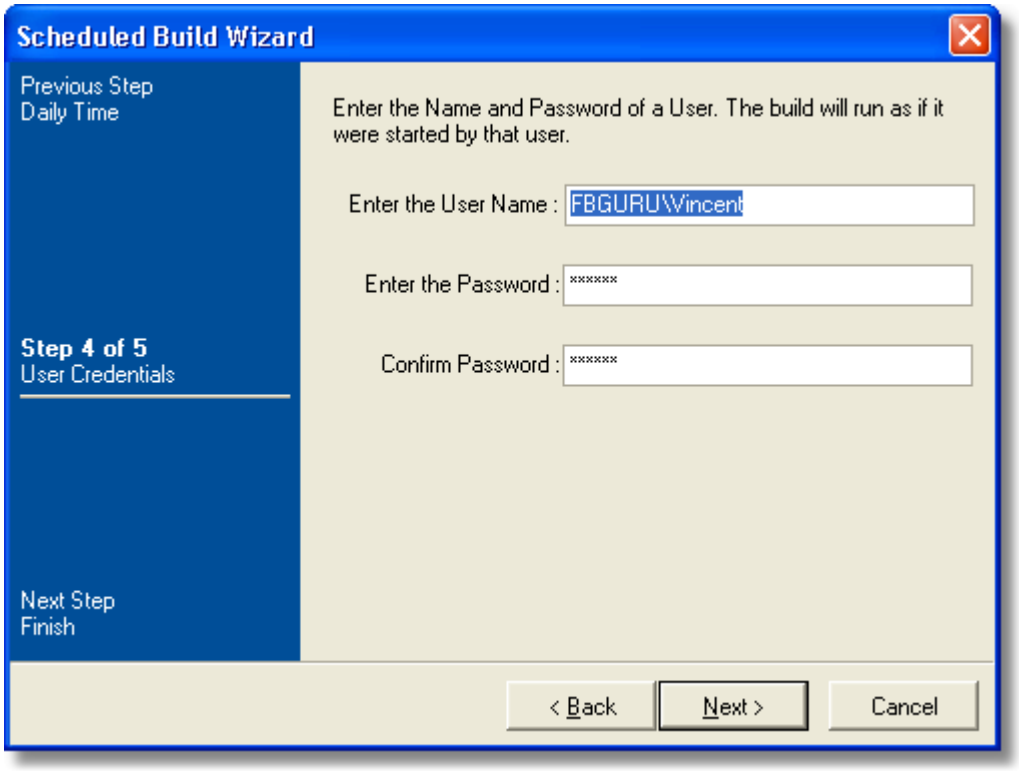


Enter a Title for the scheduled build and then specify the frequency of the build, then click on next to continue.

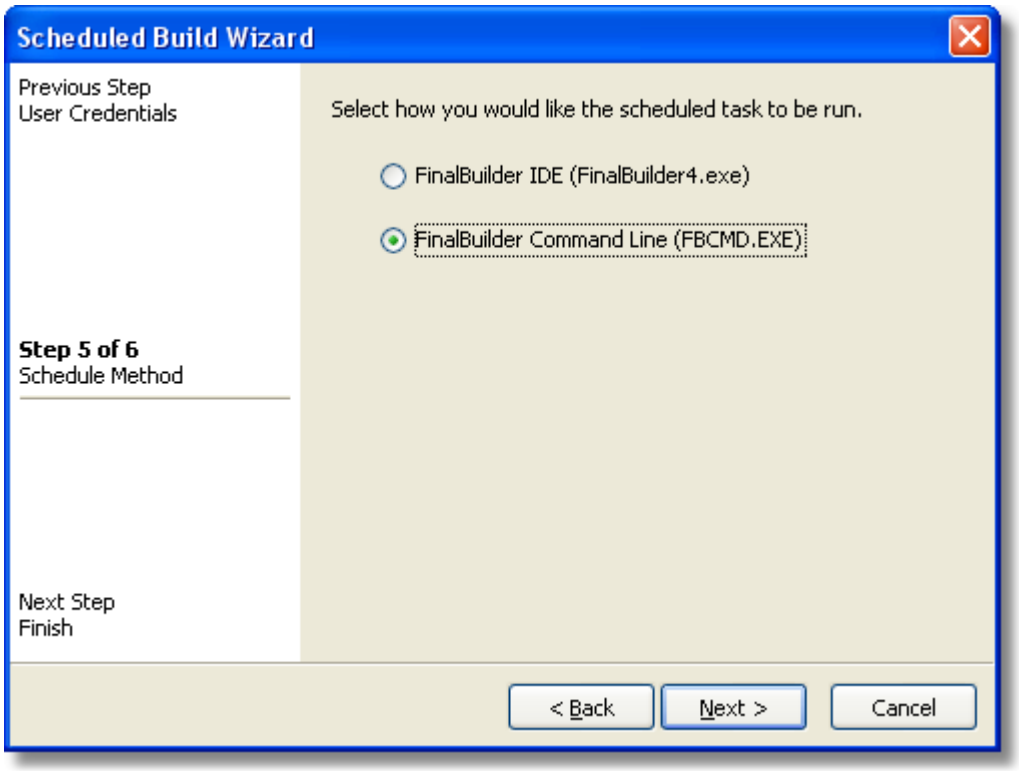


Specify the Time and starting date for the schedule, then click next to continue.

© 2001-2006 VSoft Technologies Pty Ltd

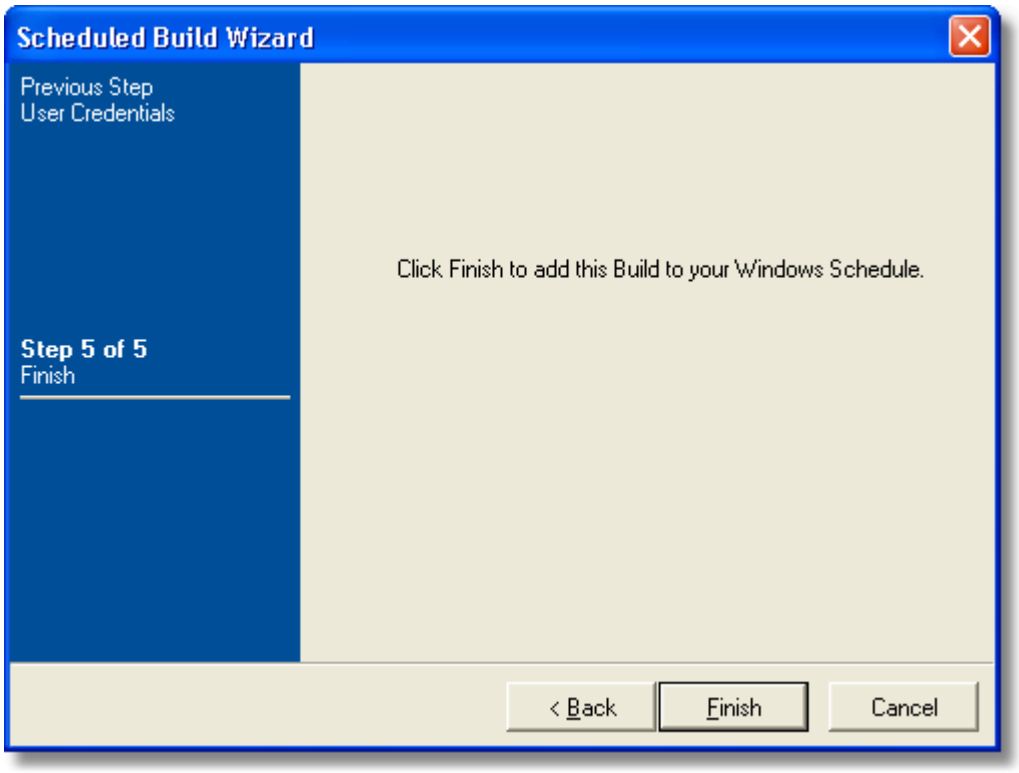


Provide the user name that the build will run as and the password, then click continue.



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.

© 2001-2006 VSoft Technologies Pty Ltd



Click Finish to add this Build to your Scheduled Tasks.

© 2001-2006 VSoft Technologies Pty Ltd

7.1.2 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

-n or /n - Hide the splash screen when starting up

-r or /r - auto build the project file passed in on the command line

-e or /e - exit when done building , only valid when autobuild switch is included

-f or /f - don't exit if error occurs , only valid when autobuild switch is included

-v or /v - set variables - for multiple variables, separate the name/value pairs with semi-colons.

-a or /a - allow interactive actions (eg. Prompt For Variable, Ask Question, etc.)

eg. :
/vOutputDIR=d:\temp\buildfiles;DCUDIR=d:\temp\dcu

Note that if the variable value has spaces in it then enclose the value in double quotes,
eg. :
/vOutputDIR="d:\temp\build files" .

If you do not included the double quotes your build may fail due to incorrect variable values. Separate multiple name=value pairs with semicolons.

Note that variables set from the command line must already be defined as project or user variables. This option is only valid when specified with the /r (auto run).

When running FinalBuilder as a scheduled task, you should always use the /n /r /e switches.

7.1.2.1 Exit Codes

The FinalBuilder IDE uses the following exit codes when run from external programs 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.
20	Invalid project when Auto run project with "Exit when done"

See the Command Line exe topic to see the extended range of exit codes available from FBCMD.exe.

See Also

IDE Command Line Interface | Scheduling Builds

© 2001-2006 VSoft Technologies Pty Ltd

7.2 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.

Running your build process with FBCMD can be faster than running your build in the IDE, as the IDE has more overhead as it repaints the window with logging information and build status information.

The command line syntax for FBCMD is as follows:

FBCMD.exe [options**] /P**<projectfile>

Options :

- **/L** Output messages to logfile instead of console (see below)
- **/N** Do not overwrite specified log file
- **/B** Supress FinalBuilder banner
- **/I** Ignore all system message (eg. Ctrl-C, Shutdown)
- **/A** Allow interactive actions
- **/V** Set FinalBuilder variables (see below)
- **/C** Validate (check) project before execution
- **/XL** Exclude Actionlist messages from log
- **/XA** Exclude Action start messages from log
- **/XM** Exclude Action output from log
- **/P** FinalBuilder project file (see below)
- **/S** Disable saving log to log archive file
- **@** Use a parameter file (see below)

Output to Logfile: /L

Syntax: /L<logfile>
Example: /Lc:\temp\mylogfile.txt
Directs all output to a logfile instead of the console
Can be used with /N to prevent any existing log file being overwritten

Set Variable: /V

Syntax: /V<variable>=<value>;<variable>=<value>
Example: /VMAJORVER=3;MINORVER=1
Sets FinalBuilder variables to specific values before the build starts

Project File: /P

Syntax: /P<projectfile>
Example: /P"c:\build\My Project.fbz4"
Specifies the FinalBuilder project file to build

Parameter File: @

Syntax: @<parameterfile>
Example: @"c:\build\BuildParams.txt"
Specifies a parameter file. Each line of the parameter file specifies a different parameter. Parameters on the command line override the param file.

© 2001-2006 VSoft Technologies Pty Ltd

Exit codes:

Exit Code	Reason
0	ERR_OK
1	ERR_BUILDFAILED
2	ERR_INVALIDARGS
3	ERR_FILENOTFOUND
4	ERR_EXPECTEDLOGFILE
5	ERR_LOGFILEPROBLEM
6	ERR_INVALIDPROJECT
7	ERR_PROJTHREADEXCEPTION
8	ERR_RUNABORTED
9	ERR_PROJECTLOADERROR
10	ERR_INITERROR
11	ERR_VALIDATIONERROR
12	ERR_PACKAGEERROR
13	ERR_REQUIREDPACKAGENOTLOADED
14	ERR_ERRORSETTINGVARIABLE
15	ERR_UNHANDLEDEXCEPTION
16	ERR_PARAMFILEMISSING
17	ERR_PARAMFILEERROR
18	ERR_WRONGEDITION

© 2001-2006 VSoft Technologies Pty Ltd

8 Reference

8.1 Regular Expression Reference

Introduction

Regular Expressions are a widely-used method of specifying patterns of text to search for. Special metacharacters 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 really they are very simple (well, usually simple ;)), handly and powerfull tool.

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://regexpstudio.com

Simple matches

Any single character matches itself, unless it is a metacharacter 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 metacharacters or escape sequences to be interpreted literally by 'escaping' them by preceding them with a backslash "\", for instance: metacharacter "^" match beginning of string, but "\^" match character "^", "\\" match "\" and so on.

Examples:

```
foobar          matchs string 'foobar'
\^FooBarPtr     matchs '^FooBarPtr'
```

Escape sequences

Characters may be specified using a escape sequences syntax much like that used in C and Perl: "\n'' matches a newline, "\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:

© 2001-2006 VSoft Technologies Pty Ltd

foo\x20bar matchs 'foo bar' (note space in the middle)
\tfoobar matchs '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] matchs 'a', 'z' and '-'

[az-] matchs 'a', 'z' and '-'
[a\-z] matchs 'a', 'z' and '-'
[a-z] matchs all twenty six small characters from 'a' to 'z'
[\n-\x0D] matchs any of #10,#11,#12,#13.
[\d-t] matchs any digit, '-' or 't'.
[]-a] matchs any char from ']'..'a'.

Metacharacters

Metacharacters are special characters which are the essence of Regular Expressions. There are different types of metacharacters, described below.

Metacharacters - line separators

^ start of line
$ end of line
\A start of text
\Z end of text
. any character in line

Examples:
^foobar matchs string 'foobar' only if it's at the beginning of line
foobar$ matchs string 'foobar' only if it's at the end of line
^foobar$ matchs string 'foobar' only if it's the only string in line

foob.r matchs strings like 'foobar', 'foobbr', 'foob1r' and so on

The "^" metacharacter by default is only guaranteed to match at the beginning of the

© 2001-2006 VSoft Technologies Pty Ltd

input string/text, the "$" metacharacter only at the end. Embedded line separators will not be matched by "^'' or "$''.
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 ".'' metacharacter 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 www.unicode.org (http://www.unicode.org/unicode/reports/tr18/):

"^" 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.

"." matchs 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) doesnot match the empty string within the sequence \x0D\x0A, but matchs the empty string within the sequence \x0A\x0D.

Multiline processing can be easely 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!

Metacharacters - predefined classes

\w	an alphanumeric character (including "_")
\W	a nonalphanumeric
\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 matchs strings like 'foob1r', ''foob6r' and so on but not 'foobar', 'foobbr' and so on

foob[\w\s]r matchs strings like 'foobar', 'foob r', 'foobbr' and so on but not 'foob1r', 'foob=r' and so on

© 2001-2006 VSoft Technologies Pty Ltd

TRegExpr uses properties SpaceChars and WordChars to define character classes \w, \W, \s, \S, so You can easely redefine it.

Metacharacters - 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.

Metacharacters - iterators

Any item of a regular expression may be followed by another type of metacharacters - iterators. Using this metacharacters You can specify number of occurences of previous character, metacharacter or subexpression.

* 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 matchs strings like 'foobar', 'foobalkjdflkj9r' and 'foobr'

foob.+r matchs strings like 'foobar', 'foobalkjdflkj9r' but not 'foobr'
foob.?r matchs strings like 'foobar', 'foobbr' and 'foobr' but not 'foobalkj9r'
fooba{2}r matchs the string 'foobaar'
fooba{2,}r matchs strings like 'foobaar', 'foobaaar', 'foobaaaar' etc.
fooba{2,3}r matchs strings like 'foobaar', or 'foobaaar' but not 'foobaaaar'

A little explanation about "greediness". "Greedy" takes as many as possible, "non-greedy" takes as few as possible. For example, 'b+' and 'b*' applied to string 'abbbbc' return 'bbbb', 'b+?' returns 'b', 'b*?' returns empty string, 'b{2,3}?' returns 'bb', 'b{2,3}' returns 'bbb'.

You can switch all iterators into "non-greedy" mode (see the modifier /g).

© 2001-2006 VSoft Technologies Pty Ltd

Metacharacters - 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) matchs strings 'foobar' or 'foofoo'.

Metacharacters - subexpressions

The bracketing construct (...) may also be used for define r.e. subexpressions (after parsing You can find subexpression positions, lengths and actual values in MatchPos, MatchLen and Match properties of TRegExpr, and substitute it in template strings by TRegExpr.Substitute).

Subexpressions are numbered based on the left to right order of their opening parenthesis.
First subexpression has number '1' (whole r.e. match has number '0' - You can substitute it in TRegExpr.Substitute as '$0' or '$&').

Examples:
  (foobar){8,10} matchs strings which contain 8, 9 or 10 instances of the 'foobar'
  foob([0-9]|a+)r matchs 'foob0r', 'foob1r' , 'foobar', 'foobaar', 'foobaar' etc.

Metacharacters - backreferences

Metacharacters \1 through \9 are interpreted as backreferences. \<n> matches previously matched subexpression #<n>.

Examples:
  (.)\1+   matchs 'aaaa' and 'cc'.
  (.+)\1+   also match 'abab' and '123123'
  (['"]?)(\d+)\1 matchs '"13" (in double quotes), or '4' (in single quotes) or 77 (without quotes) etc

Modifiers

© 2001-2006 VSoft Technologies Pty Ltd

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 metacharacter introducing a comment, for example:

```
(
(abc) # comment 1
  |   # You can use spaces to format r.e. - TRegExpr ignores it
```

© 2001-2006 VSoft Technologies Pty Ltd

(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 into r.e. for modifying modifiers by the fly. If this construction inlined into subexpression, then it effects only into this subexpression

Examples:
(?i)Saint-Petersburg matchs 'Saint-petersburg' and 'Saint-Petersburg'
(?i)Saint-(?-i)Petersburg matchs 'Saint-Petersburg' but not 'Saint-petersburg'
(?i)(Saint-)?Petersburg matchs 'Saint-petersburg' and 'saint-petersburg'

((?i)Saint-)?Petersburg matchs '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.

© 2001-2006 VSoft Technologies Pty Ltd

8.2 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

Notes:
The section name must be unique and is required.
Each name in a section must be unique, and must contain a = sign

© 2001-2006 VSoft Technologies Pty Ltd

9 Support

9.1 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 mutliple 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.)

Borland 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 messagebox or dialog and of course there is no way to respond to it while it is being run unattended.

Workaround : None known. Other than to use a different installer, one that fully supports automated builds such as Inno Setup or InstallShield. This problem may be solved with Wise 9.02, however we have not confirmed this.

Workaround : Use VBScript or JavaScript.

Borland C++ Builder Action

Problem: Some static Lib projects will not link correctly when compiled from FinalBuilder. You might see something like this :

```
J:\Borland\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.@@@'
```

© 2001-2006 VSoft Technologies Pty Ltd

** 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.

© 2001-2006 VSoft Technologies Pty Ltd

9.2 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.

© 2001-2006 VSoft Technologies Pty Ltd

9.3 FinalBuilder Support

VSoft Technologies provide support for FinalBuilder™ on our newsgroup.

News Server : nntp://news.FinalBuilder.com
Web forum interface : http://forums.FinalBuilder.com/

Support Email : support@FinalBuilder.com

© 2001-2006 VSoft Technologies Pty Ltd

Index

- . -

.Net 3rd Party Tools
  Demeanor Action 585
  Dotfuscator Action 584
  FxCop Action 584
  XenoCode Action 585
.Net Framework Tools
  Register Assembly in COM 564
  Run AL.EXE Action 563
  Run ASPNET_REGIIS.EXE Action 565
.Net SDK Tools
  Extract Public Key Action 575
  GAC Download Cache Action 580
  GAC Install Action 577
  GAC Uninstall Action 578
  Generate Key Pair Action 572
  Install Key in Container Action 573
  Re-sign Assembly Action 576
  Run ResGen.exe Action 582
  Run SN.EXE Action 576
  Type Library Export Action 581
  Type Library Import Action 580
  Verify Strong Name Action 572

- A -

Action 13
Action Behaviour 69
Action Group Action 309
Action List 13
  Exit Action List 75
  Run Action List 74
Action List Parameters 64
Action Lists 6, 7, 11
Action Properties 69
Action Types 7, 10
Actions 6, 13
Actions Summary 66
Active Script 38
Add Action List 11
Adding & Editing Variables 58
ADO Execute SQL Action 616
ADO Execute Stored Procedure Action 617
Ant Project Action 604
AQTest Action 541
Archiving
  7Zip 535
  Create Archive 535
  Create Zip File Action 530
  Delete from Archive 540
  Extract Archive 538
  Extract Zip File Action 533
  List Archive 540
  Test Archive 538
  Update Archive 539
  WinRAR Action 534
Armadillo 552
Ask Question Action 303
ASProtect 550
Assembly Info Updater Action 465
Async Action Group 90
Async Action Groups 90
Authenticode Action 357
AutomatedQA Test Complete Actions 543
Automating FinalBuilder
  Command Line Interface 694
  Console Build Tool 695
  Exit Codes 694
  Scheduling Builds 690
Auxillary project files 33

- B -

Backup Database Action 610
Beep Action 310
Borland C# Builder Compiler Action 463
Borland C++ Builder Action 455
Borland Java Compiler 429
Borland Project Group 674
Borland Resource Compiler Action 454
Build History 22
Build Log 7, 21
Build Tools
  Ant Project Action 604
  MSBuild Project Action 606
  NAnt Project Action 605
Burn CD Action 554
Burn ISO Action 557

- C -

C# Builder Compiler Action 463
C# Compiler Action 444
C# Project Compiler Action 444

© 2001-2006 VSoft Technologies Pty Ltd

C++ Builder Compiler Action 455
Calculate File CRC32 Action 365
Calculate File MD5 Action 367
Case Action 83
Case Statement
  Case Action 83
  Switch Action 82
Catch Action 84
Category 10
CD/DVD Burning
  Burn CD Action 554
  Burn ISO Action 557
  Check Ready Action 558
  Create ISO Action 556
  Erase CDRW Action 559
Check Catalogue Action 612
Check Database Action 612
Check File Exists Action 338
Check for Updates 15, 32
Check Ready Action 558
Chrome Compiler Action 467
CityDesk Action 316
Clear Build History 22
CodeHealer 619
Command Line Interface 694
  Console Build Tool 695
Comment Action 314
Compile Borland Resource Script Action 454
Compile Delphi 8 for .Net Project Action 463
Compile Delphi Project Action 450
Compile Visual Basic Project 430
Compilers
  Borland C# Compiler Action 463
  Borland C++ Builder 455
  Borland Delphi 450
  Borland Delphi 8 for .Net 463
  Borland Resource 454
  Chrome Action 467
  IncrediBuild Action 468
  Java 427
  Microsoft C# Compiler Action 444
  Microsoft C# Project Compiler Action 444
  Microsoft J# Compiler Action 448
  Microsoft J# Project Compiler Action 448
  Microsoft VB.NET Compiler Action 446
  Microsoft VB.NET Project Compiler Action 446
  Visual Basic 430
  Visual C++ 6 435
  Visual Studio .NET 439
Concatenate Files Action 355
Control Service Action 383
Copy Files Action 345
Copy/Move File List Action 349
Copying Actions 13
Create Archive 535
Create Directory Action 341
Create ISO Action 556
Create Text File Action 359
Create Zip File Action 530
CVS Actions 189
CVS Command 190

- D -

Database
  ADO Execute SQL Action 616
  ADO Execute Stored Procedure Action 617
  SQL Server Backup Database Action 610
  SQL Server Check Catalogue Action 612
  SQL Server Check Database Action 612
  SQL Server DTSRun Action 609
  SQL Server Execute SQL Action 607
  SQL Server Rebuild Indexes Action 613
  SQL Server Remove Unused Space Action 611
  SQL Server Update DB Statistics Action 613
DateTime Action 312
Delay Action 71
Delete Directory Action 339, 342
Delete Files Action 339
Delete from Archive 540
Delete Log Entry 22
Deleting Actions 13
Delphi 8 for .Net Compiler Action 463
Delphi Compiler Action 450
Demeanor Action 585
Doc-O-Matic 517
DOS Command Action 376
Dotfuscator Action 584
DTSRun Action 609

- E -

Edit XML File Action 596
Else Action 79
EMail Action 492
End Action 84
Environment Variables 62
Erase CDRW Action 559
Execute Condition 50

© 2001-2006 VSoft Technologies Pty Ltd

Execute Program Action 374
Execute SQL Action 607
Exit Action List Action 75
Exit Codes 694
Export Log 22, 311
Extract Archive 538
Extract File Version Action 364
Extract Public Key Action 575
Extract XML Fragment Action 594
Extract Zip File Action 533

- F -

FAQ 707
File Contents Iterator Action 294
File Iterator Action 290
File Types 33
Files & Directories
  Authenticode Action 357
  Calculate File CRC32 Action 365
  Calculate File MD5 Action 367
  Check File Exists Action 338
  Concatenate Files Action 355
  Copy Files Action 345
  Copy/Move File List Action 349
  Create Directory Action 341
  Create Text File Action 359
  Delete Directory Action 342
  Delete Files Action 339
  Extract File Version Action 364
  Move Directory Action 344
  Move Files Action 347
  Read Text File Action 361
  Rename File or Directory Action 356
  RoboCopy Actions 363
  Set File Attributes Action 351
  Text Replace Action 353
  Touch Files Action 352
  Write Text File Action 360
  XCopy Action 362
FileSet Copy/Move 139
FileSet Define 132
FileSet Delete 141
FileSet Filters 134
FileSet log Files 137
FileSet Patterns 132
FileSet Refresh 138
FileSet to Variable 136
FileSets 129
  Accessing FileSets 130
Filter 10
Filter Log By Action 21
FinalBuilder System Variables 63
Finally Action 84
Flow Control
  Case Action 83
  Catch Action 84
  Conditional Execution 50
  Delay Action 71
  Else Action 79
  End Action 84
  Exit Action List 75
  Finally Action 84
  If .. Then Action 76
  If Prev Action Failed Action 78
  Include FinalBuilder Project 72
  Iterators 288
  Parallelization 90
  Raise Exception Action 81
  Run Action List 74
  Stop Build Action 80
  Switch Action 82
  Try Action 84
  Waiting For Events 94
  While Loop 89
Folder Iterator Action 293
FTP Client 505
FxCop Action 584

- G -

GAC Download Cache Action 580
GAC Install Action 577
GAC Uninstall Action 578
Generate Key Pair Action 572
Get DateTime Action 312
Get Disk Free Space Action 381
Getting Started 15
Global Script Functions 40
GPInstall Action 481

- H -

Help & Manual 522
Help Compilers
  Doc-O-Matic 517
  Help & Manual 522
  HTML Help 521
  NDoc 524
  WinHelp 519

© 2001-2006 VSoft Technologies Pty Ltd

HTML Help 521
HTTP Get Action 510

- I -

ICQ Action 511
IDE 7
If .. Then Action 76
If Prev Action Failed Action 78
Include FinalBuilder Project Action 72
IncrediBuild Action 468
INI File Iterator Action 295
Ini Files
  Load Variables to INI Action 327
  Read 525
  Save Variables to INI Action 325
  Write 526
InnoSetup Action 480
Install Key in Container Action 573
InstallAnywhere .Net Action 483
InstallAnywhere Enterprise Action 482
InstallAware Action 484
Installers
  GPInstall Action 481
  InnoSetup Action 480
  InstallAnywhere .Net Action 483
  InstallAnywhere Enterprise Action 482
  InstallAware Action 484
  InstallSheild Pro Action 473
  InstallSheild Pro Windows Installer Editon Action 476
  InstallShield Developer Action 477
  InstallShield Universal Action 479
  NSIS Action 491
  Wise for WIndows Installer Action 471
  Wise InstallBuilder/InstallMaster Action 469
InstallSheild Pro Action 473
InstallSheild Pro Windows Installer Editon Action 476
InstallShield Developer Action 477
InstallShield Universal Action 479
Interactive
  Ask Question Action 303
  Multi Question Action 304
  Prompt for File or Directory 306
  Prompt for Variables 297
  Prompt for Variables (Enhanced) 299
Internet
  FTP Client 505
  HTTP Get Action 510
  ICQ Action 511
  NNTP News Post Action 516
  Send Email Action 492
  Telnet Action 509
Iterators
  File Contents Iterator 294
  File Iterator 290
  Folder Iterator 293
  For Loop 87
  General Information 288
  INI File Iterator 295
  List Iterator 292
  While Loop 89

- J -

J# Compiler Action 448
J# Project Compiler Action 448
Java Compiler Action 427
Java JDK Configuration 429
JavaScript 15, 38
JDK Configuration 429
JScript 15, 38

- K -

Known Problems 705

- L -

License 3
Licensing
  Amradillo 552
  ASProtect 550
  ProActivate 551
List Archive 540
List Iterator Action 292
Live Logging 21
Load Variables From INI Action 327
Log Action 311
Loop 290, 292, 293, 294, 295

- M -

MadExcept Action 464
Main 6, 11
Merge XML Action 593
Messages 29
Microsoft C# Compiler Action 444
Microsoft C# Project Compiler Action 444
Microsoft J# Compiler Action 448

© 2001-2006 VSoft Technologies Pty Ltd

Microsoft J# Project Compiler Action 448
Microsoft VB.NET Compiler Action 446
Microsoft VB.NET Project Compiler Action 446
Move Directory Action 344
Move Files Action 347
Moving Actions 13
MSBuild Project Action 606
MSTest Action 547
Multi Question Action 304

- N -

NAnt Project Action 605
NDoc 524
Net Send Message Action 380
NNTP News Post Action 516
NullSoft NSIS Action 491
NUnit Action 546

- O -

OnFailure 6, 11
Options 52
Options Dialog 25
Output Tree 21
Overview 1

- P -

Parallel Build Process 90
Pascal Analyzer Action 621
Perforce Command 188
Perforce Sync with View 186
Plugins 34
ProActivate 551
Project file types 33
Project Summary 28
Project Variables 60
Prompt for File or Directory Action 306
Prompt for Variables (Enhanced) Action 299
Prompt for Variables Action 297

- Q -

Quick help 15
QVCS
  Add File 194
  Check In 194
  Check Out 194
  Get Latest Version 194
  Label Files 194
  Undo Checkout 194

- R -

Raise Exception Action 81
RAR Files 534
Read Ini 525
Read Text File Action 361
Rebuild Indexes Action 613
Register Assembly in COM 564
Register DLL/OCX Action 382
Registry
  Delete Registry Value 527
  Read Registry Value 527
  Set Registry Value 527
Remove Unused Space Action 611
Rename File or Directory Action 356
Re-sign Assembly Action 576
Resource Script Compiler Action 454
RoboCopy Actions 363
Run Action List Action 74
Run AL.EXE Action 563
Run ASPNET_REGIIS.EXE Action 565
Run Command 19
Run ResGen.exe Action 582
Run Scipt Action 315
Run SN.EXE Action 576

- S -

Save Variables To INI Action 325
Scheduling Builds 690
Script Editor 15
Script Functions 40
Scripting 38, 55
  Accessing Options 52
  Accessing TStrings 55
  Events 46
  Global Functions 40
Search 10
Selecting Actions 13
Send Email 492
Send Message Action 380
Set File Attributes Action 351
Set Variable Action 321
Shell Execute Action 391
SMTP 492
Sound 310

© 2001-2006 VSoft Technologies Pty Ltd

Source Code Tools
CodeHealer 619
Pascal Analyzer Action 621
Source Safe
Add Files 180
Branch 181
Check File Status 178
Check In 173
Check Out 179
Checkout 174
Get Working Directory 183
GetLatest Version 175
Label Files 176
Override Global Options 183
Project Checkouts 177
Share 182
SQL Server
Backup Database Action 610
Check Catalogue Action 612
Check Database Action 612
DTSRun Action 609
Execute SQL Action 607
Rebuild Indexes Action 613
Remove Unused Space Action 611
Update DB Statistics Action 613
StarTeam
Add Files 193
Apply Label 192
Check In 192
Check Lock/Unlock Files 192
Check Out 192
Create Label 192
Delete Files 193
List Files 193
Update Status 193
Stop Build Action 80
Stop Command 19
Stopping a build 19
Subst Drive Action 378
Subversion
Add 238
Checkout 238
Cleanup 238
Commit 238
Copy 238
Export 238
Generic 238
Import 238
MkDir 238
Status 238
Update 238
Support 708
Switch Action 82
System Tray 30
System Variables 63

- T -

Task Scheduling
"AT" Scheduler Actions 422, 424, 425
Add Scheduled Task 415
Delete Scheduled Task 418
End Scheduled Task 420
Run Scheduled Task 419
Schedule Add 422
Schedule Delete 424
Schedule List 425
Scheduling Remote Tasks 420
Windows Scheduler Actions 415, 418, 419, 420
Team Coherence
Attach Label 202
Check In 199
Check Out 200
Connect 197
Create Label 204
Create View 207
Delete View 208
Detach Label 203
Get 199
Promote 204
Set View 198
Sync 205
Update View 208
Telnet Action 509
Terminate Command 19
Test Archive 538
Testing Tools
AQTest Action 541
AutomatedQA Test Complete Actions 543
MSTest Action 547
NUnit Action 546
Text Replace Action 353
Touch Files Action 352
Transform XML Action 592
Tray Icon 30
Try Action 84
TStrings 55
Type Library Export Action 581
Type Library Import Action 580

© 2001-2006 VSoft Technologies Pty Ltd

- U -

Update Archive 539
Update DB Statistics Action 613
Updates 15, 32
User Action Lists 6
User Variables 61
Using Variables 59

- V -

Validate XML Action 598
Validation 22
Variables
  Action List Parameters 64
  Adding & Editing Variables 58
  Environment Variables 62
  Load from INI Action 327
  Overview 56
  Predefined FinalBuilder Variables 63
  Project Variables 60
  Save To INI Action 325
  Set Variable Action 321
  System Variables 63
  User Variables 61
  Using Variables 59
Vault
  Add 219
  Branch 213
  Check In 215
  Check Out 211
  Cloak 217
  Commit 215
  Create Folder 220
  Create Label 223
  Delete File/Folder 221
  Diff 232
  File Status 234
  Get 211
  Get Label 230
  Get using Wildcards 212
  Get Version 222
  GetLabelDiffs 233
  List Checkouts 233
  Move File/Folder 224
  Pin 228
  Rename File/Folder 225
  Set Working Folder 231
  Share File/Folder 226
  UnCloak 218
  Undo Checkout 226
  UnPin 229
VB.NET Compiler Action 446
VB.NET Project Compiler Action 446
VB6 Project Group 681
VBScript 15, 38
Verify Strong Name Action 572
Version Control
  CVS Actions 189
  CVS Command 190
  Perforce Command 188
  Perforce Sync with View 186
  QVCS 194
  Source Safe 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183
  StarTeam 192, 193
  Subversion 238
  Team Coherence 197, 198, 199, 200, 202, 203, 204, 205, 207, 208
  Vault 211, 212, 213, 215, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 228, 229, 230, 231, 232, 233, 234
Version History 2
View Log 22
Visual C++ 6 435
Visual Studio .NET 439

- W -

Waiting For Events
  Certain Date or Time 108
  Command Output 106
  Files 95
  FinalBuilder Variables 97
  Ini File Change 102
  Overview 94
  Registry Change 104
  Remote Computer 101
  Script Result 100
Watches 18
While Loop Action 89
Window Exists Action 379
WinHelp 519
WinRAR Action 534
Wise for WIndows Installer Action 471
Wise Installer Action 469
Wizards
  Borland Project Group 674
  VB6 Project Group 681
WMI Kill Process Action 386
WMI Process Info Action 389

© 2001-2006 VSoft Technologies Pty Ltd

WMI Run Process Action 385
Write Ini 526
Write Text File Action 360

- X -

XCopy Action 362
XenoCode Action 585
XML
Edit XML File Action 596
Extract XML Fragment Action 594
Merge XML Action 593
Transform XML Action 592
Validate XML Action 598

- Z -

Zip Files 530, 533, 535, 538, 539, 540

© 2001-2006 VSoft Technologies Pty Ltd